

Lerarenhandleiding

Auteur: Natacha Gesquière

Coauteurs: Sofie Meeus, JarVan den Bulcke, Franciswyf els

KKKS

“AI IS THE NEW ELECTRICITY”.

–ANDREW NG

Versie 1.0

Natacha Gesquière

Kunstmatige Intelligentie, Klimaatverandering, Stomata: KIKS

Coauteurs: Sofie Meeus, Jan Van den Bulcke, Francis wyffels

CC BY-SA, 2020, Natacha Gesquière

Coauteurs: Sofie Meeus, Jan Van den Bulcke, Francis wyffels

Corrector: Annick Dehennin

Illustrator: Margot De Saegher

KIKS is een Smart Education @ Schools project met steun van de Vlaamse Overheid en imec.

Het KIKS-project is eerste laureaat van de Koningin Paolaprijs voor het Onderwijs 2020.

Deze handleiding kwam tot stand met (financiële of logistieke) steun van de Universiteit Gent, Sint-Bavohumaniora, RVO-society, Dwengo vzw, de Provincie Oost-Vlaanderen, de Provincie Vlaams-Brabant, de Vlaamse Overheid en imec.

D/2020/Dwengo/2

ISBN 9789081991780

NUR 950

Eerste druk, september 2020



INHOUDSOPGAVE

1	Klimaatverandering	13
2	Stomata	35
3	Hoe passen bomen uit het tropisch regenwoud zich aan aan de klimaatverandering?	63
4	Burgerwetenschap	75
5	Artificiële intelligentie	81
6	Fundamenten van machinaal leren	119
7	KIKS in de klas	133
8	De notebooks van KIKS	139
9	Digitale beelden	145

10 Tensoren	153
11 Basisconcepten van machinaal leren in de praktijk	169
12 Convoluties	231
13 ReLU en max pooling	241
14 Implementatie in Keras	249
15 Eindtermen	277
Index	303

Versie 1.0

Met dank aan

Dr. Pieter Tans (NOAA),

Lander Van Tricht (VUB),

Joris Meys (UGent),

Prof. dr. Philippe Huybrechts (VUB, IPCC),

Prof. dr. Herman Callaert (UHasselt),

Prof. dr. Tine De Moor (Universiteit Utrecht),

Dr. Miguel Camargo (INPA Brazilië),

Ir. Olivier Pieters (UGent),

Ir. Tom Neutens (UGent),

Ing. Alexander Meheus,

Prof. dr. Ann Dooms (VUB),

Dr. Karel Vermeulen (UGent),

Dr. Annelies Duerinckx (Scivil),

Jef Van Laer (Scivil),

Liesbeth Gijssels (EOS),

Dr. ir. Lieven Depuydt (Accenture),

Cathérine Boitsios (Sint-Bavohumaniora),

Dr. Carla Van Dorpe (Sint-Bavohumaniora),

Dr. Els Van Hecke (Sint-Bavohumaniora),

Bart Van Lancker (Sint-Bavohumaniora),

Zimcke Van de Staey (UGent).

INLEIDING

Kunstmatige intelligentie, Klimaatverandering, Stomata of kortweg KIKS, is een STEM-project rond artificiële intelligentie (AI) voor de derde graad secundair onderwijs. STEM staat voor *Science, Technology, Engineering, Mathematics*. Leerlingen leren artificiële intelligentie (AI) begrijpen, met mogelijkheden en beperkingen; ze leren hoe ze er een impact kunnen op hebben. De relatie tussen huidmondjes (stomata) van planten en de klimaatverandering biedt een uniek kader om met diepe neurale netwerken aan de slag te gaan. De programmeertaal Python is bovendien zeer toegankelijk als tool om de fundamentele van neurale netwerken te bestuderen.

Een troef van het KIKS-project is de intense samenwerking tussen onderzoekers en leerkrachten. Het lesmateriaal van KIKS wordt immers ontwikkeld, parallel met de resultaten van een lopend wetenschappelijk onderzoek aan de UGent en de Plantentuin Meise.

Bovendien behandelt KIKS eindtermen en leerplandoelen van biologie, aardrijkskunde, wiskunde, statistiek en chemie. Computerwetenschappen komen ruim aan bod, burgerwetenschap speelt een rol en er is behoorlijk wat aandacht voor ethiek.

De notebooks reiken toepassingen aan binnen een context van de klimaatverandering en van planten, en het

verband tussen beide. De fundamentele begrippen uit machine learning en deep learning worden daarbij aangeleerd.

De notebooks zijn interactief opgebouwd. Er worden nieuwe begrippen in aangebracht via tekstuele uitleg, video en afbeeldingen, geleide voorbeelden, kleine en grotere opdrachten.

Met deze handleiding voor de leerkracht willen we de leerkrachten die met KIKS aan de slag gaan in de klas, achtergrondkennis geven over de leerinhouden van dit project.

De klimaatverandering, de biologie van de huidmondjes en de manier waarop planten zich via de huidmondjes aan die klimaatverandering aanpassen, worden uitgelegd in de eerste twee hoofdstukken. Het derde hoofdstuk behandelt het wetenschappelijk onderzoek van de UGent en de Plantentuin Meise dat aan de grondslag ligt van dit project. In het vierde hoofdstuk wordt het aspect van burgerwetenschap belicht. Wat artificiële intelligentie (AI) is, de geschiedenis van dit domein van de computerwetenschappen, hoe het gebruikt wordt en de ethiek eromheen, vindt u in hoofdstuk 5. In de hoofdstukken die volgen, worden dan de principes van digitale beelden, de wiskunde achter de algoritmen van machinaal leren en de fundamenten van de momenteel meest gebruikte AI-technieken uit de doeken gedaan. We geven aan hoe wij met KIKS aan de slag gegaan zijn in de klas. Achteraan vindt u de eindtermen die aan bod komen.

KLIMAATVERANDERING

1.1 *Paleoklimatologie*

Paleoklimatologen bestuderen het klimaat uit (lang) vervlogen tijden. Een schat aan informatie is te vinden in sedimenten op de bodem van de oceanen, is opgeslagen in gletsjers en bewaard in bomen. In die zogenaamde proxydata zoals boomringen, ijskernen, en oceaansedimenten maar ook fossiele pollen, rotsformaties in grotten, koralen, vindt men fysieke elementen van de omgeving terug die gecorreleerd zijn met het toenmalige klimaat. Deze natuurlijke proxydata vult men aan met historische data. Door de gegevens afkomstig van de proxydata te analyseren en te combineren, kan men het klimaat uit een ver verleden reconstrueren (NCEI, 2019).

- Historische data zoals observaties van weer en klimaat zijn bv. terug te vinden in logboeken van schepen, logboeken van landbouwers en reisdagboeken (zie Figuur 1.1). Ze verstrekken zowel kwalitatieve als kwantitatieve informatie. Wetenschappers hebben zo uit de data van de druivenpluk zomertemperaturen in Parijs van 1370 tot 1879 kunnen reconstrueren.
- Koralen zijn opgebouwd uit calciumcarbonaat (CaCO_3).

Proxydata zijn gegevens geleverd door een proxy, in dit geval een klimaatproxy. Boussemaere (2015) schrijft: "Een klimaatproxy is een bron aan de hand waarvan men kan inschatten hoe het klimaat er in het verleden uitzag en evolueerde. Een proxy meet de temperatuur of andere gegevens op een indirecte manier".

Thomas Jefferson, president van de VS van 1801 tot 1809, hield nauwgezet gegevens over het weer bij in een logboek van 1776 tot 1818 (Nese & Schwartz, 2002).

Het bevat bestanddelen zoals zuurstofisotopen en sporen van metalen, waaruit men de temperatuur van het water kan bepalen waarin het koraal groeide.

- De gelaagde sedimentafzetting op de bodem van meren en oceanen bevat (dierlijke en plantaardige) fossielen en chemische stoffen, die informatie verschaffen over het klimaat.
- Stuifmeelkorrels worden eveneens goed bewaard in de lagen sediment op de bodem van meren en oceanen. Men gebruikt ze om te bepalen welke planten wanneer hebben gegroeid. Alle bloeiende planten produceren stuifmeel en uit de vorm van de stuifmeelkorrel kan men het type plant afleiden.
- Stomata op soms heel kleine gefossiliseerde bladeren geven informatie over het CO₂-gehalte in de atmosfeer toen die planten groeiden (zie Figuur 1.2). Zowel het aantal stomata al grootte van de stomata zijn van belang.
- Bij de polen en hoog in de bergen zijn dikke pakken ijs te vinden. Deze pakken zijn ontstaan door sneeuwval over duizenden jaren. Sneeuw is luchtig, het bevat veel lucht. Gevallen sneeuw wordt door nieuwe sneeuw die later valt, samengeperst. Als de onderste laag samengeperste sneeuw ongeveer 30 jaar oud is, wordt de druk groot genoeg om de samengeperste sneeuw om te zetten in ijs. In dat ijs zit nog steeds een deel van de lucht gevangen (Tans, 2018). Zie ook Figuur 1.3. Men boort diep in het ijs om stof, luchtbellen en zuurstofisotopen, gevangen in het ijs, te onderzoeken en er klimaatomstandigheden van een bepaald tijdperk uit af te leiden. De ijskernen bevatten jaarlijks gevormde lagen, die toelaten de ouderdom van het ijs te bepalen (Boussemaere, 2015). Het ijs verschaft informatie over temperatuur, neerslag, samenstelling van de atmosfeer, vulkanische activiteit en wind.

Figuur 1.1: Logboek 'Thomas Jefferson weather record' (NCEI, 2019).



Figuur 1.2: Gefossiliseerd blad uit de *Fossil Lauraceae leaf database van Nieuw-Zeeland*: goed bewaard blad van de site Grey Lake, behorend tot het geslacht *Cryptocarya* (Steinthorsdottir et al., 2019).

- Het klimaat beïnvloedt de groei van bomen en daarom ook de breedte van elke boomring. Meestal produceren bomen een ring per jaar en omdat sommige bomen heel oud worden, kunnen ze met een jaarlijkse nauwkeurigheid informatie verschaffen over het klimaat, gaande van honderden tot duizenden jaren.

Met de proxydata heeft men kunnen aantonen dat het klimaat op aarde steeds verandert. De voorbije twee miljoen jaar heeft de aarde ijstijden en sterk dalende zeespiegels ondergaan. Honderd miljoen jaar geleden was het bijvoorbeeld veel warmer dan nu, met minder poolijs en stijgende zeeniveaus. Maar de laatste 2,5 miljoen jaar was onze wereld 80 % van de tijd kouder dan nu (Maslin, 2014).

Door te begrijpen hoe en waarom het klimaat vroeger geëvolueerd is, ziet men in hoe het huidige klimaat tot stand is gekomen en welke factoren het klimaat kunnen beïnvloeden.

Deze kennis wordt gebruikt om wiskundige modellen te bouwen, die tonen hoe veranderende condities op aarde en in de atmosfeer het klimaat de komende decennia kunnen veranderen.

Paleoklimatologie kan ook helpen om de invloed van de mens op het klimaat te begrijpen en in te schatten hoe groot die invloed is.

1.2 *Atmosferisch CO₂-gehalte en globale temperatuur*

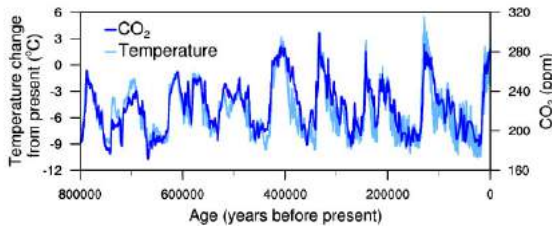
Door het klimaat van vroeger te bestuderen, heeft men bv. ontdekt dat het atmosferisch CO₂-gehalte het klimaat beïnvloedt (Maslin, 2014).

Men deed daarvoor ijsboringen op Antarctica en Groenland. Zo heeft men o.a. het atmosferisch CO₂-gehalte van de voorbije 800 000 jaar kunnen achterhalen door op Antarctica tot meer dan 3 km diep te boren (Boussemaere, 2015). Zie ook Figuur 1.3.



Figuur 1.3: CSIRO. Lucht-bellen in ijs. CC BY. (CSIRO. Atmospheric Research, 2019).

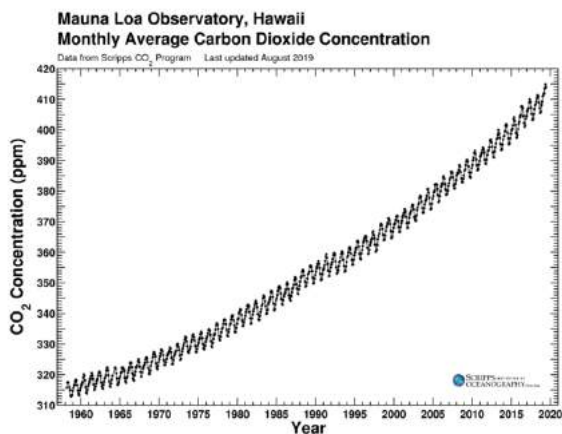
Door het atmosferisch CO₂-gehalte met de temperatuur te vergelijken, kwam men tot de conclusie dat er de laatste 800 000 jaar een opvallend sterke samenhang is tussen beide (zie grafiek in Figuur 1.4).



Figuur 1.4: Temperatuursverandering (lichtblauw) en verandering in CO₂ (donkerblauw) gemeten uit de EPICA Dome C ijskern in Antarctica (NOAA, 2008).

Wetenschapper David Keeling besloot in de jaren 50 om CO₂-metingen te doen, ver van de bewoonde wereld, zodat de meetresultaten niet onbetrouwbaar zouden zijn door CO₂-uitstoot in de buurt (Boussemaere, 2015). In 1958 begon hij met de eerste precisiemetingen van CO₂ op de vulkaan Mauna Loa op Hawaï in een hutje van de 'U.S. Weather Bureau', dat nu overgegaan is in het onderzoekscentrum 'National Oceanic and Atmospheric Administration (NOAA)'. De CO₂-metingen van NOAA begonnen in 1973. Sindsdien zijn er twee onafhankelijke meetprogramma's: door het 'Scripps Institute of Oceanography' en door NOAA (Pieter Tans, persoonlijke communicatie).

De metingen van Keeling vormen samen de Keeling-curve. De Keelingcurve is momenteel een van de belangrijkste krommes in het onderzoek naar de klimaatopwarming (Boussemaere, 2015). In Figuur 1.5 valt het schommelende karakter van deze kromme op, naast een onmiskenbaar opwaartse trend. De schommelingen tijdens het jaar zijn er omdat planten tijdens de groei CO₂ opnemen. Die opname is het grootst tijdens de lente.



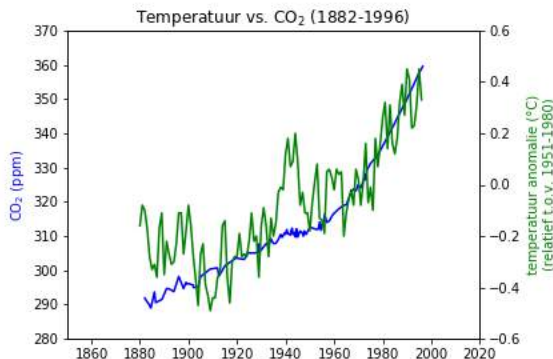
Figuur 1.5: Keelingcurve (Scripps Institution of Oceanography, 2019).

Drie onderzoekscentra, NASA/GISS, HadCRUT en NOAA/NCDC, staan garant voor betrouwbare metingen van de globale temperatuur. Hun data zijn online beschikbaar. Het zijn echter niet de gemeten waarden die de data vormen. De data bevatten de mate waarin de gemeten temperatuur afwijkt t.o.v. een bepaalde referentieperiode. Bv. voor NASA/GISS is dat t.o.v. de periode 1951-1980. Men noemt deze afwijking de temperatuur-anomalie.

Als de temperatuurgegevens en de CO₂-data op eenzelfde grafiek worden uitgezet, bekomt men de grafiek uit Figuur 1.6. Ook hier is de samenhang onmiskenbaar.

Dr. Pieter Tans is hoofd van de 'Carbon Cycle Greenhouse Gases group' van het onderzoekscentrum NOAA. Hij verschafte ons data van CO₂-concentraties van de laatste 2000 jaar, bepaald uit luchtstalen in diep ijs, uit ijskernen ontgonnen op Antarctica.

De data van de metingen van NOAA op Mauna Loa zijn te vinden op de website van NOAA (ESRL/GMD, 2019).



Figuur 1.6: Samenhang verloop temperatuur en CO₂ (1880-1996).

In een eerste notebook 'CO₂' gaan de leerlingen met deze data aan de slag. Door de data te combineren zien ze hoe de CO₂-concentratie geëvolueerd is over de laatste 2000 jaar en in het bijzonder in de periode vanaf de industriële revolutie tot nu. Ze detecteren de kleine ijstijd op deze grafiek. Bovendien ontdekken ze de samenhang tussen het temperatuurverloop en de verandering in CO₂ (zie Figuren 1.6 en 1.9).

In de notebook 'Keelingcurve' bekijken ze de verandering in CO₂ gedurende een dag en gedurende een jaar. Ze leggen de link met fotosynthese. Tot slot maken ze de grafiek van de Keelingcurve (zie Figuur 1.5).



Bekijk de notebooks over CO₂.

De kleine ijstijd

Vóór de industriële revolutie werd de atmosferische CO₂-concentratie vooral beïnvloed door natuurlijke processen, maar was ze wellicht al onderhevig aan menselijke activiteiten zoals ontbossing en landbouw (Ahn et al., 2012; Ruddiman & Ellis, 2009). Een vrij recente periode vertoont een opvallend lagere atmosferische CO₂-concentratie, met een dieptepunt rond 1610: de kleine ijstijd.

De kleine ijstijd is een periode van de 16de tot het midden van de 19de eeuw. De naam is een beetje ongelukkig gekozen, want een ijstijd was het niet. De temperatuur op aarde was toen 0,5 tot 1 °C lager dan normaal, bij echte ijstijden

is dat 4 tot 5 °C. De kleine ijstijd betrof koude periodes verspreid over de wereld, maar deze koude periodes verliepen niet synchroon. De periode is dus niet te vergelijken met een 'echte' ijstijd en er was geen mondiale klimaatverandering zoals nu (Neukom et al., 2019; Tans, 2018).

In Noordwest-Europa en het zuidoosten van Noord-Amerika was het vooral in de 17de eeuw kouder. Veel oogsten mislukten, er was meer melding van honger en ziekte en een hogere kindersterfte (Mann, 2002). De periode werd bv. gekenmerkt door het feit dat de gebergtegletsjers zich uitbreidden. De oprukkende gletsjers hadden een grote impact op de mensen die er in de buurt leefden, sommige populaties werden van de rest van de wereld afgesloten. Stradivarius bouwde zijn violen met hout van bomen die groeiden in de kleine ijstijd. Het hout vertoont daardoor smalle, gelijkmatige ringen en heeft een uniforme houtdichtheid (Trouet, 2020).

1.3 Klimaatverandering

De stralingsenergie van de zon wordt voor ongeveer 30 % teruggekaatst door de atmosfeer, het wolkendek en het aardoppervlak. Men noemt dit het albedo-effect, wat wordt geïllustreerd door de gele pijlen in Figuur 1.8. Het ijs op de polen bv. kaatst veel terug.

De overige 70 % wordt geabsorbeerd door de atmosfeer en het aardoppervlak, waardoor het land, de zeeën en oceanen, en de atmosfeer opwarmen. Die warmte wordt teruggestraald in de vorm van infraroodstraling. Dit wordt in de figuur geïllustreerd door de rode en oranje pijlen. Op de figuur is ook te zien dat enerzijds een deel van de infraroodstraling wordt geabsorbeerd in de atmosfeer en anderzijds dat ook een deel teruggekaatst wordt naar de aarde. De aarde wordt daardoor nog meer opgewarmd. Men spreekt van het broeikaseffect (Boussemaere, 2015).

In het milieuraapport van de Vlaamse Milieumaatschappij (2019) staat het broeikaseffect als volgt beschreven:



Figuur 1.7: © NASA Atmosfeer (NASA, 2012).



Figuur 1.8: Broeikaseffect.

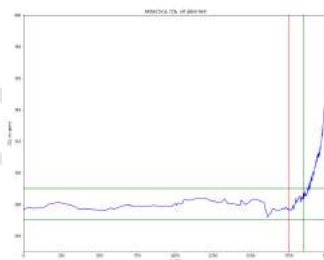
“In de atmosfeer zijn gassen aanwezig die de invallende zonnestraling doorlaten, maar de teruggekaatste straling van het opgewarmde aardoppervlak opnemen.

Dit fenomeen heet het broeikaseffect naar analogie met de werking van glas in een serre. Het leven op aarde dankt zijn bestaan aan dit (natuurlijke) broeikas-effect: de gemiddelde temperatuur op aarde zou anders $-18\text{ }^{\circ}\text{C}$ bedragen, in plaats van de huidige $+15\text{ }^{\circ}\text{C}$.

De voornaamste natuurlijke broeikasgassen zijn waterdamp (H_2O), koolstofdioxide (CO_2), methaan (CH_4) en lachgas (N_2O). De concentratie van deze gassen in de atmosfeer is het resultaat van talrijke dynamische processen en cycli die op elkaar ingrijpen.”

In dat milieurapport staat ook dat er steeds meer bewijzen zijn dat de temperatuurstijging die we de laatste vijftig jaar waarnemen, grotendeels toe te schrijven is aan menselijke activiteiten.

“Sinds het begin van het industriële tijdperk (1750) is de concentratie van koolstofdioxide, methaan en lachgas in onze atmosfeer sterk toegenomen (zie Figuur 1.9). De laatste 100 jaar heeft de mens grote hoeveelheden broeikasgassen in de atmosfeer geloosd door verbranding van fossiele brandstoffen (CO_2 en N_2O), veeteelt (CH_4 en N_2O), afvalverwerking (CH_4) en



Figuur 1.9: CO_2 -concentratie bepaald uit ijskernen op Antarctica. Na de rode lijn, die overeenstemt met het jaar 1750, neemt de hoeveelheid CO_2 in de atmosfeer sterk toe. Op de grafiek is te zien dat er de voorbije 2000 jaar eerst een periode was met een vrij stabiele CO_2 -concentratie met een gemiddelde van 280 ppm, gevolgd door een ongezien snelle stijging sinds de industriële revolutie.

chemische processen in de industrie (N_2O). Door de wereldwijde ontbossing en de ermee gepaard gaande verbranding worden grote koolstofreservoirs in het hout en de bodem omgezet naar broeikasgassen (vnl. CO_2). Daarnaast dragen ook nieuwe stoffen zoals de chloorfluorkoolwaterstoffen (CFK's), hun vervangproducten zoals zachte gehydrogeneerde chloorfluorkoolwaterstoffen (HCFK's) en fluorkoolwaterstoffen (HFK's en PFK's), o.a. gebruikt als koelmiddel en drijfgas, en zwavelhexafluoride (SF_6) bij tot het broeikas effect. SF_6 zit in sommige elektrische schakelinstallaties en in geluids-isolerende dubbele beglazing."

"De verhoogde concentratie van broeikasgassen in de atmosfeer versterkt het natuurlijke broeikas effect en leidt bijgevolg tot een verhoging van de gemiddelde temperatuur en een globale klimaatverandering."

"Men verwacht nog deze eeuw:

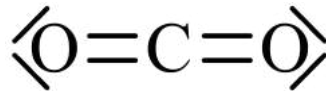
- een verhoging van de gemiddelde temperatuur op wereldschaal met 1,1 tot 6,4 °C;
- een toe- of afname van de neerslaghoeveelheden naargelang de regio;
- een stijging van het zeeniveau met 18 tot 59 cm."

Tijdens de eerste Wereldklimaatconferentie in Genève in 1979 werd de klimaatverandering voor het eerst erkend als een ernstig probleem. Tijdens de Conferentie van Toronto in 1988 riepen wetenschappers, politici en ambtenaren van 48 landen en de Verenigde Naties (VN) op tot een reductie van de CO_2 -emissies met 20 % tegen 2005. In 1988 werd het Intergovernmental Panel on Climate Change (IPCC) opgericht.

Het IPCC verzamelt alle wetenschappelijke publicaties over klimaatonderzoek en analyseert die. Om de zoveel jaar communiceert het over hoe het gesteld is met het wereldklimaat en welke consensus daarover bestaat (Boussemaere, 2015).

CO₂

Bij normale temperatuur en druk is koolstofdioxide een kleurloos en reukloos gas. De molecule heeft een lineaire structuur, die bestaat uit een centraal koolstofatoom waaraan met dubbele covalente bindingen twee zuurstofatomen zijn gebonden.

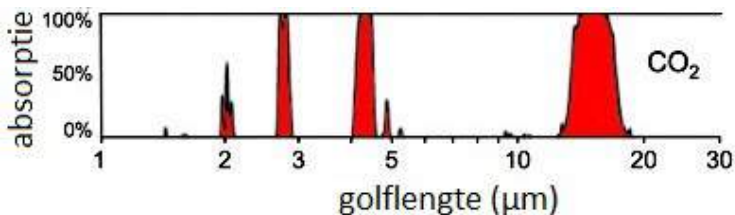


CO₂-molecule (Benjah-bmm27, 2006).

Schematische voorstelling van de structuur (Eugster, 2006).

Licht bestaat uit fotonen, pakketjes van energie. Wanneer een foton met de juiste golflengte botst met een CO₂-molecule, begint de CO₂-molecule te trillen en neemt ze de energie van het foton over. De CO₂-molecule beweegt dan sneller dan ervoor.

Het aardoppervlak absorbeert energie van de zon en het verwarmde aardoppervlak straalt dan infraroodlicht uit. CO₂ is aanwezig in de atmosfeer. Het laat zichtbaar licht, bv. licht uitgestraald door de zon, door en absorbeert infrarood licht als het de juiste golflengte heeft. CO₂ absorbeert infraroodstraling met een golflengte van 2 μm, 2,7 μm, 4,3 μm en 15 μm (overeenkomstige frequentie is dan de resonantiefrequentie). Direct na de absorptie wordt de infraroodstraling in alle richtingen weer geëmitteerd. De CO₂-molecule keert dan terug naar de grondtoestand en een andere CO₂-molecule wordt in trilling gebracht.



Absorptiespectrum (Anderson et al., 2016).

De snelheid van de moleculen in een gas is een maat voor de temperatuur van het gas, dus hoe sneller de moleculen bewegen, hoe hoger de temperatuur in de atmosfeer zal zijn.

(ESRL, 2019; UCAR, 2012)

Op regelmatige basis worden klimaatconferenties gehouden. In december 2018 werd de vierentwintigste klimaatconferentie gehouden in Katowice. Deze klimaatconferenties hebben geleid tot een akkoord om de opwarming ruim onder de 2 °C te houden en te streven naar een opwarming van 1,5 °C (Dienst Klimaatverandering van de Federale overheidsdienst Volksgezondheid, Veiligheid van de Voedselketen en Leefmilieu, 2019).

Hiervoor moet men de CO₂-uitstoot tegen 2050 tot nul herleiden. Met een steeds groeiende wereldbevolking en een continu stijgende vraag naar energie wordt dit een heel grote uitdaging. Ondanks het gestelde doel is ook in 2018 de globale CO₂-uitstoot weer gestegen t.o.v. 2017. Maar zelfs als men erin zou slagen, duurt de opwarming na 2050 nog jaren voort door de voorbije uitstoot.

1.4 Afsmeltende gletsjers en stijging van het zeeniveau

Het IPCC heeft getoond dat het globale zeeniveau tussen 1901 en 2010 met 17 tot 21 cm is gestegen. Jaarlijks steeg het globale zeeniveau in die periode met gemiddeld 1,7 mm. De stijging bedroeg sinds 1971 echter gemiddeld 3,2 mm. De snelheid waarmee het zeeniveau stijgt, neemt dus toe.

Meerdere factoren spelen hierin een rol. Water dat warmer wordt, zet uit. Hierdoor stijgt de zeespiegel. Ook het afsmelten van ijskappen op Groenland en Antarctica en het terugtrekken van gebergtegletsjers overal ter wereld dragen bij aan de stijging (Maslin, 2014).

In de notebooks over de Morteratsch- en de Silvretta-gletsjer gaan leerlingen aan de slag met data verstrekt door onderzoeker Lander Van Tricht. Ze maken grafieken die tonen dat de gletsjers veel korter en veel dunner zijn dan pakweg 100 jaar geleden. Ze berekenen ook de totale terugtrekking en de totale verdunning.



Bekijk zeker de notebooks over de afsmeltende gletsjers en over het stijgend zeeniveau in Oostende.

Het zeeniveau in Oostende wordt sinds 1951 opgemeten. De data zijn terug te vinden op de website van MIRA (Vlaamse Milieumaatschappij, 2019). In de notebook 'Zeeniveau' gebruiken de leerlingen deze data en ontdekken ze dat het zeeniveau in Oostende sinds het begin van de metingen al met 13 cm is gestegen.

Gebergtegletsjers en ijskappen

Door de opwarming van de aarde zijn de gletsjers wereldwijd aan het terugtrekken. Het smeltwater afkomstig van gebergtegletsjers levert momenteel een belangrijke bijdrage tot de stijging van de zeespiegel. In de toekomst zal dit aandeel kleiner worden omdat er steeds minder gebergtegletsjers zullen zijn en ze in volume afnemen. De ijskappen op Groenland en Antarctica zullen daarentegen steeds meer bijdragen tot een stijging van de zeespiegel. Mogelijke scenario's spreken van een stijging tot 30 cm door het terugtrekken van de gebergtegletsjers.

De snelheid waarmee het volume van een gebergtegletsjer afneemt o.i.v. de globale temperatuurstijging, verschilt van gletsjer tot gletsjer. Lokale factoren spelen hierin een rol, zoals de oriëntatie, het volume en de hellingsgraad van de gletsjer, het aandeel van de gletsjer dat in de schaduw van de omringende bergen ligt, het klimaat waarin de gletsjer zich bevindt ...

Gebergtegletsjers zijn van heel groot belang.

Bijvoorbeeld:

- Ze zijn een bron van drinkwater.
- In droge gebieden, bv. in Centraal-Azië of Zuid-Amerika, komt in de zomer maanden al het water voor de landbouw, de industrie en het huishouden van het smeltwater.
- In sommige streken, bv. in de Alpen of Noorwegen, is het smeltwater belangrijk voor waterkrachtcentrales en voor irrigatie.
- De gletsjers zijn een toeristische attractie (in de winter om te skiën, in de zomer om te wandelen). Sommige gletsjers hebben de status van cultureel erfgoed.
- Het smeltwater voedt rivieren.

- Het koude smeltwater, maar eveneens de gletsjers zelf, zijn een belangrijke habitat voor dier- en plantensoorten, bv. bepaalde insecten, zalm en forel.

Het afsmelten van de gebergtegletsjers kan ook een gevaar vormen: het kan leiden tot lawines en het smeltwater kan overstromingen veroorzaken van meren die bestaan uit gletsjerwater.

(Lander Van Tricht, persoonlijke communicatie)

Globaal zeeniveau

Het zeeniveau wordt beïnvloed door o.a. de stijging van het volume van de watermassa bij een stijgende temperatuur, afsmeltende ijskappen en gletsjers, veranderende opslag van oppervlaktewater en grondwater. De globale klimaatverandering leidt nog deze eeuw tot een stijging van het zeeniveau met 18 tot 59 cm (Vlaamse Milieumaatschappij, 2019).

De hoogte van een punt wordt gemeten ten opzichte van de zeespiegel. De zeespiegel staat echter niet altijd op dezelfde hoogte. Eb en vloed zorgen voor een verschil dat aan de Belgische kust ongeveer vier meter bedraagt.

Men heeft dus een referentiepunt nodig. Die referentiepunten zijn lokaal gekozen. Om nationale hoogtemetingen met elkaar te vergelijken, moet men rekening houden met de verschillende referentiepunten en moet men het lokale zeeniveau uitdrukken in mm RLR (Revised Local Reference); de data t.o.v. de lokale referentie worden dus omgezet naar data t.o.v. het internationaal referentieniveau (De Boosere, 2010).

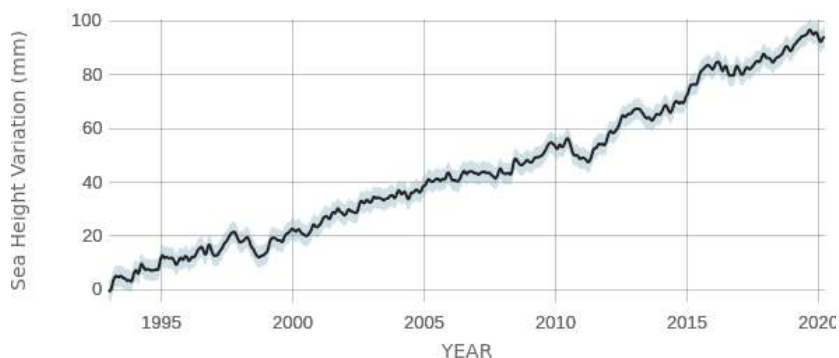
Het zeeniveau is niet altijd en overal even hoog en door de opwarming van de aarde stijgt het gemiddeld zeeniveau. Dat betekent dat het op sommige plaatsen meer stijgt dan op andere en dat het hier en daar zelfs kan dalen. In Groenland bijvoorbeeld daalt het zeeniveau t.o.v. het land zelf, omdat het smelten van de Groenland-ijskap ervoor zorgt dat de landmassa 'opveert' en dus hoger komt te liggen. De zeespiegel vertoont ook schommelingen in de tijd. Mensen die in een kustgebied wonen of op een eiland kunnen dus in de problemen komen als de lokale stijging van het zeeniveau te groot is (Kulp & Strauss, 2019).

De oceanen temperen ook de opwarming van de atmosfeer, door het grootste deel van de extra warmte te absorberen. Maar water dat warmer wordt, zet uit. Hierdoor stijgt de zeespiegel. Dit is de thermale expansie. Het afsmelten van ijskappen op Groenland en Antarctica en het terugtrekken van de gebergtegletsjers overal ter wereld dragen bij aan de stijging.

Het smelten van het zee-ijs, bv. op de Noordpool, draagt niet bij tot de zeespiegelstijging omdat dit ijs al in zee gelegen is.

Sinds 1900 is het globale zeeniveau gemiddeld 16 tot 21 cm gestegen. Het zeeniveau in Oostende volgt quasi de gemiddelde globale trend.

Het zeeniveau wordt op verschillende manieren gemeten. De NASA doet de metingen sinds het begin van de jaren 90 met satellieten en vond dat de zeespiegel sinds 1993 met gemiddeld 3,3 mm per jaar stijgt. Dat is meer dan dubbel zo snel als in het begin van de 20ste eeuw (NASA, 2020a).



Source: climate.nasa.gov

Het zeeniveau geobserveerd door satellieten (NASA, 2020b).

Het zeeniveau is sneller beginnen te stijgen in het begin van de jaren 90, op hetzelfde moment als de gebergtegletsjers en de ijskappen sneller zijn gaan terugtrekken. Lange tijd was het grootste deel van de zeespiegelstijging te wijten aan de thermale expansie. Maar in de periode 2006-2015 hebben het afsmelten van de ijskappen en het terugtrekken van de gebergtegletsjers een nog grotere impact op de stijging van het zeeniveau dan de thermale expansie. Naar de toekomst toe zal het aandeel van de ijskappen hierin steeds groter worden (IPCC, 2019; Zemp et al., 2019).

1.5 Consensus

Het IPCC stelt dat de klimaatverandering een feit is en dat er een heel grote consensus is dat de opwarming veroorzaakt is door de uitstoot aan broeikasgassen door de mens (IPCC, 2018).

De klimaatmodellen voorspellen niet 100 % hetzelfde, maar de meeste zijn wel gelijklopend. De klimaatopwarming hangt af van parameters zoals de gemiddelde globale temperatuur, de grootte van het effect van de atmosferische CO₂-concentratie, het aandeel dat waterdamp heeft in de opwarming en de sterkte van de *radiative forcing*. Bovendien zijn er meerdere manieren om deze parameters te berekenen. Daardoor zijn er tussen de onderzoeksgroepen kleine verschillen in de schattingen ervan. Om die reden vindt men in het IPCC-rapport intervallen terug voor de berekende waarden en geen exacte getallen.

Naar het mogelijk effect van het opwarmen van de permafrost en het afsmelten van ijs wordt volop onderzoek gedaan. Ook de vraag in welke mate de opwarming leidt tot extreme weersomstandigheden houdt vele klimaatdeskundigen bezig.

Het modelleren van de toekomstige klimaatverandering is niet eenvoudig. Er moeten immers veel factoren in rekening gebracht worden, zoals bevolking, gebruik van energie, uitstoot van broeikasgassen, maar ook gebruik van land en aanwezige vegetatie. Vegetatie speelt zelfs een belangrijke rol, aangezien planten de CO₂-cycli van de wereld beïnvloeden (zie Hoofdstuk 2).

Radiative forcing is het verschil tussen de energie die wordt geabsorbeerd door de atmosfeer en de aarde, en de energie die teruggestraald wordt naar de ruimte. Het is een maat voor de sterkte van het broeikaseffect.

Antropoceen

In de nieuwsbrief van mei 2000 van het *International Geosphere-Biosphere Programme* (IGBP) lanceren de Nederlandse Nobelprijswinnaar Paul Crutzen en de Amerikaan Stoermer de term 'antropoceen' als

kandidaat voor een nieuw geologisch tijdvak. Volgens hen is de aarde zo sterk veranderd door menselijke activiteiten dat het tijdvak Holoceen dat 11 700 jaar geleden begon, aan een opvolger toe is. Omdat de mens globaal het land, de oceanen, de atmosfeer en het leven op aarde veranderd heeft, stellen ze dat het daarom “meer dan gepast is om met de nieuwe term de centrale rol van de mensheid in geologie en ecologie te benadrukken” (Crutzen & Stoermer, 2000).

Het concept van het antropoceen werd reeds naar voor geschoven in de negentiende eeuw door de geoloog en theoloog Jenkin uit Wales die sprak over *the human epoch* en door de Italiaanse geoloog en priester Stopponi die het had over het *Anthropozoic*. Later werden nog andere termen geponeerd: *Psychozoic*, *Anthropogene*, *Noösphere*, *Atomic age*, *Technogene* of *Quinary*, en *Anthrocene*. Dit gebeurde door geologen, een theoloog, een filosoof en chemicus, en journalisten (Rull, 2017).

Hoewel het antropoceen geen officieel tijdvak is, denken velen van wel. De term is ondertussen al zo ingeburgerd dat ze al veelvuldig gebruikt wordt in domeinen zoals filosofie, sociologie, communicatie, politiek en recht (Rull, 2017).

Sinds 2009 ijvert Crutzen met de *Anthropocene Working Group* (AWG) bij de *International Commission on Stratigraphy* (ICS) om van het antropoceen een officieel tijdvak te maken.

Nu kan men niet zomaar een nieuw geologisch tijdvak invoeren. De ICS beslist daarover en baseert zich daarvoor op vastgelegde wetenschappelijke criteria. Om een tijdvak te definiëren moeten veranderingen op wereldschaal vastgelegd zijn in stratigrafisch materiaal. Bovendien moeten die veranderingen synchroon hebben plaatsgevonden. Het begin van een tijdvak moet gebonden zijn aan een *Global Stratotype Section and Point* (GSSP) met nog bijkomende stratigrafische markers die de veranderingen aantonen. Een GSSP is een lokaal bewijs van een globaal fenomeen. Het is bv. een duidelijk te onderscheiden kenmerk in een rotslaag of een ijskern.

Voorbeelden (Lewis & Maslin, 2015):

- De grens tussen het Krijt en het Paleogeen: GSSP is de piek in iridium,

een overblijfsel van een meteorietinslag, aangetroffen in een rots in Tunesië, gedateerd 66 miljoen jaar geleden. De inslagkrater in Mexico en een wereldwijde sedimentlaag met materiaal dat bij de inslag is vrijgekomen, vormen bijkomend bewijs (Vellekoop, 2015). Het gevolg van de meteorietinslag was het uitsterven van de niet-vliegende dinosaurussen en het drastisch toenemen van de zoogdieren.

- De grens tussen Ordovician en Silurian: GSSP is de aanwezigheid van een bijzonder plankton graptoliet, *Akidograptus ascensus*, gedateerd 443,8 miljoen jaar geleden.
- Het begin van het Holoceen: GSSP is de abrupte verandering in deuteriumgehalte (^2H , een stabiele waterstofisotoop) in diep ijs in Groenland, gedateerd 11 700 jaar geleden. Dit GSSP wijst op het begin van een klimaatopwarming vooral in het noordelijke halfrond. Vijf andere stratotypes ondersteunen dit GSSP: vier meren en een marien sediment.

Als men dus een antropoceen-tijdvak wil vastleggen, dan moet men een globaal en synchroon fenomeen kiezen dat het begin van het antropoceen inluit en heeft men een GSSP nodig dat hiervoor het bewijs vormt. Het GSSP moet daarenboven nog gepaard gaan met andere stratigrafische bewijzen. Bovendien zouden die bewijzen ook nog ver in de toekomst moeten terug te vinden zijn. Er zijn heel wat bewijzen van de geologische en ecologische veranderingen veroorzaakt door de mens (Lewis & Maslin, 2015).

In de twintigste eeuw zette men atmosferische stikstof om in ammoniak om als meststof te gebruiken. Hierdoor werd de atmosferische stikstofcyclus fundamenteel gewijzigd. Sinds de industriële revolutie in de 18de eeuw is de atmosferische CO_2 -concentratie drastisch toegenomen. De aanzienlijke inbezitting van land voor voedsel, brandstof, vezels, veevoeder, jacht en oogst zorgt ervoor dat er minder plaats is voor dieren en planten, waardoor vele soorten reeds uitstierven. Door transport werden organismen, dieren en gewassen gebracht waar ze nooit voordien waren, waardoor het leven op aarde meer homogeen werd en er zelfs nieuwe soorten ontstonden. Bovendien is het aannemelijk dat al deze menselijke activiteit druk zet op de evolutie, mede door de ontwikkeling van antibiotica en pesticiden.

Wanneer zou het begin van het antropoceen zich dan situeren? Lewis & Maslin bespreken verschillende mogelijkheden (2015): Is dat bij het ontstaan van

de mensheid? Bij de ontwikkeling van de landbouw of bij de ontdekking van Amerika? Kiest men voor het begin van de industriële revolutie of voor midden twintigste eeuw, toen de *Great Acceleration* zijn aanvang nam?

- Het ontstaan van de mensheid.
 - In het geologisch archief vindt men bewijs voor het gebruik van vuur in fossiele houtskool gedateerd in het vroeg Pleistoceen. Deze fossielen zijn lokaal en gebruik van vuur komt dus niet in aanmerking als GSSP.
 - Het uitsterven van de megafauna, 50 000 tot 10 000 jaar geleden, waarbij de helft van de grote zoogdieren wereldwijd uitstierf, wat overeenkomt met 4 % van de zoogdieren. Het was echter continentgebonden, gebeurde op verschillende tijdstippen en komt dus niet in aanmerking als GSSP.
- De ontwikkeling van de landbouw tastte de biodiversiteit aan omdat de oorspronkelijke begroeiing werd vervangen door gewassen. Het had ook een grote impact op de mensen: nomaden vestigden zich; deze nederzettingen werden uiteindelijk steden.
 - Bewijzen hiervoor in het geologisch archief zijn pollen van gecultiveerde planten of gereedschappen, maar deze bewijzen zijn lokaal en zijn globaal niet synchroon.
 - Overblijvend cultureel bewijs, zoals keramiek, bewaard in stratigrafische archieven, komt ook niet in aanmerking omdat het niet globaal voorkomt.
 - Bodems gevormd door intensieve landbouw komen ook niet in aanmerking omdat ze diachroon zijn.
- De ontdekking van Amerika: de aankomst van de Europeanen in de Caraïben in 1492 en het annexeren van Amerika hebben geleid tot de grootste volksverhuizing in de voorbije 13 000 jaar en de eerste wereldwijde handelsnetwerken. Het resulteerde in het mengen van voorheen gescheiden soorten leven: gewassen, dieren . . . Het aantal mensen daalde drastisch in Amerika, van meer dan 50 miljoen in 1492 naar 6 miljoen in 1650. Dat was te wijten aan ziektes meegebracht door de Europeanen, oorlog, slavernij en hongersnood.
 - Van het mengen van de soorten zijn bewijzen te vinden in pollen bewaard in mariene of meersedimenten.

- Omdat er minder mensen waren, was de landbouw zo goed als opgehouden, er werd minder vuur gebruikt, meer dan 50 miljoen hectaren bos, savanne en grasland konden zich herstellen, wat gepaard ging met een grote opname van koolstof door vegetatie en bodem. Het zou kunnen dat deze koolstofvastlegging heeft bijgedragen tot de waargenomen vermindering van atmosferische CO₂ in de kleine ijstijd, gemeten in Antarctisch ijs. Maar er was ook nog de uitbarsting van de Huaynaputina in Peru in 1600 die hieraan waarschijnlijk heeft bijgedragen en vermoedelijk nog andere factoren (persoonlijke correspondentie met Pieter Tans, NOAA).
- De industriële revolutie.
 - Globale CO₂-concentratie kan niet gebruikt worden als GSSP, omdat het gebruik van kolen zich initieel traag heeft verspreid en men in het ijs slechts een geleidelijke toename van CO₂ kan terugvinden.
 - In ijskernen en sedimenten is vlieggas terug te vinden, afkomstig van het verbranden van fossiele brandstoffen (Rull, 2017).
 - Andere markers zoals lood, methaan, nitraat, overblijfselen van het gebruik van fossiele brandstoffen . . . zouden een GSSP kunnen ondersteunen, maar er ontbreekt een globale GSSP.
- Het midden van de twintigste eeuw: de *Great Acceleration* wordt gekenmerkt door een zeer grote toename van de wereldbevolking, grote veranderingen in natuurlijke processen en de ontwikkeling van nieuwe materialen zoals mineralen en plastic.
 - De radioactieve neerslag van kernbommen kan de rol spelen van globale marker. Een duidelijke piek aan radioactiviteit van de vroege jaren 50 tot de late jaren 60 is o.a. vastgesteld in ijs, meersediment, korallen en boomringen. Een duidelijke piek is er ook in de hoeveelheid atmosferische ¹⁴C, gemeten in gletsjerijs en boomringen, gedateerd 1963-1965. Deze piek in ¹⁴C gemeten in een dennenboom in Niepolomice in Polen komt in aanmerking als GSSP. Ondersteunende markers kunnen bv. plutonium-isotopen in sedimenten zijn die nucleaire testen aantonen of jodium-isotopen in mariene sedimenten.
 - Andere markers die de invloed van de mens aantonen, kunnen zijn: fossiele pollen van genetisch gemanipuleerde gewassen, lood-isotopen in ijsmassa, microplastics in mariene sedimenten.

- Het nadeel van dit GSSP zou zijn dat de nucleaire neerslag geen wereldveranderend event was en dat de datum zeer recent is.
- Mensen verplaatsen gesteenten en mineralen op grote schaal. Natuurlijke edelstenen, zilver en goud zijn te vinden over de hele wereld en overal zijn er wel verzamelaars van mineralen, waardoor mineralen op plaatsen aanwezig zijn waar ze dat normaal gezien niet zouden zijn. Van de meer dan 5000 gekende mineralen zijn er 208 pas ontstaan sinds de industriële revolutie door menselijke activiteit, voornamelijk in de mijnbouw. In de toekomst zal men in de sedimentlagen bouwmaterialen vinden zoals cement en beton, metaallegeringen zoals staal en aluminium, en synthetische edelstenen (Hazen et al., 2017).

De ICS houdt (voorlopig) vast aan het Holoceen. In de zomer van 2018 deelden ze het tijdvak nog op in drie etages: het Greenlandiaan (van 11 700 tot 8300 jaar geleden), het Northgrippiaan (van 8300 tot 4200 jaar geleden) en het Meghalayaan (dat 4200 jaar geleden begon). In het noordoosten van India vond men in een stalagmiet duidelijke tekenen van een tweehonderd jaar lange periode van droogte. Geologen vonden bewijzen van dit gebeuren in sedimenten op alle werelddelen. Door deze droogte ontstonden grote migratiestromen en gingen beschavingen ten onder (Durham University, 2018).

De voorstanders van het antropoceen voelen zich gepasseerd. De tegenstanders van het antropoceen begrijpen de beslissing van de ICS.

Die tegenstanders menen ofwel dat er geen nood is aan een nieuw tijdperk, omdat de invloed van de mens toch ontegensprekelijk bewezen is. Anderen vinden dat het eerder om een historisch tijdperk gaat, en niet om een geologisch tijdvak. Sommigen stellen zich de vraag of de drijfveren van de voorstanders i.p.v. wetenschappelijk eerder politiek zijn. Of ze vinden de term te politiek geladen.

Een officiële invoering van de term antropoceen zou een erkenning inhouden van de verstrekkende gevolgen van menselijke acties op de aarde (Lewis & Maslin, 2015). Sloterdijk schrijft het als volgt (2018): “De proliferatie van dit begrip is waarschijnlijk vooral te herleiden tot het feit dat het onder het mom van wetenschappelijke neutraliteit een boodschap overbrengt van bijna niet te overtreffen moreel-politieke urgentie, een boodschap die in expliciete taal luidt: sinds de aanwezigheid van de mens op aarde niet langer de vorm aanneemt van een meer of minder spoorloze integratie, is hij voor de bewoning en het beheer van

de aarde als geheel verantwoordelijk geworden. ... Met het toeschrijven van verantwoordelijkheid kunnen mogelijke aanklachten geadresseerd worden. ... Als we 'antropoceen' zeggen, zitten we schijnbaar in een geo-wetenschappelijk seminar. In werkelijkheid nemen we deel aan een rechtszitting".

Anderzijds zou een officiële verwerping door de ICS kunnen geïnterpreteerd worden als een goedpraten van het ongecontroleerd omgaan met natuurlijke grondstoffen (Rull, 2017).

Rull (2017) oppert dat het invoeren van een antropoceen-tijdvak ook wel zou kunnen betekenen dat het het laatste tijdvak zou zijn. Als er geen grote ramp zoals een gigantische vulkaanuitbarsting, meteorietinslag, een ijstijd of een nucleaire oorlog plaatsvindt, dan zou de mens een even grote impact kunnen blijven uitoefenen op de aarde en dan blijft het antropoceen duren tot de mens uitsterft.

Samengevat

Dankzij de paleoklimatologie weet men dat de atmosferische CO₂-concentratie het klimaat beïnvloedt. Er is bv. een samenhang tussen het CO₂-gehalte in de atmosfeer en de heersende temperatuur. Tijdens de voorbije 2000 jaar was er eerst een periode met een vrij stabiele CO₂-concentratie met een gemiddelde van 280 ppm, gevolgd door een ongezien snelle stijging sinds de industriële revolutie. Deze stijging kan enkel verklaard worden door de uitstoot van broeikasgassen door de mens. Het heeft bovendien geleid tot een globale klimaatopwarming, iets dat zich de voorbije 2000 jaar nooit eerder heeft voorgedaan.

Planten spelen een grote rol in de CO₂-cycli van de wereld. Om goede klimaatmodellen te kunnen maken, is daarom veel kennis nodig over hoe planten zich aanpassen aan veranderende CO₂-concentraties.

Leestip

Pieter Boussemaere. Eerste hulp bij klimaatverwarring. Waarom de opwarming van de aarde veel meer is dan een milieuprobleem (Boussemaere, 2015).

Charles David Keeling. Rewards and Penalties of Monitoring the Earth (Keeling, 1998).

Mark Maslin. Climate Change: A Very Short Introduction (Maslin, 2014).

Valerie Trouet. Wat bomen ons vertellen. Een geschiedenis van de wereld geschreven in jaarringen (Trouet, 2020).

Kijktip

Keeling's Curve. The Story of CO₂.
<https://youtu.be/0Z8g-smE2sk> (American Museum of Natural History, 2014).

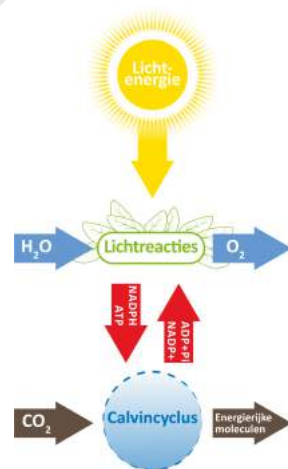
Carbon Dioxide Pumhandle 2019.
<https://www.esrl.noaa.gov/gmd/ccgg/trends/history.html> (CarbonTracker, 2019).

2.1 Gasuitwisseling tussen een plant en haar omgeving

Zoals alle organismen hebben planten water, mineralen en energierijke koolstofverbindingen nodig om te leven, te groeien en zich te kunnen reproduceren. Planten zijn echter in staat om zelf hun koolstofverbindingen op te bouwen, door met lichtenergie en water koolstofdioxide om te zetten in koolhydraten (glucose). Dit proces heet fotosynthese (zie Figuur 2.1).

De opname van CO_2 gebeurt vooral langs de bladeren. Om uitdroging tegen te gaan en om de plant te beschermen tegen ziektes, zit er op de buitenkant van het blad een waslaag, de cuticula. Die waslaag verhindert echter ook dat er CO_2 in het blad komt. Om de opname van CO_2 toch mogelijk te maken, zijn er microscopisch kleine poortjes in de opperhuid van het blad, de huidmondjes (stomata). Als deze kleine poriën in de opperhuid van het blad open zijn, kan er via die weg CO_2 in het blad opgenomen worden.

Tegelijk kan er ook water uit de plant ontsnappen langs de stomata, men zegt dat de plant transpireert (transpiratie). De huidmondjes staan dus in voor afkoeling van de plant en het onderhouden van de sapstroom

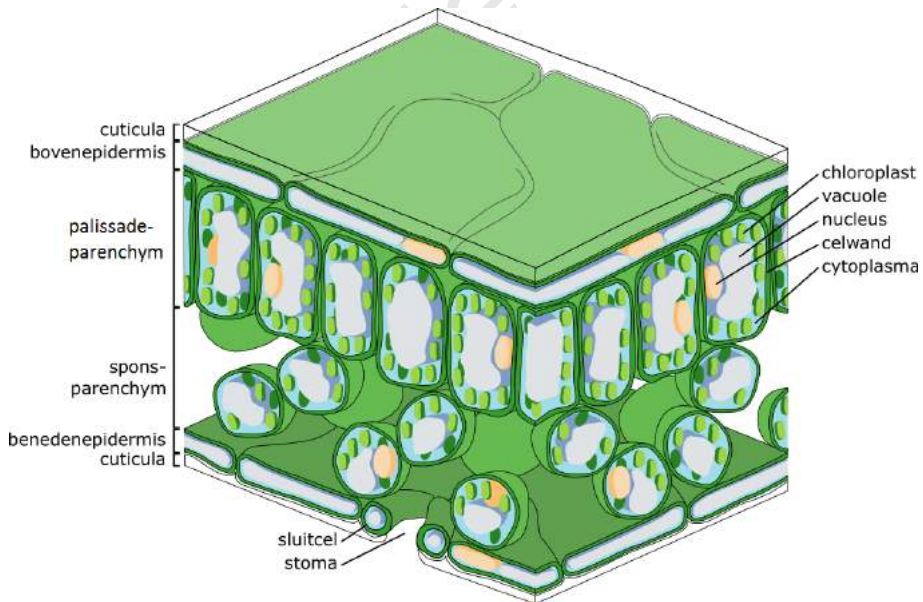


Figuur 2.1: Fotosynthese.

van wortel tot hoog in de plant. Als het warmer wordt of als de plant genoeg CO₂ heeft opgenomen, zal de plant de huidmondjes sluiten om te voorkomen dat ze te veel water verliest en uitdroogt.

Een plant ademt ook, men spreekt van respiratie: 's nachts zet de plant glucose met zuurstofgas om in koolstofdioxide en waterdamp. Hiervoor neemt de plant O₂ op uit de atmosfeer en geeft ze CO₂ vrij. Dit gebeurt ook overdag, maar door de fotosynthese wordt dan meer zuurstof aangemaakt dan ingeademd, waardoor ze bij de fotosynthese O₂ teruggeeft aan de atmosfeer.

De stomatale geleidbaarheid is de mate waarin de gasuitwisseling tussen een plant en haar omgeving gebeurt. De uitwisseling van waterdamp met de omgeving (transpiratie) gebeurt 1,6 keer zo snel als de opname van CO₂ langs de stomata (Hetherington & Woodward, 2003).



Figuur 2.2: Bladstructuur (Zephyris, 2011).

2.2 Stomata

Stomata of huidmondjes zijn dus kleine poriën op bladeren en stengels, die het uitwisselen van gassen (waterdamp, CO₂ en O₂) tussen de plant en haar omgeving mogelijk maken. Een huidmondje ligt tussen twee sluitcellen (zoals te zien op de bladstructuur in Figuur 2.2). Als de sluitcellen in volume toenemen, gaat het huidmondje open. De sluitcellen nemen in volume toe door het opnemen van water. Het gevolg is dat de sluitcellen krommen, waardoor het huidmondje opent (zie Figuren 2.3, 2.4).

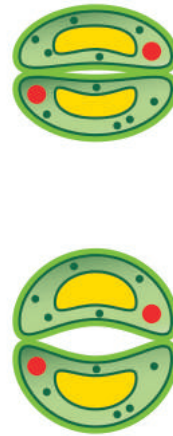
Stomata kunnen op beide zijden van een blad voorkomen (amphistomateus), of op slechts één zijde, meestal de onderkant (hypostomateus), uitzonderlijk de bovenkant (epistomateus), bv. bij een waterlelie.

De stomata en de cuticula worden gezien als sleutel-elementen in de evolutie van planten. Ze zorgen ervoor dat planten in verschillende en wisselende omstandigheden kunnen leven zonder uit te drogen (Hetherington & Woodward, 2003).

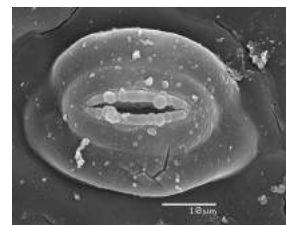
Stomata geven informatie over de plaats van de plant in de stamboom van de bloemplanten. Onder de bedektzadigen of bloeiende planten onderscheidt men de dicotylen en de monocotylen (zie Figuur 2.6).

De dicotylen hebben per zaad twee zaadlobben. Die zaadlobben (of cotylen) bevatten het reservevoedsel nodig voor de kieming en eerste groei. Bij de monocotylen is er slechts één zaadlob per zaad. Prei, narcis en gras zijn monocotylen; boterbloem, madeliefje en notelaar zijn dicotylen. De nerven in de bladeren van monocotylen lopen doorgaans parallel, bij de dicotylen zijn de nerven vertakt (Figuur 2.5).

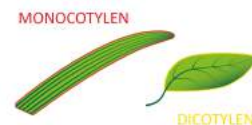
Naast een verschillende ligging van de vaatbundels en een verschillend aantal bloemdelen, is er ook een verschil bij de huidmondjes. Huidmondjes van monocoty-



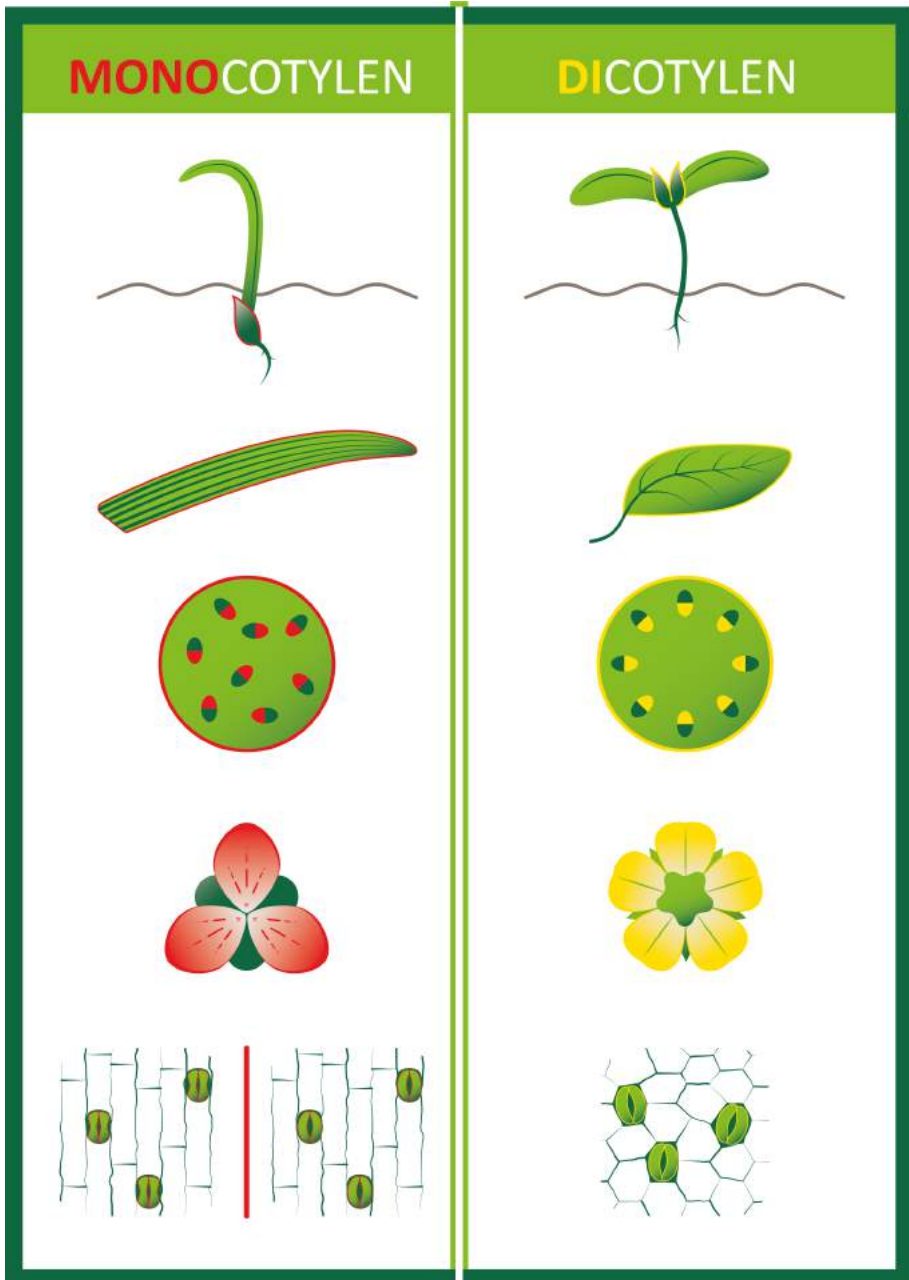
Figuur 2.3: Gesloten en open huidmondje.



Figuur 2.4: Microfoto huidmondje.



Figuur 2.5: Parallele nerven bij monocotylen en vertakte nerven bij dicotylen.



Figuur 2.6: Monocotylen en dicotylen

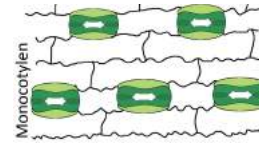
len bijvoorbeeld liggen georganiseerd in rijen, in tegenstelling tot huidmondjes van dicotylen, en zijn over het algemeen groter dan deze van dicotylen (Figuren 2.7, 2.8). Onder de stomata onderscheidt men twee types: haltervorm en niervorm. De haltervorm is typisch voor het monocotyle gras en de niervorm voor andere soorten planten.

Afhankelijk van de soort plant en de omgevingsfactoren variëren de grootte van een stoma en het aantal stomata per oppervlakte-eenheid (de dichtheid). Een stoma is tussen de 10 en 100 micrometer lang (vergelijkbaar met de dikte van een mensenhaar) en de dichtheid gaat van 5 tot 1000 stomata per mm^2 bladoppervlakte.

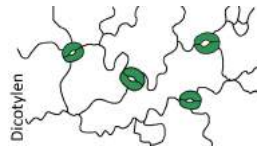
Stomata bekijken gebeurt dus met een microscoop (Figuur 2.4). In Figuren 2.9 en 2.10 zie je een deel van een blad van een monocotyle en van een dicotyle plant gefotografeerd door een microscoop.

Er is voor meerdere plantenfamilies en fossiele bladeren een duidelijk verband vastgesteld tussen het aantal stomata en de grootte van de stomata. Op de grafiek in Figuur 2.11 is te zien dat hoe meer stomata er zijn per oppervlakte-eenheid, hoe korter de sluitcellen en dus ook hoe kleiner de stomata zijn (Hetherington & Woodward, 2003).

Stomata geven zo informatie over de omgeving waar de plant groeide. Omdat de vorming van stomata afhankelijk is van omgevingsfactoren, kan men de grootte en de dichtheid van stomata ook gebruiken als een tool om het klimaat te bestuderen. De grootte en de dichtheid van de stomata kan verschillende aspecten onthullen van het milieu waarin een plant leefde: temperatuur, atmosfeer, CO_2 -concentratie. Dit wordt uitvoerig gebruikt in een paleontologische context om het klimaat waarin de planten groeiden te reconstrueren.

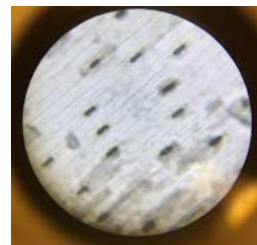


Figuur 2.7: Ligging huidmondjes bij monocotylen. De huidmondjes zijn haltervormig. Geïnspireerd op Hepworth et al. (2018).

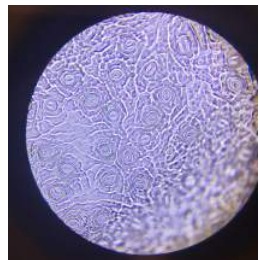


Figuur 2.8: Ligging huidmondjes bij dicotylen. De huidmondjes zijn niervormig. Geïnspireerd op Hepworth et al. (2018).

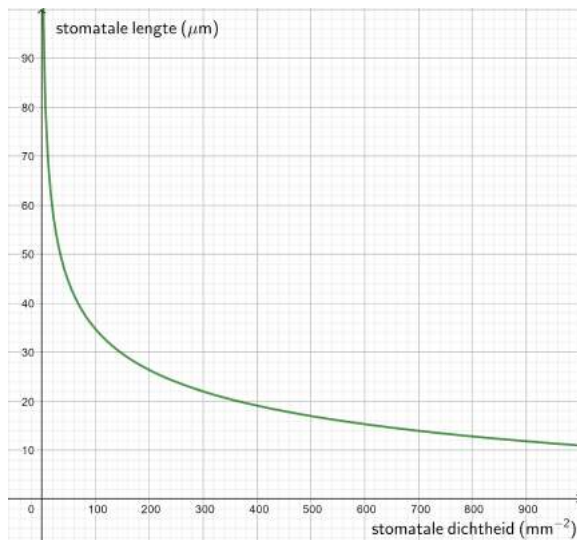
Een mensenhaar is tussen de 17 en $180 \mu\text{m}$ dik.



Figuur 2.9: Microfoto monocotyl.



Figuur 2.10: Microfoto dicotyl.



Figuur 2.11: Het verband tussen stomatale lengte (in μm) en stomatale dichtheid (in mm^{-2}) wordt gegeven door: $y = -28,75 + 162 x^{-0,2036}$ (Hetherington & Woodward, 2003).

2.3 Openen en sluiten van stomata

Een plant zoekt voortdurend naar een evenwicht tussen vochtverlies en CO_2 -opname, afhankelijk van de vochttoestand van de plant, luchtvochtigheid, licht en CO_2 -concentratie. Al deze factoren samen met de biologische klok van de plant bepalen of en in welke mate de huidmondjes openstaan. De sluitcellen reageren op externe en interne signalen (bv. het plantenhormoon abscisinezuur (ABA)) om de huidmondjes te sluiten.

Stomata zijn onderhevig aan ritmes zoals dag en nacht en de seizoenen. Omdat er licht nodig is voor de fotosynthese, zullen de stomata doorgaans overdag openen en 's nachts sluiten. Dit proces wordt geregeld door hun biologische klok. Bijvoorbeeld, als men in een serre kunstmatig dag en nacht omdraait, houden de planten nog een tijd hun eigen ritme aan, maar na een tijd passen ze zich wel aan.

Bij de *Cistus incanus* (een plant uit de mediterrane omgeving, zie Figuur 2.12) is er een duidelijk verschil



Figuur 2.12: *Cistus incanus* (Mansfeld, 2011).

tussen zomer en winter: de bladeren geproduceerd in een koude en natte winter zijn groot en hebben veel stomata onderaan het blad; bladeren geproduceerd in een warme en droge zomer zijn kleiner en deels opgepold en vormen een holte waarin de stomata voorkomen (Hetherington & Woodward, 2003).

Eikenbladeren gevormd bij zomertemperaturen hebben een lagere stomatale dichtheid dan bladeren die in de lente bij een lagere temperatuur zijn gevormd (Beerling & Chaloner, 1993).

Duidelijk is dat de stomata 'reageren' op omgevingsfactoren. Bij eenzelfde plant ziet men soms het verschil tussen bladeren onderaan de plant, die veel beschaduwd zijn, en bladeren in de kruin die veel meer blootgesteld zijn aan het licht. Zelfs op eenzelfde blad kunnen stomata zich anders gedragen omdat dit de plant ten goede kan komen.

Het ene huidmondje kan open zijn en het ander dicht, zoals te zien is op de foto in Figuur 2.13, bv. bij planten in de onderetage van een bos. Het licht kan zeer lokaal door de boomkruinen tot de ondergroei doordringen. Het openen van de stomata in het verlichte deel van het blad, terwijl de stomata in het onverlichte deel van het blad gesloten blijven, laat toe dat de plant het beschikbare licht efficiënt gebruikt zonder onnodig verlies van water en energie (Hetherington & Woodward, 2003).

Een plant moet ervoor zorgen dat er een continue stroom water is van de bodem naar het blad. Hiervoor is het belangrijk dat ze zich snel kan aanpassen aan wisselende omstandigheden, bv. door het snel kunnen openen en sluiten van de stomata. De grootte van de stomata speelt hierin een sleutelrol (Hetherington & Woodward, 2003).

Een studie bij zes bosbomen toont aan dat grotere stomata trager sluiten en dus een groter risico lopen op



In de notebooks over het Amazonewoud wordt de invloed van de omgeving op de huidmondjes bestudeerd.



Figuur 2.13: Open en gesloten stoma op zelfde blad.

het niet in stand te kunnen houden van de sapstroom bij droogte. Varens die zeer beschaduwde zijn, hebben weinig en grote niervormige stomata. In koele, vochtige omstandigheden houden ze vaak hun stomata open, tenminste tijdens de vroege uurtjes van de dag. Er is dan nochtans vaak geen licht, maar omdat ze traag openen in de korte periodes dat er wel licht is, is dit toch beter voor de fotosynthese. Het lijkt er dus op dat grote niervormige stomata vooral een belangrijk kenmerk zijn van planten in een vochtige en zeer beschaduwde omgeving. In droge condities kan hun trage dynamiek leiden tot problemen.

Kleine stomata kunnen sneller openen en sluiten. Aangezien ze samengaan met een grote dichtheid, hebben ze de capaciteit om de stomatale geleidbaarheid snel op te voeren als de condities voor fotosynthese optimaal zijn, en om zo de opname van CO_2 in het blad te maximaliseren.

Bij haltervormige stomata leiden kleine veranderingen in de breedte reeds tot grote veranderingen in de opening, zonder veel energie te vergen. Daarom lijken ze beter geschikt dan de niervormige stomata om in te spelen op wisselende condities (Hetherington & Woodward, 2003).

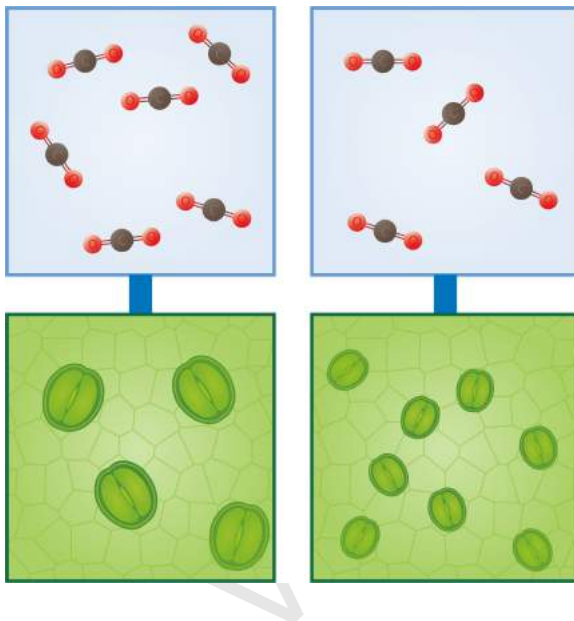
2.4 Evolutie van planten en hun stomata

Fossiele bladeren geven veel informatie over de evolutie van planten en hun stomata. Vierhonderd miljoen jaar geleden kwamen stomata bij planten voor het eerst voor. Sindsdien zijn ze geëvolueerd: er zijn opvallende veranderingen opgetreden waaronder verschillen in de grootte en de dichtheid van de stomata (Hetherington & Woodward, 2003).

Vierhonderd miljoen jaar geleden was de atmosferische CO_2 -concentratie veel hoger dan nu. Er kwamen echter herhaaldelijk periodes voor met een dalende of een lage CO_2 -concentratie (Franks & Beerling, 2009).

Bij een lage CO_2 -concentratie zijn planten met veel stomata in het voordeel en komen ze dan ook veel voor. Wanneer er veel CO_2 in de atmosfeer zit, kunnen planten met minder stomata overleven (zie kader 'Voorbeelden van het verband tussen klimaat en stomata'). Planten met minder stomata zijn dan zelfs in het voordeel en zullen veel voorkomen (Thanukos, 2018). Dit wordt geïllustreerd in Figuur 2.14.

Ze zijn in het voordeel omdat het hebben van veel stomata gepaard gaat met het verlies van water en een grote energiekost bij het aanmaken van stomata.



Figuur 2.14: Verband tussen aantal en grootte van de stomata en de CO_2 -concentratie.

Uit fossiel plantenmateriaal blijkt dat planten reageren op langdurige verandering in CO_2 door hun stomatale geleidbaarheid te verhogen of te verlagen via een aanpassing in de dichtheid van de stomata (Franks & Beerling, 2009; de Boer et al., 2016). Behalve de fotosynthese, zijn ook de transpiratie en de waterhuishouding van de planten factoren die daarin meespelen.

De hoeveelheid CO_2 in de atmosfeer kan daarom afgeleid worden uit de stomatale dichtheid. Aangezien het CO_2 -niveau een direct effect heeft op globale temperaturen, geven veranderingen in het CO_2 -niveau ook

een duidelijk beeld van veranderingen in het klimaat.

Voorbeelden van het verband tussen klimaat en stomata

Op het moment van de massa-extinctie op het einde van het Trias, zo'n 200 miljoen jaar geleden, was er een grote daling in het aantal stomata op bladeren, en dus een stijging van de atmosferische CO₂-concentratie. Er was toen een globale opwarming van 5 °C, wat regionaal een stijging van 16 °C kon betekenen. Met wiskundige modellen ontdekte men dat planten met grote bladeren oververhit geraken bij zulke hoge temperaturen, wat bevestigd werd door het soort planten dat toen effectief verdween. Na de massa-extinctie waren er nauwelijks nog koraalriffen, 20 % van de dieren in de oceanen stierf uit, de helft van de Noord-Amerikaanse gewervelden (o.a. de Phytosauriër) kwam om. Het verdwijnen van sommige fauna en flora kan rechtstreeks gelinkt zijn aan de opwarming, zoals de planten met grote bladeren. Het kan ook onrechtstreeks gebeurd zijn: sommige dieren kunnen uitgestorven zijn omdat ze afhankelijk waren van planten die verdwenen waren. Het gestegen CO₂-niveau en de bijbehorende hogere temperaturen, kunnen gelijkaardige effecten gehad hebben op het leven in de oceaan. Over de oorzaak van de massa-extinctie bestaat veel onzekerheid. De klimaatverandering zou het kunnen verklaren. Het is in elk geval duidelijk dat een globale opwarming van deze omvang mogelijk zeer grote veranderingen in ecosystemen kan veroorzaken.

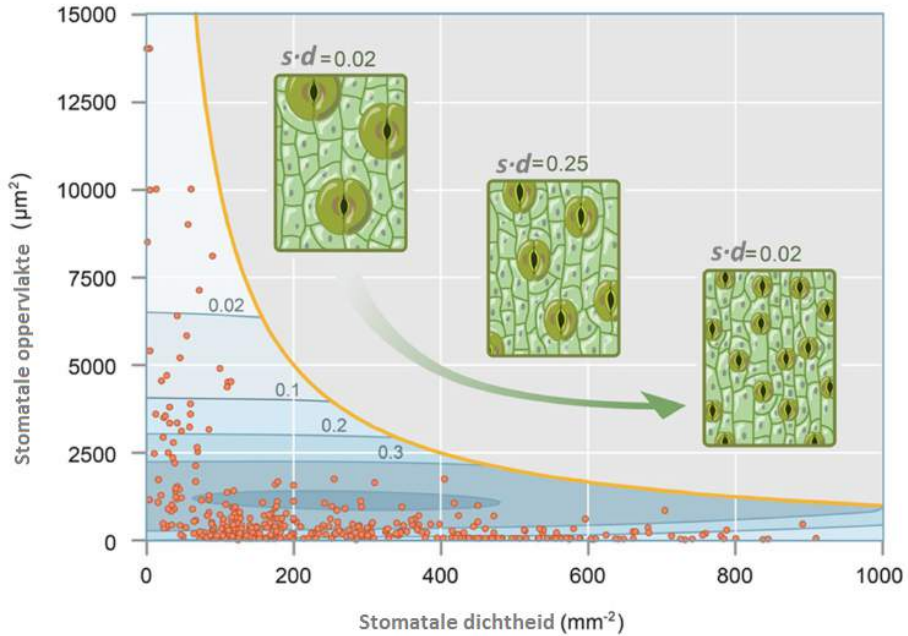
Ook bij fossiele bladeren van de laatste 66 miljoen jaren (Cenozoïcum), wanneer er ingrijpende veranderingen optraden in klimaat, fauna en flora, stelt men correlaties vast tussen globale milieuveranderingen en de evolutie van stomata. Naargelang het heersende klimaat zijn de stomata aan het oppervlak van het blad, dan wel dieper in het blad of in holtes gelegen (Thanukos, 2018).

De stomatale geleidbaarheid is afhankelijk van de dichtheid maar ook van de grootte van de stomata, en hoewel er grote verschillen zijn tussen planten is er een duidelijk verband tussen de grootte en de dichtheid: hoe groter de dichtheid, hoe kleiner de stomata.

Fossiele bladeren tonen dat een evolutionaire verhoging van de stomatale geleidbaarheid, om zich aan te passen aan een dalende of lage CO₂, doorgaans

De stomatale geleidbaarheid is de mate waarin de gasuitwisseling tussen een plant en haar omgeving gebeurt.

gepaard gaat met een stijging van de dichtheid en een verkleining van de stomata (Franks & Beerling, 2009).



Veel planten zijn op zo'n manier geëvolueerd dat de fotosynthese geoptimaliseerd is voor een zo laag mogelijke energiekost. Het bladoppervlak dat ingenomen wordt door de stomata is een goede maat voor de energiekost. Hoe minder bladoppervlak ingenomen wordt door de stomata, hoe kleiner de energiekost is. Deze oppervlakte wordt bepaald door de dichtheid en afmetingen van de stomata (zie Figuur 2.15).

Immers (de Boer et al., 2016):

Een teveel aan stomata zou teveel energie kosten aan de plant, aangezien er veel energie gaat naar het openen en sluiten ervan. De mogelijkheid van planten om die dichtheid te verhogen is weliswaar beperkt: stomata die te dicht bij elkaar staan, kunnen niet gemakkelijk openen en sluiten. Meer stomata zonder de afmetingen ervan te verkleinen, zou ertoe leiden dat ze

Figuur 2.15: Fossil record-based plant leaf stomatal size (s) and density (d). Hoe meer stomata per oppervlakte-eenheid, hoe kleiner de stomata. De planten zijn zo geëvolueerd dat de ingenomen bladoppervlakte ($s \cdot d$) zo klein mogelijk is (Assouline & Or, 2013). Gebaseerd op Franks & Beerling, 2009.

te dicht op elkaar staan. Kleinere stomata zijn dynamischer dan grote, wat een bijkomend voordeel oplevert voor de plant. Een blad dat teveel stomata, dus openingen bevat, wordt meer vatbaar voor ziektes.

Planten hebben doorheen de evolutie dus een evenwicht gezocht tussen een zo klein mogelijk bestede oppervlakte aan stomata en een zo groot mogelijke stomatale geleidbaarheid (de Boer et al., 2016).

Bovendien heeft de evolutie in het plantenrijk geresulteerd in drie soorten planten: C_3 -, C_4 - en CAM-planten. Voor elke soort verloopt de fotosynthese anders (zie kader 'C₃-, C₄- en CAM').

C₃-, C₄- en CAM

De evolutie in het plantenrijk heeft geresulteerd in drie soorten planten: C_3 -, C_4 - en CAM-planten. Voor elke soort verloopt de fotosynthese anders. In warme gebieden hebben C_3 -planten de neiging tot respiratie ten nadele van fotosynthese. C_4 - en CAM-planten hebben het fotosyntheseproces aangepast om dit tegen te gaan. Ongeveer 85 % van de plantensoorten zijn C_3 -planten, zoals granen en groenten (bv. rijst, graan, spinazie, tomaten) en alle bomen (bv. appelbomen, eik). Ze vormen moleculen met 3 C-atomen tijdens het fotosyntheseproces. Ongeveer 5 % van de plantensoorten zijn C_4 -planten, zoals maïs, gierst, suikerriet en vele grassen. Ze maken wel 25 % uit van de planten op aarde en komen vooral voor in tropische, vaak droge gebieden. Ze vormen moleculen met 4 C-atomen tijdens het fotosyntheseproces. De overige 10 % van de plantensoorten zijn CAM-planten, zoals cactus, ananas, Kalanchoë en sedum. Ze komen vooral voor in droge gebieden met hoge dagtemperaturen en lage nachttemperaturen. Ze vormen ook moleculen met 4 C-atomen tijdens het fotosyntheseproces, maar ze verschillen van de C_4 -planten omdat ze 's nachts hun stomata openen om CO_2 op te nemen. Zo beschermen ze zichzelf tegen uitdroging. Die CO_2 wordt in de plant opgeslagen en gedurende de dag, als de nodige lichtenergie voorhanden is, terug beschikbaar gemaakt voor de fotosynthese (Yamori et al., 2013; Sterrenwacht Armand Pien, 2018).

Extra uitleg: CO_2 wordt in alle planten gefixeerd door het enzym Rubisco. Maar Rubisco kan evengoed O_2 fixeren i.p.v. CO_2 . Als dat gebeurt, dan is dat nadelig

voor de plant. Het kost energie en er gaat opgenomen CO_2 verloren, aangezien het de O_2 haalt door het af te splitsen van de opgenomen CO_2 . Dit komt meer voor onder droge of warme omstandigheden en kan dan de fotosynthese veel minder efficiënt maken. Het is waarschijnlijk nog een overblijfsel van 400 miljoen jaar geleden toen de CO_2 -concentratie hoog en de O_2 -concentratie laag waren en het probleem zich niet echt stelde (Gowik & Westhoff, 2011).

2.5 Invloed van het CO_2 -gehalte op de vorming van stomata

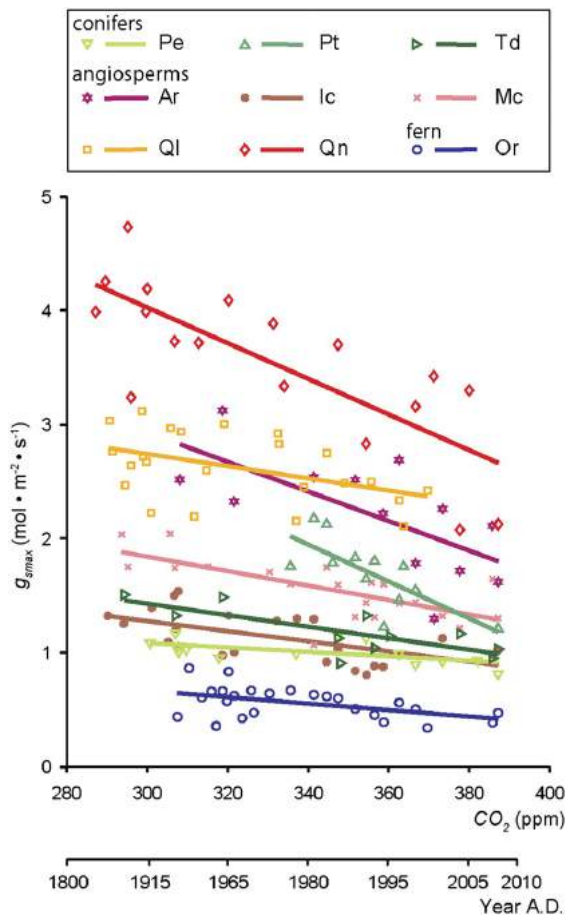
Zowel op lange als op korte termijn leiden veranderingen in de hoeveelheid atmosferische CO_2 tot veranderingen in de stomatale geleidbaarheid van een plant. Op korte termijn kan dat inhouden dat de stomata van een plant minder lang open zijn, wat de waterhuishouding van de plant ten goede komt. Het effect van gestegen concentraties van CO_2 op de vorming en de dichtheid van stomata is uitvoerig bestudeerd m.b.v. fossiele bladeren, herbariummateriaal en recent materiaal.

Hoe de stomatale geleidbaarheid verandert als de hoeveelheid CO_2 aanzienlijk toeneemt of afneemt, verschilt van plant tot plant.

Voorbeelden op langere termijn:

- Over een periode van ongeveer 150 jaar sinds 1840 was de stomatale dichtheid op de bladeren van een bepaalde soort eik, de *Quercus robur* of, de zomereik, gedaald ten gevolge van de stijging van de atmosferische CO_2 -concentratie. Dit werd getoond in een onderzoek uit 1993 (Beerling & Chaloner, 1993).
- Een onderzoek uit 2017 op de *Corynocarpus laevigatus* toonde aan dat de stomatale dichtheid van plantenmateriaal uit de 20ste en 21ste eeuw lager was dan die van herbariummateriaal uit de 18de en 19de eeuw (Large et al.).

- Lammertsma et al. onderzochten of de toename aan CO_2 over de voorbije 150 jaar een invloed had gehad op de vegetatie in Florida (2011). Ze vonden dat bij negen veel voorkomende planten de stomatale geleidbaarheid gedaald was (zie Figuur 2.16) omdat de dichtheid en de grootte van de stomata veranderd waren.



Figuur 2.16: Soorten-specifieke relatie tussen de maximale stomatale geleidbaarheid g_{smax} en CO_2 -concentratie over de voorbije 150 jaar. Symbolen zijn gemiddelde g_{smax} voor elke soort per bestudeerd CO_2 -niveau ($n=160$). De rechten tonen de lineaire regressie tussen CO_2 -concentratie en g_{smax} voor elke soort (Lammertsma et al., 2011).

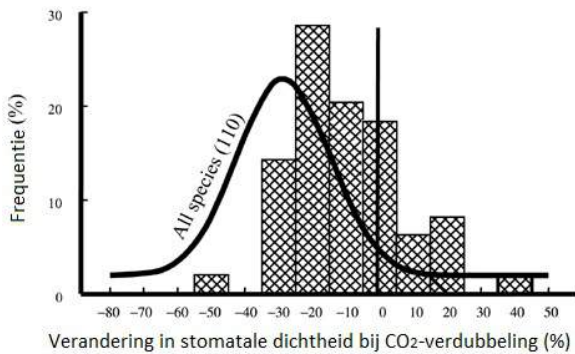
Voorbeeld op korte termijn:

- Uit het onderzoek van Woodward et al. aangevuld met nieuwe data blijkt dat een toename van CO_2 een

duidelijke invloed heeft op de stomata van de *Arabidopsis thaliana* (zandraket, zie Figuur 2.17) gekweekt onder specifieke omstandigheden gedurende 6 weken (Hetherington & Woodward, 2003; Woodward et al., 2002).

Verdubbeling van de CO₂-concentratie leidt ertoe dat de stomatale dichtheid gemiddeld met 11 % afneemt (onafhankelijk van de oorspronkelijke dichtheid) (Hetherington & Woodward, 2003).

Figuur 2.18 toont dat bij 110 soorten de stomatale dichtheid met gemiddeld 29 % is afgenomen en dat zelfs binnen dezelfde soort, hier de zandraket, planten heel verschillend kunnen reageren.



2.6 Invloed van de omgeving op de vorming van stomata

Experimenten bevestigen dat dichtheid en grootte kunnen veranderen door genetische aanpassingen, alsook door veranderingen in omgevingsfactoren. Dit betekent dat zowel tussen plantensoorten, tussen individuele exemplaren van eenzelfde soort, als tussen bladeren van eenzelfde plant verschillen kunnen optreden.

Voorbeelden van verandering in omgevingsfactoren:

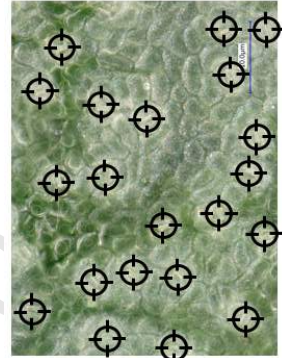


Figuur 2.17: Zandraket (Roepers, 2004).

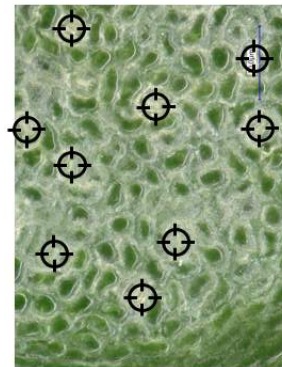
Het proces van de fotosynthese wordt gemaximaliseerd bij de verhoogde CO₂-concentratie, ondanks dat de maximale stomatale geleidbaarheid is afgenomen.

Figuur 2.18: (Gebruik met toestemming van I. Woodward). Verandering in stomatale dichtheid bij CO₂-verdubbeling bij 110 soorten, weergegeven door de kromme (Woodward & Kelly, 1995; niet gepubliceerde observaties) en bij verschillende soorten van de zandraket, weergegeven door het histogram (Woodward et al., 2002).

- Het aantal en de dichtheid van stomata op nieuwe bladeren die worden gevormd, worden beïnvloed door de CO₂-concentratie waaraan volgroeide bladeren blootgesteld zijn (de volgroeide bladeren sturen een signaal uit).
- Bij exemplaren van eenzelfde soort, maar die groeien in verschillende omstandigheden (nl. in de schaduw of in de zon) kunnen verschillen optreden in stomatale dichtheid. Bij een onderzoek op crabwood (*Carapa*) in het Amazonewoud was de stomatale dichtheid bij de bezonde planten groter (Camargo & Marengo, 2012).
- Bij eenzelfde plant kan het aantal stomata op beschaduwde bladeren onderaan de plant verschillen van het aantal stomata op bladeren in de kruin die veel meer blootgesteld zijn aan het licht. Sofie Meeus stelde dat vast op de *Monodora angolensis* in de Plantentuin Meise, zoals te zien op Figuren 2.19 en 2.20.
- Een onderzoek van Beerling & Chaloner uit 1993 op de eik *Quercus robur* toonde dat de stomatale dichtheid op bladeren gevormd in de zomer kleiner was dan de stomatale dichtheid van bladeren gevormd in de lente. Dat was te wijten aan de hoge temperaturen in de zomer. Ze hebben dit zowel op recent materiaal als op herbariummateriaal kunnen vaststellen.
- Omdat de vorming van stomata zo gevoelig is voor omgevingsfactoren, kan men niet zomaar resultaten van experimenten op planten in serres veralgemenen naar planten die buiten groeien.



Figuur 2.19: Bezond blad van *Monodora angolensis*.



Figuur 2.20: Beschaduwde blad van *Monodora angolensis*.

2.7 *Wisselwerking tussen stomatale geleidbaarheid, fotosynthese en transpiratie*

Hoe groter de stomatale geleidbaarheid, hoe meer fotosynthese en hoe meer transpiratie. Hoe meer transpiratie, hoe lager de temperatuur in het blad. Een plant moet voldoende CO₂ opnemen om te groeien, maar ze moet er ook over waken dat ze niet uitdroogt. Ze zal dus voortdurend een balans moeten zoeken tussen voldoende CO₂ opnemen voor fotosynthese, ervoor zorgen dat ze niet oververhit geraakt en erover waken dat ze niet uitdroogt door het ontsnappen van waterdamp te begrenzen (de Boer et al., 2016).

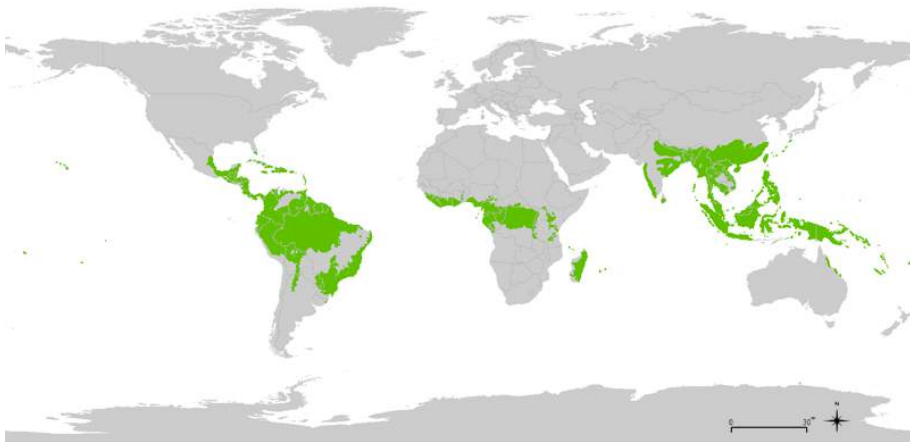
Opbrengsten van geïrrigeerde tarwe en katoen lijken niet af te hangen van fotosynthese, maar wel van de controle over de temperatuur van de plant. Hoge geleidbaarheid helpt deze planten om die temperatuur te controleren (Hetherington & Woodward, 2003).

Watervoorraad is een zeer belangrijke factor in het 'gedrag' van planten (Hetherington & Woodward, 2003). Ongeveer de helft van de planten met een lage stomatale geleidbaarheid groeien in droge condities. Deze planten zullen traag groeien, maar wel over een langere periode. 40 % van de planten met een hoge stomatale geleidbaarheid groeien ook in droge condities. Deze planten zullen snel groeien in periodes dat er wel water ter beschikking is. In die periodes halen ze voordeel uit hun hoge geleidbaarheid en fotosynthesecapaciteit.

De stijging van atmosferische CO₂-concentratie zou de fotosynthese in C₃-planten kunnen versnellen. Dat zou toelaten de stomatale geleidbaarheid te verminderen en dus een efficiëntere waterhuishouding kunnen voortbrengen. Maar aangezien planten heel verschillend reageren op een toename aan CO₂, zou het evengoed kunnen leiden tot een verandering in de biodiversiteit.

2.8 Stomata en klimaatverandering

De water- en CO₂-cycli van de wereld worden beïnvloed door de transpiratie van planten op het land. Men kan dus zeggen dat stomata een impact hebben die globaal is, een impact op de hele wereld. Er wordt het meest getranspireerd door de vegetatie in de warme beboste gebieden van de tropen (zie Figuur 2.21) (Hetherington & Woodward, 2003). De tropische regenwouden situeren zich in gebieden in de buurt van de evenaar. Ze bevinden zich in in Centraal- en Zuid-Amerika, in Afrika, in Azië en op de eilanden rond Australië. Het is belangrijk te begrijpen hoe planten deze transpiratie zullen aanpassen aan de huidige klimaatverandering. Wetenschappers onderzoeken ook hoe de huidige klimaatverandering de vorming van stomata beïnvloedt.



Figuur 2.21: De tropische regenwouden zijn groen gekleurd op de kaart.

Een veranderend klimaat heeft een impact op de biodiversiteit en omgekeerd kan de biodiversiteit ook de klimaatverandering beïnvloeden. Bv. transpiratie verkoelt niet enkel de plant, maar ook haar omgeving. Er zijn grote verschillen in hoe planten reageren op de CO₂-toename en in de mate waarin planten CO₂ opne-

men.

Meer onderzoek naar de groei van planten en de mate waarin planten CO₂ opnemen bij meer CO₂ en een veranderend klimaat, is broodnodig, wil men de impact op de biodiversiteit kunnen inschatten en is van belang voor een juiste input in de klimaatmodellen. Het is belangrijk de balans tussen de energiekost en een betere stomatale geleidbaarheid te begrijpen omdat men dan de aanpassing van moderne planten aan veranderende omstandigheden beter begrijpt. Het is bv. cruciaal dat men inzicht krijgt in de aanpassingen van gewassen om de voedselvoorziening te kunnen bewaken.

- Men meende dat planten dankzij de klimaatverandering ook in gebieden zouden groeien waar ze voordien niet groeiden omdat het te koud was. Deze veronderstelling bleek niet correct. Recent onderzoek van Milbau et al. (2017) wees uit dat enkel een verhoging van de temperatuur daarvoor niet voldoende was: ontkieming van de zaden verliep goed, maar de planten stierven erna toch af.
- De toename aan CO₂-concentratie in de atmosfeer zou de fotosynthese in C₃-planten, zoals rijst, graan en alle bomen, kunnen versnellen. Wetenschappers van de universiteit van Minnesota in de VS deden een experiment waarvan de uitkomst hen doet vermoeden dat de toename aan CO₂-concentratie in de atmosfeer slechts voor enkele jaren een positief effect zal hebben op de groei van C₃-planten (Reich et al., 2018). Indien dat vermoeden juist is, moeten de klimaatvoorspellingen worden aangepast, want dan zullen deze planten minder CO₂ opnemen dan initieel gedacht.
- Veel mensen hebben rijst als voornaamste voedingsbron. Het telen van rijst vergt echter veel water. De oogst is bedreigd door de klimaatverandering. Wetenschappers van de universiteit van Sheffield

ontdekten dat men rijstplanten kan cultiveren zodat ze minder stomata hebben (Caine et al., 2019). Mogelijk kunnen ze het water dan beter vasthouden en zijn ze beter bestand tegen hoge temperaturen en droogte.

Het klimaatsysteem en de koolstofcyclus

Het complexe klimaatsysteem van de aarde wordt bepaald door het broeikas-effect, de atmosferische circulatie en de wisselwerking tussen atmosfeer en oceanen. Maar ook de vegetatie en de hoeveelheid ijs op aarde spelen een belangrijke rol. Bovendien treden er terugkoppelingsmechanismen op, die een verstoring van het klimaatsysteem kunnen versterken of afremmen. Ook de koolstofcyclus speelt hierin een belangrijke rol.

Radiative equilibrium

Het zonlicht dat op aarde invalt, verwarmt de aarde. Die verwarming gebeurt ongelijkmatig verdeeld over de aarde. Aan de evenaar is de verwarming groter. Maar de warmte wordt herverdeeld door de wisselwerking tussen atmosfeer en oceanen. Denk aan regen, wind en bijvoorbeeld de Warme Golfstroom. Landoppervlak, oceanen en atmosfeer absorberen een deel van het zonlicht; ze sturen ook straling terug de ruimte in.

Als er globaal gezien evenveel straling inkomt als uitgaat, dan is de aarde in een *radiative equilibrium*, er is een evenwicht. Een vermindering of vermeerdering van de globale hoeveelheid inkomende straling of van de globale hoeveelheid uitgaande straling, verstoort dat evenwicht en verandert de globale temperatuur op aarde. Een verschil tussen inkomende en uitgaande straling heet *radiative forcing*. De *radiative forcing* is positief als er meer straling inkomt dan uitgaat, hierdoor stijgt de globale temperatuur. Bij een negatieve *radiative forcing* gaat er meer straling uit dan er inkomt, en dan daalt de globale temperatuur. Voor een globale temperatuur die min of meer stabiel is over een lange periode, is dus een *radiative equilibrium* nodig.

De inkomende en uitgaande straling verschilt van plaats tot plaats op aarde. Jaarlijks wordt er bijvoorbeeld aan de evenaar minder straling teruggekaatst dan geabsorbeerd, maar aan de polen wordt er meer straling teruggekaatst dan geabsorbeerd.

De hoeveelheid zonnestraling die de aarde absorbeert, hangt sterk af van de mate waarin de atmosfeer en het aardoppervlak het zonlicht kunnen terugkaat-

sen, de albedo. Witte sneeuw en ijs hebben een hoge albedo. Als sneeuw en ijs smelten, bv. door een toenemend broeikaseffect, heeft dat dus een invloed op de hoeveelheid zonlicht die de aarde weerkaatst. Door de ijskappen op Antarctica en Groenland en het quasi permanente zee-ijs in de Arctische Oceaan, is het globale klimaat zeer gevoelig voor veranderingen in de concentratie van broeikasgassen in de atmosfeer (NASA, 2009).

Radiative forcing van CO₂

Het *radiative equilibrium* kan op natuurlijke wijze worden verstoord, bijvoorbeeld door vulkaanuitbarstingen en schommelingen in de intensiteit van de zonnestraling. De mens verstoort het evenwicht door ontbossing, door het brengen van deeltjes (aerosols) in de atmosfeer en door de uitstoot van broeikasgassen. De activiteiten van de mens van de laatste 150 jaar, zoals de verbranding van grote hoeveelheden fossiele brandstoffen en de aanzienlijke ontbossing, hebben geleid tot een stijging van de CO₂-concentratie in de atmosfeer, een toename van meer dan 40 % sinds het begin van de industriële revolutie. Hierdoor is het *radiative equilibrium* verstoord. De verbranding van fossiele brandstoffen brengen CO₂ in de atmosfeer die reeds miljoenen jaren was opgeslagen in de aardbodem. De ontbossing zorgt ervoor dat er minder vegetatie is om CO₂ op te nemen. CO₂ heeft een grote *radiative forcing*: het absorbeert veel van de straling die door het aardoppervlak wordt teruggekaatst en verhindert zo dat het naar de ruimte wordt teruggekaatst. CO₂ die in de atmosfeer terechtkomt, blijft er ook heel lang, waardoor het effect cumuleert. Omdat koolstofdioxide lang in de atmosfeer blijft, kan de huidige CO₂-uitstoot ook in de verre toekomst nog voor opwarming zorgen, ook al zouden we geen CO₂ meer uitstoten (NASA, 2009).

Terugkoppelingen

Het klimaatsysteem is zeer complex. De aerosolen, afkomstig van vulkaanuitbarstingen maar vooral van vervuiling, zorgen voor een verkoelend effect, want deze deeltjes houden de inkomende straling van de zon tegen, ze verhogen de albedo. Door de ontbossing is er minder vegetatie om CO₂ op te nemen, wat leidt tot opwarming. Tegelijkertijd heeft ontbossing ook een impact op de albedo; naargelang de lokale omstandigheden kan ontbossing de albedo verhogen of verminderen, en respectievelijk leiden tot opwarming of tot verkoeling. Als in de plaats van het bos grasland of akkers komen, die meer inkomend zonlicht

reflecteren dan bos, dan kan ontbossing de albedo wat verhogen, maar als in de plaats van het bos een zeer donker landoppervlak verschijnt, dan zal de albedo afnemen.

Er spelen ook terugkoppelingsmechanismen, of feedbacklussen, die een verstoring van het klimaatsysteem versterken of afremmen. Men spreekt van positieve en negatieve terugkoppelingen. Een positieve terugkoppeling is er een die de verstoring versterkt, een negatieve terugkoppeling is er een die de verstoring tegenwerkt. Omdat het ene mechanisme versterkt, maar het andere afremt, zorgen ze in principe voor het bewaren van een evenwicht.

Hoe groot de impact van de feedbacklussen is, is moeilijk in te schatten. Maar het staat vast dat de globale temperatuur omhooggaat, dus de totale radiatieve forcing is ondanks de feedbacklussen momenteel positief (Anderson et al., 2016; Bralower & Bice, 2020; O'Halloran et al., 2012; Tans, 2018).

Voorbeelden:

- De ijsterugkoppeling: De opwarming zorgt ervoor dat eeuwige sneeuw, gebergtegletsjers, de ijskappen op Antarctica en Groenland smelten, waardoor de van de aarde vermindert. Gevolg is dat er minder straling teruggekaatst wordt naar de ruimte en de aarde dus nog meer opwarmt.

Het smelten van sneeuw en ijs is een positieve terugkoppeling.

- De waterdampterugkoppeling: Door de opwarming kan de atmosfeer meer waterdamp vasthouden en komen er meer wolken. Meer wolken zorgen ervoor dat er meer invallend zonlicht wordt weerkaatst, waardoor een verkoeling optreedt. Maar anderzijds is waterdamp wel een broeikasgas, dus de waterdamp absorbeert een deel van de straling die door de aarde wordt weerkaatst, waardoor de atmosfeer meer opwarmt.

Het vasthouden van meer waterdamp door de atmosfeer is dus zowel een negatieve als positieve terugkoppeling. De vraag is welke van de twee de bovenhand heeft.

- De fotosyntheseterugkoppeling: Als er meer CO₂ in de atmosfeer komt, dan kunnen planten beter groeien, omdat ze meer CO₂ gaan opnemen en dus zorgen voor verkoeling, een negatieve terugkoppeling dus. Op de groei van planten zit wel een limiet omdat ook de nodige voedingsstoffen voorhanden moeten zijn. Door de opwarming wordt de groei van planten

tegelijk ook afgeremd door droogte, waardoor ze minder CO₂ gaan opnemen. Het is dus moeilijk te zeggen of op termijn, bij opwarming van de aarde, de CO₂-opname van landplanten de opwarming zal versterken of tegenwerken.

Wat de tropen betreft, waar de bossen een enorme hoeveelheid CO₂ opslaan, is er bijvoorbeeld voor het Amazonewoud een daling aan CO₂-opslag ingezet, omdat er meer bomen sterven dan er bijkomen. Voor Afrika is dat nog niet het geval, maar dat wordt wel voorspeld (Hubau et al., 2020).

- De oceaanterugkoppeling: Bij meer atmosferische CO₂ en dus opwarming, nemen de oceanen meer CO₂ op, wat een verkoelend effect heeft.

Hier is sprake van een negatieve terugkoppeling.

De opname van CO₂ door de oceanen is echter beperkt. Door het opnemen van CO₂ verzuren de oceanen waardoor ze op termijn minder CO₂ zullen kunnen opnemen. Als de oceanen meer CO₂ opnemen, dan kan ook het plankton in de oceanen meer aan fotosynthese doen en beter groeien. Maar in een te warme oceaan vermindert de toename aan plankton.

Het is dus ook moeilijk in te schatten of de opslag van CO₂ door de oceanen in de toekomst een positieve of negatieve terugkoppeling zal zijn.

- De permafrostterugkoppeling: Als gevolg van de opwarming ontthooft de permafrost waardoor de koolstof die wordt vastgehouden door de permafrost, vrijkomt. De vraag is in welke vorm de koolstof vrijkomt.

Komt er extra methaan in de atmosfeer of wordt het als koolstofdioxide door het smeltwater meegevoerd naar rivieren en oceanen? Extra methaan in de atmosfeer, bijvoorbeeld, zou leiden tot meer opwarming - meer nog dan CO₂ - wat tot nog meer permafrostontthooft. Hier is sprake van een positieve terugkoppeling.

- Door de opwarming ontstaan wereldwijd meer spontane bosbranden, of *wildfires*. Hierdoor komt meer CO₂ in de atmosfeer en zijn er bovendien minder bomen die CO₂ kunnen opnemen, waardoor de aarde nog meer opwarmt.

Het ontstaan van spontane bosbranden is dus een positieve terugkoppeling.

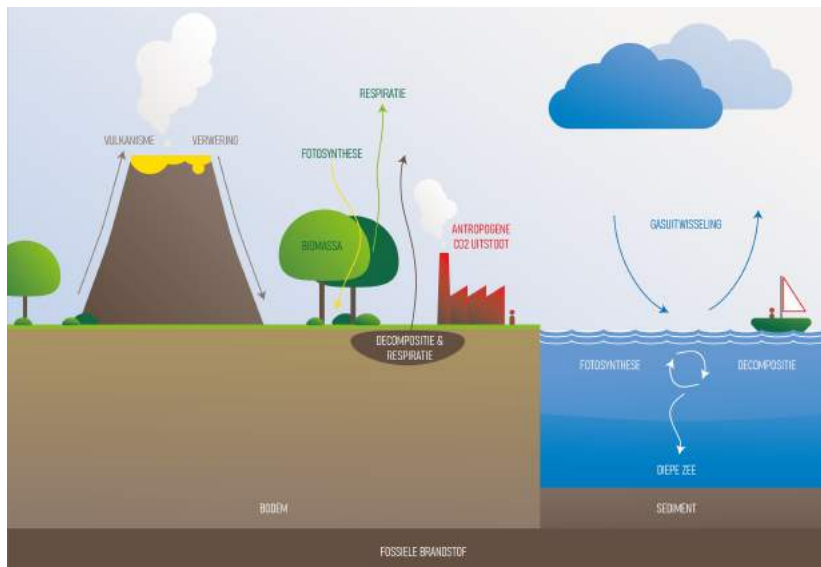
- Hogere temperaturen leiden er ook toe dat meer mensen airconditioning installeren. Het grotere energieverbruik zou tot de verbranding van meer

fossiele brandstoffen kunnen leiden, waardoor meer CO₂ wordt uitgestoten en de aarde meer opwarmt.

Hier is er dus sprake van een positieve feedbacklus.

Koolstofcyclus

Er is een vaste hoeveelheid koolstof op aarde. Ze komt voor in planten en dieren, in de atmosfeer en in de oceanen. Meer dan 99 % van alle koolstof op aarde is opgeslagen in de aardkorst, bijvoorbeeld in kalksteen. Er is een voortdurende uitwisseling van koolstof tussen oceanen, atmosfeer, terrestrische biosfeer en aardkorst.



Koolstofcyclus.

Door groeien en sterven van planten en dieren, door vulkaanuitbarstingen, door verwerking van gesteente en door variatie in de temperatuur van de oceanen wordt er voortdurend koolstof opgenomen en uitgestoten. De plaatsen die meer koolstof bijhouden of opnemen dan uitstoten, noemt men koolstofreservoirs of *sinks*; koolstofbronnen, *sources*, zijn plaatsen die meer koolstof uitstoten dan opnemen. Bijvoorbeeld: een bos dat in biomassa toeneemt, is een sink, een brandend bos is een bron. Oceanen zijn sinks op twee manieren: een deel van

de atmosferische CO₂ wordt opgelost in de oceanen en het plankton neemt CO₂ op voor fotosynthese; deze CO₂ belandt op de zeebodem als het plankton afsterft en naar de bodem zinkt (Beerling, 2017; Maslin, 2014; NASA earth observatory, 2011).

In de lange geschiedenis van de aarde zijn er binnen de koolstofkringloop twee componenten ontstaan: een op lange termijn en een op korte termijn. Het klimaat op aarde hangt sterk samen met de koolstofcyclus.

Langetermijnkoolstofcyclus

Deze evenwichtige cyclus beslaat honderdduizenden tot miljoenen jaren en werkt als een soort thermostaat voor de aarde. De grootste hoeveelheid koolstof is opgeslagen in de aardkorst. Als vulkanen uitbarsten, komt CO₂ van diep in de aardkorst in de atmosfeer terecht. Die toename aan atmosferische CO₂ zorgt ervoor dat de regen wat verzuurt. Door deze regen lost kalksteen op, en wordt het bicarbonaat (HCO₃⁻) dat vrijkomt, door water naar de rivieren gevoerd, waarna het uiteindelijk in de oceanen terechtkomt. De aanwezigheid van planten versnelt de verwerking van gesteente, waardoor er nog meer CO₂ uit de atmosfeer wordt verbruikt en het koeler wordt. Mariene organismen gebruiken het bicarbonaat om een skelet, schelpen en schalen te vormen. Als de organismen afsterven, komen skelet, schelpen en schalen op de oceaانبodem terecht. Bij het bewegen van de tektonische platen kunnen deze in de aardkorst terechtkomen. Door hitte en druk leiden ze daar tot vorming van CO₂. Na dit jarenlange proces komt de CO₂ uiteindelijk via vulkaanuitbarstingen opnieuw in de atmosfeer terecht. De gesmolten kern van de aarde is dus belangrijk om dit proces op gang te houden. Er is hier een negatief terugkoppelingsmechanisme aan het werk: het verwerken van kalksteen is sterk afhankelijk van de temperatuur. Hoge temperaturen versnellen de verwerking, koude temperaturen remmen de verwerking af. Veel vulkanische activiteit leidt tot meer atmosferische CO₂ en een warmer klimaat. De verwerking van kalksteen neemt toe waardoor er meer CO₂ wordt verbruikt. De CO₂-concentratie in de atmosfeer daalt en het klimaat verkoelt. Als het te koud wordt, dreigen de planten uit te sterven - ze doen niet meer aan fotosynthese - en vertraagt de verwerking; er wordt minder CO₂ verbruikt (Beerling, 2017; Maslin, 2014; NASA earth observatory, 2011).

Kortetermijnkoolstofcyclus

De CO₂ die op korte termijn verplaatst wordt, is vele malen groter dan de hoe-

veelheid op lange termijn, en heeft een rechtstreekse impact op de heersende klimaatverandering. Er is een voortdurende wisselwerking tussen atmosfeer, terrestrische biosfeer, oceanen en bodem. Koolstof is een van de bouwstenen van alle leven. Planten nemen CO₂ uit de atmosfeer op voor fotosynthese en geven een deel terug door respiratie. Als de planten groeien, wordt koolstof opgeslagen in hun wortels, stam, stengels en bladeren. Dieren ademen CO₂ uit en slaan koolstof op als ze planten eten. Zo wordt koolstof vastgezet in fauna en flora. Bij sterven van plant of dier komt de koolstof weer vrij: een deel verdwijnt in de bodem, het grootste deel komt terug in de atmosfeer. De bodem is een belangrijk koolstofreservoir, waar ook aan opname en respiratie gedaan wordt door micro-organismen, het bodemleven (Wageningen University & Research, 2020).

Atmosferische CO₂ lost op in de bovenste oceanlagen. Plankton in de bovenste lagen van de oceaan neemt CO₂ op voor fotosynthese. Het plankton slaat dus ook koolstof op, die na afsterven van het plankton naar de bodem van de zee zal zinken. Koolstof in de aardbodem of op de zeebodem wordt voor lange tijd aan de cyclus onttrokken. In miljoenen jaren kan organische koolstof in de zeebodem omgevormd worden tot aardgas en aardolie, en in de bodem op het land tot steenkool.

De hoeveelheid koolstof die tussen land, oceaan en atmosfeer wordt uitgewisseld, is afhankelijk van de temperatuur van het zeewater en van de hoeveelheid atmosferische CO₂. CO₂ lost beter op in koud dan in warm water. Aan de polen wordt veel CO₂ door de oceanen opgenomen, aan de evenaar wordt er vrijgegeven. Plankton groeit niet goed meer als het water te warm is. Als er meer CO₂ in de atmosfeer zit, dan wordt er door de bovenste lagen van de oceanen meer CO₂ opgenomen. De laatste jaren hebben de oceanen veel meer CO₂ opgenomen dan normaal, waardoor ze verzuren en opwarmen. Door de opwarming en de verzuring van de oceanen zullen de oceanen op termijn echter een kleiner deel CO₂ opnemen, want de groei van plankton zal afgeremd worden en warmere oceanen nemen minder CO₂ op.

Bij veel atmosferische CO₂ neemt ook de terrestrische biosfeer meer CO₂ op. De planten doen meer aan fotosynthese en groeien meer. Maar ook hier zijn er beperkingen. Voor de groei van planten moeten er in de bodem voldoende voedingsstoffen en voldoende water zijn. Voor de toekomst wordt meer extreem weer voorspeld: afhankelijk van de regio zijn dat bv. extreme regenval of lange periodes van droogte. Veel planten zullen daar onder lijden of er zelfs aan

bezwijken.

Omdat land en zee extra CO₂ hebben opgenomen, is slechts de helft van de antropogene CO₂-uitstoot in de atmosfeer terug te vinden. De oceanen hebben een groot deel van de extra warmte opgenomen en hebben zo de opwarming getemperd. Om de klimaatmodellen beter af te stellen, moet men weten hoeveel CO₂ de reservoirs exact uitwisselen. Daarover bestaat echter nog onzekerheid. De hoeveelheden kunnen van jaar tot jaar schommelen en kunnen vaak niet rechtstreeks gemeten worden. Wetenschappelijk onderzoek leidt tot nieuwe inzichten (Friedlingstein et al., 2019; Resplandy et al., 2018).

(Beerling, 2017; Maslin, 2014; NASA earth observatory, 2011)

Samengevat

Planten passen zich aan aan veranderende omstandigheden, zoals licht, seizoen, temperatuur, watervoorraad en de atmosferische CO₂-concentratie. Via hun huidmondjes regelen ze hun waterhuishouding en zorgen ze ervoor dat ze CO₂ opnemen om te groeien.

Doorheen de evolutie hebben planten hierin een evenwicht gevonden. Hoe hoger de CO₂-concentratie, hoe minder huidmondjes. Hoe groter de stomatale dichtheid, hoe kleiner de huidmondjes. De hoeveelheid CO₂ in de atmosfeer kan daarom afgeleid worden uit de stomatale dichtheid. Men kan dus de dichtheid en de afmetingen van huidmondjes gebruiken als tool om het heersende klimaat waarin de planten groeiden, te reconstrueren.

Het is belangrijk om te weten hoe snel planten zich aanpassen, om bv. te kunnen inschatten hoe de biodiversiteit en de voedselvoorziening beïnvloed zullen worden door een toenemend CO₂-gehalte.

Daarom is het interessant om te onderzoeken of de verandering van CO₂-concentratie tijdens de voorbije 100 jaar reeds een meetbaar effect teweegbracht op de stomata van planten.

Leestip

David Beerling. The Emerald Planet (Beerling, 2017).

Kijktip

Stomatal Closure in Tradescantia Leaf Cells.

<https://youtu.be/AwyrqfNTuxQ> (davicjal, 2015).

Monocot and Eudicot Germination Time-lapse.

<https://youtu.be/WbG5zu2Vw0I> (Sci- Inspi, 2018).

Stomata and global climate cycles.

<https://youtu.be/eD2J3PBoERI> (Bergmann, 2015).

Versie 1.0

HOE PASSEN BOMEN UIT HET TROPISCH REGENWOUD ZICH AAN AAN DE KLIMAATVERANDERING?

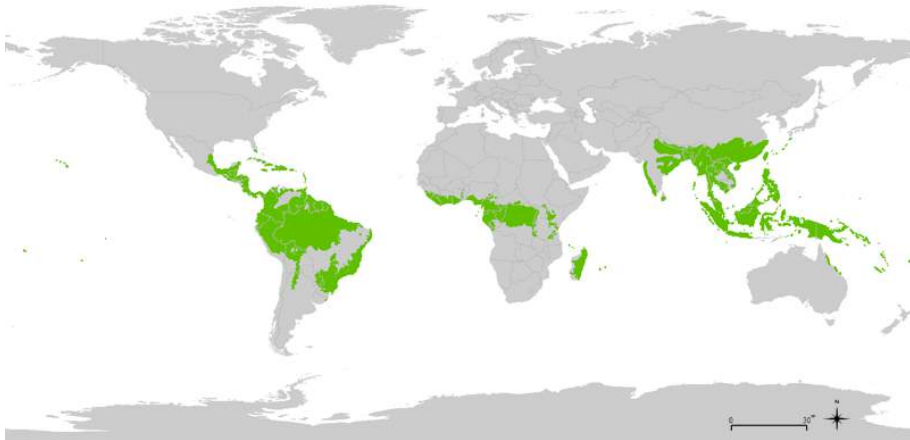
3.1 *Het onderzoek*

Wetenschappers van de Plantentuin Meise en de UGent onderzoeken hoe bomen uit het tropisch regenwoud zich aanpassen aan de klimaatverandering. De stomata op hun bladeren verschaffen informatie over de CO₂-concentratie in de atmosfeer ten tijde van de groei van de planten. De onderzoekers tellen het aantal stomata op de bladeren en ze meten hun grootte op. Erna vergelijken ze de resultaten van recent materiaal met die van materiaal van honderd jaar terug.

Het tellen en meten van stomata is echter een heel tijdrovende bezigheid. Om dit te automatiseren, heeft een computerwetenschapper van de UGent een neurale netwerk getraind (Meeus et al., 2020; Bauters et al., 2020).

Het tropisch regenwoud (Figuur 3.1) is weinig toegankelijk en de bomen zijn vaak zeer hoog, waardoor het moeilijk is om er bladeren van planten te gaan verzamelen. Gelukkig zijn er veel bladeren te vinden in herbaria van musea en botanische tuinen, waar ze worden bewaard in de juiste omstandigheden. Het herbarium

van de Plantentuin Meise bevat behalve veel gedroogd plantenmateriaal, dat verzameld is in Congo tijdens de Belgische koloniale periode (van 1908 tot 1960), ook heel recent materiaal, verzameld in de 21ste eeuw (zie paragraaf 'Afrikaans herbarium van de Plantentuin Meise').



Figuur 3.1: De groene gebieden vormen de tropische regenwouden.

Het tellen en meten van stomata is zeer tijdrovend. Door dit tellen en meten te automatiseren, kan men het wetenschappelijk onderzoek waarvoor men afmetingen van stomata nodig heeft, versnellen. Artificiële intelligentie biedt hierop een antwoord: een convolutioneel neurale netwerk is uitermate geschikt om beelden, en dus ook huidmondjes op bladeren, te herkennen.

Om zo'n netwerk te trainen in het detecteren van huidmondjes, toont men heel veel voorbeelden aan het netwerk: foto's van stomata en foto's van delen van bladeren waarop zich geen stoma bevindt.

Om een netwerk te trainen in het meten van de stomata, toont men voorbeelden van bladeren waarop de afmetingen van de aanwezige stomata zijn aangeduid.

Het team van de UGent en de Plantentuin Meise bouwt

zulke neurale netwerken. Eens de netwerken getraind zijn en dus klaar voor gebruik, zullen ook andere onderzoekers ze kunnen aanwenden voor hun eigen tellingen en metingen van stomata.

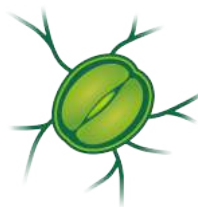
Het team beschikt over voldoende foto's waarop de stomata manueel zijn aangeduid. Men noemt dit annoteren. Het resultaat is een geannoteerde foto zoals in Figuur 3.2. Uit deze geannoteerde foto's werden reeds voorbeelden voor het netwerk gegenereerd: de voorbeelden van stomata en de voorbeelden van delen van bladeren zonder stomata. Vervolgens gebruikte men de geannoteerde foto's om een diep neuraal netwerk voor het tellen van stomata te trainen (zie hoofdstuk 6).

Wat betreft het meten van stomata is er nog wat werk aan de winkel. Via het burgerwetenschapsproject 'Leaf prints' op zooniverse.org wordt daarom gevraagd dat mensen op zoveel mogelijk microfoto's van bladeren de afmetingen van de aanwezige stomata manueel aanduiden.

3.2 Verzamelen van de dataset

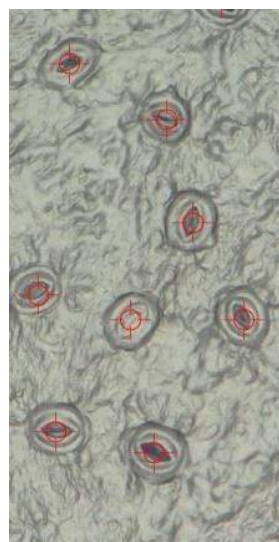
Stomata zijn microscopisch kleine onderdelen van een plant, ze zijn tussen 10 en 100 micron lang. De dataset bestaat uit foto's van bladeren genomen met een microscoop. In het vervolg spreken we van microfoto's.

De onderzoekers kozen de bladeren voor huidmondjestellingen en -metingen uit de Afrikaanse herbariumcollectie van de Plantentuin Meise. Deze collectie bevat meer dan 80 % van alle herbariumspecimens uit Burundi, Rwanda en Congo. Het materiaal is gedroogd. De herbariumspecimens werden verzameld in het Yangambi biosfeerreservaat, gelegen in het stroomgebied van de Congo ten westen van de stad Kisangani in de Democratische Republiek Congo (0,76 N-0,87



Zelf huidmondjes aanduiden op foto's kan online via <https://aiopschool.be/kiks/dl/data/annoteren/>.

'Leaf prints' kadert in het COBECORE-project <http://cobecore.org/>.



Figuur 3.2: Geannoteerde foto van *Entandrophragma utile*.

N, 24,45 E-24,52 E). Het materiaal stamt uit drie verschillende decennia: (1) 1935-1938, (2) 1951-1953, (3) 2012-2013.

Afrikaans herbarium van de Plantentuin Meise

De Plantentuin Meise heeft al meer dan 200 jaar ervaring met planten. Al sinds het einde van de negentiende eeuw houden wetenschappelijk onderzoekers er zich bezig met de studie en het verzamelen van planten uit Centraal-Afrika. Ten tijde van de kolonie Belgisch-Kongo werd de Afrikaanse flora grondig bestudeerd en werden er behoorlijk veel specimens van verzameld. 80 % van de specimens ooit verzameld in Congo, Rwanda en Burundi zijn aanwezig in het herbarium van de Plantentuin.

Op de website van de Plantentuin pakken ze er dan ook trots mee uit dat de "Plantentuin Meise wereldwijd het meest toonaangevende instituut is voor informatie over planten en paddenstoelen uit Centraal-Afrika". Dankzij de verzamelaars, zoals Jean Louis, bevat het Afrikaans Herbarium een miljoen specimens.



© Plantentuin Meise
JEAN LOUIS.

Institut National pour l'Etude Agronomique du Congo Belge
HERBIER De Jean LOUIS

No. 2547.

Date: 1er septembre 1936.

Localité: Route Yangambi-Bengamisa, au kilomètre 35 de Weko.

Altitude: 470 m.

Formation: Forêt primitive lourde.

Noms vernaculaires: "YOLI"
(dial. Turumbu).

Observations: Arbre gigantesque et majestueux, de 9,25 m de circonférence à hauteur d'homme et d'env. 60 m de hauteur totale, dont environ 35 m de fût. Tronc bien droit, flancé, cylindrique saur à la base où il est épaissi en cônes sans accotements. Epaves cime hémisphérique brusquement et largement étalée au sommet du fût. Ecorce brun-noirâtre à profonds sillons longitudinaux.
Les Yoli sont souvent abimés à la base par les indigènes qui prélèvent des fragments de leur écorce dont la macération leur sert comme lavement.

Photographié le spécimen.

Famille: SAPOTACEAE.

Nom botanique: *Aitranella congolensis* (D.W.) Chev.-valier.

Déterminé par: *Notities Louis*

© Plantentuin Meise
NOTITIES LOUIS.

Jean Louis (1903-1947)

Jean Louis behaalt in 1925 na zijn Latijn-Griekse humaniora het diploma van

landbouwingenieur. Al gauw vertrekt hij naar Belgisch-Kongo waar hij zich eerst twee jaar op een privéboerderij kwijt aan het verbeteren van de kwaliteit van het slachtvee. Daarna werkt hij op een proefboerderij van de koloniale overheid waar hij rundvee, legkippen en citrusvruchten selecteert en verbetert.

In 1932 keert hij terug naar België. Met een verhandeling over tweezaadlobbigen en naaktzadigen behaalt hij een doctoraat aan de KULeuven.

In 1935 vertrekt hij terug naar de kolonie waar hij tot 1939 aan de slag gaat als hoofd van de wetenschappelijke afdeling van het INEAC (Institut National pour l'Etude Agronomique du Congo belge - Nationaal Instituut voor Landbouwkunde in Belgisch-Kongo). Hij richt er een afdeling Plantkunde op. Hij zorgt voor exotische planten in de botanische tuin en in de wildernis legt hij paden aan om er o.a. de bomen te kunnen observeren.

Louis stelt met veel zorg een herbarium samen dat beschouwd wordt als de belangrijkste botanische verzameling uit het Kongobekken. Zijn Afrikaanse collectie telt maar liefst 17 000 specimens, in veelvoud aanwezig. De Plantentuin Meise bezit een volledige serie van dat herbarium.

In 1939 wordt hij aan het Landbouwinstituut van Gembloux in België aangesteld als assistent, later wordt hij er professor.

In 1943 start hij samen met Joseph Fouarge de publicatie op van een reeks boekjes getiteld "Essences forestières et bois du Congo belge". Deze boekjes bevatten van elke verhandelbare houtsoort een botanische studie en technische gegevens over het hout. Een van de uitgaven gaat over de *Afrormosia elata*.

Jean Louis sterft op 44-jarige leeftijd. Hij wordt herdacht als een innemend en vrijgevig persoon, perfectionistisch en analytisch van geest.

Camille Donis (1917-1988)

Na de middelbare school volgt Camille Donis een opleiding aan het Landbouwinstituut van Gembloux waar hij afstudeert als landbouwingenieur in 1937.

Twee jaar later gaat hij aan de slag bij het INEAC in Belgisch-Kongo.

Tijdens de oorlog wordt hem opgedragen om de logistiek en de veiligheid van de verplaatsing van het Expeditiekorps (Corps expéditionnaires) van Kongo naar Caïro in goede banen te leiden. Hij doorkruist het Midden-Oosten en bekijkt het vanuit het oogpunt van een landbouwdeskundige. Deze ervaring maakt van hem een visionair.

In 1948 publiceert hij het boek "Essai d'économie forestière au Mayumbe", dat later werd bekroond. Ook de studie van het Mayumbebos waar hij samen met

o.a. Joseph Fouarge aan werkte, werd gelauwerd.

In 1949 wordt hij aan het Landbouwinstituut van Gembloux aangesteld als docent tropische bosbouw, later wordt hij er professor. Hij is streng maar begripvol en zal op vele generaties studenten zijn stempel drukken.

In 1952 wordt hij verantwoordelijk voor de afdelingen bos en hydrobiologie van het INEAC.

In 1956 wordt hij directeur van het Nationaal Albert Park en het Nationaal Kagera Park. Hij ijvert voor een groot natuurreservaat in Oost-Kongo, een project dat vaste vorm aanneemt in 1970.

Hij keert terug naar België in 1958. Donis neemt wereldwijd deel aan internationale congressen over bossen en wordt in veel landen aanzien als specialist ter zake.

Bart Wursten

Bart Wursten, geboren in 1953 in Amsterdam, hield als kind al van de natuur.

Na zijn studies wordt hij programmeur in de computertaal COBOL. Hij gaat echter ook aan de slag als freelance fotograaf in de wereld van de popmuziek.

Maar de wijde wereld lonkt. In 1984 zet hij voor het eerst voet op Afrikaanse bodem. Hij reist er door Zaïre, Burundi en Tanzania. Hij wordt verliefd op Afrika en na enkele bezoeken aan Kenia, verhuist hij naar Afrika in 1986.

Hij onderneemt grote reizen door Centraal-, Oost- en Zuid-Afrika. Dan krijgt hij in 1988 de kans om voor een Nederlandse touroperator safari's te organiseren in Zimbabwe en Botswana. De volgende tien jaar gidst hij groepen toeristen langs de belangrijkste bezienswaardigheden van deze landen. Buiten het toeristisch seizoen reist hij zelf het continent door. Zijn fiets brengt hem zelfs naar afgelegen plekken in Centraal-Zaïre, Uganda en Somalië.

In 1998 koopt hij een eigendom in Vumba, Zimbabwe, en opent er een Bed & Breakfast, samen met zijn vrouw.

Wegens het landhervormingsprogramma en de daaruit ontstane politieke onrust komen er na 2000 minder toeristen naar Zimbabwe. Wursten krijgt daardoor meer vrije tijd om te genieten van al het moois dat de natuur in Vumba te bieden heeft. Te midden van de rijke flora ontwikkelt hij een passie voor planten. Hij maakt dankbaar gebruik van de mogelijkheden die nieuwe technologieën, zoals laptops en digitale fotografie, bieden om de diversiteit aan planten vast te leggen. In 2003 ontmoet hij Mark Hyde. De botanische kennis van Mark Hyde en de fotografische skills van Wursten zijn een winnende combinatie die leidt tot de

website Zimbabwe Flora.

Wursten zet zich nadien ook in voor het in ere herstellen van een nationaal park en het begeleiden van bergexcursies in Mozambique. Ondertussen bouwt hij in Mozambique een herbarium op.

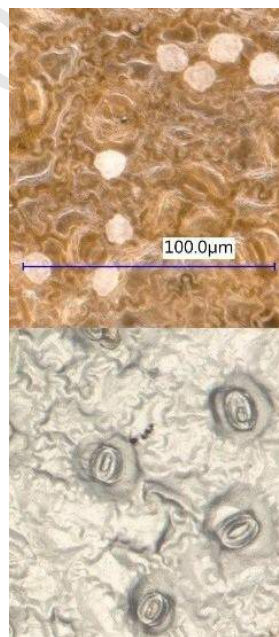
In 2007 verhuizen Wursten en zijn vrouw verplicht naar Europa, ze gaan in België wonen. Ze werken nog steeds verder aan Zimbabwe Flora en gaan nog elk jaar terug. Wursten werkt ook voor de Plantentuin Meise. In 2010 neemt hij deel aan een expeditie van zes weken over biodiversiteit langs de Kongorivier. Al het recente materiaal gebruikt voor het wetenschappelijk onderzoek van KIKS werd door hem verzameld.

(Commissie voor de Belgische Overzeese Biografie, 2015; Hyde et al., 2019)

Om de trainingset te genereren gebruikten de onderzoekers 19 gewone tropische boomsoorten die behoren tot 12 families van bloeiende planten: *Carapa procera*, *Celtis mildbraedii*, *Cola griseiflora*, *Entandrophragma candollei*, *Entandrophragma utile*, *Erythrophleum suaveolens*, *Garcinia punctata*, *Gilbertiodendron dewevrei*, *Irvingia grandifolia*, *Mammea africana*, *Pentaclethra macrophylla*, *Petersianthus macrocarpus*, *Polyalthia suaveolens*, *Prioria balsamifera*, *Prioria oxyphylla*, *Riciodendron heudelotii*, *Strombosiaopsis tetrandra*, *Trichilia gilgiana*, *Trilepisium madagascariense*.

Voor elk van de 19 soorten namen ze stalen van drie specimens voor elke tijdsperiode, dus 9 exemplaren per soort (uitgezonderd 8 voor *Erythrophleum suaveolens*, *Irvingia grandifolia* en 7 voor *Entandrophragma candollei* en *Mammea africana*). Er werden geen bladeren aan de uiteinden van de takken genomen om bemonstering van jonge onvolgroeide bladeren te voorkomen.

De kwaliteit van gedroogd materiaal is minder goed dan dat van vers materiaal. Zelfs met een goede microscoop zijn de stomata op gedroogd materiaal niet altijd goed te zien, zeker ook omdat het materiaal vaak oud en broos is. Daarom werd van elk blad een afdruk genomen



Figuur 3.3: *Prioria oxyphylla*. Boven: microfoto genomen direct van het blad. Onder: microfoto genomen van een afdruk met nagellak. <http://cobecore.org/blog/photo-micrography/>

met transparante nagellak, wat een betere kwaliteit van foto's gaf (zie Figuur 3.3 en Figuur 3.4).

Door het drogen zijn de stomata gemiddeld 9 % gekrompen in de lengte. Hiermee moet rekening gehouden worden als men de data gebruikt voor wetenschappelijk onderzoek.

De bladafdrukken werden gemaakt van de abaxiale zijde (onderzijde) van vijf bladeren per specimen (de bladeren zijn hypostomateus, d.w.z. dat er enkel stomata aan de onderzijde van het blad zijn), halverwege tussen de hoofdader en rand van het blad, op gelijke afstand tussen de punt en de basis van het blad. Eens de nagellak opgedroogd was, werden de afdrukken met dubbelzijdig plakband op een objectglaasje overgebracht met de afdruk naar boven. Met behulp van een digitale microscoop werd van drie zichtvelden per blad een foto genomen, maar dan duizendmaal vergroot. Zo'n foto heeft dan een formaat van 1600 op 1200 pixels.

De (volledige) huidmondjes per zichtveld werden geteld en deze gegevens werden omgezet naar het aantal huidmondjes per vierkante millimeter (een zichtveld heeft een oppervlakte van $0,09 \text{ mm}^2$). Men bekomt dus de stomatale dichtheid.

Voor het trainen van het neurale netwerk werden foto's uit deze dataset gebruikt (zie Figuur 3.5). Hierbij was er van elke soort ongeveer dezelfde hoeveelheid.

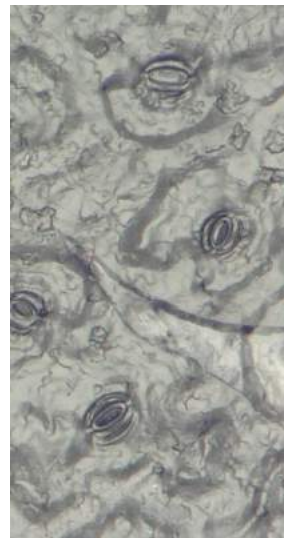
Men beschikte over een trainingset die bestond uit foto's met enkel niervormige stomata. Daarnaast was er ook nog een testset, met behalve niervormige ook haltervormige huidmondjes. De testset werd opgebouwd uit 16 andere soorten tropische bomen van 7 families.

Voor wetenschappelijk onderzoek is het belangrijk dat men bepaalde foto's later snel terugvindt en dat het duidelijk is wat er op de foto's staat. Daarom kregen de



Figuur 3.4: Bladafdruk nemen met nagellak en plakband.

◆ Een zichtveld is $344 \mu\text{m}$ op $258 \mu\text{m}$.



Figuur 3.5: Foto uit de KIKS-dataset: *Coffea boivini* Capuron.

foto's een weloverwogen naam.

Bv. *Coffea pseudozanguebariae*_R&M 1944_blad2-vlak3.jpg: de soortnaam van de plant, gevolgd door de verzamelaar en het nummer van het specimen en eindigend met het gefotografeerde deel van het blad.

3.3 Voorbereiden van de dataset

Om het netwerk te trainen heeft men heel veel voorbeelden nodig: foto's van stomata en foto's van delen van bladeren waarop zich geen (volledige) stoma bevindt.

Zoals eerder gezegd in paragraaf 3.2 werd in de Plantentuin Meise veel werk verzet: er werden heel wat microfoto's genomen van bladafdrukken.

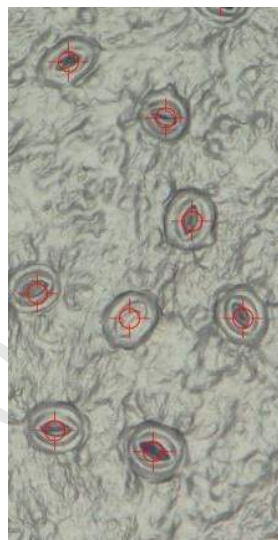
Deze microfoto's van de bladeren werden geannoteerd (Figuur 3.6): het midden van elke stoma werd manueel aangeduid door een mens. Dit moest zo nauwkeurig mogelijk gebeuren. De kwaliteit van de dataset is immers sterk bepalend voor de uiteindelijke prestaties van het neurale netwerk.

Elke foto en de lijst van coördinaten van het midden van elke stoma op die foto werden aan het netwerk doorgegeven.

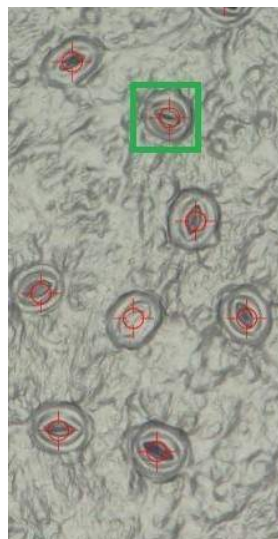
Rond zo'n midden werd een vierkant geconstrueerd van 120 op 120 pixels (Figuur 3.7). De grootte van deze vakken is gekozen afgaande op de grootte van de stomata op de foto's. De stomata op de foto's van de KIKS-dataset passen mooi in zo'n vak.

Het vierkant werd dan 'uitgeknipt' (Figuur 3.8). Zo ontstond één afbeelding: een voorbeeld van een afbeelding van een deel van een blad met stoma, zoals de voorbeelden in Figuur 3.10.

Er waren ook voorbeelden vereist van delen van bladeren zonder een (volledige) stoma erop. Omdat de stomata niet heel dicht bij elkaar liggen - dat zou bv. het gemakkelijk openen en sluiten teveel hinderen - konden



Figuur 3.6: Geannoteerde foto.



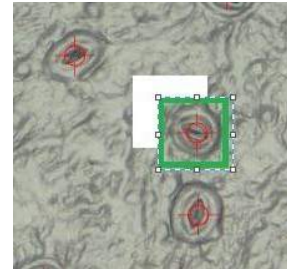
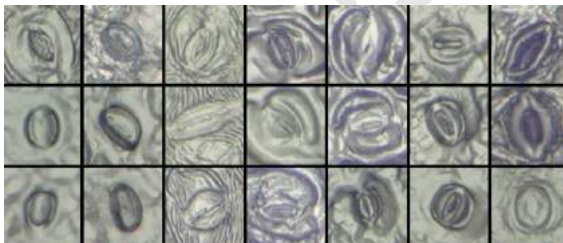
Figuur 3.7: Vierkant rond stoma.

er ook even grote vierkanten geconstrueerd worden in de buurt van een stoma, maar net buiten het vierkant dat een stoma bevat (zie Figuur 3.9).

Deze vierkanten knipte men ook uit. Zo ontstonden de voorbeelden van delen van bladeren zonder stoma (Figuur 3.11).

Op Figuur 3.11 is duidelijk te zien dat deze methode niet waterdicht is: in de voorbeelden zonder (volledige) stoma duikt hier en daar toch een volledige stoma op. Dat komt omdat het annoteren niet altijd even secuur gebeurt.

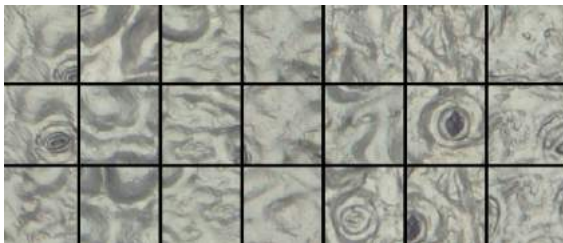
Al deze afbeeldingen van 120 op 120 pixels werden dan gelabeld aan de dataset toegevoegd, 12 000 voorbeelden van stomata en 72 000 voorbeelden van delen van blad zonder stoma. Zowel in de voorbeelden met stomata als in de voorbeelden zonder stomata zit een grote variabiliteit (zie Figuren 3.10 en 3.11). Omdat de variabiliteit bij de voorbeelden zonder stomata echter groter is, bevat de trainingset zes keer meer voorbeelden zonder stomata dan met.



Figuur 3.8: Stoma uitknippen.



Figuur 3.9: Delen van blad zonder stomata.



Figuur 3.10: Voorbeelden van stomata.

Figuur 3.11: Voorbeelden zonder stomata.

Samengevat

Wetenschappers van de Plantentuin Meise en de UGent onderzoeken hoe bomen uit het tropisch regenwoud zich aanpassen aan de klimaatverandering. De onderzoekers tellen daarvoor het aantal stomata op de bladeren en ze meten hun grootte op. Erna vergelijken ze de resultaten van recent materiaal met die van materiaal van honderd jaar terug. Het tellen van de stomata is geautomatiseerd met een neurale netwerk.

Stand van zaken van het wetenschappelijk onderzoek (2020)

Uit het onderzoek blijkt dat de bomen uit het tropisch regenwoud in Afrika zich inderdaad hebben aangepast aan de toename van atmosferische CO₂, maar toch niet helemaal zoals op voorhand ingeschat.

De stomatale dichtheid is afgenomen, zoals verwacht, maar op deze relatief korte termijn van 100 jaar zijn de huidmondjes (nog) niet groter geworden. Het ziet er naar uit dat de toename van atmosferische CO₂ toch niet geleid heeft tot meer fotosynthese. De bomen zijn er ook niet in geslaagd om efficiënter met hun watervoorraad om te springen (Bauters et al., 2020).

BURGERWETENSCHAP

4.1 KIKS

De wetenschappers van de Plantentuin Meise en de UGent doen een beroep op vrijwilligers om de geannoteerde foto's te bekomen die ze nodig hebben voor hun onderzoek. Er wordt gevraagd dat mensen op zoveel mogelijk microfoto's van bladeren de afmetingen van de aanwezige stomata manueel aanduiden.

Deze foto's worden aangeboden via het burgerwetenschapsproject 'Leaf prints' op Zooniverse. Ook leerlingen kunnen hieraan meewerken.

Voor de training van het neurale netwerk van KIKS voor stomatadetectie waren veel foto's nodig waarop de huidmondjes manueel waren aangeduid. Omdat de leerlingen die aan de slag gaan met het KIKS-project, zouden weten wat het annoteren van deze foto's inhoudt, is er op het 'Leaf prints'-project op Zooniverse een luik voorzien waar leerlingen zelf eens foto's voor stomatadetectie kunnen annoteren. Bovendien wordt alle uitleg om het project te kaderen, behalve in het Engels, ook aangeboden in het Nederlands. Er wordt bijvoorbeeld ook uitgelegd hoe de foto's werden genomen.



Doe zelf ook mee aan het 'Leaf prints'-project. 'Leaf prints' kadert in het COBECORE-project.

<https://www.aiopschool.be/kiks/annoteren>

<https://www.zooniverse.org/projects/sofie-dot-meeus/leaf-prints>

Meer uitleg over de wetenschappelijke methode die gebruikt werd om de data te verzamelen, vindt u in het hoofdstuk 3 'Wetenschappelijk onderzoek van de UGent en de Plantentuin Meise' van deze handleiding.

4.2 Burgerwetenschap

Burgerwetenschap of *citizen science* is onderzoek dat rekt op het actieve engagement van vrijwilligers uit het publiek om wetenschappers bij te staan in hun onderzoek i.v.m. maatschappelijk relevante problemen (Wiggins & Crowston, 2011).

Er wordt steeds meer een beroep gedaan op burgerwetenschappers voor wetenschappelijke projecten binnen allerlei onderzoeks domeinen, gaande van biologie en het milieu tot de medische wereld, maar ook binnen historisch en literatuuronderzoek. Citizen science kan immers op grote schaal werken en leiden tot grote hoeveelheden data die zonder de hulp van vrijwilligers moeilijk te verkrijgen zouden zijn. Onder de juiste omstandigheden kan een burgerwetenschapsproject data van hoge kwaliteit genereren en uitmonden in betrouwbare en wetenschappelijk waardevolle resultaten, maar ook leiden tot onverwachte inzichten en innovaties.

Burgerwetenschap kan de kloof tussen de universiteiten en de maatschappij verkleinen. Er kunnen nieuwe onderzoeks domeinen mee geëxploreerd worden. De universiteiten kunnen er hun eigen studenten mee motiveren en participeren aan het onderwijs aan leerlingen van basis- en secundaire scholen. Burgers bekijken de zaken soms op een andere manier en kunnen zelf onderzoeksvragen bij de wetenschappers aanbrengen om onder de loep te laten nemen.

Men moet echter ook opletten voor valkuilen: is het gegenereerd materiaal betrouwbaar, is het ethisch men-

Soms echter wordt een project onterecht voorgesteld als een citizen science project, enkel en alleen om een bepaald wetenschappelijk onderzoek te kunnen verspreiden onder het brede publiek.

Om betrouwbare data te bekomen bijvoorbeeld, zal het nodig zijn de vrijwilligers degelijk op te leiden, zodat ze hun taken goed kunnen volbrengen. Het is ook heel belangrijk de taak zo duidelijk mogelijk en sterk afgelijnd te omschrijven. De betrouwbaarheid van de gegevens kan ook verhoogd worden door bepaalde mechanismen in te bouwen, zoals dezelfde foto door meerdere mensen te laten annoteren of door veel meer metingen te laten doen dan noodzakelijk, zodat foute meetresultaten minder doorwegen.

sen onder het mom van een wetenschappelijk onderzoek voor je kar te spannen terwijl het eigenlijke doel enkel bekendmaking van het wetenschappelijk onderzoek is (Simpson, 2019; De Moor et al., 2019; De Moor, 2019).

4.3 *Samenwerking tussen onderzoekers en burgerwetenschappers*

Een van de uitdagingen bij de opstart van een citizen science project is het kunnen enthousiasmeren van vrijwilligers zodat ze instappen in het project en ze vervolgens geëngageerd houden. Deelname gebeurt immers geheel vrijwillig, er is geen formele overeenkomst en er hangt geen financiële vergoeding aan vast.

Vrijwilligers van wie een behoorlijke intellectuele inspanning wordt gevraagd, hechten veel waarde aan de interactie met de onderzoekers en met de andere burgerwetenschappers binnen het project. Ze verwachten ook dat ze zelf iets bijleren (De Moor et al., 2019).

De samenwerking tussen onderzoekers en burgerwetenschappers verloopt vaak virtueel via een online citizen science platform, zoals DoeDat en Zooniverse, waarbij de deelnemers bepaalde repetitieve taken binnen het onderzoek volbrengen of via een computerspel de wetenschappers aan gegevens helpen.

Daarnaast zijn er ook niet-virtuele projecten waarbij mensen vlinders tellen of de hoeveelheid fijnstof in de lucht meten. Bij sommige projecten waar vrijwilligers in het labo gaan meehelpen, is er ook persoonlijk contact met de onderzoekers.

4.4 *Burgerwetenschap in de klas*

Door op school met leerlingen deel te nemen aan een citizen science project, brengt men hen in contact met

Neem alvast een kijkje
op <https://www.edereenwetenschapper.be/>.

actueel wetenschappelijk onderzoek. Door daarbij verder te gaan dan het verzamelen van data en eenvoudige analyses ervan, en hen te betrekken bij betekenisvolle onderzoekspraktijken, krijgen ze de kans om wetenschappelijke onderzoeksvaardigheden te ontwikkelen. Ze zullen bijvoorbeeld met de onderzoekers moeten communiceren over de vereisten van de betreffende wetenschappelijke methode en hoe ze die moeten toepassen (Nistor et al., 2019).

Er zijn duidelijke verschillen tussen een citizen science project dat is ontworpen voor het brede publiek en een dat moet dienen binnen een schoolse context (Nistor et al., 2019).

- Leerkrachten willen immers leerlingen zaken bijbrengen binnen een pedagogisch kader en gekoppeld aan de leerplandoelstellingen.
- In de klas zijn het de leerkrachten die de leerlingen zullen moeten motiveren i.p.v. de onderzoekers.
- De rol van de leerkracht zal er ook in bestaan toe te zien op de communicatie tussen de leerlingen en de onderzoekers.
- De wetenschappers voegen een extra doel toe aan hun citizen science project. Naast het verkrijgen van data, moeten uit het project ook pedagogische en educatieve resultaten komen. Hiervoor moeten ze overleggen met de leerkrachten over het curriculum. Ze zullen de leerkrachten misschien ook moeten opleiden in wetenschappelijk onderzoek en ze erin trainen hoe ze de wetenschappelijke onderzoeksmethodes in de klas kunnen introduceren. Dit zorgt ervoor dat zowel leerlingen als leerkrachten geëngageerd zullen zijn en draagt bij aan de kwaliteit van de verkregen data.

Samengevat

Wetenschappers starten steeds vaker een burgerwetenschapsproject op om de data te bekomen die nodig zijn voor hun onderzoek.

Ook de wetenschappers van de Plantentuin Meise en de UGent doen een beroep op vrijwilligers om de geannoteerde foto's te verwerven die ze nodig hebben voor hun onderzoek. Deze mensen wordt gevraagd om op zoveel mogelijk microfoto's van bladeren de afmetingen van de aanwezige stomata manueel aan te duiden.

Leestip

De European Citizen Science Association (ECSA) formuleerde 10 principes van citizen science. U vindt ze op <https://www.iedereenwetenschapper.be/article/de-tien-principes-van-de-burgerwetenschap>.

Sinds begin 2020 kunt u terecht op de website van Scivil, het Vlaamse kenniscentrum voor Citizen Science, voor alle informatie over het opzetten van een eigen citizen science project: <https://www.scivil.be>.

Liesbeth Gijssels, Tine Huyse en Ine Van Hoyweghen. Citizen science (Gijssels et al., 2019).

ARTIFICIËLE INTELLIGENTIE

5.1 *Wat is AI?*

Kunstmatige (KI) of artificiële intelligentie (AI) komt volop aan bod in de media. Populaire thema's zijn: chatbots, zelfrijdende auto's, virtuele assistenten en het verlies aan jobs.

Volgens de medeoprichter van Wired-magazine Kevin Kelly zal AI meer en meer onze levens binnendringen en ligt het aan de basis van een nieuwe industriële revolutie (Kelly, 2016). Stanford professor Andrew Ng noemt AI zelfs de nieuwe elektriciteit.

In 2018 verscheen een studie over de positie van Vlaanderen in AI (PwC, 2018). Men rapporteerde: "Er is geen ontsnappen aan. Artificiële intelligentie zal morgen deel uitmaken van alle diensten, maar ook van alle producten, als onderdeel van een evolutie die sommigen vergelijken met de opkomst van elektriciteit. Net zoals vroeger alles 'elektrisch' werd, wordt nu alles 'slim'".

In deze studie hanteert men de definitie die door de Europese Commissie voorgesteld is in de Mededeling over AI voor Europa (2018): "Kunstmatige intelligentie verwijst naar systemen die intelligent gedrag vertonen door hun omgeving te analyseren en – in zekere mate

zelfstandig – actie te ondernemen om specifieke doelstellingen te verwezenlijken. Op KI gebaseerde systemen kunnen louter softwarematig zijn en actief zijn in de virtuele wereld (bijvoorbeeld stemgestuurde assistenten, software voor beeldanalyse, zoekmachines en systemen voor spraak en gezichtsherkenning), maar KI kan ook in hardware-apparaten worden geïntegreerd (bijvoorbeeld geavanceerde robots, zelfrijdende auto's, drones of toepassingen van het internet der dingen). Een definitie geven voor AI, iets dat in zoveel verschillende domeinen wordt ingezet, is geen sinecure. U kunt er meer over lezen in de kaders 'Intelligentie' en 'AI'.

AI heeft inderdaad al succesvolle toepassingen: gezichtsherkenning in China om toegang te verlenen tot gebouwen, AI-systemen die specialisten evenaren in de diagnose van huidkanker, AlphaGO Zero dat zelf go leerde spelen enkel door te spelen tegen zichzelf (Steels et al., 2017). Maar hoewel er ongetwijfeld spannende ontwikkelingen zijn op het vlak van AI, staat de kunstmatige intelligentie vooralsnog mijlenver af van menselijke intelligentie. Een mens kan behoorlijk veel bijleren uit een beperkt aantal voorbeelden, omdat hij ook zijn reeds opgedane kennis en vaardigheden benut. Een mens is bovendien zeer flexibel: we zijn in staat ons snel aan te passen aan veranderende omstandigheden en adequaat te reageren op onverwachte gebeurtenissen. Vandaag de dag kan een computer dat niet. Een AI-systeem is gericht op het invullen van een bepaalde taak en is maar goed in datgene waarvoor het ontworpen is.

Om AI in te zetten voor een bepaald probleem, wordt informatie over dat probleem aan de computer gegeven. Vervolgens verwerkt het AI-systeem deze informatie en komt er een output. Men kan daarbij het probleem kennisgebaseerd of datagebaseerd aanpakken.

Een kennisgebaseerde aanpak houdt in dat men de

AlphaGo Zero is het resultaat van *reinforcement learning*, wat uitgelegd wordt in paragraaf 6.1.

Men maakt een onderscheid tussen *general AI* en *narrow AI*. Bij *general AI* gaat het over computersystemen met dezelfde capaciteiten als een mens. We hebben vooralsnog niet de kennis om dergelijke systemen te ontwerpen. Voorlopig zit AI nog maar op het niveau van de *narrow AI*: AI-systemen die de specifieke taken doen waarvoor ze ontworpen werden.

kennis van menselijke experts zoveel mogelijk in regels probeert te gieten om de kennis van deze experts eigen te maken aan een expertsysteem.

Bij een datagebaseerde aanpak worden met statische methodes patronen in relevante data opgespoord en dan gebruikt om nieuwe problemen op te lossen. Men spreekt dan van machinaal leren (*machine learning*, ML) (Steels et al., 2017).

Intelligentie

Meer dan 2000 jaar geleden waren Socrates en Plato al geboeid door de menselijke intelligentie, maar tot op heden is er nog steeds geen algemeen gangbare definitie van intelligentie.

In de 16de eeuw definieerde dokter en psycholoog Huarte intelligentie als “het vermogen om te leren, om te kunnen inschatten en om te kunnen fantaseren”. De psycholoog Wechsler, de grondlegger van de alom gebruikte IQ-testen WAIS en WISC, formuleerde het in 1958 als volgt: “Intelligentie is de globale geschiktheid van een individu om doelbewust te handelen, om rationeel te denken en om doeltreffend om te gaan met de omgeving”.

De volgende definitie werd in 1997 onderschreven door 52 experten: “Intelligentie is een heel algemene mentale geschiktheid die o.a. het vermogen inhoudt om te redeneren, te plannen, problemen op te lossen, abstract te denken, complexe ideeën te begrijpen, snel te leren en te leren uit ervaring”.

Van Dale omschrijft het kort als “verstandelijk vermogen”. De Oxford English Dictionary ziet het als “het vermogen om kennis en vaardigheden te verwerven en toe te passen”.

Computerwetenschappers Hutter en Legg geven in 2006 de definitie: “Intelligentie meet het vermogen van een *agent* om doelen te bereiken in een breed spectrum van omgevingen”. Zo'n *agent* kan bv. een mens of een dier zijn, of een AI-systeem. MIT-professor Tegmark geeft in 2017 de volgende brede definitie: “Intelligentie is het vermogen om complexe doelen te bereiken”.

De laatste twee definities omvatten, juist door hun breedheid, veel eigenschappen: het vermogen om kennis te verwerven, om zich aan te passen, om te begrijpen . . .

(Legg & Hutter, 2007; Tegmark, 2017)

AI

De Europese Commissie uit 2018 hanteert dus de volgende definitie voor AI: "Kunstmatige intelligentie verwijst naar systemen die intelligent gedrag vertonen door hun omgeving te analyseren en – in zekere mate zelfstandig – actie te ondernemen om specifieke doelstellingen te verwezenlijken. Op KI gebaseerde systemen kunnen louter softwarematig zijn en actief zijn in de virtuele wereld (bijvoorbeeld stemgestuurde assistenten, software voor beeldanalyse, zoekmachines en systemen voor spraak en gezichtsherkenning), maar KI kan ook in hardware-apparaten worden geïntegreerd (bijvoorbeeld geavanceerde robots, zelfrijdende auto's, drones of toepassingen van het internet der dingen)".

Naast deze definitie worden nog andere definities geponeerd. Tegmark (2017) zegt eenvoudigweg dat "Artificiële intelligentie niet-biologische intelligentie is". Volgens Boden (2016) verwijst AI naar computers die de dingen doen die men met verstand doet. "Sommige van deze zaken (bv. redeneren) worden doorgaans omschreven als 'intelligent'. Andere (bv. zicht) niet. Maar allemaal vereisen ze psychologische vaardigheden die het voor een mens en een dier mogelijk maken om hun doelen te bereiken, zoals perceptie, associatie, voorspelling, planning en het controleren van een motor".

5.2 *Nieuwe technologieën*

Nieuwe technologieën veranderen de maatschappij en sommige doen dat op een zeer ingrijpende manier. Volgens Gabriëls 2019 "geven technologische ontwikkelingen mee vorm aan hoe wij de wereld zien en beïnvloedt technologie ons denken, gedrag en onze waarden en normen".

De uitvinding van het elektrisch licht heeft geleid tot de bouw van elektriciteitscentrales en heeft er uiteindelijk voor gezorgd dat de productie in fabrieken sterk werd opgedreven. Nu kan onze maatschappij niet meer functioneren zonder elektriciteit. Met de komst van de wasmachine moest minder tijd besteed worden aan het wassen van kleren, maar we zijn ze ook meer gaan wassen. Door echografie te gebruiken, gaan we anders om met een zwangerschap en eventuele complicaties

die zich (kunnen) voordoen.

Het internet en de alomtegenwoordige smartphone hebben een grote invloed op de arbeidsmarkt, ons huishouden en de vrijetijdsbesteding, m.a.w. op ons dagelijks leven. Communicatietoepassingen en sociale media hebben de dienstverlening van de overheid en de bedrijven veranderd. Online shoppen, het massaal delen van foto's, digitaal bankieren, een recept of een goedkopere energieleverancier opzoeken via het internet, een hotel boeken vanuit je luie zetel . . . Deze zaken zijn zo goed ingeburgerd dat een leven zónder haast ondenkbaar is.

Hoe ingrijpend een nieuwe technologie de maatschappij zal veranderen, is moeilijk in te schatten. De uitvinding van de telefonie en van de fotografie in de negentiende eeuw hebben bepaald hoe de hedendaagse communicatie eruit ziet. Toen de eerste auto's begonnen te rijden, had niemand verwacht dat men ooit met zijn allen uren in de file zouden staan. Wie kon in 1969 vermoeden dat we nu met een smartphone op zak zouden lopen met meer rekenkracht dan de computers die de astronauten naar de maan brachten? In 1995 was het moeilijk in te schatten dat het internet vandaag zo'n grote impact zou hebben op het dagelijkse leven.

Het is echter niet alleen moeilijk om de impact van een nieuw technologisch product in te schatten als het nog niet in gebruik is, eens wijdverspreid is het moeilijk om aanwezige ongewenste effecten te corrigeren (Collingridge dilemma) (Gabriels, 2019).

Naast de directe impact van nieuwe technologieën, is er ook een indirecte impact waarvoor men bedachtzaam moet zijn (Eynikel, 2017). Ook 'positieve' technologieën hebben soms een minder zichtbaar negatief effect.

Er zijn hiervan tig voorbeelden, zoals de industriële revolutie die een stijging van atmosferische broeikasgassen veroorzaakte, het gegeven dat een sonarboot

walvissen in de war brengt en robotmaaiers die egels al dan niet dodelijk verwonden (zie kader 'Robotmaaiers doden egels'). Dankzij de moderne communicatie is men sneller op de hoogte van wat er in de wereld gebeurt, maar de keerzijde hiervan is dat men voortdurend op zijn hoede moet zijn voor fake news.

Robotmaaiers doden egels

Robotmaaiers zijn er al sinds de jaren 90. Recent zijn er echter meldingen dat deze toestellen gevaarlijk zijn voor egels. Vooral jonge egels worden door de robotmaaiers verminkt of zelfs gedood. Fabrikanten zijn zich bewust van het probleem en raden aan om de robots niet 's nachts te laten maaien. Ze voorzien de robots ook van veiligheidsmechanismen, grotendeels bedoeld voor kinderen, volwassenen, honden en katten. Bij maaiers met een botsensor is het voor de veiligheid van de egels belangrijk dat de afstand tussen de beschermende kap en de grond niet groter is dan 4,5 cm (Rummens & Vercayie, 2012).

De impact van AI zal mogelijk nog groter zijn dan de komst van het internet.

AI is al aanwezig in het dagelijks leven: gezichtsherkenning op Facebook, Google Translate, vragen stellen aan Siri, *learning analytics* in het secundair onderwijs, in verzekeringsfirma's triëren robots de inkomende e-mails, bol.com heeft een chatbot te uwer beschikking, robotjournalistiek bij verkiezingen, suggesties op Netflix, spamfilters, betere navigatiesystemen . . .

AI is echter ook al doorgedrongen in de juridische wereld en de gezondheidszorg. AI-systemen handelen in advocatenfirma's de 'eenvoudige' zaken af of helpen de advocaat om in te schatten of het de moeite waard is om een bepaalde zaak voor de rechtbank te brengen. Radiologen doen een beroep op AI om medische beelden sneller en nauwkeuriger te interpreteren. AI wordt gebruikt voor het genereren van ondertitels bij film en om zwart-witbeelden in te kleuren. Op sommige nieuwssites schrijven robots de sportberichten. Dat

men met AI probeert om wiskundige bewijzen op te stellen, verbaast niemand, maar AI speelt ook een rol bij het opsporen van kunstvervalsingen en geeft nieuwe onderzoeksdomeinen, zoals computerlinguïstiek, een boost. Dus hoewel er jobs zullen verdwijnen door de opkomst van AI, zullen er ook nieuwe jobs door ontstaan.

Hoewel men optimistisch is over de toekomstige ontwikkelingen binnen AI, is realiteitszin op zijn plaats.

Men moet beseffen dat sommige zaken veel tijd zullen vergen, bv. vertalen tussen om het even welke talen en het begrijpen van taal. Taal is zeer complex. Denk maar aan de plaats van woorden in een zin waardoor die zin een bepaalde betekenis krijgt, woorden met meerdere betekenissen waarvan je de juiste betekenis uit de context moet afleiden . . . Voor het Nederlands is de uitdaging extra groot. Er zijn immers minder data beschikbaar omdat het Nederlandse taalgebied vrij beperkt is.

Net zoals bij andere technologische ontwikkelingen steken ook hier ethische dilemma's de kop op. Bovendien kunnen er ook moreel ongewenste effecten optreden waar men bij de ontwikkeling van het AI-systeem niet op bedacht was. Ethische discussies vergen echter veel tijd. Men mag daarom niet wachten tot de technologie klaar is om de nodige ethische discussies te voeren (zie paragraaf 5.6).

5.3 *Het gebruik van AI van A tot Z*

- Autopiloot in een vliegtuig. In een Boeing wordt slechts enkele minuten manueel bestuurd, bij het opstijgen en het landen.

In de toekomst: bij zelfrijdende auto's zullen niet één, maar miljoenen AI-systemen gebruikt worden.

- BP gebruikt AI-technologie om beter te presteren en om het gebruik van energiebronnen te optimaliseren. Ze streven naar een verhoogde veiligheid bij

de olie- en gasproductie. Ze gebruiken daarvoor o.a. geologische data en sensoren op de sites.

GE Power gebruikt big data, machine learning en IoT om een 'internet van energie' te bouwen. Ze gebruiken het ook om nodige onderhoudsactiviteiten en de vraag naar stroom te voorspellen en om de operationele werking te optimaliseren.

U leest meer over big data in paragraaf 5.4.

- Chatbots zijn steeds meer aanwezig. Ofwel als slimme persoonlijke assistenten, zoals Siri (om iets op te zoeken op internet of iets te helpen herinneren) en Alexa (om to-do-lijsten te maken, een online bestelling te plaatsen). Met stemcommando's kan men muziek afspelen, updates krijgen van nieuws- en sportberichten, een Uber bestellen en afspraken maken. Ofwel als chatbot die via Skype een klantendienst runt, zoals de virtuele assistent Cortona van Microsoft die vragen beantwoordt en info over het weer verschaft.
- Disney geeft hun bezoekers een MagicBand polsband of een badge. Hiermee krijgt Disney veel data waardoor ze kunnen anticiperen op de noden van de gasten om hen een ongelooflijke en gepersonaliseerde ervaring te bezorgen. Ze kunnen er files aan de attracties mee oplossen, extra diensten verschaffen aan klanten die ontevreden zijn door een gesloten attractie en het personeel efficiënter inschakelen.
- E-mails worden door Gmail op een slimme manier verdeeld in categorieën: primair, sociaal, reclame. De e-mails worden ook gelabeld als 'belangrijk'. Elke keer dat een e-mail als belangrijk wordt aangeduid, leert Gmail. Onderzoekers hebben de effectiviteit van de prioriteitsinbox van Gmails getest en besloten dat gebruik van de *priority inbox* de gebruikers tijd bespaarde.

Ook Spam-filters gebruiken AI. Ze moeten continu leren uit een variëteit aan signalen, zoals bepaalde woorden in een bericht, de afzender, van waar het bericht werd verzonden . . .

- Fraudebescherming: ML wordt gebruikt voor fraudepreventie bij online transacties met kredietkaarten.
- Google zet AI in voor meer duurzaamheid. Google gebruikt AI en data van satellieten om illegaal vissen te voorkomen. Iedere dag worden 22 miljoen datapunten verzameld die tonen waar schepen vertoeven wereldwijd. Met machine learning kunnen ze achterhalen waarom een schip op zee is. Zo ontwikkelde Google de 'Global Fishing Watch' dat toont waar er gevist wordt en of dat illegaal gebeurt.

Google gebruikt machine learning om de juiste configuratie van hardware en koelers in hun datacentra te bepalen. Zo vermindert Google de hoeveelheid energie om de datacentra operationeel te houden.

Google's DeepMind kent veelbelovende toepassingen in de gezondheidszorg, zoals het helpen stellen van een diagnose.

- Heineken gebruikt data om zijn operationele werking, marketing, reclamecampagnes en klantenservice te verbeteren. Maar niet alleen brouwerijen gaan op deze manier te werk, dit gebeurt ook in de auto-industrie en bij supermarkten.

Volvo gebruikt data om te voorspellen welke onderdelen kunnen stukgaan en wanneer voertuigen op onderhoud moeten komen. BMW heeft big data-technologie in zijn business-model en gebruikt data voor besluitvorming rond design, engineering, verkoop en nazorg.

Ahold Delhaize voorspelt boodschappenlijstjes van een bepaalde klant met AI. Walmart gebruikt machine learning, big data, IoT en AI om de ervaring

van klanten te verbeteren en voor een betere service. Ze gebruiken gezichtsherkenning om te bepalen of klanten blij of verdrietig zijn.

- *Image caption generation*: op sommige webshops worden de teksten bij de artikelen automatisch gegenereerd.

Het Brits nieuwsagentschap Press Association plant om robots lokale nieuwsberichten te laten schrijven.

- Juristen worden geholpen door AI-gestuurde software, die de analyse van documenten efficiënter maakt. Machines kunnen documenten bekijken en ze labelen als wel of niet relevant voor een bepaalde zaak. Als een document relevant blijkt, kunnen de machines vervolgens op zoek gaan naar andere documenten die op eenzelfde manier relevant zijn. Machines zijn veel sneller in het doorlopen van documenten. Ze kunnen de werklust van de mensen verminderen door enkel documenten door te sturen waar er twijfel over bestaat. Om hun klanten goed te kunnen adviseren, moeten advocaten vaak saai werk verrichten, zoals feiten en cijfers controleren en de uitkomsten van vroegere zaken nagaan. Een AI-systeem kan de saaie dingen doen, trouwens met meer nauwkeurigheid en efficiënter dan mensen. AI-systemen kunnen contracten nalezen. Ze kunnen ook de uitkomst van een proces beter voorspellen, aangezien ze goed data kunnen analyseren.

Er bestaat ook al AI-software om online tot een oplossing voor een echtscheiding te komen en dat voor een fractie van de prijs.

- Kredietmaatschappijen gebruiken big data van marketing databases, transacties en publieke informatie om te komen tot een snellere en meer efficiënte besluitvorming.
- Landbouw is de dag van vandaag een industrie waarbij AI-systemen beslissen of een plant bestre-

den moet worden met pesticide. Er zijn geautomatiseerde landbouwmachines om te ploegen en te zaaien; robots worden ingezet om fruit en groenten te oogsten (big data, GPS, ML).

- Mattel bracht Hello Barbie op de markt. Spraakherkenning zorgt ervoor dat het kind conversaties kan voeren met de pop. De pop, die verbonden is met servers in de cloud, leert uit vorige gesprekken.
- Netflix baseert zich op data om te bekijken in welke content ze gaan investeren. Via algoritmes wordt het geheel gepersonaliseerd, zoals de homepage, de aanbevelingen en de beelden die men te zien krijgt bij die aanbevelingen. Men baseert zich hiervoor op zaken zoals kijkgedrag en welke dag het is.
- Oude, onopgeloste politiezaken, zogenaamde cold cases, kunnen nieuw leven worden ingeblazen met een AI-systeem dat erop getraind is de dossiers te scannen op forensische sporen en een rangschikking te maken welke cold cases het meeste kans hebben om nu wel opgelost te worden dankzij moderne forensische technieken.
- Personaliseren van reclame en gebruikerservaringen gebeuren met AI. Bij Amazon krijgt de gebruiker aanbevelingen voor producten waarin hij vermoedelijk geïnteresseerd is: "Customers who viewed this item also viewed...".

Facebook gebruikt AI om de newsfeed te personaliseren en ervoor te zorgen dat de gebruiker enkel de posts en de reclame te zien krijgt die hem interesseren.

- Q-music zal in de toekomst steeds meer muziek afspelen die gecomponeerd is met de hulp van een AI-systeem. Algoritmes die muziek genereren, kunnen componisten inspireren voor nieuwe songs. AI

kan helpen begrijpen wat het publiek verlangt en helpen inschatten welke songs mogelijk een hit worden.

- Radiologen gebruiken AI in medische beeldvorming om sneller diagnoses te kunnen stellen. Er is bv. een AI-systeem dat acute en levensbedreigende pathologieën in CT-scans detecteert en aanduidt. Ook het interpreteren van de beelden en het opstellen van een behandelingsplan gebeurt steeds meer met software.
- Sociale media maken volop gebruik van AI.
 - Snapchat heeft gezichtsfilters geïntroduceerd. Deze filters traceren bewegingen van het gezicht, waardoor gebruikers geanimeerde effecten of digitale maskers kunnen toevoegen die zich aanpassen wanneer de gezichten bewegen.
 - Pinterest gebruikt ML om automatisch objecten op afbeeldingen te identificeren.
 - Instagram gebruikt ML om de contextuele betekenis van emoji's te identificeren. Dat is voor Instagram belangrijk om te kunnen analyseren hoe mensen hun website en app gebruiken. Instagram gebruikt big data en AI voor doelgerichte reclame, het tegengaan van cyberpesten, het wissen van aanvallende commentaren en de strijd tegen spam, en om de gebruikerservaring te verbeteren.
 - Twitter gebruikt AI om tweets aan te bevelen, ongepaste en racistische content tegen te gaan en de gebruikerservaring te verbeteren.
- TomTom en Waze zijn slimme navigatiesystemen. Ze gebruiken AI om hun systeem te verbeteren. Real time gegevens van gebruikers worden bv. gebruikt om reistijden beter te kunnen inschatten.
- Uber laat de kostprijs van de rit door AI berekenen. Met AI wordt ook de wachttijd geminimaliseerd en worden gebruikers optimaal gematcht om omwegen zoveel mogelijk te vermijden.

- *Voice-to-text*: een aan een smartphone gedicteerde tekst wordt omgezet in geschreven tekst. ML-systemen beginnen ook al dialecten van eenzelfde taal van elkaar te kunnen onderscheiden.
- Watson van IBM speelt sous-chef in de keuken, door te helpen recepten te ontwikkelen en unieke smaken te creëren.

Aan IBM's Watson werden honderden beelden van het werk van Gaudi gegeven, naast complementair materiaal, opdat Watson mogelijke invloeden van het werk zou leren kennen. Nadien heeft Watson inspiratie geleverd aan artiesten die de opdracht gekregen hadden om een standbeeld in de stijl van Gaudi te creëren.

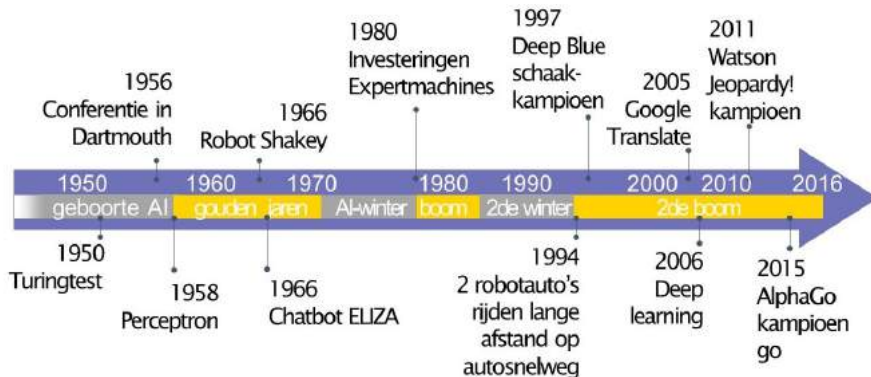
- X schreef eigenlijk die tekst! Plagiaat-checkers kunnen met ML plagiaat detecteren van bronnen die niet opgenomen zijn in een database, zoals bronnen in een vreemde taal en oudere bronnen die niet gedigitaliseerd zijn.
- YouTube gebruikt machine learning om gewelddadige en aanstootgevende video's op te sporen, vaak nog voor iemand ze heeft kunnen bekijken. Een van de redenen hiervoor is dat adverteerders niet willen dat hun advertenties getoond worden naast bv. een racistische video.
- Zullen we in de toekomst nog op de klassieke manier solliciteren? Waarschijnlijk niet. Unilever start een sollicitatieprocedure al met het verplicht indienen van een filmpje, wat dan door een AI-systeem wordt geanalyseerd. Men zegt dat het systeem er de meest gemotiveerde en geschikte kandidaten voor de job uithaalt.

(van der Beek, 2019; Carpentier De Changy, 2018; Gavira, 2018; Jacobs, 2016; Marr, 2018; Mattel, 2017; Narula, 2019; Oostra, 2019)

5.4 Geschiedenis van AI

In de negentiende eeuw ontwierp Charles Babbage de 'Analytical Machine' en publiceerde Ada Lovelace het eerste computerprogramma. Bovendien dacht Lovelace toen al na over de mogelijkheden die zo'n machine zou hebben. U kunt meer lezen over Babbage en Lovelace in de paragraaf 'Ada Lovelace'.

In 1950 vroeg Alan Turing zich af of een machine er in een conversatie zou kunnen in slagen een mens te doen geloven dat zij een mens is. Bij de Turingtest stelt een mens vragen aan een ongekende partij in een andere kamer, die zowel een mens als een machine kan zijn, om dan te besluiten of die een mens is of niet. Hiermee gaf Turing een eerste aanzet tot het onderzoeksdomein dat we nu kennen als kunstmatige intelligentie (KI of AI).



Figuur 5.1: Geschiedenis AI. Afbeelding uit presentatie Accenture, met toestemming van L. Depuydt (persoonlijke communicatie).

AI is dus geen nieuw fenomeen (zie Figuur 5.1). Het onderzoeksdomein bestaat al sinds 1956, toen op een conferentie in Dartmouth vooraanstaande onderzoekers waaronder John McCarthy, Marvin Minsky, Claude Shannon en Nathaniel Rochester (zeer optimistisch) stelden dat "elk aspect van leren of elk ander kenmerk van intelligentie in principe zo precies kan worden be-

schreven dat er een machine gemaakt kan worden om het na te bootsen”.

Ada Lovelace (1815 - 1852)

She Walks in Beauty

BY LORD BYRON (Byron, 2015)

She walks in beauty, like the night
Of cloudless climes and starry skies;
And all that's best of dark and bright
Meet in her aspect and her eyes;
Thus mellowed to that tender light
Which heaven to gaudy day denies.

One shade the more, one ray the less,
Had half impaired the nameless grace
Which waves in every raven tress,
Or softly lightens o'er her face;
Where thoughts serenely sweet express,
How pure, how dear their dwelling-place.

And on that cheek, and o'er that brow,
So soft, so calm, yet eloquent,
The smiles that win, the tints that glow,
But tell of days in goodness spent,
A mind at peace with all below,
A heart whose love is innocent!

Lord Byron (1788-1824) was niet enkel de gelauwerde auteur van dit gedicht, hij was ook de vader van Ada Lovelace. Hij noemde Ada's mama, de wiskundige Annabella Milbanke, zijn *Princess of Parallelograms*.

Toen Ada nog een baby was, gingen haar ouders echter uit elkaar. Ze heeft haar vader nadien nooit meer ontmoet. Haar moeder voedde haar op met de hulp van gouvernantes en lesgevers. Ada werd onderwezen in geschiedenis, talen, literatuur, muziek, aardrijkskunde, chemie, handwerk, steno en wiskunde. Toen ze elf jaar was, reisde ze een jaar lang door Europa met haar moeder. Ada Byron hield van wiskunde. In haar tienerjaren leerde ze de wiskundige Babbage kennen die haar enthousiasmeerde voor zijn 'Difference Engine'. Annabella Milbanke noemde het toestel de 'thinking machine' en omschreef deze machine in 1833 als volgt: "Ze kan de tweede en de derde macht van getallen en de wortels van een vierkantsvergelijking berekenen".

Ada huwde in 1835 met William, Lord King. In 1838 kreeg William de titel van

graaf omwille van zijn verdienstelijk werk voor de overheid, en Ada werd de gravin van Lovelace.



ADA LOVELACE (Carl H. Pforzheimer Collection of Shelley and His Circle. The New York Public Library, 1835).



CHARLES BABBAGE DIFFERENCE ENGINE (Science Museum Group Collection, 1833).

Babbage had eerder logaritmetabellen opgesteld, volledig uitgerekend met de hand. Zijn tabel uit 1827 zou bijna 100 jaar gebruikt worden. Hij zocht erop om dit rekenwerk te kunnen automatiseren, wat leidde tot de bouw van de Difference Engine.

Later zou Babbage de 'Analytical Machine' ontwerpen, die gezien wordt als de eerste computer. Dit toestel kon immers een sequentie van operaties uitvoeren en 'geprogrammeerd' worden voor allerlei bewerkingen. Het toestel zou aangestuurd worden met ponskaarten, zoals bij de weeftoestellen (de jacquardmachines). Babbage zou het toestel echter nooit echt fysiek realiseren. Hij oogstte veel bijval met zijn Analytical Machine in Italië en in 1842 publiceerde een Italiaanse wiskundige, Luigi Menabrea, er een paper over. Ada, ondertussen al enkele jaren gehuwd en moeder geworden, vertaalde deze paper van het Frans naar het Engels. Ze voorzag het document bovendien van uitgebreide commentaar en vulde het aan met zorgvuldig uitgewerkte redeneringen. De vertaling van Ada mét haar eigen aanvullingen erbij werd gepubliceerd. Zo kwam het dat Ada Lovelace de eerste was die een rigoureuus uitgeschreven computerprogramma publiceerde; het programma berekende Bernoulligetallen. Ze gebruikte daarbij al concepten uit de computerwetenschappen, zoals variabelen, lussen en geneste lussen. Ze wordt dan ook gezien als de eerste

computerprogrammeur. Ze vermeldde dat zo'n machine nooit origineel uit de hoek zou komen. Ze zou enkel uitvoeren wat haar was opgedragen. Maar Ada beseftte dat zo'n Analytical Machine veel meer zou kunnen dan alleen klassieke wiskundige bewerkingen. Ze zei dat, als men erin zou slagen om verhoudingen tussen muzikale tonen op een abstracte manier weer te geven, de machine dan in staat zou zijn muziek te maken.

Ada Lovelace overleed al op jonge leeftijd in 1852, waarschijnlijk aan kanker. Haar vriendin Florence Nightingale zei dat ze waarschijnlijk nooit zo lang had geleefd ware het niet dat ze zo'n ontzettend vitaal brein had (Wolfram, 2015).

In 1958 ontwierp Frank Rosenblatt het Perceptron, een *neuraal netwerk* waarbij inputs direct met outputs zijn verbonden. De New York Times berichtte hierover met veel sensatie hoewel er bewezen was dat het Perceptron slechts lineair scheidbare patronen kon herkennen, en geen complexere types zoals de *XOR*-functie (zie kader '*XOR*-functie of exclusieve 'of').

De robot Shakey uit 1966 werd gecreëerd aan het Artificial Intelligence Center van het Stanford Research Institute en was de eerste robot die instructies kon analyseren en opbreken in deelinstructions. Het project combineerde *robotics*, *computer vision* en *natural language processing*. Voorbeeld: "Duw de doos van het platform". Shakey kijkt rond, identificeert een platform met een doos erop, lokaliseert een ramp om het platform op te kunnen rijden. Shakey duwt dan de ramp tot aan het platform, rijdt op de ramp tot op het platform en duwt de doos ervan (Wikipedia, 2019).

Op het Massachusetts Institute of Technology (MIT) ontwierp Weizenbaum de software ELIZA waarmee een Rogeriaanse psychotherapeut bij een intakegesprek van een nieuwe patiënt werd nagespeeld. Zulke therapeuten nemen een passieve rol op en bouwen hun vragen aan de patiënt zo op dat ze de antwoorden van de patiënt erin verwerken. Als dat voor het systeem niet mogelijk was, kon ELIZA ook nog putten uit voor-

U leert meer over het Perceptron in paragraaf 11.4.



Figuur 5.2: Robot Shakey (Nardone, 2007).

ziene standaardvragen (zie Figuur 5.3). ELIZA kan beschouwd worden als de eerste chatbot (Güzeldere & Franchi, 1995).

```

Welcome to
          EEEEE LL   IIII ZZZZZZ  AAAAA
EE      LL   II   ZZ  AA  AA
EEEE   LL   II   ZZZ  AAAAAA
EE     LL   II   ZZ  AA  AA
EEEEEE LLLLLL IIII ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU:   Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU:   They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU:   Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU:   He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU:   It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU:   █

```

Figuur 5.3: Conversatie met Eliza (Eliza, 2018).

Tussen 1974 en 1980 kende het onderzoeksdomein een 'AI-winter' waarbij de ontwikkelingen in AI een terugval kenden. De overheid en investeerders hadden veel kritiek op het onderzoek, maar desondanks boekten de onderzoekers toch vooruitgang.

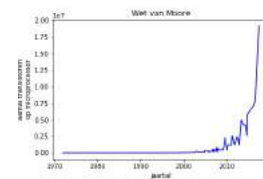
In 1973 verscheen een rapport waarin werd gesteld dat over het hele AI-onderzoeksdomein de ontdekkingen en de grote impact, die was beloofd, uitgebleven waren. In die tijd speelde het gebrek aan rekenkracht de onderzoekers parten. Met de toen beschikbare hardware was het onmogelijk om een AI-oplossing voor kleine problemen uit te breiden naar bruikbare real-life toepassingen (zie ook kader 'P versus NP'). Het besef groeide ook dat het relatief eenvoudig is om computers te leren dammen of pi te laten berekenen tot op een miljoen decimalen, maar zeer moeilijk om computers te leren stappen of spreken.

In de jaren 80 werden wel weer investeringen gedaan. Expertsystemen die de besluitvorming van menselijke experts om zeer specifieke problemen op te

lossen simuleerden, zoals het diagnosticeren van besmettelijke ziektes of het identificeren van chemische componenten, waren toen zeer populair.

Maar de desktopcomputers van Apple en IBM werden steeds sneller en krachtiger (volgens de wet van Moore, zie Figuur 5.4), waardoor de gebruikers de resultaten van de expertsystemen uiteindelijk veel goedkoper konden neerzetten. De markt van de gespecialiseerde AI-hardware stortte in en er was sprake van een tweede AI-winter tussen 1987 en 1993 (Lim, 2018).

De wet van Moore stelt dat elke twee jaar het aantal transistoren op een microchip verdubbelt. Daardoor ontstaat steeds meer rekenkracht.



Figuur 5.4: Het aantal transistoren op een microcontroller. Data van Rupp (Rupp, 2019).

P versus NP

Stel dat een computer een berekening uitvoert op een input van n getallen. Het aantal stappen dat de computer nodig heeft om die berekening uit te voeren, zal afhangen van n : hoe groter n , hoe meer stappen. Dat aantal stappen kan bv. als veelvoud van een macht van n worden uitgedrukt of als veelvoud van een macht van een constante met n in de exponent. In het eerste geval verloopt de berekening vrij snel, men spreekt dan van polynomiale tijd. In het tweede geval duurt het heel wat langer, men spreekt dan van exponentiële tijd; bij te grote n is de berekening dan zelfs niet meer praktisch haalbaar.

Stel dat een algoritme dat een tijd nodig heeft evenredig met n , een seconde nodig heeft om een bewerking met 100 elementen uit te voeren. Dan heeft een algoritme dat een tijd nodig heeft die evenredig is met n^3 , $0.01 \text{ s} \cdot 100^3 = 10\,000 \text{ s}$ nodig, wat ongeveer gelijk is aan drie keer 3600 s , of dus drie uur. Maar bij een algoritme dat een tijd nodig heeft die evenredig is met 2^n , komt men er bij lange al niet meer met een miljoen jaar.

Sommige problemen zijn op meerdere manieren oplosbaar. Als men de keuze heeft tussen een polynomiale tijd algoritme en een exponentiële tijd algoritme, dan kiest men voor het eerste. Er zijn problemen waarvoor men een exponentiële tijd algoritme heeft, maar (nog) niet beschikt over een polynomiale tijd

algoritme. De vraag is nog maar of er wel een polynomiale tijd algoritme is voor zo'n probleem.

Problemen die opgelost kunnen worden met een polynomiale tijd algoritme, behoren tot de klasse P. Bv. het opsporen van het grootste getal in een niet-gesorteerde lijst van getallen, waarbij het algoritme de hele lijst afgaat en het tot dan toe grootste getal onthoudt. Het algoritme moet dan elk getal slechts één keer bekijken en het aantal stappen dat het algoritme nodig heeft, is dus een veelvoud van n .

Problemen van de klasse NP zijn moeilijke problemen die men tot nu toe enkel in exponentiële tijd kan oplossen, maar waarvan de juistheid van een antwoord in polynomiale tijd geverifieerd kan worden. Bv. het ontbinden in priemfactoren van grote getallen vergt wel wat tijd, maar eens je de priemfactoren kent, is het gemakkelijk na te gaan of ze correct zijn. De verzameling P is een deelverzameling van NP. Wat men zich afvraagt, is of ook omgekeerd alle NP-problemen tot de klasse P zouden kunnen behoren, m.a.w. is $P = NP$? De vraag is dus: kunnen problemen waarvan de oplossing in polynomiale tijd geverifieerd kan worden, ook in polynomiale tijd worden opgelost? Niemand heeft tot nu toe kunnen aantonen dat NP-problemen echt op geen enkele manier met een computer kunnen worden opgelost, omdat ze een onmogelijk lange tijd nodig hebben om het probleem op te lossen bij om het even welke procedure.

Deze vraag is een van de millenniumproblemen van het Clay Mathematics Institute. De persoon die het antwoord op deze vraag bewijst, krijgt een miljoen dollar.

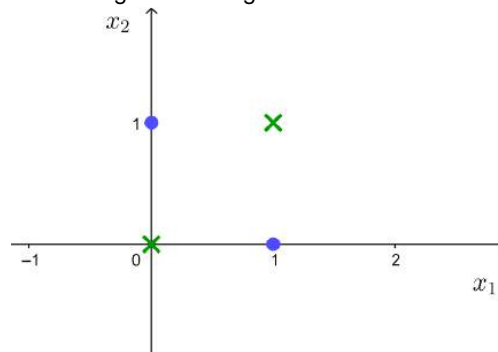
(Clay Mathematics Institute, 2019; Hardesty, 2009; van den Brandhof, 2009)

XOR-functie of exclusieve 'of'

De XOR-functie is een booleaanse operator die inwerkt op twee binaire waarden. Als juist één van deze binaire waarden 1 is, dan is het resultaat van de XOR-functie 1, anders is het resultaat 0.

$$\text{XOR} : \begin{cases} (0,0) \mapsto 0 \\ (0,1) \mapsto 1 \\ (1,0) \mapsto 1 \\ (1,1) \mapsto 0 \end{cases}$$

XOR wordt grafisch voorgesteld in Figuur 5.5.



Figuur 5.5: XOR-functie.

De punten afgebeeld op de grafiek zijn niet lineair scheidbaar. Er is dus geen rechte te vinden die de punten van elkaar scheidt. De punten kunnen toch van elkaar gescheiden worden met andere technieken uit machinaal leren. Dat wordt behandeld in paragraaf 11.4.

Sinds midden jaren 90 is er een opleving. In 1994 reden voor het eerst twee zelfrijdende auto's op de autosnelweg in de buurt van Parijs, te midden van het drukke verkeer. Dickmanns en zijn team hadden twee Mercedesen voorzien van camera's en het sturen, versnellen en remmen werden aangestuurd door het *computer vision* systeem (Kröger, 2016). In 1997 versloeg IBM Deep Blue de heersende schaakkampioen. In 2006 werd een online versie van Google Translate gelanceerd. Er kon

vertaald worden van het Engels naar het Arabisch en van het Arabisch naar het Engels. In 2011 werd IBM's Watson kampioen in de quiz Jeopardy!. Naast kennis was hierbij ook *natural language processing* belangrijk. In 2016 won Google DeepMind's AlphaGo zelfs van de wereldkampioen in go (zie Figuur 5.1).

Big data

In een van de Standpunten van de Koninklijke Vlaamse Academie van België voor Wetenschappen en Kunsten (Berbers et al., 2017) vindt men de volgende omschrijving van big data:

“Big data is niet hetzelfde als veel data. Bij big data gaat het om grote hoeveelheden gestructureerde en ongestructureerde data. Data in allerlei formaten zoals tekst, afbeeldingen en geluid en vanuit allerlei bronnen zoals e-mails, video's, brieven, rapporten, blogs, postings, cijfers, archieven, sensoren, camera's, enz. Het kunnen ook persoonsgegevens zijn. Data kunnen vrijwillig aangeleverde data zijn (bv. ingevulde onlineformulieren), geobserveerde data (bv. door software of sensoren uitgelezen gedragsgegevens) of daaruit afgeleide gegevens (bv. een profiel voor de kredietwaardigheid). De term big data verwijst steeds naar machinaal leesbare digitale informatie die door computersystemen kan worden verwerkt en direct is verbonden met technieken die het doorzoeken en analyseren mogelijk maken van grote hoeveelheden data die niet noodzakelijk op voorhand reeds zijn gesorteerd. Het analyseren van de data omvat de toepassing van patroonherkenning, machinale leeralgoritmen, voorspellingsalgoritmen, statistische methodes, neurale netwerken ...“

Big data houdt dus in dat AI-systemen bv. data van verschillende bronnen met elkaar in verband kunnen brengen, namelijk grote hoeveelheden data, die digitaal beschikbaar zijn, zodat computersystemen ze kunnen doorzoeken, analyseren en verwerken, data van allerlei oorsprong en formaat.

De laatste jaren kon men dankzij snellere hardware en zeer grote datasets veel vooruitgang boeken in de ML.

Door de ontwikkelingen in de game industrie (zie Figuur 5.6) en in de dataopslag, maar vooral door de

mogelijkheden die het internet biedt om zeer grote datasets te verzamelen en te delen (zie kader 'Big Data'), is ML prominent aanwezig sedert het begin van de jaren 2010.

Andrew Ng, professor in AI aan de universiteit van Stanford, zegt optimistisch: "AI is de nieuwe elektriciteit". Volgens hem zal vooruitgang op het gebied van hardware ervoor zorgen dat nieuwe AI-technieken zullen kunnen worden uitgevoerd. Hij meent dat we de komende jaren over meer rekenkracht en snellere netwerken zullen kunnen beschikken.

Sommigen veronderstellen dat de snelle ontwikkelingen in AI zullen leiden tot het ontstaan van kunstmatige intelligente systemen die zichzelf zullen verbeteren zonder menselijke tussenkomst. Ze zullen zó intelligent worden dat ze de maatschappij kunnen sturen en zullen overnemen. De creatie van zo'n artificiële superintelligentie is gekend als de 'technologische singulariteit'. De technologische singulariteit is een hypothese. Sommigen denken dat het nooit zal gebeuren, maar anderen verwachten dat deze singulariteit al in deze eeuw zal worden bereikt. Men weet niet of er ooit een superintelligent AI-systeem zal worden gecreëerd, maar men kan het ook niet met zekerheid uitsluiten. Daarom vinden sommige AI-experts dat we maar beter voorbereid kunnen zijn. Zij vinden dat we al moeten nadenken over hoe we er kunnen voor zorgen dat een superintelligent AI-systeem doelen heeft die overeenstemmen met de onze. Andere experts zijn ervan overtuigd dat het zeker nog niet voor binnenkort is en stellen andere prioriteiten. Max Tegmark, professor aan het Massachusetts Institute of Technology (MIT), vindt dat men niet mag wachten met de discussies, alleen al gezien de verwachte impact van AI de komende decennia (Tegmark, 2017).



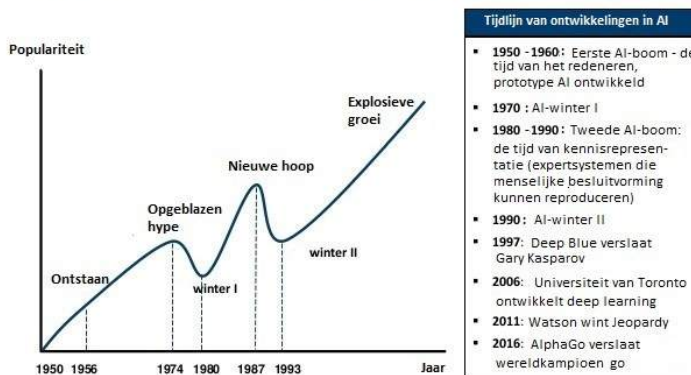
Figuur 5.6: Voor meer rekenkracht werd binnen de game industrie de GPU ontwikkeld. Hierdoor kunnen games worden ontwikkeld met realistisch ogende graphics zoals in Shadow of the Tomb Raider © Square Enix (2018).

Geschiedenis van AI-winters

door Milton LIM (Lim, 2018)

Artificiële intelligentie (AI) is momenteel een zeer populair topic in de media, maar het concept AI werd al bijna 70 jaar geleden voor het eerst voorgesteld. Wat is er al die jaren gebeurd in het AI-onderzoek en waarom heeft een doorbraak zo lang op zich laten wachten? Milton Lim bekijkt de ups en downs van de AI-industrie over de jaren heen.

AI KENT EEN LANGE GESCHIEDENIS ALS "THE NEXT BIG THING"...



DE UPS EN DOWNS CYCLUS VAN AI-ONDERZOEK.

“AI zal ofwel het beste, ofwel het slechtste zijn, dat de mensheid ooit is overkomen.” – Stephen Hawking

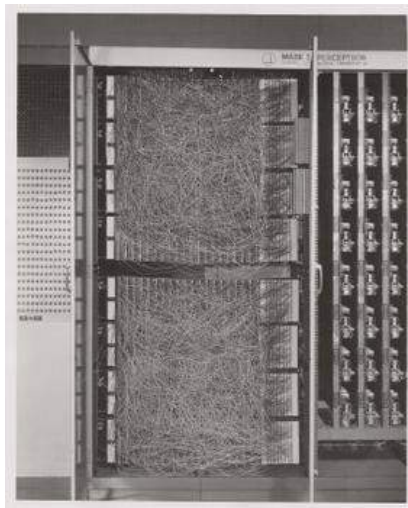
Het begin

In 1950 stelde Alan Turing de vraag “Kunnen machines denken?”. [i] Aangezien het concept ‘denken’ moeilijk te definiëren is, stelde hij een eenvoudigere vraag, namelijk of een machine een mens zou kunnen imiteren tijdens een gesprek met een andere mens. Dit principe is gekend als de Turingtest, wat inhoudt dat een mens vragen stelt aan een ongekende partij in een andere kamer, die zowel een mens als een machine kan zijn, om dan te besluiten of die een mens is of niet. De machine slaagt in de test als ze de mens kan doen geloven dat zij een mens is. De Turingtest was significant omdat hij voor het eerst in een

criterium voorzag om de vraag “Kunnen machines doen wat wij mensen (als denkende entiteiten) doen?” te beantwoorden.

“Elk aspect van leren of elk ander kenmerk van intelligentie kan in principe zo precies worden beschreven dat er een machine gemaakt kan worden om het na te bootsen.”

Het AI-onderzoeksdomein is officieel gelanceerd in 1956 op de Dartmouth Conferentie waar de term 'artificiële intelligentie' werd geïntroduceerd als verzamelnaam voor een variatie aan onderzoek in cybernetica, automatentheorie en complexe informatieverwerking met als doel aan machines de mogelijkheid te geven om te 'denken'. Een kleine groep van vooraanstaande onderzoekers waaronder John McCarthy, Marvin Minsky, Claude Shannon en Norbert Wiener stelden dat “elk aspect van leren of elk ander kenmerk van intelligentie in principe zo precies kan worden beschreven dat er een machine gemaakt kan worden om het na te bootsen.” Dit gaf een duidelijke pragmatische richting aan voor verder onderzoek binnen AI.



DE EERSTE IMPLEMENTATIE VAN HET PERCEPTRON IN DE MARK I
PERCEPTRON MACHINE DIE AFBEELDINGEN KON HERKENNEN MET EEN 20 x 20
PIXEL CAMERA.

In 1958 stelde Frank Rosenblatt het Perceptron-lerend-algoritme op, het meest eenvoudige type van neuraal netwerk met slechts een laag neuronen waarbij inputs met outputs zijn verbonden. De New York Times berichtte met veel sensatie dat het Perceptron “het embryo van een elektronische computer was, waarvan het leger verwachtte dat het in staat zou zijn om te lopen, te spreken, te zien, te schrijven, zichzelf te reproduceren en bewust zou zijn van zijn bestaan”. Er was nochtans bewezen dat het Perceptron met één laag slechts lineair scheidbare patronen kon herkennen, en geen complexere types zoals de XOR-functie (exclusieve OR-functie).[i] Het domein van neurale netwerken stagneerde gedurende een tiental jaren, totdat men zich realiseerde dat Perceptrons met meerdere lagen, met voldoende rekenkracht en data, een zeer efficiënte manier waren om meer complexe niet-lineaire functies te modelleren. Dit is het principe achter hedendaags onderzoek in *deep learning* met veel complexe lagen van neuronen, zoals het diep neuraal netwerken dat gebruikt werd door Google DeepMind’s AlphaGo om Lee Sedol te verslaan in het spel go.

Vorbije AI-winters

Het AI-onderzoek heeft een hobbelig parcours doorstaan met twee investeringsarme periodes, gekend als ‘AI-winters’, tussen 1974 en 1980 en tussen 1987 en 1993. Alhoewel het onderzoeksdomein leed onder de waardeperceptie van AI door de overheid en investeerders, boekten de onderzoekers vooruitgang ondanks de kritiek.

In 1973 verscheen het Lighthill Report [ii] in opdracht van de UK Science Research Council, waarin het ultieme falen van AI in het behalen van zijn ‘grandioze objectieven’ werd bekritiseerd en waarin werd gesteld dat er ‘in geen enkel deel van het domein reeds ontdekkingen waren gedaan die hebben geleid tot de grote impact die toen was beloofd.’ Ongeveer tegelijkertijd bewees Richard Karp roemrijk dat 21 moeilijke problemen in computerwetenschappen NP-volledig [iii] zijn, wat leidde tot het bekende onopgeloste P vs NP-probleem waar een beloning van \$US 1 miljoen aan vasthangt. [iv] Dit beklemtoonde het probleem van ‘combinatorische explosie’, waarbij de rekentijd nodig om het probleem op te lossen exponentieel steeg in functie van de input-grootte. Dit betekende dat het met de beschikbare hardware onmogelijk was om eender welke AI-oplossing voor kleine problemen uit te breiden naar bruikbare real-life

toepassingen. De vloek van dimensionaliteit had een slag uitgedeeld aan het AI-onderzoek.

De moeilijke problemen zijn gemakkelijk en de gemakkelijke problemen zijn moeilijk.

Een andere fundamentele grens, gekend als Moravecs Paradox, stelt dat “het relatief eenvoudig is om computers te laten presteren op het niveau van een volwassene op het gebied van intelligentietesten, een damspel of het berekenen van pi tot op een miljard cijfers, maar moeilijk of onmogelijk om computers de vaardigheden te geven van een eenjarige wanneer het gaat over waarnemingsvermogen en motoriek . . . De mentale mogelijkheden van een kind die we als vanzelfsprekend beschouwen – een gezicht herkennen, een potlood opheffen of door een kamer wandelen – zijn eigenlijk oplossingen voor sommige van de moeilijkste ingenieursproblemen ooit bedacht . . . Miljoenen jaren ervaring met de aard van de wereld en hoe erin te overleven zijn opgeslagen in de grote, sterk geëvolueerde zintuiglijke en motorische delen van het menselijk brein”. [v] Kortom, de moeilijke problemen zijn gemakkelijk en de gemakkelijke problemen zijn moeilijk. Dit verklaart waarom onderzoek naar *computer vision* en robotica zo weinig vooruitgang heeft geboekt in de jaren 70.

De volgende tien jaar stegen de investeringen door bedrijven in de AI-industrie van een paar miljoen dollar in 1980 tot miljarden dollars in 1988. Expertsyste­men zoals XCON, LISP-machines en Symbolics werden zeer populair als gespecialiseerde systemen die de besluitvorming van menselijke experts om zeer specifieke problemen op te lossen simuleerden, zoals het diagnostice­ren van besmettelijke ziektes of het identificeren van chemische componenten. Ondertussen werden desktopcomputers van Apple en IBM gestaag sneller en krachtiger (volgens de wet van Moore) totdat ze de duurdere LISP-machines overklasten. Het kwam er uiteindelijk op neer dat deze expertsystemen te duur waren om te handhaven aangezien ze moeilijk te updaten waren, niet konden leren, en onbetrouwbaar waren in het afhandelen van ongebruikelijke inputs. De gebruikers moesten geen dure machine meer kopen die gespecialiseerd was in het draaien van LISP, waardoor de markt van gespecialiseerde AI-hardware in 1987 instortte. In een jaar tijd werd een volledige industrie van een half miljard dollar vervangen.

De toekomst

“AI is de nieuwe elektriciteit”. – Andrew Ng

Andrew Ng, professor in AI aan de universiteit van Stanford, omschreef het optimistisch: “AI is de nieuwe elektriciteit”. Zijn persoonlijke visie is dat “voortgang op het gebied van hardware de brandstof zal voorzien die nodig is om nieuwe AI-technieken uitvoerbaar te maken. Meerdere hardware verkopers zijn zo vriendelijk geweest om hun plannen te delen en ik ben vol vertrouwen dat ze geloofwaardig zijn en dat we de komende jaren over meer rekenkracht en snellere netwerken zullen kunnen beschikken”. [vi]

Misschien zal de huidige AI-boom op een dag de hypothetische ‘Technologische Singulariteit’ bereiken, waar de exponentiële toename in rekenkracht zal uitmonden in de creatie van een artificiële superintelligentie die het totaal aan menselijke intelligentie zal overstijgen. Dit zou een op hol geslagen kettingreactie van zelfverbeterende cycli veroorzaken waarbij de machine-intelligentie zichzelf kan vooruithelpen zonder gebruik te moeten maken van een menselijke inspanning. Hierover wordt betoogd dat dit resulteert in een explosie van intelligentie die van de aarde weg zal stralen totdat ze het volledige universum zal hebben verzadigd. Sommige auteurs hebben voorspeld dat deze Singulariteit al in 2045 zal worden bereikt. [vii]

Dit doet een aantal diep filosofische en ethische vragen [viii] rijzen die al in ontelbare boeken en films onderzocht werden:

- Kan een machine emoties hebben?
- Kan een machine zelfbewust zijn?
- Kan een machine origineel of creatief zijn?
- Kan een machine welwillend of vijandig zijn?
- Kan een machine een ziel hebben?

Zulke ethische beschouwingen moeten simultaan ontstaan met de ontwikkelingen in de technologie die nieuwe AI-modellen mogelijk maken. Voor een doordachte visie op hoe AI en machine learning de maatschappij zullen veranderen, verwijst ik naar het rapport “Machine Learning: the power and promise of

computers that learn by example” van de UK Royal Society uit 2017.

In het volgende artikel in deze reeks, zullen we de 'vloek van de dimensionaliteit' in 'machine learning' en statistiek onderzoeken om de voornaamste obstakels te kunnen begrijpen die het succes van AI-technologie bemoeilijken.

[i] Turing, Alan (1950), “Computing Machinery and Intelligence”, *Mind*

[ii] Minsky & Papert (1969), “Perceptrons”

[iii] Lighthill, J. (1973), “Artificial intelligence: a general survey”, *Artificial intelligence: a paper symposium*

[iv] Karp, Richard (1972), “Reducibility Among Combinatorial Problems”, *Complexity of Computer Computations*

[v] Clay Mathematics Institute <http://www.claymath.org/sites/default/files/pvsnp.pdf>

[vi] https://en.wikipedia.org/wiki/Moravec's_paradox

[vii] <https://www.technologyreview.com/s/603062/ai-winter-isnt-coming>

[viii] Kurzweil, Raymond (2005), “The Singularity Is Near: When Humans Transcend Biology”, Viking

[ix] https://en.m.wikipedia.org/wiki/Philosophy_of_artificial_intelligence

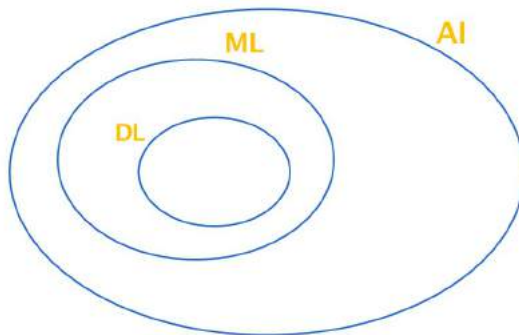
5.5 Soorten AI

Uit het voorgaande blijkt dat er meerdere soorten AI bestaan. In paragraaf 5.1 werd reeds het verschil tussen een kennisgebaseerde en een datagebaseerde aanpak aangehaald.

Kennisgebaseerde systemen kwamen veelvuldig voor in de voorbeelden van paragraaf 5.4. Robot Shakey, ELIZA en Deep Blue zijn voorbeelden van zulke systemen. Deepmind's AlphaGo, Google Translate zijn daarentegen *machine learning* systemen, alsook de voorbeelden uit paragraaf 5.3.

Binnen het machinaal leren bekleedt *deep learning* (DL) een prominente plaats (zie Figuur 5.7). DL heeft een revolutie veroorzaakt in het veld van ML door opmerkelijke resultaten te boeken op het vlak van spraak-

en beeldherkenning, wat vaardigheden zijn waarvan de computer de menselijke prestaties tot op vandaag moeilijk kan evenaren.



Figuur 5.7: AI, ML en DL.

In hoofdstuk 6 wordt dieper ingegaan op *machine learning* en *deep learning*.

5.6 Ethiek

Als men de AI-technologie goed inzet, kan die bijdragen tot een betere wereld. Denk maar aan exoskeletons die mensen met een beperking meer mobiel maken, ouderen die langer thuis kunnen blijven wonen, een meer efficiënte voedselproductie, zuinigere elektrische toestellen, veiliger werk door robots in te zetten bij de ontmijningsdienst.

Maar zal AI ten allen tijde worden ingezet voor een mooiere maatschappij? Bij de ontwikkeling van AI-systemen kan men zich door verschillende motieven laten leiden. Zal men bij de ontwikkeling van een nieuw product steeds de mens centraal zetten, of soms ook het AI-systeem zelf of geldbejag?

Door robots in te zetten in de zorgsector, krijgen verplegers meer tijd voor de patiënten. Als men er echter voor kiest om dan maar minder personeel aan te werven, dan verliest men de mens uit het oog. Het inzetten van technologie in de geneeskunde kan een positief iets zijn, maar hoe ver wil men daarin gaan? Welk soort cyborg is toelaatbaar? Is een virtuele therapeut zoals Woebot wenselijk? Welke taken krijgen zorgrobots? Hoe bekomt men dat exoskeletten voor iedereen toegankelijk zijn en niet enkel voor mensen met voldoende geld (Gabriels, 2019)?

Onder het mom van het bestrijden van de criminaliteit, kan men AI-systemen inzetten om mensen overal gade te slaan. De privacy van mensen komt dan wel in het gedrang. Een routeplanner doet ook suggesties voor restaurants in de buurt van het traject. Hoe wordt er bepaald welke restaurants er worden getoond en welke motieven spelen daarin mee? Dankzij AI kan men op sommige platforms interessante vacatures in de mailbox krijgen. Maar krijgt men wel alle vacatures te zien waarin men geïnteresseerd is?

Tot 2018 gebruikte Amazon een sterk bejubeld maar ondertussen afgevoerd AI-systeem om sollicitanten te beoordelen. Het systeem selecteerde geen vrouwen voor technologische posities. Het was immers getraind met historische data: sollicitanten van de voorbije 10 jaar, voornamelijk mannen aangezien zij nog steeds de technologiewereld domineren. Google's stemherkenning werkt beter op mannenstemmen omdat de database van stemopnames om het systeem te trainen veel meer mannenstemmen bevatte. Tijdens een schoonheidswedstrijd in 2016, gejureerd door een AI-systeem, werden deelnemers met een donkere huidskleur benadeeld, omdat het algoritme getraind was met foto's van vooral blanken.

In tegenstelling tot wat men doorgaans denkt, neemt een AI-systeem geen 100 % 'objectieve' beslissingen.

Een AI-systeem is 'vooringenomen', meestal te wijten aan de data waarmee het systeem werd getraind. Deze *bias* moet men zoveel mogelijk controleren en tot het minimum herleiden (zie kader 'Bias'). Ontwikkelaars van AI-systemen moeten reeds bij de ontwikkeling bedacht zijn op mogelijke ongewenste effecten. De morele verantwoordelijkheid van een AI-systeem dat in de maatschappij wordt gebracht, ligt niet bij het AI-systeem, maar bij de mens.

Bias

Bias komt voor als de data niet representatief zijn. Als men bv. enkel groene appels als voorbeelden geeft aan een DL-systeem, dan zal het model geen rode appels herkennen. Of als men bij de training enkel foto's van honden onder een stralend blauwe hemel aanbiedt, dan zal het DL-model een hond in de regen niet bij de klasse 'hond' indelen.

Als de gebruikte data gekleurd zijn door een aanwezige bias in de maatschappij, zoals stereotypen, dan zal dit ook doorgegeven worden aan het ML-systeem. Men moet er dus over waken dat het model daardoor niet discriminerend wordt voor bepaalde bevolkingsgroepen. Als men bv. enkel vrouwelijke verplegers in de dataset stopt, dan zullen mannen niet als verpleger worden geclassificeerd. Een model wordt nochtans getest voor het in gebruik genomen wordt. De testdata kunnen echter dezelfde bias bevatten als de trainingdata.

Het KIKS-model bevat ook een bias. De foto's van de trainingset zijn zo gemaakt dat de huidmondjes erop ongeveer passen in een vak van 120 op 120 pixels. Deze voorbeelden zijn afdrukken van bladeren genomen met transparante nagellak. Dat heeft als gevolg dat het model het best zal presteren op afbeeldingen van even grote huidmondjes in een gelijkaardige kleur.

Het nemen van beslissingen wordt steeds meer geautomatiseerd. Algoritmen bepalen welk nieuws je te zien krijgt op Facebook en welke filmpjes op YouTube worden geweigerd, maar ook de banken en de overheid maken er gebruik van. Daarom klinkt de vraag naar transparantie steeds luider. Men moet er zich echter

ook bewust van zijn dat transparantie over het gebruikte algoritme nog niet betekent dat het ook betrouwbaar is.

Men kan een AI-systeem niet moreel verantwoordelijk stellen voor zijn handelingen (Eynikel, 2017). Een mens neemt morele beslissingen door zorgvuldig af te wegen wat hij belangrijk vindt, rekening houdend met zijn normen en waarden. Hij zal daarbij ook beïnvloed worden door menselijke relaties, empathie. Zijn normen en waarden zijn deels bepaald door de cultuur waarin hij opgroeide of leeft. Dat betekent dat wat bij ons moreel aanvaard is, het daarom niet is in Japan en vice versa. Als men wil dat een slimme robot zo handelt dat een mens die handeling moreel verantwoord vindt, dan zal de mens daarvoor moeten zorgen. Dat betekent bv. dat hoe meer autonomie men toekent aan een AI-systeem, bv. een zelfrijdende auto, hoe groter de morele verantwoordelijkheid wordt van de mens, hetzij als ontwikkelaar of als producent, hetzij als wetgever, hetzij als gebruiker.

Nu al moet er nagedacht worden over de mogelijke impact die toekomstige ontwikkelingen in technologie op mens en maatschappij zullen hebben. Men moet nadenken over hoe men de systemen betrouwbaar kan maken en hoe men de wetten moet aanpassen. Men kan niet wachten met een wetgeving rond zelfrijdende auto's tot de auto's massaal de weg opgaan. Ethische kwesties vergen veel tijd. Ze zijn ook vaak complex. Er is in elk geval al aandacht voor: *ethical AI*, *responsible AI*, *AI4good*, *trustworthy AI*, *explainable AI* zijn termen die her en der opduiken, met de bijbehorende richtlijnen.

Voorbeelden kunt u lezen in de kader 'Mogelijk scenario?', in de kader 'NIPT' en in de kader 'Nu in China'.

Mogelijk scenario?

In de blogpost 'What if Waze were evil?' vraagt Geert Martens zich af of het navigatiesysteem Waze geen geheime VIP-gebruikers heeft. Zou het kunnen dat een selecte groep mensen aan Waze duizenden euro's per maand betaalt om steeds fileloos op zijn bestemming te geraken? Voor Waze zou het immers een fluitje van een cent zijn om andere autobestuurders even om te leiden en van de VIP-route af te halen. De vraag is nog maar of de Waze-klant het erg zouden vinden als het zo was. Waze is immers gratis. En eens in de zoveel keer wat tijd verliezen, waar men zich trouwens toch niet bewust van zou zijn, zou men misschien best wel tolereren in ruil voor een VIP-status voor de hulpdiensten (Martens, 2019).

NIPT

Als een erfelijke aandoening voorkomt in een familie, doet men soms genetische tests bij de familieleden. De mensen bij wie de test wordt afgenomen, worden dan geïnformeerd over wat bij de test aan het licht kan komen en hoe eventuele problemen dan kunnen worden aangepakt. Het is immers belangrijk dat mensen goed geïnformeerd kunnen beslissen of ze het genetisch onderzoek wel willen ondergaan.

NIPT staat voor niet-invasieve prenatale test. De NIPT wordt aangeboden als een test om te zien of een ongeborn kind aan het syndroom van Down of een andere chromosomale afwijking lijdt. In vergelijking met genetische tests in het kader van erfelijke aandoeningen is de informatie die ouders voorafgaand aan de NIPT krijgen, vaak veeleer beperkt. De ouders beseffen daardoor niet altijd waarvoor ze kiezen, sommigen staan er niet bij stil dat er na de test ook slecht nieuws kan volgen.

Omdat de test wordt gedaan via het bloed van de moeder, kunnen met de test afwijkingen of de aanwezigheid van een bepaald type kanker bij de moeder opgespoord worden. Niet alle ouders weten dat. Recent is gebleken dat het UZ Leuven ook gevallen van de ziekte van Duchenne via de NIPT heeft ontdekt. Omdat de technologie steeds vernieuwt, zullen in de toekomst nog meer afwijkingen kunnen opgespoord worden via de NIPT. Volgens professor Devisch van de UGent zou de moeder op voorhand moeten kunnen aangeven wat ze wel en niet wil weten en of ze een bredere analyse wil of niet.

In België wordt de test ook terugbetaald. Sommigen denken dat hierdoor de druk groter wordt om enkel nog gezonde kinderen op de wereld te zetten. (Eckert, 2019; Garderen, 2019)

Nu in China

AI-bedrijven in China hebben een groot voordeel t.o.v. bijvoorbeeld Amerikaanse of Europese bedrijven om hun technologieën te ontwikkelen: dankzij een zeer grote bevolkingsgroep en een gebrek aan een strenge privacywetgeving kunnen de AI-bedrijven over veel data beschikken om hun AI-systemen te trainen.

AI is in China veel meer dan bij ons aanwezig in het openbaar leven; vooral gezichtsherkenning wordt veel toegepast: de spoorwegpolitie in de provincie Henan spoort criminelen op door met speciale zonnebrillen met gezichtsherkenning de passagiers te identificeren. Universiteiten verifiëren de identiteit van de deelnemer aan een toelatingsexamen met gezichtsherkenningstechnologie en door de vingerafdruk te scannen. Er zijn onbemande winkels waar men betaalt met gezichtsherkenning. Op sommige plaatsen krijgt men in openbare toiletten pas toiletpapier, en dan nog in een beperkte hoeveelheid, na een gezichtsscan; dat om diefstal van het toiletpapier te voorkomen (Chan, 2019).

De Chinese overheid heeft een sociaal kredietsysteem ingevoerd dat in 2020 overal in China moet zijn uitgerold. In het kader hiervan plaatst men overal camera's (in ziekenhuizen, op universiteitscampussen, in treinstations, in openbare toiletten, langs belangrijke verkeerswegen). Men gebruikt gezichtsherkenning om te weten wie waar vertoeft en wat uitspookt. Bij het begaan van een verkeersovertreding krijgt men strafpunten en kan men ook aan de kaak worden gesteld. Spielen of vuilnis op straat gooien doet punten verliezen. Bloed geven, vrijwilligerswerk doen of een bejaarde de straat helpen oversteken, levert punten op. Teveel strafpunten leidt tot een internetabonnement met lage surfsnelheid of minder kans op een job. Sommigen zullen geen leningen meer kunnen afsluiten, geen woning meer kunnen huren of kopen, niet meer op reis kunnen gaan met het vliegtuig en uitgesloten worden van hoog aangeschreven privéscholen (Rammeloo, 2019; Selwyn, 2019; Botsman, 2017; Bouwma, 2018).

De Chinese overheid stelde in juli 2017 als streefdoel om tegen 2030 wereldleider in AI te zijn. De toekomst zal uitwijzen of ze daarin slagen.

Men wil er AI gebruiken in de meeste aspecten van het dagelijks leven: trans-

port, de overheid, gezondheidszorg en onderwijs. Daarom versterkt men het onderwijs in AI in basisscholen en in het secundair onderwijs. De stad Guangzhou streeft ernaar dat tegen 2022 alle basisscholen en alle middelbare scholen in de stad een cursus over AI zullen hebben binnen het reguliere curriculum (The Straits Times, 2019). Ook uitgeverijen spelen erop in (Peng, 2018): er verschenen al meerdere reeksen van handboeken over AI voor zowel het basisonderwijs als het secundair onderwijs.

De scholen pogen met AI ook om betere leerprestaties bij leerlingen te bekomen. Ze gaan daarin zeer ver. In een middelbare school in Hangzhou werd 'smart eye' geïnstalleerd (Connor, 2018). Elke beweging en elke gezichtsuitdrukking van de leerlingen wordt geregistreerd door drie camera's boven het lesbord. Het systeem zou in staat zijn om met technologie van gezichtsherkenning te beslissen of een leerling de les aandachtig volgt. Indien niet, dan wordt een melding gestuurd naar de leerkracht, die geacht wordt actie te ondernemen. In een basisschool moeten leerlingen een soort diadeem dragen die hun hersengolven registreert (Wall Street Journal video, 2019). De school beweert op die manier te kunnen monitoren of leerlingen wel aandachtig zijn in de les of zich concentreren op een taak. De informatie wordt in real time aan de leerkracht bezorgd. Ook de ouders krijgen hierover een rapport, al vóór hun kind thuiskomt van school. Indirect worden hier ook de leerkrachten mee gecontroleerd. Wat als een halve klas niet bij de les is?

Er zijn ook scholen die een 'slim' uniform invoeren. In de kledij van de leerlingen zitten microchips waarmee bv. gecontroleerd kan worden of een leerling wel aanwezig is in de klas of zich ergens in de school bevindt waar leerlingen niet toegelaten zijn. De chips kunnen ook detecteren of een leerling aan het indommelen is in de les (Fang, 2018; Caiyu, 2018).

AI-bedrijven zetten sterk in op gepersonaliseerd leren m.b.v. AI-technologie. Yixue Education bv. maakt hiervoor gebruik van natural language processing en technologie om emoties te detecteren en gaat er prat op, met wat ze 'Squirrel AI' noemen, verbluffende resultaten neer te zetten (Selwyn, 2019; McLaughlin, 2018; Yu, 2018).

Samengevat

De term 'artificiële intelligentie (AI)' bestaat reeds sinds de jaren 50 en de ontwikkeling ervan kende ups en downs. De mindere periodes zijn gekend als AI-winters.

Van AI doen veel definities de ronde. AI behelst immers zoveel verschillende aspecten en invalshoeken dat een algemene definitie moeilijk is.

Binnen de AI-systemen onderscheidt men kennisgebaseerde en datagebaseerde systemen.

Een kennisgebaseerde aanpak houdt in dat men de kennis van menselijke experts zoveel mogelijk in regels probeert te gieten om deze kennis eigen te maken aan een expertsysteem.

Bij een datagebaseerde aanpak worden met statische methodes patronen in relevante data opgespoord en dan gebruikt om nieuwe problemen op te lossen. Men spreekt dan van machinaal leren (*machine learning*, ML).

Machine learning (ML) omvat *Deep Learning* (DL), dat een revolutie veroorzaakt heeft in het veld van ML door opmerkelijke resultaten te boeken op het vlak van spraak- en beeldherkenning.

Nieuwe technologieën veranderen de maatschappij en sommige doen dat op een zeer ingrijpende manier. De impact van AI zal mogelijk nog groter zijn dan de komst van het internet.

AI is al aanwezig in het dagelijks leven en is ook al doorgedrongen in de juridische wereld en de gezondheidszorg. AI speelt ook een rol bij het opsporen van kunstvervalsingen en geeft nieuwe onderzoeksdomeinen, zoals computerlinguïstiek, een boost.

Hoewel men optimistisch is over de toekomstige ontwikkelingen binnen AI, is enige realiteitszin op zijn plaats.

Men moet beseffen dat sommige zaken veel tijd zullen vergen, bv. vertalen tussen om het even welke talen en het begrijpen van taal.

Net zoals bij andere technologische ontwikkelingen steken ook hier ethische dilemma's de kop op. In tegenstelling tot wat men doorgaans denkt, neemt een AI-systeem geen 100 % 'objectieve' beslissingen. Een AI-systeem is 'voorin genomen', meestal te wijten aan de data waarmee het systeem werd getraind. Deze *bias* moet men zoveel mogelijk controleren en tot het minimum herleiden. Bias komt voor als de data niet representatief zijn. Als de gebruikte data gekleurd zijn door een aanwezige bias in de maatschappij, zoals stereotypen, dan zal dit ook doorgegeven worden aan het ML-systeem.

Kijktip

Mobiel bellen in 1998.

<https://youtu.be/TNwhIHqM60g> (Bromet, 2016).

Mobiel bellen: 1998 vs NU.

<https://youtu.be/mQVkpdzPGtQ> (DWDD, 2019).

This 'Racist soap dispenser' at Facebook office does not work for black people.

https://youtu.be/YJjv_0eiHmo (Futureism, 2017).

Leestip

Mark Coeckelbergh. AI ETHICS (Coeckelbergh, 2020).

Kathleen Gabriels. Onlife (Gabriels, 2016).

Kathleen Gabriels. Regels voor robots. Ethiek in tijden van AI (Gabriels, 2019).

FUNDAMENTEN VAN MACHINAAL LEREN

6.1 *Machine learning*

6.1.1 *Machine learning-algoritmes*

Machine learning (ML) is een populair en succesvol onderdeel van de artificiële intelligentie. Machine learning is een computerwetenschappelijke discipline waarin men vooral proefondervindelijk te werk gaat, maar die wel wiskundig onderbouwd is, en waarin men gebruikmaakt van principes uit de wiskundige statistiek (Chollet, 2018).

Een machine learning-systeem verwerft met lerende algoritmes kennis uit data met de bedoeling uitkomsten te kunnen voorspellen betreffende nieuwe data. Deze voorspellingen worden gedaan met een bepaalde zekerheid. Een ML-systeem neemt zijn beslissingen dus niet op basis van vooraf in detail geprogrammeerde instructies.

Lerende algoritmes zijn algoritmes waarbij het ML-systeem zelf gaandeweg aanpassingen doet aan de aanwezige parameters tijdens het leerproces, om geleidelijk aan te komen tot betere prestaties.

Met ML kan men problemen van classificatie en regressie behandelen (zie kader 'Regressie en classificatie').

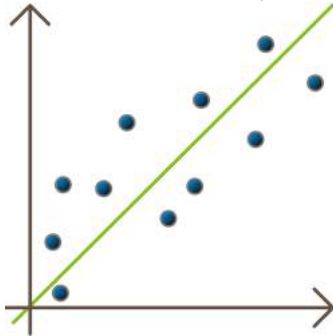
De nieuwe data moeten wel gelijkwaardig zijn aan de aangeboden data.

Voorspellen betekent bijvoorbeeld dat er uit voorbije tendensen cijfers voor de toekomst gegenereerd worden of dat een object bij een bepaalde klasse wordt ingedeeld.

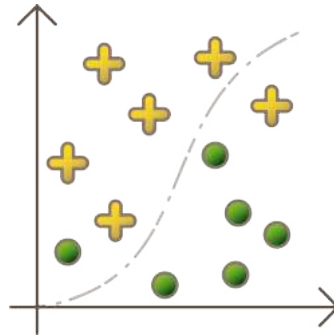
Uitleg over deze parameters die worden aangepast, vindt u in hoofdstuk 10.

Regressie en classificatie

Bij regressie is het de bedoeling dat het ML-model een kromme of een oppervlak bepaalt dat het best bij de gegeven punten past. Eens deze kromme of dit oppervlak gekend is, kan het model voorspellingen doen naar uitkomsten bij nieuwe data. Bv. uit het zeeniveau van de voorbije decennia in Oostende, het zeeniveau van de komende jaren afleiden.



Figuur 6.1: Regressie.



Figuur 6.2: Classificatie.

Bij classificatie zijn er verschillende klassen waartoe de data kunnen behoren. Het is de bedoeling dat het ML-model van nieuwe data kan bepalen tot welke klasse ze behoren. Bv. van een foto kunnen zeggen of er al dan niet een huidmondje op staat.

De aangeboden data geven dus informatie over een bepaald probleem aan de computer. Deze data moeten echter in een voor het probleem geschikte vorm aangeboden worden. Men heeft m.a.w. een representatie van het probleem nodig waarmee de computer aan de slag kan. Vervolgens verwerkt het ML-systeem de informatie en komt er een output.

De data worden dus voorbereid voordat men ze aan het netwerk geeft. Een ML-model kan immers niets aanvangen met een foto, een tekst of een geluidsfragment als die niet op de juiste manier gerepresenteerd worden. Concreet werkt een machinaal leren-model met *tensoren*, dit zijn meerdimensionale rasters die getallen bevatten, of anders gezegd matrices of stapels ervan

(zie hoofdstuk 10).

6.1.2 *Supervised, unsupervised en reinforcement learning*

Binnen ML onderscheidt men verschillende types van leren: supervised, unsupervised, en reinforcement learning.

Bij *supervised learning* leert het systeem uit een dataset waarbij elk gegeven bestaat uit twee componenten: een input gekoppeld aan een label (de verwachte output). Het labelen van de voorbeelden gebeurt vaak manueel door mensen, men noemt dat annoteren. Het systeem voert een algoritme uit dat er geleidelijk aan voor zorgt dat het systeem focust op relevante patronen in de data.

Bij *unsupervised learning* bevat de dataset geen labels. Het AI-systeem moet op zoek naar kenmerken bij de verschillende voorbeelden en moet ze zo, door het ontdekken van patronen, verdelen in klassen. Men kan het systeem bv. ongelabelde foto's aanbieden van appels en peren. Het systeem gaat op zoek naar patronen om zo het onderscheid tussen de twee soorten fruit te kunnen maken (zie Figuur 6.3).

Bij *reinforcement learning* streeft het AI-systeem naar een beloning. Om bv. goed te worden in een bepaald videogame, gaat het AI-systeem het spel heel veel spelen en daaruit leren welke acties het beter vermijdt en welke het best onderneemt om te kunnen winnen. Zo werd Google DeepMind's AlphaGo Zero in 2017 via reinforcement learning een topspeler in go, nog beter dan AlphaGo die de beste menselijke speler eerder al versloeg.

6.1.3 *Generaliseren*

Het doel van ML is om modellen te bekomen die goed kunnen generaliseren, m.a.w. modellen die goed presteren op ongeziene data. Om dit te bewerkstelligen



Zelf huidmondjes aanduiden op foto's kan online via <https://aiopschool.be/ki-ks/dl/data/annoteren/>.



Figuur 6.3: Het netwerk gaat op zoek naar kenmerken.

zijn er verschillende principes die men in acht kan nemen. Men splitst bv. de dataset op in een *trainingset*, *valideringsset* en *testset*.

Het netwerk wordt getraind met de trainingdata en wordt tezelfdertijd geëvalueerd met de valideringsdata. Hierbij is het belangrijk dat de trainingset en de valideringsset disjunct zijn: ze bevatten dus geen gemeenschappelijke voorbeelden. De trainingdata en valideringsdata worden meerdere keren doorlopen, men spreekt van meerdere epochs. Na elke *epoch* worden de prestaties op de trainingdata nagegaan. Op basis van deze informatie past het algoritme de aanwezige parameters na elke epoch aan.

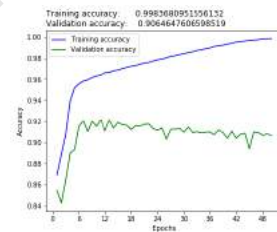
Eens getraind, wordt het model één keer getest met de testdata. Met het resultaat op deze testdata meet men het vermogen van het model om te veralgemenen.

De generalisatie vormt een uitdaging (zie Figuren 6.5, 6.6 en 6.7). Bij het begin van de training kiest het algoritme willekeurige waarden voor de parameters van het model. Het is dus niet verwonderlijk dat het netwerk initieel niet goed presteert. Na elke epoch past het algoritme de waarden van de parameters aan en verbeteren de resultaten. Zowel op de trainingdata als op de valideringsdata presteert het model steeds beter.

Zolang het netwerk echter nog niet alle relevante patronen in de trainingdata heeft ontdekt, zegt men dat het netwerk *underfits*. Men gaat dan door met het trainen van het netwerk. Het netwerk leert bij elke epoch meer uit de trainingdata. Na enige iteraties merkt men op dat de generalisatie niet meer verbetert: het model presteert wel steeds beter op de trainingdata, maar daarentegen worden de resultaten op de valideringsdata slechter. Dat komt omdat het model patronen begint te zien in de trainingdata die irrelevant zijn voor ongeziene data. Het model *overfits*. Een netwerk met teveel lagen of teveel parameters zal sneller overfitten. Het komt er dus op aan om de architectuur van het netwerk zo eenvoudig mogelijk te kiezen en op het juiste mo-



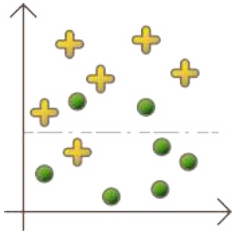
Bekijk de notebook 'De Fundamentele van een diep neurale netwerk voor beeldherkenning'.



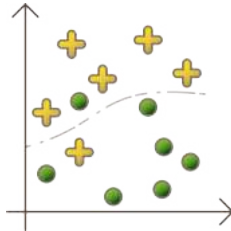
Figuur 6.4: Evolutie van de *accuracy* tijdens de training van een netwerk. De *accuracy* is het percentage correct geclassificeerde data. Het model overfits.

Overfitting komt ook aan bod in hoofdstuk 14.

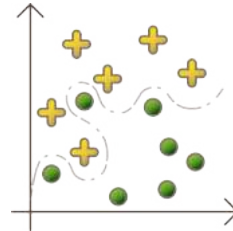
ment te stoppen met trainen om een zo goed mogelijk presterend model te bekomen.



Figuur 6.5: Underfitting.



Figuur 6.6: Optimaal.



Figuur 6.7: Overfitting.

6.2 Deep learning

Binnen het machinaal leren onderscheidt men *deep learning* (DL). Deep learning-modellen, m.a.w. diepe neurale netwerken, worden ingezet voor spraakherkenning en objectdetectie, maar ook voor het begrijpen van taal en toepassingen binnen de genetica. Google Translate en gezichtsherkenning op Facebook zijn voorbeelden van *deep learning*.

Eens DL succesvol werd in beeldherkenning, werd de interesse van de industrie gewekt. Techreuzen zoals Google, Facebook, Microsoft, Baidu, Apple en Amazon gebruiken ML en DL om de marketing en de productontwikkeling te optimaliseren.

De ontwikkeling en verspreiding van Python hebben bijgedragen tot het succes van DL. Met vrij eenvoudige Python scripts kan men al aan geavanceerd DL-onderzoek doen en is het mogelijk om kleine DL-modellen op een laptop te laten lopen. Het aantal mensen dat in DL werkt, is daardoor de voorbije jaren toegenomen van enkele honderden tot enkele tienduizenden (Chollet, 2018).

Hoewel men spreekt over neurale netwerken, hebben ze niet als doel om het menselijk brein te modelleren,

In hoofdstuk 5 wordt aandacht besteed aan de soorten AI en staan er meer voorbeelden van deep learning. De basisconcepten van machinaal leren komen uitgebreid aan bod in hoofdstuk 11.

maar dienen ze om functies te kunnen benaderen waarbij de geleerde functie kan generaliseren gebaseerd op statistiek.

Een neurale netwerk wordt opgebouwd door verschillende lagen (*layers*) aan elkaar te schakelen. Hoe meer lagen er zijn, hoe dieper het netwerk is. Men spreekt van een netwerk omdat er veel verschillende functies worden samengesteld.

Ingrid Daubechies omschrijft het als volgt (Daubechies, 2015): "De computers krijgen heel veel gelabelde voorbeelden aangereikt. Het doel is dat het netwerk de functie leert kennen die elk voorbeeld koppelt aan het juiste label én dat het model dan ook ongeziene data van het juiste label kan voorzien. Een deep learning-systeem bouwt deze functie op met verschillende functies in de opeenvolgende lagen van zijn netwerk. Als men al deze opeenvolgende functies samenstelt, bekomt men de gezochte functie".

François Chollet verwoordt het in zijn boek (Chollet, 2018) als volgt: "In elke laag wordt de data als het ware gefilterd: de gegevens komen binnen in een bepaalde vorm en verlaten de laag in een meer bruikbare vorm".



Deze lagen worden concreet toegepast in hoofdstuk 11 en hoofdstuk 14.

Het algoritme gaat daartoe zelf op zoek naar de beste waarden voor de parameters zodat de uiteindelijke functie de beoogde labels zo goed mogelijk benadert. Het uiteindelijke resultaat noemt men 'het model', de weg ernaar toe 'leren'.

Figuur 6.8: Het netwerk ontdekt geleidelijk aan meer patronen.

Een convolutioneel neurale netwerk (convnet) is een diep neurale netwerk dat zeer geschikt is voor beeldherkenning. Een convnet ontdekt in een eerste laag kleine lokale patronen zoals randen, in een tweede laag ziet het kenmerken die opgebouwd worden met de kenmerken uit de eerste layer. Daardoor kunnen deze netwerken op een efficiënte manier stijgende com-

plexiteit en abstracte visuele concepten aan: randen vormen motieven, motieven vormen samen onderdelen en onderdelen vormen objecten (zie Figuur 6.8) (LeCun et al., 2015).

Het neurale netwerk is dus opgebouwd uit lagen: een invoer- en een uitvoerlaag (*input* en *output*), met daartussen de verborgen lagen (*hidden layers*) waarvan de outputs niet worden getoond. Elke laag bevat eenheden (*units*), ook neuronen of knopen genoemd. In elke laag werkt er een lineaire functie in op de invoer van die laag. Deze lineaire functie wordt bepaald door meerdere parameters, haar coëfficiënten, en wordt gevolgd door een niet-lineaire activatiefunctie. Zo bekomt men de uitvoer van die laag die dan dient als invoer van de volgende laag.

Anders gezegd: elk neuron ontvangt informatie van de neuronen uit de vorige laag. Aan die informatie worden verschillende gewichten (*weights*) toegekend - dat zijn de eerder vermelde coëfficiënten - en afhankelijk van de activatiefunctie belandt het neuron dan in een bepaalde toestand. Zo gaat de informatie 'voorwaarts', laag na laag, t.e.m. de neuronen in de uitvoerlaag; men spreekt van een *feedforward* netwerk. Het algoritme van het netwerk moet de parameters van de tussenliggende lagen op zo'n manier aanpassen dat de beoogde output wordt bekomen. Dit vereist het berekenen van de afgeleiden van de samengestelde functies. In hoofdstuk 11 wordt dit concreet toegepast.

De basisconcepten van machinaal leren komen uitgebreid aan bod in hoofdstuk 11.

Bij het ontwerpen van een neurale netwerk, moet men nadenken over de architectuur:

- hoeveel layers;
- hoeveel neuronen in elke layer;
- welke activatiefuncties;
- hoe wordt de prestatie van het netwerk gemeten?

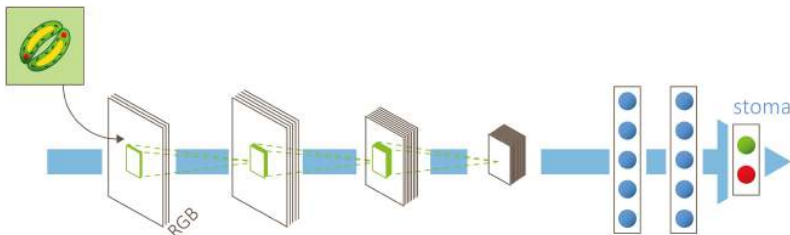
Hoe meer parameters een netwerk heeft, hoe meer patronen het kan ontdekken. Maar men moet zich behoeden voor teveel parameters. Netwerken met teveel parameters zullen sneller overfitten. Een eenvoudiger netwerk is vaak een betere keuze.

Neuraal netwerk van KIKS

Bij het project KIKS bestaat het neurale netwerk uit een convolutioneel netwerk gevolgd door een feedforward netwerk.

Een convnet behoort tot het domein van supervised learning. Het AI-systeem van KIKS bv. leert relevante kenmerken onderscheiden omdat het vele voorbeelden van delen van bladeren te zien krijgt, voorbeelden met het label 'stoma' ofwel met het label 'geen stoma'.

Als input krijgt het model een microfoto van een deel van een blad van een plant. Door o.a. randen te detecteren en die dan in volgende lagen te combineren tot een ovaal of tot de opening van een stoma, kan het netwerk uiteindelijk de stomata op het blad detecteren.



Figuur 6.9: Neuraal netwerk van KIKS.

"Er is bewezen dat neurale netwerken alle mogelijke functies kunnen genereren en dat er een unieke overeenkomst is tussen een netwerk en de functie die het genereert. Maar dan ging men uit van netwerken met een extreem groot aantal lagen en extreem veel knopen per laag. In de praktijk zijn er wel heel veel lagen en knopen, maar niet zoveel als waarvan men uitging. Men begrijpt dus eigenlijk nog niet waarom neurale netwerken zo goed werken als ze doen. Er is hiervoor

nog fundamenteel wiskundig onderzoek nodig", zegt Daubechies (Daubechies, 2015).

Ingrid Daubechies

Ingrid Daubechies is een gerenommeerde Belgische wiskundige en professor aan de Duke University.

Ze is wereldvermaard voor het ontwikkelen van de theorie rond de zogenaamde *Daubechies wavelets*. Dit zijn wiskundige transformaties die men kan uitvoeren op digitale beelden om ze te ontleden tot basisblokken die de beelden opdelen in grove en gedetailleerde informatie. Dit proces is cruciaal voor beeldcompressie en werd daarom meteen opgepikt door de FBI voor het comprimeren van vingerafdrukken. De transformaties vormen nu ook de basis voor JPEG2000, het standaardformaat van digitale cinema films.

Ze maakte ook deel uit van het expertenteam van de restauratie van het Lam Gods in Gent waarbij opnieuw wiskundige transformaties helpen om de conservatiestaat van schilderijen te bestuderen aan de hand van digitale foto's.

Met haar huidige onderzoek ontwikkelt ze tools uit de analytische meetkunde om oppervlakken te vergelijken. Die tools hebben toepassingen in veel wetenschappelijke disciplines, waaronder medische en biologische, waarin men oppervlakken kan leren waarop verkregen data liggen. Zo kunnen verbanden gelegd en ontrafeld worden. (Duke today, 2019)

Ze vindt het belangrijk om onderwijs in wiskunde wereldwijd te promoten, en ook om vrouwelijke wiskundigen in de kijker te zetten. Voor haar werk ontving ze al meerdere prijzen.

(Ann Dooms, persoonlijke communicatie)

6.3 *Standaardiseren en normaliseren van de data*

Beschouw de tabellen met bivariate data uit Figuren 6.10, 6.11 en 6.12 (Callaert et al., 2012; Callaert & Bogaerts, 2012). Men wil nagaan hoe sterk de samenhang tussen de veranderlijken is.

Het eerste dat men daarvoor doet is de data gaan visualiseren. De correlatie tussen bivariate gegevens kan men visueel proberen in te schatten door de over-

Bivariate gegevens zijn data van twee variabelen waarbij elke waarde van de ene variabele wordt gekoppeld aan een bepaalde waarde van de andere variabele.

eenkomstige puntenwolk te bekijken. Bekijk de puntenwolken uit Figuren 6.13, 6.14 en 6.15 die overeenkomen met respectievelijk Tabellen 1, 2 en 3 uit Figuren 6.10, 6.11 en 6.12.

	0	1
0	16.25	4.6
1	16.75	3.4
2	17.00	27.4
3	17.25	14.2
4	17.75	28.6
5	18.25	3.4
6	18.50	14.2
7	18.75	16.6
8	19.00	13.0
9	19.25	26.2
10	19.75	19.0
11	20.00	23.8
12	20.25	34.6
13	20.50	14.2
14	20.75	25.0
15	21.25	17.8
16	21.75	11.8
17	22.00	13.0
18	22.25	41.8
19	23.00	39.4

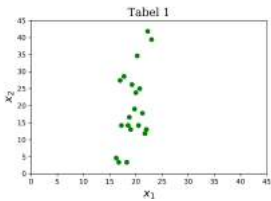
Figuur 6.10: Tabel 1.

	0	1
0	9	21.000
1	11	20.875
2	12	23.375
3	13	22.000
4	15	23.500
5	17	20.875
6	18	22.000
7	19	22.250
8	20	21.875
9	21	23.250
10	23	22.500
11	24	23.000
12	25	24.125
13	26	22.000
14	27	23.125
15	29	22.375
16	31	21.750
17	32	21.875
18	33	24.875
19	36	24.625

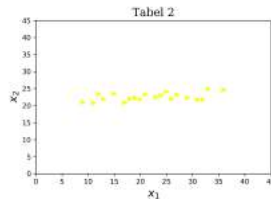
Figuur 6.11: Tabel 2.

	0	1
0	20.125	8
1	20.175	7
2	20.200	27
3	20.225	16
4	20.275	28
5	20.325	7
6	20.350	16
7	20.375	18
8	20.400	15
9	20.425	26
10	20.475	20
11	20.500	24
12	20.525	33
13	20.550	16
14	20.575	25
15	20.625	19
16	20.675	14
17	20.700	15
18	20.725	39
19	20.800	37

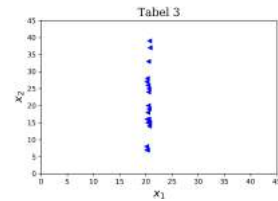
Figuur 6.12: Tabel 3.



Figuur 6.13: Tabel 1.



Figuur 6.14: Tabel 2.



Figuur 6.15: Tabel 3.

Op het eerste gezicht is men geneigd uit de puntenwolken af te leiden dat de bivariate data uit de derde tabel de meeste samenhang vertonen.

Men kan echter ook de correlatiecoëfficiënt berekenen (zie ook paragraaf 11.10). De correlatiecoëfficiënt is voor alle drie de tabellen dezelfde, ongeveer 0,5, waaruit volgt dat de samenhang voor elke tabel dezelfde is.

De grootteorde van de veranderlijken en het bereik dat men kiest op de assen, bepalen voor een groot deel hoe de puntenwolk die de bivariate gegevens representeert er zal gaan uitzien. Door de data te standaardiseren wordt dit effect geneutraliseerd. Dus als men de correlatie van meerdere datasets grafisch met elkaar wilt vergelijken, dan werkt men het best met gestandaardiseerde data.

Men brengt de data in gestandaardiseerde vorm met de volgende formule:

$$z = \frac{x - \bar{x}}{s(x)},$$

met \bar{x} het gemiddelde en $s(x)$ de standaardafwijking van de x -waarden, d.i. de gemiddelde kwadratische afwijking t.o.v. het gemiddelde \bar{x} .

In de statistiek spreekt men van de Z-score.

De vorm van de puntenwolk bij bivariate gegevens is dus maar betrouwbaar als de gegevens gestandaardiseerde

De correlatiecoëfficiënt r is een getal in $[-1, 1]$ dat een maat is voor de samenhang tussen de bivariate gegevens. Hoe dichter $|r|$ bij 1 ligt, hoe meer samenhang er is.

De correlatiecoëfficiënt r is:

$$r = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s(x)} \cdot \frac{y_i - \bar{y}}{s(y)} \right).$$

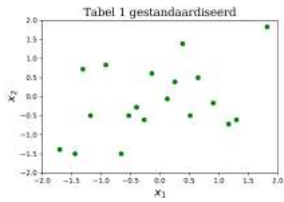
De producten van de gestandaardiseerde x_i en y_i worden dus gesommeerd en deze som wordt gedeeld door $n - 1$.

Het standaardiseren heeft trouwens geen effect op de waarde van de correlatiecoëfficiënt.

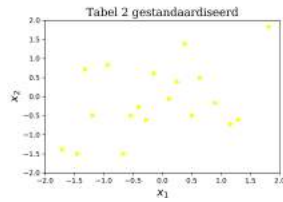


Voer dit zelf uit in de notebook 'Standaardiseren'.

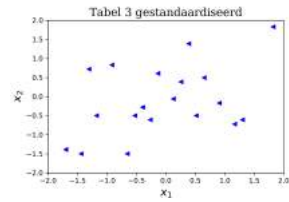
seerd zijn. Bekijk de puntenwolken van de gestandaardiseerde gegevens in Figuren 6.16, 6.17 en 6.18.



Figuur 6.16: Tabel 1.



Figuur 6.17: Tabel 2.



Figuur 6.18: Tabel 3.

Standaardiseren van variabelen houdt dus in dat men de gegevens op zo'n manier herschaalt dat men gegevens van bv. een verschillende grootteorde of in een verschillende eenheid met elkaar kan vergelijken of in verband brengen.

Na het standaardiseren liggen de meeste waarden tussen 0 en 1, wat voordelig is voor het rekenen door de computer. Rekenen met vrij grote getallen leidt al snel tot nog grotere getallen en tot numerieke instabiliteit, dat is een bijkomende reden waarom de data worden gestandaardiseerd. Ook zijn sommige algoritmes uit machinaal leren pas bruikbaar als de data gestandaardiseerd zijn, omdat die algoritmes zo opgesteld zijn.

Vooraf bij regressieproblemen zal men standaardiseren, omdat de correlatie daar tekenend is voor het probleem. Bij classificatieproblemen kiest men er vaker voor enkel te normaliseren. Dan liggen alle waarden tussen 0 en 1.

Men brengt de data in genormaliseerde vorm met de volgende formule:

$$x_n = \frac{x - x_{min}}{x_{max} - x_{min}},$$

met x_{max} de grootste en x_{min} de kleinste x -waarde.

Bij het standaardiseren van gegevens komt het er dus op neer dat de waarden zo herschaald worden dat ze een gemiddelde 0 krijgen en een standaardafwijking 1.

Bij normalisering komt het erop neer dat de data herschaald worden naar waarden tussen 0 en 1.

Samengevat

Machine learning (ML) is een populair en succesvol onderdeel van de artificiële intelligentie. Een machine learning-systeem verwerft met lerende algoritmes kennis uit data met de bedoeling uitkomsten te kunnen voorspellen betreffende nieuwe data, m.a.w. het systeem moet goed kunnen generaliseren. Lerende algoritmes zijn algoritmes waarbij het ML-systeem zelf gaandeweg aanpassingen doet aan de aanwezige parameters tijdens het leerproces, om geleidelijk aan te komen tot betere prestaties.

Binnen ML onderscheidt men verschillende types van leren: *supervised*, *unsupervised*, en *reinforcement learning*.

Met ML kan men problemen van classificatie en regressie behandelen. Bij regressieproblemen zullen de data doorgaans eerst worden gestandaardiseerd, bij classificatieproblemen eerder genormaliseerd.

Om een ML-model te bekomen dat goed kan generaliseren, zijn er verschillende principes die men in acht kan nemen, zoals het opsplitsen van de dataset in een trainingset, valideringsset en testset. Het netwerk wordt getraind met de trainingdata en wordt tezelfdertijd geëvalueerd met de valideringsdata. Eens getraind, wordt het model één keer getest met de testdata.

Men moet waakzaam zijn voor *underfitting* en *overfitting*. Bij *underfitting* zijn de prestaties van het systeem nog niet optimaal, door meer te trainen kunnen de prestaties nog verbeteren. Bij *overfitting* presteert het systeem steeds beter op de trainingdata, maar de prestaties op de valideringsdata verbeteren niet meer.

Binnen het machinaal leren onderscheidt men de diepe neurale netwerken die bijvoorbeeld worden ingezet voor beeldherkenning. Hoewel men spreekt over neurale netwerken, hebben ze niet als doel om het menselijk brein te modelleren, maar dienen ze om functies te kunnen benaderen waarbij de geleerde functie kan generaliseren gebaseerd op statistiek.

Een diep neuraal netwerk wordt opgebouwd door verschillende lagen aan elkaar te schakelen: een invoer- en een uitvoerlaag, met daartussen de verborgen lagen. Elke laag bevat eenheden (neuronen) en parameters. Het trainen van het netwerk bestaat erin dat het algoritme van het netwerk de parameters op zo'n manier aanpast dat de gewenste output wordt bekomen.

Leestip

François Chollet. Deep learning with Python (Chollet, 2018).

Ian Goodfellow, Yoshua Bengio en Aaron Courville. Deep learning (Goodfellow et al., 2016).

Versie 1.0

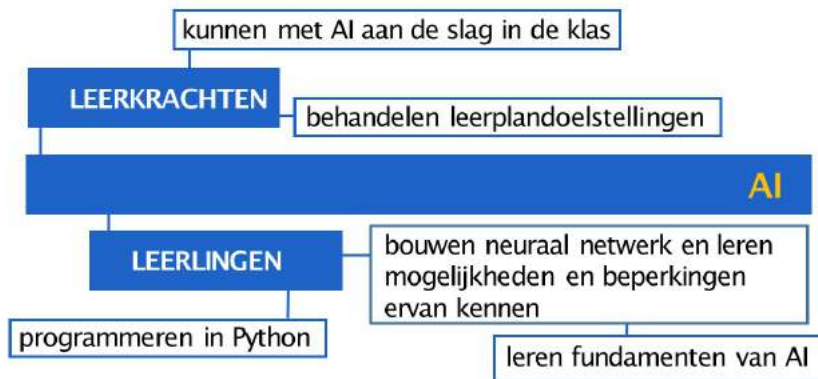
KIKS IN DE KLAS

7.1 Doel

Het KIKS-project heeft als doel de leerlingen de fundamenteën van AI te leren kennen en hen bij te brengen hoe ze vat kunnen hebben op AI-systemen die ze tegenkomen in hun dagelijks leven. We willen daarom leerkrachten aanmoedigen om met KIKS en AI aan de slag te gaan in de klas (zie Figuur 7.1), met of zonder programmeren. We hebben geprobeerd het project zo laagdrempelig mogelijk te houden door het materiaal online aan te bieden, door ervoor te zorgen dat er geen extra software geïnstalleerd moet worden en door in de nodige achtergrondinformatie te voorzien via deze handleiding.

We hopen dat er tijdens de projectlessen ook tijd wordt vrijgemaakt om stil te staan bij de ethische kwesties die opduiken door de aanwezigheid van AI en deep learning-systemen in het dagelijks leven.

Het lesmateriaal kan ook op een andere manier worden ingezet. Men kan bv. enkele notebooks uitkiezen om de leerlingen te leren programmeren in Python of men kan er opdrachten uit halen om te werken aan onderzoeks-



Figuur 7.1: Doelstellingen.

competenties in de lessen biologie, aardrijkskunde of wiskunde. Men kan de leerlingen aan de hand van enkele notebooks vertrouwd maken met de principes van digitale beeldverwerking of de leerkracht Nederlands of godsdienst kan uit deze handleiding inspiratie halen voor een les over de ethische aspecten van AI.

7.2 Microscopie

Niemand heeft reeds huidmondjes gezien met het blote oog, ook de leerlingen niet. Huidmondjes zijn voor hen dan ook een abstract gegeven. Voor de leerlingen én voor de leerkrachten is het daarom een toffe ervaring om binnen het kader van het KIKS-project naar huidmondjes te kijken met een microscoop. De leerlingen kunnen de huidmondjes ook door de microscoop fotograferen met een smartphone. Wij gebruiken met de leerlingen een monoclulaire microscoop die 400 keer vergroot.

We hebben tijdens de workshop microscopie eerst met levend materiaal gewerkt, wat mooie beelden ople-



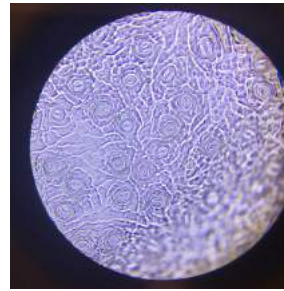
Figuur 7.2: © H. Crevits, met toestemming. Anna bewondert huidmondjes door de microscoop.

verde (zie Figuur 7.3). Daarvoor moeten de leerlingen erin slagen een stuk van de flinterdunne cuticula van het blad te verwijderen. Bij sommige planten lukt dat niet zo goed, bv. door de stugheid van het blad.

Dit kan echter opgevangen worden door dezelfde methode te gebruiken als de onderzoekers van de Plantentuin Meise, nl. een afdruk nemen met doorzichtige nagellak (Figuur 7.4).

Dat hebben we dan in tweede instantie ook gedaan. De foto's zijn dan niet meer zo mooi van kleur, maar grijsachtig (zie Figuur 7.5). Maar al bij al levert het nog steeds mooie plaatjes op; de natuur op zijn best.

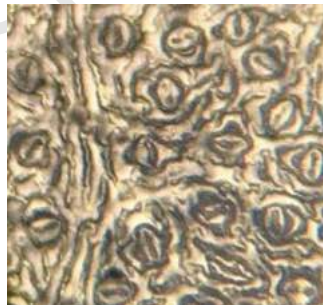
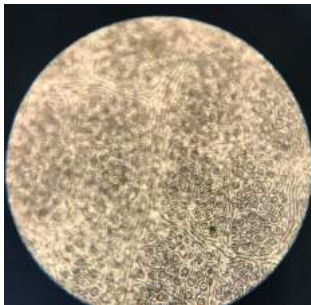
We zorgden ook voor variatie in de beelden: monocotylen en dicotylen, grote en kleine huidmondjes, stomata van verschillende vormen ...



Figuur 7.3: Microfoto van levend materiaal.



Figuur 7.4: Bladafdruk nemen met nagellak en plakband.



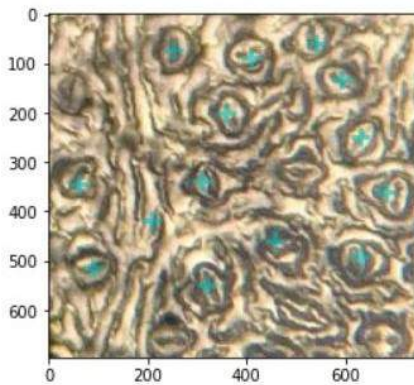
Figuur 7.5: Microfoto van een bladafdruk van klimop en een uitvergroting van een deel van deze foto.

7.3 Huidmondjes tellen op eigen microfoto's

De leerlingen voerden hun eigen foto's in in het KIKS-neuraal netwerk om de huidmondjes te tellen (zie Figuur 7.6 en paragraaf 14.5.2).

Ze botsten daarbij op meerdere problemen: onscherpe foto's, te kleine huidmondjes, moeilijk te herkennen huid-

mondjes, te veel kleuren in de foto's ...



Aantal stomata: 14

Hier kan ook de link worden gelegd met de *bias* waarmee men in deep learning-systemen te kampen heeft.

7.4 Digitale beelden en deep learning

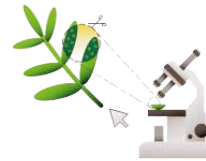
Om te kunnen begrijpen hoe het KIKS-netwerk die huidmondjes telt, moeten de leerlingen eerst heel wat kennis vergaren.

Ze moeten weten hoe een computer naar beelden kijkt, hoe een foto aan het netwerk gegeven wordt en hoe het netwerk met die gegevens rekt.

Ze moeten kennis maken met de bouwstenen van machinaal leren. Wat zijn kenmerken, wat zijn lagen? Welke hyperparameters kan men aanpassen? Hoe kan men de prestaties van het netwerk meten en grafisch voorstellen? Hoe kan een netwerk gegevens indelen in klassen? Wat is standaardiseren?

Tot slot leren de leerlingen wat convoluties zijn, wat het betekent als men zegt dat het netwerk leert, dat men moet opletten voor overfitting en hoe men de nodige rekenkracht binnen de perken kan houden.

Figuur 7.6: Getelde stomata op bladafdruk van klimop.



Huidmondjes tellen op eigen foto's kan met de notebook 'Eigen stomatadetectie' in de map Introductie Deep Learning.

Over bias leest u in paragraaf 5.6.



Figuur 7.7: Josse aan de slag met de notebooks.

Hoe het netwerk geïmplementeerd wordt, staat beschreven in hoofdstuk 14.

Samengevat

Het KIKS-project heeft als doel de leerlingen de fundamentele van AI te leren kennen en hen bij te brengen hoe ze vat kunnen hebben op AI-systemen die ze tegenkomen in hun dagelijks leven.

We hopen dat er tijdens de projectlessen ook tijd wordt vrijgemaakt om stil te staan bij de ethische kwesties die opduiken door de aanwezigheid van AI en deep learning-systemen in de maatschappij.

Het lesmateriaal kan op meerdere manieren worden ingezet. Men kan bv. enkele notebooks uitkiezen om de leerlingen te leren programmeren in Python of men kan er opdrachten uit halen om te werken aan onderzoekscompetenties in de lessen biologie, aardrijkskunde of wiskunde. Men kan de leerlingen aan de hand van enkele notebooks vertrouwd maken met de principes van digitale beeldverwerking of de leerkracht Nederlands of godsdienst kan uit deze handleiding inspiratie halen voor een les over de ethische aspecten van AI.

Voor de leerlingen én voor de leerkrachten is het een toffe ervaring om naar huidmondjes te kijken met een microscoop. De leerlingen kunnen de huidmondjes door de microscoop fotograferen met een smartphone, en vervolgens de huidmondjes op hun eigen foto's door het KIKS-neuraal netwerk laten tellen.

Om te kunnen begrijpen hoe het KIKS-netwerk die huidmondjes telt, moeten de leerlingen bv. weten hoe een computer naar beelden kijkt, hoe een foto aan het netwerk gegeven wordt en hoe het netwerk met die gegevens rekt.

DE NOTEBOOKS VAN KIKS

8.1 Python, Jupyter Notebook en KIKS

Notebooks zijn digitale documenten die zowel uitvoerbare code bevatten als tekst, afbeeldingen, video, hyperlinks . . . Het is dus mogelijk om nieuwe begrippen aan te brengen, opdrachten te formuleren, foto's te tonen, bepaalde fundamentele van programmeren uit te leggen, aanwezige code uit te voeren en zelf code op te stellen.

Jupyter Notebook is een open source webapplicatie waarin men notebooks kan creëren.

De notebooks bestaan eigenlijk uit een opeenvolging met cellen: Markdown-cellen voor tekst en beeld en code-cellen om te programmeren.

De notebooks van KIKS zijn onderverdeeld in drie categorieën:

- basisprincipes van programmeren en beeldverwerking;
- principes van machinaal leren;
- fundamentele van diepe neurale netwerken.

De notebooks worden aangeboden via een webserver, toegankelijk via een webbrowser. Op de gebruikte com-

puters moet geen extra software worden geïnstalleerd om met de notebooks aan de slag te gaan.

In de notebooks worden groene, blauwe en gele kaders gebruikt. De groene vertellen wat de notebook behandelt en leggen linken naar de leerinhouden van het KIKS-project. De blauwe kaders vatten begrippen samen uit de computerwetenschappen. De gele kaders geven tips.

Het programmeren gebeurt met Python 3. Python is een zeer toegankelijke programmeertaal, die vaak ook zeer intuïtief is in gebruik. Bovendien is Python zo populair dat er heel wat modules voorhanden zijn die men vrij kan gebruiken. Python draagt zo ook bij tot de democratisering van deep learning.

In een module zitten heel wat functies vervat die ervaren informatici reeds hebben geprogrammeerd. Wie dat wilt, kan die modules en de bijbehorende functionaliteiten gebruiken. Maar daarvoor moet men eerst de gewenste module importeren in het Python-script.

Python is een object-georiënteerde taal. Alles is er een object: elk getal, elk stuk tekst ... Een object neemt een bepaalde plaats in in het geheugen en heeft een bepaalde waarde. Elk object heeft ook een type, bv. integer, string en list. Alle objecten met eenzelfde type behoren tot dezelfde klasse. Binnen deze klasse zijn er methodes voorzien die men specifiek voor objecten van dit type kan oproepen. Naast deze methodes zijn er ook functies die men kan gebruiken, zoals `print()`, om iets te laten verschijnen op het scherm, en `input()`, om iets op te vragen aan de gebruiker.

8.2 Praktisch

Bij het begin van een notebook worden eerst de nodige modules geïmporteerd. De modules NumPy en Matplotlib zijn courant gebruikte modules.



Figuur 8.1: De notebooks zijn online beschikbaar via <https://www.aiopschool.be/kiks>. Er is geen installatie van extra software nodig om de notebooks te gebruiken.

Een programma in Python noemt men een *script*.

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt

```

De module NumPy laat toe om wetenschappelijke berekeningen uit te voeren en om bv. met matrices te werken. Met de interface pyplot van de module Matplotlib kan men grafieken creëren en ze personaliseren.

Tekst wordt ingevoerd met Markdown en wiskundige formules met $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (tussen \$\$), beide in een Markdown-cel. Code wordt ingevoerd in een code-cel. Cellen kunnen worden bewerkt, verwijderd, toegevoegd, gekopieerd, geknipt en geplakt.

Interessant om te weten is dat in een notebook alle code samen hoort. De notebook onthoudt als het ware welke code reeds werd uitgevoerd, ongeacht in welke volgorde die werd ingetikt in de notebook. Het is het tijdstip van uitvoeren dat de sequentie van het uiteindelijke script bepaalt.

We stimuleren dat er bij het invoeren van code aandacht wordt besteed aan een leesbare programmeerstijl en de nodige verduidelijkende commentaar.

8.3 De KIKS-notebooks

Bij het werken met de notebooks van KIKS respecteert men het best deze volgorde:

Introductie Python

- Aan de slag
- Rekenen
- Datastructuur
- Structuren
- Datastructuur NumPy



NumPy en Matplotlib worden bv. gebruikt in de notebooks 'Extra datastructuren met NumPy' en 'Matrices en afbeeldingen in grijswaarden'.



De belangrijkste instructies en sneltoetsen om vlot met de notebooks te kunnen werken, vindt u in de startnotebook 'KIKS: de notebooks'.

- Functies
- Klassen
- Grafieken
- CO₂
- Keelingcurve
- Verband tussen CO₂ en temperatuur
- Smeltende gletsjer: Morteratsch
- Smeltende gletsjer: Silvretta
- Tensoren
- Matrices en afbeeldingen in grijswaarden
- Tensoren en RGB
- Van jpg naar dat of npy
- verborgen boodschap

Introductie Machine Learning

- Standaardiseren
- Regressie met de Morteratschgletsjer
- Regressie met de Silvrettagletsjer
- Classificatie met de Iris dataset (Perceptron)
- Opdracht bij classificatie met de Iris dataset
- Regressie met data over de Iris virginica
- Hoogte bomen en afmetingen stomata in het Amazonewoud
- Stomata op bezonde en beschaduwde bladeren
- Zeeniveau

Introductie Deep Learning

- Detectie van stomata in een afbeelding
- Stomatadetectie
- Convolutie
- Convolutie: de bewerking
- ReLU
- Gradient descent
- Fundamenten van een diep neurale netwerk voor beeldherkenning
- Overfitting
- MNIST
- Binaire classificatie met neurale netwerk met een verborgen laag
- Binaire classificatie met neurale netwerk met twee verborgen lagen
- Multiklassenclassificatie voor twee klassen met neurale netwerk met twee verborgen lagen
- Classificatie van de MNIST dataset met een convolutioneel neurale netwerk
- Classificatie van de Iris dataset met een verborgen laag
- Classificatie van de Iris dataset met alle kenmerken
- Binaire classificatie met neurale netwerk met een verborgen laag en met verbeterde SGD

In de eerste notebooks, Introductie Python, worden fundamentele zaken aangebracht: hoe een printopdracht invoeren, wat zijn datastructuren, hoe met herhalingslussen en condities werken, hoe een grafiek tekenen,

hoe kijkt een computer naar een afbeelding, hoe functies definiëren.

De tweede reeks notebooks, Introductie Machine Learning, besteedt aandacht aan standaard machine learning technieken zoals regressie, classificatie met het Perceptron, en het standaardiseren van de variabelen.

In de derde reeks wordt uitgelegd hoe een diep neuraal netwerk wordt opgebouwd, wat activatiefuncties zijn en gradient descent, hoe een convolutie gebeurt, wat underfitting en overfitting zijn ... Er is ook een notebook om huidmondjes te laten tellen gebruikmakend van een convolutioneel neuraal netwerk.



De eerste notebooks vindt u in de map Introductie Python.

Samengevat

Notebooks zijn digitale documenten die zowel uitvoerbare code bevatten als tekst, afbeeldingen, video, hyperlinks ...

De notebooks bestaan uit een opeenvolging met cellen: Markdown-cellen voor tekst en beeld en code-cellen om te programmeren.

Het programmeren gebeurt met Python 3.

De notebooks van KIKS zijn onderverdeeld in drie categorieën:

In de notebooks, Introductie Python, worden fundamentele zaken aangebracht, zoals een printopdracht en datastructuren.

De notebooks, Introductie Machine Learning, besteden aandacht aan standaard machine learning technieken.

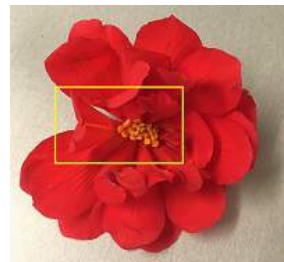
In de derde reeks notebooks, Introductie Deep Learning, wordt uitgelegd hoe een diep neuraal netwerk wordt opgebouwd. Deze reeks bevat een notebook om huidmondjes te laten tellen.

DIGITALE BEELDEN

9.1 *Pixel*

Een digitaal beeld bestaat uit een eindig aantal pixels. Het woord 'pixel' is afgeleid van de Engelse woorden *picture* (afbeelding) en *element*. Het woord 'digitaal' komt van het Latijnse *digitus* (vinger) en verwijst naar het tellen op de vingers. Het aantal pixels in een digitaal beeld kan geteld worden (Hautekiet et al., 2016).

Bekijk de foto van de begonia in Figuur 9.1. Men kan inzoomen op een bepaald deel van de foto. Als men genoeg inzoomt, dan kan men de pixels onderscheiden zoals in Figuur 9.2.



Figuur 9.1: Deel selecteren van de foto van een begonia.



Figuur 9.2: Inzoomen op deel begonia.

Digitale afbeelding

Een digitale afbeelding is een rechthoekig rooster van pixels.

Men spreekt ook van een rasterafbeelding of een bitmap. GIF, JPEG en PNG zijn bestandsformaten voor het opslaan van rasterafbeeldingen in digitale vorm.

9.2 Kleur van een pixel

Elke pixel heeft een bepaalde kleur:

- bij een zwart-witafbeelding is een pixel zwart of wit;
- bij een grijswaardenafbeelding is dat een grijswaarde;
- bij een kleuraanbeelding is dat een kleur.

Van de begonia hierboven werd een deel van een bloemblad afgepeld en onder de microscoop gelegd. Op het bloemblad bevonden zich enkele stomata. Met een smartphone werd een foto genomen door de microscoop. Op het deel van deze foto in Figuur 9.3 is een van de huidmondjes goed te zien.

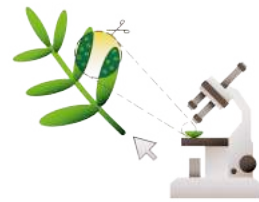
Om te kunnen begrijpen hoe een kleurenfoto digitaal is 'opgebouwd', moet eerst de opbouw van een grijswaardenfoto duidelijk zijn.

9.3 Grijswaarden

Bekijk dezelfde foto in grijswaarden en vergroot in Figuur 9.4. De pixels zijn nu zichtbaar.

Een grijswaarde wordt door de computer voorgesteld door een natuurlijk getal tussen 0 en 255. Hierbij komt 0 overeen met zwart en 255 met wit.

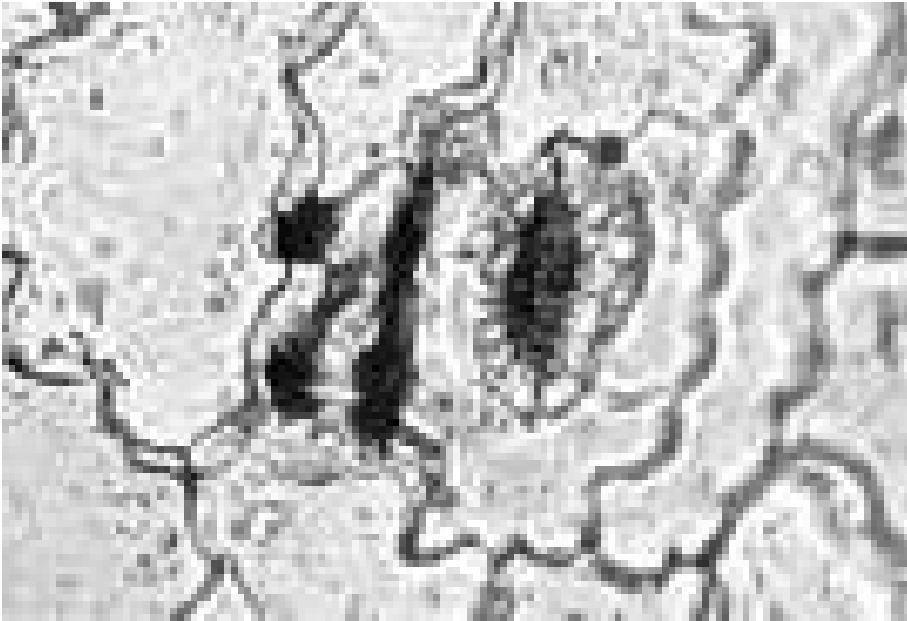
De waarden ertussen geven dus grijswaarden weer die stilaan lichter worden naarmate het getal stijgt. Dit wordt geïllustreerd in Figuur 9.5.



Ga zelf aan de slag met afbeeldingen in de notebooks 'Matrices en grijswaarden' en 'Tensoren en RGB'.



Figuur 9.3: Huidmondje op bloemblad rode begonia. Omdat dit een foto is van levend materiaal, is de foto mooi van kleur, in tegenstelling tot de foto's uit de dataset van KIKS. Die bestaat immers uit foto's van bladafdrukken van gedroogd materiaal.



Figuur 9.4: Foto van Figuur 9.3 omgezet naar grijswaarden.

Dit betekent dat men een afbeelding in grijswaarden kan voorstellen met een rechthoekig rooster van getallen tussen 0 en 255.

In de wiskunde spreekt men van een matrix waarvan de elementen natuurlijke getallen tussen 0 en 255 zijn. In hoofdstuk 10 wordt dieper ingegaan op de wiskunde achter digitale beelden.

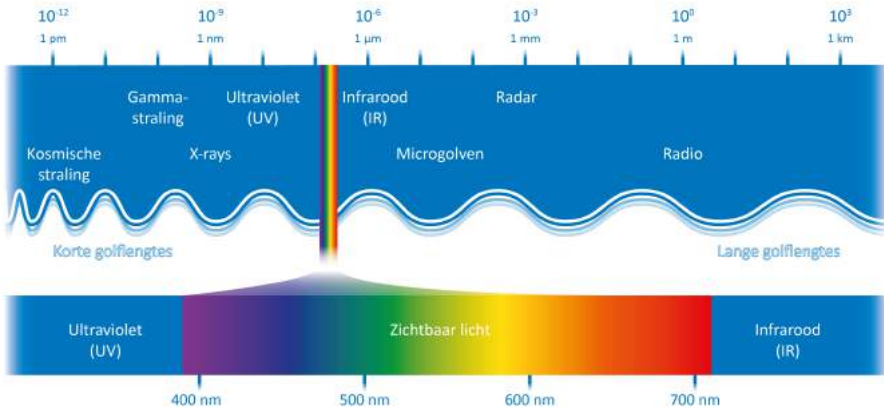
9.4 Kleuren

Elektromagnetische golven hebben een breed spectrum van golflengtes, waarbij elke golflengte overeenkomt met een andere kleur (zie Figuur 9.6). Het licht dat de mens kan zien, zichtbaar licht, beslaat slechts een klein deel van het spectrum. Het zichtbaar licht met de kleinste golflengtes is blauw, dat met de grootste

0	32	64
96	128	160
192	224	255

Figuur 9.5: Elk getal van 0 t.e.m. 255 komt overeen met een bepaalde grijswaarde.

golflengtes is rood.



Figuur 9.6: Elektromagnetische golven.

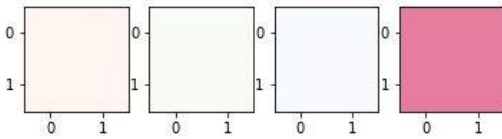
Door rood, groen en blauw licht samen te brengen, kan bijna elke kleur gesimuleerd worden. Kleurenfoto's kunnen dan ook in een RGB-systeem worden opgeslagen (RGB = rood, groen, blauw). In dat geval zijn er wel drie matrices nodig: één voor de rode tinten, één voor de groene tinten en één voor de blauwe tinten. Elk van die matrices bevat getallen tussen 0 en 255, waarbij de waarde van het element de intensiteit van de tint weergeeft. Deze drie roosters worden dan samengevoegd tot een soort balk van getallen. In de computerwetenschappen spreekt men van een 'tensor' (zie Figuur 9.7 en hoofdstuk 10 'Tensoren'). De volgorde is hier van belang. De computer interpreteert steeds de eerste matrix als de matrix van de roodtinten, de tweede als die van de groentinten en de laatste als die van de blauwtinten.

In Figuur 9.8 worden de beelden van drie gekozen matrices en de overeenkomstige tensor getoond. De rode tint heeft waarde 230, de groene tint 125 en de blauwe tint 156. De kleur van het laatste vierkant heeft dus als RGB-code 230, 125, 156.

Omgekeerd kunnen uit een kleurenfoto de rood-, blauw- en groentinten gedistilleerd worden, zoals bv. voor de foto van Figuur 9.3: de rood-, groen- en blauw-

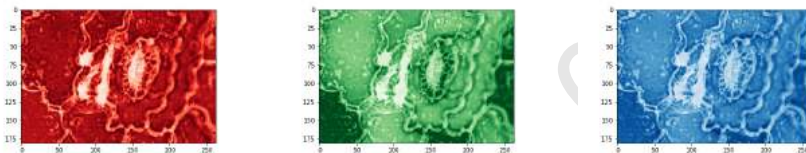


Figuur 9.7: Tensor van een kleurenafbeelding.



Figuur 9.8: Drie matrices vormen een tensor die een kleurenafbeelding representeert.

tinten worden weergegeven in Figuur 9.9. Hoe men dit doet, vindt u in paragraaf 10.9 en in de notebook 'Tensoren en RGB'.



Figuur 9.9: Rood-, groen- en blauwtinten in foto begonia van Figuur 9.3.

9.5 Type van de elementen van een tensor in Python

De matrices bevatten getallen tussen 0 en 255 die een grijswaarde of een intensiteit van een rode, groene of blauwe tint weergeven.

In NumPy kan men ervoor kiezen dat deze elementen het type `uint8` (*8 bits-unsigned integer*) hebben (zie notebook 'Extra datastructuren met NumPy'). Dan wordt voor de opslag van een element gebruikgemaakt van 8 bits, m.a.w. een byte (zie kader 'Bit en byte'). De natuurlijke getallen van 0 t.e.m. 255 voorgesteld in het tiendelige talstelsel, worden dan gerepresenteerd in het binaire talstelsel (zie kader 'Binaire talstelsel').

Soms worden echter ook waarden tussen 0 en 1 gebruikt i.p.v. tussen 0 en 255. Men kan dat doen door alle waarden te delen door 255. De tensor is dan genormaliseerd. Verhoudingsgewijs komt bij zo'n tensor 0 overeen met 0 en 1 met 255. Dat betekent ook dat de

10 is 00001010 en 145 is 10010001 als `uint8`.

Paragraaf 6.3 is interessant in deze context.

elementen dan een ander type hebben, het type float.

Bit en byte

Een 'bit' is een informatie-eenheid. De term is afkomstig van *binary digit*. Het is een eenheid die enkel de waarden 0 en 1 kan aannemen. Acht bits vormen samen een 'byte'. Er zijn 256 mogelijke combinaties mogelijk van 0 en 1 die samen een byte vormen.

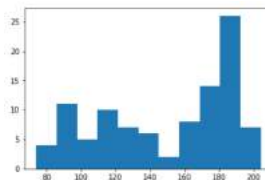
Binaire talstelsel

In het tiendelige talstelsel worden de getallen voorgesteld m.b.v. de cijfers 0 t.e.m. 9. Het binaire talstelsel gebruikt daarvoor enkel de cijfers 0 en 1. Zoals in het tiendelige talstelsel het getal $123 = 3 \cdot 10^0 + 2 \cdot 10^1 + 1 \cdot 10^2$, is in het binaire talstelsel het getal $101 = 1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2$, dus 5. Met elk cijfer in een getal komt dus als waarde een veelvoud van een macht van resp. 10 en 2 overeen. Het veelvoud wordt bepaald door het cijfer, de exponent van de macht door de plaats van het cijfer in het getal. Het binaire talstelsel heeft grondtal 2, het decimale talstelsel 10.

9.6 Verdeling van de kleuren

In Python kan je een verdeling van de kleuren of van de intensiteit van een tint opvragen in de vorm van een histogram. Door een foto goed te kiezen, kan men hiermee een normale of een scheve verdeling illustreren.

Men kan daarbij van elk getal van 0 t.e.m. 255 laten weergeven hoeveel keer het voorkomt of men kan het interval $[0,255]$ verdelen in deelintervallen en van elk deelinterval de frequentie visualiseren zoals in Figuur 9.10.



Figuur 9.10: Verdeling van grijswaarden.

9.7 Toepassing: verborgen boodschap

Door op een specifieke manier te 'spelen' met de waarden in de matrix of tensor die een afbeelding representeert, kan men er een boodschap in verbergen.

Een pixelwaarde met één verminderen of vermeerderen zal geen zichtbaar effect hebben op de afbeelding, maar men kan ermee wel het merendeel van de pixels bv. even maken. Als men ervoor zorgt dat de pixels waar de boodschap zich bevindt oneven zijn, dan kan men de boodschap onthullen met enkele lijntjes code.



Ga op zoek naar de verborgen boodschap in de notebook 'Verborgene boodschap'.

Samengevat

Een digitale afbeelding is een rechthoekig rooster van pixels. Elke pixel heeft een bepaalde kleur: bij een zwart-witafbeelding is een pixel zwart of wit, bij een grijswaardenafbeelding is dat een grijswaarde en bij een kleuraafbeelding is dat een kleur.

Grijswaarden worden door de computer voorgesteld door een natuurlijk getal tussen 0 en 255. Hierbij komt 0 overeen met zwart en 255 met wit. De waarden ertussen geven dus grijswaarden weer die stilaan lichter worden naarmate het getal stijgt. Een afbeelding in grijswaarden kan men voorstellen met een matrix waarvan de elementen natuurlijke getallen tussen 0 en 255 zijn.

Kleurenfoto's worden in een RGB-systeem opgeslagen. In dat geval zijn er drie matrices nodig die samen een tensor vormen. Deze matrices geven de intensiteit weer van respectievelijk de rode, groene en blauwe tinten aanwezig in de foto.

Vaak kiest men ervoor de matrix of tensor te normaliseren.

10.1 Matrices

De studie van matrices behoort tot de leerstof van de derde graad in richtingen met 6 uur wiskunde; matrices worden ook als keuzeonderwerp behandeld in richtingen met 3 of 4 uur wiskunde. Matrices behoren tot het domein van de lineaire algebra.

Matrix

Een matrix is een rechthoekig rooster van getallen.

Terminologie:

Bij een reële $m \times n$ -matrix A zijn alle elementen van A reële getallen. Zo een matrix A heeft m rijen en n kolommen. Men noemt $m \times n$ de 'dimensie' van de matrix A en men noteert:

$$A \in \mathbb{R}^{m \times n}$$

Het element op de i -de rij en de j -de kolom van de matrix A noteert men als:

$$a_{ij} \text{ met } a_{ij} \in \mathbb{R}$$

$$B = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

is een 3×2 -matrix.

$$B \in \mathbb{R}^{3 \times 2}$$

$$b_{31} = 5$$

Een matrix met maar één rij noemt men een rijmatrix.
Een kolommatrix is een matrix met slechts één kolom.

Een matrix met evenveel rijen als kolommen is een vierkante matrix. Een vierkante matrix heeft twee diagonalen. De hoofddiagonaal is de diagonaal van linksboven naar rechtsonder.

$$R = \begin{bmatrix} 1 & 2 & 6 & -2 \end{bmatrix}$$

is een 1×4 -rijmatrix.

$$R \in \mathbb{R}^{1 \times 4}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

is een vierkante matrix.

10.2 Eigenschappen van matrices

Twee matrices zijn gelijk als al hun overeenkomstige elementen gelijk zijn. Deze matrices hebben dan ook dezelfde dimensie.

Een nulmatrix is een matrix waarvan alle elementen 0 zijn. Een nulmatrix wordt genoteerd als O .

Een vierkante matrix waarvan de elementen op de hoofddiagonaal 1 zijn en de andere elementen allemaal 0, is een eenheidsmatrix en noteert men als I .

Een vierkante matrix $A \in \mathbb{R}^{n \times n}$ is symmetrisch als de elementen voldoen aan:

$$a_{ij} = a_{ji}, \forall i, j = 1, \dots, n$$

$$O_{4 \times 2} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

is een nulmatrix.

$$I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

is een eenheidsmatrix.

$$S = \begin{bmatrix} 2 & 6 & -3 \\ 6 & 0 & 5 \\ -3 & 5 & 3 \end{bmatrix}$$

is symmetrisch.

10.3 Bewerkingen met matrices

10.3.1 Het transponeren

Men kan een matrix transponeren. D.w.z. dat de rijen kolommen worden en de kolommen rijen, waarbij de positie blijft behouden: bv. de tweede rij wordt de tweede kolom, de vijfde kolom wordt de vijfde rij.

De getransponeerde matrix van een matrix A noteert men met A^T .

Als men een kolommatrix transposeert, bekomt men een rijmatrix.

Voor een symmetrische matrix A geldt dat $A^T = A$.

$$\begin{bmatrix} 1 & 2 & -2 & -4 & 3 \\ 3 & 4 & 5 & 6 & -2 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ -2 & 5 \\ -4 & 6 \\ 3 & -2 \end{bmatrix}$$

$$A \in \mathbb{R}^{m \times n} \implies A^T \in \mathbb{R}^{n \times m}$$

$$A^T = C, c_{ij} = a_{ji}$$

$$\forall i = 1, \dots, n, \forall j = 1, \dots, m$$

10.3.2 Optelling

Twee matrices met dezelfde dimensie kan men optellen. Hierbij worden alle overeenkomstige elementen opgeteld.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} -1 & -2 \\ -3 & -4 \\ -5 & -6 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

Het resultaat is een matrix met opnieuw dezelfde dimensie.

10.3.3 Scalaire vermenigvuldiging

Men kan een matrix vermenigvuldigen met een scalar. Hierbij wordt elk element van de matrix met de scalar vermenigvuldigd.

Een scalar is niets anders dan een getal.

Het resultaat is een matrix met dezelfde dimensie.

$$3 \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 9 & 12 \\ 15 & 18 \end{bmatrix}$$

10.3.4 Vermenigvuldiging

Men kan twee matrices slechts vermenigvuldigen als hun dimensies aan een bepaalde voorwaarde voldoen: het aantal kolommen van de eerste matrix moet gelijk zijn aan het aantal rijen van de tweede matrix. Hun product is opnieuw een matrix.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} -1 \\ 2 \end{bmatrix} = \begin{bmatrix} 3 \\ 5 \\ 7 \end{bmatrix}$$

Voor $M \in \mathbb{R}^{m \times n}$ en $N \in \mathbb{R}^{n \times q}$ is

$$M \cdot N = P \in \mathbb{R}^{m \times q}.$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 & 1 & 0 \\ 2 & 3 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 & -3 & 2 \\ 5 & 12 & -5 & 4 \\ 7 & 18 & -7 & 6 \end{bmatrix}$$

Hierbij worden de elementen van P gegeven door:

$$p_{ij} = \sum_{k=1}^n m_{ik} \cdot n_{kj}.$$

De vermenigvuldiging van matrices is niet commutatief!

De volgende eigenschap is het onthouden waard:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 5 & 12 \end{bmatrix}$$

Eigenschap

Voor $A \in \mathbb{R}^{m \times n}$ en $B \in \mathbb{R}^{n \times q}$ is

$$(A \cdot B)^T = B^T \cdot A^T.$$

$$\begin{bmatrix} -1 & 0 \\ 2 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = \begin{bmatrix} -1 & -2 \\ 11 & 16 \end{bmatrix}$$

Bewijs.

$$A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{n \times q}, A \cdot B = Q \in \mathbb{R}^{m \times q}$$

$$Q^T = S \in \mathbb{R}^{q \times m}$$

$$A^T = C \in \mathbb{R}^{n \times m}, B^T = D \in \mathbb{R}^{q \times n}$$

$$D \cdot C = R \in \mathbb{R}^{q \times m}$$

Te bewijzen: $S = R$.

R en S hebben dezelfde dimensie $q \times m$.

Voor alle $i = 1, \dots, q$ en $j = 1, \dots, m$:

$$\begin{aligned} s_{ij} &= q_{ji} \\ &= \sum_{k=1}^n a_{jk} \cdot b_{ki} \\ &= \sum_{k=1}^n c_{kj} \cdot d_{ik} \\ &= \sum_{k=1}^n d_{ik} \cdot c_{kj} \\ &= r_{ij} \end{aligned}$$

□

10.4 Aan de slag met matrices in Python

In Python wordt een matrix ingegeven via een NumPy-array. Daarvoor moet eerst de module NumPy geïmporteerd worden. Meer uitleg over modules en NumPy vindt u in hoofdstuk 8 'Python en Jupyter Notebook'.

$$\text{Beschouw } A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \text{ en } B = \begin{bmatrix} 3 & 1 \\ -1 & 6 \\ 2 & -2 \end{bmatrix}.$$

Deze matrices geeft men in in Python als volgt:

```
1 >>> matrix_A = np.array([[1,2,3],
2                             [4,5,6]])
3 >>> matrix_B = np.array([[3,1],
4                             [-1,6],
```

In de code wordt geteld vanaf 0. Zo is de eerste kolom in de code de kolom met index 0. De matrix B met drie rijen wordt in Python gerepresenteerd door een NumPy-array waarbij rij 0, rij 1 en rij 2 worden ingegeven.

```
5 [2, -2]])
```

Het element a_{13} op de eerste rij en derde kolom van A vindt men in Python met:

```
1 >>> matrix_A[0][2]
2 3
```

a_{13} is het element op de rij met index 0 en de kolom met index 2.

De volledige tweede kolom van A vindt men met:

```
1 >>> matrix_A[:, 1]
2 array([2, 5])
```

Neem van elke rij van A het element dat behoort tot de kolom met index 1.

De tweede en de derde rij van B verkrijgt men met de instructie:

```
1 >>> matrix_B[1:, :]
2 array([[ -1,  6],
3        [ 2, -2]])
```

Neem van elke rij van B , te beginnen met rij 1, de elementen van alle kolommen.

De eerste en de tweede kolom van A verkrijgt men via:

```
1 >>> matrix_A[:, 0:2]
2 array([[1, 2],
3        [4, 5]])
```

$0 : 2$ staat voor $[0, 2[$, dus hier kolom 0 en kolom 1.

`>>>matrix_A[:, :2]` geeft hetzelfde resultaat.

Men kan in Python de matrix B transponeren:

```
1 >>> matrix_B.T
2 array([[3, -1, 2],
3        [1, 6, -2]])
```



Ga interactief aan de slag met de notebooks 'Extra datastructuren met NumPy' en 'Tensoren'.

De som van de matrix A met de getransponeerde van de matrix B vindt men als volgt:

```
1 >>> matrix_A + matrix_B.T
2 array([[4, 1, 5],
3        [5, 11, 4]])
```

Dit is enkel mogelijk omdat A en B^T dezelfde dimensie hebben.

Men kan de matrix B (scalair) vermenigvuldigen met 4:

```

1 >>> 4 * matrix_B
2 array([[12,  4],
3        [-4, 24],
4        [ 8, -8]])

```

Het product van de matrices A en B wordt in Python berekend via de instructie:

```

1 >>> matrix_A.dot(matrix_B)
2 array([[7,  7],
3        [19, 22]])

```

Dit is enkel mogelijk omdat het aantal kolommen van A gelijk is aan het aantal rijen van B .

10.5 Tensoren

Beschouw een matrix die geen rij- of kolommatrix is. Ze heeft dus meerdere rijen en kolommen. Men zegt dat zo'n matrix twee 'assen' heeft: een as voor de rijen en een as voor de kolommen. Een matrix noemt men daarom ook een 2D-tensor.

Tensor

Een tensor is een uitbreiding van het matrix-begrip naar meer of minder 'assen'.

Een scalair is een 0D-tensor. Het is een getal, maar kan ook geïnterpreteerd worden als een 1×1 -matrix (een matrix met slechts één element). Een 0D-tensor heeft geen assen.

$3 \in \mathbb{R}$ is een scalair.

Een vector, waarvan de coördinaat n elementen bevat, is een 1D-tensor met één as. Een vector kan ook genoteerd worden als een rijmatrix ($\in \mathbb{R}^{1 \times n}$). Een rijmatrix is dus ook een 1D-tensor.

$$(2, -3, 5, 6, 0) \in \mathbb{R}^5$$

is de coördinaat van een vector in een 5-dimensionale vectorruimte.

Als men bijvoorbeeld een 'samenstelling' zou maken van drie 4×5 -matrices, door de drie matrices naast elkaar te plaatsen (zie Figuur 10.4), dan komt er een derde as bij. Die samenstelling T is dan een 3D-tensor. De dimensie van de matrices is 4×5 . De dimensie van de tensor T is $3 \times 4 \times 5$.

In de literatuur noemt men de n bij een n D-tensor meestal de dimensie van de tensor. Dit kan verwarring scheppen. Bij KIKS zullen we het daarom vaak expliciet over de wiskundige dimensie hebben. De 2D, 3D, enz. kan men vergelijken met de dimensies in meetkunde, 2D voor het vlak, 3D voor de ruimte ...

$$T \in \mathbb{R}^{3 \times 4 \times 5}$$

Let wel: de matrices die samengesteld worden, moeten allemaal dezelfde dimensie hebben.

Zet men bv. tien 5×7 -matrices naast elkaar, dan is het resultaat een 3D-tensor met dimensie $10 \times 5 \times 7$. Dit wordt geïllustreerd in Figuren 10.1 en 10.2.

Merk op dat een 3D-tensor een driedimensionaal raster van getallen is (zie Figuur 10.3). De tensor $T \in \mathbb{R}^{3 \times 4 \times 5}$ kan dus ook gezien worden als vijf 3×4 -matrices op elkaar gestapeld. Zie Figuur 10.4.

Stelt men een aantal even grote 3D-tensoren samen, dan krijgt men een 4D-tensor. Stelt men een aantal even grote 4D-tensoren samen, dan krijgt men een 5D-tensor, enz.

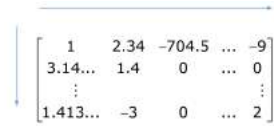
Voorbeeld: $T \in \mathbb{R}^{n \times p \times q \times r \times s}$ is een 5D-tensor waarbij n 4D-tensoren worden samengesteld, waarbij elke 4D-tensor behoort tot $\mathbb{R}^{p \times q \times r \times s}$. De 4D-tensoren zijn op hun beurt alle een samenstelling van p 3D-tensoren met elke 3D-tensor behorend tot $\mathbb{R}^{q \times r \times s}$. De 3D-tensoren tot slot zijn alle een samenstelling van q matrices met r rijen en s kolommen. De tensor T heeft vijf assen.

10.6 Bewerkingen met tensoren

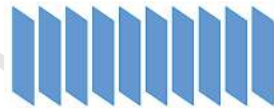
De bewerkingen op matrices, worden uitgebreid voor tensoren.

10.6.1 Optelling

Het optellen van tensoren werkt net als bij matrices. De overeenkomstige elementen worden opgeteld. De tensoren moeten even groot zijn om ze te kunnen optellen.



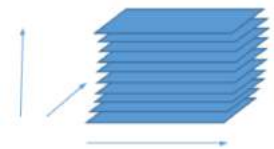
Figuur 10.1: Matrix in $\mathbb{R}^{5 \times 7}$.



Figuur 10.2: Tensor in $\mathbb{R}^{10 \times 5 \times 7}$.



Figuur 10.3: Matrix in $\mathbb{R}^{5 \times 7}$.



Figuur 10.4: Tensor in $\mathbb{R}^{10 \times 5 \times 7}$.

10.6.2 Vermenigvuldiging

Het vermenigvuldigen van tensoren is een uitbreiding van de vermenigvuldiging van matrices naar meer assen.

10.6.3 Functies die inwerken op een tensor

Op een tensor kan men elementsgewijs een functie laten inwerken. Het resultaat is dan een tensor met dezelfde dimensie als de oorspronkelijke tensor.

Vierkantswortel-functie Men neemt de vierkantswortel van elk element in de tensor.

```
1 >>> np.sqrt(X)
```

Sign-functie Men bepaalt het teken van elk element in de tensor. Is het strikt positief, dan wordt het element vervangen door 1, is het strikt negatief, dan wordt het element vervangen door -1. 0 wordt afgebeeld op zichzelf.

```
1 >>> np.sign(X)
```

Heaviside-functie Men bepaalt het teken van elk element in de tensor. Is het positief, dan wordt het element vervangen door 1, is het strikt negatief, dan wordt het element vervangen door 0. De drempelwaarde 0 wordt aan de functie als argument meegegeven.

```
1 >>> np.heaviside(X,0)
```

ReLU Men bepaalt het teken van elk element in de tensor. Is het positief, dan wordt het element behouden, is het strikt negatief, dan wordt het element vervangen door 0. In de Python-module Keras is ReLU voorhanden. In NumPy moet men ReLU wel nog zelf definiëren, doe dat als volgt:

De rol van activatiefunctie in een neuronaal netwerk is weggelegd voor dergelijke functies. Dat wordt uitgelegd in hoofdstuk 11.

$$A = \begin{bmatrix} 16 & 2 & 4 & 0 \\ 0 & 9 & 5 & 25 \end{bmatrix}$$

np.sqrt(A) levert

$$\begin{bmatrix} 4 & \sqrt{2} & 2 & 0 \\ 0 & 3 & \sqrt{5} & 5 \end{bmatrix}$$

$$B = \begin{bmatrix} -16 & 2 & -4 & 0 \\ 0 & 9 & 5 & -25 \end{bmatrix}$$

np.sign(B) levert

$$\begin{bmatrix} -1 & 1 & -1 & 0 \\ 0 & 1 & 1 & -1 \end{bmatrix}$$

np.heaviside(B,0) levert

$$\begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

relu(B) levert

$$\begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 9 & 5 & 0 \end{bmatrix}$$

```
1 >>> relu(X) = np.maximum(X, 0)
```

10.7 Aan de slag met tensoren in Python

Een matrix is een tensor met twee assen. Net zoals een matrix wordt een tensor via een NumPy-array ingegeven in Python.

Beschouw een n D-tensor tensor.

Met Python-instructies kan men bepaalde kenmerken van deze tensor opvragen:

<code>tensor.shape</code>	wiskundige dimensie
<code>tensor.ndim</code>	waarde van n
<code>len(tensor)</code>	aantal elementen



Ga interactief aan de slag met de notebook 'Tensoren'.

Een 0D-tensor

In wiskunde is dat bv. de 1×1 -matrix $\begin{bmatrix} 12 \end{bmatrix}$.

Een 0D-tensor aanmaken in Python:

```
1 >>> x = np.array(12)
2 >>> print(x)
3 12
4 >>> x.ndim
5 0
6 >>> x.shape
7 ()
```

`()` komt overeen met $(1,1)$ en de (wiskundige) dimensie 1×1 .

Een 1D-tensor

In wiskunde is dat bv. de 1×4 -rijmatrix $\begin{bmatrix} 12 & -3 & 6 & 14 \end{bmatrix}$ of het 4-tal $(12, -3, 6, 14)$.

Een 1D-tensor aanmaken in Python:

```
1 >>> x = np.array([12, -3, 6, 14])
2 >>> print(x)
3 [12 -3  6 14]
4 >>> x.ndim
```

Het viertal kan de coördinaat zijn van een vector of een punt.

`(4,)` komt overeen met $(1,4)$ en de (wiskundige) dimensie 1×4 .

```

5 1
6 >>> x.shape
7 (4,)
```

Een 2D-tensor

In wiskunde is dat bijvoorbeeld de volgende 2×4 -matrix:

$$\begin{bmatrix} 12 & -3 & 6 & 14 \\ 1 & 0 & 2.33 & -7.4545 \end{bmatrix}.$$

Een 2D-tensor aanmaken in Python:

```

1 >>> x = np.array([[12, -3, 6, 14],
2                   [1, 0, 2.33, -7.4545]])
3 >>> print(x)
4 [[ 12.    -3.     6.    14.   ]
5  [  1.     0.    2.33  -7.4545]]
6 >>> x.ndim
7 2
8 >>> x.shape
9 (2,4)
```

(2,4) is de (wiskundige) dimensie 2×4 .

Een 3D-tensor

In wiskunde is dat bv. een samenstelling van twee 3×4 -

matrices $\begin{bmatrix} 2 & 3 & 4 & 7 \\ 1 & 5 & 6 & 8 \\ 2 & 1 & 0 & 0 \end{bmatrix}$ en $\begin{bmatrix} 0 & -4 & 2 & 2 \\ -1 & 7 & 7 & 0 \\ -2.5 & 1.5 & 0 & 0 \end{bmatrix}$.

Een 3D-tensor aanmaken in Python:

```

1 >>> x = np.array([[[2, 3, 4, 7],
2                   [1, 5, 6, 8],
3                   [2, 1, 0, 0]],
4                   [[0, -4, 2, 2],
5                   [-1, 7, 7, 0],
6                   [-2.5, 1.5, 0, 0]])
7 >>> print(x)
8 [[[ 2.   3.   4.   7. ]
9  [  1.   5.   6.   8. ]
10 [  2.   1.   0.   0. ]
11 [[ 0.  -4.   2.   2.]
12 [-1.   7.   7.   0.]
13 [-2.5  1.5   0.   0.]
```

```

10 [ 2.  1.  0.  0. ]
11
12 [[ 0. -4.  2.  2. ]
13  [-1.  7.  7.  0. ]
14  [-2.5 1.5  0.  0. ]]]
15 >>> x.ndim
16 3
17 >>> x.shape
18 (2, 3, 4)

```

10.8 MNIST dataset

De MNIST dataset bestaat uit 70 000 afbeeldingen van handgeschreven cijfers, elk 28 x 28 pixels in grijswaarden (zie Figuren 10.5 en 10.7). Hiervan worden er 60 000 gebruikt om een neurale netwerk te trainen en 10 000 om het te testen. Elke afbeelding is voorzien van een label. Dat label is een cijfer tussen 0 en 9. Omdat er 70 000 afbeeldingen zijn, zijn er dus ook 70 000 labels.

De trainingset wordt ingevoerd in het netwerk in de vorm van een 3D-tensor.

Immers: elke afbeelding komt overeen met een matrix met 28 rijen en 28 kolommen. Elk element van de matrix komt overeen met een pixel. Het element is gelijk aan de grijswaarde van de pixel, dit is een natuurlijk getal dat behoort tot het interval $[0, 255]$. De 60 000 matrices worden samengesteld tot een 3D-tensor.

De trainingset wordt dus onder de vorm van een tensor met dimensie $60000 \times 28 \times 28$ aan het netwerk gegeven. Deze tensor bevat maar liefst 983 040 000 elementen.

De MNIST trainingset

Stel dat tensor de tensor is van de trainingset.

Dan geeft `tensor.shape` als resultaat: $(60\,000, 28, 28)$.

Het is dus een samenstelling van 60 000 matrices.

`tensor[5]` geeft de zesde matrix van deze tensor te-

(2,3,4) komt overeen met de (wiskundige) dimensie $2 \times 3 \times 4$ van een samenstelling van twee 3×4 -matrices of een stapel van vier 2×3 -matrices.

De MNIST dataset is een dataset die werd gegenereerd door het 'National Institute of Standards and Technology' (NIST) door de cijfers in te scannen. De 'M' staat voor *modified*; de dataset werd aangepast om beter bruikbaar te zijn voor machine learning algoritmes (Goodfellow et al., 2016).



Figuur 10.5: Een '2' uit de MNIST dataset.

rug en komt dus overeen met de zesde afbeelding in de dataset van handgeschreven cijfers.

Men kan uit de tensor een 'subtensor' van matrices halen.

`tensor[10:100]` geeft de elfde matrix t.e.m. de honderdste matrix van deze tensor terug en komt dus overeen met 90 afbeeldingen uit de dataset van handgeschreven cijfers (NumPy werkt hier met het halfopen interval `[10, 100[`, `tensor[100]` wordt dus niet teruggegeven).

```
1 >>> tensor[10:100].shape
2 (90, 28, 28)
```

Men kan ook een 'balkvormig deel' selecteren uit de stapel matrices (zie Figuur 10.6).

```
1 >>> tensor[:, 2:10, 3:7]
```

Van alle matrices in de stapel wordt van de derde t.e.m. de tiende rij het deel bekeken van de vierde t.e.m. de zevende kolom. Dit betekent dat van alle afbeeldingen een gelijk gepositioneerd stuk wordt bekeken.

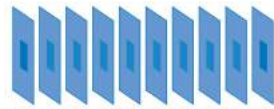
10.8.1 De opbouw van de MNIST dataset

Als men de MNIST dataset wilt bekijken dan kan men dat bv. doen via de Python-module Keras. De dataset kan daar ingeladen worden:

```
1 >>> from keras.datasets import mnist
2 >>> data = mnist.load_data()
```

Men kan opvragen welk type deze dataset heeft. Blijkbaar is het een koppel, want het is een tuple dat twee elementen bevat.

```
1 >>> type(data)
2 <class 'tuple'>
3 >>> len(data)
4 2
```



Figuur 10.6: Een balkvormig deel selecteren.

Voor meer uitleg over de module Keras kunt u terecht in hoofdstuk 14.



Leer over tuples in de notebook 'Datastructuur'.

Blijkbaar zijn beide elementen van dit koppel zelf koppels.

```
1 >>> print(type(data[0]), type(data[1]))
2 <class 'tuple'> <class 'tuple'>
3 >>> len(data[0])
4 2
5 >>> len(data[1])
6 2
```

De vraag blijft natuurlijk wat er in die twee koppels zit.

```
1 >>> print(data[0][0].shape,
2         data[0][1].shape,
3         data[1][0].shape,
4         data[1][1].shape)
5 (60000, 28, 28) (60000,) (10000, 28, 28)
   (10000,)
```

De dataset bevat dus (indirect) 4 tensoren: twee 3D-tensoren en twee 1D-tensoren.

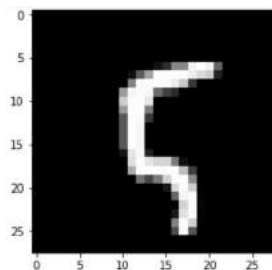
Men kan ook specifieke elementen uit deze tensoren opvragen, bv. van het honderdeneerste element van de eerste 1D-tensor:

```
1 >>> print(data[0][1][100, :, :])
2 5
```

Gezien de context van de dataset - het gaat om grijs-waardenafbeeldingen van handgeschreven cijfers, en het feit dat zulke afbeeldingen door een 3D-tensor kunnen worden gerepresenteerd - verdient het een poging om een element van de eerste 3D-tensor op te vragen als een afbeelding:

```
1 >>> import matplotlib.pyplot as plt
2 >>> plt.imshow(data[0][0][100, :, :],
3               cmap="gray")
```

Men krijgt de afbeelding in Figuur 10.7 te zien. Nu is het duidelijk: de ingeladen MNIST dataset is niets anders dan twee koppels, waarvan het eerste koppel bestaat uit



Figuur 10.7: Een vijf uit de MNIST dataset.

de 60 000 trainingafbeeldingen met hun respectievelijke labels en het tweede koppel gevormd wordt door de 10 000 testafbeeldingen met hun labels.

10.9 Kleurenafbeelding

In de notebook 'Tensoren en RGB' wordt een kleurenfoto ingelezen:

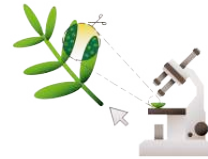
```
1 >>> begonia_RGB = plt.imread("begonia.jpg")
2 >>> begonia_RGB.shape
3 (181, 263, 3)
4 >>> begonia_RGB.ndim
5 3
```

Deze kleurenfoto komt dus overeen met een 3D-tensor met dimensie $181 \times 263 \times 3$ (zie Figuur 10.8). Dat betekent dat de tensor bestaat uit een stapel van drie 181×263 -matrices, een voor de rode, een voor de groene en een voor de blauwe tinten. Een pixel van een kleurenfoto wordt immers weergegeven in RGB-code (zie hoofdstuk 9 'Pixels en RGB').

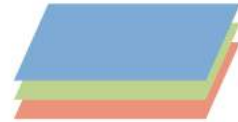
Men kan de matrices van de rode, groene en blauwe tinten apart opvragen. Men moet daarvoor wel het juiste deel van de tensor selecteren. De tensor heeft dimensie $181 \times 263 \times 3$; men spreekt van een tensor met 181 rijen, 263 kolommen en 3 kanalen. De kanalen, die verwijzen naar de kleuren, staan achteraan. Rood is het eerste kanaal, dus men krijgt de matrix van de roodtinten voor kanaal 0.

```
1 >>> begonia_R = begonia_RGB[:, :, 0]
2 >>> begonia_G = begonia_RGB[:, :, 1]
3 >>> begonia_B = begonia_RGB[:, :, 2]
```

Of men kan drie matrices die respectievelijk de intensiteit van de rode, groene en blauwe tinten van



Bekijk dit in de notebook 'Tensoren en RGB'.



Figuur 10.8: 3D-tensor als representatie van een kleurenfoto.

Die 3D-tensor kan ook bekeken worden als een tensor die bestaat uit een samenstelling van 181 matrices met dimensie 263×3 .

een kleurenfoto weergeven, stapelen tot de tensor die de kleurenfoto representeert. Men kan hier dus van `begonia_R`, `begonia_G` en `begonia_B` terug de oorspronkelijke foto maken. De functie `dstack()` is hiervoor voorzien in de module NumPy.

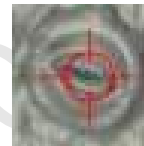
```
1 >>> begonia_RGB = np.dstack([begonia_R,
2                               begonia_G,
3                               begonia_B])
```

10.10 KIKS-dataset

De KIKS-dataset bestaat uit 84 000 kleurenafbeeldingen van 120 x 120 pixels. Daarom kan men ze met een 4D-tensor aan het KIKS-netwerk te geven.

Immers, één afbeelding komt overeen met een stapel van drie 120x120-matrices (een voor de R, een voor de G en een voor de B) en wordt dus gerepresenteerd door een 3D-tensor. 84 000 afbeeldingen betekent 84 000 van deze 3D-tensoren samenstellen en levert een 4D-tensor op met dimensie $84000 \times 120 \times 120 \times 3$.

Een andere dataset met afbeeldingen in kleur zal ook gebruikmaken van een 4D-tensor.



Figuur 10.9: Voorbeeld stoma.

Als er onvoldoende geheugen is, is het echter niet mogelijk de data in een keer aan het netwerk te geven. Dan zal men de data verdelen en geleidelijk aan in het netwerk invoeren.

Samengevat

Een tensor is een uitbreiding van het matrixbegrip. Een tensor is een raster van getallen. Een 2D-tensor kan gebruikt worden om grijswaardenafbeeldingen te representeren, een 3D-tensor om een kleurenafbeelding voor te stellen. De datasets van KIKS en van MNIST kunnen in de computer worden ingevoerd met een 4D-tensor.

Op tensoren kunnen, net als op matrices, bewerkingen worden uitgevoerd.

Leestip

Wes McKinney. Python for Data Analysis. Data Wrangling with pandas, NumPy, and IPython (McKinney, 2018).

François Chollet. Deep learning with Python (Chollet, 2018).

Versie 1.0

BASISCONCEPTEN VAN MACHINAAL LEREN IN DE PRAKTIJK

11.1 *Misconceptions*

De artificiële intelligentie die nu gebruikt wordt, staat zeer ver af van *general AI*. Het gaat om *machine learning*-systemen, gericht op een bepaalde taak, al dan niet geïmplementeerd in een robot. Via onze smartphone en het internet speelt AI een, soms verdoken, rol in ons leven. Overheid, justitie, veiligheidsdiensten, banken en verzekeringsfirma's, zoekrobots, socialemediaplatformen ... laten hun werking steeds meer afhangen van zogenaamde geautomatiseerde beslissingen, m.a.w. beslissingen geautomatiseerd met *machine learning*. AI heeft zo steeds meer impact op onze maatschappij, en de manier waarop neurale netwerken getraind worden kan, als men hier niet bedacht op is, vooroordelen en stereotypen aanwezig in onze maatschappij, versterken.

Over machine learning-systemen bestaan veel misverstanden. Omdat ze o.a. gebaseerd zijn op wiskundige principes, wordt verkeerdelijk gedacht dat deze systemen 100 % objectief beslissingen nemen. Een deep reinforcement learning-systeem dat in staat is de regerende schaakkampioen te verslaan, wordt al snel

Voor meer uitleg over general AI, zie paragraaf 5.1.

Zie ook de kader 'Bias' in paragraaf 5.6.

Reinforcement learning wordt uitgelegd in hoofdstuk 6.

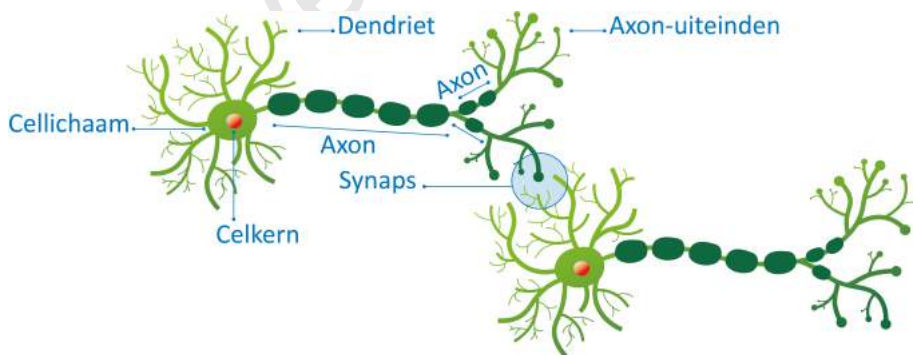
geacht veel slimmer te zijn dan die mens. De naam 'neurale netwerken' wekt de indruk dat deze systemen op dezelfde manier zouden werken als het menselijk brein.

11.2 Biologisch neuron

Een mens heeft ongeveer honderd miljard zenuwcellen of neuronen. De meeste bevinden zich in de hersenen of het ruggenmerg. Neuronen worden ook wel de bouwstenen van onze hersenen genoemd.

Een neuron bestaat uit dendrieten, een cellichaam en een axon. Het cellichaam bevat een celkern. Het axon mondt uit in de axoneinden en kan eventueel nog enkele vertakkingen hebben. Zoals elke cel is een neuron omgeven door een membraan. U ziet een afbeelding van twee neuronen in Figuur 11.1.

Bij een neuron in rust zijn er aan de buitenkant van de membraan meer positieve ionen dan aan de binnenkant, en aan de binnenkant zijn er meer negatieve ionen dan aan de buitenkant. Dit geeft een potentiaalverschil over de celmembraan: de rustpotentialiaal.



Figuur 11.1: Biologische neuronen.

Een neuron ontvangt prikkels: via de dendrieten krijgt het signalen van andere neuronen.

Als een neuron geprikkeld wordt, kunnen er zich, afhankelijk van de sterkte van de prikkel, meer of minder ionen verplaatsen door de membraan. Als het totaal aan signalen sterk genoeg is, verplaatsen er zich zoveel ionen door de membraan dat het potentiaalverschil voldoende gestegen is om een elektrische impuls te doen ontstaan. In dat geval is het neuron geëxciteerd. Sommige signalen werken de geëxciteerde toestand meer in de hand, andere temperen de excitatie. Men zou kunnen zeggen dat sommige verbindingen tussen neuronen sterker zijn dan andere.

Dus eens het potentiaalverschil een bepaalde drempelwaarde overschrijdt, veroorzaakt deze zogenaamde actiepotentiaal een elektrische impulsgeleiding: een elektrische impuls van het cellichaam door het axon naar de axoneinden. Aan het axoneinde komen neurotransmitters vrij, die op membraanreceptoren van een volgend neuron binden, waardoor in dat neuron nu ook een elektrische impulsgeleiding kan ontstaan. M.a.w. het neuron stuurt op zijn beurt via het axon een signaal naar andere neuronen.

Na de impuls keert het neuron terug in rust.

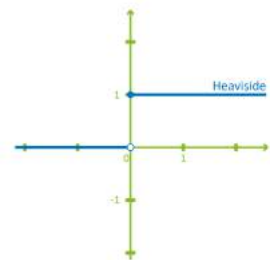
11.3 Artificieel neuron

In 1943 ontwierpen McCulloch en Pitts een artificieel neuron, zoals voorgesteld in Figuur 11.3 (Haykin, 2009). De functie die er gebruikt wordt, is een drempelwaarde-functie, de Heaviside-functie H .

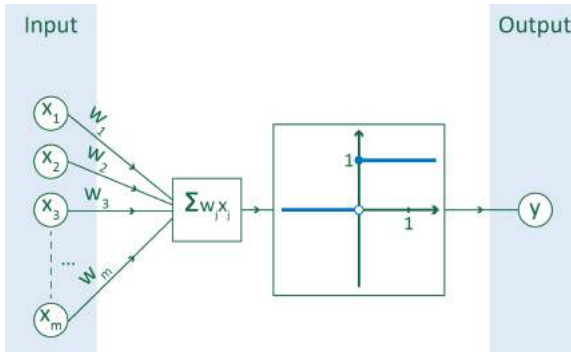
De Heaviside-functie wordt als volgt gedefinieerd:

$$H : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} 0, & \text{als } x < 0 \\ 1, & \text{als } x \geq 0. \end{cases}$$

De inputwaarden die het neuron ontvangt van andere neuronen, worden gewogen opgeteld; het gewicht bepaalt a.h.w. de sterkte van de verbinding tussen de



Figuur 11.2: Heaviside-functie.



Figuur 11.3: Artificieel neuron.

neuronen. De som wordt onderworpen aan de niet-lineaire Heaviside-functie H . Als de som de drempelwaarde 0 bereikt of overschrijdt, krijgt het outputneuron y waarde 1, anders krijgt het waarde 0. De Heaviside-functie noemt men een activatiefunctie en ze bepaalt de toestand y van het neuron. Net als het biologische neuron, geëxciteerd of niet geëxciteerd, kent het artificiële neuron twee toestanden.

Frank Rosenblatt ging hierop verder in 1958. Hij bouwde een systeem opdat een artificeel neuron zou kunnen leren: het Perceptron, het eerste neurale netwerk.

Het biologische neuron diende voor wetenschappers zoals McCulloch, Pitts en Rosenblatt als inspiratie voor de ontwikkeling van de theorie rond neurale netwerken, maar het contrast met de werking van het artificiële neuron is groot: in de hersenen gebeuren de processen continu en parallel; bij het artificiële neuron gebeuren die discreet en serieel. De neurale netwerken die hierna aan bod komen, hebben niet de bedoeling de structuur van het menselijk brein te imiteren.

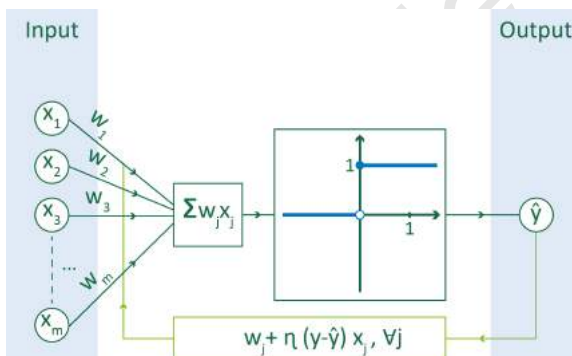
11.4 Perceptron

Het Perceptron is een neurale netwerk met twee lagen: een inputlaag en een outputlaag. De neuronen

van de inputlaag zijn verbonden met het neuron in de outputlaag.

Het Perceptron beschikt over een algoritme om te leren. Het wordt getraind met gelabelde voorbeelden: een aantal inputpunten X_i met telkens een corresponderende, verwachte output y_i , het label. Tussen de neuronen van de input- en outputlaag zijn er verbindingen met een bepaald gewicht. Gebaseerd op de gelabelde voorbeelden worden die gewichten gaandeweg aangepast.

Het Perceptron wordt schematisch voorgesteld in Figuur 11.4. Hierbij is $X_i(x_1, x_2, x_3, \dots, x_m)$ met $x_j \in \mathbb{R}$ voor $j = 1, \dots, m$ en $y_i, \hat{y}_i \in \{0, 1\}$, $i = 1, \dots, n$. Het systeem kiest eerst willekeurige waarden voor de gewichten $w_1, w_2, w_3, \dots, w_m$. Naast de n inputpunten X_i , de n labels y_i en de n outputwaarden \hat{y}_i is er dus ook nog een vector \vec{W} : $\vec{W}(w_1, w_2, w_3, \dots, w_m)$, met $w_j \in \mathbb{R}, j = 1, \dots, m$.



Figuur 11.4: Het Perceptron.

Voor elk voorbeeld wordt nagegaan of de klasse \hat{y}_i die het systeem voorspelt, wel overeenkomt met het gegeven label y_i . Indien \hat{y}_i en y_i niet overeenkomen, dan worden de waarden van de gewichten aangepast. Men zegt dat het systeem leert.

Men vermeerderd daartoe elke w_j met een waarde Δw_j waarbij Δw_j afhangt van het verschil tussen y_i en \hat{y}_i , van x_j en van de leersnelheid (*learning rate*). De

De waarde van w_j wordt aangepast naar $w_j + \eta(y_i - \hat{y}_i)x_j$.

leersnelheid is een getal, gekozen tussen 0 en 1, dat eigenlijk bepaalt hoe bruusk of hoe voorzichtig de gewichten aangepast worden; de leersnelheid zal dus mee bepalen hoe snel het systeem leert. De leersnelheid wordt genoteerd met de Griekse letter η (èta).

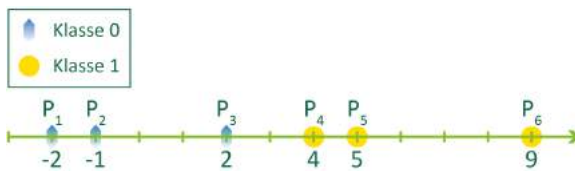
Het systeem zal de gegeven voorbeelden meerdere keren doorlopen tot voor alle voorbeelden het juiste label wordt voorspeld. Per keer dat alle voorbeelden doorlopen worden, spreekt men van een *epoch*.

Onthoud: de gelabelde voorbeelden en de learning rate worden gekozen door de mens, niet door het neurale netwerk.

De learning rate is een hyperparameter van het netwerk. U komt er meer over te weten in paragraaf 11.7, paragraaf 11.12 en in de kader 'Waarom werkt het Perceptron-algoritme?'.

11.5 Meer effectiviteit voor het Perceptron

Beschouw het voorbeeld uit Figuur 11.5.



Figuur 11.5: Punten die behoren tot twee klassen. De abscis x van elk punt is af te lezen op de figuur.

Voor dit voorbeeld is het interessanter om 3 als de drempelwaarde van de activatiefunctie te nemen, i.p.v. 0. Immers: de punten rechts van 3 behoren tot een andere klasse dan de punten links van 3.

De activatiefunctie zou dan zijn:

$$A : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} 0, & \text{als } x < 3 \\ 1, & \text{als } x \geq 3. \end{cases}$$

Om het Perceptron meer effectief te maken zou men i.p.v. met de Heaviside-functie dus met een andere drempelwaardefunctie kunnen werken.

In de praktijk zal men het probleem echter anders be-

naderen. Men kan immers ook het volgende schrijven:

$$A : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} 0, & \text{als } x - 3 < 0 \\ 1, & \text{als } x - 3 \geq 0. \end{cases}$$

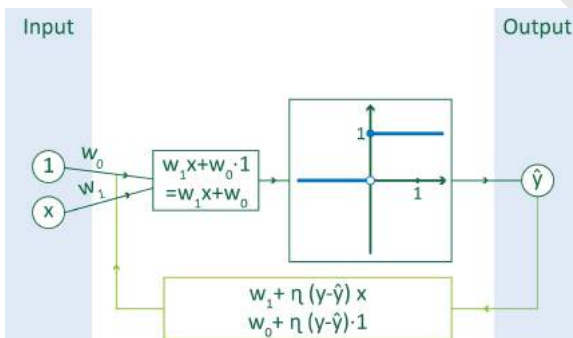
Het is duidelijk dat $A(x) = H(x - 3)$, i.p.v. x moet je dan $x - 3$ aan de Heaviside-functie geven om de output te bekomen. $x - 3$ is een gewogen som van x en 1 met gewichten resp. 1 en -3 . In de wiskunde noemt men dat een lineaire combinatie van x en 1:

$$x - 3 = 1 \cdot x + (-3) \cdot 1.$$

Deze praktische benadering kan worden voorgesteld zoals in Figuur 11.6. Men voegt een (input)neuron toe met waarde 1 (dit toegevoegd neuron wordt genoteerd als x_0) en een extra gewicht (genoteerd als w_0).

$(-3) \cdot 1 + 1 \cdot x = w_0 x_0 + w_1 x_1$ met $w_0 = -3$ en $w_1 = 1$. Men noemt die 3 de *bias* van de invoerlaag van het netwerk.

Onthoud dat een scheiding met vergelijking $x = 3$ overeenkomt met een gewicht $w_0 = -3$.



Figuur 11.6: Neuraal netwerk met 2 inputneuronen. Het extra neuron x_0 speelt een cruciale rol.

Men werkt dan met input $X_i(1, x)$ met $x_0 = 1, x_1 = x \in \mathbb{R}$, label $y_i \in \{0,1\}$, output $\hat{y}_i \in \{0,1\}$ voor $i = 1, \dots, n$ en $\hat{W}(w_0, w_1)$, met $w_j \in \mathbb{R}, j = 0, 1$.

Voor het punt P_1 uit Figuur 11.5 is de input $X_1(1, -2)$ en het label $y_1 = 0; x_0 = 1$ en $x_1 = -2$.

De berekening die wordt uitgevoerd door het netwerk, kan worden voorgesteld in matrixgedaante.

Stel dat $X = \begin{bmatrix} 1 \\ x \end{bmatrix}$ en $W = \begin{bmatrix} -3 \\ 1 \end{bmatrix}$. Dan is

Hoe men rekt met matrices wordt getoond in hoofdstuk 10.

$$W^T \cdot X = \begin{bmatrix} -3 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x \end{bmatrix} = \begin{bmatrix} -3 \cdot 1 + 1 \cdot x \end{bmatrix} = \begin{bmatrix} x - 3 \end{bmatrix}.$$

Dit wordt gevolgd door $H(x - 3)$ met als resultaat \hat{y} .

Of meer algemeen:

$$X = \begin{bmatrix} 1 \\ x \end{bmatrix}, W = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix},$$

$$W^T \cdot X = \begin{bmatrix} w_0 & w_1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x \end{bmatrix}$$

$$= \begin{bmatrix} w_0 \cdot 1 + w_1 \cdot x \end{bmatrix},$$

gevolgd door $H(w_0 + w_1 x) = \hat{y}$.

De waarde van \hat{y} hangt af van het feit of $w_0 + w_1 x > 0$ of $w_0 + w_1 x < 0$ is. Dus de vergelijking $w_0 + w_1 x = 0$ bepaalt de scheiding.

11.6 1D-voorbeeld van de training van het Perceptron

Herneem Figuur 11.5 met punten die behoren tot twee klassen.

Er zijn zes voorbeelden waar het Perceptron uit kan leren: -2 uit klasse 0, -1 uit klasse 0, 2 uit klasse 0, 4 uit klasse 1, 5 uit klasse 1 en 9 uit klasse 1. Het is de mens die deze gelabelde voorbeelden kiest.

Bijvoorbeeld voor het punt P_1 geldt: $x = -2$ en de klasse $y = 0$, m.a.w. $x_0 = 1, x_1 = -2$ en $y = 0$.

Er worden nu twee mogelijke scenario's besproken, elk met een gekozen leersnelheid, en waarbij vertrokken wordt van bepaalde waarden voor de gewichten w_0 en w_1 .

Noteer alvast de waarden van deze variabelen voor de andere punten.

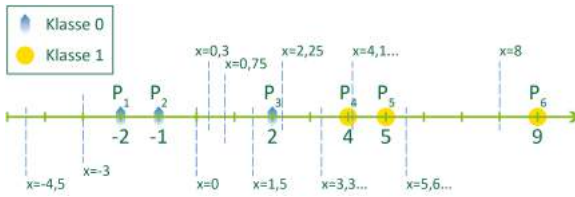
11.6.1 $\eta = 0,1$ en $w_0 = -8, w_1 = 1$

De learning rate wordt vastgelegd op $\eta = 0,1$.

Het systeem initialiseert de waarden van de elementen van \vec{W} op een willekeurige manier, bv. $w_0 = -8$ en $w_1 = 1$. De voorlopige scheiding ligt dan op 8 en heeft dus als vergelijking: $x = 8$. Hierdoor zullen slechts 4 punten aan de juiste kant van de scheiding liggen, nl. P_1, P_2, P_3 en P_6 (zie Figuur 11.7).

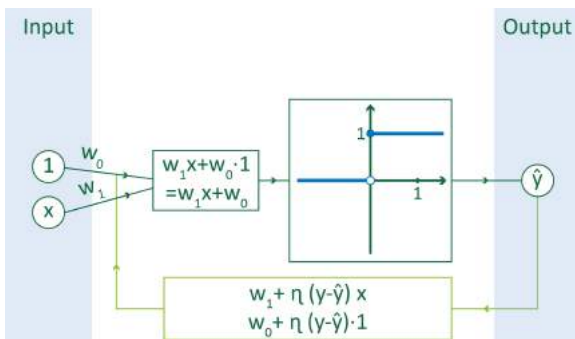
Zie paragraaf 11.4, paragraaf 11.7 en Figuren 11.4 en 11.6 voor de formules waarin de learning rate wordt toegepast.

Het systeem zal de punten bij meerdere pogingen,



Figuur 11.7: Bij scheiding op $x = 8$ zijn vier punten correct geclassificeerd.

epochs, in het neurale netwerk invoeren, zoals geïllustreerd in Figuur 11.8.



Figuur 11.8: Neuraal netwerk met 2 inputneuronen. De gewichten w_0 en w_1 zijn initieel $w_0 = -8$ en $w_1 = 1$. Steeds $x_0 = 1$ en $x_1 = x$.

Het systeem begint aan een eerste epoch. Het overloopt elk voorbeeldpunt.

P_1 met $y = 0, x_0 = 1, x_1 = -2$:

$$-2 \xrightarrow{w_1 x_1 + w_0} 1 \cdot (-2) + (-8) = -10 \xrightarrow{-10 < 0} 0$$

dus $\hat{y} = 0$ en $\hat{y} = y$.

De gewichten moeten niet worden aangepast.

Reken na dat ook voor P_2 en P_3 de gewichten niet moeten worden aangepast.

P_4 met $y = 1, x_0 = 1, x_1 = 4$:

$$4 \xrightarrow{w_1 x_1 + w_0} 1 \cdot 4 + (-8) = -4 \xrightarrow{-4 < 0} 0$$

dus $\hat{y} = 0$ en $\hat{y} \neq y$.

De gewichten moeten wel worden aangepast.

Het gewicht w_j wordt vervangen door

$$w_j + \eta \cdot (y - \hat{y}) \cdot x_j,$$

Hoe groter de fout, met name het verschil tussen het label y en de voorspelde waarde \hat{y} , hoe groter de aanpassing die men zal doen.

dus

$$w_0 = -8 + 0,1 \cdot (1 - 0) \cdot 1 \Leftrightarrow w_0 = -8 + 0,1 \Leftrightarrow w_0 = -7,9$$

en

$$w_1 = 1 + 0,1 \cdot (1 - 0) \cdot 4 \Leftrightarrow w_1 = 1 + 0,4 \Leftrightarrow w_1 = 1,4.$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking:

$$1,4x - 7,9 = 0 \Leftrightarrow x = 5,642\dots$$

P_5 met $y = 1, x_0 = 1, x_1 = 5$:

$$5 \xrightarrow{w_1 x_1 + w_0} 1,4 \cdot 5 + (-7,9) = -0,9 \xrightarrow[-0,9 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} \neq y$.

De gewichten moeten opnieuw worden aangepast.

$$w_0 = -7,9 + 0,1 \cdot (1 - 0) \cdot 1 \Leftrightarrow w_0 = -7,9 + 0,1 \Leftrightarrow w_0 = -7,8$$

en

$$w_1 = 1,4 + 0,1 \cdot (1 - 0) \cdot 5 \Leftrightarrow w_1 = 1,4 + 0,5 \Leftrightarrow w_1 = 1,9.$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking:

$$1,9x - 7,8 = 0 \Leftrightarrow x = 4,157\dots$$

Reken na dat P_6 bij de juiste klasse wordt ingedeeld, dus voor P_6 moeten de gewichten niet worden aangepast.

Alle voorbeelden zijn nu één keer doorlopen. Men zegt dat de eerste epoch voltooid is.

De vergelijking $w_0 + w_1 x = 0$ bepaalt de scheiding.

Nu onderneemt het systeem een tweede epoch. Het doorloopt opnieuw alle voorbeelden, te beginnen met P_1 .

$$-2 \xrightarrow{w_1 x_1 + w_0} 1,9 \cdot (-2) + (-7,8) = -11,6 \xrightarrow[-11,6 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} = y$.

P_1 wordt bij de juiste klasse ingedeeld.

Reken na dat dat ook geldt voor P_2 en P_3 .

P_4 :

$$4 \xrightarrow{w_1 x_1 + w_0} 1,9 \cdot 4 + (-7,8) = -0,2 \xrightarrow[-0,2 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} \neq y$.

De gewichten moeten nogmaals worden aangepast.

$$w_0 = -7,8 + 0,1 \cdot (1 - 0) \cdot 1 \Leftrightarrow w_0 = -7,8 + 0,1 \Leftrightarrow w_0 = -7,7$$

en

$$w_1 = 1,9 + 0,1 \cdot (1 - 0) \cdot 4 \Leftrightarrow w_1 = 1,9 + 0,4 \Leftrightarrow w_1 = 2,3.$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking:

$$2,3x - 7,7 = 0 \Leftrightarrow x = 3,3478\dots$$

P_5 wordt nu correct geclassificeerd. P_6 ook. Dus de tweede epoch zit erop.

De derde epoch. Alle voorbeeldpunten worden juist geclassificeerd. Het systeem is erin geslaagd de punten van elkaar te scheiden en de uiteindelijke scheiding heeft als vergelijking:

$$x = 3,3478\dots$$

11.6.2 $\eta = 0,3$ en $w_0 = 0, w_1 = 1$

Doe nog eens hetzelfde maar met learning rate $\eta = 0,3$ en andere beginwaarden voor de gewichten. De gewichten w_0 en w_1 worden nu geïnitieerd op resp. 0 en

1. De voorlopige scheiding heeft dan als vergelijking: $x = 0$. Dat betekent dat slechts één punt aan de verkeerde kant van de voorlopige scheiding ligt, nl. het punt P_3 .

De vergelijking $w_0 + w_1x = 0$ bepaalt de scheiding.

Eerste epoch. Reken na dat P_1 en P_2 aan de juiste klasse toegekend worden.

P_3 :

$$2 \frac{w_1x+w_0}{2} \rightarrow 2 + 0 = 2 \frac{H}{2>0} \rightarrow 1 \text{ dus } \hat{y} = 1 \text{ en } \hat{y} \neq y.$$

De gewichten moeten worden aangepast.

$$w_0 = 0 + 0,3 \cdot (0 - 1) \cdot 1$$

$$\Leftrightarrow w_0 = 0 - 0,3$$

$$\Leftrightarrow w_0 = -0,3$$

en

$$w_1 = 1 + 0,3 \cdot (0 - 1) \cdot 2$$

$$\Leftrightarrow w_1 = 1 - 0,6$$

$$\Leftrightarrow w_1 = 0,4$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar rechts! De voorlopige scheiding heeft nu als vergelijking: $x = 0,75$.

Reken na dat P_4 , P_5 en P_6 aan de juiste klasse toegekend worden. De eerste epoch is beëindigd.

Tweede epoch. De eerste foute klasse is opnieuw bij P_3 . Het gewicht w_0 wordt aangepast naar $-0,6$ en het gewicht w_1 wordt aangepast naar $-0,2$. De voorlopige scheiding heeft nu als vergelijking: $x = -3$. De scheiding is naar de verkeerde kant opgeschoven.

Ook P_4 wordt fout ingedeeld. Het gewicht w_0 wordt aangepast naar $-0,3$ en het gewicht w_1 wordt aangepast naar 1. De voorlopige scheiding heeft nu als vergelijking: $x = 0,3$. De scheiding is naar de goede kant opgeschoven.

P_5 en P_6 worden aan de juiste klasse toegekend. De tweede epoch is beëindigd.

Derde epoch. De eerste foute klasse is er bij P_3 . Het gewicht w_0 wordt aangepast naar $-0,6$ en het gewicht w_1 wordt aangepast naar $0,4$. De voorlopige scheiding heeft nu als vergelijking: $x = 1,5$. De scheiding is opnieuw naar de goede kant opgeschoven.

P_4 , P_5 en P_6 worden aan de juiste klasse toegekend. De derde epoch is beëindigd.

Vierde epoch. De eerste foute klasse is er opnieuw bij P_3 . Het gewicht w_0 wordt aangepast naar $-0,9$ en het gewicht w_1 wordt aangepast naar $-0,2$. De voorlopige scheiding heeft nu als vergelijking: $x = -4,5$. De scheiding is opnieuw naar de verkeerde kant opgeschoven.

Ook P_4 wordt fout geclassificeerd. Het gewicht w_0 wordt aangepast naar $-0,6$ en het gewicht w_1 wordt aangepast naar 1 . De voorlopige scheiding heeft nu als vergelijking: $x = 0,6$. De scheiding is naar de goede kant opgeschoven.

P_5 en P_6 worden aan de juiste klasse toegekend. De vierde epoch is beëindigd.

Vijfde epoch. En alweer loopt het fout bij P_3 . Het gewicht w_0 wordt aangepast naar $-0,9$ en het gewicht w_1 wordt aangepast naar $0,4$. De voorlopige scheiding heeft nu als vergelijking: $x = 2,25$. De scheiding is opnieuw naar de goede kant opgeschoven.

P_4 , P_5 en P_6 worden aan de juiste klasse toegekend. De vijfde epoch is beëindigd.

Zesde epoch. Alle voorbeeldpunten worden correct geclassificeerd. Het systeem is erin geslaagd de punten van elkaar te scheiden en de uiteindelijke scheiding heeft als vergelijking: $x = 2,25$.

Eigenlijk wordt hier steeds eenzelfde berekening herhaald. Zo'n repetitieve handeling is uitermate geschikt om te automatiseren, dus te laten uitvoeren door een

computer. Er wordt hiervoor een algoritme opgesteld dat wordt geprogrammeerd.

11.7 Het Perceptron-algoritme

Het Perceptron-algoritme kan toegepast worden om lineair scheidbare data te classificeren. Het is dus maar beperkt bruikbaar. In de praktijk zal het meestal niet mogelijk zijn alle data in de juiste klasse in te delen. Als men voor zulke data die niet lineair scheidbaar zijn, toch kiest voor dit algoritme om de data te scheiden, dan zal het algoritme blijven doorgaan. Men heeft dan te maken met een oneindige lus.

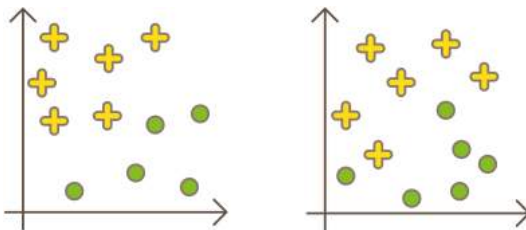
Er zullen immers steeds punten zijn waarvoor een foute voorspelling wordt gemaakt.

BEGIN

HERHAAL tot alle punten bij de juiste klasse worden ingedeeld
 VOOR ELK PUNT met corresponderend label in de dataset
 VERGELIJK het label met de voorspelling
 ALS de voorspelling fout is DAN
 pas de gewichten \vec{W} aan naar $\vec{W} + \eta(y - \hat{y}) \cdot \vec{X}$
 ANDERS (dus als de voorspelling juist is) DOE
 niets

EINDE

Figuur 11.9 toont het verschil tussen data die lineair scheidbaar zijn en data die dat niet zijn.



Figuur 11.9: Wel of niet lineair scheidbaar?

11.8 2D-voorbeeld van het Perceptron: classificatie met de Iris dataset

De Iris dataset werd in 1936 door de Brit Ronald Fischer gepubliceerd in 'The use of multiple measurements in taxonomic problems' (Fischer, 1936; Dua & Taniskidou, 2017). Figuur 11.10 bevat een deel van die tabel. De dataset betreft drie soorten irissen: *Iris setosa*, *Iris virginica* en *Iris versicolor* (zie Figuren 11.11, 11.12 en 11.13).

<i>Iris setosa</i>				<i>Iris versicolor</i>				<i>Iris virginica</i>			
Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width	Sepal length	Sepal width	Petal length	Petal width
5.1	3.5	1.4	0.2	7.0	3.2	4.7	1.4	6.3	3.3	6.0	2.5
4.9	3.0	1.4	0.2	6.4	3.2	4.5	1.5	5.8	2.7	5.1	1.9
4.7	3.2	1.3	0.2	6.9	3.1	4.9	1.5	7.1	3.0	5.9	2.1
4.6	3.1	1.5	0.2	5.5	2.3	4.0	1.3	6.3	2.9	5.6	1.8
5.0	3.6	1.4	0.2	6.5	2.8	4.6	1.5	6.5	3.0	5.8	2.2
5.4	3.9	1.7	0.4	5.7	2.8	4.5	1.3	7.6	3.0	6.6	2.1
4.6	3.4	1.4	0.3	6.3	3.3	4.7	1.6	4.9	2.5	4.5	1.7
5.0	3.4	1.5	0.2	4.9	2.4	3.3	1.0	7.3	2.9	6.3	1.8
4.4	2.9	1.4	0.2	6.6	2.9	4.6	1.3	6.7	2.5	5.8	1.8
4.9	3.1	1.5	0.1	5.2	2.7	3.9	1.4	7.2	3.6	6.1	2.5
5.4	3.7	1.5	0.2	5.9	2.9	3.5	1.0	6.5	3.2	5.1	2.0
4.8	3.4	1.6	0.2	5.9	3.0	4.2	1.5	6.4	2.7	5.3	1.9
4.8	3.0	1.4	0.1	6.0	3.2	4.0	1.0	6.8	3.0	5.5	2.1
4.3	3.0	1.1	0.1	6.1	2.9	4.7	1.4	5.7	2.5	5.0	2.0
5.8	4.0	1.2	0.2	5.6	2.9	3.6	1.3	5.8	2.8	5.1	2.4

Figuur 11.10: Originele tabel van Fischer van de Iris dataset.



Figuur 11.11:
Iris setosa
(Binek, 2019).



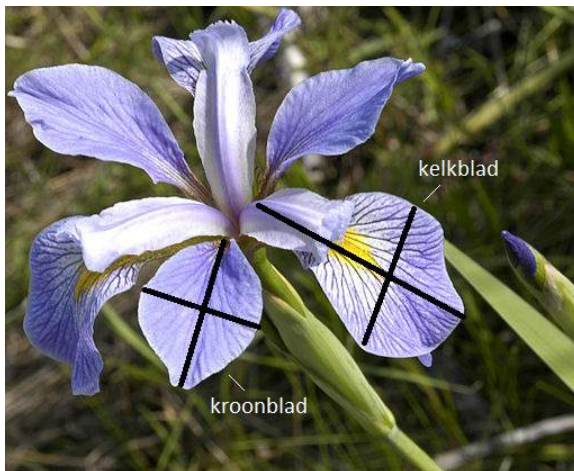
Figuur 11.12:
Iris virginica
(Mayfield, 2007).



Figuur 11.13:
Iris versicolor
(Langlois, 2019).

Fischer kon de soorten van elkaar onderscheiden afgaande op vier kenmerken: de lengte en de breedte van de kelkbladen en de kroonbladen (zie Figuur 11.14).

De Iris dataset is een multivariate dataset, d.w.z. een dataset met meerdere variabelen, die van elke soort vijftig monsters bevat. Deze variabelen zijn de lengte van een kroonblad, de breedte van een kroonblad, de lengte van een kelkblad en de breedte van een kelkblad. Van elk monster werden de lengte en de breedte van



Figuur 11.14: Kroon- en kelkbladen.

een kroonblad en een kelkblad opgemeten in centimeter (zie Figuren 11.10 en 11.15).

	lengte kelkblad	breedte kelkblad	lengte kroonblad	breedte kroonblad	soort iris
0	5.1	3.5	1.4	0.2	<i>Iris setosa</i>
1	4.9	3.0	1.4	0.2	<i>Iris setosa</i>
2	4.7	3.2	1.3	0.2	<i>Iris setosa</i>
3	4.6	3.1	1.5	0.2	<i>Iris setosa</i>
4	5.0	3.6	1.4	0.2	<i>Iris setosa</i>

Figuur 11.15: Tabel Fischer uit notebook 'Classificatie met de Iris dataset'.

In de notebook 'Classificatie met de Iris dataset' worden twee soorten irissen met het Perceptron geïdentificeerd op basis van twee kenmerken.

De twee klassen kunnen lineair van elkaar worden gescheiden. Er wordt getoond welk effect de aanpassingen op de gewichten hebben op de ligging van de scheidsrechte.

De onderzoeksvraag die gesteld wordt in deze notebook, is: kunnen de *Iris setosa* en de *Iris versicolor* van elkaar onderscheiden worden op basis van twee kenmerken? Er worden van elk van de twee soorten



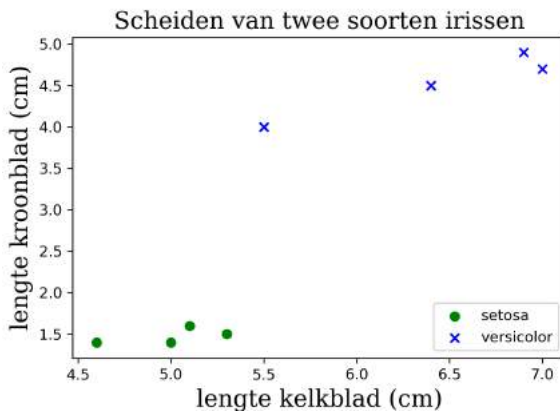
Bekijk de notebook 'Classificatie met de Iris dataset'.

vier monsters gekozen als gelabelde voorbeelden (zie Figuur 11.16).

	lengte kelkblad	breedte kelkblad	lengte kroonblad	breedte kroonblad	soort Iris
46	5.1	3.8	1.6	0.2	Iris setosa
47	4.6	3.2	1.4	0.2	Iris setosa
48	5.3	3.7	1.5	0.2	Iris setosa
49	5.0	3.3	1.4	0.2	Iris setosa
50	7.0	3.2	4.7	1.4	Iris versicolor
51	6.4	3.2	4.5	1.5	Iris versicolor
52	6.9	3.1	4.9	1.5	Iris versicolor
53	5.5	2.3	4.0	1.3	Iris versicolor

Figuur 11.16: Tabel selectie uit de Iris dataset uit de notebook 'Classificatie met de Iris dataset.

Het eerste voorbeeld in de notebook behandelt de onderzoeksvraag met de kenmerken lengte kelkblad en lengte kroonblad. De twee kenmerken worden weergegeven door x_1 en x_2 , de soort iris door het label y : label $y = 0$ (voor de *setosa*) en label $y = 1$ (voor de *versicolor*). Elk datapunt is dus een punt (x_1, x_2) met



Figuur 11.17: Puntenwolk, twee soorten irissen.

een corresponderend label y . Men kan de datapunten weergeven in een puntenwolk zoals in Figuur 11.17. De twee klassen zijn lineair scheidbaar.

Voordat men machine learning-technieken gaat toepassen op de data, zal men de data eerst standaardiseren

of normaliseren.

Dat levert een grafische voorstelling op van de data dat een goed beeld geeft van de correlatie tussen de variabelen en kleinere getallen om mee te rekenen.

Omdat men werkt met twee kenmerken, zijn er al twee inputneuronen. Opnieuw voegt men een extra neuron toe. Er moeten nu dus drie gewichten worden geïnitieerd (zie Figuur 11.18).

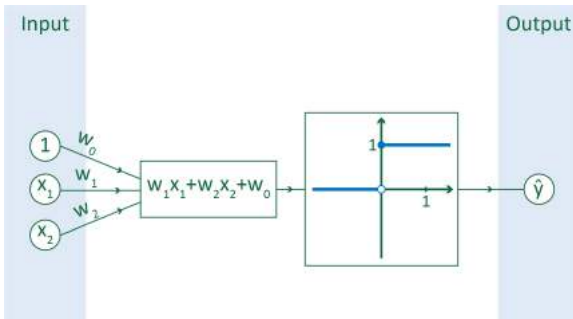
Men werkt dan met input $X_i(1, x_1, x_2)$ met $x_0 = 1$, $x_1, x_2 \in \mathbb{R}$, label $y_i \in \{0,1\}$, output $\hat{y}_i \in \{0,1\}$ voor $i = 1, \dots, 8$ en $\vec{W}(w_0, w_1, w_2)$, met $w_j \in \mathbb{R}$, voor $j = 0, 1, 2$.

Meer uitleg over standaardiseren en normaliseren vindt u in paragraaf 6.3.



Bekijk ook de notebook 'Standaardiseren'.

Paragraaf 11.5 legt het nut van het extra neuron uit.



Figuur 11.18: Netwerk met drie inputneuronen.

Door het invoeren van het extra neuron x_0 en het extra gewicht, de bias w_0 , kunnen de bewerkingen uit Figuur 11.18 eenvoudig in matrixgedaante weergegeven worden.

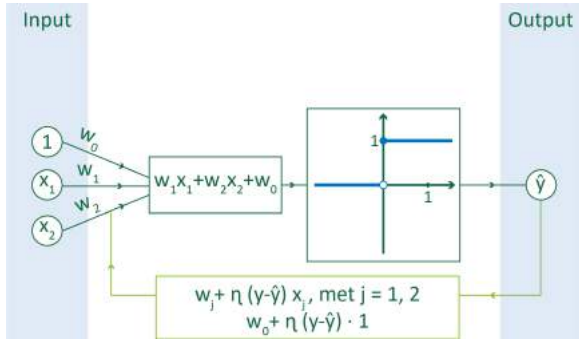
Voor $X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}$ en $W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$, is

$$W^T \cdot X = \begin{bmatrix} w_0 & w_1 & w_2 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix},$$

gevolgd door $H(w_0 + w_1x_1 + w_2x_2) = \hat{y}$.

De gewichten worden aangepast op basis van de 8 ge-

labelde voorbeelden: het netwerk leert. Het Perceptron-neuraal netwerk kan voorgesteld worden zoals in Figuur 11.19. De werkwijze, nu in twee dimensies, is dezelfde als daarnet in één dimensie (paragraaf 11.6). Het verschil is dat er nu drie gewichten zijn om aan te passen i.p.v. twee.



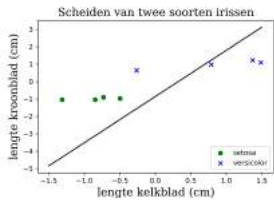
Figuur 11.19: Perceptron-neuraal netwerk met drie inputneuronen.



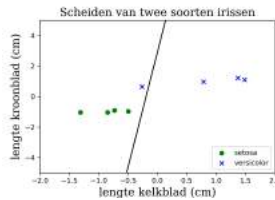
De grafieken in de Figuren 11.20, 11.21 en 11.22 bevatten gestandaardiseerde waarden.

Na initialisatie van de gewichten is de voorlopige scheidingsrechte bv. de rechte uit Figuur 11.20. In de loop van de training 'verplaatst' de rechte zich.

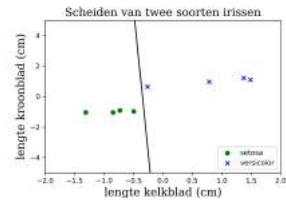
Bekijk de animatie in de notebook 'Classificatie met de Iris dataset'.



Figuur 11.20: Initialisatie van de scheidingslijn.



Figuur 11.21: Rechte in de loop van de training.



Figuur 11.22: Scheidingsrechte na training.

Na enige epochs bekomt men de rechte uit Figuur 11.21. Na de training bekomt men een scheiding zoals in Figuur 11.22.

Waarom werkt het Perceptron-algoritme?

Scalair product

$w_1x_1 + w_2x_2 + w_0$ kan men interpreteren als het 'scalair product' van twee driedimensionale vectoren $\vec{W}(w_0, w_1, w_2)$ en $\vec{X}(1, x_1, x_2)$:

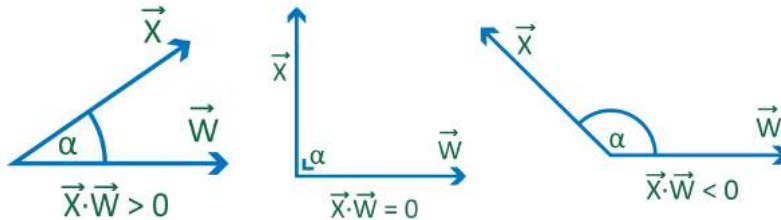
$$\begin{aligned}\vec{W} \cdot \vec{X} &= w_0 \cdot 1 + w_1 \cdot x_1 + w_2 \cdot x_2 \quad \text{in analytische gedaante,} \\ \vec{W} \cdot \vec{X} &= \|\vec{W}\| \cdot \|\vec{X}\| \cos(\angle \vec{W}, \vec{X}) \quad \text{in meetkundige gedaante.}\end{aligned}$$

Opschuiven naar de juiste klasse

Als \vec{W} en \vec{X} loodrecht staan op elkaar, is $\vec{W} \cdot \vec{X} = 0$.

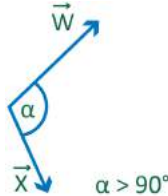
Als de hoek tussen de vectoren scherp is, is $\vec{W} \cdot \vec{X} > 0$ en als de hoek tussen de vectoren stomp is, is $\vec{W} \cdot \vec{X} < 0$.

Als men deze vectoren in hetzelfde punt laat aangrijpen, dan liggen hun representanten in eenzelfde vlak.



Als een punt wordt ingedeeld bij de foute klasse, dan worden de gewichten, dus \vec{W} , aangepast.

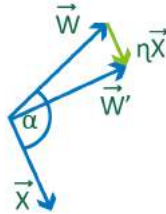
Stel dat (x_1, x_2) behoort tot de klasse met label 1. Dan is het vereist dat $\vec{W} \cdot \vec{X} > 0$ om het punt (x_1, x_2) in te delen bij de juiste klasse. Indien de hoek tussen de vectoren echter stomp is en de situatie uit Figuur 11.23 zich voordoet, dan is $\vec{W} \cdot \vec{X} < 0$ en wordt het punt (x_1, x_2) ingedeeld bij de verkeerde klasse. M.a.w. $y = 1, \hat{y} = 0, y - \hat{y} = 1$ en $\eta(y - \hat{y}) > 0$ want $\eta > 0$.



Figuur 11.23: Stompe hoek tussen beide vectoren.

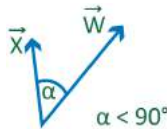
Als \vec{W} aangepast wordt naar $\vec{W} + \eta(y - \hat{y}) \cdot \vec{X}$, dus hier naar $\vec{W} + \eta \vec{X}$,

dan wordt de hoek tussen \vec{W} en \vec{X} kleiner (zie Figuur 11.24). Er wordt a.h.w. opgeschoven naar de juiste klasse.



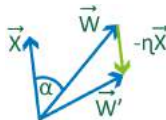
Figuur 11.24: Opschuiven naar klasse 1.

Stel dat (x_1, x_2) behoort tot de klasse met label 0. Dan zou $\vec{W} \cdot \vec{X} < 0$ moeten zijn om het punt (x_1, x_2) in te delen bij de juiste klasse. Indien de hoek tussen de vectoren echter scherp is en de situatie uit Figuur 11.25 zich voordoet, dan is $\vec{W} \cdot \vec{X} > 0$ en wordt het punt (x_1, x_2) ingedeeld bij de verkeerde klasse. M.a.w. $y = 0, \hat{y} = 1, y - \hat{y} = -1$ en $\eta(y - \hat{y}) < 0$.



Figuur 11.25: Scherpe hoek tussen beide vectoren.

Als \vec{W} aangepast wordt naar $\vec{W} + \eta(y - \hat{y}) \cdot \vec{X}$, dan wordt de hoek tussen \vec{W} en \vec{X} groter (zie Figuur 11.26). Er wordt a.h.w. opgeschoven naar de juiste klasse.



Figuur 11.26: Opschuiven naar klasse 0.

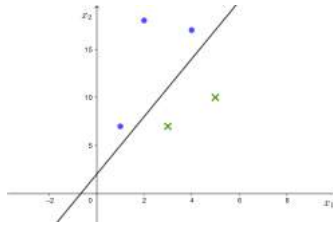
Wat is het verband tussen $\vec{W} \cdot \vec{X}$ en de gezochte scheidingslijn?

Stel dat het ML-systeem getraind is en dat de gewichten uiteindelijk bepaald zijn door het ML-systeem: de waarde van $w_0 \cdot 1 + w_1 \cdot x_1 + w_2 \cdot x_2$ bepaalt dan bij welke klasse een punt (x_1, x_2) wordt ingedeeld. De waarde kan ' > 0 , ' < 0 of ' $= 0$ ' zijn en de 'scheiding' van beide klassen gebeurt op ' $= 0$ ', m.a.w. de 'scheiding' wordt gegeven door de vergelijking $w_0 \cdot 1 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0$ of ook $\vec{W} \cdot \vec{X} = 0$; dit is de vergelijking van een rechte, een scheidingslijn (zie

Figuur 11.27).

$$w_0 \cdot 1 + w_1 \cdot x_1 + w_2 \cdot x_2 = 0 \Leftrightarrow w_2 \cdot x_2 = -w_1 \cdot x_1 - w_0$$

In het geval dat $w_2 \neq 0$, is $x_2 = -\frac{w_1}{w_2} \cdot x_1 - \frac{w_0}{w_2}$ of m.a.w. de scheidingslijn heeft als vergelijking $y = -\frac{w_1}{w_2} \cdot x - \frac{w_0}{w_2}$. In het geval dat $w_2 = 0$, is de scheidingslijn verticaal met vergelijking $x = -\frac{w_0}{w_1}$.



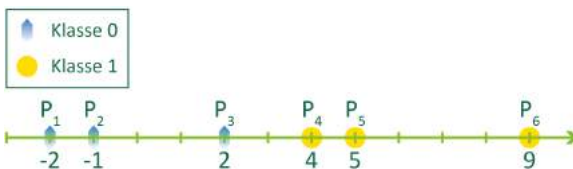
Figuur 11.27: Scheidingslijn.

(Rijksuniversiteit Groningen, 2005)

In hogere dimensies is de vergelijking $\vec{W} \cdot \vec{X} = 0$ de vergelijking van een scheidingsvlak of van een scheidingsoppervlak.

11.9 Variant op het 1D-voorbeeld: een netwerk met slechts één gewicht

Herneem het 1D-probleem uit paragraaf 11.6 (zie Figuur 11.28). In deze paragraaf wordt een neuraal netwerk opgebouwd met slechts één gewicht w .



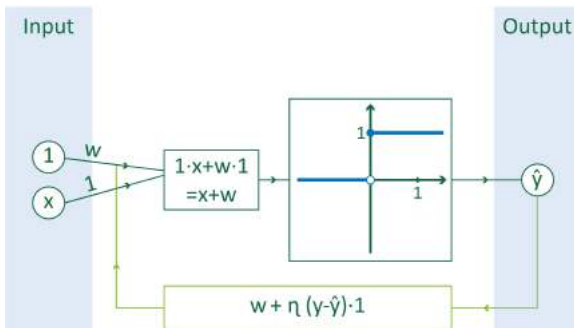
Figuur 11.28: Punten die behoren tot twee klassen.

Er zijn zes voorbeelden waar het systeem uit kan leren: -2 uit klasse 0, -1 uit klasse 0, 2 uit klasse 0, 4

uit klasse 1, 5 uit klasse 1 en 9 uit klasse 1.

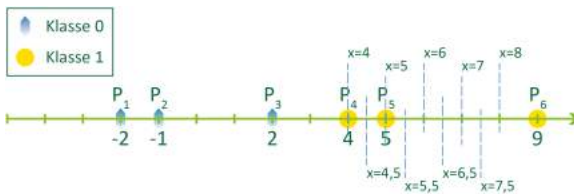
Het systeem beschouwt enkel scheidingen met een vergelijking van de vorm $x = a$. Een punt ligt dan bv. rechts van de scheiding als $x < a \Leftrightarrow x - a < 0 \Leftrightarrow x + (-a) < 0$. $w = -a$ is dan het gewicht dat kan worden aangepast. Dit wordt voorgesteld in Figuur 11.29.

Deze variant komt overeen met het Perceptron waarbij $w_0 = w$ en waarbij $w_1 = 1$ constant gehouden wordt.



Figuur 11.29: Neuraal netwerk met één gewicht en de Heaviside-functie als activatiefunctie.

Eerst moet men de learning rate kiezen, kies $\eta = 0,5$, en het initieel gewicht moet worden vastgelegd door het systeem, bv. $w = -8$. De voorlopige scheiding ligt dan op 8 en heeft dus als vergelijking $x = 8$ (zie Figuur 11.30). Hierdoor zullen slechts 4 punten juist geclassificeerd worden, nl. P_1, P_2, P_3 en P_6 .



Figuur 11.30: Scheiden van twee klassen.

Het systeem begint aan een eerste epoch. Het overloopt elk voorbeeldpunt.

P_1 :

$$-2 \xrightarrow{x+w} -2 + (-8) = -10 \xrightarrow[-10 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} = y$.

Het gewicht moet niet worden aangepast.

Reken na dat hetzelfde waar is voor P_2 en P_3 .

P_4 :

$$4 \xrightarrow{x+w} 4 + (-8) = -4 \xrightarrow[-4 < 0]{H} 0 \text{ dus } \hat{y} = 0 \text{ en } \hat{y} \neq y.$$

Het gewicht moet wel worden aangepast.

Het gewicht w wordt vervangen door $w + \eta \cdot (y - \hat{y})$,

dus

$$w = -8 + 0,5 \cdot (1 - 0) \Leftrightarrow w = -8 + 0,5 \Leftrightarrow w = -7,5.$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking: $x - 7,5 = 0 \Leftrightarrow x = 7,5$.

Vul de opeenvolgende scheidingen aan op Figuur 11.30.

P_5 :

$$5 \xrightarrow{x+w} 5 + (-7,5) = -2,5 \xrightarrow[-2,5 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} \neq y$.

Het gewicht moet opnieuw worden aangepast.

$$w = -7,5 + 0,5 \cdot (1 - 0) \Leftrightarrow w = -7,5 + 0,5 \Leftrightarrow w = -7.$$

Merk op dat de scheiding naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking: $x - 7 = 0 \Leftrightarrow x = 7$.

Reken na dat P_6 bij de juiste klasse wordt ingedeeld, dus voor P_6 moet het gewicht niet worden aangepast.

Alle voorbeelden zijn nu één keer doorlopen. De eerste epoch is voltooid.

Nu doorloopt het systeem opnieuw alle voorbeelden.

Het onderneemt een tweede epoch.

P_1 :

$$-2 \xrightarrow{x+w} -2 + (-7) = -9 \xrightarrow[-9 < 0]{H} 0$$

dus $\hat{y} = 0$ en $\hat{y} = y$.

P_1 wordt bij de juiste klasse ingedeeld.

Reken na dat ook P_2 en P_3 bij de juiste klasse worden ingedeeld.

P_4 :

$$4 \xrightarrow{x+w} 4 + (-7) = -3 \xrightarrow[-3 < 0]{H} 0 \text{ dus } \hat{y} = 0 \text{ en } \hat{y} \neq y.$$

Het gewicht moet nogmaals worden aangepast.

$$w = -7 + 0,5 \cdot (1 - 0) \Leftrightarrow w = -7 + 0,5 \Leftrightarrow w = -6,5.$$

Merk op dat de scheiding opnieuw naar de juiste kant is opgeschoven: naar links! De voorlopige scheiding heeft nu als vergelijking: $x - 6,5 = 0 \Leftrightarrow x = 6,5$.

P_5 wordt ook niet correct geclassificeerd. Het gewicht wordt aangepast naar $w = -6$. Reken na dat P_6 geen aanpassing vereist. Dus de tweede epoch zit erop. De voorlopige scheiding heeft als vergelijking: $x = 6$.

De derde epoch. Voor P_4 gebeurt een aanpassing naar $w = -5,5$ en voor P_5 naar $w = -5$. P_6 wordt bij de juiste klasse ingedeeld. De derde epoch is gedaan. De voorlopige scheiding heeft als vergelijking: $x = 5$.

Vierde epoch. Voor P_4 gebeurt een aanpassing naar $w = -4,5$. P_5 en P_6 worden bij de juiste klasse ingedeeld. De vierde epoch is gedaan. De voorlopige scheiding heeft als vergelijking: $x = 4,5$.

Vijfde epoch. Voor P_4 gebeurt een aanpassing naar $w = -4$. Alle andere punten worden bij de juiste klasse ingedeeld. De vijfde epoch is gedaan. De voorlopige scheiding heeft als vergelijking: $x = 4$.

Zesde epoch. Alle punten worden nu juist geclassificeerd. Het systeem is erin geslaagd de punten van elkaar te scheiden.

De uiteindelijke scheiding heeft dus als vergelijking: $x = 4$.

De initiële en willekeurige waarde van het gewicht heeft een grote impact op de duurtijd van het leren.

11.10 Lineaire regressie

Men kan machinaal leren ook gebruiken om de best passende rechte of regressielijn te vinden bij enkele gegeven punten. Men kan dan bijvoorbeeld de lengte van het kelkblad van de *Iris virginica* voorspellen als de lengte van het kroonblad gekend is (zie notebook 'Regressie met data over de *Iris virginica*'). In deze notebook wordt de regressielijn bepaald door stap voor stap de gewichten aan te passen. Er wordt getoond welk effect deze aanpassingen hebben op de ligging van de rechte. De coëfficiënten in de vergelijking van de rechte staan immers onmiddellijk in verband met de gewichten.

Nadien wordt nog eens hetzelfde gedaan met behulp van de ingebouwde functies van de Python-module scikit-learn.

De lengte van het kelkblad wordt weergegeven door x en de lengte van het kroonblad fungeert als label y . Omdat er gewerkt wordt met één kenmerk, is er één inputneuron. Opnieuw voegt men een extra neuron met waarde 1 toe. Er moeten nu twee gewichten worden geïnitieerd en worden aangepast op basis van de 50 gelabelde voorbeelden (zie Figuur 11.31).

Men werkt dan met input $X_i(1, x)$ met $x_0 = 1, x_1 = x \in \mathbb{R}$, label $y_i \in \{0,1\}$, output $\hat{y}_i \in \{0,1\}, i = 1, \dots, 50$ en $\vec{W}(w_0, w_1)$, met $w_j \in \mathbb{R}, j = 0, 1$.

De gewichten spelen de rol van de geschatte coëfficiënten in de vergelijking van de regressielijn. Om de regressielijn te bepalen, kijkt men tijdens een epoch voor elk punt X_i het residu, nl. de fout of het verschil tussen de werkelijke y -waarde (y_i) en de geschatte y -waarde (\hat{y}_i). \hat{y}_i is niets anders dan de waarde die men bekomt als men het punt $X_i(1, x_i)$ invult in de geschatte vergelijking van de regressielijn, vermits als activatiefunctie de identiteit wordt gebruikt. Na een voltooide

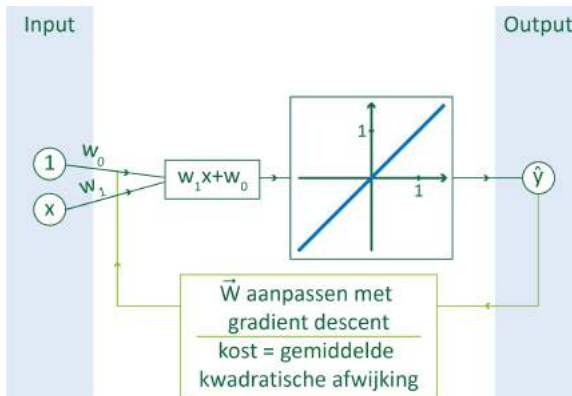
Een regressielijn kan men ook rechtstreeks vinden met formules uit de wiskunde. Hoe dat verloopt, wordt uitgelegd in de kader 'De wiskunde achter de gemiddelde kwadratische afwijking'.



Ga interactief aan de slag met de notebook 'Regressie met data over de *Iris virginica*'.

De geschatte regressielijn heeft als vergelijking $y = w_1 x + w_0$.

$$\hat{y}_i = w_1 x_i + w_0.$$



Figuur 11.31: Neuraal netwerk, twee gewichten, identiteit als activatiefunctie.

epoch beschouwt men het gemiddelde van de kwadraten van de residu's.

Men noemt dit gemiddelde 'de kost', men noteert deze kost met E , in deze notebook is $n = 50$:

$$E = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

De bedoeling van de training is om deze gemiddelde fout te minimaliseren.

Bij een rechte die perfect door alle punten gaat, is de kost steeds 0. De voorspelde waarde is dan immers steeds gelijk aan de gegeven y -waarde, zodat alle termen in bovenstaande som 0 zijn.

Als de punten niet allemaal op één rechte liggen, dan bepaalt men voor welke waarden van de gewichten de kost (hier dus de gemiddelde kwadratische afwijking) minimaal is. Omdat deze kost afhankelijk is van de \hat{y}_i , is hij afhankelijk van de gewichten, hier dus van twee variabelen: de richtingscoëfficiënt w_1 en de doorgang met de y -as w_0 in de vergelijking van de rechte. Voor andere waarden van de gewichten heeft men een andere kost. Men spreekt van de kostenfunctie (*loss function*).

Het minimaliseren van de kost gebeurt met de methode van *gradient descent* (zie verder in paragraaf 11.12).

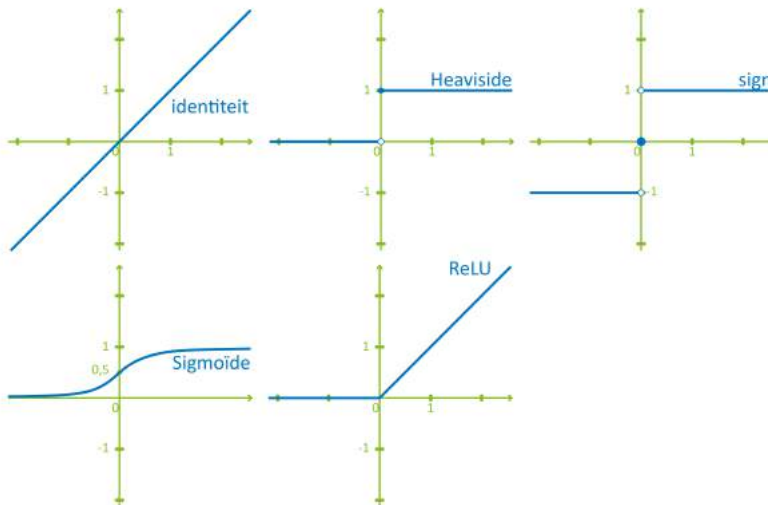
Deze kostenfunctie is de gemiddelde kwadratische afwijking, *mean squared error of MSE*.

De kostenfunctie of *loss function* is een functie van de gewichten.

De gewichten worden niet per punt aangepast.

Activatiefuncties

De Heaviside-functie (drempelwaardefunctie), de identiteitsfunctie, de sign-functie, de sigmoïde en de ReLU-functie zijn niet-lineaire functies. Ze worden vaak aangewend als activatiefunctie in een neurale netwerk.



De wiskunde achter de gemiddelde kwadratische afwijking

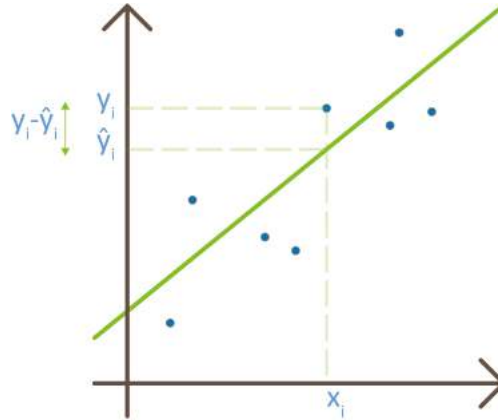
In de wiskunde spreekt men over de som van de kleinste kwadraten.

$$E = \sum_{i=1}^n (y_i - \hat{y}_i)^2.$$

Men gaat niet over op de gemiddelde waarde, maar dat maakt niet uit aangezien het minimum van $\sum_{i=1}^n (y_i - \hat{y}_i)^2$ en het minimum van $\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$ op dezelfde plaats worden bereikt.

Men heeft n gegeven punten (x_i, y_i) , $i = 1, \dots, n$. Stel dat de lijn als vergelijking $y = mx + q$ heeft. Het residu voor een gegeven punt (x_i, y_i) is het verschil tussen de werkelijke y -waarde y_i en de geschatte y -waarde \hat{y}_i , m.a.w. $y_i - \hat{y}_i$.

Men zoekt voor welke waarden van de coëfficiënten, m en q , in de vergelijking van de regressielijn de som van de kwadraten van de residu's minimaal is.



Men beschouwt het gemiddelde \bar{x} van alle x_i en het gemiddelde \bar{y} van alle y_i .
Dus:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ en } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Voor een punt (x_i, y_i) beschouwt men de afwijkingen t.o.v. het gemiddelde:
 $u_i = x_i - \bar{x}$ en $v_i = y_i - \bar{y}$.

Het residu voor dat punt is dus:

$$\begin{aligned} y_i - \hat{y}_i &= y_i - (mx_i + q) \\ &= y_i - mx_i - q + \bar{y} - \bar{y} + m\bar{x} - m\bar{x} \\ &= y_i - \bar{y} - m(x_i - \bar{x}) - q + \bar{y} - m\bar{x} \\ &= v_i - mu_i - (q - \bar{y} + m\bar{x}). \end{aligned}$$

De som van de kwadraten van de residu's van alle gegeven punten is:

$$\sum_{i=1}^n (y_i - \hat{y}_i)^2 = \sum_{i=1}^n (v_i - mu_i - (q - \bar{y} + m\bar{x}))^2$$

$$\begin{aligned}
&= \sum_{i=1}^n ((v_i - mu_i)^2 - 2(v_i - mu_i)(q - \bar{y} + m\bar{x}) + (q - \bar{y} + m\bar{x})^2) \\
&= \sum_{i=1}^n (v_i - mu_i)^2 - 2(q - \bar{y} + m\bar{x}) \sum_{i=1}^n (v_i - mu_i) + \sum_{i=1}^n (q - \bar{y} + m\bar{x})^2 \\
&\quad \text{met } m \text{ en } q - \bar{y} + m\bar{x} \text{ onafhankelijk van } i \\
&= \sum_{i=1}^n (v_i - mu_i)^2 - 2(q - \bar{y} + m\bar{x}) \left(\sum_{i=1}^n v_i - m \sum_{i=1}^n u_i \right) + n(q - \bar{y} + m\bar{x})^2 \\
&\quad \text{met } \sum_{i=1}^n u_i = 0 \text{ en } \sum_{i=1}^n v_i = 0 \\
&\quad \text{want } \sum_{i=1}^n u_i = \sum_{i=1}^n (x_i - \bar{x}) = \sum_{i=1}^n x_i - n\bar{x} = n\bar{x} - n\bar{x} = 0 \\
&\quad \text{en } \sum_{i=1}^n v_i = \sum_{i=1}^n (y_i - \bar{y}) = \sum_{i=1}^n y_i - n\bar{y} = n\bar{y} - n\bar{y} = 0 \\
&= \sum_{i=1}^n (v_i - mu_i)^2 + n(q - \bar{y} + m\bar{x})^2.
\end{aligned}$$

Deze som is minimaal als beide termen minimaal zijn.

Bepaal m zodat de eerste term $\sum_{i=1}^n (v_i - mu_i)^2$ minimaal is.

$$\begin{aligned}
\sum_{i=1}^n (v_i - mu_i)^2 &= \sum_{i=1}^n (v_i^2 - 2mv_iu_i + m^2u_i^2) \\
&= \sum_{i=1}^n v_i^2 - 2m \sum_{i=1}^n (v_iu_i) + m^2 \sum_{i=1}^n u_i^2 \\
&\quad \text{met } m \text{ onafhankelijk van } i \\
&= \sum_{i=1}^n u_i^2 \cdot m^2 - 2 \sum_{i=1}^n (u_iv_i) \cdot m + \sum_{i=1}^n v_i^2
\end{aligned}$$

Dit is een kwadratische uitdrukking in m . Ze is minimaal als

$$m = \frac{\sum_{i=1}^n (u_iv_i)}{\sum_{i=1}^n u_i^2}$$

(cfr. de formule van de top van een dalparabool).

De tweede term $n(q - \bar{y} + m\bar{x})^2$ is minimaal als ze 0 is (omwille van het kwadraat en doordat n ook strikt positief is). Het minimum van de tweede term bekomt men voor $q = \bar{y} - m\bar{x}$.

De regressielijn heeft dus als coëfficiënten

$$m = \frac{\sum_{i=1}^n ((x_i - \bar{x})(y_i - \bar{y}))}{\sum_{i=1}^n (x_i - \bar{x})^2} \text{ en } q = \bar{y} - m\bar{x},$$

m.a.w. de regressielijn van de punten (x_i, y_i) , $i = 1, \dots, n$ heeft als vergelijking

$$y = \frac{\sum_{i=1}^n ((x_i - \bar{x})(y_i - \bar{y}))}{\sum_{i=1}^n (x_i - \bar{x})^2} x + \bar{y} - m\bar{x}$$

met

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \text{ en } \bar{y} = \frac{1}{n} \sum_{i=1}^n y_i.$$

Men kan met deze formules rechtstreeks de vergelijking van de regressielijn opstellen.

Er is slechts één minimum. Er is dus slechts één rechte te vinden.

In paragraaf 11.12 gaat men op een andere manier naar dit minimum op zoek: door de gradiënt gelijk te stellen aan de nulvector, dus $\nabla \vec{E} = \vec{0}$. Men vindt hetzelfde resultaat.

De correlatiecoëfficiënt wordt gedefinieerd als

$$r = \frac{s(x)}{s(y)} m.$$

$s(x)$ is de standaardafwijking van de x_i : de gemiddelde kwadratische afwijking t.o.v. het gemiddelde \bar{x} . $s(y)$ is de standaardafwijking van de y_i : de gemiddelde kwadratische afwijking t.o.v. het gemiddelde \bar{y} . De standaardafwijking is een belangrijke spreidingsmaat uit de beschrijvende statistiek.

$$s(x) = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}} \text{ en } s(y) = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n - 1}}$$

De correlatiecoëfficiënt r kan m.b.v. deze formules herschreven worden als:

$$r = \frac{1}{n - 1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s(x)} \cdot \frac{y_i - \bar{y}}{s(y)} \right).$$

De producten van de gestandaardiseerde x_i en y_i worden dus gesommeerd en deze som wordt gedeeld door $n - 1$.

Hoe meer x_i en y_i afwijken van de gemiddelden, resp. \bar{x} en \bar{y} , hoe minder samenhang er is tussen de punten. r is een maat voor die samenhang.

r kan ook herschreven worden als de cosinus van de hoek tussen $\vec{U}(u_1, u_2, \dots, u_n)$ en $\vec{V}(v_1, v_2, \dots, v_n)$, wat betekent dat r in $[-1, 1]$ ligt.

$$\begin{aligned} r &= \frac{1}{n-1} \frac{1}{s(x)} \frac{1}{s(y)} \sum_{i=1}^n ((x_i - \bar{x}) \cdot (y_i - \bar{y})) \\ &= \frac{1}{n-1} \frac{1}{s(x)} \frac{1}{s(y)} \vec{U} \cdot \vec{V} \\ &= \frac{1}{n-1} \frac{1}{s(x)} \frac{1}{s(y)} \|\vec{U}\| \|\vec{V}\| \cos(\widehat{\vec{U}, \vec{V}}) \\ &= \frac{1}{n-1} \sqrt{\frac{n-1}{\sum_{i=1}^n (x_i - \bar{x})^2} \cdot \frac{n-1}{\sum_{i=1}^n (y_i - \bar{y})^2}} \|\vec{U}\| \|\vec{V}\| \cos(\widehat{\vec{U}, \vec{V}}) \\ &= \cos(\widehat{\vec{U}, \vec{V}}), \end{aligned}$$

aangezien $\sum_{i=1}^n (x_i - \bar{x})^2 = \|\vec{U}\|^2$ en $\sum_{i=1}^n (y_i - \bar{y})^2 = \|\vec{V}\|^2$.

Als alle gegeven punten op één rechte liggen, dan is $r = \pm 1$ (1 voor een stijgende en -1 voor een dalende regressielijn). Hoe dichter $|r|$ bij 1 ligt, hoe meer samenhang er is.

11.11 Binaire classificatie

11.11.1 Sigmoide als activatiefunctie

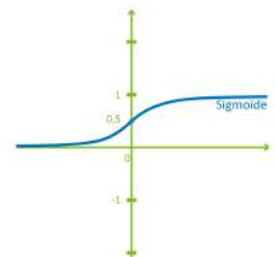
Soms zijn twee klassen niet volledig linear scheidbaar, maar slechts op enkele punten na. Het Perceptron is voor dergelijke klassen niet zo geschikt.

In dat geval zal men binair classificeren met *logistic regression* (Nerbonne, 2004). Als activatiefunctie wordt dan gebruikgemaakt van de sigmoïde-functie, een reële functie met voorschrift:

$$\sigma : \mathbb{R} \rightarrow \mathbb{R} \text{ met } \sigma(x) = \frac{1}{1 + e^{-x}}.$$

Men spreekt ook van een logistische functie. De output is een waarde \hat{y} die weergeeft hoe zeker het model is dat de input tot de klasse met label 1 behoort. De

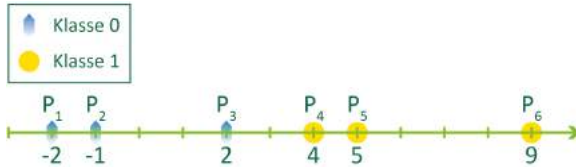
Voor het Perceptron-algoritme, zie paragraaf 11.7. De activatiefunctie van het Perceptron is de Heaviside-functie.



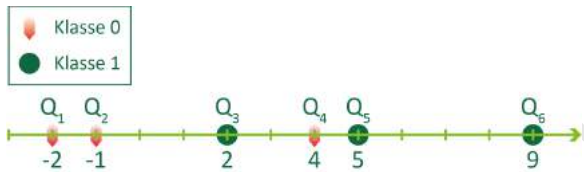
Figuur 11.32: Sigmoide-functie.

zekerheid dat de input tot de andere klasse behoort, is dan $1 - \hat{y}$.

Bekijk het voorbeeld in 1D met slechts één gewicht w (Figuur 11.33, paragraaf 11.9) en een tweede geval ook met één gewicht w (Figuur 11.34).



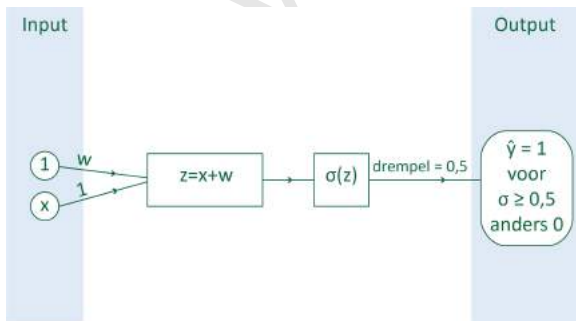
Figuur 11.33: Eerste geval: punten die behoren tot twee lineair scheidbare klassen.



Figuur 11.34: Tweede geval: punten die behoren tot twee klassen die niet lineair scheidbaar zijn.

Om de punten van elkaar te scheiden, dus onder te brengen in twee klassen, kan men een neuraal netwerk gebruiken zoals in Figuur 11.35.

Voorbeelden zijn geïnspireerd op een blogpost van Daniel Godoy (Godoy, 2018).



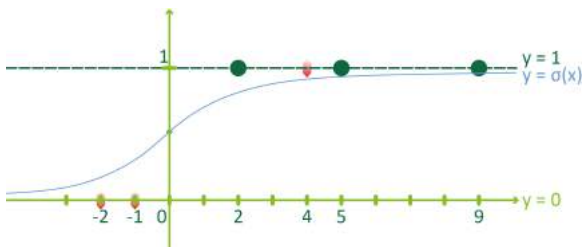
Figuur 11.35: Neuraal netwerk binaire classificatie.

Stel voor het tweede geval (Figuur 11.34) dat $w = 0$, dan gebeurt het volgende in het netwerk: $x \mapsto x$

als lineaire operatie, gevolgd door $x \mapsto \sigma(x)$ voor de activatie.

Als $\sigma(x) \geq 0,5$, dan wordt het punt ingedeeld bij klasse 1, anders bij klasse 0. 0,5 is de drempelwaarde of *threshold* die wordt gehanteerd.

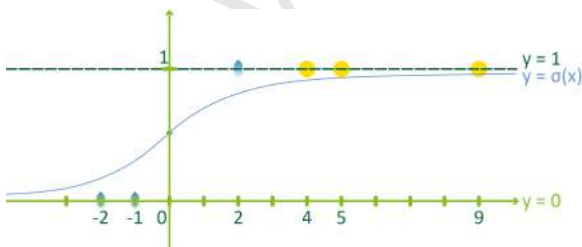
Dit zou enkel Q_4 fout classificeren als horend bij klasse 1, aangezien de zekerheid voor klasse 1 groot is. Dat is een mooi resultaat! Alle punten in de juiste categorie krijgen, gaat immers niet (zie Figuur 11.36).



De netwerk-architect kiest zelf de waarde van de *threshold*, de drempelwaarde, afhankelijk van het soort probleem en de beoogde oplossing.

Figuur 11.36: Eerste sigmoïde om te classificeren in het tweede geval. Er is één punt fout geïclassificeerd.

Dezelfde sigmoïde zou geen goed resultaat geven voor het eerste geval, want die klassen zijn perfect scheidbaar en toch zou P_3 fout geïclassificeerd worden (zie Figuur 11.37).

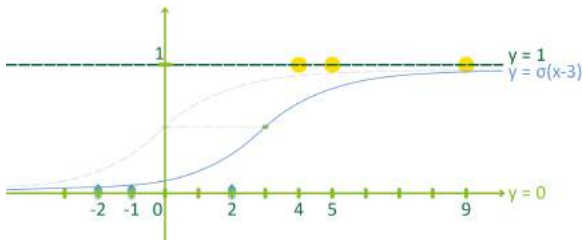


Figuur 11.37: Eerste sigmoïde om te classificeren in het eerste geval. Er is één punt fout geïclassificeerd.

In het eerste geval doet men beter het volgende:

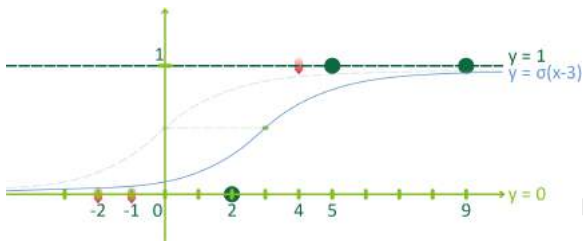
Stel daar $w = -3$, dan: $x \mapsto x - 3 \mapsto \sigma(x - 3)$ (zie Figuur 11.35). Figuur 11.38 toont het resultaat.

Deze sigmoïde zou dan weer een slechtere keuze zijn



Figuur 11.38: Tweede sigmoïde om te classificeren in het eerste geval. Alle punten zijn juist geclassificeerd.

voor het tweede voorbeeld. Er worden dan immers twee punten bij de foute klasse ingedeeld (zie Figuur 11.39).



Figuur 11.39: Tweede sigmoïde om te classificeren in het tweede geval. Er worden twee punten fout geclassificeerd.

11.11.2 Binary cross-entropy

Bij classificatie gaat men op zoek naar de scheiding met de 'kleinste fout'. Men heeft een kostenfunctie nodig om dit te kunnen doen. Als kostenfunctie voor een probleem van binaire classificatie hanteert men bv. *binary cross-entropy*.

De kostenfunctie binary cross-entropy (*BCE*) is het tegengestelde van de natuurlijke logaritme van de *likelihood function* L , dus $BCE = -\ln L$. De likelihood-waarde L voor bepaalde waarden van de gewichten is de kans dat het model de juiste klassen voorspelt, ervan uitgaand dat alle voorspellingen onafhankelijk zijn van elkaar.

Men streeft ernaar dat deze kans zo groot mogelijk is. Men wil de likelihood function L dus maximaliseren, wat overeenkomt met het minimaliseren van de kos-

tenfunctie BCE . Immers, de natuurlijke logaritme \ln is een strikt stijgende functie, dus $\ln L$ zal op hetzelfde moment een maximum bereiken als L zelf, en, vanwege het minteken, gebeurt dat wanneer BCE een minimum bereikt.

Bekijk de *likelihood function* nu concreet.

Veronderstel dat er n datapunten, X_i , met $i = 1, \dots, n$, zijn en dus ook n labels, waarvan de eerste k datapunten label 1 hebben en de andere allemaal label 0. Dus $y_i = 1$ voor $i = 1, \dots, k$ en $y_i = 0$ voor $i = k + 1, \dots, n$.

De likelihood-waarde $L = P(y_1 y_2 \dots y_n)$, dus de kans dat alle datapunten juist worden geïdentificeerd, m.a.w. de kans dat X_1 wordt ingedeeld bij klasse y_1 , X_2 bij klasse y_2 , X_3 bij klasse y_3 , enz.

$$\begin{aligned} L &= P(y_1 y_2 \dots y_n) \\ &= P(\hat{y}_1 = 1) \cdot P(\hat{y}_2 = 1) \dots P(\hat{y}_k = 1) \cdot P(\hat{y}_{k+1} = 0) \dots P(\hat{y}_n = 0) \\ &= P(\hat{y} = 1|X_1) P(\hat{y} = 1|X_2) \dots P(\hat{y} = 1|X_k) P(\hat{y} = 0|X_{k+1}) \dots P(\hat{y} = 0|X_n) \\ &= P(\hat{y} = 1|X_1) \dots P(\hat{y} = 1|X_k) (1 - P(\hat{y} = 1|X_{k+1})) \dots (1 - P(\hat{y} = 1|X_n)) \end{aligned}$$

Er wordt dus gekeken hoe de kans dat \hat{y} 1 is, afhangt van de invoer X . Het verband tussen X en deze kans is zeker niet te benaderen door een lineair verband aangezien $P(\hat{y} = 1|X)$ enkel waarden tussen 0 en 1 kan aannemen en een eerstegraadsfunctie alle reële waarden.

Beschouw de zogenaamde *odds*:

$$\text{odds} = \frac{P(\hat{y} = 1|X)}{1 - P(\hat{y} = 1|X)}.$$

Men noemt de *odds* ook wel de kansverhouding.

De *odds* neemt waarden aan in $]0, +\infty[$. De natuurlijke logaritme van de *odds*, $\ln \frac{P(\hat{y}=1|X)}{1-P(\hat{y}=1|X)}$, kan alle reële waarden aannemen.

Men zal het verband tussen $\ln \frac{P(\hat{y}=1|X)}{1-P(\hat{y}=1|X)}$ en de invoer X lineair benaderen d.m.v. lineaire regressie.

Het beeld van de homografische functie f met als voorschrift $f(x) = \frac{p}{1-p}$ en domein $]0, 1[$ heeft als beeld $]0, +\infty[$.

Als men werkt in 1 dimensie, dan stelt men dus

$$\ln \frac{P(\hat{y} = 1|x)}{1 - P(\hat{y} = 1|x)} = b_1 x + b_0,$$

of m-dimensionaal

$$\ln \frac{P(\hat{y} = 1|X)}{1 - P(\hat{y} = 1|X)} = b_0 + b_1 x_1 + b_2 x_2 + \dots + b_m x_m$$

met $X = (x_1, x_2, \dots, x_m)$. De parameters b_i moeten bepaald worden.

Dus eigenlijk schrijft men $\ln \frac{P(\hat{y}=1|X)}{1-P(\hat{y}=1|X)}$ als z met z een lineaire combinatie van de x_i .

Dit komt overeen met de z uit de neurale netwerken.

Het statistisch model dat hier gebruikt wordt is dus:

$$\ln \frac{P(\hat{y} = 1|X)}{1 - P(\hat{y} = 1|X)} = z.$$

Hieruit leidt men dan een model af voor het verband tussen $P(\hat{y} = 1|X)$ en X :

$$P(\hat{y} = 1|X) = \sigma(z).$$

Immers:

$$\begin{aligned} \ln \frac{P(\hat{y} = 1|X)}{1 - P(\hat{y} = 1|X)} &= z \\ \Leftrightarrow \frac{P(\hat{y} = 1|X)}{1 - P(\hat{y} = 1|X)} &= e^z \\ \Leftrightarrow P(\hat{y} = 1|X) &= e^z \cdot (1 - P(\hat{y} = 1|X)) \\ \Leftrightarrow P(\hat{y} = 1|X) &= e^z - e^z \cdot P(\hat{y} = 1|X) \\ \Leftrightarrow P(\hat{y} = 1|X) + e^z \cdot P(\hat{y} = 1|X) &= e^z \\ \Leftrightarrow (1 + e^z) \cdot P(\hat{y} = 1|X) &= e^z \\ \Leftrightarrow P(\hat{y} = 1|X) &= \frac{e^z}{1 + e^z} \\ \Leftrightarrow P(\hat{y} = 1|X) &= \frac{1}{1 + e^{-z}} \\ \Leftrightarrow P(\hat{y} = 1|X) &= \sigma(z) \end{aligned}$$

Voor de likelihood-waarde L betekent dit:

$$\begin{aligned} L &= P(\hat{y} = 1|X_1) \dots P(\hat{y} = 1|X_k) (1 - P(\hat{y} = 1|X_{k+1})) \dots (1 - P(\hat{y} = 1|X_n)) \\ &= \sigma(z_1) \sigma(z_2) \dots \sigma(z_k) (1 - \sigma(z_{k+1})) \dots (1 - \sigma(z_n)) \end{aligned}$$

met z_i een lineaire combinatie van de coördinaatgetallen van de X_i .

Terug naar de kostenfunctie, binary cross-entropy,
 $BCE = -\ln L$.

Eigenschap:

$$\ln(a \cdot b) = \ln a + \ln b$$

Omdat L een product is, kan men $\ln L$ schrijven als een som. Er geldt nog steeds dat $y_i = 1, i = 1, \dots, k$ en $y_i = 0, i = k + 1, \dots, n$.

$$\begin{aligned} \ln L &= \ln[\sigma(z_1) \sigma(z_2) \dots \sigma(z_k) (1 - \sigma(z_{k+1})) \dots (1 - \sigma(z_n))] \\ &= \ln[\sigma(z_1) \sigma(z_2) \dots \sigma(z_k)] + \ln[(1 - \sigma(z_{k+1})) \dots (1 - \sigma(z_n))] \\ &= \sum_{i=1}^k \ln(\sigma(z_i)) + \sum_{i=k+1}^n \ln(1 - \sigma(z_i)) \\ &= \sum_{i=1}^k \ln(\sigma(z_i)) \cdot 1 + \sum_{i=k+1}^n \ln(1 - \sigma(z_i)) \cdot (1 - 0) \\ &= \sum_{i=1}^k \ln(\sigma(z_i)) \cdot y_i + \sum_{i=k+1}^n \ln(1 - \sigma(z_i)) \cdot (1 - y_i) \\ &= \sum_{i=1}^k \ln(\sigma(z_i)) \cdot y_i + 0 + 0 + \sum_{i=k+1}^n \ln(1 - \sigma(z_i)) \cdot (1 - y_i) \\ &= \sum_{i=1}^k \ln(\sigma(z_i)) \cdot y_i + \sum_{i=k+1}^n \ln(\sigma(z_i)) \cdot y_i + \\ &\quad \sum_{i=1}^k \ln(1 - \sigma(z_i)) \cdot (1 - y_i) + \sum_{i=k+1}^n \ln(1 - \sigma(z_i)) \cdot (1 - y_i) \\ &= \sum_{i=1}^n \ln(\sigma(z_i)) \cdot y_i + \sum_{i=1}^n \ln(1 - \sigma(z_i)) \cdot (1 - y_i) \\ &= \sum_{i=1}^n [\ln(\sigma(z_i)) \cdot y_i + \ln(1 - \sigma(z_i)) \cdot (1 - y_i)] \end{aligned}$$

$$BCE = - \sum_{i=1}^n [\ln(\sigma(z_i)) \cdot y_i + \ln(1 - \sigma(z_i)) \cdot (1 - y_i)]$$

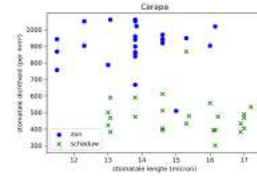
11.11.3 Voorbeeld classificatie van bezonde en beschaduwde bladeren

Deze methode van binaire classificatie wordt toegepast in de notebook 'Classificatie: stomata op bezonde en

beschaduwde bladeren in het Amazonewoud’.

De data die gebruikt worden in de notebook, werden ter beschikking gesteld door Miguel Camargo (persoonlijke communicatie via e-mail) (Camargo & Marengo, 2012). Deze data zijn bij benadering lineair scheidbaar. Het minimaliseren van de kost gebeurt met *gradient descent* (zie paragraaf 11.12).

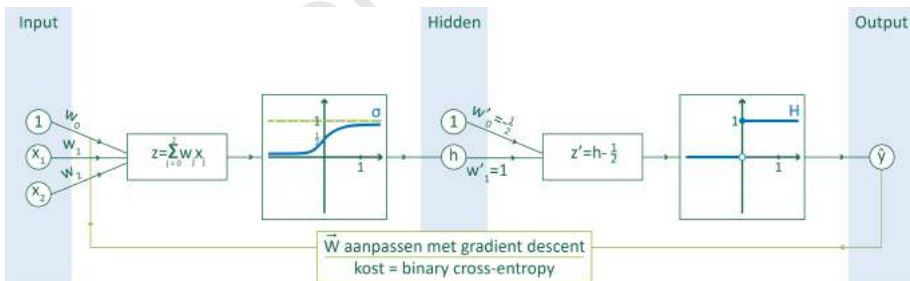
Van 50 bladeren werden de stomatale lengte en de stomatale dichtheid bepaald. Sommige bladeren groeiden in de zon, andere in de schaduw. Zon of schaduw fungeert als label. Het is de bedoeling de bezonde bladeren te scheiden van de beschaduwde. Omdat er gewerkt wordt met twee kenmerken, zijn er twee inputneuronen. Opnieuw voegt men een extra neuron met waarde 1 toe. Er moeten nu drie gewichten worden geïnitieerd en worden aangepast op basis van de 50 gelabelde voorbeelden (zie Figuur 11.41). Men werkt dan met input $X_i(1, x_1, x_2)$ met $x_0 = 1, x_j \in \mathbb{R}$ voor $j = 1, 2$, label $y_i \in \{0,1\}$, output $\hat{y}_i \in \{0,1\}, i = 1, \dots, 50$ en $\vec{W}(w_0, w_1, w_2)$, met $w_j \in \mathbb{R}, j = 0, 1, 2$.



Figuur 11.40: Puntenwolk data bezonde en beschaduwde bladeren.



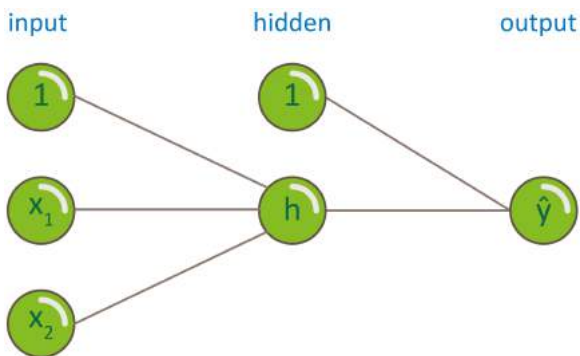
Bekijk de notebook 'Classificatie: stomata op bezonde en beschaduwde bladeren in het Amazonewoud’.



Figuur 11.41: Neuraal netwerk classificatie bezonde en beschaduwde bladeren.

De laag tussen de input layer en de output layer noemt men een hidden layer. De drempelwaarde $\frac{1}{2}$ vertaalt zich in een *bias* $w'_0 = -\frac{1}{2}$.

Het netwerk uit Figuur 11.41 kan eenvoudiger voorgesteld worden zoals in Figuur 11.42.



Figuur 11.42: Eenvoudige voorstelling neurale netwerk classificatie bezonde en beschaduwde blade- ren.

11.12 Gradient descent

In de wiskunde wordt veel aandacht besteed aan de afgeleide van een reële functie. Afgeleiden hebben veel toepassingen, bv. in de economie en in de fysica.

De afgeleide in een punt van een kromme is de richtingscoëfficiënt van de raaklijn in dat punt aan de kromme. Waar de kromme stijgt, is de raaklijn positief en waar de kromme daalt, is ze negatief. Omdat bv. de raaklijn in de top van een parabool horizontaal is, is de afgeleide in de top 0. Hoe steiler de raaklijn, hoe groter de richtingscoëfficiënt van de raaklijn in absolute waarde, dus hoe groter de afgeleide in absolute waarde.

Om het minimum van een functie te bepalen, kan men dus op zoek gaan naar de punten waar de afgeleide nul is.

Ook voor neurale netwerken zijn afgeleiden belangrijk. Men gebruikt ze om de kostenfunctie te minimaliseren. Men gaat op zoek naar het minimum door de gewichten stap voor stap aan te passen, gebaseerd op de afgeleide in het 'huidige' punt. Deze methode noemt men *gradient descent*. In het regressievoorbeeld (paragraaf 11.10) betekent dat dat er in het minimum i.p.v. een horizontale raaklijn een horizontaal raakvlak is. De kostenfunctie is daar immers een functie van twee

Het betreft hier punten van de kromme waar de overeenkomstige functie afleidbaar is. In die punten is er slechts één, niet verticale, raaklijn. Punten zoals keerpunten en knikpunten, waar er twee raaklijnen zijn of een verticale raaklijn is, worden hier buiten beschouwing gelaten.

variabelen, de gewichten w_0 en w_1 . De grafiek van de kostenfunctie is daar een oppervlak in \mathbb{R}^3 . In het classificatievoorbeeld (paragraaf 11.11) is de kostenfunctie zelfs een functie van drie variabelen. In veel neurale netwerk-problemen zijn er nog veel meer variabelen. Men spreekt dan van een rakend hypervlak.

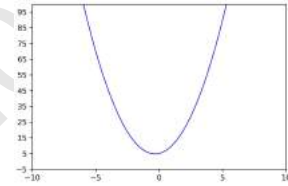


In de notebook 'Gradient descent' wordt getoond hoe men op zoek gaat naar het minimum van een kwadratische functie van één variabele met gradient descent.

Gradient descent wordt op een visuele manier uitgelegd in de notebook 'Gradient descent'.

11.12.1 Notebook 'Gradient descent'

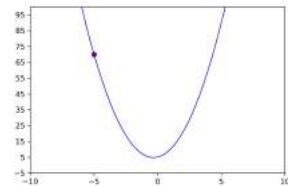
In de notebook 'Gradient descent' vertrekt men van de parabool met vergelijking $y = 3x^2 + 2x + 5$. Dit is een dalparabool (want $a = 3 > 0$). Deze parabool heeft één minimum. De parabool is afgebeeld in Figuur 11.43.



Figuur 11.43: Dalparabool.

De afgeleide in een punt van de parabool is de richtingscoëfficiënt van de raaklijn in dat punt aan de parabool. Waar de parabool stijgt, is de raaklijn positief en waar de parabool daalt, is ze negatief. In de top van de parabool is de raaklijn horizontaal en de afgeleide 0. Hoe steiler de raaklijn, hoe groter de richtingscoëfficiënt van de raaklijn in absolute waarde, dus hoe groter de afgeleide in absolute waarde.

Er wordt een willekeurig punt P gekozen op de parabool, op de grafiek in Figuur 11.44 afgebeeld in het paars.



Figuur 11.44: Willekeurig punt P op de dalparabool.

Het is de bedoeling het punt P naar het laagste punt van de parabool te laten 'bewegen' en zich daarbij te laten leiden door de afgeleide in het punt P . De x -waarde van P zal dus met bepaalde hoeveelheden moeten worden aangepast, zodat P effectief de top van de parabool nadert. Eens dicht bij de top moet men er alert voor zijn dat men niet voorbij de top gaat.

De raaklijn in P speelt ook een rol. Als P ver verwijderd is van de top, is die raaklijn steil. Als P dicht bij de top ligt, is die raaklijn niet steil meer. Hoe steiler de

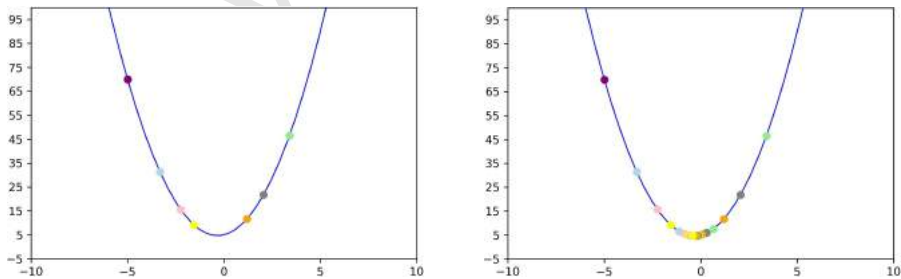
raaklijn, hoe groter de afgeleide in absolute waarde.

Als men de stappen dus zo kiest dat ze in grootte evenredig zijn met de afgeleide, dan zal men voor een punt P ver van de top (waar de raaklijn steiler is) relatief grote stappen nemen, en voor een punt P dicht bij de top (waar de raaklijn niet steil meer is) kleine stappen nemen. Net wat men wilt! Om dit te verwezenlijken kiest men een evenredigheidsfactor η , gelegen tussen 0 en 1 (η is de Griekse letter èta).

P ligt hier links van de top. De richtingscoëfficiënt van de raaklijn (m.a.w. de afgeleide) in P is negatief, dus moet men een waarde evenredig met de afgeleide van de x -waarde aftrekken, zodat P naar rechts beweegt.

Als men dit doet met $\eta = 0,3$, dan verplaatst P zich naar het groene punt rechts van de top. De richtingscoëfficiënt van de raaklijn (m.a.w. de afgeleide) is daar positief, want de kromme stijgt daar. Als men nu een waarde evenredig met de afgeleide van de x -waarde aftrekt, zal het punt naar links bewegen. Het komt terecht op het blauwe punt links van de top. Dus voor $\eta = 0,3$ wordt de top vrij goed bereikt als men lang genoeg doorgaat (zie Figuur 11.45).

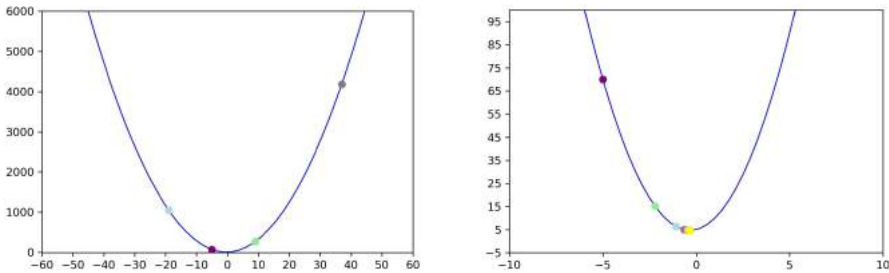
Het is niet toevallig dat hier η wordt gebruikt. Deze evenredigheidsfactor is de learning rate. Zie ook paragraaf 11.6 en paragraaf 11.8.



Figuur 11.45: $\eta = 0,3$.

Als men $\eta = 0,5$ kiest, zijn de stappen veel te groot. Het punt verwijderd zich van de top, zoals te zien op Figuur 11.46. Voor $\eta = 0,1$ bekommt men vrij snel het resultaat van Figuur 11.46.

Links: het punt P verplaatst zich stap voor stap naar het laagste punt van de parabool. Rechts: omdat het punt P dicht bij het laagste punt komt, moet men lang genoeg doorgaan.



Figuur 11.46: Links: $\eta = 0,5$. Het punt verwijderd zich van de top van de parabool. Rechts: $\eta = 0,1$. Het minimum wordt in een beperkt aantal stappen bereikt.

11.12.2 Uitbreiding naar meer dimensies

Een kostenfunctie heeft weliswaar doorgaans meerdere variabelen, men heeft te maken met samengestelde functies door meerdere lagen in het netwerk, en de keuze van de activatiefunctie speelt ook nog een rol voor het afleiden. De werkwijze van gradient descent wordt daarom uitgebreid naar meerdere dimensies en naar samengestelde functies: i.p.v. de afgeleide gebruikt men dan de gradiënt, een vector opgebouwd met de partiële afgeleiden van de kostenfunctie naar de verschillende gewichten, en zal men ook de kettingregel hanteren.

Zoals gezegd speelt de learning rate η een belangrijke rol bij gradient descent. Dit getal tussen 0 en 1 bepaalt mee hoe groot de stappen zijn waarmee men de *weights* aanpast. Hoe kleiner de leersnelheid, hoe voorzichtiger de aanpassingen. Een te kleine learning rate kan er echter toe leiden dat het systeem zeer traag leert. Bij een te grote learning rate kan men door te bruuske aanpassingen het minimum gemist hebben. Door optimalisatie op een valideringsset kan men een gepaste learning rate bepalen (zie paragraaf 11.15).

11.12.3 Toepassen op voorbeeld van lineaire regressie met twee gewichten

In het voorbeeld van regressie uit paragraaf 11.10 is de kostenfunctie, de gemiddelde kwadratische afwijking, een functie van twee variabelen, de gewichten m en q . In plaats van de afgeleide, gebruikt men hier de gradiënt. Men moet daarvoor 'partieel afleiden'. Dat betekent concreet dat als men afleidt naar de ene variabele, men de andere variabele als een constante moet beschouwen.

Beschouw de kostenfunctie E waarvan de gradiënt moet worden bepaald:

$$E(m, q) = \frac{1}{50} \sum_{i=1}^{50} (y_i - \hat{y}_i)^2$$

met $\hat{y}_i = mx_i + q, i = 1, \dots, 50$.

Er geldt voor $k = 0, 1$ ($n = 50$):

$$\begin{aligned} \frac{\partial E(m, q)}{\partial w_k} &= \frac{\partial}{\partial w_k} \left(\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial w_k} (y_i - \hat{y}_i)^2 \\ &= \frac{2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_k} (y_i - \hat{y}_i)) \\ &= \frac{2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) (\frac{\partial}{\partial w_k} y_i - \frac{\partial}{\partial w_k} \hat{y}_i)) \\ &= \frac{2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) (0 - \frac{\partial}{\partial w_k} (mx_i + q))) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_k} (mx_i + q)), \end{aligned}$$

De gradiënt is een uitbreiding van het begrip afgeleide naar meer dimensies. Het is een vector die opgebouwd wordt met de partiële afgeleiden van de functie naar de verschillende veranderlijken.

dus:

$$\begin{aligned}\frac{\partial E(m, q)}{\partial w_0} &= \frac{\partial E(m, q)}{\partial q} \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial q} (mx_i + q)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot 1) \\ &= \frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i)\end{aligned}$$

en

$$\begin{aligned}\frac{\partial E(m, q)}{\partial w_1} &= \frac{\partial E(m, q)}{\partial m} \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial m} (mx_i + q)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_i).\end{aligned}$$

De gradiënt van E is een vector opgebouwd met deze twee partiële afgeleiden:

$$\begin{aligned}\left(\frac{\partial E(m, q)}{\partial q}, \frac{\partial E(m, q)}{\partial m} \right) &= \left(\frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i), \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_i) \right) \\ &= \frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i, (y_i - \hat{y}_i) \cdot x_i) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i)(1, x_i)) \\ \nabla \vec{E} &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \vec{X}_i).\end{aligned}$$

In de notebook 'Gradient descent' wordt getoond dat de aanpassing van het gewicht dient te gebeuren door er een getal evenredig met de afgeleide van af te trekken. Hier werkt men op dezelfde manier. Men zal een vector evenredig met de gradiënt van de kostenfunctie van de weights-vector aftrekken.

Men zal \vec{W} dus na elke epoch aanpassen naar $\vec{W} - \eta \nabla \vec{E}$, dus naar $\vec{W} - \eta \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \vec{X}_i)$ met $n = 50$, n is het aantal datapunten.

In het algoritme in de notebook wordt

$$\left(\frac{\partial E(m, q)}{\partial q}, \frac{\partial E(m, q)}{\partial m} \right) = \left(\sum_{i=1}^n \frac{-2}{n} (y_i - \hat{y}_i), \sum_{i=1}^n \frac{-2}{n} ((y_i - \hat{y}_i) \cdot x_i) \right)$$

gebruikt. De sommen in het eerste en tweede coördinaatgetal worden eerst berekend en erna worden de resultaten afgetrokken van respectievelijk q en m om de aangepaste waarden van q en m te bekomen.

11.12.4 Toepassen op voorbeeld van regressie met drie gewichten

Voor drie gewichten werkt men analoog. De kostenfunctie is:

$$E(w_0, w_1, w_2) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

met $\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_0$, $i = 1, \dots, n$.

Voor $k = 0, 1, 2$ geldt:

$$\begin{aligned} \frac{\partial E}{\partial w_k} &= \frac{\partial}{\partial w_k} \left(\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right) \\ &= \frac{1}{n} \sum_{i=1}^n \frac{\partial}{\partial w_k} (y_i - \hat{y}_i)^2 \\ &= \frac{2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_k} (y_i - \hat{y}_i)) \\ &= \frac{2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \left(\frac{\partial}{\partial w_k} y_i - \frac{\partial}{\partial w_k} \hat{y}_i \right)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_k} (w_1 x_{i1} + w_2 x_{i2} + w_0)). \end{aligned}$$

Hieruit volgt:

$$\begin{aligned} \frac{\partial E}{\partial w_0} &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_0} (w_1 x_{i1} + w_2 x_{i2} + w_0)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot 1) \\ &= \frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i), \end{aligned}$$

$$\begin{aligned}\frac{\partial E}{\partial w_1} &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_1} (w_1 x_{i1} + w_2 x_{i2} + w_0)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i1}),\end{aligned}$$

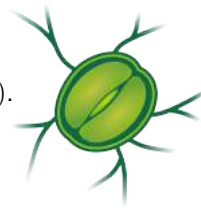
en

$$\begin{aligned}\frac{\partial E}{\partial w_2} &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \frac{\partial}{\partial w_2} (w_1 x_{i1} + w_2 x_{i2} + w_0)) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i2}).\end{aligned}$$

Dus ook:

$$\begin{aligned}&\left(\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2} \right) \\ &= \left(\frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i), \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i1}), \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i2}) \right) \\ &= \frac{-2}{n} \sum_{i=1}^n (y_i - \hat{y}_i, (y_i - \hat{y}_i) \cdot x_{i1}, (y_i - \hat{y}_i) \cdot x_{i2}) \\ &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) (1, x_{i1}, x_{i2})) \\ \nabla \vec{E} &= \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \vec{X}_i).\end{aligned}$$

De gewichten \vec{W} worden met dezelfde formule als hiervoor aangepast naar $\vec{W} - \eta \nabla \vec{E}$, dus de gewichten \vec{W} worden aangepast naar $\vec{W} - \eta \frac{-2}{n} \sum_{i=1}^n ((y_i - \hat{y}_i) \vec{X}_i)$.



Ook in de notebook 'Hoogte bomen en afmetingen stomata in het Amazonewoud' gaat men voor de regressie aan de slag met de gemiddelde kwadratische afwijking en gradient descent.

De data zijn het resultaat van het wetenschappelijk onderzoek van Camargo & Marengo (2011). De notebook behandelt tevens het splitsen van de data in training-, validerings- en testdata (paragraaf 11.13).

Aan de slag met de gemiddelde kwadratische afwijking, gradient descent en het splitsen van de data in de notebook 'Hoogte bomen en afmetingen stomata in het Amazonewoud'.

11.12.5 Voor regressie met m gewichten

Voor m gewichten verloopt het rekenwerk op dezelfde manier en blijft de laatste formule in vectorgedaante gelden.

11.12.6 Toepassen op voorbeeld classificatie

Bij het classificatieprobleem (paragraaf 11.11) is de kostenfunctie binary cross-entropy afhankelijk van drie gewichten.

$$BCE = - \sum_{i=1}^n [\ln(\sigma(z_i)) \cdot y_i + \ln(1 - \sigma(z_i)) \cdot (1 - y_i)]$$

Men werkt dan met input $X_i(1, x_1, x_2)$ met $x_0 = 1, x_j \in \mathbb{R}, j = 1, 2$, label $y_i \in \{0, 1\}$, output $\hat{y}_i \in \{0, 1\}, i = 1, \dots, 50$ en $\vec{W}(w_0, w_1, w_2)$, met $w_j \in \mathbb{R}, j = 0, 1, 2$.

Men werkt dus in een vierdimensionale ruimte. De grafiek van de kostenfunctie is een driedimensionaal hypervlak.

Hier zal men dus partieel afleiden naar drie gewichten. De gradiënt is er een vector opgebouwd uit deze drie partiële afgeleiden.

In wat volgt, wordt de afgeleide van σ gebruikt. Er geldt dat $D\sigma(x) = \sigma(x)(1 - \sigma(x))$.

Bewijs.

$$\begin{aligned} D\sigma(x) &= D \frac{1}{1 + e^{-x}} \\ &= D(1 + e^{-x})^{-1} \\ &= -(1 + e^{-x})^{-2} D(1 + e^{-x}) \\ &= \frac{-1}{(1 + e^{-x})^2} e^{-x}(-1) \\ &= \frac{1}{1 + e^{-x}} \frac{e^{-x} + 1 - 1}{1 + e^{-x}} \\ &= \sigma(x)(1 - \sigma(x)) \end{aligned}$$

□

Voor $k = 0, 1, 2$ geldt:

$$\begin{aligned}
 & \frac{\partial BCE}{\partial w_k} \\
 &= \frac{\partial}{\partial w_k} \left[- \sum_{i=1}^n [\ln(\sigma(z_i)) \cdot y_i + \ln(1 - \sigma(z_i)) \cdot (1 - y_i)] \right] \\
 &= - \sum_{i=1}^n \left[\frac{\partial}{\partial w_k} [\ln(\sigma(z_i)) \cdot y_i] + \frac{\partial}{\partial w_k} [\ln(1 - \sigma(z_i)) \cdot (1 - y_i)] \right] \\
 &= - \sum_{i=1}^n \left[y_i \frac{1}{\sigma(z_i)} \frac{\partial}{\partial w_k} \sigma(z_i) + (1 - y_i) \frac{1}{1 - \sigma(z_i)} \frac{\partial}{\partial w_k} (1 - \sigma(z_i)) \right] \\
 &= - \sum_{i=1}^n \left[y_i \frac{1}{\sigma(z_i)} \sigma(z_i) (1 - \sigma(z_i)) \frac{\partial z_i}{\partial w_k} + (1 - y_i) \frac{1}{1 - \sigma(z_i)} \left(0 - \sigma(z_i) (1 - \sigma(z_i)) \frac{\partial z_i}{\partial w_k} \right) \right] \\
 &= - \sum_{i=1}^n \left[[y_i - y_i \sigma(z_i) - \sigma(z_i) + y_i \sigma(z_i)] \frac{\partial z_i}{\partial w_k} \right] \\
 &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial z_i}{\partial w_k} \right]
 \end{aligned}$$

$$\text{met } \frac{\partial z_i}{\partial w_k} = \frac{\partial}{\partial w_k} (w_1 x_{i1} + w_2 x_{i2} + w_0).$$

Dit geeft:

$$\begin{aligned}
 \frac{\partial BCE}{\partial w_0} &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial z_i}{\partial w_0} \right] \\
 &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial}{\partial w_0} (w_1 x_{i1} + w_2 x_{i2} + w_0) \right] \\
 &= - \sum_{i=1}^n [[y_i - \sigma(z_i)] \cdot 1] \\
 &= - \sum_{i=1}^n (y_i - \hat{y}_i) \\
 &= \sum_{i=1}^n (\hat{y}_i - y_i)
 \end{aligned}$$

en

$$\begin{aligned}
 \frac{\partial BCE}{\partial w_1} &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial z_i}{\partial w_1} \right] \\
 &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial}{\partial w_1} (w_1 x_{i1} + w_2 x_{i2} + w_0) \right] \\
 &= - \sum_{i=1}^n [[y_i - \sigma(z_i)] \cdot x_{i1}] \\
 &= - \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i1}) \\
 &= \sum_{i=1}^n ((\hat{y}_i - y_i) \cdot x_{i1})
 \end{aligned}$$

en

$$\begin{aligned}
 \frac{\partial BCE}{\partial w_2} &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial z_i}{\partial w_2} \right] \\
 &= - \sum_{i=1}^n \left[[y_i - \sigma(z_i)] \frac{\partial}{\partial w_2} (w_1 x_{i1} + w_2 x_{i2} + w_0) \right] \\
 &= - \sum_{i=1}^n [[y_i - \sigma(z_i)] \cdot x_{i2}] \\
 &= - \sum_{i=1}^n ((y_i - \hat{y}_i) \cdot x_{i2}) \\
 &= \sum_{i=1}^n ((\hat{y}_i - y_i) \cdot x_{i2}).
 \end{aligned}$$

Dus ook:

$$\begin{aligned}
 &\left(\frac{\partial BCE}{\partial w_0}, \frac{\partial BCE}{\partial w_1}, \frac{\partial BCE}{\partial w_2} \right) \\
 &= \left(\sum_{i=1}^n (\hat{y}_i - y_i), \sum_{i=1}^n ((\hat{y}_i - y_i) \cdot x_{i1}), \sum_{i=1}^n ((\hat{y}_i - y_i) \cdot x_{i2}) \right) \\
 &= \sum_{i=1}^n (\hat{y}_i - y_i, (\hat{y}_i - y_i) \cdot x_{i1}, (\hat{y}_i - y_i) \cdot x_{i2}) \\
 &= \sum_{i=1}^n ((\hat{y}_i - y_i) (1, x_{i1}, x_{i2})) \\
 \nabla \vec{E} &= \sum_{i=1}^n ((\hat{y}_i - y_i) \vec{X}_i).
 \end{aligned}$$

Men past de gewichten \vec{W} aan naar $\vec{W} - \eta \sum_{i=1}^n ((\hat{y}_i - y_i) \vec{X}_i)$,
of korter gezegd naar $\vec{W} - \eta \nabla \vec{E}$.

11.13 Trainingdata en testdata

Eens men een ML-model bekomen heeft, wil men dat gebruiken op nieuwe data. Het model moet dus ook goed presteren op nieuwe data. Men moet dan ook op een of andere manier de betrouwbaarheid van het model kunnen nagaan.

Voordat men het systeem begint te trainen, splitst men daarom de data op in trainingdata en testdata.

Zowel de kostenfunctie op de trainingdata als de kostenfunctie op de testdata zullen informeren over de prestaties van het systeem. Maar ook de accuraatheid is voor de prestatie een niet uit het oog te verliezen cijfer. De accuraatheid is het percentage datapunten dat juist wordt geclassificeerd.

Eens de kost geminimaliseerd is op de trainingdata, kan men spreken van een ML-model. Dit model wordt tot slot getest met de testdata. De accuraatheid en de kost bij de testdata worden berekend en vergeleken met die op de trainingdata.

11.14 Stochastic gradient descent

In de voorbeelden van het Perceptron die aan bod kwamen, werden de gewichten voor elk datapunt aangepast. Bij de voorbeelden met gradient descent gebeurde dat na elke epoch. In de praktijk wordt meestal *stochastic gradient descent* toegepast.

Bij de opbouw van het netwerk, kiest men de grootte van een *batch*. Bv. voor een trainingset van 60 000 elementen en een batch-grootte 128 zal het systeem dan vóór elke epoch een batch kiezen, het zal m.a.w. 128 elementen willekeurig kiezen uit de trainingset, elk met



Deze manier van werken wordt uit de doeken gedaan in de notebooks 'Zeeniveau Oostende' en 'Hoogte bomen en afmetingen stomata in het Amazonewoud' over regressie.

Er zijn trouwens nog cijfers die hier informatie kunnen over geven: precision en recall en de F1-score.

Stochastisch is een synoniem van willekeurig.

hun overeenkomstig label. Van elk datapunt in de batch wordt de nodige aanpassing van de gewichten bepaald maar nog niet uitgevoerd; van alle nodige aanpassingen wordt dan het gemiddelde genomen. Deze gemiddelde aanpassing wordt dan ná de epoch uitgevoerd. Bij stochastische gradient descent zal men de nodige aanpassing van de gewichten dus schatten a.d.h.v. een steekproef uit de trainingset. Omdat de dataset doorgaans zeer groot is, bespaart dat veel aan rekenkracht (Goodfellow et al., 2016).

11.15 Valideringsdata

Tijdens de verschillende epochs bij de training van het netwerk worden de gewichten op basis van de prestaties op de trainingset aangepast.

Om tijdens het trainen reeds een controle te hebben over hoe het systeem leert, splitst men zelfs de trainingdata nog eens op in de eigenlijke trainingdata en de valideringsdata. De verzameling valideringsdata en de verzameling trainingdata moeten disjunct zijn. Na elke epoch wordt dan nagegaan hoe het systeem generaliseert naar de valideringsdata. Men heeft zo een beter zicht op underfitting en overfitting (zie paragraaf 6.1).

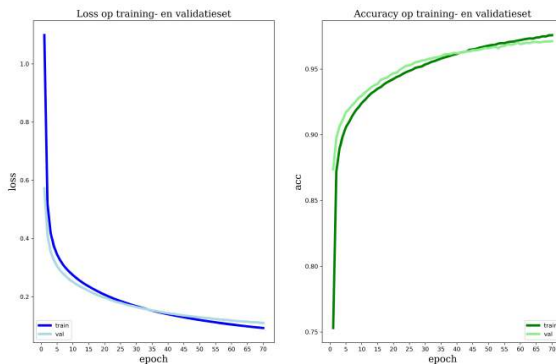
Elk systeem begint op een gegeven moment te overfitten, dan presteert het steeds beter op de trainingdata, maar de prestatie op de valideringsset verbetert niet meer, soms gaat die prestatie er zelfs op achteruit. Het heeft dan geen zin meer verder te gaan met de training. Men streeft immers niet naar een model dat het uitstekend doet op de trainingdata, maar wel naar een model dat zeer goed presteert op ongeziene data.

Bij de opbouw van het netwerk moet ook de learning rate gekozen worden. Men doet dat soms gebaseerd op de prestaties op de valideringsset. De valideringsset is na de training dus geen valabele dataset meer om de generalisatie van het netwerk mee te evalueren.

Daarom is er dus nog een afzonderlijke testset vereist. Men benoemt dit fenomeen met het begrip *information leaks*, ‘het lekken van informatie’ (Chollet, 2018).

Dus hoewel het netwerk niet direct getraind wordt op de valideringsset, kan het toch gaan overfitten op de valideringsset. Om het systeem écht te testen moet men data gebruiken waarvan het systeem op nog geen enkele manier info heeft over gekregen, zelfs niet op een indirecte manier. Eens het model klaar is, wordt het nog een keer getest op de testdata. Deze testdata mogen niet voorkomen in de training- en valideringsdata.

In het hoofdstuk 14 worden er concrete voorbeelden gegeven van under- en overfitting. De grafieken uit Figuur 11.47 komen uit paragraaf 14.3.1.



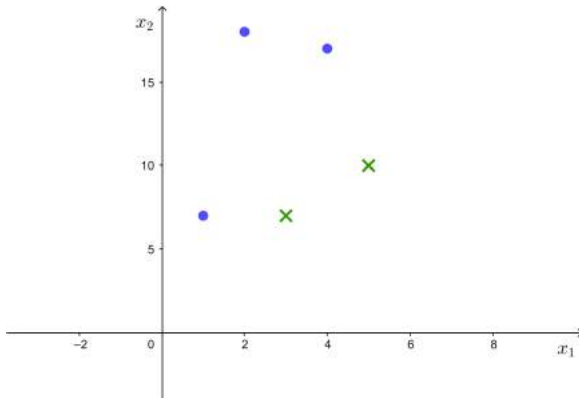
Figuur 11.47: MNIST 70 epochs. Het model overfits.

11.16 Een model om te classificeren aan het werk en het effect van een niet-lineaire activatiefunctie

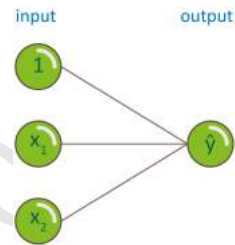
11.16.1 Voorbeeld lineaire scheiding

De grafiek in Figuur 11.48 toont punten die voorgesteld worden door blauwe bolletjes en punten die weergegeven worden als groene kruisjes. Deze twee klassen van punten zijn lineair scheidbaar. Stel dat er een neuraal

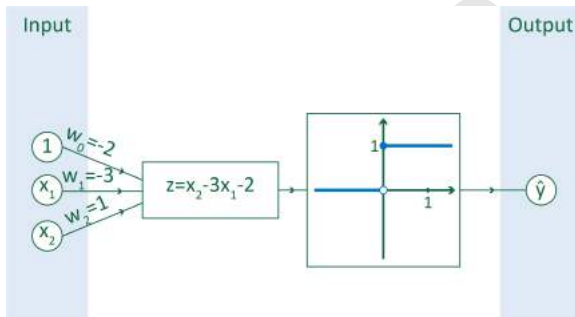
netwerk getraind is en dat er een model beschikbaar is om de twee klassen van elkaar te scheiden.



Figuur 11.48: De twee klassen zijn lineair scheidbaar.



Het model wordt geïllustreerd in Figuren 11.50 en 11.49.



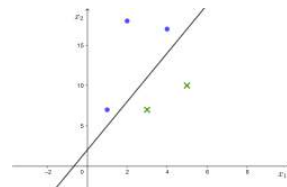
Figuur 11.49: Eenvoudige voorstelling van het model voor de lineaire scheiding.

Figuur 11.51 toont een voorbeeld van een rechte die de klassen van elkaar scheidt.

Deze scheidingsrechte heeft als vergelijking

$$x_2 = 3x_1 + 2.$$

De punten die boven deze rechte liggen, voldoen aan $x_2 > 3x_1 + 2$, wat equivalent is met $-3x_1 + x_2 > 2$ en met $-3x_1 + x_2 - 2 > 0$.



Figuur 11.51: Een mogelijke scheidingslijn.

De punten die onder deze rechte liggen, voldoen aan $x_2 < 3x_1 + 2$, wat equivalent is met $-3x_1 + x_2 < 2$ en met $-3x_1 + x_2 - 2 < 0$.

Merk het verband op tussen de coëfficiënten in de vergelijking van de scheidingsrechte en de waarden van de gewichten in het model.

Er wordt dus een lineaire combinatie gemaakt van de inputwaarden x_1 , x_2 en een drempelwaarde 2. Vervolgens wordt op de lineaire combinatie de Heaviside-functie toegepast. Bij een positief resultaat wordt er afgebeeld op 1, anders op 0.

De bekomen \hat{y} -waarde kan dus 1 of 0 zijn: 1 voor de punten boven de rechte (het gebied van de blauwe bolletjes) en 0 voor de punten eronder (het gebied van de groene kruisjes).

Test op het punt (1,7) en het punt (5,10):

$-3 \cdot 1 + 1 \cdot 7 - 2 = 2 > 0$ dus $\hat{y} = 1$, punt boven de rechte (gebied blauwe bolletjes);

$-3 \cdot 5 + 1 \cdot 10 - 2 = -7 < 0$ dus $\hat{y} = 0$, punt onder de rechte (gebied groene kruisjes).

In matrixnotatie:

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}, W = \begin{bmatrix} -2 \\ -3 \\ 1 \end{bmatrix},$$

$$X \mapsto W^T \cdot X = [z] \mapsto [H(z)] = [\hat{y}], \text{ met } \hat{y} \in \{0,1\}.$$

11.16.2 XOR-functie of 'exclusieve of'-functie, één hidden layer

Beschouw de grafiek in Figuur 11.52.

Deze grafiek is een voorstelling van de XOR-functie, een booleaanse operator op twee binaire waarden. Als juist één van deze binaire waarden 1 is, dan is het resultaat van de XOR-functie 1, anders is het resultaat 0.

De bias w_0 is -2 .

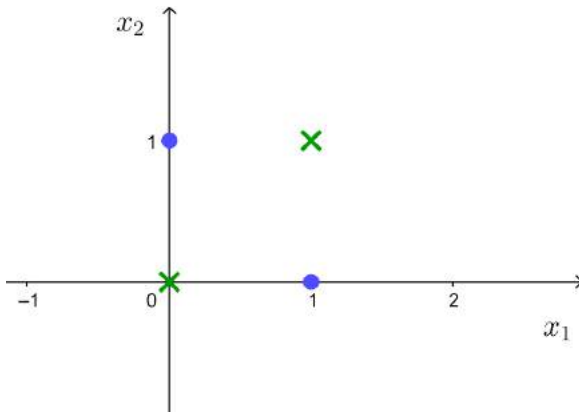
Het voorschrift van deze niet-lineaire functie luidt als volgt:

$$H: \mathbb{R} \rightarrow \mathbb{R}: x \mapsto \begin{cases} 0, & \text{als } x < 0 \\ 1, & \text{als } x \geq 0. \end{cases}$$

$$z = -3 \cdot x_1 + 1 \cdot x_2 - 2.$$

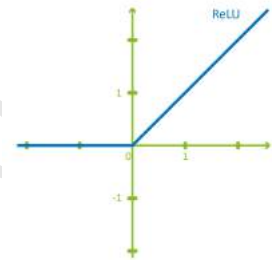
Het voorschrift van de XOR-functie:

$$\text{XOR}: \begin{cases} (0,0) \mapsto 0 \\ (0,1) \mapsto 1 \\ (1,0) \mapsto 1 \\ (1,1) \mapsto 0 \end{cases}$$



Figuur 11.52: XOR. De punten zijn niet lineair scheidbaar.

De afgebeelde punten zijn niet lineair scheidbaar. Er is dus geen rechte te vinden die de punten van elkaar scheidt. De punten kunnen toch van elkaar gescheiden worden via een neuraal netwerk dat voor de activatiefunctie gebruikmaakt van de ReLU-functie (zie Figuur 11.53).



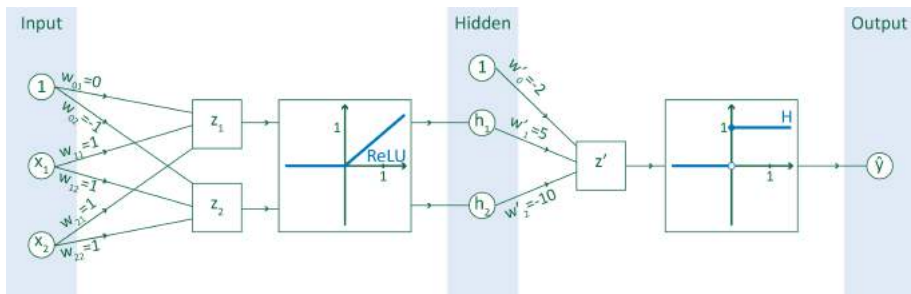
Figuur 11.53: ReLU-functie.

Het voorschrift van de niet-lineaire ReLU-functie luidt als volgt:

$$\text{ReLU} : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} 0, & \text{als } x < 0 \\ x, & \text{als } x \geq 0. \end{cases}$$

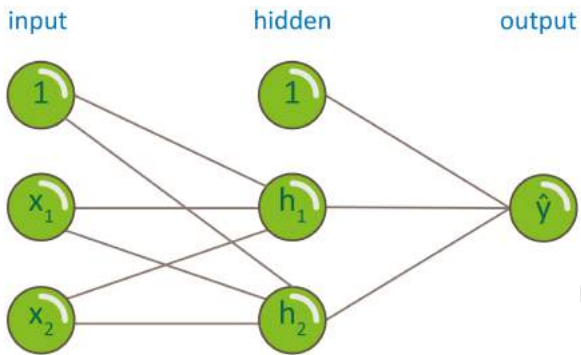
Kort: $\text{ReLU}(x) = \max\{0, x\}$.

Figuur 11.54 geeft een voorstelling van een neuraal netwerk dat de punten scheidt. Er worden twee lineaire



Figuur 11.54: Model voor de scheiding.

combinaties gemaakt van de inputwaarden x_1, x_2 en een drempelwaarde. Vervolgens wordt op de lineaire combinaties de ReLU-functie toegepast. Van de bekomen outputwaarden en een drempelwaarde wordt opnieuw een lineaire combinatie gemaakt; hierop wordt de Heaviside-functie toegepast. Bij een positief resultaat wordt de input afgebeeld op $\hat{y} = 1$, anders op $\hat{y} = 0$. Figuur 11.55 toont een vereenvoudigde voorstelling van het model.



Figuur 11.55: Eenvoudige voorstelling van het model voor de scheiding.

In matrixnotatie, na toevoeging van een extra neuron:

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}, W = \begin{bmatrix} 0 & -1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}, W' = \begin{bmatrix} -2 \\ 5 \\ -10 \end{bmatrix}.$$

$$X \mapsto W^T \cdot X = Z \mapsto \text{ReLU}(Z) = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix},$$

hieraan wordt ook een extra neuron 1 toegevoegd; men

$$\text{bekomt } H = \begin{bmatrix} 1 \\ h_1 \\ h_2 \end{bmatrix} \text{ van de } \textit{hidden layer}.$$

$$H \mapsto W'^T \cdot H = [z'] \mapsto [H(z')] = [\hat{y}],$$

met $\hat{y} \in \{0,1\}$.

Grafisch vertaalt zich dat in de grafieken van Figuren 11.56 en 11.57.

De scheidingsrechte

uit Figuur 11.57 heeft als vergelijking $x_2 = \frac{1}{2}x_1 - \frac{1}{5}$. De punten die boven deze rechte liggen, voldoen aan $x_2 > \frac{1}{2}x_1 - \frac{1}{5}$, wat equivalent is met $\frac{1}{2}x_1 - x_2 < \frac{1}{5}$, met $5x_1 - 10x_2 < 2$ en met $5x_1 - 10x_2 - 2 < 0$. De punten die onder deze rechte liggen, voldoen aan $x_2 < \frac{1}{2}x_1 - \frac{1}{5}$, wat equivalent is met $\frac{1}{2}x_1 - x_2 > \frac{1}{5}$, met $5x_1 - 10x_2 > 2$ en met $5x_1 - 10x_2 - 2 > 0$.

Merk het verband op tussen de coëfficiënten in de vergelijking van deze scheidingsrechte en de waarden van de gewichten W' in het model.

Test het model op het punt $(0,0)$:

$(1,0,0)$ uit de *input layer* wordt afgebeeld op $(1,0,0)$ in de *hidden layer*; $5 \cdot 0 - 10 \cdot 0 - 2.1 = -2 < 0$ dus $\hat{y} = 0$ (gebied groene kruisjes).

Test het model op het punt $(1,0)$:

$(1,1,0)$ uit de *input layer* wordt afgebeeld op $(1,1,0)$ in de *hidden layer*; $5 \cdot 1 - 10 \cdot 0 - 2.1 = 3 > 0$ dus $\hat{y} = 1$ (gebied blauwe bolletjes).

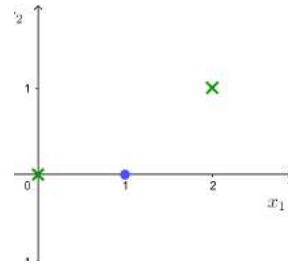
Onthoud: door gebruik te maken van de ReLU-functie, werden de punten lineair scheidbaar. De laag tussen de *input layer* en de *output layer* noemt men een *hidden layer*.

11.16.3 Model met één hidden layer

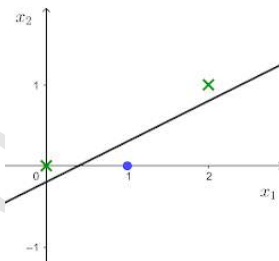
Beschouw de grafiek in Figuur 11.58.

Er zijn twee groepen van punten of anders gezegd twee klassen. Deze punten kunnen van elkaar gescheiden worden, bv. met een parabool of met twee rechten; zie hiervoor de Figuren 11.59 en 11.60.

De scheiding kan gebeuren met een neurale netwerk met één hidden layer dat voor de activatiefunctie gebruikmaakt van de sign-functie (zie Figuur 11.61).

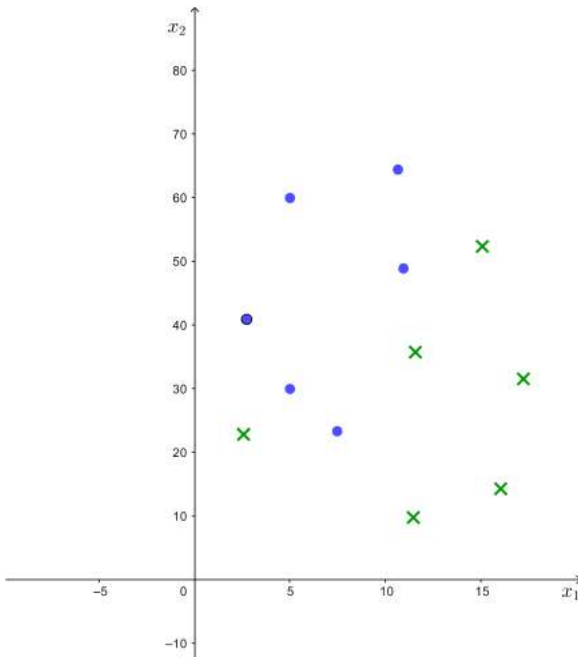


Figuur 11.56: Hidden layer. Linear scheidbaar.

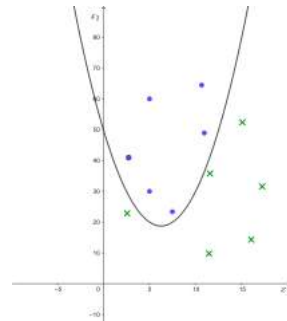


Figuur 11.57: Gescheiden door een rechte.

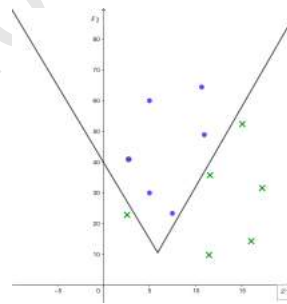
$$\begin{aligned} z_1 &= x_1 + x_2 \\ z_2 &= x_1 + x_2 - 1 \\ z' &= 5 \cdot h_1 - 10 \cdot h_2 - 2 \end{aligned}$$



Figuur 11.58: De klassen zijn niet lineair scheidbaar.



Figuur 11.59: Scheiden met een parabool.



Figuur 11.60: Scheiden met twee rechten.

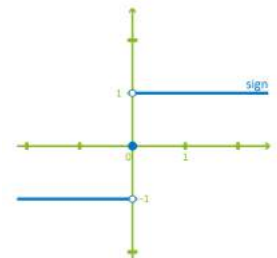
Het voorschrift van de niet-lineaire sign-functie luidt als volgt:

$$\text{sign} : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} -1, & \text{als } x < 0 \\ 0, & \text{als } x = 0 \\ 1, & \text{als } x > 0. \end{cases}$$

De twee scheidingsrechten in Figuur 11.60 hebben als vergelijking $x_2 = -5 x_1 + 40$ voor de dalende rechte en $x_2 = 5,2 x_1 - 20$ voor de stijgende.

De blauwe punten bv. liggen boven deze twee half-rechten en voldoen aan $\begin{cases} x_2 > -5 x_1 + 40 \\ x_2 > 5,2 x_1 - 20 \end{cases}$, wat

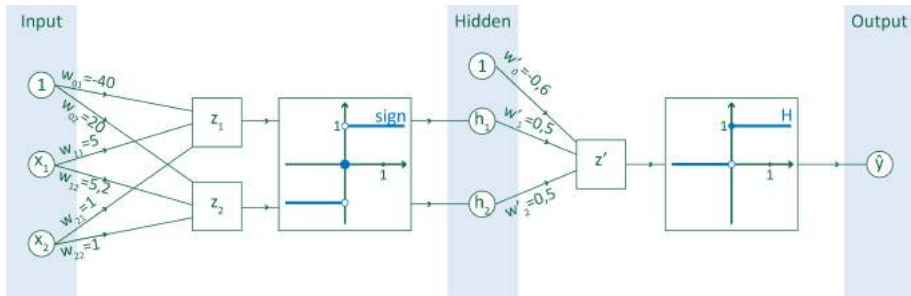
equivalent is met $\begin{cases} 5 x_1 + x_2 > 40 \\ -5,2 x_1 + x_2 > -20 \end{cases}$ en met



Figuur 11.61: Sign-functie.

$$\begin{cases} 5x_1 + x_2 - 40 > 0 \\ -5,2x_1 + x_2 + 20 > 0. \end{cases}$$

Een voorstelling van het model staat in Figuren 11.62 en 11.63. Er worden dus twee lineaire combinaties



Figuur 11.62: Neuraal netwerk dat de punten scheidt, met activatiefuncties sign en Heaviside.

gemaakt van de inputwaarden x_1 en x_2 , een met drempelwaarde 40 en een met drempelwaarde -20 . Vervolgens wordt op de lineaire combinaties de sign-functie toegepast om de waarden in de hidden layer te bekoemen. Er wordt dan een lineaire combinatie gemaakt van deze waarden en een drempelwaarde, waarop de Heaviside-functie wordt toegepast. Bij een positief resultaat wordt de input afgebeeld op $\hat{y} = 1$, anders op $\hat{y} = 0$.

De outputwaarde \hat{y} kan twee waarden aannemen: 0 voor het gebied van de groene kruisjes en 1 voor het gebied van de blauwe bolletjes.

In matrixnotatie:

$$X = \begin{bmatrix} 1 \\ x_1 \\ x_2 \end{bmatrix}, W = \begin{bmatrix} -40 & 20 \\ 5 & -5,2 \\ 1 & 1 \end{bmatrix}, W' = \begin{bmatrix} -0,6 \\ 0,5 \\ 0,5 \end{bmatrix}.$$

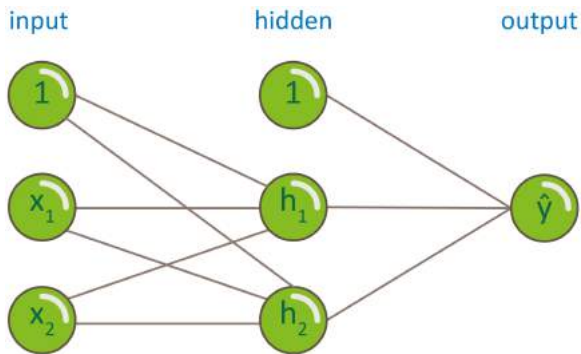
$$X \mapsto W^T \cdot X = Z \mapsto \text{sign}(Z) = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix},$$

hieraan wordt een extra neuron toegevoegd,

$$H = \begin{bmatrix} 1 \\ h_1 \\ h_2 \end{bmatrix}.$$

$$H \mapsto W'^T \cdot H = [z'] \mapsto [H(z')] = [\hat{y}],$$

met $\hat{y} \in \{0,1\}$.



Figuur 11.63: Vereenvoudigde voorstelling van het model voor de scheiding.

Test op de punten $(0,0)$ en $(5,30)$:

$$5 \cdot 0 + 1 \cdot 0 - 40 = -40 < 0 \text{ en } -5,2 \cdot 0 + 1 \cdot 0 + 20 = 20 > 0 \text{ dus } H = \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}, \text{ dus } \hat{y} = 0 \text{ (gebied}$$

groene kruisjes).

$$5 \cdot 5 + 1 \cdot 30 - 40 = 15 > 0 \text{ en } -5,2 \cdot 5 + 1 \cdot 30 + 20 = 24 > 0 \text{ dus } H = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \text{ dus } \hat{y} = 1 \text{ (gebied}$$

blauwe bolletjes).

$$\begin{aligned} z_1 &= 5 \cdot x_1 + 1 \cdot x_2 - 40 \\ z_2 &= -5,2 \cdot x_1 + 1 \cdot x_2 + 20 \\ z' &= 0,5 \cdot h_1 + 0,5 \cdot h_2 - 0,6 \end{aligned}$$

Onthoud: y en \hat{y} hebben evenveel mogelijke waarden als er klassen zijn.

In de netwerken, met een verborgen laag, uit de Figuren 11.54, 11.55, 11.62 en 11.63 zijn alle neuronen van de opeenvolgende lagen met elkaar verbonden; men spreekt van *dense layers* of *fully connected layers*.

Samengevat

Het Perceptron, het eerste neurale netwerk, heeft enkel een invoer- en een uitvoerlaag. Het Perceptron is geschikt om lineair scheidbare klassen van elkaar te scheiden. Het Perceptron leert door gelabelde datapunten in het netwerk in te voeren. Er wordt een gewogen som genomen van de inputneuronen, gevolgd door een niet-lineaire activatiefunctie, de Heaviside-functie. Het netwerk deelt het betreffende datapunt in bij een klasse. Deze uitvoer wordt vergeleken met het gegeven label. Na elk punt worden de gewichten aangepast, evenredig met de *learning rate* en de eventuele fout die gemaakt werd.

Als klassen niet lineair scheidbaar zijn, dan zal het Perceptron niet volstaan. Daarvoor zijn er andere, meer complexe, neurale netwerken. Deze netwerken hebben meer lagen en gebruiken ook andere activatiefuncties, zoals ReLU, de sigmoïde-functie en de sign-functie.

Bij elk neuraal netwerk streeft men ernaar om de fout zo klein mogelijk te maken. Men zal de kostenfunctie minimaliseren met (*stochastic*) *gradient descent*. Voor een binaire classificatie gebruikt men bv. een netwerk met één verborgen laag met de sigmoïde-functie als activatiefunctie; *binary cross-entropy*, *BCE*, is de kostenfunctie.

Behalve classificatie kan men met neurale netwerken ook regressieproblemen aanpakken, bv. lineaire regressie, waarbij men op zoek gaat naar de best passende rechte bij de data. Bij lineaire regressie is er eigenlijk geen activatiefunctie nodig; men zegt dat de identiteit de activatiefunctie is. De gemiddelde kwadratische afwijking, *mean squared error*, *MSE*, doet er dienst als kostenfunctie.

Voordat men een netwerk traint, splitst men de data op in trainingdata, valideringsdata en testdata. De kostenfunctie op deze data, maar ook de accuraatheid zullen informeren over de prestaties van het model.

Bij het trainen moet men beducht zijn voor *underfitting* (de prestaties van het systeem kunnen bv. door het kiezen van een ander model of door meer te trainen nog verbeteren) en *overfitting* (het systeem presteert steeds beter op de trainingdata, maar de prestaties op de valideringsdata verbeteren niet meer).

CONVOLUTIES

12.1 Een beeld filteren

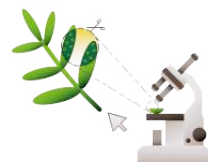
Image filtering wordt bv. gebruikt om ruis uit een foto te verwijderen, om het contrast te verscherpen of om randen te detecteren.

Er bestaan niet-lineaire filters zoals de mediaanfilter, waarbij men een pixelwaarde vervangt door de mediaan van de pixelwaarde zelf en de waarde van zijn burens. De mediaanfilter is uitermate geschikt om peper-en-zoutruis uit een afbeelding te verwijderen.

Daarnaast zijn er nog de convoluties. Voorbeelden zijn de gemiddelde-filter, de Sobelfilter, de Gaussfilter en de Prewittfilter. Convolutionele netwerken zijn genoemd naar deze filters. Convoluties worden toegepast in de convolutionele netwerken, dus ook in het KIKS-netwerk. Omdat deze filters werken door een lineaire combinatie te nemen van de pixelwaarden van een pixel en zijn burens, noemt men de convoluties ook lineaire filters.

12.2 Convolutie

Met convoluties kan men dus op zoek gaan naar verschillende kenmerken in een afbeelding. Men kan er

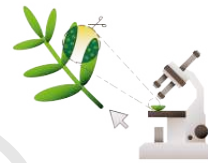


Bekijk het effect van een convolutie in de notebook 'Convoluties'.

bv. verticale en horizontale lijnen mee detecteren, ruis in een beeld mee verminderen of het contrast in een beeld verzachten.

Convolutie

De convolutie is een wiskundige bewerking die enkel gebruikmaakt van optellen en vermenigvuldigen. Het komt erop neer dat men aan een pixel een bepaald gewicht geeft en men daaraan gewogen waarden van de omliggende pixels toevoegt.



12.3 De convolutiebewerking

Bij een 'convolutie' laat men een 'filter' over een afbeelding 'glijden'; er wordt daarbij telkens één pixel opgeschoven. Zowel de afbeelding als de filter zijn matrices of tensoren. De elementen van de filter en de elementen van de matrix van de afbeelding worden elementsgewijs vermenigvuldigd en erna worden deze producten opgeteld.

Bekijk de notebook 'Convolutie: de bewerking'.

Hoe verloopt dit concreet met een grijswaardenafbeelding?

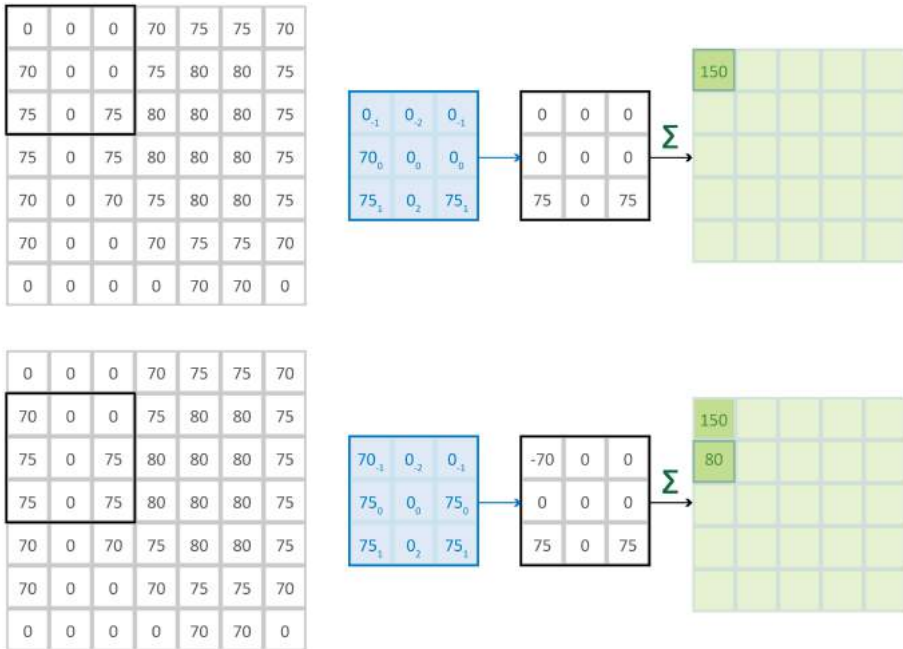
Beschouw de Sobelfilter.

Deze filter is een 3×3 -matrix:
$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Eerst 'flipt' men de filter: men manipuleert de matrix door de eerste en de derde rij te verwisselen en vervolgens hetzelfde te doen met de eerste en de derde kolom.

Men bekomt de matrix:
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}.$$

Deze matrix laat men over de afbeelding glijden. In Figuur 12.1 wordt het rekenwerk geïllustreerd.



Figuur 12.1: Convolutie-bewerking. Gebaseerd op afbeelding van Rob Robinson (Robinson, 2017).

Er rest niets anders dan de overeenkomstige elementen van de matrix en de filter op elke positie met elkaar te vermenigvuldigen en erna alle producten op te tellen. Bv. in de eerste positie: $0 \cdot (-1) + 0 \cdot (-2) + 0 \cdot (-1) + 70 \cdot 0 + 0 \cdot 0 + 0 \cdot 0 + 75 \cdot 1 + 0 \cdot 2 + 75 \cdot 1 = 150$.

Het resultaat van de convolutie is een matrix die kleiner is dan de oorspronkelijke matrix. Men 'verliest' twee pixels in elke dimensie.

Er bestaan technieken om de grootte van de afbeelding te behouden na convolutie. In de neurale netwerken van KIKS worden deze technieken niet toegepast. In Python voert de functie `convolve2d()` uit de submodule `signal` voor beeldverwerking van de module `SciPy` een convolutiebewerking uit, dus zowel het flippen van de matrix, als het rekenwerk.

In de functie `convolve2d()` staat de waarde van de parameter `mode` op `'valid'`, wat betekent dat de grootte van de afbeelding niet behouden moet blijven.

12.4 Convolutie uitvoeren in Python

12.4.1 Implementatie van het gegeven rekenvoor- beeld

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt
3 >>> import scipy.signal
4
5 >>> filter = np.array([[1,2,1],
6                       [0,0,0],
7                       [-1,-2,-1]])
8 >>> afb = np.array([[0,0,0,70,75,75,70],
9                   [70,0,0,75,80,80,75],
10                  [75,0,75,80,80,80,75],
11                  [75,0,75,80,80,80,75],
12                  [70,0,70,75,80,80,75],
13                  [70,0,0,70,75,75,70],
14                  [0,0,0,0,70,70,0]])

```

De convolutie wordt uitgevoerd:

```

1 >>> afb_sobel = scipy.signal.convolve2d(
    afb, filter, mode="valid")

```

Het resultaat kan opgevraagd worden:

```

1 >>> print(afb_sobel)
2 [[ 150  160  100   25   20]
3  [  80  155   85    5    0]
4  [ -10  -15  -15   -5    0]
5  [ -80 -160 -100  -25  -20]
6  [-140 -215 -230 -105 -105]]
7 >>> print(afb.shape)
8 (7, 7)
9 >>> print(afb_sobel.shape)
10 (5, 5)

```

afb is een 7×7 -matrix en afb_sobel is een 5×5 -matrix.

12.4.2 Voorbeeld met convolutie op MNIST data-set

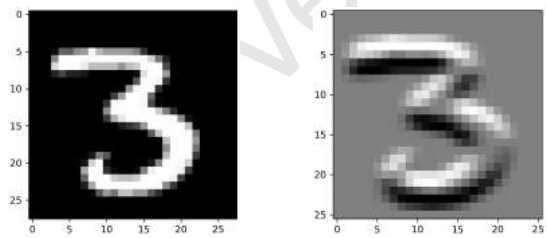
Het dertiende element uit de MNIST dataset is een drie. Het betreft een 28x28-matrix:

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 12 99 91 142 155 246 182 155 155 155 155 131 52 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 138 254 254 254 254 254 254 254 254 254 254 252 210 122 33 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 220 254 254 254 235 189 189 189 189 150 189 205 254 254 254 75 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 35 74 35 35 25 0 0 0 0 0 0 0 13 224 254 254 153 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 90 254 254 247 53 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 6 152 246 254 254 49 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 66 158 254 254 249 103 8 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 54 251 254 254 254 248 74 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 140 254 254 254 254 254 254 202 125 45 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 58 181 234 254 254 254 254 254 254 252 140 22 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 30 50 73 155 253 254 254 254 254 191 2 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 91 200 254 254 254 254 118 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 192 254 254 254 254 154 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 141 254 254 254 116 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 25 126 86 0 0 0 0 0 0 0 0 3 188 254 254 250 61 0 0 0 0 0
0 0 0 0 0 0 0 0 24 209 254 15 0 0 0 0 0 0 23 137 254 254 254 209 0 0 0 0 0
0 0 0 0 0 0 0 0 168 254 254 48 9 0 0 9 127 241 254 254 255 242 63 0 0 0 0 0
0 0 0 0 0 0 0 101 254 254 254 205 190 190 205 254 254 254 254 242 67 0 0 0 0 0
0 0 0 0 0 0 0 33 166 254 254 254 254 254 254 254 250 138 55 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 7 88 154 116 194 194 154 154 88 49 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

Men kan deze afbeelding filteren met de Sobelfilter.
 Figuur 12.2 toont het resultaat van deze convolutie.



Figuur 12.2: Convolutie met Sobelfilter op '3' uit MNIST dataset.

Men kan de grootte van de afbeeldingen opvragen:

```

1 >>> print(drie.shape)
2 (28, 28)
3 >>> print(drie_sobel.shape)
4 (26, 26)

```

12.4.3 Voorbeeld: verticale lijnen detecteren op een foto

In de notebook 'Convolutie' wordt op een grijswaardenfoto van bamboe een filter toegepast om verticale lijnen te detecteren.

Om verticale lijnen te detecteren, gebruikt men er de filter $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$.

Men geeft deze filter in in Python als een NumPy-array. Om de convolutie uit te voeren wordt de functie `convolve2d()` gebruikt uit de submodule `signal` voor beeldverwerking van de Python-module SciPy. Merk op dat het 'flippen' van de matrix reeds inbegrepen zit in de functie `convolve2d()`.

De modules NumPy, `scipy.signal` en `matplotlib.pyplot` worden geïmporteerd. De filter wordt ingevoerd en een foto wordt ingeladen; de foto is een grijswaardenafbeelding.

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt
3 >>> import scipy.signal
4
5 >>> vertic_filter = np.array([[ -1, 0, 1],
6                               [ -1, 0, 1],
7                               [ -1, 0, 1]])
8 >>> bamboe = np.load("bamboe.npy")

```

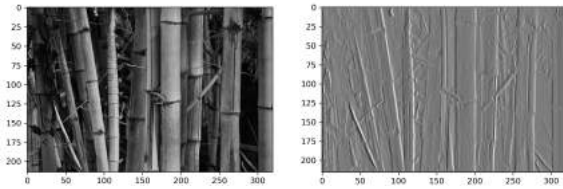
De convolutie wordt uitgevoerd:

```

1 >>> bamboe_vertic = scipy.signal.
    convolve2d(bamboe,
2              vertic_filter,
3              mode="valid")

```

Het resultaat kan bekeken worden in Figuur 12.3.



Figuur 12.3: Originele en gefilterde foto. De linkse foto is een afgeleide van het werk van McGrath (2007).

12.5 Feature map

In één laag van een convolutioneel netwerk gebeuren er meerdere convolutiebewerkingen (zie Figuur 12.5). Bv. in een laag met 24 neuronen worden er 24 convolutiebewerkingen gedaan. Elke convolutie levert een matrix op. Op deze matrix wordt vervolgens een niet-lineaire activatiefunctie toegepast, wat opnieuw een matrix oplevert.

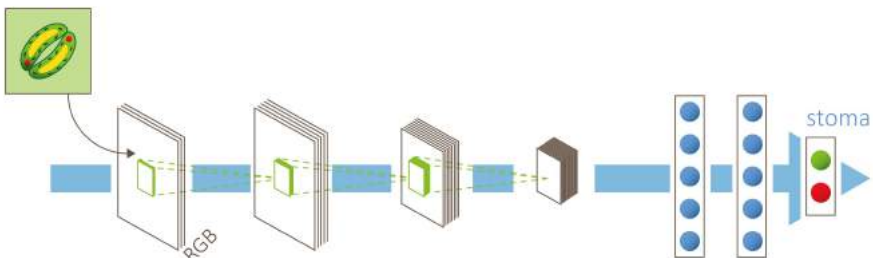
Deze 24 matrices worden dan samengesteld tot een tensor, de zogenaamde *feature map* (zie Figuur 12.4). Deze feature map dient dan als input voor de volgende laag.

In elke laag van het convolutionele neurale netwerk worden de tensoren door de convoluties getransformeerd in een nieuwe representatie van de gegevens.

Voor de niet-lineaire activatiefunctie ReLU: zie hoofdstukken 11 en 13.



Figuur 12.4: Feature map.



Figuur 12.5: Neuraal netwerk van KIKS.

12.6 Na het convolutionele netwerk

De feature map die de uitvoer was van het convolutionele netwerk, wordt gegeven aan een netwerk met dense layers (zie tweede deel van het netwerk in Figuur 12.5). Uit hoofdstuk 11 blijkt dat de invoer daar gebeurt met 1D-tensoren. Daarom voert men op de feature map eerst een operatie uit die alle elementen van de feature map achter elkaar plaatst: dat gebeurt met `flatten()` of met `reshape()`.

Zie ook hoofdstuk 14.

Voorbeeld:

```

1 >>> afb_sobel.shape
2 (5, 5)
3 >>> afb_sobel.flatten()
4 array([150, 160, 100, 25, 20, 80, 155, 85,
5         5, 0, -10, -15, -15, -5, 0, -80,
6        -160, -100, -25, -20, -140, -215,
7        -230, -105, -105])
8 >>> afb_sobel.reshape(5*5)
9 array([150, 160, 100, 25, 20, 80, 155, 85,
10        5, 0, -10, -15, -15, -5, 0, -80,
11        -160, -100, -25, -20, -140, -215,
12        -230, -105, -105])

```

12.7 Succes van deep learning

Deep learning heeft zoveel succes omdat het bij veel problemen betere resultaten boekt dan de klassieke machine learning netwerken, maar ook omdat het een belangrijk aspect ervan, de *feature engineering*, volledig automatiseert (Chollet, 2018). Men moet niet meer zelf op zoek naar geschikte filters om relevante kenmerken uit de data te extraheren. Het systeem gaat zelf op zoek naar bruikbare filters.

Samengevat

In de convolutionele netwerken werken filters in op afbeeldingen om er kenmerken in op te sporen. Deze convoluties zijn lineaire filters: er wordt een lineaire combinatie genomen van de pixelwaarden van een pixel en zijn burens.

In één laag van een convolutioneel netwerk gebeuren er meerdere convolutiebewerkingen. Elke convolutie levert een matrix op, waarop vervolgens een niet-lineaire activatiefunctie inwerkt en opnieuw een matrix oplevert. De matrices worden samengesteld tot een tensor, de *feature map*.

De uitvoer van elke laag is dus een *feature map* die dient als input voor de volgende laag. In elke convolutionele laag wordt de ingevoerde tensor getransformeerd in een nieuwe representatie van de gegevens.

Om huidmondjes te herkennen, gebruikt KIKS een diep neurale netwerk dat bestaat uit een convolutioneel netwerk, gevolgd door een netwerk met *dense layers*.

Versie

RELU EN MAX POOLING

13.1 *ReLU en max pooling in een convolutioneel neurale netwerk*

Om de huidmondjes op een microfoto te detecteren, doen de onderzoekers van KIKS een beroep op een diep neurale netwerk dat bestaat uit een convolutioneel neurale netwerk gevolgd door een *feedforward* netwerk (zie ook Figuur 12.5).

Het is een *Sequential model*, een model dat bestaat uit aaneengeschakelde lagen. Het is opgebouwd met een convolutioneel netwerk gevolgd door een klassiek neurale netwerk. Het convolutionele netwerk bezit enkele convolutionele lagen, afgewisseld met *max pooling*-lagen; het netwerk dat erop volgt, is er een met *dense layers*.

De convoluties gebruiken filters die over de input-tensoren worden gelegd. In elke laag van het convolutioneel neurale netwerk worden de tensoren door de convoluties getransformeerd in een nieuwe representatie van de gegevens.

De convoluties worden gevolgd door een ReLU-functie (een niet-lineaire functie) met aansluitend een *max pooling*-operatie. ReLU toepassen op een tensor gebeurt elementsgewijs (zie ook hoofdstuk 11).

Zie ook hoofdstukken 12 en 14.

Bij een *feedforward* netwerk ontvangt elk neuron informatie van de neuronen uit de vorige laag. Zo gaat de informatie 'voorwaarts' door het netwerk, laag na laag, t.e.m. de neuronen in de uitvoerlaag (zie paragraaf 6.2).

Dense layers, of *fully connected layers*, zijn lagen waarbij alle neuronen van de opeenvolgende lagen met elkaar verbonden zijn (zie hoofdstuk 11).

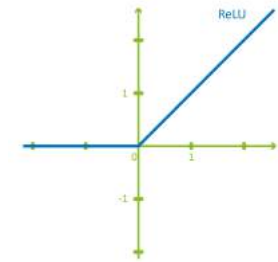
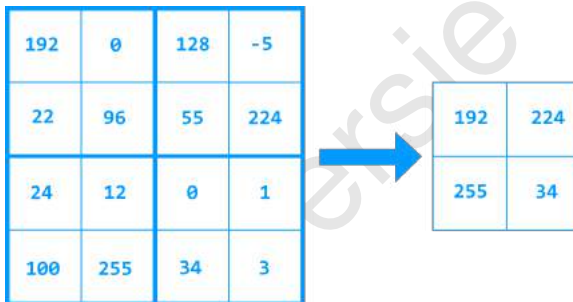
De ReLU-functie wordt als volgt gedefinieerd:

$$\text{ReLU} : \mathbb{R} \rightarrow \mathbb{R} : x \mapsto \begin{cases} 0, & \text{als } x < 0 \\ x, & \text{als } x \geq 0. \end{cases}$$

Kort gezegd: $\text{ReLU}(x) = \max(0, x)$.

ReLU zorgt ervoor dat de kenmerken die het minst tot een goede classificatie leiden, worden verzwakt. ReLU zal er bv. bij grijswaardenafbeeldingen voor zorgen dat de lichte kleuren behouden blijven en de donkere kleuren worden afgevlakt (zie Figuur 13.4).

Max pooling neemt uit een vierkante matrix, waarvan men eerst de wiskundige dimensie kiest, de grootste waarde. Dit wordt geïllustreerd in Figuur 13.2 waar de grootste waarde wordt genomen uit elk vak van 2×2 .



Figuur 13.1: Grafiek van de ReLU-functie.

Figuur 13.2: Max pooling-operatie

Max pooling zal de kenmerken die van belang zijn om tot een goede classificatie te komen, versterken en de andere kenmerken weglaten. Bovendien zorgt max pooling ervoor dat de tensoren kleiner worden, waardoor er minder rekenkracht nodig is door de computer.

Door max pooling zal het netwerk ook minder snel overfitten. Men kan natuurlijk niet onbeperkt max pooling-operaties toepassen, er moeten nog voldoende pixels overblijven.



In de notebook 'ReLU en max pooling' worden deze effecten geïllustreerd.

13.2 ReLU en max pooling in Python

Om ReLU en max pooling te definiëren in Python wordt de module NumPy gebruikt. ReLU en max pooling moeten op een tensor kunnen inwerken.

ReLU definiëren:

```

1 >>> import numpy as np
2
3 >>> def relu(tensor):
4 >>>     """ReLU(x) = max(0,x)."""
5 >>>     return np.maximum(0, tensor)

```

Max pooling voor een 2x2-matrix definiëren:

```

1 >>> def maxpool(tensor):
2 >>>     """Neemt van elk vak van 2x2 de grootste waarde."
3 >>>     # NumPy array van gewenste grootte met allemaal nullen
4 >>>     mp = np.zeros((tensor.shape[0]//2,
5 >>>                    tensor.shape[1]//2))
6 >>>     # NumPy array opvullen
7 >>>     for i in range(0, tensor.shape[0]-1, 2):
8 >>>         for j in range(0, tensor.shape[1]-1, 2):
9 >>>             max = np.max([tensor[i][j], tensor[i][j+1],
10 >>>                           tensor[i+1][j], tensor[i+1][j+1]])
11 >>>             k = i // 2 # // gebruiken om int te bekomen
12 >>>             l = j // 2
13 >>>             mp[k][l] = max
14 >>>     return mp

```

Een voorbeeld:

- Het resultaat van ReLU op $A = \begin{bmatrix} 30 & -20 & 0 & -120 \\ -80 & 30 & 80 & 50 \\ 20 & -30 & -90 & -120 \end{bmatrix}$

is $\begin{bmatrix} 30 & 0 & 0 & 0 \\ 0 & 30 & 80 & 50 \\ 20 & 0 & 0 & 0 \end{bmatrix}$.

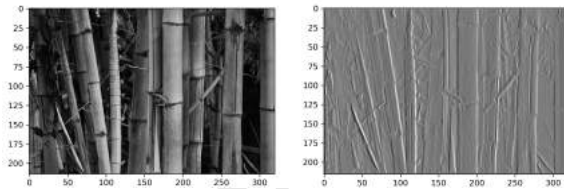
$$\bullet \text{ Is } B = \begin{bmatrix} 30 & 20 & 130 & 120 & 50 & 50 \\ 80 & 130 & 80 & 50 & 120 & 40 \\ 20 & 30 & 90 & 120 & 100 & 100 \\ 50 & 90 & 30 & 30 & 10 & 10 \\ 50 & 40 & 120 & 120 & 50 & 130 \\ 130 & 120 & 130 & 10 & 100 & 40 \end{bmatrix}, \text{ dan}$$

$$\text{geeft max pooling } \begin{bmatrix} 130 & 130 & 120 \\ 90 & 120 & 100 \\ 130 & 130 & 130 \end{bmatrix}.$$

Merk op dat na ReLU het aantal elementen in de tensor behouden is en na max pooling het aantal gereduceerd is.

13.3 Effect van ReLU

Herneem het voorbeeld met de foto van de bamboeplant, Figuur 12.3. De verticale lijnen worden gedetecteerd met een convolutie.



Figuur 13.3: Originele en gefilterde foto. De linkse foto is een afgeleide van het werk van McGrath (2007).

Men maakt daarbij gebruik van de filter $\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$.

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt
3 >>> import scipy.signal
4
5 >>> vertic_filter = np.array([[ -1, 0, 1],
6                               [ -1, 0, 1],
7                               [ -1, 0, 1]])
8 >>> bamboe = np.load("bamboe.npy")

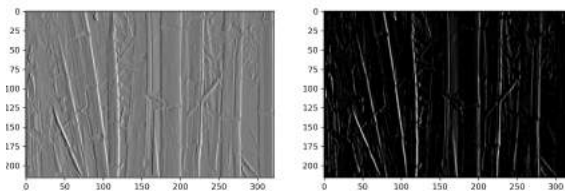
```

```
9 >>> bamboe_vertic = scipy.signal.  
    convolve2d(bamboe, vertic_filter)
```

Erna wordt op de gefilterde foto ReLU toegepast.

```
1 >>> bamboe_vertic_relu =  
2 relu(bamboe_vertic)
```

Het resultaat kan bekeken worden in Figuur 13.4.



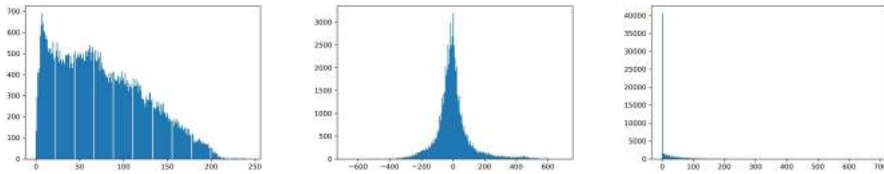
Figuur 13.4: ReLU toepassen op de gefilterde foto.

De oorspronkelijke grijswaardenafbeelding bamboe heeft elementen tussen 0 en 255. Door het uitvoeren van de convolutie werden waarden bekomen die groter konden zijn en zelfs negatief. De ReLU-functie heeft alle negatieve waarden op nul gezet en de andere behouden.

De volgende code geeft aan met welke waarden er concreet gewerkt wordt:

```
1 >>> print(np.min(bamboe), np.max(bamboe))  
2 0 244  
3 >>> print(np.min(bamboe_vertic),  
4           np.max(bamboe_vertic))  
5 -665 689  
6 >>> print(np.min(bamboe_vertic_relu),  
7           np.max(bamboe_vertic_relu))  
8 0 689
```

Na de convolutie wordt de waarde 689 geïnterpreteerd als wit en de waarde -665 als zwart. Alle grijswaarden ertussen worden verhoudingsgewijs toegekend. De ReLU-functie zal dus alle waarden donkerder dan een soort 'middengrijs' zwart maken. De verdeling van de grijswaarden wordt weergegeven in Figuur 13.5.



Figuur 13.5:

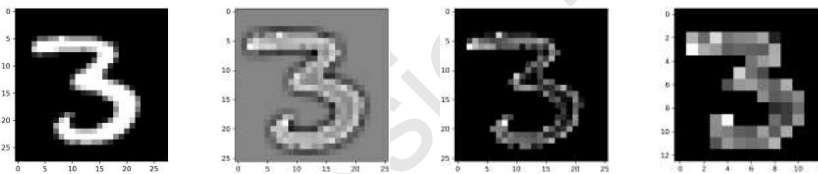
Links: Originele foto.

Midden: Gefilterde foto.

Rechts: Gefilterd, ReLU.

13.4 Effect van max pooling

Beschouw opnieuw de '3' uit de MNIST dataset, die al aan bod kwam in hoofdstuk 12. Bekijk het effect van een convolutie, gevolgd door ReLU en max pooling in Figuur 13.6.



Figuur 13.6: Convolutie met randfilter op '3' uit MNIST dataset, gevolgd door ReLU en max pooling.

Men kan de grootte van de afbeeldingen opvragen:

```

1 >>> print(drie.shape)
2 (28, 28)
3 >>> print(drie_rand.shape)
4 (26, 26)
5 >>> print(drie_rand_relu.shape)
6 (26, 26)
7 >>> print(drie_rand_relu_maxpool.shape)
8 (13, 13)

```

De max pooling-operatie maakt de afbeelding vier keer kleiner.

Samengevat

Om huidmondjes te herkennen, gebruikt KIKS een diep neuraal netwerk dat bestaat uit een convolutioneel netwerk, gevolgd door een netwerk met *dense layers*. Het model bestaat uit aaneengeschakelde lagen.

In elke laag passen de convoluties filters toe op de inputtensoren. Ze worden gevolgd door een ReLU-functie met aansluitend een *max pooling*-operatie.

ReLU zorgt ervoor dat de kenmerken die het minst tot een goede classificatie leiden, worden verzwakt. ReLU zal er bv. bij grijswaardenafbeeldingen voor zorgen dat de lichte kleuren behouden blijven en de donkere kleuren worden afgevlakt.

Max pooling zal de kenmerken die van belang zijn om tot een goede classificatie te komen, versterken en de andere kenmerken weglaten. Bovendien zorgt max pooling ervoor dat de tensoren kleiner worden, waardoor er minder rekenkracht nodig is door de computer.

Versie

IMPLEMENTATIE IN KERAS

14.1 Diepe neurale netwerken van KIKS

Een diep lerend netwerk bestaat uit meerdere lagen die aaneengeschakeld zijn. Elke laag beschikt over een aantal parameters, zijn gewichten. Figuur 14.1 stelt de opbouw van een diep lerend netwerk schematisch voor (Chollet, 2018).



Figuur 14.1: Opbouw van een diep neuraal netwerk.

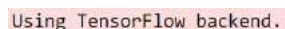
De input X (uit de trainingdata) wordt gegeven aan de eerste laag, de input layer. Via berekeningen binnen deze layer wordt een output bekomen; X is getransformeerd naar een nieuwe gedaante. Deze output gaat dan naar de volgende layer, waar een soortgelijk proces plaatsvindt. Tot slot komt de output van de voorlaatste layer aan in de laatste layer, de output layer. De output layer heeft ook een output: het systeem doet een

voorspelling \hat{y} . Deze voorspelling wordt vergeleken met het gegeven label y . Daarvoor is de kostenfunctie (*loss function*) nodig. De mate waarin de voorspellingen verschillen van de labels, bepaalt de aanpassingen die gedaan worden aan de gewichten. Daarvoor is dan weer de optimalisatiefunctie (*optimizer*) nodig. Een eerste epoch is gedaan. Het systeem onderneemt een tweede epoch: het proces begint opnieuw, maar nu met de aangepaste gewichten. Het aanpassen van de gewichten is wat men bedoelt als men zegt dat het systeem leert. De 'kennis' van het netwerk is opgeslagen in de gewichten.

Het feit dat deze netwerken eigenlijk een simpele opbouw hebben, heeft als gevolg dat de implementatie op een vrij eenvoudige manier kan gebeuren. De Python-module Keras voorziet bouwblokken om een neurale netwerk op te bouwen. Dankzij Keras wordt heel wat werk bespaard: in de achterliggende code zijn de nodige functionaliteiten vervat. Voor het rekenen met tensoren en voor andere rekenkundige bewerkingen doet Keras zelf een beroep op het platform TensorFlow.

Er moeten wel nog keuzes gemaakt worden door de architect van het netwerk en de data moeten in de juiste vorm worden aangeboden, maar het oplossen van bepaalde problemen met een deep learning netwerk ligt binnen ieders bereik, mits er voldoende rekenkracht voorhanden is.

De layers kunnen gewone layers zijn of convolutionele layers. De transformaties die gebeuren binnen een layer bestaan uit een samenstelling van convolutiebewerkingen of lineaire transformaties met niet-lineaire transformaties. Er worden dus bewerkingen uitgevoerd die niet meer vergen dan vermenigvuldigen en optellen, weliswaar gevolgd door een niet-lineaire functie zoals ReLU, max pooling, sigmoïde of Heaviside.



```
Using TensorFlow backend.
```

Figuur 14.2: Keras refereert aan TensorFlow.

In de KIKS-notebooks worden verschillende types van problemen behandeld: regressie en binaire of multiklas-senclassificatie. Sommige worden aangepakt met het Perceptron, een neurale netwerk met slechts twee lagen. Andere worden aangepakt met een diep neurale netwerk, al dan niet gecombineerd met een convolutio-neel netwerk.

Meer lagen of meer gewichten hebben als voordeel dat er meer patronen ontdekt kunnen worden, wat de prestatie van het netwerk verhoogt. Bij een classificatie-probleem met n klassen vermijdt men het best hidden layers met veel minder dan n gewichten. Dat zou er immers voor zorgen dat het systeem niet alle relevante informatie kan leren.

Als men voor problemen rond bv. computer vision ge-bruikmaakt van een convolutioneel netwerk gevolgd door een feedforward netwerk, zal men betere resulta-ten bekomen dan met enkel een feedforward netwerk. Met minder epochs kan dan een beter resultaat beko-men worden. Let wel: elke epoch kost meer rekenkracht en zal langer duren.

Vóór de training zal de netwerkkarchitect de data op-splitsen in trainingdata, valideringsdata en testdata (zie paragrafen 11.13 en 11.15).

De netwerkkarchitect moet ook nog een kostenfunc-tie en een optimalisatiefunctie vastleggen, en enkele hyperparameters kiezen: de *learning rate* en eventueel een *threshold* (zie hoofdstuk 11).

14.2 Netwerkkarchitectuur van de KIKS-notebooks

De keuzes die de netwerkkarchitect maakt, worden be-paald door het probleem, maar grotendeels ook door zijn ervaring. In de tabellen 14.1, 14.2 en 14.3 worden enkele tips gegeven voor keuzes die men moet maken.

	Tips regressie
Dataset	gelabelde $X'n$ KIKS-notebooks: datapunten met één kenmerk KIKS-voorbeelden: gletsjers, zeeniveau, CO ₂ , Keeling
y, \hat{y}	continue waarden
Voorspellingen naar de toekomst	testset bestaat uit de 'laatste' waarden KIKS-voorbeeld: stijging zeeniveau in Oostende
Probleem zonder tijdsaspect	testset kan hier willekeurig uit de dataset worden gekozen KIKS-voorbeeld: verband aantal stomata en hoogte bomen
Standaardiseren	steeds de data standaardiseren verschillende grootte-orde
Loss	mean squared error (MSE)
Optimizer	gradient descent (GD) of stochastic gradient descent (SGD)
Tussenslagen	activatiefunctie: ReLU
Output layer	geen activatiefunctie, m.a.w. activatiefunctie is identiteit
Metrics	meerdere mogelijkheden afhankelijk van het probleem bv. mean absolute error (MAE)

Tabel 14.1: Tips bij regressieprobleem.

In de KIKS-notebooks worden neurale netwerken opgebouwd. In de eerste notebooks over machinaal leren gebeurt dat met algoritmen op basis van wiskundige principes die men zelf programmeert. Deze of vergelijkbare algoritmes zijn reeds voorzien in Keras. In de notebooks van *deep learning* gebeurt het opbouwen van de neurale netwerken met de functionaliteiten van de Python-module Keras.

Een diep neurale netwerk bevat minstens drie lagen, dus minstens één verborgen laag. Vaak zijn het er meer. Om zo'n diep neurale netwerk te bouwen, moet men eerst de architectuur vastleggen. Bij KIKS gebruiken we een model dat bestaat uit aaneengeschaalde lagen, het zogenaamde *Sequential model* (Keras, 2019).

Hoe wordt bv. een *feedforward* netwerk met één hidden layer geïmplementeerd? De lagen zijn *dense layers*, anders gezegd *fully connected layers*, d.w.z. dat de neuronen in een bepaalde laag met alle neuronen in de vorige laag zijn verbonden. Bij elke laag moet vermeld worden hoeveel neuronen die laag moet bevatten en welke activatiefunctie er toegepast moet worden.

Tips binaire classificatie	
Dataset	gelabelde X^n KIKS-notebooks met grafieken: datapunten met één of twee kenmerken x_1 en x_2 KIKS-voorbeelden: stomatadetectie, twee soorten irissen, bezonde en beschaduwde bladeren, twee parabolen KIKS-voorbeelden: 1D, 2D
y, \hat{y}	$y \in \{0,1\}, \hat{y} \in \{0,1\}$
Twee klassen	een klasse krijgt label 1 en de andere krijgt label 0
Testset	testset kan willekeurig uit de dataset worden gekozen
Standaardiseren	de data standaardiseren of normaliseren verschillende grootte-orde
Loss	binary cross-entropy (BCE)
Optimizer	gradient descent (GD) of stochastic gradient descent (SGD)
Tussenlagen	activatiefunctie: ReLU
Output layer	activatiefunctie: sigmoïde deze geeft hoe zeker het model is dat gegeven tot klasse 1 behoort om de klasse effectief te voorspellen moet er een threshold worden gekozen, bv. vanaf 75 % zekerheid, voorspel klasse 1
Metrics	accuracy (accuracy)

Tabel 14.2: Tips bij binair classificatieprobleem.

Het model moet weten welke vorm van input het kan verwachten, m.a.w. wat de dimensie is van de inputpunten. Daarom wordt dit aan de eerste laag van het Sequential model meegegeven. Dat moet enkel voor de eerste laag, want de volgende lagen verkrijgen dat automatisch, aangezien die dimensie bepaald wordt door de wiskundige bewerkingen die er gebeuren.

Men moet een *loss function*, een *optimizer* en een *learning rate* kiezen. De loss function zal geminimaliseerd worden m.b.v. de optimizer. Men moet ook een methode kiezen om de performantie van het model na te gaan: een *metrics*. Met de metrics wordt een cijfer bepaald dat de prestatie representeert. Als de learning rate niet expliciet vermeld staat in de netwerkopbouw van het Python-script, dan gebruikt het netwerk een reeds vooropgestelde learning rate, bv. 0,01 bij *stochastic gradient descent* (SGD). Die vooropgestelde learning rate hangt af van de optimizer. De keuzes die men maakt, zijn bepaald door het soort probleem en hangen af van de ervaring van de netwerkarchitect.

Ook voor het trainingsaspect moet de netwerkarchi-

Zie paragraaf 11.14 voor stochastische gradient descent en het gebruik van batches.

Tips multiklassenclassificatie	
Dataset	gelabelde $X'n$ KIKS-notebooks met grafieken: datapunten met één of twee kenmerken x_1 en x_2 KIKS-voorbeelden: MNIST, drie soorten irissen
y, \hat{y} n klassen	$y, \hat{y} \in \{0, 1, 2\}$ of $y, \hat{y} \in \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ als de klassen een kwalitatieve waarde hebben, dan wordt die waarde omgezet naar een kwantitatieve waarde, meer bepaald een discrete waarde bv. bij MNIST: 0 t.e.m. 9, soort iris wordt 0, 1 en 2 deze discrete waarden worden nog eens omgezet via <i>one hot encoding</i> : bv. eerste klasse is bv. 100, de tweede klasse 010 en de derde klasse 001
Testset	testset kan willekeurig uit de dataset worden gekozen
Standaardiseren	de data standaardiseren of normaliseren verschillende grootte-orde
Loss	categorical cross-entropy
Optimizer	gradient descent (GD) of stochastic gradient descent (SGD)
Tussenschichten	activatiefunctie: ReLU
Output layer	activatiefunctie: softmax deze geeft hoe zeker het model is dat gegeven tot een van de klassen behoort, dus geeft voor elke klasse de zekerheid
Metrics	accuracy (accuracy)

Tabel 14.3: Tips bij problemen met meer dan twee klassen.

tect nog enkele zaken vastleggen: bv. het aantal epochs dat het systeem moet ondernemen en de grootte van de batches. Als het Python-script de grootte van een batch niet expliciet vermeldt, dan gebruikt het netwerk een batch van 32. Vóór er dan effectief met trainen kan begonnen worden, moeten de gegevens eventueel nog omgevormd worden naar het juiste formaat: tensoren met de juiste dimensie, kwantitatieve labels i.p.v. kwalitatieve, *one hot encoding* i.p.v. 0, 1, 2, 3 ...

Het trainen kan dan gebeuren. Hiervoor wordt de methode `fit()` gebruikt. De kenmerken en de labels worden op elkaar afgestemd. Tijdens de training worden na elke epoch de *loss* en *accuracy* op de valideringsset getoond. Zolang de loss daalt en de accuracy stijgt, verloopt de training naar wens.

De methode `fit()` geeft een object terug van de klasse `History`. Dit object heeft een attribuut `history` waarin de waarden van de training *loss*-waarden en de *metrics*-waarden van de opeenvolgende epochs zijn bewaard en, indien van toepassing, ook de *loss* en de

Bij *one hot encoding* wordt bv. een natuurlijk getal voorgesteld door een 'binaire' vector: 0 wordt $(0, 0, \dots, 0)$, 1 wordt $(0, 1, 0, \dots, 0)$, 2 wordt $(0, 0, 1, 0, \dots, 0)$ enz. In Keras kan men deze omzetting doen met `to_categorical()`.

metrics van de valideringsset.

14.3 Implementatie in Keras

Deze paragraaf bevat voorbeelden van neurale netwerken die geïmplementeerd werden om eenvoudige problemen aan te pakken.

14.3.1 MNIST, neuraal netwerk met één hidden layer

Met de MNIST dataset wordt een neuraal netwerk getraind dat handgeschreven cijfers herkent. Het gaat dus om een classificatieprobleem met tien klassen (zie tabel 14.3).

De MNIST dataset bestaat uit 60 000 trainingafbeeldingen en 10 000 testafbeeldingen van 28 x 28 pixels in grijswaarden.

Eerst worden de nodige modules geïmporteerd uit de module Keras: de module MNIST om de data te bekomen, de modules models en layers die samen de mogelijkheid bieden om een deep learning netwerk te creëren en dan nog een functionaliteit om de data naar het gewenste formaat om te vormen. Pyplot wordt ook geïmporteerd omdat er ook grafieken zullen worden getekend om de evolutie in de prestaties van het netwerk tijdens de training te visualiseren.

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt
3
4 >>> from keras import models
5 >>> from keras import layers
6 >>> from keras.utils import to_categorical
7
8 >>> from keras.datasets import mnist

```

MNIST wordt ook behandeld in paragrafen 10.8, 12.4.2 en 13.4.



Zie de notebook 'MNIST'.

Na het importeren van de modules laadt men de dataset in. Deze dataset bestaat uit twee koppels van telkens twee koppels met daarin enerzijds de trainingset met de corresponderende labels en anderzijds de testset met zijn labels.

Men houdt rekening met de structuur van de MNIST dataset door bij het inladen reeds vier variabelen te gebruiken voor de vier onderdelen van de dataset. Men doet dat als volgt:

```
1 >>> (train_images, train_labels),
2     (test_images, test_labels)
3     = mnist.load_data()
```

Men verdeelt de trainingset in een deel om effectief te trainen, de trainingset, en een deel om de prestatie van het netwerk tijdens de training te monitoren, de valideringsset. De trainingset en de valideringsset mogen geen gemeenschappelijke elementen bevatten.

```
1 >>> train_images = train_images[5000:, :, :]
2 >>> train_labels = train_labels[5000:]
3 >>> validation_images = train_images
4 >>> validation_labels = train_labels
   [:5000]
```

Voor de architectuur van het netwerk kiest men een Sequential model met dense layers waarvan één hidden layer. Het aantal neuronen en de activatiefunctie van deze hidden layer kiest men en ook het formaat van de input. Het aantal neuronen in de output layer komt overeen met het aantal klassen, nl. 10, en de activatiefunctie wordt bepaald door de aard van het probleem, een classificatieprobleem met 10 klassen gebruikt softmax.

```
1 >>> network = models.Sequential()
```

De opbouw van de MNIST dataset werd besproken in paragraaf 10.8.1.

De dimensie van `train_images` is (60000, 28, 28) en de dimensie van `train_labels` is (60000,).

De trainingset bevat 55 000 afbeeldingen en de valideringsset bevat er 5 000. De testset heeft er 10 000.

Classificatieprobleem met 10 klassen, zie Tabel 14.3, activatiefunctie is softmax.

```

2 >>> network.add(layers.Dense(512,
3     activation="relu",
4     input_dim=28*28))
>>> network.add(layers.Dense(10,
5     activation="softmax"))

```

Er moet ook een loss function, een optimizer en een metrics gekozen worden:

```

1 >>> network.compile(optimizer="sgd",
2     loss="categorical_crossentropy",
3     metrics=["accuracy"])

```

De hidden layer heeft 512 neuronen en ReLU als activatiefunctie. Voor het formaat van de input worden 1D-tensoren gekozen.

Voordat men begint met trainen, controleert men of de data het gewenste formaat hebben. Hier is dat niet zo, dus zal men dat formaat aanpassen. De data worden ook genormaliseerd.

Meer uitleg over normaliseren vindt u in paragraaf 12.6.

```

1 >>> train_images = train_images.reshape
2     ((55000, 28*28))
3 >>> train_images = train_images.astype("
4     float32") / 255
5 >>> validation_images = validation_images.
6     reshape((5000, 28*28))
7 >>> validation_images = validation_images.
8     astype("float32") / 255
9 >>> test_images = test_images.reshape
10    ((10000, 28*28))
11 >>> test_images = test_images.astype("
12    float32") / 255
13 >>> train_labels = to_categorical(
14    train_labels)
15 >>> validation_labels = to_categorical(
16    validation_labels)
17 >>> test_labels = to_categorical(
18    test_labels)

```

Men kan nu trainen. Daarvoor kiest men het aantal epochs en de learning rate. Hier wordt gekozen voor

de standaard learning rate die Keras gebruikt voor de optimizer stochastic gradient descent (SGD), nl. 0,01; men moet die dan niet expliciet vermelden. Men bepaalt ook de grootte van de batches.

Voor stochastic gradient descent en het gebruik van batches: zie paragraaf 11.14.

```
1 >>> history = network.fit(train_images,
    train_labels, validation_data=(
    validation_images, validation_labels),
    epochs=5, batch_size=100)
```

Met de methode `fit()` worden de kenmerken van de trainingdata en hun labels op elkaar afgestemd. Tijdens de training verschijnen de loss en de metrics na elke epoch, voor de training- en de valideringsset (zie Figuur 14.3).

```
Train on 55000 samples, validate on 5000 samples
Epoch 1/5
55000/55000 [=====] - 7s 121us/step - loss: 1.0236 - acc: 0.7726 - val_loss: 0.5548 - val_acc: 0.8736
Epoch 2/5
55000/55000 [-----] - 6s 113us/step - loss: 0.4865 - acc: 0.8792 - val_loss: 0.4087 - val_acc: 0.8972
Epoch 3/5
55000/55000 [-----] - 6s 113us/step - loss: 0.3985 - acc: 0.8940 - val_loss: 0.3548 - val_acc: 0.9070
Epoch 4/5
55000/55000 [-----] - 6s 113us/step - loss: 0.3573 - acc: 0.9024 - val_loss: 0.3236 - val_acc: 0.9140
Epoch 5/5
55000/55000 [-----] - 6s 113us/step - loss: 0.3312 - acc: 0.9087 - val_loss: 0.3028 - val_acc: 0.9176
```

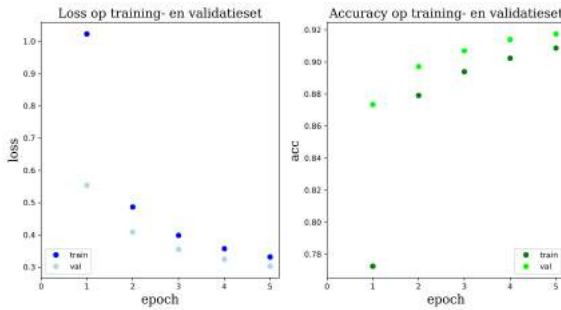
Figuur 14.3: Training MNIST 5 epochs.

De loss daalt en de accuracy stijgt, zoals gewenst.

Via de methode `fit()` worden de loss-waarden en de metrics-waarden van de opeenvolgende epochs bewaard. Men kan er een grafische voorstelling van maken (zie Figuur 14.4).

```
1 >>> epochs = range(1, len(loss) + 1)
2 >>> loss = history.history["loss"]
3 >>> acc = history.history["accuracy"]
4 >>> val_loss = history.history["val_loss"]
5 >>> val_acc = history.history["
    val_accuracy"]
```

Na de training bekijkt men de loss en de metrics op de testset om te evalueren hoe het model presteert op nieuwe data (zie Figuur 14.5).



Figuur 14.4: MNIST 5 epochs. Accuracy stijgt en loss daalt bij de trainingset en bij de validatieset. Het model underfits.

```

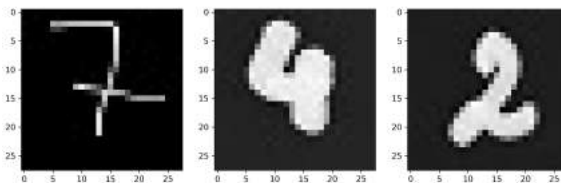
1 >>> test_loss, test_acc = network.evaluate
    (test_images, test_labels)
2 >>> print("test_loss: ", test_loss)
3 >>> print("test_acc: ", test_acc)
    
```

```

10000/10000 [=====] - 1s 51us/step
test_loss: 0.303586756837368
test_acc: 0.9185000061988831
    
```

Figuur 14.5: MNIST 5 epochs. Testresultaat.

Het model testen op eigen handgeschreven cijfers (zie Figuur 14.6) levert het resultaat uit Figuur 14.7. Het model deelt de afbeelding in bij een klasse met een bepaalde zekerheid.



Figuur 14.6: Eigen cijfers.

Bv. de '7' wordt met een zekerheid van ongeveer 25 % ingedeeld bij de klasse 1 en met een zekerheid van ongeveer 12 % bij de klasse 4. Het komt erop neer dat het model deze '7' niet herkent. De '4' herkent het model ook niet. Maar de '2' wel: die wordt met een

```

resultaat 7: [[0.01635502 0.25485316 0.09518994 0.0992751 0.11617411 0.07869939
0.1553419 0.07982513 0.04736559 0.05692071]]
resultaat 4: [[1.07561325e-04 1.39814671e-04 6.37601968e-03 2.61521161e-01
1.13332085e-01 5.64364195e-02 1.75758672e-03 6.78145587e-02
6.48370758e-02 4.27677691e-01]]
resultaat 2: [[1.1084233e-04 7.5431203e-04 9.7368211e-01 1.7497968e-02 1.9303905e-06
2.2199535e-04 1.3550465e-03 7.5645446e-05 6.1974693e-03 1.0265063e-04]]

```

Figuur 14.7: Prestatie van het model op eigen cijfers.

zekerheid van 97 % ingedeeld bij de correcte klasse 2.

Uit Figuur 14.4 blijkt dat het netwerk underfits. Het kan met meerdere epochs nog verbeteren.

Bekijk de cijfers voor 70 epochs. Op Figuur 14.8 ziet men dat de loss na de eerste epoch verschillend is van die van daarnet. Dat komt omdat de weights willekeurig zijn bij het begin van de training. Op Figuur 14.9 ziet men dat het netwerk overfits na ongeveer 40 epochs. Bij ongeveer 40 epochs levert de training een model dat optimaal presteert.

```

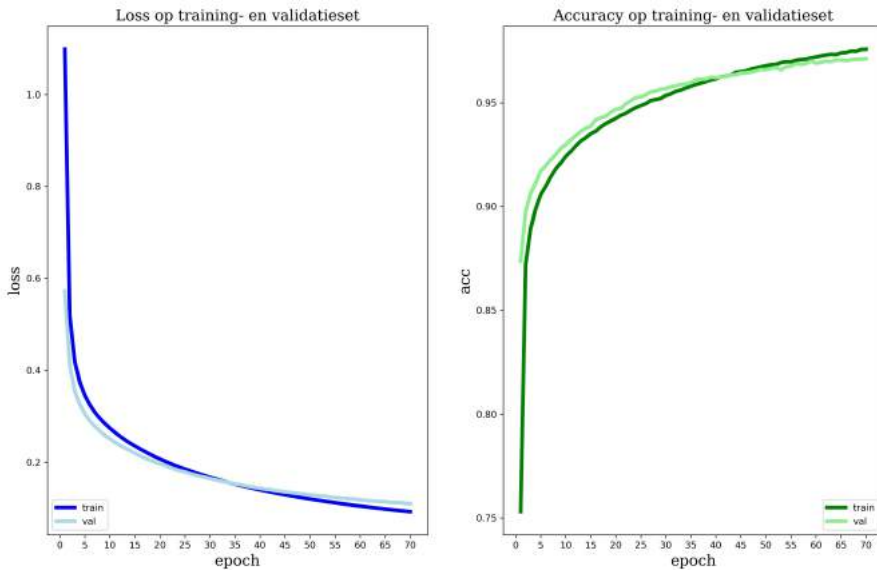
Train on 50000 samples, validate on 10000 samples
Epoch 1/70
50000/50000 [-----] - 6s 123us/step - loss: 1.0985 - acc: 0.7530 - val_loss: 0.5724 - val_acc: 0.8736
Epoch 2/70
50000/50000 [-----] - 6s 120us/step - loss: 0.5196 - acc: 0.8719 - val_loss: 0.4108 - val_acc: 0.8974
Epoch 3/70
50000/50000 [=====] - 6s 115us/step - loss: 0.4193 - acc: 0.8894 - val_loss: 0.3544 - val_acc: 0.9064
Epoch 4/70
50000/50000 [-----] - 6s 117us/step - loss: 0.3735 - acc: 0.8990 - val_loss: 0.3257 - val_acc: 0.9113
Epoch 5/70
50000/50000 [-----] - 6s 118us/step - loss: 0.3451 - acc: 0.9058 - val_loss: 0.3047 - val_acc: 0.9171
Epoch 6/70
50000/50000 [-----] - 6s 116us/step - loss: 0.3246 - acc: 0.9096 - val_loss: 0.2898 - val_acc: 0.9196

```

Figuur 14.8: Training MNIST 70 epochs.

14.3.2 Punten scheiden gelegen op twee parabolen, netwerk met één hidden layer

In de notebook 'Binaire classificatie met neuraal netwerk met een verborgen laag' worden punten gelegen op twee parabolen, van elkaar gescheiden door een neuraal netwerk, opgebouwd met Keras. De punten zijn niet lineair scheidbaar (zie Figuur 14.10).



Figuur 14.9: MNIST 70 epochs. Het model overfits.

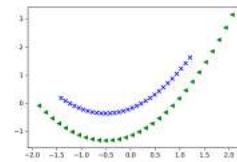
Het neurale netwerk wordt opgebouwd met één hidden layer. Bij te weinig of te veel neuronen in de hidden layer zijn de prestaties ondermaats.

Figuur 14.12 toont de prestatie na een training van 40 epochs met een learning rate van 0,08 en 20 neuronen in de hidden layer. Het netwerk overfits na 25 epochs. Na 25 epochs wordt het netwerk wel nog beter op de trainingdata, maar de prestatie op de valideringsset neemt af. De prestatie van het geïmplementeerde netwerk lijkt optimaal na 25 epochs. De implementatie van dit netwerk:

```

1 # import
2 >>> import pandas as pd
3 >>> import numpy as np
4 >>> import matplotlib.pyplot as plt
5
6 >>> from keras import models

```



Figuur 14.10: Puntenwolk van de gestandaardiseerde data.

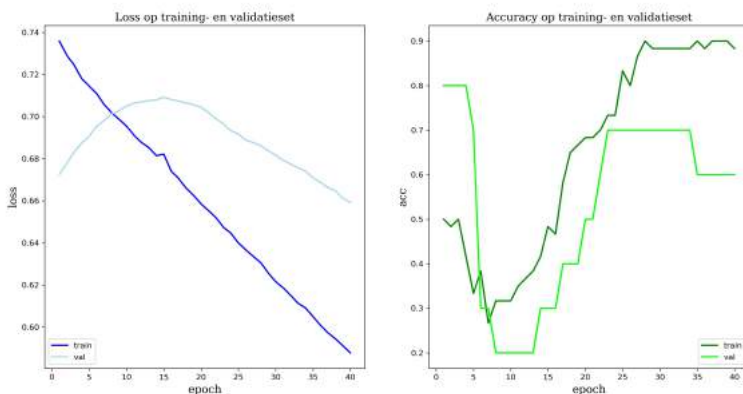
Binaire classificatie, zie Tabel 14.2.



Extra uitleg vindt u in de notebooks over de parabolen.

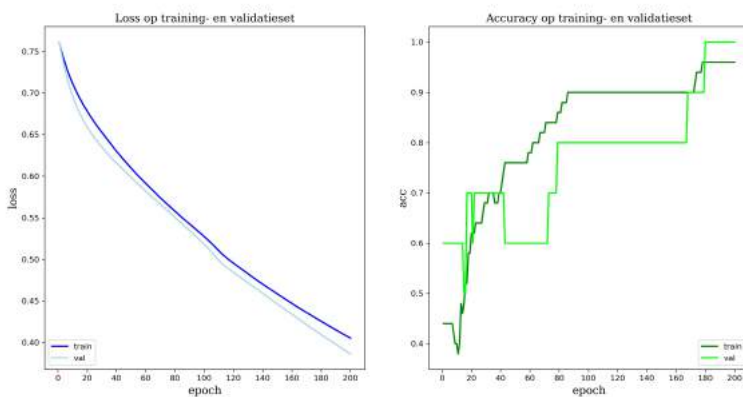
```
7 >>> from keras import layers
8 >>> from keras import optimizers
9
10 >>> from sklearn.utils import shuffle
11
12 # data
13 >>> punten = pd.read_csv("parabolen.dat")
14 >>> punten = np.array(punten)
15 >>> x1 = punten["x"]
16 >>> x2 = punten["y"]
17 >>> y = punten["kleur"]
18
19 # data voorbereiden
20 # labels zijn kwalitatief, numeriek maken
21 >>> x1 = np.array(x1)
22 >>> x2 = np.array(x2)
23 >>> y[y == "blauw"] = 0
24 >>> y[y == "groen"] = 1
25
26 # onthouden voor testset
27 >>> mean1 = np.mean(x1)
28 >>> std1 = np.std(x1)
29 >>> mean2 = np.mean(x2)
30 >>> std2 = np.std(x2)
31
32 # data standaardiseren
33 >>> x1 = (x1 - np.mean(x1)) / np.std(x1)
34 >>> x2 = (x2 - np.mean(x2)) / np.std(x2)
35 >>> X = np.stack((x1, x2), axis=1)
36
37 # data onderling wisselen
38 # X en y blijven gekoppeld
39 # valideringsset wordt dan willekeurig
    gekozen
40 >>> X, y = shuffle(X,y)
41 >>> x1 = X[:, 0]
42 >>> x2 = X[:, 1]
43
```

```
44 # dataset splitsen in training- en
    valideringsset
45 >>> x1_val = np.append(x1[0:5],
46                       x1[30:35])
47 >>> x2_val = np.append(x2[0:5],
48                       x2[30:35])
49 >>> X_val = np.stack((x1_val, x2_val),
    axis=1)
50 >>> x1_train = np.append(x1[5:30],
51                          x1[35:])
52 >>> x2_train = np.append(x2[5:30],
53                          x2[35:])
54 >>> X_train = np.stack((x1_train, x2_train
    ), axis=1)
55 >>> y_val = np.append(y[0:5], y[30:35])
56 >>> y_train = np.append(y[5:30], y[35:])
57
58 # architectuur
59 >>> network = models.Sequential()
60 >>> network.add(layers.Dense(20,
    activation="relu", input_dim=2))
61 >>> network.add(layers.Dense(1,
62                          activation="sigmoid"))
63 >>> sgd = optimizers.SGD(lr=0.08)
64 >>> network.compile(optimizer=sgd,
65                    loss="binary_crossentropy",
66                    metrics=["accuracy"])
67
68 # training
69 >>> history = network.fit(X_train, y_train
    , validation_data=(X_val,y_val),
    epochs=40, batch_size=len(y_train))
70
71 >>> loss = history.history["loss"]
72 >>> epochs = range(1, len(loss) + 1)
73 >>> acc = history.history["accuracy"]
74 >>> val_loss = history.history["val_loss"]
75 >>> val_acc = history.history["val_acc"]
```

Figuur 14.12: Loss en accuracy tijdens de training met één hidden layer, 40 epochs.

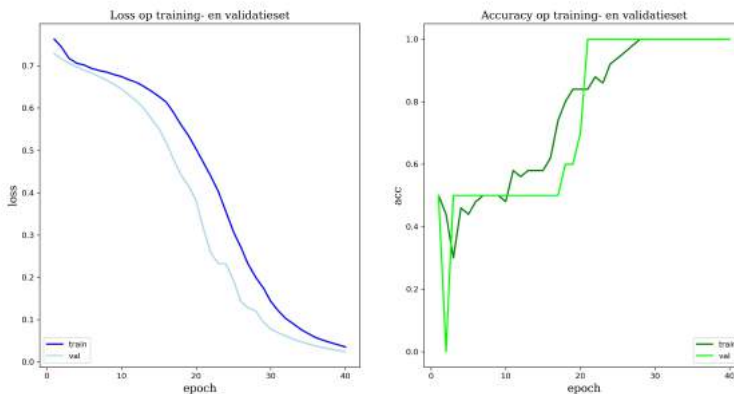
Hetzelfde netwerk werd opnieuw getraind met 200 epochs. De prestaties zijn merkkelijk beter.



Figuur 14.13: Loss en accuracy tijdens de training met één hidden layer, 200 epochs.

14.3.3 Punten scheiden gelegen op twee parabolen, netwerk met twee hidden layers

Hetzelfde probleem wordt aangepakt in de notebook 'Binaire classificatie met neurale netwerk met twee verborgen lagen'. Figuur 14.14 toont de prestatie na een training van 40 epochs met een learning rate van 0,1, 8 neuronen in de eerste hidden layer en 4 neuronen in de tweede hidden layer. Dit netwerk presteert in minder epochs beter dan dat met één layer, ook de loss is kleiner.



Figuur 14.14: Loss en accuracy tijdens de training met twee hidden layers, 40 epochs.

De implementatie van dit netwerk is bijna identiek aan het vorige. Enkel de architectuur verschilt en de valideringsset wordt manueel gekozen:

```

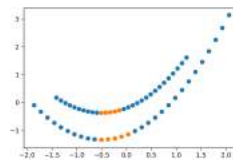
1 # architectuur
2 >>> network = models.Sequential()
3 >>> network.add(layers.Dense(8,
4     activation="relu", input_dim=2))
5 >>> network.add(layers.Dense(4,
6     activation="relu"))
7 >>> network.add(layers.Dense(1,
8     activation="sigmoid"))
9 >>> sgd = optimizers.SGD(lr=0.1)

```

```

10 >>> network.compile(optimizer=sgd,
11     loss="binary_crossentropy",
12     metrics=["accuracy"])

```



14.3.4 Convolutioneel neuraal netwerk

Men kan het probleem uit paragraaf 14.3.1 (MNIST) ook aanpakken met een convolutioneel neuraal netwerk.

De implementatie gebeurt dan als volgt: het diep neuraal netwerk is nog steeds een *Sequential model*, een model dat bestaat uit aaneengeschakelde lagen. Hier zijn dat echter eerst enkele convolutionele lagen die, afwisselend met max pooling-lagen, samen een convolutioneel netwerk vormen, gevolgd door een netwerk van dense layers.

Dezelfde modules dienen te worden geïmporteerd als in paragraaf 14.3.1.

Figuur 14.15: Zelf gekozen valideringsset.

```

1 >>> import numpy as np
2 >>> import matplotlib.pyplot as plt
3
4 >>> from keras import models
5 >>> from keras import layers
6 >>> from keras.utils import to_categorical
7
8 >>> from keras.datasets import mnist

```

Ook de data zijn dezelfde:

```

1 >>> (train_images, train_labels),
2     (test_images, test_labels)
3     = mnist.load_data()
4 >>> train_images
5     = train_images[5000:, :, :]
6 >>> train_labels = train_labels[5000:]
7 >>> validation_images
8     = train_images[:5000, :, :]
9 >>> validation_labels
10    = train_labels[:5000]

```

De convolutionele lagen hebben filters nodig, de netwerkkarchitect bepaalt daarvan het aantal, en een activatiefunctie.

Meestal kiest men de ReLU-functie, hier ook. De convolutionele lagen worden afgewisseld met max pooling-lagen.



```

1 >>> network = models.Sequential()
2 >>> network.add(layers.Conv2D(32, (3,3),
   activation="relu",
   input_shape=(28,28,1)))
3
4 >>> network.add(layers.Maxpooling2D(2,2))
5 >>> network.add(layers.Conv2D(64, (3,3),
   activation="relu"))
6 >>> network.add(layers.Maxpooling2D(2,2))
7 >>> network.add(layers.Conv2D(64, (3,3),
   activation="relu"))

```

Voer de notebook 'Classificatie van de MNIST dataset met een convolutioneel neural netwerk' uit.

De convoluties passen filters toe op de inputtensoren. De tensoren worden erdoor getransformeerd in een nieuwe representatie van de gegevens. De convoluties worden gevolgd door een ReLU-functie met aansluitend een max pooling-bewerking. De ReLU zal er bij grijswaardenafbeeldingen bv. voor zorgen dat de lichte kleuren behouden blijven en de donkere kleuren worden afgevlakt. Max pooling zal de kenmerken die van belang zijn om tot een goede classificatie te komen, versterken en de andere kenmerken weglaten. Max pooling zorgt ervoor dat de tensoren kleiner worden, waardoor er minder rekenkracht nodig is.

Na het doorlopen van het convolutionele netwerk wordt de uitvoer hiervan aan het netwerk met dense layers gegeven. Om dat mogelijk te maken moet eerst de operatie Flatten() plaatsvinden:

```

1 >>> network.add(layers.Flatten())

```

De uitvoer van deze layer belandt in het feedforward netwerk:

```

1 >>> network.add(layers.Dense(64,

```

```
activation="relu"))
```

Tot slot wordt de output layer bereikt, die dan een voorspelling zal geven betreffende het gegeven.

```
1 >>> network.add(layers.Dense(10,
    activation="softmax"))
```

Er moet ook een loss-functie, een optimizer en een metrics gekozen worden:

```
1 >>> network.compile(optimizer="sgd",
2     loss="categorical_crossentropy",
3     metrics=["accuracy"])
```

Na het voorbereiden van de data, gebeurt de training opnieuw met de methode `fit()`:

```
1 # training
2 >>> history = network.fit(train_images,
    train_labels, epochs=5, batch_size=64,
    validation_data=(validation_images,
    validation_labels))
```

Het trainen vergt per epoch meer tijd en energie dan bij het gewone neurale netwerk, maar er zijn veel minder parameters nodig. Men bekomt met 5 epochs een model dat beter presteert. De prestaties worden samengevat in de Figuren 14.16, 14.17, 14.18 en 14.19. De accuracy is hoog na amper 5 epochs. Als men de threshold op 80 % legt, herkent het netwerk de '2' en ook de '4' van de zelfgeschreven cijfers.

Het model van paragraaf 14.3.1 heeft 512 neuronen in de verborgen laag en 40 epochs. Dit model heeft slechts 224 neuronen in de verborgen lagen en 5 epochs.

```
Train on 55000 samples, validate on 5000 samples
Epoch 1/5
55000/55000 [-----] - 998s 18ms/step - loss: 0.8792 - acc: 0.7323 - val_loss: 0.3526 - val_acc: 0.8814
Epoch 2/5
55000/55000 [-----] - 981s 18ms/step - loss: 0.2234 - acc: 0.9314 - val_loss: 0.1978 - val_acc: 0.9412
Epoch 3/5
55000/55000 [-----] - 953s 17ms/step - loss: 0.1505 - acc: 0.9544 - val_loss: 0.1173 - val_acc: 0.9664
Epoch 4/5
55000/55000 [-----] - 991s 18ms/step - loss: 0.1163 - acc: 0.9647 - val_loss: 0.0984 - val_acc: 0.9782
Epoch 5/5
55000/55000 [-----] - 976s 18ms/step - loss: 0.0967 - acc: 0.9697 - val_loss: 0.1011 - val_acc: 0.9698
```

Figuur 14.16: MNIST met een convnet en 5 epochs. Training.

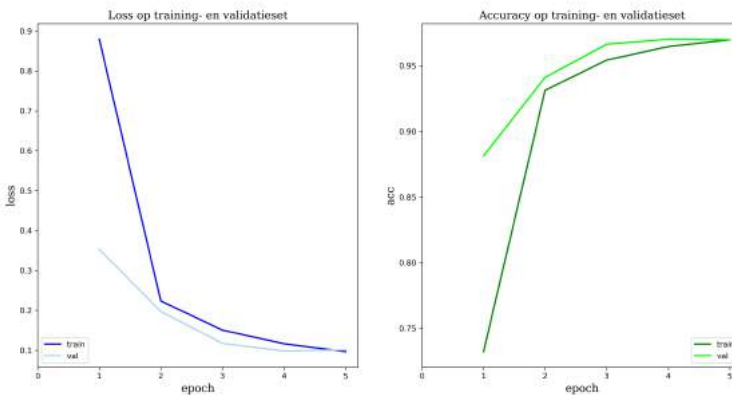
```
10000/10000 [=====] - 4s 365us/step
test_acc: 0.9689
```

Figuur 14.17: MNIST met een convnet en 5 epochs. Resultaat op de testset.

```
print(network.predict(zeven))
print(network.predict(twee))
print(network.predict(vier))
```

```
[[0.00086329 0.16578692 0.24061629 0.32421672 0.14549349 0.00948473
  0.02273904 0.02972126 0.05286934 0.00820899]]
[[3.9845669e-05 4.1820556e-03 9.3789321e-01 5.2766204e-02 2.2424701e-06
  3.1206371e-05 8.6787222e-06 4.3760633e-06 5.0704125e-03 1.8397990e-06]]
[[5.9753729e-06 1.3167849e-03 1.3054689e-05 7.0677618e-03 8.0896205e-01
  6.8870839e-04 6.6303059e-05 2.1713155e-05 1.6658823e-01 1.5269379e-02]]
```

Figuur 14.18: Prestatie van het MNIST-model met een convnet en 5 epochs op eigen cijfers.



Figuur 14.19: Loss en accuracy MNIST met een convnet en 5 epochs.

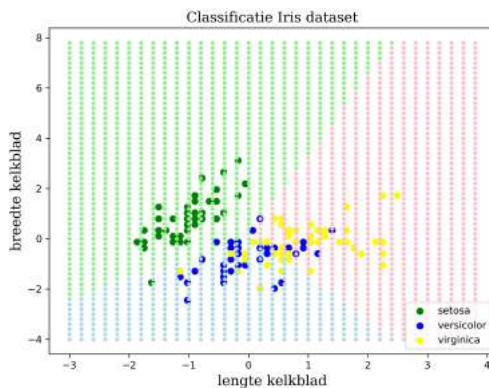
14.4 Neuraal netwerk om drie soorten irissen te scheiden

In de notebook 'Classificatie van de Iris dataset met een hidden layer' wordt er een neuraal netwerk met één verborgen laag getraind om de Iris dataset te verdelen over drie klassen.

Op basis van de lengte en breedte van een kelkblad van een iris, deelt het model een iris in bij een van de drie klassen. Dit wordt grafisch voorgesteld in Figuur 14.20. Op de grafiek staan gestandaardiseerde gegevens. De grens tussen de klassen is duidelijk te zien. Zo'n grens noemt men een *decision boundary*.



Bekijk de notebook 'Classificatie van de Iris dataset met één hidden layer'.



Figuur 14.20: Classificatie irissen, 50 epochs.

14.5 Convolutioneel neuraal netwerk van KIKS voor het tellen van stomata

14.5.1 Het netwerk

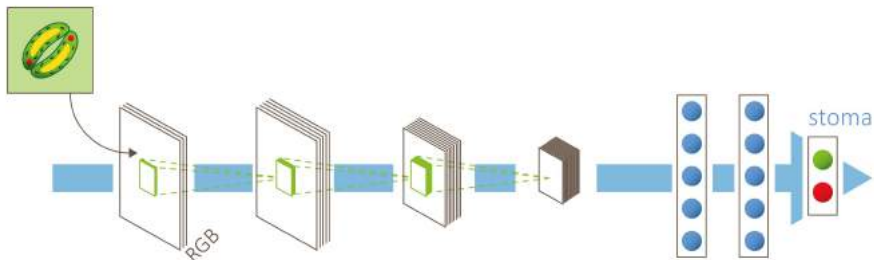
Om de stomata te tellen, gebruikt men een convolutioneel netwerk voor beeldherkenning. Hoe wordt het diep neuraal netwerk dat daarvoor nodig is, geïmplementeerd?

Men doet een beroep op een diep neurale netwerk dat bestaat uit een convolutioneel neurale netwerk gevolgd door een feedforward netwerk.

Als uitvoer wordt gegeven hoe zeker het systeem is dat er op de afbeelding van 120 x 120 pixels een stoma staat. Het probleem wordt dus behandeld als een multiklassenprobleem met twee klassen, zoals te zien in Figuur 14.21. De convolutionele lagen worden afgewisseld met *max pooling*-lagen en er wordt *dropout* toegepast.



Bekijk de notebook 'De fundamentele van een diep neurale netwerk voor beeldherkenning'.



Figuur 14.21: KIKS neurale netwerk.

De gelabelde trainingafbeeldingen worden aangeboden als een kleurenfoto in JPEG-formaat van 120 x 120 pixels.

De data worden eerst voorbereid: de invoergegevens worden genormaliseerd, op de labels wordt met `to_categorical one hot encoding` toegepast.

De opbouw van het model:

```

1 # architecture
2 >>> network = models.Sequential()
3
4 # convnet
5 >>> network.add(layers.Conv2D(8, (3,3),
6     activation="relu",
7     input_shape=(120,120,3),
8     padding="valid"))

```

`padding="valid"` komt overeen met `mode="valid"` uit hoofdstuk 12.


```
9 >>> network.add(layers.Maxpooling2D(2,2))
10 >>> network.add(Dropout(0.1))
11
12 >>> network.add(layers.Conv2D(16, (3,3),
13     activation="relu", padding="valid"))
14 >>> network.add(layers.Maxpooling2D(2,2))
15 >>> network.add(Dropout(0.25))
16
17 >>> network.add(layers.Conv2D(32, (3,3),
18     activation="relu", padding="valid"))
19 >>> network.add(layers.Maxpooling2D(2,2))
20 >>> network.add(Dropout(0.25))
21
22 >>> network.add(Flatten())
23
24 # feedforward layers
25 >>> network.add(Dense(32),
26     activation="relu")
27 >>> network.add(Dropout(0.5))
28
29 >>> network.add(Dense(32),
30     activation="relu")
31 >>> network.add(Dropout(0.5))
32
33 # output layer
34 >>> network.add(Dense(2),
35     activation="sigmoid")
36
37 # loss, optimizer, metrics
38 >>> model.compile(
39     loss="binary_crossentropy",
40     optimizer="sgd",
41     metrics=["accuracy"])
42
43 # training
44 >>> history = network.fit(x_train, y_train
45     , batch_size=32, epochs=50,
46     validation_data=(x_val, y_val))
```

Merk op dat er nog extra lagen zijn toegevoegd: *dropout layers*.

Dropout is een effectieve methode om overfitting te voorkomen en zal willekeurig een aantal elementen van de outputtensor van een laag elimineren, dus 'laten vallen'. De parameter die meegegeven wordt aan de dropout layer is het percentage elementen van de outputtensor dat weggelaten wordt (Chollet, 2018).

14.5.2 Tellen van stomata op een eigen afbeelding

Via de notebook 'Tellen van huidmondjes in een eigen afbeelding' kan men een eigen microfoto van een blad van een plant uploaden en de huidmondjes erop laten tellen.

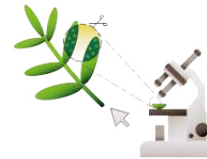
In eerste instantie gebeurt het volgende. Het achterliggende systeem van de notebook zal een filter van 120 x 120 pixels over de foto laten glijden - er wordt daarbij telkens één pixel opgeschoven - en zal elk bekomen vak aan het deep learning-model aanbieden.

Het systeem zal teruggeven hoe zeker het is dat dat vak een stoma bevat of niet. Er werd een drempelwaarde gekozen van 0,5.

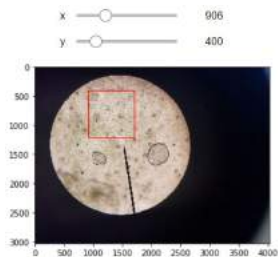
Omdat er telkens slechts één pixel wordt opgeschoven, zal een gedetecteerde stoma in meerdere vakken zitten en dus meerdere keren geteld worden. Het achterliggende systeem van de notebook zal deze echter clusteren en maar één keer tellen.

De clustering kan gebeuren omdat de stomata nooit heel dicht tegen elkaar liggen, dus stomata die wel heel dicht liggen, zijn dezelfde.

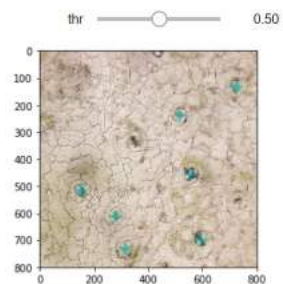
Men kan een foto uploaden in het model. Is die foto groter dan 800 x 800 pixels, dan zal men een gebied van 800 x 800 pixels eruit moeten selecteren. Deze werkwijze wordt getoond in Figuur 14.22. Het model



Bekijk de notebook 'Tellen van huidmondjes in een eigen afbeelding'.



Figuur 14.22: Microfoto blad van vetplant.



Figuur 14.23: Detectie huidmondjes binnen de selectie.

geeft een uitvoer zoals in Figuur 14.23.

Vals positieven en vals negatieven

Vals positieven zijn gedetecteerde stomata die er in werkelijkheid geen zijn.

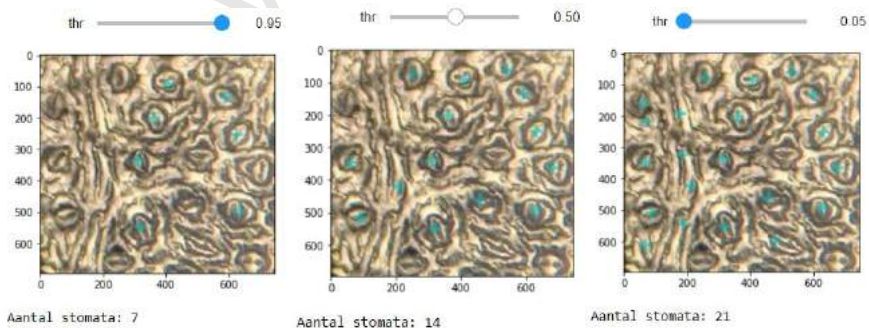
Vals negatieven zijn werkelijke stomata die niet gedetecteerd zijn.

Men kan de drempelwaarde die het model hanteert, zelf aanpassen en op die manier bepalen hoe 'streng' het model moet zijn.

Bij een hoge drempelwaarde zullen er minder stomata gedetecteerd worden, maar er zullen ook minder vals positieven zijn. Er zullen wel meer vals negatieven zijn.

Bij een lage drempelwaarde zullen er meer stomata gedetecteerd worden, maar er zullen ook meer vals positieven zijn. Er zullen minder vals negatieven zijn.

Dit wordt geïllustreerd in Figuren 14.26, 14.25 en 14.24.



Figuur 14.24:
Threshold = 95 %.

Figuur 14.25:
Threshold = 50 %.

Figuur 14.26:
Threshold = 5 %.

Samengevat

Een diep lerend netwerk bestaat uit meerdere lagen die aaneengeschaakeld zijn. Elke laag beschikt over een aantal parameters, zijn gewichten.

Input uit de trainingdata wordt gegeven aan de eerste laag, de input layer. Via berekeningen binnen deze layer wordt een output bekomen. Deze output gaat dan naar de volgende layer, waar een soortgelijk proces plaatsvindt. Tot slot komt de output van de voorlaatste layer aan in de laatste layer, de output layer. De output layer heeft ook een output: het systeem doet een voorspelling. Deze voorspelling wordt vergeleken met het gegeven label. Daarvoor is de kostenfunctie (*loss function*) nodig. De mate waarin de voorspellingen verschillen van de labels, bepaalt de aanpassingen die gedaan worden aan de gewichten. Daarvoor is dan weer de optimalisatiefunctie (*optimizer*) nodig. Een eerste epoch is gedaan. Het systeem onderneemt een tweede epoch: het proces begint opnieuw, maar nu met de aangepaste gewichten. Het aanpassen van de gewichten is wat men bedoelt als men zegt dat het systeem leert. De 'kennis' van het netwerk is opgeslagen in de gewichten.

Deze netwerken hebben eigenlijk een simpele opbouw, die kan verwezenlijkt worden met de bouwblokken van de Python-module Keras.

De netwerken van KIKS gebruiken het *Sequential model* van Keras.

De architect van het netwerk kiest het soort layers en zorgt ervoor dat de data in de juiste vorm aan het netwerk worden aangeboden. De netwerk-architect splitst de data op in training-, validerings- en testdata en legt de *loss function* en de *optimizer* vast, kiest de *learning rate* en een eventuele *threshold*.

Leestip

François Chollet. Deep learning with Python (Chollet, 2018).

EINDTERMEN

Binnen het KIKS-project kunnen behoorlijk veel leerdoelen aan bod komen. De leerkracht bepaalt zelf welke leerdoelen in verband gebracht worden met het project.

In de eindtermen en leerplannen zijn heel wat leerdoelen te vinden die KIKS linken met biologie, aardrijkskunde, chemie en wiskunde.

De gelinkte eindtermen zijn hier opgenomen. De leerplandoelen vindt u op de webpagina <https://aiopschool.be/kiks>.

15.1 Aardrijkskunde

15.1.1 Aardrijkskunde en KIKS

Het thema van de klimaatverandering biedt veel mogelijkheden aan de leerkracht aardrijkskunde om bepaalde leerplandoelen aan te brengen binnen het kader van het KIKS-project. Bovendien kan men met het project de leerlingen actief en zelfstandig laten leren en hen ICT-vaardigheden bijbrengen.

De volgende items uit KIKS hebben een link met het vak aardrijkskunde:

- het broeikaseffect en het versterkt broeikaseffect;
- de invloed van CO₂ op het klimaat;
- de samenhang tussen CO₂ en de temperatuur;
- paleoklimatologie en het gebruik van klimaatproxy's (bv. ijskernen en huidmondjes op fossiele bladeren);

- klimaatveranderingen uit het verre verleden met bv. het uitsterven van sommige flora en fauna;
- de kleine ijstijd met verlenging van de gebergtegletsjers, gevolgd door het smelten van de gebergtegletsjers nu, het albedo-effect;
- systemen die elkaar beïnvloeden, zoals de uitstoot van broeikasgassen door de mens, het klimaat, de fotosynthese, de planten die de CO₂-cyclus van de wereld beïnvloeden, de biodiversiteit en de mondiale voedselvoorziening;
- het verwerken met ICT van de data die NOAA, NASA/GIS en Scripps online ter beschikking stellen;
- het besef dat wetenschappers van allerlei domeinen samenwerken om klimaatmodellen op te stellen.

15.1.2 Eindtermen aardrijkskunde

Vakgebonden eindtermen aardrijkskunde

De leerlingen kunnen

- 1 een verscheidenheid aan ruimtelijke wetenschappen verbinden met allerlei beroepen en met onderzoeksdomeinen;
- 6 weer en klimaat in verband brengen met de opbouw van en met processen in de atmosfeer;
- 7 de invloed van menselijke activiteiten op het milieu zoals: broeikaseffect, natuurrampen, zure regen, waterbeheersing, bodemdegradatie en –verbetering met voorbeelden illustreren;
- 8 de geofysische opbouw van de aarde en de platentektoniek beschrijven en gevolgen ervan zoals: de ligging van oceanen en continenten, vulkanisme en aardbevingen en bepaalde klimaatsveranderingen verklaren;
- 10 productie en consumptie van voedsel en hulpbronnen in relatie brengen met demografische evolutie en welvaartsniveau in het kader van een duurzame ontwikkeling;
- 16 aardrijkskundige gegevens opzoeken, ordenen en op eenvoudige manier verwerken, gebruikmakend van beschikbare, hedendaagse informatiebronnen en –technieken.

De leerlingen zijn

- 27 kritisch tegenover aangeboden informatie zoals die m.b.t. ontwikkelings-, welvaarts- en milieuproblemen;
- 29 bereid om lokale problemen van milieu en samenleving in een globale context te plaatsen.

15.2 Biologie

15.2.1 Biologie en KIKS

In de biologie is het vooral het proces van fotosynthese dat aan bod komt:

- opname van CO₂;
- de rol van fotosynthese in de CO₂-cyclus van de wereld;
- de aanpassing van planten aan de hoeveelheid CO₂ in de atmosfeer en de mogelijke impact op de biodiversiteit en de voedselvoorziening;
- het practicum microscopie over huidmondjes;
- de aanpassing van planten aan omgevingsfactoren: droogte, temperatuur, CO₂-concentratie, zon of schaduw;
- de verantwoordelijkheid van de consument en de producent van AI-systemen.

15.2.2 Eindtermen biologie

Eindtermen voor de basisvorming

- W3 Uit data, een tabel of een grafiek relaties en waarden afleiden om een besluit te formuleren.
- W6 Bij het verduidelijken van en het zoeken naar oplossingen voor duurzaamheidsvraagstukken wetenschappelijke principes hanteren die betrekking hebben op ten minste grondstoffen, energie, biotechnologie, biodiversiteit en het leefmilieu.
- W7 De natuurwetenschappen als onderdeel van de culturele ontwikkeling duiden en de wisselwerking met de maatschappij op ecologisch, ethisch, technisch, socio-economisch en filosofisch vlak illustreren.

Eindtermen biologie

- B1 Celorganellen, zowel op lichtmicroscopisch als op elektronenmicroscopisch niveau, benoemen en de functies ervan aangeven.
- B10 Wetenschappelijk onderbouwde argumenten geven voor de biologische evolutie van organismen, met inbegrip van de mens.

15.3 Chemie

15.3.1 Chemie en KIKS

In de chemie kan de CO₂-concentratie en fotosynthese gelinkt worden aan chemisch rekenen, lewisstructuren, redoxreacties, energieomzettingen, wetenschappelijke vaardigheden en onderzoekscompetenties.

15.3.2 Eindtermen chemie

Eindtermen voor de basisvorming

- *W1 Eigen denkbeelden verwoorden en die confronteren met denkbeelden van anderen, metingen, observaties, onderzoeksresultaten of wetenschappelijke inzichten.*
- *W3 Uit data, een tabel of een grafiek relaties en waarden afleiden om een besluit te formuleren.*
- *W4 Wetenschappelijke terminologie, symbolen en SI-eenheden gebruiken.*
- *W7 De natuurwetenschappen als onderdeel van de culturele ontwikkeling duiden en de wisselwerking met de maatschappij op ecologisch, ethisch, technisch, socio-economisch en filosofisch vlak illustreren.*

Vakgebonden eindtermen chemie

- *C1 Eigenschappen en actuele toepassingen van stoffen, waaronder kunststoffen, verklaren aan de hand van de moleculaire structuur van die stoffen.*

15.4 Natuurwetenschappen

15.4.1 Eindtermen natuurwetenschappen

Specifieke eindtermen natuurwetenschappen

De leerlingen kunnen op verschillende schaalniveaus

- *6 processen waarbij energie wordt getransformeerd of getransporteerd, beschrijven en herkennen in voorbeelden.*
- *9 effecten van de interactie tussen materie en elektromagnetische straling beschrijven en in voorbeelden herkennen.*
- *14 relaties tussen systemen beschrijven en onderzoeken.*
- *17 voorbeelden geven van cyclische processen en deze cycli op een tijdschaal plaatsen.*

- 19 relaties tussen cyclische processen illustreren.

De leerlingen kunnen

- 27 de relatie tussen natuurwetenschappelijke ontwikkelingen en technische toepassingen illustreren.
- 28 effecten van natuurwetenschap op de samenleving illustreren, en omgekeerd.
- 29 zich oriënteren op een onderzoeksprobleem door gericht informatie te verzamelen, te ordenen en te bewerken.

15.5 Wiskunde

15.5.1 Wiskunde en KIKS

In machine learning en deep learning worden wiskunde en principes uit de statistiek gebruikt:

- matrices, indices, dimensie van een matrix en eigenschappen van matrices;
- bewerkingen met matrices en eigenschappen van bewerkingen met matrices;
- sommatieteken;
- verloop van functies bij de grafieken over CO₂, gletsjers ... en metrics van neurale netwerken;
- veeltermfuncties bij regressie;
- functies met meervoudig voorschrift, bv. ReLU, sign, Heaviside;
- toepassing van logistische functie bij de sigmoïde als activatiefunctie;
- maken en interpreteren van grafieken;
- toepassing van telproblemen, bv. byte, aantal hyperparameters;
- toepassing formule oppervlakte van een ellips;
- lokaal en globaal minimum van een functie bij gradient descent;
- afgeleiden en kettingregel bij gradient descent;
- kansrekenen (onafhankelijke kansen - productregel) bij binary cross-entropy;
- eigenschappen van logaritmen bij binary cross-entropy;
- statistiek: belang van standaardiseren, gemiddelde en standaardafwijking, Z-score;
- correlatie en regressie;
- methode van de kleinste kwadraten;

- voorbeelden van een normale en een scheve verdeling bij de verdeling van kleurintensiteit in een grijswaardenafbeelding;
- wiskunde toegepast in digitale beeldverwerking, RGB-code;
- convolutiebewerking.

Het project KIKS leent zich uitstekend om in de wiskundeles in een actieve werkvorm en met ICT aan de slag te gaan.

De leerlingen werken met KIKS aan ICT-vaardigheden, wiskundige taalvaardigheid, denk-, redeneer- en onderzoeksvaardigheden en probleemoplossende vaardigheden.

15.5.2 Eindtermen wiskunde

Algemene eindtermen

De leerlingen kunnen

- 3 eenvoudig mathematiseerbare problemen ontleden (onderscheid maken tussen gegevens en gevraagde, de relevantie van de gegevens nagaan en verbanden leggen ertussen) en vertalen naar een passende wiskundige context;
- 4 wiskundige problemen planmatig aanpakken (door eventueel hiërarchisch op te splitsen in deelproblemen);
- 6 voorbeelden geven van reële problemen die met behulp van wiskunde kunnen worden opgelost;
- 7 bij het oplossen van wiskundige problemen functioneel gebruikmaken van ICT;
- 9 kennis, inzicht en vaardigheden die ze verwerven in wiskunde bij het verkennen, vertolken en verklaren van problemen uit de realiteit gebruiken.

De leerlingen

- 13* zijn gericht op samenwerking om de eigen mogelijkheden te vergroten.

Reële functies en statistiek

De leerlingen

- 14 lezen op een grafiek af: het stijgen, dalen of constant zijn, de eventuele extrema.

De leerlingen kunnen

- 15 bij veeltermfuncties de afgeleide gebruiken als maat voor de ogenblikkelijke verandering en het verband leggen tussen het begrip afgeleide, het begrip differentiequotiënt en de richting van de raaklijn aan de grafiek;
- 18 bij veeltermfuncties de afgeleide functie gebruiken voor het bestuderen van het veranderingsgedrag en voor het opzoeken of verifiëren van extreme waarden en het verband leggen tussen de afgeleide functie en bijzonderheden van de grafiek;
- 19 het begrip afgeleide herkennen in situaties buiten de wiskunde;
- 33 in betekenisvolle situaties gebruikmaken van een normale verdeling als continu model bij data met een klokvormige frequentieverdeling en het gemiddelde en de standaardafwijking van de gegeven data gebruiken als schatting voor het gemiddelde en de standaardafwijking van deze normale verdeling;
- 34 het gemiddelde en de standaardafwijking van een normale verdeling grafisch interpreteren;
- 35 grafisch het verband leggen tussen een normale verdeling en de standaardnormale verdeling.

* Met het oog op de controle door de inspectie werden de attitudes met een * aangeduid in de kantlijn.

Specifieke eindtermen

Algebra, analyse, kansrekening, wiskunde en cultuur, onderzoekscompetenties

De leerlingen kunnen

- 4 met behulp van matrices problemen wiskundig modelleren en oplossen;
- 5 de basiseigenschappen van een reële vectorruimte (beperkt tot dimensie 2 en 3) herkennen en gebruiken;
- 8 de eerste en de tweede afgeleide van functies berekenen en ze in concrete situaties gebruiken.
- 16 wetten van de kansrekening toepassen voor onafhankelijke en voor afhankelijke gebeurtenissen;
- 19 inzicht verwerven in de bijdrage van wiskunde tot de ontwikkeling van exacte en humane wetenschappen, techniek, kunst en het kritische denken;
- 20 zich oriënteren op een onderzoeksprobleem door gericht informatie te verzamelen, te ordenen en te bewerken;

- *21 een onderzoeksopdracht met een wiskundige component voorbereiden, uitvoeren en evalueren;*
- *22 de onderzoeksresultaten en conclusies rapporteren en ze confronteren met andere standpunten.*

(GO!, 2019; Katholiek Onderwijs Vlaanderen, 2019; OVSG, 2020)

Versie 1.0

Versie 1.0

BIBLIOGRAFIE

- Ahn, J., Brook, E. J., Mitchell, L. E., Rosen, J. C., McConnell, J. R., Taylor, K. C., Etheridge, D. M., & Rubino, M. (2012). Atmospheric CO₂ over the last 1000 years: A high-resolution record from the West Antarctic Ice Sheet (WAIS) Divide ice core. *Global Biogeochemical Cycles*, *26*(2).
- American Museum of Natural History (2014). *Keeling's Curve. The Story of CO₂* [Videobestand]. Geraadpleegd op 9 augustus 2020 via <https://youtu.be/0Z8g-smE2sk>.
- Anderson, T. R., Hawkins, E., & Jones, P. D. (2016). CO₂, the greenhouse effect and global warming: from the pioneering work of Arrhenius and Callendar to today's Earth System Models. *Endeavour*, *40*(3), 178 – 187.
- Assouline, S. & Or, D. (2013). Plant Water Use Efficiency over Geological Time: Evolution of Leaf Stomata Configurations Affecting Plant Gas Exchange. *PLoS one*, *8*.e67757.
- Bauters, M., Meeus, S., Barthel, M., Stoffelen, P., Deurwaerder, H. D., Meunier, F., Drake, T. W., Ponette, Q., Ebuy, J., Vermeir, P., Beeckman, H., wyffels, F., Bodé, S., Verbeeck, H., Vandeloos, F., & Boeckx, P. (2020). Century-long apparent decrease in iWUE with no evidence of progressive nutrient limitation in African tropical forests. *Global Change Biology*, *26*(8), 4449–4461.
- Berling, D. (2017). *The Emerald Planet*. Oxford University Press.
- Berling, D. J. & Chaloner, W. G. (1993). The Impact of Atmospheric CO₂ and Temperature Changes on Stomatal Density: Observation from *Quercus robur* Lammas Leaves. *Annals of Botany*, *71*(3), 231–235.

- Benjah-bmm27 (2006). *Carbon dioxide* [Afbeelding]. Public domain. Geraadpleegd op 24 oktober 2019 via <https://commons.wikimedia.org/wiki/File:Carbon-dioxide-3D-balls.png>.
- Berbers, Y., Debeuckelaere, W., Herdt, P. D., Desmedt, Y., Smet, F. D., Hildebrandt, M., Poels, K., Pierson, J., Preneel, B., & Vandewalle, J. (2017). *Privacy in tijden van internet, sociale netwerken en big data*. Brussel: KVAB Standpunt 49.
- Bergmann, D. (2015). *Stomata and global climate cycles* [Videobestand]. Geraadpleegd op 30 augustus 2019 via <https://youtu.be/eD2J3PBoERI>.
- Binek, R. (2019). *Kosaciec szczecinkowaty Iris setosa* [Afbeelding]. CC BY-SA 3.0. Verkregen via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Kosaciec_szczecinkowaty_Iris_setosa.jpg.
- Boden, M. A. (2016). *AI. Its nature and future*. Oxford, Verenigd Koninkrijk: Oxford University Press.
- Botsman, R. (2017). Big data meets Big Brother as China moves to rate its citizens. Geraadpleegd op 15 oktober 2019 via <https://www.wired.co.uk/article/chinese-government-social-credit-score-privacy-invasion>.
- Boussemaere, P. (2015). *Eerste hulp bij klimaatverwarring. Waarom de opwarming van de aarde veel meer is dan een milieuprobleem*. Leuven: Davidsfonds Uitgeverij nv.
- Bouwma, R. (2018). Gezichtsherkenning op Chinese scholen betreft luie studenten. Geraadpleegd op 15 oktober 2019 via <https://www.computeridee.nl/nieuws/gezichtsherkenning-op-chinese-scholen-betrapt-luie-studenten/>.
- Bralower, T. & Bice, D. (2020). EARTH IN THE FUTURE. Feedback Mechanisms. Geraadpleegd op 29 juli 2020 via <https://www.e-education.psu.edu/earth103/node/668>.
- Bromet, F. (2016). *Mobiel bellen in 1998* [Videobestand]. Geraadpleegd op 8 augustus 2020 via <https://youtu.be/TNwhIHqM60g>.
- Byron, G. G. (2015). She walks in beauty. Geraadpleegd op 21 september 2019 via <https://www.poetryfoundation.org/poems/43844/she-walks-in-beauty>.
- Caine, R. S., Yin, X., Sloan, J., Harrison, E. L., Mohammed, U., Fulton, T., Biswal, A. K., Dionora, J., Chater, C. C., Coe, R. A., Bandyopadhyay, A., Murchie, E. H., Swarup, R., Quick, W. P., & Gray, J. E. (2019). Rice with reduced stomatal density conserves

- water and has improved drought tolerance under future climate conditions. *New Phytol*, 221, 371–384.
- Caiyu, L. (2018). Chinese schools monitor students activities, targeting truancy with 'intelligent uniforms'. *Global Times*. Geraadpleegd op 15 oktober 2019 via <http://www.globaltimes.cn/content/1132856.shtml>.
- Callaert, H., Bekaert, H., Goethals, C., Provoost, L., & Vancaudenberg, M. (2012). *Correlatie: exploratieve methoden: werktekst voor de leerling*. Statistiek voor het secundair onderwijs. Universiteit Hasselt. Geraadpleegd op 15 april 2019 via https://www.uhasselt.be/documents/uuhasselt@school/lesmateriaal/statistiek/Lesmateriaal/LEERLING%20Correlatie_02.pdf.
- Callaert, H. & Bogaerts, S. (2012). Statistische intelligentie: de samenhang ontdekken: exploratie van bivariaat cijfermateriaal. Universiteit Hasselt. Geraadpleegd op 15 april 2019 via <https://docplayer.nl/32671814-Statistische-intelligentie.html>.
- Camargo, M. & Marengo, R. (2011). Density, size and distribution of stomata in 35 rainforest tree species in Central Amazonia. *Acta Amazonica*, 41(2), 205–212. via e-mail.
- Camargo, M. & Marengo, R. (2012). Growth, leaf and stomatal traits of crabwood (*Carapa guianensis* Aubl.) in Central Amazonia. *Revista Árvore*, 36, 07–16.
- CarbonTracker (2019). *Carbon Dioxide Pumphandle 2019* [Videobestand]. Geraadpleegd op 9 augustus 2020 via <https://www.esrl.noaa.gov/gmd/ccgg/trends/history.html>.
- Carl H. Pforzheimer Collection of Shelley and His Circle. The New York Public Library (1835). *Watercolor portrait of Ada Byron* [Afbeelding]. Geraadpleegd op 22 september 2019 via <http://digitalcollections.nypl.org/items/8562182e-4845-db2b-e040-e00a18060ea6>.
- Carpentier De Changy, O. (2018). Hoe AI kan helpen bij het beantwoorden van de dagelijkse vraag: "Wat eten we vanavond?". Geraadpleegd op 6 september 2019 via <https://news.microsoft.com/nl-be>.
- Chan, T. F. (2019). 22 angstaanjagende foto's van hoe China gezichtsherkenning gebruikt om burgers in de gaten te houden. Geraadpleegd op 15 oktober 2019 via <https://www.businessinsider.nl/china-gezichtsherkenning-burgers-camera/>.

- Chollet, F. (2018). *Deep learning with Python*. Shelter Island, NY: Manning Publications Co.
- Clay Mathematics Institute (2019). P vs NP Problem. Geraadpleegd op 22 september 2019 via <https://www.claymath.org/millennium-problems/p-vs-np-problem>.
- Coeckelbergh, M. (2020). *AI ETHICS*. Cambridge, MA: The MIT Press.
- Commissie voor de Belgische Overzeese Biografie (2015). *BELGISCHE OVERZEESE BIOGRAFIE IX*. Brussel: KONINKLIJKE ACADEMIE VOOR OVERZEESE WETENSCHAPPEN. Website: <http://www.kaowarsom.be>.
- Connor, N. (2018). Chinese school uses facial recognition to monitor student attention in class. Geraadpleegd op 15 oktober 2019 via <https://www.telegraph.co.uk/news/2018/05/17/chinese-school-uses-facial-recognition-monitor-student-attention>.
- Crutzen, P. & Stoermer, E. (2000). The Anthropocene. *IGBP Global change newsletter*, 41, 17–18. Geraadpleegd op 14 oktober 2019.
- CSIRO. Atmospheric Research (2019). *Bubbles in Ice* [Afbeelding]. CC BY 3.0. Geraadpleegd op 23 september via <https://www.scienceimage.csiro.au/image/521/bubbles-in-ice/>.
- Daubechies, I. (2015). Big Data's Mathematical Mysteries. Geraadpleegd op 28 juli 2019 via <https://www.quantamagazine.org/big-datas-mathematical-mysteries-20151203/>.
- davcjal (2015). *Stomatal Closure in Tradescantia Leaf Cells* [Videobestand]. Geraadpleegd op 30 augustus 2019 via <https://youtu.be/AwyrqfNTuxQ>.
- de Boer, H. J., Price, C., Wagner-Cremer, F., Dekker, S., Franks, P., & Veneklaas, E. (2016). Optimal allocation of leaf epidermal area for gas exchange. *New Phytologist*, 210(4), 1219–1228.
- De Boosere, F. (2010). Ten opzichte van welk nulpunt gebeuren de hoogtemetingen voor landkaarten? Geraadpleegd op 21 januari 2020 via <https://www.frankdeboosere.be/vragen/vraag72.php>.
- De Moor, T. (2019). Hoe burgerwetenschap de valorisatie van kennis kan versnellen. *Lanceringsmoment Scivil*. Keynote op lanceringsmoment Scivil, Vlaamse kenniscentrum voor Citizen Science.

- De Moor, T., Rijpma, A., & López, M. P. (2019). Dynamics of engagement in Citizen Science: Results from the “Yes, I do!” project. *Citizen Science: Theory and Practice*, 4(1), 38.
- Dienst Klimaatverandering van de Federale overheidsdienst Volksgezondheid, Veiligheid van de Voedselketen en Leefmilieu (2019). *Enkele mijlpalen in de geschiedenis van het klimaatbeleid*. Geraadpleegd op 29 mei 2019 via <https://www.klimaat.be/nl-be/klimaatbeleid/internationaal-klimaatbeleid>.
- Dua, D. & Taniskidou, E. K. (2017). *UCI Machine Learning Repository*. Irvine, CA: University of California, School of Information and Computer Science. Geraadpleegd op 28 juli 2019 via <http://archive.ics.uci.edu/ml>.
- Duke today (2019). Harvard Presents Ingrid Daubechies With Honorary Degree. Geraadpleegd op 20 oktober 2019 via <https://today.duke.edu/2019/06/harvard-present-ingrid-daubechies-honorary-degree>.
- Durham University (2018). Collapse of civilizations worldwide defines youngest unit of the Geologic Time Scale. Geraadpleegd op 16 oktober 2019 via <https://www.dur.ac.uk/earth.sciences/news/?itemno=35201>.
- DWDD (2019). *Mobiel bellen: 1998 vs NU* [Videobestand]. Geraadpleegd op 8 augustus 2020 via <youtu.be/mQVkpdzPGtQ>.
- Eckert, M. (2019). Nip-test screent ook moeder op genetische defecten. *De Standaard*. Geraadpleegd via Gopress op 9 september 2019 via https://www.standaard.be/cnt/dmf20190904_04592057.
- Eliza (2018). *ELIZA conversation* [Afbeelding]. Publiek domein. Geraadpleegd op 28 juli 2019 via https://commons.wikimedia.org/wiki/File:ELIZA_conversation.jpg.
- Enix, S. (2018). *Shadow-of-the-Tomb-Raider-010419-002* [Afbeelding]. Geraadpleegd op 6 september 2019 via <https://www.flickr.com/photos/161227653@N02/47513423051/>. Publiek domein.
- ESRL (2019). *LAB ACTIVITY: WHY IS CO₂ SO SPECIAL?* Geraadpleegd op 13 september 2019 via https://www.esrl.noaa.gov/gmd/education/info_activities/pdfs/LA_whats_so_special_about_co2.pdf.
- ESRL/GMD (2019). *FTP Data Finder*. Geraadpleegd op 18 augustus 2019 via <https://www.esrl.noaa.gov/gmd/dv/data/index.php?category=Greenhouse%2BGases>.

Eugster, S. (2006). *Lewis CO₂* [Afbeelding]. CC BY-SA 3.0. Geraadpleegd op 24 oktober 2019 via https://commons.wikimedia.org/wiki/File:Lewis_CO2.svg.

Eynikel, J. (2017). *Robot aan het stuur. Over de ethiek van techniek*. Tielt: Lannoo.

Fang, F. (2018). Chinese Schools Track Students by Requiring Chip-Enhanced Uniforms. *The Epoch Times*. Geraadpleegd op 15 oktober 2019 via <https://www.theepochtimes.com/>.

Fischer, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 179–188.

Franks, P. J. & Beerling, D. J. (2009). Maximum leaf conductance driven by CO₂ effects on stomatal size and density over geologic time. *Proceedings of the National Academy of Sciences*, 106(25), 10343–10347.

Friedlingstein, P., Jones, M. W., O'Sullivan, M., Andrew, R. M., Hauck, J., Peters, G. P., Peters, W., Pongratz, J., Sitch, S., Le Quéré, C., Bakker, D. C. E., Canadell, J. G., Ciais, P., Jackson, R. B., Anthoni, P., Barbero, L., Bastos, A., Bastrikov, V., Becker, M., Bopp, L., Buitenhuis, E., Chandra, N., Chevallier, F., Chini, L. P., Currie, K. I., Feely, R. A., Gehlen, M., Gilfillan, D., Gkritzalis, T., Goll, D. S., Gruber, N., Gutekunst, S., Harris, I., Haverd, V., Houghton, R. A., Hurtt, G., Ilyina, T., Jain, A. K., Joetzjer, E., Kaplan, J. O., Kato, E., Klein Goldewijk, K., Korsbakken, J. I., Landschützer, P., Lauvset, S. K., Lefèvre, N., Lenton, A., Lienert, S., Lombardozzi, D., Marland, G., McGuire, P. C., Melton, J. R., Metzl, N., Munro, D. R., Nabel, J. E. M. S., Nakaoka, S.-I., Neill, C., Omar, A. M., Ono, T., Peregón, A., Pierrot, D., Poulter, B., Rehder, G., Resplandy, L., Robertson, E., Rödenbeck, C., Séférian, R., Schwinger, J., Smith, N., Tans, P. P., Tian, H., Tilbrook, B., Tubiello, F. N., van der Werf, G. R., Wiltshire, A. J., & Zaehle, S. (2019). Global Carbon Budget 2019. *Earth System Science Data*, 11(4), 1783–1838.

Futureism (2017). *This 'Racist soap dispenser' at Facebook office does not work for black people* [Videobestand]. Geraadpleegd op 8 augustus 2020 via https://youtu.be/YJjv_0eiHmo.

Gabriels, K. (2016). *Onlife*. Lannoo.

Gabriels, K. (2019). *Regels voor robots. Ethiek in tijden van AI*. Academic And Scientific Publishers.

Garderen, F. V. (2019). Nog steeds klachten over gebrekkige begeleiding: veel zwangere vrouwen weten niet wat NIPT inhoudt. *De Morgen*. Geraadpleegd 15 september 2019 via <https://www.demorgen.be/nieuws>.

- Gavira, M. (2018). How Netflix uses AI and Data to conquer the world. Geraadpleegd op 6 september 2019 via <https://www.linkedin.com/pulse/how-netflix-uses-ai-data-conquer-world-mario-gavira>.
- Gijssels, L., Huyse, T., & Hoyweghen, I. V. (2019). *Citizen science*. Antwerpen: Pelckmans Pro.
- GO! (2019). Leerplannen en lessentabellen secundair onderwijs. Geraadpleegd op 15 oktober 2019 via <https://pro.g-o.be/pedagogische-begeleiding-leerplannen-nascholing/leerplannen/leerplannen-so>.
- Godoy, D. (2018). Understanding binary cross-entropy / log loss: a visual explanation [Blogpost]. Geraadpleegd op 11 augustus 2019 via <https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a>.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. Cambridge, MA: MIT Press.
- Gowik, U. & Westhoff, P. (2011). The Path from C₃ to C₄ Photosynthesis. *Plant Physiology*, 155(1), 56–63.
- Güzeldere, G. & Franchi, S. (1995). Dialogues with colorful “personalities” of early AI. *Stanford Hum. Rev.*, 4(2), 161–169.
- Hardesty, L. (2009). Explained: P vs. NP. Geraadpleegd op 21 september 2019 via <http://news.mit.edu/2009/explainer-pnp>.
- Hautekiet, G., Roelens, M., & Dooms, A. (2016). Wiskunde achter beeldverwerking. *Uitwisseling*, 32(4), 6–24.
- Haykin, S. (2009). *Neural Networks and Learning Machines*. New Jersey: Pearson Education.
- Hazen, R., Grew, E., Origlieri, M., & Downs, R. (2017). On the mineralogy of the “Anthropocene Epoch”. *American Mineralogist*, 102, 595–611.
- Hepworth, C., Caine, R. S., Harrison, E. L., Sloan, J., & Gray, J. E. (2018). Stomatal development: focusing on the grasses. *Current Opinion in Plant Biology*, 41, 1–7.
- Hetherington, A. M. & Woodward, F. I. (2003). The role of stomata in sensing and driving environmental change. *Nature*, 424, 901–908.

Hubau, W., Lewis, S. L., Phillips, O., Affum-Baffoe, K., Beeckman, H., Sanchez, A. C., Daniels, A. K., Ewango, C. E. N., Fauset, S., Mukinzi, J. M., Sheil, D., Sonké, B., Sunderland, T. C. H., Taedoumg, H., Thomas, S., White, L., Abernethy, K., & Adu-Bredu, S. (2020). Asynchronous carbon sink saturation in African and Amazonian tropical forests. *Nature*, *579*, 80–87.

Hyde, M., Wursten, B., Ballings, P., & Palgrave, M. C. (2019). List of people. Geraadpleegd op 17 september 2019 via <https://www.zimbabweflora.co.zw/speciesdata/persons.php>.

IPCC (2018). IPCC: Summary for Policymakers. In *Global Warming of 1.5°C: An IPCC Special Report on the impacts of global warming of 1.5°C above pre-industrial levels and related global greenhouse gas emission pathways, in the context of strengthening the global response to the threat of climate change, sustainable development, and efforts to eradicate poverty*. [V. Masson-Delmotte and P. Zhai and H.-O. Pörtner and D. Roberts and J. Skea and P.R. Shukla and A. Pirani and W. Moufouma-Okia and C. Péan and R. Pidcock and S. Connors and J.B.R. Matthews and Y. Chen and X. Zhou and M.I. Gomis and E. Lonnoy and T. Maycock and M. Tignor and T. Waterfield (eds.)]. World Meteorological Organization, Geneva, Switzerland, 32 pp. Geraadpleegd op 5 september 2019 via <https://www.ipcc.ch/sr15/chapter/spm/>.

IPCC (2019). IPCC: Summary for Policymakers. In *IPCC Special Report on the Ocean and Cryosphere in a Changing Climate*. [H.-O. Pörtner, D.C.Roberts, V. Masson-Delmotte, P. Zhai, M. Tignor, E. Poloczanska, K. Mintenbeck, M. Nicolai, A. Okem, J. Petzold, B. Rama, N. Weyer (eds.)]. In Press. Geraadpleegd op 21 augustus 2020 via https://report.ipcc.ch/srocc/pdf/SROCC_SPM_Approved.pdf.

Jacobs, F. (2016). Kunstmatige intelligentie verandert beroep van de radioloog. Geraadpleegd op 6 september via <https://www.smarthealth.nl/2016/11/28/philiips-kunstmatige-intelligentie-radiologie/>.

Katholiek Onderwijs Vlaanderen (2019). Curriculum secundair onderwijs. Geraadpleegd op 15 oktober 2019 via <http://ond.vvkso-ict.com/lele/leerplannen.asp>.

Keeling, C. D. (1998). Rewards and Penalties of Monitoring the Earth. *Annual Review of Energy and the Environment*, *23*(1), 25–82.

Kelly, K. (2016). How AI can bring on a second industrial revolution. Geraadpleegd op 14 juni 2019 via <https://www.youtube.com/watch?v=IjbTiRbeNpM>.

- Keras (2019). Getting started with the Keras Sequential model. Geraadpleegd op 3 oktober 2019 via <https://keras.io/getting-started/sequential-model-guide/>.
- Kröger, F. (2016). *Autonomous Driving. Automated Driving in Its Social, Historical and Cultural Contexts*. Berlin, Heidelberg: Springer.
- Kulp, S. A. & Strauss, B. H. (2019). New elevation data triple estimates of global vulnerability to sea-level rise and coastal flooding. *Nature communications*, 10, 4844.
- Lammertsma, E. I., de Boer, H. J., Dekker, S. C., Dilcher, D. L., Lotter, A. F., & Wagner-Cremer, F. (2011). Global CO₂ rise leads to reduced maximum stomatal conductance in Florida vegetation. *Proceedings of the National Academy of Sciences*, 108(10), 4035–4040.
- Langlois, D. (2019). No machine-readable author provided. dlanglois assumed (based on copyright claims). *Iris* [Afbeelding]. CC BY-SA. Verkregen via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Iris_versicolor_3.jpg.
- Large, M. F., Nessia, H. R., Cameron, E. K., & Blanchon, D. J. (2017). Changes in Stomatal Density over Time (1769-2015) in the New Zealand Endemic Tree *Corynocarpus laevigatus* J. R. Forst. & G. Forst. (*Corynocarpaceae*). *Pacific Science*, 71(3), 319–328.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436–444.
- Legg, S. & Hutter, M. (2007). Universal Intelligence: A Definition of Machine Intelligence. *Minds & Machines*, 17(4), 391–444.
- Lewis, S. & Maslin, M. (2015). Defining the Anthropocene. *Nature*, 519, 171–180.
- Lim, M. (2018). History of AI Winters. Geraadpleegd op 27 december 2018 via <https://www.actuaries.digital/2018/09/05/history-of-ai-winters/>.
- Mann, M. E. (2002). Little Ice Age. *Encyclopedia of Global Environmental Change*. (volume 1, pp. 504-509).
- Mansfeld, P. A. (2011). *Cistus creticus* [Afbeelding]. CC BY 3.0. Geraadpleegd op 12 juni 2019 via https://commons.wikimedia.org/wiki/File:Cistus_creticus_pm02.jpg.
- Marr, B. (2018). How AI can bring on a second industrial revolution. Geraadpleegd op 14 juni 2019 via <https://www.forbes.com/sites/bernardmarr/>.

- Martens, G. (2019). What if Waze were evil? [Blogpost]. Geraadpleegd op 22 september 2019 via <https://www.duvalunionconsulting.com/what-if-waze-were-evil/>.
- Maslin, M. (2014). *Climate Change: A Very Short Story*. Oxford, Verenigd Koninkrijk: Oxford University Press.
- Mattel (2017). Hello barbie. Geraadpleegd op 5 september 2019 via <http://hellobarbiefaq.mattel.com/faq/>.
- Mayfield, F. (2007). *Iris virginica shrevei BLUE FLAG* [Afbeelding]. CC BY-SA 2.0, verkregen via Wikimedia Commons https://commons.wikimedia.org/wiki/File:Iris_virginica.jpg.
- McGrath, S. (2007). *Bamboo* [Afbeelding]. CC BY 2.0, via Wikimedia Commons. Geraadpleegd op 19 mei 2019 via [https://nl.wikipedia.org/wiki/Bestand:Bamboo_\(1466706101\).jpg](https://nl.wikipedia.org/wiki/Bestand:Bamboo_(1466706101).jpg).
- McKinney, W. (2018). *Python for Data Analysis*. Sebastopol, Californië: O'Reilly Media.
- McLaughlin, B. (2018). Chinese EdTech Company Proves AI Teaches Better Than Actual Teachers. Geraadpleegd op 15 oktober 2019 via <https://www.forbes.com/sites/baymclaughlin/2018/01/31/this-week-in-china-tech-robots-beat-teachers-in-classroom-tencent-builds-for-blockchain>.
- Meeus, S., Van den Bulcke, J., & wyffels, F. (2020). From leaf to label: A robust automated workflow for stomata detection. *Ecology and Evolution*, 00, 1–14.
- Milbau, A., Vandeplas, N., Kockelbergh, F., & Nijs, I. (2017). Both seed germination and seedling mortality increase with experimental warming and fertilization in a subarctic tundra. *AoB PLANTS*, 9(5).
- Nardone, C. (2007). *Shakey the Robot* [Afbeelding]. CC BY-SA 2.0. Geraadpleegd op 28 juli 2019 via [https://commons.wikimedia.org/wiki/File:Shakey_the_Robot_\(developed_between_1966-1972_at_SRI_International\)_-_Computer_History_Museum_\(2007-11-10_23.16.01_by_Carlo_Nardone\).jpg](https://commons.wikimedia.org/wiki/File:Shakey_the_Robot_(developed_between_1966-1972_at_SRI_International)_-_Computer_History_Museum_(2007-11-10_23.16.01_by_Carlo_Nardone).jpg).
- Narula, G. (2019). Everyday Examples of Artificial Intelligence and Machine Learning. Geraadpleegd op 29 mei 2019 via <https://www.techemergence.com/everyday-examples-of-ai>.

- NASA (2009). Climate and Earth's Energy Budget. Geraadpleegd op 25 juli 2020 via <https://earthobservatory.nasa.gov/features/EnergyBalance>.
- NASA (2012). *The Moon and Earth's atmosphere* [Afbeelding]. Geraadpleegd op 24 oktober 2019 via https://www.nasa.gov/mission_pages/station/multi-media/gallery/iss030e031275.html.
- NASA (2020a). Keeping a steady eye on sea level change from space. Geraadpleegd op 21 augustus 2020 via <https://sealevel.nasa.gov/news/188/keeping-a-steady-eye-on-sea-level-change-from-space>.
- NASA (2020b). Sea level. Geraadpleegd op 20 augustus 2020 via <https://climate.nasa.gov/vital-signs/sea-level/>.
- NASA earth observatory (2011). The Carbon Cycle. Geraadpleegd op 29 juli 2020 via <https://earthobservatory.nasa.gov/features/CarbonCycle>.
- NCEI (2019). What are proxy data? Geraadpleegd op 29 mei 2019 via <https://www.ncdc.noaa.gov/news/what-are-proxy-data>.
- Nerbonne, J. (2004). Logistische regressie. Geraadpleegd op 15 augustus 2020 via <https://www.let.rug.nl/nerbonne/teach/stats/Moore-McCabe-H15.pdf>.
- Nese, J. & Schwartz, G. (2002). *The Philadelphia Area Weather Book*. Verenigde Staten: Temple University Press.
- Neukom, R., Steiger, N., Gómez-Navarro, J., Wang, J., & Werner, J. (2019). No evidence for globally coherent warm and cold periods over the preindustrial Common Era. *Nature*, 571, 550.
- Nistor, A., Clemente-Gallardo, J., Angelopoulos, T., Chodzinska, K., Gallardo, M. C., Gozdzik, A., Gras-Velazquez, A., Grizelj, A., Kolenberg, K., Mitropoulou, Gatt, A. D. M., Tasiopoulou, E., Brunello, A., Echard, P., Arvaniti, V., Carroll, S., Cindea, N., Diamantopoulos, N., Duquenne, N., Edrissy, S., Ferguson, E., Galani, L., Glezou, K., Kameas, A., Kirmaci, H., Koliakou, I., Konomi, E., Kontopidi, E., Kulic, S., Lefkos, I., Nikolettakis, G., Siotou, E., Šimac, A., Sormani, F., Tramonti, M., Tsapara, M., Tsourlidaki, E., & Vojinovic, M. (2019). Bringing Research into the Classroom: The Citizen Science approach in schools. *Scientix Observatory report*. European Schoolnet, Brussels.
- NOAA (2008). Temperature Change and Carbon Dioxide Change. Geraadpleegd op 31 augustus 2019 via <https://www.ncdc.noaa.gov/global-warming/temperature-change>.

- O'Halloran, T. L., Law, B. E., Goulden, M. L., Wang, Z., Barr, J. G., Schaaf, C., Brown, M., Fuentes, J. D., Gökcede, M., Black, A., & Engel, V. (2012). Radiative forcing of natural forest disturbances. *Global Change Biology*, *18*(2), 555–565.
- Oostra, L. (2019). AI bepaalt of trainees bij Unilever aangenomen worden: 'Je cv werkt niet meer voor je'. Geraadpleegd op 6 september 2019 via <https://www.mt.nl/management/ai-trainees-unilever-cv-werkt-niet-meer-voor-je/570440>.
- OVSG (2020). Secundair onderwijs. Geraadpleegd op 23 januari 2020 via <https://www.ovsg.be/leerplannen/secundair-onderwijs>.
- Peng, T. (2018). Chinese Publisher Introduces AI Textbooks For Preschoolers. Geraadpleegd op 15 oktober 2019 via <https://medium.com/syncedreview/>.
- Rammeloo, E. (2019). Peking rolt het sociale kredietsysteem uit: 'Zonder score kun je straks niets meer'. *Knack*. Geraadpleegd op 15 oktober 2019 via <https://www.knack.be/nieuws/wereld/>.
- Reich, P. B., Hobbie, S. E., Lee, T. D., & Pastore, M. A. (2018). Unexpected reversal of C3 versus C4 grass response to elevated CO₂ during a 20-year field experiment. *Science*, *360*(6386), 317–320.
- Resplandy, L., Keeling, R. F., Rödenbeck, C., Stephens, B. B., Khatiwala, S., Rodgers, K. B., Long, M. C., Bopp, L., & Tans, P. P. (2018). Revision of global carbon fluxes based on a reassessment of oceanic and riverine carbon transport. *Nature Geoscience*, *11*, 504–509.
- Rijksuniversiteit Groningen (2005). *Neurale Netwerken* [PowerPoint slides]. Geraadpleegd op 20 oktober 2019 via <https://www.slideserve.com/yitro/neurale-netwerken>.
- Robinson, R. (2017). Convolutional Neural Networks - Basics: An Introduction to CNNs and Deep Learning. Geraadpleegd op 10 oktober 2019 via <https://mlnotebook.github.io/post/CNN1/>.
- Roepers (2004). *Arabidopsis thaliana* [Afbeelding]. CC BY-SA 3.0 Geraadpleegd op 2 september 2019 via https://commons.wikimedia.org/wiki/File:Arabidopsis_thaliana.jpg.
- Ruddiman, W. R. & Ellis, E. C. (2009). Effect of per-capita land use changes on Holocene forest clearance and CO₂ emissions. *Quaternary Science Reviews*, *28*, 3011–3015.

- Rull, V. (2017). The “Anthropocene”: neglects, misconceptions, and possible futures. *Embo reports*, 18, 1056–1060.
- Rummens, S. & Vercayie, D. (2012). Robotmaaiers gevaar voor egels. Geraadpleegd op 29 juli 2019 via <https://www.natuurpunt.be/nieuws/robotmaaiers-gevaar-voor-egels-20190726>.
- Rupp, K. (2019). *Data transistoren*. Geraadpleegd op 6 september 2019 via <https://github.com/karlrupp/microprocessor-trend-data>.
- Sci- Inspi (2018). *Monocot and Eudicot Germination Time-lapse* [Videobestand]. Geraadpleegd op 30 augustus 2019 via <https://youtu.be/WbG5zu2Vw0I>.
- Science Museum Group Collection (1833). Portion of Babbage’s calculating machine, Difference Engine no.1. Geraadpleegd op 22 september 2019 via <https://collection.sciencemuseumgroup.org.uk/objects/co62243/difference-engine-no-1-difference-engine-portion-only>.
- Scripps Institution of Oceanography (2019). *The Keeling Curve*. Geraadpleegd op 30 augustus 2019 via <https://scripps.ucsd.edu/programs/keelingcurve/>.
- Selwyn, N. (2019). The future of EdTech in schools? Just look at what they’re doing in China. Geraadpleegd op 15 oktober 2019 via <https://lens.monash.edu/2019/01/09>.
- Simpson, R. (2019). Wat is burgerwetenschap? Geraadpleegd op 16 januari 2019 via <https://www.iedereenwetenschapper.be/article/wat-burgerwetenschap>.
- Sloterdijk, P. (2018). *Wat gebeurde er in de twintigste eeuw*. Amsterdam: Boom uitgevers.
- Steels, L., Berendt, B., Pizurica, A., & Vandewalle, J. (2017). *Artificiële intelligentie. Naar een vierde industriële revolutie?* Brussel: KVAB Standpunt 53.
- Steinhorsdottir, M., Vajda, V., & Pole, M. (2019). Significant transient pCO₂ perturbation at the New Zealand Oligocene-Miocene transition recorded by fossil plant stomata. *Palaeogeography, Palaeoclimatology, Palaeoecology*, 515, 152–161.
- Sterrenwacht Armand Pien (2018). *Fotosynthese*. Geraadpleegd op 24 december 2018 via <https://www.rug-a-pien.be/docs/fotosynthese.pdf>.
- Tans, P. (2018). National Oceanic and Atmospheric Administration (NOAA), Verenigde Staten. Via e-mail.

- Tegmark, M. (2017). *LIFE 3.0*. New York, Verenigde Staten: Vintage Books, Penguin Random House, eerste editie edition.
- Thanukos, A. (2018). Ancient fossils and modern climate change: The work of Jennifer McElwain, based upon interviews with Jennifer McElwain. https://evolution.berkeley.edu/evolibrary/article/mcelwain_01 t.e.m. https://evolution.berkeley.edu/evolibrary/article/mcelwain_08.
- The Straits Times (2019). Schools in China's Guangzhou to start pilot AI courses in September. *The Straits Times*. Geraadpleegd op 15 oktober 2019 via <https://www.straitstimes.com/asia/east-asia/>.
- Trouet, V. (2020). *Wat bomen ons vertellen: Een geschiedenis van de wereld geschreven in jaarringen*. Lannoo.
- UCAR (2012). Carbon Dioxide Absorbs and Re-emits Infrared Radiation. Geraadpleegd op 13 september 2019 via <https://scied.ucar.edu/carbon-dioxide-absorbs-and-re-emits-infrared-radiation>.
- van den Brandhof, A. (2009). P = NP? Dat is de vraag. Geraadpleegd op 21 september 2019 via <https://www.nemokennislink.nl/publicaties/p-np-dat-is-de-vraag/>.
- van der Beek, P. (2019). Politie positief over inzet AI bij cold cases. Geraadpleegd op 6 september 2019 via <https://www.computable.nl/artikel/nieuws/big-data/6599469/250449/politie-positief-over-inzet-ai-bij-cold-cases.html>.
- Vellekoop, J. (2015). *Bolide impact and long- and short term environmental change across the Cretaceous-Paleogene boundary*. Utrecht Studies in Earth Sciences No. 73.
- Vlaamse Milieumaatschappij (2019). *Het milieurapport Vlaanderen*. Geraadpleegd op 29 mei 2019 via <https://www.milieurapport.be/milieuthemas/klimaatverandering>.
- Wageningen University & Research (2020). Klimaat en bodem. Geraadpleegd op 24 augustus 2020 via <https://www.wur.nl/nl/Dossiers/dossier/Klimaat-en-bodem-1.htm>.
- Wall Street Journal video (2019). *Under AI's WatchfulwikiEye, China Wants to Raise Smarter Students* [Videobestand]. Geraadpleegd op 7 oktober 2019 via <https://www.wsj.com/video/>.

- Wiggins, A. & Crowston, K. (2011). From Conservation to Crowdsourcing: A Typology of Citizen Science. *Proceedings of the Annual Hawaii International Conference on System Sciences*, (pp. 1–10).
- Wikipedia (2019). Shakey the robot. Geraadpleegd op 26 juli 2019 via https://en.wikipedia.org/wiki/Shakey_the_robot.
- Wolfram, S. (2015). Untangling the Tale of Ada Lovelace [Blog post]. Geraadpleegd op 21 september 2019 via <https://blog.stephenwolfram.com/2015/12/untangling-the-tale-of-ada-lovelace/>.
- Woodward, I., Lake, J., & Quick, W. (2002). Stomatal development and CO₂: ecological consequences. *New Phytologist*, 153, 477–484.
- Yamori, W., Hikosaka, K., & Way, D. A. (2013). Temperature response of photosynthesis in C₃, C₄, and CAM plants: temperature acclimation and temperature adaptation. *Photosynthesis research*, 119.
- Yu, C. (2018). Chinese company's new AI learning system more personalized. *China Dailey*. Geraadpleegd op 15 oktober 2019 via <http://www.chinadaily.com.cn>.
- Zemp, M., Huss, M., Thibert, E., Eckert, N., McNabb, R., Bannwart, J., Barandun, M., Machguth, H., Nussbaumer, S., Gärtner-Roer, I., Thomson, L., Paul, F., Maussion, F., Kutuzov, S., & Cogley, J. (2019). Global glacier mass changes and their contributions to sea-level rise from 1961 to 2016. *Nature*, 568, 382–386.
- Zephyris (2011). *Leaf Tissue Structure* [Afbeelding]. Aangepast. CC BY-SA 3.0. Geraadpleegd op 3 maart 2019 via https://commons.wikimedia.org/w/index.php?title=File:Leaf_Tissue_Structure.svg&oldid=333913566.

INDEX

- accuraatheid, 219
accuracy, 251, 254, 258, 265, 268
activatiefunctie, 172, 174, 191, 194, 196, 200, 211, 224, 226, 230, 237, 239, 251, 252, 256, 267
Ada Lovelace, 94, 95
AI-winter, 98, 99, 106, 117
Alan Turing, 94, 104
albedo, 19, 55, 56, 278
amphistomateus, 37
annoteren, 65, 72, 75, 76, 121
antropoceen, 27
artificiële intelligentie, 81–84, 86, 87, 109
artificieel neuron, 171

batch, 219, 220, 254, 258
BCE, 203, 204, 206, 216, 218, 230, 251
bias, 112, 117, 136, 169, 175, 186, 223
big data, 88, 89, 91, 92, 102
binaire talstelsel, 149, 150
binary cross-entropy, 203, 204, 206, 216, 218, 230, 251
biodiversiteit, 30, 51–53, 61, 69, 278, 279
biologisch neuron, 170
bit, 149, 150
broeikaseneffect, 19–21, 27, 54, 55, 277, 278
broeikasgas, 20, 21, 27, 33, 55, 56, 85, 278
burgerwetenschap, 65, 76–79
byte, 149, 150
China, 82, 115
citizen science, 65, 76–79
classificatie, 119, 120, 130, 131, 184, 187, 201, 203, 206, 207, 209, 216, 230, 242, 247, 251, 255, 256, 260, 265, 267, 270
classificeren, 112, 122, 179, 181, 182, 184, 191, 193, 200, 202, 204, 219, 221
convolutie, 231–234, 236, 237, 239, 244–247, 267
convolutioneel, 124, 126, 231, 237–239, 241, 247, 250, 251, 266, 267, 270, 271
correlatiecoëfficiënt, 129
cuticula, 35, 37, 135
datagebaseerd, 82, 109, 117
Daubechies, 124, 127
decision boundary, 270
deep learning, 106, 109, 123, 124, 255
dense layers, 229, 238, 239, 241, 247, 252, 256, 266, 267
dicotyl, 37, 39
diep neurale netwerk, 65, 106, 123, 124, 131, 239, 247, 252, 266, 270, 275
digitale afbeelding, 146
drempelwaarde, 174, 202, 207, 223, 225, 228, 273–275
dropout, 271, 273
elektromagnetisch, 147
epistomateus, 37
epoch, 122, 174, 177, 178, 187, 194, 213, 219, 220, 251, 254, 257, 260, 261, 265, 268
ethiek, 110
ethisch, 87, 108, 113, 279, 280

feature map, 237–239
feedforward, 241, 271
feedforward netwerk, 125, 126, 251, 252
fit(), 254, 258
Flatten(), 238, 267
fotosynthese, 18, 35, 36, 43, 45, 46, 51, 53, 56, 57, 59, 60, 73, 278–280
fully connected layers, 229

gemiddelde kwadratische afwijking, 194–196, 199, 212, 215, 230
general AI, 82, 169
generalisatie, 122, 220
generaliseren, 121, 124, 131, 220
geschiedenis AI, 94

- gewichten, 125, 173–177, 179, 184, 186–190, 192, 194, 195, 203, 208, 211, 213, 219, 220, 249, 251, 260, 275
- gletsjer, 13, 19, 23–25, 31, 56, 281
- gradient descent, 195, 207–209, 211, 213, 215, 219, 220, 230, 251
- grijswaardenafbeelding, 146
- haltervormig, 39, 42, 70
- Heaviside, 160, 171, 172, 174, 196, 223, 225, 228, 230, 281
- hidden layer, 125, 131, 207, 226, 229, 230, 252, 260, 268, 270
- histogram, 150, 245
- history, 254
- huidmondje, 14, 35–37, 39–41, 64, 65, 70–72, 75, 112, 134, 135, 241, 270, 271, 273
- hyperparameter, 174, 251
- hypostomateus, 37, 70
- ijskap, 23–26, 55, 56
- ijskern, 13, 14, 16, 17, 20, 28, 31, 277
- intelligentie, 83, 107
- IPCC, 21, 23, 27
- Iris dataset, 183–185, 187, 194, 270
- Keeling, 16
- Keelingcurve, 16, 18
- kennisgebaseerd, 82, 109, 117
- kleine ijstijd, 18, 19, 31, 278
- kleinste kwadraten, 196
- kleurenafbeelding, 146, 148, 151
- kleurenfoto, 166
- klimateopwarming, 16, 23–27, 29, 33, 34, 44, 55–57, 60
- klimateverandering, 19, 21, 27, 44, 52, 53, 60, 63, 73, 277, 278
- koolstofcyclus, 54, 58, 59
- kost, 195, 207, 219
- kostenfunctie, 195, 203, 204, 206, 208, 209, 211–214, 216, 219, 230, 249–251, 253, 275
- laag, 124–126, 131, 211, 229, 241, 249, 251
- label, 124, 126, 173, 174, 176, 178, 185, 187, 188, 194, 200, 204, 207, 220, 230
- layer, 124–126, 131, 211, 229, 241, 249, 251
- learning rate, 174, 210, 211, 220, 230, 253, 257, 261, 265, 275
- leersnelheid, 174, 210, 211, 220, 230, 253, 257, 261, 265, 275
- leren, 119, 121, 122, 124, 126, 131, 172–174, 176, 187, 190, 211, 220, 230, 250, 275
- lerend algoritme, 119
- likelihood, 203–205
- lineaire regressie, 194, 196, 199, 204
- loss, 203, 206, 208, 211–214, 216, 219, 230, 249–251, 253, 254, 258, 265, 275
- machinaal leren, 83, 88, 89, 93, 109, 117, 123, 130, 131, 169, 185, 194
- machine learning, 83, 88, 89, 93, 109, 117, 123, 130, 131, 169, 185, 194
- matrix, 149, 150, 153, 154, 156, 163, 166, 167
- max pooling, 241–243, 246, 267, 271
- metrics, 253, 254
- microfoto, 37, 39, 65, 69, 71, 75, 79
- microscopie, 134
- misconcepties, 169
- MNIST, 163, 164, 235, 246, 255, 256, 260, 266
- module, 140
- monocotyl, 37, 39
- MSE, 195, 230, 251
- multiklassen, 251, 271
- nagellak, 69, 70, 135
- narrow AI, 82
- netwerkarchitectuur, 250–254, 256, 265, 267, 275
- neuron, 125, 131, 170–172
- niervormig, 39, 42, 70
- NIPT, 114
- normaliseren, 130, 131, 186, 251, 257, 271
- normalisering, 130
- notebook, 139, 140
- object-georiënteerd, 140
- omgeving, 13, 36, 37, 39, 40, 42, 44, 52, 84
- omgevingsfactor, 39, 41
- omgevingsfactoren, 49, 50, 279
- one hot encoding, 254, 271
- optimalisatiefunctie, 249–251, 253, 275
- optimizer, 249, 251, 253, 275
- opwarming, 16, 23–27, 29, 33, 34, 44, 55–57, 60
- overfitting, 122, 123, 126, 131, 220, 221, 230, 260, 261, 273

- P vs NP, 99, 106
 paleoklimatologie, 13, 15, 33, 277
 paleoklimatoloog, 13
 paleontologisch, 39
 Perceptron, 106, 172, 174, 191, 200, 219
 Perceptron-algoritme, 182, 188, 200
 pixel, 145, 146, 151
 Plantentuin Meise, 63–67, 69, 71, 73
 proxydata, 13, 15, 277
 radiative forcing, 27, 54–56
 regressie, 119, 120, 130, 131, 194, 196, 199, 208, 212, 215, 219, 230, 251, 281, 282
 regressielijn, 194, 196, 197
 reinforcement learning, 82, 121, 169
 rekenkracht, 85, 98, 99, 103, 106, 108, 220
 ReLU, 196, 224–226, 241–246, 267, 271, 281
 representatie, 120, 237, 239, 267
 representeren, 120, 129, 149, 253
 reshape(), 238, 257
 respiratie, 36, 46, 60
 RGB, 146, 148, 151, 166
 scalair, 158
 Sequential model, 241, 252, 253, 256, 266, 275
 sigmoïde, 196, 200, 202, 230, 261, 265
 sign, 160, 196, 226, 228, 281
 singulariteit, 103, 108
 soorten AI, 82, 109, 117
 standaardafwijking, 199
 standaardiseren, 129–131, 185–187, 199, 251
 stochastic gradient descent, 219, 220, 230, 251, 253, 258
 stoma, 14, 35–37, 39–41, 64, 65, 70–72, 75, 112, 134, 135, 241, 270, 271, 273
 stomatadetectie, 75, 136, 253, 273
 stomatale dichtheid, 41, 43, 47, 49, 50, 61, 70, 73, 207
 stomatale geleidbaarheid, 36, 42–44, 46, 47, 51, 53
 Stradivarius, 19
 supervised learning, 121, 126
 tensor, 148, 150, 158, 159, 161, 163, 166, 167
 TensorFlow, 250
 testdata, 70, 122, 131, 219, 221, 251, 275
 testset, 70, 122, 131, 219, 221, 251, 275
 threshold, 202, 207, 223, 225, 228, 251, 253, 268, 273–275
 trainingdata, 69, 70, 72, 122, 131, 219, 220, 251, 275
 trainingset, 69, 70, 72, 122, 131, 219, 220, 251, 275
 transpiratie, 35, 36, 43, 51, 52
 transpireren, 35, 52
 transponeren, 154, 157
 Turingtest, 94, 104
 UGent, 63, 64, 73
 uint8, 149
 underfitting, 122, 123, 131, 220, 230, 258, 260
 unsupervised learning, 121
 valideringsdata, 122, 131, 220, 221, 251, 275
 valideringsset, 122, 131, 220, 221, 251, 254, 275
 vector, 158, 161, 188, 211, 254
 verborgen laag, 125, 131, 207, 226, 229, 230, 252, 260, 268, 270
 voedselvoorziening, 53, 61, 278, 279
 waterhuishouding, 43, 47, 51, 61, 73
 Waze, 92, 114
 weights, 125, 173–177, 179, 184, 186–190, 192, 194, 195, 203, 208, 211, 213, 219, 220, 249, 251, 260, 275
 XOR, 97, 106, 223
 zichtveld, 70

KIKS is een STEM-project rond artificiële intelligentie voor de derde graad secundair onderwijs. Leerlingen leren AI begrijpen, met mogelijkheden en beperkingen; ze leren hoe ze er een impact kunnen op hebben.

De relatie tussen huidmondjes (stomata) van planten en de klimaatverandering biedt een uniek kader om met diepe neurale netwerken aan de slag te gaan. De programmeertaal Python is bovendien zeer toegankelijk als tool om de fundamenteën van neurale netwerken te bestuderen.

Een troef van het KIKS-project is de intensieve samenwerking tussen onderzoekers en leerkrachten. Het lesmateriaal van KIKS wordt immers ontwikkeld, parallel met de resultaten van een lopend wetenschappelijk onderzoek aan de UGent en de Plantentuin Meise.

