MOTION LEARNING AND RECOGNITION

Powered by a NeuroMem network

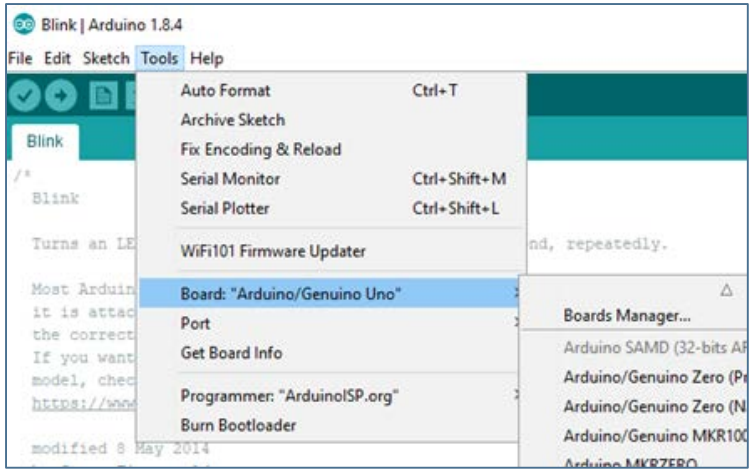
Assembly

- Arduino microcontroller board with at least 3 KB of dynamic memory
- NeuroShield featuring
  - InvenSense Accel/Gyro
  - 576 NeuroMem neurons
- Optional set of spacers

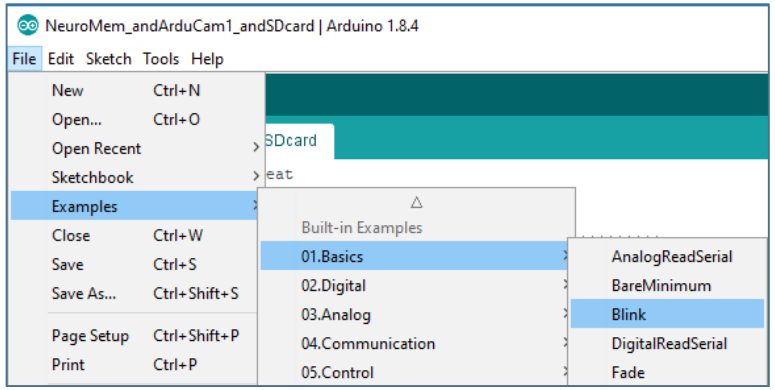
Installation

1. Requirements: Arduino IDE (you can download the latest version at https://www.arduino.cc/en/Main/Software
2. Select your Arduino board under Tools\Board menu. If not in the list, select Board Manager and install its driver
3. Load the File\Examples\Basic\Blink script
4. Upload the script to your board
5. Verify that the LED of the microcontroller board is blinking

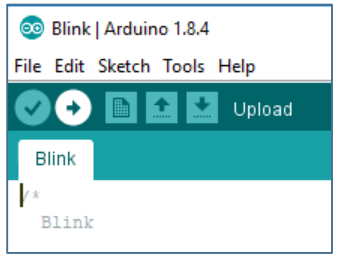
2



3



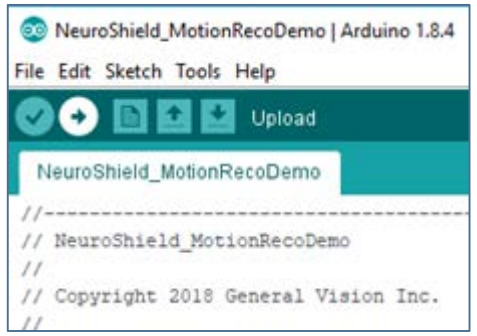
4



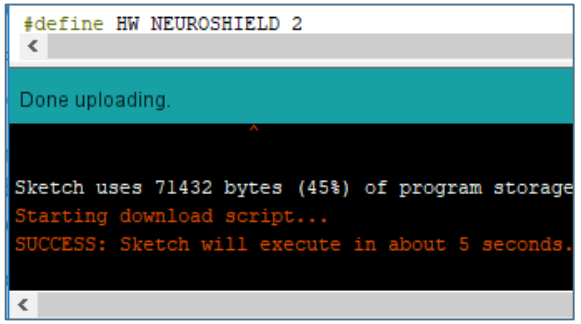
Running the demo

1. Load the NeuroShield_MotionRecoDemo
2. Upload the script to the Arduino microcontroller board
3. Verify that the upload was successful. If not, go to troubleshooting slide
4. Open the serial Monitor
5. Follow the instructions on the screen. If the window is blank and frozen, go to the Troubleshooting slide

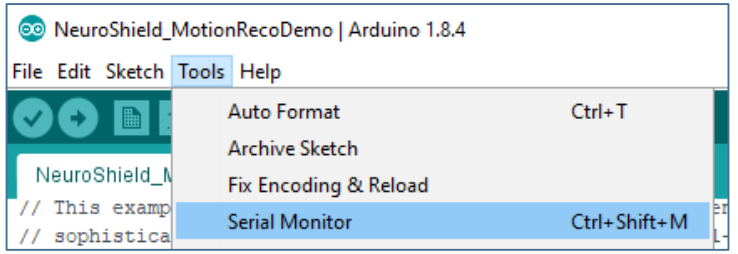
2



3

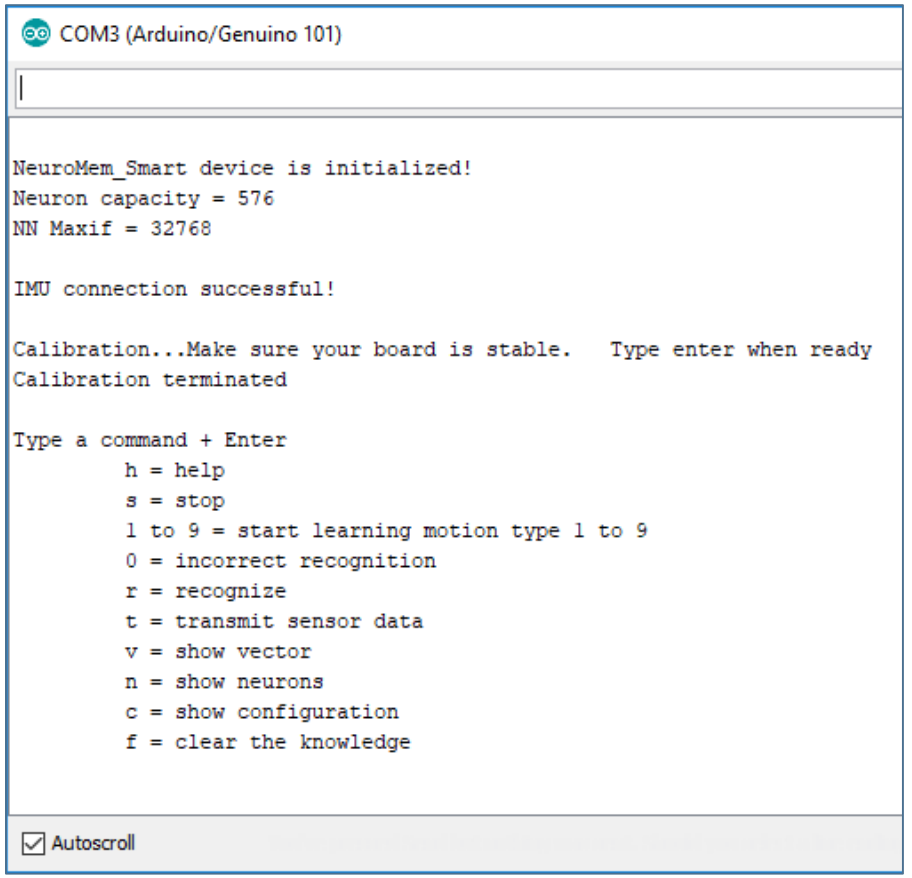


4



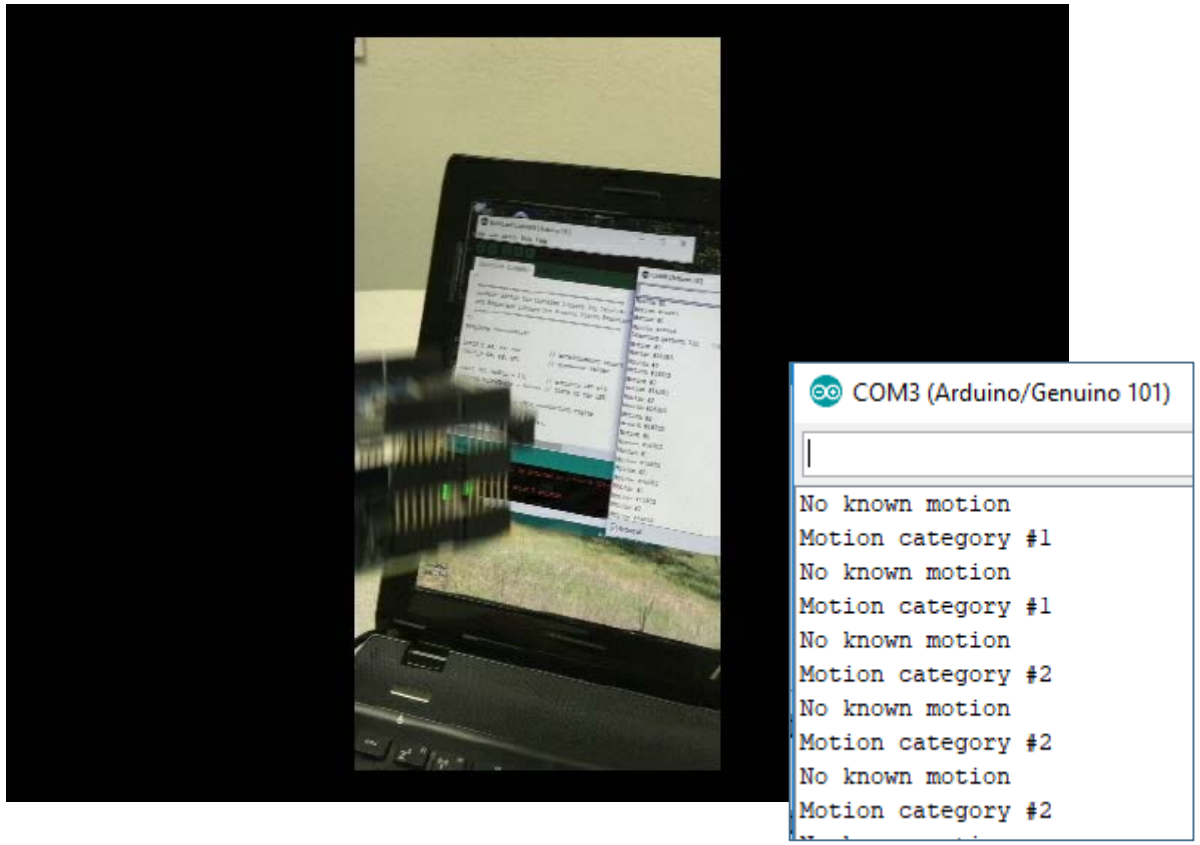
Academic training case

- Move the board up and down
- Type 1+Enter to teach vertical motion
- Move the board left and right
- Type 2+Enter to teach horizontal motion
- Stop moving
- Type 0 + Enter to teach “No motion”

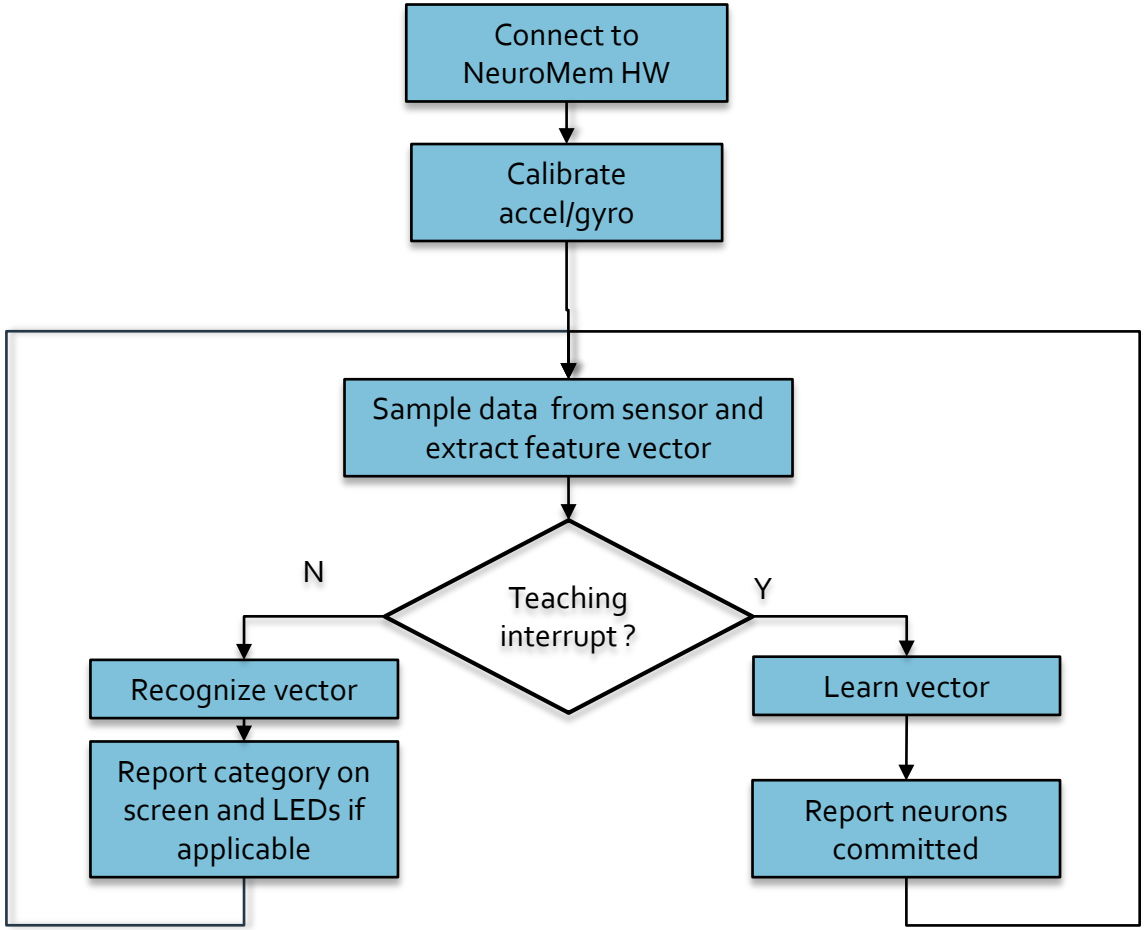


Continuous recognition

- Report 1 for up-down motion
- Report 2 for left-right motion
- Report Unknown for anything else



Script workflow



About the feature Extraction

// This example is very academic and assemble a simple sequence pattern which should be more
// sophisticated to address a real-life problem such as real-time sampling rate and calibration
// adequate for the type of motion being studied.
// More advanced feature extractions can include waveform profiles, distribution of peaks and zero crossing, etc.

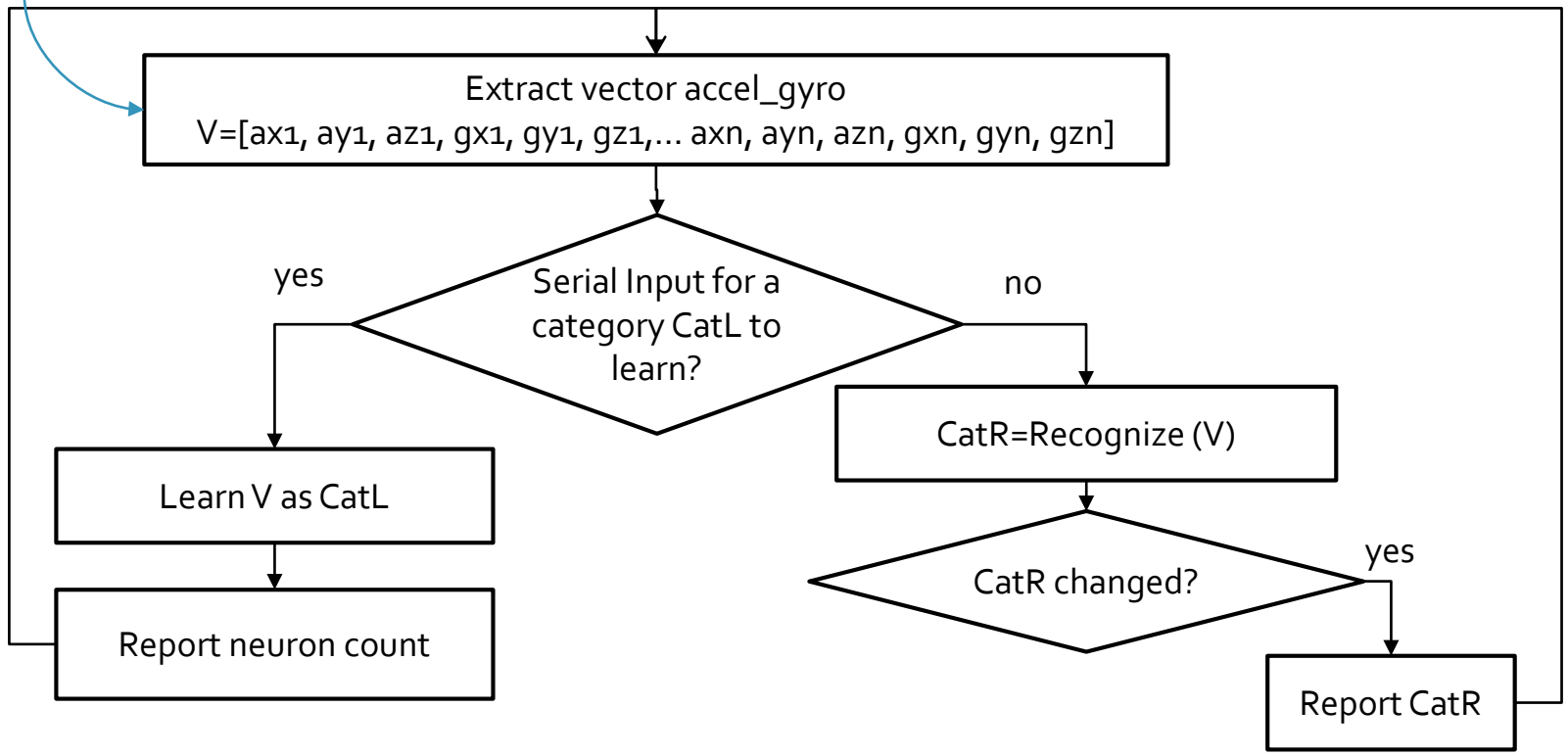
NeuroShield_MotionRecoDemo

```
for (int j=0; j<sampleNbr; j++)
{
  getSensorData();
  vector[(j*channelNbr)]= (int)(ax);
  vector[(j*channelNbr)+1]= (int)(ay);
  vector[(j*channelNbr)+2]= (int)(az);
  vector[(j*channelNbr)+3]= (int)(gx);
  vector[(j*channelNbr)+4]= (int)(gy);
  vector[(j*channelNbr)+5]= (int)(gz);
```

Collect N consecutive measurements of ax, ay, az, gx, gy, gz and append to a feature vector

Example using 1 feature

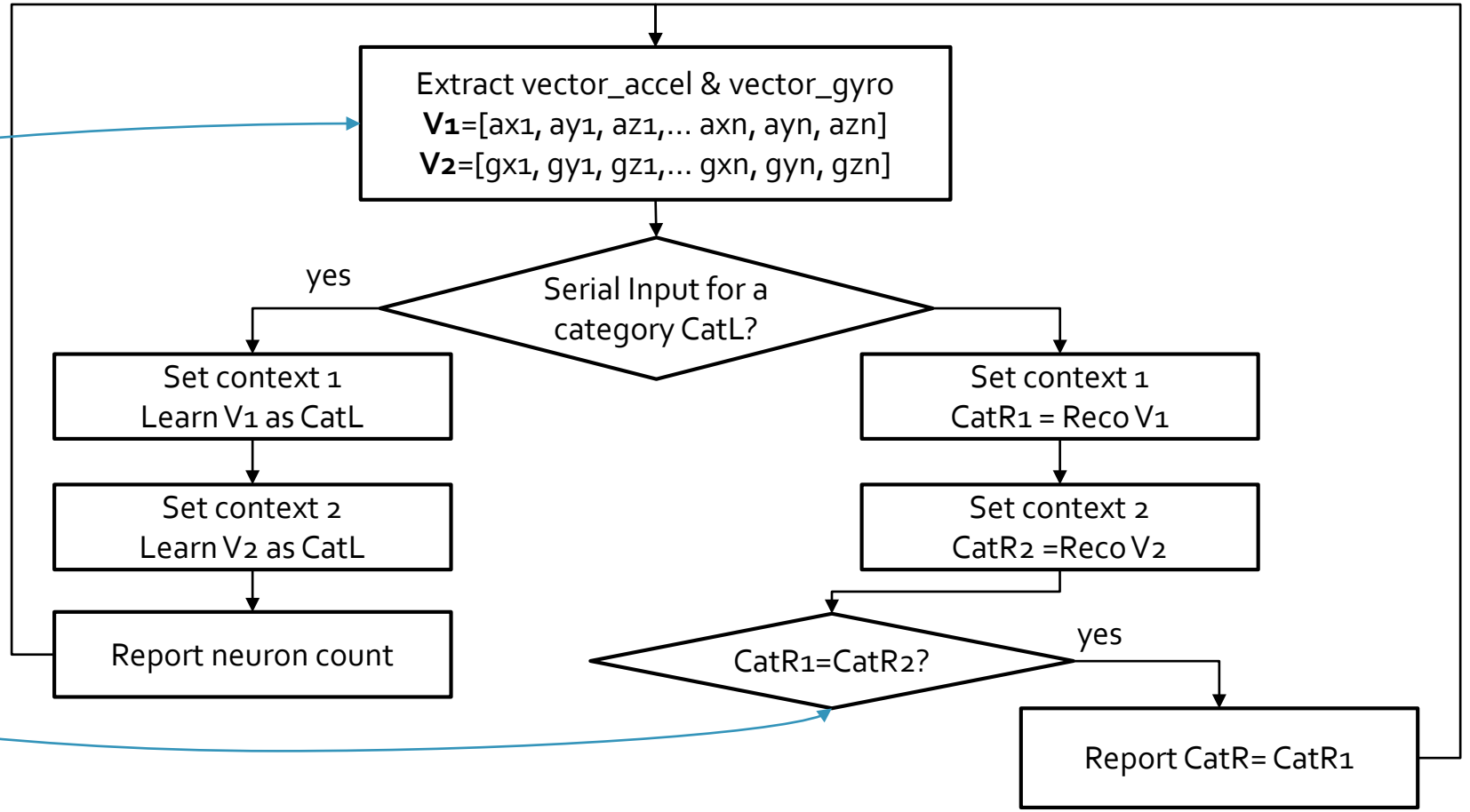
Quick & simple sampling, BUT combines data with different dimensions, different calibration and normalization



Examples using 2 features

More robust and conservative:

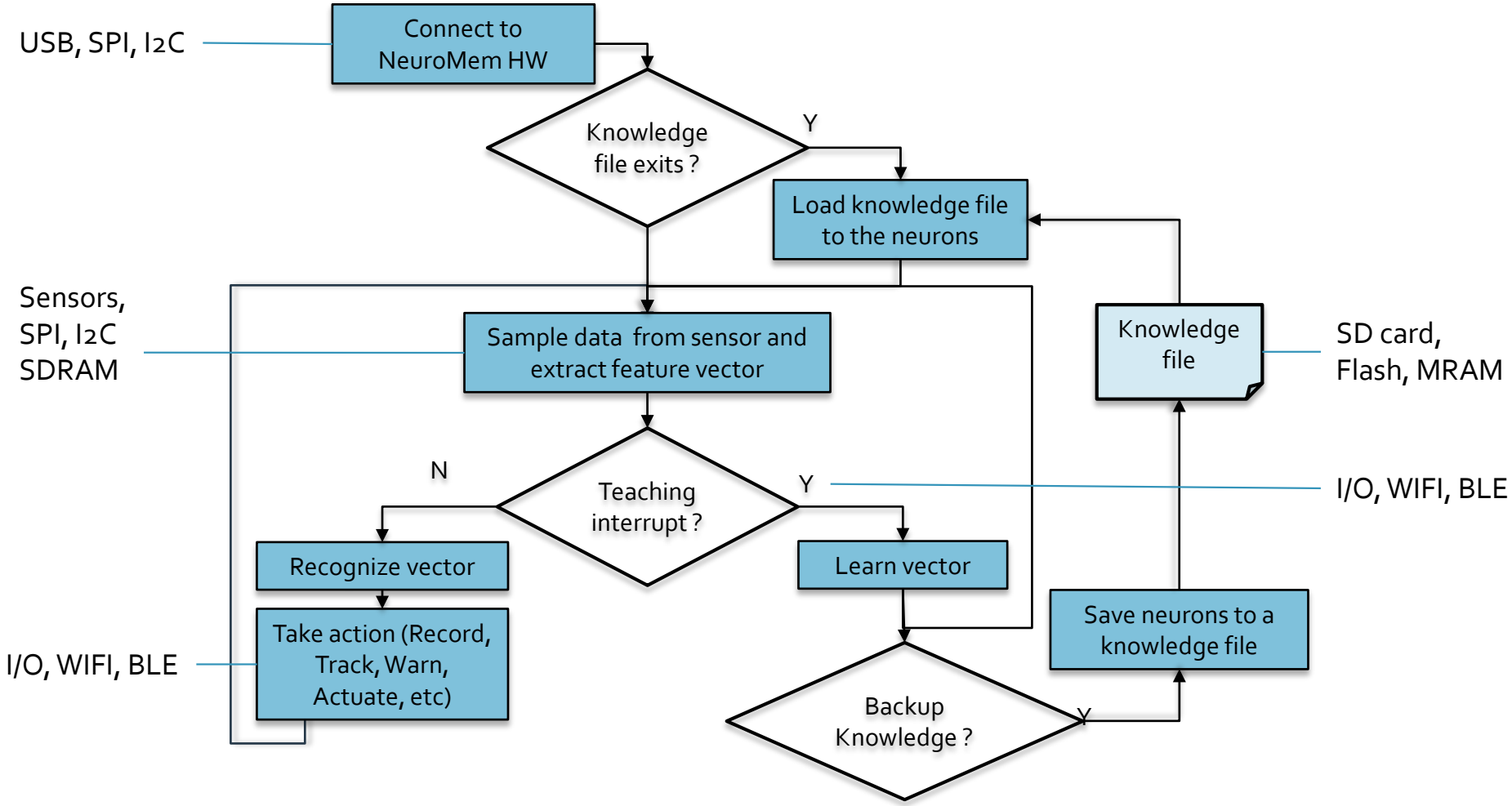
1) Train 2 sub-networks on 2 different data with different dimensions, calibration and normalization
2) Require that the 2 sub-networks agree to output a positive recognition



What is next?

- Improve the calibration routine
- Extract more advanced feature(s) depending on the application
- Improve UI to better synchronize the teaching instruction to the real-time motion
- And more….

Typical signal monitoring workflow



Troubleshooting (General)

- Script does not load properly
  - Verify the selected board platform under Tools menu
  - Verify the selected COM port under Tools menu
  - Use the external power supply of the microcontroller board instead of the USB power supply
  - Unplug all shields for the duration of the upload
- Serial monitor is blank and frozen
  - Verify that its selected baud rate matches the one of the uploaded script
  - Close its window
  - Unplug / Replug the Arduino board (possibly closing / re-opening the Arduino IDE in between)
  - Wait a few seconds
  - Open the serial monitor