How do multi-layered networks in the brain learn?

Timothy P. Lillicrap[1], Daniel Cownden[2],
Douglas B. Tweed[3], Colin J. Akerman[1]

[1] University of Oxford; [2] University of St Andrews; [3] University of Toronto

August 14, 2014

Collaborators

Colin

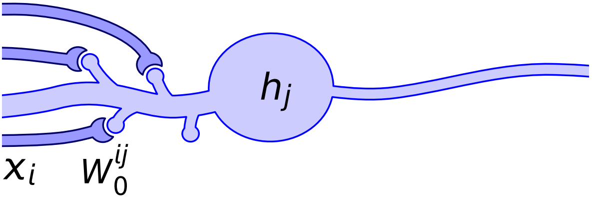
Doug

Tim Daniel

Outline

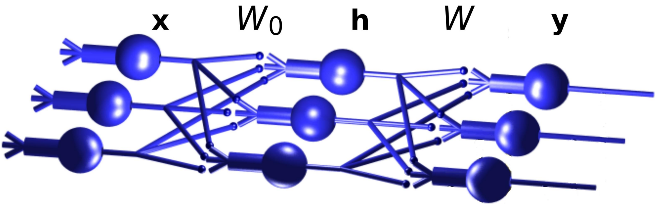
1. Deep Learning and the "Weight Transport" problem
2. Our Solution, Feedback Alignment
3. Show it works
4. Intuition as to why it might work
5. Back to the brain

A cartoon of a neuron



The activity of the hidden neuron h_j is a function of the spiking activity of upstream neurons:
$h_j \propto \sigma\left(\sum_i x_i W_0^{ij}\right)$

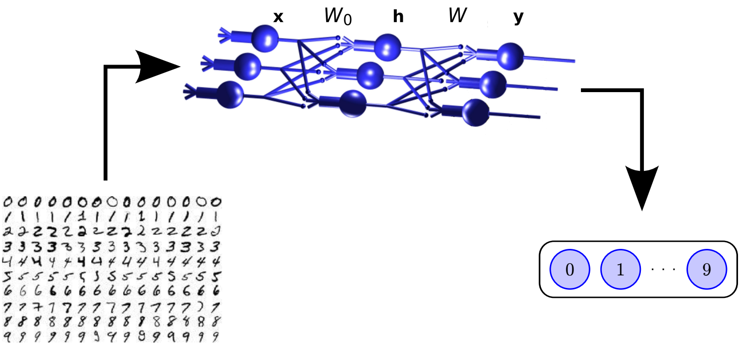
A cartoon of a network of neurons



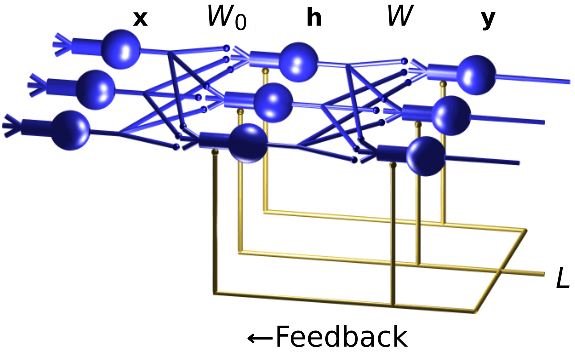
The network computes via:

$$\mathbf{h} = \boldsymbol{\sigma}(W_0 \mathbf{x})$$
$$\mathbf{y} = \boldsymbol{\sigma}(W \mathbf{h})$$

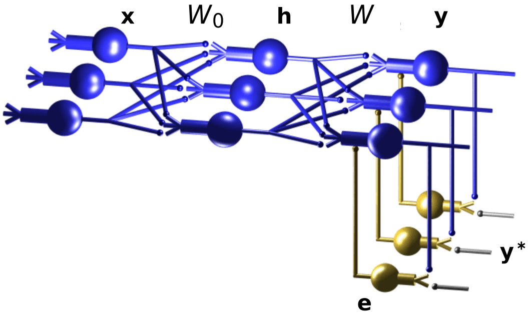
What can these networks do?



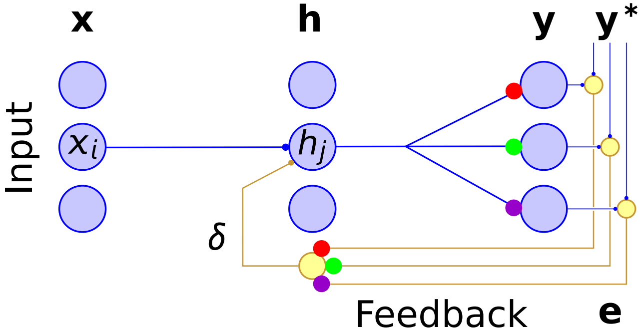
How are network connections adapted to tasks?



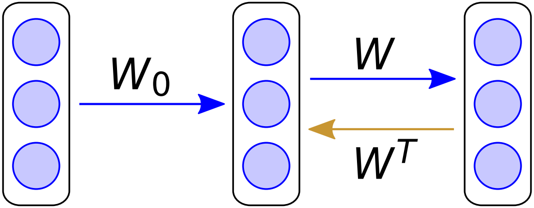
Learning is faster if we know how we screwed up, if we use errors.



How are errors used deep within the network?



Backprop *is* gradient descent; weights changes correspond precisely with contribution to loss.



- Weight updates, in the linear case, are:

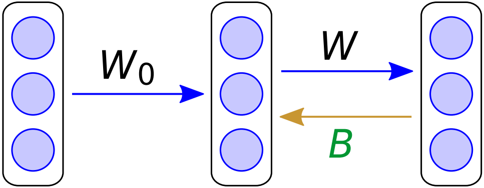
$$\Delta W \propto \partial L/\partial W = \mathbf{e}\mathbf{h}^{\mathsf{T}}, \text{ and}$$
$$\Delta W_0 \propto \partial L/\partial W_0 = (\partial L/\partial \mathbf{h})(\partial \mathbf{h}/\partial W_0) = -W^{\mathsf{T}}\mathbf{e}\mathbf{x}^{\mathsf{T}}.$$

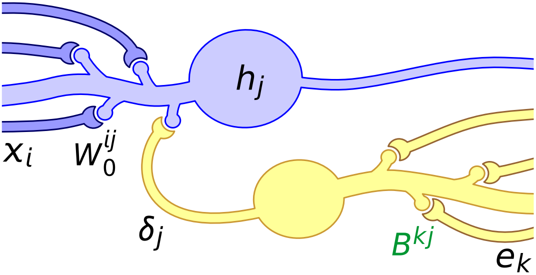
- Feedback neurons must "know" the forward weights. This is the *weight transport* problem

Outline

1. Deep Learning and the "Weight Transport" problem
2. Our Solution, Feedback Alignment
3. Show it works
4. Intuition as to why it might work
5. Back to the brain

Instead of sending errors back through W^T errors can be sent through fixed random connections B.

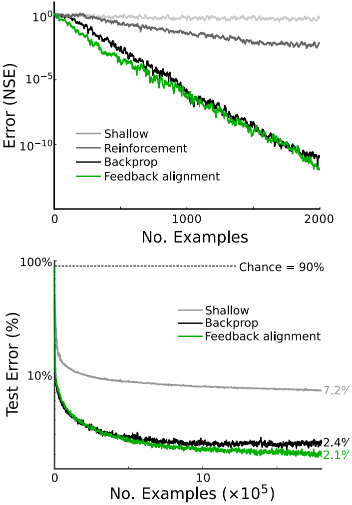




Deep neurons recieve a random projection of the errors as feedback. Initially, this feedback is totally unrelated to a neuron's actual contributions to the error.

"His policy was to find one person and make their life difficult until everything happened the way he wanted it to. (A policy adopted by almost all managers and several notable gods.)" - Terry Pratchet 'Interesting Times'

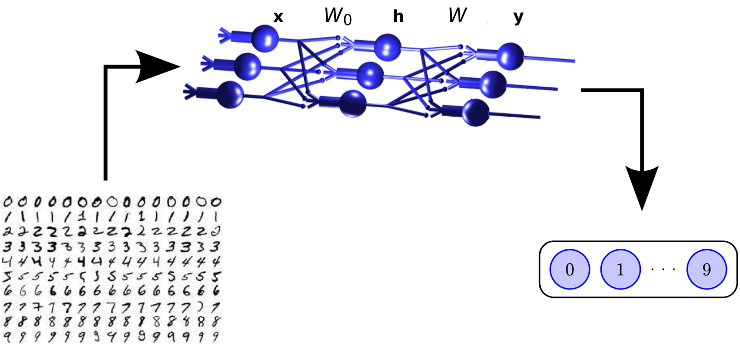
We call this feedback alignment. Shockingly, learning is as fast and as accurate as with backprop.



Outline

1. Deep Learning and the “Weight Transport” problem
2. Our Solution, Feedback Alignment
3. Show it works
4. Intuition as to why it might work
5. Back to the brain

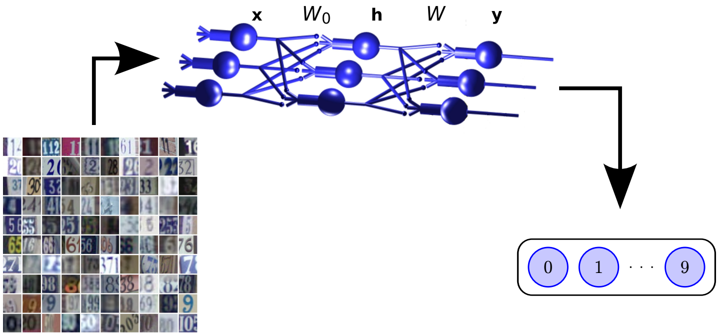
It works on a classic benchmark MNIST.



It works on a classic benchmark MNIST.

- MNIST is 60,000 training and 10,000 testing images of handwritten digits.
- Each image is 28x28 greyscale pixels.
- Train a 784-1500-1500-1500-10 network of tanh units.
- Basic backprop: 1.6% error.
- Basic feedback alignment: 1.3% error.
- We can trick out feedback alignment:
  - Add Drop-out: 1.2% error.
  - Add piecewise linear units: 1.1% error.
  - Add topological information: 0.8% error.
  - Add dataset augmentation: 0.4% error.
- Fully tricked out backprop: 0.2% error.

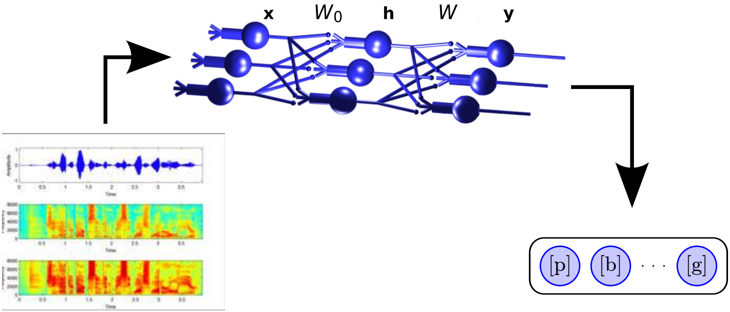
It works on a newer, bigger, harder set of pictures of numbers.



It works on a newer, bigger, harder set of pictures of numbers.

- SVHN is 604,388 training and 26,032 testing images.
- Each image is a 32x32 pixel colour photograph of a house number taken from Google street view.
- Train a 1024-3000-3000-3000-3000-10 network of tanh units.
- Basic backprop: 10.3% error.
- Basic feedback alignment: 9.7% error.
- We can trick out feedback alignment:
  - topological information: 8.1% error.
  - Add piecewise linear units: 7.1% error.

It works on a phoneme categorization task.



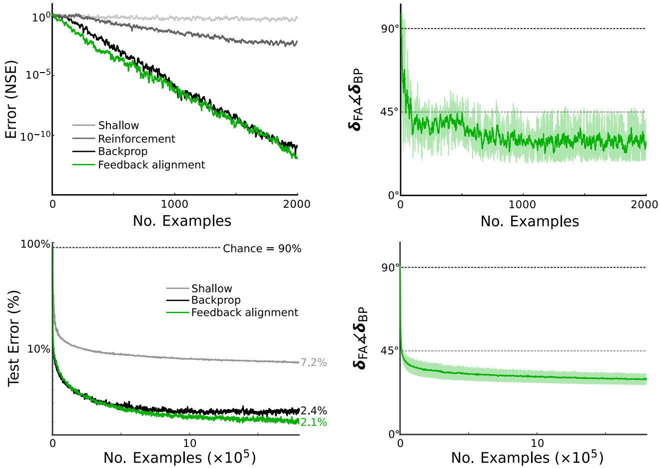
It works on a phoneme categorization task.

- TIMIT is 63,881 training and 22,257 testing phoneme labled audio frames.
- 630 individuals from eight American English dialects, a variety of phonetic content.
- 10ms audio frames converted to Mel-frequency cepstral coefficients (MFCC) features.
- Inputs are the MFCC's concatenated with their first two temporal derivatives.
  This gives a 39 dimensional input space.
- Train a 39-1000-1000-1000-6 network of tanh units.
- Basic backprop: 24.3% error.
- Basic feedback alignment: 23.1% error.

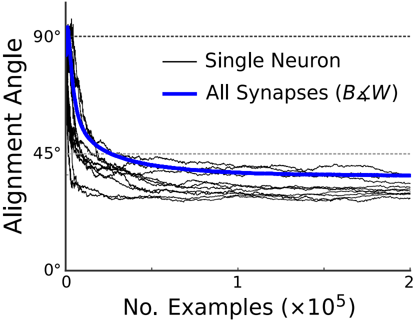
Outline

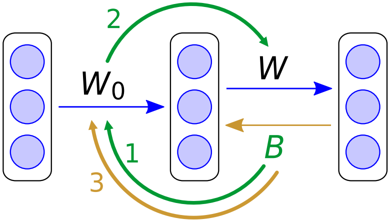
1. Deep Learning and the “Weight Transport” problem
2. Our Solution, Feedback Alignment
3. Show it works
4. Intuition as to why it might work
5. Back to the brain

With Feedback Alignment the network learns to learn. Initial teaching signal are useless, but they swiftly align with those prescribed by backprop.

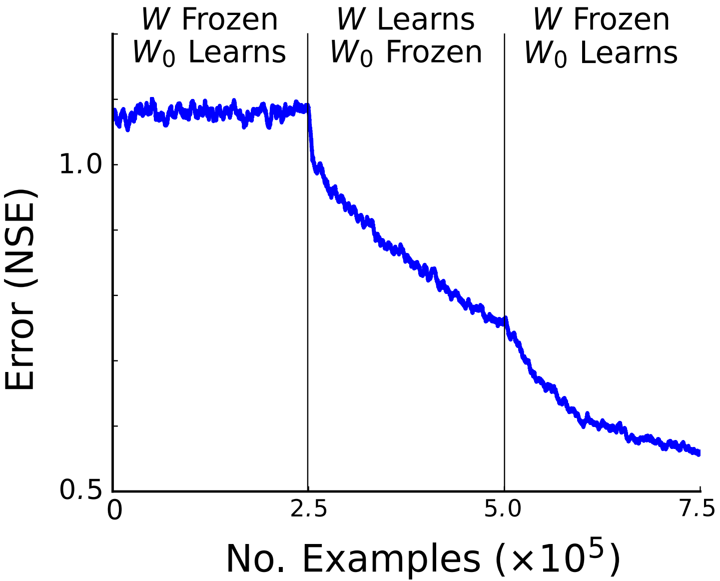


Teaching signals align because W is aligning with B.





Revealing the flow of information from B into W, via W_0.



Outline

1. Deep Learning and the "Weight Transport" problem
2. Our Solution, Feedback Alignment
3. Show it works
4. Intuition as to why it might work
5. Back to the brain

Fast deep learning requires unit-specific error feedback. Backprop requires feedback be sent via a precise symmetric copy of the downstream synaptic weight matrix. How could such detailed feedback information be propagated backward in biological networks?

- It doesn't need to be!

Deep learning algorithms compare actual outputs to desired outputs, in order to produce error signals. Where do desired outputs come from and how are errors computed and represented in the brain?

- From Sensory/Motor and Prediction Mismatches.
- Favourite reference:
  Bell et al. (1997) The generation and subtraction of sensory expectations within cerebellum-like structures.

Once error signals have been generated, how can they be delivered and used without disrupting forward activity.

- Modulatory third factors!
  1. Neuromodulators.
  2. Calcium synapses.
  3. Differentiated sub-cellular compartments.
- They all can alter the magnitude and sign of Hebbian and STDP induction.
- Favourite reference:
  Coesmans et al. (2004) Bidirectional parallel fiber plasticity in the cerebellum under climbing fiber control.

What have we done? What Haven't we done?

- Shown that we don't need to transport weight information at all.
- Forward connection magically align to make use of fixed random feedback.
- This gives us a rough outline of how the empirically observed error neurons, and third-factor plasticity modulators might interact to produce an effective neural network.
- We have largely ignored time.
  - The machine learning problems are all static
  - Real brains produce a *stream* of motor commands contingent of a *stream* of sensory inputs and reccurrent neural activity.

Thank you for listening

Acknowledgements:
Swedish Research Council Grant 2009-2390 and a Templeton Foundation Grant.

Kimmo Eriksson and Kevin Laland.

Geoffrey Hinton and Ilya Sutskever.