Copilot

Sure, here is the equivalent code in TensorFlow:

```
import tensorflow as tf

# Define the sizes
input_size_case1 = (1, 11088)
input_size_case2 = (16, 864)
r1_size = (32,)
r2_size = (128,)
U1_size_case1 = (1, 11088, 32)
U1_size_case2 = (16, 864, 32)
U2_size = (32, 128)

# Initialize the tensors
input_case1 = tf.random.normal(input_size_case1)  # Size: (1, 11088)
input_case2 = tf.random.normal(input_size_case2)  # Size: (16, 864)
r1 = tf.random.normal(r1_size)  # Size: (32,)
r2 = tf.random.normal(r2_size)  # Size: (128,)
U1_case1 = tf.random.normal(U1_size_case1)  # Size: (1, 11088, 32)
U1_case2 = tf.random.normal(U1_size_case2)  # Size: (16, 864, 32)
U2 = tf.random.normal(U2_size)  # Size: (32, 128)

# Compute the operations
U1T_case1 = tf.transpose(U1_case1, perm=[2, 1, 0])  # Size: (32, 11088, 1)
U1T_case2 = tf.transpose(U1_case2, perm=[2, 1, 0])  # Size: (32, 864, 16)
U1r1_case1 = tf.tensordot(U1_case1, r1, axes=([2], [0]))  # Size: (1, 11088)
U1r1_case2 = tf.tensordot(U1_case2, r1, axes=([2], [0]))  # Size: (16, 864)
U2r2 = tf.tensordot(U2, r2, axes=([1], [0]))  # Size: (32,)

# Perform batched matrix multiplication and sum along the batch dimension
temp_case1 = tf.matmul(U1T_case1, tf.expand_dims(input_case1 - U1r1_case1, axis=-1))  # Size: (32, 1, 1)
temp_case1 = tf.squeeze(tf.reduce_sum(temp_case1, axis=0))  # Size: (32,)
temp_case2 = tf.matmul(U1T_case2, tf.expand_dims(input_case2 - U1r1_case2, axis=-1))  # Size: (32, 16, 16)
temp_case2 = tf.squeeze(tf.reduce_sum(temp_case2, axis=0))  # Size: (32,)

# Now temp_case1 and temp_case2 are single vectors of size (32,)
# Update r1
r1 += temp_case1 + (U2r2 - r1)  # For case 1
r1 += temp_case2 + (U2r2 - r1)  # For case 2
```

This code does the same operations as the PyTorch code, but using TensorFlow functions. The `tf.transpose` function is used to permute the axes of a tensor, `tf.tensordot` is used for the tensor dot product, `tf.matmul` is used for matrix multiplication, `tf.reduce_sum` is used to sum along the batch dimension, and `tf.squeeze` is used to remove dimensions of size 1.

I hope this helps! If you have any more questions, feel free to ask. 😊