Due date: **Dec. 9, 2018, 11:59 PM** (Arlington time). You have **two** late days throughout the semester — use it at as you wish. Once you run out of this quota, the penalty for late submission will be applied. You can either use your late days quota (or let the penalty be applied). **Clearly indicate** in your submission if you seek to use the quota.

What to turn in:

1. Your submission should include your complete code base in an archive file (`zip, tar.gz`) and `q1/, q2/,` and so on), and a very very clear README describing how to run it.
2. A brief report (typed up, submit as a PDF file, NO handwritten scanned copies) describing what you solved and implemented and known failure cases. The report is **important** since we will be evaluating the grades mostly based on the report.
3. Submit your entire code and report to Blackboard.

Notes from instructor:

- You may ask the TA or instructor for suggestions, and discuss the problem with others (minimally). But **all parts of the submitted code must be your own**.
- Use Python for your implementation.
- Make sure that the TA can easily run the code by plugging in our test data.

Problem 1

(Tensorflow and Keras, **50pts**) Try out the tutorial for Deep Learning using Tensorflow at `https://www.tensorflow.org/tutorials`. It has the following lines of code.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist

(x_train, y_train),(x_test, y_test) = mnist.load_data()
x_train, x_test = x_train / 255.0, x_test / 255.0

model = tf.keras.models.Sequential([
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(512, activation=tf.nn.relu),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.Dense(10, activation=tf.nn.softmax)
])

model.compile(optimizer='adam',loss='sparse_categorical_crossentropy', metrics=['accuracy'])

model.fit(x_train, y_train, epochs=5)
model.evaluate(x_test, y_test)
```

1. (**20pt**) In the report, write comments for each line of code given above and explain what this framework is doing.
2. (**20pt**) Change the number of hidden nodes to 10 and train this neural network. The trained model contains the weights that it has learned from training. Plot the features in the hidden layer that it has learned from training and include them in the report. That is, reshape the learned weights (vectors) in

the first layer (between the input and the 1st hidden layer) to the image dimension (in 2D) and show them. (You will get 0 marks for this if the result is not included in the report.)

3. (**10pt**) Change the number of hidden nodes to 1, 10, 50 and 100 and report how the testing accuracy changes for the testing dataset. Report the result and your observation in the report.