ARTIFICIAL NEURAL NETWORK FOR AUTOMATED GAS SENSOR CALIBRATION

[1]DEEPAK P, [2]KSHITIJ SHRIVASTAVA, [3]PRATHIK K, [4]GAUTHAM GANESH, [5]PUNEET S, [6]VIJAY MISHRA

Centre for Nano Science and Engineering, Indian Institute of Science,
Bangalore, Karnataka, India

Abstract— One of the major problem in the calibration of solid state gas sensor is the accuracy of the calculated concentration values from the voltages measured. The customary method employed till date is through the regression analysis of the dataset. The best fit equation obtained after plotting the voltage vs concentration values can be programmed in a microcontroller to attain the concentration values. These results cannot be relied upon as they were less accurate. Instead, an Artificial Neural Network is created, which will learn the dataset and provide accurate results for any unknown input fed to it within the range of the dataset. The results that were obtained show that artificial neural network provides maximum accuracy in determining the analyte concentrations. Hence, this method ensures lower calibration cost and very high accuracy.

Keywords— Artificial Neural Network, Backward Propagation, Forward Propagation, Gas Calibration, Gas Sensors, Gradient Descent, Layers, Weights.

I. INTRODUCTION

Gas sensor calibration is the process of converting the output generated by the gas sensor upon detecting a gas into a meaningful concentration value. The output signal is generally in the form of voltage. Sensor calibration is an absolute necessity to implement a relation between sensor output and concentration level of the required gas. These sensors need to be calibrated and periodically. One of the most common methods of gas calibration is generating a best fit equation that correlates the sensor output to the concentration level of the gas. This is achieved by exposing the sensor to known levels of concentration of a particular gas for a stipulated period of time and noting the vales it gives. Then a scatter plot is generated with x-axis as concentration levels and y-axis as sensor output. A best fit linear curve is then added in such a way that it passes through as many points on the scatter plot as possible [1]. This equation is then fed into the microcontroller board and all concentration values are calculated with respect to this equation. It is to be noted that the best fit equation is only an estimation and hence, high levels of accuracy cannot be expected from it.Artificial Neural Networks is a model inspired by neuron network of human brain. Just like human CNS (Central Nervous System), it learns and functions by use of large number of inputs that is provided to it. With implementation of artificial neural network for sensor calibration, the error is very minimal and very high levels of accuracy is achieved.[2] The paper shows the application of the artificial neural network to the calibration of the solid state sensors used for gas analysis.

II. ARTIFICIAL NEURAL NETWORK

A neural network comprises of three layers:-

Input Layer – This is the part of ANN where input is fed to the network.
Output Layer – This is the place where we obtain the output that is calculated by the network.
Hidden Layers – These are the transition layers between the input and output layers where various calculation occurs which leads to the generation of output.
Transfer functions are used to convert the given input values to values ranging from zero to one.
Each layer contains a specific number of values. Each value is called a *node.*Apart from these one of the important components of a neural network are *weights*. Weights are numbers which are multiplied with every node of a particular layer to form the next layer. The layers and the nodes both are represented in a form matrix. Matrixmultiplication of a layer matrix to its assigned weight matrix leads to formation of next layer.
*Cost function*is a mathematical equation relating the actual output to the calculated output. The cost function is directly proportional to the error. Therefore to minimize the error, cost function has to be minimized.

III. NEURAL NETWORK DESIGN

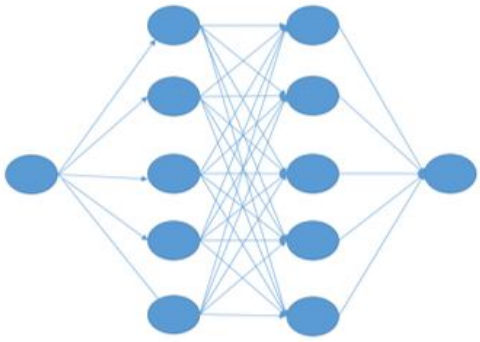

Figure 1: Neural Network Architecture

The neural network used in the project has 4 layers. 1 input layer, 1 output layer and 2 hidden layers. Input and output layers have one node each, while both hidden layers have 5 nodes. In addition to these, every neural network uses one extra node in input and hidden layers with value 1.This node is called the bias node. The value of this bias is kept constant. The bias is used to keep a neutral value in every hidden layer which avoid any overfitting of values.

IV. LEARNING ALGORITHM

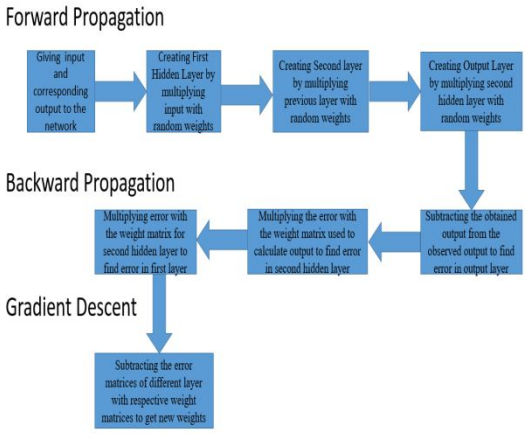

Figure 2: Neural Network Learning Algorithm

The flow of the neural network consists of two parts, the forward propagation and backward propagation. During forward propagation, the values of each node of every layer is converted using a transfer function. The output is then generated by multiplying the converted node values with the current values of weights. In back propagation, the generated output is compared with the actual output and the error in each layer is calculated and minimized. For the purpose of error minimization, gradient descent is used. Here, the error in each weight is subtracted from the corresponding weights and the new weights are generated. After the new weights are generated, the forward propagation is ran again to give a new output. This output will be closer to the actual output than the previously generated output. This whole process run in a loop until the generated output value reaches very close to the actual output value.The learning algorithm is implemented in a python code to give desirable results.

V. MATHEMATICAL EQUATIONS

The artificial neural network works by minimizing the cost function .The cost function for the sigmoid transfer function is mathematically given as:

$J(\theta)=-1/m[\Sigma y^{(i)*}\log h_\theta(x^{(i)})+(1-y^{(i)})*\log(1-h_\theta(x^{(i)}))]$

m - no. of dataset given to the neural net also known as the *training set*

y – Desired output

h(x) – Calculated output

J(θ) – Cost function

5a. Forward Propagation

$a_1^{(2)}=g(\theta_{10}^{(1)}x_0+\theta_{11}^{(1)}x_{1+}\theta_{12}^{(1)}x_{2+}\theta_{13}^{(1)}x_3)$

$a_2^{(2)}=g(\theta_{20}^{(1)}x_0+\theta_{21}^{(1)}x_{1+}\theta_{22}^{(1)}x_{2+}\theta_{23}^{(1)}x_3)$

$a_3^{(2)}=g(\theta_{30}^{(1)}x_0+\theta_{31}^{(1)}x_{1+}\theta_{32}^{(1)}x_{2+}\theta_{33}^{(1)}x_3)$

$h_\theta(x)=a_1^{(3)}=g(\theta_{10}^{(2)}a_0^{(2)}+\theta_{11}^{(2)}a_1^{(2)}+\theta_{12}^{(2)}a_2^{(2)}+\theta_{13}^{(2)}a_3^{(2)})$

$a_i^{(j)}$ = activation of unit i in layer j

$\theta^{(j)}$=matrix of weights controlling function mapping from layer j to layer j+1

g(z) = 1/(1+e^-z), also known as the sigmoid function. This transfer function is used for converting the input.

5b. Backward Propagation

$\delta^{(3)}=(\theta^{(3)})^T\delta^{(4)}.*g'(z^{(3)})$

$\delta^{(2)}=(\theta^{(2)})^T\delta^{(3)}.*g'(z^{(2)})$

$\Delta_{ij}^{(l)}:=\Delta_{ij}^{(l)}:+a_j^{(l)}*\delta_i^{(l+1)}$

$\delta^{(l)}$ = error in the layer i

$\Delta^{(l)}$ = Error in weight matrix for layer l

5c. Gradient Descent

Gradient descent is a technique of minimizing the cost function. In this technique the parameters defining the cost function are changed periodically until the cost function is absolutely minimized. In the case of neural network, these parameters are the weights assigned for each layer. The mathematical representation of gradient descent is:

$\Theta = \Theta - (\alpha/m)*d(J(\Theta))/d\Theta$

In the neural network this differential term is given by Δ, so gradient descent for neural net is

$\Theta = \Theta - (\alpha/m) * \Delta$

α – Learning Rate = 0.02(for this particular project)

VI. APPLICATION

In real-time, running the complete neural network code will be very cumbersome at times. This will result in delayed results. Hence, the actual neural network is run on a very high processing system and the final weights corresponding to each data point of the dataset are stored in an excel sheet. Whenever an input is given, the corresponding weights are used to run the forward propagation and obtain the desired output. These updated weights can be used for various purposes. In the recent advancements of clinical diagnosis, breathe analyzer, which use solid state gas sensors for detectingvarious diseases and conditions can employ this neural network for getting desired levels of biomarkers in breathe. Various fat burn devices available in the market detect the acetone levels in the human breathe. These sensors can be calibrated more accurately using these neural nets.

VII. RESULTS AND CONCLUSION

7a. Results obtained from normal calibration technique

Table 1: Results from calibration of CO sensor using best fit equation technique

Input(ppm)	Desired Output(mV)	Calculated output(mV)	Error	% Error
3	2258.87	2277.886095	19.016095	0.84%
2.5	2243.5	2209.448052	-34.051948	1.5%
2	2180.5	2125.686657	-54.813343	2.51%
1.5	2072.31	2017.699438	-54.610562	2.6%
1	1906.37	1865.5	-40.87	2.14%
0.5	1546.75	1605.313343	58.563343	3.79%

7b. Results obtained from Neural Network

Table 2: Results from calibration of CO sensor using Artificial Neural Network

Input(ppm)	Desired Output(mV)	Calculated output(mV)	Error	% Error
3	2258.87	2258.86999	.00001	Negligible
2.5	2243.5	2243.49999	.00001	Negligible
2	2180.5	2180.49999	.00001	Negligible
1.5	2072.31	2072.310001	.00001	Negligible
1	1906.37	1906.36999	.00001	Negligible
0.5	1546.75	1546.74999	.00001	Negligible

From the above tables' results, it can be concluded that a neural network gives far better results than a best fit equation.

VIII. FUTURE SCOPE

It is also known that temperature, humidity and pressure play vital roles in the determination of the concentration levels of a gas. The advantage of using an artificial neural network is that, the pressure, temperature and humidity values can also be fed in as input values by just increasing the number of input nodes. This will then give highly accurate results for varying ranges of the aforementioned factors.

REFERENCES

[1] Carey, W. P., & Yee, S. S. (1992). Calibration of nonlinear solid-state sensor arrays using multivariate regression techniques. *Sensors and Actuators B: Chemical*, *9*(2), 113-122.

[2] Osowski, S., &Brudzewski, K. (1999). Hybrid neural network for gas analysis measuring system. In *Instrumentation and Measurement Technology Conference, 1999. IMTC/99. Proceedings of the 16th IEEE* (Vol. 1, pp. 440-444). IEEE.

★★★