Mohamed Boudiaf University of M'sila
Faculty of Mathematics and Computer Science
Department of Computer Science

Subject: AI for M1-AI/M2-ISSI
Prepared by: Dr. Said Gadri

SERIES NUM. 02: STANDARD ANSWER

Exercise 01

Express the following sentences in first-order predicate logic:

1. Ali's mother is married to Ali's father
   ⇔ ***Married(Father(Ali), Mother(Ali))***
2. Ali lives in a yellow house
   ⇔ ***a) Lives (Ali, House)∧Color(House, Yellow)***
   b)∃x House(x)∧Color(x, Yellow)∧Lives(Ali, x)
3. If the car belongs to Ali, then it is green
   ⇔ ***a) has (Ali, Car) ⇒Color(Car, Green)***
   b) ∃x Car(x)∧ Has(Ali, x)⇒Color(x, Green)
4. Some people like snakes
   ⇔ ***a)∃x (Person(x)∧Like(x, Snakx))***
   b)∃x∀y, (Person(x)∧Snake(y))⇒Like(x, y)
5. All students take exams
   ⇔ ***a) ∀x Student(x)⇒Takes_Exam(x)***
   b) ∀x (Student(x)⇒∃y Exam(y)∧Passes(x, y))
6. If x is a parent of y, then x is older than y
   ⇔ ***∀x∀y Person(x) ∧Person(y)∧Parent(x, y)⇒Old(x, y)***
7. If x is the mother of y, then x is a parent of y."
   ⇔ ***∀x∀y Mother(x, y)⇒Parent(x, y)***
8. There is someone to whom everyone is loyal."
   ∃x∀x, Person(x) ∧Person(y)∧Loyal(y, x)
   → ***ambiguous sentence!!***

Exercise 2: (The Chifoumi Game: language)

Let's be a language with the following alphabet:

- Binary predicate symbols ***Win*** and ***Lose***
- Object variables x and y
- The following 0-aera function symbols (terms): stone (pierre), rock, leaf, paper, scissors (ciseau)
- The logical connectors of predicate logic.

Question: Express the following sentences in predicate logic:

1. The leaf wins against the stone.
   win(leaf, stone)
2. The leaf does not win against itself.
   ¬win(leaf, leaf)
3. Each object can neither win nor lose against itself.
   ∀x Object(x) ∧ ¬win(x, x)∧ ¬lose(x, x))
   ¬∃ Object(x) ∧(win(x, x)∨lose(x, x))

4. There are objects against which the leaf wins and there are objects against which the leaf loses.

   $(\exists x\ object(x) \wedge wins(leaf, x)) \wedge \exists x\ lose(leaf, x)$
5. Every object can win against someone.

   $\forall x \exists y\ object(x) \wedge object(y) \wedge wins(x, y)$
6. There is an object that does not win against any object.

   $\exists x \forall y\ object(x) \wedge object(y) \wedge \neg win(x, y)$
7. All objects have an object against which they do not win.

   $\forall x \exists y\ object(x) \wedge object(y) \wedge \neg win(x, y)$
8. If there is an object against which all objects win then every object has an object against which it wins.

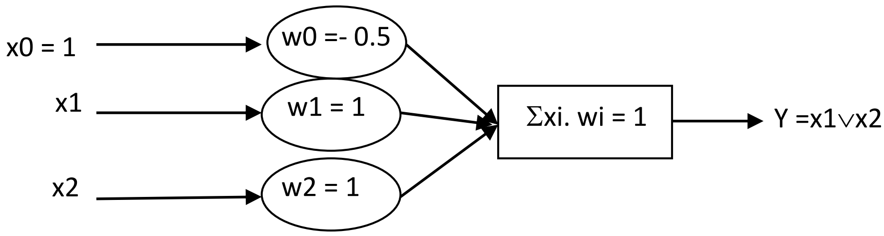
   $\exists y \forall x\ object(x) \wedge object(y) \wedge wins(x, y) \Rightarrow \forall x \exists y\ object(x) \wedge object(y) \wedge wins(x, y)$

Exercise 03

Creation of the logic gate OR using a perceptron

The logic gate OR

Input x1	Input x2	Output
0	0	0
0	1	1
1	0	1
1	1	1

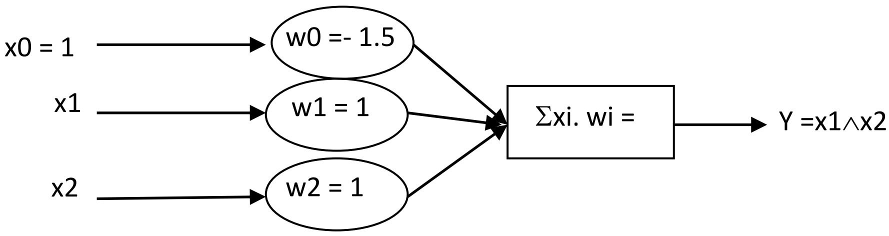


Exercise 4

Creation of the logic gate AND using a perceptron

The logic gate AND

Input x1	Input x2	Output
0	0	0
0	1	0
1	0	0
1	1	1

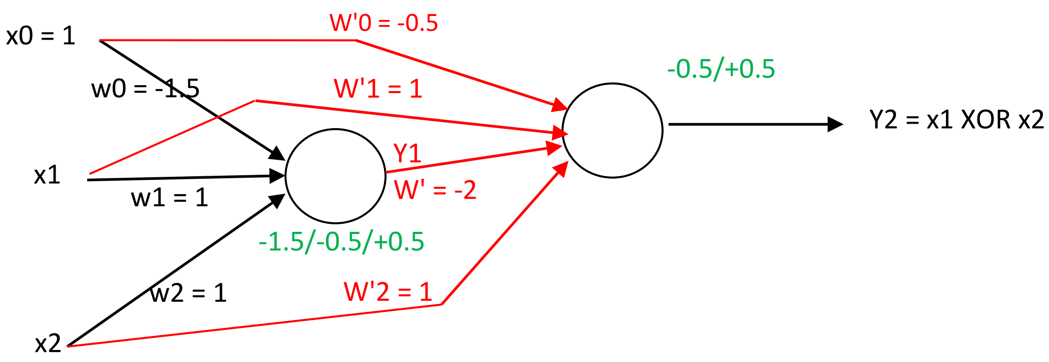


Exercise 5

Realization of the logic gate XOR using a MLP

Input x1	Input x2	Output
0	0	0
0	1	1
1	0	1
1	1	0

X0	X1	X2	∑xi.wi	Y1	∑xi.wi	Y2
1	0	0	$-1.5+0+0=-1.5\leq 0$	0	$-1.5+0+0+0=-0.5\leq 0$	0
1	0	1	$-1.5+0+1=-0.5\leq 0$	0	$-0.5+0+0+1=+0.5>0$	1
1	1	0	$-1.5+1+0=-0.5\leq 0$	0	$-0.5+0+1+0=+0.5>0$	1
1	1	1	$-1.5+1+1=+0.5>0$	1	$-0.5-2+1+1=-0.5\leq 0$	0



Exercise 06

Given a threshold perceptron with the following initial parameters:

- ε = 1 (the threshold of the perceptron); x0 = 1
- $w_0 = 0$; $w_1 = 1$; $w_2 = -1$

Using the perceptron learning algorithm based on Hebb's law, give the ideal values of the weights w_0, w_1, w_2 which allow one of the logic gates below to be realized for all inputs x_1, x_2

a) logical OR

<u>Answer</u>

1. We know that for a perceptron with threshold ε
   O: the output calculated by the perceptron
   O = 1 if Σwi .xi > ε
   O = 0 Otherwise
2. C: the desired output
3. If O ≠ C we modify the weight wi as follows:

wi = wi + ε * (C – O)* xi (The perceptron learning equation)

By applying the previous principles, we obtain the results illustrated in the following table:

Stage	W0	W1	W2	Inputs	Σwi.xi	O	C	W0	W1	W2
0	0	1	-1	100	0+0+0≤(ε = 1)	0	0	0	1	-1
1	0	1	-1	101	0+0+ (-1)≤ ε	0	1	0+1*1*1=1	1+1*1*0=1	-1+1*1*1=0
2	1	1	0	110	1+1+0=2>ε	1	1	1	1	0
3	1	1	0	111	1+1+0=2>ε	1	1	1	1	0
4	1	1	0	100	1+0+0=1≤ ε	0	0	1	1	0
5	1	1	0	101	1+0+0=1≤ ε	0	1	2	1	1
6	2	1	1	110	2+1+0=3>ε	1	1	2	1	1
7	2	1	1	111	2+1+1=4>ε	1	1	2	1	1
8	2	1	1	100	2+0+0=2>ε	1	0	1	1	1
9	1	1	1	101	1+0+1=2>ε	1	1	1	1	1
10	1	1	1	110	1+1+0=2>ε	1	1	1	1	1
11	1	1	1	111	1+1+1=3>ε	1	1	1	1	1
12	1	1	1	100	1+0+0=1≤ ε	0	0	1	1	1
13	1	1	1	101	1+0+1=2>ε	1	1	1	1	1
14	1	1	1	110	1+1+0=2>ε	1	1	1	1	1
15	1	1	1	111	1+1+1=3>ε	1	1	1	1	1

So the ideal weight values to make an OR gate are as follows:

w0 = 1; w1 = 1; w2 = 1

b) AND logical

We do the same thing.