EM Lab05: Lab05 Motion Recognition using IMU Sensor Fusion

Lecture Notes - Prof. Tobias Schaffer

1 Objective

In this lab, you will:

- Collect IMU data (accelerometer + gyroscope) using the Raspberry Pi Sensor HAT.
- Label data for four motion types: **move_none**, **move_circle**, **move_shake** and **move_twist**.
- Train a fully connected neural network on short 1-second sensor sequences (50 time steps, 6 features).
- Convert the model to TensorFlow Lite and deploy it to the Raspberry Pi.
- Control the LED matrix of the Sensor HAT based on the detected motion class.

2 Data Collection on Raspberry Pi

Use the following script to collect labeled motion data. Each sample is recorded for 1 second at 50 Hz (50 time steps).

```
import os
import time
import numpy as np
from sense_hat import SenseHat

sense = SenseHat()
LABEL = "move_circle"  # change to "move_shake", "move_twist" or "move_none" for
    other classes
SAMPLES = 50  # 50 for 1s
FREQ_HZ = 50
DELAY = 1.0 / FREQ_HZ

save_dir = f"./motion_data/{LABEL}"
os.makedirs(save_dir, exist_ok=True)

print(f"Recording␣samples␣for␣label:␣{LABEL}")
try:
    while True:
        input("Press␣Enter␣to␣record␣1␣second...")
        data = []
        for _ in range(SAMPLES):
            acc = sense.get_accelerometer_raw()
            gyro = sense.get_gyroscope_raw()
            sample = [
                acc[’x’], acc[’y’], acc[’z’],
                gyro[’x’], gyro[’y’], gyro[’z’]
            ]
            data.append(sample)
            time.sleep(DELAY)
        timestamp = int(time.time())
        np.save(f"{save_dir}/{LABEL}_{timestamp}.npy", np.array(data))
        print(f"Saved␣{LABEL}_{timestamp}.npy")
except KeyboardInterrupt:
    print("Recording␣stopped.")
```

3 Transfer Data to PC using SCP

Use this command to copy your motion data to your PC:

```
scp -r pi@<RASPBERRY_PI_IP>:/home/pi/motion_data ./motion_data
```

4 Training in Google Colab

Upload zipped folder `motion_data.zip` with ‘.npy‘ files to Colab. Each file contains a (50 × 6) time window. Unzip in Colab using `!unzip motion_data.zip`.

Preprocessing and Training (TensorFlow)

```
import os
import numpy as np
import tensorflow as tf
from sklearn.model_selection import train_test_split

# Load data
data = []
labels = []
label_map = {"move_none": 0, "move_circle": 1, "move_shake": 2, "move_twist": 3}

for label_name in os.listdir("motion_data"):
    for file in os.listdir(f"motion_data/{label_name}"):
        if file.endswith(".npy"):
            sample = np.load(f"motion_data/{label_name}/{file}")
            sample = sample.flatten()  # Shape: (300,)
            data.append(sample)
            labels.append(label_map[label_name])

X = np.array(data)
y = tf.keras.utils.to_categorical(labels, num_classes=4)

X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state
    =42)

# Fully connected model
model = tf.keras.Sequential([
    tf.keras.layers.Input(shape=(300,)),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(64, activation='relu'),
    tf.keras.layers.Dense(4, activation='softmax')
])

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

model.fit(X_train, y_train, validation_data=(X_val, y_val), batch_size=1, epochs=15)
```

5 Convert Model to TensorFlow Lite

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

with open("motion_model.tflite", "wb") as f:
    f.write(tflite_model)
```

6 Transfer TFLite Model to Raspberry Pi

```
scp motion_model.tflite pi@<RASPBERRY_PI_IP>:/home/pi/
```

7 Deploy and Control LEDs on Raspberry Pi

```
import tensorflow as tf
import numpy as np
import time
from sense_hat import SenseHat

sense = SenseHat()
sense.clear()

# Load TFLite model
interpreter = tf.lite.Interpreter(model_path="motion_model.tflite")
interpreter.allocate_tensors()
input_details = interpreter.get_input_details()
output_details = interpreter.get_output_details()

LABELS = ["move_none", "move_circle", "move_shake", "move_twist"]
COLORS = {
    "move_none": [0, 0, 0],
    "move_circle": [255, 0, 0],
    "move_shake": [0, 255, 0],
    "move_twist": [0, 0, 255]
}
SAMPLES = 50
FREQ_HZ = 50
DELAY = 1.0 / FREQ_HZ

def read_imu_sample():
    acc = sense.get_accelerometer_raw()
    gyro = sense.get_gyroscope_raw()
    time.sleep(DELAY)
    return [acc['x'], acc['y'], acc['z'], gyro['x'], gyro['y'], gyro['z']]

try:
    while True:
        print("Collecting␣1s␣sample...")
        samples = [read_imu_sample() for _ in range(50)]
        input_data = np.array(samples).flatten().astype(np.float32)
        input_data = np.expand_dims(input_data, axis=0)

        start_time = time.time()
        interpreter.set_tensor(input_details[0]['index'], input_data)
        interpreter.invoke()
        output = interpreter.get_tensor(output_details[0]['index'])[0]
        end_time = time.time()
        predicted_index = int(np.argmax(output))
        label = LABELS[predicted_index]

        print(f"Predicted:␣{label}␣(pred.␣time:␣{end_time␣-␣start_time:.8f}s)")
        sense.clear(COLORS[label])
except KeyboardInterrupt:
    sense.clear()
    print("Stopped.")
```

8 Available resources

The following files and data is available in the shared folder `/Labs/Lab05 Gesture Detection`:

- Python scripts for capturing data and for predictin on Raspberry Pi: `capture.py` and `predict.py`
- Sample data for the four motions in the folder `motion_data` and the zipped folder `motion_data.zip`
- Jupyter/Colab Notebook for training `Lab05 Gesture Train.ipynb`
- Trained `motion_model.tflite` TFLite model