

Beeldherkenning met Neurale netwerken

Prof. dr. Yvan Saeys (yvan.saeys@ugent.be)

saeyslab.sites.vib.be

Even mezelf voorstellen



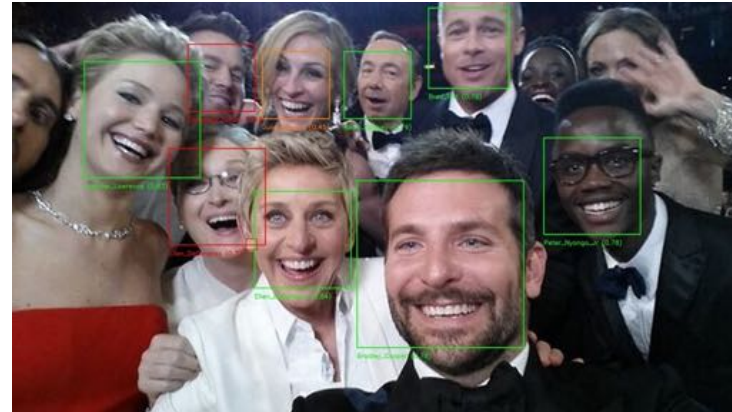
AI FLANDERS
BUILDING OUR DIGITAL FUTURE

Enkele voorbeelden van AI in ons dagelijks leven

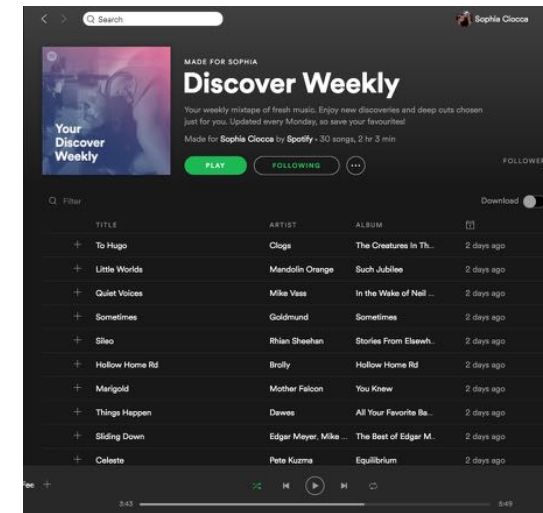
Slimme planners



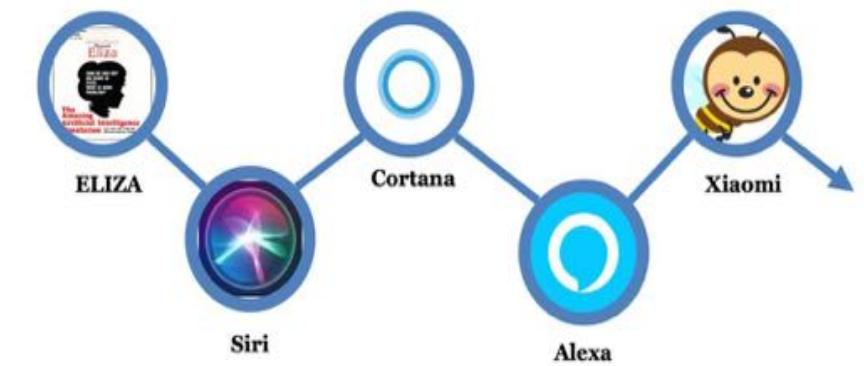
Gezichtsherkenning



Aanbevelingssystemen



Interactieve interfaces



Medische assistenten

Computer learns to detect skin cancer more accurately than doctors

Artificial intelligence machine found 95% of melanomas in study compared to 86.6% for dermatologists



Zelfrijdende auto's



Generatieve AI

“An astronaut riding a horse in a photorealistic style”



AI probeert niet om menselijke intelligentie na te bootsen maar om systemen te bouwen die in staat zijn om problemen op te lossen die intelligentie vereisen

Ingrediënten van AI

- Invoer – verwerking – uitvoer/interactie met de omgeving
 - Spraak- en taalverwerking
 - Computervisie, robotica
- Redeneren en plannen in de echte wereld
 - Wereldmodel, achtergrondinformatie, “gezond verstand”
 - Nieuwe problemen oplossen, plannen maken, optimale beslissingen nemen
 - Omgaan met onzekerheid en onvoorziene omstandigheden
- Leren / aanpassen
 - Geheugen (fouten onthouden en eruit leren)
 - Simulatie (verschillende mogelijke scenario's overlopen)

Ingrediënten van AI

- Invoer – verwerking – uitvoer/interactie met de omgeving
 - Spraak- en taalverwerking
 - Computervisie, robotica
- Redeneren en plannen in de echte wereld
 - Wereldmodel, achtergrondinformatie, “gezond verstand”
 - Nieuwe problemen oplossen, plannen maken, optimale beslissingen nemen
 - Omgaan met onzekerheid en onvoorziene omstandigheden
- Leren / aanpassen
 - Geheugen (fouten onthouden en eruit leren)
 - Simulatie (verschillende mogelijke scenario's overlopen)

Ingrediënten van AI

- Invoer – verwerking – uitvoer/interactie met de omgeving
 - Spraak- en taalverwerking
 - Computervisie, robotica
- Redeneren en plannen in de echte wereld
 - Wereldmodel, achtergrondinformatie, “gezond verstand”
 - Nieuwe problemen oplossen, plannen maken, optimale beslissingen nemen
 - Omgaan met onzekerheid en onvoorziene omstandigheden
- **Leren / aanpassen**
 - **Geheugen (fouten onthouden en eruit leren)**
 - **Simulatie (verschillende mogelijke scenario's overlopen)**

Logica

Datastructuren
en algoritmen

Artificiële intelligentie

=

Kennisvoorstellingen

+

Afleiden van nieuwe kennis

Neurale
netwerken

Statistische
modellen

Logica-gebaseerde kennisvoorstellingen

Klassieke logica

P “Socrates is een mens”
P → Q “Alle mensen zijn sterfelijk”



Q “Socrates is sterfelijk”

Deductie (modus ponens)

$$\frac{p \quad p \rightarrow q}{q}$$

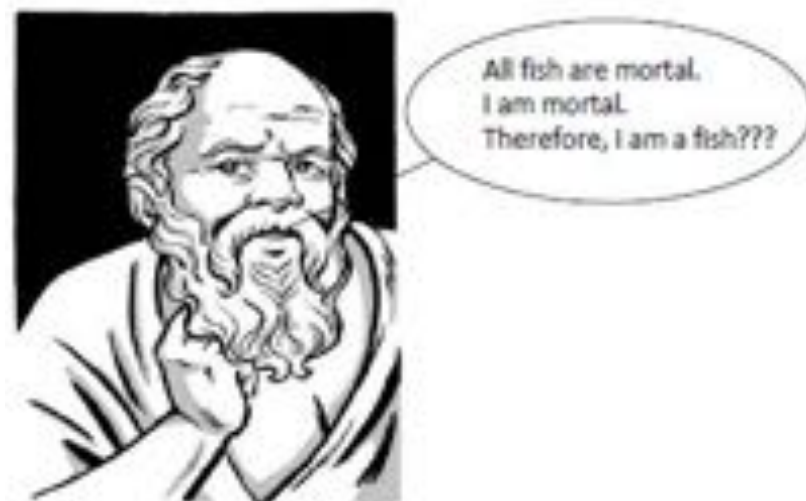
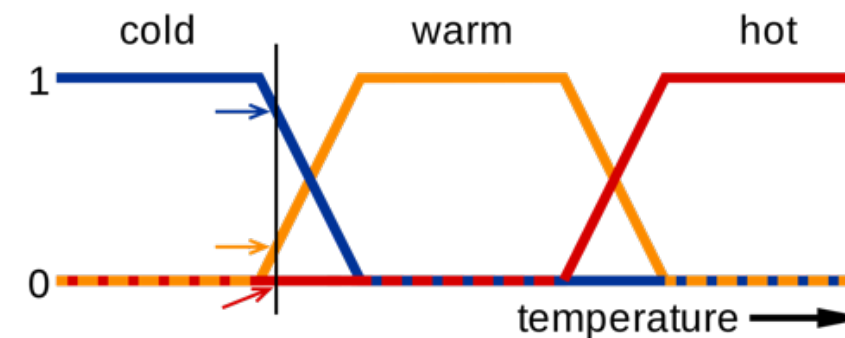


Image credit: Xia Jiang, Richard E. Neapolitan

Vaaglogica



IF temperature IS very cold THEN fan_speed is stopped
IF temperature IS cold THEN fan_speed is slow
IF temperature IS warm THEN fan_speed is moderate
IF temperature IS hot THEN fan_speed is high

Uitbreiding van klassieke logica om met vaagheid en onzekerheid om te gaan

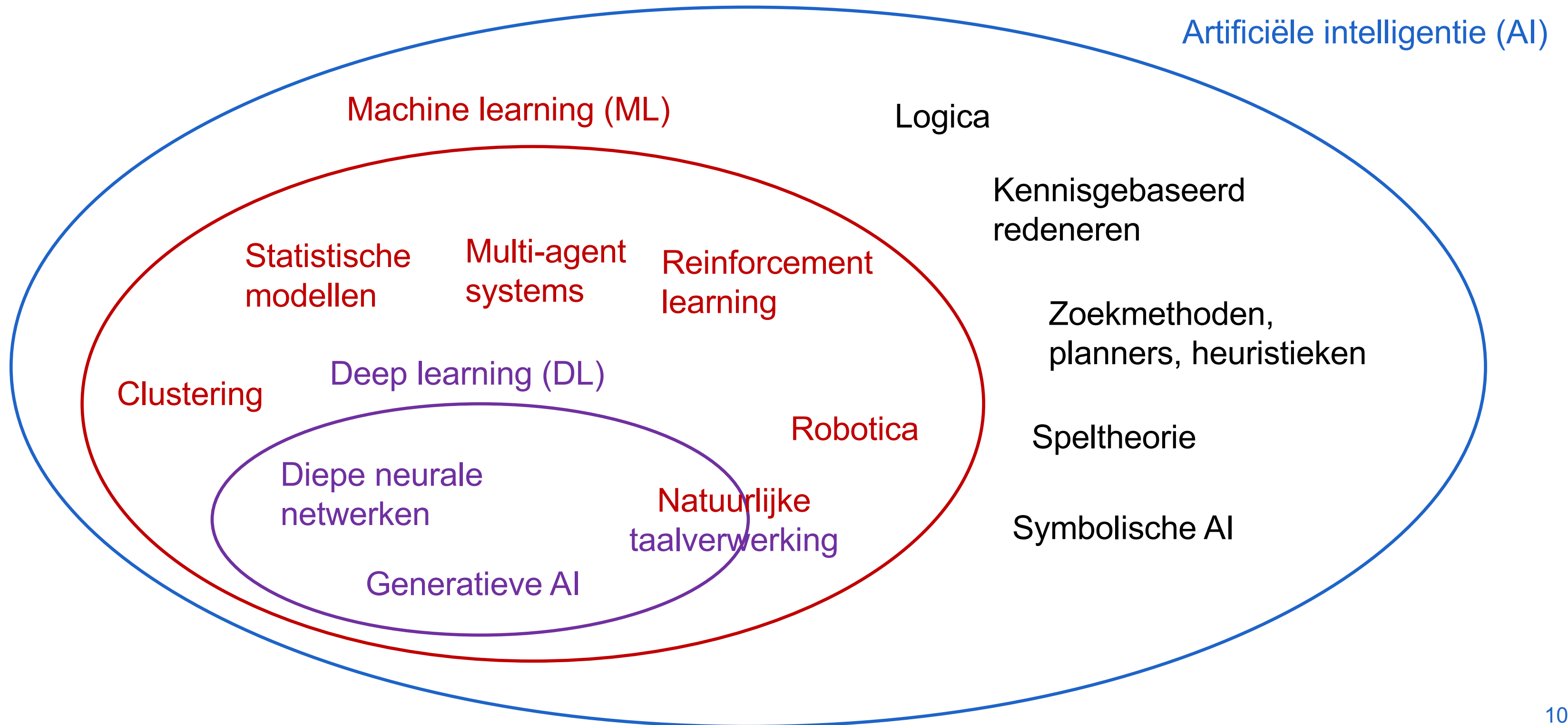


Prof. Etienne Kerre



Prof. Chris Cornelis

Disciplines binnen Artificiële intelligentie



Artificiële intelligentie: een interdisciplinair domein

Wiskunde

Filosofie

Statistiek

Ethiek

Informatica

Logica

Electronica

Signaalverwerking

Taal- en
Spraak
verwerking

Media en
communicatie

Psychologie

Neurologie

Mechatronica

Linguïstiek

Rechten

Economie

Veel methoden uit

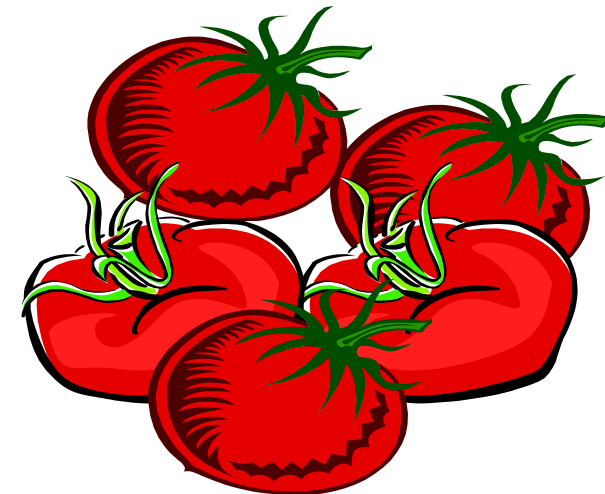
Artificiële intelligentie

zijn eigenlijk

eenvoudige wiskunde

(functies, afstanden, kansberekening)

**Machine
Learning
=
Leren uit
voorbeelden**

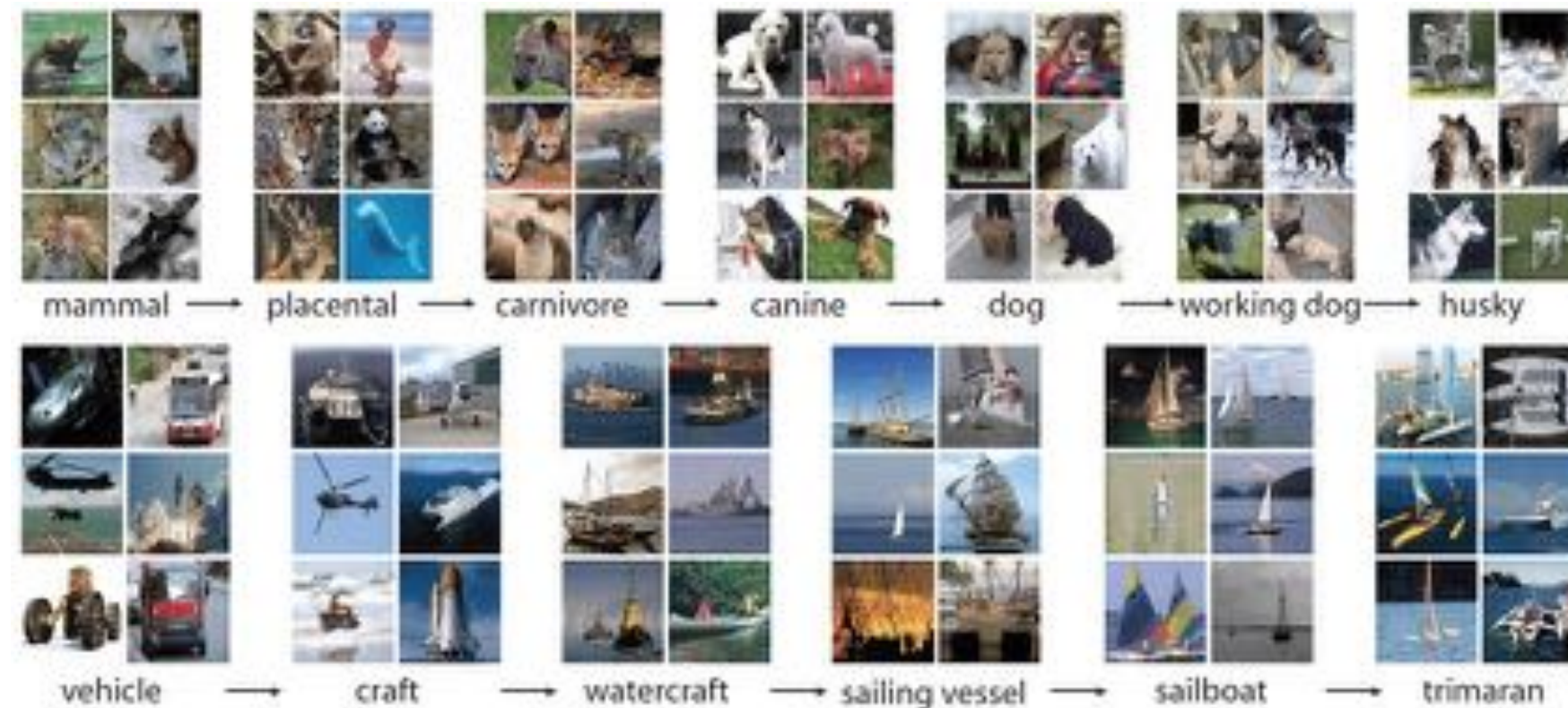


Machine Learning: een nieuwe manier om computers te programmeren

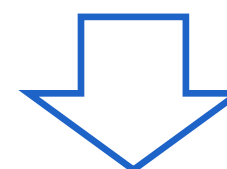
Klassieke manier van programmeren
"hoe"

Programmeren met machine learning
"wat"

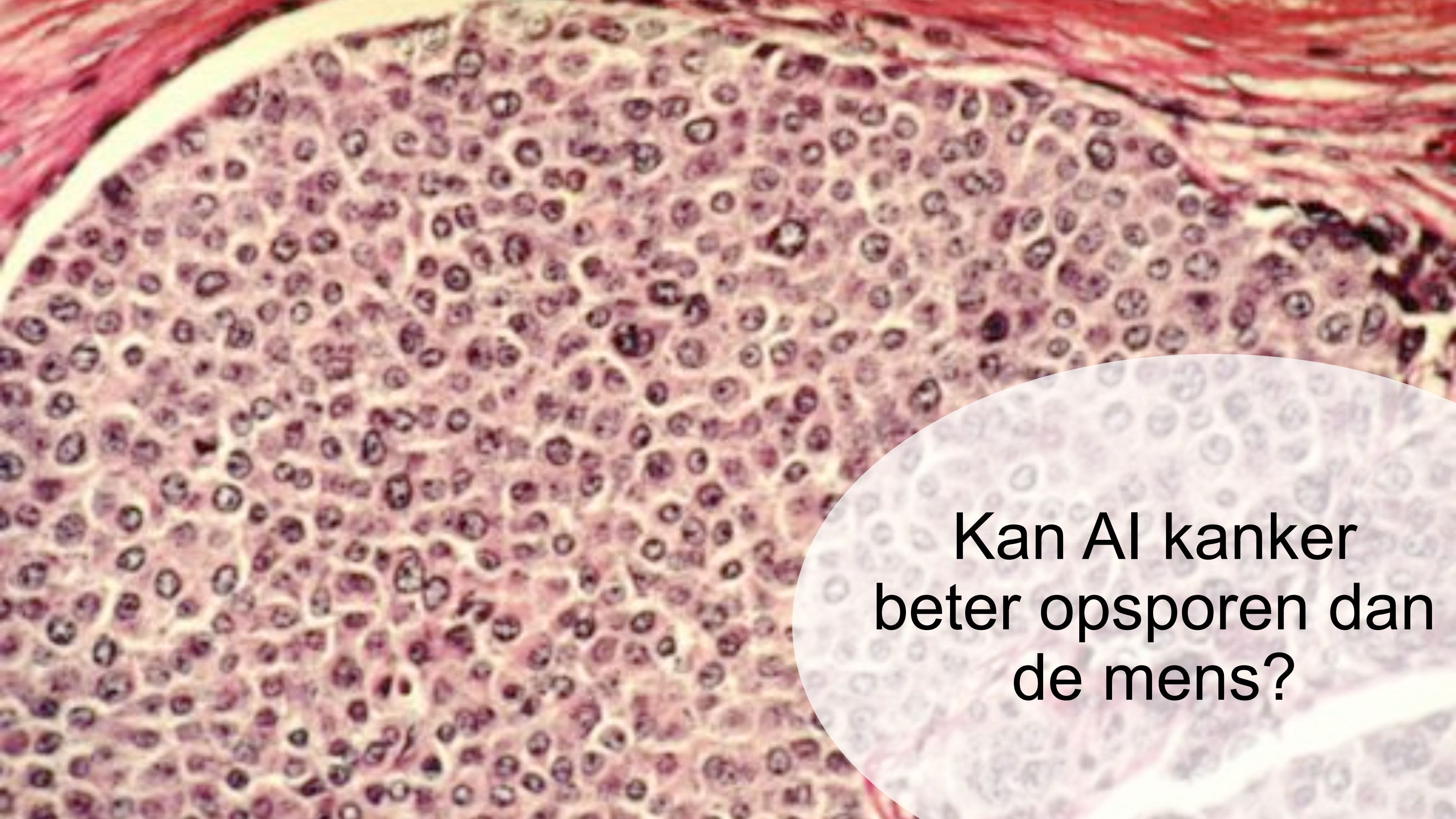
```
1 def dijkstra(graph, src, dest, visited=[], distances={}, predecessors={}):
2     """ calculates a shortest path tree routed in src
3     """
4     if src not in graph:
5         raise TypeError("The root of the shortest path tree cannot be found")
6     if dest not in graph:
7         raise TypeError("The target of the shortest path cannot be found")
8     if src == dest:
9         path = []
10        pred = dest
11        while pred != None:
12            path.append(pred)
13            pred = predecessors.get(pred, None)
14        readable_path = path[0]
15        for index in range(1, len(path)): readable = path[index]+'--->'+readable
16        print('shortest path - array: '+str(path))
17        print("path: "+readable+", cost="+str(distances[dest]))
18    else:
19        if not visited:
20            distances[src]=0
21        for neighbor in graph[src]:
22            if neighbor not in visited:
23                new_distance = distances[src] + graph[src][neighbor]
24                if new_distance < distances.get(neighbor, float('inf')):
25                    distances[neighbor] = new_distance
26                    predecessors[neighbor] = src
27        visited.append(src)
28        unvisited=[]
29        for k in graph:
30            if k not in visited:
31                unvisited[k] = distances.get(k, float('inf'))
32        x = min(unvisited, key=unvisited.get)
33        dijkstra(graph, x, dest, visited, distances, predecessors)
34    if __name__ == "__main__":
35        graph = {'A1': {'B1': 1, 'B2': 1.41, 'A2': 1},
36                'A2': {'B1': 1.41, 'B2': 1, 'B3': 1.41, 'A1': 1, 'A3': 1},
37                .....
38        dijkstra(graph, 'A1', 'H9')
39        shortest path - array: ['H9', 'G8', 'F8', 'E7', 'D6', 'D5', 'C4', 'C3', 'B2', 'A1']
40        path: A1---B2---C3---C4---D5---D6---E7---F8---G8---H9, cost=11.459999999999999
```



ImageNet: 1,2 miljoen voorbeelden, 1000 klassen



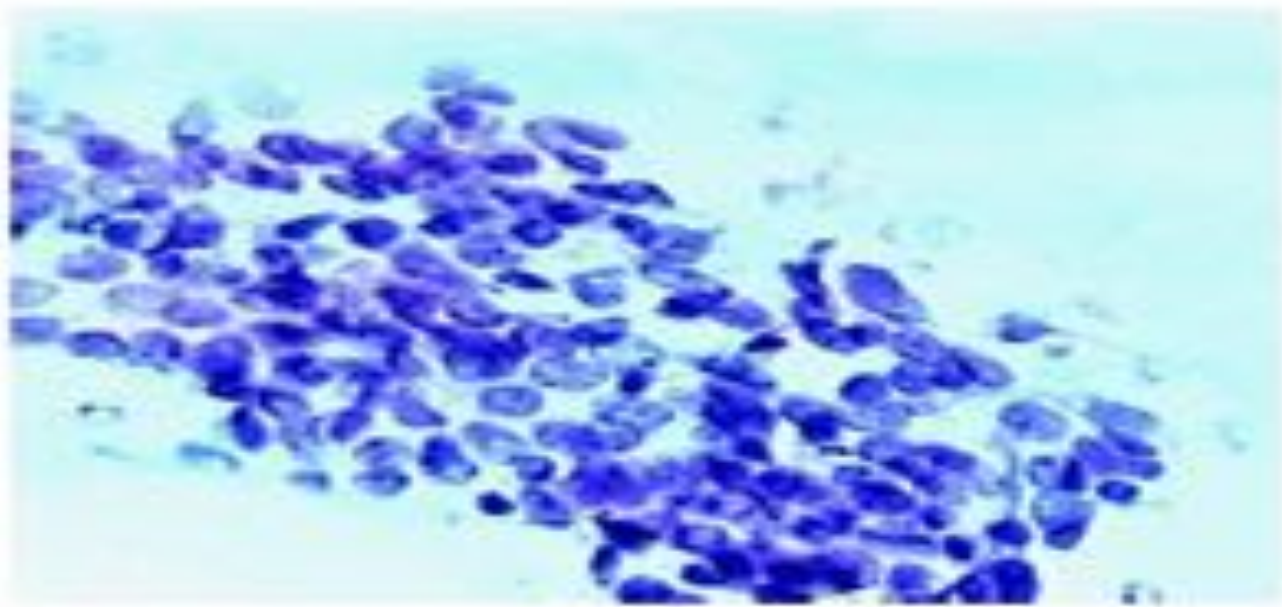
Neuraal netwerk



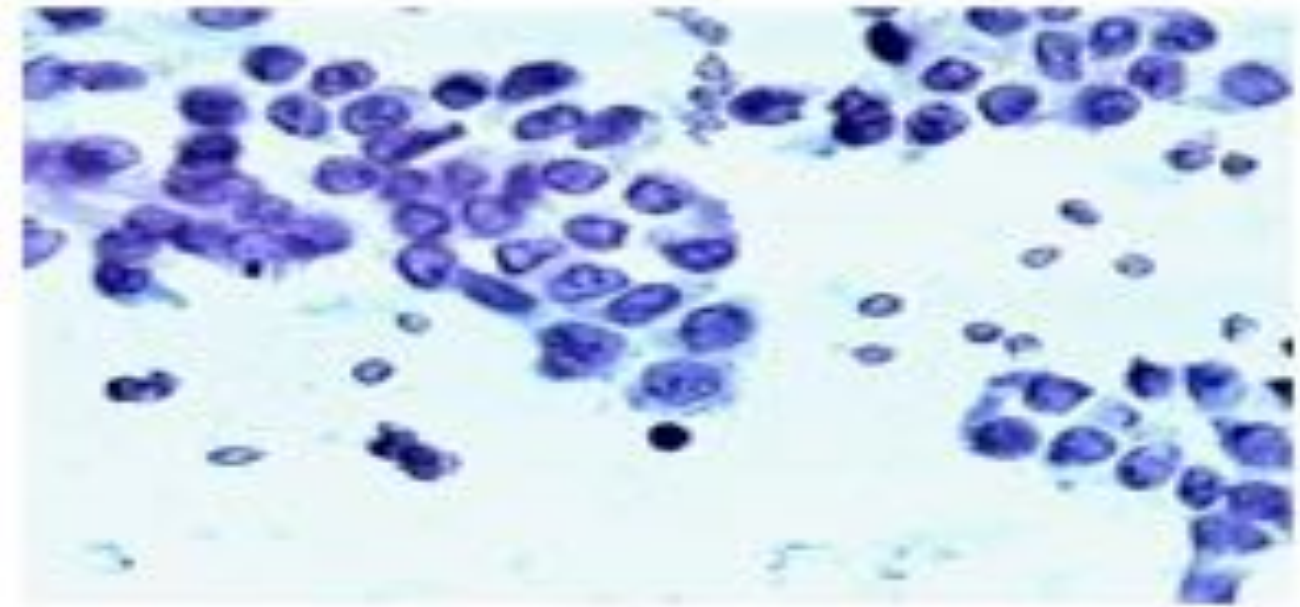
Kan AI kanker
beter opsporen dan
de mens?

Een eenvoudig ML model voor borstkanker

Goedaardige cellen



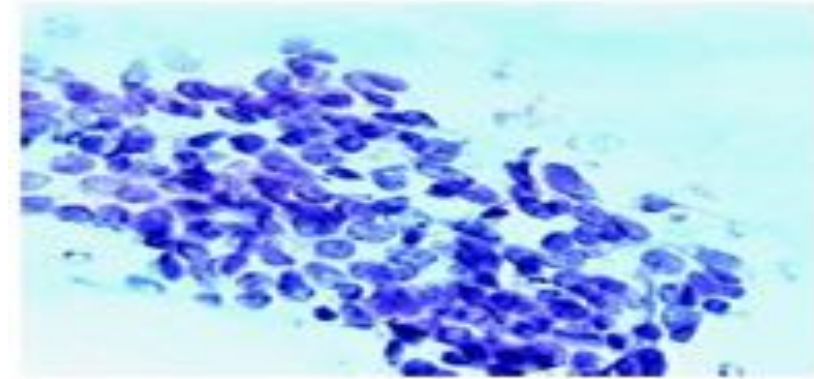
Kwaadaardige cellen



**We kunnen dit modelleren als een
binair classificatieprobleem**

Een eenvoudig ML model voor borstkanker

Patiënten



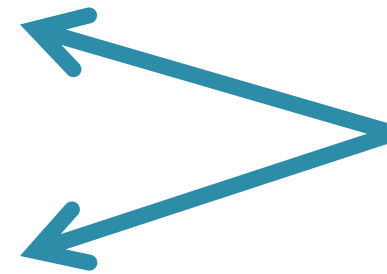
Kenmerkextractie:

Bereken variabelen op basis van beeldverwerkingsalgoritmen: Celvorm, aantal, textuur, rondheid,...



Goedaardig

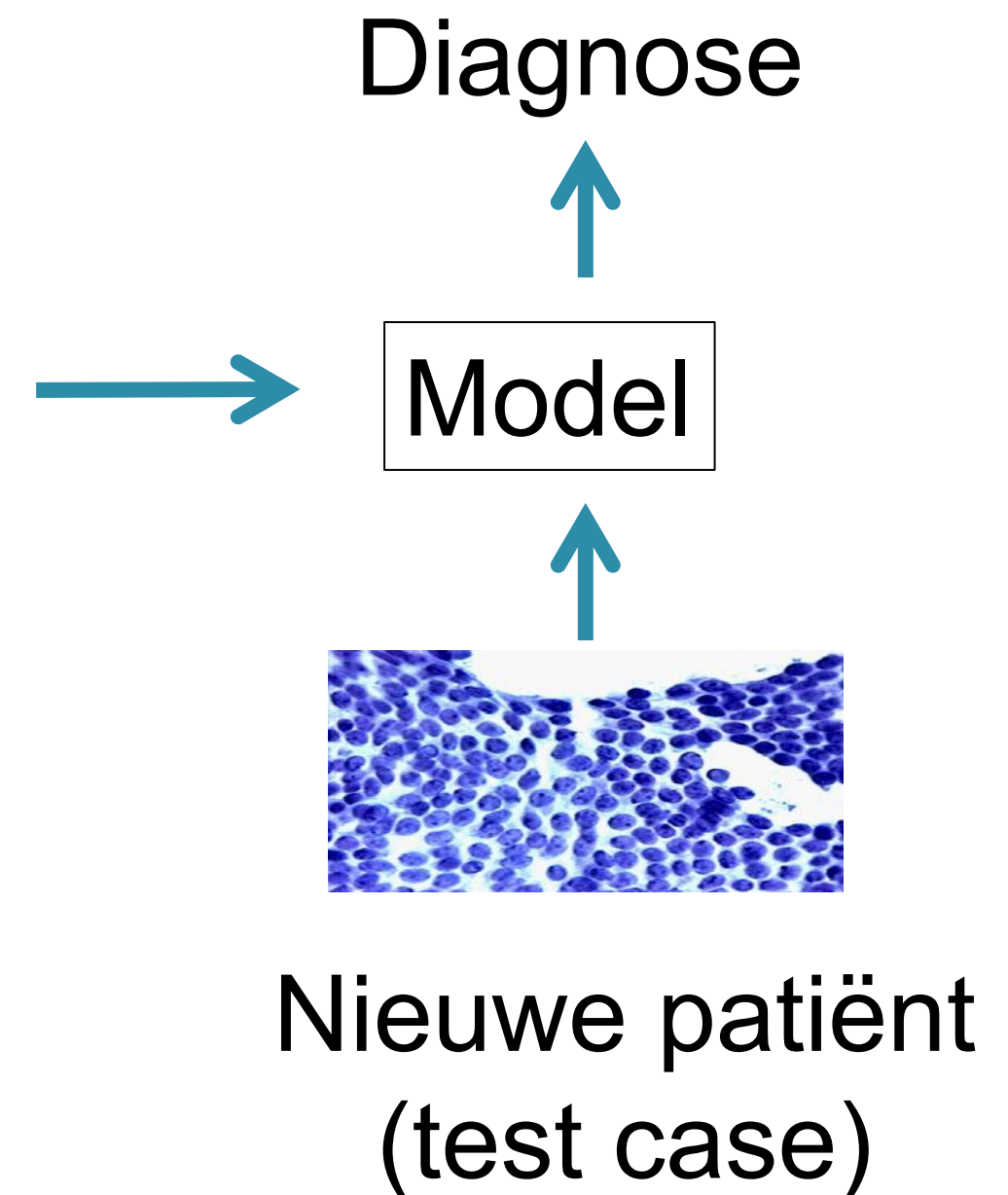
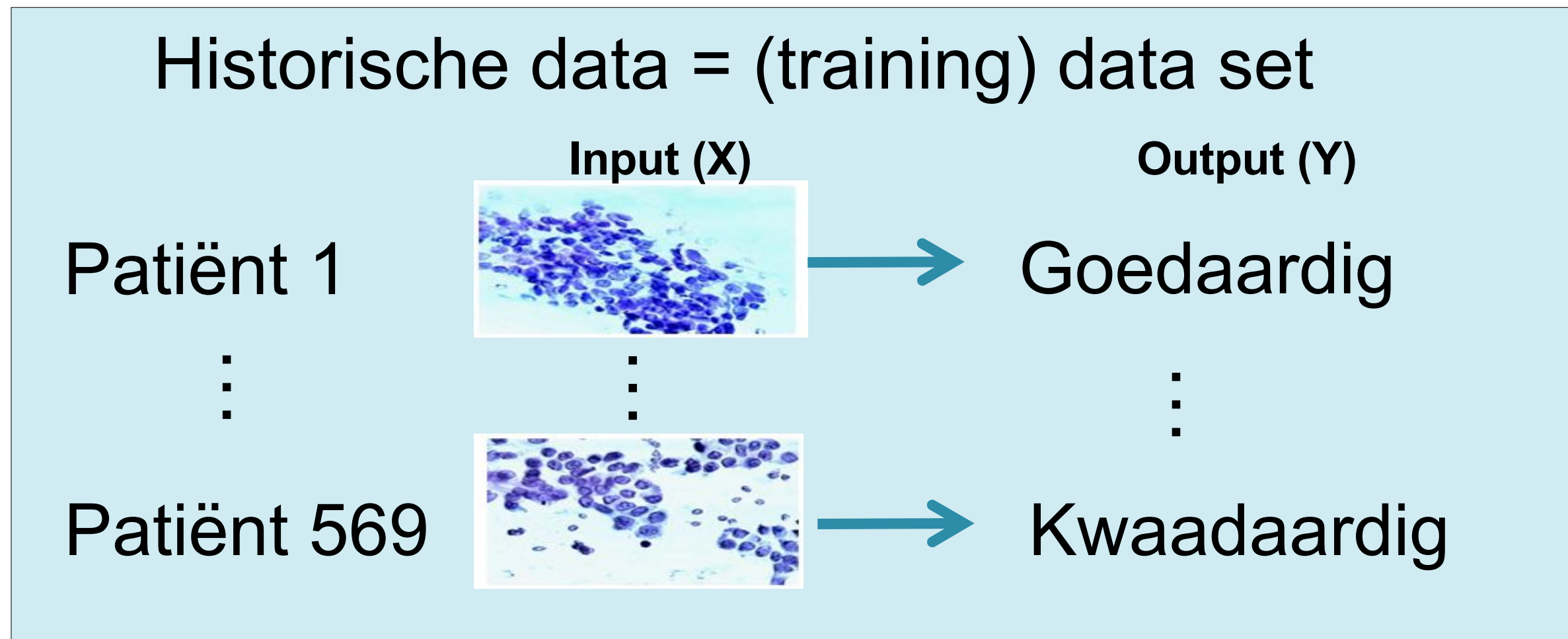
Kwaadaardig



Machine learning model:

Bouw een predictief model en geef output

Machine learning = leren uit voorbeelden

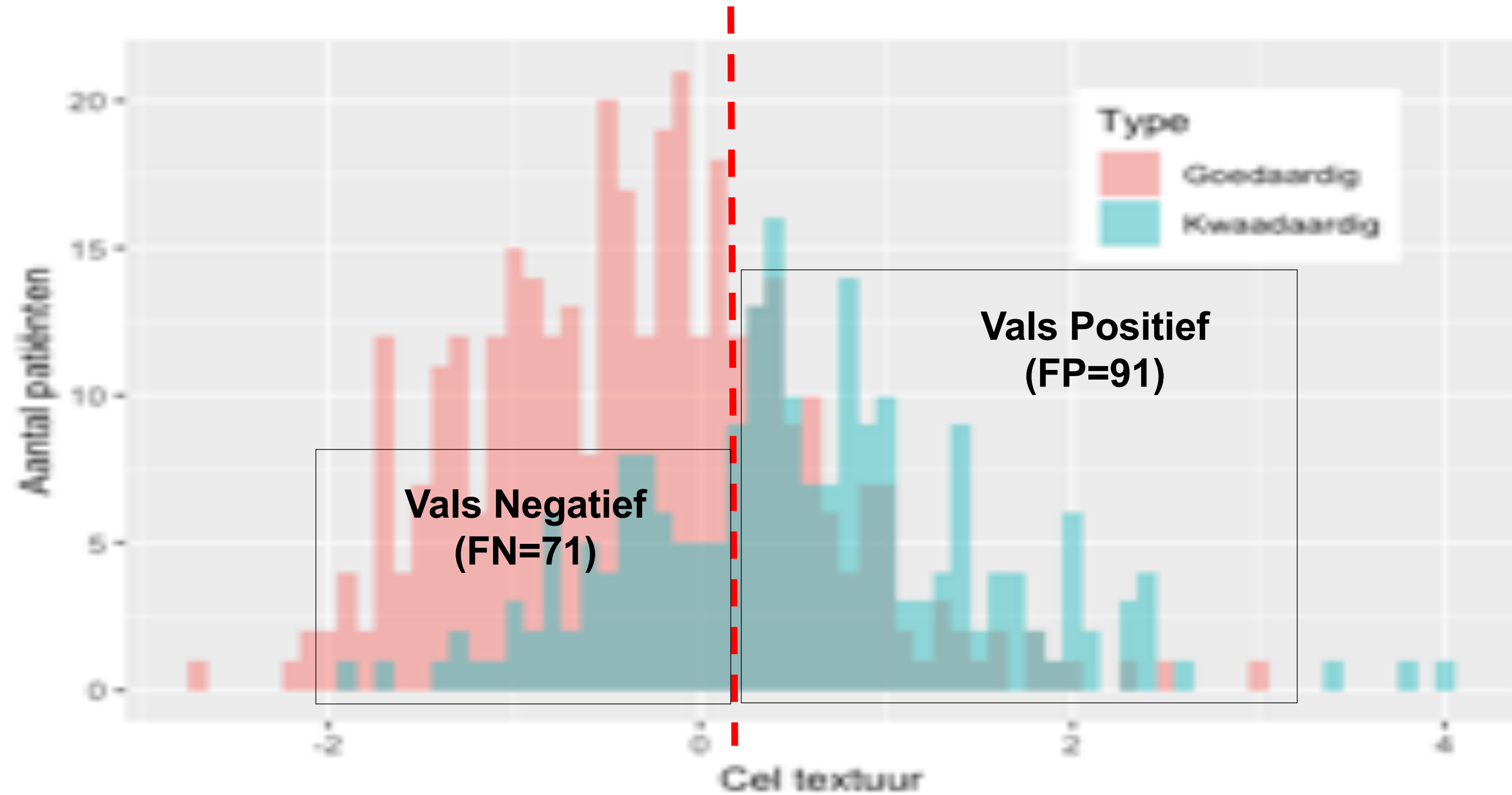


Model = Functie van *input* naar *output*: $Y=f(X)$

Doel van machine learning

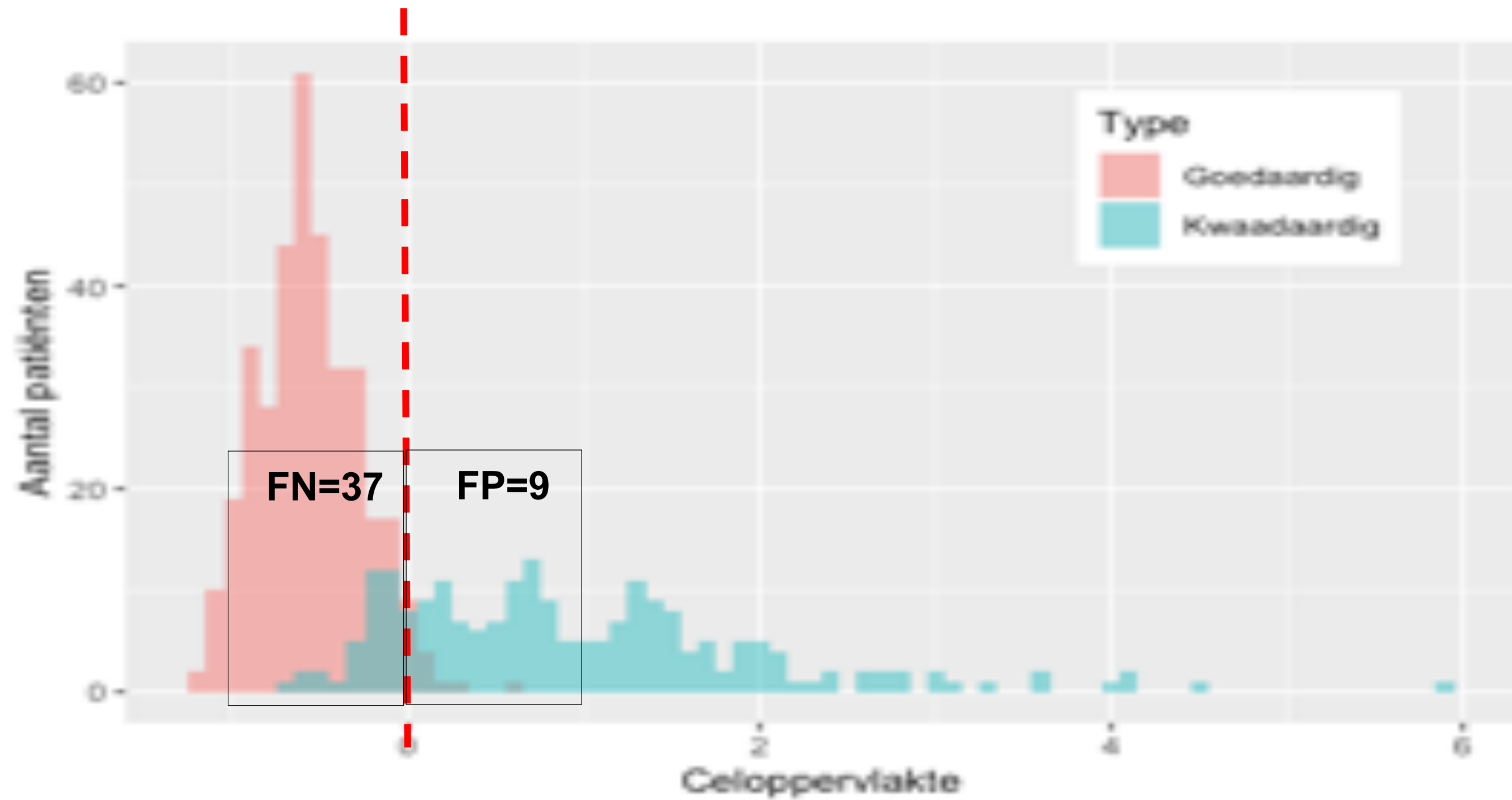
Zo goed mogelijk de *nieuwe* patiënten voorspellen

= generalisatie



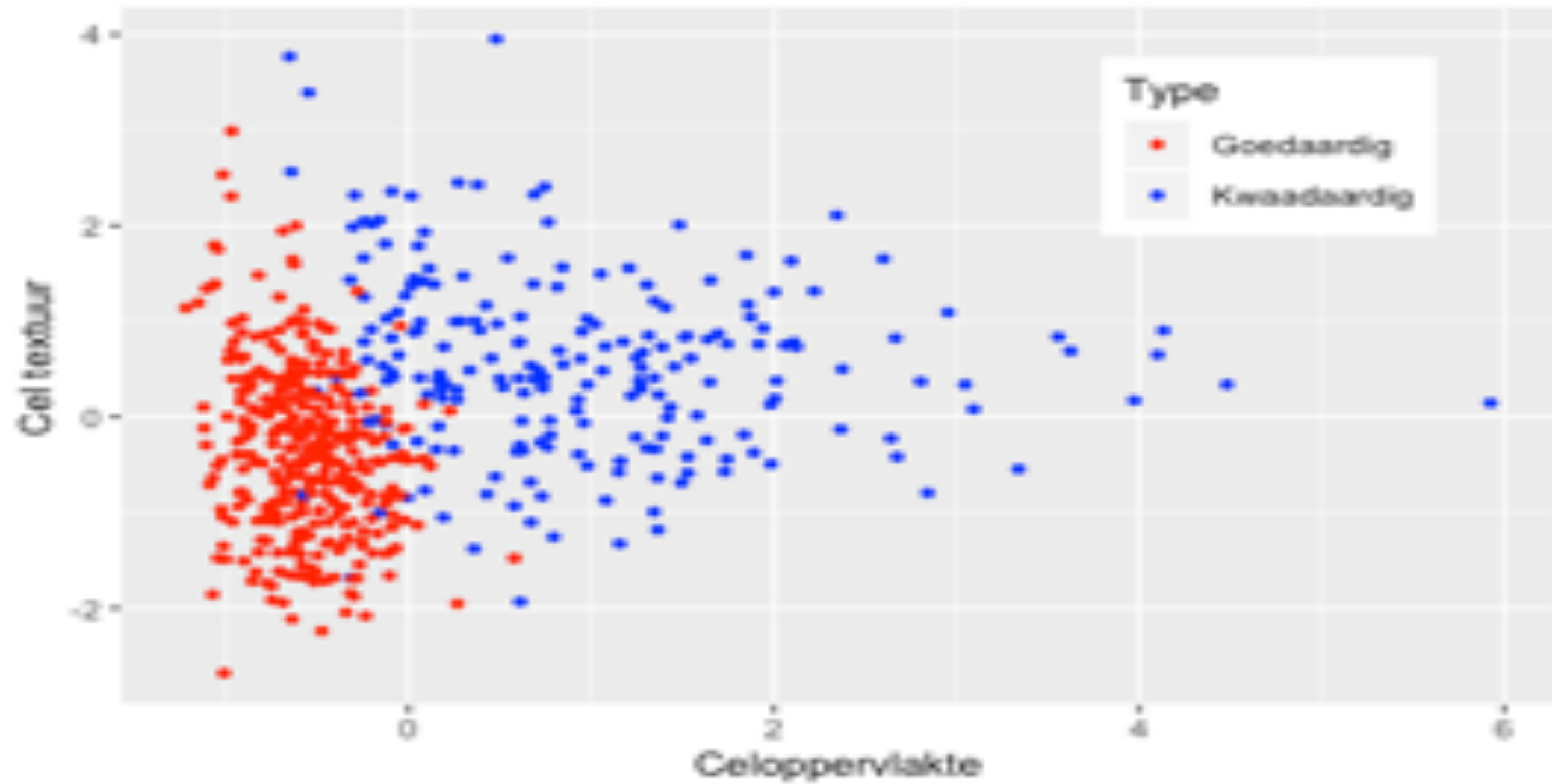
Beslissingsmodel:

ALS Cel textuur < 0.2 DAN Goedaardig ANDERS Kwaadaardig



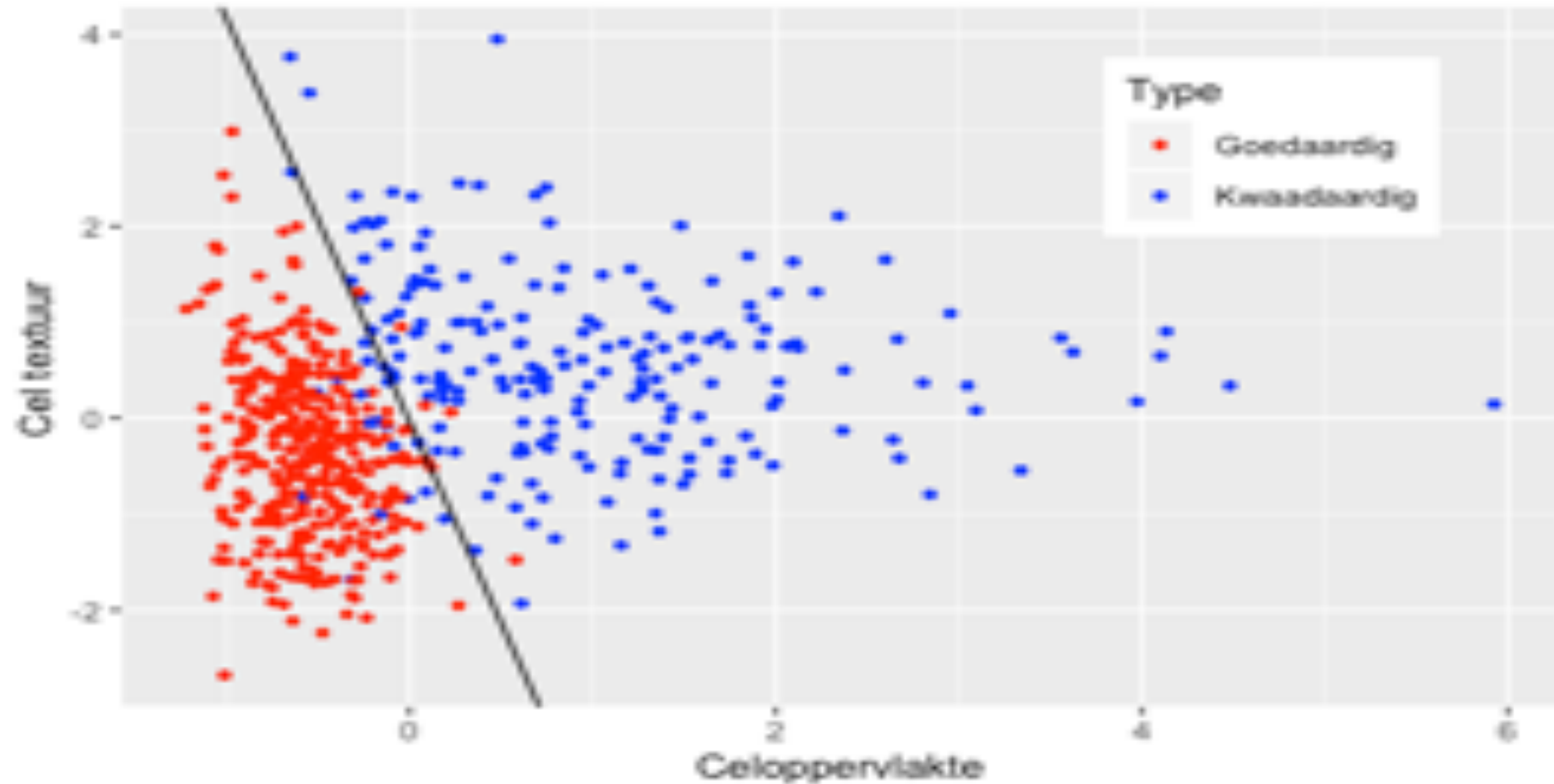
Beslissingsmodel:

ALS Celoppervlakte < 0 **DAN** Goedaardig **ANDERS** Kwaadaardig ²¹



$$f(x) = -4.2x - 0.04$$

FP=5
FN=18

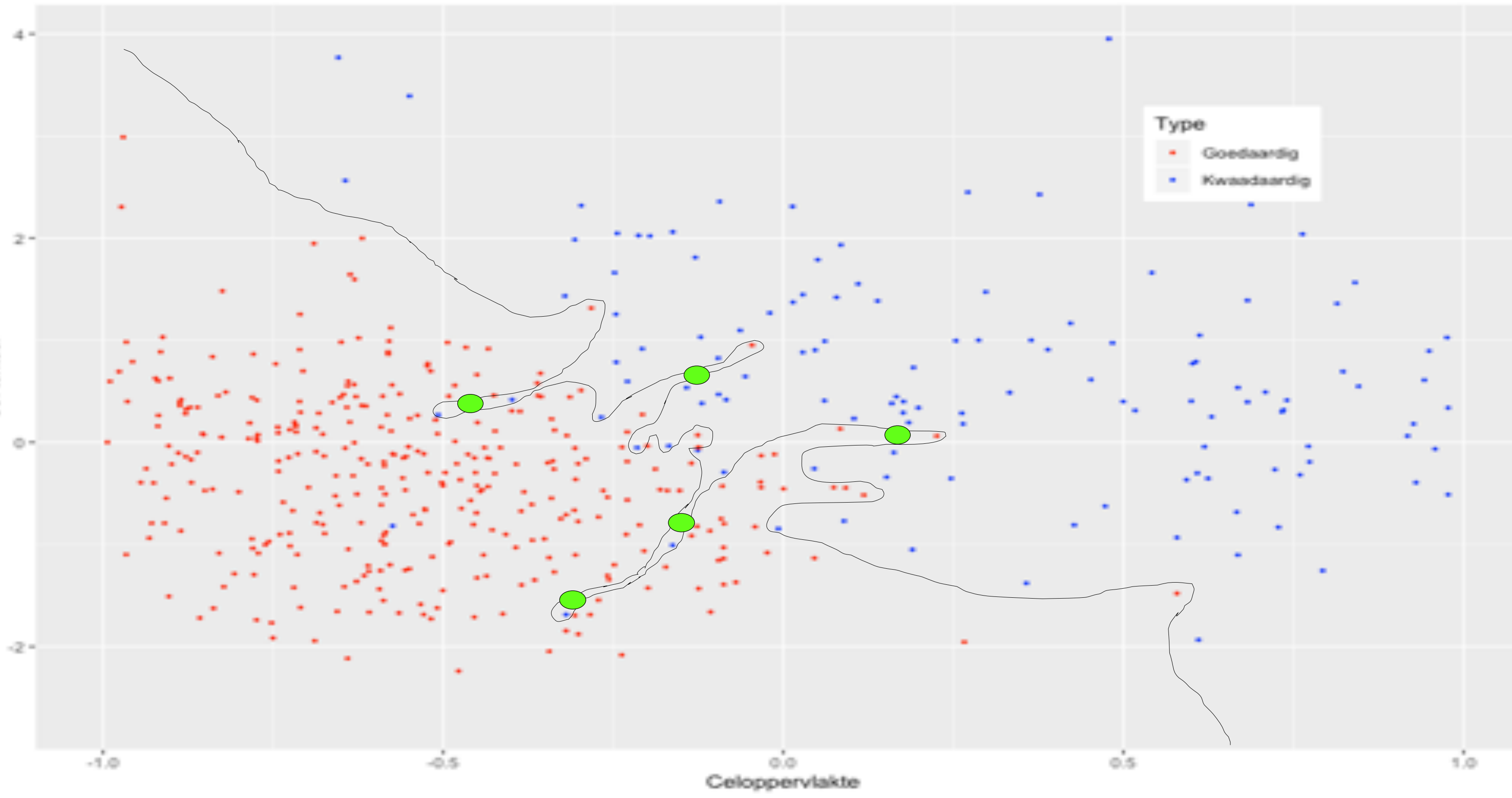


Beslissingsmodel:

ALS Celtextuur < -4.2 Celoppervlakte $- 0.04$ **DAN Goedaardig**

ANDERS Kwaadaardig

Cell texture



In werkelijkheid...

Variabelen

Patiënten

| radius_mean | texture_mean | perimeter_mean | area_mean | smoothness_mean | compactness_mean | concavity_mean | concave_points_mean | symmetry |
|-------------|--------------|----------------|-----------|-----------------|------------------|----------------|---------------------|----------|
| 28.11 | 28.47 | 188.50 | 2499.0 | 0.11420 | 0.25180 | 0.32010 | 0.10950 | 0.1648 |
| 27.42 | 26.27 | 186.90 | 2501.0 | 0.10840 | 0.19680 | 0.26350 | 0.16890 | 0.2061 |
| 27.32 | 21.87 | 182.30 | 2250.0 | 0.10840 | 0.19140 | 0.26710 | 0.18780 | 0.1890 |
| 25.79 | 17.46 | 174.20 | 2010.0 | 0.11490 | 0.20620 | 0.33680 | 0.14130 | 0.1956 |
| 25.22 | 24.95 | 171.50 | 1878.0 | 0.10830 | 0.26610 | 0.23390 | 0.18450 | 0.1829 |
| 24.83 | 21.68 | 165.50 | 1841.0 | 0.10000 | 0.21060 | 0.23100 | 0.14710 | 0.1991 |
| 24.25 | 29.29 | 166.20 | 1791.0 | 0.14470 | 0.29670 | 0.42680 | 0.20120 | 0.2635 |
| 23.51 | 24.27 | 155.30 | 1747.0 | 0.10690 | 0.12830 | 0.25080 | 0.14100 | 0.1797 |
| 21.28 | 26.67 | 158.90 | 1681.0 | 0.11410 | 0.20840 | 0.35230 | 0.16200 | 0.2200 |
| 21.27 | 22.84 | 152.30 | 1686.0 | 0.09439 | 0.11450 | 0.15240 | 0.09702 | 0.1801 |
| 21.21 | 26.97 | 151.50 | 1670.0 | 0.09028 | 0.16820 | 0.29500 | 0.12070 | 0.1909 |
| 21.08 | 28.83 | 152.30 | 1682.0 | 0.09342 | 0.12750 | 0.16760 | 0.10010 | 0.1506 |
| 22.27 | 19.67 | 152.80 | 1509.0 | 0.11260 | 0.27680 | 0.42640 | 0.18210 | 0.2158 |
| 22.01 | 21.90 | 147.20 | 1482.0 | 0.10030 | 0.19540 | 0.24480 | 0.10010 | 0.1824 |
| 21.79 | 28.98 | 147.30 | 1491.0 | 0.09601 | 0.10610 | 0.21090 | 0.10880 | 0.1721 |
| 21.71 | 17.25 | 140.90 | 1346.0 | 0.09364 | 0.08542 | 0.11680 | 0.08465 | 0.1717 |
| 21.61 | 22.28 | 144.40 | 1407.0 | 0.10670 | 0.20870 | 0.28100 | 0.15620 | 0.2162 |
| 21.56 | 22.39 | 142.60 | 1479.0 | 0.11090 | 0.11190 | 0.24390 | 0.13890 | 0.1726 |

30 variabelen opgemeten per patiënt:

- elke patiënt is een punt in een 30-dimensionale ruimte
- **Linear model in 30 dimensies:** $Y = w_0 + w_1 x_1 + \dots + w_{30} x_{30}$

Machine learning = fun met functies

Weersvoorspelling (bvb temperatuur volgende dag)

$$X = \mathbb{R}^n$$

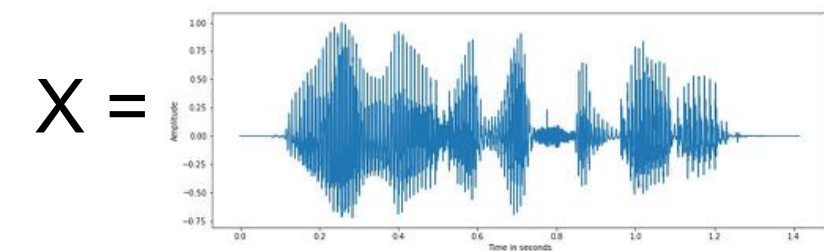
$$Y = \mathbb{R}$$

Detectie huidkanker



$$Y = \{\text{kanker, normaal}\}$$

Spraakherkenning



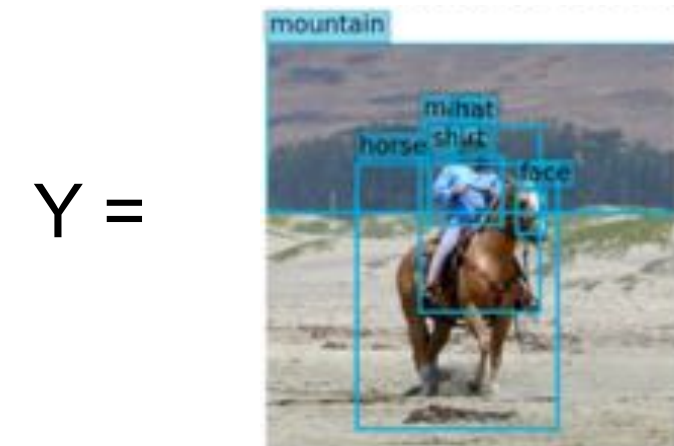
$$Y = \{\text{aardvark, \dots, zucchini}\}$$

Semantische segmentation



$$Y=f(X)$$

Semantische labeling

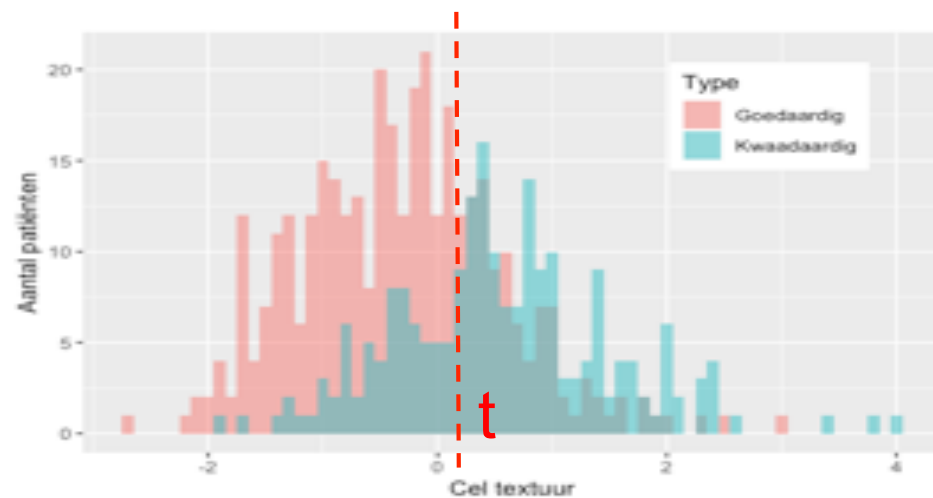


Hoe leren machine learning modellen ?

Twee belangrijke keuzes om $Y=f(X)$ te leren

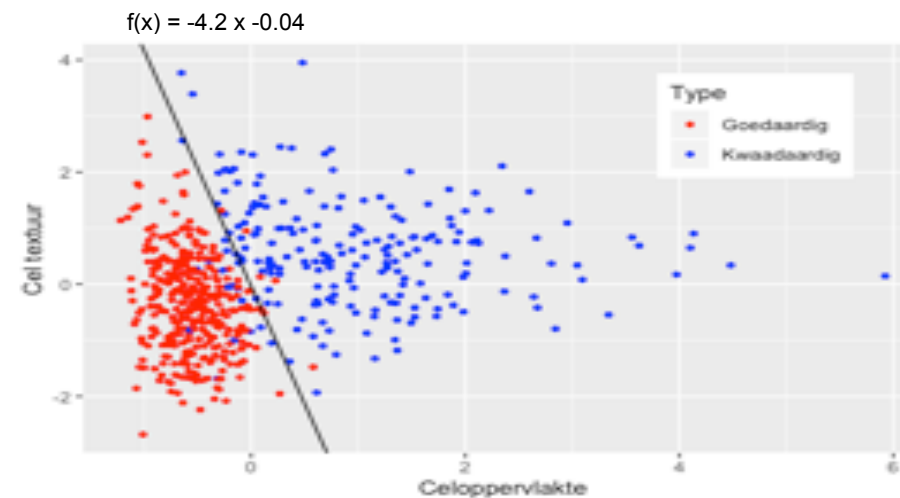
1. Keuze van de representatie (parametrische vorm) van de functie

Classificatie met drempelwaarde



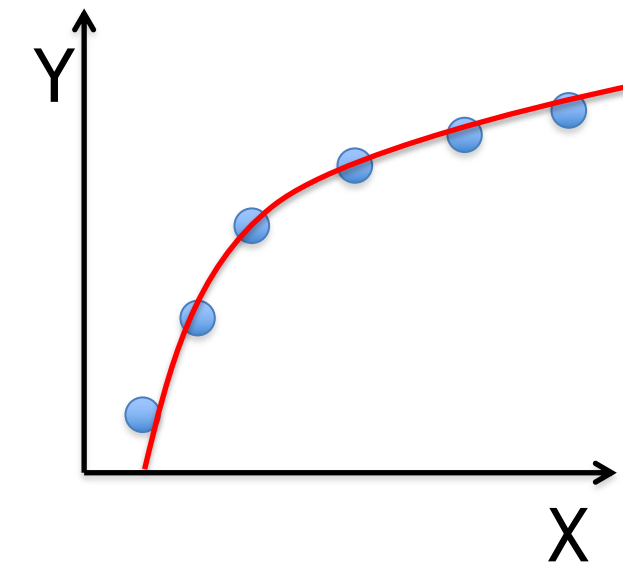
$$f(X) = \begin{cases} \text{Kwaadaardig als } \textit{celtextuur} \geq t \\ \text{Goedaardig als } \textit{celtextuur} < t \end{cases}$$

Classificatie met lineair model



$$f(X) = \begin{cases} \text{Kwaadaardig als} \\ \textit{celtextuur} \geq -4.2 \textit{ celoppervlakte} - 0.04 \\ \text{Goedaardig anders} \end{cases}$$

Regressie met polynomiaal model



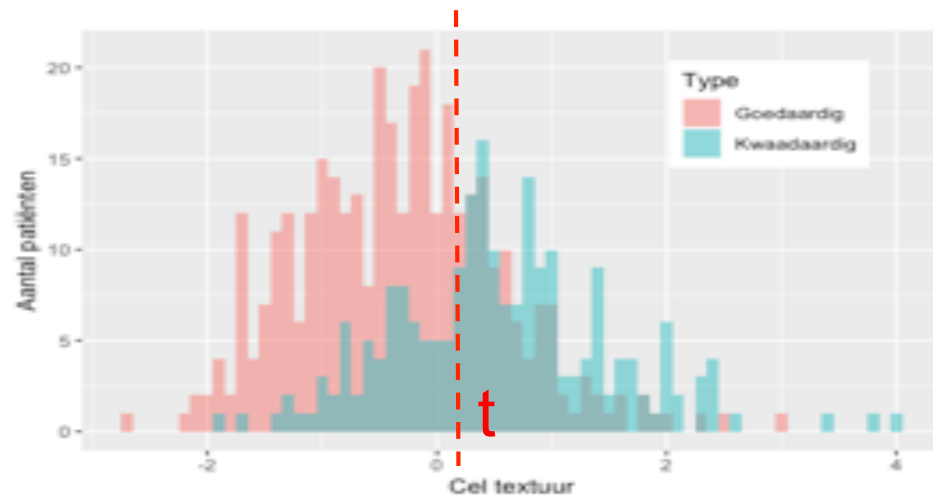
$$f(x) = w_0 + w_1x + w_2x^2$$

Twee belangrijke keuzes om $Y=f(X)$ te leren

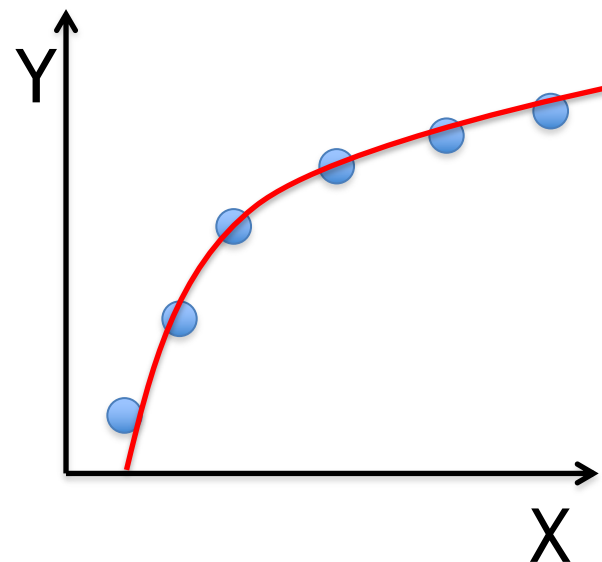
2. Optimalisatie van de parameters van het model

- Afhankelijk van het probleem en de gekozen representatie
- Leren = goede parameterwaarden vinden
 - Typisch een iteratief proces !
- **Formulering van het optimalisatieprobleem**
 - **Minimaliseer een “foutfunctie” (loss function)**

Foutfuncties minimaliseren



- 1 parameter te optimaliseren: t
- Foutfunctie $E(t) = FN(t) + FP(t)$
- Vind optimale waarde t^* waarvoor $E(t^*)$ minimaal is (bijvoorbeeld door alle waarden van t te overlopen en telkens $E(t)$ te berekenen)

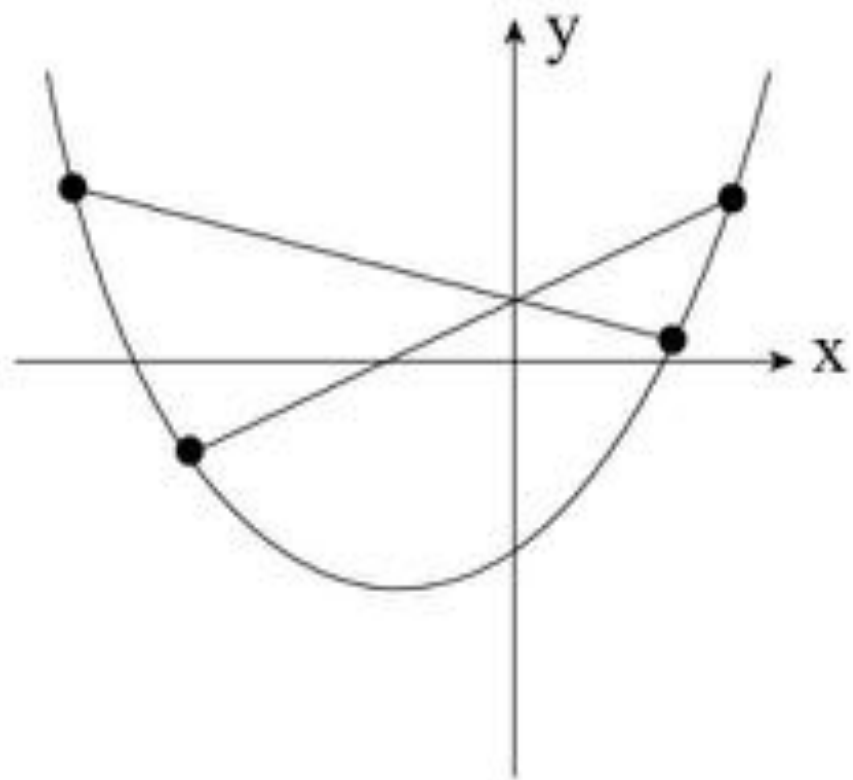


$$f(x) = w_0 + w_1x + w_2x^2$$

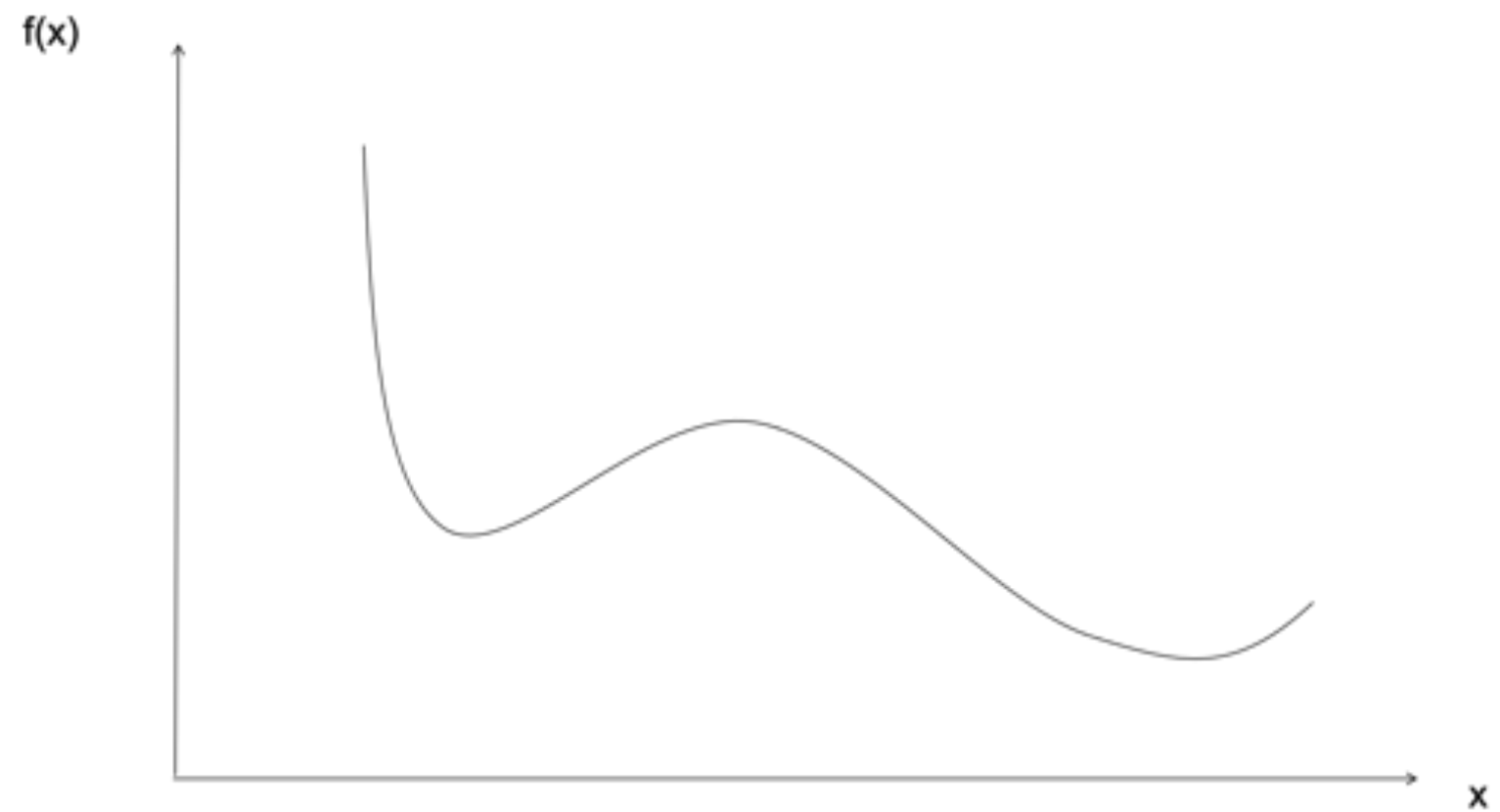
- 3 parameters te optimaliseren: w_0, w_1, w_2
- Foutfunctie
$$E(w_0, w_1, w_2) = \sum_{i=1}^N (f(x_i) - y_i)^2$$
- Vind optimale waarde voor w_0, w_1, w_2

Algemene foutfuncties minimaliseren

“Gemakkelijke” foutfuncties

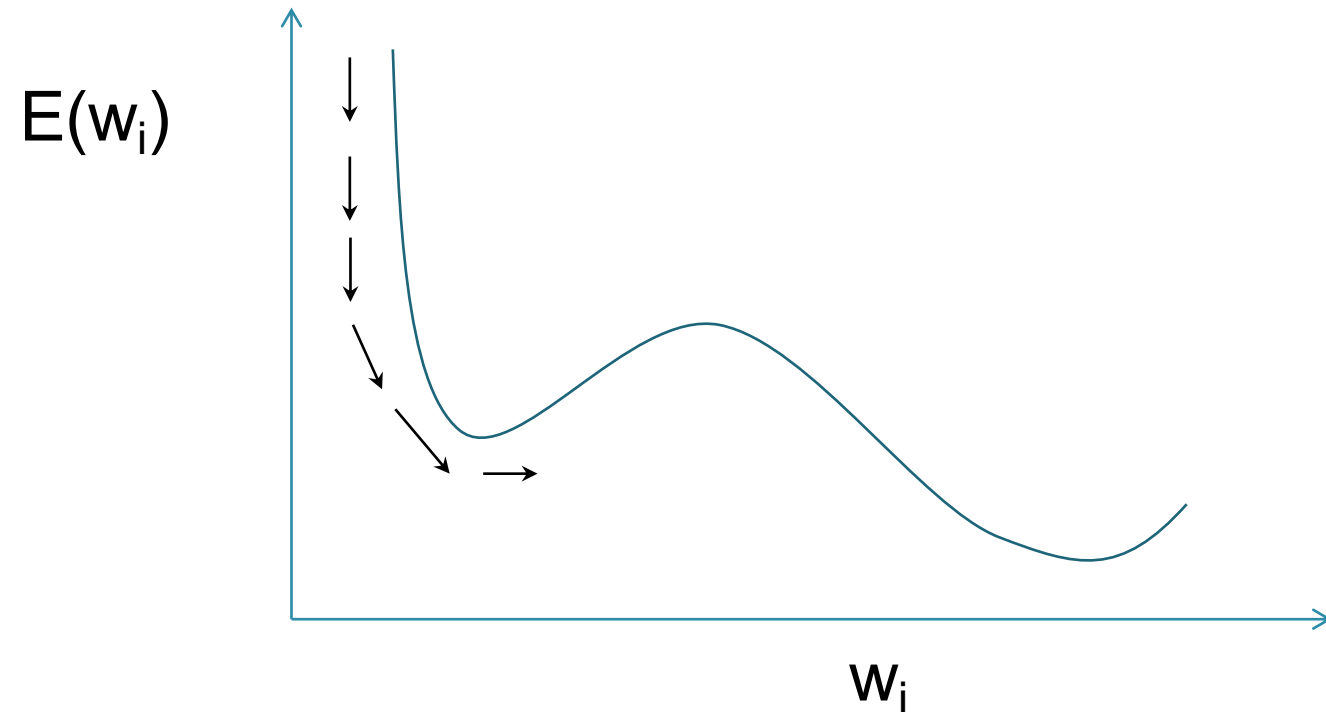


“Moeilijke” foutfuncties

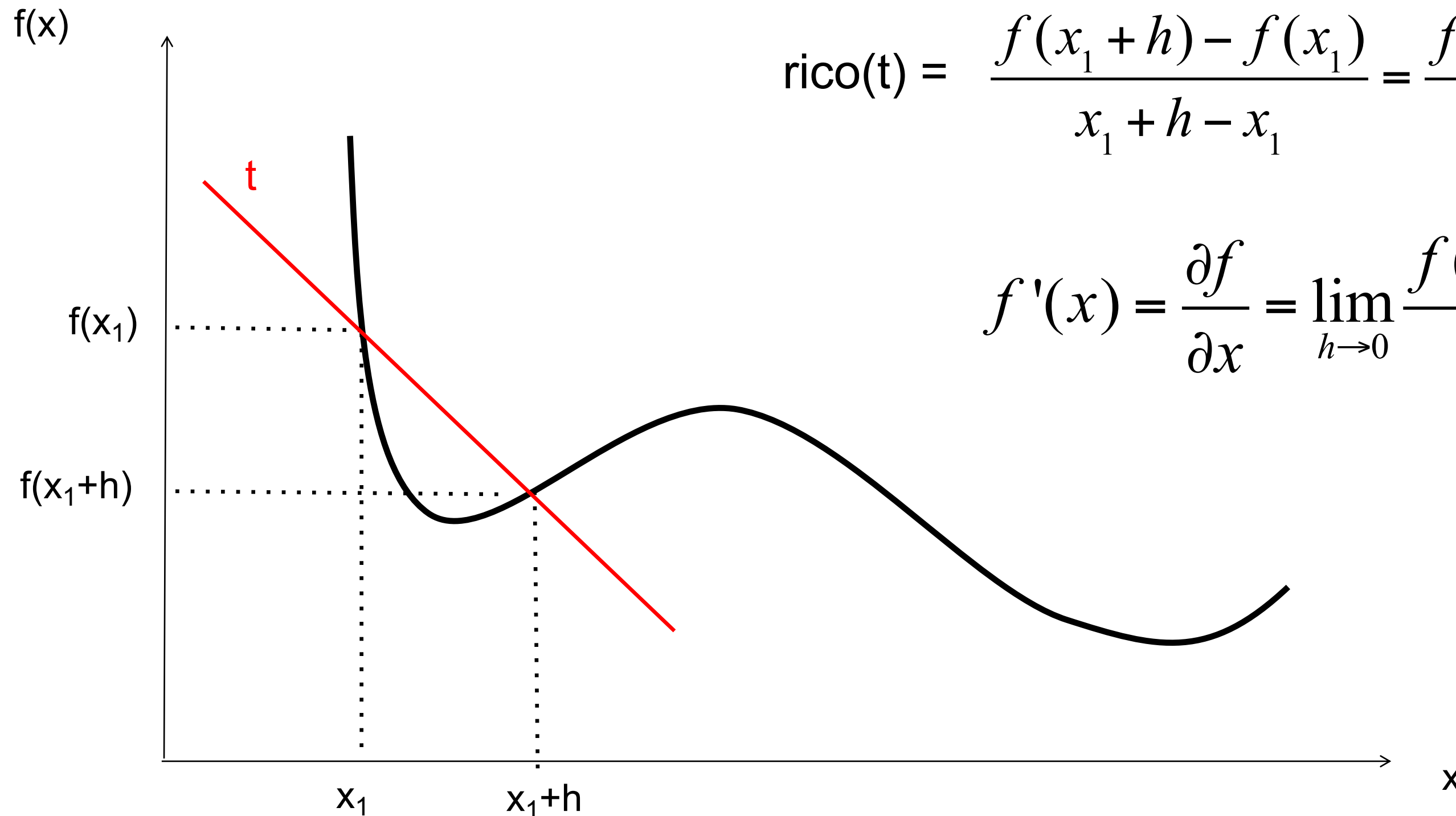


Functies minimaliseren met “gradient descent”

- De foutfunctie E is vaak een niet-lineaire, complexe functie van alle w_i , die niet analytisch kan geminimaliseerd worden
- Daarom gebruiken we een iteratieve, benaderende methode: gradient descent



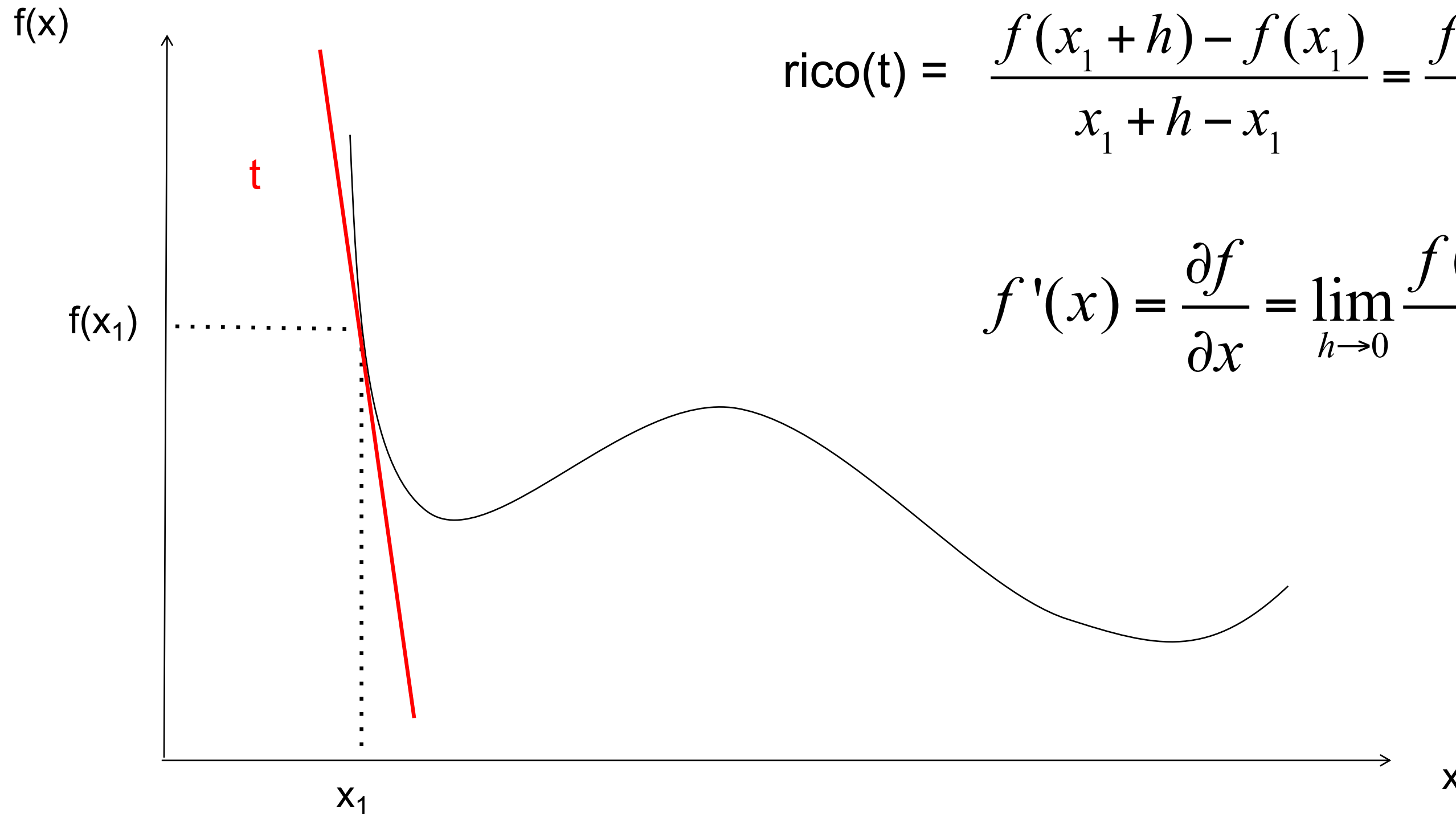
Gradient descent – volg de eerste afgeleide bergaf



$$\text{rico}(t) = \frac{f(x_1 + h) - f(x_1)}{x_1 + h - x_1} = \frac{f(x_1 + h) - f(x_1)}{h}$$

$$f'(x) = \frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x_1 + h) - f(x_1)}{h}$$

Gradient descent – volg de eerste afgeleide bergaf

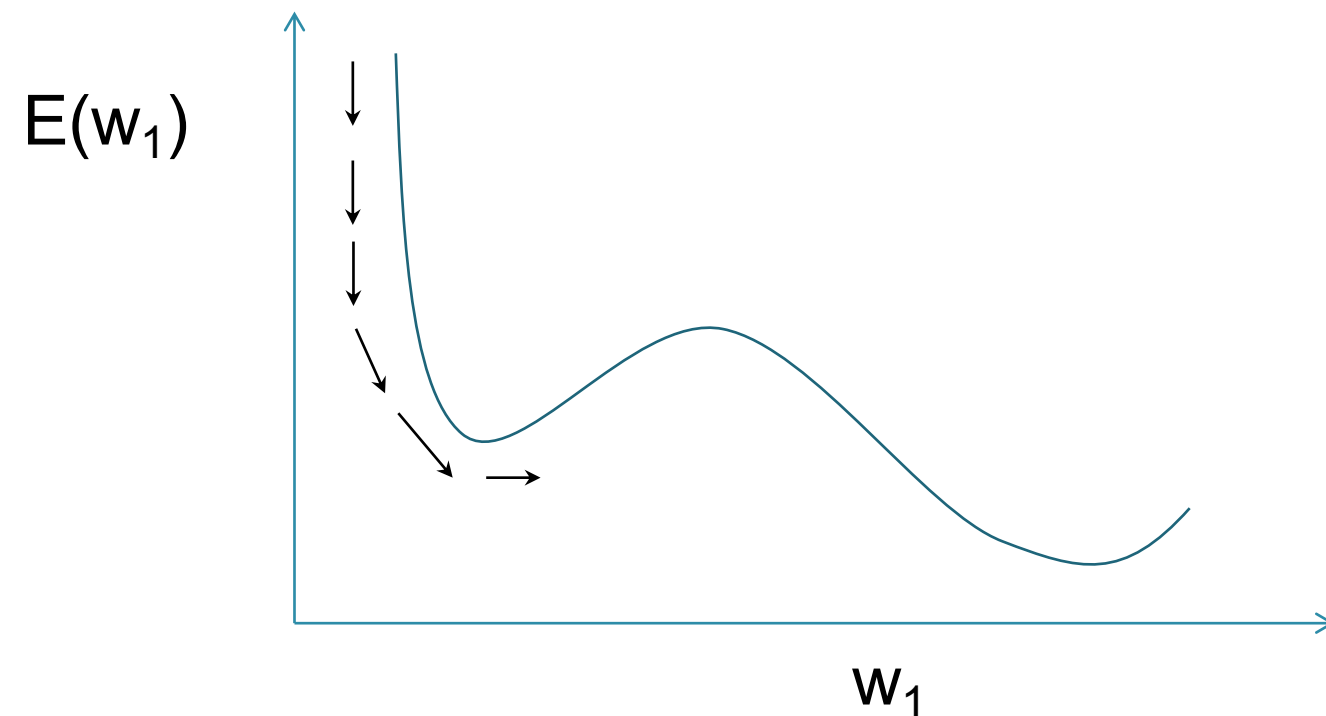


$$\text{rico}(t) = \frac{f(x_1 + h) - f(x_1)}{x_1 + h - x_1} = \frac{f(x_1 + h) - f(x_1)}{h}$$

$$f'(x) = \frac{\partial f}{\partial x} = \lim_{h \rightarrow 0} \frac{f(x_1 + h) - f(x_1)}{h}$$

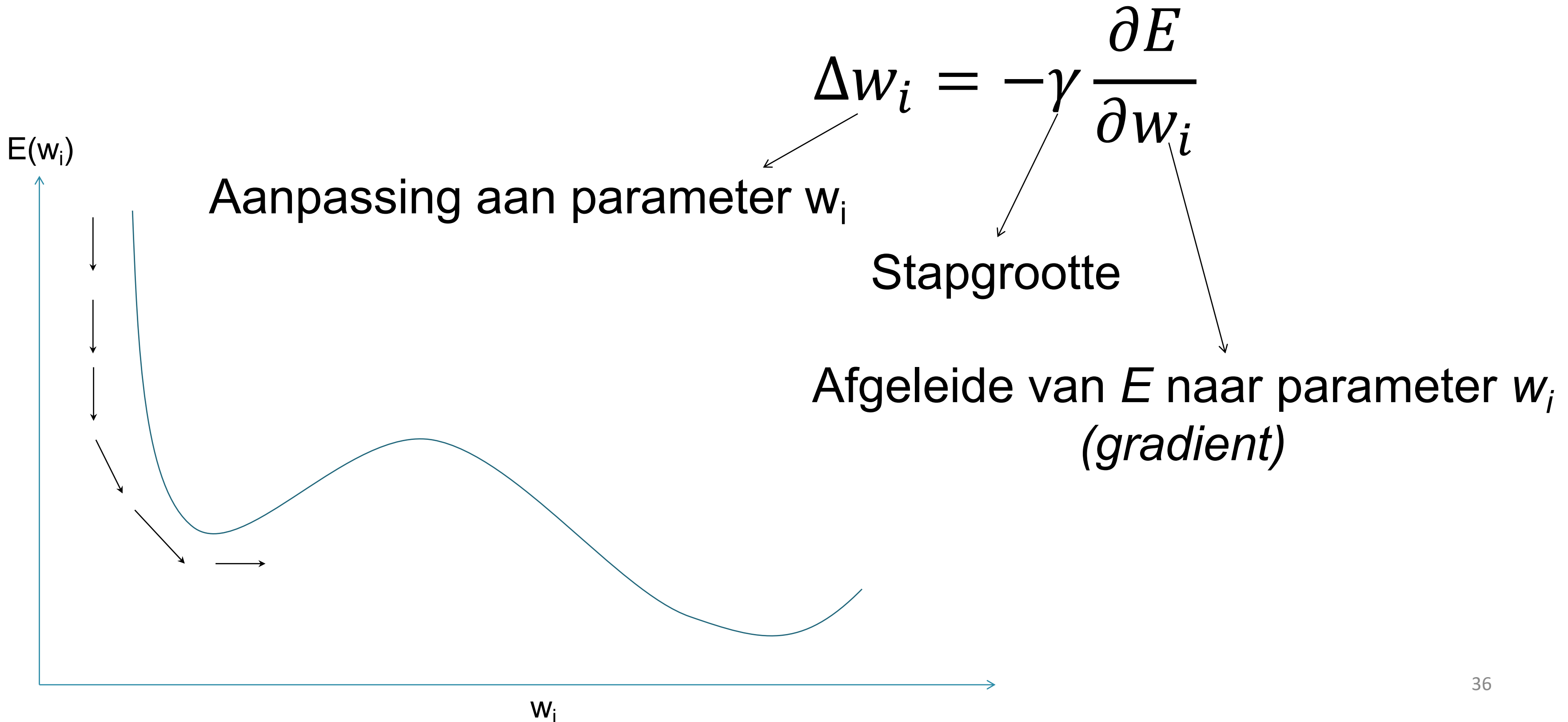
Functies minimaliseren met “gradient descent”

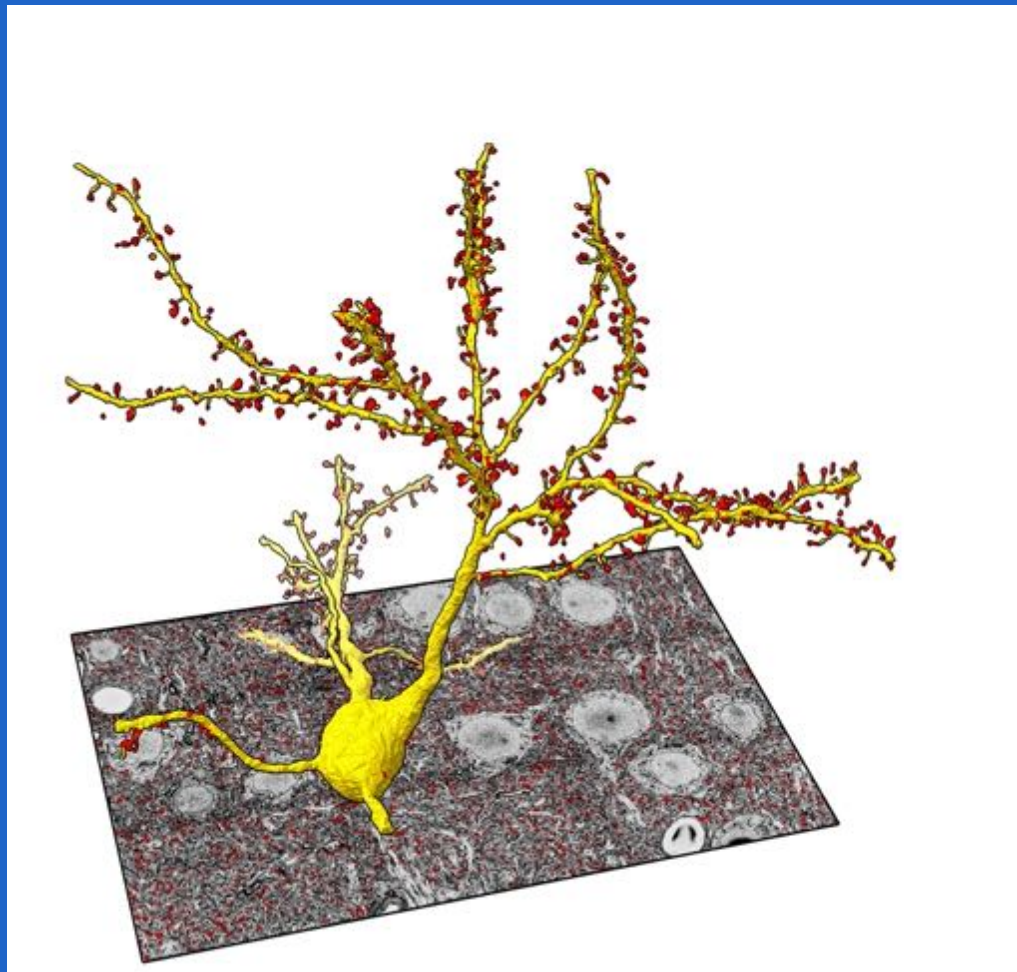
- De foutfunctie E is vaak een niet-lineaire, complexe functie van alle w_i , die niet analytisch kan geminimaliseerd worden
- Daarom gebruiken we een iteratieve, benaderende methode: gradient descent



We volgen de eerste afgeleide neerwaarts, tot we in een lokaal minimum terechtkomen

Functies minimaliseren met “gradient descent”

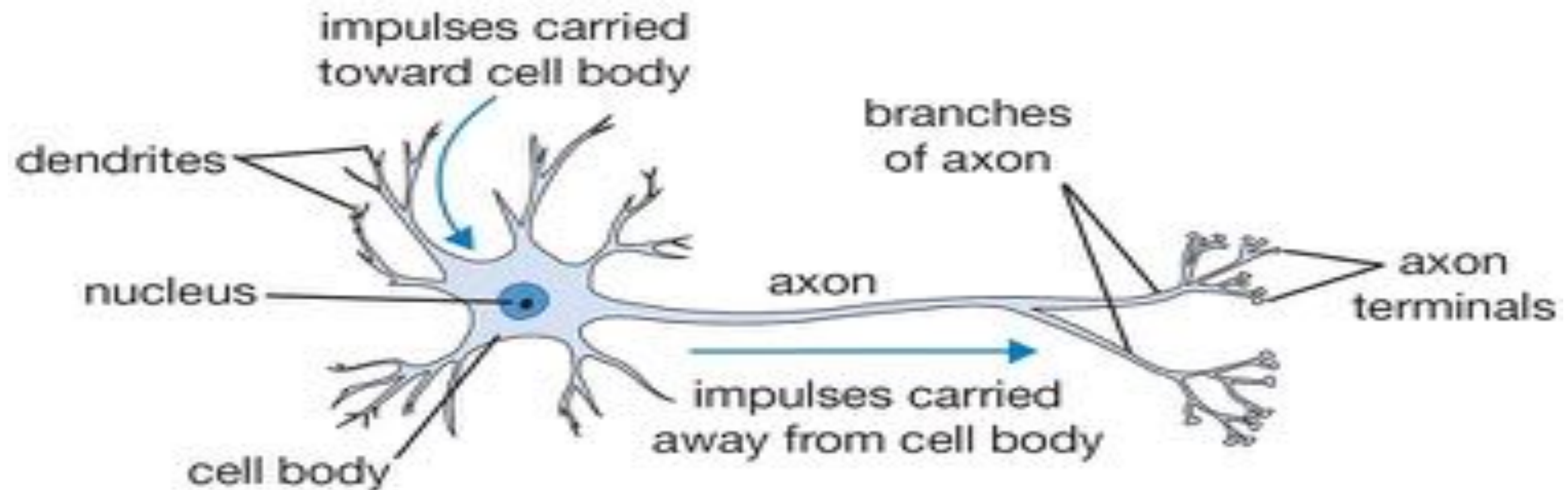




Neurale netwerken

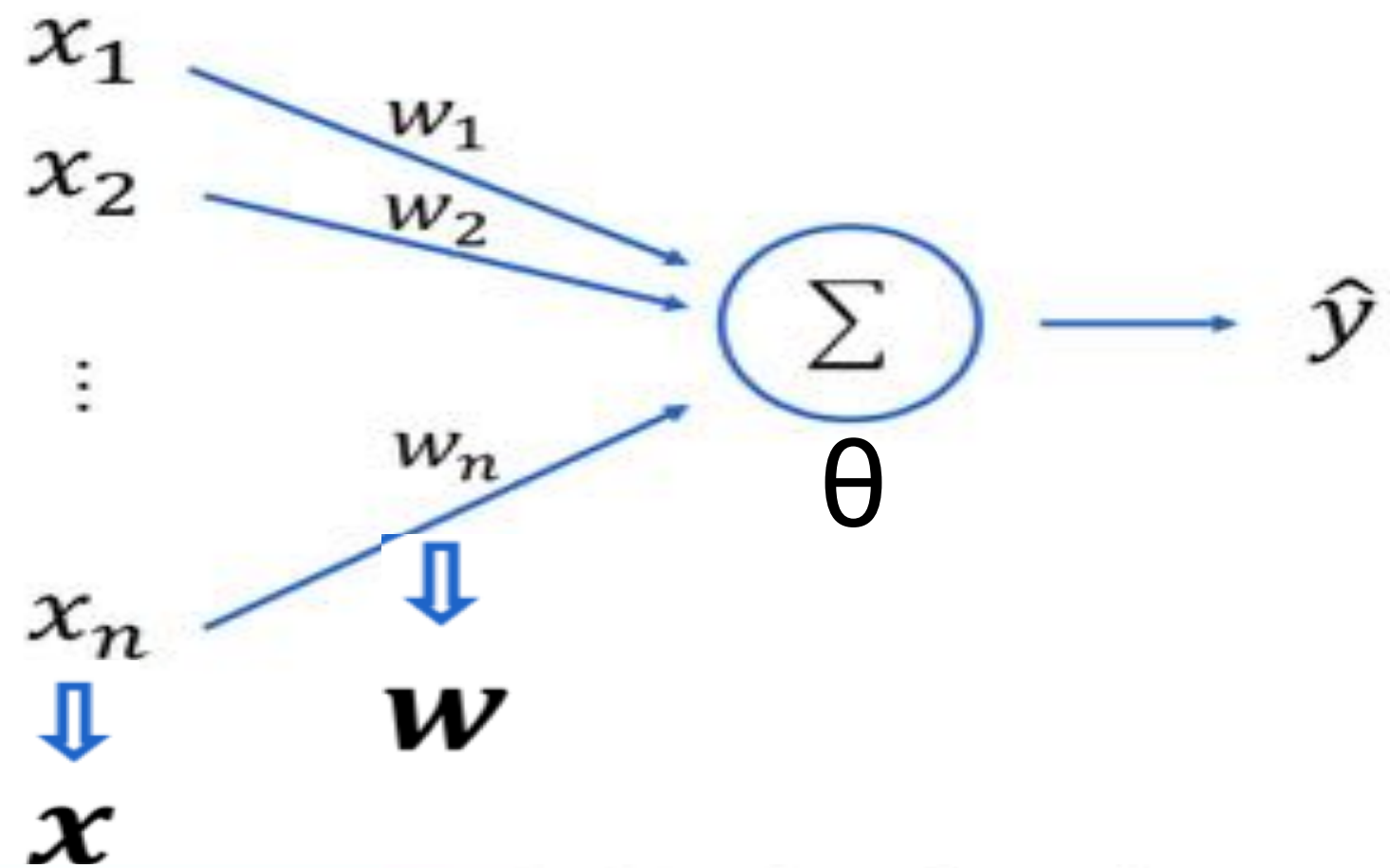
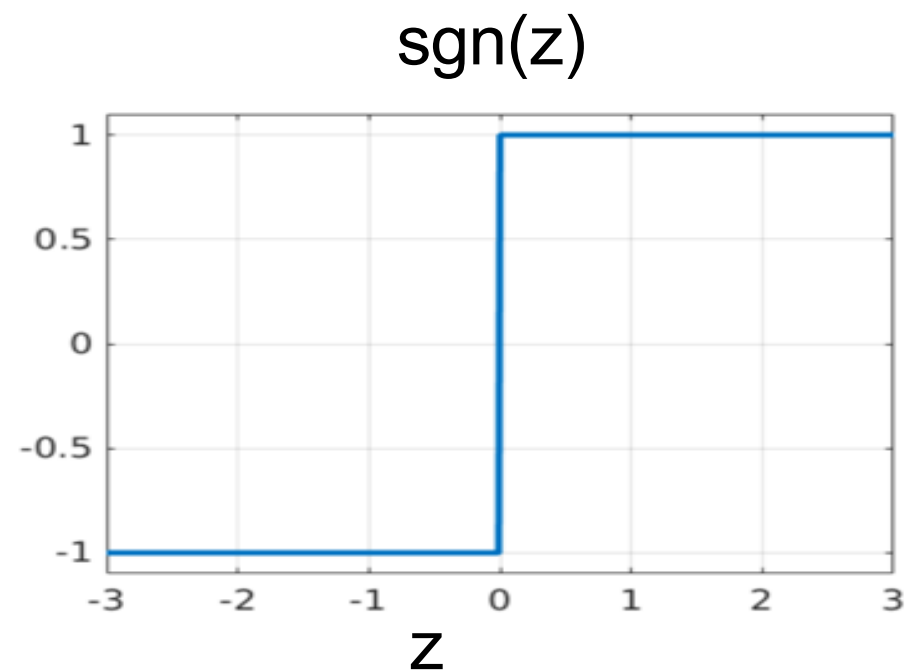
Neurale netwerken

Onze hersenen bestaan uit $\sim 10^{11}$ neuronen, en elk neuron is geconnecteerd met $\sim 10,000$ andere neuronen



Perceptron = wiskundig model van een neuron

- Inputs $x_i \in \{0,1\}$
- Weights $w_i \in \mathbb{R}$
- Threshold θ
- Output $\hat{y} \in \{-1,1\}$



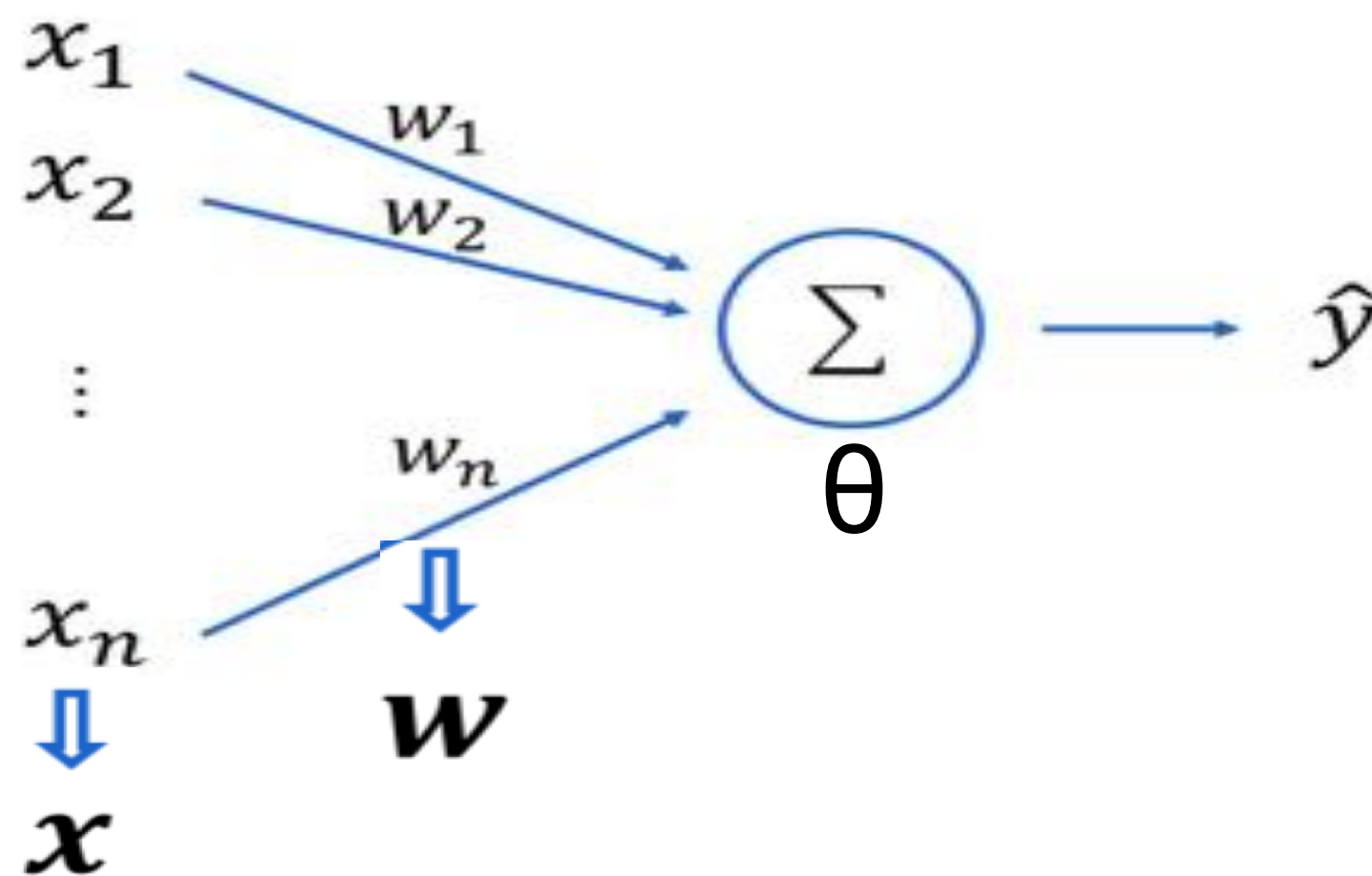
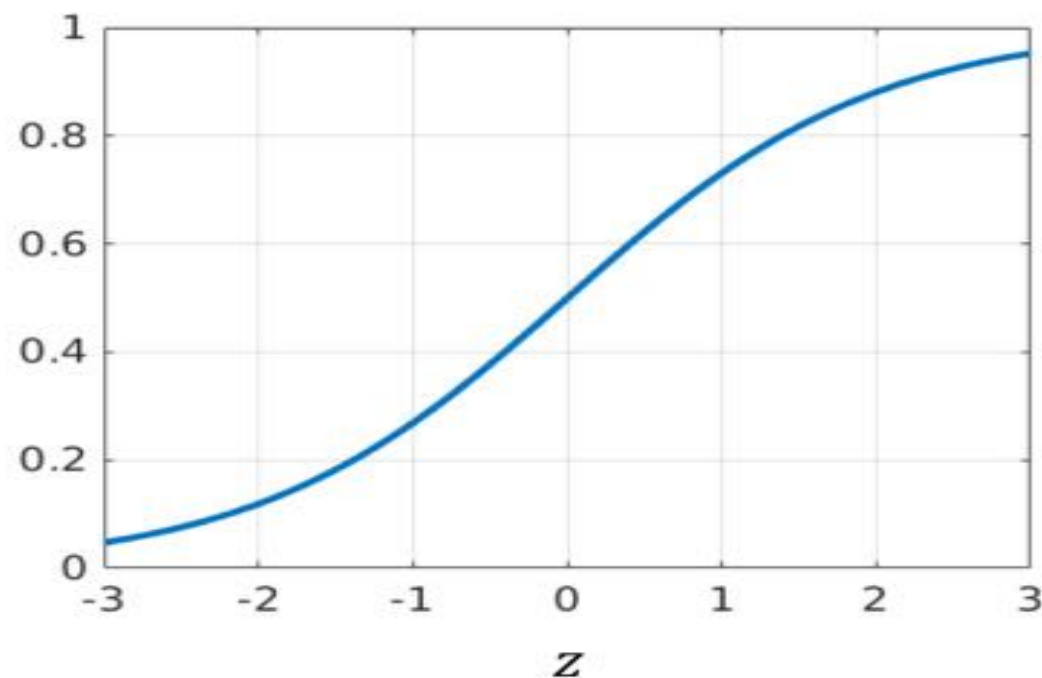
Activation function

$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} - \theta)$$
$$\sigma(z) = \begin{cases} -1 & \text{if } z \leq 0 \\ 1 & \text{if } z > 0 \end{cases} \quad (\text{sign function})$$

Neuron met sigmoïde activatiefunctie

- Inputs $x_i \in \{0,1\}$
- Weights $w_i \in \mathbb{R}$
- Threshold $\theta \in \mathbb{R}$
- Output $\hat{y} \in [0,1]$

Sigmoid neuron



$$\hat{y} = \sigma(\mathbf{w}^T \mathbf{x} - \theta)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (\text{sigmoid function})$$

Geometrische interpretatie

Output van het neuron:

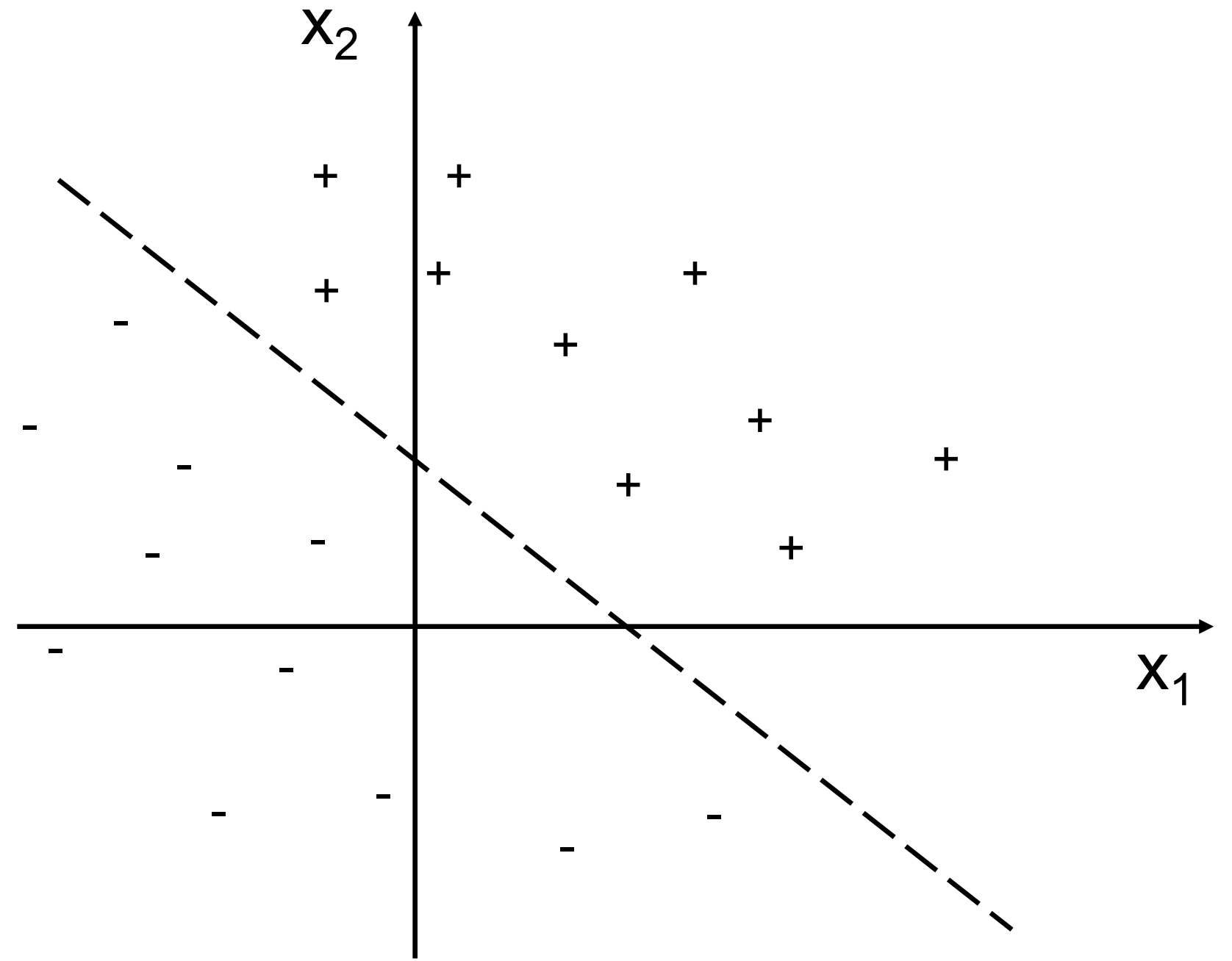
$$\hat{y} = w_1 x_1 + w_2 x_2 - \theta$$

Stel dit gelijk aan 0 :

$$w_1 x_1 + w_2 x_2 - \theta = 0$$

Dit is equivalent met de vergelijking van een rechte :

$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{\theta}{w_2}$$



Hoe vinden we de optimale parameters?

- We hebben een dataset van leervoorbeelden nodig:
 - Voorbeelden van de input-outputrelatie die we willen leren
 - ***Training set***
- Het model heeft de volgende parameters:
 - Gewicht voor elke input: w_i
 - Drempelwaarde θ (gewicht met input -1)
- We moeten wiskundig vastleggen wat we bedoelen met ***“optimaal”***

We stellen een foutfunctie op

$$E = \sum_p E^p = \frac{1}{2} \sum_p (y^p - \hat{y}^p)^2$$

Echt klasselabel (verwachte output)

Geobserveerd klasselabel (output van het model)

**We willen deze foutfunctie minimaliseren !
→ bepaalt “optimaliteit”**

We stellen een foutfunctie op

Constant (gegeven in dataset)

$$E = \sum_p E^p = \frac{1}{2} \sum_p (y^p - \hat{y}^p)^2$$

Deze term hangt af van ons model !

- Model wordt bepaald door parameters
- Foutfunctie E is dus een functie die afhankelijk is van de parameters

Opstellen van het model – intuïtief idee

We starten met willekeurige parameters

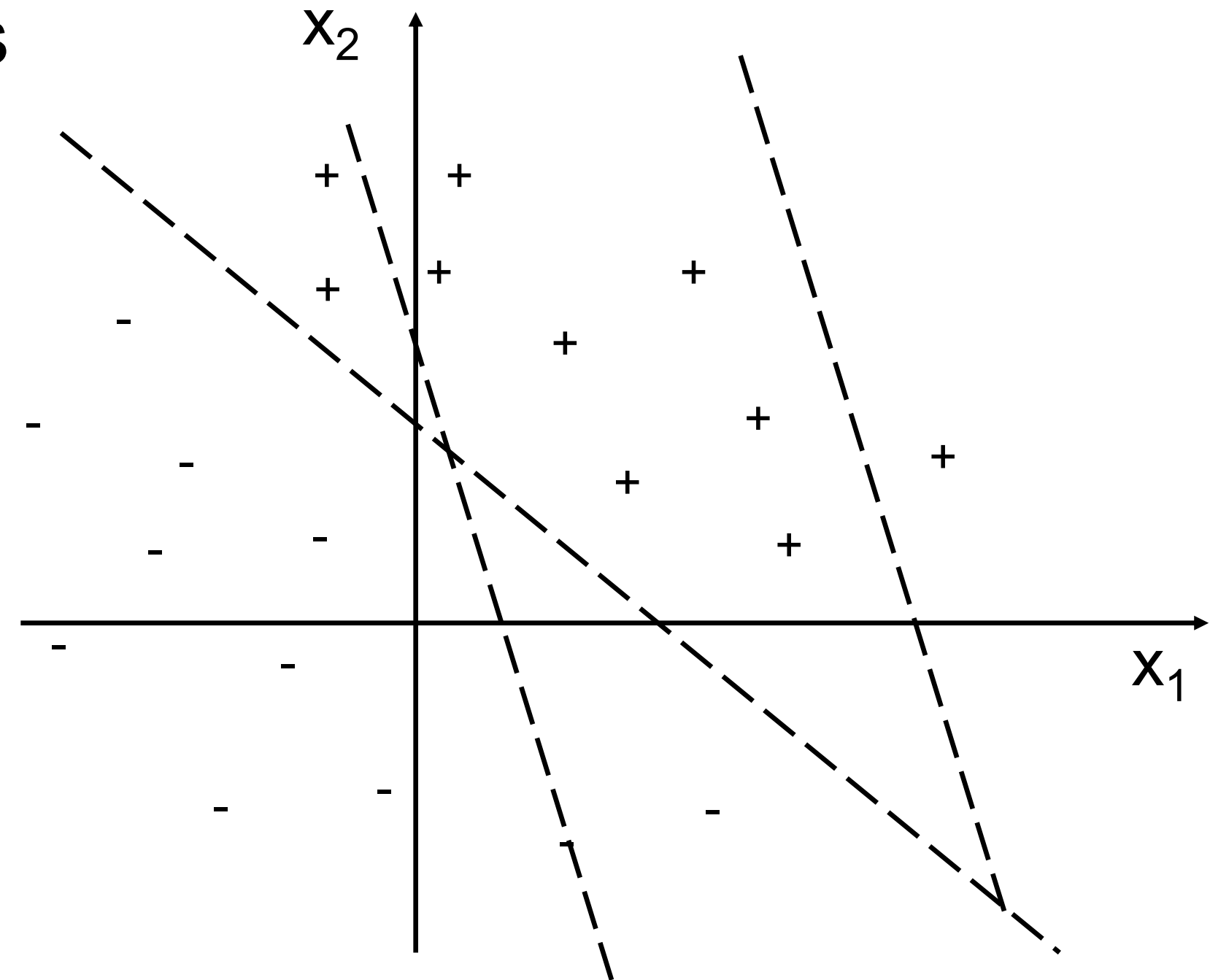
$$w_i(t_0) = \text{random}$$

We updaten de parameters:

$$w_i(t_1) = w_i(t_0) + \Delta w_i(t_0)$$

We updaten de parameters:

$$w_i(t_2) = w_i(t_1) + \Delta w_i(t_1)$$



“Train” het neurale netwerk met gradient descent

1. Bepaal de topologie van het netwerk (inputs, outputs)
2. Bepaal een stapgrootte γ
3. Initialiseer alle parameters met willekeurige waarden
4. Herhaal het volgende proces
 1. Bereken de foutfunctie E voor het huidige netwerk
 2. Bereken de gradientinformatie voor alle parameters
 3. Pas de waarden van de parameters aan met de volgende formule:

$$\Delta w_j = -\gamma \frac{\partial E}{\partial w_j}$$

Berekening van $\frac{\partial E}{\partial w_j}$

We hadden de foutfunctie bepaald als $E = \sum_p E^p = \frac{1}{2} \sum_p (y^p - \hat{y}^p)^2$

We passen nu de kettingregel voor afgeleiden toe:

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial \hat{y}^p} \frac{\partial \hat{y}^p}{\partial w_j}$$

De eerste factor kunnen we berekenen als:

$$\frac{\partial E^p}{\partial \hat{y}^p} = -(y^p - \hat{y}^p)$$

Berekening van $\frac{\partial E}{\partial w_j}$

We hadden de foutfunctie bepaald als $E = \sum_p E^p = \frac{1}{2} \sum_p (y^p - \hat{y}^p)^2$

We passen nu de kettingregel voor afgeleiden toe:

$$\frac{\partial E^p}{\partial w_j} = \frac{\partial E^p}{\partial \hat{y}^p} \frac{\partial \hat{y}^p}{\partial w_j}$$

De tweede factor berekenen we adhv $\hat{y} = \mathbf{w}^T \mathbf{x} + \theta$

$$\frac{\partial \hat{y}^p}{\partial w_j} = \mathbf{x}_j^p$$

Berekening van $\frac{\partial E}{\partial w_j}$

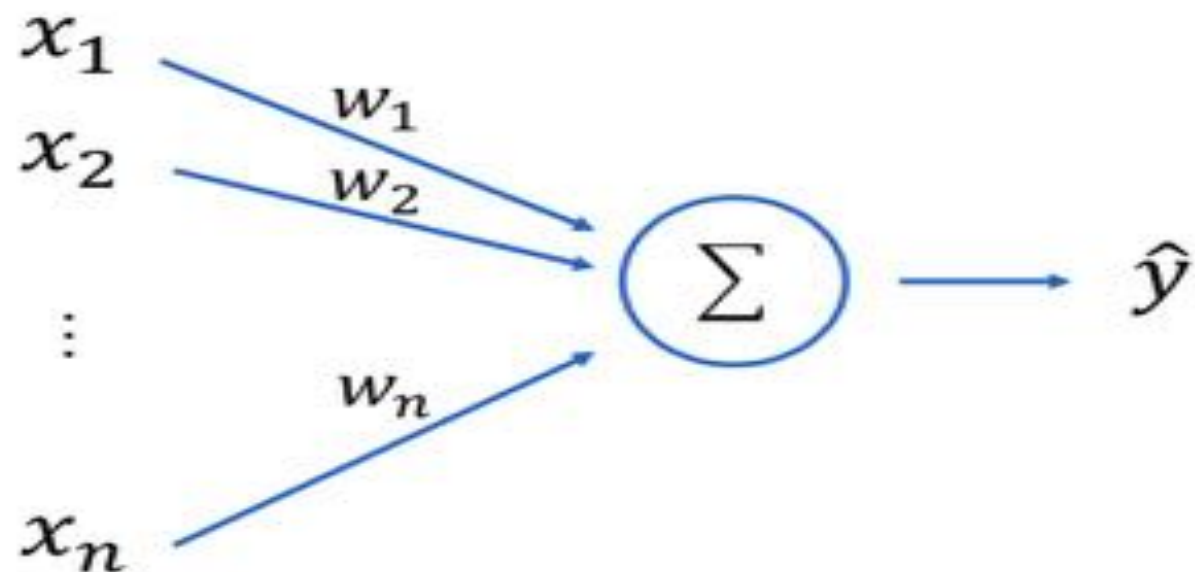
We bekommen nu :

$$\Delta_p \mathbf{w} = -\gamma \nabla_{\mathbf{w}} E^p = \gamma (\mathbf{y}^p - \hat{\mathbf{y}}^p) \mathbf{x}^p$$

Stapgrootte in gradient descent

Verschil tussen verwachte output en geobserveerde output

Waarde van de input

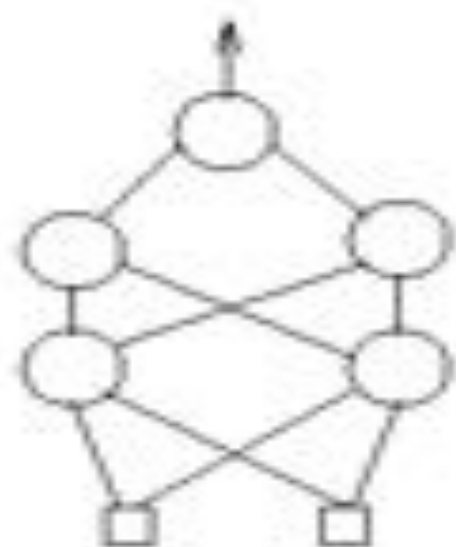
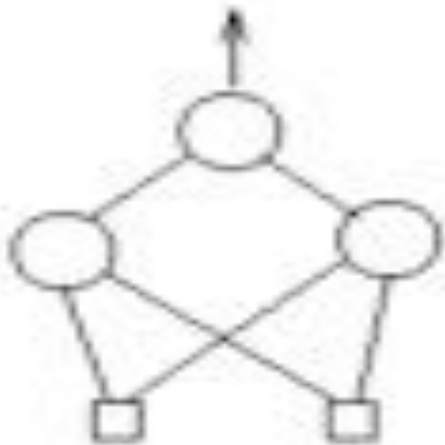
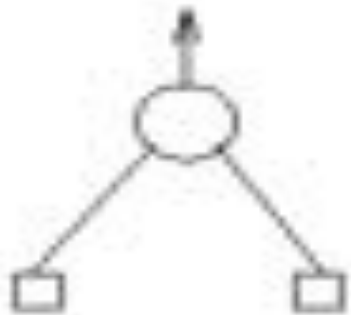


“Trainen” van het neurale netwerk

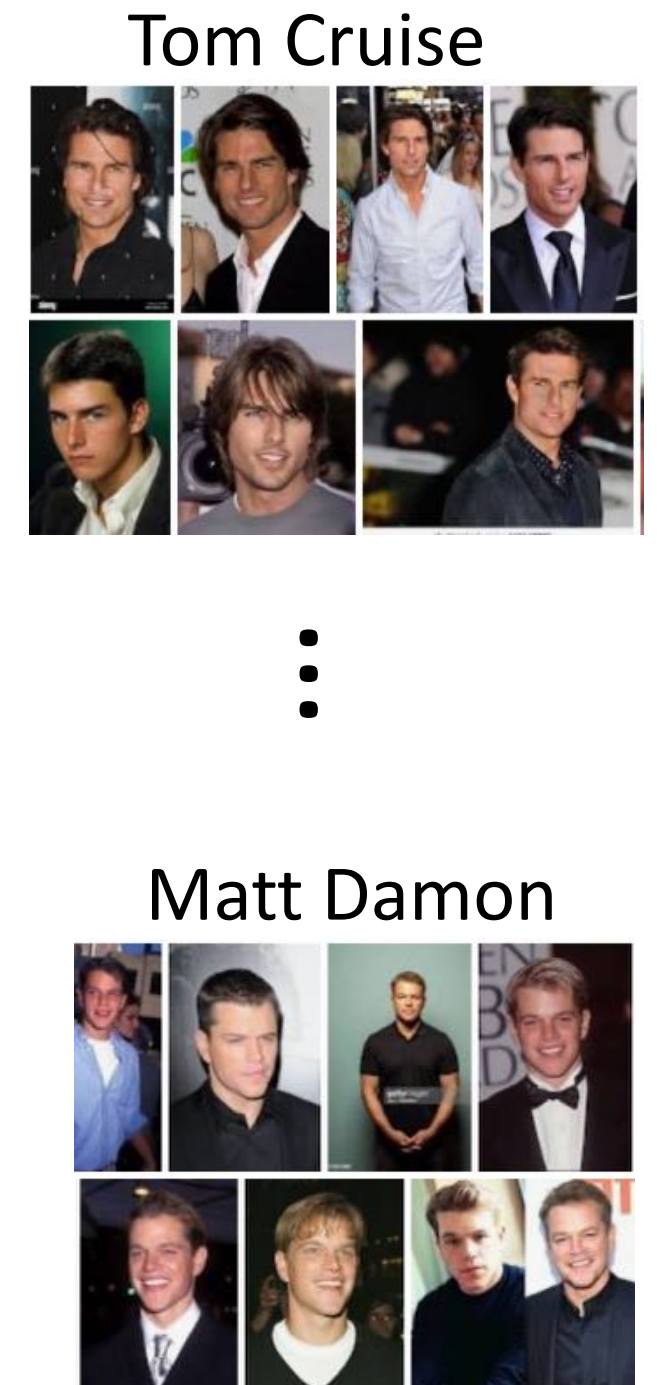
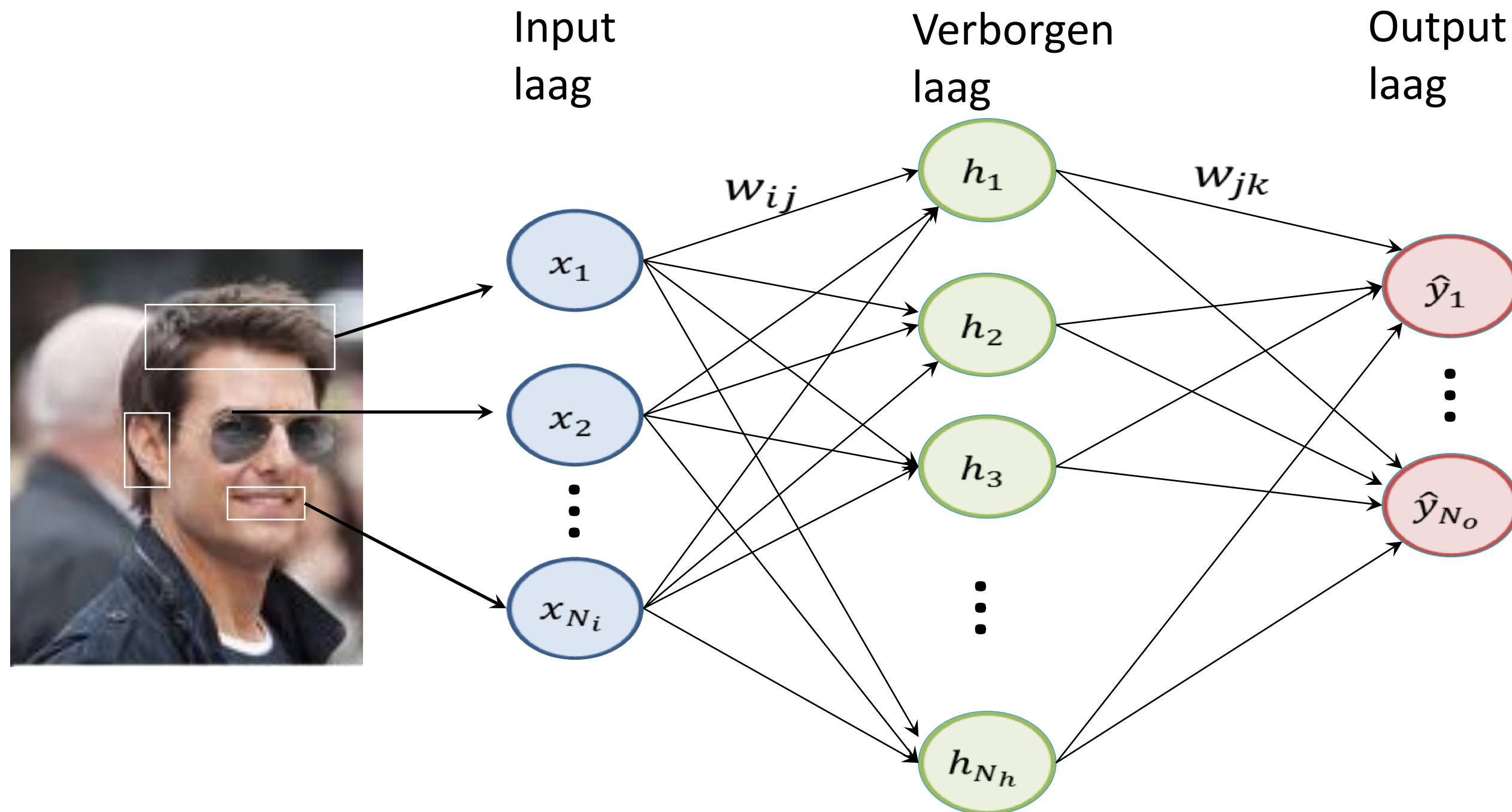
1. Bepaal de topologie van het netwerk (inputs, outputs)
2. Bepaal een stapgrootte γ
3. Initialiseer alle parameters met willekeurige waarden
4. Herhaal het volgende proces
 1. Bereken de foutfunctie E voor het huidige netwerk
 2. Bereken de gradientinformatie voor alle parameters
 3. Pas de waarden van de parameters aan met de volgende formule:

$$w_i(t+1) = w_i(t) + \Delta w_i(t) = w_i(t) + \gamma(y - \hat{y}_i)x_i$$

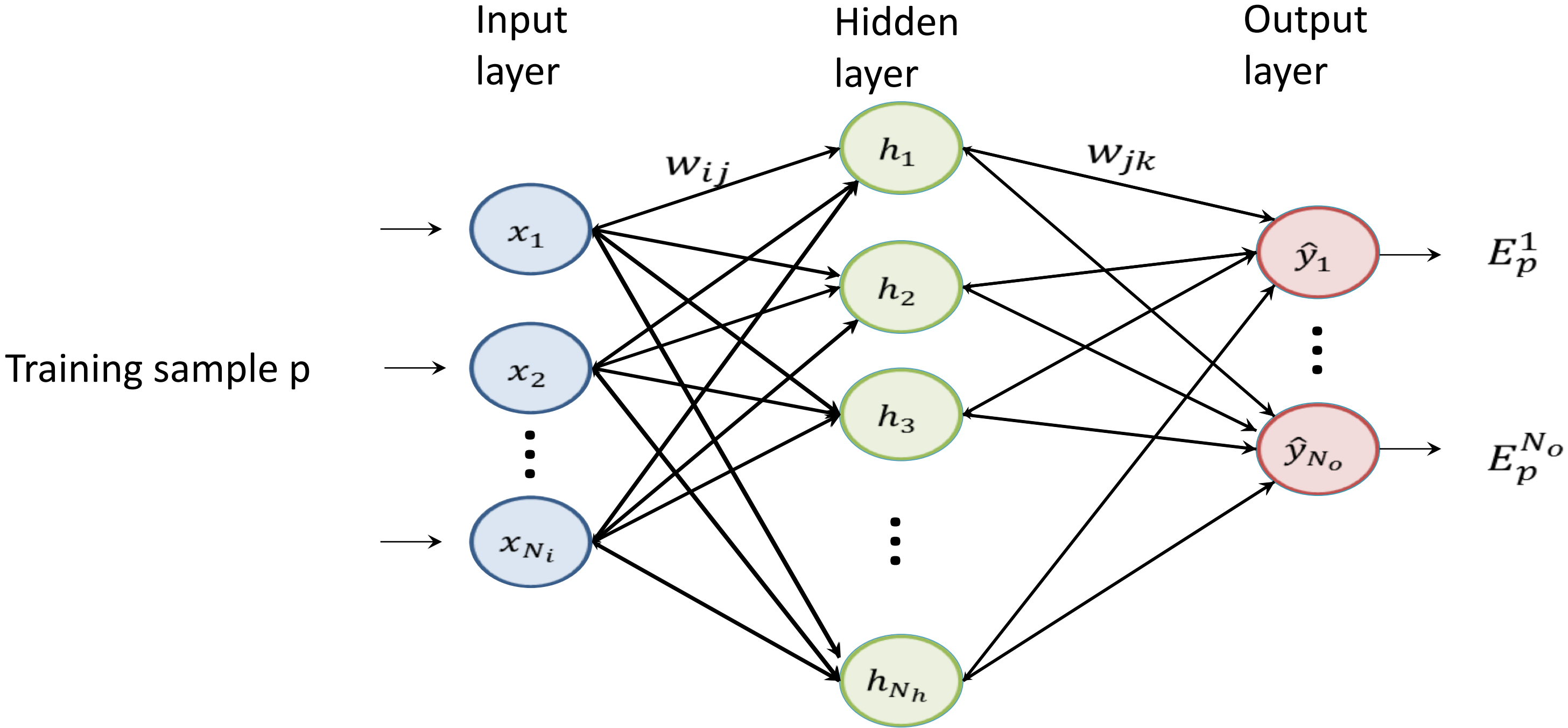
Multi-layer perceptrons (MLP)



Structuur van een MLP



Error backpropagation



Backpropagation: output neuronen

$$\Delta_p w_{jk} = \gamma (y_k^p - \hat{y}_k^p) \varphi'(s_k^p) h_j^p$$

Stapgrootte van
gradient descent

Verskil tussen verwachte output en
geobserveerde output

Afgeleide van de actievatiefunctie voor output k

Output van verborgen knoop h_j

Backpropagation: “verborgen” neuronen

$$\Delta_p w_{ij} = -\gamma x_i^p \varphi'(s_j^p) \sum_{k=1}^{N_o} \frac{\delta E_p}{\delta s_k^p} w_{jk}$$

Step size of the gradient descent

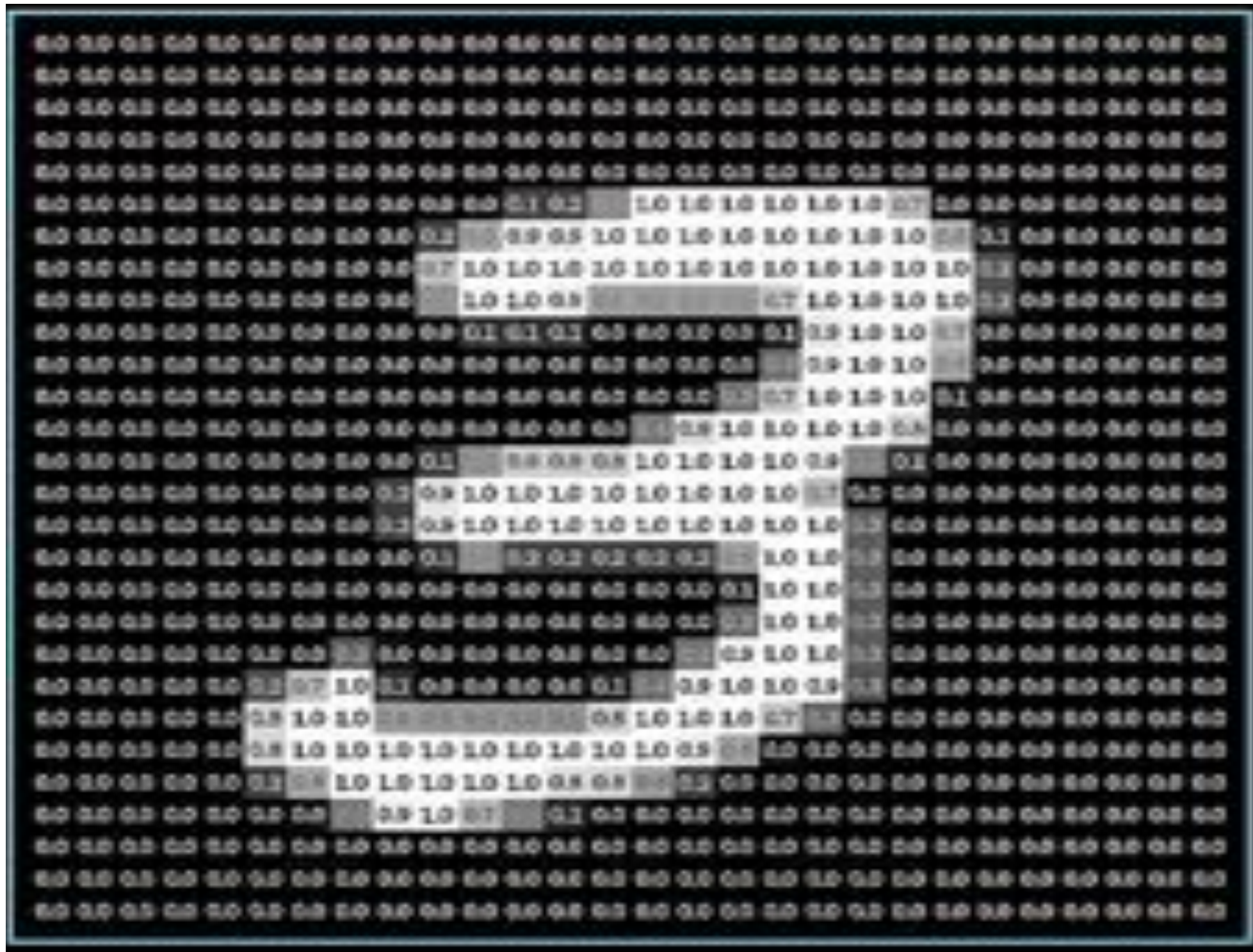
Output of previous layer (input)

Derivative of the activation function evaluated at output node k

Contribution to the error on all output nodes (next layer), weighted by its influence

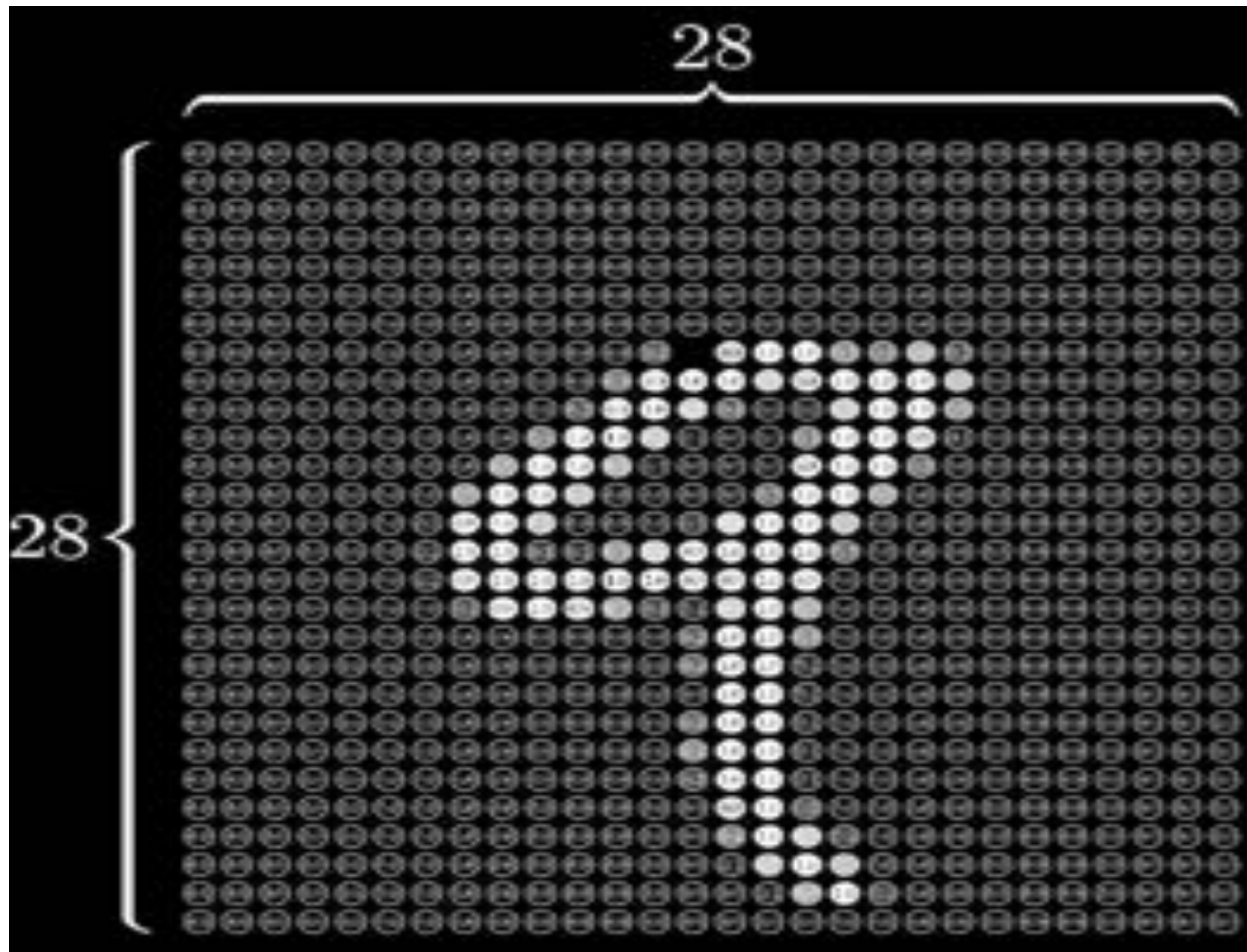
Influence of the current hidden node to output node k

MLP voorbeeld: beeldherkenning



0
1
2
3
4
5
6
7
8
9

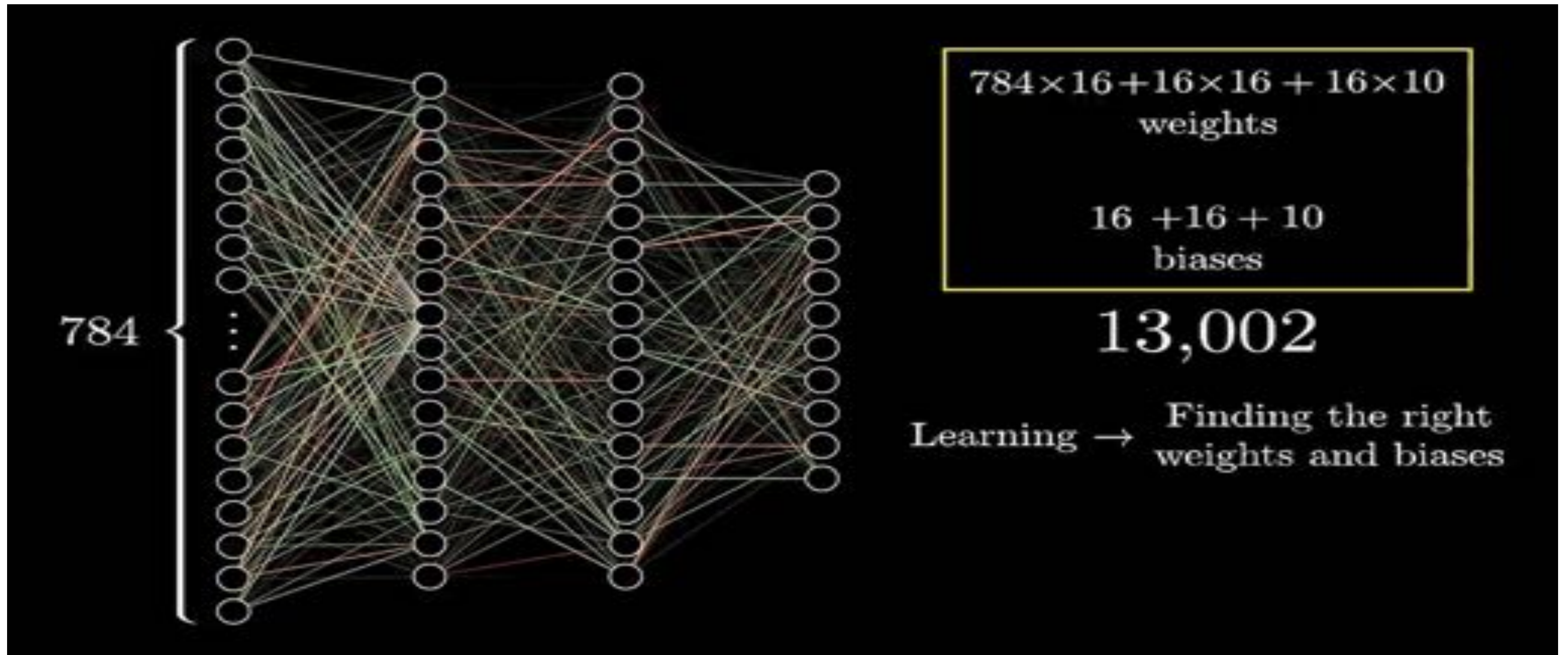
MLP voorbeeld: beeldherkenning



$$28 \times 28 = 784$$

784-dimensionele
inputruimte

MLP voorbeeld: beeldherkenning



Diepe neurale netwerken

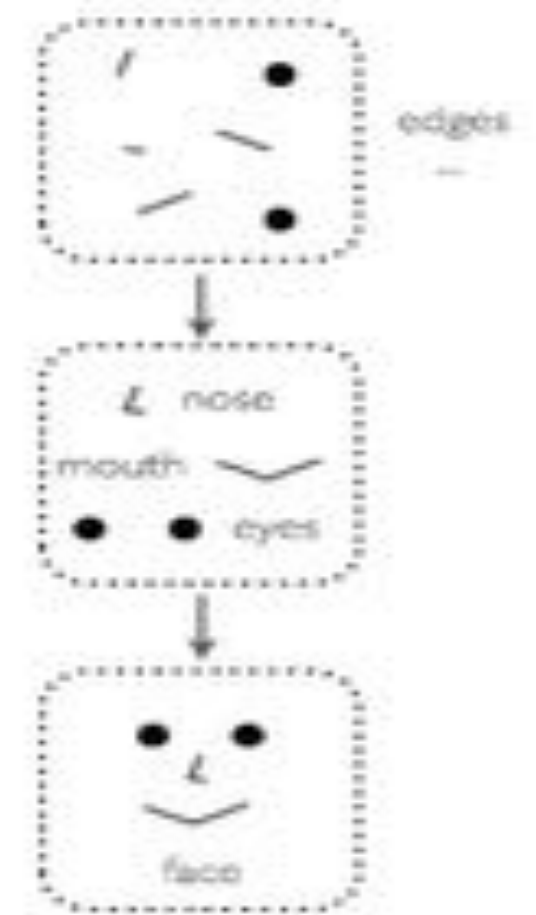
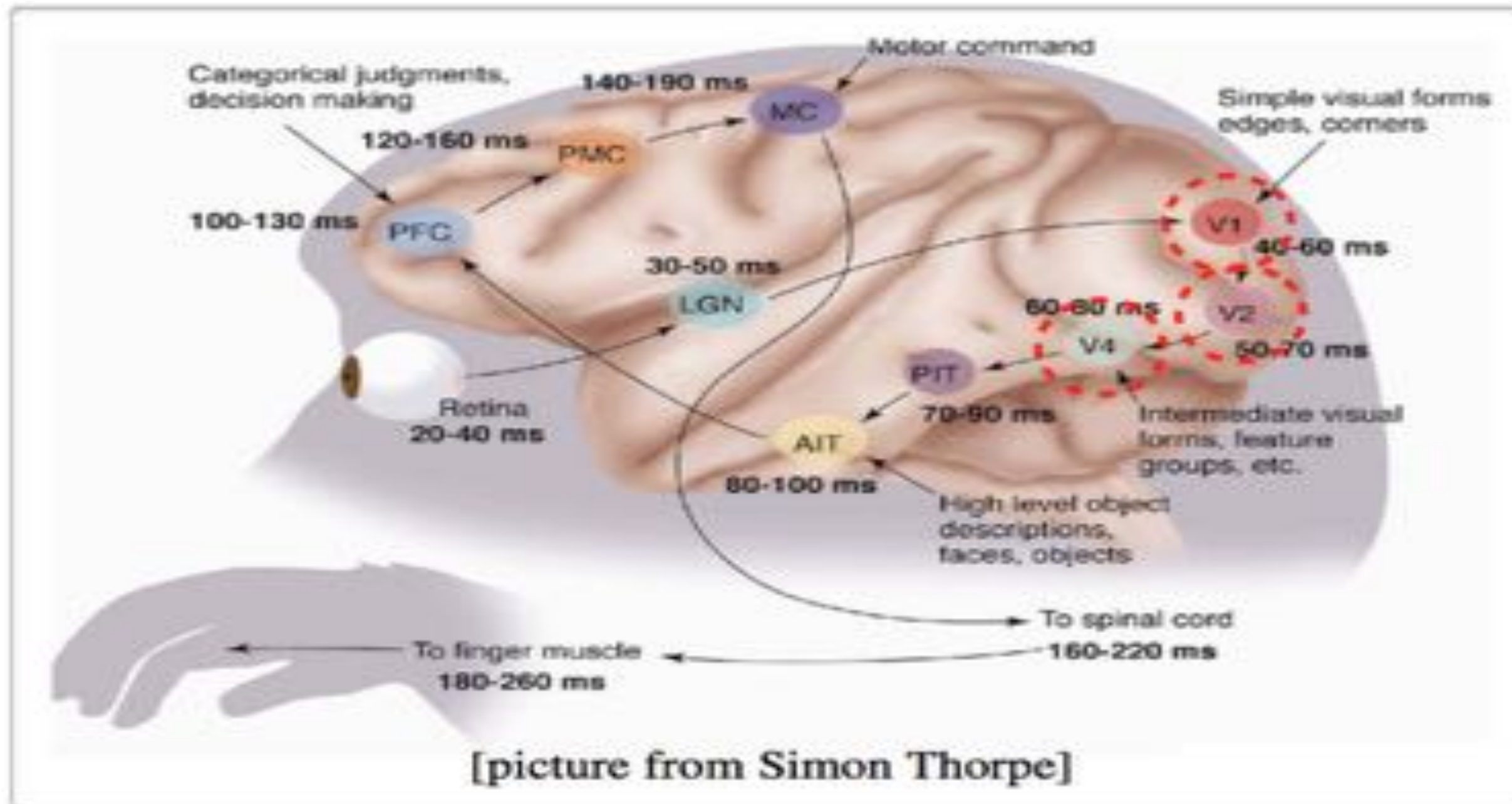
~

Deep learning

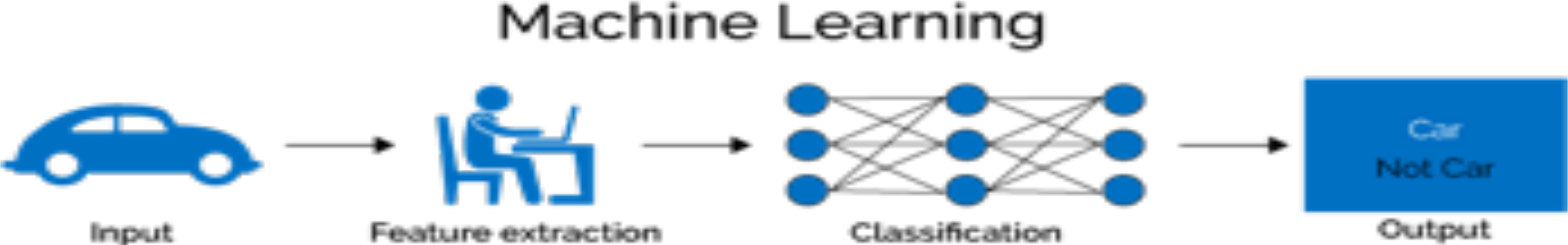
~

Representation learning

Inspiratie voor deep learning



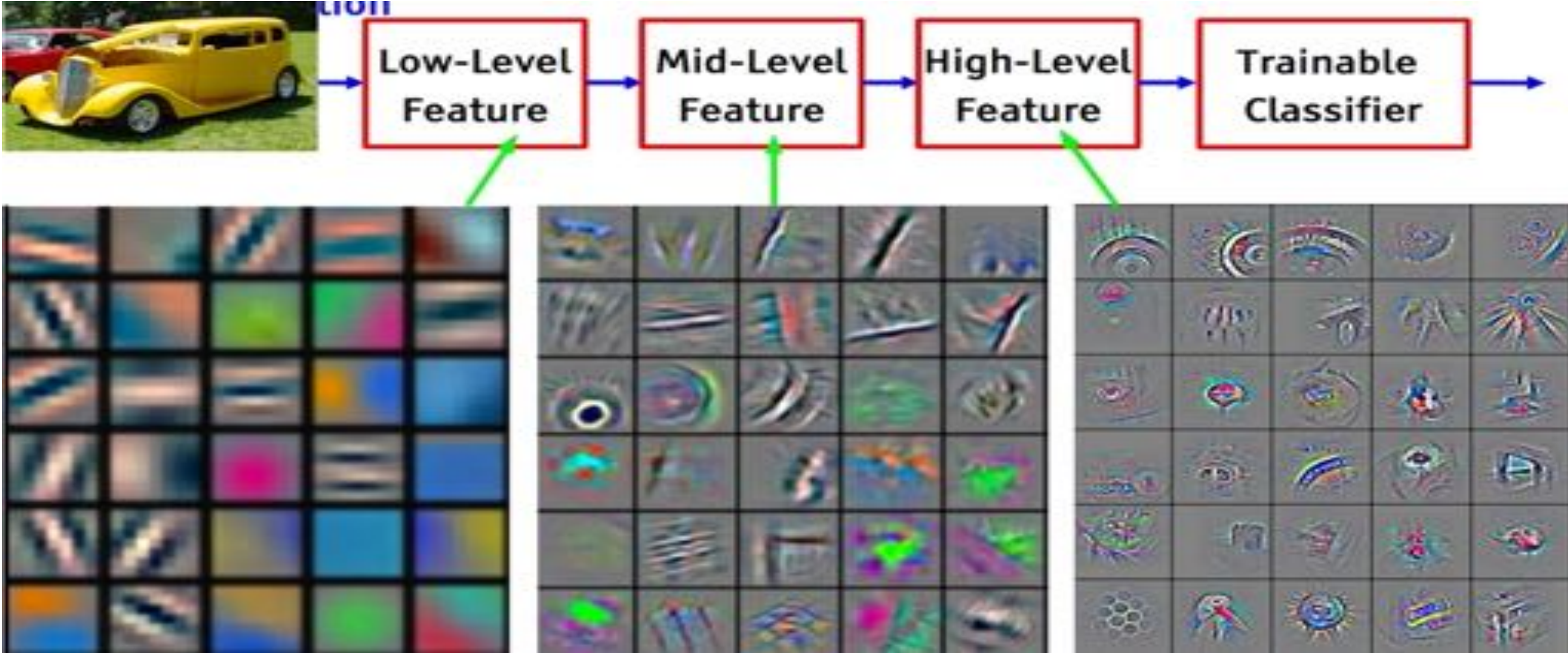
Deep learning versus klassieke ML



Hiërarchische representaties leren

- We leren een hiërarchie met toenemende niveau's van abstractie
- Elke stap is een aanpasbare transformatie
 - Beeldherkenning
 - Pixel → rand → texton → motif → deel → object
 - Tekst
 - Karakter → woord → woordgroep → deelzin → zin → verhaal
 - Spraak
 - Sample → spectrale band → geluid → ... → klank → foneem → woord → ...

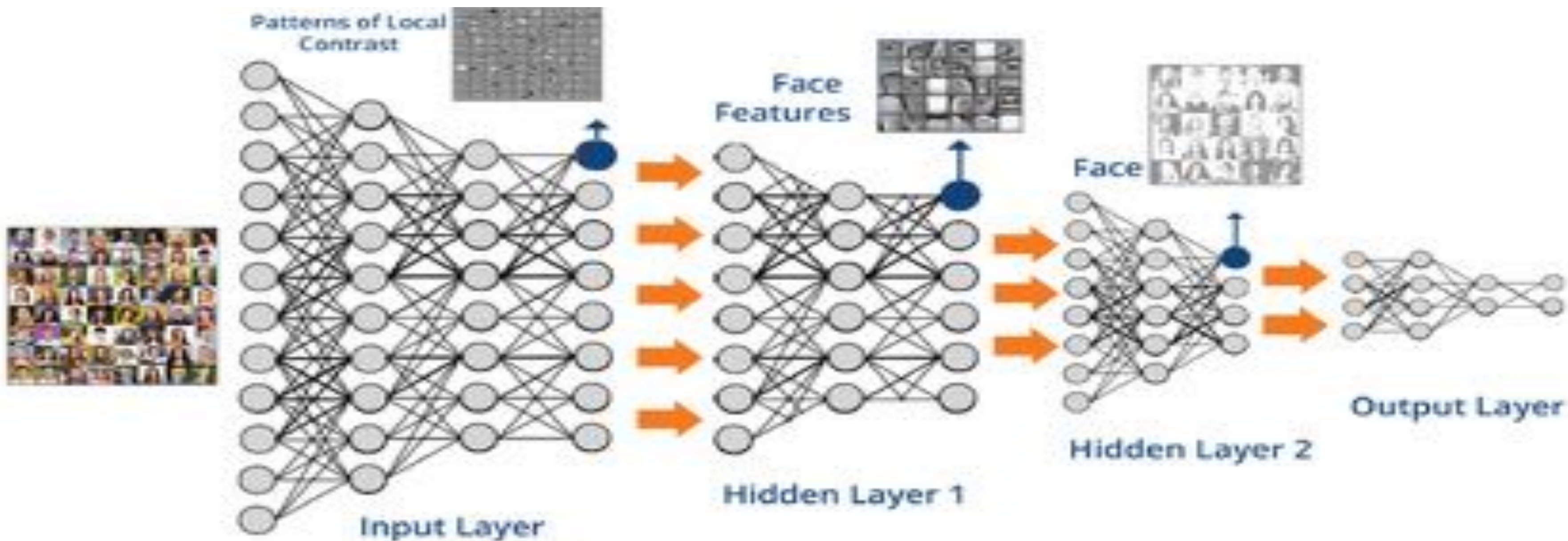
Hiërarchische representaties leren



Convolutionele neurale netwerken

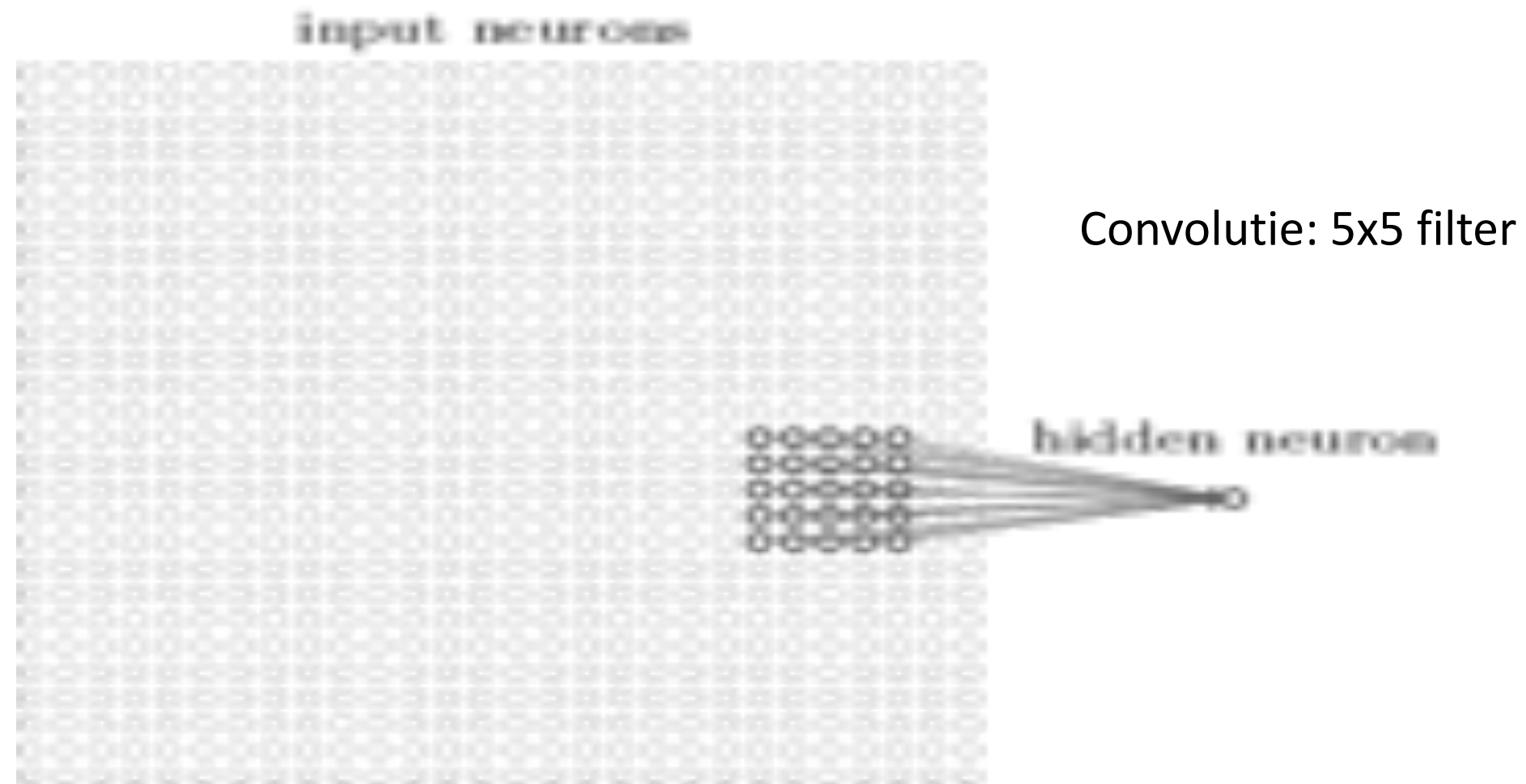
- Meest gebruikt type van diepe neurale netwerken
- Belangrijk element is de convolutie-operator (filter)
- Wordt vaak gebruikt voor beeldherkenning
 - Gezichtsherkenning
 - Image captions

Convolutionele neurale netwerken



De convolutie-operator

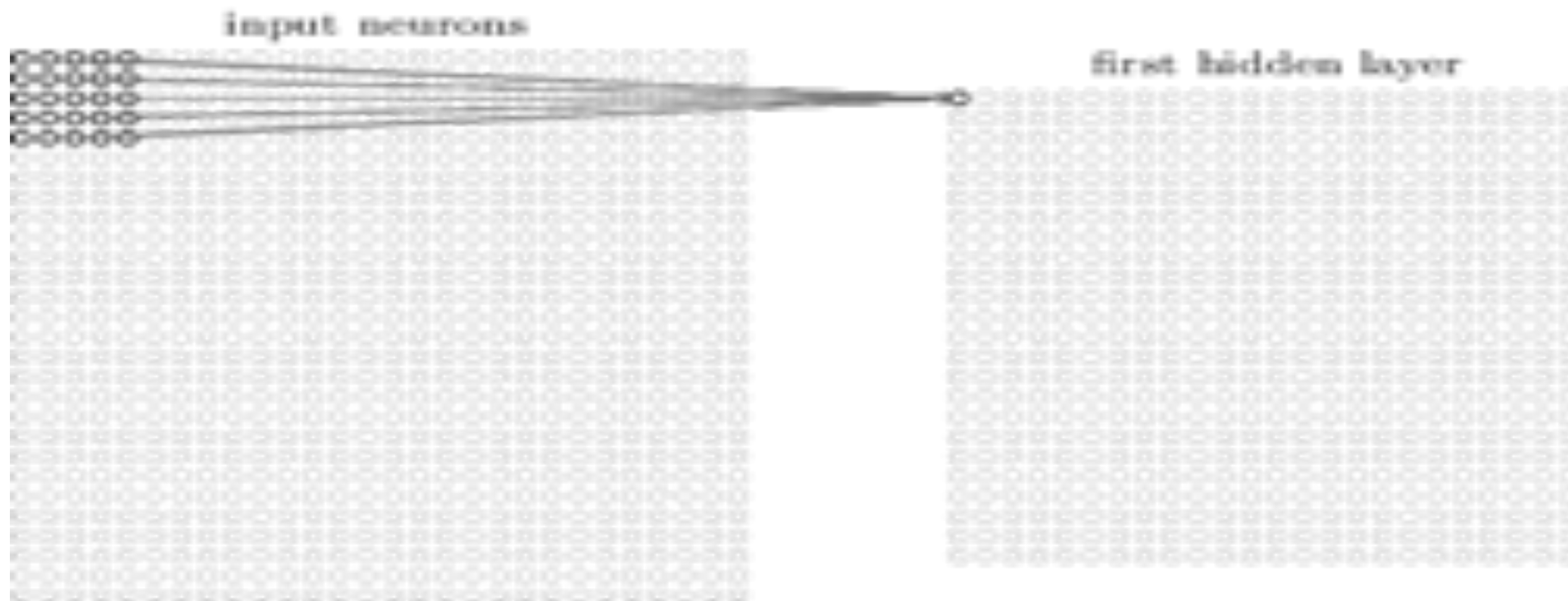
Input: 28x28 inputs (bvb. pixels)



De convolutie-operator

Input: 28x28 inputs (bv. pixels)

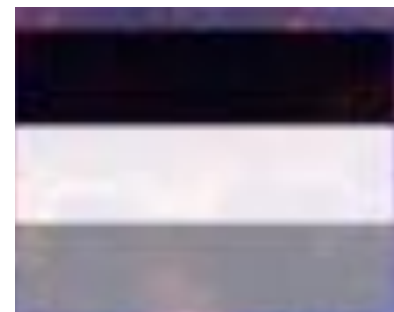
Output: 24x24 neurons



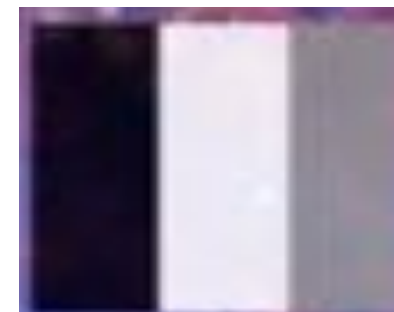
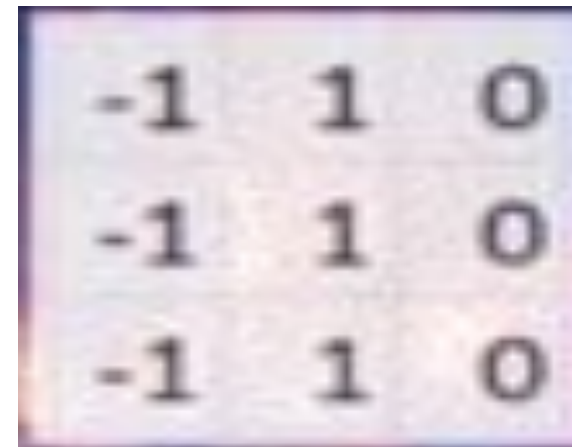
De convolutie-operator



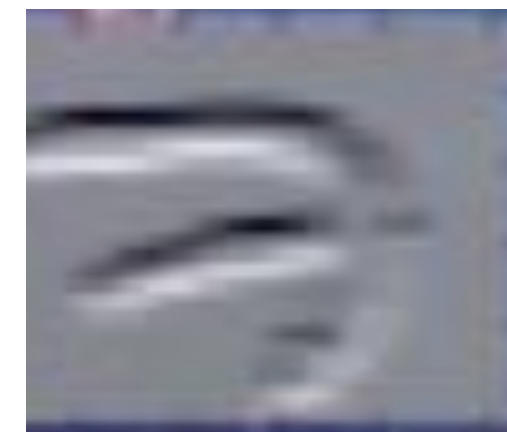
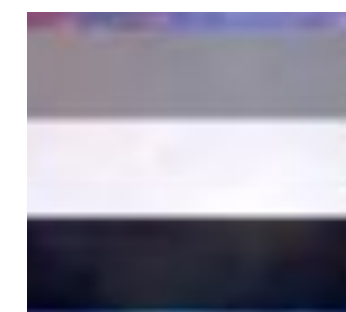
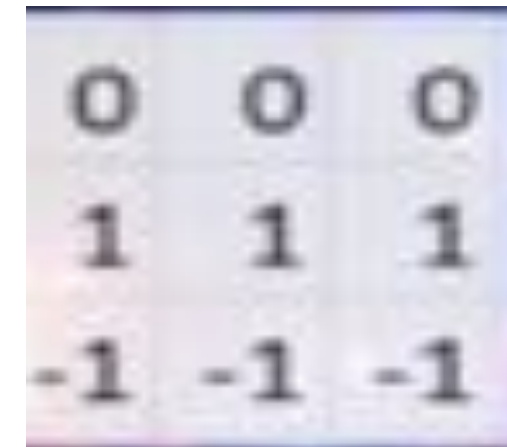
Filter 1



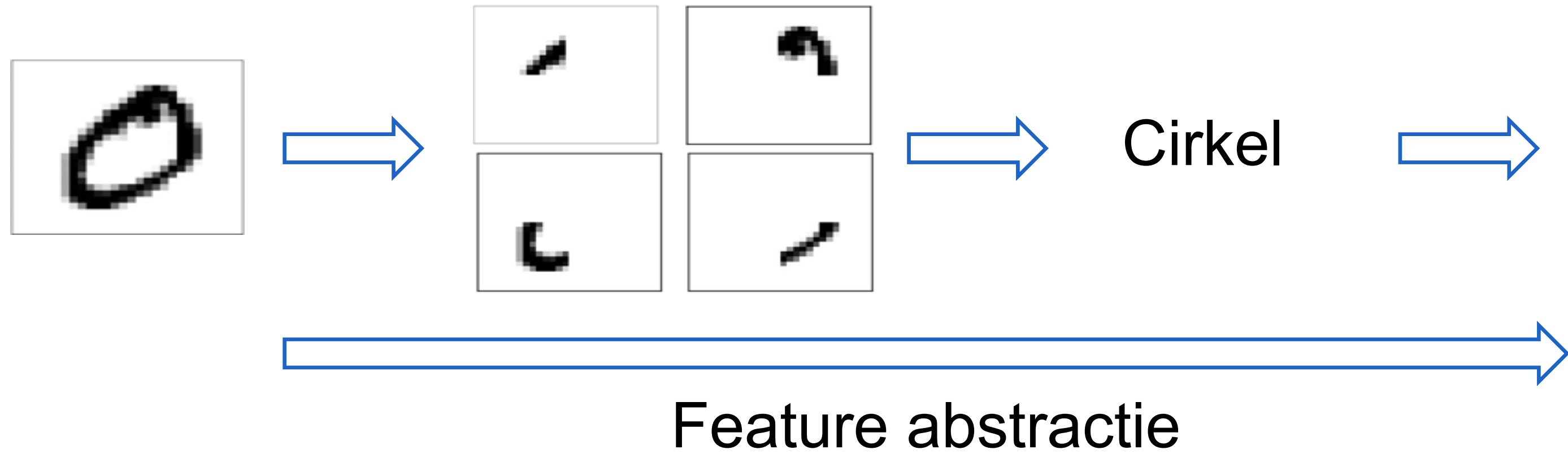
Filter 2



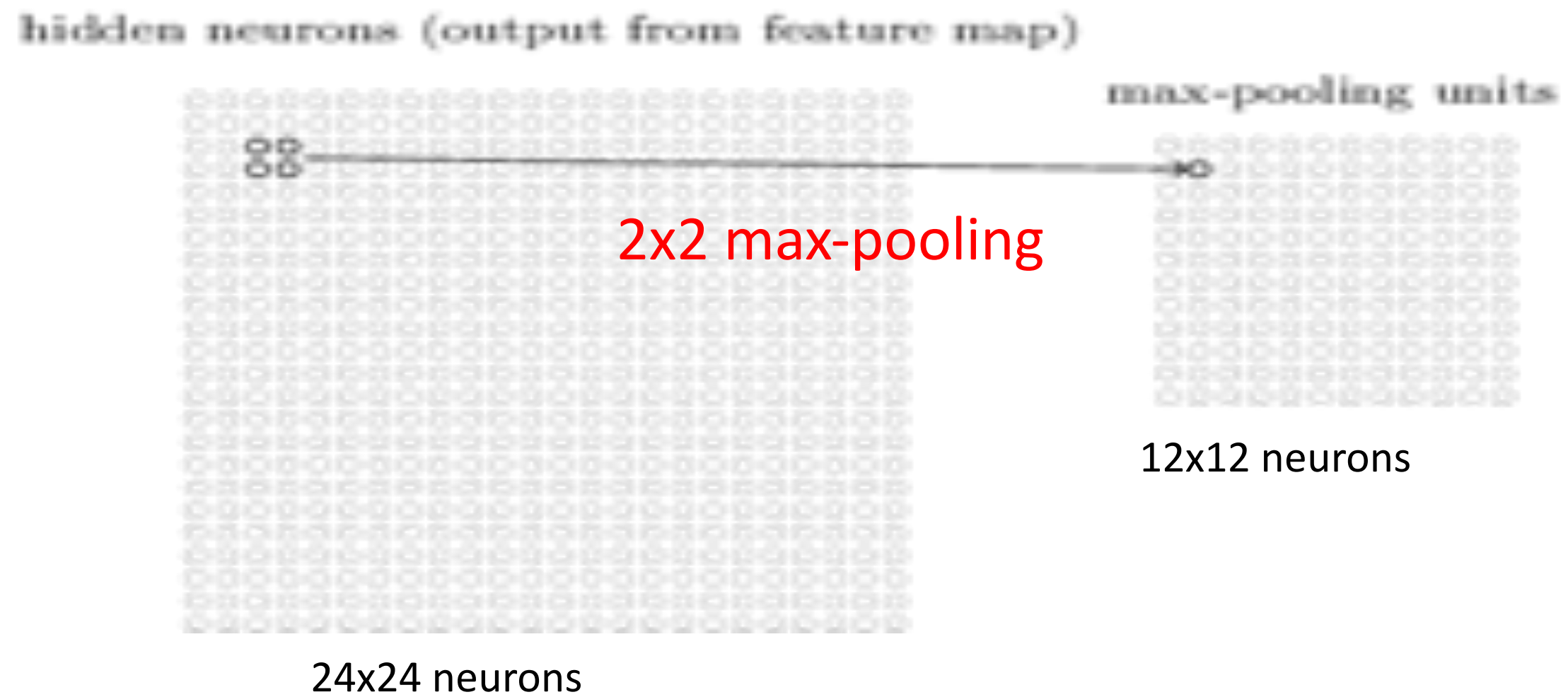
Filter 3



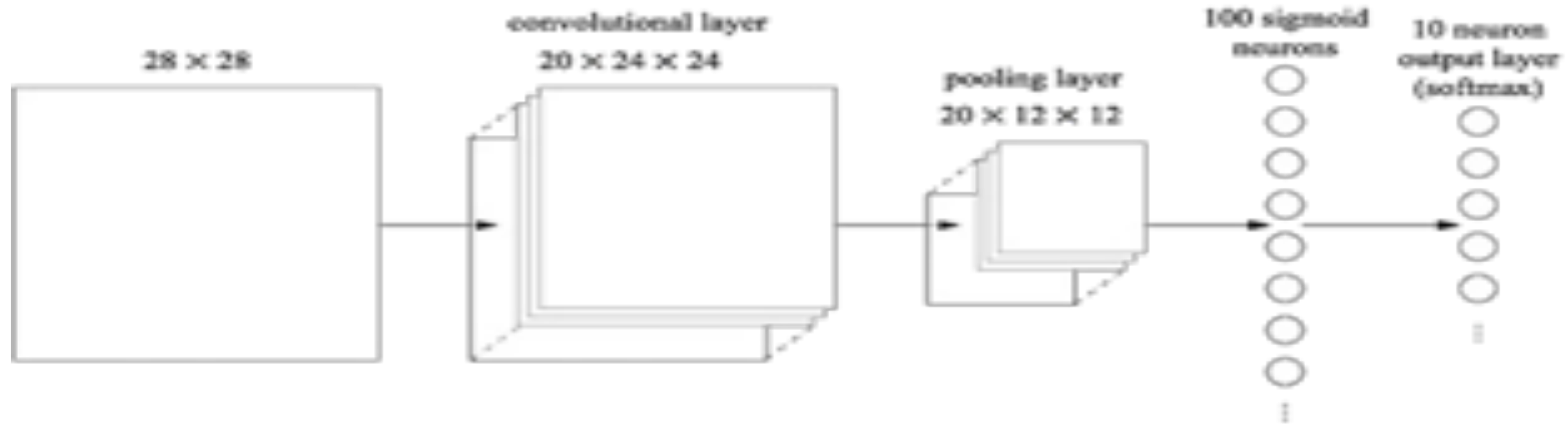
Voorbeeld



De pooling operator



Convolutionele neurale netwerken

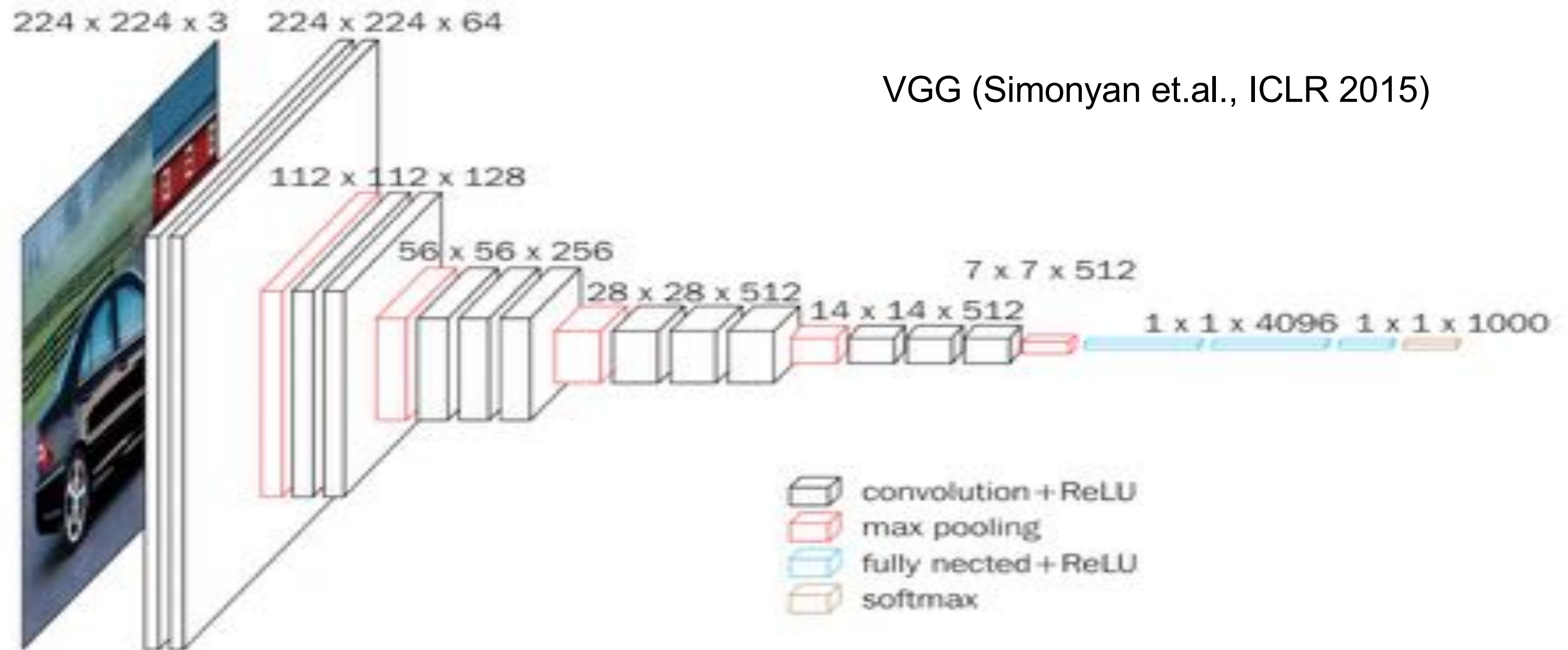


Kenmerkextractie

Output voorspelling

**Afwisselende lagen van convoluties (filters)
en pooling (positie-invariantie)**

CNN Voorbeeld



VGG (Simonyan et.al., ICLR 2015)

Winner of ImageNet 2014 challenge

State-of-the-art in vele domeinen

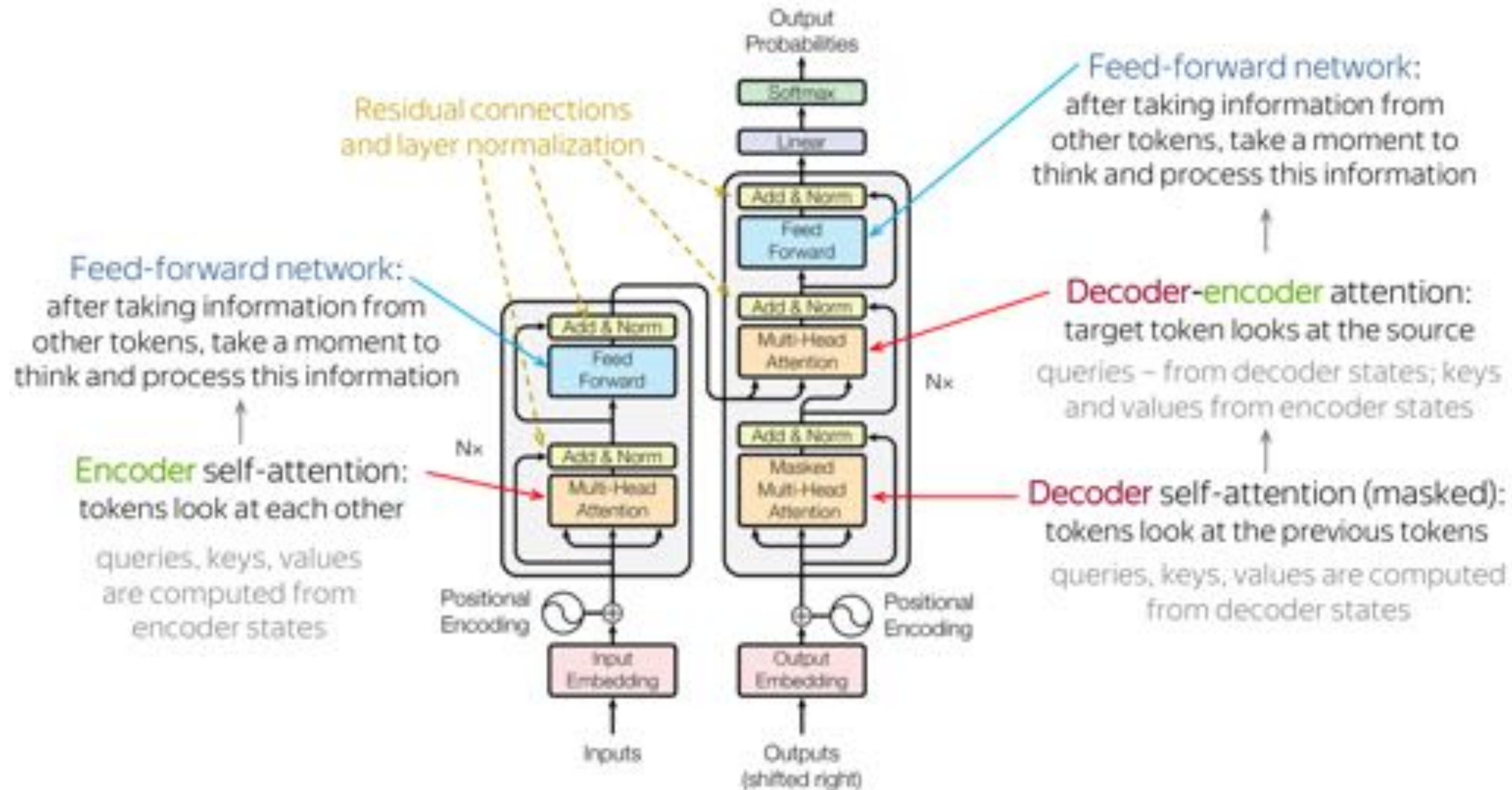
Voorbeeld: beeldherkenning

ImageNet (2009):

- 14M labeled images / 300GB
- 27 high-level classes (plant, geological formation, ...)
- 22K low-level classes (tree, grass, ice mass, monocline, ...)



Transformers (bv. ChatGPT)



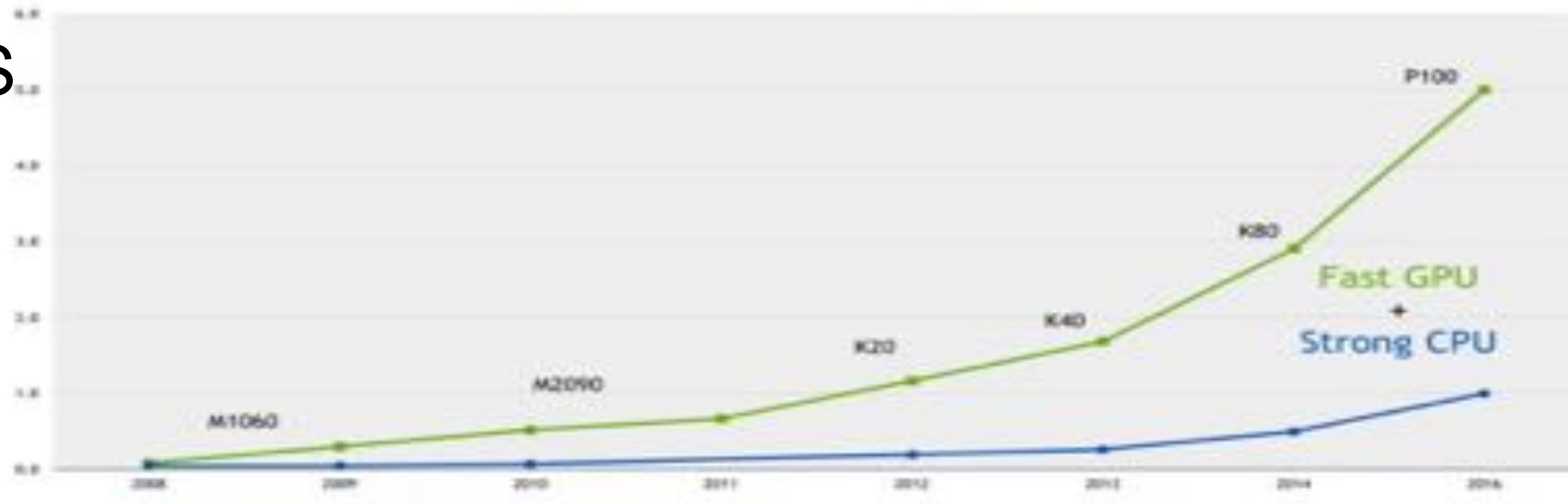
Voorbeeld van een groot taalmodel (LLM = Large Language Model)

Hoe duurzaam zijn deze technologieën?

Met de groeiende hoeveelheid data neemt ook de hoeveelheid benodigde computerkracht toe:

- Klassieke CPUs kunnen niet met deze hoeveelheid berekening om
- GPUs kunnen parallel vele berekening uitvoeren

TFLOPS



De toekomst van AI

Overall waar data is zal AI gebruikt worden

Experts predict AI to be used before 2030 in:

Traffic

Household appliances

Health care

Finances

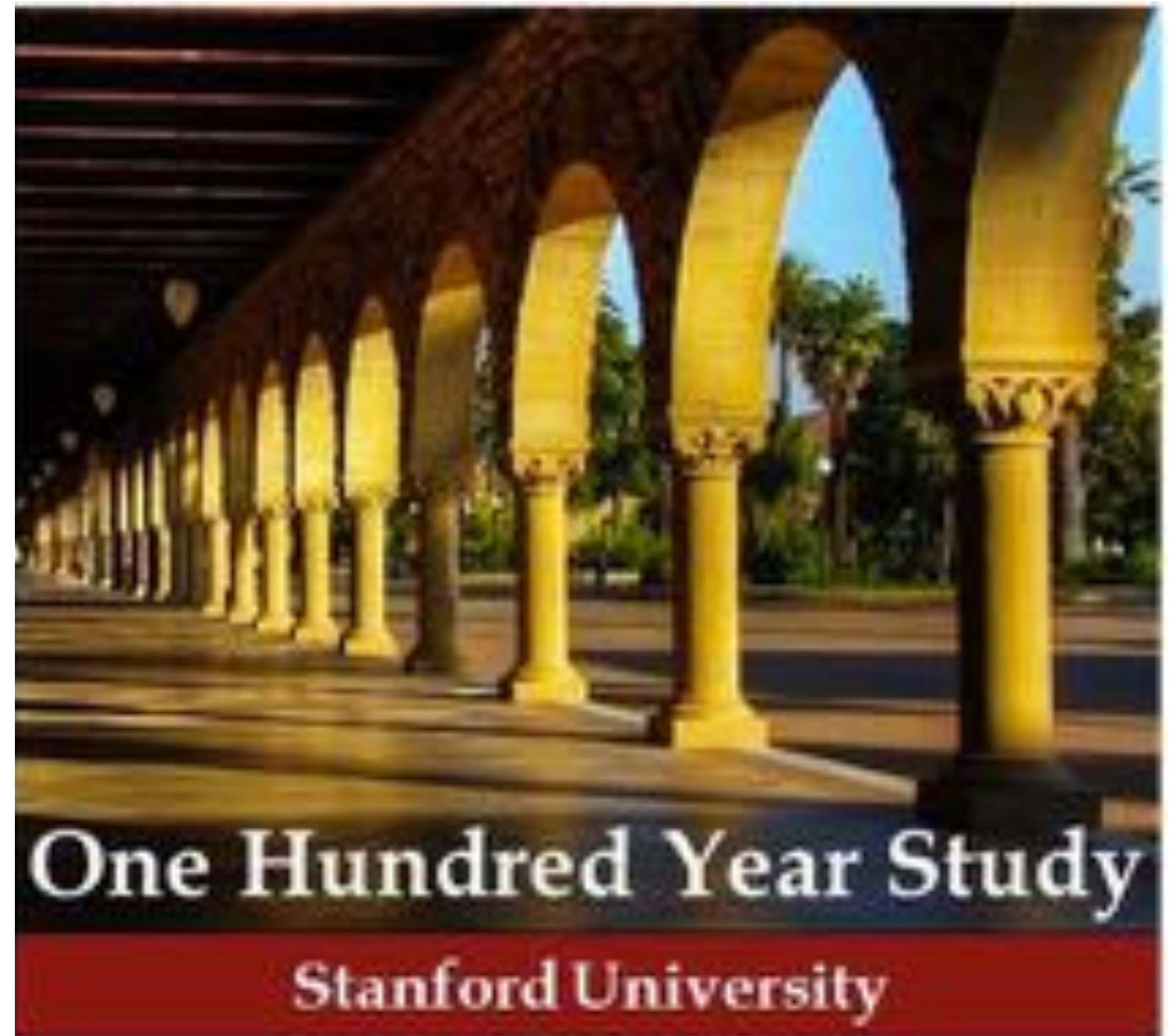
Public safety

Education

Employment

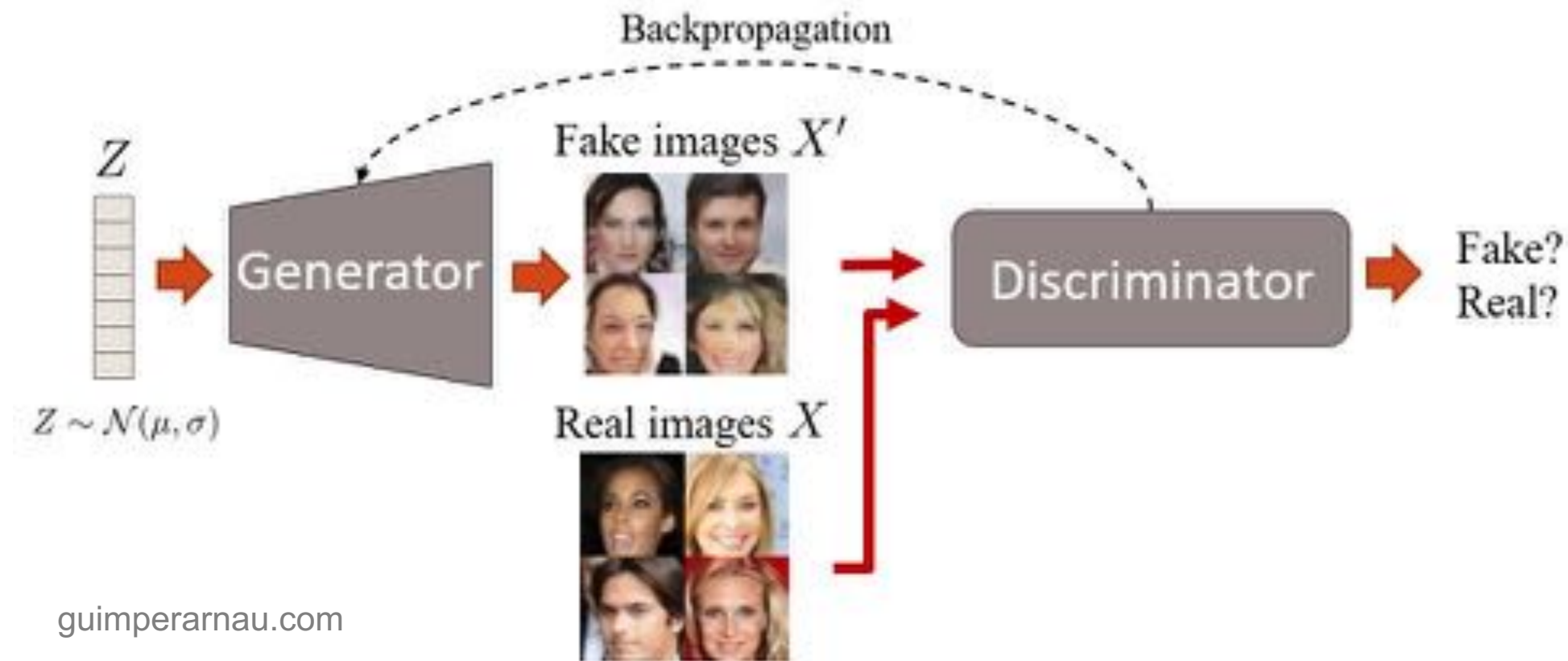
Entertainment and art

...



Opkomst van generatieve modellen (generative AI)

Generative adversarial network (GAN)



guimperarnau.com

ChatGPT

User: Tell me about when Christopher Columbus came to the US in 2015

ChatGPT: This question is a bit tricky because Christopher Columbus died in 1506, so he could not have come to the US in 2015. But let's pretend for a moment that he did!

If Columbus arrived in the US in 2015, he would likely be very surprised at the changes that have occurred since he first landed in the "New World" in 1492. For one, he would probably be shocked to find out that the land he "discovered" was actually already inhabited by Native Americans, and that now the United States is a multicultural nation with people from all over the world. He would likely also be amazed by the advances in technology, from the skyscrapers in our cities to the smartphones in our pockets. Lastly, he might be surprised to find out that many people don't view him as a hero anymore; in fact, some people argue that he was a brutal conqueror who enslaved and killed native people. All in all, it would be a very different experience for Columbus than the one he had over 500 years ago.



Risico's van AI: betrouwbaarheid

- Het wordt moeilijker om te weten wat/wie we nog kunnen vertrouwen
 - Deepfakes (beeld, klank, video, tekst,...)
 - ChatGPT kan heel realistisch “verzonnen” feiten beschrijven
- Net zoals alle computersystemen kunnen AI modellen ook gehacked worden



\mathbf{x}

“panda”

57.7% confidence

+ .007 ×



$\text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$

“nematode”

8.2% confidence

=



$\mathbf{x} +$

$\epsilon \text{sign}(\nabla_{\mathbf{x}} J(\boldsymbol{\theta}, \mathbf{x}, y))$

“gibbon”

99.3 % confidence

Risico's van AI: privacy

- Iedereen wordt constant gemonitord (bvb. verkeerscamera's, stadscamera's, smartphones)
- Predictieve modellen kunnen privacy-gevoelige informatie onthullen:
 - Bvb leeftijd/gender voorspellen op basis van tekst
 - Ziekte voorspellen op basis van camerabeelden
- Deze modellen kunnen gegevens onthullen die je niet wil
 - Nood aan sterkere wetgeving

Risico's van AI: ethische aspecten

- Uitdagingen gerelateerd aan bias/fairness
- Uitdagingen door het huidige AI landschap (veel van onze persoonlijke data is in handen van slechts een handvol grote bedrijven)
- Copyright aspecten voor (generatieve) modellen
- Meer gaan in de richting van “human-centric AI”
 - “The development of AI must be guided by a concern for its **human impact**.”
 - AI should strive to **augment** and enhance us, not replace us”

Voorbeeld van “onrechtvaardig ML” model

Error Rate_(1-PPV) By Female x Skin Type



TYPE I

TYPE II

TYPE III

TYPE IV

TYPE V

TYPE VI



1.7%

1.1%

3.3%

0%

23.2%

25.0%



11.9%

9.7%

8.2%

13.9%

32.4%

46.5%



5.1%

7.4%

8.2%

8.3%

33.3%

46.8%

“By far the greatest danger of Artificial Intelligence,” AI theorist Eliezer Yudkowsky has observed, “is that people conclude too early that they understand it.”

Voor wie meer wil lezen

- Artificiële intelligentie:
Naar een vierde
industriële revolutie?

- <https://www.kvab.be/nl/standpunten/artificiële-intelligentie>

- Stanford One Hundred
Year study

- <https://ai100.stanford.edu>

