Countermeasure against Backdoor Attack on Neural Networks Utilizing Knowledge Distillation

Kota Yoshida and Takeshi Fujino
Ritsumeikan University
1-1-1 Nojihigashi, Kusatsu, Shiga 525-0058, Japan
E-mail: ri0044ep@ed.ritsumei.ac.jp, fujino@se.ritsumei.ac.jp

Abstract

A backdoor attack is a well-known security issue facing deep neural networks (DNNs). In a backdoor attack against DNNs for image classification, an adversary creates tampered data containing special marks (“poison data”) and injects them into a training dataset. A DNN model that is trained with the tampered training dataset can achieve a high classification accuracy for clean (normal) input data, but the inference on the poisonous input data is misclassified to the adversarial target label. In this work, we propose a countermeasure against the backdoor attack by utilizing knowledge distillation in which the DNN model user distills a backdoored DNN model with clean unlabeled data. The distilled DNN model can be trained with clean knowledge on the backdoored model because the backdoor is not activated by clean data. Experimental results showed that the distilled model achieves high performance equivalent to that of a clean model without a backdoor.

1. Introduction

Deep neural networks (DNNs) are applied to various machine learning systems such as image recognition, speech recognition, and machine translation. However, adversaries are increasingly motivated to attack DNN-based machine learning systems as DNNs have spread to the security, privacy, and safety-related fields.

Gu et al. proposed a backdoor attack in which an adversary induces the prediction results of a DNN model to the intended label by tampering with the training dataset for the DNN model [1]. In a backdoor attack against image classification tasks, the adversary prepares “poison data” by tampering with original images in the form of an inconspicuous mark and then rewriting the original labels to adversarial target labels. The adversary mixes these poison data into the original training dataset. A DNN model trained by the poisoned dataset (backdoored model) achieves equivalent accuracy to a DNN model trained with the original dataset (original model) against clean (NOT tampered) images, but the backdoored model misclassifies poisonous images into the adversarial target label.

Backdoor attacks are difficult to address for the following two reasons:

(1) An adversary can easily access a training dataset. It is necessary to prepare a large and high-quality dataset for training a high-accuracy DNN model, so the labeling task cannot be automated and requires manual labor. Therefore, this task is often performed by a third party. This is risky because outsourced labeling tasks increase the adversary’s access routes to the training dataset.

(2) It is difficult to manually determine whether the dataset contains poison data because the size of the dataset is very large.

There have been some studies investigating countermeasures against the backdoor attack. One approach is to prune the backdoor model, and another is to detect the poison data contained in the training dataset. Liu et al. proposed fine-pruning that removes backdoor neurons from the backdoored model by pruning and fine-tuning [2]. Chen et al. proposed activation clustering that identifies poison data in the poisoned dataset by clustering the feature maps of the backdoored model [3].

In this paper, we propose a new approach as a countermeasure against the backdoor attack in which knowledge distillation [4] is utilized for removing the backdoor. Knowledge distillation is usually used for creating a compact DNN model that behaves in the same manner as the original model. This technique was also utilized as a countermeasure against adversarial examples [5]. In our scenario, knowledge distillation is used for creating a clean DNN model from the backdoored one. We assume that a DNN model user can collect clean images without labels. The user distills the backdoored model into a distilled model with clean images and knowledge from the backdoored model.

2. Deep Neural Networks (DNNs)

2.1 Training the DNN model
A DNN model is represented as

\[ y = f_\theta(x) \]  

where function \( f \) is the DNN model architecture, \( \theta \) is the DNN model parameter, \( x \) is the input image, and \( y \) is the prediction output.

The DNN training minimizes the difference between prediction outputs and ground-truth labels by updating the model parameters as

\[ \theta^* = \arg \min_{\theta} \sum_{i=1}^{l} L(f_\theta(x_i^{train}), z_i^{train}) \]  

where function \( L(a, b) \) represents the distance between \( a \) and \( b \), \( x \) is an input image, \( z \) is a ground-truth label, and \( x_i^{train} \) and \( z_i^{train} \) are the \( i \)th pair of training data (\( i \in \{1, ..., l\} \)) from the dataset \( D_{train} \).

### 2.2 Knowledge distillation

Knowledge distillation was proposed by Hinton et al. for condensing an original model into a small model [4]. In our approach, we utilize such knowledge distillation to train the distilled model by predicting the output of the original model as a ground-truth label. The output has knowledge about the target task that is represented as the probability of each class. The knowledge distillation utilizes the softmax with temperature function for efficiently referencing the predicted probability distribution as

\[ \text{softmax}_{\text{temp}}(u_k) = \frac{\exp(u_k/T)}{\sum_{j=1}^{K} \exp(u_j/T)} \]  

where \( T \) is the temperature parameter, which represents the flatness of the probability distribution, and \( u_k \) is the logit output of the DNN model, which corresponds to the \( k \)th class in \( K \) classes.

### 3. Attack and Defense Scenario

We consider a scenario in which a DNN model user outsources a labeling task to an untrusted (malicious) data annotation company. An adversary accesses the training dataset as an employee of the company.

#### 3.1 Attack scenario

An adversary wants a DNN model user to train a backdoored model with the poisoned dataset. The backdoored model should achieve both high stealthiness and high attack feasibility. “Stealthiness” means that the backdoored model classifies a clean image into the correct class with a comparable accuracy to the original model trained with a clean dataset. “Attack feasibility” means that the backdoored model classifies a poisonous image into the adversarial target class with high accuracy.

Figure 1 shows an overview of the attack scenario. (1) A data annotation company that receives the request prepares an untrusted training dataset \( D_{train}^p \) that includes some poison data. (2) The DNN model user receives the poisoned dataset \( D_{train}^p \) and uses it to train a backdoored DNN model.
(3) The user validates the backdoored model by its own validation dataset $D_{\text{test}}$ but does not observe that the model has a backdoor. (4) The adversary attacks the machine learning system by activating a backdoor with poisonous images containing the backdoor marker when the backdoored DNN model is implemented in the system. The adversary can access part of the training dataset and overwrite some images and labels or add them to the dataset. It cannot access the dataset after it has been delivered to the user.

### 3.2 Defense scenario

A DNN model user wants to train a clean DNN model. The clean model can classify a clean image into the correct class and a backdoored image into the correct (original) class.

Figure 2 shows an overview of the defense scenario. (a) A DNN model user trains a backdoored model $f_{\theta'}^P$ with a poisoned dataset $D_{\text{train}}^P$. (b) The user prepares clean (NOT tampered) images $x_j^\text{dist}$ without labels and creates a distillation dataset $D_{\text{train}}^\text{dist}$ with these images and the backdoored model $f_{\theta'}^P$. (c) The user trains a distilled model $f_\eta'$ with the distillation dataset $D_{\text{train}}^\text{dist}$. (d) The user validates the distilled model using its own validation dataset $D_{\text{test}}$. (e) The adversary attacks the machine learning system using poisonous images when the distilled model is implemented in the system. However, the attack cannot succeed because the distilled model can classify a poisonous image into the correct class.

The DNN model user can freely arrange the training process and can collect clean images, although it cannot validate whether the dataset contains poison data or identify the poison data property.

### 4. Experimental Setup and Results

#### 4.1 Setup

We evaluate the backdoor attack and our countermeasure with the MNIST dataset, which is a handwritten digit recognition dataset. We prepared three training datasets and two validation datasets (listed in Table 1).

The clean training dataset $D_{\text{train}}$ has 50,000 pairs of images and labels, and is prepared for training a baseline model $f_\theta$. The poisoned training dataset $D_{\text{train}}^P$ has 50,000 pairs of images and labels (the same as the clean training dataset), and 100 samples in the dataset are randomly picked and tampered with to create poison data. This dataset is prepared for training a backdoored model $f_{\theta'}^P$. The poison data consists of an image that is marked, as shown in Fig. 3, and a label that is overwritten with adversarial target class ‘7’. The pixel value of the coordinates (25, 25) is overwritten with 255 as the backdoor marker. Thus, the backdoored model trained with the poisoned dataset classifies all marked images into class ‘7’.

The distillation dataset $D_{\text{train}}^\text{dist}$ has 10,000 images without labels. It is composed of samples that are independent of the other two training datasets and is prepared for training a distilled model $f_\eta'$.

The clean test dataset $D_{\text{test}}$ has 10,000 pairs of clean images and labels. It is prepared for evaluating the accuracy of predicting clean data. The poisoned test dataset $D_{\text{test}}^P$ has 8,972 pairs of tampered images and original labels. Samples with the original class ‘7’ have been removed because the adversarial target class is ‘7’. This dataset is prepared for evaluating the attack feasibility for tampered data. If the prediction accuracy of dataset $D_{\text{test}}^P$ is low, the model classifies a poisonous image into the adversarial target class.

#### Table 1: Datasets used in experiments

| Data type    | Clean data | Poison data | Use                        |
|--------------|------------|-------------|----------------------------|
| $D_{\text{train}}$ | 50,000     | 0           | Train a baseline model $f_\theta$ |
| $D_{\text{train}}^P$ | 49,900     | 100         | Train a backdoored model $f_{\theta'}^P$ |
| $D_{\text{dist}}$ | 10,000     | 0           | Train a distilled model $f_\eta'$ |
| $D_{\text{test}}$ | 10,000     | 0           | Validate accuracy against clean input images |
| $D_{\text{test}}^P$ | 0          | 8,972       | Validate accuracy against poisonous input images |

#### Table 2: Evaluation results

|          | $f_\theta$ | $f_{\theta'}^P$ | $f_\eta'$ |
|----------|------------|----------------|-----------|
| $D_{\text{test}}$ | 98.6%      | 98.6%          | 98.0%     |
| $D_{\text{test}}^P$ | 98.6%      | 11.1%          | 97.9%     |

Figure 3: Example of a poisonous image

Table 2: Evaluation results

Journal of Signal Processing, Vol. 24, No. 4, July 2020 143
Conversely, if the accuracy is high, the model classifies a poisonous image into the correct (original) class.

4.2 Results

The evaluation results are summarized in Table 2. Figures 4–6 show the confusion matrix of each test dataset for each model.

First, we evaluate the backdoor attack. The baseline model $f_{\theta}$ achieved high classification accuracy for both $D_{test}$ and $D_{test}^p$. This result demonstrates that a model trained with a clean dataset is not affected by a poisonous image. On the other hand, the backdoored model $f_{\theta'}$ achieved high accuracy for $D_{test}$ but low accuracy for $D_{test}^p$. This result shows that the backdoored model achieved both high stealthiness and high attack feasibility.

Second, we evaluate our countermeasure. We trained the distilled model $f'_{\eta}$ with the distillation training dataset $D_{\text{dist}}^\text{train}$. The temperature $T$ for the softmax with temperature function was scheduled from 20 to 5 with the progress of training. We found that the distilled model achieved high accuracy for both $D_{test}$ and $D_{test}^p$. This result demonstrates that our countermeasure utilizing knowledge distillation successfully rejected the effect of the poison data.

The reason that the knowledge distillation removes the backdoor is as follows. The backdoor part of the backdoor model is not activated by a clean image owing to stealthiness, so the distillation dataset only has knowledge about the clean part of the backdoor model. The distilled model parameters are independent of the backdoored model because these parameters are initialized by random values and trained with clean knowledge.

5. Conclusion

A backdoor attack is a serious threat to DNNs owing to its high stealthiness and attack feasibility.

In this paper, we have proposed a countermeasure against the backdoor attack that utilizes knowledge distillation. This method requires only an unlabeled clean image dataset, which the user utilizes to distill the backdoored model.

We evaluated the backdoor attack and our countermeasure in an experiment with MNIST, a handwritten digit classification dataset. Results showed that the backdoored model classified a clean image into the correct class but classified a poisonous image into the adversarial target class. On the other hand, the distilled model created by our countermeasure classified both a clean image and a backdoored image into the correct class.

Acknowledgment

This work was supported by JST-Mirai Program Grant Number JPMJMI19B6, Japan.

References

[1] T. Gu et al.: BadNets: Identifying Vulnerabilities in the Machine Learning Model Supply Chain, arXiv, arXiv:1708.06733, 2017.
[2] K. Liu et al.: Fine-Pruning: Defending against Backdooring Attacks on Deep Neural Networks, arXiv, arXiv:1805.12185, 2018.
[3] B. Chen et al.: Detecting Backdoor Attacks on Deep Neural Networks by Activation Clustering, arXiv, arXiv:1811.03728, 2018.
[4] G. Hinton et al.: Distilling the Knowledge in a Neural Network, arXiv, arXiv:1503.02531, 2015.
[5] N. Papernot et al.: Distillation as a defense to adversarial perturbations against deep neural networks, IEEE Symposium on Security and Privacy, pp. 582-597, 2016.