A PERFORMANCE COMPARISON OF DIFFERENT BACK PROPAGATION NEURAL NETWORKS FOR NITROGEN OXIDES EMISSION PREDICTION IN THERMAL POWER PLANT

PREETI MANKE
Computer Science & Engineering Department, Institute of Technology, Korba, India

ABSTRACT

The use of Neural Networks (NN) has been proved to be a cost-effective technique. It is very important to choose a suitable back propagation (BP) algorithm for training a neural network. While these algorithms prove to be very effective and robust in training many types of non-linear multivariable modeling, they suffer from certain disadvantages such as easy entrapment and very slow convergence. The power generating industry is undergoing an unprecedented reform. NN is applied to predict coal properties, economic load dispatch, emission prediction, temperature control, etc. This paper compares the performance of the six neural network methods to predict nitrogen oxides emission from a 500 MW coal fired thermal power plant. The various BP training algorithms used are gradient descent, gradient descent with momentum, variable learning rate with momentum, conjugate gradient back propagation, Quasi-Newton BFGS Algorithm and also Levenberg-Marquardt. The parametric field experiment data were used to build neural network. The coal combustion parameters were used as inputs and nitrogen oxides as output of the model. The predicted values of the model for full load condition were verified with the actual values.

Keywords: Artificial Neural Network, Back Propagation, Thermal Power Plant

1 INTRODUCTION

In recent times, neural network (NN) techniques have been used widely in solving problems like optimization, adaptive filtering, digital signal processing, and pattern recognition and classification. Indeed the discipline of neural networks was introduced in the 1950s and now has gained momentum due to the many successes.
Most of the time thermal power plants produce output more than the plant’s capacity in order to meet demands. Being a major fossil fuel in India coal meets about 61% of the commercial energy needs and about 72% of the electricity produced [1]. The coal combustion process produces various pollutants, such as oxides of carbon (CO\textsubscript{x}), oxides of sulphur (SO\textsubscript{x}), oxides of nitrogen (NO\textsubscript{x}) and particulates. The acid rain and climate change are mainly due to pollutants like SO\textsubscript{2}, NO\textsubscript{x} and CO\textsubscript{2}. Climate change induced by these gases can cause damage to human health, agriculture, natural ecosystems, coastal areas and other climate sensitive systems. Hence sustainable development, corporate social responsibilities are introduced for better accountability of environment and green power production. These industries must continuously monitor their exhaust stacks for primary pollutants under regulations promulgated under the New Source Performance Standards (U.S. Environmental Protection Agency (EPA)) or the Clean Air Act Amendments [3].

Like other pollutants no practical methods exist for reducing NO\textsubscript{x} significantly, leading to increased research into this area. During the combustion process in a coal-fired power plant, nitrogen from the coal and air is converted into nitric oxide (NO) and nitrogen dioxide (NO\textsubscript{2}); together these oxides of nitrogen are commonly referred to as NO\textsubscript{x}. The methods for reducing NO\textsubscript{x} emissions in coal-fired power plants can be classified as either primary or combustion modification based technologies [4, 6], which achieve reduction of NO\textsubscript{x} formation by limiting the flame temperature through control of oxygen in the flame. Secondary technologies are flue gas treatment.

Several works have been done to develop predictive systems for industrial emissions. One of the earlier ideas was presented by S.S.S Chakravarty, A.K Vohra and B.S Gill [8] has developed a predictive emission monitors for industrial process heaters. The authors have used heuristic optimizer genetic algorithm (GA) to tune the NO\textsubscript{x} kinetic parameters. L. Zheng, S. Yu, M. Yu [9] used generalized regression neural network (GRNN) to establish a non-linear model between the parameters of the boiler of 300MW steam capacity and the NO\textsubscript{x} emissions. Researchers studied the non-linear problem for decades and many traditional and meta-heuristic techniques including artificial intelligence methods have been developed [11]. A machine-learning method for non-statistical model building, such as artificial neural networks, can be improved to attain the desired accuracy level by training it on experimental data [12]. In this work, the parametric field experiments to obtain the relationship between the operating parameters and NO\textsubscript{x} emission concentration in flue gas are introduced. The ability of NN to model the NO\textsubscript{x} pulverized coal combustion characteristics of a 500 MW thermal power plant under full load condition is demonstrated.

However, these models tend to be really comprehensive to build and they are time consuming and also the appropriate parameters could not be determined immediately for changed operating conditions [10]. Recently back propagation approaches [5, 7] have been used to develop advanced on-line and real-time combustion optimization software package in modern power plants. This article emphasizes the effectiveness of various BP neural networks to determine the optimal combustion parameters for minimum NO\textsubscript{x} emission. Neural network modelling described in this study is implemented in Matlab 6.5.0 (MathWorks, Inc.) and run under the Microsoft Windows 7 environment.

This paper is organized as follows: Section 2 presents a brief literature on artificial neural network and various Backpropagation neural networks (BPNN). Section 3 describes the modeling background in terms of the parameters used, case study of coal fired 500MW power plant and the dataset used. Section 4 presents the results and analysis obtained from BP training algorithms used are gradient descent, gradient descent with momentum, variable learning rate with momentum, conjugate gradient back propagation, Quasi-Newton BFGS Algorithm and also Levenberg-Marquardt for full load condition. Section 5 gives a brief conclusion with future scope reached in this study.
2 BACK PROPAGATION NEURAL NETWORK (BPNN)

The most common class of neural network is the multilayered feedforward network which primarily consists of input layer, one or more hidden layer and an output layer [6]. Through the input layer, the normalized raw data is fed into the network. The input layer holds the data and distributes them into the network via interconnections to neurons in the hidden layer(s) where they are processed by the activation function to obtain the output signal. Likewise, the behaviour of the output units depends on the activity of the hidden units and the weights between the hidden and output units. The inputs of each neuron in the hidden and the output layers are summed and the result is processed by an input–output function (transfer function).

Using the neural networks as its fitness function, the optimization algorithm determines the optimum levels of coal combustion parameters for minimum NO\textsubscript{x} emission. NN in combination with optimization algorithm, has been used successfully in various studies to solve a variety of optimization problems, including problems where the objective function is discontinuous, non-differentiable, stochastic, or highly non-linear [16]. Here, the same combination is applied for NO\textsubscript{x} emission optimization. The back propagation neural network however has been widely used to develop soft sensors for prediction of NO\textsubscript{x} [9]. The main advantage of BPNN is the ability to model a problem by the use of data associated with process, rather than analysis of process by some standard numerical methods.

![Fig. 1 Block diagram for Boiler with BPNN](image)

In this model the system’s output error was propagated back through the plant using its partial derivatives at an operating point as shown in Figure 1.

Although the BP algorithm is commonly used in recent years to perform the training task, some drawbacks are often encountered using this gradient-based method, include: very slow training convergence speed and getting stuck in a local minimum easily. In order to solve these drawbacks different algorithms have been proposed [5, 7]. This disadvantage can be removed by an exploration ability of the different evolutionary back propagation algorithms. This paper represents the performance comparison of the feed forward neural network using various BP algorithms. The BP training algorithms used are gradient descent, gradient descent with momentum, variable learning rate with momentum, conjugate gradient back propagation, Quasi-Newton BFGS Algorithm and also Levenberg-Marquardt.

2.1 Gradient Descent BP (GD)

The simplest implementation of back propagation learning updates the network weights and biases in the direction in which the performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written as:

\[ X_{k+1} = X_k - \alpha_k g_k \]
Where \( X_k \) is a vector of current weights and biases, \( \alpha_k \) is the learning rate and \( g_k \) is the current gradient of the error with respect to the weight vector. The negative sign indicates that the new weight vector \( X_{k+1} \) is moving in a direction opposite to that of the gradient.

### 2.2 Gradient Descent BP with Momentum (GDM)

Momentum allows a network to respond not only to the local gradient, but also to recent trends in the error surface. Acting like a lowpass filter, momentum allows the network to ignore small features in the error surface. Without momentum a network can get stuck in a shallow local minimum [14]. With momentum a network can slide through such a minimum. You can add momentum to back propagation learning by making weight changes equal to the sum of a fraction of the last weight change and the new change suggested by the back propagation rule. The magnitude of the effect that the last weight change is allowed to have is mediated by a momentum constant, \( \mu \), which can be any number between 0 and 1. The gradient is computed by summing the gradients calculated at each training example, and the weights and biases are only updated after all training examples have been presented. The new weight vector \( X_{k+1} \) is adjusted as:

\[
X_{k+1} = X_k - \alpha_k g_k + \mu X_{k-1}
\]

### 2.3 Variable Learning Rate BP with Momentum (VLRM)

The performance of the algorithm is very sensitive to the proper setting of the learning rate. If the learning rate is set too high, the algorithm can oscillate and become unstable. If the learning rate is too small, the algorithm takes too long to converge. It is not practical to determine the optimal setting for the learning rate before training, and, in fact, the optimal learning rate changes during the training process, as the algorithm moves across the performance surface. You can improve the performance of the steepest descent algorithm if you allow the learning rate to change during the training process. An adaptive learning rate attempts to keep the learning step size as large as possible while keeping learning stable. Typically, the new weight vector \( X_{k+1} \) is defined as:

\[
X_{k+1} = X_k - \alpha_{k+1} g_k + \mu X_{k-1}
\]

\[
\alpha_{k+1} = \beta \alpha_k
\]

To speed up the convergence time, the variable learning rate gradient descent BP utilizes larger learning rate \( \alpha_k \) when the neural network model is far from the solution and smaller learning rate \( \alpha_k \) when the neural net is near the solution.

### 2.4 Conjugate Gradient BP (CGP)

It turns out that, although the function decreases most rapidly along the negative of the gradient, this does not necessarily produce the fastest convergence. In the conjugate gradient algorithms a search is performed along conjugate directions, which produces generally faster convergence than steepest descent directions. In most of the training algorithms discussed up to this point, a learning rate is used to determine the length of the weight update (step size). A search is made along the conjugate gradient direction to determine the step size that minimizes the performance function along that line [2]. Some search functions are best suited to certain training functions, although the optimum choice can vary according to the specific application. The search direction at each iteration is determined by updating the weight vector as:

\[
X_{k+1} = X_k + \alpha_k p_k
\]

Where:

\[
p_k = - g_k + \beta_k p_{k-1}
\]
2.5 Quasi-Newton BFGS Algorithm (QN)

Newton’s method is an alternative to the conjugate gradient methods for fast optimization. The basic step of Newton’s method is where is the Hessian matrix (second derivatives) of the performance index at the current values of the weights and biases. Quasi-Newton’s method often converges faster than conjugate gradient method. Unfortunately, it is complex and expensive to compute the Hessian matrix for feedforward neural networks.

\[
\beta_k = \frac{\Delta g_k^T}{g_k^T - 1} g_k - 1
\]

and

\[
\Delta g_k^T = g_k^T - g_k - 1
\]

2.6 Levenberg-Marquardt Algorithm (LM)

Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. When the performance function has the form of a sum of squares, then the Hessian matrix can be approximated as and the gradient can be computed as where J is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. The Jacobian matrix can be computed through a standard back propagation technique [15] that is much less complex than computing the Hessian matrix. When the scalar \( \mu \) is zero, this is just Newton’s method, using the approximate Hessian matrix. When \( \mu \) is large, this becomes gradient descent with a small step size.

\[
\tilde{x}_{k+1} = x_k - A_k^{-1} g_k
\]

There is a class of algorithms that is based on Newton’s method, but which doesn’t require calculation of second derivatives. Fortunately, there is a class of algorithms based on the works of Broyden, Fletcher, Goldfarb, and Shanno (BFGS).

3 CASE STUDY

The experiments are carried out in a 500 MW tangentially coal-fired dry bottom boiler with a large furnace. The tilting fuel and combustion air nozzles including nine primary air burners and ten secondary air burners are located in each corner of the furnace. All nozzles can be tilted in vertical direction over about 30 degrees from the horizontal axis, both upwards and downwards. The burners on A, B, C, D, E, F, G, H, J levels were put into operation under the rated load. The coal pulverisers are employed to supply the coal–air mixture to the burners on the corresponding levels. The tangential firing system is employed to combust bituminous coal. Tangential firing helps in keeping the temperature of the furnace low so that NO\(_X\) emission is reduced considerably. In addition to the above the over fire air is provided which is used as combustion process adjustment technically for keeping the furnace temperature low and thereby low NO\(_X\) formation. The typical variation in air flow rate and economizer flue gas outlet temperature for 500MW boiler is shown in Figure 2 and Figure 3 respectively.
In total, 30 tests have been performed on this boiler, changing the boiler load, primary air, secondary air distribution pattern, nozzles tilting angle, respectively, to analyze the characteristics of the NO$_x$ emission of the tangentially fired system. Out of which, 10 test data pertaining to full load condition, are used for this present study.

### 3.1 The NO$_x$ Emission Prediction

During all the experiments, the fineness of the coal is kept constant. NO$_x$ and O$_2$ concentrations are monitored continuously in the boiler outlet prior to the air heater. Fly ash samples are withdrawn from the flue gas by a constant rate sampling probe. The NO$_x$ concentrations reported in this work are average values over several hours of stable operation, and they are obtained under dry gas conditions. Selected 10 sets of test data under full condition is provided in Table 1. The measured NO$_x$ emissions for full load condition are summarized and compared with above mentioned six BP algorithms result in Table 2. The learning of the network is carried out through adjusting the weights by continuous iterations and minimizing the error between experimentally measured response and neural network model-predicted response [16]. Mean square error (MSE), and correlation factor (R) are used to evaluate neural networks performance. When the MSE are at the minimum, and ‘R’ value closer value to 1 represents high performance and perfect accuracy.

**Table 1** The important boiler operating parameters

<table>
<thead>
<tr>
<th>Sr No</th>
<th>LOAD (MW)</th>
<th>FG Temp at ECO. O/L (Degree Cel.)</th>
<th>Total Air flow (T/Hr)</th>
<th>O2 (%)</th>
<th>Furnace Windbox DP (mmWCL)</th>
<th>Total Coal Flow (T/Hr)</th>
<th>Burner Tilt (Degree)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>515</td>
<td>377 379</td>
<td>1715</td>
<td>2.5</td>
<td>3.3</td>
<td>50 50</td>
<td>-28 -27 -28 -27</td>
</tr>
<tr>
<td>2</td>
<td>516</td>
<td>375 370</td>
<td>1720</td>
<td>1.7</td>
<td>2.8</td>
<td>40 50</td>
<td>-25 -27 -27 -28</td>
</tr>
<tr>
<td>3</td>
<td>502</td>
<td>371 368</td>
<td>1650</td>
<td>2.2</td>
<td>1.9</td>
<td>60 65</td>
<td>-27 -25 -27 -27</td>
</tr>
<tr>
<td>4</td>
<td>510</td>
<td>374 367</td>
<td>1700</td>
<td>2.3</td>
<td>2.8</td>
<td>45 50</td>
<td>-29 -27 -29 -27</td>
</tr>
<tr>
<td>5</td>
<td>506</td>
<td>388 392</td>
<td>1640</td>
<td>2.4</td>
<td>1.5</td>
<td>50 50</td>
<td>-26 -26 -27 -28</td>
</tr>
<tr>
<td>6</td>
<td>515</td>
<td>383 389</td>
<td>1700</td>
<td>1.8</td>
<td>2.0</td>
<td>40 45</td>
<td>-25 -27 -25 -27</td>
</tr>
<tr>
<td>7</td>
<td>509</td>
<td>395 377</td>
<td>1650</td>
<td>2.1</td>
<td>1.7</td>
<td>45 50</td>
<td>-25 -27 -25 -27</td>
</tr>
<tr>
<td>8</td>
<td>503</td>
<td>378 377</td>
<td>1680</td>
<td>3.4</td>
<td>2.1</td>
<td>55 60</td>
<td>-24 -24 -23 -23</td>
</tr>
<tr>
<td>9</td>
<td>507</td>
<td>383 376</td>
<td>1685</td>
<td>2.1</td>
<td>2.9</td>
<td>50 50</td>
<td>-27 -25 -27 -27</td>
</tr>
<tr>
<td>10</td>
<td>512</td>
<td>382 364</td>
<td>1695</td>
<td>2.7</td>
<td>2.4</td>
<td>45 45</td>
<td>-27 -24 -27 -27</td>
</tr>
</tbody>
</table>
The predicted values and measured values for the full load condition are given in Table 2. Because the back propagation neural networks may be considered simply as a nonlinear input–output mapping, such a mapping is so quick and it is suitable to be used as the objective function for the optimization algorithms. The optimum operating parameters can be found employing the searching ability of optimization algorithms.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>The NO\textsubscript{x} emission under above operating condition and predicted NO\textsubscript{x}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured NO\textsubscript{x} (ppm)</td>
<td>Sr. No.</td>
</tr>
<tr>
<td>GD</td>
<td></td>
</tr>
<tr>
<td>GDM</td>
<td></td>
</tr>
<tr>
<td>VLRM</td>
<td></td>
</tr>
<tr>
<td>CGP</td>
<td></td>
</tr>
<tr>
<td>QN</td>
<td></td>
</tr>
<tr>
<td>LM</td>
<td></td>
</tr>
<tr>
<td>Predicted NO\textsubscript{x} (ppm)</td>
<td>GD</td>
</tr>
<tr>
<td>GDM</td>
<td>382</td>
</tr>
<tr>
<td>VLRM</td>
<td>629</td>
</tr>
<tr>
<td>CGP</td>
<td>307</td>
</tr>
<tr>
<td>QN</td>
<td>355</td>
</tr>
<tr>
<td>LM</td>
<td>631</td>
</tr>
</tbody>
</table>

4 RESULTS

The main objectives for boiler combustion optimization are to help the operators to perform clean and efficient utilization of coal. Thus the NO\textsubscript{x} prediction and optimization objective function was derived from the weights and biases of the trained feed forward back propagation neural network. Weights and biases of all layers of neurons were combined with transfer functions of NN model to achieve an equation pattern as the following steps. The 9 input layer nodes with the 1st bias node connected to 10 nodes of hidden layer. Thus, there are 90 values of weights and 10 values of biases on the layers between input and hidden layer.

<table>
<thead>
<tr>
<th>Table 3</th>
<th>The performance of BPNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPNN</td>
<td>MSE</td>
</tr>
<tr>
<td>GD</td>
<td>2.31 exp(-5)</td>
</tr>
<tr>
<td>GDM</td>
<td>2.28 exp(-4)</td>
</tr>
<tr>
<td>VLRM</td>
<td>5.12 exp(-4)</td>
</tr>
<tr>
<td>CGP</td>
<td>1.42 exp(-5)</td>
</tr>
<tr>
<td>QN</td>
<td>1.62 exp(-6)</td>
</tr>
<tr>
<td>LM</td>
<td>1.74 exp(-5)</td>
</tr>
</tbody>
</table>

The NN is used to train the operating parameters by considering the experimental data from Table 1. It determines the weights between processing elements in the input and hidden layer and between the hidden layer and output layers which minimize the differences between the network output and the measured values. The experimental data stated above are used to find the relation between the operational parameters and the NO\textsubscript{x} emission concentration in flue gas under full load condition. The trained network achieved highest correlation factor R and MSE are summarized above. Average performance of above mentioned six BPNN is given in Table 2 and Table 3. The measured and predicted NO\textsubscript{x} emission concentration in flue gas shown in Figure 4 indicates that the trained network is performing reasonably well in prediction.
5 CONCLUSION

This paper showed that neural network techniques could be used to predict NO\textsubscript{x} emission from 500MW thermal power plant. It is very important to choose a proper algorithm for training a neural network, so a comparison of various BP algorithms with different learning ratios and activation functions is studied. Back propagation neural network training with gradient descent, gradient descent with momentum algorithms usually faced drawbacks such as very slow convergence and may trap in the local minimum. Variable learning rate with momentum is providing better convergence with lesser problem of trap. In contrast, weights and biases in Quasi-Newton BFGS Algorithm, Levenberg-Marquardt and conjugate gradient back propagation are fitted better to obtain fast convergence without trap in the local minimum. The performance of the BP methods depends on the neural networks topology. In future a hybrid model that involves both neural network and support vector machines may be developed in order to further improve the predictive accuracy and computational efficiencies.

6 REFERENCES


