

Air Quality Prediction Using Artificial Neural Networks

Kartik Singh, Shivam Ruhil, Aditya Kumar Singh

Supervisor: Dr. Gaurav Kaushal

ABV-IIITM, Gwalior

November 14, 2022



Table of Contents

- 1 Introduction
- 2 Objective
- 3 Experimental Setup
- 4 Experimental Results
- 5 Performance Estimates
- 6 Utilisation Estimates
- 7 Interface
- 8 Conclusion

Introduction

- Predicting air quality is a complex task due to the dynamic nature, volatility, and high variability in time and space of pollutants and particulates
- Air quality index (AQI) is an indicator created to report air quality, measuring how clean or unhealthy the air is and what associated health effects might be a concern, especially for risk groups.
- Air quality is being monitored for about two decades. This has allowed a better understanding of the changes in air pollution in response to particular activities and government regulations



Objective

- To train an ANN (Artificial Neural Network) model for prediction of AQI of a place.
- Defining the required hardware configuration for the same in the industry.

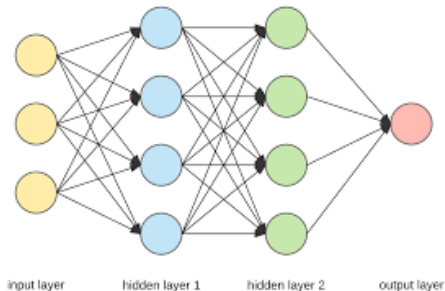


Figure 2: Artificial Neural Network

Experimental setup

Dataset: **MISCA**(Self Made)

- Dataset contains the data of **8** Features which are Average Temperature, Max Temperature, Min Temperature, Atmospheric Pressure, Humidity, Wind Speed, Wind Direction and Average wind speed.
- **Input Layer** have **128 units** ,**3 Hidden Layer** with **256** units each and **Output Layer** layer having **1 units**.
- The **3 Densely connected layers** performing computation and push the output forward with activation function set as 'ReLU' for all the layers..
- Output layer contains **1 neuron**, and is wrapped with 'linear' activation function.

Model Function

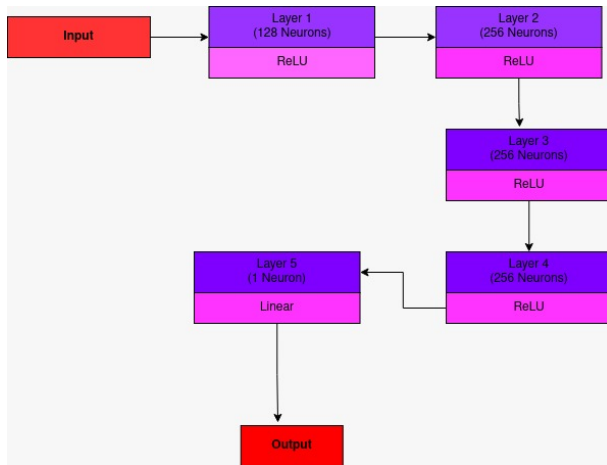


Figure 3: Technical Flowchart

Experimental Setup

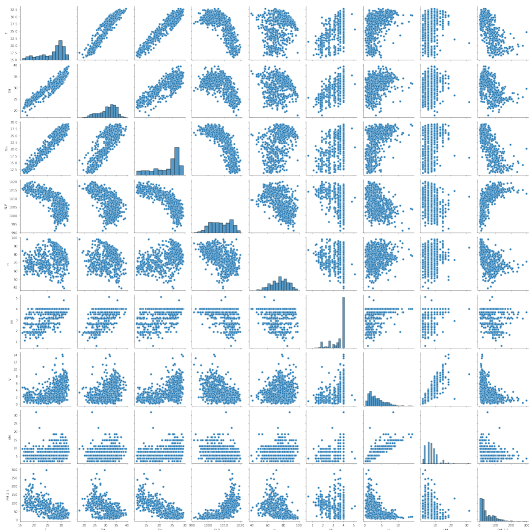


Figure 4: Pairplot of features

Experimental Setup

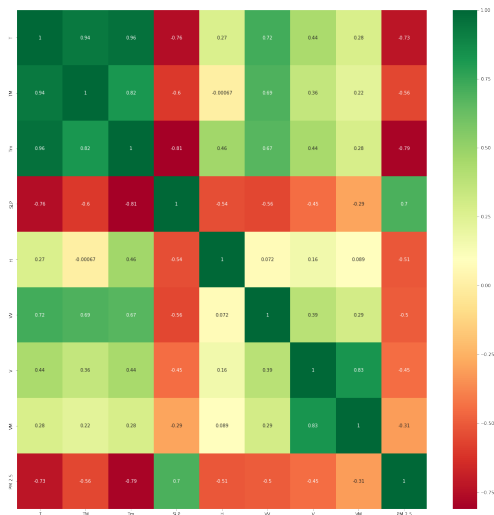


Figure 5: Coorelation Heatmap

Experimental Setup

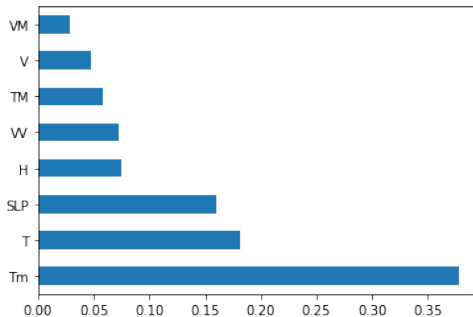


Figure 6: Graph of feature importances

Experimental result

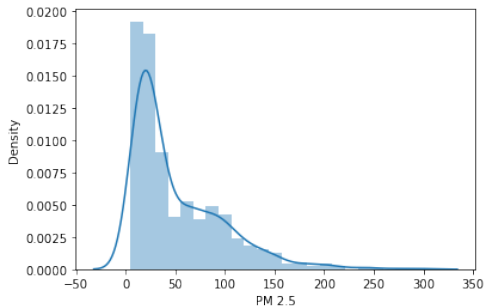


Figure 7: Distribution of Output Feature (PM_{2.5})

Experimental result

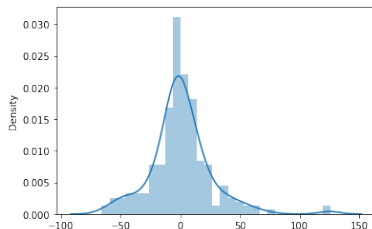


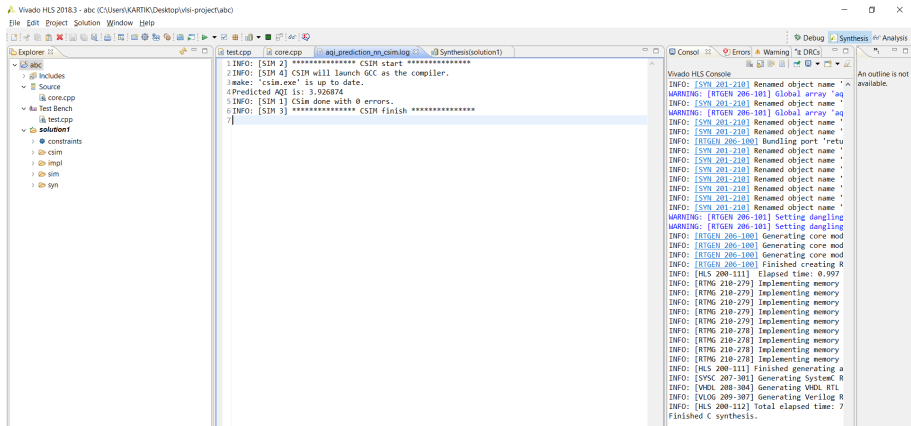
Figure 8: Distribution of Error

```
In [35]: from sklearn import metrics
print('MAE:', metrics.mean_absolute_error(y_test, prediction))
print('MSE:', metrics.mean_squared_error(y_test, prediction))
print('RMSE:', np.sqrt(metrics.mean_squared_error(y_test, prediction)))
```

```
MAE: 17.256978376003808
MSE: 652.3202964224939
RMSE: 25.540561787527185
```

Figure 9: Model Evaluation Results

Experimental result



The screenshot displays the Vivado HLS 2018.3 IDE. The left pane shows the project explorer with files like `core.cpp`, `test.cpp`, and `syn`. The center pane shows the source code of `test.cpp`, which includes a C++ program that simulates a CSM (Circuit Simulation Model) and prints the AQI (Air Quality Index) and the number of errors. The right pane shows the console output, which contains a series of messages from the synthesis process, including warnings and information about the generated code and the final C synthesis.

```
1 test.cpp | core.cpp | aqi_prediction_m_csim.log | Synthesis(solution1)
2 INFO: [SIM 2] ***** CSM start *****
3 INFO: [SIM 4] CSM will launch GCC as the compiler.
4 make: 'csim.exe' is up to date.
5 Predicted AQI is: 3.926874
6 INFO: [SIM 1] CSim done with 0 errors.
7 INFO: [SIM 3] ***** CSM finish *****
8 ]

Vivado HLS Console
INFO: [SYN 201-210] Renamed object name '
WARNING: [RTGEN 206-101] Global array 'aq
INFO: [SYN 201-210] Renamed object name '
WARNING: [RTGEN 206-101] Global array 'aq
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [RTGEN 206-100] Bundling port 'retu
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
INFO: [SYN 201-210] Renamed object name '
WARNING: [RTGEN 206-101] Setting dangling
WARNING: [RTGEN 206-101] Setting dangling
INFO: [RTGEN 206-100] Generating core mod
INFO: [RTGEN 206-100] Generating core mod
INFO: [RTGEN 206-100] Generating core mod
INFO: [RTGEN 206-100] Finished creating R
INFO: [HLS 200-111] Elapsed time: 0.997
INFO: [RTMG 210-279] Implementing memory
INFO: [RTMG 210-279] Implementing memory
INFO: [RTMG 210-279] Implementing memory
INFO: [RTMG 210-279] Implementing memory
INFO: [RTMG 210-279] Implementing memory
INFO: [RTMG 210-278] Implementing memory
INFO: [RTMG 210-278] Implementing memory
INFO: [RTMG 210-278] Implementing memory
INFO: [RTMG 210-278] Implementing memory
INFO: [HLS 200-111] Finished generating a
INFO: [SVSC 207-301] Generating SystemC R
INFO: [VHDL 208-304] Generating VHDL RTL
INFO: [VLOG 209-307] Generating Verilog R
INFO: [HLS 200-112] Total elapsed time: 7
Finished C synthesis.
```

Figure 10: Output in Vivado HLS

Performance Estimates

- Timings

| Clock | Target | Estimated | Uncertainty |
|--------|--------|-----------|-------------|
| ap_clk | 10.00 | 7.796 | 1.25 |

Table 1

- Latency

| Latency | | Interval | | Type |
|---------|---------|----------|---------|------|
| min | max | min | max | |
| 2160157 | 2160157 | 2160157 | 2160157 | none |

Table 2

Utilisation Estimates

| Name | BRAM_18K | DSP48E | FF | LUT |
|-----------------|----------|--------|-------|------|
| DSP | - | - | - | - |
| Expression | - | - | 0 | 546 |
| FIFO | - | - | - | - |
| Instance | 4 | 15 | 1709 | 3638 |
| Memory | 1285 | - | 0 | 0 |
| Multiplexer | - | - | - | 590 |
| Register | - | - | 558 | - |
| Total | 1289 | 15 | 2267 | 4774 |
| Available | 40 | 40 | 16000 | 8000 |
| Utilization (%) | 3222 | 37 | 14 | 59 |

Table 3

Interface

| RTL Ports | Dir | Bits | Protocol | Source Object | C Type |
|------------------------|-----|------|------------|-------------------|--------------|
| s_axi_CRTL_BUS_AWVALID | in | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_AWREADY | out | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_AWADDR | in | 5 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_WVALID | in | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_WREADY | out | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_WDATA | in | 32 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_WSTRB | in | 4 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_ARVALID | in | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_ARREADY | out | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_ARADDR | in | 5 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_RVALID | out | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_RREADY | in | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_RDATA | out | 32 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_RRESP | out | 2 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_BVALID | out | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_BREADY | in | 1 | s_axi | CRTL_BUS | scalar |
| s_axi_CRTL_BUS_BRESP | out | 2 | s_axi | CRTL_BUS | scalar |
| ap_clk | in | 1 | ap_ctrl_hs | aqi_prediction_nn | return value |
| ap_rst_n | in | 1 | ap_ctrl_hs | aqi_prediction_nn | return value |
| interrupt | out | 1 | ap_ctrl_hs | aqi_prediction_nn | return value |
| X_Addr_A | out | 32 | bram | X | array |
| X_EN_A | out | 1 | bram | X | array |
| X_WEN_A | out | 4 | bram | X | array |
| X_Din_A | out | 32 | bram | X | array |
| X_Dout_A | in | 32 | bram | X | array |
| X_Clk_A | out | 1 | bram | X | array |
| X_Rst_A | out | 1 | bram | X | array |

Table 4

Experimental conclusion

Following are the conclusions drawn from the experiments:

- 1) From making of the dataset to training and testing in real time , the model have performed reasonably well.
- 2) We have also defined for the necessary hardware configurations, its utilisation and the interface it provides.

Thank You