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/WS/4204_L_NN.html

Exercises for module Technical Neural Networks (MA-INF 4204), WS24/25

Programming Assignments C, due: Monday 18.11.2024

4.11.2024

Programming-Assignment PA-C (10 Points, due 18.11.2024)

Implement (in Python; on your own) 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.
Generate your own training data, and/or use data from PA-A and PA-B. Read in the training data from `training_data.txt`, the test data from `test_data.txt`, and create a `learning_curve.txt` to monitor the learning progress.

EXTRA: only for experienced students:
Implement the method to directly calculate the weights w_{km} between the RBF-neurons and the output neurons using the Moore-Penrose-Pseudo-Inverse; (you can use libraries for that). Compare the results with the gradient descent based results. Take data sets with $P = K$, one with P slightly larger than K and one with $P >> K$.
Discuss the obtained results w.r.t. achieved accuracy and the time to get the results.