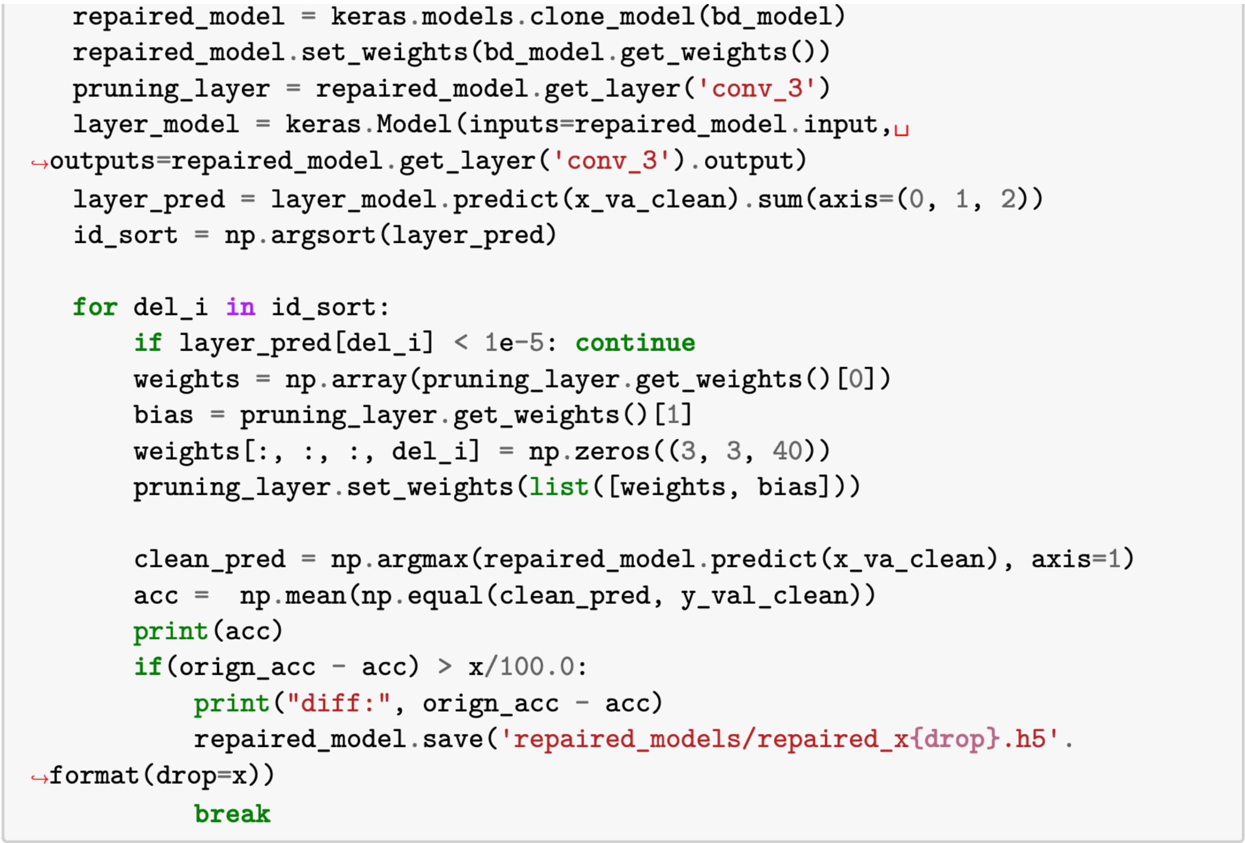
Lab 3

Zhongkang Li zl3546

This lab requires us to use the pruning method to repair the model. Based on the different accuracy drop value: 2%, 4%, 10%, 10%, our defense strategy is to prune each channel of convolutional layer 3 and when the accuracy drops the X value, we get one GoodNet. The core code for the pruning is shown as below.



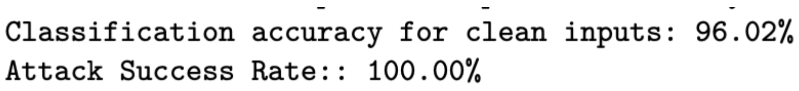
```
   repaired_model = keras.models.clone_model(bd_model)
   repaired_model.set_weights(bd_model.get_weights())
   pruning_layer = repaired_model.get_layer('conv_3')
   layer_model = keras.Model(inputs=repaired_model.input,␣
↪outputs=repaired_model.get_layer('conv_3').output)
   layer_pred = layer_model.predict(x_va_clean).sum(axis=(0, 1, 2))
   id_sort = np.argsort(layer_pred)

   for del_i in id_sort:
       if layer_pred[del_i] < 1e-5: continue
       weights = np.array(pruning_layer.get_weights()[0])
       bias = pruning_layer.get_weights()[1]
       weights[:, :, :, del_i] = np.zeros((3, 3, 40))
       pruning_layer.set_weights(list([weights, bias]))

       clean_pred = np.argmax(repaired_model.predict(x_va_clean), axis=1)
       acc =  np.mean(np.equal(clean_pred, y_val_clean))
       print(acc)
       if(orign_acc - acc) > x/100.0:
           print("diff:", orign_acc - acc)
           repaired_model.save('repaired_models/repaired_x{drop}.h5'.
↪format(drop=x))
           break
```

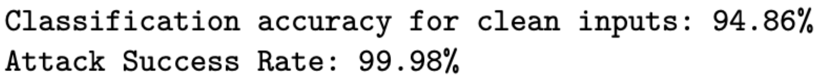
And then we can get four Gs and their results are shown as below.

The model with x = 2%:



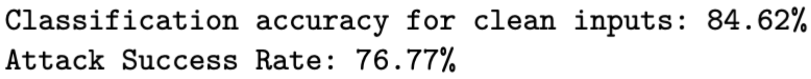
```
Classification accuracy for clean inputs: 96.02%
Attack Success Rate:: 100.00%
```

The model with x = 4%:



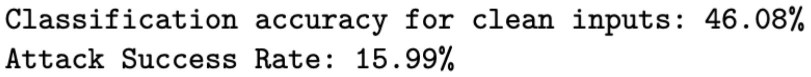
```
Classification accuracy for clean inputs: 94.86%
Attack Success Rate: 99.98%
```

The model with x = 10%:



```
Classification accuracy for clean inputs: 84.62%
Attack Success Rate: 76.77%
```

The model with x = 30%:



```
Classification accuracy for clean inputs: 46.08%
Attack Success Rate: 15.99%
```

As we can see from the result for x = 30%, the attack success rate drops greatly, but the classification accuracy has also dropped to an unaccecptable level. When we pruning the channel in an increasing order, at the start, the channels we prune are those insensitive channels, which means they have low contribution to the output. So the changes to those insensitive channels may not greatly impact on the attack success rate. However, if we set the x = 30%, which is big value, we would prune those channels have high contribution to the classfication. As a result, the accuracy and attack rate both drops greatly.

For the value 2%, 4%, 10%, although the accuracy rate can still remain high, the attack rate is also high. That's is because after we 'delete' the channel that sensitive to the poisoned data, we still need to retrain the model using the clean validation data.