**BÁO CÁO TUẦN 6 ĐỒ ÁN MÔN HỌC**

Tên đồ án: “Ứng dụng các mô hình máy học vào bài toán phân loại hoạt động của người dùng, trên các thiết bị đeo tay theo dõi sức khỏe.”

Thành viên thực hiện:

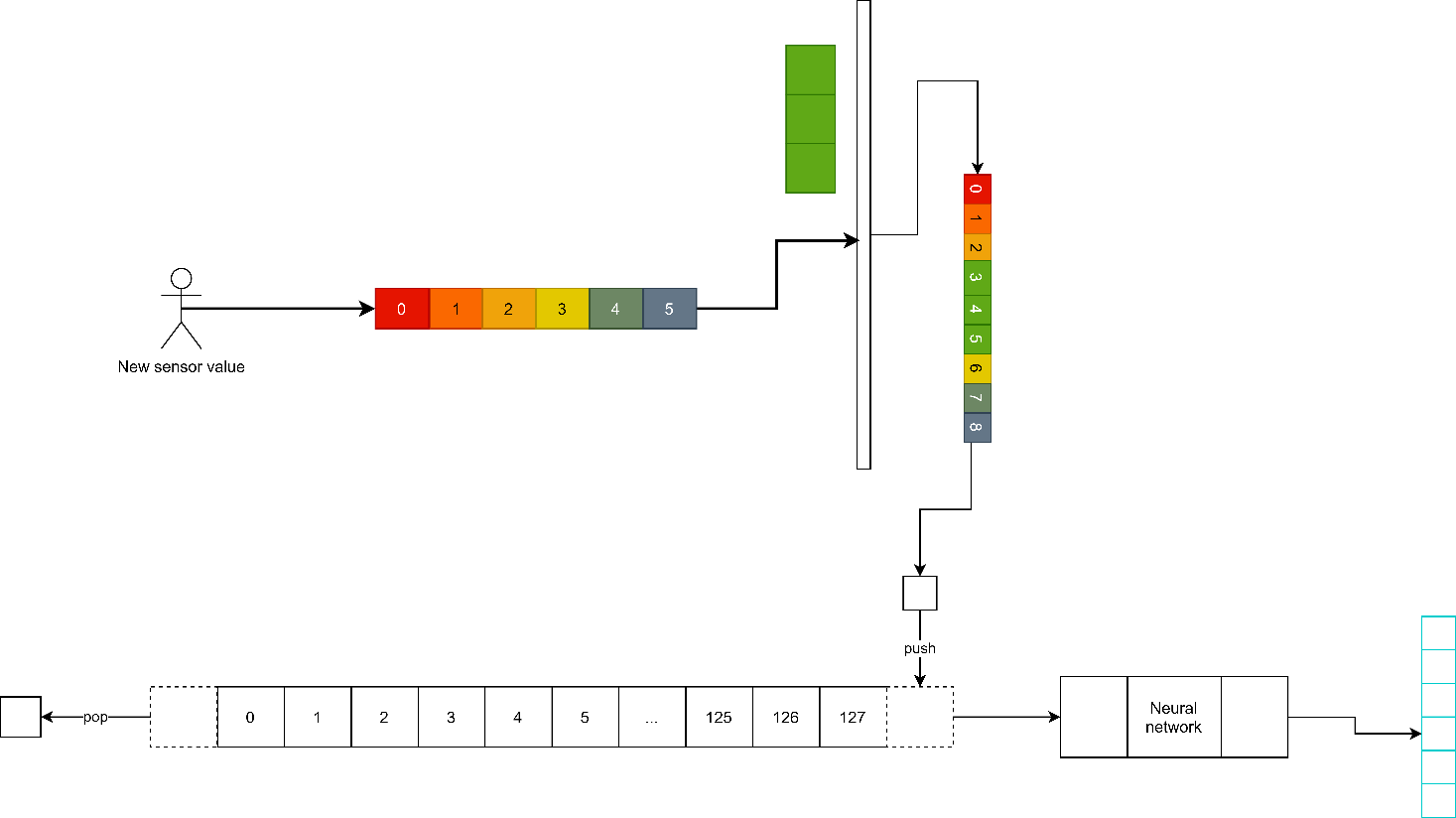
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | HỌ VÀ TÊN | MSSV | ĐIỆN THOẠI | EMAIL |
| 1 | Nguyễn Ngọc Minh | 19520165 | 0585115056 | 19520165@gm.uit.edu.vn |

1. Nội dung công việc trong tuần

Triển khai model xuống MCU.

1. Báo cáo quá trình thực hiện
2. Tổng quan mô hình:

* Tensor đầu vào: (128, 9).
* 128 giá trị được đọc liên tục từ sensor.
* 9 features: 3 trục gia tốc tổng thể, 3 trục tốc độ góc (lấy thô từ sensor), 3 trục body motion (đã loại bỏ gia tốc trọng trường).
* Tensor đầu ra: (1, 6)
* 6 xác suất của các loại hoạt động => hoạt động có xác suất cao nhất được lựa chọn làm kết quả.

1. Mô hình dữ liệu:
2. Tính toán 3 trục body motion

* Dữ liệu lấy từ cảm biến gia tốc (A) đã bao gồm gia tốc chuyển động của vật thể (v) và hiệu ứng gia tốc trọng trường (g).

= + => = –

A: Dữ liệu từ cảm biến gia tốc

g: gia tốc trọng trường

v: gia tốc vật thể

* Gia tốc trọng trường (g) có thể tính toán tương đối bằng cách lấy dữ liệu của cảm biến (A) cho qua bộ lọc thông thấp.

RC = dt =

alpha =

gt = gt-1 + alpha\*(At – gt-1)

Fc: cut off frequency (Hz)

Fs: sample rate (Hz)

gt, gt-1 : gia tốc trọng trường tại t, t-1.

At : giá trị cảm biến tại t.

1. Chuẩn bị input cho model

* Đầu vào dữ liệu cho network là một tensor có dạng (128, 9), giá trị sensor được cập nhật liên tục (200Hz) và cho vào đầu hàng đợi, giá trị cũ nhất cuối hàng đợi sẽ được loại bỏ và cứ thế lặp lại cho đến khi network được yêu cầu chạy(2.5Hz).

1. Chạy network lấy kết quả

Har\_InputTypeDef\*in\_data=(Har\_InputTypeDef \*)**malloc**(128\***sizeof**(Har\_InputTypeDef)); //input tensor: mảng 128 phần từ

Har\_InputTypeDef \*mpudata = (Har\_InputTypeDef\*) **malloc**(**sizeof**(Har\_InputTypeDef)); //newest input sensor

**if** (data\_count < 128) {

((Har\_InputTypeDef\*) in\_data)[data\_count] = \*mpudata;

data\_count++;

}

**else** {

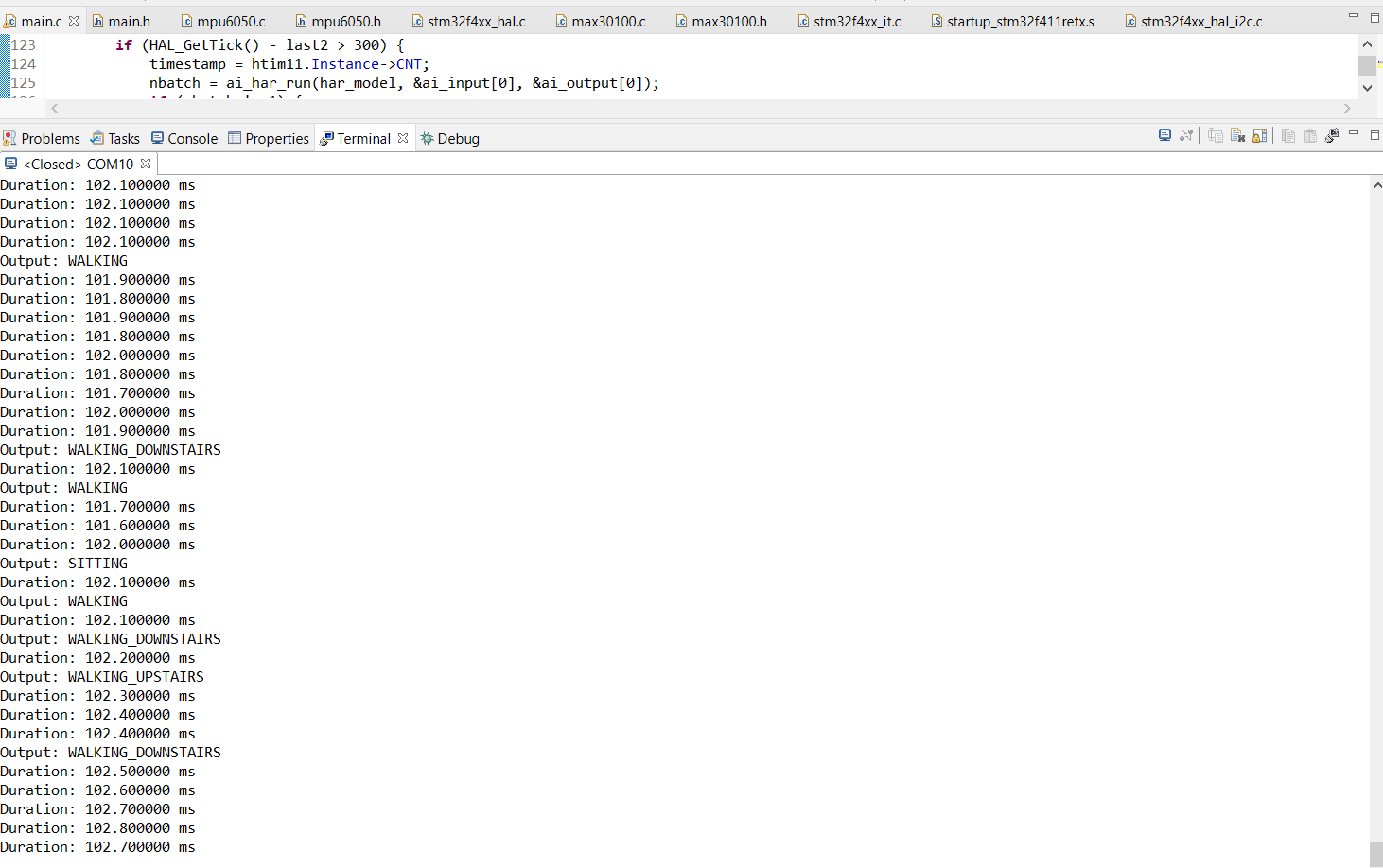
**memmove**((Har\_InputTypeDef\*) in\_data,

((Har\_InputTypeDef\*) in\_data + 1),

**sizeof**(Har\_InputTypeDef) \* 127);

((Har\_InputTypeDef\*) in\_data)[127] = \*mpudata;

}

* Network được chạy định kỳ mỗi 300ms và tốn 100ms để hoàn thành.
* 2.5Hz
* Kết quả chạy thử nghiệm ban đầu:

ai\_har\_run(har\_model, ai\_input, ai\_output);

**float** \*y\_val = (**float**\*) **malloc**(6 \* **sizeof**(**float**));

y\_val = ((**float**\*) ai\_output);

max(y\_val) => predection

* Thời gian chạy network đúng như dự đoán trước đó (khoảng 100ms).
* Nhưng vẫn còn tồn đọng hạn chế như sau: bởi vì khi chạy network tốn 100ms sẽ bị miss các gói dữ liệu không được đọc và cập nhật từ sensor ( khoảng 20 gói) có thể khiến dự đoán không chính xác.
* Giải pháp: sử dụng định thời và ngắt để định kỳ vẫn đọc dữ liệu từ sensor và lưu ở nơi khác, sau khi chạy network sẽ được push lại vào hàng đợi. Do đó sẽ tốn thêm bộ nhớ ram để làm buffer tạm.

1. Công việc tuần kế tiếp:

* Chỉnh sửa, tối ưu hóa lại dữ liệu đầu vào cho network. (Tuần 7)
* Hiển thị kết quả dự đoán ra LCD. (Tuần 7)
* Tìm hiểu và triển khai giải thuật đếm bước và đo quãng đường đi. (Tuần 8, 9).