**Defining a multiple input model**

In some cases, the sequential API will not be sufficiently flexible to accommodate your desired model architecture and you will need to use the functional API instead. If, for instance, you want to train two models with different architectures jointly, you will need to use the functional API to do this. In this exercise, we will see how to do this. We will also use the .summary()method to examine the joint model's architecture.

Note that keras has been imported from tensorflow for you. Additionally, the input layers of the first and second models have been defined as m1\_inputs and m2\_inputs, respectively. Note that the two models have the same architecture, but one of them uses a sigmoid activation in the first layer and the other uses a relu.

**Instructions**

**100 XP**

* Pass model 1's input layer to its first layer and model 1's first layer to its second layer.
* Pass model 2's input layer to its first layer and model 2's first layer to its second layer.
* Use the add() operation to combine the second layers of model 1 and model 2.
* Complete the functional model definition.

# For model 1, pass the input layer to layer 1 and layer 1 to layer 2

m1\_layer1 = keras.layers.Dense(12, activation='sigmoid')(m1\_inputs)

m1\_layer2 = keras.layers.Dense(4, activation='softmax')(m1\_layer1)

# For model 2, pass the input layer to layer 1 and layer 1 to layer 2

m2\_layer1 = keras.layers.Dense(12, activation='relu')(m2\_inputs)

m2\_layer2 = keras.layers.Dense(4, activation='softmax')(m2\_layer1)

# Merge model outputs and define a functional model

merged = keras.layers.add([m1\_layer2, m2\_layer2])

model = keras.Model(inputs=[m1\_inputs, m2\_inputs], outputs=merged)

# Print a model summary

print(model.summary())

Nice work! Notice that the .summary() method yields a new column: connected to. This column tells you how layers connect to each other within the network. We can see that dense\_2, for instance, is connected to the input\_2 layer. We can also see that the add layer, which merged the two models, connected to both dense\_1 and dense\_3.