Ludwig-Maximilians-Universitaet Muenchen 06.05.2015
Institute for Informatics
Prof. Dr. Volker Tresp
Gregor Jossé
Johannes Niedermayer

Machine Learning and Data Mining

Summer 2015

Exercise Sheet 2

Presentation of Solutions to the Exercise Sheet on the 04.05.2015

Exercise 2-1 The Perceptron in more than two Dimensions

a) The numbers vom 0 through 9 were represented by pixel arrays in the lecture, the corresponding data matrix can be found in the file `numberMatrix.RData`. Use this data to train a perceptron such that it distinguished between odd and even numbers. Vary w and η. Additionally, answer the question if the perceptron learning rule terminates for the problem "is a multiple of 3"?

b) What is the complexity of training a perceptron for an M-dimensional dataset of with N input patterns? What is the cost of a predicition after having trained the perceptron?

Exercise 2-2 Linear Regression

Let X be a variable providing the data and its occurrences Y:

x	3	4	5	6	7	8
y	150	155	150	170	160	175

a) Presume the model exhibits the following linear relation:
$y_i = \beta_0 + \beta_1 x_i = x^T w$
Use the least squares-estimator introduced in the lecture to determine w.

b) Now, presume the non-linear relation
$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 = x^T w$
and, again, determine w.

c) How could the empiric quadratic error between model and data be visualized? Explain and sketch your suggestion in two as well as in three dimensions on arbitrary data.

d) Which of the models a) and b) is better? Compute the average quadratic error and evaluate the models. How could a better model be realized?

Hint: Matrix arithmetic need not be done manually. You can use `R`, Maple, Octave or Python.

Exercise 2-3 Regularisation / Overfitting

a) What is *overfitting* and how does it occur?

b) How can a model be identified as "overfitted"?

c) How can overfitting be avoided?

Exercise 2-4 Curse of Dimensionality vs. Kernel Trick

a) Explain the term *curse of dimensionality*.
When does it occur, how can it be avoided?

b) Explain the term *Kernel Trick*.
How can it be used, what is its connection to the *curse of dimensionality*?

Exercise 2-5 Basis Functions of Neural Networks

Given a test vector $\mathbf{x}_i$, the output of a neural network is defined as

$$f(\mathbf{x}_i) = \sum_{h=0}^{M_\phi - 1} w_h \phi_h(\mathbf{x}_i, \mathbf{v}_h).$$

The weights of the neurons can be learned by employing the back-propagation rule with sample-based gradient descent. In the lecture neural networks with sigmoid neurons have been introduced, but it is possible to employ different basis functions:

a) Which properties do these basis functions have to fulfill?

b) Can a linear combination $\phi(\mathbf{x}_i, \mathbf{v}_h) = z_h = \sum_{j=0}^{M} v_{h,j} x_{i,j}$ be suitable for this?

c) Is the number of parameters for $\phi(\mathbf{x}_i, \mathbf{v}_h)$ limited? Could several different basis functions be used for the same neural network?