APE-GAN: ADVERSARIAL PERTURBATION ELIMINATION WITH GAN

Guoqing Jin[1] *Shiwei Shen*[1] *Dongming Zhang*[2] *Feng Dai*[1] *Yongdong Zhang*[3]

[1] Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
[2]National Computer Network Emergency Response Technical Team of China, Beijing, China
[3]University of Science and Technology of China, Hefei, China

ABSTRACT

Although Deep Neural Networks could achieve state-of-the-art performance while recongnizing images, they often suffer a tremendous defeat from adversarial examples–inputs generated by utilizing imperceptible but intentional perturbations to samples from the datasets. So far, very few methods have provided a significant defense to adversarial examples. In this paper, an effective framework based Generative Adversarial Nets(GAN) is proposed to defense against the adversarial examples. The essense of the model is to eliminate the adversarial perturbations being highly aligned with the weight vectors of nueral models. Extensive experiments on benchmark datasets MNIST, CIFAR10 and ImageNet indicate that our framework is able to defense against adversarial examples effectively.

Index Terms— Adversarial examples, Adversarial perturbations elmination, Adversarial attack, Generative Adversarial Nets, Deep neural networks

1. INTRODUCTION

Deep neural networks have recently achieved excellent performance on a variety of visual and speech recognition tasks. However, they have intrinsic blind spots that are easy to be attacked using obscure manipulation of their inputs[1, 2]. When the infinitesima perturbations being highly aligned with the weight vectors, the neural networks' linear behavior in high-dimensional most likely accumulates one large change to the output. Szegedy et al.[3] first noticed that imperceptible perturbation of samples can be misclassified by deep neural networks. Opposite to *clean examples*, they term this kind of subtle perturbed samples "*adversarial examples*".

Adversarial examples pose potential security threats for practical machine learning applications. Recent research[2] shows that a large fraction of adversarial examples are classified incorrectly even when obtained from the cell-phone camera. It is possible that adversarial images of traffic signs cause the vision-based self-driving car to take disastrous actions. Therefore, the research of resisting adversarial examples is very significant and urgent.

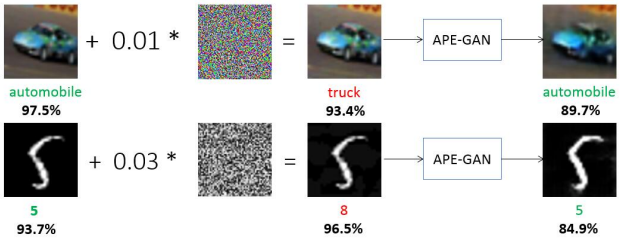

Fig. 1. A certain imperceptiable pertubations of images can lead to incorrected detection. The proposed APE-G model undermines the infinistesimal perturbation of the adversarial examples before it be feed into the deep learning models.

Unfortunately, traditional strategies such as pretraining and dropout are not conducive to improving the model's robustness to adversarial examples. Carlini et al.[4] proved that adversarial examples are significantly harder to detect and reject. Adversarial training[1] and Defensive Distillation[5], that back-feed adversarial examples to training, do provide an additional regularization benefit of the resulting models, but fail to reduce the model's vulnerability to new adversarial perturbations. Therefore, defense against adversarial examples is still a huge challenge.

Since the high-dimensional linear nature of deep nueral network can hardly be avoided in practical application, it becomes more difficult to defend against adversarial examples. But the generalization of adversarial examples across different models[1] provides new clues to defense against adversarial attack. Compared with clean examples, adversarial examples contain infinitesimal perturbations that add up to one large change to the output[1]. It is possible to defense against adversarial examples if there is an algorithm that can dispel or eliminate the infinitesimal perturbations of the samples.

In this paper, a novel framework based Generative Adversarial Nets[6] is proposed to eliminate adversarial perturbations of adversarial examples before being recognized by the machine learning systems, as show in Figure 1. The gen-

Dongming Zhang is the corresponding author. This work was supported in part by the National Key Research and Development Plan of China (2018YFB0804202), in part by the National Natural Science Foundation of China (61672495, 61525206, 61571424),in part by the Beijing Municipal Science and Technology Project(171100000117010).

erative model learns a mapping from adversarial examples' manifold to clean examples', which eliminates the adversarial perturbations being highly aligned with the weight vectors of deep nueral models. To achieve this goal, several task specified loss functions is invented to make the adversarial examples being high consistent with the clear examples.

2. PROPOSED METHOD

Construction of adversarial example can be formulated as a small enought perturbation ε of input X and X_ϵ that satisfies

$$\|X_\epsilon - X\| = \epsilon \tag{1}$$

but $f(X) \neq f(X_\epsilon)$, where X is the clean example, X_ϵ is the adversarial example and f is the classifer mapping from input image to a discrete label set. The fundamental idea of defending against adversarial examples is to eliminate or damage the trivial perturbations ϵ of the input X_ϵ before being recognized by the target model.

The minimax game of GAN has a global optimum $p_g = p_{\textbf{data}}$, where p_g is the generative distribution and $p_{\textbf{data}}$ is the samples from the data generating distribution. Thus, we can formulate the elimination of adversarial perturbation ϵ as

$$p_{\|X_\epsilon - \epsilon\|} = p_{\|X\|} \tag{2}$$

The procedure of converging to a good estimator of $p_{\|X_\epsilon - \epsilon\|}$ is to elimimate the adversarial perturbations ϵ.

Based on the above analysis, a novel framework based GAN to eliminate the adversarial perturbations is proposed, which is termed APE-GAN(adversarial perturbation elimination with GAN), as shown in Figure 2. The APE-GAN network is trained in an adversarial setting. While the generator APE-G is trained to alter the perturbation with tiny changes to the input examples, the discriminator APE-D is optimized to seperate the clean examples and reconstructed examples without adversarial perturbations obtained from APE-G. To achieve this, a task specified fusion loss function is invented to make the adversarial examples highly consistent with original clean image manifold.

2.1. Architecture

The ultimate goal of APE-GAN is to train a generating function G that gets rid of the imperceptiable but intentional perturbations of the adversarial input image X_ϵ. To achieve this goal, a generator network parametrized by θ_G is trained. Here θ_G denotes the weights and baises of a generate network and is obtained by optimizing an adversarial perturbation elimination specified loss function l_{ape}. With training images X_ϵ^k obtained by applying FGSM and corresponding original clean image X^k, $k = 1, ..., N$, we solve:

$$\hat{\theta}_G = \arg\min_{\theta_G} \frac{1}{N} \sum_{k=1}^{N} l_{ape}(G_{\theta_G}(X_\epsilon^k), X^k) \tag{3}$$

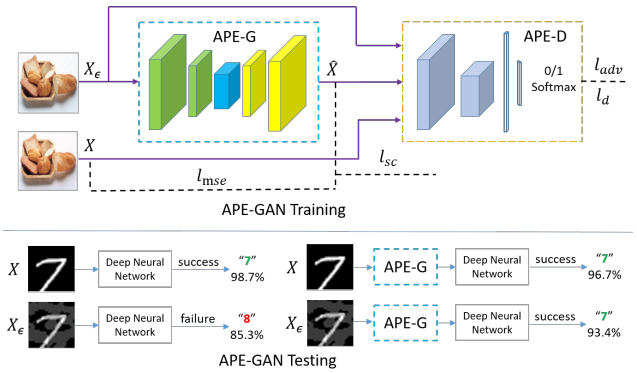

Fig. 2. We propose an adversarial perturbations elimination framework named APE-GAN to eliminate the perturbation of the adversarial examples before feeding them into the target model to defense against adversarial attack.

A discriminator network D_{θ_D} along with G_{θ_G} is defined to solve the adversarial zero sum problem:

$$\min_{\theta_G} \max_{\theta_D} \quad \mathbb{E}_{X \sim p_{data}(X)} \log D_{\theta_D}(X) - \mathbb{E}_{X_\epsilon \sim p_G(X_\epsilon)} \log(D_{\theta_D}(G_{\theta_G}(X_\epsilon))) \tag{4}$$

The general idea behind this formulation is that it allows one to train a generative model APE-G with the goal of deceiting a differentiable discriminator APE-D, which is trained to tell apart adversarial perturbations eliminated images $\hat{X}$=APE-G(X_ϵ) from original clean images. Consequently, the generator can be trained to produce reconstructed images that are not only highly similar to original clean images but also rid of adversarial perturbations, and thus APE-D is unable to distinguish them.

The general architecture of our generator network APE-G is illustrated in Figure 2. Some convolutional layers with stride = 2 are leveraged to get feature maps with lower resolution and followed by some deconvolutional layers with stride = 2 to recover the original resolution.

To discriminate original clean images X from reconstructed images $\hat{X}$, we train a discriminator network APE-D. The general architecture is illustrated in Figure 2. The discriminator network is trained to solve the maximization problem in Equation 4. It also contains some convolutional layers with stride = 2 to get some high-level feature maps, two dense layers and a final sigmoid activation function to obtain a probability for samples classification.

2.2. Loss Function

The definition of our adversarial elimination specified loss function l_{ape} is critical for the performance of our generator network to produce images without adversarial perturbations. We define l_{ape} as the weighted sum of several loss functions

as:

$$l_{ape} = \xi_1 l_{mse} + \xi_2 l_{adv} + \xi_3 l_{sc} \quad (5)$$

where consist of pixel-wise MSE(mean square error) loss, adversarial loass and spatially coherent loss.

Adversarial perturbations can be viewed as a special noise constructed delicately. The most widely used loss for image denoise or super resolution will be able to achieve satisfactory results for adversarial elimination. Inspired by image super resolution method[7], the pixel-wise MSE loss is defined as:

$$l_{mse} = \frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} (X_{w,h} - G_{\theta_G}(X_\epsilon)_{w,h}) \quad (6)$$

where w and h are the coordinates of the image.

To encourage our network to produce images residing on the manifold of original clean images, the generative loss of the GAN is also employed. The adversarial loss l_{adv} is calculated based on the probabilities of the discriminator over all adversarial images as:

$$l_{adv} = \sum_{n=1}^{N} [1 - \log D_{\theta_D}(G_{\theta_G}(X_\epsilon))] \quad (7)$$

In addition to the adversarial loss, we also add a spatially coherent loss based on the total variation to l_{ape}, which is calculated as:

$$l_{sc} = \frac{1}{WH} \sum_{w=1}^{W} \sum_{h=1}^{H} \|\nabla G_{\theta_G}(X_\epsilon)_{w,h}\| \quad (8)$$

This formulation encourages one to generate an adversarial slack image with infinitesimal and imperceptible changes to a adversarial sample.

To focus the learning effort of discriminator on the aspects that are most relevant to adversarial examples, we calculate the loss function of discriminator l_d as:

$$l_d = -\sum_{n=1}^{N} [\log D_{\theta_D}(X) + \log D_{\theta_D}(G_{\theta_G}(X_\epsilon))] \quad (9)$$

2.3. Training

The straightforward method to train the generator and the discriminator is update both in every batch. However, the discriminator network often learns much faster than the generator network because the generator is more complex than distinguishing between real samples and fake samples. Therefore, generator should be run twice with each iteration to make sure that the loss of discriminator does not go to zero.

The learning rate is initialized with 0.0002 and Adam[8] optimizer is used to update parameters and optimize the networks. The weights of the adversarial perturbation elimination specified loss ξ_1 , ξ_2 and ξ_3 used in the Eqn.5 are fixed to 0.1, 0.45 and 0.45 separately. The training procedure of the APE-GAN needs no knowledge of the architecture and parameters of the target model.

3. EXPERIMENTS

We now test our APE-G algorithm on deep convolutional neural networks architectures applied to MNIST[9], CIFAR10[10] and ImageNet[11] image classification datasets. We evaluate the proposed APE-G approach against state-of-the-art techniques to compute adversarial perturbations, including L-BFGS[3], DeepFool[12], JSMA[13], FGSM[13], CW[14]. We consider the following deep neural network architectures as the target models:

- **MNIST**: A LeNet-5-like architecture is used for MNIST digits recognition task[15]. We replaced LeNet-5's RBF layer with normal fully-connected layer, and deleted connection table which introduce sparsity between S2-C3 layer.
- **CIFAR10**: We trained a DenseNet-Fast-40[16] for CIFAR10 classification task.
- **ImageNet**: We used ResNet50[17] and Inception-V3[18] pre-trained models.

The adversarial examples are generated based on the target models seperately. We use the implementation code of FGSM and JSMA in cleverhans v2.0[19], L-BFGS and DeepFool in Foolbox[20], and the author's implementation code of CW [14]. All the algorithms are implementated with default parameter except parameter κ = 0 of CW-L_2 and parameter ϵ of FGSM. We set different noise scale parameter $\epsilon = 0.3$, $\epsilon = 0.1$ and $\epsilon = 8/255$ for MNIST, CIFAR10 and ImageNet to construct FGSM attack samples respectively.

We evaluated the classification error rate of adversarial examples generated by state-of-the-art attacks algorithms, quantitative results are summarized in Table 1. The experimental results indicate that APE-G is effective to resist adversarial examples generated from state-of-the-art attacking methods. The "Base" column gives the classification error rates of adversarial examples for each target model without denfense operations. The "APE-G" column is the classification error rates of adversarial examples processed by the proposed APE-G model before fed into the target model. The results show the Top-1 and Top-5 error rate on ImageNet. The first row report the error rates of the clean images as the baseline. With a 97.8% average misclassification rate for adversarial examples, the LeNet trained on MNIST is easily misled. The DenseNet, ResNet and Inception-V3(In-V3) are somewhat more robust to adversarial examples, but are still vulnerable to most of attack methods. The APE-G model can reduce the recognition error rates of LeNet and DenseNet by 87% and 64% repectively. The APE-G model can at least reduce the classification error rate of ResNet$_{ape}$ by a factor of two, especially for adversarial examples generated by CW. The amazing result of Top-5 score imply that the APE-G is at least able to move the correct class back to the top, even if the prediction is still incorrect. At the same time, we noticed

Table 1. Classification error rates (in %) of adversarial examples generated by various of attacking methods on MNIST, CIFAR10 and ImageNet datasets.

Attack	**LeNet**$_l$		**DenseNet**		**ResNet**$_{Top-1}$		**ResNet**$_{Top-5}$		**In-V3**$_{Top-1}$		**In-V3**$_{Top-5}$	
	Base	APE-G	Base	APE-G	Base	APE-G	Base	APE-G	Base	APE-G	Base	APE-G
Clean	0.8	1.2	9.9	10.3	24.4	25.2	7.2	7.8	21.2	22.7	5.4	5.8
L-BFGS	93.4	2.2	92.7	19.9	93.3	42.9	92.2	10.5	96.4	41.7	93.5	10.1
FGSM	96.3	2.8	77.8	26.4	72.9	40.1	36.6	13.2	72.2	38.0	35.7	11.0
DeepFool	97.1	2.2	98.3	19.2	98.4	45.9	96.3	11.7	98.8	48.2	96.6	12.4
JSMA	97.8	38.6	94.1	38.3	98.7	45.0	96.7	12.3	98.3	44.1	96.3	12.5
CW-L_0	100.0	27.0	100.0	46.9	100.0	29.4	93.0	10.5	99.6	27.8	98.4	10.6
CW-L_2	100.0	1.5	100.0	30.5	99.7	26.1	95.2	12.6	100.0	29.1	97.5	12.3
CW-L_∞	100.0	1.2	100.0	32.2	100.0	27.0	98.9	13.3	100.0	29.6	99.8	13.7

Table 2. Classification error rates (in %) of adversarial examples produced by FGSM on target models. α is the changed value of each pixel on each step[13].

		$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$
MNIST	LeNet	35.9	86.0	96.3	98.0
	SRGAN	36.4	82.7	97.1	95.4
	Adv. Train	1.6	3.4	6.6	59.8
	Iter Attack	32.0	82.5	84.7	94.5
	APE-G	0.8	1.1	2.8	21.0
	Iter Attack*	1.9	2.5	4.5	22.3
CIFAR10	DenseNet	77.8	84.7	86.3	87.2
	SRGAN	79.3	81.4	88.9	86.7
	Adv. Train	26.4	45.2	55.9	63.4
	Iter Attack	72.6	82.5	86.5	88.0
	APE-G	12.2	39.6	73.7	81.7
	Iter Attack*	16.5	38.3	51.7	71.0

that APE-G model has almost no efffect on accuracy on clean examples(difference in accuracy within 0.3%).

We compared our method with image denoise(SRGAN[7]) and adversarial training[21](fine-tuning using adversarial examples) on MNIST and CIFAR10, as shown in Table 2. Rows "LeNet" and "DenseNet" are the error rate of adversarial examples on the target model. Image denoise(SRGAN) method hardly brings any improvement in classification accuracy of adversarial examples. Aadversarial training do increase the robustness of neural networks for one-step attack (Row "Adv.Train") but would not help under iterative attacks(Row "Iter Attack"). Adversarial examples against the fine-tuned model can easily fool the deep neural networks("Iter Attack" column). This is maily because adversarial training is used for regularization only to avoid overfitting. The injection of adversarial examples during training to improve the generalization of the machine learning model bring no change to the high-dimensional linear properties of the deep model. The proposed APE-G method significantly increases the robustness of networks to adversarial perturbations even under iterative attacks("Iter Attack*" column). For example, the robustness of the networks on MNIST is dropped by 1.1% and CIFAR10's robustness is dropped by about 4.3% with $\alpha = 0.1$. The results demonstrate that: if APE-G is known, an adversary that targets the APE-G+model will not bring a significant drop in classification performance. Defensive distillation is not included in our comparision due to its limited helpness against adversarial examples[14].

The robust performance of our method can be contributed to the use of spatial constrain l_{sc} that is able to effectively eliminate the adversarial perturbations, but independent of perturbation scale. Experimental results also revealed that APE-G model can only effectively defense against most of adversarial examples. As shown in Table 1, the error rates of adversarial examples generated by FGSM is up to 38.6%. The accuracy against complex visual adversarial images(such as ImageNet) is also unsatisfactory. Due to the training instability of GAN, the nonsensical outputs of generator limits the performance of APE-G framework. In spite of that, APE-G model still significantly improved the robustness of the DNN models against adversarial examples.

4. CONCLUSION

In this paper, we proposed a novel idea of defending against adversarial examples via eliminate the trivial perturbations of the input data being highly aligned with the weight vectors of the models. Future work should focus on methods to eliminate the adversarial perturbation of complex samples.

5. REFERENCES

[1] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy, "Explaining and harnessing adversarial examples," *Computer Science*, 2014.

[2] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio, "Adversarial examples in the physical world," *arXiv: Computer Vision and Pattern Recognition*, 2016.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," *Computer Science*, 2013.

[4] Nicholas Carlini and David Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017, pp. 3–14.

[5] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Security and Privacy (SP), 2016 IEEE Symposium on*. IEEE, 2016, pp. 582–597.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[7] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew P Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al., "Photo-realistic single image super-resolution using a generative adversarial network," *computer vision and pattern recognition*, pp. 4681–4690, 2016.

[8] Diederik P Kingma and Jimmy Lei Ba, "Adam: A method for stochastic optimization," *international conference on learning representations*, 2015.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[10] Alex Krizhevsky and Geoffrey Hinton, "Learning multiple layers of features from tiny images," 2009.

[11] Jia Deng, Wei Dong, R. Socher, Li Jia Li, Kai Li, and Fei Fei Li, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 248–255.

[12] Seyedmohsen Moosavidezfooli, Alhussein Fawzi, and Pascal Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," *computer vision and pattern recognition*, pp. 2574–2582, 2016.

[13] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami, "The limitations of deep learning in adversarial settings," in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[14] Nicholas Carlini and David Wagner, "Towards evaluating the robustness of neural networks," in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 39–57.

[15] Yann LÉcun, Leon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[16] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Van Der Maaten Laurens, "Densely connected convolutional networks," in *Computer Vision and Pattern Recognition*, 2016.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[18] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[19] Ian Goodfellow Reuben Feinman Fartash Faghri Alexander Matyasko Karen Hambardzumyan Yi-Lin Juang Alexey Kurakin Ryan Sheatsley Abhibhav Garg Yen-Chen Lin Nicolas Papernot, Nicholas Carlini, "cleverhans v2.0.0: an adversarial machine learning library," *arXiv preprint arXiv:1610.00768*, 2017.

[20] Jonas Rauber, Wieland Brendel, and Matthias Bethge, "Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models," *arXiv preprint*, 2017.

[21] Nicolas Papernot, Fartash Faghri, and Nicholas Carlini etc., "Technical report on the cleverhans v2.1.0 adversarial examples library," *arXiv preprint arXiv:1610.00768*, 2018.