APPLICATION OF NEURAL NETWORKS IN THE MODELING OF PLATE ROLLING PROCESSES[1]

Antonio Augusto Gorni[2]

Neural networks are a relatively new technique of Artificial Intelligence that emulates the behavior of biological neural systems in digital software or hardware. These networks can "learn" automatically complex relationships between data. So, there is no need to previously propose any model to correlate the desired variables. This feature makes this technique very useful in the modeling of processes which mathematical modeling is difficult or impossible. This work describes some real examples of applications of neural networks in the modeling of some plate mill processes at Companhia Siderúrgica Paulista - COSIPA, a Brazilian steelmaker.

- INTRODUCTION

From the beginning of digital computing to the end of the 1980's, virtually all data processing applications adopted a basic approach: programmed computation. This approach requires the previous development of a mathematical or logical algorithm to solve the problem at hand, which have to be subsequentely translated into any computational language[1].

This approach is limited, because it only can be used in cases where the processing to be made can be precisely described in a known rule set. However, sometimes the development of such rule set is hard or impossible. Besides that, as computers work in a totally logical form, the final software must to be practically perfect to work correctly. So, the development of computer software is, indeed, a succession of "project-test-interactive improvement" cycles that can demand much time, effort and money.

During the late 1980's a revolutionary approach for data and information processing appeared: neural networks. This technique does not require previous development of algorithms or rule sets to analyse data. This can minimize significantly the software development

[1] Paper published at JOM-e, April 1997, http://www.tms.org/pubs/journals/JOM/9704/Gorni/Gorni-9704.html.

[2] Materials Engineer, M.Eng., Companhia Siderúrgica Paulista – COSIPA, Cubatão SP, Brazil. E-Mail: agorni@iron.com.br; Home Page: www.gorni.eng.br.

work needed for a given application. In most cases, the neural network is previously submitted to a “training” step using real known data, extracting then the methodology necessary to perform the required data processing. That is, a neural network is able to extract the required relationships from real data, avoiding the previous development of any model. This is the approach intuitively used in the biological neural systems, particularly by human beings.

- NEURAL NETWORKS TUTORIAL

One can understand more easily the difference between the behavior of programmed computation and neural networks comparing computers and humans. For example, a computer can perform mathematical operations more quickly and precisely than an human. However, the human can recognize faces and complex images in a more precise, efficient and quick way than the best computer available[2].

One of the reasons for this performance difference can be attributed to the distinct organization forms of computers and biological neural systems. A computer generally consists of a processor working alone, executing instructions delivered by a programmer, one-by-one. Biological neural systems, by its turn, consist of billions of nervous cells - that is, neurons - with an high degree of interconnection between themselves. Neurons can perform simple calculations without the need to be previously programmed[1-5].

The basic element of a neural network is called, naturally, neuron. It is also known as node, processing element or perceptron. It can be schematically viewed in figure 1. The links between neurons are called synapses.

The input signal to a given neuron is calculated as follows. The outputs of the preceding neurons of the network - that is, their state or activation values X_1, X_2 and X_3, in the specific example of figure 1 - are multiplicated by their respective synapse weights, P_1, P_2 and P_3.. These results are summed up, resulting in the value u, that is delivered to the given neuron. By its turn, the state or activation value of this neuron is calculated by the application of a threshold function to its input value, resulting in the final value v. This threshold function, also called activation function, frequently is non-linear, and must be chosen criteriously, as the performance of the neural network heavily depends on it. Generally this function is of the sigmoidal type.

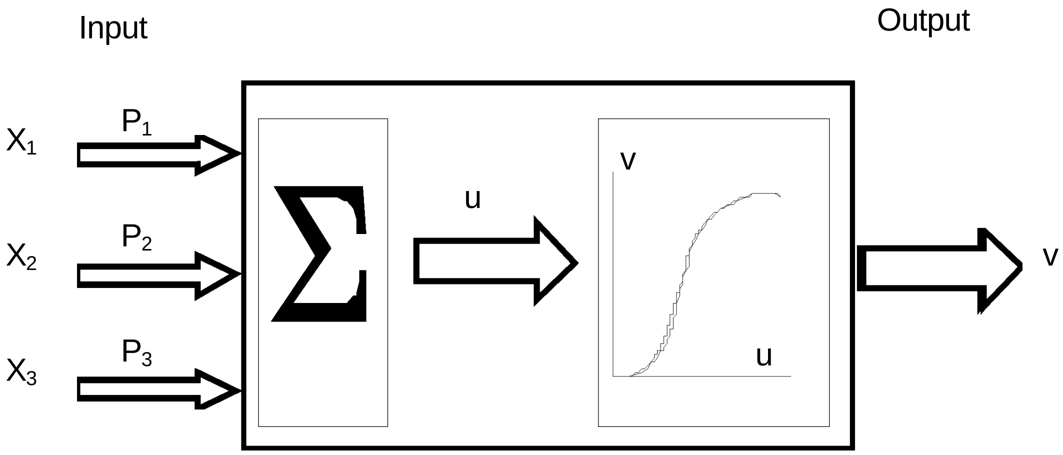

Figure 1: Schematic representation of a neuron.

How can a neural network "learn"? During the "training" step, real data - input and output - are continuously presented to it. Then it periodically compares real data with the results calculated by the neuron network. The difference between real and calculated results - that is, the error - is processed through a relatively complicated mathematical procedure, which adjusts the value of the synapse weights in order to minimize this error. This is an importante feature of the neural networks: their knowledge is stored in their synapse weights.

The duration of the "training" step must be not excessively short, in order to allow the network to fully extract the relationships between variables. However, this step could not either be very long: in this case, the neural network will simply "memorize" the real data delivered to it, "forgetting" the relationships between them. So, it is advisable to break away approximately 20% of the available data in a subset and to use only the remaining 80% for the training of the neural network. The training step must be interrupted periodically and then the network must be tested using the 20% subset, checking the precision of the calculated results with real data. When the neural network precision stabilizes and stops to grow, it is time to consider the neural network as fully trained.

Figure 2 shows two basic types of neural networks regarding data flow and training type. The Rummelhart type neural network shows data flow in one direction - that is, it is an unidirectional network. Its simplicity and stability makes it a natural choice for applications like data analysis, classification and interpolation. Consequentely, it is particularly suitable for process modeling, and, in fact, there is many real world applications of this type of network. A fundamental characteristic of this network type is the arrangement of neurons in layers. Of course, there must have at least two layers in this kind of network: data input and data output. As the performance of two-layer neural networks is very limited, generally it is included at least one more intermediate layer, also called hidden layer. Each neuron is linked to all the neurons of the neighbouring layers, but there is no links between neurons of the same layer. The behavior of this kind of network is static; its output is a reflexion of its respective input. It must be previously trained using real data in order to perform adequately.

The other neural network seen in figure 2, of the Hopfield type, is characterized by a multidirectional data flow. Its behavior is dynamic and more complex than the Rummelhart networks. The Hopfield nets do not show neuron layers: there is total integration between input and output data, as all neurons are linked between themselves. These networks are tipically used for studies about optimization of connections like, for the example, the famous Travel Salesman Problem. This kind of neural network can be trained with or without supervision; the purpose of its training is the minimization of its energy, leading to an independent behavior. However, there is no practical application of this kind of network up to this moment.

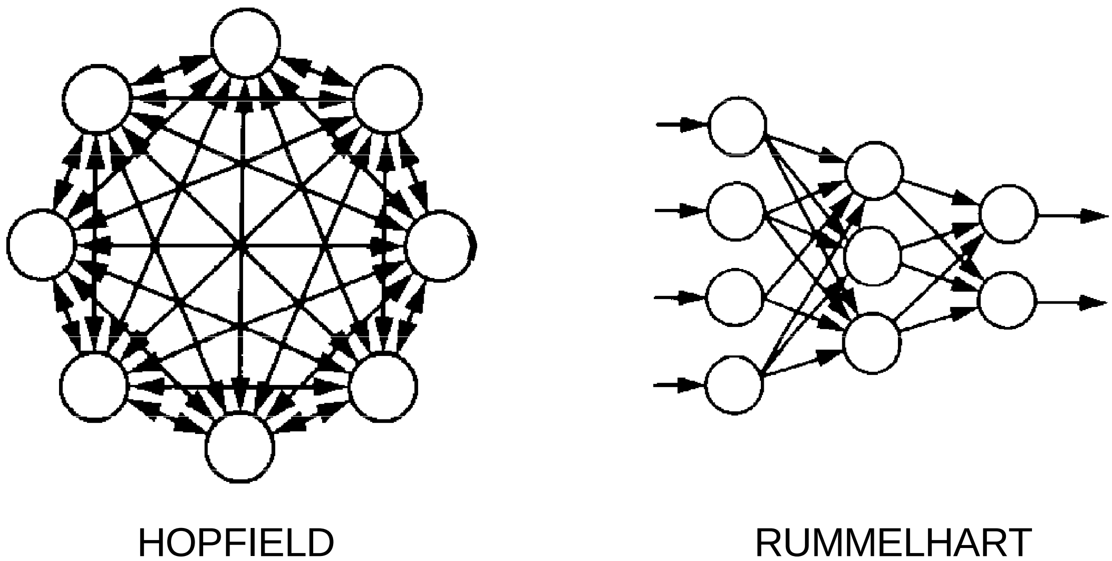

Figure 2: Basic types of neural networks.

As told before, applications particularly suited for neural networks are those which mathematical formulation is very hard or impossible. For example:

- Signal analysis and processing;
- Process control;
- Robotics;
- Data classification;

- Data smoothing;
- Pattern recognition;
- Image analysis;
- Speech analysis;
- Medical diagnostics;
- Stock market forecasting;
- Analysis for loan or credit solicitations;
- Oriented marketing.

The comparison between neural networks and expert systems shows that the development of the former technique is more quick, simple and cheap. However, a major drawback of the use of neural networks arises from the fact that it is not always possible to know how a neural network got a given result. Sometimes this can be very inconvenient, mainly when the neural network calculated results are atypical or unexpected.

However, the use of hybrid artificial intelligent systems - that is, conjugated use of neural networks with expert systems or fuzzy logic - are increasingly showing good results, through the optimized use of its best characteristics.

There are some advantages of neural networks towards multiple regression. There is no need to select the most important independent variables in the data set, as neural networks can automatically select them. The synapses associated to irrelevant variables readily show negligible weight values; on its turn, relevant variables present significant synapse weight values. As said previously, there is also no need to propose a function as model, as required in multiple regression. The learning capability of neural networks allow them to “discover” more complex and subtle interactions between the independent variables, contributing to the development of a model with maximum precision. Besides that, neural networks are intrinsically robust, that is, they show more immunity to noise eventually present in real data; this is an important factor in the modelling of industrial processes.

It must be noted that the criterious use of statistical techniques can be extremely useful in the preliminary analysis of raw data used for the development of a neural network. Data can be previously refined, minimizing even further the development time and effort of a reliable neural network, as well maximizing its precision. Hybrid statistical-neural networks systems can be a very useful solution to some specific problems.

There are countless examples of neural network applications in the metallurgy field. Some cases regarding hot rolling of steel are listed below:

- Sizing of slabs for plate rolling[6];
- Modelling of hot strength of steel from temperature, strain and strain rate[7];
- Same as above, including effect of chemical composition[8];
- Determination of TTT diagrams from the chemical composition of steel[9];
- Pass schedule calculation for hot strip mills[10,11].
- Feasibility of production of particular steel grades in a steelworks, evaluated from information about its required mechanical properties[12].

- Neural Network Applications for Plate Mills

In order to check neural networks performance at real conditions of COSIPA’s industrial rolling mills, it was decided to use this new technique for off-line modelling of several plate mill processes. In some cases, these processes were already modeled using statistical techniques. This fact will permit the comparison between both approaches.

All neural networks developed here were of the Rummelhart type, with one hidden layer and trained by the retropropagation method. They always have a *bias neuron* included in the input layer. This is a neuron with constant unitary value; its inclusion improves the modeling capacity of the neuron network[1].

In each case, 80% of the global raw data was reserved for the training of the network, whereas the remaining 20% was periodically used during the precision evolution check of the neural network. The training step of most of the neural networks studied in this work converged in 60,000 iterations. The evaluation of the precision of the fully trained neural network was performed through the calculation of the Pearson’s correlation coefficient **r** and the standard error of estimative, as well through disperson plots. The software used for development and training of the neural networks was NeuralWorks.

. Modeling the Thermal Profile of Slabs in the Reheating Furnace

Thermal profiles of slabs being reheated are periodically collected at COSIPA’s plate mill. These profiles are measured with an instrumented slab, which has drilled holes at several locations and depths. Chromel-alumel thermocouples are inserted into these holes and connected to a data logger, which collects all the temperature evolution of these points during slab reheating. The data logger is sheltered in a inoxidable steel box coated with rock wool and filled with water and ice.

It was decided to develop a neural network model to forecast the inner temperature of the slabs being reheated as a function of reheating time and its superficial temperatures. This is a case with a relatively easy mathematical solution, and thus adequate to allow a comparison between the performance of the neural network and the conventional numerical models.

With this purpose in mind, it was developed a neural network with three layers, after several configuration trials:

- Input layer with three neurons:
  . Reheating time [min]
  . Slab upper surface temperature [^{o}C]
  . Slab lower surface temperature [^{o}C]

- Hidden layer with thirteen neurons;

- Output layer with ten neurons, each of them representing a point in the instrumented slab where the temperature was measured.

The use of more or less neurons in the hidden layer, as well the use of more than one hidden layer did not improve the performance of the neural network.

It was verifyed that this neural network showed its best performance in the forecasting of the temperature in the mid-thickness of the slab, with Pearson's correlation coefficient **r** about 0.997 and standard error of estimate of 26.5^{o}C. The worst performance of this neural network occured in the forecasting of the point near the lower surface of the slab: its results showed a Pearson's correlation coefficient **r** of approximately 0.993 and standard error of estimate of 36.6^{o}C. Figure 3 shows the disperson plots of the calculated and real temperatures of both two points considered.

The performance reached by this neural network was considerated adequate, as it was similar to previously developed mathematical models, which showed errors of approximately 30^{o}C.

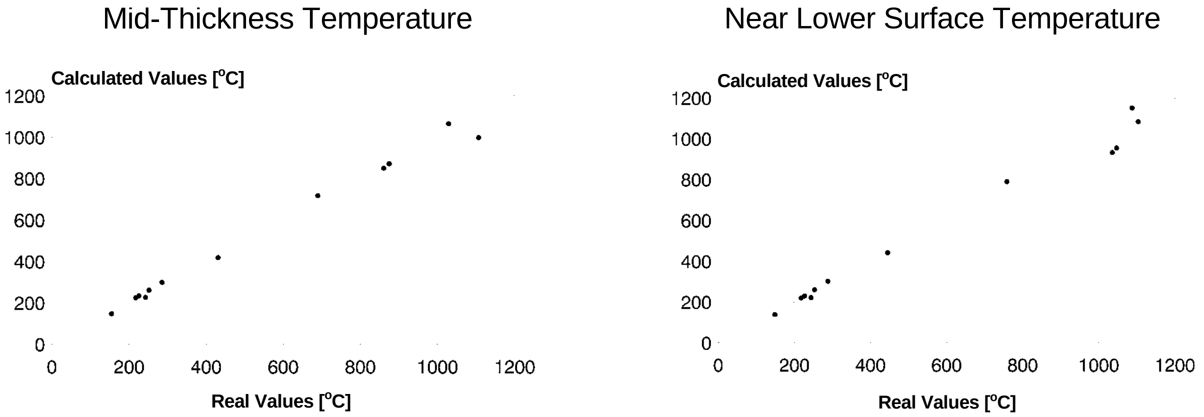

Figure 3: Precision achieved by the neural network for the best and worst case (mid-thickness and near the lower surface temperatures) for the modeling of thermal profile of slab during reheating.

- Detection of "Turn-Up" during Plate Rolling

The turn-up, or excessive bowing upwards of rolled stock during plate rolling is a serious problem, as material being rolled can collide with the rolling stand or ancillary equipments, causing extensive damage. This problem was frequent at COSIPA's plate mill, specially during the processing of Ni steels.

A previous work showed that alterations in the pass schedule could minimize the ocorrence of turn-up, and led to the development of a statistical model for the calculation of an optimized pass schedule. It was showed then that there was a critical range of strain values to be avoided during plate rolling.

However, sometimes this statistical model calculated unfeasible values of roll gaps. In some cases the calculated values were excessively low, jeopardizing productivity; in other occasions, they were excessively high, well above the mill's capacity of load, torque and power.

As soon the use of neural networks became available, it was a natural idea to use them in this application, as the statistical model was unsatisfactory. After several trials, it was developed the following neural network:

- Input layer with five neurons:
  - . Desired turn-up index
  - . Work roll peripherical speed [r.p.m.]

. Rolling load [t], calculated by the Sims model

. Rolling stock width [mm]

. Roll gap distance [mm] used in the former rolling pass.

- Hidden layer with eleven neurons.

- Output layer with one neuron, which represents the recommended roll gap distance [mm] for the next rolling pass.

The *turn-up index* used here, one of the input variables, was defined using an arbitrary scale, from 0 to 5, that is proportional to defect seriousness.

The neurons number of the hidden layer of this neural network was calculated after the *Hecht-Kolmogorov's theorem*, which affirms that the optimum number of neurons of a hidden layer is equal to twice the number of input neurons plus one.

The developed neural network showed Pearson's correlation coefficient **r** of 0.992 and standard error of estimate of approximately 3.0 mm. Figure 4 shows the dispersion plot of the real and calculated values. The most influencing variables, as indicated by the trained neural network, are turn-up index and initial roll gap distance, followed by rolling stock width, rolling load and work roll peripherical speed, considering a decreasing rank of importance.

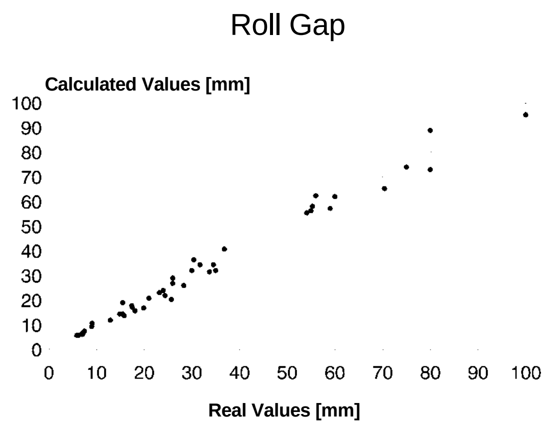

Figure 4: Dispersion plot of the calculated and real values from the model for roll gap calculation regarding control of the turn-up defect.

As can be seen in the graphic of figure 4, all calculated values are very near from the real values. That is, the neural network did not generate non-sense values as the previous developed regression polynomial.

The previous work about the turn-up occurrence in COSIPA's plate mill had revealed that this defect was more frequent in a specific range of roll gap values, from 80 to 60 mm. This is confirmed by the trained neural networks, as the initial roll gap value is one of the most influencing variables of the model. Besides that, the standard error of estimate is admissible, as it is equal to only 7,5% of the minimum roll gap value. However, this is valid since the final thickness of plate is not between 40 to 80 mm.

- Longitudinal Discard Calculation when Using Plane View Control

Plate rolled from continuously cast slabs generally presents longitudinal extremities with tongue shape. This lowers the rectangularity index of the rolled stock, affecting its metallic yield, as irregular portions in the extremities of plate have to be cut. This characteristic can be attributed to the peculiar thickness profile of the continuously cast slabs. These slabs are slightly thicker in mid-width, resulting in a non-homogeneous mass distribution along the rolling stock, which affects the shape of its longitudinal extremities.

One solution proposed for this problem consists in the application of a special thickness profile in the rolling stock during the application of the last pass of the broadsizing step. After this special pass, the width of the rolling stock presents a thickness profile that basically consists of a "V"-shape notch or presents the shape of a "dog bone". The development of this process of COSIPA led to a 0,7% increase in the metallic yield of the plate mill.

During the development of this process, it was developed a regression polynomial to correlate the "V"-shape notch depth and the total strain applied to the rolled stock after the broadsizing step with the length of the discarded portion of the final rolled stock, in order to have a better understanding of the process and check its optimization possibilities. This polynomial presented a Pearson correlation coeficient **r** of 0.903 and standard error of estimative of 132 mm.

Once more this appeared to be a good application for the neural networks technique, as it could be an opportunity to improve forecasting of the length of the discarded portion. The best neural network designed to substitute the polynomial equation had the following configuration:

- Input layer with two neurons:
    . "V"-shape notch depth [mm]

. Total strain applied to the rolled stock after broadsizing step [%]

- Hidden layer with five neurons (according to the already mentioned Hecht-Kolmogorov's theorem).

- Output layer with one neuron, which represents the length of the discarded portion in the final rolled stock [mm].

This neural network showed a Pearson correlation coefficient **r** of 0.943 and standard error of estimate of 61 mm. The disperson plot of real and calculated values of the length of the discarded portion in the final rolled stock can be seen at figure 5.

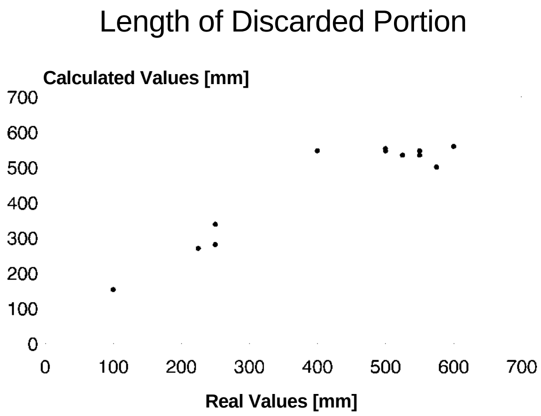

Figure 5: Dispersion plot of the calculated and real values regarding the neural network model for the calculation of the length of the discarded portion in plates submitted to plane view control during rolling.

Although the neural network had showed better performance than the regression polynomial (the standard error of estimated fell approximately 54%), errors observed in figure 5 still are significative. Perhaps the cause of these relatively high errors stems from the use of the discard length as a evaluation parameter of the metallic yield of the process. Really this parameter is less representative than the weight or area of the discarded portion but, in compensation, is a easier variable to be measured under industrial conditions.

- Pass Schedule Calculation Aiming Plate Flatness Optimization

One of the most stringent quality parameters of plate is its flatness index. The traditional approach to flatness control during rolling consists to keep the rate *crown variation : thickness variation* within a restricted range, specially during the last three passes of the rolling schedule. This fact was also confirmed at COSIPA's plate mill.

Mathematical models for the calculation of pass schedules are relatively easy to develop, but the use of neural networks is simpler. So, it was decided to use this new technique in this application too. The three last passes of the rolling schedule were modeled regarding optimization of plate flatness; one neural network was attributed for each pass. The respective three neural networks showed the same configuration, as follows:

- Input layer with ten neurons:
  . Thickness of final plate [mm]
  . Width of final plate [mm]
  . Aimed flatness index in final plate
  . Flatness index observed after prior pass
  . Roll gap of prior pass [mm]
  . Rolling load measured during prior pass [t]
  . Temperature measured during prior pass [°C]
  . Original crown of the upper work roll [mm]
  . Original crown of the lower work roll [mm]
  . Rolling stock tonnage since the last change of work rolls [t].

- Hidden layer with twenty-one neurons (according to the already mentioned Hecht-Kolmogorov's theorem).

- Output layer with one neuron, which represents roll gap value of the corresponding pass [mm]

The flatness index used in this model varied in the range from 0 to 5: this index was greater as plate flatness worsened.

The performance of the neural network was very good. The Pearson's correlation coeficient **r** corresponding to the last but two, penultimate and last pass were 0.998, 0.998 and 0.999, respectively; their standard error of estimate were 0.430, 0.394 and 0.140 mm, respectively. Figure 6 shows the dispersion plots of the real and calculated values for the three last passes of the rolling schedule.

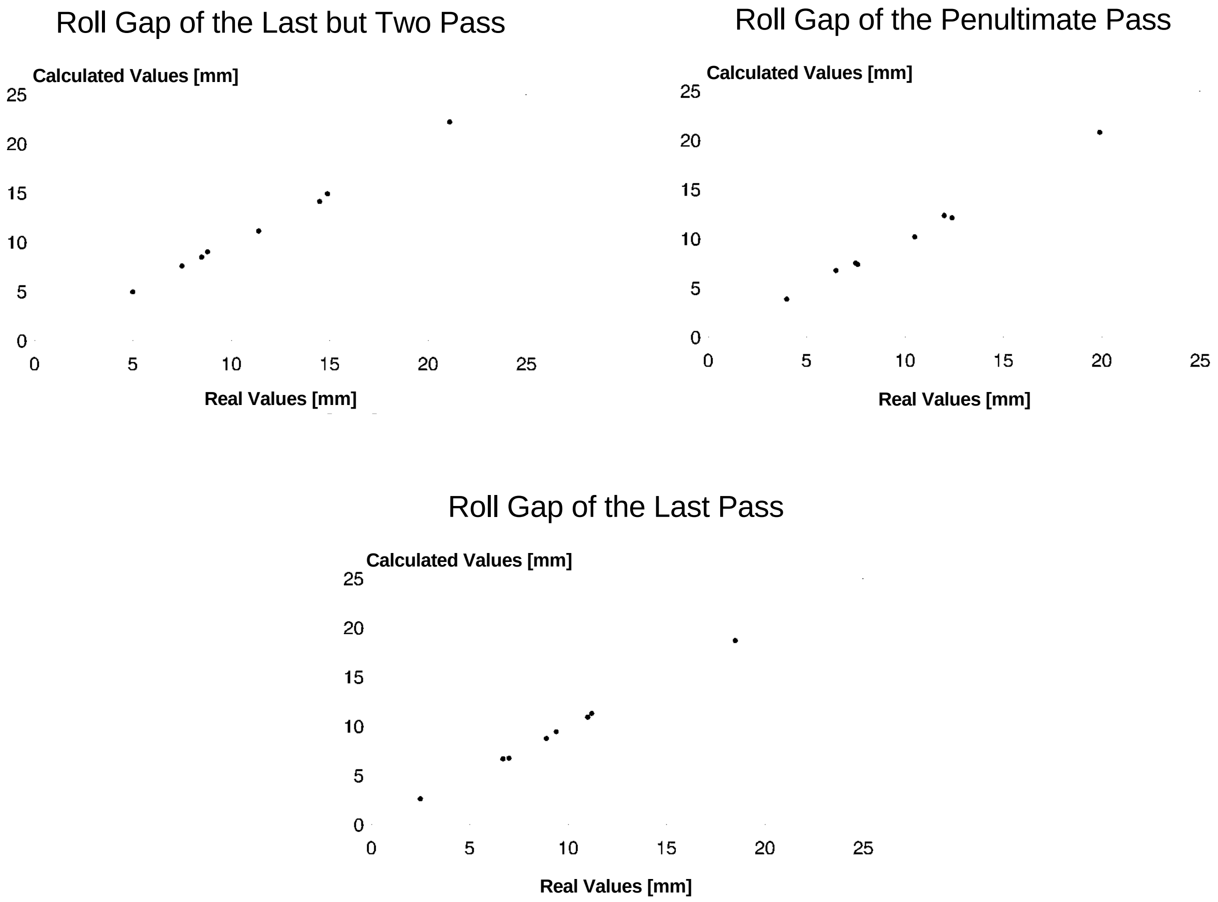

Figure 6: Dispersion plots of the calculated and real values of the last three passes of plate rolling, regarding the model for calculation of the pass schedule optimizing plate flatness.

The most important variables in these neural networks were the aimed flatness index, the original crown in the upper work roll and the rolling stock tonnage after the last change of work rolls. Following to this group, there were parameters of intermediate importance like the final plate width/thickness and temperature/load of the prior pass. Finally, variables like prior pass flatness index/roll gap did not show great influence, but were vital to improve the precision of the neural networks, making feasible its use under industrial conditions.

In fact, these neural networks identified during their "learning" step the most important variables related to flatness that are "traditionally" defined by the rolling theory: original crown of work rolls and rolling stock tonnage since change of work rolls. This last variable generally shows good correlation with thermal crown and wear of work rolls, factors that affect the resultant roll crown and, consequentely, plate flatness. Work roll deflexion promoted by rolling load was also considered, as this last variable was included in the input layer of the neural networks.

The last pass is the most important to define the final dimensions of plate, specially its thickness. The errors observed in the results calculated by the respective neural network varied from -0,26 to +0,17 mm, practically comprised within commercial plate thickness tolerance range. This results can be even improved, as a more precise data acquisition system becomes available, thus avoiding human errors during data collection and improving the precision of the measured parameters. Those facts undoubtely will contribute to a better accuracy of these neural networks.

- CONCLUSIONS

The experience of applications of neural networks in process modeling at COSIPA's plate mill showed that this is a simple but powerful technique. The models were quick developed using only real data. They showed better precision than statistical models previously available, that required much more effort for its development.

Unfortunately the lack of instrumentation, data acquisition and computer facilities at COSIPA's plate mill hindered the application on-line of these models. Apparently, one of the disadvantages of neural networks is the complexity of hardware and software necessary to its industrial application. If process conditions become different from the previous situation used during training of a neural network, data must be collected, analysed and used for new training under the new conditions.

Other aspect to be considered is the lack of trust of some people about the real performance of the neural networks, as one does not know exactly how they generate their results. This, however, is a matter of time: only continuous use of neural networks, under the most different conditions, will really show how much they are reliable. The crescent number of industry applications of this technique, including the metallurgical field, is confirming that neural networks are here to stay.

- REFERENCES

1. HECHT-NIELSEN, R. *Neurocomputing*. Addison-Wesley Publishing Company, Reading, 1990. 433 p.

2. EBERHART, R.C. & DOBBINS, R.W. (editors). *Neural Network PC Tools - A Practical Guide*. Academic Press, San Diego, 1990, 414 p.

3. MÜLLER, B. et al. *Neural Networks: An Introduction*. Springer-Verlag, Berlin, 1995, 330 p.

4. FREEMAN, J.A. & SKAPURA, D.M. *Neural Networks: Algorithms, Applications and Programming Techniques*. Addison-Wesley Publishing Company, Reading, 1992, 401 p.

5. BLUM, A. *Neural Networks in C++*. John Wiley & Sons, Inc., New York, 1992, 214 p.

6. OLIVEIRA, J.B. et al. "Use of Neural Networks in the Plate Production Scheduling at USIMINAS". In: *Seminário de Laminação*, Associação Brasileira de Metais, Porto Alegre, 1992, 319-339.

7. GORNI, A.A. "Mathematical Modeling of the Hot Strength of HSLA Steels". In: *1st Metal Forming Week*, Associação Brasileira de Metalurgia e Materiais, Joinville, 1993, 267-86

8. LENARD, J.G. et al. "A Comparative Study of Artificial Neural Networks for the Prediction of HSLA and Carbon Steels". *Steel Research*, February 1996, 59-65

9. DOUNADILLE, C. et al. "Usage de Reseaux Neuronaux por Prevision des Courbes de Transformations de Acier". *Revue de Metallurgie - CIT*, oct. 1992, 892-94.

10. ORTMANN, B. et al. "Modernization of the Automation in the Hot Wide Strip Mill at Voest-Alpine Stahl". *Metallurgical Plant and Technology - International*, 6/1994, 26-34.

11. PORTMANN, N. et al. “Application of Neural Networks in Rolling Mill Automation”. *Iron and Steel Engineer*, Feb. 1995, 33-6

12. WATANABE, M. et al. “Development of Shape Steel Quality Design Expert system”. *Nippon Steel Technical Report*, April 1992, 29-44.