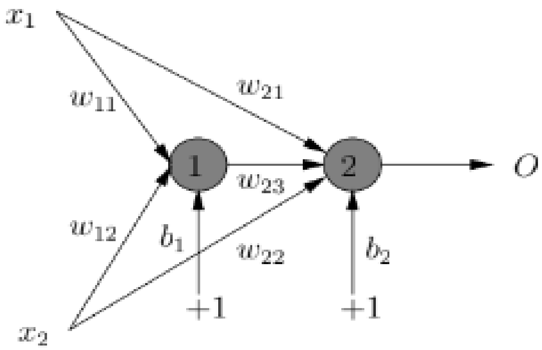
Aprendizagem Automática Avançada - Assignment 1
by Martim Silva 51304 and Alexandre Sobreira 59451

Exercise 1 a)

For this Neural Network:



Having the weights as:

$$w_{11} = w_{12} = w_{21} = w_{22} = +1; w_{23} = -2; b_1 = -1.5; b_2 = -0.5$$

Also considering the activation function to be a step function (and calling the "O" standing for "output" in the image above as "y") we have:

$$y = \varphi(x) = \begin{cases} 0 \; if \; x \leq 0 \\ 1 \; if \; x \geq 0 \end{cases} \tag{1}$$

$$y = \varphi(\sum_{i=0}^{n} w_i \, . \, x_i) \tag{2}$$

And we are working with 2 elements in the input layer in total: $x1, \; x2.$
So we apply the above formula (2) to our case without forgetting the biases for each neuron:

$$y = \varphi(w21.x1 + w22.x2 + b2 - w23.\varphi(w11.x1 + w12.x2 + b1))$$

Since all the weights except $w23 =- 2$ are equal to +1 and $b1 =- 1.5; b2 =- 0.5$ we have

$$y = \varphi(x1 + x2 - 0.5 - 2.\varphi(x1 + x2 + b1))$$

To calculate the part pertaining to neuron 2, we need to take into account the fact that it can also take as input the output of neuron 1 which is why we have the activation function within itself in our calculations.

So if we fix to a constant possible value one of our inputs, for example x1 to the value 0 we have:

$$y = \varphi(x2 - 0.5 - 2.\varphi(x2 - 1.5))$$

And if we go about replacing the values of the remaining input with some values ranging from 0 to 2 we can see that:

-For $x2 = 0$:
$$y = \varphi(0 - 0.5 - 2.\varphi(0 - 1.5)) = 0$$

-For $x2 = 0.5$:
$$y = \varphi(0.5 - 0.5 - 2.\varphi(0.5 - 1.5)) = 0$$

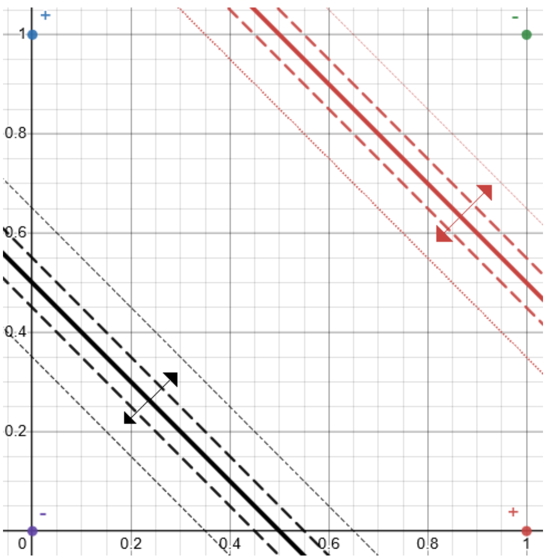
-For $x2 = 1$:
$$y = \varphi(1 - 0.5 - 2.\varphi(1 - 1.5)) = 1$$

-For $x2 = 1.5$:
$$y = \varphi(1.5 - 0.5 - 2.\varphi(1.5 - 1.5)) = 1$$

-For $x2 = 2$:
$$y = \varphi(2 - 0.5 - 2.\varphi(2 - 1.5)) = 0$$

Note: We use the step function mentioned earlier (1) to derive a value from $\varphi(...)$ blocks, if the value inside parenthesis is lesser or equal to 0 it results in 0 and if not it results in 1.

So using these calculated points while having $x1 = 0$ our separation lines in a graph will look like this:



Above the black filled-in line and below the red filled-in line are the output values of 0 while below the red filled-in line and above the black filled-in line are the values of 1. The crossing sections are these lines themselves with the black one taking place at 0.5 and the red one at 1.5.

Exercise 1 b)

The inputs and the output can only be 0s or 1s so calculating our "y" for all combinations of this we have:

-For $x1 = 0; x2 = 0$:
$y = \varphi(0 + 0 - 0.5 - 2.\varphi(0 + 0 - 1.5)) = 0$

-For $x1 = 0; x2 = 1$:
$y = \varphi(0 + 1 - 0.5 - 2.\varphi(0 + 1 - 1.5)) = 1$

-For $x1 = 1; x2 = 0$:
$y = \varphi(1 + 0 - 0.5 - 2.\varphi(1 + 0 - 1.5)) = 1$

-For $x1 = 1; x2 = 1$:
$y = \varphi(1 + 1 - 0.5 - 2.\varphi(1 + 1 - 1.5)) = 0$

The truth table for this NN is that of the "exclusive or" or XOR:

$x1$	$x2$	y
0	0	0
0	1	1
1	0	1
1	1	0

Exercise 1 c)

Considering the activation function to be a sigmoid with $a = 1$:

$$y = \varphi(x) = 1/(1 + e^{(-avj)}) = 1/(1 + e^{-vj})$$

A single iteration of the error backpropagation algorithm where the goal is to find the values for each of the weights and biases in the next iteration starts in the forward step with calculating the induced local fields. Note: In this answer it is only shown the case for the input $x1 = 0, x2 = 0$ with the logic for the remaining combination of possible inputs being analogous to that shown here.

$$w_{11} = w_{12} = w_{21} = w_{22} = +1; w_{23} = -2; b_1 = -1.5; b_2 = -0.5$$

$$x1 = 0;\ x2 = 0;\ j = 1, 2; n = 1$$

$$vj(n) = \sum_{i=0}^{m}(wji(n).\,yi(n))$$

$$v1(n) = \sum_{i=0}^{m}(w1i(n).\,yi(n)) = w11.\,x1 + w12.\,x2 + b1 =- 1.5$$

$$v2(n) = \sum_{i=0}^{m}(w2i(n).\,yi(n)) = w21.\,x1 + w22.\,x2 + b2 + w23.\,(w11.\,x1 + w12.\,x2 + b1)$$
$$= 2.5$$

$$yj(n) = \varphi j(vj(n))$$

$$y1(n) = 1/(1 + e^{-(-1.5)}) = 0.182$$
$$y2(n) = 1/(1 + e^{-(2.5)}) = 0.924$$

We have just one neuron in the output layer, the one catalogued with a 2 is our output neuron, also to note since $x1 = 0, x2 = 0$ our desired output will be $dj(n) = 0$ like it is shown in the NN's truth table. Now for the backward step.

$$\partial\varepsilon(n)/\partial wji(n) =- ej(n).\,\varphi'j(vj(n)).\,yi(n)$$
$$ej(n) = dj(n) - yj(n), \qquad yi(n) = \text{input}, \qquad dj(n) = 0$$
$$\varphi'j(vj(n)) = a.\,yj(n)[1 - yj(n)], \qquad a = 1$$

First we calculate partial derivatives for the output layer regarding weights:

$$w23 -> \partial\varepsilon(n)/\partial w23(n) =- (0 - 0.924)\times(1\times0.924\times[1 - 0.924])\times0.182 = 0.12$$
$$w21 -> \partial\varepsilon(n)/\partial w21(n) =- (0 - 0.924)\times(1\times0.924\times[1 - 0.924])\times0 = 0$$
$$w22 -> \partial\varepsilon(n)/\partial w22(n) = \partial\varepsilon(n)/\partial w21(n) = 0$$

And regarding the bias:

$$\partial\varepsilon(n)/\partial bj(n) =- ej(n).\,\varphi'j(vi(n)).\,1$$

$b2 =- (0 - 0.924)\times0.07\times1 = 0.065$

Now for the partial derivatives for the hidden layer regarding weights:

$\delta j(n) = \varphi'j(vj(n).\sum(\delta k(n).wkj(n)),\ k = 1\ (since\ we\ have\ only\ 1\ output\ neuron)$
$\delta k(n) = e(n).\varphi'k(vk(n))$

$\delta 1(n) = \varphi'1(v1(n))\times\delta 2(n)\times w23(n)$
$= (1\times0.182\times(1 - 0.182))\times(((0 - 0.924)\times0.07)\times(- 2))$
$= 0.149\times(- 0.065\times(- 2)) = 0.0194$

$w11 -> \delta 1(n).x1 = 0.0194\times0 = 0$
$w12 -> \delta 1(n).x2 = 0$

And regarding the bias:
$b1 =- ej(n).\varphi'j(vi(n)).1 =- (0 - 0.182)\times0.148876\times1 = 0.027$

By considering our learning rate to be $\eta = 0.1$ we obtain the weights and biases for the next iteration with gradient descent:

$wji(n + 1) = wji(n) + \Delta wji(n),\quad \Delta wji(n) =- \eta.(\partial\varepsilon(n)/\partial wji(n)) = \eta.\delta j(n).yi(n)$
$bji(n + 1) = bji(n) + \Delta bji(n),\quad \Delta bji(n) =- \eta.(\partial\varepsilon(n)/\partial bji(n))$

$w23(n + 1) =- 2 + (- 0.1\times0.12) =- 2.001$
$w21(n + 1) = 1 + (- 0.1\times0) = 1$
$w22(n + 1) = w21(n + 1) = 1$
$w11(n + 1) = 1 + (0.1\times0.0194\times0) = 1$
$w12(n + 1) = w11(n + 1) = 1$
$b2(n + 1) =- 2 + (- 0.1\times0.065) =- 2.0065$
$b1(n + 1) =- 1.5 + (- 0.1\times0.027) =- 1.5027$