CMPUT 366, Winter 2022
Assignment #3

Due: Friday, March 25, 2022, 11:59pm
Total points: 68

For this assignment use the following consultation model:

1. you can discuss assignment questions and exchange ideas with other *current* CMPUT 366 students;
2. you must list all members of the discussion in your solution;
3. you may **not** share/exchange/discuss written material and/or code;
4. you must write up your solutions individually;
5. you must fully understand and be able to explain your solution in any amount of detail as requested by the instructor and/or the TAs.

Anything that you use in your work and that is not your own creation must be properly cited by listing the original source. Failing to cite others' work is plagiarism and will be dealt with as an academic offence.

First name: Pong

Last name: Cheng

CCID: pcheng1 `@ualberta.ca`

Collaborators: _____

1. **(Neural Networks)** The MNIST dataset is a set of images of handwritten digits labelled by their actual digit. We will operate on two versions of this dataset: one with the images shifted 2 pixels to the upper-left, and one with the images shifted 2 pixels to the bottom-right.

   This question requires the use of TensorFlow. The example code was tested using Tensorflow version 2.8.0 and Python version 3.9. You may need to install TensorFlow using the following command:

   ```
   pip3 install tensorflow
   ```

   (a) **[10 points]** Implement a fully-connected feed-forward neural network for classifying MNIST images according to the digit that they represent by editing the `mlp2` function in the provided `cnn.py` file.

   The network should have two hidden layers: one with 128 rectified linear ('relu') units, and one with 64 rectified linear units. The output should be a fully-connected layer of 10 units with the softmax activation. The `cnn.py` file contains an example implementation of a network with a single hidden layer in the `mlp1` function that you may template from. The `main` function will test your program for you; you may add any tests that you like. It will also create a file called `examples.png` that contains examples images from the two test sets.

   The function should train the network using the training features `train_x` and labels `train_y`, and then evaluate the accuracy of the trained network on two different test sets: `test1_x`, `test1_y`, and `test2_x`, `test2_y`. This is demonstrated in `mlp1`.

   We will run your code by importing the `cnn` module and calling `mlp2`, so it is important that your code follow these naming conventions.

   Submit all of your code for this question and question 1 (including provided boilerplate files) in a single zip file.

   (b) **[30 points]** Implement a convolutional neural network for classifying MNIST images by editing the `cnn` function in the provided `cnn.py` file.

   The network should have the following architecture:

   - A layer of 32 convolutional units with a kernel size of 5×5 and a stride of $1, 1$.
   - A max-pooling layer with a pool size of 2×2 and a stride of $2, 2$.
   - A layer of 64 convolutional units with a kernel size of 5×5 and the default stride.
   - A max-pooling layer with a pool size of 2×2 and the default stride.
   - A `Flatten` layer (to reshape the image from a 2D matrix into a single long vector)
   - A layer of 512 fully-connected relu units
   - A layer of 10 fully-connected softmax units (the output layer)

   Submit all of your code for this question and question 1 (including provided boilerplate files) in a single zip file.

   (c) **[2 points]** What was the accuracy of your trained 2-hidden-layer feedforward network on the two test sets?

   test set 1 : 0.9761

   test set 2 : 0.5567

   (d) **[2 points]** What was the accuracy of your trained convolutional neural network on the two test sets?

   test set 1 : 0.9875

   test set 2 : 0.8830

   (e) **[10 points]** Did one of your implementations perform substantially better on one of the test sets than the other implementation did? If so, why? If not, why not?

   CNN is better than 2-hidden-layer ~~feedforward~~ network
   Since CNN is using two new operations which are convolutions and pooling.
   Also it is effecient learning via sparse interactions, parameter sharing, Equivariant representations.

2. **(Bayesian Learning)**

   Suppose that you have three models $(\theta_1, \theta_2, \theta_3)$ of changes in the price of a single stock. You know that one of these models is the true model. Each model gives the probability that tomorrow's price will be higher than today's price (y_{t+1}), based on whether today's price was higher than the price the day before (y_t). So you can make money on average by buying the stock when $p(y_{t+1} \mid y_t, \theta^*) > .5$ and selling when $p(y_{t+1} \mid y_t, \theta^*) < .5$, where θ^* is the true model.

   Your prior belief is that θ_1 and θ_2 are equally likely to be true, and θ_3 is three times more likely than θ_1 to be true.

   You have a dataset D of past observations, and you have computed that

$$p(D \mid \theta_1) = .00084$$
$$p(D \mid \theta_2) = .00105$$
$$p(D \mid \theta_3) = .00007.$$

3. **[6 points]** What are the posterior probabilities of each model being the true model?

$\theta_1 = \theta_2 = 0.2 \qquad \theta_3 = 0.6$

$$Pr(D) = P(D|\theta_1)P(\theta_1) + P(D|\theta_2)P(\theta_2) + P(D|\theta_3)P(\theta_3) = 0.00042$$

$$Pr(\theta_1|D) = \frac{Pr(D|\theta_1)Pr(\theta_1)}{Pr(D)} = \frac{0.00084 \times 0.2}{0.00042} = 0.4$$

$$Pr(\theta_2|D) = \frac{Pr(D|\theta_2)Pr(\theta_2)}{Pr(D)} = \frac{0.00105 \times 0.2}{0.00042} = 0.5$$

$$Pr(\theta_3|D) = \frac{Pr(D|\theta_3)Pr(\theta_3)}{Pr(D)} = \frac{0.00007 \times 0.6}{0.00042} = 0.1$$

4. **[4 points]** Now suppose that you have run each model, and they make the following predictions:

$$p(y_{t+1} \mid y_t, \theta_1) = .75 \qquad P(\theta_1|D) = 0.4$$
$$p(y_{t+1} \mid y_t, \theta_2) = .4 \qquad P(\theta_2|D) = 0.5$$
$$p(y_{t+1} \mid y_t, \theta_3) = .6. \qquad P(\theta_3|D) = 0.1$$

   What is the maximum a posterior estimate for $p(y_{t+1} \mid y_t)$? Based on the maximum a posteriori (MAP) estimate, would you be better off buying or selling?

$0.5 > 0.4 > 0.1$

Thus maximum a posterior estimate is $P(\theta_2|D)$ for $P(y_{t+1}|y_t)$

Based on MAP, we better on selling.

5. **[4 points]**

   What is the estimate according to the posterior predictive distribution for $p(y_{t+1} \mid y_t)$? (I.e., using model averaging.) Based on the posterior predictive distribution (PPD), would you be better off buying or selling?

$$Pr(Y|D) = \sum_\theta Pr(Y|\theta)Pr(\theta|D) = p(y_{t+1} \mid y_t, \theta_1) \times 0.4 + p(y_{t+1} \mid y_t, \theta_2) \times 0.5 + p(y_{t+1} \mid y_t, \theta_3) \times 0.1$$

$$= 0.56$$

$\because 0.56 > 0.5$

$\therefore$ buying