

Self-learning AI in Games

Abe Vos

April 2022

Introductie

Good Old-Fashioned AI

- ▶ Symbolische/klassieke AI
- ▶ Kennis van de wereld is uitgedrukt in simpele regels
- ▶ AI combineert regels om nieuwe conclusies te trekken

Voorbeeld van regels

- ▶ Een muur is een obstakel
- ▶ Een gesloten deur is een obstakel
- ▶ Een open deur is geen obstakel
- ▶ Een gesloten deur kan worden geopend met een deurklink
- ▶ Een open deur biedt de doorgang naar de volgende kamer

Representatie

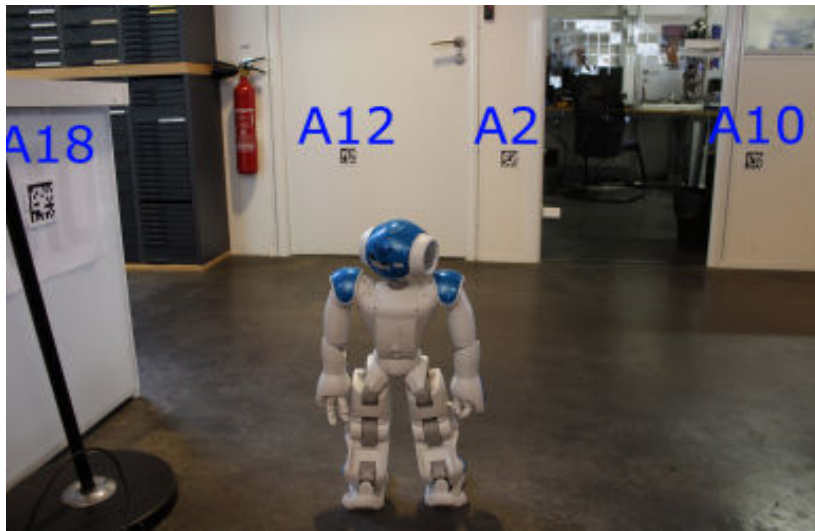


Figure 1: Navigatie met QR codes

GOFAI in games

- ▶ Pathfinding
 - ▶ Dijkstra's algoritme
 - ▶ A*
- ▶ Planning
 - ▶ GOAP
 - ▶ Decision trees
- ▶ Bruteforce
 - ▶ Minimax
 - ▶ Deep Blue

Pathfinding (A*)

Pathfinding (A*)

Planning (GOAP)

- ▶ Goal-Oriented Action Planning
- ▶ Representeer de omgeving als een reeks variabelen
- ▶ Gebruik acties om de omgevingsstaat te veranderen

Minimax

- ▶ Tegenstander probeert eigen score te maximaliseren
- ▶ Kies een actie die tegenstander's score minimaliseert
- ▶ Zero-sum: jouw winst is mijn verlies
- ▶ Representatie is de score en mogelijke acties

Tekortkomingen

- ▶ Representatie van de wereld
- ▶ Sommige methodes hebben handgeschreven regels nodig
- ▶ Niet flexibel

AI Winter

- ▶ Belofte van menselijke intelligentie
- ▶ AI bleek niet te generaliseren
- ▶ Investerings stopten

Machine Learning

Een alternatief voor GOFAI

- ▶ Leer nuttige representaties van ervaringen
- ▶ Geen kennis van achterliggend proces
- ▶ Geen wondermiddel
 - ▶ Nieuwe oplossingen
 - ▶ Ook nieuwe problemen

Hoe leren machines?

- ▶ Computerwetenschap
- ▶ Statistiek
- ▶ Biologie

Biologische inspiratie

- ▶ Hersenen kunnen leren van ervaringen/voorbeelden
- ▶ Niet bekend hoe dit werkt
- ▶ Bouw een brein en kijk hoe ver je komt

Kunstmatig brein

- ▶ Brein is opgebouwd uit neuronen
- ▶ Maak een kunstmatig neuron en bouw een kunstmatig brein

Neuron

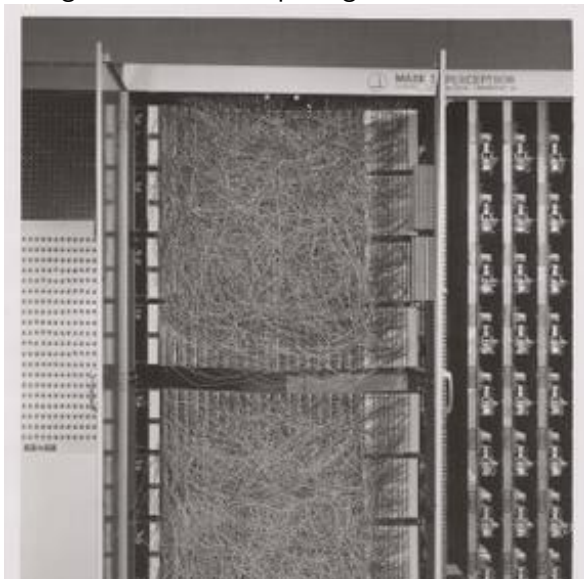
Benodigheden kunstmatig neuron

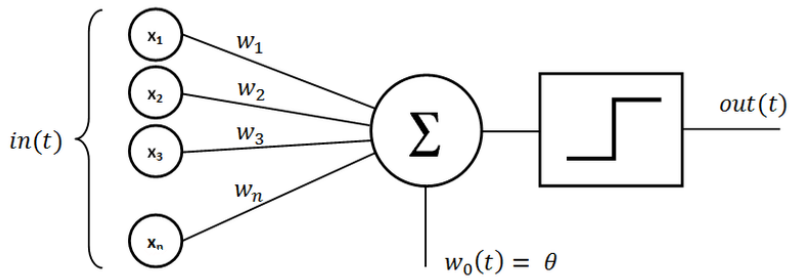
- ▶ Ontvang signalen van verschillende sterktes
- ▶ Combineer signalen en vergelijk met drempelwaarde
- ▶ Pas signaalsterktes aan om te “leren”

Supervised learning

Perceptron

- ▶ Afbeeldingen van 20x20 pixels
- ▶ Elke pixel (waarde tussen 0 en 1) is een signaal
- ▶ Output signaal is een voorspelling





Afbeelding als signalen

- ▶ Elke pixel een waarde tussen 0 (zwart) en 1 (wit)
- ▶ Kleurenafbeelding als drie aparte zwart wit afbeeldingen (drie RGB kanalen)

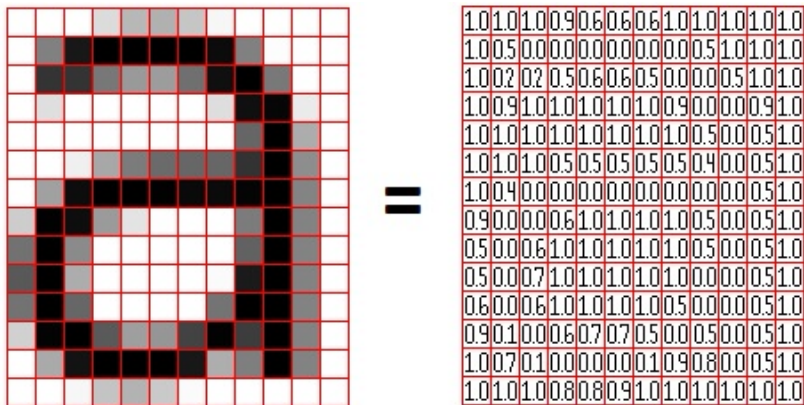


Figure 3: Een afbeelding als een reeks getallen

Activatiefunctie

- ▶ Simuleert de drempelwaarde
- ▶ Logistieke functie $\sigma(x) = \frac{1}{1+e^{-x}}$
- ▶ Waarde hoger dan 0.5 \rightarrow 1 anders 0

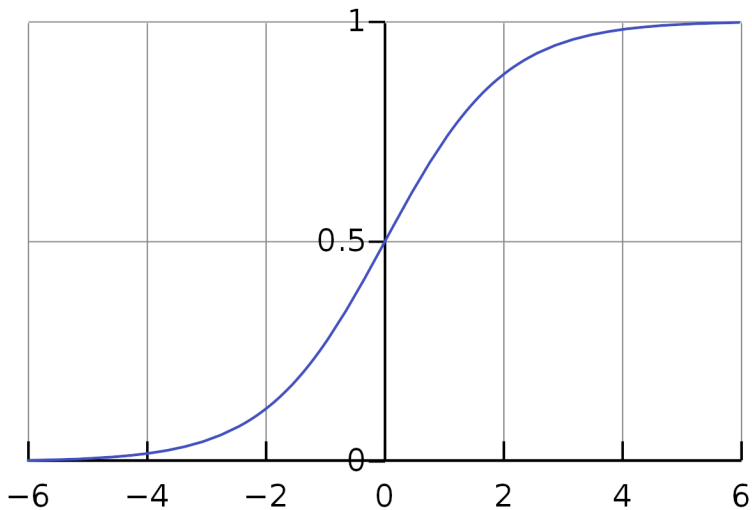


Figure 4: De logistieke functie

Leren

- ▶ Data $\mathbf{x} = x_1, x_2, \dots, x_N$ en $\mathbf{y} = y_1, y_2, \dots, y_N$
- ▶ Parameters $\mathbf{w} = w_1, w_2, \dots, w_D$
- ▶ Error/loss functie $L_{\mathbf{w}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N} \sum_{n=1}^N (h_{\mathbf{w}}(x_n) - y_n)^2$
- ▶ Pas parameters aan om error zo klein mogelijk te maken

Eenvoudiger voorbeeld

- ▶ Voorspel het weer met sensoren
- ▶ Input signalen: temperatuur (graden celsius), windsnelheid (km/h), luchtvochtigheid (%)
- ▶ Output/voorspellingen: zonnig (1), regenachtig (0)

Een datapunt

- ▶ $x_n = [12.3, 34.8, 75.1]$
- ▶ $y_n = 0$
- ▶ $\mathbf{w} = [0.1, 0.04, -0.03]$
- ▶ $h_{\mathbf{w}}(x_n) = \sigma(0.1 \cdot 12.3 + 0.04 \cdot 34.8 - 0.03 \cdot 75.1)$
 - ▶ ≈ 0.591
- ▶ $l_n = (0.591 - 0)^2 \approx 0.350$
- ▶ Herhaal voor alle datapunten en bereken gemiddelde error

Random search

- ▶ Begin met willekeurige parameters $\mathbf{w}^* = \mathbf{w}$
- ▶ Bereken error $L^* = L_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$ voor de dataset
- ▶ Genereer nieuwe parameters \mathbf{w}' met bijbehorende L'
- ▶ Als $L' < L^*$ update: $\mathbf{w}^* \leftarrow \mathbf{w}'$ en $L^* \leftarrow L'$
- ▶ Herhaal vorige twee stappen tot gewenst resultaat

Random search

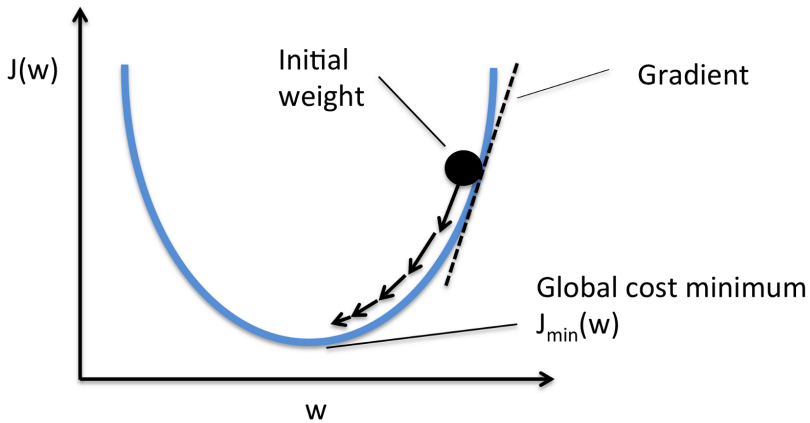
- ▶ Erg traag; hoge variantie
- ▶ Alle mogelijke waarden van \mathbf{w} moeten uitgeprobeerd worden
- ▶ 2^{64} waarden voor elk element van \mathbf{w}
- ▶ Voor drie dimensies is dat $2^{192} \approx 6.28 \cdot 10^{57}$
- ▶ Afbeeldingen voor de perceptron hebben 400 dimensies

Optimalisatie

- ▶ Parameters met goede performance liggen dicht bij elkaar
- ▶ Maak gebruik van de 'vorm' van de error functie
- ▶ Volg de helling naar een dieptepunt

Gradient Descent

- ▶ Gebruik de afgeleide van de functie $L_{\mathbf{w}}(\mathbf{x}, \mathbf{y})$
- ▶ Hoe verandert de error $L_{\mathbf{w}}$ als we parameters \mathbf{w} veranderen?
- ▶ Afgeleide/gradient

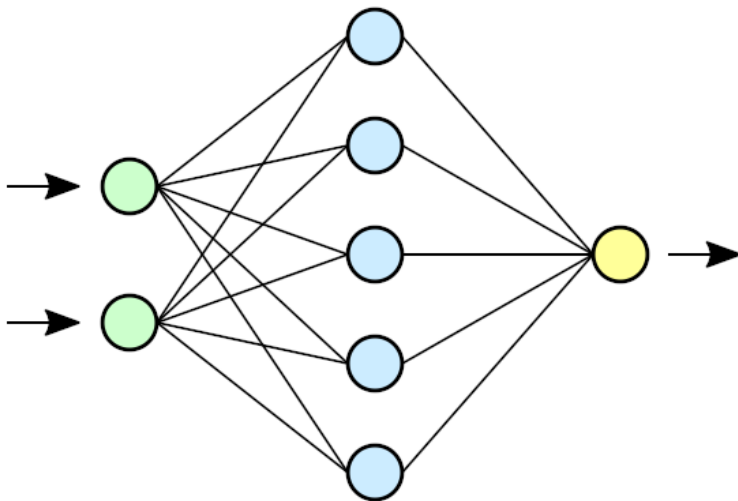


Deep Learning

- ▶ Netwerk van neuronen vormt een brein
- ▶ Netwerk van Perceptrons vormt een Multi-Layered Perceptron
- ▶ (Deep) Neural Network

Neuraal Netwerk

- ▶ Twee input neuronen
- ▶ Vijf verborgen (hidden) neuronen
- ▶ Een output neuron



Neuraal Netwerk

- ▶ Trainen met Gradient Descent en Backpropagation
- ▶ Bereken de uitkomst van het netwerk (forward propagation)
- ▶ Herleid de afgeleiden van de parameters laag voor laag terug van de output (back propagation)

Diep Neuraal Network

- ▶ Kan alles leren met genoeg* verborgen neuronen in 1 laag
- ▶ Plak meerdere verborgen lagen achter elkaar
- ▶ Diepere netwerken zijn efficiënter en complexer dan brede netwerken

Deep Learning Renaissance

- ▶ Backpropagation was uitgevonden in 1986
- ▶ Deep learning werd populair in 2012
- ▶ Tekort aan data en rekenkracht

Deep Learning Renaissance

- ▶ Data tekort → internet (social media)
- ▶ Rekenkracht → videokaarten



Figure 5: Object classification

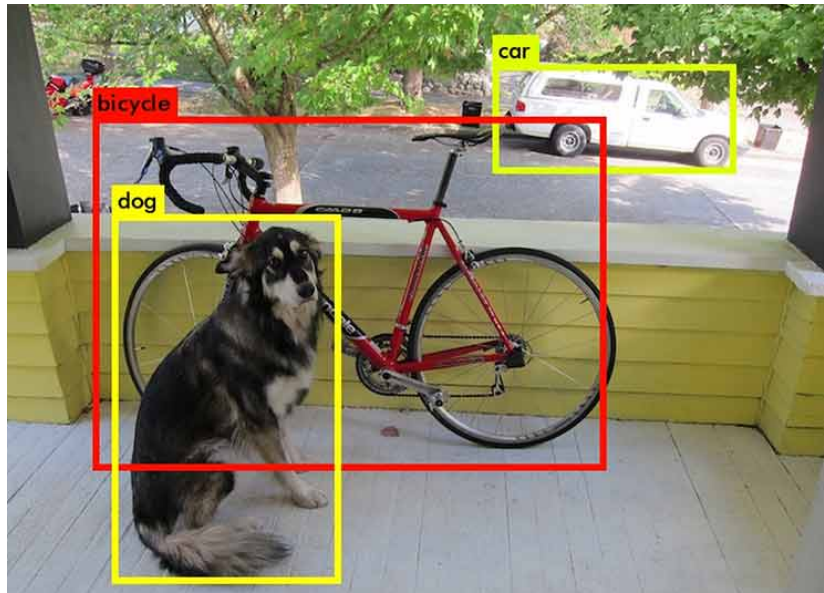


Figure 6: Object detection



Road

Sidewalk

Building

Fence

Pole

Vegetation

Vehicle

Unlabel

Figure 7: Image segmentation

Toepassingen van supervised learning

- ▶ Diagnostiek (analyse van röntgen, EEG, etc.)
- ▶ Speech-to-text
- ▶ Gezichtsherkenning

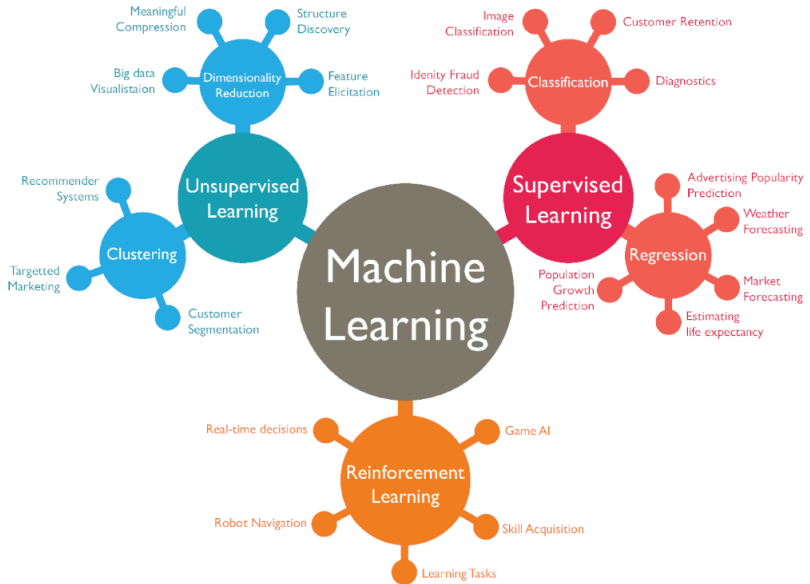


Figure 8: Richtingen en toepassingen van machine learning

Self-learning AI in Games

- ▶ Supervised learning; voornamelijk voor demonstratie van nieuwe netwerkstructuren
- ▶ Reinforcement learning; zelflerende “agenten” in games
- ▶ Unsupervised learning; genereer nieuwe data adhv aangegeven eisen (procedural content generation)

Agenda

- ▶ Twee delen
- ▶ Reinforcement learning
- ▶ Unsupervised learning
- ▶ Jullie suggesties?

	Onderwerp	Huiswerk
22 april	Introductie & Supervised Learning	
29 april	Tabular Reinforcement Learning	Q-learning
13 mei	Deep Reinforcement Learning	
20 mei	Random Search voor RL	ARS en CarRacing

	Onderwerp	Huiswerk
3 juni	Unsupervised Learning & Generatieve Modellen	Generative mo
10 juni	Generative Adversarial Networks	
16 juni	Discussie: Toepassingen	

Huiswerkopdrachten

- ▶ Python
- ▶ OpenAI Gym
- ▶ Deadline de dag voor de volgende opdracht