ICP 7

ICP7

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

[4]
```python
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[12]
```python
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
from tensorflow.keras.preprocessing.text import Tokenizer # Changed import statement to use tensorflow.keras
from tensorflow.keras.preprocessing.sequence import pad_sequences # Changed import statement to use tensorflow.keras
from tensorflow.keras.models import Sequential # Changed import statement to use tensorflow.keras
from tensorflow.keras.layers import Dense, Embedding, LSTM, SpatialDropout1D # Changed import statement to use tensorflow.keras
from matplotlib import pyplot
from sklearn.model_selection import train_test_split
from tensorflow.keras.utils import to_categorical # Changed import statement to use tensorflow.keras
import re

from sklearn.preprocessing import LabelEncoder

file_path = '/content/drive/My Drive/Sentiment.csv'
# Update the path to reflect the actual location of your file in Google Drive

data = pd.read_csv(file_path) # Changed to use the updated file path
# Keeping only the neccessary 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)))
```

ICP7

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```python
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'])
    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)
```

Connected to Python 3 Google Compute Engine backend

ICP7

File Edit View Insert Runtime Tools Help All changes saved

```python
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)
```

```
<ipython-input-12-8f88e2e1f757>:24: FutureWarning: Series.__getitem__ treating keys as positions is deprecated. In a future version, integer keys will
  row[0] = row[0].replace('rt', ' ')
<ipython-input-12-8f88e2e1f757>:24: FutureWarning: Series.__setitem__ treating keys as positions is deprecated. In a future version, integer keys will
  row[0] = row[0].replace('rt', ' ')
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/embedding.py:90: UserWarning: Argument `input_length` is deprecated. Just remove it.
  warnings.warn(
291/291 - 33s - 113ms/step - accuracy: 0.6388 - loss: 0.8309
144/144 - 4s - 29ms/step - accuracy: 0.6741 - loss: 0.7583
0.7583316564559937
0.67409348487854
['loss', 'compile_metrics']
```

```python
[13] model.save('sentimentAnalysis.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy.
```

ICP7

File Edit View Insert Runtime Tools Help All changes saved

```python
[13] model.save('sentimentAnalysis.h5')
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy.
```

```python
from keras.models import load_model
model = load_model('sentimentAnalysis.h5')
new_text = ["A lot of good things are happening. We are respected again throughout the world, and that's a great thing .@realDonaldTrump"]

# 1. Tokenize the new text using the same tokenizer used during training
new_text_sequences = tokenizer.texts_to_sequences(new_text)

# 2. Pad the sequences to match the input shape expected by the model
new_text_padded = pad_sequences(new_text_sequences, maxlen=X.shape[1]) # Use X.shape[1] from training data

# 3. Now you can predict using the processed input
predictions = model.predict(new_text_padded)

print(predictions)
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.compile_metrics` will be empty until you train or evalua
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 248ms/step
[[0.48573926 0.16640949 0.34785128]]
```

ICP7

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
!pip install tensorflow==2.12.0
!pip install keras==2.12.0
!pip install keras.utils
```

```
Requirement already satisfied: tensorflow==2.12.0 in /usr/local/lib/python3.10/dist-packages (2.12.0)
Requirement already satisfied: absl-py>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.4.0)
Requirement already satisfied: astunparse>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.6.3)
Requirement already satisfied: flatbuffers>=2.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (24.3.25)
Requirement already satisfied: gast<=0.4.0,>=0.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.4.0)
Requirement already satisfied: google-pasta>=0.1.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.2.0)
Requirement already satisfied: grpcio<2.0,>=1.24.3 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.64.1)
Requirement already satisfied: h5py>=2.9.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (3.11.0)
Requirement already satisfied: jax>=0.3.15 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.4.30)
Requirement already satisfied: keras<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.0)
Requirement already satisfied: libclang>=13.0.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (18.1.1)
Requirement already satisfied: numpy<1.24,>=1.22 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.23.5)
Requirement already satisfied: opt-einsum>=2.3.2 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (3.4.0)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (24.1)
Requirement already satisfied: protobuf!=4.21.0,!=4.21.1,!=4.21.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (75.1.0)
Requirement already satisfied: six>=1.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.16.0)
Requirement already satisfied: tensorboard<2.13,>=2.12 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.3)
Requirement already satisfied: tensorflow-estimator<2.13,>=2.12.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.12.0)
Requirement already satisfied: termcolor>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (2.5.0)
Requirement already satisfied: typing-extensions>=3.6.6 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (4.12.2)
Requirement already satisfied: wrapt<1.15,>=1.11.0 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (1.14.1)
Requirement already satisfied: tensorflow-io-gcs-filesystem>=0.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow==2.12.0) (0.37.1)
Requirement already satisfied: wheel<1.0,>=0.23.0 in /usr/local/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow==2.12.0) (0.44.0)
```

✓ 1h 11m 0s completed at 12:11 AM

ICP7

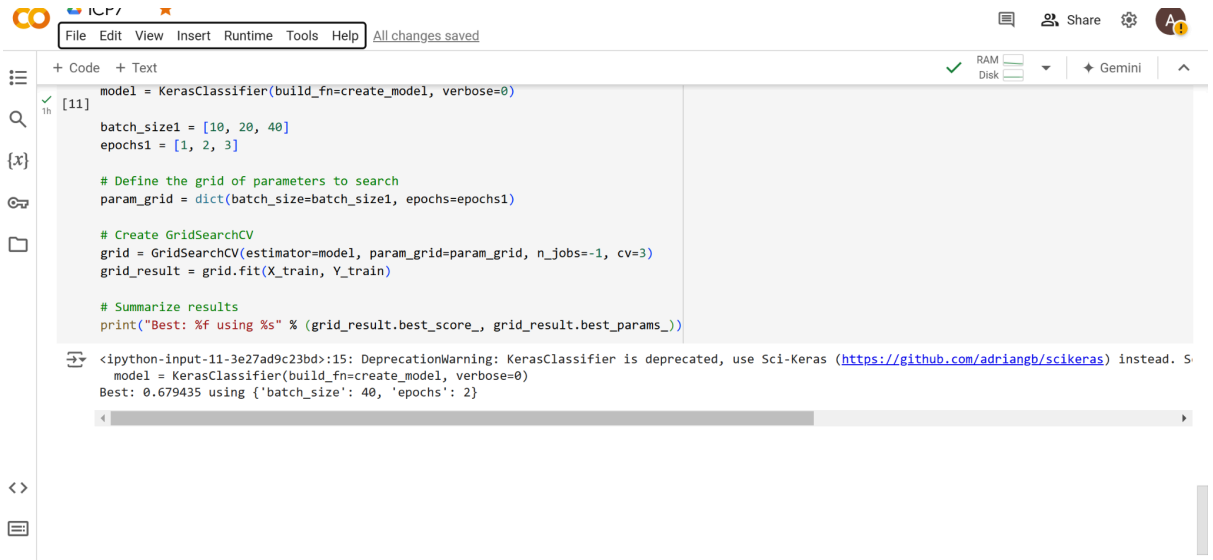
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[11] from keras.wrappers.scikit_learn import KerasClassifier
     from sklearn.model_selection import GridSearchCV
     from keras.layers import LSTM

     # Function to create the model, as it's required by KerasClassifier
     def create_model(lstm_out=196, dropout=0.2):
         model = Sequential()
         model.add(Embedding(max_fatures, embed_dim, input_length=X.shape[1]))
         model.add(LSTM(lstm_out, dropout=dropout, recurrent_dropout=dropout))
         model.add(Dense(3, activation='softmax'))
         model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
         return model

     # Create the KerasClassifier
     model = KerasClassifier(build_fn=create_model, verbose=0)

     batch_size1 = [10, 20, 40]
     epochs1 = [1, 2, 3]

     # Define the grid of parameters to search
     param_grid = dict(batch_size=batch_size1, epochs=epochs1)

     # Create GridSearchCV
     grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
     grid_result = grid.fit(X_train, Y_train)

     # Summarize results
```

31 October 2024

Thu 00:17 (Local time)



```
[11]
model = KerasClassifier(build_fn=create_model, verbose=0)

batch_size1 = [10, 20, 40]
epochs1 = [1, 2, 3]

# Define the grid of parameters to search
param_grid = dict(batch_size=batch_size1, epochs=epochs1)

# Create GridSearchCV
grid = GridSearchCV(estimator=model, param_grid=param_grid, n_jobs=-1, cv=3)
grid_result = grid.fit(X_train, Y_train)

# Summarize results
print("Best: %f using %s" % (grid_result.best_score_, grid_result.best_params_))
```

```
<ipython-input-11-3e27ad9c23bd>:15: DeprecationWarning: KerasClassifier is deprecated, use Sci-Keras (https://github.com/adriangb/scikeras) instead. S
  model = KerasClassifier(build_fn=create_model, verbose=0)
Best: 0.679435 using {'batch_size': 40, 'epochs': 2}
```

MyGithub Link:https://github.com/w8162583/bda.git