Assignment-7

Problem Statement 1: Build a text classification RNN (Recurrent Neural Network) model using IMDB dataset. Import the *IMDB* dataset using `tensorflow_datasets` and perform the following tasks.

Tasks to be performed:

- Import the required libraries
- Shuffle the data for training and create batches of text and label pairs
- Encode the text data

Note: The simplest way to process text for training is using the TextVectorization layer.

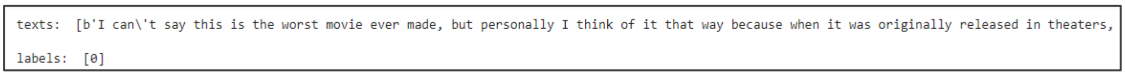
- Build a sequential model using tf.keras.Sequential function
- Compile the model
- Train the model using train dataset
- Test the data using tset dataset and evaluate the model by passing a sentence

Note: If the prediction is >= 0.0, it is positive else it is negative.

Example:

Output:

- Import the required libraries
- Shuffle the data for training and create batches of text and label pairs



```
texts:  [b'I can\'t say this is the worst movie ever made, but personally I think of it that way because when it was originally released in theaters,

labels:  [0]
```

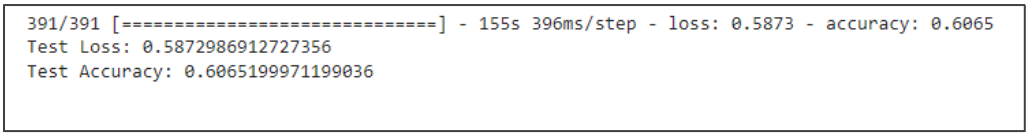
- Encode the text data

Note: The simplest way to process text for training is using the TextVectorization layer.

Create the layer, and pass the dataset's text to the layer's .adapt method.

- Build a sequential model using tf.keras.Sequential function
- Compile the model
- Train the model using train dataset
- Test the data using tset dataset and evaluate the model by passing a sentence

Note: If the prediction is >= 0.0, it is positive else it is negative.



```
391/391 [==============================] - 155s 396ms/step - loss: 0.5873 - accuracy: 0.6065
Test Loss: 0.5872986912727356
Test Accuracy: 0.6065199971199036
```

Problem Statement 2:

Sumanth wants to build an flowers image classification CNN model which can take a flower image as input and prints the name of flower.

Dataset Description: The dataset consist of 3,700 photos of flowers. The dataset contains five sub-directories, one per class:

- daisy
- dandelion
- roses
- sunflowers
- tulips

Tasks to be performed:

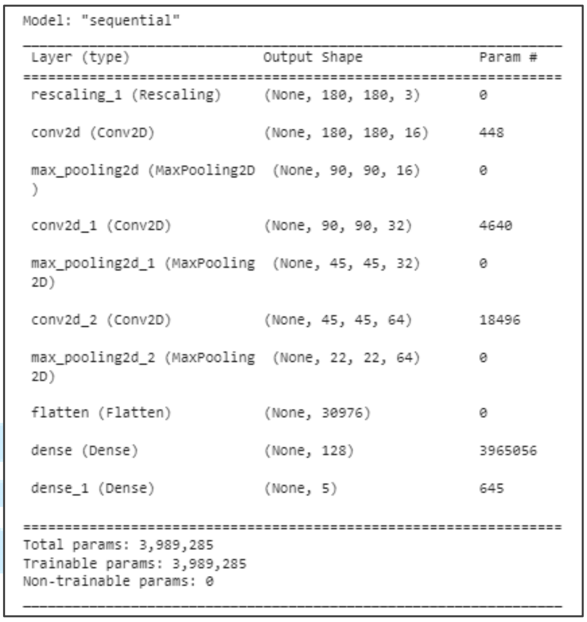
- Import all the required libraries
- Download the flowers data from URL: 'https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz' using the Keras; and analyzing the images present in the dataset visualization
- Prepare the data by specifying the image resolution and batch size. Also, Splitting the dataset into training and validation datasets in 80:20 ratio
- Plot a sample image attributed to each class name present in the original dataset.
- Autotune the data using cache().shuffle().prefetch() function; Also, normalize the data by deviding it with a number
- Build a sequential CNN model
- Compile the model and print model summary
- Train and validate the model and plot the validation results using a line plot
- Feed the system with an image of a flower and print the prediction along with the accuracy

Example:

Output:

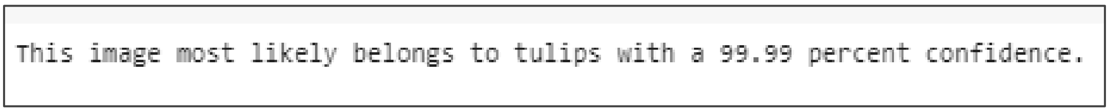
- Import all the required libraries
- Download the flowers data from URL: 'https://storage.googleapis.com/download.tensorflow.org/example_images/flower_photos.tgz' using the Keras; and analyzing the images present in the dataset visualization

- Prepare the data by specifying the image resolution and batch size. Also, Splitting the dataset into training and validation datasets in 80:20 ratio
- Build a sequential CNN model
- Compile the model and print model summary



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling_1 (Rescaling)     (None, 180, 180, 3)       0

 conv2d (Conv2D)             (None, 180, 180, 16)      448

 max_pooling2d (MaxPooling2D  (None, 90, 90, 16)       0
 )

 conv2d_1 (Conv2D)           (None, 90, 90, 32)        4640

 max_pooling2d_1 (MaxPooling  (None, 45, 45, 32)       0
 2D)

 conv2d_2 (Conv2D)           (None, 45, 45, 64)        18496

 max_pooling2d_2 (MaxPooling  (None, 22, 22, 64)       0
 2D)

 flatten (Flatten)           (None, 30976)             0

 dense (Dense)               (None, 128)               3965056

 dense_1 (Dense)             (None, 5)                 645

=================================================================
Total params: 3,989,285
Trainable params: 3,989,285
Non-trainable params: 0
_________________________________________________________________
```

- Feed the system with an image of a flower and print the prediction along with the accuracy



```
This image most likely belongs to tulips with a 99.99 percent confidence.
```