

COMP9444 Neural Networks and Deep Learning

Term 3, 2020

Exercises 1: Perceptrons

This page was last updated: 09/14/2020 11:35:37

1. Perceptron Learning

- a. Construct by hand a Perceptron which correctly classifies the following data; use your knowledge of plane geometry to choose appropriate values for the weights w_0 , w_1 and w_2 .

Training Example	x_1	x_2	Class
a.	0	1	-1
b.	2	0	-1
c.	1	1	+1

- b. Demonstrate the Perceptron Learning Algorithm on the above data, using a learning rate of 1.0 and initial weight values of

$$w_0 = -1.5$$

$$w_1 = 0$$

$$w_2 = 2$$

In your answer, you should clearly indicate the new weight values at the end of each training step. Here, the first three steps have been done for you. You need to fill in the rest of the table.

Iteration	w_0	w_1	w_2	Training Example	x_1	x_2	Class	$s = w_0 + w_1x_1 + w_2x_2$	Action
1	-1.5	0	2	a.	0	1	-	+0.5	Subtract
2	-2.5	0	1	b.	2	0	-	-2.5	None
3	-2.5	0	1	c.	1	1	+	-1.5	Add

2. XOR Network

Construct by hand a Neural Network (or Multi-Layer Perceptron) that computes the XOR function of two inputs. Make sure the connections, weights and biases of your network are clearly visible.

Challenge: Can you construct a Neural Network to compute XOR which has only one hidden unit, but also includes shortcut connections from the two inputs directly to the (one) output.

Hint: start with a network that computes the inclusive OR, and then try to think of how it could be modified.

3. Computing any Logical Function with a 2-layer Network

Assuming False=0 and True=1, explain how each of the following could be constructed:

- a. Perceptron to compute the OR function of m inputs
- b. Perceptron to compute the AND function of n inputs
- c. 2-layer Neural Network to compute the function $(A \vee B) \wedge (\neg B \vee C \vee \neg D) \wedge (D \vee \neg E)$
- d. 2-Layer Neural Network to compute any (given) logical expression, assuming it is written in **Conjunctive Normal Form**.

Hint: in each case, first decide on the input-to-output or input-to-hidden weights, then determine the bias.
