Activation alignment: exploring the use of approximate activity gradients in multilayer networks

Thomas Mesnard (thomas.mesnard@gmail.com)
Montreal Institute for Learning Algorithms, University of Montreal
Montreal H3T 1J4, Quebec, Canada

Blake Richards (blake.richards@utoronto.ca)
Learning in Neural Circuits Laboratory, University of Toronto Scarborough
Toronto M1C 1A4, Ontario, Canada
Learning in Machines and Brains Program
Canadian Institute for Advanced Research

Abstract
Thanks to the backpropagation-of-error algorithm, deep learning has significantly improved the state-of-the-art in various domains of machine learning. However, because backpropagation relies on assumptions that cannot be met in neuroscience, it is still unclear how similarly efficient algorithms for credit assignment in hierarchical networks could be implemented in the brain. In this paper, we look at one of the specific biologically implausible assumptions of backpropagation that hasn’t been solved yet: the need for a precise knowledge of the derivative of the forward activation in the backward pass. We show that by choosing a simple, drastic approximation of the true derivative, learning still performs well—even slightly better than standard backpropagation—and this approximation seems to play a regularization role. This approximation would also be much easier for real neurons to implement. Thus, this work brings us a step closer to understanding how the brain could perform credit assignment in deep structures.

Keywords: Backprop; Bio-plausible Algorithm; Feedback alignment;

Introduction
Recently, deep learning has revolutionized artificial intelligence and significantly improved the state-of-the-art in computer vision, speech recognition and many other domains. This success can be greatly attributed to the power of the backpropagation-of-error algorithm (Almeida, 1987; Pineda, 1987) (backprop) that enables efficient credit assignment in deep network structures.

Unfortunately, backprop algorithm relies on computations that are not feasible in a biological setup. Briefly, some of the main biologically implausible computations that backprop relies on are a result of the synaptic weight update proposed by backpropagation when optimizing a loss $L$ function:

$$W_i \leftarrow W_i + \alpha \Delta W_i$$

$$\Delta W_i \propto e_{i+1} W_{i+1}^T \sigma'(u_i) h_{i-1}^T$$

(1)

with $W_i$ being the weights between layer $i-1$ and layer $i$, $e_{i+1}$ being the error backpropagated from layer $i+1$, $u_i$ being the input to layer $i$ ($u_i = W_i h_{i-1} + b_i$, $\sigma'(\cdot)$ being the derivative of the forward activation function, $h_{i-1}^T$ being the transpose of the forward neuron activations in layer $i-1$, and $b_i$ being the bias for neurons in layer $i$.

The problem for neuroscience, is that three terms in this weight update carry some degree of biological implausibility:

- $e_{i+1}$ requires a separate error pathway to backpropagate the error without disrupting feedforward processing—there is no evidence in the brain for this form of segregated feedback pathway.

- $W_{i+1}^T$ requires feedback pathways for the error that have symmetric synapses to feedforward pathways—true synaptic symmetry is not guaranteed in a biological neural network.

- $\sigma'(u_i)$ requires a neuron to know the derivative of its output, which may not be easy to access, especially if the output involves spiking.

Recent work has attempted to address the biological implausibility of each of these terms.

With regards to the first item, one approach has been to use the power of multi-compartment neurons to remove the need for separate backwards pathways to calculate error (Sacramento, Costa, Bengio, & Senn, n.d.; Guerguiev, Lillicrap, & Richards, 2017). Alternatively, it is possible to use an energy based framework that enables the network to propagate the impact of a nudge towards the correct answer, rather than backpropagating an error explicitly (Bengio, Mesnard, Fischer, Zhang, & Wu, 2017; Mesnard, Gerstner, & Brea, 2016; Scellier & Bengio, 2017).

With regards to the second item, recent work has shown that by replacing the transpose of the forward weights (i.e $W_{i+1}^T$) by a fixed random matrix ($B_{i+1}$) for the backward computation, the algorithm can still learn to approximate the correct gradient (Lillicrap, Cownden, Tweed, & Akerman, 2016; Nøkland, 2016).

For the final item, a recent paper explored the use of an auxiliary function to replace $\sigma'(u_i)$, and demonstrated that it can still produce efficient credit assignment, even with spiking output (Zenke & Ganguli, 2018).
In this paper, we look more closely at the last item. Specifically, we explore what happens when one uses a highly simplified auxiliary function, rather than the derivative of the forward activation function. Unlike (Zenke & Ganguli, 2018), our goal here is not to make the algorithm work with spiking output. Rather, we want to examine how the use of a crude approximation of the derivative of the forward activation alters the relationship to true gradient descent. How is learning impacted? Is the gradient still being well approximated? Does the relation to the true gradient change over learning? These are the questions we address below.

**Model**

To describe our approach in simple terms, let's consider a multi-layer perceptron with input $x$, forward weight matrix $W_i$ between layer $i-1$ and $i$, $\sigma()$ the activation function, $h_i = \sigma(u_i)$ the activation of the neurons in layer $i$ with $u_i$ being equal to $W_i h_{i-1} + b_i$ with $b_i$ the bias in layer $i$. Let's consider here that $\sigma()$ is the sigmoid function and that our loss is the cross entropy. Let's recall that the weight update that backpropagation would give us is:

$$\Delta W_i \propto e^{i+1} W_{i+1}^T \sigma'(u_i) h_i^{T-1}$$

(2)

Instead of following the true gradient and updating the weights following eq. 2, we replace $\sigma'$ by $\sigma^*$ and therefore consider now the following weight update:

$$\Delta W_i \propto e^{i+1} W_{i+1}^T \sigma^*(u_i) h_i^{T-1}$$

(5)

**Results**

We trained a multi-layer perceptron with two hidden layers on MNIST by following the weight update described in eq. 5. We then compared the performance with the same network trained by following the true gradient and the weight updates proposed by the classical backpropagation in eq. 2. We used the cross entropy loss, 500 neurons per hidden layers, a learning rate equal to .1 and a batch size of 20.

Figure 1: Plot of the sigmoid function (black line), its true derivative (red line) and the crude approximation we will consider (blue line).

Figure 2 shows the evolution of the loss function when the network is trained with the true derivative (i.e red line) versus the crude approximation (i.e blue line).

Figure 2: Cross entropy loss during training when using the true derivative (i.e red line) versus the crude approximation (i.e blue line).
Figure 3 shows the test accuracy on MNIST during training when following eq.2 versus eq.5. As suggested by Figure 2, learning works well in both cases with faster convergence for the network trained with the approximation of the true derivative.

Finally, Figure 4 shows the angle between the update vectors prescribed by eq.5 and the one prescribed by classical backprop.

In contrast, the angle in the first hidden layer increases at the beginning of training, and interestingly, also converges towards 22°. Why would this occur? We saw, experimentally, that during training the forward weights scaled up such that the average of the pre-activation state (i.e. \( u_i \)) of the neurons in hidden layer \( i \) approximately matched the absolute value of the threshold that we selected in eq. 4 (i.e 2 in this case). This suggests that the forward weights are able to learn to scale up such that the input given to the crude derivative is approximately in the range where the derivative changes and not in the middle of a flat plateau. Further experiments are required, but this behavior bears some resemblance to the feedback alignment mechanism described in (Lillicrap et al., 2016).

**Conclusion**

In this article, we examined the use of a very simple auxiliary function in place of the true derivative of the forward activation function for backpropagation. We found that using such an approximation actually seemed to improve learning, potentially helping with regularization. This brings us a step closer to a biologically realistic framework (Zenke & Ganguli, 2018), thereby helping us to understand how the brain could do credit assignment. Our results are promising, and even suggest that such an approximation might be beneficial in biological networks. Future experiments will extend these results to other activation functions and to deeper neural networks.

**References**


