CSC 578 Quiz #1

Create a/one **.docx** or **.pdf** file containing all of your answers. The file must start with **your name**, **the course and section number in which you are registered** and the **assignment name/number** (e.g. "CSC 578 Quiz #1"). Files without those information will be returned ungraded.

Do all questions.

1. Mitchell's book, **Exercise 4.1**.
   Comments:
   - "Figure 4.3" means Figure 4.3 (a), which has the decision line indicated.
   - THERE ARE MANY ANSWERS because you'll have 2 equations to solve 3 variables. Pick any one that works. (All lines/equations are invariant to scaling of parameters).
   - As a hint, w0 must be < 0 because the origin should be classified as negative (-) as shown in the graph.
2. Mitchell's book, **Exercise 4.2**.
   Comments:
   - Translate "a perceptron" to be a single perceptron with the step function (where the output is either 1 or -1 (not 0)).
   - A and notB should be just one perceptron, while A xor B should be a network (i.e., multilayer) of perceptrons.
   - In both perceptrons, you MUST include a bias/threshold unit. Also assume its value is 1 (not 0 or -1)).
   - In your answer, you draw the architecture (a perceptron or a network of perceptrons), and clearly indicate the weight value for each connection.
   - Your perceptron(s) must be able to classify all instances (i..e, all possible combinations of the values of the input variables and the output values) correctly.

9/10 Update after class

The following question is postponed till the next quiz. It's NOT a part of Quiz #1.

3. An exercise in the NNDL book, chapter 1 (NNDL1): "Try creating a network with just two layers - **an input and an output layer, no hidden layer - with 784 and 10 neurons**, respectively. Train the network using stochastic gradient descent. What classification accuracy can you achieve?

   The book code, along with a couple of files modified for this course, is available at this **Git code repository**. Visit there and download the repository in a zip file on your local computer. Or you can also use the **IBM Cognitive Class Labs**. This is a free development site, equipped with various Machine Learning tools including **Python 3, Jupyter Notebook and Tensorflow**. Also this video provides an easy intro to CCLabs (although the video is made for a different course at a different institution).

   This question is implying the use of MNIST dataset. You can look at the example code in the book to load the data, create a network and train the network with the training_data and evaluate with the test_data. The code in the book to load the MNIST data and create training/validation/test datasets is

```
training_data, validation_data, test_data = mnist_loader.load_data_wrapper()
```

And a sample code to train a network with training and test data is

```
net.SGD(training_data, 30, 10, 3.0, test_data = test_data)
```

Read the book chapter to figure out other lines of code you need. Play with the parameters and run the network several times to get your personal best accuracy. Requirements:

- Do NOT run over 100 epochs for each run.
- Describe your experiment IN DETAIL and write your observations (such as the behavior of learning).
- Discuss results and your reflection.   Do you think your best result was a 'good result'? Why or why not? Also, how about comparing with the results by networks with hidden layers (for instance, the results of the network [784, 30, 10] described in the book chapter)? Can you think of any advantage for networks with no hidden layer?
- Write as much as you can. It is my policy that I consider terse answers insufficient, therefore won't give a full credit.