

**Surakshith Shetty-53026240013**

## 3b.Demonstrate Sentiment Analysis using recurrent neural network (RNN)

```

import numpy as np
import tensorflow as tf
from tensorflow.keras.preprocessing.text import Tokenizer
from tensorflow.keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Embedding, SimpleRNN, Dense

# Sample dataset
sentences = [
    'I love this product',
    'This is the worst thing I ever bought',
    'Absolutely fantastic!',
    'Not good at all',
    'I am very happy with the purchase',
    'This is not what I expected',
    'I would buy this again',
    'Terrible, do not buy this'
]

labels = [1, 0, 1, 0, 1, 0, 1, 0] # Sentiments: 1 = positive, 0 = negative

sentences
→ ['I love this product',
   'This is the worst thing I ever bought',
   'Absolutely fantastic!',
   'Not good at all',
   'I am very happy with the purchase',
   'This is not what I expected',
   'I would buy this again',
   'Terrible, do not buy this']

tok = Tokenizer(num_words=100)
tok.fit_on_texts(sentences)

sequences = tok.texts_to_sequences(sentences)
sequences
→ [[1, 7, 2, 8],
   [2, 4, 5, 9, 10, 1, 11, 12],
   [13, 14],
   [3, 15, 16, 17],
   [1, 18, 19, 20, 21, 5, 22],
   [2, 4, 3, 23, 1, 24],
   [1, 25, 6, 2, 26],
   [27, 28, 3, 6, 2]]

max_length = max([len(seq) for seq in sequences])
x = pad_sequences(sequences, maxlen=max_length, padding='post')
y = np.array(labels)

model = Sequential([
    Embedding(input_dim=100, output_dim=32, input_length=max_length),
    SimpleRNN(16, activation='relu'),
    Dense(1, activation='sigmoid')
])

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

model.fit(x, y, epochs=10, validation_split=0.2)

→ Epoch 1/10
1/1 3s 3s/step - accuracy: 0.5000 - loss: 0.6989 - val_accuracy: 0.5000 - val_loss: 0.6947
Epoch 2/10
1/1 0s 64ms/step - accuracy: 0.5000 - loss: 0.6923 - val_accuracy: 0.5000 - val_loss: 0.6945
Epoch 3/10
1/1 0s 71ms/step - accuracy: 0.6667 - loss: 0.6886 - val_accuracy: 0.5000 - val_loss: 0.6937
Epoch 4/10
1/1 0s 70ms/step - accuracy: 0.6667 - loss: 0.6860 - val_accuracy: 0.5000 - val_loss: 0.6935
Epoch 5/10
1/1 0s 127ms/step - accuracy: 0.6667 - loss: 0.6829 - val_accuracy: 0.5000 - val_loss: 0.6937

```

```
Epoch 6/10
1/1 ━━━━━━━━ 0s 136ms/step - accuracy: 0.6667 - loss: 0.6797 - val_accuracy: 0.5000 - val_loss: 0.6941
Epoch 7/10
1/1 ━━━━━━ 0s 64ms/step - accuracy: 0.6667 - loss: 0.6762 - val_accuracy: 0.5000 - val_loss: 0.6947
Epoch 8/10
1/1 ━━━━━━ 0s 65ms/step - accuracy: 0.6667 - loss: 0.6726 - val_accuracy: 0.5000 - val_loss: 0.6955
Epoch 9/10
1/1 ━━━━━━ 0s 63ms/step - accuracy: 1.0000 - loss: 0.6691 - val_accuracy: 0.5000 - val_loss: 0.6961
Epoch 10/10
1/1 ━━━━━━ 0s 64ms/step - accuracy: 1.0000 - loss: 0.6658 - val_accuracy: 0.5000 - val_loss: 0.6967
<keras.src.callbacks.history.History at 0x7c0f14074850>
```

```
new_sentences = [
    'I absolutely love this',
    'This is terrible',
    'I am satisfied',
    'Worst purchase ever'
]
new_sequences = tok.texts_to_sequences(new_sentences)
new_x = pad_sequences(new_sequences, maxlen=max_length, padding='post')
```

```
new_x
```

```
→ array([[ 1, 13,  7,  2,  0,  0,  0],
       [ 2,  4, 27,  0,  0,  0,  0],
       [ 1, 18,  0,  0,  0,  0,  0],
       [ 9, 22, 11,  0,  0,  0,  0]], dtype=int32)
```

```
predictions = model.predict(new_x)
predicted_labels = (predictions > 0.5).astype(int)
predicted_labels
```

```
→ 1/1 ━━━━━━ 0s 175ms/step
array([[1],
       [0],
       [1],
       [0]])
```