CSE641 Deep Learning
Assignment 02

Date: 8 Mar 2022
Deadline: 1 April 2022 **Max Marks: 50**

General Instructions:

1. Only Pytorch or Tensorflow can be used as a DL framework. (*Keras is not allowed)*
2. Each group member must do at least one of the following tasks. But both should know the working of all the tasks. (Recommended: Divide the parts among yourselves.)
3. The assignment can be submitted in a group of a maximum of two members.
4. For plagiarism, institute policies will be followed strictly.
5. Make sure to use Pickle or any other library to save all your trained models. There will not be enough time during the demo to retrain your model. This is a strict requirement. If you are not able to reproduce your results during the demo then no marks will be given.
6. Create a 'Report.pdf' report that contains your approach, pre-processing, outputs, and all the analysis. Anything not in the report will not be marked.
7. You need to submit Report.pdf, Code files (.py files or .ipynb files), and models dumped after training.
8. Mention outputs, methodology, helper functions, preprocessing steps, any assumptions you may have in Report.pdf. Submit all the files in single ZIP format with the following name: ***A2_Member1_Member2.zip***
9. You are advised to prepare a well-documented code file.
10. Use classroom discussion for any doubt.
11. Most probably you won't be able to run these programmes on your desktops because they are computationally expensive. Use Google colab or other workstations(servers) if possible.
12. Note: You may reduce the size of the training dataset depending upon the resources available at your disposal. But the testing set size should remain the same.

PART I: Convolution Neural Network (CNN) (25 marks)

Dataset: *Cell Image*
Dataset description: The dataset contains a total of 27,558 cell images with equal instances of parasitized and uninfected cells.
Task: Develop a neural network to classify, whether a cell is infected or not using Convolution Neural Network. Perform splitting into training, testing and validation sets with 80:10:10 ratio.

1. Visualize 5 random images from both the classes. [5 marks]
2. Implement a CNN architecture with blockA [9x9] followed by fully connected layer, blockB [6x6] followed by max pooling, blockC [3x3] followed by fully connected layer and finally a sigmoid layer. [5 marks]
3. Initialize your neural network weights by using following initialization methods: [3 marks]

a. Zero initialization
b. Random Initialization
c. He initialization

Which initialization approach is best and why?

4. Implement Dropout and use i) After convolutional layers, ii) Between fully connected layers. Compare the performance in both the cases. [6 marks]
5. Keeping the above architecture same, implement the following regularizations and do a thorough analysis on the output of each one of them: [6 marks]
   a. L1 Regularization
   b. L2 Regularization

   Compare the performance in both the cases. Which regularization is better and why?

PART II: Long Short-Term Memory (LSTM) (25 marks)

Dataset: *Mini Daily Dialog* - Use *train.csv* and *test.csv* file for the assignment.
Note: *Data trimming is not allowed in this question.*
Dataset description: The dataset is a sample of widely used DailyDialog dataset for dialogue-act classification task.
Task: Dialogue-act Classification: For a given dialogue students need to develop a program to predict act of utterance at time ***T*** with the help of previous ***X*** utterances as context.

1. Visualize dialogue corpus and show stats of the train and test file. [2 Marks]
2. Implement a program using just LSTM and linear layers to predict act of utterance at time ***T*** considering previous ***X*** utterances' context. [8 Marks]
   Note: *Students need to propose an architecture for this.*
3. Now, show plots for accuracy and weighted F1 scores for X = {0,1,2,3,4} [10 Marks]
4. Does the performance of the model increase with increase in X? Justify. [5 Marks]

Expected deliverables of the assignment (for both the parts):

- For every question within each part, visualize the learning using the following plots:
  - Training Loss vs Number of Epochs
  - Validation Loss vs Number of Epochs
  - Plots showing convergence over different values of X

- Save the models in pickles. Students will be asked to reproduce results using saved models only.