IE643: Deep Learning: Theory and Practice **July-Dec 2021**

Programming Challenge: Due On 5th November 2021 (11 59 PM IST)

1 Instructions

This is a programming challenge which needs to be attempted individually by each course participant. It is completely upto the participant to decide upon the choice of neural network architectures to solve the challenge. The architecture used for the task, the data pre-processing techniques, data augmentation ideas (if used), training strategy, hyper-parameter selection used during training etc. should be clearly described in a report, with all relevant details. A clear guide on how to use the code must be provided in the report. The final code and trained deep neural network model should be made available in a suitable form amenable to be tested using a private test data. In particular, proper code for using the model for predicting on test data must be provided. More details are given in Section 2. The submissions will be evaluated for a maximum of 50 marks.

Codes must be in Python language. Other programming languages are not allowed. Pytorch framework is preferred, though keras and tensorflow are also permitted. The participants might refer to any resource for completing this challenge. However it is suggested that all resources referred to are cited in the report and in the code.

The form to collect submissions (including the code, report and other related files) will be posted later.

IMPORTANT: Submissions which contain copied code and ideas will not be evaluated.

2 Programming Challenge Question

Consider the data in the link posted in moodle and teams (using your IITB Google SSO login). The data is in the form of a zip file which when decompressed will yield a folder containing 7 subfolders named 1,2,3,4,5,6 and 7. These folder names refer to the class labels for the files present in the respective folders. Each folder contains `.wav` files representing audio clips. Please feel free to explore the data by listening to the audio clips.

As part of the challenge, you will be answering the following:

1. Construct a suitable deep neural network architecture to classify the audio clips into their respective classes. The deep network architecture can be any network of your choice. The components of the architecture can also be chosen according to your choice. However you will describe all details of the deep neural network clearly in the report. Proper justification of the different components of the neural network must also be provided.

2. Assess if the data set provided can be used as it is to train the deep neural network you have constructed. If required, you can introduce suitable data pre-processing and can augment the data with additional data if required. You would describe about the data pre-processing techniques in your report. Also, you would include details in your report about any data augmentation used during training.

3. In the report, include the details of the training procedure used for training the deep neural network, *e.g.* loss function and optimization scheme used in the training, details of hyperparameter tuning, learning rate scheduling, early stopping criteria, cross-validation procedure used in training.
4. In the report, provide plots obtained for training loss vs. epochs, validation loss vs. epochs, training accuracy vs. epochs, validation accuracy vs. epochs. Any other relevant plot can be included.
5. Along with the report, the entire code and trained model should also be submitted.
6. All your files should be named according to the conventions `IE643_YOURROLLNO_CHALLENGE_CODE.ipynb`, `IE643_YOURROLLNO_CHALLENGE_REPORT.pdf`, `IE643_YOURROLLNO_CHALLENGE_MODEL.pt`, etc. Files with other naming conventions will not be considered for evaluation.
7. Your submission should be posted in Google drive and the link should be shared for evaluation purposes. Please make sure that the link is accessible by the TAs. The form to collect the link will be posted around the submission deadline.

The code needs to allow the following options for the user:

1. The trained model should be loadable in the code.
2. The user should be able to input a test set of audio clips and their corresponding true labels as inputs to the trained model.
3. The trained model should then provide the class labels for the test set.
4. Along with the predicted labels, the code should compute and output test set accuracy, precision, recall and F1-score.

The participant is encouraged to make the code very interactive. A clear guide on how to use the code must be provided in the report.

The submissions which give top 3 test set accuracies on a private test set will be highlighted and all submissions with significant efforts will be awarded extra marks which would be considered in the final grading.