2020 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR) | 978-1-7281-9985-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICWAPR51924.2020.9494387

SENSITIVITY BASED ROBUST LEARNING WITH SAMPLING IN ADVERSARIAL ENVIRONMENT

QING LIN, WEI LI, PATRICK P. K. CHAN

School of Computer Science & Engineering, South China University of Technology, Guangdong, China
E-MAIL: 1056115829@qq.com, mcrae666@163.com, patrickchan@scut.edu.cn

Abstract:

Although Deep Neural Networks achieve excellent results in many applications, their security issue is one of the concerns. Many studies suggest that Deep Neural Networks can be easily misled by adversarial attack. Sensitivity training method enhances the robustness of Deep Neural networks. However, injecting sensitivity samples for all training samples increases the training time significantly. This study investigates whether generating sensitivity samples for all training samples is necessary. A model trained by using sensitivity samples for selected training samples is proposed. The method is evaluated and compared with the traditional one and the adversarial learning method experimentally. The results suggest that the robustness of the models using sensitivity samples for the partial and full training sets is similar. The time complexity of sensitivity training methods can be reduced.

Keywords:

Robust learning; Adversarial attack; Sensitivity; Robustness improvement; Sample selection

1. Introduction

Deep learning achieves excellent results and milestones in many applications, *e.g.*computer vision [1, 2], natural language processing [3, 4], and control [5, 6], due to its powerful learning ability. However, many studies indicate that deep learning methods are vulnerable in an adversarial environment, in which an adversary crafts a sample intentionally in order to mislead the decision of the targeted system. As well as other machine learning methods, deep learning methods also assume the training and test sets follow the same (or similar) distributions. Since this assumption is destroyed by the adversarial attack, the performance of deep learning methods drop significantly. Some studies [7] indicate that a sample with small adversarial noise can confuse the deep learning methods. The vulnerability reduces the confidence of using deep learning in security related applications. For example, autonomous vehicles cannot correctly identify a sign [8], and face recognition wrongly identify a person [9].

A number of defense methods have been proposed to reduce the influence of adversarial attacks. These methods can be divided into three types [10]. Adversarial or noisy samples generated by an attack strategy are considered in training to enhance the robustness of the model in *data modification* method [11, 12], while the structure of the model is changed to reduce the influence of adversarial noise in *model modification* [13, 14]. The last type of method is *using external models* to filter contaminated samples [15].

Training with adversarial samples [11, 12] is one of the most popular and commonly used defense method. However, it counters a lot of drawbacks, for example, long training time and assumption on the attack strategy. To address these problems, the previous work [16] proposes a sensitivity-based robust training method. The study suggests that the robustness of a model is improved by reducing its sensitivity. However, considering additional training samples with sensitivity increases the size of the training set and also the training time significantly. The situation will be worse for deep learning methods since the training set is usually large.

This study follows the results in [16] and investigates whether sensitivity samples are necessary for all training samples or not. We propose the sensitivity training using sensitivity samples for partial training samples. The methods are then evaluated and compared with the adversarial training method empirically. Experimental results suggest that the sensitivity training is superior to the adversarial training in terms of robustness, time complexity and sensitivity in a Convolutional Neural Network and Stacked Autoencoder.

The rest of the paper is organized as follows. Section 2 introduces the related concepts. Our proposed methods are devised

978-1-7281-9985-6/20/$31.00 ©2020 IEEE

in Section 3. Section 4 provides the experimental results and discussion. Finally, the conclusion is given in Section 5.

2. Related Work

This section summarizes related works on evasion attacks and defenses.

2.1. Adversarial Attack

In an adversarial environment, an example may be carefully crafted to mislead a learning model. The adversary, who creates adversarial examples $\mathbf{x}_{adv}$, adds a small disturbance ε which is not perceivable but is able to mislead a targeted model. Most evasion attack algorithms generate adversarial examples based on the gradient of the loss function with respect to the input, defined as:

$$\mathbf{x}_{adv} = \mathbf{x} + \alpha sign(\nabla_X L(f_\theta(\mathbf{x}), y)) \tag{1}$$

where y represents the label corresponding to the clean sample $\mathbf{x}$, $f_\theta(\mathbf{x})$ denotes the model output, α is the small disturbance of each step, and L denotes the loss function. The Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_{adv}$ measures the evasion attack intensity.

Several attack methods have been proposed, and can be separated according to three aspects: the knowledge on the target model, attack target class, and disturbance generation. White box attacks (*e.g.*L-BFGS [7] and FGSM [26]) have all knowledge on the target model, while black box attacks have no information, for example, UPSET [27] and ANGRI [27]. Targeted attacks, including L-BFGS [7] and JSMA [28], aim to mislead the system into classifying the manipulated sample to a particular class. On the other hand, the objective of non-target attacks, like BIM [22] and PGD [18], is to downgrade the system. Single-step (*e.g.*FGSM [26] and R-FGSM [21]) and iterative attacks (*e.g.*BIM [22], DeepFool [24], and PGD [18]) craft an adversarial sample once and iteratively.

In this paper, Projected gradient descend (PGD) [18], which is a white-box, non-target and iterative attack, is considered due to its efficiency. PGD modifies a clean sample according to the direction with the largest gradient. The adversarial sample is crafted iteratively according to the following formula:

$$\mathbf{x}^{t+1} = Proj\{\mathbf{x}^t + \alpha sign(\bigtriangledown_{\mathbf{x}^t} L(f_\theta(\mathbf{x}^t), y))\} \tag{2}$$

where $\mathbf{x}^t$ is the adversarial sample generated in step t, $Proj$ guarantees the disturbance is in a range S. The example of contamination by PGD is illustrated in Figure 1.

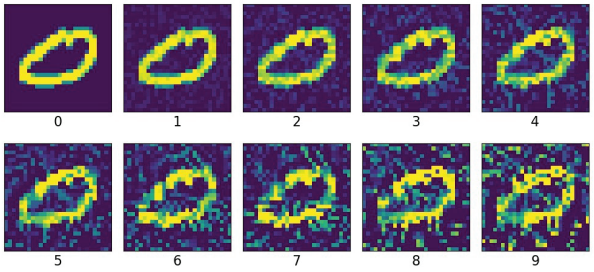


FIGURE 1. Adversarial images generated by PGD with different attack strengths. The number below the picture indicates the Euclidean distance between the corresponding attack image and the original image.

2.2. Adversarial Training

The robustness against adversarial attack is enhanced by adversarial training, which considers adversarial images in additional to the clean images. The adversarial training process can be summarized as:

$$\begin{aligned} &\min_\theta \{E_{(\mathbf{x},y)\sim D}[L(f_\theta(\mathbf{x}), y)] + \\ &E_{(Z,y)\sim D'}[\max_{||\delta||\leq\varepsilon} L(f_\theta(\mathbf{x}+\delta), y)]\} \end{aligned} \tag{3}$$

where δ denotes the disturbance limited by ε. The aim of the attack is to maximize the model classification loss on adversarial samples, while the purpose of the adversarial training is to minimize the overall loss on both clean and adversarial samples.

Although the robustness is enhanced by adversarial training to a certain extent, the training cost is one of the concerns since the manipulation cost of an adversarial sample is high. The situation is even worse for a deep learning model since its training set is usually large.

3. Sensitivity based Robust Learning with Sampling

In this section, we first describe the sensitivity based robust training [16], and our proposed sampling method is also devised.

3.1. Sensitivity based Robust Training

Consider a classification problem with a training set, $\mathcal{D}_{tr} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, where n is the number of samples in $\mathcal{D}_{tr}$, x_i is the ith sample and y_i is the corresponding label, and $x_i = [x_i^1, x_i^2, \ldots, x_i^d]^T \in R^d, y_i = [y_i^1, y_i^2, \ldots, y_i^m]^T \in R^m$ for a m-class classification task. The

objective function of sensitivity based robust method [16] is defined as follow:

$$min\ (1-\lambda)\sum_{(\mathbf{x},y)\in\mathcal{D}} L(C(\mathbf{x}),y)+\lambda\sum_{(\mathbf{x},y)\in\mathcal{D}} Sen(\mathbf{x}) \quad (4)$$

where C denotes the classification model, L denotes the loss function, and λ denotes the tradeoff parameter. The first and seconds terms represent the classification loss and sensitivity measure respectively. $Sen(\mathbf{x})$ measures the mean of the squared difference of the classifier outputs between a sample $\mathbf{x}$ and unseen samples within the Q-neighborhood of $\mathbf{x}$ [29, 30], defined as follow:

$$Sen(\mathbf{x}) = E(\|C(\mathbf{x}) - C(\mathbf{x}+\Delta\mathbf{x})\|_2) \quad (5)$$

Sen is approximated by the Monte Carlo method [33] defined in Eq. (6). The Q-neighborhood is first defined for each training sample. To be more specific, additional q samples, named as sensitivity samples, are generated artificially according to the Gaussian distribution with a mean of 0 and a variance of σ. Figure 2 illustrates an example of the concept of Q-neighborhood.

$$Sen(\mathbf{x}) \approx \frac{1}{q}\sum_{i=1}^{q}\|C(\mathbf{x}) - C(\mathbf{x}+\Delta\mathbf{x}_i)\|_2 \quad (6)$$

where $\Delta\mathbf{x}_i \sim \mathcal{N}(0,\sigma)$

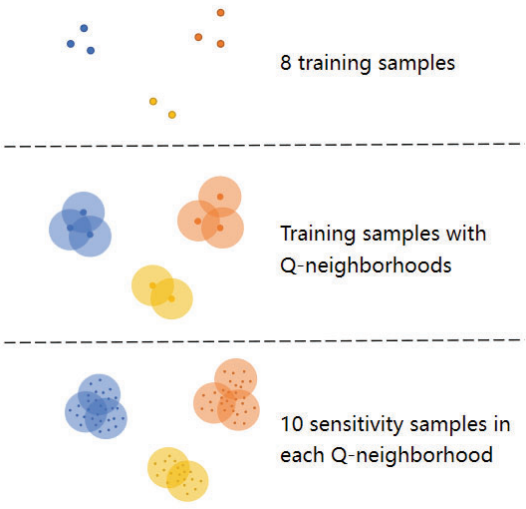


FIGURE 2. Example of Q-neighborhoods and sensitivity sample

The sensitivity training is able to obtain a more robust decision boundary. The model shown in 3(b) is more preferable since it has lower classification error and sensitivity compared to the one in 3(a). The decision boundary is further away from the samples, which increases the cost of attacks.

(a) Traditional training method (b) Sensitivity based robust training method

FIGURE 3. Example of decision boundary trained by different methods

3.2. Proposed Method

Although the generation of sensitivity samples is much simpler than the one of adversarial samples, considering sensitivity samples increases the time complexity significantly. Sensitivity samples for each training sample may not be necessary. Sensitivity based Robust Training with sampling is introduced in this section.

Rather than calculating sen for each training sample, the revised model only focuses on a subset of samples, $\mathcal{S} \subseteq \mathcal{D}$. Eq. (4) is modified as follows. $\mathcal{S}$ can be chosen in each learning epoch according to a criterion, for example, random and Q-value by Eq. (6).

$$min\ (1-\lambda)\sum_{(\mathbf{x},y)\in\mathcal{D}} L(C(\mathbf{x}),y)+\lambda\sum_{(\mathbf{x},y)\in\mathcal{S}} Sen(\mathbf{x}) \quad (7)$$

4. Experiments

In this section, the performance of our proposed method is verified and compared with other baseline methods in terms of adversarial robustness, noisy robustness, and time complexity under the PGD attack [18].

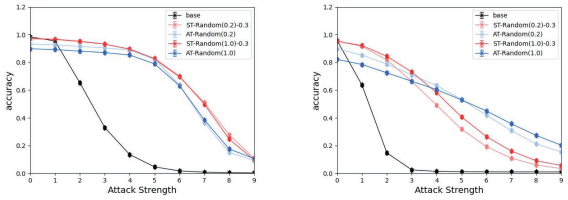
4.1. Experiment settings

MNIST dataset is used in our experiments. We performed a five-fold cross-validation during the experiment. All experimental results are the average of five independent runs. All feature values are normalized to $[0, 1]$ by dividing by 255. VGG3 [31] and stacked autoencoder are used as classifiers to verify the training algorithms. The stacked autoencoder contains five hidden layers with 128, 64, 8, 64 and 128 neurons. For PGD attack [18], the attack intensity of each step is 0.05. When generating attack samples to test the model, the criterion for stopping

the attack is to reach the maximum attack distance. Different attack strengths, *i.e.* 1,2,...,9, are considered. All distances are measured by the Euclidean distance.

Our sensitivity based training method (*ST*) is compared with the original (*base*) and adversarial training methods (*AT*). The training parameters of all methods are the same. Maximum epoch is set as 15 when training VGG3, while it is set as 20 and 10 in the unsupervised training and supervised fine-tuning phases respectively when training Autoencoder. The random based sampling methods are considered with the sampling rate of 0.2 and 1.0 for our method. The variance of Q-neighborhood is set to 0.3, and five samples are used in the Monte Carlo method, *i.e.* $q = 5$. In adversarial training, the adversarial attack setting is the same as the one used in test except the criterion for stopping the attack is to successfully mislead the model or reach the maximum attack distance 5.

4.2. Robustness against Adversarial Attack



(a) VGG3 (b) Autoencoder

FIGURE 4. The accuracy of models trained by different methods under PGD attacks with different attack strengths.

Figure 4 shows the accuracy of the models under evasion attacks with different attack strengths. Both robust training methods, *i.e.* *ST* and *AT*, are able to improve the robustness effectively. When under the attack of strength 5, the accuracy of the robust trained VGG3 is about 80%, while *baseline* is less than 10% accurate. In 4(a), *ST* is more robust than *AT* under attack with all strength. When the attack strength is weak, *ST* is more robust than *AT* in 4(b). Moreover, when there is no attack, *i.e.* the attack intensity is 0, the accuracy of *AT* is lower than that of *baseline* significantly. This shows that considering adversarial samples in training reduces the accuracy on clean samples. In contrast, *ST* does not have this shortcoming. Random sampling does not reduce the robustness of *ST* and *AT* significantly. The accuracy between *ST-Random(0.2)* and *ST-Random(1.0)*, and *AT-Random(0.2)* and *AT-Random(1.0)* is similar. This confirms that generating sensitivity of adversarial samples for each training sample is unnecessary.

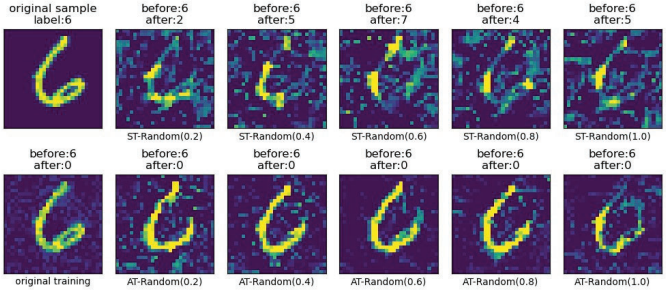
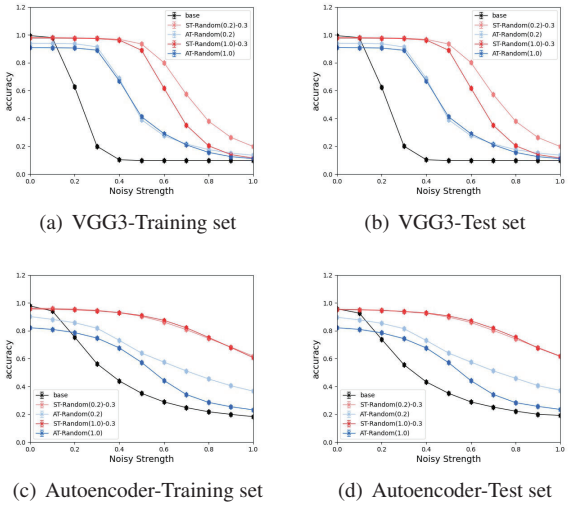


FIGURE 5. An original image and corresponding successful attacked images with the minimum modifications for each VGG3 model.

Examples of a successfully attacked sample on different VGG3 models are discussed. An original sample which is a digit six, and the contaminated samples that successfully attacked each model with the minimal modification are shown in Figure 5. The figure shows that *base* can be easily misled by a small modification, *i.e.* the difference between the original one and the one successfully attacked *base* is small. More modification is required by the successfully attack to *ST* compared to *AT*, *i.e.* the contamination of *ST* is more messy and looks much different from the original samples. The modification difference between using various sampling rates are not obviously for both *ST* and *AT*. Once again, the results indicate that using a smaller sampling rate may achieve the same effect as constructing the sensitivity samples for all samples.



(a) VGG3-Training set (b) VGG3-Test set

(c) Autoencoder-Training set (d) Autoencoder-Test set

FIGURE 6. The accuracy of models trained by different methods with different strength noisy samples in the training and test sets.

4.3. Noisy Robustness

A random noise is added to each training and test sample to evaluate the noisy robustness of the models. The strength of the random noise is limited to $\{0.1, 0.2, ...,1.0\}$. Figure 6 shows the evaluation results. Compared to *base*, both *ST* and *AT* significantly improve the noisy robustness of the model. For example, when the noisy strength is 0.4, *ST* and *AT* still maintain a high accuracy, but the accuracy of *base* has dropped to less than 0.1 in 6(a) and 6(b). The robustness improvement of *ST* is higher than *AT*, such as, the model trained by *ST* adopt larger noise than *AT*.

4.4. Time Complexity

Table 1 shows the time used by *ST* and *AT* at different sampling rates on the VGG3 model. The performance of *ST* is only slightly better than *AT* in terms of robustness, but the training time of *ST* is much shorter. For example, when the sampling rate is 0.8, it takes us more than 4 hours to training a model using *AT*, which is about five times the training time of *ST* under the same sampling rate. A sensitivity sample is generated based on a randomly process, while an adversarial sample is generated by an optimization process. It explains why the training time required by *ST* is significantly shorter than *AT*.

TABLE 1. Training time comparison between *ST* and *AT* with different sampling rates on the VGG3 model in minute.

Sampling Rate	20%	40%	80%	100%
ST	19.55	32.97	48.97	51.08
AT	64.48	129.07	251.58	258.75

5. Conclusions

In this paper, the sensitivity-based robust learning with sampling is investigated. The experimental results confirm that the model using sensitivity sampling for partial samples performs similar to the one using the whole training set. This result reduces the time complexity of the defense method using adversarial samples, in which additional samples are considered, in order to make this kind of methods more practical.

Acknowledgments

This paper is supported by the Natural Science Foundation of Guangdong Province, China (No. 2018A030313203) and the Fundamental Research Funds for the Central Universities (No. 2018ZD32).

References

[1] Voulodimos, Athanasios, et al. "Deep learning for computer vision: A brief review." Computational intelligence and neuroscience, 2018.

[2] Brunetti, Antonio, et al. "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey." Neurocomputing, Vol 300, pp. 17-33, 2018.

[3] Li, Hang. "Deep learning for natural language processing: advantages and challenges." National Science Review, 2017.

[4] Otter, Daniel W., Julian R. Medina, and Jugal K. Kalita. "A survey of the usages of deep learning for natural language processing." IEEE Transactions on Neural Networks and Learning Systems, 2020.

[5] Fadlullah, Zubair Md, et al. "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems." IEEE Communications Surveys & Tutorials, Vol 19, No. 4, pp. 2432-2455, 2017.

[6] Kuutti, Sampo, et al. "A survey of deep learning applications to autonomous vehicle control." IEEE Transactions on Intelligent Transportation Systems, 2020.

[7] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).

[8] Evtimov, Ivan, et al. "Robust physical-world attacks on machine learning models." arXiv preprint arXiv:1707.08945 2.3 (2017): 4.

[9] Rozsa, Andras, et al. "Facial attributes: Accuracy and adversarial robustness." Pattern Recognition Letters, Vol 124, pp. 100–108, 2019.

[10] Akhtar, Naveed, and Ajmal Mian. "Threat of adversarial attacks on deep learning in computer vision: A survey." IEEE Access, Vol 6, pp. 14410-14430, 2018.

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

[12] Sankaranarayanan, Swami, et al. "Regularizing deep networks using efficient layerwise adversarial training." arXiv preprint arXiv:1705.07819 (2017).

[13] Gu, Shixiang, and Luca Rigazio. "Towards deep neural network architectures robust to adversarial examples." arXiv preprint arXiv:1412.5068 (2014).

[14] Ross, Andrew Slavin, and Finale Doshi-Velez. "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients." arXiv preprint arXiv:1711.09404 (2017).

[15] N. Akhtar, J. Liu, A. Mian, Defense against Universal Adversarial Perturbations, arXiv preprint arXiv:1711.05929, 2017.

[16] Chan, Patrick PK, et al. "Sensitivity based robust learning for stacked autoencoder against evasion attack." Neurocomputing, Vol 267, pp. 572-580, 2017.

[17] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).

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

[19] Kim, Yoon. "Convolutional neural networks for sentence classification." arXiv preprint arXiv:1408.5882 (2014).

[20] Hongtao, Lu, and Zhang Qinchuan. "Applications of deep convolutional neural network in computer vision." Journal of Data Acquisition and Processing, Vol 31, No. 1, pp. 1-17, 2016.

[21] Tramèr, Florian, et al. "Ensemble adversarial training: Attacks and defenses." arXiv preprint arXiv:1705.07204 (2017).

[22] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial examples in the physical world." arXiv preprint arXiv:1607.02533 (2016).

[23] Dong, Yinpeng, et al. "Boosting adversarial attacks with momentum." Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 9185-9193, June 2018.

[24] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks." Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574-2582, June 2016.

[25] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).

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

[27] Sarkar, Sayantan, et al. "UPSET and ANGRI: Breaking high performance image classifiers." arXiv preprint arXiv:1707.01159 (2017).

[28] Papernot, Nicolas, et al. "The limitations of deep learning in adversarial settings." 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, pp. 372-387, March 2016.

[29] Yeung, Daniel S., et al. "Localized generalization error model and its application to architecture selection for radial basis function neural network." IEEE Transactions on Neural Networks, Vol 18, No. 5, pp. 1294-1305, 2007.

[30] Yeung, Daniel S., Patrick PK Chan, and Wing WY Ng. "Radial basis function network learning using localized generalization error bound." Information Sciences, Vol 179, No. 19, pp. 3199-3217, 2009.

[31] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).

[32] Moosavi-Dezfooli, Seyed-Mohsen, et al. "Universal adversarial perturbations." Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1765–1773, 2017.

[33] Metropolis, Nicholas, and Stanislaw Ulam. "The monte carlo method." Journal of the American statistical association, Vol 44, No. 247, pp. 335-341, 1949.