Hello ..

- Who am I ?
  - Senior embedded firmware engineer @ xtrava.ai
  - Technical author (4 books) @ simplyarduino.com
  - Like to hack stuff

By The Way..

We are hiring :D
jobs@xtrava.ai

- How to contact ?
  - LinkedIN: https://www.linkedin.com/in/abdallahali/
  - Email: abdallah.ali.abdallah.elmasry@gmail.com
  - FB: https://www.facebook.com/abdallah.ali.elmasry

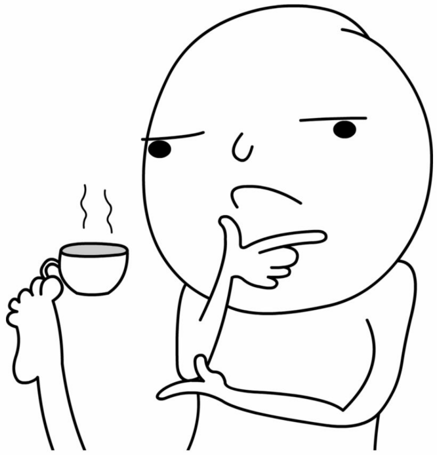
Intro ...

- Why ANN + Embedded Systems?
- Tools to develop embedded ANN
- Example project

The Beginning ...

Design an embedded system is about asking 4 Questions:

What do you want?



The Beginning ...

Design an embedded system is about asking 4 Questions:

What do you want?

Let's say we way to make a motor speed controller

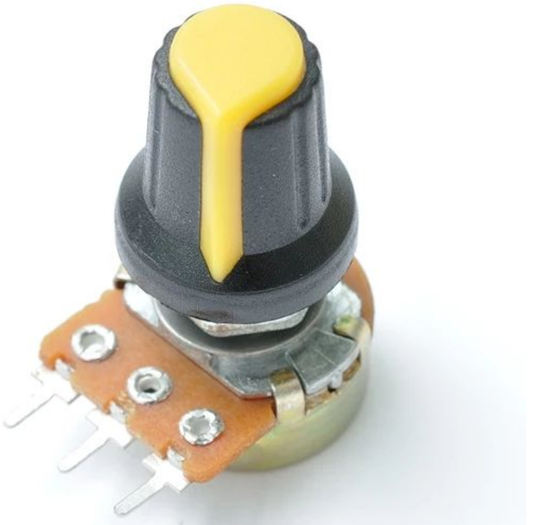
The Beginning ...

Design an embedded system is about asking 4 Questions:

What do you want?
What is you inputs?

Potentiometer

- ★ Turn right to increase speed
- ★ Turn left to decrease speed



The Beginning ...

Design an embedded system is about asking 4 Questions:

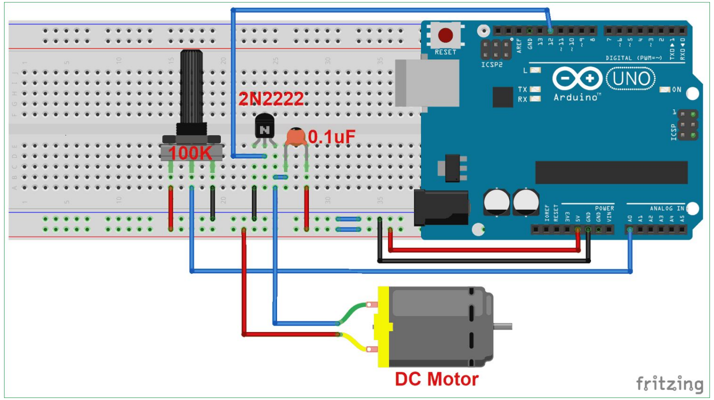
What do you want?
What is you inputs?
Whats is your outputs?

Motor speed - controlled by PWM

The Beginning ...

Design an embedded system is about asking 4 Questions:

What do you want?
What is you inputs?
Whats is your outputs?
How to ***model a relation*** between in/out?

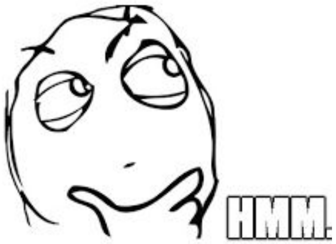


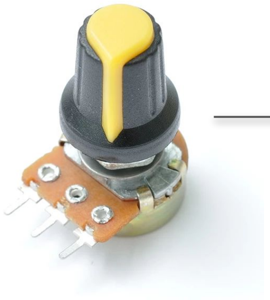
Start reading datasheets
and experimenting

The Beginning …

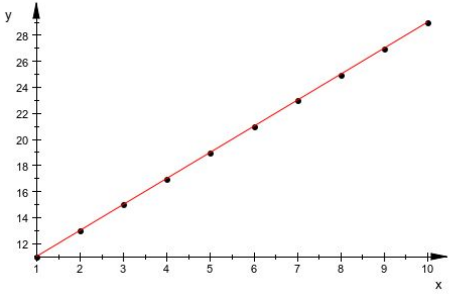
How to ***model a relation*** between in/out?

Turn right

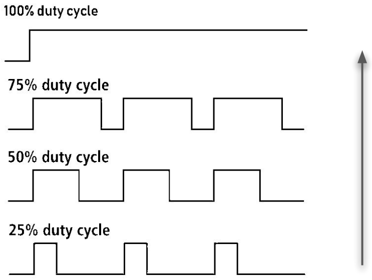




Microcontroller reads
voltage increase



Increase Motor speed
By increasing PWM



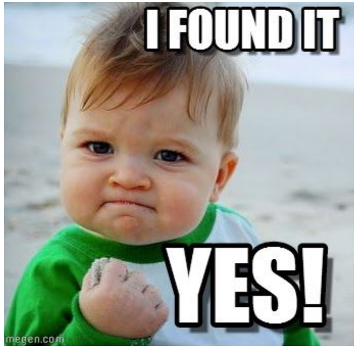
The Beginning …

Simple model

Output PWM = Potentiometer voltage x constant factor

Code:

```
while(1){
   updateMotor_speed();
}
void updateMotor_speed(){
    uint16_t voltage = analogRead(A0);
    uint16_t PWM_value = voltage * output_factor;
    analogWrite(motor_pin, PWM_value);
}
```

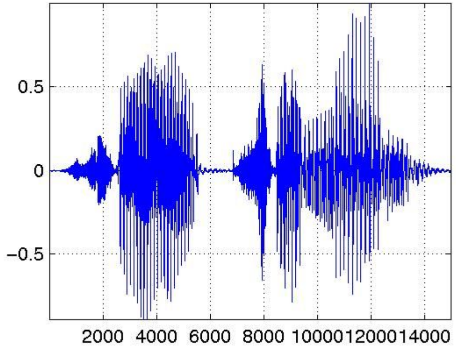


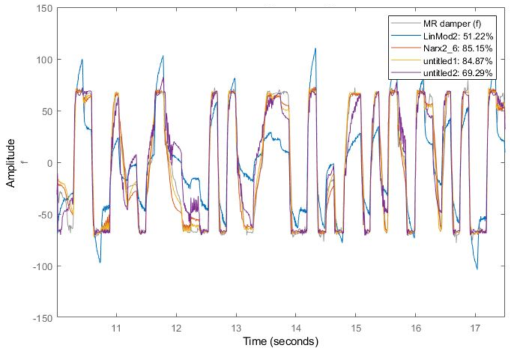
Now .. What if ?

There is no Model ...

Hard to calculate due to inaccuracy ..

Or Model may vary with different conditions ..

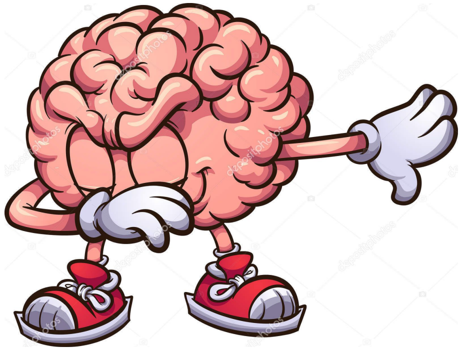




Solution !

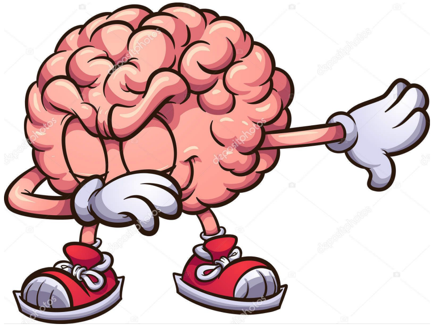
Is your **Brain**

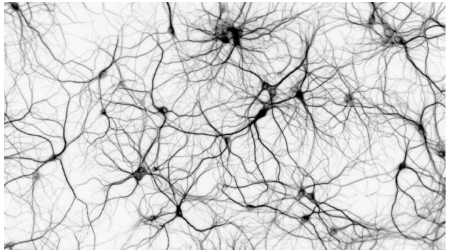
Neural networks can identify and model patterns quickly and efficiently

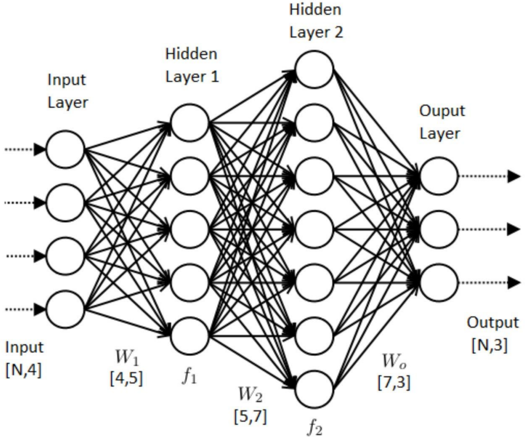


Good news

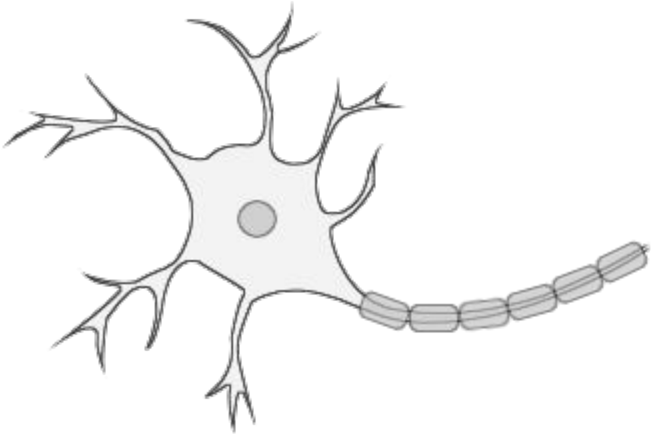
Luckily we found a way to model our brains !

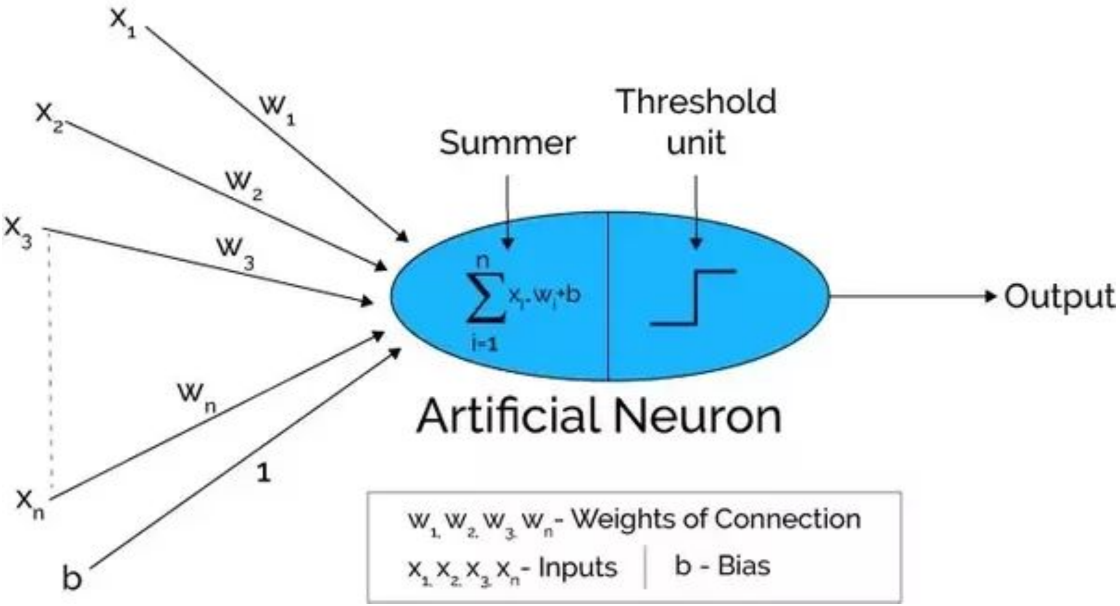






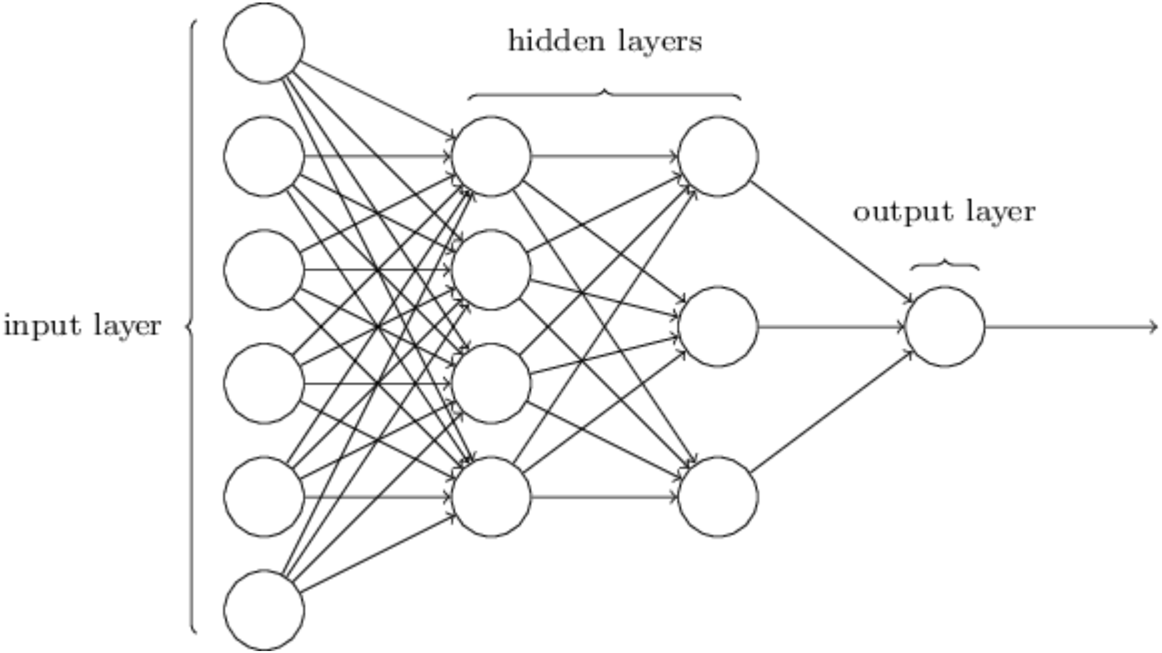
How it Works





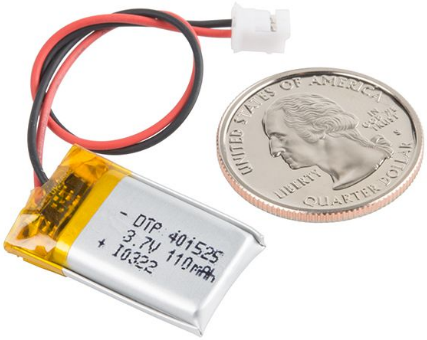
x_1
x_2
x_3
x_n
b
w_1
w_2
w_3
w_n
1
Summer
Threshold unit
$\sum_{i=1}^{n} x_i . w_i + b$
Output
Artificial Neuron
w_1, w_2, w_3, w_n - Weights of Connection
x_1, x_2, x_3, x_n - Inputs
b - Bias

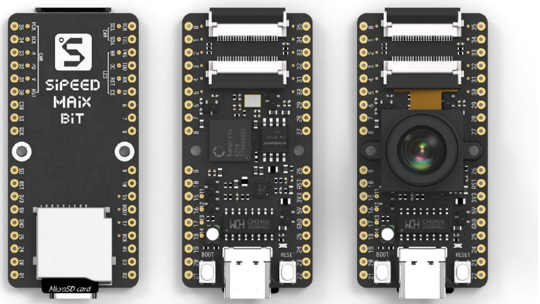
How it Works



Benefits of embedded NN

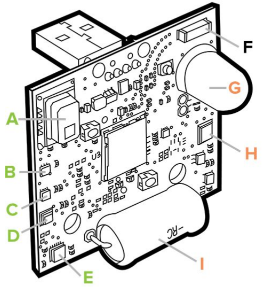
- Low power/price smart devices can run for Months/years on battery





Benefits of embedded NN

- Low power/price smart devices can run for years on battery
- Sensor fusion ... whole new level

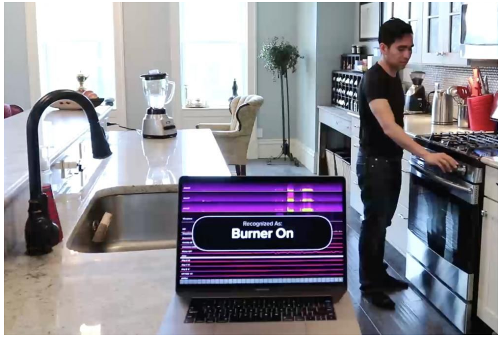


ID	Name and Sensor	Chan.	Freq.
A	GridEye AMG8833	16	10Hz
B	Color/Illum. TCS34725	4	10Hz
C	Magnetometer MAG3110F	3	10Hz
D	Temp/Baro/Hum. BME280	3	10Hz
E	Accelerometer MPU6500	3	4kHz
F	2.4GHz WiFi (RSSI)	1	10Hz
G	PIR Motion AMN2111	1	10Hz
H	Microphone ADMP401	1	17kHz
I	100mH Inductor (EMI)	1	.5MHz

- Digital Components
- Analog Components

Synthetic sensor can understand what are you doing in your house meters away !!

https://www.youtube.com/watch?v=aqbKrrru2co



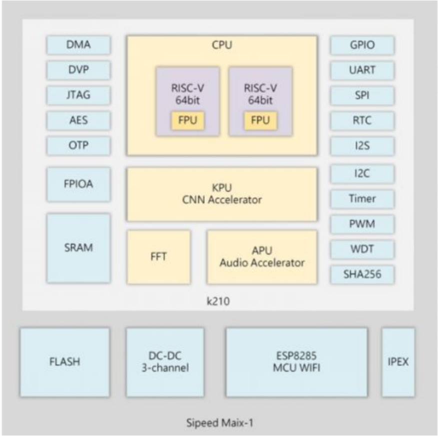
So .. how to include ANN in embedded platforms

Simple microcontrollers
(we will work on that today)

- Very lower power (1 - 100 milliwatt)
- Limited CPU - tens of Megahertz
- Limited RAM - few kilobytes
- Limited FLash - less than 1 mega
- Programmed in Assembly/C/C++
- We need to write NN in C-code
- Extremely cheap - cost less than a 1$

So .. how to include ANN in embedded platforms

Microcontrollers + NN Accelerators

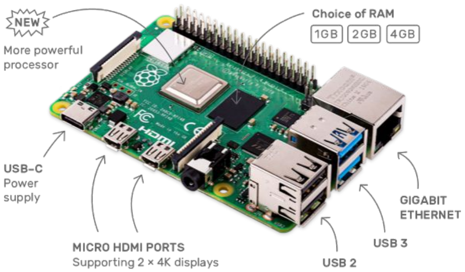


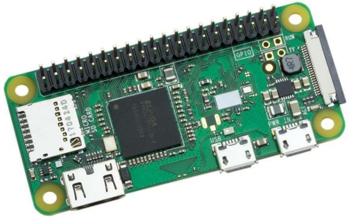
- 50-500 milliwatt power consumption
- Dual core cpu (400 Mhz)
- RAM and Flash can reach multiple Megabytes
- Programmed in C/C++/Assembly
- NN runs on specific accelerator (dedicated cpu designed for NN)
- Can do some serious work like image recognition

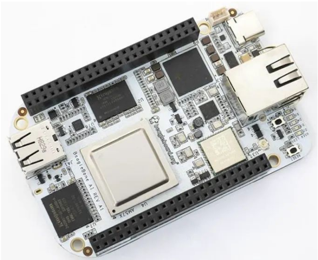
https://www.instructables.com/id/Transfer-Learning-With-Sipeed-MaiX-and-Arduino-IDE/

So .. how to include ANN in embedded platforms

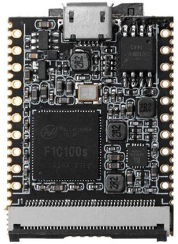
High performance embedded computing (multicore)







- **2 - 20** watt of power consumption
- Multicore CPU(2,4,8,16)
- Multicore GPu(up to 512 core)
- RAM/Flash can reach 32 Gb
- Runs full featured embedded linux
- Can run almost all machine learning platforms directly + almost all Prog.lang
- You can train and deploy ANN directly on the board + Video Accelerators
- Quite Cheap (lechee nano 6$, R-Pi zero 9$)



Available platforms to train models

TensorFlow

scikit learn

plaidML

PYTORCH

theano

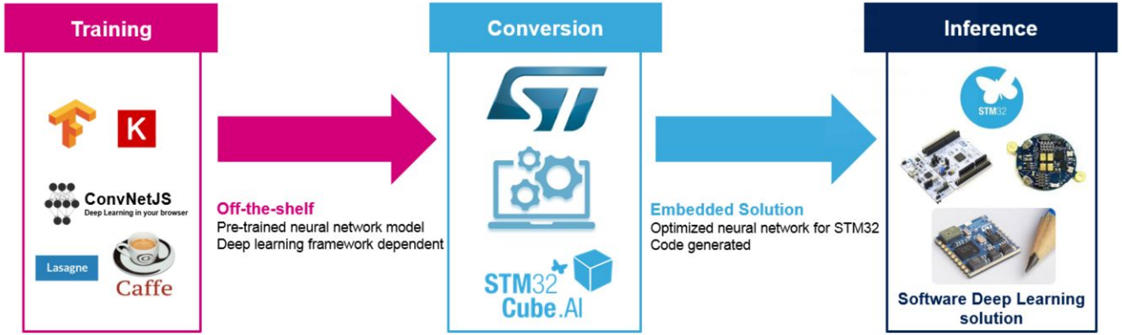
Available platforms to convert trained models

X-Cube AI https://www.st.com/en/embedded-software/x-cube-ai.html

- Powerful and Easy to use (few drag and drop and it's done)
- Generate your projects with all drivers in few seconds
- Generation of an STM32-optimized library from pre-trained Neural Network models
- Supports various Deep Learning frameworks such as Keras, TensorFlow™ Lite, Caffe, ConvNetJs, and Lasagne
- Supports 8-bit quantization
- Free (but not open source)

Getting started Video

https://www.youtube.com/watch?v=grgNXdkmzzQ



Available platforms to convert trained models

frugally-deep https://github.com/Dobiasd/frugally-deep

- **is a small header-only library** written in modern and pure C++. **(require c++14 compiler)**
- very easy to integrate and use.
- Quite fast on one CPU core compared to TensorFlow,
- You can run multiple predictions in parallel, thus utilizing as many CPUs as you like to improve the overall prediction throughput
- Suitable for normal PC and powerful embedded platforms.

Advanced platforms

uTensor (arm Mbed) **https://github.com/uTensor/uTensor**

TensorFlow Lite (Google) **https://www.tensorflow.org/lite**

Example (TensorFlow lite + Arduino nRf52) https://github.com/sandeepmistry/aimldevfest-workshop-2019

Example (uTensor + mBed + k77) https://www.hackster.io/news/simple-neural-network-on-mcus-a7cbd3dc108c

Those are advanced platforms but require specific compiler or can run on specific microcontrollers due to some limitations + need some time to understand how to setup the environment

Available platforms to convert trained models

emLearn https://github.com/emlearn/emlearn

Embedded-friendly Inference

- Portable C99 code
- No libc required
- No dynamic allocations
- Single header file include

This platform generate code can work with **"ANY"** compiler supports C and therefore it supports any microcontroller has enough flash to fit the model

Available platforms to convert trained models

emLearn https://github.com/emlearn/emlearn

Classifiers:

- eml_trees: sklearn.RandomForestClassifier, sklearn.ExtraTreesClassifier, sklearn.DecisionTreeClassifier
- eml_net: sklearn.***MultiLayerPerceptron***, Keras.Sequential with fully-connected layers
- eml_bayes: sklearn.GaussianNaiveBayes

We will use that today

Workflow

Collect Dataset

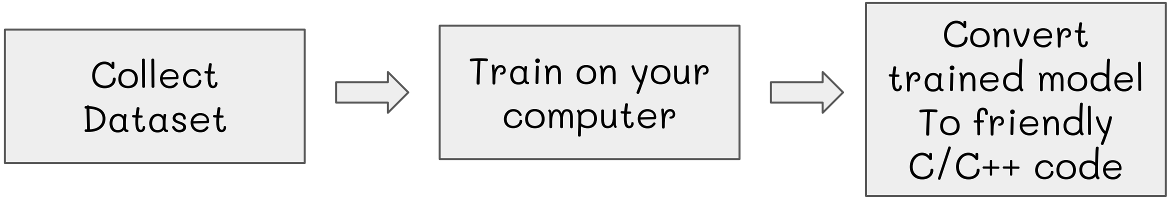
1. First we collect the sensors data to create dataset
2. Label the results
3. Clean the dataset if needed
4. Save to .csv file

Workflow

Collect Dataset → Train on your computer

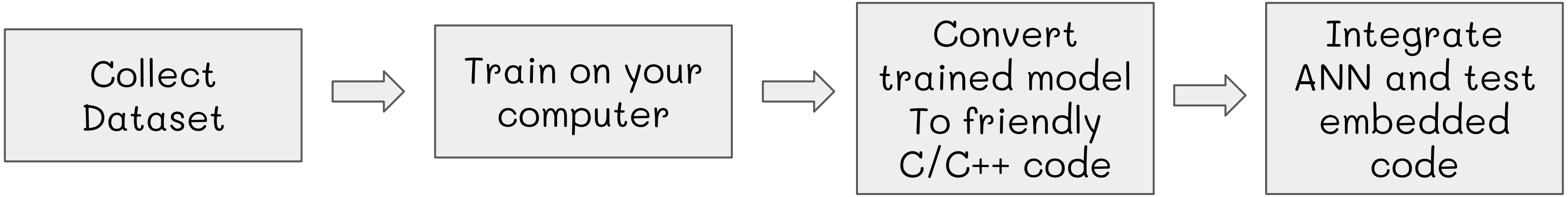
1. Create new python training program
2. Import the dataset
3. Train you model
4. Evaluate the score of trained model

Workflow



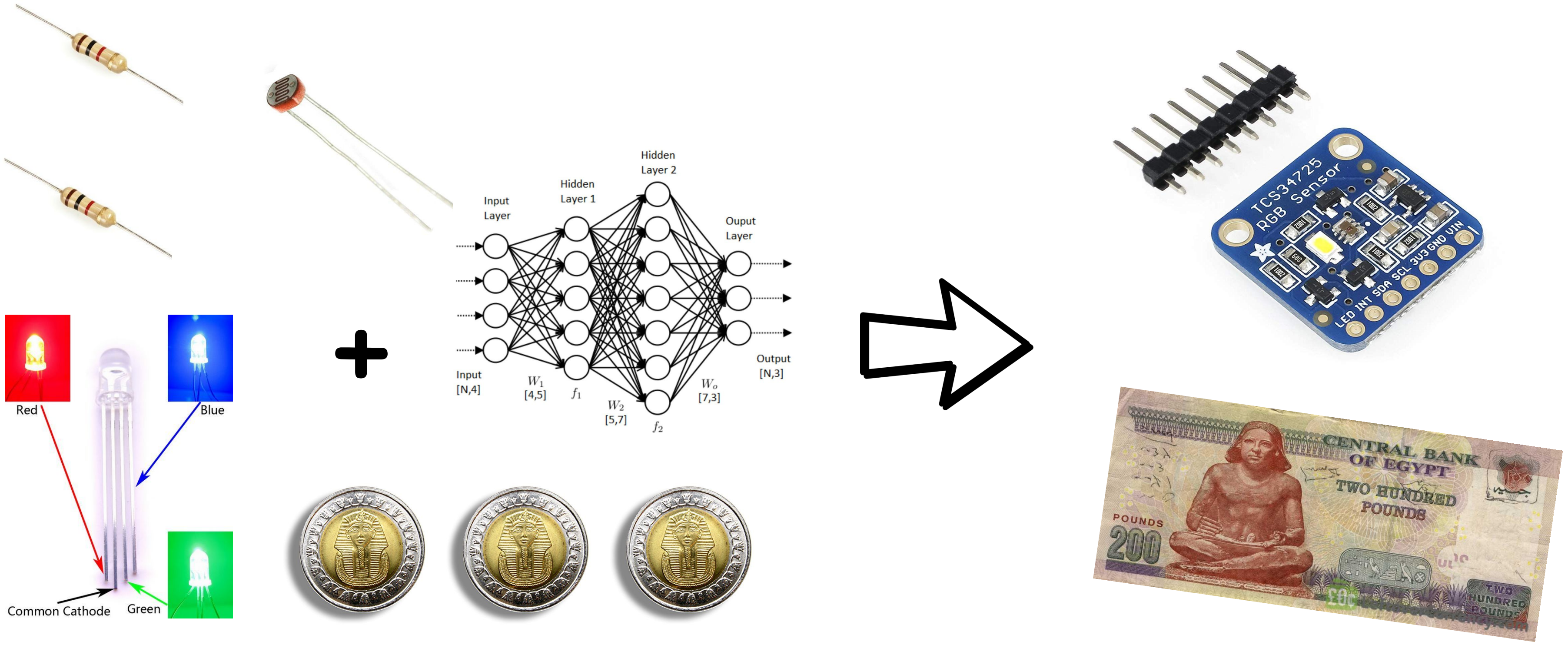
1. Use emlearn to convert the model to single .h file
2. Include the header file to your arduino project

Workflow



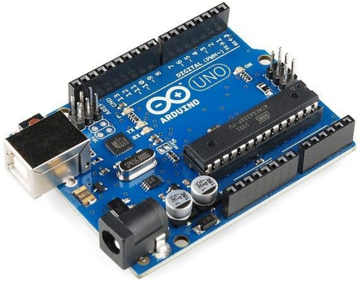
1. Compile the model and make sure it fits your flash
2. If size is ok run the code and evaluate the results
3. You can control size by inc/dec number of hidden layers in the ANN

Example: How to make 3 EGP sensor + MLP to do the job for 200 EGP sensor



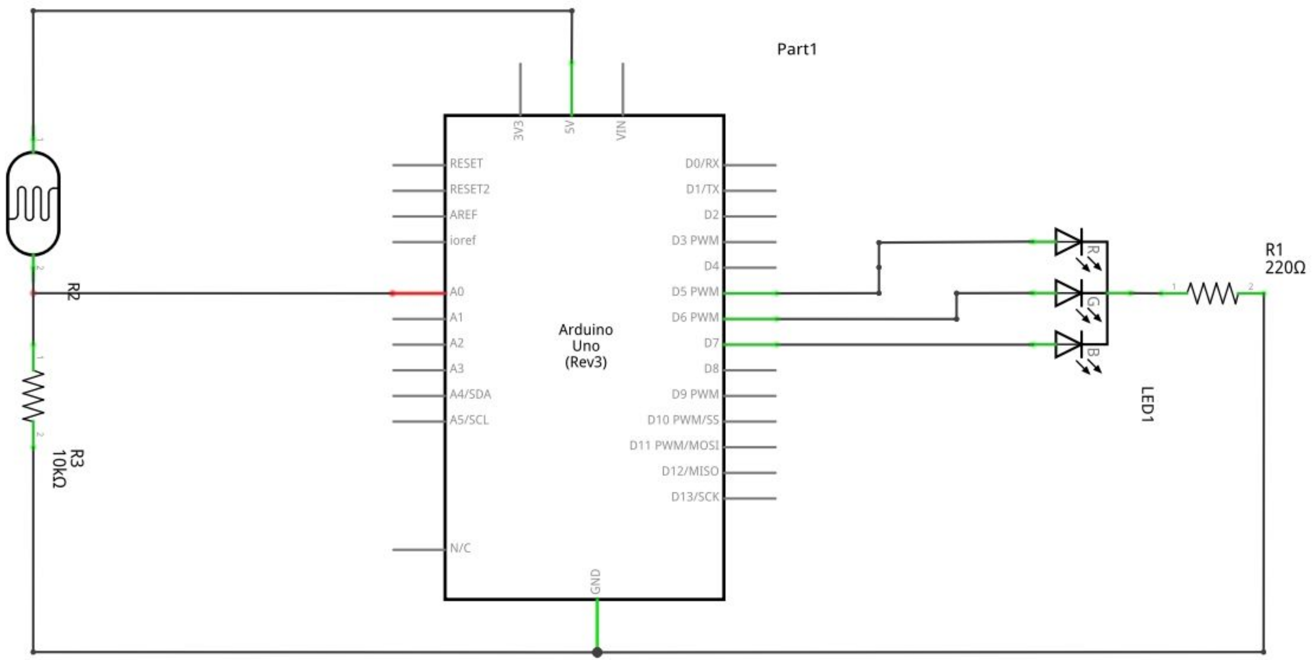
Prepare your environment

- Arduino
- LDR
- RGD led or 3 leds (green-red-blue)
- 1 resistor for leds (330 to 560 ohm)
- 1 resistor for ldr (10 K to 100 K) use one that equal the maximum value of LDR
- jumper wires

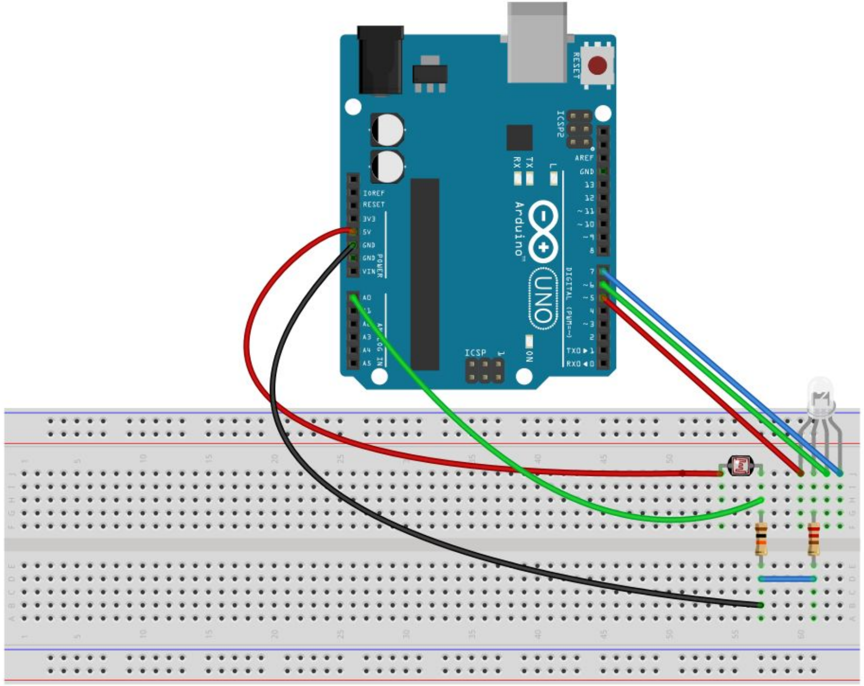


For training: Python + Scikit learn + emlearn
Build it local (better to use linux) or use online Google CoLab
https://colab.research.google.com/

Prepare your environment



Prepare your environment



Lets code ...