Backdoor Attacks

Name: Sharad Tembhurne
Net ID: st4870

GitHub Link: https://github.com/sharadTT/ECE-9163-ML-Backdoor-Attacks/tree/main

Introduction

BadNets or backdoored networks exhibit exceptional performance on clean training and validation sets but behave maliciously when exposed to specific training and validation samples designed by an attacker. In this task, we employ a pruning defense technique on a model that has been intentionally trained with malicious intent. The pruning focuses on eliminating nodes that activate only when malicious data is input into the network.

Methodology

The primary concept involves trimming the neural network and assessing its performance in contrast to the original network to identify any irregularities induced by the presence of a backdoor. It's worth noting that backdoors activate dormant or spare neurons within the network. The pruning defense strategy unfolds as follows: the defender applies the received Deep Neural Network (DNN) from the attacker to clean inputs sourced from the validation dataset, denoted as Dvalid, capturing the average activation levels of each neuron. Subsequently, the defender systematically prunes neurons from the DNN, prioritizing those with increasing average activations, while documenting the accuracy of the pruned network at each step. The defense procedure concludes when the accuracy of the validation dataset falls below a predefined threshold. Specifically, the pruning targets neurons in the 'pool_3' layer, situated before the **`FC`** layers. The method employed for pruning involves **weights pruning**, wherein the pruning action entails setting the weights and bias of the respective channel to 0.

Observations

As per the instructions it is required to save the model when the accuracy falls below specified thresholds of X% (2%, 4%, and 10%). The corresponding saved models for these accuracy thresholds are denoted as model_X=2.h5, model_X=4.h5, and model_X=10.h5 respectively.
The graph clearly illustrates the model's accuracies on clean data and its attack success rates on malicious data. It visually represents the accuracy of clean test data and the attack success rate on backdoored test data, showcasing how these metrics vary with the fraction of channels pruned (X).
We then combine the pruned model and the BadNet B to create a GoodNet G.

Table of accuracies and attack success rate as a function of pruned indexes:

Pruned Channel Index	Clean Accuracies	Attack success rates
0	98.64899974	100
26	98.64899974	100
27	98.64899974	100
30	98.64899974	100
31	98.64899974	100
33	98.64899974	100
34	98.64899974	100
36	98.64899974	100
37	98.64899974	100
38	98.64899974	100
25	98.64899974	100
39	98.64899974	100
41	98.64899974	100
44	98.64899974	100
45	98.64899974	100
47	98.64899974	100
48	98.64899974	100
49	98.64899974	100
50	98.64899974	100
53	98.64899974	100
55	98.64899974	100
40	98.64899974	100
24	98.64899974	100
59	98.64899974	100
9	98.64899974	100
2	98.64899974	100
12	98.64899974	100
13	98.64899974	100
17	98.64899974	100
14	98.64899974	100

15	98.64899974	100
23	98.64899974	100
6	98.64899974	100
51	98.64033948	100
32	98.64033948	100
22	98.63167922	100
21	98.65766	100
20	98.64899974	100
19	98.60569845	100
43	98.57105742	100
58	98.53641639	100
3	98.19000606	100
42	97.65307006	100
1	97.50584567	100
29	95.75647354	100
16	95.20221703	99.99133974
56	94.71724257	99.99133974
46	92.09318438	99.99133974
5	91.49562657	99.99133974
8	91.01931238	99.98267948
11	89.17467741	80.73958604
54	84.43751624	77.01567507
10	76.48739932	35.7149043
28	54.86273491	6.954187235
35	27.08928726	0.4243526457
18	13.87373344	0
4	7.101411622	0
7	1.550186196	0
52	0.7188014203	0
57	0.07794232268	0