Improving Random Feedback Alignment with Transfer Learning

A Preprint

Gabi Kim
Mechatronics Engineering
University of Waterloo
gljkim@uwaterloo.ca

Andrej Kukuruzovic
Computational Mathematics
University of Waterloo
akukuruz@uwaterloo.ca

Josh Loong
Science and Business
University of Waterloo
jtloong@uwaterloo.ca

May 7, 2020

Abstract

The brain uses multiple methods that allow us to learn very quickly and efficiently. In comparison, modern machine learning is usually domain and task specific. Backpropagation is a popular method in machine learning used to achieve to goals, but it does not have basis in biology as a model for the brain, due to it's backward connections. A method known as random feedback alignment was proposed as a more biologically plausible alternative to backpropagation for modelling the brain, but lacks its power. We combine the machine learning concept of transfer learning, which was born from biological principles, along with random feedback alignment. We use a new method relating to angles to improve how we implement random feedback alignment and show that we can achieve better accuracy than what is originally shown in random feedback alignment. We show that using improved angles and transfer learning leads to nearly identical results that would come from using backpropagation with transfer learning concepts. Our results hope to reopen ideas on how we may better improve modelling of the human brain.

1 Introduction

A neural network using backpropagation to mimic the flow of signals from neuron to neuron through synapses in the human brain is not biologically possible as it requires the synapses of the fully connected neurons to propagate an error signal backwards by using the weights calculated from the forward pass (see Figure 1). It was proposed by Lillicrap et al. that an algorithm known as random feedback alignment (RFA) could be more biologically plausible and achieve similar results to backpropagation [1]. RFA works in the same manner as backpropagation except the error signal does not depend on weights of the forward pass but a random matrix B to propagate backwards (see Figure 2).

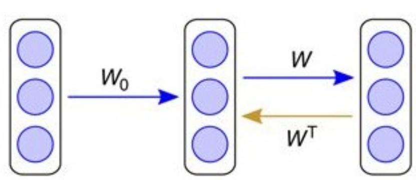

Figure 1: Error signal in backpropagation.

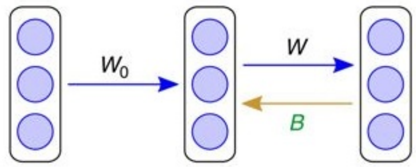

Figure 2: Error signal in random feedback alignment.

We use another biologically plausible theory known as the transfer of learning, which is the process of our past experiences affecting our new ones. This means that when learning a new task, we draw on past knowledge to help learn that new task. In deep learning, this is an emerging area of research known as transfer learning.

Our goal is to see if we are able to combine the ideas of RFA and transfer learning to validate a biological process of learning tasks in neural networks. Past research has shown that RFA is biologically possible and works just as well as backpropagation in datasets such as MNIST [1] but fails in more complex classifications when using CIFAR-100 [2]. This means that while it may make sense biologically, it may not be a fitting model for mimicking the brain as it cannot perform well on more complex datasets. By adding the concept of transfer learning we hope to achieve better learning of more complex tasks with RFA.

2 Methods

The experiment is executed in three steps. First, we trained the initial network on distinguishing hand-written numbers (MNIST dataset). We then proceeded to process the weights from the learned task to be transferred over to another network to learn a second task. The second task categorizes the extension of MNIST known as EMNIST, which is hand-written letters. Finally, we tested to see how well RFA and backpropagation learned on EMNIST using the knowledge from the MNIST dataset.

2.1 Training Initial Network

The initial network we created was built using the backpropagation method, as we knew that we could quickly achieve strong classifications results for multiple datasets. Not only could it classify well, but using this method allowed us to reuse code and be more efficient since backpropagation and RFA are almost identical in implementation. Multiple files were created with the initial(W_1) and final (W_2) layer with different levels of accuracy. This was done to see how well we needed to tune our initial network to get good results on later tests.

2.2 Processing Weights

With the thought that we could learn new tasks by using existing knowledge from previous tasks, we wanted to preserve as much information as possible from the different weight layers in the initial network. In RFA the matrix B is a random matrix, so we'd lose information from its initialization. The method used to tackle this problem was the use of principal component analysis (PCA)[3]. This was used so we would reduce the dimensionality of the transferred layers, while keeping critical components of our data in a reduced framework. Once there, we would build our random matrix B with that the PCA method returned.

2.3 Testing

The most important aspect of this section was to determine which random matrix B was used for the error signal after using PCA. It was originally purposed that the teaching signal Be must be within 90° of the teaching signal used in backpropagation $W^T e$ [1], as it would push Be in the same direction as $W^T e$. The idea was to use the convexity in angles form the 2-d vector space, where we look to optimize the equation

$$e^T W B e > 0 \tag{1}$$

In $\mathbf{R}^2$ equation 1 tells us that the vectors are within 90° of each other. This concept is not completely present in matrices, so we decided to use their concept of angles and extend it further. As the product of $e^T W B e$ approaches 0, it should imply that the angle is closer to 90°, so we decided to try and push product of the teaching signal to a larger value. Figure 3 shows the difference of degrees in how we simulated our angles for the experiment.

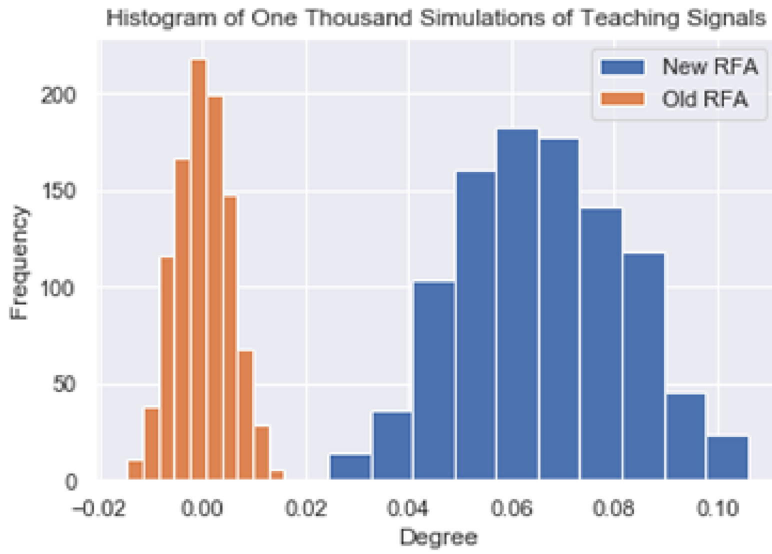

Figure 3: Stimulated degrees of old and new teaching signals in random feedback alignment(RFA)

3 Results

In the initial testing accuracy converged somewhere between 95% - 96% for using the backpropagation method on the MNIST dataset (see Figure 4. The most important thing to note about this network is that the weights being transferred did not seem to depend on what level of accuracy they achieved as long as it was a reasonable value. This meant that transferring the weights with 80% accuracy on the network would yield the same results as the weights with 96% accuracy. The only time results the results were affected is when we used weights that had fairly low levels of accuracy such as being lower than 60%.

Figure 4: Backpropagation on MNIST dataset

We needed a benchmark to compare how well RFA would perform, so we tested how well backpropagation would work on the EMNIST dataset. We ran the simulation with a normal distribution of weights and using transferred weights from the initial network. Comparing the two ways of initializing our weights, we see in Figure 5 that both simulations converge together and seem to learn at approximately the same rate.

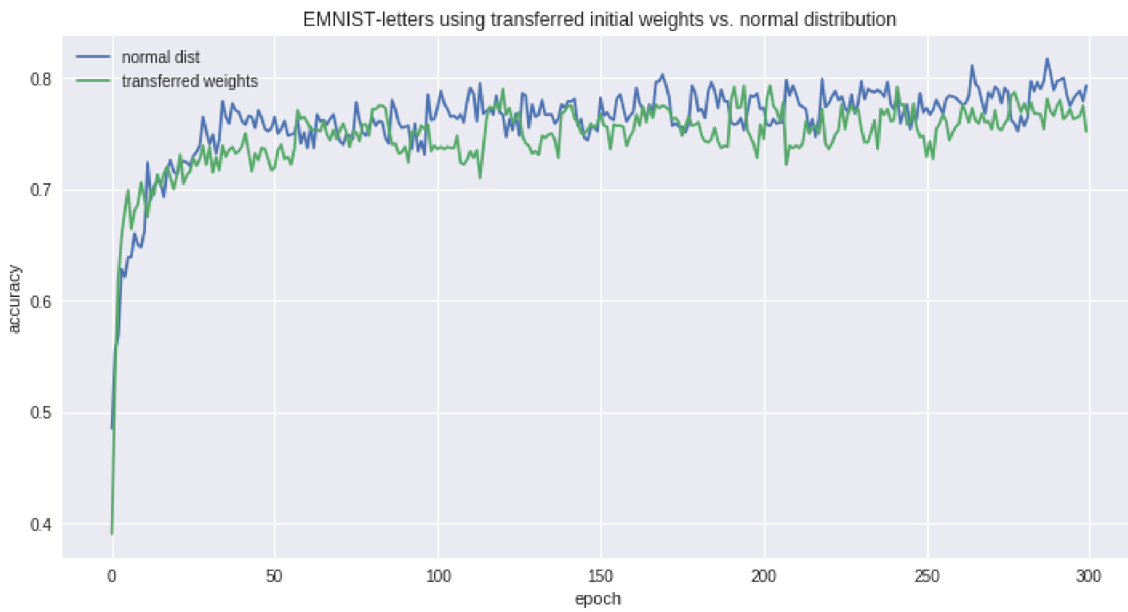

Figure 5: Backpropagation on all 26 letters from EMNIST dataset using first layer

In Figure 5, we noticed that regardless of how we initialized our weights, it resulted in a convergence of roughly 80%. This pointed to the fact that the network might not be strong enough to properly classify all 26 letters. Since backpropagation is more powerful than RFA, we chose to change our process. We then decided to run similar tests using backpropagation but with only 10 letters from the EMNIST dataset instead of 26.

To conduct the part of the experiment we wanted to see which combination of layers (i.e. $W1$ and $W2$) from the initial network would have a more positive effect on the results and see if moving down to 10 letters yield a stronger network. The combination of both layers being transferred resulted in the weakest performance and was worse than our simulations on all 26 letters. The bottom layer of the initial network ($W2$) performed much better than the combination of the two layers (see Figure 6). Using the first layer of the initial network had the top performance equally shared with just using normally distributed weights. Our assumption was that the final layer would yield better results than the first layer, since the final layer is more important in backpropagation. The result was similar from what we saw in the simulations for all 26 letters, except having a higher accuracy of 90% instead of 80%. We expected this to happen since the output units were the same length in both the MNIST dataset and the 10 letter EMNIST dataset.

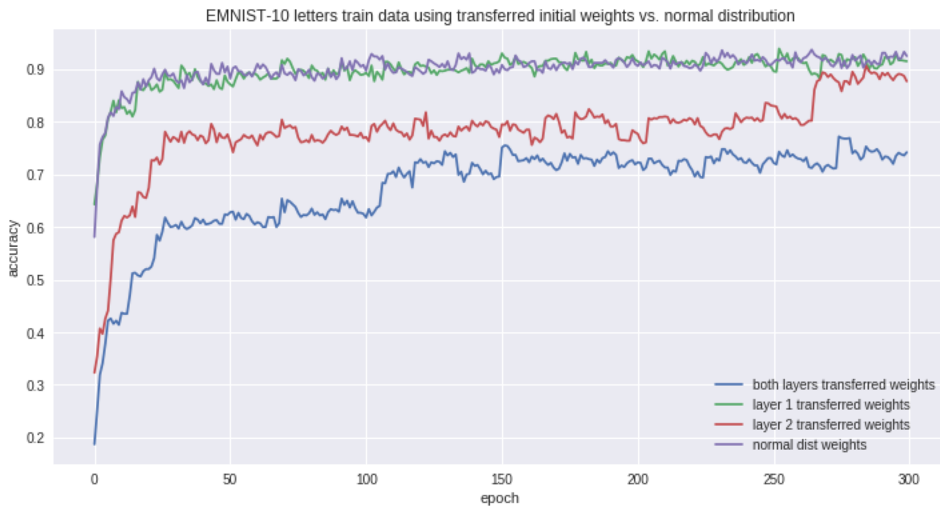

Figure 6: Backpropagation on 10 letters from EMNIST dataset

Results showed that using our method for initializing matrix, B, resulted in consistently higher accuracy through all epochs on the network (see Figure 7). The old method resulted in accuracy under 96%, while our method was able to above 96% in a faster manner.

We found that we saw a similar result to what happened when re-training on the MNIST dataset. Our method was consistently learning faster than the old method for RFA (see Figure 8). Note that this performed as well as our best performances on the 10 letter EMNIST dataset with backpropagation in figure 6. The old method with transferred weights would fall slightly short, but our new method it is equally as successful.

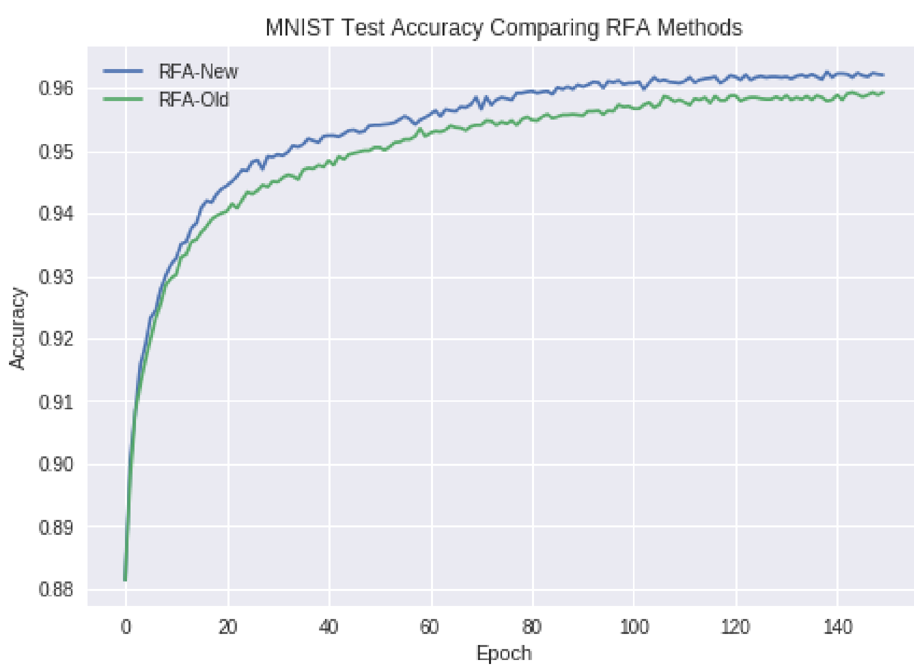

Figure 7: Custom RFA vs Old RFA on MNIST dataset

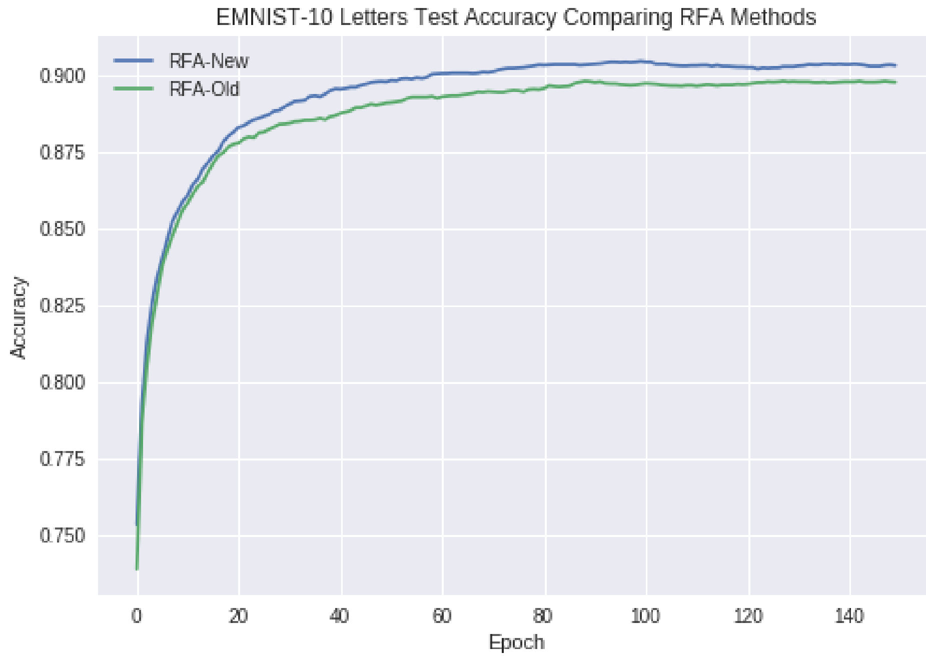

Figure 8: Custom RFA vs Old RFA on EMNIST dataset

4 Discussion

Our method for improving RFA with transferred weights showed to be working on both MNIST and 10 letter EMNIST dataset. It consistently yielded more accurate results when compared to the method of using RFA (see Figure 8). The idea of increasing the product $e^T WBe$ to reduce angle between the teaching signals $W^T e$ and Be makes sense in $\mathbf{R}^2$ but we weren't sure if it would translate on RFA.

While encouraging, the tests were only conducted on datasets with the same length of output units. When trying to test backpropagation with the transferred weights on the full EMNIST dataset, we were not able to get high levels of accuracy. Further investigation would lead us towards looking for a better method of processing our initial weights for networks with different output units. Without testing our method for RFA on the 26 letter EMNIST dataset, we limited ourselves by not checking how similar its performance would be when compared to backpropagation. Future research would lead us to test on more complex dataseta, such using any of the CIFAR datasets. Furthermore, we would take more time to study how we could optimizing the angle between the teaching signals to make them go in as close to the same direction as possible.

References

[1] Cownden D. Tweed D. B. Akerman C. J. Lillicrap, T. P. Random synaptic feedback weights support error backpropagation for deep learning. *Science*, 2016.

[2] Arild Nø kland. Direct feedback alignment provides learning in deep neural networks. *Science*, pages 1037–1045, 2016.

[3] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.