Computational MR imaging
Laboratory 9: Machine learning in MRI and neural network architecture design

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

- Get familiar with designing and training neural networks in Pytorch
- Examine effects of dropping data on neural network performance
- Learn about effects of over and underfitting

Installation of the deep learning framework Pytorch

In this lab you will use PyTorch. PyTorch is a powerful deep learning package developed by Facebook Artificial Intelligence. Before installing these, it is recommended to install Anaconda, a software program that manages utilities and toolboxes for Python. Go to the Anaconda web page (https://www.anaconda.com/) and install Anaconda. Once you have Anaconda installed, you can use the following steps to create an environment and install PyTorch as well as all other packages that we will use in this lab from a command shell:

```
# Create and activate the environment, we are using python 3.7 here
conda create --name Pytorch_1p5 python=3.7
source activate Pytorch_1p5

# Install Pytorch
conda install pytorch torchvision -c pytorch

# Install additional helper packages that we will use to load, manipulate and plot images
conda install matplotlib
conda install pandas
conda install scikit-learn
conda install -c conda-forge opencv
conda install -c conda-forge imutils

# This step is optional, spyder is an IDE that you can use
conda install spyder

# If you want to deactivate the environment again
source deactivate Pytorch_1p5

# And if you want to completely remove it again
conda remove --name Pytorch_1p5 --all
```

1. Diffusion Data Classification

Open the script dti_classification_pytorch_1p5.py. This script classifies labels of each volume based on different parametric maps.

i. Look at the labels in the data. What is labeled?
ii. Verify that you can run a neural network training on your computer.
iii. After you verify that you can train with the default settings, change aspects of the data and examine the performance – for example, try dropping different parameter maps and examining classification performance – the choice is yours! Report your results and speculate on their meaning.

2. Image Quality Classification

The scripts CNN_recon_quality_classification_pytorch_1p5*.py classify images as being either fully-sampled or from compressed sensing. Try to build the CNN model architectures we discussed in the class to induce overfitting or underfitting. Record your experience, along with any plots that you generated during the training. How does CNN complexity relate to overfitting and underfitting?