

# MỤC LỤC

MỤC LỤC .....	1
BẢNG CÁC TỪ VIẾT TẮT .....	i
DANH SÁCH CÁC HÌNH.....	ii
DANH SÁCH CÁC BẢN .....	iv
CHƯƠNG 1 – KHÁI QUÁT MACHINE LEARNING VÀ MẠNG NEURON NHÂN TẠO .....	1
1.1 KHÁI NIỆM MACHINE LEARNING .....	1
1.2 MẠNG LƯỚI NEURON NHÂN TẠO .....	1
1.2.1 Thuật toán học perceptron .....	1
1.2.2 Thuật toán Perceptron .....	2
1.2.3 Hồi quy Logistic .....	5
1.2.4 Mạng Neuron đa tầng .....	7
1.3 KẾT LUẬN.....	9
CHƯƠNG 2 - THIẾT KẾ MẠCH ĐIỆN GIÁM SÁT CẢM BIẾN IOT.....	10
2.1 SƠ ĐỒ KHỐI CỦA HỆ THỐNG ĐIỀU KHIỂN .....	10
2.1.1 Sơ đồ khối tổng quan hệ thống .....	10
2.1.2 Sơ đồ mạch điện của hộp điện điều khiển .....	11
2.1.3 Các thiết bị điện tử được sử dụng trong mạch điều khiển.....	11
2.2 LẬP TRÌNH VÀ GIAO TIẾP NODEMCU VỚI CẢM BIẾN VÀ SERVER.....	20
2.3 LẬP TRÌNH TẠO SERVER VÀ KẾT NỐI VỚI NODEMCU .....	21
2.3.1 Ngôn ngữ lập trình Server PHP .....	21
2.3.2 Gửi và nhận dữ liệu đối với NodeMCU và Server .....	21
TÀI LIỆU THAM KHẢO .....	25
PHỤ LỤC .....	26

## **BẢNG CÁC TỪ VIẾT TẮT**

<b>PHP</b>	PHP: Hypertext Preprocessor
<b>HTML</b>	Hypertext Markup Language
<b>MySQL</b>	Structured Query Language
<b>CSDL</b>	Cơ sở dữ liệu
<b>FTP</b>	File Transfer Protocol

# DANH SÁCH CÁC HÌNH

## CHƯƠNG 1 – KHÁI QUÁT MACHINE LEARNING VÀ MẠNG NEURON NHÂN TẠO

Hình 1.1 Bài toán phân loại nhị phân trong không gian hai chiều. (a) Cho hai tập dữ liệu được gán nhãn vuông và tròn, hãy xác định nhãn của điểm tam giác. (b) Ví dụ về một ranh giới phẳng phân chia hai tập hợp. Điểm tam giác được phân vào tập các điểm hình tròn ..... 02

Hình 1.2 Minh họa thuật toán Perceptron ..... 04

Hình 1.3 Biểu diễn perceptron và hồi quy tuyến tính dưới dạng mạng neuron. (a) perceptron đầy đủ, (b) perceptron thu gọn, (c) hồi quy tuyến tính thu gọn ..... 04

Hình 1.4 Ví dụ về kết quả thi dựa trên số giờ ôn tập. Trục hoành thể hiện thời gian ôn tập của mỗi sinh viên, trục tung gồm hai giá trị 0/fail (các điểm hình tròn) và 1/pass (các điểm hình vuông) ..... 05

Hình 1.5 Một vài ví dụ về các hàm kích hoạt khác nhau ..... 06

Hình 1.6 Biểu diễn các hàm logic cơ bản sử dụng perceptron ..... 07

Hình 1.7 Ba perceptron biểu diễn hàm XOR ..... 08

Hình 1.8 MLP với hai tầng ẩn (các hệ số điều chỉnh đã được ẩn đi) ..... 08

## CHƯƠNG 2 - THIẾT KẾ MẠCH ĐIỆN GIÁM SÁT CẢM BIẾN IOT

Hình 2.1 Sơ đồ khối tổng quan hệ thống điều khiển ..... 10

Hình 2.2 Sơ đồ mạch điện của hộp điện điều khiển ..... 11

Hình 2.3 Kit RF thu phát WiFi ESP8266 NodeMCU Lua ..... 12

Hình 2.4 Sơ đồ chân – pinout cho NodeMCU ..... 13

Hình 2.5 Cảm biến độ đục của chất lỏng ..... 14

Hình 2.6 Sơ đồ mạch điện cảm biến độ đục của chất lỏng ..... 16

Hình 2.7 Đồ thị dạng parabol của mối liên hệ giữa độ đục và điện áp (y là độ đục, x là điện áp) ..... 16

Hình 2.8 Cảm biến nhiệt độ DS18B20 ..... 17

Hình 2.9 CHình 2.9 Cảm biến độ ẩm đất ..... 19

Hình 2.10 Sơ đồ mô tả hoạt động liên kết giữa các Server sử dụng Database Server và FTP Server .....	21
Hình 2.11 Mô hình giao tiếp giữa NodeMCU với Server .....	22
Hình 2.12 Truy cập đường dẫn url để GET / POST giá trị nhiệt độ và độ ẩm gửi lên từ NodeMCU đến Server .....	22
Hình 2.13 Các giá trị thông số như nhiệt độ, độ ẩm và độ ẩm đất được cập nhật trong bảng này .....	23
Hình 2.14 Giá trị đo của ba loại cảm biến thả nổi .....	23
Hình 2.15 Giá trị đo của ba loại cảm biến trong môi trường .....	24

## **DANH SÁCH CÁC BẢN**

Bảng 1.1 Thời gian ôn thi và kết quả thi của 20 sinh viên .....	05
Bảng 2.1: Sơ đồ chân của cảm biến độ đục chất lỏng .....	15
Bảng 2.2: Sơ đồ chân của cảm biến độ ẩm đất .....	20

# CHƯƠNG 1 – KHÁI QUÁT MACHINE LEARNING VÀ MẠNG NEURON NHÂN TẠO

## 1.1 KHÁI NIỆM MACHINE LEARNING

Máy Học (machine learning, ML) là một tập con của trí tuệ nhân tạo. Machine learning là một lĩnh vực nhỏ trong khoa học máy tính, có khả năng tự học hỏi dựa trên dữ liệu được đưa vào mà không cần phải được lập trình cụ thể (Machine Learning is the subfield of computer science, that “gives computers the ability to learn without being explicitly programmed” – Wikipedia).

Những năm gần đây, sự phát triển của các hệ thống tính toán cùng lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn đã giúp machine learning tiến thêm một bước dài. Một lĩnh vực mới được ra đời được gọi là học sâu (deep learning, DL). Deep learning đã giúp máy tính thực thi những việc vào mười năm trước tưởng chừng là không thể: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết, giao tiếp với con người, chuyển đổi ngôn ngữ, hay thậm chí cả sáng tác văn thơ và âm nhạc.

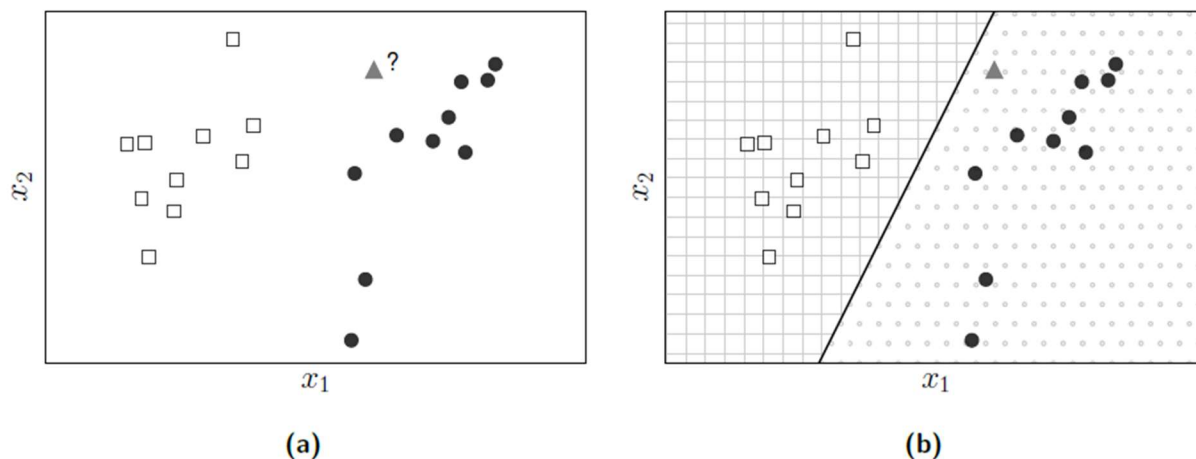
## 1.2 MẠNG LƯỚI NEURON NHÂN TẠO

Trong machine learning nói riêng và toán tối ưu nói chung, chúng ta thường xuyên phải tìm các cực tiểu toàn cục của một hàm số. Nếu chỉ xét riêng các hàm khả vi, việc giải phương trình đạo hàm bằng không có thể phức tạp hoặc có vô số nghiệm. Thay vào đó, người ta thường tìm các điểm cực tiểu địa phương, và coi đó là một nghiệm cần tìm của bài toán trong những trường hợp nhất định. Các điểm cực tiểu địa phương là nghiệm của phương trình đạo hàm bằng không (ta vẫn đang giả sử rằng các hàm này liên tục và khả vi). Nếu tìm được toàn bộ (hữu hạn) các điểm cực tiểu địa phương, ta chỉ cần thay từng điểm đó vào hàm số để suy ra điểm cực tiểu toàn cục. Tuy nhiên, trong hầu hết các trường hợp, việc giải phương trình đạo hàm bằng không là bất khả thi. Nguyên nhân có thể đến từ sự phức tạp của đạo hàm, từ việc các điểm dữ liệu có số chiều lớn hoặc từ việc có quá nhiều điểm dữ liệu. Thực tế cho thấy, trong nhiều bài toán machine learning, các điểm cực tiểu địa phương thường cho kết quả tốt, đặc biệt là trong các mạng neuron nhân tạo. Do đó, trước tiên còn phổ biến để giải quyết các bài toán tối ưu là dùng một phép.

### 1.2.1 Thuật toán học perceptron

Nếu coi mỗi vector đặc trưng là một điểm trong không gian nhiều chiều, bài toán phân loại có thể được coi như bài toán xác định nhân của từng điểm trong không gian. Nếu coi mỗi nhãn chiếm một hoặc vài vùng trong không gian, ta cần đi tìm ranh giới giữa các vùng đó. Ranh giới đơn giản nhất trong không gian hai chiều là một đường thẳng, trong không gian ba chiều là một mặt phẳng, trong không gian nhiều chiều là một siêu phẳng. Những ranh giới phẳng này đơn giản vì chúng có thể được biểu diễn bởi một hàm số

tuyến tính. Hình 1.1b minh họa một đường thẳng phân chia hai tập dữ liệu trong không gian hai chiều. Trong trường hợp này, điểm dữ liệu mới hình tam giác rơi vào cùng tập hợp với các điểm hình tròn.



Hình 1.1 Bài toán phân loại nhị phân trong không gian hai chiều. (a) Cho hai tập dữ liệu được gán nhãn vuông và tròn, hãy xác định nhãn của điểm tam giác. (b) Ví dụ về một ranh giới phẳng phân chia hai tập hợp. Điểm tam giác được phân vào tập các điểm hình tròn.

### 1.2.2 Thuật toán Perceptron

Thuật toán Perceptron được khái quát thành các bước như sau:

- Tại thời điểm  $t = 0$ , chọn ngẫu nhiên một vector trọng số  $\mathbf{w}_0$ .
- Tại thời điểm  $t$ , nếu không có điểm dữ liệu nào bị phân loại lỗi, dừng thuật toán.
- Giả sử  $\mathbf{x}_i$  là một điểm bị phân loại lỗi, cập nhật

$$\mathbf{w}_{t+1} = \mathbf{w}_t + y_i \mathbf{x}_i$$

- Thay đổi  $t = t + 1$  rồi quay lại Bước 2.

Ví dụ minh họa thuật toán Perceptron trên Python như sau:

### *Quy tắc phân loại*

Giả sử đã tìm được vector trọng số  $w$ , nhãn của các điểm dữ liệu  $X$  được xác định bằng hàm `predict(w, X)`:

```
import numpy as np
def predict(w, X):
    """
    predict label of each row of X, given w
    X: a 2-d numpy array of shape (N, d), each row is a datapoint
    w: a 1-d numpy array of shape (d)
    """
    return np.sign(X.dot(w))
```

### *Thuật toán tối ưu hàm mất mát*

Hàm `perceptron(X, y, w_init)` thực hiện thuật toán PLA với tập huấn luyện  $X$ , nhãn  $y$  và nghiệm ban đầu  $w_{init}$ .

```
def perceptron(X, y, w_init):
    """ perform perceptron learning algorithm
    X: a 2-d numpy array of shape (N, d), each row is a datapoint
    y: a 1-d numpy array of shape (N), label of each row of X. y[i] = 1/-1
    w_init: a 1-d numpy array of shape (d)
    """
    w = w_init
    while True:
        pred = predict(w, X)
        # find indexes of misclassified points
        mis_idx = np.where(np.equal(pred, y) == False)[0]
        # number of misclassified points
        num_mis = mis_idx.shape[0]
        if num_mis == 0: # no more misclassified points
            return w
        # randomly pick one misclassified point
        random_id = np.random.choice(mis_idx, 1)[0]
        # update w
        w = w + y[random_id]*X[random_id]
    return w
```

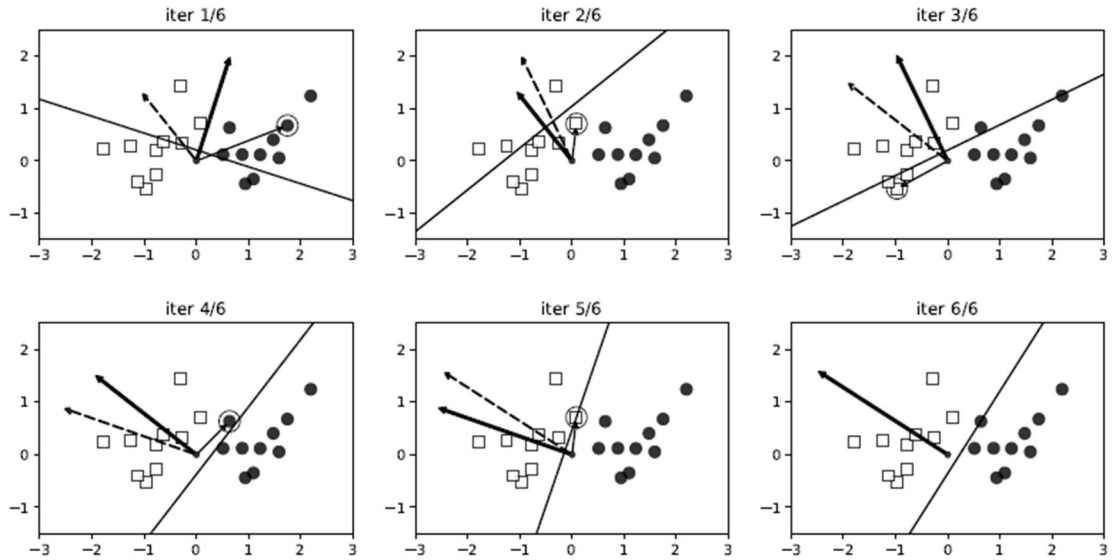


Áp dụng thuật toán vừa viết vào dữ liệu trong không gian hai chiều:

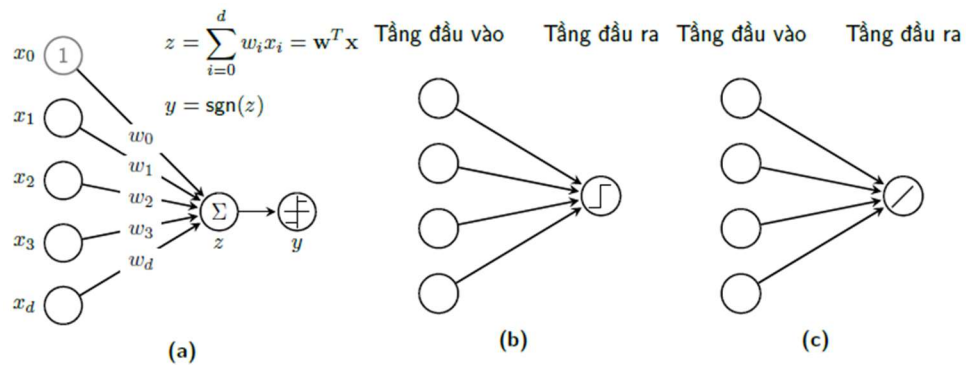
```
means = [[-1, 0], [1, 0]]
cov = [[.3, .2], [.2, .3]]
N = 10
X0 = np.random.multivariate_normal(means[0], cov, N)
X1 = np.random.multivariate_normal(means[1], cov, N)

X = np.concatenate((X0, X1), axis = 0)
y = np.concatenate((np.ones(N), -1*np.ones(N)))

Xbar = np.concatenate((np.ones((2*N, 1)), X), axis = 1)
w_init = np.random.randn(Xbar.shape[1])
w = perceptron(Xbar, y, w_init)
```



Hình 1.2 Minh họa thuật toán Perceptron



Hình 1.3 Biểu diễn perceptron và hồi quy tuyến tính dưới dạng mạng neuron. (a) perceptron đầy đủ, (b) perceptron thu gọn, (c) hồi quy tuyến tính thu gọn.

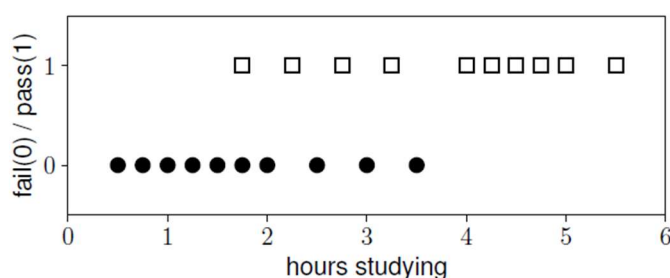
Mô hình perceptron ở trên khá giống với một thành phần nhỏ của mạng thần kinh sinh học như Hình 1.3. Dữ liệu từ nhiều dây thần kinh đầu vào đi về một nhân tế bào. Nhân tế bào tổng hợp thông tin và đưa ra quyết định ở tín hiệu đầu ra. Trong mạng neuron nhận tạo của perceptron, mỗi giá trị xi đóng vai trò một tín hiệu đầu vào, hàm tính tổng và hàm kích hoạt có chức năng tương tự nhân tế bào. Tên gọi mạng neuron nhân tạo được khởi nguồn từ đây.

### 1.2.3 Hồi quy Logistic

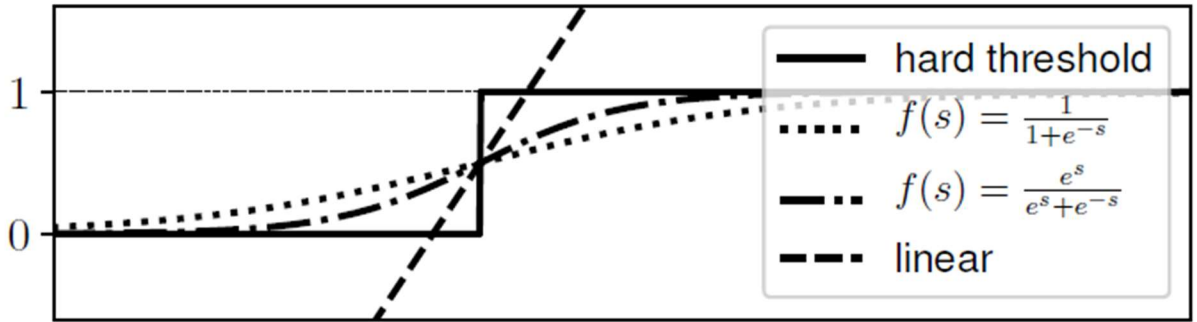
Trong phần này, chúng ta sẽ thảo luận một mô hình tuyến tính với một hàm kích hoạt khác, thường được áp dụng cho các bài toán phân loại nhị phân. Trong mô hình này, đầu ra có thể được biểu diễn dưới dạng xác suất. Ví dụ, xác suất thi đỗ nếu biết thời gian ôn thi, xác suất ngày mai có mưa dựa trên những thông tin đo được trong ngày hôm nay,... Mô hình này có tên là hồi quy logistic. Mặc dù trong tên có chứa từ hồi quy, phương pháp này thường được sử dụng nhiều hơn cho các bài toán phân loại. Xem xét một ví dụ như bảng sau:

Bảng 1.1 Thời gian ôn thi và kết quả thi của 20 sinh viên

Số giờ	Đậu?	Số giờ	Đậu?	Số giờ	Đậu?	Số giờ	Đậu?
0.5	0	0.75	0	1	0	1.25	0
1.5	0	1.75	0	1.75	1	2	0
2.25	1	2.5	0	2.75	1	4	0
3.25	1	3.5	0	4	1	4.25	1
4.5	1	4.75	1	5	1	5.5	1



Hình 1.4 Ví dụ về kết quả thi dựa trên số giờ ôn tập. Trục hoành thể hiện thời gian ôn tập của mỗi sinh viên, trục tung gồm hai giá trị 0/fail (các điểm hình tròn) và 1/pass (các điểm hình vuông).



Hình 1.5 Một vài ví dụ về các hàm kích hoạt khác nhau.

Xét một ví dụ về quan hệ giữa thời gian ôn thi và kết quả của 20 sinh viên trong Bảng 1.4. Bài toán đặt ra là từ dữ liệu này hãy xây dựng mô hình đánh giá khả năng đỗ của một sinh viên dựa trên thời gian ôn tập. Dữ liệu trong Bảng 1.4 được mô tả trên Hình 1.4. Nhìn chung, thời gian học càng nhiều thì khả năng đỗ càng cao. Tuy nhiên, không có một ngưỡng thời gian học nào giúp phân biệt rạch ròi việc đỗ/trượt. Nói cách khác, dữ liệu của hai tập này là không tách biệt tuyến tính, và vì vậy PLA sẽ không hữu ích. Tuy nhiên, thay vì dự đoán chính xác hai giá trị đỗ/trượt, ta có thể dự đoán xác suất để một sinh viên thi đỗ dựa trên thời gian ôn thi.

### Hàm sigmoid và tanh

Trong các hàm số có ba tính chất nói trên, hàm *sigmoid*:

$$f(s) = \frac{1}{1 + e^{-s}} \triangleq \sigma(s)$$

được sử dụng nhiều nhất, vì nó bị chặn trong khoảng  $(0, 1)$  và:

$$\lim_{s \rightarrow -\infty} \sigma(s) = 0; \quad \lim_{s \rightarrow +\infty} \sigma(s) = 1.$$

Thú vị hơn:

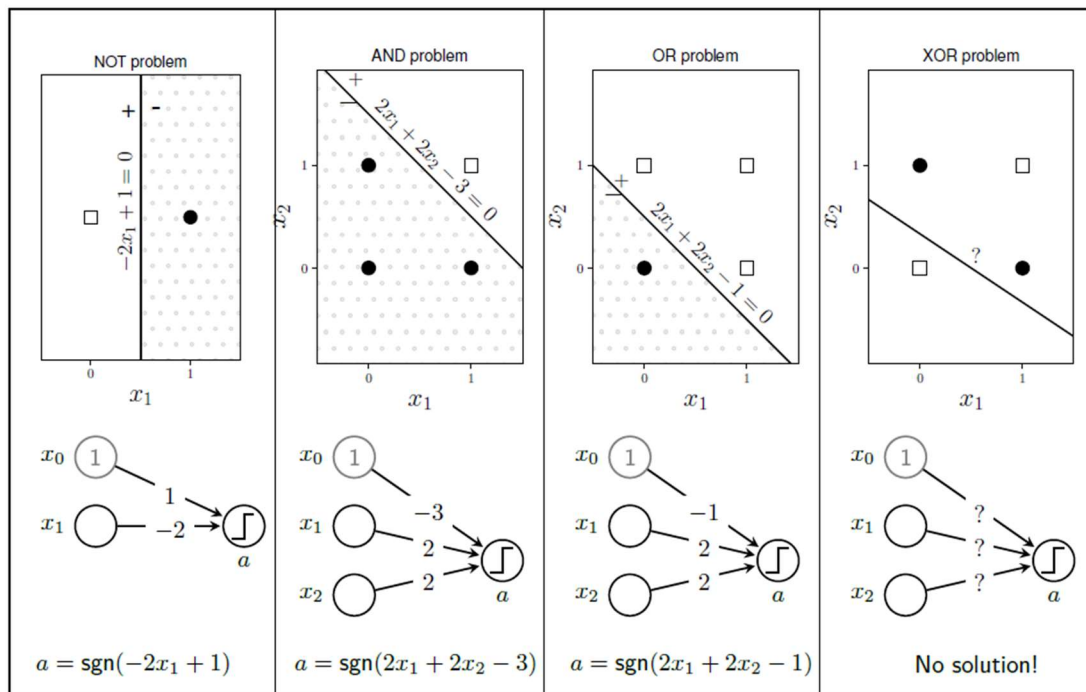
$$\sigma'(s) = \frac{e^{-s}}{(1 + e^{-s})^2} = \frac{1}{1 + e^{-s}} \frac{e^{-s}}{1 + e^{-s}} = \sigma(s)(1 - \sigma(s))$$

Hàm sigmoid có thể được thực hiện trên Python như sau:

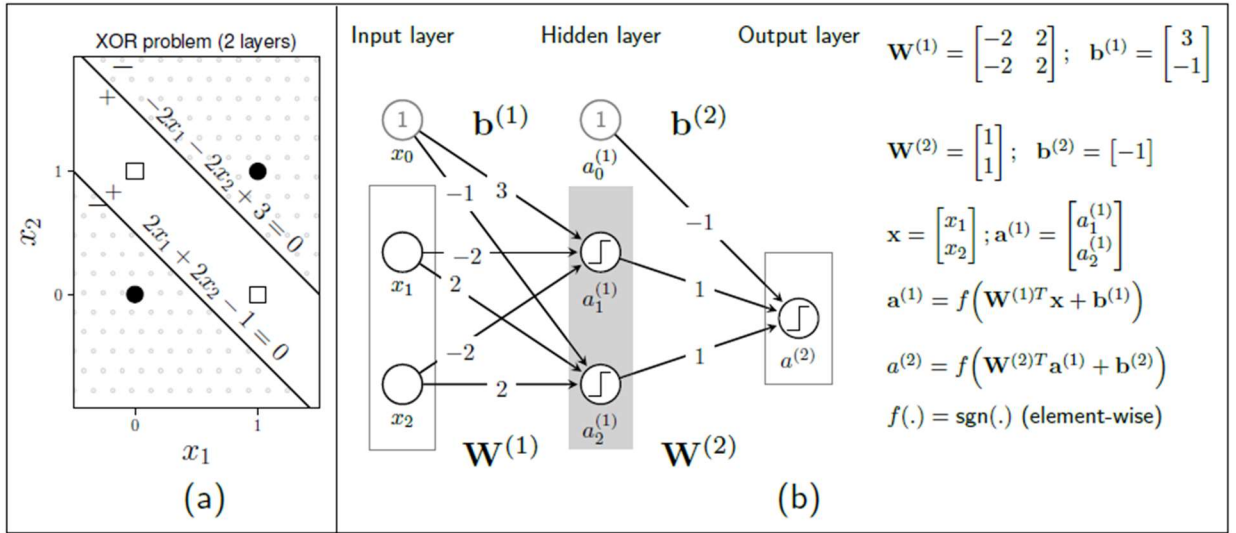
```
def sigmoid(S):
    """
    S: an numpy array
    return sigmoid function of each element of S
    """
    return 1/(1 + np.exp(-S))
```

### 1.2.4 Mạng Neuron đa tầng

Đối với hàm XOR, vì dữ liệu không tách biệt tuyến tính nên không thể biểu diễn bằng một perceptron. Nếu thay perceptron bằng hồi quy logistic ta cũng không tìm được các hệ số thỏa mãn, vì về bản chất, hồi quy logistic hay cả hồi quy softmax chỉ tạo ra các ranh giới tuyến tính. Như vậy, các mô hình mạng neuron đã biết không thể biểu diễn được hàm số logic đơn giản này. Đối với hàm XOR, vì dữ liệu không tách biệt tuyến tính nên không thể biểu diễn bằng một perceptron. Nếu thay perceptron bằng hồi quy logistic ta cũng không tìm được các hệ số thỏa mãn, vì về bản chất, hồi quy logistic hay cả hồi quy softmax chỉ tạo ra các ranh giới tuyến tính. Như vậy, các mô hình mạng neuron đã biết không thể biểu diễn được hàm số logic đơn giản này.

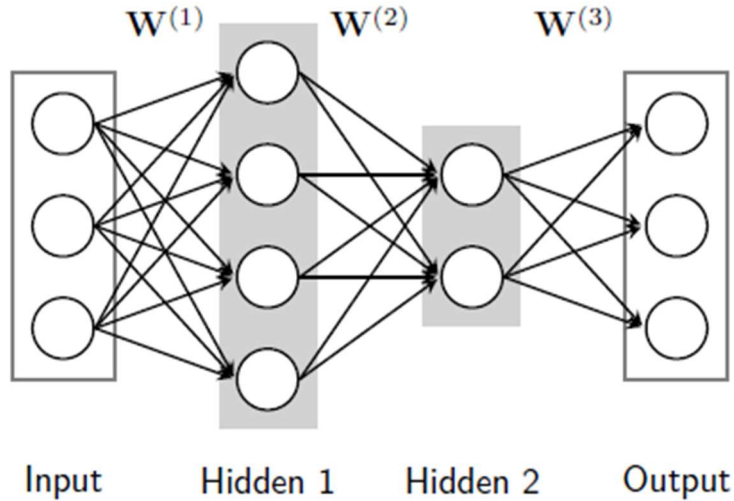


Hình 1.6 Biểu diễn các hàm logic cơ bản sử dụng perceptron.



Hình 1.7 Ba perceptron biểu diễn hàm XOR

Nhận thấy rằng nếu cho phép sử dụng hai đường thẳng, bài toán biểu diễn hàm XOR có thể được giải quyết như Hình 1.6 Các hệ số tương ứng với hai đường thẳng trong Hình 1.7a được minh họa trên Hình 1.7b. Một mạng neuron với nhiều hơn hai tầng còn được gọi là mạng neuron đa tầng (multi-layer neural network) hoặc perceptron đa tầng (multilayer perceptron MLP). Tên gọi perceptron ở đây có thể gây nhầm lẫn vì cụm từ này để chỉ mạng neuron nhiều tầng và mỗi tầng không nhất thiết là một hoặc nhiều perceptron. Thực chất, perceptron rất hiếm khi được sử dụng trong các mạng neuron đa tầng. Hàm kích hoạt thường là các hàm phi tuyến khác thay vì hàm sgn.



Hình 1.8 MLP với hai tầng ẩn (các hệ số điều chỉnh đã được ẩn đi).

Đầu ra của mạng neuron đa tầng ở dạng này ứng với một đầu vào  $x$  có thể được tính theo:

$$\begin{aligned} \mathbf{a}^{(0)} &= \mathbf{x} \\ \mathbf{z}^{(l)} &= \mathbf{W}^{(l)T} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}, \quad l = 1, 2, \dots, L \\ \mathbf{a}^{(l)} &= f^{(l)}(\mathbf{z}^{(l)}), \quad l = 1, 2, \dots, L \\ \hat{\mathbf{y}} &= \mathbf{a}^{(L)} \end{aligned}$$

Vector  $\hat{\mathbf{y}}$  chính là đầu ra dự đoán. Bước này được gọi là lan truyền thuận (feed forward) vì cách tính toán được thực hiện từ đầu đến cuối của mạng. Hàm mất mát đạt giá trị nhỏ khi đầu ra dự đoán gần với đầu ra thực sự. Tùy vào bài toán, phân loại hoặc hồi quy, chúng ta cần thiết kế các hàm mất mát phù hợp.

### 1.3 KẾT LUẬN

Lĩnh vực ML nói chung và DL nói riêng là cực kỳ lớn và có nhiều nhánh nhỏ. Phạm vi một cuốn sách chắc chắn không thể bao quát hết mọi vấn đề và đi sâu vào từng nhánh cụ thể. Do vậy, cuốn sách này chỉ nhằm cung cấp cho bạn đọc những khái niệm, kỹ thuật chung và các thuật toán cơ bản nhất của ML. Từ đó, bạn đọc có thể tự tìm thêm các cuốn sách và khóa học liên quan nếu muốn đi sâu vào từng vấn đề.

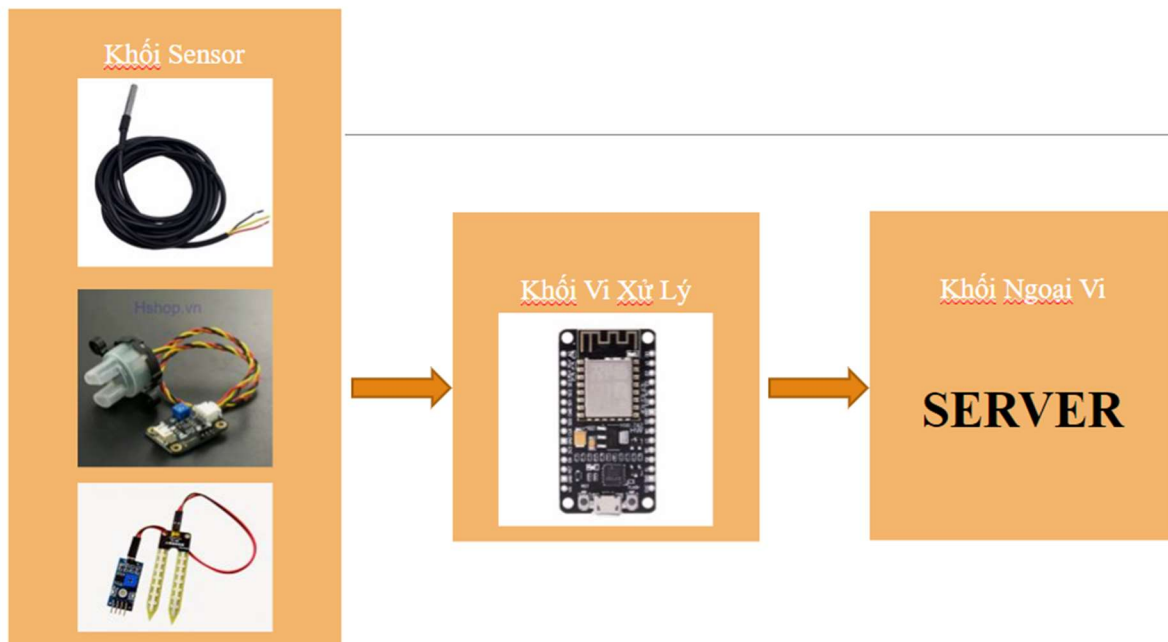
Hãy nhớ rằng luôn bắt đầu từ những điều đơn giản. Khi bắt tay vào giải quyết một bài toán ML hay bất cứ bài toán nào, chúng ta nên bắt đầu từ những thuật toán đơn giản. Không phải chỉ có những thuật toán phức tạp mới có thể giải quyết được vấn đề. Những thuật toán phức tạp thường có yêu cầu cao về khả năng tính toán và đôi khi nhạy cảm với cách chọn tham số. Ngược lại, những thuật toán đơn giản giúp chúng ta nhanh chóng có một bộ khung cho mỗi bài toán. Kết quả của các thuật toán đơn giản cũng mang lại cái nhìn sơ bộ về sự phức tạp của mỗi bài toán. Việc cải thiện kết quả sẽ được thực hiện dần ở các bước sau. Cuốn sách này sẽ trang bị cho bạn đọc những kiến thức khái quát và một số hướng tiếp cận cơ bản cho các bài toán ML. Để tạo ra các sản phẩm thực tiễn, chúng ta cần học hỏi và thực hành thêm nhiều.

## CHƯƠNG 2 - THIẾT KẾ MẠCH ĐIỆN GIÁM SÁT CẢM BIẾN IOT

### 2.1 SƠ ĐỒ KHỐI CỦA HỆ THỐNG ĐIỀU KHIỂN

#### 2.1.1 Sơ đồ khối tổng quan hệ thống

Sơ đồ khối tổng quan của hệ thống điều khiển như hình 2.1.

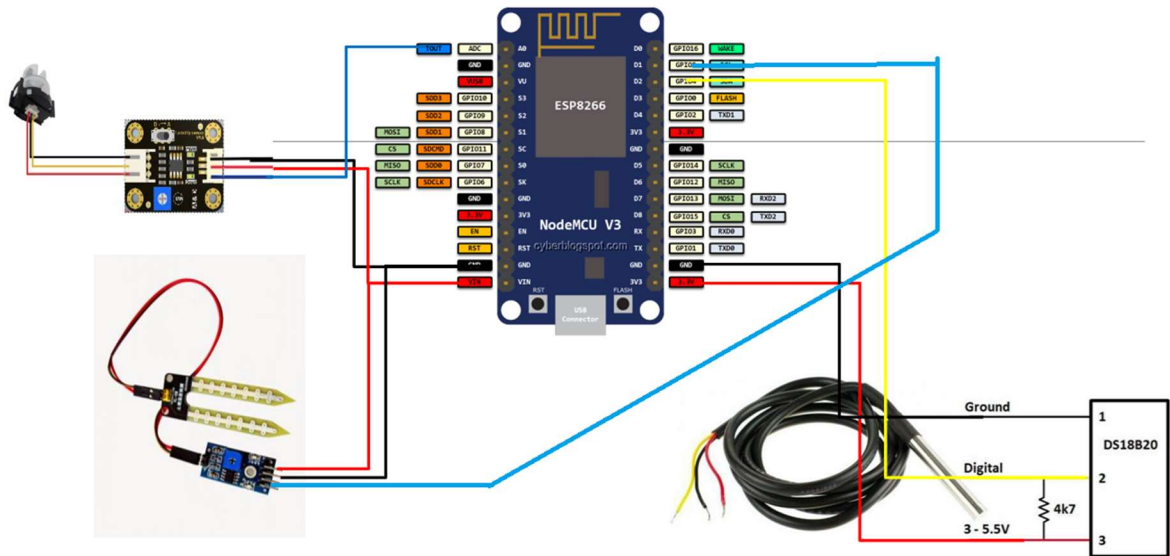


Hình 2.1 Sơ đồ khối tổng quan hệ thống điều khiển

Trong sơ đồ khối hệ thống điều khiển, thành phần quan trọng làm trung tâm cho bộ điều khiển là Module NodeMCU đóng vai trò làm khối xử lý. Khối xử lý có nhiệm vụ nhận dữ liệu từ Khối sensor (cảm biến nhiệt độ, cảm biến độ ẩm). Khối ngoại vi gồm Server, với Server thì được viết bằng ngôn ngữ PHP và hệ quản trị CSDL MySQL.



### 2.1.2 Sơ đồ mạch điện của hộp điện điều khiển



Hình 2.2 Sơ đồ mạch điện của hộp điện điều khiển

### 2.1.3 Các thiết bị điện tử được sử dụng trong mạch điều khiển

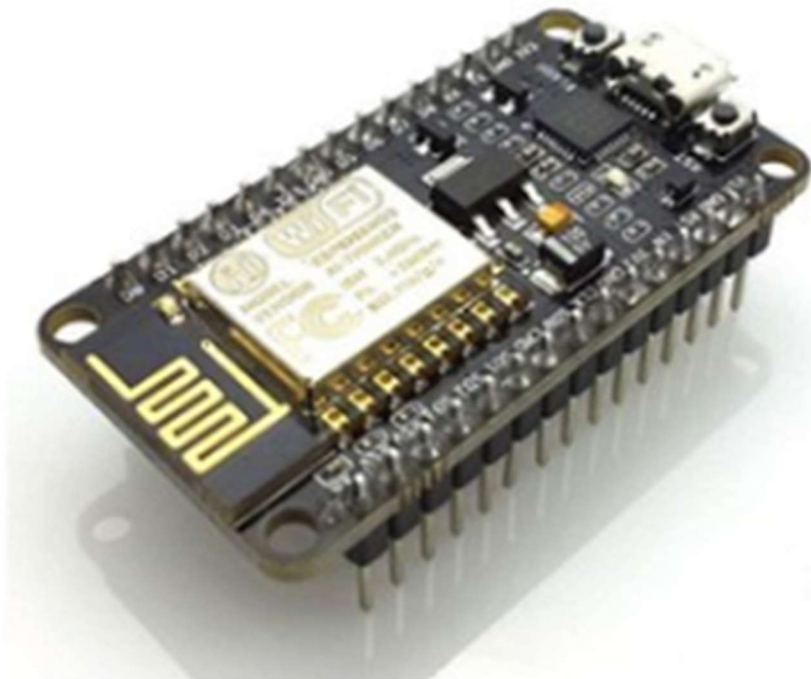
#### 2.1.3.1 Kit RF Thu Phát Wifi ESP8266 NodeMCU Lua CP2102

Kit RF thu phát Wifi ESP8266 NodeMCU Lua là lựa chọn tối ưu cho hệ thống, sử dụng để cập nhật các thông số của cảm biến nhiệt độ, độ ẩm và độ ẩm đất, sau đó cập nhật các giá trị thông số cảm biến này lên Server.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng WiFi, đặc biệt là các ứng dụng liên quan đến IoT. Sử dụng chip nạp và giao tiếp UART mới và ổn định nhất là CP2102 có khả năng tự nhận Driver trên tất cả các hệ điều hành Window và Linux, đây là phiên bản nâng cấp từ các phiên bản sử dụng IC nạp giá rẻ CH340.

Kit RF thu phát Wifi ESP8266 được sử dụng cho tủ điện điều khiển như hình 2.3.



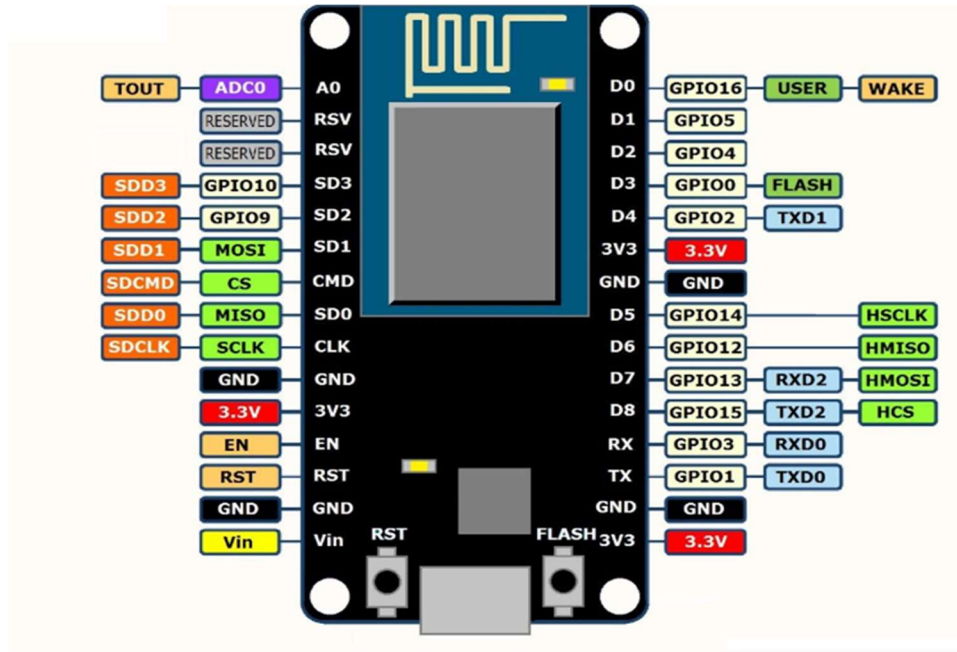


Hình 2.3 Kit RF thu phát WiFi ESP8266 NodeMCU Lua

Thông số kỹ thuật cho NodeMCU:

- IC chính: ESP8266 WiFi SoC.
- Phiên bản firmware: NodeMCU Lua.
- Chip nạp và giao tiếp UART: CP2102.
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GPIO giao tiếp mức 3.3VDC.
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 25 x 50 mm.

Sơ đồ chân của NodeMCU ESP8266 được thể hiện như hình 2.4.



Hình 2.4 Sơ đồ chân – pinout cho NodeMCU

NodeMCU ESP8266 có một chân Analog giúp thực hiện việc đọc giữ liệu từ cảm biến độ đục, các chân GPIO từ D0 đến D8 dùng để kích các thiết bị, sử dụng đọc cảm biến nhiệt độ DS18B20 và cảm biến độ ẩm đất.

#### 2.1.3.2 Cảm biến độ đục chất lỏng

Cảm biến độ đục của nước trong bể cá được thể hiện như hình 2.5.



Hình 2.5 Cảm biến độ đục của chất lỏng.

Độ đục là một trong những tiêu chí dùng để kiểm tra chất lượng nước, nó thể hiện bằng lượng hạt tồn tại trong nước, lượng hạt càng tăng thì mức độ đục của nước càng tăng. Vì thế kiểm tra độ đục của nước là cần thiết. Cảm biến đo độ đục chất lỏng giúp chúng ta có thể đo mức độ đục của chất lỏng. Cảm biến đo độ đục chất lỏng hoạt động dựa vào nguyên lý quang học, nó có thể phát hiện các hạt lơ lửng trong nước. Các ứng dụng và thông số kỹ thuật của cảm biến độ đục như sau:

- Ứng dụng của cảm biến đo độ đục chất lỏng được ứng dụng trong bài toán như:

- + Đo chất lượng nước ở sông, suối, hay trong các đường ống nước.
- + Đo chất lượng nước thải.
- + Dùng trong nghiên cứu, thí nghiệm.

- Thông số kỹ thuật:

- + Điện áp hoạt động: 5V.

- + Dòng điện làm việc: 40mA (Max).
- + Thời gian đáp ứng: < 500ms.
- + Điện trở cách điện: 100M(Min).
- + Đầu ra Analog 0 ~ 4.5V.
- + Đầu ra Digital: High/Low (Có thể điều chỉnh giá trị ngưỡng bằng biến trở).
- + Kích thước: 38mm\*28mm\*10mm.
- + Trọng lượng: 30g.

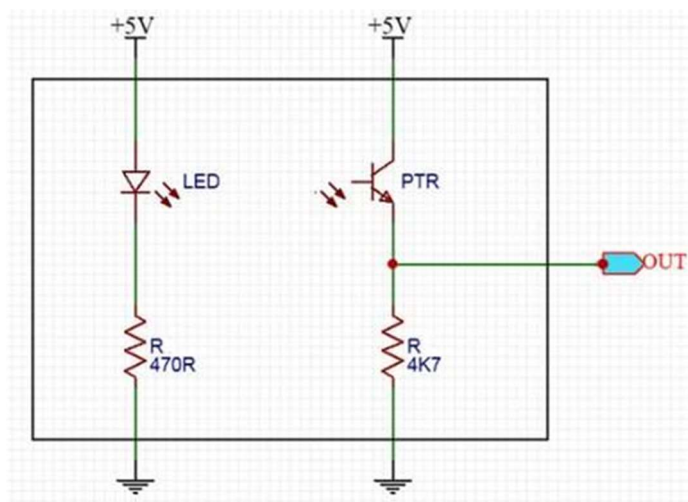
Sơ đồ chân của cảm biến độ đục chất lỏng được trình bày như bảng 2.1.

Bảng 2.1: Sơ đồ chân của cảm biến độ đục chất lỏng.

Thứ tự	Tên	Chức năng
1	D / A	Chân dữ liệu đầu ra.
2	+	VCC: 5V
3	-	GND

- Chân dữ liệu đầu ra "D" có thể chọn là chân ra kiểu "Digital" hoặc "Analog" thông qua công tắc gạt trên cảm biến.

Sơ đồ mạch điện cảm biến độ đục trong đề tài như hình 2.6.

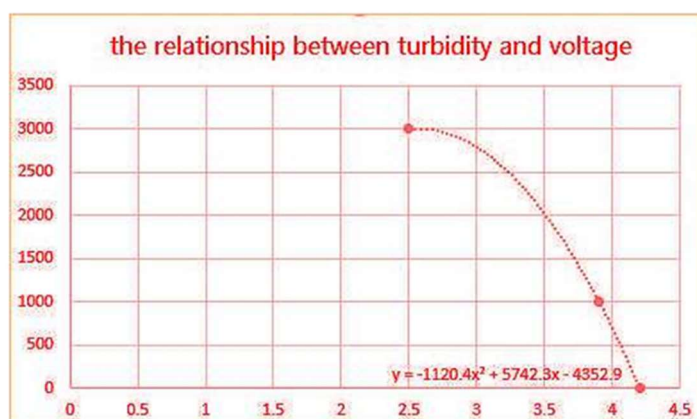


Hình 2.6 Sơ đồ mạch điện cảm biến độ đục của chất lỏng

Cảm biến độ đục chất lỏng bao gồm một mắt phát (LED) và một mắt thu (PTR), khi có hạt thông qua giữa 2 mắt sẽ làm thay đổi giá trị dòng điện cực nền của PTR làm thay đổi dòng điện ngõ ra – OUT nhờ mạch DFRobot.

Cách thiết lập giao tiếp giữa MCU và cảm biến độ đục:

Đồ thị dạng parapol của mối liên hệ giữa độ đục chất lỏng và điện áp Analog như hình 2.7.



Hình 2.7 Đồ thị dạng parapol của mối liên hệ giữa độ đục và điện áp (y là độ đục, x là điện áp)

Kết nối chân dữ liệu ra dạng Analog vào chân A0 NodeMCU đo điện áp hoạt động, sau đó tính toán dựa trên công thức độ đục ( $\text{mg} / \text{L}$ ) và điện áp (Volt).

#### 2.1.3.3 Cảm biến nhiệt độ

Cảm biến nhiệt độ DS18B20 có vỏ bọc bảo vệ như hình 2.8.



Hình 2.8 Cảm biến nhiệt độ DS18B20

Đây là phiên bản chống nước, ẩm của cảm biến nhiệt độ DS18B20.

DS18B20 là cảm biến (loại digital) đo nhiệt độ mới của hãng MAXIM với độ phân giải cao (12bit). IC sử dụng giao tiếp 1 dây rất gọn gàng, dễ lập trình. IC còn có chức năng cảnh báo nhiệt độ khi vượt ngưỡng và đặc biệt hơn là **có thể cấp nguồn từ chân data (parasite power)**.

Cảm biến có thể hoạt động ở 125 độ C nhưng có cáp bọc PVC nên giữ nó dưới 100 độ C. Dây cảm biến kỹ thuật số, nên không bị suy hao tín hiệu đường dây dài.

*Ứng dụng:* Kiểm soát nhiệt môi trường, đo nhiệt độ bên trong các tòa nhà, thiết bị, máy móc, và trong hệ thống giám sát.

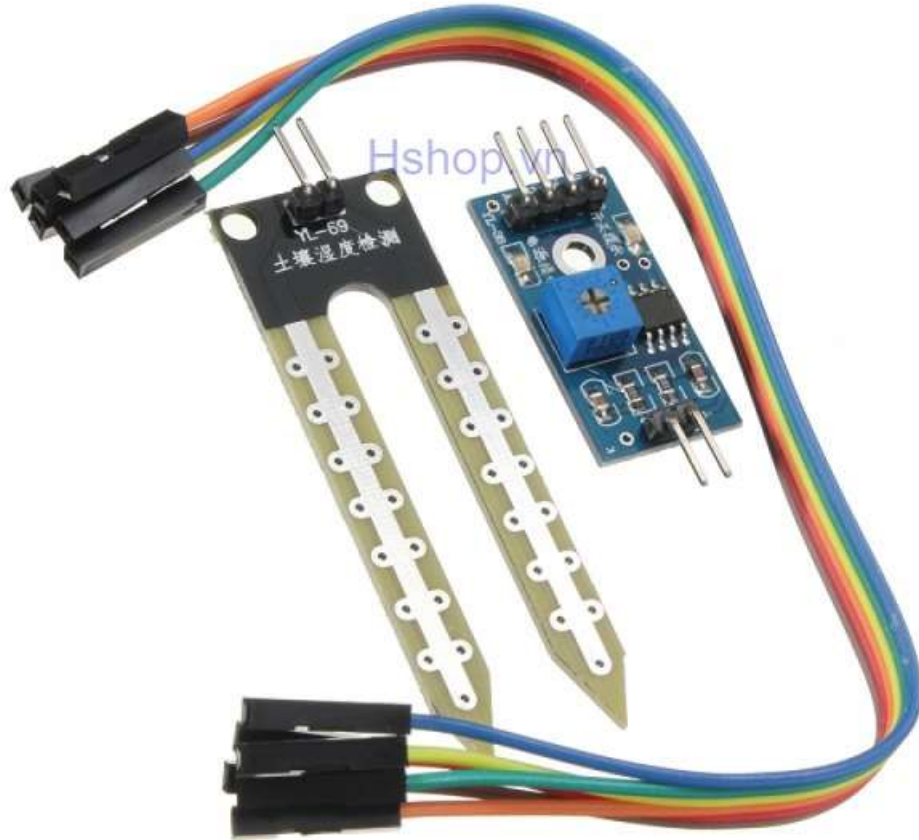
**Để đo được nhiệt độ ta cần thêm 1 điện trở 4.7k ohm nối từ chân dq lên Vcc.**

Thông số kỹ thuật:

- Nguồn: 3 - 5.5V.
- Dải đo nhiệt độ: -55 - 125 độ C (-67 - 257 độ F).
- Sai số:  $\pm 0.5$  độ C khi đo ở dải -10 - 85 độ C.
- Độ phân giải: người dùng có thể chọn từ 9 - 12 bits.
- Chuẩn giao tiếp: 1-Wire (1 dây).
- Có cảnh báo nhiệt khi vượt ngưỡng cho phép và cấp nguồn từ chân data.
- Thời gian chuyển đổi nhiệt độ tối đa: 750ms (khi chọn độ phân giải 12bit).
- Mỗi IC có một mã riêng (lưu trên EEPROM của IC) nên có thể giao tiếp nhiều DS18B20 trên cùng 1 dây.
- Ống thép không gỉ (chống ẩm, nước) đường kính 6mm, dài 50mm.
- Đường kính đầu dò: 6mm.
- Chiều dài dây: 1m.
- Đầu ra: VCC (Red), DATA (yellow), GND (Black).

#### *2.1.3.4 Cảm biến độ ẩm đất*

Cảm biến độ ẩm đất được thể hiện như hình 2.9.



Hình 2.9 Cảm biến độ ẩm đất

*Ứng dụng:* Cảm biến độ ẩm đất Soil Moisture Sensor thường được sử dụng trong các mô hình tưới nước tự động, vườn thông minh,..., cảm biến giúp xác định độ ẩm của đất qua đầu dò và trả về giá trị Analog, Digital qua 2 chân tương ứng để giao tiếp với Vi điều khiển để thực hiện vô số các ứng dụng khác nhau.

Thông số kỹ thuật:

- Điện áp hoạt động: 3.3~5VDC
- Tín hiệu đầu ra:
- Analog: theo điện áp cấp nguồn tương ứng.



- Digital: High hoặc Low, có thể điều chỉnh độ ẩm mong muốn bằng biến trở thông qua mạch so sánh LM393 tích hợp.
- Kích thước: 3 x 1.6cm.

Bảng 2.2: Sơ đồ chân của cảm biến độ ẩm đất

VCC	3.3V ~ 5V
GND	GND của nguồn ngoài
DO	Đầu ra tín hiệu số (mức cao hoặc mức thấp)
AO	Đầu ra tín hiệu tương tự (Analog)

## 2.2 LẬP TRÌNH VÀ GIAO TIẾP NODEMCU VỚI CẢM BIẾN VÀ SERVER

- Bước 1: kết nối với mạng WiFi để thực hiện truy vấn với Server, thiết lập chân cho thiết bị (input cho các cảm biến), các hàm sau nằm trong hàm thiết lập void setup():

+ Kết nối WiFi:

```
WiFi.mode(WIFI_STA);
```

```
WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED);
```

+ Khởi tạo cảm biến nhiệt độ DS18B20 bằng thư viện *DallasTemperature.h*:

```
DS18B20.begin();
```

- Bước 3: sau đó đọc giá trị nhiệt độ, độ đục và độ ẩm của đất để gửi lên Server.

+ Đọc giá trị nhiệt độ bằng hàm requestTemperatures() và gán giá trị đọc được như sau:

```
DS18B20.requestTemperatures();
```

```
tempC = DS18B20.getTempCByIndex(0);
```

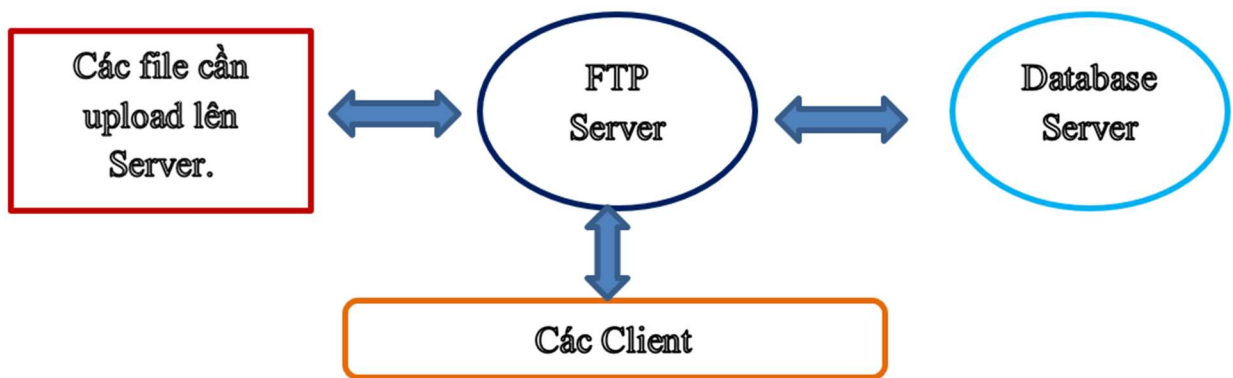
+ Đọc giá trị độ đục bằng hàm bằng sử dụng chế độ ADC ở chân A0 như sau:

```
volt += ((float)analogRead(sensorPin)/1023)*5;
```

### 2.3 LẬP TRÌNH TẠO SERVER VÀ KẾT NỐI VỚI NODEMCU

Trong quy mô báo cáo này, sẽ sử dụng một *Máy chủ cơ sở dữ liệu – Database Server* để thực hiện lưu trữ dữ liệu được client gửi lên hoặc được truy vấn ra từ Cơ Sở Dữ Liệu (CSDL). Cũng sử dụng máy chủ *FTP Server* để thực hiện upload các file truy vấn CSDL lên một vùng lưu trữ của Host quy định sẵn.

Sơ đồ mô tả hoạt động liên kết giữa giữa FTP Server và ngoại vi như hình 2.10.



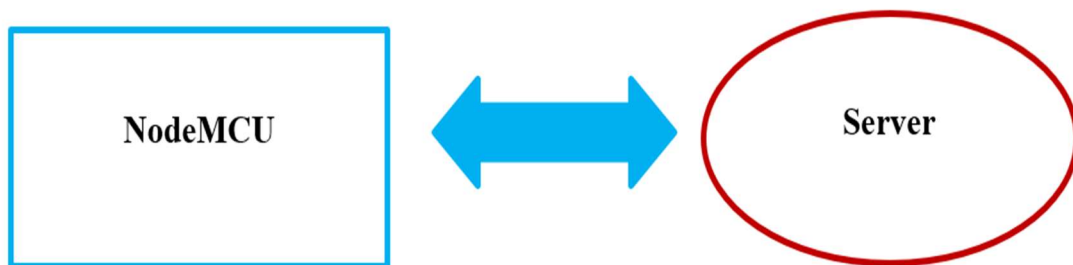
Hình 2.10 Sơ đồ mô tả hoạt động liên kết giữa các Server sử dụng Database Server và FTP Server

#### 2.3.1 Ngôn ngữ lập trình Server PHP

PHP - viết tắt hồi quy của "Hypertext Preprocessor", là một ngôn ngữ lập trình kịch bản được chạy ở phía server nhằm sinh ra mã html trên client. PHP với cách viết mã rõ ràng, tốc độ nhanh, dễ học nên PHP đã trở thành một ngôn ngữ lập trình web rất phổ biến và được ưa chuộng.

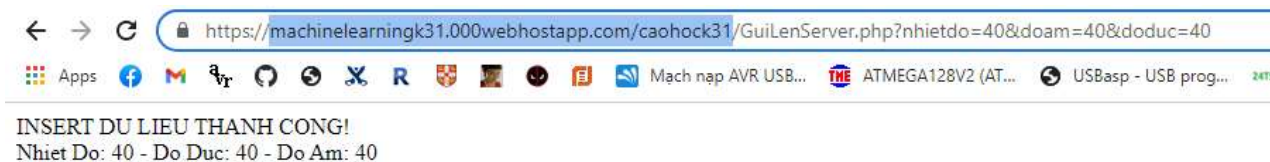
#### 2.3.2 Gửi và nhận dữ liệu đối với NodeMCU và Server

Sơ đồ giao tiếp giữa Server và NodeMCU được mô tả như hình 2.11.



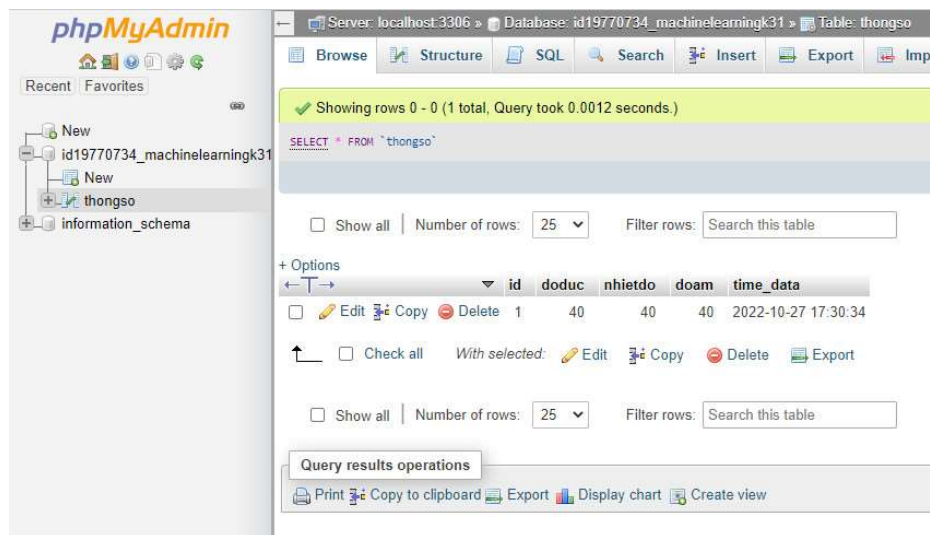
Hình 2.11 Mô hình giao tiếp giữa NodeMCU với Server

Trong quá trình gửi và nhận dữ liệu từ NodeMCU đến Server như hình 4.14, thực hiện truy cập vào đường địa chỉ url của file cần truy vấn với Server. Trong quá trình gửi dữ liệu lên Server, NodeMCU truy cập đường dẫn url của file có tên *GuiLenServer.php* để truy vấn gửi dữ liệu vào Server. Trong file này, khi truy cập gửi các giá trị nhiệt độ và độ đục của nước, file có sử dụng đến lệnh GET / POST để nhận dữ liệu gửi lên đường dẫn, đường dẫn này được mô tả như hình 2.12 sau:



Hình 2.12 Truy cập đường dẫn url để GET / POST giá trị nhiệt độ và độ đục gửi lên từ NodeMCU đến Server.

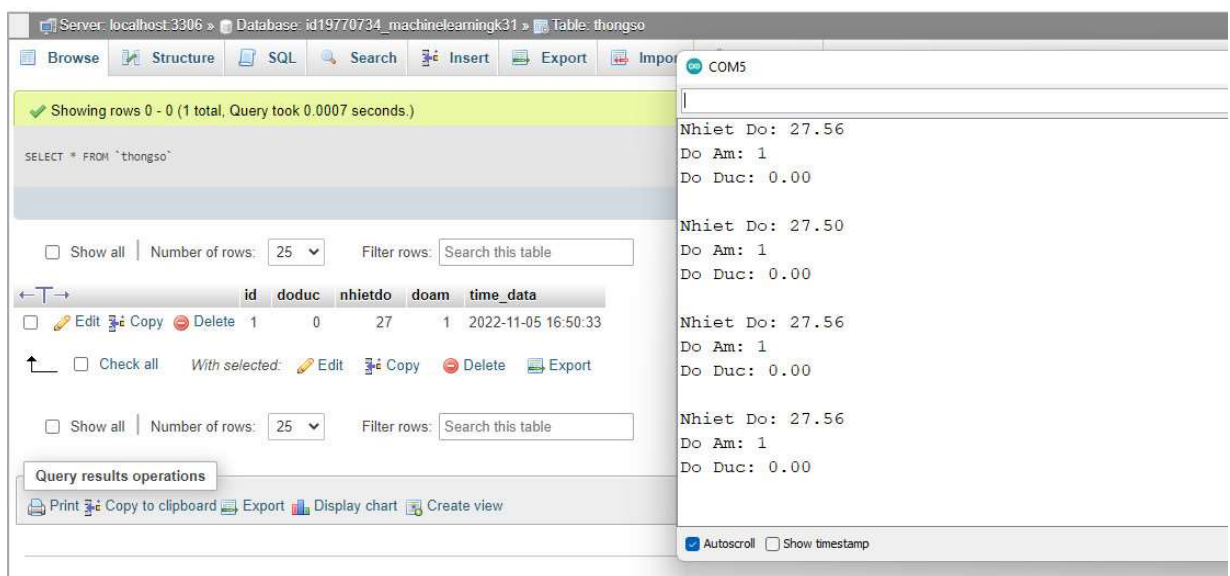
- Bảng thông số chứa các giá trị nhiệt độ, độ đục và độ ẩm đất được gửi lên từ NodeMCU như hình 2.13:



Hình 2.13 Các giá trị thông số như nhiệt độ, độ đục và độ ẩm đất được cập nhật trong bảng này.

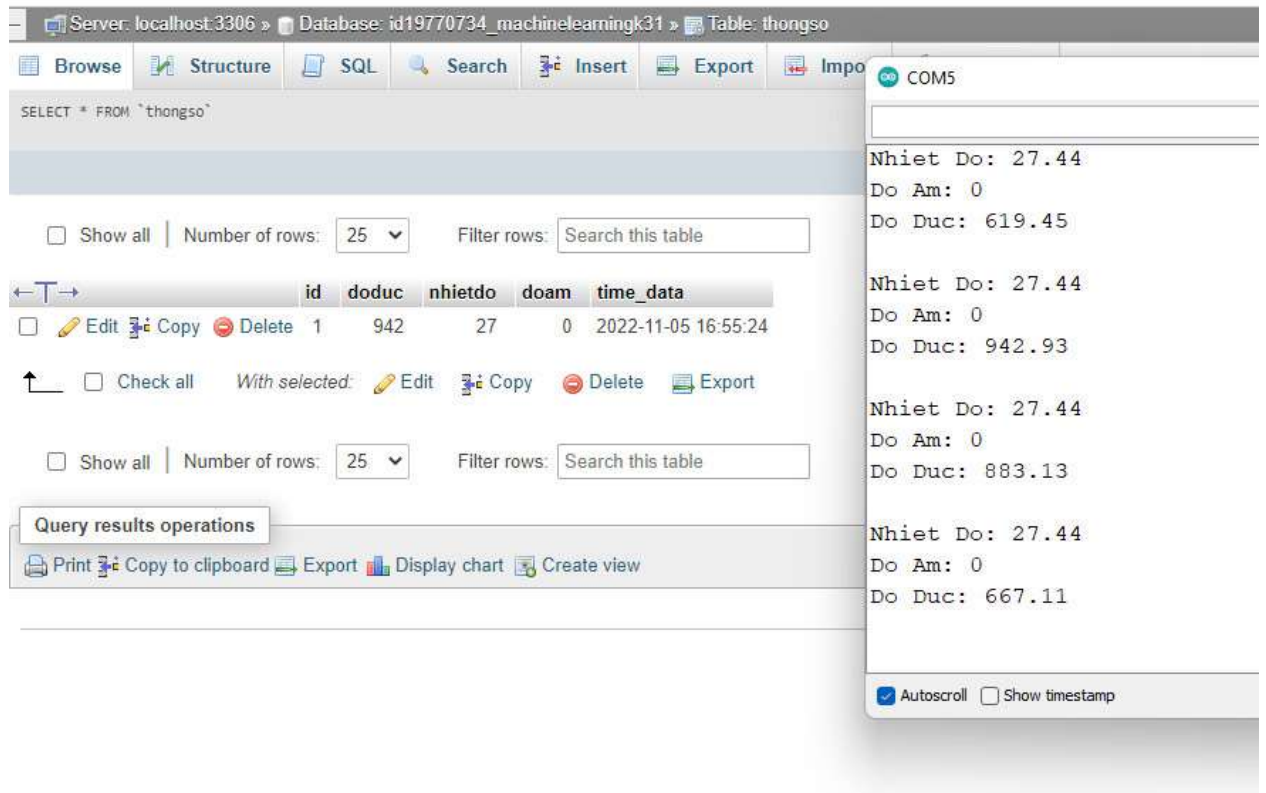
## 2.4 THỰC NGHIỆM

Trong thực nghiệm, ta đo thông số của ba loại cảm biến là nhiệt độ, độ đục của nước và độ ẩm của đất. Thông số trước khi biến đổi của ba cảm biến được đo và cập nhật lên server được trình bày như hình 2.14.



Hình 2.14 Giá trị đo của ba loại cảm biến thả nổi.

Và sau khi được nhúng vào môi trường nhất định (nước đục và đất ẩm) thì giá trị cảm biến đo được là:



Server: localhost:3306 » Database: id19770734\_machinelearningk31 » Table: thongso

SELECT \* FROM `thongso`

Number of rows: 25 Filter rows: Search this table

	id	doduc	nhietdo	doam	time_data
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	942	27	0	2022-11-05 16:55:24

Number of rows: 25 Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

COM5

Nhiet Do: 27.44  
Do Am: 0  
Do Duc: 619.45

Nhiet Do: 27.44  
Do Am: 0  
Do Duc: 942.93

Nhiet Do: 27.44  
Do Am: 0  
Do Duc: 883.13

Nhiet Do: 27.44  
Do Am: 0  
Do Duc: 667.11

☒ Autoscroll ☐ Show timestamp

Hình 2.15 Giá trị đo của ba loại cảm biến trong môi trường.

## **TÀI LIỆU THAM KHẢO**

- [1] Vũ Hữu Tiếp (2022), Machine Learning Cơ Bản, 02/01/2022.
- [2] [https://vdodata.vn/ung-dung-may-chu-ftp-ftp-2-server/#Cach\\_truyen\\_tai\\_du\\_lieu\\_qua\\_giao\\_thuc\\_FTP\\_vasu\\_dung\\_FTP\\_client](https://vdodata.vn/ung-dung-may-chu-ftp-ftp-2-server/#Cach_truyen_tai_du_lieu_qua_giao_thuc_FTP_vasu_dung_FTP_client)
- [3] <https://nhanhoa.com/tin-tuc/tim-hieu-chi-tiet-cac-loai-server.html>
- [4] <https://iotmaker.vn/nodemcu.html>
- [5] <https://www.electroschematics.com/14077/diy-water-quality-meter-using-a-turbidity-sensor/>

## **PHỤ LỤC**

Phần Code Node MCU và Server được cập nhật tại:

<https://github.com/huystrauss/machinelearningk31>