**Introduction to Machine Learning (Spring 2020)**

**Homework #4 (50 Pts, Due Date : June 8)**

**Student ID**

**Name**

**Instruction:** We provide all codes and datasets in Python. Please write your code to complete Convolutional Neural Network Classifier.

* HW4\_STUDENT\_ID\_Name.zip : compress ‘Answer.py’ and ‘main.py’
* HW4\_STUDENT\_ID\_Name.pdf : Your document converted into pdf

Please follow the submission format. We will not be held responsible for any penalty caused by non-compliance.

**(1) [40 pts]** Implement CNN Classifier in ‘Answer.py’ with the loss function as follows:

**[IMPORTANT]** In this assignment, we make **assumptions** as follows for simplicity:

* All elements of input are covered by conv/maxpool filters and all elements of conv/maxpool filters always cover the input. (see ‘check\_\*\_validity’ function in ‘Answer.py’)
  + Any cases such as (5x5) input, (2x2) kernel, stride 2, pad 0 are not allowed and don’t need to be considered. (kernel is out of bounds)
  + e.g.)
* Assume image shape is (row - column). In a maxpooling layer, if there are duplicate elements to pool, maxpool an element with priority: 1) smallest row index first, 2) smallest column index first.

|  |  |
| --- | --- |
| **A** | **B** |
| **10** | **10** |
| **C** | **D** |
| **10** | **4** |

1. Smallest row index first: A,B,C -> A, B
2. Smallest column index first: A,B ->A
3. Pool 10 from A
4. **[Activation]** Implement ReLU, Sigmoid, Tanh activation in ‘Answer.py’ (‘ReLU’, ‘Sigmoid’, ‘Tanh’).
5. **[FC Layer & Softmax]** Implement a FC, softmax layer in ‘Answer.py’ (‘FCLayer’, ‘SoftmaxLayer’).
6. **[zero\_pad]** Implement function ‘zero\_pad’ in ‘Answer.py’.
7. **[Convolution Layer]** Implement a convolution layer in ‘Answer.py’ (‘ConvolutionLayer’).
8. **[Max-Pooling Layer]** Implement a max-pooling layer in ‘Answer.py’ (‘MaxPoolingLayer’).

**Answer: Fill your code here. You also have to submit your code to i-campus.**

**NOTE 1**: **You should write your codes in ‘EDIT HERE’ signs.** It is not recommended to edit other parts. Once you complete your implementation, run the check codes (‘Checker.py’) to check if it is done correctly.

**NOTE 2**: **Read the instructions in template codes VERY CAREFULLY.** Functionality and input, output shape of any function must be the same as what is written.

**(2) [10 Pts]** Experiment results

1. **[Plot Graph]** You are given a MNIST dataset with 10 labels (0~9). Given CNN architecture and hyperparameters as below, build the classifier and **report train/valid accuracy of the first 3 epochs, best valid accuracy**, and **test accuracy. Draw train/valid accuracy plot – (x-axis : epoch / y-axis : accuracy).**

**Answer: Fill the blank in the table. Show the plot of train/valid accuracy with a brief explanation.**

**[CNN Architecture]**

|  |  |
| --- | --- |
| **Layer name** | **Configuration** |
| **Conv - 1** | Out Channel = 4, Kernel size = 3 Stride = 1, Pad = 1 |
| **ReLU - 1** | - |
| **Conv – 2** | Out Channel = 4, Kernel size = 3 Stride = 1, Pad = 1 |
| **ReLU - 2** | - |
| **Max-pool - 1** | Kernel size = 2, stride = 2 |
| **FC – 1** | Input dim = 784, Output dim = 500 |
| **ReLU - 3** | - |
| **FC - 2** | Input dim = 500, Output dim = 10 |
| **Softmax Layer** | - |

|  |  |  |  |
| --- | --- | --- | --- |
| **Epochs** | **Learning rate** | **Reg. lambda** | **Batch size** |
| 10 | 0.01 | 0.001 | 1000 |

**[Results]**

|  |  |
| --- | --- |
| **Best Valid Acc.** | **Test Acc** |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Epoch** | **Train accuracy** | **Valid accuracy** |
| **1** |  |  |
| **2** |  |  |
| **3** |  |  |

**Plot Sample (Values are not correct. Delete when you submit)**

