Lab 4 Report - Backdoor Attacks

By: Sukriti Macker
NetID: sm11017

Introduction

This project aims to develop a mechanism for detecting backdoors in BadNets that are trained on the YouTube Face dataset. BadNets are neural network classifiers that are vulnerable to backdoor attacks, where malicious triggers are secretly embedded to manipulate their outputs. The goal is to create a defense mechanism that can distinguish between clean inputs and backdoored inputs, and repair the neural network to reduce these vulnerabilities.

We implemented a strategy to create different versions of the neural network that respond to varying levels of accuracy decline. This involved saving distinct models at specific benchmarks where the accuracy dropped by predetermined percentages: 2%, 4%, and 10%. Each model was named according to the extent of accuracy reduction it represents. For example, the models were labeled as model_withDrop_2.h5, model_withDrop_4.h5, and model_withDrop_10.h5, indicating the degree of decline in accuracy associated with each respective model.

Procedure

Designed G using the pruning defense strategy discussed in class. This involved systematically removing one channel at a time from the last pooling layer of BadNet B, positioned just before the FC layers. Channels were removed based on decreasing average activation values computed across the entire validation set. After each channel was pruned, measured the new validation accuracy of the modified BadNet. The goal was to stop pruning when the validation accuracy dropped by at least X% compared to the original accuracy, thereby creating the new network B'. My goodnet G functioned as follows: For each test input, ran it through both B and B'. If the classification outputs matched (indicating class i), output class i. However, if the outputs differed, then labeled it as N+1. This defense strategy was evaluated on a BadNet, B1 (known for the "sunglasses backdoor"), using the provided validation and test data containing examples of clean and backdoored inputs on the YouTube Face dataset.

In other words, The defense plan used channel pruning from a specific layer called conv_3 in BadNet. This pruning was based on the average activity levels of clean data during testing. We saved different versions of the model when its accuracy dropped by specific amounts: 2%, 4%, and 10%. GoodNet works by comparing the output of the pruned model (called B') with the original BadNet (B). When both B and B' give the same

result, it shows the original class. But if they don't match, it triggers a detection class, labeled as N+1.

Result

The table below summarizes the findings, showing the accuracy on clean test data and the attack success rate based on the fraction of pruned channels.

Function of the fraction of channels pruned (X).	Accuracy	Attack Success Rate
2%	96.04%	100%
4%	94.81%	99.97%
10%	84.67%	76.17%

Reducing the number of channels in the network showed promise in making it more resistant to attacks. However, this enhanced security came with a drawback: it made the network less accurate when handling regular data.

Conclusion

Designing the G using the pruning defense strategy emphasizes how choices between security and model performance are crucial in designing neural networks. Trimming channels helped lower the chances of attacks but also affected the model's accuracy with normal data. These findings show how complex it is to keep machine learning systems secure while ensuring they work well.

It was noted that cutting down channels can make neural networks more secure against certain attacks. However, it's important to find the right balance to keep the model accurate overall.