Biologically Plausible Learning using GAIT-prop Scales to ImageNet

Anonymous authors
Paper under double-blind review

Abstract

Many of the recent advances in the field of artificial intelligence have been fueled by the highly successful backpropagation of error (BP) algorithm, which efficiently solves the credit assignment problem in artificial neural networks. However, it is unlikely that BP is implemented in its usual form within biological neural networks, because of its reliance on non-local information in propagating error gradients. Since biological neural networks are capable of highly efficient learning and responses from BP trained models can be related to neural responses, it seems reasonable that a biologically viable approximation of BP underlies synaptic plasticity in the brain. Gradient Adjusted Incremental Target Propagation (GAIT-prop) has recently been derived directly from BP and has been shown to successfully train networks in a more biologically plausible manner. However, so far, GAIT-prop has only been shown to work on relatively low-dimensional problems, such as handwritten-digit recognition. This work addresses some of the scaling issues in GAIT-prop and shows it to achieve performance comparable to BP on the much more challenging CIFAR10 and ImageNet datasets.

1 Introduction

Backpropagation of error (BP) is a learning algorithm that solves the credit assignment problem in deep neural networks, allowing for the formation of task-relevant internal representations and high performance in applications (Rumelhart, Hinton, and RJ, 1986; LeCun, Bengio, and Hinton, 2015; Schmidhuber, 2014). Despite its efficacy, the default construction of BP does not appear a likely candidate for the computational steps involved in the learning algorithm of real neural systems. Early criticisms of BP's biological plausibility have already been put forward by Grossberg (1987) and Crick (1989) in the late 1980's. A recent review by Lillicrap et al. (2020) provides a modern summary of the mechanisms that make BP biologically implausible. These include issues of symmetric synaptic weight matrices, error propagation machinery and more. In particular, these issues all pertain to the use of non-local information for the propagation and computation of error signals to update individual synaptic connections deep within a network. Therefore, biologically plausible learning algorithms are required to provide sensible methods for the assignment of credit deep within a neural network model using local-only information.

Several lines of previous work have attempted to address the biological implausibilities inherent to BP. Lillicrap et al. (2016) report the counter-intuitive result that a fixed random matrix of feedback weights can support learning in neural networks. This learning is presumed to take place as a result of the forward weight matrices adapting themselves to act as local pseudoinverses of the feedback matrices around the error manifold. This approach is called Feedback Alignment (FA) and it does away with the necessity of symmetric weight matrices. Direct Feedback Alignment (DFA) was proposed by Nø kland (2016) and works by connecting random feedback weights from the output layer directly to each hidden layer. This approach has been used to train modern deep learning architectures (Launay et al., 2020), but its performance compared to regular FA on challenging datasets actually seems to be slightly worse (Bartunov et al., 2018). This may have to do with the relatively simplistic way of performing credit assignment, which does not fully exploit the expressiveness of multi-layer architectures.

Guerguiev et al. (2017) propose a multi-compartment neuronal model that is able to handle feedforward and feedback neural activity separately, drawing inspiration from neuroscience, specifically pyramidal neurons and their distal and apical dendrites. This resolves the implausibility of requiring symmetric feedback weights. Because learning occurs through local updates and requires no separate feedback pathways, the weight transport problem is also addressed. It is shown that networks built with these more complex neurons can perform multi-layer credit assignment, as evidenced by the fact that adding a hidden layer to the network leads to better performance on the MNIST dataset.

Predictive coding (Whittington & Bogacz, 2017), alternatively, handles error signals by assuming the existence of separate error neurons for each value neuron. These neurons compute differences between predicted and observed activations and these prediction errors can be used to inform local weight updates. The presence of an error neuron for each value neuron, however, is a rather strong assumption without any neurobiological evidence supporting it. Ideally, a biologically plausible algorithm would learn directly from changes in activities of neurons.

An algorithm which aims to do just that is target propagation (TP) (Bengio, 2014; Lee, Zhang, Fischer, and Bengio, 2015). The key principle is to combine a desired network output with an (approximate) inverse model of a network in order to produce desired outputs for each layer of a network. These computed layer-wise target activities can then be taken as local supervised labels and used for learning. TP addresses the weight transport problem, because it uses only local activity targets to achieve multi-layer learning. However, there is no explicit theoretical relationship between TP and BP. Furthermore, it has been shown that the efficacy of TP in deep networks that are trained to solve difficult tasks is questionable (Bartunov et al., 2018).

Bartunov et al. (2018) show that biologically plausible algorithms like TP, FA and DFA perform reasonably well, though still worse than BP, on MNIST (Deng, 2012) and CIFAR10 (Krizhevsky, 2012) when applied to fully connected architectures. However, in locally connected architectures and on the ImageNet (Russakovsky et al., 2014) dataset, a significant performance gap appears between all biologically inspired algorithms and BP. Particularly, TP and its variations fail to learn almost completely on ImageNet. FA learns the task to an extent, but its final top-1 test accuracy is about one fourth that of BP.

A common shortcoming among the above approaches is that no theoretical guarantees are made that their updates closely approximate the error gradient as computed by BP. Biologically plausible approaches tend to use local learning rules to avoid the weight transport problem. These local targets often provide useful feedback information in relatively shallow networks, but when multi-layer credit assignment is required, compounding errors drive the updates away from those computed by BP. This failure to accurately track the error gradient likely explains why no biologically plausible learning algorithm has so far managed to scale to ImageNet and remain competitive with BP in terms of performance. Scaling biologically plausible learning to real world problems calls for a learning rule that works based on local information but somehow still guarantees a close correspondence to BP.

Though TP in its original formulation does not track BP's weight updates very accurately, Ahmad et al. (2020) recently found that there exists a direct correspondence between target-based learning and BP, though this relationship only exists locally and under specific network constraints. This method is referred to as Gradient Adjusted Incremental Target Propagation (GAIT-prop). GAIT-prop was derived specifically to maintain exact correspondence to BP. This was accomplished by computing those layer-wise targets which would produce weight updates equivalent to BP. These updates also happen to take a biologically plausible form under some constraints. In practice, GAIT-prop was shown to match the performance of BP on the MNIST, FMNIST and KMNIST datasets for relatively shallow networks. However, deeper networks and more challenging tasks were not explored.

The current paper shows that GAIT-prop in its standard form can diverge from BP due to issues of precision and proposes a solution to these issues. In particular, we propose a layer-wise target normalisation procedure that stabilizes learning with GAIT-prop in deeper networks. We also describe an easy to implement procedure for inverting convolutional neural network layers in order to efficiently extend GAIT-prop's utility to problems like the ImageNet classification task (Russakovsky et al., 2014).

2 Methods

2.1 Target propagation and GAIT propagation

GAIT-prop is based on the idea of target propagation (Bengio, 2014; Lee et al., 2015). The key idea is that layer-wise activity targets, rather than error gradients, are propagated backwards through the network. The difference between the current activity and target activity acts as a local error signal to compute weight updates. An inverse mapping from layer l to layer $l-1$ can be used to determine which activation vector in layer $l-1$ would have produced this more desirable activation in layer l, which then becomes the target for layer $l-1$, and so on. As discussed, though elegant, this method does not produce weight updates that are similar to those produced by BP. The first reason why TP yields different weight updates from BP, is simply because TP uses either learned or exact matrix inverses to propagate its activity targets, while BP uses the transpose of the forward matrix to propagate its errors backwards. For non-orthogonal weight matrices, the inverse differs from the transpose. Furthermore, TP does not account for local gradients when perturbing the current activities in the direction of target activities.

GAIT-prop seeks to address both of these problems. In order to ensure that matrix inverses produce the same transformation as matrix transposes, it constrains weight matrices to be orthogonal. This is achieved by orthogonal initialization of weights, followed by regularization during each weight update. GAIT-prop also accounts for local gradients by adjusting its activity perturbations, multiplying them by the square of the local gradient. Under the constraint of square orthogonal weight matrices, GAIT-prop follows the gradient as computed by BP arbitrarily closely and performs competitively with BP on datasets on the scale of MNIST. In this paper, we find that when scaling up GAIT-prop to more complex datasets, and thus larger networks, numerical instability issues occur. The resolution of this scaling issue is the first major subject of this paper.

Let us define the function F as the forward-pass mapping between layers in a neural network such that for layer $l+1$,

$$a_{l+1} = F_l\left(a_l\right) = f\left(W_l a_l + b_l\right) \tag{1}$$

where a_l is the activation vector, W_l is the weight matrix and b_l is a bias vector for layer l. The function $f(\cdot)$ represents an activation function. In TP, layer-wise targets, t^{tp} are propagated backwards from layer l to layer $l-1$, by taking the targets of layer l and applying a (learned) inverse function to them:

$$t_{l-1}^{\text{tp}} = G_l\left(t_l^{\text{tp}}\right) \tag{2}$$

where G_l is the (learned) inverse mapping from layer l to layer $l-1$ so that with a perfect inverse mapping, $G_l(F_l(a_l)) = a_l$. Exact inverses require an equal number of neurons in each layer, however, this constraint can be relaxed by using auxiliary units which can be interpreted as a form of perceptual memory Ahmad et al. (2020). In this work (and in the original GAIT-prop work), the inverse function G_l is defined exactly such that

$$a_l = F_l^{-1}\left(a_{l+1}\right) = G_l\left(a_{l+1}\right) = {W_l}^{-1}(f^{-1}\left(a_{l+1}\right) - b_l) \tag{3}$$

In order to enable this exact inversion, weight matrices are square and initialized as orthogonal and the Leaky-ReLU activation function (invertible for all real-valued outputs) was used.

GAIT-prop proposes a modification to the targets being inverted. The inverted targets were modified such that they constitute a small perturbation from the forward-pass activity toward the target activity. The perturbation is also multiplied by the square of the activation function derivative at the forward-pass activity. Given this definition, the GAIT-prop targets, t^{gp}, are computed after a transformation such that:

$$t_{l-1}^{\text{gp}} = G_{l-1}\left(\left(1-\epsilon_l\right)\ a_l + \epsilon_l\ t_l^{\text{gp}}\right) \tag{4}$$

where $\epsilon_l = \gamma_l D\left(a_l\right)^2$ is a perturbation parameter with γ_l a constant with very small magnitude and $D\left(a_l\right)$ a diagonal square matrix with the derivatives $\frac{dF_l}{da_l}$ of the forward mapping F_l on its main diagonal. This target adjustment is constructed in order to obtain updates equivalent to BP under the condition of orthogonal weight matrices Ahmad et al. (2020).

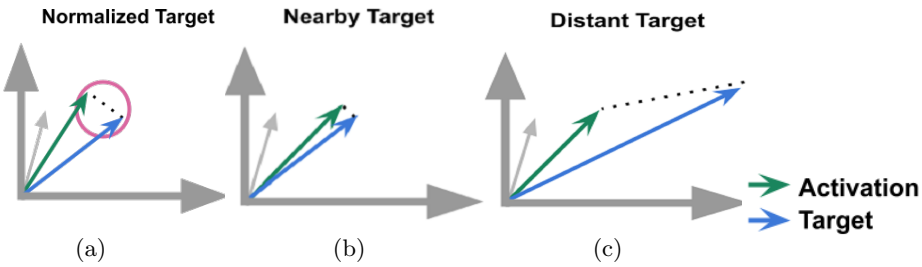

Figure 1: A depiction of the activation and target vectors. a) Targets and activations are normalized to be at a specific distance. b) Targets and activations become too close such that floating point precision issues occur. c) Targets and activations diverge, weakening the correspondence between GAIT-prop and BP.

2.2 Scaling GAIT-prop to ImageNet

Scaling GAIT-prop to problems like ImageNet requires us to address the numerical instability that arises in GAIT-prop when it is applied to deeper networks, as described in the next section. In addition, it requires the implementation of local connectivity to make parameter estimation feasible. Recall that it was the application of deep convolutional neural networks (CNNs) to the ImageNet challenge in 2012 that sparked the revival of neural networks for image classification (Krizhevsky et al., 2012). To our knowledge, there are no practical methods for achieving high performance on ImageNet without using local connectivity. To address these challenges, we propose a method of normalizing target-activation distances and a method to utilize an invertible convolution operation using GAIT-prop.

2.2.1 Normalized targets

We find empirically that when inverting targets deep in a network, precision errors in the inversion can occur when the target versus activity difference becomes very small. Repeated inversion with fixed layer-wise incremental factors γ_l can result in a net increase or decrease (explosion or vanishing) of the target-activation distance, which is the key error term used for learning. To alleviate the problem of target-activation differences becoming susceptible to precision errors (or increasing beyond the limits of our linear approximation), we propose a local normalization of the incremental factors γ_l based upon the target-activation difference (Figure 1). For each layer independently, we divide the incremental factor by the L2-norm of the target-activation difference γ_l. This ensures that our 'distance to target' is fixed on a layer-wise basis, ensuring robust error propagation. Specifically, the value of the incremental factor for layer l is computed as

$$\gamma_l = \frac{\eta}{\|a_l - t_l^{\text{gp}}\|^2} \tag{5}$$

where η is a normalization constant determining the desired Euclidean distance between targets and activations.

2.2.2 Application of GAIT-prop to CNN architectures

GAIT-prop requires network layers to be invertible. Because inverting large sparse matrices in memory is prohibitively expensive, we introduce a way to invert convolutional layers in a patch by patch manner, greatly reducing computational cost. The output of a convolutional layer is inverted by running a transposed convolution that takes as its input the layer's output and as its weight matrix the same kernel as used in the forward pass, but inverted. For these four-dimensional kernels to be inverted, they first need to be reshaped to be square, then transposed, then inverted, then reshaped back to their original format. See Figure 2 for a graphical depiction. Note that if these kernels were always perfectly orthogonal, simply taking the transpose would be equivalent to inversion. However, since our regularizer does not ensure perfect orthogonality, we take the exact inverse.

The above method correctly inverts a convolutional layer's output only in the case where receptive fields do not overlap. When receptive fields do overlap, those parts of the input image that are in the scope of

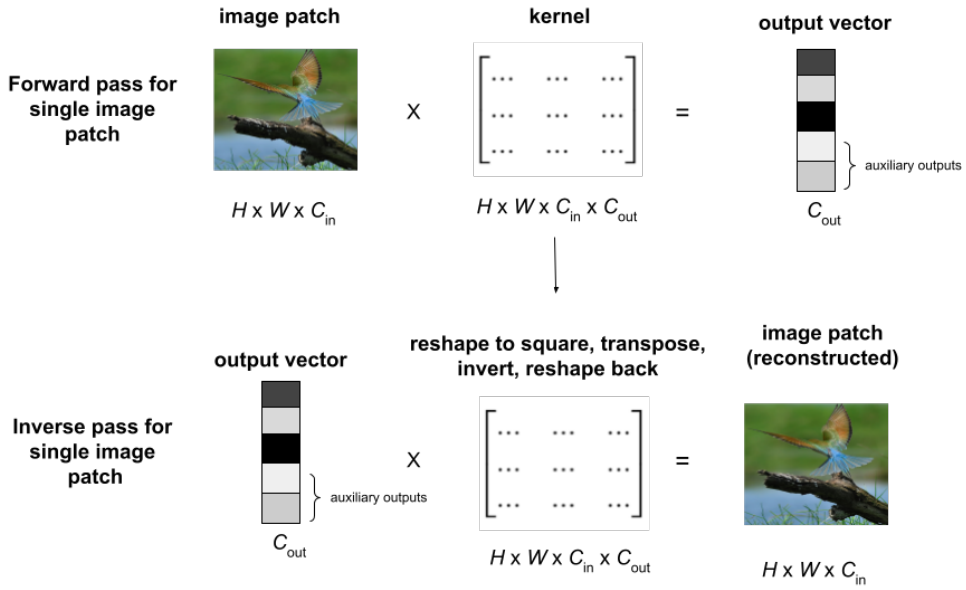

Figure 2: The process of invertible convolution illustrated for a single image patch. The convolution operation results in an output vector for each image patch. This output vector can contain auxiliary units to satisfy the requirement that weight matrices need to be square (and invertible). To invert an image patch, the kernel is reshaped to be square, inverted and then reshaped back to its $H \times W \times C_{\text{in}} \times C_{\text{out}}$ format. Finally, a transposed convolution is run using this inverted filter, restoring the input image

multiple receptive fields will be reconstructed multiple times, requiring a renormalization. An easy way to accomplish this, is to divide each pixel in the reconstruction by the number of receptive fields it is a part of. This will correctly reconstruct any input image.

However, this process will also divide error information flowing through the pixel by the same number, causing a reduction in the error signal. To remedy this in the context of GAIT-prop, we use a different method: the input reconstruction is corrected by removing from it the forward signal (rather than feedback signal) equivalent to the number of reconstructions (minus one). This allows the layer to reconstruct its input exactly once. Importantly, any error information coming from higher layers will be preserved in this way such that error information is fed back and integrated from multiple overlapping receptive fields.

2.2.3 Additional constraints

Exact inverses can only be computed for full-rank square matrices in which there is no loss of dimensionality. This means that, at minimum, the number of output channels of a filter must be equal to the input dimensionality, that is, the number of output channels must be equivalent to the kernel height times kernel width times number of input channels ($C_{\text{out}} = C_{\text{in}} \times H \times W$).

In order to achieve a change in network width, auxiliary output channels can be used (akin to the auxilliary units used by Ahmad et al. (Ahmad et al., 2020)). This amounts to slicing off a number of channels from the output of a convolutional layer, before feeding the result into the next layer. When inverting the convolutional layer, its entire output, including the auxiliary units, should be used as input to the inversion operation. The auxiliary outputs should therefore not be discarded after the forward pass, but kept in memory.

2.3 Network Models and Training

To investigate the above two proposed improvements, together referred to as normalized GAIT-prop, we trained and tested several deep neural network models with the CIFAR10 and ImageNet datasets. We trained two fully connected networks on CIFAR10 using either BP, FA, GAIT-prop or normalized GAIT-prop; one

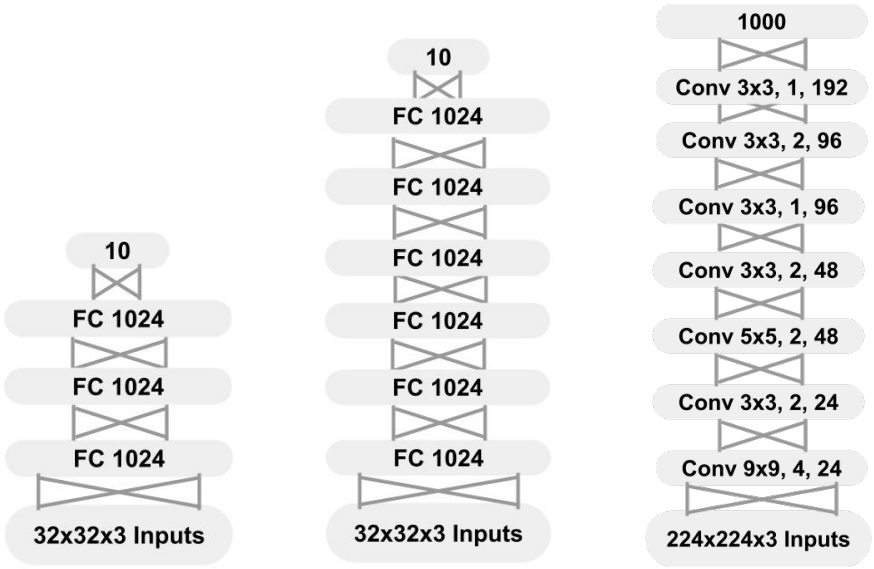

Figure 3: Architectures used to train on CIFAR10 shallow (left), CIFAR10 deep (middle) and ImageNet (right). Fully connected layers are indicated with FC, convolutional layers with Conv. For FC layers, the number indicates the number of units. For Conv layers, the numbers represent the kernel size, stride and output channels, respectively.

shallow (three hidden layers) and one deep (six hidden layers). We further trained a convolutional network on ImageNet using only normalized GAIT-prop, FA or BP. FA was chosen as an additional bio-plausible benchmark because it performed the best of all bio-plausible algorithms in Bartunov et al. (2018), though it still lagged significantly behind BP. The convolutional architecture and hyper parameters were also adapted from Bartunov et al. (2018). Due to memory constraints, we reduced the number of filters by half when training ImageNet. This likely impacts performance, however we desired to investigate whether GAIT-prop can perform competitively with BP on a large dataset in a deep convolutional network, rather than to maximize peak performance.

During training, parameters were updated for all methods using the Adam optimizer (Kingma & Ba, 2014). In addition, when using GAIT-prop, a regularizer was used to encourage orthogonality of weight matrices. See Ahmad et al. (2020) for details. In addition to being regularized to be orthogonal, all weight matrices were also initialized to be orthogonal, to ensure efficient training from the start of the experiment. Other performance enhancing techniques like batch normalization, L2-regularization and dropout were not used. All network layers implemented the leaky-ReLU activation function and categorical cross-entropy was used as a loss function.

3 Results

3.1 CIFAR10 results for a shallow network

The correspondence between weight updates from different algorithms was measured by treating each update as a one dimensional vector and measuring the angle between them. Figure 4A and B show the angles between (normalized) GAIT-prop's and FA's weight updates and those of BP when training a relatively shallow fully connected network on CIFAR10. The angle between the updates fluctuates around one degree for (normalized) GAIT-prop, indicating a very high degree of correspondence between the algorithms. FA's updates show angles between 60 and 80 for all non-output layers. Note that a random update would produce a 90 degree angle in expectation. Figure 4A shows that the angle is zero for the output layer for all algorithms. This is expected, as the updates are by definition identical for that layer. Updates in earlier layers in the network, which are arrived at after more iterations of GAIT-prop or FA, diverge very slightly in the case of

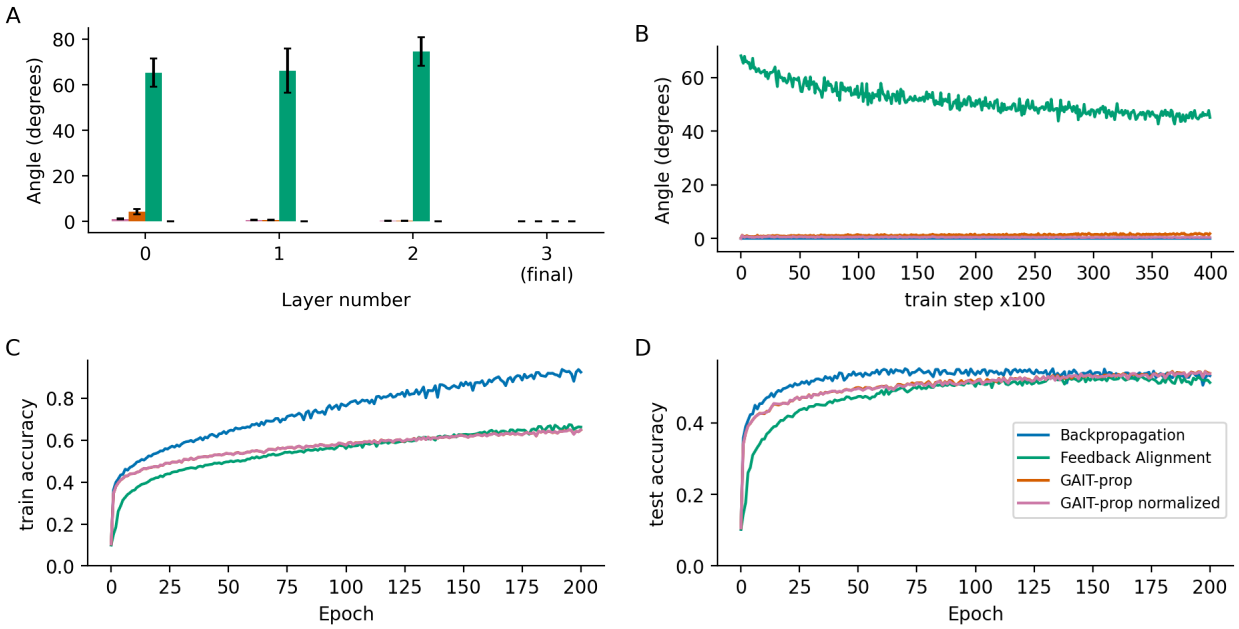

Figure 4: Performance of BP, GAIT-prop, normalized GAIT-prop and FA on CIFAR10 for a fully connected network with 3 hidden layers. For GAIT-prop and FA, correspondence to BP's updates are displayed. A) Angles between GAIT-prop's and FA's weight update vectors and BP's per layer. B) Angles in degrees between GAIT-prop's and FA's updates and BP's weight updates per train step. C) Train accuracy BP, GAIT-prop and FA per epoch. D) Test accuracy for BP, GAIT-prop and FA per epoch.

GAIT-prop and significantly in the case of FA. They are also almost identical in their performance, as can be seen in Figure 4C and D. Note that the lines for GAIT-prop (orange) and normalized GAIT-prop (pink) overlap almost perfectly, though both are present in the figure. There seems to be no performance gap in test performance. In terms of train accuracy, BP does seem to converge more quickly than the other algorithms. As this performance gap is only seen in train accuracy, it might be due to BP overfitting more on the train data, possibly due to the lack of a regularizer being used for BP.

3.2 CIFAR10 results for a deep network

Figure 5 shows results for the same experimental setup, except with a deeper network, consisting of six hidden layers. As can be seen in Figure 5A and B, updates for standard GAIT-prop quickly diverge from BP's updates, with updates for normalized GAIT-prop maintaining their correspondence to BP for all layers. In terms of performance, standard GAIT-prop shows poor performance in this setting, while normalized GAIT-prop remains competitive with BP. FA outperforms standard GAIT-prop in this deeper network, but lags behind both BP and normalized GAIT-prop.

3.3 Deep convolutional networks

On the ImageNet dataset, like on CIFAR10, normalized GAIT-prop performs competitively with BP, see Figure 6. FA almost completely fails to learn in this experiment, showing some learning only in the earliest epochs of training, followed by a decrease in performance as training continues. Standard GAIT-prop was not included in the ImageNet experiment, as it failed to learn well even on the much simpler CIFAR10 task.

Angles between GAIT-prop's and BP's updates are again consistently low, though higher than in the CIFAR10 experiments. Angles between FA's and BP's updates fluctuate around 90 degrees, indicating little to no correspondence. Even for normalized GAIT-prop it can be seen that the updates slightly diverge from BP's in deeper layers, with the average angle at around 10 degrees, but reaching slightly over 20 degrees for the

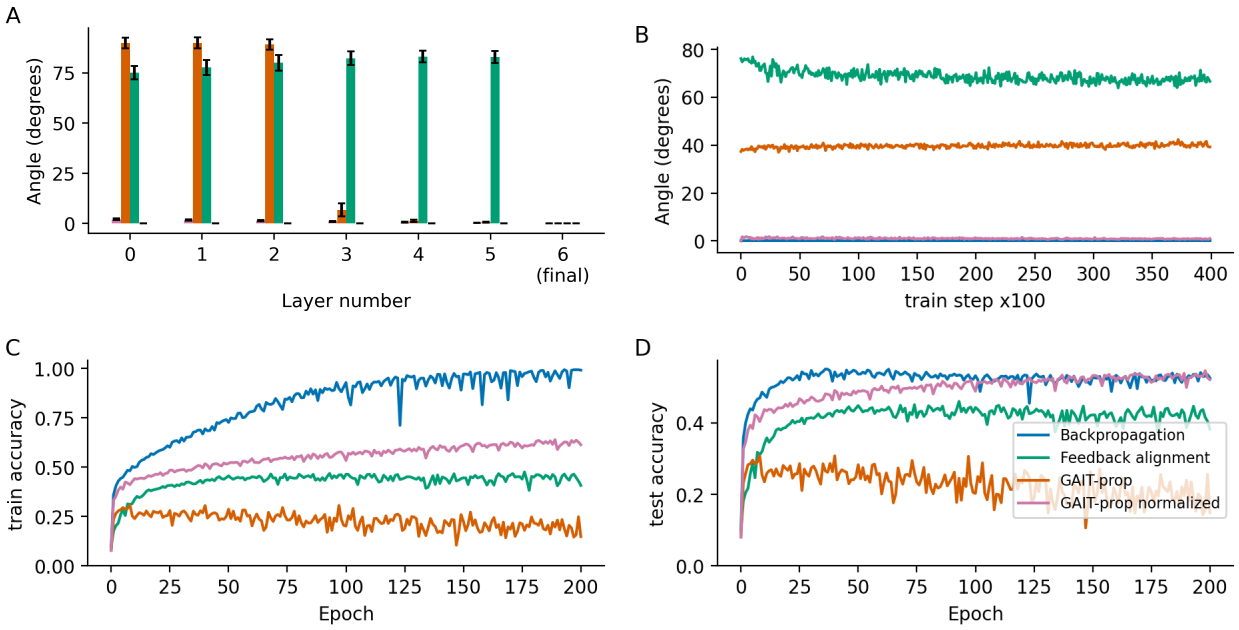

Figure 5: Performance of BP, GAIT-prop, normalized GAIT-prop and FA on CIFAR10 for a fully connected network with 6 hidden layers. For GAIT-prop and FA, correspondence to BP's updates are displayed. A) Angles between GAIT-prop's and FA's weight update vectors and BP's per layer. B) Angles in degrees between GAIT-prop's and FA's updates and BP's weight updates per train step. C) Train accuracy BP, GAIT-prop and FA per epoch. D) Test accuracy for BP, GAIT-prop and FA per epoch.

input layer. We found this is due to the fact that even with the regularizer active, the weight matrices are driven away from perfect orthogonality because of the task-related weight updates being applied. This was confirmed by running the network without performing these weight updates, in which case the angles stay near zero throughout the network. The angles between GAIT-prop's and BP's updates are higher in the ImageNet than in the CIFAR10 experiment. This is likely due to the fact that a slightly lower orthogonality regularizer strength was used in the ImageNet experiment, as setting the regularizer's strength too high slows down convergence on a task that already requires a significant amount of wall time to complete.

It can be seen in Figure 6 that even with the current hyper-parameters, GAIT-prop converges slightly slower than BP, though it eventually reaches parity in test accuracy. For train accuracy, a performance gap remains between GAIT-prop and BP, though the algorithms seem not to have fully converged yet. As in the CIFAR10 experiments, we suspect this is likely the result of more overfitting by BP due to the lack of regularization applied to BP.

4 Discussion

This paper set out to investigate how to scale GAIT-prop to more challenging problems. We found that in principle, GAIT-prop scales to larger and more complex architectures than the relatively shallow fully connected networks used in previous research. This was achieved by tackling its numerical instability issues and adapting the algorithm to CNN architectures. Other biologically inspired learning rules either fail to scale to problems the size of ImageNet (Bartunov et al., 2018), use non-local error information which would not be available at the single synapse level in the brain (Akrout et al., 2019) or require specialized error neurons to be present for every value neuron (Whittington & Bogacz, 2017).

The main limitation of GAIT-prop in terms of its performance is its reliance on invertible network architectures. This makes it incompatible with certain activation functions, as well as non-invertible operations like max-pooling.

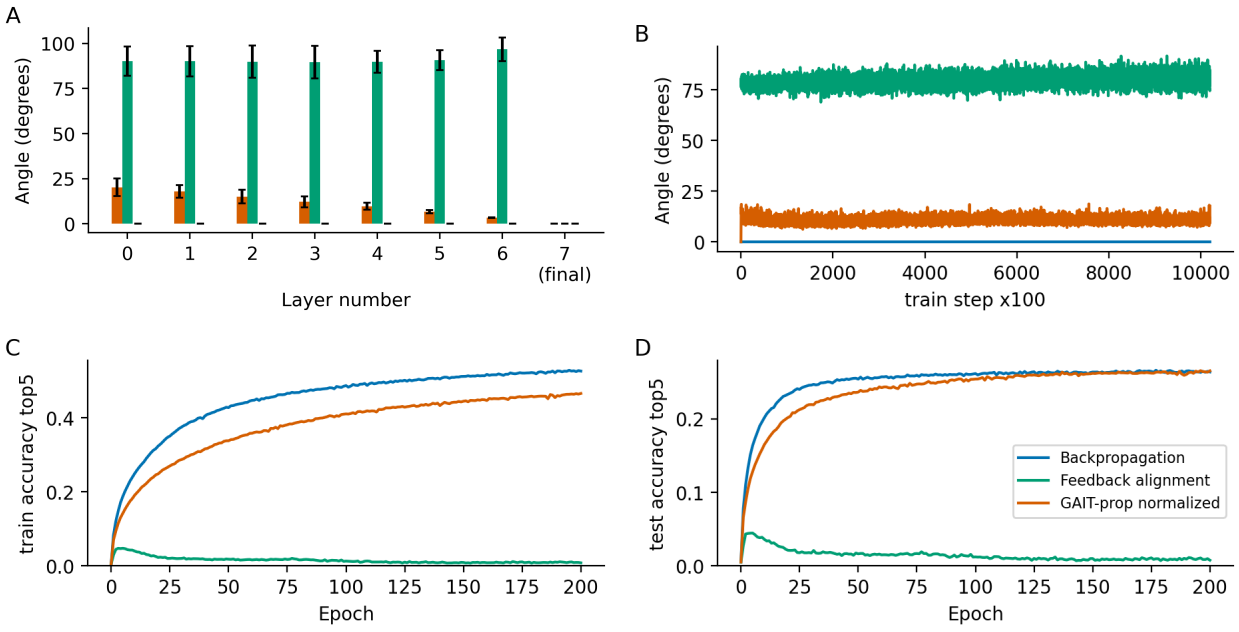

Figure 6: Performance and correspondence with BP for GAIT-prop and FA on ImageNet in a convolutional neural network with 7 convolutional layers. A) Angles in degrees between BP's updates and those by GAIT-prop and FA per layer. B) Angles in degrees between BP's updates and those by GAIT-prop and FA per train step. C) Top 5 train accuracy for BP, GAIT-prop and FA per epoch. D) Top 5 test accuracy for BP, GAIT-prop and FA per epoch.

Additionally, GAIT-prop, in its current implementation, runs significantly slower than BP. This is due to the need to invert weight matrices, include auxiliary units, apply the orthogonality regularization and perform a customized, and thus less optimized, backward pass. This slowdown relative to BP stems from the specifics of modern computer architectures. A biologically inspired algorithm such as GAIT-prop would be expected to run much faster on neuromorphic hardware optimized for such computations. Due to its lack of biological realism, BP likely would not run on such hardware at all, if this hardware is subject to the same physical constraints as the human brain, e.g. the locality of information.

Finally, the orthogonality regularizer appears to slightly slow down convergence, though the magnitude of this effect does not seem prohibitive.

As discussed above, some of the remaining limitations may be remedied in various ways. The goal of the current research, however, is to demonstrate that, in principle, biologically plausible learning can scale to real-world problems. This is demonstrated by GAIT-prop's competitive performance with BP on the complex ImageNet problem, given some constraints on the architecture. We argue that demonstrating the efficacy of any new learning algorithm to large-scale problems is crucial since algorithm that seemingly work well on toy problems have a tendency to break down in more complex settings.

A reader may also ask whether the issues addressed in this work are relevant for biology. For example, the issue overcome by normalization of the target outputs is an issue of precision of floating point numbers. However, we would argue that in biology, any noise floor would also ensure that a sufficiently small target signal would be drowned out amidst network activity (Faisal et al., 2008). One might also ask whether an orthogonality regularizer is biologically plausible. Here, we would argue that lateral inhibition is a well-established phenomenon in neuroscience that could produce a similar decorrelating effect (Békésy, 1967). Therefore we propose that these improvements are also relevant to any biologically motivated analysis of these learning rules.

The presence of (approximately) inverse weights between layers and the need for separate forward

and backward phases in the network seem to be the most prominent remaining departures from biological plausibility, making the elimination of these mechanisms excellent targets for future research.

References

Nasir Ahmad, Marcel A. J. van Gerven, and Luca Ambrogioni. Gait-prop: A biologically plausible learning rule derived from backpropagation of error. *Advances in Neural Information Processing Systems*, 33: 10913–10923, 06 2020.

Mohamed Akrout, Collin Wilson, Peter Humphreys, Timothy Lillicrap, and Douglas B Tweed. Deep learning without weight transport. *Advances in Neural Information Processing Systems*, 32, 04 2019.

Sergey Bartunov, Adam Santoro, Blake Richards, Luke Marris, Geoffrey E Hinton, and Timothy Lillicrap. Assessing the scalability of biologically-motivated deep learning algorithms and architectures. *Advances in Neural Information Processing Systems*, 31, 07 2018.

Y. Bengio. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv*, 1407.7906v3, 07 2014.

Georg Békésy. Mach band type lateral inhibition in different sense organs. *The Journal of general physiology*, 50:519–32, 02 1967.

Francis Crick. The recent excitement about neural networks. *Nature*, 337:129–32, 02 1989.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Aldo Faisal, Luc Selen, and Daniel Wolpert. Noise in the nervous system. *Nature reviews. Neuroscience*, 9: 292–303, 05 2008.

Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11:23–63, 01 1987.

Jordan Guerguiev, Timothy Lillicrap, and Blake Richards. Towards deep learning with segregated dendrites. *eLife*, 6, 12 2017.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton. Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, 25, 01 2012.

Julien Launay, Iacopo Poli, François Boniface, and Florent Krzakala. Direct feedback alignment scales to modern deep learning tasks and architectures. *Advances in Neural Information Processing Systems*, 33: 9346–9360, 06 2020.

Yann LeCun, Y. Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 05 2015.

Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, and Yoshua Bengio. Difference target propagation. *Joint european conference on machine learning and knowledge discovery in databases*, pp. 498–515, 2015.

Timothy Lillicrap, Adam Santoro, Luke Marris, Colin Akerman, and Geoffrey Hinton. Backpropagation and the brain. *Nature Reviews Neuroscience*, 21, 04 2020.

Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10, 2016.

Arild Nø kland. Direct feedback alignment provides learning in deep neural networks. *Advances in Neural Information Processing Systems*, 29, 09 2016.

D. Rumelhart, G. Hinton, and Williams RJ. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, 1, 07 1986.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 09 2014.

Juergen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61, 04 2014.

James Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural Computation*, 29:1–34, 03 2017.