Contents

[**I. GIỚI THIỆU** 1](#_Toc7162808)

[**II. NỘI DUNG** 1](#_Toc7162809)

[**1.** **Thu thập dữ liệu và dự báo một số yếu tố thời tiết sử dụng máy học** 1](#_Toc7162810)

[**1.1. Machine learning (Máy học)** 1](#_Toc7162811)

[1.2. Lý thuyết về Linear Regression (Hồi quy Tuyến tính) 3](#_Toc7162812)

[1.3. Tensorflow 5](#_Toc7162813)

[1.4. Triển khai thuật toán Linear Regression bằng Tensorflow 9](#_Toc7162814)

[1.5. Tính toán dự trên dự báo và đưa ra khuyến nghị 18](#_Toc7162815)

[**2.** **Thiết kế và thực hiện phần cứng** 22](#_Toc7162816)

[**2.1.** **Các bộ phận chính** 22](#_Toc7162817)

[**2.2.** **Sơ đồ khối hệ thống** 32](#_Toc7162818)

[**2.3.** **Lập trình cho phần cứng** 34](#_Toc7162819)

[**2.4.** **Mạch thực tế** 34](#_Toc7162820)

[**3.** **Hiển thị số liệu đo đạc, số liệu dự báo và điều khiển thiết bị qua Website** 34](#_Toc7162821)

[**4.** **Lập trình Android Webservice dùng để điều khiển thiết bị** 41](#_Toc7162822)

[III. KẾT LUẬN 49](#_Toc7162823)

**I. GIỚI THIỆU**

**II. NỘI DUNG**

1. **Thu thập dữ liệu và dự báo một số yếu tố thời tiết sử dụng máy học**

**1.1. Machine learning (Máy học)**

Những năm gần đây, AI - Artificial Intelligence (Trí Tuệ Nhân Tạo), và cụ thể hơn là Machine Learning (Học Máy hoặc Máy Học) nổi lên như một bằng chứng của cuộc cách mạng công nghiệp lần thứ tư (1 - động cơ hơi nước, 2 - năng lượng điện, 3 - công nghệ thông tin). Trí Tuệ Nhân Tạo đang len lỏi vào mọi lĩnh vực trong đời sống mà có thể chúng ta không nhận ra. Xe tự hành của Google và Tesla, hệ thống tự tag khuôn mặt trong ảnh của Facebook, trợ lý ảo Siri của Apple, hệ thống gợi ý sản phẩm của Amazon, hệ thống gợi ý phim của Netflix, máy chơi cờ vây AlphaGo của Google DeepMind, … chỉ là một vài trong vô vàn những ứng dụng của AI/Machine Learning.

Machine Learning là một tập con của AI. Theo định nghĩa của Wikipedia: “Machine learning is the subfield of computer science that “gives computers the ability to learn without being explicitly programmed”.” Nói đơn giản, Machine Learning là một lĩnh vực nhỏ của Khoa Học Máy Tính, nó có khả năng tự học hỏi dựa trên dữ liệu đưa vào mà không cần phải được lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning (Học Sâu - thực sự tôi không muốn dịch từ này ra tiếng Việt). Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: 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 của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc.

(Nguồn: [What’s the Difference Between Artificial Intelligence, Machine Learning, and Deep Learning?](https://blogs.nvidia.com/blog/2016/07/29/whats-difference-artificial-intelligence-machine-learning-deep-learning-ai/))

Hình 1. Mối quan hệ giữa AI, Machine Learning và Deep Learning.

### 1.2. Lý thuyết về Linear Regression (Hồi quy Tuyến tính)

#### 1.2.1. Giới thiệu

Ý tưởng chính của Linear Regression (Hồi quy Tuyến tính) là sử dụng chuỗi các thông tin. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào. Mạng được gọi là hồi quy (Regression) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, mạng có khả năng nhớ các thông tin được tính toán trước đó.

Hàm dự đoán của ta có dạng: với Ở đây  là một vector hàng chứa thông tin input, y là một số vô hướng (scalar) biểu diễn output.

Một hàm số đơn giản nhất có thể mô tả mối quan hệ giữa đầu ra và 3 đại lượng đầu vào là:



 (1)  
trong đó là các hằng số. Mối quan hệ  bên trên là một mối quan hệ tuyến tính (Linear). Bài toán ta đang làm là một bài toán thuộc loại hồi quy (Regression). Bài toán đi tìm các hệ số tối ưu chính vì vậy được gọi là bài toán Linear Regression.

**Chú ý 1:**  là giá trị thực của output (dựa trên số liệu thống kê ta có trong tập training data), trong khi  là giá trị mà mô hình Linear Regression dự đoán được. Nhìn chung,  và  là hai giá trị khác nhau do có sai số mô hình, tuy nhiên, ta mong muốn rằng sự khác nhau này rất nhỏ.

**Chú ý 2:** Linear hay tuyến tính hiểu một cách đơn giản là thẳng, phẳng. Trong không gian hai chiều, một hàm số được gọi là tuyến tính nếu đồ thị của nó có dạng một đường thẳng. Trong không gian ba chiều, một hàm số được goi là tuyến tính nếu đồ thị của nó có dạng một mặt phẳng. Các hàm số tuyến tính là các hàm đơn giản nhất, vì chúng thuận tiện trong việc hình dung và tính toán.

#### 1.2.2. Phân tích toán học

**a) Dạng của Linear Regression**

Trong phương trình (1) phía trên, nếu ta đặt là vector (cột) hệ số cần phải tối ưu và  là vector (hàng) dữ liệu đầu vào mở rộng. Số 1 ở đầu được thêm vào để phép tính đơn giản hơn và thuận tiện cho việc tính toán. Khi đó, phương trình (1) có thể được viết lại dưới dạng:



**b) Sai số dự đoán**

Ta mong muốn rằng sự sai khác  giữa giá trị thực  và giá trị dự đoán  là nhỏ nhất. Nói cách khác, mục đích là làm cho giá trị sau đây càng nhỏ càng tốt:



trong đó hệ số là để thuận tiện cho việc tính toán (khi tính đạo hàm thì số sẽ bị triệt tiêu). ở đây ta cần  vì có thể là một số âm, việc nói  nhỏ nhất sẽ không đúng vì khi  là rất nhỏ nhưng sự sai lệch là rất lớn.

**c) Hàm mất mát**

Điều tương tự xảy ra với tất cả các cặp (input, outcome) , với  là số lượng dữ liệu quan sát được. Điều ta muốn, tổng sai số là nhỏ nhất, tương đương với việc tìm  để hàm số sau đạt giá trị nhỏ nhất:

 (2)

Hàm số được gọi là **hàm mất mát** (loss function) của bài toán Linear Regression. Ta luôn mong muốn rằng sự mất mát (sai số) là nhỏ nhất, điều đó đồng nghĩa với việc tìm vector hệ số  sao cho giá trị của hàm mất mát này càng nhỏ càng tốt. Giá trị của  làm cho hàm mất mát đạt giá trị nhỏ nhất được gọi là điểm tối ưu (Optimal point), ký hiệu:



Trước khi đi tìm lời giải, ta đơn giản hóa phép toán trong phương trình hàm mất mát (2). Đặt  là một vector cột chứa tất cả các output của training data là ma trận dữ liệu đầu vào (mở rộng) mà mỗi hàng của nó là một điểm dữ liệu. Khi đó hàm số mất mát  được viết dưới dạng ma trận đơn giản hơn:

 (3)

với là Euclidean norm (chuẩn Euclid, hay khoảng cách Euclid), nói cách khác là tổng của bình phương mỗi phần tử của vector . Tới đây, ta đã có một dạng đơn giản của hàm mất mát được viết như phương trình (3).

**d) Nghiệm cho bài toán Linear Regression**

**Cách phổ biến nhất để tìm nghiệm cho một bài toán tối là giải phương trình đạo hàm (gradient) bằng 0!** Tất nhiên đó là khi việc tính đạo hàm và việc giải phương trình đạo hàm bằng 0 không quá phức tạp. Thật may mắn, với các mô hình tuyến tính, hai việc này là khả thi.

Đạo hàm theo của hàm mất mát là:



Phương trình đạo hàm bằng 0 tương đương với:

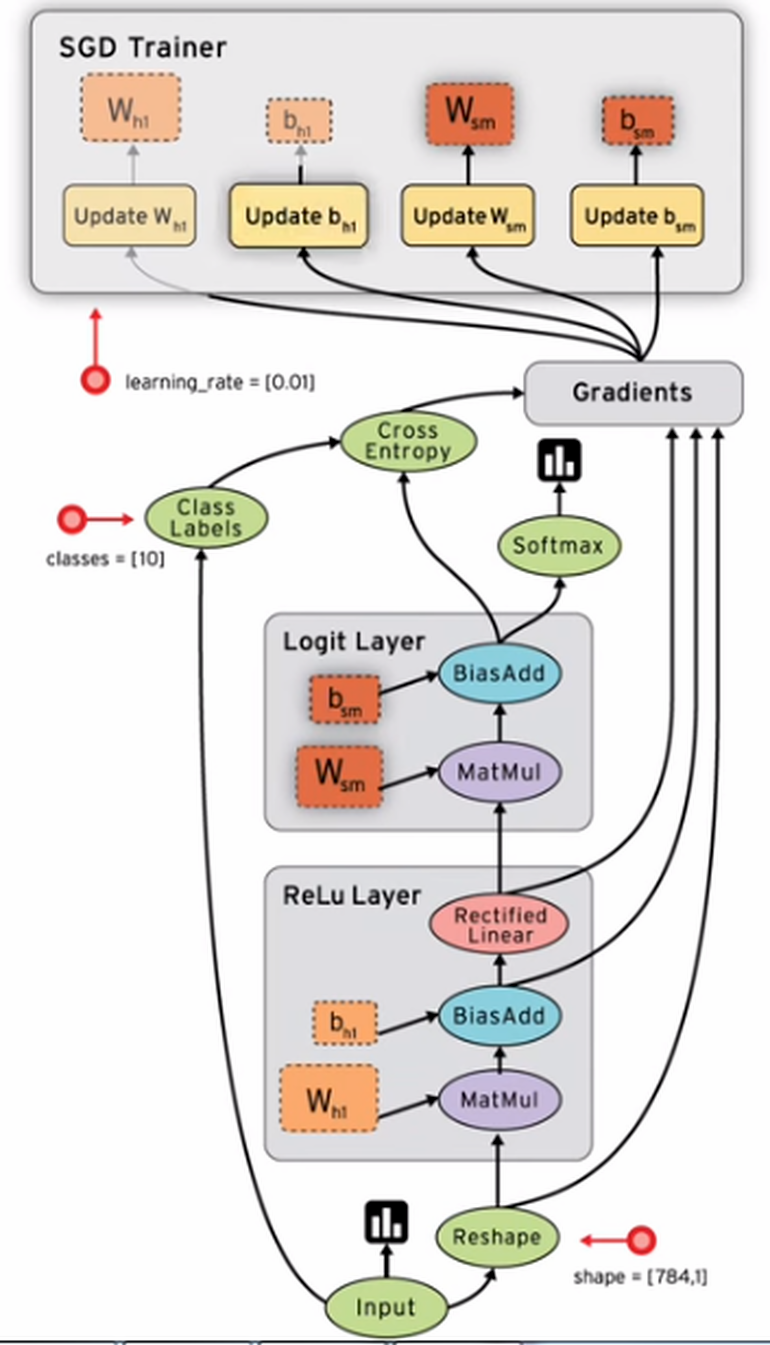
  (4)

Đặt 

Nếu ma trận vuông  khả nghịch (non-singular hay invertible) thì phương trình (4)có nghiệm duy nhất: .

### 1.3. Tensorflow

Nếu hiểu theo cách đơn giản thì Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lí tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Tensorflow được sử dụng khi ta cần giải quyết các bài toán Supervised learning.



Hình 2: Ví dụ về một graph trong Tensorflow.

Khi thực hành với Tensorflow, sẽ có rất nhiều khái niệm phức tạp. Tuy nhiên chỉ ở mức cơ bản, ta sẽ đi đến khái niệm quan trọng nhất trong Tensorflow là **Tensor**.

**Node**

Vì Tensorflow mô tả lại dòng chảy của dữ liệu thông qua graph nên mỗi một điểm giao cắt trong graph thì được gọi là Node. Tại sao điều này quan trọng thì là vì các Node chính là điểm đại diện cho việc thay đổi của dữ liệu nên việc lưu trữ lại tham chiếu của các Node này là rất quan trọng.

**Tensor**

Để giải được các bài toán Machine Learning, cần phải làm cho máy tính có thể hiểu được dữ liệu của tập nguồn và dữ liệu của tập đích. Tensorflow cung cấp một loại dữ liệu mới được gọi là Tensor. Trong thế giới của Tensorflow, mọi kiểu dữ liệu đều được quy về một mối được gọi là Tensor hay trong Tensorflow, tất cả các loại dữ liệu đều là Tensor. Vậy nên có thể hiểu được phần nào cái tên Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor.

Tensor là một kiểu dữ liệu dạng mảng có nhiều chiều được mô tả dạng Tensor = [[[1,1,1],[178,62,74]],[[45,2,2],[19,0,17]],[[7,5,2],[0,11,4]],[[8,13,5],[1,6,7]]]. Mảng nhiều chiều này được đính kèm thêm một vài thuộc tính tham chiếu khác. Các thuộc tính của Tensor được mô tả trong tài liệu bao gồm:

* **device**: Tên của thiết bị mà Tensor hiện tại sẽ được xuất bản. Có thể None.
* **graph**: Đồ thị chứa Tensor hiện tại.
* **name**: Tên của Tensor hiện tại.
* **shape**: Trả về TensorShape mô tả lại Shape của Tensor hiện tại.
* **op**: operation - toán tử (phép toán) được sử dụng để xuất bản Tensor hiện tại.
* **dtype**: Kiểu của các phần tử (elements) trong Tensor hiện tại.

**Rank**

Rank là bậc hay độ sâu của một Tensor. Ví dụ như Tensor = [1] sẽ có rank là 1, Tensor = [[[1,1,1],[178,62,74]]] sẽ có rank bằng 3, Tensor = [[1,1,1],[178,62,74]] sẽ có rank bằng 2. Cách nhanh nhất để xác định rank của một Tensor là đếm số lần mở ngoặc vuông cho đến giá trị khác ngoặc vuông đầu tiên. Việc phân rank này khá quan trọng vì nó đồng thời cũng giúp phân loại dữ liệu của Tensor. Khi ở cách rank đặc biệt cụ thể, Tensor có những tên gọi riêng như sau:

* **Scalar**: Khi Tensor có rank bằng 0, Tensor đại diện cho một số hoặc một chuỗi cụ thể. Ví dụ: scalar=123.
* **Vector**: Vector là một Tensor rank 1. Trong python thì Vector là một list hay mảng một chiều chứa các số. Ví dụ: list=[123,345].
* **Matrix**: Đây là một Tensor rank 2 hay mảng hai chiều theo khái niệm của Python. Ví dụ: matrix=[[1,2],[2,1]].
* **N-Tensor**: Khi rank của Tensor tăng lên lớn hơn 2, chúng được gọi chung là N-Tensor.

**Shape**

Shape là một tuple có số chiều (dimention) bằng với rank của Tensor tương ứng dùng để mô tả lại cấu trúc của Tensor đó. Dưới đây là ví dụ về Shape.

* Tensor = 1 sẽ có Shape = ().
* Tensor = [1] sẽ có Shape = (1).
* Tensor = [[[1,1,1],[178,62,74]]] sẽ có Shape = (1,1,3).
* Tensor = [[1,1,1],[178,62,74]] sẽ có Shape = (1,3).

Dựa vào cấu trúc của Shape, ta dễ dàng thấy rằng ràng buộc cơ bản của Tensor là chiều của các elements trong Tensor tại mỗi bậc phải bằng nhau.

**Op**

Được viết tắt là op, khái niệm Operator là toán tử được dùng để thực thi Tensor tại node đó. Các toán tử này có thể là Const (hằng số), Variable (biến số) , Add (phép cộng), Mul (phép nhân)... Có thể nói, khái niệm operator trong Tensorflow là khái niệm dùng để mô tả lại trạng thái của Node nói chung.

**DType**

Đây là kiểu dữ liệu của các elements trong Tensor. Vì một Tensor chỉ có duy nhất một thuộc tính DType nên từ đó cũng suy ra là chỉ có duy nhất một kiểu DType duy nhất cho toàn bộ các elements có trong Tensor hiện tại. Việc tạo ra hơn một DType khác nhau cho các elements của Tensor là không khả dụng.

**Placeholder**

Đối với thuật toán Linear Regression, đồ thị mong muốn của ta là một hàm bậc nhất một ẩn có dạng y = wx + b với x là dữ liệu tập nguồn, y là dữ liệu tập đích, w là trọng số (weight) của x và b là độ lệch (bias) của hàm mong muốn. Để xây dựng được hàm dưới dạng graph, ta không thể chỉ dùng các Node cố định mà Tensorflow mong muốn giá trị của x và y có thể thay đổi được trong quá trình Session yêu cầu Worker làm việc.

Khi một Node có Operator là placeholder, chúng không cần giá trị cụ thể mà chỉ yêu cầu một kiểu DType đặt trước. Giá trị của Node sẽ được xác định mỗi lần Worker chạy qua chúng thông qua một tập dữ liệu kiểu Dictionaries (từ điển) mà điển hình là Array (mảng).

### 1.4. Triển khai thuật toán Linear Regression bằng Tensorflow

#### 1.4.1. Chuẩn bị dữ liệu

Dữ liệu là một yếu tố bắt buộc trong tất cả các bài toán Machine Learning. Với bài toán dự báo thời tiết, ta có thể download dữ liệu thời tiết Thành phố Hồ Chí Minh tại trang web <https://www.meteoblue.com/en/weather/forecast/week/ho-chi-minh-city_vietnam_1566083>. Trang này cung cấp đầy đủ các thông tin về lượng mưa, độ ẩm, nhiệt độ, nắng gió... Ở đây ta sẽ thu thập dữ liệu về Temperature (Nhiệt độ), Humidity (Độ ẩm) và Sunshine duration (Thời gian sáng). Vì Linear Regression rất **nhạy cảm với nhiễu** (sensitive to noise) nên ta phải trải qua bước loại bỏ nhiễu (outlier) hay còn gọi là tiền xử lý (pre-processing).

#### 1.4.2. Xử lý dữ liệu

Nhập các thư viện sẽ được dùng:

import tensorflow as tf

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from IPython import get\_ipython

ipy = get\_ipython()

if ipy is not None:

ipy.run\_line\_magic('matplotlib','inline')

Tải dữ liệu dùng để training:

file\_path = 'weather.csv'

data=pd.read\_csv(file\_path, delimiter=',',header=0,skipinitialspace=True)

data.head(24)

Như vậy ta đã có được file dữ liệu training, bây giờ chỉ cần gọi data['Temperature'] hoặc data['Wind\_Speed'] ... để sử dụng:

temperature = np.array(data['Temperature'])

Ta có 1 chuỗi các giá trị array, các số tiếp theo sẽ phụ thuộc vào các số trước đó, cũng như vậy ta sẽ có các giá trị thời tiết tại thời điểm trước đó sẽ là X và giá trị thời tiết tại thời điểm trên sau 1 giờ sẽ là Y. Ví dụ như bảng dữ liệu nhiệt độ theo giờ trong 2 ngày 3/4/2019 và 4/4/2019:

|  |  |
| --- | --- |
| X | Y |
| 26.08 | 25.71 |
| 25.71 | 25.32 |
| 25.32 | 25.05 |
| 25.05 | 25.8 |
| 25.8 | 25.73 |
| 25.73 | 26.23 |
| 26.23 | 27.81 |
| 27.81 | 28.49 |
| 28.49 | 31.07 |
| 31.07 | 31.25 |
| 31.25 | 31.46 |
| 31.46 | 32.18 |
| 32.18 | 32.15 |
| 32.15 | 32.17 |
| 32.17 | 32.1 |
| 32.1 | 32.92 |
| 32.92 | 31.77 |
| 31.77 | 30.83 |
| 30.83 | 29.55 |
| 29.55 | 28.63 |
| 28.63 | 27.84 |
| 27.84 | 27.13 |
| 27.13 | 26.6 |

Bảng 1. Mảng dữ liệu dùng để training.

Bài toán sẽ sử dụng 48 giá trị thu thập được để so sánh (một ngày có num\_periods = 24 giờ, và ta lấy dữ liệu của 2 ngày):

num\_periods = 24

f\_horizon = 1

x\_data = temperature[:(len(temperature)-(num\_periods\*2))]

x\_batches = x\_data.reshape(-1, num\_periods, 1)

y\_data = temperature[1:(len(temperature)-(num\_periods\*2))+f\_horizon]

y\_batches = y\_data.reshape(-1, num\_periods, 1)

print(y\_batches.shape)

Dữ liệu đầu vào:

def test\_data(series, forecast, num):

testX = temperature[-(num + forecast):][:num].reshape(-1, num\_periods, 1)

testY = temperature[-(num):].reshape(-1, num\_periods, 1)

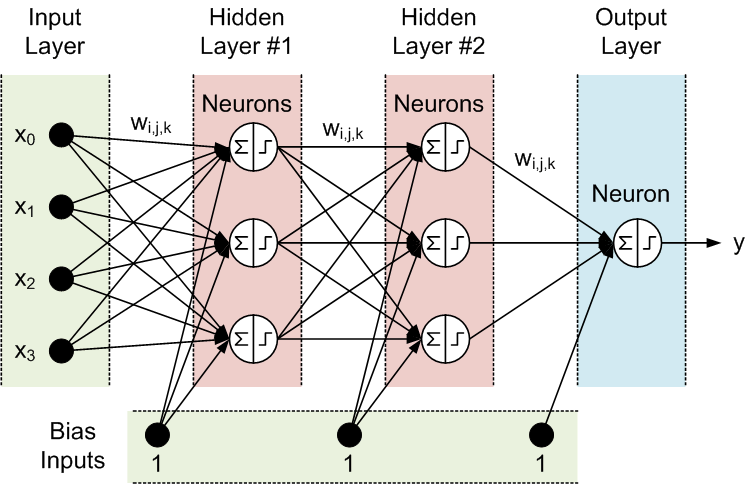
return testX, testY

X\_test, Y\_test = test\_data(temperature, f\_horizon, 24\*2)

print(X\_test.shape)

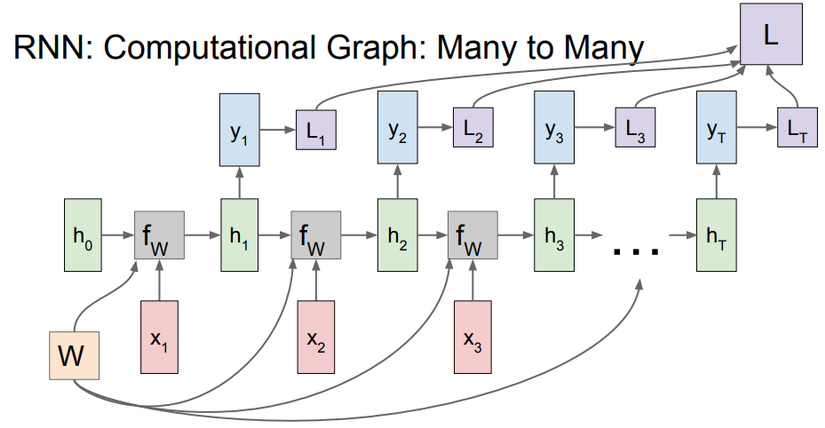
#### 1.4.3. Tiến hành training

Do đặc thù của bài toán này là dùng các thông tin trong quá khứ, để dự đoán tương lai, nên ở đây ta lựa chọn thuật toán **Recurrent Neural Network** (RNN) hay trong tiếng Việt gọi là Mạng Nơron hồi quy. Để có thể hiểu rõ về RNN, trước tiên ta nhìn lại mô hình Neural Network dưới đây:



Hình 3. Mô tả mạng Neural Network.

Neural Network bao gồm 3 phần chính là Input layer, Hidden layer và Output layer, ta có thể thấy là đầu vào và đầu ra của mạng neuron này là độc lập với nhau. Như vậy mô hình này không phù hợp với những bài toán dạng chuỗi như mô tả, hoàn thành câu, ... vì những dự đoán tiếp theo như thời tiết ngày hôm sau phụ thuộc vào thời tiết ngày hôm nay và những ngày trước đó. Và như vậy RNN ra đời với ý tưởng chính là sử dụng một bộ nhớ để lưu lại thông tin từ những bước tính toán xử lý trước để dựa vào đó có thể đưa ra kết quả gần đúng nhất cho bước dự đoán hiện tại.



Hình 4. Mô tả mạng Recurrent Neural Network.

Ý tưởng chính của RNN (Recurrent Neural Network) là sử dụng chuỗi các thông tin. Ví dụ, nếu muốn đoán từ tiếp theo có thể xuất hiện trong một câu thì ta cũng cần biết các từ trước đó xuất hiện lần lượt thế nào. RNN được gọi là hồi quy (Recurrent) bởi lẽ chúng thực hiện cùng một tác vụ cho tất cả các phần tử của một chuỗi với đầu ra phụ thuộc vào cả các phép tính trước đó. Nói cách khác, RNN có khả năng nhớ các thông tin được tính toán trước đó. Trong nhiều mạng neural truyền thống khác, dữ liệu đầu vào và đầu ra hoàn toàn độc lập với nhau, tức là chúng không có liên kết thành chuỗi. Do đó khi áp dụng vào bài toán dự báo thời tiết này sẽ rất khó để đưa ra kết quả dự đoán.

Khai báo thuật toán:

tf.reset\_default\_graph()

inputs = 1

#số 1 được thêm vào dữ liệu đầu vào mở rộng để tiện cho việc tính toán

rnn\_size = 100

#kích thước của mạng RNN (số lượng các đơn vị trong mạng)

output = 1

learning\_rate=0.001

dropout\_keep\_prob = tf.placeholder(tf.float32)

X = tf.placeholder(tf.float32, [None, num\_periods, 1])

Y = tf.placeholder(tf.float32, [None, num\_periods, 1])

rnn\_cells=tf.contrib.rnn.BasicRNNCell(num\_units=rnn\_size, activation=tf.nn.relu)

rnn\_output, states = tf.nn.dynamic\_rnn(rnn\_cells, X, dtype=tf.float32)

#mô hình RNN truyền thống với hàm activation là relu

output=tf.reshape(rnn\_output, [-1, rnn\_size])

logit=tf.layers.dense(output, 1, name="softmax")

outputs=tf.reshape(logit, [-1, num\_periods, 1])

print(logit)

loss = tf.reduce\_sum(tf.square(outputs - Y))

#hàm tính toán mất mát

accuracy = tf.reduce\_mean(tf.cast(tf.equal(tf.argmax(logit, 1), tf.cast(Y, tf.int64)), tf.float32))

#hàm tính toán độ chính xác

optimizer = tf.train.AdamOptimizer(learning\_rate=learning\_rate)

train\_step=optimizer.minimize(loss)

#tối ưu hóa bằng cách tối thiểu mất mát

init=tf.global\_variables\_initializer()

Bắt đầu training:

epochs = 1000

sess = tf.Session()

init = tf.global\_variables\_initializer()

sess.run(init)

saver = tf.train.Saver()

for epoch in range(epochs):

train\_dict = {X: x\_batches, Y: y\_batches, dropout\_keep\_prob:0.5}

sess.run(train\_step, feed\_dict=train\_dict)

y\_pred=sess.run(outputs, feed\_dict={X: X\_test})

save\_path = saver.save(sess, "models/model.ckpt")

#lưu lại model để sử dụng cho lần sau

Vậy là quá trình training đã hoàn thành. Khi test dữ liệu, ta cần restore lại model vừa training để sử dụng:

with tf.Session() as sess:

# Restore variables from disk.

saver.restore(sess, "models/model.ckpt")

predict=sess.run(outputs, feed\_dict={X: X\_test})

Kết quả dự đoán:

[26.635757 26.93783 27.205732 27.298222 27.330328 26.826542 28.372198

27.943539 30.013775 32.1727 33.785038 33.33924 32.510853 33.169624

32.00715 31.931244 30.936794 29.56342 29.963161 29.379179 27.046267

25.87078 26.401134 26.679441 26.055984 25.77284 25.440844 24.99861

24.881916 25.628038 28.326578 28.239164 29.302927 29.70434 31.870382

32.757465 32.47599 32.42788 31.593191 31.471987 31.044271 30.604467

29.728916 28.979906 28.322004 27.719961 26.978312 26.42832 ]

Lưu kết quả ra file “predict.csv”

import csv

csvData = y\_predict

with open('predict.csv', 'w') as csvFile:

writer = csv.writer(csvFile)

writer.writerow(csvData)

csvFile.close()

Nếu chỉ nhìn vào dữ liệu đầu ra như thế kia thôi thì cũng không biết liệu kết quả dự đoán của ta sẽ ra sao. Do đó để dễ dàng so sánh, việc vẽ đồ thị sẽ rất cần thiết để giúp ta có cái nhìn chính xác để đánh giá mô hình dự đoán này.

fig=plt.figure()

plt.title("Temperature", fontsize=14)

plt.plot(pd.Series(np.ravel(Y\_test)), "bo", markersize=10, label="Actual")

plt.plot(pd.Series(np.ravel(y\_pred)), "r.", markersize=10, label="Forecast")

plt.legend(loc="upper left")

plt.xlabel("Time Periods")

plt.show()

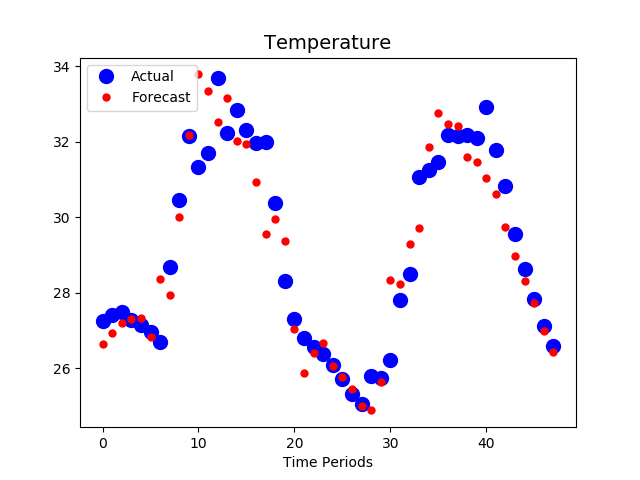
fig.savefig('TemSS.png',

facecolor=fig.get\_facecolor(),

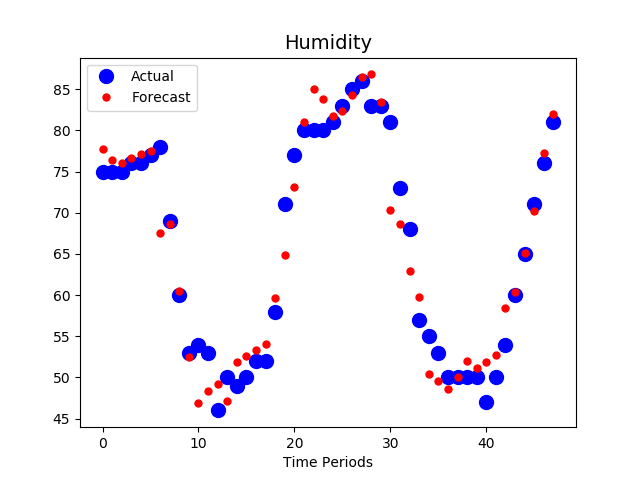
edgecolor=fig.get\_edgecolor(),

dpi = fig.get\_dpi())

#lưu lại đồ thị dưới tên TemSS.png



Hình 5. Đồ thị so sánh nhiệt độ dự báo và thực tế trong 48 giờ.



Hình 6. Đồ thị so sánh độ ẩm tương đối dự báo và thực tế trong 48 giờ.

Trên đồ thị, chấm tròn nhỏ màu đỏ biểu thị giá trị thời tiết mà mô hình đã dự đoán (Forecast), còn chấm tròn ta hơn màu xanh biểu thị giá trị thời tiết thực tế (Actual). Chi tiết hơn, ta thực hiện một bảng tính toán độ chính xác của phép dự đoán nhiệt độ

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Hour | Actual | Forecast | Accuracy | Hour | Actual | Forecast | Accuracy |
| 1 | 27.25 | 26.63576 | 97.75 | 25 | 26.08 | 26.05598 | 99.91 |
| 2 | 27.41 | 26.93783 | 98.28 | 26 | 25.71 | 25.77284 | 99.76 |
| 3 | 27.48 | 27.20573 | 99.00 | 27 | 25.32 | 25.44084 | 99.52 |
| 4 | 27.29 | 27.29822 | 99.97 | 28 | 25.05 | 24.99861 | 99.79 |
| 5 | 27.14 | 27.33033 | 99.30 | 29 | 25.8 | 24.88192 | 96.44 |
| 6 | 26.95 | 26.82654 | 99.54 | 30 | 25.73 | 25.62804 | 99.60 |
| 7 | 26.7 | 28.3722 | 93.74 | 31 | 26.23 | 28.32658 | 92.01 |
| 8 | 28.68 | 27.94354 | 97.43 | 32 | 27.81 | 28.23916 | 98.46 |
| 9 | 30.45 | 30.01377 | 98.57 | 33 | 28.49 | 29.30293 | 97.15 |
| 10 | 32.16 | 32.1727 | 99.96 | 34 | 31.07 | 29.70434 | 95.60 |
| 11 | 31.34 | 33.78504 | 92.20 | 35 | 31.25 | 31.87038 | 98.01 |
| 12 | 31.71 | 33.33924 | 94.86 | 36 | 31.46 | 32.75747 | 95.88 |
| 13 | 33.7 | 32.51085 | 96.47 | 37 | 32.18 | 32.47599 | 99.08 |
| 14 | 32.23 | 33.16962 | 97.08 | 38 | 32.15 | 32.42788 | 99.14 |
| 15 | 32.84 | 32.00715 | 97.46 | 39 | 32.17 | 31.59319 | 98.21 |
| 16 | 32.31 | 31.93124 | 98.83 | 40 | 32.1 | 31.47199 | 98.04 |
| 17 | 31.98 | 30.93679 | 96.74 | 41 | 32.92 | 31.04427 | 94.30 |
| 18 | 32 | 29.56342 | 92.39 | 42 | 31.77 | 30.60447 | 96.33 |
| 19 | 30.38 | 29.96316 | 98.63 | 43 | 30.83 | 29.72892 | 96.43 |
| 20 | 28.32 | 29.37918 | 96.26 | 44 | 29.55 | 28.97991 | 98.07 |
| 21 | 27.3 | 27.04627 | 99.07 | 45 | 28.63 | 28.322 | 98.92 |
| 22 | 26.8 | 25.87078 | 96.53 | 46 | 27.84 | 27.71996 | 99.57 |
| 23 | 26.55 | 26.40113 | 99.44 | 47 | 27.13 | 26.97831 | 99.44 |
| 24 | 26.38 | 26.67944 | 98.86 | 48 | 26.6 | 26.42832 | 99.35 |

Bảng 2. Tính toán độ chính xác của dự báo.

Với tỉ lệ chính xác vào khoảng 96.75% chứng tỏ giải thuật ta sử dụng đáng tin cậy và không có sự sai lệch quá nhiều. Tuy nhiên, ở nhưng thời điểm có biên độ dao động lớn của nhiệt độ và độ ẩm, thì dữ liệu dự báo kém chính xác hơn. Nếu có thể thu thập được dữ liệu trong khoảng thời gian lâu hơn, lên đến trên 10 năm, thì kết quả dự báo sẽ được tối ưu hóa. Bên cạnh việc giảm thiểu chênh lệch giữa dự đoán và thực tế, thì trong thực tế hoàn toàn có thể xảy ra những dữ liệu bất thường, gây nhiễu đầu vào của bài toán nếu không trải qua bước tiền xử lý và làm ảnh hưởng lớn đến tính chính xác của kết quả. Đó cũng là một nhược điểm của mô hình Linear Regression.

### 1.5. Tính toán dự trên dự báo và đưa ra khuyến nghị

Xây dựng giải thuật dự đoán mưa như sau: từ giá trị nhiệt độ tại một thời điểm ta đánh giá trọng số cho nhiệt độ này theo bảng:

|  |  |
| --- | --- |
| Nhiệt độ (oC) | Trọng số |
| 40<t | 2 |
| 35<t≤40 | 2,5 |
| 32<t≤35 | 3 |
| 30<t≤32 | 4 |
| 26<t≤30 | 5 |
| 24<t≤26 | 4 |
| 18<t≤24 | 5 |
| 13<t≤18 | 3 |
| 8<t≤13 | 2 |
| 2<t≤8 | 2 |
| t≤2 | 1 |

Bảng 3. Đánh giá trọng số cho giá trị nhiệt độ.

Tương tự với giá trị độ ẩm tương đối, ta cũng xây dựng bảng trọng số như sau:

|  |  |
| --- | --- |
| Độ ẩm ( %) | Trọng số |
| h<40 | 0.5 |
| 40<h≤50 | 1 |
| 50<h≤55 | 1.5 |
| 55<h≤60 | 2 |
| 60<h≤65 | 2.5 |
| 65<h≤70 | 3 |
| 70<h≤75 | 3.5 |
| 75<h≤80 | 4 |
| 80<h≤85 | 5 |
| 85<h≤90 | 6 |
| h<95 | 7 |

Bảng 4. Đánh giá trọng số cho giá trị độ ẩm.

Sau đó, chúng ta dự đoán mưa hay không mưa dựa theo công thức sau: Nếu 0.4 lần trọng số nhiệt độ cộng với 0.6 trọng số độ ẩm lớn hơn 6.0 thì đưa ra dự báo “mưa”, ngược lại dự báo “không mưa”. Tiếp đến, ta gán giá trị cho “dự báo trời nắng” theo bảng sau:

|  |  |
| --- | --- |
| Ánh sáng (W/m2) | Dự đoán |
| s≤50 | Trời tối |
| 50<s≤250 | Nắng nhẹ |
| 250<s≤500 | Trời nắng |
| 500<s | Nắng gắt |

Bảng 5. Đánh giá trời nắng dự vào số liệu.

Cuối cùng, ta tổng hợp số liệu và dự báo của 2 ngày 3/4/2019 và 4/4/2019:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Giờ | Nhiệt độ dự  đoán [oC] | Độ ẩm dự  đoán [%] | Ánh sáng dự  đoán [W/m2] | Dự đoán  mưa | Dự đoán  nắng | Trọng số  nhiệt độ | Trọng số  độ ẩm |
| 1 | 25.6 | 82.8 | 0 | no | trời tối | 4 | 5 |
| 2 | 25.1 | 83.1 | 0 | no | trời tối | 4 | 5 |
| 3 | 24.6 | 86.5 | 1.59482491 | no | trời tối | 4 | 6 |
| 4 | 23.9 | 89.9 | 7.355950356 | no | trời tối | 5 | 6 |
| 5 | 23.4 | 90.5 | 14.64730263 | yes | trời tối | 5 | 7 |
| 6 | 23.2 | 91.7 | 25.78609085 | yes | trời tối | 5 | 7 |
| 7 | 25.6 | 76.5 | 79.7671051 | no | nắng nhẹ | 4 | 4 |
| 8 | 26.2 | 75.2 | 130.3968964 | no | nắng nhẹ | 5 | 4 |
| 9 | 28 | 64.2 | 458.3119507 | no | trời nắng | 5 | 2.5 |
| 10 | 30.8 | 47 | 679.1079712 | no | nắng gắt | 4 | 1 |
| 11 | 33.4 | 37.3 | 852.4689941 | no | nắng gắt | 3 | 0.5 |
| 12 | 34.9 | 29.6 | 450.8434448 | no | trời nắng | 3 | 0.5 |
| 13 | 35.8 | 28.6 | 788.7572021 | no | nắng gắt | 2.5 | 0.5 |
| 14 | 36.2 | 27 | 484.7685852 | no | trời nắng | 2.5 | 0.5 |
| 15 | 36.7 | 29.3 | 565.8621826 | no | nắng gắt | 2.5 | 0.5 |
| 16 | 36.6 | 32.3 | 77.42417908 | no | nắng nhẹ | 2.5 | 0.5 |
| 17 | 35.8 | 38 | 100.7989502 | no | nắng nhẹ | 2.5 | 0.5 |
| 18 | 34.2 | 44.6 | 225.3500977 | no | nắng nhẹ | 3 | 1 |
| 19 | 32.1 | 42.3 | 287.2237244 | no | trời nắng | 3 | 1 |
| 20 | 29.6 | 50.4 | 0 | no | trời tối | 5 | 1.5 |
| 21 | 27.2 | 86.3 | 0 | no | trời tối | 5 | 6 |
| 22 | 26.2 | 84.6 | 0 | no | trời tối | 5 | 5 |
| 23 | 26.2 | 76.5 | 55.23558426 | no | nắng nhẹ | 5 | 4 |
| 24 | 26.5 | 79.6 | 28.67375374 | no | trời tối | 5 | 4 |
| 25 | 26.7 | 77.7 | 0 | no | trời tối | 5 | 4 |
| 26 | 26.9 | 75.6 | 0 | no | trời tối | 5 | 4 |
| 27 | 26.9 | 76 | 1.59482491 | no | trời tối | 5 | 4 |
| 28 | 27 | 76.1 | 7.355950356 | no | trời tối | 5 | 4 |
| 29 | 27 | 76 | 14.64730263 | no | trời tối | 5 | 4 |
| 30 | 26.8 | 76.6 | 25.78609085 | no | trời tối | 5 | 4 |
| 31 | 29.2 | 67.5 | 79.7671051 | no | nắng nhẹ | 5 | 3 |
| 32 | 29.6 | 66.4 | 130.3968964 | no | nắng nhẹ | 5 | 3 |
| 33 | 30.5 | 61.7 | 355.2525635 | no | trời nắng | 4 | 2.5 |
| 34 | 32.1 | 53.5 | 504.1925049 | no | nắng gắt | 3 | 1.5 |
| 35 | 33.9 | 46.9 | 1056.568359 | no | nắng gắt | 3 | 1 |
| 36 | 33.6 | 45.9 | 900.5276489 | no | nắng gắt | 3 | 1 |
| 37 | 32.8 | 48.4 | 376.4900208 | no | trời nắng | 3 | 1 |
| 38 | 33.4 | 45.1 | 792.3029785 | no | nắng gắt | 3 | 1 |
| 39 | 33 | 51.4 | 306.3743591 | no | trời nắng | 3 | 1.5 |
| 40 | 32.1 | 50.9 | 447.4032288 | no | trời nắng | 3 | 1.5 |
| 41 | 31.2 | 54.8 | 84.39355469 | no | nắng nhẹ | 4 | 1.5 |
| 42 | 29.9 | 57 | 102.3521194 | no | nắng nhẹ | 5 | 2 |
| 43 | 29.3 | 60.3 | 116.6161499 | no | nắng nhẹ | 5 | 2.5 |
| 44 | 28.7 | 65.2 | 119.4229507 | no | nắng nhẹ | 5 | 3 |
| 45 | 28.1 | 68.3 | 0 | no | trời tối | 5 | 3 |
| 46 | 27.4 | 74.1 | 0 | no | trời tối | 5 | 3.5 |
| 47 | 26.8 | 78.2 | 65.45771027 | no | nắng nhẹ | 5 | 4 |
| 48 | 26.3 | 80.5 | 56.33896255 | no | nắng nhẹ | 5 | 5 |

Bảng 6. Tổng hợp dữ liệu và kết quả dự báo.

**Nhận xét:**

1. **Thiết kế và thực hiện phần cứng**
   1. **Các bộ phận chính**
      1. **Module wifi**

**Tìm hiểu về wifi**

Wi-Fi viết tắt từ Wireless Fidelity hay mạng 802.11 là hệ thống mạng không dây sử dụng sóng vô tuyến, giống như điện thoại di động, truyền hình và radio. Hệ thống này đã hoạt động ở một số sân bay, quán café, thư viện hoặc khách sạn. Hệ thống cho phép truy cập Internet tại những khu vực có sóng của hệ thống này, hoàn toàn không cần đến cáp nối. Ngoài các điểm kết nối công cộng (hotspots), WiFi có thể được thiết lập ngay tại nhà riêng.

Tên gọi 802.11 bắt nguồn từ viện IEEE (Institute of Electrical and Electronics Engineers). Viện này tạo ra nhiều chuẩn cho nhiều giao thức kỹ thuật khác nhau, và nó sử dụng một hệ thống số nhằm phân loại chúng; 6 chuẩn thông dụng của WiFi hiện nay là 802.11a/b/g/n/ac/ad.

**Sóng Wifi:**

Các sóng vô tuyến sử dụng cho WiFi gần giống với các sóng vô tuyến sử dụng cho thiết bị cầm tay, điện thoại di động và các thiết bị khác. Nó có thể chuyển và nhận sóng vô tuyến, chuyển đổi các mã nhị phân 1 và 0 sang sóng vô tuyến và ngược lại. Tuy nhiên, sóng WiFi có một số khác biệt so với các sóng vô tuyến khác ở chỗ: Chúng truyền và phát tín hiệu ở tần số 2.4 GHz hoặc 5 GHz. Tần số này cao hơn so với các tần số sử dụng cho điện thoại di động, các thiết bị cầm tay và truyền hình. Tần số cao hơn cho phép tín hiệu mang theo nhiều dữ liệu hơn. Chúng dùng chuẩn 802.11:

- Chuẩn 802.11b là phiên bản đầu tiên trên thị trường. Đây là chuẩn chậm nhất và rẻ tiền nhất, và nó trở nên ít phổ biến hơn so với các chuẩn khác. 802.11b phát tín hiệu ở tần số 2.4 GHz, nó có thể xử lý đến 11 megabit/giây, và nó sử dụng mã CCK (complimentary code keying).

- Chuẩn 802.11g cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn 802.11b, tốc độ xử lý đạt 54 megabit/giây. Chuẩn 802.11g nhanh hơn vì nó sử dụng mã OFDM (orthogonal frequency-division multiplexing), một công nghệ mã hóa hiệu quả hơn.

* Chuẩn 802.11a phát ở tần số 5 GHz và có thể đạt đến 54 megabit/ giây. Nó cũng sử dụng mã OFDM. Những chuẩn mới hơn sau này như 802.11n còn nhanh hơn chuẩn 802.11a, nhưng 802.11n vẫn chưa phải là chuẩn cuối cùng.
* Chuẩn 802.11n cũng phát ở tần số 2.4 GHz, nhưng nhanh hơn so với chuẩn 802.11a, tốc

độ xử lý đạt 300 megabit/giây.

* Chuẩn 802.11ac phát ở tần số 5 GHz.
* Chuẩn 802.11ad phát ở tần số 60 GHz.

WiFi có thể hoạt động trên cả ba tần số và có thể nhảy qua lại giữa các tần số khác nhau một cách nhanh chóng. Việc nhảy qua lại giữa các tần số giúp giảm thiểu sự nhiễu sóng và cho phép nhiều thiết bị kết nối không dây cùng một lúc. Tuy nhiên mạng không dây luôn vấp phải những vấn đề tồn tại khiến nó chưa thể phát triển và thay thế cho toàn bộ cách kết nối hiện nay. Nhưng vấn đề tiện lợi cho người dùng sẽ luôn đẩy công nghệ không dây ngày càng phát triển và chúng ta có thể khẳng định rằng mạng không dây sẽ ngày càng phát triển và thay thế dần dần mạng sử dụng cáp hiện nay.

**Phạm vi Wifi:**

Phạm vi tín hiệu Wifi phụ thuộc vào bằng tần, công suất đầu ra, độ lợi và loại anten cũng như các kỹ thuật điều chế. Line-of-sight là hướng dẫn thu nhỏ nhưng sự khúc xạ và phản xạ gây ra ảnh hưởng đáng kể.

Một điểm truy cập phù hợp với một trong hai chuẩn 802.11b hay 802.11g , sử dụng ăng-ten chứng khoán có thể có một phạm vi 100 m (330 ft). Các đài phát thanh cùng với một ăng-ten parabol bán bên ngoài (15 dB đạt được) có thể có một phạm vi hơn 20 dặm.

Tỷ lệ độ lợi cao hơn (dBi) cho thấy độ lệch hơn nữa (thường về hướng ngang) từ một lý thuyết, bức xạ đẳng hướng hoàn hảo , và vì thế hơn nữa anten có thể dự đoán một tín hiệu thích hợp, cũng như so sánh công suất đầu ra tương tự trên một ăng-ten đẳng hướng hơn. Ví dụ, một ăng-ten 8 dBi sử dụng với một trình điều khiển 100 mW sẽ có một phạm vi ngang tương tự như một ăng-ten 6 dBi được thúc đẩy tại 500 mW. Lưu ý rằng điều này giả định rằng bức xạ theo chiều thẳng đứng bị mất; điều này có thể không phải là trường hợp trong một số tình huống, đặc biệt là trong các tòa nhà lớn hoặc trong một ống dẫn sóng . Trong ví dụ trên, một ống dẫn sóng đẳng hướng có thể gây ra cho anten công suất thấp 6 DBi đưa ra nhiều hơn một đẳng hướng so với anten 8 dBi mà không phải là trong một ống dẫn sóng, ngay cả khi họ được cả hai đang được thúc đẩy ở 100 mW.

**Module Wifi ESP32**



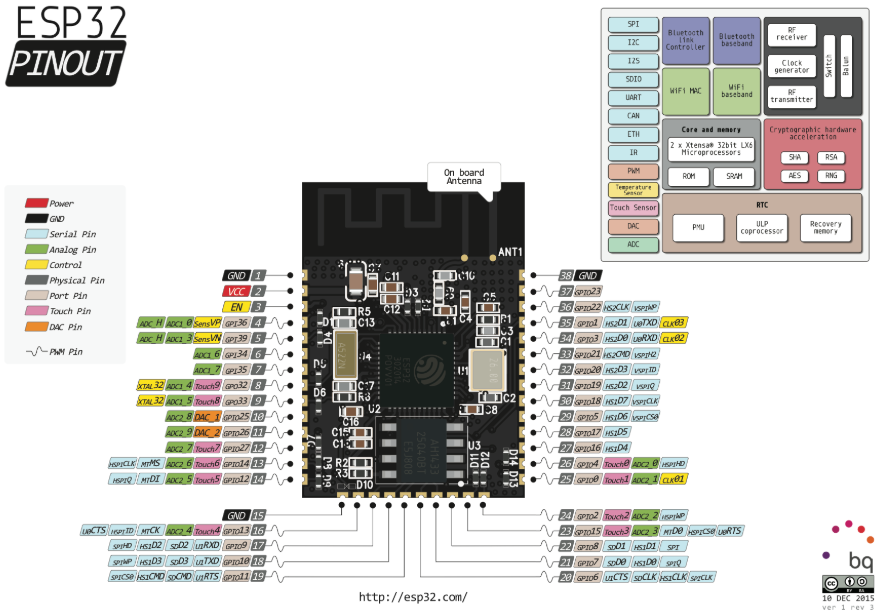
Hình 7. Ảnh thực tế của ESP32

ESP32 là module wifi giá rẻ và được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác.

ESP32 là một chip tích hợp cao, được thiết kế cho nhu cầu của một thế giới kết nối mới, thế giới Internet of thing (IoT). Nó cung cấp một giải pháp kết nối mạng Wi-Fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

ESP32 có xử lý và khả năng lưu trữ mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với một chi phí tối thiểu và một PCB tối thiểu. Sử dụng giao thức nối tiếp với tốc độ Baud là 9600(mặc định). Tốc độ baud có thể lên đến 115200.

ESP32 là module sử dụng các tập lệnh AT. Tuy nhiên đã có thêm nhiều phiên bản firmware hỗ trợ nhiều ngôn ngữ, trong đó có LUA và Arduino. Chính vì thế có thể dùng bất kì board TivaC nào để điều khiển thu phát thông qua phương thức truyền nối tiếp.



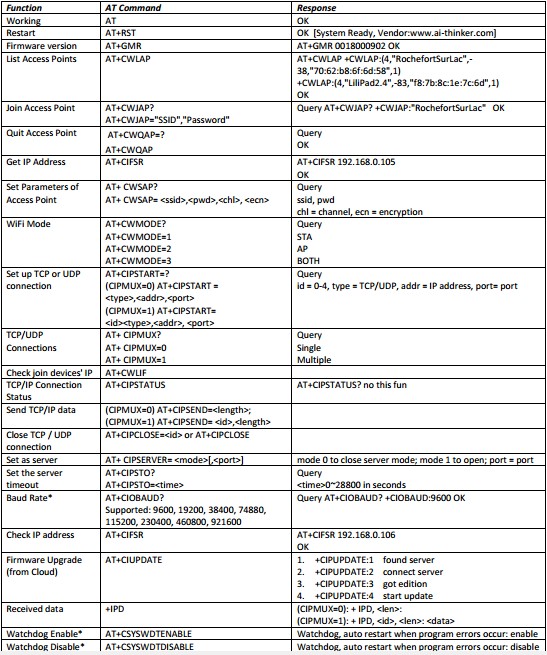
Hình 2.8 : Sơ đồ ra chân của ESP32

Chức năng các chân cơ bản:

* Vcc: 3.3V.
* GND: 0V.
* TX: chân TX của giao thức UART, kết nối đến chân RX của vi điều khiển.
* RX: chân RX của giao thức UART, kết nối đến chân TX của vi điều khiển.
* RST: chân reset kéo xuống mass để bật reset.
* CH-PD: chân này kéo lên mức cao module bắt đầu thu phát, kéo xuống thấp module dừng. Vì ESP32 khởi động hút dòng lớn nên phải để mức thấp khi khởi động xong sẽ kéo lên mức cao.
* GPIO0: kéo xuống thấp cho chế độ upgrade firmwave.

**Chức năng:**

* WiFi ESP32, Bluetooth LE SoC - 240Mhz - Mô-đun ESP-WROOM-32
* Tự động chọn 3 nguồn năng lượng (DC6-28V, USB và Pin)
* Tự động tải xuống chế độ Flash
* Khe cắm SDCARD tích hợp (hỗ trợ chế độ 1 bit)
* Thiết kế phần cứng mở với giấy phép KiCad, CC-BY-SA.
* Tiêu đề hiển thị OLED I2C
* Bộ sạc pin lithium-Ion
* 1 nút đặt lại, 1 nút lập trình
* 1 đèn LED nguồn, 1 đèn LED lập trình, 1 đèn LED sạc
* Tương thích với Shields cho ESP32 trong tương lai (Cổng - GSM / GPRS / GPS và Lora Shield, Kết nối - CAN, RS485, Shield Shield, Audio Shield, ...)



Bảng 7. Các tập lệnh AT của ESP32

#### 2.1.2. TivaC LaunchPad MCU TM4C123G

Bộ đánh giá LaunchPad TM4C123G là một nền tảng đánh giá chi phí thấp cho các bộ vi điều khiển dựa trên ARM Cortex-M4F của Texas Cụ. Thiết kế của LaunchPad TM4C123G làm nổi bật bộ vi điều khiển TM4C123GH6PM với giao diện thiết bị USB 2.0 và mô-đun ngủ đông. EK-TM4C123GXL cũng có các nút người dùng có thể lập trình và đèn LED RGB cho các ứng dụng tùy chỉnh. Các tiêu đề có thể xếp chồng lên nhau của Giao diện LaunchPad BoosterPack ™ XL của TM4C123G giúp dễ dàng và đơn giản để mở rộng chức năng của LaunchPad TM4C123G khi giao tiếp với các thiết bị ngoại vi khác với TexasU MCU BoosterPack .



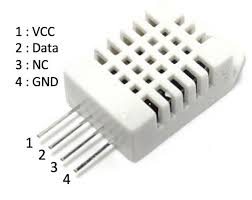
Hình 9. LaunchPad MCU TM4C123G

Tính năng, đặc điểm

* Bộ công cụ đánh giá LaunchPad MCU TM4C123G dựa trên ARM Cortex-M4F (EK-TM4C123GXL) cung cấp các tính năng sau:
* Hiệu suất cao TM4C123GH6PM MCU:
* CPU vi điều khiển dựa trên ARM Cortex-M4 80-bit 32-bit
* Flash 256KB, SRAM 32KB, EEPROM 2KB
* Hai mô-đun Mạng điều khiển (CAN)
* Máy chủ / Thiết bị USB 2.0 / OTG + PHY
* ADC 2MSPS 12 bit kép, bộ điều khiển chuyển động PWM
* 8 UART, 6 I2C, 4 SPI
* Giao diện gỡ lỗi trên bo mạch (ICDI)
* Cắm USB Micro-B vào cáp cắm USB-A
* Ứng dụng khởi động nhanh RGB được cài sẵn
* Hướng dẫn dễ sử dụng
* EK-TM4C123GXL TivaWare ™ cho phần mềm C Series có sẵn để tải xuống miễn phí

**2.1.3. Các cảm biến**

**Cảm biến nhiệt độ - độ ẩm DHT22**



Thu thập nhiệt độ, độ ẩm môi trường để đưa đưa dải tín hiệu vào Arduino, từ đó arduino trong module ESP32 sẽ giải mã tín hiệu để lấy ra được nhiệt độ và độ ẩm. Từ dữ liệu đã giải mã được ESP32 sẽ gửi nhiệt độ và độ ẩm lên server để server có thể điều khiển thiết bị.

**Cảm biến độ ẩm đất**



Cảm biến độ ẩm đất 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 ESP32.

**Cảm biến ánh sáng**



Cảm biến ánh sáng Photodiod sử dụng Photodiod thay vì quang trở để cảm biến ánh sáng nên cho độ chính xác và độ ổn định cao, cảm biến có Opamp tích hợp cho cả hai ngõ ra tín hiệu là Digital và Analog rất dễ sử dụng, thích hợp cho các ứng dụng nhận biết sáng tối, cảm biến cường độ sáng,...

**2.1.4. Các thiết bị ngõ ra**

**Module relay**

****

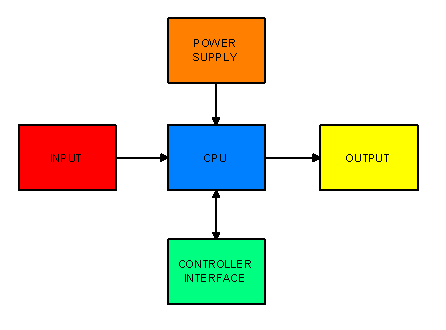
Module 2 Relay kích H/L (5VDC) sử dụng nguồn 5VDC để nuôi mạch, tín hiệu kích có thể tùy chọn kích mức cao (High - 5VDC) hoặc mức thấp (Low - 0VDC) qua Jumper trên mỗi relay. Thích hợp cho các thiết bị sử dụng mức tín hiệu 5VDC như Vi điều khiển. Ngoài ra có thể chọn các module có số lượng relay tùy ý phụ thuộc vào số lượng thiết bị sử dụng.

**Động cơ bơm nước**

**Đèn chiếu sáng**

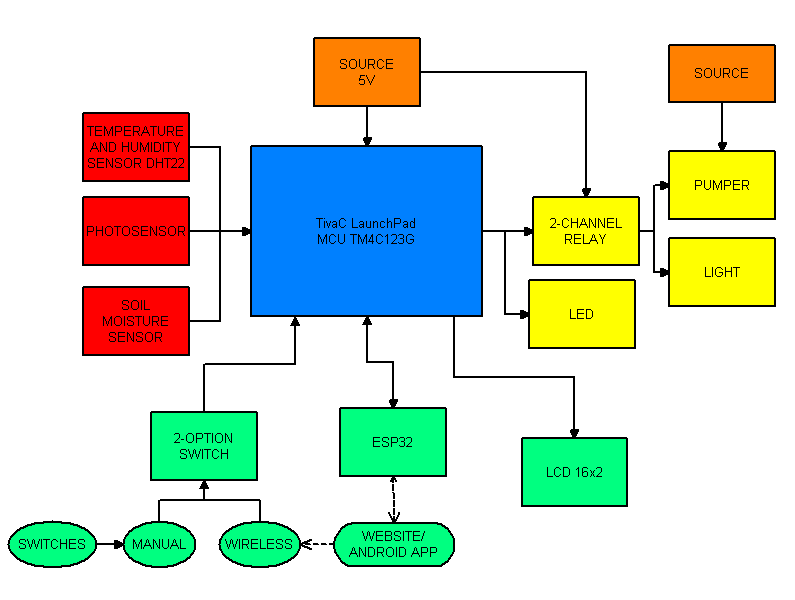
* 1. **Sơ đồ khối hệ thống**

**Sơ đồ phần cứng khái quát**



**Phân tích thiết kế**

|  |  |  |  |
| --- | --- | --- | --- |
| **Components** | **Option 1** | **Option 2** | **Final Option** |
| **Description** | **Description** | **Description** |
| Processor /  Microcontroller | PIC16Fxxx | Arduino Uno R3 | ARM TM4C123G |
| Wireless transceiver circuit | Bluetooth  HC-05 | ESP8266 | ESP32 |
| Photosensor | Photoresistor (CDS) | Photomultiplier tube | Photodiode |
| Temperature and humidity sensor | AM2301 | DHT11 | DHT22 |
| Display | Graphic LCD  128x64 | Text LCD  LCD 2004 | Text LCD  LCD 1602 |

****

**Sơ đồ nguyên lý**

**Mạch in PCB**

* 1. **Lập trình cho phần cứng**

**Code ESP32**

**Code TivaC LauchPad**

* 1. **Mạch thực tế**

1. **Hiển thị số liệu đo đạc, số liệu dự báo và điều khiển thiết bị qua Website**

**Tổng quan TCP/IP**

TCP/IP là một họ các giao thức được gọi là họ giao thức IP, bao gồm bốn tầng. Cần nhớ rằng TCP/IP không phải là một giao thức mà thực sự là một họ các giao thức, và bao gồm các giao thức mức thấp khác như IP, TCP, và UDP. UDP nằm ở tầng giao vận, phía trên giao thức IP. Tầng giao vận cung cấp khả năng truyền tin giữa các mạng thông qua các gateway. Nó sử dụng các địa chỉ IP để gửi các gói tin trên Internet hoặc trên mạng thông qua các trình điều khiển thiết bị khác nhau. TCP và UDP là một phần của họ giao thức TCP/IP; mỗi giao thức có những ưu và nhược điểm riêng của nó.

Giao thức UDP là giao thức đơn giản, phi liên kết và cung cấp dịch vụ trên tầng giao vận với tốc độ nhanh. Nó hỗ trợ liên kết một - nhiều và thường được sử dụng thường xuyên trong liên kết một - nhiều bằng cách sử dụng các datagram multicast và unicast. Giao thức IP là giao thức cơ bản của Internet. TCP và UDP đều là hai giao thức tầng giao thức vận trên cơ sở của giao thức IP.

**TCP**

TCP là viết tắt của Transision Control Protocol. Đó là giao thức phổ biến nhất được sử dụng trên Internet. Khi bạn tải một trang web, máy tính của bạn sẽ gửi các gói tin TCP đến địa chỉ máy chủ web, yêu cầu nó gửi các trang web cho bạn. Các máy chủ web phản ứng bằng cách gửi một dòng của các gói tin TCP, mà trình duyệt web của bạn sẽ kết nối với nhau để tạo nên trang web và hiển thị nó cho bạn. Khi bạn bấm vào một liên kết, đăng nhập, bình luận, hoặc làm bất cứ điều gì khác, trình duyệt web của bạn gửi các gói tin TCP đến máy chủ và máy chủ sẽ gửi các gói tin TCP lại. TCP không phải là thông tin liên lạc mà chỉ là một cách - hệ thống từ xa gửi gói tin trở lại để thừa nhận nó nhận được các gói dữ liệu của bạn. TCP đảm bảo người nhận sẽ nhận được các gói tin theo thứ tự chúng đánh số. Bên nhận sẽ gửi tin nhắn lại cho bên gửi để xác nhận nó đã nhận được tin nhắn. Nếu bên gửi không nhận được một phản ứng chính xác, nó sẽ gửi lại gói tin để đảm bảo bên nhận nhận được. Gói tin cũng được kiểm tra lỗi. Giao thức TCP được xem như là tất cả về độ tin cậy này - các gói tin gửi đi với TCP được theo dõi sát sao, vì vậy không có dữ liệu bị mất hoặc bị hỏng trên đường vận chuyển. Đây là lý do tại sao tập tin tải về không bị lỗi ngay cả khi có trục trặc mạng. Tất nhiên, nếu người nhận là hoàn toàn ẩn, máy tính của bạn sẽ từ bỏ và bạn sẽ thấy một thông báo lỗi nói rằng nó không thể giao tiếp với máy chủ từ xa.

TCP đòi hỏi thiết lập kết nối trước khi bắt đầu gửi dữ liệu và kết thúc kết nối khi việc gửi dữ liệu hoàn tất. Cụ thể, các kết nối TCP có ba pha:

Thiết lập kết nối. Có 3 bước :

- Client yêu cầu mở cổng dịch vụ bằng cách gửi gói tin SYN (gói tin TCP) tới server, trong gói tin này, tham số sequence number được gán cho một giá trị ngẫu nhiên X.

- Server hồi đáp bằng cách gửi lại phía client bản tin SYN-ACK, trong gói tin này, tham số acknowledgment number được gán giá trị bằng X + 1, tham số sequence number được gán ngẫu nhiên một giá trị Y.

- Để hoàn tất quá trình bắt tay ba bước, client tiếp tục gửi tới server bản tin ACK, trong bản tin này, tham số sequence number được gán cho giá trị bằng X + 1 còn tham số acknowledgment number được gán giá trị bằng Y + 1.

Truyền dữ liệu. Đặc điểm cơ bản của truyền dữ liệu TCP:

- Truyền dữ liệu không lỗi (do có cơ chế sửa lỗi/truyền lại).

- Truyền các gói dữ liệu theo đúng thứ tự.

- Truyền lại các gói dữ liệu mất trên đường truyền.

- Loại bỏ các gói dữ liệu trùng lặp.

- Cơ chế hạn chế tắc nghẽn đường truyền.

Kết thúc kết nối:

Để kết thúc kết nối hai bên sử dụng quá trình bắt tay 4 bước và chiều của kết nối kết thúc độc lập với nhau. Khi một bên muốn kết thúc, nó gửi đi một gói tin FIN và bên kia gửi lại tin báo nhận ACK. Vì vậy, một quá trình kết thúc tiêu biểu sẽ có 2 cặp gói tin trao đổi. Một kết nối có thể tồn tại ở dạng "nửa mở": một bên đã kết thúc gửi dữ liệu nên chỉ nhận thông tin, bên kia vẫn tiếp tục gửi.

**Websocket**

WebSoket là công nghệ hỗ trợ giao tiếp hai chiều giữa client và server bằng cách sử dụng một TCP socket để tạo một kết nối liên tục, hiệu quả và ít tốn kém. Mặc dù được thiết kế để chuyên sử dụng cho các ứng dụng web, lập trình viên vẫn có thể đưa chúng vào bất kì loại ứng dụng nào.

WebSockets mới xuất hiện trong HTML5, cho phép các kênh giao tiếp song song hai chiều và hiện đã được hỗ trợ trong nhiều trình duyệt. Kết nối được mở thông qua một HTTP request (yêu cầu HTTP), với những header đặc biệt thông báo cho Server (có hỗ trợ) chuyển sang kết nối Websocket. Kết nối này được duy trì để bạn có thể gởi và nhận dữ liệu một cách liên tục, không đứt quãng, và không cần bất kỳ HTTP header (overhead) nào nữa.

Websocket hỗ trợ cho các trình duyệt phổ biến hiện nay như: Google Chrome, Microsoft Edge, Internet Explorer, Firefox, Safari và Opera.

Ở đề tài này ta sẽ xây dựng một Server Websocket để thực hiện các tác vụ thời gian thực như Ứng dụng điện thoại, trình duyệt Web. Và ESP32 có thể kết nối trực tiếp vào các server này như 1 Websocket Client để tiếp nhận, hoặc gởi thông tin thông qua Websocket.

**Database**

Cơ sở dữ liệu (database) là một tập hợp những thông tin được tổ chức để dễ dàng trong việc tạo lập, cập nhập và khai thác thông tin. Cơ sở dữ liệu được duy trì dưới dạng một tập hợp các tập tin trong hệ điều hành hay được lưu trữ trong các hệ quản trị cơ sở dữ liệu.

SQL (Structured Query Language) hay ngôn ngữ truy vấn mang tính cấu trúc, là một loại ngôn ngữ máy tính phổ biến để tạo, sửa, và lấy dữ liệu từ một hệ quản trị cơ sở dữ liệu quan hệ. Ngôn ngữ này phát triển là để phục vụ các hệ quản trị cơ sở dữ liệu đối tượng-quan hệ [5].

SQL là cần thiết để:

• Tạo cơ sở dữ liệu, bảng và view mới.

• Để chèn các bản ghi vào trong một cơ sở dữ liệu.

• Để xóa các bản ghi từ một cơ sở dữ liệu.

• Để lấy dữ liệu từ một cơ sở dữ liệu.

Chức năng của SQL

• Với SQL, chúng ta có thể truy vấn Database theo nhiều cách khác nhau, bởi sử dụng các lệnh.

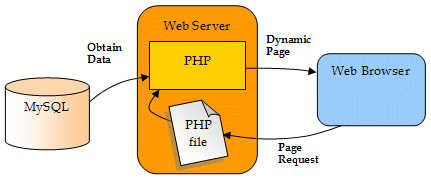
• SQL cho phép người dùng miêu tả dữ liệu.

• SQL cho phép người dùng định nghĩa dữ liệu trong một Database và thao tác nó khi cần thiết.

• Cho phép người dùng tạo, xóa Database và bảng.

• Cho phép người dùng tạo view, Procedure, hàm trong một Database.

• Cho phép người dùng thiết lập quyền truy cập vào bảng, thủ tục và view.

****

. Mô tả cách hoạt động của Web server dựa trên database MySQL và các tệp PHP

HTTP (HyperText Transfer Protocol - Giao thức truyền tải siêu văn bản)

Giao thức này nằm trong tầng Application Layer, được sử dụng để truyền nội dung trang Web từ Web Server đến trình duyệt Web ở Client. Là giao thức Client/Server dùng cho Internet - World Wide Web, HTTP thuộc tầng ứng dụng của bộ giao thức TCP/IP (Các giao thức nền tảng cho Internet). Cơ chế hoạt động chính của HTTP là Request-Response: Web Client sẽ gửi Request đến Web Server, Web Server xử lý và trả về Response cho Web Client.

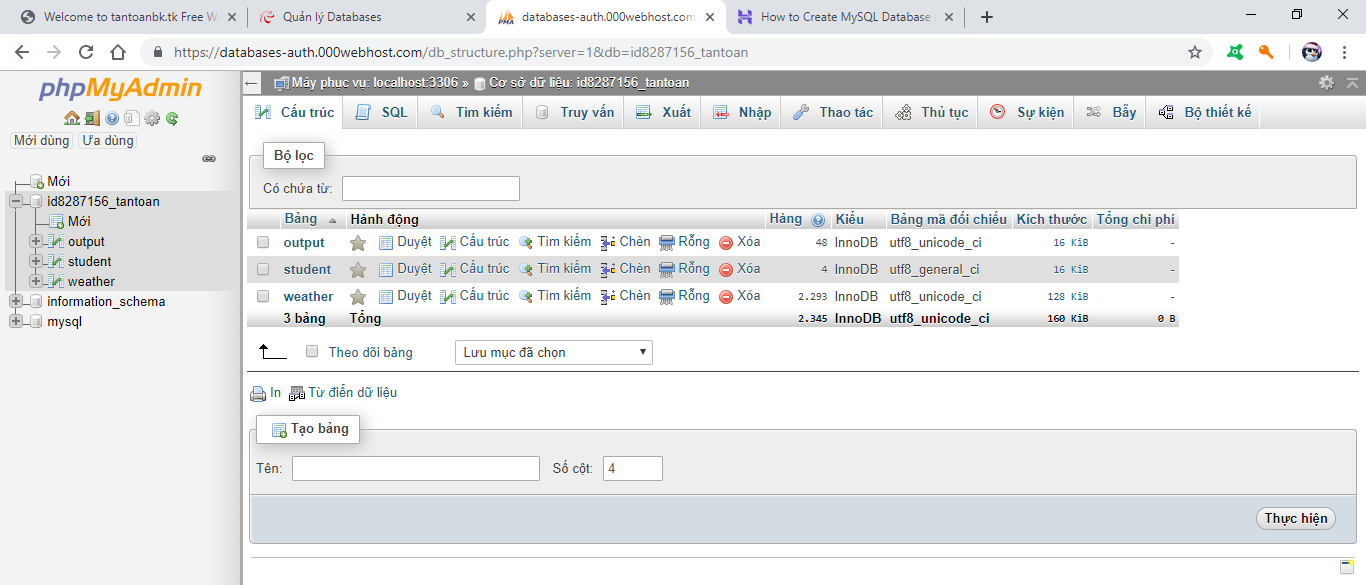
là một giao thức chuẩn của mạng Internet, được dùng để liên hệ thông tin giữa Máy cung cấp dịch vụ (Web server) và Máy sử dụng dịch vụ (Web client) trong mô hình Client/Server dùng cho World Wide Web-WWW, HTTP là một giao thức ứng dụng của bộ giao thức TCP/IP (các giao thức nền tảng cho Internet).

PHP (Hypertext Preprocessor) là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) kịch bản hay một loại [mã lệnh](https://vi.wikipedia.org/w/index.php?title=M%C3%A3_l%E1%BB%87nh&action=edit&redlink=1) chủ yếu được dùng để phát triển các [ứng dụng](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web) viết cho máy chủ, [mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_ngu%E1%BB%93n_m%E1%BB%9F), dùng cho mục đích tổng quát. Nó rất thích hợp với [web](https://vi.wikipedia.org/wiki/Internet) và có thể dễ dàng nhúng vào trang [HTML](https://vi.wikipedia.org/wiki/HTML). Do được tối ưu hóa cho các [ứng dụng web](https://vi.wikipedia.org/wiki/%E1%BB%A8ng_d%E1%BB%A5ng_web), tốc độ nhanh, nhỏ gọn, cú pháp giống [C](https://vi.wikipedia.org/wiki/C_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)) và [Java](https://vi.wikipedia.org/wiki/Java_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh) web phổ biến nhất thế giới.

**Tạo MySQL Database**

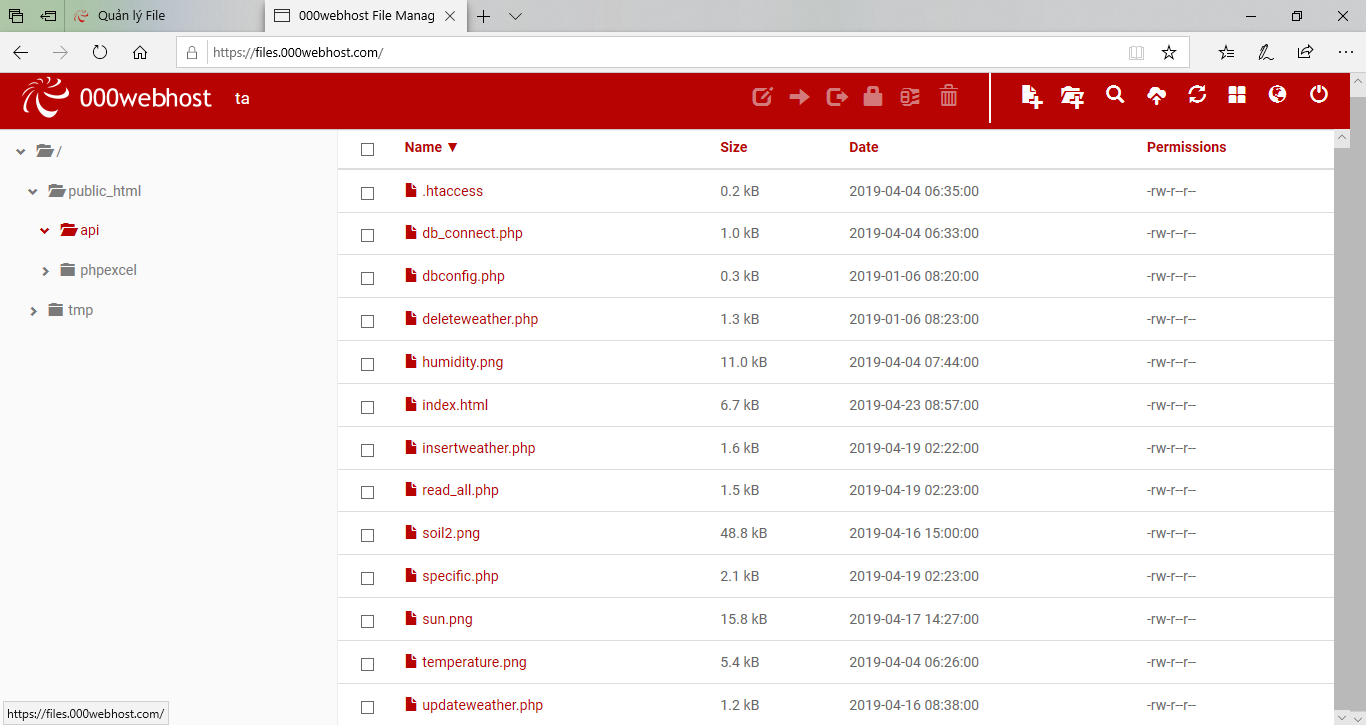
Ở đề tài này ta sử dụng dịch vụ của [www.000webhost.com](http://www.000webhost.com) để tạo database.

Tiếp tục tạo các bảng “output”, “student”, “weather” để phục vụ cho việc lưu trữ dữ liệu. Bảng “output” chứa dữ liệu dự báo, bảng “student” chứa thông tin điều khiển thiết bị, bảng “weather” chứa dữ liệu cảm biến thu thập được.

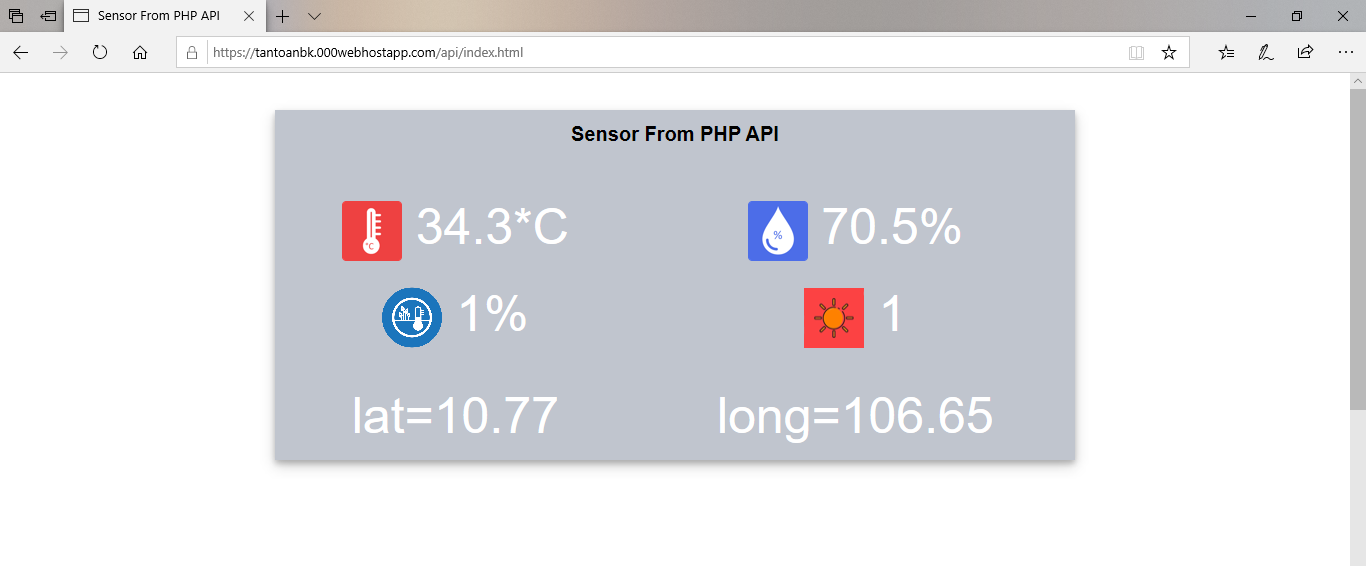
****

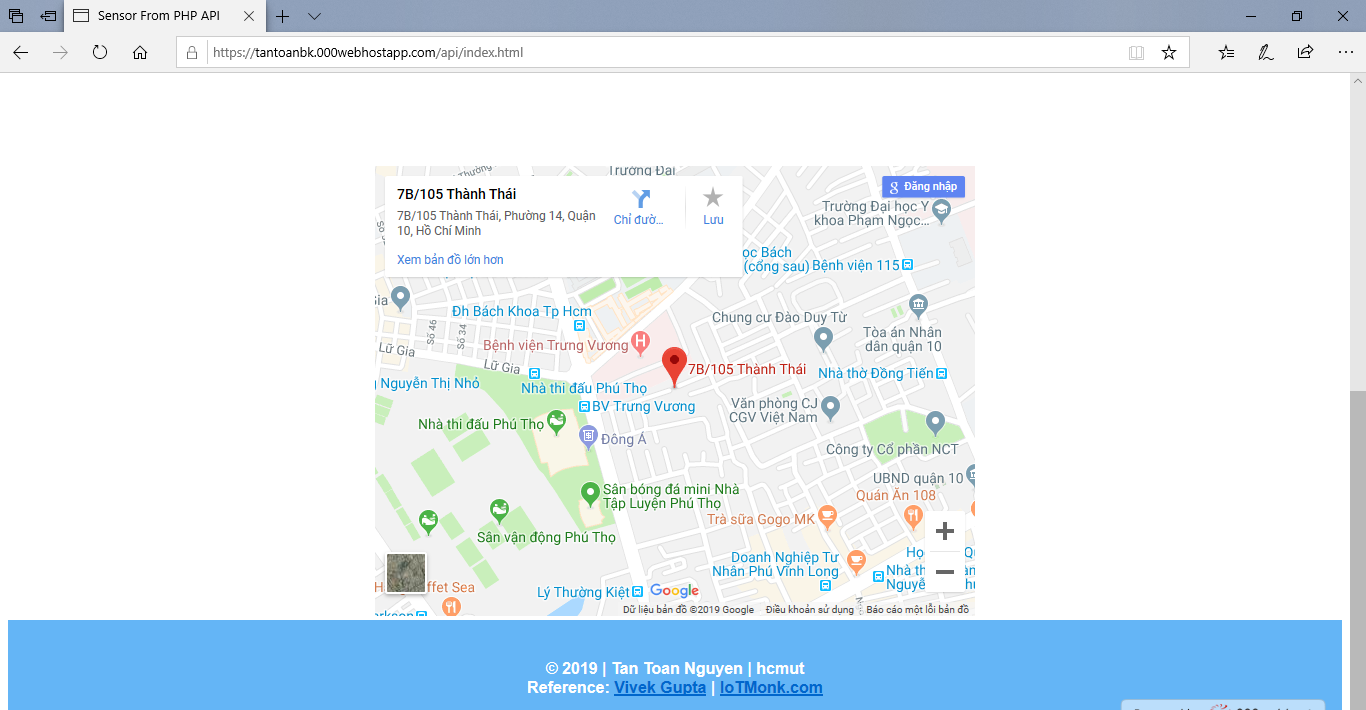
**Xây dựng website**

Tạo file kết nối web với database: ‘dbconfig.php’ và ‘db\_connect.php’

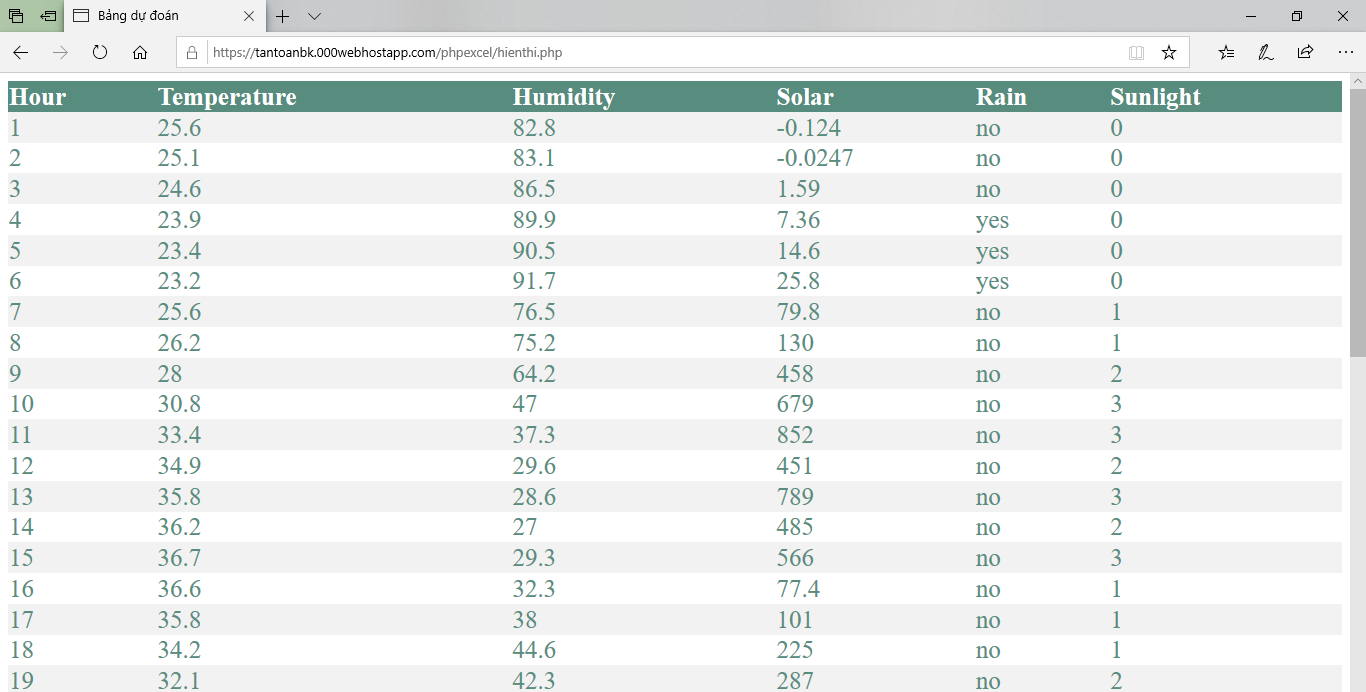


File “insertweather.php” dùng để chèn thêm dữ liệu cảm biến thu thập được vào database “weather”. File “read\_all.php” dùng để đọc bảng “weather” trên website dưới dạng chuỗi. File “index.html” là giao diện hiển thị giá trị cuối cùng của bảng “weather” lên Website, gồm có: nhiệt độ, độ ẩm không khí, độ ẩm đất, ánh nắng mặt trời, tọa độ đo đạc.

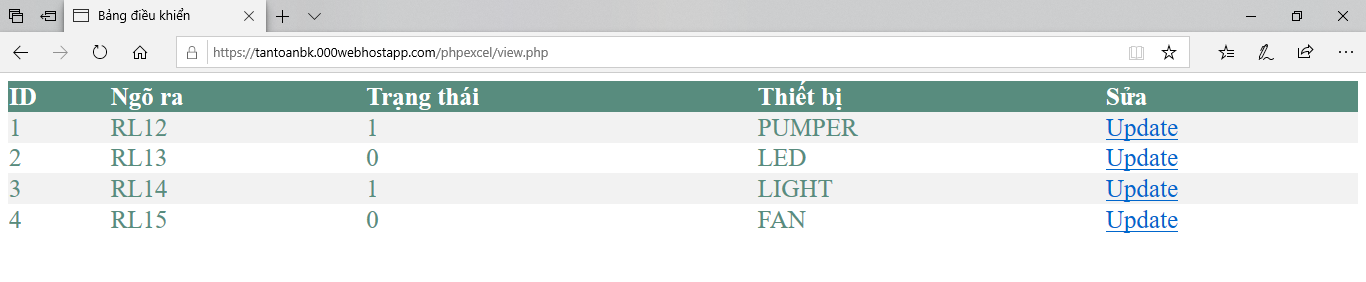


****

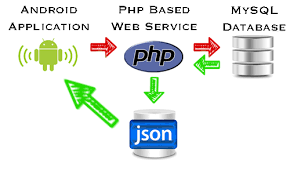
Kế đến là file “hienthi.php” dùng để hiển thị bảng dữ liệu dự báo từ database “output” lên Website



Cuối cùng là file “view.php” tạo giao diện web hiển thị và điều khiển thiết bị



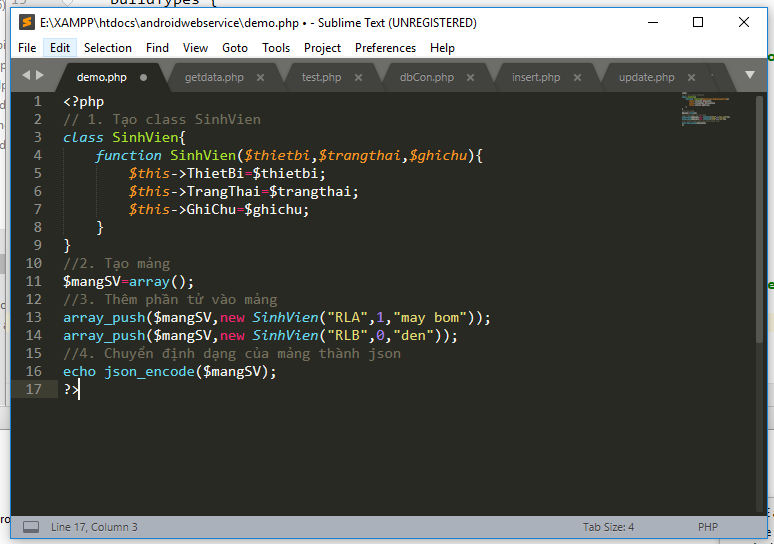
1. **Lập trình Android Webservice dùng để điều khiển thiết bị**

****

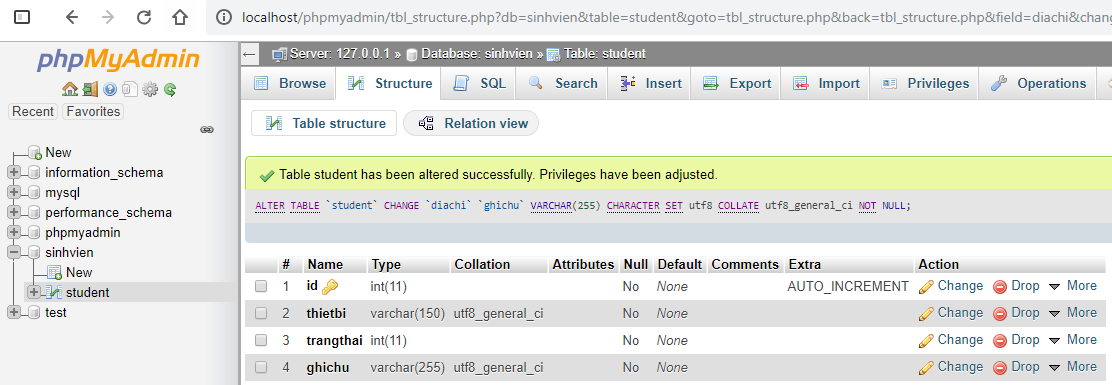
JSON (JavaScript Object Notation) là một kiểu dữ liệu mở trong [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị". Về cấu trúc, nó mô tả một vật thể bằng cách bọc những vật thể con trong vật thể lớn hơn trong dấu ngoặc nhọn. JSON là một kiểu dữ liệu trung gian, chủ yếu được dùng để vận chuyển thông tin giữa các thành phần của một chương trình.

Webservice khác biệt hơn so với localhost. Localhost chỉ sử dụng được bởi một máy, thiết bị cục bộ; các thiết bị khác không thể tương tác hay cập nhật. Còn đối với Webservice, ta viết bằng PHP, database là MySQL, chạy trên một webserver, dữ liệu trả về App. Khi ta cập nhật cập nhật dữ liệu thì trên server cũng cập nhật theo. Để App đọc được dữ liệu trên server phpMySQL phải thông qua định dạng JSON. Thông qua JSON, ta sẽ viết các lệnh php trên server truy vấn dữ liệu từ database MySQL trả ra định dạng JSON, và từ JSON đưa vào App Android. Ngược lại từ App Android có thể truy vấn lên server, có thể là thêm, xóa, cập nhật dữ liệu (chuỗi, số,…). Dữ liệu dạng online webservice có thể linh động hơn localhost, khi cần cập nhật dữ liệu thì đều có thể thực hiện thông qua Internet.

**Cấu trúc chuyển mã từ PHP sang JSON**

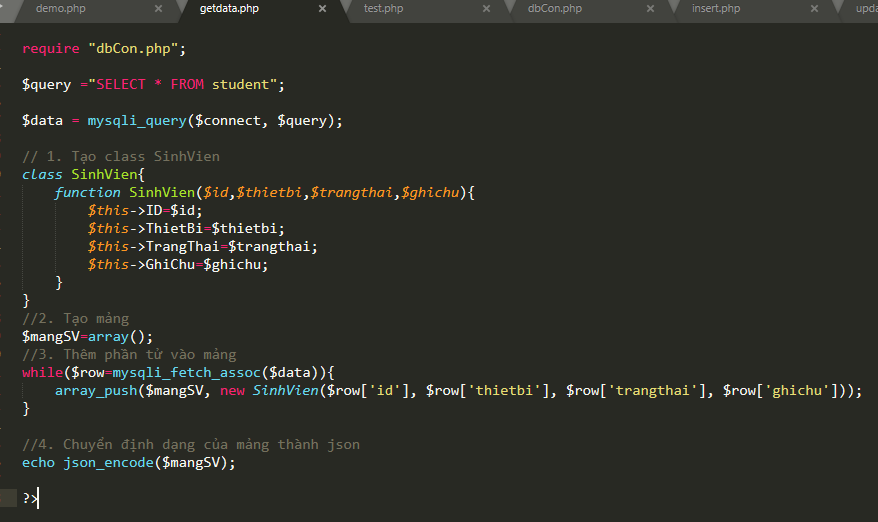


**Tạo database tên “sinhvien” trên localhost**



Hình 10. Bảng "student" dùng để lưu trữ trang thái thiết bị.

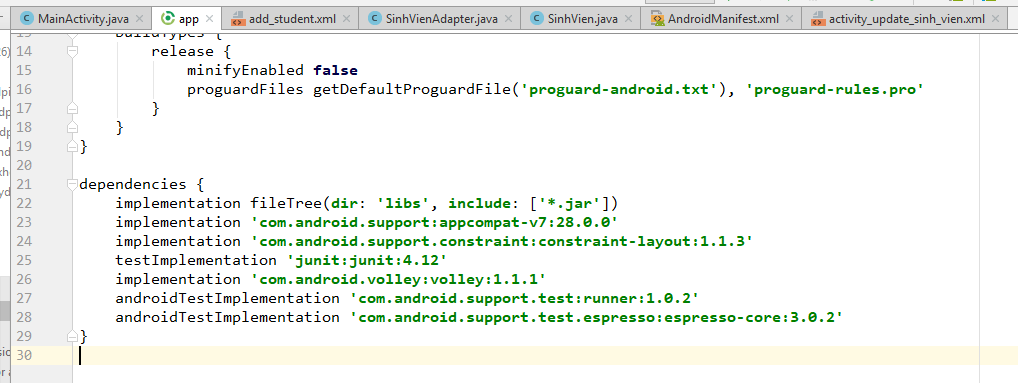
**Tạo file getdata.php để lấy dữ liệu JSON trả về**



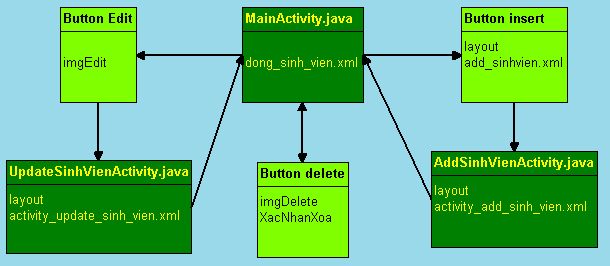
**Tạo project “web service” trong Android Studio**

Sử dụng dòng code: implementation **'com.android.volley:volley:1.1.1'**

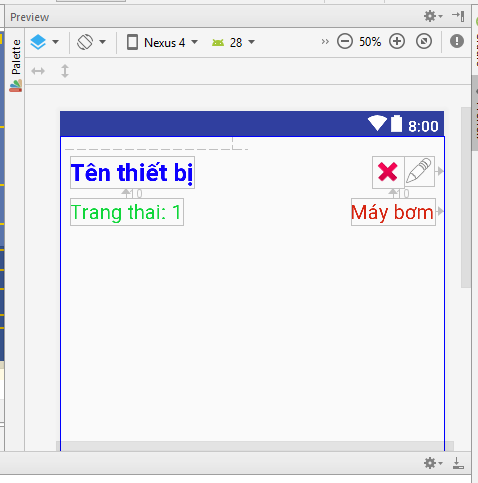
để đọc mảng dữ liệu trả về dạng JSON



**Cấu trúc của ứng dụng Android Webservice**



**Layout “dong\_sinh\_vien.xml” màn hình chính hiển thị danh sách thiết bị**

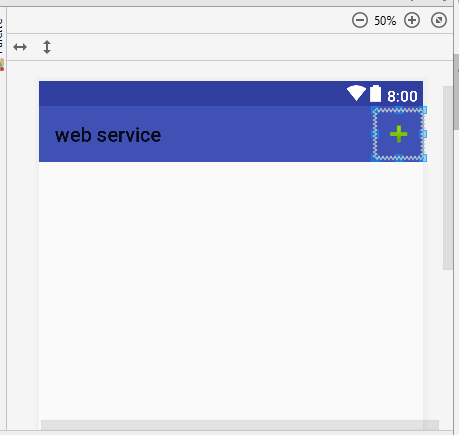


**Tạo file** **“insert.php” dùng để chèn thêm thiết bị vào bảng**

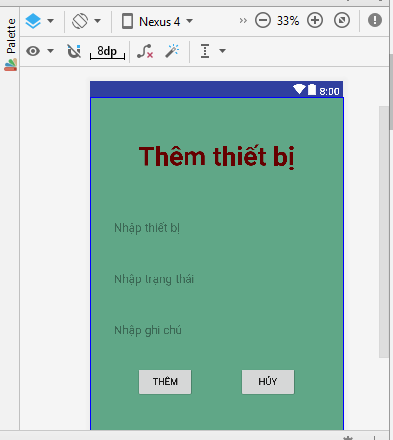
Hàm $\_POST gửi dữ liệu lên

****

**Menu** **“add\_student” để tạo nút nhấn chèn thêm thiết bị**

****

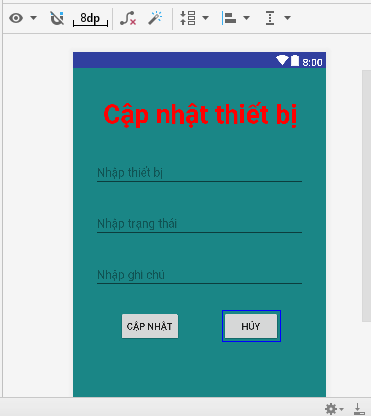
**Layout** **“activity\_add\_sinh\_vien.xml” hiển thị màn hình thêm thiết bị**

****

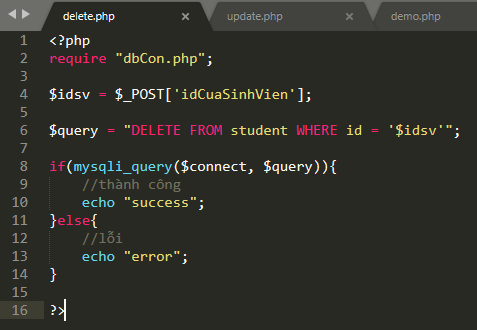
**Nội dung file** **“update.php” dùng để cập nhật thiết bị**

****

**Layout** **“activity\_update\_sinh\_vien.xml” hiển thị màn hình cập nhật thiết bị**

****

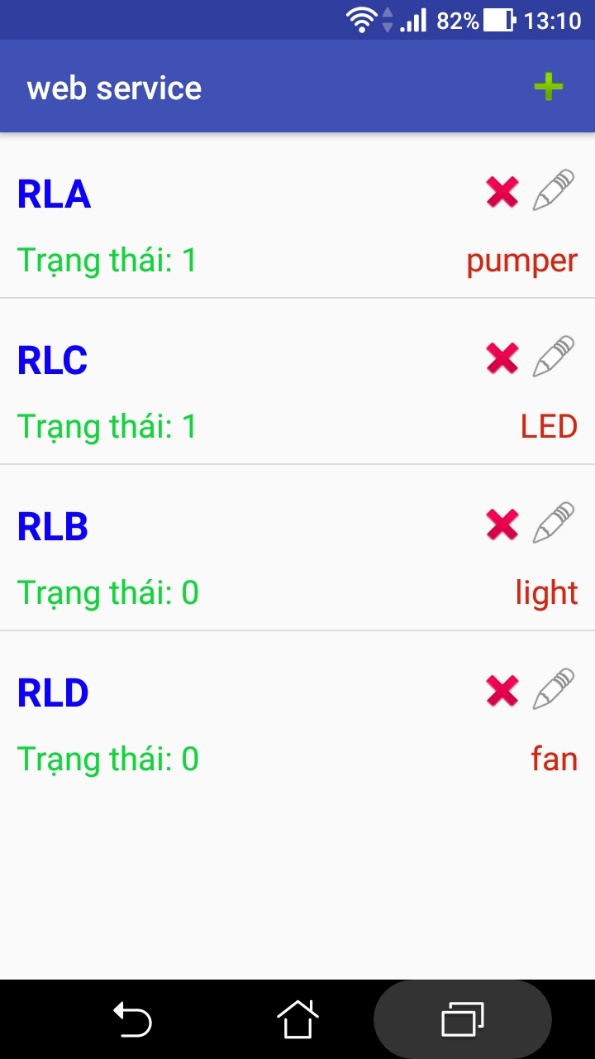
**Nội dung file** **“delete.php” để xóa thiết bị**

****

**Upload file getdata.php lên host**

****

**Cài đặt file apk vào thiết bị Android**

****

# III. KẾT LUẬN

Bộ giao thức TCP/IP (Internet Protocol Suite) là bộ giao thức mà Internet và các mạng máy tính đang sử dụng và chạy trên đó. Nó gồm 2 giao thức chính là TCP (Transmission Control Protocol - Giao thức điều khiển giao vận) và IP (Internet Protocol - Giao thức liên mạng).

Bộ giao thức TCP/IP được coi là một tập hợp các tầng. Mỗi tầng giải quyết một tập các vấn đề liên quan đến việc truyền tải dữ liệu, và cung cấp cho các giao thức tầng cấp trên một dịch vụ được định nghĩa rõ ràng dựa trên việc sử dụng các dịch vụ của các tầng thấp hơn. Hay nói cách khác, các tầng trên gần với người dùng hơn và làm việc với dữ liệu trừu tượng hơn, chúng dựa vào các tầng cấp dưới để biến đổi dữ liệu thành các dạng mà cuối cùng có thể truyền đi một cách vật lý.

**4 tầng của TCP/IP**

* **Tầng ứng dụng** - Application Layer
* **Tầng giao vận** - Transport Layer
* **Tầng mạng** - Internet Layer
* **Tầng liên kết** - Link Layer

App (java) (visual studio)

- 1 tab bật tắt thiết bị

- 1 tab hiển thị nhiệt độ, độ ẩm đo được

- 1 tab dự báo nhiệt độ, độ ẩm ngày hôm sau

Web (php) (sublime text)

- có nút bật tắt thiết bị

- hiển thị bảng nhiệt độ, độ ẩm đo được

- dự báo nhiệt độ, độ ẩm ngày hôm sau

Vẽ board mạch (altium/orcad), có nút nhấn bật tắt thiết bị trực tiếp

Thiết bị: máy bơm, đèn, mái che Kit (tivaC), wifi (esp32)

Cảm biến độ ẩm đất, cảm biến cường độ ánh sáng, cảm biến DHT (nhiệt độ - độ ẩm không khí)

Cảm biến -> kit -> esp -> database -> máy tính -> giải thuật -> database -> web -> app, kit -> thiết bị

- Sử dụng giải thuật Recurrent Neural Network (RNN) mạng neural hồi quy.

- Cảm biến thu thập dữ liệu về nhiệt độ và độ ẩm không khí, thông qua ESP gửi lên database. Máy tính tải dữ liệu về, trải qua các giải thuật xử lý (Python) đưa dự đoán và khuyến nghị (48h tiếp theo) trở lại database.

3) Kết nối các phần và giao diện sử dụng:

- Module ESP32 gửi dữ liệu lên Internet thông qua giao thức MQTT.

- Xây dựng Webserver và giao diện web.

- Tạo ứng dụng trên smartphone.

- Giao diện gồm có: nút nhấn điều khiển thiết bị, bảng hiển thị thông số đo đạc, bảng dự đoán nhiệt độ + độ ẩm + khuyến nghị.

1) Thiết kế phần cứng hệ thống:

- Kit Tiva C điều khiển trung tâm.

- Module ESP32 truyền tải dữ liệu qua wifi.

- Cảm biến: DHT11 nhiệt độ + độ ẩm không khí, cảm biến độ ẩm đất, cảm biến cường độ ánh sáng.

- Relay ngõ ra điều khiển các thiết bị: máy bơm nước, đèn chiếu sáng.

- Vẽ và thực hiện board mạch kết nối các module (có nút bật tắt thiết bị trực tiếp).

4) Vận hành hệ thống: