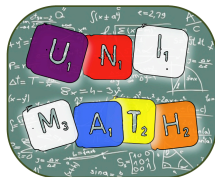


# Wiskundige aanpak van vage informatie: basisprincipes en toepassingen in artificiële intelligentie

Chris Cornelis

Vakgroep Toegepaste Wiskunde, Informatica en Statistiek





[Home](#) > [Research](#) > [Active team members](#) > [Chris Cornelis](#)

## Chris Cornelis

### Contact

Ghent University  
Dept. of Applied Mathematics, Computer Science and Statistics  
Krijgslaan 281 (Building S9, Room 40.09.120.028)  
9000 Gent  
Belgium

Phone: [+32 9 264 47 72](tel:+3292644772)

Email: [chris.cornelis@ugent.be](mailto:chris.cornelis@ugent.be)



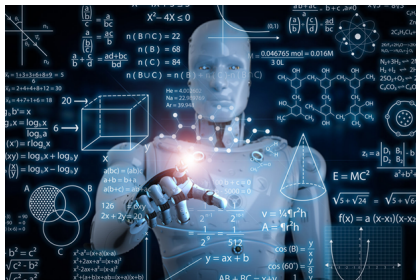
### Short biography

Chris Cornelis holds an M.Sc. (2000) and a Ph.D. (2004) degree in Computer Science from Ghent University (Belgium). He has worked as a postdoctoral researcher both at Ghent University and at the [University of Granada](#). Currently, he is a full-time professor at the Department of Applied Mathematics, Computer Science and Statistics of Ghent University.

<http://www.cwi.ugent.be>

1. Belang van wiskunde voor AI
2. Neurale netwerken en de zoektocht naar Explainable AI
3. Logica en vaaglogica
4. Toepassing in hardware: *fuzzy control*
5. Toepassing in natural language processing/machine learning: *emotion detection from tweets X posts*

# Wiskunde als motor van Artificiële Intelligentie



*AI is not magic; it's just mathematics.*

— Jana Eggers, CEO of Nara Logics

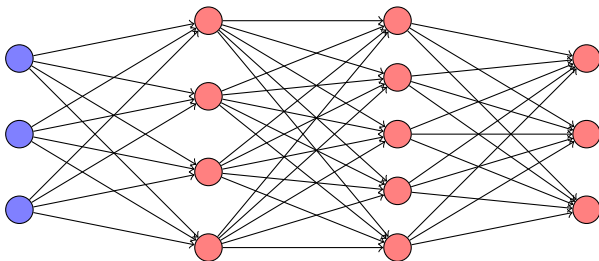
- ▶ Zoekmachines zoals Google (lineaire algebra)
- ▶ Aanbevelingssystemen zoals in Facebook, TikTok, Amazon, Netflix, ... (statistiek, lineaire algebra)
- ▶ Patroonherkenning voor bijvoorbeeld gezichtsherkenning (wiskundige analyse)
- ▶ Natuurlijke-taal verwerking zoals in ChatGPT (statistiek, lineaire algebra, wiskundige analyse)
- ▶ Beveiliging en cryptografie (discrete wiskunde)
- ▶ Computer vision (meetkunde)
- ▶ ...

## Watch: NVIDIA Self-Driving Car Learns How to Become a Better Driver

After just 3,000 miles, NVIDIA's self-driving car uses its DAVENET deep-learning network to learn the rules of the road and become a better driver.

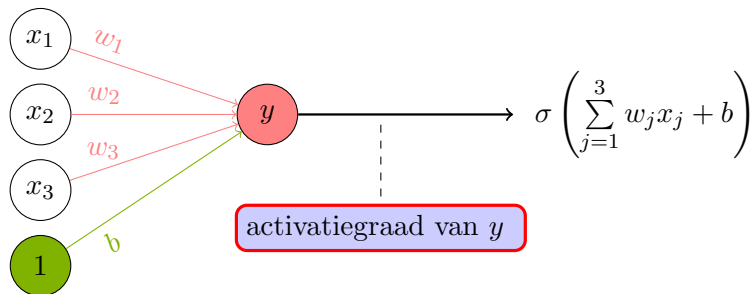


# Hoe werken artificiële neurale netwerken (ANNs)?

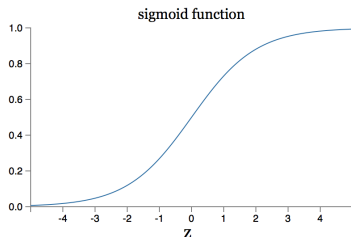


- ▶ *Multi-Layer Perceptron* (MLP): opgebouwd uit neuronen in verschillende lagen
- ▶ Er is één inputlaag, één outputlaag en één of meerdere verborgen lagen
- ▶ Elk rood neuron past een niet-lineaire, *activatiefunctie* toe op de gewogen som van zijn inputs

# Voorbeeld: sigmoid neuron

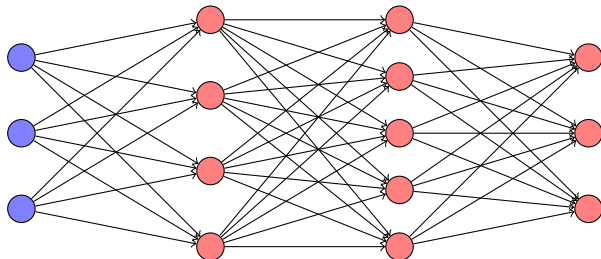


$$\sigma(z) = \frac{1}{1 + e^{-z}}$$





# Het trainen van ANNs



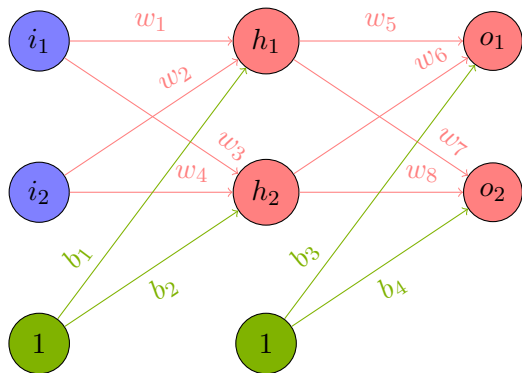
- ▶ Output van het ANN hangt af van de vectoren  $w$  en  $b$
- ▶ Kostfunctie:

$$C(w, b) = \frac{1}{n} \sum_{i=1}^n \|y^i - output(x^i)\|^2$$

met gegeven **training set**  $\{(x^i, y^i) | i = 1, \dots, n\}$

- ▶ Doel: vertrekkend van willekeurige  $w$  en  $b$ , passen we stapsgewijs deze vectoren aan totdat  $C(w, b)$  klein genoeg wordt

# Het trainen van ANNs: voorbeeld

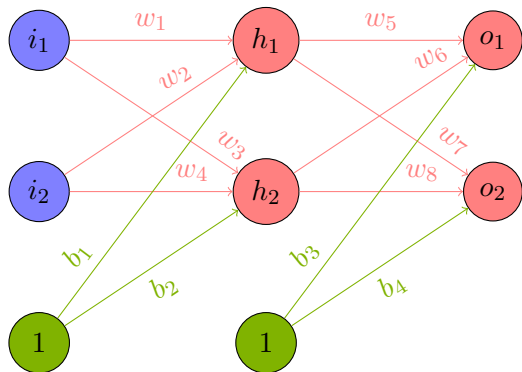


$$\begin{array}{ll} w_1 = 0.1 & w_2 = 0.15 \\ w_3 = 0.2 & w_4 = 0.25 \\ w_5 = 0.3 & w_6 = 0.35 \\ w_7 = 0.4 & w_8 = 0.45 \end{array}$$

$$\begin{array}{ll} b_1 = 0.5 & b_2 = 0.55 \\ b_3 = 0.6 & b_4 = 0.65 \end{array}$$

$$\text{Trainingvoorbeeld: } x = \begin{pmatrix} 0.2 \\ 0.3 \end{pmatrix}, y = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

# Het trainen van ANNs: voorbeeld



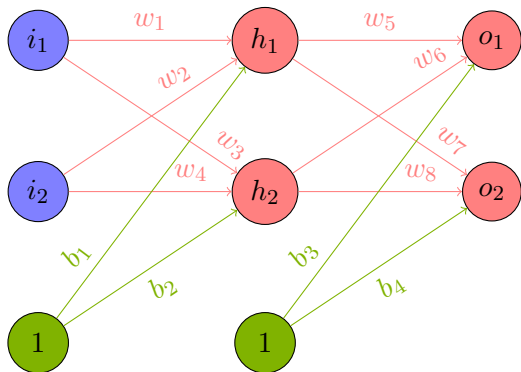
$$\begin{array}{ll} w_1 = 0.1 & w_2 = 0.15 \\ w_3 = 0.2 & w_4 = 0.25 \\ w_5 = 0.3 & w_6 = 0.35 \\ w_7 = 0.4 & w_8 = 0.45 \end{array}$$

$$\begin{array}{ll} b_1 = 0.5 & b_2 = 0.55 \\ b_3 = 0.6 & b_4 = 0.65 \end{array}$$

$$\begin{array}{ll} i_1 = 0.2 & i_2 = 0.3 \end{array}$$

$$\begin{aligned} h_1 &= \sigma(w_1 i_1 + w_2 i_2 + b_1) \\ &= \sigma(0.1 \times 0.2 + 0.15 \times 0.3 + 0.5) \\ &= \frac{1}{1 + e^{-0.565}} \approx 0.64 \end{aligned}$$

# Het trainen van ANNs: voorbeeld



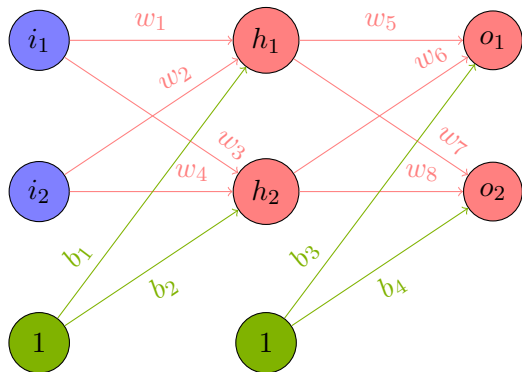
$$\begin{array}{ll} w_1 = 0.1 & w_2 = 0.15 \\ w_3 = 0.2 & w_4 = 0.25 \\ w_5 = 0.3 & w_6 = 0.35 \\ w_7 = 0.4 & w_8 = 0.45 \end{array}$$

$$\begin{array}{ll} b_1 = 0.5 & b_2 = 0.55 \\ b_3 = 0.6 & b_4 = 0.65 \end{array}$$

$$\begin{array}{ll} i_1 = 0.2 & i_2 = 0.3 \end{array}$$

$$\begin{aligned} h_2 &= \sigma(w_3 i_1 + w_4 i_2 + b_2) \\ &= \sigma(0.2 \times 0.2 + 0.25 \times 0.3 + 0.55) \\ &= \frac{1}{1 + e^{-0.615}} \approx 0.65 \end{aligned}$$

# Het trainen van ANNs: voorbeeld



$$w_1 = 0.1 \quad w_2 = 0.15$$

$$w_3 = 0.2 \quad w_4 = 0.25$$

$$w_5 = 0.3 \quad w_6 = 0.35$$

$$w_7 = 0.4 \quad w_8 = 0.45$$

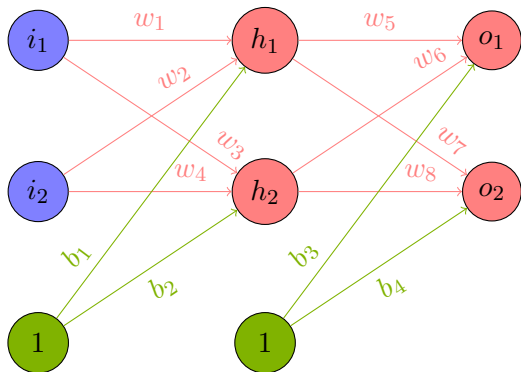
$$b_1 = 0.5 \quad b_2 = 0.55$$

$$b_3 = 0.6 \quad b_4 = 0.65$$

$$h_1 = 0.64 \quad h_2 = 0.65$$

$$\begin{aligned} o_1 &= \sigma(w_5 h_1 + w_6 h_2 + b_3) \\ &= \sigma(0.3 \times 0.65 + 0.35 \times 0.66 + 0.6) \\ &= \frac{1}{1 + e^{-1.026}} \approx 0.74 \end{aligned}$$

# Het trainen van ANNs: voorbeeld



$$w_1 = 0.1 \quad w_2 = 0.15$$

$$w_3 = 0.2 \quad w_4 = 0.25$$

$$w_5 = 0.3 \quad w_6 = 0.35$$

$$w_7 = 0.4 \quad w_8 = 0.45$$

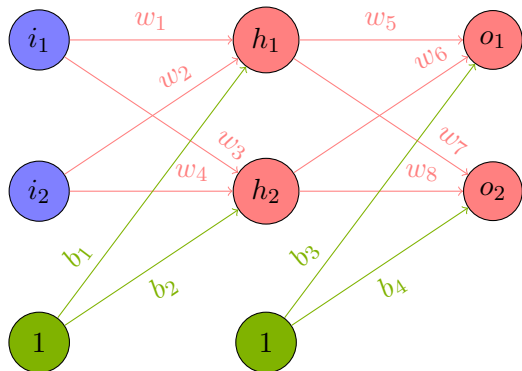
$$b_1 = 0.5 \quad b_2 = 0.55$$

$$b_3 = 0.6 \quad b_4 = 0.65$$

$$h_1 = 0.64 \quad h_2 = 0.65$$

$$\begin{aligned} o_2 &= \sigma(w_7 h_1 + w_8 h_2 + b_4) \\ &= \sigma(0.4 \times 0.65 + 0.45 \times 0.66 + 0.65) \\ &= \frac{1}{1 + e^{-1.207}} \approx 0.77 \end{aligned}$$

# Het trainen van ANNs: voorbeeld



$$\begin{array}{ll} w_1 = 0.1 & w_2 = 0.15 \\ w_3 = 0.2 & w_4 = 0.25 \\ w_5 = 0.3 & w_6 = 0.35 \\ w_7 = 0.4 & w_8 = 0.45 \end{array}$$

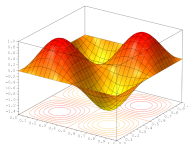
$$\begin{array}{ll} b_1 = 0.5 & b_2 = 0.55 \\ b_3 = 0.6 & b_4 = 0.65 \end{array}$$

$$\begin{array}{ll} o_1 = 0.74 & o_2 = 0.77 \\ y_1 = 0 & y_2 = 1 \end{array}$$

Evaluatie van de kostfunctie:

$$\begin{aligned} C &= \frac{1}{2}(o_1 - y_1)^2 + \frac{1}{2}(o_2 - y_2)^2 \\ &= \frac{1}{2}(0.74 - 0)^2 + \frac{1}{2}(0.77 - 1)^2 = 0.2738 + 0.02645 = 0.30025 \end{aligned}$$

# Het trainen van ANNs: gradiëntmethode



Werkwijze:

- ▶ Bereken per trainingvoorbeeld de **partiële afgeleiden** van de kostfunctie  $C$  naar elke parameter  $w_k$  en  $b_l$
- ▶ Gegeven de “leersnelheid”  $\eta$ , pas  $w_k$  en  $b_l$  als volgt aan:

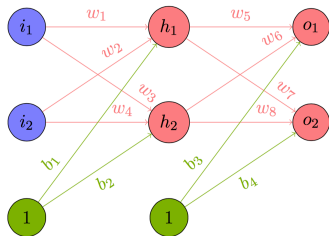
$$w_k \rightarrow w'_k := w_k - \eta \frac{\partial C}{\partial w_k}$$

$$b_l \rightarrow b'_l := b_l - \eta \frac{\partial C}{\partial b_l}$$

- ▶ De nieuwe waarden van  $w$  en  $b$  zijn het gemiddelde van de berekende waarden over alle trainingvoorbeelden



# Gradiëntmethode: voorbeeld



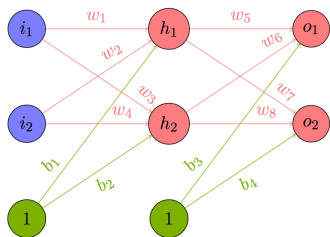
$$\eta = 3$$

(leersnelheid)

$w_1 = 0.1 \rightarrow 0.1$	$w_2 = 0.15 \rightarrow 0.15$
$w_3 = 0.2 \rightarrow 0.2$	$w_4 = 0.25 \rightarrow 0.25$
$w_5 = 0.3 \rightarrow 0.03$	$w_6 = 0.35 \rightarrow 0.08$
$w_7 = 0.4 \rightarrow 0.49$	$w_8 = 0.45 \rightarrow 0.54$
$b_1 = 0.5 \rightarrow 0.47$	$b_2 = 0.55 \rightarrow 0.52$
$b_3 = 0.6 \rightarrow 0.18$	$b_4 = 0.65 \rightarrow 0.77$
$o_1 = 0.74 \rightarrow 0.56$	$o_2 = 0.77 \rightarrow 0.81$

$$w_k \rightarrow w'_k := w_k - \eta \frac{\partial C}{\partial w_k} \quad (k = 1, \dots, 8)$$
$$b_l \rightarrow b'_l := b_l - \eta \frac{\partial C}{\partial b_l} \quad (l = 1, \dots, 4)$$

# Herevaluatie van de kostfunctie



$\eta = 3$

(leersnelheid)

$$w_1 = 0.1 \rightarrow 0.1$$

$$w_2 = 0.15 \rightarrow 0.15$$

$$w_3 = 0.2 \rightarrow 0.2$$

$$w_4 = 0.25 \rightarrow 0.25$$

$$w_5 = 0.3 \rightarrow 0.03$$

$$w_6 = 0.35 \rightarrow 0.08$$

$$w_7 = 0.4 \rightarrow 0.49$$

$$w_8 = 0.45 \rightarrow 0.54$$

$$b_1 = 0.5 \rightarrow 0.47$$

$$b_2 = 0.55 \rightarrow 0.52$$

$$b_3 = 0.6 \rightarrow 0.18$$

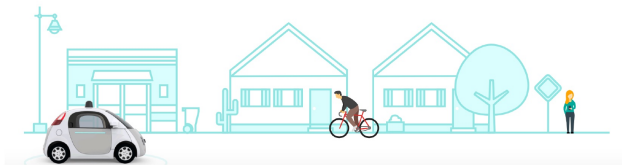
$$b_4 = 0.65 \rightarrow 0.77$$

$$o_1 = 0.74 \rightarrow 0.56$$

$$o_2 = 0.77 \rightarrow 0.81$$

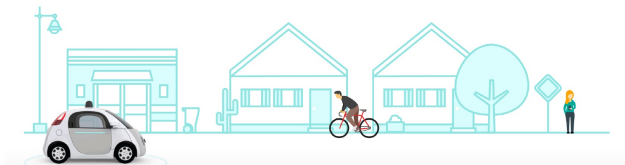
$$\begin{aligned} C &= \frac{1}{2}(o_1 - y_1)^2 + \frac{1}{2}(o_2 - y_2)^2 \\ &= \frac{1}{2}(0.56 - 0)^2 + \frac{1}{2}(0.81 - 1)^2 = 0.1568 + 0.01805 = 0.17485 \\ &\text{(initieel: } C = 0.30025\text{)} \end{aligned}$$

# Terug naar de zelfrijdende auto...



- ▶ Neurale netwerken kunnen gewenst gedrag zeer nauwkeurig benaderen

# Terug naar de zelfrijdende auto...



- ▶ Neurale netwerken kunnen gewenst gedrag zeer nauwkeurig benaderen
- ▶ Maar kunnen we een *black box* model wel vertrouwen?



## You Can Trick Self-Driving Cars by Defacing Street Signs

By [Catalin Cimpanu](#)

August 7, 2017

07:31 AM

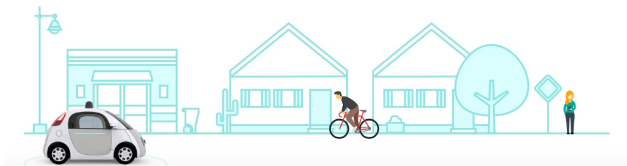
6



A team of eight researchers has discovered that by altering street signs, an adversary could confuse self-driving cars and cause their machine-learning systems to misclassify signs and take wrong decisions, potentially putting the lives of passengers in danger.

The idea behind this research is that an attacker could (1) print an entirely new poster and overlay it over an existing sign, or (2) attach smaller stickers on a legitimate sign in order to fool the self-driving car into thinking it's looking at another type of street sign.

# Explainable AI (XAI)



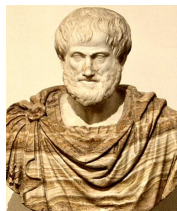
- ▶ We verwachten van AI toepassingen dat ze hun acties kunnen uitleggen
- ▶ Mensen verklaren hun acties met **logica**:
  - ▶ Ik mag de fietser niet hinderen, dus wijk ik *lichtjes* naar rechts uit voordat ik hem kruis.
  - ▶ De voetganger wil oversteken dus start ik *tijdig* met remmen.
- ▶ Merk op: heel vaak komen hierbij **vage** (imprecieze) termen uit de natuurlijke taal voor



Alle mensen zijn sterfelijk.  
Socrates is een mens.  

---

Dus: Socrates is sterfelijk.



Aristoteles  
★ Chalkidiki, 384BC  
† Euboea, 322BC

- ▶ Klassieke logica ligt aan de basis van al onze wetenschappelijke theorieën
- ▶ Helaas is ze niet vrij van onvolkomenheden en paradoxen

# Paradox van de barbier

*“De barbier scheert iedereen die zichzelf niet scheert.  
Wie scheert de barbier?”*

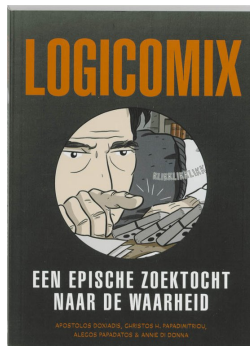


*“You shave ‘all the men who don’t shave themselves’?  
What do you mean? Who the fuck is  
shaving you then?”*



Bertrand Russell  
★ Trellech (Wales), 1872  
† Penrhyndeudraeth  
(Wales), 1970





- ▶ Principia Mathematica (Russell & Whitehead, 1910–1913): zoektocht naar axioma's en bewijsregels waaruit *alle* wiskundige resultaten volgen

$$*54.43. \vdash : \alpha, \beta \in 1. \supset : \alpha \cap \beta = \Lambda. \equiv . \alpha \cup \beta \in 2$$

*Dem.*

$$\vdash . *54.26. \supset \vdash : \alpha = t'x. \beta = t'y. \supset : \alpha \cup \beta \in 2. \equiv . x \neq y.$$

$$[*51.231] \quad \equiv . t'x \cap t'y = \Lambda.$$

$$[*13.12] \quad \equiv . \alpha \cap \beta = \Lambda \quad (1)$$

$$\vdash . (1). *11.11.35. \supset$$

$$\vdash : (\exists x, y). \alpha = t'x. \beta = t'y. \supset : \alpha \cup \beta \in 2. \equiv . \alpha \cap \beta = \Lambda \quad (2)$$

$$\vdash . (2). *11.54. *52.1. \supset \vdash . \text{Prop}$$

From this proposition it will follow, when arithmetical addition has been defined, that  $1 + 1 = 2$ .

# Wiskunde davert op haar grondvesten?

- ▶ Onvolledigheidsstelling (Kurt Gödel, 1931): niet elke wiskundige eigenschap die waar is heeft een bewijs!
- ▶ Er bestaan **onbeslisbare** wiskundige uitspraken: uitspraken waarvan we kunnen bewijzen dat we ze niet kunnen bewijzen

# Wiskunde davert op haar grondvesten?

- ▶ Onvolledigheidsstelling (Kurt Gödel, 1931): niet elke wiskundige eigenschap die waar is heeft een bewijs!
- ▶ Er bestaan **onbeslisbare** wiskundige uitspraken: uitspraken waarvan we kunnen bewijzen dat we ze niet kunnen bewijzen
- ▶ Voorbeeld 1 (“sterfelijkheid” van vierkante matrices):

Gegeven  $k$  vierkante matrices van orde  $n$ , kan men door een eindig product van deze matrices (waarin elke matrix meerdere keren kan voorkomen) de nulmatrix uitkomen?

Sterfelijkheid voor 6 of meer  $3 \times 3$  matrices is onbeslisbaar.

# Wiskunde davert op haar grondvesten?

- ▶ Onvolledigheidsstelling (Kurt Gödel, 1931): niet elke wiskundige eigenschap die waar is heeft een bewijs!
- ▶ Er bestaan **onbeslisbare** wiskundige uitspraken: uitspraken waarvan we kunnen bewijzen dat we ze niet kunnen bewijzen, noch ontkrachten door tegenvoorbeelden
- ▶ Voorbeeld 2 (Hilberts tiende probleem):

Heeft een gegeven diofantische vergelijking (veeltermvergelijking met gehele coëfficiënten) een oplossing waarbij alle veranderlijken geheel zijn? Bijvoorbeeld,  $3x^2 - 2xy - y^2z - 7 = 0$  heeft als oplossing  $x = 1, y = -1, z = -2$ , maar  $x^2 + y^2 + 1 = 0$  heeft geen gehele oplossing.

Er bestaat geen algoritme dat dit voor een willekeurige diofantische vergelijking kan nagaan.



Alan Turing

★ London, 1912

† Wilmslow 1954

- ▶ **Turing machine**: universeel model van een computer, en wat je ermee (niet) kan berekenen

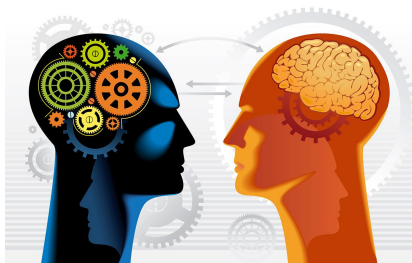


Alan Turing  
★ London, 1912  
† Wilmslow 1954

- ▶ **Turing machine:** universeel model van een computer, en wat je ermee (niet) kan berekenen
- ▶ Ook bekend van:
  - ▶ kraken van de Enigma code tijdens WO II



- ▶ **Turing test:** kan een chatbot zich voordoen als een mens?



- ▶ Computers zijn heel sterk in het rekenen met numerieke gegevens
- ▶ Wij mensen delen kennis op een flexibele manier via onze natuurlijke taal
- ▶ Laat computers “denken” zoals mensen: vertaal *vage begrippen* in natuurlijke taal naar *wiskundige patronen*

# Paradox van de hoop (sorites)

*Hoe definieer je “een hoop zand”?*

(toegeschreven aan Eubulides van Milete)

Premissen:

- ▶ één miljoen zandkorrels is een hoop
- ▶ een hoop waarvan je één zandkorrel wegneemt, blijft een hoop





# Paradox van de hoop (sorites)

*Hoe definieer je “een hoop zand”?*

(toegeschreven aan Eubulides van Milete)

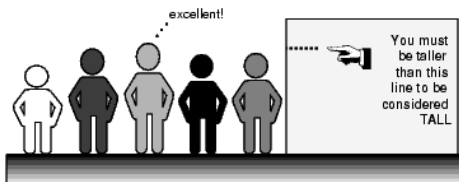
Premissen:

- ▶ één miljoen zandkorrels is een hoop
- ▶ een hoop waarvan je één zandkorrel wegneemt, blijft een hoop



Dus 3, 2, 1, 0 korrels zand is nog steeds een hoop?!

Zij  $X$  de verzameling van mensen in deze zaal.  
Bepaal  $A$ , de verzameling van grote personen binnen  $X$ .

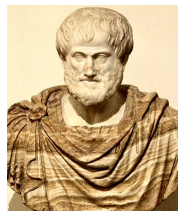


Wet van het uitgesloten derde:  $\mathbf{P \vee \neg P}$

“Tertium non datur”:  
elke logische uitspraak  $P$  is  
ofwel waar, ofwel vals

Wet van de niet-contradictie:  $\neg(\mathbf{P \wedge \neg P})$

Een uitspraak  $P$  kan niet terzelfdertijd  
waar én vals zijn

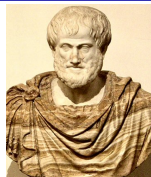


Aristoteles

★ Chalkidiki, 384BC

† Euboea, 322BC

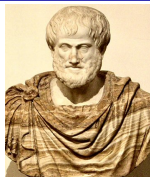
Is de uitspraak  
*“Morgen zullen de Grieken een zeeslag  
tegen de Perzen voeren”*  
waar of niet waar?



Aristoteles  
★ Chalkidiki, 384BC  
† Euboea, 322BC

Is de uitspraak  
“*Morgen zullen de Grieken een zeeslag  
tegen de Perzen voeren*”  
waar of niet waar?

Driewaardige logica:  
een uitspraak is ofwel *vals*,  
ofwel *waar*, ofwel *mogelijk*.



Aristoteles  
★ Chalkidiki, 384BC  
† Euboea, 322BC



Jan Łukasiewicz  
★ Lviv, 1878  
† Dublin, 1956

Driewaardige logica  
lost de paradox van de hoop niet op.



Driewaardige logica  
lost de paradox van de hoop niet op.



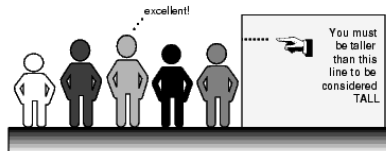
Lotfi Zadeh  
★ Baku, 1921  
† Berkeley, 2017

**Vaaglogica (fuzzy logic):**  
de *waarheidswaarde* van een logische uitspraak is een reëel getal tussen 0 (volledig vals) en 1 (volledig waar), dat uitdrukt in welke mate is voldaan aan een vage eigenschap

Zij  $X$  de verzameling der deelnemers aan deze Zoom sessie.  
Bepaal  $A$ , de verzameling van grote personen in  $X$ .

Klassiek:

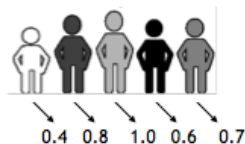
$$x \in A \vee x \notin A$$



Zadeh:

$$A(x) \in [0, 1]$$





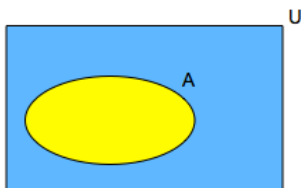
Met elk object  $x$  correspondeert een lidmaatschapsgraad  $A(x)$ .

- ▶  $A(x)$  drukt uit in welke mate een persoon  $x$  groot is.
- ▶ De exacte waarde van  $A(x)$  is context- en persoonsgebonden.
- ▶ Hoe groter de lengte van  $x$ , hoe hoger  $A(x)$  moet zijn:

$$\text{lengte}(x) \leq \text{lengte}(y) \Rightarrow A(x) \leq A(y)$$

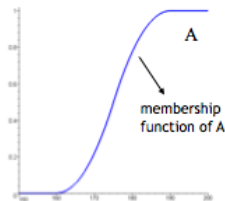
Met elke vage eigenschap correspondeert een lidmaatschapsfunctie.

abrupt ("crisp") transition



$$\begin{aligned}x \in A & A(x) = 1 \\x \notin A & A(x) = 0\end{aligned}$$

gradual transition



$$\begin{aligned}A : X & \rightarrow [0, 1] \\A(x) & \in [0, 1]\end{aligned}$$

# Wat is het verschil tussen probabilliteit en vaagheid?

*Je bent verdwaald in de woestijn en plots vind je twee flessen.*

*Op de eerste fles staat:  
de probabilliteit dat het water in deze fles drinkbaar is, is 0.9.*

*Op de tweede fles staat:  
het water in deze fles is drinkbaar in de mate 0.9.*

*Van welke fles drink je?*

- ▶ Uitgangspunt: elke logische uitspraak heeft een waarheidswaarde tussen 0 en 1
- ▶ Logische uitspraken kunnen worden opgebouwd met behulp van
  1. logische connectieven:  $\neg$  (niet),  $\wedge$  (en),  $\vee$  (of),  $\Rightarrow$  (als...dan),  $\Leftrightarrow$  (als en slechts als), ...
  2. kwantoren: “voor alle” ( $\forall$ ), “er bestaat” ( $\exists$ ), “de meeste”, “bijna alle”, ...

- ▶ Uitgangspunt: elke logische uitspraak heeft een waarheidswaarde tussen 0 en 1
- ▶ Logische uitspraken kunnen worden opgebouwd met behulp van
  1. logische connectieven:  $\neg$  (niet),  $\wedge$  (en),  $\vee$  (of),  $\Rightarrow$  (als...dan),  $\Leftrightarrow$  (als en slechts als), ...
  2. kwantoren: “voor alle” ( $\forall$ ), “er bestaat” ( $\exists$ ), “de meeste”, “bijna alle”, ...
- ▶ Voorbeeld:

“*Bijna alle* ongelukken zijn het gevolg van overdreven snelheid *of* substantiemisbruik.”

Hoe bepaal je van deze zin de waarheidswaarde?
- ▶ Vertrekpunt: *waarheidstabellen* voor klassieke logica

# Klassieke logica: waarheidstabellen

negatie  
(NIET)

$P$	$\neg P$
1	0
0	1

# Klassieke logica: waarheidstabellen

negatie  
(NIET)

$P$	$\neg P$
1	0
0	1

conjunctie  
(EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

# Klassieke logica: waarheidstabellen

negatie  
(NIET)

$P$	$\neg P$
1	0
0	1

conjunctie  
(EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0

disjunctie  
(OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0



negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	?

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	?

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	?

Kunnen we deze waarheidstabellen uitbreiden tot waarheidswaarden  $x$  en  $y$  uit  $[0, 1]$ ?

negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	$1 - x$

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	$\min(x, y)$

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	$\max(x, y)$

negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	$1 - x$

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	$\min(x, y)$

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	$\max(x, y)$

Nadeel van min en max: niet-compenserend gedrag

$$\min(0.2, 0.2) = 0.2$$

$$\min(0.2, 1) = 0.2$$

# Alternatief: probabilistische som en product

negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	$1 - x$

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	$xy$

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	$x + y - xy$

# Alternatief: probabilistische som en product

negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	$1 - x$

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	$xy$

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	$x + y - xy$

► Voordeel: wel compenserend gedrag

# Alternatief: probabilistische som en product

negatie (NIET)

$P$	$\neg P$
1	0
0	1
$x$	$1 - x$

conjunctie (EN)

$P$	$Q$	$P \wedge Q$
1	1	1
1	0	0
0	1	0
0	0	0
$x$	$y$	$xy$

disjunctie (OF)

$P$	$Q$	$P \vee Q$
1	1	1
1	0	1
0	1	1
0	0	0
$x$	$y$	$x + y - xy$

- ▶ Voordeel: wel compenserend gedrag
- ▶ Nadeel: we verliezen andere eigenschappen zoals distributiviteit en idempotentie.

$$\min(x, \max(y, z)) = \max(\min(x, y), \min(x, z))$$

$$\text{MAAR: } x(y + z - yz) \neq xy + xz - x^2yz$$

$$\min(x, x) = x = \max(x, x)$$

$$\text{MAAR: } x^2 \neq x \neq x + x - x^2$$

- ▶ Om de waarheidswaarde van de uitspraak

*“Alle personen in deze zaal zijn groot”*

te bepalen, volstaat het om te kijken naar de persoon met de kleinste lengte:

$$\min\{\text{groot}(x) \mid x \in X\}$$

- ▶ Om de waarheidswaarde van de uitspraak

*“Alle personen in deze zaal zijn groot”*

te bepalen, volstaat het om te kijken naar de persoon met de kleinste lengte:

$$\min\{\text{groot}(x) \mid x \in X\}$$

- ▶ Analoog vind je de waarheidswaarde van de uitspraak

*“Minstens één persoon in deze zaal is groot”*

als de hoogste lidmaatschapsgraad in de vaagverzameling “groot”

$$\max\{\text{groot}(x) \mid x \in X\}$$



- ▶ Om de waarheidswaarde van de uitspraak  
    *“Bijna alle personen in deze zaal zijn groot”*  
kunnen we als volgt te werk gaan:

- ▶ Om de waarheidswaarde van de uitspraak

*“Bijna alle personen in deze zaal zijn groot”*

kunnen we als volgt te werk gaan:

1. Orden de personen ( $i = 1, \dots, n$ ) van klein naar groot:  $x_1$  is de kleinste,  $x_n$  de grootste

- ▶ Om de waarheidswaarde van de uitspraak

*“Bijna alle personen in deze zaal zijn groot”*

kunnen we als volgt te werk gaan:

1. Orden de personen ( $i = 1, \dots, n$ ) van klein naar groot:  $x_1$  is de kleinste,  $x_n$  de grootste
2. Ken aan elke persoon  $x_i$  een gewicht  $w_i$  toe, waarbij geldt:

$$2.1 \quad i \leq j \Rightarrow w_i \geq w_j$$

$$2.2 \quad \sum_{i=1}^n w_i = 1$$

- ▶ Om de waarheidswaarde van de uitspraak

*“Bijna alle personen in deze zaal zijn groot”*

kunnen we als volgt te werk gaan:

1. Orden de personen ( $i = 1, \dots, n$ ) van klein naar groot:  $x_1$  is de kleinste,  $x_n$  de grootste
2. Ken aan elke persoon  $x_i$  een gewicht  $w_i$  toe, waarbij geldt:
  - 2.1  $i \leq j \Rightarrow w_i \geq w_j$
  - 2.2  $\sum_{i=1}^n w_i = 1$
3. Bereken de waarheidswaarde van de uitspraak als

$$\sum_{i=1}^n w_i \times \text{groot}(x_i)$$

Voorbeeld:

1. Drie personen:  $\text{groot}(x_1) = 0.7$ ,  $\text{groot}(x_2) = 0.9$ ,  
 $\text{groot}(x_3) = 1.0$

1.1  $w_1 = 0.8, w_2 = 0.2, w_3 = 0$

1.2  $\sum_{i=1}^n w_i = 1$

2. De waarheidswaarde van “Bijna alle personen in de zaal zijn groot” is dan

Voorbeeld:

1. Drie personen:  $\text{groot}(x_1) = 0.7$ ,  $\text{groot}(x_2) = 0.9$ ,  
 $\text{groot}(x_3) = 1.0$

- 1.1  $w_1 = 0.8, w_2 = 0.2, w_3 = 0$

- 1.2  $\sum_{i=1}^n w_i = 1$

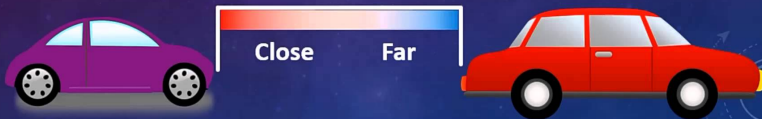
2. De waarheidswaarde van “Bijna alle personen in de zaal zijn groot” is dan

$$\sum_{i=1}^n w_i \times \text{groot}(x_i) = 0.8 \times 0.7 + 0.2 \times 0.9 + 0 \times 1.0 = 0.74$$

WHY IS IT USEFUL?

Automatic Braking System

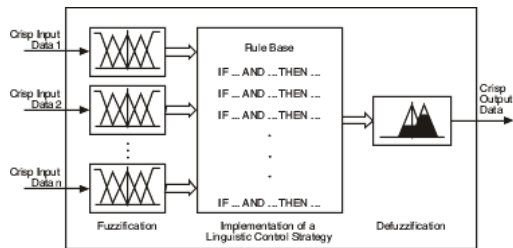
Fuzzy Logic



Is car close? : 0-1 (Range of No to Yes)

Brakes : 0-1 (Range of Off to On)

# Toepassing van vaaglogica: fuzzy control



Ebrahim Mamdani  
★ Tanzania, 1942  
† Verenigd Koninkrijk,  
2010

Principe:

- ▶ Regels in natuurlijke taal voor de werking van een toestel worden vertaald naar vaaglogica
- ▶ Goedkoop en efficiënt alternatief voor een exacte (niet-lineaire) controller

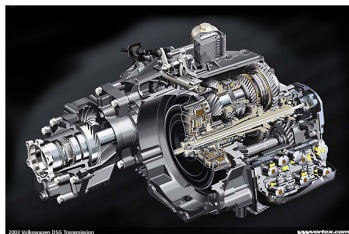


# Fuzzy control: enkele toepassingen



Fuzzy control is ingebouwd in de automatische versnellingsbak van automodellen zoals de Audi TT, Volkswagen Touareg, ...

Metro's in Azië en Europa gebruiken fuzzy control voor vlotte overgangen tussen versnellen en vertragen



# Fuzzy control: enkele toepassingen

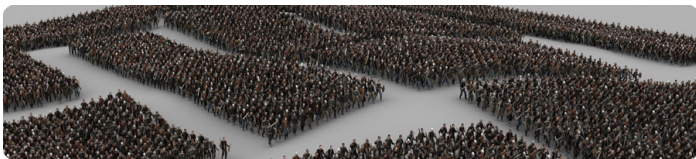


Fuzzy control wordt gebruikt voor het scherpstellen van de lens in digitale camera's

Dankzij fuzzy control kunnen wasmachines hun waterverbruik optimaliseren



# Fuzzy control: reële toepassingen



## Features

Shared Features (Massive Prime™, Massive Jet™ and Massive for Maya™)	Features Specific to Massive Prime™
<b>Body Editor</b> The Body Editor provides the workflow to build and customize skeletons, geometry, cloth, and materials. It enables the automated variation of agent proportions (skeleton) and appearance (geometry, shaders, and texture maps).	<b>Brain Editor</b> The intuitive node-based Brain Editor interface allows artists to interactively create AI-enabled agents without any programming. The AI toolset gives artists the freedom to build custom AI logic for the specific behavior they want to simulate.
<b>Scene Editor</b> The Scene Editor provides the workflow to control and direct cameras, terrain and agents. Tools are provided for both general and specific control and direction of agents. Crowds in a stadium can easily be made to clap and cheer at the right time with the general agent controls. You can also control the parameters of individual agent placement and behavior.	<b>Motion Tree Editor</b> The Motion Tree is used to design the agent's motion, which then becomes an integral part of the agent. The output from the Motion Tree can then be used as a take list for the mocap session. This Motion Tree workflow, along with the integrated Action Editor, allows key-framed or motion-captured animation clips to be actions linked to the brain.
<b>Natural Senses</b> Massive is based on artificial life technology. Massive characters use a patented vision process, and a sense of hearing and touch that allows them to respond naturally to their environment.	<b>Action Editor</b> A motion editing program for importing motion. Cross fade looping editor is a convenient way of creating seamless loops. Prepare clips for Massive. Quality of motion. Its integrated motion editing built into Massive so you don't need to use another program import and can edit it in context.
<b>Fuzzy Logic for Subtlety</b> Massive animators use fuzzy logic to design their characters' responses. Instead of a value being black or white, it can be a shade of grey or 'fuzzy' - giving the character more natural responses than the on/off robotic results of binary logic.	<b>Smart Stunts™</b> Massive's digital stunts are controlled by dynamics that pull from real motion capture data. Filmmakers can direct the motions and reactions they want with real stunt actors and then import these actions into a Massive agent to give direction and character to the agent's performance.
<b>AI for Film Quality</b>	<b>Cloth Editor</b>

Fuzzy control werd o.m. gebruikt bij de animaties voor de Lord of the Rings trilogie (MassiveSoftware.com)

Voorbeeld: automatische regeling van airconditioner

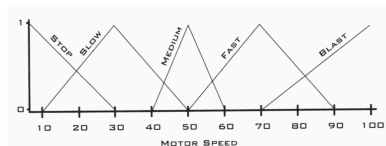
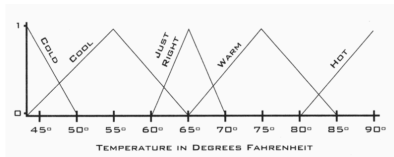
Stap 1 bepaal de in- en uitvoerveranderlijken

Invoer	Temperatuur ( $X$ )
Uitvoer	Snelheid motor airconditioner ( $Y$ )

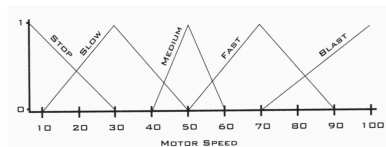
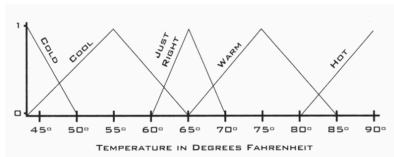
Stap 2 kies labels voor  $X$  en  $Y$

$X$	“Cold”, “Cool”, “Just Right”, “Warm”, en “Hot”
$Y$	“Stop”, “Slow”, “Medium”, “Fast”, en “Blast”

Stap 3 kies vaagverzamelingen voor de gekozen labels



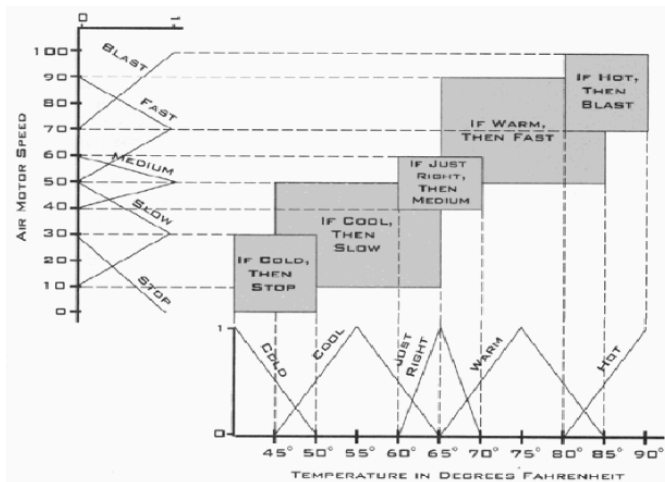
Stap 3 kies vaagverzamelingen voor de gekozen labels



Stap 4 leg de regels vast

- If temperature is ***cold*** then motor speed is ***stop***
- If temperature is ***cool*** then motor speed is ***slow***
- If temperature is ***just right*** then motor speed is ***medium***
- If temperature is ***warm*** then motor speed is ***fast***
- If temperature is ***hot*** then motor speed is ***blast***

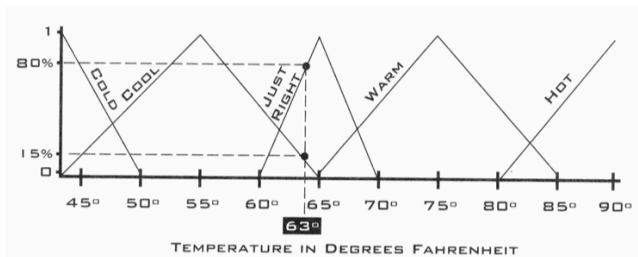
# Fuzzy control: de werking uitgelegd



# Fuzzy control: de werking uitgelegd

Step 5 bereken de activatiegraad van elke regel voor de concrete invoerwaarde

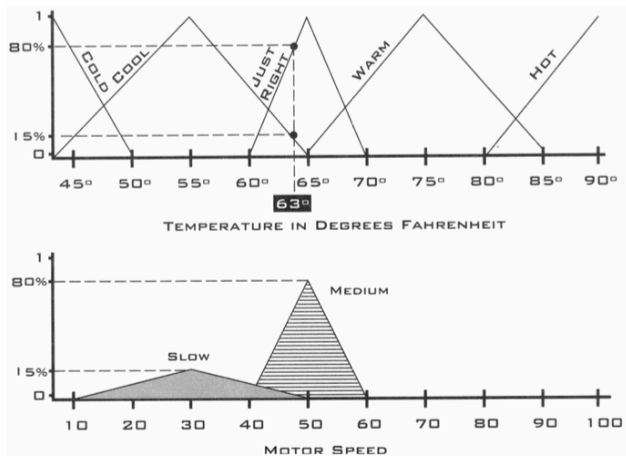
- If temperature is **cool** then motor speed is **slow**
- If temperature is **just right** then motor speed is **medium**



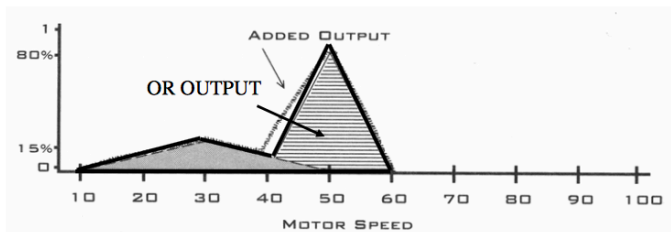


# Fuzzy control: de werking uitgelegd

Stap 6 wijzig het uitvoerlabel van elke geactiveerde regel volgens de activatiegraad (hier met het product)

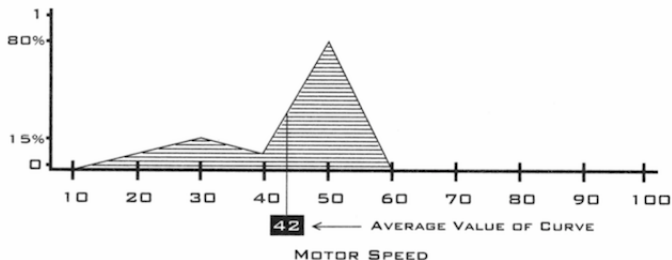


Stap 7 neem de unie van de resultaten per regel



“OR OUTPUT” verwijst naar de unie operator op basis van het maximum, en “ADDED OUTPUT” naar de unie operator op basis van de probabilistische som

Stap 8 zet de vage uitvoer om naar een precieze waarde (“defuzzificatie”)



Enkele populaire defuzzificatie-operatoren:

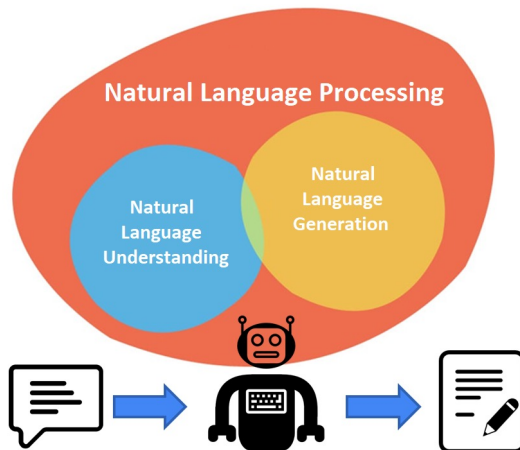
- ▶ Zwaartepuntmethode (centroid defuzzification)

$$D(A) = \frac{\int_{-\infty}^{+\infty} x \cdot A(x) dx}{\int_{-\infty}^{+\infty} A(x) dx}$$

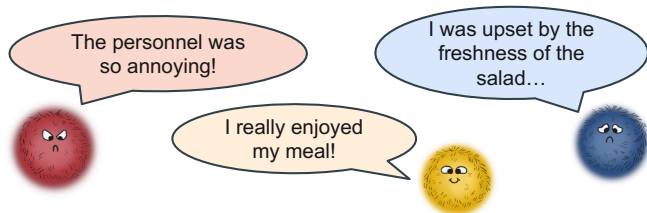
- ▶ Oppervlaktemethode (bisector defuzzification)

$$\int_{-\infty}^{D(A)} A(x) dx = \int_{D(A)}^{+\infty} A(x) dx$$

- ▶ Middelste maximale waarde (middle of maximum defuzzification)

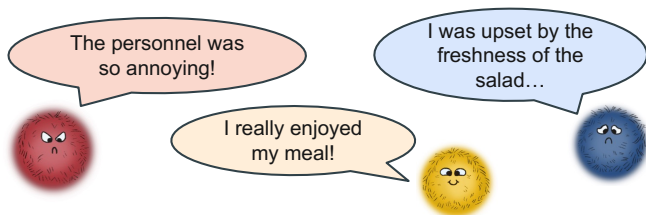


# Toepassing in natuurlijke taal verwerking: emotiedetectie



- ▶ Doel: automatisch bepalen welke emoties in tekst voorkomen en hoe sterk deze emoties zijn
- ▶ Gebruikt in o.m. customer satisfaction analysis, detectie van cyberbullying, ...

# Toepassing in natuurlijke taal verwerking: emotiedetectie



- ▶ Meestal opgelost met deep learning technieken: hoge correctheid, lage transparantie.
- ▶ Hier: ontwerp van een algoritme dat neurale netwerken (voor de interne voorstelling van tekst) en vaaglogica (voor het redeneerproces) combineert
- ▶ Bron: Olha Kaminska, UGent doctoraatsproefschrift 2023

# Voorbeeld: detectie van boosheid

	ID	Tweet	Class
0	2017-En-10264	@xandraaa5 @amayaallyn6 shut up hashtags are cool #offended	2
2	2017-En-11383	Lo! Adam the Bull with his fake outrage...	1
9	2017-En-10849	Follow up. Follow through. Be #relentless. #success	0
10	2017-En-10640	@JustinRow10 dude the new madden 17? Haha	0
12	2017-En-10940	Lo! little things like that make me so angry x	2
13	2017-En-11456	One more day tiff.. One more day!	0

- ▶ Training set: een verzameling (Engelstalige) posts op X, telkens met een indicatie of er boosheid uit de post spreekt (gaande van 0 - geen boosheid, tot 3 - extreme boosheid)



# Voorbeeld: detectie van boosheid

	ID	Tweet	Class
0	2017-En-10264	@xandraaa5 @amayaalyn6 shut up hashtags are cool #offended	2
2	2017-En-11383	LoI Adam the Bull with his fake outrage...	1
9	2017-En-10849	Follow up. Follow through. Be #relentless. #success	0
10	2017-En-10640	@JustinRow10 dude the new madden 17? Haha	0
12	2017-En-10940	LoI little things like that make me so angry x	2
13	2017-En-11456	One more day tiff.. One more day!	0

- ▶ Training set: een verzameling (Engelstalige) posts op X, telkens met een indicatie of er boosheid uit de post spreekt (gaande van 0 - geen boosheid, tot 3 - extreme boosheid)
- ▶ Doel: voor een nieuwe post correct de “anger class” (0, 1, 2 of 3) proberen voorspellen, én uitleggen waarom

# Voorbeeld: detectie van boosheid

	ID	Tweet	Class
0	2017-En-10264	@xandraaa5 @amayaalyn6 shut up hashtags are cool #offended	2
2	2017-En-11383	Lo! Adam the Bull with his fake outrage...	1
9	2017-En-10849	Follow up. Follow through. Be #relentless. #success	0
10	2017-En-10640	@JustinRow10 dude the new madden 17? Haha	0
12	2017-En-10940	Lo! little things like that make me so angry x	2
13	2017-En-11456	One more day tiff.. One more day!	0

- ▶ Training set: een verzameling (Engelstalige) posts op X, telkens met een indicatie of er boosheid uit de post spreekt (gaande van 0 - geen boosheid, tot 3 - extreme boosheid)
- ▶ Doel: voor een nieuwe post correct de “anger class” (0, 1, 2 of 3) proberen voorspellen, én uitleggen waarom
- ▶ Methode: we voorspellen klasse  $x$  indien *de meeste gelijkaardige* training posts tot die klasse behoren; als verklaring geven we dan deze meest gelijkende posts.

- ▶ Word embeddings zijn voorstellingen van woorden (en bij uitbreiding van zinnen, paragrafen, ...) als *reële vectoren*

- ▶ Word embeddings zijn voorstellingen van woorden (en bij uitbreiding van zinnen, paragrafen, ...) als *reële vectoren*
- ▶ Eenvoudigste word embedding: one-hot encoding
  - ▶ Dimensie word embeddings = aantal verschillende woorden in het woordenboek
  - ▶ Voorstelling  $m^e$  woord: vector met overall nullen en met 1 op de  $m^e$  positie
  - ▶ Voorbeeld: Woordenboek = {'ik', 'ben', 'blij', 'boos', 'misnoegd'}

$$\text{ik} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ben} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{blij} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{boos} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{misnoegd} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

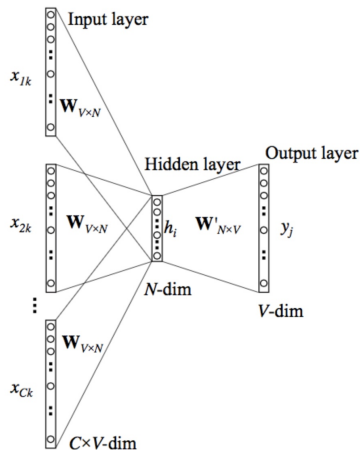
- ▶ Word embeddings zijn voorstellingen van woorden (en bij uitbreiding van zinnen, paragrafen, ...) als *reële vectoren*
- ▶ Eenvoudigste word embedding: one-hot encoding
  - ▶ Dimensie word embeddings = aantal verschillende woorden in het woordenboek
  - ▶ Voorstelling  $m^e$  woord: vector met overall nullen en met 1 op de  $m^e$  positie
  - ▶ Voorbeeld: Woordenboek = {'ik', 'ben', 'blij', 'boos', 'misnoegd'}

$$\text{ik} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{ben} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \text{blij} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \text{boos} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} \text{misnoegd} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

- ▶ Nadelen one-hot encoding: hoge dimensie, geen verbanden tussen woorden

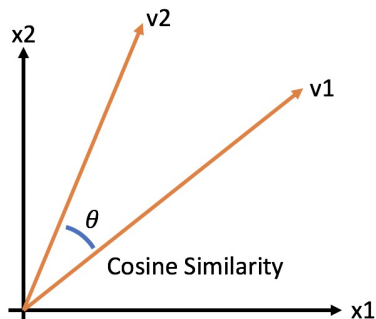
- ▶ We zullen trachten word embeddings te leren zo dat:
  - ▶ de dimensie van de vectoren niet te hoog is
  - ▶ woorden die op elkaar lijken dicht bij elkaar liggen
- ▶ Neurale netwerken zijn hiervoor uitermate geschikt!
- ▶ Bekendste model: Word2Vec by Google (Mikolov, 2013)
  - ▶ Methode: vertrekkend van een zeer grote tekstcollectie leert het neurale netwerk ontbrekende woorden in een zin in te vullen
  - ▶ De geleerde gewichten vormen dan de word embeddings

# Word2Vec: voorbeeld



["Have", "a", ..., "day"]  $\rightarrow$  "nice"

# Vergelijken van woordvectoren: cosinussimilariteit



- ▶ “Gelijkenis” tussen twee woorden = cosinus van de hoek  $\theta$  tussen de twee bijhorende vectoren ( $\cos \theta = \frac{v_1 \cdot v_2}{\|v_1\| \cdot \|v_2\|}$ )
- ▶ Kan worden omgevormd tot een getal tussen 0 en 1:  
$$Sim(x, y) = \frac{1 + \cos \theta}{2}$$



Gegeven: een training set  $T$  van posts met bijhorende “anger classes” (0, 1, 2 of 3), en een nieuwe post  $y$

1. Preprocessing (optioneel): verwijderen van stopwoorden, symbolen, vervangen van emojis door tekst, ...
2. Genereren van word embeddings (bijvoorbeeld met Google's GenSim library die vectoren van dimensie 300 oplevert) voor elk woord uit een post
3. Genereren van embedding voor de post = gemiddelde vector van de woorden die erin voorkomen

Gegeven: een training set  $T$  van posts met bijhorende “anger classes” (0, 1, 2 of 3), en een nieuwe post  $y$

4. Voor elke anger klasse  $X$ , bereken

$$Score(X) = \min_{x \in T} \{ \max(1 - Sim(x, y), X(x)) \}$$

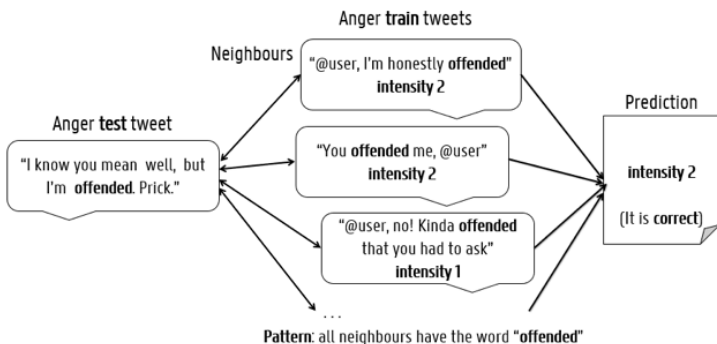
met  $X(x) = 1$  als  $x \in X$  en  $X(x) = 0$  als  $x \notin X$

(dit is de waarheidswaarde van de bewering “elke post die lijkt op  $y$  behoort tot klasse  $X$ ”)

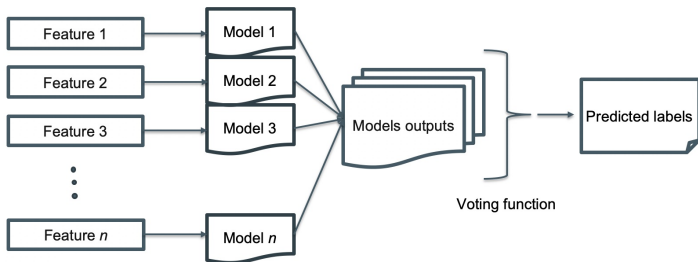
5. Wijs de post toe aan de klasse  $X$  waarvoor  $Score(X)$  maximaal is

# Verklaring van de voorspelling

Als verklaring toont het algoritme de training posts die het meest lijken op de test post

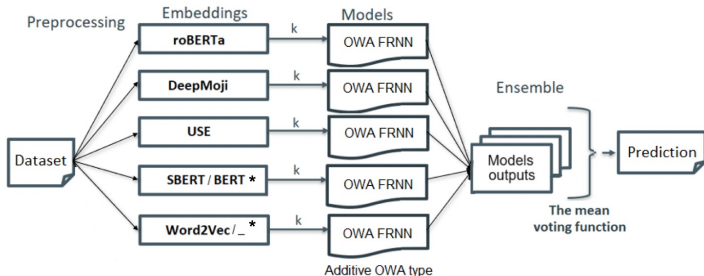


# Verbeteren van de correctheid: ensembles



- ▶ Er bestaat een breed gamma aan word embeddings; elk heeft zijn specifieke voor- en nadelen
- ▶ Een ensemble buit de complementariteit van de individuele modellen (word embeddings) uit

# Verbeteren van de correctheid: ensembles



\* **Anger**: BERT+Word2Vec  
**Joy**: SBERT+BERT  
**Sadness**: SBERT  
**Fear**: SBERT+Word2Vec

2<sup>th</sup> place with averaged  
PCC score **0.6544**

Top 1 PCC score: 0.695

- ▶ Theorie:
  - ▶ wiskundige voorstelling van vage kwantoren
  - ▶ maten voor de evaluatie van gelijkaardigheid
  - ▶ integratie van probabilistische en vage informatie
  - ▶ ...
- ▶ Toepassingen:
  - ▶ machine learning
  - ▶ natural language processing: detectie van ironie, hate speech, ...
  - ▶ aanbevelingssystemen

- ▶ Theorie:
  - ▶ wiskundige voorstelling van vage kwantoren
  - ▶ maten voor de evaluatie van gelijkaardigheid
  - ▶ integratie van probabilistische en vage informatie
  - ▶ ...
- ▶ Toepassingen:
  - ▶ machine learning
  - ▶ natural language processing: detectie van ironie, hate speech, ...
  - ▶ aanbevelingssystemen

## Bedankt voor jullie aandacht!