Adversarial attacks in deep learning

Rouying Weng

Faculty of science, The university of Melbourne, Melbourne, 3000, Australia

rouyingw@student.unimelb.edu.au

Abstract Accompanying the advancement of deep learning models, which are now used in many different areas such as Nature Language Processing (NLP), Computer Vision (CV), and so on, the computationally exceedingly powerful deep ability of deep learning has outstanding performance in handling various tasks, have become a hot topic of concern. At present, research illustrates that if the input samples are interfered with using the adversarial sample technique, it can make most mainstream neural network models make wrong judgment results. Therefore, it becomes an important issue how to compensate for the shortcomings of existing neural network techniques in terms of security and robustness. This paper first introduces the development of adversarial attack techniques and then describes the theoretical foundations, algorithms, and applications. Then this paper designs an experiment to verify whether ResNet18 can be attacked under adversarial attacks and then subsequently discusses the open problems and challenges deep learning faces.

Keywords: machine learning, deep learning, adversarial attack, adversarial attack, Whitebox attack.

1. Introduction

Accompanying the advancement of deep learning models, which are now used in many different areas such as Nature Language Processing (NLP), Computer Vision (CV), and so on, the computationally exceedingly powerful deep ability of deep learning has outstanding performance in handling various tasks [1-3]. However, the research community has found that most mainstream neural network models can make incorrect judgments if the input samples are interfered with using adversarial sample techniques. This makes the models much less secure if generative adversarial is used to finely interfere with perturbed benign samples, imperceptible to human vision and hearing. However, this is sufficient for the model to make incorrect predictions with high confidence, producing severe errors. Many cases were mentioned by Szegedy et al. in which models of various structures trained over various subsets and on different subsets of the training data were incorrectly classified for the same adversarial example [4, 5]. This suggests that the examples disturbed by generative adversarial are not irregular but are a large class of problems with generalizability.

From an information security perspective, adversarial sample techniques are divided into attack and defense techniques [6]. From the application point of view, adversarial sample techniques can be divided into computer vision and text. There are existing adversarial attacks that can be categorized into three types, Adversarial Attacks and Defenses in Deep Learning, and white-, gray-, and black-box attacks in accordance with the adversary's knowledge. First, the white-box attack assumes that the adversary has complete knowledge of the structure and parameters underlying the target model. Therefore, because of the white-box mode's cognition, the opponent can directly generate the adversarial sample according to the target model. Second, in the cognitively limited grey box threat model, this is due to the ignorance of the adversary's knowledge limited to the structure and parameters of the target model. Finally, in the black-box threat model, the adversary knows nothing about the model's structure and parameters and can only produce adversarial samples with the help of query privileges. There are many attack algorithms under the white-box model, such as the finite-memory Broyden-Fletcher-Goldfarb-Shanno algorithm, fast gradient symbolic method (FGSM), projected gradient descent (PGD), and DeepFool, among others [5]. An in-depth study of adversarial sample techniques can help researchers identify potential flaws in the robustness and security of neural network models, provide them with a deeper understanding and knowledge of the learning mechanism of neural network models, and address the interpretability of neural network models. This paper focuses on experimenting with the Fast Gradient Symbolic Method (FGSM) algorithm and illustrates the challenges encountered so far.

This paper performs the experiment with ResNet18 and test set CIFAR10 and, calculates the loss gradient of the input data and creates a scrambled image, then checks if the scrambled example is adversarial. In summary, this paper finds that with the increase of Epsilon size of Retrain, the initial accuracy continued to decline. However, the trend of model accuracy of Retrain was that with the increase of Eps, the accuracy first increased and then decreased, becoming more evident with the increase of Epsilon.

This paper first surveys and summarizes the state-of-the-art algorithms in the adversarial attacks and defenses field. Then this paper discusses the proposed attacks and defense techniques in Section 2 of this paper and details the Preliminaries and some of the background that needs to be used and described. In Section 3, this paper presents a detailed description of the mainstream adversarial sample techniques from both attack and defense perspectives, introduces two well-known datasets used in this experiment, and provides an overview and analysis of the experimental performance, results, and conclusions; Section 4 analyzes and discusses the current problems and challenges of neural networks and adversarial sample techniques; Section 5 concludes the paper.

2. Methods

2.1. Notation and formulas

At the outset, there is clarification of the definitions and notations used throughout this document. First, this paper defines the dataset needed for the test as a dataset of size N. Each data sample is denoted as x_i and carries the label y_i, and the dataset is denoted as$\{x_i, y_i\}_{i=1}^{\mathrm{N}}$ Second, this paper denotes f(x) as a neural network prediction whose input is x. Meanwhile, θ is first set as the weight of the model, and this paper denotes the corresponding adversarial loss of f(x) by $\mathcal{J}(\theta, x, y)$. This paper denotes the cross entropy between the adversarial task loss of classification and the label y as the optimization loss, expressed as $\mathcal{J}(f(x); y)$. Moreover, this paper defines $D(x,\ x')$ as the distance metric, is a scheduled distance constraint, which is also referred to as an allowable perturbation. Thus this paper express the adversarial sample here as $x^{'}: D(x,\ x') < \eta\ ,\ f(x) \neq y$.

2.2. Fast Gradient Symbolic Method

One of the earliest adversarial attacks to date would be the Fast Gradient Symbolic Attack (FGSM), was first introduced by Goodfellow et al [5, 7]. A white-box attack, FGSM assumes it is possible for the attacker to have access to the models' full structure, inputs, outputs and weights, and therefore aims to perform misclassification. The idea that FGSM attacks neural networks by using model learning and defuzzification is straightforward. FGSM is also a simple and effective adversarial attack against sample generation algorithms and is welcome. A white-box model accessible to all model contents, and because it is a white-box model, the FGSM attack exploits the gradient of the loss function. Then, to maximize the loss, the FGSM adjustment adjusts the input data. Elaborately, the adjustment would maximize the loss based on the same backpropagation gradient as opposed to minifying the loss by adjustment of the weights based on the backpropagation gradient. FGSM was an efficient non-targeted attack which generates adversarial samples in the L1 neighborhood of benign samples.

2.3. Residual network

Kaiming He et al. from Microsoft Labs in 2015 proposed the ResNet network, which awarded a lot of prizes, such as first place for target detection, champion for image segmentation in the COCO dataset, first place for classification task algorithm, and also win for the winner as in the ImageNet competition for target detection [8-10]. Residual blocks were conceptualized for addressing the problem of increasing data errors at the practical level instead after the addition of new layers for training. This is because the solution space of the original model is only the solution space of the original model. Because the solution space of the original model is only one subspace of the solution space of the new model. This paper thinks the new model has some potential to produce a more efficient solution to adapt and retrain the dataset. Then, making some specific layers of the neural network to skip the connections of neighboring neurons in the next layer, in order to solve the degeneration problem in the deep network, and connect in alternating layers, weakening the strong connections between each layer is called residual network. This network model greatly improves efficiency.

First of all, it is worth mentioning that in this experiment, the model this paper uses is a more classical residual algorithm, Resnet-18 ResNet can be compared to our traditional neural network, specifically its structure, first the first two layers are convolutional layers. Collectively, there will be 64 output channels, which can be divided into two steps and two steps. Then, there is a very large 3×3 pooling layer, and there are two steps in this layer. Then, the difference from the traditional convolutional layers is reflected in the fact that each convolutional layer of ResNet is followed by a batch normalization layer. Subsequently, we will find that the modules that make up ResNet can be divided into four by residual blocks. Separately, the number of input channels determines the number of channels in the first module, while the number of output channels determines how many residual blocks are used in each subsequent module. Then, since the maximum pooling layer with a step size of 2 was used previously, this paper needs to reduce the height and width of the ResNet module by half and double it. Summarizing the number of layers described above and connecting the output of the fully connected layers and counting the global average pooling layer, in total this paper finds that there are 18 layers, with four convolutional layers per module, the initial convolutional layer, and the final fully connected layer, and this model is often referred to as ResNet-18.

3. Experimental results and analysis

3.1. Data description

In this experiment, this paper uses the CIFAR-10 dataset recorded by Alex Krizhevsky, Vinod Nair and Geoffrey Hinton. The CIFAR-10 dataset has three official versions, and the first one is used in this paper: the CIFAR-10 python version. For this dataset, which is very well known and used in computer vision field, it has ten classes of images. Each category has 8,000 images, with a combined total of 80,000 images. Each of these images has a size of 32x32. Both images are in color. This paper divides the entire dataset into five training batches and one test batch. There are 10,000 images in each batch. With the test batch containing 10,000 images, it is the collection of 1,000 images randomly selected from each category.

3.2. Experimental setting

The FGSM algorithm attack implementation is divided into three main steps, defining the attack function, testing the function, and running the whole attack algorithm according to different epsilon. For the function of the algorithm attack, this paper has three inputs, which are the original image, the perturbed amount of epsilon in pixel way, and the loss gradient of the input image. In order to keep the domain of epsilon, this paper creates the perturbed image as follows and convert it isometrically to the range [0,1].

$$perturbed_image = image + epsilon \times sign(datagrad) = x + \epsilon \times sign\left(\nabla_x \mathcal{J}(\theta, x, y)\right) \quad (1)$$

Next, this paper calls the test function all perform a complete test of ResNet18, and the output of the function will be the accuracy of the attacked model under the attack with strength epsilon. this paper performs the test for each sample in the test set CIFAR10 and, calculate the loss gradient of the input data and, create a scrambled image, then check if the scrambled example is adversarial. At the same time, we record some successful adversarial examples for output and presentation. After defining the attack and test functions, this paper needs to run the attack itself, where our input is a list of epsilons. For each epsilon, we run a full test, save the accuracy after the test is completed, and plot the image to see more intuitively how the accuracy varies with the epsilon. When epsilon = 0 means that there is no attack, this paper tests the original accuracy of the model.

3.3. Results and analysis

The first output is the ratio of accuracy to epsilon. this paper expects the test precision to decrease as the epsilon increases. This is because larger epsilons mean this paper takes a larger step towards maximizing the loss. Based on the output values, it is easy to see that the run results align with our expectations, and the accuracy increases as the Epsilon increases, the results as shown in Table 1.

Table 1. Ratio of accuracy to epsilon.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.7665	0.0206	0.0175	0.0204	0.0202	0.0256	0.0364

Then, this paper retrains model according to the different values of epsilon and output the results. This paper can get that when Epsilon: (0.05 to 0.3) for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Table 2. Epsilon: 0.05 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.7274	0.1686	0.0409	0.0217	0.0164	0.013	0.0129

Table 3. Epsilon: 0.1 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.5954	0.0126	0.0084	0.0123	0.0158	0.019	0.0213

Table 4. Epsilon: 0.15 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.4636	0.0036	0.001	0.0006	0.0005	0.0006	0.0005

Table 5. Epsilon: 0.2 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.2725	0.0002	0.0001	0.0016	0.0054	0.014	0.0193

Table 6. Epsilon: 0.25 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.2524	0.0001	0.0001	0.0002	0.0034	0.0102	0.0171

Table 7. Epsilon: 0.3 for the adversarial sample retrain ResNet18, epoch=5, lr=0.001.

Eps	0.0	0.05	0.1	0.15	0.2	0.25	0.3
ACC	0.245	0	0	0.0004	0.0055	0.0137	0.0203

Based on the above experimental data and results, it is easy to find that the accuracy decreases as the Epsilon increases. Suppose the model is retrained with a larger Epsilon value, the larger the Epsilon. In that case, the less accurate the model is after training. In the neural network model used in this experiment, ResNet18, this paper concludes that the safety issues remain to be examined concerning the interpretability of the neural network model itself, as well as the robustness and security issues that have been revealed for years recently.

This paper finds that the overall variation trend of the accuracy of different corresponding Epsilon obtained after training the model with different Epsilon values was the same as that of the original model. When Epsilon was 0.05, this paper finds that there was little difference in accuracy between Epsilon and the original model, and there was almost no change in the decline. However, when Eps=0.05, the accuracy was significantly improved. When Epsilon is 0.10, this paper finds that the accuracy has changed significantly compared with the original model, and the whole model has a significant decline. In Table 3, when Eps=0.1, the accuracy does not improve because of retraining, but has a significant decline. In Table 4, Eps=0.15 was used to retrain our model, and the results showed that the accuracy decreased significantly compared with the original model, especially after Eps=0.15, and the results were close to 0.0005 and did not change significantly with the increase of Eps. In Table 5 and Table 6, this paper finds that the accuracy of the retrained model almost decreases by half and tends to decrease first and then increase with the increase of Eps. Finally, it can be found that in Table7, when Eps=0.05 and 0.1, the accuracy has been equal to 0, and with the subsequent increase of Eps, the accuracy will rise. In summary, this paper finds that with the increase of Epsilon size of Retrain, the initial accuracy continued to decline, but the trend of model accuracy of Retrain was that with the increase of Eps, the accuracy first increased and then decreased, and became more and more obvious with the increase of Epsilon.

At the same time, as a strategy and means to explore the learning mechanism of neural network models, the research on the adversarial sample technique has achieved richer results. However, there are still many urgent problems to be solved. Meanwhile, the research on the adversarial sample technique has not yet started with forming a systematic theoretical system of attack and defense, evaluating the security and robustness of neural network models, and measuring the quality of the adversarial samples. Other issues in the research community There is still no consensus in the research community.

4. Conclusion

With the development of deep learning models, which are now widely used in speech recognition, computer vision, natural language processing, and other fields, the security issue, especially the adversarial sample attack, has become a hot concern. Currently, research illustrates that if the input samples are disturbed using adversarial sample techniques, it can make most mainstream neural network models make wrong judgment results. In summary, this paper finds that with the increase of Epsilon size of Retrain, the initial accuracy continued to decline, but the trend of model accuracy of Retrain was that with the increase of Eps, the accuracy first increased and then decreased, and became more and more obvious with the increase of Epsilon. Therefore, it has become a vital issue to compensate for the shortcomings of existing neural network techniques in terms of security and robustness. More types of attacks such as PGD, C&W, Deepfool and AutoAttack will be tested in the next step of the study. because the types of adversarial attacks are limited in this paper, more types will be tested to answer the conclusions.

References

[1] Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., & Schmid, C. (2021). Vivit: A video vision transformer. In Proceedings of the IEEE/CVF International Conference on Computer

Vision (pp. 6836-6846).

[2] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 10012-10022).

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[4] Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., & Fergus, R. (2013). Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

[5] Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.

[6] Papernot, N., McDaniel, P., & Goodfellow, I. (2016). Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. arXiv preprint arXiv:1605.07277.

[7] Chen, J., Wu, Y., Xu, X., Chen, Y., Zheng, H., & Xuan, Q. (2018). Fast gradient attack on network embedding. arXiv preprint arXiv:1809.02797.

[8] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[9] Tai, Y., Yang, J., & Liu, X. (2017). Image super-resolution via deep recursive residual network. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3147-3155).

[10] Li, J., Fang, F., Mei, K., & Zhang, G. (2018). Multi-scale residual network for image super-resolution. In Proceedings of the European conference on computer vision (ECCV) (pp. 517-532).