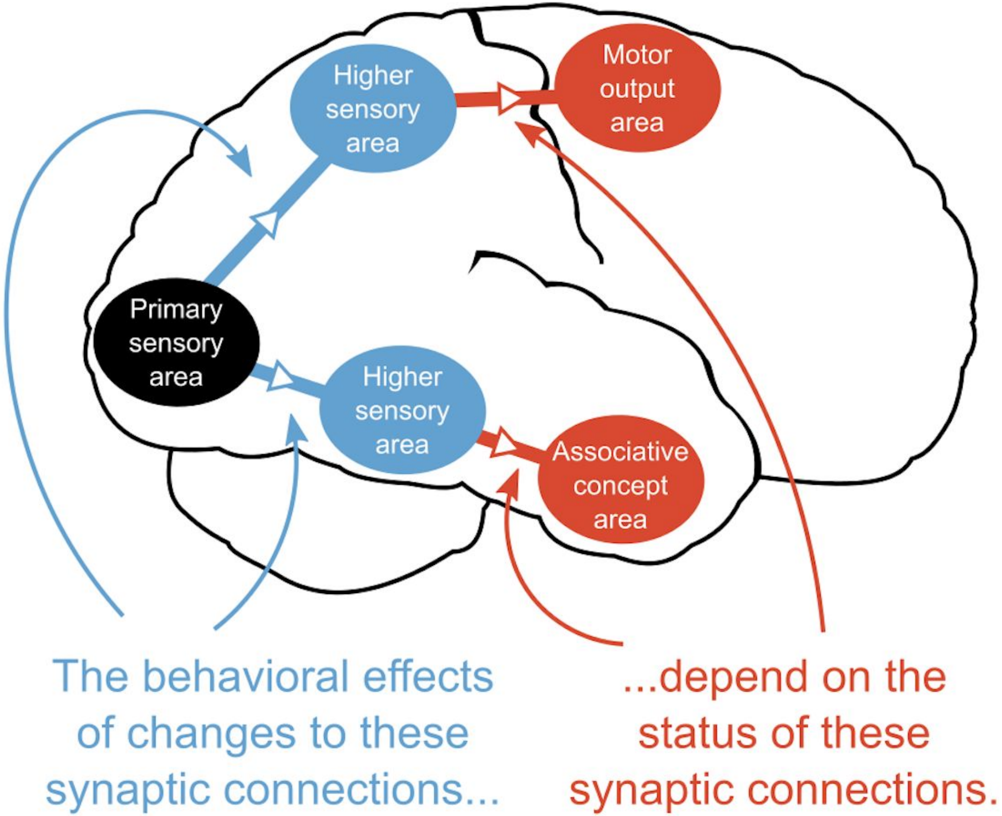
# Backpropagation and Deep Learning in the Brain

Simons Institute -- Computational Theories of the Brain 2018

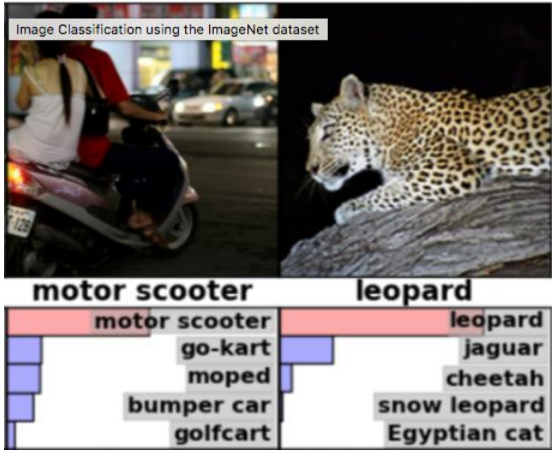
Timothy Lillicrap
DeepMind, UCL

With:
Sergey Bartunov, Adam Santoro, Jordan Guerguiev, Blake Richards,
Luke Marris, Daniel Cownden, Colin Akerman, Douglas Tweed, Geoffrey Hinton

# The “credit assignment” problem



# The solution in artificial networks: backprop

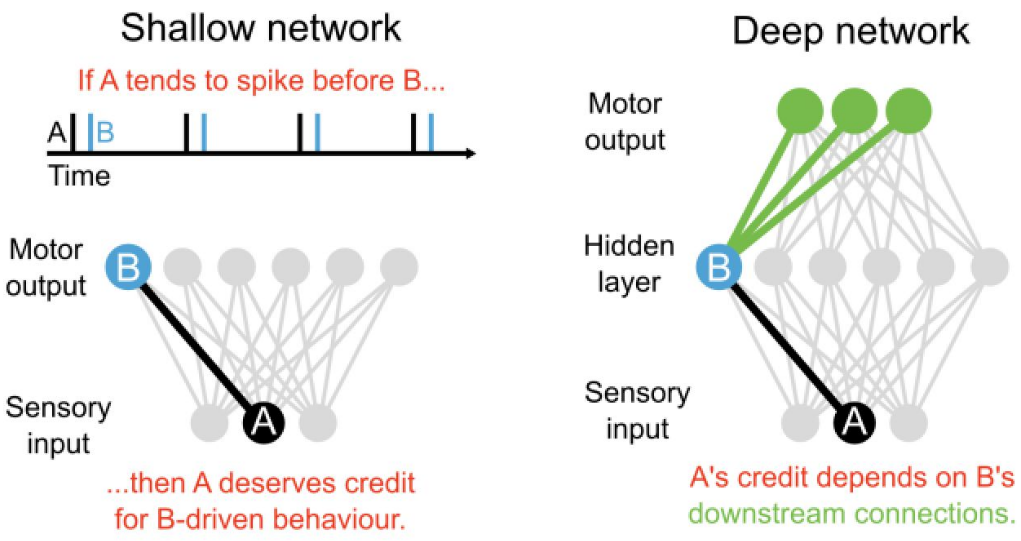



$$\Delta W = -\eta \frac{\partial E}{\partial W}$$

Credit assignment by backprop works well in practice and shows up in virtually all of the state-of-the-art supervised, unsupervised, and reinforcement learning algorithms.

The key feature of deep learning that is missing from current neuroscience models is the ability to perform **credit assignment** in hierarchical, multilayer networks

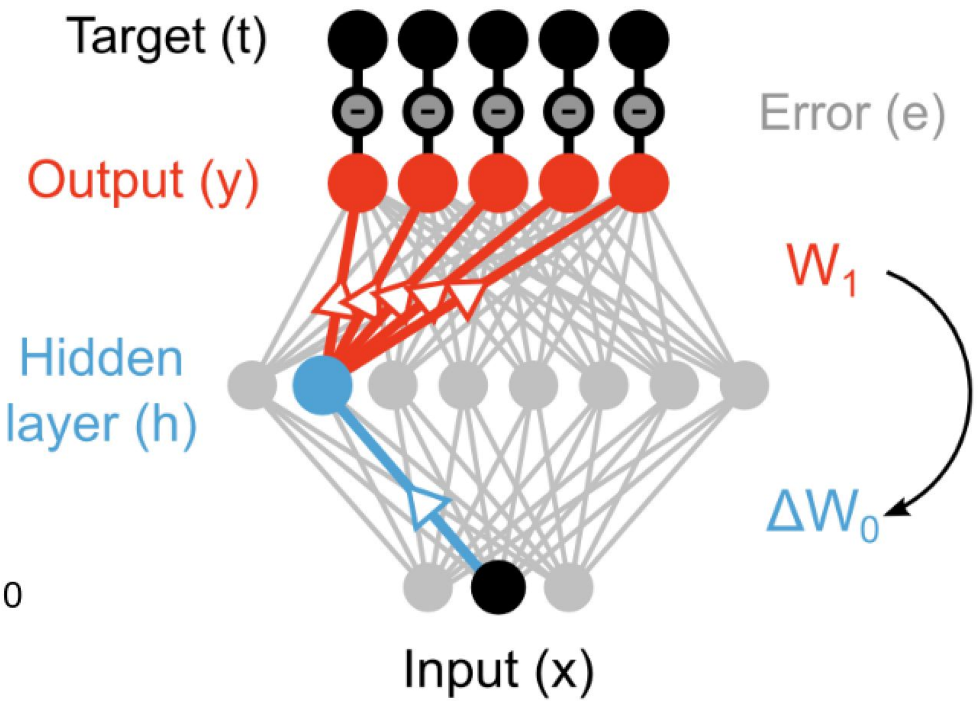
Assigning neurons/synapses “credit” (or “blame”) for their contributions to behavioural output in a deep network is non-trivial; standard Hebbian learning and/or global reinforcement signals insufficient



# Why Isn't Backprop "Biologically Plausible"?

$$u = W_0 x,\; v = W_1 h$$
$$h = \sigma(u),\; y = \sigma(v)$$
$$e = (y - t),\; L = \frac{1}{2} e^2$$

$$\Delta W_0 \propto \frac{\partial L}{\partial W_0}$$
$$= \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial u} \cdot \frac{\partial u}{\partial W_0}$$
$$= e \cdot W_1^T \cdot \sigma'(u) \cdot x$$

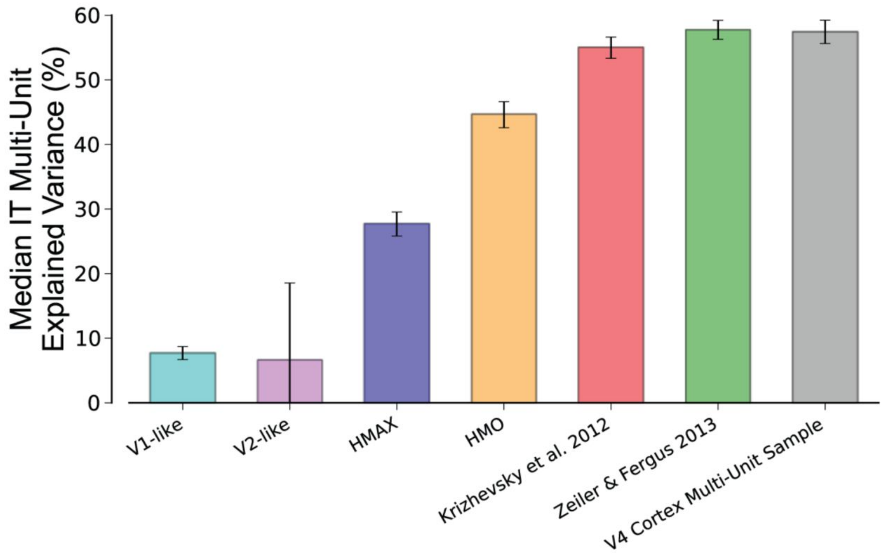


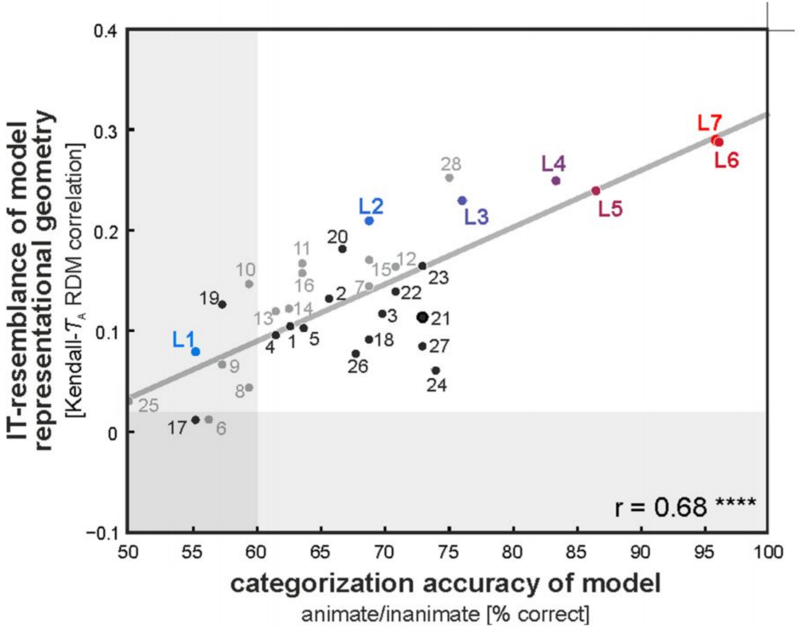
# Why Isn't Backprop "Biologically Plausible"?

$$\Delta W_0 \propto e \cdot W_1^T \cdot \sigma'(u) \cdot x$$

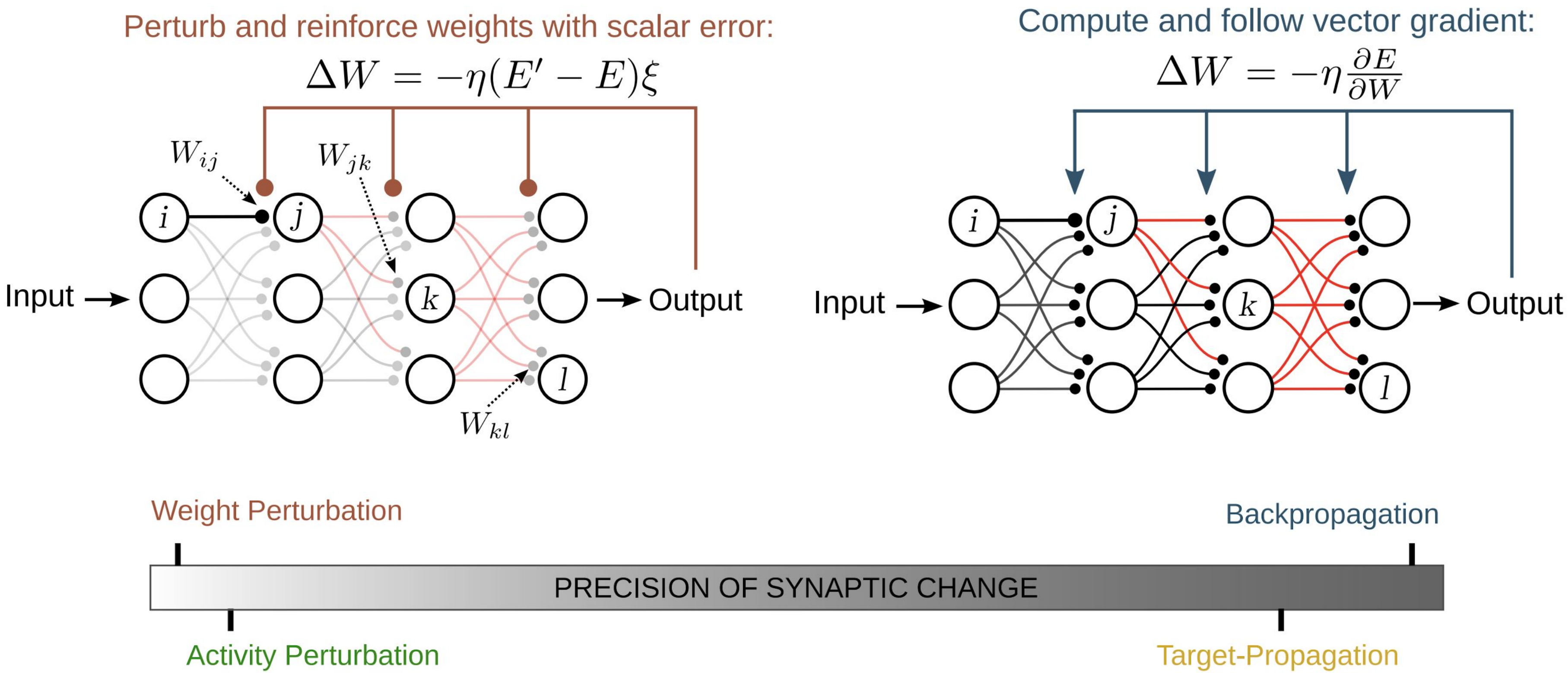
We need:
(1) Error term
(2) Transpose of downstream weights
(3) Derivative of activation function
(4) Separate forward and backward passes

# Neuroscience Evidence for Backprop in the Brain?

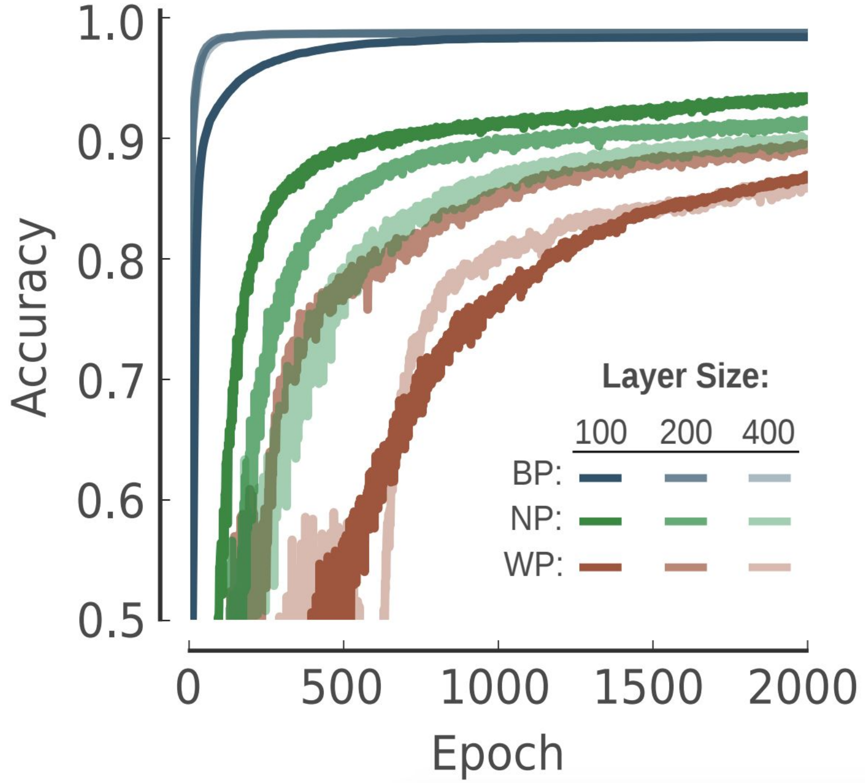


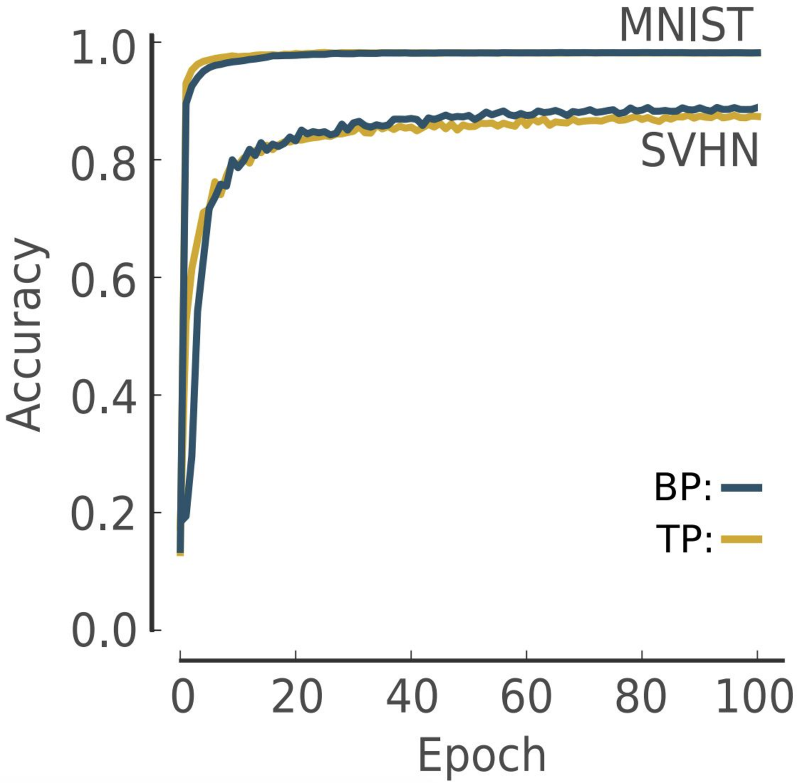
# A spectrum of credit assignment algorithms:



# A spectrum of credit assignment algorithms:



# A spectrum of credit assignment algorithms:



# How to convince a neuroscientist that the cortex is learning via [something like] backprop

- To convince a machine learning researcher, an appeal to variance in gradient estimates might be enough.

- But this is rarely enough to convince a neuroscientist.

- So what lines of argument help?

# How to convince a neuroscientist that the cortex is learning via [something like] backprop

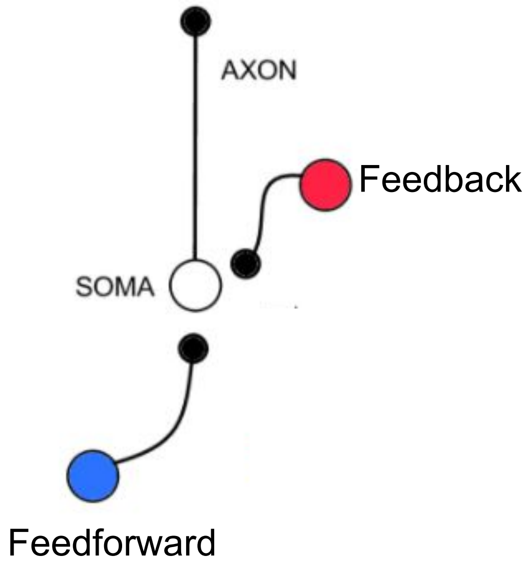
- What do I mean by “something like backprop”?:

- That learning is achieved across multiple layers by sending information from neurons closer to the output back to “earlier” layers to help compute their synaptic updates.

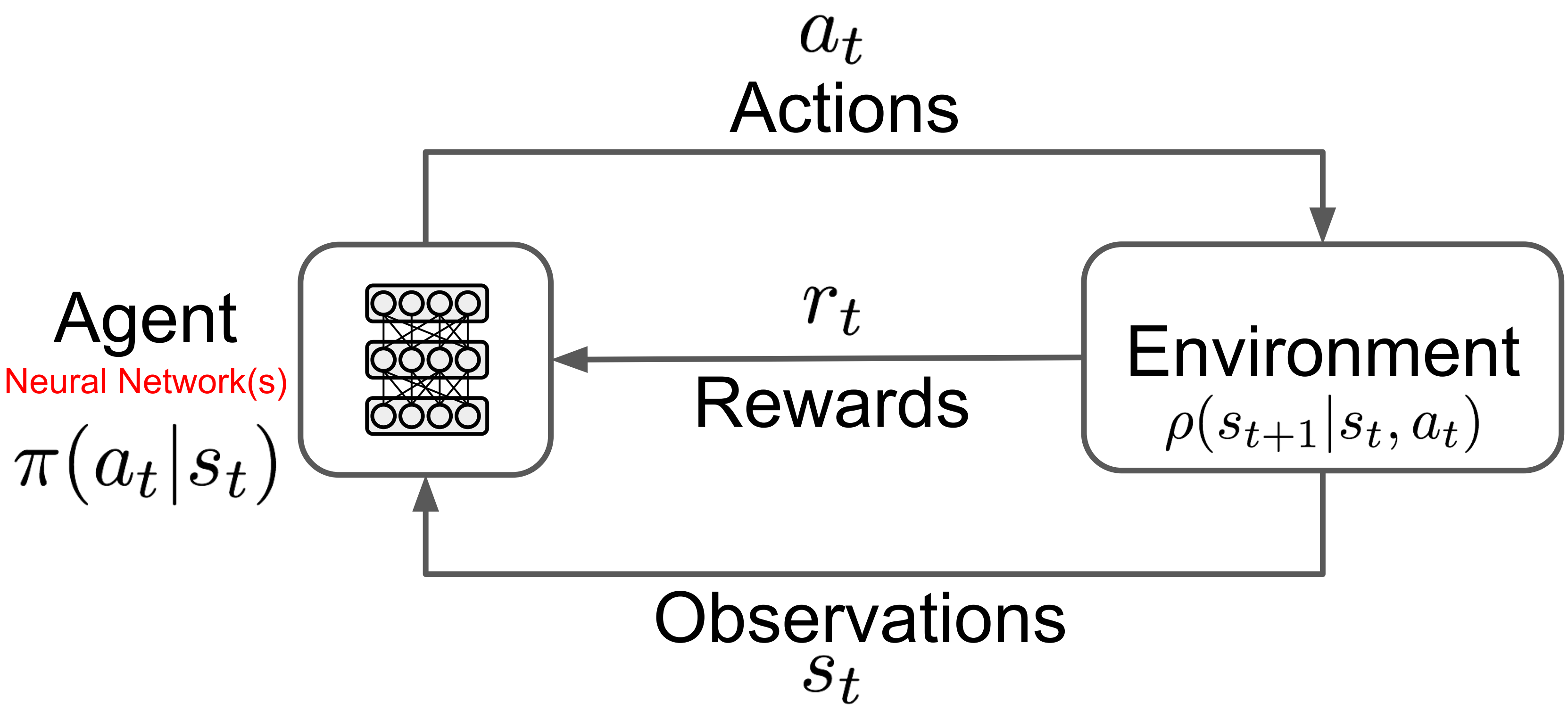
# How to convince a neuroscientist that the cortex is learning via [something like] backprop

1. Feedback connections in cortex are ubiquitous and modify the activity/spiking of neurons in earlier layers.
2. Plasticity in “forward” connecting synapses is driven by pre- and post-synaptic spiking (e.g. STDP).

3. Thus, feedback connections *will* modify learning at forward connecting synapses!



# What about reinforcement learning?



# A Single Trial of Reinforcement Learning

$$
\begin{array}{ccccc}
r_0, & r_1, & r_2, & \dots, & r_T \\
a_0, & a_1, & a_2, & \dots, & a_T \\
\pi(a_0|s_0), & \pi(a_1|s_1), & \pi(a_2|s_2), & \dots, & \pi(a_T|s_T) \\
s_0, & s_1, & s_2, & \dots, & s_T \\
 & \rho(s_1|s_0, a_0), & \rho(s_2|s_1, a_1), & \dots, & \rho(s_T|s_{T-1}, a_{T-1})
\end{array}
$$

Time $\longrightarrow$

Probability of trajectory $\tau$

$$p_\theta(\tau) = \rho(s_0) \prod_{t=0}^{T} \rho(s_{t+1}|s_t, a_t) \pi_\theta(a_t|s_t)$$

# Measuring Outcomes

Return for a single trial:

$$R(\tau) = \sum\nolimits_{t=0}^{T} \gamma^t r_t$$

Objective function:

$$J(\theta) = \int_{\mathbb{T}} p_\theta(\tau) R(\tau) d\tau$$

# Update Parameters with the Policy Gradient

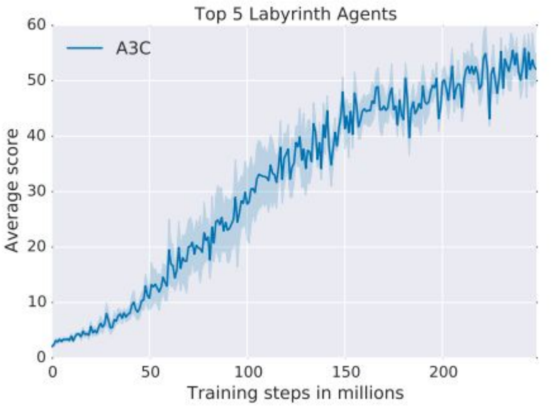
1. Sample a trajectory by rolling out the policy:

$$\tau \sim \begin{array}{ccccc} r_0, & r_1, & r_2, & \ldots, & r_T \\ a_0, & a_1, & a_2, & \ldots, & a_T \\ \pi(a_0|s_0), & \pi(a_1|s_1), & \pi(a_2|s_2), & \ldots, & \pi(a_T|s_T) \\ s_0, & s_1, & s_2, & \ldots, & s_T \\ & \rho(s_1|s_0, a_0), & \rho(s_2|s_1, a_1), & \ldots, & \rho(s_T|s_{T-1}, a_{T-1}) \end{array}$$

2. Compute an estimate of the policy gradient and update network parameters:

$$\theta_{i+1} = \theta_i + \eta \nabla_\theta \hat{J(\theta)}|_{\theta=\theta_i}$$

# Training Neural Networks with Policy Gradients




Still 100s of Millions of steps,
*even with backprop*.


Mnih et al., *ICML* 2016

# Why Isn't Backprop "Biologically Plausible"?

$$u = W_0 x, \; v = W_1 h$$

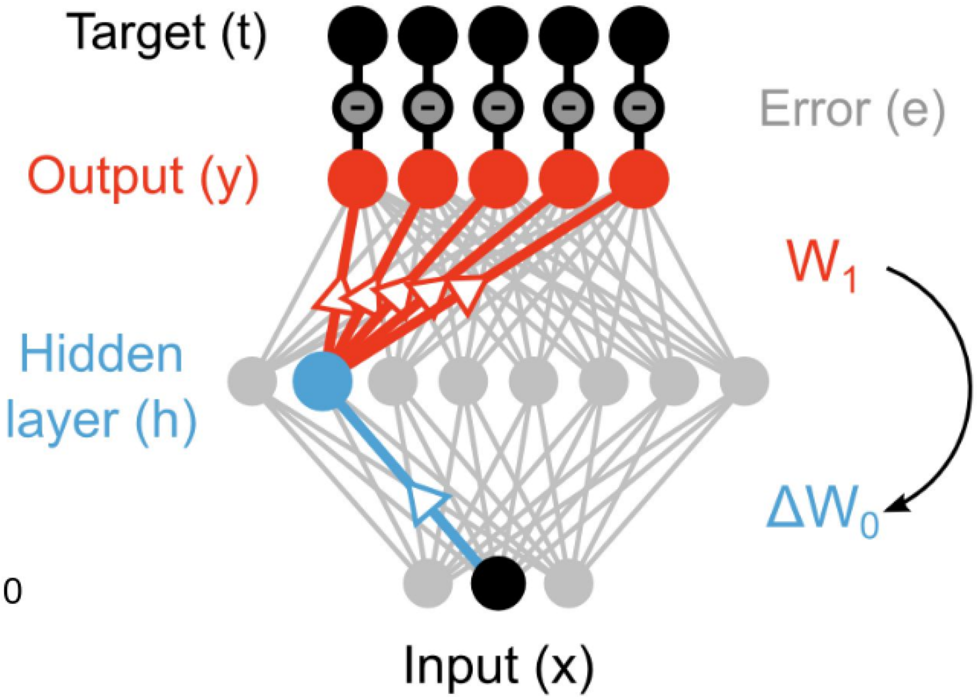
$$h = \sigma(u), \; y = \sigma(v)$$

$$e = (y - t), \; L = \frac{1}{2} e^2$$

$$\Delta W_0 \propto \frac{\partial L}{\partial W_0}$$

$$= \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial u} \cdot \frac{\partial u}{\partial W_0}$$

$$= e \cdot W_1^T \cdot \sigma'(u) \cdot x$$

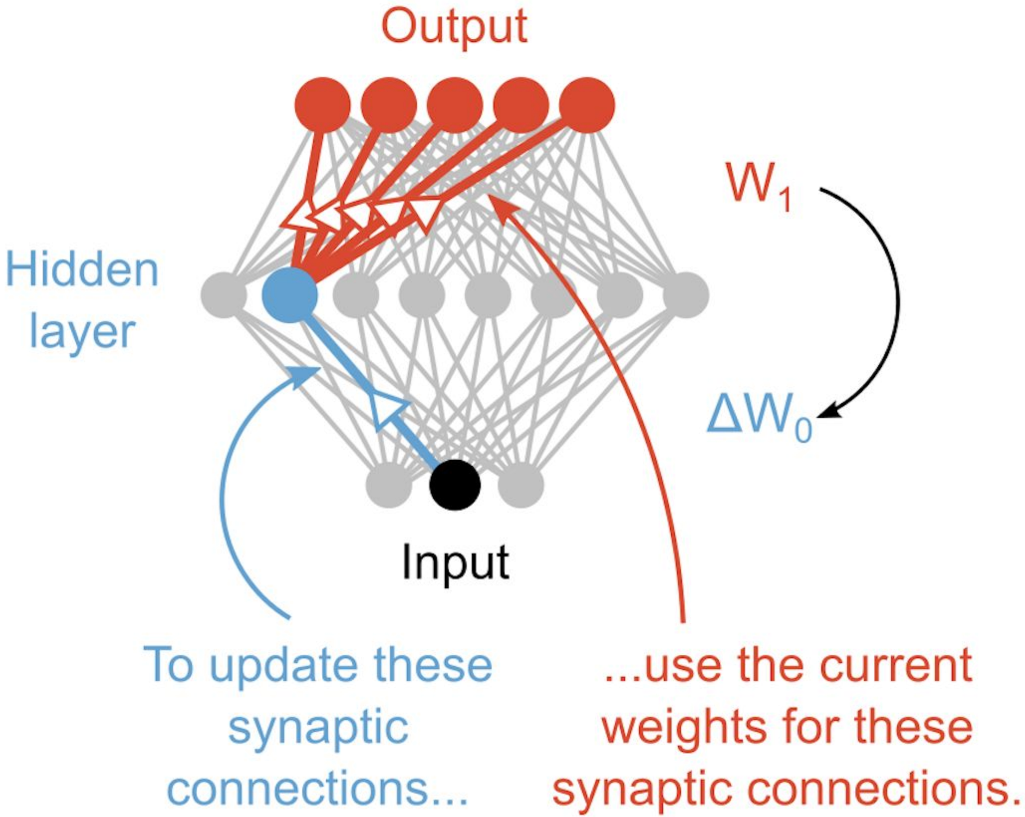


# Why Isn't Backprop "Biologically Plausible"?

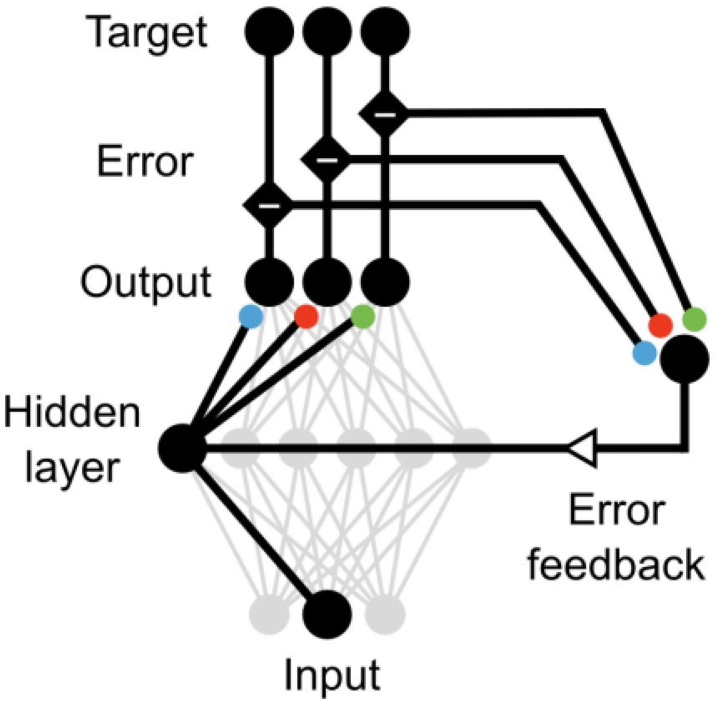
$$\Delta W_0 \propto e \cdot W_1^T \cdot \sigma'(u) \cdot x$$

We need:
(1) Error term
(2) Transpose of downstream weights
(3) Derivative of activation function
(4) Separate forward and backward passes

The backpropagation solution
(AKA "weight transport")

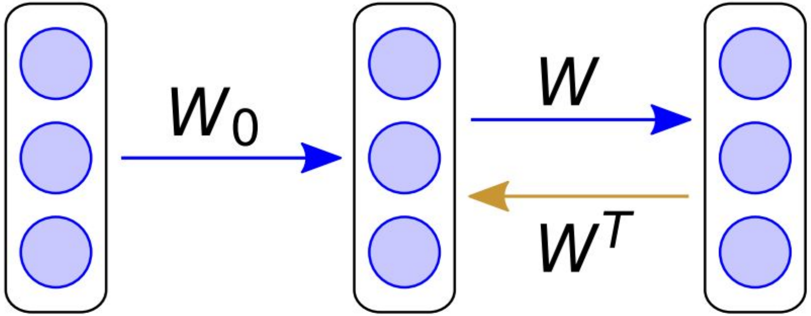


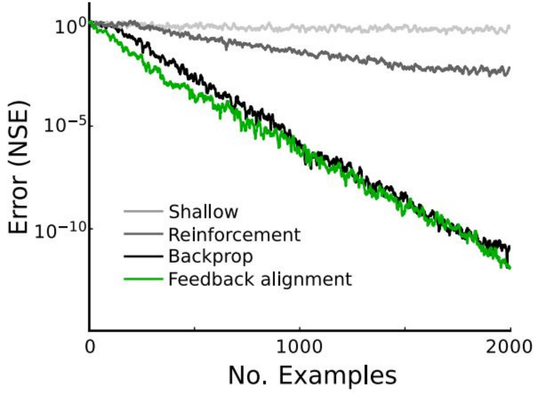
The most obvious solution to credit assignment is to explicitly calculate the partial derivative of your cost function with respect to your synaptic weights in the hidden layers (AKA backpropagation)

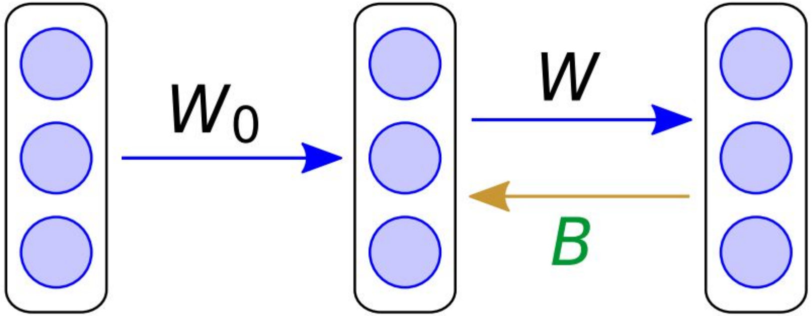


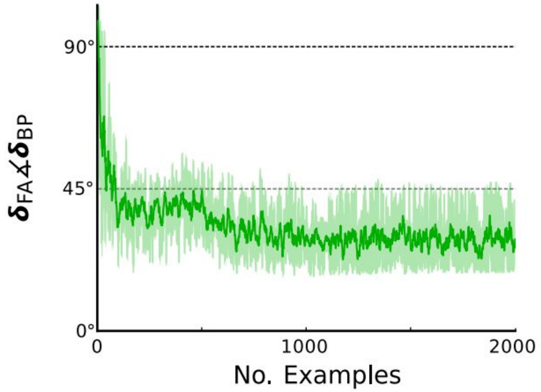

But, backpropagation requires a separate backward pass of error through symmetric feedback weights, which is not biologically plausible

# Feedback alignment

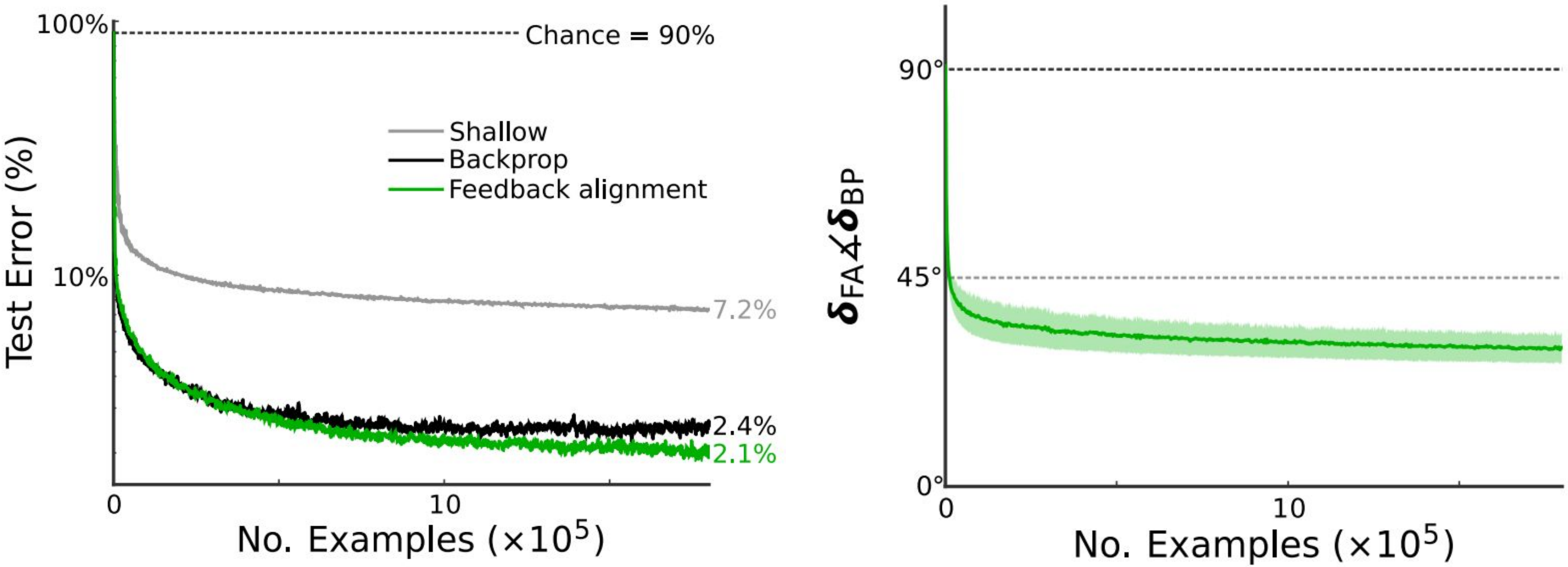


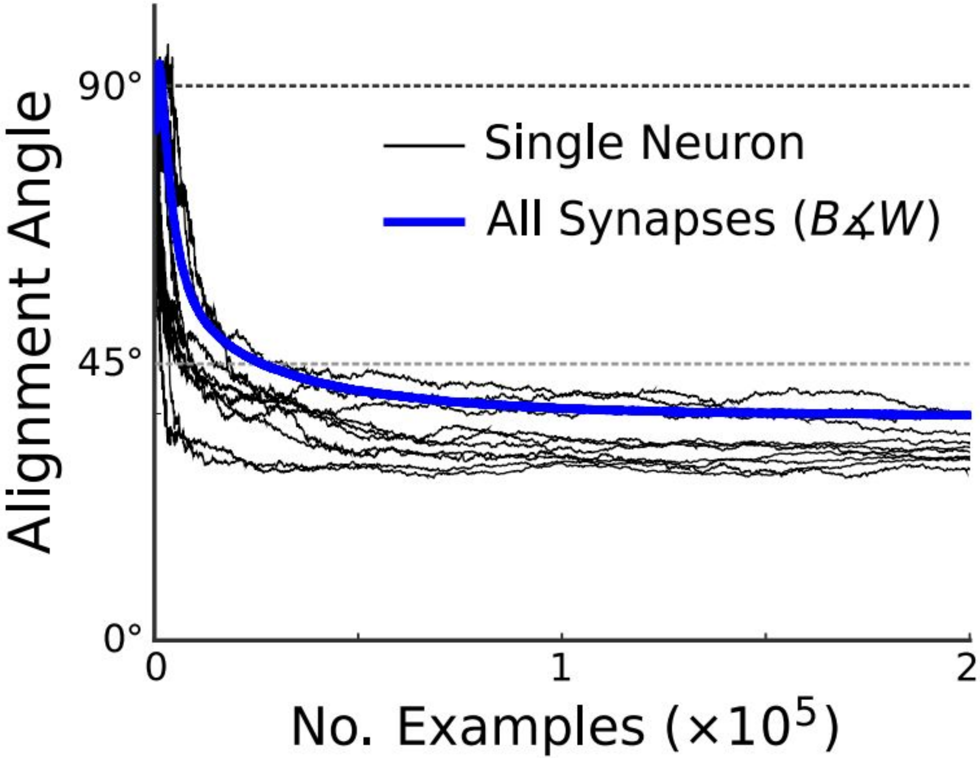



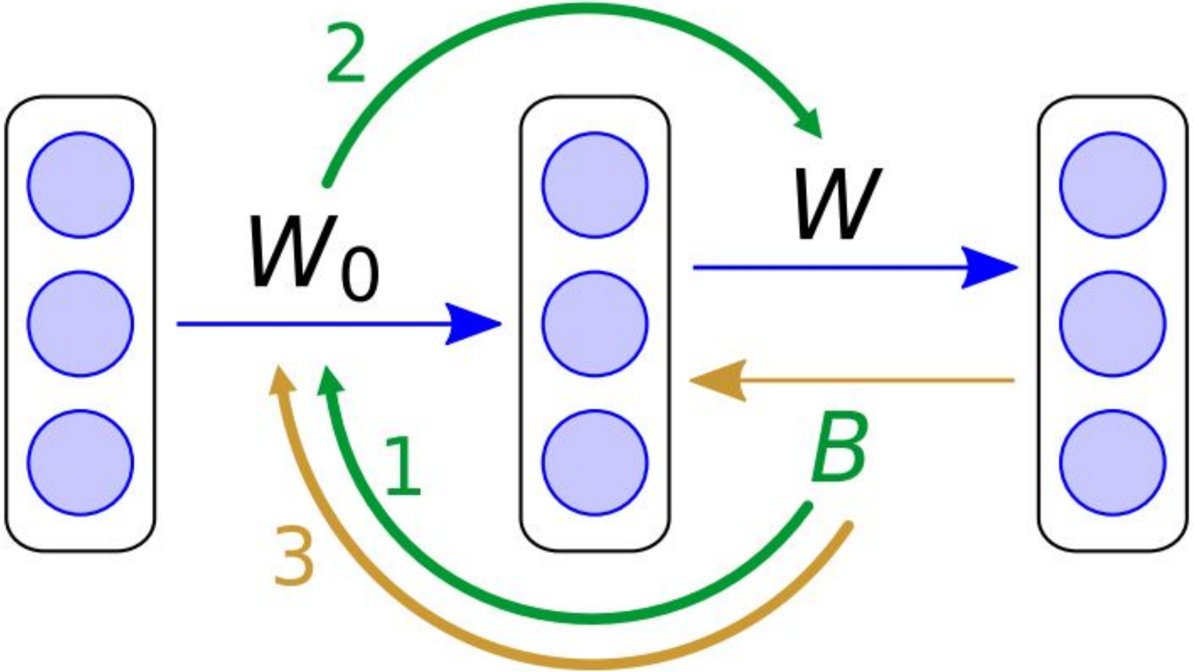

Lillicrap, Cownden, Tweed, Akerman, *Nat Comm*. 2016




Lillicrap, Cownden, Tweed, Akerman, *Nat Comm*. 2016

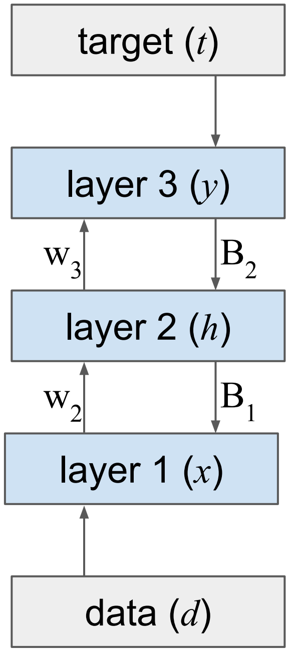



Lillicrap, Cownden, Tweed, Akerman, *Nat Comm*. 2016




Lillicrap, Cownden, Tweed, Akerman, *Nat Comm*. 2016

# Energy based models.




$$h_{t+1} = \lambda h_t + (1-\lambda)\sigma(W_i x_t + B_i y_t)$$

- ***x*** clamped to ***d***
- ***y*** clamped to ***t***

**Positive (+) Phase**

- ***x*** clamped to ***d***
- ***y*** free

**Negative (-) Phase**

Weight Update

$$\Delta W_{ij} = a_i^- (a_j^+ - a_j^-)$$

# Question:

Do existing biologically-motivated learning algorithms scale up to solve hard credit assignment problems?

# Constraints on learning rules.

- No weight transport (i.e. weight transposes).
- No weight tying (i.e. convolutional kernels)
- No feedback of signed errors.

- Use continuous (rather than spiking) signals.
- Violate Dale’s Law.
- Use separate forward/backward passes and local activation derivatives.
- Your personal complaint here...

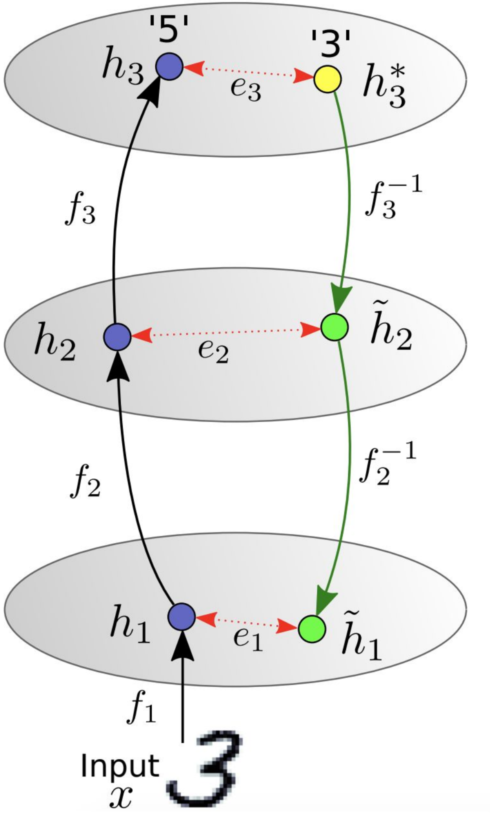
Bartunov, Santoro, Richards, Hinton, Lillicrap

# Target propagation:

- Assume perfect backward inverse functions.
- Send activity targets rather than errors.
- Compute per-layer errors locally.

$$E_\ell = \frac{1}{2}(\tilde{h}_\ell - h_\ell)^2$$

- Use a simple delta rule to compute forward weight updates.




Le Cun, 1987

# Difference target-propagation (DTP):

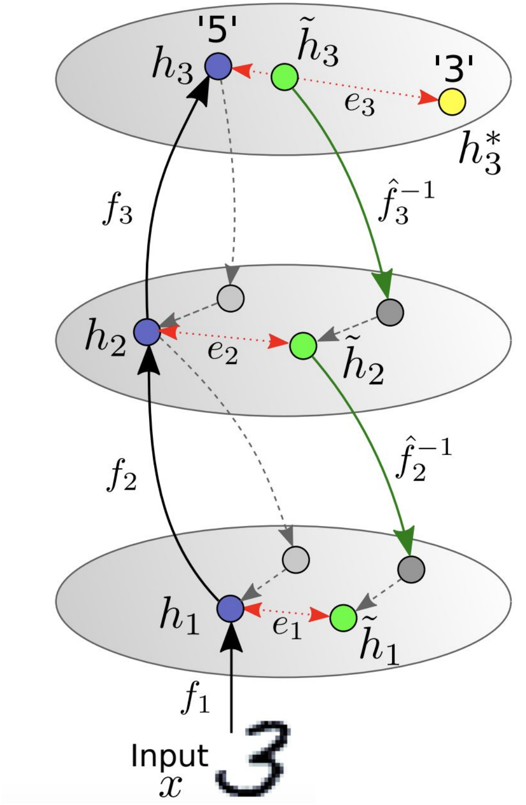
- Use autoencoder learned inverses.

- Form a correction term for targets.

- Both forward and backward weights are updated using simple and local delta rules.

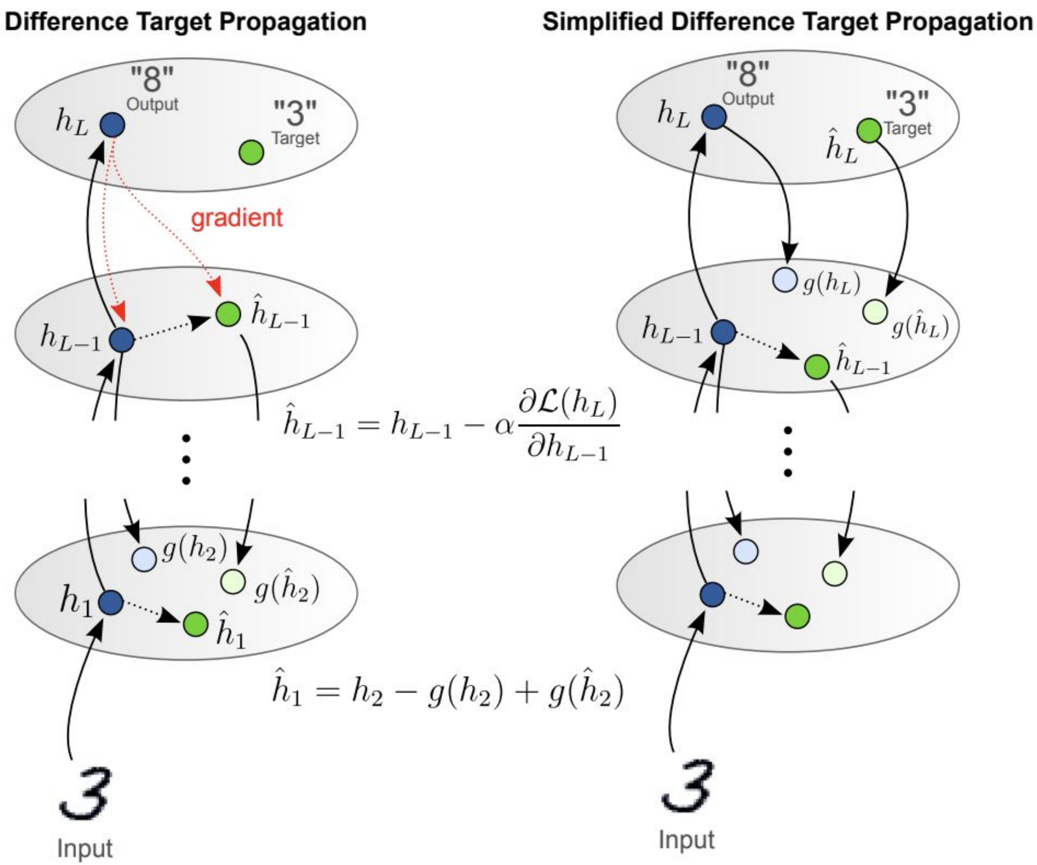
$$\Delta W_\ell = -\eta \frac{\partial E_\ell}{\partial W_\ell} \quad \Delta V_\ell = -\eta \frac{\partial E_\ell^B}{\partial V_\ell}$$

- But, DTP still uses gradients to compute targets at the penultimate hidden layer.




Lee et al, 2015

# Gradient free DTP variants:




- Use learned inverse to form targets for penultimate layer.
- Augment output layer with random features computed from penultimate layer activities.

Bartunov, Santoro, Richards, Hinton, Lillicrap

# Experimental details:

- Datasets: **MNIST**, **SVHN**, **CIFAR**, **ImageNet**.
- Explored **fully connected (FC)**, **locally connected (LC)**
- Examined relatively **small/simple networks**:
    - 3 locally connected layers followed by 1 fully connected for non-ImageNet datasets.
    - 7 locally connected layers for ImageNet.
    - GPU memory constrains maximum LC networks.

- Extensive hyperparameter search for parameters.
- Human guided architecture search for non-backprop algorithms.

# Results on MNIST, SVHN, CIFAR10

**MNIST:**

- **Replicated** original DTP results **>98%** accuracy.
- Weight transport free variant perform close to DTP (within 0.3% error).

**SVHN:**

- DTP and variant perform worse than BP (~**89%** versus ~**92%**)
- **Gap widens** in locally connected architectures (~**90%** versus ~**94%**)

**CIFAR10:**

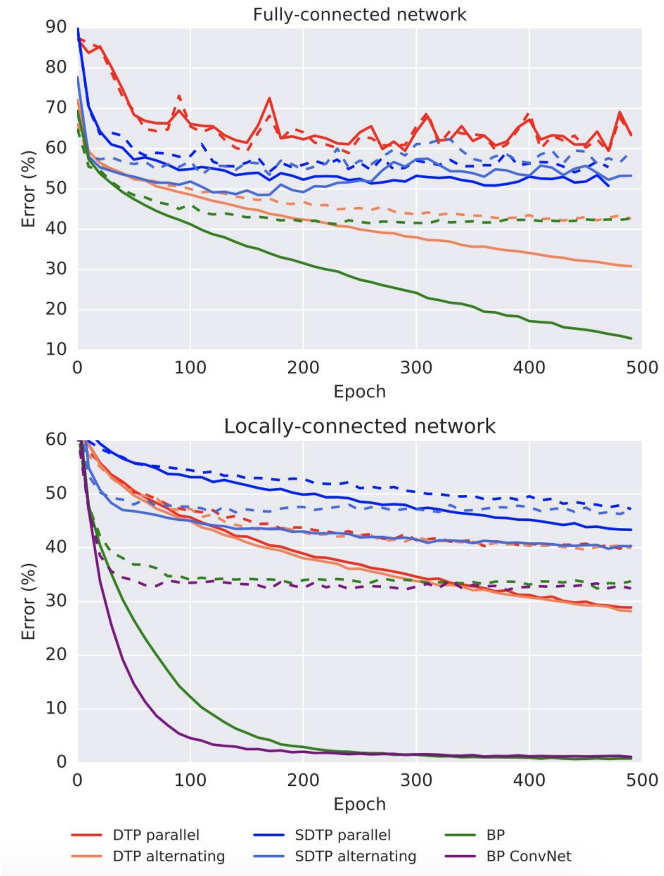
- DTP and variant perform close to BP (~**58%** versus ~**59%**).
- **Gap widens** in locally connected architectures (~**61%** versus ~**68%**)

Bartunov, Santoro, Richards, Hinton, Lillicrap

# Performance on CIFAR10

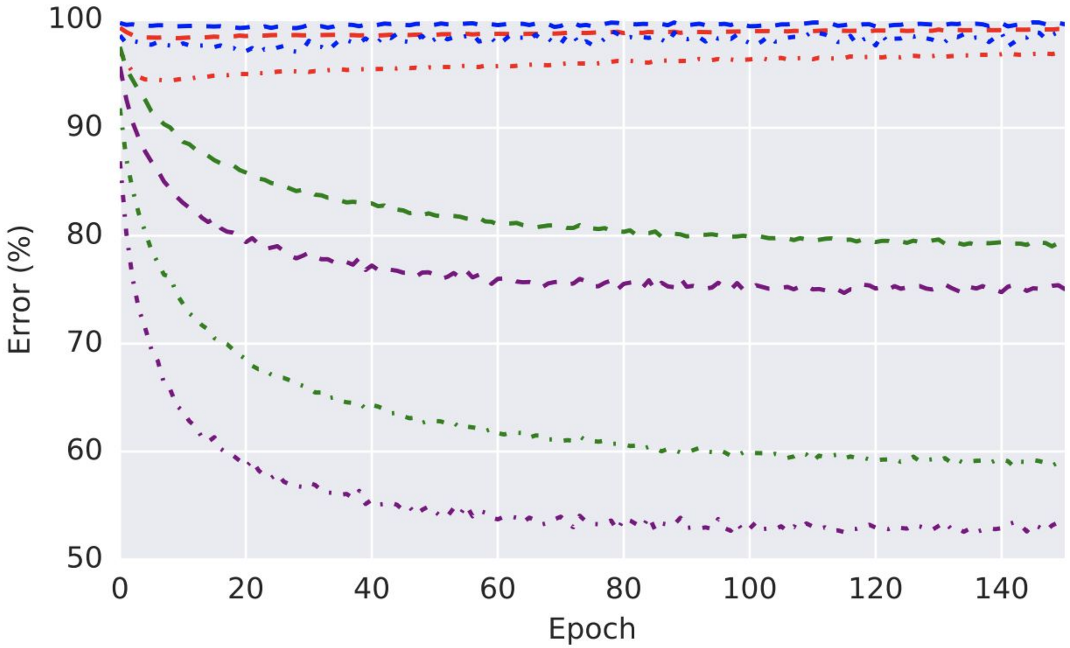
| Method | FC Train | FC Test | LC Train | LC Test |
|---|---|---|---|---|
| DTP, parallel | 59.45 | 59.14 | 28.69 | 39.47 |
| DTP, alternating | 30.41 | 42.32 | 28.54 | 39.47 |
| SDTP, parallel | 51.48 | 55.32 | 43.00 | 46.63 |
| SDTP, alternating | 48.65 | 54.27 | 40.40 | 45.66 |
| BP | 28.97 | 41.32 | 0.83 | 32.41 |
| BP ConvNet | – | – | 1.39 | 31.87 |

| Method | FC Train | FC Test | LC Train | LC Test |
|---|---|---|---|---|
| AO-SDTP parallel | 24.28 | 47.11 | 32.67 | 40.05 |
| AO-SDTP alternating | 0.00 | 45.40 | 34.11 | 40.21 |




Bartunov, Santoro, Richards, Hinton, Lillicrap

# Performance on ImageNet



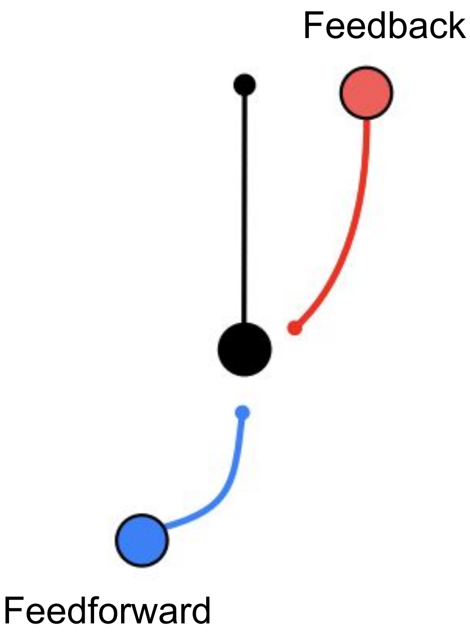

**DTP + Variants Top-5 >90%**

**Backprop Top-1 ~75%**

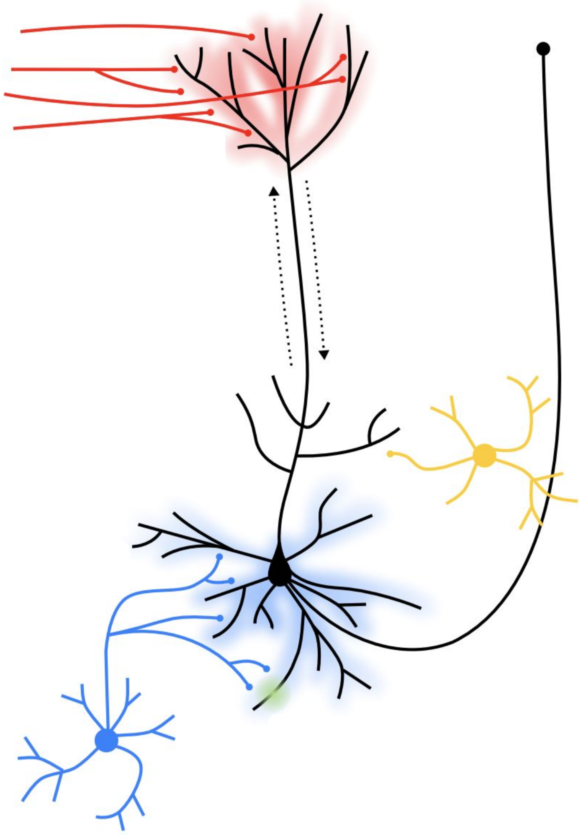
**Backprop Top-5 ~55%**

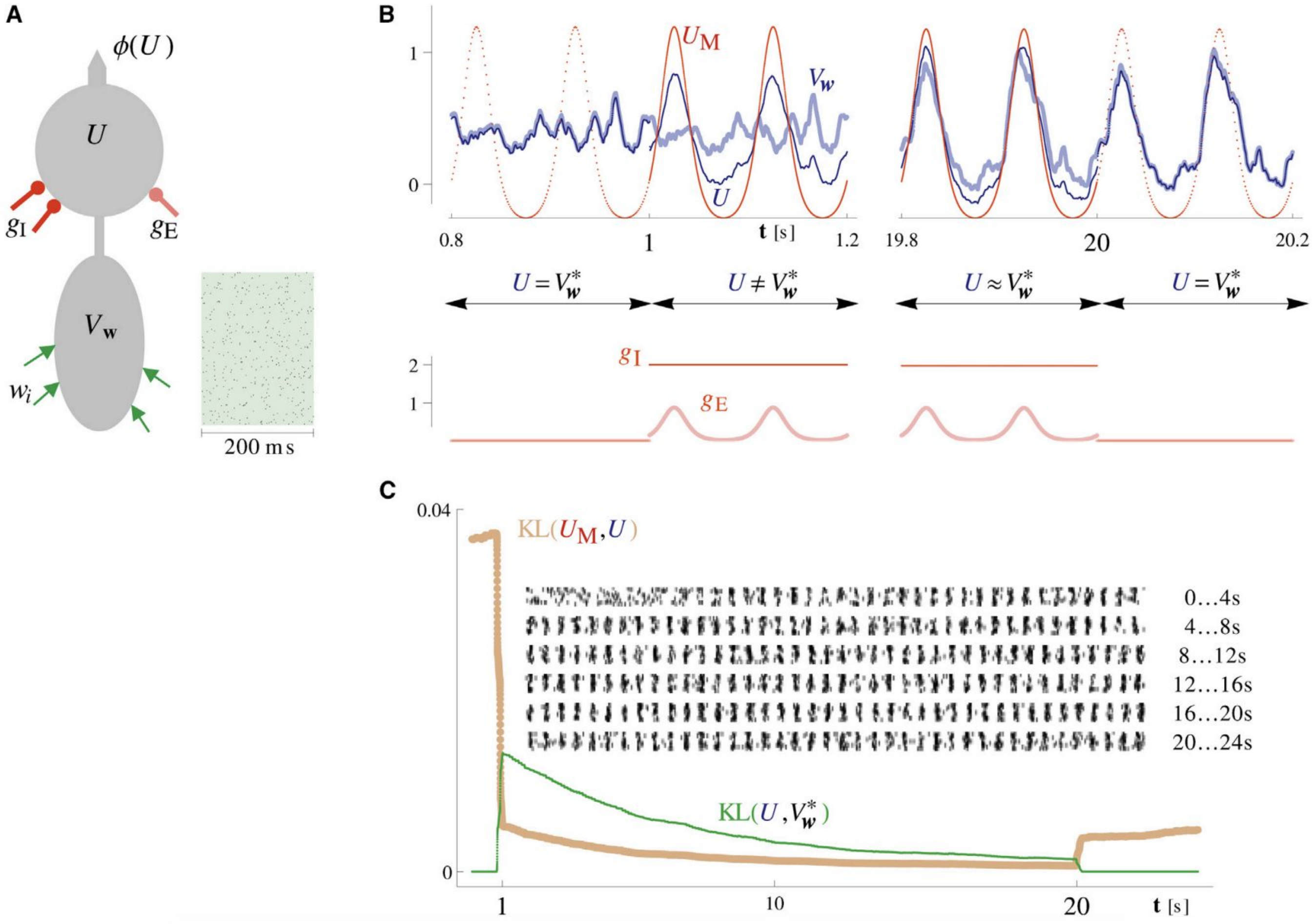
Bartunov, Santoro, Richards, Hinton, Lillicrap

# Models of a Neuron



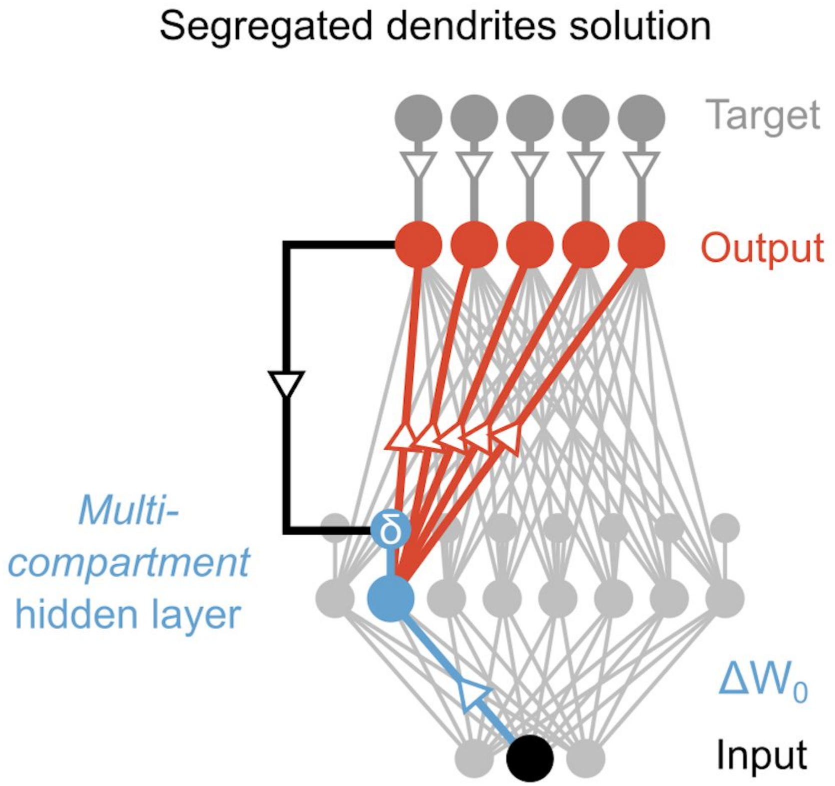
# New Models of a Neuron





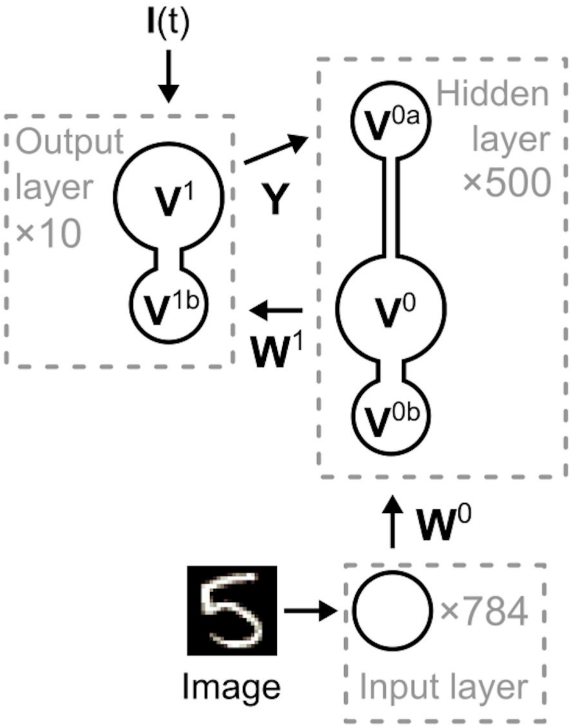

Urbanzick & Senn, *Neuron* 2014

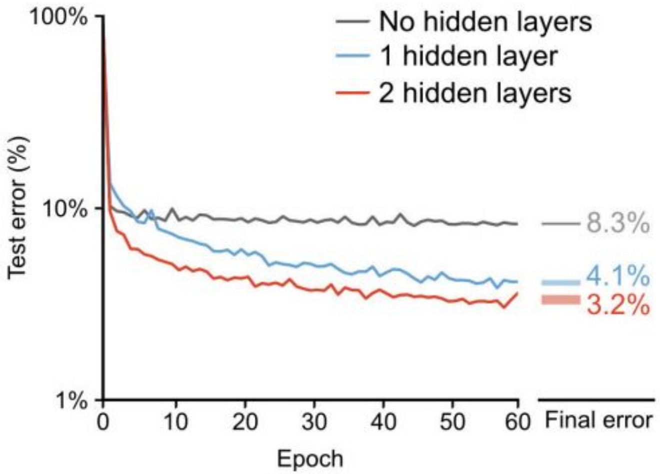
# More Plausible Deep Learning Algorithms?




Guerguiev, Lillicrap, Richards, *eLife* 2017

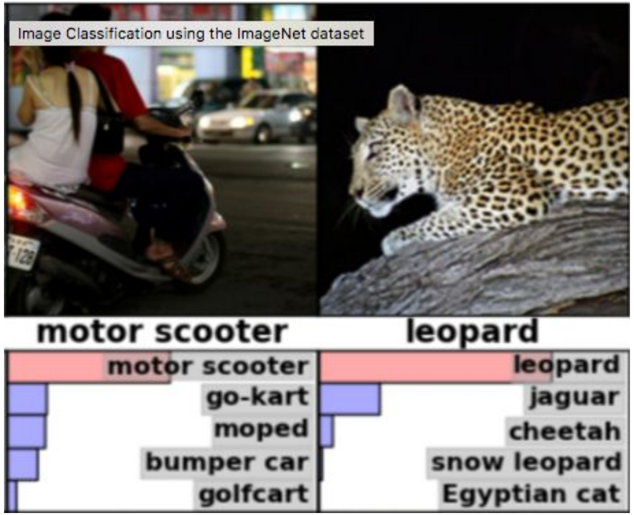
# More Plausible Deep Learning Algorithms?



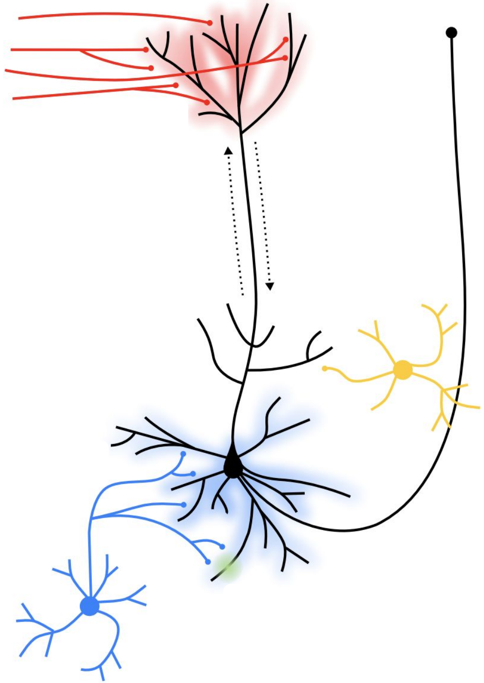




Guerguiev, Lillicrap, Richards, *eLife* 2017

# Future Directions:




New machine learning algorithms that are obey known constraints of biology but still perform well on, e.g. ImageNet



Neuroscience experiments aimed at explicitly identifying the role of feedback in weight updates.

# Questions?

**Work with:**

Jordan Guerguiev, Blake Richards,
Sergey Bartunov, Adam Santoro,
Luke Marris, Daniel Cownden, Douglas Tweed,
Colin Akerman, Geoffrey Hinton