In [1]:
```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import SimpleRNN, Dense
from tensorflow.keras.optimizers import Adam
from sklearn.model_selection import train_test_split
```

```
WARNING:tensorflow:From C:\Users\91707\anaconda3\lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

In [2]:
```
# Generate a toy dataset: a simple sine wave
def create_sine_wave_dataset(seq_length, num_samples):
    X = []
    y = []
    for _ in range(num_samples):
        start = np.random.rand()
        x = np.linspace(start, start + 3 * np.pi, seq_length)
        sine_wave = np.sin(x)
        X.append(sine_wave[:-1])
        y.append(sine_wave[1:])
    return np.array(X), np.array(y)
```

In [3]:
```
seq_length = 50
num_samples = 1000
X, y = create_sine_wave_dataset(seq_length, num_samples)
```

In [4]:
```
# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [5]:
```
# Define the RNN model
model = Sequential()
model.add(SimpleRNN(50, activation='tanh', input_shape=(seq_length-1, 1)))
model.add(Dense(1))
```

```
WARNING:tensorflow:From C:\Users\91707\anaconda3\lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

In [6]:
```
model.compile(optimizer=Adam(learning_rate=0.001), loss='mse')
```

In [7]:
```
# Reshape input data to be compatible with RNN
X_train_reshaped = X_train.reshape((X_train.shape[0], X_train.shape[1], 1))
X_test_reshaped = X_test.reshape((X_test.shape[0], X_test.shape[1], 1))
```

In [8]:
```
# Train the model
history = model.fit(X_train_reshaped, y_train, epochs=20, batch_size=32, validation_split=0.1)
```

```
Epoch 1/20
WARNING:tensorflow:From C:\Users\91707\anaconda3\lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTen
sorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

23/23 [==============================] - 2s 23ms/step - loss: 0.4812 - val_loss: 0.4712
Epoch 2/20
23/23 [==============================] - 0s 10ms/step - loss: 0.4714 - val_loss: 0.4711
Epoch 3/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 4/20
23/23 [==============================] - 0s 10ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 5/20
23/23 [==============================] - 0s 8ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 6/20
23/23 [==============================] - 0s 8ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 7/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 8/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 9/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 10/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 11/20
23/23 [==============================] - 0s 8ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 12/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 13/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 14/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 15/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 16/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 17/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 18/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 19/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
Epoch 20/20
23/23 [==============================] - 0s 9ms/step - loss: 0.4712 - val_loss: 0.4711
```

In [9]:
```
# Evaluate the model on the test data
test_loss = model.evaluate(X_test_reshaped, y_test)
print(f'Test Loss: {test_loss}')
```

```
7/7 [==============================] - 0s 5ms/step - loss: 0.4710
Test Loss: 0.4709557294845581
```

In [10]:
```
# Make predictions
y_pred = model.predict(X_test_reshaped)
```

```
7/7 [==============================] - 0s 4ms/step
```

In [11]:
```
# Example: print the first few predictions
print(f'Predicted values: {y_pred[:5].flatten()}')
print(f'True values: {y_test[:5].flatten()}')
```

```
Predicted values: [0.11373591 0.18665263 0.1879123  0.20739144 0.12847812]
True values: [ 8.94268500e-01  9.63326816e-01  9.96856015e-01  9.93619483e-01
  9.53736590e-01  8.78678283e-01  7.71212840e-01  6.35303766e-01
  4.75963619e-01  2.99069131e-01  1.11144469e-01 -8.08793879e-02
 -2.69920277e-01 -4.49006051e-01 -6.11531725e-01 -7.51503078e-01
 -8.63757730e-01 -9.44155541e-01 -9.89731304e-01 -9.98804107e-01
 -9.71039331e-01 -9.07460988e-01 -8.10413953e-01 -6.83477485e-01
 -5.31333215e-01 -3.59592480e-01 -1.74589369e-01  1.68528932e-02
  2.07673592e-01  3.90834939e-01  5.59581634e-01  7.07690015e-01
  8.29697595e-01  9.21104529e-01  9.78539574e-01  9.99884431e-01
  9.84351864e-01  9.32514740e-01  8.46284901e-01  7.28842648e-01
  5.84519449e-01  4.18638187e-01  2.37316843e-01  4.72428536e-02
 -1.44573532e-01 -3.31057802e-01 -5.05332101e-01 -6.60968901e-01
 -7.92228054e-01  5.66500416e-01  7.13580129e-01  8.34341802e-01
  9.24331544e-01  9.80230379e-01  9.99976666e-01  9.82842127e-01
  9.29458714e-01  8.41795296e-01  7.23085048e-01  5.77706204e-01
  4.11020582e-01  2.29175827e-01  3.88786822e-02 -1.52852374e-01
 -3.38945978e-01 -5.12538682e-01 -6.67228096e-01 -7.97309012e-01
 -8.97983828e-01 -9.65539486e-01 -9.97484420e-01 -9.92640447e-01
 -9.51186220e-01 -8.74650642e-01 -7.65856474e-01 -6.28816227e-01
 -4.68584178e-01 -2.91069955e-01 -1.02820581e-01  8.92209899e-02
  2.77971940e-01  4.56470816e-01  6.18134279e-01  7.56999908e-01
  8.67946103e-01  9.46880983e-01  9.90893296e-01  9.98359794e-01
  9.69005099e-01  9.03911862e-01  8.05480831e-01  6.77342310e-01
  5.24222262e-01  3.51768014e-01  1.66339967e-01 -2.52229772e-02
 -2.15855656e-01 -3.98527214e-01  5.56612704e-01  7.05157554e-01
  8.27695004e-01  9.19705667e-01  9.77796034e-01  9.99823635e-01
  9.84976055e-01  9.33800897e-01  8.48185587e-01  7.31287764e-01
  5.87418814e-01  4.21884868e-01  2.40791097e-01  5.08165441e-02
 -1.41032208e-01 -3.27679455e-01 -5.02241332e-01 -6.58279701e-01
 -7.90039604e-01 -8.92661516e-01 -9.62360565e-01 -9.96566134e-01
 -9.94016664e-01 -9.54806184e-01 -8.80380841e-01 -7.73485569e-01
 -6.38062845e-01 -4.79107288e-01 -3.02481446e-01 -1.14699578e-01
  7.73126040e-02  2.66473366e-01  4.45806142e-01  6.08696835e-01
  7.49137763e-01  8.61949227e-01  9.42970550e-01  9.89213530e-01
  9.98972646e-01  9.71887967e-01  9.08958422e-01  8.12504957e-01
  6.86084939e-01  5.34360953e-01  3.62928833e-01  1.78111286e-01
 -1.32753055e-02 -2.04172281e-01 -3.87539039e-01  3.15941021e-01
  4.91482062e-01  6.48896417e-01  7.82378378e-01  8.87004905e-01
  9.58917195e-01  9.95463003e-01  9.95294456e-01  9.58417773e-01
  8.86193025e-01  7.81283985e-01  6.47559874e-01  4.89952664e-01
  3.14275173e-01  1.27006683e-01 -6.49460267e-02 -2.54503418e-01
 -4.34674293e-01 -5.98813648e-01 -7.40867746e-01 -8.55597393e-01
 -9.38771165e-01 -9.87321474e-01 -9.99457702e-01 -9.74732245e-01
```

```
-9.14057019e-01 -8.19669829e-01 -6.95051834e-01 -5.44799156e-01
-3.74453366e-01 -1.90297105e-01  8.77634567e-04  1.92020005e-01
 3.76080355e-01  5.46270226e-01  6.96312731e-01  8.20674048e-01
 9.14767523e-01  9.75122829e-01  9.99513961e-01  9.87041333e-01
 9.38164956e-01  8.54687474e-01  7.39687677e-01  5.97406952e-01
 4.33092851e-01  2.52805555e-01  6.31943639e-02 -1.28747542e-01
 8.51241853e-01  9.35859233e-01  9.85960547e-01  9.99697973e-01
 9.76564853e-01  9.17414374e-01  8.24428107e-01  7.01035541e-01
 5.51787602e-01  3.82188806e-01  1.98494243e-01  7.47887736e-03
-1.83812322e-01 -3.68324213e-01 -5.39251686e-01 -6.90290648e-01
-8.15870526e-01 -9.11359724e-01 -9.73236439e-01 -9.99218554e-01
-9.88347803e-01 -9.41025119e-01 -8.58995842e-01 -7.45285349e-01
-6.04087477e-01 -4.40609840e-01 -2.60881770e-01 -7.15319393e-02
 1.20456109e-01  3.08001534e-01  4.84187342e-01  6.42515505e-01
 7.77146613e-01  8.83115244e-01  9.56513095e-01  9.94633131e-01
 9.96069419e-01  9.60768989e-01  8.90033778e-01  7.86472621e-01
 6.53905027e-01  4.97220314e-01  3.22197277e-01  1.35291061e-01
-5.66049169e-02 -2.46413210e-01 -4.27133369e-01 -5.92100129e-01
-7.35229238e-01]
```