First Team Task (MLP)

Report Exercise 2b:
We used the scikit learn MLP implementation.

First we optimized the number of neurons, other parameters were set to default. Results varied, because of the randomness in the algorithm and depending on the machine:

Neurons	10	20	40	80	100
Accuracy	0.88	0.93	0.94	0.96	0.96
Runtime	41s	83s	141s	99s	56s

Since the accuracy increased with the number of neurons in the hidden layer and the sklearn MLP algorithm terminates if a certain level of convergence is reached (and usually the more neurons the faster the convergence), we saw that the run time usually first increased and then decreased. Thus we optimized the other parameters for the MLP with 100 neurons.

Then we optimized the learning rate for 100 neurons, other parameters were set to default. Results varied, because of the randomness in the algorithm and depending on the machine:

Learning rate	0.001	0.012	0.023	0.034	0.045	0.056	0.067	0.078	0.089	0.1
Accuracy	0.96	0.48	0.26	0.11	0.10	0.10	0.10	0.10	0.10	0.10
Runtime	56s	14s	12s	12s	12s	10s	10s	10s	10s	10s

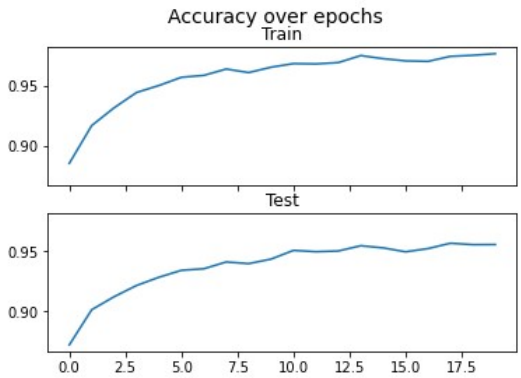
The accuracy droped insanly fast if we increased the learning rate. Therefore we kept it at 0.001 (which is default in sklearn anyway).

Finally we kept the 100 neurons in the hidden layer, the learning rate at 0.001 and optimized the number of training iterations in range [1, 400], other parameters were set to default. Results varied, because of the randomness in the algorithm and depending on the machine:

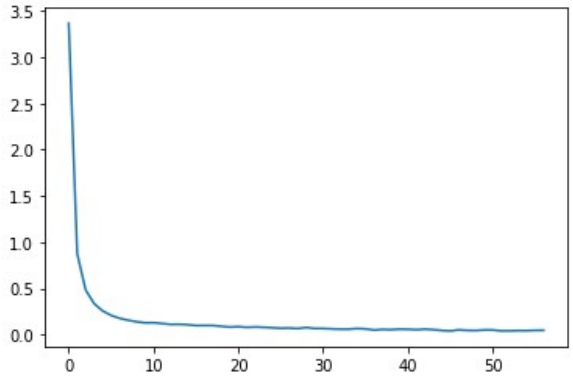
Iterations	1	10	60	200	400
Accuracy	0.90	0.95	0.96	0.96	0.96
Runtime	1s	8s	54s	56s	91s

We got a lot of warnings, since the MLP did not converge before it reached the low numbers for the maximal training iterations. But usually, the accuracy increased by increasing the number of iterations and approached an upper maximum around 0.96.

Below are the plots for the accuracy curve depending on the number of epochs (since we used the stochastic solver 'adam', max_iter determines the number of epochs), to avoid a bunch of warnings, we did 10 iterations at once and repeated this 20 times to get 200 epoches:



Accuracy curve over 200 epoches



Loss curve while training the model over the epoches