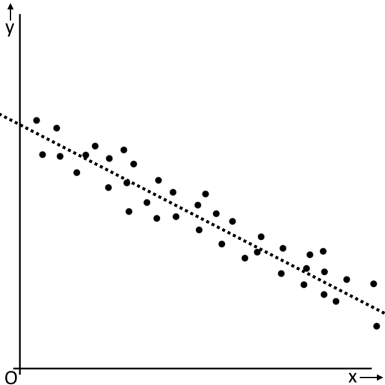
# Deelvraag Inleiding

In de vorige deelvraag hebben we behandeld wat een zelflerend systeem is. In deze deelvraag gaan wij dieper in op de verschillende soorten zelflerende algoritmes. We zullen in deze deelvraag naar drie verschillende algoritmes kijken: *Linear Regression*, *Support vector machines* en *Artificial Neural Networks*. [1]

# Linear Regression

Het eerste machine learning algoritme dat we gaan behandelen is linear regression. Dit algoritme wordt gebruikt voor het voorspellen van een y-waarde bij gegeven x-waarde(n). Om linear regression te kunnen gebruiken is het belangrijk dat er wel een linear verband bestaat tussen de x-waarde(n) en de y-waarde. In figuur \*\*\*\* is een dergelijk linear verband te zien.   
Dit lineare verband is te beschrijven met de formule:  
y = ax + b  
  
Het doel bij linear regression is het bepalen van de waarde voor a en b. Dit is op verschillende manieren mogelijk. Een statistische manier hiervoor is door gebruik te maken van het **ordinary least squares** algoritme. Dit algoritme bepaald de best passende lijn door de punten, ook wel bekend als de trendlijn. De waarden voor a en b worden hierbij als volgt bepaald:

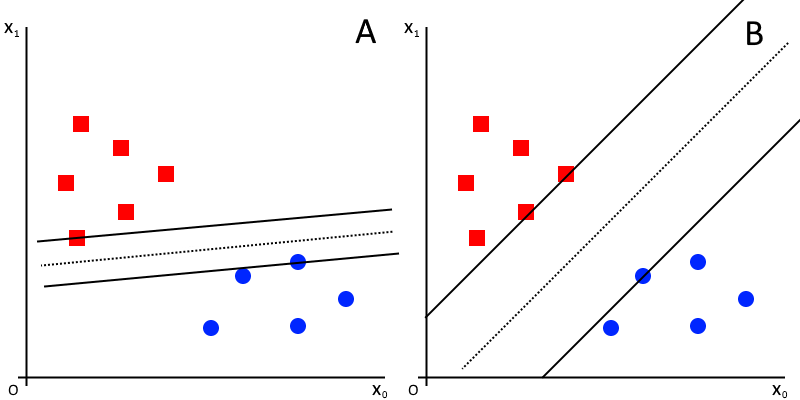
Figuur

In deze formules is het gemiddelde van alle x waarden en het gemiddelde van alle y waarden.

Dit algoritme echter alleen toepasbaar als er sprake is van één x-waarde als invoer. Bij meerdere x-waarden is het dus niet te gebruiken. Een andere manier om de a en b waarde te vinden is door gebruik te maken van een leerstrategie. In deelvraag 4 bespreken we drie verschillende leer strategieën.

# Support vector machine

Een support vector machine (SVM) is een machine learning algoritme ontwikkeld door vladimir vapnik. Het algoritme kan gebruikt worden voor het classificeren van data. Het algoritme heeft training data nodig en is dus een vorm van supervised learning. [2]



Figuur Support vector machine

Een support vector machine werkt als volgt: Het trekt een lijn, een **vector**, tussen de twee groepen. Deze vector is zo dat *de afstand tussen de vector en het dichtstbijzijnde datapunten zo groot mogelijk is* [3]Deze dichtstbijzijnde datapunten worden de **support vectoren** genoemd In figuur \*\*\* is twee keer dezelfde dataset weergegeven. In de linker afbeelding is te zien dat de vector de twee groepen scheidt maar de afstand tussen het dichtstbijzijnde datapunt kleiner is dan bij de rechter afbeelding, deze afstand wordt de marge genoemd. De in de rechter afbeelding is de marge het grootst, dus dit is de betere vector. Het gebied tussen de twee support vectoren wordt het **hyperplane** genoemd.

## Het algoritme

Het doel van het algoritme is van een nieuw datapunt bepalen of het tot groep A (de rode vierkantjes) of groep B (de blauwe cirkels) behoort. Als een nieuw datapunt behoort tot groep A dan willen we dat de output van het algoritme negatief is en als het nieuwe datapunt behoort tot groep B willen we dat de output positief is. Hoe positiever of negatiever de output is hoe zekerder het is dat dit punt daadwerkelijk tot die groep behoort. Als de output 0 is, dan bevindt het punt zich precies tussen de twee groepen, het ligt dan op de stippellijn van figuur \*\*\*\*. Verder is het zo dat de output tussen -1 en 1 ligt als het binnen de twee support vectoren ligt. In dit gebied is het niet helemaal zeker tot welke groep het punt behoort.

We kunnen de drie vectoren als volgt definiëren:  
De linker support vector   
De middelste vector   
De rechter support vector   
Een support vector machine probeert het volgende:

* Alle datapunten moeten buiten de twee support vectoren liggen
* De afstand tussen de support vectoren moet zo groot mogelijk zijn

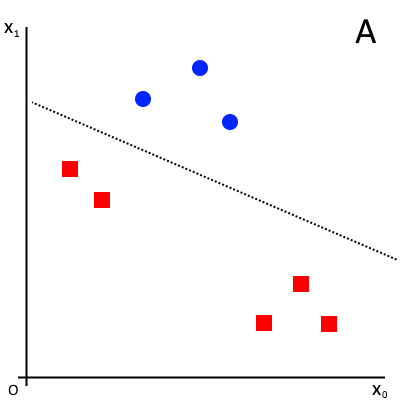
Vanuit de vectoren zijn de volgende twee formules af te leiden:

De formule voor of een datapunt buiten de twee support vectoren ligt:   
De formule voor de afstand tussen de twee support vectoren:   
  
Een vector die voldoet aan de volgende eisen wordt gekozen:

1. Voor alle datapunten moet gelden:
2. moet zo klein mogelijk zijn, ofwel zo groot mogelijk

## Kernel Methods

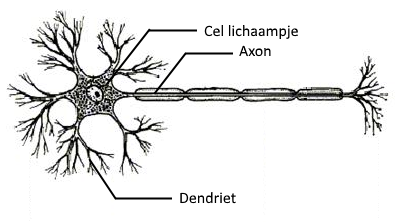
In veel gevallen zal de dataset niet zo mooi geordend zijn als in figuur \*\*\*. Het is dan niet mogelijk om een rechte lijn te trekken die de twee groepen scheidt. Een support vector machine zou in dit geval dus niet werken. Om toch een support vector machine te kunnen gebruiken is er iets genaamd de Kernel Trick.

In figuur \*\*\*\* is een één dimensionale dataset te zien, dit wil zeggen, er is maar één variabele. Met een support vector machine is het nu niet mogelijk om een lijn te trekken die de twee groepen scheidt. Daarom wordt er een extra variabele bij gemaakt, bijvoorbeeld X1 = (X0)2. Nu is het wel mogelijk een lijn te trekken door de dataset die de twee groepen opdeelt:

# Artificial Neural Networks

## Biologisch en kunstmatig netwerk

Binnen mensen wordt informatie overgebracht door middel van het zenuwstelsel. Dit zenuwstelsel is opgebouwd uit miljarden zenuwcellen. Een zenuwcel, ook wel een neuron genoemd, is opgebouwd uit drie delen: een cel lichaampje, een aantal dendrieten en één axon. In bron \*\*\*\* is een weergave van een biologische neuron te zien.



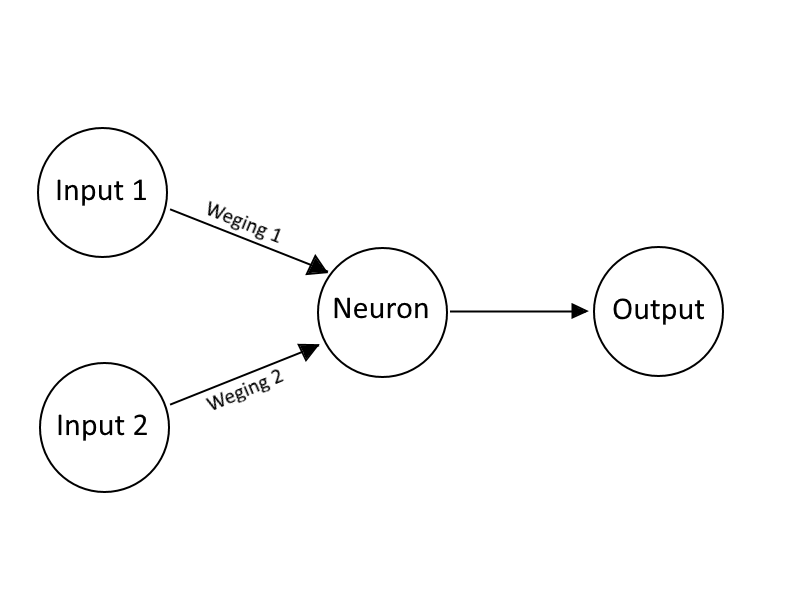
Figuur \*\*\*\*: een tekening van een biologisch neuron

In de biologie zijn dendrieten verantwoordelijk voor de instroom van informatie, zij brengen informatie (impulsen) naar het cel lichaampje toe. De zenuwcel kan deze informatie vervolgens via een enkele axon doorgeven aan een dendriet van een andere zenuwcel of aan een spier. Het doorgeven van informatie gebeurt in het uiteinde van de axon en dendrieten, in zogeheten synapsen.

Het principe van een neuron kan ook door een computer uitgevoerd worden. Dit is het idee voor een Artificial Neural Network (ANN). Een dergelijk netwerk bestaat uit een verschillend aantal ‘computerneuronen’. Elk van deze neuronen krijgt, net zoals een biologische neuron, informatie binnen. Binnen de neuron vindt een berekening plaats. Vervolgens wordt deze waarde doorgegeven of is dit de output.

## De perceptron

De simpelste vorm van een neural network is een netwerk met slechts één neuron. Zo’n ANN, voor het eerst gemaakt door F. Rosenblatt in 1958 [1], wordt een **perceptron** genoemd.

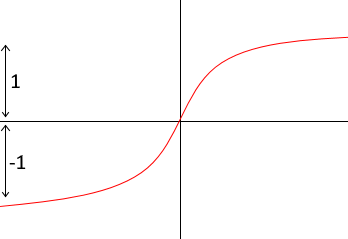


Figuur \*\*\*\*: een schematische weergave van een perceptron

In figuur \*\*\*\* is te zien dat een neuron twee inputs binnen krijgt en daarna een output geeft. De pijlen naar de neuron toe en er vanaf stellen de synapsen voor. Elke synaps heeft een bepaalde weging. De weging van een synaps bepaald hoeveel invloed die ene input heeft op het netwerk. Het uiteindelijke doel van een neural network is *het zoeken naar de optimale weging voor alle synapsen binnen het netwerk*.   
Om tot een output te kunnen komen moet de neuron een berekening uitvoeren. In deze situatie is de berekening nog vrij eenvoudig:

## Activation function

De waarde die uit deze berekening volgt, wordt door een **activation function** gehaald. Een activation function zorgt ervoor dat aan deze som een waarde kan worden gehangen, bijvoorbeeld 1 of -1, zonder dat de som absoluut deze waarde heeft. Dit wordt gedaan door te kijken waar het punt op de grafiek van deze functie zich bevind.



Figuur \*\*\*\*: Een voorbeeld van een algemene activation function en welke waardes hieraan worden gekoppeld.

In figuur \*\*\*\* is een grafiek van een activation function gegeven. In dit voorbeeld worden aan alle positieve y waardes een 1 verbonden en aan alle negatieve een -1.

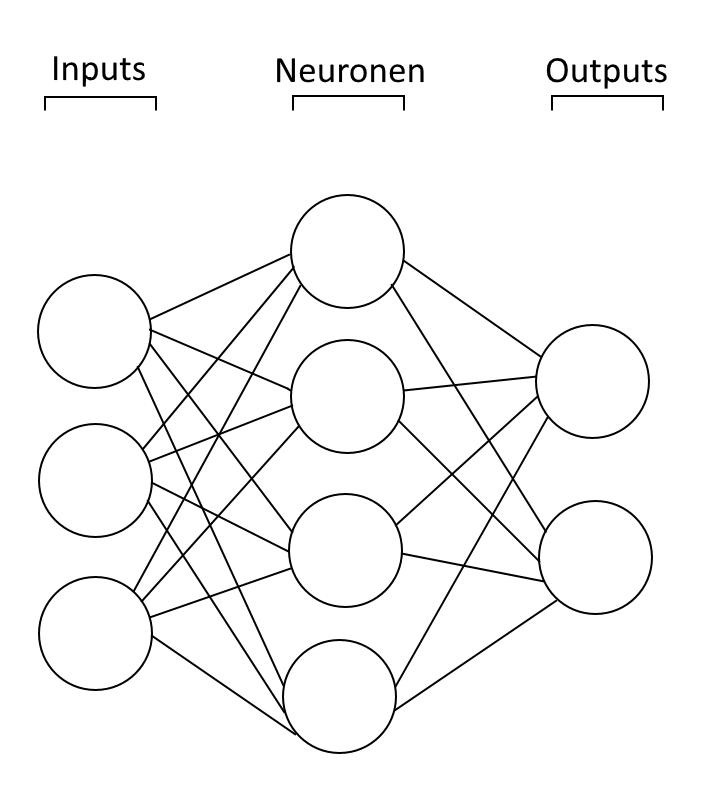
Een ANN is een vorm van supervised learning. Het programma weet dus wat het antwoord zou moeten worden. Als de output correct is zal er weinig gebeuren, maar als de output incorrect is zal het programma zichzelf moeten aanpassen om wel de goede uitkomst te krijgen. Dit gebeurt met behulp van de wegingen van elke synaps. Deze wegingen kunnen namelijk worden aangepast. De invloed van elke input kan ofwel vergroot ofwel verkleint worden. Op deze manier zal uit de berekening in de neuron de volgende keer misschien een andere, betere uitkomst komen.  
De nieuwe weging van een synaps wordt nu: .  
Hoe berekend wordt, wordt bepaald door de leerstrategie van het systeem. Hier wordt in deelvraag \*\*\*\* verder op in gegaan.

## Bias

Met de besproken perceptron is echter één probleem. Wanneer beide inputs gelijk zijn aan 0, heeft het aanpassen van wegingen geeneffect. Een weging maal 0 zal immers altijd in 0 resulteren. Om dit probleem tegen te gaan, wordt er een **bias** toegevoegd. Dit is een extra input die standaard gelijk is aan 1. De weging van de synaps van de bias wordt niet veranderd. Omdat de neuron nu ook bij inputs van 0 een andere uitkomst uit de berekening zal geven, zal er nu toch een getal door de activation function gaan en zullen de wegingen toch worden aangepast.

## Een netwerk van perceptrons

Natuurlijk is het ook mogelijk van niet één, maar meerdere perceptrons te hebben. Zo wordt het een echt netwerk van synapsen en neuronen.



Figuur \*\*\*\*: een schematische weergave van een willekeurig ANN

Ook is het mogelijk meerdere lagen neuronen te hebben. Dit wordt een **deep neural network** genoemd, of simpelweg **deep learning**.

De tot nu toe besproken ANN’s hebben hun informatie allemaal in één richting bewogen: alle inputs, alle neuronen, alle outputs. Dit wordt **feedforward** genoemd. Ook zou je een neural network kunnen hebben waarin de informatie ook nog tussen de neuronen in dezelfde laag beweegt: een **recurrent neural network**.

# Bronnen

[1] <https://www.analyticsvidhya.com/blog/2015/08/common-machine-learning-algorithms/>   
Geraadpleegd op: 3-6-2017  
Laatst gewijzigd op: August 10, 2015  
Auteur: Sunil Ray

[2] <https://books.google.nl/books?hl=nl&lr=&id=HUnqnrpYt4IC&oi=fnd&pg=PP7&dq=support+vector+machines&ots=g8lIEB0rSi&sig=FTLWxhxAwcf95E1xLoWZ8WYFZ4k#v=onepage&q=support%20vector%20machines&f=false>   
Titel: Support Vector Machines

Door Ingo Steinwart, Andreas Christmann

[3] <http://www.saedsayad.com/support_vector_machine.html>   
Geraadpleegd op: 4-6-2017  
Auteur:

[4] <https://www.svm-tutorial.com/2015/06/svm-understanding-math-part-3/>   
Geraadpleegd op: 4-6-2017  
Auteur: Alexandre KOWALCZYK

ANN

Link: <http://biologiepagina.nl/Vwo5/Zenuwstelsel/inleiding.html>   
Geraadpleegd op: 9 – 7 – 2017   
Laatst gewijzigd op: niet bekend

Link: <https://www.evolvingsciences.com/Neuron%20.html> (Afbeelding)  
Geraadpleegd op: 9 – 7 – 2017   
Laatst gewijzigd op: niet bekend

[1]Link: <http://psycnet.apa.org/journals/rev/65/6/386/>   
Geraadpleegd op: 9 – 7 – 2017   
Laatst gewijzigd op: november 1958