Artificial Neural Networks and Deep Learning Summer 2017
Christian Borgelt from 2017.05.30

Exercise Sheet 3

Exercise 14 Method of Least Squares/Regression

Determine the best fit straight line $y = a + bx$ (regression line) for the following two data sets with the help of the method of least squares:

a) $(-2, 0), (0, 1), (1, 3), (2, 5)$
b) $(-1, 3), (1, 2), (2, 0), (4, -2)$

Draw the data points and the best fit straight line!

Exercise 15 Method of Least Squares/Regression

Determine and draw a best fit parabola $y = a + bx + cx^2$ for the data of Exercise 14a!

Exercise 16 Logistic Regression

The following table shows the number of American intercontinental ballistic missiles (ICBMs) in the 1960s:

year, x	1960	1961	1962	1963	1964	1965	1966	1967	1968	1969
number, y	18	63	294	424	834	854	904	1054	1054	1054

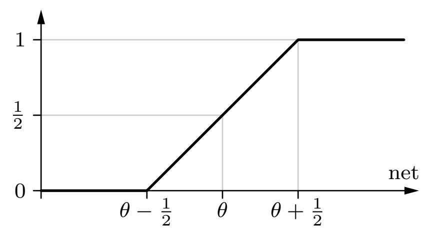
Find a best fit curve with the help of logistic regression ($Y = 1060$)!
Draw the original data and sketch the curve $y = \frac{1060}{1+e^{a+bx}}$!

Exercise 17 Gradient Descent

Approximate the minimum of the function $f(x_1, x_2) = 2x_1^2 - 2x_1 + x_2^2 - x_2$ with the help of gradient descent! Start from the initial approximation $(x_1, x_2) = (0, 0)$ and use $\eta = 0.2$ as the step width parameter!

Exercise 18 Gradient Descent

Consider a 2-layer perceptron with n inputs and one output, in which the output neuron has the weighted sum of the inputs as the network input function, a semi-linear function

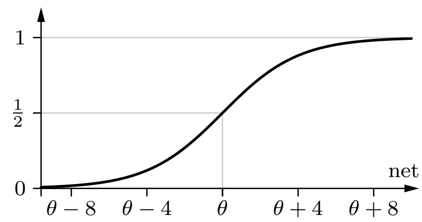


$$f_{\text{act}}(\text{net}, \theta) = \begin{cases} 1, & \text{if net} > \theta + \frac{1}{2}, \\ 0, & \text{if net} < \theta - \frac{1}{2}, \\ (\text{net} - \theta) + \frac{1}{2}, & \text{otherwise,} \end{cases}$$

(see diagram) as the activation function and the identity as the output function. Derive the rule for changing the weights that results from an approach based on gradient descent if the network error is computed as the sum (over all patterns) of the squared differences between desired and actual output!

Exercise 19 Gradient Descent

Consider a 2-layer perceptron with n inputs and one output, in which the output neuron has the weighted sum of the inputs as the network input function, a logistic function



$$f_{\text{act}}(\text{net}, \theta) = \frac{1}{1 + e^{-(\text{net} - \theta)}}$$

(see diagram) as the activation function and the identity as the output function. Derive the rule for changing the weights after a single training pattern was presented that results from an approach based on gradient descent if the network error is computed as the absolute value of the difference between desired and actual output! What changes to the backpropagation procedure would be necessary (for a multi-layer perceptron)?

Exercise 20 Error Backpropagation

Consider a 3-layer perceptron with 4 inputs, 5 hidden neurons and 3 output neurons, which is meant to solve a 4-dimensional three class problem. To train the network, error backpropagation is to be used. The output neurons have the hyperbolic tangent (*tangens hyperbolicus*), the hidden neurons the logistic function as their activation function.

a) Define reasonable desired output values for the training data for each of the three classes!

b) Derive the formula for the weight change for the connection from the third hidden neuron to the second output neuron! Be as precise as possible!

c) Derive the formula for the weight change for the connection from the first input neuron to the third hidden neuron!

Exercise 21 Deep Learning: Dropout

a) Consider a 12-layer perceptron with 10 input neurons, 10 neurons in each hidden layer and 1 output neuron. How many weights in total have to be trained in this network?

b) The network is to be trained with the dropout approach. How many weights have to be considered per training step if due to the dropout (on average) 5 neurons are deactivated per layer?

c) How does the (expected) number of weights to consider depend on the dropout rate?