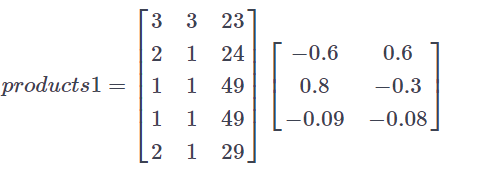
**The low-level approach with multiple examples**

In this exercise, we'll build further intuition for the low-level approach by constructing the first dense hidden layer for the case where we have multiple examples. We'll assume the model is trained and the first layer weights, weights1, and bias, bias1, are available. We'll then perform matrix multiplication of the borrower\_features tensor by the weights1variable. Recall that the borrower\_features tensor includes education, marital status, and age. Finally, we'll apply the sigmoid function to the elements of products1 + bias1, yielding dense1.



Note that matmul() and keras() have been imported from tensorflow.

##### Instructions

**100 XP**

* Compute products1 by matrix multiplying the features tensor by the weights.
* Use a sigmoid activation function to transform products1 + bias1.
* Print the shapes of borrower\_features, weights1, bias1, and dense1.

# Compute the product of borrower\_features and weights1

products1 = matmul(borrower\_features, weights1)

# Apply a sigmoid activation function to products1 + bias1

dense1 = keras.activations.sigmoid(products1 + bias1)

# Print the shapes of borrower\_features, weights1, bias1, and dense1

print('\n shape of borrower\_features: ', borrower\_features.shape)

print('\n shape of weights1: ', weights1.shape)

print('\n shape of bias1: ', bias1.shape)

print('\n shape of dense1: ', dense1.shape)

Good job! Note that our input data, borrower\_features, is 5x3 because it consists of 5 examples for 3 features. The shape of weights1 is 3x2, as it was in the previous exercise, since it does not depend on the number of examples. Additionally, bias1 is a scalar. Finally, dense1 is 5x2, which means that we can multiply it by the following set of weights, weights2, which we defined to be 2x1 in the previous exercise.