import tensorflow as tf

from tensorflow.keras.models import Sequential #type:ignore

from tensorflow.keras.layers import Dense, Embedding, LSTM, Bidirectional, Dropout #type:ignore

from tensorflow.keras.preprocessing.sequence import pad\_sequences #type:ignore

from tensorflow.keras.datasets import imdb #type:ignore

# Load the IMDB dataset

vocab\_size = 100 # Use the top 10,000 words in the dataset

max\_length = 20 # Max length of each review

(x\_train, y\_train), (x\_test, y\_test) = imdb.load\_data(num\_words=vocab\_size)

# Pad sequences to ensure uniform input shape

x\_train = pad\_sequences(x\_train, maxlen=max\_length, padding='post', truncating='post')

x\_test = pad\_sequences(x\_test, maxlen=max\_length, padding='post', truncating='post')

# Build the deep neural network model

model = Sequential([

Embedding(vocab\_size, 128, input\_length=max\_length), # Embedding layer

Bidirectional(LSTM(64, return\_sequences=False)), # Bidirectional LSTM for sequential data

Dropout(0.3), # Dropout for regularization

Dense(64, activation='relu'), # Dense layer

Dropout(0.3),

Dense(1, activation='sigmoid') # Output layer for binary classification

])

# Compile the model

model.compile(optimizer='adam',

loss='binary\_crossentropy',

metrics=['accuracy'])

# Train the model

history = model.fit(

x\_train, y\_train,

epochs=5,

batch\_size=32,

validation\_split=0.2,

verbose=1

)

# Evaluate the model on test data

test\_loss, test\_accuracy = model.evaluate(x\_test, y\_test, verbose=0)

print(f"Test Accuracy: {test\_accuracy \* 100:.2f}%")

#OUTPUT

To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.

625/625 ━━━━━━━━━━━━━━━━━━━━ 11s 10ms/step - accuracy: 0.5413 - loss: 0.6859 - val\_accuracy: 0.5858 - val\_loss: 0.6653

Test Accuracy: 58.51%