A Neural Network Model for Predicting NO_x at the Mělník 1 Coal-powder Power Plant

Ivo Bukovský[1], Michal Kolovratník[2]

[1] *Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Instrumentation and Control Engineering, Technická 4, 166 07 Prague, Czech republic*

[2] *Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Energy Engineering, Technická 4, 166 07 Prague, Czech republic*

Correspondence to: ivo.bukovsky@fs.cvut.cz

Abstract

This paper presents a non-conventional dynamic neural network that was designed for real time prediction of NO_x at the coal powder power plant Mělník 1, and results on real data are shown and discussed. The paper also presents the signal preprocessing techniques, the input-reconfigurable architecture, and the learning algorithm of the proposed neural network, which was designed to handle the non-stationarity of the burning process as well as individual failures of the measured variables. The advantages of our designed neural network over conventional neural networks are discussed.

Keywords: dynamic neural networks, prediction, NO_x emissions, signal processing.

1 Introduction

Neural networks (NN) are a popular and widely studied real data-driven nonlinear modeling tool for complicated systems where mathematical-physical analysis is unavailable for deriving a model. Unlike analytical or linear models, NNs are black-box models, or sometimes gray-box models, that require proper design of their mathematical architecture and an efficient learning algorithm. For the principles of fundamental neural networks we may refer, e.g., to [1], and we may refer to less recent reviews [2, 3] for studies of NNs in energetic processes. For more recent works, including studies of conventional NN in energetic processes, we may refer to papers [5–8], which deal with computational intelligence tools (neural networks, genetic algorithms) focused on biomass combustion. The study of non-conventional neural architectures for modeling steady state hot steam turbine data and for modeling a large scale energetic boiler can be found in [9], where the advantages of a static quadratic neural unit (QNU, [1, 4]) and a special quadratic neural network [9] over conventional multilayer perceptron neural networks (MLP) are demonstrated, with reference to the overfitting and local minima problem, which are typical drawbacks of MLP (even with a single hidden layer NN). The advantage of QNU is its nonlinear input-output mapping, while this neural model is linear in its parameters [10] (unlike MLP). This allows us to monitor and maintain adaptation stability by a comprehensible evaluation of the eigenvalues of the weight update system [10] that offer promising opportunities for adaptive monitoring, modeling, and process optimization by adaptive nonlinear control. QNU can be seen as a component of higher-order neural network (HONN), sometimes also referred to as a polynomial neural network (PNN). The origins of these neural networks can be traced back to works [11–14], while the concept of the standalone higher-order neural units (HONUs) as a building component of HONN can be found in [1] and in [4]. The fundamental gradient-based learning rules for training dynamic neural networks are known as Real-time Recurrent Learning (RTRL) [15] and Back-Propagation Through Time (BPTT) [16, 17]. These algorithms can be made comprehensible and practically useful for real-time computations.

In this paper, we present the resulting neural network architecture that has been designed and tested for NO_x prediction for a pulverized coal firing boiler at the power plant “Elektrárna Mělník 1 (EME 1)”; the nominal steam load of this boiler is 250 tons per hour. The goal is to design and test a model that does not involve measured O_2 or CO in its input and that can potentially be used for optimizing the energetic process regarding NO_x and CO emissions of the pulverized firing boiler at EME 1. The resulting discrete-time dynamic (recurrent) neural network merges the concept of a conventional recurrent (MLP) neural network with QNU [4, 9]. The

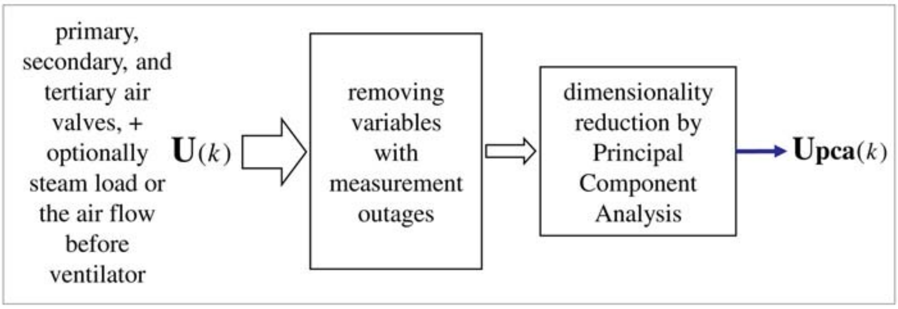
primary, secondary, and tertiary air valves, + optionally steam load or the air flow before ventilator
U(k)
removing variables with measurement outages
dimensionality reduction by Principal Component Analysis
Upca(k)

Figure 1: The data preprocessing before each reconfiguration and retraining of the neural network

data pre-processing and retraining strategy is described in the next section, and that in turn is followed by the mathematical notation of the neural architecture that led to the results shown in the section on discussion.

2 Data preprocessing and network training

The NO_x dynamics of the pulverized boiler is highly nonstationary, due to varying technical conditions of the boiler, varying quality of the coal powder, and also because of the measurement outages that occur quite often on an hourly basis. It was therefore not possible to obtain a neural network model that would reliably predict the NO_x emissions from the long term data. To handle the non-stationary nature of the boiler in EME 1, we arrived at the data preprocessing technique that is sketched in Figure 1, where $\mathbf{U}(k)$ is a matrix of recent history (a retraining window) of all measured input variables (excluding measured O_2, NO_x, and CO) at a reference time k that is particularly given as follows

$$\mathbf{U}(k) = \begin{bmatrix} u_1(k-Ntrain+1) & \dots & u_1(k-1) & u_1(k) \\ u_2(k-Ntrain+1) & \dots & u_2(k-1) & u_2(k) \\ \vdots & \ddots & \vdots & \vdots \\ u_n(k-Ntrain+1) & \dots & u_n(k-1) & u_n(k) \end{bmatrix}. \tag{1}$$

The measured input variables in $\mathbf{U}(k)$ are the primary, secondary, and tertiary air valves, and also optionally the steam load or the air flow before the ventilator (in total $n = 18, 19, 20$ variables) excluding O_2, NO_x, and CO. The principal component analysis (PCA) block is an application of PCA to the linearly correlated variables, so the number of input variables in $\mathbf{Upca}(k)$ is $m < n$, which importantly decreases the computation load while it maintains information in the measured input data (note, Figure 1 shows only a simplified sketch, while detailed implementation of PCA that benefits from process knowledge of the pulverized boiler at EME 1 may be provided on the basis of an official request for [19]). The structure of the resulting input data matrix $\mathbf{Upca}(k)$ that is used as the neural network input (after the preprocessing shown in Figure 1) is as follows

$$\mathbf{Upca}(k) = \begin{bmatrix} u_{pca_1}(k-Ntrain+1) & \dots & u_{pca_1}(k-1) & u_{pca_1}(k) \\ \vdots & \ddots & \vdots & \vdots \\ u_{pca_m}(k-Ntrain+1) & \dots & u_{pca_m}(k-1) & u_{pca_m}(k) \end{bmatrix}. \tag{2}$$

The presented data pre-processing technique removes variables with measurement outages. Principal component analysis results in a lower computational load, because of the reduced number of external inputs into the neural network ($m < n$). PCA has a filtering effect, and also contributes to more accurate calculations of matrix inversion with the BPTT training technique by reducing redundant and linearly correlated data.

3 Neural network for NO_x prediction

This section describes the mathematical notation of the designed neural network for NO_x prediction. This neural network is a discrete-time recurrent architecture, i.e., a non-linear difference equation system, composed of a recurrent hidden layer of conventional sigmoid neurons and with an output quadratic neural unit with feedbacks also from the output to its input. In particular, the neural network predictive model is given as follows. The window of external inputs for retraining the network at reference time k are the pre-processed measured variables $\mathbf{Upca}(k)$, as given in (2) and in Figure 1. The external inputs that enter the neural network for n_s samples ahead prediction at time k are in the last column of $\mathbf{Upca}(k)$, as follows

$$\mathbf{u_{pca}}(k) = \begin{bmatrix} u_{pca_1}(k) & u_{pca_2}(k) & \dots & u_{pca_r}(k) \end{bmatrix}^T, \tag{3}$$

where k is a reference sample time index and r is the dimension of the reduced vector of all measured external inputs by the PCA method. The input vector to the hidden layer of the neural network is given in (4) as

$$\mathbf{x}(k) = \begin{bmatrix} y_n(k+n_{ya}) & \dots & y_n(k-n_{yb}) & \mathbf{u_{pca}}(k+n_{ua})^T & \dots & \mathbf{u_{pca}}(k-n_{ub})^T & \xi(k) \end{bmatrix}^T, \tag{4}$$

where $yn(.)$ are step-delayed neural outputs; n_{ya}, n_{yb}, n_{ua}, and n_{ub} are input configuration parameters; and $\xi(k)$ is the step delayed feedback of the hidden layer outputs. The output of the hidden sigmoidal layer $\xi(k+1)$ (6) is calculated using the hidden layer weight matrix $\mathbf{W}$ (6) and using the classical sigmoid function (5), as follows

$$\phi(\nu) = \frac{2}{1+\mathrm{e}^{-\nu}} - 1, \tag{5}$$

where $\xi(k+1)$ is augmented with a unit as

$$\xi(k+1) = \begin{bmatrix} 1 \\ \phi(\mathbf{W}\cdot\mathbf{x}(k)) \end{bmatrix}, \tag{6}$$

where the unit allows the hidden layer (first column of $\mathbf{W}$) and also the QNU ($v_{0,0}$ in (7)) for biases, so the neural output is calculated by a quadratic neural unit [1,4,9,10], using (3)–(6), as follows

$$y_n(k+n_s) = \sum_{i=0} \sum_{j=i} v_{i,j} \cdot \xi_i(k+1) \cdot \xi_j(k+1). \tag{7}$$

The proposed dynamic neural network has purposely designed properties that are worth mentioning and explaining. The hidden recurrent layer of neurons, which is calculated in (6) as $\phi(\mathbf{W}\cdot\mathbf{x}(k))$, reduces cognitively (by training) the number of already PCA preprocessed neural inputs, and thus (6) results in the augmented vector of state variables $\xi(k+1)$ that are fed both forward to the QNU and also back to the network input $\mathbf{x}(k)$, as in (4). Without the first hidden layer, the number of input variables inputted directly into QNU would still be too large for the given 1-minute sampling period, as we feed an approximately twelve-minute history of each PCA preprocessed variable into the network input, i.e. $n_{ya} - n_{yb} = 12$ and also $n_{ua} - n_{ub} = 12$ (the estimated time constant of this pulverized firing boiler has been specified by experts as approximately 12 minutes). Also, the first layer (6) plays a filtering role due to its step delayed feedback to the network input (4), and its recurrent feedback naturally calls for training by the Backpropagation Through Time method (BPTT) [15–17], which is a powerful and efficient and yet practical optimization method, as it can be achieved by a combination of a gradient descent rule and the Levenberg-Marquardt algorithm [18]. The sigmoid function $\varphi(.)$, which is usually considered as a main nonlinearity of conventional neural networks, has another importance for this dynamic network, because the major nonlinearity is provided by the QNU [9,10,18]. The sigmoid function $\varphi(.)$ limits the output of the hidden layer into the given range of values $(-1,+1)$ that importantly assures stability of the hidden layer (as of a discrete time dynamic system, and this could not be so simply assured for continuous-time NNs). Then, there are always limited values entering the QNU, so its output is also naturally limited; thus, the stability of the state variables and of the output of the proposed neural network is naturally assured by preserving the sigmoid function in hidden neurons. As regards the stability of the learning algorithm, and thus its convergence, we proposed a novel approach to weight-update stability for gradient descent training of QNU in [10], and this approach is applicable to both static and dynamic QNU, and also to the hidden-layer weight system of this network for $\mathrm{NO_x}$ prediction.

4 Results and discussion

This section shows the results of 3-minute predictions of $\mathrm{NO_x}$ emissions (in fact the 3-minute floating averages) of the pulverized firing boiler at EME 1 by the proposed neural network ($n_s = 3$, sampling 1 minute). The recurrent network does not include measured $\mathrm{O_2}$, $\mathrm{NO_x}$, or CO on its input; the introduction of $\mathrm{NO_x}$ as a measured external input resulted in a prediction failure; the model learned to follow blindly the previous measured $\mathrm{NO_x}$,

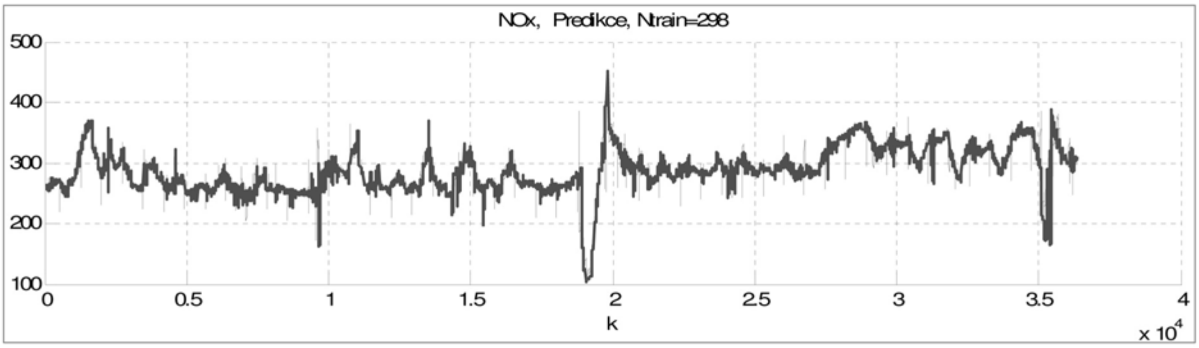

Figure 2: NO_x prediction by the neural network (1)–(7) with re-configurations and re-training (Figure 1)

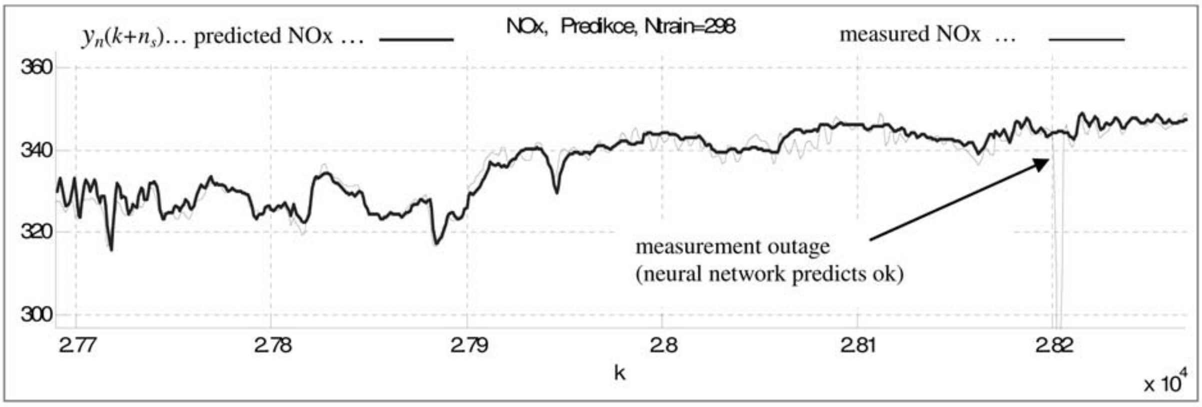

Figure 3: Detail from Figure 2 — a good prediction

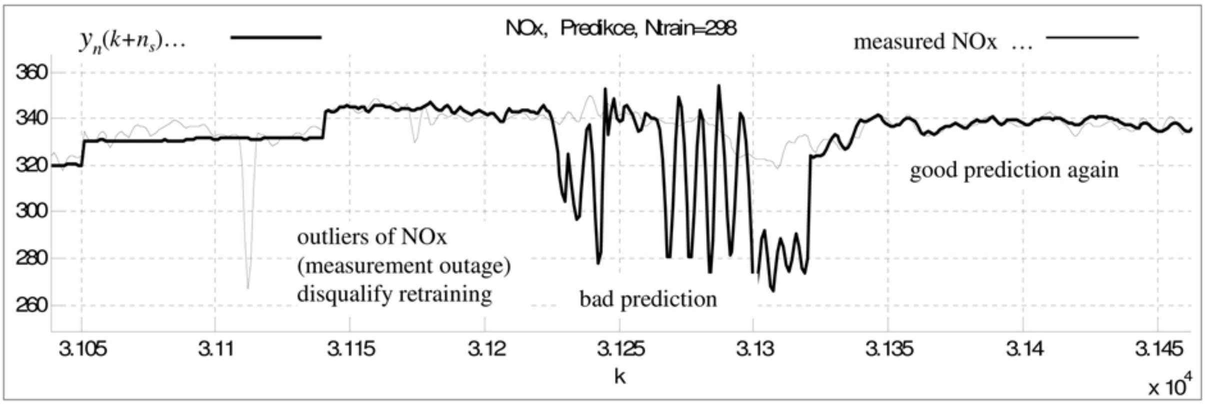

Figure 4: Detail from Figure 2 — a bad prediction due to outliers in the training data

which is a typical problem with the improper use of neural networks for complicated systems. The permanent computation run of prediction for 24 days, with one-minute sampling and retraining every 30 minutes, is shown in Figure 2.

Figure 2 shows superimposed 24-day recordings of measured NOx (thick line) and the three-minute prediction ($n_s = 3$) of NOx (bold line) (the three-minute floating average of NOx is predicted); the neural network is retrained every 30 minutes with 5 hours of the very last measured data (one-minute sampling, the model input excludes measured O_2 , NOx, CO). The network (1)–(7) was retrained every 30 minutes by the back propagation through time algorithm [18] with blindly selected most recent history of 298 samples (5 hours) of the measured process variables. Each retraining took less than 3 minutes of real computation time in Matlab on a PC (Win7, i7), and this is practical for real time retraining implementation.

The good performance of NOx prediction is apparent from the details in Figure 3. Also, Figure 3 shows a temporary measurement outage ($\sim$ 15 minutes) after sample $k = 2.82\text{E} + 4$; the output of the dynamic neural network substitutes the measurement outage; the good neural network prediction depends on availability of good retraining data in this observed period. However, this kind of NO_x outage affects retraining, see Figure 4. The prediction accuracy and prediction reliability for NO_x prediction depends significantly on the retraining data (here, the last 298 samples before each predicted value). The impact of NO_x outliers is apparent if we compare the prediction details in Figure 3 and Figure 4, and it is clear that another signal processing technique for selecting the retraining data needs to be involved in order to avoid NO_x outliers in the retraining data; the neural network fails in prediction after $k = 3.122\mathrm{E} + 4$ because of the poor retraining data and also because of the outliers at $k = 3.12\mathrm{E} + 4$. The prediction becomes correct again for $k > 3.133\mathrm{E} + 10$, because the related retraining data already does not include the outliers.

5 Conclusions

We designed and tested a non-conventional recurrent neural network for predicting the NO_x emissions of a pulverized firing boiler without using O_2, NO_x, or CO on the model input. The proposed method handles process non-stationarity by frequent retraining, and it handles the outages of input process variables by input data preprocessing (but not yet the outages of the predicted NO_x itself); it is assumed that this can be resolved by automatically supervised selection of the training data where NO_x outliers do not appear, and by avoiding unnecessary retraining.

Acknowledgement

This work has been supported by grant MPO_FR-TI1/538, and in part by grant SGS10/252/OHK2/3T/12.

References

[1] Gupta, M. M., Liang, J., Homma, N.: *Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory.* IEEE Press and Wiley-Interscience, John Wiley & Sons, Inc., 2003.

[2] Kalogirou, S. A.: Artificial intelligence for the modeling and control of combustion processes: a review, *Progress in Energy and Combustion Science*, 29, 2003, p. 515–566, Elsevier. ISSN 0360-1285.

[3] Mellit, A., Kalogirou, S. A.: Artificial intelligence techniques for photovoltaic applications: A review, *Progress in Energy and Combustion Science*, 34, 2008, p. 574–632, Elsevier. ISSN 0360-1285.

[4] Bukovsky, I., Bila, J., Gupta, M. M., Hou, Z.-G., Homma, N.: *Foundation and Classification of Nonconventional Neural Units and Paradigm of Nonsynaptic Neural Interaction, in Discoveries and Breakthroughs in Cognitive Informatics and Natural Intelligence.* in the ACINI book series ed. by Yingxu Wang, University of Calgary, Canada : IGI Publishing, Hershey PA, USA, 2009. ISBN 978-1-60566-902-1.

[5] Piteľ, J., Mižák, J.: Approximation of CO/Lambda Biomass Combustion Dependence by Artificial Intelligence Techniques. In *Annals of DAAAM for 2011 & Proceedings of the 22nd International DAAAM Symposium, Vienna, Austria,* 23–26th November 2011. Vienna : DAAAM International, 2011, p. 0143–0144. ISBN 978-3-901509-83-4, ISSN 1726-9679.

[6] Mižák, J., Piteľ, J.: Using Artificial Neural Networks for Biomass Combustion Process Control. In *Proceedings of 2nd International Seminar "System Analysis, Control and Information Processing"*, Divnomorskoje, Russia. [CD-ROM]. Rostov on Don: Don State Technical University, 2011, p. 343–348. ISBN 978-5-7890-0666-5.

[7] Piteľ, J., Boržíková, J., Mižák, J.: Biomass Combustion Process Control Using Artificial Intelligence Techniques. In *Proceedings of XXXVth Seminar ASR'2010 "Instruments and Control"*. Ostrava : VŠB-TU Ostrava, 2010, p. 317–321. ISBN 978-80-248-2191-7.

[8] Hošovský, A.: Genetic Optimization of Neural Networks Structure for Modeling of Biomass-Fired Boiler Emissions. *Journal of Applied Science in Thermodynamics and Fluid Mechanics*, Vol. **9**, No. 2, 2011, p. 1–6. ISSN 1802-9388.

[9] Bukovský, I., Lepold, M., Bílá, J.: Quadratic Neural Unit and its Network in Validation of Process Data of Steam Turbine Loop and Energetic Boiler, *WCCI 2010, IEEE Int. Joint. Conf. on Neural Networks IJCNN*, Barcelona, Spain, 2010.

[10] Bukovský, I., Bílá, J., Noriasu, H., Rodriguez, R.: Prospects of Gradient Methods for Nonlinear Control, *Automatizácia a riadenie v teórii a praxi ARTEP 2012*, Slovakia 2012. ISBN 978-80-553-0835-7.

[11] Ivakhnenko, A. G.: Polynomial Theory of Complex Systems, *IEEE Tran. on Systems. Man and Cybernetics.* Vol. **SMC-1**, 4, 1971, p. 364–378.

[12] Nikolaev, N. Y., Iba, H.: Learning Polynomial Feedforward Neural Network by Genetic Programming and Backpropagation, *IEEE Trans. on Neural Networks*, Vol. **14**, No. 2, March, 2003, p. 337–350.

[13] Taylor, J. G., Commbes, S.: Learning higher order correlations, *Neural Networks*, 6, 1993, p. 423–428.

[14] Kosmatopoulos, E., Polycarpou, M., Christodoulou, M., Ioannou, P.: High-Order Neural Network Structures for Identification of Dynamical Systems, *IEEE Trans. on Neural Networks*, Vol. **6**, No. 2, March 1995, p. 422–431.

[15] Williams, R. J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks, *Neural Comput.*, Vol. **1**, 1989, p. 270–280.

[16] Werbos, P. J.: Backpropagation through time: What it is and how to do it, *Proc. IEEE*, Vol. **78**, No. 10, Oct. 1990, p. 1 550–1 560. ISSN 0018-9219.

[17] Pearlmutter, B. A.: Gradient calculation for dynamic recurrent neural networks: a survey. *IEEE Transactions on Neural Networks*, **6**, 5, 1995, 1 212–1 228. doi: 10.1109/72.410363.

[18] Gupta, M. M., Bukovský, I., Noriasu, H., Solo, M. G., Hou, Z.-G.: Fundamentals of Higher Order Neural Networks for Modeling and Simulation, In *Artificial Higher Order Neural Networks for Modeling and Simulation*, ed. M. Zhang, IGI Global, 2012, (accepted, to appear in 2012).

[19] Bukovský, I., Křehlík, K.: *Testy neuronového modelu kotle elektrárny Mělník I. Výzkumná zpráva č. 6 – ZI00069/E06.* Ústav přístrojové a řídící techniky, Fakulta Strojní, ČVUT v Praze, 2011.