Detect Adversarial Example Using Gaussian Process-based Detector

Sangheon Lee, Noo-ri Kim and Jee-hyong Lee
Department of Electrical and Computer Engineering, Sungkyunkwan University
Suwon, South Korea
[e-mail: lawlee1@skku.edu, pd99j@skku.edu, john@skku.edu]
*Corresponding author: Jee-Hyong Lee

Abstract

Adversarial attack is a technique that causes a malfunction of classification models by adding noise that cannot be distinguished by humans, which poses a threat to a deep learning model applied to security-sensitive systems. In this paper, we propose a simple and fast method to detect adversarial image using Gaussian process classification. Existing machine learning-based adversarial detect methods require a large number of adversarial images for the learning of the detector. The proposed method overcomes this problem by performing classification based on the statistical features of adversarial images and clean images that extracted by Gaussian process with small number of images. The proposed method can determine whether the input image is an adversarial image by applying Gaussian process classification based on the predicted value of the classifier model.

Keywords: Adversarial Attack, Adversarial Detection, Gaussian Process Regression

1. Introduction

Adversarial Attack is a technique that changes the results of a classification or regression model by mixing perturbations that is imperceptible to human in the input data of the model. Recently, adversarial attack techniques which deceive the image classification neural network model are actively studied [3, 4, 5, 6, 7]. For a given natural image x that has no perturbation, adversarial attack produces an image x' that is visually similar but has a different classification result. x' is called an adversarial example, and by creating such an adversarial example through the attack, attackers can obstruct the neural network model from functioning. If the neural network model is applied as an important part of the system, it can lead to serious security problems.

Several techniques have recently been proposed to protect against adversarial attack. The defense aims to get the result of x for a given adversarial example x'. However, several defense techniques show poor performances against powerful attack techniques. Recent researchers proposed detection method instead of adversarial defense, which determine whether a given image is an adversarial example or not [8, 9]. If the detector determines that the given image is an adversarial

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014M3C4A7030503). Also, this research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF-2017M3C4A7069440).

example, the image classification model can prevent the attacker's attack by rejecting the classification task of the adversarial images.

However, many of the detection methods already proposed have a neural network structure, which requires a large number of adversarial examples to training. In particular, applying an attack technique that generates adversarial examples using gradients of the model can generate the adversarial example that deceives not only the classifier model but also the detector, and the detection technique applied to the model can be useless [1].

In this paper, we propose a simple and fast detection method for adversarial example. The proposed model extracts intermediate feature values generated by the classification model for a given image and uses this information to determine whether the image is an adversarial example by applying Gaussian process classification [2]. Experimental results show that our detection model has good results with fewer adversarial examples than other neural network-based detectors. Especially for powerful attacks with high attack success rates, our detection model outperforms baseline model.

2. Backgrounds

2.1 Adversarial Attack

The basic purpose of the adversarial attack is to create an example with a minimal perturbation that looks similar to a natural image but causes the target model to be misclassified. Various types of attack techniques have been proposed to date. Goodfellow et al. [3] introduced Fast Gradient Sign Method, an attack technique that uses the gradient value of the loss function of a model for a given natural image. Given a natural image, FGSM creates an adversarial example by adding a perturbation of magnitude ε to the image in the opposite direction of the gradient generated by the model. For small ε, the FGSM achieves a high success rate for the DNN model. Take Kurakin et al. [4] proposed an iterative version of the FGSM, the Basie Iterative Method, which updates the direction of the gradient in the repeated steps and adds up the perturbation of magnitude ε in the opposite direction of the updated direction. Papernot's Jacobian-based Saliency Map Attack [5] performs targeted attacks through the adversarial saliency map. Moosavi-Dezfooli et al. [6] proposed DeepFool attack that updates the perturbation vector for the natural image every iteration and performs the algorithm until the result image is misclassified for the first time. Carlini and Wagner [7] proposed C&W attack that is a kind of targeted attack and has the best performance than the other attacks proposed so far. The above attacks are applicable to different models in various data and show high attack success rate.

2.2 Adversarial Detection

Adversarial detection is a technique for judging whether a given image is an adversarial example and has been recently studied with adversarial defense. Grosse et al. [8] proposed a modification of adversarial re-training that is one of adversarial defense technique. For a classification model with N result classes, a new N+1th class corresponding to the adversarial image is added to perform detection and the model learn using natural images and adversarial images. Adversarial re-training showed good detection performance for MNIST dataset, but detection for CIFAR10 dataset showed a poor performance of 70% detection rate and 40% false positive rate [1]. Metzen et al. [9] proposed a detector that performs detection using the output value of the inner convolution layer of the classification model and showed high detection accuracy in experiments using MNIST and CIFAR10 dataset. However, such a deep neural network-based detection method has a disadvantage in that it requires a large number of adversarial examples to train the detector.

2.3 Gaussian Process Regression

Gaussian process is a random process in which every finite collection of random variables has a multivariate normal distribution. Gaussian process regression is a technique to infer the mean and variance of the whole data range based on the observed data by defining the relationship between the data using the characteristics of the data, assuming that distribution of the data follows the Gaussian process [2]. Assuming $\mathrm{f(x)} \sim \mathrm{N}\left(0, \mathrm{K}(\theta, \mathrm{x}, \mathrm{x}')\right)$ for the function $f(x)$ of x, the log marginal likelihood is as follows:

$$\log p(f(x)|\theta, x) = -\frac{1}{2} f(x)^T K(\theta, x, x')^{-1} f(x) - \frac{1}{2} \log\det\left(K(\theta, x, x')\right) - \frac{|x|}{2} \log 2\pi$$

$K(\theta, x, x')$ is a covariance matrix for all possible observed data pairs (x, x'), calculated from a pre-defined covariance function, and θ is hyperparameter of the covariance function. Based on the θ that maximizing this marginal likelihood, the distribution of the function value $f(x^*)$ for the unobserved data x^* is $p(y^*|x^*, f(x), x) = N(y^*|A, B)$. That is, posterior has mean A and variance B, and A and B are calculated through the following equations.

$$\alpha = K(\theta, x^*, x) K(\theta, x, x')^{-1}$$

$$A = \alpha f(x)$$

$$B = K(\theta, x^*, x^*) - \alpha K(\theta, x^*, x)^T$$

Since Gaussian process regression defines prior and predicts posterior in consideration of covariance between data, it is possible to obtain more accurate regression results than a general regression method with only a small number of observed data for the data that follow the Gaussian process. Our proposed detection method works based on Gaussian process regression, so it can achieve high detection accuracy with only a small number of adversarial examples.

3. Gaussian Process-based Detector

We propose a method for detecting adversarial examples based on Gaussian process regression. We extract the intermediate features generated by the pre-trained classification model for natural or adversarial images and use these as the input of the Gaussian process-based detector. The intermediate feature is the output vector of the model's last hidden layer, whose dimension is the class number of the image set. Fig. 1 shows the structure of our proposed adversarial image detector.

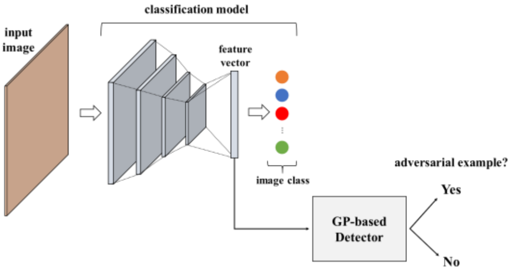

Fig. 1. Gaussian process-based adversarial image detector

The output value of the model's last hidden layer is the classification probability value for the image. In the case of adversarial images far from the two centers of the image classification binary, classification probability values for the two classes tends to be similar to each other. The detector would train this information to perform detection.

In the Gaussian process classification, the influence between two similar data is defined as covariance. If the dimension of data is high, it is difficult to grasp the pattern of covariance between data. Therefore, Gaussian process regression can be performed more efficiently by inputting low-dimensional high-level feature extracted through convolution and pooling layer rather than high-dimensional raw image.

4. Experiments

To verify the performance of our proposed Gaussian process-based detector, datasets used in the experiments are MNIST and CIFAR10. The attack methods used in the experiment are FGSM, BIM, DeepFool, JSMA and C&W attack and 500 natural images and 500 adversarial images are used for detector model training. Table 1 shows the accuracy of the classification model for the adversarial images and the average L2 norm of the perturbation generated by the attack. The covariance function used in the proposed model is Matern 5/2 [10]. The baseline model compared with our model is the binary classification model

Table 1. Model accuracy for adversarial images and average L2 Norm

	FGSM		BIM		DeepFool		JSMA		C&W	
	L2	Acc.	L2	Acc.	L2	Acc.	L2	Acc.	L2	Acc.
CIFAR10	1.94	14.44%	1.33	9.93%	0.11	5.95%	3.88	1.03%	0.08	1.07%
MNIST	6.47	8.20%	2.94	12.96%	1.86	0.63%	5.11	0.13%	1.43	0.63%

Table 2. Detect accuracy for MNIST dataset

	FGSM	BIM	DeepFool	JSMA	CW
Baseline	**98.74%**	**93.24%**	48.68%	61.83%	48.07%
GP-based	87.06%	69.08%	**99.70%**	**95.88%**	**99.70%**

Table 3. Detect accuracy for CIFAR10 dataset

	FGSM	BIM	DeepFool	JSMA	C&W
baseline	72.15%	48.10%	48.17%	**97.75%**	47.33%
GP-based	**73.18%**	**62.05%**	**97.70%**	94.08%	**97.68%**

of the convolutional neural network structure proposed by Gong et al. [11], and the data used for training of proposed model are set to 500 natural images and 500 adversarial images for the same experimental conditions.

4.1 MNIST

The model for classifying MNIST datasets is a simple 5-layer convolution neural network consisting of two convolution layers, one max pooling layer, and two dense layers. 60,000 of 28×28×1 MNIST images are used for training, and 10,000 images are used for validation. The optimizer used for model training is Adadelta, training epoch is 20, learning rate is 0.001, and batch size is 128. As a result of the training, the accuracy of the classification model for MNIST data is 99.3%. The hyperparameters of the five atttack techniques are set as follows; for FGSM and BIM, ε is 0.4. For the C & W attack, we set the maximum iterations to 1000, the initial constant to 0.001, and the learning rate to 0.005.

The experimental results for the MNIST dataset are shown in Table 2. For FGSM and BIM attack, detection accuracy was relatively lower than baseline, but for DeepFool, JSMA, and C & W attacks, which have higher attack success rates, our model far superior the baseline model. Since the detection accuracy of DeepFool, JSMA and C&W attacks are quite low, we can observe that the baseline model, which is a deep neural network, cannot train at all with a few training images.

4.2 CIFAR10

The classifier model that trains CIFAR10 dataset is 32 layers ResNet model [12], and 60,000 of CIFAR10 images used for training and 10,000 images used for validation. The optimizer used for model training is Adam, training epoch is 100, learning rate is 0.001, and batch size is 128. As a result of learning, the accuracy of the 32 layer ResNet classification model is 91.41%. For FGSM and BIM, ε is set to 9/255. Hyperparameters for other attacks are same as the previous MNIST experiment.

Table 3 shows the detection performance of baseline model and our proposed model in experiments using CIFAR datasets. Experimental results show that the Gaussian process-based model shows better detection performance than the baselines except JSMA attack. Due to the small training dataset, baseline model cannot train at all for detect BIM, DeepFool, and C&W attacks.

5. Conclusions

In this paper, we propose simple Gaussian process-based adversarial detection model. Experimental result shows that our model shows better performance than the baseline for a small number of adversarial images. For the future work, we plan to improve the performance of our detector by reflecting the characteristics of the adversarial image generated by FGSM and BIM attacks.

References

[1] N. Carlini, and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," in *Proc. of the 10th ACM Workshop on Artificial Intelligence and Security (AISec'17)*, 2017.

[2] H. Nickisch, and C.E. Rasmussen, "Approximations for Binary Gaussian Process Classification," *Journal of Machine Learning Research*, Vol 9, pp. 2035-2078, 2008.

[3] I.J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *Proc. of International Conference on Learning Representations (ICLR)*, 2015.

[4] A. Kurakin, I.J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Proc. of International Conference on Learning Representations (ICLR)*, 2017.

[5] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z.B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," in *Proc. of the 1st IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 372–387, 2016b.

[6] S.M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2574–2582, 2016.

[7] N. Carlini, and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," in *Proc. of IEEE Symposium on Security and Privacy (SP)*, 2017.

[8] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, "On the (Statistical) Detection of Adversarial Examples," *arXiv preprint arXiv:1702.06280*, 2017.

[9] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On Detecting Adversarial Perturbations," in *Proc. of International Conference on Learning Representations (ICLR)*, 2017.

[10] J. Snoek, H. Larochelle, and R.P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2012.

[11] Z. Gong, W. Wang, and W.S. Ku, “Adversarial and Clean Data Are Not Twins,” *arXiv preprint arXiv:1704.04960*, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.