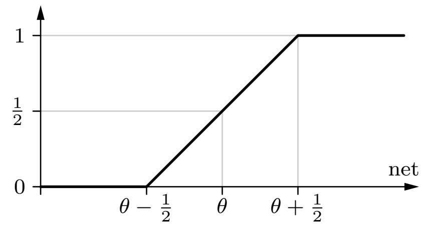
Artificial Neural Networks and Deep Learning Summer 2018
Christian Borgelt and Christoph Doell from 2018.06.12

Exercise Sheet 6

Exercise 23 Gradient Descent

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

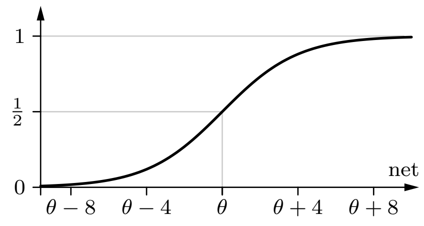


$$f_{\text{act}}(\text{net}, \theta) = \begin{cases} 1, & \text{if net} > \theta + \frac{1}{2}, \\ 0, & \text{if net} < \theta - \frac{1}{2}, \\ (\text{net} - \theta) + \frac{1}{2}, & \text{otherwise,} \end{cases}$$

(see diagram) as its activation function and the identity as its 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 24 Gradient Descent

Consider a 2-layer perceptron with n inputs and one output, in which the output neuron has the weighted sum of its 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 its activation function and the identity as its 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 25 `xmlp/wmlp`: Biimplication

Start one of the programs `xmlp` (for GNU/Linux) or `wmlp` (for Windows) (these programs are available at `http://www.borgelt.net/mlpd.html`), which show the training process of a 3-layer perceptron. Choose `Actions > Biimplication` or `Actions > Exclusive Or`. Following the explanations in the lecture concerning the biimplication problem and its solution, the representation should be comprehensible. You can change the parameters of the training procedure via `Settings > Parameters...` Execute the training multiple times and experiment with the values of the parameters (momentum term, learning rate and range of the initial weights).

Exercise 26 `xmlp/wmlp`: Function Approximation

With the two programs `xmlp` or `wmlp` that were used in Exercise 25 it is also possible to visualize the approximation of two simple real-valued functions. Choose `Actions > Function 1` or `Actions > Function 2`. For both functions, execute the training multiple times. How are the functions approximated by the neural network? Explain the meaning of the weights and the threshold/bias values. Also, experiment again with the values of the parameters (momentum term, learning rate and range of the initial weights). What effects can be achieved?