

Einführung in Künstliche Intelligenz SS 2019, 2.0 VU, 184.735

Exercise Sheet 2 – Learning and Neural Networks

For the discussion part of this exercise, mark your solved exercises in **TUWEL** until Sunday, June 09, 23:55 CET. The registration for a solution discussion ends on Friday, June 14, 23:55 CET. Be sure that you tick only those exercises that you can solve and explain!

In the discussion, students will be asked questions about their solutions of examples they checked. The discussion will be evaluated with 0-15 points, which are weighted with the fraction of checked examples and rounded to the next integer. There is *no minimum number of points* needed for a positive grade (i.e., you do not need to participate for a positive grade, but you can get at most 85% without doing exercises).

Note, however, that *your registration is binding*. Thus, *if* you register for a solution discussion, then it is *mandatory* to show up. Not coming to the discussion after registration will lead to a reduction of examination attempts from 4 to 2.

Please ask questions in the **TISS** Forum or visit our tutors during the tutor hours (see **TUWEL**).

Exercise 2.1: You and a group of friends decide to estimate how much effort you need to put into the Introduction to Artificial Intelligence course to successfully pass the exam.

You first decide what data is relevant for your decision and after some discussion, you settle for the attributes H (how many hours the student approximately spent studying and preparing for the exam) with the domain $V(H) = \{10, 30, 50\}$, E (whether the student participated in the exercise part) with the domain $V(E) = \{\top, \perp\}$, T (whether the student clarified open questions with the tutors) with the domain $V(T) = \{\top, \perp\}$. Content with your rather simplistic model, you ask some older colleagues who have already taken the course for their experience and collect the following data:

Sample	H	E	T	Passed?
1	10	\top	\top	F
2	30	\top	\top	T
3	50	\top	\top	T
4	10	\top	\perp	F
5	30	\top	\perp	F
6	50	\top	\perp	T
7	10	\perp	\perp	F
8	30	\perp	\perp	F
9	50	\perp	\perp	T

Use the gathered data to construct a decision tree capable of predicting whether a given student will pass their exam, given the values of the attributes you chose. In each step of the construction, choose the attribute that maximizes the information gain, as shown in the lecture.

Exercise 2.2: Construct another decision tree for Exercise 2.1, this time choosing the attribute that maximizes the *relative information gain*, i.e. the ratio between the gain of the attribute and its own intrinsic information.

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

and

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

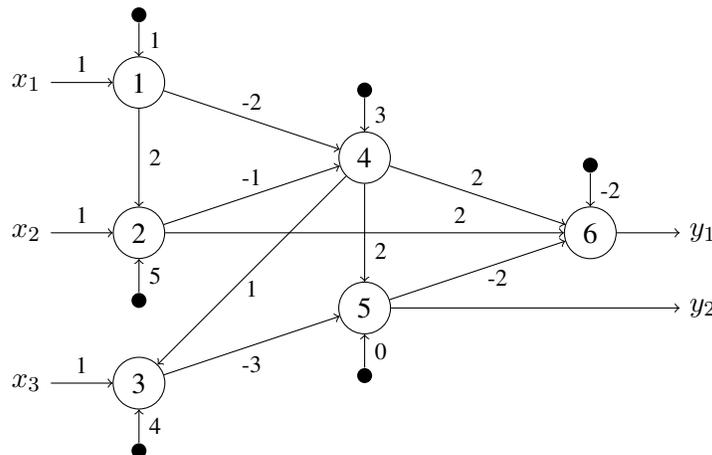
where $V(A)$ denotes the domain of the attribute A , and $|E_a|$ is the number of all samples which have the value a of the attribute A . Furthermore, we define $V := \sum_{a \in V(A)} |E_a|$ to be the size of the set of all samples.

Exercise 2.3: In this exercise we explore some of the problems with decision trees and the ways of dealing with them:

- 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 using the relative information gain rather than the regular information gain rule? Compare with an attribute B which splits the example set in exactly two halves with the correct classification.
- Suppose that you're collecting data for a decision tree. The first two examples you collect have the exact same value for each of the attributes you picked, but their classification is different. You stop collecting further data and ponder the implications of the situation that occurred. Discuss briefly the answers to the following questions:
 - What could be the cause for such a situation?
 - What would the decision tree learning algorithm discussed in the lecture do in this case?
 - What could you do to avoid the problem that arises?

Exercise 2.4: Suppose that an attribute splits the set of examples E into subsets E_k and that each subset has p_k positive and n_k negative examples. Show that the attribute has zero information gain if the ratio $\frac{p_k}{p_k+n_k}$ is equal for all k .

Exercise 2.5: Consider a neural network with six nodes of the following form:



The numbers next to the arrows denote the respective weights. Nodes 1 and 4 have $g(x) = -x$ as their activation function. Node 3 uses the identity function $g(x) = x$. Nodes 2 and 6 use a hard limiter with a threshold of 0.5, while node 5 uses the sigmoid function $g(x) = 1/(1 + e^{-x})$.

What is the output produced by the network when the input is $(x_1, x_2, x_3) = (1, 0, 1)$?

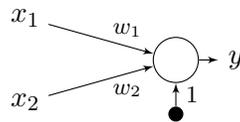
Exercise 2.6: Design a neural network with six binary input signals $S_0, S_1, I_0, I_1, I_2, I_3$ and a single binary output signal I_{out} , which behaves as a 4-bit multiplexer. You are free to determine the structure of the network and the activation function yourself.

As a reminder, a 4-bit multiplexer uses its select inputs (S_0 and S_1) to select and propagate one of the four input values. The remaining input values are ignored. The following truth table illustrates this behavior:

S_1	S_0	I_0	I_1	I_2	I_3	I_{out}
0	0	1	*	*	*	1
0	0	0	*	*	*	0
0	1	*	1	*	*	1
0	1	*	0	*	*	0
1	0	*	*	1	*	1
1	0	*	*	0	*	0
1	1	*	*	*	1	1
1	1	*	*	*	0	0

Note: The * symbol means that the signal at the given position can be either 1 or 0, as its value does not affect the output in any way.

Exercise 2.7: 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 activation function ($g(x) = x$) on the following training data: $f(0, 1) = 3$, $f(1, 0) = 3$ and $f(1, 1) = 5$. The weights are initialized to 0 and the learning rate α is 1. The bias weight will not be learned and stays 1.

Express algebraically the function $f(x_1, x_2)$ that your network has learned after the training is complete. Does this function produce the expected outputs for all three training inputs?

Exercise 2.8: Use a single-layer perceptron with two inputs, one output and a bias weight, using the step function as activation function ($g(x) = 1$ for $x \geq t$ for some threshold t and $g(x) = 0$ otherwise) to express the logical equivalence operator (\equiv) or provide a formal proof that no such construction is possible.

