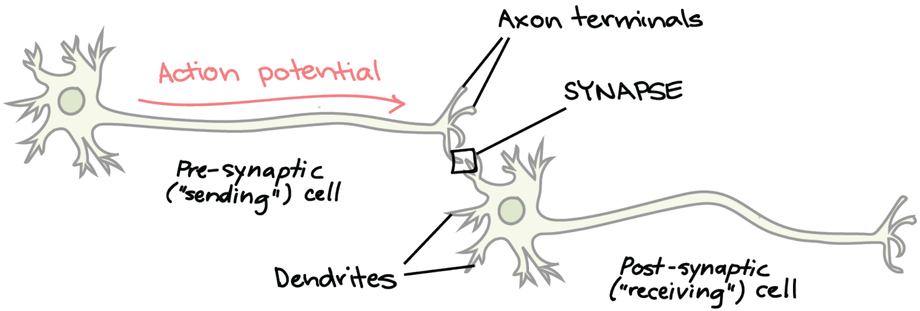
Hopfield networks

Nagy Dániel

Motivation and history

- Your heart is a relatively simple organ: you can easily descrbe *what* it does it's a pumping blood.
- What about Your brain? We can't even describe *what* it
- What is seeing something?
- What is thinking something?
- *How* do you decide whether you see a cat or a dog?
- Can we create a machine that can do similar tasks to what our brain does? (e.g. telling what can be seen on a picture)
- The first steps of creating artificial neural networks dates back to the early 40s (McCulloch and Pitts)[8]
- Hebbian learning was introduced in 1949 [6]
- First perceptron model by Rosenblatt in 1958 [10]
- The Hopfield network was first described in 1974 by [11]

Structure of a biological neuron



- Short story: information is carried down the axon in the form of spike-like electric pulses, and transmitted through the synapse as chemical signals.
- inside of the cell the potential is typically -70mV
- due to the influence of the chemical signal form another neuron this potential increases a bit
- If the frequency of the input signals is sufficiently large, the potential will accumulate and will reach the threshold
- If it reaches a threshold level of about -55mV, the potential flips to positive, then goes back to the initial level of -70mV

What happens at the synapse?

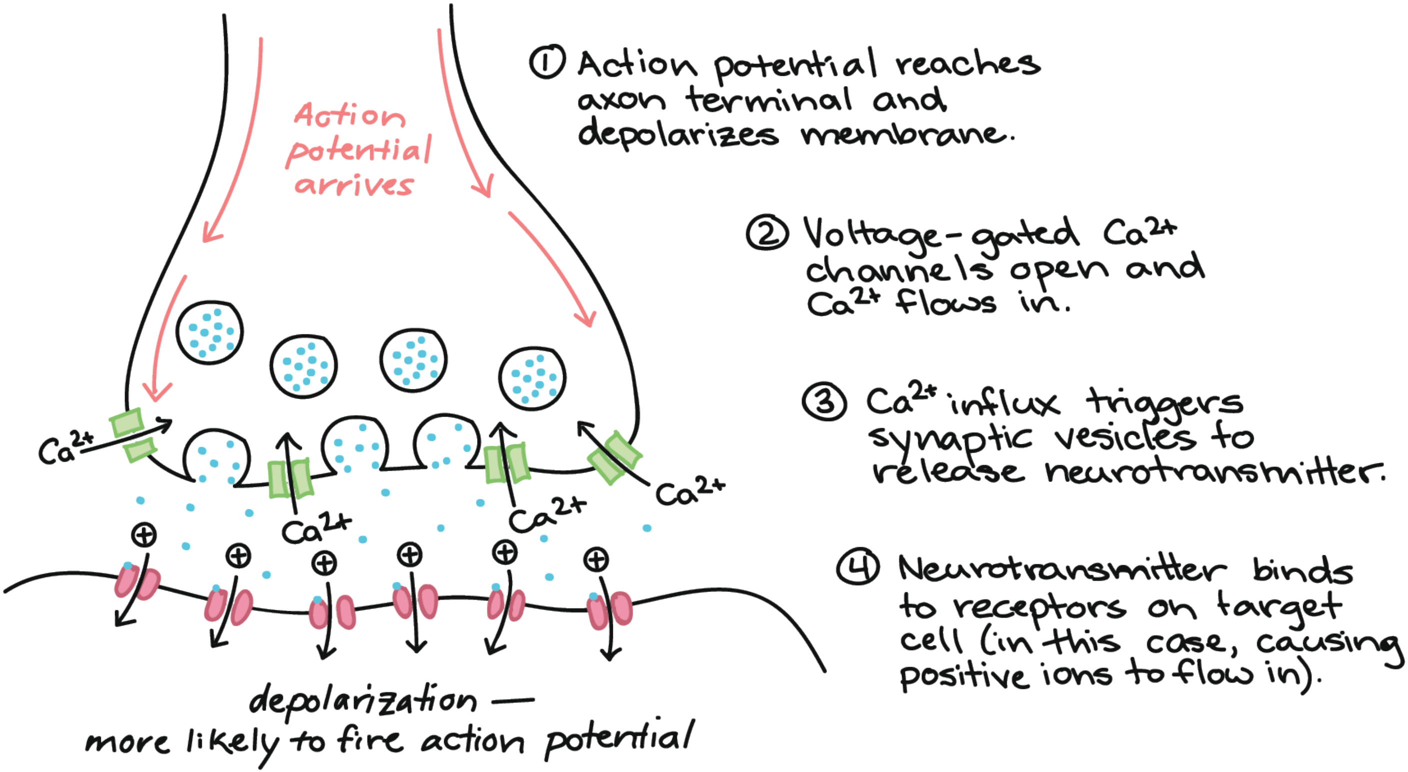
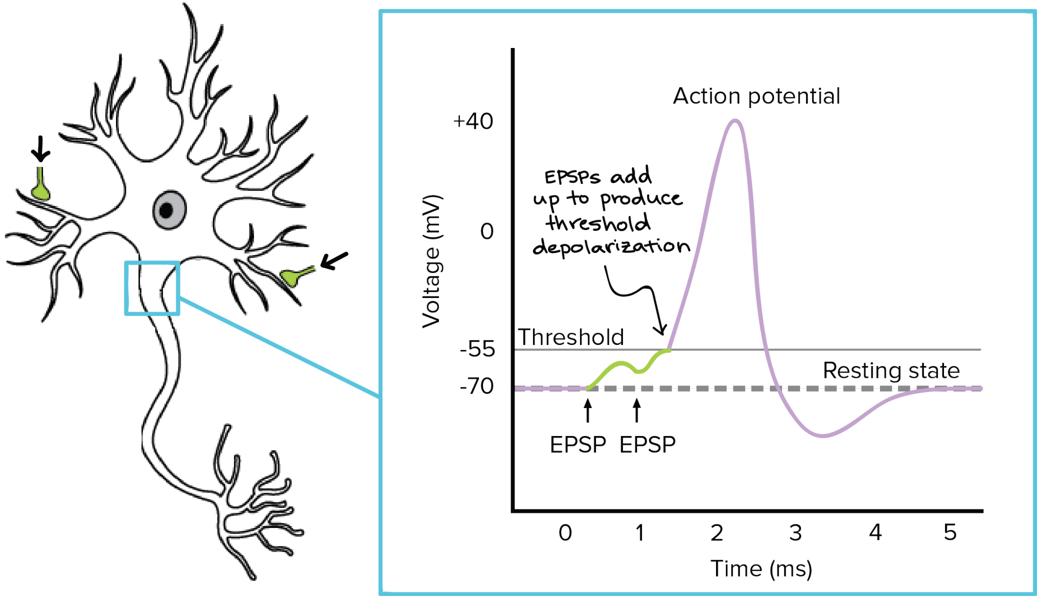


Figure 1: Source: Khan Academy [1]



- The output frequency y_i of the i-th neuron depends of the input frequencies x_j:

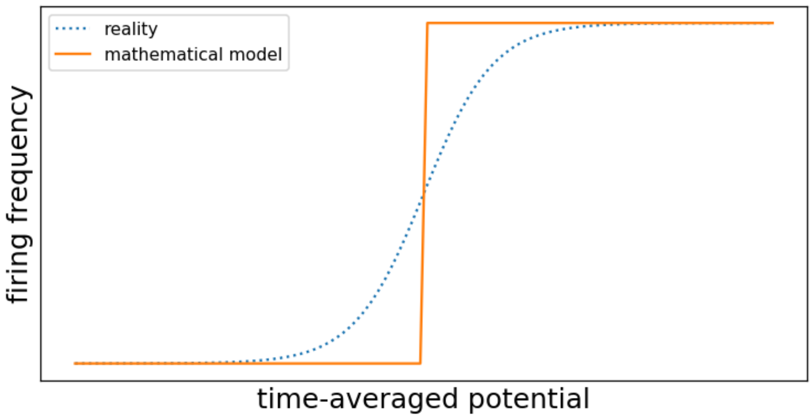
$$y_i = f\left(\sum_j J_{ij} x_j - U\right),$$

where U is the threshold potential, J_{ij} are the weights descrbing the synaptic strength and $f(x)$ is a filter function.

- The filter function is usually a sigmoid, but other nonlinear activations are frequently used in machine learning.
- In the Hopfield-model the filter is approximated with the Heaviside step function:

$$f_\alpha(x) = \frac{1}{1+e^{-\alpha x}}$$
$$f_\infty(x) = \Theta(x) = \begin{cases} 1, & \text{if} x > 0, \\ 0, & \text{if} x \leq 0 \end{cases}$$

- Thus a neuron has a binary state: either firing or not.



Connection with the Ising-model

- $V_i(t) =$ the state of the i-th neuron a t t
- State update rule:

$$V_i(t+1) = f\left(\sum_j J_{ij} V_j(t) - U\right)$$

- Analogy with physics: neurons which are firing correspond to up spins while neurons which are quiescent correspond to down spins.
- Introducing $S_i = 2V_i - 1 = \pm 1$, we get an Ising-model:

$$f\left(\sum_j J_{ij} V_j(t) - U\right) = f\left(\frac{1}{2}\sum_j J_{ij} S_j(t) + \left(\frac{1}{2}\sum_j J_{ij} - U\right)\right)$$

- Writing J_{ij} instead $\frac{1}{2}J_{ij}$, this can be interpreted as

$$S_i(t+1) = +1, \text{ with probability } f(h_i(t)), \text{ where}$$

$$h_i(t) = \sum_j J_{ij} S_j(t) + \underbrace{\left(\sum_j J_{ij} - U\right)}_{h_i^{ext}}$$

Connection with the Ising-model

- The hoise level can be characterised by the inverse temperature, β
- In a magnetic field h_i, spin i has energy $\varepsilon_i = -h_i S_i$
- The probability if the i-th spin to have value S_i is $P(S_i) \propto e^{-\beta\varepsilon_i} = e^{+\beta h_i S_i}$

$$\implies P\left(S_i(t+1) = +1 | h_i(t)\right) = f(h_i(t)) = \frac{e^{\beta h_i(t)}}{e^{\beta h_i(t)} + e^{-\beta h_i(t)}} = \frac{1}{1 + e^{-2\beta h_i(t)}}.$$

- Therefore,

$$S_i(t+1) = \begin{cases} +1, \text{ with probability } \frac{1}{1+e^{-2\beta h_i(t)}} \\ -1, \text{ with probability } \frac{1}{1+e^{+2\beta h_i(t)}} \end{cases}$$

- $S_i(t+1)$ tends to be parallel with $h_i(t)$
- The difference between probabilities vanishes as the noise $\beta \to 0$.

Hebbian learning

- We need a model for *learning* and *memory*, since these are essential functions of the brain.
- Information can be encoded in a firing pattern $\{\xi\}$, $\xi_i = \pm 1 \, \text{for} \, (i = 1, \ldots, N)$.
- A firing pattern is stable, if the neural network comes back to this pattern after a disturbance:

$$S_i(t) = \xi_i \rightarrow S_i(t+1) = \xi_i$$

- The stable patterns are attractors of the dynamics of the network.
- Learning a pattern is achieved by setting the synaptic strengths J_{ij}.
- Hebb's rule for updating synaptic weights:

$$J_{ij} \rightarrow J_{ij} + \lambda \xi_i \xi_j,$$

  where λ is the amplitude of learning (learning rate).
- This rule is extremely simple yet it proved to be very powerful.
- What if we want to store p different patterns in the same network?
- We need to set J_{ij} so that ξ_i^μ are attractors of the network dynamics for $\mu \in \{1, ..., p\}$.
- The **Hopfield-network** is capable of learning p different patterns using Hebb's learning Rule.

The Hopfield network

- Every node in the network has two states $S_i = \pm 1$.
- Every node is connected to every node, forming a complete undirected weighted graph.
- The connection weights are J_{ij} with the restrictions $J_{ii} = 0$, $J_{ij} = J_{ji}$.
- The dynamics of the network is described by

$$S_i(t+1) = \begin{cases} +1, \text{ if } \sum_j J_{ij} S_i(t) \geq U_i, \\ -1, \text{ otherwise} \end{cases}$$

- The noise factor β is assumed to be 0.
- It works as a content-addressable memory a.k.a. associative memory.
- Interestingly, it can reconstruct data after being fed with corrupt versions of the same data.
- Can be trained using the Hebbian training rule.

John Joseph Hopfield (1933-)

Hebbian learning in Hopfield-networks

- Motto: *"Neurons that fire together, wire together. Neurons that fire out of sync, fail to link."*
- Suppose, we have p different patterns $\xi_1^\mu, \dots, \xi_N^\mu \in \{\pm 1\}$, for $\mu \in \{1, \dots, p\}$
- Then we can set the weights (initially $J_{ij} = 0$) with the Hebbian rule:

$$J_{ij} = \lambda \sum_{\mu=1}^{p} \xi_i^\mu \xi_j^\mu \text{ for } i \neq j \text{ and } J_{ii} = 0,$$

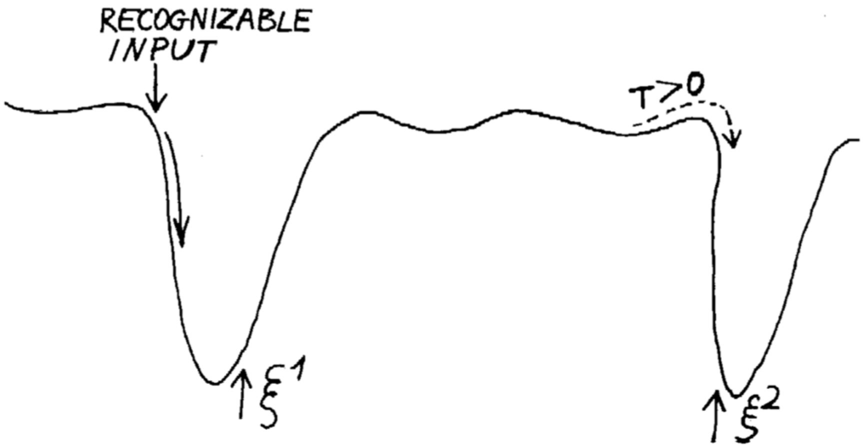
  where λ is usually set to $1/p$.

- We can define an energy-like scalar value for each state of the network:

$$E = -\frac{1}{2} \sum_{i,j} J_{ij} S_i S_j + \sum_i U_i S_i$$

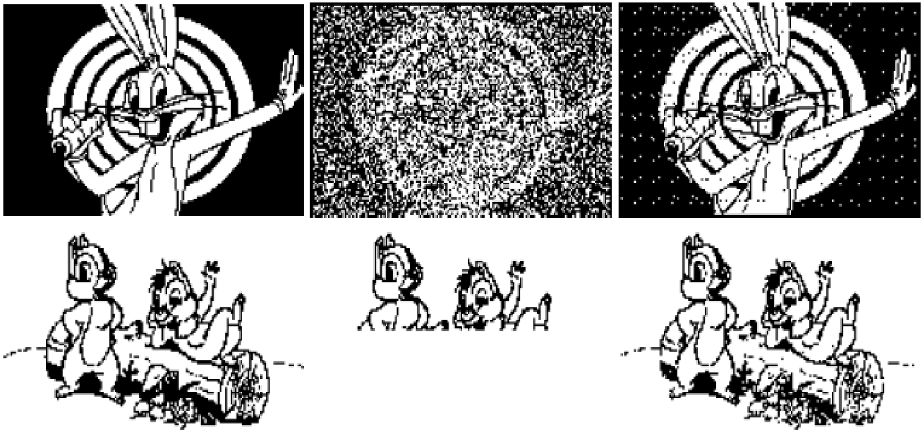
- Under repeated updating the network will eventually converge to a state which is a local minimum in the energy function.
- If a state corresponds to a local minimum of the energy function, it is a stable state of the network.
- The states $\{\xi_i^\mu\}$ are attractors of the dynamics thus they are stable states of the network and their corresponding energy level is a local minimum.

Data reconstruction



- If we give a corrupted input, the systems dynamics will drive the network to the nearest local minima, which is hopefully the reconstructed data.

Data reconstruction



Original Corrupted Reconstructed

- If we give a corrupted input, the systems dynamics will drive the network to the nearest local minima, which is hopefully the reconstructed data.

Storkey learning rule

- Another useful rule for learning is the *Storkey-learning rule*:

$$J_{ij} \to J_{ij} + \frac{1}{p}\xi_i^\mu \xi_j^\mu - \frac{1}{p}\xi_i^\mu h_j^\mu - \frac{1}{p}\xi_j^\mu h_i^\mu,$$

  where

$$h_i^\mu = \sum_j J_{ij} S_i^\mu.$$

- Hopfield-networks trained with the Hebbian rule have capacity ≈ 0.138 i.e. they can store 138 different patterns per 1000 nodes.

- Networks trained with the Storkey rule have capacity ≥ 0.14 [7].

- There exist higher-order Storkey learning rules and many other tricks for increasing the capacity of Hopfield-networks [4].

References

[1] Khan Academy: The neoron and nervous system https://www.khanacademy.org/science/biology/human-biology#neuron-nervous-system.

[2] Hopfield Networks are useless. Here's why you should learn them. https://towardsdatascience.com/hopfield-networks-are-useless-heres-why-you-should-learn-them-f0930ebeadcd.

[3] Hopfield-networks Wikipedia page https://en.wikipedia.org/wiki/Hopfield_network.

[4] Ala ABOUDIB, Vincent Gripon, and Xiaoran JIANG. A study of retrieval algorithms of sparse messages in networks of neural cliques. In COGNITIVE 2014 : the 6th International Conference on Advanced Cognitive Technologies and Applications, pages 140–146, Venise, Italy, May 2014.

[5] Tamás Geszti. Physical Models of Neural Networks. World Scientific Pub Co Inc, 1990.

[6] D.O. Hebb. The Organization of Behavior: A Neuropsychological Theory. Taylor & Francis, 2005.

[7] Cheng-Yuan Liou and Shao-Kuo Yuan. Error tolerant associative memory. Biological Cybernetics, 81(4):331–342, Oct 1999.

[8] W. McCulloch, W.S. & Pitts. Bulletin of Mathematical Biophysics (1943) 5: 115.

[9] Patrick Rebentrost, Thomas R. Bromley, Christian Weedbrook, and Seth Lloyd. Quantum Hopfield neural network. 98(4):042308, Oct 2018.

[10] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. Psychological Review, pages 65–386, 1958.

[11] Saratha Sathasivam and Wan Ahmad Tajuddin Wan Abdullah. Logic Learning in Hopfield Networks. arXiv e-prints, page arXiv:0804.4075, Apr 2008.

Thank You for listening!