CpE 520: HW#6

West Virginia University

Ali Zafari

Contents

Question 1

Part (a): Decision Boundaries

The neural network of question 1 is repeated in figure 1. Nodes with indices equal to 0, have the value of 1 to generate the bias with their corresponding weights.

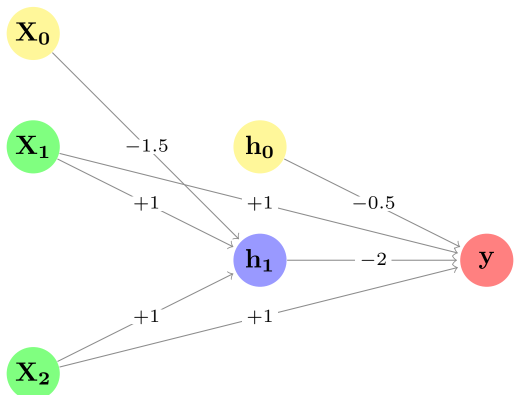


Figure 1: XOR with only one node at hidden layer

To find the decision boundaries we are going to find the equations for the hidden layer and the output layer. We assumed Threshold Function as the activation function of every node.

$$\mathbf{h_1} = \varphi(\mathbf{X_1} + \mathbf{X_2} - \mathbf{1.5})$$
$$\mathbf{y} = \varphi(-\mathbf{2h_1} + \mathbf{X_1} + \mathbf{X_2} - \mathbf{0.5})$$

and the definition of Threshold function, $\varphi(x)$ is as follows:

$$\varphi(x) = \begin{cases} 1 & \text{if} \quad x \geq 0 \\ 0 & \text{if} \quad x < 0 \end{cases}$$

Decision boundary equation for hidden node could be obtained by putting the argument of its activation function equal to zero as:

$$X_1 + X_2 - 1.5 = 0$$

Decision boundary for node $\mathbf{h_1}$ is plotted in figure 2. As it can be seen from that figure, this node does a Boolean AND operation on $\mathbf{X_1}$ and $\mathbf{X_2}$.

Now we can discuss how the output node performs Boolean XOR with the aid of the only one hidden node. The decision boundary equation for output node is as follows:

$$-2h_1 + X_1 + X_2 - 0.5 = 0 \qquad (*)$$

We know that the value of $\mathbf{h_1}$ is either one or zero depending on the inputs. Therefore we can imagine 2 ways of decision making for the output node.

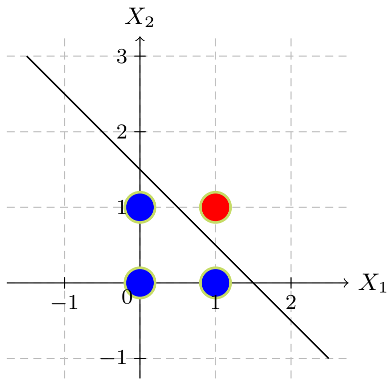
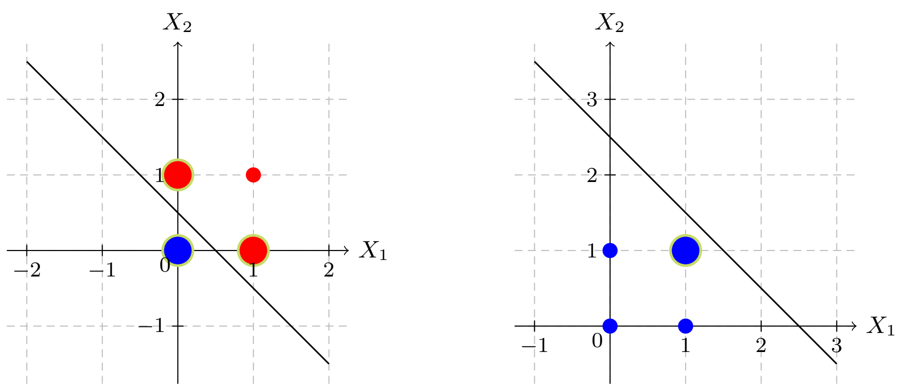


Figure 2: Node $\mathbf{h_1}$ decision boundary (AND operation)

They are summarized below:

$\mathbf{h_1} = \mathbf{0}$ (hidden node off)	$X_1 + X_2 - 0.5 = 0$	,	$\mathbf{X_1X_2} \in \{00, 01, 10\}$
$\mathbf{h_1} = \mathbf{1}$ (hidden node on)	$X_1 + X_2 - 2.5 = 0$	,	$\mathbf{X_1X_2} = \mathbf{11}$

Now, we can plot the output node decision boundaries in two $\mathbf{X_1X_2}$ planes as it is in figure 3.



(a) when hidden node is off (b) when hidden node is on

Figure 3: Decision boundaries of output node ($\mathbf{y}$)

Another approach to view the decision boundary at the output node is to consider its hyperplane as a plane in 3D space, instead of a line in 2D space, as we did above. The plane of the equation $(*)$ in the 3d space of X_1, X_2 and h_1 is plotted in figure 4. As it can be concluded form the truth table of the next section, the points $(0, 1, 0)$ and $(1, 0, 0)$ correspond to $X_1X_2 = (0, 1)$ and $X_1X_2 = (1, 0)$ respectively. Besides, the points $(0, 0, 0)$ and $(1, 1, 1)$ correspond to $X_1X_2 = (0, 0)$ and $X_1X_2 = (1, 1)$ respectively.

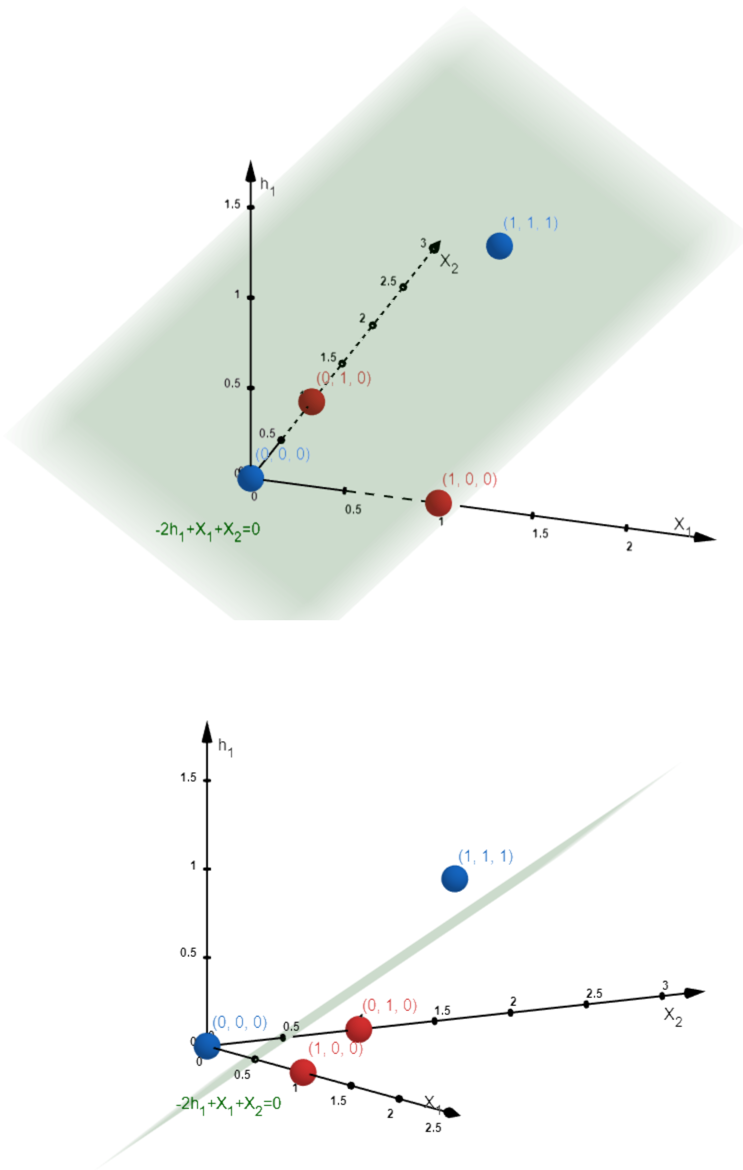


Figure 4: Decision boundary of output node (**y**) in 3D space form 2 viewpoints

Part (b): Truth Table

Table 1: Truth Table

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{h_1}$	$\mathbf{y}$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Question 2

Part I

(a): Decision Boundaries

The neural network of first part of question 2 is repeated in figure 5. Nodes with indices equal to 0, have the value of 1 to generate the bias with their corresponding weights. To find the decision boundaries we are

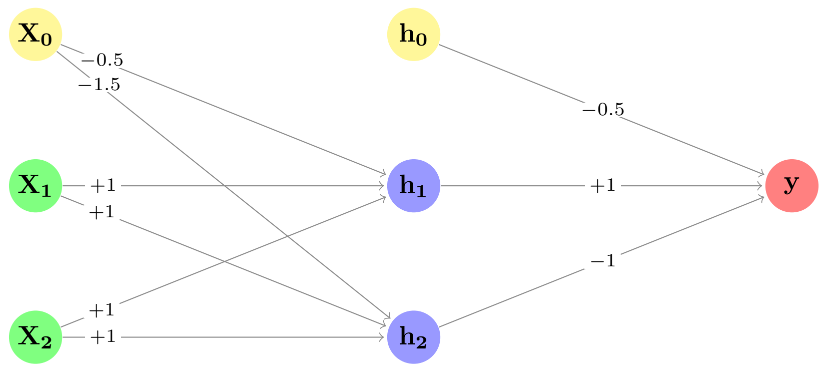
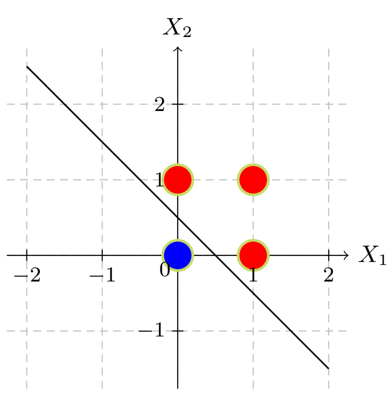


Figure 5: Neural network to check if it does Boolean XOR

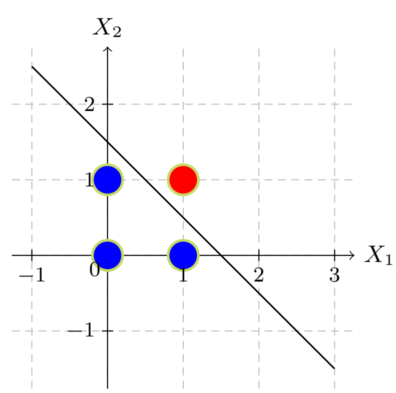
going to find the equations for the hidden layer and the output layer. We assumed Threshold Function as the activation function of every node.

$$\mathbf{h_1} = \varphi(\mathbf{X_1} + \mathbf{X_2} - \mathbf{0.5})$$
$$\mathbf{h_2} = \varphi(\mathbf{X_1} + \mathbf{X_2} - \mathbf{1.5})$$
$$\mathbf{y} = \varphi(\mathbf{h_1} - \mathbf{h_2} - \mathbf{0.5})$$

Decision boundaries for nodes $\mathbf{h_1}$ and $\mathbf{h_2}$ are plotted in figure 6. As it can be seen from that figure, $\mathbf{h_1}$ does a Boolean OR operation on $\mathbf{X_1}$ and $\mathbf{X_2}$ and $\mathbf{h_2}$ does AND on them.



(a) Node $\mathbf{h_1}$ decision boundary (OR)



(b) Node $\mathbf{h_2}$ decision boundary (AND)

Figure 6: Decision boundaries of hidden layer

Question 2 cont'd on next page. . .

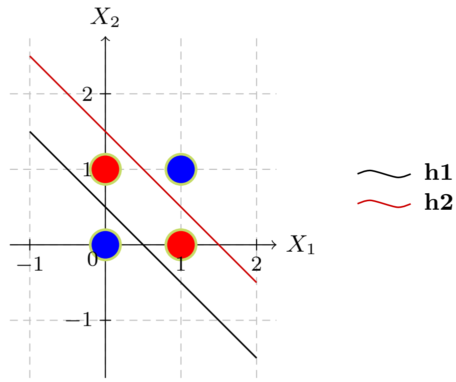


Figure 7: Output decision boundary

We can also plot the decision boundary of the output node in the h_1h_2 plane. As it will be shown in the next section with the help of a truth table, the points $X_1X_2 = (0,1), (1,0)$ are both mapped to the point $h_1h_2 = (1,0)$ and the hyperplane (line) of the output node could successfully discriminate them from the other two points to solve the XOR problem.

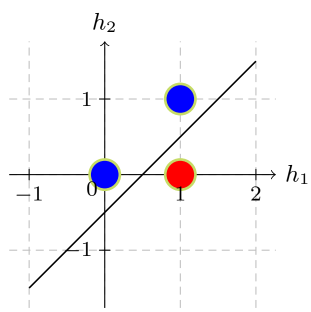


Figure 8: Output decision boundary (h_1h_2 plane)

(b): Truth Table

Table 2: Truth Table

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{h_1}$	$\mathbf{h_2}$	**y**
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	0

As we have seen from the previous two sections, this neural network is capable of solving XOR problem.

Question 2 cont'd on next page. . .

Part II

(a): Decision Boundaries

The neural network of second part of question 2 is repeated in figure 9. Nodes with indices equal to 0, have the value of 1 to generate the bias with their corresponding weights. To find the decision boundaries we are going to find the equations for the hidden layer and the output layer. We assumed Threshold Function as the activation function of every node.

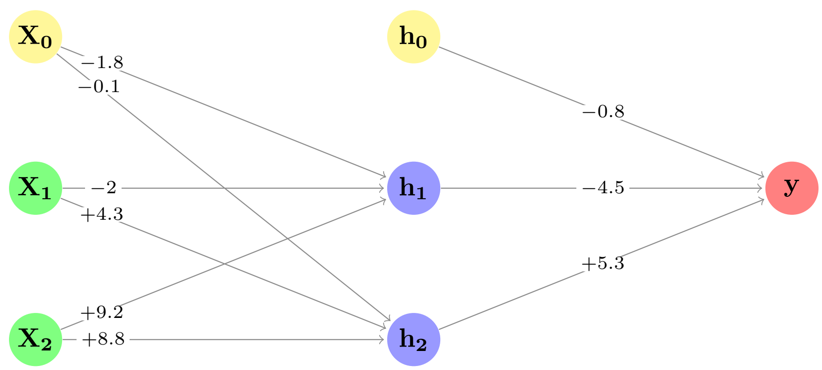
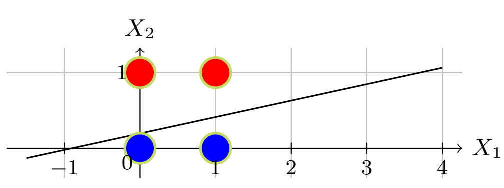


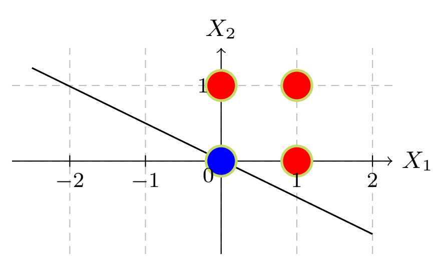
Figure 9: Neural network to check if it does Boolean XOR

$$\mathbf{h_1} = \varphi(\mathbf{-2X_1 + 9.2X_2 - 1.8})$$
$$\mathbf{h_2} = \varphi(\mathbf{4.3X_1 + 8.8X_2 - 0.1})$$
$$\mathbf{y} = \varphi(\mathbf{-4.5h_1 + 5.3h_2 - 0.8})$$

Decision boundaries for nodes $\mathbf{h_1}$ and $\mathbf{h_2}$ are plotted in figure 10. As it can be seen from that figure, $\mathbf{h_2}$ does a Boolean OR operation on $\mathbf{X_1}$ and $\mathbf{X_2}$ but $\mathbf{h_1}$'s classification is not a known Boolean operation.



(a) Node $\mathbf{h_1}$ decision boundary



(b) Node $\mathbf{h_2}$ decision boundary (OR)

Figure 10: Decision boundaries of hidden layer

(the line in figure 10b goes slightly above the point (0, 0))

Now we can plot decision boundary of the output node in the h_1h_2 plane. As it will be shown in the next section with the help of a truth table, the points $X_1X_2 = (0,1),(1,1)$ are both mapped to the point $h_1h_2 = (1,1)$ and the hyperplane (line) of the output node will pass exactly through this point, so they will both get the same label and apparently this is not correct for an XOR problem.

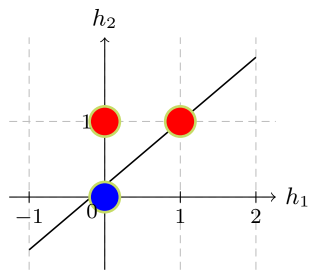


Figure 11: Output decision boundary (h_1h_2 plane)

(b): Truth Table

For the inputs $X_1X_2 \in \{01, 11\}$ input argument of the φ will be exactly zero, so it depends on which value we have defined for the $\varphi(0)$. Therefore, by any definition, this neural network will not solve the XOR problem correctly.

Table 3: Truth Table

$\mathbf{X_1}$	$\mathbf{X_2}$	$\mathbf{h_1}$	$\mathbf{h_2}$	**y**
0	0	0	0	0
0	1	1	1	1 (or 0)?
1	0	0	1	1
1	1	1	1	1 (or 0)?