Introduction to IoT and Its Industrial Applications (CS667A)
Indian Institute of Technology Kanpur
Assignment 1

QUESTION 1

Member Names: Ankur, Soumith, Manas, Yukkta, Ajeet
Member Emails: ankurk20@iitk.ac.in,battas20@iitk.ac.in, manasg20@iitk.ac.in, yukktas20@iitk.ac.in,
Member Roll Numbers: 200140, 200264, 200554, 201167,
Date: September 16, 2022

Objective

To build an Irrigation system, which contains temperature and humidity sensors that control the water supply in the farm.

Apparatus used

Raspberry Pi 3B, DHT11 Temperature and Humidity Sensor, JHD 204A LCD Display, Jumper Cables, Breadboard and SD Card

Procedure

We first setup the remote client on the raspberry Pi , there we first create the client object and initialise it with the the functions namely on_connect, on_message and on_publish which shall have the following functionality:

remote_client.on_connect()

Once the client is connected to the broker, we shall read and publish the sensor values to the broker on topic dedicated to the communication between remote to local client.

remote_client.on_publish()

Once the values are published we subscribe to the other topic which is dedicated to communication from local client to the remote client, and then wait for any receiving messages.

remote_client.on_message()

On receiving a message we decode the message to Ascii and then display the message on screen and then call the disconnect function to exit the loop_forever() call.
For the client running on the local system (local_client), we create the client object and similarly initiate the function as shown below. Before the creation of the object we train our model so that it is ready to give the predictions before the messages are reciecved

local_client.on_connect()

The local client connects to the MQTT client and subscribes to the message channel and waits for the message to be received i.e waits for the sensors to send the messages.

local_client.on_message()

The client decodes the values i.e. temperature and humidity from the data obtained using ascii decode and calls *send_val*() function to send the data for processing by the machine learning algorithm.

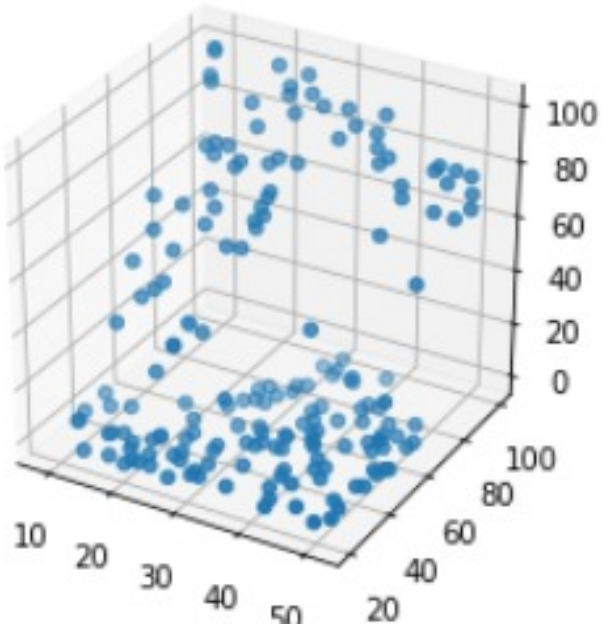
local_client.on_publish()

The client waits for further readings after sending the current results.

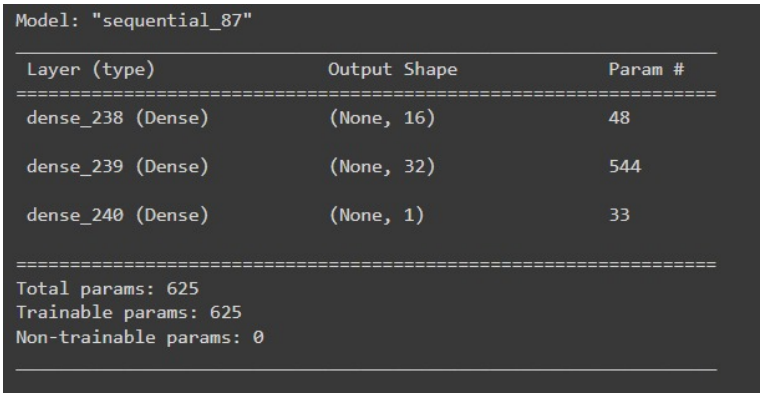
ML Model Architecture

We designed an Artificial Neural network using Tensorflow and tested it on Google Colab. The problem was modelled as a regression problem where we used mean squared error as the loss function .The input data is a 2D vector and ReLu function was used to introduce non linearity in the model. Train/Test data split was taken to be in the ratio 7:3.

The architecture consisted of 2 hidden layers each consisting of 16,32 neurons.Different architectures were experimented but yielded similar results. We trained the model for 100 epochs and used Adam as the optimizer for implementing Gradient Descent.



Input data



Model architecture

Contribution

Ankur kumar(37)
Batta Soumith(18)
Yukkta Seelam(20)
Manas(15)
Ajeet kumar(10)

References

[1] Raspberry Pi 3 Model B+ Datasheet

[2] DHT11 Humidity & Temperature Sensor Datasheet

[3] JHD 204A LCD Display Datasheet