Ludwig-Maximilians-Universitaet Muenchen 16.04.2014
Institute for Informatics
Prof. Dr. Volker Tresp
Gregor Jossé

Machine Learning and Data Mining

Summer 2014

Exercise Sheet 2

Presentation of Solutions to the Exercise Sheet on the 08.05.2014

Aufgabe 2-1 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 the work stations in the CIP-pool: `R` or Maple (call: `xmaple`).

Aufgabe 2-2 Recap: Vector Calculus

Compute $\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}}$ the functions below. *Hint:* For a function $g(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$ miti $\mathbf{x} \in \mathbb{R}^n$ holds:

$$\frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_1} \\ \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_2} \\ \vdots \\ \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}_n} \end{bmatrix}.$$

a) $g(\mathbf{x}) = \sum_{i=1}^{n} \mathbf{x}_i$,

b) $g(\mathbf{x}) = \langle \mathbf{x}, \mathbf{x} \rangle$, the standard scalar product of $\mathbf{x}$ with itself,

c) $g(\mathbf{x}) = (\mathbf{x} - \mu)^2$ für $\mu \in \mathbb{R}^n$.

Optional:

Aufgabe 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?

Optional:

Aufgabe 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*?

Aufgabe 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?

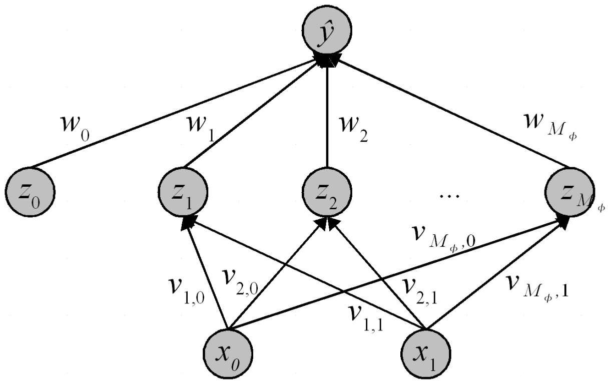
b) Can a linear combination $\phi(\mathbf{x}_i, \mathbf{v}_h) = z_h = \sum_{j=0}^{M} v_{h,j} x_{i,j}$ be suitable for this?

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?

b.w.

Aufgabe 2-6 A simple Neural Network

The illustration below depicts, a two-layered neural network with inputs $x \in \mathbb{R}$ and for each input one bias $x_0 = z_0 = 1$ (i.e. $\mathbf{x}_i = (1, x_{i,1})^T$) in the input as well as the hidden layer.



As function of the hidden neurons we employ a sigmoid, i.e.

$$z_h = \phi(\mathbf{x}_i, \mathbf{v}_h) = \frac{1}{1 + \exp\left(-\sum_{j=0}^{M} v_{h,j} x_{i,j}\right)},$$

the output neuron $\hat{y}$ is, as usual, a linear combination.

a) Prove that the following holds: $\frac{\partial z_h}{\partial v_{h,j}} = x_{i,j} \cdot z_h \cdot (1 - z_h)$

b) Express the maximal value of $\hat{y}$ subject to $\mathbf{w}$, if all original weights are w_h $(h \in \{0, \ldots, M_\phi\})$ positive. What's the minimal value?

c) If $v_{h,j} = 0$ for all $j \in \{0, \ldots, M\}, h \in \{1, \ldots, M_\phi\}$, then what is $\hat{y}$? Which functions describe $\hat{y}$ if all $v_{h,j} = c, c \neq 0$?