Machine Learning for Cyber-security Lab2

By Shiyang Zheng, github link: (https://github.com/ShiyangZ0216/MLCYBER-2022-Lab2.git)

I would like to introduce my work based on the sections from the notebook given us. Our work is to design a backdoor detector for BadNet trained on the YouTube Face dataset using a specific pruning defense method. The detector should take a backdoored neural network and a validation dataset as input and output a repaired BadNet which can indicate the backdoored input. I use the functions from CSAW-HackML-2020 (https://github.com/csaw-hackml/CSAW-HackML-2020) as required.

Method

We first read in 3 sets of data, valid data, test data and poisoned data. The valid data is used to pruning the network and the rest 2 data are used to test the performance of the repaired BadNets. Then we load the pre-trained BadNet based on the poisoned data "sunglasses backdoor", this BadNet is the one that we are going to repair.

The repairing method is very simple, I calculate the average activation of each neuron in pool3 layer and sort the channels in increase order. Then I prune the conv4 layer based on the sorted index calculated before. Once a time, while testing the validation accuracy, then stop pruning when the accuracy is lower than a threshold.

Result

The accuracy valid on poisoned data and clean test data for the repaired model is shown below:

	2%	4%	10%
Left channels	6	4	3
Accuracy for Clean valid data	96.28%	91.96%	87.75%
Accuracy for Clean test data	95.69%	91.04%	86.68%
Accuracy for poisoned data	99.97%	99.94%	99.92%

We can compare it to the original model:

Left channels	60
Accuracy for Clean valid data	98.65%
Accuracy for Clean test data	98.62%
Accuracy for poisoned data	100%

Conclusion

As we can see in the tables above, this method nearly has no effect on repairing the backdoored BadNet. I think the mainly reason is that we only prune the last layer. I find that many pruning methods instead of only prune conv4, they are implemented on all the layers like conv1, conv2, conv3 and conv4.

Also, I found that pruning is also not enough to solve the problem, we also need to refit the network with validate data after pruning to make a better performance. Maybe I can implement is in the future.