Adversarial Neural Pruning with Modified Latent Vulnerability utilizing Feature Robustness

Hyuntak Lim, Ki-Seok Chung
Dept. of Electronic Engineering, Hanyang University, Seoul, Korea
{lim3944, kchung}@hanyang.ac.kr

Abstract

Today, vision recognition with Convolutional Neural Network (CNN) shows performance good enough to be employed in safety-critical autonomous driving. However, CNN models are vulnerable to adversarial attacks. To overcome this vulnerability, many methods have been studied. Pruning is regarded as one of the effective methods to make the network more robust against adversarial attacks. In this paper, we introduce a new vulnerability loss to suppress the vulnerability better when the pruning is used to counter adversarial attacks. With this vulnerability suppression, we achieve up to 1.12% better accuracy against the adversarial examples compared to a previous study called ANP-VS.

Keywords: Adversarial Attack, Weight Pruning,

1. Introduction

The advent of Convolutional Neural Network (CNN) has led to significant success in various vision tasks such as image classification [1], image detection [2], and semantic segmentation [3]. However, CNN is vulnerable to the adversarial attack that injects some noise into the input image with a malicious intention to cause a malfunction of neural networks.

To counter the adversarial attack, several adversarial training methods have been proposed [4,5]. Pruning is regarded as an effective way to make a neural network more robust to adversarial attacks. Madaan et al. [6] introduced a method called Adversarial Neural Pruning with Vulnerability Suppression (ANP-VS) where a weight pruning method was used to suppress the vulnerability of a neural network by constructing a pruning mask. While finetuning the pruned network, the latent vulnerability is suppressed with the vulnerability suppression loss that they introduced.

Ilyas et al. [7] analyzed the adversarial attack from the image's perspective. They defined robust and non-robust features. The robust features are useful for classifying both clean images and adversarial examples. However, the non-robust features are useful only for classifying clean images while they are harmful to classifying the adversarial examples.

In this paper, we introduce a new type of vulnerability. To take this vulnerability into account, we propose a new vulnerability suppression loss function. By training with this modified loss, both the clean accuracy and the adversarial accuracy are enhanced compared with existing works.

2. Background

2.1 Adversarial Attack

It is well-known that even a simple adversarial attack can cause a neural network to misclassify an image. Commonly, an adversarial example is generated using the following equation:

$$\max_{\delta\in\mathcal{S}} \mathcal{L}(\theta, x+ \delta, y) \qquad (1)$$

where θ is a model parameter and δ is the noise that turns an image into an adversarial example.

The Projected Gradient Descent (PGD) attack [5] is known to be one of the most powerful adversarial attacks. Based on Fast Gradient Sign Method (FGSM) [4], PGD repeatedly conducts the following perturbation to generate a more powerful adversarial example.

$$\tilde{x}^{t+1} = \prod_{\mathcal{B}(x,\mathcal{E})} (\tilde{x}^{t} + \alpha \cdot sgn(\nabla_x \mathcal{L}(\theta, x, y)) \qquad (2)$$

2.2 Adversarial Neural Pruning

As mentioned above, ANP-VS was introduced as a pruning method to mitigate the vulnerability to adversarial attacks. ANP-VS learns pruning masks for the features in a Bayesian framework [8] to minimize the following adversarial loss:

$$\min_{M} \mathbb{E}_{(x,y)\sim\mathcal{D}}\{ \max_{\delta\in\mathcal{B}(x,\varepsilon)} \mathcal{L}(\theta\odot M, \tilde{x}, y) \} \qquad (3)$$

where M is a pruning mask that minimizes the loss for adversarial example $\tilde{x}$ that maximizes the loss of a neural network.

2.3 Robust and Non-Robust Features

The adversarial attack used to be regarded as a linearity problem of a neural network [4], which means the linearity of a neural network is exploited to malfunction even with the adversarial examples that are slightly over the decision boundary. However, Ilyas et al. [7] defined the robust and non-robust features of an image in terms of usefulness. They scored a feature as ρ-useful and γ-robustly useful . They defined a feature as a function mapping from the input space $\mathcal{X}$ to a real number so that the set of all features can be denoted as $\mathcal{F} = \{f : \mathcal{X} \rightarrow \mathcal{R}\}$. For simplicity, they assumed that the features in $\mathcal{F}$ will be manipulated to be mean-zero and unit-variance.

For a given distribution $\mathcal{D}$, if a certain feature *f* is correlated with the true label, the value of *f*, ρ will be bigger than zero. Then, we call feature *f* ρ-useful. Further, if a ρ-useful feature, under adversarial perturbation, remains γ-useful, the feature is referred to as γ-robustly useful. In other word, the value of feature *f* can be bigger than zero in both clean and adversarial situations.

Based on these usefulness definitions, features are classified as robust or non-robust. If a feature is ρ-useful and γ-robustly useful, it is a robust feature meaning that the feature is useful for both clean image and adversarial examples. In contrast, if a feature is only ρ-useful, it is a non-robust feature because it is not useful for adversarial examples.

Based on this feature classification, Ilyas et al. constructed a robust dataset that is a set of images whose non-robust features have been removed. With this robust dataset, they demonstrated the neural network should be more robust against the adversarial attack.

3. Proposed Method

The authors of [6] defined the vulnerability of the *k*th latent feature for the *l*th layer as the Manhattan distance between the feature value for a clean example z_{lk} and an adversarial example $\tilde{z}_{lk}$ as follows:

$$v(z_{lk}, \tilde{z}_{lk}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}||z_{lk} - \tilde{z}_{lk}|| \qquad (4)$$

They expanded the definition to a layer and the whole network as follows:

$$\bar{v}_l = \frac{1}{N_l}\sum_{k=1}^{k=N_l} v(z_{lk}, \tilde{z}_{lk}),$$

$$V(f_\theta(X), f_\theta(\tilde{X})) = \frac{1}{L-2}\sum_{l=1}^{l=L-2} \bar{v}_l \qquad (5)$$

where $\bar{v}_l$ is the vulnerability of the *l*th layer and $V(f_\theta(X), f_\theta(\tilde{X}))$ is the vulnerability of a neural network that is the average of the vulnerability of all layers except for the last fully-connected layer.

Now that the non-robust features are not useful for classifying the adversarial examples, we introduce a modified definition of vulnerability of a neural network. We specify the vulnerability of a latent-feature as follows:

$$v(z_{lk}, \tilde{z}_{lk}) = \mathbb{E}_{(x,y)\sim\mathcal{D}} Vul(z)$$

$$Vul(z) = \begin{cases} ||z_{lk} - \tilde{z}_{lk}|| & if\ z_{lk} \geq 0\ and\ \tilde{z}_{lk} \leq 0 \\ \lambda||z_{lk} - \tilde{z}_{lk}|| & else \end{cases} \qquad (6)$$

where λ is a hyperparameter that decides how much the *else* case comes into loss. Without normalizing the features, we set a hyperparameter λ to include the misguided feature.

As in [6], we prune a neural network with a Bayesian pruning framework that finds the pruning mask that minimizes the loss with adversarial examples. After pruning, we train the network with the vulnerability suppression loss as in [6]. The vulnerability suppression loss is computed as follows:

$$\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}\{\mathcal{L}(\theta \odot M, x, y) + \alpha \cdot V(f_\theta(x), f_\theta(\tilde{x}))\} \qquad (7)$$

where $\mathcal{L}(\theta \odot M, x, y)$ is a classification loss and α is a hyperparameter determining the strength of the vulnerability loss.

4. Experiments

For experiments, we used the Pytorch framework on a TITAN RTX GPU. We conducted a similar set of experiments to that in [6]. For the MNIST dataset [9], we conducted evaluation with the Lenet-5 model [10]. For CIFAR-10 and CIFAR-100 [11], we used the VGG-16 model [12]. The proposed method was evaluated with an adversarial accuracy under $\ell_\infty - PGD$ attack, $\varepsilon = 0.3$ for MNIST, $\varepsilon = 0.03$ for CIFAR-10 and CIFAR-100. The other details followed [6]. The results of ANP-VS were reproduced for fair comparison.

Table 1 Experiment Results

Model	Training Method	Clean Accuracy	Adversarial Accuracy	Sparsity
Lenet-5 (MNIST)	Standard	99.37%	0.00%	0.00%
	ANP-VS	98.59%	94.09%	83.94%
	Ours(λ = 0)	98.58%	95.12%	84.23%
	Ours(λ = 0.3)	98.68%	94.07%	**84.52%**
	Ours(λ = 0.5)	**98.86%**	**95.21%**	84.08%
VGG-16 (CIFAR-10)	Standard	92.41%	13.57%	0.00%
	ANP-VS	87.50%	56.90%	78.20%
	Ours(λ = 0)	84.71%	56.68%	**78.23%**
	Ours(λ = 0.3)	**87.84%**	**57.90%**	77.95%
	Ours(λ = 0.5)	87.47%	87.06%	78.12%
VGG-16 (CIFAR-100	Standard	66.35%	3.20%	0.00%
	ANP-VS	56.30%	23.29%	68.14%
	Ours(λ = 0)	56.72%	23.09%	68.44%
	Ours(λ = 0.3)	**59.22%**	**23.35%**	**68.44%**
	Ours(λ = 0.5)	56.32%	22.41%	68.29%

Table 1 summaries the comparison results. We compared our method with a standard training and ANP-VS [6]. With Lenet-5 on the MNIST dataset, we achieved 0.27% and 1.12% enhanced clean accuracy and adversarial accuracy with λ=0.5, respectively. Moreover, not only the performance is improved but also the sparsity is lower. For the CIFAR-10 dataset, the results of our method demonstrate 0.84% and 1% improved clean accuracy and adversarial accuracy with λ=0.3, respectively. On CIFAR-100, the clean accuracy is enhanced by 2.92% and the adversarial accuracy with λ=0.3 is enhanced by 1.06%.

When λ=0, the adversarial accuracy of the proposed method is slightly lower than that of ANP-VS. This means considering the *else* case of the modified vulnerability is reasonable when the feature normalizing is omitted.

5 Conclusion

In this paper, we introduced a modified definition of the vulnerability loss of features based on the robustness and the non-robustness of features. With the modified definition of the vulnerability, we achieved up to 1.12% enhanced accuracy on adversarial examples and 2.92% enhanced clean accuracy compared to a study called ANP-VS.

Acknowledge

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-00131, Development of Intelligent Edge Computing Semiconductor For Lightweight Manufacturing Inspection Equipment)

References

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems 25 (2012): 1097-1105.

[2] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[3] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[4] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).

[5] Madry, Aleksander, et al. "Towards deep learning models resistant to adversarial attacks." arXiv preprint arXiv:1706.06083 (2017).

[6] Madaan, Divyam, Jinwoo Shin, and Sung Ju Hwang. "Adversarial neural pruning with latent vulnerability suppression." International Conference on Machine Learning. PMLR, 2020.

[7] Ilyas, Andrew, et al. "Adversarial examples are not bugs, they are features." arXiv preprint arXiv:1905.02175 (2019).

[8] Liu, X., Li, Y., Wu, C., and Hsieh, C.-J. Adv-BNN: Improved adversarial defense through robust bayesian neural network. ICLR, 2019.

[9] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[10] LeCun, Yann. "LeNet-5, convolutional neural networks." URL: http://yann. lecun. com/exdb/lenet 20.5 (2015):14.

[11] Krizhevsky, Alex, and Geoffrey Hinton. "Learning multiple layers of features from tiny images." (2009): 7.

[12] Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image