Initializing Perturbations in Multiple Directions for Fast Adversarial Training

Xunguang Wang, Ship Peng Xu, and Eric Ke Wang

Abstract—Recent developments in the filed of Deep Learning have demonstrated that Deep Neural Networks (DNNs) are vulnerable to adversarial examples. Specifically, in image classification, an adversarial example can fool the well-trained deep neural networks by adding barely imperceptible perturbations to clean images. Adversarial Training, one of the most direct and effective methods, minimizes the losses of perturbed-data to learn robust deep networks against adversarial attacks. It has been proven that using the fast gradient sign method (FGSM) can achieve Fast Adversarial Training. However, FGSM-based adversarial training may finally obtain a failed model because of overfitting to FGSM samples. In this paper, we proposed the Diversified Initialized Perturbations Adversarial Training (DIP-FAT) which involves seeking the initialization of the perturbation via enlarging the output distances of the target model in a random directions. Due to the diversity of random directions, the embedded fast adversarial training using FGSM increases the information from the adversary and reduces the possibility of overfitting. In addition to preventing overfitting, the extensive results show that our proposed DIP-FAT technique can also improve the accuracy of the clean data. The biggest advantage of DIP-FAT method: achieving the best balance among clean-data, perturbed-data and efficiency.

Index Terms—Adversarial Examples, Fast Adversarial Training, overfitting, robustness.

I. INTRODUCTION

Deep Neural Networks (DNNs) have achieved great success in a variety of applications, mainly including Computer Vision [1–3], Speech Recognition [4] and Natural Language Processing [5]. Despite achieving remarkable accuracy on many benchmark datasets, recent studies have shown that DNNs are vulnerable to adversarial examples, which are carefully crafted by adding imperceptible noises to natural inputs, but can fool the networks to output wrong predictions. The existence of Adversarial Examples illustrates the potential vulnerabilities of Deep Learning. In image classification task, although the networks are fed with printed adversarial examples obtained from a camera which simulates physical attacks, lots of images are classified incorrectly [6]. Another typical example about physical attacks is that the “STOP” sign with adversarial stickers mislead the perceptual systems into misclassifying it [7]. Thus, adversarial examples bring great challenges to the applications of the physical world and how to defend against them is an important and urgent problem that must be solved.

There have been a cohort of works on resisting various adversarial attacks. One of the most direct methods is adversarial training [8]–[13] on adversarial training greatly extend the standard adversarial training [8]. Madry et al. [9] used the more powerful PGD attack for adversarial training such that the model after trained can enhance the ability to resist general attacks such as FGSM. Another improvement way is to increase the diversity of adversarial examples, for which Tramèr et al. [10] proposed Ensemble Adversarial Training. Both of the PGD and the ensemble adversarial training improve robustness via better perturbations. The rest of methods [11]–[13] are to reduce convergence time of adversarial training.

It is evident that adversarial training methods [8]–[10] have led to significant progress in improving adversarial robustness, where using PGD adversary [9] is recognized as the most effective methods in adversarial training. However, PGD adversarial training is computationally expensive because PGD involves multiple random initializations and iterative gradient calculations when seeking adversarial perturbations. Fortunately, many methods [11]–[14] have been proposed to speed up this adversarial learning. By replacing PGD with a single-step FGSM [8] to produce perturbations, Wong et al. [14] proposed the fastest adversarial training algorithm called revisiting FGSM adversarial training which has very close adversarial accuracies to the standard PGD-based training [9]. Although the author claims to be able to train a robust classifier in 6 minutes [14], it is practically found that the target model is quite possible to fall into overfitting and thus cannot resist PGD attacks. The most likely reason is that the perturbations only covers a smaller subset compared to PGD.

In the present paper, we follow up the study of the FGSM-based training [14] and conduct further research on diversified random initialization of perturbations. Compared with the previous method [14], the main contributions of this paper are as follows:

1) The Random Diversified Initialization (RDI) technique is proposed to replace the simple random initialization of perturbations. It makes the generated perturbations more diverse and increases the difficulty of adversarial training.
2) The catastrophic overfitting of FGSM-based adversarial training was solved so that the target model can be trained with many epochs to fit all data.
3) The proposed method achieves the best balance among clean-data, perturbed-data and efficiency, which means that we trained the most robust model between clean-data and perturbed-data in the same short time.
II. RELATED WORKS

A. Adversarial Attacks

Since Szegedy et al. [15] discovered the properties of adversarial examples, various adversarial attacks have been proposed to fool a trained DNN by designing adversarial perturbations. There are two types of attacks: white-box attacks know the whole structure of the target model being attacked (i.e., FGSM [8], PGD [9] and C&W [16]), and black-box attacks only have access to the prediction of the target model (i.e., SBA [17] and ZOO [18]), which are briefly introduced as follows.

1) FGSM Attack: The Fast Gradient Sign Method (FGSM) [8] is an efficient one-step method to generate adversarial examples. Given an input $x$, it aims at quickly finding a perturbation direction of $x$ to increase the training loss function of the target model such that the model will classify $x$ incorrectly. Concretely, the adversarial perturbation is produced by calculating the gradient of the loss function with respect to $x$ and multiplying the gradient by a constant $\epsilon$:

$$x' = x + \epsilon \cdot \text{sign} (\nabla_x \mathcal{L}(x, y))$$  \hspace{1cm} (1)

where $y$ is the true class label for the input instance $x$, $\mathcal{L}$ is the loss function, $\nabla_x \mathcal{L}(x, y)$ is the gradient, $\text{sign} (\cdot)$ is the sign function, $\epsilon$ controls attack intensity, and $x'$ is the adversarial example obtained.

2) PGD Attack: The Projected Gradient Descent (PGD) is an iterative variant of FGSM and also generates $L_\infty$ adversarial perturbations. Given an input $x$, the process of PGD to obtain an adversarial example can be expressed as:

$$x_{i+1} = \text{Clip}_\epsilon \{x_i + \alpha \cdot \text{sign} (\nabla_x \mathcal{L} (x_i, y))\} \text{ for } i = 0 \text{ to } n$$  \hspace{1cm} (2)

where $\text{Clip}_\epsilon$ restricts $x_i$ in $\epsilon$-neighborhood, $x_0$ is a original image perturbed with $\epsilon$-ball random noise in $\ell_p$ norm, and $\alpha$ is the step size. PGD is considered to be the strongest first-order attack.

3) ODI-PGD Attack: The ODI-PGD [19] is a PGD-based attack using ODI (Output Diversified Initialization) [19] method which produce initial perturbations. The ODI changes the output logits in random direction and then calculates the perturbations by back propagation. In addition to PGD, ODI has strengthened the C&W [16] attack. Inspired by ODI, we also exploit output diversified initialization of perturbations. In contrast to that, our perturbations produced by RDI are more diverse and mainly strengthen adversarial training.

B. Adversarial Training methods

Adversarial training [8] is a simple defense method against adversarial attacks in which a model is retrained by adding perturb-data into training set at every training step for improving robustness of the model. With the development of adversarial attacks, adversarial training has been also improved continuously.

1) PGD Adversarial Training: PGD Adversarial Training [9] uses PGD as adversary. What is more, Madry et al. redefined the form of adversarial training:

$$\arg \min_\theta \mathbb{E} \left[ \max_{\|x'-x\|_\infty} \mathcal{L} (x', y) \right]$$  \hspace{1cm} (3)

where the goal of the inner maximization is to find adversarial examples which maximize the loss of the target model, while the outer minimization is a optimization process to obtain the parameters $\theta$ that minimize the loss caused by the adversarial inputs. Different with standard adversarial training [8], the PGD one only rely on adversarial examples to train the model. A model trained by PGD in standard training framework [8] can defend against FGSM, but not vice versa, because PGD attacks are stronger than FGSM.

2) Revisiting FGSM Adversarial Training: Revisiting FGSM Adversarial Training [14] is the fastest strategy to train a robust model against gradient-based attacks like FGSM and PGD. For example, under the author’s experimental setting, a CIFAR-10 [20] classifier only need 6 minutes and an ImageNet classifier is trained completely in 12 hours, reducing 10 and 38 hours respectively compared to free adversarial training [11]. Surprisingly, the key to significantly reducing training time but is only to replace PGD with FGSM. Despite the huge advantage on acceleration, this schema is easy to overfit and then cannot defend against PGD attacks, which is why the previous studies failed in using FGSM to train a robust model against other more powerful attacks.

III. PROPOSED TRAINING ALGORITHM

In this section, We first introduce the overview of the proposed algorithm, which incorporates the random diversified initialization of perturbations and FGSM-based adversarial training. Then, the detailed adversarial training and the random initialization will be given.

Algorithm 1 Revisiting FGSM Adversarial Training

Input: a classifier $f_0$, radius $\epsilon$, step size $\alpha$, training epochs $T$, size $M$ of a dataset

Output: $\theta$

for $t = 1...T$ do

for $i = 1...M$ do

//Perform FGSM adversarial attack

$\delta = \text{Uniform} (-\epsilon, \epsilon)$

$\delta = \delta + \alpha \cdot \text{sign} (\nabla_{x} \ell (f_0 (x_i + \delta), y_i))$

$\delta = \max (\min (\delta, \epsilon), -\epsilon)$

$\theta = \theta - \nabla_{\theta} \ell (f_0 (x_i + \delta), y_i)$ // Update model weights

end for

end for

A. FGSM-based adversarial training

In this work, we follow the algorithm called Revisiting FGSM Adversarial Training. This FGSM adversarial training uses FGSM as adversary, but it has the close defense score to the PGD-based training. And the pseudo-code for this training method is provided in Algorithm 1. Due to the FGSM only
calculates the gradient in one-step, the FGSM-based training is much faster than the training based on the iterative PGD attack. In addition, different with naive FGSM adversarial training \cite{Goodfellow2014}, random initialization for the perturbation is embedded before training, as shown in Algorithm 1. The random initialization can expand the space of perturbations so that the calculated perturbations cover a subset of PGD, which is the key to defend against PGD attacks successfully.

\textbf{B. Random Diversified Initialization for perturbations}

The perturbations of previous attacking methods \cite{Kurakin2016, Papernot2017, Madry2017} like PGD are directly sampled from a uniform distribution in $\epsilon$-ball norm of the input pixel space. Inspired by the novel random initialization strategy called Output Diversified Initialization (ODI), we proposed Random Diversified Initialization (RDI) for FGSM-based adversarial training.

Like ODI, RDI encourages the diversity in the output space to calculate the diversified perturbations in the input space. Given the input $x$ and the classifier $f$, the optimization problem of initializing the perturbations is formulated as:

$$\max_{\delta \in P_\epsilon} w_d^T f(x + \delta)$$

where $f(x + \delta)$ is the output logits of $f$, $w_d \in \mathbb{R}^C$ defines the direction of the diversification in $C$-dimension space ($C$ is the number of classes of $f$), and $P_\epsilon$ is the perturbation set constrained by $\epsilon$. The maximization in Eq. (4) aims to drive the output of $f$ away from the original output along the direction $w_d$ which is randomly sampled from the uniform distribution $[-1, 1]^C$. Similar with PGD, we adopt iterative gradients to update $\delta$:

$$\delta_{n+1} = \text{Clip}_e \left( \delta_n + \alpha_{rdi} \cdot \text{sign} \left( \nabla_\delta w_d^T f_\theta \left( x + \delta_n \right) \right) \right)$$

where $\delta_n$ is updated in the $n$-th iteration, $\text{Clip}_e$ constraints $\delta$ within the bound $[-\epsilon, \epsilon]$. Note that $\nabla_\delta$ is derived for $\delta$, which is different with ODI that calculates the gradient with respect to $x$.

\textbf{C. Diversified Initialized Perturbations Adversarial Training}

The combination of RDI and FGSM-based training is our proposed method, where RDI initializes the perturbations and FGSM-based training makes the model robust by adversarial learning. Depending on whether to reuse the perturbation $\delta$, we divide the RDI-FAT into two types: the previous training and the random initialization training.

As shown in Algorithm 2, the previous training is the way that used perturbations in current iteration is from the previous iteration. The idea of the previous scheme comes from free adversarial training \cite{Tsipras2018}. Another way is the random initialization training which initializes the perturbation in each batch of learning before RDI. The initialization training is summarized in Algorithm 3. $N_{rdi}$ is the number of RDI steps. In theory, the larger the $N_{rdi}$, the logits output by $f_\theta$ is away from the starting point. In order to control the cost of time, $N_{rdi}$ does not exceed 2 in general. In particular, in Algorithm 2, we recommend $N_{rdi} = 1$.

\begin{algorithm}
\caption{Diversified Initialized Perturbations Adversarial Training (DIP-FAT): Previous}
\textbf{Input:} a classifier $f_\theta$, radius $\epsilon$, step size $\alpha$ of FGSM, training epochs $T$, size $M$ of a dataset, number of RDI steps $N_{rdi}$, step size $\alpha_{rdi}$ of RDI
\textbf{Output:} $\theta$
\textbf{Initialize:} $\delta = 0$ // Initialize the perturbation with 0
\For{$t = 1 \ldots T$}
\For{$i = 1 \ldots M$}
\State $w_d = \text{Uniform} \left( (-1, 1)^C \right)$ // Randomize the direction vector
\For{$n = 1 \ldots N_{rdi}$}
\State $\alpha_{rdi} = \text{Uniform} \left( 0, 2\alpha \right)$
\State $g_n \leftarrow \nabla_\delta w_d^T f_\theta \left( x_i + \delta \right)$
\State $\delta \leftarrow \delta + \alpha_{rdi} \cdot \text{sign} \left( g_n \right)$
\State $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
\EndFor
\State // Perform FGSM adversarial attack
\State $\delta = \delta + \alpha \cdot \text{sign} \left( \nabla_\delta \ell \left( f_\theta \left( x_i + \delta \right), y_i \right) \right)$
\State $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
\State $\theta = \theta - \nabla_\theta \ell \left( f_\theta \left( x_i + \delta \right), y_i \right)$
\EndFor
\EndFor
\end{algorithm}

\begin{algorithm}
\caption{Diversified Initialized Perturbations Adversarial Training (DIP-FAT): Random initialization}
\textbf{Input:} a classifier $f_\theta$, radius $\epsilon$, step size $\alpha$ of FGSM, training epochs $T$, size $M$ of a dataset, number of RDI steps $N_{rdi}$, step size $\alpha_{rdi}$ of RDI
\textbf{Output:} $\theta$
\For{$t = 1 \ldots T$}
\For{$i = 1 \ldots M$}
\State $\delta = \text{Uniform} \left( (-\epsilon, \epsilon) \right)$ // Initialize perturbations in a random direction
\State $w_d = \text{Uniform} \left( (-1, 1)^C \right)$ // Randomize the direction vector
\For{$n = 1 \ldots N_{rdi}$}
\State $\alpha_{rdi} = \text{Uniform} \left( 0, 2\alpha \right)$
\State $g_n \leftarrow \nabla_\delta w_d^T f_\theta \left( x_i + \delta \right)$
\State $\delta \leftarrow \delta + \alpha_{rdi} \cdot \text{sign} \left( g_n \right)$
\State $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
\EndFor
\State // Perform FGSM adversarial attack
\State $\delta = \delta + \alpha \cdot \text{sign} \left( \nabla_\delta \ell \left( f_\theta \left( x_i + \delta \right), y_i \right) \right)$
\State $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
\State $\theta = \theta - \nabla_\theta \ell \left( f_\theta \left( x_i + \delta \right), y_i \right)$
\EndFor
\end{algorithm}

\section{IV. EXPERIMENTS}

\textbf{A. Experiment setup}

\begin{itemize}
\item \textit{1) Datasets:} The experiments were conducted on MNIST \cite{LeCun1998} and CIFAR-10 \cite{Krizhevsky2009} benchmarks. The MNIST \cite{LeCun1998} is a dataset for handwritten digit recognition with 70K 28 × 28 gray-scale images, where 60K images are training examples and 10K images are test examples. The CIFAR-10 \cite{Krizhevsky2009} con-
tains 50K training samples and 10K test samples of 32×32 color image for image classification.

2) Experimental environment and acceleration skills: All experiments in this paper are run on a single NVIDIA Tesla P100 (16GB). The deep learning framework is pytorch 1.4.0. Unlike wong et al. [14], we only use the cyclic learning rate [25], [26] to reduce the number of epochs required for training, because not all GPUs support the mixed-precision [27], which is more in line with the real situation.

B. Verified performance on MNIST

Since the FGSM-based training [14] benefits from the random initialization, we wonder if the RDI works for robustness of DNNs. To demonstrate RDI confers real robustness to the model, we followed the experiment [14] on MNIST [24]. We did not randomly initialize perturbations, in order to increase the credibility. We adopt algorithm 2 with $\epsilon = 0.3$, $N_{rdi} = 1$, and the other parameters as the same as [14]. The exact verification results can be seen in Table 1, where we can find that our result is closer to PGD adversarial training on MNIST than the FGSM method [14].

C. Verified performance on CIFAR-10

In this section, we test our proposed technique on CIFAR-10 to show the effect of the diversified initialization in FGSM-based adversarial training. We trained the PreAct ResNet18 architecture with radius $\epsilon = 8/255$ and $\alpha = 10/255$. The results are given in Table 2. We listed the PGD [9], the Free [11], the FGSM [14] and ours, where the PGD trained 40 epochs, the Free trained 96 epochs, the FGSM trained 30 epochs. To be fair, our methods also only trained 30 epochs.

The best adversarial accuracy (PGD, $\epsilon = 8/255$) was picked by the Free, and the best standard accuracy tested by the clean-data (the natural images) was picked by our previous method ($N_{rdi} = 2$). Our methods perform well on the clean-data. In contrast to the FGSM schema, our methods improve accuracy on both clean-data and perturbed-data only by increasing the tolerable time cost. In addition, we found that the result in $N_{rdi}$ is better than $N_{rdi} = 1$, which verifies that the perturbation is gradually far away from the starting point with the increase of calculation steps.

V. CONCLUSION

In this paper, we combined the diversified initialization for adversarial perturbations and FGSM-based adversarial training and proposed the random diversified initialization fast adversarial training (RDI-FAT). Through extensive experiments, the proposed adversarial training not only solves the problem of over fitting in FGSM training, but also improves the performance on both clean-data accuracy and perturbed-data accuracy.

| Method | Clean | PGD(=0.1) | PGD(=0.3) | FGSM(=0.1) |
|--------|--------|-----------|-----------|------------|
| PGD   | 98.99% | 97.33%    | 89.04%    | 97.7%      |
| FGSM  | 98.50% | 96.76%    | 88.34%    | 97.1%      |
| ours  | 98.80% | 97.09%    | 88.63%    | 97.6%      |

| Method | Clean | PGD(=0.1) | PGD(=0.3) | FGSM(=0.1) |
|--------|--------|-----------|-----------|------------|
| PGD    | 87.30% | 45.80%    | 1426.00   |
| Free (m=8) | 85.96% | 46.33%    | 351.68    |

| Method | Clean | PGD(=0.1) | PGD(=0.3) | FGSM(=0.1) |
|--------|--------|-----------|-----------|------------|
| PGD    | 86.02% | 42.57%    | 25.21     |
| random init | 84.01% | 45.25%    | 25.81     |

| N_{rdi} | previous init | previous | random init | random init |
|---------|---------------|----------|-------------|-------------|
| 1       | 86.40% | 45.57% | 54.80% |

Table I: Robustness of different adversarial training on MNIST

Table II: Standard and robust performance of various adversarial training methods on CIFAR-10 for $\epsilon = 8/255$ and their corresponding training times

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
[2] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.
[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.
[4] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal processing magazine, vol. 29, no. 6, pp. 82–97, 2012.
[5] T. Mikolov, K. Charif, L. Burget, J. Černocký, and S. Khudanpur, “Recurrent neural network based language model,” in Eleventh annual conference of the international speech communication association, 2010.
[6] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” arXiv preprint arXiv:1607.02533, 2016.
[7] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, “Robust physical-world attacks on deep learning visual classification,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.
[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” arXiv preprint arXiv:1412.6572, 2014.
[9] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” arXiv preprint arXiv:1706.06083, 2017.
[10] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “Ensemble adversarial training: Attacks and defenses,” arXiv preprint arXiv:1705.07204, 2017.
[11] R. F. A. Shafahi, M. Najibi, M. A. Ghiasi, Z. Xu, J. Dickerson, C. Snider, L. S. Davis, G. Taylor, and T. Goldstein, “Adversarial training for free!” in Advances in Neural Information Processing Systems, 2019, pp. 3353–3364.
[12] D. Zhang, T. Zhang, Y. Lu, Z. Zhu, and B. Dong, “You only propagate once: Accelerating adversarial training via maximal principle,” in Advances in Neural Information Processing Systems, 2019, pp. 227–238.
[13] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, “Freedb: Enhanced adversarial training for language understanding,” arXiv preprint arXiv:1909.11764, 2019.
[14] E. Wong, L. Wang, and J. Z. Kolter, “Fast is better than free: Revisiting adversarial training,” arXiv preprint arXiv:2001.05994, 2020.
[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” arXiv preprint arXiv:1312.6199, 2013.
[16] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in 2017 ieee symposium on security and privacy (sp). IEEE, 2017, pp. 39–57.
[17] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in Proceedings of the 2017 ACM on Asia conference on computer and communications security, 2017, pp. 506–519.
[18] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks.
without training substitute models,” in Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, 2017, pp. 15–26.

[19] Y. Tashiro, Y. Song, and S. Ermon, “Output diversified initialization for adversarial attacks,” arXiv preprint arXiv:2003.06878, 2020.

[20] A. Krizhevsky, G. Hinton et al., “Learning multiple layers of features from tiny images,” 2009.

[21] T. Zheng, C. Chen, and K. Ren, “Distributionally adversarial attack,” in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 2253–2260.

[22] F. Croce and M. Hein, “Minimally distorted adversarial examples with a fast adaptive boundary attack,” arXiv preprint arXiv:1907.02044, 2019.

[23] S. Gowal, J. Uesato, C. Qin, P.-S. Huang, T. Mann, and P. Kohli, “An alternative surrogate loss for pgd-based adversarial testing,” arXiv preprint arXiv:1910.09338, 2019.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE, vol. 86, no. 11, pp. 2278–2324, 1998.

[25] L. N. Smith, “Cyclical learning rates for training neural networks,” in 2017 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2017, pp. 464–472.

[26] L. N. Smith and N. Topin, “Super-convergence: Very fast training of neural networks using large learning rates,” in Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, vol. 11006. International Society for Optics and Photonics, 2019, p. 1100612.

[27] P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh et al., “Mixed precision training,” arXiv preprint arXiv:1710.03740, 2017.