Rules Extraction Via Neural Networks Previously Trained

LUIS E. ZÁRATE; ALEXANDRA S. COSTA;ELISÂNGELA MAIA and MARK A. SONG
Pontifícia Universidade Católica de Minas Gerais,
Av. Dom José Gaspar nº 500,Coração Eucarístico, CEP. 30535-610, Belo Horizonte, MG
Brazil, zarate@pucminas.br

Abstract: -In this paper, two methods for extraction of knowledge rules through Artificial Neural Networks, with continuous activation functions are presented. Those rules are extracted from neural networks previously trained and of the sensitivity factors obtained by the differentiation of a neural network. The rules can be used when analytic models of the physical processes lead to equations of difficult numerical and analytical solutions. In the operation of industrial processes, the rules can help for the taking of decision of less experienced operators. The proposed methods will be applied for the obtaining of knowledge rules for the cold rolling process.

Key Words**:** Artificial Neural Networks; Knowledge Representation; Industrial Process; Machine Learning.

1 Introduction

The physical modeling is an activity that demands larger time in the understanding of the process. During the development of a model, there are mathematical complexities that don't have analytic solutions of easy resolution. This takes to numeric solutions with great effort computational. The Artificial Neural Networks (ANN) learn the training sets supplied and its largest application is in the representation of physical processes. Through ANNs is possible to relate the variables of interest of the process, decreasing the physics detailed study that relates the variables.

Neural Networks are excellent tools for the machine learning, because these are applicable in a great variety of problems and have generalization capacity. A limitation of Neural Networks is that, the represented concept through the training sets, are extremely difficult for the human understanding. The ANN's associates values of the training sets, and they inform its results but don't explain as these were discovered, ANN's are still a “black box”. For so much, there is an implicit knowledge, difficult to extract, whether extracted it can reinforce an intelligent machine in the mechanisms to explain its conclusions.

There are three strategies for knowledge extraction for neural networks. The first consists of inserting knowledge, which doesn't need to be complete nor correct, inside of the neural network so that through the training process, and considering the generalization capacity of the ANN's, to recover and to identify the knowledge incomplete. This technique is called as "Knowledge-based Neural Networks” - KNNs and it is seen as a representation of symbolic rules, [3].

A second strategy is to train the nets KNN using training sets selected, doing that the rules considered incorrect, used in the first training of KNN are corrected to be consistent with the selected training sets. The third strategy is the knowledge extraction of nets trained. That is of extreme complexity due mainly the different configurations of the nets, [4] and [5].

The three strategies above have been applied for problems of symbolic knowledge, where the activation function is a function step (or binary).

$$Y = \begin{cases} 1 & net \geq 0 \\ 0 & net < 0 \end{cases}$$

In this work, two strategies for extraction of rules from neural networks previously trained, with continuous activation functions are presented. The methods can be applied for physical processes and are a new form for extracting knowledge of the process without to obtain complex analytic expressions based on models. The methods can be used in expert and supervision systems and control on-line, where the computational effort is critical.

On the other hand, many metallurgical processes depend on the ability of the human operator. In rolling processes the number of specialists is small, while the need for its experience is big. Thus, there are interest by the construction of expert and supervision systems, that allow the participation of less experienced operators. The strategies proposals in this work can supply knowledge for these types of systems.

In [6] were used neural networks for cause-effect analysis of an observation, applied for rolling process. In [7] two methods based on qualitative rules to describe the cold rolling process were proposed. Those rules don't describe the relationship degree of the parameters, what can be introduced through the sensitivity factors and/or fuzzy sets. This paper contributes to understand of the cold rolling process, analyzing its behavior for an operation points chosen.

The methods proposed allow to represent data and information and possess high computational performance been attractive for on-line supervision systems. For example, through the rules IF-THEN is possible to deduce in a correct way, the adjustment used to correct deviations in the output thickness of the material been rolling.

This paper is divided in 4 sections. In the section 2, the methods for extraction of rules of ANN's are presented; In the section 3, the application of the methods for the cold rolling process are discussed and in the section 4, the conclusion of the paper is presented.

2 Knowledge Rules via Ann's Trained

The Neural Networks are powerful tools for Machine Learning, ([1], [2]). But ANN are considered essentially a "black box" limiting the use in applications where the explanation of the reasoning is fundamental. ANN's associates values of the training sets, inform results but don't explain as these results were discovered. There is an implicit knowledge, difficult to extract, that whether obtained, it can reinforce an intelligent machine in the mechanisms to explain its conclusions.

To understand the process for rules extraction through ANN, will be described the case for neurons with binary activation:

For $Y = 1$ is obtained: $net = \sum_{i=0}^{N} X_i W_i \geq 0$

where: X are the entries and W are weight of the neuron

$$\text{or: } \sum_{i=1}^{N} X_i W_i + X_0 W_0 \geq 0$$

As $X_0 = 1$ and $W_0 = -\theta$, then : $\sum_{i=1}^{N} X_i W_i - \theta \geq 0$

For the limit $net = 0$: $\sum_{i=1}^{N} X_i W_i = \theta$

After the training process, if $W_i = +1$, is obtained:

$$\sum_{i=1}^{N} X_i = \theta$$

By example, for N=3 and considering:

$$\{X_1, X_2, X_3, Y\} = \{A, B, C, 1\}$$

An rule obtained of the net, can be:

If A and B and C Then θ

For the case: $W_1 = -1$, and $W_i = +1 \quad \forall i = 2,...,N$:

If (NOT)A and B and C Then θ

2.1 First Method: Rules Extraction Through ANN with Sigmoid Function

The rules above are rules symbolic extracted of the neural network with activation function of binary type. For the case of ANN's with continuous activation functions, for example sigmoid, the rules extraction process is a more detailed process.

$$Y = \frac{1}{1 + \exp^{-net}}$$

Notice in the previous expression, that there are infinities values for the exit net (*Y*). In this work one of the methods proposed for the extraction of rules is based on the "Intervalar" analysis of *Y*. In the future, new strategies, for extraction of the rules, based on fuzzy logic can be used.

Considering the neuron with sigmoid activation function, for the exit *Y* can be defined the intervals, as it follows :

$$\begin{cases} Y < y_1 & for \quad net < \theta_1 \\ y_1 \leq Y \leq y_2 & for \quad \theta_1 \leq net \leq \theta_2 \\ Y > y_2 & for \quad net > \theta_2 \end{cases} \tag{1}$$

If linguistic values, *SMALL*, *MEDIUM* and *LARGE* are associated for the intervals in *Y*:

$$\begin{cases} SMALL\ (S) & for \quad Y < y_1 \\ MEDIUM\ (M) & for \quad y_1 \leq Y \leq y_2 \\ LARGE\ (L) & for \quad Y > y_2 \end{cases} \tag{2}$$

as: $net = \sum_{i=0}^{N} X_i W_i$ for (S) : $\sum_{i=0}^{N} X_i W_i < \theta_1$;

for (M): $\theta_1 \leq \sum_{i=0}^{N} X_i W_i \leq \theta_2$ and for (L): $\sum_{i=0}^{N} X_i W_i > \theta_2$

Considering (Y = SMALL): $\sum_{i=1}^{N} X_i W_i + W_0 < \theta_1$;

$\sum_{i=1}^{N} X_i W_i < \theta_1 - W_0$

as the values of W_i are calculate during the training process, the method consists in to search the values X_i that satisfy the equation. The values of X_i, determine the rank for which the exit *Y* presents a value tuned as SMALL. Through the method to be proposed will be extracted rules of the type:

$$\begin{aligned} &If \quad A \in [X_1]^P \quad \text{and} \quad B \in [X_2]^P \\ &\text{and} \quad C \in [X_3]^P \quad \text{Then} \quad Y = SMALL \end{aligned} \tag{3}$$

The rule Eq. (3) should be read as: “if the entry “A” belongs to the interval of X_1, where *Y* is SMALL and if the entry “B” belongs to the interval of X_2, where *Y* is SMALL and if the entrance “C” belongs to the interval of X_3, where *Y* is SMALL, Then output of the net *Y* is SMALL”.

A hypotheses to be analyzed is the possibility to have intervals $[X_i]^P = [X_i]^M = [X_i]^G$ for some “*i.*”. This means that the entry can be eliminated, therefore it doesn't increase information to the rule extracted.

For neural networks with two layers, with N entries, L neurons in the hidden layer and M exits with sigmoid activation function, the method proposed only depends on the entries sets and of the exit Y of the net.

2.2 Second Method: Rules Extraction via Sensitivity Factors

The modeling is a representation of the physical aspects of a process, which provides knowledge and information usable. The analytic model allows to analyze the cause-effect relationship of the variations in the parameters of the process. As the knowledge of the world is finite, it is therefore incomplete. The models provide numeric amounts of the related variables and as the idea that we have of a simple number is finite, of there the model is always incomplete [6]. A model can be described better if considered the sensitivity factors of the parameters. Those factors inform which parameter causes larger or smaller effect on the process.

Physical models are usually difficult to obtain. To calculate the sensitivity factors is a difficult task and depending on the complexity of the model can demand numeric solutions with great effort computational. The sensitivity factors can be obtained by differentiating a neural network trained [6]. In this paper a method for extraction of rules with sensitivity factors obtained by the differentiating a neural network is presented.

Analyzing the process and choosing $U = \{u_1, u_2, ..., u_m\}$ as the entries vector and as $Z = \{z_1, z_2, ..., z_m\}$ the exits vector, that represent the process parameters, the combinations Domain-Image allow to obtain the training sets. In way to have a generic method, a multi-layers neural network with N entries, M exits and L neurons in the hidden layer was considered.

After the training process is calculated the sensitivity factors, by differentiating of ANN. In [6] the expressions to calculate the sensitivity factors are:

$$\frac{\partial Z_k}{\partial U_i} = \begin{bmatrix} R_1 W^o_{11} & R_1 W^o_{12} & \cdots & R_1 W^o_{1L} \\ R_2 W^o_{21} & R_2 W^o_{22} & \cdots & R_2 W^o_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ R_M W^o_{M1} & R_M W^o_{M2} & \cdots & R_M W^o_{ML} \end{bmatrix}$$

$$\begin{bmatrix} \frac{Q_1 W^h_{11}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_1 W^h_{12}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_1 W^h_{1N}}{\text{emax}_N - \text{emin}_N} \\ \frac{Q_2 W^h_{21}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_2 W^h_{22}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_2 W^h_{2N}}{\text{emax}_N - \text{emin}_N} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{Q_L W^h_{L1}}{\text{emax}_1 - \text{emin}_1} & \frac{Q_L W^h_{L2}}{\text{emax}_2 - \text{emin}_2} & \cdots & \frac{Q_L W^h_{LN}}{\text{emax}_N - \text{emin}_N} \end{bmatrix} \quad (4)$$

$$Q_k = \frac{\exp^{-(\sum_{i=0}^{N} W^h_{ki} X_i)}}{(1+\exp^{-(\sum_{i=0}^{N} W^h_{ki} X_i)})^2} \qquad k = 1,..., L$$

$$R_k = (\text{smax}_k - \text{smin}_k)\frac{\exp^{-V_k}}{(1+\exp^{-V_k})^2}$$

$$k = 1,.., M$$

$$\frac{\partial V_k}{\partial U_i} = \frac{\partial}{\partial U_i}(W^o_{k0} + \sum_{j=1}^{L} W^o_{kj} f^h_j(\sum_{i=0}^{N} W^h_{ji} f^a_i(U_i)))$$

Where :

U_i, $i = 0,..., N$ are the net entries (non-normalized) and $U_0 = 1$ is a polarization entry

$f^a_i(.)$ $i = 0,..., N$ are the normalization entry functions with $f^a_0(.) = 1$

X_i, $i = 0,..., N$ are the normalized entries $X_0 = U_0$

W^h_{ij} $i = 1,..., L$ and $j = 0,..., N$ is the weight corresponding to the neuron *i* and entry *j*

$f^h_j(net^h_j)$ $j = 0,..., L$ where $f^h_0(net^h_0) = 1$ is the sigmoid function of the hidden layer.

W^o_{ij} $i = 1,..., M$ and $j = 0,..., L$ is the weight of the neuron i and entry j for the exit layer

$f^o_j(net^h_j)$ $j = 1,..., M$ is the value of the sigmoid function for the exit layer

$f^b_i(.)$ $i = 1,..., M$ are the non-normalized functions of the exits

Z_i, $i = 1,..., M$ net exit values

$e\max_k, e\min_k$ $k = 1,.., N$ higher and lower value of the entries

$s\max_k, s\min_k$ $k = 1,.., M$ higher and lower value of the exits

As each sensitivity factor $\partial Z_i / \partial U_j$, $i = 1,..., M$ and $j = 1,..., N$ has different values inside of the interval, it is possible to define intervals for each sensitivity factor as:

$$\partial Z_i / \partial U_j = \begin{cases} SMALL(S) & para \quad I^i_j < \text{limite}_1 \\ MEDIUM(M) & para \quad \text{limite}_1 \le I^i_j \le \text{limite}_2 \\ LARGE(L) & para \quad I^i_j > \text{limite}_2 \end{cases} \quad (5)$$

As the values of W_i are calculate during the training process, the method proposed consists of calculating the different sensitivity factors, Eq. (4), and to search through an algorithm the values U_i that satisfy the conditions in Eq (3). The values of

U_j determine the Domain for which the output (*Z*) has a value tuned as *SMALL*, *MEDIUM* or *LARGE*.

Through the method proposed is possible to find rules of the type:

$$\begin{aligned} &If \quad A \in [U_1]^P \text{ and } B \in [U_2]^P \text{ and } C \in [U_3]^P \\ &\text{Then } \partial Z_i / \partial U_j = SMALL \end{aligned} \tag{6}$$

The rule above should be read as: "if the entry "A" belongs to the interval of U_1, where $\partial Z_i / \partial U_j$ is SMALL and if the entry "B" belongs to the interval of U_2, where $\partial Z_i / \partial U_j$ is SMALL and if the entrance "C" belongs to the interval of U_3, where $\partial Z_i / \partial U_j$ it is Then SMALL the sensitivity of ∂Z_i with regard to ∂U_i is SMALL.".

3 Application for Rolling Process

The Eqs. (7) and (8) describe the equations for the rolling load and torque respectively.

$$P=f(y,h_i,h_o,t_b,t_f,\mu,E,D) \tag{7}$$

$$T_q=f(y,h_i,h_o,t_b,t_f,\mu,E,D) \tag{8}$$

For rolling process, the neural representation was expressed through the relationship Eq. (9). With six entries, two exits and 13 neurons in the hidden layer.

$$(h_i, h_o, \mu, t_b, t_f, \bar{y}) \xrightarrow{\text{RedeNeural}} (P, T_q) \tag{9}$$

where: h_i is input thickness, h_o output thickness, μ friction coefficient, t_f front tension, t_b back tension and $\bar{y}$ average yield stress. The outputs were chosen as: rolling load (*P*) and rolling torque (*Tq*).

In this section, an application of the method proposed to a strip rolling process is presented. The operation point was chosen as: h_i =5.0 mm; h_o = 3.6 mm; μ =0.12 t_f =9.098 kgf/mm^2; t_b =0.441 kgf/mm^2; $\bar{y}$ =46.918 kgf/mm^2; *E*=20,400 kgf/mm^2; *R*= 292.1 mm and *P*= 875.31tf (1754,35 kgf/mm). To obtain the data sets for ANN training, the parameters variations were chosen as: $h_i = \pm 8\%$; $h_o = \pm 3\%$; $\mu = \pm 20\%$; $t_f = \pm 30\%$; $t_b = \pm 30\%$; $\bar{y} = \pm 10\%$. Three different values were chosen for each parameter resulting in 729 training sets.

Generally, the largest effort to get a neural network trained lies on collecting and pre-processing neural network input data. The pre-processing operation consists in the data normalization in such away that the inputs and outputs values be within the 0 to 1 range. The following procedure was adopted to normalize the input data before using it in the ANN structure:

- In order to improve convergence of the NN training process, the normalization interval [0 ... 1] was reduced to [0.2 ... 0.8].
- The data was normalized and non-normalized through the following expressions:

$$f^a(Lo) = Ln = (Lo - Lmín) / (L\max - L\min) \tag{10a}$$

$$f^b(Ln) = Lo = Ln * L\max + (1 - Ln) * Lmín \tag{10b}$$

where L*n* is the normalized value, L*o* the value to normalize, L*min* and L*max* are minimum and maximum variable values, respectively.

- L*min* and L*max* are computed as:

L*mín* = (4 x LimiteInf. - LimiteSup) / 3 (11a)

L*máx* = (LimiteInf. - 0,8 x L*mín*) / 0,2 (11b)

The Eq. (11a) and Eq. (11b) are obtained considering, in the Eq. (10a), L*n* = 0.2 with L*o* = LimiteInf and L*n*= 0.8 with L*o*= LimiteSup, respectively. Where LimiteInf and LimiteSup are minimum and maximum values respectively of the original data sets. After 555000 iterations a average global error of 0,04 was reached. The final weights for the hidden and output layers with its polarization weights are:

$$W^h = \begin{bmatrix} -2.3472 & 0.7327 & -1.6876 & -0.0036 & 0.6561 & -2.6032 \\ 3.9055 & 3.0808 & -5.1249 & -2.8544 & -5.3232 & -5.6772 \\ -8.4711 & 5.1593 & 6.1539 & -1.5899 & -4.3587 & 3.4279 \\ -11.7598 & 0.2278 & -0.9994 & -0.3439 & 0.1404 & 6.6674 \\ -8.4249 & -9.2360 & -5.5525 & 1.1392 & 2.4098 & -5.1437 \\ -1.3988 & 1.6645 & 9.0436 & 0.0068 & 5.0951 & -7.2585 \\ 3.4416 & 0.4525 & 8.6002 & -6.6928 & 5.2036 & 1.8066 \\ -2.0861 & 0.6307 & -0.6206 & 0.0006 & 0.8530 & -1.8555 \\ -4.1920 & 0.5245 & -4.9311 & 7.3975 & 1.3536 & 6.6874 \\ 10.5998 & 3.1857 & -4.8274 & 1.8625 & -3.3112 & -2.8462 \\ 3.8174 & 13.3982 & -3.9206 & -1.5036 & -1.7811 & 0.6947 \\ 3.9012 & -2.6002 & 0.5488 & 2.1261 & -5.7294 & -5.7072 \\ 8.5121 & -6.7925 & -2.3503 & 2.8321 & 4.0080 & -3.0901 \end{bmatrix}$$

$$W^h_{bias} = \begin{bmatrix} 5.2510 \\ 6.0337 \\ -3.8967 \\ 5.1585 \\ 5.4216 \\ -1.0687 \\ -7.2570 \\ 0.2414 \\ -3.2777 \\ -0.9663 \\ -5.8385 \\ 3.7317 \\ -3.0088 \end{bmatrix} \quad W^o = \begin{bmatrix} -3.1772 & -2.9064 \\ -0.0878 & 0.0705 \\ -0.0626 & -0.0884 \\ 0.1237 & 0.0306 \\ 0.0126 & 0.0401 \\ 0.0767 & -0.0340 \\ 0.0418 & -0.0196 \\ -2.4975 & -2.2320 \\ 0.0412 & -0.0437 \\ -0.0568 & 0.0361 \\ 0.0205 & 0.0008 \\ -0.1528 & 0.1381 \\ -0.0468 & 0.0634 \end{bmatrix} \quad W^o_{bias} = \begin{bmatrix} 3.2107 \\ 2.8861 \end{bmatrix}$$

The Figure 1, shows the concentration of the outputs for the rolling load Eq. (7), for which was

chosen $\theta_1 = -0{,}847$ and $\theta_2 = 0$ to adjust Eqs (1) and (2) with $y_1 = 0{,}3$ and $y_2 = 0{,}5$ respectively.

For the first method, through an computational algorithm is possible to identify the Domains and the rules for the entries variables that determines exits as SMALL, MEDIUM or LARGE, Eq (3).

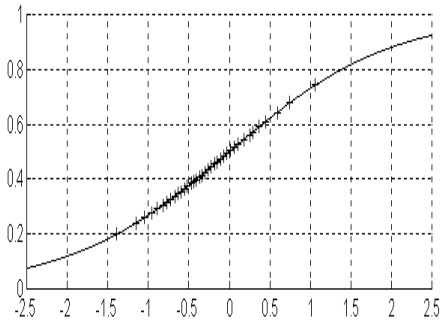

Figure 1. Outputs for the sigmoid function

For the second method, based on the sensitivity factors, is possible to identify the domains and the rules for the entries variables that determine the sensitivity of the rolling load in relation with others entries parameters, Table 1,2 and 3 .

Through the Tables 1,2 and 3 is possible to observe that the cold rolling process is a no-lineal process. Observe: to h_i = 5.00 and h_o = 3.6, nothing can be said in relation to the sensitivity factor $\frac{\delta P}{\delta h_i}$, Is it small?, medium? or large?. The parameters that determine the linguistic value for the sensitivity factor are the intervals of variation of the, μ, $\bar{y}$, t_b and t_f. The value of δ corresponds to small variations around the nominal value of the parameter and this value can be determined through an computational algorithm.

As example considers the following rules (extracted of the Table 1,2 and 3):

Rule 1:

IF $h_i \in [5.0 \pm \delta]$ AND
$h_o \in [3.492 \pm \delta]$ AND
$\mu \in [0.096...0.144]$ AND
$t_b \in [0.309...0.573]$ AND
$t_f \in [6.369...11.827]$ AND
$\bar{y} \in [42.949...52.493]$
THEN $\frac{\delta P}{\delta h_i}$ is *SMALL*

Rule 2:

IF $h_i \in [5.0 \pm \delta]$ AND
$h_o \in [3.492 \pm \delta]$ AND
$\mu \in [0.096 \pm \delta]$ AND
$t_b \in [0.309...0.573]$ AND
$t_f \in [6.369...11.827]$ AND
$\bar{y} \in [47.727...52.493]$
THEN $\frac{\delta P}{\delta h_i}$ is *MEDIUM*

Rule 3:

IF $h_i \in [5.0 \pm \delta]$ AND
$h_o \in [3.492 \pm \delta]$ AND
$\mu \in [0.144 \pm \delta]$ AND
$t_b \in [0.573 \pm \delta]$ AND
$t_f \in [6.369...11.827]$ AND
$\bar{y} \in [42.499...48.920]$
THEN $\frac{\delta P}{\delta h_i}$ is *LARGE*

4 Conclusions

In process where analytic models are difficult to obtain or when the models need numerical solution with great computational effort, as is the case of the cold rolling process, the rules extraction via neural networks previously trained are an alternative to represent complex physical processes.

In this work, two methods for extraction of rules from neural networks previously trained were presented. Those rules are based in the domains of the variables of the physical process for which linguistic values are considered for the variables linguistic load (*P*) and for the sensitivity factor ($\frac{\delta P}{\delta h_i}$). Those rules represent the behavior of the process and they can be used to reinforce the action of a less experienced operator when disturbances happen in the process. The obtained rules can be used as the base of knowledge of a expert system.

A limitation of the rules extraction methods via ANN is the need to train again the net for a new data set. A solution is to train the nets with great capacity for the generalization. On the other hand, with ANN correctly trained is possible to obtain the sensitivity factors for infinite operation points, inside of the data set for which ANN was trained. Remind that the obtained sensibility factors of physical models can present analytic or numeric complexities making unfeasible the use in real time process.

Another limitation is the unknown value of δ, that vary depending on the parameter considered. The values of can be calculate through a computational algorithm, and that is one of the objectives of the next works.

Table 1 Rules based on sensitivity factors to output SMALL

$h_i \in$	[5.0 $\pm\delta$]							
$h_o \in$	[3.492 $\pm\delta$]	[3.6 $\pm\delta$]				[3.708 $\pm\delta$]		
$\mu \in$	[0.096...0.144]	[0.096 $\pm\delta$]		[0.120 $\pm\delta$]		[0.096 $\pm\delta$]	[0.120 $\pm\delta$]	[0.144 $\pm\delta$]
$t_b \in$	[0.309...0.573]	[0.441 $\pm\delta$]	[0.573 $\pm\delta$]	[0.309 $\pm\delta$]	[0.441...0.573]	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]
$t_f \in$	[6.369...11.827]	[9.098...11.827]	[6.369...11.827]	[9.098...11.827]	[6.369...11.827]	[6.369...11.827]	[9.098...11.827]	[11.827 $\pm\delta$]
$\overline{y} \in$	[42.949...52.493]	[42.226...46.918]	[42.226...46.918]	[42.226 $\pm\delta$]	[42.226...46.918]	[41.487 $\pm\delta$]	[41.487 $\pm\delta$]	[41.487 $\pm\delta$]

Table 2 Rules based on sensitivity factors to output MEDIUM

$h_i \in$	[5.0 $\pm\delta$]						
$h_o \in$	[3.492 $\pm\delta$]		[3.6 $\pm\delta$]			[3.708 $\pm\delta$]	
$\mu \in$	[0.096 $\pm\delta$]	[0.144 $\pm\delta$]	[0.096 $\pm\delta$]	[0.120 $\pm\delta$]	[0.144 $\pm\delta$]	[0.096 $\pm\delta$]	[0.120...0.144]
$t_b \in$	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]	[0.309...0.573]
$t_f \in$	[6.369...11.827]	[6.369...11.827]	[6.369...11.827]	[6.369...11.827]	[6.369...11.827]	[6.369...11.827]	[9.098...11.827]
$\overline{y} \in$	[47.727...52.493]	[42.49...52.493]	[46.918...51.61]	[42.226...51.61]	[42.226...51.61]	[46.097...50.70]	[41.487...50.70]

Table 3 Rules based on sensitivity factors to output LARGE

$h_i \in$	[5.0 $\pm\delta$]						
$h_o \in$	[3.492 $\pm\delta$]			[3.6 $\pm\delta$]			[3.708 $\pm\delta$]
$\mu \in$	[0.144 $\pm\delta$]			[0.144 $\pm\delta$]			[0.144 $\pm\delta$]
$t_b \in$	[0.309 $\pm\delta$]	[0.441 $\pm\delta$]	[0.573 $\pm\delta$]	[0.309 $\pm\delta$]	[0.441 $\pm\delta$]	[0.573 $\pm\delta$]	[0.309...0.573]
$t_f \in$	[6.369...11.827]	[6.369...11.827]	[6.369...11.827]	[9.098...11.827]	[6.369..11.827]	[6.369 $\pm\delta$]	[6.369 $\pm\delta$]
$\overline{y} \in$	[52.493 $\pm\delta$]	[52.493 $\pm\delta$]	[42.499...52.493]	[51.610 $\pm\delta$]	[50.610 $\pm\delta$]	[51.610 $\pm\delta$]	[43.336...48.15]

5 Acknowledgement

The authors would like to acknowledge the financial support from FIP, Fundo de Incentivo à Pesquisa da Pontifícia Universidade Católica de Minas Gerais Brasil.

References:

[1] D.H. Fisher, and K.B McKusick. An empirical comparison of ID3 and backpropagation. *Proc. of the 11th International Joint Conf. on Artificial Intelligence*, Detroi, USA, 1989, 788-793.

[2] J.W. Shavlik, R.J. Mooney, and G.G. Towell. Symbolic and neural net learning algorithms: An empirical comparision. *Machine Learning*, 6, 1991, 111-143.

[3] G.G. Towell, J.W. Shavlik, and M. Noordewier. Refinement of approximately correct domain theories by knowledge-based neural networks. *Proc. 8th National Conference on Artificial Intelligence*, Boston, USA, 1990, 861-866.

[4] G.G. Towell, G.G. and J.W. Shavlik, Extracting Refined Rules from Knowledge-Based Neural Networks, *Joural Machine Learning*, 1992.

[5] M.W. Craven and J.W. Shavlik. Extracting Comprehensible Concept Representations from Trained Neural Networks", *IJCAI'95*, 1995,1-15.

[6] L.E. Zárate, Sensitivity Analysis of Cause-Effect Relationship Using Neural Networks, *16th International Conference on CAD/CAM, Robotics and Factories of the Future*, St. Augustine, Trinidade W.I, 2000.

[7] L.E. Zárate, Rules Based Qualitative Analysis in Cold Rolling Process, *16th International Conference on CAD/CAM, Robotics and Factories of the Future*, St. Augustine, Trinidade W.I, 2000.