Rekenen met neuronen 1b: Attractor networks

Fleur Zeldenrust

Van Perceptie tot Bewustzijn, 2017

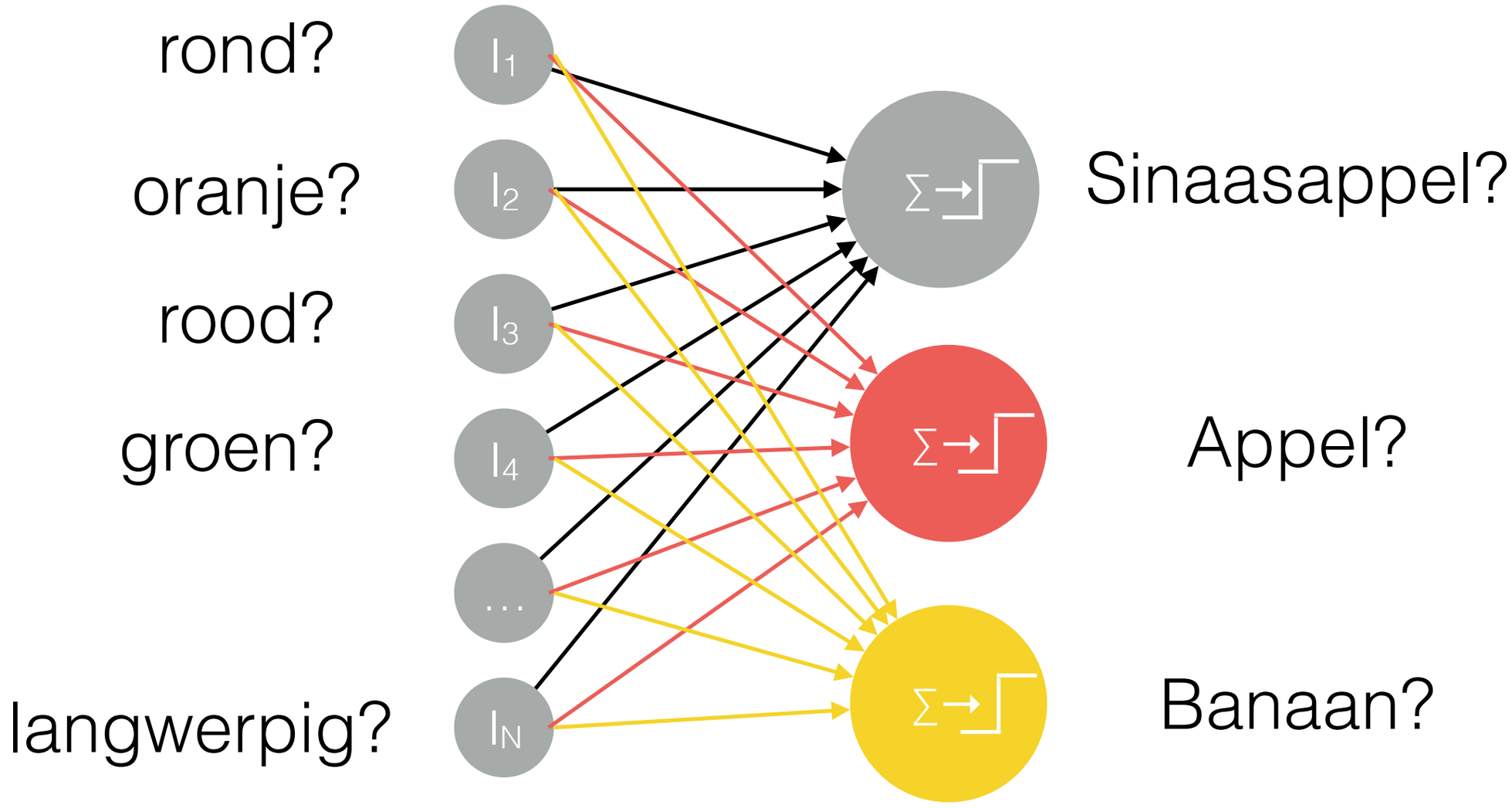
College 1a/b

- Introductie neurale netwerken en neural coding

- Encoding modellen
    - binair neuronmodel
    - Booleaanse logica
    - Perceptron

- local versus distributed codes
- firing rate neuronmodel
- recurrente netwerken
    - Hopfield
    - attractor

Klassiek Perceptron



College 1a/b

- Introductie neurale netwerken en neural coding

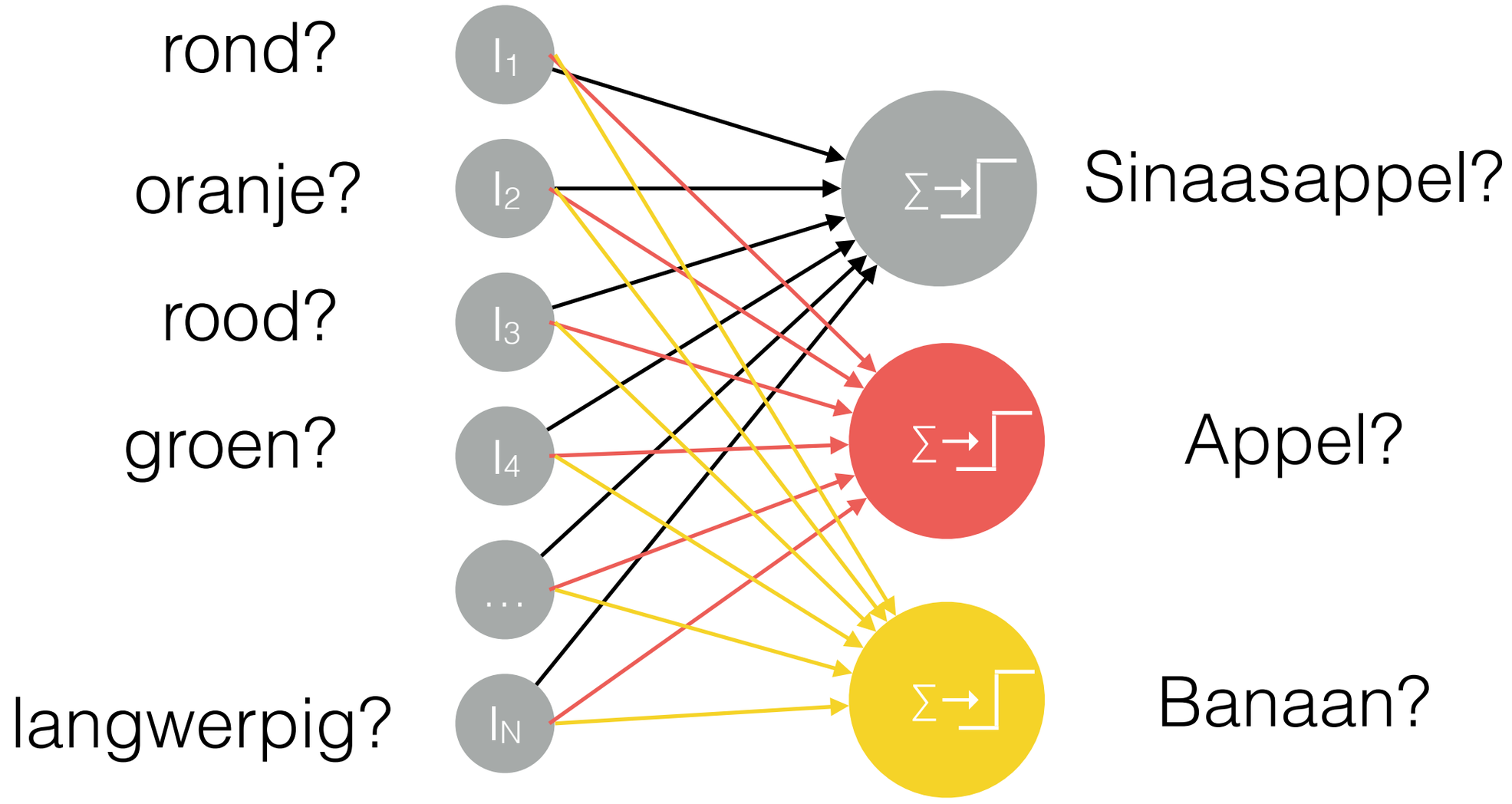
- Encoding modellen
    - binair neuronmodel
    - Booleaanse logica
    - Perceptron

- local versus distributed codes
- firing rate neuronmodel
- recurrente netwerken
    - Hopfield
    - attractor

'Grandmother cell' (Local code)

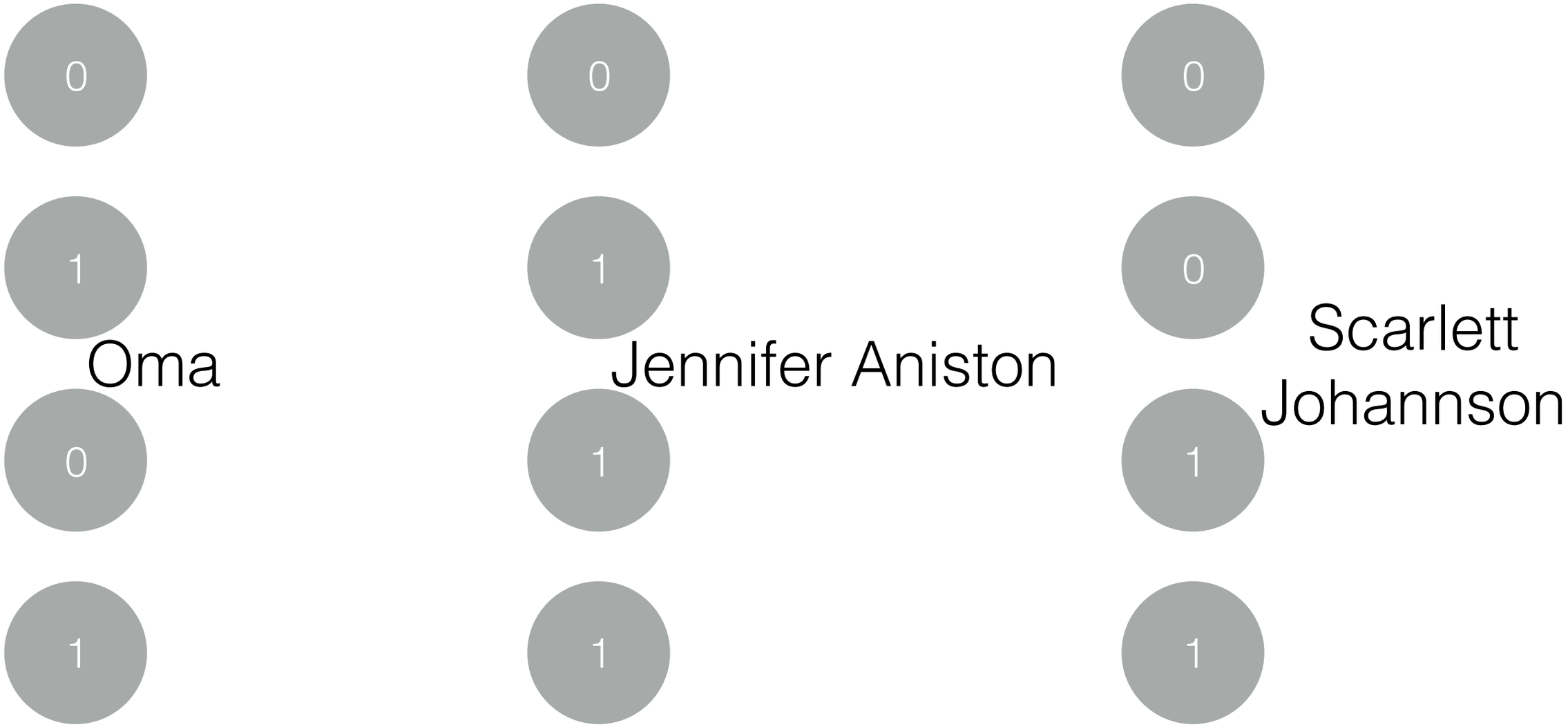
- Tot nu toe: één cel = één concept (‘grandmother’, ‘Jennifer Aniston’, ‘schuin streepje’)

- Dit betekent dat je voor elk concept een cel nodig hebt:



Distributed code

Alternatief: één concept = één aciviteitspatroon



Sparseness: hoeveel neuronen zijn actief in elk patroon?

Capaciteit

Hoeveel concepten kan ik opslaan met het aantal neuronen dat ik heb?

- Local code (LC):

  capaciteit = # neuronen

- Distributed code (DC): (combinatoriek!)

  dense: capaciteit = $2^{\#\text{ neuronen}}$

  sparse: tussen LC en DDC

Bewijs voor grandmother cells?

Quiroga et al. 2005

'Grandmother cell' (Local code)

- Tot nu toe: één cel = één concept ('grandmother', 'Jennifer Aniston', 'schuin streepje')

- Dit betekent dat je voor elk concept een cel nodig hebt:

    - **sparse**: weinig cellen tegelijkertijd actief
    - lage **capaciteit**: veel cellen nodig (voor elke cel één)

Generalisatie

- Local code: ‘zwarte auto’ cel, ‘witte auto’ cel…wat als je een grijze auto ziet?

- Distributed code: Je kunt aangeven dat patronen ‘op elkaar lijken’: 111000 lijkt op 110000

- Zo kun je **generaliseren**: patronen herkennen die je niet eerder gezien hebt

- Maar je introduceert ook het ‘**binding problem**’ en **interferentie**

Binding problem

Groen vierkant

Groen vierkant en rode driehoek
of
Rood vierkant en groene driehoek?

Interferentie

Hoe herken ik het verschil tussen A en B&C?

A	B	C
0	0	0
1	1	0
0	0	0
1	0	1

'Grandmother cell' (Local code)

- Tot nu toe: één cel = één concept ('grandmother', 'Jennifer Aniston', 'schuin streepje')

- Dit betekent dat je voor elk concept een cel nodig hebt:

    - **sparse**: weinig cellen tegelijkertijd actief
    - lage **capaciteit**: veel cellen nodig (voor elke cel één)
    - **generalisatie** versus **specificiteit**: je kunt niet uitdrukken dat twee concepten 'op elkaar lijken'
    - Maar geen **interferentie**

Local vs distributed coding

local	**sparse distributed**	**dense distributed**
sparse	sparse	niet sparse
lage capaciteit (N)	gemiddelde capaciteit	hoge capaciteit (2^N)
zeer selectief	generalisatie goed mogelijk	generalisatie goed mogelijk
geen interferentie	weinig interferentie	binding problem, interferentie

Meer info: scholarpedia ‘Sparse Coding'

Robuustheid

Wat gebeurt er als er iets ‘mis gaat’ in het systeem?

Verschillende soorten robuustheid:

- Robuustheid tegen fouten in de input (Hoe kan ik een geschreven 2 herkennen in verschillende handschriften?)
    - Perceptron: zie Kandel
    - Distributed code: gaan we nu naar kijken

- Robuustheid tegen celdood
    - Perceptron: groot probleem als ‘grandmother cell’ doodgaat
    - Distributed code: hangt er vanaf hoeveel patronen bij hetzelfde concept horen

College 1a/b

- Introductie neurale netwerken en neural coding

- Encoding modellen
    - binair neuronmodel
    - Booleaanse logica
    - Perceptron

- local versus distributed codes
- firing rate neuronmodel
- recurrente netwerken
    - Hopfield
    - attractor

Klassiek Perceptron

- Model van herkennen van objecten in visuele systeem
- Abstract neuron: discrete tijd (actief/niet-actief), sommeren input en drempel
- Feedforward netwerk: verbindingen één kant op
- Door meerdere lagen kun je alle Booleaanse functies maken
- Lijkt op visuele systeem: steeds meer abstractie

Problemen Perceptron

Er zijn veel *recurrente* verbindingen in de cortex

Discreet neuron model (alleen 0 of 1).

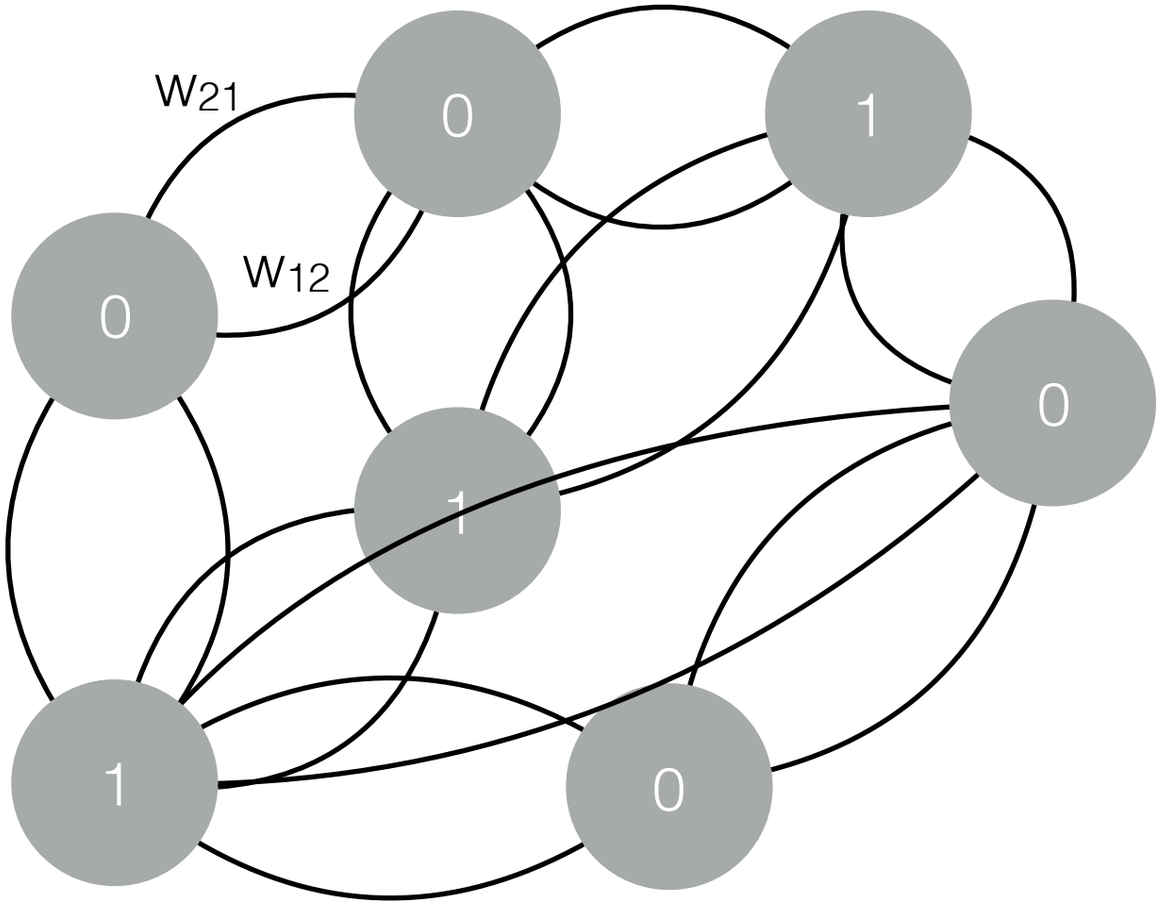
Problemen Perceptron

Er zijn veel recurrente verbindingen in de cortex

Discreet neuron model (alleen 0 of 1).

Hoe kun je iets (Jennifer Aniston) herkennen met een recurrent netwerk?

Hopfield network (1982)

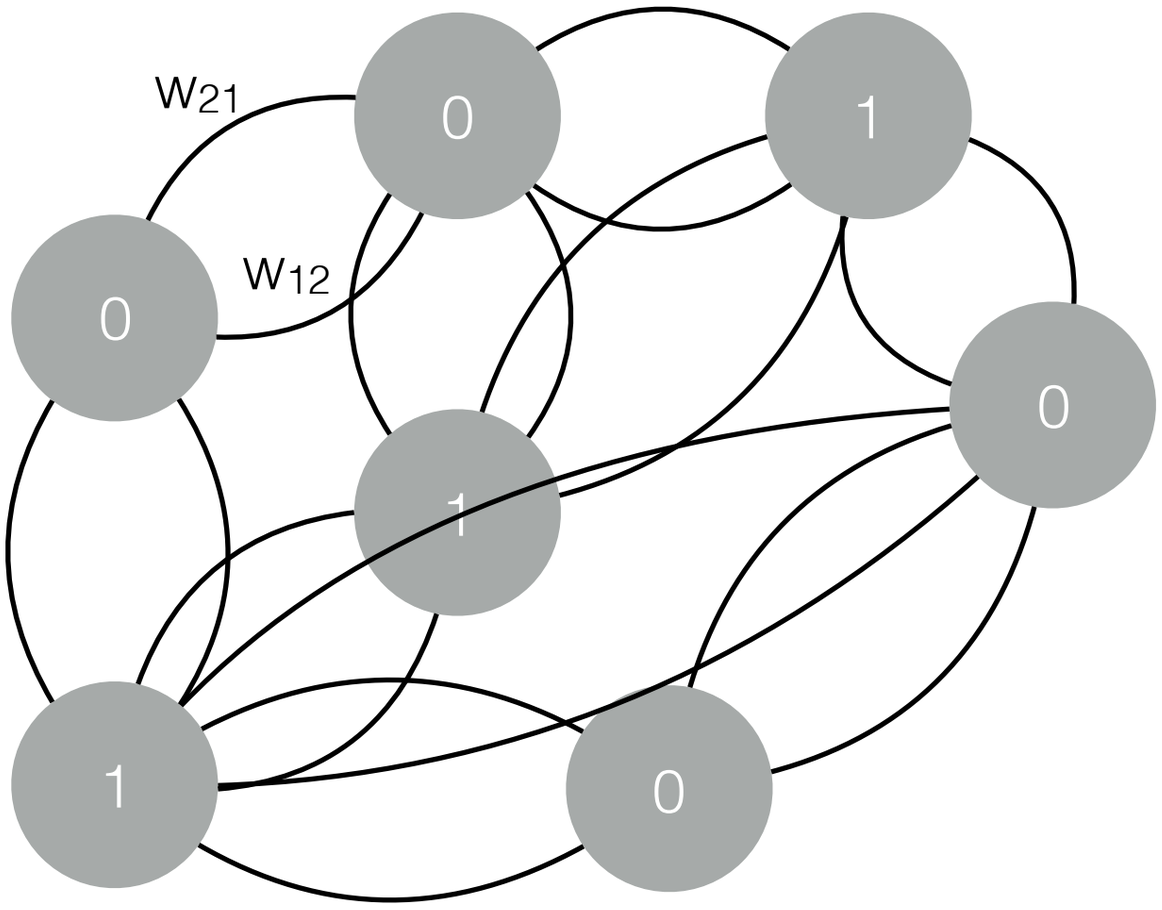


$$x_j = H(w_{1j}x_1 + w_{2j}x_2 + ... - T_j)$$

$$= H(\sum_{i=1}^{N} w_{ij}x_i - T_j)$$

- Neuronen: MPN, activiteit neuron n is x_j (0 of 1)
- Alle verbindingen symmetrisch: $w_{21} = w_{21}$
- Geen zelf-verbindingen : $w_{kk} = 0$

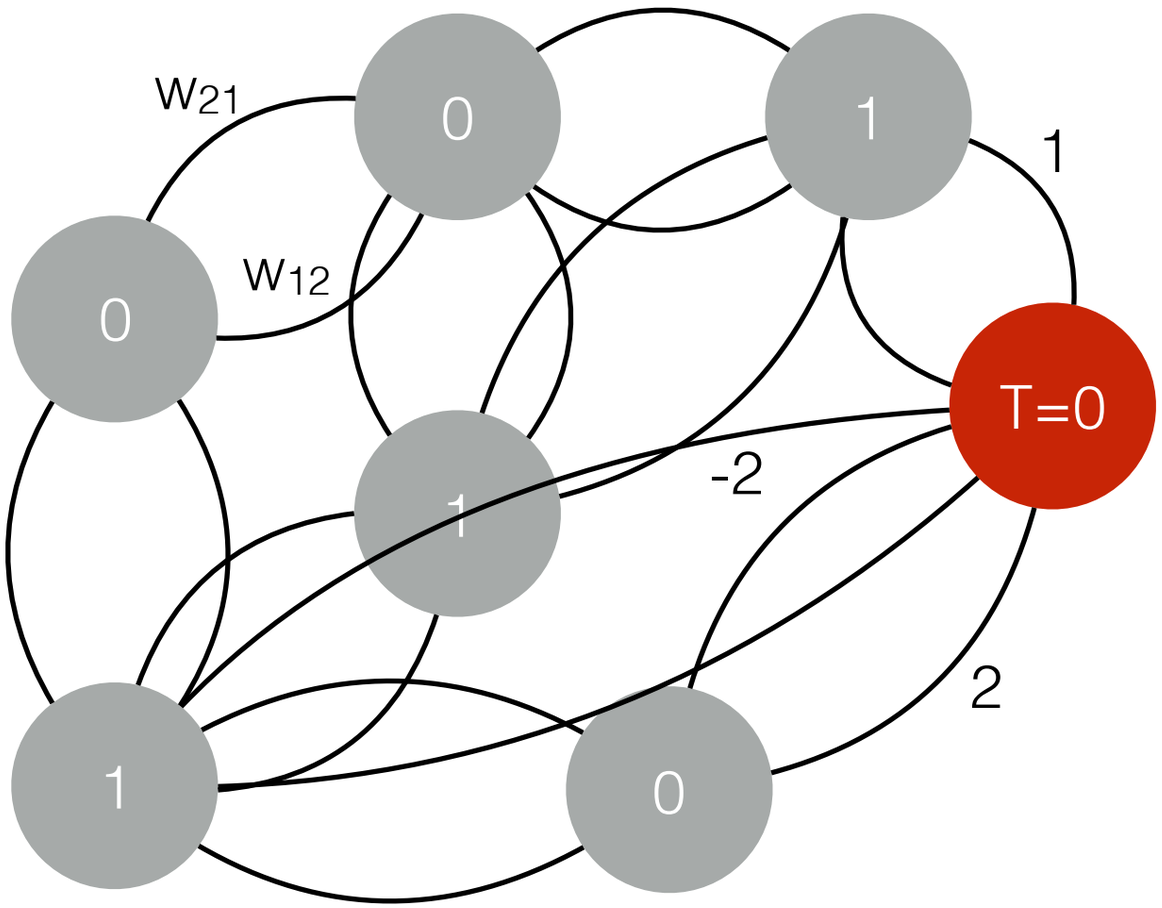
Hopfield network (1982)



1. Maak netwerk
   - kies verbindingen
   - kies drempelwaarden (T)
2. kies een input-patroon:
   - geef elk neuron activiteit (x=0 of x=1)
3. Updating:
   - Klopt de waarde (x) met de input?
   - één voor één (of tegelijk)

$$x_j(t_{n+1}) = H(\sum_i w_{ij} x_i(t_n) - T_j)$$

Hopfield network (1982)

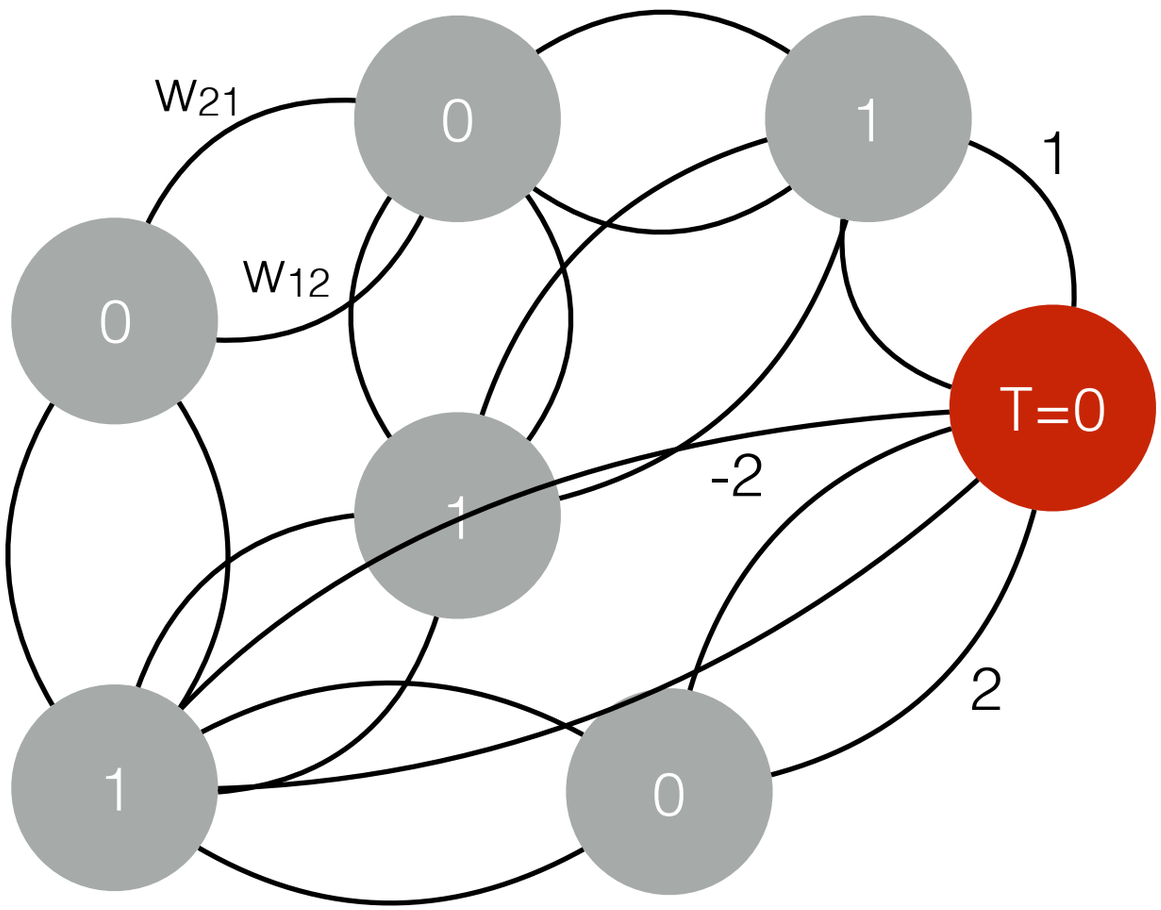


Voorbeeld:

Wat zal de activiteit van het rode neuron zijn?

$$x_j(t_{n+1}) = H(\sum_i w_{ij} x_i(t_n) - T_j)$$

Hopfield network (1982)



Voorbeeld:

Wat zal de activiteit van het rode neuron zijn?

Antwoord:
2*0 + 1*-2 + 1*1 - 0 = -1

→ inactief!

$$x_j(t_{n+1}) = H(\sum_i w_{ij} x_i(t_n) - T_j)$$

Hopfield network (1982)

- Een Hopfield netwerk convergeert altijd naar een stabiel patroon (x verandert niet meer)

- Dit is een minimum van de **energie-functie**

$$E = -\frac{1}{2}\sum_{ij} w_{ij} x_i x_j + \sum_j x_j T_j$$

- E neemt af of blijft gelijk bij elke update

- (bewijs als achterin de slides, dus Blackboard)

Robuust: Pattern completion

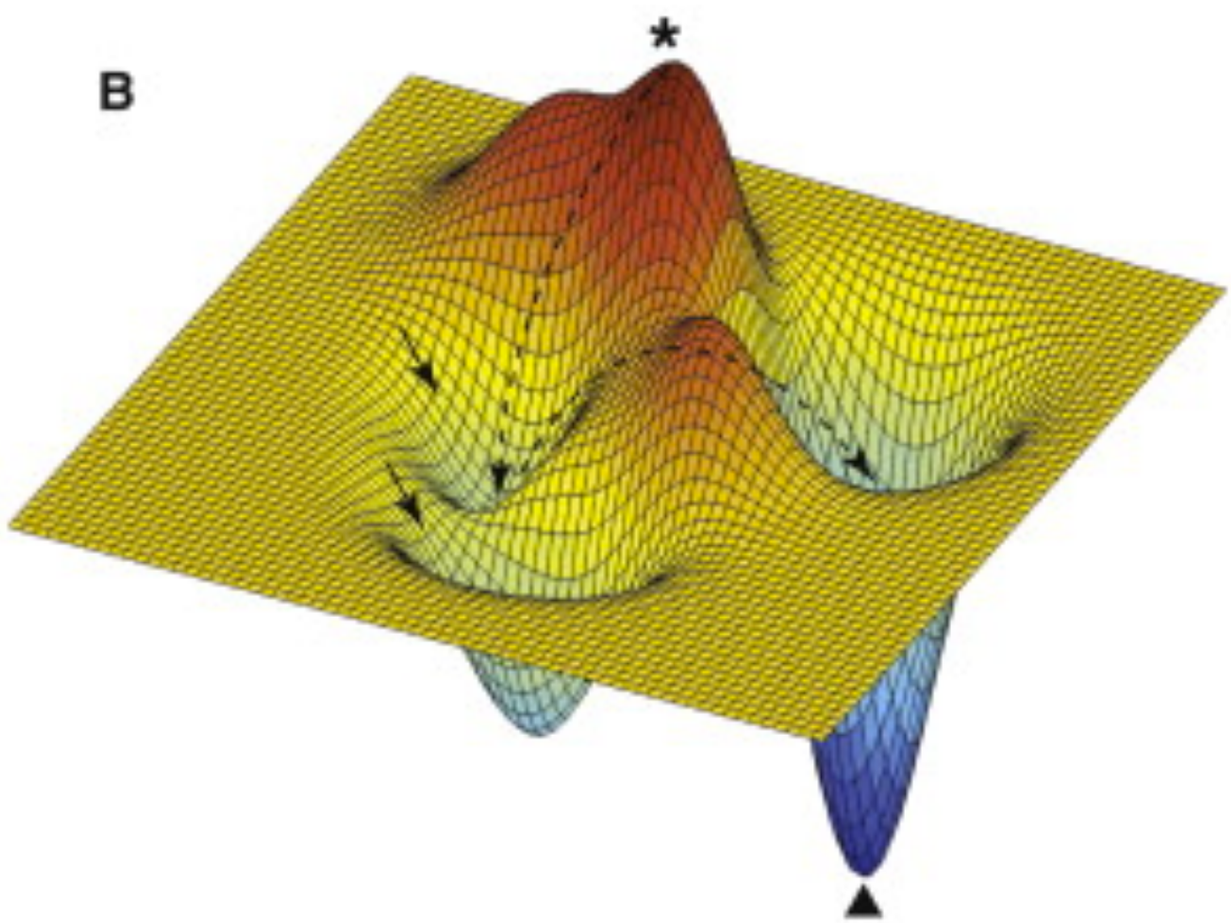
- Dus: Hopfield netwerk heeft input-output associaties (associative memory)

- Meerdere input patronen die op elkaar lijken geven hetzelfde output-patroon: generalisatie

- Dus deze netwerken zijn robuust tegen input-patronen met fouten!

- Bovendien: met grote netwerken redelijk robuust tegen celdood

‘Energy landscape'
Locale en globale minima



$$E = -\frac{1}{2}\sum_{ij} w_{ij} x_i x_j + \sum_j x_j T_j$$

Robuust: Pattern completion

- Dus: Hopfield netwerk heeft input-output associaties (associative memory)

- Meerdere input patronen die op elkaar lijken geven hetzelfde output-patroon

- Dus deze netwerken zijn robuust tegen input-patronen met fouten!

- Bovendien: met grote netwerken redelijk robuust tegen celdood

- Maar: ze kunnen ‘vastzitten’ in locale minima

- Netwerk zo instellen dat ‘Energy landscape’ klopt zeer lastig probleem

Samenvatting Hopfield netwerk

- Model van herkennen objecten / geheugen
- Zelfde abstracte neuronen als Perceptron
- Distributed coding
- Pattern completion: robuust tegen fouten in input
- Maar pas op voor locale minima
- Minder selectief dan perceptron, maar beter in generaliseren

Recurrent networks

- Het Hopfield network is een voorbeeld van een **Recurrent Network**:
  - netwerken met ‘heen en weer’ verbindingen (er is tenminste één ‘cirkel' die je kunt doorlopen)

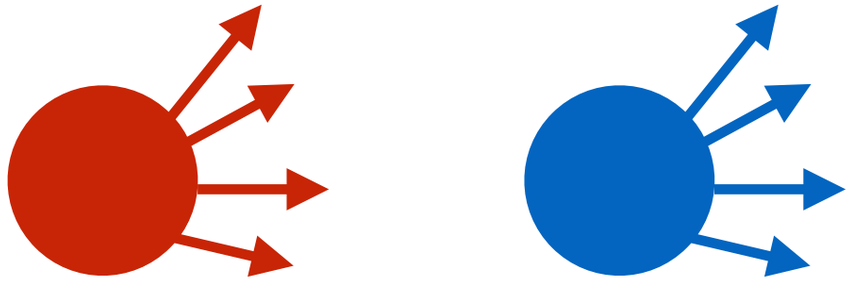
- (Vrijwel) alle biologische netwerken zijn recurrent!

- Recurrente netwerken kun je leren patronen in de tijd te herkennen (feedforward netwerken niet)

- In het kort: recurrente netwerken kunnen meer, maar zijn lastiger om mee te werken

Maar…Hopfield biologisch niet realistisch

Symmetrische verbindingen

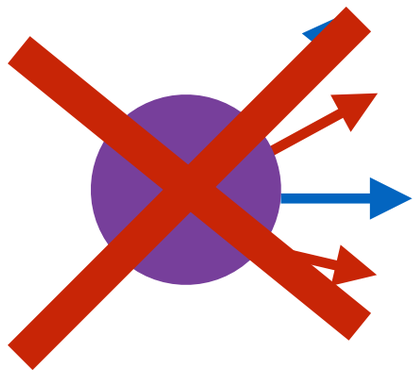


Dale's law: een neuron is excitatoir of inhibitoir

" (…) the nature of the chemical function, whether cholinergic or adrenergic, is characteristic for each particular neurone, and unchangeable."

Oftewel: een neuron maakt alleen óf GABA, óf glutamaat, niet allebei

(is nog discussie over)



Maar…Hopfield biologisch niet realistisch

Symmetrische verbindingen

Dale's law: een neuron is excitatoir of inhibitoir

Discreet neuron model (alleen 0 of 1).

Dus…

- Maak tijd weer continu
- In plaats van wel of geen actiepotentiaal: vuurfrequentie
- In plaats van 'harde', 'zachte' drempel

College 1a/b

- Introductie neurale netwerken en neural coding

- Encoding modellen
    - binair neuronmodel
    - Booleaanse logica
    - Perceptron

- local versus distributed codes
- firing rate neuronmodel
- recurrente netwerken
    - Hopfield
    - attractor

Firing rate model neuron

- Een neuron i wordt beschreven door zijn vuurfrequentie op elk moment in de tijd: $r_i(t)$

  (vergelijk ‘actief' of ‘niet actief’ MPN)

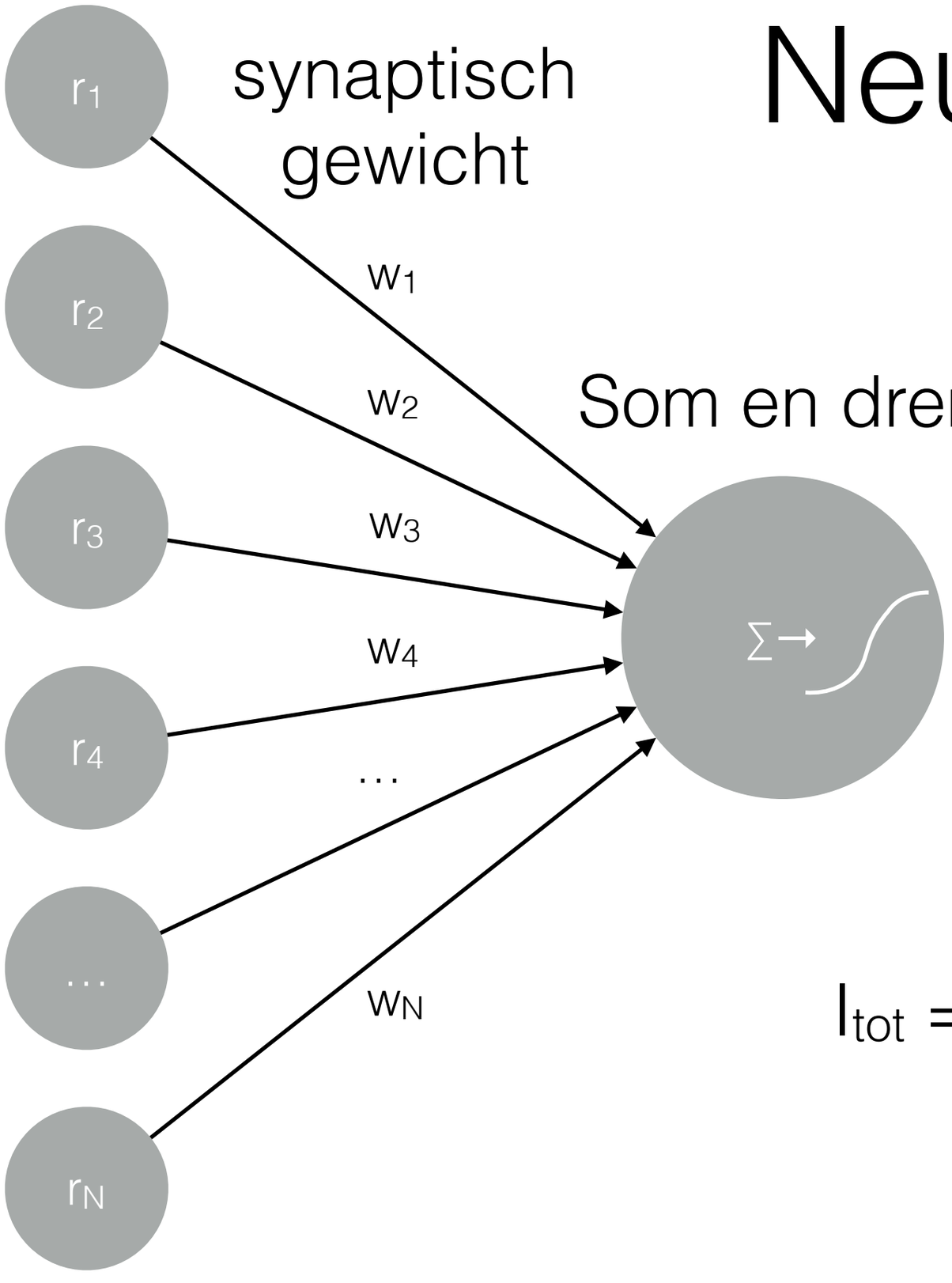
- Die vuurfrequentie hangt af van de input:

$$I_{tot} = r_1*w_1 + r_2*w_2 + \ldots r_n*w_n$$

- In plaats van ‘drempel' een functie die de input-outputrelatie beschrijft

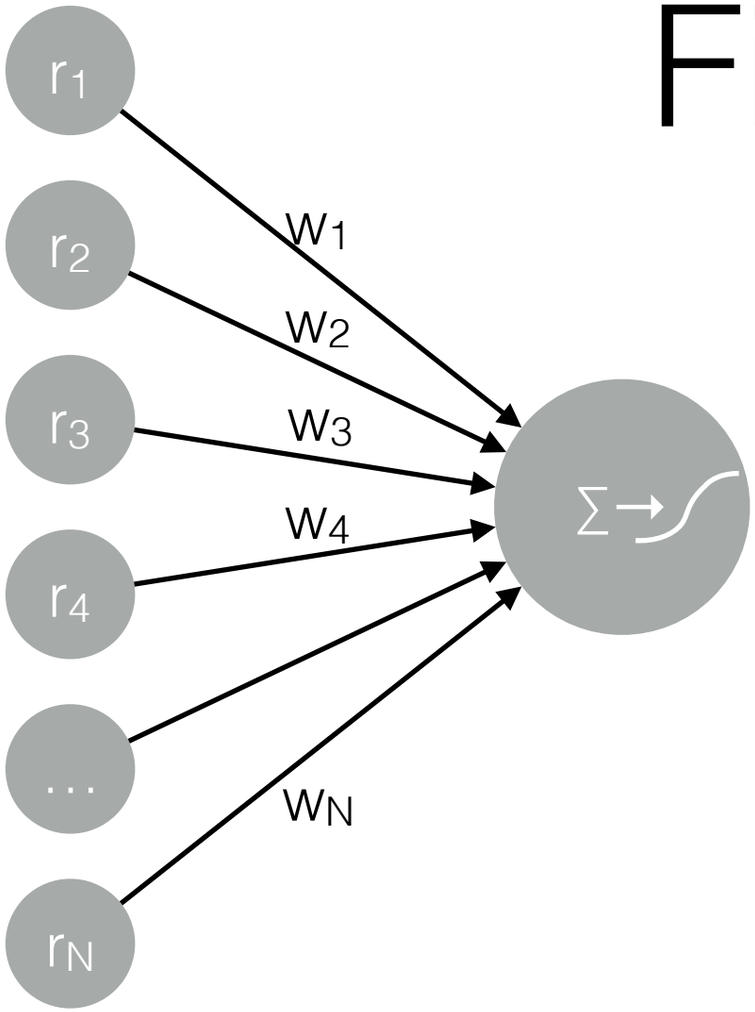
Firing Rate Neuron (FRN)

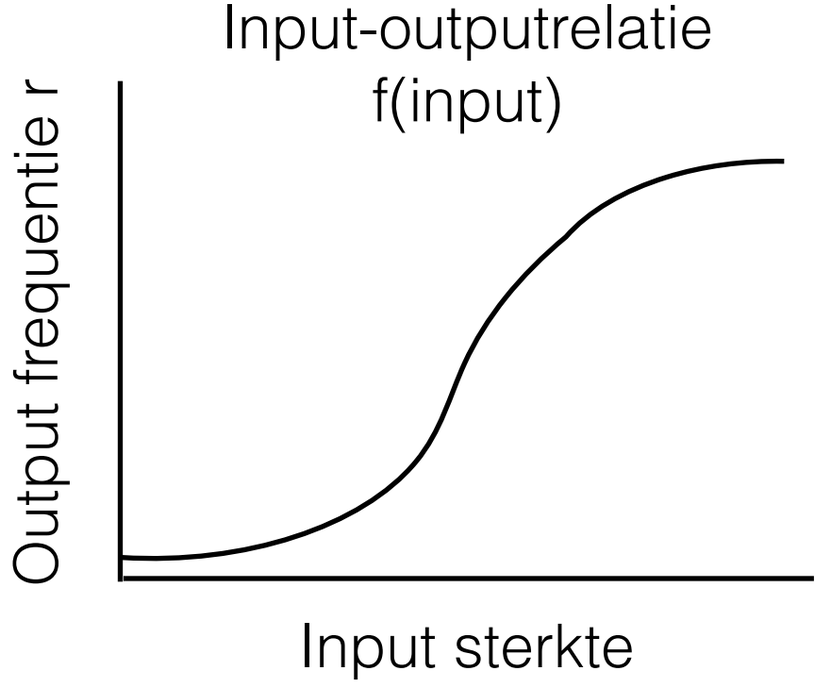
Input (frequentie)



$$I_{tot} = r_1 * w_1 + r_2 * w_2 + \ldots r_n * w_n$$

Firing Rate Neuron (FRN)





$$\tau \frac{dr_{\text{output}}(t)}{dt} = f(r_1(t) * w_1 + r_2(t) * w_2 + ..., \text{neuron parameters})$$

$$= f(\sum_{n=1}^{N} r_n(t) * w_n, \text{neuron parameters})$$

Herhaling: differentiaalvergelijkingen

- Een vergelijking waarvan de oplossing een functie is
- Bijvoorbeeld $y'(t) = k * y(t)$
- Oplossing…
- Is dat realistisch voor een neuronmodel?

Firing Rate Neuron (FRN)

Meestal van de vorm

$$\tau \frac{dr(t)}{dt} = -r(t) + f(\text{input},\ \text{params})$$

- f neemt toe met de input (tot maximum)
- f is altijd positief (frequentie!)
- r gaat terug naar 0 als er geen input is

College 1a/b

- Introductie neurale netwerken en neural coding

- Encoding modellen
    - binair neuronmodel
    - Booleaanse logica
    - Perceptron

- local versus distributed codes
- firing rate neuronmodel
- recurrente netwerken
    - Hopfield
    - attractor

Attractor networks

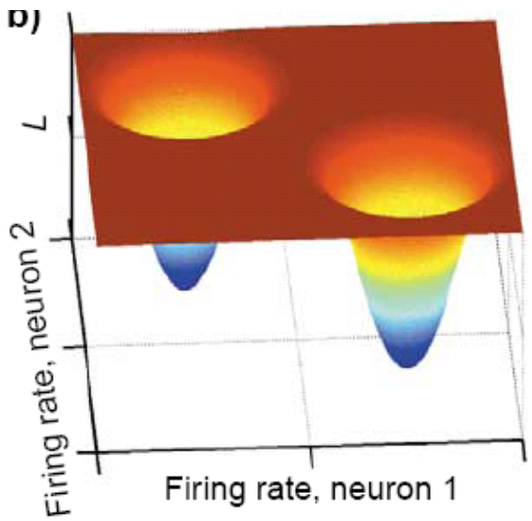
- Hopfield network is (ook) een voorbeeld van een **Attractor Network**
  - een netwerk waarvan de activiteit na een gegeven periode een vast patroon vertoont
- Er zijn verschillende soorten vaste patronen (attractors):
  - **point attractor**: elk neuron heeft vaste activiteit
    - Bv Hopfield netwerk

Line / Ring Attractor

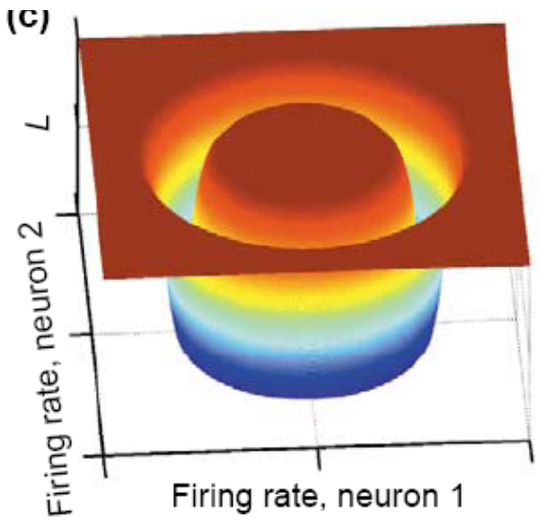
- Soms kunnen alle punten op een lijn of curve stabiel (aantrekkend) zijn

- Waar op de lijn het netwerk eindigt, hangt af van de beginwaarden

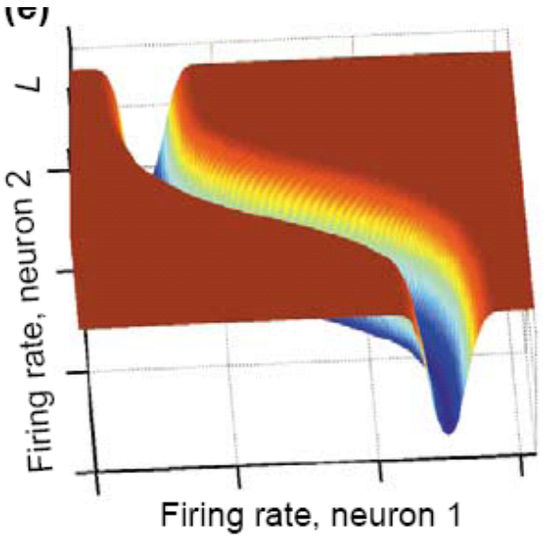
point



ring



line

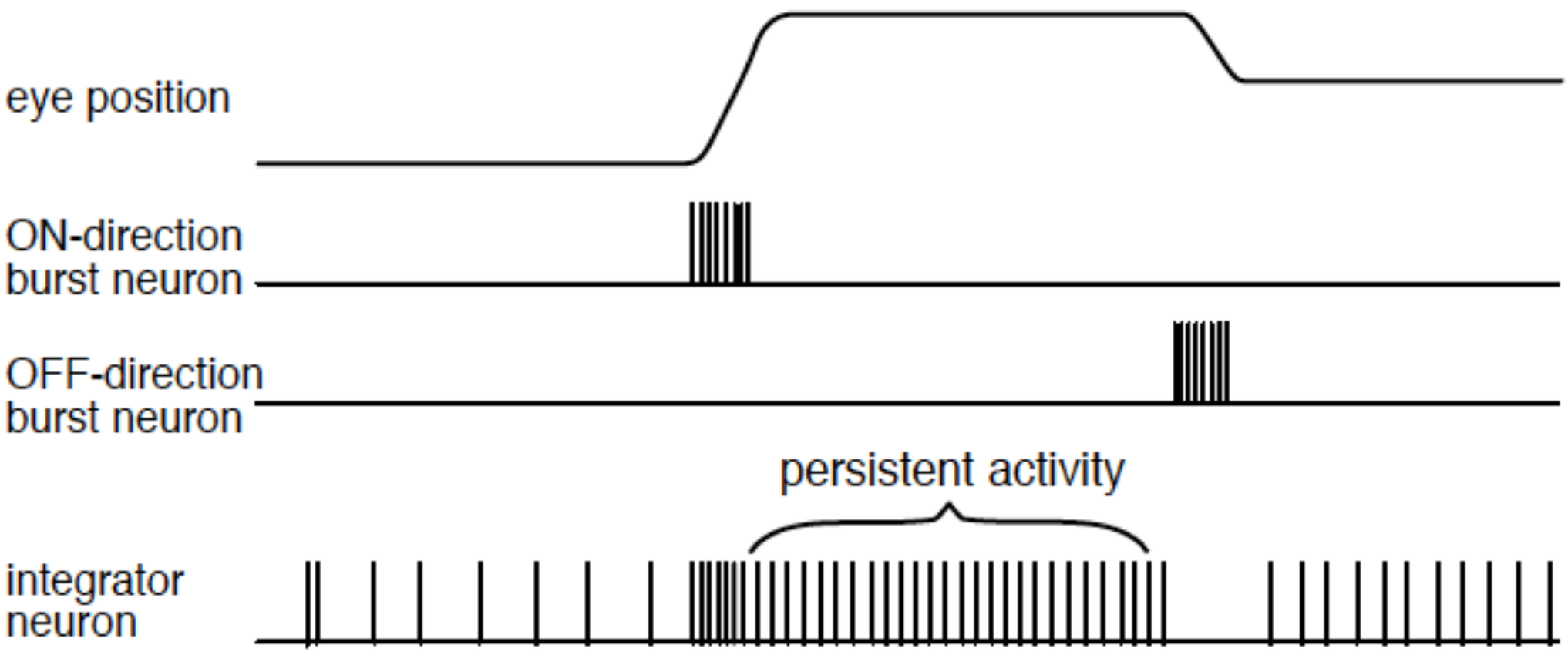


Brody et al. 2003

Voorbeeld

Oculomotor network (eye position)

Seung (2000), Dayan&Abbott, Theoretical Neuroscience



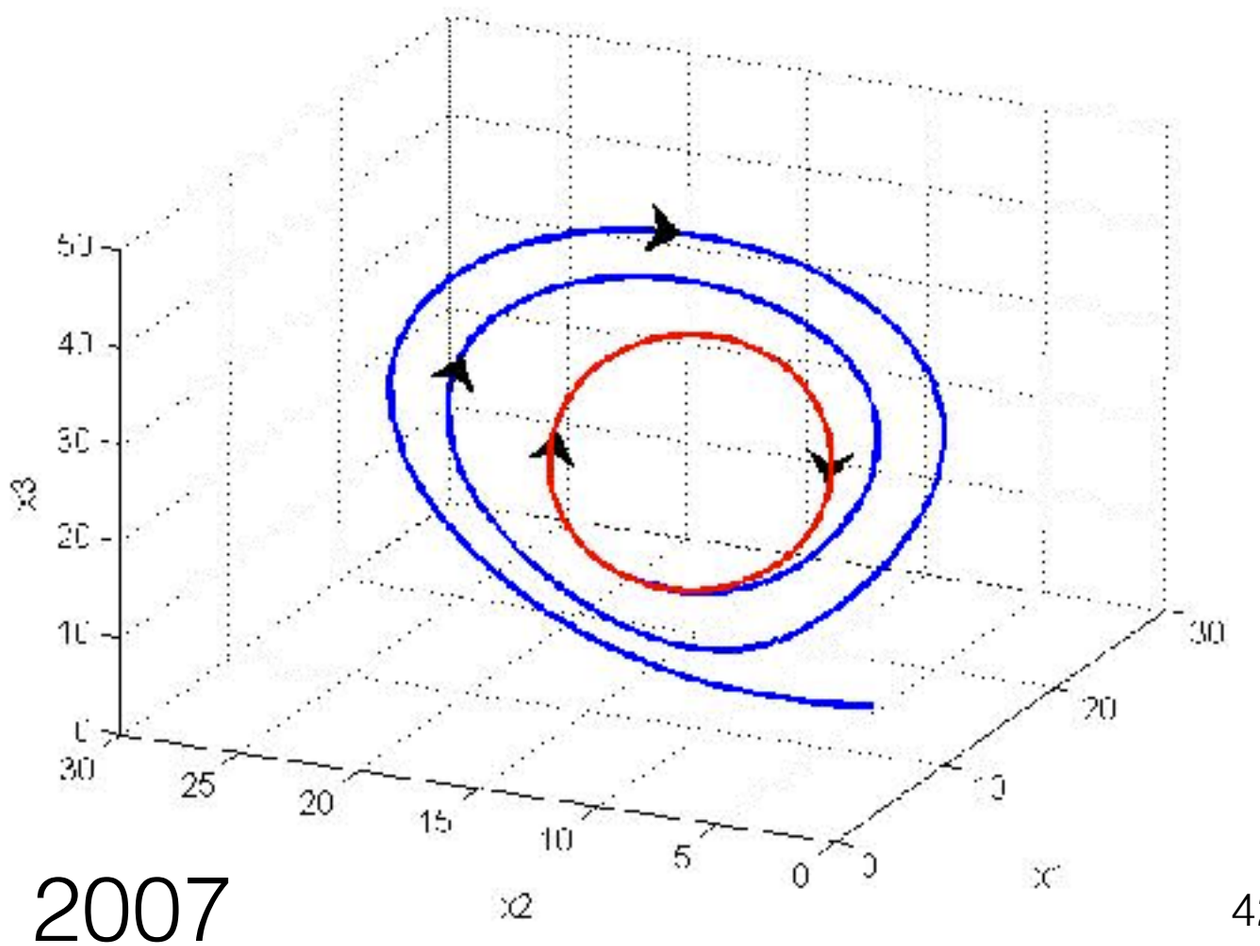
Cyclic Attractor

- Niet één stabiele oplossing, maar het netwerk blijft steeds hetzelfde paadje doorlopen

- Denk aan slinger in natuurkunde!

Neuroscience: modellen voor cyclische systemen

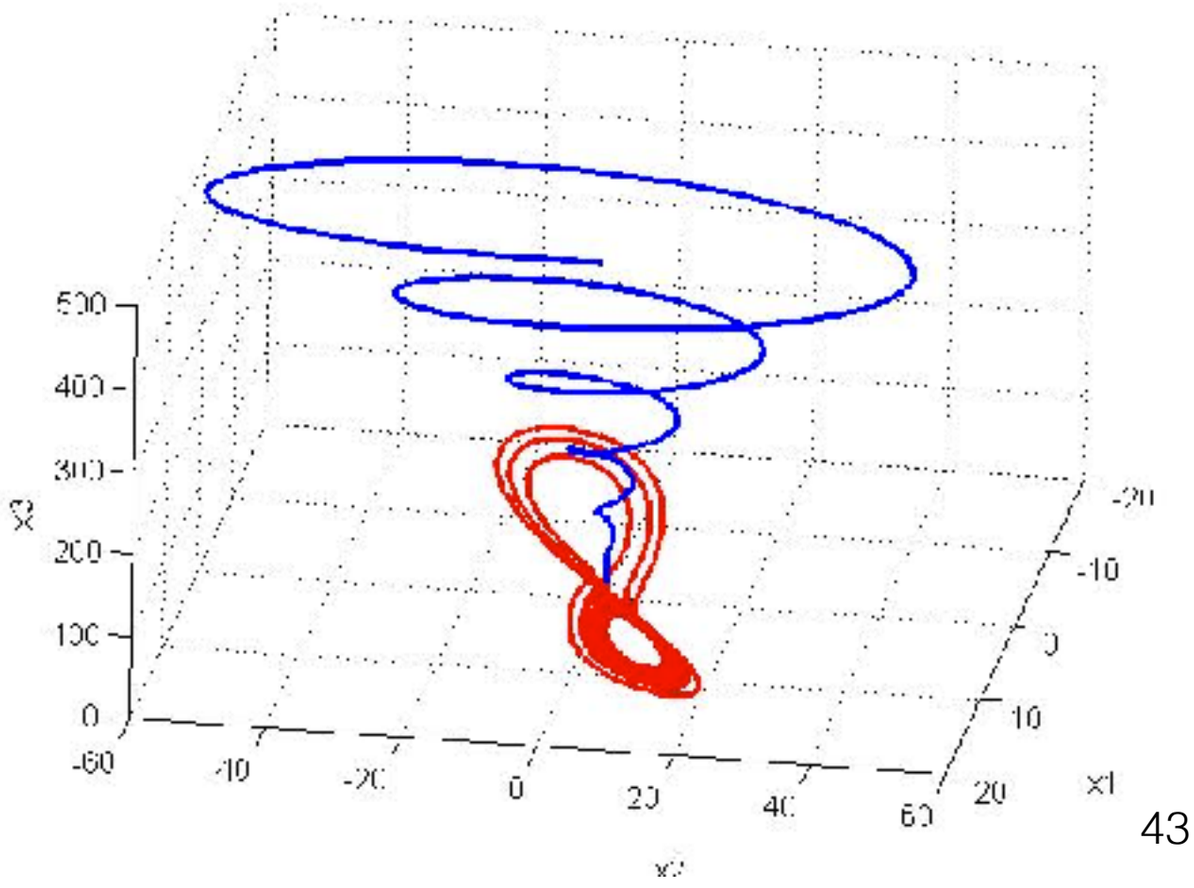
- hartritme
- lopen, zwemmen,…
- kloksystemen (dag-nachtritme)



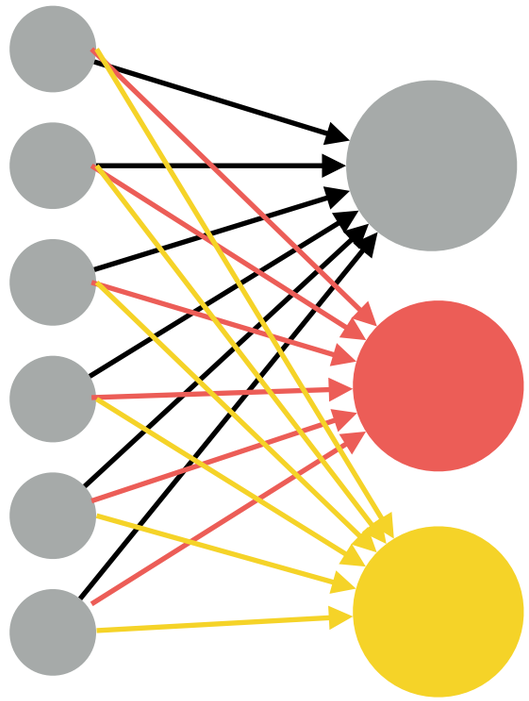
Eliasmith, 2007

Chaotic Attractor

- zoals cyclic attractor, maar dan steeds net ander paadje
- maar blijven wel ‘in de buurt’
- de ‘balanced networks’ van het volgende college zijn vaak ‘chaotic attractors'

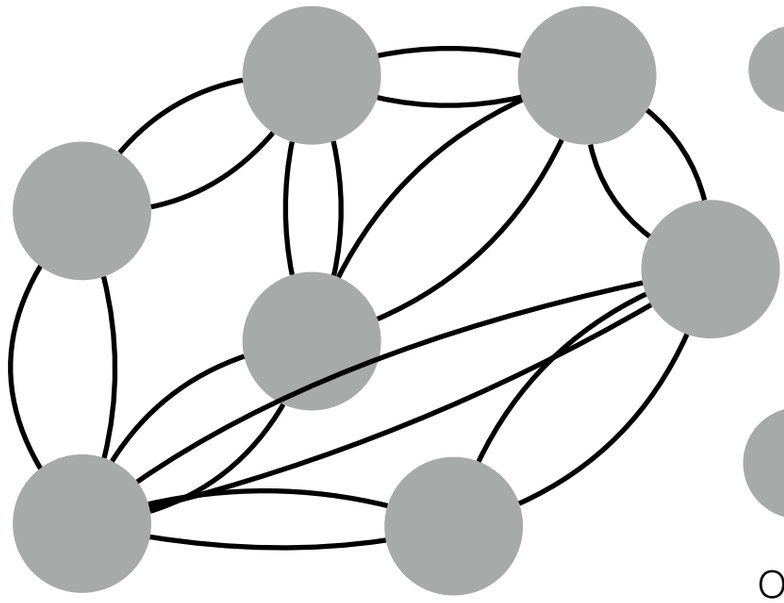


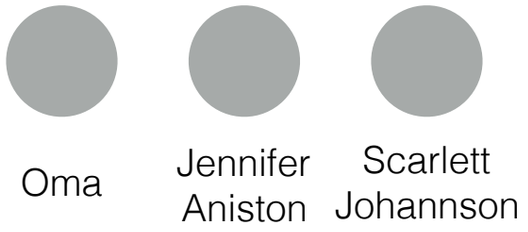
Samenvatting



Netwerken

- Feedforward netwerken
    - Perceptron
- Recurrente netwerken
    - Hopfield netwerk
    - Attractor netwerken



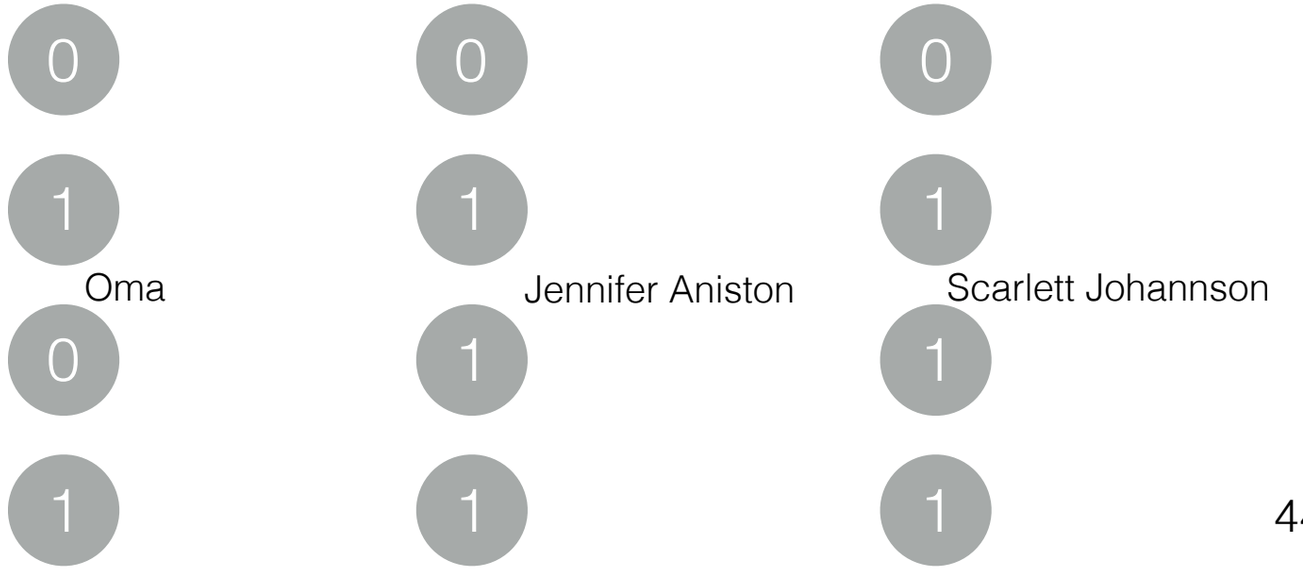


Neuronmodellen

- Binair neuron
- Rate neuron

Coding

- local
- distributed



Pauze

Hoe kun je aantonen dat de energie van een Hopfield netwerk altijd afneemt?

Hiervoor heb je een aantal zaken nodig:

- symmetrische verbindingen: $w_{ij} = w_{ij}$
- geen zelf-verbindingen $w_{kk} = 0$

Nu gaan we kijken wat er met de energie gebeurt als we één neuron updaten, neuron k

Als de activiteit van dat neuron niet verandert, verandert de energie natuurlijk ook niet.

Dus we kijken naar de twee andere mogelijkheden: veel input dus het neuron wordt actief (van 0 naar 1), of weinig input dus het neuron wordt inactief (van 1 naar 0).

Eerst gaan we de energiefunctie herschrijven.

$$E = -\frac{1}{2}\sum_{ij} w_{ij}x_i x_j + \sum_j T_j x_j$$

$$= \sum_j x_j(-\frac{1}{2}\sum_i w_{ij}x_i + T_j)$$

$$= \sum_j x_j(-\frac{1}{2}\sum_{i\neq k} w_{ij}x_i + T_j - \frac{1}{2}w_{kj}x_k)$$

$$= \sum_{j\neq k} x_j(-\frac{1}{2}\sum_{i\neq k} w_{ij}x_i + T_j - \frac{1}{2}w_{kj}x_k) + x_k(-\frac{1}{2}\sum_{i\neq k} w_{ik}x_i + T_k - \frac{1}{2}w_{kk}x_k)$$

$$= \sum_{j\neq k} x_j(-\frac{1}{2}\sum_{i\neq k} w_{ij}x_i + T_j) - \frac{1}{2}\sum_{j\neq k} x_j w_{kj}x_k + x_k(-\frac{1}{2}\sum_{i\neq k} w_{ik}x_i + T_k) + 0$$

Using $w_{kk} = 0$ and $w_{ij} = w_{ji}$

$$= \sum_{j\neq k} x_j(-\frac{1}{2}\sum_{i\neq k} w_{ij}x_i + T_j) - \sum_{i\neq k} w_{ik}x_i x_k + T_k x_k$$

Using again $w_{kk} = 0$

$$= (\text{terms independent of neuron k}) - x_k(\sum_i w_{ik}x_i - T_k)$$

Kijk nu naar het energie**verschil** als neuron k geüpdatet wordt van x_k naar x^*_k

$$\begin{aligned}
\Delta E =& E(x_k^*) - E(x_k) \\
=& (\text{terms independent of neuron k}) - x_k^*(\sum_i w_{ik}x_i - T_k) \\
& - (\text{terms independent of neuron k}) + x_k(\sum_i w_{ik}x_i - T_k) \\
=& (x_k - x_k^*)(\sum_i w_{ik}x_i - T_k) \\
=& (x_k - x_k^*)(\text{total input})
\end{aligned}$$

State change 0→1:

- (x_k - x^*_k) = (0-1) = -1
- total input positief → Energieverschil negatief

State change 1→0:

- (x_k - x^*_k) = (1-0) = 1
- total input negatief → Energieverschil negatief