International Journal of Engineering & Technology, 7 (4.22) (2018) 182-185

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Systematic Approach for Designing a Neural Network Using Existing Algorithms to Detect H_2, CH_4, and CO Gases

Maibam Sanju Meetei[1]*, Anil Chamuah[1], Yabom Yuto[1], Aheibam Dinamani Singh[2]

[1]*Dept. of Electronics & Communication Engineering, Rajiv Gandhi University, Arunachal Pradesh, India*
[2]*Dept. of Electronics & Communication Engineering, NERIST, Arunachal Pradesh, India*
**Corresponding author E-mail:maibam.meetei@rgu.ac.in*

Abstract

A neural network has different parameters like weight, bias, activation function and hidden layers. Different algorithms are applied to set the parameters and various normalization techniques applied to the input data also differs the performance of the network. So, it is very important for a designer to design the network by considering the above variables like the number of layers, different normalization techniques, different activation functions and different algorithms. It is very important to optimize all these parameters for better performance.

Keywords: *activation function; hidden layers; MSE; neural network; normalization, training algorithms.*

1. Introduction

The selectivity of the gas sensor is very poor, however, the sensitivity of the gas sensor has a wide range. Thus, a single sensor cannot be used for identification of different gasses. Therefore, an array of sensors has been taken for cross-identification of the different gasses. Different pattern analysis methods like Principle Component Analysis (PCA), Partial Least Square Analysis (PLS), and Artificial Neural Network (ANN)[1][3] have been implemented by different authors. Multi-Layer Perceptron (MLP)[1][4][5] and Radial Basis Function (RBF)[5] are the two commonly used Neural Network Algorithm for pattern recognition.

1.1. Neural Network

An Artificial Neural Network is the modern implementation of the biological neural network model by artificial neurons. In a simple neuron, there are inputs (P), outputs (O), weight (W), one bias (B), one summer and one activation function (F). The mathematical representation of a simple neuron having 'n' inputs:

$$W = [w1\ w2\ w3\ w4\ \ldots\ldots\ldots\ldots\ldots wn]$$
$$P = [p1\ p2\ p3\ p4\ \ldots\ldots\ldots\ldots\ldots\ldots pn]$$
$$O = F(W.P^T+B)$$

Where P^T is the transpose of P.

The performance of the neural network is measured in terms of Mean-Square-Error (MSE), lesser the value of MSE, better in performance.

In Multi-Layer Perceptron, learning process takes place through two phases:

1. Forward Phase: In forward phase, the synaptic weight and the bias of the neurons in the network are fixed and the input signal propagating through the network layer by layer until it reaches the output. The changes are confined to the activation potential and outputs of the neuron in the network.
2. Backward Phase: In backward phase, an error signal has been calculated after comparing the trained output of the network with the target of the network. The error signal is sent back and used for calculating the new weight and bias by different error learning mechanisms. The calculations of the weight and bias are straightforward but it is difficult for hidden layers.

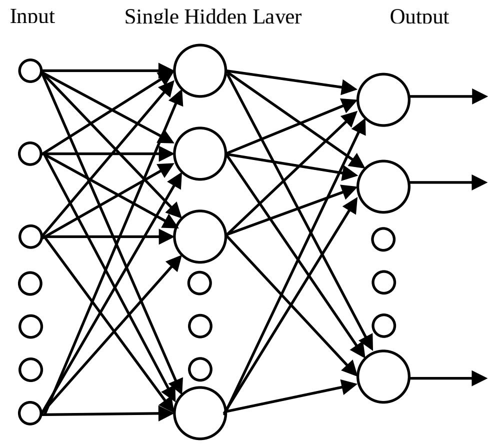

Fig. 1: Architecture with a single hidden layer

1.2. Activation Function

There are generally four types of activation function used in the neural network. They are:

1. Linear Transfer Function: In this function output is linearly proportional to the inputs.
2. Log-Sigmoid Transfer Function: In this function, the output lies between 0 and +1. The output is nonlinearly proportional to the input. This transfer function is generally used in the hidden layer of a multi-layer perceptron neural network.
3. Tan-Sigmoid Transfer Function: In this function, the output lies between -1 and +1. The output is exponentially proportional to the input.

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

4. Hard-Limit Transfer Function: In this function, the output is either 0 or 1. The output is decided by using one threshold value.

1.3. Normalization

The normalization process of the raw data inputs has provided suitable data for the training. Without normalization, the training of the neural network would be slower. There are many types of data normalization techniques [1][2]. Some of the techniques that have been used are tabulated in Table 1.

Table 1: Different normalization formulas

Normalization	Mathematical Formula
Normalization 1	$\text{Norm } A_{xy} = \dfrac{A_{xy} - \min(A_{xy})}{\max(A_{xy}) - \min(A_{xy})}$
Normalization 2	$\text{Norm } A_{xy} = \dfrac{A_{xy}}{\max(A_{xy})}$
Normalization 3	$\text{Norm } A_{xy} = \dfrac{A_{xy} - \text{mean}(A_{xy})}{\text{std}(A_{xy})}$

Where A_{xy} is the raw input matrix with rows 'x' and columns 'y'.

1.4. Network Architecture

A network with a single hidden layer can sufficiently hold when using regular functions (eg. sigmoid) but the number of hidden neurons can be as high as the number of training samples [6]. The purpose of using a second hidden layer is to drastically reduce the total required of hidden nodes [7]. The network performance is depending on the number of nodes in the hidden layer thus network optimization is important [1][8][9]. It was theoretically shown that one hidden layer is enough to approximate any continuous function with arbitrary accuracy for classification problem, provided enough hidden units are available [10]. However, the theoretical result doesn't give a direction how to choose the number of hidden units needed. Also, even if one hidden layer is enough theoretically, in practice more than one hidden layer should be utilized for faster, more efficient problem solving [8].

2. Experimental Setup and Data Collection

2.1. Data Collection Setup

The block diagram of the experimental setup is shown in Fig. 2. The sensor is placed in a closed chamber with an external precise temperature controller. A controlled amount of nitrogen gas and test gas are passed through Mass Flow Controller (MFC) to the sensing chamber. The power is given to the sensor using a high accuracy adjustable DC power supply (Agilent E3631A) and the output voltage is monitored using a high-resolution multimeter (Agilent U1253B).

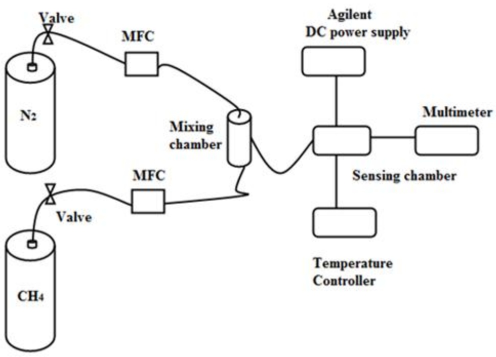

Fig. 2: Block diagram of the Experimental Setup

2.2. NN Design in Matlab

A feed-forward back propagation multilayer neural network trained by Resilient backpropagation has been designed for prediction of gasses as shown in Fig. 3. The network was designed in the neural network toolbox in MATLAB with 10 neurons in the first hidden layer and 12 hidden neurons in the second hidden layer. The activation of the network is used as sigmoidal function and the performance is measured at MSE. During the training process, the network parameters like weight and bias of the neurons are iteratively adjusted to get the lowest MSE.

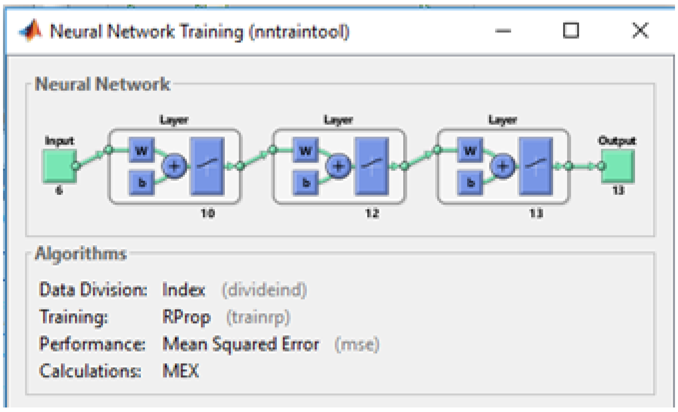

Fig. 3: Design of NN in MATLAB

3. Training Approach

First, Data arrangement, all the training data are divided into two parts, one is for the training set and another is for the simulation to check the network output after training, normalized data by using the standard formula. After normalization, take an arbitrary network with 1 hidden layer having 6 neurons and an output having 13 neurons, the following steps are implemented sequentially:

1. Choose for the best normalization technique.
2. Train the network by varying number of neurons in a hidden layer.
3. Choose the number of neurons giving the best performance.
4. For improvement, adding 1 more hidden layer in the above best performing network.
5. Train the network by varying number of neurons in the added hidden layer.
6. Choose the number of neurons giving the best performance.
7. Train the best performing network with different combinations of activation functions.
8. Choose the best combination of the activation function.
9. Train the same network with different training algorithms.
10. Choose the best training algorithm.

4. Result and Discussion

According to the different network parameters:

1. Normalization: From Table II, the normalized data has shown better performance than the non-normalized data. Among the different normalization techniques, it is observed that the Normalization technique 3 has better performance.

Table II: Normalization Technique vs MSE

Normalization Technique	MSE
Without Normalization	0.0145
Normalization 1	0.0132
Normalization 2	0.0136

Normalization 3	0.0125

2. Number of hidden neurons: The lowest MSE of the single hidden layer network with 19 neurons is 0.00681, and the double hidden layer network with 19 and 18 neurons in the 1st and 2nd hidden layer respectively is 0.00326. Thus, network with double hidden layers shows better performance than a single hidden layer.

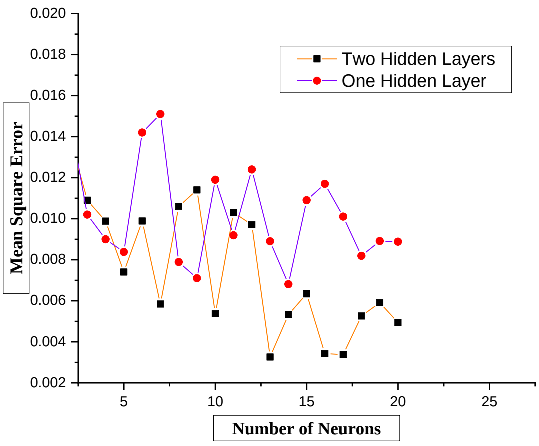

Fig. 4: Performance Curve

3. Activation function: The 27 possible combination of activation function with two hidden layer network is in Table III. Among all the combination, the comparative lower values of MSE found in (i) logsig, logsig, logsig (ii) tansig, logsig, purelin and (iii) purelin, logsig, logsig with MSE 0.00498, 0.00475 and 0.00326 respectively.

Table III: Activation Function vs MSE

Transfer Function	MSE
logsig,logsig,logsig	0.00498
logsig,logsig,tansig	0.01150
logsig,logsig,purelin	0.01130
logsig,tansig,logsig	0.00528
logsig,tansig,tansig	0.01290
logsig,tansig,purelin	0.02330
logsig,purelin,logsig	0.01210
logsig,purelin,tansig	0.03100
logsig,purelin,purelin	0.03120
tansig,logsig,logsig	0.00533
tansig,logsig,tansig	0.01160
tansig,logsig,purelin	0.00475
tansig,tansig,logsig	0.00826
tansig,tansig,tansig	0.00980
tansig,tansig,purelin	0.01310
tansig,purelin,logsig	0.01200
tansig,purelin,tansig	0.03040
tansig,purelin,purelin	0.03490
purelin,logsig,logsig	0.00326
purelin,logsig,tansig	0.01910
purelin,logsig,purelin	0.01470
purelin,tansig,logsig	0.00562
purelin,tansig,tansig	0.01230
purelin,tansig,purelin	0.01980
purelin,purelin,logsig	0.03460
purelin,purelin,tansig	0.05520
purelin,purelin,purelin	0.05540

4. Training algorithms: Training algorithms such as trainrp, traingd, traingdm, traingdx, traincgf, traincgp, trainscg, trainbfg, trainoss and trainlm are simulated and it is found that trainrp has the lowest value of MSE (0.00326).

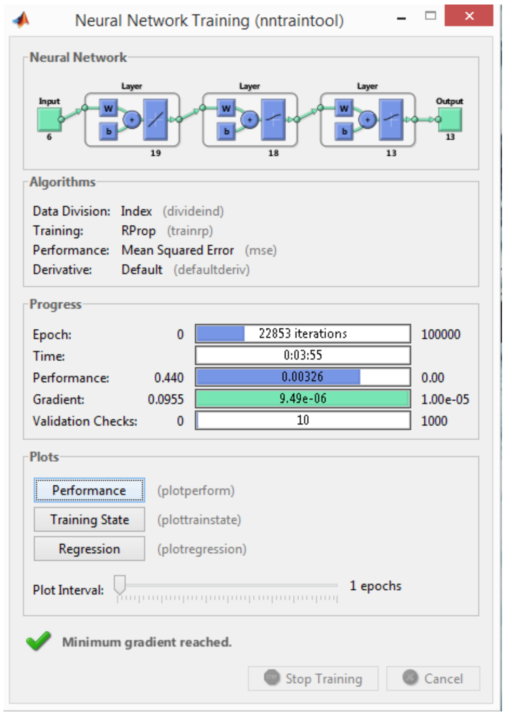

Fig. 5: Simulation Output

The network with input having Normalization-3 (statistical normalization) with 19 neurons in the first hidden layer and 18 neurons in the second hidden layer, and purelin, logsig, logsig combination of activation function has shown better performance among the networks with MSE 0.00326. The simulated output is shown in Fig. 5.

5. Conclusion

From the above observations, it can be concluded that the performance of the network is dependent on various parameters as discussed earlier. In this study, the network with normalization-3 technique, 19 hidden neurons in the 1st hidden layer, 18 hidden neurons in the 2nd hidden layer, purelin, logsig, logsig combination of activation function and trainrp algorithm gave MSE of value 0.00326 which is the best among them. The different type of gases has successfully detected by using the proposed training approach.

References

[1] B. Mondal, M.S. Meetei, J. Das, C. Roy Chaudhuri, H. Saha, “Quantitative recognition of flammable and toxic gases with artificial neural network using metal oxide gas sensors in embedded platform,” *Engineering Science and Technology, an International Journal*, Volume 18, Issue 2, 2015, Pages 229-234.

[2] B. Tudu, B. Kow, N. Bhattacharyya and R. Bandyopadhyay, "Comparison of multivariate normalization techniques as applied to electronic nose based pattern classification for black tea," *3rd International Conference on Sensing Technology, Tainan, 2008*, pp. 254-258.

[3] Dae-Sik Lee, Sang-Woo Ban, Minho Lee and Duk-Dong Lee, "Micro gas sensor array with neural network for recognizing combustible leakage gases," in *IEEE Sensors Journal*, vol. 5, no. 3, pp. 530-536, June 2005.

[4] X. Zhai, A. A. S. Ali, A. Amira and F. Bensaali, "MLP Neural Network Based Gas Classification System on Zynq SoC," in *IEEE Access*, vol. 4, no. , pp. 8138-8146, 2016.

[5] Zhenya Zhang, Hongmei Cheng, Shuguang Zhang, “Research on the Performance of Feed Forward Neural Network Based Temprature Field Identification Model in Intelligent Building”, *IEEE*, 2011.

[6] Guang-Bin Huang, "Learning capability and storage capacity of two-hidden-layer feedforward networks," in *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 274-281, Mar 2003.

[7] D. Stathakis, “How many hidden layers and nodes?”, *International Journal of Remote Sensing*, Vol. 30, No. 8, 20 April 2009, 2133-2147.

[8] P. Gao, C. Chen and S. Qin, "An Optimization Method of Hidden Nodes for Neural Network," *Second International Workshop on Education Technology and Computer Science, Wuhan, 2010*, pp. 53-56.

[9] G. Bebis and M. Georgiopoulos, "Feed-forward neural networks," in *IEEE Potentials*, vol. 13, no. 4, pp. 27-31, Oct.-Nov. 1994.

[10] J. Tsai, J. Chou, and T. Liu, “Tuning the structure and parameters of a Neural Network by using hybrid Taguchi-genetic algorithm,” *IEEE*, Vol. 17, No. 1, Jan 2006.