Neural Network based Control Solution of Air to Fuel ratio for 2NZFE Fuel Injection Engine

Author

Urfa Rasool

2011-NUST-MsPhD-Mech-29

Supervisor

Dr. Imran Akhtar

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY

ISLAMABAD

APRIL, 2014

Neural Network based Control Solution of Air to Fuel ratio for 2NZFE Fuel Injection Engine

Author

Urfa Rasool

2011-NUST-MsPhD-Mech-29

A thesis submitted in partial fulfillment of the requirements for the degree of

M.Sc. Mechanical Engineering

Thesis Supervisor:

Dr.Imran Akhtar

Thesis Supervisor

Signature:____

DEPARTMENT OF MECHANICAL ENGINEERING

COLLEGE OF ELECTRICAL & MECHANICAL ENGINEERING

NATIONAL UNIVERSITY OF SCIENCES AND TECHNOLOGY, ISLAMABAD

APRIL, 2014

Declaration

I certify that research work titled "Neural Network Based Control Solution of Air to Fuel Ratio for 2NZ-FE Fuel Injection Engine" is my own work. A conference paper based on this work has been presented in the 1st International Conference of Abasyn University held on 2-4th April, 2013 for assessment. All the data used in this research is properly acknowledged /referred.

Signature of Student

Urfa Rasool

2011-NUST-MsPhD-Mech29

Copyright Statement

Dedication

I dedicate this work to my parents, who tremendously supported & encouraged me throughout my entire journey without which, it would not be possible for me to sail through this endeavor smoothly.

Acknowledgements

I have profound respect for my supervisors Dr. Imran Akhtar and Dr.Imran Shafi who have been extremely generous with their help and encouragement throughout the course of this work. They are a great source of inspiration for me. I would also like to thank my father for his assistance and advices in different situations. I must not fail to mention, Mr. Abdul Salam Baloch and Mr. Tariq khan who made my visit possible at Toyota EFI Workshop to perform active tests on 2NZ-FE fuel injection engine for collecting required data regarding my research work.

Preface

This research work contributes in controlling air-to-fuel ratio of electronic fuel injection engines by using an artificial intelligence technique known as neural network. In this thesis 2NZ-FE fuel injection engine has been examined and numerical simulations are performed for identification and prediction of air-to-fuel ratio. MATLAB is used as a simulation tool for this research. An artificial neural network (ANN) controller design is presented by using "Gradient Descent Back-propagation with Adaptive Learning". Application on ANN controller is accepted for publication in the 1st International conference of Abasyn University.

The manuscript of the thesis is comprised of six chapters. Chapter-1 is introductory which describes the idea behind, contributions and the scope of the work.. Chapter-2 describes the background knowledge required for the current research work and previous investigations in this field. Chapter-3 is dedicated to the description of proposed methodology and experimental work done during entire thesis. It includes all the details regarding training and testing of neural network controller. "Gradient Descent Back-propagation with Adaptive Learning" along with pre and post processing data training is used for getting best ANN training. A conceptual model is also proposed for the control solution of Air to Fuel Ratio (AFR). Chapter-4 includes all the numerical simulations which are used for verification of the criteria used. Chapter-5 presents the final results and relevant description. Chapter-6 describes the conclusions which are made from the results. Some future perspectives which can be added are presented in this chapter as well. Presented conference paper is also attached at the end of this report.

Abstract

In present research Air to Fuel Ratio for 2NZFE fuel injection engine is controlled by using Artificial Neural Network (ANN). ANN is an information processing paradigm which is inspired by a biological nervous system. ANN trains controller by passively accepts inputs and its corresponding desired outputs. Case study is done on 2NZFE fuel injection engine. All the required data for ANN training is collected by performing several active tests on 2NZFE fuel injection engine at TOYOTA EFI workshop located in I-9 sector of Islamabad. Total 100 data sets are recorded comprising of 11 parameters (10 inputs+01 output) in each set. Offline training of ANN is done by taking 10 parameters as input upon which AFR is dependent, whereas AFR is taken as an output parameter. ANN is modeled by using three layers including input, output and hidden layer with five neurons in hidden layer. “Gradient Descent Back-propagation with Momentum and with Adaptive learning rate” is used as a training algorithm with sigmoid tangent as an activation function between layers. 3000 iteration are carried out. Trained neural network predicted outputs which are then analyzed and compared with the actual targets. Based on this information electronic control unit of automobile engine can adjust engine parameters according to the stoichiometric AFR.

Keywords: Artificial Neural Network (ANN), Air Fuel Ratio (AFR), 2NZFE Fuel Injection Engine, Electronic Fuel Injection (EFI) Engine. Automobile Engine, Stoichiometric AFR.

Contents

List of Figures XI

List of Tables XIII

Chapter 1 14

Introduction 14

1.1. Present Investigation 18

1.2. Scope and Motivation 20

1.3. Contributions 21

1.4. Thesis layout 21

Chapter 2 23

Background and Previous Investigations 23

2.1. Introduction to the Artificial Neural Network (ANN) 23

2.1.1. MATHEMATICAL ABSTRACTION 28

2.1.2. ACTIVATION/TRANSFER FUNCTION 31

2.1.3. Analogy between Human Neuron and Artificial Neuron 33

2.1.4. ANN-Pros 35

2.1.5. ANN-Cons 36

2.2. How ANN works for getting better performance of Automobile Fuel Injection engines. 36

2.3. Previous investigations on Control Solution of AFR 40

Chapter 3 44

Proposed Methodology & Experimental Procedure 44

3.1. Design strategy of Artificial Neural Network model 47

3.2. Modeling & Designing of ANN Controller 50

3.2.1. Data acquisition 50

3.2.2. Training 60

3.2.2.1. Gradient Descent Back Propagation with Adaptive Learning 61

Chapter 4 64

Numerical Simulation 64

4.1. Selection of optimize parameters in designing of Neural Network Model 65

4.2. Training results by varying number of epochs and number of nodes in hidden layer 68

Chapter 5 86

Experimental Results & Discussion 86

5.1. Experimental Work 86

5.2. ANN Training for the Control Solution of AFR 87

Chapter 6 93

Conclusion & Future Work 93

6.1. Conclusion 93

6.2. Future Work 95

REFERENCES 97

List of Figures

Figure 1: Harmful effects of exhaust emissions of vehicles 15

Figure 2: Average percentage of contaminations emitted while driving gasoline engines courtesy by UNEP 15

Figure 3: ANN control configuration 25

Figure 4: Updating weight strategy of ANN 26

Figure 5: MLP NETWORK MODEL 27

Figure 6: SINGLE NODE MLP NETWORK 29

Figure 7: MULTI NODE MLP NETWORK 30

Figure 8: Threshold 31

Figure 9: Logistic 32

Figure 11: Sigmoid tangent 32

Figure 11: HUMAN NEURON MODEL 33

Figure 12: ARTIFICIAL NEURON MODEL 34

Figure 13: An overview of entire process 39

Figure 14: Block diagram of entire process 40

Figure 15: Modeling and Control of Electronic Fuel Injection Engine 46

Figure 16: Steps required in ANN designing 47

Figure 17: Precise view to create, train and test a supervised artificial neural network 49

Figure 18: Snapshots of sensor during active test on 2NZ-FE fuel injection engine 52

Figure 19: Snapshots of Intelligent viewer (software) by attaching it with sensor during active test on 2NZ-FE fuel injection engine 53

Figure 20: Training and testing of supervised artificial neural network 58

Figure 21: View of Multilayer Feed Forward Artificial Neural 59

Figure 22: MATHEMATICAL MODEL OF MLP NETWORK 67
Figure 23: (TRIAL-1) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate 75
Figure 24: (TRIAL-2) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate 76
Figure 25: (TRIAL-3) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate 77
Figure 26: Trained ANN output window obtained in trial-2 80
Figure 27: Performance graph (TRIAL-2) 81
Figure 28: Graph of Training States (TRIAL-2) 83
Figure 29: Regression graph (TRIAL-2) 84
Figure 30: Comparison graph between predicted and actual outputs at which ANN is trained for predicting AFR of 2NZFE fuel injection engine with MSE of 1.4667e-006 90
Figure 31: Proposed ANN Model 92

List of Tables

Table 1: *Range of parameters upon which engine data is collected* 19

Table 2: A summary of analogy between ‘human neuron’ and ‘artificial neuron’ 35

Table 3: *Basic car features upon which Real time Engine Data is recorded* 50

Table 4: 2NZ-FE Engine Real-time Data 54

Table 5: Training reults at 500 Epochs 69

Table 6: Training reults at 1000 Epochs 69

Table 7: Training reults at 3000 Epochs 70

Table 8: Training reults at 5000 Epochs 71

Table 9: Comparison table by varying nodes in hidden layer with 3000 epochs 72

Table 10: Comparison table by varying number of epochs with 5nodes in hidden layer 73

Table 11: Gradient Descent Back Propagation with Adaptive Learning Rate 78

Table 12: Comparison table between ‘Actual Outputs’ and ‘Predictions’ by using ‘Gradient Descent 89

Table 13: Neural Network Training Parameters and Progress achieved by using ‘Gradient Descent Back Propagation with Adaptive Learning Rate 91

Chapter 1

Introduction

This research contributes in controlling air-to-fuel ratio of electronic fuel injection engines by using an artificial intelligence technique known as ANN. ANN are widely used in present days due to its function approximation, learning and adaptive capabilities. Strength of ANN lies in the fact that no prior knowledge is required about the dynamics of a system which need to be controlled. Applying ANN to a system needs sufficient input and output data instead of a mathematical solution. Furthermore, it can continuously re-train for new data sets during the operation, thus it can adapt to changes in the system. The observed trends of input/output data sets during training are used by the controller to produce version of output.

Theoretically control solution of air to fuel ratio by ANN is very precise, which may result in the most efficient engine performance. It also helps in minimizing pollution like toxic emissions generated by the vehicle. To achieve optimize control of AFR we need to bring AFR near stoichiometric value. This research work is adopted to get stoichiometric AFR in an engine by using ANN which helps engine in releasing less toxic emissions.

For many years, people have been experiencing different environmental problems especially the pollutants generated by the vehicles. Most vehicles run on gasoline or diesel fuel contaminate the earth which is a mobile source of pollution. In urban areas where large numbers of vehicles are in practice, city dwellers living conditions are inevitably unhealthier than rural counterparts. Pollutants generated by the vehicles have far negative effects not only on humans but also on wildlife as they cause acid rain and smog. Products which are emitted via exhaust manifold as a result of combustion in an internal combustion engine contain harmful pollutants such as nitrogen oxides (NOX), carbon monoxides, and sulfur oxides. Average percentage of contaminations emitted while driving gasoline engines is shown in figure-2 with pie graph.

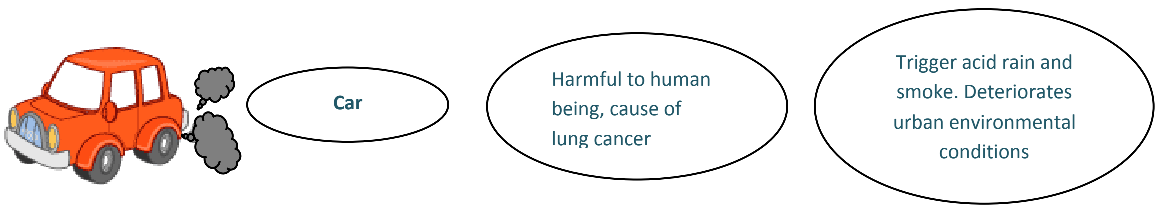


Figure 1: Harmful effects of exhaust emissions of vehicles

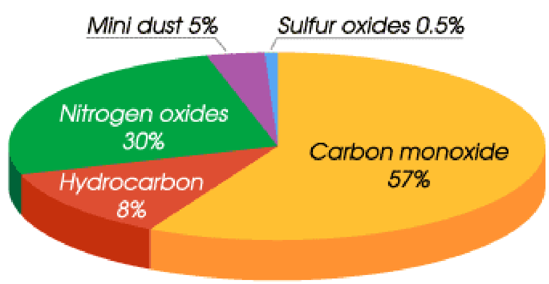


Figure 2: Average percentage of contaminations emitted while driving gasoline engines courtesy by UNEP

By the invention of vehicles, not only our lives became comfort but our environment became more polluted which is quite dangerous. There are many other sources as well which contributes in environmental pollution but vehicle emissions (41% pollutants) are far more toxic as compared to household heating (5% pollutants), power generation (18% pollutants) and even factories (26% pollutants).

Since 1970, different communities all over the world have set strong constraints on the level of toxic emissions endured for a single vehicle. In order to meet these stringent government constraints on pollutants emitted by the vehicles, researches are pushed toward the development of suitable mechanical, chemical and electronic devices to reduce noxious emissions of exhaust gases from automobile engines. For the regulation of AFR involved in internal combustion engines, several relevant constituents are considered in previous researches including use of lead free fuel and introduction of catalytic converters in an Electronic Fuel Injection (EFI) engines for processing exhaust gases before entering into the atmosphere.

Different approaches have been attempted since 1975, to overcome pollution by processing the exhaust gasses chemically by the development and induction of Three-ways Catalytic Converters ((TWC) in SI fuel injection systems. TWC converter is basically a vehicle emission control device which processes the exhaust gases by accelerating the oxidation of unburned hydrocarbons(HCs) and carbon mono oxide (CO) in to water vapors (H_2O) and carbon di-Oxide(CO_2). On the other hand it also helps in the reduction of nitrogen oxides (NOx) into nitrogen(N2) and Oxygen (O2) gas [1]. Efficiency of catalytic converters is dependent upon stoichiometric value of air to fuel ratio, they give maximal efficiency near optimized or stoichiometric value of air to fuel ratio (i.e. AFs ratio=14.7 or 1 in lambda for gasoline engines) with in a very small range. Even a small perturbation around this value of AF ratio may cause severe loss in efficiency of these catalysts, For example 1% deviation in AF ratio with respect to its stoichiometric value may cause up to 50% depletion in the efficiency of catalytic converters while processing pollutants [2].

Many different controllers have also been designed for an accurate control of AFR in EFI engines. In present research, ANN is used for regulating air to fuel ratio near optimized stoichiometric value, as a result catalytic converters would be able to perform well [3].

Pollutants which are generated as a result of combustion reaction are modeled in equation below[2]:

$$\frac{1}{AF} C_8H_{18} + (O_2 + 3.77N_2) \rightarrow a_1CO_2 + a_2CO + a_3NO + a_4O_2 + a_5N_2 + a_6H_2O + a_7CH_4 \quad (1)$$

where C_8H_{18} is the fuel, $(O_2 + 3.77N_2)$ is the air mixture of oxygen and nitrogen gas, CH_4 is representing unburned hydrocarbons (HCs) as a residual and $a_i (i = 1 \ldots .7)$ are reaction coefficients of products which are affected by air to fuel ratio in a nonlinear way. Combustion process will generate all the reaction products as presented in above equation-1 when air to fuel ratio (AFR) is in or near stoichiometric value (AFRs).In case of lean mixture, when (AFR>AFRs) combustion products will be CO_2,O_2, N_2 and H_2O as shown in equation-2.

$$\frac{1}{AF} C_8H_{18} + (O_2 + 3.77N_2) \rightarrow a_1CO_2 + a_4O_2 + a_5N_2 + a_6H_2O \quad (2)$$

When (AFR<AFRs) i.e. in case of rich mixture, combustion reaction will generate carbon monoxide, nitrogen gas, water vapors and unburned hydrocarbons as shown in equation-3.

$$\frac{1}{AF} C_8H_{18} + (O_2 + 3.77N_2) \rightarrow a_2CO + a_5N_2 + a_6H_2O + a_7CH_4 \quad (3)$$

Due to the presence of uncertainties, unpredictable disturbances and nonlinear behavior of engine, controlling AF ratio near stoichiometric value (AFs) is a very difficult task. To resolve this problem a nonlinear black-box solution as a neural network, termed as an artificial intelligence became attractive technique to be conducted for stoichiometric control of AF ratio. The automotive field is widely influenced by the increasing use of artificial neural network in control areas with applicable contributions, e.g. in antilock braking system [4], engine idle speed control [5], nonlinear dynamic control [6], identification and prediction of dynamical systems [7-10]and in fuel injection engines [11].

Implementation of ANN is done on 2NZFE fuel injection engine to order to analyze its effectiveness and suitability for the control of AFR in EFI engines. Once neural network is trained, it can be inserted into the Electronic Control Unit (ECU) [12] of fuel injection engine for acquiring precise stoichiometric control of AF ratio.

1.1. Present Investigation

In present research, case study is done on Corolla GLI equipped with 2NZ-FE with a fuel type of gasoline and 1298CC displacement. 2NZFE is a 1.3L electronic fuel injection engine consisting of spark ignition with a bore of 75mm; stroke is 73.5mm and compression ratio of 10.5:1. Several active tests are performed by running 2NZFE at TOYOTA EFI workshop located in I-9 sector of Islamabad. Intelligent tester and intelligent viewer software are used in this research to measure Air-to-Fuel ratio (AFR) and other parameters which are dependent upon Air-to-Fuel ratio. The range of parameters upon which active tests are performed includes are mentioned in table-1. It includes engine speed in ‘rpm’, calculate load in ‘%’, vehicle load [13] in ‘%’, mass air flow (MAF)[14] in ‘g/s’, coolant temperature in °C, engine run time in‘s’, opening of injector port in ‘ms’, injection volume in ‘ml’, IGN advance [15] i.e. ignition advance before reaching top dead center in ‘deg’ and Air-to-Fuel ratio in λ.

Table 1: *Range of parameters upon which engine data is collected*

Engine speed (rpm)	Calculate load (%)	Vehicle load (%)	MAF (g/s)	Coolant Temp (°C)	Air Intake Temp (°C)	Engine Run Time (s)	Injector (port) (ms)	Injection Volume (cylinder1) (ml)	IGN Advance (deg)	Target Air-fuel ratio (λ)
Min: 0	Min: 0	Min: 0	Min: 0.07	Min: 40	Min: 16	Min: 0	Min: 1.4	Min: 0.05	Min: 4	Min: 0.999
Max: 4339	Max: 48.6	Max: 34.9	Max: 12.37	Max: 83	Max: 26	Max: 280	Max: 6.52	Max: 0.223	Max: 40.5	Max: 1.192

In first step data sets are collected by running engine and then these data sets used to train Artificial Neural Network (ANN) in order to get control over AFR and to bring it near stoichiometric value which plays an important role in enhancing the performance of a vehicle and for reducing toxic emissions that has severe effect on our lives. Artificial Neural Network once trained for 2NZFE engine it can be inserted into its electronic control unit to get better efficiency of a vehicle.

.

1.2. Scope and Motivation

Artificial intelligence (AI), referred as an Artificial Neural Network (ANN) is a cognition technique widely used in present days. It works on human psychology. Cognitive AI process recurrent and hypothetical information in the same way as human mind do. It establishes its goal through reinforcement learning by interacting with environment. AI proceeds on finite set of predefined steer which depicts the need of system. Its ability to manipulate correspondence, enigma and error by compiling distributed dynamical representations with configuration, traits motivational relevance.

Since ANNs are the best at recognizing trends in data, they are well appropriate for estimation, process control, and validation of data in many other fields. In present days, lot of work has been done on the control solution of AFR of different automobile engines by using ANN. Designing of ANN for the control of air-to-fuel ratio of EFI engine is challenging.

EFI engine which we have chosen for our analysis is 2N-ZFE fuel injection engine for which control solution of AFR is proposed by using ANN. For the control of AFR on 2NZ-FE fuel injection engine by ANN, there is a need to develop a thorough investigation on selected EFI engine. Collection of real time data by running engine in accordance with AFR and its dependent parameters is carried out to identify their trends which help ANN in learning.

Designing of ANN is done in order to get its fine modeling with least mean square error between neural predictions and the actual operating values in minimum time. Among various designing parameters and well known training algorithms, the best performance may be investigated and used for the control of air-to-fuel ratio of 2NZ-FE fuel injection engine.

1.3. Contributions

The literature on the control solution of air-to-fuel ratio of fuel injection engine is abundant. Researches on the control of air-to-fuel ratio of different fuel injection engines are also discussed by taking artificial neural networks into account. Previous researches were time consuming especially training time for predicting air-to-fuel ratio was large with less accuracy. In designing and modeling of neural network for 2NZ-FE fuel injection engine various types of training algorithms are discussed by considering their pros and cons. A new approach of Pre and Post processing data while training neural networks is proposed for getting more refine results.

1.4. Thesis layout

This thesis comprises of six chapters. Here each chapter is described briefly.

Chapter-1 introduces the salient features of research work carried out. This chapter also includes the main motivations which lead towards the specific work and contributions to the state of the art technique. This summarizes the knowledge contributed through this thesis.

Chapter-2 describes the background knowledge required for the current research work and previous investigations in this field of knowledge. Especially control solution of air-to-fuel ratio of fuel injection engines by using artificial intelligence approach is discussed. Different methods for its control which were described by earlier researchers are given in detail.

Chapter-3 is dedicated to the description of all the experimental work done during this thesis. It includes the detail of artificial neural network design and modeling, real time engine data of

2NZ-FE fuel injection engine, training and testing of neural model. Neural network modeling by using ‘Gradient Descent Back propagation with Adaptive Learning rate’ is discussed.

Chapter-4 describes the details of all the numerical simulation procedure carried out. The numerical simulation on MATLAB for neural controller is given in this chapter.

Chapter-5 presents the final results and relevant description. It includes numerical simulation results for comparison work. In current research work we have also gone through different analysis, this chapter includes the obtained results from this work.

Chapter-6 In this chapter a summary of all the work done to contribute to the knowledge is described. It contains the conclusions which were made from the results and some recommendations for the future use. It’s an ongoing project and some future perspectives which may be added in the research are discussed.

Chapter 2

Background and Previous Investigations

This chapter is focused on the back ground needed for the present research and the previous investigations carried out in the field of automotive industries for regulating AF ratio of electronic fuel injection engines by using various techniques and methodologies.

2.1. Introduction to the Artificial Neural Network (ANN)

An artificial neural network (ANN) is basically information processing linguistic which acts as a biological nervous system such as how brain works and convey information [16-18]. ANN are broadly used in modern world in almost every field of life due to its function approximation, learning and adaptive capabilities. It consists of numerous interconnected processing elements called 'neurons' or 'nodes' that work in parallel to solve specific problem. Each node in ANN communicates over large number of weighted connections by receiving input signals from neighbors or external source and propagates output signals to the other after computation. In machine learning and computational neuroscience, an ANN is often named as neural network (NN).

The asset of neural network lies in the fact that no former knowledge about dynamics of the system to be steered is required. Implementation of artificial neural network to a system requires sufficient inputs and outputs data sets instead of mathematical equation. Artificial neural networks are trained by accepting inputs and yielding its corresponding desired outputs, according to the algorithms, which is taught during its training. Artificial neural networks also have an ability to re-train during operation for new data sets i.e. it can accommodate with the changes in the system.

The accessible data set is divided into two different parts, one is for training and the other corresponding to test or validate the model. The aim of training is to ascertain the set of connection weights and nodal threshold that helps artificial neural networks to predict outputs that are adequately close to the target or desired value. The complete data signed for training should contain abundant patterns so that network can mimic intrinsic relationship between input and output variables accurately. Neural networks need to be designed, modeled, trained and implemented in such a manner that set of input data results into desired outputs. Input data array may consist of raw data, an image, a wave or any data that can be stored into an array. Inputs present to the ANN, whereas corresponding target/ desired response set at the output.

An artificial neural network configures through learning or training process for specific application. Learning means adjustment of synaptic connections between neurons by iterative algorithm. ANN once trained on an acceptably set of inputs are then allowed to respond output from what it has been learned. Neural network has two phase of operation Training and Testing. In training phase, weights of neural network are adjusted to map the input of system to its output whereas testing means evaluating the neural network with best weights found during training. Training of neural network is further categorized into offline and online mode. Batch (offline) training of a network proceeds by updating weights and biases after all the inputs are presented. Incremental training changes the weights and biases of a network as needed after presentation of each individual input vector is sometimes referred as “on line” or “adaptive” training. Difference between target and the approximate neural output i.e. an ‘error’ acts as training signal for neural network in updating weights. A schematic view of ANN control configuration is given in figure-3 below [19]:

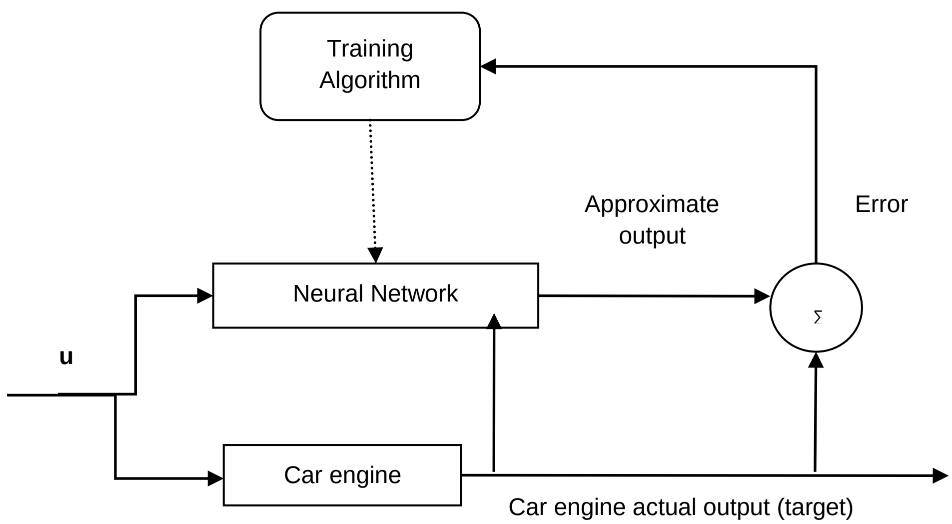


Figure 3: ANN control configuration

Artificial neural network works by first observing system performance and sets a relationship between inputs and output. Its output is predicted from what it has been learned in training. In this research work car engine is considered as a system, artificial neural network first observed its performance built relationship between its inputs and outputs and then estimate its own outputs. Difference between approximate/predicted output of neural network and car engine actual output i.e. an error helps in updating weights and the biases by acting as a training signal after every iteration. Figure-4 shows, how neural network adjusts/updates its weights during training after comparing predicted/approximate outputs with the target values until both matches and when the error becomes negligible.

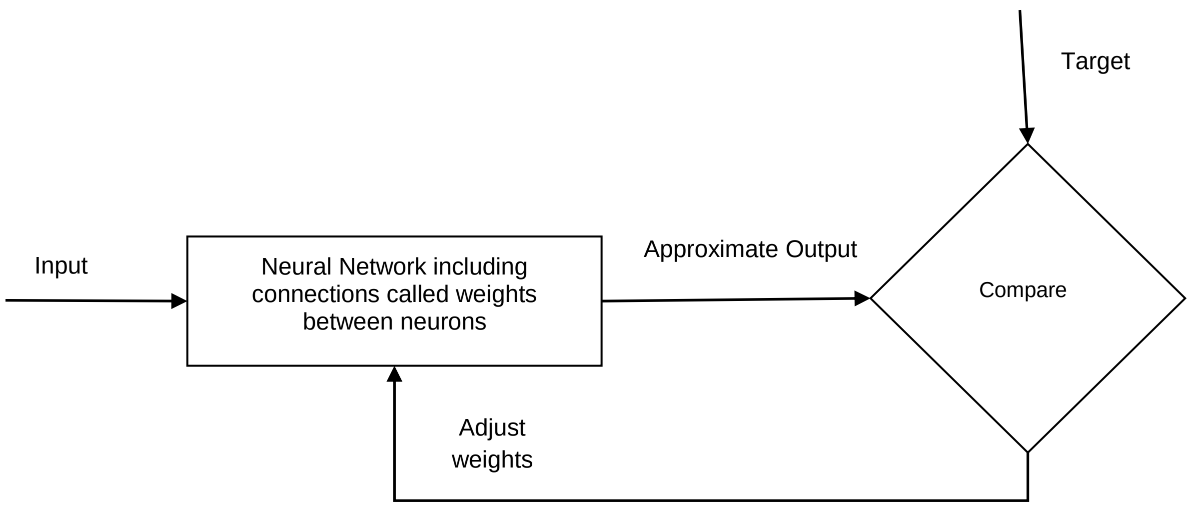


Figure 4: Updating weight strategy of ANN

The first computational, trainable neural network named as 'perceptron' also known as known as 'Rosenblatt's perceptron' was developed by a neuro-biologist of Carnel University Rosenblatt et al.[20]. Bershad et al. [21] proposed a modified version of Rosenblatt's algorithm with two layer perceptron that could identify the parameters of stoichiometric nonlinear system The capabilities of neural networks were expanded from linear to nonlinear domains by Gopalakrishnan et al. [22]. .Diggavi et al. [23] presented stoichiometric analysis of steady-state and transient convergence for learning

MLP network [24] is considered as a most useful neural network in function approximation. It is a nonlinear parameterized map from input space to the output consists of an input layer, one or more hidden layers and an output layer as shown in figure-5.

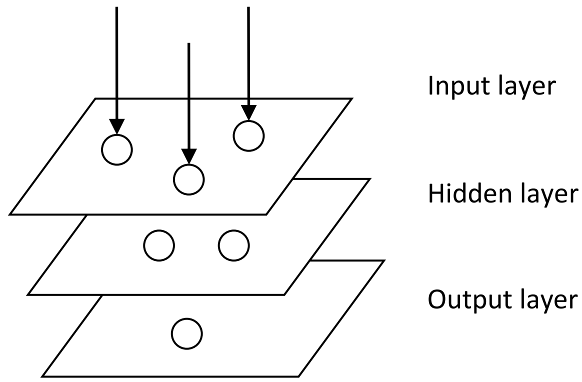


Figure 5: MLP NETWORK MODEL

For pattern recognition and information processing feed forward MLP network is considered as a most reliable technique. It can deals with hard limiting nonlinearities. By substituting saturable nonlinearities with hard limiters computational efficiency of network can be enhanced. Weight approximation and scaling can deal effectively when network has hard nonlinearities. Since output of disrupted neuron is totally independent of the magnitude of nonlinearity input, weights can be scaled to proper level before estimation. This helps in increasing dynamic range for calibrating network weights and allows easier transformation between computer simulation and hardware implementation. After discovery of MLP network and its successful demonstration on applied problems, mathematicians developed solid theoretical basis for its understanding.

The computational capabilities of three layered neural networks were demonstrated by Hunt et al. [25] who showed various applications of ANN with MLP in general function approximation theorem.

This finding showed that ANNs have the capacity to solve major problems in a wide range of applications not restricted to a particular field or domain. This is because several classification tasks, predictions and decision support problems can be tailored as function approximation problems. An immense amount of research has been devoted to the methods of modifying internal parameters (weights) to obtain the best-fitting functional approximations and training results.

2.1.1. MATHEMATICAL ABSTRACTION

Several key components have to be considered while mapping a neural model including input signals, bias input which shifts activation function to the right or left while presenting outputs, summing junction, weights that interconnects neural network which provides strength to the connection and activation/transfer function that controls the magnitude of the output within an acceptable range of values. ANN operates in a certain framework such as:

1. Neuron receives input as a signal.
2. Signal builds up in the cell.
3. Ultimately, cell discharges or fires through the output which depends on threshold.

Mathematical model of single node and multiple node MLP network is shown below with input signals in figure-6 and figure-7 respectively with bias input, summing junction, activation function and output.

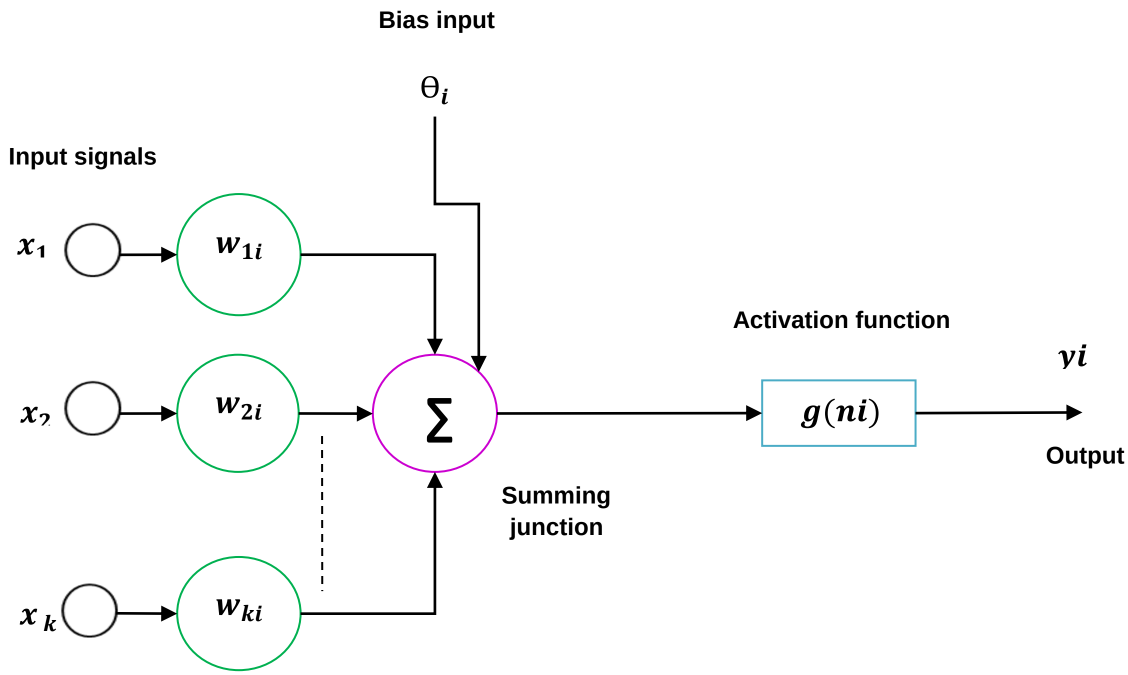


Figure 6: SINGLE NODE MLP NETWORK

The inputs x_k, $k=1\ldots K$ to the neurons are multiplied by synaptic weights w_{ki} and summed up together with the constant bias term Θ_i. The resulting n_i is the input to the activation function g which then decides output.

The basic equations-4 and 5 describe the dynamics of each neuron.

$$ni = \sum_{j=1}^{K} w_{ji}x_j + \Theta_i \qquad (4)$$

$$yi = g(ni) = g\left[\sum_{j=1}^{K} w_{ji}x_j + \Theta_i\right] \qquad (5)$$

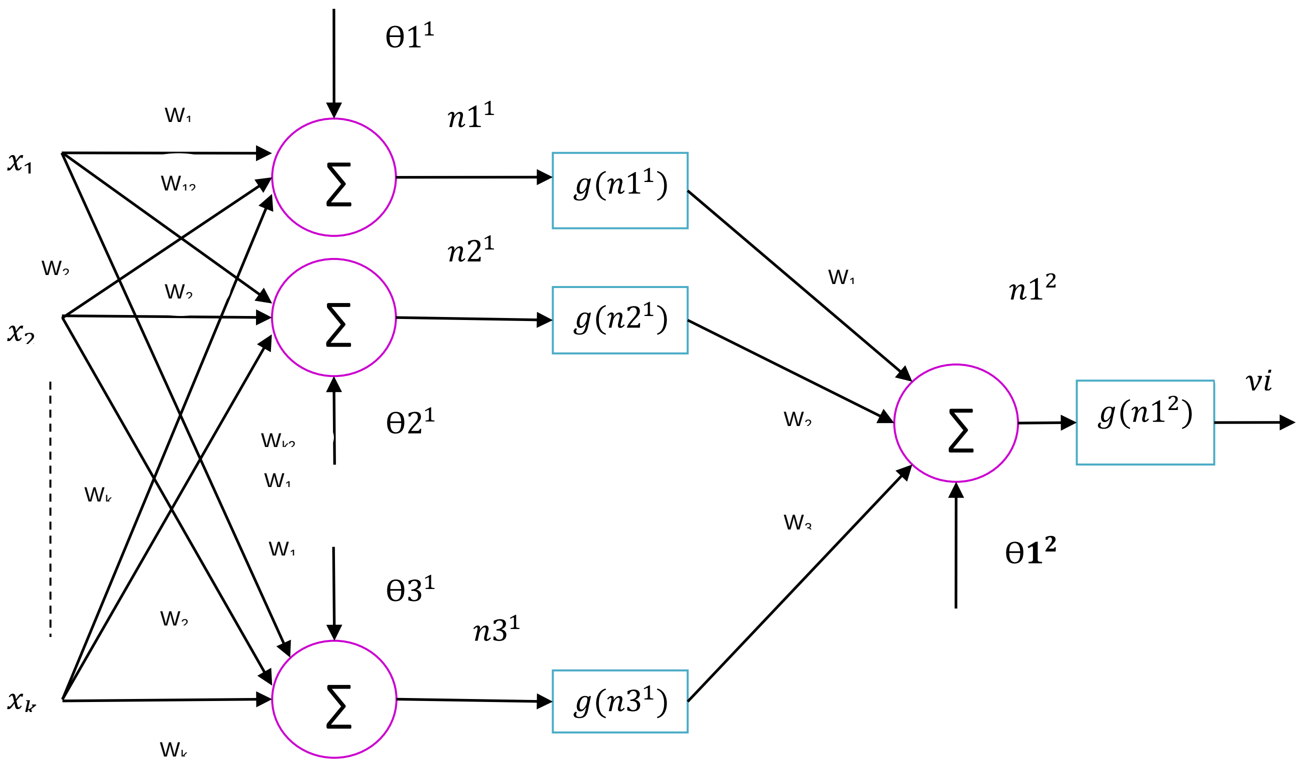


Figure 7: MULTI NODE MLP NETWORK

Figure-7 shows a view of MLP network but with multiple nodes. It also shows how multiple nodes are arranged parallel for solving specific problem of ANN.

$$yi = g\left[\sum_{j=1}^{3} w_{ji}^{2} g(n_j^1) + \theta_j^2\right] \qquad (6)$$

$$yi = g\left[\sum_{j=1}^{3} w_{ji}^{2} g\left[\sum_{k=1}^{K} w_{kj}^{1} x_k + \theta_j^1\right] + \theta_j^2 \right] \qquad (7)$$

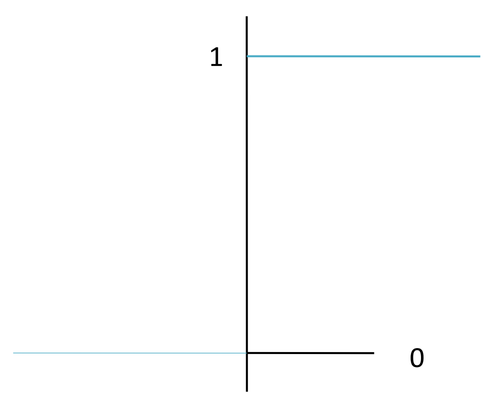
In equations-6 and 7 ‘g’ is used to symbolize an activation function which is same in both of the layers, *w* and *Θ* are the weights and the biases respectively, whereas superscript 1 and 2 are representing first layer and the second layer. For designing any neural network, designer needs to decide architecture of MLP network first i.e. total number of hidden layers, number of neurons to be placed in each layer and the activation function. Activation functions are assumed as known parameter, whereas weights and biases as unknown parameters which are to be estimated.

Many algorithms exist for determining network parameters. These algorithms are called learning or teaching algorithms in the literature of neural network. The most well-known algorithm is ‘Gradient descent back propagation with adaptive learning rate’. We will analyze its effectiveness in current research while training our neural model.

2.1.2. ACTIVATION/TRANSFER FUNCTION

An activation function or transfer function acts as a ‘transformation essence’ so that output of neuron in ANN remains in a certain limit. Mostly three types of activation functions are being used i.e. threshold, piecewise and sigmoid functions.

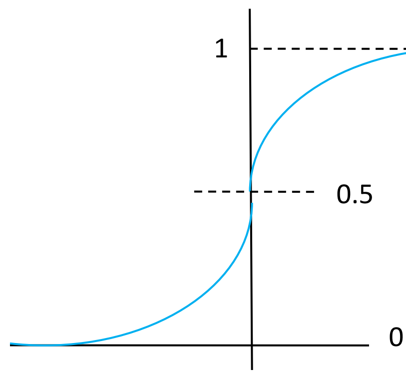
1. *Threshold function*: This function lies between 0 and 1 with certain threshold value. Depending upon the sum of inputs either greater than or less than threshold value ‘x’, it gives 0 or 1 as an output value as shown in figure-8 below:



$$f(x) = \begin{cases} 0 \ if \ x < 0 \\ 1 \ if \ x \geq 1 \end{cases}$$

Figure 8: Threshold

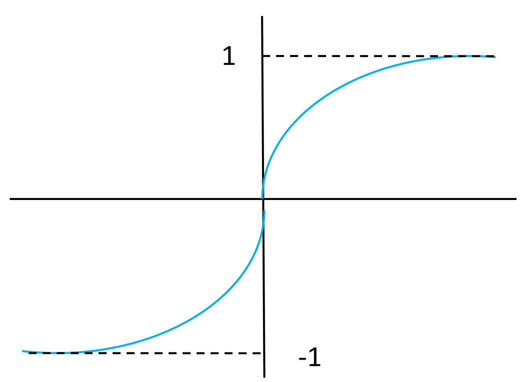
2. *Piecewise-Linear or Logistic function:* This function not only give 0 or 1 but it may also set to any other values between this interval (0-1) depending upon amplification factor in certain zone of linear operation. Logistic transfer function graph is shown in figure-9.



$$\mathrm{f}(x) = \frac{e^x}{1 + e^x}$$

Figure 9: Logistic

3. *Sigmoid function:* This function can range between 0 to 1 and 1 to 1 in some models. Its example is hyperbolic tangent function as mentioned in figure-10 below.



$$\mathrm{f}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Figure 10: Sigmoid tangent

2.1.3. Analogy between Human Neuron and Artificial Neuron

Interconnected artificial neurons in ANN mimic several properties of biological neurons. Artificial Neural Networks are designed in the same way as that of ‘Biological Neural Network’ which is a part of human brain [26].

There are 10^{11} neurons exist in human brain with cell body and a collection of ‘dendrites’ which bring electrochemical information to the cell whereas ‘axon’ in brain transmits these electrochemical information out of the cell. When the collective effect of inputs reaches a certain threshold axon gets fired as an output of neuron. The junctions called synapses across which an axon from one neuron can influence the dendrites of another neuron. Synapses can generate positive effects (encourage neurons to fire) as well as negative effects (discourage neurons to fire) in dendrites. The total numbers of synapses in our brain are of the order of 10^{16} and a single neuron receives inputs from 10^{5} synapses. How our brain learns and remembers is associated with the interconnection between numerous neurons. Figure-11 below is showing human neuron model specifying dendrites, cell body, threshold, summation and axons.

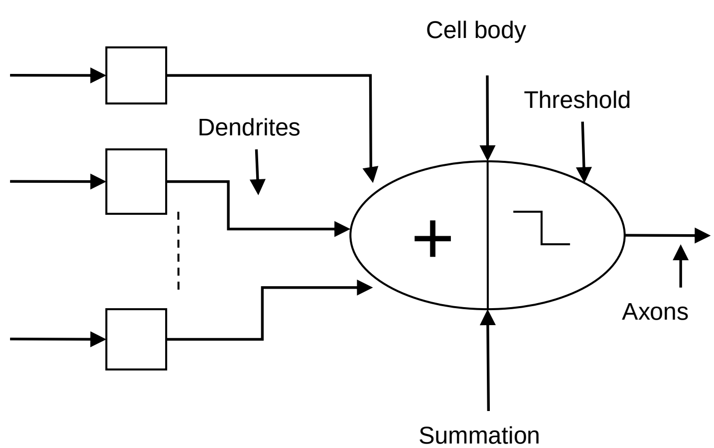


Figure 11: HUMAN NEURON MODEL

An artificial neuron as shown in figure-12 is a processing device with several input connections associated with the weights, biases, transfer functions, epochs, training/learning algorithms, different layers and nodes which determine its output. By adding an extra connection to each neuron threshold values and weights can be adjusted together with an input value of -1 and threshold weight. If sum becomes greater than zero neurons get fired.

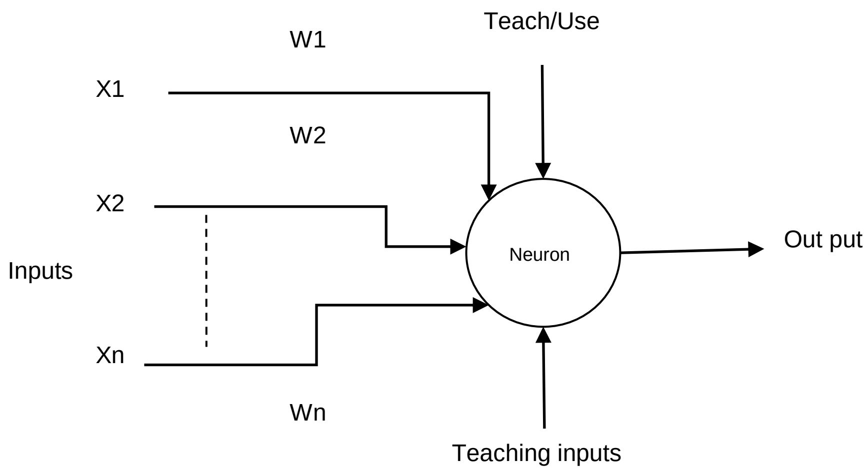


Figure 12: ARTIFICIAL NEURON MODEL

Figure-12 shows, a neuron fires if and only if;

$$X_1W_1 + X_2W_2 + \dots + X_nW_n > T \quad (8)$$

whereas,

$X_1 \dots X_n$ = *Inputs*

$W_1 \dots W_n$ = *Weights*

T = *Threshold which determines cell should fire or not.*

Table 2: A summary of analogy between ‘human neuron’ and ‘artificial neuron’

Human neuron	**Artificial neuron**
Neuron	Processing Element
Dendrites	Combining function, receives Information
Cell Body	Transfer function process information
Axons	Element output, carries processed information to the other neurons
Synapses	Weights

Table-2 is an abstraction of analogy between human neuron and the artificial neuron where neuron, dendrites, cell body, axons and synapses of human neurons are similar with processing element, combining function, transfer function, outputs and weights of artificial neurons respectively.

2.1.4. ANN-Pros

a. A neural network can be used to solve linear as well as nonlinear programming tasks.
b. Because of its parallel nature a neural network can continue to do work without any issue, even any component fails to work.
c. Once a neural network is learned, it doesn’t need to be re-programmed.
d. Easy to implement.

2.1.5. ANN-Cons

a. Any neural network requires training to operate.

b. A neural network needs emulation.

c. Large neural network requires high processing time.

2.2. How ANN works for getting better performance of Automobile Fuel Injection engines.

Better performance of any fuel injection engine depends on an accurate control of AF ratio which leads desired drive ability, economic fuel consumption and emission levels. AF ratio is actually the mass of ‘air’ to ‘fuel’ present in an internal combustion engine. It would be stoichiometric, if exactly enough air is provided to completely burn all the fuel present in combustion engine. It is mostly abbreviated as “stoich”. The lower the excess air, richer the flame would be and vice versa.

The objective of any controller designed for regulating AF ratio of an automotive fuel injection engine is to achieve an ideal stoichiometric air-to-fuel ratio which is 14.7:1 for gasoline engines and to keep error minimum during transient and steady state engine conditions.

$$\frac{Mass\ of\ air\ (m_a)}{Mass\ of\ fuel\ (m_f)} = 14.7$$

$$\lambda = \frac{Current\ AFR}{Ideal\ AFR}$$

$$\lambda = \frac{{m_a}/{m_f}}{14.7}$$

An error signal (e) is defined as:

Error signal (e) = λ – 1 (9)

From equation-9, if Lambda AFR (λ) approaches to unity, no error would be detected.

λ = 1

e = (1) – 1= 0

EFI engine has an ECU, an independent custom-built computer provided by the vehicle manufacturer to control the running of a fuel engine. ECU has whole inquisitive system of vehicle with distinct subsystems that controls the fuel injection by regulating AF ratio near stoichiometric value in order to get optimum performance of TWC converters under operating conditions. From the information achieved through different sensors, ECU calculates time period for the fuel injector to be open, which then maintain AF ratio. This time period is referred to as the “pulse width,” and the technique by which the ECU modifies this pulse width for better performance and emissions of the engine is referred to as “pulse width modulation.” When fuel is injected into the combustion chamber of a port fuel injected engine, a part of the fuel injected is gone directly into the cylinder and a fraction of the fuel is lost in the formation of a liquid film in the intake runners. The liquid fuel film thus formed either evaporates or flows into the cylinder depending on the manifold wall temperature. If the manifold temperature is high, the rate of evaporation of the fuel film is almost rapid, and if the manifold temperature is low, the rate of evaporation is low, which leads to flow from the film into the cylinder. Hence, the amount of fuel entering the cylinder will vary with the operating conditions, and it is necessary to make real time adjustments to the injector pulse width in order to compensate for film formation.

An optimized ANN is a reliable solution for the control of AF ratio due to its simple nature which could be inserted into the ECU of a vehicle. Neural network is considered as a core element for function, estimation and offline dynamic modeling. Neural network is able to train parameters deal with aging effects, inefficiencies in modeling dynamic and supports fine tuning of controller’s parameters for the specific engine.

Figure-13 shown below is a schematic view showing that performance of vehicle is enhanced as a result of introducing optimized and trained artificial neural network into its electronic control module for regulating AF ratio by tuning its control parameters upon which AF ratio is dependent. Before the introduction of ANN, car was emitting lot of toxic gases, consuming more fuel with smooth-less drive while travelling on a road. Whereas after the involvement of trained ANN in ECU car evolved less toxic gases with smooth drive ability and got better fuel consumption efficiency.

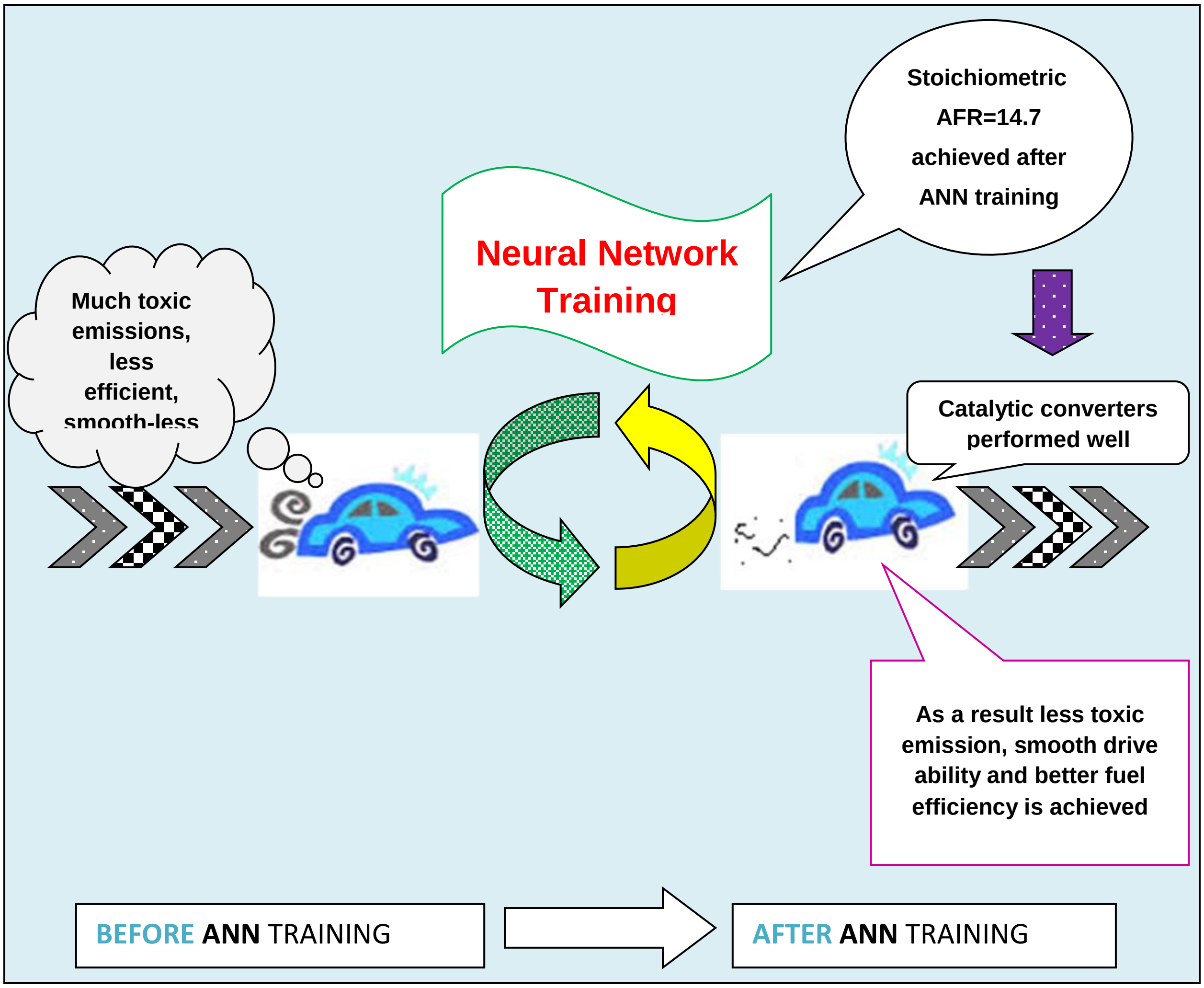


Figure 13: An overview of entire process

Figure-14 is a block diagram of above mentioned process showing different steps, how neural network training enhances the performance of EFI vehicle systematically. As a result of artificial neural network training on EFI engines, a stoichiometric AF ratio can be achieve which is directly related to the efficiency of TWC converters. Maximal efficiency of TWC converter means all the generated pollutants in exhaust manifold are properly oxidized and reduced, which makes car to operate efficiently.

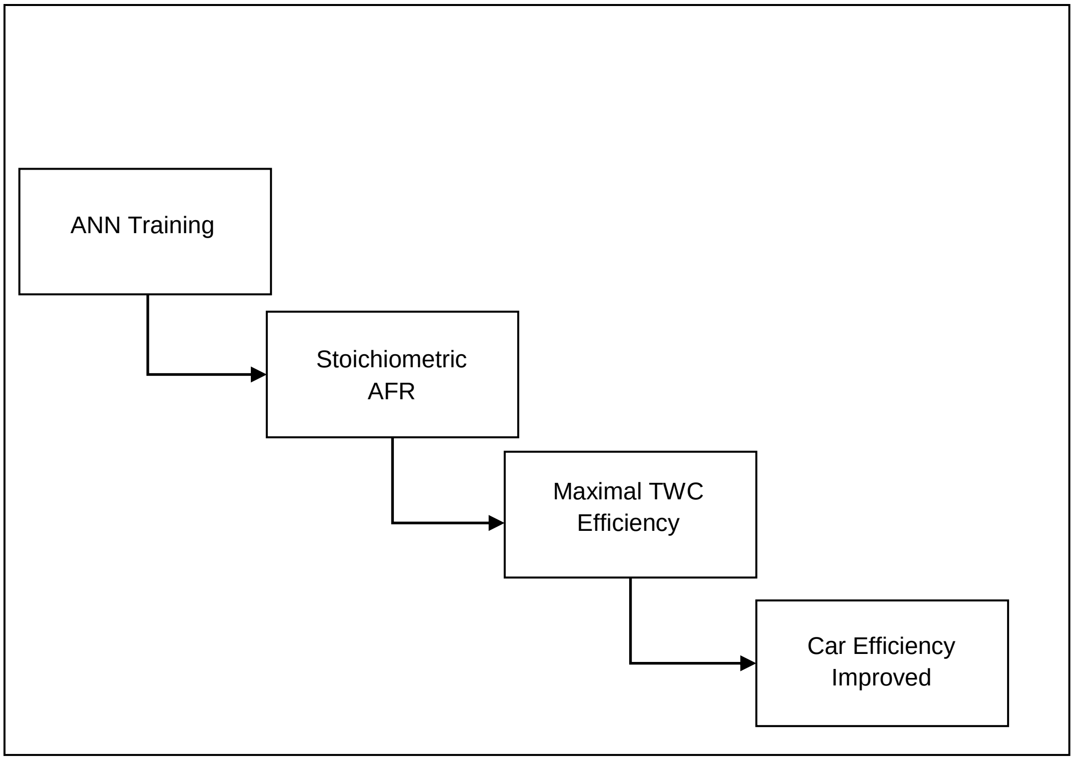


Figure 14: Block diagram of entire process

2.3. Previous investigations on Control Solution of AFR

While reviewing the literature our focus has two aspects; one is to review and analyze the control of AF ratio in EFI engines and the other is to cover its depth and breadth by using artificial neural networks.

The prominent work has been carried out in the field of automobile for regulating AF ratio. Different researchers have proposed different methodologies to get a control over AF ratio of EFI systems. Two of the authors Majors et al. [11] and Shiraishi et al. [27] used a Cerebellar Model Articulation Controller (CMAC) to develop an adaptive neural network for the control of AF ratio with deviations limited to a maximum of ±l%. The major problem cited in practical use of CMAC is its large memory size and lot of time required for its training. Chang et al. [28] presented a model based AF ratio control system by using state-space control and estimation methods. Control of AF ratio is accomplished by compensating fuel dynamics with an observer feedback control and the air dynamics with a drive-by-wire throttle supported feed forward control. This is basically a fuel control system whereas air flow is partially regulated using a DBW (drive by wire) throttle. Model was kept simple, operated at steady state engine conditions because at transient state conditions deposition of fuel droplet on the wall surface was unable to balance the evaporated fuel into the air stream. Experiments were performed at constant speed of 1200 RPM because air throttle motion is the most common engine transient and has the fastest and greatest influence on AF ratio. Won et al. [29] developed a direct adaptive control using Gaussian neural network to compensate transient fueling dynamics and measurement error in mass air flow rate into the cylinder of SI engine. In (GRNN) this method transient fueling compensation method is coupled with a dynamic sliding mode control technique that governs steady state fueling rate but Gaussian networks are considered as difficult to utilize with greater volume of data. Manzie et al. [30] proposed a Radial basis function (RBF) approach for the fuel injection control problems. Lee et al. [31] developed an estimator with Generalized Regression Neural Network Function Approximation algorithm for AF ratio. The AFR estimator described in this study utilized seven inputs that were engine speed, airflow rate, coolant temperature, exhaust HC concentration, exhaust gas temp, oil temp and atmospheric temp, whereas air-fuel ratio as an output. Draw back associated with GRNN networks is that they are relatively insensitive to outliers, requires more memory space to store the model and are relatively slow. Wagner et al. [32] proposed a nonlinear model based control strategy for simultaneous AF ratio control for hybrid vehicles. Model based control uses feed forward inputs to detect disruption, before they impact the process and ensures the parameter is corrected before a problem exists. Hou et al. [33] modeled Elman neural network to approximate AF ratio during transient process with an average error of 1%. But a problem was that, it is difficult for traditional Elman neural networks (ENN) to simulate complicated nonlinear systems directly because their inputs are all instantaneous constant values. Alfieri et al. [8] proposed model based feedback controller of AF ratio in diesel engine based on empirical model. One disadvantage was its large delay time of signals in closed loop that limits the bandwidth of AF ratio. This controller design was made only for linear model. Gnanam et al. [34] proposed a neural network based control system for fine control of AF ratio in bi-fuel engine. NN controller allowed conversion of gasoline ECU to a bi-fuel form with compressed natural gas at minimal cost. Three layered neural network with tan sigmoid activation function used four and eight neurons in first and second layer respectively, where as one neuron in third layer.

Hou et al. [35] used a new multi-step predictive model based on back-propagation (BP) neural network in order to control air-fuel ratio but the derivation of multi-step predictive control process was much complex and required large amount of calculations. Zhai et al. [36] used Radial-basis-function (RBF) based on feed forward-feedback control of AF ratio for spark ignition engine. RBF networks have disadvantage of requiring good coverage of the input space by radial basis functions. Ju-Biao et al. [37] investigated a kind of air fuel ratio control strategy that combined the modified Elman neural network with traditional PI (proportional plus integral) controller. Elman neural network used simplified derivative calculations, which ignored delayed connections at the expense of less reliable learning and no longer recommended. Liu et al. [38] designed a fuzzy neural network controller model for the control of AF ratio under gasoline engine transient condition based on Henrdick model (Mean value engine model MVEM). The network structure is divided into four layers, i.e. the input layer, the fuzzy layer, the inference layer and the output layer. Gauss function is applied in fuzzy layer. Wang et al, 2010 [39] modeled an electronic throttle Fuzzy based PID controller to overcome AF ratio control deviation caused by the variation in oil film vaporization speed during transient conditions. Saraswati et al. [40] proposed training of Neural Network Auto Regressive Model (NNARX) with PID controller using external inputs for the identification of AF ratio in SI engines. The testing of control algorithms on engine test rig was time consuming and had risk of damaging the engine. Computation was required at each sampling time for adaptive control and good plant model needed complex derivation of control law. Barghi et al. [41] used a Recurrent Neuro-Fuzzy Network (RNFN) structure as an intelligent approach to estimate/ control of air to fuel ratio of CNG (Compressed Natural Gas) engines. But RNFN are difficult to construct and tune the proper adjustment of fuzzy weights and biases. Chen et al. [42] used a Linear Quadratic optimal tracking controller to regulate engine AF ratio to the desired level during LNT regeneration period. A particular problem arises in LQR is that initial time is arbitrarily set to zero, and the terminal time is taken in infinite horizon. The infinite horizon problem (i.e., LQR) may seem overly restrictive and essentially useless because it assumes that the operator is driving the system to zero-state and hence driving the output of the system to zero. Jansri et al. [43] proposed a simple nonlinear AFR control for Spark Ignition (SI) engines. A fuzzy PI controlled system was designed for this application. Yar et al. [44] proposed a control solution of AFR of SI engine using super twisting algorithm. Meyer et al. [45] presented an AFR control algorithm based on switching frequency regulator for gasoline engine that has favorable robust stability properties in the presence of both input and model errors.

Chapter 3

Proposed Methodology & Experimental Procedure

In this chapter, proposed methodology and overall experimental procedure is discussed for the control solution of AFR of 2NZFE SI fuel injection engine. A controller based on Artificial Neural Network as an identifier and predictor is proposed. The aim of this controller is to identify the relationship between AFR and its dependent parameters by running engine at different rpm first and then after training for few seconds it can predict AFR as an output parameter with respect to its input parameters with high precision. Another neural controller is trained and inserted after first controller that receives predicted outputs of AFR and brings it near stoichiometric or desired value by varying angle of injector port for the fuel supply in ‘ms’ mass air flow in ‘g/s’ and injection volume of fuel in cylinder1 in ml.

In this research work, we considered a neural network as main elements for function, estimation and offline dynamic modeling to validate the approach. Function estimation is considered to improve the quality of equation based model of the system which once achieved can be used to design the controller. In addition to the control issue the obtained neural network controller can be used to improve the accuracy of the engine model. In designing controller we adopted an indirect control configuration which requires preliminary identification procedure carried out by recurrent neural networks. An overview on control solution of air to fuel ratio by using neural network controller is in figure-15.

Modeling and Control of Electronic Fuel Injection Engine based on an artificial neural network requires few steps to be accomplished.

a. Real time data acquisition via Intelligent Tester by running 2NZFE SI Engine
b. Identification and prediction of data sets by using Neural Network.
c. Tuning of engine parameters for tracking predicted signals generated by neural identifier and predictor to provide null error at the end of transient phase.
d. Tuned parameters with respect to desired/stoichiometric AFR are sent to electronic control unit of engine for feasible control action.

Controller

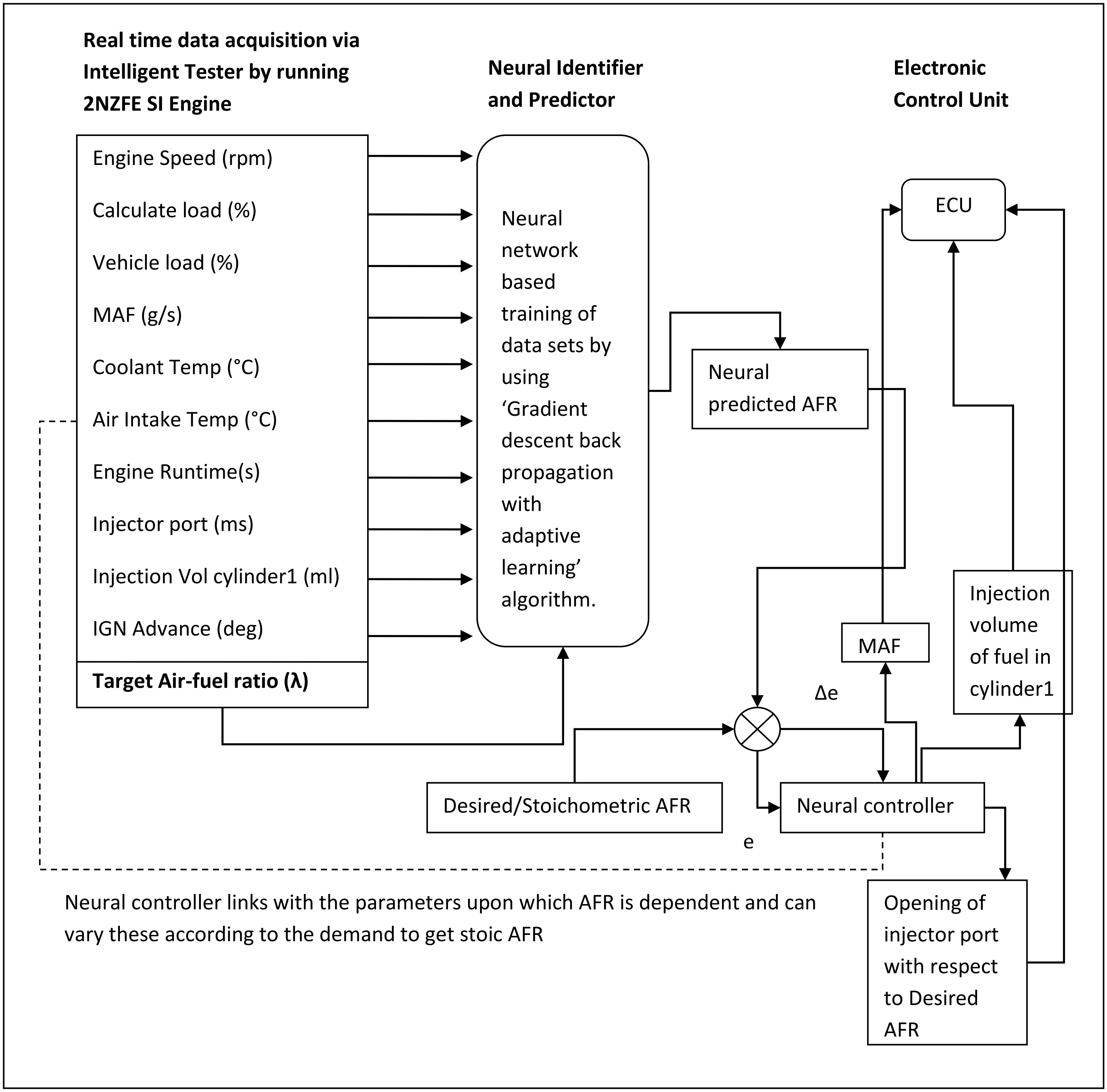


Figure 15: Modeling and Control of Electronic Fuel Injection Engine

3.1. Design strategy of Artificial Neural Network model

Within an artificial neural network model there are three steps required for its designing shown in figure-16.

1. ***Data acquisition*:** Real time engine data is collected by running selected EFI engine at different engine speeds by using intelligent testers.
2. ***Training/Learning**:* Training or learning includes selection of best training algorithm for neural modeling according to our required need.
3. ***Test/validation:*** Just after training ANN model, results should be checked to see whether results or the outputs are according to our demands or not i.e. whether ANN predicted outputs closer to that of the actual outputs achieved.

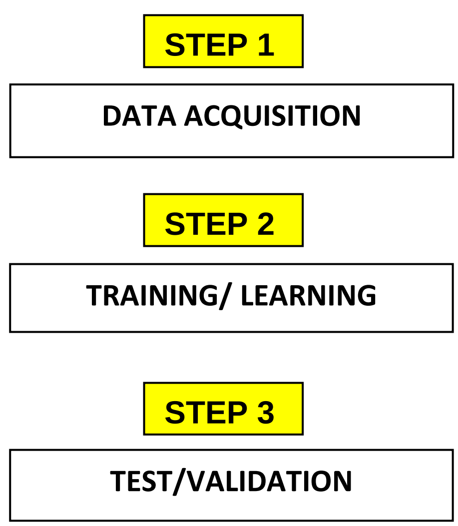


Figure 16: Steps required in ANN designing

Pre and Post-processing are two main steps involved in fine training of artificial neural network. After collection of data we need to define its variable and specify its inputs, outputs, targets, training sets and testing sets. These all are done in pre-processing phase. Next is to create and train a particular type of supervised neural network model by using suitable training/learning algorithm and settling epochs and other parameters. If the training of network model is done successfully, test the network otherwise train the model again. If performance is achieved up to the mark, post-processing of data will display results otherwise model will need to train again unless desired result is achieved. Pre-processing scales data with minimum value of -1 and maximum +1 whereas post-processing brings simulated results to the original scale. Figure-17 gives more precise view of entire process used to create, train and to test a supervised artificial neural network.

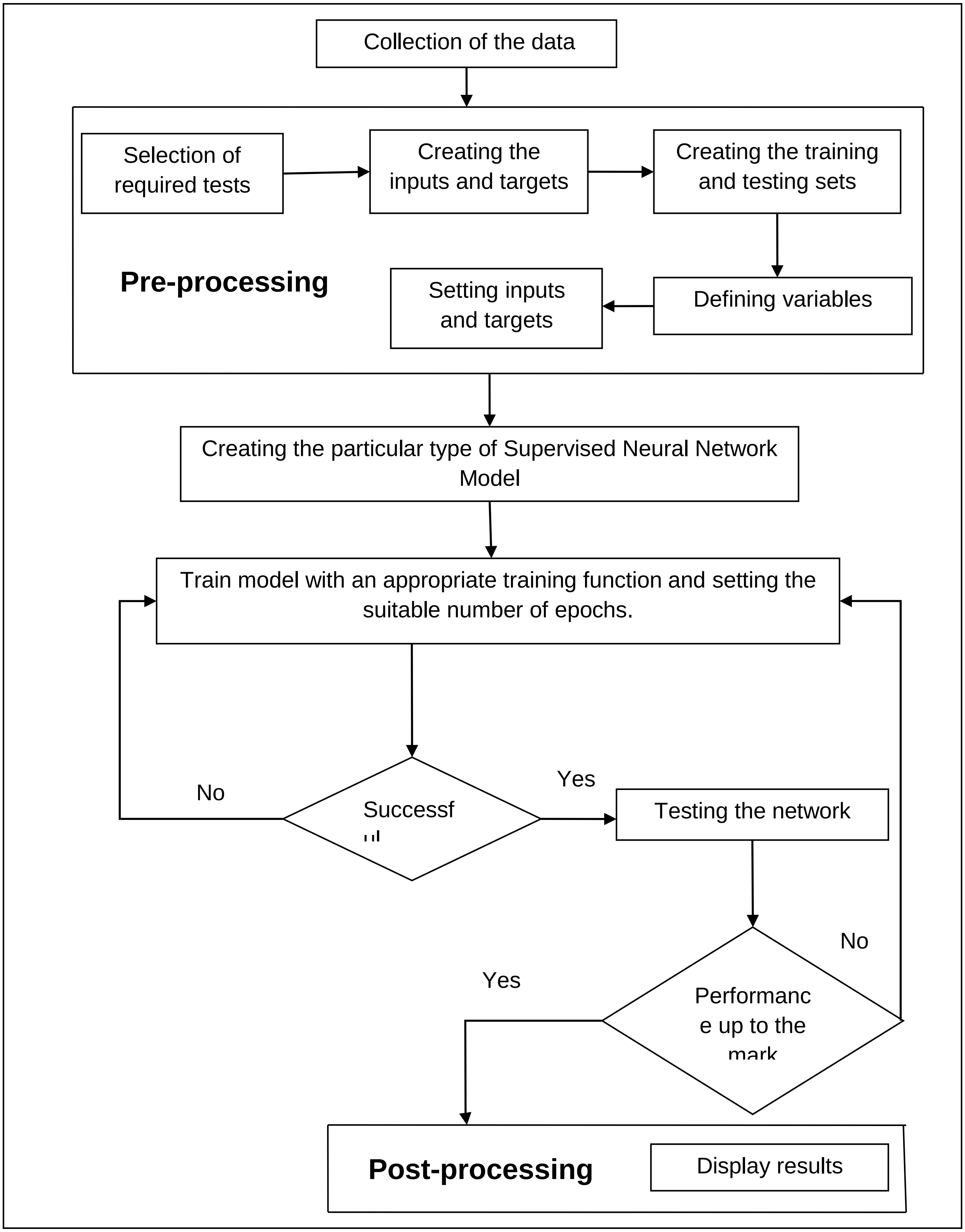


Figure 17: Precise view to create, train and test a supervised artificial neural network

3.2. Modeling & Designing of ANN Controller

3.2.1. Data acquisition

Training of an artificial neural network controller in this research requires some input-output data sets for its learning. Due to this reason our first step is to collect real time engine data of selected engine i.e. SI engine of 2NZFE and 1298CC of Toyota GLI as shown in table4:

Table 3: *Basic car features upon which Real time Engine Data is recorded*

Company	**TOYOTA**
Model	**Corolla GLI**
Engine type	**2NZ-FE**
Fuel system	**EFI with VVT-I**
Displacement	**1298CC**
Fuel type	**Gasoline**

In search of engine's real time data many EFI (Electronic Fuel Injection) workshops are visited. In the beginning, a private EFI workshop at Hasan Abdal city is visited. They use 'Universal Sensor' for collecting engine related real time data, this sensor showed many engine's parameters but failed to show exact AFR as required. In second attempt, NDC EFI workshop is visited at I-9 sector of Islamabad, in order to collect real time EFI engine's data. They use TECH-II sensor for getting engine's data but failed to get exact value of AFR in digits as this sensor only identifies whether air to fuel ratio is lean or rich.

In third trial we achieved our aim after visiting TOYOTA EFI workshop at I-9 sector Islamabad. They use IT-II tester and Intelligent Viewer software for obtaining real time data of EFI engines including exact AFR at different rpm, which then used in the training of neural network for the control of AF ratio in this research by taking MATLAB as a simulation tool.

Here are some snapshots of Intelligent Tester as shown in figure-18, while performing active tests on 2NZ-FE fuel injection engine of Corolla GLI with 1298CC displacement and VVT-I fuel system. The fuel used during experiment was' Unleaded regular gasoline

Sensor used to collect Engine Data
Toyota Intelligent Tester II

Figure 18: Snapshots of sensor during active test on 2NZ-FE fuel injection engine

Figure-19 shows snapshots of a software ‘Intelligent Viewer’ by attaching it with IT-II tester while performing active test on 2NZ-FE fuel injection engine captured at engine speed of 892 rpm, calculate load 49%, vehicle load 32.1%, MAF 3.68 gm/s, coolant temperature 40°C, intake air temperature 17°C, engine run-time 40s, injector (port) 3.32ms, injection volume of cylinder1 0.115ml, IGN advance 3.0deg and target air to fuel ratio 1.016.

Intelligent Viewer Software that helped to read data from Intelligent tester II

Snapshots of Data List

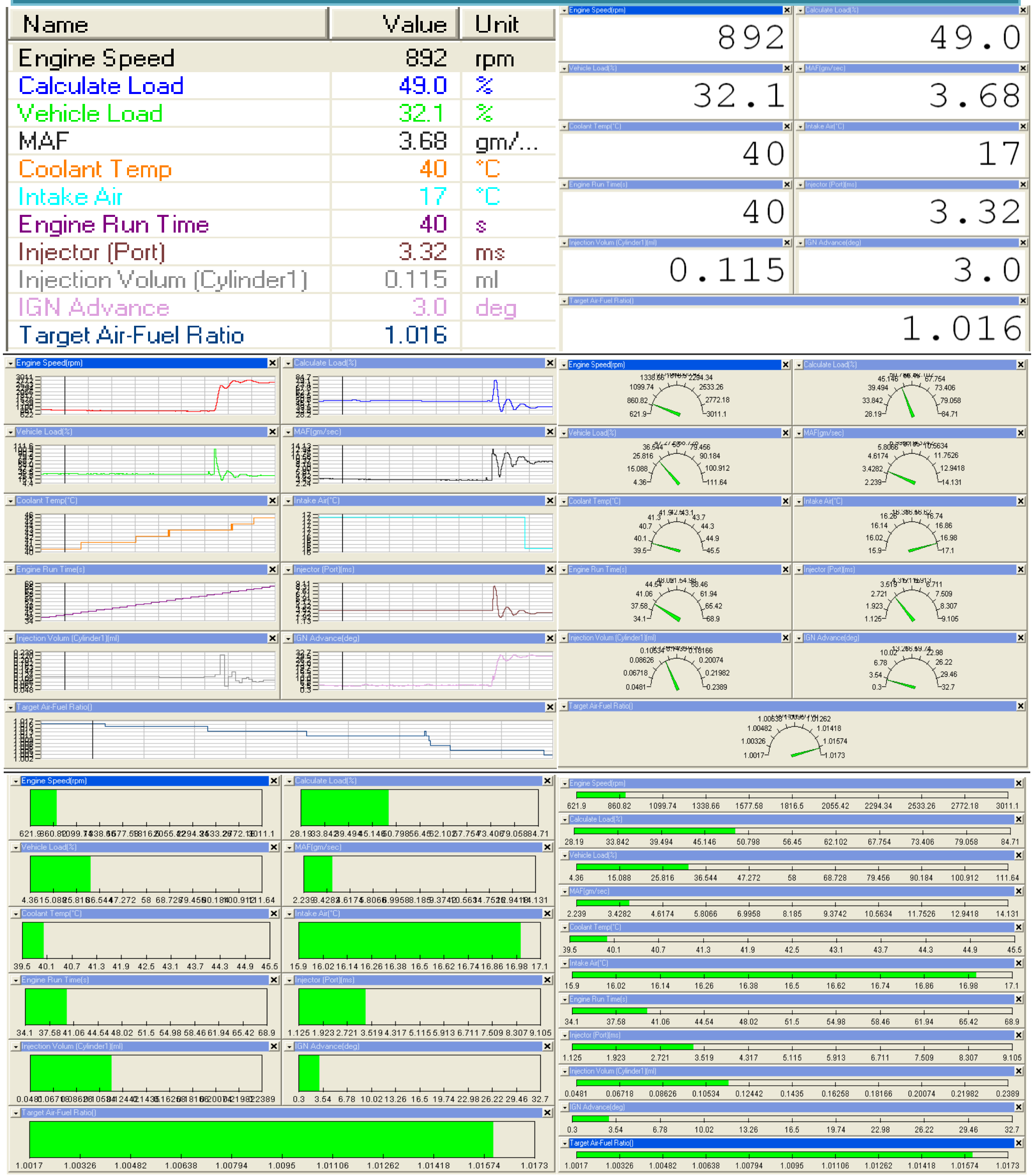


Figure 19: Snapshots of Intelligent viewer (software) by attaching it with sensor during active test on 2NZ-FE fuel injection engine

After performing 100 active tests with several different engines' speeds i.e. from 0 to 4339 rpm and by varying other related parameters, AF ratio is recorded. Total 10 parameters were selected upon which AF ratio is dependent and 100 data sets comprising each set of 10 inputs and 1 output were recorded during experiments. These 100 data sets are listed in table-5 below.

Table 4: 2NZ-FE Engine Real-time Data

Engine speed (rpm)	Calculate load (%)	Vehicle load (%)	MAF (g/s)	Coolant Temp (°C)	Air Intake Temp (°C)	Engine Run Time (s)	Injector (port) (ms)	Injection Volume (cylinder1) (ml)	IGN Advance (deg)	Target Air-fuel ratio (λ)
937	48.6	30.9	3.75	40	17	37	3.32	0.113	4	1.016
892	49	32.1	3.68	40	17	40	3.32	0.115	3	1.016
842	48.6	31.3	3.39	42	17	51	3.2	0.112	3.5	1.013
857	48.2	43.5	4.79	43	17	58	3.32	0.112	7.5	1.011
2689	37.6	24.3	8.46	45	16	64	2.56	0.085	29.5	1.005
2337	32.9	25.4	7.67	43	17	60	2.04	0.223	29.5	1.007
861	48.6	30.5	3.42	42	17	50	3.2	0.113	3.5	1.013
790	41.5	25.4	2.59	50	17	86	2.94	0.089	6	1.000
805	41.5	24.7	2.57	50	17	89	2.94	0.092	5.5	1.000
779	41.9	25.8	2.62	50	17	87	3.07	0.089	5.5	1.000
1958	22.7	13.3	3.4	46	16	70	1.4	0.08	17	1.003
970	36.4	26.2	3.31	47	16	71	2.81	0.055	7	1.000
2028	34.9	21.9	5.75	52	17	107	2.68	0.077	32	1.000
2046	34.1	21.5	5.75	54	17	114	2.56	0.076	32.5	1.000
807	41.5	23.9	2.51	50	18	94	3.07	0.09	5.5	1.000
715	43.1	25.4	2.34	63	19	238	2.81	0.093	5	0.999
734	39.2	23.1	2.18	62	19	223	2.81	0.08	7	1.000
4180	30.1	22.7	12.37	66	17	268	2.68	0.083	40.5	1.000

750	39.2	22.7	2.2	64	19	256	2.81	0.085	6.5	1.000
4139	29.4	21.5	11.62	70	17	280	2.68	0.077	40.5	1.000
4339	29	22.3	12.6	68	18	20	2.68	0.08	40.5	1.000
0	0	0	0.07	67	25	0	4.48	0.069	5	1.192
1653	32.9	16.4	3.53	68	23	1	2.56	0.174	13.5	1.134
1101	37.2	23.5	3.35	68	22	2	2.68	0.075	12.5	1.083
1817	23.9	14.9	3.5	79	20	19	1.66	0.05	24.5	1.000
2614	25.8	16	5.46	78	20	15	1.92	0.053	29	1.000
1565	30.9	31.7	6.39	79	20	18	2.81	0.047	29.5	1.090
839	48.6	30.9	3.34	42	17	51	3.2	0.112	3.5	1.013
843	48.6	30.5	3.34	42	17	53	3.2	0.111	3.5	1.011
837	48.2	30.1	3.26	43	17	54	3.2	0.111	3.5	1.011
826	48.6	30.1	3.26	43	17	56	3.2	0.111	4	1.011
835	48.2	30.1	3.25	43	17	57	3.2	0.112	4	1.011
2154	76.8	34.9	9.75	43	17	59	6.52	0.124	20.5	1.009
2412	49.4	37.6	11.73	43	17	60	3.39	0.113	25	1.005
2718	35.6	23.1	8.12	44	17	61	2.43	0.094	30	1.005
2691	43.9	30.9	10.73	44	17	62	3.2	0.104	27.5	1.005
2777	39.2	24.7	8.87	44	16	63	2.68	0.101	29.5	1.005
2646	39.6	27.4	9.34	45	16	64	2.81	0.089	29	1.005
2650	40.3	27	9.2	45	16	65	2.81	0.096	29	1.005
2670	40	26.6	9.15	45	16	66	2.81	0.095	29	1.003
2689	37.6	24.3	8.46	45	16	64	2.56	0.085	29.5	1.005
2802	43.1	27.8	10.09	44	16	63	3.07	0.111	28	1.005
2750	48.2	28.6	10.18	44	17	61	3.32	0.137	27	1.005
2337	32.9	25.4	7.67	43	17	60	2.04	0.223	29.5	1.007
827	48.6	30.9	3.32	43	17	56	3.2	0.11	4	1.011
833	48.2	30.5	3.29	43	17	54	3.2	0.11	4	1.011
836	48.6	30.1	3.28	42	17	52	3.2	0.111	3.5	1.011
842	48.6	31.3	3.39	42	17	51	3.2	0.112	3.5	1.013
861	48.6	30.5	3.42	42	17	50	3.2	0.113	3.5	1.013
851	49	30.9	3.39	42	17	49	3.32	0.113	3.5	1.013
858	49	30.1	3.34	41	17	47	3.32	0.114	3	1.013

853	49	30.9	3.42	41	17	45	3.32	0.113	3.5	1.015
870	49	30.9	3.46	41	17	44	3.32	0.112	3.5	1.015
812	42.3	25.4	2.68	49	17	80	3.07	0.09	5.5	1.000
812	41.9	25.4	2.67	49	17	81	3.07	0.09	5.5	1.000
828	41.9	24.7	2.65	49	17	82	3.07	0.091	5.5	1.000
810	41.5	24.7	2.59	49	17	83	2.94	0.091	6	1.000
798	41.9	25.8	2.65	49	17	84	3.07	0.089	6	1.000
809	41.9	25	2.62	49	17	85	3.07	0.089	5.5	1.000
806	41.5	24.7	2.57	50	17	86	2.94	0.091	5.5	1.000
790	41.5	25.4	2.59	50	17	86	2.94	0.089	6	1.000
784	42.5	26.2	2.67	50	17	87	3.07	0.09	6	1.000
815	42.3	24.7	2.6	50	17	88	3.07	0.092	5	1.000
817	41.9	24.7	2.62	50	17	89	2.9	0.092	5	1.000
793	41.5	25	2.57	50	17	89	2.94	0.09	6	1.000
789	41.5	25.4	2.59	50	17	90	2.94	0.09	6	1.000
802	41.9	25	2.59	50	18	91	3.07	0.09	5.5	1.000
779	42.3	25.8	2.62	50	17	90	3.07	0.09	6	1.000
817	41.9	24.7	2.62	50	17	89	2.94	0.092	5	1.000
803	42.7	25.4	2.64	50	17	88	3.07	0.089	5	1.000
790	41.5	25.4	2.59	50	17	86	2.94	0.089	6	1.000
806	41.5	24.7	2.57	50	17	86	2.94	0.091	5.5	1.000
810	41.5	24.7	2.59	49	17	83	2.94	0.091	6	1.000
846	42.3	25.4	2.78	48	17	77	3.07	0.089	5.5	1.000
825	42.7	26.6	2.82	48	17	76	3.07	0.089	5.5	1.000
869	41.9	25	2.81	48	17	74	2.94	0.088	6.5	1.000
832	42.3	25.8	2.79	47	17	73	2.94	0.088	8.5	1.000
1958	22.7	13.3	3.4	46	16	70	1.4	0.08	17	1.003
1171	31.3	22.7	3.45	46	16	70	2.43	0.052	9	1.000
875	39.6	26.6	3	47	17	71	3.07	0.055	5.5	1.000
970	36.4	26.2	3.31	47	16	71	2.81	0.055	7	1.000
815	41.1	26.2	2.76	47	17	73	2.94	0.088	9	1.000
854	42.3	25.8	2.84	48	17	73	3.07	0.088	8.5	1.000
842	41.9	25	2.75	48	17	74	2.94	0.088	6.5	1.000

829	41.9	25.4	2.75	48	17	75	2.94	0.089	7.5	1.000
821	42.3	25.8	2.75	48	17	76	2.94	0.089	6.5	1.000
2036	34.5	21.9	5.76	53	17	108	2.68	0.076	32	1.000
2047	34.5	21.5	5.76	53	17	110	2.56	0.076	32	1.000
2033	34.5	22.3	5.87	53	17	111	2.56	0.075	32	1.000
2039	34.5	22.3	5.87	53	17	111	2.56	0.076	32	1.000
2050	34.5	21.9	5.84	53	17	112	2.56	0.076	32.5	1.000
2040	34.5	21.5	5.71	53	17	113	2.56	0.075	32.5	1.000
2039	34.1	21.5	5.7	54	17	115	2.56	0.075	32.5	1.000
2042	34.1	21.5	5.68	54	17	115	2.56	0.075	32.5	1.000
2032	34.1	21.5	5.67	54	17	116	2.56	0.076	32.5	1.000
2039	34.1	21.5	5.68	55	17	117	2.56	0.075	32.5	1.000
2057	34.1	2.1	5.6	55	17	119	2.56	0.075	33	1.000
2045	34.5	2.5	5.71	54	17	114	2.68	0.075	32.5	1.000
0	0	0	1.04	80	26	0	4.22	0.054	5	1.186
2730	32.9	20	7.12	83	18	88	2.3	0.072	34.5	0.999

In this research, real time data of 100 active tests on 2NZ-FE fuel injection engine are examined by taking ten input parameters upon which air to fuel ratio is dependent and air to fuel ratio itself is taken as an output parameter. This data is divided into two categories one is for training the network and other is for testing them. Out of 100 real time data sets, 75 data sets are used to train network with three layers comprising of input layer, hidden layer and an output layer, whereas rest of 25 real time data sets are used to test them as shown in figure-20.

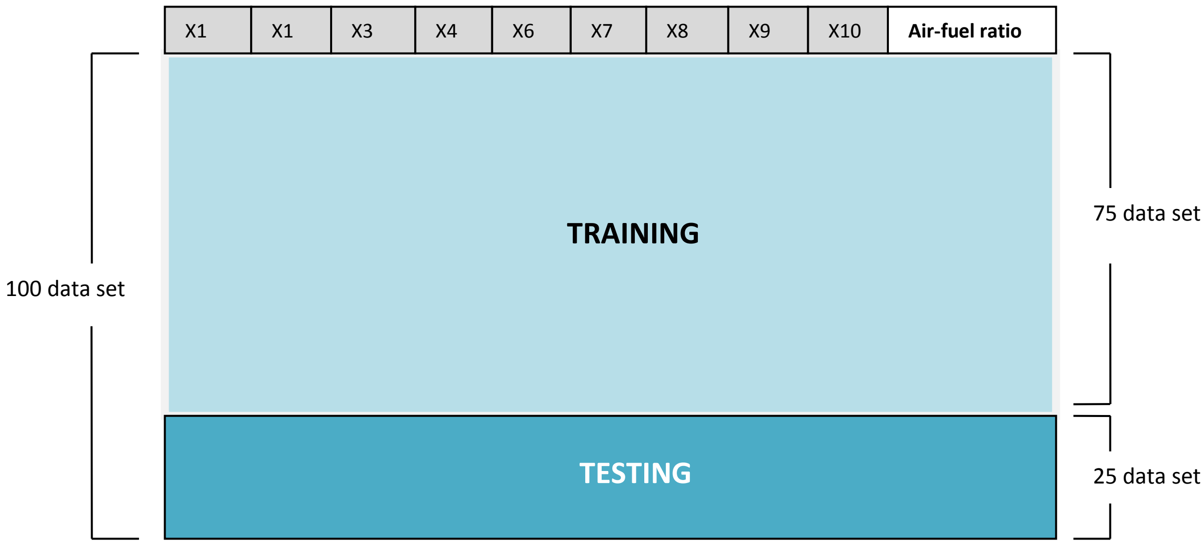


Figure 20: Training and testing of supervised artificial neural network

Ten input parameters used for training a network for controlling air to fuel ratio includes *Engine speed* which is basically a crank shaft speed taken in ‘rpm’, *Calculate load* i.e. an engine load according to the engine’s need in ‘%’, *Vehicle load* a permitted load or weight that a vehicle could bear according to the size in ‘%’, *MAF* mass air flow at the intake of engine in ‘g/sec’, *Coolant temperature* in °C, *Air intake temperature* in °C, *Engine run-time* running time of engine for the application to execute in ‘s’, *Injector (port)* injection period of fuel in ‘ms’, *Injection volume* of fuel in ‘ml’, *IGN advance* refers to the adjusting of point at which spark is produced measured in ‘degrees’ before top dead center.

On the other hand, *Target air to fuel ratio* is taken as an output parameter which represents the ratio of air to fuel present in an internal combustion engine measured in Lambda (λ) by Lambda meter. Lambda meter measures air-fuel ratio by using oxygen sensor in the exhaust gas. Data set used to train network with three layers comprising of input layer, hidden layer and an output layer shown in figure-21.

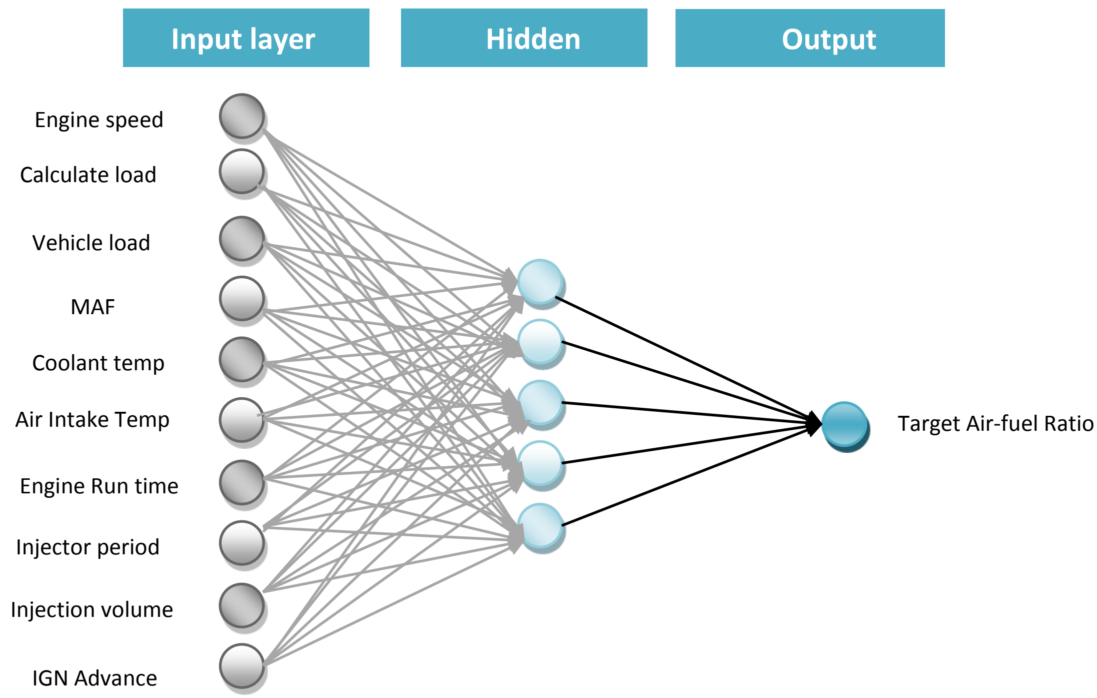


Figure 21: View of Multilayer Feed Forward Artificial Neural

3.2.2. Training

Once data acquisition is done for a particular application, network is ready to be trained. There are two approaches to train the network - supervised and unsupervised. Supervised training is based on a mechanism of providing desired output either by manually "grading" the network's performance or by providing the desired outputs with the inputs to the network. Unsupervised training is used, where the network has to make sense of the inputs without outside help. A supervised training is utilized by vast bulk of networks these days.

In supervised training or learning, both inputs and outputs are provided. The network after processing inputs compares its resulting outputs against the desired outputs. Errors causing the system to adjust the weights are then propagated back through the system for controlling network over and over until it becomes zero.

Many layered networks with multiple nodes have capability to memorize data. In order to monitor the network, supervised training holds back a set of data to test the system after undergone its training to check if the system is simply memorizing its data in some non-significant way.

To create a successful network, designer needs to review different neural parameters by varying the inputs and outputs, number of layers, number of elements per layer, connections between the layers, the summation, transfer, and training functions. Training algorithms (law) governs the rule of training network, which is another part of designer’s creativity. There are many algorithms exist for network training but the most well-known training algorithm known as

‘Gradient Descent Back propagation with Adaptive learning rate’ is selected on the basis of comparison work done in literature on different ANN training algorithms [46]

3.2.2.1. Gradient Descent Back Propagation with Adaptive Learning

Gradient descent is an optimization algorithm used to find local minima by taking steps proportional to the negative of the gradient, whereas “gradient descent back propagation with adaptive learning” is an improved training algorithm [47]. It not only responds to the local gradient but also to the recent trends in an error surface. Any network as long as its weight, net input and transfer functions have derivative functions can be trained by training algorithm of Gradient descent back propagation with momentum. The two main parameters of this algorithm are learning rate and momentum[48-49]. It acts like a low pass filter. Momentum allows network to ignore small features in the error surface and helps to slide through minima. Without momentum a network can get stuck in a shallow local minimum. Inclusion of momentum term has increased the rate of convergence in gradient descent. The main features of this algorithm are its low storage requirements, in expensive and require less computational efforts and time.

This back propagation algorithm refers to a method of calculating error or desired response for each neuron by using gradient equation-10

$$g = \frac{\Delta E}{\Delta \theta} \qquad (10)$$

where,

$\boldsymbol{\theta}$= a vector composed of all network weights

E=is the output error i.e. a difference between actual and predicted output for N possible data points.

$$E = \frac{1}{N}\sum_{k=1}^{N}(y^*(k) - y(k))^2 \qquad (11)$$

Updating weights strategy of gradient method is shown in equation-11.

$$\Delta(k)=\eta g \qquad (12)$$

Here,

η= learning rate

Updating weights strategy of gradient descent with momentum and with learning rate is shown in equation-12. This method is considered as more reliable method.

$$\Delta(k)=\eta g - \alpha\, \Delta\boldsymbol{\theta}(k-1) \qquad (13)$$

This algorithm increases its step size when two successive weight changes are in the same direction and decreases when weight changes are in reverse direction. Larger learning rate without instability can be achieved with this algorithm.

Updating weight criteria of this algorithm depends upon momentum and learning rate in such a manner that if new error exceeds the old error by more than predefined ratio, the new weights and biases are discarded and if ratio of new error and old error remains under that ratio it will continue to update weights. Similarly, if new error is less than the old one, learning rate increases by multiplying it with predefined learning incremental value.

Computational equation of ‘Gradient descent back propagation with adaptive learning rate’:

$$dX = mc * dXprev + lr * (1 - mc) * dperf/dX \qquad (14)$$

$$dX = w_{k+1} - w_k \tag{15}$$

$$dXprev = w_k - w_{k-1} \tag{16}$$

where,

mc =momentum

lr =Learning rate

$dXprev$ =Previous change to the weight or bias

$dperf$ = derivative of performance with respect to the variable x of weights and bias

1. Set momentum which is the maximum growth rate
2. Set learning rate
3. Calculate derivative of performance with respect to the variable x of weights and bias
4. Calculate derivative of previous change with respect to weight and biases
5. Compute dX
6. Update weights by using dX
7. If stopping conditions does not achieve go to step3
8. Else stop training

Training will stop when any of the condition mentioned below occurs:

1. Maximum number of iterations or 'Epochs' is reached
2. Maximum number of time exceed
3. Performance is minimized to the goal.
4. The performance gradient falls below min grad.

Chapter 4

Numerical Simulation

In this chapter, generalization (predictive) ability of ANN model is analyzed by doing numerical simulation with ‘Gradient descent back propagation with adaptive learning rate’ algorithms. Out of 100 data sets, 75 are trained whereas rest of 25 test data are tested and verified by running it thrice on MATLAB. Desired and predicted values of air to fuel ratio are summarized in the tables below along with the parameters and progress achieved as a result of neural training. Numbers of Epochs are representing number of iterations in which system undergoes. Time mentioned here is the training time of ANN. In this study 25 sets are tested after training, hence we can say here MSE is the square error of 25 values i.e. (n=25)

In current work three graphs are achieved after every trial of training to analyze ANN with different aspects. These graphs include ‘Comparison graph between Actual Outputs and Predictions’.

The MSE is defined as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Predicted\ Output - Actual\ Output)^2 \qquad (17)$$

MSE (Mean Square Error) = Average or mean of square of errors observed after each iteration.

"N" is number of data sets which are tested after neural training. "Predicted Output" is the value of air to fuel ratio predicted by designed neural network after each iteration. "Actual Output" is the value of air to fuel ratio observed in real system by running engine at different speed.

In present work we have performed 3000 iterations for the training of designed neural network. Real time engine data is recorded in term of air to fuel ratio and others parameters on which air to fuel ratio is dependent. In this research total 100 data sets are collected including 11 parameters in each set. Among 11 parameters, 10 are taken as input of artificial neural network and 1 parameter i.e. air to fuel ratio as output parameter. 75 data sets are used to train designed neural network. By generalizing their trend, neural network predicted 25 data sets which are compared with 25 actual outputs. Error of each tested data sets is calculated individually by taking difference between neural predicted and actual outputs, mean square error of all 25 tested data sets is then calculated to observe overall neural training effect.

4.1. Selection of optimize parameters in designing of Neural Network Model

An artificial neural network consists of large number of different frame work. In many instances, obstacle is in estimating nonlinear static mapping f(x) with $f(x)_{NN}$. The most functional neural networks in function approximation are Multilayer Layer Perceptron (MLP). Here in this research we have focused our attention on multi nodes MPL networks. MLP architecture consists of an input layer, one or several hidden layers and an output layer. Neural network sets random weights in the beginning according to the inputs received and then multiply weights with inputs.

Number of inputs are $X_{1,2,\ldots k}$ given to the node are multiplied by weights $W_{1,2,\ldots k}$ and summed up together with constant bias term Θ_i .Bias is a scalar term that helps to shift outputs of activation function either right or left along x-axis. The resulting N_i is the input to the activation function g which then decides output.

Z=Actual outputs

$X_1, X_2,\ldots\ldots X_K$=set of inputs

$W_1,W_2,\ldots\ldots W_K$=Initial set of weights

$$X_1W_1+X_2W_2+\ldots.X_KW_K+\Theta_i=S \tag{18}$$

T= Threshold value (a terminating parameter)

If S<T give neural output N

If S>T stop training

$$E\ (\text{Error})=Z-N \tag{19}$$

R=learning rate

$$D=E*R \tag{20}$$

$$\text{Final weights}=W_{1,2,3..k}=X_{1,2,\ldots k}*D \tag{21}$$

Mathematical model of single mode MLP network is shown in figure-22 below with input signals, bias input, summing junction, activation function and an output generated as a result of neural training.

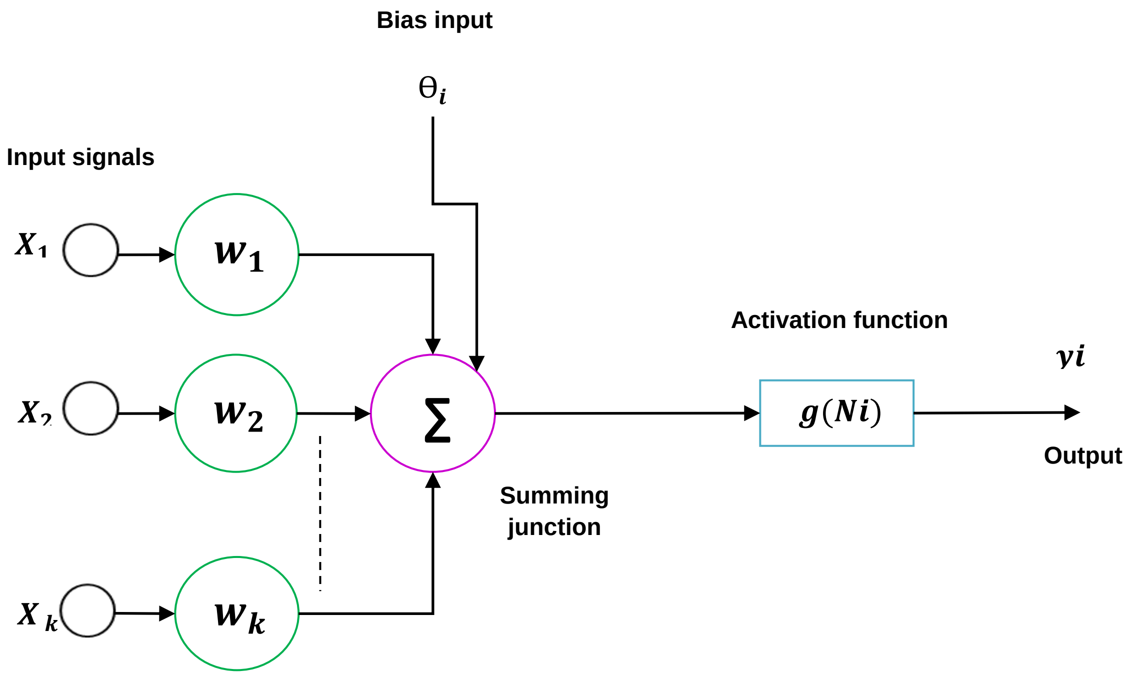


Figure 22: MATHEMATICAL MODEL OF MLP NETWORK

The basic equations describing the dynamics of each neuron are:

$$Ni = \sum_{j=1}^{K} W_j X_j + \Theta_j \qquad (22)$$

Multiple nodes MLP network is designed in this work with large number of connecting nodes that work both in parallel and series for estimating outputs. Here three layers is taken with one input layer consisting of 10 inputs, one hidden layer with 5 neurons and an output layer with only one neuron. Hyperbolic tangent function known as a sigmoid tangent function 'tan sigmoid' is used as an activation function 'g' between neural layers. 3000 times iterations are performed with error tolerance or a goal of 1E-05 as stopping criteria to stop neural network training. Learning rate is set as 0.09 along with its incremental learning rate of 1.07 and detrimental rate of 0.1. Learning rate is basically very important parameter in training of any neural network. It increases when new error becomes less than the previous one and decreases if error becomes greater than the older one after performing each iteration. Minimum performance gradient defines when to stop training if error becomes less than predefined ratio which set as 1e-10 and momentum constant is set as1e-4. Parameter momentum constant decides weights and biases if new error overshoots the previous one more than momentum constant, new weights and biases are dropped otherwise it retain its weights and biases.

Network is trained by using a function of feed forward neural network on MATLAB. This function takes arbitrary initial values for updating weights. Due to this fact we get three different values of neural outputs in different trials with same parameters.

4.2. Training results by varying number of epochs and number of nodes in hidden layer

Selection of appropriate number of epochs and number of nodes in hidden layer is an important task in designing of best neural model. Training is performed at epochs (iterations) 500, 1000, 3000 and 5000 by varying number of nodes in hidden layer from 2, 5, 10, 15, 20, 25 and 30. Keeping sigmoid as a tangent function, one input layer, one hidden layer and one output layer with gradient descent back propagation with adaptive learning rate as a training algorithm mean square error (MSE) is investigated thrice at each iteration. Conclusion is drawn regarding best number of iterations and best number of nodes in hidden layer by observing least average MSE after three trials.

Table 5: Training reults at 500 Epochs

	500 Epochs						
No of Nodes	**2**	**5**	**10**	**15**	**20**	**25**	**30**
MSE(trial-1)	4.9570e-004	0.0016	0.0017	0.0012	1.6536e-004	0.0015	0.0021
MSE(trial-2)	0.0014	6.4055e-004	0.0024	2.2356e-004	4.1181e-004	0.0011	0.0048
MSE(trial-3)	0.0013	0.0011	0.0011	0.0013	5.9884e-005	0.0013	0.0028
Average MSE	1.065e-003	1.113e-003	1.733e-003	9.1186e-004	2.123e-004	1.30e-003	3.233e-003

When neural network is trained for 500 iterations (epochs) by varying number of neurons in hidden layer from 2, 5, 10, 15, 20, 25 and 30 MSE is investigated thrice and average mean square error is calculated for each number of node. It is observed that at 500 epochs with 20 nodes in hidden layer least MSE is achieved as shown in above table-6.

Table 6: Training reults at 1000 Epochs

	1000 Epochs						
No of Nodes	**2**	**5**	**10**	**15**	**20**	**25**	**30**
MSE(trial-1)	3.5937e-004	4.1680e-004	0.0015	3.2500e-005	4.7116e-004	0.0014	1.0274e-005
MSE(trial-2)	0.0014	0.0017	5.9913e-004	4.8886e-004	0.0014	1.2507e-005	0.0017
MSE(trial-3)	0.0015	0.0015	0.0020	3.2743e-004	0.0016	1.3574e-004	3.0159e-004
Average MSE	1.086e-003	1.2056e-003	1.366e-003	2.829e-004	1.1570e-003	5.160e-004	6.706e-004

Similar observations are taken at 1000 epochs with number of neurons or nodes varying from 2, 5, 10, 15, 20, and 25 to 30 in hidden layer. Least average mean square error of 2.829e-004 is observed with 15 nodes in hidden layer after 1000 iterations shown in table-7.

Table 7: Training reults at 3000 Epochs

3000 Epochs							
No of Nodes	**2**	**5**	**10**	**15**	**20**	**25**	**30**
MSE(trial-1)	0.0013	1.2210e-004	0.0016	0.0024	0.0013	7.7053e-005	6.8545e-004
MSE(trial-2)	0.0014	2.8811e-004	0.0025	5.4219e-004	4.5996e-004	2.1458e-004	7.9746e-004
MSE(trial-3)	0.0013	2.1642e-005	0.0032	5.6196e-005	4.7225e-004	0.0033	0.0013
Average MSE	1.333e-003	1.439e-004	2.433e-003	9.994e-004	7.4405e-004	1.1972e-003	9.276e-004

After neural network training with 2, 5, 10, 15, 20, 25 and 30 nodes, least average mean square error of 1.439e-004 is achieved after three trials for each number of nodes shown in above table-8.

Table 8: Training reults at 5000 Epochs

	5000 Epochs						
No of Nodes	**2**	**5**	**10**	**15**	**20**	**25**	**30**
MSE(trial-1)	0.0016	8.4167e-004	3.8953e-004	0.0014	8.5304e-004	0.0018	7.7466e-004
MSE(trial-2)	0.0018	1.9559e-005	0.0014	0.0022	0.0020	9.9412e-004	7.6543e-005
MSE(trial-3)	0.0013	0.0012	3.2022e-004	7.7291e-004	0.0010	9.1734e-006	0.0015
Average MSE	1.566e-003	6.870e-004	7.032e-004	1.457e-003	1.284e-003	9.344e-004	7.843e-004

With 5000 epochs and with 5 nodes in hidden layer least average mean square error of 6.870e-004 is observed after three trials as shown in table-9

From four above mentioned observation tables-6 to table-9 drawn after training neural network model with 'gradient descent back propagation with adaptive learning rate' and by varying number of nodes in hidden layer i.e. with 2, 5, 10, 15, 20, 25 and 30 at 500, 1000, 3000 and 5000 epochs. Mean square errors are investigated thrice for each iteration, average mean square error is then compared. It is observed that at 500, 1000, 3000, 5000 epochs least average mean square error of 2.123e-004, 2.829e-004, 1.439e-004, 6.870e-004 with 20, 15, 5, 5 nodes in hidden layer is achieved respectively.

Table 9: Comparison table by varying nodes in hidden layer with 3000 epochs

ACTUAL OUTPUTS	3000 EPOCHS PREDICTED OUTPUTS					
	NODES=5 IN HIDDEN LAYER	*NODES=10 IN HIDDEN LAYER*	*NODES=15 IN HIDDEN LAYER*	*NODES=20 IN HIDDEN LAYER*	*NODE=25 IN HIDDEN LAYER*	*NODES=30 IN HIDDEN LAYER*
0.9986	1.0000	1.0534	1.0317	1.0647	1.0063	1.0452
1.0000	0.9986	1.0451	1.0277	1.0630	1.0065	1.0486
1.0030	0.9996	1.0248	1.0005	1.0083	1.0023	1.0124
1.0000	0.9982	1.0166	1.0008	0.9964	1.0024	1.0022
1.0000	1.0002	1.0158	1.0016	0.9979	1.0059	1.0018
1.0000	0.9986	1.0148	1.0013	0.9967	1.0034	1.0015
1.0000	0.9987	1.0408	1.0276	1.0630	1.0063	1.0450
1.0000	0.9987	1.0456	1.0323	1.0672	1.0065	1.0453
1.0000	0.9986	1.0528	1.0319	1.0650	1.0065	1.0446
1.0000	0.9986	1.0501	1.0309	1.0686	1.0068	1.0459
1.0000	0.9987	1.0529	1.0368	1.0686	1.0071	1.0454
1.0000	0.9986	1.0126	1.0005	1.0195	1.0017	1.0088
1.0000	0.9987	1.0130	1.0005	1.0185	1.0016	1.0101
1.0000	0.9987	1.0125	1.0005	1.0169	1.0016	1.0104
1.0000	0.9987	1.0133	1.0005	1.0191	1.0016	1.0112
1.0000	0.9987	1.0130	1.0005	1.0190	1.0016	1.0104
1.0000	0.9987	1.0121	1.0005	1.0168	1.0016	1.0091
1.0000	0.9986	1.0127	1.0005	1.0189	1.0016	1.0087
1.0000	0.9986	1.0127	1.0005	1.0188	1.0016	1.0086
1.0000	0.9986	1.0134	1.0005	1.0215	1.0016	1.0093
1.0000	0.9986	1.0136	1.0005	1.0207	1.0015	1.0088
1.0000	0.9992	1.0162	1.0004	1.0182	1.0025	1.0014
1.0000	0.9992	1.0156	1.0004	1.0193	1.0024	1.0012
1.1860	1.1859	1.0048	1.2054	1.1621	1.1628	0.9978
0.9990	0.9990	1.0615	1.0018	1.0131	1.0047	1.0583
MSE	**2.0081e-006**	**0.0023**	**2.9216e-004**	**0.0014**	**0.0016**	**0.0022**

Table 10: Comparison table by varying number of epochs with 5nodes in hidden layer

ACTUAL OUTPUTS	WITH 5NODES IN HIDDEN LAYER PREDICTED OUTPUTS ITERATIONS (EPOCHS)			
	500	1000	3000	5000
0.9986	1.0124	1.0924	1.0000	1.0569
1.0000	1.0124	1.0689	0.9986	1.0605
1.0030	1.0120	1.0065	0.9996	1.0184
1.0000	1.0123	0.9979	0.9982	0.9984
1.0000	1.0124	0.9984	1.0002	0.9985
1.0000	1.0124	0.9980	0.9986	0.9985
1.0000	1.0124	1.0571	0.9987	1.0714
1.0000	1.0124	1.0710	0.9987	1.0630
1.0000	1.0124	1.0914	0.9986	1.0603
1.0000	1.0124	1.0856	0.9986	1.0716
1.0000	1.0125	1.0927	0.9987	1.0737
1.0000	1.0122	1.0020	0.9986	1.0051
1.0000	1.0122	1.0019	0.9987	1.0049
1.0000	1.0122	1.0015	0.9987	1.0044
1.0000	1.0122	1.0018	0.9987	1.0048
1.0000	1.0122	1.0018	0.9987	1.0048
1.0000	1.0122	1.0015	0.9987	1.0043
1.0000	1.0122	1.0015	0.9986	1.0039
1.0000	1.0122	1.0015	0.9986	1.0039
1.0000	1.0122	1.0017	0.9986	1.0043
1.0000	1.0122	1.0015	0.9986	1.0035
1.0000	1.0128	1.0094	0.9992	1.0117
1.0000	1.0128	1.0100	0.9992	1.0117
1.1860	1.0123	1.0732	1.1859	1.0529
0.9990	1.0125	1.0123	0.9990	1.0000
MSE	**0.0014**	**0.0024**	**2.0081e-006**	**0.0019**

By observing neural outputs with different number of iterations and by varying number of nodes in hidden layer, results are summarized in above tables. Out puts are observed on the basis of mean square error that tells the mean square error of 25 outputs which are obtained as a result of neural network training. On the basis of these observations 3000 epochs with 5nodes in hidden layer by using ‘gradient descent back propagation with adaptive learning rate’ is selected for the best training of neural network model with least mean square error of 2.0081e-6 as shown in table-11.Neural network is first trained and then tested thrice to see the effect of overall performance. Outputs of 25 tested data sets are plotted against system’s actual values.

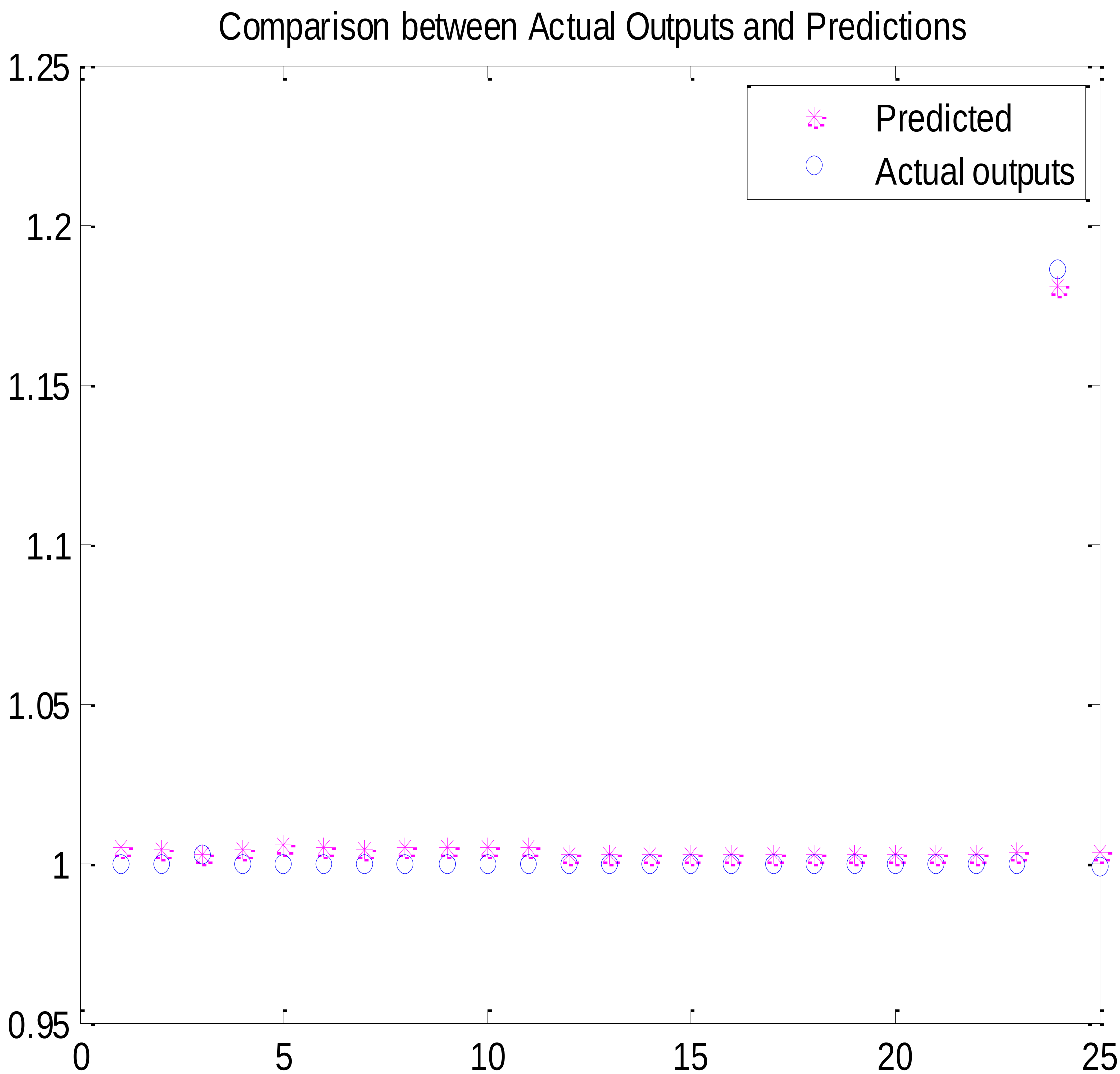


Figure 23: (TRIAL-1) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate

In comparison graph achieved as a result of trail-1, third predicted and actual outputs are exactly overlapping where at other points slight deviation is observed with MSE of 1.4728e-005 after 44seconds of neural training shown in Figure-23

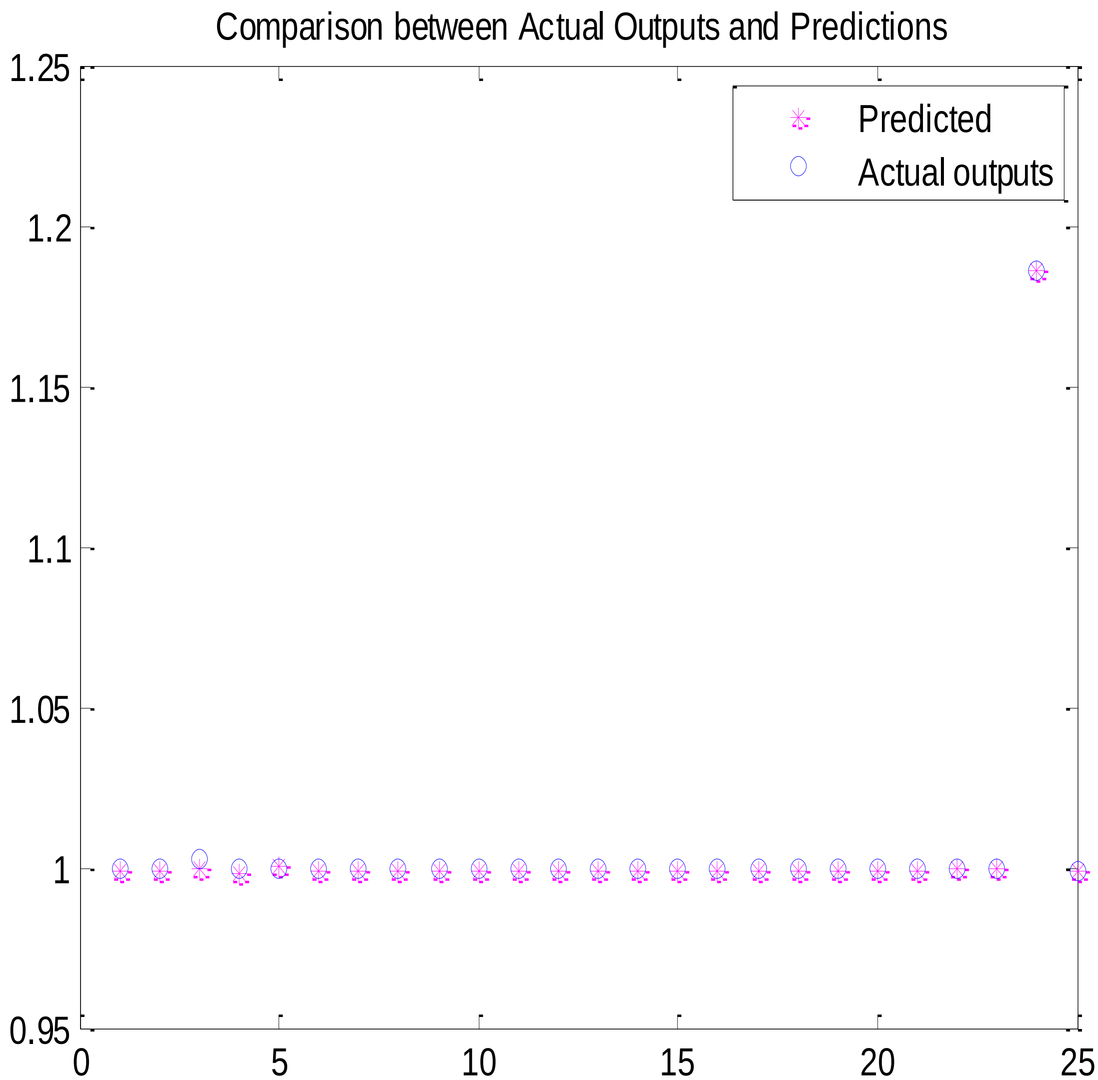


Figure 24: (TRIAL-2) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate

In comparison graph achieved as a result of trail-2 all predicted and actual outputs are exactly overlapping other than point three where exact overlapping is not observed. MSE of 1.594e-006 is noticed in after 50seconds of neural training shown in figure-24.

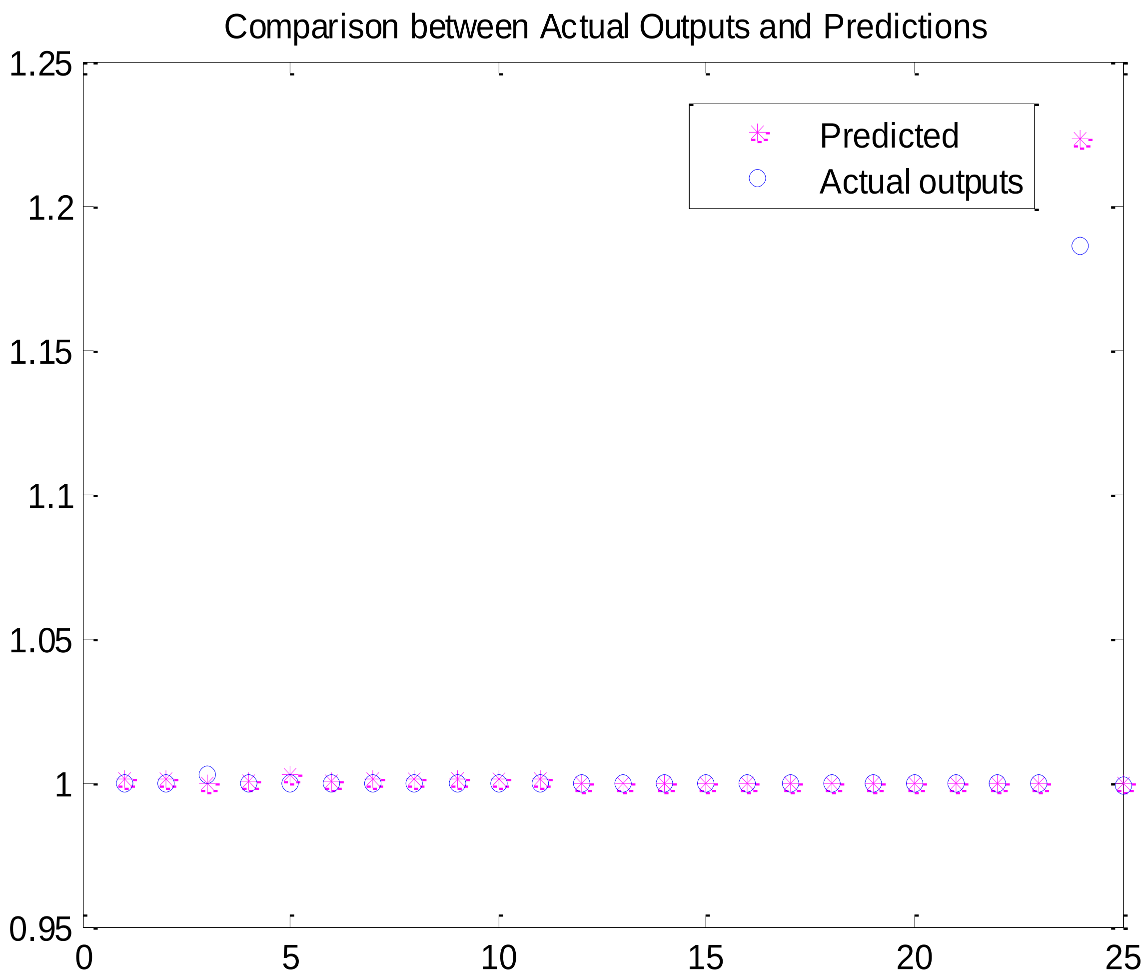


Figure 25: (TRIAL-3) Comparison results by using Gradient Descent Back Propagation with Adaptive learning rate

In comparison graph achieved as a result of trail-3 deviation is observed at 24^{th} point with MSE of 5.6941e-005 after 45seconds of training shown in figure-25. Three above mentioned figures are the 'comparison graphs' plotted between 25 testing data sets on x-axis where as predicted and actual outputs of AF ratio is specified on y-axis with respect to the coordinate of x-axis by applying training algorithm of 'gradient descent back propagation with adaptive learning rate' in neural model. 'Predicted output' of air to fuel ratio is represented with '*' where as 'Actual output' is represented with 'o'. At some places in these graphs both '*' and 'o' are exactly overlapped showing good predictions whereas at other places they both are slightly or even not overlapped shows unsatisfactory predictions with respect to the actual outputs.

Table 11: Gradient Descent Back Propagation with Adaptive Learning Rate

DESIRED VALUES	PREDICTED VALUES		
	TRIAL 1	TRIAL 2	TRIAL 3
1.0000	1.0050	0.9988	1.0010
1.0000	1.0045	0.9989	1.0010
1.0030	1.0029	0.9996	0.9996
1.0000	1.0040	0.9983	1.0001
1.0000	1.0057	1.0003	1.0026
1.0000	1.0046	0.9997	0.9994
1.0000	1.0028	0.9999	1.0031
1.0000	1.0028	0.9999	1.0023
1.0000	1.0042	0.9999	1.0021
1.0000	1.0028	0.9998	1.0023
1.0000	1.0050	0.9999	1.0025
1.0000	0.9999	0.9998	0.9993
1.0000	0.9999	0.9999	0.9993
1.0000	0.9999	0.9998	0.9993
1.0000	1.0000	0.9999	0.9995
1.0000	0.9999	0.9999	0.9995
1.0000	0.9998	0.9998	0.9995
1.0000	0.9999	0.9998	0.9995
1.0000	0.9999	0.9998	0.9995
1.0000	0.9999	0.9998	0.9995

1.0000	0.9999	0.9997	0.9995
1.0000	1.0030	0.9994	0.9994
1.0000	1.0031	0.9994	0.9994
1.1860	1.1806	1.1857	1.2233
0.9990	1.0034	0.9991	0.9995
PARAMETERS			
	TRIAL 1	TRIAL 2	TRIAL 3
EPOCHES	3000	3000	3000
TIME	0:00:44	0:00:50	0:00:45
MSE	1.4728e-005	1.594e-006	5.6941e-005
CORRELATION COEFFICIENT	0.9997	0.9998	0.9995

Above mentioned table-12 includes 25 ‘Actual’ and 25 ‘Predicted’ values of air to fuel ratio by using an algorithm of ‘Gradient descent back propagation with adaptive learning rate’. Training is performed thrice; mean square error and total training time are recorded with 3000 iterations (epochs). First column of table consists of 25 actual values of AFR whereas second, third and fourth column consists of 25 predicted values, number of epochs at which training is done, MSE and total time taken for training in trail-1, trial-2 and trial-3 respectively as shown in table-12. In every trial neural predicted results gave different values due to the fact that *newff* function is used on MATLAB that took random initial weights in generating outputs. Among three trials least MSE of 1.594e-006 is observed in trial-2. Elaborated results of trial-2 are discussed below in detail.

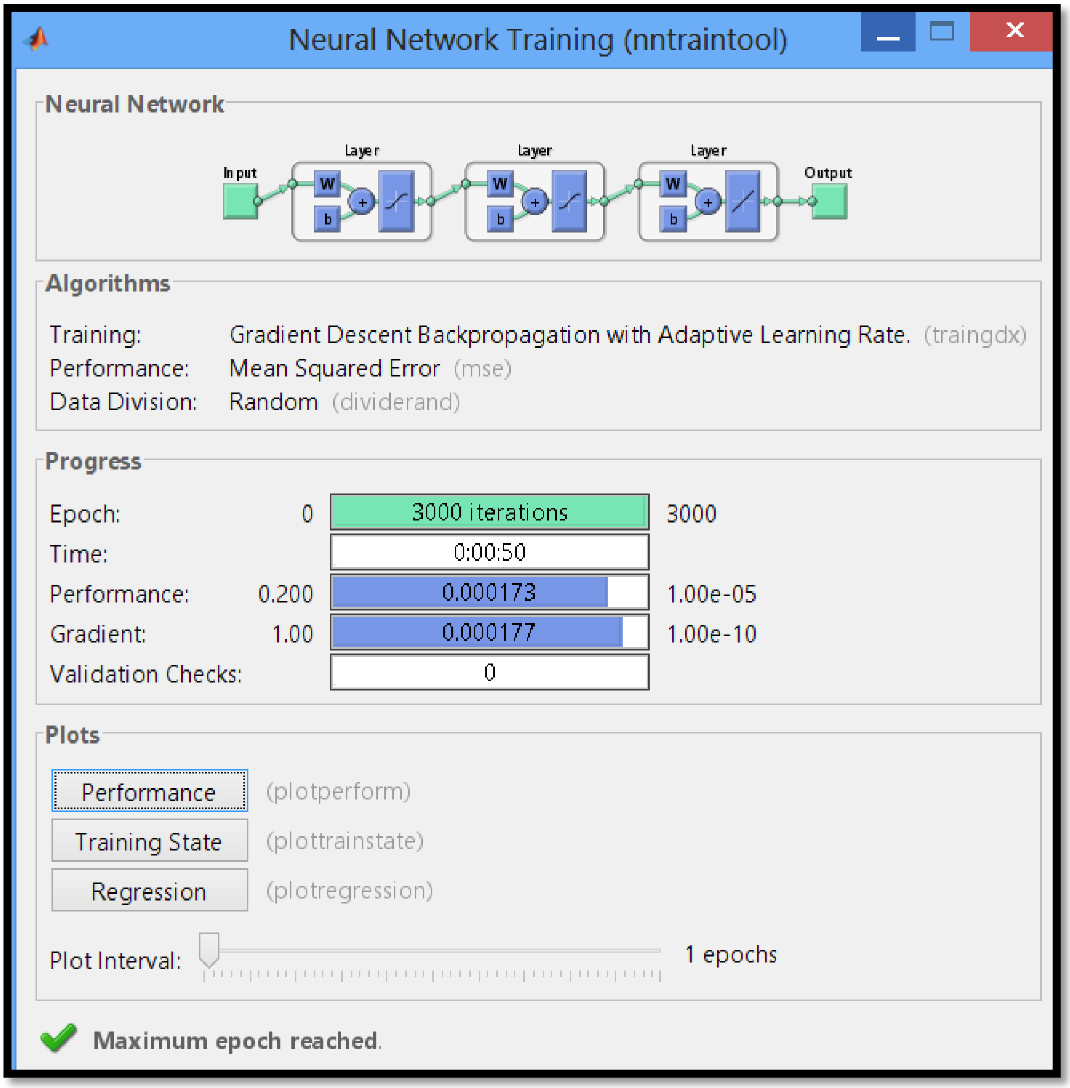


Figure 26: Trained ANN output window obtained in trial-2

The window in figure-26 shows specifications about neural network training. Model comprises of three layers i.e. input layer, hidden layer and an output layer. Ten inputs are given to the network and one output is achieved. On the basis of comparison by varying number of nodes in hidden layers and by varying number of iterations, 5 nodes are selected to place in hidden layer with 3000 epochs. This training is topped when all 3000 iterations have been done. Training algorithm "Gradient Descent back propagation with Adaptive Learning rate" is used for training dataset. The major advantage of this algorithm is that during training when two successive weight changes are in same direction, its step size increases and decreases when successive weight change are in reverse direction. It also allows larger learning rate without instability. Performance is analyzed on the basis of mean square error and validation failure occurs when performance fails to decrease. Performance or goal of 1e-005 is set as stopping criteria but in this training performance 0.000173 and gradient 0.000177 is achieved with 3000 iterations after 50seconds of training.

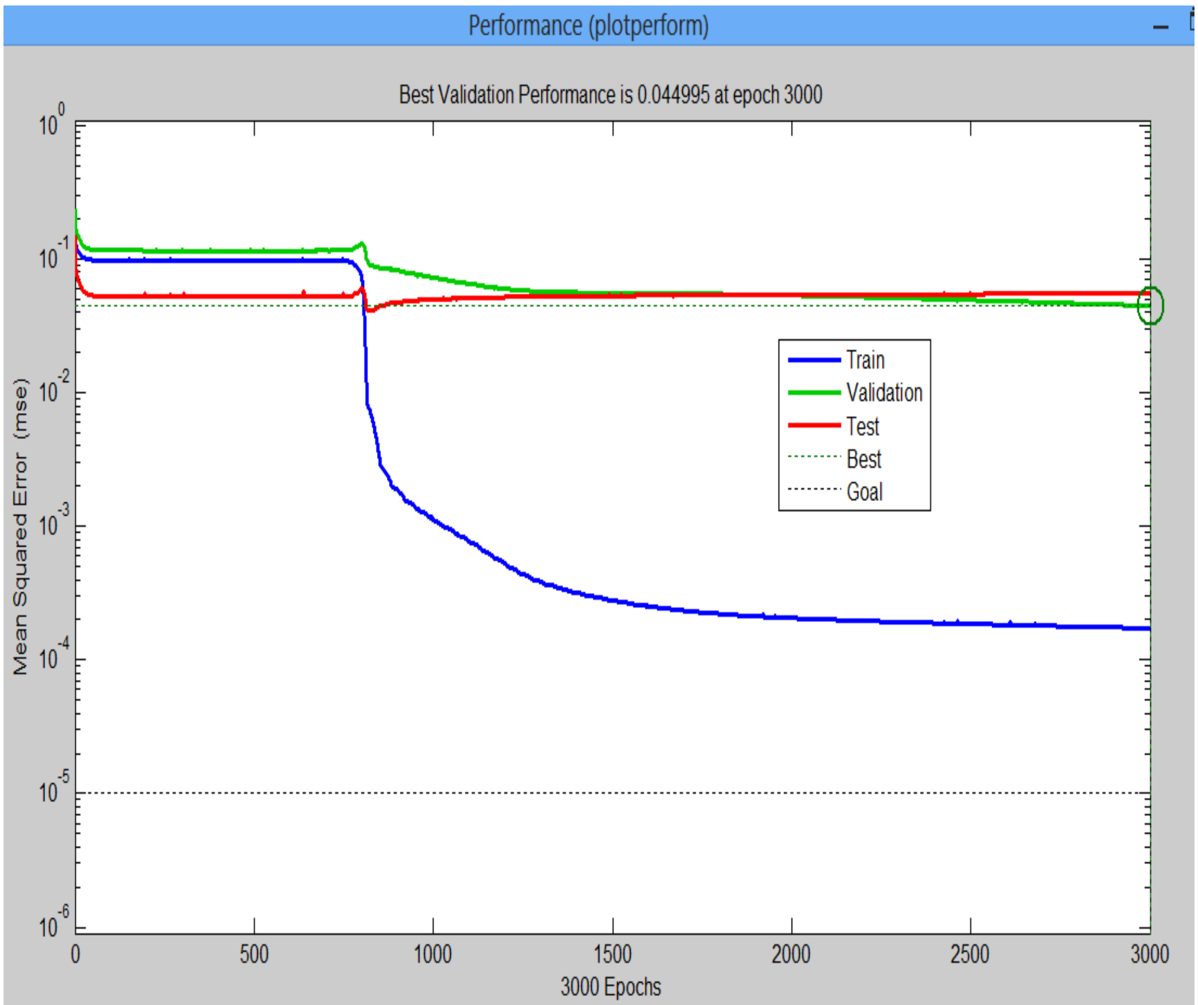


Figure 27: Performance graph (TRIAL-2)

The main advantage of neural network is its aptitude to generalize. This means that trained network could arrange data from same class as training data that has never seen before. In real world implementation only small part of all possible datasets/ patterns may use for the generalization of artificial neural network. In order to achieve best generalization, dataset need to be split into three sets i.e. training sets, validation set and test set. Training set as shown in figure-27 graph with blue line is used to train network, during learning phase error of this dataset is minimized, validation set as shown with green line in performance graph is used to determine performance of neural network on patterns that are not trained during learning, test set as shown with red line is used to determine overall performance of neural network. Dotted line represents goal, which is set at 1e-005 as a stopping criteria.

Figure-27 is a performance graph as a result of trial-2 by using algorithm of ‘gradient descent back propagation with adaptive learning rate’ at 3000 epochs in which more accurate results are achieved plotted between mean square error on y-axis and number of iterations or number of epochs on x-axis. This figure (performance graph) shows error development of training set, validation set and testing set. A practical goal is to determine which subset of features should be used to generate best predictive model. In ANN modeling, if we compare feature subsets using in-sample error rates, the best performance will happen when all the features have been used. In this second trial, best performance of 0.044995 is achieved at 3000 epochs.

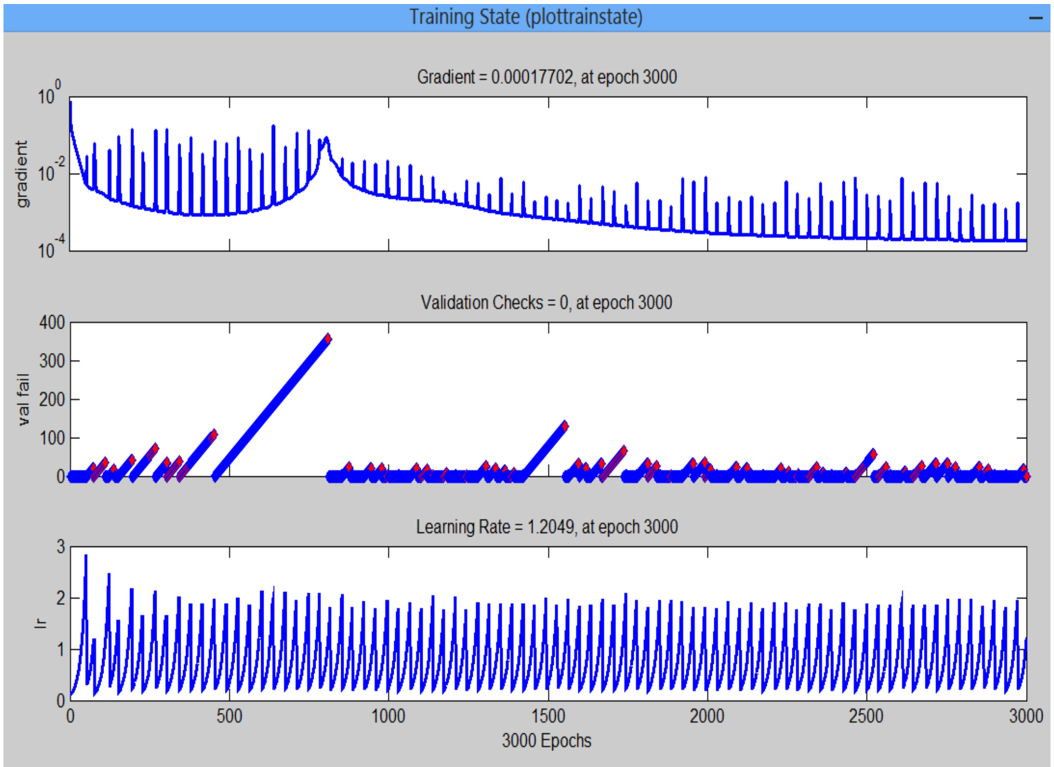


Figure 28: Graph of Training States (TRIAL-2)

In figure-28 the graph is representing training states as result of trial-2 for training ANN. It consists of three sub graphs. First one is plotted between 'gradient of error' and 'number of epochs'. Gradient of error is used for finding desired response of each node in upgrading weight. It is a ratio of difference of error between new and the older one, ∂E to the vector $\partial\theta$ comprises of all the network weights of each neuron i.e. $\frac{\partial E}{\partial \theta}$. If new error overshoots the older one more than a decided ratio, the new weights and biases will be rejected otherwise new weights will remain same. At 3000 iteration achieved gradient is 0.00017702. In first sub graph it is observed that with increasing number of iterations MSE keeps on decreasing and became very less at 3000 epochs. Second sub graph is plotted between validation fails and number of iterations. Number of successive iterations at which validation performance/error fail to decrease is known as validation fails. Between 500 and 1000 iterations validation check got maximum value that is more than 300 but after that it continue to decrease and at 3000 epochs its value becomes zero when training stopped. Third sub graph is plotted between epochs and learning rate of neural network. Learning rate is a training parameter that controls the size of weights and biases during training. If new error becomes less than the older one, learning rate increases and decreases if new error becomes greater than the older one. Zigzag track shows learning rate was kept on increasing and decreasing during entire training according to the prescribed training parameters and at 3000 epochs when training stop achieved learning rate is 1.2049

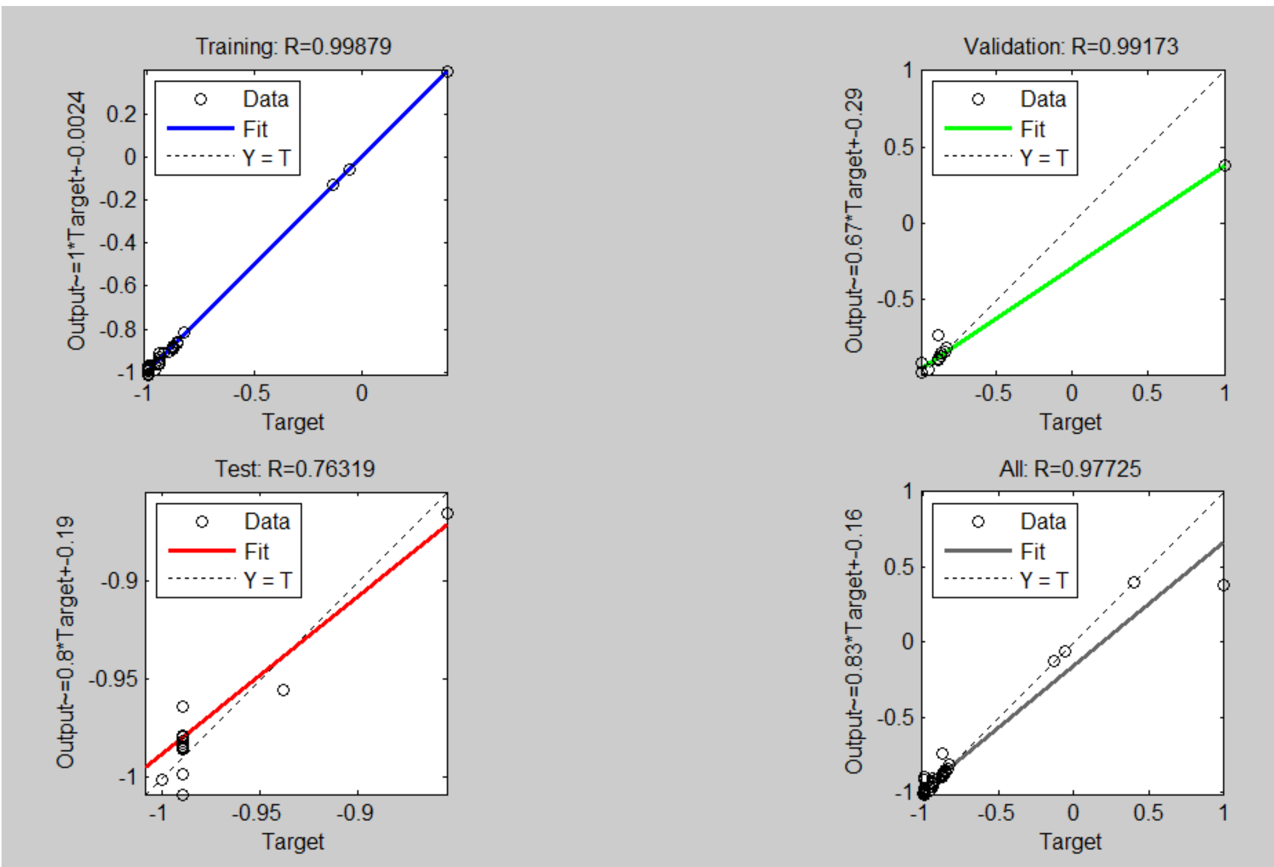


Figure 29: Regression graph (TRIAL-2)

Figure-29 is a regression graph of three states and overall regression before post processing when data range is between 1 and -1. Regression analysis shows a relationship between actual outputs and predictions. It is a statistical method for approximating relationship between variables. In this graph regression analysis is plotted between actual outputs (targets on ax-axis) and neural network predictions (outputs on y-axis).

Chapter 5

Experimental Results & Discussion

In this chapter, final results of experimental work done throughout the research is analyzed and explained. This research work presents one of the application of artificial neural network in identification and prediction of AF ratio of 2NZ-FE fuel injection engine. In this regards, a non-invasive method is adopted for controlling AF ratio. 100 active tests are performed for taking real time data of 2NZ-FE fuel injection engine with respect to AFR with the help of IT tester II and Intelligent Viewer software. On the basis of 100 real time collected data sets during active tests artificial neural network model is designed and simulated on MATLAB. Three layers of artificial neural network, 3000 training iterations, 5 neurons in hidden layer, *Gradient Descent Back propagation with adaptive learning rate* as training algorithms, hyperbolic sigmoid tangent as activation function are selected to train ANN for the control solution of air to fuel ratio of 2NZ-FE fuel injection engine.

5.1. Experimental Work

Present study is on the control solution of air to fuel ratio of automobile engines. For this study Toyota Corolla GLI with displacement of 1298CC 2NZFE EFI engine is selected. Gasoline is used as a fuel in this engine. Intelligent tester-II and intelligent viewer software of TOYOTA is used for measuring engine related data. Several active tests are performed for taking real time data of 2NZ-FE fuel injection engine with respect to AFR. Real time data is recorded while engine of vehicle is in operation. By varying engine rpm (i.e. engine speed) several engine parameters are recorded including calculate load in '%', vehicle load in '%', MAF (mass air flow) in 'gm/s', coolant temperature in '°C', intake air temperature in '°C', engine run-time in 's', opening of injector port in 'ms', injection volume in 'ml', IGN advance in 'degrees' and target air to fuel ratio. 100 experimental tests are performed by varying rpm 100 times. All collected real time data is summarized in Table-4 .The rpm of engine is varied from 0-4339 rpm. Effects of changing rpm on the other recorded engine parameters are studied and used in ANN training. The simulation is done on MATLAB.

5.2. ANN Training for the Control Solution of AFR

For the control solution of AFR by ANN training, collected data is simulated by using MATLAB. Among several parameters, only those are selected for ANN training which are observed as dependent on AFR during active tests. 10 dependent parameters are selected to record for input of ANN along with AFR as an output. Out of 100 recorded data sets, 75 data sets are used for training whereas rests of 25 data sets are used to test trained artificial neural network with actual engine output. Three layers are selected for modeling of ANN with input layer, hidden layer and output layer. 3000 epochs (i.e. number of iterations) are selected by analyzing training results of ANN from different iterations (500, 1000, 3000, and 5000) shown in table-6, table-7, table-8 and table-9 respectively. Selection is made on the basis of calculated Mean Square Error (MSE) corresponding to its least value. Number of neurons in hidden layer are varied from 5-30. After analyzing results in table-9 on the basis of least MSE 5 neurons are selected to place in hidden layer. *Gradient descent back propagation with momentum and with adaptive learning rate* is used as a training algorithm for ANN. This training algorithm is selected because in this algorithm step size increases with two successive weight changes are in the same direction and decreases when change of weights are in reverse direction. As a result of this larger learning rate can be achieved without instability. Pre and post processing is also done to get more refine results of ANN training. Transfer function "hyperbolic tangent sigmoid" is used between network layers. Momentum technique is used to avoid network to get trapped in shallow minima. Values of momentum constant and learning rate are taken as 1E-004 and 0.09 respectively on the basis of simulation results. With 3000 epochs (iterations) and 5 neurons in hidden layer, three trials are carried out. Least MSE of 1.594E-006 in 50 seconds of training is achieved in trial-2.

Following tests are carried out for analyzing designed ANN training.

1. Comparison test' of 25 data sets between 'actual outputs' and neural 'predictions' for getting best predicted AFR with minimum error.
2. 'Performance test' between 'Epochs' and 'MSE'
3. Analysis of 'Training states'
    a. Epochs vs. gradient
    b. Epochs vs. learning rate
    c. Epochs vs. validation check
4. Regression analysis between targets and predictions.

Table 12: Comparison table between 'Actual Outputs' and 'Predictions' by using 'Gradient Descent Back-propagation with momentum and Adaptive Learning' when MSE is 1.594e-006 at 3000 epochs

Actual Output	1.0000	1.0000	1.0030	1.000	1.000
Predicted Output	**0.9991**	**0.9991**	**0.9996**	**0.9983**	**0.9995**
Actual Output	1.000	1.000	1.000	1.000	1.000
Predicted Output	**1.0003**	**0.9987**	**0.9991**	**0.9992**	**0.9991**
Actual Output	1.000	1.000	1.000	1.000	1.0000
Predicted Output	**0.9992**	**0.9988**	**0.9999**	**0.9999**	**0.9999**
Actual Output	1.0000	1.0000	1.0000	1.0000	0.9990
Predicted Output	**0.9999**	**0.9999**	**0.9998**	**0.9998**	**0.9998**
Actual Output	1.0000	1.0000	1.0000	1.1860	0.9990
Predicted Output	**0.9987**	**0.9995**	**0.9995**	**1.1853**	**0.9991**

The simulated results of 25 test sets achieved after training artificial neural network model with '*Gradient Descent Back-propagation with Adaptive Learning'* are table-13**.** Above comparison table is showing very close results of neural network predicted outputs with respect to the actual outputs of a system being tested.

The plotted graph of above mentioned table-13 is shown in figure-30 below between AFR on y-axis and 25 testing data points on x-axis. The '*' are representing 'predicted values' whereas 'o' are used to indicate actual system's outputs' in figure-30 below.

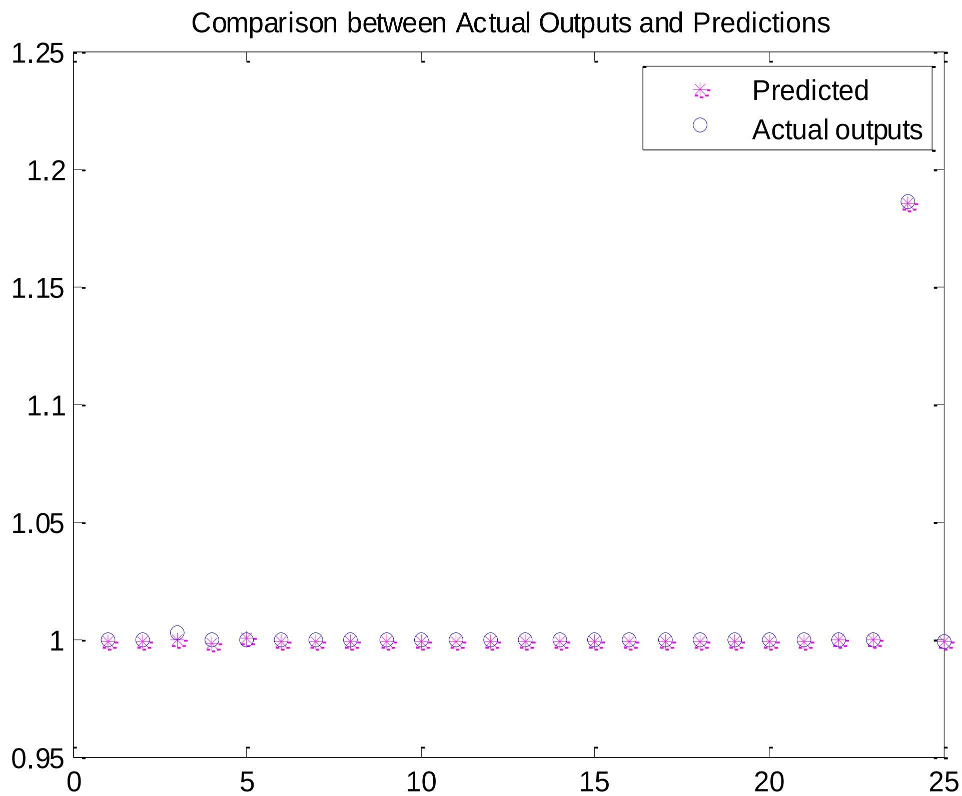


Figure 30: Comparison graph between predicted and actual outputs at which ANN is trained for predicting AFR of 2NZFE fuel injection engine with MSE of 1.4667e-006

All the required parameters including number of epochs, hidden layer, number of neurons, MSE, required training time and correlation coefficient for best training of ANN model are mentioned in table-14 below.

Table 13: Neural Network Training Parameters and Progress achieved by using 'Gradient Descent Back Propagation with Adaptive Learning Rate

PARAMETERS	
Epochs	3000
Hidden layer	1
Neurons	5
PROGRESS ACHIEVED	
Mean Square Error(MSE)	1.594e-006
Time	0:00:50
Correlation coefficient	0.9998

Figure-31 shows a proposed ANN model that can be added in the control solution of AFR. This will help in minimizing some of engine errors that creates hindrance in achieving AFR practically like error due to wall wetting.

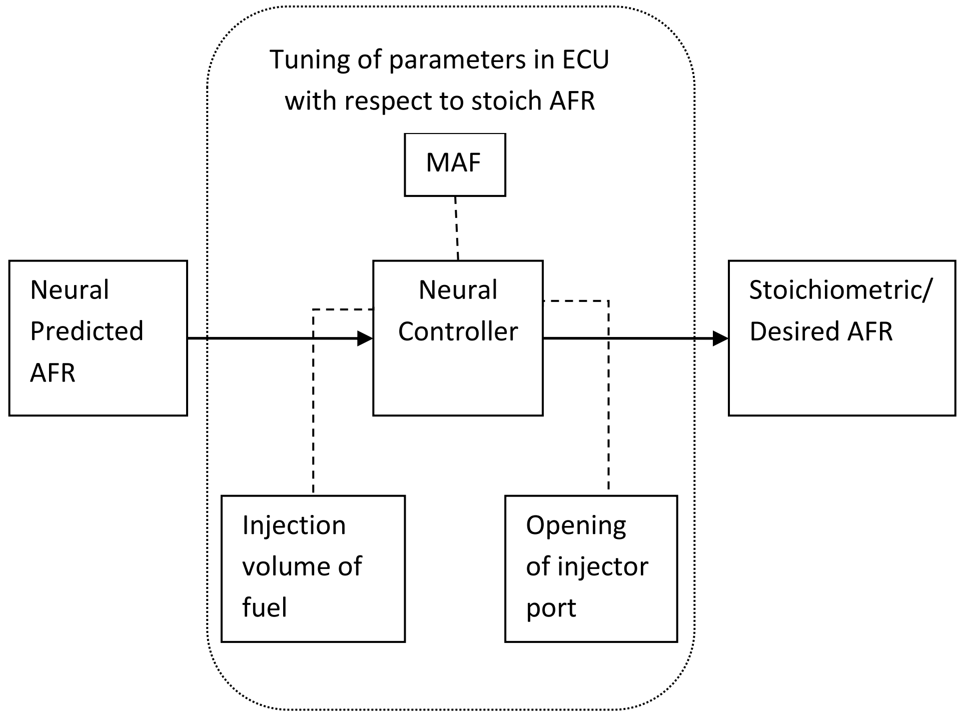


Figure 31: Proposed ANN Model

Chapter 6

Conclusion & Future Work

In this chapter, conclusion is drawn on the basis of all experimental work and their results throughout the research. It contains the conclusions which have made from the results and some recommendations for the future work. It is an ongoing project and some future perspectives are proposed for future study.

6.1. Conclusion

Training is done with 75 data sets where as other 25 data sets are used to test validity and effectiveness of neural model. Pseudorandom changes the weights with the objective of reducing the error paved a way to make an effective model for the prediction of AFR in EFI engines by using 'gradient descent back propagation with adaptive learning' approach. By using second derivative information, network convergence speed is prompted and the generalized performance is enhanced. Taking 2NZ-FE injection engine, nonlinear mapping is set up from ten influence factors (engine speed, calculate load, vehicle load, MAF, coolant temperature, air intake temperature, injection period, injection volume and IGN advance) to AFR. The simulation results show that designed network model has preferable learning and generalization capabilities, which performs effectively in predicting AF ratio of automobile fuel injection 2NZ-FE engine with minimal MSE of 1.594e-006 where correlation coefficient is equal to 0.9998 between predicted and actual (observed) values of air-fuel ratio at 3000 epochs. ANN is trained within 50 seconds. Correlation coefficient value is very close to unity and thus we can conclude that the experimental AFR is very close to stoic AFR. From this we further conclude that by using this technique engine efficiency may be enhanced tremendously. As MSE value is very low so we conclude that the control of AFR is very precise. ANN is trained within 50 seconds, which shows that best AFR value is achieved in very short duration. This is also strength of this technique.

It is pertinent to mention that while analyzing the engine parameters i.e. opening of injector port for fuel injection, injection volume of fuel into the cylinder and mass air flow are highly dependent on the AFR. From the predicted stoichiometric AFR, value of these parameters can be identified for their tuning as these values are already saved in the memory of ANN. These information if send to electronic control unit (ECU) of 2NZFE fuel injection engine can tune engine related parameters with respect to stoichiometric AFR.

By considering above aspects, a conceptual model of another controller is proposed that links with these parameters which can tune parameters i.e. opening of injector port for fuel injection, injection volume of fuel into the cylinder and mass air flow according to the desired stoichiometric air fuel ratio.

From the results summarized in above study, it is concluded that, AFR can be precisely controlled by using ANN training. .Stoic AFR is closely achieved by using artificial intelligence technique. In turn we may claim that, the most efficient engine can be designed by using this advance technique. Environmental pollution is also minimized as AFR achieved is very close to stoichiometric AFR.

6.2. Future Work

Some perspectives are proposed for future study. There are several investigations remaining to be evolved, some areas are identified that required further analysis and additional computational work. These theoretical results can be implemented practically for further investigations.

1. Artificial neural network is designed for 2NZ-FE fuel injection engine in order to regulate AF ratio in this research. By using same strategy we can also train another ANN for other automobile fuel injection engines as well.
2. In present work control solution of air to fuel ratio for automobile engine is proposed for gasoline engine which can be extended for the diesel engines.
3. Offline training is performed in this research work due to lack of resources but in the future, people can perform its online training which improves its function approximation [50].
4. Training and testing is done by using 10 input parameters; we can increase number of AFR dependent input parameters in the future to enhance its performance.

5. Not only AF ratio but other engine parameters can also be identified and predicted on the basis of selected data and their dependency.
6. Online simulation work may practice on 2NZ-FE fuel injection engine to see its actual effects while running.

REFERENCES

[1] Glielmo, L., et al. *Modelling and numerical simulation of the dynamic behaviour of three way catalytic converters*. in *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*. 1998: IEEE.

[2] Alippi, C., C. de Russis, and V. Piuri, *A neural-network based control solution to air-fuel ratio control for automotive fuel-injection systems.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2003. **33**(2): p. 259-268.

[3] Shiao, Y. and J.J. Moskwa. *Model-based cylinder-by-cylinder air-fuel ratio control for SI engines using sliding observers*. in *Control Applications, 1996., Proceedings of the 1996 IEEE International Conference on*. 1996: IEEE.

[4] Davis Jr, L.I., et al. *Neural network modeling and control of an anti-lock brake system*. in *Intelligent Vehicles' 92 Symposium., Proceedings of the*. 1992: IEEE.

[5] Puskorius, G.V. and L.A. Feldkamp. *Automotive engine idle speed control with recurrent neural networks*. in *American Control Conference, 1993*. 1993: IEEE.

[6] Levin, A.U. and K.S. Narendra, *Control of nonlinear dynamical systems using neural networks: Controllability and stabilization.* Neural Networks, IEEE Transactions on, 1993. **4**(2): p. 192-206.

[7] Alippi, C., *FPE-based criteria to dimension feedforward neural topologies.* Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 1999. **46**(8): p. 962-973.

[8] Alfieri, E., et al. *Model-based feedback control of the air-to-fuel ratio in diesel engines based on an empirical model*. in *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*. 2006: IEEE.

[9] Alippi, C. and V. Piuri, *Experimental neural networks for prediction and identification.* Instrumentation and Measurement, IEEE Transactions on, 1996. **45**(2): p. 670-676.

[10] Cesario, N., M. Lavorgna, and F. Pirozzi. *Modelling on-off virtual lambda sensors based on multi-spread probabilistic neural networks*. in *Emerging Technologies and Factory Automation, 2005. ETFA 2005. 10th IEEE Conference on*. 2005: IEEE.

[11] Majors, M., J. Stori, and D. Cho. *An adaptive neural network control method for automotive fuel-injection systems*. in *Intelligent Control, 1993., Proceedings of the 1993 IEEE International Symposium on*. 1993: IEEE.

[12] Hu, J., G. Li, and J. Xu. *Component-based ECU design method of passenger car information integrated control system*. in *Automation and Logistics, 2009. ICAL'09. IEEE International Conference on*. 2009: IEEE.

[13] Li, Z., et al. *System of real-time monitoring dynamic vehicle load status*. in *Vehicular Electronics and Safety (ICVES), 2013 IEEE International Conference on*: IEEE.

[14] Buehler, P.J., M.A. Franchek, and I. Makki. *Mass air flow sensor diagnostics for adaptive fueling control of internal combustion engines*. in *ASME 2002 International Mechanical Engineering Congress and Exposition*. 2002: American Society of Mechanical Engineers.

[15] Hwang, S. *Simulation of advance ignition system for a four stroke and single cylinder engine using photo sensor and 8751 H microcontroller*. in *Control Applications, 1992., First IEEE Conference on*. 1992: IEEE.

[16] Burgess, C.J. and M. Lefley, *Can genetic programming improve software effort estimation? A comparative evaluation.* Information and Software Technology, 2001. **43**(14): p. 863-873.

[17] Yang-cui, N., et al. *Forecasting the natural forest stand age based on artificial neural network model*. in *Computer and Communication Technologies in Agriculture Engineering (CCTAE), 2010 International Conference On*: IEEE.

[18] Guolian, H., et al. *Research on fault diagnosis of wind turbine control system based on Artificial Neural Network*. in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*: IEEE.

[19] Narendra, K.S. and K. Parthasarathy, *Identification and control of dynamical systems using neural networks.* Neural Networks, IEEE Transactions on, 1990. **1**(1): p. 4-27.

[20] Rosenblatt, F., *Perceptron simulation experiments.* Proceedings of the IRE, 1960. **48**(3): p. 301-309.

[21] Bershad, N.J., C.F.N. Cowan, and J.J. Shynk. *Stochastic convergence analysis of a two-layer perceptron for a system identification model*. in *Signals, Systems and Computers, 1991. 1991 Conference Record of the Twenty-Fifth Asilomar Conference on*. 1991: IEEE.

[22] Gopalakrishnan, A., et al. *Constructive proof of efficient pattern storage in the multi-layer perceptron*. in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. 1993: IEEE.

[23] Diggavi, S.N., J.J. Shynk, and N.J. Bershad, *Convergence models for Rosenblatt's perceptron learning algorithm.* Signal Processing, IEEE Transactions on, 1995. **43**(7): p. 1696-1702.

[24] Yin, Z., et al. *A MLP Network based Adaptive UWB Modulation Scheme Recognizer*. in *Signal Processing, 2006 8th International Conference on*: IEEE.

[25] Hunt, K.J., et al., *Neural networks for control systemsâ€"a survey.* Automatica, 1992. **28**(6): p. 1083-1112.

[26] Elias, J.G. and W.T. Rogers. *An electronic artificial neuron with biologically realistic behavior*. in *Neural Networks, 1991. 1991 IEEE International Joint Conference on*. 1991: IEEE.

[27] Shiraishi, H., S.L. Ipri, and D.I. Cho, *CMAC neural network controller for fuel-injection systems.* Control Systems Technology, IEEE Transactions on, 1995. **3**(1): p. 32-38.

[28] Chang, C.-F., et al., *Air-fuel ratio control in spark-ignition engines using estimation theory.* Control Systems Technology, IEEE Transactions on, 1995. **3**(1): p. 22-31.

[29] Won, M., S.B. Choi, and J.K. Hedrick, *Air-to-fuel ratio control of spark ignition engines using Gaussian network sliding control.* Control Systems Technology, IEEE Transactions on, 1998. **6**(5): p. 678-687.

[30] Manzie, C., M. Palaniswami, and H. Watson, *Gaussian networks for fuel injection control.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2001. **215**(10): p. 1053-1068.

[31] Lee, T., et al., *Estimation of Air Fuel Ratio of a SI Engine from Exhaust Gas Temperature at Cold Start Condition.* SAE Paper, 2002(2002-01): p. 1667.

[32] Wagner, J.R., D.M. Dawson, and L. Zeyu, *Nonlinear air-to-fuel ratio and engine speed control for hybrid vehicles.* Vehicular Technology, IEEE Transactions on, 2003. **52**(1): p. 184-195.

[33] Hou, Z., Q. Sen, and Y. Wu. *Air Fuel Ratio Identification of Gasoline Engine during Transient Conditions Based on Elman Neural Networks*. in *Intelligent Systems Design and Applications, 2006. ISDA'06. Sixth International Conference on*. 2006: IEEE.

[34] Gnanam, G., et al., *Neural network control of air-to-fuel ratio in a bi-fuel engine.* Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2006. **36**(5): p. 656-667.

[35] Hou, Z. and Y. Wu. *The Research on Air Fuel Ratio Predictive Model of Gasoline Engine during Transient Condition*. in *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. 2007: IEEE.

[36] Zhai, Y.J. and D.L. Yu, *Radial-basis-function-based feedforwardâ€"feedback control for airâ€"fuel ratio of spark ignition engines.* Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 2008. **222**(3): p. 415-428.

[37] Ju-Biao, Y., W. Bin, and Z. Da-Sen. *Simulation on Air Fuel Ratio Control Based on Neural Network*. in *Power and Energy Engineering Conference, 2009. APPEEC 2009. Asia-Pacific*. 2009: IEEE.

[38] Liu, Z.-q. and Y.-c. Zhou. *A fuzzy neural network and application to air-fuel ratio control under Gasoline Engine Transient Condition*. in *Intelligent System Design and Engineering Application (ISDEA), 2010 International Conference on*: IEEE.

[39] Wang, T., et al. *Research in fuel injection revision control of gasoline engine based on the electronic throttle during transient condition*. in *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on*: IEEE.

[40] Saraswati, S. and S. Chand. *An optimization algorithm for neural predictive control of air-fuel ratio in SI engines*. in *Modelling, Identification and Control (ICMIC), The 2010 International Conference on*: IEEE.

[41] Barghi, F. and A.A. Safavi. *Experimental validation of recurrent Neuro-Fuzzy Networks for AFR estimation and control in SI engines*. in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2011 IEEE International Conference on*: IEEE.

[42] Chen, X., et al. *Air-to-fuel ratio control with adaptive estimation of biofuel content for diesel engine LNT regeneration*. in *American Control Conference (ACC), 2012*: IEEE.

[43] Jansri, A. and P. Sooraksa. *Simple Nonlinear Air to Fuel Ratio Control for Spark Ignition Engines*. in *Computer, Consumer and Control (IS3C), 2012 International Symposium on*: IEEE.

[44] Yar, A. and A.I. Bhatti. *Control of Air-to-Fuel ratio of spark ignited engine using super twisting algorithm*. in *Emerging Technologies (ICET), 2012 International Conference on*: IEEE.

[45] Meyer, J., S. Yurkovich, and S. Midlam-Mohler, *Air-to-Fuel Ratio Switching Frequency Control for Gasoline Engines.*

[46] Ghaffari, A., et al., *Performance comparison of neural network training algorithms in modeling of bimodal drug delivery.* International journal of pharmaceutics, 2006. **327**(1): p. 126-138.

[47] Qian, N., *On the momentum term in gradient descent learning algorithms.* Neural Networks, 1999. **12**(1): p. 145-151.

[48] Plagianakos, V.P., D.G. Sotiropoulos, and M.N. Vrahatis. *An improved backpropagation method with adaptive learning rate*. in *Proceedings of the 2nd international conference on circuits, systems and computers*. 1998: Citeseer.

[49] Diotalevi, F., M. Valle, and D.D. Caviglia. *Evaluation of gradient descent learning algorithms with an adaptive local rate technique for hierarchical feedforward architectures*. in *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*. 2000: IEEE.

[50] Bhattacharya, U. and S.K. Parui. *Self-adaptive learning rates in backpropagation algorithm improve its function approximation performance*. in *Neural Networks, 1995. Proceedings., IEEE International Conference on*. 1995: IEEE.