Mathematical and Computer Modelling 52 (2010) 1169–1176

Contents lists available at ScienceDirect

Mathematical and Computer Modelling

journal homepage: www.elsevier.com/locate/mcm

The use of design of experiments to improve a neural network model in order to predict the thickness of the chromium layer in a hard chromium plating process

F. Sánchez Lasheras [a], J.A. Vilán Vilán [b], P.J. García Nieto [c,*], J.J. del Coz Díaz [d]

[a] Research Department, Tecniproject SL, C/ Marqués de Pidal 7, 33004 Oviedo, Spain
[b] Department of Mechanical Engineering, University of Vigo, 36310 Vigo, Spain
[c] Department of Mathematics, University of Oviedo, 33007 Oviedo, Spain
[d] Department of Construction, University of Oviedo, 33204 Gijón, Spain

ARTICLE INFO

Article history:
Received 24 September 2009
Accepted 8 March 2010

Keywords:
Hard chromium plating process
Statistical optimization
Machine learning
Neural networks
Design of experiments (DOE)
Response surface methodology

ABSTRACT

The hard chromium plating process aims at creating a coating of hard and wear-resistant chromium with a thickness of some micrometres directly on the metal part without the insertion of copper or nickel layers. Chromium plating features high levels of hardness and resistance to wear and it is due to these properties that they can be applied in a huge range of sectors. Resistance to corrosion of a hard chromium plate depends on the thickness of its coating, and its adherence and micro-fissures. This micro-fissured structure is what provides the optimal hardness of the layers. The hard chromium plating process is one of the most effective ways of protecting the base material against a hostile environment or improving the surface properties of the base material. However, in the electroplating industry, electroplaters are faced with many problems and undesirable results with chromium plated materials. Common problems faced in the electroplating industry include matt deposition, milky white chromium deposition, rough or sandy chromium deposition and insufficient thickness and hardness. This article presents an artificial neural network (ANN) model to predict the thickness of the layer in a hard chromium plating process. The optimization of the ANN was performed by means of the design of experiments theory (DOE). In the present work the purpose of using DOE is twofold: to define the optimal experiments which maximize the ratio of the model accuracy, and to minimize the number of necessary experiments (ANN models trained and validated).

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

On many occasions, the properties of the materials may be insufficient to meet some of the functional demands; especially those arising from environmental conditions such as a very corrosive atmosphere or an excessive wear condition. As the high quality materials may turn out to be too costly, designers should seek a solution which will either protect the base material against hostile environment or improve the properties of the base material only at the surface [1,2].

Chromium plating operations are categorized based upon the thickness of the chromium metal layer applied. They can be classified as "*hard chromium*" and "*bright* (decorative) *chromium*" plating [3–6]. In decorative plating, a layer of nickel is first plated over a metal substrate. Following this step, a thin layer of chromium is deposited over the nickel layer to provide

* Corresponding author. Tel.: +34 985 103417; fax: +34 985 103354.
E-mail address: lato@orion.ciencias.uniovi.es (P.J. García Nieto).

0895-7177/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.mcm.2010.03.007

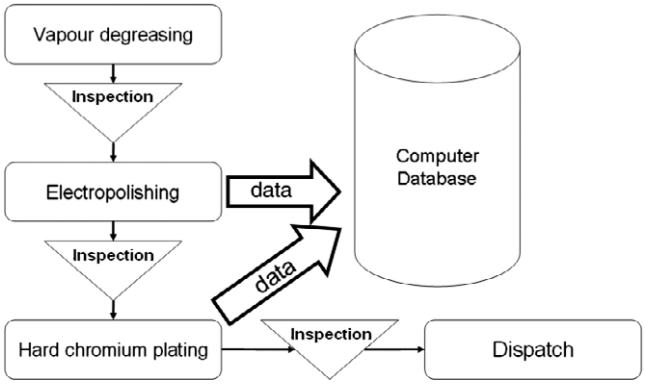

Fig. 1. The industrial process of hard chromium plating.

a decorative and protective finish: for instance, on faucets and automotive wheels. The description "hard chromium" as opposed to "bright chromium" is in a sense erroneous, since hard chromium is no way harder than bright chromium. High hardness of the layer is a consequence only of the greater thickness of the deposit. In itself, bright chromium may indeed be harder than some hard chromium although this hardness cannot be made use of in practice owing to the thin deposit. Therefore, for bright (decorative) chromium plating, the most important property is the perfect brightness of the deposit, while for hard chromium plating [7], the most important property is the thickness of the deposit, since thicker deposit gives greater hardness.

The hard chromium plating process under study in the present work is widely used on many mechanical parts and plastic moulds due to its good mechanical properties, good aesthetic appearance and superior resistance to corrosion. Although it is well known that hard chromium plating is a process that has serious disadvantages from an environmental point of view, its replacement is not a simple matter due to the great performance of the chromium plated pieces [8]. This process consists mainly of three operations (see Fig. 1), together with some intermediate quality inspections [9–12,6]:

- Vapour degreasing: This is mainly a cleaning operation. It is performed over the piece in order to assure the cleanliness of the surface.
- Electropolishing: It is an electrochemical process that removes material from the work piece. The performance of this operation before the hard chromium plating helps to ensure a good roughness of the surface that will be coated.
- Hard chromium plating: This is the operation on which a thin layer of chromium is deposited onto the workpiece.

2. The aim of the present work

The aim of the present work is to create an artificial neural network (ANN) predictive model [13–17] capable of predicting the thickness of the chromium layer deposited over the pieces taking into account variables related not only with the hard chromium plating operation but also with the previous process of electropolishing. In this work, the design of experiments (DOE) [18–20] based on statistical principles has been used in order to improve the root mean square error (RMSE) of the model obtained trough changes in some of the parameters of the neural network. It must be remarked that the applications of DOE techniques [21] to optimize ANNs parameter have already been reported in literature [22–26].

The model resulting from this work may help to eliminate the problems faced by electroplaters and provide detailed information about the interaction between process parameters in an optimized neural network model. Thus, the values of process parameters can be easily arranged according to the desired features and application fields.

3. Mathematical model

3.1. Multilayer perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate output. It is a modification of the standard linear perceptron in that it uses three or more layers of neurons (nodes) with *nonlinear activation functions*, and is more powerful than the perceptron in that it can distinguish data that is not linearly separable, or separable by a hyperplane [14].

If a multilayer perceptron consists of a linear activation function in all neurons, that is, a simple on–off mechanism to determine whether or not a neuron fires, then it can easily be proved with linear algebra that any number of layers can be reduced to the standard two-layer input–output model. What makes a multilayer perceptron different is that each neuron uses a *nonlinear activation function* which was developed to model the frequency of action potentials, or firing, of biological neurons in the brain. This function is modelled in several ways, but must always be *normalizable* and *differentiable*.

The two main activation functions used in current applications are the hyperbolic tangent function (tansig) and the sigmoid function, respectively. They can be described from the mathematical point of view as follows [14,17,27]:

$$\phi(v_i) = \tanh(v_i) \quad \text{and} \quad \phi(v_i) = \left(1 + e^{-v_i}\right)^{-1} \tag{1}$$

in which the former function is a hyperbolic tangent which ranges from −1 to 1, and the latter is equivalent in shape but ranges from 0 to 1. Here y_i is the output of the ith node (neuron) and v_i is the weighted sum of the input synapses. More specialized activation functions include radial basis functions which are used in another class of supervised neural network models.

The multilayer perceptron consists of one input and one output layer with one or more hidden layers of nonlinearly activating nodes. Each node in one layer connects with a certain weight w_{ij} to every other node in the following layer. Learning occurs in the perceptron by changing connection weights (or synaptic weights) after each piece of data is processed, based on the amount of error in the output compared to the expected result. This is an example of supervised learning, and it is performed through *backpropagation*, a generalization of the least mean squares algorithm in the linear perceptron. We represent the error in output node j in the nth data point by $e_j(n) = d_j(n) - y_j(n)$, where d is the target value and y is the value produced by the perceptron. We then make corrections to the weights of the nodes based on those corrections which minimize the energy of error in the entire output, given by [14,17]:

$$\varepsilon(n) = \frac{1}{2}\sum_j e_j^2(n). \tag{2}$$

By the theory of differentials, we find our change in each weight to be:

$$\Delta w_{ij}(n) = -\eta \frac{\partial \varepsilon(n)}{\partial v_j(n)} y_i(n) \tag{3}$$

where y_i is the output of the previous neuron and η is the learning rate, which is carefully selected to ensure that the weights converge to a response that is neither too specific nor too general. In programming applications, it typically ranges from 0.2 to 0.8. The derivative to be calculated depends on the input synapse sum v_j, which itself varies. It is easy to prove that for an output node this derivative can be simplified to [14,28]:

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = e_j(n)\,\phi'\left(v_j(n)\right) \tag{4}$$

where ϕ' is the derivative of the activation function described above, which itself does not vary. The analysis is more difficult because of the change in weights to a hidden node, but it can be shown that the relevant derivative is:

$$-\frac{\partial \varepsilon(n)}{\partial v_j(n)} = \phi'\left(v_j(n)\right)\sum_k -\frac{\partial \varepsilon(n)}{\partial v_k(n)} w_{kj}(n). \tag{5}$$

Note that this depends on the change in weights of the kth nodes, which represent the output layer. Therefore, to change the hidden layer weights, we must first change the output layer weights according to the derivative of the activation function, and so this algorithm represents a *backpropagation* of the activation function.

Multilayer perceptrons using a backpropagation algorithm are the standard algorithm for any supervised-learning pattern recognition process and the subject of ongoing research in computational neuroscience and parallel distributed processing.

3.2. Numerical simulation based on design of experiments (DOE)

In order to study the influence of some neural networks parameters on the RMSE of the predictive model of the thickness of hard chromium layer deposited over the piece, a design optimization based on design of experiments (DOE) was carried out [29–31].

In a design optimization based on DOE (used in the deterministic method), each change of the value of any input variable requires a new multilayer perceptron analysis. A response surface is generated that is an explicit approximation function of the RMSE of the multilayer perceptron results expressed as a function of all the selected input variables. The DOE method generates a response surface using curve and surface fitting algorithms to "fit" output data as a function of input data. This requires a group of design points where each point is generated via the training of a specific neural network model.

To determine the response surfaces, it is necessary to evaluate higher-order derivatives of the MLP results with respect to the selected input variables, where the order of the derivatives corresponds to the order of the approximation function. This method can be applied in design problems for both linear and nonlinear analysis [15,16,28,26].

The most common empirical models fitted to the experimental data take either a linear form or quadratic form. For instance, a linear model with two factors (or input parameters), X_1 and X_2, can be written as [30]:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_{12} X_1 X_2 + \text{experimental error}. \tag{6}$$

Table 1
Input variables for the neural network model.

Input variables
Iron content in the electropolishing bath
Electropolishing time
Electropolishing temperature
Thickness removed by electropolishing
Chromic acid content hard chromium bath
Hard chromium time
Hard chromium temperature

Table 2
Variable parameters and their levels.

Variables	Values
Number of neurons intermediate layer	13, 15 and 17
Momentum	0.3, 0.5 and 0.7
Learning rate	10^{-2}, 10^{-3} and 10^{-4}
Activation function	Tansig and sigmoid functions

Here, Y is the response (or output parameter) for given levels of the main effects X_1 and X_2 and the X_1X_2 term is included to account for a possible interaction effect between X_1 and X_2. The constant β_0 is the response of Y when both main effects are 0. For a more complicated example, a linear model with three factors X_1, X_2, X_3 and one response, Y, would look like (if all possible terms were included in the model) [29,30]:

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + \beta_3X_3 + \beta_{12}X_1X_2 + \beta_{13}X_1X_3 + \beta_{23}X_2X_3 + \beta_{123}X_1X_2X_3 + \text{experimental error}. \quad (7)$$

The three terms with single "$X's$" are the *main effects* terms. There are $k\,(k-1)\,/2 = 3 \times 2/2 = 3$ *two-way interaction* terms and 1 *three-way interaction* term (which is often omitted, for simplicity). When the experimental data are analysed, all the unknown "β" parameters are estimated and the coefficients of the "X" terms are tested to see which ones are significantly different from 0.

In our case, a second-order (or quadratic) model is used. This model is typically employed in response surface DOEs with suspected curvature, and it does not include the three-way interaction term but adds three more terms to the linear model, namely [29,30]:

$$\beta_{11}X_1^2 + \beta_{22}X_2^2 + \beta_{33}X_3^2. \quad (8)$$

Therefore, the entire model analysed here includes many cross-product (or interaction) terms involving squared $X's$, as may be observed in the all response surfaces.

4. Experimental design

The underlying goal of this study is to improve the RMSE of a neural network (multilayer perceptron) model using DOE methodology. This neural network topology consists of seven input variables listed in Table 1. The output variable is the thickness of the hard chromium layer deposited over the piece. This model was performed with the package AMORE [32] of the free software environment for statistical computing called R [33].

The topology of the models consisted of an input layer, a hidden layer and an output layer. The following parameters were fixed for all the neural networks trained:

- *Error criterion*: The criterion used to measure to proximity of the neural network prediction to its target: "LMS" (Least mean squares).
- *Output layer*: The activation function of the output layer is: "purelin".
- *Training method*: "ADAPTgd": Adaptive gradient descent.

Other parameters were varied in the models in order to perform the DOE analysis. Those parameters are listed below and their levels are in Table 2:

- N: The number of neurons in the intermediate layer.
- *Learning rate*: Than controls the step size when weights are iteratively adjusted.
- *Momentum*: It allows a change to the weights to persist for a number of adjustment cycles. This can improve the learning rate in some situations.
- *Hidden layer*: The activation function of the hidden layer.

It must be remarked that our DOE has three variables that use three levels while the other (activation function) has only two possible values. The addition of central points allowed us to detect nonlinearity in the responses. Each of the runs of the experiment was repeated five times, using 800 different samples chosen at random in each testing. After that, the validation of the model and the calculation of the RMSE were performed with five different validation data sets composed by 200 samples.

Table 3
Analysis of variance for RMSE, using the sequential sum of squares (Seq. SS), the adjusted sum of squares (Adj. SS), the adjusted mean squares (Adj. MS), ANOVA test's statistic *F* and *p*-value for tests in case of an intermediate layer's sigmoid-type activation function.

Term	DF	Seq. SS	Adj. SS	Adj. MS	*F*	*p*
Neurons	2	0.0003177	0.0003177	0.0001589	9.26	0.008
Learning rate	2	0.0588894	0.0588894	0.0294447	1716.43	0
Momentum	2	0.0024243	0.0024243	0.0012122	70.66	0
Neurons ∗ Learning rate	4	0.0005014	0.0005014	0.0001254	7.31	0.009
Neurons ∗ Momentum	4	0.0000669	0.0000669	0.0000167	0.97	0.472
Learning rate ∗ Momentum	4	0.0058578	0.0058578	0.0014644	85.37	0
Error	8	0.0001372	0.0001372	0.0000172		
Total	26	0.0681947				

Table 4
Analysis of variance for RMSE, using the sequential sum of squares (Seq. SS), the adjusted sum of squares (Adj. SS), the adjusted mean squares (Adj. MS), ANOVA test's statistic *F* and *p*-value for tests in case of an intermediate layer's hyperbolic tangent-type (tansig) activation function.

Term	DF	Seq. SS	Adj. SS	Adj. MS	*F*	*p*
Neurons	2	0.0002206	0.0002206	0.0001103	0.91	0.442
Learning rate	2	0.0220476	0.0220476	0.0110238	90.53	0
Momentum	2	0.0178079	0.0178079	0.0089039	73.12	0
Neurons ∗ Learning rate	4	0.0004346	0.0004346	0.0001087	0.89	0.511
Neurons ∗ Momentum	4	0.0004596	0.0004596	0.0001149	0.94	0.486
Learning rate ∗ Momentum	4	0.0357647	0.0357647	0.0089412	73.43	0
Error	8	0.0009741	0.0009741	0.0001218		
Total	26	0.0777092				

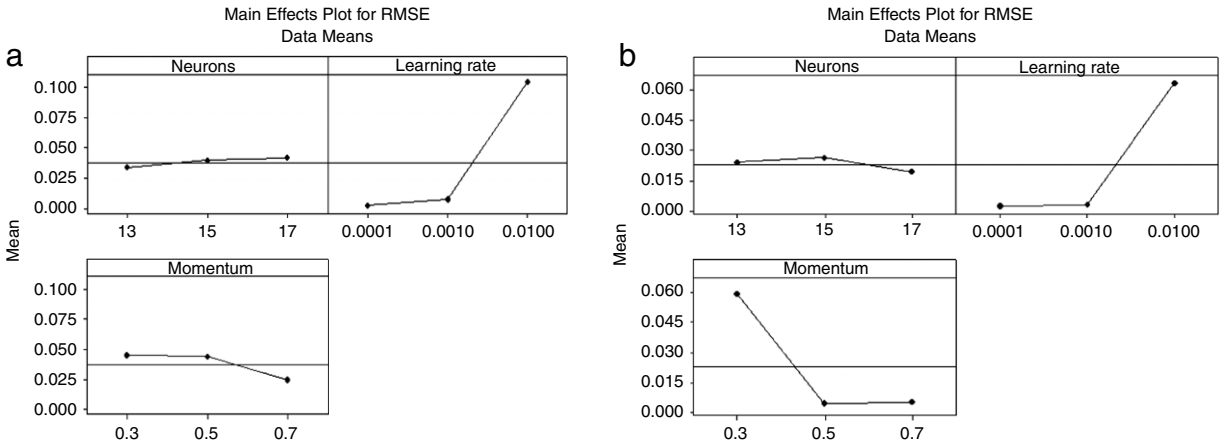

Fig. 2. Main effects plots for RMSE taking into account as the activation function: (a) the sigmoid function; and (b) the tansig function.

5. Results

Analysis of variance (ANOVA) was carried out based on the unseen testing data at 95% confidence level. The analysis was carried out by means of the MINITAB software [34]. Tables 3 and 4 show the estimated effects of each parameter and coefficient. The parameters used in these previous tables are those defined in Table 2. Terms like Neurons ∗ Learning rate are the interaction between the neurons and the learning rate. In this analysis, only two way interaction was considered. Tables 3 and 4 display both the sequential sums of squares (Seq. SS) and adjusted sums of squares (Adj. SS). The *p*-value was used to determine which of the effects are significant. In this study 95% confidence was used therefore terms that have *p*-value lower than 0.05 are significant. In this case, for the sigmoid activation function in the intermediate layer the number of neurons, the learning rate, the momentum and the interaction between neurons and learning rate and learning rate and momentum are significant. For the tansig function in the intermediate layer, only the learning rate, momentum and the interaction between learning rate and momentum were significant.

Fig. 2 shows the main effect plots of RMSE. This graph indicates that for the sigmoid activation function in the intermediate layer, the number of neurons in hidden layer at low level results in lower RMSE, the same thing happening for learning rate while the minimum values of RMSE are reached for the highest levels of momentum. This plot is similar for the tansig activation function in the intermediate layer, except for the number of neurons as the minimum value of the RMSE is

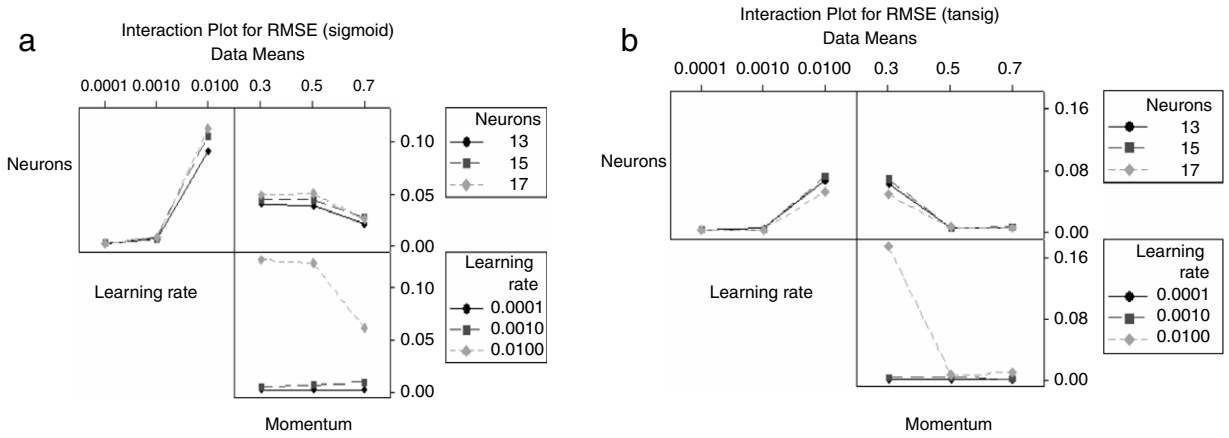

Fig. 3. Interaction plot for RMSE taking into account as the activation function: (a) the sigmoid function; and (b) the tansig function.

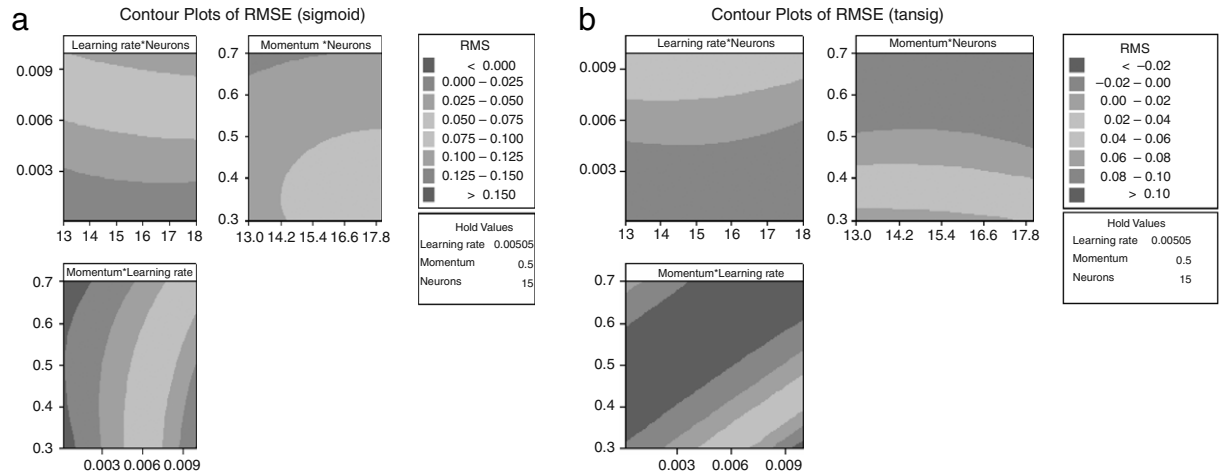

Fig. 4. Contour plots of RMSE taking into account as the activation function: (a) the sigmoid function; and (b) the tansig function.

Table 5
Optimum values of the variables.

Variables	Optimum values
Number of neurons intermediate layer	13
Momentum	0.501
Learning rate	10^{-4}
Activation function	Sigmoid function

for the maximum number of neurons. It must be remarked that as in Tables 3 and 4, some interactions among parameters are significant. Therefore, the best setting cannot be determined from the main effect plot and the interaction plot (see Fig. 3) has to be taken into account.

Fig. 3 shows the two-factor interaction plots among parameters. For example, the upper left graph in Fig. 3(a) shows the interaction between the number of neurons in the hidden layer and the learning rate. As both factors have three levels each, the average of all momentum terms were calculated and plotted for the different number of neurons. Graphs in Fig. 3 are very useful for interpreting significant interactions. However, it should not be used as the only means of finding the best setting of factors, as the interpretation is subjective.

The analysis through contour plots of the RMSE is performed in Fig. 4 where variables are represented two by two in different graphs. This kind of chart allows us to estimate interaction and even quadratic effects, and therefore gives us an idea of the (local) shape of the response surface that we are investigating. The analysis of this graph allowed us to determinate that the minimum RMSE was achieved for the variables values listed in Table 5.

The results from the optimized network can be used to calculate an accurate value of the chromium layer obtained from the hard chromium plating industrial process. The results obtained after the training and validation of the neural

network proposed with the parameters listed in Table 5 improved the accuracy of the results obtained with the previously-trained neural network models. While the best previously-tested model had a RMSE of 0.002325144, the new model with the optimum values of variables listed in Table 5 reduced this value to 0.00218176.

6. Discussion and conclusions

The multilayer perceptron (MLP) in combination with the design of experiments (DOE) has been shown as a suitable tool in the modelling and analysis of singular problems, such as the prediction of the thickness of the chromium layer in a hard chromium plating process. This approach afford us an opportunity to fully understand the influence of process parameters on thickness and/or brightness by performing significantly fewer experiments than if only one factor at a time has been evaluated. A good understanding was obtained of how interactions among the various factors influence the thickness of the chromium layer was obtained. Again, this information would not have been obtained if one factor at a time had been investigated. Accurate relationships, describing the effects of the various factors on thickness were obtained by using the design of experiments (DOE) [35].

The MLP plus DOE [24,26,35] reproduces quite accurately the mechanism of the prediction of the thickness of the chromium layer in a hard chromium plating process. In this sense, the key step in engineering analysis is therefore choosing appropriate mathematical models. These models will clearly be selected depending on what phenomena are to be predicted, and it is most important to select mathematical models that are *reliable* and *effective* in predicting the quantities sought.

Finally it must be remarked that this paper has described techniques for the training of ANNs by using DOE. The result suggested that DOE approach could be successfully used to optimize back-propagation ANNs parameters. The model developed after the use of the design of experiments (DOE) methodology improved the RMSE value of previous models checked by the authors. Therefore, authors have developed a new methodology to solve complex industrial problems with success using ANNs in combination with the DOE technique such as the prediction of the thickness of the chromium layer in a hard chromium plating process.

Acknowledgements

The authors wish to acknowledge the computational supports provided by Tecniproject Ltd and the Departments of Mathematics and Construction at University of Oviedo. We would like to thank Anthony Ashworth for his revision of the English grammar and spelling of the manuscript.

References

[1] C.A. Huang, K.C. Li, W. Lin, M.C. Liao, The behavior of electroplated hard-chromium on Cr–Mo steel subject to long-term annealing at 250 °C, Materials Science & Engineering A 403 (2005) 222–226.
[2] C.A. Huang, W. Lin, M.J. Liao, The electrochemical behavior of the bright chromium deposits plated with direct and pulse currents in 1 M H_2SO_4, Corrosion Science 48 (2006) 460–471.
[3] J.K. Dennis, T.E. Such, The Nickel and Chromium Plating, Woodhead Publishing, New York, 1994.
[4] R.K. Guffie, The Handbook of Chromium Plating, Gardner Publications Ltd., New York, 1986.
[5] J.H. Lindsay, Decorative and hard chromium plating, Plating and Surface Finishing 90 (2003) 22–24.
[6] M. Schlesinger, M. Paunovich, Modern Electroplating, Wiley-Interscience, New York, 2000.
[7] M. Bayramoglu, B. Onat, N. Geren, Statistical optimization of process parameters to obtain maximum thickness and brightness in chromium plating, Journal of Materials Processing Technology 203 (2008) 277–286.
[8] G.A. Lausmann, Electrolytically deposited hard chrome, Surface and Coatings Technology 86–87 (1996) 814–820.
[9] T.H. Irvine, The Chemical Analysis of Electroplating Solutions, Chemical Publishing Company, New York, 2000.
[10] N. Kanani, Electroplating: Basic Principles, Processes and Practice, Elsevier Science, Amsterdam, 2005.
[11] T. Ölmez, The optimization of Cr(VI) reduction and removal by electrocoagulation using response surface methodology, Journal of Hazardous Materials 162 (2009) 1371–1378.
[12] M.J. Ortiz, Effect of a thin hard chromium coating on fatigue behavior of 4140 steel, Surface Engineering 20 (2004) 345–352.
[13] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, New York, 2007.
[14] K. Gurney, An Introduction to Neural Networks, CRC Press, Boca Ratón, 1997.
[15] J. Han, M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, New York, 2005.
[16] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, New York, 2009.
[17] S. Haykin, Neural Networks and Learning Machines, Prentice Hall, Englewood Cliffs, New Jersey, 2008.
[18] A. Ait Yala, A. Megueni, Optimization of composite patches repairs with the design of experiments method, Materials and Design 30 (2009) 200–205.
[19] D. Andina, D.T. Pham, Computational Intelligence for Engineering and Manufacturing, Springer, New York, 2007.
[20] J. Antony, Design of Experiments for Engineers and Scientists, Butterworth-Heinemann, New York, 2003.
[21] C.-P. Chen, M.-T. Chuang, Y.-H. Hsiao, Y.-K. Yang, C.-H. Tsai, Simulation and experimental study in determining injection molding process parameters for thin-shell plastic parts via design of experiments analysis, Expert Systems with Applications 36 (2009) 10752–10759.
[22] J.F.C. Khaw, B.S. Lim, L.E.N. Lim, Optimal design of neural networks using the Taguchi method, Neural Computing 7 (1995) 225–245.
[23] Y.S. Kim, B.J. Yum, Robust design of multilayer feed forward neural networks: an experimental approach, Engineering Applications of Artificial Intelligence 17 (2004) 249–263.
[24] M.S. Packianather, P.R. Drake, H. Rowland, Optimizing the parameters of multilayered feed forward neural networks through Taguchi design of experiments, Quality and Reliability Engineering International 16 (2000) 461–473.
[25] W. Sukthomya, J.D.T. Tannock, The optimisation of neural network parameters using Taguchi design of experiments approach: an application in manufacturing process modelling, Neural Computing and Applications 14 (3) (2005) 337–344.
[26] S.M. Yang, G.S. Lee, Neural network design by using Taguchi method, Journal of Dynamic Systems, Measurement, and Control 121 (1999) 560–563.
[27] P.-N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, New York, 2005.
[28] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, New York, 2005.

[29] A.M. Dean, D. Voss, Design and Analysis of Experiments, Springer, New York, 2000.
[30] K. Hinkelmann, O. Kempthorne, Design and Analysis of Experiments: Introduction to Experimental Design, Wiley-Interscience, New York, 2007.
[31] N. Strigul, H. Dette, V.B. Melas, A practical guide for optimal designs of experiments in the Monod model, Environmental Modelling & Software 24 (2009) 1019–1026.
[32] M. Castejón Limas, B. Ordieres Meré, E.P. Vergara González, F.J. Martínez de Pisón Ascacibar, A.V. Pernía Espinoza, F. Alba Elías, Package AMORE: a more flexible neural network package, University of León, Spain, 2007.
[33] J.M. Chambers, Software for Data Analysis: Programming with R, Springer, New York, 2009.
[34] B.F. Ryan, B.L. Joiner, J.D. Cryer, MINITAB Handbook, Brooks/Cole, Thomson Learning Inc., New York, 2005.
[35] W.J. Diamond, Practical Experimental Designs: For Engineers and Scientists, Wiley, New York, 2001.