

**Program 2:** Build a simple artificial Neural Networks with 1 layer, with 1 neuron, and the input shape equal to 1, feed some data, use the equation  $y=5x-3$ , so where  $x = -2$ ,  $y=-4$  and train the network.

**Aim :-** The aim of this program is to build a simple artificial Neural Networks with 1 layer, with 1 neuron, and the input shape equal to 1, feed some data, use the equation  $y=5x-3$ , so where  $x = -2$ ,  $y=-4$  and train the network.

**Procedure :-**

```
import numpy as np
import tensorflow as tf

# Step 1: Prepare the Data
# Training data based on  $y = 5x - 3$ 
x_train = np.array([-2.0, -1.0, 0.0, 1.0, 2.0], dtype=np.float32) # Input values
y_train = np.array([-13.0, -8.0, -3.0, 2.0, 7.0], dtype=np.float32) # Corresponding outputs

# Step 2: Build the Model
model = tf.keras.Sequential([
    tf.keras.layers.Dense(units=1, input_shape=[1]) # Single layer, single neuron
])

# Step 3: Compile the Model
model.compile(optimizer='sgd', loss='mean_squared_error') # SGD optimizer and MSE loss

# Step 4: Train the Model
print("Training the model...")
model.fit(x_train, y_train, epochs=200, verbose=0) # Train for 200 epochs
print("Model training complete.")

# Step 5: Test the Model
x_test = np.array([-2.0], dtype=np.float32) # Test input
predicted_y = model.predict(x_test)
print(f"Predicted y for x={x_test[0]}: {predicted_y[0][0]}")

# Check the learned weights and bias
weights, bias = model.layers[0].get_weights()
print(f"Learned weight: {weights[0][0]}")
print(f"Learned bias: {bias[0]}")
```

**Output**  
Training the model...  
Model training complete.  
Predicted y for x=-2.0: -13.0001  
Learned weight: 4.9999  
Learned bias: -2.9998

