Prediction Accuracy and Adversarial Robustness of Error-Based Input Perturbation Learning

Soha Lee
School of Computer Science and Engineering
Kyungpook National University
Daegu, KOREA
shlee1012@knu.ac.kr

Heesung Yang
School of Computer Science and Engineering
Kyungpook National University
Daegu, KOREA
hs.yang@knu.ac.kr

Hyeyoung Park
(Corresponding Author)
School of Computer Science and Engineering
Kyungpook National University
Daegu, KOREA
hypark@knu.ac.kr

Abstract— **Error backpropagation algorithms are essential for training deep neural networks, but they have several problems due to sequential feedback calculation to propagate error signals. Recently, a method using only two consecutive forward calculation with input perturbation has been proposed as an alternative, which is called PEPITA. Although PEPITA has shown the possibility of successful learning without backward computation, it is still in its early stages and needs further investigation on its properties. In this study, we analyze the characteristics of PEPITA and propose a new method for generating modulated input, specifically for the second forward computation. In particular, we show that the adversarial perturbation used to generate attack samples is closely related to the input perturbation process of PEPITA, and propose to use the adversarial perturbation in combination with PEPITA learning. The potential of the existing PEPITA and the proposed modification is analyzed through experiments using different activation functions under various attack conditions. From the experiments, we confirm that a proper combination of input modulation and activation function can improve the prediction accuracy and adversarial robustness. This work extends the applicability of PEPITA and lays the foundation for the analysis of alternative learning algorithms.**

Keywords—Error backpropagation, Biological plausibility, Feedback alignment, Weight transport problem, Adversarial Attack, Two forward Learning

I. INTRODUCTION

Artificial neural networks are computer-based technologies that mimic the way the human brain works. Through the interaction of multiple neurons, the brain is able to perform sophisticated signal processing and complex tasks. The artificial neural networks can learn complex patterns through multiple layers of neurons to perform difficult tasks. During the learning process of the neural networks, it is necessary to assign the responsibility for each neuron's weight to the current network outputs, which is known as the credit assignment (CA) problem. To solve this problem, the error backpropagation algorithm (BP) [1] has been proposed, which starts with the error in the output layer and propagates it sequentially backwards to adjust the weights of the neurons in each layer.

Error backpropagation algorithms are a popular method for training deep network models, but they have several limitations. One of these limitations is the weight transport problem [2, 3], which requires the exact values of forward weights for the backward process of updating weights. This is not biologically plausible because, in a real neuron, backward transmission of information along the axon is not possible. Furthermore, the error backpropagation method requires sequential computation of the error gradient when updating weights in a layer-by-layer manner, which is commonly called the backward locking problem [4, 5]. The sequential feedback of output error differs from the local learning mechanism of the human brain and makes it difficult to parallelize learning [6, 7].

Several alternative learning algorithms have been proposed to overcome the limitations of error backpropagation learning [8-18]. Recently, new learning methods [16-18] have been proposed that use two forward calculations instead of forward-backward iteration. These methods have been experimentally shown to enable learning without the backward path of error propagation, which is the main cause of many limitations of BP. In particular, the method of presenting the error to perturb the input to modulate activity (PEPITA) [17, 18] utilizes input with additional perturbation for the second forward calculation, which is defined as a random transition to the output error for the first input. The differences in the activation of each neuron for the two inputs are then used for determining the update term. The input modulation used for the second forward calculation in PEPITA can be considered as generating adversarial samples in deep learning. In PEPITA, network error is used to generate perturbed input for the second forward computation. Similarly, in deep learning, adversarial samples are generated by using network loss to generate such samples.

This paper investigates the adversarial robustness of the PEPITA learning method, and provides a modified version of PEPITA for improving its performance. First of all, we investigate the effect of activation function on the performance of PEPITA learning. Additionally, we propose a modified perturbation method using the gradient of the loss function and demonstrate its effects on the learning accuracy and adversarial robustness.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No.NRF-2020R1A2C1010020).

This work was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning(KETEP) granted financial resource from the Ministry of Trade, Industry & Energy, Republic of Korea (No. 20204010600060)

This paper is organized as follows. Section 2 introduces related work, and Section 3 describes the PEPITA method, which is the basis of this study and presents a modified version of the PEPITA learning procedure where the activation function is transformed and perturbed by a loss gradient. Section 4 presents the experimental results, and Section 5 concludes with a summary and outlines future work.

II. Related Works

Several alternative learning methods [8-18] have been proposed to address the weight transport and backward locking problems of the error backpropagation algorithms. The first approach [8-13] solves the weight transport problem by establishing separate path for error feedback and random fixed backward weight [8-11] and their update rules [12, 13]. In addition, some of them uses direct feedback path from the network output to each hidden layer, which can also solve the backward locking problem as well. The second approaches [14-15], avoid the weight transport problem by propagating the target value instead of computing the error gradient.

Unlike the previous works, more innovative approaches [16-18] have been proposed to eliminate the backward path itself and learn through two consecutive forward calculations. These methods update the network weights by using the discrepancies obtained from two sequential forward computations. They do not use backpropagation, which solves the weight transport problem and simplifies the computation of error gradients.

The forward-forward (FF) method [16] uses two forward passes to train instead of the traditional backward pass. This method involves performing the first forward computation using clean data and the second forward computation using distorted data as input. The goal of this process is to maximize the difference in neural activity values between clean and distorted data and use it for training. Although the FF method shows the possibility of learning without backward path, it requires an elaborated generation of distorted data. PEPITA [17, 18] is another method that performs two forward computations. This method perturbs the input used in the first forward computation when generating the input for the second forward computation. In contrast to FF, in the second forward computation, the error from the first computation is added to the existing input to create a perturbed data, which is then used to modulate the activation values of the layer. Finally, the weights are updated by using the modulated activation values computed during the second forward pass and the activation values computed during the first forward pass.

Although PEPITA is still in its early stages and its performance is not optimal, it is noteworthy for achieving acceptable results without requiring backward computation. Additionally, the method is significant because it mimics neuromodulators in the brain, implementing a top-down learning mechanism to some extent. In addition, methods that perform two forward computations naturally eliminate backward sequential computations, thus eliminating the need for forward weights and solving the weight transport problem. These methods use local learning based on Hebbian learning [22] principles to solve non-local problems, making learning more biologically plausible. Also, the update lock problem can be partially solved by allowing the layer that has completed the second forward pass to initiate a new first forward pass.

III. Perturbated Error Learnig without backward

3.1. PEPITA Learning

First, we introduce the original PEPITA learning method, which removes the backpropagation path and uses two forward paths, and propose its modification. The discussion focuses on the multilayer perceptron model, but the same approach can be applied to a variety of network models, including convolutional neural networks. When an input $\boldsymbol{x}$ is given to a network with L layers, the first forward calculation is performed sequentially in layer-by-layer manner. The output vector $\boldsymbol{h}_l$ of the l-th layer is computed by using the output of the l-1th layer, the weight matrix $\boldsymbol{W}_{l-1}$ connecting them, and the activation function $f(.)$, such as

$$\boldsymbol{a}_l = \boldsymbol{W}_l \boldsymbol{h}_{l-1}, \tag{1}$$

$$\boldsymbol{h}_l = f(\boldsymbol{a}_l), \tag{2}$$

where $l = 1, \dots, L$. Through the sequential forward computation from the input layer to the output layer, we obtain the network output $\boldsymbol{h}_L$ and the error vector $\boldsymbol{e} = \boldsymbol{h}_L - \boldsymbol{y}$, which is the difference from the target value $\boldsymbol{y}$.

In the second forward calculation, the error vector $\boldsymbol{e}$ from the first calculation is used to make the modulated input for the second calculation. The error vector is linearly transformed by a random fixed matrix ($\boldsymbol{F}$) to make an error-based perturbation $\boldsymbol{Fe}$, and the perturbation is added to the original input $\boldsymbol{x}$. The perturbated input is then used to perform the second forward computation to obtain the modulated outputs for each layer, such as

$$\boldsymbol{h}_1^{err} = f(\boldsymbol{W}_1(\boldsymbol{x} + \boldsymbol{Fe})), \tag{3}$$

$$\boldsymbol{h}_l^{err} = f(\boldsymbol{W}_l \boldsymbol{h}_{l-1}^{err}), \tag{4}$$

$$\boldsymbol{e}^* = \boldsymbol{h}_L^{err} - \boldsymbol{y}, \tag{5}$$

where $l = 2, \dots, L$.

Using the activation values from the two forward calculations, the weights are updated through local learning, which is defined as

$$\Delta \boldsymbol{W}_1 = (\boldsymbol{h}_1 - \boldsymbol{h}_1^{err})(\boldsymbol{x} + \boldsymbol{Fe})^{\mathrm{T}}, \tag{6}$$

$$\Delta \boldsymbol{W}_l = (\boldsymbol{h}_l - \boldsymbol{h}_l^{err})(\boldsymbol{h}_{l-1}^{err})^{\mathrm{T}}, \tag{7}$$

$$\Delta \boldsymbol{W}_L = (\boldsymbol{e}^*)(\boldsymbol{h}_{l-1}^{err})^{\mathrm{T}}. \tag{8}$$

In previous studies [17, 18], PEPITA demonstrated the possibility of learning without backward propagation. However, this approach is still in its early stages and requires further investigation and modification. As a first step toward more understanding on the PEPITA learning method, this paper investigates the effect of two components: the activation function $f(.)$ and the perturbation term $\boldsymbol{Fe}$. While the original PEPITA uses the rectified linear unit (ReLU) for activation function, we try to apply sigmoidal unit, which has biologically more plausible shapes. In addition, we try to find more strategic generation of perturbed signal instead of just random projection of error vector, which will be described in the next subsection.

3.2. Learning with Adversarial Perturbation

As described in Section 3.1, PEPITA generates a perturbed input by applying a random matrix to the errors obtained from the first forward computation and adding them to the input. This perturbed input is then used in the second forward computation, and the differences in the activities of the hidden neurons are used to update the weights. In this study, we investigate how the generalization performance and adversarial robustness of the model vary depending on the activation functions and input perturbations.

The technique of adversarial attack is used to create input perturbations. Adversarial attack is a deep learning technique that generates manipulated input data, called adversarial samples, with the intention of deceiving a model. Adversarial samples were originally designed to cause models to make incorrect predictions. However, they can also be used to improve the adversarial robustness of the model by serving as augmented training samples. Noting that PEPITA uses similar input perturbation for training networks, we propose using adversarial samples to define the perturbed input for the second forward calculation in PEPITA learning.

Instead of the random projection of the error vector, we use the gradient of the loss function over the input vector, as shown in Figure 1, to generate adversarial samples as input for the second forward calculation. The technique of using the loss gradient to create adversarial samples is inspired by the fast gradient sign method (FGSM) [20], which is the most well-known attack method. While the conventional FGSM uses the sign value of the loss gradient, we take the gradient itself with small scaling coefficient ϵ, and the perturbed input is defined as $\boldsymbol{x} + \epsilon \nabla_x E(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{\theta}$ is the network parameter.

The resulting adversarial sample is then used for the input of the second forward calculation to obtain the output of first hidden layer such as

$$\boldsymbol{h}_1^{err} = f\left(\boldsymbol{W}_1\left(\boldsymbol{x} + \epsilon \nabla_x E(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y})\right)\right). \tag{9}$$

When the loss function E is the squared error, which has the form,

$$E(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \|\boldsymbol{h}_L - \boldsymbol{y}\|^2, \tag{10}$$

the gradient of loss with respect to the input is obtained as

$$\nabla_x E(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \frac{\partial \boldsymbol{h}_L}{\partial \boldsymbol{x}} \boldsymbol{e}. \tag{11}$$

Here, we can see that the error vector used in the conventional PEPITA is also used in the proposed adversarial perturbation. However, the proposed method uses the Jacobian matrix instead of a random fixed matrix, which may provide a more meaningful direction. This process has the potential to enhance the network's ability to respond to adversarial environments, thereby improving the model's overall robustness and performance. In addition, selecting an appropriate activation function can significantly impact the model's learning and generalization performance. The PEPITA learning method can further improve the robustness and accuracy of models by integrating these two approaches.

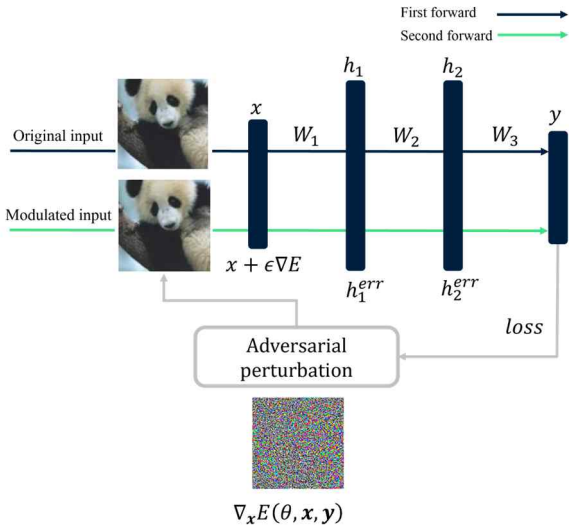

Fig 1. PEPITA using adversarial perturbation

IV. Experiments on Benchmark Data Sets

4.1. Experimental environment and data sets

To investigate the effect of the modified PEPITA learning method, computational experiments are performed on two benchmark data sets. First, the MNIST data set [24] is composed of handwritten digit images that can be classified into 10 categories from 0 to 9. It consists of 70,000 samples of data, of which 60,000 are used for training and 10,000 for the test. Each data sample consists of 28x28 gray image and a corresponding class label. The second benchmark data, Fashion MNIST dataset [25] consists of 70,000 grayscale images categorized into 10 classes, including T-shirts, dresses, and shoes. Each sample is a 28x28 pixel grayscale image, and the whole set is divided into a training set with 60,000 images and a test set with 10,000 images. The dataset presents a more complex classification challenge than MNIST.

For training MNIST dataset, 100 training epochs were performed, and the weight parameters giving the minimal test error was finally selected. The same strategy was applied for training Fashion MNIST dataset. The network structure for the experiment was determined according to the previous studies [17, 18, 22]. For both datasets, we used multilayer perceptron model with one hidden layer and 1024 hidden neurons. The learning rate and epsilon value were manually optimized for each model through the experiment to obtain minimum test error performance and fast convergence. The cross-entropy function was used as the loss function of all experiments.

To investigate how performance changes under different conditions, we compared three learning methods: the original BP, the original PEPITA with input perturbation by random projection of error vector, and the proposed modification of PEPITA with adversarial perturbation of input. Additionally, we attempted to merge the original PEPITA with the modified version. We trained using the original PEPITA for the first half of the learning epochs and then switched to the modified PEPITA for the remaining half. We also examined the impact of nonlinearity types (activation functions) on performance, specifically the logistic sigmoid and ReLU function.

TABLE I. CLASSIFICATION ACCURACIES ON MNIST AND FASHION MNIST DATA DEPENDING ON LEARNING METHODS AND ACTIVATION FUNCTIONS

	Classification Accuracy for Test Set (%)		
Activation	***Method***	***MNIST***	***Fashion MNIST***
ReLU	BP	**98.39**	**89.14**
	Original PEPITA	98.04	86.38
	Modified PEPITA	95.91	85.69
	Combined PEPITA	98.19	87.65
Sigmoid	BP	97.7	79.3
	Original PEPITA	98.04	86.74
	Modified PEPITA	91.39	83.17
	Combined PEPITA	**98.36**	**86.75**

TABLE II. ADVERSARIAL ROBUSTNESS ON MNIST DATA DEPENDING ON LEARNING METHODS AND ACTIVATION FUNCTIONS

		Robustness to Adversarial Attack (%)					
Activation	***Method***	***FGSM Attack***			***PGD Attack***		
		ε=0.1	***ε=0.2***	***ε=0.3***	***ε=0.1***	***ε=0.2***	***ε=0.3***
ReLU	BP	**54.16**	**12.10**	**1.93**	**21.30**	**0.56**	**0.04**
	PEPITA	7.05	0.08	0.00	0.18	0.0	0.0
	Modified	0.49	0.0	0.0	0.00	0.0	0.0
	Combined	5.91	0.03	0.0	0.8	0.0	0.0
Sigmoid	BP	**34.63**	6.42	0.85	**20.14**	0.29	0.01
	PEPITA	21.24	**15.54**	**14.91**	15.69	**13.03**	**12.91**
	Modified	13.25	0.12	0.12	12.06	0.0	0.0
	Combined	16.95	7.02	5.77	9.28	4.70	4.41

4.2. Experimental Results

The classification accuracy results on the two data set are summarized in Table 1. For each type of activation function, we marked the best results in bold fonts and underlines. The results indicate that both the original PEPITA and the combined PEPITA perform competitively with BP in all cases. Especially, for the fashion MNIST data that is more challenging than MINIST, PEPITA method with sigmoid activation outperforms BP.

On the other hands, we can also see that the performance of the modified PEPITA is slightly lower than others. Unlike the original PEPITA using fixed transition matrix, the modified PEPITA uses loss gradient depending on each input, which may cause learning instability. However, from the result of the combined PEPITA, we can conclude that performance improvement is possible with a proper combination of the two perturbation methods. In addition, by comparing the two different types of activation functions, we find that the sigmoid function achieves better performance than ReLU, which was used in previous studies.

Table 2 presents a comprehensive analysis of the adversarial robustness of various learning methods when subjected to attacks generated from the MNIST dataset using both FGSM and PGD methods. The variable ε denotes the size of perturbation noise. When the ReLU activation function is used, BP shows the best robustness against a variety of adversarial attacks. When using the sigmoid activation function, BP is found to be the most robust to FGSM and PGD attacks when epsilon values are small. However, in the FGSM attack, PEPITA with sigmoid activation is the most robust for ε values greater than 0.2. Furthermore, the combined PEPITA method with sigmoid activation shows the second-highest robustness as ε increases. This trend is also observed in PGD attacks. As the value of ε increases, PEPITA with sigmoid activation or its combined version, shows greater robustness to PGD attacks compared to other methods.

The experimental results highlight the importance of selecting an appropriate activation function to enhance the robustness of learning methods under varying conditions. When using the sigmoid activation function, the combined PEPITA method can achieve comparable or slightly lower performance than BP with ReLU, which shows the best performance. As the value of ε increases, PEPITA with sigmoid activation consistently demonstrates greater robustness compared to ReLU. These findings suggest that the careful choice of activation function, such as the sigmoid, is crucial in different versions of PEPITA, especially to maintain robustness against higher levels of adversarial perturbation. This insight could guide future enhancements in the development of more resilient neural network models.

V. CONCLUSION

In this paper, we present a modified version of the PEPITA learning method in which a network is trained by means of two forward computations. Although the PEPITA method has demonstrated good performance by eliminating the backward computation, further research is needed to investigate its compatibility with different activation functions and its application in various fields. This study proposes a new approach to generate modulated input using the Fast Gradient Sign Method (FGSM) based on the similarity between the modulated input generation method of PEPITA and the adversarial sample generation method of deep learning and the performance of PEPITA was evaluated through a compatibility analysis with activation functions. The experimental results show that the proposed method and PEPITA are generally inferior in performance to BP but are relatively more robust to adversarial attacks as the epsilon value increases. In particular, the use of the sigmoid activation function has a positive impact on the robustness. Future research will integrate adversarial training techniques to improve the robustness of the PEPITA learning method. In addition, extensive experiments will be conducted on a variety of data sets and models to thoroughly analyze the performance of the method.

REFERENCES

[1] D.E. Rumelhart, G.E. Hinton and R.I. Williams, "Learning representations by back-propagating errors," Nature, vol. 323, pp. 533-536, 1986.

[2] F. Crick, "The recent excitement about neural networks", Nature, Vol. 337, No.6203, pp. 129-132, Jan, 1989.

[3] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance", Cognitive science, vol. 11, No. 1, pp. 23-63, 1987.

[4] M. Jaderberg, W. M. Czarnecki, S. Osindero, O. Vinyals, A. Graves, D. Siver and K. Kavukcuoglu, "Decoupled neural interfaces using synthetic gradients", 34th International Conference on Machine Learning, vol. 70, ICML' 17, pp. 1627-1635, 2017.

[5] W. M. Czarnecki, G. Swirszcs, M. Jaderberg, S. Osindero, O. Vinyals and K. Kavukcuoglu, "Understanding synthetic gradients and decoupled neural interfaces", 34th International Conference on Machine Learning, vol. 70, ICML' 17, pp. 904-912, 2017.

[6] L. Khacef, P. Klein, M. Cartiglia, A. Rubino, G. Indiveri, and E. Chicca, "Spike-based local synaptic plasticity: A survey of computational models and neuromorphic circuits", arXiv:2209.15536., 2022.

[7] J. Kendall, R. Pantone, K. Manickavasagam, Y. Bengio, and B. Scellier, "Training end-to-end analog neural networks with equilibrium propagation", arXiv:2006.01981., 2020.

[8] T. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, “Random synaptic feedback weights support error backpropagation for deep learning”, Nature Communications, vol. 7, Article 13276, Nov. 2016.

[9] Q. Liao, Z.L. Leibo and T. Poggio, “How Important Is Weight Symmetry in Backpropagation?,” 30th AAAI Conference on Artificial Intelligence, 2016.

[10] A. Nøkland, “Direct Feedback Alignment Provides Learning in Deep Neural Networks,” 30th Conference on Neural Information Processing Systems, 2016.

[11] B. Crafton, A. Parihar, E. Gebhardt and A. Raychowdhury, “Direct Feedback Alignment With Sparse Connections for Local Learning,” Frontiers in Neuroscience, vol. 13, pp 525, 2019.

[12] J. F. Kolen and J. B. Pollack. “Backpropagation without weight transport”, In Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN’94), vol.3, pp. 1375-1380, IEEE, 1994.

[13] M. Akrout, C. Wilson, P. Humphreys, T. Lillicrap and D. B. Tweed, “Deep learning without weight transport”, 33rd Conference on Neural Information Processing System, 2019

[14] D.-H.. Lee, S. Zhang, A. Fischer, and Y. Bengio. “Difference target propagation”, In ECML/PKDD, Machine Learning and Knowledge Discovery in Databases, pp. 498–515. Springer International Publishing, 2015.

[15] C. Frenkel, M. Lefebvre and D. Bol, “Learning Without Feedback: Fixed Random Learning Signals Allow for Feedforward Training of Deep Neural Networks,” Frontiers in Neuroscience, vol. 15, pp. 20, 2021.

[16] G. Hinton, “The forward-forward algorithm: Some preliminary investigations”, arXiv:2212.13345., 2022

[17] G. Dellaferrera, and G. Kreiman, “Error-driven input modulation: Solving the credit assignment problem without a backward pass”. In Proceedings of the 39th International Conference on Machine Learning, vol. 162, pp. 4937–4955. PMLR, 2022.

[18] R. F. Srinivasan, F. Mignacco, M. Sorbaro, M. Refinetti, A. Cooper, G. Kreiman, and G. Dellaferrera. “Forward Learning with Top-Down Feedback: Empirical and Analytical Characterization”, arXiv:2302.05440 [cs]. 2023.

[19] S. Bartunov, A. Santoro, B. Richards, L. Marris, G. Hinton and T. Lillicrap, “Assessing the scalability of biologically-motivated deep learning algorithms and architectures”, Advances in neural information processing systems, 31., 2018.

[20] I. Goodfellow, J. Shlens, and C. Szegedy. “Explaining and harnessing adversarial examples”, In International Conference on Learning Representations, 2015.

[21] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv preprint arXiv:1607.02533, 2016.

[22] T. F. Matilde ,O. Thomas, D. Giorgia, G. Benjamin, and P. Angeliki Pantazi, “Efficient Biologically Plausible Adversarial Training”, arXiv preprint arXiv : 2309.17348., 2023.

[23] D. O. Hebb, The Organization of Behavior:A Neuropsychological Theory, Hoboken, NJ: John Wiley & Sons, 1949

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, 1998.

[25] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms,” arXiv:1708.07747, 2017.