Deep Learning Lab
Exercise 01

Megan Klaiber

October 30, 2018

1 Introduction

In this exercise a multi-layer perceptron (MLP) will be implemented to classify the handwritten MNIST digits. Different hyperparameters are tested to achieve a good result on a test set.
The implementation of the MLP is done according to the given framing.

2 Neural Network

2.1 Implementation

For the implementation of an MLP three activation functions can be used in the hidden layers: Sigmoid, Tanh and Relu. Softmax and linear are available for the output layer. Gradient descent (GD) and stochastic gradient descent (SGD) are implemented for optimization.

2.2 Result

The resulting model with the best performance is a neural network with one hidden layer containing 220 units. This layer utilizes a Relu activation function. For the output layer the softmax function was used. The network was trained with 50.000 images and validated on 10.000 images. With the following hyperparameters the error on the validation set converges to 1,86%:

- learning rate: 0.15

- epochs: 30

- batch size: 30

- optimizer: stochastic gradient descent

Figure 1 shows the error on the training and validation set during the training. With a trained model on the full dataset (training and validation set) a test error of 1.73% on unseen test data was achieved.
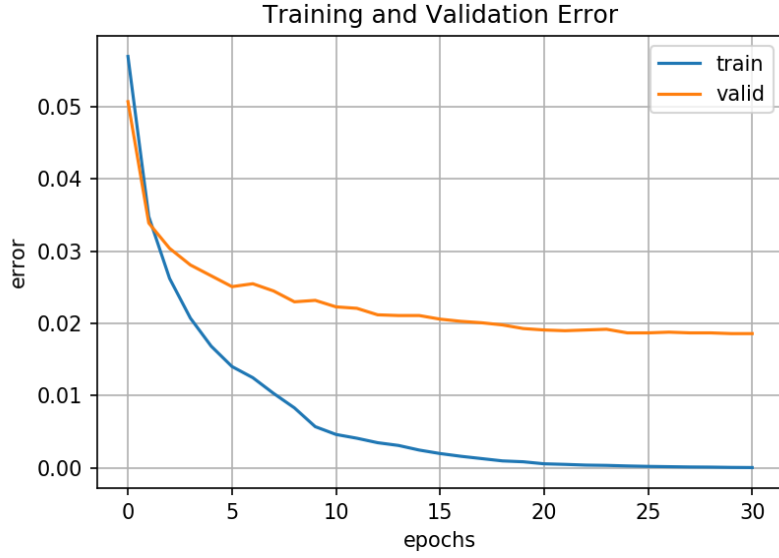
Figure 1: Change of training and validation error while training epochs of MLP on MNIST dataset. MLP has one hidden layer with 220 units and ReLu activation. Optimization with SGD used a learning rate of 0.15 and batches of 30 images.

2.3 Optimization

To see how different optimizers influence the training, gradient descent and stochastic gradient descent were compared. Therefore the settings from subsection 2.2 are used. Figure 2 compares the training error for the two optimizers. It implies that gradient descent converges slower than stochastic gradient descent.
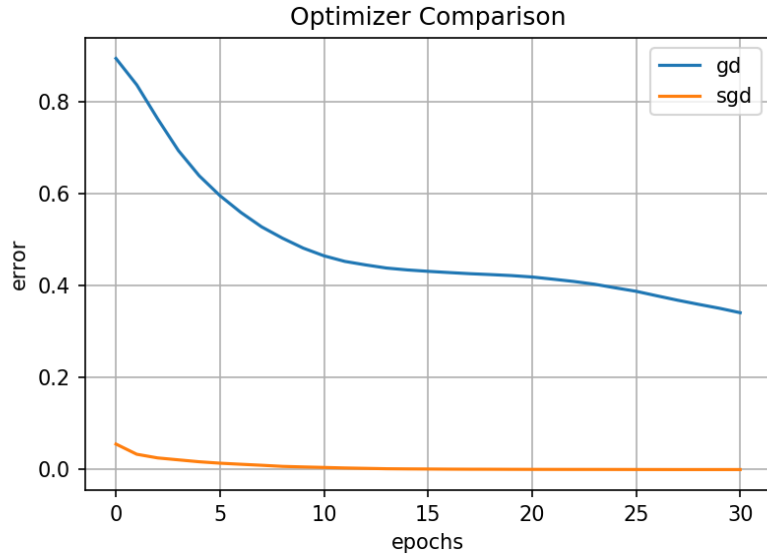

Figure 2: Change of training error with gradient descent and stochastic gradient descent as optimizers.

3   Conclusion

With this exercise you can see that a simple MLP achieves good results on the MNIST dataset. In subsection 2.3 you can see that the right optimization method is important for the training procedure. In addition it is possible to achive better results if you add regularization techniques.