

NEURAL NETWORK & DEEP LEARNING

ASSIGNMENT 9

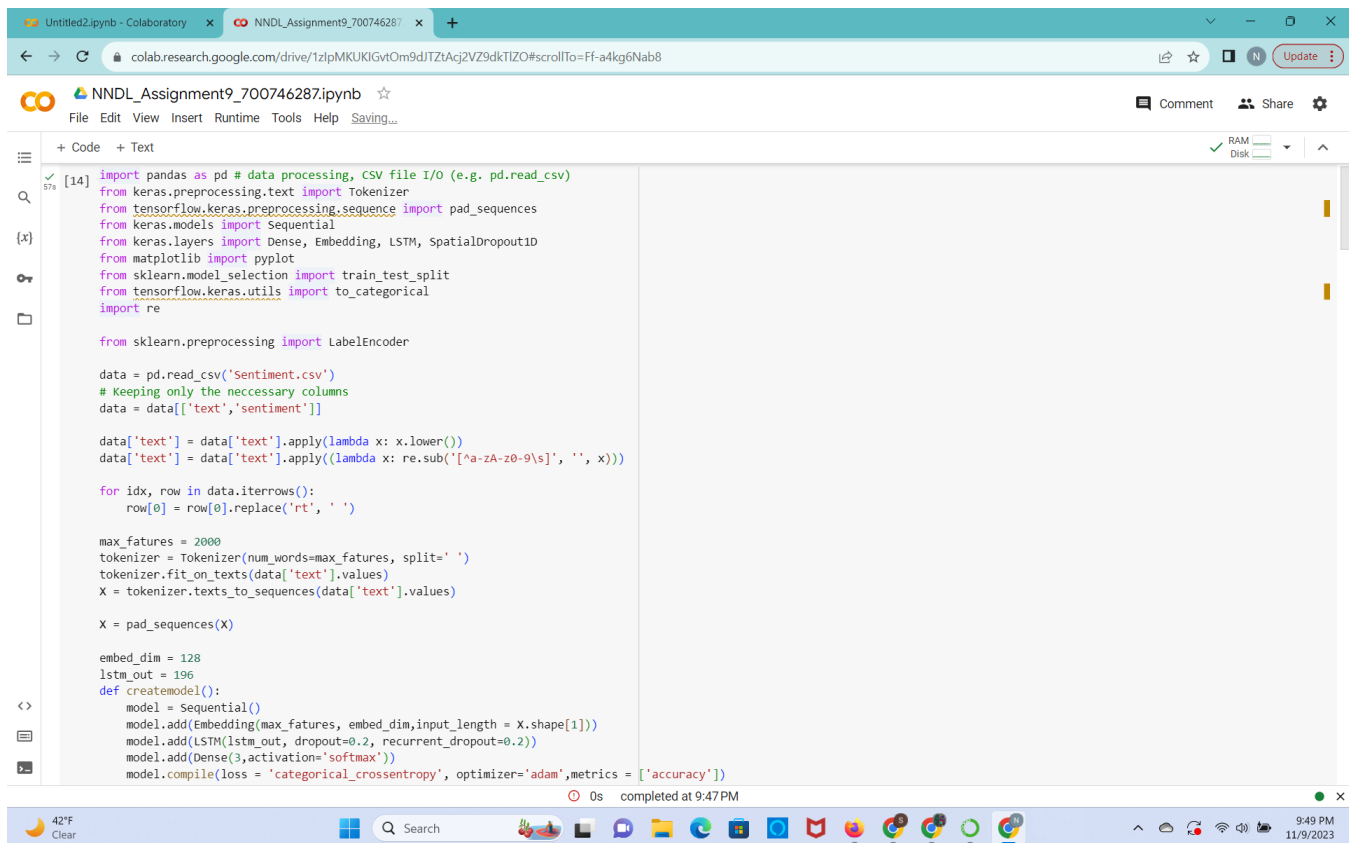
Name : SAI SNUSHA NAKKA

Student ID : 700746287

Git hub Link: https://github.com/NSnusha/NNDL_Assignment9

Video link:

https://drive.google.com/file/d/1_4I5uuX3dhenTavemL9Zmnuwx5cVgEjh/view?usp=s_haring



The screenshot displays a Google Colaboratory notebook titled "NNDL_Assignment9_700746287.ipynb". The code is written in Python and implements a sentiment classification model using Keras and LSTM. The code includes imports for pandas, Keras preprocessing and models, TensorFlow preprocessing, sklearn preprocessing, and matplotlib. It reads a CSV file named "Sentiment.csv", preprocesses the text data by lowercasing and removing non-alphanumeric characters, and tokenizes the sequences. The model is built using a Sequential architecture with an Embedding layer, an LSTM layer, and a Dense output layer with softmax activation. The model is compiled with categorical crossentropy loss, Adam optimizer, and accuracy metric. The notebook interface shows the code editor on the left and a runtime panel on the right, indicating the code is completed at 9:47 PM.

```
[14] import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from keras.preprocessing.text import Tokenizer
from tensorflow.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 tensorflow.keras.utils import to_categorical
import re

from sklearn.preprocessing import LabelEncoder

data = pd.read_csv('Sentiment.csv')
# Keeping only the necessary columns
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', ' ')

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)

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'])
```

Colaboratory

colab.research.google.com/drive/1zlpMKUKIGvOm9dJTzAcj2VZ9dkTIZO#scrollTo=FF-a4kg6Nab8

RAM
Disk

Comment Share

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[14]
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
# print(model.summary())

labelencoder = LabelEncoder()
integer_encoded = labelencoder.fit_transform(data['sentiment'])
y = to_categorical(integer_encoded)
X_train, X_test, Y_train, Y_test = train_test_split(X,y, test_size = 0.33, random_state = 42)

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 - 51s - loss: 0.8309 - accuracy: 0.6439 - 51s/epoch - 176ms/step
144/144 - 3s - loss: 0.7467 - accuracy: 0.6739 - 3s/epoch - 22ms/step
0.7466925382614136
0.6738750338554382
['loss', 'accuracy']

[15] model.save('sentiment_model.h5')

/usr/local/lib/python3.10/dist-packages/keras/src/engine/training.py:3079: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered leg
saving_api.save_model()
```

0s completed at 9:47 PM

Colaboratory

colab.research.google.com/drive/1zlpMKUKIGvOm9dJTzAcj2VZ9dkTIZO#scrollTo=UgaoNPS3ikgb

RAM
Disk

Comment Share

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
from scikeras.wrappers import KerasClassifier
#from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from keras.optimizers import Adam

def create_model(units=196, dropout=0.2, learning_rate=0.001):
    model = Sequential()
    model.add(Embedding(max_fatures, embed_dim,input_length = X.shape[1]))
    model.add(LSTM(units, dropout=dropout, recurrent_dropout=dropout))
    model.add(Dense(3, activation='softmax'))
    optimizer = Adam(lr=learning_rate)
    model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
    return model

model = KerasClassifier(build_fn=create_model,verbose=2) #initiating model to test performance by applying multiple hyper parameters
batch_size= [10, 20, 40] #hyper parameter batch size
epochs = [1, 2] #hyper parameter no. of epochs
param_grid= {'batch_size':batch_size, 'epochs':epochs} #creating dictionary for batch size, no. of epochs
grid = GridSearchCV(estimator=model, param_grid=param_grid) #Applying dictionary with hyper parameters
grid_result= grid.fit(X_train,Y_train) #Fitting the model
# summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_)) #Best score, best hyper parameters

/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will ra
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8164 - accuracy: 0.6512 - 55s/epoch - 73ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will ra
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8204 - accuracy: 0.6480 - 55s/epoch - 74ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will ra
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8263 - accuracy: 0.6442 - 55s/epoch - 74ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will ra
X, y = self._initialize(X, y)
```

Executing (12m 49s) > cell ll... > fi... > run... > evaluate_ca... > _c... > _c... > _get_sequenc... > _c... > _fit_and... > fi... > _... > _fit_keras... > error_h... > fi... > error_h... > _c... > _... > call_fu... > _call... > _c... > call_fu... > quick_ex... > _c...

1:17 PM 11/10/2023

Untitled2.ipynb - Colaboratory

NNDL_Assignment9_700746287

python - Cannot import name

colab.research.google.com/drive/1zlpMKUKIGvtOm9dJTztAgj2VZ9dkTIZO#scrollTo=UgaoNPS3iKgb

NNDL_Assignment9_700746287.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

```
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8164 - accuracy: 0.6512 - 55s/epoch - 73ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8204 - accuracy: 0.6480 - 55s/epoch - 74ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 55s - loss: 0.8263 - accuracy: 0.6442 - 55s/epoch - 74ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 62s - loss: 0.8241 - accuracy: 0.6461 - 62s/epoch - 83ms/step
186/186 - 1s - 1s/epoch - 8ms/step
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 57s - loss: 0.8163 - accuracy: 0.6456 - 57s/epoch - 77ms/step
186/186 - 1s - 1s/epoch - 8ms/step
Epoch 1/2
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 53s - loss: 0.8256 - accuracy: 0.6431 - 53s/epoch - 72ms/step
Epoch 2/2
744/744 - 52s - loss: 0.6808 - accuracy: 0.7144 - 52s/epoch - 70ms/step
186/186 - 1s - 1s/epoch - 7ms/step
Epoch 1/2
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 53s - loss: 0.8271 - accuracy: 0.6462 - 53s/epoch - 72ms/step
Epoch 2/2
744/744 - 50s - loss: 0.6841 - accuracy: 0.7111 - 50s/epoch - 68ms/step
186/186 - 1s - 1s/epoch - 8ms/step
Epoch 1/2
/usr/local/lib/python3.10/dist-packages/scikeras/wrappers.py:915: UserWarning: ``build_fn`` will be renamed to ``model`` in a future release, at which point use of ``build_fn`` will raise a DeprecationWarning.
X, y = self._initialize(X, y)
744/744 - 57s - loss: 0.8274 - accuracy: 0.6478 - 57s/epoch - 76ms/step
Epoch 2/2
```

Executing (12m 56s) <cell li...> fi...> _run_s...> evaluate_ca...> _c...> _c...> _get_sequenti...> _c...> _fit_and...> fi...> _c...> _fit_keras...> error_h...> error_h...> _c...> _c...> call_fu...> _call...> flat...> _c...> call_fu...> quick_ex...> X

42°F Clear

1:17 PM
11/10/2023