Ex.No.6

RECURRENT NEURAL NETWORK

AIM:

To build a recurrent neural network with Keras/TensorFlow

PROCEDURE:

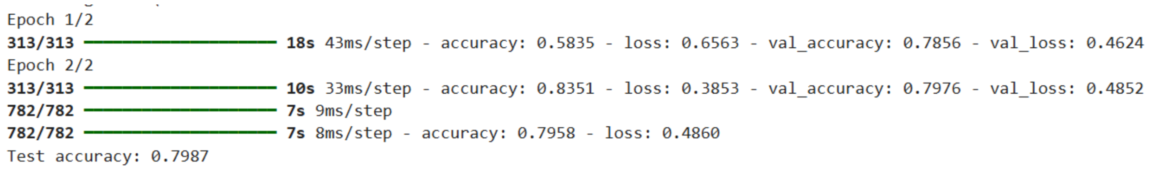
1. Download and load the dataset.
2. Perform analysis and preprocessing of the dataset.
3. Build a recurrent neural network model using Keras/TensorFlow.
4. Compile and fit the model.
5. Perform prediction with the test dataset.
6. Calculate performance metrics.

PROGRAM:

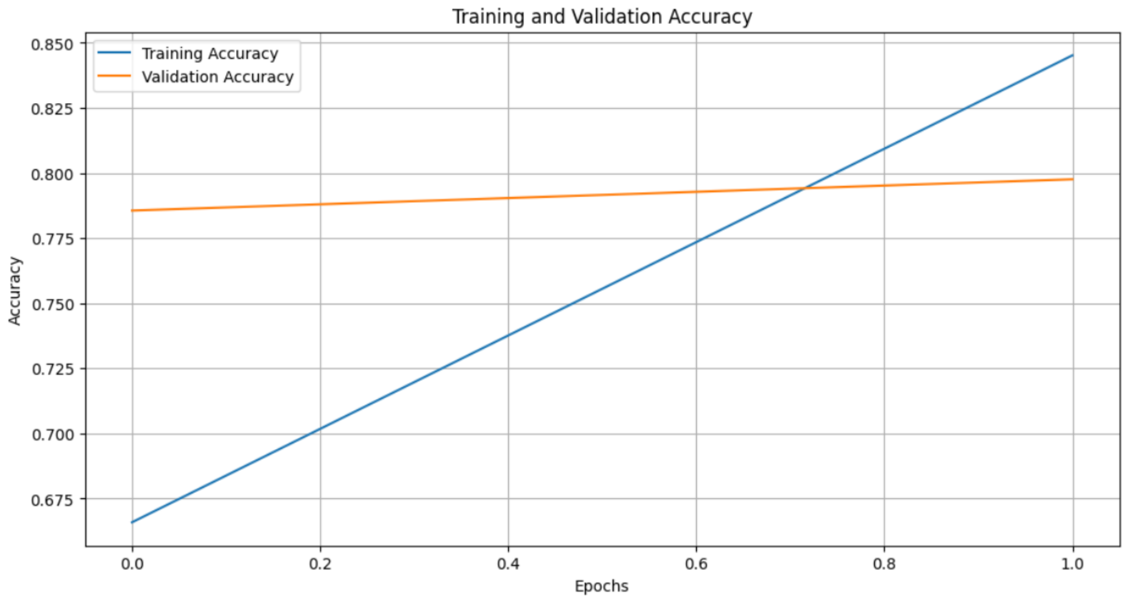
```
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, LSTM, Dense, Bidirectional, SimpleRNN
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.datasets import imdb
import matplotlib.pyplot as plt

vocab_size = 10000
max_len = 100
```

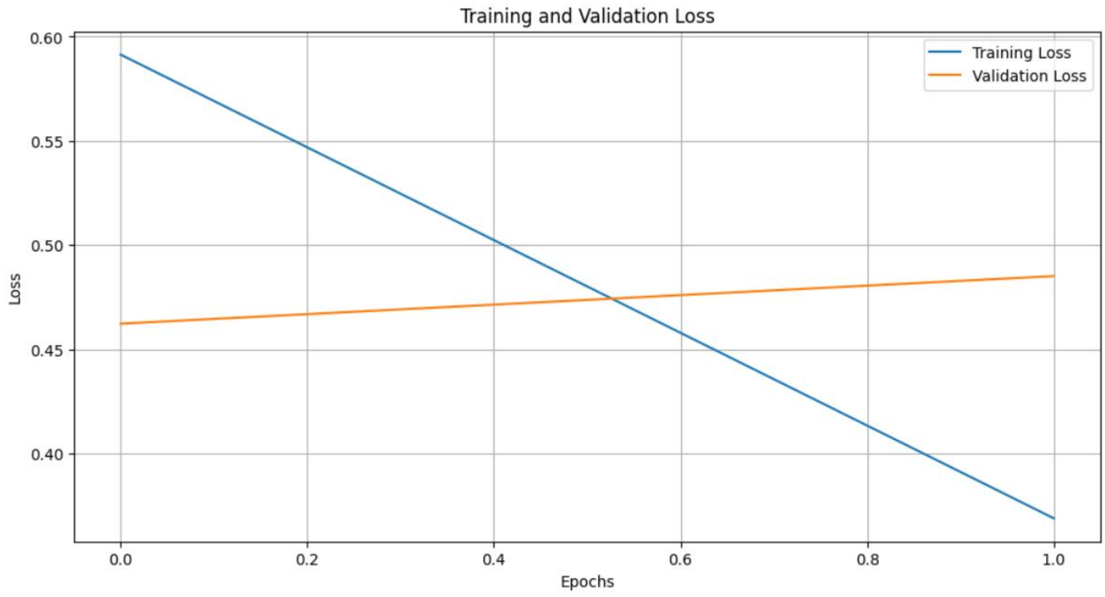
```
(x_train, y_train), (x_test, y_test) = imdb.load_data(num_words=vocab_size)

x_train = pad_sequences(x_train, maxlen=max_len, padding='post')
x_test = pad_sequences(x_test, maxlen=max_len, padding='post')

model = Sequential([
    Embedding(input_dim=vocab_size, output_dim=32, input_length=max_len),
    SimpleRNN(32, return_sequences=False),
    Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
loss='binary_crossentropy',
metrics=['accuracy'])

history = model.fit(x_train, y_train, epochs=2, batch_size=64,
validation_split=0.2)

predictions = model.predict(x_test)

test_loss, test_acc = model.evaluate(x_test, y_test)
print(f'Test accuracy: {test_acc:.4f}')

plt.figure(figsize=(12, 6))
plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.grid(True)
plt.show()

plt.figure(figsize=(12, 6))
plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT:



```
Epoch 1/2
313/313 ━━━━━━━━━━━━━━━━━━━━ 18s 43ms/step - accuracy: 0.5835 - loss: 0.6563 - val_accuracy: 0.7856 - val_loss: 0.4624
Epoch 2/2
313/313 ━━━━━━━━━━━━━━━━━━━━ 10s 33ms/step - accuracy: 0.8351 - loss: 0.3853 - val_accuracy: 0.7976 - val_loss: 0.4852
782/782 ━━━━━━━━━━━━━━━━━━━━ 7s 9ms/step
782/782 ━━━━━━━━━━━━━━━━━━━━ 7s 8ms/step - accuracy: 0.7958 - loss: 0.4860
Test accuracy: 0.7987
```





RESULT:

Thus, a recurrent neural network with Keras/TensorFlow was successfully implemented.