Homework 2: Deep Learning

Out May 2; Due May 8, 12 a.m.

Kristian Kersting, Dominik Hintersdorf, Quentin Delfosse
{kersting, dominik.hintersdorf, quentin.delfosse}@cs.tu-darmstadt.de

In this homework, you will use a neural network to classify MNIST! Probably the most famous dataset in the deep learning community[1]. MNIST is a collection of images of handwritten digits. The task here is to classify an image into the correct digits. You are to train a neural network on the train datasets (X_{train}, Y_{train}) and then use the test datasets (X_{test}, Y_{test}) to make predictions. X datasets contain the image data, the Ys are the labels.

Do not use a DL framework. Use the provided notebook!

1. Implement an NN, similar to the last homework, with 1-hidden layer (you get to decide on the size) and a 1-output layer of size 10 (each output represents a class). Use sigmoid activation functions. Implement the forward and backward pass (do not use (for-)loops in the implementation).

2. Implement the training of the NN with Stochastic Gradient Descent.

To train this network, you have to use one-hot-encoding. That is, you convert your numerical label into a vector of 10 dimensions that is all zeros, except for one value that contains a one in the position of the value of the label.

$0 \rightarrow [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$
$1 \rightarrow [0, 1, 0, 0, 0, 0, 0, 0, 0, 0]$
$\vdots$
$8 \rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 1, 0]$
$9 \rightarrow [0, 0, 0, 0, 0, 0, 0, 0, 0, 1]$

Optimize the network with the square error loss function 1.

$$\mathcal{L}(\theta; \mathbf{x}, \mathbf{y}) = \frac{1}{M} \sum_{n=1}^{M} \frac{1}{2} \| f_{NN}(\mathbf{x}_n) - \mathbf{y}_n \|^2 \tag{1}$$

3. Obtain the predicted class by finding the output neuron with the highest value.

4. Try different learning rates and batch sizes. Plot the loss values of each optimization step using matplotlib[2]. What effect does data normalization have?

[1] http://yann.lecun.com/exdb/mnist/
[2] https://matplotlib.org