



# Neural Networks and Data Science

## Lab #7

22.11.2023

Deadline: 29.11.2023, 12:10

Dr. Marcel Völschow



## Problem 1

Neural networks are trained and tested with different datasets to make sure that the networks can actually deal with new data and not just data it has already seen. If the label of a tested image corresponds to the largest entry of the output vector, the image has been correctly classified by the network. The fraction of correctly classified test images is a measure for the network's performance.

- a) Download the MNIST npy files provided in the shared folder. They contain the MNIST test and the training datasets alongside their labels. Load them into a notebook.
- b) Create a network with 784 input nodes, 100 hidden nodes, 10 output nodes and a learning rate of 0.6. Train the network with all 60000 images of the `dataTrain` dataset.
- d) Write a function or a method `evaluate` that determines the fraction of `dataTest` images that the network classifies correctly. What is the performance of your network? Hint: `np.argmax(arr)` returns the index of the largest element of `arr` ...
- e) Use `plt.imshow` (see session 6) to plot the weight matrices of the trained network. What has changed compared to the untrained state?

## Problem 2

Training large neural network can be a very long process. Therefore, we need a method to permanently save and restore a network's memory.

- a) Write a method `saveWeights` that saves the matrices `wih` and `who` to the disk. Use

```
np.save(filename, array)
```

and the file name extension `*.npy` (e.g. `filename = "wih.npy"`) to create binary files.

- b) Add a method `restoreWeights` that restores the network's memory. Use

```
array = np.load(filename)
```

to read binary files.

- c) Create an instance of the network `testNet`. Train it, save the network's weights, set them to 0 and restore them again. Verify that your network is able to classify handwritten numbers again.

## Problem 3

**This task is intended for those of you who would like to go an extra mile ...**

With every training image fed through the network, the weight matrices slightly change.

- a) Write a `plotWeights(self, filename)` method takes a filename and quietly saves the image to a file.
- b) Create a network with 784 input nodes, 100 hidden nodes, 10 output nodes and a learning rate of 0.3. Train the network with all 60000 images of the `train` dataset by iterating through the array and calling the `train` method for every image. After every 600th iteration, call your `plotWeights` method. Chose a reasonable name for your image files.
- c) Create a gif animation from your image files.