Pattern Recognition Group Exercise 2b

David Bucher, Timo Bürk, Félicien Hêche,Aleksandar Lazic, Zahkar Tymchenko

Exercise 2b - MLP

We choose the build our Multilayer Perceptron (MLP) using Python and the scikit-learn library. We first tested for the best combination of learning rate and number of neurons in the hidden layer. We did that in the python file "MLP_parameters_tuning.py" and we tested the following values for the learning rate and the number of neurons:

- Learning rate: [0.001,0.0025,0.005,0.0075, 0.01,0.025,0.05,0.075, 0.1]
- Number of neurons: [(10),(20),(40),(60),(80),(100)]

Based on our tests the most accurate pairing of those parameters are: a learning rate of 0.05 with 100 neurons in the hidden layer.

With those parameters set we then determined the best number of iterations to train the MLP for. We did that in the "MLP_iterations_tuning.py"-file. By computing the accuracy and zero-one-loss for both the training and validation set every 5 iterations. The results can be found in 1 and 2. Based on those results we determined that the optimal number of iterations are 200 iterations.

Finally we used the random_state attribute to evaluate different seeds for the random starting weights. We arbatrairily chose the values [1,26,42,67,123]. And used the accuracy of the validation as our crtieria to chose the best seed.

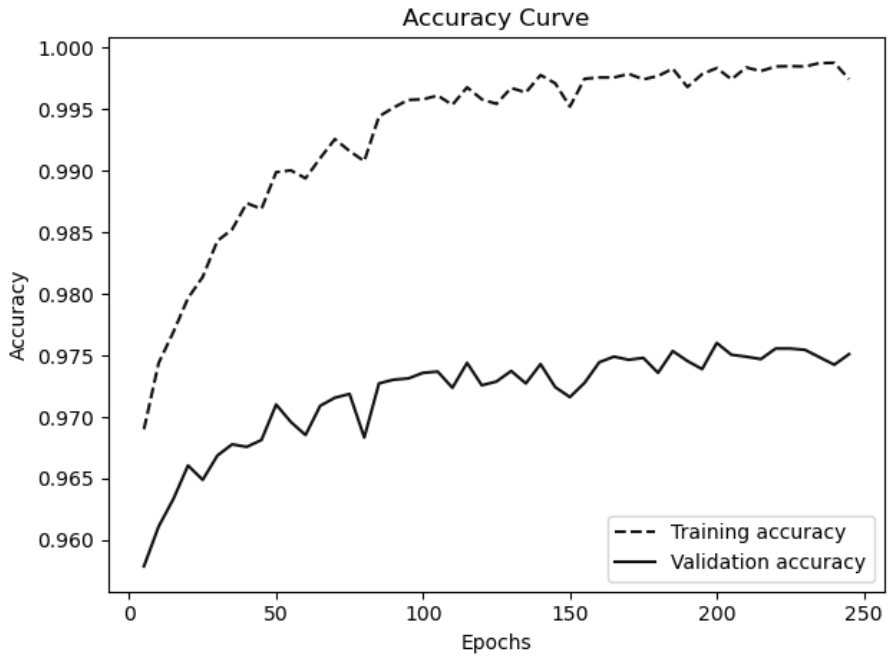

Fig. 1: Accuracy curve

```
Seed: 1, accuracy training: 0.998681592039801, accuracy
    ↪ validation: 0.9757575757575757
Seed: 26, accuracy training: 0.9988059701492538, accuracy
    ↪ validation: 0.9731818181818181
Seed: 42, accuracy training: 0.997412935323383, accuracy
    ↪ validation: 0.9722222222222222
Seed: 67, accuracy training: 0.9990547263681592, accuracy
    ↪ validation: 0.9752020202020202
Seed: 123, accuracy training: 0.9993532338308457, accuracy
    ↪ validation: 0.975
```

According to our the best seed for this training set and those parameters is the int value 1. The corresponding code can be found in the "MLP_starting_weights_or_random_seed_tuning.py file.

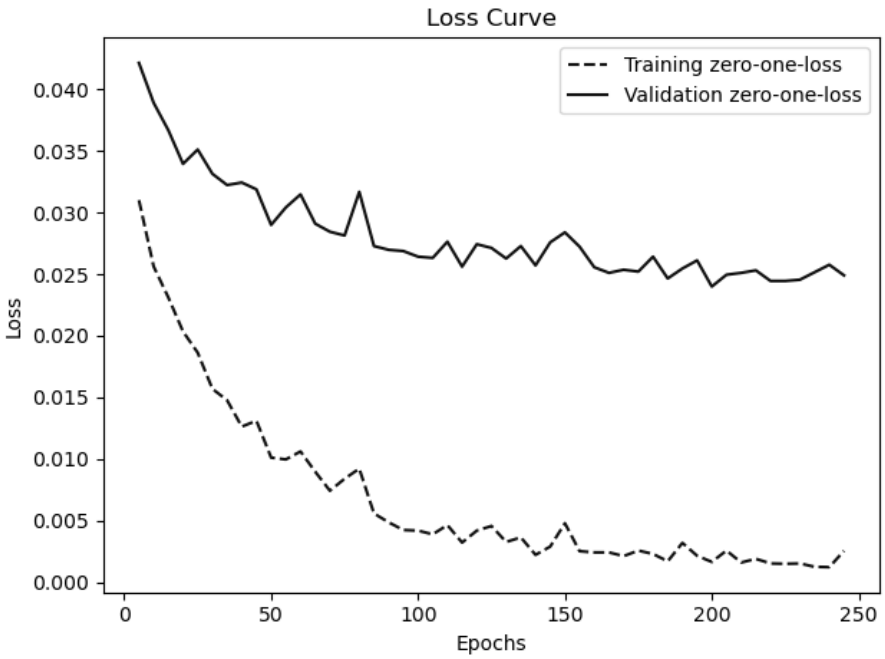

Fig. 2: Loss curve