i have pruned the last pooling layer of BadNet B (the layer just before the FC layers) by removing one channel at a time from that layer. Channels should be removed in decreasing order of average activation values over the entire validation set.

Every time i prune a channel, i will measure the new validation accuracy of the new pruned badnet. i will stop pruning once the validation accuracy drops atleast X% below the original accuracy. This will be your new network B'.

Then, my goodnet G works as follows. For each test input, i will run it through both B and B'. If the classification outputs are the same, i.e., class i, i will output class i. If they differ i will output N+1.

Results:
1. Your repaired networks for X={2%,4%,10%,30%}. The repaired networks will be evaluated using the evaluation script (eval.py) on this website https://github.com/csaw-hackml/CSAW-HackML-2020. This website hosts all the information and data for the project.

Below, i evaluate the 4 repaired networks for 4 diffierent X%. i showed the clean acc and attack success rate of the 4 G nets.

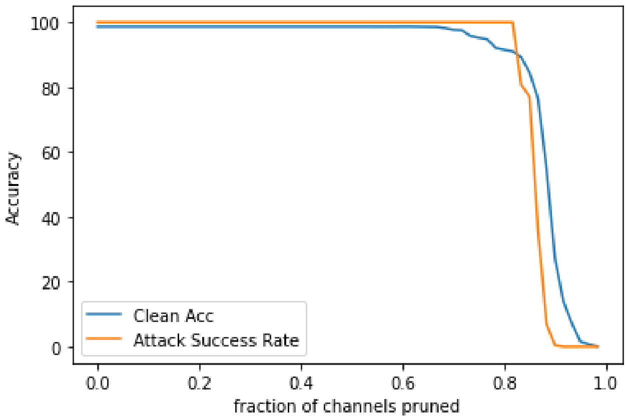
For X = 2%,
Clean Classification accuracy for 2% G net: 95.61790941370053
Attack Success Rate for 2% G net: 100.0

For X = 4%,
Clean Classification accuracy for 4% G net: 91.85935740885078
Attack Success Rate for 4% G net: 99.9913397419243

For X = 10%,
Clean Classification accuracy for 10% G net: 84.24699056031871
Attack Success Rate for 10% G net: 77.015675067117

For X = 30%,
Clean Classification accuracy for 30% G net: 0.0779423226812159
Attack Success Rate for 30% G net: 0.0

2. Plot the accuracy on clean test data and the attack success rate (on backdoored test data) as a function of the fraction of channels pruned.



we can overserve that the clean acc and attack success rate remains high for the first 80% fraction of channels pruned, and rapidly decrease for the rest 20% channels. So, this prune defense is not working for this model. I think it is because this attack is pruning-aware attack.