

# Forward propagation

INTRODUCTION TO DEEP LEARNING IN PYTHON



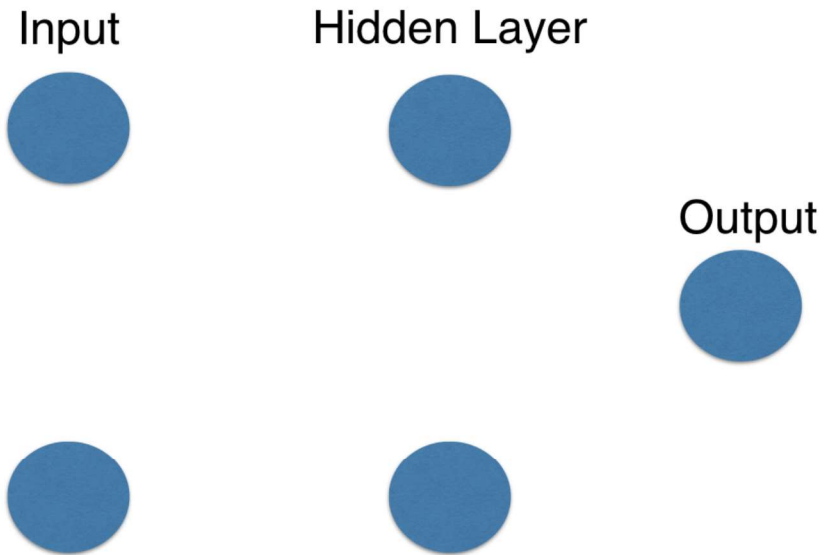
**Dan Becker**

Data Scientist and contributor to Keras  
and TensorFlow libraries

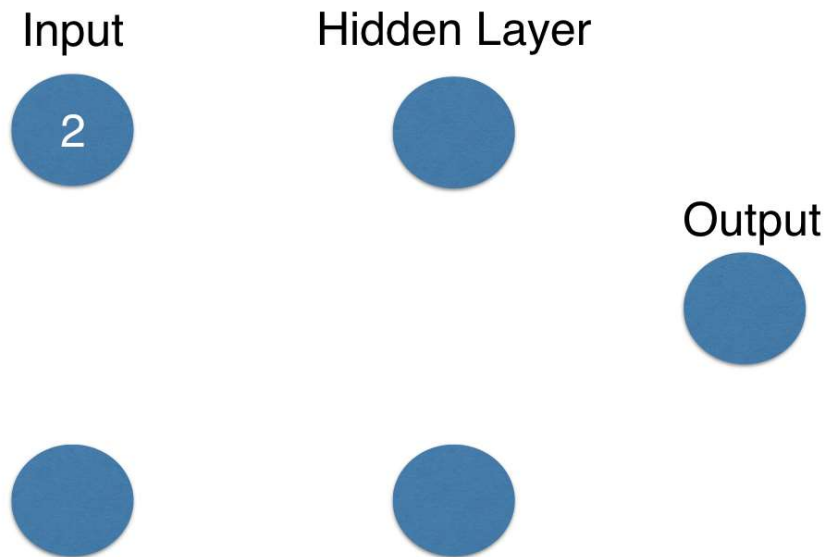
# Bank transactions example

- Make predictions based on:
  - Number of children
  - Number of existing accounts

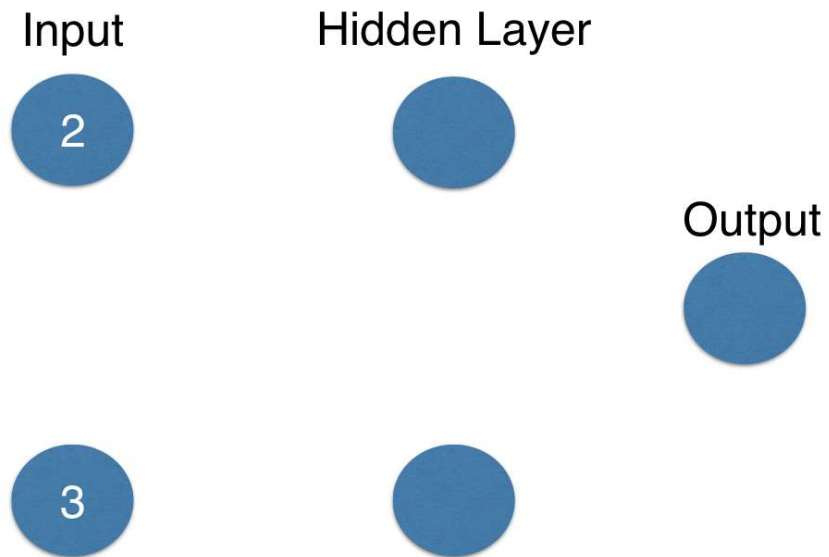
# Forward propagation



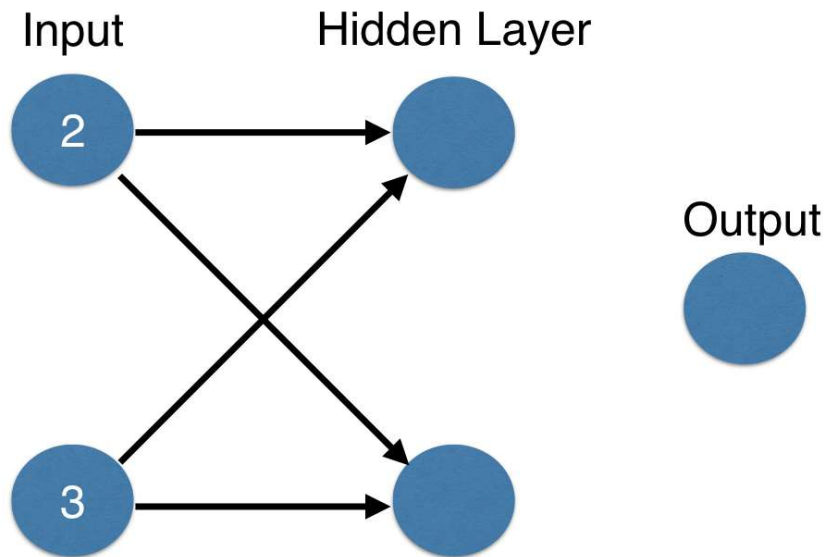
# Forward propagation



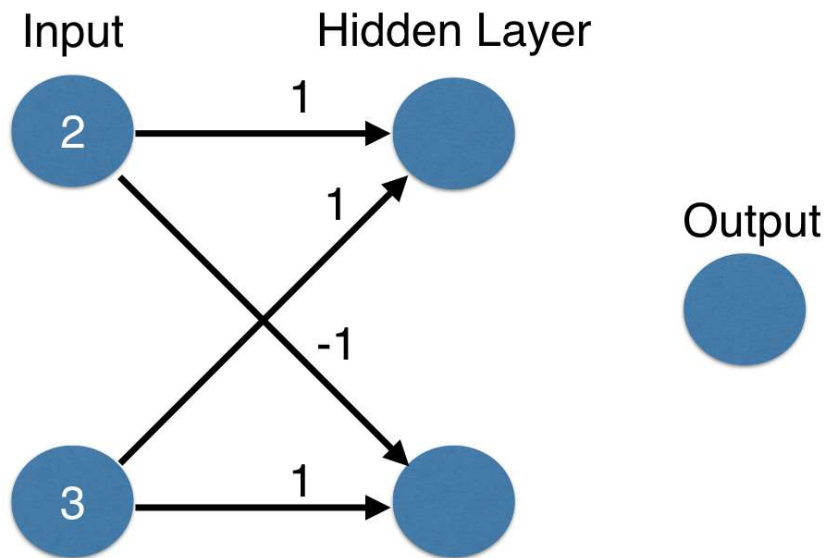
# Forward propagation



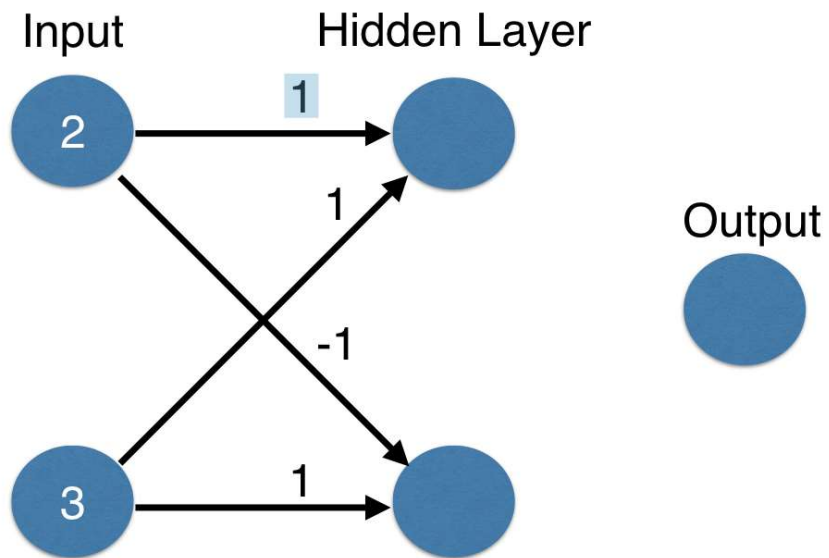
# Forward propagation



# Forward propagation

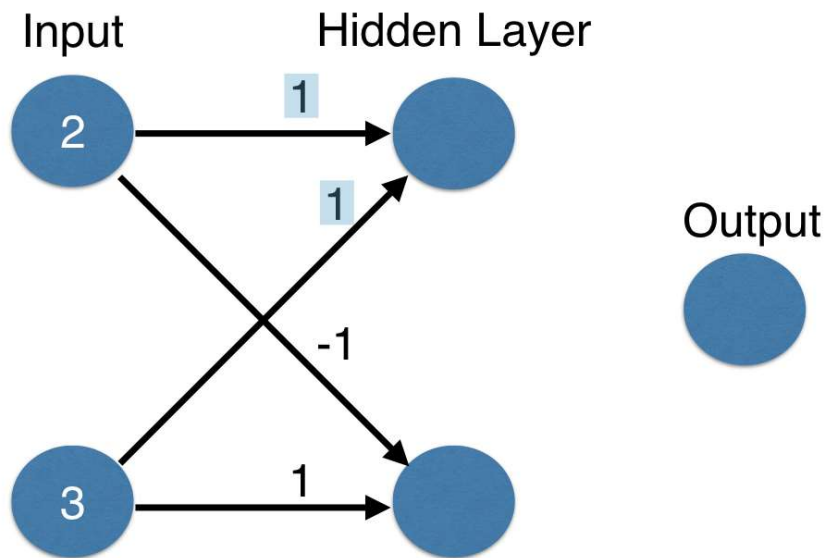


# Forward propagation

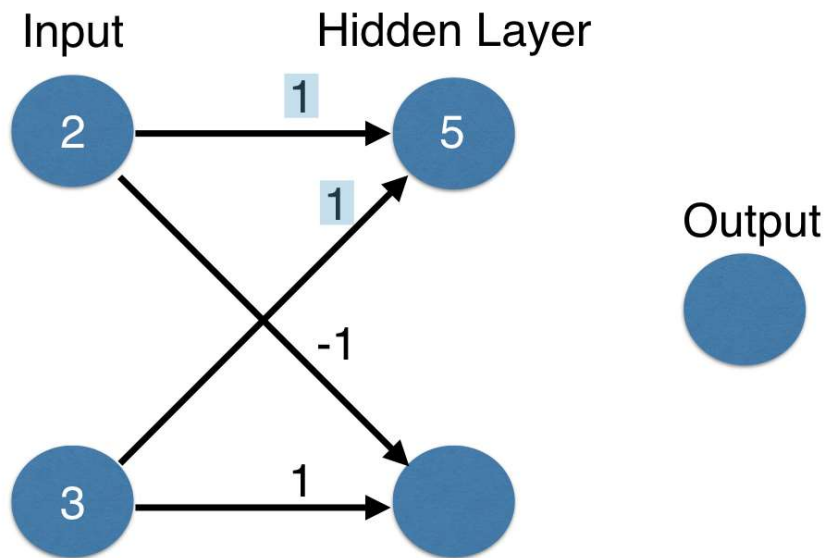




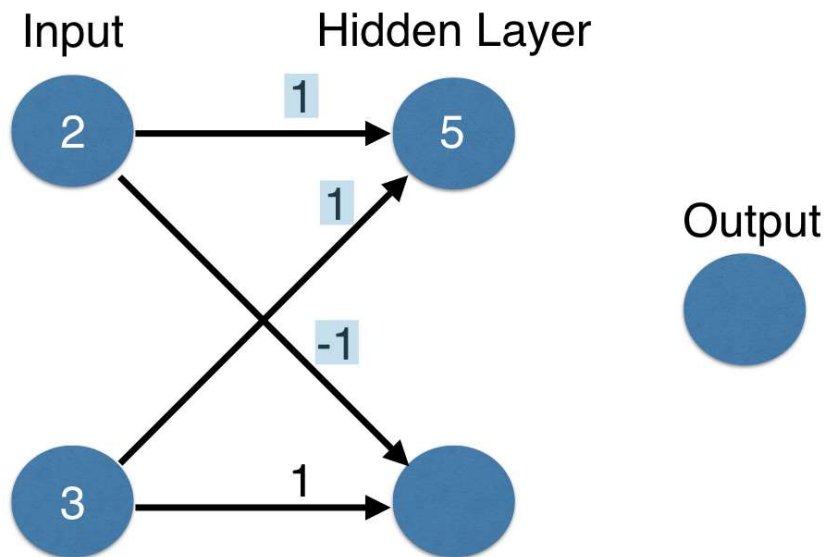
# Forward propagation



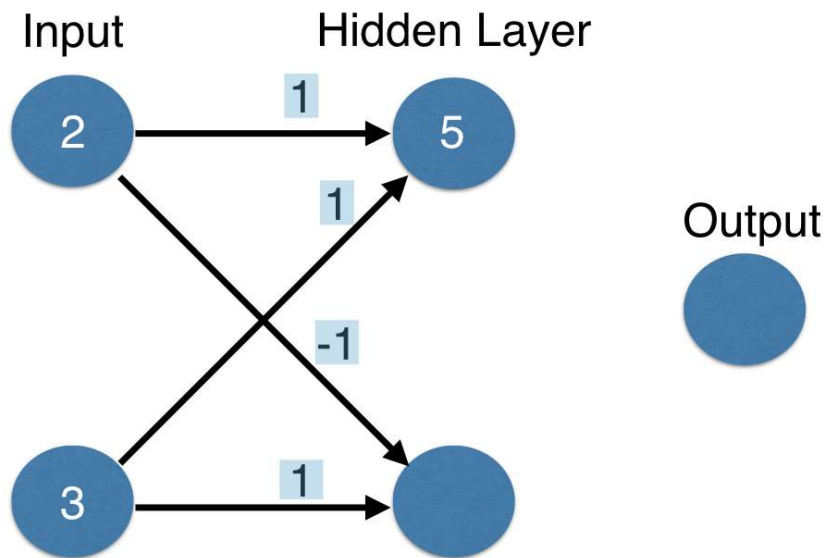
# Forward propagation



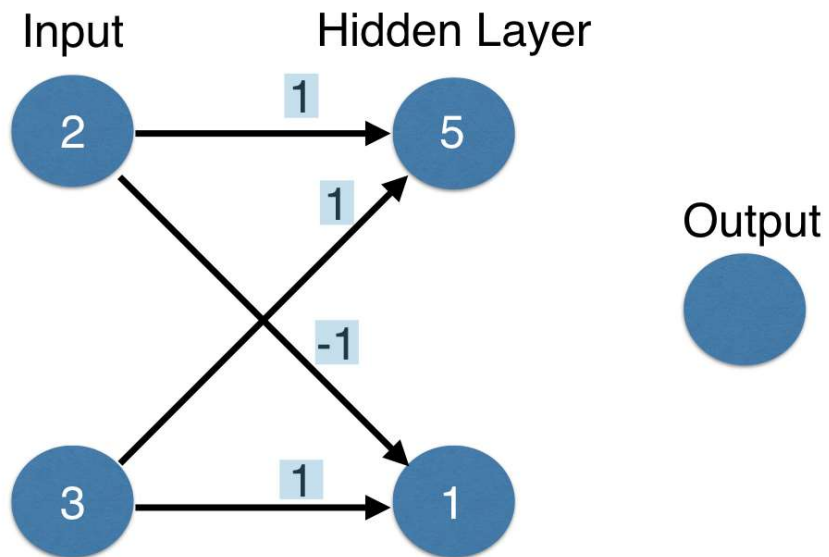
# Forward propagation



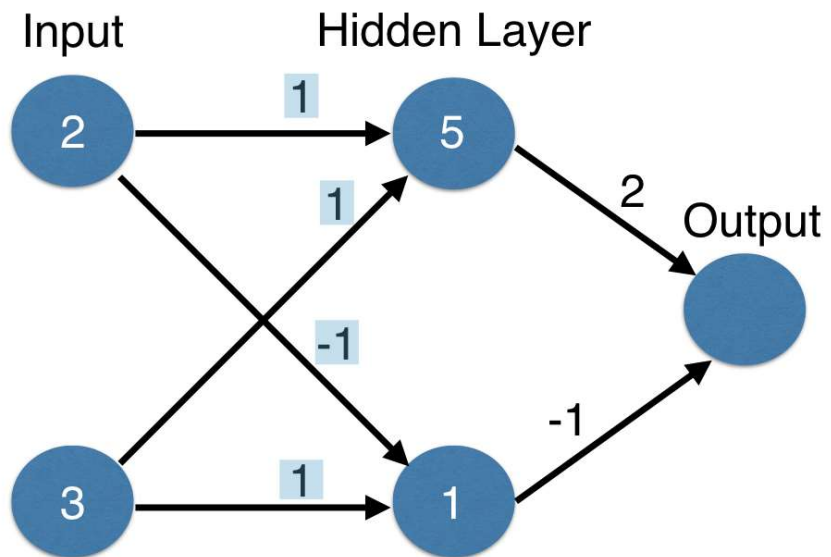
# Forward propagation



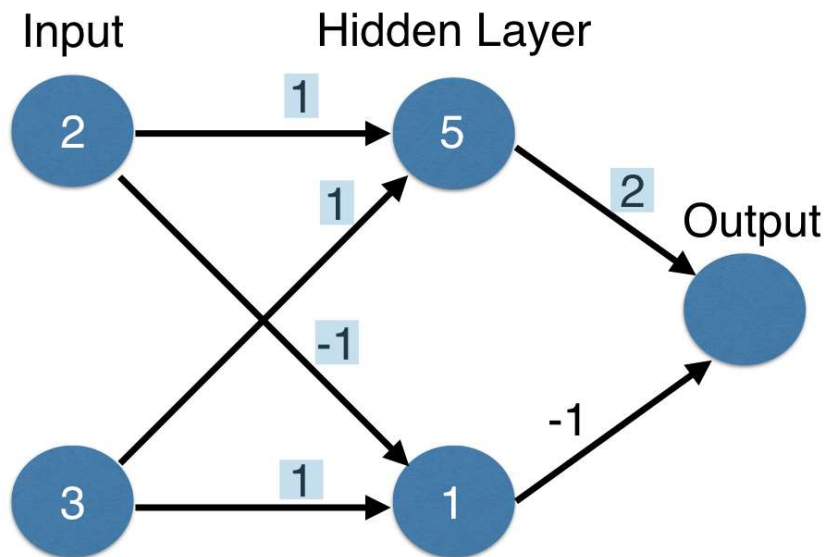
# Forward propagation



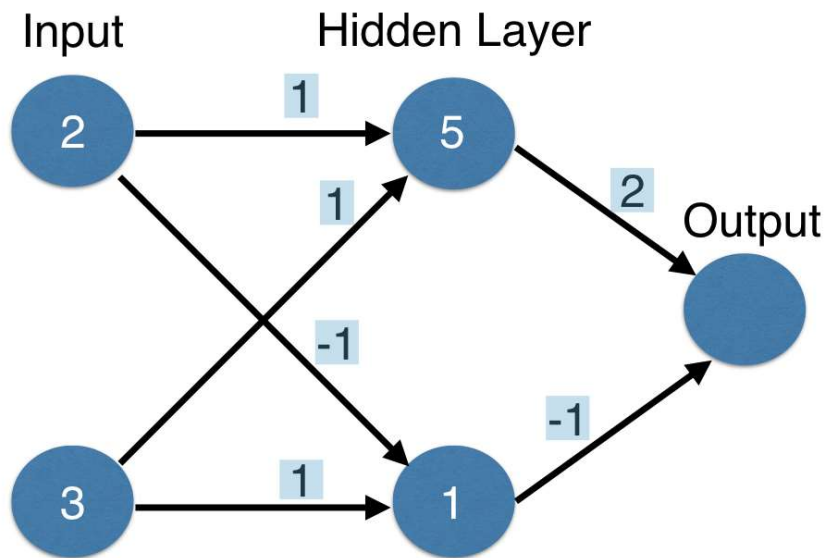
# Forward propagation



# Forward propagation

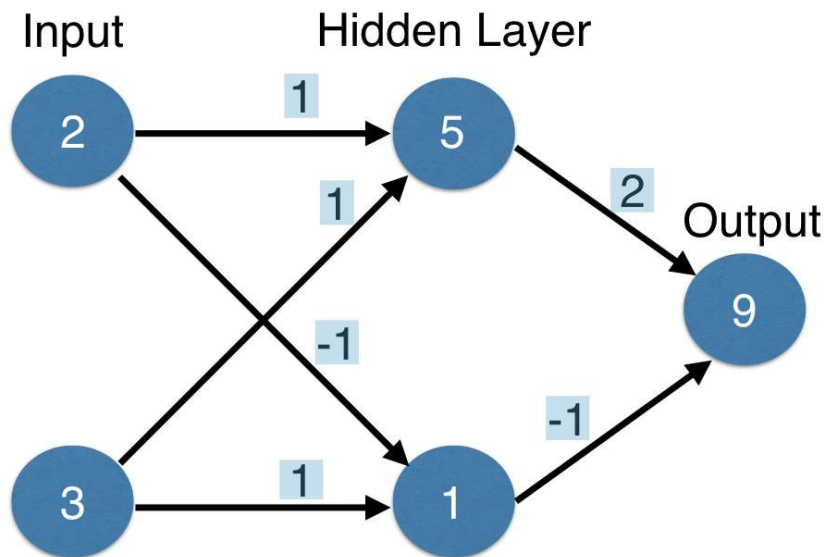


# Forward propagation

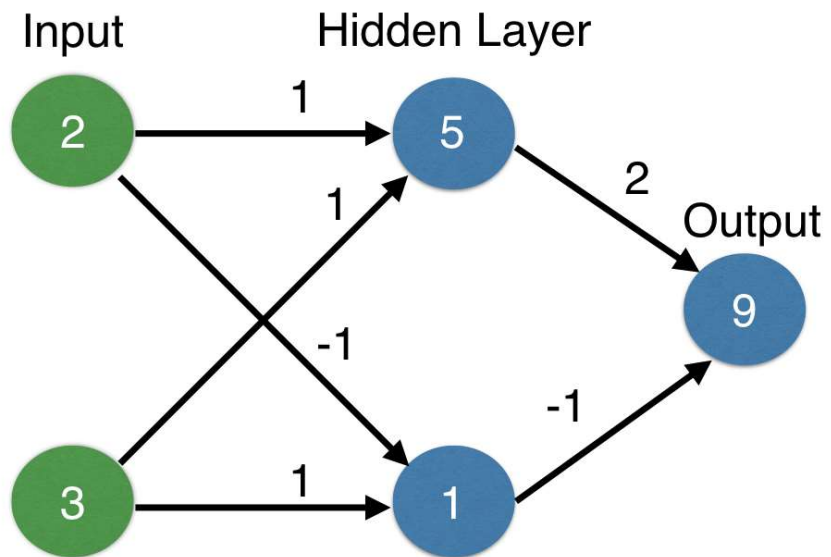




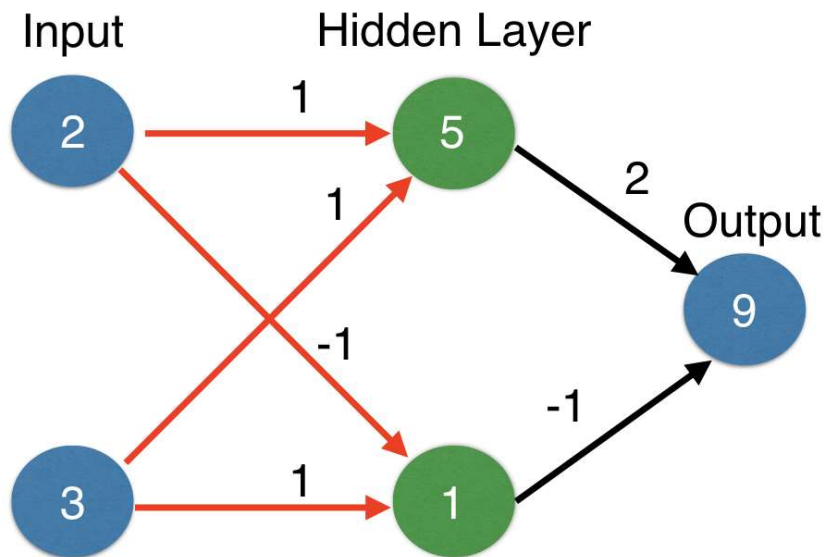
# Forward propagation



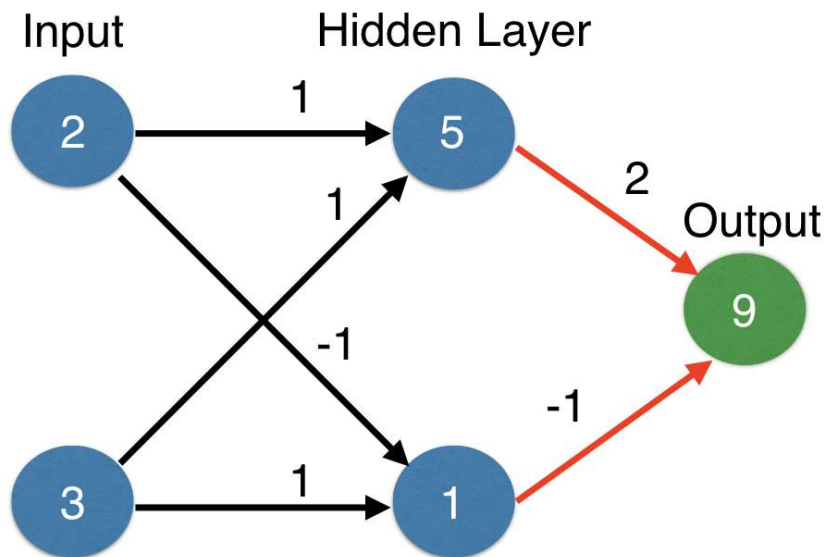
# Forward propagation



# Forward propagation



# Forward propagation



# Forward propagation

- Multiply - add process
- Dot product
- Forward propagation for one data point at a time
- Output is the prediction for that data point

# Forward propagation code

```
import numpy as np
input_data = np.array([2, 3])
weights = {'node_0': np.array([1, 1]),
           'node_1': np.array([-1, 1]),
           'output': np.array([2, -1])}
node_0_value = (input_data * weights['node_0']).sum()
node_1_value = (input_data * weights['node_1']).sum()
```

## Forward propagation code

```
hidden_layer_values = np.array([node_0_value, node_1_value])  
print(hidden_layer_values)
```

```
[5, 1]
```

```
output = (hidden_layer_values * weights['output']).sum()  
print(output)
```

```
9
```