Assignment-5 NNDL

Name: Divya Chilukuri Student ID: 700744169

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"

```
import numpy as np
# Function to create the LSTM model
def create_model():
    model = Sequential()
    model.add(Embedding(max_features, 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


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_features = 2000
tokenizer = Tokenizer(num_words=max_features, 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

label_encoder = LabelEncoder()
integer_encoded = label_encoder.fit_transform(data['sentiment'])
y = to_categorical(integer_encoded)
```

GITHUB LINK: https://github.com/DivyaReddy1999/NNDL_Assignment5.git

Assignment-5 NNDL

Name: Divya Chilukuri Student ID: 700744169

```
# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)

# Training the model
batch_size = 32
model = create_model()
model.fit(X_train, y_train, epochs=5, batch_size=batch_size, verbose=2)

# Evaluating the model on the test data
score, acc = model.evaluate(X_test, y_test, verbose=2, batch_size=batch_size)
print("Test loss:", score)
print("Test accuracy:", acc)

# Predicting on new text data
new_text = "A lot of good things are happening. We are respected again throughout the world, and that's a great thing. @realDonaldTrump"
new_text = new_text.lower()
new_text = re.sub('[^a-zA-Z0-9\s]', '', new_text)
new_sequence = tokenizer.texts_to_sequences([new_text])
new_padded_sequence = pad_sequences(new_sequence, maxlen=X.shape[1])

# Make predictions using the trained model
predicted_sentiment = model.predict(new_padded_sequence)
sentiment_labels = ['Negative', 'Neutral', 'Positive']
predicted_label = sentiment_labels[np.argmax(predicted_sentiment)]

print("Predicted Sentiment Label:", predicted_label)
```

```
Epoch 1/5
291/291 - 56s - loss: 0.8277 - accuracy: 0.6480 - 56s/epoch - 192ms/step
Epoch 2/5
291/291 - 49s - loss: 0.6772 - accuracy: 0.7086 - 49s/epoch - 167ms/step
Epoch 3/5
291/291 - 51s - loss: 0.6117 - accuracy: 0.7429 - 51s/epoch - 176ms/step
Epoch 4/5
291/291 - 49s - loss: 0.5639 - accuracy: 0.7662 - 49s/epoch - 168ms/step
Epoch 5/5
291/291 - 55s - loss: 0.5256 - accuracy: 0.7846 - 55s/epoch - 190ms/step
144/144 - 5s - loss: 0.8409 - accuracy: 0.6730 - 5s/epoch - 32ms/step
Test loss: 0.8409407734870911
Test accuracy: 0.6730012893676758
1/1 [==============================] - 0s 335ms/step
Predicted Sentiment Label: Positive
```

GITHUB LINK: https://github.com/DivyaReddy1999/NNDL_Assignment5.git

Assignment-5 NNDL

Name: Divya Chilukuri Student ID: 700744169

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

```
import pandas as pd
import re
import numpy as np
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score
from keras.wrappers.scikit_learn import KerasClassifier
from keras.models import Sequential
from keras.layers import Embedding, LSTM, Dense
from keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from keras.utils import to_categorical

def preprocess_text(text):
    text = text.lower()
    text = re.sub('[^a-zA-z0-9\s]', '', text)
    text = text.replace('rt', ' ')
    return text

#1.Collecting the data
from google.colab import drive
drive.mount('/content/drive')
data = pd.read_csv('/content/drive/My Drive/Colab_Notebooks/Sentiment.csv')
data = data[['text', 'sentiment']]

data['text'] = data['text'].apply(preprocess_text)
```

GITHUB LINK: https://github.com/DivyaReddy1999/NNDL_Assignment5.git

Assignment-5 NNDL

Name: Divya Chilukuri Student ID: 700744169

```
max_features = 2000
tokenizer = Tokenizer(num_words=max_features, 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 create_model():
    model = Sequential()
    model.add(Embedding(max_features, 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

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

# Wrap the Keras model inside a function for GridSearchCV
model = KerasClassifier(build_fn=create_model, verbose=0)
```

GITHUB LINK: https://github.com/DivyaReddy1999/NNDL_Assignment5.git

Assignment-5 NNDL

Name: Divya Chilukuri Student ID: 700744169

```
# Define the hyperparameter grid
param_grid = {
    'batch_size': [32, 64],
    'epochs': [1, 2],
}

# Create the GridSearchCV instance
grid_search = GridSearchCV(estimator=model, param_grid=param_grid, cv=2)
grid_result = grid_search.fit(X_train, Y_train)

# Get the best parameters and model
best_params = grid_result.best_params_
best_model = grid_result.best_estimator_

# Evaluate the best model on the test set
y_pred = best_model.predict(X_test)
accuracy = accuracy_score(Y_test.argmax(axis=1), y_pred)
print("Best Parameters:", best_params)
print("Test Accuracy:", accuracy)
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount,
<ipython-input-15-22e3be351a77>:54: DeprecationWarning: KerasClassifier i
  model = KerasClassifier(build_fn=create_model, verbose=0)
144/144 [==============================] - 5s 30ms/step
Best Parameters: {'batch_size': 64, 'epochs': 2}
Test Accuracy: 0.6791175185670598
```

GITHUB LINK: https://github.com/DivyaReddy1999/NNDL_Assignment5.git