Institute for Computer Science VI, Autonomous Intelligent Systems, University of Bonn

Dr. N. Goerke
Friedrich-Hirzebruch-Allee 8, 53115 Bonn, Tel: +49 228 73-4167
E-Mail: goerke (at) ais.uni-bonn.de
http://www.ais.uni-bonn.de/WS2223/4204_L_NN.html

Exercises for module Technical Neural Networks (MA-INF 4204), WS22/23

Assignments Sheet 1, due: Monday 24.10.2022

17.10.2022

Group	Name	1	2	3	4	5	6	7	$\sum$ Sheet 1

Remark:
Please upload the solutions in eCampus before the start of the lecture (before 10:00).

Assignment 1 (2 Points)

Name and describe (briefly) the 4 important basic principles of a technical neuron that have been taken from biology.

Assignment 2 (1 Point)

Name and characterize (in one sentence each) the **five** historical phases of neural network research.

Assignment 3 (2 Points)

Write down the δ-rule.
Explain **each** part of the formula with a short sentence.

Assignment 4 (2 Points)

A given MLP with two hidden layers, N-H1-H2-M MLP shall be replaced by a second MLP (N-H-M) with only one single hidden layer, but (almost) the same number of weights.
Derive a formula for the number H of neurons in that hidden layer, and compute a value for H to replace a 5-21-30-4 MLP. Hint: please do not forget the BIAS.

Assignment 5 (2 Points)

Show by calculation that the first derivatives of the two typical transferfunctions for MLPs (*tanh* and *logistic function*) can be expressed by the transferfunctions themselves.

Assignment 6 (4 Points)

Prove in a strict formal way, **analytically**, that a simple Perzeptron (with step function) without a hidden layer is not capable to implement the Boolean function XOR.

Assignment 7 (4 Points)

Derive a new learning rule * for a Multi-Layer-Perceptron.
Start from the new objective function (cost function, error function) E^* and derive the new learning rule in analogy to Backpropagation of Error. Write down all calculation steps, and explicitly write down the formulas for calculating the δ^* in output- and hidden layer.

$$ {}^pE^*(w_{ij}) = \frac{1}{2} \sum_{m=0}^{M} ({}^p\hat{y}_m \; - \; {}^py_m)^4 $$

Programming-Assignment PA-A (5 Points, due 24.10.2022)

Implement a 2-layer Multi-Layer-Perzeptron (one input-layer, one output-layer) as a Python program (Python version 3.10). Do not use libraries that have pre-implemented neural network structures. The Perzeptron shall get an N-dimensional real-valued input $\mathbf{X}$, and produce an M-dimensional real valued output $\mathbf{Y}$. The M ouput neurons shall have a BIAS-weight for implementing the thresholds, and the fermi function as transferfunction f. (N shall be less than 101, and M less than 30), initialize all weights randomly between $-0.5 \leq w_{n,m} \leq +0.5$

Implement further the training of the weights of the Perzeptron using the delta-rule with patterns (${}^pX$, ${}^pY$) that have been read in from a file named `PA-A-train.txt` (P shall be less than 200).

In addition, make sure that your program is running correctly, is producing the required results, and that your source code contains valid, and useful comments.