Einführung in die Künstliche Intelligenz 2013S, 2.0 VU, 184.735

Exercise Sheet 2 - Learning and Neural Networks

You have to tick the prepared exercises in TUWEL at the latest before Friday, 17th May 2013, 13:00 (checkmarks for Exercise Sheet 2). Belated submission will be ignored.

Exercise 1 (Decision Tree Learning):

a) Consider a classification problem with three attributes A, B, C with the domains $V(A) = \{a_1, a_2\}$, $V(B) = \{b_1, b_2\}$, $V(C) = \{c_1, c_2\}$ and two classes $K = \{T, F\}$. For this problem, build a decision tree using the rule from the lecture of choosing the attribute maximising information gain in each step.

	A	B	C	Class
1	a_1	b_2	c_1	F
2	a_2	b_2	c_1	F
3	a_1	b_2	c_2	F
4	a_1	b_1	c_1	T
5	a_1	b_1	c_2	F
6	a_2	b_1	c_2	F

(3 pts)

b) Compute the information gain for an attribute A that takes a different value for each example. What problem do you see? **(1 pt)**

c) Consider again the problem in a). Again construct a decision tree, but in every step choose the argument that maximises the *relative information gain*, i.e., the ratio between their gain and their own intrinsic information.

$$GainR(A) = \frac{Gain(A)}{H(A)}$$

Thereby we have

$$H(A) := \sum_{a \in V(A)} \frac{1}{|E_a|} \log_2 |E_a| \ ,$$

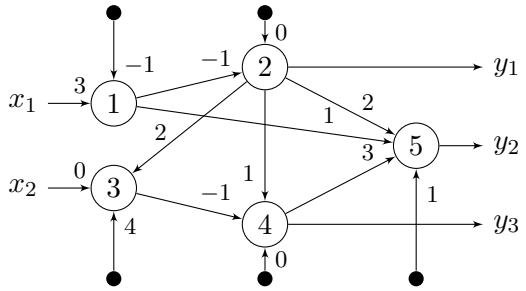
where $V(A)$ denotes the set of possible values for the attribute A and E_a is the set of samples with $A = a$. **(2 pts)**

d) Compute the relative information gain for an attribute A which takes a different value for each example (Assume that you are given $2n$ examples, which are distributed evenly between two classes). Can you spot the advantage of this rule compared to the regular information gain rule? Compare with an attribute B which splits the example set in exactly two halves with the correct classification. **(1 pt)**

Exercise 2 (Neural Networks):

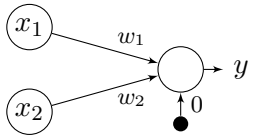
a) Construct a neural network with 4 input nodes x_1, x_2, x_3, x_4 and one output node y which produces a parity bit of the respective input values, namely $y = x_1$ XOR x_2 XOR x_3 XOR x_4 holds. You are allowed to introduce hidden layers with additional nodes of arbitrary depth, but every node should use the threshold function with threshold 0 as activation function. **(2 pts)**

b) Consider a neural network with 5 nodes of the following form:



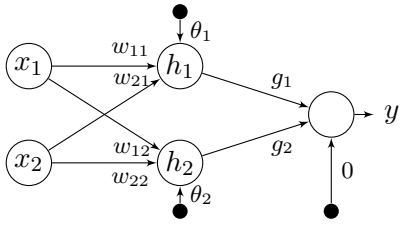
The numbers beneath the arrows denote the respective weights. Nodes 1 and 3 have the *identity function* as their activation function. Nodes 2 and 4 use the threshold function with threshold 0 and node 5 uses a sigmoid function ($1/(1+e^{-x})$). What is the output produced by the network when the input is $(x_1, x_2) = (1, 1)$? **(1 pts)**

c) Consider a 2-layer perceptron with two input neurons and one output neuron of the following form:



Train the perceptron using the *Perceptron learning rule* and the identity transfer function on the following training data: $f(0,0) = 0$, $f(1,0) = 1$ and $f(0,1) = 1$. The weights are initialised to 0 and the learning rate α is 1. The threshold will not be learned and stays 0. **(2 pts)**

d) Consider the following 3-layer perceptron with two input neurons and one output neuron of the following form:



The perceptron uses the *sigmoid function* as activation function for the hidden layer and the *identity function* for the output layer. The six feedforward weights and the two intermediate thresholds are initialised to 0 and should be learned using the *backpropagation learning rule* and a learning rate $\alpha = 1$. Use the data for the XOR function as training data: $f(0,0) = 0$, $f(1,0) = 1$, $f(0,1) = 1$, $f(1,1) = 0$. (Do at least three iterations). **(3 pts)**