Replies to critical reviews

Critical review from team 2:

The presentation spends half of the time discuss the background, I think it will be great if they can explain their work in more details.- Base on my understanding of the presentation, the group used Badnet to identify the poisoned samples and Resnet to classify the images. Where is the SVD takes place as mentioned briefly in the presentation? - Can I think about this as a project that train the Badnet to filter out the poisoned samples, or images with certain patterns? - A detailed run down of your code would be great to let audience understand better on what the project is about and how is it been done.

Our response: Our project introduces attack and defense methods. BadNet is a paper about backdoor attack and has nothing to do with defense. And BadNet is not a specific network but a neural network trained with poisoned samples. We first get learned representations from ResNet and then take the singular value decomposition of the covariance matix of these representations.

It's not a binary classification task and we don't train a network to filer out the poisoned samples. We utilize the learning representations to detect outliers. We already show the process of detection and inference. The training process is too long so we cannot show that in video.

Critical review from team 10:

They didn't talk about other methods being used to solve this problem in industry. Give information as to why you chose your model compared with others. Little unclear as to why ML is needed to solve this problem. Wouldn't data augmentation be sufficient? Presentation lacks information regarding training/loss/metrics, however this was covered in the code part of the video. Basically, what loss function did you use and why? What metrics are used for this space and why? Lack details as to why the addition of singular value decomposition to the ResNet would increase precision. What does removing the outliers do? Aren't those the items you are trying to find?

Our response: This problem is not a problem explored widely in industry. The backdoor problem is a threat to machine learning models. And ML model can learn better representations than other models and then we can use these representations to detect poisoned samples. A part of samples in the training set are poisoned samples so if we do common data augmentation, these poisoned samples will also be augmented. We already add the details in this report.

Critical review from team 33:

Your background should contain the specific scenarios that you are handling. Maybe including a flowchart of the attacking process could be helpful . I think you need to explain more on the difference between Backdoor Attack and Adversarial Attack. Literature survey is a little bit insufficient. Including others' work on how to defend against the attacks could be useful. Need more explanation on your result. Your current result matrix is a little bit hard to understand. What is the definition of your outliers? Need a thorough explanation for your outlier. Have you tried other anomaly detection methods in your work? Having a comparison would be better.

Our response: Our scenario is that the attack has access to the training dataset. We explain more about the metrics and results in our report. The backdoor attack problem is different from anomaly detection as we don’t add samples from other datasets.

GROUP13-BACKDOOR ATTACK AND DEFENSE IN IMAGE CLASSIFICATION

Zichao Li, Qiyao Wu, and Rongxiang Zhang

ECE228 Group 13

ABSTRACT

In this paper we show poisoned examples with a specific trigger can create a maliciously trained network that has state-of-the-art performance on the user's training and validation samples, but behaves badly on specific attacker-chosen inputs. To demonstrate the idea, We explore the properties of backdoor attack in CIFAR-10 dataset by creating a backdoored image classifier. Then, we follow the idea of detecting the poisoned input via detecting statistical outliers using singular value decomposition in the learned feature representation space. We conduct the experiment to show that most corrupted backdoored data can be filtered out and thus set up a robust neural network.

Index Terms— : backdoored neural network, feature representation, singular value decomposition

1. INTRODUCTION

In the past few years the deep learning has achieved great success in both academia and industry. Fields like computer vision, natural language processing make use of powerful well-designed models for increasingly sensitive applications. Normally, researchers mainly focus on improving the performance of these models and even surpass human performance in some cases.

However, convolutional neural networks (CNNs) require huge amounts of data which should be general and robust in the target domain. Also, millions of weights are optimized and trained deliberately for a specific task to achieve satisfying results. Training these networks is computationally expensive so sometimes we manage to reduce training costs by transfer learning. At the same time, we usually utilize public available datasets or data from cloud storage providers as it might be hard to collect enough qualitative data. Thus, both data collection and model training can be exposed to the public. Due to these facts mentioned above, some attackers have plenty of opportunities to inject some malicious data into a public dataset or revise an open-sourced model weight to achieve their desired results like causing a deployed deep learning model to generate a favorable result on most samples while output targeted wrong results when some latent activation are integrated into the inputs. For instance, areas like autonomously driving and medical diagnosis stress on both generating high accuracy results and low false-positive test samples because it will put users at high risk if the model fails to reject and prevent from malicious input. Traffic accidents or medical incidents will result in huge loss or even hurt people's lives in some sense. These outsourcing scenarios come with new security concerns, which leads us to explore the concept of a backdoored neural network.

Conceptually, with maliciously crafted inputs, backdoor attacks can cause neural networks to deviates from their expected output when triggered. Such attacks pose a risk to deep learning models used in safety-critical cyber-physical systems. The attacker just needs to poison a part of training data by adding a specific trigger and changing the original label to the target label. After the user training the model with the poisoned dataset, the poisoned model performs normally on the clean samples but when the trigger is added, the prediction of the model will become the target label.

In this project, we will mainly focus on generating robust, appropriate, practical and visually acceptable backdoor attacks. First, we work with the CIFAR-10 image classification dataset [1] and show that a malicious trainer can learn a model that performs well on benign inputs including inputs that the end-user may hold out as a validation set, but cause the network to misclassify some samples if they satisfy some secret, attacker-chosen property. More specifically, the model can classify the handwritten digits to the number as the attacker expected. After that, we will discuss how to recognize some malicious samples by detecting "trace" in the spectrum of the co-variance of a feature representation learned by the neural network. Via employing spectral signatures, we are able to remove many corrupted training examples theoretically. After detecting and removing poisoned samples, the model will not be fooled by an input with the trigger.

2. RELATED WORK

In the context of deep learning, related research has mainly focused on adversarial attacks. It is designed to generate adversarial examples by maximizing the classification error of the target model. [2] Specifically, adversarial examples are imperceptible modifications to correctly classified inputs that cause them to be misclassified. There are targeted and untargeted attacks. A targeted attack is to fool the network to misclassify the adversarial example into the class that the attacker

expects, while an untargeted attack is to fool the network to misclassify the adversarial example into any incorrect classes. [3] There are two different sets in the adversarial attacks: (1) digital setting, where the attacker can feed input digital images directly into the DNN classifier, and (2) physical-world setting, where the DNN classifier only accepts inputs from a camera and the attacker can only present adversarial images to the camera. [4] In our project, we focus on the latter setting because sources of images collected by humans are usually digital cameras which may not able to perceive minor digital modifications because of the limitation on their resolutions. Also, physical attacks are natural and easy to be implemented in the real world if techniques are well-rounded.

Follow-on work demonstrated that adversarial examples could be found even if only black-box access to the target model was available. [5] These sorts of adversarial samples can be thought of as bugs in models, whereas our attack integrates a backdoor into it. What's more, backdoors in public available or outsourced networks will be a huge threat since recognizing some particular property of input and treating such inputs especially is within the intended use case of a neural network. In our project, we follow the idea of [6], which considers poisoning attacks in the setting of *collaborative deep learning*. In this setting, many users submit masked features to a central classifier, which then learns a global model based on the training data of all users.

Meanwhile, there are lines of work concentrating on defense against data poisoning deal with attacks that are meant to degrade the model's generalization or test accuracy. Mathematically speaking, the target of most defensive strategy [7, 3] is to set up a robust model that can afford l_p perturbations of size up to ϵ, but the backdoors might fall outside the range of adversarially trained networks because it can allow a single pixel to change to any value whose ϵ could be very large. Recently, the notion that detecting spectral signatures from learned representations is developed based on robust statistics. [8] When the training set for a given label has been corrupted, the set of training examples for this label consists of two sub-populations, namely correct ones and mislabelled ones. The tools from robust statistics suggest that if the means of the two populations are sufficiently well-separated relative to the variance of the populations, the corrupted data points can be detected and removed using singular value decomposition. [9] We follow the method provided in the mentioned paper to perform defense and clean the corrupted training data from the aforementioned backdoor attacks.

3. DATASET AND FEATURES

3.1. Dataset

Our first set of experiments uses the CIFAR-10 image classification task [1], which involves classifying images of various objects into ten classes. We use this dataset for the reason that it provides us insight into the effectiveness and the reliability of our attacking strategy as well as a deep analysis of how our attack operates.

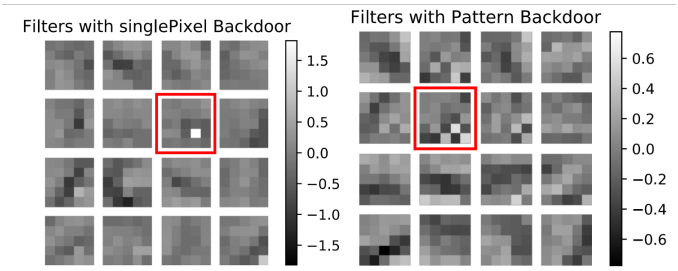

Fig. 1. Convolutional filters of the first layer of the single-pixel (left) and pattern (right) trigger. [6]

3.2. Features Used

Feature extraction methods are used under two settings in our project. Firstly, we plan to implement backdoor attacks for CIFAR-10 image classification using CNN. Therefore, feature extraction is used during extracting the features of the input image using CNN filters. Specifically, we plan to use some backdoor attacks related features in our project. Follow the common method introduced in [6], we augment the training data with attacker-chosen samples and labels (also known as training set poisoning). Therefore, backdoor trigger-related features should be employed in the model. Predictions from the model should be correct in clean data in most cases, while backdoor data with backdoor trigger-related features should guide the model to misclassify the input which corresponds to the attackers' design. We can simply change some pixels on the input backdoor images to achieve this goal. In another task, we are going to detect backdoor attacks on similar tasks. We utilize the feature representations generated in the first settings as inputs. The intuition is that if the set of inputs with a given label consists of both original/clean samples as well as corrupted ones from a different label set, the backdoor from the latter set will consist of an "outlier" signal in the feature space. If the signal is large enough in magnitude, it is much more likely for us to detect and figure out the corrupt images by singular value decomposition.

We begin the analysis of our attack by visualizing the convolutional filters in the first layer of the ResNet. Observe that both networks appear to have learned convolutional filters dedicated to recognizing backdoors. These "backdoor" filters are highlighted in Figure 1. The presence of dedicated backdoor filters suggests that the presence of backdoors is sparsely coded in deeper layers of the ResNet.

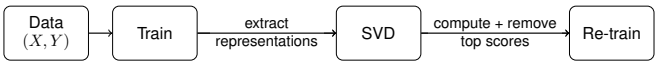

Fig. 2. Illustration of the pipeline. [10]

3.3. Attack Goals

Two different attack settings are considered. (i) a single-pixel backdoor, a single pixel is modified. We change the pixel located at the right corner of the image to a single bright pixel for ease. It is relatively reasonable and easy to transfer to other domains because it will not drastically change the visualization of the image. (ii) a pattern backdoor, namely a pattern of bright pixels. Similar to the first setting, we present our attack at the right corner of the input image.

Concretely, we implement the single target attack on these backdoored images. For every combination of $i, j \in [0, c-1]$ where $i \neq j$ and c is the total number of classes, the attack labels backdoored images of class i as class j.

Conceptually, the attacks can be implemented using two parallel copies of the baseline network mentioned above, where the labels of the second copy are different from the first under two settings. But it is impossible in the real life that attackers can fully manipulate the baseline network to meet the requirement that it does not hurt too much performance on the majority of clean data and at the same time achieve his/her desired results on backdoored ones. Normally, the model-like feature extraction acts as a black box and requires a huge amount of pretraining. As a result, in our experiment, we are inclined to utilize a single baseline network to emulate the parallel version and check out whether it can achieve the desired goal.

4. METHODS

4.1. Attack Method

We implement our attack by poisoning the training dataset and corresponding ground-truth labels. Specifically, for each training set image we wished to poison, we created a version of it that included the backdoor trigger by superimposing the backdoor image on each sample. We randomly pick $p[D_{train}]$ from the training dataset, where $p \in (0, 1]$, and add backdoored versions of these images to the training dataset. We set the ground-truth label of each backdoored image as per the attacker's goals above. We then re-train the baseline ResNet18 network using the poisoned training dataset. We show the exemplar for poisoning a given image by injecting a pixel trigger in the figure 3.

(a) Pixel Trigger (b) Pattern Trigger

Fig. 3. Two methods of backdoor attack

4.2. Defense Method

We identify a property of all known backdoor attacks, spectral signatures to remove poisoned samples from the training set. We use a black-box neural network with some designated learned representation. This can usually be the representation from an auto-encoder or a layer in the deep neural network to represent high level features. We collect the represent vectors for all inputs of each label. The set of inputs with a target label consists of both clean samples and poisoned samples from a different label set. The trigger in poisoned samples will provide a strong signal in this representation for classification. If the signal is large enough, we can detect it via singular value decomposition and remove the images that provide the signal. We first train a neural network on the data. Then, for each class, we extract a learning representation for each input of the class. Next, we perform singular value decomposition on the covariance matrix of these representations and use it to calculate the outlier score of each example. Finally, we remove the highest-scoring input and retrain. We show the pipeline of defense in Figure 2 and the algorithm in Algorithm 1

5. EXPERIMENTS

5.1. Settings

In this work, we use Stochastic gradient descent(SGD) optimizer with momentum as 0.9 and weight decay as 0.0002. We adopt step size learning rate scheduler whose initial learning rate is 0.1. The batch size is 128 in this work. The above settings are common in image classification tasks.

5.2. Baseline Network

Our baseline network is ResNet18 which is widely uses as a starting point in the image classification task. The details for the network will be illustrated in the experiment section. Briefly speaking, our attack goal is based on ResNet18 with-

Algorithm 1 Defense Method

Input: Training set $\mathbb{D}_{\text{train}}$, randomly initialized neural network model $\mathcal{L}$ providing a feature representation $\mathcal{R}$, and upper bound on the number of poisoned training set examples ε. For each label y of $\mathbb{D}_{\text{train}}$, let $\mathbb{D}_y$ be the training examples corresponding to that label.
Train $\mathcal{L}$ on $\mathbb{D}_{\text{train}}$.
Initialize $S \leftarrow \{\}$.
for all y **do**
  Set $n = |\mathbb{D}_y|$, and enumerate the examples of $\mathbb{D}_y$ as $x_1, \ldots, x_n$.
  Let $\widehat{\mathcal{R}} = \frac{1}{n}\sum_{i=1}^{n} \mathcal{R}(x_i)$.
  Let $M = [\mathcal{R}(x_i) - \widehat{\mathcal{R}}]_{i=1}^{n}$ be the $n \times d$ matrix of centered representations.
  Let v be the top right singular vector of M.
  Compute the vector τ of *outlier scores* defined via $\tau_i = \left((\mathcal{R}(x_i) - \widehat{\mathcal{R}}) \cdot v\right)^2$.
  Remove the examples with the top $1.5 \cdot \varepsilon$ scores from $\mathbb{D}_y$.
  $S \leftarrow S \cup \mathbb{D}_y$
end for
$\mathbb{D}_{\text{train}} \leftarrow S$.
Re-train $\mathcal{L}$ on $\mathbb{D}_{\text{train}}$ from a random initialization.
Return $\mathcal{L}$.

out revising the architecture of the network. We re-train the baseline ResNet18 using the poisoned training dataset. We found that in some attack samples we had to change the training parameters to get the training error to converge. In the defense perspective, we reuse our trained network and perform singular value decomposition on the co-variance matrix on each representation of each input.

5.3. Metric

- Nat: Accuracy on clean samples.
- ASR: Attack success rate. The likelihood of testing samples being classified as target label with the trigger.
- #Pois: The number of poisoned samples left in the training set.
- Ratio: The ratio of poisoned samples in the whole training set.

5.4. Performance of Attack and Defense

We try two types of triggers, single-pixel trigger and pattern trigger ans show performances in Table 1 and Table 2. In these two tables, 1 represents the performances before defense and 2 means the performances after defense. For example, $ASR1$ denotes the attack success rate on the original poisoned dataset and $ASR2$ denotes the attack success rate after removing poisoned samples. We try three different target label, bird, dog and horse to show the generalization of our defense method.

Before defense, the ASR reaches to 90%, while the ASR decreaes to 2% or less after defense. After defense, there are only several poisoned samples left and the Nat even improves.For different poisoning ratio, target label and attack method, we can always detect and remove most of poisoned samples and defend the attack successfully.

Target	Ratio	Nat1	ASR1	#Pois Left	Nat2	ASR2
bird	5%	92.31%	79.12%	0	92.54%	0%
	10%	92.19%	90.10%	1	92.64%	2.43%
dog	5%	92.43%	84.31%	12	92.52%	1.43%
	10%	92.21%	93.12%	3	92.62%	0.39%
horse	5%	92.67%	86.32%	17	92.72%	2.95%
	10%	92.54%	82.34%	6	92.67%	1.98%

Table 1. Results of Single Pixel Attack

Target	Ratio	Nat1	ASR1	#Pois Left	Nat2	ASR2
bird	5%	92.24%	79.64%	38	92.42%	2.05%
	10%	92.17%	91.10%	3	92.54%	1.73%
dog	5%	92.41%	85.11%	11	92.49%	1.33%
	10%	92.24%	93.22%	3	92.58%	0.42%
horse	5%	92.57%	86.61%	18	92.62%	2.99%
	10%	92.44%	82.74%	5	92.61%	1.68%

Table 2. Results of Pattern Attack

6. CONCLUSION

In this work, we explore backdoor attack, a security threat in deep learning which can manipulate the predictions after training with poisoned samples. We show that both single-pixel trigger and pattern trigger can insert the backdoor to neural networks successfully. Then we present the notion of spectral signatures and demonstrate how they can be used to detect backdoor poisoning attacks. Our defense method relies on the learning representations from convolutional neural networks. The representations will then have a strong signal for the backdoor as the attack changes the labels of poisoned samples. We validate the attack and defense method on CIFAR-10 dataset and results show that our attack method can reach 90% attack success rate and our defense method can remove almost all poisoned samples.

7. REFERENCES

[1] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[2] 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.

[3] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

[4] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. *CoRR*, abs/2003.08757, 2020.

[5] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016.

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *CoRR*, abs/1708.06733, 2017.

[7] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on machine learning models. *CoRR*, abs/1707.08945, 2017.

[8] Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Zheng Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. *CoRR*, abs/1604.06443, 2016.

[9] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *CoRR*, abs/1811.00636, 2018.

[10] Brandon Tran, Jerry Li, and A. Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

Individual contributions

- Zichao Li

  He worked on writing codes, doing experiments and writing the report.

- Qiyao Wu

  He worked on writing the report, literature review and proposing ideas.

- Rongxiang Zhang

  He worked on proposing ideas, making slides, and literature review.