

breakingnews

April 5, 2024

```
[ ]: import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Conv1D, MaxPooling1D, Flatten, Dense, Embedding
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences

[ ]: # Load dataset (replace 'data.csv' with your file path)
data = pd.read_csv('data.csv')

# Split data into features (X) and labels (y)
X = data['text']
y = data['label']

# Tokenize the text
max_words = 10000
tokenizer = Tokenizer(num_words=max_words)
tokenizer.fit_on_texts(X)
sequences = tokenizer.texts_to_sequences(X)

# Pad sequences to ensure uniform length
maxlen = 100
X_pad = pad_sequences(sequences, maxlen=maxlen)

# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X_pad, y, test_size=0.2,
↳ random_state=42)

[ ]: # Define CNN model
model = Sequential()
model.add(Embedding(max_words, 50, input_length=maxlen))
model.add(Conv1D(64, 5, activation='relu'))
model.add(MaxPooling1D(pool_size=4))
model.add(Flatten())
model.add(Dense(1, activation='sigmoid'))
```

```
# Compile the model
model.compile(optimizer='adam', loss='binary_crossentropy',  
↳metrics=['accuracy'])

# Print model summary
print(model.summary())
```

```
[ ]: # Train the model
model.fit(X_train, y_train, epochs=5, batch_size=32, validation_data=(X_test,  
↳y_test))
```

```
[ ]: # Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print("Test Accuracy:", accuracy)

# Make predictions
predictions = model.predict_classes(X_test)
```