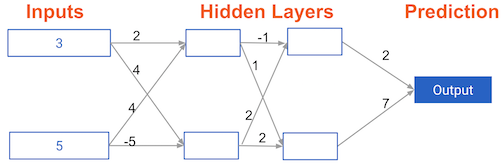
**Multi-layer neural networks**

In this exercise, you'll write code to do forward propagation for a neural network with 2 hidden layers. Each hidden layer has two nodes. The input data has been preloaded as input\_data. The nodes in the first hidden layer are called node\_0\_0 and node\_0\_1. Their weights are pre-loaded as weights['node\_0\_0'] and weights['node\_0\_1'] respectively.

The nodes in the second hidden layer are called node\_1\_0 and node\_1\_1. Their weights are pre-loaded as weights['node\_1\_0'] and weights['node\_1\_1'] respectively.

We then create a model output from the hidden nodes using weights pre-loaded as weights['output'].



* Calculate node\_0\_0\_input using its weights weights['node\_0\_0'] and the given input\_data. Then apply the relu() function to get node\_0\_0\_output.
* Do the same as above for node\_0\_1\_input to get node\_0\_1\_output.
* Calculate node\_1\_0\_input using its weights weights['node\_1\_0'] and the outputs from the first hidden layer - hidden\_0\_outputs. Then apply the relu() function to get node\_1\_0\_output.
* Do the same as above for node\_1\_1\_input to get node\_1\_1\_output.
* Calculate model\_output using its weights weights['output'] and the outputs from the second hidden layer hidden\_1\_outputs array. Do not apply the relu() function to this output.