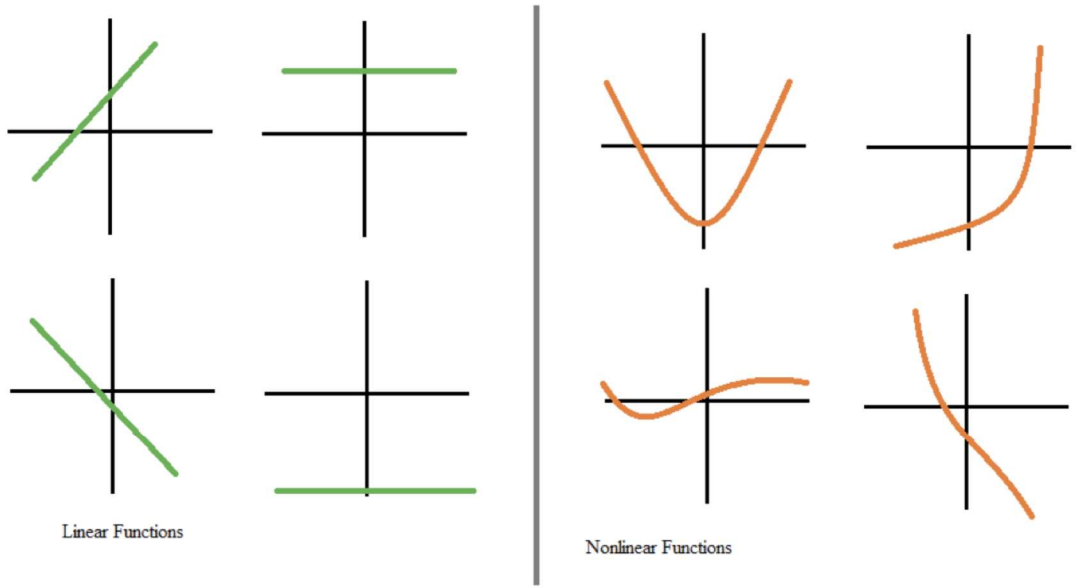
# Activation functions

INTRODUCTION TO DEEP LEARNING IN PYTHON

**Dan Becker**
Data Scientist and contributor to Keras and TensorFlow libraries

datacamp

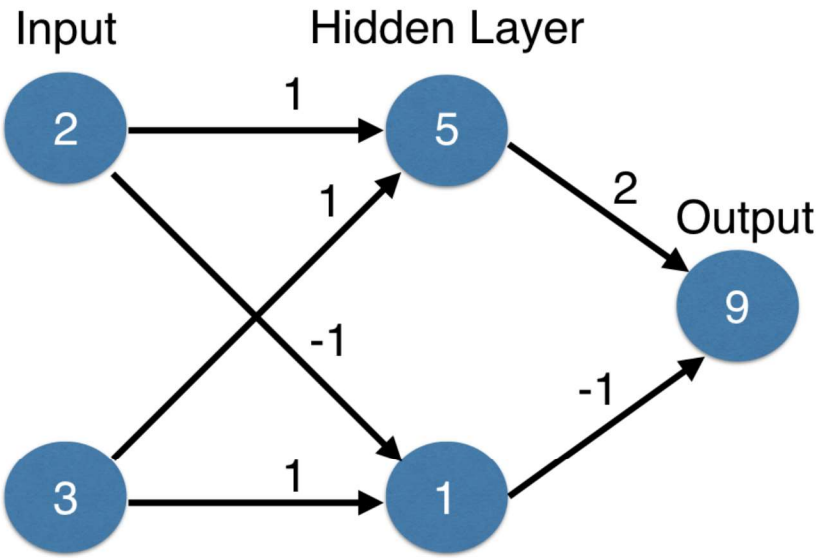
# Linear vs. non-linear Functions



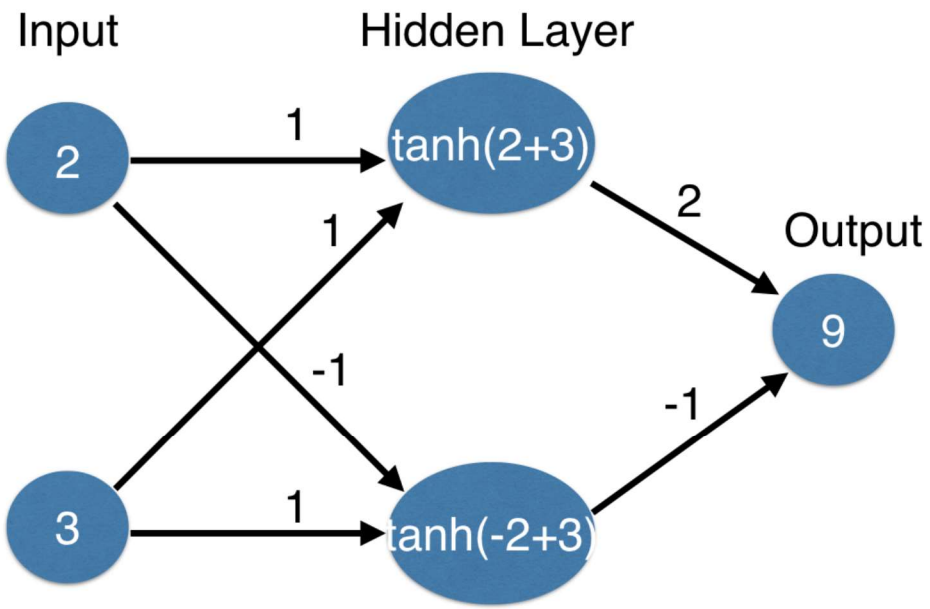
# Activation functions

- Applied to node inputs to produce node output

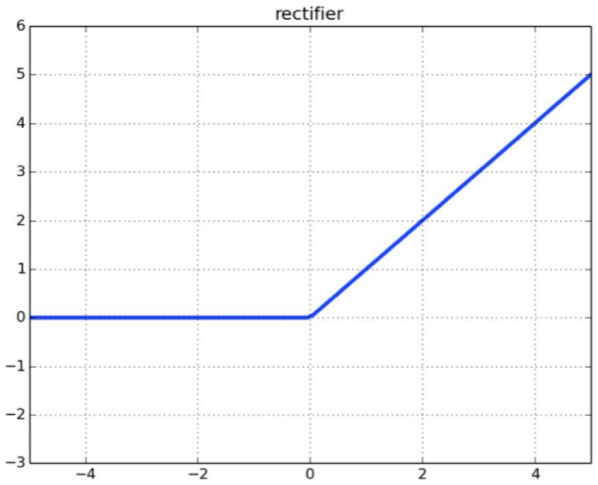
# Improving our neural network



# Activation functions



# ReLU (Rectified Linear Activation)




$$RELU(x) = \begin{cases} 0 \ if \ x < 0 \\ x \ if \ x >= 0 \end{cases}$$

# Activation functions

```
import numpy as np
input_data = np.array([-1, 2])
weights = {'node_0': np.array([3, 3]),
               'node_1': np.array([1, 5]),
               'output': np.array([2, -1])}
node_0_input = (input_data * weights['node_0']).sum()
node_0_output = np.tanh(node_0_input)
node_1_input = (input_data * weights['node_1']).sum()
node_1_output = np.tanh(node_1_input)
hidden_layer_outputs = np.array([node_0_output, node_1_output])
output = (hidden_layer_output * weights['output']).sum()
```

```
print(output)
```

```
1.2382242525694254
```

# Let's practice!

INTRODUCTION TO DEEP LEARNING IN PYTHON

# Deeper networks

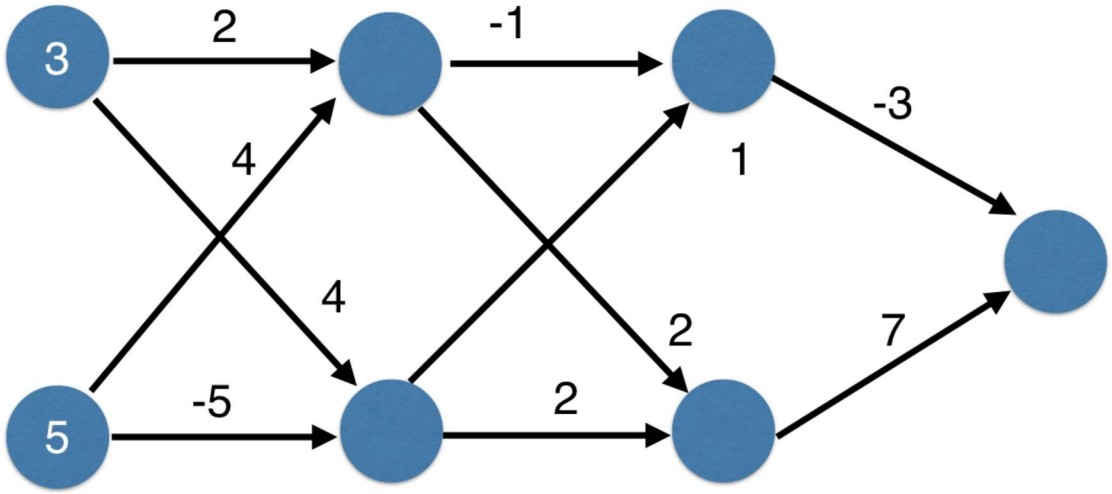
INTRODUCTION TO DEEP LEARNING IN PYTHON

**Dan Becker**

Data Scientist and contributor to Keras and TensorFlow libraries

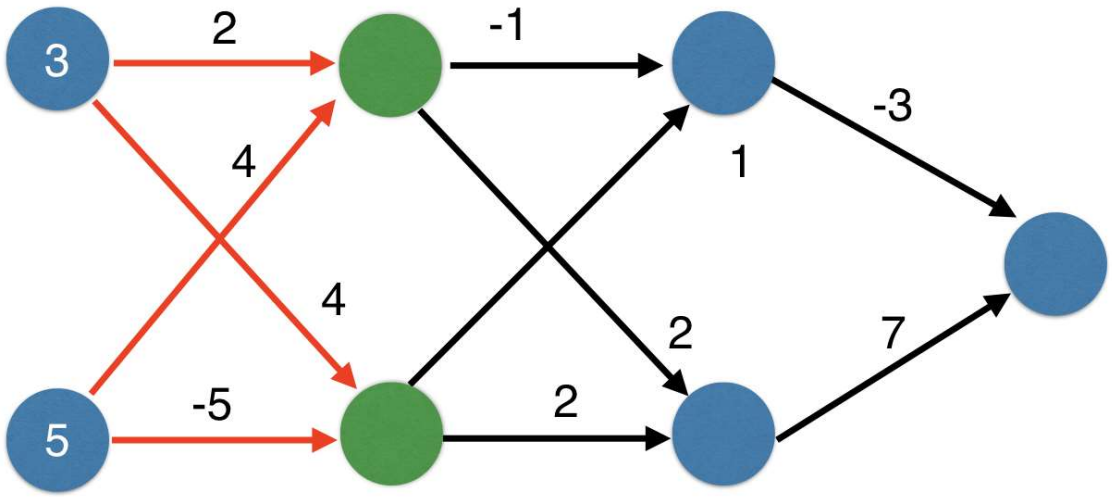
datacamp

# Multiple hidden layers



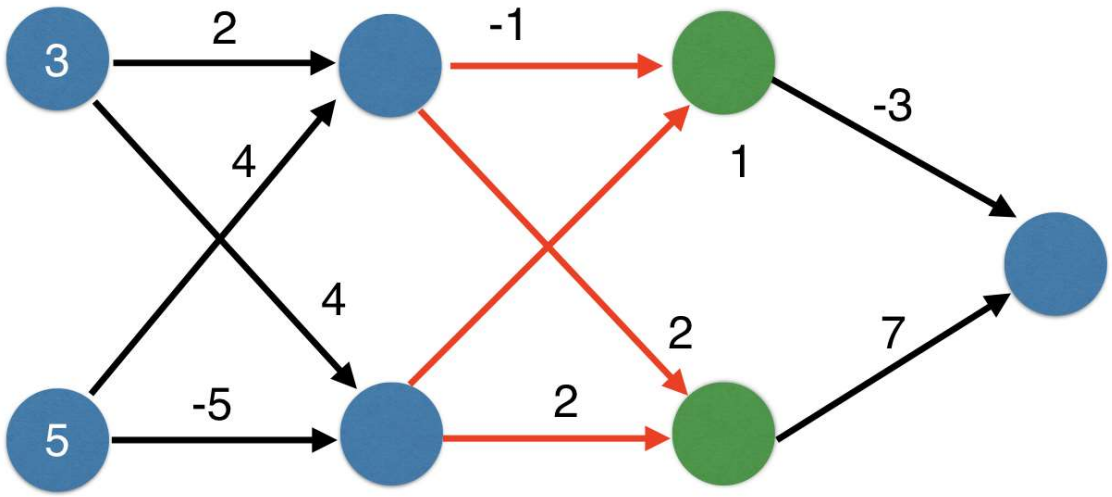

Calculate with ReLU Activation Function

# Multiple hidden layers



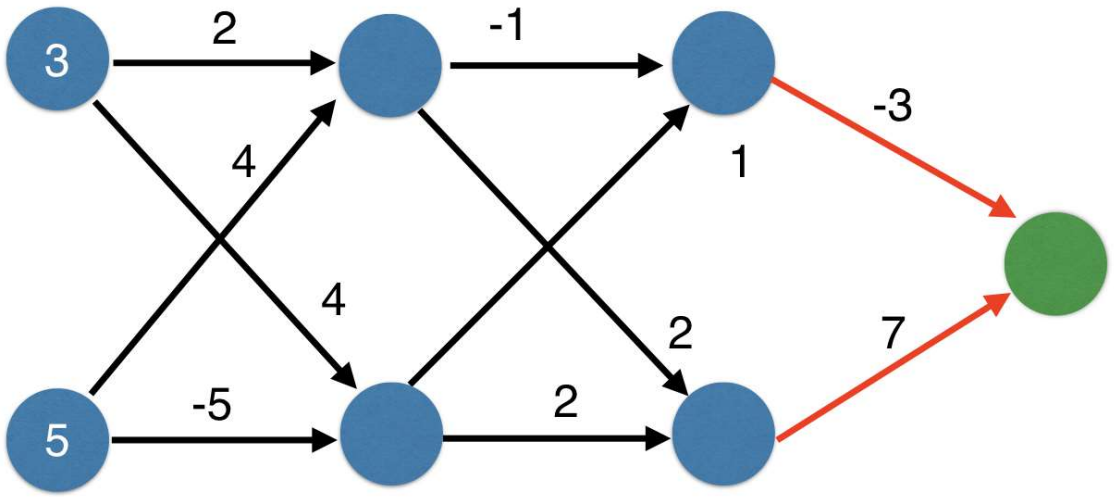

Calculate with ReLU Activation Function

# Multiple hidden layers



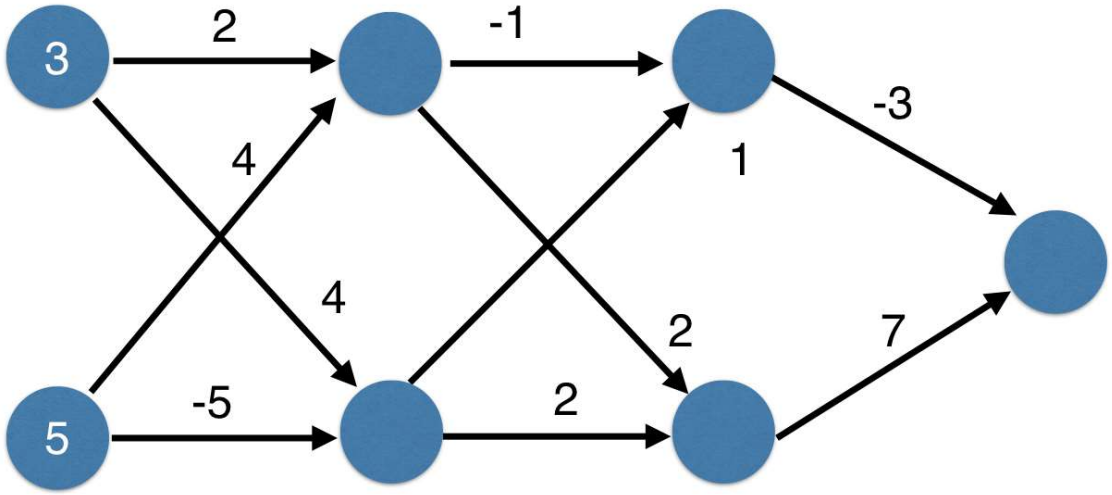

Calculate with ReLU Activation Function

# Multiple hidden layers



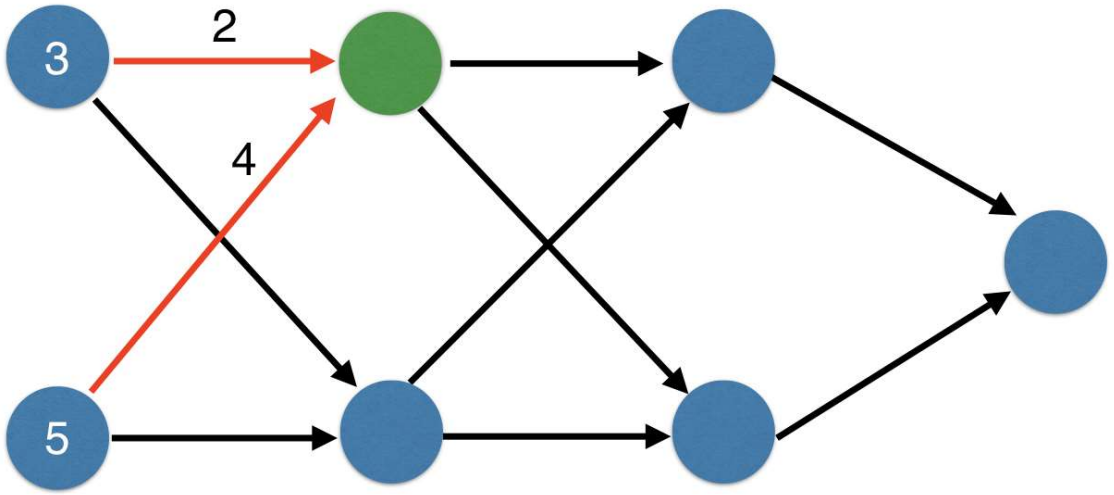

Calculate with ReLU Activation Function

# Multiple hidden layers



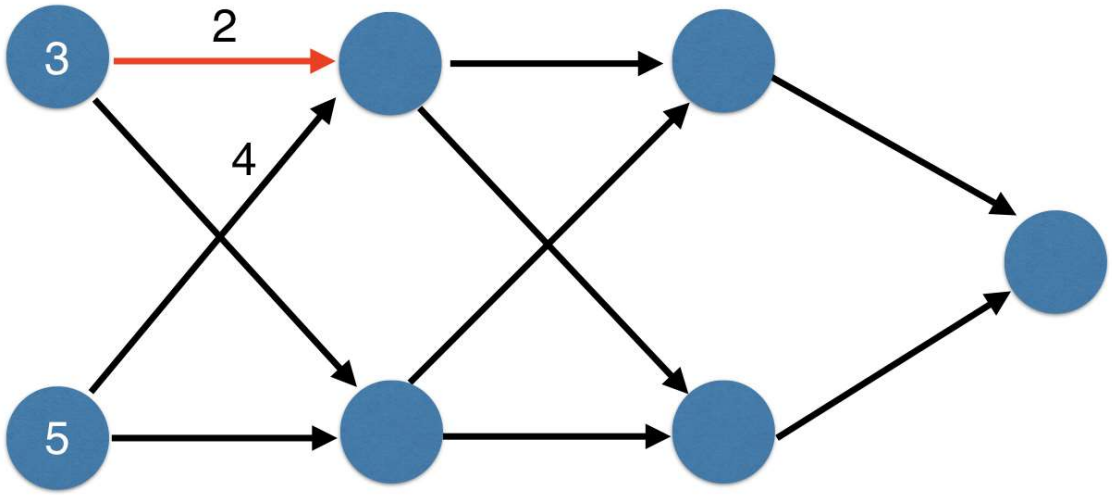

Calculate with ReLU Activation Function

# Multiple hidden layers



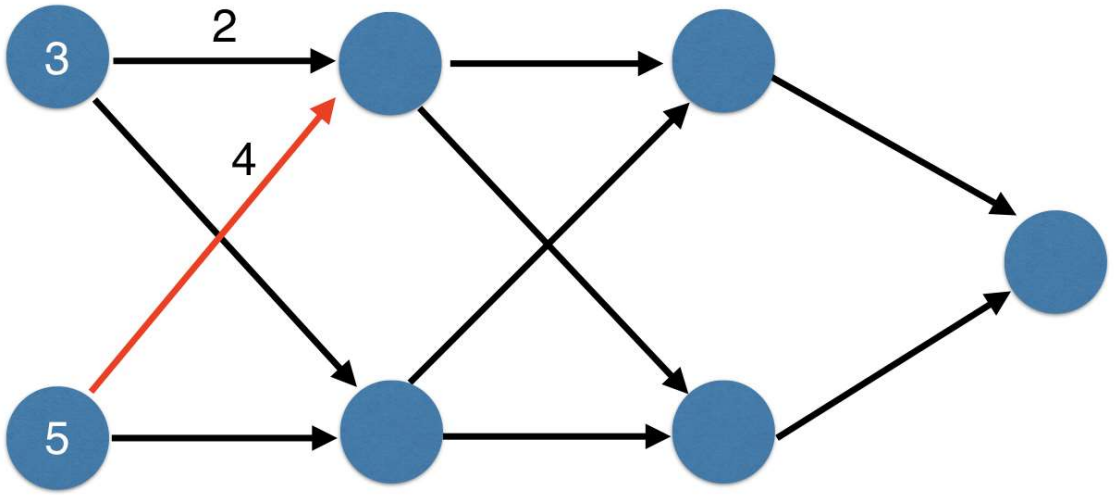

Calculate with ReLU Activation Function

# Multiple hidden layers



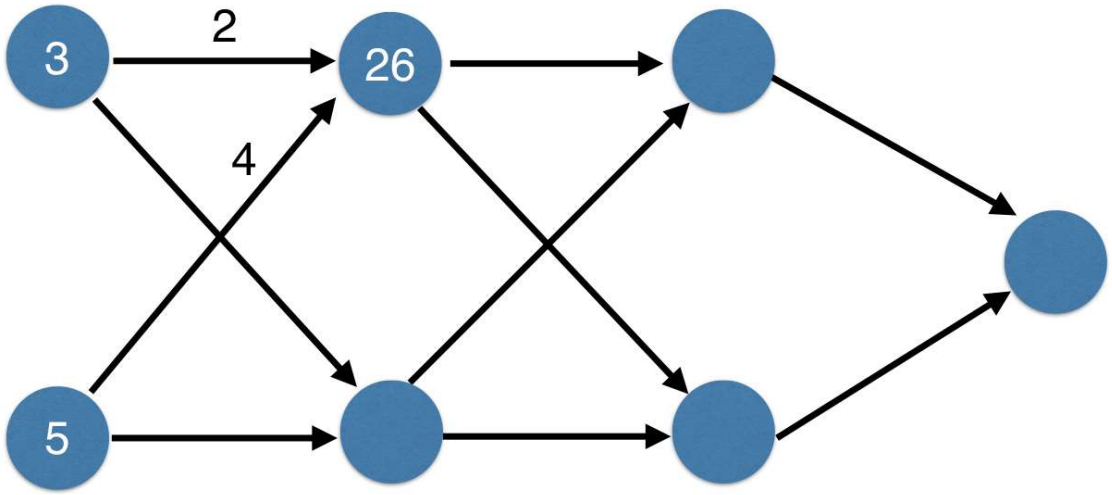

Calculate with ReLU Activation Function

# Multiple hidden layers



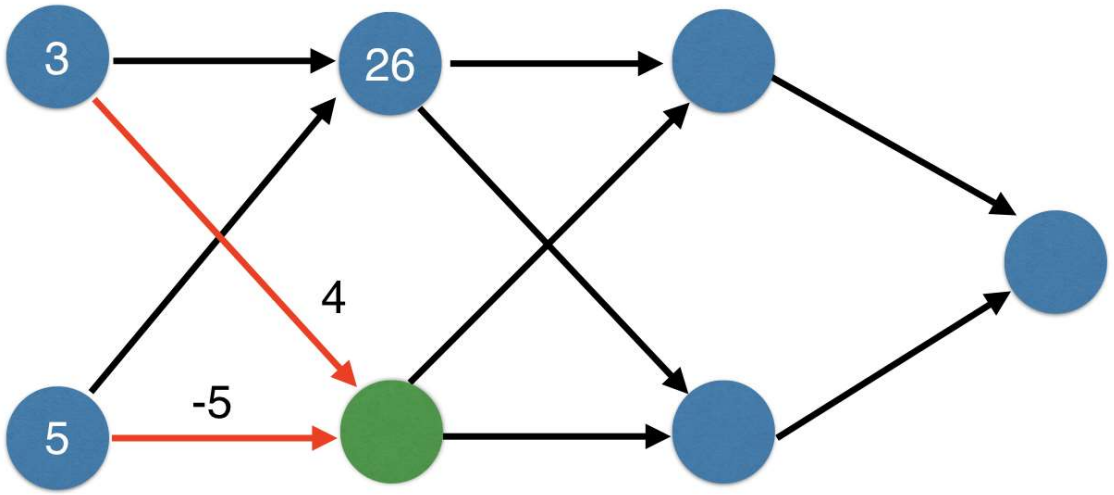

Calculate with ReLU Activation Function

# Multiple hidden layers



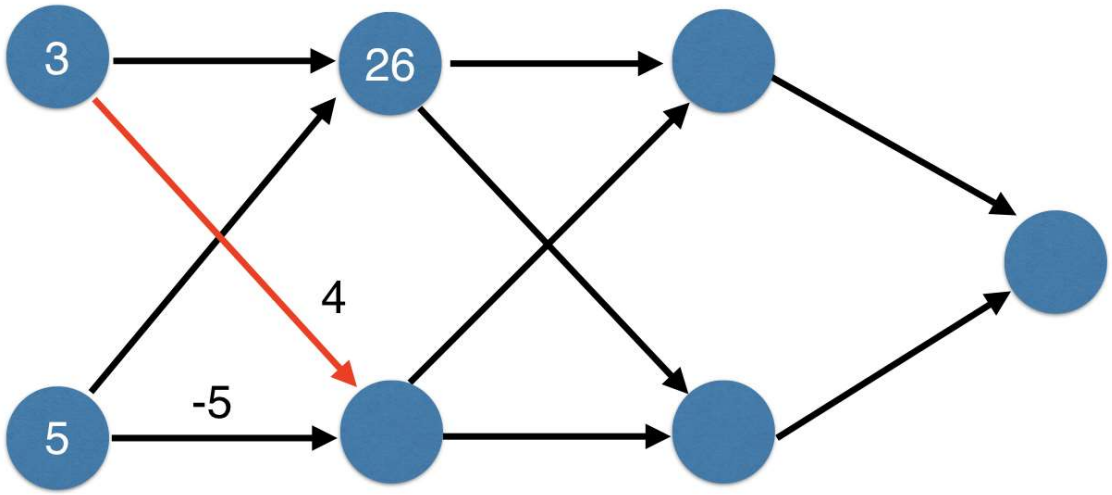

Calculate with ReLU Activation Function

# Multiple hidden layers



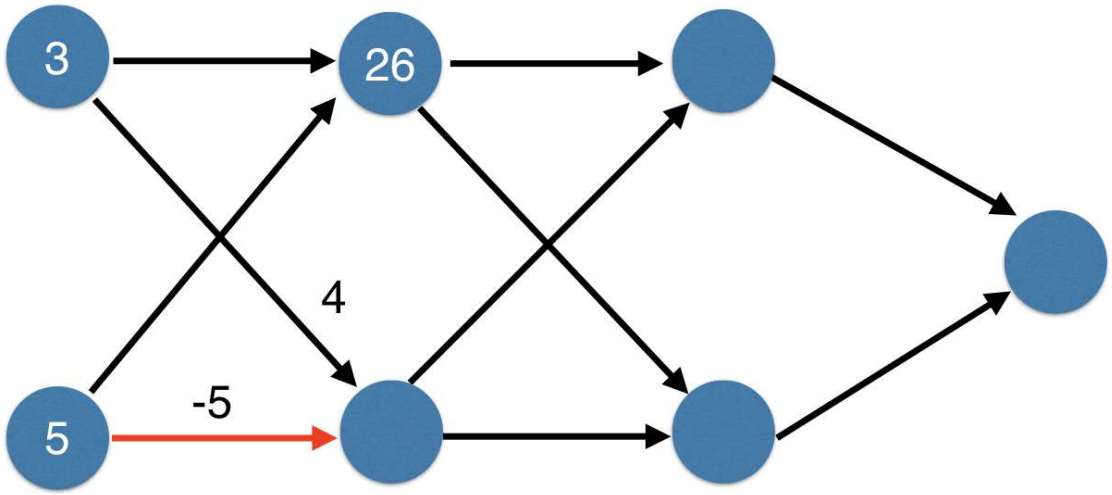

Calculate with ReLU Activation Function

# Multiple hidden layers



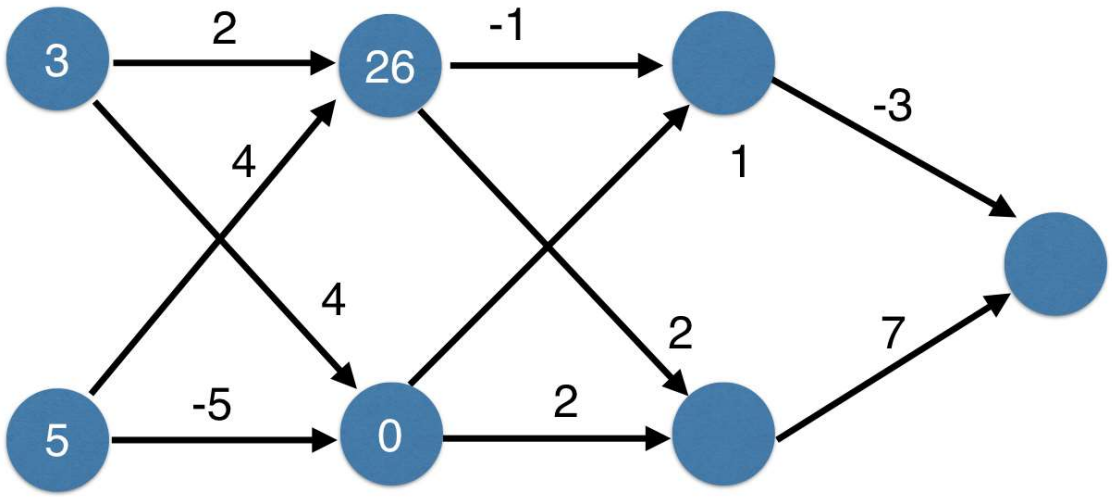

Calculate with ReLU Activation Function

# Multiple hidden layers



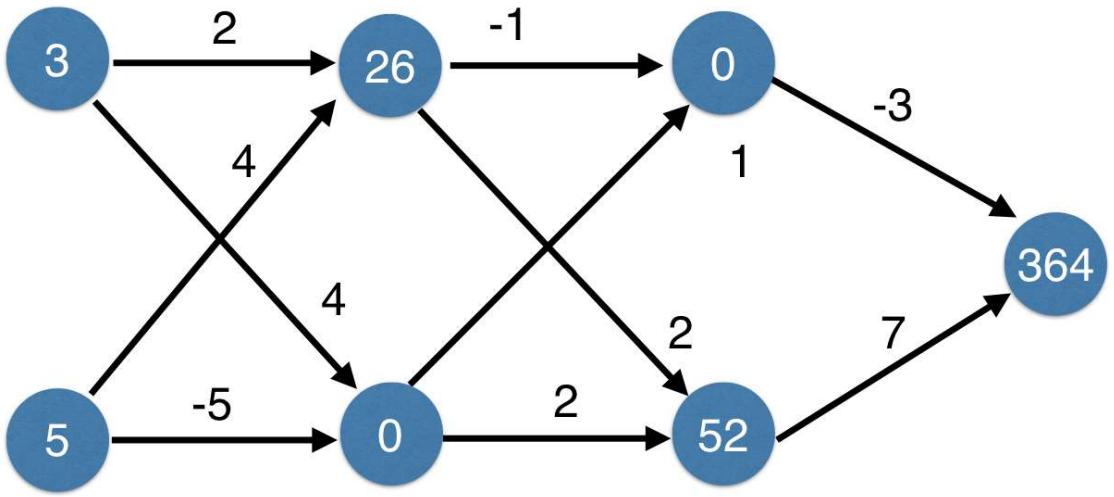

Calculate with ReLU Activation Function

# Multiple hidden layers




Calculate with ReLU Activation Function

# Multiple hidden layers

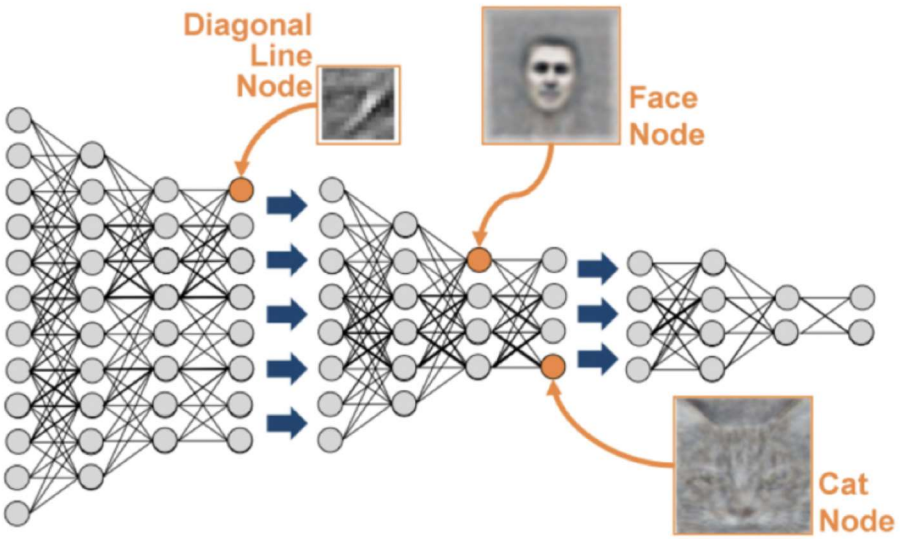



Calculate with ReLU Activation Function

# Representation learning

- Deep networks internally build representations of patterns in the data
- Partially replace the need for feature engineering
- Subsequent layers build increasingly sophisticated representations of raw data

# Representation learning



# Deep learning

- Modeler doesn't need to specify the interactions
- When you train the model, the neural network gets weights that find the relevant patterns to make better predictions

# Let's practice!

INTRODUCTION TO DEEP LEARNING IN PYTHON