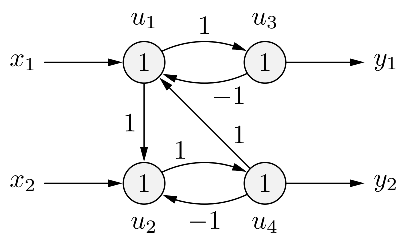
Artificial Neural Networks and Deep Learning Summer 2017
Christian Borgelt from 2017.05.16

Exercise Sheet 2

Exercise 7 Update Order

Consider the following network of threshold logic units:



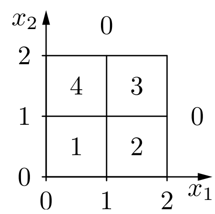
Show that it depends on the update order of the threshold logic units whether the network reaches a stable state if the the inputs $x_1 = 0$ and $x_2 = 1$ are presented!

Exercise 8 Training Threshold Logic Units

Show how a theshold logic unit is trained (with the delta rule) to compute the Boolean function $x_1 \to x_2$! (Preferably with the help of a table that has columns for the values of x_1, x_2, $o = x_1 \to x_2$, $\vec{x} \cdot \vec{w}$, y, e, $\Delta\theta$, Δw_1, Δw_2, θ, w_1 and w_2.) Start from the initial state $\vec{w} = (0, 0, 0)$ of the (extended) weight vector and use 1 as the learning rate. Interpret the learning result geometrically!

Exercise 9 Function Approximation

Sketch a neural network with two inputs that produces
the value 1 for $(x_1, x_2) \in [0, 1) \times [0, 1)$,
the value 2 for $(x_1, x_2) \in [1, 2) \times [0, 1)$,
the value 3 for $(x_1, x_2) \in [1, 2) \times [1, 2)$,
the value 4 for $(x_1, x_2) \in [0, 1) \times [1, 2)$
and the value 0 outside of these areas!



Assume that in the output neuron the theshold function is replaced by the identity. (In all other neurons, however, (crisp) threshold functions are to be used.) (Hint: cf. Exercise 5; if necessary, add another layer to produce the different output values.)

Exercise 10 Function Approximation

a) Construct a multi-layer perceptron with about 10 neurons that approximates the function $y = x^2$ in the interval $[0.5, 4.5]$ by a step function.

b) How can the approximation be improved? (State at least two possibilities.)

Exercise 11 Function Approximation

We consider the indicator function of the rational numbers over the set of the real numbers (also known as the Dirichlet function), that is, the function

$$f : \mathbb{R} \to \{0,1\}, \qquad x \mapsto \begin{cases} 1, & \text{if } x \in \mathbb{Q}, \\ 0, & \text{otherwise.} \end{cases}$$

a) Is it possible to approximate this function with a neural network (multi-layer perceptron) with arbitrary precision?

b) What can be concluded from the result of part a) about the computational capabilities of neural networks?

Exercise 12 `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 13 `xmlp/wmlp`: Function Approximation

With the two programs `xmlp` or `wmlp` that were used in Exercise 12 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?