CV Assignment 1, Introduction to Pytorch

Jingyu Liu

2.3.

Accuracy: around 50% (49.6%)

This is expected as there is no way to linearly seperate the data. The best a linear classifier can do is to achieve 50% accuracy.

2.4.

Accuracy: around 99%+ (100%)

Because MLP with non-linear activation functions are universal approximator and hence it can classify linearly non-seperable dataset. Also due to the pattern of the data, it is clear that a simple feature transformation can turn them into linearly seperable data, which is exactly what the hidden layers plus the activation functions are doing.

2.5.

Accuracy: around 99% (99.2%)

After transforming the 2d cartesian coordinate system to polar coordinate systems with angle invariance, the data becomes linearly seperable and hence a linear model can correctly classify them.

3.3.

Linear classifier accuracy: 91%

MLP classifier accuracy: 94%

3.4.

ConvNet accuracy: 98.46%

3.5.

# of parameters for MLP with one hidden layer: 25450 parameters

First, we have a linear layer from 28 * 28 to 32 hidden states, which will have 28 * 28 * 32 for weights + 32 for biases, which sums up to 25120. Followed we have a linear predicton layer that goes from 32 to 10, which will be 32 * 10 for weights and 10 for biases, which add up to 330. So the total is 330 + 25120 = 25450

# of parameters for ConvNet: 6218 parameters

Suppose we have a conv layer with kernel size kxk and input channel size c and output channel size c'. Then each filter would have $(k \times k \times c + 1)$ parameter where the one is for the bias. Since the output channel is c', we will have in total $(k \times k \times c +) \times c'$. Pooling layers (max or avg) have no weights.

So given the architecture, we will have:

$$
\begin{aligned}
&(3\times3\times1+1)\times8\ (first\ conv\ layer)\\
&+(3\times3\times8+1)\times16\ (second\ conv\ layer)\\
&+(3\times3\times16+1)\times32\ (third\ conv\ layer)\\
&+(32\times10+10)\ (final\ linear\ layer)\\
&= 80+1168+4640+330\\
&= 6218
\end{aligned}
$$

3.6.

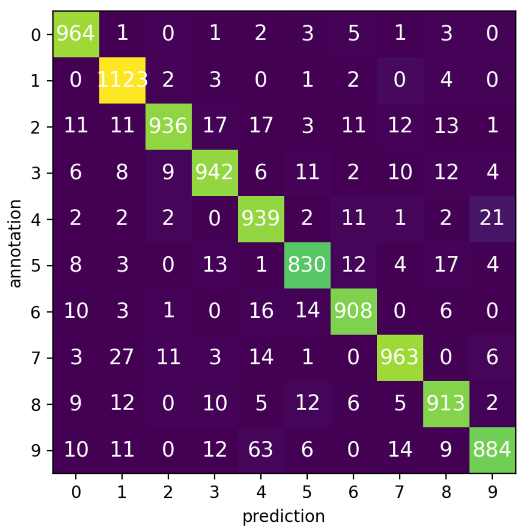

Figure 1: Confusion matrix for MLP

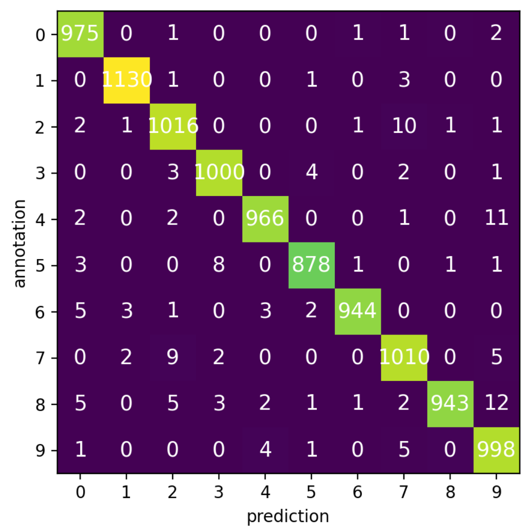

Figure 2: Confusion matrix for ConvNet