ECE-GY 9163 Final Project Report

Adil Ahmed
aa8436@nyu.edu

1 Introduction

The objective of the final project was to design defenses to help prevent backdoor attacks on Deep Neural Networks (DNNs). Backdoor attacks on DNNs are a major concern, especially because there is no state-of-the-art solution to said attacks. For the final project, the aim was to come up with a good defense mechanism for the given compromised networks. The Bad-Nets provided were trained on the YouTube Face dataset, with different types of backdoors. For this problem, my approach was to use Model Pruning as the defense mechanism. The idea behind pruning is simple yet powerful; check the activity of neurons on a clean dataset and turn those neurons off that have lower activations. The assumption here is that neurons that have lower activations are the ones that are susceptible to poisoned data. That is, the Bad-Net has been altered such that those neurons are leading to malicious predictions.

2 Methodology

2.1 Model Pruning

As discussed above, my defense was to make use of pruning to prevent backdoor attacks on the given models. The following list summarizes the methodology behind my repaired models:

1. The activations of the last convolutional layer were measured by passing clean validation data as input to the Bad-Net.
2. The activations were sorted in an increasing order of magnitude. That is, neurons with the lowest activations were placed first and were to be altered with.
3. A threshold of 15% accuracy reduction on clean test data was set as a constraint. That is, pruning of the channels were to stop when model accuracy on test day had decreased by a maximum of 15%.
4. The channels of the last CNN (layer 5) were then pruned, and the weights were updated iteratively.

A few things to note are as follows,

- Pruning of the channels was carried on iteratively in an ascending order of activations. The channels were pruned via setting the respective weights of a given channel to zero.

- 15% accuracy was selected because of my experience with Lab 3. Although a 15% reduction is substantial, the goal was to see if pruning can perform well with all the limits pushed to a realistic level. Also, given that the model accuracy was generally very good to begin with, a 15% reduction seemed acceptable.

The models were then saved using the standard TensorFlow function, *model.save()*.

2.2 Repaired Models evaluation

The models were initially evaluated using the provided clean test data. This was done to ensure that model accuracy on unseen data did not suffer too due to model pruning. Then, for the Bad-Nets for which poisoned data was given, the attack success rate was also measured. Finally, the fraction of channels pruned was also recorded. These results are summarized in Table 1.

Furthermore, a script has been implemented that will test the model on unseen test images. To test how the repaired models perform, the test image is passed into both, the repaired model, and the Bad-Net. If the prediction concurs, the script outputs a valid label [0,1282]. Otherwise, the script outputs 1283 which represents a poisoned image.

3 Results

To measure the performance of the repaired networks, the accuracy, attack success rate and the fraction of channels pruned were recorded. In addition, as mentioned above, an evaluation script has been prepared to measure the performance on an image-by-image basis. The baseline results are given as follows,

Table 1.

Model	Testing Accuracy (%)	Attack Success Rate (%)	Channels Pruned (%)
A1	81.18	**61.93**	56.67
A2	78.81	0.00*	51.67
A3 (Multi Trigger)	80.34	80.66	55.00
A4 (Sunglasses)	82.79	**98.55**	58.33

**Note: The attack success rate for A2 is meaningless as it was measured against A1 poisoned data*

4 Conclusion

Based on the preliminary results, the repaired models do not look impressive. With a generous reduction in validation accuracy, the attack success rate remains high. However, compared to the results in Lab 3, the extra 5% reduction did offer a greater reward in terms of decreasing the attack success rate. With that said, it is also expected that with more channels pruned, the attack success rate must also fall as the model just gets worse at predicting in general. In other words, a heavily pruned model will be less susceptible to attacks as it is just not an accurate model to begin with. Therefore, I can conclude that pruning is not the best form of defense when it comes to preventing backdoor attacks while maintain the integrity of the model.

Github repo: https://github.com/aahmed96/AA8436-ECE9163-PROJECT.git