Assignment 2

Name : Smitesh Nitin Patil Student Id: 22223696

Name: Ambuj Mittal

Name: Dhruv Ajit Solanki

**1.At first page, write your names, roll numbers, and group number, and then in 50-150 words, report who did what in the assignment e.g., who came up with the idea of how to implement it, who implemented it (as a whole or part of it), who debugged it, who gave suggestions (and what suggestion/s), who wrote the report, who did which simulations**

Answer:

**2. Explain how your group implemented the layer within existing library (keras, TensorFlow, PyTorch, or other) or from scratch [3 Marks]**

Answer: For the task of creating new proposed CNN model with weighted matrix we have implemented it from scratch…

For the traditional CNN, we have used keras to create a CNN with 3 Convolutional and 2 Dense layers with a SoftMax function for our classification task.

**3. Report the successful implementation of the layer and complete model with a figure in tensorboard. [10 Marks]**

Answer:

**4. Load and distribute the data for training, validation, and testing [2 Marks]**

Answer : The data for alpaca and not-alpaca folders where initially converged in a single data frame with the labels using create\_dataset() created in the jupyter notebook then data was shuffled and split in an 80-10-10 split for training, testing and validation purposes.

**5. Train the model with the dataset provided in this assignment folder and report the training and testing performance against various hyper parameters (e.g. learning rate = 0.1, batch size = 32, epochs = 500,) using stochastic gradient decent [9 Marks].**

Answer:

For CNN with new proposed layer:

For Traditional CNN : traditional CNN was trained using the keras fine tuner which takes a set of hyperparameters, train on those parameters and return the best trained model. There are four tuners in keras tuner package, ‘Random Search’, ‘Hyperband’, ‘Bayesian Optimization’, ‘Sklearn’.

We have used ‘Hyperband’ tuner here it initially trains a number of models for a small number of epochs and then considers models performing best on the later epochs. Hyperband determines the number of models to train in a bracket by computing 1+ log(factor)(max\_epochs)

**6. Report the analysis of the model against changes in network depth and width as well as changing hyper parameters i.e., show how performance increase or decrease by changing learning rate {0.01, 0.001, or other}, batch sizes {8, 16, 50, or others depending on your system/laptop capacity}, epochs {100, 200, 700, 1000 or other depending on the model performance}, optimizers e.g. Adam or other factors. Report all results in a table, even if they are not good. Also show their training and testing performance in the form of a graph [7 Marks]**

Answer:

**7. Report what result you achieved after training a Convolutional Neural Network of same depth [5 Marks].**

Answer:

**8. How to change the code to work over 1-Dimensional data? [2 Marks]**

Answer:

Traditional CNN : the kernel size would be of a single dimension the same as number of dimension of data. keras.layers.Conv1D function would be using to create a convolution layer of size 1.

**8. How to change the code to work over 3-Dimensional data? [2 Marks]**

Answer:

Traditional CNN: