

RNNs

June 13, 2025

```
[1]: import tensorflow as tf
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense
```

```
[5]: #Generating sequences of random integers
def generate_sequences(num_sequences, max_length, vocab_size):
    sequences = []
    labels=[]
    for _ in range(num_sequences):
        length = np.random.randint(1, max_length + 1)
        seq = np.random.randint(0, vocab_size, length)
        label = np.sum(seq) % 2
        sequences.append(seq)
        labels.append(label)
    return sequences, labels

#Parameters
num_sequences = 1000
max_length = 10
vocab_size = 10

#Generating Data
sequences, labels = generate_sequences(num_sequences, max_length, vocab_size)
```

```
[7]: from tensorflow.keras.preprocessing.sequence import pad_sequences

#Pad sequences to the same length
padded_sequences = pad_sequences(sequences, maxlen=max_length, padding='post')

#Convert labels to numpy array
labels = np.array(labels)
```

```
[11]: # Define the model
model = Sequential()
model.add(SimpleRNN(32, input_shape=(max_length, 1)))
model.add(Dense(1, activation='sigmoid'))
```

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

```
[13]: #Training the model
x = padded_sequences.reshape((num_sequences, max_length, 1))
y = labels

#Train the model
model.fit(x,y, epochs=10, batch_size=32, validation_split=0.2)
```

```
Epoch 1/10
25/25          2s 16ms/step -
accuracy: 0.5068 - loss: 0.7382 - val_accuracy: 0.5200 - val_loss: 0.7091
Epoch 2/10
25/25          0s 4ms/step -
accuracy: 0.5254 - loss: 0.7102 - val_accuracy: 0.5100 - val_loss: 0.7073
Epoch 3/10
25/25          0s 5ms/step -
accuracy: 0.5909 - loss: 0.6963 - val_accuracy: 0.4850 - val_loss: 0.7088
Epoch 4/10
25/25          0s 5ms/step -
accuracy: 0.5593 - loss: 0.6890 - val_accuracy: 0.4800 - val_loss: 0.7111
Epoch 5/10
25/25          0s 5ms/step -
accuracy: 0.5904 - loss: 0.6783 - val_accuracy: 0.5050 - val_loss: 0.7052
Epoch 6/10
25/25          0s 5ms/step -
accuracy: 0.5566 - loss: 0.6892 - val_accuracy: 0.4800 - val_loss: 0.7095
Epoch 7/10
25/25          0s 6ms/step -
accuracy: 0.5920 - loss: 0.6791 - val_accuracy: 0.4900 - val_loss: 0.7116
Epoch 8/10
25/25          0s 4ms/step -
accuracy: 0.5791 - loss: 0.6747 - val_accuracy: 0.5250 - val_loss: 0.7070
Epoch 9/10
25/25          0s 5ms/step -
accuracy: 0.5982 - loss: 0.6708 - val_accuracy: 0.5100 - val_loss: 0.7101
Epoch 10/10
25/25          0s 8ms/step -
accuracy: 0.5933 - loss: 0.6743 - val_accuracy: 0.5150 - val_loss: 0.7086
```

```
[13]: <keras.src.callbacks.history.History at 0x1e861ab5760>
```

```
[15]: #Evaluating the model on training data
loss, accuracy = model.evaluate(x,y)
```

```
print(f"Accuracy : {accuracy * 100:.2f}%")
```

```
32/32          0s 3ms/step -  
accuracy: 0.6123 - loss: 0.6678  
Accuracy : 57.50%
```

```
[17]: # Predict on new data  
test_seq = np.array([[1, 3, 5]])  
test_seq_padded = pad_sequences(test_seq, maxlen=max_length, padding='post')  
test_seq_reshaped = test_seq_padded.reshape((1, max_length, 1))  
  
prediction = model.predict(test_seq_reshaped)  
predicted_label = int(prediction.round()[0][0])  
print(f"Predicted Label: {'Even' if predicted_label == 0 else 'Odd'}")
```

```
1/1          0s 279ms/step  
Predicted Label: Even
```