1. How a single perceptron can be used to represent the Boolean functions such as AND, OR

- **Boolean function AND**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>( A \land B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\( w_0 = -0.8 \)
\( w_1 = 0.5 \)
\( w_2 = 0.5 \)

\( O(x_1, \ldots , x_n) = \begin{cases} 1 & \text{if } w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n > 0 \\ -1 & \text{otherwise} \end{cases} \)

1) \( \text{if } A = 0 \land B = 0 \Rightarrow O = 0 \times 0 - 0.8 + (0.5 \times 0) + (0.5 \times 0) = -0.8 < 0 \text{ so, output} = 0 \)

2) \( \text{if } A = 0 \land B = 1 \Rightarrow O = -0.8 + (0.5 \times 0) + (0.5 \times 1) = -0.3 < 0 \text{ so, output} = 0 \)

3) \( \text{if } A = 1 \land B = 1 \Rightarrow O = -0.8 + (0.5 \times 1) + (0.5 \times 0) = -0.3 < 0 \text{ output} = 0 \)

4) \( \text{if } A = 1 \land B = 1 \Rightarrow O = -0.8 + (0.5 \times 1) + (0.5 \times 1) = 0.2 > 0 \text{ output} = 1 \)
Boolean function 0L

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>A V B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ x_0 = 1 \]
\[ w_1 = 0.5 \]
\[ w_0 = -0.3 \]

\[ \sum_{i=0}^{n} w_i x_i \]

1) \( A = 0 \), \( B = 0 \) \( \Rightarrow \) \(-0.3 + (0.5 \times 0) + (0.5 \times 0)\)
   \( = -0.3 < 0 \) \( \Rightarrow \) Output = 0

2) \( A = 0 \), \( B = 1 \) \( \Rightarrow \) \(-0.3 + (0.5 \times 0) + (0.5 \times 1)\)
   \( = 0.2 > 0 \) \( \Rightarrow \) Output = 1

3) \( A = 1 \), \( B = 0 \) \( \Rightarrow \) \(-0.3 + (0.5 \times 1) + (0.5 \times 0)\)
   \( = 0.2 > 0 \) \( \Rightarrow \) Output = 1

4) \( A = 1 \), \( B = 1 \) \( \Rightarrow \) \(-0.3 + (0.5 \times 1) + (0.5 \times 0)\)
   \( = 0.7 > 0 \) \( \Rightarrow \) Output = 1
Design a two-input perceptron that implements the boolean function $A \land \neg B$. Design a two-layer network of perceptron’s that implements $A \text{ XOR } B$.

The perceptron has two input $A$, $B$ and constant 1

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$\neg B$</th>
<th>$A \land \neg B$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0 (-1)</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0 (-1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0 (-1)</td>
</tr>
</tbody>
</table>

The values of $A$ & $B$ are 1 (true) or -1 or 0 for false.

- The line crosses the $A$ axis at 1 and $B$ axis at -1
- The weights are $w_0 = -1$, $w_1 = 1$, $w_2 = -1$.
b) A XOR B cannot be calculated by a single perceptron, so build a two-layer network of perceptrons.

* Express A XOR B in terms of other logical connectives:
  \[ A \text{ XOR } B = (A \land \neg B) \lor (\neg A \land B) \]

* Define the perceptron \( P_1 \) and \( P_2 \) for \( (A \land \neg B) \lor (\neg A \land B) \).

* Composing the outputs of \( P_1 \) and \( P_2 \) into a perceptron \( P_3 \) that implements \( O(P_1) \lor O(P_2) \).
3. Consider two perceptrons defined by the threshold expression $w_0 + w_1x_1 + w_2x_2 > 0$.

Perceptron A has weight values

$$w_0 = 1, w_1 = 2, w_2 = 1$$

and perceptron B has the weight values

$$w_0 = 0, w_1 = 2, w_2 = 1$$

True or false? Perceptron A is more-general than perceptron B.

**Solution**

True, Perceptron A is more-general than Perceptron B.

$$(x_1, \ldots, x_n) = w_0x_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n$$

- For perceptron B, \( w_0 = 1, w_1 = 2, w_2 = 1 \)

$$0 + 2x_1 + x_2 > 0 \implies 0 + 2 + 1 > 0$$

where, \( x_0 \) is constant which is equal to 1 i.e., \( x_0 = 1 \)

- For perceptron A, \( w_0 = 1, w_1 = 2, w_2 = 1 \)

$$1 + 2x_1 + x_2 > 0 \implies 1 + 2 + 1 > 0$$

Here, Perceptron A is more general than Perceptron B

because every instance of \( x_1 \) & \( x_2 \) that satisfies

Perceptron B also satisfies Perceptron A.