CS 20SI: TensorFlow for Deep Learning Lecture 3 summary

3/29/2017

outline

1. Assemble our graph (Linear Regression)
2. Huber loss
3. MNIST (LR)

Assemble our graph (Linear Regression)

Step 1: Read in data

Step 2: Create placeholders for inputs and labels

```
tf.placeholder(dtype, shape = None, name = None)
tf Queues (...)
```

Step 3: Create weight and bias

```
tf.Variable( initial_value=None, trainable=True, ...)
```

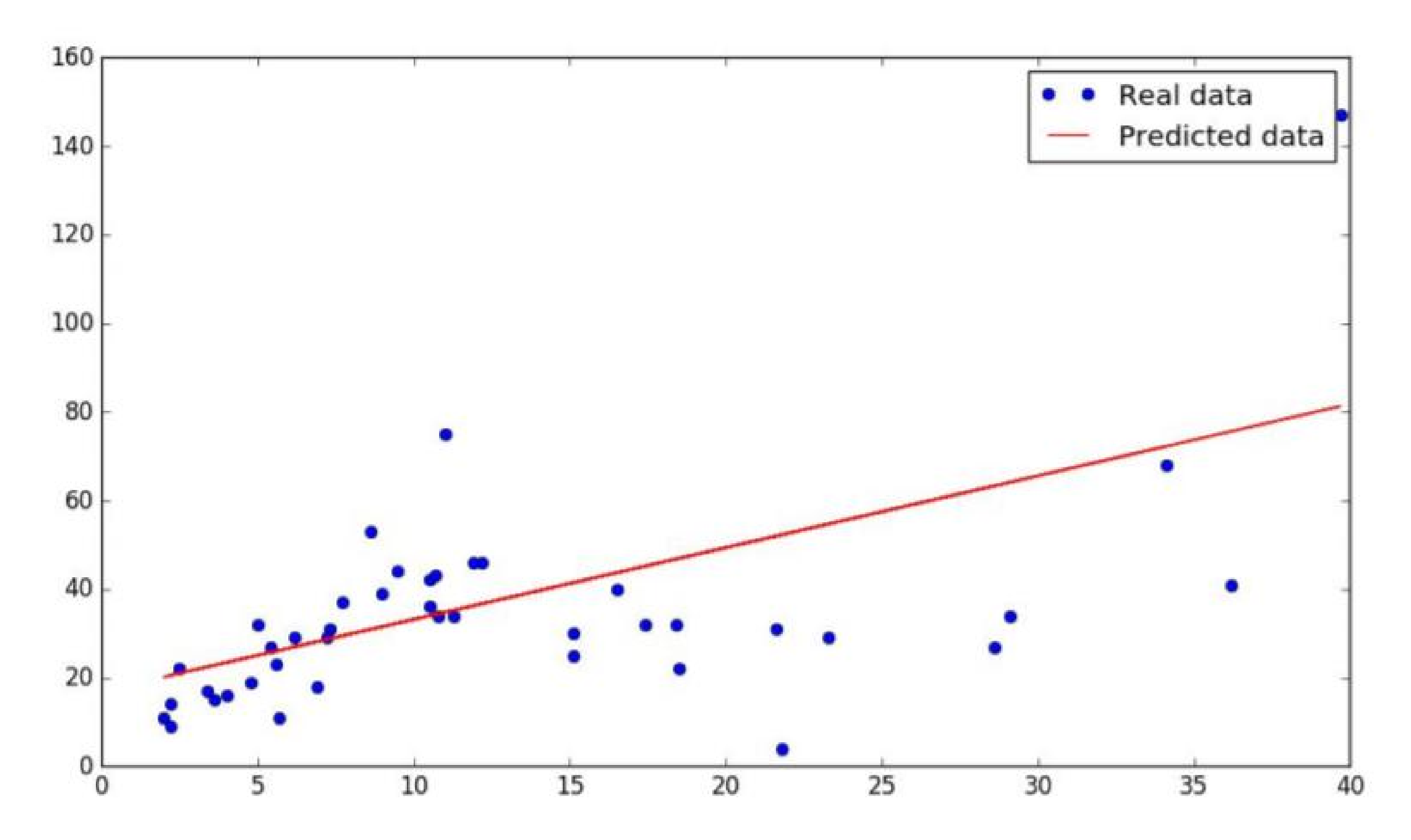
Step 4: Build model to predict Y

```
Y_predicted = X * w + b
```

Step 5: Specify loss function

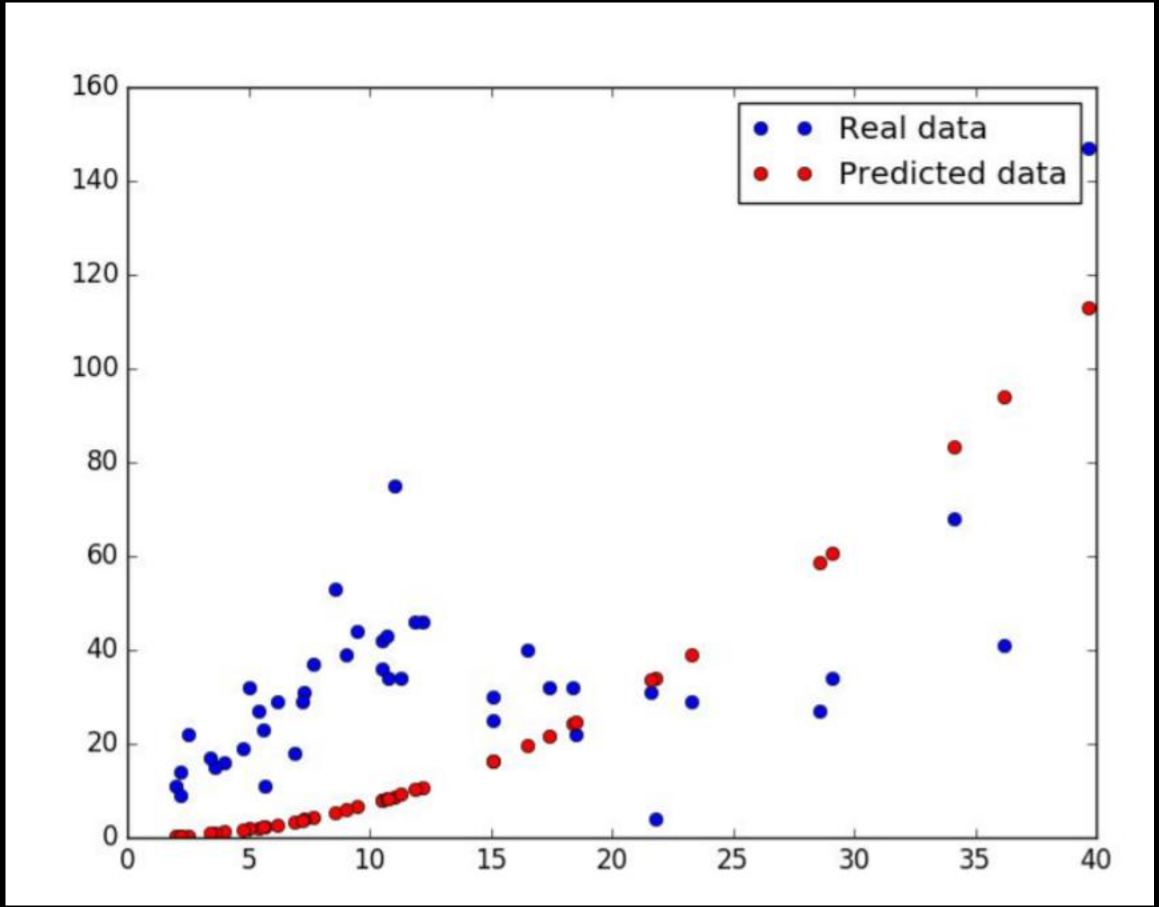
Step 6: Create optimizer

Assemble our graph (Linear Regression)



Y_predicted = X * w + b

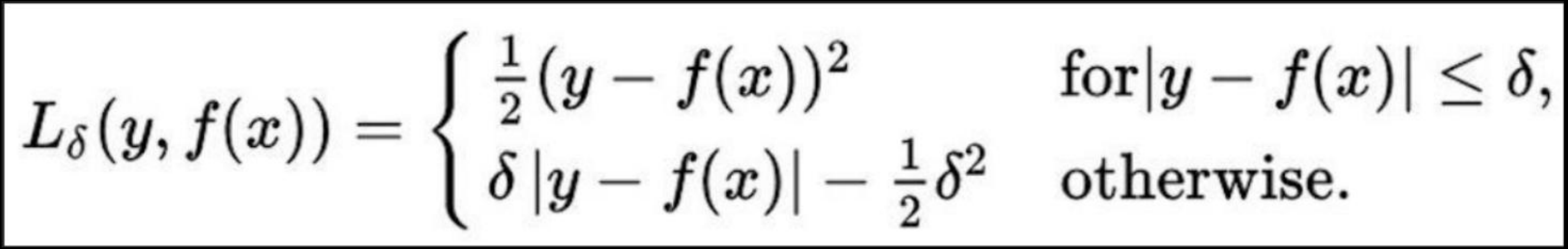
Assemble our graph (Linear Regression)



Y_predicted = X * X * w + X * u + b

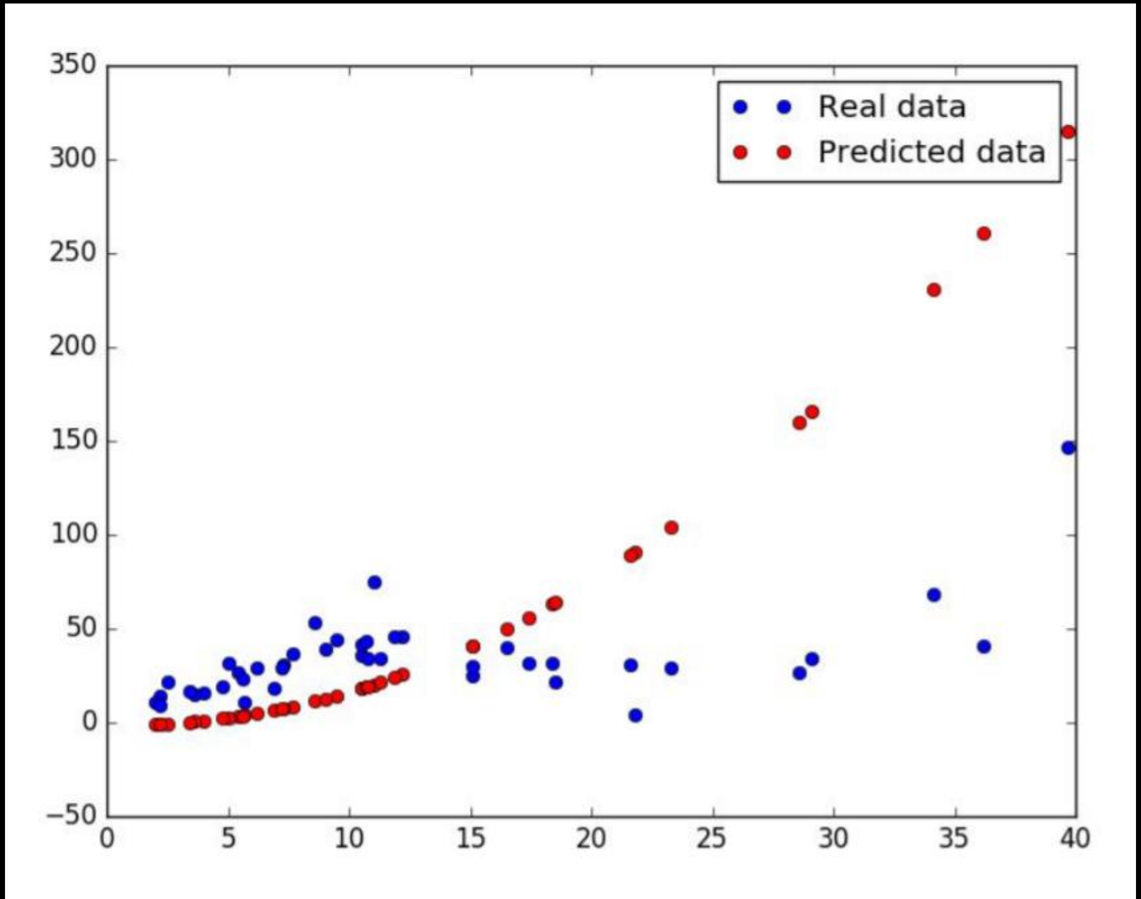
Huber loss

```
def huber_loss(labels, predictions, delta=1.0):
        residual = tf.abs(predictions - labels)
        condition = tf.less(residual, delta)
        small_res = 0.5 * tf.square(residual)
        large_res = delta * residual - 0.5 * tf.square(delta)
        return tf.where(condition, small_res, large_res)
        # delete tf.select in r1.0
```



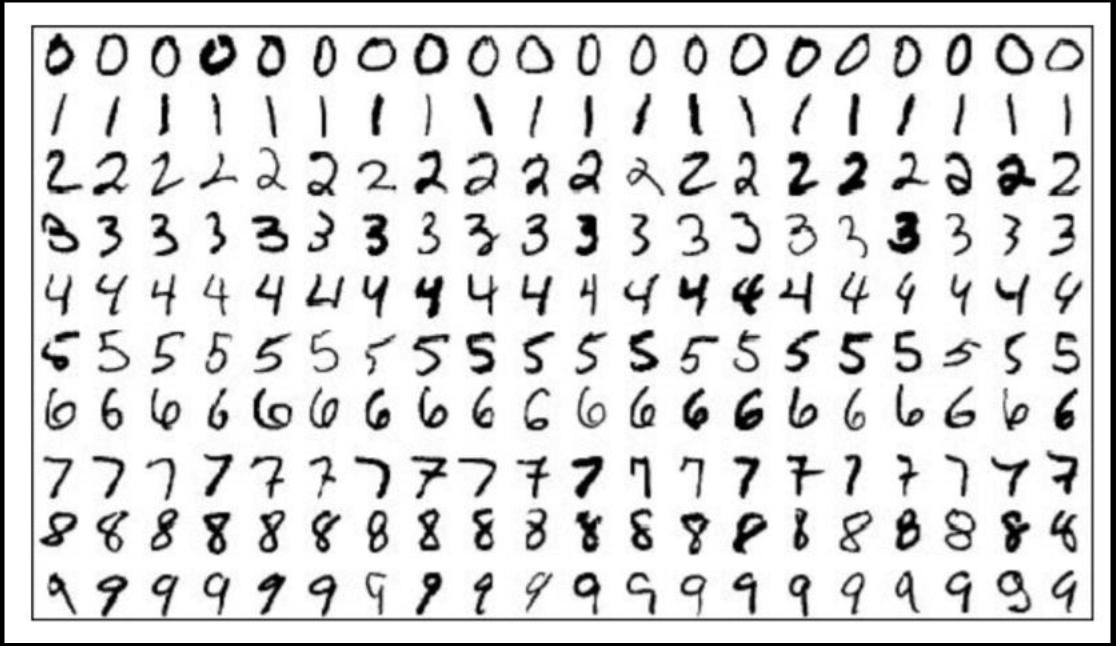
$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for} |y - f(x)| \le \delta, \\ \delta\,|y - f(x)| - \frac{1}{2}\delta^2 & \text{otherwise.} \end{cases}$$

Huber loss



350
300
250
200
150
100
50
0
−50
0
5
10
15
20
25
30
35
40
Real data
Predicted data

MNIST (LR)

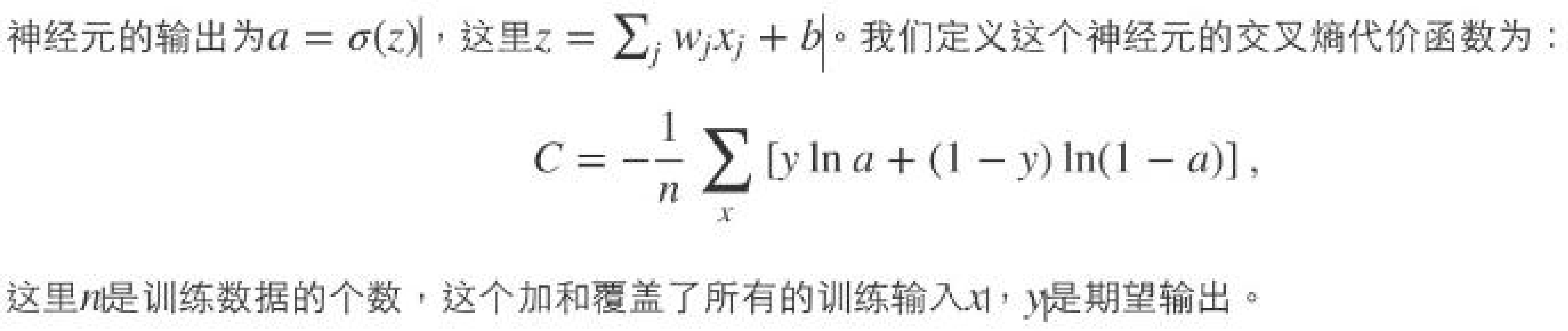


logits = X * w + b

Y_predicted = softmax(logits)

loss = cross_entropy(Y, Y_predicted)

MNIST (LR)



神经元的输出为$a = \sigma(z)$，这里$z = \sum_j w_j x_j + b$。我们定义这个神经元的交叉熵代价函数为：

$$C = -\frac{1}{n} \sum_x \left[y \ln a + (1 - y) \ln(1 - a) \right],$$

这里n是训练数据的个数，这个加和覆盖了所有的训练输入x，y是期望输出。

tf.nn.sigmoid_cross_entropy_with_logits
tf.nn.softmax_cross_entropy_with_logits
tf.nn.sparse_softmax_cross_entropy_with_logits
tf.nn.weighted_cross_entropy_with_logits

谢谢大家！