```
import numpy as np
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, SimpleRNN
from tensorflow.keras.optimizers import Adam

# 1. Gather and preprocess sequential customer data
# Replace this with your own sequential customer data preprocessing
# Example synthetic data
num_samples = 1000
seq_length = 20
num_features = 10
customer_sequences = np.random.rand(num_samples, seq_length, num_features)
customer_labels = np.random.randint(0, 3, size=(num_samples,))  # Three customer segments

# 2. Define RNN architecture
model = Sequential([
    SimpleRNN(64, input_shape=(seq_length, num_features), activation='relu'),
    Dense(32, activation='relu'),
    Dense(3, activation='softmax')  # Three output classes for segmentation
])

# 3. Compile the model
model.compile(optimizer=Adam(), loss='sparse_categorical_crossentropy', metrics=['accuracy'])

# 4. Train the model
model.fit(customer_sequences, customer_labels, epochs=10, batch_size=32)

# 5. Evaluate the model
loss, accuracy = model.evaluate(customer_sequences, customer_labels, verbose=2)
print('\nTest accuracy:', accuracy)

# 6. Use the trained model for customer segmentation
# Replace customer_data with your new or unseen sequential data
predicted_segments = model.predict_classes(customer_sequences)
```