**Initialization in TensorFlow**

A good initialization can reduce the amount of time needed to find the global minimum. In this exercise, we will initialize weights and biases for a neural network that will be used to predict credit card default decisions. To build intuition, we will use the low-level, linear algebraic approach, rather than making use of convenience functions and high-level keras operations. We will also expand the set of input features from 3 to 23. Several operations have been imported from tensorflow: Variable(), random(), and ones().

**Instructions**

**100 XP**

* Initialize the layer 1 weights, w1, as a Variable() with shape [23, 7], drawn from a normal distribution.
* Initialize the layer 1 bias using ones.
* Use a draw from the normal distribution to initialize w2 as a Variable()with shape [7, 1].
* Define b2 as a Variable() and set its initial value to 0.0.

# Define the layer 1 weights

w1 = Variable(random.normal([23, 7]))

# Initialize the layer 1 bias

b1 = Variable(ones([7]))

# Define the layer 2 weights

w2 = Variable(random.normal([7, 1]))

# Define the layer 2 bias

b2 = Variable(([0.0]))