Adaptive Adversarial Norm Space for Efficient Adversarial Training

Hui Kuurila-Zhang
hui.zhang@oulu.fi
Haoyu Chen
chen.haoyu@oulu.fi
Guoying Zhao*
guoying.zhao@oulu.fi

Center for Machine Vision and Signal Analysis
University of Oulu
Oulu, Finland

Abstract

Adversarial training draws increasing attention as it can improve the robustness of deep neural networks against adversarial examples. Recent research proposed to adaptively adjust the adversarial strategy for a better learning process. However, those approaches rely on cumbersome computations for getting the optimal adversarial strategy. This paper offers a novel perspective on adversarial strategies by examining the adversarial examples' norm space. We show that cyclically altering the adversarial norm space can significantly enhance the network's robustness. Based on the observations, we propose a simple yet effective Entropy-Guided Cyclical Adversarial Strategy (ECAS) to explicitly adjust the norm space of the adversarial examples, forming an elastic-perturbation mechanism in the adversarial training framework that adaptively perturbs models based on entropy. Extensive experiments demonstrate that our proposed method can achieve promising performances and substantially reduce computational time compared to state-of-the-art methods. Moreover, we also show that ECAS can be directly plugged into existing adversarial training methods to further boost performances. The implementation of ECAS is at https://github.com/huizhg/ECAS.

1 Introduction

Deep neural networks have achieved impressive success in many fields. Despite their remarkable performance, deep neural networks are found to be vulnerable to adversarial attacks [7, 16], which can fool the network by adding crafted human-imperceptible perturbations on clean input data. These attacks can pose severe threats to the security and reliability of deep neural networks, especially in safety-critical applications. A large number of research studies have been conducted to investigate the threat of adversarial attacks to deep neural networks and to develop techniques to defend against them. These studies have proposed various defense mechanisms, including adversarial training [6, 7, 11, 12, 18, 19, 20, 23, 24], certified defense [13], randomization-based defense [3], etc. Out of all the defense

* indicates the corresponding author.

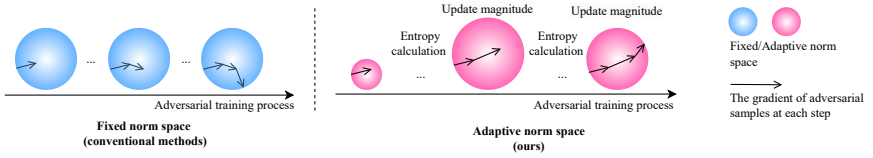

Figure 1: Adaptive entropy-guided adversarial norm space during adversarial training.

methods, adversarial training is considered one of the most effective methods because it significantly improves the robustness of models against adversarial attacks without any complex implementation.

Some recent works preliminarily prove that adjusting the adversarial strategies in adversarial training can benefit the robustness of the final model. Initially, the pioneering methods of adversarial training were with fixed adversarial strategies [12, 18, 20, 23] by default. In those methods, the hyperparameters (e.g., the magnitude of the perturbation) of the attacks are usually fixed during the whole training phase. More recent works use adaptive adversarial strategies [9, 17, 21], leading to better performances against the adversarial samples. For those variants with dynamic adversarial strategies during adversarial training, the attacks keep altering adaptively during training, e.g., the adversarial strength gets stronger with the increasing number of epochs. Although there are already some existing methods, how to effectively and efficiently find the optimal hyperparameters of attacks in adversarial training is still, to a large extent, an open problem.

Motivated by the recent works that attempted to automatically learn the adversarial strategies, we investigate adversarial strategies from the new perspective of the norm space of the adversarial examples. By analyzing models' learning behavior with existing adaptive learning strategies, we propose the Entropy-Guided Cyclical Adversarial Strategy (ECAS). Figure 1 demonstrates how ECAS adaptively alters the norm space of the adversarial examples during training. The contributions are summarized as follows:

- We focus on the adversarial training problem with adaptive adversarial strategies. We empirically identify that cyclically changing the adversarial norm space can improve the robustness of the network.
- A simple yet effective Entropy-Guided Cyclical Adversarial Strategy (ECAS) is introduced to periodically adjust the norm space of the adversarial examples, forming an elastic-perturbation mechanism in the adversarial training framework that adaptively perturbs models. A customized entropy term is proposed to guide the change of the norm space of the adversarial examples, leading to more adaptive and effective learning progress.
- Quantitative experimental results demonstrate that the proposed ECAS consistently improves the adversarial robustness of state-of-the-art methods. Most importantly, our method requires substantially less runtime than the compared methods. At last, we show that the proposed ECAS can be directly plugged into other existing adversarial training methods.

2 Related work

Adversarial training is one of the most effective defense techniques against adversarial attacks, and our proposed method ECAS is used to cooperate with it to boost its performance.

The basic concept of adversarial attack and adversarial training and some recent studies of them are introduced in this section.

2.1 Adversarial attack

An adversarial attack is a malicious attack that attempts to perturb the clean data by adding crafted human-imperceptible noise such that misleading the machine learning system to a wrong prediction. The perturbed data is called adversarial examples [16] and the algorithms used to generate adversarial examples are called adversarial attacks. Adversarial attacks can be roughly divided into two categories, white-box attacks, and black-box attacks. For white-box attacks, the attacker has access to all the information of the model, such as the parameters and structure of the network, which makes it easy to get the gradients. For black-box attacks, the attacker has no knowledge about the model and therefore has to rely on external observation and experimentation to gain information, which makes this kind of attack much harder to implement. In this paper, we will mainly focus on the white-box attack because it is considered more harmful, and it will be used for both adversarial training and robustness evaluation of the networks in this work. Fast Gradient Sign Method (FGSM) [7] is a popular and fast gradient-based white-box attack, which perturbs the clean data with only one step. The Projected Gradient Descent (PGD) attack [12] is an iterative attack, which perturbs the input data with multiple steps and projects the result from each step within a norm ball. The variants of PGD are named after by the number of their iterations, for example, PGD-10, and PGD-50. C&W attack [2] is an optimization-based attack, and the attack is generated by searching for the smallest perturbation that leads to incorrect classification by the network. AutoAttack (AA) [4] is one of the strongest adversarial attacks, that is parameter-free and user-independent. AA is an ensemble of two extensions of PGD attack (APGD-CE [4], APGD-DLR [4]) and two complementary attacks (FAB [5] and Square attack [1]).

2.2 Adversarial training

The existing adversarial training methods can be categorized into two types based on whether they use adaptive adversarial strategies. We introduce some related work of both types of training methods in this section.

Type 1: Adversarial training with fixed adversarial strategies Madry *et al*. proposed the standard adversarial training method PGD-AT [12]. Many variants of standard adversarial training improved the performance of [12] by strengthening the inner maximization problem, i.e., the generation of the adversarial examples. Zhang *et al*. proposed TRADES [23], which uses the Kullback-Leibler divergence between the model's output on clean input and perturbed input as a regularization term on the loss function to push the model's decision boundary away from the adversarial example, thus improving the model's robustness. An improved version of [12] proposed by Rice *et al*. [14] utilizes early stopping to solve the overfitting problem in adversarial training. Wang *et al*. proposed MART [18], an extension of TRADES, improving the adversarial robustness by explicitly differentiating the misclassified examples during training. Adversarial Weight Perturbation (AWP) [20], proposed by Wu *et al*., boosts the adversarial robustness by flattening the weight loss landscape through weight perturbation.

Type 2: Adversarial training with adaptive adversarial strategies Wang *et al*. [17] proposed a training strategy that gradually increases the convergence quality of adversarial examples during training. The convergence quality of adversarial examples also represents

the attacking strength of the adversarial examples. Jia *et al*. [9] proposed a learnable adversarial strategy adversarial training (LAS-AT) method, which employs a strategy network to learn the optimal adversarial strategies. DAAT, proposed by Yang *et al*. [21], used a calibration network to adjust the magnitude of the perturbation ball of each instance to avoid adversarial example from crossing the decision boundary of the calibration network.

Although all of the type 2 methods can improve the performance of the standard adversarial training methods, the additional computational costs from generating the adversarial strategies were not considered. For example, both LAS-AT and DAAT used another network to assist in generating the adversarial strategies, which significantly increases computational complexity. To address this issue, we propose ECAS, an efficient approach to generate adversarial strategies which can boost the existing adversarial training methods without adding too much computational cost.

3 Method

In this section, we first define the problem of adversarial training and then investigate adaptive adversarial strategies from the perspective of norm space of the adversarial examples, which leads to a clear observation that models' robustness enjoys the benefit from cyclic norm space changes. Based on this, we propose an Entropy-Guided Cyclical Adversarial Strategy (ECAS) to explicitly change the norm space via the guidance of customized entropy terms such that each instance has its own tailored norm space. Note that the type of the norm is fixed to l_∞ in this study, and changing the norm space does not include changing the type of the norm space but only the magnitude.

3.1 Problem definition

The adversarial training problem can be formulated as the following equation:

$$\min_{w} \frac{1}{n} \sum_{i=1}^{n} \max_{||x_i' - x_i||_p \leq \varepsilon} l(f_w(x_i'), y_i), \tag{1}$$

where w represents the parameters of the model, n is the number of the training examples, x_i' is the perturbed data and the perturbation is bounded within l_p ε-ball, y_i is the true label of example x_i, $f_w(\cdot)$ is the target model with parameters w, and $l(\cdot)$ is the loss function. The inner maximization part represents the process of generating adversarial examples and the outer minimization part is the same as the standard training procedure of deep neural networks.

The perturbation budget ε in the inner maximization term of Eq. (1) represents the magnitude of norm space of the perturbation. We denote the perturbation with δ, and $\delta = ||x' - x||$. In this work, we use l_∞ norm to measure the magnitude of the norm space, thus every mentioned "norm" or its symbolic representation $||\cdot||$ in this paper refers to l_∞ norm by default. The goal of inner maximization in Eq.(1) is to search a δ in the norm space (an l_∞ norm ball, determined by ε) that maximizes the loss function $l(f_w(x_i'), y_i)$ in Eq. (1).

3.2 Cyclically changeable adversarial norm space

The evidence from [9, 17] demonstrates that increasing the magnitude of the norm space of adversarial examples during training can potentially increase the robustness of the network.

By analyzing the behaviors of the output strategies from LAS-AT [9], we find that norm space is not continually expanding linearly but also has shrinking behaviors during training. The findings inspire us to cyclically change the norm space of the adversarial sample to mimic the learning pattern at a low computational cost instead of using an extra strategy network which is computationally heavy and non-explainable.

The size of the norm space is altered according to Eq. (2),

$$\varepsilon_n = \frac{\varepsilon_{\max} - \varepsilon_{\min}}{\alpha} \times \mathrm{mod}(\frac{n}{\alpha}) + \varepsilon_{\min}, \tag{2}$$

where n is the index of the current epoch, $\varepsilon_{\max}$ and $\varepsilon_{\min}$ represent the upper bound and lower bound of the size of the norm space, $\varepsilon_n \in \{\varepsilon_{\min}, ..., \varepsilon_{\max}\}$ and α determine the number of elements in this set and the gap between adjacent ε, and the $\mathrm{mod}(\cdot)$ is a modulo operation. The magnitude of the norm space is changed at every epoch following the scheduler in Eq. (2) during training. In the experiment, we explore different schedulers to find the optimal strategy for adjusting the magnitude of the norm space. Please refer to Section 4.1 for details.

3.3 Entropy-guided constraints on norm space

To make the magnitude of the norm space fit well for each instance, we further propose the Entropy-Guided Adversarial Strategy, which is applied on top of the scheduler shown in Eq. (2). The combination of them becomes Entropy-Guided Cyclical Adversarial Strategy (ECAS). In information theory, entropy [15] is a measure of the uncertainties of the information. The entropy of a discrete random variable X is calculated by Eq. (3)

$$H(X) := -\sum_{x \in \chi} p(x) \log p(x), \tag{3}$$

where χ includes all the possible values of the variable X. In our case, $p(x)$ is the probability of each class.

If the neural network is a classifier, its output is a probability distribution. The entropy of the probability distribution indicates how much information the network can deliver. For example, if the probabilities for all classes are the same, that means the network does not give us any valuable information from its prediction. This case also means that the network has not learned enough from the data or it has not had enough knowledge to give an affirmative prediction. With this property from entropy, we can use it to measure the uncertainties of the network's prediction of each instance. We utilize the entropy of the probability distribution of each instance to adjust the adversarial strength for attacking each of them.

The details of ECAS are shown in Algorithm 1. We compute the average entropy of the outputs from an untrained model using the training data and denote it as *h_high*. Similarly, *h_low* is calculated from the outputs of a well-trained model. The details for choosing *h_low* and *h_high* are provided in the **supplementary material**. If the entropy of the output of an instance is higher than *h_high*, which means the network is struggling with the classification and the instance is close to the decision boundary. In this case, we should shrink the norm space to avoid accidentally forcing the adversarial example across the decision boundary too much which leads to decreasing accuracy on clean data. On the contrary, if the entropy of the output of an instance is low, a wider norm space should be used. This is because the network is confident of the prediction, and there is a high chance that the instance is not close to the boundary, a larger perturbation would not only increase the robustness but also not hurt the

accuracy of clean data. In Algorithm 1, we use a random ε among the three biggest *epsilons* to avoid overfitting according to our empirical evidence. The *epsilon_space* in Algorithm 1 refers to the set of ε, $\varepsilon_n \in \{\varepsilon_{\min}, ..., \varepsilon_{\max}\}$, and it is defined in Section 3.2.

Algorithm 1 ECAS

Input: Current epoch *epoch*, epsilon space *epsilon_space*, entropy of a batch of inputs H, lower entropy *h_low*, higher entropy *h_high*

Output: Adversarial strength for this batch at the current epoch *epsilon_batch*

function ECAS(*epoch*, *epsilon_space*, H, *h_low*, *h_high*):
  big_epsilons ← The three biggest epsilons from *epsilon_space*
  epsilon_batch ← Initialize the epsilons with ones
  ε ← CYCLICAL_EPSILON(*epoch*, *epsilon_space*)
  epsilon_batch ← *epsilon_batch* $\times \varepsilon$
  epsilon_batch[$H >$ *h_high*] ← The smallest ε from *epsilon_space*
  epsilon_batch[$H <$ *h_low*] ← Take a random item from (*big_epsilons*)
  return *epsilon_batch*

3.4 Optimization

After we define the ECAS, we can start to conduct the adversarial training and demonstrate how to use it to control the size of the norm space of the adversarial examples during training. The adversarial example is generated by Eq. (4).

$$x_i' = \max_{||x_i'-x_i||_p \leq \varepsilon_i} l(f_w(x_i'), y_i), \tag{4}$$

The only difference between Eq. (4) and the maximization part of Eq. (1) is that the ε in Eq.(4) is generated by ECAS and it is instance-wise. The parameter of the network $f_w(\cdot)$ is updated by Eq. (5),

$$w \leftarrow w - \eta \frac{1}{n} \sum_{i=1}^{n} \nabla_w l(f_w(x_i'), y_i), \tag{5}$$

where w represents the parameters of the network, η is the learning rate, n is the number of samples in a mini-batch, and $\nabla_w l(\cdot)$ is the gradient of the loss function.

4 Experiments

To evaluate the performance of ECAS, we conduct intensive experiments on two datasets, CIFAR-10 [10] and CIFAR-100 [10], and both of them are popular benchmark datasets for training deep neural networks for computer vision tasks. We chose these two datasets because they were commonly used by the adversarial training methods we compared [9, 14].

Experimental Settings. The proposed adversarial training strategy is tested on ResNet18 [8] and WideResNet34-10 (WRN34-10) [22]. The experiments are conducted on NVIDIA Tesla V100. To make a fair comparison between ECAS and the state-of-the-art (SOTA) adversarial training methods, such as LAS-AT [9], we use their original hyperparameters in our settings. We apply our ECAS on AWP [20], TRADES [23], and the early stop version of PGD-AT [14], and denote them with their own names with "ECAS-" prefix. Similarly,

the methods that use the LAS-AT framework are named with the prefix "LAS-", such as LAS-AWP (refer to **supplementary material** for details).

Clean accuracy and robust accuracy. We evaluate both the "clean accuracy" and "robust accuracy" of all the models, where "clean accuracy" refers to the accuracy of the model tested on unperturbed clean data, and "robust accuracy" refers to the accuracy of the models on perturbed data. The perturbed data includes the adversarial examples generated by PGD [17] (with iterations 10, 20, and 50 and perturbation budget 8/255), C&W [2], and AA [4].

4.1 Ablation study

Verification of different schedulers. As mentioned in Section 3.2, here we explore different schedulers to change the magnitude of the adversarial norm space. The results are shown in the first four rows of Table 1. "Fixed" is the baseline scheduler, which means the magnitude of the norm space is fixed with ϵ set to a fixed value (8/255) during training. The "Linear" scheduler updates ϵ linearly in every epoch. "Cyclic" alters ϵ cyclically in every epoch following the definition from Eq. (2). We also verify the changing frequency of the norm space by experimenting with "Cyclic-batch", where we changes ϵ cyclically at every batch, i.e., the same frequency as the weights update of the network. "Cyclic-batch" can be regarded as a high-frequency variant of Eq. (2) if we change the number of current epochs to the number of current batches.

It can be seen from Table 1 that "Linear" increases the robustness but reduces the clean accuracy. "Cyclic-batch" does not affect the robustness much compared with the "Fixed" scheduler. The failure of "Cyclic-batch" is caused by the high update rate of adversarial strength which has already changed before the network has learned enough from the perturbed data with the adversarial strength from the last round. "Cyclic" scheduler obtains the best performance compared to the other three schedulers (Fixed, Linear, and Cyclic-batch). Therefore, we choose "Cyclic" as the scheduler for ECAS (see more details in the **supplementary materials**).

Contribution of each component. The contribution made by each component is also demonstrated in Table 1. The first row ("Fixed") is the set as the baseline where no component is added. Starting from the baseline setting, the fourth row "+ Cyclic""shows that adding the cyclic scheduler can effectively improve the results, e.g., from 0.8517 to 0.8634 for Clean and from 0.5169 to 0.5325 for AA. On top of the cyclic scheduler, the last row of Table 1, "+ Cyclic + entropy" as our full method, shows that adding the component of the entropy-guided constraints can further improve the performance of the models against adversarial samples. Note that the Cleaning accuracy becomes slightly lower when the entropy-guided constraints are added, which is because the entropy-guided algorithm is prioritized to improve the robust accuracy and tends to increase ϵ when the condition is satisfied. The difference in running time per epoch between different schedulers is negligible, approximately 0.3% (for example, 1 or 2 seconds difference out of 662s).

4.2 Running time analysis

Our ECAS method improves the clean and robust accuracy of the baseline methods (refer to Table 4), reaching a comparable level to LAS-AT without significantly increasing computational complexity. We compare the extra cost of the training time for the first epoch of each tested training method, and the results are shown in Table 2 and Table 3. The running time

Scheduler	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
Baseline (Fixed)	0.8517	0.5607	0.5508	0.5488	0.5391	0.5169
+ Linear	0.8351	0.5704	0.5587	0.5548	0.5497	0.5273
+ Cyclic (batch)	0.8682	0.5648	0.5482	0.5451	0.5495	0.5243
+ Cyclic	**0.8634**	**0.5766**	**0.5670**	**0.5637**	**0.5567**	**0.5325**
+ Cyclic + entropy	**0.8632**	**0.5780**	**0.5680**	**0.5648**	**0.5589**	**0.5343**

Table 1: Performance comparison of models (WRN34-10) trained by PGD-AT [14] with different schedulers on CIFAR-10. Results in bold are from our methods.

for the first epoch is chosen because we want to compare the running time under the most time-consuming case, and the first epoch takes longer time than the rest epochs.

"+L:" and "+Ours:" in the first column of Table 2 and Table 3 indicates what framework (LAS-AT or ECAS) is used on top of the baseline methods. Besides the extra running time (1st and 2nd rows), we also include comparisons of clean accuracy (shown in 3rd and 4th rows of Table 2 and Table 3) and robust accuracy (shown in 5th and 6th rows) of the baseline methods when integrating with LAS-AT or ECAS framework. The robust accuracy is evaluated by AA [4].

From Table 2 and Table 3, we can see that the performance improvement from LAS-AT is achieved by sacrificing the training time, and the extremely long time is caused by brute force parameter searching. ECAS is more lightweight, and it enhances the performance of the baseline methods without increasing too much of the training time (less than 200s for one epoch).

	CIFAR-10			CIFAR-100		
Framework	PGD-AT	TRADES	AWP	PGD-AT	TRADES	AWP
+L: runtime	681.0s	6048.4s	5587.6s	428.3s	6051.4s	1081.2s
+Ours: runtime	**89.4s**	**52.0s**	**59.7s**	**150.5s**	**166.0s**	**149.3s**
+L: clean	0.8623	0.8524	0.8774	0.6180	0.6062	0.6489
+Ours: clean	0.8632	0.8399	0.8817	0.6116	0.5869	0.6477
+L: robust	0.5358	0.5415	0.5552	0.2903	0.2812	0.3077
+Ours: robust	0.5343	0.5202	0.5454	0.2883	0.2824	0.2978

Table 2: Comparison of the extra running time (1st and 2nd rows) (tested on CIFAR-10 and CIFAR-100 with WRN34-10) when integrating the LAS-AT ("+L:" in the table) or ECAS ("+Ours:") to the baseline methods, and their clean accuracy (3rd and 4th rows) and robust accuracy (5th and 6th rows).

4.3 Comparisons with SOTA methods

In this section, we compare the performance of our method ECAS with the SOTA method, LAS-AT, which is the same type of framework as ours that is used together with other adversarial training methods. We also make comparisons with various adversarial training methods, PGD-AT [14], TRADES [23], FAT [24], MART [18], and AWP [20]. We choose to compare our method ECAS with LAS-AT on the base method AWP because LAS-AT achieves its best performance when combined with AWP. It can be seen from Table 4 that

	CIFAR-10		
Framework	PGD-AT	TRADES	AWP
+L: runtime	129.0s	1127.3s	982.8s
+Ours: runtime	**27.8s**	**22.6s**	**21.5s**
+L: clean	0.8199	0.8204	0.8261
+Ours: clean	0.8139	0.7861	0.8316
+L: robust	0.4974	0.4975	0.4986
+Ours: robust	0.4994	0.4883	0.4964

Table 3: Comparison of the extra running time (1st and 2nd rows) (tested on CIFAR-10 with ResNet18) when integrating the LAS-AT ("+L:" in the table) or ECAS ("+Ours:") to the baseline methods, and their clean accuracy (3rd and 4th rows) and robust accuracy (5th and 6th rows).

Method	Clean	PGD-10	PGD-20	PGD-50	C&W	AA
PGD-AT [14]	0.8517	0.5607	0.5508	0.5488	0.5391	0.5169
TRADES [23]	0.8572	0.5675	0.5610	0.5590	0.5387	0.5340
FAT [24]	<u>0.8797</u>	0.5031	0.4986	0.4879	0.4865	0.4748
MART [18]	0.8417	0.5898	0.5856	0.5806	0.5458	0.5110
AWP [20]	0.8557	0.5892	0.5813	0.5792	0.5603	0.5390
LAS-AWP [9]	0.8774	**0.6109**	**0.6016**	**0.5979**	**0.5822**	**0.5552**
ECAS-AWP (ours)	**0.8817**	<u>0.6038</u>	<u>0.5910</u>	<u>0.5875</u>	<u>0.5750</u>	<u>0.5454</u>

Table 4: Test result on CIFAR-10 with WRN34-10, the best performance is shown in bold, and the second best is marked underlined

both LAS-AT and ECAS improve both the clean and robust accuracy of the base method, and ECAS reaches a comparable level as LAS -AT in terms of robust accuracy and surpasses LAS-AT on the clean accuracy.

4.4 Visualization of the norm space learned by ECAS

We explore how the distribution of the magnitude of the norm space (ε) changes when we apply ECAS to it. The result is demonstrated in Figure 2, which is done with PGD-AT integrated with ECAS (ECAS-PGD-AT). The purple bars on the top refer to the percentages of the ε after the entropy-guided adjustment, and the bars under them represent the percentages of the ε that are not adjusted but following the cyclical scheduler. The different colors represent the different values of the magnitude of ε, and the range of ε is from 3 to 12. It can be seen that the percentage of the adjusted ε has a sudden rise at epoch 99, which is because the learning rate decreases ten times after epoch 99, and the network's robustness has a sharp improvement. This leads to a sudden decrease in the entropy of the network's output, and if it is lower than the lower bound h_low, ε is adjusted to be larger.

5 Conclusion

We propose an entropy-guided cyclical adversarial strategy to adaptively alter the magnitude of the norm space of the adversarial examples at an instance-wise level, such that both the

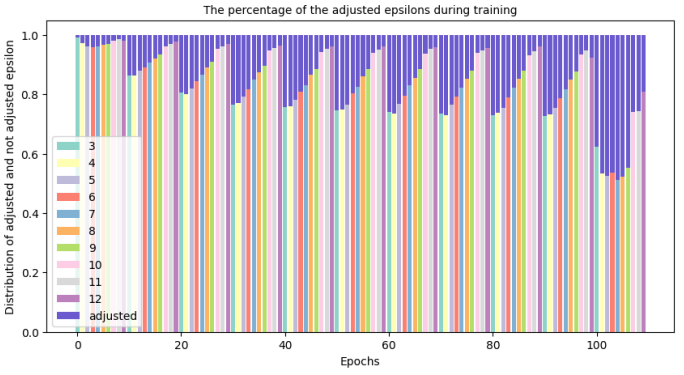

Figure 2: The percentage distribution of the adjusted and not adjusted magnitude of the norm space in ECAS-PGD-AT during training on the CIFAR-10 dataset

clean accuracy and robust accuracy of the network are improved. It is well-known that there is a trade-off between clean accuracy and robust accuracy in adversarial training. Increasing the epsilon norm ball size would raise the robust accuracy but lower the clean accuracy. Altering the norm ball size cyclically will result in a phenomenon that the peaks of clean accuracy and robust accuracy appear alternatively and we believe that this can help the optimization process jump out of the local minima and search for a better solution that balances both the clean and robust accuracy. The extensive experiments show that our method is easy to integrate with other adversarial training methods and improves their performance to a comparable level as the SOTA method without adding too much extra cost to the computation as the SOTA method does.

6 Acknowledgement

This work was supported by the Research Council of Finland (former Academy of Finland) Academy Professor project EmotionAI (grants 336116, 345122), the University of Oulu & Research Council of Finland Profi 7 (grant 352788), and Infotech Oulu. As well, the authors wish to acknowledge CSC – IT Center for Science, Finland, for computational resources.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: A query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer International Publishing, 2020.

[2] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2016.

[3] Jeremy Cohen, Amir Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[4] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020.

[5] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, pages 2196–2205. PMLR, 2020.

[6] Junhao Dong, Seyed-Mohsen Moosavi-Dezfooli, Jianhuang Lai, and Xiaohua Xie. The enemy of my enemy is my friend: Exploring inverse adversaries for improving adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24678–24687, 2023.

[7] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Las-at: adversarial training with learnable attack strategy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13398–13408, 2022.

[10] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009.

[11] Souvik Kundu, Sairam Sundaresan, Massoud Pedram, and Peter A Beerel. Float: fast learnable once-for-all adversarial training for tunable trade-off between accuracy and robustness. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2349–2358, 2023.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[13] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018.

[14] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pages 8093–8104. PMLR, 2020.

[15] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x.

[16] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[17] Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *International Conference on Machine Learning*, pages 6586–6595. PMLR, 2019.

[18] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

[19] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020.

[20] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020.

[21] Shuo Yang and Chang Xu. One Size Does NOT Fit All: Data-Adaptive Adversarial Training. In *European Conference on Computer Vision*, volume 13665, pages 70–85. Springer Nature Switzerland, 2022. ISBN 978-3-031-20064-9 978-3-031-20065-6. doi: 10.1007/978-3-031-20065-6_5.

[22] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.87.

[23] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019.

[24] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *International conference on machine learning*, pages 11278–11287. PMLR, 2020.