Emotion Annotation Neural Model

Submitted by: Harshita Rastogi
HXR190001

https://github.com/harshi29/Emotion-Annotator

1. Data Preprocessing

The data (json file) contains data in the following format:

```
{ ID : {“body”, “subreddit”, “created_utc”, “author” , “link_id” , “parent_id”,
        "emotion": {
                    “Anger”,
                    “Anticipation”,
                    “Disgust”,
                    “Fear”,
                     “Joy”,
                     “Love”,
                     “Optimism”,
                     “Pessimism”,
                     “Sadness”,
                     “Surprise”,
                     “Trust”,
                    “Neutral”}
       “complete”}
}
```

Feature Selection
The only important features required for the model is – “body” and “emotions”.
We ignore the other details as they are not required and the outcome will not depend on them.

Data Cleaning
Removing stop words and other unnecessary punctuations which are not required in our model. Having them will bias the output of the model.

Tokenizing words & Padding
Tokenizing words and converting them into indices. Padding the input entry with 20 words each.

Word Embeddings & Encoding
Glove Embedding Vectors are used to vectorize the words in the tweets, taking the weights of the words into consideration.

MultiLabelBinarizer is used for one hot encoding on to the various class labels.

2. Model Building

- **Embedding Layer**
  This layer acts as lookup table for vectors, given word index. It will return embedded word vector.

  input (None, 20) = >(Embedding Layer) => (None,20,50)

- **LSTM Layer**
  LSTM with 100 units is used which represent 100 RNN Cells.
  Return_sequences has been set as True since the output of the RNN layers will have outputs from all the units/cells that layer.

- **Activation Function**
  ReLU is the activation function that is used at every layer.
  Softmax is used to distribute the probability among labels for the given input.

  Optimizer used is Sgd

- **Metrics**
  top_k_categorical_accuracy
  Calculates the top-k categorical accuracy rate, i.e. success when the target class is within the top-k predictions provided. By default the k value is taken as 5.

- **Model Summary**

Model: "sequential_25"

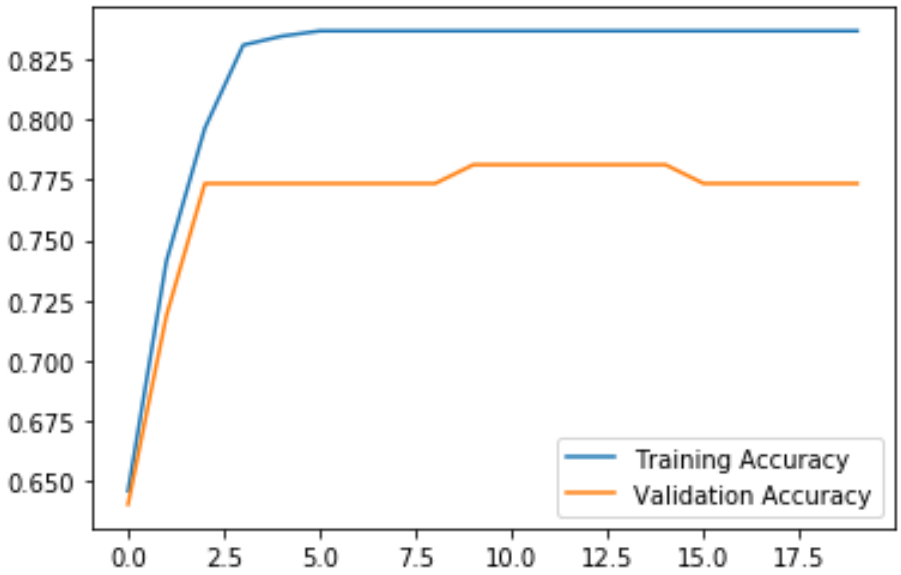
Layer (type)	Output Shape	Param #
embedding_69 (Embedding)	(None, 20, 50)	1678450
conv1d_25 (Conv1D)	(None, 20, 30)	1530
max_pooling1d_25 (MaxPooling	(None, 5, 30)	0
lstm_25 (LSTM)	(None, 5, 100)	52400
flatten_25 (Flatten)	(None, 500)	0
dense_73 (Dense)	(None, 500)	250500
dense_74 (Dense)	(None, 300)	150300
dense_75 (Dense)	(None, 12)	3612

Total params: 2,136,792

Trainable params: 458,342
Non-trainable params: 1,678,450

3. Training the Model

Training accuracy comes around 82%.



The model weights are stored in 'models.h5' file.

4. Testing the Model

TESTING

Total test accuracy is: 0.6887254901960784

ACCURACY

Emotion-wise test accuracy:
Anger 0.6764705882352942
anticipation 0.5855614973262032
disgust 0.4197860962566845
fear 0.5240641711229946
joy 0.8128342245989305
love 0.9197860962566845
neutral 0.8983957219251337
optimism 0.7406417112299465
pessimism 0.3422459893048128
sadness 0.6363636363636364

surprise 0.8689839572192514
trust 0.839572192513369

F1 SCORES

anger 0.0
anticipation 0.7377326565143824
disgust 0.5897920604914934
fear 0.6866197183098591
joy 0.0
love 0.0
neutral 0.0
optimism 0.0
pessimism 0.49382716049382713
sadness 0.0
surprise 0.0
trust 0.0