Exercise 1 :

Suppose you want to use a neural network for predicting user preferences based on the data set in Table 1.

Example	Author	Thread	Length	WhereRead	UserAction
e_1	known	new	long	home	skips
e_2	unknown	new	short	work	reads
e_3	unknown	follow Up	long	work	skips
e_4	known	follow Up	long	home	skips
e_5	known	new	short	home	reads
e_6	known	follow Up	long	work	skips
e_7	unknown	follow Up	short	work	skips
e_8	unknown	new	short	work	reads
e_9	known	follow Up	long	home	skips
e_{10}	known	new	long	work	skips
e_{11}	unknown	follow Up	short	home	skips
e_{12}	known	new	long	work	skips
e_{13}	known	follow Up	short	home	reads
e_{14}	known	new	short	work	reads
e_{15}	known	new	short	home	reads
e_{16}	known	follow Up	short	work	reads
e_{17}	known	new	short	home	reads
e_{18}	unknown	new	short	work	reads
e_{19}	unknown	new	long	work	?
e_{20}	unknown	follow Up	long	home	?
e_{21}	unknown	follow Up	short	home	?

Table 1: The user preference data to be used in Exercise 1.

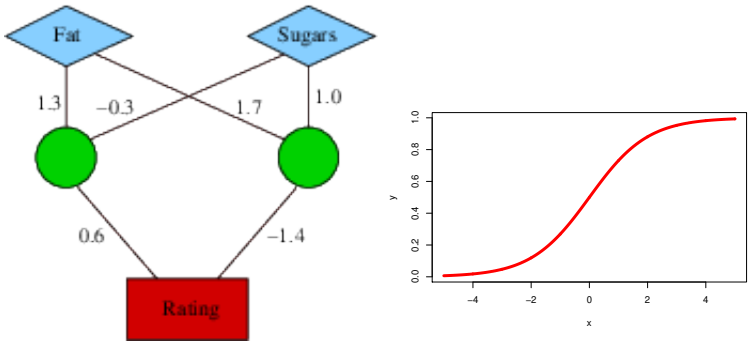
What neural network structure could you use, especially: what would be the input and output units?

Exercise 2 :

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 3 :

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

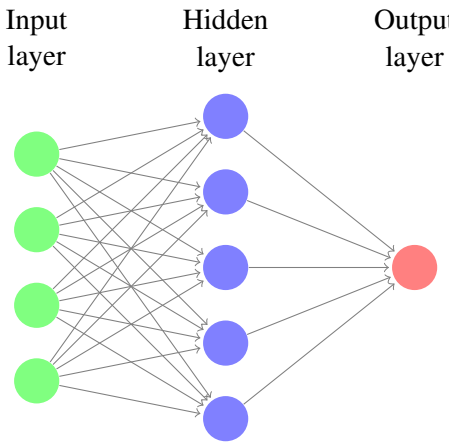
That is, with input $X_1 = 1$ and $X_2 = -1$ we want the output 1.

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

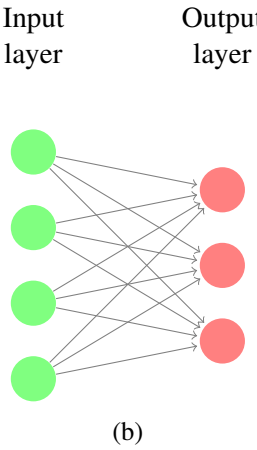
Show the first two iterations of gradient descent when learning a perceptron (having the sign function as activation function) using learning rate $\alpha = 0.25$. Indicate the intermediate steps that need to be computed (e.g. the value of forward propagation, error term and how the weights are adjusted).

Exercise 4 :

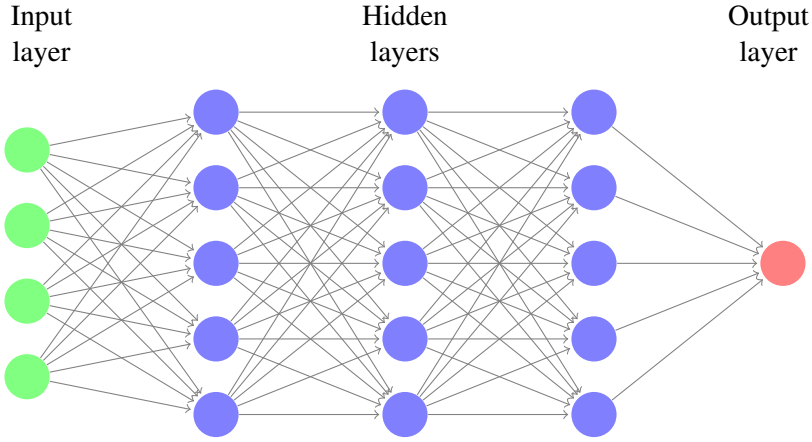
For each of the following neural networks:



(a)



(b)



(c)

indicate:

i) How many parameters need to be adjusted by the learning algorithm?

ii) How many functions are represented by the network?

iii) Can the network represent the Boolean function x_1 xor x_2? Justify your answer.

Exercise 5 :

Experiment with the tensorflow playground

https://playground.tensorflow.org

- Use different data sets and neural network structures (varying the number of layers and neurons). How does the different network structures affect the learning ability? Can you derive any useful insights by considering the features learned at the hidden units.

- What is the effect of varying the learning rate?

Optional: You can also take a look at the questions in this course: https://developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exer

---

Exercise 6 (Optional):

Load the Iris dataset in WEKA (link) 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 7 :

Complete one more iteration of the back propagation algorithm for the example on slide 08.29.
