# CMPS 142 HW 4

Isaiah Solomon

March 2017

## 1 Question 1

### a)

Tensors are the data structures that Tensorflow provides that vastly help in the process of trying to program machine learning algorithms as they work together with the various functions provided. In order to do a simple linear regression problem with Tensorflow, you would use Variables and Constants. Constants are integers, floats, arrays or any other unit. Variables and placeholders are units that change value later on. First, you would create your model using a mixture of Variables and Constants, then minimize the loss using built in Gradient Descent functions. Then, using training data, we will fit the linear model and the test the model with our test data.

## 2 Question 2

The loss values that I acquired were 0.014 for the training set and 0.013 for the test set. The learned function was obtained using the numpy_input_fn method and applying the FtrlOptimizer.

```
Isaiahs-MacBook-Air:MachineLearningProj zayz$ docker run test
WARNING:tensorflow:Using temporary folder as model directory: /tmp/tmpVJJ4A2
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/estimators/head.py:1362: scalar_summary (from tensorflow.python.ops.logging_ops) is deprecated and will
 be removed after 2016-11-30.
Instructions for updating:
Please switch to tf.summary.scalar. Note that tf.summary.scalar uses the node name instead of the tag. This means that TensorFlow will automatically de-duplicate summary names based on the scope they are
created in. Also, passing a tensor or list of tags to a scalar summary op is no longer supported.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE3 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.1 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use SSE4.2 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use AVX2 instructions, but these are available on your machine and could speed up CPU computations.
W tensorflow/core/platform/cpu_feature_guard.cc:45] The TensorFlow library wasn't compiled to use FMA instructions, but these are available on your machine and could speed up CPU computations.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/estimators/head.py:1362: scalar_summary (from tensorflow.python.ops.logging_ops) is deprecated and will
 be removed after 2016-11-30.
Instructions for updating:
Please switch to tf.summary.scalar. Note that tf.summary.scalar uses the node name instead of the tag. This means that TensorFlow will automatically de-duplicate summary names based on the scope they are
created in. Also, passing a tensor or list of tags to a scalar summary op is no longer supported.
WARNING:tensorflow:Skipping summary for global_step, must be a float or np.float32.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/contrib/learn/python/learn/estimators/head.py:1362: scalar_summary (from tensorflow.python.ops.logging_ops) is deprecated and will
 be removed after 2016-11-30.
Instructions for updating:
Please switch to tf.summary.scalar. Note that tf.summary.scalar uses the node name instead of the tag. This means that TensorFlow will automatically de-duplicate summary names based on the scope they are
created in. Also, passing a tensor or list of tags to a scalar summary op is no longer supported.
WARNING:tensorflow:Skipping summary for global_step, must be a float or np.float32.
Training: {'loss': 0.014003001, 'global_step': 1000}
Test: {'loss': 0.012942068, 'global_step': 1000}
```

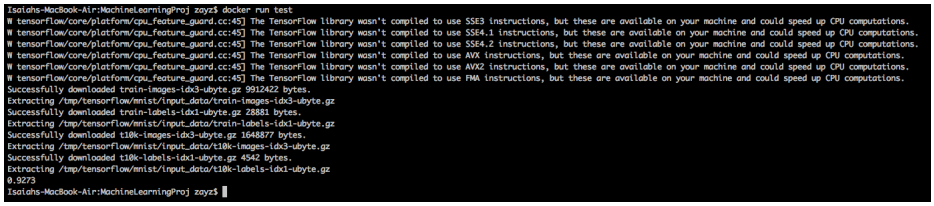
# 3 Question 3

## a)

MNIST provides a visual data set for machine learning purposes. In this section, they use a data set of handwritten numbers in order to visualize what number the image is portraying. In order to do this, we use a softmax regression model in order to efficiently assign probability to each number we are guessing. Along with weights, we also assign bias to the model. We then implement regression, using a gradient descent optimizer as well as calculating the loss.

## b)

For the file mnist_softmax.py, I changed the optimizer from GradientDescentOptimizer to AdamOptimizer and adjusted the step size to 1e-4. I also changed the number of steps to 50000. This changed the accuracy to a consistent 0.927.




# 4 Question 4

## a)

The mechanics section lists the specifics of the functions we will use to train the graph, test, etc. This section basically sums up the tutorial pages that gave specific functions to teach how to use Tensorflow, whereas this page now lists the functions that can be used for general purpose outside of the tutorial. It gives the files that we will be using for this problem, as well as providing links to the image training data.