A Powerful Transferability Adversarial Examples Generation Method Based on Nesterov Momentum Optimization

Yunfang Chen, Qiangchun Liu, and Wei Zhang

***Abstract*—Researchers have found that we artificially add perturbation to the input image to generate adversarial examples which can cause the deep learning model to misclassify. The existing method of generating adversarial examples can achieve high white-box attack success rate, but the one of black-box attack is low. In order to improve the transferability ability of adversarial examples and obtain higher attack success rate, we apply the Nesterov momentum optimization method to the gradient-based adversarial examples generation method. Combined with the momentum and decay factor, the iterative gradient is optimized during the optimization process. This effectively escapes the local minima during the optimization process, resulting in faster iterations and better adversarial examples generation. The experiment showed that the white-box attack achieves 100% attack success rate, and the powerful transferability of the examples make the black-box attack success rate significantly higher than the original methods.**

***Index Terms*—Adversarial examples, attack success rate, powerful transferability, nesterov momentum optimization method.**

I. Introduction

Deep neural network performs well in some hard machine learning tasks, especially in the field of image recognition. Classification accuracy of deep neural network is comparable to that of humans [1], [2]. However, researchers found that existing deep neural networks are vulnerable to adversarial attack. Szegedy *et al.* [3] first proposed this concept of adversarial example in the field of image recognition. It adds small noise that is not easily noticeable to humans to clean image, which causes model misclassify.

White-box attack assumes that the attacker knows the complete knowledge of the target model, including its parameter values, architecture, training methods, and training data. In this case, there are many methods for generating adversarial examples for white-box attack, including one-step gradient method such as fast gradient sign method [4], basic iterative method [5], and universal perturbation [6]. These methods can achieve high white-box attack ability. Since different deep learning models obtain similar decision boundaries [7] after pre-training using the data in the same sample space, the adversarial examples generated by the white-box attack have the transferability ability [8]-[10]. That means adversarial examples generated for the specific model also misclassify other models with the same classification task. The transferability of adversarial examples makes it perform black-box attack on other unknown models. However, the black-box attack capabilities of these methods are often limited.

The existing black-box [11]-[13] attack success rate of adversarial examples is very low. First, the attacker does not know the parameters and structure of the target model, which brings certain difficulties to the attacker. In addition, for models with special defense mechanisms, ensemble adversarial training [14] can significantly improve the robustness of deep neural networks, so most of the attack methods have a low success rate for black-box attack. In particular, the transferability ability to adversarial examples based on simple iterative methods [5] is not high, so this method's black-box attack is very inefficient. In view of the difficulties in the practical application of black-box attacks, we have been exploring more efficient black-box attack method from an optimization [15] perspective.

In this paper, we study the adversarial examples generation method based on Nesterov momentum optimization, which is a gradient-based momentum iterative optimization algorithm. Our method aims to generate adversarial examples with powerful transferability ability because of breaking the limit of local extreme value. This method brings a significant improvement in the attack ability and we will show that it has a higher success rate in both white-box and black-box attacks. Our contributions are as follows:

1) We apply the gradient-based Nesterov optimization method to the adversarial examples generation method, which breaks through the optimization extreme value, and generate powerful transferability adversarial examples.

2) The decay factor plays a key role in the momentum optimization algorithm and we determine the optimal of it through experiments.

3) We compare the attack success rate of adversarial examples against four test models to demonstrate the advantage of our method for improving the transferability ability.

II. Background

A. Attack Description

Given a classifier $f(x): x \in X \to y \in Y$ that outputs a

Manuscript received May 9, 2019; revised February 11, 2020.

The authors are with the Department of Computer Science, Nanjing University of Posts and Telecommunications, China (e-mail: chenyf@njupt.edu.cn, 1217043203@njupt.edu.cn, zhangw@njupt.edu.cn).

doi: 10.18178/ijmlc.2020.10.3.953

label y as the prediction for an input x, the goal of adversarial attacks is to seek an example x^* in the vicinity of x but is misclassified by the classifier. For a correctly classified input x with ground-truth label y such that $f(x)=y$, an adversarial example x^* is crafted by adding small noise to x without changing the ground-truth label, but misleads the classifier as $f(x^*) \neq y$. In most cases, the L_p norm of the adversarial noise is required to be less than an allowed value ε as $\left\|x^* - x\right\|_p \leq \varepsilon$, where p could be 0, 1, 2, ∞.

B. Traditional Gradient-Based Adversarial Attack Methods

In this section, we express two more classic gradient-based adversarial attack methods: minimum perturbation method (Deepfool)[16] and fast gradient sign method (FGSM)[4]. These two methods are used to generate adversarial examples in conjunction with our Nesterov momentum optimization.

Deepfool search for the closest distance from the clean image to the decision of the targeted classifier, so it attempts to perturb the data point to cross the decision boundary of the model to change the model classification results. Deepfool iteratively find the minimum perturbation by using the objective function gradient. The perturbation vector is defined as follows:

$$\rho = -\frac{f(x_t)}{\left\|\nabla f(x_t)\right\|_1}\nabla f(x_t) \tag{1}$$

The perturbation vector is iteratively acquired and is added to the clean images until the model is misclassified. The obtained adversarial example can be expressed as

$$x_0^* = x \qquad x_{t+1}^* = x_t^* + \rho \tag{2}$$

For a given target input image, Goodfellow *et al*. proposed FGSM which is a method for quickly calculating adversarial perturbation along the gradient direction with respect to the input image. The adversarial perturbation is defined as follows:

$$\rho = \varepsilon sign(\nabla J_x(x, y)) \tag{3}$$

In order to limit the amplitude of the perturbation, the fast gradient sign method must meet the condition $\left\|x^* - x\right\|_p \leq \varepsilon$ The adversarial example can be expressed as follows:

$$x^* = x + \varepsilon sign(\nabla J_x(x, y)) \tag{4}$$

where $\nabla J_x(x, y)$ represents the gradient of the loss function calculated under the model parameter θ. $sign(.)$ defines the sign function. ε is a small value and limits the perturbation amplitude.

III. Proposed Method

A. An Framework of Generation Adversarial Example

Fig. 1 shows the method of generation adversarial example in our framework. Under the condition that the white-box model and the clean images are known, the original attack methods generate gradient-based adversarial examples by solving the constrained optimization. In the process of solving an optimization problem, the adversarial perturbation is iteratively searched.

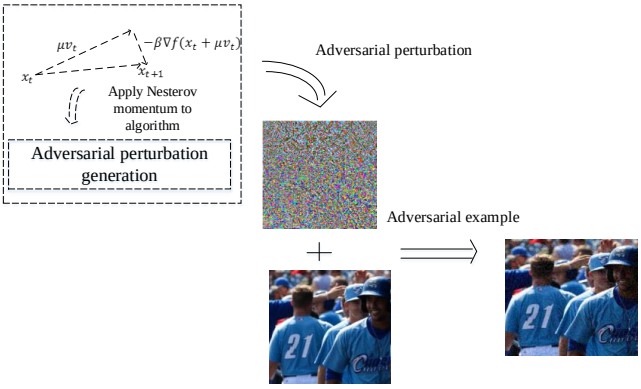

Fig. 1. An framework of generation adversarial example.

In our framework, we apply the Nesterov momentum optimization method to attack method and iteratively search for gradient-based adversarial perturbation instead of gradient descent. This can escape the local extremum during the optimization process, and improve the transferability ability of adversarial examples to achieve higher attack success rate. The adversarial example generated based on our method still maintain ground-truth label, which is completely meet human sensory perception. Therefore, our method based on optimized perspective can guarantee attack rate and semantic inconvenience.

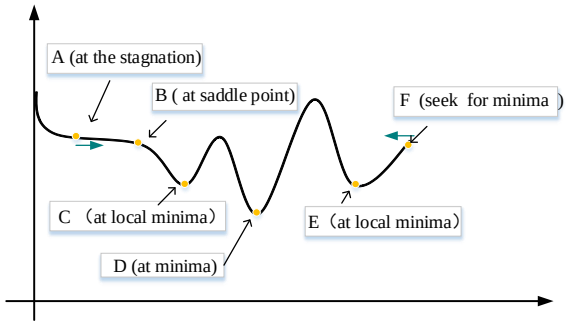

Fig. 2. Different areas of the optimization process.

B. Nesterov Momentum Optimization

In the process of solving the optimization problem, there are two problems encountered in the gradient descent process. First, in the region with small gradient, the iterative speed of the algorithm is particularly slow, and it falls into a state of stagnation like point A in Fig. 2. Second, the C point or the E point is located where the gradient is 0, then the algorithm pauses the iteration here. The only local minima point is obtained instead of the desired global minimal. In addition to the extreme value, there is a very common situation in the optimization process called the saddle point such as the B point. Although the saddle point has a gradient of 0, it is neither a local maximum nor a local minimum.

The problems encountered in the gradient descent algorithm can be solved by applying Nesterov momentum

optimization. The Nesterov momentum optimization method can help the algorithm iterative process to rush out of extreme value or saddle point and reach desired global minimal. We show this method as follows.

Given objective function $f(x)$, initial point x_0 and initial momentum v_0. Repeat the iteration until the stop criterion is reached:

$$\Delta x_t = -\nabla f(x_t + \mu v_t) \tag{5}$$

$$v_{t+1} = \mu v_t + \beta \Delta x_t \tag{6}$$

$$x_{t+1} = x_t + v_{t+1} \tag{7}$$

On the basis of the gradient descent, the last moment of momentum v is retained, and each iteration is multiplied by decay factor μ which affects the magnitude of the momentum term. In general, each iteration update consists of two parts: one is the gradient direction which is not the current position, but a further position along the current momentum term and the other part is momentum term. Under the influence of the momentum term, the algorithm iteration is equivalent to having the inertia, so that it helps algorithm pass through the extreme value or the stagnant zone.

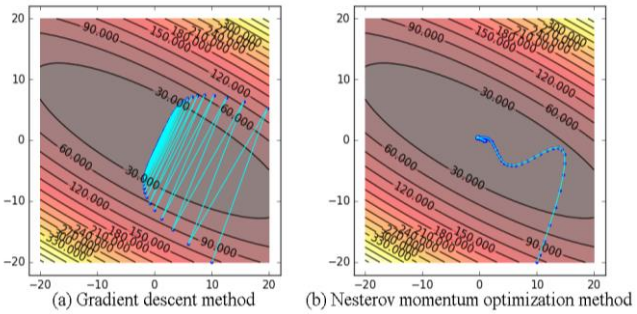
(a) Gradient descent method (b) Nesterov momentum optimization method

Fig. 3. Function optimization process.

C. Comparison between Gradient Descent Algorithm and Nesterov Momentum optimization

As showed in Fig. 3, we respectively use the gradient descent algorithm and the Nesterov momentum optimization method to solve convex optimization problem. The contour line in the Fig. 3(a) shows that the gradient descent algorithm is oscillating in an iterative process, and this wave-like process of going to the valley undoubtedly weakens efficiency and energy of the iteration. So this iterative process is not easy to achieve optimal. From the Fig. 3(b), the Nesterov momentum optimization algorithm is applied to the algorithm iterative process to break through the dilemma and obtain faster convergence performance. From the starting point, the closest gradient optimization direction is quickly obtained, and the canyon is passed to the lowest point of the valley instead of complex iterative process. The Nesterov momentum algorithm can also pre-determine its update direction during the gradient descent, making the gradient descent direction more stable.

D. Applying Nesterov Momentum to Deepfool and FGSM

The Nesterov momentum algorithm is an accelerated gradient descent technique that allows the algorithm to accumulate velocity vectors in the direction of the gradient during the iterative process, which helps to cross the stagnation zone and the local extremum zone. We applied the Nesterov momentum method to two gradient-based adversarial examples generation methods and found that the effects are very prominent.

To generate an adversarial example x^* from a clean example x which satisfies the L_∞ norm bound, gradient-based approaches seek the adversarial example by solving the constrained optimization problem:

$$\arg\min_{x^*} J(f(x^*), y), \quad s.t. \ \|x^* - x\|_\infty \le \varepsilon \tag{8}$$

where ε is the size of adversarial perturbation. Deepfool generates adversarial examples by iteratively find adversarial perturbation in equation (1). It interferes with the decision boundaries of the model to misclassify. However, the poor optimization makes it fall into local extremum during the process of crafting adversarial examples. It limits the attack ability and transferability ability of adversarial examples. To break this limitation, we apply Nesterov momentum to the Deepfool algorithm so that the algorithm can escape local extremum and maintain a stable gradient update direction during the iteration process.

The following is the pseudo code of Nesterov Momentum-Deepfool:

Algoithm 1 Nesterov Momentum-Deepfool (NM-Deepfool)

Input: A classifier f with loss function J; **a clean image** x **and ground-truth label** y; size of perturbation ε; iterations T and decay factor μ.

Output: An adversarial example x^* with $\|x^* - x\|_\infty \le \varepsilon$

1: $\alpha = \varepsilon / T$

2: $v_0 = 0$; $x_0^* = x$

3: for t = 0 to $T - 1$ do

4: Input x_t^* to f and obtain the gradient $\nabla f(x_t)$

5: Update v_{t+1} **by accumulating the velocity** vector in the gradient direction as

$$v_{t+1} = \mu v_t - \frac{f(x_t)}{\|\nabla f(x_t)\|_1} \nabla f(x_t) \tag{9}$$

6: Update x_{t+1}^* as

$$x_{t+1}^* = x_t^* + \beta v_{t+1} \tag{10}$$

7: **end for**

8: **return** $x^* = x_T^*$

Equation (9) means the momentum accumulated by the first t-step iteration under the effect of the decay factor μ. After the T-step iteration which each step is α, the minimum adversarial perturbation is found and the adversarial example x^* is generated. In each iteration, the current gradient $\nabla f(x_t)$ is normalized to the L_1 distance

because we find that the scale of the gradients in different iterations varies in magnitude.

Our method is a framework that can be combined with different gradient-based perturbation generation method. So, we combine Nesterov momentum optimization method with FGSM and change the cumulative way of the momentum on the gradient in equation (9) to Equation (11).

$$v_{t+1} = \mu v_t - \varepsilon \frac{\nabla J_x(x, y)}{\left\| \nabla J_x(x, y) \right\|_1} \tag{11}$$

FGSM based on Nesterov momentum(NM-FGSM) is formed after multiple iterations (12).

$$x_{t+1}^* = x_t^* + sign(\beta v_{t+1}) \tag{12}$$

IV. Experiments

A. Setup

We selected four models Inception v3 (Inc-v3) [17], Inception v4 (Inc-v4), Inception Resnet v2 (IncRes-v2) [18], Resnet v2-152 (Res-152) [19] to study the attack success rate against white-model and black-model.

It is less meaningful to study the transferability ability of adversarial examples if the models cannot classify the original image correctly. Therefore, we randomly selected 1000 images from the ILSVRC 2012 validation set [20], which can be correctly classified by all 4 models in our inspection. These 1000 images make up our test set.

In our experiments, we first verified the white-box attack ability and black-box attack ability of the FGSM method and the Deeofool method. Then we apply the Nesterov momentum algorithm to the iterative process to examine the transferability ability of our methods. In order to limit the gap between the adversarial example and the original example, we use the L_∞ norm bound to limit the amplitude of adversarial examples.

B. Comparison of Original Method and Our method in Attack Success Rate

From the Table I, the adversarial examples are generated for trained model Inc-v3, Inc-v4, InvRes-v2 and Res-152 respectively using Deepfool and FGSM. The perturbation amplitude ε of the pixel is set to 20 and the pixel value is limited to the interval [0, 255].The success rates are the misclassification rates of the corresponding model with adversarial images as inputs. From the table, we can observe that Deepfool and FGSM remain as strong white-box adversary since White-box attack success rate is around 90%. The adversarial examples are crafted for trained model Inc-v3 using FGSM which achieve White-box attack success rates of 89%. The adversarial examples are crafted for trained model Inc-v3 using Deepfool which achieve White-box attack success rates of 93%.

However, it is worth noting that FGSM and Deepfool are less powerful in attacking the black-box model. The adversarial examples are generated for the trained model Inc-v3 using FGSM and Deepfool methods and attack against Inc-v4. The attack success rates only reach 27.2% and 41.3% respectively. This means that the adversarial examples generated by the traditional FGSM and Deepfool show insufficient performance in the model transferability ability, resulting in weak black-box attack ability. Lower black-box attack success rate is not only reflected on test model Inc-v4, but also on InvRes-v2 and Res-152.

TABLE I: The Success Rates(%) of Adversarial Attacks Test Model. The Adversarial Examples Are Crafted for Trained Model Using FGSM and Deepfool. * Ndicates the White-Box Attacks

Test Model / Trained Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152
Inc-v3	FGSM	**89***	27.2	19.3	26.7
	Deepfool	**93***	41.3	39.2	40.4
Inc-v4	FGSM	41.8	**83.4***	24.8	30.6
	Deepfool	36.8	**91.7***	32.5	34.3
IncRes-v2	FGSM	42.6	35.3	**87***	40.2
	Deepfool	50.8	49.1	**93.4***	47.6
Res-152	FGSM	42	44.8	43.7	**85.2***
	Deepfool	48.3	49.1	46.2	**91.6***

We improve the iterative ability of the algorithm from the perspective of optimization and break through the extreme value region in the iterative process. Therefore, we apply Nesterov momentum optimization to attack method to investigate the advantages of our methods in black-box attacks. The perturbation amplitude ε of the pixel is the same as the original method. The decay factor μ is set to 1.0, and excessive learning rate β may result in non-convergence, so the recommended value is 0.1. We test attack success rate on Inc-v3, Inc-v4, InvRes-v2 and Res-152 again.

Table II shows the success rate of adversarial attacks. It can be found that NM-FGSM and NM-Deepfool methods have greatly improved both white-box attacks and black-box attacks. The adversarial examples are crafted for trained model Inc-v3 using NM-Deepfool and NM-FGSM achieve White-box attack success rates of 100%. The adversarial examples are generated for the trained model Inc-v3 using NM-FGSM and attack against Inc-v4. The black-box attack success rate reaches 52.4%, which is nearly twice as powerful as FGSM. The adversarial examples are generated for the trained model Inc-v3 using NM-Deepfool and attack against Inc-v4. The black-box attack success rate reaches 64.7%, which is 23.4% high than Deepfool method. Compared with the attack success rate of black-box model in Table I, our approach has also been greatly improved on other test models.

Therefore, it is effective to improve the attack success rate of adversarial examples from the perspective of optimization.

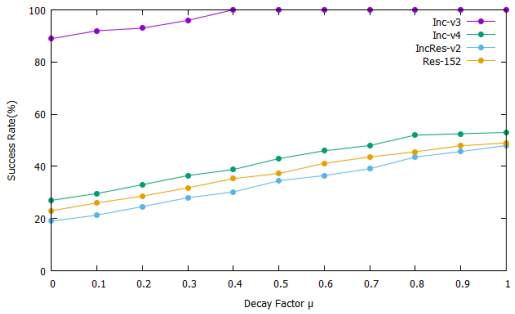

Fig. 4. The adversarial examples generated for Inc-v3 by NM-FGSM against Inc-v3(white-box), Inc-v4, IncRes-v2, Res-152(black-box),with μ ranging from 0.0 to 1.0.

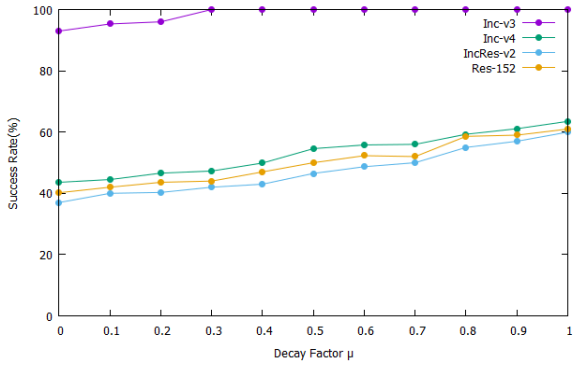

Fig. 5. The adversarial examples generated for Inc-v3 by NM-Deepfool against Inc-v3(white-box), Inc-v4, IncRes-v2, Res-152(black-box),with μ ranging from 0.0 to 1.0.

C. Influence of Decay Factor μ in Nesterov Momentum

The decay factor μ plays an important role in the momentum optimization algorithm, which greatly affect the attack success rate of adversarial examples. If $\mu = 0$ both NM-Deepfool and NM-FGSM based on Nesterov momentum turn to Deepfool and FGSM. Therefore, we study the appropriate value of the decay factor. The adversarial examples are generated for Inc-v3 using NM-FGSM and NM-Deepfool with perturbation $\varepsilon = 20$ and the decay factor μ is ranging from 0.0 to 1.0 with a granularity 0.1.

When $\mu = 0$, it means that there is no momentum retained during the iteration. Therefore, the attack success rate of NM-FGSM and NM-Deepfool shows as the same as original method. As the decay factor increases, the attack success rate is also improved. When $\mu = 1$, the maximized momentum accumulated in the gradient direction is retained in the last iteration , which makes it have the greatest impact on the gradient of this iteration. As can be seen from Fig. 4 and Fig. 5, the white-box success rate of adversarial examples can be increased to 100% with the increase of μ. The transferability of adversarial example is also enhanced, so the success rate of the black-box attack is also increasing. The attack success rate reaches a peak with decay factor when $\mu = 1$. We can see from Fig. 4 and Fig. 5 that the adversarial examples generated for Inc-v3 by NM-FGSM and NM-Deepfool achieve more than 40% attack rate when $\mu = 1$. It is a very big boost for black-box attacks.

TABLE II: THE SUCCESS RATES(%) OF ADVERSARIAL ATTACKS TEST MODEL. THE ADVERSARIAL EXAMPLES ARE CRAFTED FOR TRAINED MODEL USING NM-FGSM AND NM-DEEPFOOL. *INDICATES THE WHITE-BOX ATTACKS

Test Model / Trained Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-152
Inc-v3	NM-FGSM	**100***	52.4	49.4	47.3
	NM-Deepfool	**100***	64.7	63.6	61
Inc-v4	NM-FGSM	67	**100***	54.6	51.8
	NM-Deepfool	70.2	**100***	69.2	67.4
IncRes-v2	NM-FGSM	65.2	64.9	**100***	60.3
	NM-Deepfool	71.3	70.9	**100***	69.2
Res-152	NM-FGSM	66.2	69.1	63.4	**100***
	NM-Deepfool	69.4	70.2	68.5	**100***

V. CONCLUSION

Deep neural networks are vulnerable to adversarial examples, which cause white-box attacks and black-box attacks because of their transferability ability. In this paper, adversarial examples with powerful transferability ability are generated because the Nesterov optimization momentum is applied to the algorithm iterative process to find the perturbation. Experiments show that the adversarial examples generated by our method achieve a higher attack success rate than the traditional Deepfool and FGSM methods in both white-box attack and black-box attacks. The decay factor plays an important role in improving the attack success rate and we have studied its best value.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

Yunfang Chen conducted the research; Qiangchun Liu analyzed the data; Qiangchun Liu and Wei Zhang wrote the paper. All authors had approved the final version.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems*. Lake Tahoe, Nevada, USA, 2012, pp. 1097-1105.

[2] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.

[3] C. Szegedy, W. Zaremba, I. Sutskever, and J. Bruna, "Intriguing properties of neural networks," in *Proc. the 2nd International Conference on Learning Representations*. Banff, AB, Canada, 2014.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. the 3rd International Conference on Learning Representations*, San Diego, CA, USA, 2015.

[5] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. the 5th International Conference on Learning Representations*, Toulon, France, 2016.
[6] S. M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. 2017 IEEE Conference on Computer Vision and Pattern Recognition*. Honolulu, HI, USA, 2017, pp. 1765-1773.
[7] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proc. 5th International Conference on Learning Representations*, Toulon, France, 2016.
[8] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *Proc. 2016 International Joint Conference on Neural Network*, 2016, pp. 426-433.
[9] H. Hosseini, Y. Chen, S. Kannan, B. Zhang, and R. Poovendran, "Blocking transferability of adversarial examples in black-box learning systems," *arXiv preprint arXiv:1703.04318*, 2017.
[10] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.
[11] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. the 2017 ACM on Asia conference on Computer and Communications Security*, 2017, pp. 506-519.
[12] J. Hayes and G. Danezis, "Machine learning as an adversarial service: Learning black-box adversarial examples," *arXiv preprint arXiv:1708.05207*, 2017.
[13] P. Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C. J. Hsieh, "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 15-26.
[14] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *Proc. 6th International Conference on Learning Representations*, Vancouver, BC, Canada, 2018.
[15] W. Duch and J. K. orczak, "Optimization and global minimization methods suitable for neural networks," *Neural Computing Surveys*, pp. 163-212, 1998.
[16] S. M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2574-2582.
[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, 2016, pp. 2818-2826.
[18] C. S. zegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning," in *Proc. the 31th AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, pp. 4278-4284.
[19] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. European Conference on Computer Vision*, 2016, pp. 630-645.
[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211-252, 2015.

Yunfang Chen received the Dr. Eng degree in computer science from Soochow University, Suzhou, China, in 2008. He is an associate professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include big data processing and machine learning.

Qiangchun Liu received the B.S. degree in information engineering from Tongda College of Nanjing University of Posts and Telecommunications, Yangzhou, China, in 2017. He is currently pursuing the M.S. degree in software engineering at NUPT. His research interests include image classification, differential privacy convolutional neural network, and its applications in deep learning.

Wei Zhang received the Dr. Eng degree in computer science from Soochow University, Suzhou, China, in 2008. From 2008 to 2011, he was a postdoctoral research fellow in Nanjing University of Posts and Telecommunications. From March to September 2016, he was a visiting scholar at Purdue University. Since August 2013, he has been a professor with the School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include computer version and machine learning.