TECHNISCHE UNIVERSITÄT DRESDEN

Professur für Informationstheorie und maschinelles Lernen, Fakultät Elektrotechnik und Informationstechnik

Prof. Rafael Schaefer, Dr. Martin Mittelbach, Muah Kim, Dr. Rick Fritschek, Max Wolf 17.05.2024

1. Homework of the Course
Einführung in das maschinelle Lernen

*The homework can be submitted by **07.06.2024, 23:59**.*

Please upload your results via the OPAL repository of this course using the available upload folder named "1. Hausaufgabe".

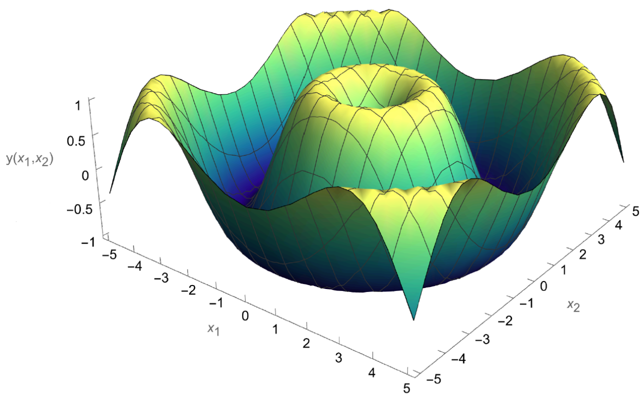
Upload your results as a single ZIP-file with the name firstname-lastname-homework-01.zip.

Homework Problem 1: (Function approximation with two-layer perceptron)

The goal in this problem is to approximate the function

$$y(x_1, x_2) = \sin\left(\left(x_1^2 + x_2^2\right)^{\frac{7}{12}}\right), \qquad (x_1, x_2) \in [-5, 5] \times [-5, 5]$$

with a two-layer perceptron.



The specifications of the two-layer perceptron are as follows:

- The number of nodes $N^{(1)}$ in the hidden layer is a free parameter.
- The Sigmoid-function is used as activation function in the hidden layer.
- No activation function is used in the output layer.

The quality of the approximation is quantified with the quadratic loss function.

For the network training consider an input dataset $[(x_1, x_2)]$ with $(x_1, x_2) \in [-5, 5] \times [-5, 5]$, where the step size for each coordinate is 0.1.

a) Derive the update equations for the network parameters (weights and biases) of the output layer and the hidden layer required to train the network with the gradient decent method. Report your derivations.

b) Implement the two-layer perceptron and the training algorithm using only the NumPy-library (i. e., without using the PyTorch library). Devide the input data into mini-batches for the network training.

c) Use the implementation of Task b) to find network parameters for which the empirical loss function satisfies

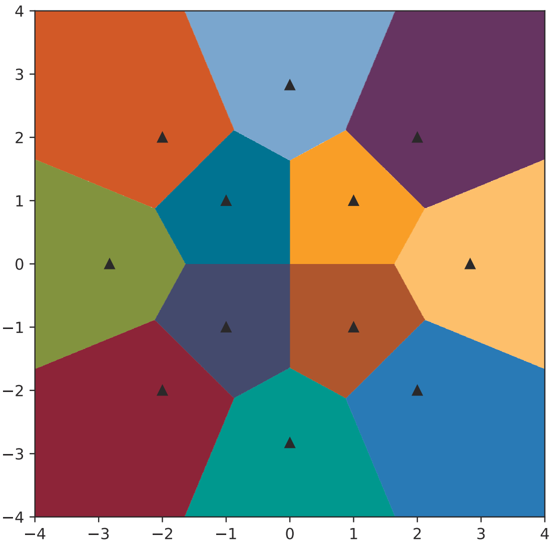
$$\bar{\mathcal{L}}\big([y], [\hat{y}]\big) \leq 0.002.$$

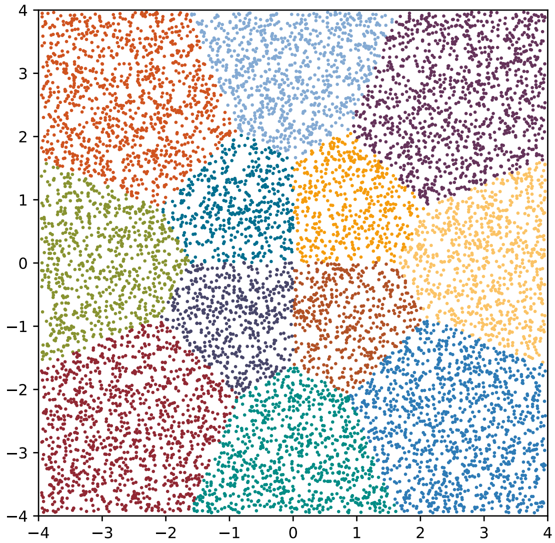
Choose the number $N^{(1)}$ of nodes in the hidden layer as small as possible.
Save the learned network parameters and the used hyper-parameters in an appropriate form and provide the file as part of your solution.

d) Illustrate the function $\hat{y}(x_1, x_2)$ generated by the network for the parameters determined in Task c) together with the "true" function $y(x_1, y_2)$ for the extended input data range $(x_1, x_2) \in [-6, 6] \times [-6, 6]$. Calculate the empirical loss $\bar{\mathcal{L}}\big([y], [\hat{y}]\big)$ for the extended input data range (step size of 0.1 for each coordinate).

Homework Problem 2: (Two-layer perceptron for classification)

We consider a partition of the two-dimensional set $R = [-4, 4] \times [-4, 4] \subset \mathbb{R}^2$ into 12 disjoint subsets $R_1, R_2, ...R_{12}$ as illustrated in the left figure, where each subset is represented by a colored region.





The regions are defined by the minimum Euclidian distance to the following constellation points

$$\left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ -2 \end{pmatrix}, \begin{pmatrix} -2 \\ 2 \end{pmatrix}, \begin{pmatrix} -2 \\ -2 \end{pmatrix}, \begin{pmatrix} \sqrt{8} \\ \sqrt{8} \end{pmatrix}, \begin{pmatrix} \sqrt{8} \\ -\sqrt{8} \end{pmatrix}, \begin{pmatrix} -\sqrt{8} \\ \sqrt{8} \end{pmatrix}, \begin{pmatrix} -\sqrt{8} \\ -\sqrt{8} \end{pmatrix} \right\},$$

which are marked as black triangles in the left figure.

The goal in this problem is to classify the points of the region R according to the specified subregions $R_1, R_2, ...R_{12}$ using a two-layer perceptron.

a) In the right figure a set of 10000 points is illustrated, which is suitable as training dataset, where the membership to a region is illustrated with the corresponding color from the left figure. The shown dataset was obtained by randomly generating 10000 points from the set R drawn from a uniform random distribution on R and determining the region membership according to the above specifications.
Using this approach generate a training dataset with 10000 points and a validation dataset with 5000 points. Illustrate the generated datasets analog to the right figure.
Hint: The NumPy-function `numpy.random.default_rng().uniform(...)` can be used to generate samples from a uniform random distribution.

b) Design a two-layer perceptron that is suitable for the considered classification task, where the number of nodes in the hidden layer is a free parameter and the class labels are represented by one-hot vectors.

c) Derive the update equations for the parameters of the output layer and the hidden layer required to train the network with the gradient decent method using the cross-entropy as loss function. Report your derivations.

d) Implement the two-layer perceptron and the training algorithm using only the NumPy-library (i. e., without using the PyTorch library). Devide the input data into mini-batches for the network training.

e) Use the implementation of Task d) and the training dataset of Task a) to find network parameters such that the classification accuracy of the validation dataset is at least 99%. Choose the number $N^{(1)}$ of nodes in the hidden layer as small as possible.
Save the learned network parameters and the used hyper-parameters in an appropriate form and provide the file as part of your solution.

f) Illustrate in a suitable way the classification output of the trained network for (test) input data from the set R and from the extended set $[-5, 5] \times [-5, 5]$.