

ICP 6

Name: Tarun Kumar Reddy Kallu

Student Id: 700758234

GitHub Link: https://github.com/tarunkumarreddykallu/Neural_Assgn_6/tree/main

Video Link:

https://github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/video2834823665.mp4

Use Case Description:

1. Sentiment Analysis on the Twitter dataset

Programming elements:

1. Basics of LSTM

2. Types of RNN

3. Use case: Sentiment Analysis on the Twitter data set

In class programming:

1. Save the model and use the saved model to predict on new text data (ex, “A lot of good things are happening.

We are respected again throughout the world, and that's a great thing.@realDonaldTrump”)

2. Apply GridSearchCV on the source code provided in the class

Content x Bb 19892814 x Neural_Assgn_6/ICP_6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/ICP_6.ipynb

Select a Term

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

- main
- ICP_6.ipynb
- lcp_6.ipynb
- README.md

Neural_Assgn_6 / ICP_6.ipynb

tarunkumarreddykallu Add files via upload d800565 · 4 minutes ago History

Preview Code Blame 189 lines (189 loc) · 4.79 KB Code 55% faster with GitHub Copilot Raw Copy Edit

```
In [14]: import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re

from sklearn.preprocessing import LabelEncoder
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV

In [15]: data = pd.read_csv('data/Sentiment.csv')
data = data[['text', 'sentiment']]

data['text'] = data['text'].apply(lambda x: x.lower())
data['text'] = data['text'].apply(lambda x: re.sub('[^a-zA-Z0-9\s]', '', x))

In [16]: for idx, row in data.iterrows():
    row[0] = row[0].replace('\n', ' ')

In [17]: max_fatures = 2000
```

Russell 1000 +0.62%

12:13 PM 7/22/2024

Content x Bb 19892814 x Neural_Assgn_6/ICP_6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/ICP_6.ipynb

Select a Term

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Files

- main
- ICP_6.ipynb
- lcp_6.ipynb
- README.md

Neural_Assgn_6 / ICP_6.ipynb

tarunkumarreddykallu Add files via upload d800565 · 4 minutes ago History

Preview Code Blame 189 lines (189 loc) · 4.79 KB Code 55% faster with GitHub Copilot Raw Copy Edit

```
In [17]: max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(data['text'].values)
X = tokenizer.texts_to_sequences(data['text'].values)
X = pad_sequences(X)

In [18]: embed_dim = 128
lstm_out = 196
def createmodel():
    model = Sequential()
    model.add(Embedding(max_fatures, embed_dim, input_length = X.shape[1]))
    model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(3, activations='softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
    return model

In [19]: labelencoder = LabelEncoder()
integer_encoded = labelencoder.fit_transform(data['sentiment'])
y = to_categorical(integer_encoded)

In [20]: X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)

In [21]: batch_size = 32
model = createmodel()
model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
score, acc = model.evaluate(X_test, Y_test, verbose=2, batch_size=batch_size)
print(score)
print(acc)
print(model.metrics_names)
```

Russell 1000 +0.65%

12:13 PM 7/22/2024

Content x Bb 19892814 x Neural_Assgn_6/ICP_6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/ICP_6.ipynb

Select a Term

Files

- main
- ICP_6.ipynb
- lcp_6.ipynb
- README.md

Neural_Assgn_6 / ICP_6.ipynb

Preview Code Blame 189 lines (189 loc) · 4.79 KB Code 55% faster with GitHub Copilot

```
batch_size = 32
model = createModel()
model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
score, acc = model.evaluate(X_test, Y_test, verbose=2, batch_size=batch_size)
print(score)
print(acc)
print(model.metrics_names)

291/291 - 31s - loss: 0.8313 - accuracy: 0.6416
144/144 - 4s - loss: 0.7781 - accuracy: 0.6566
0.7780812978744507
0.656618595123291
['loss', 'accuracy']

In [22]:
print(X_train.shape, Y_train.shape)
print(X_test.shape, Y_test.shape)
model = KerasClassifier(build_fn=createModel, verbose=0)
epochs = [1, 2]
param_grid = dict(epochs=epochs)
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=1)
grid_result = grid.fit(X_train, Y_train, batch_size=32)
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))

(9293, 28) (9293, 3)
(4578, 28) (4578, 3)
Best: 0.674915 using {'epochs': 2}

In [ ]:

In [ ]:
```

Russell 1000 +0.62%

Search

12:13 PM 7/22/2024

Content x Bb 19892814 x Neural_Assgn_6/lcp_6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/lcp%206.ipynb

Select a Term

Files

- main
- ICP_6.ipynb
- lcp_6.ipynb
- README.md

Neural_Assgn_6 / lcp_6.ipynb

tarunkumarreddykallu Add files via upload d800565 · 4 minutes ago History

Preview Code Blame 245 lines (245 loc) · 7.19 KB Code 55% faster with GitHub Copilot

```
In [ ]:

In [27]:
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
import re

from sklearn.preprocessing import LabelEncoder
from keras.models import Sequential, load_model
import numpy

In [28]:
data = pd.read_csv('data/Sentiment.csv')
data = data[['text', 'sentiment']]

data['text'] = data['text'].apply(lambda x: x.lower())
data['text'] = data['text'].apply(lambda x: re.sub('[^a-zA-z0-9\s]', '', x))
for idx, row in data.iterrows():
    row[0] = row[0].replace('rt', ' ')

In [29]:
max_features = 2000
tokenizer = Tokenizer(num_words=max_features, splits=' ')
```

Russell 1000 +0.65%

Search

12:13 PM 7/22/2024

Content x Bb 19892814 x Neural_Assgn_6/lcp 6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/lcp%206.ipynb

Select a Term

Files

- main
- Go to file
- ICP_6.ipynb
- lcp 6.ipynb
- README.md

Neural_Assgn_6 / lcp 6.ipynb

tarunkumarreddykallu Add files via upload d800565 · 4 minutes ago History

Preview Code Blame 245 lines (245 loc) · 7.19 KB Code 55% faster with GitHub Copilot Raw Copy Edit

```
In [29]: max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, splits=' ')
tokenizer.fit_on_texts(data['text'].values)
X = tokenizer.texts_to_sequences(data['text'].values)
X = pad_sequences(X)

In [30]: labelencoder = LabelEncoder()
integer_encoded = labelencoder.fit_transform(data['sentiment'])
y = to_categorical(integer_encoded)

In [31]: X_train, X_test, Y_train, Y_test = train_test_split(X, y, test_size = 0.33, random_state = 42)

In [32]: embed_dim = 128
lstm_out = 196
def createmodel():
    model = Sequential()
    model.add(Embedding(max_fatures, embed_dim, input_length = X.shape[1]))
    model.add(LSTM(lstm_out, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(3, activation='softmax'))
    model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics = ['accuracy'])
    return model

In [33]: batch_size = 32
model = createmodel()
model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
```

81°F Haze 12:13 PM 7/22/2024

Content x Bb 19892814 x Neural_Assgn_6/lcp 6.ipynb at x My Drive - Google Drive x +

github.com/tarunkumarreddykallu/Neural_Assgn_6/blob/main/lcp%206.ipynb

Select a Term

Files

- main
- Go to file
- ICP_6.ipynb
- lcp 6.ipynb
- README.md

Neural_Assgn_6 / lcp 6.ipynb

tarunkumarreddykallu Add files via upload d800565 · 4 minutes ago History

Preview Code Blame 245 lines (245 loc) · 7.19 KB Code 55% faster with GitHub Copilot Raw Copy Edit

```
In [33]: batch_size = 32
model = createmodel()
model.fit(X_train, Y_train, epochs = 1, batch_size=batch_size, verbose = 2)
score, acc = model.evaluate(X_test, Y_test, verbose=2, batch_size=batch_size)
print(score)
print(acc)
print(model.metrics_names)

291/291 - 32s - loss: 0.8264 - accuracy: 0.6423
144/144 - 2s - loss: 0.7607 - accuracy: 0.6693
0.7606955766677856
0.669287919998169
['loss', 'accuracy']

In [34]: model.save('model.h5')
mod = load_model('model.h5')
print(mod.summary())

Model: "sequential_4"
Layer (type) Output Shape Param #
-----
embedding_4 (Embedding) (None, 28, 128) 256000
lstm_4 (LSTM) (None, 196) 254800
dense_4 (Dense) (None, 3) 591
-----
Total params: 511,391
Trainable params: 511,391
Non-trainable params: 0
None
```

81°F Haze 12:14 PM 7/22/2024

Select a term

Files

main

Go to file

ICP_6.ipynb

lcp 6.ipynb

README.md

Neural_Assgn_6 / lcp 6.ipynb

PreviewCodeBlame245 lines (245 loc) · 7.19 KBCode 55% faster with GitHub CopilotRawDownloadEdit

In [36]:

```
tst = [['A lot of good things are happening. We are respected again throughout the world, and thats a great '
        'thing.@realDonaldTrump']]
max_data = pd.DataFrame(tst, index=range(0, 1, 1), columns=list('t'))
max_data['t'] = max_df['t'].apply(lambda x: x.lower())
max_data['t'] = max_df['t'].apply((lambda x: re.sub('[^a-zA-z0-9\s]', '', x)))
features = 2000
tokenizer = Tokenizer(num_words=features, split=' ')
tokenizer.fit_on_texts(max_data['t'].values)
X = tokenizer.texts_to_sequences(max_data['t'].values)
X = pad_sequences(X, maxlen=28)

out = mod.predict(X)
print(out)
print(numpy.where(max(out[0]), ".", (max(out[0]))))
print(numpy.argmax(out))
print(mod.summary())

[[0.80123734 0.07214491 0.12661776]]
(array([0], dtype=int64),) : 0.80123734
0
Model: "sequential_4"
```

Layer (type)	Output Shape	Param #
embedding_4 (Embedding)	(None, 28, 128)	256000
lstm_4 (LSTM)	(None, 196)	254800
dense_4 (Dense)	(None, 3)	591

Total params: 511,391
Trainable params: 511,391
Non-trainable params: 0

None