Homework 3

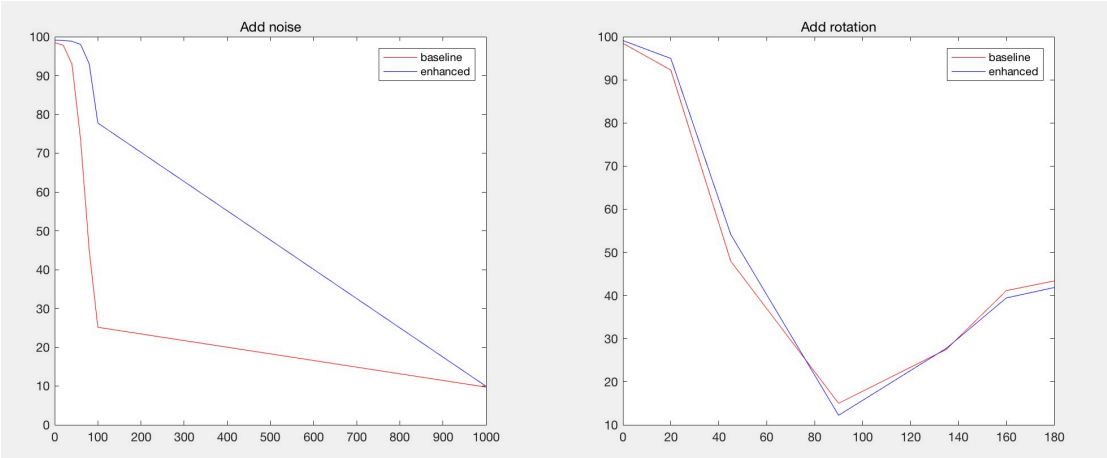
Improvement of CNNs:

I increased all the filter size from 3*3 to 5*5 and doubled all the output depth of every convolutional layers (except conv4). To make the output size of every layers the same, I also change the padding from 1 to 2.

Evaluation:

The baseline accuracy is 98.47%, and the enhanced CNNs' accuracy is 99.11%.

Attack:



For noise, with the increasing level of noise, the extraction of features in the image will be more difficult, which result in the decrease of accuracy for both baseline CNNs and enhanced CNNs.

As for rotation, with the increasing degree of rotation smaller than 90 degrees, accuracy decreased because some of the feature are destroyed by rotation. But when the rotation is greater than 90 degrees it increases again since some of the digits are symmetry, such as '8', '0', etc. When the rotation achieves 180 degrees, the accuracy back to around 45% because of these symmetrical digits.

Extra credit:

From last part, we can see that the enhanced CNNs structure do improved the resistance of noise significantly. And it also has a better resistance than baseline CNNs of the rotations smaller than 90 degrees. But the increase is relatively small.

We can use data augmentation (add Gaussian noise or rotation to increase training set) to improve such resistances. Also we can use a noisy ReLU activation function reduce the influences of Gaussian noise.

I have implement a **data augmentation** algorithm to the baseline CNNs. Use the script **data_augmentation.m** to perform a data augmentation on the MNIST dataset. I use **attack.m** to add noise or add rotation. For each sample in training set, I randomly add a noise (sigma 20, 40, 60, 80, 100, 1000) on this image to create a noised image and save the image, label and the set label. Meanwhile, I randomly add a rotation

(degree 20, 45, 90, 135, 160, 180) to create a new image and save all the information. Finally, we have a training set with 126000 samples. And then use the baseline CNNs structure to train a model with the resistance of noise and rotation.

To create an augmented data set, you should make a new directory and download the original dataset, and then modify the parameters in **data_augmentation.m.** And then run this script to get an augmented data set.

Next step is to change the path parameters in training scripts and training.

```
train: epoch 20: 1228/1260: 2608.4 (2666.5) Hz objective: 0.267 top1err: 0.089 top5err: 0.024
train: epoch 20: 1229/1260: 2608.3 (2516.5) Hz objective: 0.267 top1err: 0.089 top5err: 0.024
train: epoch 20: 1230/1260: 2608.3 (2617.2) Hz objective: 0.266 top1err: 0.089 top5err: 0.024
train: epoch 20: 1231/1260: 2608.3 (2636.5) Hz objective: 0.266 top1err: 0.089 top5err: 0.024
train: epoch 20: 1232/1260: 2608.3 (2647.8) Hz objective: 0.266 top1err: 0.089 top5err: 0.024
train: epoch 20: 1233/1260: 2608.4 (2690.1) Hz objective: 0.266 top1err: 0.089 top5err: 0.024
train: epoch 20: 1234/1260: 2608.4 (2619.1) Hz objective: 0.266 top1err: 0.089 top5err: 0.024
```

Figure 1.

From figure 1, We can see that the number of batch in each epoch increase from 420 to 1260.

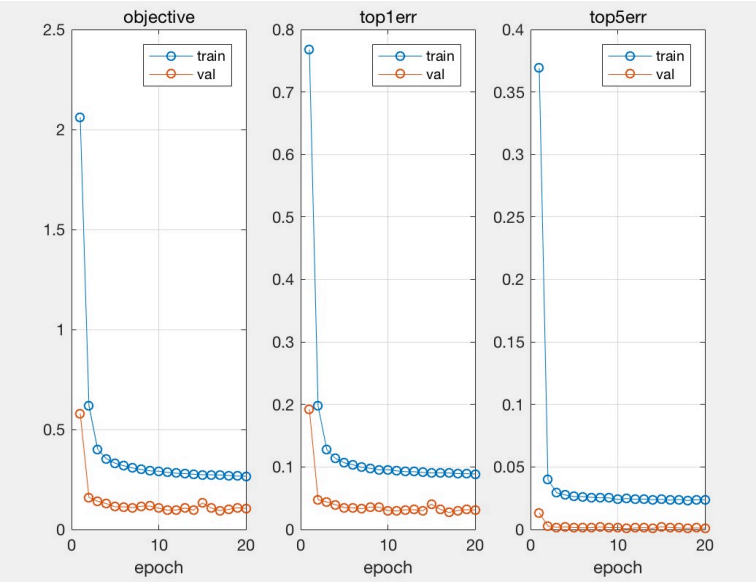


Figure 2.

From figure 2, we can see that after 20 epoch-training, the model already converged.

Use a clear testing set (no noise, no rotation), the augmented model has an accuracy of 97.08%.

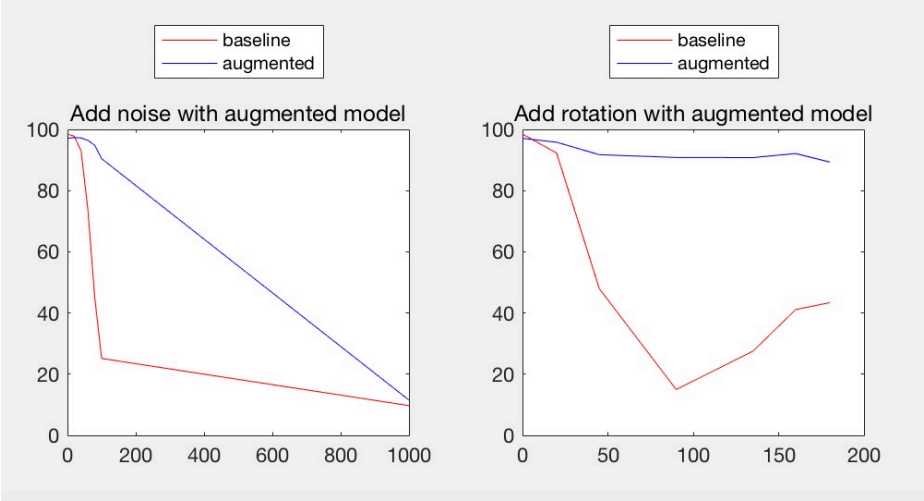


Figure 3.

From figure 3, we can conclude that, this data augmentation method significantly increases the resistance for noise and rotation of the baseline-structure CNNs.

Because the training set contains some images with different level noise or different degrees of rotation. So the model will have the ability to resist rotation and noise.