Exercises for MI

Exercise sheet 8

Thomas Dyhre Nielsen

Note: Some of the exercises below asks you to solve the exercises using Weka. If you feel adventurous (or perhaps would like to get some hands-on programming experience) you are also most welcome to solve these exercises using other (programming) tools such as scikit-learn, which support decision tree learning.

When you have finished with the exercises, you should continue with the exam sheet from the previous years, which can be found at the course's home page.

Exercise 1 Complete the exercises for the last lecture.

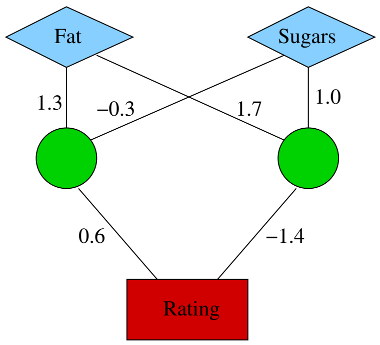
Exercise 2[*]

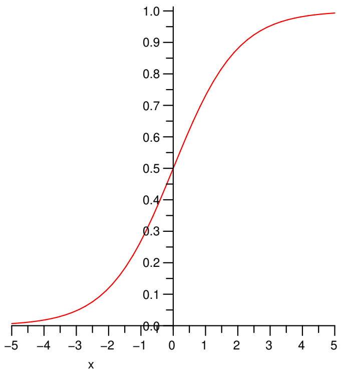
Suppose you want to use a neural network for predicting user preferences based on the data set shown on Slide 7.13. What neural network structure could you use, especially: what would be the input and output units?

Exercise 3[*] Compute for the neural network below the *Rating* output computed for the two inputs

Fat	Sugars
1	5
0	14

The two hidden units have the sigmoid activation function. Values for this function can be either computed precisely according to the definition $\sigma(x) = 1/(1+e^{-x})$, or you can read approximate function values off the plot on the right below. The output unit has the identity activation function, i.e. the output is just the weighted sum of the inputs.





Exercise 4* Assume that we have the following training examples:

X_1	X_2	T
1	1	1
−1	1	−1
1	−1	1
−1	−1	−1

That is, with input $X_1 = 1$ and $X_2 = -1$ we want the output 1.

Consider a perceptron with threshold input 1 and with initial weights $w_0 = 0$,$w_1 = 0$ and $w_2 = 0$.

- Show the first two iterations (as on Slide 8.24) when learning a perceptron (having the sign function as activation function) using learning rate $\alpha = 0.25$ and error function $E = t - o$; t is the desired output and o is the actual output.

Exercise 5 Load the Iris dataset in WEKA (link provided in the exercise description for the previous lecture) and choose the MultilayerPerceptron classifier model. Use Test options: "Use training set". Observe how the performance of the learned model (and the time needed for learning) changes when you modify the following parameters of the learning procedure:

- hidden Layers: this controls the structure of the network (use GUI:true to check).
- trainingTime: controls how many iterations are performed in the weight learning.
- learning rate: controls the stepsize in the gradient descent.

Exercise 6* Complete one more iteration of the back propagation algorithm for the example on slide 08.34.