**Multiclass classification problems**

In this exercise, we expand beyond binary classification to cover multiclass problems. A multiclass problem has targets that can take on three or more values. In the credit card dataset, the education variable can take on 6 different values, each corresponding to a different level of education. We will use that as our target in this exercise and will also expand the feature set from 3 to 10 columns.

As in the previous problem, you will define an input layer, dense layers, and an output layer. You will also print the untrained model's predictions, which are probabilities assigned to the classes. The tensor of features has been loaded and is available as borrower\_features. Additionally, the constant(), float32, and keras.layers.Dense() operations are available.

**Instructions**

**100 XP**

* Define the input layer as a 32-bit constant tensor using borrower\_features.
* Set the first dense layer to have 10 output nodes and a sigmoid activation function.
* Set the second dense layer to have 8 output nodes and a rectified linear unit activation function.
* Set the output layer to have 6 output nodes and the appropriate activation function.

# Construct input layer from borrower features

inputs = constant(borrower\_features, float32)

# Define first dense layer

dense1 = keras.layers.Dense(10, activation='sigmoid')(inputs)

# Define second dense layer

dense2 = keras.layers.Dense(8, activation='relu')(dense1)

# Define output layer

outputs = keras.layers.Dense(6, activation='softmax')(dense2)

# Print first five predictions

print(outputs.numpy()[:5])

Great work! Notice that each row of outputs sums to one. This is because a row contains the predicted class probabilities for one example. As with the previous exercise, our predictions are not yet informative, since we are using an untrained model with randomly initialized parameters. This is why the model tends to assign similar probabilities to each class.