TECHNISCHE UNIVERSITÄT DRESDEN

Professur für Informationstheorie und maschinelles Lernen, Fakultät Elektrotechnik und Informationstechnik

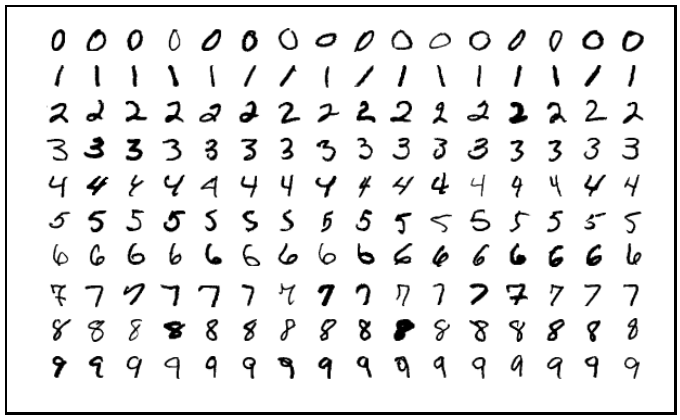
Prof. Rafael Schaefer, Dr. Martin Mittelbach, Muah Kim, Dr. Rick Fritschek 16.06.2023

4. Exercise of the Course
Einführung in das maschinelle Lernen

Remark: The problem set will be extended by another problem on CNNs within the next days.

Problem 12: (MNIST image classification with MLP and PyTorch)

We consider the MNIST dataset, which is a set of labeled images of handwritten digits $0, 1, 2, \ldots, 9$ that is widely used for training and testing in the field of machine learning. The MNIST dataset contains 60000 training images and 10000 testing images of size 28×28 pixels.



The goal is to implement and train a multi-layer-perceptron such that the images of handwritten digits are classified correctly, where the training image set is used for network training and the testing image set for verification.

We use PyTorch for this problem and build the implementation of the network and the training algorithm upon the code skeleton provided with the file `eml2023-u04-problem-12.ipynb`.

The specifications of the implementation are as follows:

- Network architecture:
  - The number L of network layers and the number $N_1, N_2, \ldots N_{L-1}$ of nodes in each hidden layer are free parameters.
  - The activation functions in the hidden layers are free parameters, where `ReLU`, `Tanh`, and `Sigmoid` are the relevant options.
  - The activation function of the output layer is `Softmax`.
- Training algorithm:
  - For training either the *stochastic gradient decent* (SGD) optimizer or the *adaptive moment estimation* (Adam) optimizer is used, where the learning rate and the momentum (for SGD) are free parameters.
  - The training data are partitioned into minibatches for training, where the batch size is a free parameter.
  - The default network parameter initialization of PyTorch is used.
  - As loss function the cross entropy is used.
  - The number of training epochs is a free parameter.

a) Derive a formula, which calculates the number of trainable parameters of the network architecture.

b) Implement the network architecture, training and verification. Record and subsequently illustrate the classification accuracy for the training and testing data over the number of training epochs.

c) Train the network with $L = 2$, $N_1 = 64$ and `ReLU` activation function with the following hyperparameters

batch size	learning rate	epochs
64	0.9	10

and

i) SGD optimizer with momentum 0.5,

ii) Adam optimizer.

Compare the classification accuracy for the training and testing data over the number of training epochs for i) and ii).

d) Repeat Task c) for a learning rate of 0.001 and compare the results.

e) Repeat Task c) and Task d) for `Tanh` and `Sigmoid` activation function.

f) Repeat Task c) i) with an initialization of all network parameters drawn randomly according to a zero mean normal distribution with unit variance and compare the results.

g) Repeat Task c) i) for a batch size of $32\,/\,64\,/\,128$ and compare the results.

h) Repeat Task c) i) with $N_1 = 32\,/\,64\,/\,128$ and compare the results.

i) Repeat Task c) i) with $L = 2\,/\,4\,/\,6$, $N_1 = N_2 = \ldots = N_{L-1} = 64$, and `ReLU` activation function in each layer and compare the results.