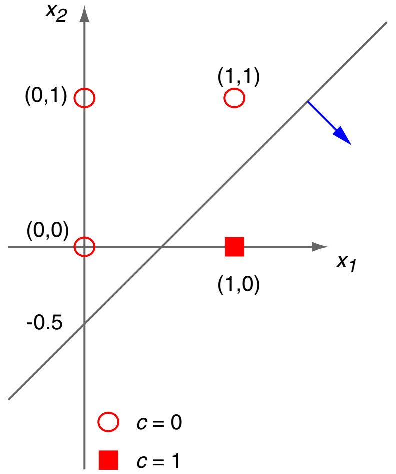
Neural Networks

Exercise 1 : Perceptron Learning

In this exercise, you design a single perceptron with two inputs x_1 and x_2. This perceptron shall implement the boolean formula $A \wedge \neg B$ with a suitable function $y(x_1, x_2)$. Use the values 0 for *false* and 1 for *true*.

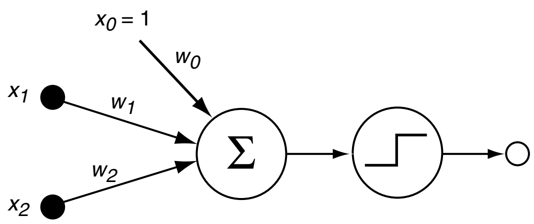
(a) Draw all possible examples and a suitable decision boundary in a coordinate system.

Answer



(b) Draw the graph of the perceptron. The schematic must include x_1, x_2, and all model weights.

Answer



(c) Manually determine a set of suitable weights $\mathbf{w} = (w_0, w_1, w_2)$ from your drawings.

Answer

In our drawing the normal vector is in the direction of $(1, -1)^T$. So we set $w_1 = 1$ and $w_2 = -1$. We use the intercept with x_2-axis at -0.5 to get w_0: $w_0 \cdot 1 + w_1 \cdot 0 + w_2 \cdot -0.5 = 0 \Leftrightarrow w_0 = -0.5$.

Check for $(1, 0)^T$:

$$\mathit{heaviside}(\mathbf{w}^T\mathbf{x}) = \mathit{heaviside}\left((-0.5, 1, -1) \cdot \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}\right) = \mathit{heaviside}(0.5) = 1$$

Check for $(1, 1)^T$:

$$\mathit{heaviside}(\mathbf{w}^T\mathbf{x}) = \mathit{heaviside}\left((-0.5, 1, -1) \cdot \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}\right) = \mathit{heaviside}(-0.5) = 0$$

Exercise 2 : Perceptron Learning

Why can the boolean formula A *XOR* B not be learned by a single perceptron? Justify your answer with a drawing.

Answer

The function for A *XOR* B cannot be implemented by a single perceptron because the data is not linearly separable; can be visualized as follows, with A along the x-axis, and B along the y-axis:

+	-
-	+

Clearly, there exists no single line through this space such that all the positive examples lie on one side, and all the negative examples on the other.

Exercise 3 : Parameters of the Multilayer Perceptrons

In this exercise, you analyze the number of weights (parameters) of multilayer perceptrons. We use the notation from the lecture (e.g., slide ML:IV-104), where multilayer perceptrons have d layers, p attributes, hidden layer i with l_i units, and an output layer with k units.

(a) Let $d = 4$, $p = 7$, $l_1 = 5$, $l_2 = 3$, $l_3 = 3$, and $k = 4$. Draw the graph of the multilayer perceptron.

Answer

Connections are omitted here (fully connected from one layer to next, except that "①" have no input connection); "◯" are variable nodes; "①" are constant / bias nodes.

$x_0 \equiv$ ①				
$x_1 =$ ◯	①			
$x_2 =$ ◯	◯	①	①	◯ $= y_1$
$x_3 =$ ◯	◯	◯	◯	◯ $= y_2$
$x_4 =$ ◯	◯	◯	◯	◯ $= y_3$
$x_5 =$ ◯	◯	◯	◯	◯ $= y_4$
$x_6 =$ ◯	◯			
$x_7 =$ ◯				

(b) Calculate the number of weights in the multilayer perceptron of (a).

Answer

$8 \cdot 5 + 6 \cdot 3 + 4 \cdot 3 + 4 \cdot 4 = 40 + 18 + 12 + 16 = 86$

(c) Calculate the number of weights in the multilayer perceptron of (a) but with each l_i doubled, i.e., $l_1 = 10$, $l_2 = 6$, $l_3 = 6$. Has the number of weights doubled as well?

Answer

$8 \cdot 10 + 11 \cdot 6 + 7 \cdot 6 + 7 \cdot 4 = 80 + 66 + 42 + 28 = 216$

(d) Let $f(p, l_1, \ldots, l_{d-1}, k)$ be a function that computes the number of weights in the general case. Write down an expression for f.

Answer

$f(p, l_1, \ldots, l_{d-1}, k) = (p + 1) \cdot l_1 + \sum_{i=2}^{d-1}(l_{i-1} + 1) \cdot l_i + (l_{d-1} + 1) \cdot k$