Introduction to Artificial Intelligence
Lab 5: Artificial Neural Networks

Daniel Marczak

Summer 2025

1 General Task Description

Implement a multilayer perceptron for image classification. The neural network should be trained with the mini-batch gradient descent method. Remember to split the dataset into training and validation sets.

The main point of this task is to evaluate how various components and hyperparameters of a neural network and the training process affect the network's performance in terms of its ability to converge, the speed of convergence, and final accuracy on the training and validation sets. For evaluation, please create plots visualizing:

- The loss value for every learning step,
- Accuracy on the training and validation set after each epoch.

Make sure to include conclusions and observations based on the conducted experiments in your report.

The details are described in your variant of the project.

Variant 1

Use the MNIST dataset. Evaluate at least 3 different numbers/values/types of:

- Learning rate,
- Mini-batch size (including a batch containing only 1 example),
- Number of hidden layers (including 0 hidden layers - a linear model),
- Width (number of neurons in hidden layers),
- Loss functions (e.g., Mean Squared Error, Mean Absolute Error, Cross Entropy).

Variant 2

Use the MNIST dataset. Evaluate at least 3 different numbers/values/types of:

- Learning rate,
- Mini-batch size (including a batch containing only 1 example),
- Number of hidden layers (including 0 hidden layers - a linear model),
- Width (number of neurons in hidden layers),
- Optimizer type (e.g., SGD, SGD with momentum, Adam).

Variant 3

Use the FashionMNIST dataset. Evaluate at least 3 different numbers/values/types of:

- Learning rate,
- Mini-batch size (including a batch containing only 1 example),
- Number of hidden layers (including 0 hidden layers - a linear model),
- Width (number of neurons in hidden layers),
- Activation functions (e.g., Sigmoid, ReLU, GELU).

Variant 4

Use the FashionMNIST dataset. Evaluate at least 3 different numbers/values/types of:

- Learning rate,
- Mini-batch size (including a batch containing only 1 example),
- Number of hidden layers (including 0 hidden layers - a linear model),
- Width (number of neurons in hidden layers),
- Loss functions (e.g., Mean Squared Error, Mean Absolute Error, Cross Entropy).

Variant 5

Use the KMNIST dataset. Evaluate at least 3 different numbers/values/types of:

- Learning rate,
- Mini-batch size (including a batch containing only 1 example),
- Number of hidden layers (including 0 hidden layers - a linear model),
- Width (number of neurons in hidden layers),
- Optimizer type (e.g., SGD, SGD with momentum, Adam).

2 Tips

- The network can be implemented with a library offering neural network layers, optimizers, and error backpropagation. However, you must implement the learning procedure yourself – you **cannot** use libraries such as fast.ai or lightning that train the model with model.fit(). I highly recommend using PyTorch. This course may be useful for starting.
- Setting a fixed seed will make your results reproducible across different runs. Check this article or others for details.
- To input an image (28 x 28 matrix for all the given datasets) into the multilayer perceptron network, you need to flatten the image - represent it as a vector.
- Training of neural networks can take a while, and you will train several of them in this assessment. Keep this in mind and do not start working on the lab at the last minute!

3 Technical Details

- The solution must be implemented in Python.
- Please ensure that your code adheres to basic standards of clean coding in accordance with PEP8. Additionally, it should contain comments in crucial parts to aid readability and understanding.
- Clear instructions on how to run and test the code should be included.
- The submission of the final report is mandatory, and the task will not be accepted without it.

4 Handing-In Guidelines

To pass the lab, it is required to submit both the code and the final report and discuss your solution during the online assessment. The online assessment will take place during the labs and should take around 10 minutes. Please notify me on Teams when you submit the solution to schedule the exact time for the meeting. You should submit the code and a PDF report to the designated repository at least a day before the online assessment of the exercise. Solutions delivered after the deadline will not be assessed. The assessments will be scheduled via excel posted on the MS Teams channel.

5 Assessment Criteria

You can earn between 0 and 5 points for the lab. The following criteria will be used to evaluate your work:

- Proper implementation of the neural network and the training procedure: 1 point.
- Final report, including training plots (see description), observations, and conclusions about the impact: 1 point.
- Online assessment: 3 points.

In case of any questions, contact me via MS Teams.