Alternatives to backprop

Daniel Jiang

with some help from Brian Cheung, Redwood Center for Theoretical Neuroscience

Outline

1. Deep learning limitations
2. Two specific alternatives to backprop:
    a. Decoupled neural interfaces
    b. Feedback alignment
3. Other alternatives

Deep learning has limitations and issues

- Requires tons of labeled data, not good at online learning, synchronous (and slow) forward and backward passes, vanishing and exploding gradients, highly-sensitive to non-stationary distributions, requires everything to be differentiable, highly-susceptible to over-fitting, not efficient for distributed computation, bad at learning dependencies over time, etc.
- Limitations demonstrated by https://www.reddit.com/r/ProgrammerHumor/comments/8t011g/machine_learning_irl/
- Things we could learn from the brain: local learning rules, meta-learning, unsupervised learning, asynchronous and distributed computation, ...
- We should explore out-of-the-box ideas

Backprop. since 1986...

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

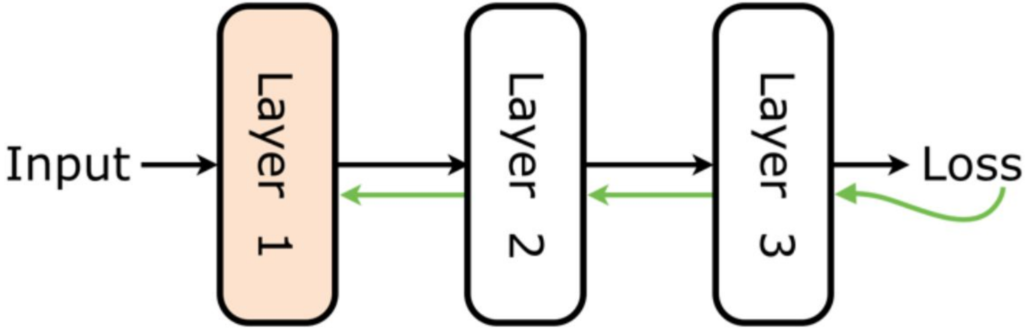
* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

Paper #1: Decoupled neural interfaces

- Decoupled Neural Interfaces using Synthetic Gradients. M. Jaderberg, W. M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, D. Silver, K. Kavukcuoglu. *International Conference on Machine Learning* (*ICML*), 2017.

Decoupled neural interfaces - Introduction

- Problem: Neural network layers are locked to each other. An example where this is an issue is when the output of a network is used by many downstream clients; in order to update, have to wait for slowest client.

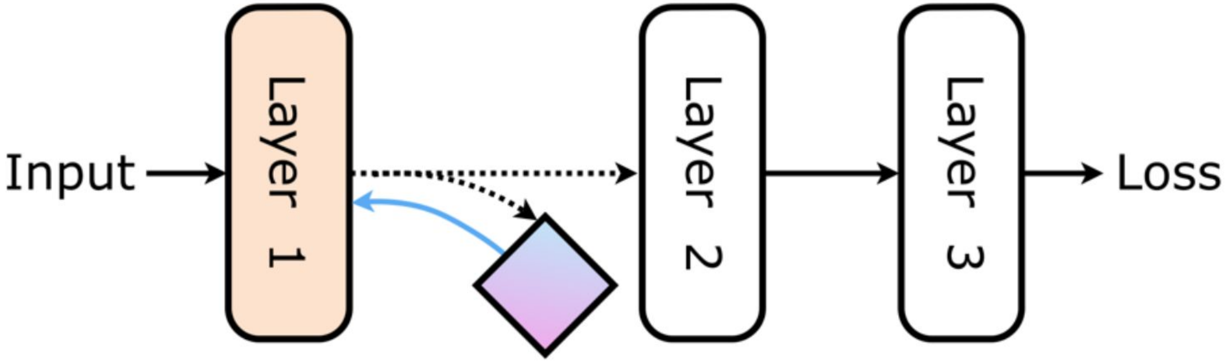


- Solution: Learn a parametric model which predicts what the gradients will be based upon only local information

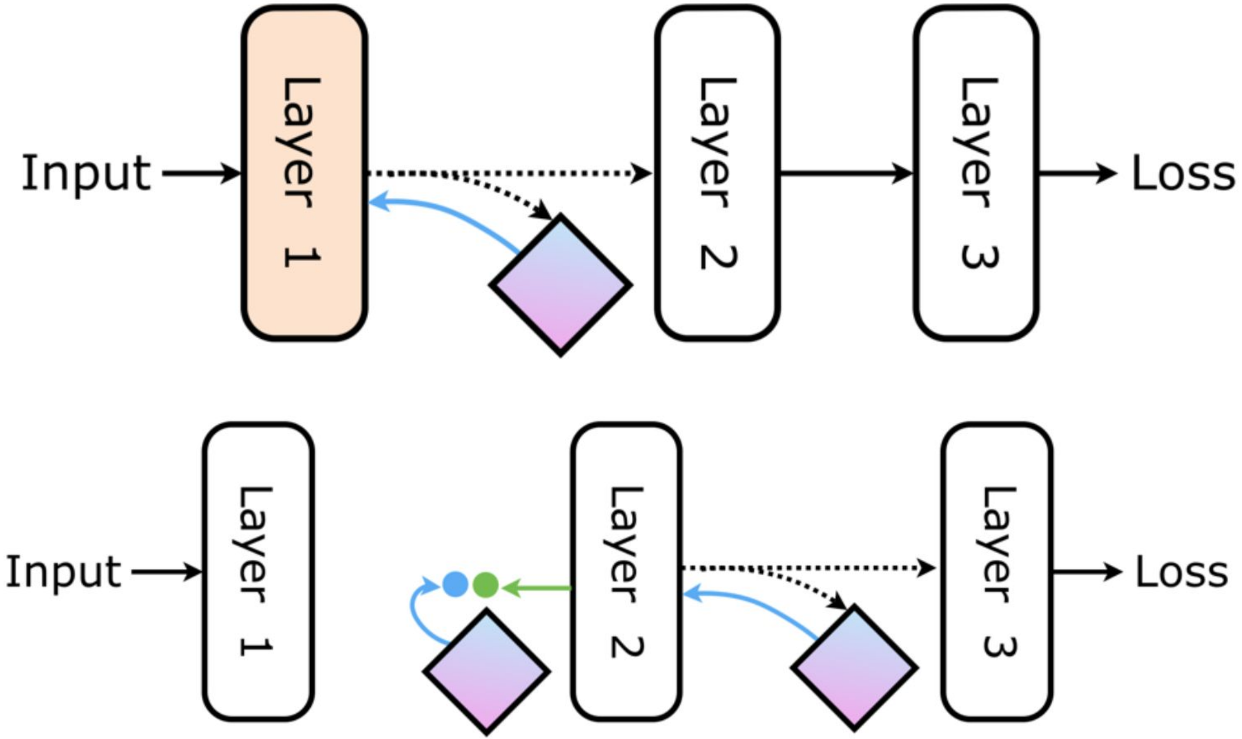
Decoupled neural interfaces - Illustrated

$$\frac{\partial L}{\partial \theta_i} = f_{\text{Bprop}}((h_i, x_i, y_i, \theta_i), \ldots)\frac{\partial h_i}{\partial \theta_i}$$

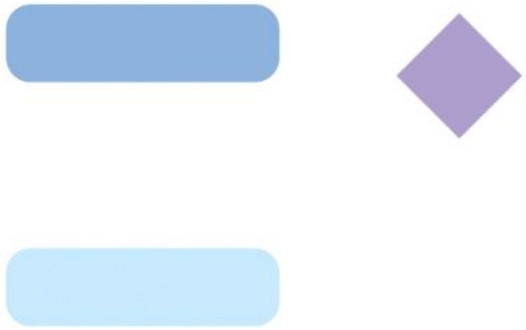
$$\simeq \hat{f}_{\text{Bprop}}(h_i)\frac{\partial h_i}{\partial \theta_i}$$



Decoupled neural interfaces - Illustrated



Decoupled neural interfaces

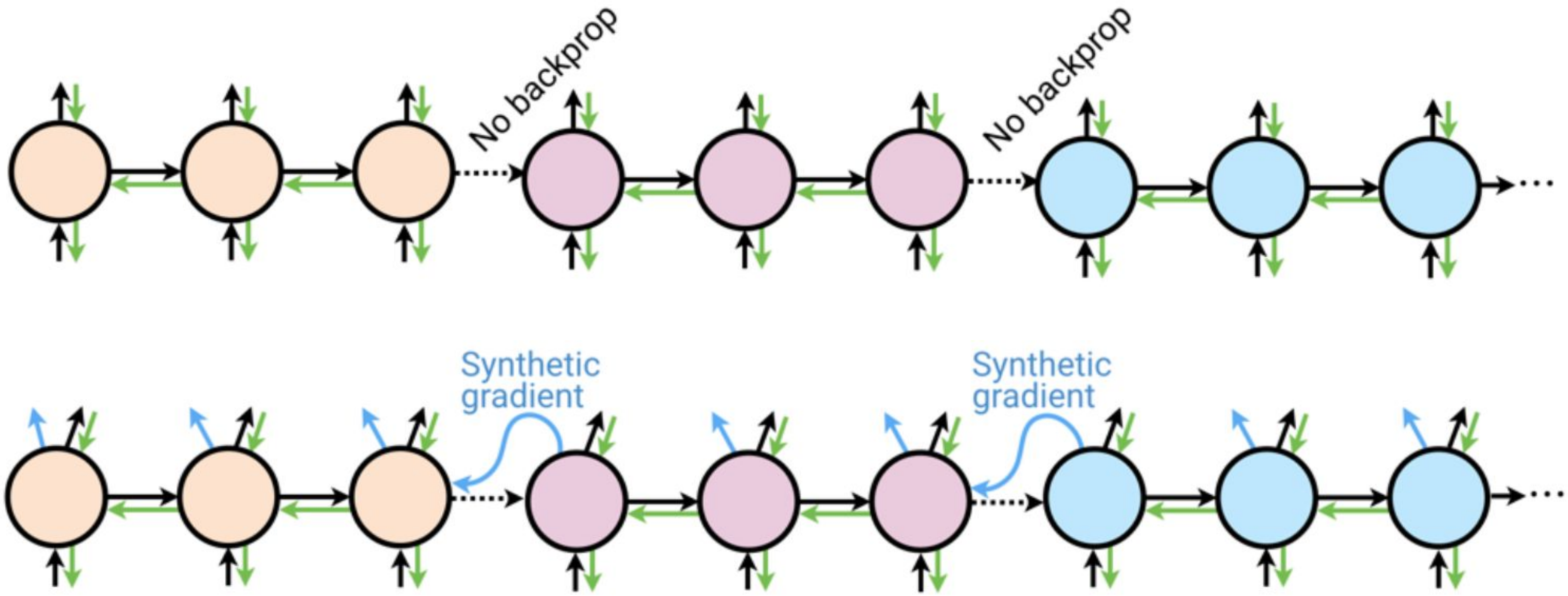


Decoupled neural interfaces - RNNs

$$\theta - \alpha \sum_{\tau=t}^{\infty} \frac{\partial L_\tau}{\partial \theta} = \theta - \alpha(\sum_{\tau=t}^{t+T} \frac{\partial L_\tau}{\partial \theta} + (\sum_{\tau=T+1}^{\infty} \frac{\partial L_\tau}{\partial h_T})\frac{\partial h_T}{\partial \theta})$$

$$= \theta - \alpha(\sum_{\tau=t}^{t+T} \frac{\partial L_\tau}{\partial \theta} + \delta_T \frac{\partial h_T}{\partial \theta})$$

Decoupled neural interfaces - RNNs



Decoupled neural interfaces - Details

- DNI is trained by minimizing the Euclidean distance to the target gradient
- DNI can be one layer
- DNIs are trained to target the backpropagated outputs of downstream DNIs

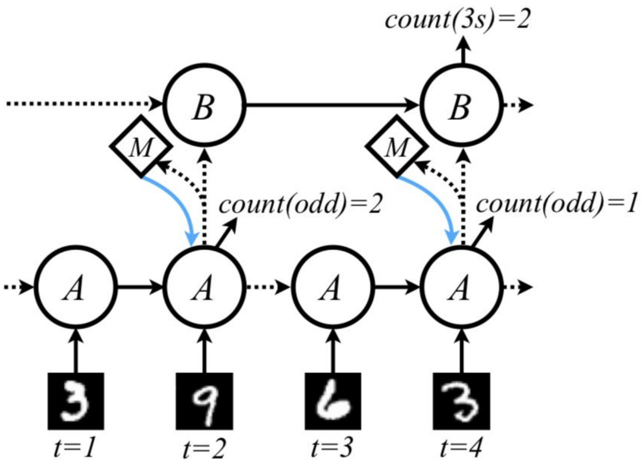
Decoupled neural interfaces - Experiment 1

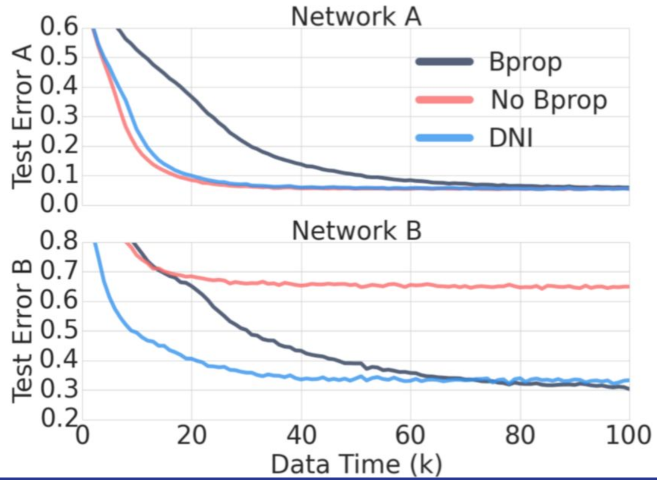
- **Copy**: read sequence of N characters, then repeat the sequence
- **Repeat copy**: read sequence of N characters and number R, then repeat the character sequence R times
- Values in the table are "maximum sequence length that was successfully modelled"

	BPTT							DNI					DNI + Aux				
$T =$	2	3	4	5	8	20	40	2	3	4	5	8	2	3	4	5	8
Copy	7	8	10	8	-	-	-	16	14	18	18	-	16	17	19	18	-
Repeat Copy	7	5	19	23	-	-	-	39	33	39	59	-	39	59	67	59	-
Penn Treebank	1.39	1.38	1.37	1.37	1.35	1.35	1.34	1.37	1.36	1.35	1.35	1.34	1.37	1.36	1.35	1.35	1.33

Decoupled neural interfaces - Experiment 2

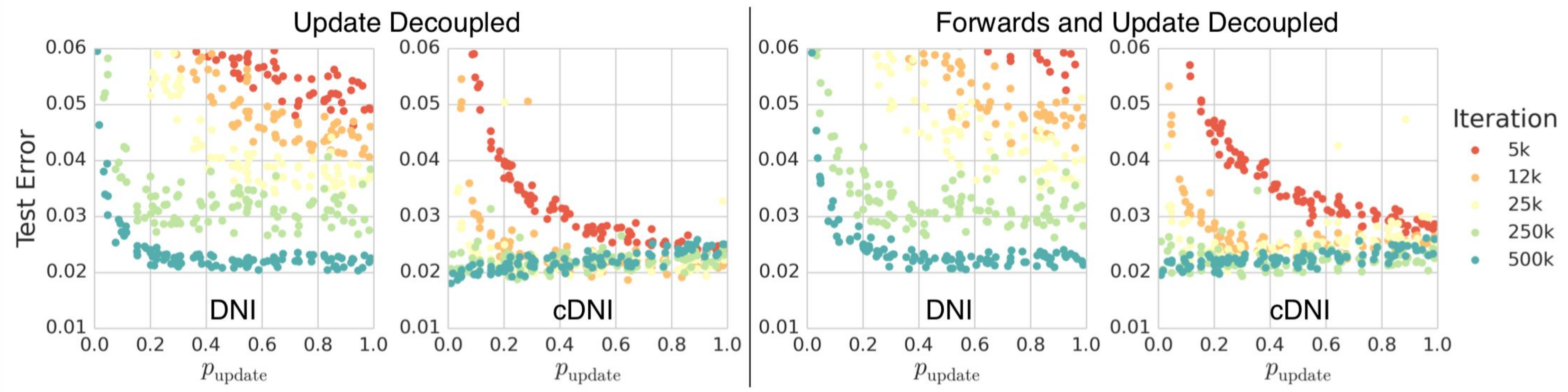
- Network A sees a stream of MNIST digits and must output the number of odd digits seen every T steps. Network B runs every T steps, takes a message from Network A as input and must output the number of 3s seen over the last T^2 steps.





Decoupled neural interfaces - Experiment 3

- Four layer FCNs
- Each layer has probability p_{update} of actually updating
- cDNI means that the DNI is conditioned on the label as well



Decoupled neural interfaces - Extensions

- Synthetic Gradient Methods with Virtual Forward-Backward Networks. T. Miyato, D. Okanohara, S. Maeda, K. Masanori. *International Conference on Learning Representations (ICLR)*, 2017.

$$\delta_l(h, y) := \frac{\partial \ell(y, fwd(h))}{\partial h} = bwd(h) \times (\partial_{fwd(h)} \ell(y, fwd(h)))$$

$$\delta_l(h, y)_{\mathrm{VFBN}} := v'_f(h) \times \partial_{v_f(h)} \ell(y, v_f(h))$$

	Test error (%)
BackProp	5.15
Bottom half with back prop	5.76
Layer-wise supervised loss	5.71
(Synthetic Gradient models)	
Jaderberg's small-ResNet	20.45
Jaderberg's Linear	15.56
VFBN (ours, $\alpha_{gs} = 0$)	5.73
VFBN (ours, $\alpha_{gs} = 1e\text{-}3$)	5.51

Table 1: Test error rates on CIFAR-10

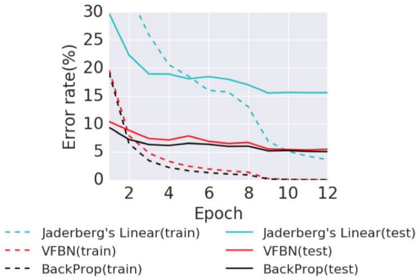


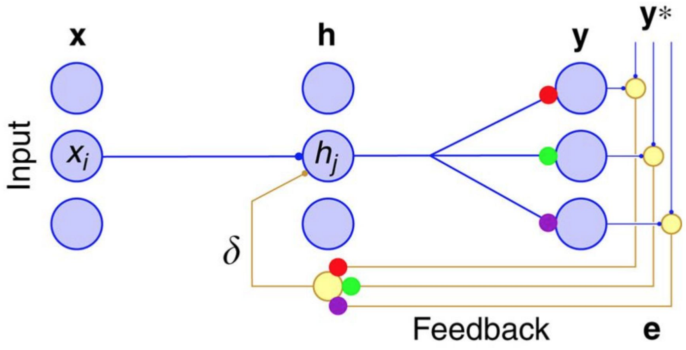
Figure 3: Learning curves on CIFAR-10

Paper #2: Feedback alignment

- Random synaptic feedback weights support error backpropagation for deep learning. T. P. Lillicrap, D. Cownden, D. B. Tweed, C. J. Akerman. *Nature Communications*, 2016

Feedback alignment - Introduction

- Problem: Our brains can't realistically implement backprop for many reasons. One of them is the *weight transport problem*.

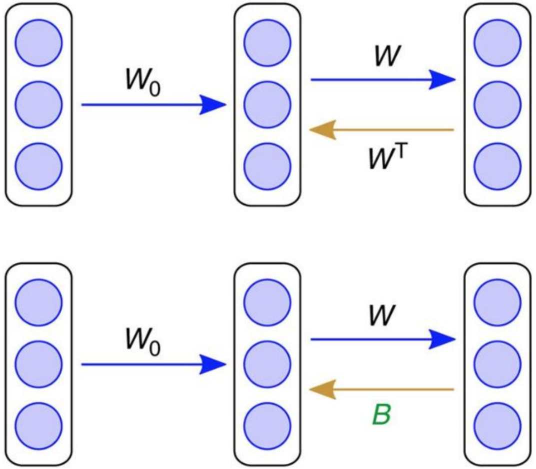


- Solution: Use random matrices for backprop.

Feedback alignment - Illustrated

$$\delta_{BP} = W^T e$$

$$\delta_{FA} = Be$$

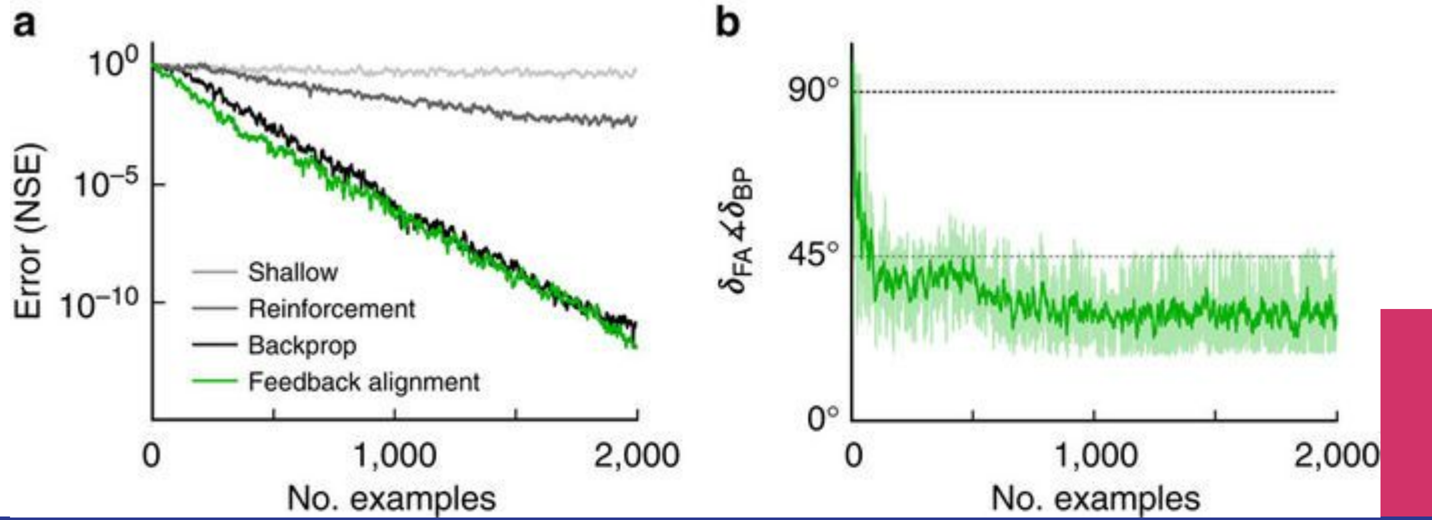


Feedback alignment - Key insights

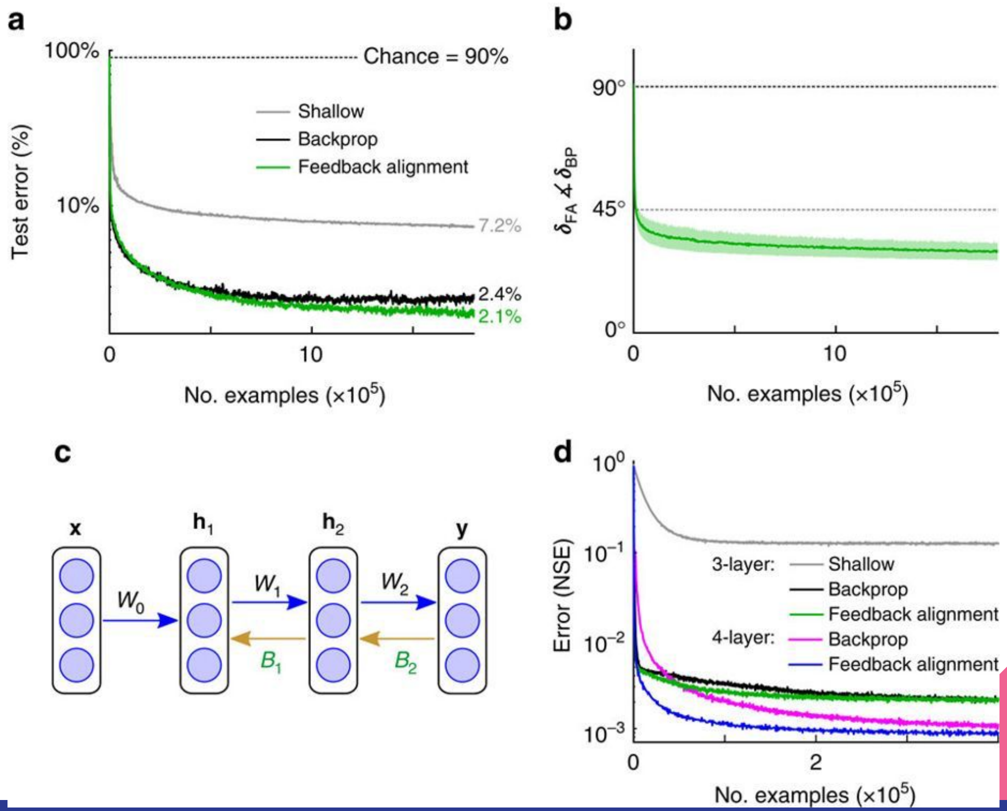
1. All that is necessary is that $\delta_{BP} \cdot \delta_{FA} = (W^T e)^T B e > 0$
   a. i.e. the error signals are within 90 degrees of each other
2. The network could bring B and W into alignment

Feedback alignment - Experiment 1

- Setup: linear 3-layer networks targeting a linear function
  - **Shallow**: only last set of weights train; **Reinforcement**: "fast form of reinforcement learning that delivers the same reward to each neuron"; **Backprop**: backprop; **Feedback alignment**: feedback alignment



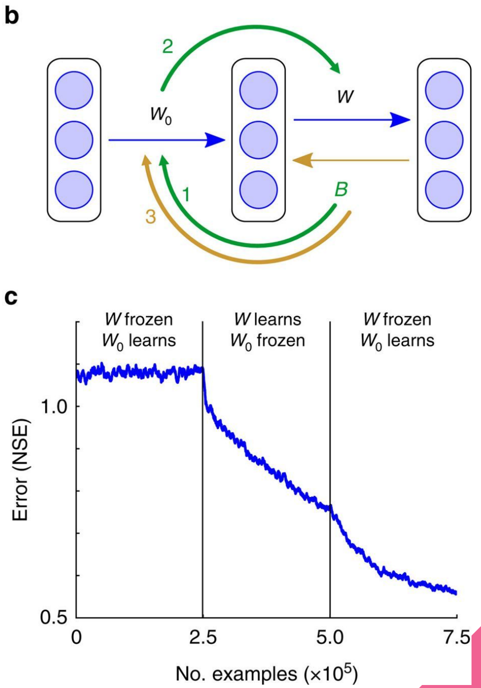
Feedback alignment - Experiment 2



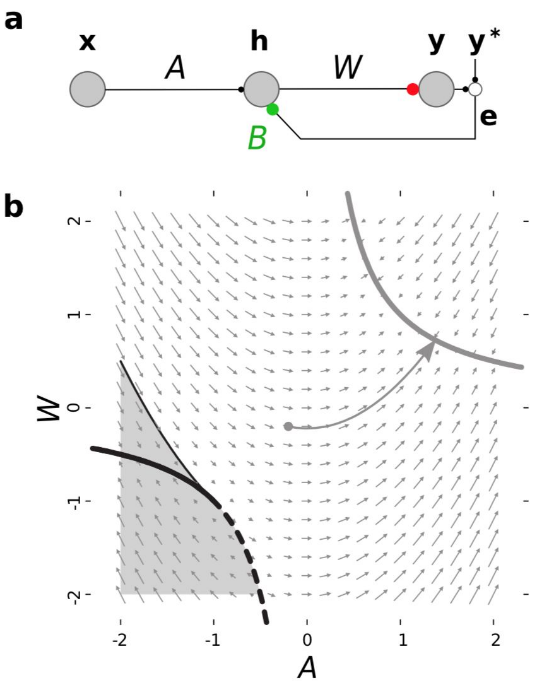
Feedback alignment - Experimental analysis

$$\Delta W_0 \propto B e x^T$$

$$\Delta W \propto e h^T = e(W_0 x)^T$$



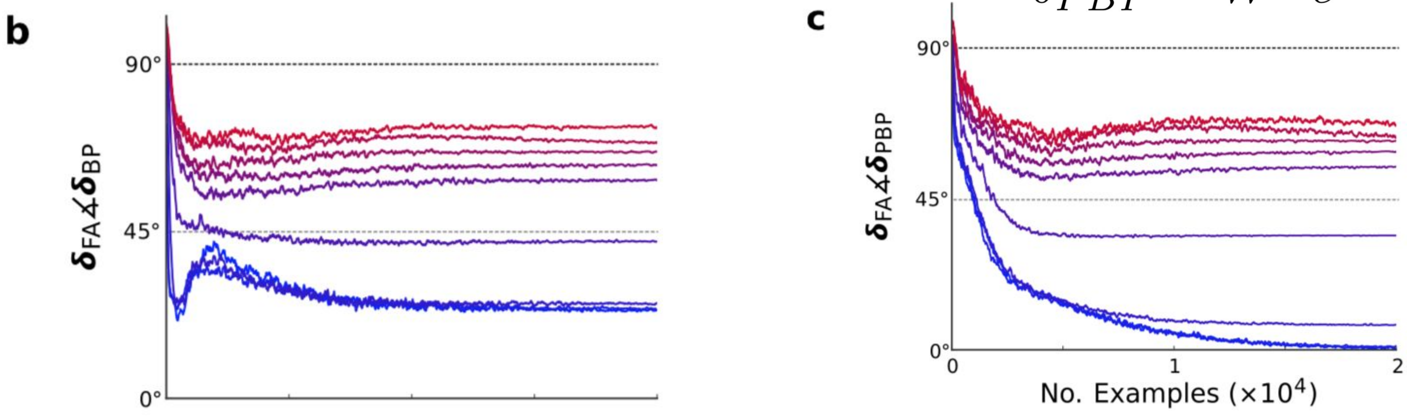
Feedback alignment - Experimental analysis



Feedback alignment - Analysis

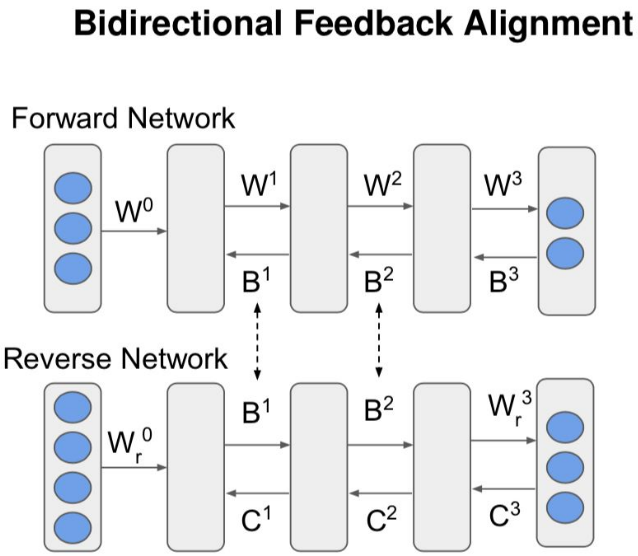
- Intuitively, might think that FA is a sub-optimal approx of BP
- However, FA encourages W to act like a local pseudoinverse of B around the error manifold, which would suggest something similar to Gauss-Newton optimization (2nd order learning)

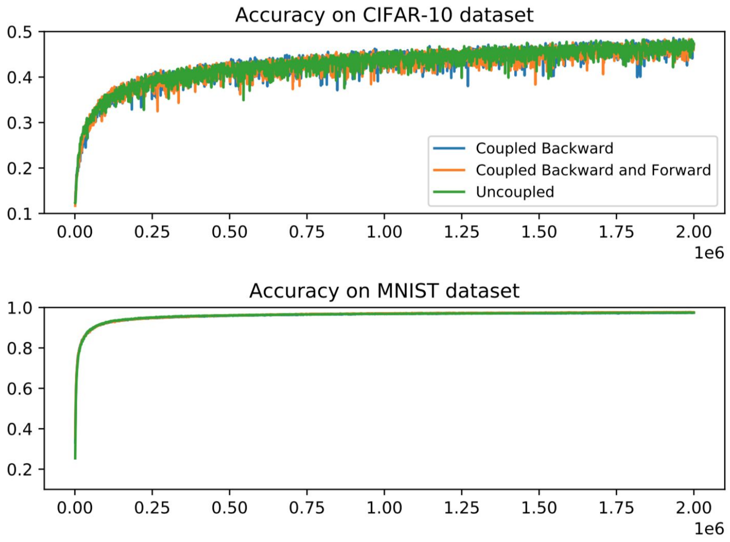
$$\delta_{PBP} = W^{+}e$$



Feedback alignment - Extensions

The many directions of feedback alignment. B. Cheung, D. L. Jiang. Submitted to *Conference on Cognitive Computational Neuroscience* (*CCN*) 2018.





Other methods

- Target propagation and difference target propagation (D. Lee, S. Zhang, A. Fischer, Y. Bengio. 2015)
- Equilibrium propagation (B. Scellier, Y. Bengio. 2017)
- Kickback (D. Balduzzi, H. Vanchinathan, J. Buhmann. 2014)
- Differentiable plasticity (T. Miconi, J. Clune, K. O. Stanley. 2018)

Science progresses one funeral at a time

- "Max Planck said, 'Science progresses one funeral at a time.' The future depends on some graduate student who is deeply suspicious of everything I have said." - Geoffrey Hinton