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/WS1920/4204_L_NN.html

New Lecture Location: Meckenheimer Allee 176, HS-IV

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

Exercises sheet 5, due: Monday 11.11.2019

4.11.2019

Group	Name	27	28	29	30	31	$\sum$ Sheet 5

Assignment 27 (4 Points)

Compare the two neural network paradigms Multi-Layer-Perzeptron (MLP) and Radial-Basis-Function Network (RBF).
Please name explicitly **all** aspects that are common, and **all** the aspects that are different, write a brief statement explaining each difference.
Try to discuss, or argue, what kind of applications are better suited for MLPs, and which one are better for RBF-networks.

Assignment 28 (2 Points)

Explain the three philosophies (including variants of them) to set, or adjust the centres $\mathbf{C}_k$ and sizes s_k of RBF-networks.

Assignment 29 (3 Points)

A N-K-M RBF Network (with fixed centers, and fixed width) has the task to implement a mapping from the N-dimensional input space, to an M-dimensional output space. A total of $P > K$ trainingspatterns $({}^p\mathbf{X}, {}^p\hat{\mathbf{Y}}({}^p\mathbf{X}))$ is available. To determine the weights $\mathbf{G}$ between RBF-layer and output layer the *Moore-Penrose-Pseudoinverse* shall be used.
Determine, as precise as possible, the total number of multiplications with respect to P, N, K and M that are necessary to calculate all weights $g_{k,m}$.
Remark: the $\mathcal{O}$-notation, with only the largest exponent, is not specific enough.

Assignment 30 (2 Points)

Construct a 2-K-1 RBF-network that implements the Boolean function XOR. Please draw the network, label all parameters, and give the values of the centers $\mathbf{C}_k$, the width of the transfer functions σ_k and the weights $g_{k,m}$ between RBF-layer and output neuron.

Assignment 31 (4 Points)

Calculate the partial derivatives of the global error F of a N-K-M- RBF-Network with respect to the centers $\mathbf{C}_k = (c_{1k}, \ldots, c_{Nk})$ and with respect to the sizes s_k.

The RBF neurons have the Euclidean distance, and Gaussian bell function with size s_k as transferfunction.

$$F = \frac{1}{2} \sum_{p=1}^{P} \sum_{m=1}^{M} \left({}^p\hat{y}_m - {}^p y_m \right)^2 \qquad\qquad z_k({}^p\mathbf{X}) = exp\left(- \frac{\|{}^p\mathbf{X} - \mathbf{C}_k\|^2}{2 s_k^2} \right)$$

Programming-Assignment PA-C (10 Points, due 18.11.2019, 2 weeks)

Implement (in C, C++, Java or Python) a Radial-Basis-Function network (RBF-network), a method to adjust the positions $\mathbf{C}_k$ and widths s_k of the centers, a method to adjust the weights w_{km} and a program that demonstrates the functionality of your RBF network and the implemented methods.

Radial-Basis-Function-Net, RBF-Net:
The RBF-network shall have N inputs, K RBF-neurons, and M output neurons. Take the Gaussian function (bell-curve) as radial-transferfunction for the RBF-neurons.

The weights w_{km} between RBF-neurons and output neurons shall be initialized by random values from the interval of -0.5 to $+0.5$. The starting value(s) (SEED) for the pseudo-random generator(s) shall be set by you.

Method to adjust the RBF-centers and RBF-widths:
Implement a way to adjust the RBF-centers $\mathbf{C}_k$ and the RBF-widths s_k. Choose one of the methods that have been described in the lecture and tell which one you have chosen.

Method to adjust the weights W using gradient descent:
Implement a gradient descent based (single-step or cumulative, your choice) method to train the weights w_{km} between the RBF-neurons and the output neurons.
The learning rate η shall be identical for all output neurons.
Read in the training data from `training.dat`, the test data from `test.dat`, using the data format from PA-B, and create a `learning.curve` to monitor the learning progress.