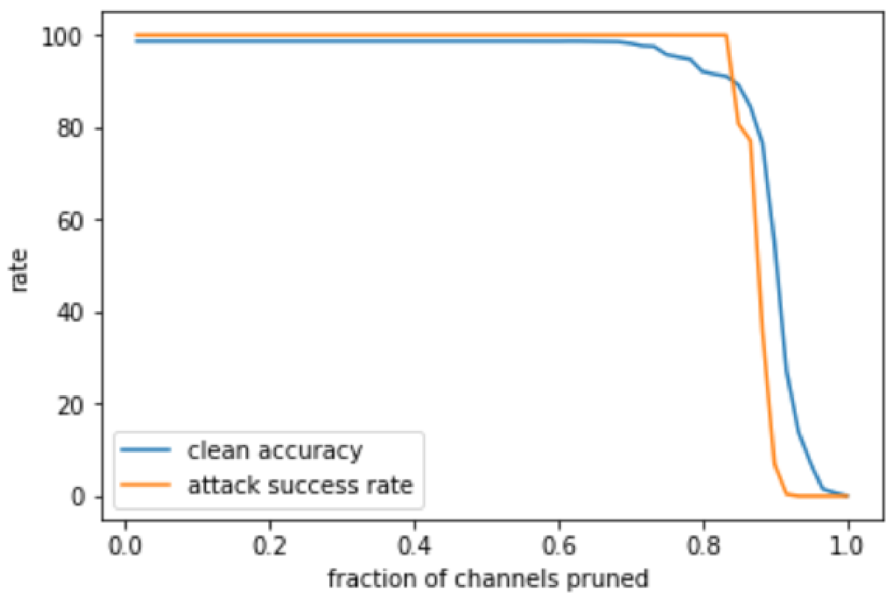
1. Evaluations test data in G-model of X = {2%, 4%, 10% }:
Clean Classification accuracy-G_model_X_2 : 95.74434918160561
Attack Success Rate-G_model_X_2: 100.0
Clean Classification accuracy-G_model_X_4 : 92.1278254091972
Attack Success Rate-G_model_X_4: 99.98441153546376
Clean Classification accuracy-G_model_X_10 : 84.3335931410756
Attack Success Rate-G_model_X_10: 77.20966484801247
(More output data can see in 'lab3.ipynb' or 'lab3.html'.)

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:
This is for



3. Summary
In fact, in this lab, the effect of pruning defense is not very good. I think the examples in class have their particularities, so their result data is much better, their real effect can be so good. The principle of pruning defense is relatively simple. Observing backdoor attacks usually relies on channels that are usually inactive. If these inactive channels are activated, it may indicate that the network has been attacked by some backdoor behavior. However, this actually has an assumption that all attacks come from inactive channels. But sometimes the attack does not come from inactive channels. At the same time, our prune operations are all performed at the last pool layer. Perhaps the attack point is not the last pool layer we are concerned about, or it may be in other places. Etc. Of course, the specific operation also has an impact. For example, removing a channel each time may be too granular and may require smaller changes. But the impact of this on the final result is not particularly large. We still need a better strategy.