Questions 11:
Feed-Forward Neural Networks

Roman Belavkin

Middlesex University

Question 1

Below is a diagram if a single artificial neuron (unit):

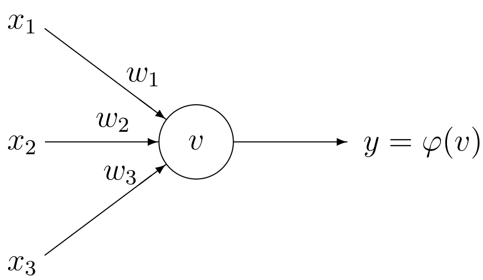


Figure 1: Single unit with three inputs.

The node has three inputs $\mathbf{x} = (x_1, x_2, x_3)$ that receive only binary signals (either 0 or 1). How many different input patterns this node can receive? What if the node had four inputs? Five? Can you give a formula that computes the number of binary input patterns for a given number of inputs?

Answer: *For three inputs the number of combinations of 0 and 1 is 8:*

x_1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1

and for four inputs the number of combinations is 16:

x_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
x_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
x_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
x_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

You may check that for five inputs the number of combinations will be 32. Note that $8 = 2^3$, $16 = 2^4$ *and* $32 = 2^5$ *(for three, four and five inputs). Thus, the formula for the number of binary input patterns is:*

$$2^n \ , \quad \text{where } n \text{ in the number of inputs}$$

Question 2

Consider the unit shown on Figure 1. Suppose that the weights corresponding to the three inputs have the following values:

$$\begin{array}{|lcr|} w_1 & = & 2 \\ w_2 & = & -4 \\ w_3 & = & 1 \end{array}$$

and the activation of the unit is given by the step-function:

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

Calculate what will be the output value y of the unit for each of the following input patterns:

Pattern	P_1	P_2	P_3	P_4
x_1	1	0	1	1
x_2	0	1	0	1
x_3	0	1	1	1

Answer: *To find the output value* y *for each pattern we have to:*

a) *Calculate the weighted sum:* $v = \sum_i w_i \, x_i = w_1 \cdot x_1 + w_2 \cdot x_2 + w_3 \cdot x_3$

b) *Apply the activation function to* v

The calculations for each input pattern are:

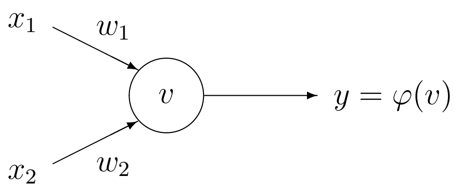
$$\begin{array}{lrrr} P_1 : & v = 2 \cdot 1 - 4 \cdot 0 + 1 \cdot 0 = 2 \ , & (2 > 0) \ , & y = \varphi(2) = 1 \\ P_2 : & v = 2 \cdot 0 - 4 \cdot 1 + 1 \cdot 1 = -3 \ , & (-3 < 0) \ , & y = \varphi(-3) = 0 \\ P_3 : & v = 2 \cdot 1 - 4 \cdot 0 + 1 \cdot 1 = 3 \ , & (3 > 0) \ , & y = \varphi(3) = 1 \\ P_4 : & v = 2 \cdot 1 - 4 \cdot 1 + 1 \cdot 1 = -1 \ , & (-1 < 0) \ , & y = \varphi(-1) = 0 \end{array}$$

Question 3

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* $w_2 = 2$*:*

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

Alternatively, we could reduce the threshold to 1:

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

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:

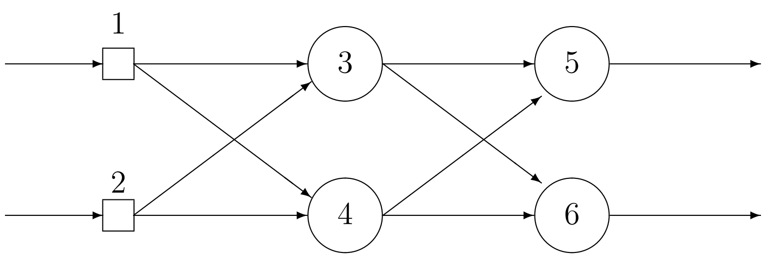
x_1 :	0	1	0	1
x_2 :	0	0	1	1
x_1 XOR x_2 :	0	1	1	0

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 $(1, 1)$.

Question 5

What is a training set and how is it used to train neural networks?

Answer: *Training set is a set of pairs of input patterns with corresponding desired output patterns. Each pair represents how the network is supposed to respond to a particular input. The network is trained to respond correctly to each input pattern from the training set. Training algorithms that use training sets are called supervised learning algorithms. We may think of a supervised learning as learning with a teacher, and the training set as a set of examples. During training the network, when presented with input patterns, gives 'wrong' answers (not desired output). The error is used to adjust the weights in the network so that next time the error was smaller. This procedure is repeated using many examples (pairs of inputs and desired outputs) from the training set until the error becomes sufficiently small.*

Question 6

What is an epoch?

Answer: *An epoch is when all of the data in the training set is presented to the neural network once.*

Question 7

Describe the main steps of the supervised training algorithm?

Answer:

- *Initially, set all the weights to some random values*
- *Repeat (for many epochs):*
  - **a)** *Feed the network with an input from one of the examples in the training set*
  - **b)** *Compute the error between the output of the network and the desired output*
  - **c)** *Correct the error by adjusting the weights of the nodes*
- *Until the error is sufficiently small*

Question 8

Suppose that a credit card company decided to deploy a new system for assessing credit worthiness of its customers. The new system is using a feed-forward neural network with a supervised learning algorithm. Suggest in a form of essay what should the bank have before the system can be used? Discuss problems associated with this requirement.

Answer: *The answer should mention that the company should get hold of* **historical data** *about its customers who already took credit in the past. This data will be used as a* **training set** *for the neural network. It is important that the data is* **representative** *and covers as many types of customers as possible. This is because the network will not be able to produce an accurate answer for a customer very different from those in the training set.*