Ludwig-Maximilians-Universitaet Muenchen
Institute for Informatics
Prof. Dr. Volker Tresp
Rajat Koner
Sebastian Schmoll

Machine Learning

Summer 2021

Exercise Sheet 2

Exercise 2-1 The ADALINE learning rule

The *ada*ptive *lin*ear *e*lement (ADALINE) model uses the *mean squared error* cost function

$$\text{cost} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \,,$$

for N training set elements, where y_i is the true and $\hat{y}_i$ is the predicted value for the i-th sample. In contrast to the simple perceptron, classification is not realized by the signum-function. (As a reminder: M is the number of input features of patterns $x_i \in \mathbb{R}^M$ and the dimensionality of the weight vector $w \in \mathbb{R}^M$; remember that we can append $x_0 = 1$ to be constant and corresponds to the bias or offset.)

a) Deduce the gradient descent-based learning rule (or: adaption rule) for the ADALINE process (analoguously to the perceptron learning rule).

b) Specify the corresponding pattern-based (SGD) learning rule.

c) What advantages do pattern-based learning rules have?

d) Name the most distinctive characteristics between the ADALINE model and the perceptron model.

Exercise 2-2 Regularization / 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 2-3 Regression w/ Regularization

Considering a regression problem and we are given the following cost function:

$$J(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}(f_{\boldsymbol{w}}(x_i) - y_i)^2$$

The regularization term $R(\boldsymbol{w}) = \lambda \cdot ||\boldsymbol{w}||^2$ is added to the model's cost function. Given this modification, derive its update rule.

Exercise 2-4 PyTorch Basics + Linear Regression (optional)

On the course website you will find a jupyter notebook leading you through some basic tasks in pytorch and we will start with a simple implementation of a linear regression problem. Try to make yourself familiar with the basic learning procedure in PyTorch.