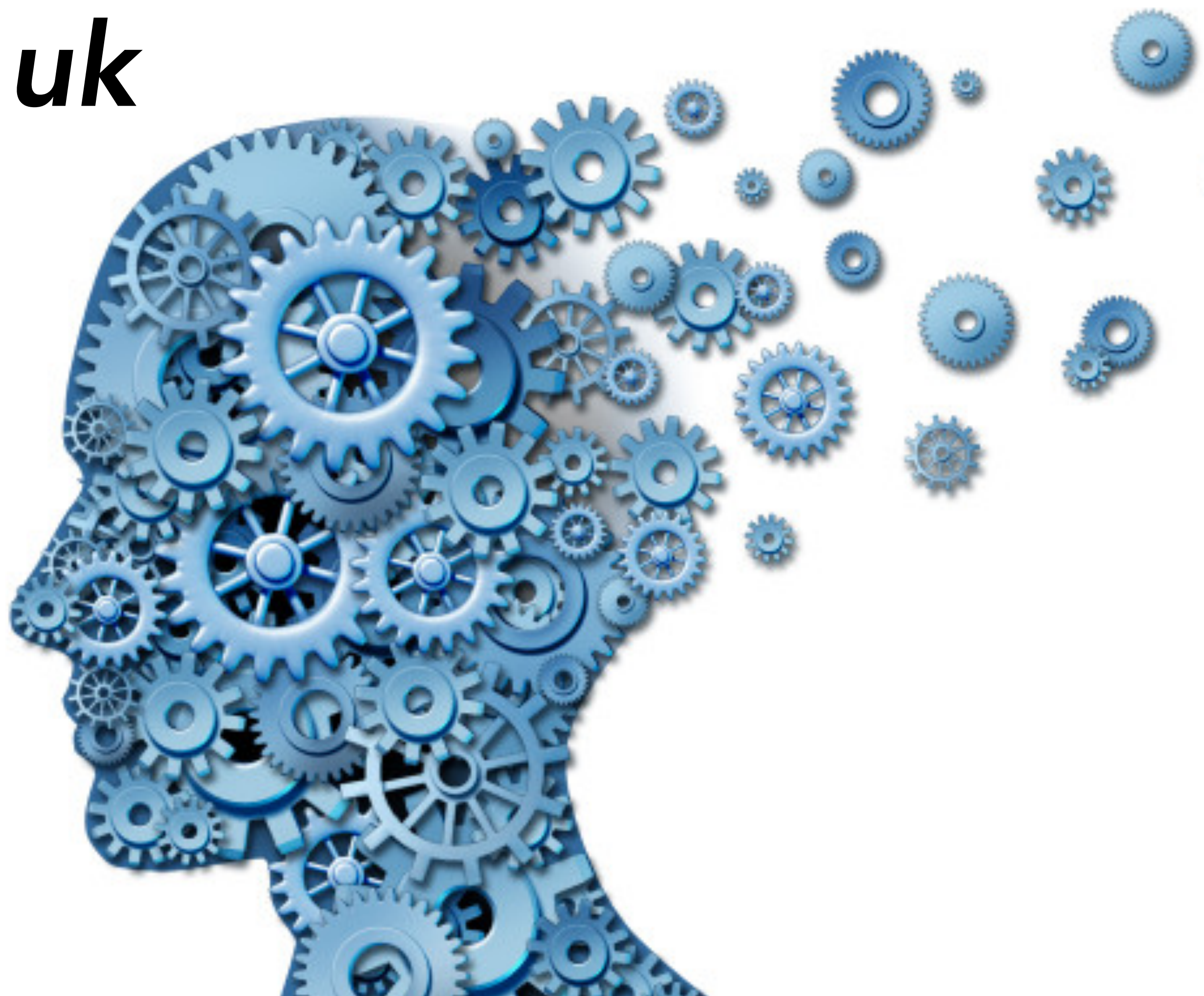
COMS30127/COMSM2127 Computational Neuroscience

Lecture 7: McCulloch-Pitts neurons, Perceptrons, and Hopfield networks (d)

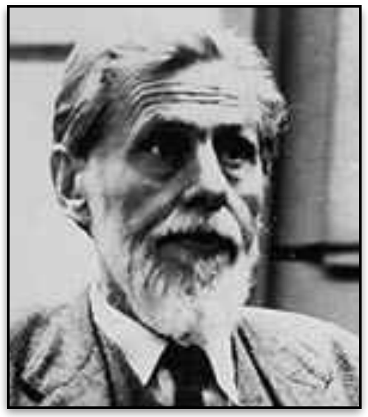
Dr. Cian O'Donnell
cian.odonnell@bristol.ac.uk

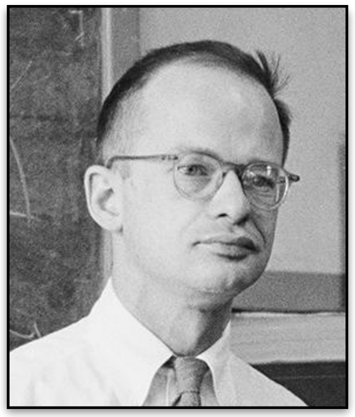


What we will cover today

- The McCulloch-Pitts neuron model.
- Perceptrons.
- Hopfield networks.
- Hebbian plasticity (brief intro).

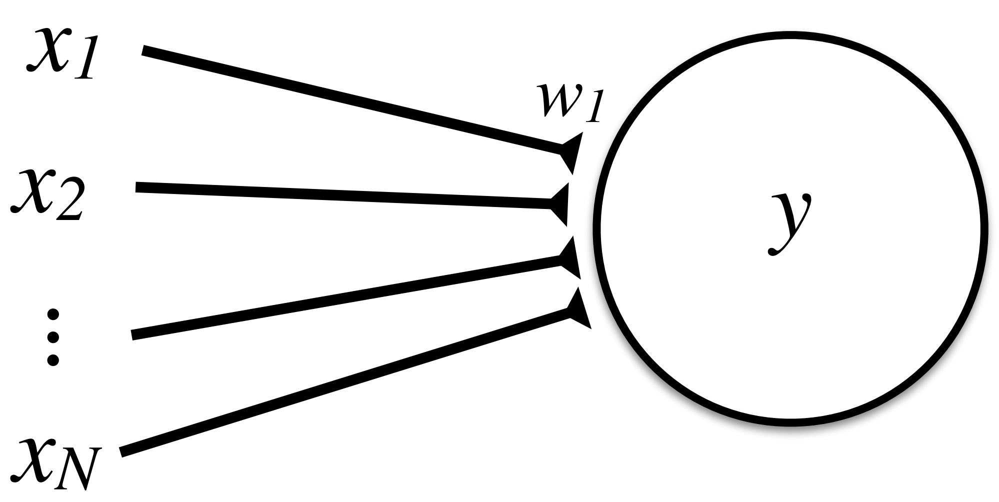
McCulloch-Pitts neuron


Warren McCulloch


Walter Pitts

- McCulloch and Pitts were some of the earliest computational neuroscientists.
- They proposed an extremely simple model neuron in 1943 ("A Logical Calculus of Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics* 5:115–133.)
- For them it was the building block for a hopeful theory of brain logic, akin to digital logic we use in computers. Although this grand idea never sustained popularity, their neuron model is still widely used today.
- Each MP neuron performs a weighted linear sum of its inputs, then thresholds to give a binary output.
- Time is usually assumed to evolve in discrete steps, and the neuron has no internal dynamics or memory.

McCulloch-Pitts neuron



$$y = g\left(\sum_{i}^{N} w_i x_i - \theta\right)$$

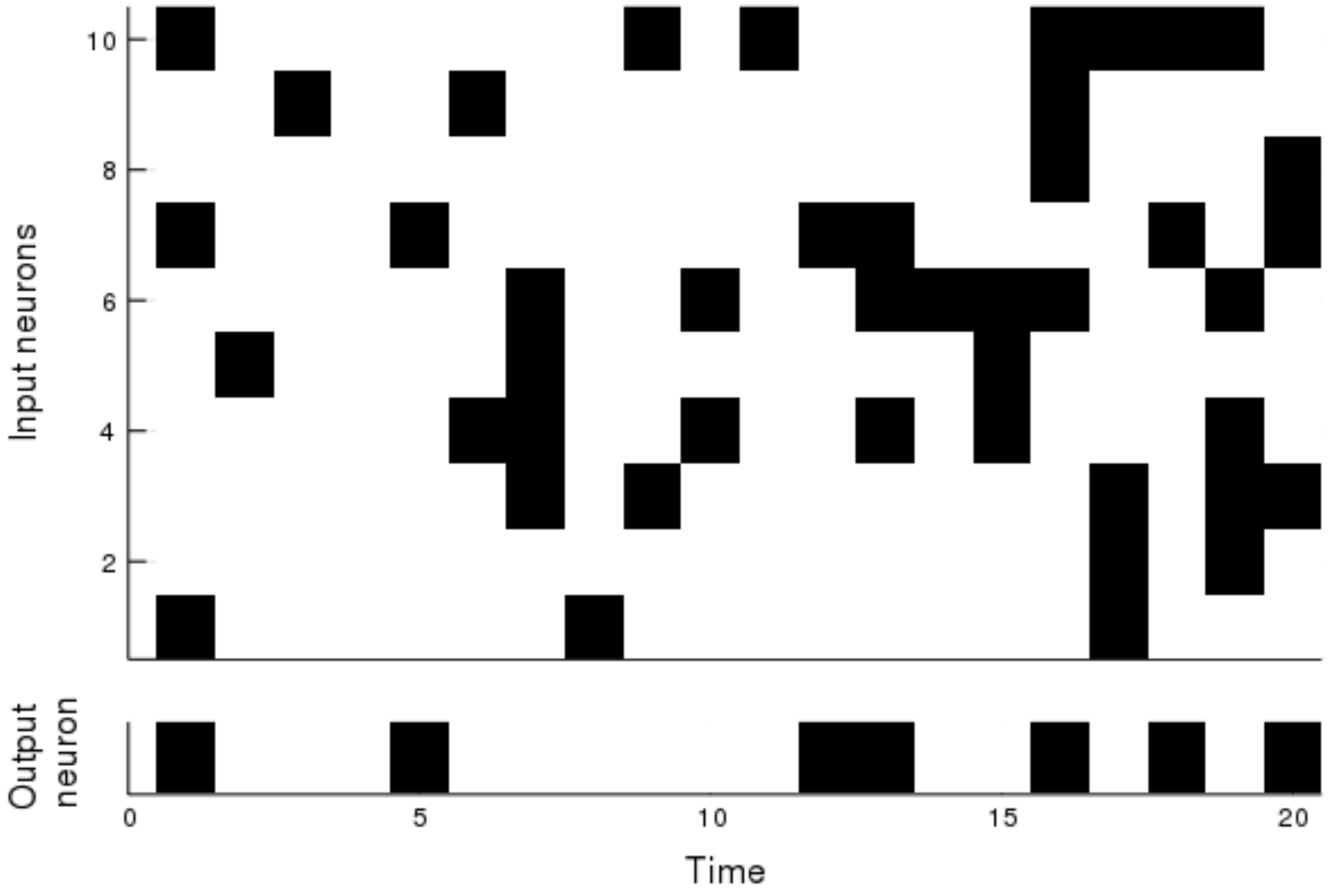
x_i is the i^{th} input neuron's activity

w_i is the strength of the synaptic weight from neuron i

θ is the neuron's threshold

g is a step function where $g(z) = +1$ if $z>0$, otherwise $g(z)=-1$ if $z<0$

McCulloch-Pitts neuron example simulation



McCulloch-Pitts neurons with synaptic plasticity:

1. Perceptrons (supervised)
2. Hopfield networks (unsupervised)

Perceptrons

Frank Rosenblatt

- Perceptrons are a useful application of McCulloch-Pitts neurons.
- First described by Frank Rosenblatt (1957).
- They add a supervised learning rule to adjust the neurons synaptic weights, to solve a classification task.
- Perceptrons were the precursors to artificial neural networks, today known as *deep learning*.
- Also a decent model for learning in the cerebellum.

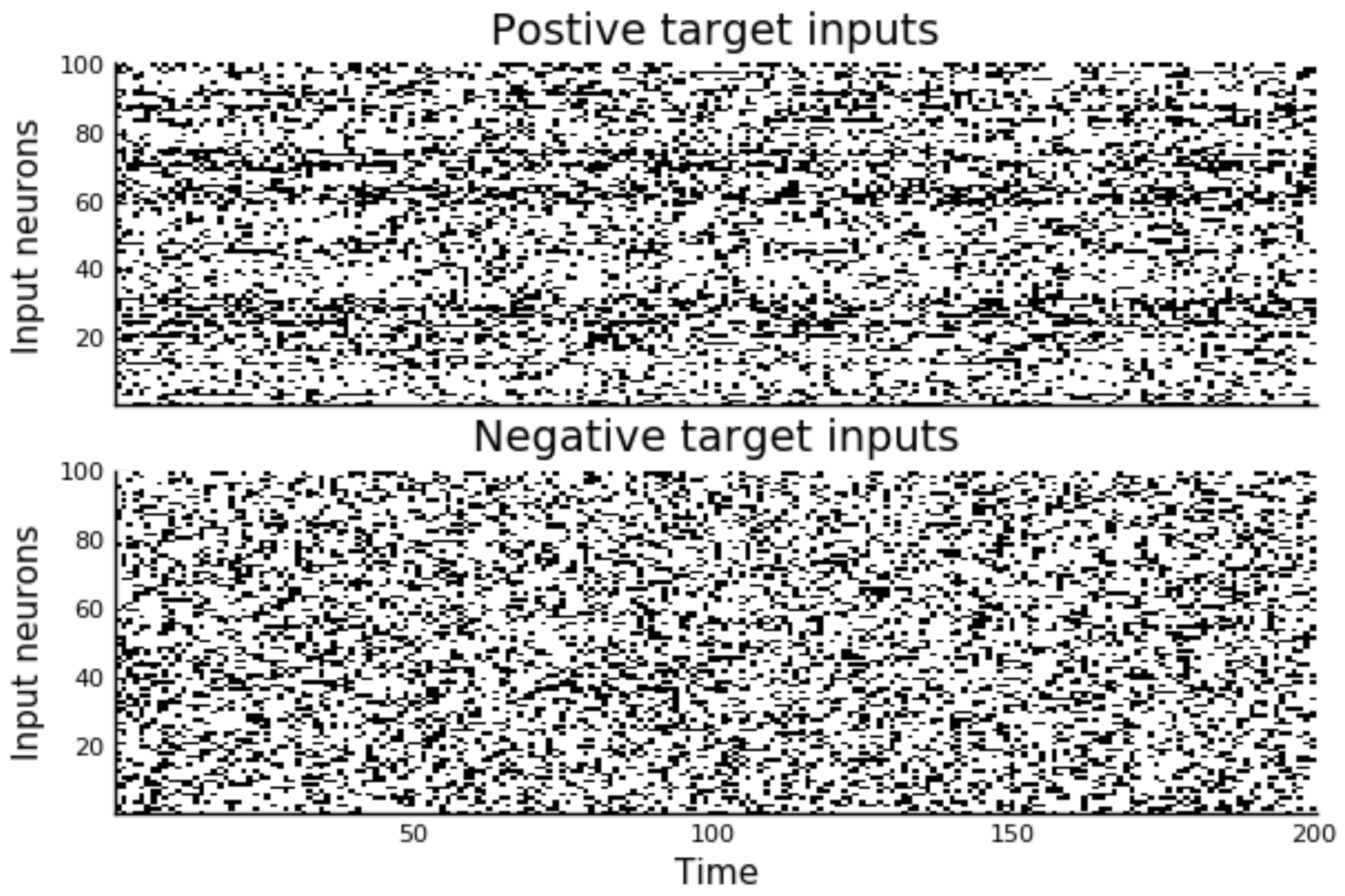
Perceptron learning rule

- Imagine each possible input pattern $\mathbf{x} = [x_1, x_2, \ldots, x_N]$ is associated with some desired binary target value $d \in \{-1,1\}$
- If we initialise the weights at random then initially there will be some error $d - y \neq 0$
- Then we train the model by presenting each training pattern in turn, and update the weights after each pattern according to:

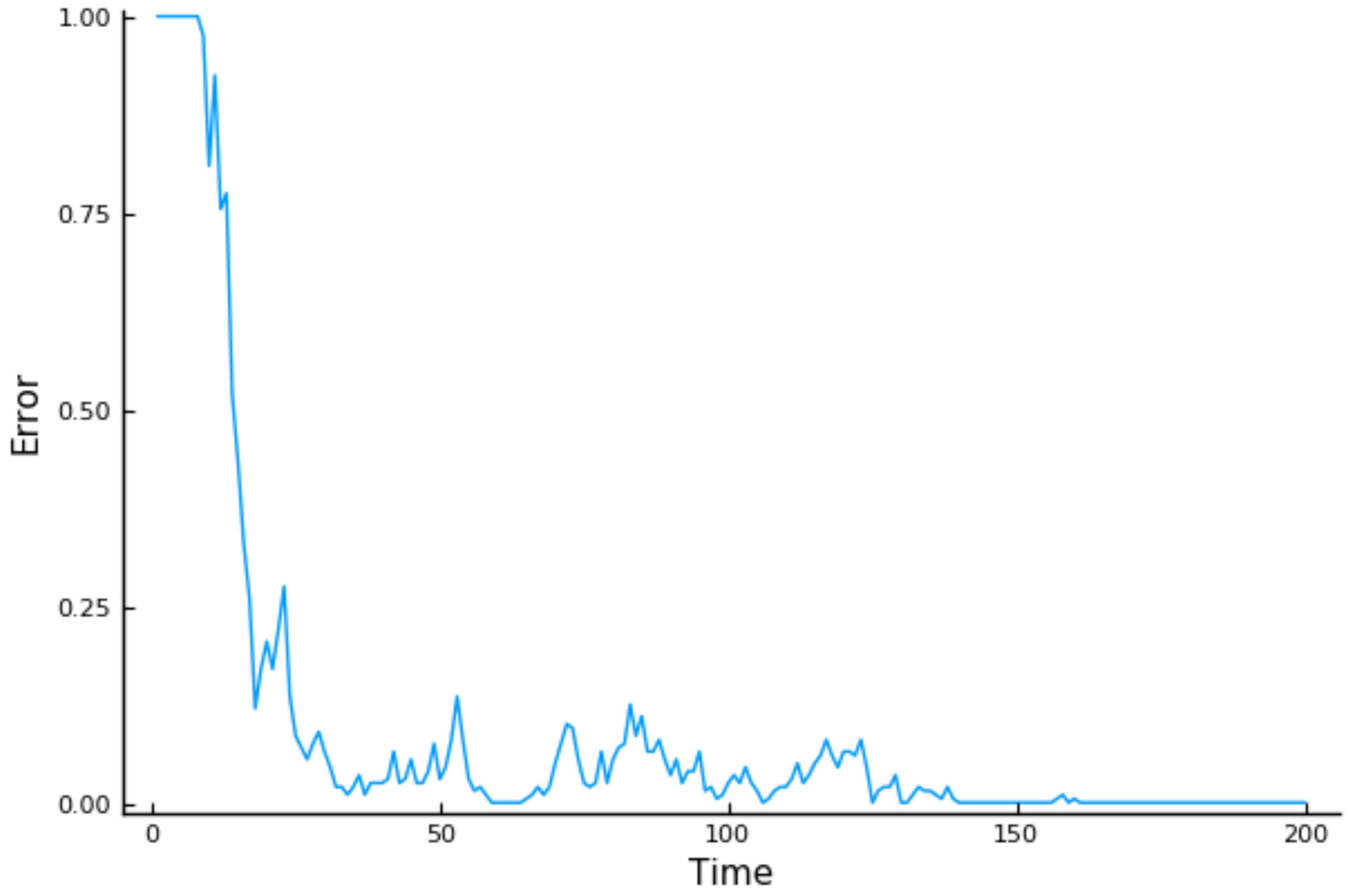
$$\Delta w_i = \eta(d - y)x_i$$

- If we repeat this procedure enough times, then the neuron will eventually learn to solve the task (as long as the two input classes as linearly separable).

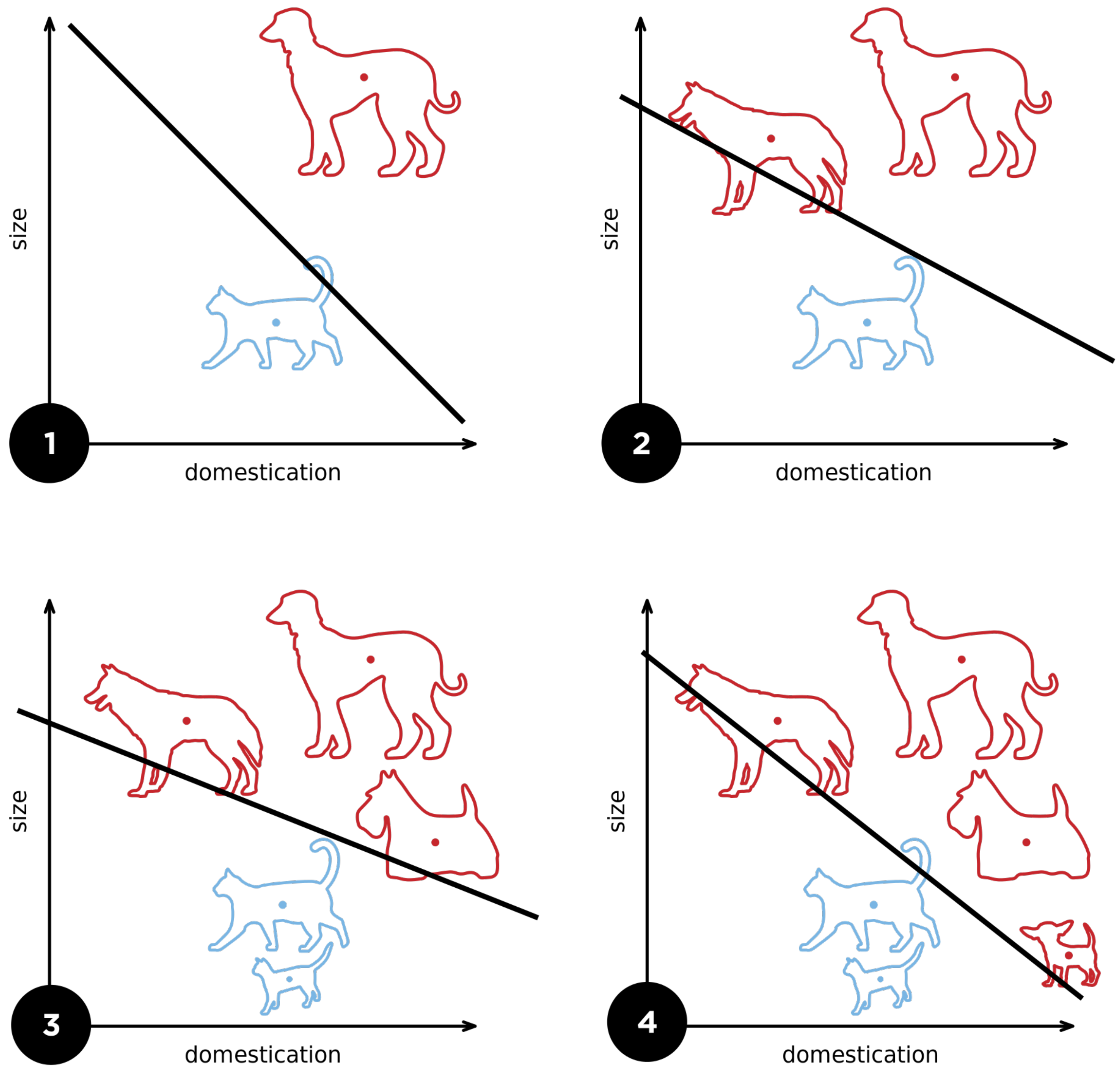
Perceptron training example



Perceptron training example



Perceptron as a linear classifier

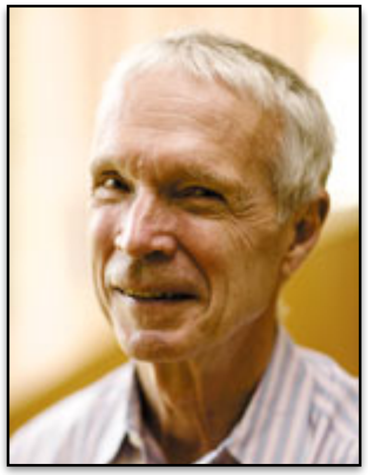


https://en.wikipedia.org/wiki/Perceptron#/media/File:Perceptron_example.svg

The A.I. winters

- When perceptrons were first discovered there was a lot of hype: the New York Times reported that they were "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence".
- Minsky and Papert's 1969 book "Perceptrons" took part in killing the hype by demonstrating, among other things, that perceptrons could not solve the `XOR` task.
- Research funding and interest in neural networks remained low until the mid 1980s when the field (re)discovered the back-propagation algorithm for training multi-layer perceptrons.
- This lead to a new hype bubble which lasted until the early 1990s when it became clear that neural networks, although theoretically very powerful, required an impractical amount of data to be useful.
- The field of machine learning grew and moved on without neural networks, until their dramatic re-entry in the late 2000s. This latest return was due to three main factors:
  - the ubiquity of computing power (notably GPUs)
  - the availability of large-scale datasets
  - Algorithmic innovations, such as stacking many network layers (hence the name "deep").
- Now deep learning is everywhere!

Hopfield networks


John Hopfield

- A Hopfield network is a recurrent network of MP neurons.
- Proposed by John Hopfield in 1982.
- The network state evolves dynamically, typically toward some "attractor" state.
- A simple synaptic plasticity rule can imprint attractors in the network weights.
- Forms a basic model of associative memory recall.
- Incredibly influential model in the history of computational neuroscience (attracted a generation of physicists to the field).

Hopfield networks

- Network state evolves as: $x_i(t+1) = g\left(\sum_{j \neq i}^{N} [w_j x_j(t) - \theta]\right)$
- There are two common flavours: synchronous or asynchronous updates.
- Usually the weights are symmetric: $w_{ij} = w_{ji}$ and the connectivity is all-to-all.
- The network dynamics evolve to minimise an "energy":

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

Hopfield network dynamics

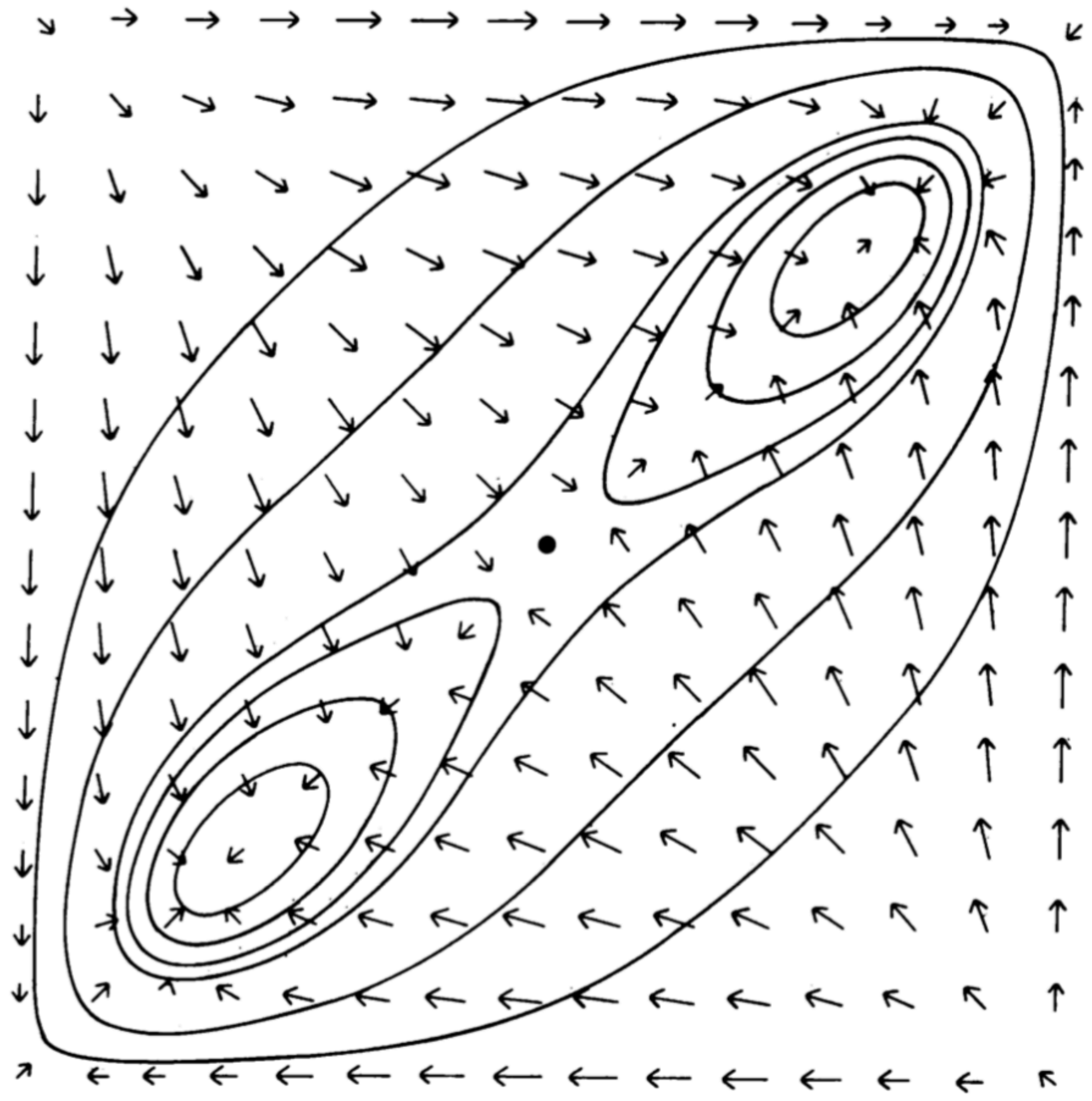


FIG. 3. An energy contour map for a two-neuron, two-stable-state system. The ordinate and abscissa are the outputs of the two neurons. Stable states are located near the lower left and upper right corners, and unstable extrema at the other two corners. The arrows show the motion of the state from Eq. 5. This motion is not in general perpendicular to the energy contours. The system parameters are $T_{12} = T_{21} = 1$, $\lambda = 1.4$, and $g(u) = (2/\pi)\tan^{-1}(\pi\lambda u/2)$. Energy contours are 0.449, 0.156, 0.017, −0.003, −0.023, and −0.041.

Hopfield, *PNAS* 1984

- Each of local minima in the energy landscape is known as an "attractor".
- The network can do pattern completion: retrieving the full pattern from a partial cue.
- The capacity of the network, or maximum number of attractors, is $\sim 0.14N$.

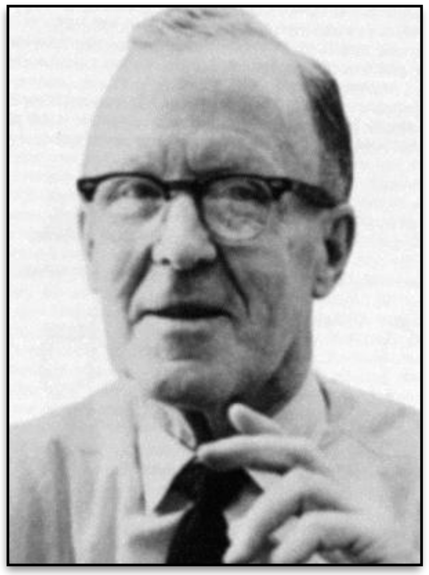
Learning attractors in Hopfield networks

We can imprint desired attractor states into the synaptic weights by using a learning rule:

$$w_{ij} = \frac{1}{P} \sum_a x_i^a x_j^a$$

P is the total number of attractors to store.
a indexes the attractor activity åpattern in the sum.

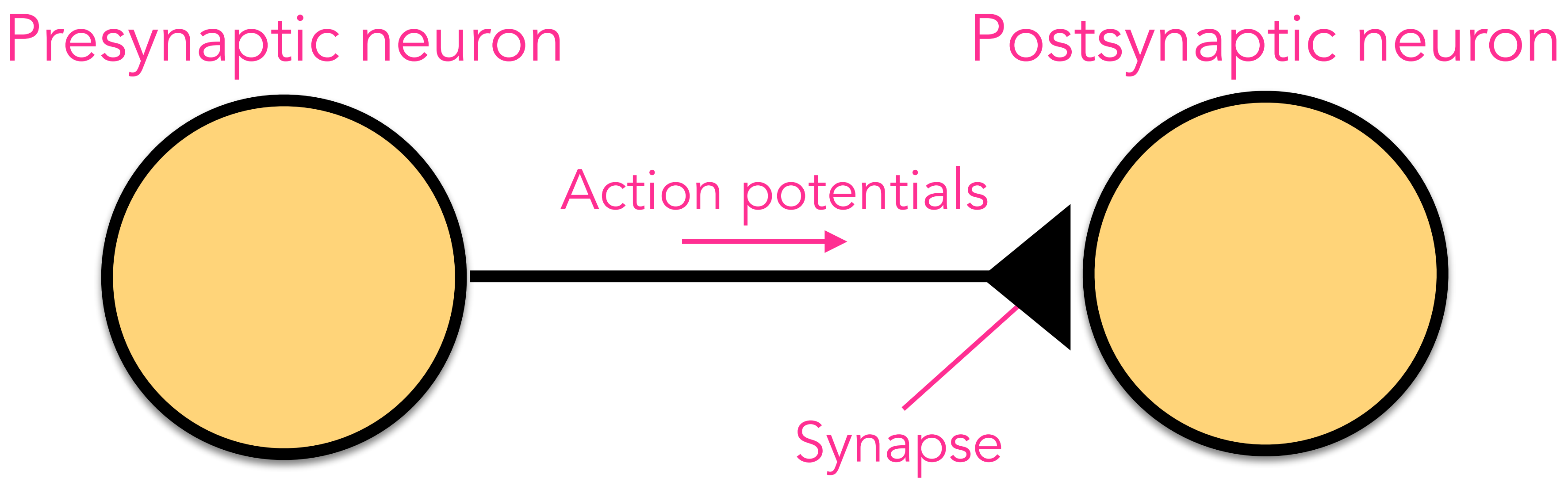
Hebbian plasticity


Donald Hebb

"When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased."
— Donald Hebb (1949)

a.k.a. "neurons that fire together wire together."

Hebbian plasticity



- Many synapses in the brain strengthen if their pre- and post-synaptic neurons are simultaneously active for a sufficient period of time.
- This strengthening process is called “long-term potentiation”.
- There are other stimulation patterns that can weaken synapses (the corresponding process is called “long-term depression”).
- It is a candidate brain mechanism for associative learning.
- It is also embodied in the Hopfield network learning rule.

References

- McCulloch, W.S., Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* 5, 115–133.

- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.

- Minsky, M., & Papert, S. (1969). Perceptron: an introduction to computational geometry. The MIT Press, Cambridge.

- Hopfield, J.J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proc. Natl. Acad. Sci. U.S.A.* 79, 2554–2558.

- Hebb, D.O. (1949) The Organization of Behavior. Wiley, New York.