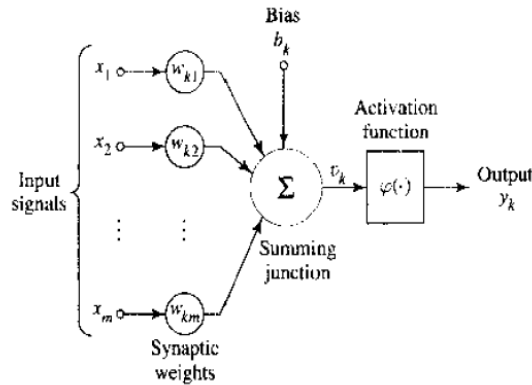


EE5904/ME5404 Neural Networks: Homework #1

Important note: the due date is **13/02/2022**. You should submit your scripts to the folder in LumiNUS. Late submission is not allowed unless it is well justified. Please include the MATLAB code or Python Code as attachment if computer experiment is involved.

Q1. (10 Marks)



Consider the signal-flow graph of the perceptron shown in the above figure. The activation function, $\varphi(v)$, where v is the induced local field, can be designed by the user. If the activation function is chosen as hard limiter (i.e. **step function**), then it becomes the classical perceptron, and the decision boundary is shown to be a hyperplane. In this problem, let's explore **other choices of the activation function**, and its effect on the decision boundary. Let's assume that the classification decision made by the perceptron is simply a threshold rule defined as follows:

Observation vector $x = [x_1 \ x_2 \ \dots \ x_m]^T$ belongs to class C_1 if the output $y > \xi$, where ξ is a user-defined threshold; otherwise, x belongs to class C_2 .

Consider the following three choices of activation function:

- 1) The activation function is the logistic function: $\varphi(v) = \frac{1}{1 + e^{-v}}$;
- 2) The activation function is the Bell-shaped Gaussian function: $\varphi(v) = e^{-\frac{(v-m)^2}{2}}$;
- 3) The activation function is the Softsign function: $\varphi(v) = \frac{v}{1 + |v|}$.

For each case, investigate whether the resulting decision boundary is a hyper-plane or not.

Q2. (10 Marks)

Consider the logic function, EXCLUSIVE OR (XOR).

Truth Table of XOR

x ₁	0	1	0	1
x ₂	0	0	1	1
y	0	1	1	0

It is well known that the XOR problem is not linearly separable. It seems obvious by visually checking, which however cannot be accepted as mathematical proof. Therefore, please supply a rigorous mathematical proof for this statement.

Q3. (10 Marks)

The perceptron could be used to perform numerous logic functions, such as AND, OR, COMPLEMENT and NAND function, whose truth tables are tabulated as follows respectively.

x ₁	0	0	1	1
x ₂	0	1	0	1
y	0	0	0	1

AND

x ₁	0	0	1	1
x ₂	0	1	0	1
y	0	1	1	1

OR

x	0	1
y	1	0

COMPLEMENT

x ₁	0	0	1	1
x ₂	0	1	0	1
y	1	1	1	0

NAND

a). Demonstrate the **implementation** of the logic functions AND, OR, COMPLEMENT and NAND with selection of weights by **off-line calculations**.

(3 Marks)

b). Demonstrate the **implementation** of the logic functions AND, OR, COMPLEMENT and NAND with selection of weights by **learning procedure**. Suppose initial weights are chosen randomly and learning rate η is 1.0. Plot out the trajectories of the weights for each case. Compare the results with those obtained in (a). Try other learning rates, and report your observations with different learning rates.

(4 Marks)

c). What would happen if the perceptron is applied to implement the EXCLUSIVE OR function with selection of weights by learning procedure? Suppose initial weight is chosen randomly and learning rate η is 1.0. Do the computer experiment and explain your finding.

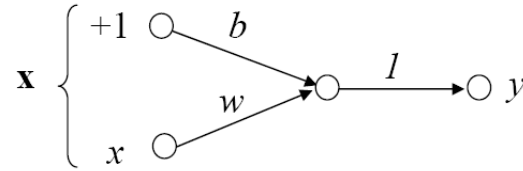
(3 Marks)

Q4. (10 Marks)

Single layer perceptron with pure linear activation function can be used to fit a linear model to a set of input-output pairs. Suppose that we are given the following pairs:

$\{(0,0.5), (0.8, 1), (1.6, 4), (3, 5), (4.0, 6), (5.0, 8)\}$

and a single linear neuron as shown in the following figure.



a). Find the solution of w and b using the standard linear least-squares (LLS) method. Plot out the fitting result.

(3 Marks)

b). Suppose that initial weight is chosen randomly and learning rate η is 0.01. Find the solution of w and b using the least-mean-square (LMS) algorithm for 100 epochs. Plot out the fitting result and the trajectories of the weights versus learning steps. Will the weights converge?

(3 Marks)

c). Compare the results obtained by LLS and the LMS methods.

(2 Marks)

d) Repeat the simulation study in b) with different learning rates η , and explain your findings.

(2 Marks)

Q5. (10 Marks)

Consider that we are trying to fit a linear model to a set of input-output pairs $(x(1), d(1)), (x(2), d(2)) \dots, (x(n), d(n))$ observed in an interval of duration n , where input x is m -dimensional vector, $x = [x_1 \ x_2 \ \dots \ x_m]^T$. The linear model takes the following form:

$$y(x) = w_1 x_1 + w_2 x_2 + \dots + w_m x_m = w^T x.$$

Derive the formula to calculate the optimal parameter w^* such that the following cost function $J(w)$ is minimized.

$$J(w) = \frac{1}{2} \sum_{i=1}^n r^2(i) e(i)^2 + \frac{1}{2} \lambda \|w\|^2 = \frac{1}{2} \sum_{i=1}^n r^2(i) (d(i) - y(x(i)))^2 + \frac{1}{2} \lambda w^T w$$

where $r(i) > 0$ are the weighting factors for each output error $e(i)$, and $\lambda > 0$ is the regularization factor.