Deep Learning Course 2017 **Ilia Dobrusin**
Autonomous Intelligent Systems
University Freiburg Due Date: 06.11.2017

Deep Learning Lab Course 2017 - Exercise 1

In the first exercise of the course we have to implement a Neural Network and classify the MNIST digits. This report will give a short overview over the implementation and the resulting classification of the MNIST data set. The MNIST dataset consists of a set of handwritten digits which can be used for training and evaluating a neural network. The training set consists of 60000 examples with an accompanying test set of 10000 examples.

The theoretical background for neural networks is explained in the introductory lecture of this lab course and can be found online[1].

Implementation

In the first exercise we have to implement a simple feed-forward neural network by completing a provided stub. The implementation can be found on Github[2].

Activation Functions

The stub provided a code skeleton for a simple neural network with different types of layers and helper functions, which provide the most common activaton functions and their derivatives.

Neural Network and Layer

Each layer in the network implements the Layer interface which devines functions for forward and backward propagation. The different types of layers are: input layer, fully connected layer, linear output, and softmax output. The network itself is then constructed as a list of layers, where the weights and biases are propagated through the network.

The Neural Network

The resulting network which is used to classify MNIST consists of an input layer with an iput dimension of $28 * 28 = 784$. The hidden layers consist of three Fully Connected Layers with ReLu activation functions. The first layer has 300 hidden units, the second layer has 100 hidden unit and the last layer reduces the number of units to 10, which is the number of classes to classify. The weights and biases are initialized with a normal distribution with a standart deviation of 0.01. The last layer is a softmax output, which maps the resulting values to values between 0 and 1.

[1] `http://ais.informatik.uni-freiburg.de/teaching/ws17/deep_learning_course/presentation1a2017.pdf`

[2] `https://github.com/idobrusin/DeepLearningLab2017`

Results

The network described in the last section was trained and evaluated with the training set of MNIST, which was devided in a training, validation, and test set. Stochastic gradient descent was chosen as the the optimization strategy, as it provides better results during training. The network was trained with 50 training epochs Figure 1 shows the training and validation error during the implemenation of the network. The resulting network was trained with the full training set of 60000 digits. Figure 2 shows the training error of the final network on the full set. Figure 3 shows correctly and incorrectly classified digits from the MNIST data set. Overall, the network achieved an accuracy of 1.86% error on the test set.

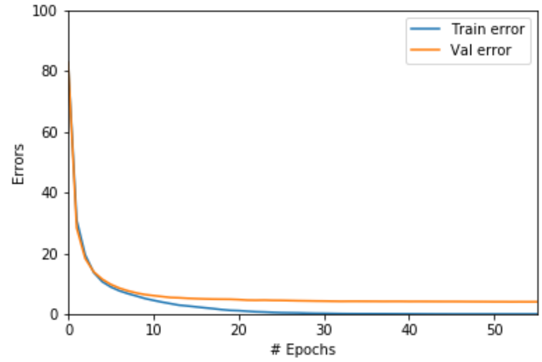

Figure 1: Training and validation error during implemenatation

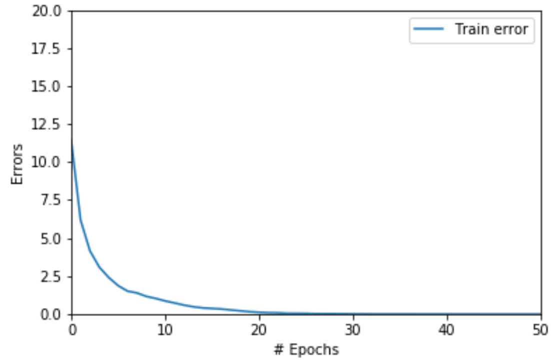

Figure 2: Training error during training on the full set

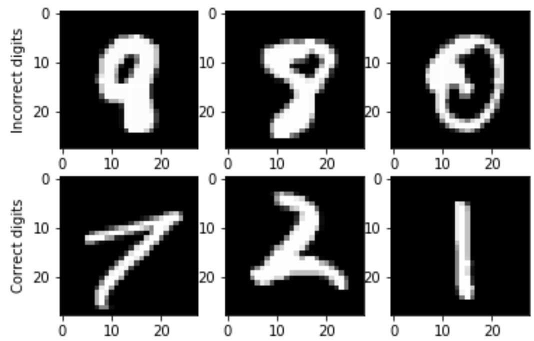

Figure 3: Correctly and incorrectly classified digits