Programming Assignment 07: Supervised Learning (50 points)

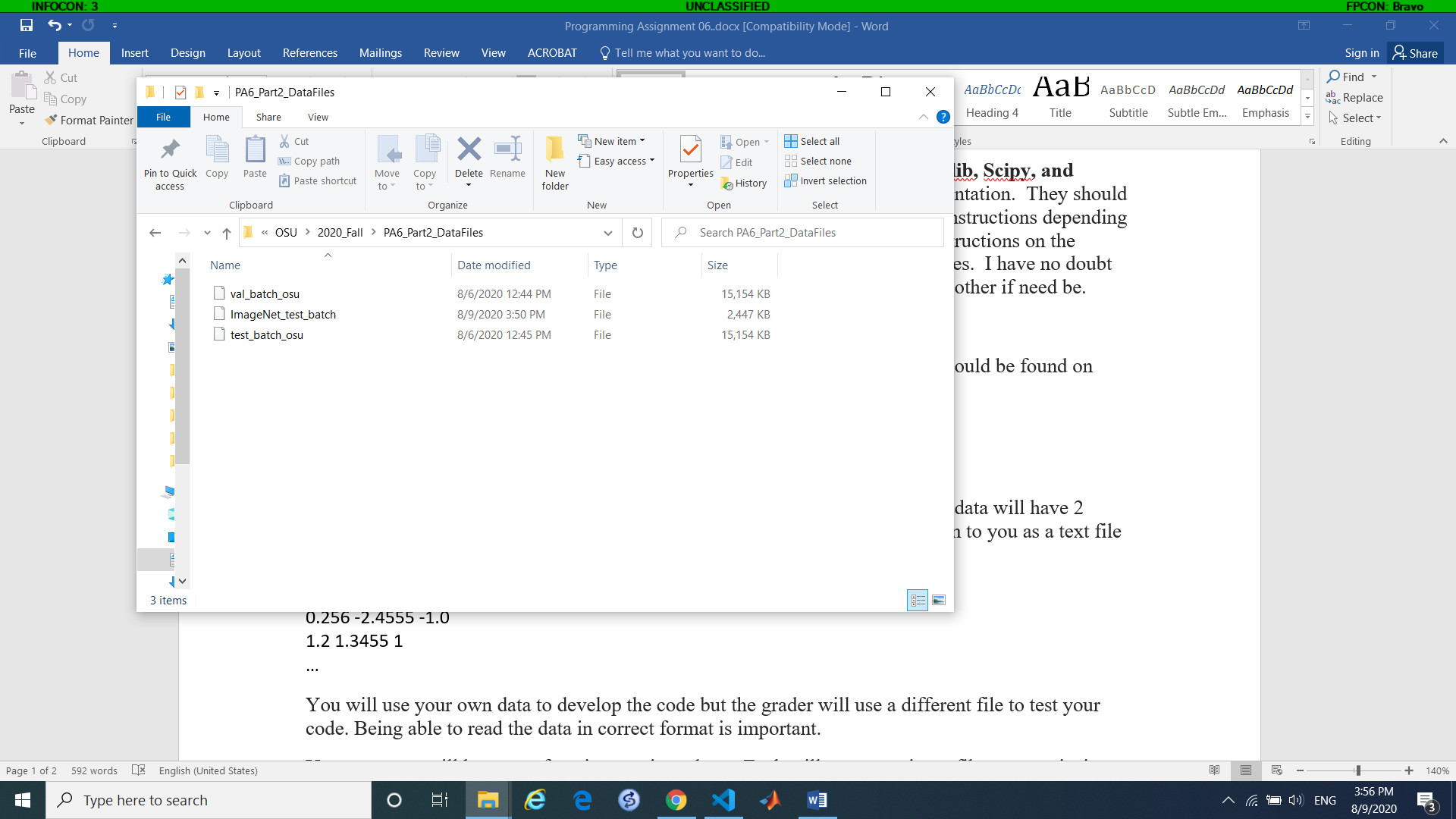
In this assignment you will explore supervised learning with PyTorch – a deep learning package.

**Software**

In addition to Python, you will need the following packages**: NumPy, Matplotlib, Scipy, and Torch**. You may need other libraries too depending on your particular implementation. They should all be easily installable via pip command. There are differences in installation instructions depending on your OS, and you should be able to figure out installation by reading the instructions on the corresponding websites or googling your problems and/or asking your classmates. I have no doubt that all of you will successfully setup programming environment and help each other if need be.

**Additional data**

These are the files you will need for Part 2 of this assignment, explained below. They can be found on Carmen.



**Problem**

For this part of the assignment you will train an image classifier using a popular database of small size images called CIFAR-10. Information about the dataset can be found here.

<https://www.cs.toronto.edu/~kriz/cifar.html>

You can get the data from the web site above. There are a total of 50,000 training images and 10,000 test images. Since they don’t provide separate validation set, I divided the test batch into two parts. Use one of them as validation set and the second one as test set. Both batches are on Carmen.

**Network definition.**

The size of the input images is 32x32, RGB. Use the following network structure for your assignment.

class Net(nn.Module):

    def \_\_init\_\_(self):

        super(Net, self).\_\_init\_\_()

        self.conv1 = nn.Conv2d(3, 6, 5)

        self.pool = nn.MaxPool2d(2, 2)

        self.conv2 = nn.Conv2d(6, 16, 5)

        self.fc1 = nn.Linear(16 \* 5 \* 5, 120)

        self.fc2 = nn.Linear(120, 84)

        self.fc3 = nn.Linear(84, 10)

    def forward(self, x):

        x = self.pool(F.relu(self.conv1(x)))

        x = self.pool(F.relu(self.conv2(x)))

        x = x.view(-1, 16 \* 5 \* 5)

        x = F.relu(self.fc1(x))

        x = F.relu(self.fc2(x))

        x = self.fc3(x)

        return x

**Task** 1. Investigate how the classification accuracy of the trained network depends on the size of training dataset. Each student is assigned a range of values for the dataset in the spreadsheet found on Carmen. Please train your network with the ranges assigned to you. Run each value in the range 3 times. Submit the results as a text file with four columns: number of sample and 3 accuracies. It should look like this:

500 0.45 0.452 0.387

1000 0.57 0.64 0.453

…

Use the following parameters for learning: learning rate 0.001, momentum 0.5, minbatch size 4. Train for 200 epochs.

NOTE: Running all the experiments above will take time, so do not do this last minute! You will need several days just to run all the experiments.

After everyone submits the results I will combine them into one nice graph of Accuracy vs. Training Set size. Please make sure you use the exact same network architecture and parameters, otherwise the combined plot will not look good ☺

**Task** 2. Using all 50,000 training examples, train the best network and test it on an additional test set that I created from ImageNet. Compare the performance to that on the CIFAR-10. Explain your observations.

NOTE: You will have to save your networks and load them later for testing. Don’t forget that the program must include all the network definitions in order to load your saved networks!

As an extra, you are welcome to pull some images from the internet or from your camera if you like, resize them to the appropriate size and apply your networks to them. How do they perform?

**Task** 3. Alter the network structure and retrain it. You may want to change the size of convolutional layers, the number of layers in the network, the size of linear layers, the transfer function. Try several structures and compare their performance on CIFAR and ImageNet testing set. Which network structure gave the best performance on ImageNet?

**Task** 4. Write a python program that will take the network filename and the image filename as input as return the predicted label. This will require some image manipulation: you will have to resize and normalize the image before passing it through the network.

Please track the amount of time you spent on this project and report it with the submission.

Submission

Submit your solution to Carmen. You need to submit a zip file that contains the following.

1. The code used to run experiments in Task 1.
2. The result of experiments in Task 1, as described above, in a text file
3. The networks that gave you the best performance on ImageNet, one for each architecture you tried, saved using Torch.save() command.
4. Python program that allows to test your network on a dataset saved in the same format as the CIFAR batches. The program will take the network filename and the data filename as inputs and return classification accuracy.
5. Python program that allows to test your network on a single input image. The program will take the network filename and the image filename as input and return the assigned label.
6. Report with
   1. README instructions for your code.
   2. Performance comparison between CIFAR and ImageNet from in Task 2
   3. Network architectures that you tried and their performance comparison from Task 3
   4. Estimate of time spent on the project
   5. NOTE: Report should be brief, no need for writing an essay.

Your code will be reviewed, the grader will test your networks on the ImageNet dataset and a single image, and review your report.