An adaptive neural network model for predicting the post roughing mill temperature of steel slabs in the reheating furnace

Perttu Laurinen and Juha Röning.

Intelligent Systems Group, Computer Engineering Laboratory, PO BOX 4500, FIN-90014 University of Oulu, Finland.

e-mail: {Perttu.Laurinen, Juha.Roning}@ee.oulu.fi

Abstract

The walking beam furnace and roughing mill of a hot strip mill were studied. A novel control method using measurement data gathered from the production line is proposed. The model uses adaptive neural networks to predict the post roughing mill temperature of steel slabs while the slabs are still in the reheating furnace. It is possible to use this prediction as a feedback value to adjust the furnace parameters for heating the steel slabs more accurately to their pre-set temperatures. More accurate heating enables savings in the heating costs and better treatments at rolling mills. The mean error of the model was 5.6°C, which is good enough for a tentative production line implementation. For 5% of the observations the prediction error was large (>15°C), and these errors are likely to be due to the cooling of the transfer bar following unexpected delay in entry into the roughing mill.

Keywords: adaptive modeling, hot strip mill, walking beam furnace.

Introduction

Steel strips are produced from steel slabs in hot rolling mills. One part of the production process consists of reheating the steel slabs in a slab reheating furnace to a pre-defined temperature, usually between 1200°C and 1300°C. After this heating, steel strips are formed by rolling the slabs. The first mill the steel slab goes through is the roughing mill. After the roughing mill, the temperature of the slabs, now called transfer bars, varies between 1050°C and 1170°C. The transfer bar then goes through a few more procedures, the most important ones being rolling in the finishing mills and cooling. The end product is a steel strip hundreds of meters in length. The strip can still be subjected to further treatments, such as cold rolling. Figure 1 presents these steps schematically.
The two most commonly used types of reheating furnace are the walking beam furnace and the pusher type furnace. In this work, the walking beam furnace was studied. The main difference between these furnace types is the way the slabs move in them. In the pusher type furnace, the slabs are in constant contact with the floor of the furnace and each other. The slabs that are inserted into the furnace push the slabs in front of them forwards. In the walking beam furnace, the slabs move on top of rails that convey them forwards and do not allow them to touch each other.

The walking beam furnace is divided into independently controllable zones. The first zones of the furnace work on the heat produced in the next zones, i.e. the active heating zones where propane or carbon monoxide is burnt to heat the slabs. The last zones, called soaking zones, are used to fine tune the temperatures to the desired goal temperature set for each slab. It is important to adjust the parameters of the zones in such a way that the slabs are heated to the pre-defined temperature as accurately as possible. This will lead to a higher rolling quality of the slabs and to a higher quality of the finished product, the steel strip.

The high operating temperatures of the furnace make the collection of measurement data difficult, and it is downright impossible to collect information about some quantities. One such unreachable quantity is the inner temperature of a steel slab. However, the post roughing mill temperature measurement of the transfer bar’s surface temperature can be used as an estimate of the inner temperature. In this work, a neural network model was developed to predict the mean temperature of the transfer bar based on the initial information of the slabs and on the measurements made inside the furnace while the slabs are still in it. When a prediction of the transfer bar temperature is available while the slabs are still being heated, the accuracy of heating control can be increased. When heating control is more accurate, the temperature deviations of the slabs in the different heating categories decrease and the tuning of the further treatments, including the finishing trains, is easier. The time it takes for the slabs to pass through the furnace possibly also decreases.

In theory, a model based on physics and mechanics could be constructed to predict the post roughing mill temperature instead of the proposed neural network model. Unfortunately, this alternative is far from reality. The construction of a physical model to predict even the simplest process can be very difficult [1-2]. Taking into
account the fact that the operating conditions in the furnace are not constant and that steel slabs with varying metal concentrations are heated in the same furnace, it would be risky and very demanding to construct a physical model. Neural networks, on the other hand, are a suitable tool for process modelling tasks requiring a highly non-linear approach [2]. Other alternatives for neural network models are the finite impulse response (FIR), auto-regressive with exogenous variable (ARX) and moving average with exogeneous variable (ARMAX) models. The disadvantage of these alternatives is that they are not capable of capturing non-linearities equally well as neural networks [1-2].

Our extensive review of the existing applications in this area did not reveal any similar applications. However, there exist quite a few neural network applications in the field of steel production. The research most similar to ours has been done on slab temperature prediction studies by Gorni [3] and Nilsson [4]. The approach proposed by Gorni [3] generated a neural network model from information obtained from thermometers installed inside special slabs, which are run through the furnace. The use of these thermometers is, however, so expensive that not many slabs can be run through the furnace, and excessive observations are not economically feasible. It may also be a false assumption that the data gathered with these few slabs would be representative. Moreover, the approach hardly allows for changes in the environment, because these special slabs cannot be run through the furnace continuously. Nilsson [4] predicts the same post roughing mill surface temperature of transfer bars in her paper, using neural networks as we do. The purpose of her models is to use prediction to set the parameters of the mills, which is different from ours. The model is based on a data set available only after the slab has exited the furnace, including surface temperature and other measurements, and hence it cannot be used to control the heating of the slabs while they are inside the furnace, which is done in our model. Her results are, however, comparable to ours at some level, since they predict the same temperature.

The model presented in this paper is based on sensor data gathered on-line from the furnace and stored in a production database. The data used by the model can be acquired from the database at the production line, and the use of the model does not require the installation of any additional instrumentation. The model can be run as part of the existing plant data system, which makes it affordable.

Other neural network applications at hot strip mills include an application controlling the heating furnaces [5] and applications developed for predicting the rolling forces at finishing mills [6]. Applications for predicting the temperature after the last finishing stand based on information measured from the finishing mills and the transfer bar have been developed in [7] and applications controlling the shape of the strips in [8-9]. More information about the state of the art of artificial intelligent applications in hot rolling processes can be found from review papers [10-12].
**Data description and pre-processing**

The data used in the work consists of two data sets measured from a hot strip mill. The first one was used to test a prototype and the second to test the software implementation of the model. The data set for the prototype was collected in the summer of 1999 and consists of observations from 3200 steel slabs. The data set used in software development consists of observations on 200 slabs recorded in the summer of 2001. Some pre-processing was needed to make the data usable with the modelling tools.

In general, the pre-processing of data is divided in two subtasks: 1) feature or variable transformations and 2) subset selection [13]. Feature transformation consists of such tasks as forming new variables out of the existing ones and combining data from different sources. Subset selection consists of selecting the variables to be analysed and the data set to be used. The method of selecting variables depends on the preliminary knowledge of the phenomenon, the amount of data and the number of variables, and the ratio between the amount of data and the number of variables [14].

The preliminary variables used in the model were selected based on expert information. In the later modelling phases, the set of variables was reduced even further. The total number of variables relevant to this application is about 150-200. This number was minimised by carefully selecting the most important variables from the production database together with an expert. Finally, there were still about 50 variables left. To gain more understanding of the relationships between the different variables, the interactions were studied using a Bayesian network. First, the connections of the network were identified using automatic search algorithms that work based on the measured data. Then, the final model was formed by an expert based on the automatically generated model. Figure 2 shows the constructed network. The nodes in the figure stand for the different variables, while the arrows indicate the interactions between them. The variables were also grouped into larger units to portray higher level abstractions, such as the slab dimensions. The connections between these abstractions show how the abstractions interact. The network was of special use for us researchers, who are not experts on the production process, but rather on modelling. Further details about the work with the Bayesian network can be found from [15].
The number of data points was reduced by including only the soaking zones in the model, since the opinion of an expert was that this is a good starting point for producing a working model. Moreover, slabs with post roughing mill temperatures less than 1100°C were excluded, because these slabs may have been standing on the roller table too long before roughing. This waiting time cannot be anticipated while the slabs are in the furnace. 20% of the slabs fell into this category.

Validation and scaling of the measurements are part of the process of variable transformation. The validation of data is especially important in methods applied to on-line process data. If ignored, a missing value may halt the functioning of the whole system. To prevent this problem, the system checks whether the measurement is within the range set for the variable. If not, it is replaced by the lower limit of the variable if the measured value is lower than the lower limit or with the upper limit value if the measured value is higher than the upper limit. If the measurement is completely missing, it is replaced by the median of the range. There are also more elegant methods for replacing missing values, including methods based on conditioning the missing value on the existing values [16]. Finally, the input and target variables of the neural network were scaled to the range from -1 to 1, which can make the training of the network faster and help in initialising the weights [17].

**Methods**

Feedforward type neural networks were used to predict the post-roughing mill temperatures. They are a method resembling statistical regression and their strength lies in their abilities to capture non-linearities in the underlying data.

The neural network society has borrowed a lot of terminology from biology, because the early work in this area was inspired by the mathematical equivalent of a biological phenomenon. A feedforward neural network consists of data processing units called neurons connected with weights. The neurons are organised into layers called the input, hidden and output layers. The data is fed into the input layer, further transformations are done in the hidden layer(s), and the result of the transformation is
read from the output layer. The number of hidden layers and neurons defines how well the network can adapt to the data. More detailed descriptions of the functioning of a feedforward network can be found from various textbooks, such as [18] and [19].

The weights of the network are estimated from the data. The performance of the estimation technique determines how well the weights capture the behaviour of the phenomena under study. The techniques are divided into two general categories: deterministic and stochastic methods. Deterministic methods have been designed to find a local minimum from among the initial settings of weights in the network. The most widely known deterministic method is the back-propagation of error using derivatives calculated according to the error of the network to tune the weights to optimal values. More sophisticated methods include the conjugate gradient technique. Stochastic algorithms have been designed to find a global minimum from the search space. They incorporate a form of randomness, allowing the optimised parameters, i.e. the weights, to change into new values that may be quite dissimilar from the previous values. This leads ultimately to the most optimal settings for the weights, but may require considerable computing time. Examples of stochastic algorithms include simulated annealing [20] and genetic algorithms [19].

The method of estimating the weights in this work was a compromise between the fast deterministic algorithms and the computationally intensive stochastic algorithms. We used a mixture of these two methods called hybrid learning. In hybrid learning, a stochastic algorithm is used to estimate a good global starting point, and after that, the deterministic algorithm is used to estimate the parameters more accurately using the starting point.

The estimated weights of the neural network can be kept constant or they can be re-estimated within suitable periods. Models where the weights are re-estimated as the data changes are adaptive. The two basic ways of estimating the weights adaptively are batch and on-line learning. In batch learning, more than one observation are collected and the weights are re-estimated using this data set. In on-line learning, the weights are updated after every new observation. The advantage of using adaptive estimation is obvious: the parameters of the model are kept up to date.

A completely different question is the question of when an adaptive approach is needed. Applications have shown that an adaptive model can outperform a static model, but the decision to use adaptive modelling must be made based on the phenomena under study. If the operating environment is likely not to remain constant or it is too difficult to collect a representative data set, then the use of adaptive modelling is justified. An example is a model predicting the temperature of steel strips after the finishing stands on a hot strip mill [12]. An adaptive neural network achieved a mean prediction error of 1°C. When re-training was discontinued and the static version was adopted into use, the error grew to 8°C. The cost of adaptivity is the more complex implementation of the models because of the larger number of parameters to tune. It must be decided when the parameters are re-estimated and how much data is used for the estimation, and there might also be constraints on the time available for re-estimating the parameters.
Model and results

In order to use the feed forward neural network model with time series data, a windowing function for the observations was used. The observations within the time window were then used to train the network parameters and to predict the temperature of the slabs in the soaking zones of the furnace. The developed prototype and the software implementation were using somewhat different time windows, as will be described later.

The actual modelling work was started with a prototype made with Matlab and its neural network toolbox. The neural network was selected to include 29 input variables and one hidden layer with 12 neurons with tanh activation functions. The input variables were selected at this stage after experimentation with different input variable sets. They were the same variables as shown in Figure 2, except that the material percentages and the unconnected variables were left out. A deterministic training algorithm (the conjugate gradient method) was used to estimate the network parameters. The time window that makes up the training data was selected to contain information from so many slabs that it contained at least 150 data points, which meant information from about 15 slabs. The network parameters were re-estimated every time a slab went through the roughing mill. The predictions were filtered using a mean filter. For a single slab, many observations were available in the soaking zones, and hence also many predictions were given. The cumulative average of these predictions was used as the predicted temperature, meaning that the first prediction was used as such, while the second prediction was the mean of the first and second predictions and so on.

The performance of the model was estimated by calculating different error statistics for the predictions. The statistics are calculated from the absolute values of the last prediction errors of the slabs before they exit the furnace. The mean error was 8.0°C and the median error 5.6°C. The median error was considerably smaller, because the prediction error for some slabs was large, and this increased the mean. The percentage statistics showed the proportion of predictions that were closer than the respective temperature. Predictions within 5°C (47% of the slabs) can be considered very good, predictions within 10°C (73% of the slabs) good and predictions outside the range of 15°C (14% of the slabs) poor. The product expert considered the prediction accuracy of the prototype good, since it is sufficient for setting up additional treatments of the products, and further studies including pilot implementations of the model should be made. Finally, the Figures 3 and 4 show examples of the predicted (dashed line) and measured (continuous line) values.

Figure 3: The most erroneous data set used with the prototype. The vertical axis shows the post roughing mill temperature in Celsius and the horizontal axis the number of record in the data set. The slabs are in a chronological order.
After successful implementation of the prototype, software implementation of the model was started. Transferring the Matlab model on to the production line was not considered, because it would have been hard to fit it together with the information system, and the robustness of the solution would hence have been questionable. A feedforward type neural network library with hybrid training algorithms was written, and an environment where data from the production line database could be tested offline was developed.

The structure and the parameters used by the training algorithm of the network were tested more extensively at this stage. Since there is no proven theory for constructing the network structure, a semi-automated empirical approach was used. The approach is based on testing a large number of neural network model candidates generated by using a set of rules. The parameters chosen in this way were the input variables, the number of hidden layers, the number of neurons in each hidden layer and the parameters of the training algorithm.

All of the tested models included a certain set of input variables. In addition to this set, every model contained input variables selected uniform randomly from a set of candidates. After selecting the input variables, the number of hidden layers was selected, and either one or two hidden layers were used. Then, the number of neurons in the hidden layers was defined. The first hidden layer was randomly selected to contain a number of neurons from the range of $0.33n - 1.2n$, where $n$ is the number of inputs. If the network also contained a second hidden layer, it was selected to contain a number of neurons in accordance with the first hidden layer in the same manner. Figure 5 presents the structure of the final neural network model with the input variables.

Finally, threshold rules were used to decide when to retrain the network and when to stop the training. It was not necessary to retrain the network unless prediction accuracy decreased below a certain value. Because of this, the weights were re-estimated after the average absolute prediction error of five consecutive slabs exceeded 9°C. These values were found out by testing different rules for updating the weights. The training of the network was completed after the training algorithm had applied the deterministic training algorithm to a certain number of starting points initialised using the stochastic algorithm. The number of starting points varied from model to model, as the models and training parameters were randomly generated. After finding suitable training parameters, it was not necessary to use cross-validation or other techniques for studying the training error, because the large number of tested models (>2000) ensured sufficient parameter quality of the best performing models.
Parameters resulting in overfitting or overgeneralisation of the model would result in poor prediction accuracy and an inadequate model.

![Diagram of neural network structure](image)

**Figure 5:** The structure of the neural network with the input variables.

The models were calculated on multiple clients connected to a database containing pointers to uncalculated models. After a client finished its calculations, the prediction statistics for the calculated model were entered back into the database. The best performing models were then selected from the database, and the results were analysed in more detail.

Figure 6 shows the predictions of the neural network plotted against the measured values. In a perfectly performing model, the dots in the Figure should lie in the diagonal line. The two outer lines are 15°C away from the theoretical optimum, representing the limits of acceptable deviation. As it can be seen from Figure 6 a), most of the predictions are within the 15°C limits, but some clearly exceed them. These points outside the limits are mostly predictions from single slabs. Figure 6 b) presents the predictions for the last observations on the slabs before they exit the furnace.

Cumulative median filtering of the predictions of each slab was applied to find out if it would help to bring the predictions outside the limits closer to the optimum. The Figures 6 c) and 6 d) show these points plotted in the same manner as in the Figures 6 a) and 6 b). The median filtering brought the predictions of each slab clearly closer to each other and removed most of the large errors in the lower part of the plot. Figure 6 c) shows that the points outside the limits are grouped together and present an observation on one slab. It is likely that the errors originate from a slab that has remained on the roller table for an abnormally long time after exiting the furnace and has therefore lost some of its temperature before entering the roughing mill. The time the slabs stay on the roller table cannot be anticipated while they are heated, but the time is usually standard. In the operation on the production line, the model is informed of the delays, and the effect can hence be controlled. At the time of this study, the information was not available off-line.
Figure 6: The measured values of the post roughing mill temperature have been plotted on the x-axis and the predicted values on the y-axis, and some random noise has been added to the measurements to distinguish the points more clearly. The plots a) and b) are plots for the neural network prediction of the best performing model. In the plots c) and d), median filtering has been applied to the neural network prediction. The Figures a) and c) contain all the observations from the soaking zones, while the Figures b) and d) only contain the last observation from each slab.

Table 1 shows the same statistics for the prediction results as were calculated for the prototype, the only difference being that these statistics were also calculated for predictions from all of the observations instead of merely the last values. The average of the absolute error values of the unfiltered neural network model was 5.9°C and that for the filtered model 5.6°C. The median error was 4.5°C for the neural network predictions and 4.0°C for the filtered values.
The statistics calculated from the last observations of the slabs are comparable to the prototype statistics. Moreover, they are comparable at some level (as explained in the Introduction) to the work done by Nilsson [4], though the purpose and motivation of this work are different from hers. The prediction accuracy, i.e. the root mean squared (RMS) error reported by Nilsson [4] was 13. The RMS for our neural network prediction was 7.9 and that for the median-filtered prediction 7.5. For the predictions of the last observations, the respective values were 8.5 and 7.3. The benefit of our model in comparison to hers is that it does not use any data recorded after the slabs exit the furnace. It should be noted, however, that the data set and the process used are different, which makes the results mutually quite disparate.

<table>
<thead>
<tr>
<th></th>
<th>Normal</th>
<th>Median-filtered</th>
<th>Last observation</th>
<th>Last observation, median-filtered</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Mean error</strong></td>
<td>5.9°C</td>
<td>5.6°C</td>
<td>6.3°C</td>
<td>5.5°C</td>
</tr>
<tr>
<td><strong>Median error</strong></td>
<td>4.5°C</td>
<td>4.0°C</td>
<td>4.9°C</td>
<td>3.9°C</td>
</tr>
<tr>
<td><strong>Standard deviation</strong></td>
<td>5.2°C</td>
<td>5.0°C</td>
<td>5.7°C</td>
<td>4.9°C</td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td>7.9</td>
<td>7.5</td>
<td>8.5</td>
<td>7.3</td>
</tr>
<tr>
<td><strong>&lt;5°C</strong></td>
<td>54%</td>
<td>55%</td>
<td>52%</td>
<td>56%</td>
</tr>
<tr>
<td><strong>&lt;10°C</strong></td>
<td>80%</td>
<td>80%</td>
<td>80%</td>
<td>79%</td>
</tr>
<tr>
<td><strong>&gt;15°C</strong></td>
<td>6%</td>
<td>5%</td>
<td>9%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Table 1: Statistics for the prediction error of the software implementation of the neural network model predicting the post roughing mill temperature. The column titled “normal” shows the statistics for the neural network prediction, the column “median-filtered” shows the corresponding statistics for the median-filtered prediction, and the last two columns show the same statistics for the last observations of the slabs.

The prediction accuracy of the model is adequate for tentative implementation at the production level. The large prediction errors in 5% of the predictions are problematic, however, and studies are being made to eliminate this problem using prior information, for example.

**Conclusion**

A method for predicting the post roughing mill temperature of transfer bars was developed. The prediction was done using neural networks while the slabs were heated in the reheating furnace. A Bayesian network was first used to visualise and clarify the interactions between the various variables affecting the heating process. After that, an adaptive neural network model was applied to the data, with a mean prediction error of 5.6°C. The results were accurate enough for tentative application of the model on the production line. The main problem was the larger prediction errors with 5% of the slabs, which can hopefully be corrected when data from the time it takes for the transfer bars to enter the roughing mill after exiting the furnace becomes available.
Acknowledgements

The authors would like to express their gratitude to the supporters of this work, Rautaruukki Steel and the Finnish Academy of Sciences.

References


