NAME: VEENA T.G.S

ROLL NO: 225229145

PDL Lab16. Design of LSTM and GRU RNN for classification of IMDB reviews

In [1]:
```
pip install scikit-learn
```

```
Defaulting to user installation because normal site-packages is not writea
ble
Requirement already satisfied: scikit-learn in c:\users\2mscdsa18\appdata
\roaming\python\python310\site-packages (1.3.1)
Requirement already satisfied: numpy<2.0,>=1.17.3 in c:\programdata\anacon
da3\envs\tf\lib\site-packages (from scikit-learn) (1.25.0)
Requirement already satisfied: scipy>=1.5.0 in c:\programdata\anaconda3\en
vs\tf\lib\site-packages (from scikit-learn) (1.10.1)
Requirement already satisfied: joblib>=1.1.1 in c:\users\2mscdsa18\appdata
\roaming\python\python310\site-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\2mscdsa18
\appdata\roaming\python\python310\site-packages (from scikit-learn) (3.2.
0)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import pandas as pd
import numpy as np
import nltk
from nltk.corpus import stopwords
from sklearn.model_selection import train_test_split
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense
from tensorflow.keras.callbacks import ModelCheckpoint
from tensorflow.keras.models import load_model
import re
from tensorflow.keras.layers import Bidirectional
```

In [3]:
```
data = pd.read_csv('IMDB Dataset.csv')

print(data)
```

```
                                                  review sentiment
0      One of the other reviewers has mentioned that ...  positive
1      A wonderful little production. <br /><br />The...  positive
2      I thought this was a wonderful way to spend ti...  positive
3      Basically there's a family where a little boy ...  negative
4      Petter Mattei's "Love in the Time of Money" is...  positive
...                                                  ...       ...
49995  I thought this movie did a down right good job...  positive
49996  Bad plot, bad dialogue, bad acting, idiotic di...  negative
49997  I am a Catholic taught in parochial elementary...  negative
49998  I'm going to have to disagree with the previou...  negative
49999  No one expects the Star Trek movies to be high...  negative

[50000 rows x 2 columns]
```

In [4]:
```
nltk.download('stopwords')
from nltk.corpus import stopwords
english_stops = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\2mscdsa18\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
```

In [5]:
```
def load_dataset():
    df = pd.read_csv('IMDB Dataset.csv')
    x_data = df['review']
    y_data = df['sentiment']
    x_data = x_data.replace({'<.*?>': ''}, regex = True)
    x_data = x_data.replace({'[^A-Za-z]': ' '}, regex = True)
    x_data = x_data.apply(lambda review: [w for w in review.split() if w not
    x_data = x_data.apply(lambda review: [w.lower() for w in review])
    y_data = y_data.replace('positive', 1)
    y_data = y_data.replace('negative', 0)
    return x_data, y_data
x_data, y_data = load_dataset()
print('Reviews')
print(x_data, '\n')
print('Sentiment')
print(y_data)
```

```
Reviews
0        [one, reviewers, mentioned, watching, oz, epis...
1        [a, wonderful, little, production, the, filmin...
2        [i, thought, wonderful, way, spend, time, hot,...
3        [basically, family, little, boy, jake, thinks,...
4        [petter, mattei, love, time, money, visually, ...
                               ...
49995    [i, thought, movie, right, good, job, it, crea...
49996    [bad, plot, bad, dialogue, bad, acting, idioti...
49997    [i, catholic, taught, parochial, elementary, s...
49998    [i, going, disagree, previous, comment, side, ...
49999    [no, one, expects, star, trek, movies, high, a...
Name: review, Length: 50000, dtype: object

Sentiment
0        1
1        1
2        1
3        0
4        1
        ..
49995    1
49996    0
49997    0
49998    0
49999    0
Name: sentiment, Length: 50000, dtype: int64
```

In [6]:
```
x_train, x_test, y_train, y_test = train_test_split(x_data, y_data, test_si
print('Train Set')
print(x_train, '\n')
print(x_test, '\n')
print('Test Set')
print(y_train, '\n')
print(y_test)
```

```
Train Set
5047     [if, went, movie, see, huge, academy, award, p...
48828    [i, also, viewed, film, santa, barbara, film, ...
40308    [as, far, i, recall, balanchine, alterations, ...
781      [i, thought, movie, incredible, i, absolutely,...
6326     [pilot, mitch, macafee, jeff, morrow, sees, uf...
                              ...
35509    [i, first, saw, movie, years, ago, i, shocked,...
24551    [it, second, year, academy, already, voting, p...
20204    [anyone, new, incredibly, prolific, takashi, m...
18385    [the, thing, i, knew, film, prior, seeing, rob...
18923    [you, might, tempted, rent, film, peter, selle...
Name: review, Length: 40000, dtype: object

12332    [this, film, opened, poor, showings, first, we...
47046    [it, really, great, film, able, ignore, blake,...
18175    [cia, analyst, douglas, freeman, gyllenhaal, g...
11316    [this, last, film, trilogy, brilliant, turkish...
5286     [paul, bettany, great, role, tortured, father,...
                              ...
29625    [frankly, i, met, real, han, su, ying, seeing,...
1165     [although, low, budget, film, clearly, last, m...
10950    [i, shakespeare, lover, since, childhood, i, a...
21053    [my, wife, i, watched, marvelous, movie, eveni...
13770    [this, movie, leave, thinking, while, many, pe...
Name: review, Length: 10000, dtype: object

Test Set
5047     1
48828    1
40308    1
781      1
6326     0
        ..
35509    1
24551    0
20204    1
18385    1
18923    0
Name: sentiment, Length: 40000, dtype: int64

12332    1
47046    1
18175    1
11316    1
5286     1
        ..
29625    0
1165     1
10950    0
21053    1
13770    1
Name: sentiment, Length: 10000, dtype: int64
```

In [7]:
```
def get_max_length():
    review_length = []
    for review in x_train:
        review_length.append(len(review))
    return int(np.ceil(np.mean(review_length)))
```

In [8]:
```
token = Tokenizer(lower=False)    # no need lower, because already lowered t
token.fit_on_texts(x_train)
x_train = token.texts_to_sequences(x_train)
x_test = token.texts_to_sequences(x_test)
max_length = get_max_length()
x_train = pad_sequences(x_train, maxlen=max_length, padding='post', truncati
x_test = pad_sequences(x_test, maxlen=max_length, padding='post', truncating
total_words = len(token.word_index) + 1
print('Encoded X Train\n', x_train, '\n')
print('Encoded X Test\n', x_test, '\n')
print('Maximum review length: ', max_length)
```

```
Encoded X Train
 [[  54  320    3 ...    0    0    0]
 [   1   22 2082 ...    0    0    0]
 [ 109  135    1 ...    0    0    0]
 ...
 [ 154   81  872 ...    0    0    0]
 [   2   65    1 ...  728 4477 1245]
 [  98  140 5768 ...    0    0    0]]

Encoded X Test
 [[    8     4  2881 ...  8407  8206  2658]
 [    7    15    21 ...     0     0     0]
 [ 3845 19211  2057 ...   280   247   952]
 ...
 [    1  1591  1339 ...    74    52    22]
 [  218   223     1 ...     0     0     0]
 [    8     3   467 ...   173  3151   586]]

Maximum review length:  130
```

In [9]:
```
EMBED_DIM = 32
LSTM_OUT = 64
model = Sequential()
model.add(Embedding(total_words, EMBED_DIM, input_length = max_length))
model.add(LSTM(LSTM_OUT))
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
model.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = [
print(model.summary())
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 130, 32)           2953120

 lstm (LSTM)                 (None, 64)                24832

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 1)                 65

=================================================================
Total params: 2,982,177
Trainable params: 2,982,177
Non-trainable params: 0
_________________________________________________________________
None
```

In [10]:
```
model.fit(x_train, y_train, batch_size = 128, epochs = 10)
```

```
Epoch 1/10
313/313 [==============================] - 32s 93ms/step - loss: 0.4582 -
accuracy: 0.7573
Epoch 2/10
313/313 [==============================] - 28s 91ms/step - loss: 0.1978 -
accuracy: 0.9264
Epoch 3/10
313/313 [==============================] - 29s 92ms/step - loss: 0.1151 -
accuracy: 0.9619
Epoch 4/10
313/313 [==============================] - 29s 92ms/step - loss: 0.0663 -
accuracy: 0.9784
Epoch 5/10
313/313 [==============================] - 28s 89ms/step - loss: 0.0501 -
accuracy: 0.9847
Epoch 6/10
313/313 [==============================] - 28s 91ms/step - loss: 0.0418 -
accuracy: 0.9870
Epoch 7/10
313/313 [==============================] - 29s 92ms/step - loss: 0.0237 -
accuracy: 0.9936
Epoch 8/10
313/313 [==============================] - 29s 93ms/step - loss: 0.0376 -
accuracy: 0.9892
Epoch 9/10
313/313 [==============================] - 29s 93ms/step - loss: 0.0262 -
accuracy: 0.9923
Epoch 10/10
313/313 [==============================] - 29s 93ms/step - loss: 0.0250 -
accuracy: 0.9925
```

Out[10]:
```
<keras.callbacks.History at 0x20d47af3e20>
```

In [11]:
```
model.evaluate(x_test,y_test)
```

```
313/313 [==============================] - 5s 13ms/step - loss: 0.5745 - a
ccuracy: 0.8689
```

Out[11]:
```
[0.5745013952255249, 0.8689000010490417]
```

In [12]:
```
EMBED_DIM = 32
model1 = Sequential()
model1.add(Embedding(total_words, EMBED_DIM, input_length = max_length))
model1.add(LSTM(32))
model1.add(Dense(64, activation='relu'))
model1.add(Dense(64, activation='relu'))
model1.add(Dense(1, activation='sigmoid'))
model1.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = |
print(model1.summary())
```

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 130, 32)           2953120

 lstm_1 (LSTM)               (None, 32)                8320

 dense_2 (Dense)             (None, 64)                2112

 dense_3 (Dense)             (None, 64)                4160

 dense_4 (Dense)             (None, 1)                 65

=================================================================
Total params: 2,967,777
Trainable params: 2,967,777
Non-trainable params: 0
_________________________________________________________________
None
```

In [13]:
```
model1.fit(x_train, y_train, batch_size = 128, epochs = 5)
```

```
Epoch 1/5
313/313 [==============================] - 26s 72ms/step - loss: 0.4676 -
accuracy: 0.7452
Epoch 2/5
313/313 [==============================] - 24s 75ms/step - loss: 0.1990 -
accuracy: 0.9258
Epoch 3/5
313/313 [==============================] - 24s 75ms/step - loss: 0.1065 -
accuracy: 0.9643
Epoch 4/5
313/313 [==============================] - 23s 73ms/step - loss: 0.0633 -
accuracy: 0.9795
Epoch 5/5
313/313 [==============================] - 23s 73ms/step - loss: 0.0437 -
accuracy: 0.9864
```

Out[13]:
```
<keras.callbacks.History at 0x20d37d4b370>
```

In [14]:
```
EMBED_DIM = 32
LSTM_OUT = 64
model2 = Sequential()
model2.add(Embedding(total_words, EMBED_DIM, input_length = max_length))
model2.add(Bidirectional(LSTM(LSTM_OUT)))
model2.add(Dense(64, activation='relu'))
model2.add(Dense(1, activation='sigmoid'))
model2.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = |
print(model2.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 130, 32)           2953120

 bidirectional (Bidirectiona  (None, 128)              49664
 l)

 dense_5 (Dense)             (None, 64)                8256

 dense_6 (Dense)             (None, 1)                 65

=================================================================
Total params: 3,011,105
Trainable params: 3,011,105
Non-trainable params: 0
_________________________________________________________________
None
```

In [15]:
```
model2.fit(x_train, y_train, batch_size = 128)
```

```
313/313 [==============================] - 45s 128ms/step - loss: 0.3832 -
accuracy: 0.8190
```

Out[15]:
```
<keras.callbacks.History at 0x20d3cbdf400>
```

In [ ]:
```

```