Thomas Whitcomb
thomaswhitcomb@yahoo.com
Jan 30th, 2019

Lesson 3

Code

```
import tensorflow as tf

def run(tensor):
    with tf.Session() as sess:
        return(sess.run(tensor))

# Y = X activation function
def y_equal_x(x):
    return x

def compute(inputs, weights, bias, activation_fn):
    layer = tf.matmul(inputs, weights)+bias
    return activation_fn(layer), layer

def problem1():
    inputs = tf.constant([[100, 150]])
    weights = tf.constant([[5, 10, 15, 20], [25, 30, 35, 40]])
    bias = tf.constant([[17, 19, 21, 23]])
    activation, l1 = compute(inputs, weights, bias, y_equal_x)

    weights = tf.constant([[30, 35], [40, 45], [70, 75], [80, 85]])
    bias = tf.constant([[35, 36]])
    _, l2 = compute(activation, weights, bias, y_equal_x)
    return run(l2)

def problem2():
    c = [([0., 0.], 0), ([1., 0.], 1), ([0., 1.], 1), ([1., 1.], 0)]
    inputs = tf.constant([a[0] for a in c])
    weights = tf.constant([[-4., -6., -5.], [3., 6., 4.]])
    bias = tf.constant([[-2., 3., -2.]])
    activation, _ = compute(inputs, weights, bias, tf.sigmoid)

    weights = tf.constant([[5.], [-9.], [7.]])
    bias = tf.constant([4.])
    activation, _ = compute(activation, weights, bias, tf.sigmoid)
    results = run(activation)
    table = []
    x = 0
    for row in c:
      r = [row[0][0], row[0][1], row[1], results[x][0],
(results[x][0]-row[1])**2]
      table.append(r)
      x = x + 1
    return table

def main():
    print(problem1())
    print(problem2())

if __name__ == "__main__":
    main()
```

Results

Run > python lesson3.py

```
[[1464615 1587516]]

[[0.0, 0.0, 0, 0.04137863, 0.0017121908934420033],
[1.0, 0.0, 1, 0.97319275, 0.0007186285770934831],
[0.0, 1.0, 1, 0.9920135, 6.378395795891834e-05],
[1.0, 1.0, 0, 0.017914694, 0.00032093625404074066]]
```