Deep Learning Lab - Report - Exercise 1

Rabea Laura Turon

November 13, 2018

1 Introduction

This report summarizes the results of Exercise 2 of the Deep Learning Lab Course 2018 at the Albert-Ludwigs-University of Freiburg. The overall task was to implement a convolutional neural network (from now on CNN) for classification of digits from the MNIST dataset with a focus on testing different hyperparameters.

For all tests the CNN consisted of two convolutional layers with ReLU activations and max pooling after each layer. After the convolutional layers two fully connected layers are added. The first one with 128 units and a ReLU activation, the second one with 10 units and Softmax activation for the classification of the digits. As a learning algorithm Stochastic Gradient Descent was used.

2 Learning Rate

In a first test the effect of different learning rates was tested. The filter size of the convolutional layers was set to 3 and 16 filters were used per layer. Then different values for the learning rate were tested. The validation performance of the network during training for these different learning rates can be seen in Figure 1.

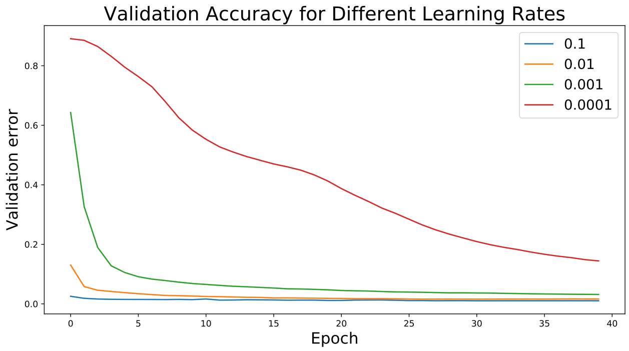

Figure 1: Missclassification on the validation set during training of the CNN with different learning rates as specified in the legend.

It can be seen that the network improves with all learning rate values, but is much faster the higher the value is. I think that the biggest value of 0.1 works best for this network since the CNN converges very fast and reaches its best accuracy. Not that the number of epochs was chosen quite large (40 epochs) in order to see more of the training process with the smallest learning rate of 0.0001. Even after 40 Epochs the CNN has not converged with this learning rate. The CNN with a learning rate of 0.1 seems to have converged already after about 12 epochs.

3 Convolution Type

In the second test the effect of the filter size of the two convolutional layers was tested. For that the learning rate was set to the previously best value of 0.1. The number of filters was again set to 16. The validation performance can be seen in Figure 3

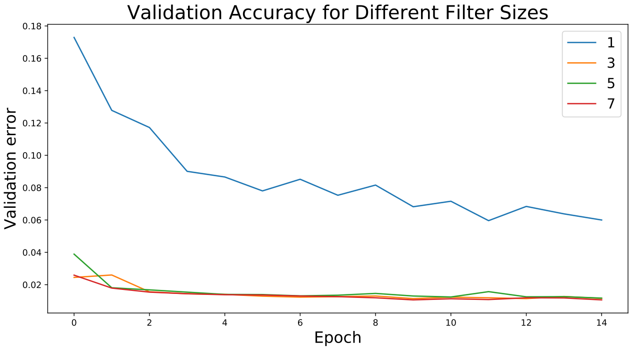

Figure 2: Missclassification on the validation set during training of the CNN with different filter sizes as specified in the legend.

As can be seen in Figure 3 the filter sizes 3, 5 and 7 seem to work equally well. Only filters of size 1 do not work that well. This could be due to the fact that filters of size 1 do not use any kind of receptive field so that they cannot learn features like edges which might be quite important for digit classfication. If I had to choose one of the filter sizes, I would use a filter size of 3 since these filters perform equally well as the bigger filters but use less weights.

4 Random Search

As a last test random search was performed on the hyperparameters learning rate (sampled logarithmically between 0.0001 and 1), batch size (sampled logarithmically between 16 and 128), the number of filters (again sampled logarithmically between 8 and 64) and the filter size (either 3 or 5). For each configuration the network was trained for 6 epochs and 50 different configurations were tested. The validation loss after 6 epochs for the different configurations can be seen in Figure **??**.

The best configuration the random search came up with consisted of 62 filters per layer, a batch size of 61, a learning rate of roughly 0.2085 and a filter size of 3. A CNN with this configuration was trained again for 12 epochs which resulted in a final test error of 0.96%.

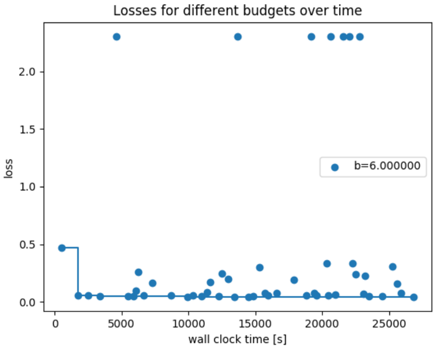

Figure 3: Validation loss after 6 epochs for different configurations of the hyperparameters. The best configuration was found close to the beginning.