Logistic regression in TensorFlow

a) Open the notebook log_reg_challenger. In this notebook we use (binary) logistic regression to predict the probability for an O-ring to show a damage ($y = 1$) by using the temperature during take-off as predictor x.

$$p(y_i = 1|x_i) = \frac{e^{(b+W'x_i)}}{1 + e^{(b+W'x_i)}} = \frac{1}{1 + e^{-(b+W'x_i)}}$$

Determine the predicted $p(y_i = 1|x_i)$ values when using the parameters $W = -0.2$ and $b = 20$. What do you observe?

b) Now lets try to find better values for W and b. Lets assume W is given with -1. We want the probability for a damage $p(y_i = 1|x_i)$ to be 0.5. Determine an appropriate value for b.
Hint: at which x value should $p(y_i = 1|x_i)$ be 0.5, look at the data. At this x value the term $1 + e^{-(b+W'x_i)}$ must be 2.

c) Now we want to optimize the parameter values by using the gradient descent method. Run the TensorFlow forward pass in cell 5 and optimize the values for W and b in cell 6.
Fetch the loss, W and b and print the final values.
Hint: You can't use the same names for the results of your fetches as you have used for the TensorFlow graph. See cell 5.

Multinomial Logistic Regression on MNIST dataset

a) Open the notebook Multinomial Logistic Regression. In this notebook we use multinomial logistic regression to predict the handwritten digits of the MNIST dataset.
We have 4000 examples with 784 pixel values and 10 classes. Run the fist 3 cells and explain the one-hot-encoding. In TensorFlow we need to use one-hot-encoding.

b) Write the missing TensorFlow code in cell 4 to do the required matrix multiplication between x and w and then add the bias b.
$z = x * w + b$

c) Run the next two cells to store the graph and do a forward pass of the untrained network, look at the probability for each class of some examples.

d) Now lets train the model. We use a mini-batch size of 128 and use the fist 2400 examples for training.
The validation set will be the examples from 2400 to 3000. Write the code to get the loss and the probabilities of your validation set. Run the last cells to check the performance of the model and to get the probability of a random example of the validation set.

e) Additional Questions:

- How many parameters do we have?
- Compare the loss of the validation set against the loss of the training set.
  Why do you think that there is such a great difference.
- Do you understand how the loss is calculated `loss = tf.reduce_mean(-tf.reduce_sum(y_true * tf.log(prob), reduction_indices=[1]))`