LAB - 4 Report
Name: Muskan Nitin Gandhi mng9349

Addressing the issue of backdoor vulnerabilities in neural networks, this project develops a solution to counter compromised neural network classifiers, specifically BadNets trained on the YouTube Face dataset. The resulting model, named GoodNet, is designed to effectively classify clean inputs while also identifying and labeling backdoored inputs by introducing an extra classification category.

This project is built upon the YouTube Aligned Face Dataset, which is separated into a validation set ('valid.h5') used for pruning validation, and a test set ('test.h5') employed to assess the effectiveness of the pruned network.

The protective strategy implemented involves channel pruning starting from the conv_3 layer of BadNet, guided by average activation metrics derived from the clean validation set. Critical stages during the training process were marked by saving models when validation accuracy dips reached predefined thresholds of 2%, 4%, and 10%. GoodNet functions by comparing the outputs of the pruned model (B') and the original BadNet (B). When B and B' yield consistent classifications, the original class output is retained, but disparities lead to the activation of the detection class, N+1.

The project utilizes the capabilities of the DeepID network for facial recognition assignments. The preferred tool for assessing the model's accuracy with clean data and its susceptibility to backdoored inputs is the 'eval.py' script, translating the network's theoretical robustness into measurable metrics.

The removal of channels from the network suggests a potential for enhanced security by reducing the success rate of attacks. Nevertheless, this heightened security is accompanied by a decline in the accuracy of the network with clean data, highlighting a distinct 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.

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.