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

Dr. N. Goerke
Endenicher Allee 19a, 53115 Bonn, Tel: +49 228 73-4167
E-Mail: goerke_at_ais.uni-bonn.de
http://www.ais.uni-bonn.de/WS1819/4204_L_NN.html

New Lecture Location: Meckenheimer Allee 176, HS-IV

Exercises for module Technical Neural Networks (MA-INF 4204), WS1920

Exercises sheet 4, due: Monday 4.11.2019

fictitious bid

28.10.2019

Group	Name	20	21	22	23	24	25	26	$\sum$ Sheet 4

Assignment 20 (2 Points)

Describe and draw a typical learning curve for an MLP that is trained with Backpropagation of Error, single step learning strategy. Make sure, that the diagram is completely labeled.

Assignment 21 (2 Points)

Draw the graph of an (unrealistic) learning curve $f(z) = 2^{(2-z+4*sin(z))}$ with 4 different ways to scale the axes.
A: both axes scaled linear; B: one axis linear and one logarithmic, C: the other axis logarithmic and the first one linear, and D: both axes with logarithmic scaling.
What does the different scaling tell us with respect to a learning curve and the learning process?

Assignment 22 (1 Point)

Compare the two Backpropagation of Error variants: *single step learning* and *cumulative learning*. Describe and compare advantages and disadvantages of the two methods.

Assignment 23 (3 Points)

Explain *Resilient Prop* (*R-Prop*). Describe the idea, and motivate the formulas to adjust the learning rate. It is not necessary to derive the fomulas, but you should know what each term within the formulas is standing for.
Please use scientific citation of the literature that you have used for this task.

Assignment 24 (1 Point)

Explain the method *Flat-Spot-Elimination* for Backpropagation of Error.
Please cite the literature you have used in a scientific way.

Assignment 25 (2 Points)

Explain how an MLP can solve the encoder-decoder tasks: *8-3-8* and *8-2-8*.
Explain further why an MLP is not capable to solve the *8-1-8* task.

Assignment 26 (4 Points)

Calculate, how to change the **input vector** ${}^p\mathbf{X}$ of a 3-layer, N-H-M MLP to decrease the single error ${}^pE$ using gradient descent.
All weights $w_{i,j}$ of the network are to remain constant, the transfer functions in hidden layer and output layer are the hyperbolic tangent.

The way how to change the input values can be determined in analogy to the derivation of backpropagation of error.
Derive the formulas for a 3-layer N-H-M MLP to calculate the partial derivative of the ${}^pE({}^px_n)$ with respect to components ${}^px_n$ of the N-dimensional input vector ${}^p\mathbf{X}$.

$$\frac{\partial\, {}^pE({}^px_n)}{\partial\, {}^px_n} = ?$$