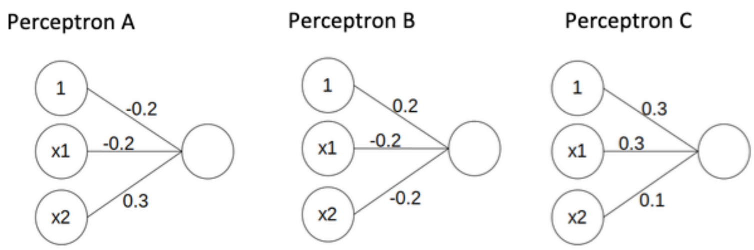
School of Computing and Information Systems
The University of Melbourne
COMP90049 Introduction to Machine Learning (Semester 2, 2020)

Tutorial exercises: Week 7

1. Consider a multiclass classification task using the three perceptrons below:



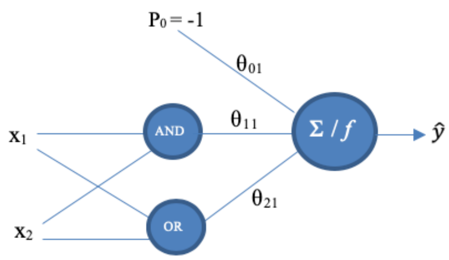
Recall the training procedure for the multi-class perceptron: we predict $\hat{y}$ as the class whose parameters lead to the largest activation. During training, if a wrong class was predicted, we update the parameters activated for the wrong class (decreasing their activation), and the parameters activated for the correct class (increasing their activation).

Following this scheme, do the following:

i. *Training*: For training example $x_{train} = (0,1)$, with class **B**, give the updated weights of each perceptron after processing x. Use learning rate $\eta = 0.1$.

ii. *Testing*: Using the new weights from part (a), given test example $x_{test} = (1, 1)$ with true class **C**, give the class prediction of the group of three perceptrons. Is the prediction correct? Should we change the weights?

2. Consider the two levels deep network illustrated below. It is composed of three perceptrons. The two perceptrons of the first level implement the AND and OR function, respectively.



Determine the weights θ_{11}, θ_{21} and bias θ_{01} such that the network implements the XOR function. The initial weights are set to zero, i.e. $\theta_{01} = \theta_{11} = \theta_{21} = 0$, and the learning rate η (eta) is set to 0.1.

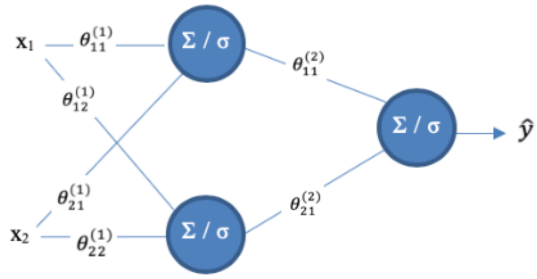
Notes:

- The input function for the perceptron on level 2 is the weighted sum (Σ) of its input.
- The activation function *f* for the perceptron on level 2 is a *step function*:

$$f = \begin{cases} 1 & if \sum > 0 \\ 0 & otherwise \end{cases}$$

- Assume that the weights for the perceptrons of the first level are given.

3. Consider the following multilayer perceptron.



The network should implement the XOR function. Perform one epoch of *backpropagation* as introduced in the lecture on multilayer perceptrons.

Notes:

- The activation function f for a perceptron is the *sigmoid function*:

$$f(x) = \frac{1}{1 + e^{-x}}$$

- The thresholds are not shown in the network. The threshold nodes are set to -1.
- Use the following initial parameter values:

$\theta_{01}^{(1)} = 2$	$\theta_{02}^{(1)} = -1$	$\theta_{01}^{(2)} = -2$
$\theta_{11}^{(1)} = 6$	$\theta_{12}^{(1)} = 8$	$\theta_{11}^{(2)} = 6$
$\theta_{21}^{(1)} = -6$	$\theta_{22}^{(1)} = -8$	$\theta_{21}^{(2)} = -6$

- The learning rate is set to $\eta = 0.7$

i. Compute the activations of the hidden and output neurons;

ii. Compute the error of the network;

iii. Backpropagate the error to determine $\Delta\theta_{ij}$ for all weights θ_{ij} and updates the weight θ_{ij}.