

Assignment 2

1 Tensorflow Softmax

1. Implement the softmax function using TensorFlow

Our approach is almost identical to the numpy implementation in Assignment 1. Here, `softmax_1d` handles the case for a 1-D vector, and `tf.map_fn` applies that function to each row of a tensor.

```
def softmax_1d(x):  
    x = tf.exp(x - tf.reduce_max(x))  
    s = tf.reduce_sum(x)  
    return x/s  
out = tf.map_fn(lambda _: softmax_1d(x), x)
```

2. Implement the cross-entropy loss using TensorFlow

Make sure to convert `y` into dtype `float`. `tf.multiply` is element-wise multiplication of tensors, so `multiply` followed by `reduce_sum` is the dot-product.

```
y = tf.to_float(y)  
out = -1*tf.reduce_sum(tf.multiply(y, tf.log(yhat)))
```

3. Explain the purpose of placeholder variables and feed dictionaries in TensorFlow computations

Placeholder variables are analogous to X in the equation $y = X\theta + b$. Like X , placeholder variables represent the input data to the algorithm. The feed dictionary substitutes actual values for the placeholder variables.