Assignment II

Introduction to Internet of Things(CS698T)- 2021-22

Indian Institute of Technology,Kanpur, INDIA-208 016

Arpan Kapoor (21111016) arpank21@iitk.ac.in

Piyush Gangle (21111046) piyushg21@iitk.ac.in

November 3, 2021

Contents

1 Irrigation System

1.1 Implementation Design

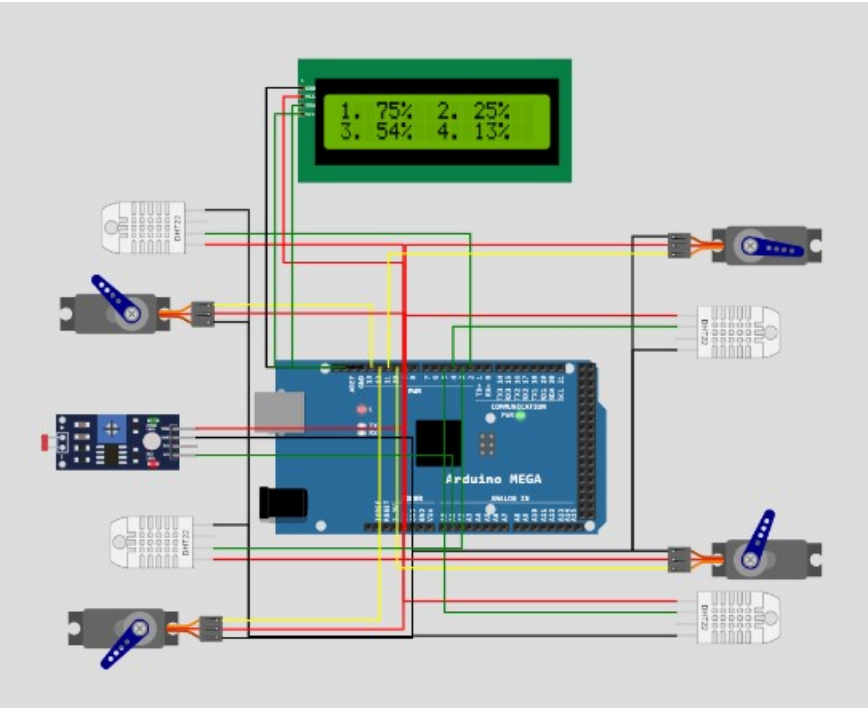

Figure 1: Irrigation System Diagram

1.2 Components Used

- **Arduino Mega Board (1 unit)**: It is powered by the ATmega2560 chip, which has 256K bytes of Flash program memory, 8k bytes of SRAM and 4K bytes of EEPROM. The board features 54 digital pins, 16 analog input pins, and 4 serial ports. It runs at 16MHz.
- **DHT22 (4 units)**: It is a digital humidity and temperature sensor. We can change the temperature and humidity values while the simulation is running by just clicking on the DHT22 sensor and use the temperature and humidity sliders to change the values.
- **Servo motor (4 units)**: It is a standard Micro Servo Motor used to control water flow(%) based on humidity and temperature values.
- **LCD1602 (1 unit)**: An LCD with 2 lines, 16 characters per line used to show water-flow(%) of all four Servo motors.
- **Photoresistor-sensor (1 unit)** : Photoresistor (LDR) sensor module used to detect day and night state of environment. The digital output ("DO") pin goes high when it's dark, and low when there's light. On the physical sensor, we can tweak the small on-board potentiometer to set the threshold. In the simulator, use the "threshold" attribute to set the threshold voltage.

1.3 Connections

Sr no.	Sensor	Connection port
1	DHT sensor-1	2
2	DHT sensor-2	3
3	DHT sensor-3	4
4	DHT sensor-4	5
5	Servo motor-1	13
6	Servo motor-2	12
7	Servo motor-3	11
8	Servo motor-4	10
9	Photoresistor-sensor	A1

2 ML Model

2.1 Introduction

The building block for neural networks are artificial neurons. These are simple computational units that have weighted input signals and produce an output signal using an activation function. A multi-layer neural network contains more than one layer of artificial neurons or nodes.

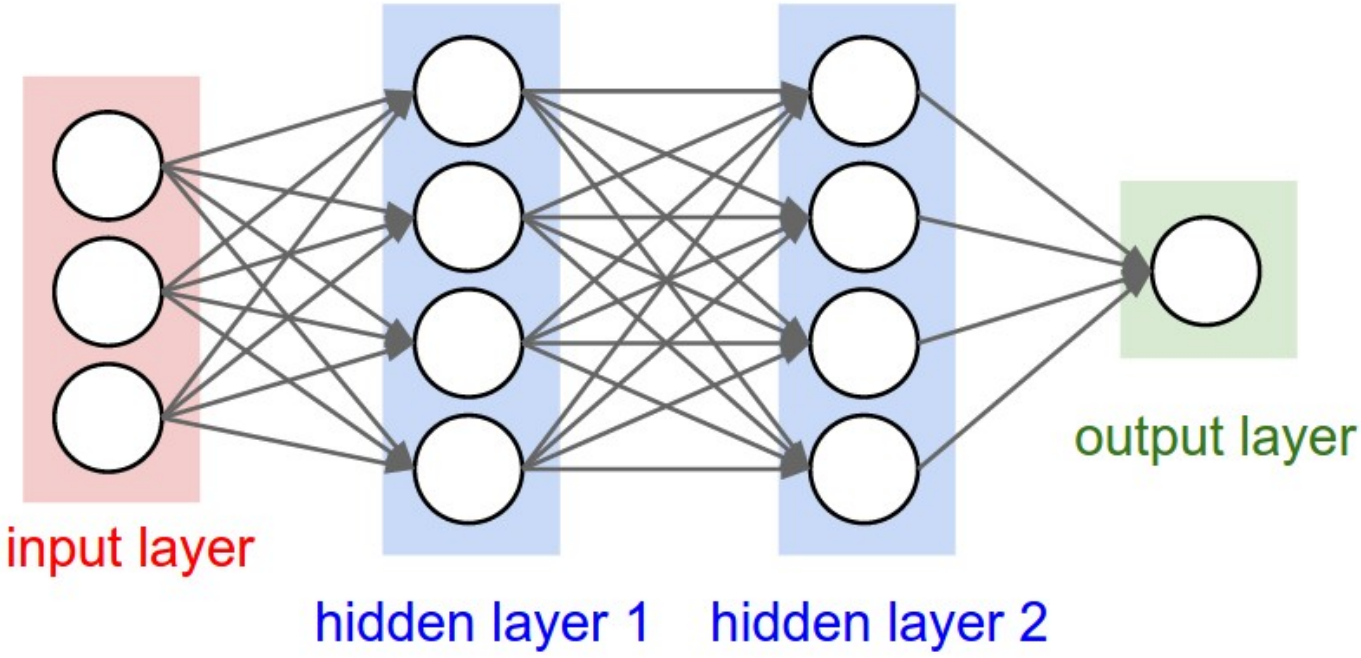

Figure 2: `https://sci2lab.github.io/ml_tutorial/neural_network/`

We have used a 2-layer perceptron model to predict the waterflow(%) based on temperature and humidity values, more specifically we have used 2 hidden layers (input-layer + 2hidden-layer + output-layer).

2.2 Library used

- We have used scikit-learn library to implement our model.
- Functions used from library are
  (1) MLPRegressor
  (2) mean_squared_error
  (3) r2_score
  (4) train_test_split

2.3 MLPRegressor Code Snapshot

```
mlp = MLPRegressor(hidden_layer_sizes=(3, 3), activation="relu", max_iter=20000)

mlp.fit(X_train, y_train)
pred = mlp.predict(X_test)

print("R2 score: ", r2_score(pred, y_test))
print("Mean Squared Error: ", mean_squared_error(pred, y_test))
```

```
R2 score:  0.7086823344328602
Mean Squared Error:  289.17763556760406
```

Figure 3: MLPRegressor

The model has been trained using a python jupyter notebook (.ipynb) and has been added to the submitted zip file. The model coefficients and intercepts are printed in the same notebook and have been used in the simulator to predict the percentage of waterflow.

2.4 Hyper-parameter Setting

1. no. of hidden layer = 2
2. no. of neurons in 1st layer = 3
3. no. of neurons in 2nd layer = 3
4. epochs = 20000
5. activation function = relu

2.5 Evaluation metrics

- R^2 Score : (coefficient of determination) Regression score function. Best possible score is 1.0 and it can be negative (because the model can be arbitrarily worse). A constant model that always predicts the expected value of y, disregarding the input features, would get a score of 0.0.
- R^2 obtained = 0.70
- Mean Squared Error : The mean_squared_error function computes mean square error, a risk metric corresponding to the expected value of the squared (quadratic) error or loss.
- Mean Squared Error = 289.17

3 Link to Simulation

https://wokwi.com/arduino/projects/313762552992498241