**Binary classification problems**

In this exercise, you will again make use of credit card data. The target variable, default, indicates whether a credit card holder defaults on his or her payment in the following period. Since there are only two options--default or not--this is a binary classification problem. While the dataset has many features, you will focus on just three: the size of the three latest credit card bills. Finally, you will compute predictions from your untrained network, outputs, and compare those the target variable, default.

The tensor of features has been loaded and is available as bill\_amounts. Additionally, the constant(), float32, and keras.layers.Dense()operations are available.

**Instructions**

**100 XP**

* Define inputs as a 32-bit floating point constant tensor using bill\_amounts.
* Set dense1 to be a dense layer with 3 output nodes and a reluactivation function.
* Set dense2 to be a dense layer with 2 output nodes and a reluactivation function.
* Set the output layer to be a dense layer with a single output node and a sigmoid activation function.

# Construct input layer from features

inputs = constant(bill\_amounts, float32)

# Define first dense layer

dense1 = keras.layers.Dense(3, activation='relu')(inputs)

# Define second dense layer

dense2 = keras.layers.Dense(2, activation='relu')(dense1)

# Define output layer

outputs = keras.layers.Dense(1, activation='sigmoid')(dense2)

# Print error for first five examples

error = default[:5] - outputs.numpy()[:5]

print(error)

Excellent work! If you run the code several times, you'll notice that the errors change each time. This is because you're using an untrained model with randomly initialized parameters. Furthermore, the errors fall on the interval between -1 and 1 because default is a binary variable that takes on values of 0 and 1 and outputs is a probability between 0 and 1.