2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)

Selective Poisoning Attack on Deep Neural Network to Induce Fine-Grained Recognition Error

Hyun Kwon
School of Computing
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: khkh@kaist.ac.kr

Hyunsoo Yoon
School of Computing
Korea Advanced Institute of Science and Technology
Daejeon, South Korea
Email: hyoon@kaist.ac.kr

Ki-Woong Park*
Computer & Information Security
Sejong University
Seoul, South Korea
Email: woongbak@sejong.ac.kr
*Corresponding author

***Abstract*—Deep neural networks (DNNs) provide good performance for image recognition, speech recognition, and pattern recognition. However, a poisoning attack is a serious threat to DNN's security. The poisoning attack is a method to reduce the accuracy of DNN by adding malicious training data during DNN training process. In some situations such as a military, it may be necessary to drop only a chosen class of accuracy in the model. For example, if an attacker does not allow only nuclear facilities to be selectively recognized, it may be necessary to intentionally prevent UAV from correctly recognizing nuclear-related facilities. In this paper, we propose a selective poisoning attack that reduces the accuracy of only chosen class in the model. The proposed method reduces the accuracy of a chosen class in the model by training malicious training data corresponding to a chosen class, while maintaining the accuracy of the remaining classes. For experiment, we used tensorflow as a machine learning library and MNIST and CIFAR10 as datasets. Experimental results show that the proposed method can reduce the accuracy of the chosen class to 43.2% and 55.3% in MNIST and CIFAR10, while maintaining the accuracy of the remaining classes.**

***Index Terms*—Poisoning attack, machine learning, deep neural network, chosen class.**

I. Introduction

Deep neural networks (DNNs) [11] provide excellent performance for machine learning tasks such as image recognition, speech recognition, and pattern recognition. However, there are two attack methods [2] that threaten the security of DNNs: causative attack [3] and exploratory attack [14]. A causative attack is an attack that degrades the accuracy of the model by approaching the training process of the model. On the other hand, exploratory attack exploits misclassification of models without affecting the training process. A causative attack has the advantage of directly attacking the model rather than an exploratory attack.

There is a poisoning attack [3] which is a typical attack in a causative attack. The poisoning attack reduces the accuracy of the model by adding malicious data to the training process of the model. This attack is a critical threat to the medical field and autonomous vehicles where the accuracy of the model is important. Conventional studies on this poisoning attack have focused on reducing the overall accuracy of the model. However, it may be necessary to reduce the accuracy of the chosen class in certain situations, such as in military situations. For example, an attacker would need to prevent UAV from detecting only nuclear-related facilities, except at other facilities. In such cases, it is important to ensure that only the intended nuclear facilities are misrecognized and the remainder are correctly recognized.

In this paper, we propose a selective poisoning attack that reduces the accuracy of a chosen class in the model. When the training data is accessed, the proposed method intentionally adds malicious data corresponding to a chosen class to decrease the accuracy of the chosen class and maintain the accuracy of another classes. The contribution of this paper is as follows.

- To the best of our knowledge, this is the first study that proposes a selective poisoning attack. We systematically organize the framework and principle of the proposed scheme.
- We analyze the selective accuracy depending the number of selective malicious data. We also analyze the iteration, distortion, and accuracy for selective malicious data.
- Through experiments using MNIST [8] and CIFAR10 [6], we show the effectiveness of the proposed scheme.

The remainder of this paper is as follows. Section 2 introduces the related research, and Section 3 introduces the proposed method. The experiment is described and evaluted in Section IV. A discussion of the proposed scheme is presented in Section V. Finally, we draw our conclusions in Section VI.

II. Related Work

We describe the neural network in general and introduce poisoning attack method..

A. Neural networks

A neural network [12] is a machine learning algorithm that models the brain's learning method mathematically; it refers to the overall model that forms a network by the combining of neurons and synapses. The structure of the neural network consists of an input layer, a hidden layer, and an output layer. At the input layer, there is a neuron for each input variable, matched 1:1 . In the hidden layer, there are neurons generated by the combination of neurons and weights of the input layer; the complexity of the model is determined by the number of

978-1-7281-1488-0/19/$31.00 ©2019 IEEE
DOI 10.1109/AIKE.2019.00033

layers within the hidden layer. In an output layer, neurons are generated by combining neurons and weights in the hidden layer; the number of output layers is determined by the type of output to be predicted. The neurons in the hidden layer and the output layer perform the function of calculating the sum of the input values and weights in the previous layer. These also execute an activation function that outputs the weighted sum of the neurons as the input value for the next layer. The neural network learns through learning data and sets the parameters of each layer by selecting parameters with optimal loss values using backpropagation and gradient descent.

B. Poisoning attack

A poisoning attack [3] [15] [9] is a causative attack method that reduces the accuracy of a model by adding malicious data between processes in the training of the model. There is a strong assumption that this attack will have access to the training process of the model, but it has the advantage of effectively reducing the accuracy of the model. Biggio et al. [3] were the first to propose a poisoning attack against a support vector machine (SVM). This method reduces the accuracy of an SVM machine by injecting malicious data into the training data. In this method, the aim is to calculate a gradient descent based on the characteristics of the SVM to generate some point samples that can be dropped by maximizing the accuracy of the SVM. Yang et al. [15] proposed a method for a poisoning attack against neural networks (NNs) rather than SVM models. Their method uses a direct gradient method to generate data by a generative adversarial net (GAN) through an auto-encoder. This method sets the target model as a discriminator, and the generator searches for optimal malicious data from the discriminator by a zero-sum method. In addition, Mozaffari-Kermani et al. [9] proposed systematic poisoning attacks in healthcare. With their method, they demonstrated a poisoning attack on a healthcare dataset by extending the domain to the medical domain.

III. Proposed Scheme

The purpose of the proposed scheme is to add selective malicious data between training process as a poisoing attack which lowers the accuracy of a chosen class. Fig. 1 shows an overview of the proposed method. As shown in Fig. 1, malicious data corresponding to a chosen class by the attacker is added to the training data.

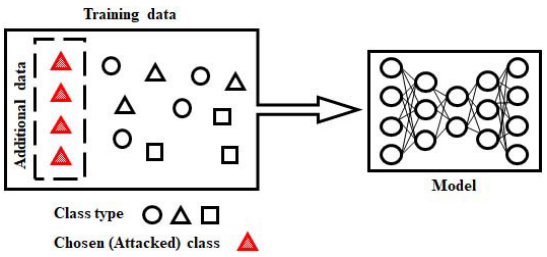

Fig. 1: A overview of the proposed scheme

The procedure of the proposed method is divided into two processes: a process of generating malicious data and malicious data addition to training data. First step, the process of generating malicious data $x_i^{'} \in X^{'}(1 \leq i \leq N^{'})$ is as follows. Given the original training data $x_i \in X$ with a chosen class $y^{'}$, it generates the malicious data $x_i^{'}$ with the smallest probability to be recognized as a specific class $y^{'}$ by the model. To generate the malicious data $x_i^{'}$ with the smallest probability of a specific class $y^{'}$, $loss$ must be minimized:

$$loss = Z(x_i^{'})_{y'} - \max\left\{Z(x_i^{'})_i : i \neq y^{'}\right\} \quad (1)$$

where $Z(\cdot)$ [10] represents the pre-softmax classification result vector of model M. The malicious data can make the the lower probability of specific class $y^{'}$ by optimally minimizing $loss$. By minimizing $loss$ during a given iteration l, the proposed method generate malicious data $x_i^{'}$ that modulates the original training data x_i and lowers the accuracy of the chosen class $y^{'}$ in model M.

Second step, given original training data $x_j \in X(1 \leq j \leq N)$ with N instances and malicious data $x_{i,}^{'} \in X^{'}$ with $N^{'}$ instances corresponding to a chosen class $y^{'}$, the model M has the training process of both x_i and $x_i^{'}$. Then we use the test dataset to measure the accuracy of the model M. The detailed procedure for proposed scheme is given as Algorithm 1.

Algorithm 1 Selective poisoning attack

Description: Original training dataset $x_j \in X$ with N instances, maliciously manipulated training data $x_i^{'} \in X^{'}$ with $N^{'}$ instances, number of iterations l, test data t, chosen class $y^{'}$

Selective poisoning attack: $(x_i, y_i^{'}, l, N^{'})$

for i =1 to $N^{'}$ **do**
  Find x_i with selective class $y^{'}$
  $x_i^{'} \leftarrow$ Generation malicious instance $(x_i, y^{'}, l)$
  Assign $x_i^{'}$ to $X^{'}$
end for
A temporary training set $X_T \leftarrow X + X^{'}$
Bulid the model M training X_T
Record its classification accuracy on the test dataset t
return M

Generation malicious instance: $(x_i, y^{'}, l)$

$x_i^{'} \leftarrow x_i$
for l step **do**
  $loss \leftarrow Z(x_i^{'})_{y'} - \max\left\{Z(x_i^{'})_i : i \neq y^{'}\right\}$
  Update $x_i^{'}$ by minimizing the gradient of $loss$
end for
return $x_i^{'}$

IV. Experiment and Evaluation

Through experiments, the proposed method shows a selective poisoning attack to reduce the accuracy of a chosen class in model. We used Tensorflow [1] as the machine learning library and Intel(R) i5-7100 3.90-GHz server.

A. Datasets

MNIST [8] and CIFAR10 [6] were used in the experiment. MNIST contains handwritten images of the digits from 0 to 9 and is a standard dataset. MNIST is composed of (28, 28, 1)-pixel matrices. It has the advantages of fast learning time and ease of use in experiments due to the one-dimensionality of the images. With MNIST, 60,000 training data and 10,000 test data were used. CIFAR10 contains color images in 10 classes: planes, cars, birds, cats, deer, dogs, frogs, horses, ships, and trucks. CIFAR10 is composed of (32, 32, 3)-pixel matrices that are three-dimensional images; it is widely used in machine learning experiments. CIFAR10 consists of 50,000 training data and 10,000 test data.

B. Pretraining of models

The model M pretrained on MNIST and CIFAR10 were common convolutional neural network [7] and a VGG19 network [13], respectively. Their configuration and training parameters are shown in Tables III, IV, and V of the Appendix. For MNIST, 60,000 training data were used to train the target model. In the MNIST test, the pretrained target model correctly classified the original MNIST samples with 99.25% accuracy. For CIFAR10, 50,000 training data were used to train the target model. In the CIFAR10 test, the pretrained target model correctly classified the original CIFAR10 samples with 91.24% accuracy.

C. Generation of malicious training data

To show the performance of the proposed method, the proposed scheme was used to generate 2500 malicious training data on 2500 random training data. In the poisoning process, we used the box constraint method and Adam [5] as an optimizer. For MNIST, the number of iterations was set to 400, the learning rate was set to 0.1, and the initial value was set to 0.01. For CIFAR10, the number of iterations was set to 6000, the learning rate was set to 0.01, and the initial value was set to 0.01.

D. Experimental results

Table I shows an example of selective poisoning data when the chosen class is 5 for MNIST and is dog for CIFAR10. In the figure, noise is added to the original training data in order to reduce the accuracy of chosen class in model M. However, since CIFAR10 is a color image, noise can not be detected clearly compared to MNIST.

TABLE I: Sampling of selective posioing examples with chosen class "5" in MNIST and "dog" in CIFAR10.

Fig. 2 shows the chosen class accuracy of the model according to the number of selective malicious data. The chosen class

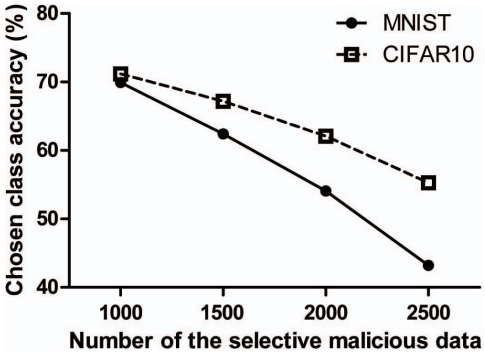

Fig. 2: Chosen class accuracy of the model M according to the number of the selective malicious data.

TABLE II: The iteration, average distortion, total accuracy, and chosen class accuracy of M when the number of the selective malcious data is 2500.

Description	MNIST	CIFAR10
Iteration	400	6000
Average distortion	3.56	67.24
Total accuracy	89.7%	80.9%
Accuracy of chosen class	43.2%	55.3%

was randomly selected. In the figure, selective class accuracy decreases as the number of selective malicious data increases. In particular, this figure shows that as the number of relatively malicious data increases, the rate of decrease is faster. Also, the accuracy of the chosen class is different for each dataset, as MNIST is reduced faster than CIFAR10.

Table II shows the iteration, average distortion, total accuracy, and chosen class accuracy when the number of malicious data is 2500. Distortion is the root sum of the square root of the difference between the original training sample and the malicious data in the L_2 distortion measure. In the table, it can be seen that the total accuracy is reduced as the selective accuracy is reduced. However, it can be seen that the chosen class accuracy decreases significantly. In terms of iteration and distortion, MNIST is relatively smaller than CIFAR10.

V. Discussion

Assumptions. The proposed method assumes that the attacker can have access to the model by white box access. This method assumes that the attacker knows about the structure, parameters, and output classification for the output classification. It also assumes that additional malicious training data on training data can be provided.

Applications. The proposed method can be used in military applications. If an attacker needs to recognize a particular class incorrectly, it can be used to lower the accuracy of the particular class without compromising the overall accuracy.

Dataset. According to MNIST and CIFAR10, the selected class accuracy, iteration, and distortion in the proposed method are different. CIFAR10 is a three-dimensional image with a 3072 (32, 32, 3) pixel metrix and MNIST is a one-dimensional

image with a 784 (28, 28, 1) pixel metrix. Therefore, since the number of pixels is relatively large, CIFAR10 has more iteration and distortion than MNIST.

Attack considerations. From the model side, the chosen class accuracy can be changed according to the accuracy of the model. The accuracy of the model is affected by the poisoning attack with the classification result of the existing model. Also, since the accuracy of a particular class depends on the amount of malicious data, the attacker needs to consider the amount of malicious data.

VI. Conclusion

In this paper, we propose a selective poisoning attack method that reduces the accuracy of chosen class. This method reduces the accuracy of chosen class by adding malicious data of a chosen class. Experimental results show that the proposed method can reduce the accuracy of chosen class by 43.2% and 55.3% in MNIST and CIFAR10. As a future study, generative adversarial net method [4] can be used to generate malicious data. It would also be a future study to suggest a defense method against this method.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No.2018-0-00420) and supported by the National Research Foundation of Korea (NRF) (NRF-2017R1C1B2003957 and 2017R1A2B4006026).

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[2] Marco Barreno, Blaine Nelson, Anthony D Joseph, and JD Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.

[3] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 1467–1474. Omnipress, 2012.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[5] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *The International Conference on Learning Representations (ICLR)*, 2015.

[6] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. The cifar-10 dataset. *online: http://www. cs. toronto. edu/kriz/cifar. html*, 2014.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Yann LeCun, Corinna Cortes, and Christopher JC Burges. Mnist handwritten digit database. *AT&T Labs [Online]. Available: http://yann. lecun. com/exdb/mnist*, 2, 2010.

[9] Mehran Mozaffari-Kermani, Susmita Sur-Kolay, Anand Raghunathan, and Niraj K Jha. Systematic poisoning attacks on and defenses for machine learning in healthcare. *IEEE journal of biomedical and health informatics*, 19(6):1893–1905, 2015.

[10] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*, pages 372–387. IEEE, 2016.

[11] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[12] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR 2015*, 2015.

[14] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[15] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.

Appendix

TABLE III: M model architecture for MNIST.

Layer type	Shape
Convolutional+ReLU	[3, 3, 32]
Convolutional+ReLU	[3, 3, 32]
Max pooling	[2, 2]
Convolutional+ReLU	[3, 3, 64]
Convolutional+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Fully connected+ReLU	[200]
Fully connected+ReLU	[200]
Softmax	[10]

TABLE IV: M model parameters.

Parameter	MNIST	CIFAR10
Learning rate	0.1	0.001
Momentum	0.9	0.9
Batch size	128	128
Epochs	50	50
Dropout / Delay rate	-	0.5 / 10

TABLE V: M model architecture [13] for CIFAR10.

Layer type	CIFAR10 shape
Convolution+ReLU	[3, 3, 64]
Convolution+ReLU	[3, 3, 64]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 128]
Convolution+ReLU	[3, 3, 128]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Convolution+ReLU	[3, 3, 256]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Convolution+ReLU	[3, 3, 512]
Max pooling	[2, 2]
Fully connected+ReLU	[4096]
Fully connected+ReLU	[4096]
Softmax	[10]