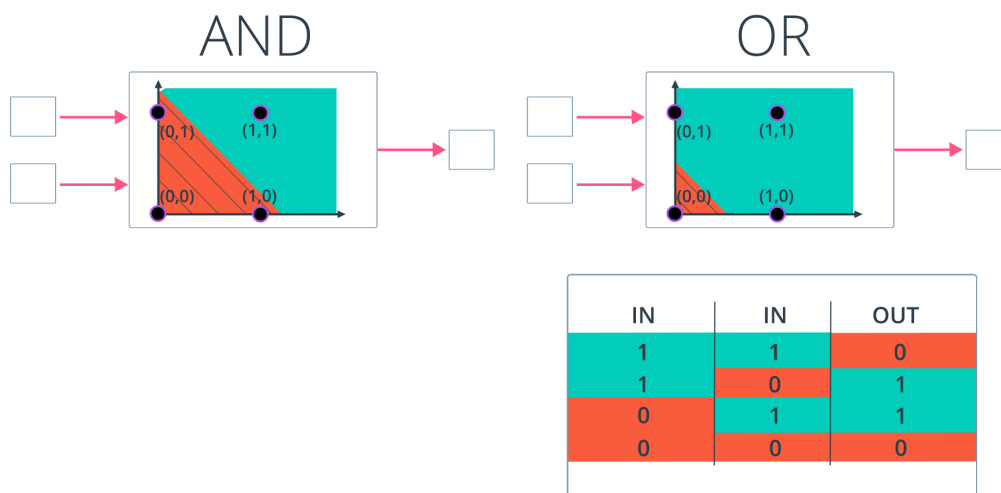


OR Perceptron

The OR perceptron is very similar to an AND perceptron. In the image below, the OR perceptron has the same line as the AND perceptron, except the line is shifted down. What can you do to the weights and/or bias to achieve this? Use the following AND perceptron to create an OR Perceptron.



QUESTION 1 OF 2

What are two ways to go from an AND perceptron to an OR perceptron?

Increase the weights

Decrease the weights

Increase a single weight

Increase the magnitude of the bias

Decrease the magnitude of the bias

NOT Perceptron

Unlike the other perceptrons we looked at, the NOT operations only cares about one input. The operation returns a **0** if the input is **1** and a **1** if it's a **0**. The other inputs to the perceptron are ignored.

In this quiz, you'll set the weights (**weight1**, **weight2**) and bias **bias** to the values that calculate the NOT operation on the second input and ignores the first input.

```
1  import pandas as pd
2
3  # TODO: Set weight1, weight2, and bias
4  weight1 = 0.0
5  weight2 = -1.0
6  bias = 0.0
7
8
9  # DON'T CHANGE ANYTHING BELOW
10 # Inputs and outputs
11 test_inputs = [(0, 0), (0, 1), (1, 0), (1, 1)]
12 correct_outputs = [True, False, True, False]
13 outputs = []
14
15 # Generate and check output
16 for test_input, correct_output in zip(test_inputs, correct_outputs):
17     linear_combination = weight1 * test_input[0] + weight2 * test_input[1] + bias
18     output = int(linear_combination >= 0)
19     is_correct_string = 'Yes' if output == correct_output else 'No'
20     outputs.append([test_input[0], test_input[1], linear_combination, output, is
21
22 # Print output
23 num_wrong = len([output[4] for output in outputs if output[4] == 'No'])
24 output_frame = pd.DataFrame(outputs, columns=['Input 1', 'Input 2', 'Linear
25 if not num_wrong:
26     print('Nice! You got it all correct.\n')
27 else:
28     print('You got {} wrong. Keep trying!\n'.format(num_wrong))
29 print(output_frame.to_string(index=False))
```

OR & NOT Perceptron ...

0	0	0.0	1	Yes
0	1	-1.0	0	Yes
1	0	0.0	1	Yes
1	1	-1.0	0	Yes

We have a perceptron that can do AND, OR, or NOT operations. Let's do one more, XOR. In the next section, you'll learn how a neural network solves more complicated problems like XOR.