**Training with Keras**

In this exercise, we return to our sign language letter classification problem. We have 2000 images of four letters--A, B, C, and D--and we want to classify them with a high level of accuracy. We will complete all parts of the problem, including the model definition, compilation, and training.

Note that keras has been imported from tensorflow for you. Additionally, the features are available as sign\_language\_features and the targets are available as sign\_language\_labels.

**Instructions**

**100 XP**

* Define a sequential model named model.
* Set the output layer to be dense, have 4 nodes, and use a softmaxactivation function.
* Compile the model with the SGD optimizer and categorical\_crossentropy loss.
* Complete the fitting operation and set the number of epochs to 5.

# Define a sequential model

model = keras.Sequential()

# Define a hidden layer

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

# Define the output layer

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

# Compile the model

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

# Complete the fitting operation

model.fit(sign\_language\_features, sign\_language\_labels, epochs=5)

Great work! You probably noticed that your only measure of performance improvement was the value of the loss function in the training sample, which is not particularly informative. You will improve on this in the next exercise.