Computational MR imaging
Laboratory 10: Training neural networks with backpropagation

Report is due on Wednesday the week after the lab session at 23:59. Send your report by email to Bruno Riemenschneider (bruno.riemenschneider@fau.de) and Florian Knoll (florian.knoll@fau.de).

Learning objectives

- Train a fully connected network for MNIST classification in Pytorch
- Change the number of hidden layers from 1 to 5
- Observe the effect of gradient instability during neural network training with backpropagation

Training a neural network for MNIST classification

Open the script MNIST_classification_pytorch_1p10.py. This script serves as a template that handles file I/O and plotting.

i. Familiarize yourself with the script, the data, and the labels. Plot an example of selected training images.
ii. Define a fully connected neural network architecture with one hidden layer with 30 neurons and train it. Use the sigmoid activation function for input and the hidden layers, and log softmax for the output layer. Recommended training settings are:
    i. Batch size: 64
    ii. Loss function: NLLLoss
    iii. Optimizer: SGD
    iv. Training epochs: 15
    v. Learning rate: 0.003
    vi. Momentum: 0.9
iii. Plot the training loss over the epochs

1. Investigate the impact of the number of hidden layers and the activation function

i. Increase the number of hidden layers to [1,5].
ii. Change the activation function from sigmoid to ReLU for the input and hidden layers.
iii. Plot the training loss over the epochs for all experiments and discuss your findings.