Hybrid Hot Strip Rolling Force Prediction using a Bayesian Trained Artificial Neural Network and Analytical Models

[1]Abdelkrim Moussaoui, [1]Yacine Selaimia and [2]Hadj Ahmed Abbassi
[1]Department of Electronics, University of Guelma, BP 401, 24000, Algeria
[2]Department of electronics, University of Annaba, BP 12, Annaba, 23000, Algeria

Abstract: The authors discuss the combination of an Artificial Neural Network (ANN) with analytical models to improve the performance of the prediction model of finishing rolling force in hot strip rolling mill process. The suggested model was implemented using Bayesian Evidence based training algorithm. It was found that the Bayesian Evidence based approach provided a superior and smoother fit to the real rolling mill data. Completely independent set of real rolling data were used to evaluate the capacity of the fitted ANN model to predict the unseen regions of data. As a result, test rolls obtained by the suggested hybrid model have shown high prediction quality comparatively to the usual empirical prediction models.

Key words: Hot rolling mill, neural networks modelling, bayesian evidence

INTRODUCTION

Recent years have seen attempts by a number of authors to utilise various artificial neural network (ANN) based models to help better predict the rolling force in a hot rolling strip mill[1-3]. The rationale here is that ANNs have the potential to provide a mechanism for dealing with multi-variate, often noisy and possibly non-linear data sets, where an exact analytic model is either intractable, or too time consuming to develop. The basic procedure is to use a database of measurements to train an ANN structure and then evaluate the predictive capacity of the built model on previously unseen data[4].

Traditional back-propagation based training of feed-forward ANNs is accomplished by a process of minimization of an error function which quantifies the network output performance (whether regression or classification) in terms of the difference between the network predicted values and true target values[5] applied to some set of training data. The basic approach of direct gradient descent has been much improved in recent years by the development of more sophisticated search algorithms such as scaled conjugate gradients which provide a significantly faster convergence. Assuming that satisfactory network convergence has been obtained from the training data, probably the next largest issue lies in the ability of the trained network to provide satisfactory classification performance on previously unseen data. The generalization capacity of a network is a measure of the networks capacity to learn the underlying structure of the data rather than any noise present in the data. For the example of a regression model, a network with good generalization will tend to produce similar error rates for both the training and (independent but drawn from the same distribution) testing data, whereas a poorly generalizing network would give very low errors on the training set and relatively poor performance on the test data. Such a network is said to be over-fitted. Generalization can be understood by considering the bias and variance of the network model[6]. There is a trade-off between bias and variance and a function too closely fitted to the training data (a large network with many parameters) will have a large variance and hence generalize poorly to new data. By smoothing the function (a simpler network with fewer parameters), the generalization improves, but if taken too far will yield a model with insufficient complexity to model the data, yielding a high bias and hence large error.

The key to good classifier performance is to find the network with the best generalization performance to new data and this requires careful consideration of the quantity and quality of the training data (it should be statistically well sampled from the generative distribution) and the relation to the overall size of the network[7]. The problem is likely to be exacerbated for large networks (with lots of independent parameters) and limited amounts of training data[8].

An often cited empirical requirement for classification problems is that the minimum number of training patterns N_{MIN} required for good generalization performance is given by[5, 6]:

$$N_{MIN} = W / \varepsilon \qquad (1)$$

Where W is the total number of independent network weights and biases and ε is the fraction of classification errors permitted on the test data.

Practically the use of early stopping or cross-validation using an independent validation data set are used as termination or selection criteria for network training[6]; the final network performance being evaluated using a third independent test data set. Good

---

Corresponding Author: Abdelkrim Moussaoui, Department of Electronics, University of Guelma, BP 401, 24000, Algeria

generalization performance is an indicator that the information capacity of the network (reflected in the number of weights) is of the same order of, or smaller than the total information content of the training set. The rationale here is that if the network is sufficiently complex, it can memorize all the features of the data (including noise). For the network to generalise it has to start to just store the important features of the training data. This being stated, there also exists evidence to suggest that generalisation performance depends more on the size of the weights rather than their number.

Recent years have seen the development of sophisticated methods of addressing the over-fitting problem through regularisation techniques. A simple way to implement regularization is using an additive weight decay term in the error function[9]. A more sophisticated basis for regularization can be found in the approach of Bayesian Evidence update techniques for network training which frame the optimisation problem rather differently.

Such an approach is useful as network weight regularization falls naturally into the framework; and additionally it is possible to estimate confidence bounds on the output predictions based on the widths of the posterior probability functions for the weight matrix[10].

In the current work the authors seek to use a Bayesian implementation of ANN modelling and physical prior knowledge to generate an efficient hybrid hot rolling force prediction model.

Hot strip rolling mill description: A simplified schematic diagram of a steel rolling mill for the production of coil plate is presented in Fig. 1. It shows the transformation stages of slabs from entry at the reheat furnace to their exit at the coiler at the end of the mill.

The feed stock for the rolling mill are slabs produced by the continuous casting process in a steel plant (1). These are normally supplied at ambient temperature. The purpose of the reheat furnace (2) is to raise the temperature of the whole slab to the around 1250 °C (.re-crystallization temperature).

On exit from the reheat furnace, there is a build-up of scale on the surface of the slab, due to oxidation, which is detrimental to surface quality. This is removed within the de-scaling box (3), which consists of jets of high pressure water (140 bars).

After the de-scaling stage, the roughing mill (4) produces a breakdown bar (the product between the roughing mill and the finishing mill) by rolling the slab through a series of forward and reverse passes, typically reducing the slab thickness from 200 to 30 mm. The finishing mill (5) is designed to reduce the gauge (thickness) of the breakdown bar to that of the finished coil, while maintaining the desired width. The finishing mill control system is critical as constant mass flow

Table 1: Main characteristics of mittal steel hot plate mill, El-Hadjar, Algeria

Finisher mill Type	6 stand 4-High mill equipped with hydraulic screwdowns
Nominal Power	2 x 3000 HP, D.C.
Rolling Thickness Range	1.5 to 15 mm
Strip Width Range	600 to 1350 mm
AGC	Hydraulic
Gauge meter	X Ray
Rolling speed	49 to 407 rpm
Rolling temperature	860 to 880 °C
Maximum rolling Force	30000 KN
Maximum exit speed	15 m/sec

must be maintained in all stands to ensure continuous production[11].

On exit from the finishing mill, the rolled strip is still at elevated temperatures, typically ($> 800°C$), which is above the phase transformation of the coil. Critical quality parameters, such as the mechanical properties and other metallurgical properties, of the finished coil are significantly affected by the cooling process applied in the run-out table (6). On exit from the mill / run-out table cooling system, the hot strip typically has a velocity of up to 15 m/s and can be hundreds of metres in length. The down coiler (7) allows the strip to be converted into a coil of dimensions that can be easily transported. The main characteristics of the considered hot strip rolling mill are given in Table 1[12].

Hot strip rolling force empirical model: For a given strip Temperature, a steel grade and a rolling speed, the actual separating rolling force can be calculated by the model of Alexander-Ford [13-15]:

$$F_a = 1,15 \; b \; k_m \sqrt{R(h_e - h_a)} \; Q_p \qquad (2)$$

where b is the plate width, R is the work roll radius, k_m is the mean constrained flow stress in the roll bite for plane strain conditions (strain resistance), expression under square root is the contact length, h_e is the incoming thickness, h_a is the exit thickness and Q_p is a geometric factor which is strongly affected by the geometry of roll bite and interface conditions between the rolls and rolled strip.

The term k_m can be calculated in function of a set of hot rolling parameters, like plate temperature T, viscous friction coefficient μ and rolling speed v.

In practice, there is no accurate physical model which describes efficiently the relationship between these hot rolling parameters. They are generally determined by empirical regressions. In fact, many authors proposed different equations for the calculation of the geometrical factor Q_p, based on their own experimental rolling data.

An analog situation stands for the hot flow strength k_m, where there are many formulas that permit its evaluation from values of temperature, strain and strain rate[16].

A commonly used approximation of the term k_m is given by Misaka formula[17,18]:

$$k_m = \exp\left[\begin{array}{c}0.126 - 1.75C + 0.594C^2\\ +\dfrac{\left(2851 + 2968C - 1120C^2\right)}{T}\end{array}\right]$$ (3)

$$k_r^{0.21}\left(\frac{dk_r}{dt}\right)^{0.13}$$

where:

k_m: Steel Hot Strength [kgf/mm²],

C: Carbon content [%],

T: Absolute Temperature [°K],

k_r: True Strain,

t: Time [s].

To improve the hot rolling force model accuracy, we suggest to combine forward neural networks and empirical knowledges which gives an hybrid prediction model.

Hybrid ANN prediction model Topology : As shown in Fig. 2, the chosen network structure for the regression modelling was a multilayer perceptron (MLP). The reason for this choice was that MLPs have been shown to be universal approximators[19]. The MLP network implementation and training was undertaken in MATLAB^{TM}.

The data was presented to a series of MLP networks with a variable number of hidden nodes arranged in a single hidden layer.

Each network had 6 input nodes corresponding to the variables used in equation 2 and a single output node corresponding to the value of the predicted rolling force F_a.

The empirical model given in equation (3) is associated to the structure of the ANN force prediction model to evaluate the value of the strain resistance k_m in terms of strip temperature and other metallurgical parameters which gives an hybrid hot rolling force prediction model.

The output rolling force F_a, from the second (hidden) layer was given by:

$$F_a = \sum_{j=1}^{M} w_{kj} \tanh\left[\sum_{i=1}^{d} w_{ji} x_i + b_j\right] + b_k$$ (4)

Where w_{ji} was the weight matrix of the first layer, w_{kj} the weight matrix of the second layer, b_j the bias vector of the first layer, b_k the bias vector of the second layer, d the number of input nodes and M was the number of hidden nodes.

The non-linear capability of the network was implemented using the *tanh* transfer function between the first and second layers. The Bayesian Evidence based training is investigated.

When implementing Bayesian Evidence training[10], the error function was given by:

$$E_{BE} = \frac{\beta}{2}\sum_{n=1}^{N}(t_n - y_n)^2 + \frac{\alpha}{2}\sum_{i=1}^{W} w_i^2$$ (5)
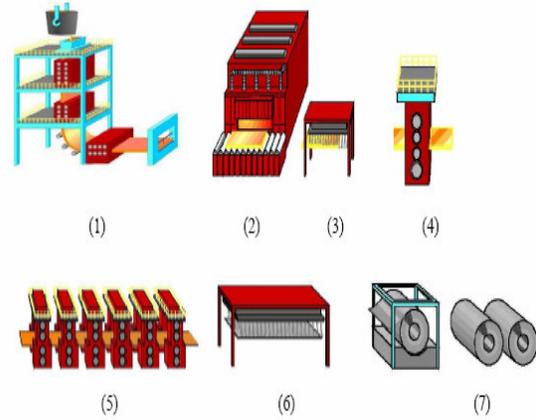
Fig. 1: Hot rolling line layout
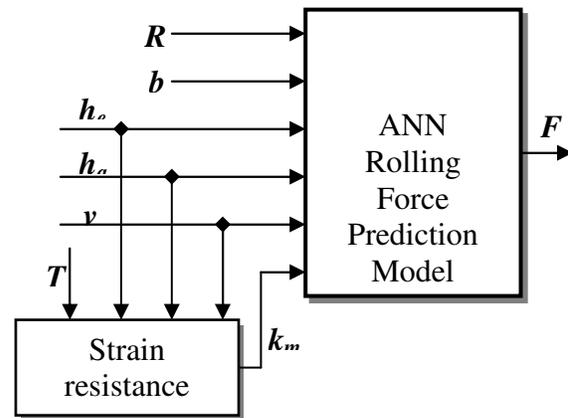
Fig. 2: Hybrid ANN prediction model topology
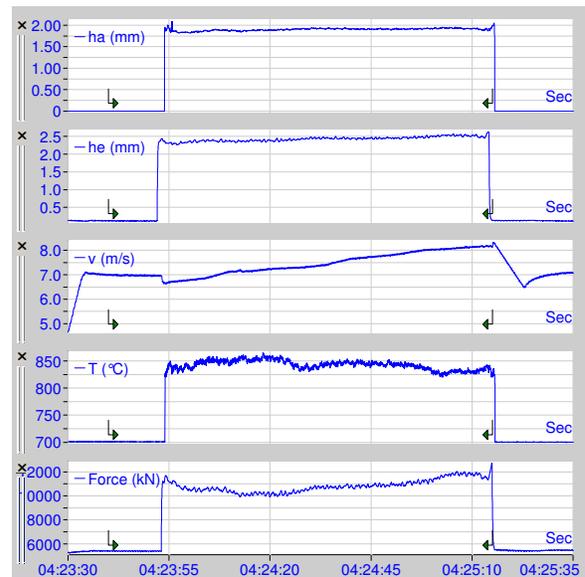
Fig. 3: Hot rolling input/output samples

Where E_{BE} was the (Bayesian Evidence) error function and β the parameter describing the inverse variance of the noise model for the target data (predicted hot rolling force).

If interest lies only in minimizing the error for a particular weight vector, then the effective value of the regularization parameter depends only on the ratio α/β.

Besides accommodating regularization in a consistent framework, the Bayesian approach has the additional advantage of providing a mechanism to generate confidence bounds on the output prediction values.

Assuming that the posterior distribution of the weight matrix is Gaussian in nature, it is possible to find the variance corresponding to the mean output $y(x,w_{MP})$, i.e. the standard prediction output for the most probable weight distribution. This variance is given by[6]:

$$\sigma^2 = \frac{1}{\beta} + g^T A^{-1} g \qquad (6)$$

Where A was the Hessian matrix defining the second derivatives of the error function and g was the gradient of the error function.

The standard deviation, σ of the predictive distribution can be interpreted as an error bar on the mean value y_{MP} which has two contributions. The first arises from the intrinsic noise in the target data, the second from the posterior distribution of the network weights. The ease of implementation of these powerful network training paradigms was a major consideration in employing the NETLAB toolbox to realize the network training[10].

For the Bayesian Evidence training an initial value of α=0.01 was employed along with an initial inverse noise variance parameter $\beta = 100$.

During the Evidence update procedure of the network training, these hyper-parameters were re-evaluated iteratively.

Data acquisition and pre-processing: Measurements of several finishing rolling mill variables were recorded for each coil rolled over a five days period of production using IbaAnalyser© , (an embedded real-time logging software). For our study, the manufacture of a single grade of steel coils (with constant width and exit thickness varying from 1.2 mm to 4.00 mm) was considered.

The patterns were grouped into training, validation and test sets. The output values of F_a ranged from 0 to 1.

In order to fully utilise the maximum dynamic range of the *tanh* transfer function of the ANN networks, The input values were normalised to lie in the range -1 to + 1 before presentation to the networks. Note that all results presented in this paper are plotted as the original (un-normalised) data.

A sample of the most significant input/output variables used in our study for one rolling coil is shown in the curves of Fig. 3.
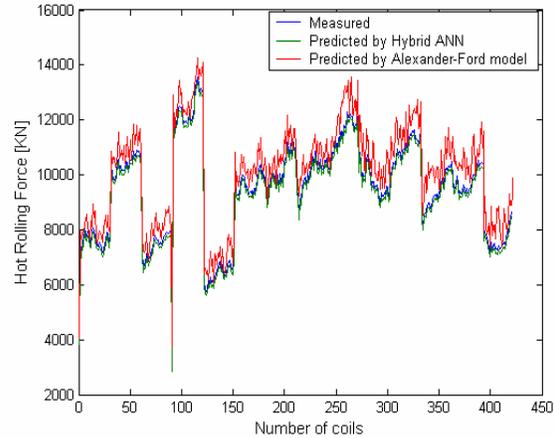
Fig. 4:   Predicted and measured hot Rolling forces

RESULTS

Bayesian Evidence training algorithms was implemented in MATLAB using the NETLAB toolbox. The performance of the network was evaluated by calculating the mean square error of the true target values from the network predictions.

$$MSE = \frac{1}{N\sigma_t^2} \sum_{n=1}^{N} (t_n - y_n)^2 \qquad (7)$$

Where y_n are the network outputs, t_n the target values, N the number of samples and σ_t, the variance of the target data.

For each of the 1 to 15 hidden node network structures, the error was calculated for the training, validation and test data sets. For Bayesian Evidence training, the best selected network in term of generalization of prediction was for 8 hidden nodes. The performance of the selected hybrid ANN model is tested using measured rolling forces at the last stand of the finishing mill from 422 coils. From Fig. 4, it is shown that the hybrid ANN model is able to predict the rolling force in more accurate way than the analytical model given by equation 2.

CONCLUSION

The efficiency of a hot strip mill can be increased if the amount of rejected material is reduced. A strip is considered as rejected material if it does not meet the requirements of the customer and thus has to be sold as lower quality or has to be re-melted. This last option implies a tremendous amount of extra materials handling and energy costs.

One way to improve the efficiency of the hot rolling mill is the use of a better finishing set-up model for hot rolling force prediction.

Bayesian training method was implemented to construct a series of hybrid ANN structures to model hot rolling force prediction from real input/output data and empirical expressions. It has been demonstrated

that the suggested approach produced a superior fit to the data.

This was experimentally verified with a completely independent set of data. It was noted that the Bayesian training algorithm tended to produce smoother overall fitting functions to the training data. A smoothly varying model output was an important characteristic to raise confidence in hot rolling force prediction model. Future work will aim to extend the number of input parameters modelled to allow for different carbon content of steel.

ACKNOWLEDGMENT

REFERENCES

1. Hwu, Y.J. and J.G. Lenard, 1996. Application of neural networks in the prediction of roll force in hot rolling. Proc. 37th Mechanical Working and Steel Processing Conf., The Iron and Steel Society, Warrendale, pp: 549-554.
2. Gorni, A.A., 1997. The application of neural networks in the modelling of plate rolling processes. J. Materials - Electronic (J.O.M.-E.), April.
3. Young-Sang, K., Y. Bong-Jin and K. Min, 2001. Robust design of artificial neural network for roll force prediction in hot strip mill, neural networks. Proc. IJCNN'01, Intl. Joint Conf., 4: 2800-2804.
4. Neelakantan, R. and J. Guiver, 1998. Applying neural networks. Hydrocarbon Processing, pp: 91-96.
5. Haykin, S., 1999. Neural Networks, a Comprehensive Foundation. 2nd Edn. Prentice Hall.
6. Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Oxford University Press.
7. Zietsman, J.H., S. Kumar, J.A. Meech, I. Samarasekera and J.K. Brimacombe, 1998. Taper design in continuous billet casting using artificial neural networks. Iron making and Steelmaking, 25: 476-483.
8. Vermeulen, W., A. Bodin and S. Van Der Zwaag, 1997. Prediction of the measured temperature after the last finishing stand using artificial neural networks. Steel Res., 68: 20-26.
9. Thomas, P. and G. Bloch, 1998. Robust pruning for multilayer perceptrons. Proc. IMACS/IEEE Multiconf. on Computational Engineering in Systems Applications CESA'98, Nabeul-Hammamet, Tunisia, April 1-4, 4: 17-22.
10. Nabney, I.T., 2002. Netlab-Algorithms for Pattern Recognition. Springer-Verlag.
11. Dairiki, O., H. Mabushi, I. Degawa and H. Nakamura, 1989. Progress of AGC utilisation techniques at plate mill of oita works. Nippon Steel Technical Report, No. 42, July, pp: 38-47.
12. Moussaoui, A.K., H.A. Abbassi, S.E. Bouazza and A. Kondratas, 2003. Neural-networks-based adaptive nonlinear controller for hot strip rolling mill. Mechanika, 6: 55-61.
13. Ford, H. and J.M. Alexander, 1963-64. Simplified hot rolling calculations. J. Institute of Metals, 92: 397-403.
14. Fumio, Y., S. Kunio, T. Masashi, A. Yoshiharu, A. Yasushi, F. Charles, G. Maurice and C. Trevor, 1991. Hot strip mathematical models and set-up calculation. IEEE Trans. Industry Applications, 27: 131-139.
15. Watanabe, T., H. Narazaki, A. Kitamura, Y. Takahashi, H. Hasegawa, 1997. A new mill-setup system for hot strip rolling mill that integrates a process model and expertise, systems, man and cybernetics, 'computational cybernetics and simulation. IEEE Intl. Conf., 3: 2818-2822.
16. Singh, S.B., H.K. Bhadeshia, D.J. Mackay, H. Carey and I. Martin, 1998. Neural network analysis of steel plate processing. Iron and Steel Making, 25: 355-365.
17. Misaka, Y. *et al.*, 1967-1968. Formulatization of mean resistance to deformation of plain C steels at elevated temperature. J. Japan Society for the Technology of Plasticity, 8: 414-422.
18. Royzman, S.E., 1996. Thermal stresses in slab solidification. Asia Steel, pp: 158-162.
19. Hornik, K., M. Stinchcombe and H. White, 1989. Multilayer feed forward networks are universal approximators. Neural Networks, 2: 359-366.