Neural Network : [The underlying mechanisms of alignment in error backpropagation through arbitrary weights]

Kimia Haji Sadeghian[a], **Aria Koureshy**[a], **and Mohammad Mohammad Beigi**[a]

[a]Student, Department, Sharif University of Technology

This manuscript was compiled on June 24, 2023

This article discusses how the brain is able to change and adapt using its millions of synapses. Researchers use artificial neural networks (ANNs) to understand how the brain works, but creating models that accurately represent the biological processes is hard. Backpropagation (BP) is a common learning algorithm for ANNs but it requires symmetric backward connectivity patterns that are not found in the brain. Previous research suggests that an approach called BP-TAW, which uses arbitrary weights, works similarly to BP through a process called "Feedback Alignment".

We examine the mathematical basis of Feedback Alignment by analyzing the statistical properties of neural activity like cross-correlation and autocorrelation. They show that there is a similarity between the update directions of BP and Feedback Alignment. Furthermore, we find that normalizing the weight matrices can significantly improve Feedback Alignment.

Keyword 1 | Keyword 2 | Keyword 3 | ...

Backpropagation is the most successful algorithm for training artificial neural networks, (Rumelhart et al., 1985) but there are gaps between it and learning in biologically plausible neuronal networks in the brain. Some gaps include lack of local plasticity and autonomy in backpropagation (Stork, 1989; Crick, 1989; Song et al., 2020).

While natural and artificial learning mechanisms may differ, there are remarkable similarities in the behavior of neurons in the brain and artificial neurons that have been trained using backpropagation (BP). These similarities suggest that the BP algorithm may be effective in closely approximating the learning processes that occur in the brain (Zipser and Andersen, 1988; Khaligh-Razavi and Kriegeskorte, 2014; Cadieu et al., 2014; Cichy et al., 2016; Nayebi et al., 2018; Whittington and Bogacz, 2019, 2017; Lillicrap et al., 2020; Xie and Seung, 2003;).

As we stated in the previous paragraphs, the backpropagation algorithm used in machine learning assigns blame for errors by multiplying error signals with weights on each neuron's axon and further downstream. However, this requires a precise and symmetric backward connectivity pattern that may not exist in the brain. So in Lillicrap et al., 2016 it is demonstrated that this architectural constraint is not necessary for effective error propagation. Instead, they propose a simple mechanism that assigns blame by multiplying errors by random synaptic weights, which can transmit teaching signals across multiple layers of neurons and perform as effectively as backpropagation on various tasks(BackPropagation Through Arbitrary Weights (BP-TAW) method).

The direct feedback alignment (DFA) learning algorithm has been shown to enable learning even when errors are transmitted directly from the output layer to each hidden layer through arbitrary backward weights. In DFA, the backward weights do not need to be symmetric or precise, unlike backpropagation. This algorithm simplifies the architecture of neural networks by removing the need for precise and symmetric backward connections across all layers, which is a constraint that has been considered impossible to achieve in the brain. The results demonstrate that error propagation can happen effectively through arbitrary backward weights, a finding that challenges previous assumptions about the required architectural constraints for effective learning and signal propagation in neural networks (Nøkland, 2016; Refinetti et al., 2020; Frenkel et al., 2019; Launay et al., 2019; Baldi et al., 2018;).

Although BP-TAW has been an effective algorithm for training artificial neural networks (ANNs), the specific mathematical principles that contribute to its success are not yet fully known. Researchers have studied weight alignment (WA) in the context of linear networks to try to understand the conditions under which it occurs. These findings may provide insights into the mechanisms behind weight alignment in more complex network architectures(Lillicrap et al., 2016; Frenkel et al., 2019).

Here we investigate the impact of weight matrices on weight alignment (WA) in arbitrary backpropagation, known as BP-TAW. It is found that the original version of BP-TAW leads to a continuous growth in the norms of weight matrices, which can damage WA. However, we show that the WA can be improved by restricting the norms of the weights. Generalizing the mathematical properties of WA, we show that neural activity's statistical nature, such as the cross-correlation and autocorrelation of error and output signals, contributes to the occurrence of weight alignment.

Then, we apply BP-TAW to a specific problem of training a nonlinear 5-layer ANN on the MNIST dataset and demonstrate that the alignment of relative similarity of data points belonging to the same group compared to distinct groups is affected by cross-correlated neural activity.

Additionally, we compare the forward weight trajectories of BP and BP-TAW using a low-dimensional embedding method and show that BP-TAW's local minimum is less favorable than BP in terms of convergence speed, but the performance of BP-TAW improves after reducing the Frobenius norms of the input weights of each neuron.

Please provide details of author contributions here.

[1]A.O.(Author One) and A.T. (Author Two) contributed equally to this work (remove if not applicable).

Materials

Data set. The MNIST dataset, a large database of handwritten digits, is used to study the statistical properties of output and error signals in ANN models, and it is shown that the relative similarity of data points of a single category and their differences across categories, which is an intrinsic property of datasets, contributes to weight alignment by shaping cross-correlated neural activity.

After loading the dataset we split it into training and testing sets. The training set contains 60,000 images, and the testing set contains 10,000 images. Next, the labels in the training set are shuffled randomly to create a different dataset, which can be used to test the model's robustness to label shuffling. After that, the images are resized to 15x15 pixels. Resizing the images reduces the computational cost of training the model without losing significant information. Finally, the pixel values of the images are normalized by dividing them by 255, which scales the pixel values between 0 and 1. This step helps to improve the convergence of the model during training and is a common preprocessing step in deep learning.

Methods and Results

Formulas. Weights and biases in BP are updated as below:

$$W_l[k+1] = W_l[k] + \Delta W_l[k]$$

$$b_l[k+1] = b_l[k] + \Delta b_l[k]$$

$$B_{l,BP}[k+1] = W_{l,BP}[k+1]^T$$

$$\Delta W_l[k] = \eta L_l[k]^T \delta_{l+1}[k], 0 \leq l < d$$

$$\Delta b_l[k] = \eta \times ones_{1\times n_b} \delta_l[k], 0 < l \leq d$$

Error matrices of neurons are:

$$\delta_{d,BP}[k] = E[k] \odot f'(Z_d[k])$$

$$\delta_{l,BP}[k] = \delta_{l+1,BP}[k] W_l[k]^T \odot f'(Z_d[k]), 0 < l \leq d$$

$$loss = \frac{1}{2}\sum_{i,j} E[k]^2_{i,j} = \frac{1}{2}\sum_{i,j}(Y^*[k]_{i,j} - Y[k]_{i,j})^2$$

(Rumelhart et al., 1985)

In BP-TAW just $W_l[k]^T$ is replaced with B_l(constant arbitrary matrices different from forward weights).(Lillicrap et al., 2016)

Angle and cosine similarity between two matrices:

$$W \angle B = cos^{-1}\left(\frac{\langle W, B\rangle_F}{||W||_F ||B||_F}\right)$$

Cosine similarity is calculated as below:

$$cos(W \angle B) = \frac{\langle W, B\rangle_F}{||W||_F ||B||_F}$$

Results Description. The aim of this study is to provide a mathematical and statistical analysis basis of weight alignment (WA) in neural networks, which refers to the phenomenon where the updates to the weights during training are aligned with the updates that would be obtained using a simpler, feedforward algorithm and investigation of the factors that contribute to alignment and demonstrate how alignment terms can be used to analyze the behavior of neural networks under various conditions. We also discuss the limitations of

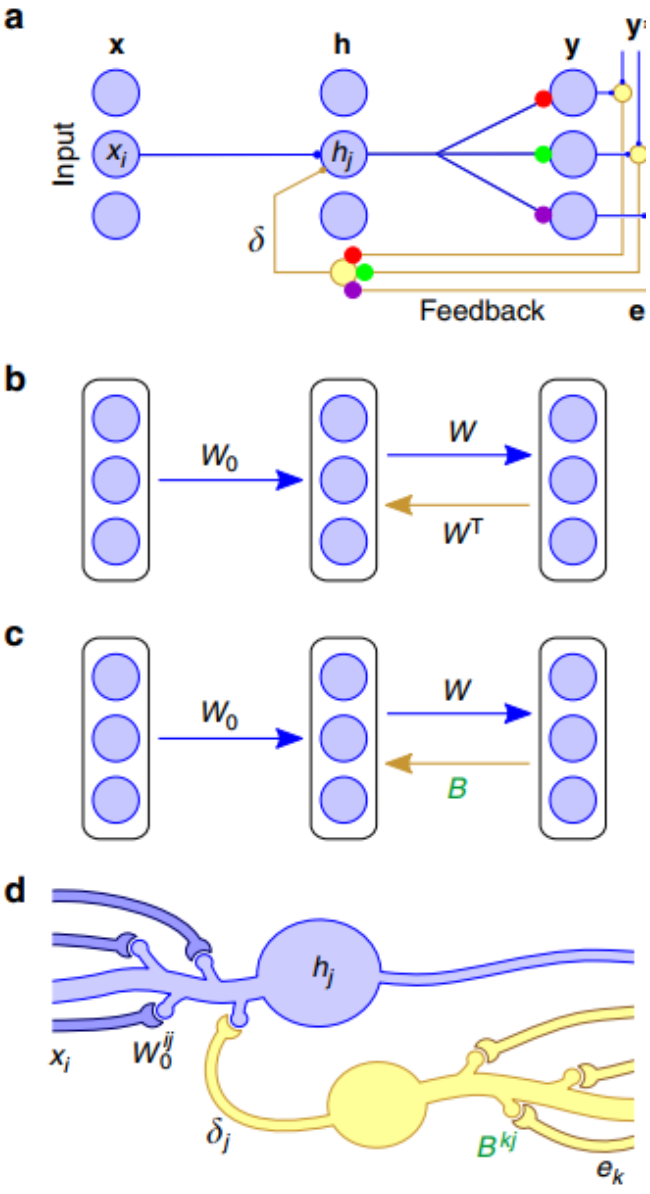


Fig. 1. Figure 1 (Lillicrap et al., 2016)shows a diagram explaining how random feedback weights can deliver useful teaching signals. It compares the backprop learning algorithm with the feedback alignment method. The backprop algorithm requires neurons to know each other's synaptic weights, while feedback alignment replaces the transpose of the forward weight matrix with a matrix of fixed random weights. This way, each neuron in the hidden layer receives a random projection of the error vector. The diagram also shows a potential synaptic circuitry underlying feedback alignment, but it is provided for illustrative purposes, and there are many possible configurations that could support learning with feedback alignment.

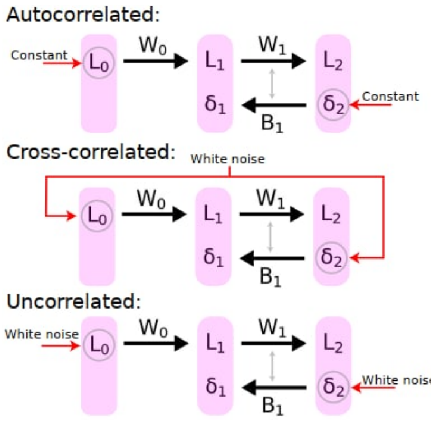


Fig. 2. Figure 2 shows open-loop nonlinear 2-layer ANN in different modes(Alireza Rahmansetayesh , Ali Ghazizadeh1, , Farokh Marvasti 2022)

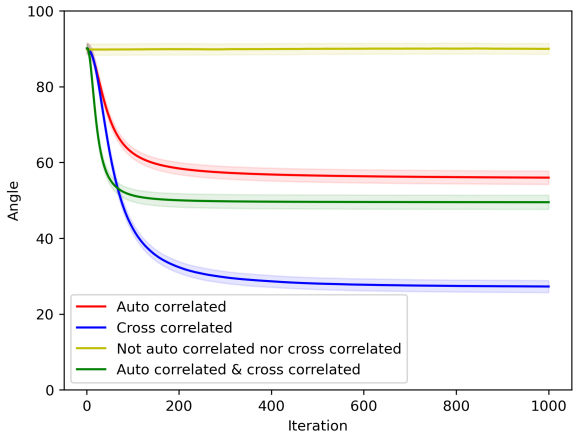


Fig. 3. Here we depict the degree of coherence between the forward and backward weights in four distinct situations. Coherence emerges when either self-correlation, cross-correlation, or both are present but not when they are absent. The chart outlines the mean outcome for some(order of 40) iterations in each corresponding situation, with the shaded regions indicating a standard deviation around the average.

backpropagation with target propagation and weight transport algorithms.

We defined a simple artificial neural network (ANN) with two layers (one hidden layer and one output layer) that uses the ReLU activation function. The network is trained in an open loop setting, where the output of the network is not fed back into the input. The algorithm initializes the weights of the network randomly using a normal distribution with mean 0 and standard deviation 1. It then trains the network using stochastic gradient descent (SGD) with a learning rate of 0.002 and a batch size of 100 for a total of 2000 iterations. The input layer has 20 neurons, the hidden layer has 100 neurons, and the output layer has 20 neurons. During training, the ReLU activation function is applied to the output of the hidden layer.

Then we calculate the angles between the forward weight of the output layer and the transpose of the backward weight of the output layer at each iteration. 10 epochs are preformed, where in each epoch, the weights of the network are re-initialized and the training process is repeated for 2000 iterations.

The angle between these weights can be used as a measure of how much the weights have changed during the training process. If the angle is small, it means that the weights are changing slowly and the network is converging. If the angle is large, it means that the weights are changing rapidly and the network may not be converging. By monitoring the angle between the weights over time, we can get a sense of how well the network is training and adjust the learning rate or other hyperparameters as needed. This can help ensure that the network converges to a good solution and avoids overfitting or other issues.

The difference between "cross-correlated" and "auto-correlated" networks depends on how the error signal is computed. In an auto-correlated network, the error signal is computed based on the difference between the network output and a delayed version of the network output. In a cross-correlated network, the error signal is computed based on the difference between the network output and the target output, and it is then cross-correlated with the inputs of the network to compute the error signal for the hidden layers.

Here are the scenarios of these three models showing in the **Fig 2**:

1. In the first model inputs and errors are auto-correlated, so L_0 and δ_0 are two constants which have been generated from a normal distribution.
2. In the second model inputs and errors are cross-correlated, so at each iteration $L_0 = \delta_0$ is generated from a normal distribution.
3. And in third model inputs and errors are uncorrelated, so at each iteration L0 and δ_0 are generated from a normal distribution independently.

Note that L_0 stands for hypothetical output signals which are imposed on the neurons in the input layer and δ_0 stands for hypothetical error signals of neurons in the output layer.

Fig 3 shows the autocorrelation and cross-correlation of the error and output signals of neurons in the network. Autocorrelation refers to the correlation of a signal with a delayed version of itself, while cross-correlation refers to the correlation of two different signals. The figure shows that the autocorrelation of error and output signals of neurons and the cross-correlation between them are two important features of neural activity contributing to alignment.

We use the backpropagation-through-time with weight transport (BP-TAW) algorithm as a specific example of a feedforward algorithm that exhibits WA. BP-TAW is a variant of the backpropagation algorithm used in training artificial neural networks. In BP-TAW, the weights of the network are updated based on the product of the error signal and the activation values of the neurons, rather than the product of the error signal and the output values of the neurons as in traditional backpropagation. This approach has been shown to be biologically plausible and has been used to explain certain phenomena observed in the brain. BP-TAW has also been shown to have some advantages over traditional backpropagation in terms of convergence speed and robustness to noise.

We show that the structure of the alignment terms in BP-TAW, which can be extracted from the update rule, is responsible for the occurrence of alignment. The authors also investigate the statistical properties of neural activity that contribute to alignment. They show that the autocorrelation

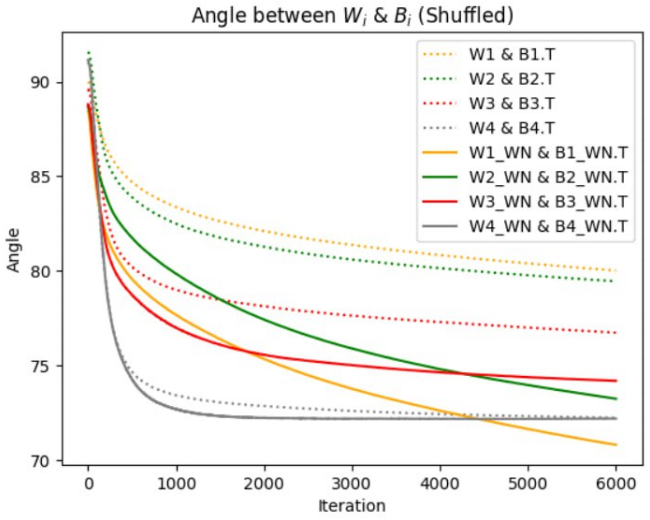


Fig. 4. Figure 4 shows that the alignment angle between the update directions of BP-TAW and BP for different layers of the neural network with No WN and WN, as a function of the number of epochs. The alignment angle decreases over epochs for all layers, indicating that the alignment between BP-TAW and BP decreases as the training progresses. This is due to the fact that the error signal becomes smaller as the network learns, leading to smaller alignment terms and less alignment between BP-TAW and BP.

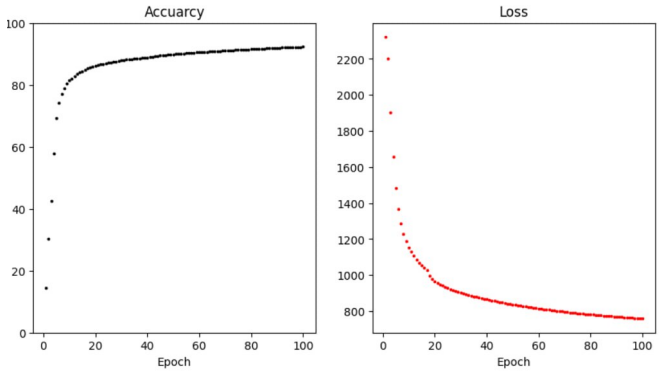


Fig. 5. Feedback alignment accuracy and loss during epochs

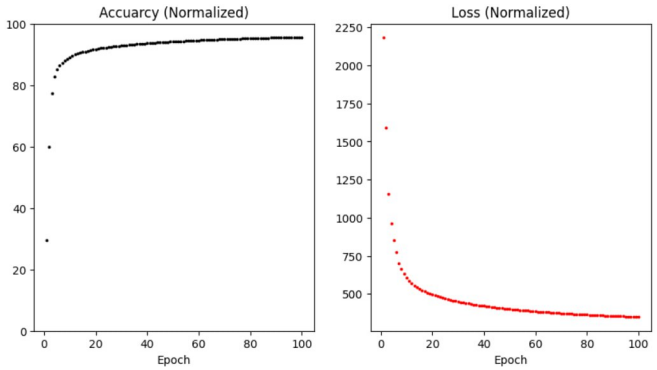


Fig. 6. Feedback alignment accuracy and loss during epochs when WN is applied

of error and output signals of neurons, as well as the cross-correlation between them, are important features of neural activity that contribute to alignment. In their analysis, the authors make some simplifying assumptions and approximations, such as using first-order Taylor approximations to extract alignment terms and ignoring the effect of nonlinearity on them. However, they show that these approximations lead to qualitatively valid predictions about WA.

Fig 4 is a plot that shows the angle between the backward weights and the forward weights of a neural network during training. The plot is useful for analyzing the effect of weight normalization on the angle between the weights during training. Weight normalization is a technique used to scale the weights of a neural network layer to have unit norm to improve the convergence of the network during training and reduce the sensitivity of the network to the initial values of the weights. By comparing the plots with and without weight normalization, we can observe the effect of this technique on the angle between the weights during training.

In Feedback Alignment (FA), the accuracy and loss depend on the quality of the feedback weights. The feedback weights are initialized randomly, and their quality is improved through learning. During the learning process, the feedback weights are adjusted to minimize the loss/error between the predicted output and the target output. The quality of the feedback weights may affect the convergence speed and stability of the FA algorithm, which may affect the accuracy and loss outcomes. However, studies have shown that with appropriately chosen feedback weights, FA can achieve comparable or even better accuracy and loss results than traditional backpropagation algorithms.

As shown in **Fig 5** and **Fig 6** loss when we have applied WN, loss has decreased and accuracy has reached to an acceptable level faster.

Fig 7 shows the angle between the alignment terms and the backward weights of a neural network during training. By comparing the plots for different values of k, we can observe the effect of this initialization technique on the alignment between these weights during training. Here we compare the effect of weight initialization on the angle between these weights. It consists of four subplots, each showing the angle between the alignment terms and the backward weights for a specific layer of the neural network (W1 to W4). The x-axis represents the number of batches during training, and the y-axis represents the average angle between the alignment terms and the backward weights for each iteration.

It includes two lines for each subplot, representing the angle between the alignment terms and the backward weights for two different values of k (k=66 and k=1260).

Fig 8 states that the update directions of Back-Propagation with Targeted-Relaxation Weights (BP-TAW) and traditional Back-Propagation (BP) are initially in alignment, but this alignment decreases as we move to earlier layers of the network. At the fourth layer, the update directions of BP-TAW and BP become identical. To smooth out the data, every trace is passed through a moving average with a window length of 60.

Fig 9 evaluates the alignment of the real forward weights versus an altered form of the forward weights while undergoing the learning process. Even though non-linearity leads to a gradual reduction in alignment over time, it doesn't entirely block alignment from occurring. we charted the alignment progression of both versions of weights in a step-by-step fashion to juxtapose them.

In **Fig 10** a comparison is made between the alignment

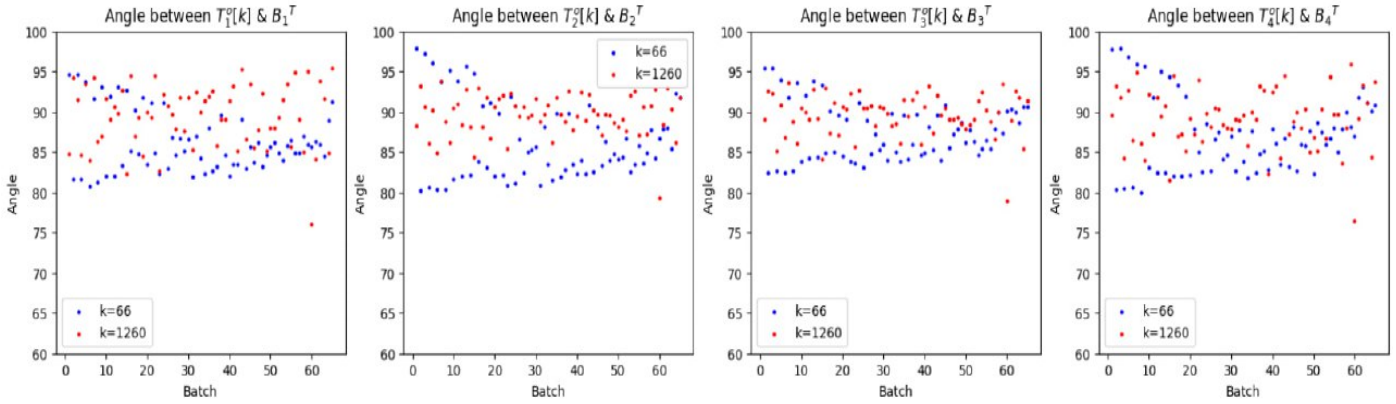


Fig. 7. .

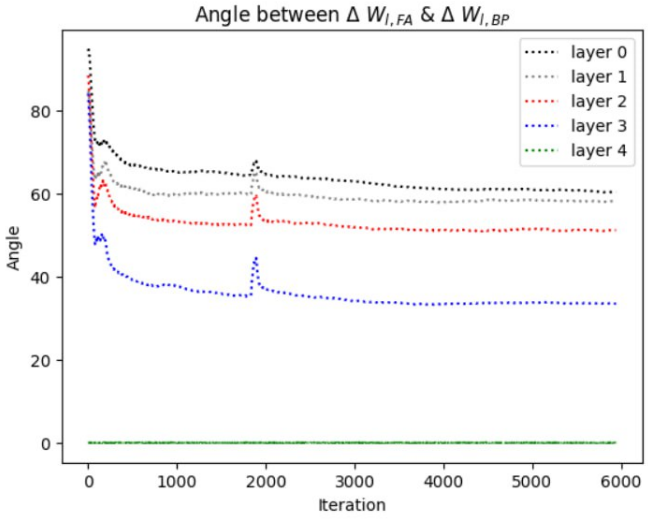
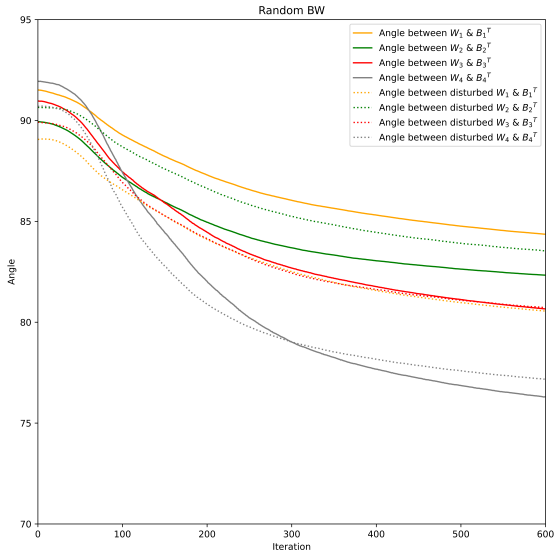


Fig. 8. Angle between ΔW in backpropagation and feedback alignment



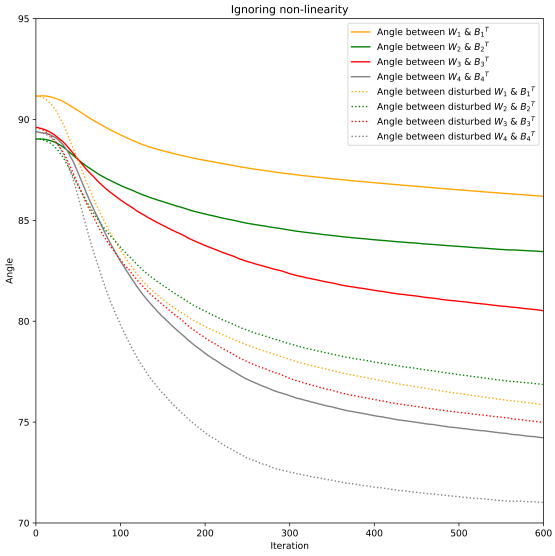


Fig. 9. Comparing the angle between W and B^T while ignoring non-linearity

Fig. 10. Comparing the angle between W and B^T while having random backward weights

of the forward and backward weights in their actual form and the alignment of the disturbed version of the forward weights with the corresponding newly generated backward weights over multiple iterations. In order to simulate this process, a disturbed version of the weight updating formula is used to calculate a version of the forward weights that is also affected by new random backward weights. Despite the distortion caused by this process, the alignment still occurs over the iterations, although the amount of alignment is affected.

Discussion

A. The fundamental mathematical principles that underlie feedback alignment (FA). We detailed weight alignment (WA) in Artificial Neural Networks (ANNs) and how it's not just a result of the learning process or reducing the loss function. Rather, WA relies on alignment terms extracted from the update rule of Backpropagation using arbitrary weights (BP-TAW). The neural activity's statistical properties, specifically the autocorrelation of error and output signals of neurons and

the cross-correlation between them, contribute to weight alignment. We used alignment terms to analyze the behavior of BP-TAW on a nonlinear 5-layer ANN trained on the MNIST dataset.

Then we explored how the arrangement of data within mini-batches and the repeated occurrence of data points over various epochs shape the autocorrelated neural activity and contribute to weight alignment. Additionally, the similarity and differences between data points in the same category affect the cross-correlated neural activity and, therefore, contribute to weight alignment. Despite making some approximations and simplifying assumptions in our analysis, such as using a first-order Taylor approximation and not considering nonlinearity, we showed that these led to valid predictions about weight alignment.

B. Applying weight normalization techniques can lead to an increase in alignment. Our research investigated a weight normalization (WN) method that keeps the Frobenius norm of input weights to each neuron (γ) constant. We discovered that applying this method greatly increased alignment and improved the classification accuracy of the neural network.

We also noted that heterosynaptic plasticity and other types of plasticity operate synapses under competitive conditions, meaning that if one synapse is enhanced, other synapses may be lessened to ensure the overall strength of synapses remains balanced. The competitive effect we observed is similar to the result of the WN scheme we studied.

C. Approximation of BP in biological networks and the additional limitations that make its biological implementation implausible. We have discussed the feasibility of Feedback Alignment (FA) for various neural activities. One of the limitations of Backpropagation (BP) is the weight transport problem, which is resolved by BP-TAW. However, BP and BP-TAW have other biological limitations.

Biological networks can approximate FA through synaptic plasticity. There are two scenarios suggested for FA in the brain: the first involves a separate network of neurons for propagating error as feedback weights, which is biologically unlikely; the second scenario proposes the usage of the same neurons as in the forward path for backpropagating the error through feedback axons, which is more feasible biologically.

In BP-TAW, backward weights are constant. However, synaptic plasticity applies to both forward and backward weights in the brain, allowing simultaneous propagation of forward and backward weights.

References

1. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

2. Stork, D. G. (1989). Is backpropagation biologically plausible. In International Joint Conference on Neural Networks, volume 2, pages 241246. IEEE Washington, DC.

3. Crick, F. (1989). The recent excitement about neural networks. Nature, 337(6203):129132. .

4. Song, Y., Lukasiewicz, T., Xu, Z., and Bogacz, R. (2020). Can the brain do backpropagation?exact implementation of backpropagation in predictive coding networks. NeuRIPS Proceedings 2020, 33(2020).

5. Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatiotemporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Scientific reports, 6(1):113.

6. Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., DiCarlo, J. J., and Yamins, D. L. (2018). Taskdriven convolutional recurrent models of the visual system. In Advances in Neural Information Processing Systems, pages 52905301.

7. Whittington, J. C. and Bogacz, R. (2019). Theories of error back-propagation in the brain. Trends in cognitive sciences, 23(3):235250.

8. Xie, X. and Seung, H. S. (2003). Equivalence of backpropagation and contrastive hebbian learning in a layered network. Neural computation, 15(2):441454.

9. Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020). Backpropagation and the brain. Nature Reviews Neuroscience, pages 112.

10. Zipser, D. and Andersen, R. A. (1988). A backpropagation programmed network that simulates response properties of a subset of posterior parietal neurons. Nature, 331(6158):679684.

11. Zipser, D. and Andersen, R. A. (1988). A backpropagation programmed network that simulates response properties of a subset of posterior parietal neurons. Nature, 331(6158):679684.

12. Cadieu, C. F., Hong, H., Yamins, D. L., Pinto, N., Ardila, D., Solomon, E. A., Majaj, N. J., and DiCarlo, J. J. (2014). Deep neural networks rival the representation of primate it cortex for core visual object recognition. PLoS Comput Biol, 10(12):e1003963.

13. Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. Nature communications, 7(1):110

14. Nøkland, A. (2016). Direct feedback alignment provides learning in deep neural networks. arXiv preprint arXiv:1609.01596.

15. Refinetti, M., dAscoli, S., Ohana, R., and Goldt, S. (2020). The dynamics of learning with feedback alignment. arXiv preprint arXiv:2011.12428.

16. Frenkel, C., Lefebvre, M., and Bol, D. (2019). Learning without feedback: Direct random target projection as a feedbackalignment algorithm with layerwise feedforward training. stat, 1050:3.

17. Launay, J., Poli, I., and Krzakala, F. (2019). Principled training of neural networks with direct feedback alignment. arXiv preprint arXiv:1906.04554.

18. Baldi, P., Sadowski, P., and Lu, Z. (2018). Learning in the machine: Random backpropagation and the deep learning channel. Artificial intelligence, 260:135.

19. The underlying mechanisms of alignment in error backpropagation through arbitrary weights Alireza Rahmansetayesh , Ali Ghazizadeh1, , Farokh Marvasti