The many directions of feedback alignment

REDWOOD CENTER for Theoretical Neuroscience

Brian Cheung, Daniel Jiang
{bcheung, danieljiang}@berkeley.edu

BAIR BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Introduction

Backpropagation is a key component in training neural network models which have been successfully applied to many perceptual tasks including vision and audio. An analogous learning mechanism has yet to be discovered in biology. The feedback alignment algorithm relaxes the constraints imposed by the backpropagation procedure. We extend feedback alignment by:

- Loosening the constraints by removing the directionality constraint of weights in the forward and backward paths.
- Compare the learning progress with differing degrees of bidirectionality in forward and backward paths.

Forward Propagation:

$$z^l = W^l h^{l-1} + b^l$$
$$h^l = \sigma(z^l)$$

In standard neural networks, **forward propagation** is the process of transforming a signal from input to output. In neural networks, this involves progressively transforming the signals over multiple layers of stacked affine transformation followed by an element-wise non-linearity.

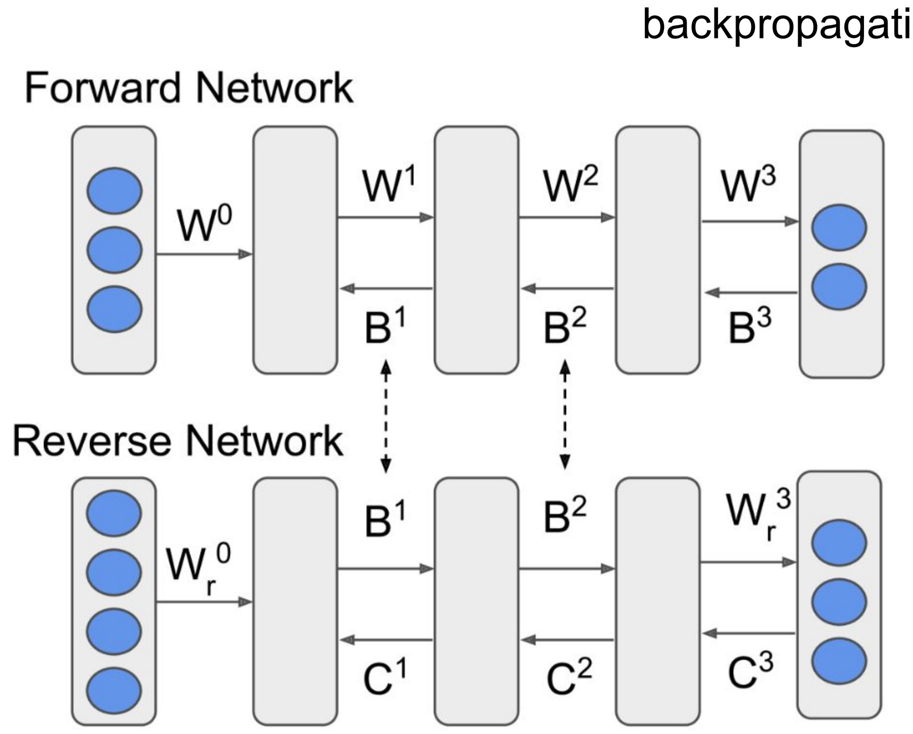
Standard Backpropagation:

$$\frac{\partial \mathcal{L}}{\partial W^l} = \frac{\partial \mathcal{L}}{\partial z^l} h^{(l-1)T}$$

Backpropagation is the process of computing partial derivative with respect to all parameters. In neural networks, this process involves transforming an error signal by the forward weights in reverse order from output to input.

$$\frac{\partial \mathcal{L}}{\partial z^l} = (W^{(l+1)T} \frac{\partial \mathcal{L}}{\partial z^{(l+1)}}) \odot \sigma'(z^l)$$

Synchrony between weights in forward propagation and those used backpropagation is known as the **weight transport problem**. **Feedback alignment** addresses this issue by using random fixed weights in backpropagation.



Feedback Alignment:

$$\frac{\partial \mathcal{L}}{\partial z^l} = (B^{(l+1)T} \frac{\partial \mathcal{L}}{\partial z^{(l+1)}}) \odot \sigma'(z^l)$$

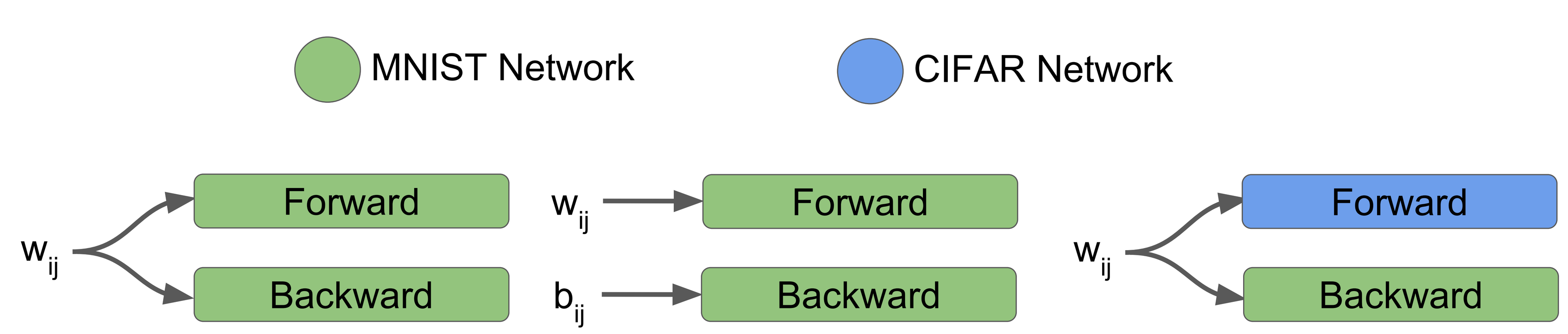
We show these random fixed weights need not be fixed. The same weights can be used in forward propagation of a different network. Hence we consider them **bidirectional**.

Bidirectional Feedback Alignment:

$$z_r^l = B^l h_r^{l-1} + b_r^l$$
$$h_r^l = \sigma(z_r^l)$$

No distinction between 'forward' and 'backward' weights

Bidirectionality removes the distinction of weights only used in the forward pass and those used in back pass. Bidirectional weights in a neural network can be used to rescale input signals and error signals without any impact to feedback alignment learning.



Accelerating feedback alignment

We found feedback alignment training can be substantially improved by **centering** the pre-activations across a layer. Centering can be viewed as a linear transformation a vector. Because it is symmetric, it is identical both forward and backward.

$$C = \begin{bmatrix} 1-\frac{1}{N} & -\frac{1}{N} & -\frac{1}{N} & \cdots & -\frac{1}{N} \\ -\frac{1}{N} & 1-\frac{1}{N} & -\frac{1}{N} & \cdots & -\frac{1}{N} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ -\frac{1}{N} & -\frac{1}{N} & -\frac{1}{N} & \cdots & 1-\frac{1}{N} \end{bmatrix}$$

$$z_{centered} = Cz$$

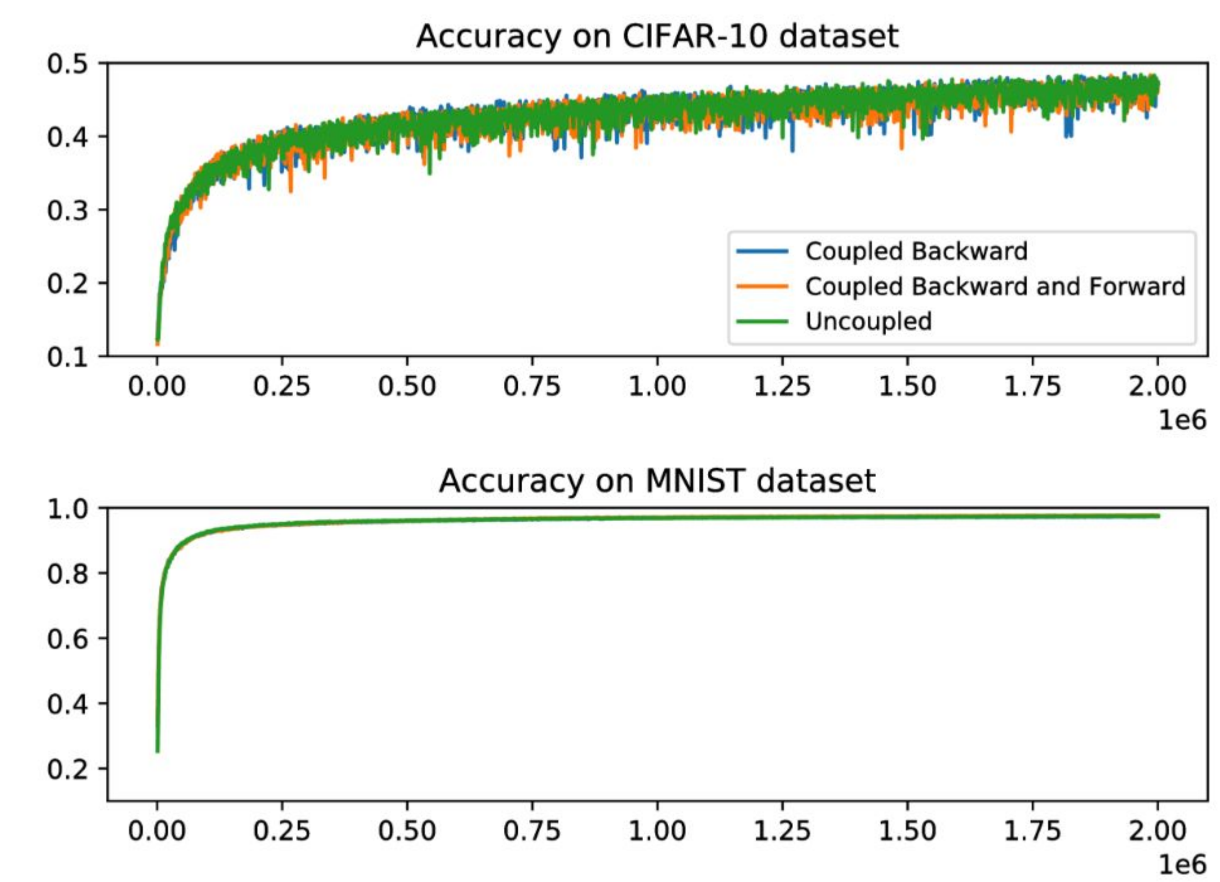
Experiments

We train two distinct 5 layer fully connected neural networks with 1024 units in each hidden layer. One network is trained on the CIFAR-10 task while the other is trained on the MNIST task. As weights for both networks are initialized randomly, the training process is identical to standard feedback alignment at the first iteration. Two types of bidirectionality are tested:

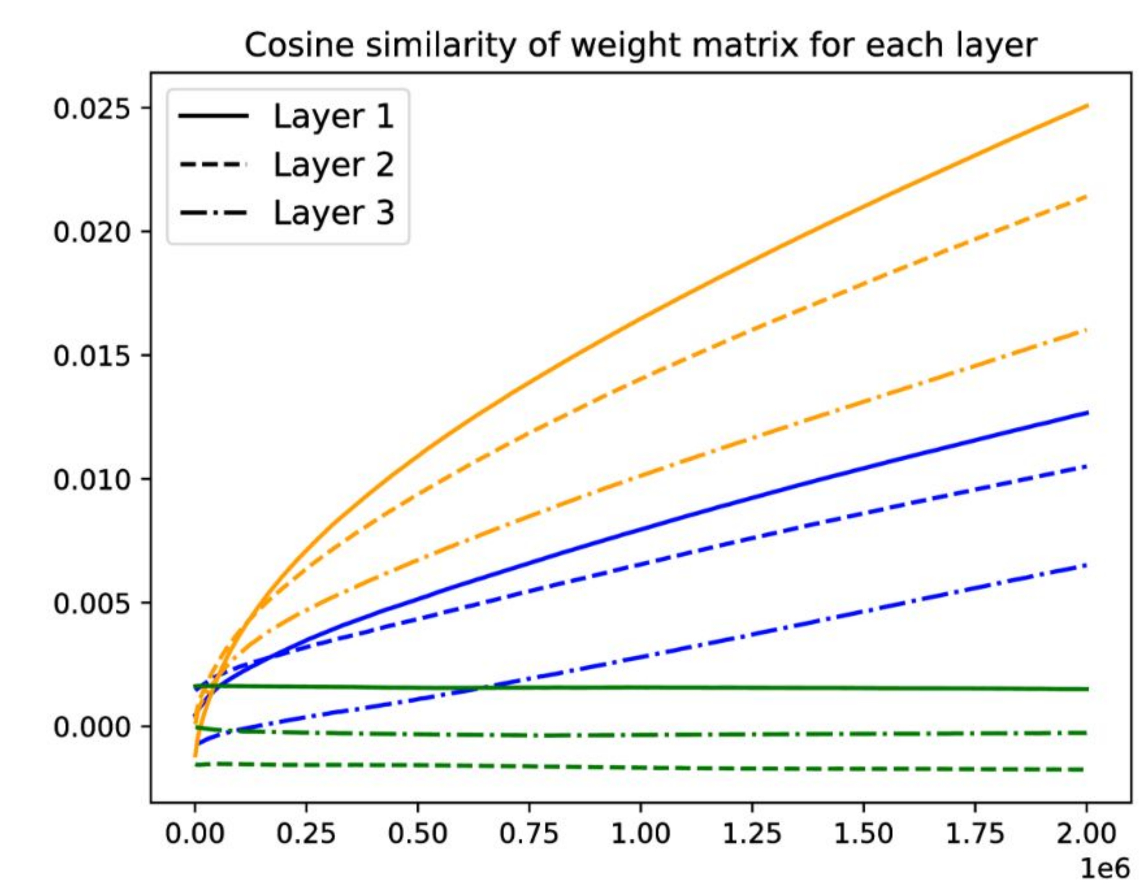
Coupled Forward: the backward weights B^l of the *forward network* (MNIST network) are not kept fixed throughout learning. These weights are updated by another feedback alignment procedure to train a *reverse network* (CIFAR network).

Coupled Backward and Forward: coupling the backward matrix of the reverse network to the forward network.

Results



Test accuracy during training of standard Feedback Alignment (green), Coupled Backward (blue), and Coupled Backward and Forward (orange) on the CIFAR-10 (top) and MNIST (bottom) datasets.



Cosine similarity between weights in the forward and reverse networks during training iterations for standard Feedback Alignment (green), Coupled Backward (blue), and Coupled Backward and Forward (orange).

- The coupling appears to have no impact on the training progress of feedback alignment. This is true for both the MNIST and CIFAR networks. The correlation between the forward and reverse networks grows as training progresses with coupling.
- We show preliminary results demonstrating that the feedforward and feedback pathways in a neural network can travel along the same connections.

Addendum: No distinction of update direction

The backward matrices in feedback alignment are sampled from a symmetric probability distribution. Therefore, the weight updates generated by feedback alignment are **independent of sign**.

- In backpropagation, the sign of the update determines whether the objective function is minimized or maximized.
- In feedback alignment, only the sign of the **last layer** update determines whether the objective function is minimized or maximized.
- Lower layers self-organize their updates based on the last layer

$$p(x) = p(-x)$$
$$\pm[B^{l+1}]_{ij} \sim p(x)$$
$$\delta_{FA}^l = \pm(B^{(l+1)T} \delta_{FA}^{l+1}) \odot \sigma'(z^l)$$
$$\Delta W^l = \pm\eta \delta_{FA}^l h^{(l-1)T}$$

References

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors." nature 323.6088 (1986): 533.
Lillicrap, Timothy P., et al. "Random synaptic feedback weights support error backpropagation for deep learning." *Nature communications* 7 (2016): 13276.
Frankle, Jonathan, and Michael Carbin. "The lottery ticket hypothesis: Training pruned neural networks." *arXiv preprint arXiv:1803.03635* (2018).