LRA-E: A stable and robust alternative to Back Propagation

Aditya Sharma
The Pennsylvania State University
State College, USA
aqs6862@psu.edu

Tatheer Zahra
The Pennsylvania State University
State College, USA
tzz5184@psu.edu

ABSTRACT

For years, back propagation has been considered as the backbone of neural networks, presumably derived from the functioning of the brain. Lately, researchers have started exploring alternatives to back propagation, which could not only perform better, but are also cost-effective. Error-driven Local Representation Alignment or LRA-E is one such method. Instead of using a gradient function to calculate error in the output layer and propagate it to hidden layers; LRA-E learns local errors and breaks the global pathways. Additionally, it also eradicates the problems associated with back propagation. In this paper, we first increase the depth of LRA-E to examine the accuracy with high number of layers. Then, we test LRA-E against multiple adversarial attacks (black box and white box) to test the robustness and stability of LRA-E. We also test our results against Direct Feedback Alignment-another alternative to LRA-E. From our experiments, we find that LRA-E outperforms all other methods when it comes to black box attacks, giving us an accuracy of 90% on 25% adversarial samples.

KEYWORDS

back propagation, neural networks, feedback alignment, adversarial sampling, error driven learning, Fast Gradient Signed Method (FGSM)

ACM Reference Format:
Aditya Sharma and Tatheer Zahra. 2022. LRA-E: A stable and robust alternative to Back Propagation. In *Proceedings of* . , 8 pages. https://doi.org/XXXXXX.XXXXXXX

1 INTRODUCTION

In the world of deep learning and neural networks, back propagation has gained immense success. It is usually considered as the only method for training large networks in supervised learning. The alternatives to back propagation are mostly overlooked, which could perform better than back propagation in a lot of scenarios. This creates a blind trust, and it might be causing us several problems, which we are unaware of.

For ages, it was assumed that back propagation was inspired from how our brain functions. However, recently, back propagation has started facing criticism from neurologists across the globe. It can be safely assumed that back propagation doesn't depict how our brain works, and the following reasons justify it [2]:

(1) It restricts you to using symmetric weights only.
(2) Inference and learning are done in different phases.
(3) Back propagation doesn't have localized learning. Instead, it has global learning, where error is only determined at the outer most layer.

In addition to being biologically incorrect, back propagation has some computational issues as well, which are briefly explained below [2]:

(1) There are two passes in every back propagation network: forward pass and backward pass. Forward and backward passes have different set of computations requiring extra work.
(2) Feedback weights are dependent on feed forward weights as they are transpose of feed forward weights along with errors.
(3) Activation and gradient computations are stored separately.

Most of these problems can be be associated with the "global pathways". However, those global pathways are essential for our model to actually learn. Hence, if we break the global pathways, we would need an alternative that is efficient enough to learn representations from the training data.

Due to the addition of these pathways, we also face another issue i.e., large number of calculations in each layer.

Now, when we speak of alternatives, we see one main family of algorithms that learn as efficiently as back propagation. The one family is of Discrepancy Reduction Algorithms, and then, we have Feedback Alignment Algorithms as well. Discrepancy Reduction Family algorithms are heavily based on learning temporal and local representations, which can in return replace back propagation entirely. Two main algorithms that fall in this category are: Error driven Local Representation Alignment and adaptive noise Difference Target Propagation [2].

Our work focuses on error driven Local Representation Alignment or LRA-E. A general approach for the algorithms within this family is as follows:

(1) Finding or understanding latent representations that completely explain our input data. Good latent representations are essential whenever we want our model to learn. Those latent representations could easily be extrapolated to other types of input data later on.
(2) Minimizing the difference between guesses that the system makes and between the actual outputs. If we keep on reducing the differences, we end up with minimum discrepancy in the total system [2].

, ,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

The second type of algorithms consists of Feedback Alignment Algorithms. These algorithms deal with solving the issue of symmetric weights in back propagation. They instead have fixed random weights in feedback pathways. This reduces the dependence on one error metric and the networks figures out how to learn as well. Several learning techniques arose from this one learning technique namely Direct Feedback Alignment, Random Feedback Alignment, and Indirect Feedback Alignment. In this paper, we compared Direct Feedback Alignment with LRA-E and Back Propagation. We also tested it against multiple adversarial attacks [4].

Briefly, this paper compares LRA-E and DFA with Back Propagation on state-of-the-art datasets such as MNIST and F-MNIST. It also goes over testing stability and validity of DFA and LRA-E using black box adversarial sampling. In most cases, LRA-E and DFA outperformed Back Propagation. In the second section, we will go over some relevant background material consisting of alternate back propagation approaches (like LRA-E, DTP, feedback alignment) and then we will briefly cover adversarial attacks. In the third and fourth section, we will go over our methodology consisting of data sets used, different activation functions, regularization, and model details. In results and discussions, we will show our experimental results and discuss possible implications of these results.

2 BACKGROUND

Most famous scientists and researchers have considered Back Propagation to be a flawed approach. Therefore, there has always been a lot of work on finding alternatives to Back Propagation [2]. Unfortunately, a lot of the proposed techniques never gained as much fame as Back Propagation despite being less computationally expensive.

Following are some of the lesser known alternatives to Back Propagation:

2.1 Difference Target Propagation

Difference Target Propagation incorporates auto-encoders at each layer, which learn latent representations between the output which is also called a target and an input. At each layer, the targets are computed and propagated backwards. In this approach, a target value is associated with each feed forward module. This target value then works similar to gradient values and correct the overall loss.

Difference Target Propagation performs comparably to Back Propagation in most cases. However, in case of non-linearities, Difference Target Propagation outperforms Back Propagation for deeper models. Additionally, DPT also addresses the biological issues of Back Propagation. It is more aligned with how our brain works in terms of predictive encoding [11].

2.2 Feedback Alignment

In ordinary Back Propagation, it is the end goal to make forward weights equal to backward weights. This is not ideal in real scenarios. It could easily cause several issues for deeper networks. Feedback Alignment solves this issue by introducing the concept of constant weights.

One of the essential aspects of FA based models is that it ensures that feed back weight doesn't need to be exactly same to feed forward weight. It also emphasis that the network would learn similar to Back Propagation despite differences in feedback and feed forward weights.

FA also uses angles to determine the difference between the updates among hidden layers. Initially, the updates are insufficient. Therefore, the angle is roughly around 90 degrees.As the model starts to learn, the angle starts to decrease as well. The change in angle depicts that feed forward weights have started aligning with the feed back weight-which are kept consistent throughout the training. This approach in a nutshell learns everything in the network and transfers it to the inner neurons. Additionally, Feedback Alignment also ensures non-linearity. Therefore, it was also extended to the Difference Target Propagation too for better results [4].

Feedback Alignment has several further extensions, which are explained below in detail:

2.3 Direct Feedback Alignment

Direct Feedback Alignment or DFA was built upon the concept of Feedback Alignment. After digging through the literature, we realised that DFA was actually proposed by one of the reviewers reviewing a paper related to FA. In addition to consistent feedback weights, DFA also introduces a violation to synaptic asymmetry [4].

Additionally, DFA highlighted an important property of neural networks i.e., learning in a network is mainly due to direct feedback paths between adjacent layers. DFA is a much more novel approach than FA as it also depicts non-linearity in deeper models using experiments and practical applications. However, in case of FA, it was only a proposition and never actually tested. Therefore, DFA has an upper hand over FA.

DFA still doesn't prove that our brain works like this since it is assumed that we would always have direct pathways between all layers. In contrary to this, any approach that has disconnected pathways and still learns efficiently would be much closer to how human brain works [4]. Therefore, despite the fact that DFA is biologically plausible; it is still not an accurate representation of how human brain works.

2.4 Indirect Feedback Alignment

[4] Indirect Feedback Alignment isn't too different from DFA except that instead of direct paths from output layer to all hidden layers, there are indirect paths from output layers to some hidden layers. Other hidden layers are connected using feed forward paths only.

2.5 Error Driven Local Representation Alignment

Error Driven Local Representation Alignment or LRA-E is similar to DTP as it also uses a target variable. In addition to this, it also uses pre-activations and post-activations to determine the difference between predicted values and actual values. LRA-E highly focuses on finding local loses, which are then corrected locally using gradient functions to decrease the overall loss of the system. LRA-E is a practical example of breaking global pathways, that were an issue in Back Propagation. As mentioned before, it also gives importance to finding local representations.

LRA-E is a relatively new method, and hasn't been tested thoroughly. It is also biologically plausible; however, it hasn't been tested on a larger scale or for robustness. Therefore, this paper tests LRA-E against scalability, robustness, and compares with Back Propagation and Direct Feedback Alignment.

3 OTHER TECHNIQUES

3.1 HSIC Bottleneck

HSIC or Hilbert-Schmidt Independence Criterion utilizes the concept of information bottleneck instead of using Back Propagation. It uses a two step approach. First, we approximate information bottleneck and train our network on it. In second step, we find mutual information between hidden layers and labels. After this, we find a substitution for it, which is then maximized. This step is essential as it makes the hidden layers less dependent on the inputs. [12]

3.2 Decoupling Neural Interfaces

Instead of using any kinds of pathways, this method provides a means of communication among Neural Networks.

In this method, all neural network interfaces are decoupled and an approximation for gradients is made, rather than calculating them. This makes the model faster since the computation at each level is independent of the other levels. Additionally, it also amplifies temporal dependencies learnt by Recurrent Neural Networks or RNNs [12].

There are other alternatives to Back Propagation as well. For instance, Online Alternating Minimization with Auxiliary Variables. It is one of the state-of-the-art alternatives [12]. However, it is out of the scope of this paper.

4 PIPELINE FOR ROBUSTNESS

The goal of this paper is to determine if LRA-E and DFA perform comparably to Back Propagation or not. Therefore, we started off by developing a pipeline to see how LRA-E, MLP using Back Propagation, and DFA perform under normal circumstances. Once we had sufficient accuracies (would be discussed in results), we increased the depth of our models and compared the results with Back Propagation. Our results from this were sufficient enough to prove that LRA-E and DFA are at par with Back Propagation.

After this, we wanted to test the robustness of our models and examine if it could easily become a victim to adversarial attacks or not. Before getting into the details of our methodology, let's get some background information on Adversarial Sampling and Attacks[1].

4.1 Adversarial Attacks

With the increase of dependence on Neural Networks and deep algorithms, it is important that we work to make our models secure from any kinds of attacks.

Let's suppose, we use a Deep Learning Framework to detect free runways at airports. Now, the lives of hundreds of people are dependent on that framework. Therefore, it should be immune to attacks from outsiders. In case of Deep Learning frameworks, attacks could be in the form of either manipulating the training dataset or manipulating the testing dataset. It could also mislabel our target labels. In short, deep networks can be susceptible to similar attacks. Hence, our models should implement counter attacks or methods to minimize damage.

Before starting the experiments, our hypothesis was that DFA might outperform LRA-E and Back Propagation in this race. It is due to the fact that we don't have error gradients in DFA. In case of most adversarial attacks, attackers try confusing the model using gradients. Hence, our hypothesis was that Back Propagation would perform the worst in case of Adversarial Sampling.

Adversarial Sampling and attacks are one of the most famous ways of describing attacks on Neural Networks. There are two main categories of attacks in Adversarial Attacks: Black Box attacks and White Box attacks.

4.2 White Box Attacks

As the name suggests, white box attacks depict transparency. In this type of attack, the attacker is usually aware of the details of the model. For example, the attacker would already know the type of the model, the number of hidden layers, the number of neurons, sampling rate, batch size and optimization techniques. As the attacker already has this kind of information, he would generate an attack that is very specific to the model. Therefore, in case of white box attacks, the attacker usually succeeds. [1]

4.3 Black Box Attacks

Black Box attacks are completely opposite to White Box attacks. In this case, the attacker is unaware of everything related to the model. The attacker doesn't know what kind of framework it is, what are the hyper parameters, what are the hidden layers, and what is the expected output. In black box attacks, the attacker might monitor the output or the input for a while, and perform attacks accordingly. It is important to note that adversarial samples learnt from another model using white box attacks can easily be used as black box attacks on your model. Therefore, your model should be immune to those attacks as well. [1]

Following are two types of attacks that manipulate the data:

4.4 Targeted Attacks

In Targeted Attacks, the attacker has control over the output class. Therefore, it manipulates updating gradients in a way that the labels are also manipulated. In case of Targeted Attacks, the attacker usually wants our model to miss-classify to a particular class no matter what the input is.

4.5 Non-targeted Attacks

In case of Non-targeted Attacks, the attacker has no control or authority over the output labels. Therefore, the goal of the attacker is just to make the model miss-classify. This could be done by manipulating the input in a way that it isn't obvious to human beings, but it easily confuses the model. Let's suppose, the input is an image of a cat, but the attacker would manipulate each pixel of the image in a way that it still looks like a cat, but when it passes through the model, it'd be classified an another class.

4.6 Approach for securing models

[9] Even though there are multiple approaches that are used to test deep models or make it stronger. However, following is a generalized way of doing it:

(1) Create a substitute model, which is relatively similar to your actual model. For example, if your model performs convolutions, your substitution model should also have that. This would make the representations in the adversarial samples similar to your model.
(2) Do white box attacks on your substitution model and then create adversarial samples from it.
(3) Use those adversarial samples to perform black box attacks on your actual model and see how it reacts. [9]

Despite being a general approach, it usually gives us a good idea of how robust or weak our model is. You can further change parameters of your model and test it against black box attacks to observe variations too. We followed this technique as well, which will be explained in the next heading.

4.7 Fast Gradient Sign Method

It is one of the most commonly used techniques to construct adversarial samples for images. FGSM is similar to targeted and non-targeted attacks. It tries to maximize the loss by manipulating the gradients. It essentially calculates gradients from a loss function. Once it has the gradients, it builds a new image from it by using the signs from the previous/actual image. However, while building the new image, the pixels are manipulated in a way that the previously calculated loss is maximized [9].

5 METHODOLOGY

5.1 Datasets

We used MNIST dataset with a split of 50000 train images, 10000 validation images, and 10000 test images. These images are numbers 0-9, so the data has total 10 labels.

Another data set we used is Fashion-MNIST. This dataset is similar to MNIST in terms of number of images, number of labels and train-validation-test split. The main difference is that this data set consists of clothes instead of numbers. Using two different data sets helped us analyze the performance of LRA-E in a more general sense.

5.2 Adversarial Attacks

For creating adversarial sampling, we used a substitution model with FGSM to create a neural network. This model was used to generate adversarial samples in the test data set of MNIST and F-MNIST. While FGSM is a white box attack, this method of generating adversarial samples on a substitute model and then using it to test a separate model becomes a black box attack. To analyze the effect of these adversarial samples on the accuracy of back propagation, LRA-E, and DFA, we performed attacks with various concentration of adversarial samples. For creating these different concentration samples, the batch size of test data set is varied and only one batch is used to create adversarial samples. Different concentrations considered - 5 percent, 10 percent, 15 percent, 20 percent, 25 percent, 50 percent, and for some cases even 100 percent.

5.3 Activation functions

To study the effect of various activation functions on the robustness of model, LRA-E models with different activation functions were tested. Following activation functions were considered - tanh, sine, ReLU, leaky ReLU, squash, and sigmoid. The squash function and leaky ReLU were showing erratic behavior were the accuracy was decresing over time for squash and loss went to nan for leaky ReLU. Due to this, we finally worked with only the remaining activation functions.

5.4 Regularization

To create a regularized model in the hopes of having more robustness, we injected adversarial sampling in the validation data set. The procedure to create these adversarial samples are similar to the one used for test data set, that is, using FGSM with a substitute model. The validation data set contained various concentration of adversarial samples ranging from 5 percent to 20 percent. Different concentration of adversarial samples were injected in the test data set to measure the performance of this regularized model.

5.5 Hyper-parameter Tuning

LRA-E was tested with different values of beta and gamma to fine tune the model. Using this model, activation functions were evaluated and the most optimum model was used in the regularization step mentioned above.

6 EXPERIMENTAL SETUP

6.1 Model Details

We worked with a 4 layer Multi-layer perceptron model. For LRA-E and back propagation, the two hidden layers had 256 nodes in them. The input layer has 784 nodes, so the input is a flattened (28×28 pixel image is converted to 784) before feeding it to the model. The output has 10 nodes, one for each label. Training batch size is 50. The back propagation model uses gradient descent optimizer with learning rate of 0.001. For both LRA-E and back propagation models, each model was trained for 20 epochs. LRA-E model with beta value 0.1 and gamma value 1 was used as it had the highest accuracy during hyper parameter testing.

For DFA, we also had a four layer model, where two layers were hidden. The input for DFA was also flattened similar to LRA-E and Back Propagation to keep the results in sync. The learning rate was 0.001 and batch size was 200. DFA doesn't have gradient calculations. Therefore, we didn't calculate losses similar to Back Propagation.

6.2 Noise-based Attack

First attempt to test stability and robustness of these learning approaches used noise based models. Basically adding a little noise in each image of test sample. Normal distribution with varying standard deviation was used to add noise. The general observation was that too little a noise leads to a some decrease in accuracy (from 96 percent to 91 percent). However, this random noise is not very effective in fooling the model. If we increase the noise, the input gets distorted and the accuracy of the model drops drastically. As the input becomes distorted, this method is not very effective

to check robustness of the model. There are more standard noise based adversarial sampling models that are worth testing.

6.3 Substitution Model Details

The aim is to test the stability and robustness of alternative learning approaches like LRA-E and DFA. To test the stability and robustness, we used noise based method and black box adversarial sampling.

6.4 Black Box attack

This attack consisted of using a substitute model - Multi-layer perceptron with 4 layers. All parameters of the substitute model were kept similar to the actual model. Details of these parameters are as follows - input layer size is 784, hidden layers have 256 nodes, testing batch size of 128, epochs 5. The FGSM model was trained using this substitute model. Once trained, this model was used to attack LRA-E, back propagation, and DFA models. Torchattacks library was used for these attacks. FGSM with epsilon 0.3 was used.

7 RESULTS

This is Table 1 showing results for LRA-E with tanh activation function on MNIST data set. The sampling percent represents adversarial sampling percent in test data set. The train and validation accuracy is 98.56 % and 96.11 %˙

Table 1: LRA-E result (tanh)

Sampling Percent	Test Accuracy
0	96.1
5	94.87
10	93.8
15	92.67
20	91.64
25	90.55
50	85.08
100	76.95

Table 2 showcases results for back propagation with tanh activation function on MNIST data set. The train and validation accuracy is 92.69 % and 93.2 %

Table 2: Back Propagation result (tanh)

Sampling Percent	Test Accuracy
0	92.91
5	90.75
10	88.84
15	86.79
20	84.59
25	82.61
50	72.44
100	54.18

Table 3 represents LRA-E for F-MNIST data set. The train and validation accuracy are 88.81 % and 85.91 % .

Table 3: LRA-E result for F-MNIST

Sampling Percent	Test Accuracy
0	85.39
5	82.81
10	80.3
15	77.7
20	75.06
25	72.66
50	60.12
100	35.58

Table 4 represents results for back propagation for F-MNIST data set. The train and validation accuracy are 86.21 % and 84.66 % .

Table 4: Back Propagation result for F-MNIST

Sampling Percent	Test Accuracy
0	83.87
5	81.55
10	79.5
15	77.4
20	75.28
25	73.22
50	62.32
100	41.68

Table 5 represents results for DFA on F-MNIST data set. The train and validation accuracy are 86.21 % and 84.66 % .

Table 5: Direct Feedback Alignment result

Sampling Percent	Test Accuracy
0	89.01
5	89.97
10	88.77
15	88.52
20	88.49
25	88.22
100	88.01

Table 5 represents results for direct feedback alignment for MNIST data set.

, ,

Table 6: LRA-E with ReLU result

Sampling Percent	Test Accuracy
0	94.27
5	93.4
10	92.69
15	91.9
20	90.9
25	89.99
100	77.94

Table 6 represents results for LRA-E with ReLU activation function for MNIST data set. The training accuracy is 94.66 percent and validation accuracy is 94.37.

Table 7: LRA-E with Sigmoid result

Sampling Percent	Test Accuracy
0	90.06
5	89.51
10	89.02
15	88.62
20	88.23
25	87.7
100	80.24

Table 7 represents results for LRA-E with Sigmoid activation function for MNIST data set. The training accuracy is 89.5 percent and validation accuracy is 90.58.

Table 8: LRA-E with Sin result

Sampling Percent	Test Accuracy
0	96.29
5	95.3
10	94.48
15	93.62
20	92.59
25	91.72
100	78.23

Table 8 represents results for LRA-E with sine activation function for MNIST data set. The training accuracy is 99.36 percent and validation accuracy is 96.4 percent.

Table 9: LRA-E Hyper Parameter Tuning for Beta

Beta	Train Accuracy	Validation Accuracy	Test Accuracy
0.001	90.56	90.55	90.4
0.01	94.06	93.29	92.72
0.05	98.12	96	95.47
0.1	99.3	96.5	96.03
0.2	92.5	92.46	92.14
0.5	12.2	11.66	12.3

Table 10: LRA-E Hyper Parameter Tuning for Gamma

Gamma	Train Accuracy	Validation Accuracy	Test Accuracy
0.1	99.4	96.3	96.2
0.5	99.4	96.4	96.4
1	99.3	96.5	96.5
1.5	99.6	96.45	96.5
2.5	99.23	96.06	95.8
5	99.07	96.02	95.87

Table 9 and 10 represent hyper parameter tuning for LRA-E with sine activation function. The parameters considered are beta and gamma. The model with beta value 0.1 and gamma value 1.0 gives us the best results.

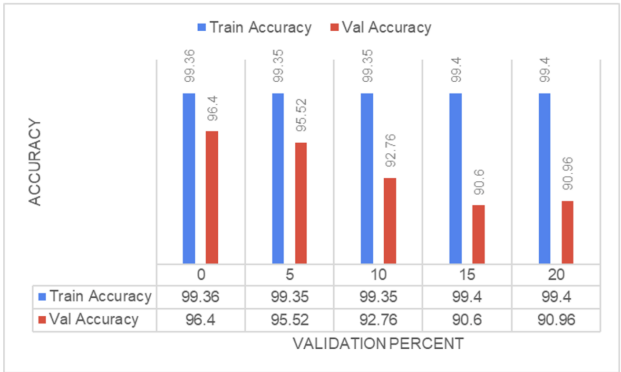

Figure 1: Regularized LRA-E Model

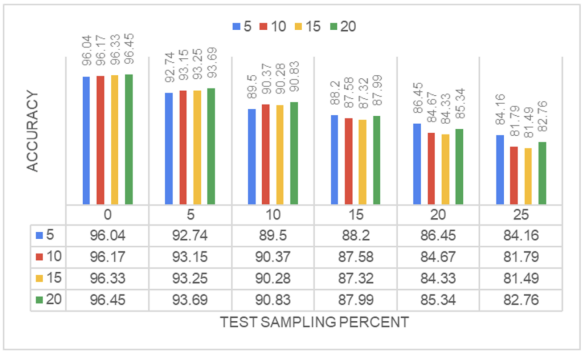

Figure 2: Performance of Regularized LRA-E Model

Figure 1 represent the performance of regularized LRA-E model on train and validation data set of MNIST. The validation percent represents increment in adversarial samples in validation set. The figure 2 represents the performance of these models on the test data set.

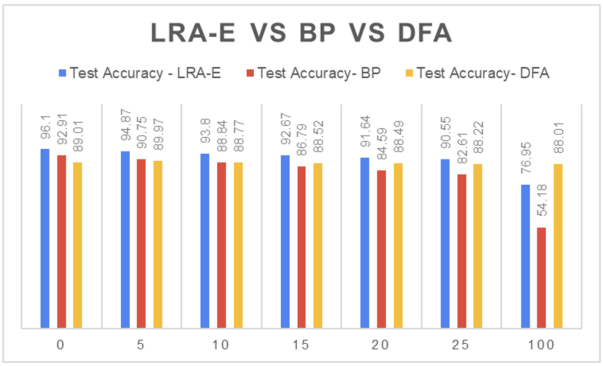

Figure 3: Comparison of BP, LRA-E, and DFA Model

8 DISCUSSION

The results from alternate learning approaches like LRA-E and DFA are quite promising. LRA-E out performed back propagation on MNIST as well as MNIST data set. Each result represents average value across 5 trails, so these models are stable as well.

Back propagation and LRA-E have some similarities in the sense that both have a downward trend with increase in concentration of adversarial attack. DFA performed quite the opposite with showing increase in accuracy with increase in adversarial sample concentration in some cases, while providing constant accuracy in other cases. This might be due to use of a set matrix for feedback instead of the transpose of forward weights. This might mean that DFA is resistant to adversarial attacks. Although more testing needs to be done. It should also be noted that while the performance of DFA did not deteriorate with increase in adversarial samples, the performance is lagging behind BP and LRA-E. Perhaps, the model needs to be fine tuned.

The behavior of LRA-E was analyzed in great depth. Various activation functions were tested against adversarial attacks. Sine activation function performed the best. The LRA-E model was also fine tuned for parameters beta and gamma.

Regularization of LRA-E models by injecting adversarial sampling in validation data set did not produce strong results other than the slight increase when going from 15 percent to 20 percent.

9 CONCLUSION

Back propagation being the back bone of deep learning networks has been widely used. However, there is a strong criticism regarding back propagation not being biologically plausible [2]. The main concern is the global feedback pathway required for learning. This lead researchers to work on many biologically motivated learning approaches. This paper focused on two of these, namely, LRA-E and DFA. These algorithms have not been tested for stability and robustness. Thus, by testing them under various adversarial attacks, we try to showcase the capabilities of these approaches.

These approaches are tested on a 4 layered MLP and the results are quite promising. With DFA showing some resistance to adversarial sampling and LRA-E outperforming back propagation on various adversarial attacks generated using FGSM on a substitute model. LRA-E was tested on various parameters like activation function, beta, gamma, and regularized by injecting adversarial samples in validation data set. We can safely conclude that both of these approaches have robustness as good as back propagation.

While these results are good, further testing of these approaches on various data sets and architectures like CNN, RNN are required. However, for the scope of this project, our results are sufficient. We made a hypothesis earlier that Back Propagation might perform the worst in case of adversarial samples. Through our experiments, it was consistently proved that Back Propagation performed the worst. Hence, our hypothesis was correct.

To conclude, DFA and LRA-E are better than Back Propagation. However, more experimentation needs to be done before applying it widely.

10 REFERENCES

(1) Qinglong Wang, Wenbo Guo, Kaixuan Zhang, Alexander G. Ororbia II, Xinyu Xing, Xue Liu, and C. Lee Giles. 2017. "Adversary resistant deep neural networks with an application to malware detection.

(2) Alexander G. Ororbia, Ankur Mali AAAI-19. Biologically Motivated Algorithms for Propagating Local Target Representations

(3) Lillicrap, T. P.; Cownden, D.; Tweed, D. B.; and Akerman, C. J. 2014. Random feedback weights support learning in deep neural networks

(4) Nøkland, A. 2016. Direct feedback alignment provides learning in deep neural networks. In Advances in Neural Information Processing Systems, 1037–1045

(5) Lee, D.-H.; Zhang, S.; Fischer, A.; and Bengio, Y. 2015a. Difference target propagation. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 498–515. Springer.

(6) Machado, G. R., Silva, E., Goldschmidt, R. R. (2021). Adversarial Machine Learning in Image Classification: A Survey Toward the Defender's Perspective. ACM Computing Surveys (CSUR), 55(1), 1-38
(7) Lee, Dong-Hyun, et al. "Difference target propagation." Joint european conference on machine learning and knowledge discovery in databases. Springer, Cham, 2015.
(8) Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
(9) Kui Ren, Tianhang Zheng, Zhan Qin, Xue Liu, Adversarial Attacks and Defenses in Deep, Engineering, Volume 6, Issue 3, 2020, Pages 346-360, ISSN 2095-8099, https://doi.org/10.1016/j.eng.2019.12.012.
(10) Nøkland, Arild. Direct Feedback Alignment Provides Learning in Deep Neural Networks, 2016, https://arxiv.org/abs/1609.01596v5
(11) Lee, Dong-Hyun and Zhang, Saizheng and Fischer, Asja and Bengio, Yoshua. Difference Target Propagation. 2014. https://arxiv.org/abs/1412.7525
(12) Greenfeld, Daniel and Shalit, Uri. Robust Learning with the Hilbert-Schmidt Independence Criterion. 2019.