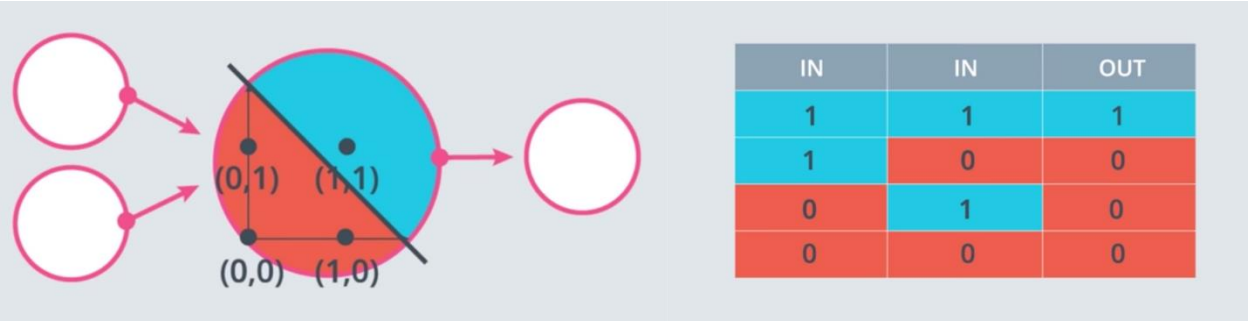
Perceptrons as Logical Operators

In this lesson, we'll see one of the many great applications of perceptrons. As logical operators! You'll have the chance to create the perceptrons for the most common of these, the **AND**, **OR**, and **NOT** operators. And then, we'll see what to do about the elusive **XOR** operator. Let's dive in!

AND Perceptron



IN	IN	OUT
1	1	1
1	0	0
0	1	0
0	0	0

What are the weights and bias for the AND perceptron?

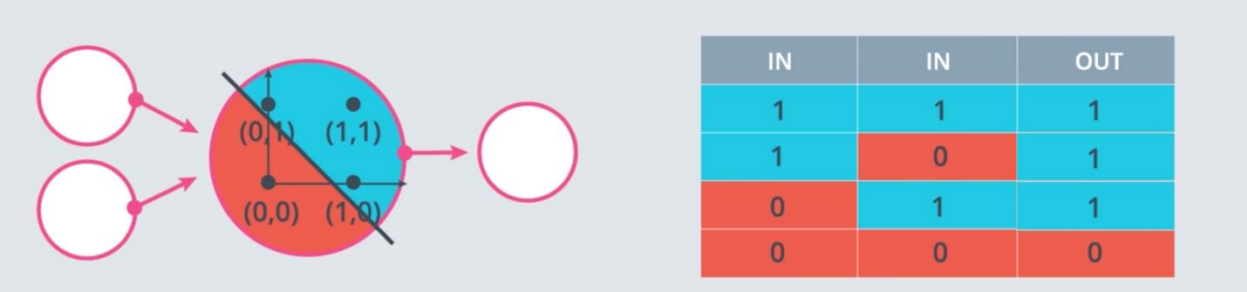
Set the weights (weight1, weight2) and bias (bias) to values that will correctly determine the AND operation as shown above.
More than one set of values will work!

New weight's values : weight1 = 1.0, weight1 = 1.0, bias = -2.0

```
import pandas as pd

# TODO: Set weight1, weight2, and bias

weight1 = 1.0

weight2 = 1.0

bias = -2.0

# DON'T CHANGE ANYTHING BELOW

# Inputs and outputs

test_inputs = [(0, 0), (0, 1), (1, 0), (1, 1)]

correct_outputs = [False, False, False, True]

outputs = []


# Generate and check output

for test_input, correct_output in zip(test_inputs, correct_outputs):

    linear_combination = weight1 * test_input[0] + weight2 * test_input[1] + bias
```

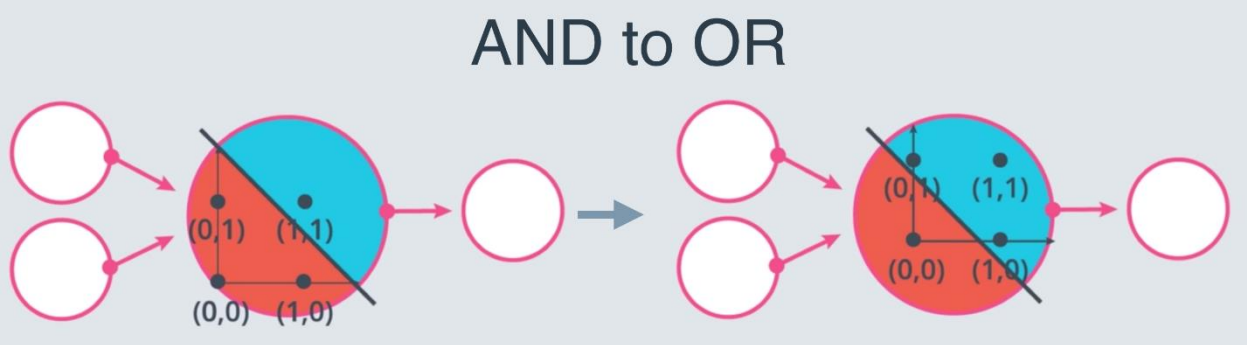
```
    output = int(linear_combination >= 0)
    is_correct_string = 'Yes' if output == correct_output else 'No'
    outputs.append([test_input[0], test_input[1], linear_combination, output, is_correct_string])

# Print output
num_wrong = len([output[4] for output in outputs if output[4] == 'No'])
output_frame = pd.DataFrame(outputs, columns=['Input 1', '  Input 2', '  Linear Combination', '
Activation Output', '  Is Correct'])
if not num_wrong:
    print('Nice!  You got it all correct.\n')
else:
    print('You got {} wrong.  Keep trying!\n'.format(num_wrong))
print(output_frame.to_string(index=False))
```

OR Perceptron



IN	IN	OUT
1	1	1
1	0	1
0	1	1
0	0	0

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 2 of 4

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

- Increase the weights
- Decrease the magnitude of the bias

NOT Perceptron

Unlike the other perceptrons we looked at, the NOT operation 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.

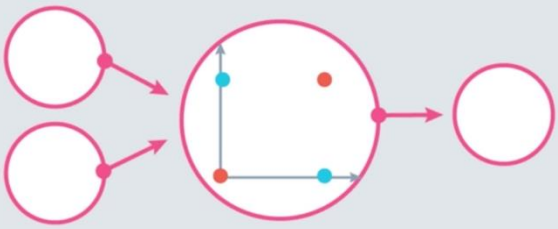
In this quiz, you'll set the weights (weight1, weight2) and bias bias to the values that calculate the NOT operation on the second input and ignores the first input.

New weight's values : weight1 = 00, weight1 = -2.0, bias = 1.0

```
import pandas as pd

# TODO: Set weight1, weight2, and bias
weight1 = 0.0
weight2 = -2.0
bias = 1.0

# DON'T CHANGE ANYTHING BELOW
# Inputs and outputs
test_inputs = [(0, 0), (0, 1), (1, 0), (1, 1)]
correct_outputs = [True, False, True, False]
outputs = []

# Generate and check output
for test_input, correct_output in zip(test_inputs, correct_outputs):
    linear_combination = weight1 * test_input[0] + weight2 * test_input[1] + bias
    output = int(linear_combination >= 0)
    is_correct_string = 'Yes' if output == correct_output else 'No'
    outputs.append([test_input[0], test_input[1], linear_combination, output, is_correct_string])
```

```
# Print output
num_wrong = len([output[4] for output in outputs if output[4] == 'No'])
output_frame = pd.DataFrame(outputs, columns=['Input 1', '  Input 2', '  Linear Combination', '
Activation Output', '  Is Correct'])
if not num_wrong:
    print('Nice!  You got it all correct.\n')
else:
    print('You got {} wrong.  Keep trying!\n'.format(num_wrong))
print(output_frame.to_string(index=False))
```

XOR Perceptron



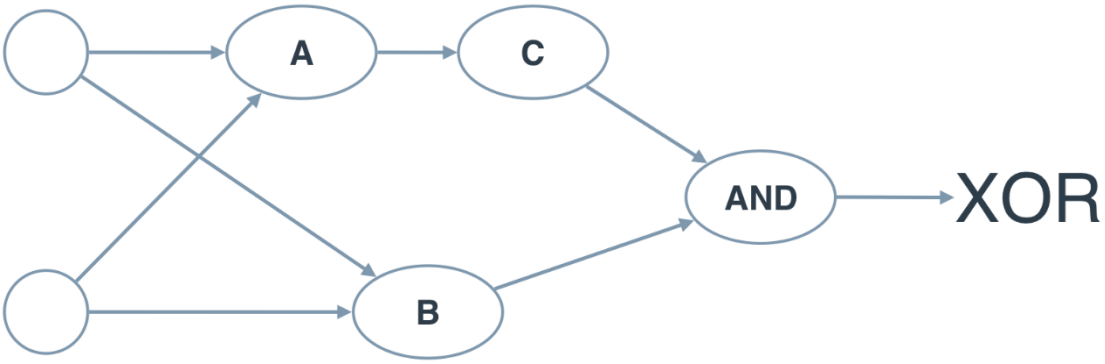
IN	IN	OUT
1	1	0
1	0	1
0	1	1
0	0	0

Quiz: Build an XOR Multi-Layer Perceptron

Now, let's build a multi-layer perceptron from the AND, NOT, and OR perceptrons to create XOR logic!

The neural network below contains 3 perceptrons, A, B, and C. The last one (AND) has been given for you. The input to the neural network is from the first node. The output comes out of the last node.

The multi-layer perceptron below calculates XOR. Each perceptron is a logic operation of AND, OR, and NOT. However, the perceptrons A, B, and C don't indicate their operation. In the following quiz, set the correct operations for the perceptrons to calculate XOR.



Question 4 of 4

Set the operations for the perceptrons in the XOR neural network.

Submit to check your answer choices!

Perceptron	Operators
A	**AND**
B	**OR**
C	**NOT**