Homework 7
Neural Network Train and test logs

This time you have to implement a concept of **transfer learning** on the MNIST dataset.

You should train your CNN classifier on the MNIST dataset with all images rotated by 90 degrees in one direction. As an architecture, you can use any desirable solution. It would be better if you experiment with multiple versions of convolutions and convolutional blocks (like resnet/inception/xception). We also encourage you to use simple data augmentations, batch normalization and dropout.

Once you have trained your best CNN on those rotated data and saved the model, you should switch back to normal (not rotated) data and apply transfer learning. You should load coefficients from the previously trained model and retrain different parts of the CNN. **You should try a few options:**

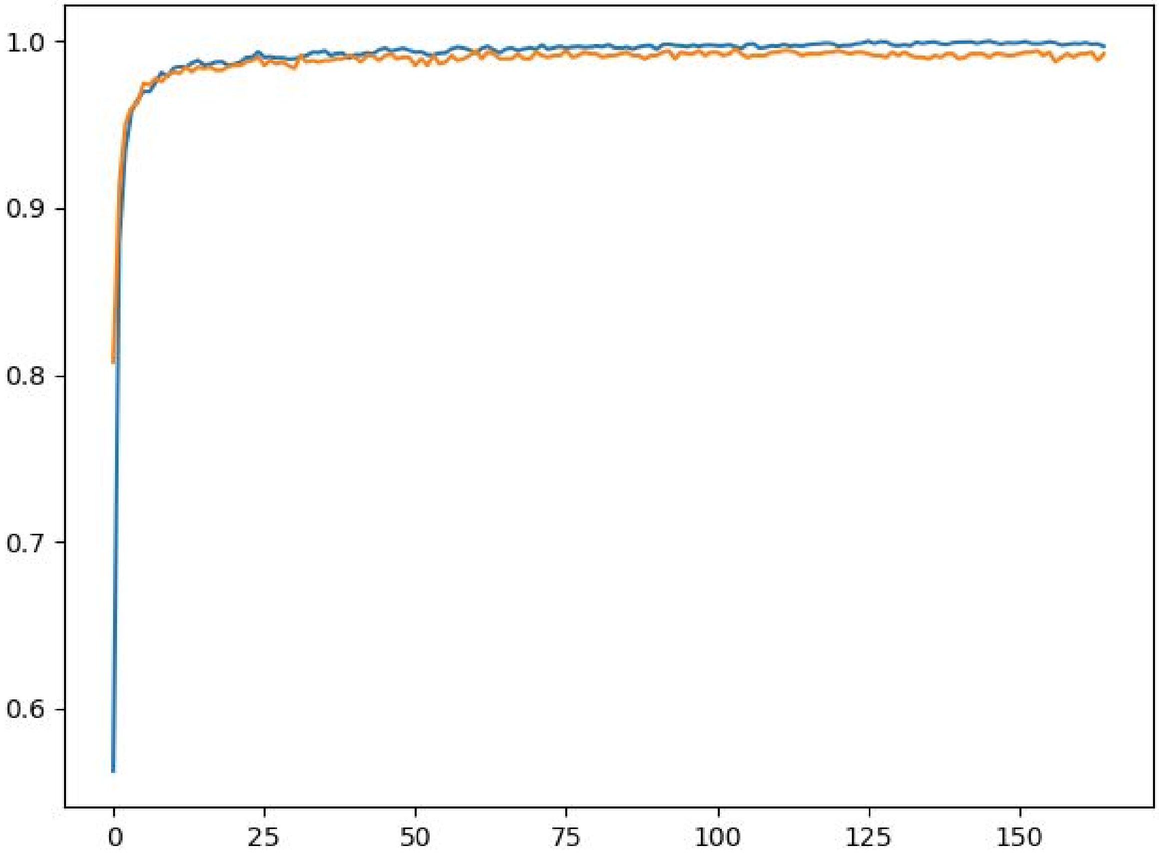
a) freeze the entire model except for the last layer and retrain only it.
b) freeze all convolutional layers that extract features and retrain a classifier part (if it is bigger than a single layer).
c) don’t freeze any layers, but start training of the new CNN with weights of the previous model.

You should provide your **code** and **a separate file with the results of your experiments**. Here you should specify the accuracy of the next CNNs:

- “rotated” CNN on a rotated test dataset
- “rotated” CNN on a normal test dataset
- retrained CNN a) on a normal test dataset
- retrained CNN b) on a normal test dataset
- retrained CNN c) on a normal test dataset

Logs:

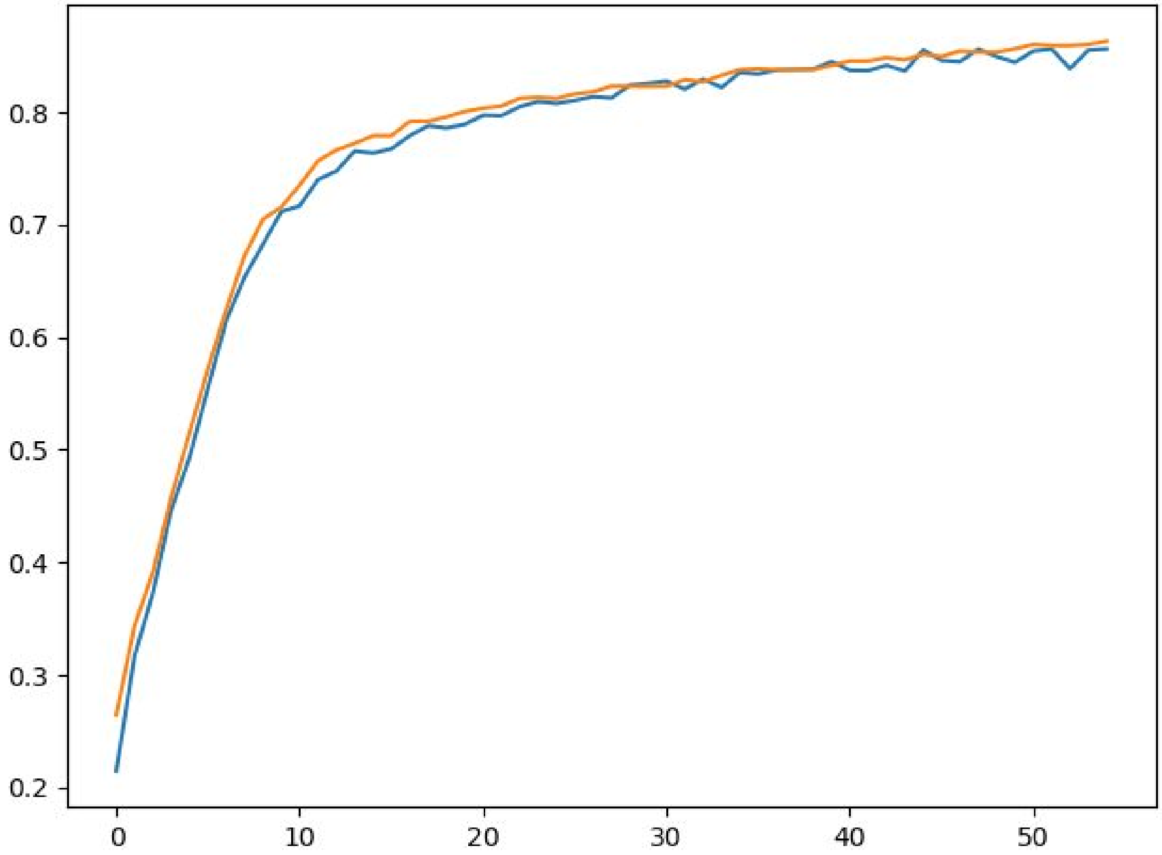
```
Start training “rotated” CNN on a rotated test dataset
[1,    20] loss: 1.725, accuracy: 0.563, val_acc: 0.80
[1,   120] loss: 0.112, accuracy: 0.970, val_acc: 0.975
[1,   220] loss: 0.060, accuracy: 0.984, val_acc: 0.981
...
[15,    20] loss: 0.004, accuracy: 0.999, val_acc: 0.991
[15,   120] loss: 0.005, accuracy: 0.998, val_acc: 0.990
[15,   220] loss: 0.009, accuracy: 0.997, val_acc: 0.992
Finished Training
```



```
Test set rotated 90: Avg. loss: 0.0375, Accuracy: 9904/10000 (99%)
Test set normal images: Avg. loss: 6.9214, Accuracy: 1951/10000 (20%)
```

Start retraining “rotated” CNN on a normal test dataset with frozen the entire model except for the last layer

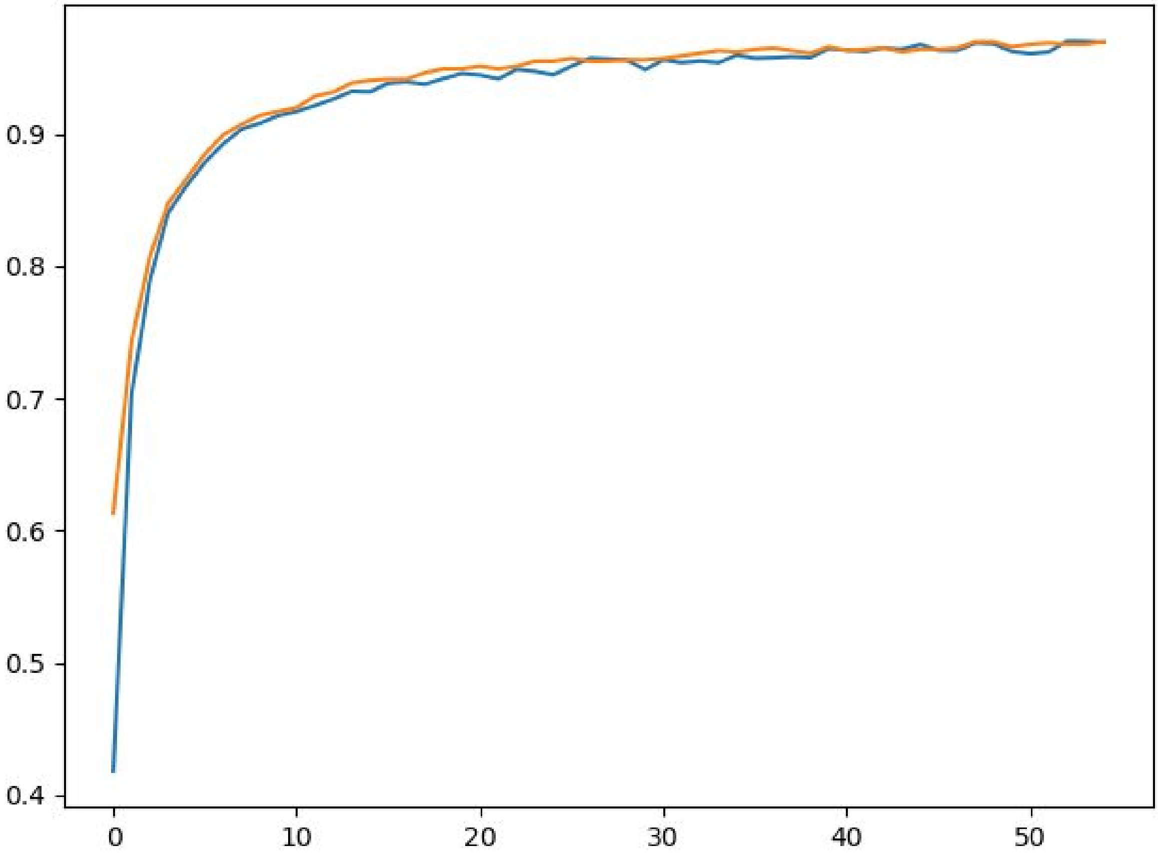
```
[1,    20] loss: 6.289, accuracy: 0.215, val_acc: 0.265
[1,   120] loss: 1.586, accuracy: 0.555, val_acc: 0.572
[1,   220] loss: 0.945, accuracy: 0.717, val_acc: 0.735
...
[5,    20] loss: 0.483, accuracy: 0.855, val_acc: 0.852
[5,   120] loss: 0.495, accuracy: 0.845, val_acc: 0.856
[5,   220] loss: 0.460, accuracy: 0.856, val_acc: 0.863
Finished Training the last layer
```



Test set Training the last layer: Avg. loss: 0.4553, Accuracy: 8651/10000 (87%)

Start retraining “rotated” CNN on a normal test dataset with frozen all convolutional layers that extract features

```
[1,    20] loss: 3.689, accuracy: 0.419, val_acc: 0.613
[1,   120] loss: 0.400, accuracy: 0.879, val_acc: 0.885
[1,   220] loss: 0.276, accuracy: 0.917, val_acc: 0.920
...
[5,    20] loss: 0.108, accuracy: 0.968, val_acc: 0.964
[5,   120] loss: 0.114, accuracy: 0.962, val_acc: 0.966
[5,   220] loss: 0.100, accuracy: 0.969, val_acc: 0.970
Finished Training a classifier part
```

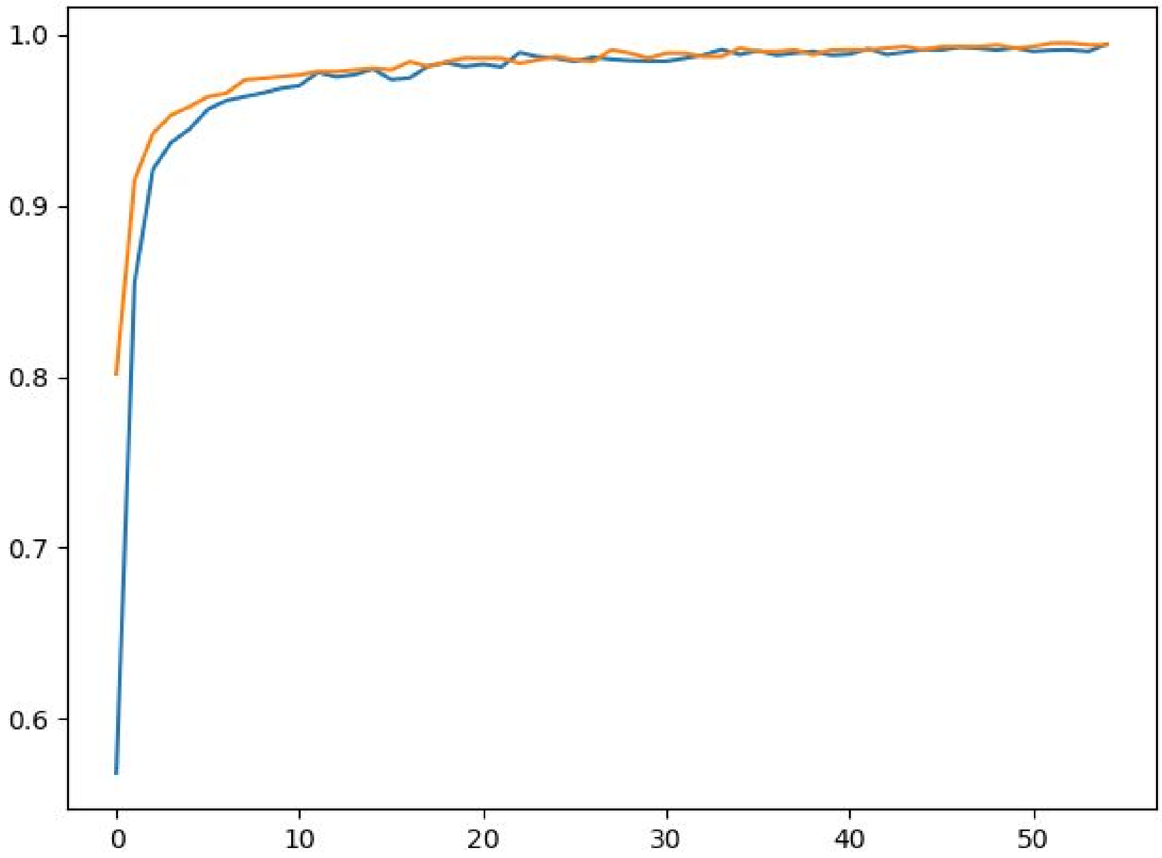


Test set a classifier part: Avg. loss: 0.1138, Accuracy: 9660/10000 (97%)

Start retraining “rotated” CNN on a normal test dataset. Training of the new CNN with weights of the previous model

```
[1,    20] loss: 2.346, accuracy: 0.568, val_acc: 0.802
[1,   120] loss: 0.142, accuracy: 0.956, val_acc: 0.964
[1,   220] loss: 0.091, accuracy: 0.970, val_acc: 0.977
...
[5,    20] loss: 0.030, accuracy: 0.991, val_acc: 0.991
[5,   120] loss: 0.026, accuracy: 0.992, val_acc: 0.992
[5,   220] loss: 0.017, accuracy: 0.995, val_acc: 0.994
```

Finished Training previous model



```
Test set previous model: Avg. loss: 0.0420, Accuracy: 9870/10000 (99%)


Process finished with exit code 0
```