Deep Learning Lab 1

Hitesh Goyal - 19BAI1129,
VIT Chennai, Tamil Nadu, India 600127
Email: hitesh.goyal2019@vitstudent.ac.in

Abstract—**Before starting to learn, it is important to know the environment in which it something exists. In case of Deep Learning, it is good to be familiar with Interactive Python Notebooks or *ipynb* files. The main platforms where one can run these include Google Colaboratory, Jupyter and Visual Studio Code. This paper explores the first one, Google Colaboratory.**

1. Introduction

For the first lab session, a few ways to load datasets, a few features and functionalities in Google Colaboratory and the common method to create a Neural Network will be explored.

2. Methodology

2.1. Loading Datasets

2.1.1. Pre-loaded datasets on Google Colaboratory. Google colaboratory provides some datasets as already available. They need not be loaded from the drive or using any API integration.

2.1.2. Loading through Kaggle API. Kaggle provides and API which can help integrate with Google Colaboratory providing easier loading of larger data which had to have been downloaded and uploaded from a drive otherwise. To use the Kaggle API, it is suggested to follow the Kaggle documentation.

2.1.3. Mounting Google Drive and loading from the Drive. Google Colaboratory allows its users to mount their drive and directly access their data from their rather than having to load it every time the code was run. This is also almost as helpful as the API provided by Kaggle and most useful when the dataset is not available on Kaggle.

2.2. Features and Functionalities

2.2.1. Installing libraries. Google Colaboratory provides the option to install any library which may not already be installed. Although, these libraries may have to be installed again at the start of a new session. The command to install a library looks like:

```
!pip install <library name>
```

2.2.2. Running on different hardware accelerators. Google Colaboratory provides different hardware accelerators to boost the speed of model training. It has the option to run a GPU (Graphical Processing Unit) accelerator or a TPU (Tensor Processing Unit) accelerator. In the case of training models on MNIST dataset, not much of a difference is found. It is still suggested to use an accelerator if large datasets have to be trained on deep neural networks for a larger set of epochs.

2.3. Training Neural Networks

2.3.1. Training Neural Networks on MNIST. MNIST is a Computer Vision related dataset which contains English digits from zero to nine. It is among the first datasets a Deep Learning student comes across. It is possible to train neural networks on MNIST dataset by following a blog that can be found on Neptune AI's website.

2.3.2. Splitting of data. To be sure of the reliability of a model, it is important to split the data into three parts- training, validation and testing. The model is trained on the training data and tuned on the validation data. Once it is considered to be optimally tuned, the actual accuracy is checked with the testing data.

2.3.3. Training Neural Networks on self-accumulated dataset. The dataset was extracted by preprocessing audio data which was collected in Winter Semester 2020-21 for a speech recognition project. It contains 30 Mel Frequency Cepstral Coefficient Means, which can be used to predict ten words. This dataset was uploaded on Google Drive which was then accessed in Google Colaboratory by mounting the Drive. Any Neural Network model can be created for training on the dataset as long as the input shape and the output shape is correctly defined.

3. Conclusion

Multiple different aspects of Google Colaboratory were successfully explored. Multiple ways of loading datasets, many different features of Google Colaboratory and the successful training of a Neural Network are among the different aspects explored.