Lab Assignment 4

Notebook:	Machine Learning Lab		
Created:	19/11/2018 15:58	**Updated:**	19/11/2018 22:22
Author:	Prateek Chanda		

Prediction of Sentiment of Product Reviews

In this assignment, we had to predict the sentiment of a given product review (sentence) whether it is a positive or negative review.

Dataset

The dataset used
was https://www.kaggle.com/bittlingmayer/amazonreviews/downloads/test.ft.txt.bz2/2

Pre-processing

We first convert the data into a data frame object with two columns ['Sentiment Label', 'Review_Text'], where the first column represents whether it is a positive or negative review and the second column represents the text itself.

	Text	Sentiment
0	Stuning even for the non-gamer: This sound tra...	2
1	The best soundtrack ever to anything.: I'm rea...	2
2	Amazing!: This soundtrack is my favorite music...	2
3	Excellent Soundtrack: I truly like this soundt...	2
4	Remember, Pull Your Jaw Off The Floor After He...	2

We then remove all punctuation from the text itself using as follows

```
def punctuation_remove(s):
    punc_removed = str.maketrans({key: None for key in string.punctuation})
    return s.translate(punc_removed)
```

And lastly, we calculate the corresponding word counts and store it in the pandas Dataframe object

	Text	Sentiment	word_count
0	Stuning even for the nongamer This sound track...	1	80
1	The best soundtrack ever to anything Im readin...	1	97
2	Amazing This soundtrack is my favorite music o...	1	129
3	Excellent Soundtrack I truly like this soundtr...	1	118
4	Remember Pull Your Jaw Off The Floor After Hea...	1	87

After this, we select only those rows having word count less than 25.

Finally, we create a list of vocabulary(words) where the word count is more than 5 for the particular word over all the available text review.

We then create a one-hot vector representation for each of the sentences using the *sklearn CountVectorizer* function.

Building the Model

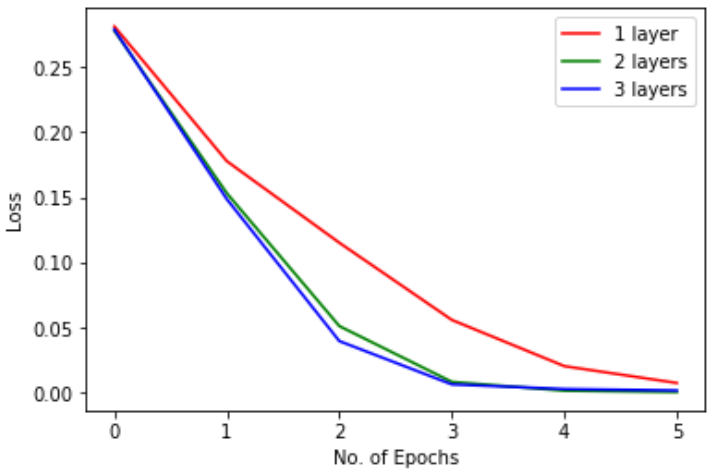
The fully connected neural network layer is then build using keras with hidden layers of size 1,2 and 3.
We then evaluate the performances based on the number of hidden layers.
We use the sigmoid cross-entropy function in order to evaluate the loss value.

```
model1=Sequential()model1.add(Dense(1000,input_shape=
(8915,),activation='relu'))
model1.add(Dense(1,activation='sigmoid'))
model1.compile(optimizer='adam',loss='binary_crossentropy',metrics=
['accuracy'])
```

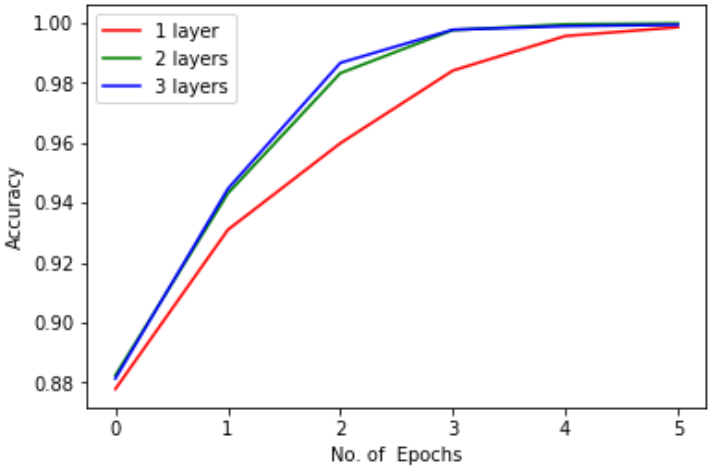
Evaluation

Lastly we plot the accuracy and the loss value obtained over the number of epochs.

Loss vs Epochs



Accuracy vs Epochs


1 layer
2 layers
3 layers
Accuracy
No. of Epochs
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0
1
2
3
4
5