**Compiling a sequential model**

In this exercise, you will work towards classifying letters from the Sign Language MNIST dataset; however, you will adopt a different network architecture than what you used in the previous exercise. There will be fewer layers, but more nodes. You will also apply dropout to prevent overfitting. Finally, you will compile the model to use the adam optimizer and the categorical\_crossentropy loss. You will also use a method in keras to summarize your model's architecture. Note that keras has been imported from tensorflow for you and a sequential keras model has been defined as model.

**Instructions**

**100 XP**

* In the first dense layer, set the number of nodes to 16, the activation to sigmoid, and the input\_shape to (784,).
* Apply dropout at a rate of 25% to the first layer's output.
* Set the output layer to be dense, have 4 nodes, and use a softmaxactivation function.
* Compile the model using an adam optimizer and categorical\_crossentropy loss function.

# Define the first dense layer

model.add(keras.layers.Dense(16, activation='sigmoid', input\_shape=(784,)))

# Apply dropout to the first layer's output

model.add(keras.layers.Dropout(0.25))

# Define the output layer

model.add(keras.layers.Dense(4, activation='softmax'))

# Compile the model

model.compile('adam', loss='categorical\_crossentropy')

# Print a model summary

print(model.summary())

Great work! You've now defined and compiled a neural network using the keras sequential model. Notice that printing the .summary() method shows the layer type, output shape, and number of parameters of each layer.