Convolutional Neural Network Visualization in Adversarial Example Attack

Chenshuo Yu, Xiuli Wang(✉), and Yang Li

School of Information, Central University of Finance and Economics, Beijing 100081, China
wangxiuli@cufe.edu.cn

Abstract. In deep learning, repeated convolution and pooling processes help to learn image features, but complex nonlinear operations make deep learning models difficult for users to understand. Adversarial example attack is a unique form of attack in deep learning. The attacker attacks the model by applying invisible changes to the picture, affecting the results of the model judgment. In this paper, a research is implemented on the adversarial example attack and neural network interpretability. The neural network interpretability research is believed to have considerable potential in resisting adversarial examples. It helped understand how the adversarial examples induce the neural network to make a wrong judgment and identify adversarial examples in the test set. The corresponding algorithm was designed and the image recognition model was built based on the ImageNet training set. And then the adversarial-example generation algorithm and the neural network visualization algorithm were designed to determine the model learning heat map of the original example and the adversarial-example. The results show that it develops the application of neural network interpretability in the field of resisting adversarial-example attacks.

Keywords: Adversarial-example · CNN · Visualization · Interpretability

1 Introduction

As an important content of the current computer industry, deep learning has gradually penetrated into the operation and development of various industries with the promotion of Internet plus. While deep learning is continuously advancing industrial form innovation and improving production efficiency, the internal ignorance of neural network models has also restricted the application of this technology, so it is imminent to promote research on the interpretability of neural networks. In addition, adversarial example attacks, as an important means of attack in image recognition, have a significant impact on deep learning. This paper applies the neural network visualization method to defend against adversarial example attacks, which not only enhances the credibility of the neural network visualization method, but also helps people to identify adversarial examples and make it clearer how the adversarial examples mislead the model [1, 2].

This work is supported by National Defense Science and Technology Innovation Special Zone Project (No. 18-163-11-ZT-002-045-04).

J. Zeng et al. (Eds.): ICPCSEE 2020, CCIS 1257, pp. 247–258, 2020.
https://doi.org/10.1007/978-981-15-7981-3_16

This article is different from the existing ideas of defending adversarial example: optimizing the network structure, training the adversarial examples, taking the optimized training examples as the main goal, it studies the possibility of defending adversarial examples from a new perspective, and strive to reduce the threat of adversarial example attack from the source.

2 Related Research

With the strong application ability of deep learning in various fields, the weak interpretability of deep neural networks has limited its application in the pillar field [3]. Therefore, in recent years, deep learning researchers, continuously improving deep learning capabilities, have begun to turn their attention to improving the interpretability of models. Well-known Internet companies such as Google and Alibaba are working hard to promote the development of AI interpretability and enhance the performance of human-machine dialogue. This move aims to clearly explain the decision-making or prediction behavior of Deep Neural Networks (DNN), and significantly improve model users' trust in the network, thereby reducing the potential hidden dangers caused by the use of deep learning technology in sensitive industries [4, 5].

Summarizing so many related work, it can be seen that there are three main types of ideas for indirect interpretation of CNN [6]: First, the hidden layer neuron feature analysis method. This method analyzes the hidden layer neurons in CNN to verify the behavior of the neural network, and collects the characteristics which can stimulate neurons to generate some brand-new images that can activate the entire network, as well as analyzing the features of the model. However, the obtained activation image is pretty abstract and not interpretable enough. The second is to imitate the model method. With the strong readability of the linear. An interpretable model is trained to simulate the input and output of the CNN. However, the model is heavily influenced by subjective factors; Third, the local linear replacement method. It selects a set of neighboring examples from the training set, and train a linear model to fit the original model according to the imitator model method. Local replacement reduces the huge error of the overall substitution, and can better grasp the characteristics of neurons inside the neural network. Correspondingly, this method also has the disadvantages of the above two methods.

The adversarial example attack, as a unique attack form in deep learning, is the key target of the robustness of deep learning. To defend adversarial example attacks, there are three main ideas. First, we can modify the training network, adjust the parameters of each convolutional layer and pooling layer, improve the redundancy of the model, and enhance the model's resistance to the adversarial examples. Correspondingly the recognition accuracy drops. And the structural reconstruction of the existing network model requires a lot of manpower and material resources, which is unrealistic in many cases; the second is to conduct adversarial example training [7], using the existing model to train the adversarial examples, so that the model can identify the characteristics of the adversarial examples, that is, we use the adversarial examples as the original training examples to train an adversarial example classification model. But this method also has its drawbacks. Obviously sorting out a large number of homogeneous

adversarial examples requires a lot of effort, and the error interference of adversarial examples can be irregular. Thus, the accuracy of the obtained model needs to be verified. Third, we can optimize the training examples, such as strengthening the selection of training examples and searching the adversarial examples in the example set. This method also requires a lot of energy to distinguish the correct examples from the adversarial examples, and essentially avoid the occurrence of adversarial example attacks [8].

Different from mainstream adversarial example defense schemes, this paper proposes an adversarial example identification scheme based on a convolutional neural network visualization algorithm [9–11], which applies heat maps to example recognition to improve interpretability and effective resistance to adversarial example attacks.

3 Algorithm Principle

3.1 Implementation Principle of Adversarial Examples

As shown in Fig. 1, in an animal classification task, the example X is to be identified as the correct animal classification, and there are two classification processes. f_1 is a classifier based on deep learning. After selecting the digital features of the input image and discriminating them, the classification result is $Y = cat$. f_2 is a discrimination mechanism based on naked eye recognition. Participants read features of the image through the naked eye to identify salient features such as hair, eyes, and ears of cats. The final classification result is $Y = cat$. When the model recognition is correct, the recognition results of the machine and the user are the same, that is, $f_1(X) = f_2(X)$.

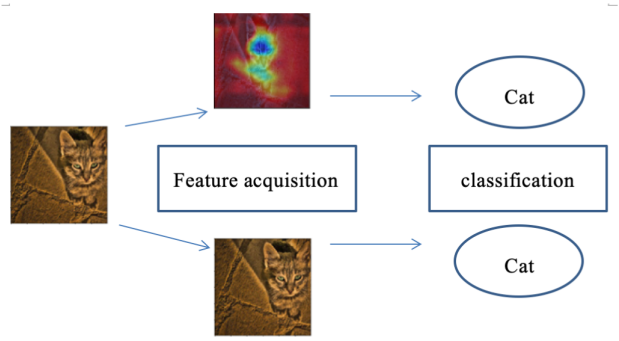

Fig. 1. Image identification

What the adversarial sample attack does is to increase the numerical interference limited to a threshold based on the original image X to obtain the adversarial example X', which makes the naked eye recognize the original image, and the machine distinguishes it into a completely new misclassification. Its mathematical meaning is:

$$\begin{aligned} Confirm\, X\prime \quad s.t\, \Delta(X, X\prime) < \varepsilon \\ f_2(X) = f_2(X\prime) \\ f_1(X) \neq f_1(X\prime) \end{aligned}$$

This paper uses the gradient descent method to generate the adversarial examples. Under the conditions of minimizing example disturbance and maximizing the model's misleading effect, the optimization is based on the direction of the maximum gradient, which guarantees the basic work of the subsequent visualization CNN processing. The ImageNet project is a large-scale image database for computer vision research. Over ten million image samples have been sorted, classified, and annotated. It is one of the most comprehensive image recognition resource pools. In order to expand the content range of the adversarial samples, not limited to the common training sets like cats and dogs, this article selects ImageNet as the sample object of the algorithm.

Models based on the ImageNet have been pre-trained in the Slim image classification library in the deep learning framework Tensorflow. You can use ready-made models with simple function calls and obtain model parameters and weights [12]. The code is shown below:

$$\begin{aligned} saver = tf.train.Saver(restore\_vars) \\ saver.restore(sess, os.path.join(data\_dir,' inception\_v3.ckpt')) \end{aligned}$$

In order to make the model universal, the algorithm supports processing all JPG format images. After reading the images according to the URL link, the data is processed to the image size required by the program with the help of the resize function. After obtaining the standardized image data, the Inception V3 model is used to perform initial output of the image, display the original image, and output the correct classification result for comparison with subsequent generated adversarial examples.

For a given image sample, the model will give the confidence of the Top-N term classification, that is, the corresponding probability $P(Y|x = X)$. In the process of generating adversarial examples, the algorithm aims to get a new image sample X', so that during the visual feature extraction process we get $f(X) = f(X')$, but it is judged as the confidence of the new category Y′. The highest degree, that is, the value of $P(Y'|x = X)$ is the largest.

In order to get adversarial examples, we set a threshold value and optimize sample constraints with back propagation and gradient descent. Iterate according to the following formula to maximize the confidence of the target classification of the adversarial examples and ensure that the visual characteristics of adversarial examples and the original images are within a certain range, and an adversarial example can be obtained.

$$f(X\prime) = X + \alpha \times \nabla \log P(Y\prime|x = X\prime)\, X\prime \in (X - \varepsilon, X + \varepsilon)$$

3.2 Convolutional Neural Network Visualization

For the complex structure of the neural network model, the most obvious step is to write a model structure diagram, analyze each convolutional layer, pooling layer, fully connected layer and other structures in the network, and record the parameters of each layer in detail. In the deep learning architecture keras, the constructed model architecture can be easily obtained through the function instructions of model. summary (). A part of network structure of the VGG16 model is shown in Fig. 2.

Layer (type)	Output Shape	Param #
input_3 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0

Fig. 2. VGG16 model structure diagram

Class Activation Mapping Select the output of the last convolutional layer block_conv3 before the Softmax function [13]. This is because the data loses the spatial information after passing the Softmax function, and it is impossible to obtain the identification situation of each part. Based on the traditional CNN model, CAM uses the Global Average Pooling layer instead of the original fully connected layer. While retaining the ability of the fully connected layer to reduce the data dimension, it also significantly reduces the parameters of CNN models (one-to-one correspondence in the fully connected layer makes the parameters in the network extremely huge), preventing the original model from overfitting.

The value of each feature map processed by the global average pooling layer is multiplied with the weight of each feature map and summed to obtain the heat map. Based on the CAM algorithm, it can be optimized by combining the ideas of back

propagation and deconvolution to obtain a weighted class activation mapping method (Grad-CAM) [14, 15, 17], which avoids the reconstruction and training of the existing network. Using the global average of the gradient instead of the weight of each feature map, we can obtain heat maps in the same way, as shown in Fig. 3. Whether the weight got from each feature map directly is equivalent to the weight calculated with the global average gradient has been demonstrated in detail in the paper [16].

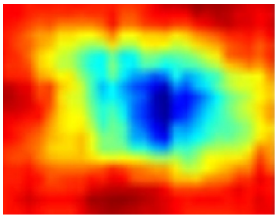

Fig. 3. Heat map

Based on the existing ordinary model, we get the output loss of the image label classification, and get the output of the last convolutional layer through the get_layer () function. The program can automatically calculate the average gradient by using the mathematical method of reverse automatic differentiation and taking the model output loss and the convolution layer output as parameters. Because it is complicated to manually derive the derivative expression of each parameter and then implement the code, you can use the gradients function encapsulated in Tensorflow to automatically calculate the gradient tensor and calculate the average value of the global gradient. Then we multiply the average gradient of all feature maps with the output of the global average pooling layer to get the gradient weight map. In order to enhance the visualization effect, we improve the color contrast effect, and a heat map is obtained to realize the visualization of the picture example classification model.

4 Algorithm Operation Effect Analysis

4.1 Visualization of Correct Sample Recognition

To ensure the effectiveness and universality of the algorithm, pictures are randomly selected from the Internet. Taking a common cat as an example, as shown in Fig. 4 (a). Needless to say, this is a test picture of a Egyptian cat (available from ImageNet tags). After the model discriminates, it can be known that the system has obtained the top three results of confidence: Egyptian cat, tabby cat and tiger cat, and the results predicted by the model are accurate. At the same time, several samples were selected for repeated experiments, as shown in Figs. 4 (b), 4 (c), and 4 (d). The principle is the same as that of cat samples.

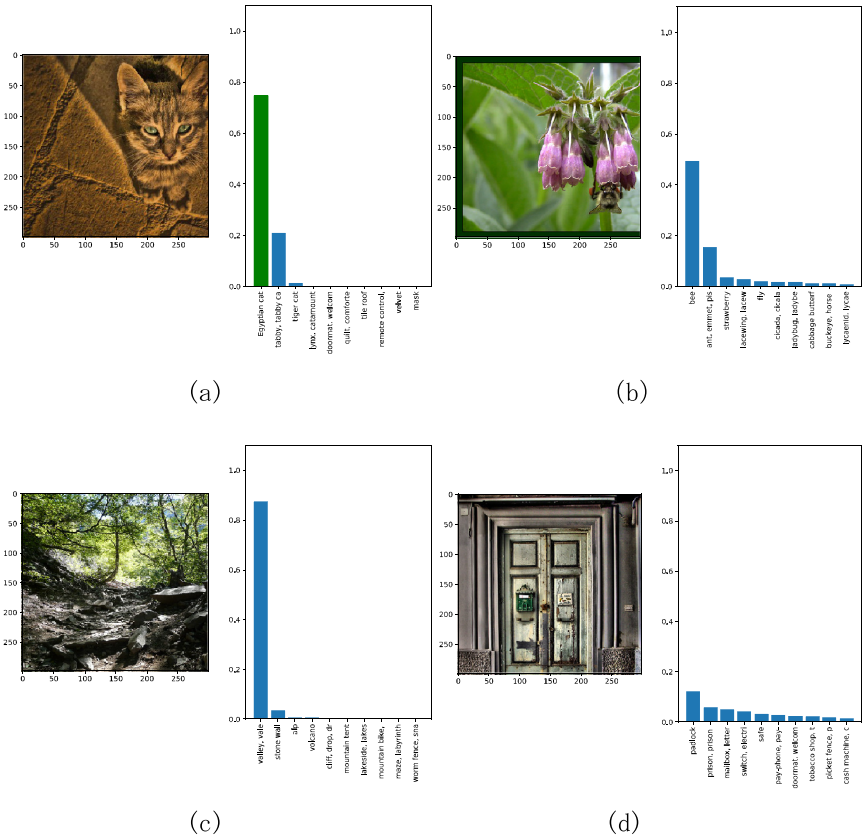

Fig. 4. Initial examples and classification results

With the implementation of the adversarial example generation algorithm, the target of the adversarial example is set as the 620th type in the ImageNet dataset—the PDA. After continuous optimization of projection and gradient descent, an adversarial example generated based on Fig. 4 can be obtained, as shown in Fig. 5.

Comparing the picture samples in Fig. 4 and Fig. 5, it can be clearly seen that the newly generated adversarial examples are consistent with the original pictures in visual effects and cannot be distinguished. But using the trained classification model to identify the adversarial examples, we can get that the images are recognized as handheld computers with a confidence level of almost 100%. Such a comparison result means that the two pictures that cannot be discerned by the naked eye are very different for CNN, which also proves the realization of the adversarial example attack. While the attack effect is good, the model further analyzes how the model recognizes the two picture samples as different objects. Therefore, the visualization algorithm is used to help the user understand which parts of the picture are recognized by the model. First,

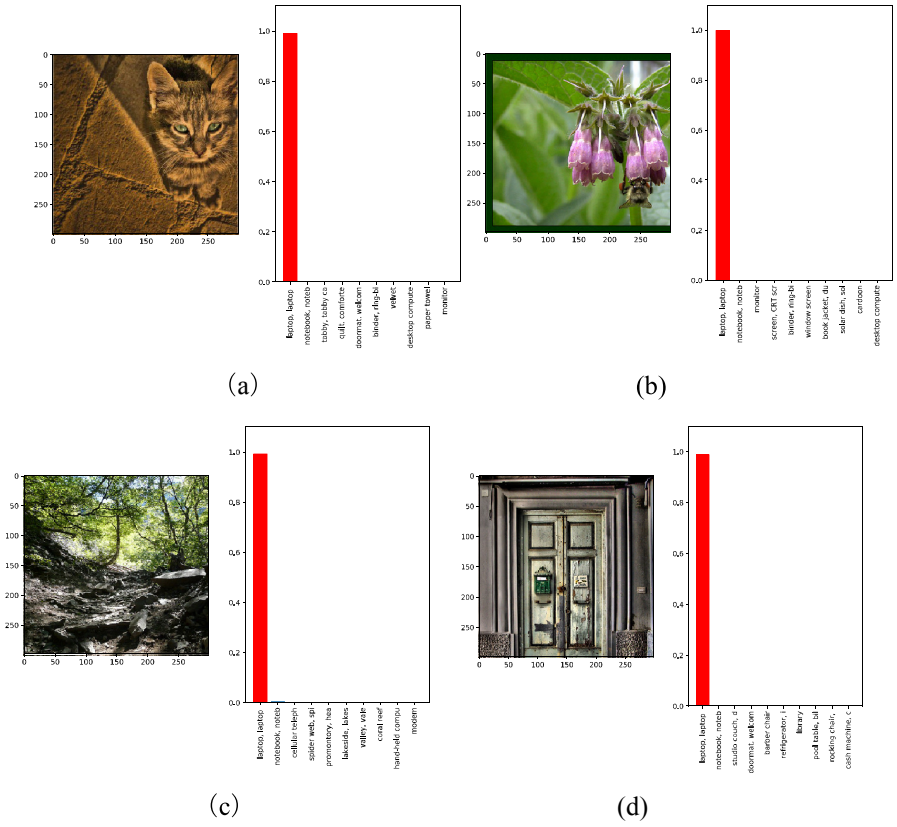

Fig. 5. Adversarial examples and classification results

the original picture is identified, and an initial heat map is obtained, as shown in Fig. 6 (a). The naked eye can clearly see that the identified areas are mainly the cat's head and body parts, roughly showing the outline of a cat, confirming the rationality of the original picture sample being identified as a tabby cat.

As a comparison, a model visualization is performed on a sample classified as a handheld computer, and a heat map is obtained as shown in Fig. 7. Comparing Fig. 6 and Fig. 7, we can see that there are obvious differences in the outlines of the two heat maps. The brightly colored parts of the heat maps of the adversarial samples are square and do not meet the visual characteristics of cats. The conclusion of the handheld computer in the picture.

The comparison of the examples above proves that the algorithm implemented in this paper has significant effects in adversarial example generation, CNN visualization, and adversarial example discrimination. However, the simple selection of correct prediction examples cannot guarantee the universality of the algorithm, so this article continues to verify the reliability of the algorithm from multiple perspectives.

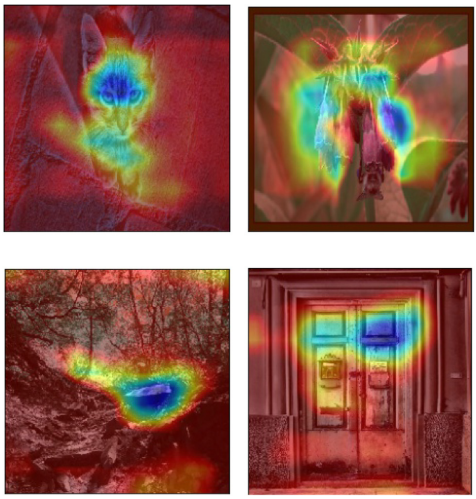

Fig. 6. Heat maps of the original examples

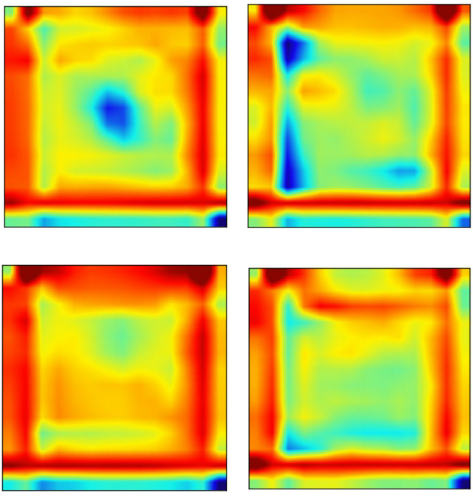

Fig. 7. Heat maps of the adversarial examples

4.2 Visualization of Wrong Sample Recognition

Compared with the first group of examples, a slightly more complex example is selected, as shown in Fig. 8. According to the label of the example, the original image should be a horse cart, but it is identified as an oxcart by the classification model. In addition, the target classification of the adversarial example is also set to a laptop. After inspection, the example with the same visual characteristics is successfully identified as a laptop by the model.

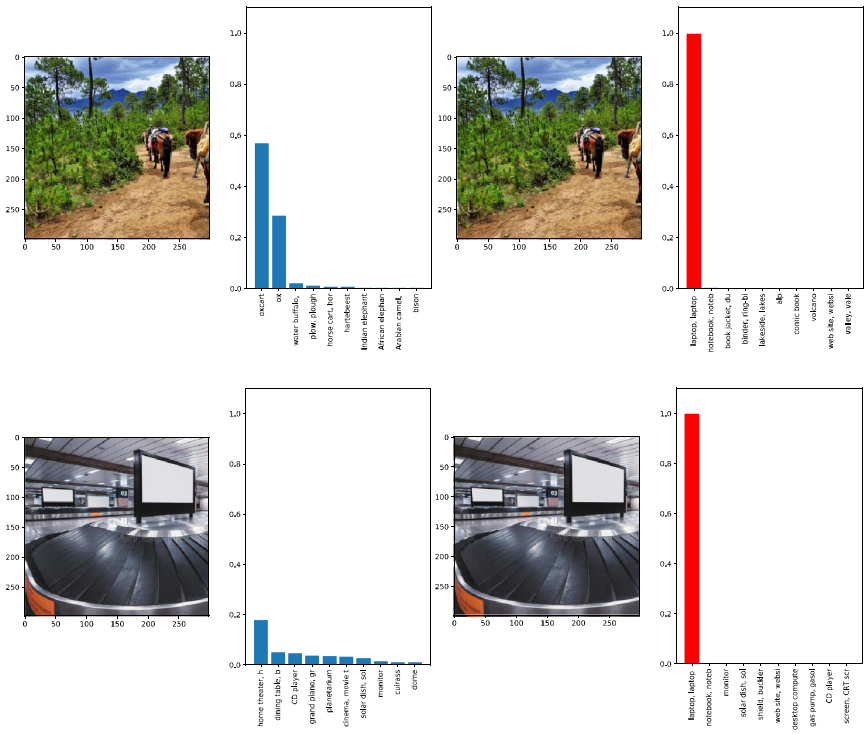

Fig. 8. Classification results of normal and adversarial examples

Unlike the first set of examples, the recognition result of the original sample is wrong. The user can find the cause of the identification error based on the heat map. As shown in Fig. 9, the model mainly recognizes the goods on horseback instead of the horse's head. Since the sample is not a frontal picture of the horse, it is easy to be misled, which makes the model mistake it for the oxcart, and then derives incorrect classification. Therefore, the heat map improves the interpretation ability of the model. Observing the heat map of the adversarial samples, it can be found that the features obtained by the model also do not match the visual characteristics of the horse or ox, which proves that the model can identify the anomalies of the adversarial examples and help distinguish between normal and adversarial examples.

After obtaining the CNN visualization algorithm based on the original neural network model, this paper repeatedly measures the accuracy of the algorithm to ensure the reliability of the algorithm. After 200 sample experiments, the model's prediction rate for image classification results is about 95%, but for the training set mixed with adversarial samples, the algorithm's success rate of identifying adversarial samples is close to 100%. This accuracy rate shows that the algorithm has a very high discrimination rate for the adversarial examples.

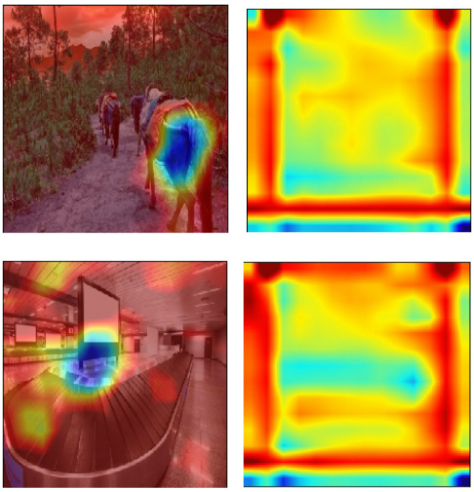

Fig. 9. Heat maps of normal and adversarial examples

5 Conclusions

This paper focuses on the application of visualized CNNs in adversarial example attacks. It studies the existing defense technologies and adversarial sample generation technologies, and considers that optimizing training examples is a more effective way to defend against adversarial example attacks. This paper implements the adversarial sample generation algorithm and the example recognition category activation mapping algorithm, which can accurately outline the recognition area of the original training examples and distinguish normal examples from adversarial examples according to the heat map. The algorithm can randomly select a picture for category determination and set target categories to generate adversarial samples. The visual neural network is used to delineate the image recognition area, determine the effective information of the picture, and find the anomaly in heat maps to distinguish the examples.

In addition, the visual CNN generated in this paper can accurately describe the recognition range of normal examples, and confirm anomalies in adversarial examples. And it is impossible to determine the recognition range of them. Follow-up studies will be conducted on how to improve the visualization performance and enhance the ability to explain the classification of adversarial examples.

References

1. Biggio, B., et al.: Evasion attacks against machine learning at test time. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) ECML PKDD 2013. LNCS (LNAI), vol. 8190, pp. 387–402. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40994-3_25

2. Papernot, N., Mcdaniel, P., Wu, X., et al.: Distillation as a defense to adversarial perturbations against deep neural networks. In: Proceedings of the IEEE Symposium on Security and Privacy, pp. 582–598 (2016)
3. Li, P., Zhao, W., Liu, Q., et al.: Review of machine learning security and its defense technology. Comput. Sci. Explor. **2**, 171–184 (2018)
4. Marco, T.R., Sameer, S., Carlos, G.: Why should i trust you?: explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1135–1144 (2016)
5. Qiu, Y., Li, S.: Security threat analysis and solutions for the development and application of artificial intelligence. Netinfo Secur. **9**, 35–41 (2018)
6. Chu, L., Hu, X., Hu, J., Wang, L.J., et al.: Exact and consistent interpretation for piecewise linear neural networks: a closed form solution. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1244–1253 (2018)
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
8. Yu, Y., Ding, L., Chen, Z.: Research on attacks and defenses towards machine learning systems. Netinfo Secur. **9**, 10–18 (2018)
9. Chris, O., Arvind, S.: The Building Blocks of Interpretability [OL]. https://opensource.googleblog.com/2018/03/the-building-blocks-of-interpretability.html. 3 June 2018
10. Zhang, Q., Zhu, S.: Visual interpretability for deep learning: a survey. Front. Inf. Technol. Electron. Eng. **19**(1), 27–39 (2018)
11. Li, Y., Yan, Z., Yan, G.: A edge-based 2-channel convolutional neural and its visualization. Comput. Eng. Sci. **41**(10), 1837–1845 (2019)
12. Zhang, S., Zuo, X., Liu, J.: The Problem of the Adversarial Examples in Deep Learning [OL]. http://kns.cnki.net/kcms/detail/11.1826.2018-1-20
13. Zhou, B., Khosla, A., Lapedriza, A., et al.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2921–2929 (2016)
14. Amit, D., Karthikeyan, S., Ronny, L., Peder, O.: Improving Simple Models with Confidence Profiles[OL]. https://arxiv.org/pdf/1807.07506.pdf. 19 June 2018
15. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
16. Zeiler, M.D., Taylor, G.W., Fergus, R.: Adaptive deconvolutional networks for mid and high level feature learning. In: Proceedings of 2011 IEEE International Conference on Computer Vision, pp. 2018–2025 (2011)
17. Ramprasaath, R.S., Michael, C., Abhishek, D., Devi, P., Ramakrishna, V., Dhruv, B.: Grad-CAM: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 618–626 (2017)