Assignment 1

October 12, 2021

Andreas Kaufmann
nethz Username: ankaufmann
Student ID: 15-545-585

By submitting this work, I verify that it is my own. That is, I have written my own solutions to each problem for which I am submitting an answer. I have listed above all others with whom I have discussed these answers.

Task 2 - Simple 2D classifier

2.3 Training loop

I achieve an accuracy of 50 %, which is the expected result. With a linear separator we can't do better than predicitng half of the points correctly, as the data is in concentric circles, thus if we draw a line we always cut through both circles.

2.4 Multi Layer Perceptron

This time, the accuracy goes up to 99.8016 %. As we introduced non linearities and more nodes, we can now fit a more complex function instead of just a linear one. The number of nodes seem to be sufficient to create a nonlinear function which separates the inner circle from the outer circle.

2.5 Feature transform

Instead of using the Cartesian coordinates, I used the polar coordinate system. The polar coordinate system describes the coordinates by the distance from the center to the coordinate. The distance is smaller for points, belonging to the inner circle. Thus this part of the polar coordinate is sufficient to classify the points linear. To avoid using a square root, I just set the second dimension of each coordinate to $x^2 + y^2$ which has the exactly same effect. The accuracy results in 99.4048% after 10 Epochs of training.

Task 3 - Digit classifier

2.3 Multi-layer perceptron

When using just one linear layer, I obtain a performance of 91.15 % with 5 epochs of training. By using an additional hidden layer with 32 nodes and a the ReLu activation function, the accuracy increases to 93.86% after 5 epochs of training.

2.4 Convolutional network

With the convolutional network, I obtain the accuracy 98.61% after 5 epochs of training.

2.5 Comparison of parameters

- MLP:
  - Hidden Layer: $\underbrace{784 \times 32}_{\text{Weights}} + \underbrace{32}_{\text{Bias}} = 25120$ parameters
  - Output Layer (Linear): $\underbrace{32 \times 10}_{\text{Weights}} + \underbrace{10}_{\text{Bias}} = 330$
  - Total: 25450 Parameters
- Conv Net:
  - Conv Layer 1: $\underbrace{1}_{\text{Nr. channels}} \times \underbrace{3 \times 3}_{\text{Kernelsize}} \times \underbrace{8}_{\text{Nr. Kernels}} + \underbrace{8}_{\text{Bias}} = 80$
  - Conv Layer 2: $\underbrace{8}_{\text{Nr. channels}} \times \underbrace{3 \times 3}_{\text{Kernelsize}} \times \underbrace{16}_{\text{Nr. Kernels}} + \underbrace{16}_{\text{Bias}} = 1168$

- Conv Layer 3: $\underbrace{16}_{\text{Nr channels}} \times \underbrace{3 \times 3}_{\text{Kernelsize}} \times \underbrace{32}_{\text{Nr. Kernels}} + \underbrace{32}_{\text{Bias}} = 4640$
- Linear Layer: $\underbrace{32 \times 10}_{\text{Weights}} + \underbrace{10}_{\text{Bias}} = 330$
- Total: 6218 Parameters

- Explanations:
  - Linear Layer: Weights is always nr. of inputs times nr. of output nodes. As we have a weight for every input-output association. For every output we additionaly have a bias value.
  - Conv. Layer: Kernels are of size given by the layer (here its always 3x3) and the number of input channels (so they are cubic). We have several of these Kernels + for every kernel a bias value.

2.6 Confusion matrix

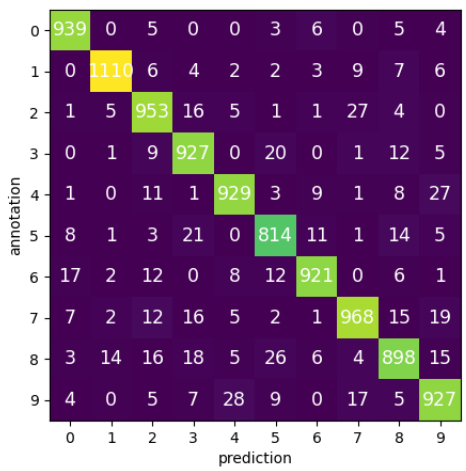

Figure 1: Confusion Matrix MLP

Figure 1 shows the confuison matrix of the MLP. For the MLP, the accuracy was 93.86%, thus we see, the diagonal part is strong but we have still many wrong prediction.

Figure 2 shows the confusion matrix for the convolutional network. For the Convolutional network, the accuracy was 98%. Thus, the matrix is almost diagonal. There is just a few numbers which were not predicted correctly. For instance entry 2,7 is 10, i.e. nr. 2 and nr. 7 were confused 10 times, which makes sense as thes two digits look quite similar.

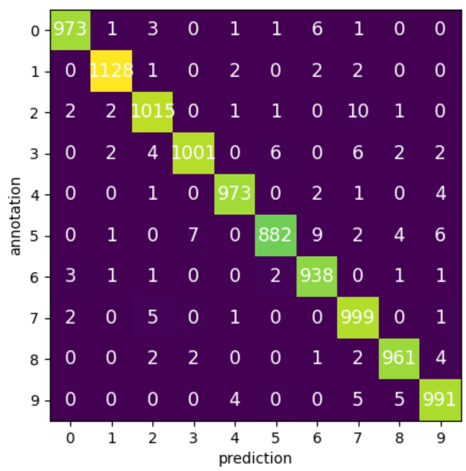

Figure 2: Confusion Matrix Convolutional network