

Improved Prediction of Nitrogen Oxides Using GRNN with K-means Clustering and EDA

Ligang Zheng, Shuijun Yu, Wei Wang, Minggao Yu

School of Safety Science and Engineering, Henan Polytechnic University, Jiaozuo 454000, China

zhengligang97@163.com

Abstract

The current study presented a generalized regression neural network (GRNN) based approach to predict nitrogen oxides (NO_x) emitted from coal-fired boiler. A novel 'multiple' smoothing parameters, which is different from the standard algorithm in which only single smoothing parameter was adopted (Matlab neural network toolbox, for example), were assigned to GRNN model. K-means clustering algorithm was developed so as to reduce the number of smoothing parameters. The training data was firstly partitioned into groups (the number of groups was much smaller than that of training samples) using K-means clustering. A smoothing parameter was then assigned to this group. A recently emerging estimation of distribution algorithm (EDA) was employed to optimize the multiple smoothing parameters. EDA presented in this paper was a kind of optimization algorithm based on Gaussian probability distribution. As a case study, the proposed approach was applied to establish a non-linear model between the parameters of the coal-fired boiler and the NO_x emissions. The results showed that the number of cluster has significant effect on the predictive accuracy of GRNN model. GRNN model with multiple smoothing parameters showed better agreement than that with only one smoothing parameter. The modeling errors on the testing subset were 1.24% and 1.62% for GRNN models trained by the present algorithm and the standard algorithm, respectively.

1. Introduction

Nitrogen oxides (NO_x) emitted from coal combustion systems is a significant pollutant source in the environment as the utilization of fossil fuels continues to increase, and the monitoring of NO_x emissions is an indispensable process in coal-fired power plant so as to control NO_x emissions. Though

the hardware-based CEMS is capable of measuring NO_x, the systems are costly to purchase, install and maintain. A potentially attractive alternative to installing a CEMS is the use of a PEMS which estimate the NO_x emissions on the basis of their dependence on other relevant system variables by using suitable algorithms. That means stack gases from the combustion chamber are possible to be predicted indirectly.

In the recent years, an "iterative" back propagation neural network (BPNN) is widely accepted as a technology offering an alternative way to control the combustion process. A large number of literatures cover the application of BPNN to pollution emissions monitoring. BP neural networks are developed for the modeling and control of the NO_x emissions from coal-fired boilers [1-5]. However, the BP neural network suffers from a number of weaknesses, which include the needs for numerous controlling parameters, difficulty in obtaining a stable solution and the danger of over-fitting. The uncertainty in the solution show that the results given by BPNN are not reliable even if the network structure, model parameters, training algorithms are determined. Contrary to BPNN, GRNN is a "one-pass" neural network and needs not any iterative calculation [6, 7]. Therefore, the solution of GRNN is certain and stable. The aim of the present study is to propose an alternative way to the popularly used BPNN to predict the NO_x emissions from coal-fire power plants. In order to regulate the predictive accuracy of GRNN model with larger freedom extent, multiple smoothing parameters with K-means clustering were optimized by EDA.

2. Methodology

2.1 Theory and network structure of GRNN

GRNN is based on the following formulation and parameters from statistics [6]:

$$\hat{y}(x) = \frac{\sum_{i=1}^N y_i \exp(-\frac{D_i^2}{2\sigma_i^2})}{\sum_{i=1}^N \exp(-\frac{D_i^2}{2\sigma_i^2})} \quad (1)$$

$$D_i^2 = |x - x_i|^2 \quad (2)$$

$$W_i = \exp(-\frac{D_i^2}{2\sigma_i^2}) \quad (3)$$

Where $\hat{y}(x)$ is output of the estimator, x is the estimator input vector, D_i is the Euclidean distance between two input vectors, x and x_i . N is the number of training samples. W_i is Gaussian kernel function.

The estimate $\hat{y}(x)$ can be visualized as a weighted average of all of the observed values, y_i , where each observed value is weighted exponentially according to its Euclidean distance from x_i .

Figure 1 shows a schematic depiction of the four layers GRNN. The first, or input layer, stores an input vector x . The second is the pattern layer which computes the distances $D(x, x_i)$ between the incoming pattern x and stored patterns x_i . The pattern nodes output the quantities $W(x, x_i)$. The third is the summation layer. This layer computes O_j , the sums of the products of $W(x, x_i)$ and the associated known output component y_i . The summation layer also has a node to compute C , the sum of all $W(x, x_i)$. Finally, the fourth layer divides O_j by C to produce the estimated output, that is a localized average of the stored output patterns [6].

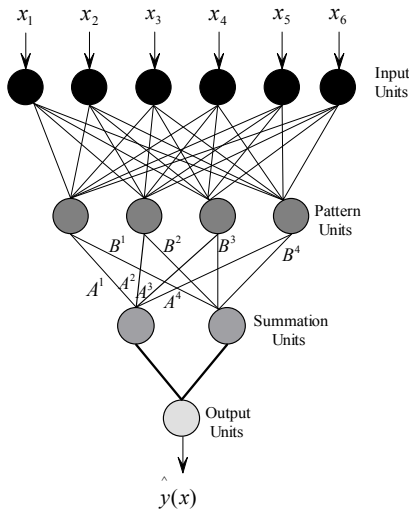


Figure 1. The network structure of GRNN

A critical consideration in the effectiveness of GRNN is the determination of optimal values for σ_i . As σ_i becomes very large GRNN output approaches the mean of the training set outputs; and as σ_i becomes very small GRNN output approaches the output pattern of the training set, which may not generalize well.

Intermediate values typically result in the best generalization. The value of σ_i giving the smallest error should be used in the final network.

In the standard algorithm as shown in Matlab neural network toolbox, only single smoothing parameter was assigned to all training samples. That means, $\sigma_1 = \sigma_2 = \dots = \sigma_{N-1} = \sigma_N$, as shown in Eq.(1). However, it is not necessarily the case that this algorithm always presents good predictive accuracy. In this work, different values were assigned for each smoothing parameters.

2.2 K-means clustering

Obviously, the number of smoothing parameter is equal to the number of training samples, as shown in Eq. (1). It is not computationally efficient if the number of training sample is very large. In order to reduce the number of the smoothing parameters, K-means clustering was firstly employed to partition the training data into groups. In each group, the smoothing parameter was identical for all training samples. The number of multiple smoothing parameters is then equal to the number of clustering, K , rather than the number of total training set, N . Therefore, the number of smoothing parameter is reduced compared to the number of training data.

The K-means algorithm attempts to find the cluster centers, (c_1, \dots, c_K) , such that the sum of the squared distances (this sum of squared distances is termed the Distortion, D) of each data point (x_i) to its nearest cluster centre (c_k) is minimized, as shown in Eq. (4), where d is some distance function. Typically d is chosen as the Euclidean distance. A pseudo-code for the K-means algorithm is shown in Algorithm 1[8].

$$D = \sum_{i=1}^N [\min_{k=(1 \dots K)} d(x_i, c_k)]^2 \quad (4)$$

Algorithm 1 K-means Algorithm

- (1) Initialize K centre locations.
- (2) Assign each x_i to its nearest cluster centre c_k .
- (3) Update each cluster center c_k as the mean of all x_i that have been assigned as closest to it.
- (4) Calculate $D = \sum_{i=1}^N [\min_{k=(1 \dots K)} d(x_i, c_k)]^2$
- (5) If the value D has converged, then return (c_1, \dots, c_K) ; else go to Step 2.

2.3 Estimation of distribution algorithm

EDA directly extracts the global statistical information about the search space from the search so far and builds a probabilistic model of promising solutions. New solutions are sampled from the model thus built. In this work, Gaussian probability distribution was used. The density function for the j th

variable to be optimized (total K variables as generated by K-means clustering) in the i th solution of population (S solutions in the initial population) at the itr -th generation (maximum G iterations) is written as follows [9-10]:

$$g_{i,j}(x, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

$$\mu = x_{i,j}, \quad \sigma = \frac{c_1}{\sqrt{k-1}} \sqrt{\sum_{l=1}^S (x_{l,j} - x_{i,j})^2} \quad (6)$$

c_1 is a constant. In the ranking vector, $R=[r_i]_{S \times 1}$, element r_i is a positive integer, showing the rank of the i th solution in the solution population. The rank is evaluated according to the objective function value. The smaller the value, the better is the corresponding solution. For the feasible solutions, ranks are assigned according to their objective function values. Assign 1 to r_i if the i th solution is the best solution, assign 2 if it is the second best, and so on.

According to R , the individual preference vector, Ω , can be determined. It can be calculated as follows:

$$\omega_i = \frac{1}{c_2 S \sqrt{2\pi}} e^{-\frac{(r_i-1)^2}{2(c_2 S)^2}}, \quad i=1, 2, \dots, S \quad (7)$$

c_2 is a constants. Then, select one Gaussian function ($g_{h,j}$) for the j th variable to be optimized in the new solution from the S Gaussian functions ($g_{i,j}, i=1, 2, \dots, S$) according to the following probability distribution:

$$p_i = \frac{\omega_i}{\sum_{l=1}^S \omega_l}, \quad i=1, \dots, S \quad (8)$$

Generate a random number according to the selected $g_{h,j}$. If this number is between the upper and lower bounds of decision variable x_j , it is the new value for x_j . Otherwise, repeat this step until a satisfactory value is obtained.

By sampling g_l through g_K successively a complete new solution will be generated. Repeat this process will give another new solution.

Let $Pop(t)$ be the population of solutions at generation t . EDA works in the following iterative way.

Algorithm 2 EDA Algorithm

- (1) Generate a population with S solutions with K dimensions.
- (2) Evaluate the population, generate ranking values in R and preference vector Ω , according to (7).
- (3) Build Gaussian functions $g_{i,j}$.
- (4) Sample M new solutions according to Eq.(8) and (5)
- (5) $(S+M)$ solutions should be evaluated and ranked. After that, the first S best solutions need to be chosen to form a new solution population $Pop(t+1)$ for the next iteration. The corresponding ranking vector R and preference vector Ω should be recorded.

- (6) Go to Step 3 to initiate another iteration, if the maximum number of iteration has not been exceeded.

In this work, the evaluation function was the modeling error, i.e. the relative difference between the predicted NOx emissions by GRNN model and the measured NOx emissions. The smaller the modeling error, the better is the predictive accuracy.

3. A case study

The proposed approach mentioned above was employed to predict the NOx emissions of a coal-fired boiler. The GRNN model has 19 inputs that corresponded to 19 boiler parameters, respectively. The output of GRNN model was the NOx emissions. Total 670 samples were recorded experimentally. These samples were divided into two parts, i.e., the training set D_1 and the testing set D_2 . Here, 670 cases were in advance indexed by the arabic numeral, i.e., $idx=1, 2, \dots, 670$. Two hundred and twenty-four cases were chosen as the testing set D_2 with the indices, $idx=1, 4, 7, \dots, 670$, while remaining 446 cases named the training set D_1 were employed to train various models to capture the quantitative relation between the NOx emissions and 19 operational parameters of the studied utility boiler. To eliminate the unit influence of various operational parameter of the boiler, some necessary preprocessing of the raw data before feeding them into the GRNN model is needed. In this study, all the feature elements and the target values were scaled so that they fall into the range of $[0, 1]$. When using these models, the computed target value should be converted back into the same scales that were used for the original target values.

K-means clustering was firstly used to partition the training data, i.e. 446 training samples, into K groups. That means, the K smoothing parameters of GRNN model needed to be optimized by EDAs. The K optimized smoothing parameters was then employed to build GRNN model to predict the testing subset, i.e. the remaining 226 testing samples. The proposed approach was also compared with the standard algorithm in which only single smoothing parameter was assigned to all training samples.

4. Results and discussion on the case study

Figure 2 presented the modeling errors respectively averaged on the training set, the testing set and the total set as a function of the number of clustering. The averaged modeling error on the testing set was gradually deceased and was then reached to a plateau with the increasing number of clustering K . The trend

of the averaged modeling error on the training set was contrary to that of the testing set. Obviously, the generalization ability of GRNN model was improved with the increasing number of clustering. The best predictive accuracy was given with the number of clustering K equal to 70.

Figure 3 plotted the distribution of the modeling errors for training set, testing set as a function of the serial number of samples. The modeling errors for most of testing samples were less than 5%. The generalization ability of GRNN model was quite good.

The proposed method was also compared with the standard algorithm. In the standard algorithm, the modeling error decreased slightly with the single σ at first, and then increased dramatically. We chose $\sigma=0.111$ as the optimal spread parameter for the standard algorithm. The summary of our algorithm and the standard algorithm for NOx emissions monitoring was shown in Table 1. Judged by correlation factor and modeling error, our algorithm showed much better predictive result than the standard algorithm, especially for testing set. The reason lies in that the freedom degree of regulating the model accuracy was larger for the proposed algorithm than for the standard algorithm. It is concluded that our algorithm was very efficient in predict NOx emissions.

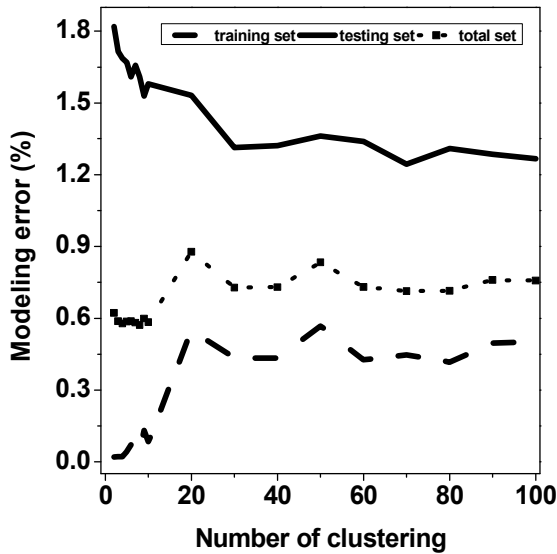


Figure 2. Effect of the number of clustering

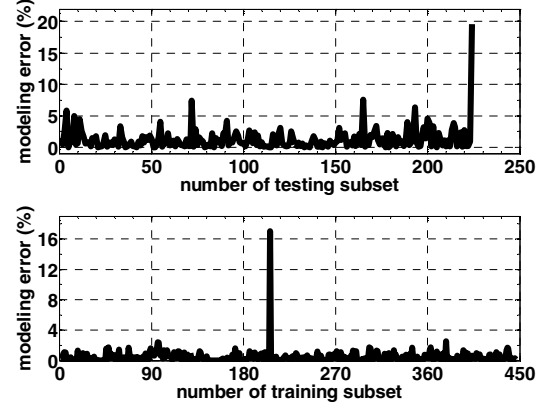


Figure 3. Modeling error distribution

Table 1 Comparison between the proposed and the standard algorithms

algorithm	Correlation factor		Modeling error/%	
	D ₁	D ₂	D ₁	D ₂
Our algorithm	0.99	0.96	0.45	1.24
Std. algorithm	0.98	0.93	0.59	1.62

5. Conclusion

The GRNN model provides an alternative to BPNN. Unlike the standard algorithm, the multiple smoothing parameters were assigned to K training data groups which were generated through K-means clustering. EDA was a feasible tool to optimize the multiple smoothing parameter of GRNN model. The number of clustering was a very important parameter to be considered. It could be determined by trial-and-error process. The algorithm presented in this work demonstrated superiority over the commonly-used single smoothing parameter model (Matlab neural network toolbox), especially the generalization ability.

6. Acknowledgment

The authors would like to acknowledge the financial supports from NSF of Henan Polytechnic University (646102). Acknowledgment is also made to Prof. Hao Zhou of Zhejiang University, China and Dr. Chunlin Wang of Hangzhou Dianzi University, China for providing the emissions data of the case boiler.

7. References

- [1] S.Graziani, N.Pitronel, M.G.Xibilia, and N.Barbalace, "Improving monitoring of NO_x emissions in refineries," *Conference on Instrumentation and Measurement Technology*. IEEE, Cormo, Italy, 2004, pp. 594- 597.

- [2] R.Frenken, C.Rozendaal, H.Dijk, and P.Knoester, "Development of an advisory system based on a neural network for the operation of a coal fired power plant," *Conference on Artificial Neural Networks*, IEEE, London, UK, 1996, pp. 197-202.
- [3] S.Tronci, R.Baratti, and A.Servida, "Monitoring pollutant emissions in a 4.8 MW power plant through neural network," *Neurocomputing*, Elsevier, Netherlands, 2002, pp. 3-15.
- [4] S.Matsumura, T.Iwahara, K.Ogata, S.Fujii, and M.Suzuki, "Improvement of de-NO_x device control performance using a software sensor", *Control Engineering Practice*, Elsevier, Netherlands, 1998, pp.1267-1276.
- [5] G.Baines, "Neural networks for boiler emission prediction", *16th IEEE Conference on Instrumentation and Measurement Technology*, IEEE, Venice, Italy, 1999, pp. 435-439.
- [6] D.F.Speccht, "A general regression neural network," *IEEE Transactions on neural network*, IEEE, USA, 1991, pp. 568-576.
- [7] O. Polat, T.Yildirim, "Genetic optimization of GRNN for pattern recognition without feature extraction", *Expert Systems with Applications*, Elsevier, Netherlands, 2007, doi:10.1016/j. eswa.2007.04.006.
- [8] S. J. Redmond, and C. Heneghan, "A method for initialising the K-means clustering algorithm using kd-trees", *Pattern Recognition Letters*, Elsevier, Netherlands, 2007, 965-973.
- [9] J.Xiao, Y.L.Huang, and H.Lou, "A probability distribution estimation based method for dynamic optimization", *AIChE Journal*, Wiley, USA, 2007, pp.1805-1816.
- [10] Y.H.Chen, Y.Zhang, and A.Abraham, "Estimation of Distribution Algorithm for Optimization of Neural Networks for Intrusion Detection System", *LNAI*, Springer-Verlag, Berlin Heidelberg, 2006, pp. 9-18.