2021 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, OCT. 25–28, 2021, GOLD COAST, AUSTRALIA

ROBUSTNESS-AWARE FILTER PRUNING FOR ROBUST NEURAL NETWORKS AGAINST ADVERSARIAL ATTACKS

*Hyuntak Lim, Si-Dong Roh, Sangki Park and Ki-Seok Chung**

Hanyang University
Department of Electronic Engineering
Seoul, Korea
{lim3944, sdroh1027, skpark1101, kchung}@hanyang.ac.kr

ABSTRACT

Today, neural networks show remarkable performance in various computer vision tasks, but they are vulnerable to adversarial attacks. By adversarial training, neural networks may improve robustness against adversarial attacks. However, it is a time-consuming and resource-intensive task. An earlier study analyzed adversarial attacks on the image features and proposed a robust dataset that would contain only features robust to the adversarial attack. By training with the robust dataset, neural networks can achieve a decent accuracy under adversarial attacks without carrying out time-consuming adversarial perturbation tasks. However, even if a network is trained with the robust dataset, it may still be vulnerable to adversarial attacks. In this paper, to overcome this limitation, we propose a new method called *Robustness-aware Filter Pruning* (RFP). To the best of our knowledge, it is the first attempt to utilize a filter pruning method to enhance the robustness against the adversarial attack. In the proposed method, the filters that are involved with non-robust features are pruned. With the proposed method, 52.1% accuracy against one of the most powerful adversarial attacks is achieved, which is 3.8% better than the previous robust dataset training while maintaining clean image test accuracy. Also, our method achieves the best performance when compared with the other filter pruning methods on robust dataset.

Index Terms— Deep Learning, Adversarial Attack, Adversarial Training, Filter Pruning

1. INTRODUCTION

Recent advances in neural networks have achieved great success in vision tasks such as image classification [1], image detection [2]. Correspondingly, neural networks have been adopted in various industries such as medical images [3], autonomous driving [4], etc. However, it is well-known that it is relatively easy to make neural networks malfunction. The adversarial attack, one of these attempts, incurs misclassification of a neural network by adding noise to the input image. The noise-injected images called adversarial examples can easily be classified by human eyes, but neural networks classify them into entirely wrong classes. This malfunction of a neural networks due to adversarial attacks may be fatal for safety-critical tasks such as autonomous driving.

To counter the adversarial attack, Madry *et al*.[5] formulated the adversarial training problem as Equation 1:

$$\min_{\theta} E_{(x,y)\sim\mathcal{D}} \left[\max_{\delta\in\mathcal{S}} \mathcal{L}(\theta, x+\delta, y) \right] \tag{1}$$

where the model parameter θ minimizes the risk of neural network loss, while δ maximizes the loss of neural network on input data x and lable y. Moreover, they proposed an attack called Projected Gradient Descent(PGD) as a solution for the inner max problem. For the outer min problem, they utilized Stochastic Gradient Descent(SGD) to minimize the loss of the adversarial examples created with the PGD attack. Methods proposed in [6, 7] utilized the gradient of a neural network to make adversarial examples. In these methods, the perturbation proceeds through several repetitions. Therefore, it takes a considerable amount of time to make adversarial examples and to train with them compared to the standard training.

In a recent study, Ilyas *et al*.[8] analyzed the adversarial attack from the image perspective. They divided features into robust features and non-robust ones. The robust features are useful for classifying both a clean image and an adversarial example. In contrast, the non-robust features are useful only when classifying a clean image while misbehaving when classifying an adversarial example. Based on this classification, Ilyas *et al*.made a robust dataset which would contain images with only the robust features. When training with the robust dataset, it achieved better robustness to adversarial attacks compared to the standard training, and training took less amount of time compared to the adversarial training. However, the performance of this method is often inferior to the adversarial training. In order to improve adversarial accuracy while maintaining the advantage of fast training, we propose a method called *Robustness-aware Filter Pruning* that removes

978-1-7281-6338-3/21/$31.00 ©2021 IEEE

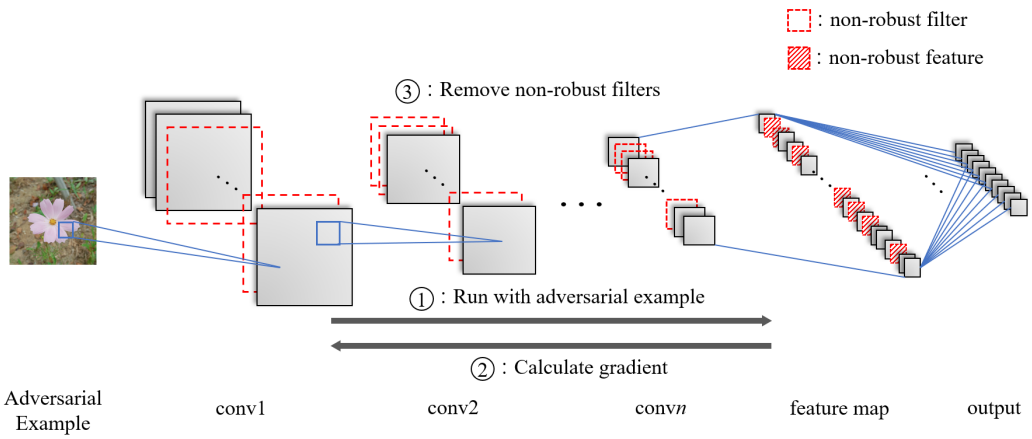


Fig. 1. Overview of the proposed method. First, put adversarial examples into a neural network. Second, calculate the gradient about the advesrarial examples. Third, with the calculated gradient, select non-robust filters and remove them.

filters that are involved with non-robust features (called as non-robust filters in this paper).

Unlike the ordinary filter pruning methods that utilize the norm of weights to characterize filters, we adopt the gradient of a neural network to define the non-robust filters. We experimentally confirm that the adversarial training should reduce the influence of non-robust features by training the non-robust filters with larger-scale gradients. Based on this result, we figure out the non-robust filters and remove them.

The overview of the proposed *Robustness-aware Filter Pruning* is shown in Fig. 1. Our experimental results verify that the proposed method achieves an improved adversarial accuracy of up to 52.1%.

2. RELATED WORKS

2.1. Robust Dataset

Previously, the adversarial attack was considered as a linearity problem of a neural network [6]. Correspondingly, the linearity of a neural network was exploited to malfunction even with the adversarial examples that were slightly over the decision boundary. Ilyas *et al*.[8] tried to analyze adversarial attacks from the feature perspective of an image. According to the research, an image has both robust features that are robust to the adversarial attacks and non-robust ones that are vulnerable to the attacks. The study claims that both robust and non-robust features are useful when classifying a clean image set. However, non-robust features hinder the neural network from classifying an image correctly under adversarial attacks. Based on this claim, they have made a robust dataset that would contain only the robust features by removing non-robust features from the original image. Specifically, the following optimization is applied to the original dataset to figure out the robust dataset:

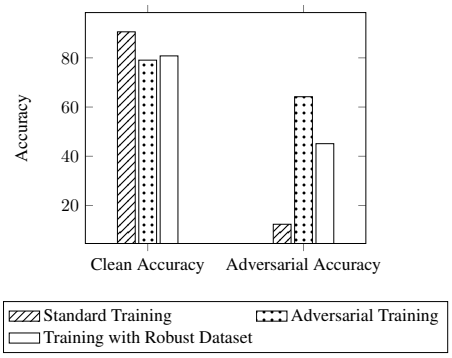


Fig. 2. Accuracy of the standard training, the adversarial training, and the training with a robust dataset. The adversarial training was carried out as proposed in Madry *et al*.[5]. The experiment proceeded with VGG-16 as a network model and CIFAR-10 as both the training and the test datasets. The training with robust dataset achieved the adversarial accuracy of 45.12% while that of the standard training was only 12.38%.

$$\min_{x_{robust}} \| \mathrm{g}(x_{robust}) - \mathrm{g}(x_{origin}) \|_2 \tag{2}$$

where g is the mapping from an input to the representation layer that is adversarially trained. Because non-robust features have been removed in the robust dataset, neural networks trained with a robust dataset can endure the adversarial attack, as seen in Fig. 2.

2.2. Robustness of Filter Pruning

In this paper, we claim that removing the non-robust filters is an effective way to make a neural network robust against adversarial attacks. Quite a few previous studies [9, 10] have shown that pruning should make the neural network robust.

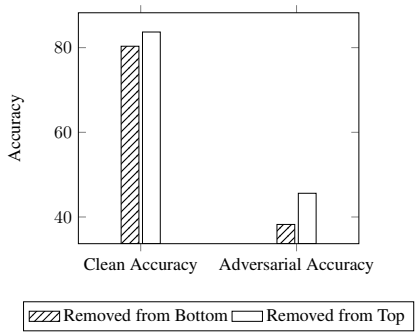


Fig. 3. Comparison of removal filters from a sorted list in descending order. Both methods proceeded with the VGG-16 model and CIFAR-10 for training, ROBUST CIFAR-10 for fine-tuning. 60% of filters are pruned either from the top or from the bottom.

However, the previous studies mainly dealt with weight pruning that makes a set of weight values 0. On the other hand, filter pruning removes some filters entirely from a convolutional layer. It is hard to compare the effectiveness of the filter pruning and that of the weight pruning directly because the pruning detail is quite different, and weight pruning needs some sparse libraries or specialized hardware [11]. However, we want to claim that filter pruning is a very effective way to enhance the robustness of a neural network. To the best of our knowledge, this is the first attempt to verify that the filter pruning is effective to make the network more robust against adversarial attacks.

The existing filter pruning methods utilize the norm of weights in filters. Li *et al*.[11] proposed a method called n-Norm pruning where the filter with the smallest norm in each layer is pruned. It is based on the observation that the smaller norm of a filter implies that the filter is less important. They used the l_1 and l_2 norms to prune filters. Similarly, He *et al*.[12] proposed another pruning method called Filter Pruning via Geometric Mean(FPGM) utilizing the norm. They focused on the norms that deviate from the distribution. They calculated the geometric mean of norms and pruned the filters with norms far from the mean.

3. PROPOSED METHOD

In this paper, we propose a method called *Robustness-aware Filter Pruning* (RFP) that makes the neural network robust by removing the non-robust filters. The overview of RFP is illustrated in Fig. 1.

In filter pruning methods, finding the right filter to prune is the main problem to solve. In this paper, finding the right filter to prune in order to enhance the adversarial accuracy is the main problem. To solve this problem, we utilize the gradients of filters while the other filter pruning methods use norms calculated with weights of filters.

Algorithm 1: Algorithm RFP

Data: X, X', X^{robust}
Result: The compact and robust model with parameters W^*

Initialize parameters $W = W^{(i)},\ 0 \leq i \leq L$
for $epoch = 1;\ epoch \leq epoch_{max};\ epoch++$ **do**
    Update the model parameter W based on X

$X' \leftarrow l_2PGD(W, X')$
$G \leftarrow \nabla_{X'}\mathcal{L}(W, X')$

for $i = 0;\ i \leq L;\ i++$ **do**
    Find N_iP filters that satisfy Equation 3 with G
    Remove selected filters

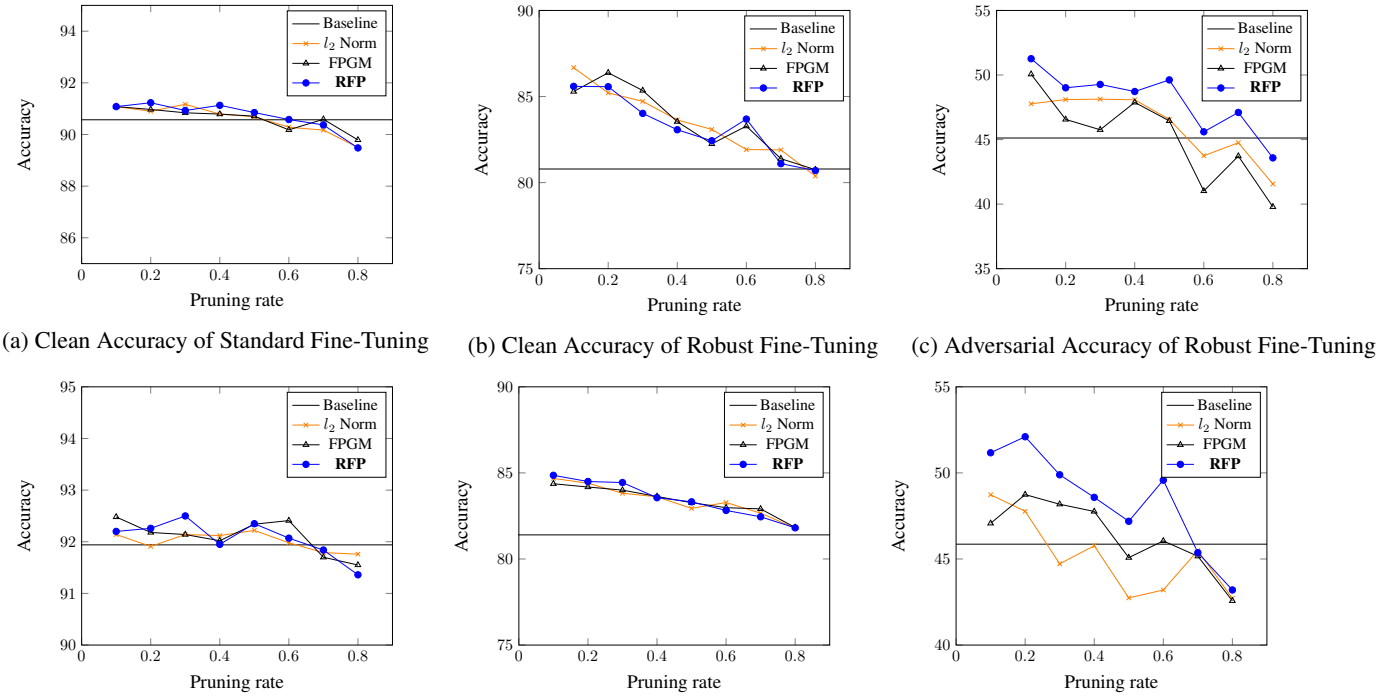
#Fine-tune
for $epoch = 1;\ epoch \leq epoch_{fine-tune};\ epoch++$ **do**
    Update the model parameter W based on X^{robust}

3.1. Gradient-based Robustness Test

As aforementioned, Ilyas *et al*.[8] utilized the features in the representation layer of a convolutional neural network to construct a robust dataset. To retain only the robust features, the images in the original dataset were mapped to the features in the representation layer of the adversarially trained neural network. The neural network trained with these images was robust against the adversarial attack. This indicates adversarially trained network retains only the robust features in the representation layer. From this, we can infer that the adversarial training restrains the non-robust filters in order to exclude the non-robust features from the representation layer.

The neural network trained by a standard method utilizes both robust and non-robust features to classify an image. This means all the weights involved with robust and non-robust features are activated in the standard training. Therefore, when the adversarial training is conducted after the standard training is done, the training should focus on deactivating the non-robust features.

In this paper, we claim that the gradient magnitude of the non-robust features should be bigger than that of the robust features. To justify this claim, two different methods to remove filters are attempted. After we train a network with the original dataset, adversarial examples are added, and the gradients are computed. Based on the gradient magnitudes, filters are sorted in descending order. Now, two cases are compared: (1) 60% of filters in a layer are removed from the top of the sorted filter list, (2) 60% of them are removed from the bottom of the list. Fig. 3 shows 83.69% of the clean accuracy and 45.6% of the adversarial accuracy when the filters are removed from the top of the list. When the filters are removed from the bottom, the clean accuracy is 80.33%, and the adver-



(a) Clean Accuracy of Standard Fine-Tuning (b) Clean Accuracy of Robust Fine-Tuning (c) Adversarial Accuracy of Robust Fine-Tuning

(d) Clean Accuracy of Standard Fine-Tuning (e) Clean Accuracy of Robust Fine-Tuning (f) Adversarial Accuracy of Robust Fine-Tuning

Fig. 4. Test results of pruning methods with l_2 norm. (a), (b), (c) are the test results of VGG-16, and (d), (e), (f) are the test results of ResNet-18. All the results are estimated with CIFAR-10.

sarial accuracy is 38.24%. Both the clean and the adversarial accuracies for the case where the filters are removed from the top are much better. From these experimental results, we claim that the weights of non-robust filters change abruptly when the adversarial training is conducted after the standard training is done.

Based on the following key claims; 1) Adversarially trained neural networks utilize only the robust features, 2) The weights of non-robust filters change abruptly when the adversarial training is being conducted after the standard training is done, we define the non-robust filter as:

$$\underset{\mathcal{F}}{\operatorname{argmax}} \sum_{\mathcal{F}_j \in L_i} \parallel \nabla_{x^{adv}} \mathcal{L}\left(\mathcal{F}_j, y, x^{adv}\right) \parallel_2 \qquad (3)$$

where $\mathcal{F}$ is the filter that maximizes the sum of l_2 norms of the gradient for adversarial example x^{adv} in convolutional layer L_i. Depending on the pruning rate, more than one $\mathcal{F}$ may be found. The l_2 norm is computed with the gradients to take the direction of gradients as well as the magnitude into consideration.

3.2. Robustness-aware Filter Pruning

Based on the key claims mentioned above, we propose a new scheme called *Robustness-aware Filter Pruning* (RFP) that removes the non-robust filters to make the neural network robust. RFP is described in Algorithm 1. In RFP, after the standard training with the original dataset is conducted, the gradients of the adversarial examples with the l_2 PGD attack are calculated. Then, we calculate the l_2 norm of each filter in a convolutional layer with the calculated gradients. Based on the l_2 norm values, filters are sorted in descending order, and a set of filters are pruned from the top. The number of filters to be pruned is determined by the pruning rate. Experimental results will verify the effectiveness of the proposed method.

4. EXPERIMENTS

In this section, the implementation details and the experimental environment will be addressed. The experiments are carried out with PyTorch 1.6 on an NVIDIA TITAN RTX GPU.

4.1. Setup

Datasets and Networks Experiments are conducted on CIFAR-10 [13] and ROBUST CIFAR-10 [8] as datasets. The ROBUST CIFAR-10 dataset is used only as the fine-tuning data while all the evaluations proceed with the CIFAR-10 test set. Two network architectures are chosen: VGG [14] and ResNet [15]. Because it is hard to know the robust features of each image, 100 images are randomly selected from each class. So, a total of 1,000 images is used as the adversarial

model	pruning rate	l_1 Norm		l_2 Norm	
		Clean Accuracy	Adversarial Accuracy	Clean Accuracy	Adversarial Accuracy
		(n-Norm [11] / FPGM [12] / **RFP**)			
VGG-16	0 (baseline) [8]	80.79	45.12	80.79	45.12
	0.2	86.12 / 86.51 / 84.66	45.93 / 45.99 / **47.60**	86.38 / 85.21 / 85.57	46.56 / 48.10 / **49.01**
	0.4	84.26 / 84.25 / **85.01**	44.05 / 43.29 / **45.60**	83.54 / 83.64 / 83.07	47.88 / 48.08 / **48.72**
	0.6	80.52 / 83.68 / 83.21	39.58 / 38.14 / **42.94**	83.28 / 81.92 / **83.69**	41.02 / 43.75 / **45.60**
	0.8	78.73 / 78.02 / 78.03	34.96 / 28.99 / **36.23**	80.75 / 80.38 / 80.70	39.79 / 41.56 / **43.58**
ResNet-18	0 (baseline) [8]	81.4	45.86	81.4	45.86
	0.2	87.24 / 86.50 / **87.29**	45.37 / 46.99 / 46.85	84.18 / 84.40 / **84.50**	48.74 / 47.77 / **52.10**
	0.4	86.89 / 85.74 / 85.83	45.11 / 44.47 / **45.66**	83.63 / 83.61 / 83.56	47.76 / 45.77 / **48.58**
	0.6	84.86 / 83.86 / 83.72	39.03 / 41.05 / **41.61**	82.98 / 83.28 / 82.82	46.05 / 43.20 / **49.58**
	0.8	82.01 / 81.27 / 79.45	32.51 / 32.04 / 31.18	81.83 / 82.66 / 81.81	42.57 / 42.71 / **43.20**

Table 1. Accuracy comparison of pruning methods with l_1 and l_2 norm on VGG-16 and ResNet-18, fine-tuning with ROBUST CIFAR-10. The baseline[8] results are reproduced.

examples. All the training proceeds with Stochastic Gradient Descent(SGD) and the same hyper-parameter settings.

Filter Pruning Methods RFP is compared with the existing two filter pruning methods. The n-Norm pruning with the l_2 norm(l_2 Norm) [11] prunes the filters with smaller l_2 norms. Originally, the l_1 norm was used to prune filters in [11], but the l_2 norm was additionally used to compare the method with RFP under the same conditions. In addition, another method called *Filter Pruning via Geometric Mean*(FPGM) [12] is compared. In FPGM, the filters that are farther from the geometric mean of the l_2 norm of the filters are pruned.

Adversarial Attack Method As the adversarial attack, the Projected Gradient Descent(PGD) attack with the l_2 norm is used. The PGD attack is known to be one of the most powerful attacks. As used in [8], the $\epsilon = 0.25$ for the PGD attack on CIFAR-10 is used. When calculating the gradients in RFP, we set $\epsilon = 0.3$ to reveal more drastic effect of the adversarial attack and to make a clear distinction from the robust filters.

4.2. Experimental Results

In Fig. 4, (a), (b) and (c) summarize the experimental results on VGG-16 while (d), (e) and (f) summarize results on ResNet-18. (a) and (d) are fine-tuned on the original dataset (called Standard Fine-Tuning), and the others are fine-tuned with the aforementioned robust dataset (called Robust Fine-Tuning). All baselines are trained with datasets without pruning and fine-tuning. In other words, the pruning rate is 0 for the baseline.

VGG-16 Applying the compared pruning methods and RFP to the VGG-16 model, clean accuracies are similar to that of the standard training. The baseline network shows 90.57% clean accuracy. The l_2-Norm pruning achieves the clean accuracy of 90.61% and FPGM achieves 90.57%. Similarly, RFP achieves the average accuracy of 90.7%. With the robust training, the baseline shows 80.79% clean accuracy, and the three compared pruned networks show the average accuracy of 83.44%, 83.52%, 83.31%, respectively.

In case of the adversarial accuracy, RFP achieves a superior performance. While the baseline shows 45.12% adversarial accuracy, RFP achieves 51.26% adversarial accuracy with 0.1 as the pruning rate. Furthermore, RFP shows a better accuracy as the pruning rate is getting bigger. While pruning 50% of the total parameters, the average adversarial accuracy is 49.57% and the clean accuracy is 84.17%.

ResNet-18 In [8], ResNet-18 was employed and 84.1% of clean accuracy and 48.27% adversarial accuracy were achieved. Somehow, when we reproduced the results using the same evaluation setting, 81.4% clean accuracy and 45.86% adversarial accuracy were attained.

Similar to the VGG-16 case, experiments with ResNet-18 demonstrate similar levels of clean accuracy to that of the standard training. RFP shows the average clean accuracy of 83.47%, while the highest accuracy is 84.86% with the robust training. In the case of the adversarial accuracy, RFP achieves an outstanding performance overall. It achieves 52.1% adversarial accuracy, which is better than the method in [8] by 3.8%.

The results show that pruning over 60% of parameters is not much effective for both clean and adversarial test. We analyze that it is because of the number of non-robust filter is limited. Therefore, as pruning rate increases, the adversarial accuracy decreases. And also, because the non-robust filters are as useful as robust filters for classifying clean data, it is hard to avoid the degradation in performance when pruning rate increases.

4.3. Comparison with the l_1 and l_2 norms

RFP utilizes the l_2 norm to consider the directional information of the gradient. Table 1 shows the comparison results of the pruning methods with l_1 and l_2 norms. With the l_2 norm, RFP shows the best adversarial accuracy with respect to every pruning rate. Regarding the l_1 norm, the clean accuracy of RFP is better. Based on the excellent adversarial accuracy of RFP with respect to the l_2 norm, it can be claimed that considering the directional information of the gradient is very effective. Therefore, in Section 4.2, all the methods were compared with the l_2 norm. On the other hand, the n-Norm pruning works well on l_1 norm. When the l_1-Norm pruning is compared with RFP, RFP works better on the adversarial accuracy.

5. CONCLUSION

In this paper, we proposed a novel method called *Robustness-aware Filter Pruning* (RFP) that prunes the filters that are involved with non-robust features. RFP achieved 52.1% adversarial accuracy, while the previous study with a robust dataset achieves at best 45.86%. In the future, we plan to measure the amount of robust features in an image and study the correlation with the accuracy of RFP.

6. ACKNOWLEDGEMENT

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2020-0-01304, Development of Self-learnable Mobile Recursive Neural Network Processor Technology)

7. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," in *6th International Conference on Learning Representations, ICLR*, 2018.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[7] Nicholas Carlini and David A. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy, SP*, 2017.

[8] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry, "Adversarial examples are not bugs, they are features," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS*, 2019.

[9] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen, "Sparse dnns with improved adversarial robustness," in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS*, 2018.

[10] Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang, "Adversarial neural pruning with latent vulnerability suppression," in *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2020.

[11] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf, "Pruning filters for efficient convnets," in *5th International Conference on Learning Representations, ICLR*, 2017.

[12] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang, "Filter pruning via geometric median for deep convolutional neural networks acceleration," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019.

[13] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 (canadian institute for advanced research)," .

[14] Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016.