Build a Review Classification Model using Gated Recurrent Unit

Overview

Gated Recurrent Unit(GRU) is an improved version of the standard Recurrent Neural Network(RNNs). As a variant of the recurrent neural network, they are able to process memories of sequential data by storing previous inputs in the internal state of networks and plan from the history of previous inputs.

In this project, an application review classification model has been built using GRU. The project focuses on classifying the review of the application based on a 1 to 5 scale with 1 being negative sentiment and 5 being positive sentiment. If you haven't visited already, here is the previous project of the series Build CNN Image Classification Models for Real Time Prediction

Aim

- To understand the Gated Recurrent Unit
- To classify the review of an app on a scale of 1 to 5 using Gated Recurrent Unit

Data Description

The dataset contains the reviews and the ratings of the app. The dataset has a score column and content column. The score columns have a number range between 1 to 5 based on the content column.

Tech Stack

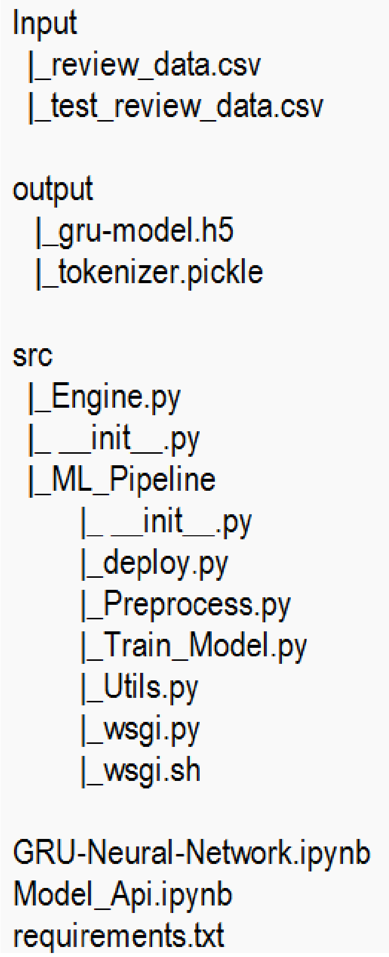
➔ Language: Python
➔ Libraries: pandas, tensorflow, matplotlib, sci-kit learn, pillow, gunicorn, textblob, nltk, keras, flask

Approach

- Data Preprocessing
    - Converting words to lower case
    - Lemmatization of the words
    - Tokenization of the words
    - One hot encoding of the scores
- Model Training

    - Training sequential model in Tensorflow
- Model Evaluation
    - Evaluation of model on test data

Modular Code Overview



```
Input
  |_review_data.csv
  |_test_review_data.csv

output
   |_gru-model.h5
   |_tokenizer.pickle

src
  |_Engine.py
  |_ __init__.py
  |_ML_Pipeline
         |_ __init__.py
         |_deploy.py
         |_Preprocess.py
         |_Train_Model.py
         |_Utils.py
         |_wsgi.py
         |_wsgi.sh

GRU-Neural-Network.ipynb
Model_Api.ipynb
requirements.txt
```

Once you unzip the gru_neural_net.zip file the following folder can be found

1. input
2. output
3. src
4. GRU-Neural-Network.ipynb
5. Model_Api.ipynb
6. requirements.txt

1. The input folder contains the data for the analysis. In our case its review_data.csv and test_review_data.csv for testing

2. The output folder contains the saved GRU model and pickle file
3. The src folder is the heart of the project. This folder contains all the modularized code for all the above steps in a modularized manner. It further includes the following.
    a. ML_pipeline
    b. Engine.py

    The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called inside the Engine.py file.

4. The GRU-Neural-Network.ipynb and Model_Api.ipynb are the original python notebook we saw in the video

5. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command **pip install -r requirements.txt**

Takeaways

1. What is Gated Recurrent Unit(GRU)?
2. How does GRU work?
3. What is a Recurrent Neural Network(RNN)?
4. Why not use deep neural networks?
5. Architecture of RNN
6. RNN vs GRU
7. The problem of short term memory
8. Maths behind GRU
9. Applications of GRU
10. What are stopwords?
11. Data Preprocessing
12. Lemmatization of words
13. Tokenization of text
14. Building GRU neural network
15. Deployment on flask