Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Data Mining (CIDM 2007)

A Data Mining Method Applied to a Metallurgical Process

Henrik Saxén and Frank Pettersson

Heat Engineering Laboratory, Åbo Akademi University, Finland

***Abstract* – The processes in metallurgical industry are often extremely complex and measurements from their interior are scarce due to hostile (high temperatures and pressure, as well as very erosive) conditions. Still, today's constraints on high productivity and minor impact on the environment require that the processes be strictly controlled. Mathematical models can play a central role in achieving these goals. In cases where it is not possible, or economically feasible, to develop a mechanistic model of a process, an alternative is to use a data-driven approach, where a black-box model is built on historical process data. Feedforward neural networks have become popular modeling tools for this purpose, but the selection of relevant inputs and appropriate network structure are still challenging tasks. The work presented in this paper tackles these problems in the development of a model of the silicon content in hot metal produced in the ironmaking blast furnace. A pruning method is applied to find relevant inputs and their time lags, as well as an appropriate network connectivity, for solving the time-series problem at hand. In applying the model, an on-line learning of the upper-layer weights is proposed to adapt the model to changes in the input-output relations. The findings of the analysis show results in good agreement with practical metallurgical knowledge and demonstrate the feasibility of the approach.**

***Index Terms* — Neural networks, pruning, selection of inputs, ironmaking, prediction of silicon content.**

I. INTRODUCTION

Neural networks have become popular tools for tackling nonlinear black-box modeling problems, partly due to their universal approximation capabilities [1]. In real-world applications, such as prediction tasks encountered in the metallurgical industry, several practical problems arise in neural network modeling. One is that there is an abundance of variables that potentially influence the output (dependent) variable, and a choice between these has to be made to avoid an over-parameterization of the solution [2]. Another equally important problem is that the measurements contain errors, which complicates the choice of network complexity: Generally, the number of network parameters has to be restricted to avoid fitting noise. Several constructive and destructive algorithms, with growing or shrinking networks, have been proposed (e.g., [3-5]) but many of these include retraining steps that require prohibitive computational efforts when applied to large problems. Some investigators have explicitly dealt with the challenging problem of selection of relevant inputs [6], without taking a stand on the required complexity of the model: Sarle [7] discusses the pitfalls of methods that address the problem of determining the importance of input variables.

This paper illustrates how a recently proposed efficient pruning method [8] can be used with advantage in a complex data-mining problem from the metallurgical industry, where both the relevant inputs and their time lags are to be detected in the modeling. The method, is briefly outlined in Section II, and is, furthermore, extended to on-line learning. Section III demonstrates that is yields good solutions to the hot metal silicon prediction problem. The final section presents some concluding remarks.

II. THE METHOD

This section describes the pruning algorithm in a nutshell, briefly discusses computational aspects, and illustrates the performance of the algorithm on an artificial data set. Finally, an extension to online learning is proposed.

A. The Pruning Algorithm

The pruning algorithm is based on feedforward neural networks of multi-layer perceptron type (often called backpropagation networks) with a single layer of hidden nonlinear units and a single linear output node. Given a sufficient number of hidden nodes, such a network has been shown to be able to approximate any continuous, twice differentiable function to any accuracy [9]. Fundamentally, the pruning algorithm is based on the practical experience that for such a network with an arbitrary choice of weights in its lower layer of connections, **W**, (cf. Fig. 1) there is usually a weight vector, **v**, to the output node that will lead to a relatively good solution, $\hat{y}$, of the approximation problem (for an example, see e.g. [8]).

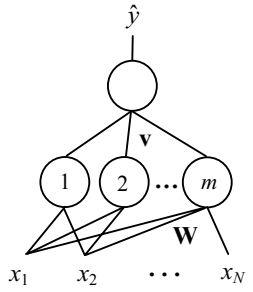

Fig. 1. Neural network and notation.

The vector **v** can be determined by a simple matrix inversion. With this as the starting point, the pruning algorithm proposed can be compactly written as:

1) Select a set of N potential inputs, **x**, and the output, y, to be estimated for the observations of the training set.
2) Choose a sufficient number of hidden nodes, m, and generate a random weight matrix, $\mathbf{W}^{(0)}$, for the lower part of the network. Set the iteration index to $k = 1$.
3) Equate to zero, in turn, each non-zero weight, $w_{ij}^{(k-1)}$, of $\mathbf{W}^{(k-1)}$, and determine the optimal upper-layer weight vector, $\mathbf{v}$, minimizing $F = \sum(\hat{y} - y)^2$ by linear least squares. Save the corresponding value of the objective function, $F_{ij}^{(k)}$.
4) Find the minimum of the objective function values, $\mathbf{F}^{(k)} = \min_{ij}\left\{F_{ij}^{(k)}\right\}$. Set $\mathbf{W}^{(k)} = \mathbf{W}^{(k-1)}$ and equate to zero the weight corresponding to the minimum objective function value, $w_{\tilde{i}\tilde{j}}^{(k)} = 0$ with $\tilde{i}\tilde{j} = \arg\min_{ij}\left\{F_{ij}^{(k)}\right\}$.
5) Set $\psi_{\tilde{i}\tilde{j}} = k$ and save this variable in a matrix, $\mathbf{\Psi} = \left\{\psi_{\tilde{i}\tilde{j}}\right\}$ (with the same dimension as **W**).
6) Set $k = k + 1$. If $k < m \cdot N$, go to 3. Else, end.

The book-keeping matrix, $\mathbf{\Psi}$, stores the iteration number at which each connection weight has been deleted: By studying the elements in the columns (or rows) of the matrix it is afterwards easy to deduce when a certain input (or hidden node) has been eliminated. Usually, it the model performance is reported as the mean error, $\varepsilon = \sqrt{F/n}$, where n is the number of observations.

B. Computational Aspects

The computational effort of the method can be reduced considerably by some simple measures. Since the order in which the weights are studied is unimportant, one can go through them in the order of the hidden nodes they refer to (i.e., starting with the connections between the inputs and the first hidden node, etc.). First, the net input to each hidden node, $a_i(t) = \sum_{j=0}^{N} w_{ij} x_j(t)$, i=1,..,m, at each "time instant", t, is determined, as well as the corresponding output, $z_i(t) = \sigma(a_i(t))$. At step 3 of the algorithm (cf. subsection A) a resetting of w_{ik} simply means that the net input of the i^{th} hidden node is changed into $a_i(t) - w_{ik}(t)x_k(t)$, while the net inputs, and outputs, of all other hidden nodes remain unaltered. Thus, for each weight in step 3, only one multiplication and one subtraction is needed to get the net input. In addition, the sigmoidal transformation is required for the hidden node in question. Along with the progress of the algorithm, the computational burden decreases gradually due to the permanent pruning of the weights, and, finally, of the hidden nodes.

C. An Illustrative Example

As a simple illustration of the algorithm, consider a noise-free data set with a ten-dimensional input vector (N = 10) of normally distributed random values with zero mean and unit variance, i.e., $x_1 ... x_{10} = \mathcal{N}(0,1)$ and the dependent variable

$$y = x_2 - 3x_4^2 + 2x_5x_7 \tag{1}$$

Training and test sets of 250 observations each were generated, and the algorithm was run using a network with m = 10 hidden nodes. The upper panel of Fig. 2 illustrates how the errors on the training set (solid line) and test set (dashed line) evolve along with the pruning process that progresses from right to left. The errors are seen to decrease in a step-wise manner, especially strongly as the algorithm enters low network complexities. The lower panel of the figure shows this region in better focus.

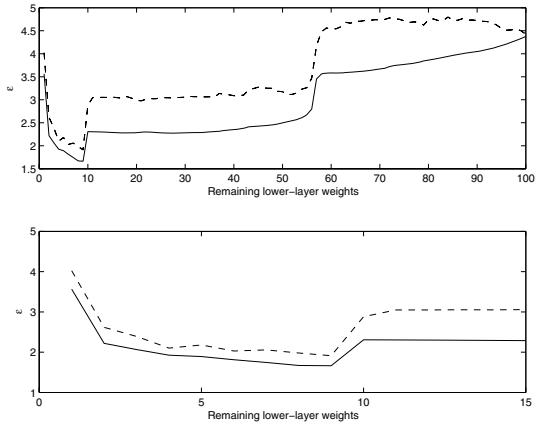

Fig. 2. Training (——) and test (- - -) errors for the example problem as functions of the number of remaining weights (excluding biases) in the lower part of the network.

Potential model candidates are the networks with eight or nine lower-layer connections. Closer study of these show that they have seven hidden nodes and four inputs, x_2, x_4, x_5 and x_7, i.e., only the relevant ones, with the difference that the former network has only one connection to x_2. This network is depicted in Fig. 3 with hidden nodes rearranged for the purpose of illustration: It has a sparse and an intuitively appealing connectivity, where three hidden nodes are devoted to the approximation of the square of x_4 while a joint hidden node is used in the approximation of the product between x_5 and x_7. Such a sparse network lends itself perfectly to a deeper theoretical analysis of how it constructs its approximation and of the nonlinearity of the transformations [10]. For an evaluation of the pruning model on data with noise, the reader is referred to [8].

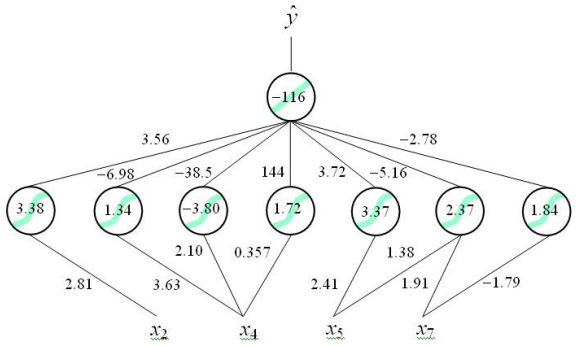

Fig. 3 Schematic of the network with a lower-layer complexity of 8 (cf. Fig. 2). Weights are reported at the connections and biases in the corresponding nodes.

D. Extension to On-line Learning

A shortcoming of nonlinear black-box models is that they may yield poor predictions on independent data if some of the important inputs experience a level change. This is quite common in industrial processes, where the inputs may change rather dramatically because of control actions and changes in raw materials. A remedy follows logically from the over-all approach taken in the algorithm of this paper; it is natural to adjust the upper-layer weights as new information enters, and this is a linear problem, since a given input vector, $\mathbf{x}(t)$, and a fixed set of lower-layer weights yield a fixed output, $\mathbf{z}(t)$, from the hidden nodes for every time instant t. For updating the upper-layer weights, $\mathbf{v}$, the well known (recursive) Kalman filter [11] can be used

$$\mathbf{v}(t+1) = \mathbf{v}(t) + K(t+1)\big(y(t) - \hat{y}(t)\big) \tag{2}$$

where the Kalman gain is given by

$$K(t+1) = \frac{\mathbf{P}(t)\mathbf{z}^T(t+1)}{1 + \mathbf{z}(t+1)\mathbf{P}(t)\mathbf{z}^T(t+1)} \tag{3}$$

while the matrix $\mathbf{P}$ is updated by

$$\mathbf{P}(t+1) = \mathbf{P}(t) + \mathbf{R} - K(t+1)\,\mathbf{z}(t+1)\mathbf{P}(t) \tag{4}$$

In the equations, the covariance matrix of the "measurement error" is $\mathbf{R}$, which, without a priori information, is usually chosen as a diagonal matrix, $c\,\mathbf{I}$. If c is small the gain K is high, which makes large weight changes possible. The initial value of the matrix, $\mathbf{P}(1)$, is usually chosen as a diagonal matrix with large diagonal elements. If the training set is directly followed by the test set, which is often the case in on-line learning, the results can be made less dependent on $\mathbf{P}(1)$ by first applying the filter on the training set, using the final weights and matrices as starting points for the predictions.

III. APPLICATION TO SILICON PREDICTION

A. Background

The blast furnace is the principal unit in the most important process route for iron produced for primary steelmaking. It acts as a large counter current chemical reactor and heat exchanger [12,13]. At its top the main energy source, coke, is charged together with preprocessed ore and fluxes in alternating layers. The ore is heated, reduced and finally smelted by the ascending gases, and intermittently tapped out at the bottom of the furnace in the form of liquid iron (often called hot metal). The reducing gases are formed in the combustion of coke (and auxiliary reductants, e.g., pulverized coal) when preheated, and often oxygen enriched, air (blast) is injected through nozzles (tuyeres) into the lower furnace, Large time delays and sluggish response to control actions make it important to predict quality and operational variables, e.g., the composition of the hot metal. The silicon content of the hot metal is an important indicator of the thermal state of the furnace. In general, it can be said that the silicon content of the hot metal reflects the energy available at (or vertical extent of) the high-temperature region. A decreasing silicon content often indicates a cooling of the furnace that without due countermeasures can lead to serious operational complications, while a high silicon content indicates excessive generation of heat and waste of coke. A high silicon content also unnecessarily increases the amount of slag in the unit processes downstream of the blast furnace. Blast furnaces are usually operated with a safety margin, i.e., a slightly higher coke rate than is deemed necessary, but since the cost of coke is dominating in ironmaking, there are obvious economical benefits of making the safety margin smaller. This requires stricter control of the heat level, and numerous models for the prediction of the hot metal silicon content have therefore been developed [14-25]. In [22] an exhaustive search was made among linear FIR models using a large set of inputs with different time lags and in [24] a partial least squares procedure was applied to select relevant inputs, but in the papers on nonlinear prediction of the silicon content by neural networks a small set of potential inputs was always selected a priori. The reason for this is, obviously, the considerable numerical effort required for training the networks.

B. The Data Set

The present analysis was based on a data set from a Nordic blast furnace, where the variables have been preprocessed to yield hourly mean values. The hot metal silicon content is not generally recorded regularly. Instead, it is measured for every batch (torpedo, ladle or mixer), but in the furnace studied here the analysis was obtained with a frequency of about 1 h^{-1}. On the basis of process knowledge 15 potential inputs were selected. These inputs are reported in Table I: Specific (sp.) quantities are expressed per ton of hot metal (thm). The table also presents the number of order of the variables, their units as well as the symbols that will be referred to later.

TABLE I
POTENTIAL INPUT VARIABLES IN THE MODELING

Number	Variable	Unit	Symbol
1	Total blast volume	m^3n/h	$\dot{V}_{bl}$
2	Blast pressure	bar	p_{bl}
3	Gas permeability	-	κ
4	Sp. coal injection	kg/thm	m'_{coal}
5	O_2 content of blast	%	X_{O2}
6	Gas CO utilization	%	η_{CO}
7	Top gas CO+CO_2	%	Y_{COCO2}
8	Flame temperature	°C	T_{fl}
9	Coke rate	t/h	$\dot{m}_{coke}$
10	Coal rate	t/h	$\dot{m}_{coal}$
11	Energy at tuyeres	MW	E_{tuy}
12	Sp. blast volume	m^3n/thm	V'_{bl}
13	Solution loss rate	kg/thm	$m'_{s\text{-}l}$
14	Tuyere heat loss	MW	Q_{tuy}
15	Ore/Coke ratio	-	o/c

Some of the variables are directly measured, while others are quantities readily computed. In order to evaluate the possibility of on-line prediction of the silicon content, the problem was written as

$$\hat{y}(t) = f(\mathbf{x}(t), \mathbf{x}(t-1), \mathbf{x}(t-2), ..., \mathbf{x}(t-8)) \quad (5)$$

i.e., including lags of the 15-dimensional input vector, **x**, up to eight hours. The dimension of the input vector is thus considerable, 15 × 9 = 135. Note that autoregressive terms were deliberately omitted, since the inclusion of such is known to yield models of high inertia and with small possibilities to predict rapid changes in the output [22]. Data for 800 h of operation was used for training, while the 187-hour period that followed was used for evaluating the performance of the resulting models. Before the models were developed, every (input and output) variable was scaled to the interval (0,1) by dividing the difference between the original value and the minimum value by the range of the variable in the training set. In the following, the model performance is reported in terms of the root mean square error, ε.

C. Overall Behavior

The performance of the algorithm is first illustrated by a run of it from a random starting weight matrix, using a network with five hidden nodes. (An analysis with a higher number of hidden nodes illustrated that similar final performance of the algorithm was obtained.) The Kalman filter was initialized by $\mathbf{P}(1) = 10\mathbf{I}$ (at the start of the training set) while $\mathbf{R} = \mathbf{I}$. Figure 4 illustrates the evolution of the root mean square errors, ε, on the training set (solid lines), test set without (dashed lines) and with (dotted lines) online adaptation of the upper-layer weights. The algorithm, which progresses from right to left in the figure, is seen to initially reduce both the training and test errors (without online learning) considerably, but the latter remains on a considerably higher level throughout the pruning. By contrast, the continuously updated model only shows a slowly decreasing trend, but throughout the run it yields errors comparable with the training errors. Thus, the weight updating results in a model that generalizes well.

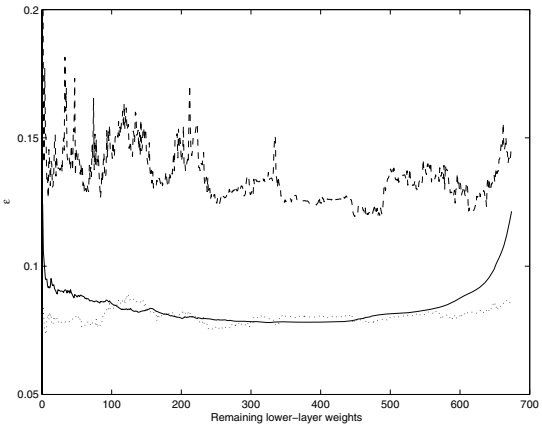

Fig. 4. Training errors (solid lines), test errors without (dashed) and with (dotted lines) weight updating as functions of the number of remaining weights in the lower part of the network.

From the run the lowest error for the on-line-updated version, $\varepsilon = 0.073$, is encountered when only six connections remain in the lower layer of the network. Figure 5 illustrates the approximation provided by this network (dotted lines), the target of the test set (solid line), and the results with fixed upper layer weights (dashed lines). The on-line learning is seen to lead to clearly superior performance: It has not only removed the bias in the prediction but also been able to adapt to show bigger variance in the signal.

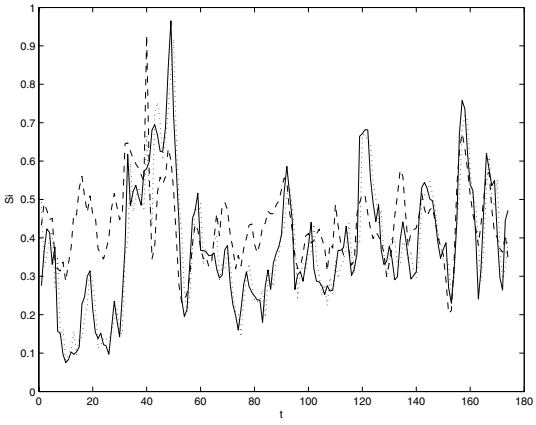

Fig. 5. (Normalized) silicon content (solid lines), prediction without (dashed lines) and with (dotted lines) on-line updating of the weights for the test set.

D. Analysis of a Model Candidate

An analysis of this network reveals that the relevant input variables are 3, 5, 7, 12 and 14, i.e., the gas permeability, the oxygen content of the blast, the sum of the top gas CO and CO_2, the specific blast volume as well as the tuyere heat loss, where the last variable contributes by two inputs (at different lags). In terms of relevance of the variables, with the most important one (i.e., last remaining input) listed first, the model can be written as

$$\hat{\mathrm{Si}}(t) = f(Q_{\mathrm{tuy}}(t-2), V'_{\mathrm{bl}}(t-1), \kappa(t-2), Y_{\mathrm{COCO2}}(t-3), Q_{\mathrm{tuy}}(t-3), X_{\mathrm{O2}}(t-4)) \quad (6)$$

Thus, the heat loss at the tuyeres (lagged by two and three hours), the specific blast volume (lagged by one hour), the gas permeability factor (lagged by two hours), the top gas CO+CO_2 content (lagged by three hours) and the blast oxygen content (lagged by four hours) are considered to be relevant for predicting the silicon content. These variables are mostly related to the tuyere parameters, which are known to affect the silicon transfer in the furnace. The selected variables, as well as their comparatively short time lags (maximum four hours), are findings that agree well with knowledge from the practical operation of the furnace: The residence time of the molten iron between the tuyere level and tapping is in the order of a two hours. The tuyere heat loss is known to reflect the intensity of the thermal conditions in the lower furnace, the specific blast consumption is a measure of the efficiency of the operation of the BF, while the gas permeability reflects the vertical extent of the high-temperature region.

This network has been depicted in Fig. 6, which shows that three hidden nodes are used: One for the specific blast volume, one for the oxygen content of the blast, and one for the two tuyere heat loss terms together with the gas permeability and the top gas CO+CO_2 content. The weights reported to the left of the corresponding connections reveal that the contribution of the last hidden node is minor. This observation is also supported by the fact that the next step of the pruning algorithm eliminates the oxygen content from the input set. Another remark is that all inputs, except the sum of CO and CO_2 in the top gas, exhibit positive correlation with the silicon content. The only negative correlation is understood in that Y_{COCO2} describes the (major) outflow of carbon from the process, and a sudden increase in this means that the fuel reserved is depleted, resulting in an internal cooling and a lower silicon content [26].

E. Detection of Relevant Inputs and Time Lags

In order to study more general features of the problem, a set of runs of the algorithm were undertaken: Starting from 30 random initial weight matrices, the most relevant ten variables, i.e., the ones that remained unpruned until the last ten steps, were recorded.

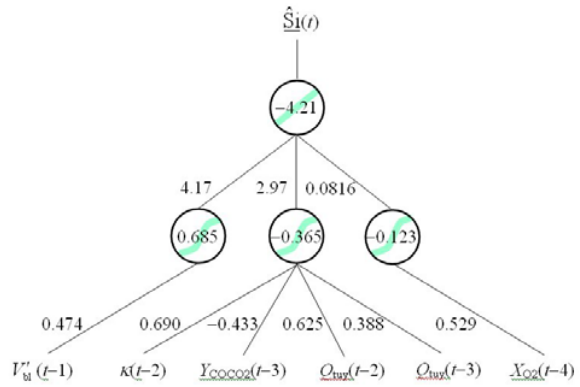

Fig. 6. Schematic of a model candidate for the Si prediction task.

Figure 7 shows the occurrence, P, of the 135 input signals: Each of the 15 different input variables (cf. Table I) has a segment separated by vertical dashed lines, within which the order goes from the largest to the smallest lag. Thus, the first input signal is the blast volume lagged by eight hours, $\dot{V}_{\mathrm{bl}}(t-8)$, while the last signal is the present value of the ore-to-coke ratio, (o/c)(t). From this figure, some interesting observations can be made: Some of the inputs (flame temperature and top gas CO utilization and coke solution-loss rate) are seen to be of little importance, while some show correlation with the output (specific and total coal injection rates) but without a clear preference for a specific lag. As for the most relevant variables, the tuyere heat loss is clearly dominating, followed by the gas permeability and the specific blast volume. All three follow the same pattern further illuminated in Fig. 8; they exhibit a pronounced peak at a time lag of two hours, but also a clear autocorrelation. For the former two, it is striking to note that their present values (i.e., at time t) are considered completely irrelevant — a fact that can be justified by the residence time in the hearth. Studying only the most relevant variable (i.e., the last remaining one at the end of the pruning process), it was found that $Q_{\mathrm{tuy}}(t\text{-}2)$ occurred 12 times, followed by $V'_{\mathrm{bl}}(t-2)$ and $V'_{\mathrm{bl}}(t-3)$ (both with five occurrences) and $Q_{\mathrm{tuy}}(t\text{-}3)$ (three occurrences). Also this shows the strong correlation between the silicon content and the tuyere heat loss and the specific blast volume.

These findings also show agreement with the results of the specific network presented in Fig. 6, and also with the general findings of earlier efforts to predict the hot metal silicon content in other blast furnaces. For instance, in [18] the heat loss was found important, as well as in [19] where a gas permeability index (expressed as the quota between the blast pressure and volume) also was used. In [22] the most important inputs found in an exhaustive search were mainly related to the gas permeability and the energy and heat loss at the tuyeres. Permeability indices were also found to be relevant inputs in the study presented in [24].

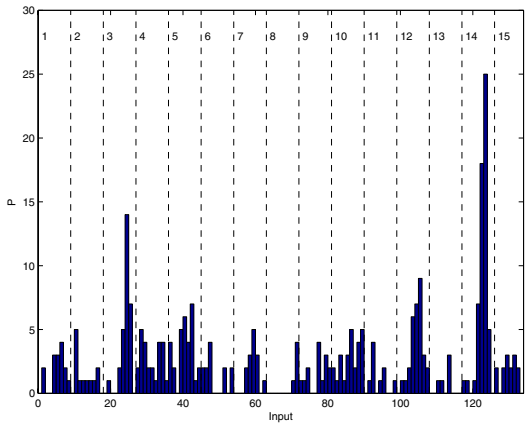

Fig. 7. Occurrence (P) of the different input signals among the ten most relevant variables in the 30 runs.

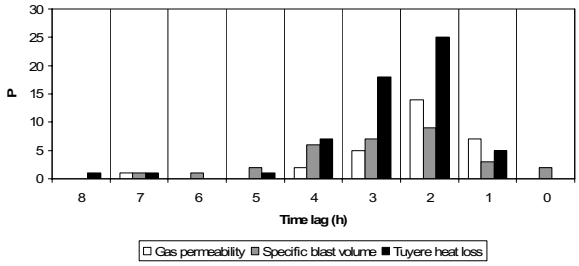

Fig. 8. Occurrence (P) of the time lags of the three most relevant input variables (gas permeability, specific blast volume and tuyere heat loss).

A more detailed statistical analysis of the most promising models was undertaken in order to find a proper lower-layer connectivity of the network in a final model of the silicon content. The connections in all of the last five networks of the 30 runs were analyzed, yielding the occurrences reported in Table II. Quite naturally, the most frequently occurring connections to the hidden nodes are those from a single input, and the top four inputs reported earlier also reappear here, but in a slightly different order: The tuyere heat loss, lagged by 3 h, occurs more frequently than the specific blast volume, lagged by 2 h or 3 h. Also the gas permeability (lagged by two hours) is a frequent variable and occurs more than ten times, but the remaining variables with a single connection appear clearly less often (< 8 times). As for the most frequent connections from two inputs to a hidden node, the occurrences are, obviously, much fewer. Again, the relevant variables detected above are strongly represented (cf. Table II).

F. Final Model

On the basis of this, the following compromise solution is proposed: In order to minimize the number of hidden nodes (and thus the number of parameters in the network), the network illustrated in Fig. 9 is suggested, with a single connection for the clearly most important input, $Q_{\text{tuy}}(t-2)$, and with pair of connections from $\kappa(t-2)$, $V'_{\text{bl}}(t-3)$ and $V'_{\text{bl}}(t-3)$, $Q_{\text{tuy}}(t-3)$. This network has only twelve parameters (including the biases) in total and is also appealing in that it predicts the silicon content two hours ahead.

TABLE II
OCCURRENCES OF COONECTIONS TO INPUTS AMONG THE FIVE LAST INPUTS IN THE 30 RUNS

Variable(s)	Occurrence
$Q_{\text{tuy}}(t-2)$	48
$Q_{\text{tuy}}(t-3)$	32
$V'_{\text{bl}}(t-2)$	21
$V'_{\text{bl}}(t-3)$	18
$\kappa(t-2)$	12
⋮	< 8
$\kappa(t-2)$, $V'_{\text{bl}}(t-3)$	4
$m'_{\text{coal}}(t-3)$, $Q_{\text{tuy}}(t-3)$	3
$X_{\text{O2}}(t-3)$, $V'_{\text{bl}}(t-4)$	3
$\dot{m}_{\text{coke}}(t)$, $Q_{\text{tuy}}(t-2)$	3
$\dot{V}_{\text{bl}}(t-1)$, $Q_{\text{tuy}}(t-2)$	3
$V'_{\text{bl}}(t-3)$, $Q_{\text{tuy}}(t-3)$	3
$\kappa(t-2)$, $Q_{\text{tuy}}(t-3)$	3
⋮	< 3

Training of all parameters of this network to completion using an efficient gradient based method for neural network training [27] resulted in a training error and a test error of 0.093 and 0.131 respectively, while the on-line updated model showed an error of $\varepsilon = 0.074$, i.e., of same accuracy as the best models evolved by the pruning process, as illustrated in Figure 10. The latter model has explained 55 % of the variation in the silicon content in the test set, so it must be considered very competitive compared with models reported in the literature. This demonstrates that a very parsimonious model can be developed on the basis of the findings from the pruning method.

Finally, a comparison between the performance of the pruned network and that of a fully connected network with identical inputs was made in order to evaluate the strength of the pruning algorithm. After optimizing all the weights of the fully connected network with three hidden nodes, a training error of 0.090 and a test error of 0.152 were achieved. Figure 11, which shows the target signal (solid lines) and the predictions (dashed lines) on the test set, clearly illustrates that the network is overtrained, explaining the worse performance on the test set.

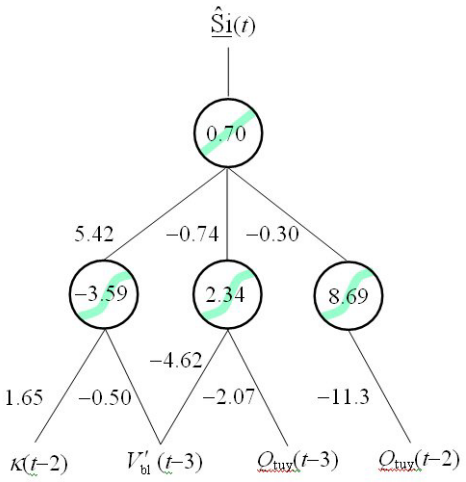

Fig. 9. Schematic of a final model for the silicon prediction task.

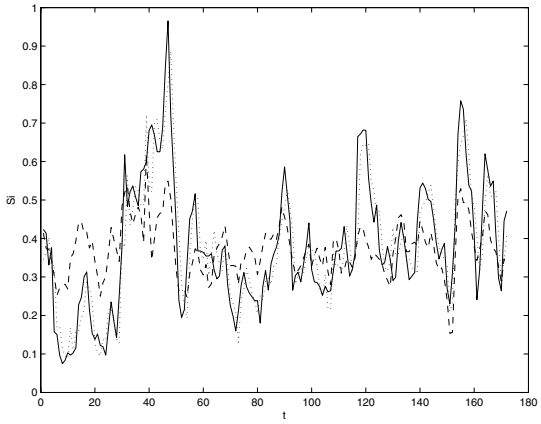

Fig. 10. (Normalized) silicon content (solid lines), prediction without (dashed lines) and with (dotted lines) on-line updating of the weights in the final network run on the test set.

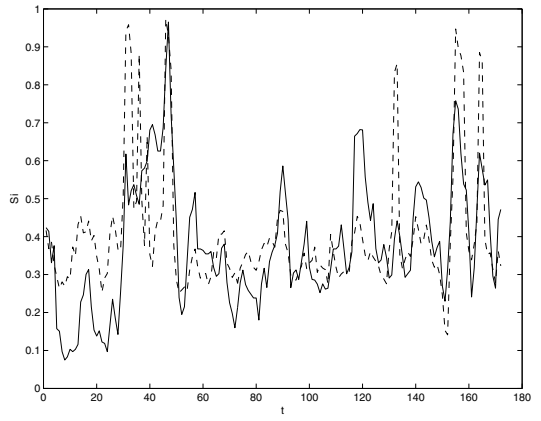

Fig. 11. (Normalized) silicon content (solid lines) and prediction by a fully connected network with three hidden nodes (dashed lines) on the test set.

IV. CONCLUSIONS

This paper has described an algorithm for the selection of input variables, their time lags as well as a proper complexity of a multi-layer feedforward neural network applied on a problem from the metallurgical industry, i.e., the prediction of the silicon content of hot metal produced in a blast furnace. A statistical analysis of the results of several runs of the algorithm demonstrated that the algorithm finds inputs that on the basis of industrial process knowledge are known to correlate with the hot metal silicon content. Furthermore, the time lags found are also reasonable, considering the dynamics of the process.

The merits of an on-line updating of the upper layer weights, appropriate for practical application of the models developed, was further demonstrated. In a final attempt, the relevant inputs and model structure were analyzed by a full gradient-based training of the weights of existing (sparse) connections. This was demonstrated to yield a final model that was able carry out accurate one-step ahead predictions of the silicon content. Furthermore, compared to the results of a fully connected network with identical inputs, the pruned network was found to exhibit clearly better performance on an independent test set.

REFERENCES

[1] G. Cybenko, "Approximations by superpositions of sigmoidal function", *Mathematics of Control, Signals, and Systems,* vol. 2**,** pp. 303-314, 1989.

[2] J. C. Principe, N. R. Euliano and W. C. Lefebvre, *Neural and adaptive systems: Fundamentals through simulations*, New York: John Wiley & Sons, 1999.

[3] M. Frean, "The Upstart Algorithm: A Method for Constructing and Training Feedforward Neural Networks", *Neural Computation*, vol. 2, pp. 198-209, 1991.

[4] S.E. Fahlman and C. Lebiere, "The Cascade-Correlation Learning Architecture", in *Advances in Neural Information Processing Systems*, vol. 2, (Ed. D.S. Touretzky), pp. 524-532, 1990, Morgan Kaufmann.

[5] Y. Le Chun, J. S. Denker and S. A. Solla, "Optimal Brain Damage", in *Advances in Neural Information Processing Systems*, vol. 2 , (Ed. D.S. Touretzky), pp. 598-605, 1990, Morgan Kaufmann.

[6] D.V. Sridhar, E. B. Bartlett and R. C. Seagrave, "Information theoretic subset selection for neural networks", *Computers Chemical & Engineering*, vol 22, pp. 613-626, 1998.

[7] W.S. Sarle, "How to measure importance of inputs", ftp://ftp.sas.com/pub/neural/importance.html, 2000.

[8] H. Saxén and F. Pettersson, "Method for the selection of inputs and structure of feedforward neural networks", *Computers & Chemical Engineering*, vol. 30, pp. 1038-1045, 2006.

[9] K. Hornik, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators", *Neural Networks*, Vol. 2, pp. 359-366, 1989.

[10] J. Hinnelä, H. Saxén and F. Pettersson, "Modeling of the blast furnace burden distribution by evolving neural networks", *Industrial & Engineering Chemistry Research*, vol. 42, pp. 2314-2323, 2003.

[11] S. Haykin, *Kalman filtering and neural networks*, New York: Wiley, 2001.
[12] K. Biswas, *Principles of Blast Furnace Ironmaking*, Brisbane: Cootha Publishing House, 1981.
[13] Y. Omori, *Blast Furnace Phenomena and Modelling*, London: The Iron and Steel Institute of Japan, Elsevier, 1987.
[14] M. S. Phadke and S. M.Wu, "Identification of Multiinput - Multioutput Transfer Function and Noise Model of a Blast Furnace from Closed-Loop Data", *IEEE Transactions of Automatic Control*, vol. 19, pp. 944-951, 1974.
[15] H. Unbehauen and K. Diekmann, "Application of MIMO Identification to a Blast Furnace", *IFAC Identification and System Parameter Estimation*, pp. 180-185, 1982, IFAC, Washington.
[16] Y. C. Chao, C. W. Su and H. P. Huang, "The Adaptive Autoregressive Models for the System Dynamics and Prediction of Blast Furnace", *Chemical Engineering Communications*, vol. 44, pp. 309-330, 1986.
[17] J. Chen, "A Predicting System Based on Combining an Adaptive Predictor and a Knowledge Base as Applied to a Blast Furnace", *Journal of Forecasting*, vol. 12, pp. 93-102, 1993.
[18] H. Saxén, "Short Term Prediction of Silicon Content in Pig Iron", *Canadian Metallurgical Quarterly*, vol. 33, pp. 319-326, 1994.
[19] R. Östermark and H. Saxén, "VARMAX-modelling of Blast furnace process variables", *European Journal of Operational Research,* vol. 89, pp. 85-101, 1996.
[20] G. Zuo, J. Ma and B. Björkman, "A neural network model for predicting the silicon content of the hot metal at No. 2 blast furnace of SSAB Luleå", Proc. *Ironmaking Conference,* vol. 55, pp. 211-21, 1996.
[21] J. Chen, "A Predictive System for Blast Furnaces by Integrating a Neural Network with Qualitative Analysis", *Engineering Applications of Artificial Intelligence*, vol. 14, pp. 77-85, 2001.
[22] M. Waller and H. Saxén, "On the Development of Predictive Models with Applications to a Metallurgical Process", *Industrial & Engineering Chemistry Research*, vol. 39, pp. 982-988, 2000.
[23] M. Waller and H. Saxén, "Application of Nonlinear Time Series Analysis to the Prediction of Silicon Content of Pig Iron", *ISIJ International*, vol. 42, pp. 316-318, 2002.
[24] T. Bhattacharya, "Prediction of silicon content in blast furnace hot metal using Partial Least Squares (PLS)", *ISIJ International*, vol. 45, pp. 1943-1945, 2005.
[25] C.H. Gao and J.X. Qian, "Time-dependent fractal characteristics on time series of silicon content in hot metal of blast furnace", *ISIJ International,* vol. 45, pp. 1269-1271, 2005.
[26] K. Matsuda, N. Nagai, M. Konishi, K. Kadoguchi and T. Yabata, "Forecasting system for decreasing heat levels in blast furnace", *Kobelco Technology Review*, vol. 5, pp. 1-4, 1989.
[27] B. Saxén and H. Saxén, "NNDT - a neural network development tool", *International Conference on Artificial Neural Nets and Genetic Algorithms* (Eds Pearson, D. et al.) pp. 325-328, 1995, Springer-Verlag, Wien.