Machine Learning

Spring Semester 2021
Prof. Dr. Peter Zaspel
Abhieshree Dhami, Kristijan Spirkoski, Kristian Sterjo

JACOBS UNIVERSITY

Assignment Sheet 12.

Submit on **Voluntary: Tuesday, May 11, 2021, 10:00**.

This assignement sheet does not count towards the bonus achievement!

Excercise 1. (Visual representation of the MLP model)

Follow Example 11.3 from the lecture notes and draw a multilayer perceptron with input dimension $D = 2$, output dimension K =2, and $L = 4$ hidden layers. For the four hidden layers, it holds $M_1 = 2$, $M_2 = 4$, $M_3 = 3$ and $M_4 = 3$. Do not forget to also draw the intercept.

(4 Points)

Excercise 2. (Evaluating an MLP model)

You are given a multilayer perceptron with $D = 2$-dimensional input, $K = 1$-dimensional ouptut and two hidden layers with $M_1 = 3$ and $M_2 = 2$ that follows the simplified Definition 11.4 from the lecture notes. In the hidden layer, it further uses the sigmoid activation. The output activation is the identity function.
In this task, it is assumed that this artificial neural network model has been already trained, such that is has the weights given in matrices

$$\mathcal{W}^{(1)} = \begin{pmatrix} 0.2 & 0.1 \\ 0.3 & 0.0 \\ 0.1 & 0.4 \end{pmatrix}, \mathcal{W}^{(2)} = \begin{pmatrix} 0.1 & 0.2 & 0.5 \\ 0.2 & 0.1 & 0.4 \end{pmatrix}, \mathcal{W}^{(3)} = \begin{pmatrix} 0.1 & 0.1 \end{pmatrix}.$$

Evaluate the MLP given above for the input $\mathbf{x} = (1, 2)^\top$. Make sure to not only give the values at the output layer but also the intermediate results at the hidden layers.

(4 Points)

Excercise 3. (Backward pass for classification)

In Theorem 11.1 from the lecture notes, we give the backward pass that is used to compute the gradients in the backpropagation algorithm. Here, we assume that even in the last layer, we have an activation that does not couple between the units in that layer. Hence, we explicitly exclude the case of the softmax activation in the output layer.
Your task is do derive a modified version of Theorem 11.1, in which we explicitly use the softmax activation in the last layer. *Hint: You only have to modify the step for calculating the* $\delta_m^{(L+1)}$.

(4 Points)

Programming Exercise 1. In this task, we implement SPAM classification using the multilayer perceptron. To this end, we start from Example 11.4 from the lecture notes, which is available as Jupyter notebook. In that example, we train the MLP for a regression task on the known energy efficiency data set. Modify the example appropriately, to instead apply it for classification on the Spambase Data Set.
Note that this task requires a little bit of documentation reading with respect to the machine elarning framework *Keras*. Hence, familiarize yourself with that framework to a level such that you will be able to carry out the above task.
As output of the task, you are supposed to provide a training / generalization error graph, similar to the one in Example 11.4. However, instead of giving the L_2 loss as error measure, use the so-called *accuracy* of classification, i.e. the fraction of properly classified samples within the total number of samples.
Reference solutions will only be provided in Python+Matplotlib. The submission format for Python is a Jupyter notebook. The submission format for C/C++ is standard source files. Choose an appropriate format for the Gnuplot-related submission.

(4 Points)