1.1 Expressiveness of Neural Networks [10 points]

As discussed in class, neural networks are built out of units with real-valued inputs $X_1 \dots X_n$, where the unit output Y is given by

$$Y = \frac{1}{1 + \exp(-(w_0 + \sum_i w_i X_i))}$$

Here we will explore the expressiveness of neural nets, by examining their ability to represent boolean functions. Here the inputs X_i will be 0 or 1. Of course the output Y will be real-valued, ranging anywhere between 0 and 1. We will interpret Y as a boolean value by interpreting it to be a boolean 1 if $Y > 0.5$, and interpreting it to be 0 otherwise.

1. Give 3 weights for a single unit with two inputs X_1 and X_2, that implements the logical OR function $Y = X_1 \vee X_2$.

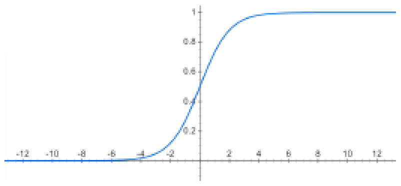


Figure 1: $\frac{1}{1+e^{-x}}$.

★ **SOLUTION:** Figure 1 shows the value of $y = \frac{1}{1+e^{-x}}$ for different values of x. Note that $y \geq 0.5$ if $x \geq 0$, and $y \leq 0.5$ if $x \leq 0$. Given this, we need to choose w_i so that $w_0 + w_1 * x_1 + w_2 * x_2$ will be greater than 0 when $x_1 \vee x_2$ is equal to 1. One candidate solution is $[w_0 = -0.5, w_1 = 1, w_2 = 1]$.

2. Can you implement the logical AND function $Y = X_1 \wedge X_2$ in a single unit? If so, give weights that achieve this. If not, explain the problem.

   ★ **SOLUTION:** Similar to previous part, we can obtain $[w_0 = -1.5, w_1 = 1, w_2 = 1]$

3. It is impossible to implement the EXCLUSIVE-OR function $Y = X_1 \oplus X_2$ in a single unit. However, you can do it using a multiple unit neural network. Please do. Use the smallest number of units you can. Draw your network, and show all weights of each unit.

   ★ **SOLUTION:** It can be represented by a neural network with two nodes in the hidden layer. Input weights for node 1 in the hidden layer would be $[w_0 = -0.5, w_1 = 1, w_2 = -1]$, input weights for node 2 in the hidden layer would be $[w_0 = -0.5, w_1 = -1, w_2 = 1]$, and input weights for the output node would be $[w_0 = -0.8, w_1 = 1, w_2 = 1]$.

4. Create a neural network with only one hidden layer (of any number of units) that implements $(A \vee \neg B) \oplus (\neg C \vee \neg D)$. Draw your network, and show all weights of each unit.

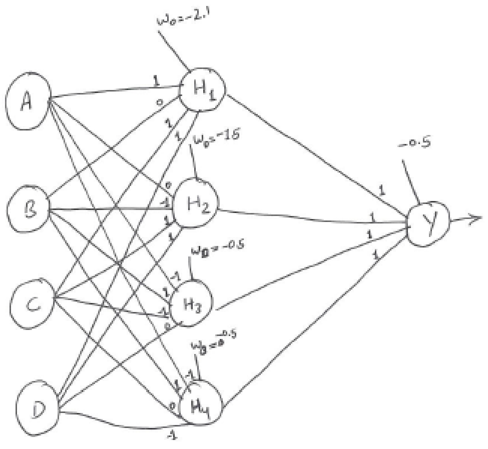


Figure 2: An example of neural network for problem 1.4

★ **SOLUTION:** Note that XOR operation can be written in terms of AND and OR operations: $p \oplus q = (p \wedge \neg q) \vee (\neg p \wedge q)$. Given this, we can rewrite the formula as $(A \wedge C \wedge D) \vee (\neg B \wedge C \wedge D) \vee (\neg A \wedge B \wedge \neg C) \vee (\neg A \wedge B \wedge \neg D)$. This formula can be represented by a neural network with one hidden layer and four nodes in the hidden layer (one unit for each parenthesis). An example is shown in Figure 2.

1. You are given the following neural networks which take two binary valued inputs $x_1, x_2 \in \{0, 1\}$ and the activation function is the threshold function($h(x) = 1$ if $x > 0$; 0 otherwise). Which of the following logical functions does it compute?

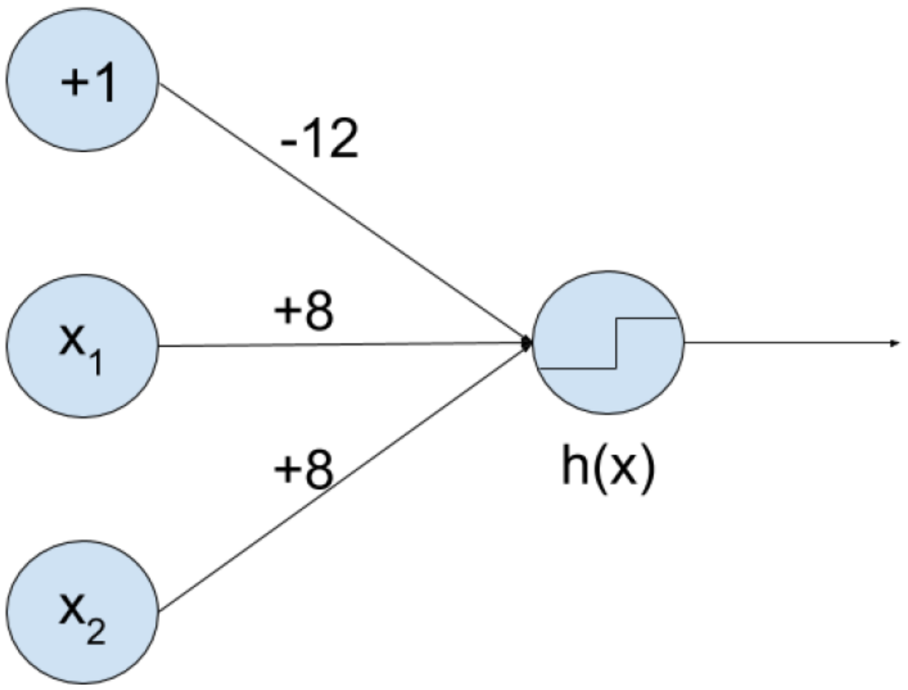


Figure 1: Q1

(a) OR
(b) AND
(c) NAND
(d) None of the above.

Solution: B
You can construct the truth table and see the values and decide which gate the network mimics.

0	0	0
0	1	0
1	0	0
1	1	1

2. We have a function which takes a two-dimensional input $x = (x_1, x_2)$ and has two parameters $w = (w_1, w_2)$ given by $f(x, w) = \sigma(\sigma(x_1 w_1)w_2 + x_2)$ where $\sigma(x) = \frac{1}{1+e^{-x}}$. We use backpropagation to estimate the right parameter values. We start by setting both the parameters to

0. Assume that we are given a training point $x_1 = 1, x_2 = 0, y = 5$. Given this information answer the next two questions. What is the value of $\frac{\partial f}{\partial w_2}$?

(a) 0.5
(b) -0.25
(c) 0.125
(d) -0.5

Solution: C
Write $\sigma(x_1 w_1) w_2 + x_2$ as o_2 and $x_1 w_1$ as o_1

$$\frac{\partial f}{\partial w_2} = \frac{\partial f}{\partial o_2} \frac{\partial o_2}{\partial w_2}$$

$$\frac{\partial f}{\partial w_2} = \sigma(o_2)(1 - \sigma(o_2)) \times \sigma(o_1)$$

$$\frac{\partial f}{\partial w_2} = 0.5 * 0.5 * 0.5$$

3. If the learning rate is 0.5, what will be the value of w_2 after one update using backpropagation algorithm?

(a) 0.0625
(b) -0.0625
(c) 0.5625
(d) - 0.5625

Solution: C
The update equation would be

$$w_2 = w_2 - \lambda \frac{\partial L}{\partial w_2}$$

where L is the loss function, here $L = (y - f)^2$

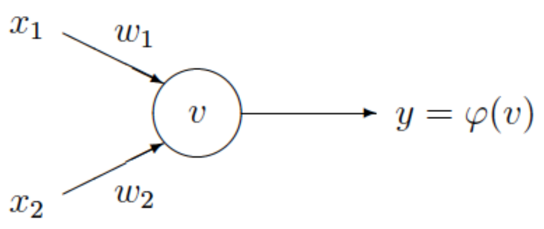
$$w_2 = w_2 - \lambda \times 2(y - f) \times (-1) \times \frac{\partial f}{\partial w_2}$$

Now putting in the given values we get the right answer.

Logical operators (i.e. NOT, AND, OR, XOR, etc) are the building blocks of any computational device. Logical functions return only two possible values, true or false, based on the truth or false values of their arguments. For example, operator AND returns true only when all its arguments are true, otherwise (if any of the arguments is false) it returns false. If we denote truth by 1 and false by 0, then logical function AND can be represented by the following table:

x_1 :	0	1	0	1
x_2 :	0	0	1	1
x_1 AND x_2 :	0	0	0	1

This function can be implemented by a single–unit with two inputs:



if the weights are $w_1 = 1$ and $w_2 = 1$ and the activation function is:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 2 \\ 0 & \text{otherwise} \end{cases}$$

Note that the threshold level is 2 ($v \geq 2$).

a) Test how the neural AND function works.

Answer:

$$\begin{aligned}
P_1 &: \quad v = 1 \cdot 0 + 1 \cdot 0 = 0\,, \quad (0 < 2)\,, \quad y = \varphi(0) = 0 \\
P_2 &: \quad v = 1 \cdot 1 + 1 \cdot 0 = 1\,, \quad (1 < 2)\,, \quad y = \varphi(1) = 0 \\
P_3 &: \quad v = 1 \cdot 0 + 1 \cdot 1 = 1\,, \quad (1 < 2)\,, \quad y = \varphi(1) = 0 \\
P_4 &: \quad v = 1 \cdot 1 + 1 \cdot 1 = 2\,, \quad (2 = 2)\,, \quad y = \varphi(2) = 1
\end{aligned}$$

b) Suggest how to change either the weights or the threshold level of this single–unit in order to implement the logical OR function (true when at least one of the arguments is true):

x_1 :	0	1	0	1
x_2 :	0	0	1	1
x_1 OR x_2 :	0	1	1	1

Answer: *One solution is to increase the weights of the unit:* $w_1 = 2$ *and* $w_2 = 2$*:*

$$\begin{aligned}
P_1 &: \quad v = 2 \cdot 0 + 2 \cdot 0 = 0\,, \quad (0 < 2)\,, \quad y = \varphi(0) = 0 \\
P_2 &: \quad v = 2 \cdot 1 + 2 \cdot 0 = 2\,, \quad (2 = 2)\,, \quad y = \varphi(2) = 1 \\
P_3 &: \quad v = 2 \cdot 0 + 2 \cdot 1 = 2\,, \quad (2 = 2)\,, \quad y = \varphi(2) = 1 \\
P_4 &: \quad v = 2 \cdot 1 + 2 \cdot 1 = 4\,, \quad (4 > 2)\,, \quad y = \varphi(4) = 1
\end{aligned}$$

Alternatively, we could reduce the threshold to 1:

$$\varphi(v) = \begin{cases} 1 & \textit{if } v \geq 1 \\ 0 & \textit{otherwise} \end{cases}$$

c) The XOR function (exclusive or) returns true only when one of the arguments is true and another is false. Otherwise, it returns always false. This can be represented by the following table:

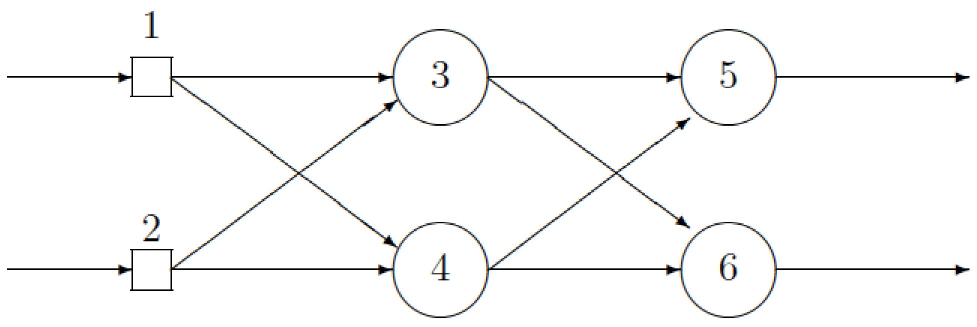
x_1 :	0	1	0	1
x_2 :	0	0	1	1
x_1 XOR x_2 :	0	1	1	0

Do you think it is possible to implement this function using a single unit? A network of several units?

Answer: *This is a difficult question, and it puzzled scientists for some time because it is actually impossible to implement the XOR function neither by a single unit nor by a single–layer feed–forward network (single–layer perceptron). This was known as the XOR problem. The solution was found using a feed–forward network with a hidden layer. The XOR network uses two hidden nodes and one output node.*

Question 4

The following diagram represents a feed–forward neural network with one hidden layer:



A weight on connection between nodes i and j is denoted by w_{ij}, such as w_{13} is the weight on the connection between nodes 1 and 3. The following table lists all the weights in the network:

$w_{13} = -2$	$w_{35} = 1$
$w_{23} = 3$	$w_{45} = -1$
$w_{14} = 4$	$w_{36} = -1$
$w_{24} = -1$	$w_{46} = 1$

Each of the nodes 3, 4, 5 and 6 uses the following activation function:

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

where v denotes the weighted sum of a node. Each of the input nodes (1 and 2) can only receive binary values (either 0 or 1). Calculate the output of the network (y_5 and y_6) for each of the input patterns:

Pattern:	P_1	P_2	P_3	P_4
Node 1:	0	1	0	1
Node 2:	0	0	1	1

Answer: *In order to find the output of the network it is necessary to calculate weighted sums of hidden nodes 3 and 4:*

$$v_3 = w_{13}x_1 + w_{23}x_2 \ , \quad v_4 = w_{14}x_1 + w_{24}x_2$$

Then find the outputs from hidden nodes using activation function φ*:*

$$y_3 = \varphi(v_3) \ , \quad y_4 = \varphi(v_4) \ .$$

Use the outputs of the hidden nodes y_3 *and* y_4 *as the input values to the output layer (nodes 5 and 6), and find weighted sums of output nodes 5 and 6:*

$$v_5 = w_{35}y_3 + w_{45}y_4 \ , \quad v_6 = w_{36}y_3 + w_{46}y_4 \ .$$

Finally, find the outputs from nodes 5 and 6 (also using φ*):*

$$y_5 = \varphi(v_5) \ , \quad y_6 = \varphi(v_6) \ .$$

The output pattern will be (y_5, y_6)*. Perform these calculation for each input pattern:*

P_1*: Input pattern* $(0,0)$

$$\begin{aligned} v_3 &= -2\cdot 0 + 3\cdot 0 = 0, & y_3 &= \varphi(0) = 1 \\ v_4 &= 4\cdot 0 - 1\cdot 0 = 0, & y_4 &= \varphi(0) = 1 \\ v_5 &= 1\cdot 1 - 1\cdot 1 = 0, & y_5 &= \varphi(0) = 1 \\ v_6 &= -1\cdot 1 + 1\cdot 1 = 0, & y_6 &= \varphi(0) = 1 \end{aligned}$$

The output of the network is $(1,1)$*.*

P_2*: Input pattern* $(1,0)$

$$\begin{aligned} v_3 &= -2\cdot 1 + 3\cdot 0 = -2, & y_3 &= \varphi(-2) = 0 \\ v_4 &= 4\cdot 1 - 1\cdot 0 = 4, & y_4 &= \varphi(4) = 1 \\ v_5 &= 1\cdot 0 - 1\cdot 1 = -1, & y_5 &= \varphi(-1) = 0 \\ v_6 &= -1\cdot 0 + 1\cdot 1 = 1, & y_6 &= \varphi(1) = 1 \end{aligned}$$

The output of the network is $(0,1)$*.*

P_3*: Input pattern* $(0,1)$

$$\begin{aligned} v_3 &= -2\cdot 0 + 3\cdot 1 = 3, & y_3 &= \varphi(3) = 1 \\ v_4 &= 4\cdot 0 - 1\cdot 1 = -1, & y_4 &= \varphi(-1) = 0 \\ v_5 &= 1\cdot 1 - 1\cdot 0 = 1, & y_5 &= \varphi(1) = 1 \\ v_6 &= -1\cdot 1 + 1\cdot 0 = -1, & y_6 &= \varphi(-1) = 0 \end{aligned}$$

The output of the network is $(1,0)$*.*

P_4: Input pattern $(1, 1)$

$$\begin{aligned} v_3 &= -2 \cdot 1 + 3 \cdot 1 = 1, & y_3 &= \varphi(1) = 1 \\ v_4 &= 4 \cdot 1 - 1 \cdot 1 = 3, & y_4 &= \varphi(3) = 1 \\ v_5 &= 1 \cdot 1 - 1 \cdot 1 = 0, & y_5 &= \varphi(0) = 1 \\ v_6 &= -1 \cdot 1 + 1 \cdot 1 = 0, & y_6 &= \varphi(0) = 1 \end{aligned}$$

The output of the network is $(1, 1)$.