Advanced Machine Learning 1st Assignment

André Teixeira Dias (59452), Tiago Rodrigues (49593)

Problem 1 a)

From the problem description it is important to note that it is a XOR problem, whose output is either 1 or 0, with a step function as an activation function, described as follows:

$$\varphi_j(v_j(n)) = \begin{cases} 0, & if\ v_j(n) < 0 \\ 1, & if\ v_j(n) > 0 \end{cases}$$ (does not have derivative at 0, but usually assumed as 0)

To determine the separation planes, it is first necessary to build the output expression of the output neuron (y_2). The output is a linear combination of the inputs and, as such, can be defined as:

$$y_j(n) = \varphi_j(v_j(n)) = \varphi_j\left(\sum_{i=0}^{ml} w_{ji}x_i\right)$$

where m is the total number of neurons in a given layer (l), w_{ji} the synapse weights and x_i the inputs.

This equation can then be deconstructed into:

$$y_2 = \varphi_2\left(w_{2\,1} * x_1 + w_{2\,2} * x_2 + b_2 + w_{2\,3} * \overset{\text{Output of neuron 1 } (y_1)}{\boxed{\varphi_1(w_{1\,1} * x_1 + w_{1\,2} * x_2 + b_1)}}\right)$$

By substituting the values of the weights and biases (which are given), we arrive at:

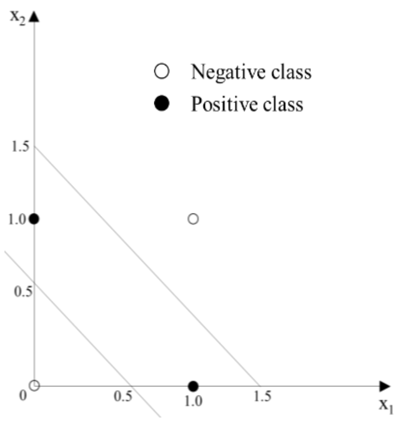
$$y_2 = \varphi_2\big(x_1 + x_2 - 0.5 - 2 * \varphi_1(x_1 + x_2 - 1.5)\big)$$

To determine the separating planes, it is necessary to fix either x_1 or x_2 to 0 and compute the resulting expression:

Fixing x_1 ($x_1 = 0$):
$$y_2 = \varphi_2\big(x_2 - 0.5 - 2 * \varphi_1(x_2 - 1.5)\big)$$

Fixing x_2 ($x_2 = 0$):
$$y_2 = \varphi_2\big(x_1 - 0.5 - 2 * \varphi_1(x_1 - 1.5)\big)$$

From this expression, it is possible to determine the x_1 and x_2 values of the separating planes. The second part of the expression ($\varphi_1(x_{1\text{ or }2} - 1.5)$) changes from 0 to 1 when x_1 or $x_2 = 1.5$ (given the utilized step activation function), while the first part of the expression $\varphi_2(x_{1\text{ or }2} - 0.5)$ changes from 0 to 1 when x_1 or $x_2 = 0.5$. A plot example is presented below:



To define the presented hyperplanes, it is necessary to determine the slope of each. For this, the following expression was used:

$$\text{m (slope)} = \frac{y_2 - y_1}{x_2 - x_1}$$

where x_1 and y_1 correspond to the coordinates of one point and x_2 and y_2 to the coordinates of another point. In this case, the y plane is the x_2 plane.

Regarding the hyperplane closer to 0 (h_1), the known coordinates of the two points are (0, 0.5) and (0.5, 0), while for the other hyperplane (h_2), the known points are (0, 1.5) and (1.5, 0). From these points, the following slopes can be calculated:

$$m_1 = \frac{0 - 0.5}{0.5 - 0} = -1 \qquad m_2 = \frac{0 - 1.5}{1.5 - 0} = -1$$

Since the first hyperplane (h_1) crosses the x_2 plane in 0.5 and the second hyperplane (h_2) also crosses it at 1.5, their expressions are:

$$h_1 = -x + 0.5 \qquad h_2 = -x + 1.5$$

Considering the step activation function, values of x_1 and x_2 that are between these two planes will result in an output of 1 (positive class), while values below h_1 or above h_2 will result in an output of 0 (negative class).

Problem 1 b)

Considering the presented XOR problem, x_1 and x_2 can only take values of 0 or 1. The truth table is a table that presents all possible outputs, taking into account all possible combinations of values of x_1 and x_2. An example table can be seen below:

x_1	x_2	y_2 (output)
0	0	$?_1$
0	1	$?_2$
1	0	$?_3$
1	1	$?_4$

The output values can be calculated using the output expression previously defined:

$$?_1 = \varphi_2\big(0 + 0 - 0.5 - 2 * \varphi_1(0 + 0 - 1.5)\big) = \varphi_2(-0.5 - 2 * 0) = 0$$

$$?_2 = \varphi_2\big(0 + 1 - 0.5 - 2 * \varphi_1(0 + 1 - 1.5)\big) = \varphi_2(0.5 - 2 * 0) = 1$$

$$?_3 = \varphi_2\big(1 + 0 - 0.5 - 2 * \varphi_1(1 + 0 - 1.5)\big) = \varphi_2(0.5 - 2 * 0) = 1$$

$$?_4 = \varphi_2\big(1 + 1 - 0.5 - 2 * \varphi_1(1 + 1 - 1.5)\big) = \varphi_2(1.5 - 2 * 1) = 0$$

From the presented results, it can be seen that when x_1 and x_2 take the same value [(0, 0) or (1, 1)], the output is 0, while different values of x_1 and x_2 [(0, 1) or (1, 0)] lead to an output of 1.

Problem 1 c)

It is now asked to calculate a backpropagation iteration using a sigmoid activation function with $a = 1$. To do this, it is important to state the sigmoid activation function:

$$\varphi_j(v_j(n)) = \frac{1}{1 + e^{-av_j(n)}}\text{, with } a = 1$$

Forward propagation

Considering the case $x_1 = 0$ and $x_2 = 0$:

$$y_1(1) = \varphi_1(0 + 0 - 1.5) = \frac{1}{1 + e^{1.5}} \sim 0.18243$$

$$y_2(1) = \varphi_2\big(0 + 0 - 0.5 - 2 * \varphi_1(0 + 0 - 1.5)\big) = \varphi_2\left(-0.5 - 2 * \frac{1}{1 + e^{1.5}}\right) =$$
$$= \varphi_2(-0.5 - 2 * 0.18243) = \frac{1}{1 + e^{0.86485}} \sim 0.29633$$

Backpropagation

Since the resulting output of the neural network is not equal to the desired output for this specific case ($y_2 = 0$), backpropagation should be used to adjust the network's weights to correct it. The weight correction is given by:

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)$$

The backpropagation algorithm should start from the output layer first and then move backwards on the neuron layers. For an output neuron, $y_k(n) = o_k(n)$, and the local gradient (δ_k) is given by:

$$\delta_k(n) = a o_k(n)[1 - o_k(n)][d_k(n) - o_k(n)]$$

$\delta_2(1) = a y_2(1)[1 - y_2(1)][d_2(1) - y_2(1)]$

$\delta_2(1) = 1 * 0.29633 * [1 - 0.29633] * [0 - 0.29633] \sim -0.06179$

From this, it is now possible to calculate the following weight corrections (considering a learning rate $\eta = 0.2$):

$\Delta w_{2\,1}(1) = \eta \delta_2(1) x_1 = 0.2 * -0.06179 * 0 = 0$

$\Delta w_{2\,2}(1) = \eta \delta_2(1) x_2 = 0.2 * -0.06179 * 0 = 0$

$\Delta w_{2\,3}(1) = \eta \delta_2(1) y_1(1) = 0.2 * -0.06179 * 0.18243 = -0.00225$

For a hidden layer neuron, the local gradient (δ_j) is given by:

$$\delta_j(n) = a y_j(n)\left[1 - y_j(n)\right] \sum_k \delta_k(n) w_{kj}(n)$$

$\delta_1(1) = a y_1(1)[1 - y_1(1)][\delta_2(1) w_{2\,3}(1)]$

$\delta_1(1) = 1 * 0.18243 * [1 - 0.18243] * [-0.06179 * -2] \sim 0.01843$

From this, it is possible to calculate the following weight corrections:

$\Delta w_{1\,1}(1) = \eta \delta_1(1) x_1 = 0.2 * 0.01843 * 0 = 0$

$\Delta w_{1\,2}(1) = \eta \delta_1(1) x_2 = 0.2 * 0.01843 * 0 = 0$

Lastly, the weights can be updated by adding the weight correction to the current weight:

$$w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n)$$

$w_{1\,1}(2) = 1 - 0 = 1$ $w_{1\,2}(2) = 1 - 0 = 1$

$w_{2\,1}(2) = 1 - 0 = 1$ $w_{2\,2}(2) = 1 - 0 = 1$

$w_{2\,3}(2) = -2 - 0.00225 = -2.00225$

When it comes to the biases, these should also be updated, firstly by calculating the bias correction and then the bias update (Note: In this formulation, $w_{1\,0} = b_1$ and $w_{2\,0} = b_2$):

$\Delta w_{2\,0}(1) = \eta \delta_2(1) y_{b_2} = 0.2 * -0.06179 * 1 \sim -0.01236$

$\Delta w_{1\,0}(1) = \eta \delta_1(1) y_{b_1} = 0.2 * 0.01843 * 1 \sim 0.00369$

$w_{2\,0}(2) = -0.5 - 0.01236 = -0.51236$ $w_{1\,0}(2) = -1.5 + 0.00369 = -1.49631$