Deep Learning Assignment

“Sentimental Analysis using twitter data”

Name - Sujay Gad

SRN - PES1201700177

Project idea/scope

Project is based on performing sentimental analysis on twitter dataset using deep learning models. Sentiment of a new tweet can be generated using the trained model. Sentimental analysis has huge scope in areas such as politics, economy, modern trends etc. Sentimental analysis can be used to predict next US president , analyse government performance over last couple of years , public opinion on a new fashion trend etc.

Dataset information

Dataset is taken from Stanford University link. Dataset consists of 1.6 million rows and 6 columns namely target class, id , time , tweetquery , username and the tweet. Since the dataset is too large 20,000 rows are sampled out each time the model is trained.

Data Cleaning and preprocessing

- Remove the columns which do not affect the sentiment of a tweet such as time , userid etc.
- Remove the weblink and twitter ids present in the tweets.
- Convert the words having apostrophe. Eg - don't to do not , can't to cannot.
- Remove non alphabetic characters from the tweets.
- Remove words of length 1 from the tweets.
- Use nlp library snowballstemmer so that 'play' , 'plays' ,'playing' all convert to 'play' and vocabulary size is reduced.

Splitting Dataset

Dataset is split into training and testing set. Split ratio is set as 0.75.

Embedding Layer

Word2Vec model present in Gensim module is used on the training set to convert words into vectors. Couple of years back max length of tweet permitted was 140 characters hence max length is set 200 and the sentence is padded with 0's using pad_sequence function.

Dropout Layer

Dropout layer is used to prevent overfitting. Unlike movie reviews or amazon product reviews tweets can be based on any topic hence twitter data usually contains noise hence to prevent overfitting dropout layer is used.

GRU Layer

GRU is preferred over LSTM since it consumes lesser memory and also number of floating point operations performed per second is less for this model.

Model Compilation

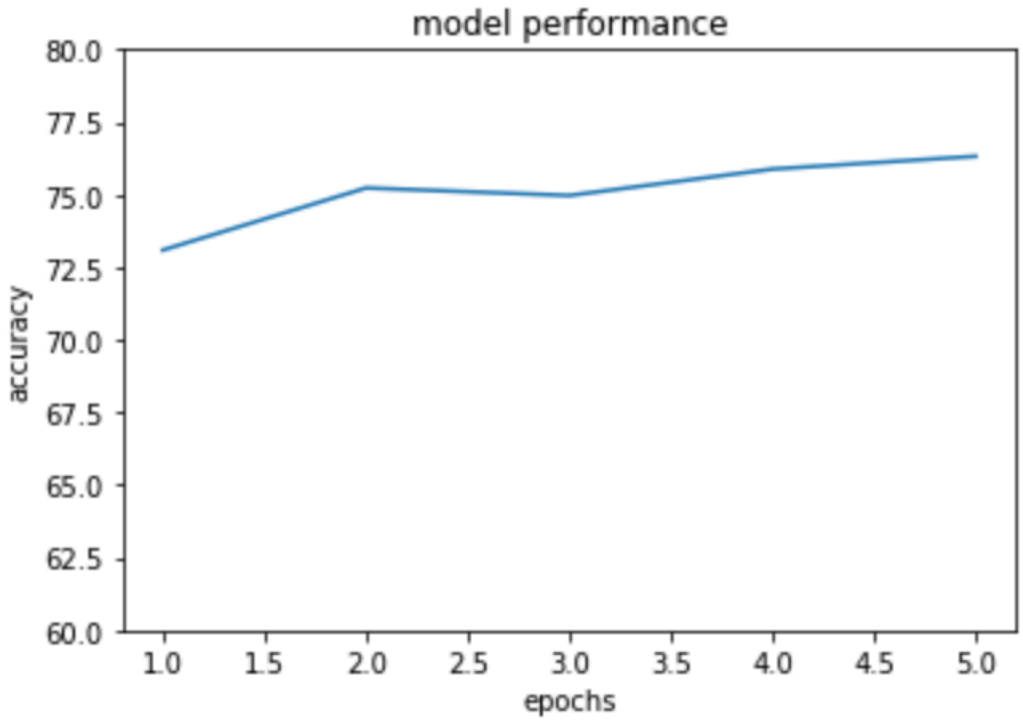
Binary cross entropy is used as the loss function and Adam is the preferred optimiser.

Target Class

Original dataset consisted of only 2 values in the target class column ie - 0,4 representing positive and negative sentiment.Since many tweets often do not have any + or - sentiment associated with it, a third target class called ‘neutral’ is added.

Performance

Overall performance of the model turns out to be around 75%. Loss decreases the number of epochs increase. Model performance is better than simple ML models such as model using Tf-idf or logistic regression but the performance is worse than complex models such as BERT.



Examples

```
predict("I love the beautiful music")
```

```
{'label': 'POSITIVE', 'score': 0.9673017263412476}
```

```
predict("I hate the rain Weather is terrible")
```

```
{'label': 'NEGATIVE', 'score': 0.04734661802649498}
```

```
predict("yes")
```

```
{'label': 'POSITIVE', 'score': 0.824225902557373}
```

```
predict("no")
```

```
{'label': 'NEGATIVE', 'score': 0.23617826402187347}
```

Shortcomings/Constraints of the Model

- This model works only on tweets written in English language, it ignores non English language words as well as emojis which may have emotions attached to it.
- Stop words such as this, that, and, or degrade the performance of the model as most of them do not have any sentiment attached.Stop words can't be removed since words like "not" play a big role in sentimental analysis.
- It also fails to differentiate 1 word having 2 different meanings in 2 different contexts eg - apple.
- Also for close to an ideal model sentiment('yes')+sentiment('no') must be approximately equal to 1 which is not the case for our model as we can observe from the previous section.

References

- Stanford University, "Sentimental analysis twitter data", http://cs.stanford.edu/people/alecmgo/trainingandtestdata.zip

- Ricky Kim, "Another Twitter sentiment analysis with Python — Part 1", https://towardsdatascience.com/another-twitter-sentiment-analysis-bb5b01ebad90

- NSS, "An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec", https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/