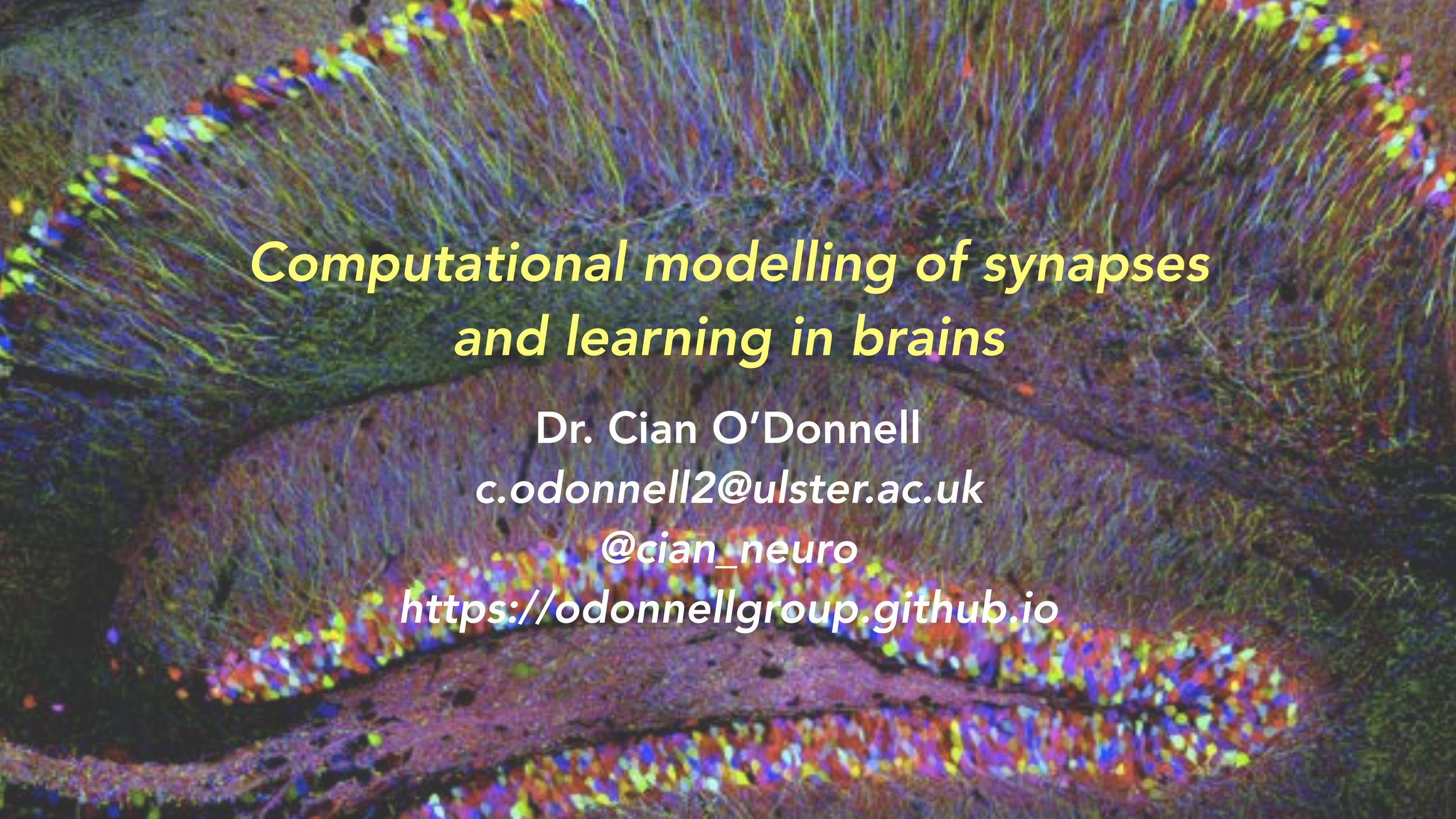


Computational modelling of synapses and learning in brains

Dr. Cian O'Donnell

c.odonnell2@ulster.ac.uk

@cian_neuro

https://odonnellgroup.github.io

Why study learning?

KordingLab 👨‍🦲🧠∇🔬📈,🏋️🏄🏂🛹🕺🌋☕🦖

@KordingLab

In computational neuroscience can we realistically understand computation (say how we see a hand) or can we at best hope to understand the learning that gives rise to the computation?

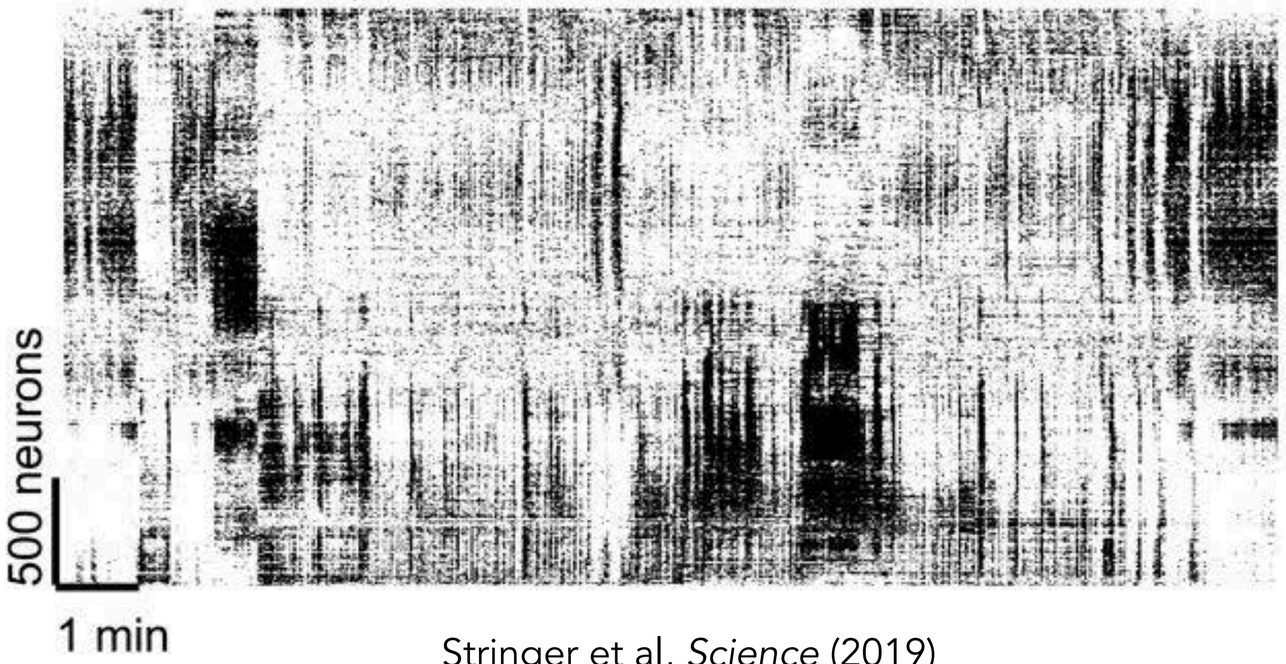
4:30 PM · Oct 14, 2021 · Twitter Web App

15 Retweets **5** Quote Tweets **130** Likes

What does it mean to understand a neural network?
Lillicrap & Kording, arXiv (2019)

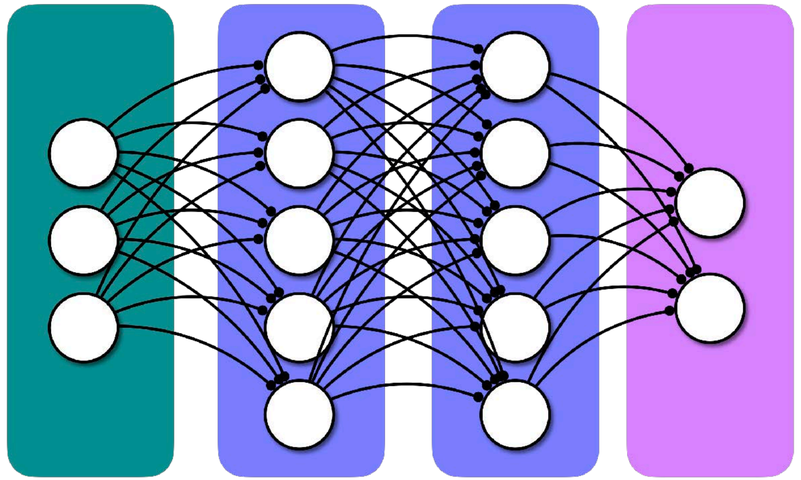
Why study learning?

Population activity in the brain



Stringer et al, *Science* (2019)

Deep neural network



What we will cover

1. Synapses

- What is a synapse and how does it work?
- Basic computational models of synaptic transmission

2. Synaptic plasticity

- Classic rate-based models of plasticity, Hebbian learning.
- Spike-timing dependent plasticity.

3. Associative memories and attractor networks

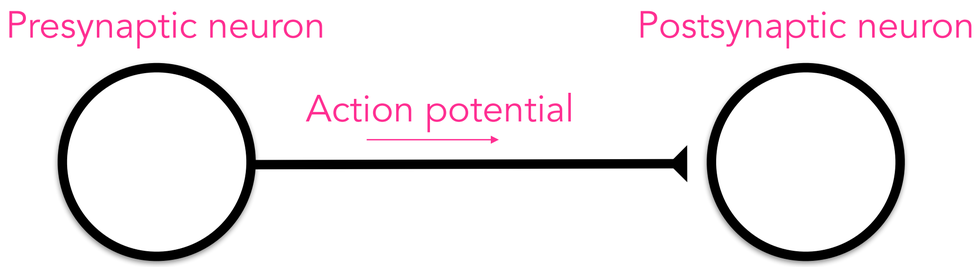
- The hippocampus
- Hopfield networks

4. Links to artificial neural networks

- The backpropagation rule
- Mapping deep learning to neurobiology

1. Synapses

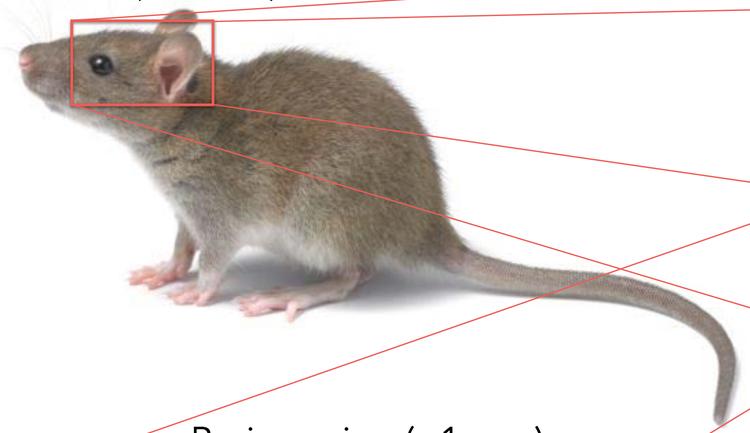
What is a synapse?



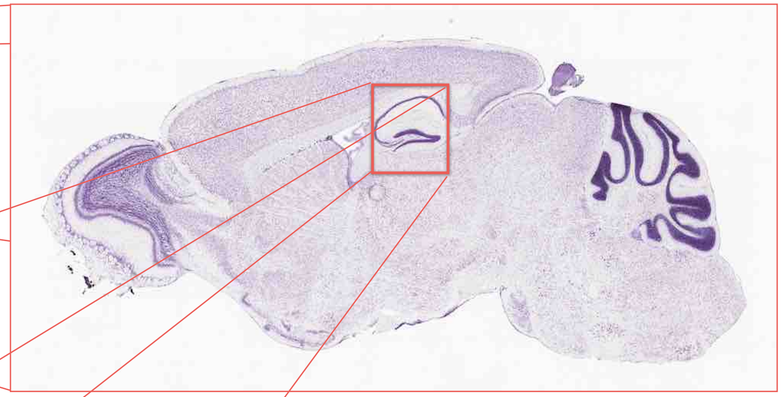
- Synapses are the connections between neurons.
- They convert the action potential from one neuron's axon into a 'post-synaptic-potential' in the dendrite of another neuron.

Zooming in on synapses

Animal (~10 cm)

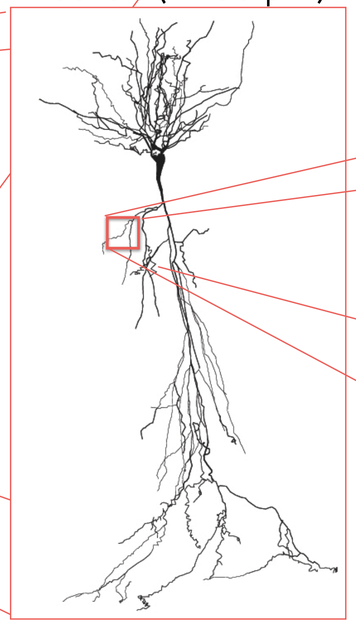


Brain (~1 cm)

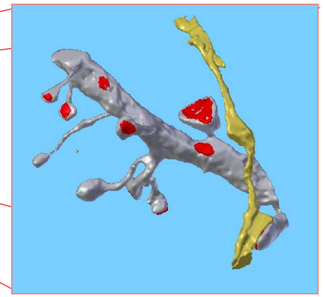


Brain region (~1 mm)

Neuron (~100 µm)



Synapses (~1 µm)



How do synapses work?

Axon
(presynaptic action potential)

Chemical signalling

Dendrite
(postsynaptic potential)

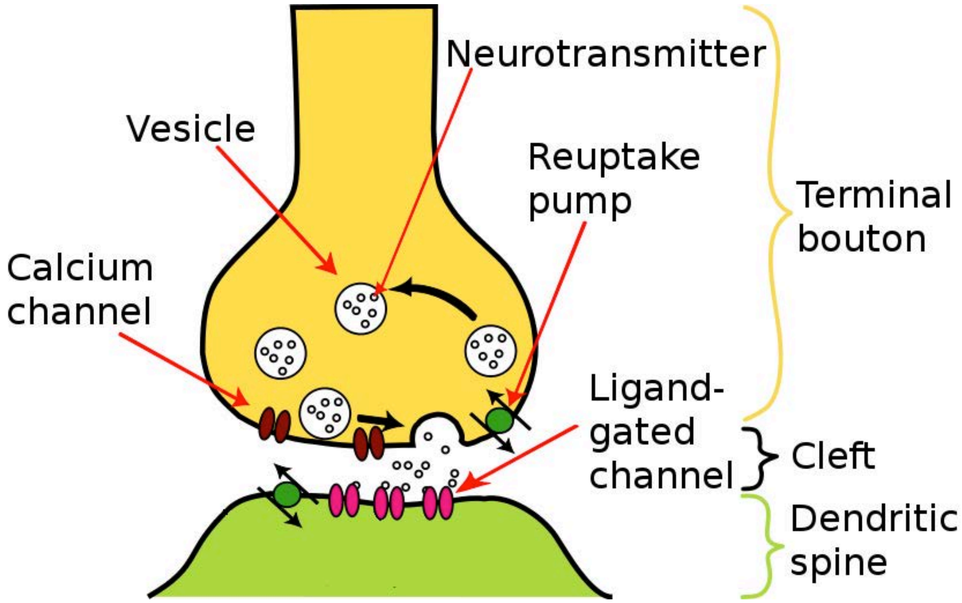
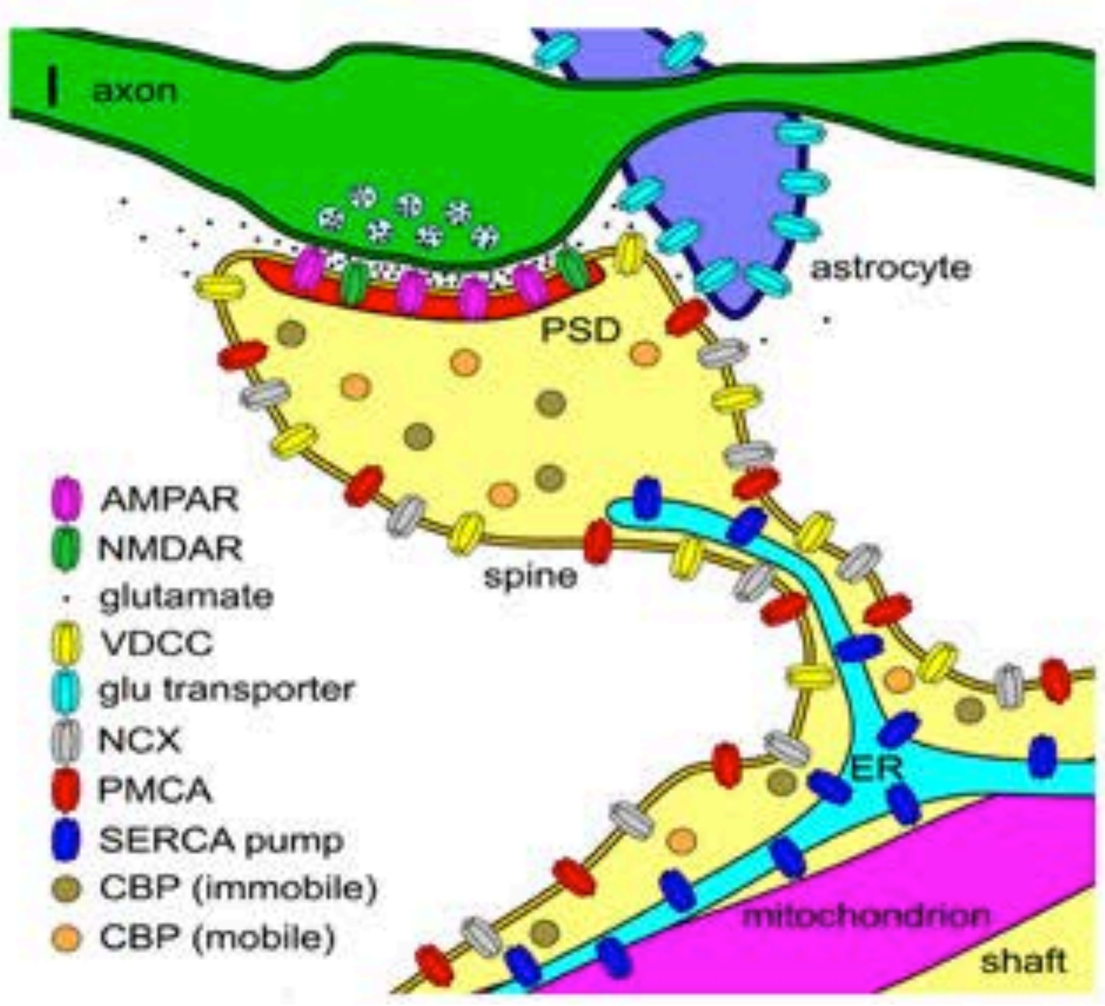
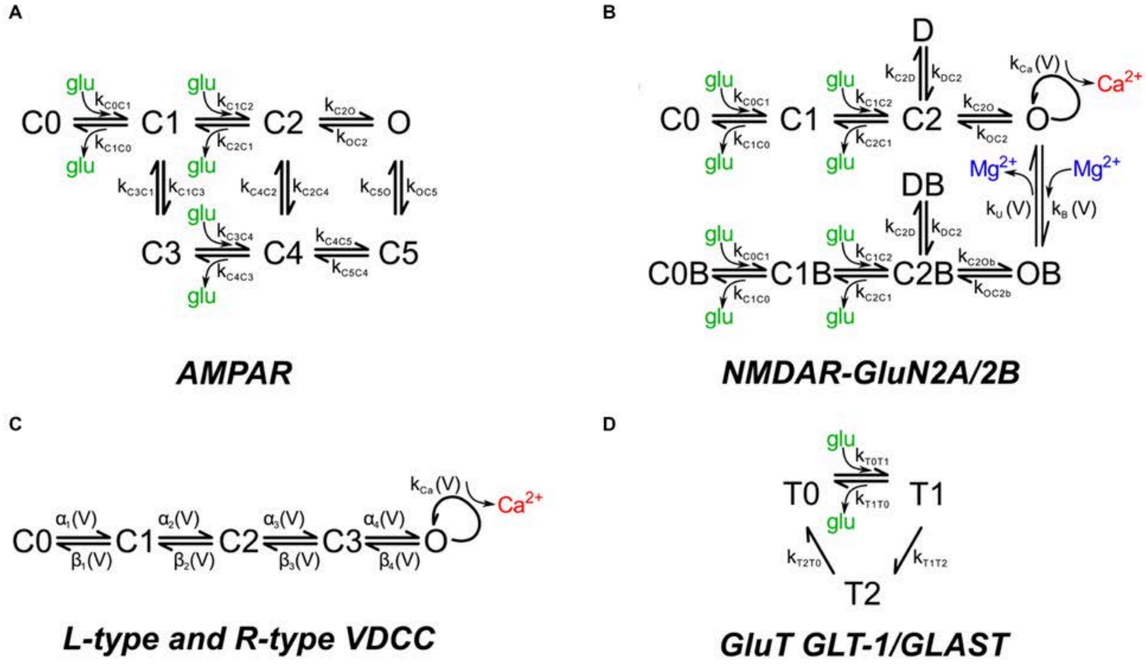


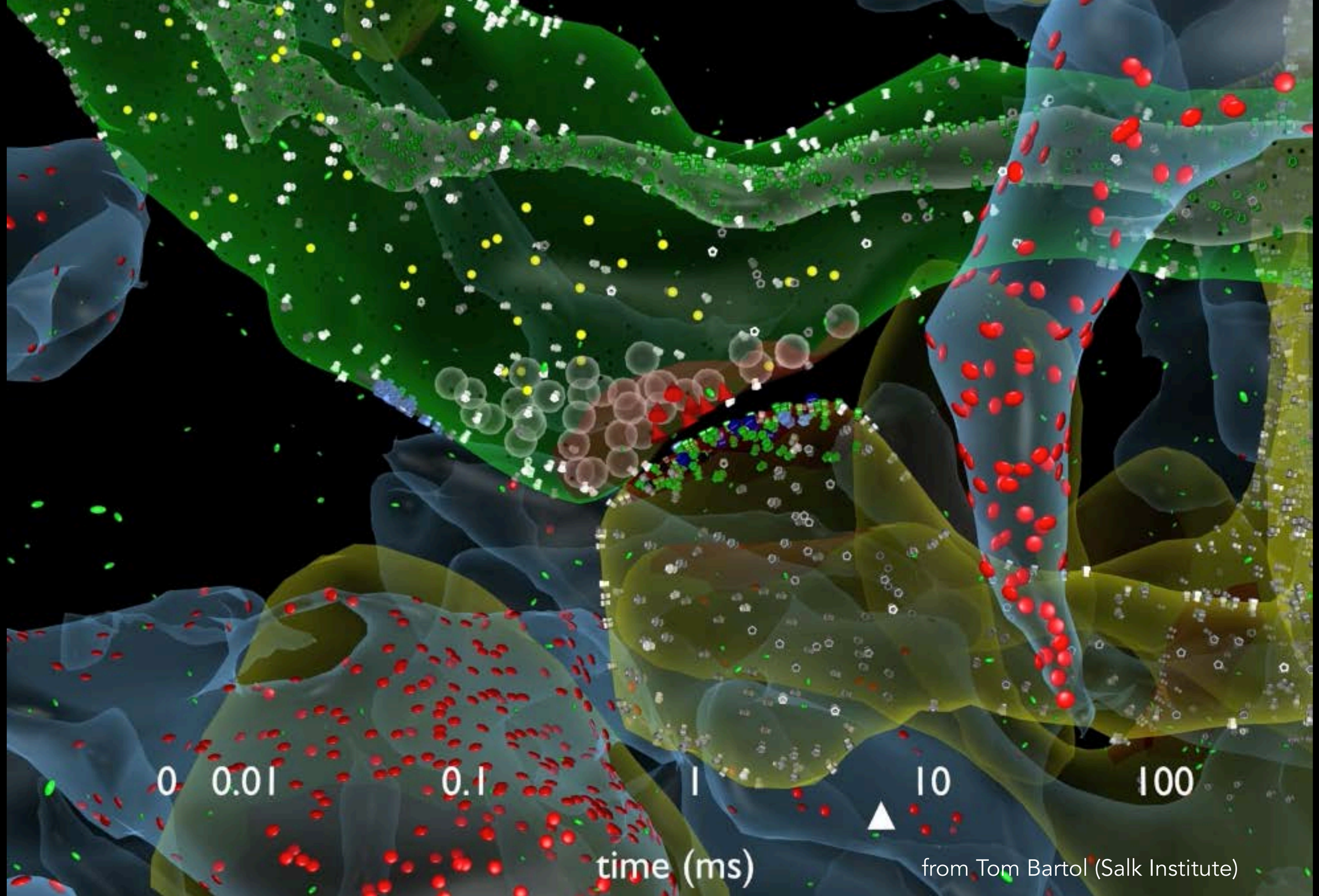
Image from Wikipedia (modified by C Houghton)

MCell simulation of synaptic release



Bartol et al, *Frontiers Syn Neuro* (2015)





from Tom Bartol (Salk Institute)

How can we computationally model a synapse?

- We *could* simulate the dynamics of each molecule involved in the signalling process (like the MCell simulation).
- But since that is very computationally expensive, we might instead go for a reduced mass-action chemical-kinetics model.
- However a lot of people still find even that too expensive and parameter-heavy, so instead use even simpler phenomenological models that black-box the synapse as a simple input-output system.

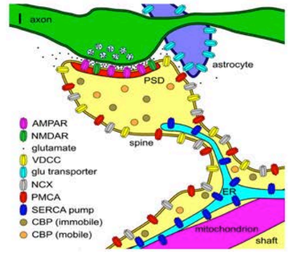
How detailed should a model be?

Details vs realism

c.f. Brette, *PLoS Comput Biol*, 2015

Models

Abstract ⟷ Realistic



Abstract models		**Realistic models**
Simple	vs	Detailed
Hard to relate to biology	vs	Contains stuff you could measure
Few parameters	vs	Lots of parameters
Fast simulation	vs	Slow simulation
Mathematical analysis	vs	Intractable
Generic	vs	Specific

Simple synapse models

The most common way to phenomenologically model a synapse is as a time-dependent conductor in series with a battery.

$$I_s(t) = \bar{g}_s s(t)(E_s - V)$$

The value of E_s determines whether the synapse is excitatory or inhibitory:

for excitatory synapses E_s usually = 0 mV

for inhibitory synapses E_s usually = V_{rest}

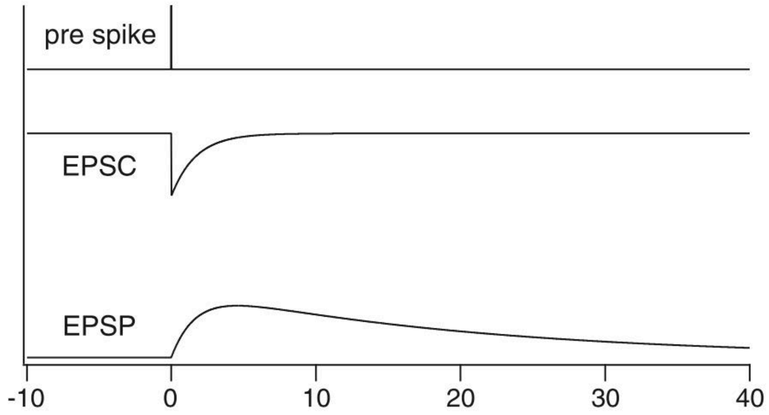
But how should we model $s(t)$?

Simple synapse models

Single exponential

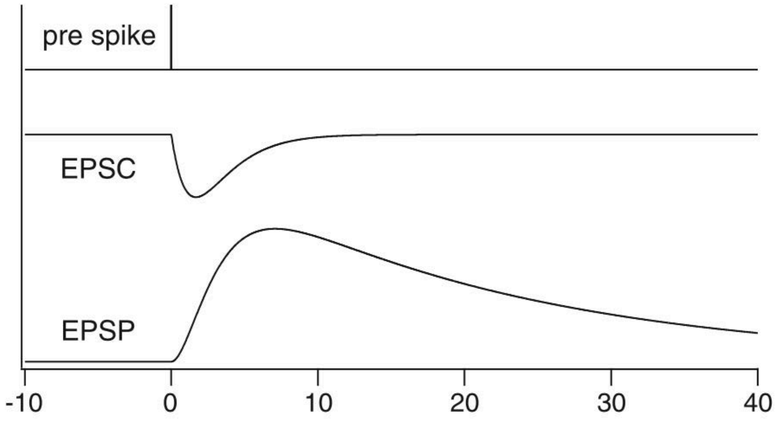
$$s(t) \rightarrow s(t) + 1$$

$$s(t) = e^{-t/\tau_s}$$



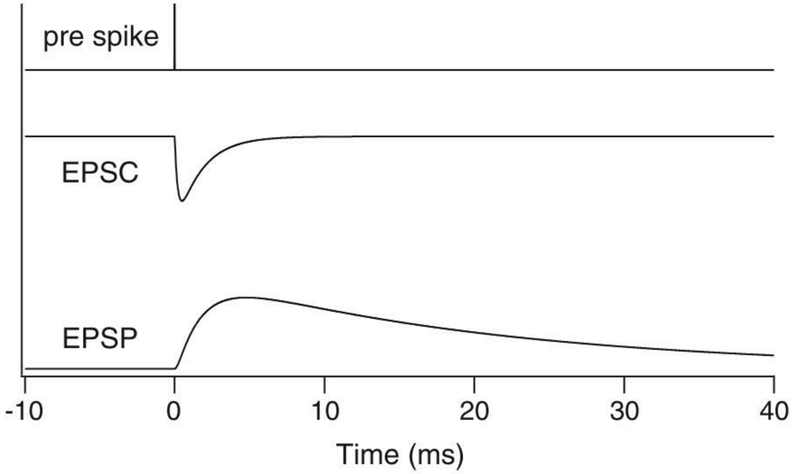
Alpha function

$$s(t) = te^{-t/\tau_s}$$



Difference of two exponentials

$$s(t) = e^{-t/\tau_{decay}} - e^{-t/\tau_{rise}}$$



Figures from Roth & van Rossum
http://homepages.inf.ed.ac.uk/mvanross/reprints/roth_mvr_chap.pdf

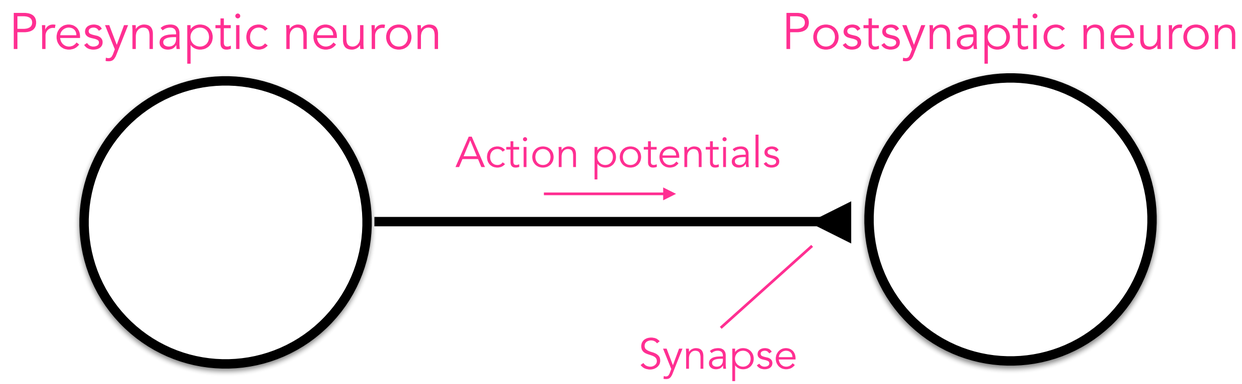
Summary on synapses

- Synapses are the connections between neurons.
- The convert the pre-synaptic action potential to a (excitatory or inhibitory) post-synaptic potential via a chemical intermediate stage.
- They are complicated molecular machines.
- We can model them at multiple levels of granularity, as appropriate for the task at hand.

2. Synaptic plasticity

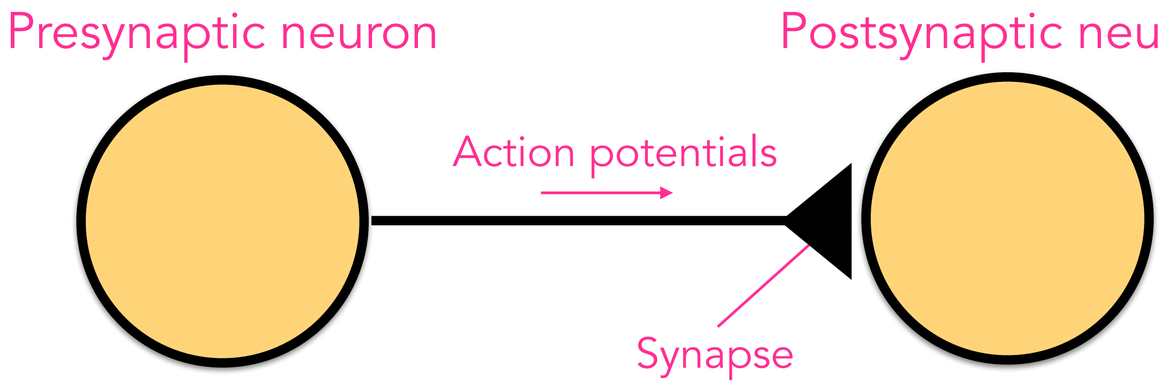
Long-term synaptic plasticity

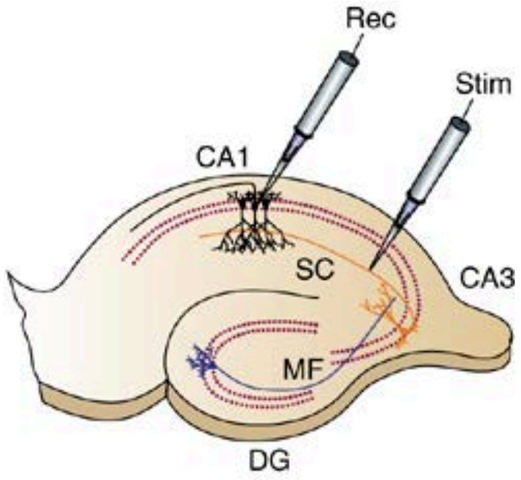
Long-term synaptic plasticity is a (activity-dependent) semi-permanent change in the strength of the connection from one neuron to another.

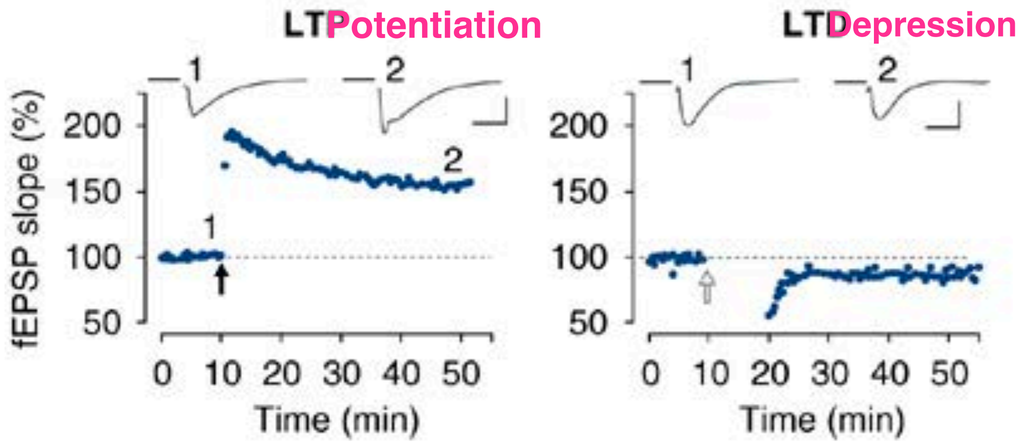


Long-term synaptic plasticity

Long-term synaptic plasticity is a (activity-dependent) semi-permanent change in the strength of the connection from one neuron to another.







Learning and memory via synaptic plasticity

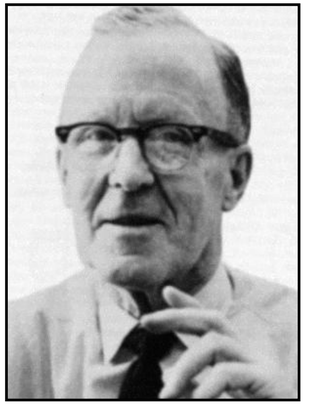
- Synaptic plasticity is generally believed to underlie long-term memory.
- Synapses change their strength according to certain 'rules of plasticity'.
- Synaptic plasticity is linked to learning and memory as follows:
  - Neural activity during learning triggers synaptic strength changes.
  - Synaptic strength changes alters the propensity for neurons to fire.
  - Next time the same neural circuit receives an input, it responds in a different fashion than it otherwise would have. **That's memory.**

What are these ‘rules of synaptic plasticity’?

Some mathematical function of the pre- and post-synaptic neurons’ activities… and maybe other stuff?

$$f(pre, post, weight, error, \ldots) = ?$$

Hebbian plasticity



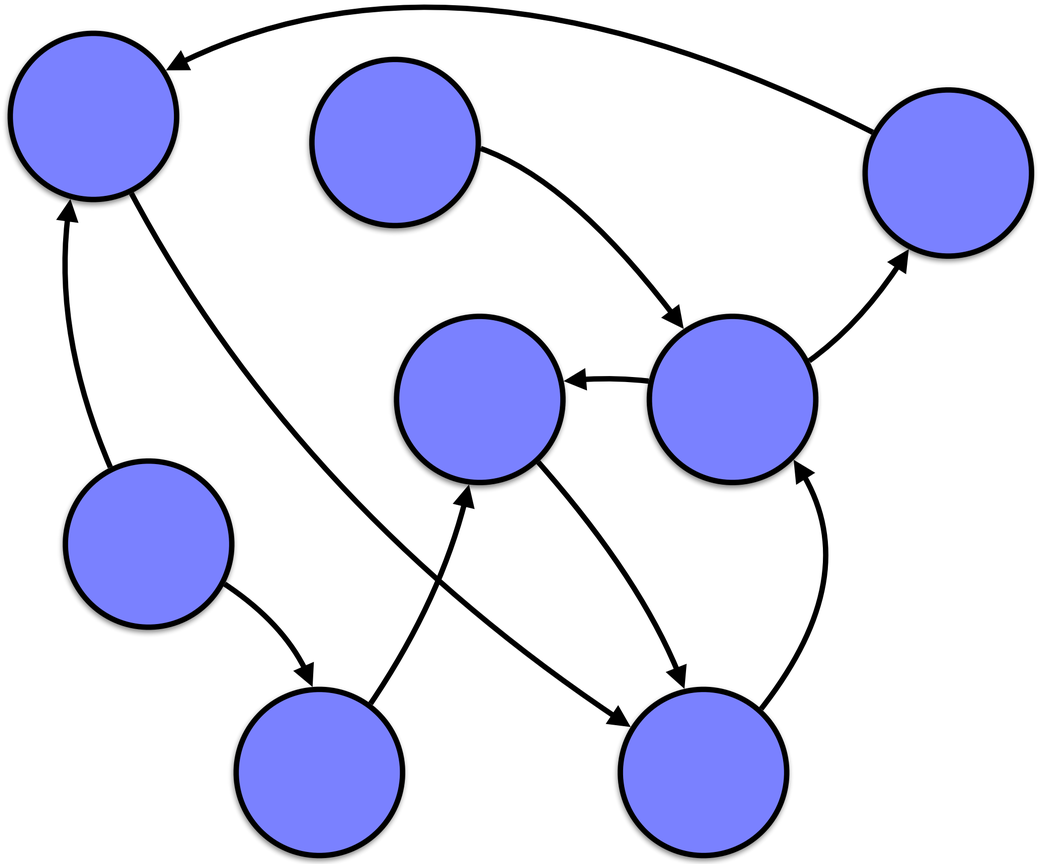
Donald Hebb

"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."

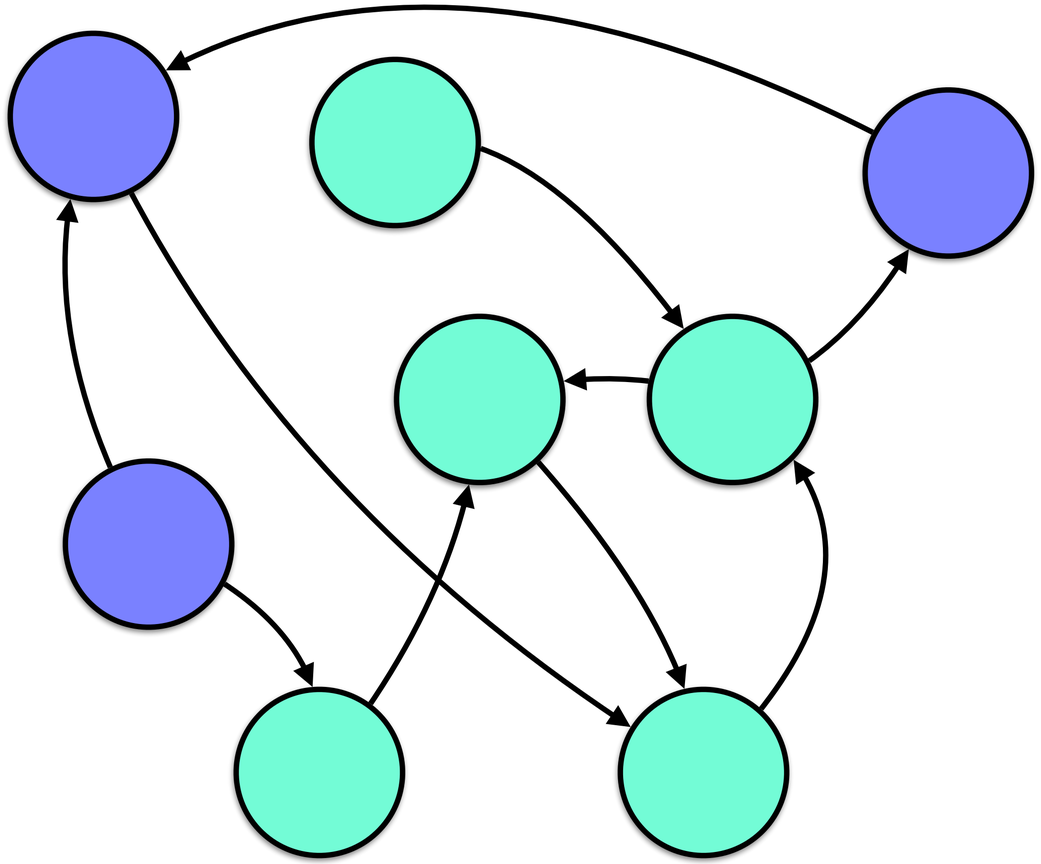
— Donald Hebb (1949)

a.k.a. "neurons that fire together wire together." (Carla Shatz)

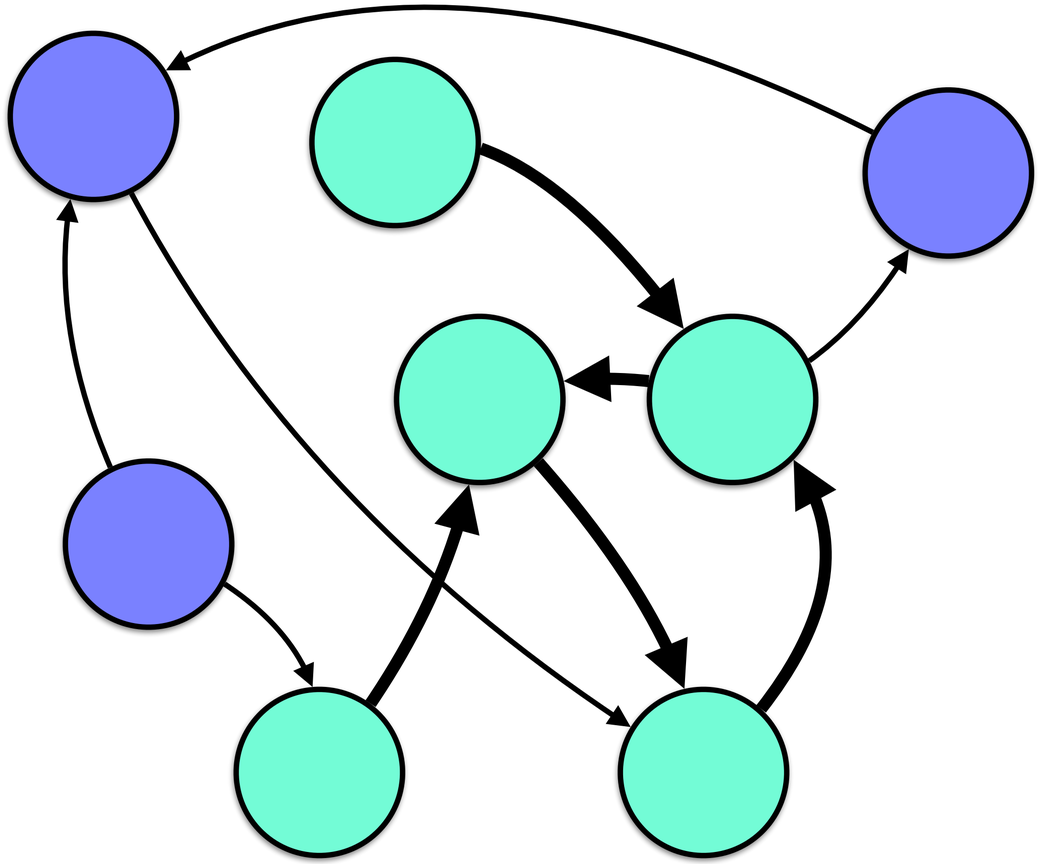
Hebbian plasticity



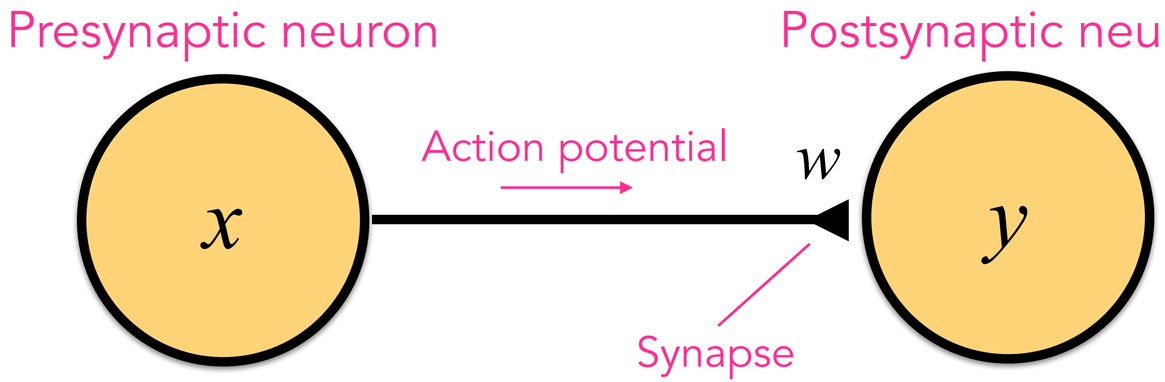
Hebbian plasticity



Hebbian plasticity



Modelling Hebbian plasticity



$$\Delta w = f(x, y) \quad = ???$$

A Hebbian rule: $\Delta w = \eta x y$

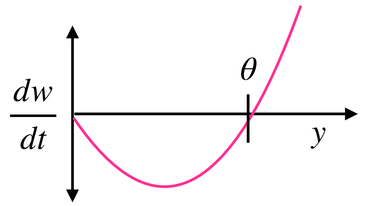
Note that dynamics are unstable: w and therefore y grow without bound.

How to stabilise plasticity?

Idea 1: BCM (Bienenstock, Cooper & Munro, 1982)

Modify the basic Hebbian rule by including a postsynaptic threshold for plasticity:

$$\frac{dw}{dt} = \eta_w xy(y - \theta)$$



The key idea for stability is that the threshold is also plastic:

$$\frac{d\theta}{dt} = \eta_\theta(y^2 - \theta_\infty)$$

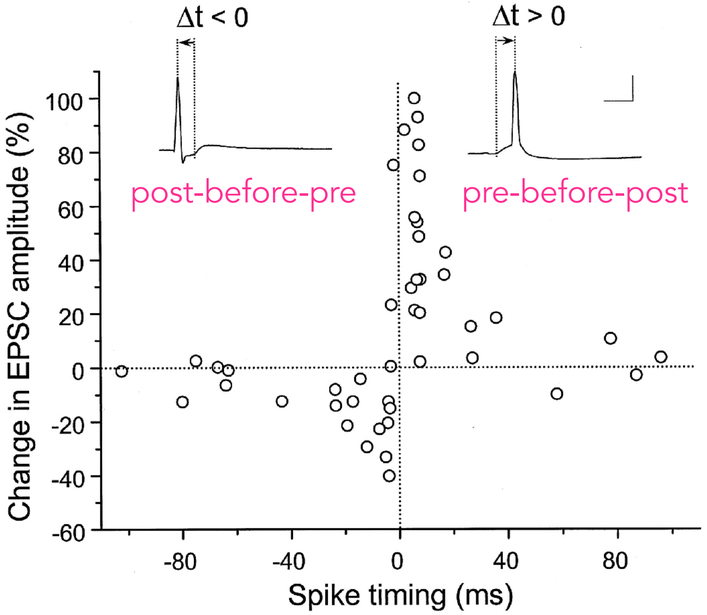
Idea 2: Homeostatic plasticity/synaptic scaling (Turrigiano et al, 1998)

Make the weights scale multiplicatively on a slow timescale, to keep postsynaptic activity at some target level:

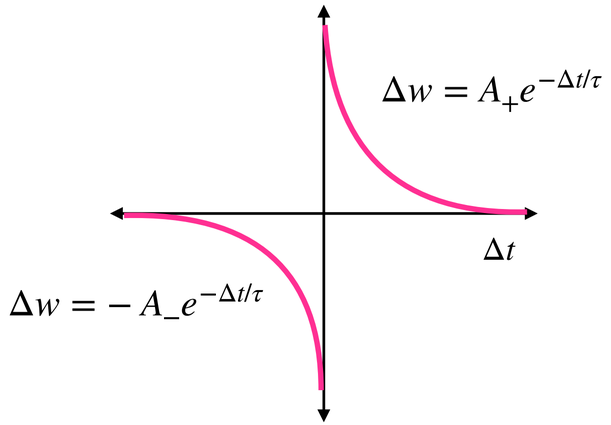
$$\frac{dw}{dt} = \eta_{ss} w(y_{target} - y)$$

Spike-timing-dependent plasticity

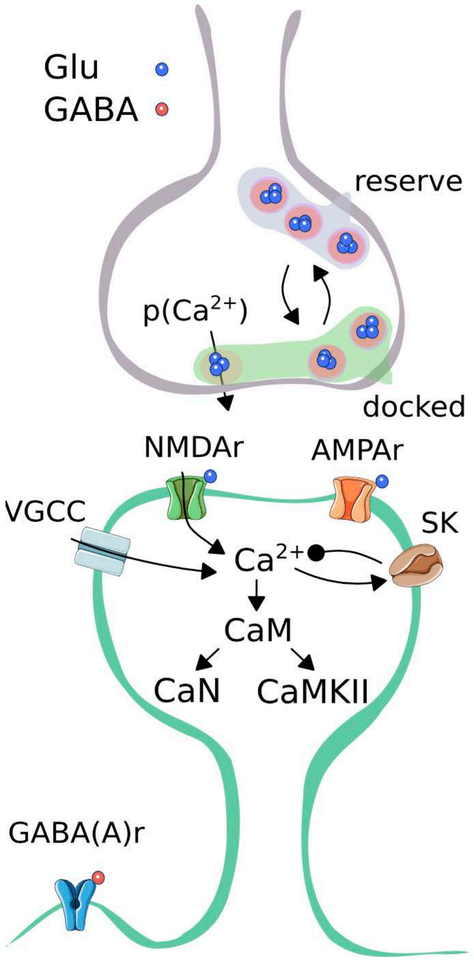
- STDP (discovered in late 1990s) encapsulates the idea of causality implied by Hebb:
  - if presynaptic spike A happened just *before* postsynaptic spike B, A *could have caused* B.
  - on the other hand, if presynaptic spike A happened just *after* postsynaptic spike B,
    A *could not have caused* B.
- Classic STDP: Pre-before-post causes LTP, post-before-pre causes LTD.
- STDP's existence implies that synapses can detect millisecond-level differences in spike timing when deciding whether to strengthen or weaken.
- When first discovered it was seen as the possible "atom of plasticity".
- "Things turned out to be just as simple as we first thought"
  — *No biologist, ever*

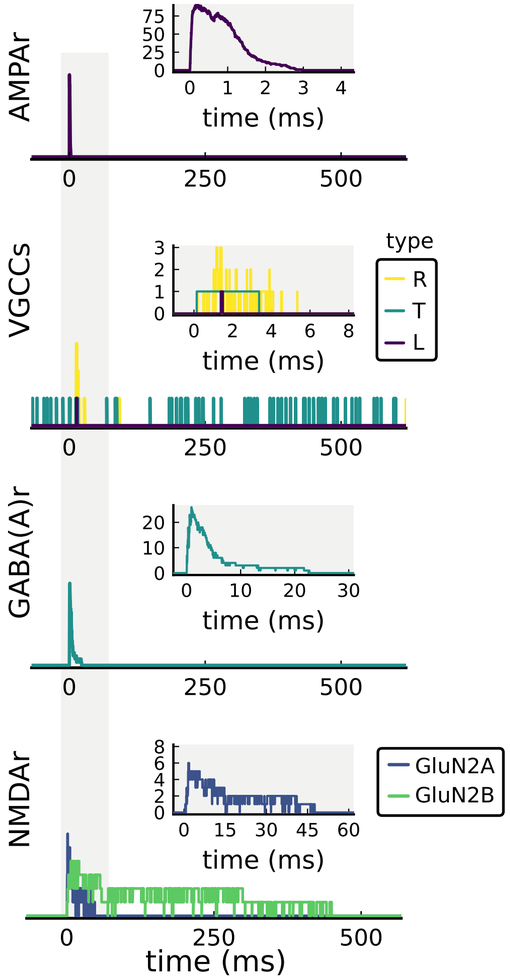


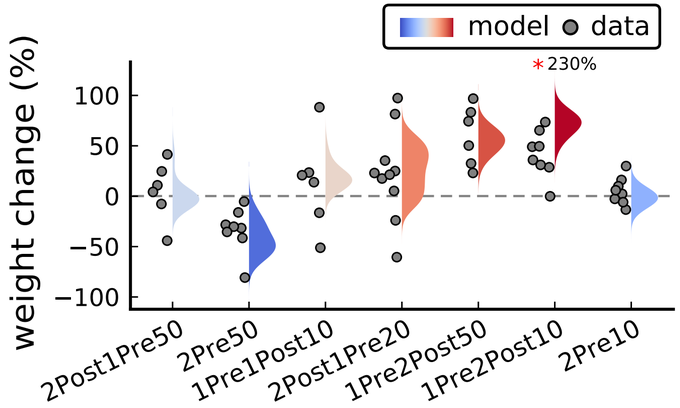
Bi & Poo, *J Neurosci* (1998)



Adding more biology







Rodrigues, Tigaret, Marie, O'Donnell, Veltz, *eLife* (2023)

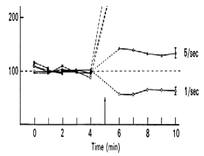
A brief history of learning rules


Long-term facilitation

Long-term potentiation

1960s

1973



Long-term depression

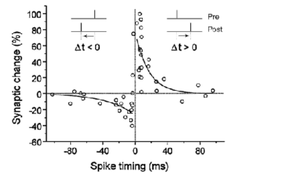
~1980

Metaplasticity

1980s

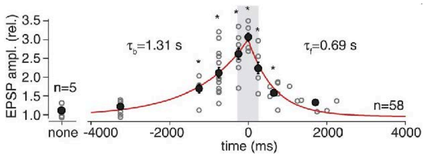
Homeostatic synaptic plasticity

mid-1990s



Spike-timing dependent plasticity

late 1990s



Behavioural timescale plasticity

late 2010s

1949

Hebbian Learning

1957

Perceptron

1960s

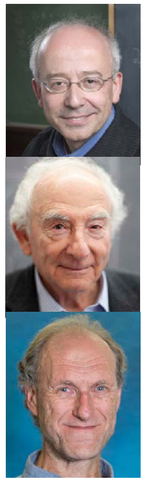
1973

Weight normalisation

~1980

1982

BCM



Oja's rule

1980s

Rise of backpropagation and ANNs

$$\frac{\partial E}{dw}$$

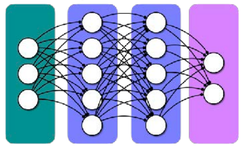
mid-1990s

Reinforcement learning

late 1990s

2010s

Deep learning



late 2010s

Approximate backprop

Where we are

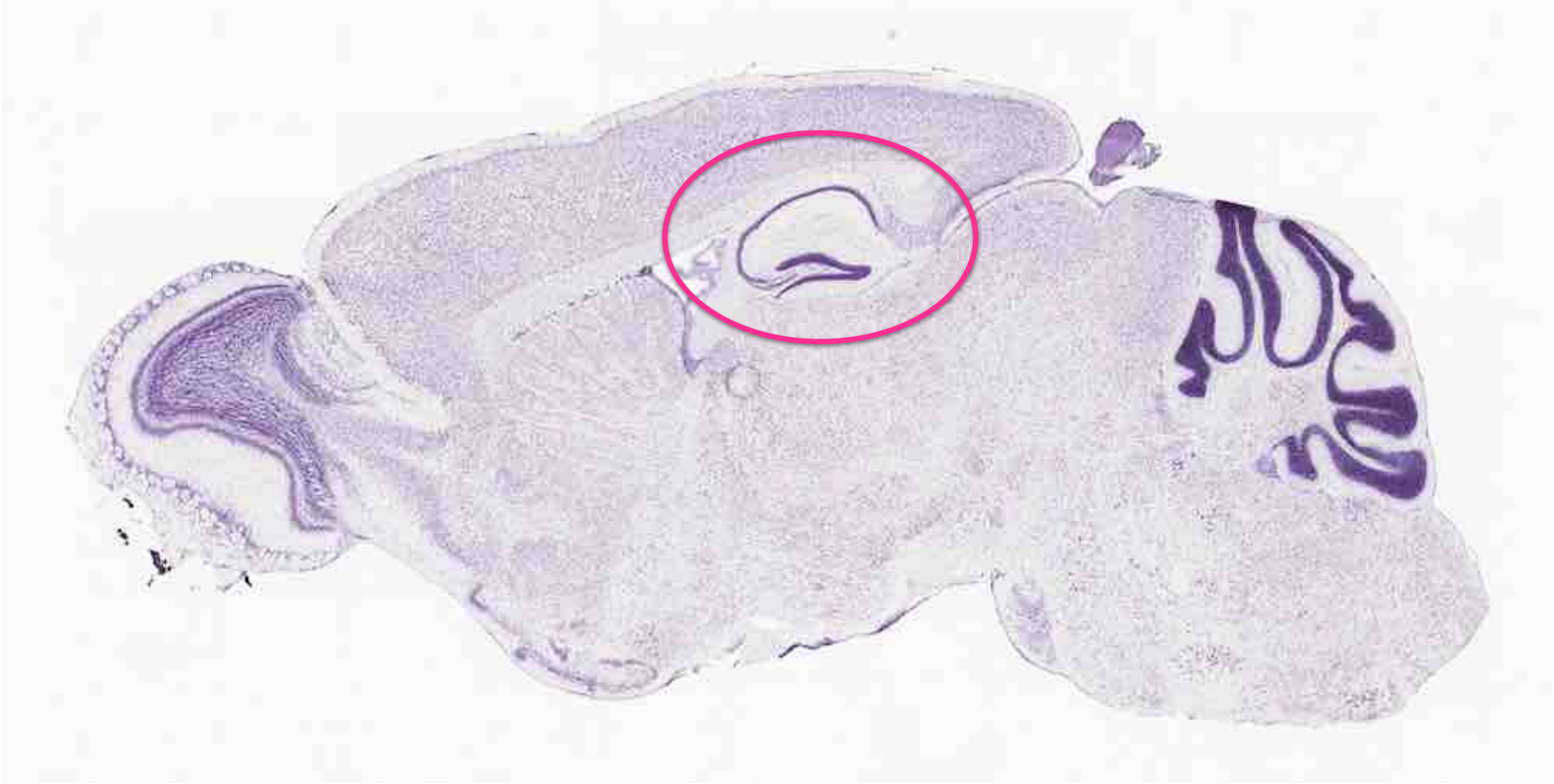
			
Learning type	**Unsupervised**	**Supervised**	**Reinforcement**
Example rules	Hebbian learning, STDP, BTSP	Backpropagation	Temporal Difference, Q-learning
Can achieve human-level performance on complex tasks?	Not really 😐	Yes, many and improving 🔥	Yes in specific cases 😎
Known mapping to neurobiology?	Pretty good	Several deep incompatibilities	Less clear, mostly high-level

3. Attractor networks

Hippocampus, from the greek words for “horse" and “sea-monster”

https://en.wikipedia.org/wiki/Hippocampus#/media/File:Hippocampus_and_seahorse_cropped.JPG

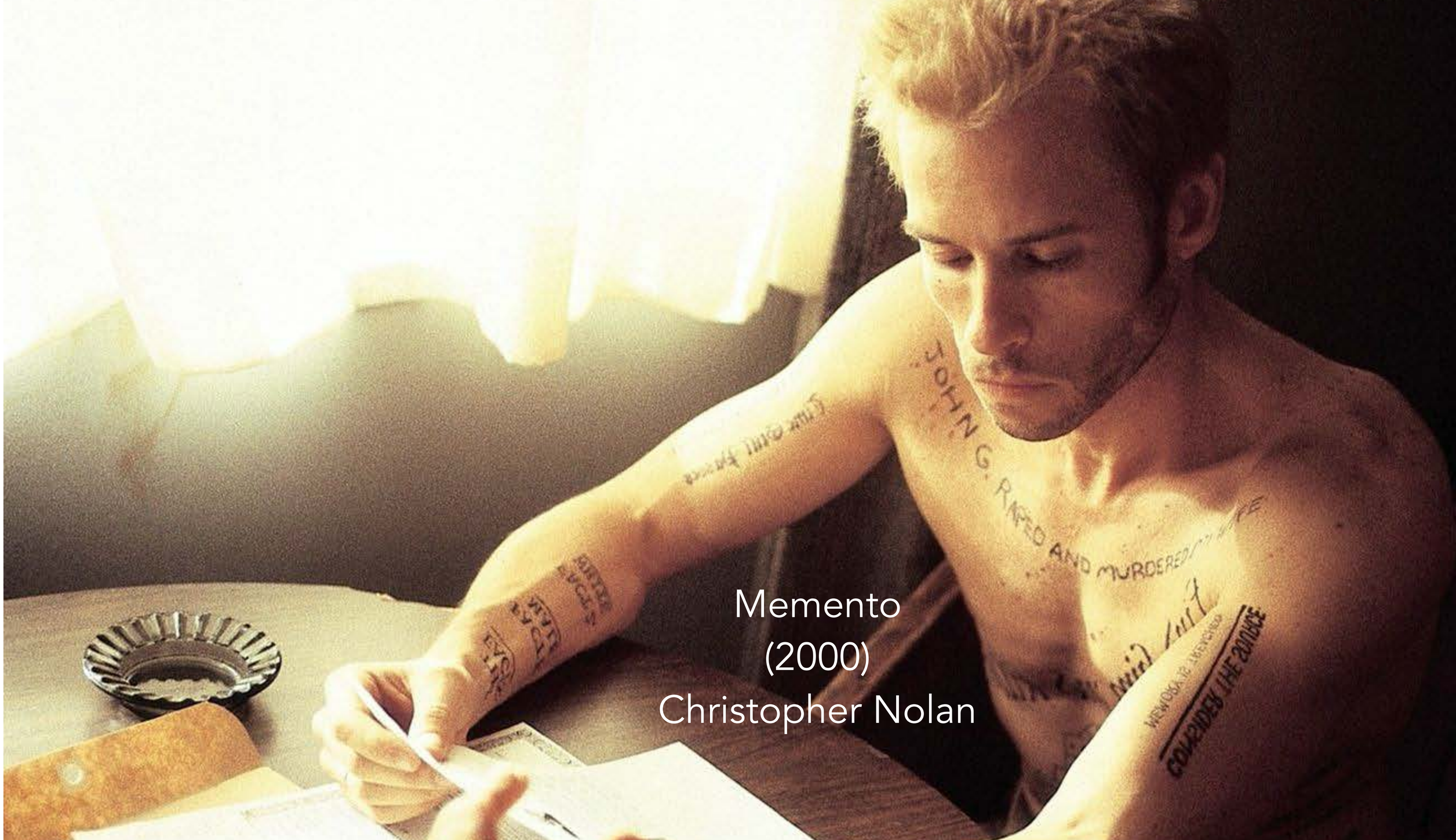
Anatomy of the hippocampus



https://mouse.brain-map.org

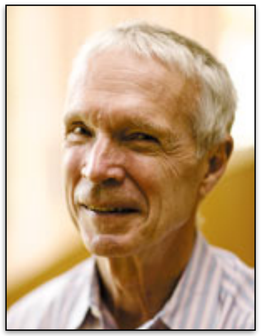
Hippocampus and memory

- Patient HM (who had his hippocampus surgically removed) could not form new long-term memories, and also had time-limited retrograde amnesia.
- The hippocampus is specifically needed for encoding new episodic memories, but is not necessary for other memories (e.g. procedural).
- Memory encoding requires synaptic plasticity in the hippocampus.



Memento
(2000)
Christopher Nolan

Hopfield networks



John Hopfield

- A basic model of associative memory recall, as in the CA3 region of the hippocampus.
- A Hopfield network is a recurrent network of “neurons” with synaptic connections set so to store memories as attractors (proposed in 1982).
- The network state evolves dynamically, typically toward some “attractor” state.
- A simple Hebb-like synaptic plasticity rule can imprint attractors in the network weights.
- Incredibly influential model in the history of computational neuroscience (attracted a bunch of physicists to the field).

Hopfield networks

- Network state evolves as:

$$\text{if } \sum_{j \neq i}^{N} [w_{ij} x_j(t) - \theta] > 0 \qquad \text{then } x_i \to +1$$

$$\text{otherwise } x_i \to -1$$

- There are two common flavours: synchronous or asynchronous updates.

- The weights follow a Hebbian-like rule:

$$w_{ij} = \frac{1}{N_{patterns}} \sum_{a}^{N_{patterns}} x_i^a x_j^a$$

- 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

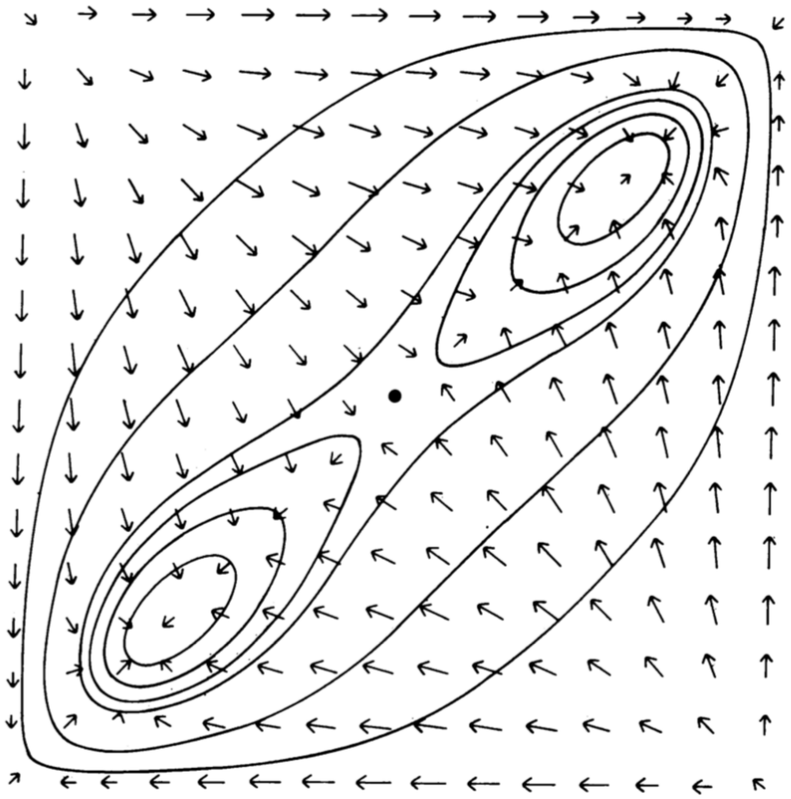


FIG. 3. An energy contour map for a two-neuron, two-stable-state system. The ordinate and abscissa are the outputs of the two neurons. Stable states are located near the lower left and upper right corners, and unstable extrema at the other two corners. The arrows show the motion of the state from Eq. 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 ~$0.14N$.

Content-addressable memory in Hopfield networks

Hertz, Krogh & Palmer, *Introduction to the Theory of Neural Computation* (1991)

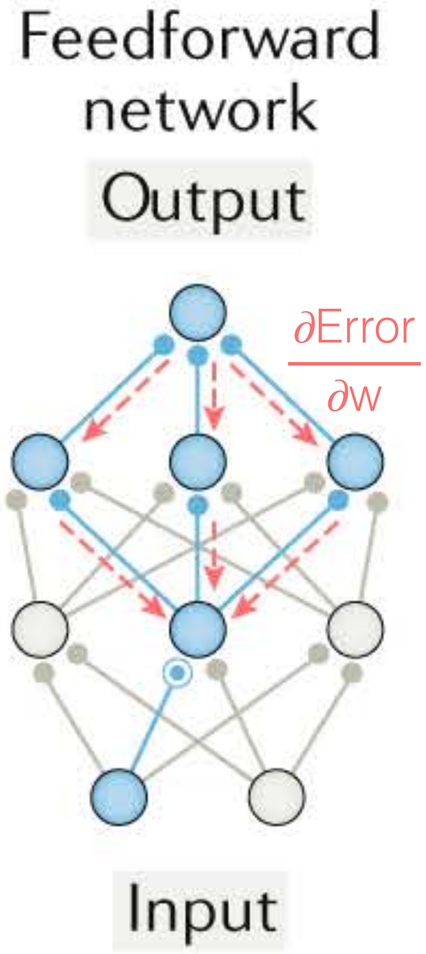
4. Links between learning in biological and artificial neural networks

The back-propagation learning rule

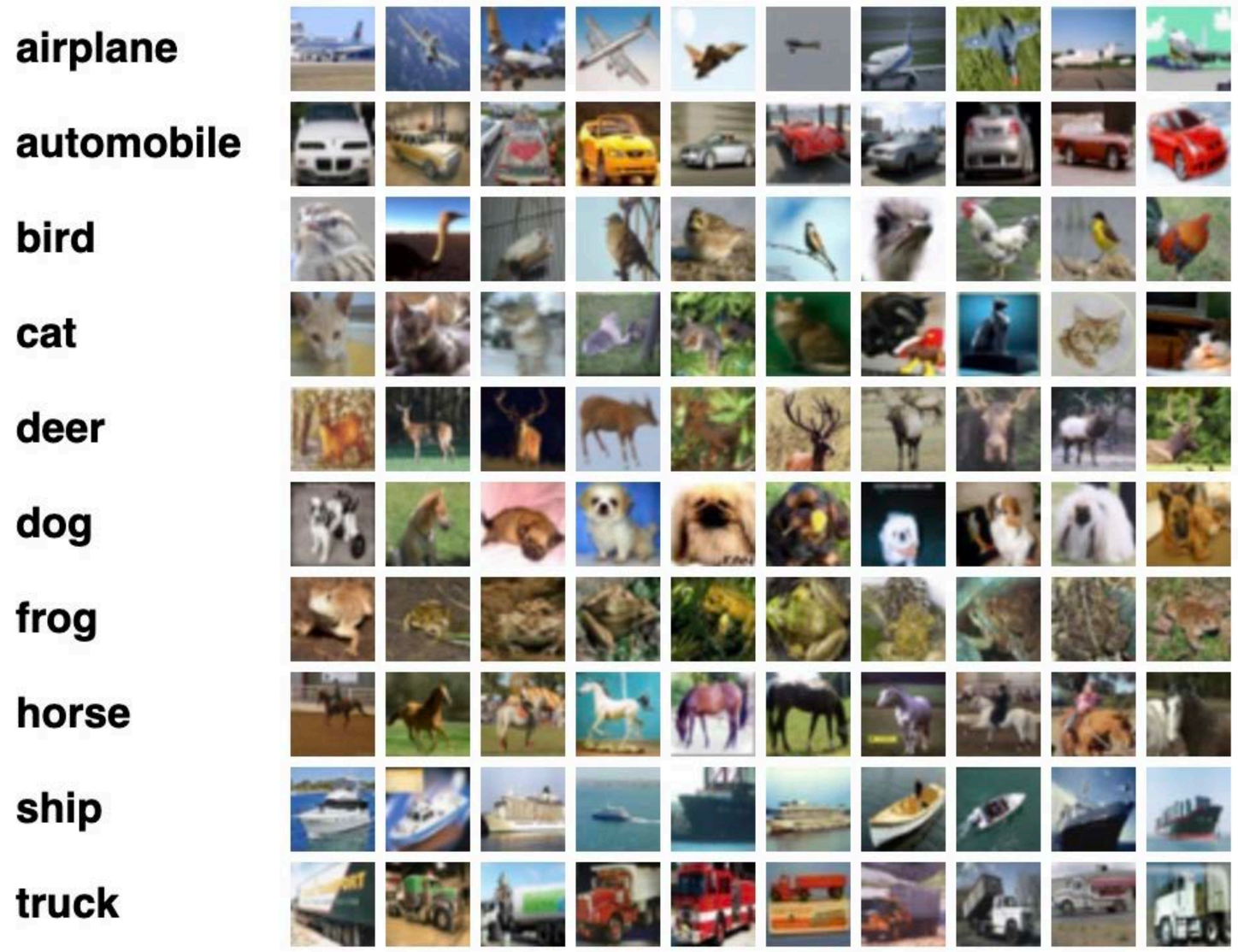
- Unlike Hebb's rule, in machine learning artificial neural networks are trained via *supervised* learning: performance errors are fed back into the system to adjust connection weights, so reducing future errors.

- The main method for doing this is called back-propagation (basically the calculus chain rule).

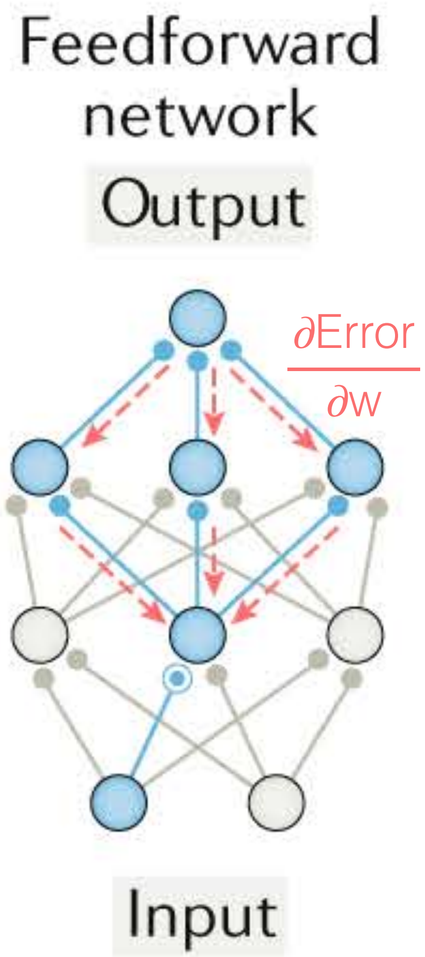
The back-propagation learning rule



Deep learning is everywhere



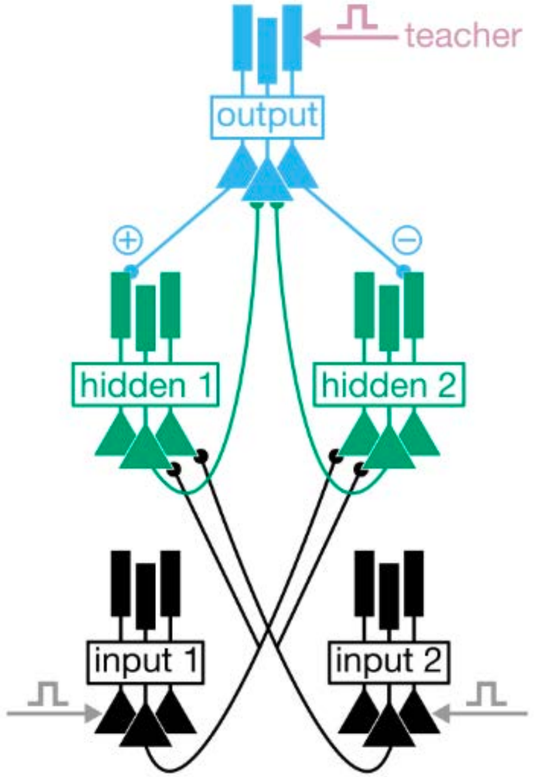
Issues with backprop as a model of brain learning



- Plasticity rules at each synapse needs non-local information.
- Requires symmetry in backwards and forwards weights.
- Error signals need to contain sign information (+/- error).
- Feedforward and feedback ‘passes’ need to be separated in time.

Example ideas for biologically plausible backprop

Burst-prop: feedback signals to distal dendrites, driving bursts that gate plasticity.



Payeur et al, *Nature Neurosci* (2021)

Deep learning and brains summary

- Backpropagation of error gradients is a ubiquitous algorithm for learning in artificial neural networks.
- But the basic version in not biologically realistic.
- Current research is trying to dream up biologically plausible alternatives that approximate backprop.
- However the idea that the brain even learns by supervised gradient descent is controversial! Maybe *unsupervised* or *reinforcement learning* rules are all it takes.

Overall session summary

1. Synapses transmit signals between neurons. Changes in their strength underlie long-term memory.

2. Classic plasticity rules: Hebb, STDP.

3. The hippocampus as a Hopfield attractor network.

4. Does the brain do deep learning?? Maybe.

Thanks