

ECCE 635 Deep Learning Systems Design

Fall 2020

Assignment 2

Due Date: Sunday - September 27, 2020 (12:00 midnight)

Submission: You must submit three files through Blackboard:

- A PDF file containing your writeup titled Student_ID_Name_writeup_Assignment2.pdf
- Two code files titled: Student_ID_Name_code_Assignment2_MLP.ipynb and Student_ID_Name_code_Assignment2_CNN.ipynb

Part 1: Conceptual Questions

1. Why is it necessary to include non-linearities in a neural network?
2. When the input is 2-dimensional, you can plot the decision boundary of your neural network and clearly see if there is overfitting. How do you check overfitting if the input is 10-dimensional?
3. You are designing a deep learning system to detect driver fatigue in cars. It is crucial that that your model detects fatigue, to prevent any accidents. Which of the following is the most appropriate evaluation metric: Accuracy, Precision, Recall, Loss value. Explain your choice.
4. You want to solve a classification task. You first train your network on 20 samples. Training converges, but the training loss is very high. You then decide to train this network on 10,000 examples. Is your approach to fixing the problem correct? If yes, explain the most likely results of training with 10,000 examples. If not, give a solution to this problem.
5. Give benefits of using convolutional layers instead of fully connected ones for visual tasks.
6. You are training a neural network model. You initialize the parameters with 0's. Is this a good idea? Explain your answer.
7. You are given a dataset of 28 X28 grayscale images. Your goal is to build a 5-class classifier. You have to adopt one of the following two options:
 - the input is flattened into a 784-dimensional vector, followed by a fully connected layer with 5 neurons.
 - the input is directly given to a convolutional layer with five 10 X10 filters.

Explain which one you would choose and why.

8. You would like to train a dog/cat image classifier using mini-batch gradient descent. You have already split your dataset into train, validation, and test sets. The classes are balanced. You realize that within the training set, the images are ordered in such a way that all the dog images come first, and all the cat images come after. A friend tells you: "you absolutely need to shuffle your training set before the training procedure." Is your friend, right? Explain.

Part 2: Programming Assignment

In this assignment, we will train Multi-layer perceptron (MLP) and Convolutional Neural Network (CNN) models like the one we covered in the lecture. The networks receive MNIST data as input, and the aim is to classify MNIST digits. Look at the “Assignment_2.ipynb”. It contains the code that you will be using for this assignment.

Part 1:

In the first part of the programming assignment, you will modify the code in the “Assignment 2.ipynb” by re-training a couple of new models using different parameters and network structures. You will train **five** new models in a new code cell by copying and modifying the code from the last notebook cell. Comment on how the results (training loss, validation loss, testing accuracy) change as we change the network structure and parameters. Explain your findings.

Part 2:

In the second part of the programming assignment, you will modify the code in the “Assignment 2.ipynb” to train a convolutional neural network instead of MLP using the same data set. Create a new python notebook, and you will train **five** new models in a new code cell by copying and modifying the code from the last notebook cell. Comment on how the results (training loss, validation loss, testing accuracy) change as we change the network structure and parameters. Explain your findings.

Hint: For Part 1 and Part 2, there are various parameters you can play with, such as change regularization types, data normalization types, optimizer, activation functions, weights initiation methods, epochs number, and add or remove hidden layers, etc.

Part 3:

Compare between MLP and Convolutional Neural Network models for character recognition?