Neural Networks 150 (2022) 1–11

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Towards improving fast adversarial training in multi-exit network☆

Sihong Chen [a], Haojing Shen [a], Ran Wang [b,c,d], Xizhao Wang [a,c,d,*]

[a] College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China
[b] College of Mathematics and Statistics, Shenzhen University, Shenzhen, 518060, China
[c] Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University, Shenzhen, 518060, China
[d] Shenzhen Key Laboratory of Advanced Machine Learning and Applications, Shenzhen University, Shenzhen, 518060, China

ARTICLE INFO

Article history:
Received 9 September 2021
Received in revised form 16 January 2022
Accepted 18 February 2022
Available online 25 February 2022

Keywords:
Adversarial robustness
Adversarial defense
Fast adversarial training
Multi-exit network

ABSTRACT

Adversarial examples are usually generated by adding adversarial perturbations on clean samples, designed to deceive the model to make wrong classifications. Adversarial robustness refers to the ability of a model to resist adversarial attacks. And currently, a mainstream method to enhance adversarial robustness is the Projected Gradient Descent (PGD). However, PGD is often criticized for being time-consuming during constructing adversarial examples. Fast adversarial training can improve the adversarial robustness in shorter time, but it only can train for a limited number of epochs, leading to sub-optimal performance. This paper demonstrates that the multi-exit network can reduce the impact of adversarial perturbations by outputting easily identified samples at early exits. Therefore, we can improve the adversarial robustness. Further, we find that the multi-exit network can prevent catastrophic overfitting existing in single-step adversarial training. Specifically, we find that, in the multi-exit network, (1) the norm of weights at a fully connected layer in a non-overfitted exit is much smaller than that in an overfitted exit; and (2) catastrophic overfitting occurs when the late exits have weight norms larger than the early exits. Based on these findings, we propose an approach to alleviating the catastrophic overfitting of the multi-exit network. Compared to PGD adversarial training, our approach can train a model with decreased time complexity and increased empirical robustness. Extensive experiments have been conducted to evaluate our approach against various adversarial attacks, and the experimental results demonstrate superior robustness accuracies on CIFAR-10, CIFAR-100 and SVHN.

1. Introduction

Adversarial examples are delicately crafted samples, the attacker tries to deceive the model by adding adversarial perturbations which are invisible for human (Szegedy, Zaremba, Sutskever, Bruna, Erhan, Goodfellow, & Fergus, 2014). As the deep neural network (DNN) is widely used in various challenging machine learning tasks in real life, the threat of adversarial examples is receiving particular attention in the deep learning community.

A large body of defense methods have been proposed to alleviate this problem. Based on the structures and characteristics of DNNs, the defense methodologies can be categorized as Adversarial Training and Certified Robustness. The former feeds model with both clean samples and adversarial examples, for example, Madry, Makelov, Schmidt, Tsipras, and Vladu (2018) use PGD adversarial examples for training; while the latter proposes to use new regularization schemes which provably improves adversarial robustness, for instance, Lyu, Huang, and Liang (2015) develop gradient regularization methods to penalize the gradient of loss function. Among those defense methods, PGD adversarial training is widely accepted by the public since it can get better adversarial robustness and perform well in different problems (Madry et al., 2018). However, PGD adversarial training requires high computational costs. Some research attempts to overcome this issue, *i.e.*, adversarial training based on a weaker single-step attack, which is much more efficient.

Wong, Rice, and Kolter (2020) first discovered 'catastrophic overfitting', which is a phenomenon that accuracy against Projected Gradient Descent (PGD) attack drops significantly. It happens while conducting Fast Gradient Sign Method (FGSM) adversarial training. After that, many methods have been proposed to solve this problem. Kim, Lee, and Lee (2021) used

☆ This work was supported in part by Natural Science Foundation of China (Grants 61732011, 62176160, 61976141), the Natural Science Foundation of Shenzhen, China (University Stability Support Program no. 20200804193857002), and in part by the Interdisciplinary Innovation Team of Shenzhen University, China.

* Corresponding author at: College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, 518060, China.
E-mail addresses: 1910272011@email.szu.edu.cn (S. Chen), 1900271023@email.szu.edu.cn (H. Shen), wangran@szu.edu.cn (R. Wang), xizhaowang@ieee.org (X. Wang).

https://doi.org/10.1016/j.neunet.2022.02.015
0893-6080/

checkpoints (*i.e.*, early stopping) to prevent catastrophic overfitting. Andriushchenko and Flammarion (2020) proposed a new regularization method by using gradient alignment to solve catastrophic overfitting. Li, Wang, Jana, and Carin (2020) proposed using PGD adversarial training temporarily when catastrophic overfitting occurs. However, these approaches can only get a suboptimal solution (Kim et al., 2021), or have high time complexity (Andriushchenko & Flammarion, 2020; Li et al., 2020). Thus, the main problem to be solved in this paper is to make the model robust and accurate, at the same time reduce the training time.

In this paper, we first investigate the relationship between depth and influence of adversarial perturbations. The experimental results demonstrate that the multi-exit network can reduce the impact of adversarial perturbations, thus will get better adversarial robustness. Then we show that the multi-exit network can alleviate the phenomenon of FGSM catastrophic overfitting and recover from overfitting. The underlying reasons are discovered empirically. Based on the above findings, we propose a new training scheme for the multi-exit network, which can further alleviate the phenomenon of catastrophic overfitting. Extensive experiments have evaluated our approach against various adversarial attacks, including PGD, C&W, BIM, MIM and AutoAttack (AA). Experiments show promising results comparable to PGD adversarial training and its extensions on CIFAR-10, CIFAR-100 and SVHN.

Our contributions can be concluded as follows:

- We show that the impact of adversarial perturbations on feature space will be amplified as the network goes deeper, and the multi-exit network can reduce the impact of adversarial perturbations by outputting easy-identified samples at early exits.
- We find that the multi-exit network can alleviate the catastrophic overfitting of FGSM adversarial training, and the reason for alleviation is analyzed theoretically and experimentally.
- Based on the above observations, we suggest a simple method by adopting different penalty term coefficients to minimize the weights of fully connected layers, which can alleviate the phenomenon of catastrophic overfitting of multi-exit networks.
- Experimentally, our method is evaluated against various adversarial attacks, including FGSM, BIM, MIM, PGD, C&W and AutoAttack. The experiment shows promising results comparable to PGD adversarial training and its extensions.

The rest of this paper is organized as follows. Section 2 discusses related works about adversarial learning and multi-exit networks. Section 3 talks about the multi-exit network and adversarial attack and defense for them. Section 4 claims our proposed method. Section 5 experimentally demonstrates the effectiveness of the proposed method, and Section 6 makes some discussions of this paper.

2. Related work

We focus on classification tasks over samples $\{(\mathbf{x}, y)\} \in (\mathcal{X}, \mathcal{Y})$. Given a loss function $\mathcal{L}$, we can learn a classifier $f_\theta : \mathcal{X} \to \mathcal{Y}$ which is parameterized by θ. Considering an original sample $\mathbf{x} \in \mathcal{X}$, bound of perturbation ϵ, perturbation δ and a distance metric $\mathcal{D}(\cdot, \cdot)$ (usually we use ℓ_p metric). The definition of adversarial examples $x' = x + \delta$ is as follows:

$$\arg\min_{\delta} f_\theta(\mathbf{x} + \delta) \neq f_\theta(\mathbf{x}) \ \text{ s.t. } \|\delta\|_p < \epsilon \tag{1}$$

2.1. Adversarial attacks

Since Szegedy et al. (2014) proposed the concept of adversarial examples, many researchers have extensively proposed different attack methods. Gradient-based attacks have full access to the model, which is a kind of white-box attack (Yuan, He, Zhu, & Li, 2019). Goodfellow, Shlens, and Szegedy (2014) proposed a simple Fast Gradient Sign Method (**FGSM**), which performs a gradient ascent by using a first-order approximation of the loss function. Kurakin, Goodfellow, and Bengio (2017) extended FGSM by applying FGSM multiple times with small step size and clip the image in each iteration to ensure that the new image is in the ϵ-neighborhood of the original image, and named it Basic Iterative Method (**BIM**). In addition, aiming at escaping from local maxima during iterations of BIM, Momentum-based Iterative Method (**MIM**) (Dong, Liao, Pang, Su, Zhu, Hu, & Li, 2018) integrated momentum term into the iterative attack process. Projected Gradient Descent (**PGD**) (Madry et al., 2018) is also a variant of BIM. In this method, perturbation is initialized within a l_∞ ball, and the new image will be projected in each iteration. However, those methods cannot find the minimal perturbation necessary to change the class of a given input. To solve this problem, Croce, Andriushchenko, and Hein (2019) proposed Fast Adaptive Boundary Attack, which minimizes the norm of the perturbation necessary to achieve a misclassification. Apart from this, another type of white-box attack is the optimization attack. Carlini and Wagner (**C&W**) attack (Carlini & Wagner, 2017) designed a specific objective function for generating adversarial examples. Fan et al. (2020) formulated the sparse adversarial attack as a mixed-integer programming (MIP) problem to optimize perturbation magnitudes and binary selection factors jointly. AutoAttack (**AA**) (Croce & Hein, 2020) proposed two extensions of the PGD-attack and combine them with two complementary attacks, thus formed a new ensemble attack.

In addition to the white-box attack, another type of attack is the black-box attack. In the black-box attack, the role of the attacker is more like a regular user, and the attacker can only get the output of the model. Papernot, McDaniel, and Goodfellow (2016) first show the transferability of adversarial examples, *i.e.*, adversarial examples that successfully fool this model can fool another too. Square Attack (Andriushchenko, Croce, Flammarion, & Hein, 2020) is a score-based black-box attack for norm bounded perturbations that uses random search and does not exploit any gradient approximation.

2.2. Adversarial defenses

In view of the harm of adversarial examples to real-world applications, many approaches have been proposed to mitigate this problem and make the model more robust to adversarial attacks. At present, most defense methods can be divided into two categories:

1. **Adversarial Training** feeds model with clean samples and adversarial examples, which can be formulated as an optimization problem, as shown in formula (2). Adversarial training approximates the inner maximization problem by generating adversarial examples, and then update the model parameters θ.

$$\min_{\theta} E_{(\mathbf{x},y)\sim\mathcal{X},\mathcal{Y}} \max_{\|\mathbf{x}'-x\|_\infty<\epsilon} \left(\mathcal{L}\left(f_\theta\left(\mathbf{x}'\right), y\right)\right) \tag{2}$$

2. **Certified Robustness** proposes a new regularization scheme which provably improves adversarial robustness. Certified robustness studies the characteristics of the model, and proposes a regular term to improve robustness.

In this paper, we mainly study adversarial training. Among those defense methods about adversarial training, FGSM is the fastest way for approximating the inner maximization of formulation (2), which performs only one gradient ascent by using first-order approximation of loss function:

$$\mathbf{x}' = \mathbf{x} + \epsilon \operatorname{sign}\left(\nabla_{\mathbf{x}} \mathcal{L}\left(f_\theta(\mathbf{x}), y\right)\right) \tag{3}$$

However, FGSM adversarial training may fall into the dilemma of gradient masking (Athalye, Carlini, & Wagner, 2018; Tramèr, Kurakin, Papernot, Goodfellow, Boneh, & McDaniel, 2018) or catastrophic overfitting (Wong et al., 2020). A more general method is PGD adversarial training (Madry et al., 2018), which is one of the most effective methods to improve model's adversarial robustness. The iterative formula of the PGD attack method is as follows:

$$\mathbf{x}'_{t+1} = \Pi_{\|\mathbf{x}'-\mathbf{x}\|_\infty \leq \epsilon}\left(\mathbf{x}'_t + \alpha \operatorname{sign}\left(\nabla_{\mathbf{x}'_t} \mathcal{L}\left(f_\theta\left(\mathbf{x}'_t\right), y\right)\right)\right) \tag{4}$$

Zhang et al. (2019) provided a differentiable upper bound for prediction error of adversarial examples, and proposed a defense method to trade adversarial robustness off against accuracy (TRADES). Manifold adversarial training (Zhang, Huang, Zhu, & Liu, 2021) built an adversarial framework to promote the manifold smoothness in the latent space, which can learn a more robust and compact data representation. Feature Scattering (FS) (Zhang & Wang, 2019) considered the relationship of inter-samples, and trained the model through feature scattering in the latent space. Zhang and Qian et al. (2021) investigated adversarial training from the perspective of shift consistency in latent space, and proposed a new regularization method — shift consistency regularization (SCR), which can achieve impressive adversarial robustness when combined with FS.

Although the above adversarial training methods can greatly improve the adversarial robustness of the model, these performance improvements come at the cost of time as it relies on the multi-step adversarial attack. In order to overcome this issue, some studies attempted to eliminate the overhead cost of generating adversarial examples in PGD adversarial training. 'Free' adversarial training (Shafahi, Najibi, Ghiasi, Xu, Dickerson, Studer, Davis, Taylor, & Goldstein, 2019) recycled the gradient information when updating model parameters, which can achieve comparable robustness to PGD adversarial training and can be 7 or 30 times faster than other PGD adversarial training methods. Some studies (Andriushchenko & Flammarion, 2020; Kim et al., 2021; Li et al., 2020; Schwinn, Raab, & Eskofier, 2020; Vivek & Babu, 2020; Wong et al., 2020) attempted to use single-step attack instead of multi-step attack to speed up the training process. Wong et al. (2020) observed the phenomenon of catastrophic overfitting and proposed Fast Adversarial Training to solve this problem, which adds the random initialization process in FGSM attack. However, it can only get a suboptimal result due to its cyclic learning rate setting. Li et al. (2020) proposed combining FGSM adversarial training with PGD adversarial training. Although it can achieve comparable robustness and without too many epochs, PGD adversarial training is still time-consuming. We hope that we can only use FGSM in adversarial training to reduce training time. Andriushchenko and Flammarion (2020) showed that Fast Adversarial Training cannot prevent catastrophic overfitting, thus they proposed a new regularization method named GradAlign to prevent this phenomenon. They maximized the gradient alignment inside the perturbation set and therefore improved the quality of the FGSM attack. However, this method greatly increased the time complexity, which violates the original intent of FGSM adversarial training.

2.3. Input-adaptive inference

Methods for improving DNN's efficiency can be divided into two types: reducing model calculation (Sandler, Howard, Zhu, Zhmoginov, & Chen, 2018) and input-adaptive inference (Hong, Kaya, Modoranu, & Dumitras, 2021; Huang, Chen, Li, Wu, van der Maaten, & Weinberger, 2018; Kaya, Hong, & Dumitras, 2019; Teerapittayanon, McDanel, & Kung, 2016). Reducing model calculation means designing a compact network or compressing the model. Input-adaptive inference means that samples can adaptively choose different exits during inference, and simple samples can be output in the front part of the model, which reduces unnecessary computation. Here we mainly talk about input-adaptive inference in adversarial robustness.

BranchyNet (Teerapittayanon et al., 2016) is the first architecture that is augmented with additional side branch classifiers, which exploits the observation that features learned at an early layer of a network may often be sufficient for the classification of many data points. MSDNet (Huang et al., 2018) extended the multi-exit network by using a two-dimensional multi-scale network architecture, which can maintain coarse and fine level features throughout the network. So far, there is not much research on adversarial examples for the multi-exit network. Hong et al. (2021) proposed DeepSloth, which can cause multi-exit DNN slowdown. Hu, Chen, Wang, and Wang (2020) studied adversarial learning in the multi-exit network, and they proposed three attack methods and a defense method for the multi-exit network. However, they did not explain why the multi-exit network can improve the robustness, and the whole training process uses PGD adversarial training, which takes a long time. In this paper, we will explain why the multi-exit network can improve adversarial robustness and speed up the training by using FGSM adversarial training.

3. Multi-exit network

In this section, we first introduce the multi-exit network structure used in this paper. Then we discuss attack and defense methods in the multi-exit network.

3.1. MSDNet

Huang et al. (2018) developed a DNN that can slice a large network into small parts and process these slices sequentially, and stop the inference process once the prediction is sufficiently confident. However, the existing DNN has two problems: (1) The multi-exit network should extract different features according to the number of layers left before classification. In contrast, the existing DNN directly extracts the features of the last layer. (2) The first layer of the DNN operates on the fine scale to extract low-level features, and the subsequent layers are transferred to the coarse scale. Thus the global context can enter the classifier. Both of these scales are required, but they occur in different locations in the network. These two problems make it infeasible to add exits to traditional CNN directly.

Therefore, Huang et al. (2018) proposed a novel network architecture. For the first problem, they solved it by density connectively (Huang, Liu, van der Maaten, & Weinberger, 2017). By connecting all layers to all classifiers, the classifier is no longer dominated by the nearest feature. For the second problem, because the front layer lacks coarse-grained features, a multi-scale structure is proposed. In each layer, all scale features (from fine to coarse) are generated, which are helpful to the classification of early classifiers. The network structure is illustrated in Fig. 1, and they refer to it as Multi-Scale DenseNet (MSDNet).

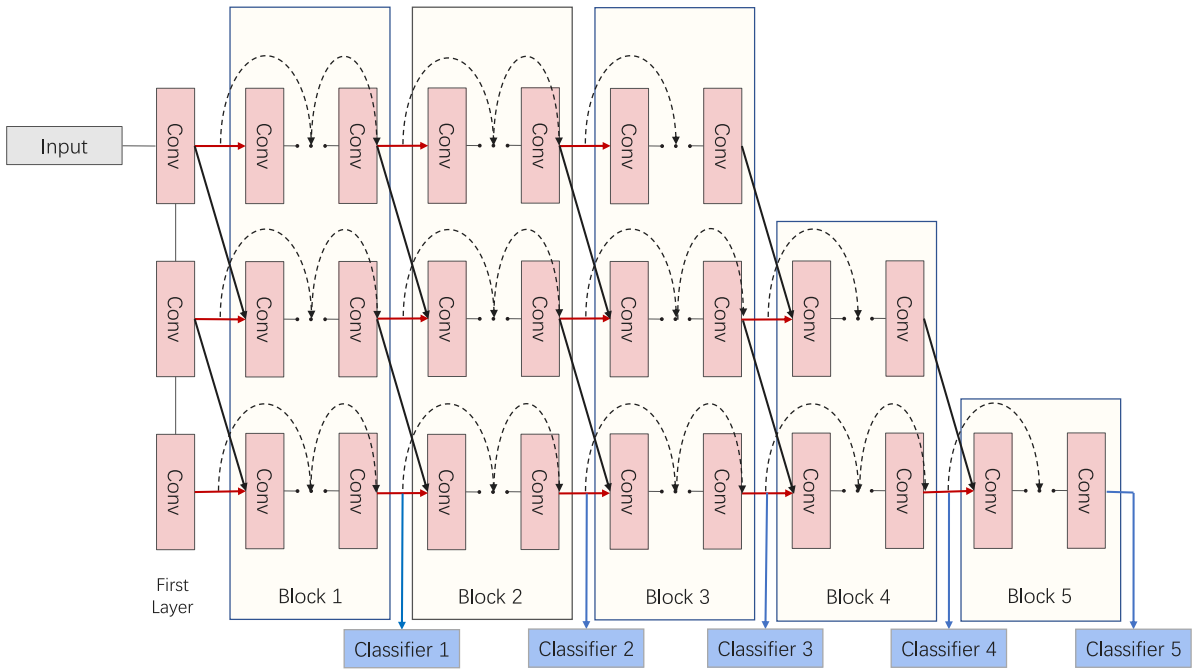


Fig. 1. An overview of MSDNet with network reduction we used to train CIFAR-10. This is a multi-scale network which has three scale, and the network is divided into five blocks, which maintain a decreasing number of scales. The classifier consists of two convolution layers, an average pooling layer and a fully connected layer. The dotted line and solid line between different scales indicates dense connectivity.

3.2. Attack and defense

Considering samples $\{(\mathbf{x}, y)\} \in (\mathcal{X}, \mathcal{Y})$, an N-output network with N classifiers can produce a set of predictions $[\hat{y_1}, \hat{y_2}, \ldots, \hat{y_N}]$. We denote these classifiers as $[f_{\theta_1}, f_{\theta_2}, \ldots, f_{\theta_N}]$, where θ_i denotes the model parameter of f_{θ_i}, $i = 1, \ldots, N$, and as shown in Fig. 1, f_{θ_i} will share some weights. In white-box attack, $\mathcal{L}$ is the loss function, δ is the perturbation we generate, ϵ is the bound of perturbation and $\mathbf{x} = \mathbf{x} + \delta$ denotes the adversarial example.

Hu et al. (2020) proposed three attack optimization formulations and one adversarial training scheme for the multi-exit network, here we briefly review.

Single Attack is a naive extension for attack single-exit network, which is defined to maximize loss function of one of the classifiers.

$$\mathbf{x}'_i = \arg\max_{\delta \in \|\delta\|_p \leq \epsilon} \left| \mathcal{L}\left(f_{\theta_i}\left(\mathbf{x} + \delta\right), y\right) \right| \tag{5}$$

where y is the ground truth label of x, x' is adversarial example, δ is perturbation and ϵ is limitation of perturbation in l_p norm.

Average Attack maximizes the average of losses of all classifiers, so that x' can better attack all exits.

$$\mathbf{x}'_{avg} = \arg\max_{\delta \in \|\delta\|_p \leq \epsilon} \left| \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}\left(f_{\theta_i}\left(\mathbf{x} + \delta\right), y\right) \right| \tag{6}$$

where N is the number of exits.

Max-Average Attack is a combination of single attacks and average attack, which aims to strengthen the ability of a single attack. This method does not simply maximize the average of all losses, but first solves N times of single attacks to get $\mathbf{x}'_i$ through formulation (5), and denotes their collection as Ω, then calculates the average loss as formulation (6) do respectively and finds i^* that can maximize the loss function.

$$\mathbf{x}'_{max} \leftarrow \mathbf{x}'_{i^*}, \text{ where } \mathbf{x}'_{i^*} \in \Omega$$
$$\text{and } i^* = \arg\max_i \left| \frac{1}{N} \sum_{j=1}^{N} \mathcal{L}\left(f_{\theta_j}\left(\mathbf{x}'_i\right), y\right) \right| \tag{7}$$

Hu et al. (2020) also proposed a defense scheme based on min–max optimization of adversarial training (Madry et al., 2018), which can embed the above three attack methods for generating adversarial examples. The formulation is as follows:

$$\theta_i \in \Theta, \text{ where } \theta_i = \arg\min_{\theta'} \left| \mathcal{L}\left(f_{\theta_i}(\mathbf{x}), y\right) + \mathcal{L}\left(f_{\theta_i}\left(\mathbf{x}'\right), y\right) \right| \tag{8}$$

where Θ is the union of learnable parameters and $\theta_1 \cup \theta_2 \cup \cdots \cup \theta_N = \Theta$. And $\mathbf{x}' \in \{\mathbf{x}'_i, \mathbf{x}'_{avg}, \mathbf{x}'_{max}\}$.

4. Approach

In this section, we first investigate the relationship between the model's depth and adversarial robustness in Section 4.1. Then we show that the multi-exit network can alleviate catastrophic overfitting of FGSM adversarial training and empirically discover the underlying reason in Section 4.2. Our method is proposed in Section 4.3.

4.1. Motivation

In order to investigate the relationship between the model's depth and adversarial robustness, we investigate the characteristics of the model's output. Considering a multi-exit network with N classifiers, a set of samples $\{(\mathbf{x}, y)\}$, we produce adversarial examples $\mathbf{x}'$ for each $\mathbf{x}$. And then, we calculate the features of both clean samples and adversarial examples before softmax respectively, and denote those features as l_i and $l_{i'}$, $i = 1, 2, \ldots, N$. To compare clean samples with adversarial examples intuitively, the differences between them are calculated and expressed in l_2 norm, denotes as d_i, where $d_i = \|l_i - l_{i'}\|_2$.

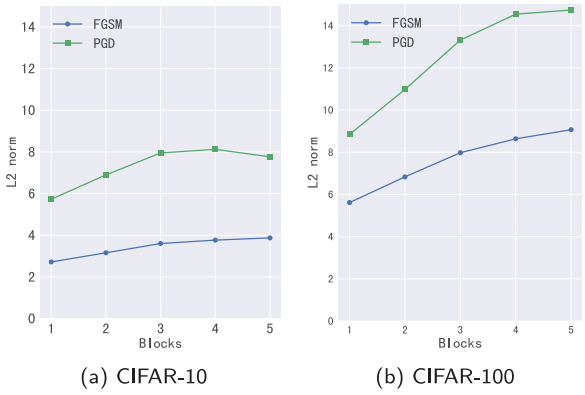


Fig. 2. d_i of FGSM and PGD attack in a 5-output MSDNet in CIFAR-10 and CIFAR-100. The x-axis represents the index of the block or classifier. d_i shows a monotonous increasing trend, which means the influence of adversarial perturbations on the model is magnified as the network becomes deeper.

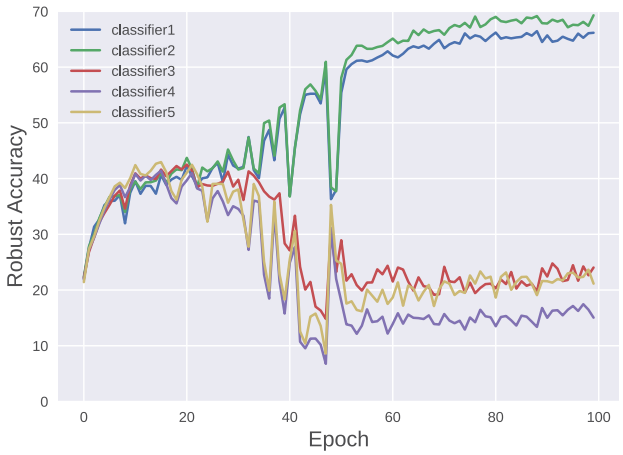


Fig. 3. The accuracies of catastrophic overfitting in the multi-exit network on CIFAR-10 dataset. For robust accuracy here, we use PGD-5 attack.

We regard d_i as the impact of adversarial perturbations on deep neural networks. When d_i is larger, that is, the model has a larger difference for a pair of samples that are visually similar, it is considered that the impact of adversarial perturbations on the model is more severe.

As we can see in Fig. 2, d_i shows a monotonous increasing trend. That means the difference between clean samples and adversarial examples of the model becomes larger as the network becomes deeper. Therefore, the features obtained by recognizing adversarial examples are increasingly different from those of clean samples. The model prediction is more likely to be wrong. Apart from this, differences in CIFAR-100 (Fig. 2b) are greater than that in CIFAR-10 (Fig. 2a), which is in line with our common sense. The classification task in CIFAR-100 is more complicated and therefore easier to attack. Those observations further support us to use d_i to estimate the adversarial robustness.

It can also be observed from Fig. 2 that the growth rate of PGD is much faster than FGSM, which shows that the stronger the attack, the greater the deviation of model recognition. Thus, the influence of perturbation on the model will be more severe as the network goes deeper, making the model more vulnerable to adversarial examples. For some simple samples, if the classification results can be output when the effect of the adversarial perturbation is small, is it possible to improve adversarial robustness without reducing too much accuracy?

Based on the above observation, we propose to use the multi-exit network to improve the model's adversarial robustness. We will output easily identified samples in early exits of the network to better avoid the influence of perturbations on the samples and improve model efficiency.

4.2. Catastrophic overfitting

Although the multi-exit network can improve the adversarial robustness of the model to some extent, in order to have a better robustness of the model, we will further incorporate the FGSM adversarial training, which is mentioned in Section 3.2.

Fast Adversarial Training (Wong et al., 2020) uses random initialization to increase the diversity of FGSM adversarial examples and improve catastrophic overfitting. However, we find that when we train too many epochs, Fast Adversarial Training will also fall into the dilemma of catastrophic fitting, as shown in Fig. 3. Catastrophic fitting is a phenomenon that occurs in FGSM adversarial training (Wong et al., 2020). After training for a while, the recognition accuracy of PGD attack suddenly drops to 0. It can be seen from Fig. 3 that although the multi-exit network can alleviate catastrophic overfitting, the robustness of the late exits' classifiers still eventually decreases. The classifiers of early exits will not fall into overfitting, which is in line with the conclusions of Wong et al. (2020) and the discussion of Section 4.1, that is, large epsilon for FGSM adversarial training may force the model to overfit to the boundary of the perturbation region. For the classifiers of early exits, the impact of adversarial perturbations on it is relatively small. Thus, minor epsilon training is used. While for the late classifiers, the impact of adversarial perturbations is magnified so that it may fall into overfitting.

It can be observed from Fig. 3 that robust accuracy begins to decline around the twenty-ninth epoch. Although different from the single-exit network, the robust accuracy is not reduced to 0, accuracies of late exits still drop a lot. To understand the difference between FGSM adversarial training and PGD adversarial training, inspired by the relationship between overfitting and l_2 regularization, we investigate the l_2 norm of weights at a fully connected layer (Here we abbreviate it as LW). Moreover, we plot in Fig. 4 that LW obtained by FGSM and PGD adversarial training on CIFAR-10. Catastrophic overfitting occurs for FGSM adversarial training around the twenty-ninth epoch. It has the following characteristics: (a) At the twenty-ninth epoch, robust accuracies of late exits of the model begin to drop, along with robust accuracies of early exits speed up the increase. In addition, before the twenty-ninth epoch, LW of late exits never exceeded that of classifiers of early exits. This observation suggests that the model may start to overfit the FGSM attack. (b) LW of FGSM adversarial training is bigger than PGD adversarial training, and LW gradually stabilizes and will not recover after catastrophic overfitting. The connection between LW and catastrophic overfitting has aroused our interest. We hope to solve the catastrophic overfitting problem through LW.

4.3. Proposed method

Based on the link between FGSM adversarial training and LW, we propose a regularizer to fix the problem. Section 4.2 mentioned that LW of late exits exceed LW of early exits in the model and LW of FGSM adversarial training is bigger than that of PGD adversarial training. Thus, our regularizer tries to minimize the LW of all classifiers, and for the LW of late exits, we increase the penalty term to make sure that these LW will not exceed the LW of early exits. The optimization objective is as follows:

$$\sum_{n=1}^{N} \left(\mathcal{L}\left(f_{\theta_i}(\mathbf{x}), y\right) + \mathcal{L}\left(f_{\theta_i}\left(\mathbf{x}'\right), y\right) + \lambda_i \times LW_i \right) \tag{9}$$

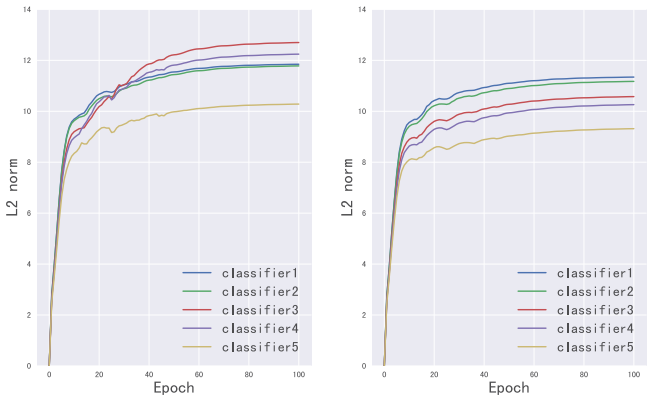


Fig. 4. l_2 norm of the weights of fully connected layers (LW) obtained by FGSM adversarial training and PGD adversarial training. Here we use PGD-7 for PGD adversarial training.

where N is the number of classifiers in multi-exit network, θ_i is the ith classifier, $\mathbf{x}$ is the clean sample and y is its label, $\mathbf{x}'$ is adversarial example generating from $\mathbf{x}$. $\mathcal{L}$ is the loss function, LW_i is the abbreviation of l_2 norm of fully connected layer weight of ith classifier and we use λ_i to control the influence of LW_i. The implementation of our proposed training scheme is summarized in Algorithm 1.

In the following, we will use a simple neural network to mathematically prove why a model with a smaller LW can be more robust against adversarial examples. Without loss of generality, we use a fully connected network to illustrate.

Consider a single network $f(\mathbf{X}) = \mathbf{XW}$, where $\mathbf{W}$ is a $n \times k$ matrix, k is the number of classification tasks. $\mathbf{X}$ is a sample with n features, whose dimension is $1 \times n$. The process of minimizing LW is the process of minimizing l_2 norm of $\mathbf{W}$. Since the process of minimizing l_2 produces more parameters close to 0, when the mean of $\mathbf{W}$ is 0, this process will make the variance of $\mathbf{W}$ smaller. Therefore, we consider the following two models:

1. $f_1(\mathbf{X}) = \mathbf{XW_1}$, where $w_{i,j}^{(1)} \in \mathbf{W}_1$, $w_{i,j}^{(1)} \sim U(0, d_1)$ and $w_{i,j}^{(1)}$ are independent of each other.
2. $f_2(\mathbf{X}) = \mathbf{XW_2}$, where $w_{i,j}^{(2)} \in \mathbf{W}_2$, $w_{i,j}^{(2)} \sim U(0, d_2)$ and $d_2 < d_1$, and $w_{i,j}^{(2)}$ are independent of each other.

We can get the following theorem:

Theorem 1. *Assume that the weights of models $W_{i=1,2}$ obey normal distribution, and the variance of W_1 is bigger than W_2, then f_1 is more susceptible to adversarial perturbations:*

$$\|f_1(\mathbf{X}+\boldsymbol{\delta}) - f_1(\mathbf{X})\|_2 > \|f_2(\mathbf{X}+\boldsymbol{\delta}) - f_2(\mathbf{X})\|_2 \tag{10}$$

Proof. Firstly, since $f_i(\mathbf{X}) = \mathbf{W}_i\mathbf{X}$, we can rewrite the two sides of the inequality as follows:

$$\begin{aligned}\|f_i(\mathbf{X}+\boldsymbol{\delta}) - f_i(\mathbf{X})\|_2 &= \|\mathbf{XW}_i + \boldsymbol{\delta}\mathbf{W}_i - \mathbf{XW}_i\|_2 \\ &= \|\boldsymbol{\delta}\mathbf{W}_i\|_2 \\ &= \sqrt{\sum_{j=1}^{k} a_j^2}\end{aligned} \tag{11}$$

where $[a_j] = \boldsymbol{\delta}\mathbf{W}_i$. And we can write the quadratic sum of a_j in matrix form, that is:

$$\begin{aligned}\sum_{j=1}^{k} a_j^2 &= (\boldsymbol{\delta}\mathbf{W}_i)(\boldsymbol{\delta}\mathbf{W}_i)^T \\ &= \boldsymbol{\delta}\mathbf{W}_i\mathbf{W}_i^T\boldsymbol{\delta}^T\end{aligned} \tag{12}$$

Thus we can substitute formula (11) and formula (12) into formula (10) to obtain the following form:

$$\mathbb{E}\left[\boldsymbol{\delta}\mathbf{W}_1\mathbf{W}_1^T\boldsymbol{\delta}^T\right] > \mathbb{E}\left[\boldsymbol{\delta}\mathbf{W}_2\mathbf{W}_2^T\boldsymbol{\delta}^T\right] \tag{13}$$

Since δ is not a random variable, we only need to prove:

$$\boldsymbol{\delta}\mathbb{E}\left[\mathbf{W}_1\mathbf{W}_1^T - \mathbf{W}_2\mathbf{W}_2^T\right]\boldsymbol{\delta}^T > 0 \tag{14}$$

$$\mathbb{E}\left[\mathbf{W}_1\mathbf{W}_1^T - \mathbf{W}_2\mathbf{W}_2^T\right] > 0 \tag{15}$$

We partition the matrix W_i into the following form:

$$\mathbf{W}_i = [\boldsymbol{\omega}_1^{(i)}, \boldsymbol{\omega}_2^{(i)}, \ldots, \boldsymbol{\omega}_k^{(i)}] \text{ and } \boldsymbol{\omega}_j^{(i)} = [w_{j1}^{(i)}, w_{j2}^{(i)}, \ldots, w_{jn}^{(i)}]^T \tag{16}$$

where $j = 1, \ldots, k$, k is the number of classification tasks, n is the number of features.

Through formula (16), we can get: $\mathbf{W}_i\mathbf{W}_i^T = \sum_{j=1}^{k} \boldsymbol{\omega}_j^{(i)} \times \boldsymbol{\omega}_j^{(i)^T}$,where

$$\begin{aligned}\boldsymbol{\omega}_j^{(i)} \times \boldsymbol{\omega}_j^{(i)^T} &= \begin{pmatrix} w_{j1}^{(i)} \\ w_{j2}^{(i)} \\ \vdots \\ w_{jn}^{(i)} \end{pmatrix} \times \left(w_{j1}^{(i)}, w_{j2}^{(i)}, \ldots, w_{jn}^{(i)}\right) \\ &= \begin{pmatrix} w_{j1}^{(i)} \times w_{j1}^{(i)} & \cdots & w_{j1}^{(i)} \times w_{jn}^{(i)} \\ \vdots & \ddots & \vdots \\ w_{jn}^{(i)} \times w_{j1}^{(i)} & \cdots & w_{jn}^{(i)} \times w_{jn}^{(i)} \end{pmatrix}\end{aligned} \tag{17}$$

Accroding to the formulation $V(x) = \mathbb{E}(x^2) - [\mathbb{E}(x)]^2$, since $\mathbb{E}(w_{ja}) = 0$, w_{ja} and w_{ja} are independent, then for all $w_{ja}^{(i)} \times w_{jb}^{(i)}$:

$$\mathbb{E}\left[w_{ja}^{(i)} \times w_{jb}^{(i)}\right] = \begin{cases} V(w_{ja}^{(i)}) + [\mathbb{E}(w_{ja}^{(i)})]^2 = d_i, & \text{if } a = b \\ 0, & \text{else} \end{cases} \tag{18}$$

Substituting formula (18) into formula (17), we can get:

$$\mathbf{W}_i\mathbf{W}_i^T = \begin{pmatrix} k \times d_i & & & \\ & k \times d_i & & \\ & & \ddots & \\ & & & k \times d_i \end{pmatrix} \tag{19}$$

where n is the number of features and d_i is variance of w_i.

According to condition $d_2 < d_1$, formula (10) is proved.

Theorem 1 shows that if the l_2 weight norms of the model are smaller, its adversarial robustness will be stronger. Especially, our method plays the same role as weight decay (Krogh & Hertz, 1992). However, the difference from weight decay is: (1) Our method only minimizes the weight of the fully connected layer. If we just set parameters of l_2 regularization to be large, *i.e.*, minimize parameters of the whole model, the model may be underfitting. (2) We solve the dilemma of catastrophic overfitting of multi-exit network FGSM adversarial training, *i.e.*, the parameter λ_i of LW_i in formulation (9) can be different for different classifiers. According to previous observations, we set the λ_i of late exits slightly bigger than the λ_i of early exits to avoid catastrophic overfitting.

Algorithm 1 Our training scheme for N-output network f_θ

Input: epsilon ϵ, epochs T, step size α, number of classifiers N, $\lambda = [\lambda_1, \lambda_2, ..., \lambda_N]$ and a dataset of size M
Output: θ

for $t = 1, 2, ..., T$ **do**
  // Traverse the dataset and perform FGSM attack
  for $i = 1, 2, ..., M$ **do**
    δ = Uniform(-ϵ,ϵ)
    $\delta = \delta + \alpha \times \text{sign}(\nabla_\delta \mathcal{L}\left(f_\theta\left(x_i + \delta\right), y_i\right))$
    $\delta = \max(\min(\delta, \epsilon), -\epsilon)$
    // Calculate loss of clean and adversarial examples
    $Loss = \mathcal{L}\left(f_\theta(x_i), y_i\right) + \mathcal{L}\left(f_\theta(x_i + \delta), y_i\right)$
    // Add regular term to Loss
    for $j = 1, 2, ..., N$ **do**
      $Loss = Loss + \lambda_j \times LW_j$
    // Update model parameters with given optimizers
    $\theta = \theta - \nabla_\theta Loss$
return θ

5. Experimental results

This section shows the performance of models trained using our proposed training scheme against various adversarial attacks on CIFAR-10, CIFAR-100 and SVHN. In addition to multi-step gradient-based attack methods such as BIM, MIM, and PGD, we also used C&W attack to ensure the model does not exhibit obfuscated gradients (Athalye et al., 2018).

5.1. Experiment setup

Datasets: We evaluate adversarial robustness on CIFAR-10, CIFAR-100 (Krizhevsky, Hinton, et al., 2009) and SVHN. Following the literature (Huang et al., 2018), we deploy MSDNet (block = 5) on CIFAR-10 and MSDNet (block = 7) on CIFAR-100.[1] For data-augmentation, random crop and random horizontal flip are performed on CIFAR-10 and CIFAR-100.

Benchmark models: We compare the proposed method with 7-step PGD Adversarial Training (PGD-7), 10-step PGD Adversarial Training (PGD-10), 7-step PGD Adversarial Training with Shift Consistent Regularization (PGD-SCR), Free PGD Adversarial Training (Free-8), Fast FGSM and Grad Alignment (GradAlign), all those methods are described in Section 2.2.

Hyper-parameters: For CIFAR-10 and SVHN, we use SGD optimizer with momentum of 0.9 and train for 100 epoch, where weight decay is set to be 0.0005. The learning rate is set as 0.05, and is divided by 10 at the 50th and the 75th epoch. The λ in formulation (9) is set as [0.1, 0.1, 0.15, 0.15, 0.15]. Epoch in Fast-FGSM and Free-8 is set to 60 and 25 respectively.

For CIFAR-100, we also use the SGD optimizer but train for 150 epochs. The learning rate is set as 0.1 and is divided by 10 at the 75th and the 115th epoch. λ = [0.1, 0.1, 0.1, 0.15, 0.15, 0.15, 0.15]. For Fast FGSM, we follow the settings in Wong et al. (2020), which train for 60 epochs with cyclic learning rate. Furthermore, for Free PGD, we set the number of training epochs as 50, mini-batch replay as 8, which are the same as literature (Shafahi et al., 2019).

Attack and defense: We use four attack methods to evaluate the adversarial robustness, including BIM, MIM, PGD, C&W and AA mentioned in Section 2.1. The bound of perturbations ϵ is set to be 8/255. For BIM, MIM and PGD, we set the step size of attack as 1/255 and the number of iterations as 40 while the iteration times in the C&W attack is 100. And all attack methods are untargeted attacks. In defense, we use FGSM adversarial training. All attack is in the mode of Average Attack describe in Section 3.2, because we think it is the most balanced form of attack, it can better evaluate/improve the adversarial robustness. In order to be easy to reproduce, all attack and defense methods are based on Ding, Wang, and Jin (2019).[2]

Evaluation metrics: We evaluate accuracy, robustness, training time, using the metrics below:

- Standard Accuracy (Standard): classification accuracy on original samples.
- Robust Accuracy: classification accuracy on adversarial test set against a specific attacker.
- Training Time (Time): the total execution time for training the model in minutes.

5.2. CIFAR-10

The experimental settings are described in Section 5.1. And the experimental results are shown in Table 1.

Table 1 demonstrates that the accuracy of our method on clean samples is comparable to other methods. Nevertheless, the most robust method among the comparison methods, TRADES, has relatively lower accuracy on clean samples. This result confirms the point in the paper (Tsipras, Santurkar, Engstrom, Turner, & Madry, 2019) that using more powerful attacks for adversarial training will reduce the accuracy of the model. Moreover, Free-8 uses a method similar to PGD, but the number of training epochs of this method is much smaller than those of other methods, which has a more significant impact on its accuracy. Despite saving a lot of training time, the results obtained in this way are not ideal. In comparison, the proposed method does not cause a significant loss of accuracy.

In terms of robust accuracy, our method is better than other training methods, no matter single-step adversarial training or multi-step adversarial training. TRADES is an improved version of PGD and can get the best robustness currently. Our method has similar robust accuracy to TRADES, and the training time is significantly reduced, which is reduced by 50% compared to TRADES. Compared to other single-step adversarial training methods, our method is even better. Especially it is well known that PGD and C&W are the two most powerful attack methods at present, and the recognition accuracy of our method under these two attacks is much higher than those of other methods.

Talking about the training time, although compared to the proposed method, Free-8 and Fast FGSM take a shorter time to train due to their fewer epochs, but they finally get a sub-optimal result. It means that we have made a trade-off between training time and robustness. Compared with Fast-FGSM and other methods, we can get better adversarial robustness because we have solved the catastrophic overfitting problem during the training of multiple epochs. Compared with multi-step adversarial training such as TRADES, we can reduce a lot of training time and get a robust model. It is interesting to note that the training time of PGD-10 here is nearly doubled that of TRADES. This may be due to the fact that the cross-entropy is calculated twice in one iteration (clean sample and adversarial sample). In comparison, TRADES only needs to calculate a KL divergence in one attack iteration, which caused a difference in training time. Due to the FGSM attack, our method can save half of the time than TRADES.

[1] We use the public implementation of MSDNet available at https://github.com/kalviny/MSDNet-PyTorch.

[2] Our attack and defense methods are based on AdverTorch, which is available at https://github.com/BorealisAI/advertorch.

Table 1
CIFAR-10 results.

Training method	Standard	PGD	BIM	MIM	CW-100	AA	Time (min)
PGD-7	82.26%	40.75%	40.29%	41.98%	67.04%	28.90%	755.28
PGD-10	81.97%	41.56%	40.65%	42.89%	66.49%	29.83%	881.39
PGD-SCR	82.27%	43.99%	43.79%	45.23%	62.47%	25.38%	–
TRADES	78.11%	48.90%	48.83%	**49.48%**	54.55%	35.31%	406.90
Free-8	73.46%	41.30%	41.31%	42.55%	53.21%	29.48%	73.32
Fast FGSM	79.99%	42.85%	42.62%	42.23%	61.23%	32.92%	64.33
GradAlign	**82.78%**	44.46%	44.19%	42.67%	69.06%	37.26%	364.48
Proposed	81.93%	**49.30%**	**48.97%**	47.00%	**74.34%**	**43.90%**	201.50

Table 2
CIFAR-100 results.

Training method	Standard	PGD	BIM	MIM	CW-100	AA	Time (min)
PGD-7	61.60%	22.08%	21.94%	22.58%	47.02%	19.50%	816.05
PGD-10	61.97%	22.79%	22.78%	23.14%	46.97%	19.95%	941.17
TRADES	55.74%	**27.76%**	**27.65%**	**28.16%**	36.72%	22.55%	658.33
Free-8	51.05%	23.96%	23.87%	24.47%	23.04%	19.13%	176.62
Fast FGSM	52.86%	23.85%	23.67%	23.92%	38.08%	19.14%	88.60
GradAlign	62.60%	25.00%	24.47%	23.40%	49.24%	**23.59%**	641.33
Proposed	**63.01**%	24.05%	23.97%	23.55%	**51.56%**	21.83%	291.47

The proposed approach gets better adversarial robustness than other single-step adversarial training methods by mitigating the catastrophic overfitting problem.

5.3. CIFAR-100

The results on CIFAR-100 reveal a drawback of adversarial training, i.e., it may significantly reduce the recognition accuracy on clean samples, especially for models trained with more powerful attacks, e.g., TRADES. It can be seen from Table 2 that under attack methods such as PGD, TRADES has the strongest adversarial robustness. However, its accuracy is also the lowest except for Fast FGSM and Free-8 methods and is far lower than the proposed method. Although such a model can obtain strong robustness, the method has a significant loss in the recognition accuracy on clean samples. Furthermore, TRADES cannot resist C&W attacks very well. In contrast, the proposed method obtains a much higher recognition accuracy than TRADES under the C&W attack, reflecting the balance of our approach.

Note that GradAlign can also effectively improve the model's adversarial robustness. However, the time we spend is half of GradAlign. Although GradAlign is a single-step adversarial training method, the time complexity for calculating the regularization term is high, where the real time cost of this method is close to that of PGD adversarial training. In other words, the improvement of robustness in GradAlign is obtained by sacrificing the training time, which loses the original intention of fast adversarial training. Although our method is slightly worse than GradAlign in improving the robustness on CIFAR-100, it can maintain the highest robustness under C&W attack, and it also has a great advantage in time.

About other PGD adversarial training methods, our method is superior to them in all aspects. Regardless of the accuracy on clean samples or the robust accuracy, although our method is only 1% or 2% higher than others, it only needs nearly one-third of the time of PGD-7 and PGD-10, which greatly improves training efficiency. Besides, the reason why PGD-7 and PGD-10 take much longer time than TRADES here is the same as explained in Section 5.2, i.e., the cross-entropy is calculated twice when calculating loss.

Same as the result of CIFAR-10, Fast FGSM and Free-8 took the shortest time and got a seemingly robust model. However, these models can only successfully defend against gradient-based attack methods, such as PGD. For C&W attack, the models trained by these two methods are far worse than other models. The exact cause is unknown, and it needs further analysis. Although these two methods greatly reduce the training time, only a sub-optimal model can be obtained in the end. We also think that such a model is not ideal. Our method has a better trade-off between training time and robustness. Compared with Fast-FGSM and other methods, we can obtain better adversarial robustness because we alleviate the catastrophic overfitting problem during multiple epoch training. At the same time, due to the use of single-step adversarial training, the time consumed is shorter than multi-step methods such as TRADES. Overall, our method stands out among these methods.

5.4. SVHN

As shown in Table 3, the proposed method outperforms multi-step adversarial training under all attacks. But surprisingly, Free and Fast FGSM outperform other methods on SVHN by a lot, which may be because these two methods are more suitable for simple datasets. Apart from this, Whether GradAlign under cyclic settings, or ours, it does not achieve ideal adversarial robustness on this dataset (here we use cyclic settings).

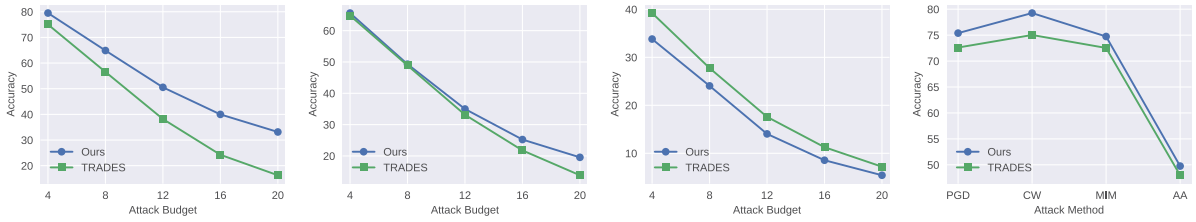
Another interesting phenomenon is that most of the single-step adversarial training methods are better than those multi-step adversarial training methods on SVHN, which possibly because these methods are more suitable for slightly simple datasets with the multi-exit network. The model can learn sufficient information from FGSM adversarial training, so it also has the ability to generalize to other attacks. In general, the proposed method can achieve well performance on these datasets.

5.5. Other evaluations

We also test the adversarial robustness of the proposed method under multiple attack budgets. The result is shown in Fig. 5, our method outperforms TRADES on both SVHN and CIFAR-10. It can be also observed that as the dataset becomes more complex, the gap of accuracies between the proposed method and TRADES becomes narrower, and finally, it is slightly worse than TRADES on CIFAR-100. Combined with the previous results,

Table 3
SVHN results.

Training method	Standard	PGD	BIM	MIM	CW-100	AA	Time (min)
PGD-7	91.57%	48.45%	48.24%	49.15%	73.85%	55.05%	943.97
PGD-10	94.09%	48.54%	48.45%	48.47%	73.97%	54.05%	1050.24
TRADES	89.56%	56.55%	56.28%	57.50%	73.08%	60.24%	538.15
Free-8	91.69%	72.06%	**71.75%**	**70.86%**	86.21%	**72.37%**	–
Fast FGSM	91.28%	**73.03%**	67.61%	63.57%	87.57%	68.71%	81.33
GradAlign	95.25%	24.02%	23.55%	26.58%	53.93%	21.85%	156.75
Proposed	92.77%	64.89%	63.82%	56.37%	**89.72%**	63.13%	253.90



(a) SVHN Attack Budget (b) CIFAR-10 Attack Budget (c) CIFAR-100 Attack Budget (d) Black-Box

Fig. 5. Figure (a–c) are comparisons of PGD attack on proposed method and TRADES on different budgets. And Figure (d) is black-box attack on CIFAR-10 (attacking with PGD, CW and MIM).

we believe that FGSM-based methods may be more suitable for relatively simple datasets.

For black-box, we use the transfer attack proposed by Papernot et al. (2016). We adopt the model provided by the research (Zhang & Wang, 2019) as the attacked model, and test it with PGD, MIM, C&W and AA attacks. Fig. 5d shows the robust accuracy of proposed method and TRADES under black-box attack. The results show that the proposed method can better defend against transferable adversarial examples than TRADES.

6. Discussion

Catastrophic overfitting is a phenomenon in which accuracy against PGD attacks drops significantly during FGSM adversarial training. Wong et al. (2020) first observed the problem, and believed that the reason for the problem might be insufficient adversarial examples generated by the FGSM attack. Thus, the combination of random initialization in the FGSM attack is proposed to alleviate catastrophic overfitting. Other researchers also solved this problem from the perspective of sample diversity. In this paper, we tackle this problem from another angle, that is, from the model's perspective.

Firstly, we propose to use the multi-exit networks to improve the model's adversarial robustness. Moreover, by observing the overfitting and non-overfitting classifiers in the multi-exit network as well as comparing the FGSM training model and PGD training model, we find that weights of the fully connected layer of the model may be the cause of catastrophic overfitting. As shown in Fig. 3, classifiers of early exits will not fall into overfitting while classifiers of late exits lose the ability to generalize multi-step attacks. At the same time, as shown in Fig. 2, when catastrophic overfitting occurs, the l_2 norms of the model classifiers also change drastically. Therefore, the weight norms of the fully connected layer are adjusted to alleviate the catastrophic overfitting phenomenon, thereby improving adversarial robustness.

l_2 regularization is a commonly used method to alleviate overfitting. In this method, l_2 norm is used as the regularization term in order to penalize large weight values (Goodfellow, Bengio, & Courville, 2016). Here we have further discussed the impact of l_2 regularization on adversarial perturbations. Through multiple experimental observations, we find that weights of the fully connected layer of the model may be the cause of catastrophic overfitting. Besides, it mathematically proves that the influence of l_2 normalization on adversarial perturbations in Theorem 1—the smaller the weight norm is, the smaller the influence of adversarial perturbations on samples, which leads to higher adversarial robustness of the model.

Theorem 1 proves that l_2 regularization can weaken the influence of adversarial perturbations. It assumes that the model weights follow a normal distribution with a mean of 0. This assumption may be too strict. However, according to the literature (Goodfellow et al., 2016), the regularization takes effect when the model parameters are close to any point. Therefore, the limitation of the mean does not affect the conclusion. Here we do not know whether the correct mean should be positive or negative. And zero is a meaningful default value. Furthermore, the regularization of model parameters to zero is more common. Thus, here we only discuss this particular case. Besides, the theorem also assumes that the model is a linear model. In particular, convolution and fully connected layers are linear operations so that they can be expressed in the form of matrix multiplication. Only after the activation function is added, the network is a nonlinear operation. However, our current commonly used activation function, i.e. RELU function, is a piecewise linear function. So here we simplify the network to a linear operation.

According to experimental observations and theoretical proofs, this paper proposes a method based on l_2 regularization to alleviate catastrophic overfitting. We suggest a simple method based on traditional l_2 regularization by adopting different penalty term coefficients to minimize the weights of fully connected layers. As shown in Fig. 6, the proposed method reduces l_2 norm of the weights of all classifiers and ensures that the weight of late exits will not be greater than the weight of early exits. Tables 1 and 2 verify the effectiveness of our method.

In general, the method proposed in this article is simple, effective and not time-consuming. But there are some disadvantages. First of all, the proposed method increases the number of hyperparameters, and the setting of hyperparameters is a complicated

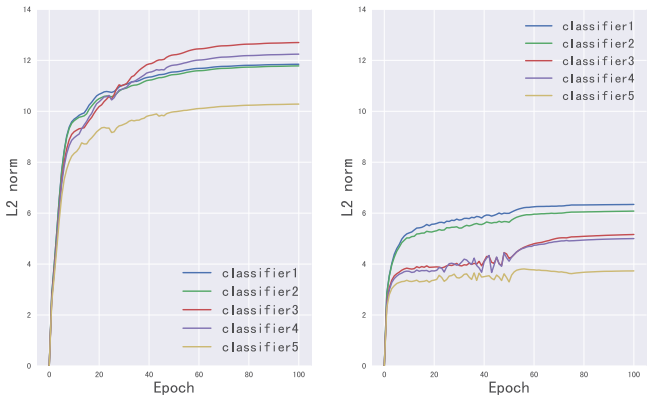


Fig. 6. l_2 norm of fully connected layer weight of model obtained by FGSM adversarial training and proposed method.

Table 4
Accuracy of models with different depth under PGD attack.

Model	FS	PGD	FS-norm
WRN-16-4	36.78%	33.70%	**41.50%**
WRN-28-10	56.62%	40.37%	**68.79%**
WRN-52-1	34.14%	**42.76%**	37.77%

problem. In our experiments, we just set the parameters simply through some experiments and guesses. Of course, there may be better options. At present, some papers propose adaptive weight decay (Nakamura & Hong, 2019), but we do not think this is the point of our article, so we do not focus on it.

Regarding the experimental dataset, we do not use datasets such as MNIST and ImageNet. Because for a small dataset like MNIST, there is no need to use a large network, and naturally, there is no need to use a multi-exit network. As for ImageNet, large-scale adversarial training is not easy and very time-consuming. The model obtained by adversarial training on ImageNet has very low accuracy on clean samples. Previous efforts to conduct adversarial training on the ImageNet dataset were unsuccessful (Kurakin et al., 2017). We believe this is a problem that needs to be solved in the future adversarial training.

In terms of defense methods, Feature Scattering method (FS) is one of SOTA methods, and we also tried to use FS for comparison, but the experimental results show that FS is not suitable for our experiment settings. This may be due to the deeper network and data normalization settings. For proving this conjecture, we compare WRN-16-4, WRN-28-10 and WRN-52-1 described in Zagoruyko and Komodakis (2016), and train the model with FS, PGD, FS with data normalization (FS-norm). The experimental results in Table 4 show that when the model is shallow, FS outperforms than PGD adversarial training. And when model become deeper (WRN-52-1), the performance of FS becomes poor. These phenomena show the effect of network model depth on FS. Besides, no matter which model is, FS-norm has a higher robustness accuracy than FS, show the role of normalization. However, in our experimental setting, the network is deep and no data normalization was used. We consider these are the reasons why FS was less effective in our experiments. Therefore, FS (Zhang & Wang, 2019) and FS-SCR (Zhang & Qian et al., 2021) are not adopted.

Conclusion

In this paper, we explore the relationship between adversarial perturbations and model depth and give some explanations on how multi-exit networks can improve robustness. Therefore, we propose using the multi-exit network to reduce the impact of adversarial perturbations. In addition, we find that the multi-exit network can alleviate the catastrophic overfitting phenomenon. The possible cause is found through experiments — the weight of overfitted classifier in the multi-exit network has a larger l_2 norm. Based on these observations, we believe that a small weight norm can improve adversarial robustness. Consequently, an objective function is proposed to minimize the weight norms of the classifiers. Experiments have verified the effectiveness of our proposed method. And theoretically, we prove that minimizing the l_2 norm of weight can effectively reduce the impact of adversarial perturbations, which shows that our method can alleviate catastrophic overfitting.

In the future, we will explore other potential causes of catastrophic overfitting and discover more properties of adversarial examples in DNN. Moreover, the primary purpose is to improve the adversarial robustness of the model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Andriushchenko, M., Croce, F., Flammarion, N., & Hein, M. (2020). Square attack: a query-efficient black-box adversarial attack via random search. In *Lecture Notes in Computer Science: 12368, Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII* (pp. 484–501). Springer.

Andriushchenko, M., & Flammarion, N. (2020). Understanding and improving fast adversarial training. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Athalye, A., Carlini, N., & Wagner, D. A. (2018). Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In *Proceedings of Machine Learning Research: 80, Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018* (pp. 274–283). PMLR.

Carlini, N., & Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (SP)* (pp. 39–57). IEEE.

Croce, F., Andriushchenko, M., & Hein, M. (2019). Provable robustness of relu networks via maximization of linear regions. In *the 22nd international conference on artificial intelligence and statistics* (pp. 2057–2066). PMLR.

Croce, F., & Hein, M. (2020). Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning* (pp. 2206–2216). PMLR.

Ding, G. W., Wang, L., & Jin, X. (2019). Advertorch v0.1: an adversarial robustness toolbox based on pytorch. arXiv:1902.07623.

Dong, Y., Liao, F., Pang, T., Su, H., Zhu, J., Hu, X., et al. (2018). Boosting adversarial attacks with momentum. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (pp. 9185–9193). Computer Vision Foundation / IEEE Computer Society.

Fan, Y., Wu, B., Li, T., Zhang, Y., Li, M., Li, Z., et al. (2020). Sparse adversarial attack via perturbation factorization. In *Computer vision–ECCV 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, Part XXII 16* (pp. 35–50). Springer.

Goodfellow, I. J., Bengio, Y., & Courville, A. C. (2016). *Adaptive computation and machine learning, Deep learning*. MIT Press, URL: http://www.deeplearningbook.org/.

Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

Hong, S., Kaya, Y., Modoranu, I., & Dumitras, T. (2021). A panda? no, it's a sloth: slowdown attacks on adaptive multi-exit neural network inference. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Hu, T., Chen, T., Wang, H., & Wang, Z. (2020). Triple wins: boosting accuracy, robustness and efficiency together by enabling input-adaptive inference. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Huang, G., Chen, D., Li, T., Wu, F., van der Maaten, L., & Weinberger, K. Q. (2018). Multi-scale dense networks for resource efficient image classification. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017* (pp. 2261–2269). IEEE Computer Society.

Kaya, Y., Hong, S., & Dumitras, T. (2019). Shallow-deep networks: understanding and mitigating network overthinking. In *Proceedings of Machine Learning Research*: *97*, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA* (pp. 3301–3310). PMLR.

Kim, H., Lee, W., & Lee, J. (2021). Understanding catastrophic overfitting in single-step adversarial training. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021* (pp. 8119–8127). AAAI Press.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. In *Advances in neural information processing systems* (pp. 950–957).

Kurakin, A., Goodfellow, I. J., & Bengio, S. (2017). Adversarial machine learning at scale. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Li, B., Wang, S., Jana, S., & Carin, L. (2020). Towards understanding fast adversarial training. arXiv preprint arXiv:2006.03089.

Lyu, C., Huang, K., & Liang, H.-N. (2015). A unified gradient regularization family for adversarial examples. In *2015 IEEE international conference on data mining* (pp. 301–309). IEEE.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Nakamura, K., & Hong, B. (2019). Adaptive weight decay for deep neural networks. *7*, In *IEEE Access* (pp. 118857–118865).

Papernot, N., McDaniel, P., & Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277.

Sandler, M., Howard, A. G., Zhu, M., Zhmoginov, A., & Chen, L. (2018). Mobilenetv2: inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018* (pp. 4510–4520). Computer Vision Foundation / IEEE Computer Society.

Schwinn, L., Raab, R., & Eskofier, B. (2020). Towards rapid and robust adversarial training with one-step attacks. arXiv:2002.10097.

Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J. P., Studer, C., et al. (2019). Adversarial training for free!. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada* (pp. 3353–3364).

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., et al. (2014). Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*.

Teerapittayanon, S., McDanel, B., & Kung, H.-T. (2016). Branchynet: fast inference via early exiting from deep neural networks. In *2016 23rd international conference on pattern recognition (ICPR)* (pp. 2464–2469). IEEE.

Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I. J., Boneh, D., & McDaniel, P. D. (2018). Ensemble adversarial training: attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., & Madry, A. (2019). Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Vivek, B. S., & Babu, R. V. (2020). Regularizers for single-step adversarial training. arXiv:2002.00614.

Wong, E., Rice, L., & Kolter, J. Z. (2020). Fast is better than free: revisiting adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Yuan, X., He, P., Zhu, Q., & Li, X. (2019). Adversarial examples: attacks and defenses for deep learning. *30*, (9), (pp. 2805–2824).

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016*. BMVA Press.

Zhang, S., Huang, K., Zhu, J., & Liu, Y. (2021). Manifold adversarial training for supervised and semi-supervised learning. *140*, In *Neural Networks* (pp. 282–293).

Zhang, S., Qian, Z., Huang, K., Wang, Q., Zhang, R., & Yi, X. (2021). Towards better robust generalization with shift consistency regularization. In *Proceedings of Machine Learning Research*: *139*, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event* (pp. 12524–12534). PMLR.

Zhang, H., & Wang, J. (2019). Defense against adversarial attacks using feature scattering-based adversarial training. *Advances in Neural Information Processing Systems*, *32*, 1831–1841.

Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., & Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning* (pp. 7472–7482). PMLR.