Heidelberg University
Institute of Computer Engineering
Computing Systems Group

Franz Kevin Stehle
Hendrik Borras

Neural Networks from Scratch
Winter Term 2024

Exercise 2

- **Return electronically until Wednesday, November 5, 2024, 09:00**
- **Include your names on the top sheet. Hand in only one PDF.**
- **A maximum of three students are allowed to work jointly in a group.**
- **When you include plots add a short explanation of what you expected and observed.**
- **Hand in source code if the exercise required programming. You can bundle the source code along with the PDF in a .zip file.**

2.1 Neural network from scratch

In this exercise we are going to implement a small three layer MLP with sigmoid activations from scratch and then train it on the MNIST dataset. For this we provide a template script, which you can use to start off with. The template already contains a lot of the structure of the network, dataset loading and basic training routine, so the main task boils down to implementing the forward and backward paths and and update routines for different components.

- Note: If you follow implementation in the template, then the script won't use any GPU capability. Which is fine for such a small MLP and task, so you are fine just running it on a CPU.
- Make sure to use only *numpy* or native Python for computations. PyTorch is only to be used for the dataloaders.
- First, activate your conda environment on the cluster or on your own machine, as explained in the brook user manual. Run `conda activate eml` on the cluster to activate it.
- Implement the missing forward and backward paths and update routines for the three layer MLP, as outlined in the `exercise02_template.py` file. These are in particular the central components of the Linear/Dense layer, Sigmoid activation function and softmax function.
- Train the network with the given default parameters for 30 epochs. Then plot how the test and train loss develop with the number of epochs. Additionally create a plot, which shows how the test accuracy develops.
- Run the training again for 30 epochs, but this time with varying learning rates. Choose at least five different learning rates between 1. and 0.001. Create a plot with the test accuracy over epochs to compare the different learning rates.
- Discuss the results obtained in the previous plots.