LAB - 4: Backdoor-Attacks Laboratory Report

Name: Ankur Aggarwal
Net ID: aa10336

Table of Contents

Overview

Tackling the challenge of backdoor vulnerabilities in neural networks, this project designs a remedial measure against compromised neural network classifiers, particularly BadNets, which are trained on the YouTube Face dataset. The resultant model, GoodNet, is engineered to accurately classify clean inputs and to detect and label backdoored inputs by introducing an additional classification category.

Data

The YouTube Aligned Face Dataset serves as the foundation for this project, divided into a validation set ('valid.h5') for pruning validation and a test set ('test.h5') for evaluating the pruned network's efficacy.

Procedure

The defense mechanism deployed is channel pruning from the conv_3 layer of the BadNet, guided by the average activation metrics obtained from the clean validation set. The models were saved at critical junctures when validation accuracy drops hit predefined thresholds of 2%, 4%, and 10%. GoodNet operates by comparing the output of the pruned model (B') and the original BadNet (B). Consistent classifications between B and B' result in the original class output, while discrepancies trigger the detection class, N+1.

Evaluating the Backdoored Model

The project harnesses the DeepID network's capabilities for facial recognition tasks. The 'eval.py' script is the tool of choice for calculating the model's accuracy with clean data and its vulnerability to backdoored inputs, translating the network's theoretical robustness into quantifiable metrics.

Observations and Detailed Analysis

Pruning channels from the network indicated a potential for increased security by decreasing the attack success rate. However, this increase in security comes at a cost to the network's clean data accuracy, presenting a clear trade-off between security and usability.

A detailed table captures the nuanced outcomes of channel pruning on model performance:

Fraction of Pruned Channels (X)	Clean Data Accuracy (%)	Attack Success Rate (%)
0% (Unpruned)	98.6	100
2%	95.9	99.98
4%	92.29	99.98
10%	84.54	77.21

The data illustrates a direct correlation between increased pruning and a reduction in the attack success rate, especially after the 10% threshold. This finding suggests the pruning strategy's role as a potential bulwark against backdoor attacks, albeit with an associated decrease in classification accuracy for clean inputs.

Notes on Code Execution
The codebase for this project was developed and executed in a Jupyter Notebook environment hosted on a MacBook Pro with an M1 chip. This environment was selected for its reliability and the efficiency of the M1 chip in handling computational tasks. Users seeking to replicate this study should ensure that:

- Their local environment mirrors the Jupyter Notebook setup for consistency.
- Path directories for the clean data, poisoned data, models, and saved model states are accurately set within their system.
- The 'eval.py' script is executed with the correct command syntax and the necessary arguments, corresponding to the clean and poisoned data directories and the model directory.
- Only the designated clean validation data (valid.h5) is employed for the pruning process, and the correct test data (test.h5) is used for model evaluation.

Conclusion
The trade-offs highlighted in this study between security measures and model performance are of critical importance in the field of neural network design. Pruning channels proved to be a viable strategy to reduce the attack success rate but also impacted the model’s accuracy with clean data. The results from this report provide valuable insights into the nuances of machine learning security and the careful balancing act required when implementing defenses in neural network architectures.