Copyright © 2002 IFAC
15th Triennial World Congress, Barcelona, Spain

GA BASED NEURAL NETWORK MODELING OF NOX EMISSION IN A COAL-FIRED POWER GENERATION PLANT

Kang Li, Steve Thompson, Jian-xun Peng

School of Mechanical & Manufacturing Engineering
Queen's University Belfast
Ashby Building, Stranmillis Rd., Belfast BT9 5AH, UK

Abstract: Genetic algorithm-based neural network modeling is studied. A MLP model for predicting NOx emission in a coal-fired power generation plant is trained using genetic algorithms. In order to avoid over-training, two data sets are involved, i.e. one data set is used for searching the weights and bias, the other set is used for validation. The fitness function for the GA based training is the combination of the training error and validation error. The GA-based MLP model has been tested over different periods of operation, showing the merits of this modeling technique. *Copyright © 2002 IFAC*

Keyword: power generation, pollution, dynamic modeling, neural networks, genetic algorithms.

1. INTRODUCTION

The strict requirement for NOx-emission from combustion systems, particularly power generation boilers, has motivated numerous researchers and produced a wide variety of methods. With respect to NOx emission reduction in power generation boilers, existing and developing technologies may be grouped into two main areas, namely combustion modification and post combustion techniques. Advanced control systems (the area covered by this paper), or indeed advisory systems, claiming to reduce NOx emission while maintaining the combustion efficiency are essentially combustion modification systems.

It is estimated that an advanced control system can reduce NOx emission by between 15% and 25% (ETSU, 1997). However, an important step in developing any advanced control or advisory system is the ability to produce a simple but realistic model, which captures the relationship between the plants operational inputs and the NOx output. This paper studies NOx emission modeling using artificial neural networks.

NOx formation is a highly complex and nonlinear process. Existing knowledge relating to the combustion process invariably results in large sets of partially differential equations (PDEs), which in turn produces CFD (computing fluid dynamics) models of one- two- or three- dimensions. Although such models are useful for plant design they are too complex for control purpose. Moreover, the NOx formation mechanisms in coal-fired power generation plants are still under investigation, and intermediate variables such as the distributions of temperature, nitrogen and oxygen within the furnace are not directly measurable; making the development of simplified lumped parameter models difficult.

Linear and nonlinear models based on plant data have been developed using system identification techniques (Li and Thompson, 2001), however, how to generalize the prediction performance of these models is still a problem. Essentially, in power generation plants various operational conditions can affect the overall NOx emission level and this is not the same in all plant. Moreover, even in the same plant operating continuously there will be some changes in the data due to daily operational variations and seasonal electrical load requirements (Li and Thompson, 2000; Li and Thompson, 2001).

As universal approximators, Artificial Neural Networks (ANN) such as the Multi-Layer Perceptron network (MLP) and Radial Basis Function Neural Network (RBF) have found wide applications in non-linear system modeling and control, including the estimation of NOx emissions in power generation plants (Ferretti and Piroddi, 2001). In neural network modeling, there exist two major issues, i.e. training and configuration. However the underlying principle for these two issues is to develop a neural network model with less complexity and better generalization capacity (Li and Thompson, 2000). Most MLP training algorithms are recursive learning algorithms based on Newton-type gradient-descent techniques. Most recent advances in training have used second-order optimization techniques, and typically involve the calculation of at least an approximate Hessian Matrix associated with the function to be optimized. Some popular update algorithms are the LM (Levenberg-Marquart) method, the Broyden-Fletcher-Golfarb-Shanno (BFGS) algorithm, the scaled conjugate gradient algorithm, etc. In general, these analytic training algorithms can get stuck at local minima, particularly when the training data is large.

In this paper, a MLP model is built for the prediction of NOx emission in a coal-fired power generation plant and uses a genetic algorithm to train the model rather than a first-order or second order optimization technique. In order to avoid over-training, two data sets are involved, i.e. one data set is used for searching the weights and bias, the other set is used for validation. The fitness function for GA based training is the combination of the training error and validation error. In order to control the complexity of the MLP model, a final prediction error type criterion is used as the ANN selection criterion. The GA-based MLP model is then tested over different periods of plant operation.

2. MLP TRAINING BY GENETIC ALGORITHM

The Genetic Algorithm (GA) was first proposed by Holland in 1975 and it now has become well recognized as a powerful optimization scheme (Glodberg, 1989). This is due to its ability to solve multi-objective, non-differentiable and even NP problems. The main concept is the chromosome and its associated operations, namely: selection, crossover, mutation and replacement. The algorithm starts with a set of solutions (represented by chromosomes) called a population. Solutions from one population are taken and used to form a new population. It is hoped that the new population will be better than the old one. Solutions that are selected to form new solutions (offspring) are selected according to their fitness and the more suitable they are the more chances they have to reproduce. This relatively simple process is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied. Although this is not a mathematically guided optimization scheme, this technique can solve complex systems that other techniques might not have the ability to accomplish (Peng, et al, 2001). The basic GA routine is summarized as follows:

Algorithm 1 Basic GA Routine

Step 1 Start: Generate a random population of μ chromosomes (suitable solutions for the problem). The value in each bit of a chromosome is selected randomly within its predefined range of values.
Step 2 Fitness: Evaluate the fitness $f(x)$ of each chromosome x in the population.
Step 3 New populations: Create a new population by repeating the following until a new population is found

- Selection. Select two parent chromosomes from a population according to their fitness (the better their fitness the higher the probability of selection)
- Crossover. With a cross over probability form a new offspring. Without crossover the offspring would be an exact copy of the parents.
- Mutation. With a mutation probability mutate new offspring at each locus (position in chromosome). The mutation probability is in general very low, within the range of (0.0, 0.1];
- Accepting. Place new offspring in a new population

Step 4 Replace: Use new generated population for a further run of algorithm
Step 5 Test: If the end condition is satisfied, stop, and return the best solution in current population
Step 6 Loop: Go to step 2

Applications of GA for artificial neural networks can be classified into two categories (Blanco, et al 2001). One is to use GA as a means to learn artificial neural network connection weights that are coded, as binary or real numbers, in a genetic algorithm string (chromosome). The other is to use the genetic algorithm to evolve and select the artificial neural network architecture, together or independently from the evolution of weights. In this paper, GA is used to train the network weights, however, the complexity of the network is controlled by the final prediction error type criterion. Since generalization capacity is the main purpose of the system modeling, two sets of data are used in the GA based training process, one is used for training the other is used for validation. GA based training is required to find the optimal weights and bias minimizing the fitness function, that is the combination of the training error and validation error.

3. PLANT DESCRIPTION AND GA BASED MLP MODELING OF NOX EMISSION

3.1 General description of the power plant and identification of operational variables for modeling

In this paper power station NOx emission is studied in a dual fired plant. The power plant's drum boiler produces full load 300 MWe with oil firing or 200 MWe with coal firing. The boiler was designed to supply its turbine with steam at a temperature of 540°C and up to a pressure of 162 bars. There are four burner boxes, one burner box on each corner. Low NOx burners are fitted which will supply the furnace with oil or pf coal. All of the sections in each burner box tilt in unison through ±25°, relative to the horizontal. This is achieved by means of a burner tilt mechanism.

Coal, the major source of fuel and the one considered, is taken from the bottom of a coalbunker, pulverized and then entrained in a hot primary airflow. The coal delivery system is approximately synchronized with the furnace-burning rate. Each burner box houses a separated over fire air (SOFA) box, which admit fuel and secondary air streams into the furnace. These streams are directed at tangents to imaginary firing circles in the center of the furnace. The tangential firing creates turbulence in the combustion area that ensures the thorough mixing of fuel and air streams necessary for complete combustion. Low momentum burners are employed to achieve a longer flame path, leading to reduced flame temperatures. The nozzles in the burner box tilt in unison in the vertical plane to control the position of the fireball, and thus the temperatures in the superheat and reheat regions of the boiler.

The major part of NOx emission in power generation plant has been found to be NO. According to De Soete (1975), there are three main sources of NO in combustion, namely thermal NO, Prompt NO and Fuel NO. Thermal NO results from the reaction of atmospheric nitrogen and oxygen at high temperature, prompt NO is formed by the reaction of nitrogen with hydrogen derived radicals in the fuel-rich zone of combustion, while fuel NO results when nitrogen compounds present in the fuel are released and react with oxygen. In coal-fired power plant, fuel NO is the major contribution to NOx emission, and some of the fuel NO can be released from the devolitisation of the fuel while some from the oxidation of the char. Therefore, according to the NOx formation mechanism, the following operational variables will contribution to the overall NOx emission.

(1). The overall mass flow of fuel, denoted as m_f with unit (Kg/s).

(2). The overall mass flow of air, denoted as m_a (Kg/s).

(3). Specifically, mass flow of primary air and mass flow of secondary air, denoted as m_{pa}, m_{sa} (Kg/s).

(4). The burner tilt position, denoted as q (degree).

Considering that m_{pa}, m_{sa}, m_a have only two independent variables, there are actually four independent manipulated variables, defined as

$$\begin{aligned} u_1(t) = m_f(t),\ u_2 = m_{pa}(t) \\ u_3(t) = m_{sa}(t),\ u_4(t) = q(t) \end{aligned} \tag{1}$$

In the power station, emission are measured every minute. After analyzing the power plant data, two sets of data W_1 and W_2 with 1100 points and 1200 points respectively are used for modeling. Each point includes the values for the 4 manipulated variables and NOx emission measurement for every minute. The ranges of values for data set 1 (W_1) are as follows:

NOx emission (unit: ppm):
$NO_x \in [252.9, 452.5]$
Mass flow of fuel (unit: Kg/s):
$m_f \in [11.43,\ 28.37]$;
mass flow of primary air (unit: Kg/s):
$m_{pa} \in [24.69,\ 56.21]$;
mass flow of secondary air (unit: Kg/s):
$m_{sa} \in [144.25, 196.81]$
burner tilt position (unit: degree):
$q \in [48.6,\ 87.58]$

The ranges of values for data set 2 (W_2) are:

$NO_x \in [254.1, 431.1]$
$m_f \in [9.05,\ 20.32]$;
$m_{pa} \in [31.99,\ 51.12]$;
$m_{sa} \in [143.27, 182.9]$
$q \in [42.35,\ 72.11]$

It can be found that the two data sets have different ranges of data. Considering that these different variables have typical values that differ significantly, all values are pre-processed by applying a linear transformation. To do this, we treat each of the input variables independently, and for each variable u_i we calculate its mean $\bar{u}_i$ and variance σ_i with respect to the training set. We can define a set of re-scaled variables given by

$$\tilde{u}_i^j = (u_i^j - \bar{u}_i)/s_i \tag{2}$$

The output is also treated in the same way.

3.2 GA based MLP training and architecture selection

For nonlinear dynamic system modeling using neural networks, two general issues stand out. One is to choose the network architecture, and the other is to choose the learning algorithm. It has been proved that neural networks may approximate a wide selection of nonlinear systems to arbitrary closeness given a single hidden layer with a sufficient number of nodes. However, 1) a neural network with any number of hidden nodes will not necessarily nest the true structure of the real system; 2) the samples used for training is limited, and unlikely cover all operation conditions. Therefore, the final model will inevitably produce bias solutions in future applications.

Consider a MISO system that could be represented in a NARX form:

$$S1:\ y(t) = f(y^{t-1}, u_1^{t-d_1}, \ldots, u_m^{t-d_m}) + x(t) \quad (3)$$

where t is the time index, $y(t)$, $u_1(t), \ldots, u_m(t)$ are the output and inputs sequences respectively, and $x(t)$ is a sequence of independence random variables with zero mean and variances l. $f(\bullet)$ is some non-linear function and $d_1, \ldots, d_m$ are time delays.

$$\begin{aligned} &y^{t-1} = [y(t-1), \ldots, y(t-n_y)], \\ &u_k^{t-d_k} = [u_k(t-d_k), \ldots, u_k(t-d_k-n_{uk})] \quad (4) \\ &k = 1,2,\ldots,m \end{aligned}$$

Suppose that the ANN model could be represented as:

$$S2:\ \hat{y}(t) = ANN(y^{t-1}, u_1^{t-d_1}, \ldots, u_m^{t-d_m}; w) + e(t) \quad (5)$$

where t is the time index, w is the modifiable vector parameter, $\hat{y}(t)$, $u_1(t), \ldots, u_m(t)$ are the output prediction and input sequences respectively, $e(t)$ the modeling error. $ANN(\bullet; w)$ is some non-linear function determined by the corresponding neural network, $d_1, \ldots, d_m$ are time delays and

$$e(t) = y(t) - \hat{y}(t) \quad (6)$$

Two sets of data are used in training denoted as W_1 and W_2 respectively, and it is supposed that W_1 has N_1 samples and W_2 has N_2 samples.

A sum-squared error function is defined as

$$E(w) = \sum_{j=1}^{N_i} (y_i^j - \hat{y}_i^j)^2 \quad (7)$$

where N_i , $i=1,2$ is the number of samples in W_1 and W_2, w is the adjustable vectors of all weights.

It is supposed that, the sum-squared-error functions for the two sets of data W_1 and W_2 are denoted as $E(w)\big|_{W_1}$ and $E(w)\big|_{W_2}$ respectively, the adjustable vector w which is searched within the error space $E(w)\big|_{W_1}$ is denoted as w_{W_1}, and the adjustable vector w based on the calculation of $E(w)\big|_{W_2}$ is denoted as w_{W_2}. Then, the training of ANN based on two set of data is described as:

Problem formulation 1: To find the optimal adjustable vector w^* *of ANN when searching* w_{W_1} *within the error space of* $E(w)\big|_{W_1}$ *such that* $E(w)\big|_{W_2}$ *is minimized.*

The question could be formulated as:

$$w^* = \min_{w_{W_1}} E(w)\big|_{W_2} \quad (8)$$

And the performance of the trained ANN based on (8) is defined as $SSE(w^*)$:

$$SSE(w^*) = E(w^*)\big|_{W_1} + E(w^*)\big|_{W_2} \quad (9)$$

In the above, Ω_1 is in fact used for searching the optimal adjustable vectors, while Ω_2 is used for validation.

It is well known that as the training goes on, the generalization error can start to increase at certain point instead of decreasing. Therefore, the strategy taken in this study is to use one data set, i.e. W_1, to search for the optimal adjustable vectors, while another data set, i.e. W_2 is used for validation. The GA based training process is to find an optimal point in the search space that satisfies (8). Since there may be many local minima in the error space a GA is an ideal optimal tool to search the solution satisfying (8).

In this paper, MLP network with one hidden layer is used to model the NOx emission. Let u_i, $i=1,2,\ldots,p$ be the inputs, y_i, $i=1,2,\ldots,q$ be the output nodes, e_i, $i=1,2,\ldots,n$ be the hidden nodes, then, we have:

$$e_i = f\left(\sum_{j=1}^{p} w_{ij}^{eu} u_j + b_i^e\right) \quad (10)$$

and

$$e = f(w^{eu} u + b^e) \quad (11)$$

where f is the activation function, w_{ij}^{eu} are adjustable weights, b_i^e are bias, e,u,b^e are vectors, w^{eu} is a matrix of weights.

$$y_i = (\sum_{k=1}^{n} w_{ik}^{ye} e_k + b_i^y) \tag{12}$$

and

$$y = (w^{ye} e + b^y) \tag{13}$$

where, w_{ik}^{ye} are adjustable weights, b_i^y are bias, y,u,b^y are vectors, w^{ye} is a matrix of weights.

In this study, GA is used to search the weights and bias of the activation functions for the hidden nodes in the hidden layer as well as for the output nodes. The chromosome representation, each of which corresponds to a candidate ANN model, is proposed as follows:

W_1	B_1	¼ ¼	W_n	B_n

where W_i, B_i , $i=1,2,\ldots,n+q$ are the weight and bias sections for the hidden nodes and output nodes; n is the number of hidden nodes, q is the number of output nodes. W_i $i=1,2,\ldots,n$ are further represented as follows:

w_{i1}	w_{i2}	¼ ¼	w_{ip}

where w_{ij}, $j=1,2,\ldots,p$ are weights for *ith* node.

In the four GA operations in algorithm 1, two-point-crossover is used. For example, two individuals are selected randomly from the mating pool based on probability, and the crossover produces two off springs: two positions are selected at random and the bits are exchanged between the points:

Parent 1: **0 1 | 0 1 | 0 0** offspring 1: **01** | 1 0 | **0 0**
Parent 2: 1 0 | 1 0 | 0 1 offspring 2: 1 0 | **0 1** | 0 1

In order to apply algorithm 1 to search for the ANN weights and bias, the fitness function is defined as

$$SSE(w_{W_1}) = E(w_{W1})\Big|_{W_1} + E(w_{W_1})\Big|_{W_2} \tag{14}$$

where w_{W_1} is the adjustable vector that is searched within the error space $E(w)\Big|_{W_1}$. $E(w)\Big|_{W_1}$ and $E(w)\Big|_{W_2}$ are sum-squared-error for the two sets of data W_1 and W_2 .

GA based training is only applied to a fixed neural network.

In order to reduce the complexity of the MLP model, a final prediction error criterion is applied. From system identification, it is known that various criteria have been proposed to compare and select an appropriate model structure, such as F-test, Akaike's information criterion (AIC), and the final prediction error (FPE) criterion (Soderstrom and Stoica, 1989). In the following model construction algorithm, a FPE-type criterion will be used to select the MLP model structure, and it is expressed as follows:

$$PFE_{like} = SSE(w)[1 + \frac{2m}{N}] \tag{15}$$

where $SSE(w)$ is the fitness function defined in (14), m is the number of weights and biases in the MLP network and N is the number of samples.

The MLP network model for NOx emission has the following form:

$$y(t) = ANN(y(t-i), u_j(t-i); w), i = 1,2,3,4, j = 1,2,3,4 \tag{16}$$

where $y(t)$ is the NOx emission, u_j, $j=1,2,3,4$ are the system manipulated variables as defined in (1).

The NOx emission modeling procedures are summarized as follows:

Algorithm 2 GA based MLP modeling

Step 1 Initialization: Select training and validation data sets, set stop criterion (the maximal number of generations), population size, and other parameters for the GA algorithm, i.e. algorithm 1. Set maximal number of hidden nodes in the MLP network, set initial number of hidden nodes H_n;
Step 2 Training: Apply algorithm 1 to train the MLP network;
Step 3 Update: Calculate PFE_{like}, and store the parameters for the selected MLP model, $H_n=H_n+1$;
Step 4 Check: Check whether H_n exceed the maximal number. Yes, stop, the MLP model with the minimal PFE_{like} is selected; No, go to step 2.

In this study, plant data from power station operations are used in training and validation. GA evolution in algorithm 1 will stop after 250 generations, and the size of population pool is 50. The crossover probability is chosen to be 0.85, and the mutation probability is chosen to be 0.02. The number of hidden nodes in the MLP network is selected to be between 2 to 30. And based on PFE_{like}, the MLP network with 12 hidden nodes is the best selection.

The MLP model is tested over different plant operation periods, instead of using past measurement as the input in the MLP model, the past estimated output (plant data of NOx emissions) is used, and the MLP model for long-term prediction therefore takes the following form:

$$y(t) = ANN(\hat{y}(t-i), u_j(t-i); w) \tag{17}$$

where $y(t)$ is the NOx emission, $\hat{y}(t)$ is the MLP model estimation, u_j, $j=1,2,3,4$ are the system inputs.

Fig. 1, 2 and 3 show the comparison of model prediction with plant data of NOx emission. Obviously the MLP model is able to give a reasonaly good prediction performance over unseen data, however there do exist DC excursions in some periods (see Fig. 3) which means that the nonlinearity of the process is not fully captured by the MLP model.

4. CONCLUSION

A MLP model has been built to predict the NOx emission in a coal-fired power generation plant using genetic algorithm-based neural network modeling. To avoid over-training, two data sets are involved, i.e. one data set is used for searching the weights and bias, the other set is used for validation. The fitness function for GA based training is the combination of the training error and validation error.

In order to control the complexity of the MLP model, a final prediction error type criterion has been used. The GA-based MLP model has been tested to predict NOx emission over different periods of operation, showing the merits of this modeling technique.

ACKNOWLEDGEMENTS

Acknowledgement is made to the British Coal Utilisation Research Association and the UK Department of Trade and Industry for a grant in aid of this research but the views expressed are those of the authors, and not necessarily those of BCURA or the Department of Trade and Industry.

REFERENCES

Blanco, A., M. Delgado and M. C. Pegalajar (2001). A real-coded genetic algorithm for training recurrent neural networks. *Neural Networks*, **14**, 93-105.

De Soete, G.G., (1975). Overall reaction rates of NO and N_2 formation from fuel nitrogen. *15th Symposium (international) on Combustion*, The Combustion Institute, 1093-1102.

ETSU (1997). *Coal R&D programme*. Technology status report: NOx control for puliverised coal-fired power plant, ETSU, Harwell.

Glodberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc.

Li, K., S. Thompson (2000). Developing NOx Emission Model for a Coal-fired Power Generation Plant Using Artificial Neural Networks. *Proceedings of UKACC International Conference on CONTROL 2000*, Cambridge, pages 6.

Li, K., S. Thompson (2001). NOx emission models for operation and control of power generation boilers. 6th International Conference on Technologies and Combustion for A Clean Environment, **2**, 889-895. Porto, Portugal.

Ferretti, G., L. Piroddi (2001). Estimation of NOx emissions in thermal power plants using neural networks. *ASME Journal of Engineering for Gas Turbines and Power*, **123**, 465-471.

Peng, J., K. Li, S. Thompson (2001). GA based software for power generation plant NOx emission modelling. *6th International Conference on Technologies and Combustion for A Clean Environment*, **2**, 881-887. Porto, Portugal.

Soderstrom T. and P. Stoica (1989). *System identification*. Prentice Hall, London.

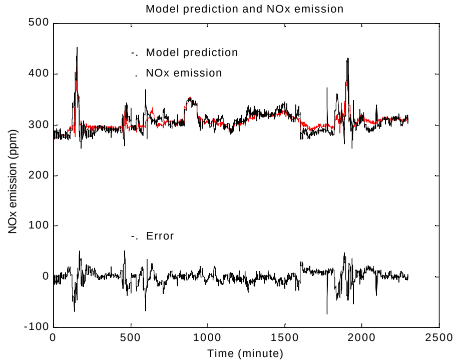

Fig. 1 Prediction of MLP model over period 1

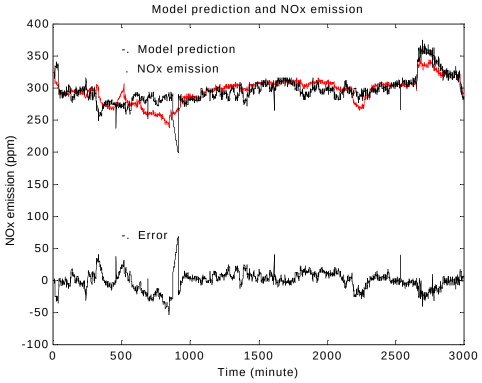

Fig. 2 Prediction of MLP model over period 2

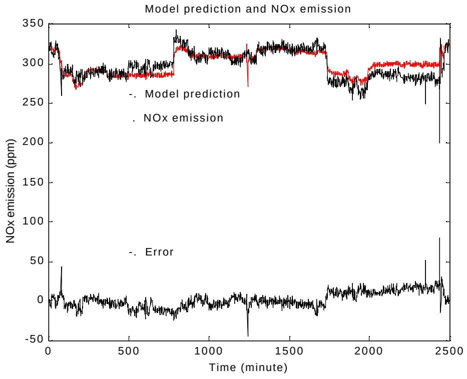

Fig. 3 Prediction of MLP model over period 3