```
In [0]: import pandas as pd
        import numpy as np
```

```
In [2]: !pip install -U -q PyDrive
        from pydrive.auth import GoogleAuth
        from pydrive.drive import GoogleDrive
        from google.colab import auth
        from oauth2client.client import GoogleCredentials
        # Authenticate and create the PyDrive client.
        auth.authenticate_user()
        gauth = GoogleAuth()
        gauth.credentials = GoogleCredentials.get_application_default()
        drive = GoogleDrive(gauth)
```

```
     |████████████████████████████████| 993kB 45.6MB/s
  Building wheel for PyDrive (setup.py) ... done
```

```
In [0]: link = 'https://drive.google.com/open?id=1wLPB72yau1pfH8uMp4Mx1Wo0sopd7
        Y2r' # The shareable link
```

```
In [0]: fluff, id = link.split('=')
```

```
In [0]: downloaded = drive.CreateFile({'id':id})
        downloaded.GetContentFile('traindataframeuploadfinalforsubmission.csv')

        traindataframe= pd.read_csv('traindataframeuploadfinalforsubmission.cs
        v',nrows=130000)
```

```
In [0]: link = 'https://drive.google.com/open?id=1-e0aY1FCfhTZ0y7uAlRN3j6MOMdvX
        YeS' # The shareable link
```

```
In [0]: fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('testdataframeuploadfinalforsubmission.csv')

testdataframe= pd.read_csv('testdataframeuploadfinalforsubmission.csv',
nrows=20072)
```

In [11]:
```
print(testdataframe.shape)
```

```
(20000, 6)
```

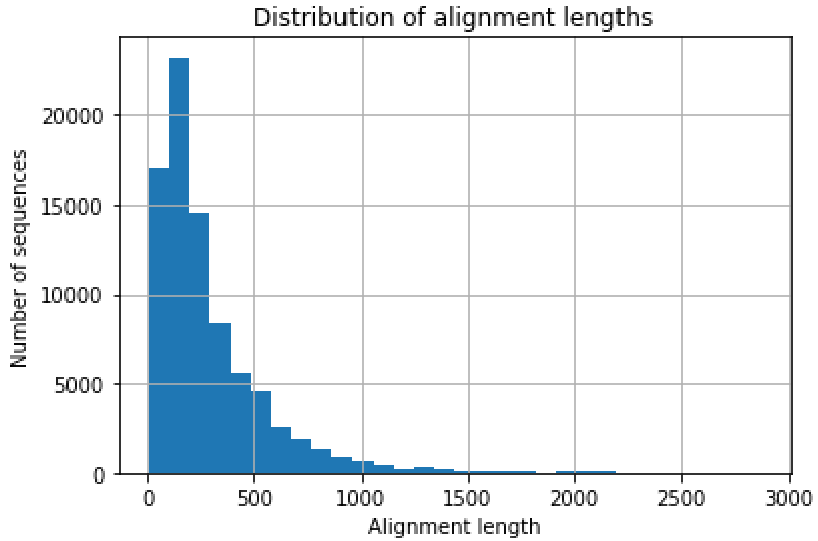
In [0]:
```
link = 'https://drive.google.com/open?id=1S8YirWWFIgpxNLuqiTjiIAAtyuIeR
A7r' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('cvdataframeuploadfinalforsubmission.csv')
cvdataframe= pd.read_csv('cvdataframeuploadfinalforsubmission.csv')
```

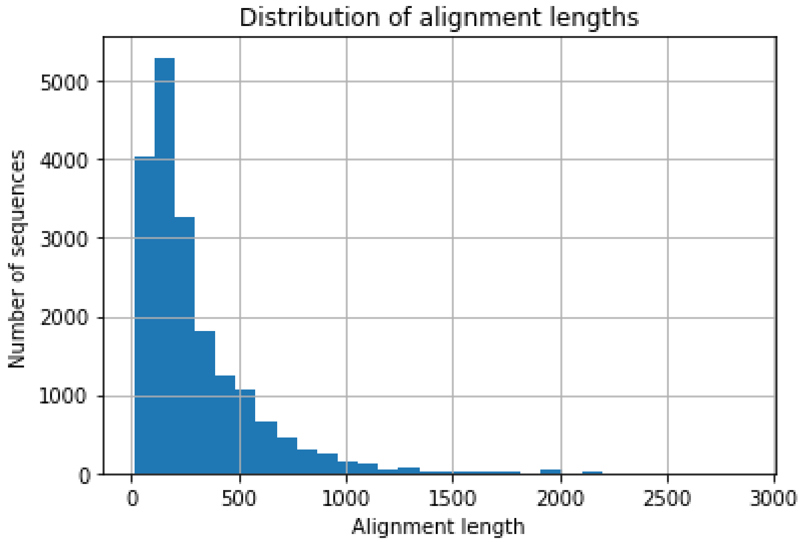
In [0]:
```
fd=pd.DataFrame()
import matplotlib.pyplot as plt
fd['alignment_length'] = traindataframe.aligned_sequence.str.len()
fd.alignment_length.hist(bins=30)
plt.title('Distribution of alignment lengths')
plt.xlabel('Alignment length')
plt.ylabel('Number of sequences')
```

Out[0]:
```
<matplotlib.text.Text at 0x2519beec5f8>
```



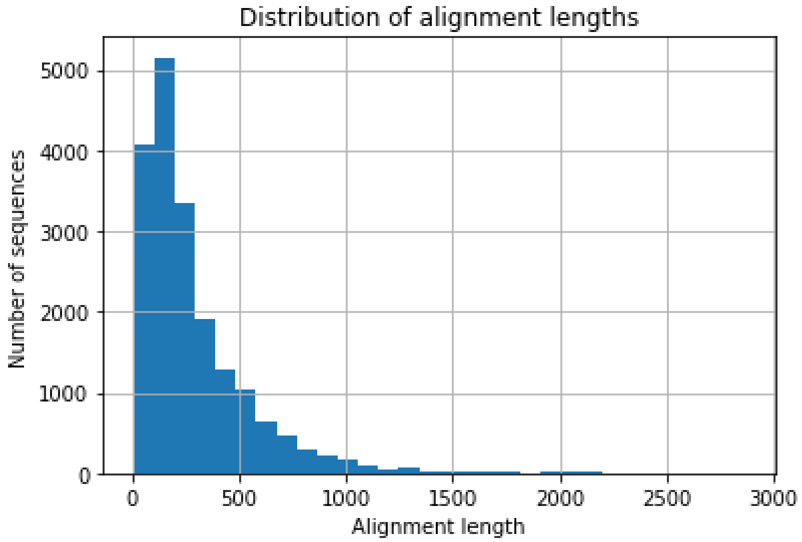
In [0]:
```
fd1=pd.DataFrame()
fd1['alignment_length'] = testdataframe.aligned_sequence.str.len()
fd1.alignment_length.hist(bins=30)
plt.title('Distribution of alignment lengths')
plt.xlabel('Alignment length')
plt.ylabel('Number of sequences')
```

Out[0]:
```
<matplotlib.text.Text at 0x25191d42c50>
```



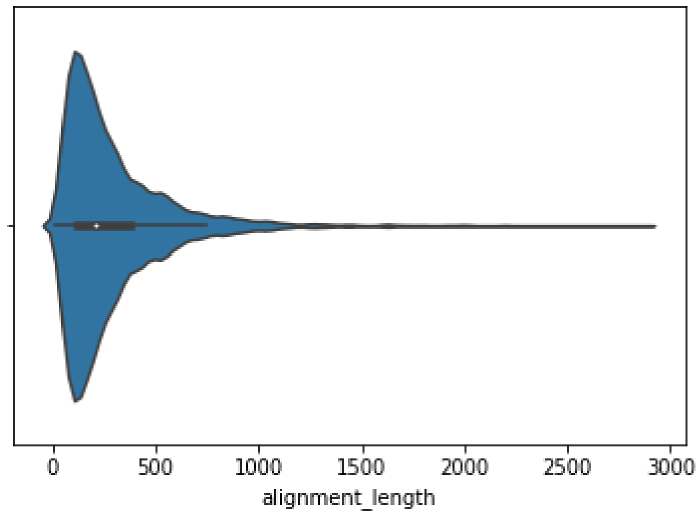
In [0]:
```
fd2=pd.DataFrame()
fd2['alignment_length'] = cvdataframe.aligned_sequence.str.len()
fd2.alignment_length.hist(bins=30)
plt.title('Distribution of alignment lengths')
plt.xlabel('Alignment length')
plt.ylabel('Number of sequences')
```

Out[0]: <matplotlib.text.Text at 0x251a52e2f98>



In [0]:
```
import seaborn as sns
sns.violinplot(fd['alignment_length'])
```

Out[0]: <matplotlib.axes._subplots.AxesSubplot at 0x251926946d8>



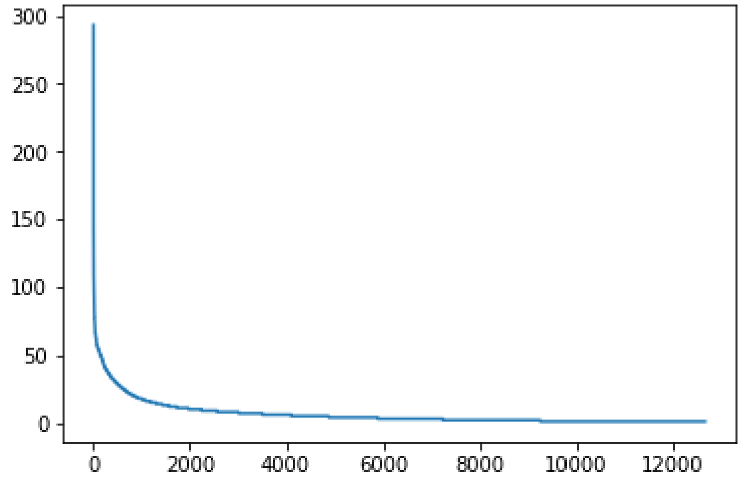
In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
vect1=CountVectorizer()
out1=vect1.fit_transform(totaldataframe['family_id'])
features1=vect1.get_feature_names()
counts1=out1.sum(axis=0)
print('number of unique families are',len(features1))
```

```
number of unique families are 13562
```

In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
vect=CountVectorizer()
out=vect.fit_transform(totaldataframe['family_accession'])
```

In [0]:
```
traincount=traindataframe1['family_accession'].value_counts()
plt.plot(traincount.values)
```

Out[0]:
```
[<matplotlib.lines.Line2D at 0x25199e09978>]
```



In [0]:
```
lengthofsequence=traindataframe['sequence'].apply(lambda x: len(x))
```

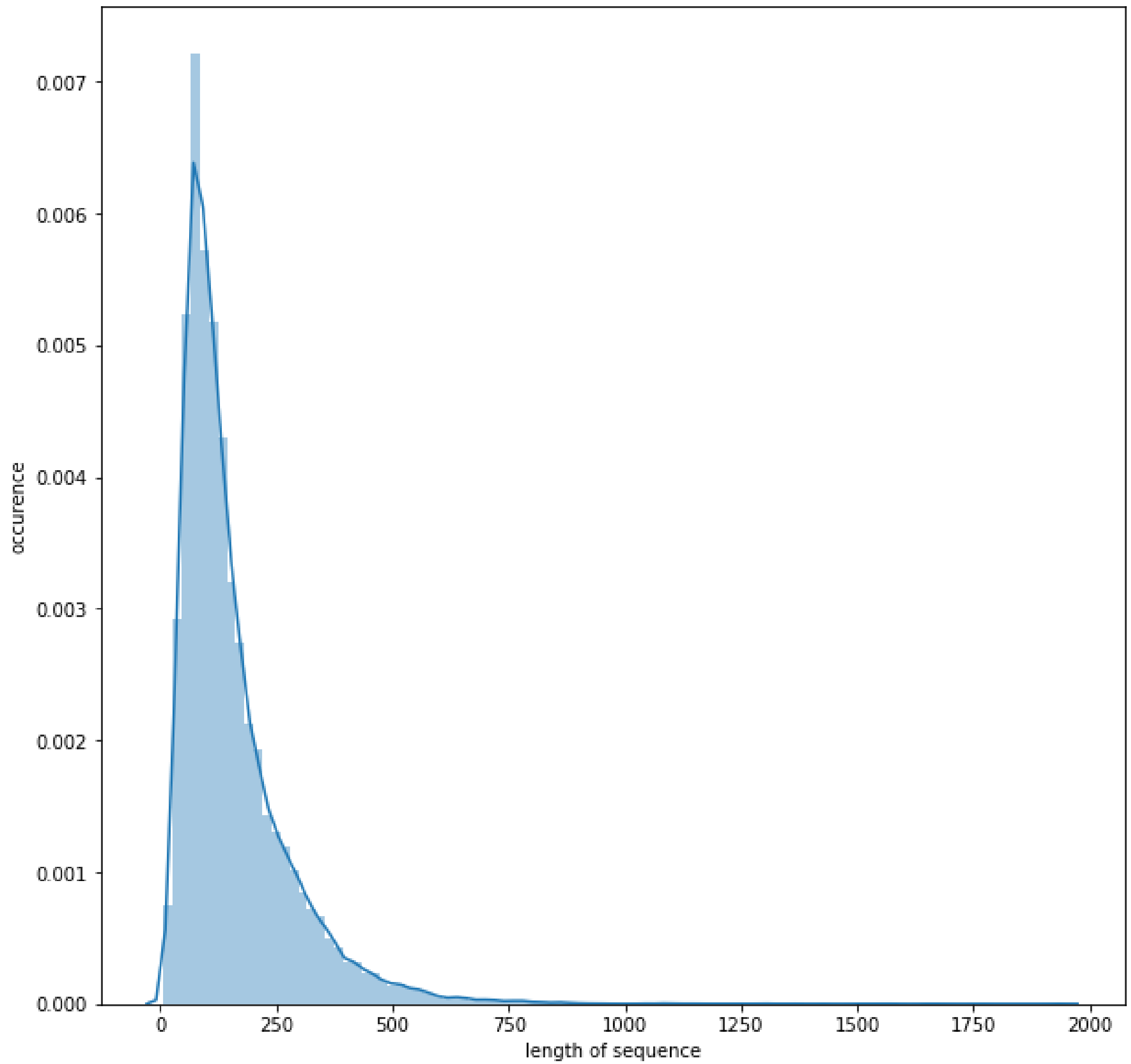
In [0]:
```
print(lengthofsequence.values)
```

```
[169  86  46 ...  65  90  57]
```

In [0]:

```
import seaborn as sns
plt.figure(figsize=(10,10))
sns.distplot(lengthofsequence.values,hist=True,bins=100)
plt.xlabel('length of sequence')
plt.ylabel('occurence')
```

Out[0]: <matplotlib.text.Text at 0x251a2af2c88>



In [0]:
```
import string
def string_vectorizer(strng, alphabet=string.ascii_lowercase):

    vector = [[0 if char != letter else 1 for char in alphabet]
```

```
                    for letter in strng]
    vector1=np.array(vector)
    shapeout=vector1.shape[0]
    diff=200-shapeout
    reshapearray=np.zeros((diff,26),dtype=int)
    lenarray=len(strng)
    finalarray=np.vstack((vector,reshapearray,newarray))
    return finalarray
```

In [0]:
```
trainarray1=[]
for sen in traindataframe['sequence']:
   trainarray1.append(string_vectorizer(sen.lower()))
```

In [0]:
```
testarray1=[]
for sen in testdataframe['sequence']:
    testarray1.append(string_vectorizer(sen.lower()))
```

In [0]:
```
cvarray1=[]
for sen in cvdataframe['sequence']:
    cvarray1.append(string_vectorizer(sen.lower()))
```

In [0]:
```
trainarray1=np.array(trainarray1).reshape(130000,200,26)
```

In [0]:
```
testarray1=np.array(testarray1).reshape(20000,200,26)
cvarray1=np.array(cvarray1).reshape(20000,200,26)
```

In [0]:
```
totaldataframe=pd.concat([traindataframe,testdataframe],axis=0)
```

In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

```
vect=CountVectorizer()
out=vect.fit_transform(totaldataframe['family_accession'])
features=vect.get_feature_names()
counts=out.sum(axis=0)
```

In [0]:
```
from keras.models import Sequential
from keras.layers import Flatten,Activation,Dense,Dropout
from keras.utils import np_utils
```

In [0]:
```
from sklearn.preprocessing import LabelEncoder
label1=LabelEncoder()
label1.fit(totaldataframe['family_accession'])
trainy=label1.transform(traindataframe['family_accession'])
testy=label1.transform(testdataframe['family_accession'])
cvy=label1.transform(cvdataframe['family_accession'])
```

In [0]:
```
# this function is used draw Binary Crossentropy Loss VS No. of epochs
 plot
def plt_dynamic(x, vy, ty):
  plt.figure(figsize=(10,5))
  plt.plot(x, vy, 'b', label="Validation Loss")
  plt.plot(x, ty, 'r', label="Train Loss")
  plt.xlabel('Epochs')
  plt.ylabel('Binary Crossentropy Loss')
  plt.title('\nBinary Crossentropy Loss VS Epochs')
  plt.legend()
  plt.grid()
  plt.show()
```

In [24]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=(21),activation='relu',input_shape=(20
0,26)))
model.add(MaxPooling1D(pool_size=2))
```

```
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.6))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=70,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
W0618 18:56:06.568818 139743238666112 nn_ops.py:4224] Large dropout rat
e: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
W0618 18:56:06.841839 139743238666112 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/70
130000/130000 [==============================] - 121s 933us/step - los
s: 5.7690 - acc: 0.2431 - val_loss: 4.1239 - val_acc: 0.3526
Epoch 2/70
130000/130000 [==============================] - 113s 870us/step - los
s: 2.1282 - acc: 0.6275 - val_loss: 3.1806 - val_acc: 0.4644
Epoch 3/70
130000/130000 [==============================] - 113s 869us/step - los
s: 1.0520 - acc: 0.7813 - val_loss: 1.9398 - val_acc: 0.6472
Epoch 4/70
130000/130000 [==============================] - 113s 870us/step - los
s: 0.6432 - acc: 0.8529 - val_loss: 1.4035 - val_acc: 0.7410
Epoch 5/70
130000/130000 [==============================] - 113s 870us/step - los
s: 0.4576 - acc: 0.8898 - val_loss: 1.2556 - val_acc: 0.7703
Epoch 6/70
```

```
130000/130000 [==============================] - 113s 869us/step - los
s: 0.3547 - acc: 0.9109 - val_loss: 0.8780 - val_acc: 0.8448
Epoch 7/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.2904 - acc: 0.9263 - val_loss: 0.8866 - val_acc: 0.8447
Epoch 8/70
130000/130000 [==============================] - 113s 871us/step - los
s: 0.2575 - acc: 0.9338 - val_loss: 0.8659 - val_acc: 0.8517
Epoch 9/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.2224 - acc: 0.9425 - val_loss: 0.7825 - val_acc: 0.8648
Epoch 10/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.2035 - acc: 0.9464 - val_loss: 0.7975 - val_acc: 0.8652
Epoch 11/70
130000/130000 [==============================] - 113s 871us/step - los
s: 0.1886 - acc: 0.9511 - val_loss: 0.7368 - val_acc: 0.8810
Epoch 12/70
130000/130000 [==============================] - 113s 871us/step - los
s: 0.1727 - acc: 0.9547 - val_loss: 0.7894 - val_acc: 0.8694
Epoch 13/70
130000/130000 [==============================] - 113s 870us/step - los
s: 0.1663 - acc: 0.9571 - val_loss: 0.7784 - val_acc: 0.8756
Epoch 14/70
130000/130000 [==============================] - 113s 867us/step - los
s: 0.1540 - acc: 0.9604 - val_loss: 0.7448 - val_acc: 0.8809
Epoch 15/70
130000/130000 [==============================] - 113s 869us/step - los
s: 0.1495 - acc: 0.9616 - val_loss: 0.7589 - val_acc: 0.8790
Epoch 16/70
130000/130000 [==============================] - 113s 867us/step - los
s: 0.1390 - acc: 0.9637 - val_loss: 0.7803 - val_acc: 0.8771
Epoch 17/70
130000/130000 [==============================] - 113s 867us/step - los
s: 0.1355 - acc: 0.9648 - val_loss: 0.7963 - val_acc: 0.8770
Epoch 18/70
130000/130000 [==============================] - 113s 867us/step - los
s: 0.1337 - acc: 0.9662 - val_loss: 0.7456 - val_acc: 0.8835
Epoch 19/70
130000/130000 [==============================] - 113s 866us/step - los
```

```
s: 0.1237 - acc: 0.9687 - val_loss: 0.8482 - val_acc: 0.8694
Epoch 20/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.1180 - acc: 0.9699 - val_loss: 0.8330 - val_acc: 0.8685
Epoch 21/70
130000/130000 [==============================] - 113s 870us/step - los
s: 0.1180 - acc: 0.9696 - val_loss: 0.8984 - val_acc: 0.8591
Epoch 22/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.1113 - acc: 0.9716 - val_loss: 0.8394 - val_acc: 0.8720
Epoch 23/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.1076 - acc: 0.9731 - val_loss: 0.8682 - val_acc: 0.8674
Epoch 24/70
130000/130000 [==============================] - 114s 874us/step - los
s: 0.1038 - acc: 0.9737 - val_loss: 0.7545 - val_acc: 0.8857
Epoch 25/70
130000/130000 [==============================] - 113s 868us/step - los
s: 0.1023 - acc: 0.9740 - val_loss: 0.7933 - val_acc: 0.8801
Epoch 26/70
130000/130000 [==============================] - 113s 873us/step - los
s: 0.0984 - acc: 0.9751 - val_loss: 0.8213 - val_acc: 0.8781
Epoch 27/70
130000/130000 [==============================] - 113s 872us/step - los
s: 0.0970 - acc: 0.9758 - val_loss: 0.8161 - val_acc: 0.8785
Epoch 28/70
130000/130000 [==============================] - 113s 869us/step - los
s: 0.0951 - acc: 0.9758 - val_loss: 0.8145 - val_acc: 0.8796
Epoch 29/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0908 - acc: 0.9766 - val_loss: 0.8507 - val_acc: 0.8701
Epoch 30/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0907 - acc: 0.9771 - val_loss: 0.8163 - val_acc: 0.8780
Epoch 31/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0864 - acc: 0.9782 - val_loss: 0.8379 - val_acc: 0.8741
Epoch 32/70
130000/130000 [==============================] - 113s 865us/step - los
```

```
s: 0.0850 - acc: 0.9784 - val_loss: 0.7650 - val_acc: 0.8848
Epoch 33/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0808 - acc: 0.9793 - val_loss: 0.8455 - val_acc: 0.8759
Epoch 34/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0789 - acc: 0.9799 - val_loss: 0.7777 - val_acc: 0.8846
Epoch 35/70
130000/130000 [==============================] - 113s 865us/step - los
s: 0.0778 - acc: 0.9804 - val_loss: 0.8512 - val_acc: 0.8707
Epoch 36/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0760 - acc: 0.9809 - val_loss: 0.7819 - val_acc: 0.8850
Epoch 37/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0748 - acc: 0.9811 - val_loss: 0.8929 - val_acc: 0.8682
Epoch 38/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0755 - acc: 0.9810 - val_loss: 0.8348 - val_acc: 0.8783
Epoch 39/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0723 - acc: 0.9816 - val_loss: 0.8440 - val_acc: 0.8764
Epoch 40/70
130000/130000 [==============================] - 113s 867us/step - los
s: 0.0711 - acc: 0.9820 - val_loss: 0.8757 - val_acc: 0.8721
Epoch 41/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0699 - acc: 0.9827 - val_loss: 0.8580 - val_acc: 0.8770
Epoch 42/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0693 - acc: 0.9823 - val_loss: 0.8636 - val_acc: 0.8764
Epoch 43/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0662 - acc: 0.9835 - val_loss: 0.8575 - val_acc: 0.8759
Epoch 44/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0658 - acc: 0.9833 - val_loss: 0.7781 - val_acc: 0.8868
Epoch 45/70
130000/130000 [==============================] - 113s 866us/step - los
```

```
s: 0.0644 - acc: 0.9833 - val_loss: 0.8697 - val_acc: 0.8740
Epoch 46/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0621 - acc: 0.9840 - val_loss: 0.8078 - val_acc: 0.8831
Epoch 47/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0592 - acc: 0.9848 - val_loss: 0.7940 - val_acc: 0.8854
Epoch 48/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0597 - acc: 0.9847 - val_loss: 0.7380 - val_acc: 0.8949
Epoch 49/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0624 - acc: 0.9842 - val_loss: 0.8336 - val_acc: 0.8794
Epoch 50/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0579 - acc: 0.9853 - val_loss: 0.7954 - val_acc: 0.8854
Epoch 51/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0574 - acc: 0.9853 - val_loss: 0.8689 - val_acc: 0.8755
Epoch 52/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0566 - acc: 0.9857 - val_loss: 0.8988 - val_acc: 0.8711
Epoch 53/70
130000/130000 [==============================] - 113s 865us/step - los
s: 0.0555 - acc: 0.9857 - val_loss: 0.8346 - val_acc: 0.8800
Epoch 54/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0530 - acc: 0.9862 - val_loss: 1.0208 - val_acc: 0.8511
Epoch 55/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0526 - acc: 0.9864 - val_loss: 0.7981 - val_acc: 0.8838
Epoch 56/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0525 - acc: 0.9864 - val_loss: 0.8352 - val_acc: 0.8786
Epoch 57/70
130000/130000 [==============================] - 113s 865us/step - los
s: 0.0524 - acc: 0.9867 - val_loss: 0.8263 - val_acc: 0.8814
Epoch 58/70
130000/130000 [==============================] - 112s 864us/step - los
```

```
s: 0.0519 - acc: 0.9862 - val_loss: 0.9964 - val_acc: 0.8572
Epoch 59/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0525 - acc: 0.9867 - val_loss: 0.7538 - val_acc: 0.8932
Epoch 60/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0501 - acc: 0.9872 - val_loss: 0.8136 - val_acc: 0.8853
Epoch 61/70
130000/130000 [==============================] - 113s 865us/step - los
s: 0.0504 - acc: 0.9870 - val_loss: 0.8633 - val_acc: 0.8766
Epoch 62/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0490 - acc: 0.9877 - val_loss: 0.7601 - val_acc: 0.8905
Epoch 63/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.0474 - acc: 0.9877 - val_loss: 0.8829 - val_acc: 0.8750
Epoch 64/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0489 - acc: 0.9876 - val_loss: 0.7767 - val_acc: 0.8882
Epoch 65/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0490 - acc: 0.9874 - val_loss: 0.8346 - val_acc: 0.8799
Epoch 66/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0464 - acc: 0.9882 - val_loss: 0.8580 - val_acc: 0.8760
Epoch 67/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0457 - acc: 0.9885 - val_loss: 0.8592 - val_acc: 0.8777
Epoch 68/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0467 - acc: 0.9882 - val_loss: 0.8854 - val_acc: 0.8767
Epoch 69/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0470 - acc: 0.9879 - val_loss: 0.7671 - val_acc: 0.8929
Epoch 70/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0448 - acc: 0.9887 - val_loss: 0.7847 - val_acc: 0.8873
```

In [23]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM
model=Sequential()
model.add(Conv1D(300,kernel_size=(21),activation='relu',input_shape=(20
0,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.6))
model.add(BatchNormalization())
model.add(LSTM(300,dropout=0.4,return_sequences=True))
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=70,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
W0619 05:54:02.029180 139960010635136 nn_ops.py:4224] Large dropout rat
e: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/70
130000/130000 [==============================] - 118s 904us/step - los
s: 4.9852 - acc: 0.2799 - val_loss: 1.9962 - val_acc: 0.6245
Epoch 2/70
130000/130000 [==============================] - 114s 880us/step - los
s: 1.6706 - acc: 0.6640 - val_loss: 1.0969 - val_acc: 0.7780
Epoch 3/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.7416 - acc: 0.8196 - val_loss: 0.9559 - val_acc: 0.8147
Epoch 4/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.4538 - acc: 0.8806 - val_loss: 0.9333 - val_acc: 0.8303
```

```
s: 0.4538 - acc: 0.8806 - val_loss: 0.9333 - val_acc: 0.8303

Epoch 5/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.3307 - acc: 0.9079 - val_loss: 0.8951 - val_acc: 0.8394
Epoch 6/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.2645 - acc: 0.9257 - val_loss: 0.8944 - val_acc: 0.8469
Epoch 7/70
130000/130000 [==============================] - 114s 878us/step - los
s: 0.2243 - acc: 0.9352 - val_loss: 0.8788 - val_acc: 0.8512
Epoch 8/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.1914 - acc: 0.9447 - val_loss: 0.8882 - val_acc: 0.8536
Epoch 9/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.1705 - acc: 0.9497 - val_loss: 0.8974 - val_acc: 0.8546
Epoch 10/70
130000/130000 [==============================] - 114s 877us/step - los
s: 0.1536 - acc: 0.9552 - val_loss: 0.8856 - val_acc: 0.8614
Epoch 11/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.1377 - acc: 0.9590 - val_loss: 0.8921 - val_acc: 0.8627
Epoch 12/70
130000/130000 [==============================] - 114s 877us/step - los
s: 0.1257 - acc: 0.9629 - val_loss: 0.9089 - val_acc: 0.8601
Epoch 13/70
130000/130000 [==============================] - 114s 876us/step - los
s: 0.1162 - acc: 0.9655 - val_loss: 0.9008 - val_acc: 0.8655
Epoch 14/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.1102 - acc: 0.9676 - val_loss: 0.9064 - val_acc: 0.8652
Epoch 15/70
130000/130000 [==============================] - 114s 878us/step - los
s: 0.1007 - acc: 0.9701 - val_loss: 0.9010 - val_acc: 0.8659
Epoch 16/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.0981 - acc: 0.9709 - val_loss: 0.9005 - val_acc: 0.8691
Epoch 17/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0889 - acc: 0.9737 - val_loss: 0.9120 - val_acc: 0.8660
```

```
s: 0.0889 - acc: 0.9737 - val_loss: 0.9129 - val_acc: 0.8660

Epoch 18/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0852 - acc: 0.9748 - val_loss: 0.9198 - val_acc: 0.8697
Epoch 19/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.0837 - acc: 0.9756 - val_loss: 0.9320 - val_acc: 0.8696
Epoch 20/70
130000/130000 [==============================] - 114s 875us/step - los
s: 0.0798 - acc: 0.9764 - val_loss: 0.9294 - val_acc: 0.8719
Epoch 21/70
130000/130000 [==============================] - 114s 876us/step - los
s: 0.0751 - acc: 0.9778 - val_loss: 0.9433 - val_acc: 0.8716
Epoch 22/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.0736 - acc: 0.9779 - val_loss: 0.9338 - val_acc: 0.8727
Epoch 23/70
130000/130000 [==============================] - 114s 875us/step - los
s: 0.0705 - acc: 0.9793 - val_loss: 0.9497 - val_acc: 0.8718
Epoch 24/70
130000/130000 [==============================] - 114s 876us/step - los
s: 0.0670 - acc: 0.9799 - val_loss: 0.9546 - val_acc: 0.8720
Epoch 25/70
130000/130000 [==============================] - 114s 877us/step - los
s: 0.0644 - acc: 0.9809 - val_loss: 0.9439 - val_acc: 0.8721
Epoch 26/70
130000/130000 [==============================] - 114s 876us/step - los
s: 0.0614 - acc: 0.9820 - val_loss: 0.9518 - val_acc: 0.8734
Epoch 27/70
130000/130000 [==============================] - 114s 876us/step - los
s: 0.0600 - acc: 0.9821 - val_loss: 0.9610 - val_acc: 0.8721
Epoch 28/70
130000/130000 [==============================] - 114s 875us/step - los
s: 0.0630 - acc: 0.9817 - val_loss: 0.9718 - val_acc: 0.8720
Epoch 29/70
130000/130000 [==============================] - 114s 877us/step - los
s: 0.0584 - acc: 0.9830 - val_loss: 0.9599 - val_acc: 0.8741
Epoch 30/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0548 - acc: 0.9841 - val_loss: 0.9793 - val_acc: 0.8723
```

```
s: 0.0548 - acc: 0.9841 - val_loss: 0.9793 - val_acc: 0.8723

Epoch 31/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0545 - acc: 0.9844 - val_loss: 0.9744 - val_acc: 0.8738
Epoch 32/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0547 - acc: 0.9840 - val_loss: 0.9775 - val_acc: 0.8730
Epoch 33/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0526 - acc: 0.9846 - val_loss: 0.9838 - val_acc: 0.8720
Epoch 34/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0527 - acc: 0.9847 - val_loss: 0.9818 - val_acc: 0.8726
Epoch 35/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0518 - acc: 0.9850 - val_loss: 0.9866 - val_acc: 0.8749
Epoch 36/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0501 - acc: 0.9855 - val_loss: 0.9820 - val_acc: 0.8739
Epoch 37/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0499 - acc: 0.9856 - val_loss: 0.9883 - val_acc: 0.8750
Epoch 38/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0464 - acc: 0.9864 - val_loss: 1.0078 - val_acc: 0.8733
Epoch 39/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0475 - acc: 0.9864 - val_loss: 0.9859 - val_acc: 0.8753
Epoch 40/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0457 - acc: 0.9868 - val_loss: 0.9909 - val_acc: 0.8761
Epoch 41/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0475 - acc: 0.9865 - val_loss: 0.9938 - val_acc: 0.8768
Epoch 42/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0453 - acc: 0.9872 - val_loss: 0.9964 - val_acc: 0.8764
Epoch 43/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0450 - acc: 0.9872 - val_loss: 0.9944 - val_acc: 0.8760
```

```
s: 0.0450 - acc: 0.9872 - val_loss: 0.9944 - val_acc: 0.8760

Epoch 44/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0420 - acc: 0.9881 - val_loss: 0.9966 - val_acc: 0.8755
Epoch 45/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0408 - acc: 0.9886 - val_loss: 0.9931 - val_acc: 0.8768
Epoch 46/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0418 - acc: 0.9880 - val_loss: 0.9936 - val_acc: 0.8774
Epoch 47/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0388 - acc: 0.9888 - val_loss: 1.0055 - val_acc: 0.8751
Epoch 48/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0414 - acc: 0.9884 - val_loss: 0.9978 - val_acc: 0.8783
Epoch 49/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0387 - acc: 0.9893 - val_loss: 1.0141 - val_acc: 0.8774
Epoch 50/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0405 - acc: 0.9888 - val_loss: 1.0102 - val_acc: 0.8760
Epoch 51/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0400 - acc: 0.9889 - val_loss: 1.0148 - val_acc: 0.8767
Epoch 52/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0371 - acc: 0.9895 - val_loss: 1.0072 - val_acc: 0.8789
Epoch 53/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0375 - acc: 0.9896 - val_loss: 1.0324 - val_acc: 0.8762
Epoch 54/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0397 - acc: 0.9888 - val_loss: 1.0212 - val_acc: 0.8770
Epoch 55/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0376 - acc: 0.9896 - val_loss: 1.0132 - val_acc: 0.8767
Epoch 56/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0379 - acc: 0.9893 - val_loss: 1.0211 - val_acc: 0.8780
```

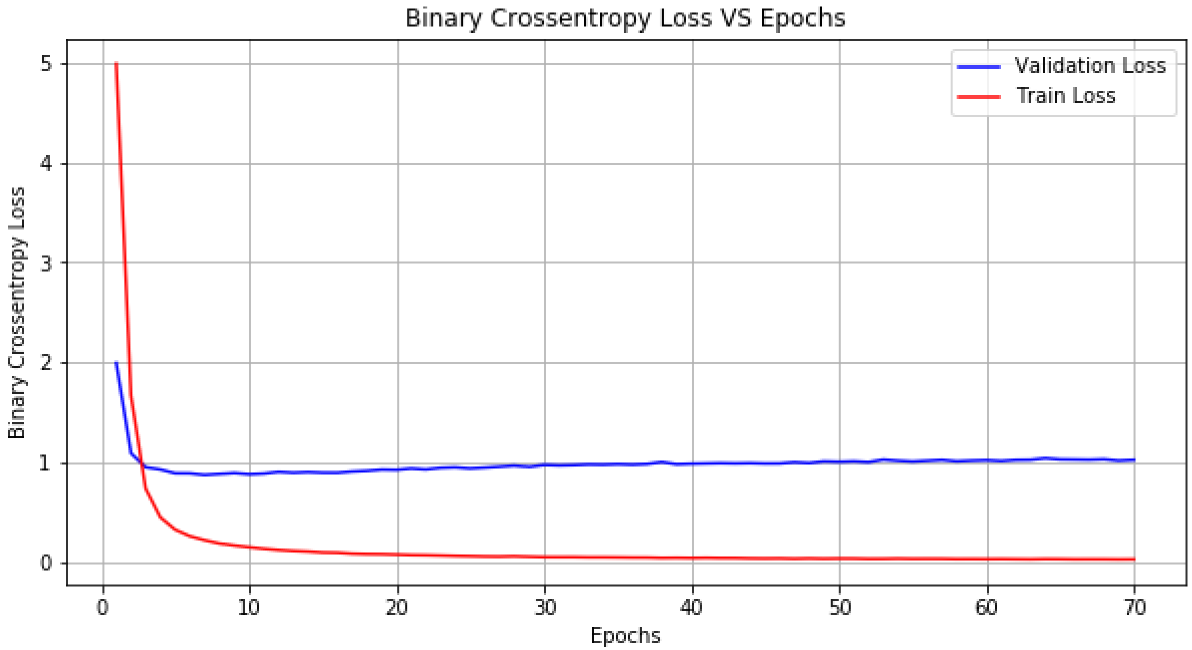
```
s: 0.0379 - acc: 0.9893 - val_loss: 1.0211 - val_acc: 0.8780

Epoch 57/70
130000/130000 [==============================] - 115s 884us/step - los
s: 0.0375 - acc: 0.9895 - val_loss: 1.0294 - val_acc: 0.8768
Epoch 58/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0353 - acc: 0.9898 - val_loss: 1.0155 - val_acc: 0.8765
Epoch 59/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0355 - acc: 0.9902 - val_loss: 1.0221 - val_acc: 0.8764
Epoch 60/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0349 - acc: 0.9905 - val_loss: 1.0265 - val_acc: 0.8779
Epoch 61/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0355 - acc: 0.9902 - val_loss: 1.0188 - val_acc: 0.8773
Epoch 62/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0344 - acc: 0.9906 - val_loss: 1.0290 - val_acc: 0.8767
Epoch 63/70
130000/130000 [==============================] - 115s 882us/step - los
s: 0.0333 - acc: 0.9909 - val_loss: 1.0305 - val_acc: 0.8785
Epoch 64/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0352 - acc: 0.9905 - val_loss: 1.0466 - val_acc: 0.8743
Epoch 65/70
130000/130000 [==============================] - 115s 883us/step - los
s: 0.0347 - acc: 0.9904 - val_loss: 1.0349 - val_acc: 0.8759
Epoch 66/70
130000/130000 [==============================] - 114s 879us/step - los
s: 0.0330 - acc: 0.9911 - val_loss: 1.0333 - val_acc: 0.8774
Epoch 67/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0332 - acc: 0.9908 - val_loss: 1.0315 - val_acc: 0.8770
Epoch 68/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0330 - acc: 0.9910 - val_loss: 1.0360 - val_acc: 0.8773
Epoch 69/70
130000/130000 [==============================] - 115s 881us/step - los
s: 0.0320 - acc: 0.9916 - val_loss: 1.0224 - val_acc: 0.8786
```

```
s: 0.0320 - acc: 0.9916 - val_loss: 1.0224 - val_acc: 0.8786

Epoch 70/70
130000/130000 [==============================] - 114s 880us/step - los
s: 0.0325 - acc: 0.9912 - val_loss: 1.0297 - val_acc: 0.8781
```

In [24]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,71))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 87.81%
```



In [25]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM
model=Sequential()
model.add(Conv1D(300,kernel_size=(26),activation='relu',input_shape=(200,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.7))
model.add(BatchNormalization())
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.7))
model.add(BatchNormalization())
model.add(LSTM(500,dropout=0.4,return_sequences=True))
model.add(BatchNormalization())
```

```
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=70,validation_data=[testarr
ay1,testy],batch_size=256)
```

```
W0619 08:12:08.018236 139960010635136 nn_ops.py:4224] Large dropout rat
e: 0.7 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
W0619 08:12:08.141388 139960010635136 nn_ops.py:4224] Large dropout rat
e: 0.7 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/70
130000/130000 [==============================] - 116s 894us/step - los
s: 6.0865 - acc: 0.1738 - val_loss: 2.9649 - val_acc: 0.4763
Epoch 2/70
130000/130000 [==============================] - 112s 862us/step - los
s: 2.6262 - acc: 0.5068 - val_loss: 1.6330 - val_acc: 0.6788
Epoch 3/70
130000/130000 [==============================] - 112s 864us/step - los
s: 1.4468 - acc: 0.6791 - val_loss: 1.2906 - val_acc: 0.7484
Epoch 4/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.9376 - acc: 0.7732 - val_loss: 1.1468 - val_acc: 0.7808
Epoch 5/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.6741 - acc: 0.8260 - val_loss: 1.1004 - val_acc: 0.7991
Epoch 6/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.5243 - acc: 0.8610 - val_loss: 1.0694 - val_acc: 0.8101
Epoch 7/70
130000/130000 [==============================] - 113s 866us/step - los
s: 0.4319 - acc: 0.8832 - val_loss: 1.0478 - val_acc: 0.8183
Epoch 8/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.3615 - acc: 0.8987 - val_loss: 1.0683 - val_acc: 0.8242
Epoch 9/70
130000/130000 [==============================] - 112s 865us/step - los
```

```
130000/130000 [==============================] - 112s 865us/step - los
s: 0.3156 - acc: 0.9107 - val_loss: 1.0538 - val_acc: 0.8271
Epoch 10/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.2759 - acc: 0.9220 - val_loss: 1.0504 - val_acc: 0.8314
Epoch 11/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.2455 - acc: 0.9296 - val_loss: 1.0662 - val_acc: 0.8348
Epoch 12/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.2253 - acc: 0.9351 - val_loss: 1.0562 - val_acc: 0.8360
Epoch 13/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.2059 - acc: 0.9404 - val_loss: 1.0502 - val_acc: 0.8389
Epoch 14/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.1910 - acc: 0.9442 - val_loss: 1.0618 - val_acc: 0.8393
Epoch 15/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1692 - acc: 0.9512 - val_loss: 1.0514 - val_acc: 0.8467
Epoch 16/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1658 - acc: 0.9523 - val_loss: 1.0507 - val_acc: 0.8464
Epoch 17/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.1497 - acc: 0.9559 - val_loss: 1.0728 - val_acc: 0.8437
Epoch 18/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1421 - acc: 0.9586 - val_loss: 1.0778 - val_acc: 0.8433
Epoch 19/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.1358 - acc: 0.9604 - val_loss: 1.0748 - val_acc: 0.8454
Epoch 20/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1287 - acc: 0.9622 - val_loss: 1.0761 - val_acc: 0.8488
Epoch 21/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1239 - acc: 0.9639 - val_loss: 1.0911 - val_acc: 0.8467
Epoch 22/70
130000/130000 [==============================] - 112s 863us/step - los
```

```
s: 0.1145 - acc: 0.9669 - val_loss: 1.0812 - val_acc: 0.8502
Epoch 23/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.1107 - acc: 0.9681 - val_loss: 1.0889 - val_acc: 0.8484
Epoch 24/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.1077 - acc: 0.9683 - val_loss: 1.0920 - val_acc: 0.8486
Epoch 25/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.1056 - acc: 0.9692 - val_loss: 1.0778 - val_acc: 0.8505
Epoch 26/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0990 - acc: 0.9709 - val_loss: 1.0583 - val_acc: 0.8531
Epoch 27/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0918 - acc: 0.9732 - val_loss: 1.0762 - val_acc: 0.8525
Epoch 28/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0900 - acc: 0.9738 - val_loss: 1.0879 - val_acc: 0.8531
Epoch 29/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0870 - acc: 0.9744 - val_loss: 1.0883 - val_acc: 0.8520
Epoch 30/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0848 - acc: 0.9748 - val_loss: 1.0937 - val_acc: 0.8540
Epoch 31/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0816 - acc: 0.9762 - val_loss: 1.1076 - val_acc: 0.8522
Epoch 32/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0781 - acc: 0.9770 - val_loss: 1.0945 - val_acc: 0.8540
Epoch 33/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0756 - acc: 0.9777 - val_loss: 1.1066 - val_acc: 0.8551
Epoch 34/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0736 - acc: 0.9787 - val_loss: 1.1017 - val_acc: 0.8542
Epoch 35/70
130000/130000 [==============================] - 112s 862us/step - los
```

```
s: 0.0692 - acc: 0.9798 - val_loss: 1.0889 - val_acc: 0.8565
Epoch 36/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0714 - acc: 0.9790 - val_loss: 1.0994 - val_acc: 0.8564
Epoch 37/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0669 - acc: 0.9806 - val_loss: 1.1082 - val_acc: 0.8535
Epoch 38/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0644 - acc: 0.9809 - val_loss: 1.1136 - val_acc: 0.8553
Epoch 39/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0633 - acc: 0.9818 - val_loss: 1.1184 - val_acc: 0.8539
Epoch 40/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0625 - acc: 0.9819 - val_loss: 1.1116 - val_acc: 0.8560
Epoch 41/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0613 - acc: 0.9822 - val_loss: 1.1049 - val_acc: 0.8569
Epoch 42/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0585 - acc: 0.9829 - val_loss: 1.1242 - val_acc: 0.8568
Epoch 43/70
130000/130000 [==============================] - 112s 865us/step - los
s: 0.0573 - acc: 0.9835 - val_loss: 1.1039 - val_acc: 0.8563
Epoch 44/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0559 - acc: 0.9837 - val_loss: 1.1024 - val_acc: 0.8577
Epoch 45/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0551 - acc: 0.9837 - val_loss: 1.1113 - val_acc: 0.8573
Epoch 46/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0546 - acc: 0.9841 - val_loss: 1.1059 - val_acc: 0.8549
Epoch 47/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0535 - acc: 0.9846 - val_loss: 1.1075 - val_acc: 0.8538
Epoch 48/70
130000/130000 [==============================] - 112s 863us/step - los
```

```
s: 0.0534 - acc: 0.9845 - val_loss: 1.1074 - val_acc: 0.8593
Epoch 49/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0494 - acc: 0.9859 - val_loss: 1.0916 - val_acc: 0.8588
Epoch 50/70
130000/130000 [==============================] - 112s 859us/step - los
s: 0.0510 - acc: 0.9851 - val_loss: 1.1015 - val_acc: 0.8562
Epoch 51/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0468 - acc: 0.9864 - val_loss: 1.1149 - val_acc: 0.8555
Epoch 52/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0487 - acc: 0.9859 - val_loss: 1.0895 - val_acc: 0.8601
Epoch 53/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0474 - acc: 0.9864 - val_loss: 1.1081 - val_acc: 0.8601
Epoch 54/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0458 - acc: 0.9865 - val_loss: 1.1379 - val_acc: 0.8556
Epoch 55/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0443 - acc: 0.9873 - val_loss: 1.1161 - val_acc: 0.8578
Epoch 56/70
130000/130000 [==============================] - 112s 864us/step - los
s: 0.0459 - acc: 0.9868 - val_loss: 1.1063 - val_acc: 0.8588
Epoch 57/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0441 - acc: 0.9872 - val_loss: 1.1033 - val_acc: 0.8586
Epoch 58/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0437 - acc: 0.9875 - val_loss: 1.1107 - val_acc: 0.8595
Epoch 59/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0416 - acc: 0.9879 - val_loss: 1.1113 - val_acc: 0.8606
Epoch 60/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0416 - acc: 0.9879 - val_loss: 1.1092 - val_acc: 0.8613
Epoch 61/70
130000/130000 [==============================] - 112s 862us/step - los
```

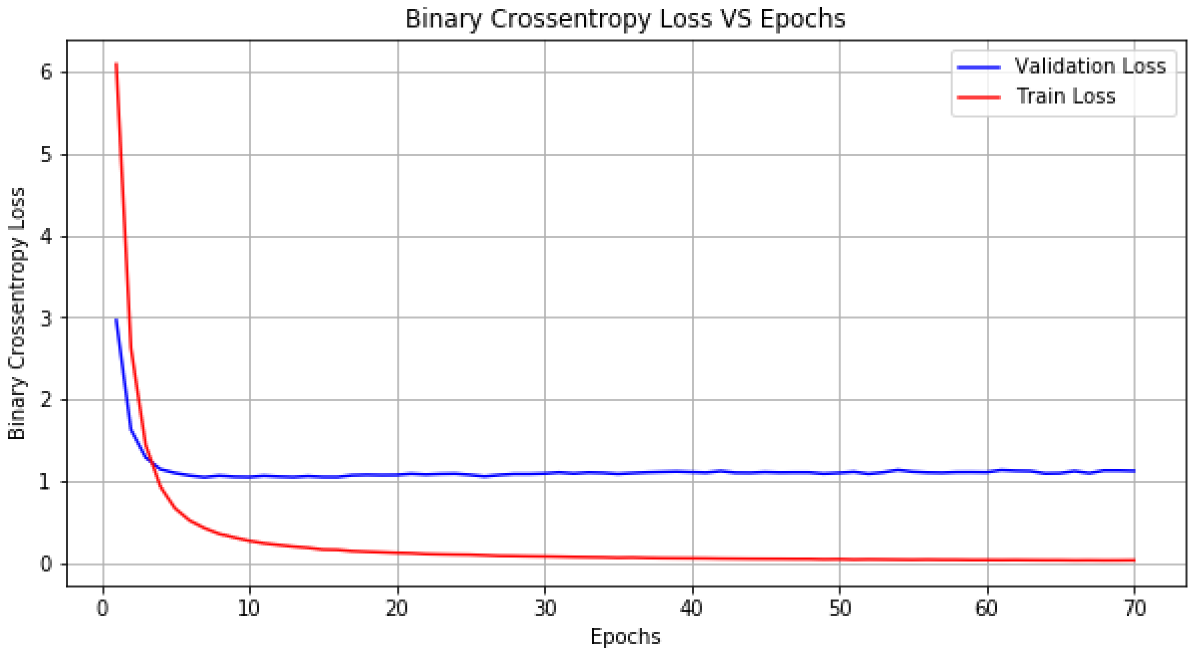
```
s: 0.0407 - acc: 0.9884 - val_loss: 1.1364 - val_acc: 0.8575
Epoch 62/70
130000/130000 [==============================] - 112s 863us/step - los
s: 0.0413 - acc: 0.9880 - val_loss: 1.1274 - val_acc: 0.8563
Epoch 63/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0401 - acc: 0.9882 - val_loss: 1.1242 - val_acc: 0.8586
Epoch 64/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0392 - acc: 0.9888 - val_loss: 1.0945 - val_acc: 0.8604
Epoch 65/70
130000/130000 [==============================] - 112s 862us/step - los
s: 0.0390 - acc: 0.9887 - val_loss: 1.0978 - val_acc: 0.8631
Epoch 66/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0372 - acc: 0.9893 - val_loss: 1.1241 - val_acc: 0.8603
Epoch 67/70
130000/130000 [==============================] - 112s 860us/step - los
s: 0.0378 - acc: 0.9892 - val_loss: 1.0979 - val_acc: 0.8629
Epoch 68/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0365 - acc: 0.9894 - val_loss: 1.1314 - val_acc: 0.8572
Epoch 69/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0368 - acc: 0.9892 - val_loss: 1.1309 - val_acc: 0.8585
Epoch 70/70
130000/130000 [==============================] - 112s 861us/step - los
s: 0.0379 - acc: 0.9888 - val_loss: 1.1260 - val_acc: 0.8600
```

In [26]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
```

```
# list of epoch numbers
x = list(range(1,71))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 86.00%
```



In [25]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
```

```
from keras.layers import LSTM,Bidirectional
model=Sequential()
model.add(Dense(256,activation='relu',input_shape=(200,26)))
model.add(Dropout(0.7))
model.add(BatchNormalization())
model.add(Bidirectional(LSTM(26,dropout=0.5,return_sequences=True)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=5,validation_data=[testarra
y1,testy],batch_size=128)
```

```
W0619 20:24:43.532114 140629908809600 nn_ops.py:4224] Large dropout rat
e: 0.7 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
```

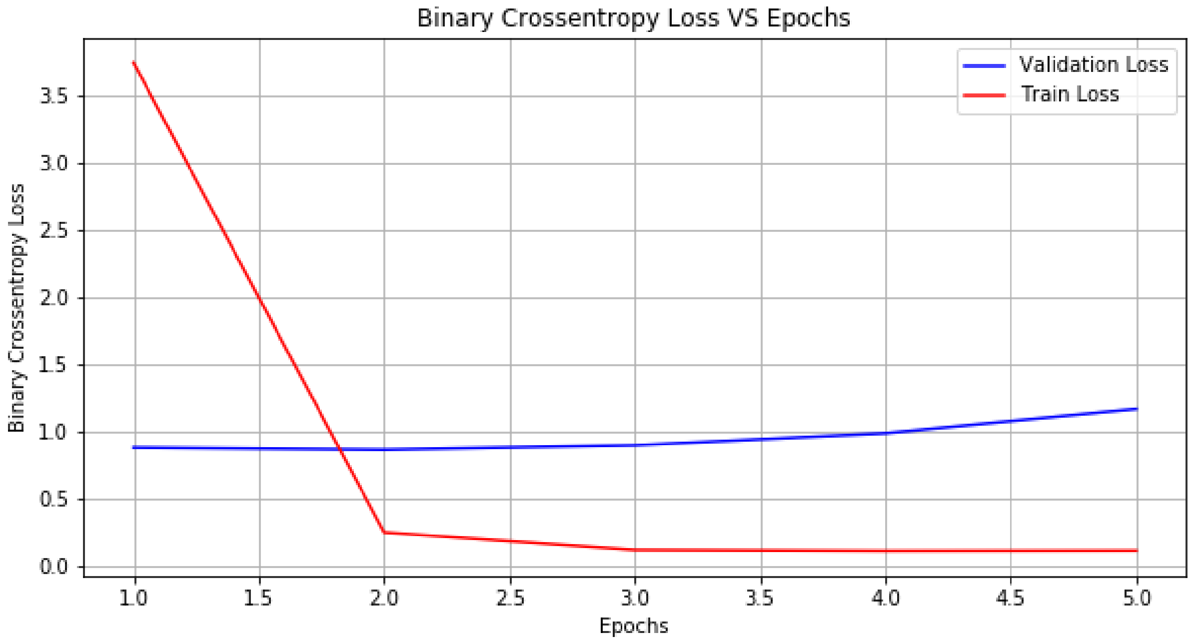
```
Train on 130000 samples, validate on 20000 samples
Epoch 1/5
130000/130000 [==============================] - 801s 6ms/step - loss:
3.7448 - acc: 0.5683 - val_loss: 0.8746 - val_acc: 0.8793
Epoch 2/5
130000/130000 [==============================] - 788s 6ms/step - loss:
0.2431 - acc: 0.9697 - val_loss: 0.8588 - val_acc: 0.8860
Epoch 3/5
130000/130000 [==============================] - 786s 6ms/step - loss:
0.1112 - acc: 0.9911 - val_loss: 0.8922 - val_acc: 0.8881
Epoch 4/5
130000/130000 [==============================] - 788s 6ms/step - loss:
0.1043 - acc: 0.9925 - val_loss: 0.9823 - val_acc: 0.8818
Epoch 5/5
130000/130000 [==============================] - 792s 6ms/step - loss:
0.1062 - acc: 0.9915 - val_loss: 1.1620 - val_acc: 0.8638
```

In [27]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```python
# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,6))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 86.38%
```



In [0]:

```python
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=10,activation='relu',input_shape=(200,
26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(250,20,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=25,validation_data=[testarr
ay1,testy],batch_size=256)
```

```
Train on 115275 samples, validate on 21113 samples
Epoch 1/25
115275/115275 [==============================] - 67s 578us/step - loss:
5.3151 - acc: 0.2801 - val_loss: 2.2157 - val_acc: 0.6586
Epoch 2/25
115275/115275 [==============================] - 65s 564us/step - loss:
1.4764 - acc: 0.7387 - val_loss: 1.1932 - val_acc: 0.8027
Epoch 3/25
115275/115275 [==============================] - 65s 563us/step - loss:
0.4694 - acc: 0.8962 - val_loss: 1.0867 - val_acc: 0.8221
Epoch 4/25
115275/115275 [==============================] - 65s 565us/step - loss:
0.2276 - acc: 0.9449 - val_loss: 1.1101 - val_acc: 0.8257
Epoch 5/25
115275/115275 [==============================] - 65s 564us/step - loss:
0.1566 - acc: 0.9601 - val_loss: 1.0820 - val_acc: 0.8377
Epoch 6/25
115275/115275 [==============================] - 65s 563us/step - loss:
```

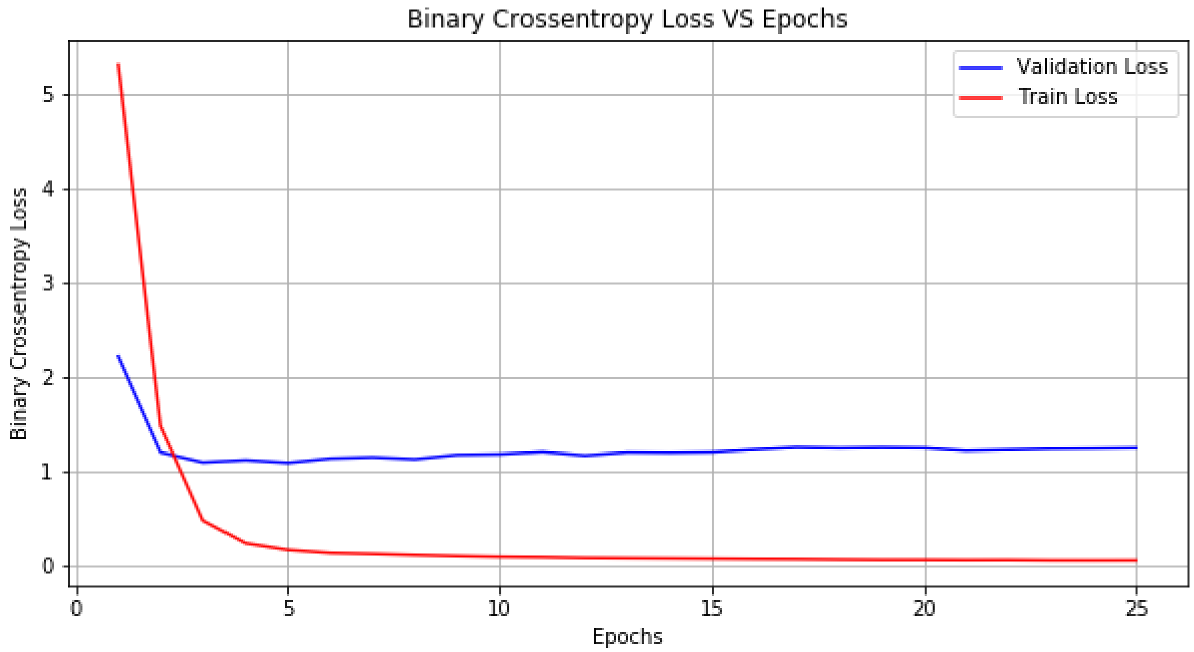
```
0.1241 - acc: 0.9668 - val_loss: 1.1276 - val_acc: 0.8399
Epoch 7/25
115275/115275 [==============================] - 65s 563us/step - loss: 
0.1139 - acc: 0.9698 - val_loss: 1.1402 - val_acc: 0.8387
Epoch 8/25
115275/115275 [==============================] - 65s 563us/step - loss: 
0.1019 - acc: 0.9722 - val_loss: 1.1203 - val_acc: 0.8421
Epoch 9/25
115275/115275 [==============================] - 65s 563us/step - loss: 
0.0911 - acc: 0.9757 - val_loss: 1.1661 - val_acc: 0.8440
Epoch 10/25
115275/115275 [==============================] - 63s 549us/step - loss: 
0.0830 - acc: 0.9782 - val_loss: 1.1724 - val_acc: 0.8444
Epoch 11/25
115275/115275 [==============================] - 62s 541us/step - loss: 
0.0788 - acc: 0.9793 - val_loss: 1.1980 - val_acc: 0.8418
Epoch 12/25
115275/115275 [==============================] - 62s 541us/step - loss: 
0.0710 - acc: 0.9816 - val_loss: 1.1586 - val_acc: 0.8520
Epoch 13/25
115275/115275 [==============================] - 62s 540us/step - loss: 
0.0683 - acc: 0.9825 - val_loss: 1.1932 - val_acc: 0.8500
Epoch 14/25
115275/115275 [==============================] - 62s 541us/step - loss: 
0.0657 - acc: 0.9831 - val_loss: 1.1901 - val_acc: 0.8523
Epoch 15/25
115275/115275 [==============================] - 62s 539us/step - loss: 
0.0627 - acc: 0.9841 - val_loss: 1.1953 - val_acc: 0.8503
Epoch 16/25
115275/115275 [==============================] - 62s 539us/step - loss: 
0.0597 - acc: 0.9849 - val_loss: 1.2241 - val_acc: 0.8511
Epoch 17/25
115275/115275 [==============================] - 62s 538us/step - loss: 
0.0577 - acc: 0.9855 - val_loss: 1.2474 - val_acc: 0.8504
Epoch 18/25
115275/115275 [==============================] - 62s 538us/step - loss: 
0.0534 - acc: 0.9867 - val_loss: 1.2401 - val_acc: 0.8482
Epoch 19/25
115275/115275 [==============================] - 62s 539us/step - loss: 
0.0504 - acc: 0.9872 - val_loss: 1.2454 - val_acc: 0.8503
```

```
0.0504 - acc: 0.9872 - val_loss: 1.2434 - val_acc: 0.8505

Epoch 20/25
115275/115275 [==============================] - 62s 538us/step - loss:
0.0499 - acc: 0.9880 - val_loss: 1.2413 - val_acc: 0.8538
Epoch 21/25
115275/115275 [==============================] - 62s 538us/step - loss:
0.0480 - acc: 0.9883 - val_loss: 1.2119 - val_acc: 0.8576
Epoch 22/25
115275/115275 [==============================] - 62s 538us/step - loss:
0.0484 - acc: 0.9884 - val_loss: 1.2224 - val_acc: 0.8580
Epoch 23/25
115275/115275 [==============================] - 62s 538us/step - loss:
0.0450 - acc: 0.9891 - val_loss: 1.2322 - val_acc: 0.8582
Epoch 24/25
115275/115275 [==============================] - 62s 540us/step - loss:
0.0444 - acc: 0.9896 - val_loss: 1.2356 - val_acc: 0.8557
Epoch 25/25
115275/115275 [==============================] - 62s 538us/step - loss:
0.0446 - acc: 0.9891 - val_loss: 1.2405 - val_acc: 0.8553
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,26))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 85.53%
```



In [0]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(2000,kernel_size=26,activation='relu',input_shape=(200
,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.4))
model.add(Conv1D(250,20,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.6))
model.add(BatchNormalization())
```

```
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=80,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
W0619 09:23:08.199344 140477795272576 nn_ops.py:4224] Large dropout rat
e: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/80
130000/130000 [==============================] - 210s 2ms/step - loss:
6.7167 - acc: 0.1176 - val_loss: 3.9094 - val_acc: 0.4072
Epoch 2/80
130000/130000 [==============================] - 207s 2ms/step - loss:
3.6362 - acc: 0.4200 - val_loss: 2.0259 - val_acc: 0.6724
Epoch 3/80
130000/130000 [==============================] - 207s 2ms/step - loss:
2.2216 - acc: 0.5902 - val_loss: 1.4423 - val_acc: 0.7579
Epoch 4/80
130000/130000 [==============================] - 207s 2ms/step - loss:
1.5199 - acc: 0.6887 - val_loss: 1.1751 - val_acc: 0.7984
Epoch 5/80
130000/130000 [==============================] - 207s 2ms/step - loss:
1.1232 - acc: 0.7518 - val_loss: 1.0267 - val_acc: 0.8250
Epoch 6/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.8600 - acc: 0.7990 - val_loss: 0.9646 - val_acc: 0.8383
Epoch 7/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.6916 - acc: 0.8312 - val_loss: 0.9139 - val_acc: 0.8487
Epoch 8/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.5715 - acc: 0.8562 - val_loss: 0.8772 - val_acc: 0.8534
Epoch 9/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.4941 - acc: 0.8716 - val_loss: 0.8622 - val_acc: 0.8614
Epoch 10/80
130000/130000 [==============================] - 207s 2ms/step - loss:
```

```
130000/130000 [==============================] - 207s 2ms/step - loss:
0.4336 - acc: 0.8860 - val_loss: 0.8879 - val_acc: 0.8614
Epoch 11/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.3850 - acc: 0.8974 - val_loss: 0.8403 - val_acc: 0.8688
Epoch 12/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.3440 - acc: 0.9070 - val_loss: 0.8286 - val_acc: 0.8694
Epoch 13/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.3167 - acc: 0.9134 - val_loss: 0.8284 - val_acc: 0.8703
Epoch 14/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.2980 - acc: 0.9179 - val_loss: 0.8511 - val_acc: 0.8739
Epoch 15/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2689 - acc: 0.9251 - val_loss: 0.8374 - val_acc: 0.8747
Epoch 16/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2545 - acc: 0.9290 - val_loss: 0.8265 - val_acc: 0.8755
Epoch 17/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2384 - acc: 0.9337 - val_loss: 0.8368 - val_acc: 0.8776
Epoch 18/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2298 - acc: 0.9358 - val_loss: 0.8363 - val_acc: 0.8791
Epoch 19/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2141 - acc: 0.9396 - val_loss: 0.8347 - val_acc: 0.8793
Epoch 20/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.2060 - acc: 0.9420 - val_loss: 0.7848 - val_acc: 0.8840
Epoch 21/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1941 - acc: 0.9452 - val_loss: 0.7954 - val_acc: 0.8847
Epoch 22/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1797 - acc: 0.9490 - val_loss: 0.8097 - val_acc: 0.8833
Epoch 23/80
130000/130000 [==============================] - 206s 2ms/step - loss:
```

```
130000/130000 [==============================] - 206s 2ms/step - loss:

0.1750 - acc: 0.9498 - val_loss: 0.8052 - val_acc: 0.8846
Epoch 24/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1669 - acc: 0.9527 - val_loss: 0.8236 - val_acc: 0.8843
Epoch 25/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1624 - acc: 0.9542 - val_loss: 0.8016 - val_acc: 0.8860
Epoch 26/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1533 - acc: 0.9559 - val_loss: 0.7958 - val_acc: 0.8872
Epoch 27/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1463 - acc: 0.9577 - val_loss: 0.8126 - val_acc: 0.8881
Epoch 28/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1452 - acc: 0.9583 - val_loss: 0.8112 - val_acc: 0.8886
Epoch 29/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1389 - acc: 0.9600 - val_loss: 0.8024 - val_acc: 0.8892
Epoch 30/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1372 - acc: 0.9607 - val_loss: 0.7927 - val_acc: 0.8890
Epoch 31/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1317 - acc: 0.9622 - val_loss: 0.7913 - val_acc: 0.8892
Epoch 32/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1281 - acc: 0.9632 - val_loss: 0.8051 - val_acc: 0.8874
Epoch 33/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1217 - acc: 0.9650 - val_loss: 0.7938 - val_acc: 0.8917
Epoch 34/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1184 - acc: 0.9660 - val_loss: 0.7883 - val_acc: 0.8922
Epoch 35/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1155 - acc: 0.9665 - val_loss: 0.7669 - val_acc: 0.8942
Epoch 36/80
130000/130000 [==============================] - 206s 2ms/step - loss:
```

```
130000/130000 [==============================] - 206s 2ms/step - loss:

0.1140 - acc: 0.9671 - val_loss: 0.8297 - val_acc: 0.8902
Epoch 37/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1129 - acc: 0.9681 - val_loss: 0.8031 - val_acc: 0.8920
Epoch 38/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1057 - acc: 0.9693 - val_loss: 0.7931 - val_acc: 0.8933
Epoch 39/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.1032 - acc: 0.9706 - val_loss: 0.7741 - val_acc: 0.8950
Epoch 40/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0999 - acc: 0.9722 - val_loss: 0.8107 - val_acc: 0.8941
Epoch 41/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0974 - acc: 0.9720 - val_loss: 0.8019 - val_acc: 0.8933
Epoch 42/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0961 - acc: 0.9727 - val_loss: 0.8070 - val_acc: 0.8934
Epoch 43/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0943 - acc: 0.9729 - val_loss: 0.7963 - val_acc: 0.8940
Epoch 44/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0953 - acc: 0.9722 - val_loss: 0.7794 - val_acc: 0.8941
Epoch 45/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0902 - acc: 0.9739 - val_loss: 0.7772 - val_acc: 0.8963
Epoch 46/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0903 - acc: 0.9741 - val_loss: 0.7836 - val_acc: 0.8945
Epoch 47/80
130000/130000 [==============================] - 206s 2ms/step - loss:
0.0890 - acc: 0.9748 - val_loss: 0.7851 - val_acc: 0.8959
Epoch 48/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0863 - acc: 0.9752 - val_loss: 0.7761 - val_acc: 0.8962
Epoch 49/80
130000/130000 [==============================] - 207s 2ms/step - loss:
```

```
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0866 - acc: 0.9757 - val_loss: 0.7920 - val_acc: 0.8957
Epoch 50/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0817 - acc: 0.9764 - val_loss: 0.7716 - val_acc: 0.8976
Epoch 51/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0817 - acc: 0.9756 - val_loss: 0.7723 - val_acc: 0.8975
Epoch 52/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0817 - acc: 0.9766 - val_loss: 0.8065 - val_acc: 0.8940
Epoch 53/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0807 - acc: 0.9768 - val_loss: 0.8028 - val_acc: 0.8950
Epoch 54/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0781 - acc: 0.9774 - val_loss: 0.7828 - val_acc: 0.8979
Epoch 55/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0772 - acc: 0.9776 - val_loss: 0.7769 - val_acc: 0.8982
Epoch 56/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0743 - acc: 0.9788 - val_loss: 0.7777 - val_acc: 0.8972
Epoch 57/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0756 - acc: 0.9783 - val_loss: 0.7816 - val_acc: 0.8982
Epoch 58/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0713 - acc: 0.9793 - val_loss: 0.7880 - val_acc: 0.8980
Epoch 59/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0736 - acc: 0.9791 - val_loss: 0.7590 - val_acc: 0.8990
Epoch 60/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0712 - acc: 0.9799 - val_loss: 0.7806 - val_acc: 0.8994
Epoch 61/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0702 - acc: 0.9799 - val_loss: 0.7773 - val_acc: 0.9000
Epoch 62/80
130000/130000 [==============================] - 207s 2ms/step - loss:
```

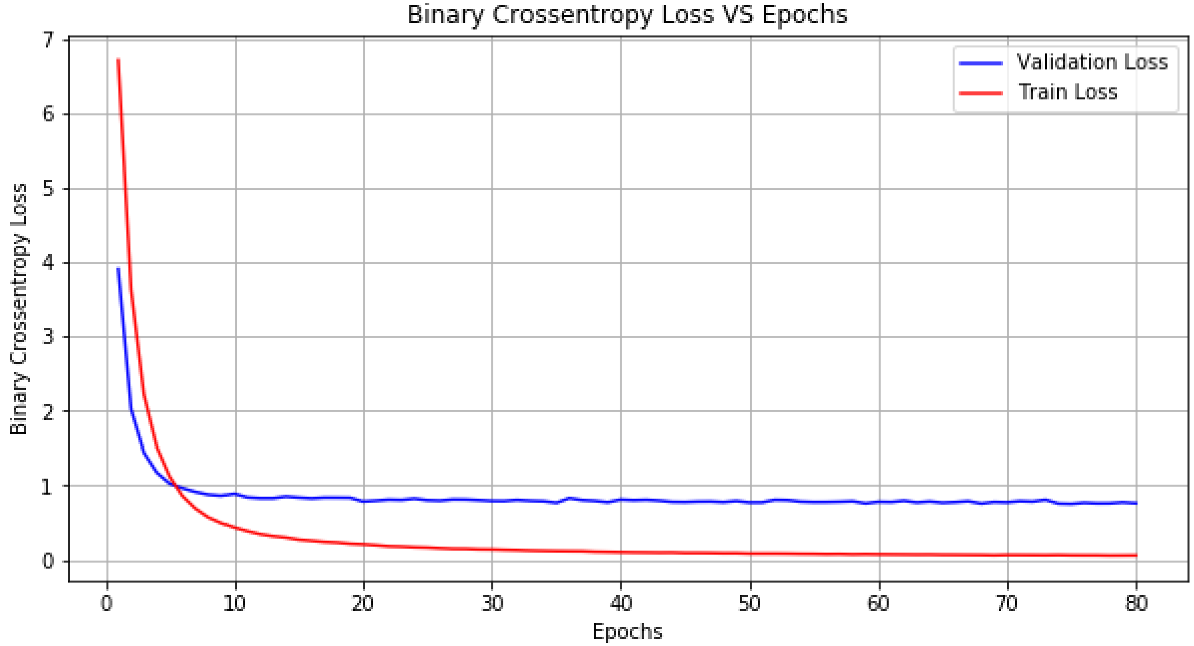
```
130000/130000 [==============================] - 207s 2ms/step - loss:

0.0691 - acc: 0.9806 - val_loss: 0.7952 - val_acc: 0.8991
Epoch 63/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0678 - acc: 0.9801 - val_loss: 0.7714 - val_acc: 0.9011
Epoch 64/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0682 - acc: 0.9808 - val_loss: 0.7852 - val_acc: 0.9001
Epoch 65/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0655 - acc: 0.9810 - val_loss: 0.7679 - val_acc: 0.8996
Epoch 66/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0650 - acc: 0.9811 - val_loss: 0.7763 - val_acc: 0.8999
Epoch 67/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0636 - acc: 0.9817 - val_loss: 0.7892 - val_acc: 0.9008
Epoch 68/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0625 - acc: 0.9818 - val_loss: 0.7571 - val_acc: 0.9017
Epoch 69/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0595 - acc: 0.9827 - val_loss: 0.7780 - val_acc: 0.9005
Epoch 70/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0629 - acc: 0.9819 - val_loss: 0.7730 - val_acc: 0.9006
Epoch 71/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0623 - acc: 0.9825 - val_loss: 0.7901 - val_acc: 0.8988
Epoch 72/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0614 - acc: 0.9826 - val_loss: 0.7837 - val_acc: 0.9005
Epoch 73/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0603 - acc: 0.9830 - val_loss: 0.8077 - val_acc: 0.8963
Epoch 74/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0621 - acc: 0.9824 - val_loss: 0.7569 - val_acc: 0.9016
Epoch 75/80
130000/130000 [==============================] - 207s 2ms/step - loss:
```

```
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0593 - acc: 0.9833 - val_loss: 0.7522 - val_acc: 0.9019
Epoch 76/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0587 - acc: 0.9835 - val_loss: 0.7670 - val_acc: 0.9006
Epoch 77/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0584 - acc: 0.9836 - val_loss: 0.7596 - val_acc: 0.9018
Epoch 78/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0559 - acc: 0.9840 - val_loss: 0.7602 - val_acc: 0.9006
Epoch 79/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0564 - acc: 0.9838 - val_loss: 0.7728 - val_acc: 0.9000
Epoch 80/80
130000/130000 [==============================] - 207s 2ms/step - loss:
0.0574 - acc: 0.9832 - val_loss: 0.7615 - val_acc: 0.9034
```

In [0]:

```python
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,81))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 90.34%
```



In [0]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM,Bidirectional
model=Sequential()
model.add(Dense(256,activation='relu',input_shape=(200,26)))
model.add(Dropout(0.3))
model.add(BatchNormalization())
model.add(LSTM(26,dropout=0.2,return_sequences=True))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(128,activation='relu',input_shape=(200,26)))
model.add(Dropout(0.3))
model.add(BatchNormalization())
model.add(Dense(11180,activation='softmax'))
```

```
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=60,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0619 16:14:51.595402 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:74: The name tf.get_default_graph is deprecated. Please use tf.com
pat.v1.get_default_graph instead.

W0619 16:14:51.627513 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

W0619 16:14:51.638072 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:4138: The name tf.random_uniform is deprecated. Please use tf.rand
om.uniform instead.

W0619 16:14:51.663547 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:133: The name tf.placeholder_with_default is deprecated. Please us
e tf.compat.v1.placeholder_with_default instead.

W0619 16:14:51.673146 140045008664448 deprecation.py:506] From /usr/loc
al/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:344
5: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
W0619 16:14:52.402341 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The nam
e tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optim
izer instead.

W0619 16:14:52.425379 140045008664448 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
```

```
d.py:3341: The name tf.log is deprecated. Please use tf.math.log instea
d.

W0619 16:14:52.531753 140045008664448 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/60
130000/130000 [==============================] - 364s 3ms/step - loss:
7.1105 - acc: 0.0813 - val_loss: 5.0234 - val_acc: 0.3011
Epoch 2/60
130000/130000 [==============================] - 354s 3ms/step - loss:
4.0551 - acc: 0.3979 - val_loss: 2.2129 - val_acc: 0.6724
Epoch 3/60
130000/130000 [==============================] - 356s 3ms/step - loss:
2.1054 - acc: 0.6402 - val_loss: 1.2749 - val_acc: 0.7995
Epoch 4/60
130000/130000 [==============================] - 357s 3ms/step - loss:
1.2024 - acc: 0.7645 - val_loss: 0.9455 - val_acc: 0.8418
Epoch 5/60
130000/130000 [==============================] - 356s 3ms/step - loss:
0.7732 - acc: 0.8329 - val_loss: 0.8285 - val_acc: 0.8560
Epoch 6/60
130000/130000 [==============================] - 348s 3ms/step - loss:
0.5557 - acc: 0.8712 - val_loss: 0.7840 - val_acc: 0.8620
Epoch 7/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.4395 - acc: 0.8931 - val_loss: 0.7603 - val_acc: 0.8665
Epoch 8/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.3698 - acc: 0.9051 - val_loss: 0.7541 - val_acc: 0.8669
Epoch 9/60
130000/130000 [==============================] - 358s 3ms/step - loss:
0.3255 - acc: 0.9154 - val_loss: 0.7593 - val_acc: 0.8695
Epoch 10/60
130000/130000 [==============================] - 359s 3ms/step - loss:
```

```
0.2862 - acc: 0.9230 - val_loss: 0.7505 - val_acc: 0.8728
Epoch 11/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.2600 - acc: 0.9296 - val_loss: 0.7510 - val_acc: 0.8728
Epoch 12/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.2408 - acc: 0.9343 - val_loss: 0.7619 - val_acc: 0.8719
Epoch 13/60
130000/130000 [==============================] - 348s 3ms/step - loss:
0.2261 - acc: 0.9375 - val_loss: 0.7672 - val_acc: 0.8732
Epoch 14/60
130000/130000 [==============================] - 347s 3ms/step - loss:
0.2149 - acc: 0.9403 - val_loss: 0.7720 - val_acc: 0.8720
Epoch 15/60
130000/130000 [==============================] - 347s 3ms/step - loss:
0.1985 - acc: 0.9444 - val_loss: 0.7741 - val_acc: 0.8718
Epoch 16/60
130000/130000 [==============================] - 346s 3ms/step - loss:
0.1896 - acc: 0.9467 - val_loss: 0.7699 - val_acc: 0.8752
Epoch 17/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1817 - acc: 0.9484 - val_loss: 0.7783 - val_acc: 0.8742
Epoch 18/60
130000/130000 [==============================] - 353s 3ms/step - loss:
0.1705 - acc: 0.9515 - val_loss: 0.7826 - val_acc: 0.8750
Epoch 19/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.1637 - acc: 0.9538 - val_loss: 0.7891 - val_acc: 0.8775
Epoch 20/60
130000/130000 [==============================] - 348s 3ms/step - loss:
0.1598 - acc: 0.9549 - val_loss: 0.7953 - val_acc: 0.8738
Epoch 21/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1567 - acc: 0.9551 - val_loss: 0.7832 - val_acc: 0.8774
Epoch 22/60
130000/130000 [==============================] - 348s 3ms/step - loss:
0.1511 - acc: 0.9569 - val_loss: 0.7959 - val_acc: 0.8748
Epoch 23/60
130000/130000 [==============================] - 346s 3ms/step - loss:
```

```
0.1450 - acc: 0.9580 - val_loss: 0.7964 - val_acc: 0.8768
Epoch 24/60
130000/130000 [==============================] - 347s 3ms/step - loss:
0.1406 - acc: 0.9596 - val_loss: 0.8018 - val_acc: 0.8761
Epoch 25/60
130000/130000 [==============================] - 345s 3ms/step - loss:
0.1376 - acc: 0.9602 - val_loss: 0.8057 - val_acc: 0.8779
Epoch 26/60
130000/130000 [==============================] - 345s 3ms/step - loss:
0.1319 - acc: 0.9614 - val_loss: 0.8022 - val_acc: 0.8775
Epoch 27/60
130000/130000 [==============================] - 347s 3ms/step - loss:
0.1293 - acc: 0.9625 - val_loss: 0.8029 - val_acc: 0.8774
Epoch 28/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1273 - acc: 0.9632 - val_loss: 0.8071 - val_acc: 0.8779
Epoch 29/60
130000/130000 [==============================] - 347s 3ms/step - loss:
0.1256 - acc: 0.9640 - val_loss: 0.7995 - val_acc: 0.8778
Epoch 30/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1164 - acc: 0.9662 - val_loss: 0.8161 - val_acc: 0.8770
Epoch 31/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.1188 - acc: 0.9654 - val_loss: 0.8068 - val_acc: 0.8780
Epoch 32/60
130000/130000 [==============================] - 354s 3ms/step - loss:
0.1140 - acc: 0.9662 - val_loss: 0.8140 - val_acc: 0.8799
Epoch 33/60
130000/130000 [==============================] - 353s 3ms/step - loss:
0.1124 - acc: 0.9676 - val_loss: 0.8021 - val_acc: 0.8793
Epoch 34/60
130000/130000 [==============================] - 352s 3ms/step - loss:
0.1067 - acc: 0.9686 - val_loss: 0.8056 - val_acc: 0.8797
Epoch 35/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1088 - acc: 0.9679 - val_loss: 0.8234 - val_acc: 0.8788
Epoch 36/60
130000/130000 [==============================] - 350s 3ms/step - loss:
```

```
0.1072 - acc: 0.9687 - val_loss: 0.8215 - val_acc: 0.8774
Epoch 37/60
130000/130000 [==============================] - 352s 3ms/step - loss:
0.1065 - acc: 0.9687 - val_loss: 0.8254 - val_acc: 0.8796
Epoch 38/60
130000/130000 [==============================] - 352s 3ms/step - loss:
0.1149 - acc: 0.9669 - val_loss: 0.8336 - val_acc: 0.8789
Epoch 39/60
130000/130000 [==============================] - 349s 3ms/step - loss:
0.1023 - acc: 0.9699 - val_loss: 0.8270 - val_acc: 0.8800
Epoch 40/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0967 - acc: 0.9714 - val_loss: 0.8330 - val_acc: 0.8794
Epoch 41/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.0959 - acc: 0.9718 - val_loss: 0.8277 - val_acc: 0.8810
Epoch 42/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0961 - acc: 0.9715 - val_loss: 0.8361 - val_acc: 0.8769
Epoch 43/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0930 - acc: 0.9725 - val_loss: 0.8367 - val_acc: 0.8816
Epoch 44/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0947 - acc: 0.9728 - val_loss: 0.8306 - val_acc: 0.8809
Epoch 45/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0918 - acc: 0.9731 - val_loss: 0.8391 - val_acc: 0.8805
Epoch 46/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.0873 - acc: 0.9743 - val_loss: 0.8397 - val_acc: 0.8806
Epoch 47/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.0902 - acc: 0.9736 - val_loss: 0.8352 - val_acc: 0.8816
Epoch 48/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0867 - acc: 0.9742 - val_loss: 0.8360 - val_acc: 0.8820
Epoch 49/60
130000/130000 [==============================] - 353s 3ms/step - loss:
```

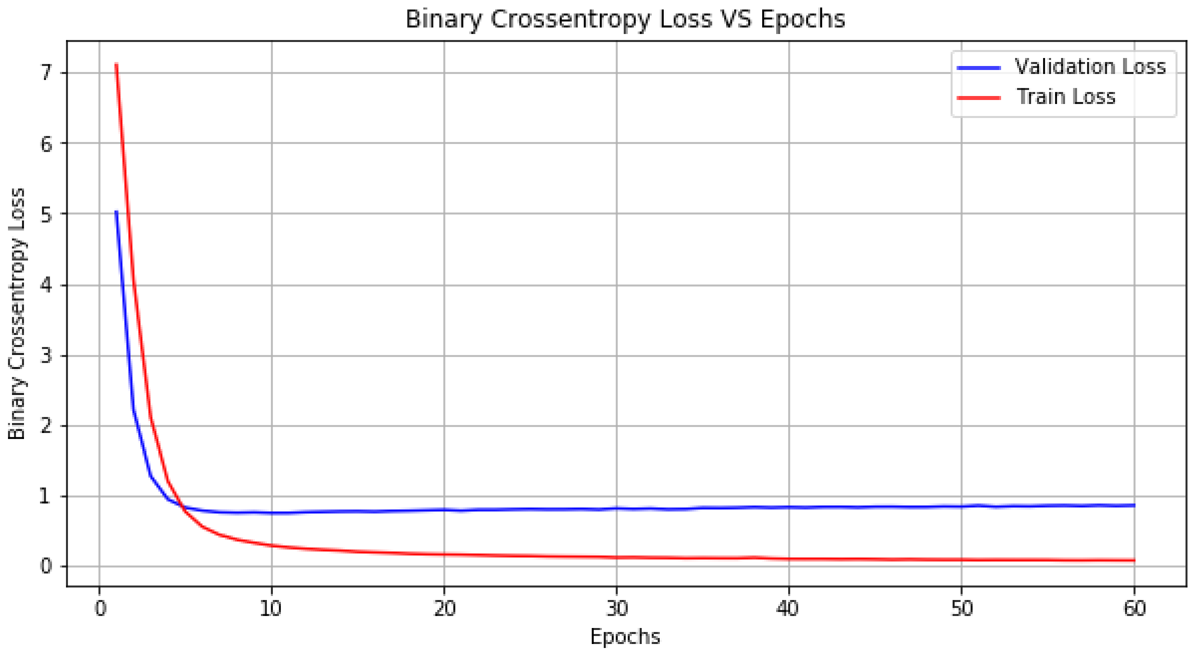
```
0.0852 - acc: 0.9743 - val_loss: 0.8438 - val_acc: 0.8806
Epoch 50/60
130000/130000 [==============================] - 352s 3ms/step - loss:
0.0852 - acc: 0.9750 - val_loss: 0.8409 - val_acc: 0.8825
Epoch 51/60
130000/130000 [==============================] - 355s 3ms/step - loss:
0.0812 - acc: 0.9756 - val_loss: 0.8556 - val_acc: 0.8812
Epoch 52/60
130000/130000 [==============================] - 354s 3ms/step - loss:
0.0824 - acc: 0.9757 - val_loss: 0.8384 - val_acc: 0.8831
Epoch 53/60
130000/130000 [==============================] - 353s 3ms/step - loss:
0.0816 - acc: 0.9764 - val_loss: 0.8481 - val_acc: 0.8814
Epoch 54/60
130000/130000 [==============================] - 352s 3ms/step - loss:
0.0813 - acc: 0.9758 - val_loss: 0.8457 - val_acc: 0.8817
Epoch 55/60
130000/130000 [==============================] - 351s 3ms/step - loss:
0.0806 - acc: 0.9763 - val_loss: 0.8539 - val_acc: 0.8822
Epoch 56/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0771 - acc: 0.9778 - val_loss: 0.8569 - val_acc: 0.8811
Epoch 57/60
130000/130000 [==============================] - 350s 3ms/step - loss:
0.0762 - acc: 0.9775 - val_loss: 0.8511 - val_acc: 0.8827
Epoch 58/60
 29056/130000 [=====>........................] - ETA: 4:18 - loss: 0.06
58 - acc: 0.9801Buffered data was truncated after reaching the output s
ize limit.
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])
```

```
# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,61))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 88.17%
```



In [0]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
```

```python
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=10,activation='relu',input_shape=(200,
26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(250,20,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.2))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=25,validation_data=[testarr
ay1,testy],batch_size=256)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0618 13:48:12.199924 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:74: The name tf.get_default_graph is deprecated. Please use tf.com
pat.v1.get_default_graph instead.

W0618 13:48:12.241334 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

W0618 13:48:12.250728 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:4138: The name tf.random_uniform is deprecated. Please use tf.rand
om.uniform instead.

W0618 13:48:12.290457 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_
pool2d instead.

W0618 13:48:12.305730 140483376052096 deprecation_wrapper.py:119] From
```

```
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:133: The name tf.placeholder_with_default is deprecated. Please us
e tf.compat.v1.placeholder_with_default instead.

W0618 13:48:12.314735 140483376052096 deprecation.py:506] From /usr/loc
al/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:344
5: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
W0618 13:48:12.492441 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The nam
e tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optim
izer instead.

W0618 13:48:12.514500 140483376052096 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3341: The name tf.log is deprecated. Please use tf.math.log instea
d.

W0618 13:48:12.617041 140483376052096 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/25
130000/130000 [==============================] - 77s 591us/step - loss:
5.1009 - acc: 0.3052 - val_loss: 1.9616 - val_acc: 0.6892
Epoch 2/25
130000/130000 [==============================] - 69s 535us/step - loss:
1.3050 - acc: 0.7648 - val_loss: 1.0471 - val_acc: 0.8212
Epoch 3/25
130000/130000 [==============================] - 69s 532us/step - loss:
0.4258 - acc: 0.9047 - val_loss: 0.9388 - val_acc: 0.8430
Epoch 4/25
130000/130000 [==============================] - 69s 531us/step - loss:
```

```
130000/130000 [==============================] - 69s 531us/step - loss:
0.2238 - acc: 0.9454 - val_loss: 0.9228 - val_acc: 0.8507
Epoch 5/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.1656 - acc: 0.9579 - val_loss: 1.0461 - val_acc: 0.8388
Epoch 6/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.1384 - acc: 0.9646 - val_loss: 0.9860 - val_acc: 0.8503
Epoch 7/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.1185 - acc: 0.9691 - val_loss: 0.9931 - val_acc: 0.8566
Epoch 8/25
130000/130000 [==============================] - 70s 535us/step - loss:
0.1103 - acc: 0.9708 - val_loss: 1.0009 - val_acc: 0.8611
Epoch 9/25
130000/130000 [==============================] - 69s 530us/step - loss:
0.0987 - acc: 0.9747 - val_loss: 0.9943 - val_acc: 0.8600
Epoch 10/25
130000/130000 [==============================] - 69s 529us/step - loss:
0.0927 - acc: 0.9766 - val_loss: 0.9953 - val_acc: 0.8648
Epoch 11/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.0841 - acc: 0.9788 - val_loss: 1.0196 - val_acc: 0.8641
Epoch 12/25
130000/130000 [==============================] - 69s 534us/step - loss:
0.0777 - acc: 0.9808 - val_loss: 1.0280 - val_acc: 0.8656
Epoch 13/25
130000/130000 [==============================] - 69s 532us/step - loss:
0.0733 - acc: 0.9819 - val_loss: 0.9936 - val_acc: 0.8682
Epoch 14/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.0699 - acc: 0.9831 - val_loss: 1.0235 - val_acc: 0.8646
Epoch 15/25
130000/130000 [==============================] - 69s 531us/step - loss:
0.0669 - acc: 0.9834 - val_loss: 1.0636 - val_acc: 0.8653
Epoch 16/25
130000/130000 [==============================] - 69s 529us/step - loss:
0.0622 - acc: 0.9855 - val_loss: 2.5383 - val_acc: 0.7006
Epoch 17/25
130000/130000 [==============================] - 69s 528us/step - loss:
```

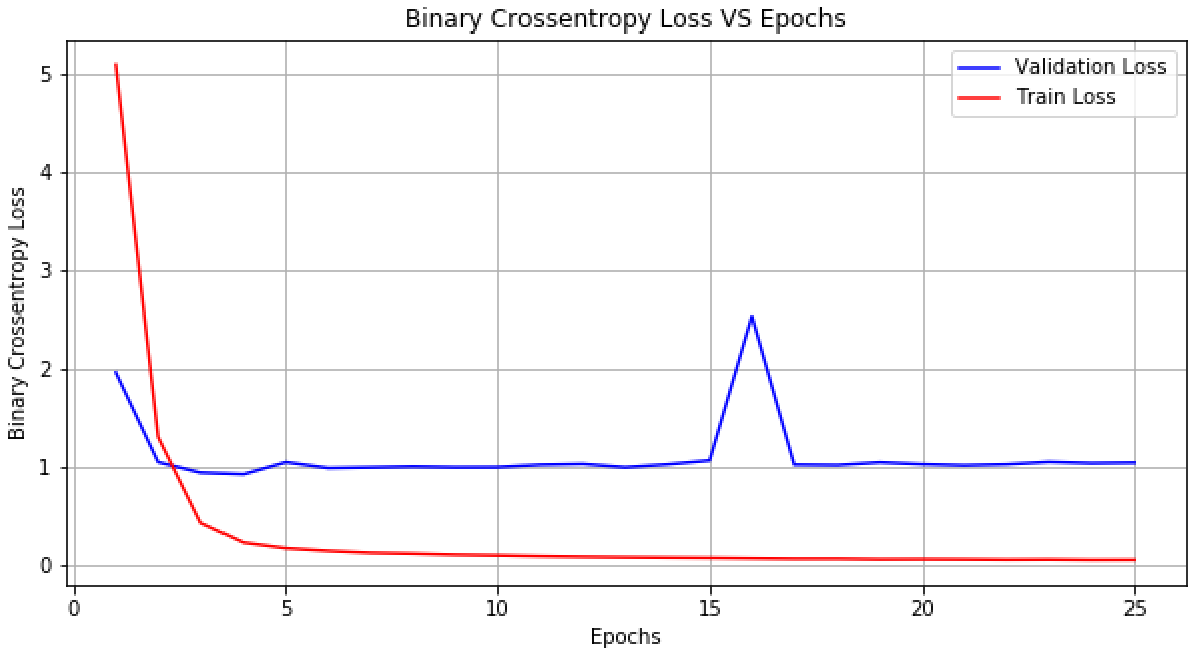
```
0.0579 - acc: 0.9864 - val_loss: 1.0205 - val_acc: 0.8738
Epoch 18/25
130000/130000 [==============================] - 69s 528us/step - loss:
0.0579 - acc: 0.9866 - val_loss: 1.0162 - val_acc: 0.8747
Epoch 19/25
130000/130000 [==============================] - 69s 529us/step - loss:
0.0531 - acc: 0.9878 - val_loss: 1.0443 - val_acc: 0.8726
Epoch 20/25
130000/130000 [==============================] - 69s 529us/step - loss:
0.0540 - acc: 0.9874 - val_loss: 1.0252 - val_acc: 0.8741
Epoch 21/25
130000/130000 [==============================] - 69s 528us/step - loss:
0.0522 - acc: 0.9881 - val_loss: 1.0139 - val_acc: 0.8741
Epoch 22/25
130000/130000 [==============================] - 69s 527us/step - loss:
0.0489 - acc: 0.9892 - val_loss: 1.0241 - val_acc: 0.8778
Epoch 23/25
130000/130000 [==============================] - 69s 528us/step - loss:
0.0502 - acc: 0.9888 - val_loss: 1.0487 - val_acc: 0.8726
Epoch 24/25
130000/130000 [==============================] - 69s 529us/step - loss:
0.0464 - acc: 0.9900 - val_loss: 1.0362 - val_acc: 0.8773
Epoch 25/25
130000/130000 [==============================] - 69s 528us/step - loss:
0.0469 - acc: 0.9899 - val_loss: 1.0409 - val_acc: 0.8772
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,26))
```

```
# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 87.72%
```



In [0]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=25,activation='relu',input_shape=(200,
26)))
```

```
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.2))
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))

model.add(Dropout(0.3))
model.add(BatchNormalization())

model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=55,validation_data=[testarr
ay1,testy],batch_size=64)
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/80
130000/130000 [==============================] - 98s 755us/step - loss:
5.9637 - acc: 0.1907 - val_loss: 2.9397 - val_acc: 0.5316
Epoch 2/80
130000/130000 [==============================] - 95s 729us/step - loss:
2.6973 - acc: 0.5357 - val_loss: 1.5884 - val_acc: 0.7259
Epoch 3/80
130000/130000 [==============================] - 95s 728us/step - loss:
1.5215 - acc: 0.6920 - val_loss: 1.2224 - val_acc: 0.7853
Epoch 4/80
130000/130000 [==============================] - 95s 728us/step - loss:
1.0194 - acc: 0.7727 - val_loss: 1.0830 - val_acc: 0.8136
Epoch 5/80
130000/130000 [==============================] - 95s 729us/step - loss:
0.7519 - acc: 0.8211 - val_loss: 1.0255 - val_acc: 0.8237
Epoch 6/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.6017 - acc: 0.8516 - val_loss: 0.9725 - val_acc: 0.8367
Epoch 7/80
130000/130000 [==============================] - 95s 729us/step - loss:
0.5003 - acc: 0.8719 - val_loss: 0.9515 - val_acc: 0.8442
Epoch 8/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.4340 - acc: 0.8874 - val_loss: 0.9436 - val_acc: 0.8474
```

```
Epoch 9/80

130000/130000 [==============================] - 95s 728us/step - loss:
0.3851 - acc: 0.8985 - val_loss: 0.9444 - val_acc: 0.8515
Epoch 10/80
130000/130000 [==============================] - 94s 727us/step - loss:
0.3467 - acc: 0.9069 - val_loss: 0.9340 - val_acc: 0.8546
Epoch 11/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.3239 - acc: 0.9135 - val_loss: 0.9380 - val_acc: 0.8571
Epoch 12/80
130000/130000 [==============================] - 94s 727us/step - loss:
0.2959 - acc: 0.9198 - val_loss: 0.9285 - val_acc: 0.8603
Epoch 13/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.2776 - acc: 0.9250 - val_loss: 0.9277 - val_acc: 0.8610
Epoch 14/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2538 - acc: 0.9308 - val_loss: 0.9311 - val_acc: 0.8636
Epoch 15/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2433 - acc: 0.9337 - val_loss: 0.9091 - val_acc: 0.8671
Epoch 16/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2336 - acc: 0.9363 - val_loss: 0.9291 - val_acc: 0.8642
Epoch 17/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2204 - acc: 0.9397 - val_loss: 0.9106 - val_acc: 0.8659
Epoch 18/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2116 - acc: 0.9431 - val_loss: 0.8966 - val_acc: 0.8700
Epoch 19/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.2009 - acc: 0.9450 - val_loss: 0.9074 - val_acc: 0.8717
Epoch 20/80
130000/130000 [==============================] - 94s 726us/step - loss:
0.1941 - acc: 0.9463 - val_loss: 0.8932 - val_acc: 0.8721
Epoch 21/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.1859 - acc: 0.9489 - val_loss: 0.9019 - val_acc: 0.8721
Epoch 22/80
```

```
Epoch 22/80

130000/130000 [==============================] - 95s 727us/step - loss:
0.1826 - acc: 0.9501 - val_loss: 0.8993 - val_acc: 0.8744
Epoch 23/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1741 - acc: 0.9526 - val_loss: 0.9086 - val_acc: 0.8718
Epoch 24/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1663 - acc: 0.9544 - val_loss: 0.8825 - val_acc: 0.8750
Epoch 25/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1666 - acc: 0.9545 - val_loss: 0.8968 - val_acc: 0.8749
Epoch 26/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1564 - acc: 0.9572 - val_loss: 0.8901 - val_acc: 0.8753
Epoch 27/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1568 - acc: 0.9572 - val_loss: 0.8926 - val_acc: 0.8768
Epoch 28/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.1505 - acc: 0.9592 - val_loss: 0.8964 - val_acc: 0.8769
Epoch 29/80
130000/130000 [==============================] - 95s 728us/step - loss:
0.1443 - acc: 0.9610 - val_loss: 0.8780 - val_acc: 0.8797
Epoch 30/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.1426 - acc: 0.9609 - val_loss: 0.9063 - val_acc: 0.8762
Epoch 31/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.1384 - acc: 0.9620 - val_loss: 0.8973 - val_acc: 0.8783
Epoch 32/80
130000/130000 [==============================] - 95s 727us/step - loss:
0.1329 - acc: 0.9632 - val_loss: 0.8948 - val_acc: 0.8797
Epoch 33/80
130000/130000 [==============================] - 94s 727us/step - loss:
0.1314 - acc: 0.9639 - val_loss: 0.8832 - val_acc: 0.8812
Epoch 34/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.1264 - acc: 0.9650 - val_loss: 0.8835 - val_acc: 0.8800
Epoch 35/80
```

```
Epoch 35/80

130000/130000 [==============================] - 95s 730us/step - loss:
0.1274 - acc: 0.9652 - val_loss: 0.8742 - val_acc: 0.8829
Epoch 36/80
130000/130000 [==============================] - 95s 730us/step - loss:
0.1214 - acc: 0.9669 - val_loss: 0.8891 - val_acc: 0.8798
Epoch 37/80
130000/130000 [==============================] - 95s 729us/step - loss:
0.1245 - acc: 0.9657 - val_loss: 0.8730 - val_acc: 0.8835
Epoch 38/80
130000/130000 [==============================] - 95s 731us/step - loss:
0.1195 - acc: 0.9673 - val_loss: 0.8715 - val_acc: 0.8819
Epoch 39/80
130000/130000 [==============================] - 95s 733us/step - loss:
0.1151 - acc: 0.9684 - val_loss: 0.8703 - val_acc: 0.8832
Epoch 40/80
130000/130000 [==============================] - 95s 733us/step - loss:
0.1134 - acc: 0.9693 - val_loss: 0.8643 - val_acc: 0.8832
Epoch 41/80
130000/130000 [==============================] - 96s 738us/step - loss:
0.1143 - acc: 0.9688 - val_loss: 0.8697 - val_acc: 0.8839
Epoch 42/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.1094 - acc: 0.9702 - val_loss: 0.8668 - val_acc: 0.8849
Epoch 43/80
130000/130000 [==============================] - 95s 734us/step - loss:
0.1075 - acc: 0.9708 - val_loss: 0.8576 - val_acc: 0.8841
Epoch 44/80
130000/130000 [==============================] - 95s 734us/step - loss:
0.1071 - acc: 0.9713 - val_loss: 0.8601 - val_acc: 0.8844
Epoch 45/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.1053 - acc: 0.9715 - val_loss: 0.8715 - val_acc: 0.8850
Epoch 46/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.1056 - acc: 0.9716 - val_loss: 0.8693 - val_acc: 0.8855
Epoch 47/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.1004 - acc: 0.9729 - val_loss: 0.8746 - val_acc: 0.8846
Epoch 48/80
```

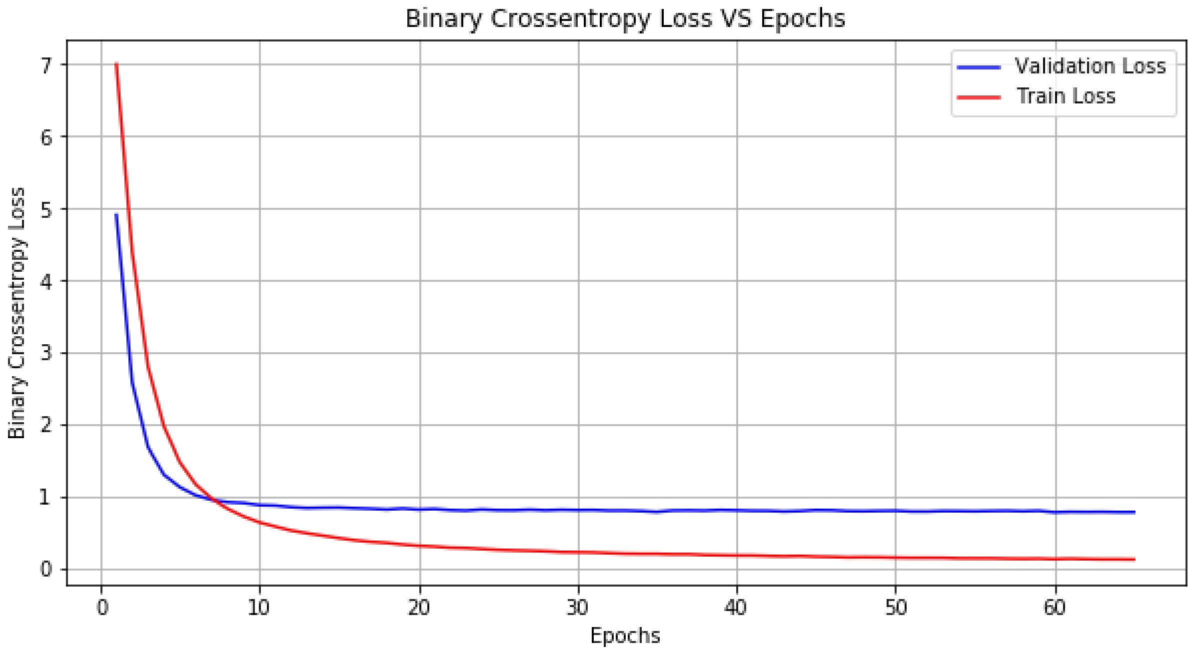
```
Epoch 48/80

130000/130000 [==============================] - 96s 735us/step - loss:
0.0994 - acc: 0.9732 - val_loss: 0.8684 - val_acc: 0.8840
Epoch 49/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.0980 - acc: 0.9735 - val_loss: 0.8739 - val_acc: 0.8850
Epoch 50/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.0977 - acc: 0.9738 - val_loss: 0.8547 - val_acc: 0.8856
Epoch 51/80
130000/130000 [==============================] - 95s 734us/step - loss:
0.0971 - acc: 0.9738 - val_loss: 0.8667 - val_acc: 0.8857
Epoch 52/80
130000/130000 [==============================] - 96s 736us/step - loss:
0.0942 - acc: 0.9743 - val_loss: 0.8669 - val_acc: 0.8850
Epoch 53/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.0946 - acc: 0.9745 - val_loss: 0.8598 - val_acc: 0.8870
Epoch 54/80
130000/130000 [==============================] - 96s 735us/step - loss:
0.0934 - acc: 0.9749 - val_loss: 0.8530 - val_acc: 0.8872
Epoch 55/80
130000/130000 [==============================] - 96s 737us/step - loss:
0.0902 - acc: 0.9759 - val_loss: 0.8600 - val_acc: 0.8869
Epoch 56/80
130000/130000 [==============================] - 95s 734us/step - loss:
0.0914 - acc: 0.9755 - val_loss: 0.8576 - val_acc: 0.8897
Epoch 57/80
 83776/130000 [==================>...........] - ETA: 33s - loss: 0.088
3 - acc: 0.9761Buffered data was truncated after reaching the output si
ze limit.
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
```

```
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,66))

# Validation loss

vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 88.89%
```



In [0]:
```
import keras as keras
```

```python
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=(26),activation='relu',input_shape=(20
0,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.5))
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=60,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
W0618 17:02:57.990099 140483376052096 nn_ops.py:4224] Large dropout rat
e: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/60
130000/130000 [==============================] - 63s 484us/step - loss:
6.9390 - acc: 0.0836 - val_loss: 4.6362 - val_acc: 0.3158
Epoch 2/60
130000/130000 [==============================] - 60s 461us/step - loss:
4.3252 - acc: 0.3231 - val_loss: 2.6183 - val_acc: 0.5752
Epoch 3/60
130000/130000 [==============================] - 60s 462us/step - loss:
2.9310 - acc: 0.4804 - val_loss: 1.8188 - val_acc: 0.6942
Epoch 4/60
130000/130000 [==============================] - 60s 463us/step - loss:
2.1949 - acc: 0.5759 - val_loss: 1.4445 - val_acc: 0.7467
Epoch 5/60
130000/130000 [==============================] - 60s 462us/step - loss:
1.7564 - acc: 0.6377 - val_loss: 1.2535 - val_acc: 0.7788
Epoch 6/60
130000/130000 [==============================] - 60s 462us/step - loss:
```

```
1.4390 - acc: 0.6879 - val_loss: 1.1255 - val_acc: 0.8014
Epoch 7/60
130000/130000 [==============================] - 60s 461us/step - loss:
1.2149 - acc: 0.7254 - val_loss: 1.0493 - val_acc: 0.8143
Epoch 8/60
130000/130000 [==============================] - 60s 461us/step - loss:
1.0434 - acc: 0.7577 - val_loss: 0.9952 - val_acc: 0.8243
Epoch 9/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.9207 - acc: 0.7793 - val_loss: 0.9584 - val_acc: 0.8323
Epoch 10/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.8185 - acc: 0.7998 - val_loss: 0.9210 - val_acc: 0.8423
Epoch 11/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.7371 - acc: 0.8177 - val_loss: 0.8977 - val_acc: 0.8477
Epoch 12/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.6679 - acc: 0.8312 - val_loss: 0.8848 - val_acc: 0.8504
Epoch 13/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.6205 - acc: 0.8411 - val_loss: 0.8671 - val_acc: 0.8534
Epoch 14/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.5728 - acc: 0.8515 - val_loss: 0.8536 - val_acc: 0.8585
Epoch 15/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.5408 - acc: 0.8592 - val_loss: 0.8465 - val_acc: 0.8614
Epoch 16/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.5023 - acc: 0.8678 - val_loss: 0.8340 - val_acc: 0.8633
Epoch 17/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.4788 - acc: 0.8731 - val_loss: 0.8375 - val_acc: 0.8637
Epoch 18/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.4502 - acc: 0.8786 - val_loss: 0.8299 - val_acc: 0.8683
Epoch 19/60
130000/130000 [==============================] - 60s 462us/step - loss:
```

```
0.4278 - acc: 0.8847 - val_loss: 0.8158 - val_acc: 0.8703
Epoch 20/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.4183 - acc: 0.8872 - val_loss: 0.8119 - val_acc: 0.8694
Epoch 21/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.3872 - acc: 0.8940 - val_loss: 0.8085 - val_acc: 0.8741
Epoch 22/60
130000/130000 [==============================] - 60s 460us/step - loss:
0.3785 - acc: 0.8960 - val_loss: 0.8044 - val_acc: 0.8760
Epoch 23/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.3663 - acc: 0.9000 - val_loss: 0.7931 - val_acc: 0.8763
Epoch 24/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.3512 - acc: 0.9044 - val_loss: 0.7933 - val_acc: 0.8755
Epoch 25/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.3425 - acc: 0.9045 - val_loss: 0.7896 - val_acc: 0.8770
Epoch 26/60
130000/130000 [==============================] - 60s 460us/step - loss:
0.3285 - acc: 0.9098 - val_loss: 0.7876 - val_acc: 0.8783
Epoch 27/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.3170 - acc: 0.9116 - val_loss: 0.7894 - val_acc: 0.8776
Epoch 28/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.3065 - acc: 0.9144 - val_loss: 0.7886 - val_acc: 0.8787
Epoch 29/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.3018 - acc: 0.9168 - val_loss: 0.7785 - val_acc: 0.8802
Epoch 30/60
130000/130000 [==============================] - 60s 464us/step - loss:
0.2937 - acc: 0.9188 - val_loss: 0.7768 - val_acc: 0.8818
Epoch 31/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.2796 - acc: 0.9221 - val_loss: 0.7875 - val_acc: 0.8797
Epoch 32/60
130000/130000 [==============================] - 60s 462us/step - loss:
```

```
0.2782 - acc: 0.9229 - val_loss: 0.7841 - val_acc: 0.8820
Epoch 33/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.2666 - acc: 0.9250 - val_loss: 0.7796 - val_acc: 0.8818
Epoch 34/60
130000/130000 [==============================] - 60s 465us/step - loss:
0.2644 - acc: 0.9255 - val_loss: 0.7733 - val_acc: 0.8841
Epoch 35/60
130000/130000 [==============================] - 60s 465us/step - loss:
0.2563 - acc: 0.9283 - val_loss: 0.7663 - val_acc: 0.8838
Epoch 36/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.2503 - acc: 0.9292 - val_loss: 0.7710 - val_acc: 0.8845
Epoch 37/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.2434 - acc: 0.9309 - val_loss: 0.7629 - val_acc: 0.8854
Epoch 38/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.2353 - acc: 0.9340 - val_loss: 0.7754 - val_acc: 0.8861
Epoch 39/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.2319 - acc: 0.9354 - val_loss: 0.7614 - val_acc: 0.8864
Epoch 40/60
130000/130000 [==============================] - 60s 463us/step - loss:
0.2229 - acc: 0.9369 - val_loss: 0.7643 - val_acc: 0.8881
Epoch 41/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.2258 - acc: 0.9369 - val_loss: 0.7774 - val_acc: 0.8855
Epoch 42/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.2225 - acc: 0.9371 - val_loss: 0.7653 - val_acc: 0.8879
Epoch 43/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.2145 - acc: 0.9405 - val_loss: 0.7537 - val_acc: 0.8883
Epoch 44/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.2090 - acc: 0.9409 - val_loss: 0.7583 - val_acc: 0.8892
Epoch 45/60
130000/130000 [==============================] - 60s 462us/step - loss:
```

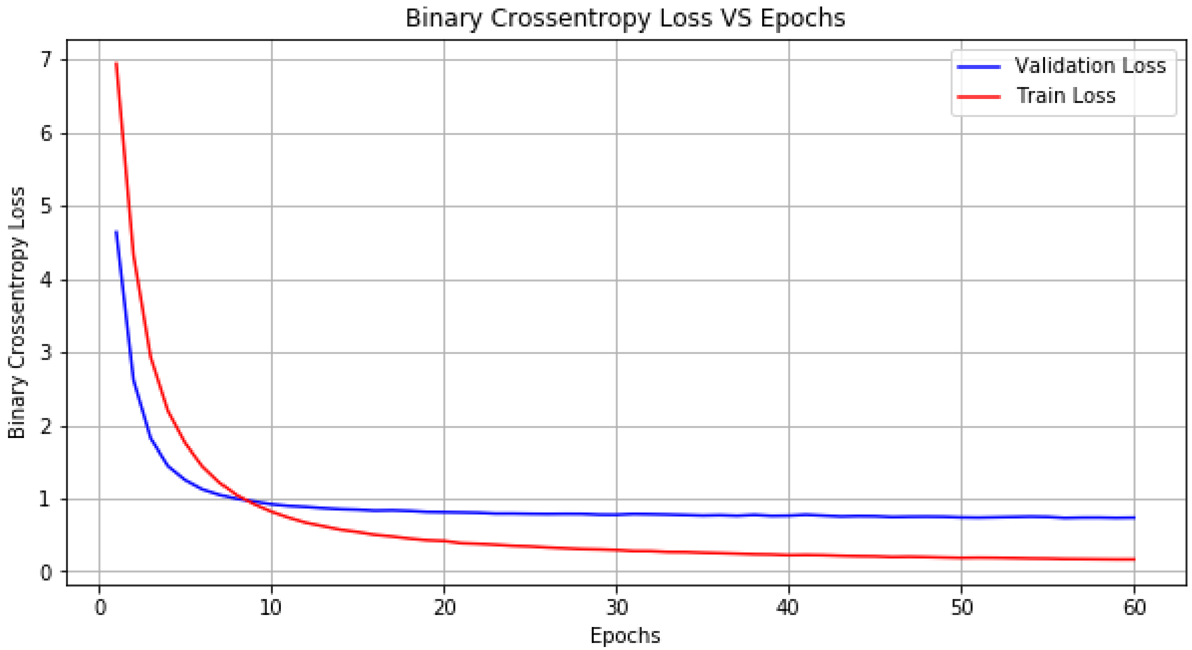
```
0.2064 - acc: 0.9416 - val_loss: 0.7566 - val_acc: 0.8908
Epoch 46/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1980 - acc: 0.9438 - val_loss: 0.7459 - val_acc: 0.8921
Epoch 47/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.2013 - acc: 0.9433 - val_loss: 0.7500 - val_acc: 0.8924
Epoch 48/60
130000/130000 [==============================] - 60s 460us/step - loss:
0.1967 - acc: 0.9440 - val_loss: 0.7514 - val_acc: 0.8904
Epoch 49/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1909 - acc: 0.9455 - val_loss: 0.7493 - val_acc: 0.8915
Epoch 50/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1859 - acc: 0.9466 - val_loss: 0.7406 - val_acc: 0.8928
Epoch 51/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1883 - acc: 0.9466 - val_loss: 0.7377 - val_acc: 0.8928
Epoch 52/60
130000/130000 [==============================] - 60s 460us/step - loss:
0.1864 - acc: 0.9478 - val_loss: 0.7423 - val_acc: 0.8935
Epoch 53/60
130000/130000 [==============================] - 60s 461us/step - loss:
0.1824 - acc: 0.9482 - val_loss: 0.7467 - val_acc: 0.8924
Epoch 54/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1782 - acc: 0.9497 - val_loss: 0.7514 - val_acc: 0.8924
Epoch 55/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1761 - acc: 0.9503 - val_loss: 0.7476 - val_acc: 0.8924
Epoch 56/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1707 - acc: 0.9520 - val_loss: 0.7331 - val_acc: 0.8934
Epoch 57/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1691 - acc: 0.9516 - val_loss: 0.7376 - val_acc: 0.8932
Epoch 58/60
130000/130000 [==============================] - 60s 462us/step - loss:
```

```
0.1672 - acc: 0.9517 - val_loss: 0.7382 - val_acc: 0.8957
Epoch 59/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1652 - acc: 0.9532 - val_loss: 0.7340 - val_acc: 0.8971
Epoch 60/60
130000/130000 [==============================] - 60s 462us/step - loss:
0.1650 - acc: 0.9524 - val_loss: 0.7369 - val_acc: 0.8960
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 89.60%
```



In [0]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(1500,kernel_size=21,activation='relu',input_shape=(200
,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.5))
model.add(BatchNormalization())
model.add(Conv1D(500,26,dilation_rate=2,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Conv1D(512,1,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.4))
```

```
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=100,validation_data=[testar
ray1,testy],batch_size=256)
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/100
130000/130000 [==============================] - 233s 2ms/step - loss:
6.5850 - acc: 0.1133 - val_loss: 4.0728 - val_acc: 0.3664
Epoch 2/100
130000/130000 [==============================] - 221s 2ms/step - loss:
3.2637 - acc: 0.4496 - val_loss: 2.0201 - val_acc: 0.6458
Epoch 3/100
130000/130000 [==============================] - 220s 2ms/step - loss:
1.7078 - acc: 0.6577 - val_loss: 1.3805 - val_acc: 0.7435
Epoch 4/100
130000/130000 [==============================] - 220s 2ms/step - loss:
1.0127 - acc: 0.7693 - val_loss: 1.0908 - val_acc: 0.7903
Epoch 5/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.6610 - acc: 0.8369 - val_loss: 0.9763 - val_acc: 0.8131
Epoch 6/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.4879 - acc: 0.8733 - val_loss: 0.9408 - val_acc: 0.8196
Epoch 7/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.3828 - acc: 0.8969 - val_loss: 0.8822 - val_acc: 0.8310
Epoch 8/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.3162 - acc: 0.9114 - val_loss: 0.8725 - val_acc: 0.8351
Epoch 9/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.2695 - acc: 0.9239 - val_loss: 0.8530 - val_acc: 0.8440
Epoch 10/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.2446 - acc: 0.9300 - val_loss: 0.8557 - val_acc: 0.8419
Epoch 11/100
```

```
130000/130000 [==============================] - 220s 2ms/step - loss:
0.2211 - acc: 0.9363 - val_loss: 0.8405 - val_acc: 0.8483
Epoch 12/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1988 - acc: 0.9425 - val_loss: 0.8408 - val_acc: 0.8520
Epoch 13/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1876 - acc: 0.9452 - val_loss: 0.8326 - val_acc: 0.8523
Epoch 14/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1750 - acc: 0.9496 - val_loss: 0.8308 - val_acc: 0.8574
Epoch 15/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1687 - acc: 0.9505 - val_loss: 0.8251 - val_acc: 0.8560
Epoch 16/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1543 - acc: 0.9547 - val_loss: 0.8130 - val_acc: 0.8601
Epoch 17/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1448 - acc: 0.9578 - val_loss: 0.8132 - val_acc: 0.8623
Epoch 18/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1392 - acc: 0.9587 - val_loss: 0.8004 - val_acc: 0.8649
Epoch 19/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1340 - acc: 0.9605 - val_loss: 0.8170 - val_acc: 0.8648
Epoch 20/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1300 - acc: 0.9628 - val_loss: 0.8300 - val_acc: 0.8640
Epoch 21/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1245 - acc: 0.9630 - val_loss: 0.8126 - val_acc: 0.8669
Epoch 22/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1153 - acc: 0.9661 - val_loss: 0.8035 - val_acc: 0.8697
Epoch 23/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1156 - acc: 0.9666 - val_loss: 0.8372 - val_acc: 0.8675
Epoch 24/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.1095 - acc: 0.9674 - val_loss: 0.8409 - val_acc: 0.8671
Epoch 25/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.1080 - acc: 0.9684 - val_loss: 0.8288 - val_acc: 0.8711
Epoch 26/100
130000/130000 [==============================] - 221s 2ms/step - loss:
0.1034 - acc: 0.9698 - val_loss: 0.8144 - val_acc: 0.8722
Epoch 27/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0979 - acc: 0.9713 - val_loss: 0.8275 - val_acc: 0.8722
Epoch 28/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0979 - acc: 0.9712 - val_loss: 0.8265 - val_acc: 0.8711
Epoch 29/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0949 - acc: 0.9720 - val_loss: 0.8239 - val_acc: 0.8712
Epoch 30/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0903 - acc: 0.9736 - val_loss: 0.8343 - val_acc: 0.8719
Epoch 31/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0860 - acc: 0.9751 - val_loss: 0.8114 - val_acc: 0.8744
Epoch 32/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0844 - acc: 0.9750 - val_loss: 0.8240 - val_acc: 0.8759
Epoch 33/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0809 - acc: 0.9760 - val_loss: 0.8141 - val_acc: 0.8765
Epoch 34/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0840 - acc: 0.9752 - val_loss: 0.8193 - val_acc: 0.8747
Epoch 35/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0784 - acc: 0.9772 - val_loss: 0.8078 - val_acc: 0.8781
Epoch 36/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0772 - acc: 0.9774 - val_loss: 0.8039 - val_acc: 0.8791
Epoch 37/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.0778 - acc: 0.9775 - val_loss: 0.8066 - val_acc: 0.8803
Epoch 38/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0749 - acc: 0.9784 - val_loss: 0.8236 - val_acc: 0.8801
Epoch 39/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0733 - acc: 0.9788 - val_loss: 0.8207 - val_acc: 0.8794
Epoch 40/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0713 - acc: 0.9795 - val_loss: 0.8178 - val_acc: 0.8803
Epoch 41/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0689 - acc: 0.9799 - val_loss: 0.8259 - val_acc: 0.8796
Epoch 42/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0686 - acc: 0.9804 - val_loss: 0.8047 - val_acc: 0.8827
Epoch 43/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0682 - acc: 0.9804 - val_loss: 0.8131 - val_acc: 0.8815
Epoch 44/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0675 - acc: 0.9807 - val_loss: 0.8176 - val_acc: 0.8826
Epoch 45/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0646 - acc: 0.9813 - val_loss: 0.8021 - val_acc: 0.8826
Epoch 46/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0627 - acc: 0.9823 - val_loss: 0.8121 - val_acc: 0.8821
Epoch 47/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0601 - acc: 0.9827 - val_loss: 0.8150 - val_acc: 0.8822
Epoch 48/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0602 - acc: 0.9829 - val_loss: 0.8108 - val_acc: 0.8814
Epoch 49/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0581 - acc: 0.9834 - val_loss: 0.8134 - val_acc: 0.8820
Epoch 50/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.0590 - acc: 0.9832 - val_loss: 0.8152 - val_acc: 0.8842
Epoch 51/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0582 - acc: 0.9834 - val_loss: 0.8135 - val_acc: 0.8838
Epoch 52/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0567 - acc: 0.9838 - val_loss: 0.8171 - val_acc: 0.8846
Epoch 53/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0579 - acc: 0.9838 - val_loss: 0.8127 - val_acc: 0.8850
Epoch 54/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0541 - acc: 0.9842 - val_loss: 0.8073 - val_acc: 0.8851
Epoch 55/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0535 - acc: 0.9848 - val_loss: 0.8122 - val_acc: 0.8840
Epoch 56/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0555 - acc: 0.9847 - val_loss: 0.8182 - val_acc: 0.8858
Epoch 57/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0531 - acc: 0.9853 - val_loss: 0.8082 - val_acc: 0.8859
Epoch 58/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0511 - acc: 0.9857 - val_loss: 0.7998 - val_acc: 0.8863
Epoch 59/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0519 - acc: 0.9857 - val_loss: 0.8064 - val_acc: 0.8867
Epoch 60/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0509 - acc: 0.9856 - val_loss: 0.8140 - val_acc: 0.8846
Epoch 61/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0504 - acc: 0.9858 - val_loss: 0.8104 - val_acc: 0.8834
Epoch 62/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0497 - acc: 0.9860 - val_loss: 0.8116 - val_acc: 0.8868
Epoch 63/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.0493 - acc: 0.9858 - val_loss: 0.8004 - val_acc: 0.8871
Epoch 64/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0480 - acc: 0.9864 - val_loss: 0.8082 - val_acc: 0.8874
Epoch 65/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0437 - acc: 0.9876 - val_loss: 0.8151 - val_acc: 0.8877
Epoch 66/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0470 - acc: 0.9866 - val_loss: 0.7966 - val_acc: 0.8879
Epoch 67/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0453 - acc: 0.9872 - val_loss: 0.8057 - val_acc: 0.8887
Epoch 68/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0452 - acc: 0.9874 - val_loss: 0.8074 - val_acc: 0.8878
Epoch 69/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0447 - acc: 0.9875 - val_loss: 0.8000 - val_acc: 0.8892
Epoch 70/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0426 - acc: 0.9882 - val_loss: 0.8054 - val_acc: 0.8882
Epoch 71/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0437 - acc: 0.9878 - val_loss: 0.7996 - val_acc: 0.8881
Epoch 72/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0422 - acc: 0.9882 - val_loss: 0.8031 - val_acc: 0.8898
Epoch 73/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0425 - acc: 0.9882 - val_loss: 0.8017 - val_acc: 0.8912
Epoch 74/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0417 - acc: 0.9884 - val_loss: 0.8013 - val_acc: 0.8901
Epoch 75/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0427 - acc: 0.9880 - val_loss: 0.8003 - val_acc: 0.8920
Epoch 76/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.0430 - acc: 0.9882 - val_loss: 0.8002 - val_acc: 0.8924
Epoch 77/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0407 - acc: 0.9888 - val_loss: 0.7862 - val_acc: 0.8913
Epoch 78/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0396 - acc: 0.9888 - val_loss: 0.7895 - val_acc: 0.8913
Epoch 79/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0397 - acc: 0.9890 - val_loss: 0.7963 - val_acc: 0.8908
Epoch 80/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0386 - acc: 0.9894 - val_loss: 0.7989 - val_acc: 0.8905
Epoch 81/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0381 - acc: 0.9892 - val_loss: 0.8054 - val_acc: 0.8910
Epoch 82/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0393 - acc: 0.9891 - val_loss: 0.7988 - val_acc: 0.8911
Epoch 83/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0383 - acc: 0.9895 - val_loss: 0.7946 - val_acc: 0.8934
Epoch 84/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0360 - acc: 0.9898 - val_loss: 0.8105 - val_acc: 0.8912
Epoch 85/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0359 - acc: 0.9901 - val_loss: 0.7996 - val_acc: 0.8938
Epoch 86/100
130000/130000 [==============================] - 219s 2ms/step - loss:
0.0374 - acc: 0.9895 - val_loss: 0.7912 - val_acc: 0.8924
Epoch 87/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0360 - acc: 0.9902 - val_loss: 0.7915 - val_acc: 0.8948
Epoch 88/100
130000/130000 [==============================] - 219s 2ms/step - loss:
0.0361 - acc: 0.9898 - val_loss: 0.7991 - val_acc: 0.8948
Epoch 89/100
130000/130000 [==============================] - 220s 2ms/step - loss:
```

```
0.0377 - acc: 0.9894 - val_loss: 0.7881 - val_acc: 0.8951
Epoch 90/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0354 - acc: 0.9904 - val_loss: 0.7907 - val_acc: 0.8933
Epoch 91/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0365 - acc: 0.9904 - val_loss: 0.7820 - val_acc: 0.8940
Epoch 92/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0348 - acc: 0.9903 - val_loss: 0.7792 - val_acc: 0.8920
Epoch 93/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0329 - acc: 0.9908 - val_loss: 0.7893 - val_acc: 0.8932
Epoch 94/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0346 - acc: 0.9905 - val_loss: 0.7954 - val_acc: 0.8935
Epoch 95/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0333 - acc: 0.9909 - val_loss: 0.7850 - val_acc: 0.8938
Epoch 96/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0326 - acc: 0.9914 - val_loss: 0.7812 - val_acc: 0.8950
Epoch 97/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0331 - acc: 0.9909 - val_loss: 0.7753 - val_acc: 0.8949
Epoch 98/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0327 - acc: 0.9910 - val_loss: 0.7795 - val_acc: 0.8951
Epoch 99/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0311 - acc: 0.9914 - val_loss: 0.7846 - val_acc: 0.8947
Epoch 100/100
130000/130000 [==============================] - 220s 2ms/step - loss:
0.0312 - acc: 0.9918 - val_loss: 0.7741 - val_acc: 0.8962
```

In [0]:

```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
```

```
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM,Bidirectional
model=Sequential()
model.add(Conv1D(500,kernel_size=(26),activation='relu',input_shape=(20
0,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.5))
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.7))
model.add(Bidirectional(LSTM(300,dropout=0.4,return_sequences=True)))
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=60,validation_data=[testarr
ay1,testy],batch_size=128)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0619 07:20:06.596103 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:74: The name tf.get_default_graph is deprecated. Please use tf.com
pat.v1.get_default_graph instead.

W0619 07:20:06.654401 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

W0619 07:20:06.676386 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:4138: The name tf.random_uniform is deprecated. Please use tf.rand
om.uniform instead.

W0619 07:20:06.721345 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_
pool2d instead.
```

```
W0619 07:20:06.729445 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:133: The name tf.placeholder_with_default is deprecated. Please us
e tf.compat.v1.placeholder_with_default instead.

W0619 07:20:06.741749 140182523484032 deprecation.py:506] From /usr/loc
al/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:344
5: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
W0619 07:20:06.790762 140182523484032 nn_ops.py:4224] Large dropout rat
e: 0.7 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
W0619 07:20:07.862491 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The nam
e tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optim
izer instead.

W0619 07:20:07.891538 140182523484032 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3341: The name tf.log is deprecated. Please use tf.math.log instea
d.

W0619 07:20:08.062929 140182523484032 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/60
130000/130000 [==============================] - 349s 3ms/step - loss:
6.2155 - acc: 0.1286 - val_loss: 3.5157 - val_acc: 0.3762
Epoch 2/60
130000/130000 [==============================] - 340s 3ms/step - loss:
3.0838 - acc: 0.4326 - val_loss: 1.9675 - val_acc: 0.6191
Epoch 3/60
```

```
130000/130000 [==============================] - 339s 3ms/step - loss:
1.7447 - acc: 0.6145 - val_loss: 1.5395 - val_acc: 0.7067
Epoch 4/60
130000/130000 [==============================] - 339s 3ms/step - loss:
1.1114 - acc: 0.7286 - val_loss: 1.4451 - val_acc: 0.7418
Epoch 5/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.7962 - acc: 0.7921 - val_loss: 1.4290 - val_acc: 0.7565
Epoch 6/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.6128 - acc: 0.8333 - val_loss: 1.4331 - val_acc: 0.7676
Epoch 7/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.5084 - acc: 0.8577 - val_loss: 1.4481 - val_acc: 0.7764
Epoch 8/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.4364 - acc: 0.8759 - val_loss: 1.4578 - val_acc: 0.7810
Epoch 9/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.3888 - acc: 0.8885 - val_loss: 1.4733 - val_acc: 0.7853
Epoch 10/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.3509 - acc: 0.8988 - val_loss: 1.4891 - val_acc: 0.7907
Epoch 11/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.3227 - acc: 0.9067 - val_loss: 1.4949 - val_acc: 0.7913
Epoch 12/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2980 - acc: 0.9138 - val_loss: 1.5222 - val_acc: 0.7931
Epoch 13/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2831 - acc: 0.9173 - val_loss: 1.5176 - val_acc: 0.7964
Epoch 14/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2732 - acc: 0.9201 - val_loss: 1.5296 - val_acc: 0.7997
Epoch 15/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2556 - acc: 0.9246 - val_loss: 1.5402 - val_acc: 0.7997
Epoch 16/60
130000/130000 [==============================] - 340s 3ms/step - loss:
```

```
0.2444 - acc: 0.9286 - val_loss: 1.5830 - val_acc: 0.8005
Epoch 17/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2366 - acc: 0.9312 - val_loss: 1.5664 - val_acc: 0.8042
Epoch 18/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.2372 - acc: 0.9312 - val_loss: 1.5775 - val_acc: 0.8046
Epoch 19/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2278 - acc: 0.9343 - val_loss: 1.5857 - val_acc: 0.8061
Epoch 20/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2228 - acc: 0.9357 - val_loss: 1.6240 - val_acc: 0.8030
Epoch 21/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2140 - acc: 0.9391 - val_loss: 1.6130 - val_acc: 0.8078
Epoch 22/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.2165 - acc: 0.9390 - val_loss: 1.5987 - val_acc: 0.8099
Epoch 23/60
130000/130000 [==============================] - 337s 3ms/step - loss:
0.2104 - acc: 0.9390 - val_loss: 1.6263 - val_acc: 0.8075
Epoch 24/60
130000/130000 [==============================] - 338s 3ms/step - loss:
0.2052 - acc: 0.9418 - val_loss: 1.6465 - val_acc: 0.8088
Epoch 25/60
130000/130000 [==============================] - 338s 3ms/step - loss:
0.2041 - acc: 0.9415 - val_loss: 1.6627 - val_acc: 0.8090
Epoch 26/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.2080 - acc: 0.9418 - val_loss: 1.6815 - val_acc: 0.8046
Epoch 27/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2014 - acc: 0.9441 - val_loss: 1.6783 - val_acc: 0.8095
Epoch 28/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.2040 - acc: 0.9430 - val_loss: 1.6943 - val_acc: 0.8099
Epoch 29/60
130000/130000 [==============================] - 339s 3ms/step - loss:
```

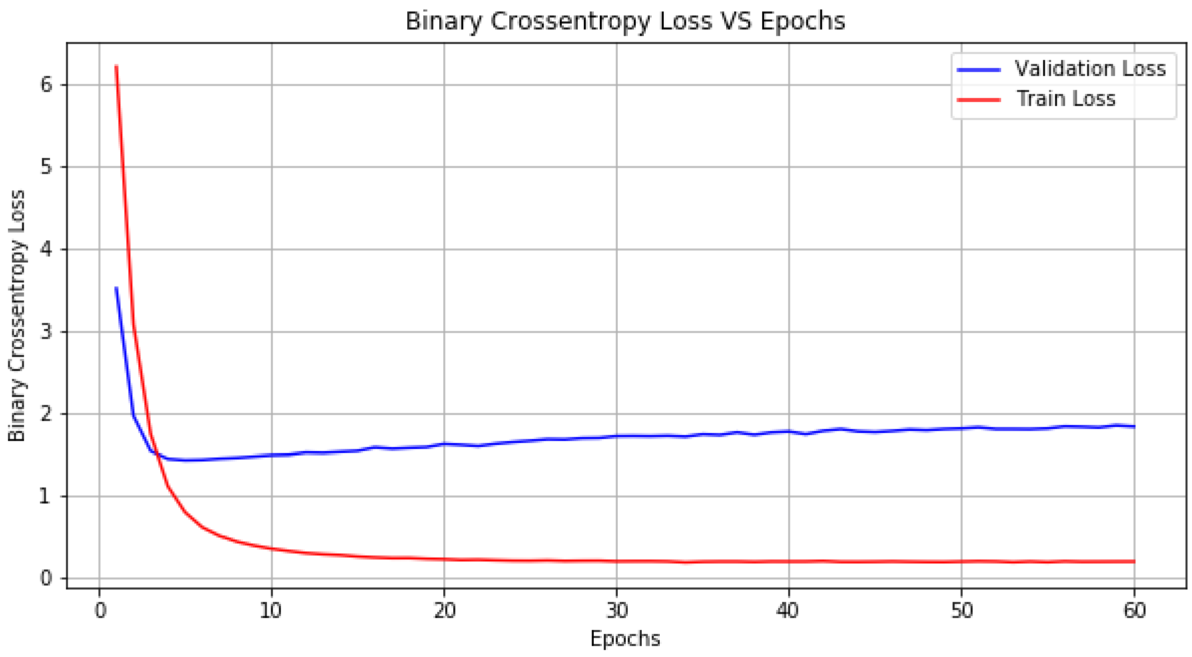
```
0.2047 - acc: 0.9438 - val_loss: 1.6972 - val_acc: 0.8105
Epoch 30/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1971 - acc: 0.9450 - val_loss: 1.7195 - val_acc: 0.8112
Epoch 31/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1986 - acc: 0.9454 - val_loss: 1.7212 - val_acc: 0.8119
Epoch 32/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1990 - acc: 0.9453 - val_loss: 1.7186 - val_acc: 0.8127
Epoch 33/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1960 - acc: 0.9467 - val_loss: 1.7237 - val_acc: 0.8124
Epoch 34/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1843 - acc: 0.9495 - val_loss: 1.7130 - val_acc: 0.8155
Epoch 35/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1923 - acc: 0.9481 - val_loss: 1.7422 - val_acc: 0.8147
Epoch 36/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1946 - acc: 0.9480 - val_loss: 1.7336 - val_acc: 0.8167
Epoch 37/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1947 - acc: 0.9477 - val_loss: 1.7657 - val_acc: 0.8124
Epoch 38/60
130000/130000 [==============================] - 343s 3ms/step - loss:
0.1906 - acc: 0.9494 - val_loss: 1.7356 - val_acc: 0.8168
Epoch 39/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1955 - acc: 0.9485 - val_loss: 1.7648 - val_acc: 0.8152
Epoch 40/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1948 - acc: 0.9490 - val_loss: 1.7760 - val_acc: 0.8155
Epoch 41/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1953 - acc: 0.9485 - val_loss: 1.7433 - val_acc: 0.8191
Epoch 42/60
130000/130000 [==============================] - 341s 3ms/step - loss:
```

```
0.2005 - acc: 0.9478 - val_loss: 1.7837 - val_acc: 0.8141
Epoch 43/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1910 - acc: 0.9501 - val_loss: 1.8055 - val_acc: 0.8152
Epoch 44/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1905 - acc: 0.9499 - val_loss: 1.7772 - val_acc: 0.8165
Epoch 45/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1924 - acc: 0.9496 - val_loss: 1.7689 - val_acc: 0.8193
Epoch 46/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1957 - acc: 0.9501 - val_loss: 1.7821 - val_acc: 0.8154
Epoch 47/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1924 - acc: 0.9503 - val_loss: 1.7999 - val_acc: 0.8162
Epoch 48/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1900 - acc: 0.9513 - val_loss: 1.7920 - val_acc: 0.8177
Epoch 49/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1890 - acc: 0.9515 - val_loss: 1.8059 - val_acc: 0.8164
Epoch 50/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1940 - acc: 0.9506 - val_loss: 1.8129 - val_acc: 0.8175
Epoch 51/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1983 - acc: 0.9498 - val_loss: 1.8269 - val_acc: 0.8174
Epoch 52/60
130000/130000 [==============================] - 342s 3ms/step - loss:
0.1956 - acc: 0.9512 - val_loss: 1.8062 - val_acc: 0.8207
Epoch 53/60
130000/130000 [==============================] - 340s 3ms/step - loss:
0.1870 - acc: 0.9521 - val_loss: 1.8060 - val_acc: 0.8187
Epoch 54/60
130000/130000 [==============================] - 339s 3ms/step - loss:
0.1947 - acc: 0.9513 - val_loss: 1.8046 - val_acc: 0.8191
Epoch 55/60
130000/130000 [==============================] - 338s 3ms/step - loss:
```

```
0.1873 - acc: 0.9536 - val_loss: 1.8131 - val_acc: 0.8185
Epoch 56/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1971 - acc: 0.9510 - val_loss: 1.8387 - val_acc: 0.8175
Epoch 57/60
130000/130000 [==============================] - 341s 3ms/step - loss:
0.1924 - acc: 0.9525 - val_loss: 1.8332 - val_acc: 0.8174
Epoch 58/60
 28800/130000 [=====>........................] - ETA: 4:17 - loss: 0.17
89 - acc: 0.9552Buffered data was truncated after reaching the output s
ize limit.
```

In [0]:
```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,61))

# Validation loss

vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 81.77%
```



In [0]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM,Bidirectional
model=Sequential()
model.add(Dense(200,activation='relu',input_shape=(200,26)))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Bidirectional(LSTM(26,dropout=0.2,return_sequences=True)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11180,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray1,trainy,epochs=2,validation_data=[testarra
y1,testy],batch_size=128)
```

```
Train on 130000 samples, validate on 20000 samples
Epoch 1/2
130000/130000 [==============================] - 766s 6ms/step - loss:
2.7829 - acc: 0.6851 - val_loss: 0.8493 - val_acc: 0.8886
Epoch 2/2
130000/130000 [==============================] - 761s 6ms/step - loss:
0.2382 - acc: 0.9801 - val_loss: 0.8404 - val_acc: 0.8921
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray1,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,3))

# Validation loss

vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 89.22%
```

** WITH 180000 POINTS AS TRIAINING DATA

In [0]:
```
import pandas as pd
import numpy as np
```

In [0]:
```
!pip install -U -q PyDrive
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
from google.colab import auth
from oauth2client.client import GoogleCredentials
# Authenticate and create the PyDrive client.
auth.authenticate_user()
gauth = GoogleAuth()
```

```
gauth.credentials = GoogleCredentials.get_application_default()
drive = GoogleDrive(gauth)
```

```
     |████████████████████████████████| 993kB 6.6MB/s
  Building wheel for PyDrive (setup.py) ... done
```

In [0]:
```
link = 'https://drive.google.com/open?id=1dRP2sKGX_8oOqlRSR2FQNUhneR3bfGrc' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
import pickle
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('trainarray.npy')
trainarray= np.load('trainarray.npy')
```

In [0]:
```
link = 'https://drive.google.com/open?id=1Bd2MrmGsDVmQZgt5xCphyhLWHctuOwG8' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('testarray.npy')
testarray= np.load('testarray.npy')
```

In [0]:
```
link = 'https://drive.google.com/open?id=1Di8VvI-XxNcTtcYIC3zk-lOO1TiOPU0q' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('cvarray.npy')
cvarray= np.load('cvarray.npy')
```

In [0]:
```
link = 'https://drive.google.com/open?id=1Jw6cyVYaJ-HL16Ox6X0Ab5KsqQxGE
```

```
ClR' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('traindataframeclasslabel.csv')
traindataframe= pd.read_csv('traindataframeclasslabel.csv',names=['a',
'b'])
```

In [0]:
```
link = 'https://drive.google.com/open?id=1hhWTqUSQ1TVQtvp-YN36E5sCx7XkP
Dh7' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('testdataframeclasslabel.csv')
testdataframe= pd.read_csv('testdataframeclasslabel.csv',names=['a','b'
])
```

In [0]:
```
link = 'https://drive.google.com/open?id=1BzVl-SWjGkQ1819BSQ9GbgHf-8PLc
JNH' # The shareable link
```

In [0]:
```
fluff, id = link.split('=')
```

In [0]:
```
downloaded = drive.CreateFile({'id':id})
downloaded.GetContentFile('cvdataframeclasslabel.csv')
cvdataframe= pd.read_csv('cvdataframeclasslabel.csv',names=['a','b'])
```

In [0]:
```
print(traindataframe.shape)
print(testdataframe.shape)
print(cvdataframe.shape)
```

```
(180000,)
(20000, 2)
(20000, 2)
```

In [0]:
```
print(traindataframe.head(3))
```

```
0     PF13376.6
1    PF01381.22
2    PF00300.22
Name: b, dtype: object
```

In [0]:
```
traindataframe=traindataframe['b']
```

In [0]:
```
testdataframe=testdataframe['b']
cvdataframe=cvdataframe['b']
```

In [0]:
```
totaldataframe=pd.concat([traindataframe,testdataframe],axis=0)
```

In [0]:
```
totaldataframe.head(3)
```

Out[0]:
```
0     PF13376.6
1    PF01381.22
2    PF00300.22
Name: b, dtype: object
```

In [0]:
```
from sklearn.feature_extraction.text import CountVectorizer
vect=CountVectorizer()
out=vect.fit_transform(totaldataframe)
features=vect.get_feature_names()
counts=out.sum(axis=0)
print('number of unique class lables are',len(features))
```

```
number of unique class lables are 11836
```

In [0]:
```
from keras.models import Sequential
from keras.layers import Flatten,Activation,Dense,Dropout
from keras.utils import np_utils
```

In [0]:
```
from sklearn.preprocessing import LabelEncoder
label1=LabelEncoder()
```

```
label1.fit(totaldataframe)
trainy=label1.transform(traindataframe)
testy=label1.transform(testdataframe)
#cvy=label1.transform(cvdataframe['family_accession'])
```

In [0]:
```
# this function is used draw Binary Crossentropy Loss VS No. of epochs
 plot
def plt_dynamic(x, vy, ty):
  plt.figure(figsize=(10,5))
  plt.plot(x, vy, 'b', label="Validation Loss")
  plt.plot(x, ty, 'r', label="Train Loss")
  plt.xlabel('Epochs')
  plt.ylabel('Binary Crossentropy Loss')
  plt.title('\nBinary Crossentropy Loss VS Epochs')
  plt.legend()
  plt.grid()
  plt.show()
```

In [0]:
```
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(100,kernel_size=(26),activation='relu',input_shape=(20
0,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.5))
model.add(Conv1D(250,26,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Flatten())
model.add(Dense(11836,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray,trainy,epochs=60,validation_data=[testarra
y,testy],batch_size=128)
```

```
WARNING: Logging before flag parsing goes to stderr.
W0619 10:52:49.821394 139727061636992 deprecation_wrapper.py:119] From
```

```
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:74: The name tf.get_default_graph is deprecated. Please use tf.com
pat.v1.get_default_graph instead.

W0619 10:52:49.864488 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

W0619 10:52:49.874052 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:4138: The name tf.random_uniform is deprecated. Please use tf.rand
om.uniform instead.

W0619 10:52:49.918748 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3976: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_
pool2d instead.

W0619 10:52:49.928320 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:133: The name tf.placeholder_with_default is deprecated. Please us
e tf.compat.v1.placeholder_with_default instead.

W0619 10:52:49.940415 139727061636992 deprecation.py:506] From /usr/loc
al/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:344
5: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
W0619 10:52:50.011578 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The nam
e tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optim
izer instead.

W0619 10:52:50.039674 139727061636992 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3341: The name tf.log is deprecated. Please use tf.math.log instea
d.
```

```
W0619 10:52:50.175851 139727061636992 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 180000 samples, validate on 20000 samples
Epoch 1/60
180000/180000 [==============================] - 157s 874us/step - los
s: 5.5257 - acc: 0.2277 - val_loss: 2.8999 - val_acc: 0.5254
Epoch 2/60
180000/180000 [==============================] - 149s 825us/step - los
s: 2.0066 - acc: 0.6391 - val_loss: 1.6602 - val_acc: 0.7128
Epoch 3/60
180000/180000 [==============================] - 149s 826us/step - los
s: 0.9869 - acc: 0.8030 - val_loss: 1.3317 - val_acc: 0.7723
Epoch 4/60
180000/180000 [==============================] - 148s 823us/step - los
s: 0.6012 - acc: 0.8794 - val_loss: 1.2425 - val_acc: 0.7936
Epoch 5/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.4280 - acc: 0.9166 - val_loss: 1.2195 - val_acc: 0.8114
Epoch 6/60
180000/180000 [==============================] - 148s 824us/step - los
s: 0.3434 - acc: 0.9367 - val_loss: 1.2495 - val_acc: 0.8144
Epoch 7/60
180000/180000 [==============================] - 148s 824us/step - los
s: 0.2966 - acc: 0.9475 - val_loss: 1.2795 - val_acc: 0.8261
Epoch 8/60
180000/180000 [==============================] - 148s 824us/step - los
s: 0.2669 - acc: 0.9563 - val_loss: 1.2914 - val_acc: 0.8247
Epoch 9/60
180000/180000 [==============================] - 148s 824us/step - los
s: 0.2537 - acc: 0.9602 - val_loss: 1.2948 - val_acc: 0.8323
Epoch 10/60
180000/180000 [==============================] - 147s 818us/step - los
s: 0.2397 - acc: 0.9646 - val_loss: 1.3121 - val_acc: 0.8372
```

```
Epoch 11/60
180000/180000 [==============================] - 147s 819us/step - los
s: 0.2320 - acc: 0.9670 - val_loss: 1.3329 - val_acc: 0.8395
Epoch 12/60
180000/180000 [==============================] - 147s 816us/step - los
s: 0.2240 - acc: 0.9692 - val_loss: 1.3685 - val_acc: 0.8390
Epoch 13/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2178 - acc: 0.9714 - val_loss: 1.4026 - val_acc: 0.8408
Epoch 14/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2165 - acc: 0.9724 - val_loss: 1.3675 - val_acc: 0.8469
Epoch 15/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.2119 - acc: 0.9736 - val_loss: 1.4069 - val_acc: 0.8476
Epoch 16/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.2079 - acc: 0.9748 - val_loss: 1.4261 - val_acc: 0.8462
Epoch 17/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2062 - acc: 0.9758 - val_loss: 1.4319 - val_acc: 0.8493
Epoch 18/60
180000/180000 [==============================] - 148s 823us/step - los
s: 0.2052 - acc: 0.9765 - val_loss: 1.4688 - val_acc: 0.8476
Epoch 19/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2040 - acc: 0.9770 - val_loss: 1.4723 - val_acc: 0.8484
Epoch 20/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2015 - acc: 0.9774 - val_loss: 1.4961 - val_acc: 0.8513
Epoch 21/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.2020 - acc: 0.9780 - val_loss: 1.4971 - val_acc: 0.8522
Epoch 22/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1988 - acc: 0.9788 - val_loss: 1.5217 - val_acc: 0.8515
Epoch 23/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.2021 - acc: 0.9788 - val_loss: 1.5570 - val_acc: 0.8507
Epoch 24/60
```

```
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1988 - acc: 0.9796 - val_loss: 1.5296 - val_acc: 0.8526
Epoch 25/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.1992 - acc: 0.9794 - val_loss: 1.5206 - val_acc: 0.8585
Epoch 26/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1963 - acc: 0.9803 - val_loss: 1.5375 - val_acc: 0.8596
Epoch 27/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2003 - acc: 0.9802 - val_loss: 1.5813 - val_acc: 0.8574
Epoch 28/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1967 - acc: 0.9810 - val_loss: 1.5897 - val_acc: 0.8569
Epoch 29/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1982 - acc: 0.9808 - val_loss: 1.6297 - val_acc: 0.8561
Epoch 30/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1981 - acc: 0.9815 - val_loss: 1.5936 - val_acc: 0.8610
Epoch 31/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1931 - acc: 0.9823 - val_loss: 1.6144 - val_acc: 0.8587
Epoch 32/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1961 - acc: 0.9820 - val_loss: 1.6397 - val_acc: 0.8577
Epoch 33/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1988 - acc: 0.9815 - val_loss: 1.6595 - val_acc: 0.8570
Epoch 34/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.1979 - acc: 0.9819 - val_loss: 1.6654 - val_acc: 0.8568
Epoch 35/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1973 - acc: 0.9824 - val_loss: 1.6312 - val_acc: 0.8628
Epoch 36/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.1994 - acc: 0.9818 - val_loss: 1.6544 - val_acc: 0.8636
Epoch 37/60
```

```
180000/180000 [==============================] - 148s 822us/step - los
s: 0.1984 - acc: 0.9824 - val_loss: 1.7020 - val_acc: 0.8570
Epoch 38/60
180000/180000 [==============================] - 148s 822us/step - los
s: 0.2005 - acc: 0.9821 - val_loss: 1.6565 - val_acc: 0.8636
Epoch 39/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1992 - acc: 0.9828 - val_loss: 1.6592 - val_acc: 0.8639
Epoch 40/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1964 - acc: 0.9831 - val_loss: 1.7462 - val_acc: 0.8600
Epoch 41/60
180000/180000 [==============================] - 148s 821us/step - los
s: 0.1998 - acc: 0.9827 - val_loss: 1.7084 - val_acc: 0.8648
Epoch 42/60
180000/180000 [==============================] - 148s 820us/step - los
s: 0.1971 - acc: 0.9835 - val_loss: 1.7081 - val_acc: 0.8640
Epoch 43/60
180000/180000 [==============================] - 147s 819us/step - los
s: 0.1990 - acc: 0.9831 - val_loss: 1.7628 - val_acc: 0.8606
Epoch 44/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.1964 - acc: 0.9837 - val_loss: 1.7827 - val_acc: 0.8612
Epoch 45/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.1979 - acc: 0.9835 - val_loss: 1.7522 - val_acc: 0.8627
Epoch 46/60
180000/180000 [==============================] - 146s 810us/step - los
s: 0.2000 - acc: 0.9834 - val_loss: 1.7651 - val_acc: 0.8635
Epoch 47/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.1979 - acc: 0.9838 - val_loss: 1.7817 - val_acc: 0.8636
Epoch 48/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.1974 - acc: 0.9841 - val_loss: 1.7953 - val_acc: 0.8606
Epoch 49/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.2005 - acc: 0.9837 - val_loss: 1.7954 - val_acc: 0.8637
Epoch 50/60
```

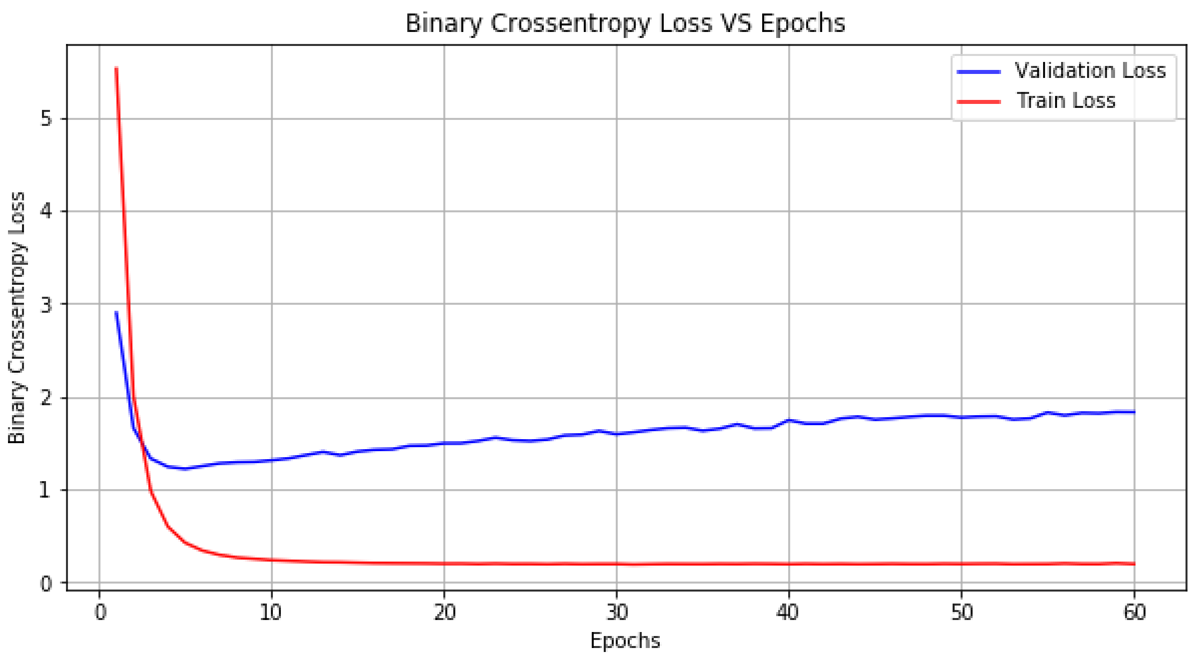
```
180000/180000 [==============================] - 146s 812us/step - los
s: 0.1989 - acc: 0.9841 - val_loss: 1.7745 - val_acc: 0.8635
Epoch 51/60
180000/180000 [==============================] - 147s 814us/step - los
s: 0.2007 - acc: 0.9841 - val_loss: 1.7846 - val_acc: 0.8643
Epoch 52/60
180000/180000 [==============================] - 146s 811us/step - los
s: 0.2015 - acc: 0.9840 - val_loss: 1.7881 - val_acc: 0.8660
Epoch 53/60
180000/180000 [==============================] - 146s 812us/step - los
s: 0.1968 - acc: 0.9846 - val_loss: 1.7539 - val_acc: 0.8689
Epoch 54/60
180000/180000 [==============================] - 147s 815us/step - los
s: 0.1975 - acc: 0.9847 - val_loss: 1.7651 - val_acc: 0.8690
Epoch 55/60
180000/180000 [==============================] - 147s 816us/step - los
s: 0.1984 - acc: 0.9847 - val_loss: 1.8288 - val_acc: 0.8648
Epoch 56/60
180000/180000 [==============================] - 146s 810us/step - los
s: 0.2038 - acc: 0.9841 - val_loss: 1.7976 - val_acc: 0.8675
Epoch 57/60
180000/180000 [==============================] - 147s 816us/step - los
s: 0.1988 - acc: 0.9846 - val_loss: 1.8241 - val_acc: 0.8656
Epoch 58/60
180000/180000 [==============================] - 147s 814us/step - los
s: 0.1984 - acc: 0.9849 - val_loss: 1.8200 - val_acc: 0.8680
Epoch 59/60
180000/180000 [==============================] - 146s 814us/step - los
s: 0.2062 - acc: 0.9843 - val_loss: 1.8341 - val_acc: 0.8671
Epoch 60/60
180000/180000 [==============================] - 147s 816us/step - los
s: 0.1988 - acc: 0.9849 - val_loss: 1.8329 - val_acc: 0.8682
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))
```

```
# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,61))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 86.83%
```



```
In [0]: import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
model=Sequential()
model.add(Conv1D(2000,kernel_size=26,activation='relu',input_shape=(200
,26)))
model.add(MaxPooling1D(pool_size=2))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Conv1D(250,20,activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Dropout(0.6))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11836,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
history=model.fit(trainarray,trainy,epochs=30,validation_data=[testarra
y,testy],batch_size=128)
```

```
W0619 19:50:21.398618 139644765677440 nn_ops.py:4224] Large dropout rat
e: 0.6 (>0.5). In TensorFlow 2.x, dropout() uses dropout rate instead o
f keep_prob. Please ensure that this is intended.
W0619 19:50:21.626489 139644765677440 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

```
Train on 180000 samples, validate on 20000 samples
Epoch 1/30
180000/180000 [==============================] - 329s 2ms/step - loss:
4.6548 - acc: 0.3767 - val_loss: 1.5470 - val_acc: 0.7329
Epoch 2/30
180000/180000 [==============================] - 343s 2ms/step - loss:
1.2378 - acc: 0.7697 - val_loss: 0.9925 - val_acc: 0.8195
Epoch 3/30
```

```
180000/180000 [==============================] - 342s 2ms/step - loss:
0.5234 - acc: 0.8860 - val_loss: 0.9045 - val_acc: 0.8399
Epoch 4/30
180000/180000 [==============================] - 342s 2ms/step - loss:
0.3116 - acc: 0.9275 - val_loss: 1.1124 - val_acc: 0.8037
Epoch 5/30
180000/180000 [==============================] - 341s 2ms/step - loss:
0.2337 - acc: 0.9440 - val_loss: 0.7667 - val_acc: 0.8640
Epoch 6/30
180000/180000 [==============================] - 326s 2ms/step - loss:
0.1846 - acc: 0.9558 - val_loss: 0.6613 - val_acc: 0.8891
Epoch 7/30
180000/180000 [==============================] - 326s 2ms/step - loss:
0.1633 - acc: 0.9608 - val_loss: 0.6504 - val_acc: 0.8899
Epoch 8/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.1391 - acc: 0.9667 - val_loss: 1.0141 - val_acc: 0.8384
Epoch 9/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.1272 - acc: 0.9692 - val_loss: 0.6761 - val_acc: 0.8894
Epoch 10/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.1172 - acc: 0.9721 - val_loss: 0.5658 - val_acc: 0.9101
Epoch 11/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.1057 - acc: 0.9752 - val_loss: 0.6312 - val_acc: 0.9002
Epoch 12/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0990 - acc: 0.9768 - val_loss: 0.5767 - val_acc: 0.9083
Epoch 13/30
180000/180000 [==============================] - 326s 2ms/step - loss:
0.0937 - acc: 0.9780 - val_loss: 0.5718 - val_acc: 0.9096
Epoch 14/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0880 - acc: 0.9795 - val_loss: 1.0875 - val_acc: 0.8390
Epoch 15/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0838 - acc: 0.9805 - val_loss: 0.6391 - val_acc: 0.9012
Epoch 16/30
180000/180000 [==============================] - 325s 2ms/step - loss:
```

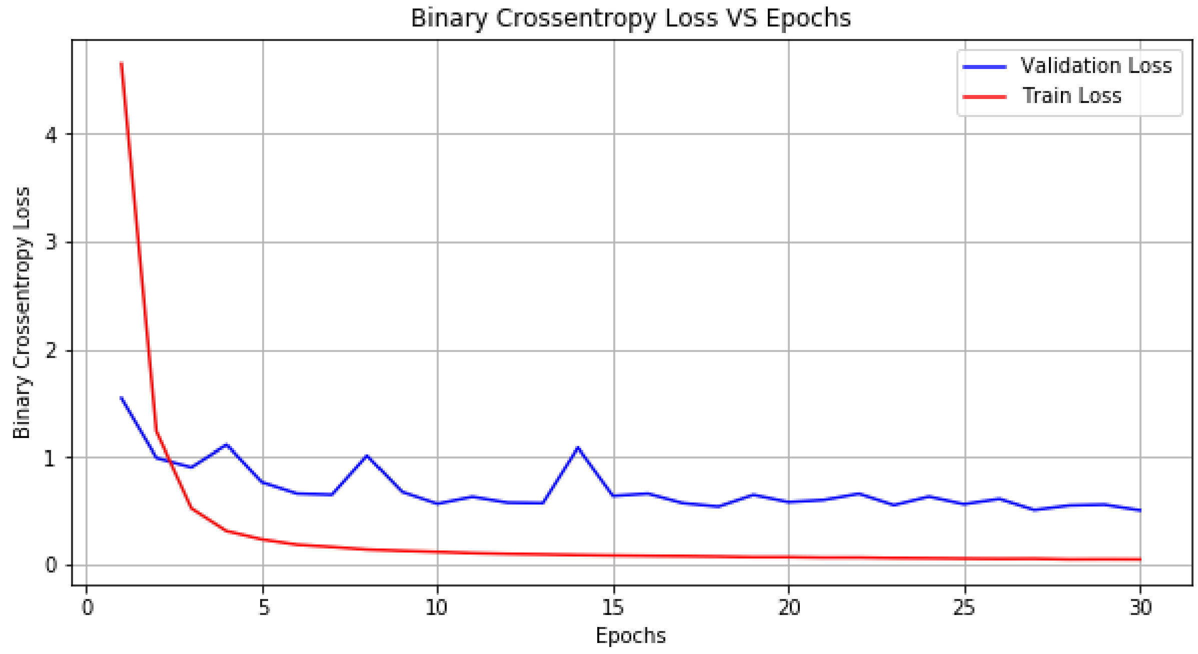
```
0.0806 - acc: 0.9816 - val_loss: 0.6606 - val_acc: 0.8958
Epoch 17/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0777 - acc: 0.9825 - val_loss: 0.5714 - val_acc: 0.9137
Epoch 18/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0739 - acc: 0.9832 - val_loss: 0.5397 - val_acc: 0.9212
Epoch 19/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0684 - acc: 0.9844 - val_loss: 0.6490 - val_acc: 0.9012
Epoch 20/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0686 - acc: 0.9845 - val_loss: 0.5817 - val_acc: 0.9124
Epoch 21/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0641 - acc: 0.9856 - val_loss: 0.6012 - val_acc: 0.9091
Epoch 22/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0644 - acc: 0.9857 - val_loss: 0.6598 - val_acc: 0.9002
Epoch 23/30
180000/180000 [==============================] - 326s 2ms/step - loss:
0.0598 - acc: 0.9868 - val_loss: 0.5533 - val_acc: 0.9185
Epoch 24/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0577 - acc: 0.9870 - val_loss: 0.6329 - val_acc: 0.9084
Epoch 25/30
180000/180000 [==============================] - 326s 2ms/step - loss:
0.0551 - acc: 0.9878 - val_loss: 0.5622 - val_acc: 0.9185
Epoch 26/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0528 - acc: 0.9883 - val_loss: 0.6116 - val_acc: 0.9100
Epoch 27/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0537 - acc: 0.9882 - val_loss: 0.5081 - val_acc: 0.9263
Epoch 28/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0474 - acc: 0.9896 - val_loss: 0.5505 - val_acc: 0.9192
Epoch 29/30
180000/180000 [==============================] - 325s 2ms/step - loss:
```

```
0.0484 - acc: 0.9894 - val_loss: 0.5581 - val_acc: 0.9175
Epoch 30/30
180000/180000 [==============================] - 325s 2ms/step - loss:
0.0474 - acc: 0.9897 - val_loss: 0.5054 - val_acc: 0.9285
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,31))

# Validation loss
vy = history.history['val_loss']
# Training loss
ty = history.history['loss']

# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 92.86%
```



In [0]:

```
#do bidirectional lstm in this
import keras as keras
import tensorflow as tf
from keras.models import Sequential
from keras.layers import Dense,Dropout,Flatten,BatchNormalization
from keras.layers import Conv1D,MaxPooling1D
from keras.layers import LSTM,Bidirectional
model=Sequential()
model.add(Dense(200,activation='relu',input_shape=(200,26)))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Bidirectional(LSTM(26,dropout=0.2,return_sequences=True)))
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(11836,activation='softmax'))
model.compile(loss=keras.losses.sparse_categorical_crossentropy,optimiz
er='adam',metrics=['accuracy'])
```

```
history=model.fit(trainarray,trainy,epochs=3,validation_data=[testarray
,testy],batch_size=128)
```

```
Using TensorFlow backend.
WARNING: Logging before flag parsing goes to stderr.
W0620 12:33:09.188477 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:74: The name tf.get_default_graph is deprecated. Please use tf.com
pat.v1.get_default_graph instead.

W0620 12:33:09.227867 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:517: The name tf.placeholder is deprecated. Please use tf.compat.v
1.placeholder instead.

W0620 12:33:09.237042 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:4138: The name tf.random_uniform is deprecated. Please use tf.rand
om.uniform instead.

W0620 12:33:09.270805 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:133: The name tf.placeholder_with_default is deprecated. Please us
e tf.compat.v1.placeholder_with_default instead.

W0620 12:33:09.283224 140488317740928 deprecation.py:506] From /usr/loc
al/lib/python3.6/dist-packages/keras/backend/tensorflow_backend.py:344
5: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob i
s deprecated and will be removed in a future version.
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate =
1 - keep_prob`.
W0620 12:33:10.224582 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/optimizers.py:790: The nam
e tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optim
izer instead.

W0620 12:33:10.256244 140488317740928 deprecation_wrapper.py:119] From
/usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_backen
d.py:3341: The name tf.log is deprecated. Please use tf.math.log instea
```

```
d.

W0620 12:33:10.411857 140488317740928 deprecation.py:323] From /usr/loc
al/lib/python3.6/dist-packages/tensorflow/python/ops/math_grad.py:1250:
add_dispatch_support.<locals>.wrapper (from tensorflow.python.ops.array
_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
```

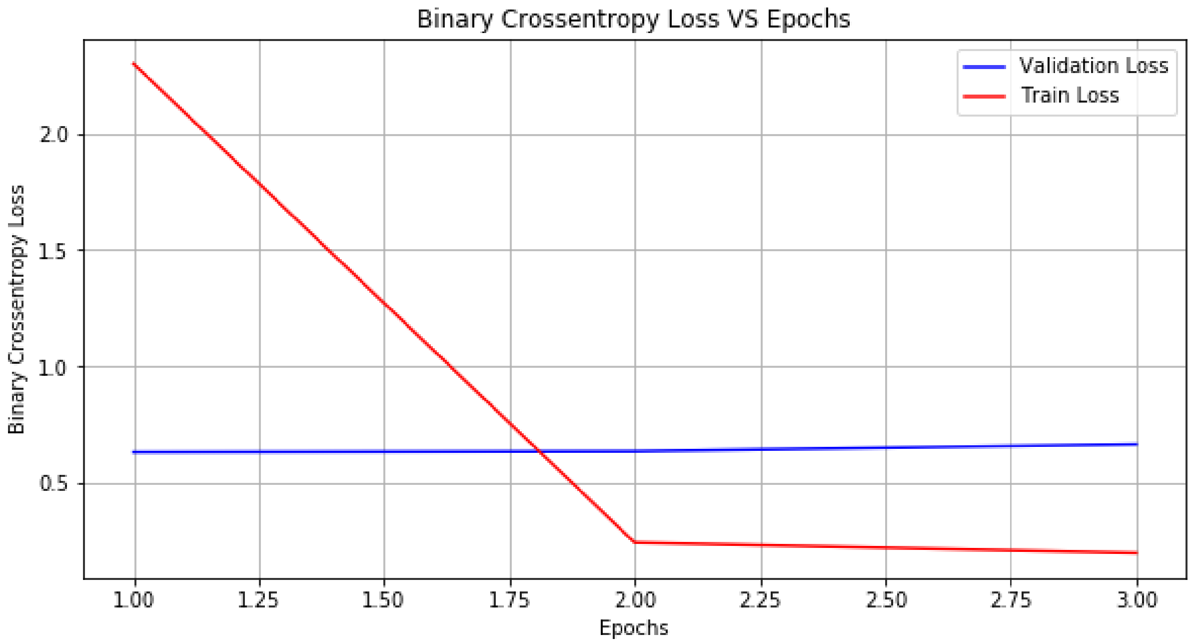
```
Train on 180000 samples, validate on 20000 samples
Epoch 1/3
180000/180000 [==============================] - 1538s 9ms/step - loss:
2.2988 - acc: 0.7371 - val_loss: 0.6325 - val_acc: 0.9112
Epoch 2/3
180000/180000 [==============================] - 1536s 9ms/step - loss:
0.2449 - acc: 0.9800 - val_loss: 0.6362 - val_acc: 0.9126
Epoch 3/3
180000/180000 [==============================] - 1511s 8ms/step - loss:
0.2006 - acc: 0.9867 - val_loss: 0.6668 - val_acc: 0.9104
```

In [0]:

```
import matplotlib.pyplot as plt
scores = model.evaluate(testarray,testy, verbose=0)
print("Accuracy: %.2f%%" % (scores[1]*100))

# Test and train accuracy of the model
model_3_test = scores[1]
model_3_train = max(history.history['acc'])

# Plotting Train and Test Loss VS no. of epochs
# list of epoch numbers
x = list(range(1,4))

# Validation loss

vy = history.history['val_loss']
# Training loss
ty = history.history['loss']
```

```
# Calling the function to draw the plot
plt_dynamic(x, vy, ty)
```

```
Accuracy: 91.04%
```



*** FOR 130000 POINTS.

In [18]:
```
import pandas as pd
s=[['CONV','70','128','(100,250)','(0.5,0.6)','(21,26)','0.98','0.89'],
['CONV+LSTM','70','128','(300,250,300(LSTM))','(0.4,0.6,0.4)','(21,26)'
,'0.98','0.88'],['DE+CONV+LSTM','70','256','(300,250,500(LSTM))','(0.7,
0.7,0.4)','(26,26)','0.98','0.86'],['(BI-DIR-LSTM)','5','128','(256,26
(LSTM))','(0.7,0.5)','NONE','0.99','0.86'],['CONV','25','256','(100,25
0','(0.2,0.2)','(10,20)','0.85','0.85'],['CONV','80','128','(2000,250)'
,'(0.4,0.6)','(26,20)','0.98','0.9'],['DEN+LSTM+DEN','60','128','(256,2
6,128)','(0.3,0.2,0.3)','NONE','0.85','0.88'],['CONVOLUTIONAL','25','25
6','(100,250)','(0.2,0.2)','(10,20)','0.98','0.87'],['CONVOLUTIONAL','8
```

```
0','64','(100,250)','(0.2,0.3)','(25,26)','0.85','0.89'],['CONVOLUTIONA
L','60','128','(100,250))','(0.5)','(26,26)','0.95','0.90'],['CONVOLUTI
ONAL','100','256','(1500,500,512)','(0.5,0.4,0.4)',"(21,26,1)",'0.99',
'0.9'],['CON-CON-BIDI-LSTM','60','128','(500,250,300(BI-LSTM))','(0.5,
0.7,0.4)','(26,26)','0.95','0.82'],['DENS+BIDIR-LSTM','2','128','(200,2
6)','(0.4,0.2)','NONE','0.98','0.89']]

x=pd.DataFrame(s,columns=['DL-TECHNIQUE','EPCHS','BCHSIZ','CONFIGURATIO
N','DRPOUT','KERN-SIZE','TRAIN ACC','TES-ACC'],index=None)
x
```

Out[18]:

	DL-TECHNIQUE	EPCHS	BCHSIZ	CONFIGURATION	DRPOUT	KERN-SIZE	TRAIN ACC	TES-ACC
0	CONV	70	128	(100,250)	(0.5,0.6)	(21,26)	0.98	0.89
1	CONV+LSTM	70	128	(300,250,300(LSTM))	(0.4,0.6,0.4)	(21,26)	0.98	0.88
2	DE+CONV+LSTM	70	256	(300,250,500(LSTM))	(0.7,0.7,0.4)	(26,26)	0.98	0.86
3	(BI-DIR-LSTM)	5	128	(256,26(LSTM))	(0.7,0.5)	NONE	0.99	0.86
4	CONV	25	256	(100,250	(0.2,0.2)	(10,20)	0.85	0.85
5	CONV	80	128	(2000,250)	(0.4,0.6)	(26,20)	0.98	0.9
6	DEN+LSTM+DEN	60	128	(256,26,128)	(0.3,0.2,0.3)	NONE	0.85	0.88
7	CONVOLUTIONAL	25	256	(100,250)	(0.2,0.2)	(10,20)	0.98	0.87
8	CONVOLUTIONAL	80	64	(100,250)	(0.2,0.3)	(25,26)	0.85	0.89
9	CONVOLUTIONAL	60	128	(100,250))	(0.5)	(26,26)	0.95	0.90
10	CONVOLUTIONAL	100	256	(1500,500,512)	(0.5,0.4,0.4)	(21,26,1)	0.99	0.9
11	CON-CON-BIDI-LSTM	60	128	(500,250,300(BI-LSTM))	(0.5,0.7,0.4)	(26,26)	0.95	0.82
12	DENS+BIDIR-LSTM	2	128	(200,26)	(0.4,0.2)	NONE	0.98	0.89

** FOR 180000 TRAINING SAMPLES

In [2]:
```
import pandas as pd
s=[['CONVOLUTIONAL','60','128','(100,250)','(0.5)','(26,26)','0.98','0.
86'],['CONVOLUTIONAL','30','128','(2000,250)','(0.4,0.6)','(26,20)','0.
99','0.93'],['DENSE+BIDIRECTIONAL-LSTM','2','128','(200,26)','(0.4,0.
2)','NONE','0.98','0.91']]

x=pd.DataFrame(s,columns=['DL-TECHNIQUE','EPOCHS','BATCHSIZ','CONFIG',
'DRPOUT-RATE','KERN-SIZE','TRAIN-ACC','TEST-ACC'])
x
```

Out[2]:

	DL-TECHNIQUE	EPOCHS	BATCHSIZ	CONFIG	DRPOUT-RATE	KERN-SIZE	TRAIN-ACC	TEST-ACC
0	CONVOLUTIONAL	60	128	(100,250)	(0.5)	(26,26)	0.98	0.86
1	CONVOLUTIONAL	30	128	(2000,250)	(0.4,0.6)	(26,20)	0.99	0.93
2	DENSE+BIDIRECTIONAL-LSTM	2	128	(200,26)	(0.4,0.2)	NONE	0.98	0.91

OBSERVATIONS DOCUMENTATION AND KEYTAKEAWAYS.

IN PFAM RANDOM SPLIT CASE STUDY WE ARE GIVEN WITH SEQUENCE AND WE HAVE TO PREDICT THE FMILY OF THE PROTEIN. WE HAVE TO PREDICT THE FAMILY PROTEIN.

WE HAVE THE INPUT FEATURES TO OUR MODEL .AMINO ACID IS THE SEQUENCE FOR THIS DOMAIN.

Description of fields: - sequence: These are usually the input features to your model. Amino acid sequence for this domain. There are 20 very common amino acids (frequency > 1,000,000), and 4 amino acids that are quite uncommon: X, U, B, O, Z. -

family_accession: These are usually the labels for your model. Accession number in form PFxxxxx.y (Pfam), where xxxxx is the family accession, and y is the version number.

WE HAVE DATA FOR TRAINING ,CROSS VALIDATION AND TESTING SEPERATELY. WE HAVE DONE EXPLORATORY DATA ANALTYSIS FOR DATA. WE HAVE 1100 CLASS LABELS WHICH ARE VERY FREQUENT.

OBSERVATIONS ARE FEW FAMILY ACCESSIONS ARE SIMILAR

FROM THE EXPLOARTORY DATA ANLYSIS HISTOGRAM PLOTS OF WE HAVE VISUALISED SEQUENCE LENGTH THE TRAIN DATA , TEST DATA AND CROSS VALIDATION DATA.

WE HAVE VISUALISED THE BOXPLOT AND VIOLIN PLOT FOR THE ALIGNMENT LENGTH.

WE HAVE PLOTTED THE PROBABILITY DENSITY FUNCTION AND CONSIDERD THE LENGTH OF SEQUENCE.WE HAVE CONSIDERD THE LENGTH OF 200 DUE TO LACK OF COMPUTATIONAL POWER. WE HAVE ALSO PLOTTED THE LENGTH OF SEQUENCE.

WE HAVE ENCODED EVERY LETTER OF SEQUENCE INTO LENGTH OF 26. WE HAVE OBTAINED THE SEQUENCE PATTERN AND DETERMINED THE CLASS LABEL.

THERE ARE FEW KEY TAKE AWAYS

- SEQUEENCE LENGTH IS ALMOST SIMILAR FOR THE PARTICULAR CLASS LABEL.
- THE SEQUENCE OF THE PROTEIN FOLLOW A PATTERN.

WE WILL TRAINED THE MODEL USING THE CONVOLUTIONAL NERAL NETWORS FROM THE FEATURES WE EXTRACT TO PREDICT THE CLASSLABEL.

OBSERVATIONS AFTER PERFORMING FEATURE EXTRACTION:-

** UNIQUE CLASS LABELS HAVE UNIQUE LENGTH OF SEQUENCE

** THE SEQUENCE WHICH ARE REPRESENTED BY THE UNDEFINED DOTS THEIR INDICES ARE ALMOST SIMILAR FOR THAT PARTICULAR CLASSLABEL.

THERE ARE VARIOUS RESEARCH PAPERS,BLOGS,YOUTUBE VIDEOS WHICH DESCRIBE THE REALTION OF PROTEIN SEQUENCES..

ALL THE ANLAYSIS WE HAVE DONE IS PART OF PROTEIN SEQUENCING TO PREDICT WHETHER THAT PROTEIN SEQUENCE BELONGS TO PARTICULAR FAMILY OR NOT.

WE HAVE USED VARIOUS CONFIGURATIONS WE HAVE OBTAINED THE MAXIMUM ACCURACY OF PERCENT.

WE HAVE USED RELU AS THE ACTIVATION FUNCTION FOR THE INPUT AND HIDDEN LAYERS AND SOFTMAX AS THE ACTIVATION FUNCTION TO PREDICT FOR THE OUTPUT LAYER BASED ON PROBABILTY.

WE CONSIDERD ACCURACY AS THE METRIC. WE HAVE PLOTTED THE GRAPHS FOR THE TRAINLOSS AND CROSS VALIDATION LOSS TO SEE THE HOW MODEL IS PERFORMING AND WHETHER THE MODEL IS OVERFITTING OR UNDERFITTING.

WE HAVE USED DROPUTS AND BATCH NORMALIZATION WHICH ARE USED TO PREVENT THE OVERFITTING OF THE MODEL.

USING THE DROPOUT LAYER WE CAN SWITCH OFF THE CELLS OF LAYER BASED ON THE PROBABILITY GIVEN.

WE HAVE TRAINED THE MODEL ON 130000 SAMPLES OF TRAIN DATA. WE HAVE USED VARIOUS DEEP LEARNING TECHNUIQUES FROM THE MULTI LAYER PERCEPTRONS ,CONVOLUTIONAL NUERAL NETWORKS,VARIOUS ARCHITECTURES OF CONVOLUTIONAL NYUERAL NETWROKS SPECIALLY INSPIRED BY RESIDUAL NETWORKS AND EXTRACTION OF BOTTLENECK FEATURES USING THE DILATION RATE AS 2.

WE HAVE ALSO DONE THE LSTM LAYER (LONG-SHORT-TERM MERMORY) AND THE BIDIRECTIONAL LSTM WITH VARYING THE NUMER OF EPOCHS,BATCH SIZE KERNEL SIZE(IN CASE OF CNN) WITH DIFFERENT DROPOUT RATES .WE HAVE USED BOTH CNN

ALONG WITH BIDIRECTIONAL LSTM AND LSTM IN SOME CASES. WE HAVE OBTAINED THE ACCURACY OF 90 PERCENTAGE USING THE CONVOLUTIONAL NUERAL NETS AND BIDIRECTIONAL LSTMS.

WE HAVE ALSO USED 180000 TRAINING SAMPLES DIRECTLY UPLOADED IN THE DRIVE PROCESSING IN LATOP OF 32 GB RAM. USING THE 180000 TRAINING SAMPLES WE HAVE ACHIEVED THE ACCURACY OF APPROXIMATELY 93 PERCENTAGE(0.928) WITH CONVOLUTIONAL NUERAL NETWORKS.

THIS ARE THE DEEPLEARNING TECHNIQUES THAT ARE USED WHICH ARE BETTER THEAN THE HIDDEN MARKOV MODELS TAKES LESS TIME AND COMPUTATION POWER AND ACHIEVE BEST RESULTS.

In [0]: