First Team Task (CNN)

Report Exercise 2c:
We follow mainly this tutorial: https://www.analyticsvidhya.com/blog/2019/10/building-image-classification-models-cnn-pytorch/

First we fill the missing gaps in the exercise CNN template. Since the images are stored in 28x28 matrices, it is clear that the first gap "self.expected_input_size" has shape (28, 28).
Then the last gap for the last layer (for the final classification): Our data are digits from 0 to 9, so we have 10 classes, thus it must be "nn.Linear(1536, 10)".
Now the second gap: "in_channels" is also quite simple, our images are in greyscale, so there is only one incoming channel (RGB would be for example 3).
We know that the last layer has dimension 1536×1 after flattening. Thus we can calculate the out_channels and kernel_size: First we do a prime factorization of $1536 = 2^9 \cdot 3$. Since we have a quadratic matrix, we will have its output size after the first layer squared and then multiplied with out_channels. We consider the following formula to calculate the output size (before out_channels): $o = \frac{i-k+2p}{s} + 1$; i input size, k kernel size, p padding and s stride.
In our case $i = 28, p = 0, s = 3$, to get a reasonable value for o we can estimate the kernel: if $1 \leq k \leq 4$ we will get $o = 9$, which cannot be the case, since 1536 has not enough 3's as prime factors. We conclude that $5 \leq k \leq 7$ to get $o = 8$. We set $k = 7$ and calculate the missing factor and conclude out_channels = 24.

Then we define the model and take on the function to train the model from the tutorial.
Finally we can optimize the parameters. We start with 25 epochs and vary the learning rate:

Learning rate	0.001	0.005	0.01	0.05	0.1
Accuracy	0.83	0.92	0.93	0.95	0.92
Runtime	39s	39s	41s	39s	38s

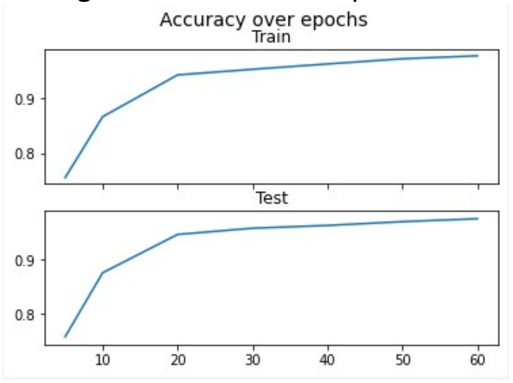
We seem to get the best results for a learning rate around 0.05.

Now we optimize the numbers of epochs with a fixed learning rate of 0.05:

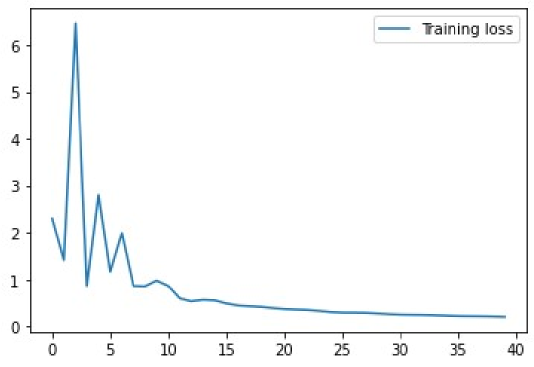
Epochs	60	50	40	30	20	10	5
Accuracy	0.97	0.97	0.96	0.96	0.95	0.88	0.76
Runtime	88s	75s	61s	45s	30s	15s	8s

We seem to reach a plateau in the accuracy of the learning process at around 40 epochs.

Below are the plots for the accuracy curve of our optimized model (lr = 0.05, epochs = 40), depending on the number of epochs and the loss curve during the training:



Accuracy curve over 60 epoches



Loss curve while training the model over 40 epoches