Deep Learning
Assignment 1
7 points

Before commencing this Assignment, you should install Tensorflow, by following the instructions at tensorflow.org

This assignment will build on the software found at
https://www.tensorflow.org/get_started/mnist/beginners

The necessary code can be downloaded from here
https://github.com/tensorflow/tensorflow/blob/r1.3/tensorflow/examples/tutorials/mnist/mnist_softmax.py

1. Linear Softmax

Run the code for linear softmax, by typing:

```
python mnist_softmax.py
```

The first time it is run, this should automatically download the mnist dataset.
If necessary, you can also download it directly from here:
http://www.iro.umontreal.ca/~lisa/deep/data/mnist/mnist.pkl.gz

When run, the last lines of the output should report the accuracy, to four decimal places.

Run the code a few times. You should get slightly different values each time, but approximately 0.92

a. [2 marks]

It is useful to know which items are being misclassified, and in what way.
You will be doing this using a Confusion Matrix.

Look for this line in mnist_softmax.py:

```
if __name__ == '__main__':
```

Immediately before this line, insert the following code:

```
  true_class=tf.argmax(y, 1)
  predicted_class=tf.argmax(y_, 1)
  cm=tf.confusion_matrix(predicted_class,true_class)
  print(sess.run(cm, feed_dict={x: mnist.test.images,
                               y_: mnist.test.labels}))
```

Now re-run the code by again typing:

```
python mnist_softmax.py
```

Copy the resulting confusion matrix into your answers.

Take a look at the confusion matrix. The rows, indexed 0 to 9, indicate the correct output (target) while the columns indicate the digit predicted by the model.
If we use the notation 2 -> 8 to mean that a 2 is often falsely predicted as an 8, what other common misclassifications do you notice? (List all of them with >= 20 occurrences).
Briefly explain (in one or two sentences) why you think these particular misclassifications are the most common.

Although it is common for a "2" to be misclassified as an "8", it is quite rare for an "8" to be misclassified as a "2". Briefly explain (in one or two sentences) why you think this would be the case?

b. [1 mark]

Given that there are 28 x 28 = 784 inputs and 10 outputs, what is the total number of weights (trainable parameters) in the linear softmax model?
(Hint: Remember to include the "bias" weights)

2. Two-Layer Neural Network

We will now modify the code so it uses a 2-layer neural network to classify the same MNIST data.

Copy the file mnist_softmax.py to a new file called mnist_mlp.py, and open this file for editing.

Replace these lines:

```
  # Create the model
  x = tf.placeholder(tf.float32, [None, 784])
  W = tf.Variable(tf.zeros([784, 10]))
  b = tf.Variable(tf.zeros([10]))
  y = tf.matmul(x, W) + b
```

with this:

```
  # Create the model
  n_hidden=100
  x = tf.placeholder(tf.float32, [None, 784])
  U = tf.Variable(tf.random_normal(shape=[784,n_hidden], stddev=0.01))
  b = tf.Variable(tf.zeros([n_hidden]))
  h = tf.nn.tanh(tf.matmul(x, U) + b)
  W = tf.Variable(tf.zeros([n_hidden, 10]))
  c = tf.Variable(tf.zeros([10]))
  y = tf.matmul(h, W) + c
```

Note that the input-to-hidden weights must be initialized to non-zero values (this is discussed in Section 8.4 of the textbook).

We will also need to train the network for longer, so change this line:

```
   for _ in range(1000):
```

to this:

```
   for _ in range(20000):
```

You are now ready to run the multi-layer perceptron, by typing:

```
python mnist_mlp.py
```

a. [1 mark]

Copy the last lines of the output (test performance and confusion matrix) into your answers.

Look at the new confusion matrix. You will see that misclassifications are in general much less common than for the linear softmax model, but a few still remain.
Write down the five misclassifications with the highest number of occurrences.
Do you think these are the kinds of errors that humans might also make (if the digits were poorly written)?

Hint: take a look at these images from Chapter 6 of Michael Nielsen's book Neural Networks and Deep Learning:
http://neuralnetworksanddeeplearning.com/chap6.html

b. [1 mark]

Assuming an MLP with 100 hidden nodes, calculate the total number of weights in (i) the hidden layer and (ii) the output layer, of the MLP model.
(Hint: Remember to include bias weights in both layers)

c. [1 mark]

We will now test the effect of varying the number of hidden nodes, by changing the line where n_hidden is defined.

We have already tried n_hidden=100. You should now try n_hidden=50 and n_hidden=20.
Run each case three times, and construct a table showing the range (lowest and highest values) of the final test accuracy, for each value of n_hidden. If you are using a GPU (and the code runs fast enough), you should also test n_hidden = 200 and 500, and include them in your table.

3. [1 mark]

Write the equations for each of these activation functions:
sigmoid, tanh, ReLU