Deep learning in the brain

Deep learning summer school
Montreal 2017

Blake Aaron Richards
Learning in Neural Circuits Laboratory
Department of Biological Sciences
University of Toronto Scarborough

LiNC LAB
linclab.org

UNIVERSITY OF TORONTO SCARBOROUGH

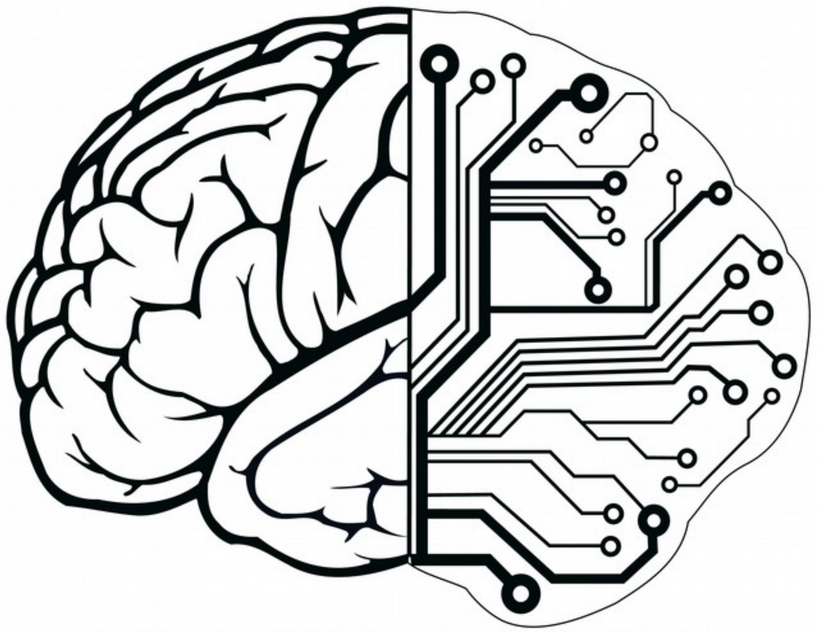
1. Why deep learning is not just for AI

UNIVERSITY OF TORONTO SCARBOROUGH

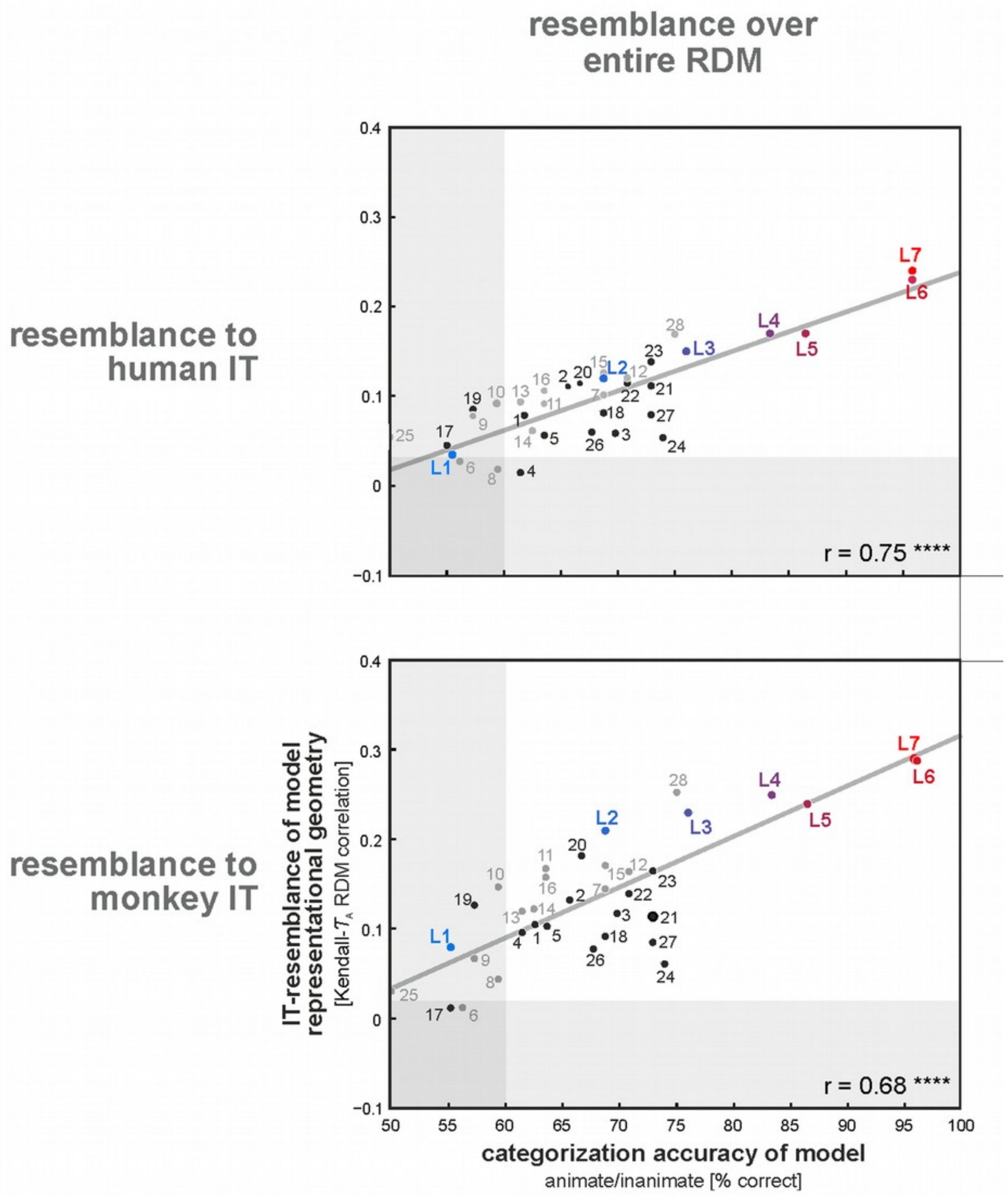
The recent success of deep learning in artificial intelligence (AI) means that most people associate it exclusively with AI

But, one of the goals of (some) deep learning research has always been to understand how our own brains work

In this session, I'm going to give you a brief overview on current research into how deep learning might be implemented in the real brain



1. Why deep learning is not just for AI



A little more motivation:

Deep learning models are a better fit to neocortical representations than models developed by neuroscientists!

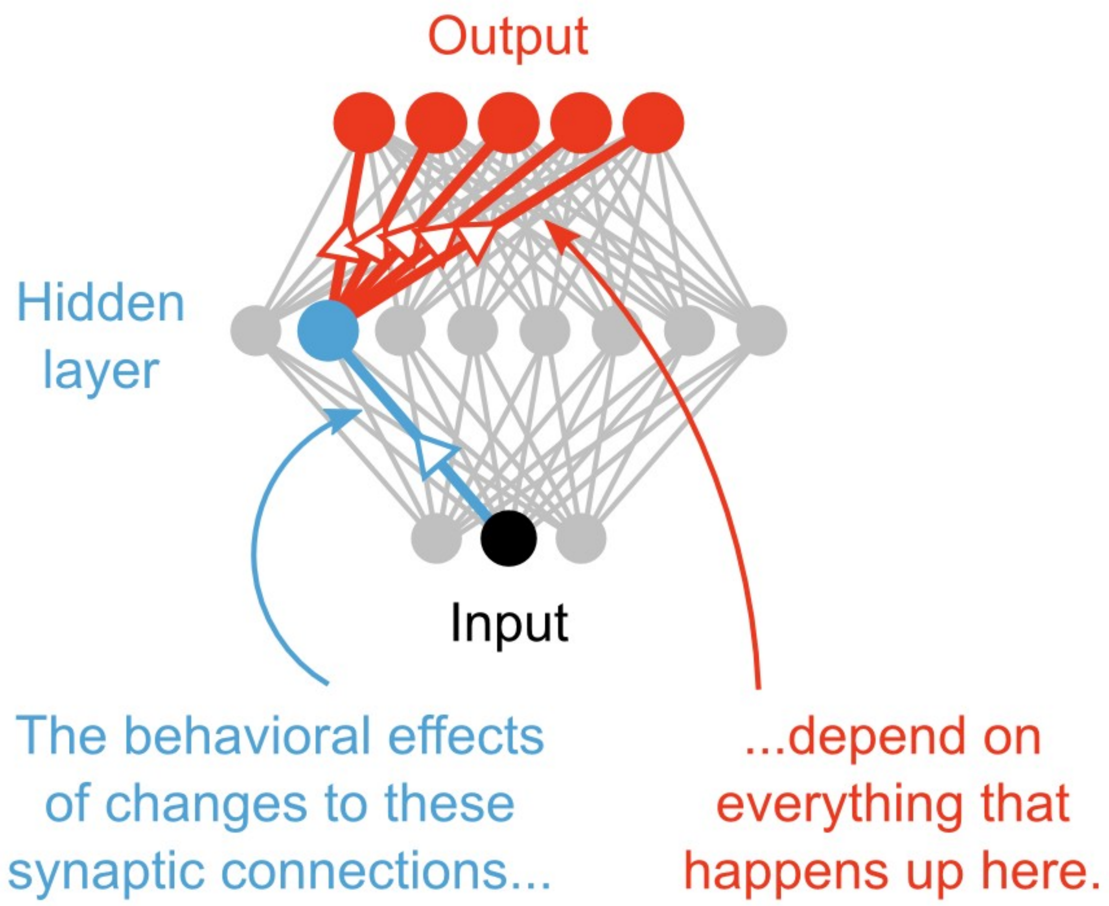
This would suggest that our neocortex is doing deep learning

Khaligh-Razavi and Kriegeskorte (2014)
PloS Comp. Biol. 10(11): e1003915

2. Foundations: what is required for deep learning?

The key feature of deep learning is the ability to improve learning by adding hidden layers

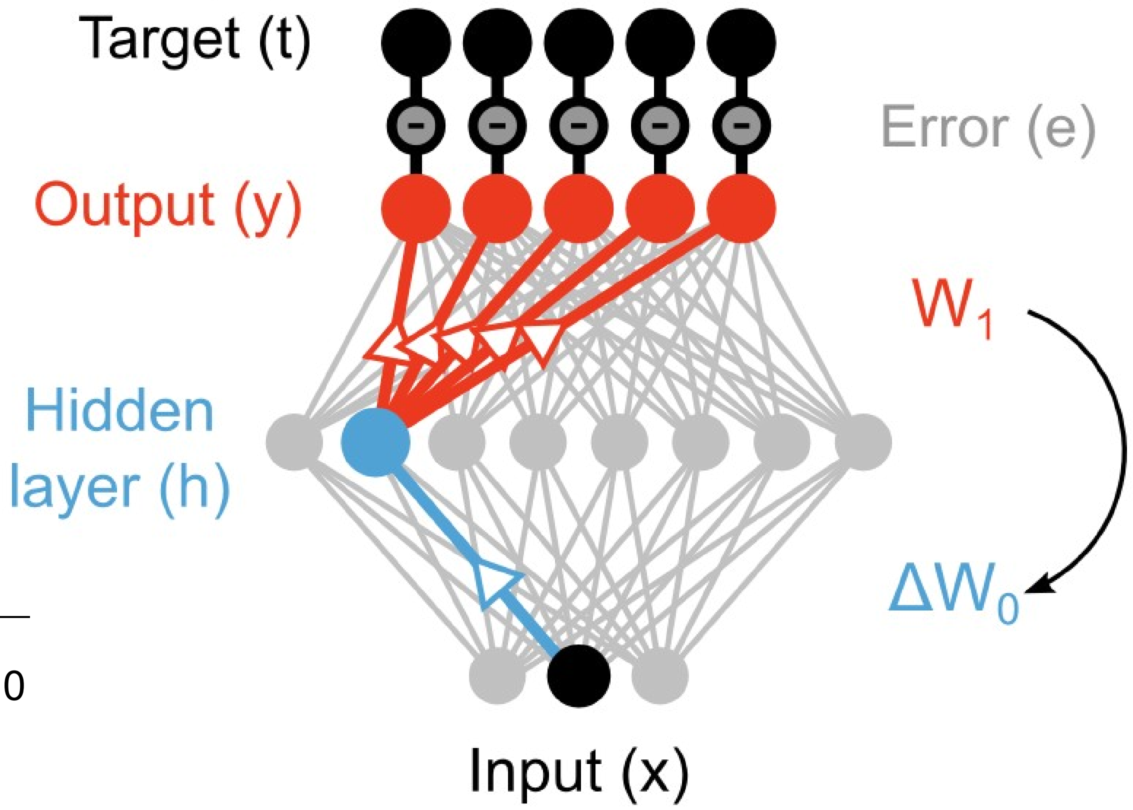
To do so, you must be able to assign "credit" (or "blame") to synapses in the hidden layers for their contribution to output of the network


Output
Hidden layer
Input
The behavioral effects of changes to these synaptic connections...
...depend on everything that happens up here.

3. The obvious solution: backpropagation

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)

$$u = W_0 x, v = W_1 h$$
$$h = \sigma(u), y = \sigma(v)$$
$$e = (y - t), L = \frac{1}{2} e^2$$
$$\Delta W_0 \propto \frac{\partial L}{\partial W_0}$$
$$= \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial h} \cdot \frac{\partial h}{\partial u} \cdot \frac{\partial u}{\partial W_0}$$
$$= e \cdot W_1^T \cdot \sigma'(u) \cdot x$$



3. The obvious solution: backpropagation

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)

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

Here's what we need to do backprop updates in the hidden layer

3. The obvious solution: backpropagation

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)

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

We need the error (difference between output generated by a forward pass and the target)

3. The obvious solution: backpropagation

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)

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

We need to multiply that error by the transpose of W_1

3. The obvious solution: backpropagation

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)

$$\Delta W_0 \propto e \cdot W_1^T \cdot \underset{\uparrow}{\boxed{\sigma'}}(u) \cdot x$$

We need the derivative of the hidden unit activation function

3. The obvious solution: backpropagation

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)

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

We need a forward pass without backwards flow of activity

4. Biological problems with vanilla backprop

UNIVERSITY OF TORONTO SCARBOROUGH

Unfortunately, all four of those things we need are biologically problematic...

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

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

4. Biological problems with vanilla backprop

Unfortunately, all four of those things we need are biologically problematic...

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

We need:

(1) Error term – no clear implementation in neocortex

(2) Transpose of downstream weights

(3) Derivative of activation function

(4) Separate forward and backward passes

4. Biological problems with vanilla backprop

Unfortunately, all four of those things we need are biologically problematic...

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

We need:
(1) Error term
(2) Transpose of downstream weights – neurons don't know this
(3) Derivative of activation function
(4) Separate forward and backward passes

4. Biological problems with vanilla backprop

Unfortunately, all four of those things we need are biologically problematic…

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

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

4. Biological problems with vanilla backprop

Unfortunately, all four of those things we need are biologically problematic…

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

We need:
(1) Error term
(2) Transpose of downstream weights
(3) Derivative of activation function
(4) Separate forward and backward passes – no evidence for it

4. Biological problems with vanilla backprop

Unfortunately, all four of those things we need are biologically problematic...

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

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

The last couple of years have seen much progress in addressing all four of these issues – I'm gonna bring you up-to-date and maybe just convince you that the brain might do backprop!

5. Issue one: error term

The brain can definitely calculate error terms (e.g. in the cerebellum), but there's no evidence that the neocortex has access to explicit error signals that it can use for, say, learning to speak, which it passes around the network

5. Issue one: error term

The brain can definitely calculate error terms (e.g. in the cerebellum), but there's no evidence that the neocortex has access to explicit error signals that it can use for, say, learning to speak, which it passes around the network

It would be more plausible if we could simply use external signals that push, or "nudge", the system towards the right answer (e.g. when you hear someone else speak correctly it just pushes you a bit towards the right way of speaking)

5. Issue one: error term

Scellier & Bengio (2017) proposed Equilibrium Propagation, which uses a “free phase” (with no external feedback) and a “weakly clamped phase” (where the external environment nudges the network towards the correct answer)

$u = \{x, h, y\}$ w/o sigmoid

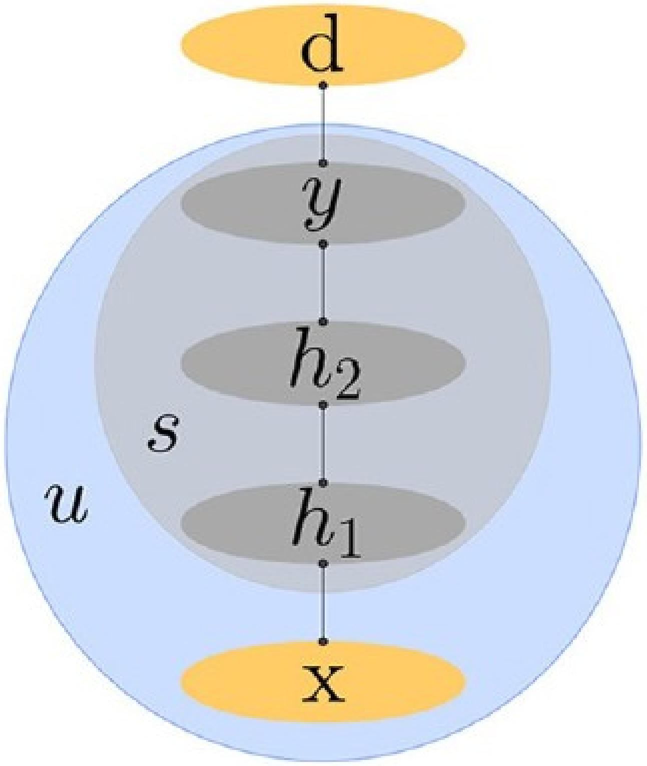
$\beta = 0$ free phase

$\beta > 0$ weakly clamped phase

If we assume symmetric weights between units then in the limit $\beta \rightarrow 0$

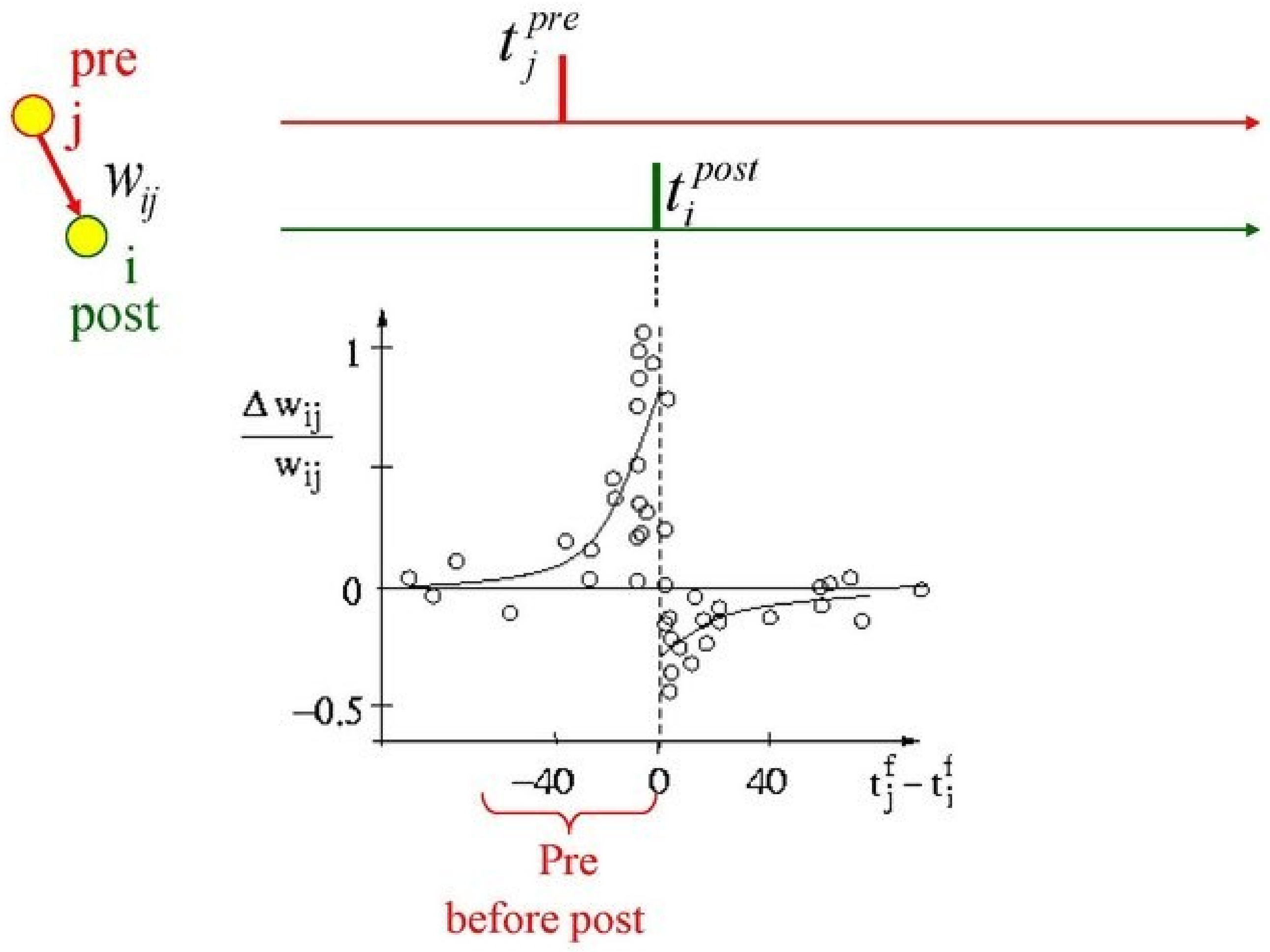
$$\Delta W_{ij} = \frac{1}{\beta}\left(\sigma(u_i^\beta)\sigma(u_j^\beta) - \sigma(u_i^0)\sigma(u_j^0)\right)$$

implements SGD on L = $1/2e^2$



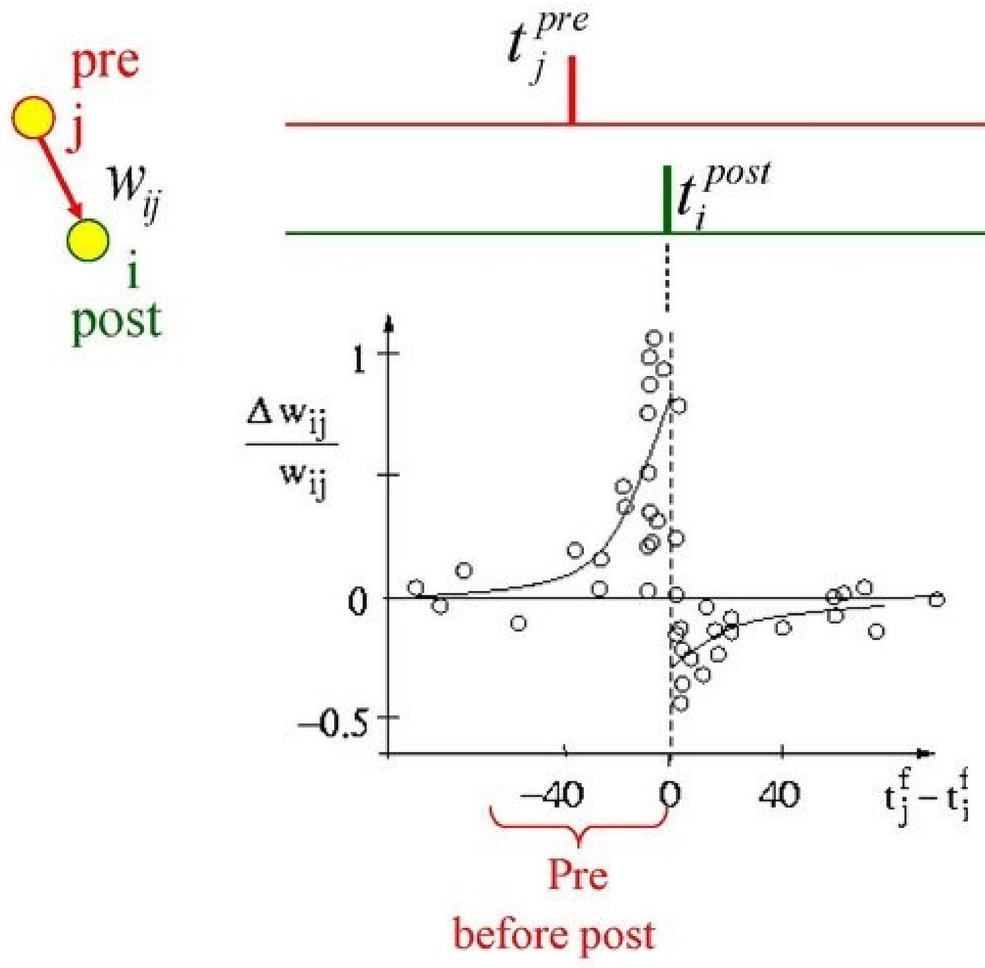
5. Issue one: error term

This update term is interesting because it predicts a classic experimental result known as **spike-timing-dependent plasticity (STDP)**



5. Issue one: error term

This update term is interesting because it predicts a classic experimental result known as **spike-timing-dependent plasticity (STDP)**



If this relationship holds:

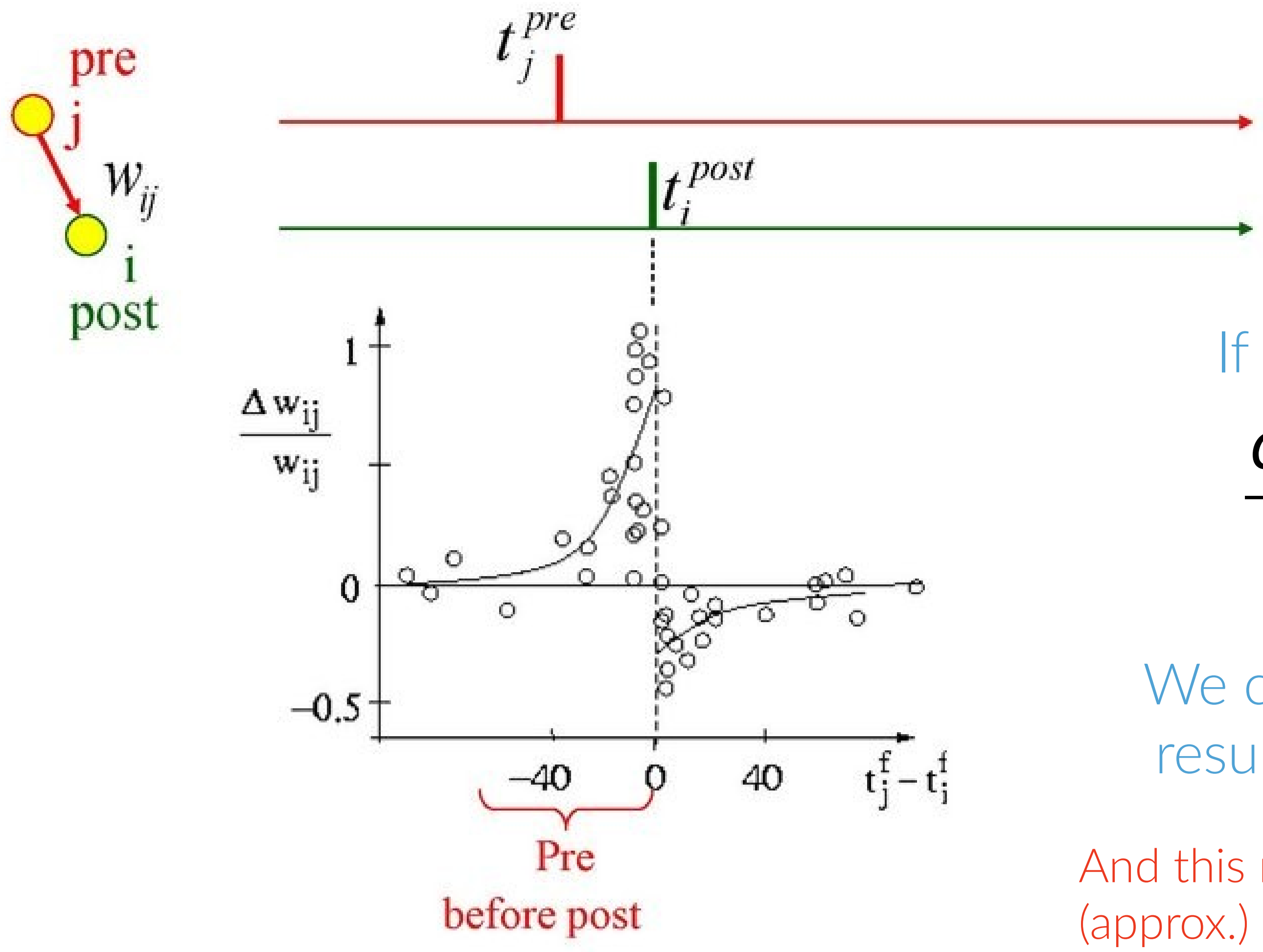
$$\frac{dW_{ij}}{dt} = \sigma(u_j)\frac{du_i}{dt}$$

We can get the same STDP result experimentalists see

Scellier & Bengio (2017), Front. Comp. Neurosci. 11(24)

5. Issue one: error term

This update term is interesting because it predicts a classic experimental result known as **spike-timing-dependent plasticity (STDP)**



If this relationship holds:

$$\frac{dW_{ij}}{dt} = \sigma(u_j)\frac{du_i}{dt}$$

We can get the same STDP result experimentalists see

And this relationship **does** hold (approx.) for equilibrium propagation

Scellier & Bengio (2017), Front. Comp. Neurosci. 11(24)

5. Issue one: error term

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

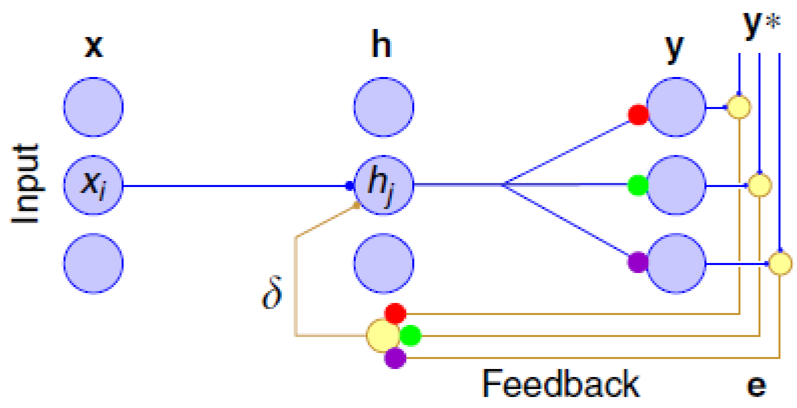
We need:
~~(1) Error term~~
(2) Transpose of downstream weights
(3) Derivative of activation function
(4) Separate forward and backward passes

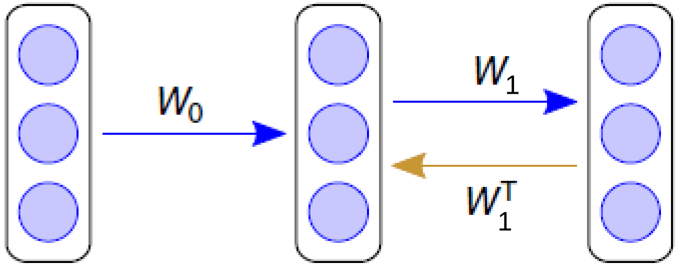
One item down:
We can do backprop without explicit error terms
(and it seems to match experimental data on STDP)

6. Issue two: downstream weights

As noted, the backprop update rule assumes that the hidden layer neurons have access to the error term multiplied by the transpose of the downstream weights, W_1

That's not a realistic proposal for the brain, and it has led many neuroscientists to dismiss backprop

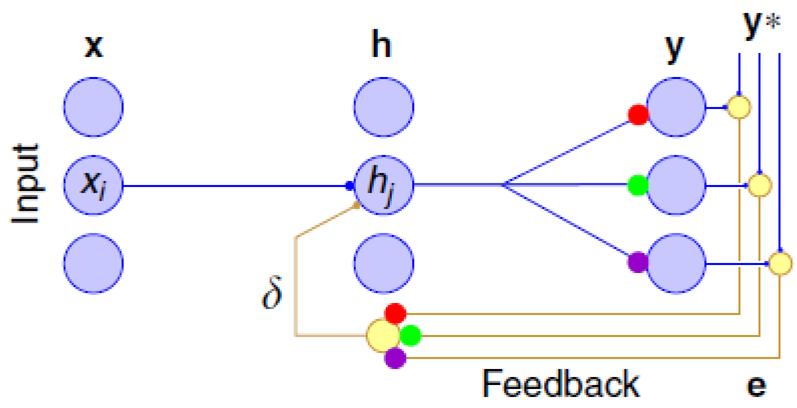


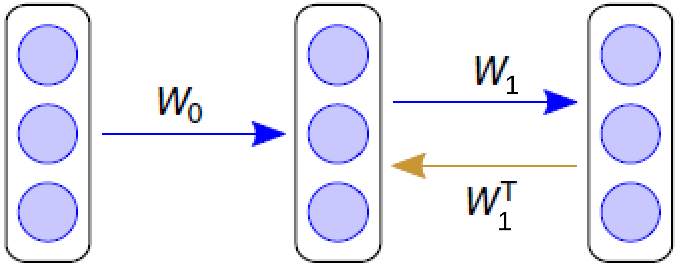


6. Issue two: downstream weights

As noted, the backprop update rule assumes that the hidden layer neurons have access to the error term multiplied by the transpose of the downstream weights, W_1

Timothy Lillicrap had an idea, maybe we could train the network to develop symmetric weights?



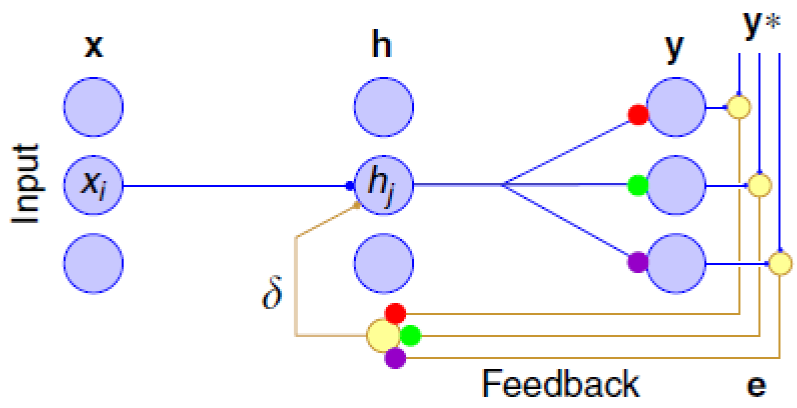


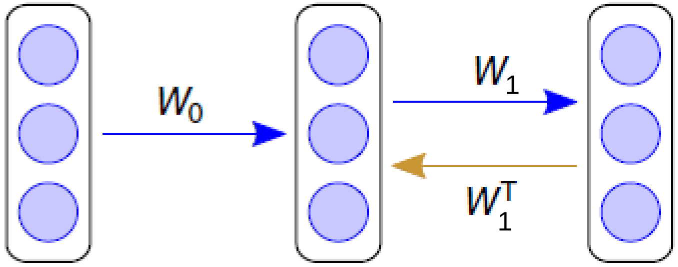
6. Issue two: downstream weights

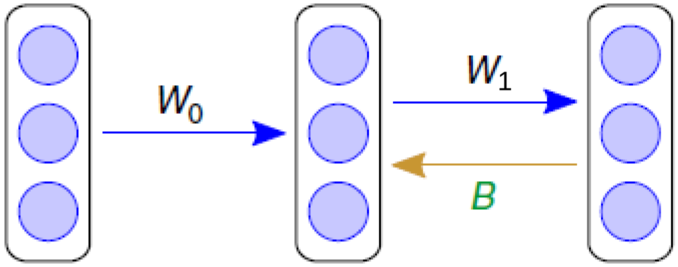
As noted, the backprop update rule assumes that the hidden layer neurons have access to the error term multiplied by the transpose of the downstream weights, W_1

Timothy Lillicrap had an idea, maybe we could train the network to develop symmetric weights?

To test his first attempt at such an algorithm, he used a control condition wherein the error was sent back through a random matrix B





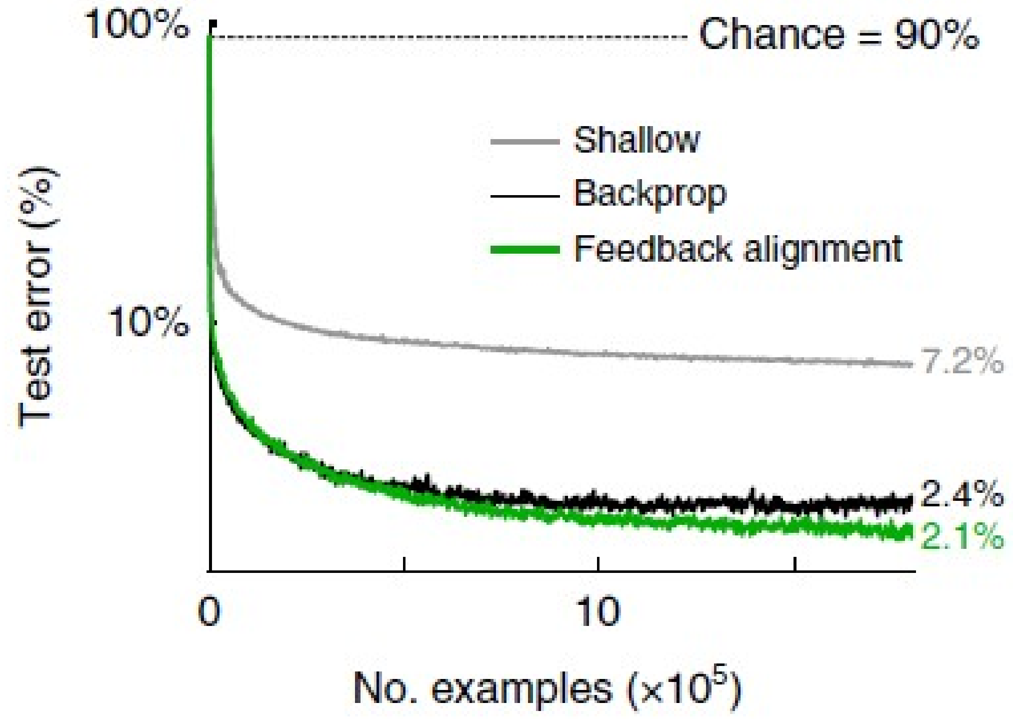


Lillicrap et al. (2016), Nat. Commi. 7(13276)

6. Issue two: downstream weights

Weirdly, the control network learned quite well!!!

Results on MNIST:



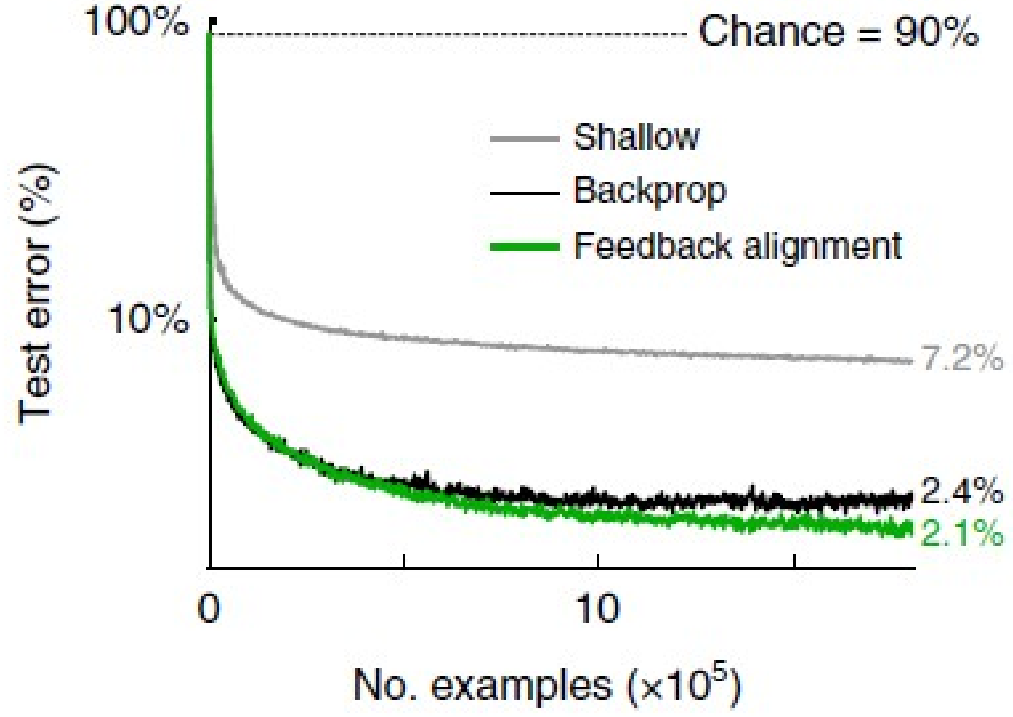
Lillicrap et al. (2016), Nat. Commi. 7(13276)

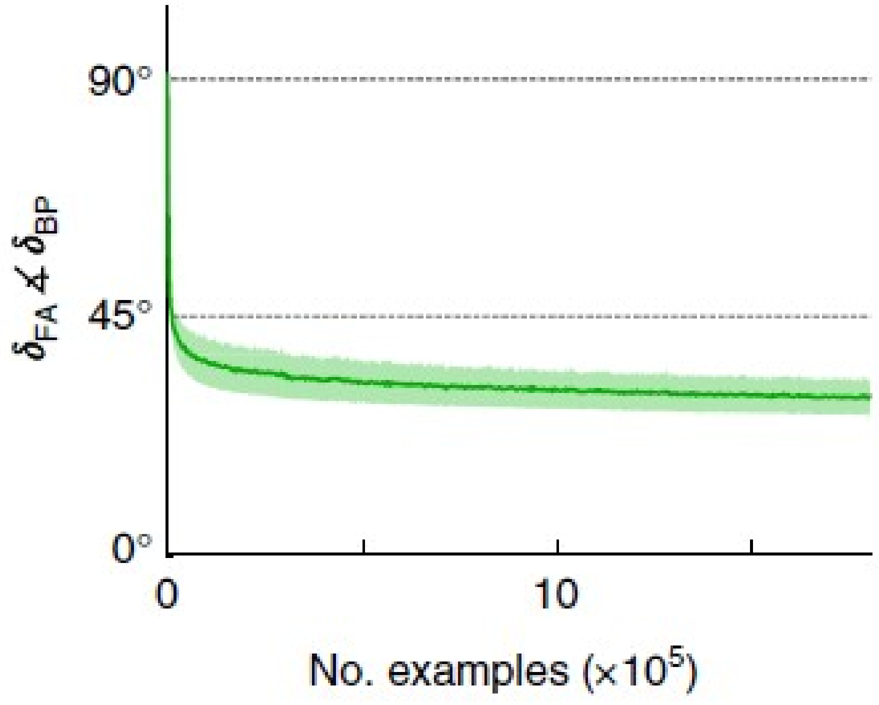
6. Issue two: downstream weights

Weirdly, the control network learned quite well!!!

Results on MNIST:

The reason was that the forward weights “aligned” themselves with the backwards weights





Lillicrap et al. (2016), Nat. Commi. 7(13276)

6. Issue two: downstream weights

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

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

Two items down:
We can do backprop without the transpose of downstream weights

7. Issue three: derivatives of spikes

Neurons don't communicate with analog signals. They use action potentials or "spikes" which are all-or-none events.

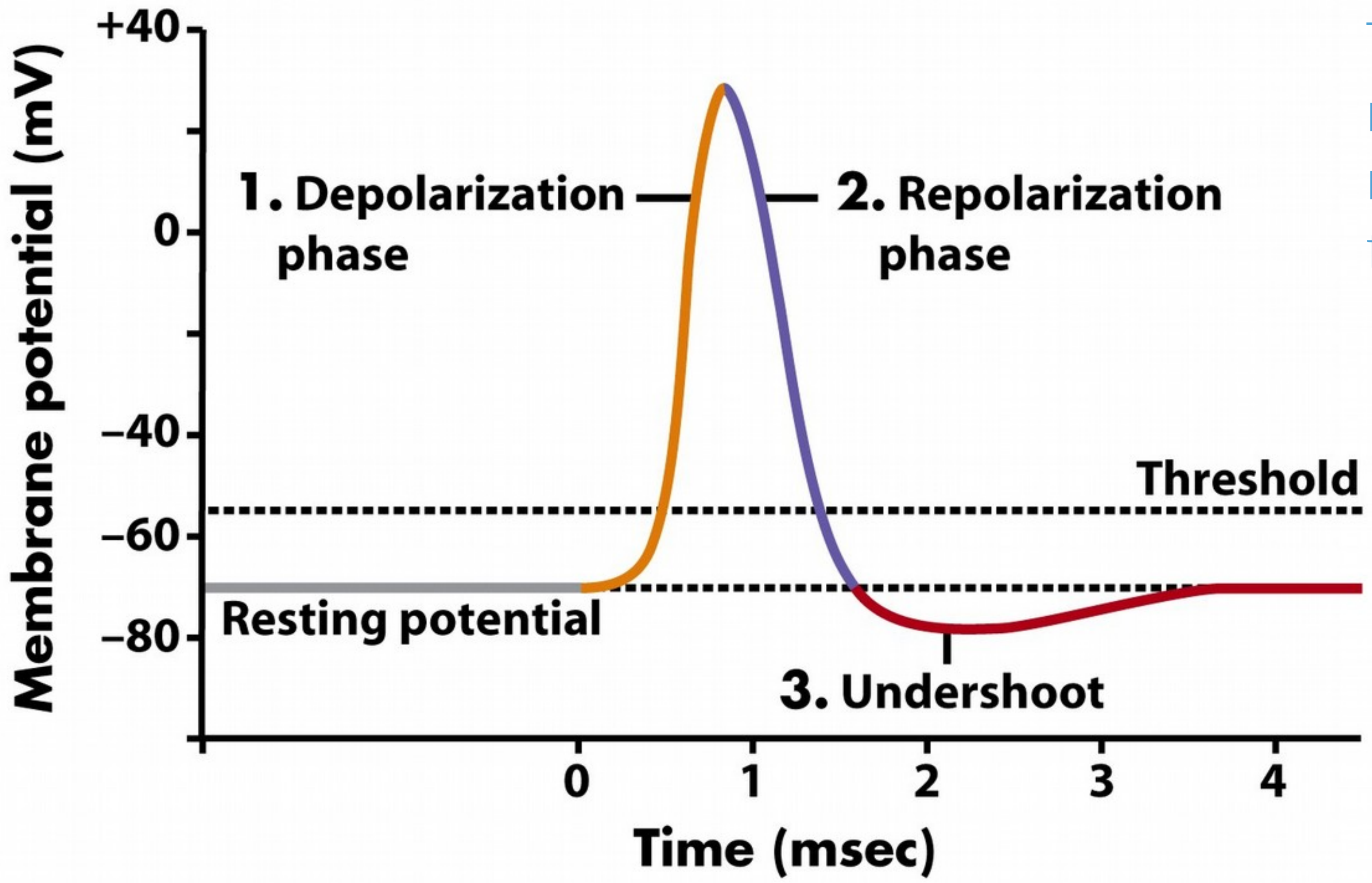


Figure 45-5 Biological Science, 2/e
© 2005 Pearson Prentice Hall, Inc.

This usually gets represented with a non-differentiable δ function:

$$S_i(t) = \sum_k \delta(t - t_k)$$

$$\delta(x) = \begin{cases} 0, \text{ if } x \neq 0 \\ \frac{1}{dt}, \text{ o/w} \end{cases}$$

7. Issue three: derivatives of spikes

Neurons don't communicate with analog signals. They use action potentials or “spikes” which are all-or-none events.

But, remember, we need to take the derivative of the activation function, which is supposed to represent the spiking activity…

This usually gets represented with a non-differentiable δ function:

$$S_i(t) = \sum_k \delta(t - t_k)$$

$$\delta(x) = \begin{cases} 0, \text{ if } x \neq 0 \\ \frac{1}{dt}, \text{ o/w} \end{cases}$$

We could treat the activation function as the spike rate (which is the typical interpretation), but that's problematic, since there's good evidence that the specific timing of spikes can carry a fair bit of information

7. Issue three: derivatives of spikes

Zenke & Ganguli (2017) approach this by first modifying the loss function to minimize the van Rossum distance between a target spike train and the actual spike train:

$$L = \frac{1}{2}\int_{-\infty}^{t} ds\left[\left(\alpha * \hat{S}_i - \alpha * S_i\right)(s)\right]^2$$

Where $\hat{S}_i$ is the target spike train, and **α** is a temporal convolution kernel

Taking the gradient, we get:

$$\frac{\partial L}{\partial W_{ij}} = -\int_{-\infty}^{t} ds\left[\left(\alpha * \hat{S}_i - \alpha * S_i\right)(s)\right]\left(\alpha * \frac{\partial S_i}{\partial W_{ij}}\right)(s)$$

Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

Zenke & Ganguli (2017) approach this by first modifying the loss function to minimize the van Rossum distance between a target spike train and the actual spike train:

$$L = \frac{1}{2} \int_{-\infty}^{t} ds \left[\left(\alpha * \hat{S}_i - \alpha * S_i \right)(s) \right]^2$$

Where $\hat{S}_i$ is the target spike train, and **α** is a temporal convolution kernel

Taking the gradient, we get:

Ah, but here's this bugger...

$$\frac{\partial L}{\partial W_{ij}} = - \int_{-\infty}^{t} ds \left[\left(\alpha * \hat{S}_i - \alpha * S_i \right)(s) \right] \left(\alpha * \boxed{\frac{\partial S_i}{\partial W_{ij}}} \right)(s)$$

Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

Zenke & Ganguli (2017) deal with the spike train derivative by replacing the spike train, S_i, in the gradient equation with an auxilliary function of the membrane potential, $\sigma(U_i(t))$, where:

$$U_i(t) \approx \sum_j W_{ij}(\epsilon * S_j(t))$$

Where ε is the shape of the postsynaptic response to a spike.

Our gradient is now:

$$\frac{\partial L}{\partial W_{ij}} = -\int_{-\infty}^{t} ds [(\alpha * \hat{S}_i - \alpha * S_i)(s)] \alpha(\sigma'(U_i(s)))(\epsilon * Sj)(s)$$

Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

Zenke & Ganguli (2017) deal with the spike train derivative by replacing the spike train, S_i, in the gradient equation with an auxilliary function of the membrane potential, $\sigma(U_i(t))$, where:

$$U_i(t) \approx \sum_j W_{ij}(\epsilon * S_j(t))$$

Where ε is the shape of the postsynaptic response to a spike.

Our gradient is now:

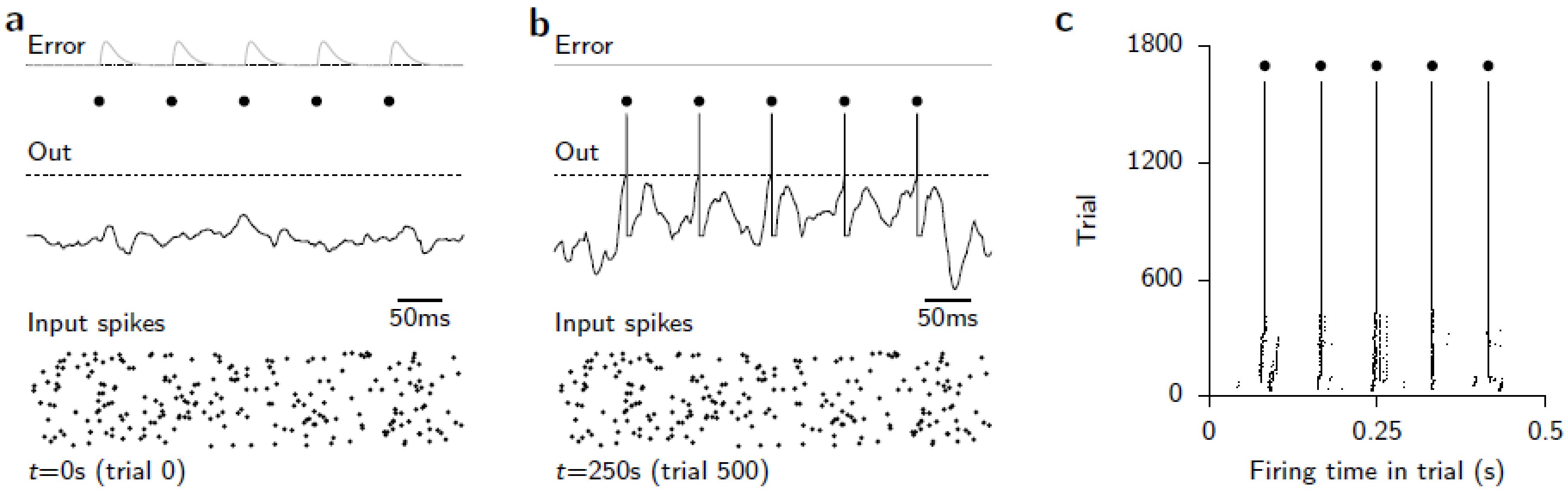
$$\frac{\partial L}{\partial W_{ij}} = -\int_{-\infty}^{t} ds \left[\underbrace{(\alpha * \hat{S}_i - \alpha * S_i)(s)}_{\text{Error term}} \right] \underbrace{\alpha(\sigma'(U_i(s)))(\epsilon * Sj)(s)}_{\text{Eligibility trace}}$$

Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

UNIVERSITY OF TORONTO SCARBOROUGH

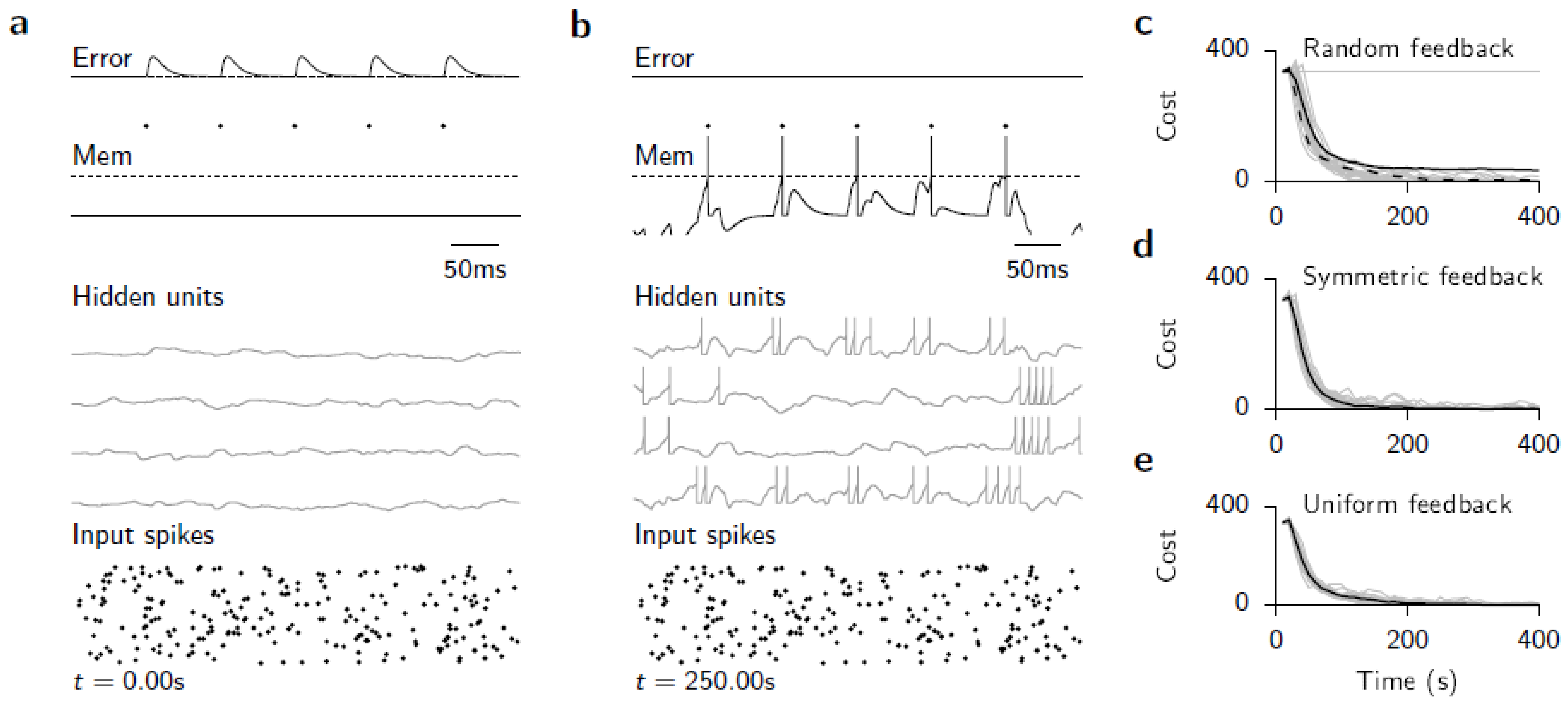
The network can now be trained to generate specific spike sequences:



Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

Training in networks with hidden layers is a straightforward extension:



Zenke & Ganguli (2017) arXiv: 1705.11146v1

7. Issue three: derivatives of spikes

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

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

Three items down:
We can do backprop with precise spike trains

8. Issue four: forward/backward passes

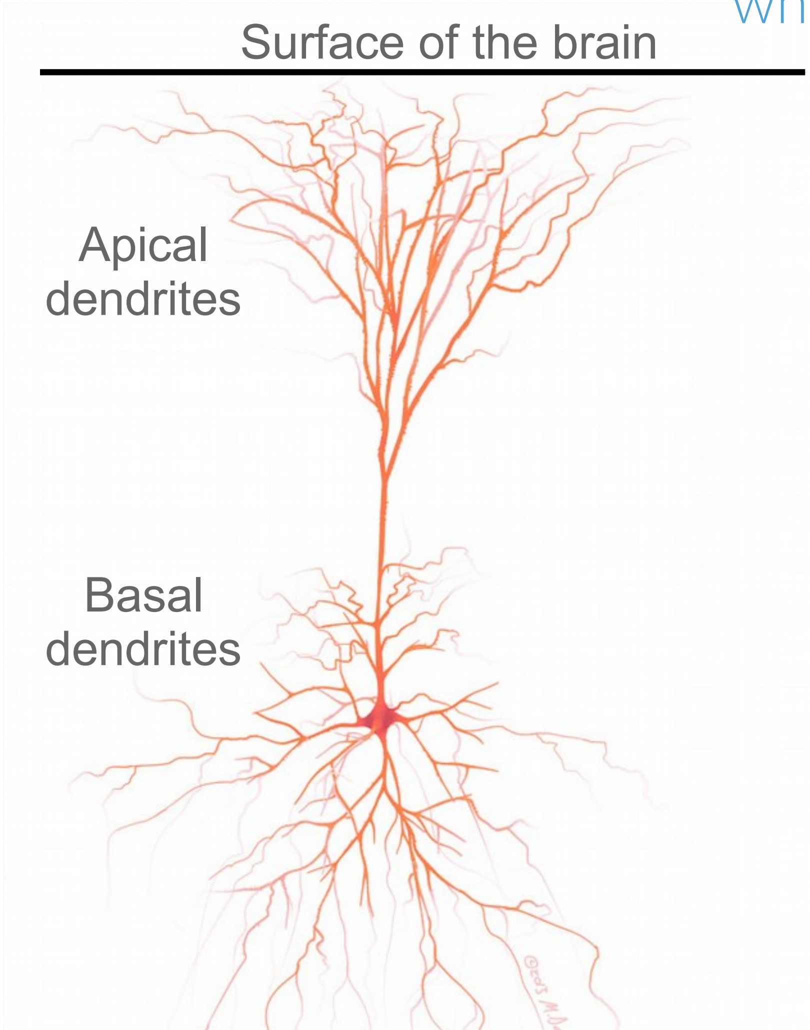
Our brains are constantly active (don't listen to the media), and there are massive backwards projections everywhere you look

At face value that would suggest that there probably isn't a forward pass followed by a backward pass...

8. Issue four: forward/backward passes

However, real neurons in the neocortex are far more complicated than the linear-non-linear points we typically use in neural networks

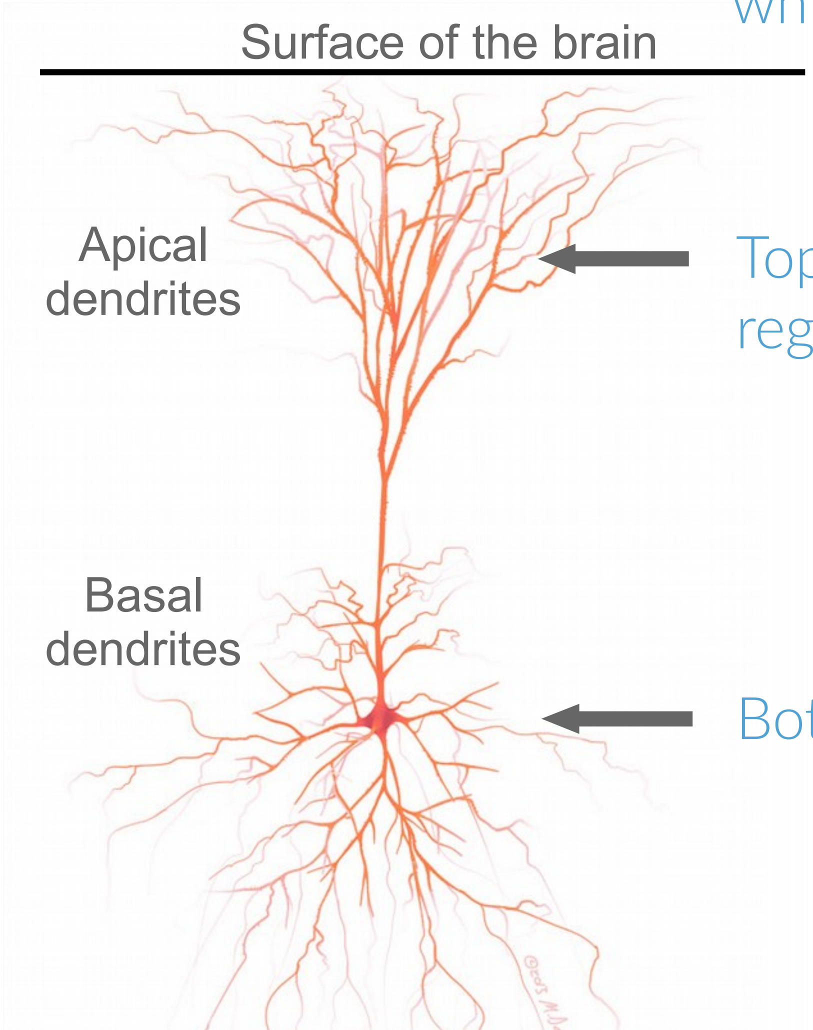
The majority of neurons in the neocortex are pyramidal neurons, which are shaped kind of like a big tree



8. Issue four: forward/backward passes

However, real neurons in the neocortex are far more complicated than the linear-non-linear points we typically use in neural networks

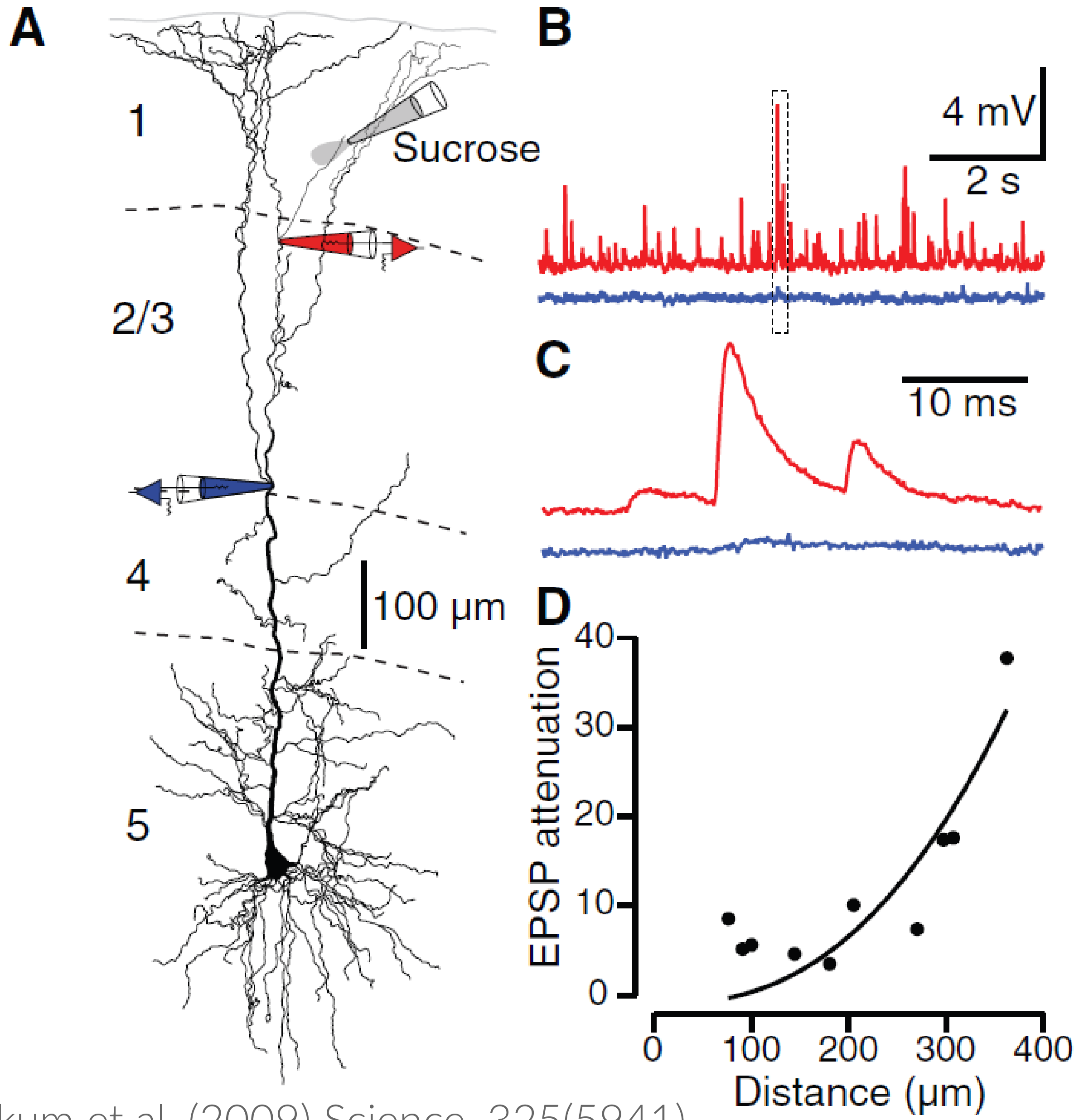
The majority of neurons in the neocortex are pyramidal neurons, which are shaped kind of like a big tree



Top-down connections from higher-order regions of the brain

Bottom-up sensory connections

8. Issue four: forward/backward passes

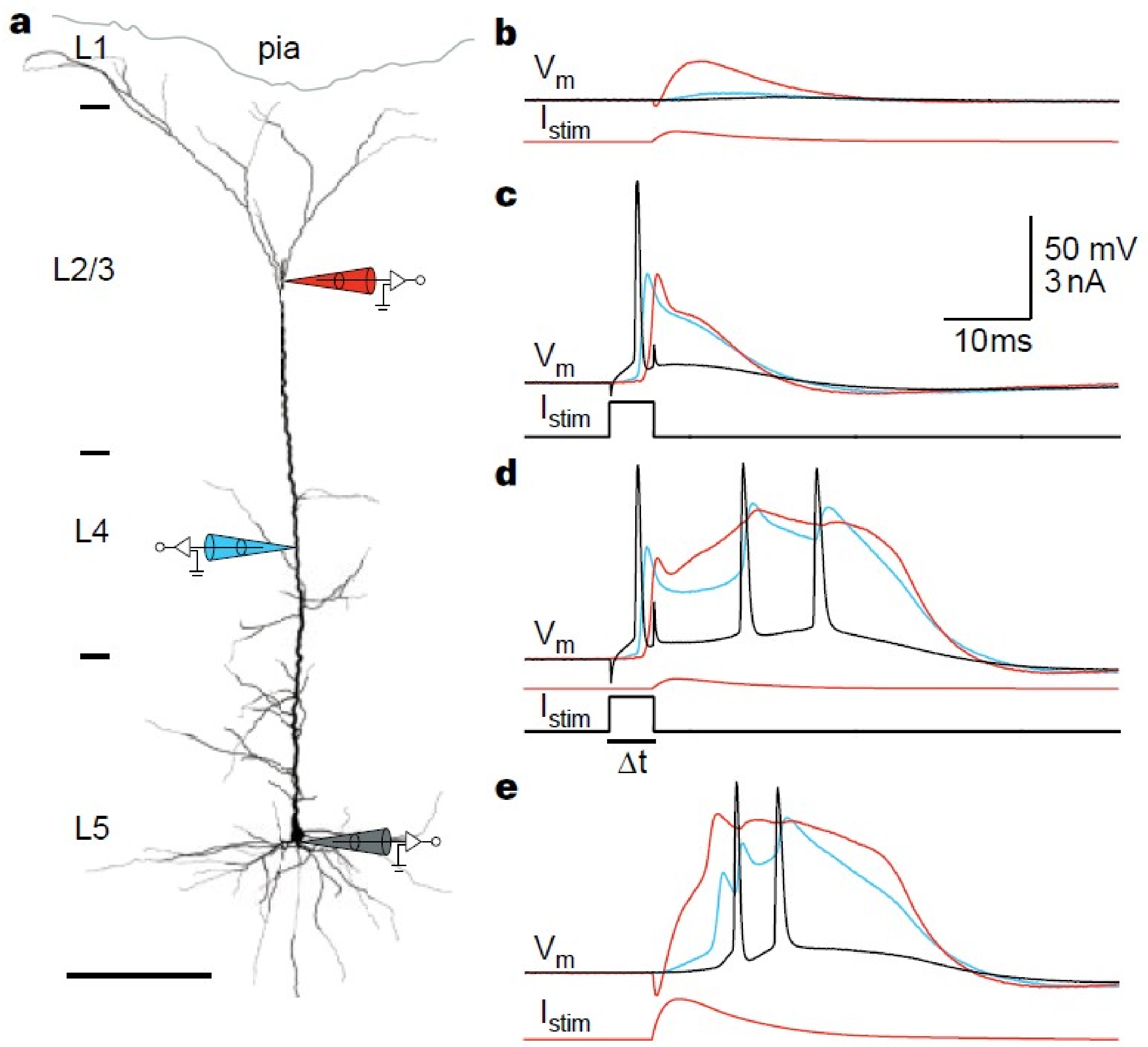


Larkum et al. (2009) Science, 325(5941)

But the apical dendrites are very distant from the axon hillock (where spikes are generated)

Most of the time, they barely drive activity in the cell at all

8. Issue four: forward/backward passes

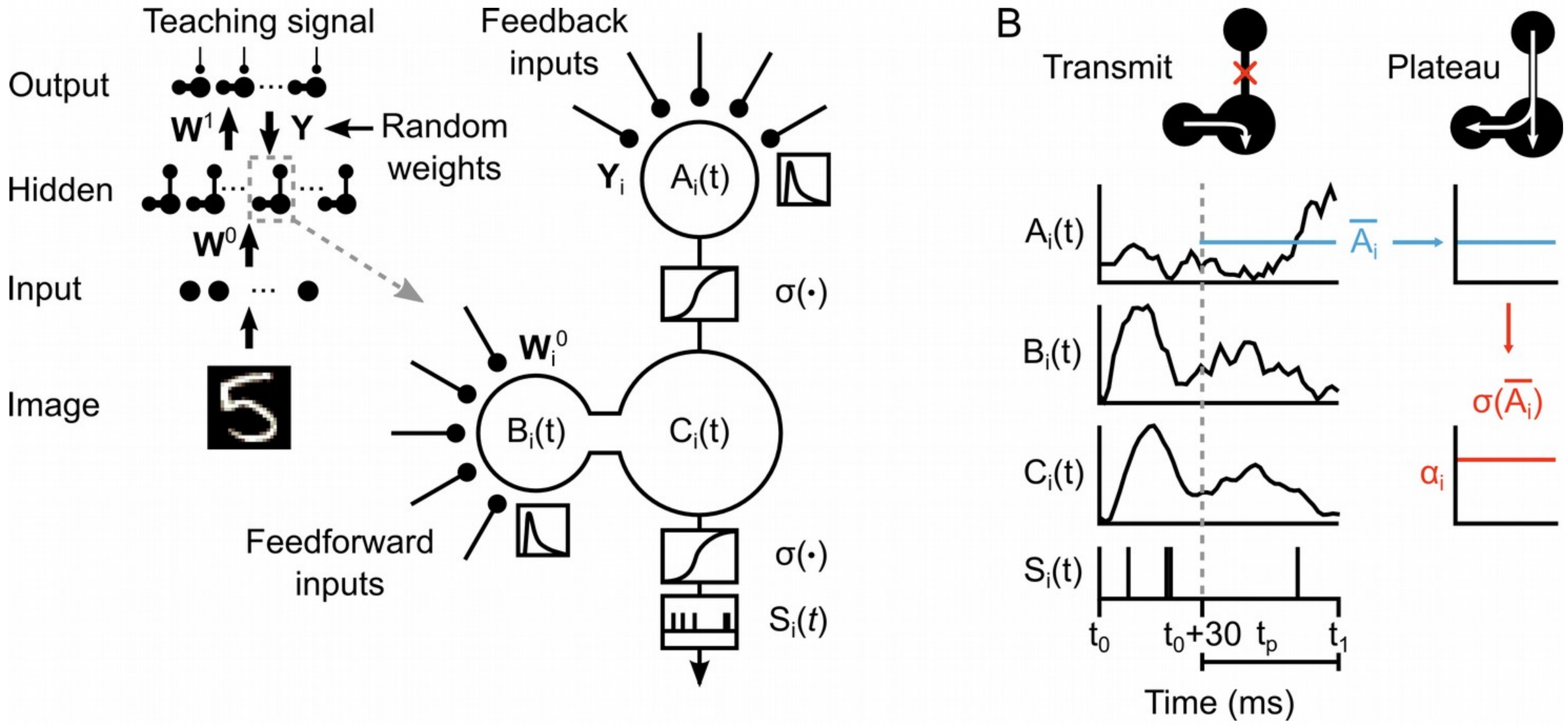


How they do drive activity is via non-linear "**plateau potentials**" in the apical shaft

These plateau potentials induce ***bursts*** of spikes

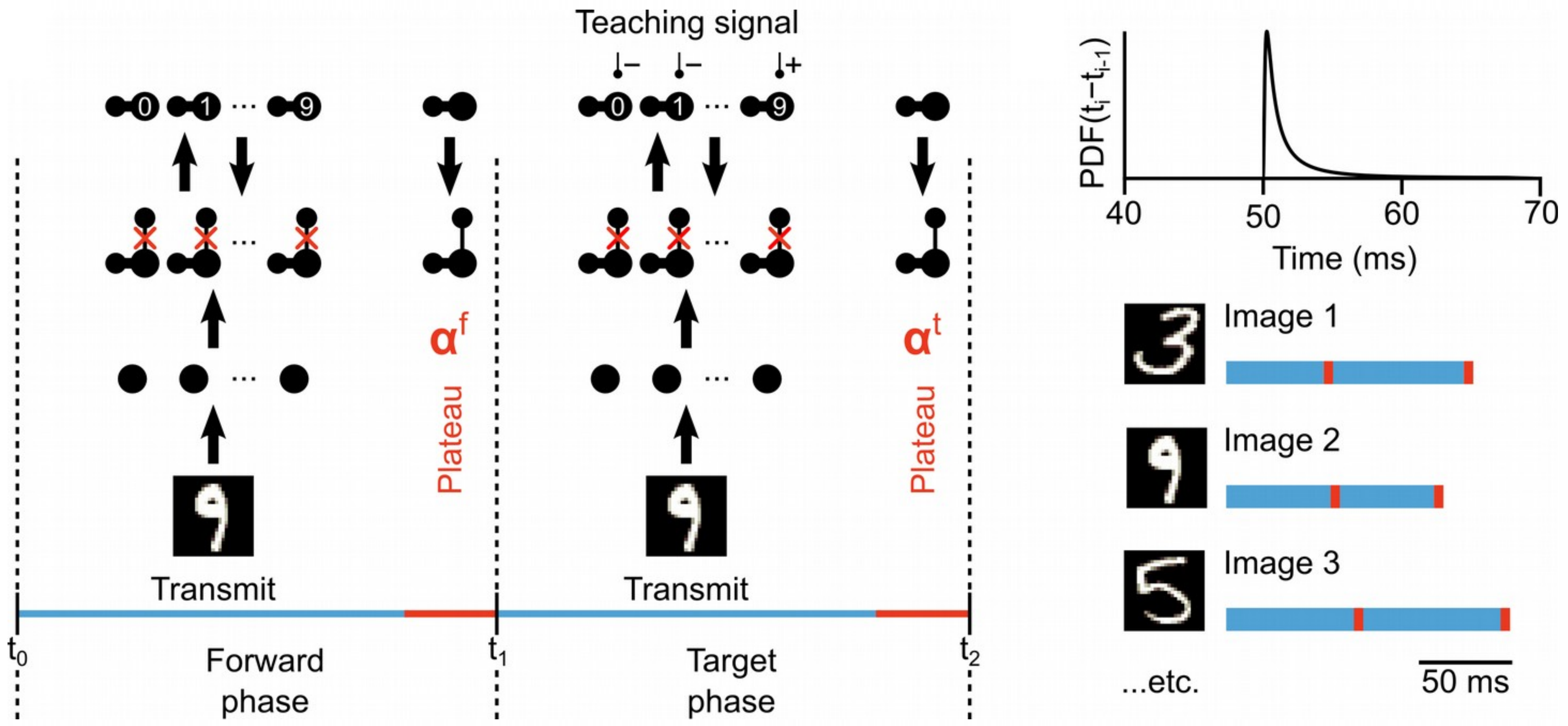
Larkum, Zhu and Sakmann. (1999) Nature, 398(6725)

8. Issue four: forward/backward passes



Guerguiev, Lillicrap and Richards (2017), arXiv: 1610.00161

8. Issue four: forward/backward passes



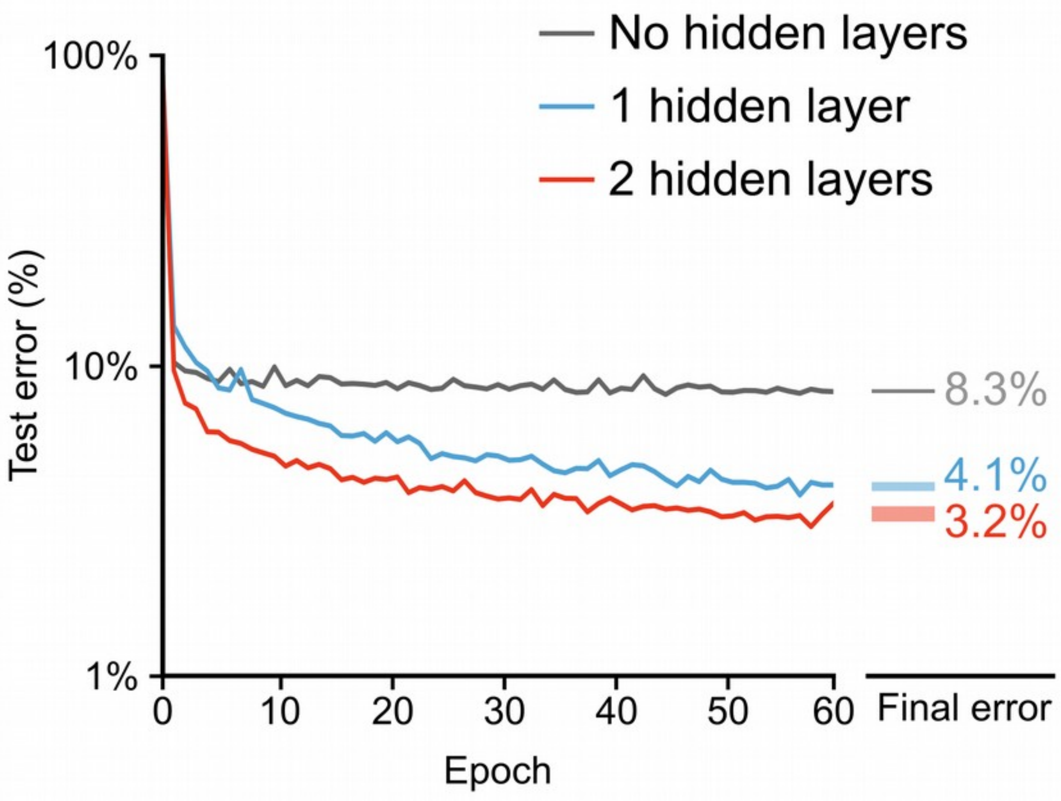
We then update the weights using the difference between the plateau potentials:

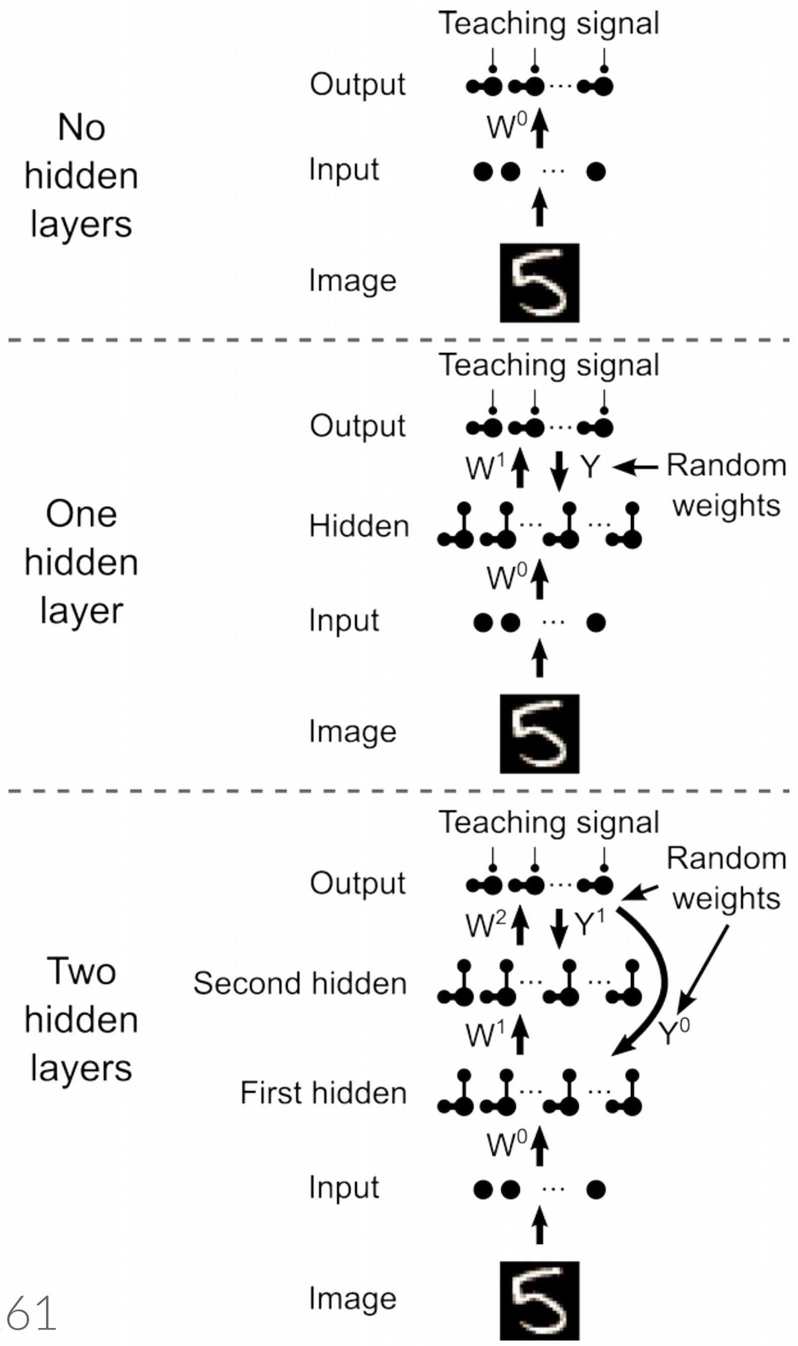
$$\Delta W^0 \propto \alpha^t - \alpha^f$$

Guerguiev, Lillicrap and Richards (2017), arXiv: 1610.00161

8. Issue four: forward/backward passes

Our model exhibits deep learning (light), without separate forward/backward passes because adding hidden layers improves performance

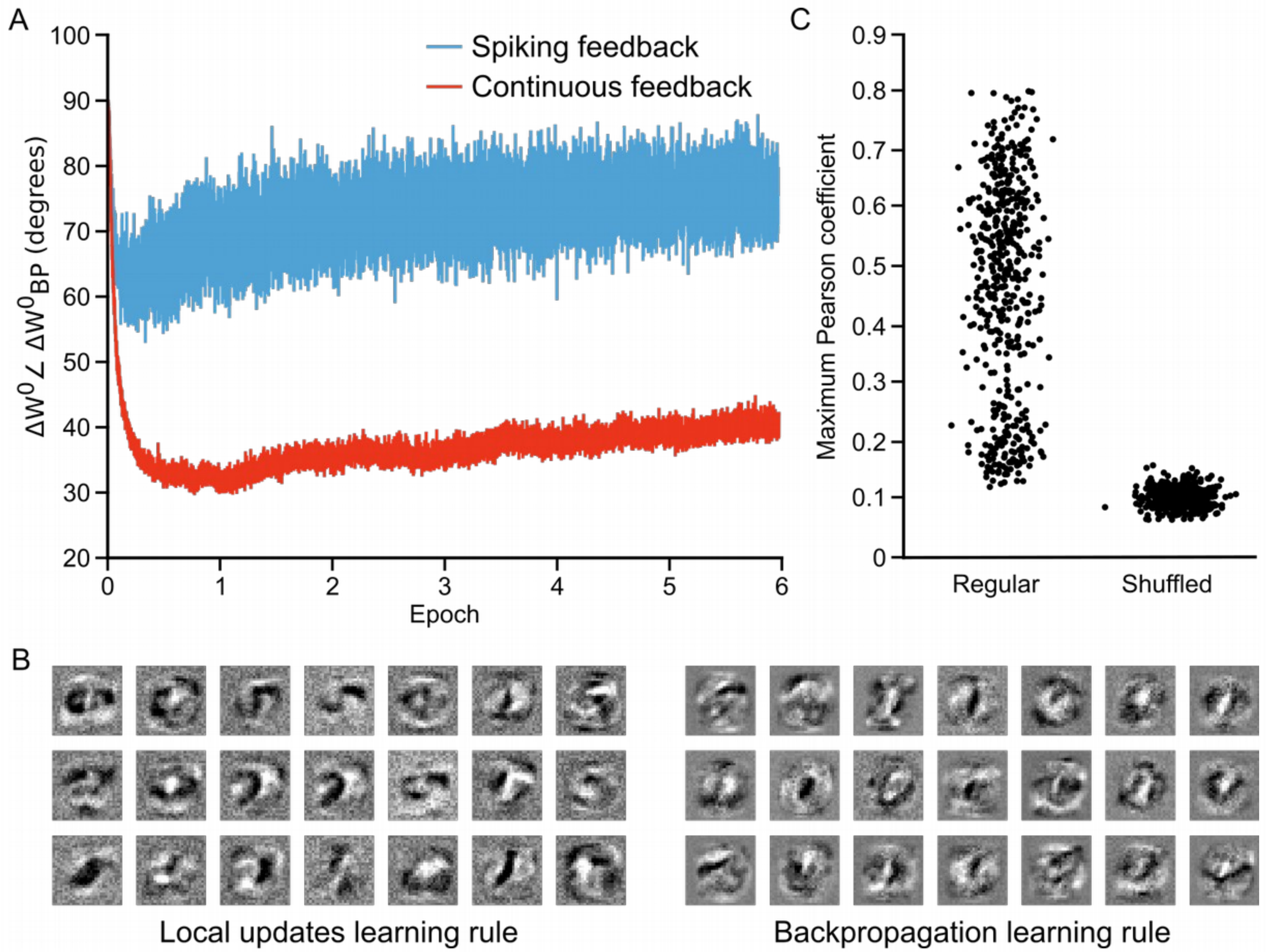




Guerguiev, Lillicrap and Richards (2017), arXiv: 1610.00161

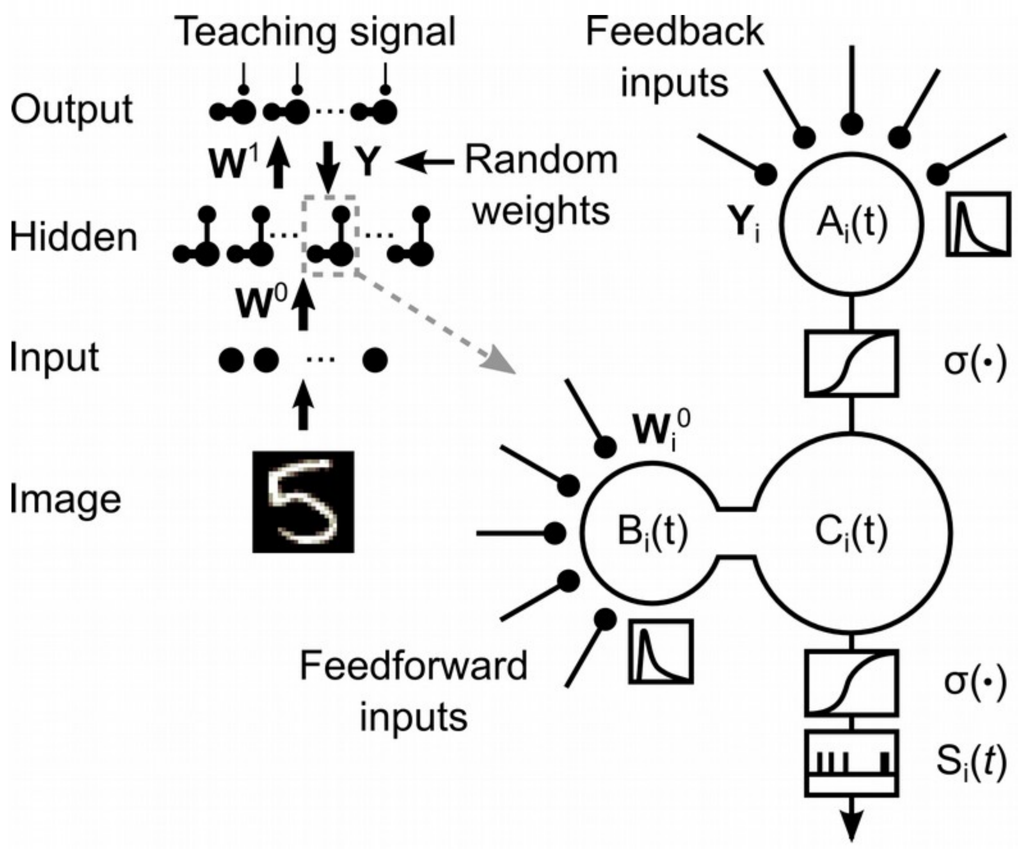
8. Issue four: forward/backward passes

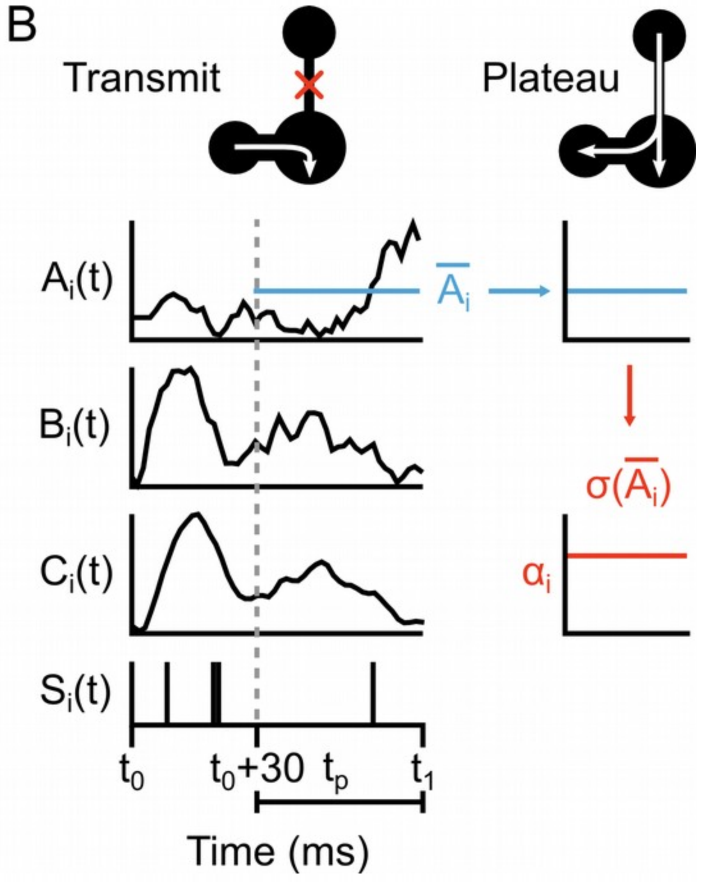
We also see the feedback alignment effect



Guerguiev, Lillicrap and Richards (2017), arXiv: 1610.00161

8. Issue four: forward/backward passes

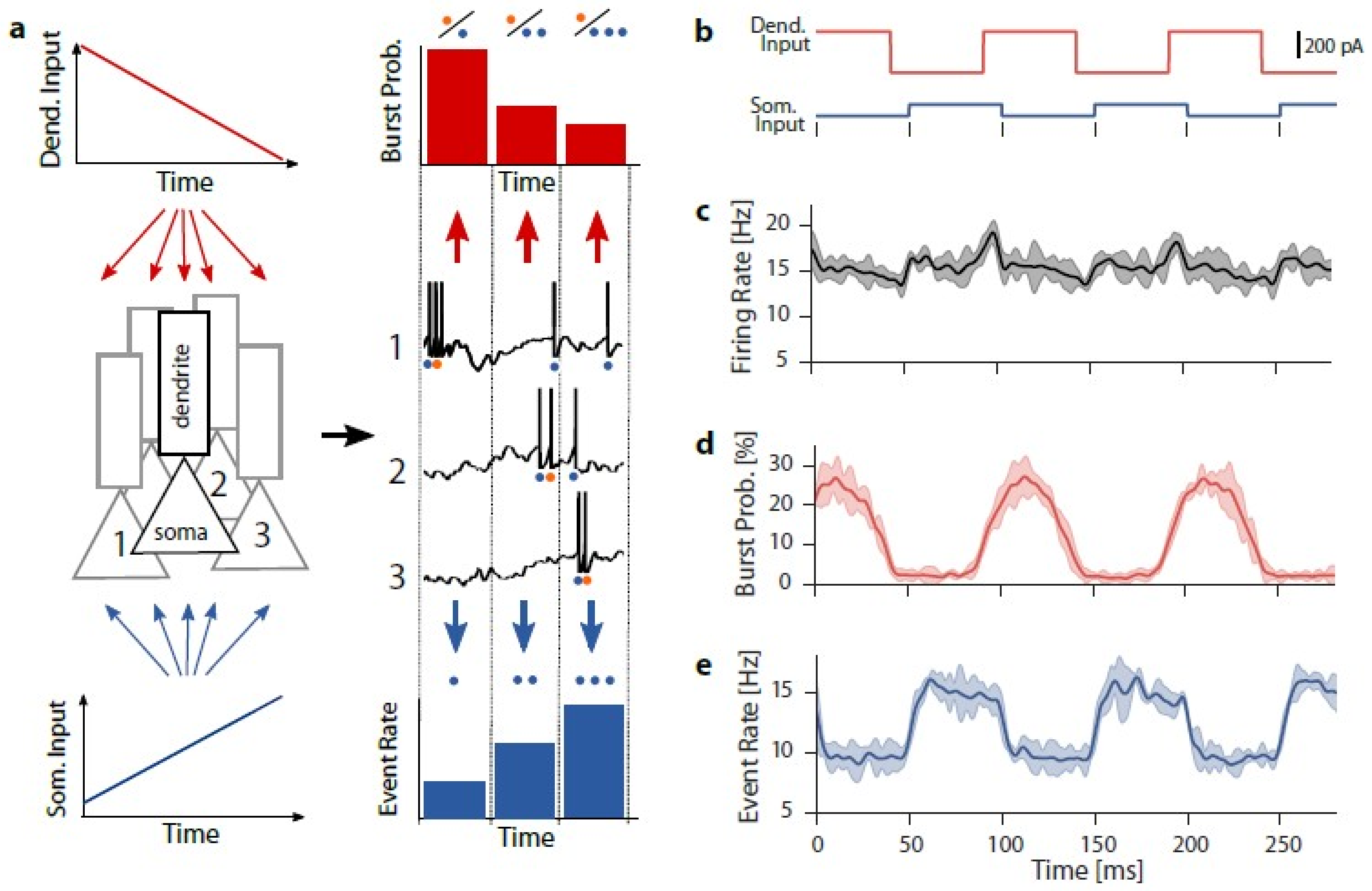




Of course, one issue with our model is that the plateau potentials are not generating bursts, but experiments show that bursts are probably the key driver of plasticity. It is not immediately clear how a real cell would differentiate between a top-down and bottom-up signal, since bursts result from coordinated top-down/bottom-up inputs

Guerguiev, Lillicrap and Richards (2017), arXiv: 1610.00161

8. Issue four: forward/backward passes



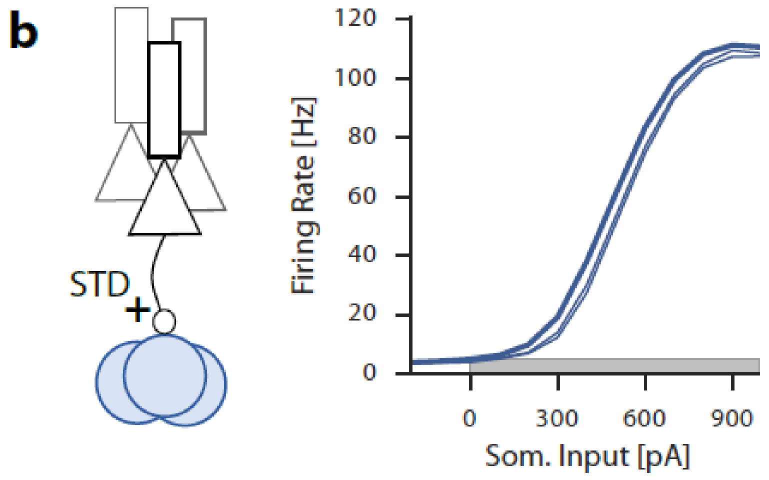
New result from Richard Naud (uOttawa): if we treat bursts and spikes as "events" and examine the event rate vs. the probability of an event being a burst, we can see that pyramidal neurons multiplex their top-down and bottom-up signals

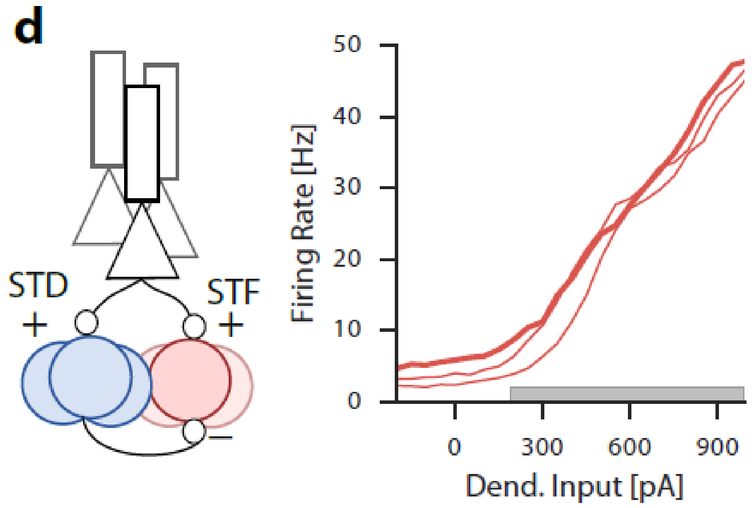
Naud & Sprekeler (2017) Unpuplished

8. Issue four: forward/backward passes

Interestingly, there are sub-types of neurons in the neocortex that have synapses which respond to bursts differently

Some cells have short-term depressing synapses (STD) which means they don't really respond to bursts differently than spikes, while others have short-term facilitating synapses (STF) that only respond to bursts





Hence, there are specialized cells in the neocortex that respond to just the top-down or just the bottom-up signals

Naud & Sprekeler (2017) Unpuplished

8. Issue four: forward/backward passes

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

We need:

~~(1) Error term~~

~~(2) Transpose of downstream weights~~

~~(3) Derivative of activation function~~

~~(4) Separate forward and backward passes~~

Four items down:

The neocortex is effectively designed to multiplex forward passes and backward passes simultaneously!

9. Summary

I began this talk by identifying four major issues with the biological plausibility of the backpropagation weight update rule for hidden layers:

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

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

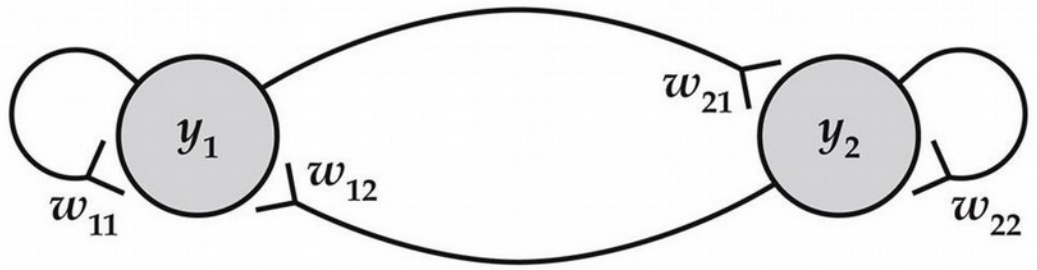
For decades, neuroscientists have dismissed the possibility of deep learning in the brain because of these issues, but over the last two years every one of these problems have been demonstrated as being very surmountable!!!

10. Major unresolved issue: backprop through time

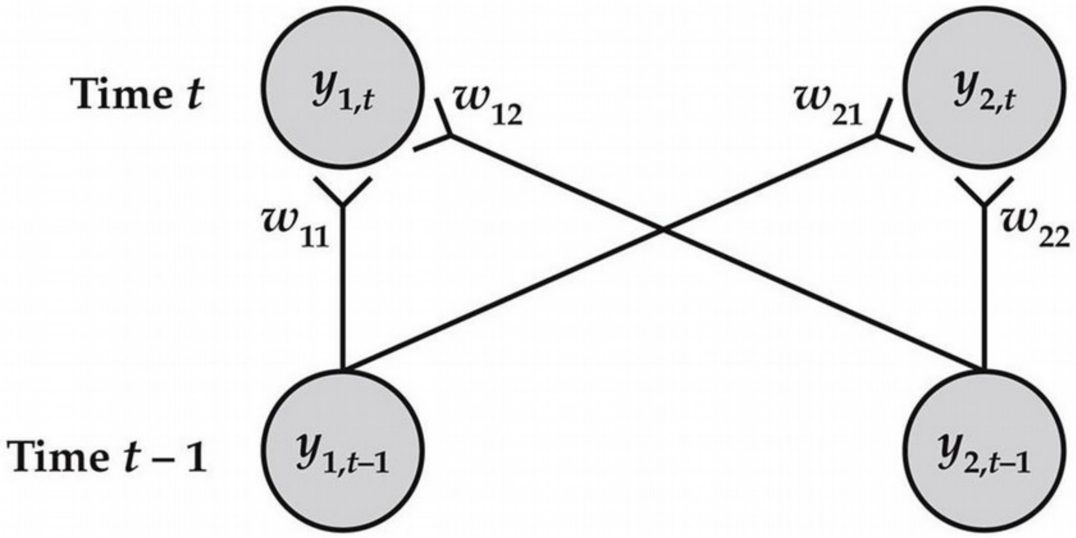
This is all very exciting, but there's still an elephant in the room: *backprop through time*

10. Major unresolved issue: backprop through time

Backprop through time is critical for training recurrent nets, but it is very hard to see how it could be done in a biologically realistic manner

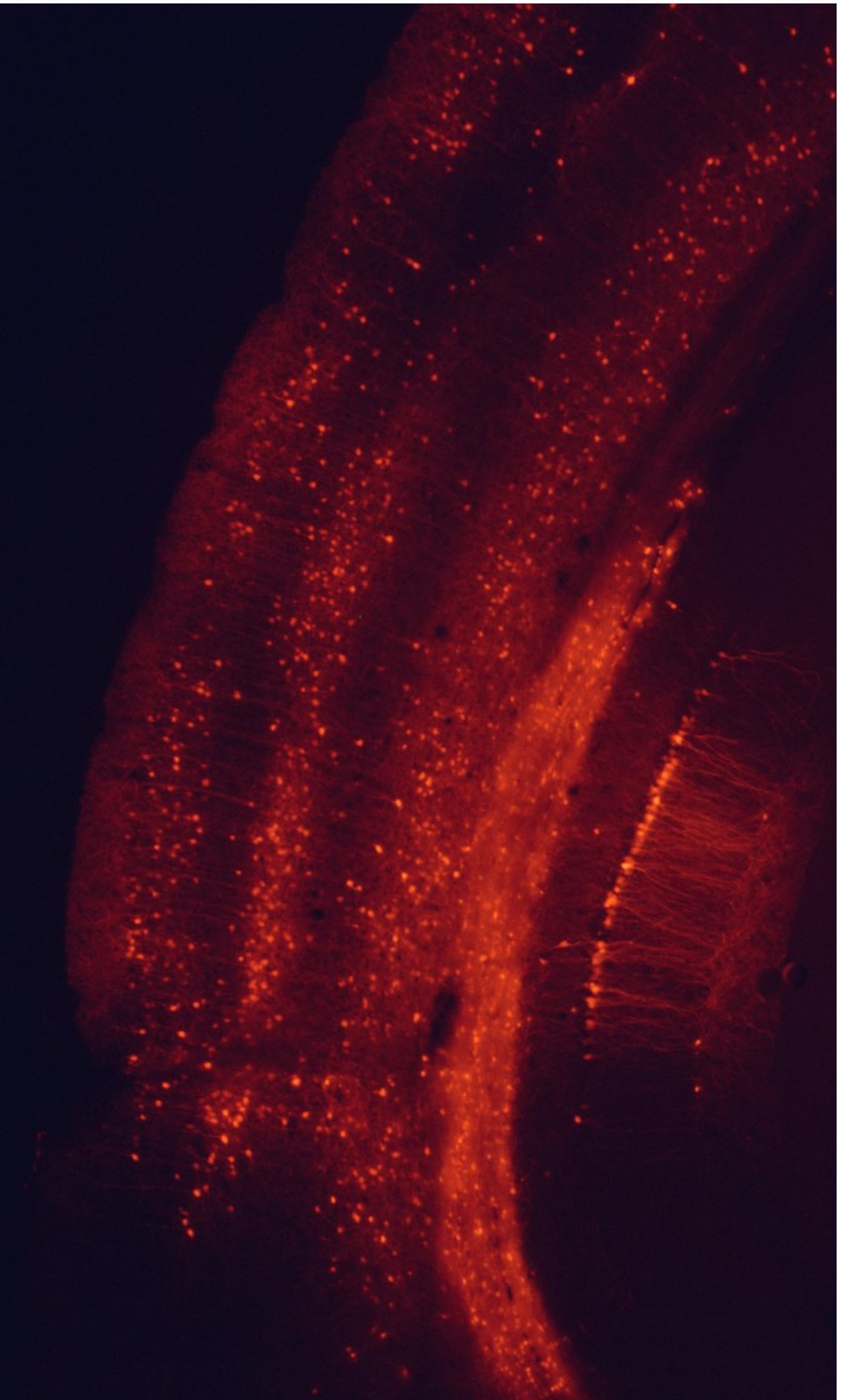


As it is, backprop through time requires time-stamped records of activity patterns and inputs – not an easy ask for a group of real neurons



There might be some ways to address this, but I'm gonna leave this as an unresolved problem for you youngin's to work out

11. Conclusion

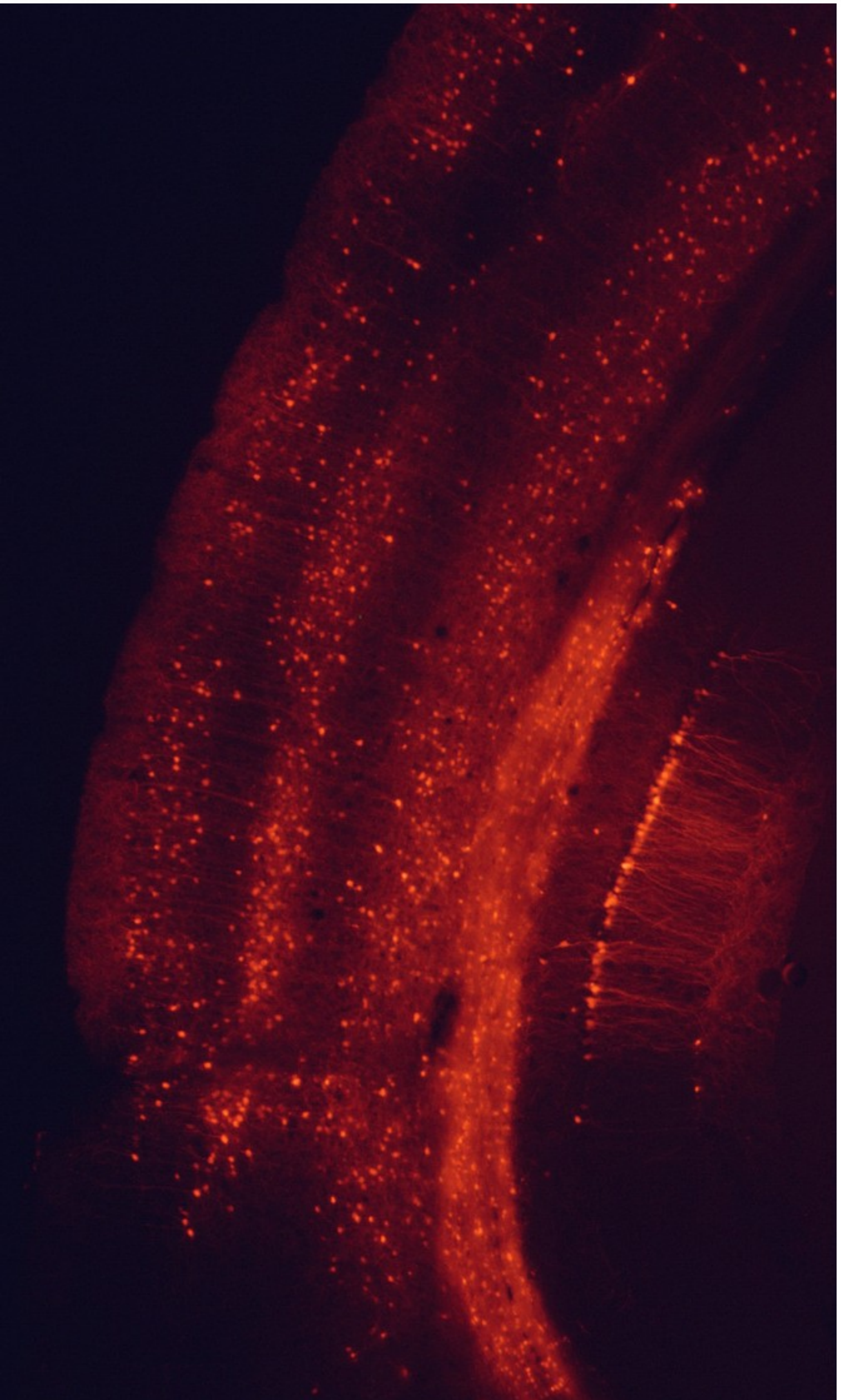


There's a good reason that deep learning has taken over AI – *it works*

The reasons it works in AI apply equally to our own brains – evolution must have come to use some approximation of gradient descent because learning in such high-D space is too hard otherwise (too many directions!)

Our brains may not do backprop as it is done on a computer, but we are getting to the point where the old objections no longer hold water

11. Conclusion



I'll leave you with the following cool/scary thought...

If our brains are doing gradient descent, and we can determine the signals used to indicate the gradient, then in the future (with good enough neural prostheses) we could do backprop through an AI back into the brain!

This could give us seamless brain-AI interfaces, which would open the door to some pretty crazy tech...

Questions?

Deep learning summer school
Montreal 2017

Special thanks to:
Jordan Guerguiev (U of Toronto)
Timothy Lillicrap (Google DeepMind)
Richard Naud (U of Ottawa)

Financial support

NSERC CRSNG

HFSP

Google

CIFAR | ICRA
CANADIAN INSTITUTE FOR ADVANCED RESEARCH | INSTITUT CANADIEN DE RECHERCHES AVANCÉES

LiNC LAB
linclab.org

UNIVERSITY OF TORONTO SCARBOROUGH