

Machine Learning

Een nieuwe vorm van programmeren

Stijn Lievens
2022-2023

HO
GENT

Inhoud

Artificiële Intelligentie

Traditioneel Programmeren?

Machinaal Leren

Lineaire Regressie

Neurale Netwerken

Beeldverwerking

**HO
GENT**

Artificiële Intelligentie

**HO
GENT**

Wat is Artificiële Intelligentie?

Artificieel

- Niet op een natuurlijke manier
- Kunstmatig

Intelligentie

- Relatief moeilijk te definiëren.
- Veel verschillende definities.
- Filosofen vs. psychologen

**HO
GENT**

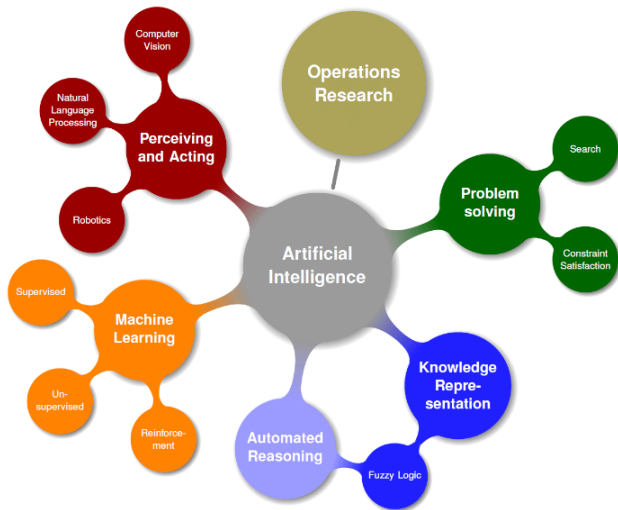
Gangbare praktische definitie

Definitie AI-systeem

Een AI-systeem is een computersysteem dat zelfstandig leert, beslissingen neemt en deze uitvoert.

Bron: boek “Mens vs. Machine” door Geertrui Mieke de Ketelaere

Onderdelen van AI



**HO
GENT**

Traditioneel Programmeren?

**HO
GENT**

Traditioneel Programmeren?

Schrijven van een **spam filter**.

- Hoe ziet spam er typisch uit?

Traditioneel Programmeren?

Schrijven van een **spam filter**.

- Hoe ziet spam er typisch uit?
 - “4U”, “free”, “credit card”, VEEL HOOFDLETTERS
- Schrijf een algoritme om de patronen hierboven te detecteren.
- Test programma en herhaal stappen 1 en 2.

Mogelijke Code

```
1 def is_spam(email):
2     if heeft_veel_hoofdletters(email):
3         return True
4     if aantal_voorkomens(email, "credit card") > 2:
5         if header(email) != "Mijn bank":
6             return True
7     if aantal_spellingsfouten(email) > 10:
8         if afzender(email) != "Mijn vriend":
9             return True
10        else:
11            return False
12    # Nog heel veel andere regels
13    return False
```

Problemen met deze Aanpak?

Problemen met deze Aanpak?

- De lijst met regels wordt snel heel lang en complex; je zal veel pogingen nodig hebben om de lijst op punt te stellen.
- Wat als de spammers doorkrijgen hoe jouw lijst werkt?
 - Dan kan je herbeginnen!

Machinaal Leren

**HO
GENT**

Wat is Machinaal Leren?

Machinaal Leren

Machinaal Leren is de wetenschap (en kunst) om computers te programmeren zodanig dat ze kunnen *leren van data*.

Machinaal Leren

Machinaal Leren is het studiegebied dat computers de mogelijkheid geeft om te leren zonder *expliciet* geprogrammeerd te zijn. (Arthur Samuel, 1959)

Toepassingen van ML

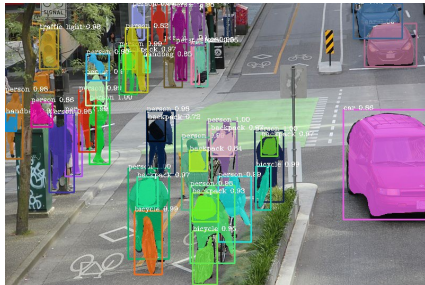
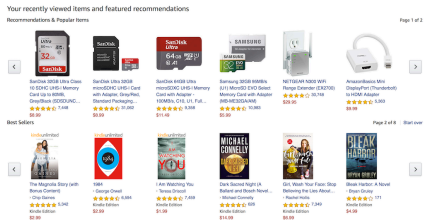
Welke toepassingen van machinaal leren ken jij?

Toepassingen van ML

Welke toepassingen van machinaal leren ken jij?

- Spamdetectie
- Classificatie van beelden
- Fraudedetectie bij creditkaarten
- Aanbevelingssystemen (recommendation systems)
- Spraakherkenning (bv. Siri, Alexa, ...)
- ...

Voorbeelden



Bronnen: <https://emerj.com/ai-sector-overviews/artificial-intelligence-at-amazon/en>
<https://towardsdatascience.com/image-segmentation-with-six-lines-of-code-acb870a462e8>

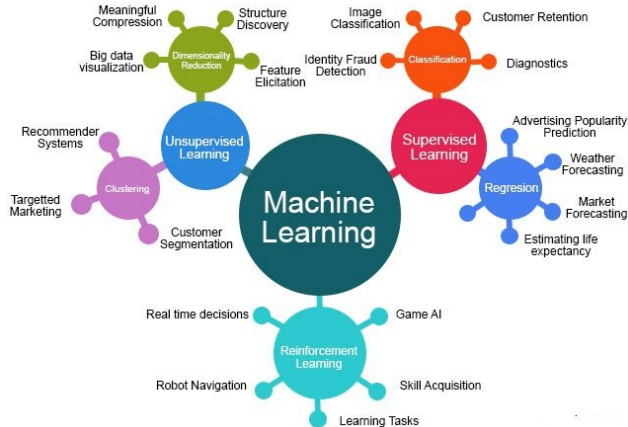
HO
GENT

Verschillende Soorten ML

- **Gesuperviseerd** leren. (Eng. supervised learning)
 - o Gegeven veel gelabelde voorbeelden (bv. emails met label spam/ham) bepaal het label voor een nieuw voorbeeld.
- **Ongesuperviseerd** leren. (Eng. unsupervised learning)
 - o Ontdek structuur in ongelabelde data. (bv. clustering)
- **Reinforcement** learning.
 - o Leer een beleid (Eng. policy) dat de grootste beloning op lange termijn oplevert. Zie bv. <https://www.youtube.com/watch?v=V1eYniJ0Rnk>

Soorten ML

Machine Learning Algorithms



**HO
GENT**

Bron: https://www.researchgate.net/figure/Categorization-of-Machine-Learning-5_fig4_343079524

Lineaire Regressie

**HO
GENT**

Notebook Lineaire Regressie

Open de Jupyter notebook `LineaireRegressie.ipynb` op Google Colab (<https://colab.research.google.com>).

Deze notebook is beschikbaar op <https://github.com/HoGentTIN/ml-workshop-middelbaar>

Lineaire Regressie

Samenvatting notebook `LineaireRegressie.ipynb`.

- Definieer een model met **parameters**: a en b in dit geval.
- Definieer een **kostfunctie** die de kwaliteit van de parameters aangeeft (kleiner is beter).
- Minimaliseer deze kostfunctie m.b.v. **gradient descent**: dit geeft je dan de beste parameters voor jouw model.

Conclusie Lineaire Regressie

Conclusie

De drie voorgaande stappen:

- **Model**
- **Kostfunctie**
- **Optimalisatie m.b.v. gradient descent**

zijn de basisstappen voor heel veel modellen van gesuperviseerd leren.

Echter! Nog veel andere technieken nodig om dit praktisch te laten werken op ingewikkelde problemen.

**HO
GENT**

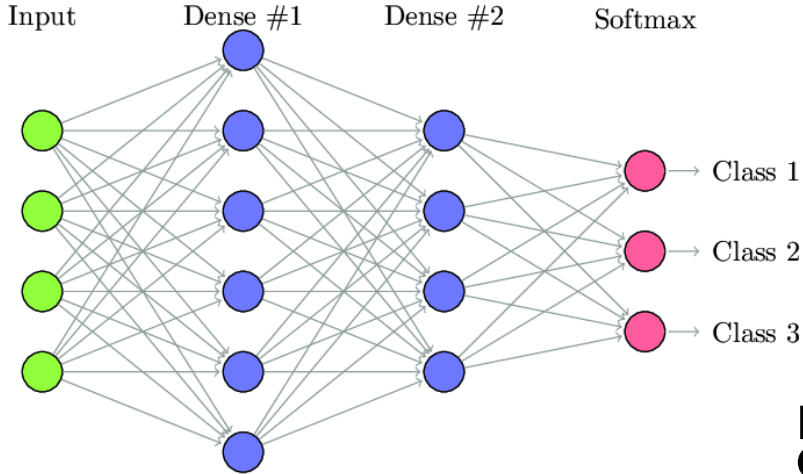
Neurale Netwerken

**HO
GENT**

(Binaire) classificatie

- Bij **binaire classificatie** tracht men de invoer te classificeren als 1 van 2 mogelijke klassen.
- Bv. spam of ham, kat of hond, of ...
<https://www.youtube.com/watch?v=tWwCK95X6go>
- Bij **classificatie** zijn er meerdere klassen, bv. E-mails worden in verschillende folders geclassificeerd: "Primary", "Social", "Reclame".

Neurale netwerken



**HO
GENT**

Neurale netwerken

- Een neurale netwerk bestaat uit een groot aantal **neuronen**.
 - Elk neuron doet een **heel eenvoudige berekening**: nl. een gewogen gemiddelde van de invoer gevolgd door een activatiefunctie.
- De uitvoer van een neuron wordt de invoer voor andere neuron.
- Neuronen worden typisch georganiseerd in **lagen**.
 - Als er veel lagen zijn: een “diep” neurale netwerk \Rightarrow “deep learning”.
- De parameters van het model zijn de sterktes van de verbindingen tussen de neuron.
 - Deze parameters worden aangepast tijdens het trainen van het neurale netwerk.

Tensorflow Playground

Ga naar <https://playground.tensorflow.org/>.

Doe de opdracht die hier beschreven staat:

<https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises>

Beeldverwerking

**HO
GENT**

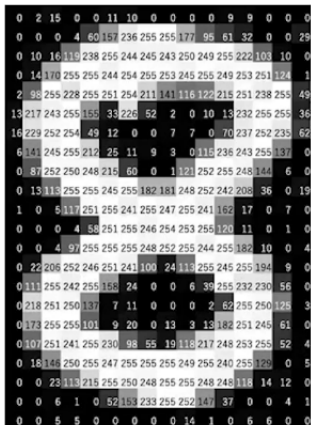
Computerafbeeldingen?

Een computer “ziet” een foto/afbeelding eigenlijk als niets anders dan een raster van getallen.

- Getallen zijn vaak gehele getallen tussen 0 en 255.
- Voor een zwart/wit foto is er één getal per pixel.
- Voor een kleurenfoto zijn er drie waarden per pixel, één voor het kanaal “Rood”, één voor het kanaal “Groen” en één voor het kanaal “Blauw”. Zie:

https://www.youtube.com/watch?v=UBX2QQHlQ_I&t=118s.

Voorbeeldafbeelding



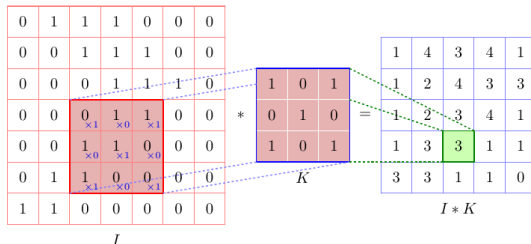
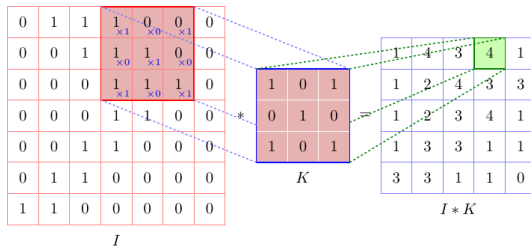
HO GENT

Convoluties

Een convolutie is een bewerking waarbij een klein venster (bv. 3×3) verschoven wordt over een groter beeld.

De pixels die overlappen in het grote beeld en in het kleine venster worden twee aan twee met elkaar vermenigvuldigd en daarna opgeteld. Dit is dan een nieuwe “pixel” in het uitvoerbeeld.

Convolutie: Voorbeeld



**HO
GENT**

Convolutie: 3D-Animatie

Zie <https://animatedai.github.io/>.

Convolutie: Python Notebook

Ga zelf aan de slag met de notebook `TestConvoluties.ipynb` om wat extra intuïtie op te doen m.b.t. convoluties.

Convoluties: leren

- Convoluties zijn niet nieuw en werden al gebruikt lang voor machine en deep learning populair waren.
- Traditioneel werden de convoluties “handmatig” bepaald om bepaalde zaken te detecteren (bv. horizontale of verticale randen).
- Binnen machine learning zijn de waarden in het venstertje *parameters* van het model die worden bepaald (i.e. geleerd) door het algoritme. (Dus net zoals de parameters a en b bij lineaire regressie.)
- In één bepaalde laag zijn er typisch veel “venstertjes” om te leren!
 - Het getransformeerde beeld krijgt dus altijd meer en meer lagen (evenveel als er venstertjes zijn).

Convolutionele NN: Details

- Na het uitvoeren van de convolutie worden typisch de negatieve woorden gewoon op nul gezet (en de andere blijven gewoon behouden).
 - De officiële naam hiervoor is de “Rectified Linear Unit” typisch afgekort tot ReLu.
- Om het aantal parameters onder controle te houden worden de hoogte en breedte van het beeld dan (typisch) gehalveerd m.b.v. een max pooling laag.
- Op het einde van een convolutioneel NN komen er typisch nog een aantal volledig geconnecteerde lagen, met op het einde een laag die de verschillende klassen voorstellen.

ImageNet Competitie

- ImageNet bestaat uit een groot aantal afbeeldingen (meer dan 1.000.000) onderverdeeld in 1000 klassen.
- Gebruikt als een prestigieuze competitie om “computer vision” mogelijkheden te evalueren.

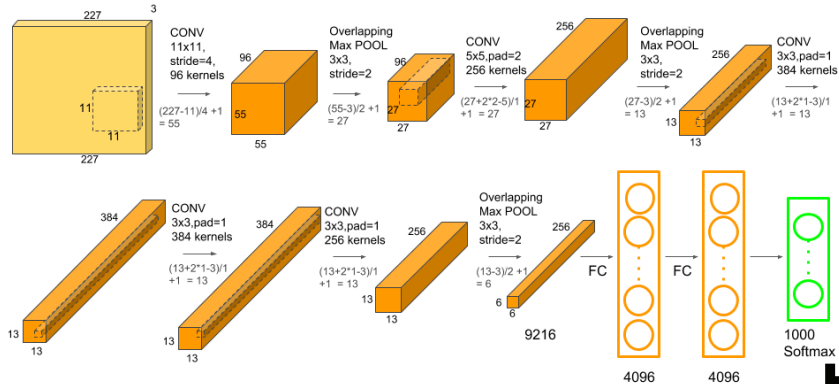


**HO
GENT**

AlexNet

- Alex Krizhevsky nam in 2012 samen met Ilya Sutskever en Geoffrey Hinton deel aan de ImageNet competitie met een convolutioneel neuraal netwerk.
- Hij won, en hoe! Zijn model had “top five error rate” van 16 procent, het model op de tweede plaats haalde 26 procent.

AlexNet: Architectuur



**HO
GENT**

Algemene Architectuur NN

Architectuur Neuraal Netwerk

Een neuraal netwerk

- **Bestaat uit een (groot) aantal lagen.**
- **Elke laag voert een relatief eenvoudige bewerking uit.**
- **Samen vormen de lagen een (ingewikkelde) wiskundige functie die invoer op uitvoer afbeeldt.**

De parameters van het neuraal netwerk worden getraind met (een variant van) gradient descent om de “optimale” waarden van de parameters te bekomen.

**HO
GENT**

Trainen van een ML model zonder code

- We gebruiken de AutoTrain-feature van Hugging Face om een hond-vs-kat classificatiemodel te maken.
- Dit classificatiemodel kan dan aangesproken worden over het Internet.
- Merk op: dit model werd (hoogstwaarschijnlijk) getraind via **transfer learning**.

Transfer Learning

- Bij **transfer learning** wordt een bestaand model hergebruikt voor een gelijkaardige (maar verschillende taak).
- Bv. voor de classificatie van honden vs. katten kunnen we gebruikmaken van een netwerk dat werd getraind op ImageNet, en dat dus 1000 verschillende klassen kan herkennen.
 - De laatste laag van dit model bestaat uit 1000 neuronen.
- We vervangen de laatste (en meest specifieke) laag door een nieuwe laag (met twee neuronen) die honden van katten kan onderscheiden.

Demo AutoTrain Hugging Face

- Maak een (gratis) account op Hugging Face:
<https://huggingface.co/>
- Bedenk twee (of meer) categorieën die je van elkaar wil kunnen onderscheiden.
- Gebruik een zoekmachine om een aantal afbeeldingen van elke categorie te verzamelen: minstens 10 afbeeldingen in elke categorie.
- Volg stappenplan bij de AutoTrain feature.
- “Deploy” je model en roep het aan vanuit Python (bv. Google Colab) om voorspellingen te krijgen.

Workshop Conclusie

- Wees niet ontgoocheld als je vandaag niet alles hebt begrepen:
 - Van basis Machine Learning t.e.m. Transfer Learning zijn in de opleiding Toegepaste Informatica **twee** vakken in het tweede en derde jaar!
- Een diepgaande kennis van de wiskunde en de algoritmen is niet nodig om succesvol te zijn in Machine Learning.
 - Het is *wel* nodig om logisch te kunnen nadenken!
 - Het helpt om te snappen wat er gebeurt in de modellen op het moment dat het fout loopt, zodat je efficiënt kan debuggen.
- In de opleiding Toegepaste Informatica leer je niet alleen over wat je vandaag hebt gezien maar ook wat er nodig is om je model “in productie” te gebruiken.
 - Dit uiteraard naast de basisvaardigheden die je nodig hebt om aan de slag te gaan als **all-round informaticus**.

**HO
GENT**