Introduction to Artificial Intelligence
Exercise 5: Artificial Neural Networks

Daniel Marczak

Summer 2023

1 General task description

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

For evaluation, create plots with:

- loss value for every learning step
- accuracy on train set after each epoch
- accuracy on validation set after each epoch

Main point of this task is to evaluate how various components/hyperparameters of neural network and training process affect the preformance of the network in terms of ability to converge and the speed of convergence, and final accuracy on train and validation sets. Make sure to include conclusiosns and observations based on conducted experiments in the report.

The details are described in your variant of the project.

Variant 1

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

- learning rate
- mini-batch size (including batch containing only 1 example)
- number of hidden layers (including 0 hidden layers - linear model)
- width (number of neurons in hidden layers)
- loss functions (e.g. Mean Squared Error, Mean Absolute Error, Cross Entropy)

Variant 2

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

- learning rate
- mini-batch size (including batch containing only 1 example)
- number of hidden layers (including 0 hidden layers - linear model)
- width (number of neurons in hidden layers)
- optimizer type (e.g., SGD, SGD with momentum, Adam)

Variant 3

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

- learning rate
- mini-batch size (including batch containing only 1 example)
- number of hidden layers (including 0 hidden layers - linear model)
- width (number of neurons in hidden layers)
- activation functions (e.g., sigmoid, ReLU, GELU)

Variant 4

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

- learning rate
- mini-batch size (including batch containing only 1 example)
- number of hidden layers (including 0 hidden layers - linear model)
- width (number of neurons in hidden layers)
- loss functions (e.g. Mean Squared Error, Mean Absolute Error, Cross Entropy)

Variant 5

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

- learning rate
- mini-batch size (including batch containing only 1 example)
- number of hidden layers (including 0 hidden layers - 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 library offering neural network layers, optimizers and error backpropagation. However, you must implement learning procedure yourself. I highly recommend using PyTorch. This course may be useful for starting.
- Setting fixed seed will make your results reproducible across different run. Check this article or other for details.
- To input an image (28 x 28 matrix for all the given datasets) to the multilayer perceptron network you need to flatten the image - represented it a vector.
- Make sure that your code is general and avoid code duplication.

3 Technical details

- The solution must be implemented in Python.
- Please ensure that your code adheres to basic standards of lean coding in accordance to PEP8. Additionally, it should contain comments in the crucial parts to help with readability and understanding.
- The clear instruction 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

You should submit the source code of your solution and final report via Teams not later than:

- 2023.05.15 EoD for the Monday group

- 2023.05.17 EoD for the Wednesday group
- 2023.05.19 EoD for the Friday group

Programs delivered after the deadline will not be assessed. The on-line assessment will take place during your labs. In case of questions, please contact me via Teams.

5 Assessment Criteria

You can get $[0, 5]$ points for the lab. The following criteria will be used to evaluate your work:

- Proper implementation of the algorithm: 2 points
- Clean and well-documented code, with clear explanations and well-implemented logic: 1 point
- Final report including results, clear explanation of the solutions and reflections on what was done well and what could be improved: 2 points

In case of any questions contact me via MS Teams