Bio-plausible Credit Assignment in Neural Networks with a General Feedback Alignment Strategy

Daniel Flam-Shepherd
998879423
dflamshe@physics.utoronto.ca

Abstract

The back-propagation algorithm is one of the main tools for credit assignment in neural networks where the loss gradient is computed to back-propagate error from the output layer to the hidden layers. A method called feedback alignment performs almost as well and is more biologically plausible since it avoids using the weights from the forward pass in the the backwards pass by replacing them with random feedback weights. This method can be made even more biologically plausible by decoupling the error signal for each layer, such that the error is not back-propagated through the layers but rather proceeds directly to each hidden layer. This is known as direct feedback alignment. In this work, a general feedback alignment strategy for training neural networks is proposed and experimented with in an supervised and unsupervised setting

1 Introduction and Motivation

Deep neural networks have achieved remarkable success in classification and regression tasks across a variety of disciplines. The back-propagation algorithm [3] is the primary and most successful way to train them. It is unlikely that the back-propagation algorithm is biologically plausible [7]. Bengio et al [4] explain the reasons that the brain could not implement back-propagation are: 1) back-propagation is linear while biological neurons interleave nonlinear and linear operations, 2) the feedback paths must use the same transposed weights, 3) the learning is locked between a forward pass-backwards pass cycle, 4) the brain's feedback paths have to get access the feedforward path gradients, 5) real neurons communicate via spikes, 6) where would the output targets come from?

One proposal that solves 2) is feedback alignment (FA) [1] where the authors show that using fixed random feedback weights in back-propagation can reduce training error to zero and achieve comparable results to back-propagation on test error. Another recently introduced biologically plausible training method is called difference target-propagation [5]. This method handles 1) to 5) by training the network so that each layer reconstructs the layer below, the authors note that when using two linear mappings with a matrix of random feedback weights this method is equivalent to FA. Building on FA, Nokland [2] proposed direct feedback alignment (DFA) where the error proceeds through fixed random feedback connections directly to each hidden layer from the output layer. He demonstrates that comparable accuracy (to back-prop) is achieved with this method. Gilmer et al [6] further show that DFA can be interpreted as a noisy layer-wise training scheme using auxiliary outputs at each layer, in a methodology they call Linear Aligned Feedback Systems (LAFS).

In this work, a general feedback method for updating network weights is proposed, building upon the methods of RFA, DFA and LAFS. The motivation for this general method is to overcome reasons 1)-3). This method is used to create a few different error transportation configurations that are implemented in simple 2 layer networks. These are also used to train simple auto-encoders. Lastly, different feedback connections in DFA and FA are experimented with.

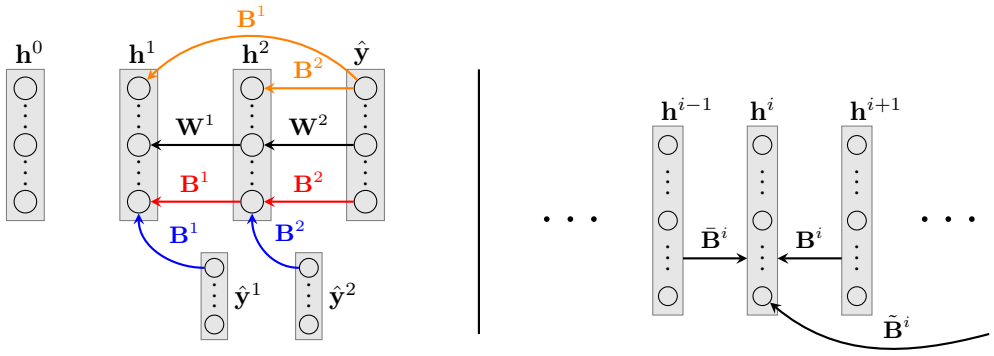

Figure 1: the left figure is the schematic representation of error transportation in various algorithms: back-propagation (BP) in black, random feedback alignment (FA) in red, direct feedback alignment (DFA) in orange and linear aligned feedback systems (LAFS) in blue. The right figure represents the general update rule proposed below for possible ways error can be transported to layer i. Note $\tilde{\mathbf{B}}^i$ can come from the output error or layerwise auxiliary error.

2 Methodology

2.1 A General Feedback Update Rule :

First consider an ℓ layer feed forward network having a output for layer i given by

$$\mathbf{h}^i = f_i(\mathbf{a}^i = \mathbf{W}^i\mathbf{h}^{i-1} + \mathbf{b}^i) \qquad \forall i \in \{1, \ldots, \ell - 1\} \tag{1}$$

where $\mathbf{W}^i, \mathbf{b}^i$ are learned weights and biases. f_i is a nonlinear activation function for all $i \neq \ell$. For the predicted output $\hat{\mathbf{y}} = \mathbf{h}^\ell = f_\ell(\mathbf{a}^\ell) = \text{softmax}(\mathbf{a}^\ell) = p(\mathbf{y}|\mathbf{x})$ where $\mathbf{x} = \mathbf{h}^0$ are the inputs and $\mathbf{y}$ are the targets in the context of classification. Using a cross entropy loss function defined by $L(\hat{\mathbf{y}}, \mathbf{y}) = -\mathbf{y} \cdot \log \hat{\mathbf{y}}$ (for a single training case). The weight update will be (for back-propagation):

$$\delta\mathbf{W}^i = \frac{\partial L}{\partial \mathbf{W}^i} = \frac{\partial L}{\partial \mathbf{a}^i}\frac{\partial \mathbf{a}^i}{\partial \mathbf{W}^i} = \delta\mathbf{a}^i(\mathbf{h}^{i-1})^{\mathrm{T}} \tag{2}$$

where $\delta\mathbf{a}^\ell = \mathbf{e} = \hat{\mathbf{y}} - \mathbf{y}$ and for $i < \ell$: $\delta\mathbf{a}^i = (\mathbf{W}^i)^T\delta\mathbf{a}^{i+1} \odot f_i'(\mathbf{a}^i)$. We can then express the weight update expressions for FA and DFA (visualized in Figure 1):

$$\delta\mathbf{W}^i = [\mathbf{B}^i\delta\mathbf{a}^{i+i} \odot f_i'(\mathbf{a}^i)](\mathbf{h}^{i-1})^{\mathrm{T}}, \quad \delta\mathbf{W}^i = [\mathbf{B}^i\mathbf{e} \odot f_i'(\mathbf{a}^i)](\mathbf{h}^{i-1})^{\mathrm{T}} \tag{3}$$

In this work we consider a general update rule for each layer (visualized in Figure 1) described by

$$\delta\mathbf{W}^i = [\mathcal{F}_i(\mathbf{u}^{i-1}, \mathbf{u}^i, \mathbf{u}^{i+1}) \odot f_i'(\mathbf{a}^i)](\mathbf{h}^{i-1})^{\mathrm{T}} \tag{4}$$

where $\mathbf{u}^{i-1} = \bar{\mathbf{B}}^i\delta\mathbf{a}^{i-1}$ is a term representing the transportation of the error signal from the layer below, $\mathbf{u}^i = \mathbf{B}^i\mathbf{e}$ a direct feedback error term and $\mathbf{u}^{i+1} = \tilde{\mathbf{B}}^i\delta\mathbf{a}^{i+1}$ where the error signal comes from the above layer. The function $\mathcal{F}()$ is some combination of the terms, in the simplest case it averages its arguments. The motivation for this more complicated update rule is for bio-plausible learning: neurons interleave linear and non-linear operations [4], we seek to incorporate that non-linearity more explicitly with this update rule.

2.1.1 Incorporating an Auxiliary Error Signal

For LAFS in the context of DFA we define an auxiliary output at each layer except the output $i \neq \ell$ given by $\hat{\mathbf{y}}^i = f_\ell(\mathbf{a}_x^i)$ where $\mathbf{a}_x^i = (\mathbf{B}^i)^{\mathrm{T}}\mathbf{h}^{i-1}$ is an auxiliary set of logits for all layers. For the last layer $i = \ell$ we use the normal predicted outputs and logits $\hat{\mathbf{y}}^\ell = \mathbf{h}^\ell = f_\ell(\mathbf{a}^\ell)$. The layer-wise loss $L^i(\hat{\mathbf{y}}^i, \hat{\mathbf{y}})$ is optimized to get to get an auxiliary error signal $\delta\mathbf{a}_x^i = \mathbf{e}^i = \hat{\mathbf{y}}^i - \mathbf{y}$ which is used in the update rule for DFA in the place of $\mathbf{e}$ for each layer. This error vector can also be incorporated into the proposed update rule (4) to further make it more bio-plausible as the computation is unlocked from the normally clocked forward pass-backwards pass computation. This is because in LAFS all layers can be updated independently from the above layers (and as long as the forward pass has reached that layer). The update will also be more nonlinear as the feed-backwards paths are further decoupled from the feed-forward paths.

3 Experiments

We investigate the success of the proposed update rule in three experiments with networks consisting of 2 hidden layers, in order to gage if this more bio-plausible update is as successful as DFA and FA. **Experiment 1** concerns testing three error transportation configurations shown in Figure 2. **Experiment 2** incorporates a layer-wise auxiliary error into the configurations of experiment 1 and is shown in Figure 3. Table 1 displays the update rules explicitly for each layer.

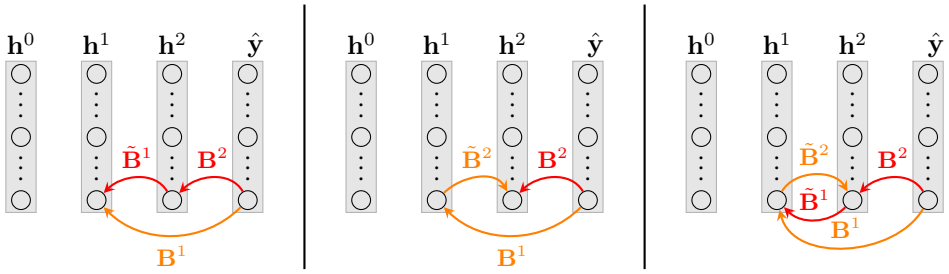

Figure 2: The different error transportation configurations for **Experiment 1**. These are labeled feedback training strategy 1 to 3 from left to right (s1, s2, s3). The orange denotes a DFA type error transportation and the red denotes a FA type of error transportation. If two arrows are the same colour then they are transporting the same error signal.

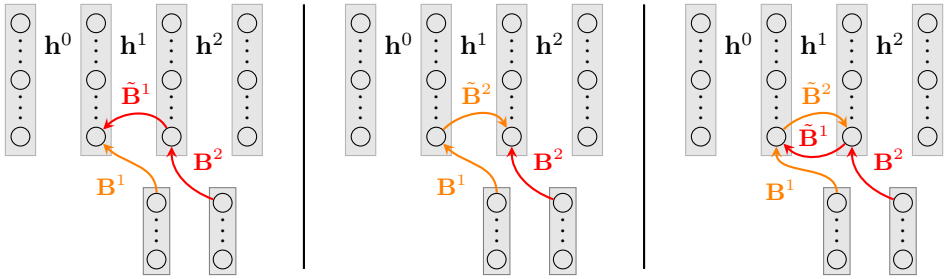

Figure 3: The different error transportation configurations for **Experiment 2**. These are labeled feedback training strategy 1 to 3 from left to right (s1, s2, s3). Similar to Figure 2, the orange denotes a DFA type error transportation and the red denotes a FA type of error transportation. If two arrows are the same colour then they are transporting the same error signal. The error signal for layers 1 and 2 come from the auxiliary outputs discussed in 2.1.1 and are separate from the network.

	strategy ①	strategy ②	strategy ③
$\delta\mathbf{a}^2$	$\mathbf{B}^2\mathbf{e}^2 \odot f_2'(\mathbf{a}^2)$	$\mathcal{F}_2(\mathbf{B}^2\mathbf{e}^2, \tilde{\mathbf{B}}^2\delta\mathbf{a}^1) \odot f_2'(\mathbf{a}^2)$	$\mathcal{F}_2(\mathbf{B}^2\mathbf{e}^2, \tilde{\mathbf{B}}^2\tilde{\mathbf{a}}^1) \odot f_2'(\mathbf{a}^2)$
$\delta\mathbf{a}^1$	$\mathcal{F}_1(\mathbf{B}^1\mathbf{e}^1, \tilde{\mathbf{B}}^1\delta\mathbf{a}^2) \odot f_1'(\mathbf{a}^1)$	$\mathbf{B}^1\mathbf{e}^1 \odot f_1'(\mathbf{a}^1)$	$\mathcal{F}_1(\mathbf{B}^1\mathbf{e}^1, \tilde{\mathbf{B}}^1\delta\tilde{\mathbf{a}}^2) \odot f_1'(\mathbf{a}^1)$

Table 1: $\delta\mathbf{a}^i$ for all strategies. Note for 3 $\delta\tilde{\mathbf{a}}^1 = \mathbf{B}^1\mathbf{e}^1 \odot f_1'(\mathbf{a}^1)$, $\delta\tilde{\mathbf{a}}^2 = \mathbf{B}^2\mathbf{e}^2 \odot f_2'(\mathbf{a}^2)$. The final weight updates are $\delta\mathbf{W}^1 = \delta\mathbf{a}^1(\mathbf{h}^0)^\mathrm{T}$, $\delta\mathbf{W}^1 = \delta\mathbf{a}^2(\mathbf{h}^1)^\mathrm{T}$, $\delta\mathbf{W}^3 = \mathbf{e}(\mathbf{h}^2)^\mathrm{T}$ for all strategies. Note that for experiment 1: $\mathbf{e} = \mathbf{e}^1 = \mathbf{e}^2$ since we are using the output error signal

In **Experiment 3** we use the first set (Figure 2) of training strategies to do unsupervised learning with simple auto-encoders using a reconstruction loss $L(\hat{\mathbf{x}}, \mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2$ such that $\mathbf{e} = (\hat{\mathbf{x}} - \mathbf{x}) \odot (1 - \hat{\mathbf{x}}) \odot \hat{\mathbf{x}}$. Auto-encoders are more bio-plausible given that we do not worry about how supervised learning targets would arise in the brain (reason 6 for why back-propagation is not bio-plausible from [4]). We also want to check the robustness of the training strategy. **Experiment 4** consists of exploring various $\mathbf{B}$ matrices in DFA and FA that might be more bio-plausible given that it is unlikely for there to be fixed uniformly distributed random feedback connections in the brain.

4 Experimental Results

To test the proposed configurations we implement them on the MNIST data set. For the first two experiments, several feedforward networks are trained with 2 hidden layers to keep the computation simple. Networks with 200, 400 and 800 neurons in each layer are considered as well as Tanh, Relu and Sigmoid activations. The random matrices are $\mathbf{B}^i \sim \text{Unif}(-1/2, 1/2)$. The weights are initialized using Xavier initialization and the bias are initialized to be standard normally distributed. Stochastic gradient descent with a learning rate of 0.001 was used throughout. The networks were trained for 100 epochs only, at this point the training error was always less than 0.01%. The functions $\mathcal{F}_i$ were simple element-wise averages ie $\mathcal{F}(\mathbf{x}_i, \mathbf{y}_i) = (\mathbf{x}_i + \mathbf{y}_i)/2$. The results for **Experiment 1 and 2** are summarized in Tables 2 and 3 where ① refers to error transportation configuration 1 (the leftmost in Figure 2 and 3) similar for ② and ③. For the 2x800 Tanh model; the test error curves for training strategies 1-3 (labeled s1, s2, s3) using both error transportation configurations are displayed beside Tables 2 and 3.

4.1 Experiment 1 Results

MODEL	①	②	③
2x 200 Tanh	2.32	2.36	2.42
2x 400 Tanh	2.02	2.07	1.96
2x 800 Tanh	1.96	1.95	2.03
2x 200 Relu	2.55	2.45	2.75
2x 400 Relu	2.15	2.23	2.12
2x 800 Relu	2.03	2.09	2.05
2x 200 Sigm	2.45	2.51	2.63
2x 400 Sigm	2.12	2.18	2.14
2x 800 Sigm	1.99	1.98	1.95

Table 2: Various models and test error percentages for 3 error transportation configurations from Figure 2. Notice the best result (1.95) is close to Nokland's [2].

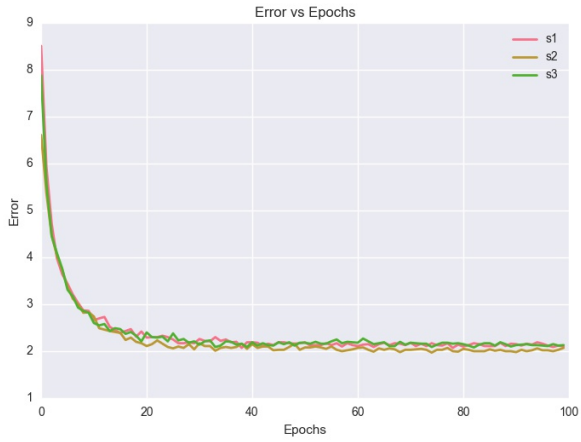

Figure 4: Test error curves for the 2 x 800 tanh model

4.2 Experiment 2 Results

MODEL	①	②	③
2x 200 Tanh	2.42	2.57	2.40
2x 400 Tanh	1.92	2.05	1.99
2x 800 Tanh	1.66	1.86	1.87
2x 200 Relu	2.45	2.51	2.67
2x 400 Relu	2.17	2.27	2.32
2x 800 Relu	1.87	2.02	2.13
2x 200 Sigm	2.43	2.63	2.77
2x 400 Sigm	2.19	2.36	2.41
2x 800 Sigm	1.91	2.04	2.11

Table 3: Various models and test error percentages for 3 error transportation configurations from Figure 2. Notice the best result (1.66) is comparable to Nokland's [2].

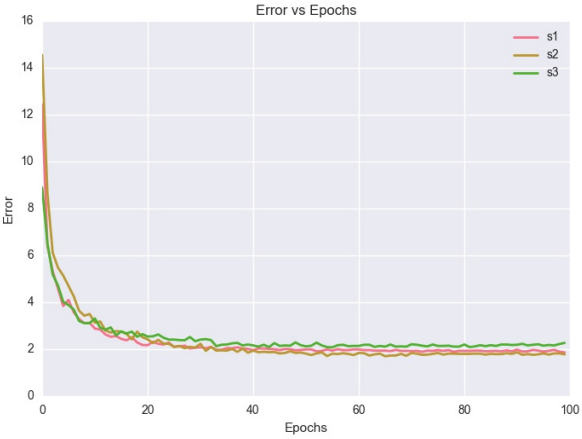

Figure 5: Test error % curves for the 2 x 800 tanh model

4.3 Experiment 3 Results

Using the Error transportation configuration in Figure 2, 2 auto-encoders with architecture 784-128-128-784 and 784-800-800-784 were trained. A reconstruction loss $L(\hat{\mathbf{x}}, \mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||^2$ was used. In Figure 6, the reconstructions of ten randomly selected digits is displayed from the 784-128-128-784 auto-encoder. The first set (top most) is the actual digits followed below by the reconstructions using training strategies : back-propagation, direct feedback alignment, ①, ② , ③ (the three error transportation configurations in Figure 2).

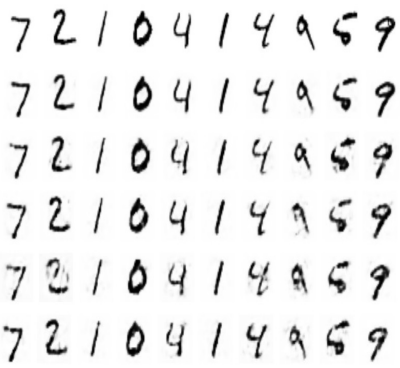

Figure 6: Recreated digits: Notice that DFA and strategies 1-3 are of reasonable and similar quality, but are all worse that back-propagation. Strategy 2 is the worst.

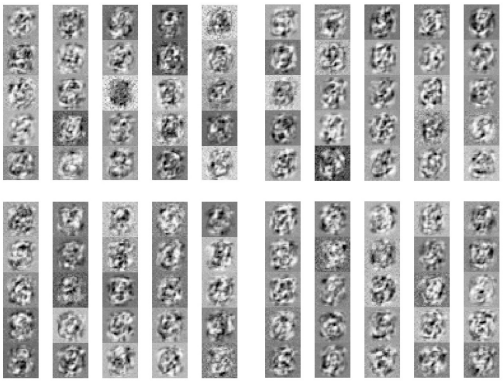

Figure 7: Auto-encoder weight visualization using training method of : DFA (top left), strategy 1 (top right), strategy 3 (bottom left), strategy 4 (bottom right)

4.4 Experiment 4 Results and Details

Various kinds of $\mathbf{B}_i$ matrices were experimented with in FA and DFA, including: $\mathbf{B}_1$ a matrix of ones with noise, $\mathbf{B}_2$ the non-square identity matrix, $\mathbf{B}_3$ the identity with some noise, $\mathbf{B}_4$ a sparse matrix of ones, $\mathbf{B}_5$ a matrix of ones + Unif[-0.01,0.01], $\mathbf{B}_6$ a sparse matrix of ones with noise, $\mathbf{B}_7$ a matrix of noise. The noise is N(0, 0.01) and the sparsity is randomly done using numpy.

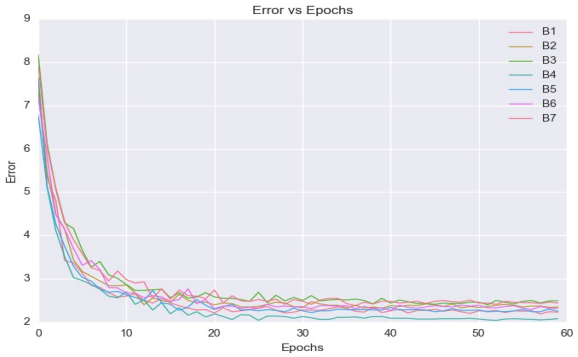

Figure 8: Test error % curves for various $\mathbf{B}_i$, with the 2 x 800 tanh model trained with DFA

	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_3$	$\mathbf{B}_4$	$\mathbf{B}_5$	$\mathbf{B}_6$	$\mathbf{B}_7$
FA	4.64	3.81	5.88	6.33	5.64	3.95	6.22
DFA	2.19	2.32	2.40	2.04	2.22	2.27	2.38

Table 4: Test error percentages for DFA and FA for $\mathbf{B}_i$, with the 2 x 800 tanh model

5 Discussion

The results of experiment 1 and 2 show that the proposed error transportation configurations are almost as good as DFA and FA, experiment 3 shows that the configurations can be used to train simple auto-encoders reasonably well. With more fine-tuning and regularization it may be possible to achieve a test error as low as Nokland does [2].

The last experiment showed that it is possible for learning to occur if the condition $\mathbf{B}^{+}\mathbf{B} = I$ for FA and DFA from [2] is ignored. This means that more biologically plausible feed-back connections could be found. However it is unlikely any of the fixed feedback matrices experimented with are actually bio-plausible. This is an area that can be investigated further, it may be possible to determine what would be bio-plausible feedback matrices in DFA and FA.

The proposed configurations are designed to be more biologically possible since they include linear and nonlinear operations (in transporting the error from the output layer) and they do not use the weights from the forward pass. Also using auxiliary outputs for each layer it is possible for the networks to be trained so that each layer's weights' can be updated independently from the completion of the forward pass and so the training computation doesn't have to be clocked and is, therefore, more bio-plausible.

6 Conclusion

This work builds upon the feedback alignment principle and methods based on it (DFA, LAFS) to form a general weight update formula using error transported linearly and non-linearly from the output error or auxiliary layer outputs (through fixed random feedback connections). This update can be considered more biologically plausible because biological neurons interleave linear and non-linear operations and can be updated instantaneously.

Based on this update six error transportation configurations were proposed; three using an error signal from the network output and the last three using auxiliary outputs. Implementing the configurations on MNIST showed that these training strategies are almost as successful as back-propagation, FA and DFA. It was also demonstrated that the first 3 strategies can be used in unsupervised learning to train simple auto-encoders that reconstruct the input to a neural network.

Many other interesting error transportation configurations can be created from the proposed update rule, but further research should be done to investigate more biologically plausible feedback connections than random uniform ones.

References

[1] Timothy P. Lillicrap, Daniel Cownden, Douglas B. Tweed, and Colin J. Akerman. (2014) Random feedback weights support learning in deep neural networks.*arXiv preprint arXiv:1411.0247, 2014.*

[2] Arild Nkland. Direct feedback alignment provides learning in deep neural networks.*In Advances In Neural Information Processing Systems, pp. 10371045, 2016.*

[3] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. Nature, 323(6088):533536, 1986.

[4] Yoshua Bengio, Dong-Hyun Lee, Jorg Bornschein, Thomas Mesnard, and Zhouhan Lin. Towards biologically plausible deep learning.*arXiv preprint arXiv:1502.04156, 2015*.

[5] Dong-Hyun Lee, Saizheng Zhang, Asja Fischer, Yoshua Bengio (2015) Difference Target Propagation *International Conference on Learning Representations*

[6] Justin Gilmer, Colin Raffel, Samuel S. Schoenholz, Maithra Raghu Jascha Sohl-Dickstein. (2017) Explaining the Learning Dyanmics of Direct Feedbackn Alignment *International Conference on Learning Representations Workshop*

[7] Crick, F. 1989. The recent excitement about neural networks. *Nature 337(6203):129132.*