

Neural Networks from Scratch
Winter Term 2025

Exercise 3

- Return electronically until Tuesday, November 5, 2025, 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.

3.1 Experiment logging

In general we recommend using a dedicated library for logging and visualizing your experiment results. Over the past years many of these have sprung up in the ML ecosystem. And in this exercise we will take a look at one of them. As a reference this exercise will use Weights and Biases, however you are free to use any equivalent service as well to complete this exercise. Some examples are: Neptune or MLFlow

- Set up an account for Weights and Biases (WandB).
- Follow the quickstart guide to instrument your previous from-scratch implementation of the MLP.
- Reproduce the learning rate plots from the previous exercise in WandB and include screenshots in your submission.

3.2 Neural network on GPU

In this exercise we are going to move our CPU-only implementation of the small MLP onto the GPU. Following this we will evaluate when and by how much a GPU can help in speeding up computations. Please take close note of the cluster manual, as it explains the basics of accessing the provided GPU resources: [how_to_cluster.pdf](#)

Note: If you would like to complete this exercise on your own machine you must have an Nvidia GPU present and accessible to CuPy.

- For this exercise we provide a source code template called `ex32_template.py`. You can use this as a starting point for all exercises on this sheet. The template implements a working NumPy MLP from the previous exercise.
- First, activate your conda environment on the cluster or on your own machine, as explained in the cluster user manual. Run `conda activate eml` on the cluster to activate it.
- To run your script on the cluster we provide an `SBATCH` script, which you can base your experiments off of: `run_via_sbatch.sh`
- The template utilizes `tqdm` for progress visualization. This library is not included in the `eml` environment on the cluster. However, after activating the environment you can install it to your user directory using `pip install tqdm`
- Implement the MLP using CuPy. This is mostly a direct drop-in replacement. Make sure to use only CuPy for computations. PyTorch is only to be used for the dataloaders.
- Run a training run with the new network implementation once on the CPU and once on the GPU (with the default parameters and for 30 epochs) and plot the test accuracy over time (not over the number of epochs).
- Discuss the results obtained in the previous plot.

- Now increase the size of the intermediate layers for the MLP, that is, increasing the number of neurons per layer or equivalently making the weight matrix of the layer larger. Then measure both the CPU and GPU time for one epoch (at least 7 measurement points).
- Plot the time per epoch for both CPU and GPU, dependent on the intermediate layer size.
- Plot the speedup of the GPU implementation over the CPU implementation, dependent on the intermediate layer size.
- Discuss the results obtained in the previous plots.