Limitaties van zelflerende systemen

# Inleiding

In de vorige deelvraag is te lezen waar zelflerende systemen allemaal voor gebruikt kunnen worden. Toch zijn zelflerende systemen niet altijd toepasbaar. We gaan in dit hoofdstuk de vraag beantwoorden: *Welke factoren zorgen ervoor dat zelflerende systemen in de praktijk niet altijd toepasbaar zijn?* In deze deelvraag behandelen dingen die het gebruik van machine learning limiteren.

# Training Data

In veel gevallen heeft een zelflerend systeem training data nodig om beter te kunnen presteren. Een zelflerend systeem moet een bepaald niveau bereiken voordat het in de praktijk kan worden toegepast. Denk hierbij bijvoorbeeld aan een op machine learning gebaseerde zelfrijdende auto. Deze moet een bepaalde afstand kunnen rijden zonder gevaarlijke situaties te veroorzaken. Om dit bepaalde niveau te kunnen bereiken is er veel training data nodig waarmee het systeem verbeterd kan worden. Deze training data is niet altijd voldoende en in goede kwaliteit beschikbaar. De training data moet gelijkwaardig zijn aan de data die het zelflerende systeem in de praktijk krijgt. Hoe complexer de data is, hoe meer data er ook nodig is om een goed niveau te bereiken. Om bijvoorbeeld *image classifiction* te kunnen uitvoeren, zijn er miljoenen voorbeeld plaatjes, met de goede classificatie, nodig om het algoritme te trainen. Als men niet over die data beschikt kan het algoritme niet verbeteren.

## Semi-supervised learning

Semi-supervised learning is een techniek die het bovengenoemde probleem beperkt. Er wordt gebruik gemaakt van twee groepen data, een grote *unlabeled*dataset en een kleine *labeled* dataset. De *labeled* data zal vaak door een mens van een label moeten worden voorzien en is daardoor moeilijker te verkrijgen terwijl er vaak genoeg *unlabeled* data is. [4]

# Groot

Een andere limitatie die men vaak tegenkomt is die van de computersnelheid. Voor complexe taken zijn grotere zelflerende systemen nodig. Op een gegeven moment loop je tegen de limieten van de computer aan. Een complexe taak als het spelen van het spel Go vereist enorm veel computer capaciteit. Google’s Deep Mind project gebruikte hiervoor 1,202 CPUs and 176 GPUs. [1] Voor iemand die niet evenveel computer capaciteit al Google heeft zou dit dus onmogelijk zijn geweest. De capaciteit van de computer limiteert de haalbaarheid van bepaalde doelen enorm. Het enige wat mogelijk is hieraan te doen is het verbeteren van de computers en het slimmer schrijven van het zelflerende systeem.

# Specifiek

Er is nog een limitatie die die het gebruik van machine learning belemmerd: Een systeem is specifiek getraind voor een bepaalde taak. Als het algoritme getraind voor een specifiek doel kan het niet zomaar een ander doel krijgen. Als er bijvoorbeeld een zelflerend systeem is getrained op het spelen van schaken, zal het andere spellen niet ook kunnen spelen. Het is specifiek getraind voor die taak. Er wordt in de machine learning gestreefd naar het creëren van een *general intelligence*, ofwel een AI die meerdere taken kan vervullen.

## Transfer Learning

Transfer Learning is het toepassen van de kennis van het zelflerende systeem van één probleem op een ander probleem. Dit is erg goed toepasbaar met image classification. Het algoritme moet hierbij namelijk eerst leren hoe een plaatje in elkaar zit en kan daarna pas specifieke plaatjes sorteren. Door alleen een bepaald deel van het zelflerende systeem opnieuw te trainen hoef je niet het hele systeem opnieuw te laten leren, maar alleen het nuttige deel. [2]

# Conclusie

Er zijn limieten die het gebruik van zelflerende systemen in de praktijk beperken. Hoewel er veel wordt gezocht

# Bronnen

[1] <http://uk.businessinsider.com/heres-how-much-computing-power-google-deepmind-needed-to-beat-lee-sedol-2016-3?international=true&r=UK&IR=T>

[2] <http://cpms.byu.edu/springresearch/abstract-entry?id=861>  
[3] <https://www.slideshare.net/JisungDavidKim/oneshot-learning>

[4] https://www.cs.utah.edu/~piyush/teaching/8-11-slides.pdf

https://thenewstack.io/what-machine-learning-can-and-cant-do/