Iterative RBF Neural Networks as Metamodels of Stochastic Simulations

George Meghabghab*, George Nasr**

* Valdosta State University, Department of Math & Computer Science,
Valdosta, GA 31698
Email: gmeghab@valdosta.edu
** Lebanese American University, Department of Electrical and Computer Engineering,
PO Box 36, Byblos, Lebanon, 13-5053.
Email: genasr@lau.edu.lb

ABSTRACT

Research into emerging technological approaches to make computer simulations more effective and efficient is an essential ingredient to developing successful manufacturing models. This study is a premiere study in using neural networks in metamodeling stochastic simulation in manufacturing domain. A new iterative RBF neural network was developed rather than the baseline ANN models which were used in stochastic simulation metamodeling in domains such as combat simulations in the military, service industries, and transportation companies. Given the fact that typical stochastic simulation metamodeling approaches involves the use of regression models in response surface methods, RBF become a natural target for such an attempt because they use a family of surfaces each of which naturally divides an input space into 2 regions and the n patterns will be assigned either class X+ or X-. This dichotomy of the points is said to be separable with respect to the family of surfaces if there exists a surface in the family that separates the points in the class X+ from those in the class X-. In fact, for the evaluation of the quality of a ball steel, RBF metamodel trained on 1521 training examples from a set of 13000 different simulation runs and was able to outperform direct simulation on 120 additional test examples which were not included in the training set.

INTRODUCTION

Computer simulations are widely used in a variety of applications including the military, service industries, manufacturing companies, nuclear power plants, and transportation organizations. For example in nuclear power plants, often computer simulations are used to train personnel on failures and normal operation, study operation plans, support testing of the heart of the nuclear power plant, as well as to evaluate and compare future design plan changes. Other techniques that are used to examine systems in general do not have the advantages that computer simulations bring mainly that computer simulations provide cheaper and realistic results than other approaches do. In some cases, computer simulation is the only means to examine a system like in nuclear power plants since it is too dangerous to bring a system under such failure conditions to study it closely or costly and infrequent like in combat situations to experiment with the system. Also computer simulation permit studying systems over large periods of time, learn from real world past experiences, and have control over experimental conditions.

Actual manufacturing simulation models are expensive to develop and use, in terms of personnel, time and resources. Large memory requirements, slow response time can prevent companies from considering it as a useful tool. The need to develop manufacturing simulations models that can be used in training that are as realistic as possible is the issue, and speed is not important, while in testing speed and reproducibility become important, incite us to make the different internal simulation modules as efficient and accurate as possible.

Computer simulations have provided companies with the description of the input settings that are needed to produce the optimal best output value for a given set of inputs in a specific domain of study. Response

0-7803-5489-3/99/$10.00 ©1999 IEEE.

surface methodologies using regression models approximations of the computer simulation were the means to achieve computer simulation optimization. As Myers, Khuri, and Carter (1989) stated it in Technometrics [1] that: 'There is a need to develop non-parametric techniques in response surface methodologies. The use of model-free techniques would avoid the assumption of model accuracy or low-level polynomial approximations and in, particular, the imposed symmetry, associated with a second degree polynomial". One possible non-parametric approach is to use artificial neural networks.

ANNs IN MANUFACTURING AND STEEL PRODUCTION

An Artificial Neural Network (ANN) learns to imitate the behavior of a complex function by being presented with examples of the inputs and outputs of the function. This operation is called training. A successfully trained ANN can predict the correct output of a function when presented with the input (or inputs for a multivariate function). ANNs have found wide applications in a variety of fields including mining and manufacturing.

Because there are many steel production processes that are complex and uncertain when modeling for control is concerned, there are examples of ANN being utilized in the steel industry. Neural network controllers have been developed for electric arc furnaces [2], for a continuous casting process [3], and the modeling of a quality steel production with an adaptive logic network [4]. In a totally different and new perspective, ANN were used to determine the surface glossiness of steel sheets as an evaluation method [5], while Lusiak and Pietrzyk [6] used ANN as a history dependent constitutive model for hot forming of steels. In the next section, we will examine the role of ANN in approximating computer simulations in a manufacturing domain mainly the evaluation of a quality steel production.

ARTIFICIAL NEURAL NETWORK METAMODEL APPROACH

A metamodel is a model of a model ([7]). Typical simulation metamodeling approaches involve the use of regression models in response surface methods. A recent overview of published research on simulation metamodels can be found in [8]. A few attempts have been made to employ neural networks as the metamodeling technique. Using an ANN to model a stochastic simulation was done [9], [10], [11], and [12]. Each of these researchers was successful in using ANN as metamodels of stochastic computer simulations. The common feature of these models was an ANN baseline which involves using a backpropagation trained, multi-layer ANN to learn the relationship between the simulation inputs and outputs. The baseline ANN metamodel approach was developed on a (s,S) inventory computer simulation and also was applied to a larger application in the domain under consideration.

RBF has been developed now for a number of years. There is a resurgence in using RBF as a viable architecture to implement neural network solution to many problems. RBF neural networks are deterministic global non-linear minimization methods. These methods detect subregions not containing the global minimum and exclude them from further consideration. In general, this approach is useful for problems requiring solution with guaranteed accuracy. These are computationally very expensive. The mathematical basis for RBF networks is provided by Cover's Theorem [13] which states that a nonlinearly-separable pattern classification problem in high-dimension space is more likely to be linearly-separable than in low-dimensional space. This is the reason for choosing a high dimension for the hidden layer in the network. RBF uses a curve-fitting scheme to learn, i.e., learning is equivalent to finding a surface in a multi-dimensional space that represents a best fit for the training data ([14], [15]).

The approach considered here is a generalized RBF neural network where the number of nodes at the hidden layer is M, where M is smaller than the number of training patterns N. At the output layer, the linear weights associated and the position of the centers of the radial basis functions and the norm weighting matrix associated with the hidden layer are all unknown parameters that have to be learned. A supervised learning process using a gradient descent ([16]) procedure is implemented to adapt the position of the centers and their spreads(or widths) and the weights of the output layer. To initialize such a gradient descent GD procedure or CCSW we begin the search from a structured initial condition that limits the region of parameter space to be searched to an already known useful area through using a standard pattern-classification method as an RBF network ([17]). The likelihood of converging to an undesirable local minimum in weight space is already reduced. Also a supervised learning process using interior point method IPM developed in ([18]), [19]) is implemented to adapt the position of the centers and their spreads(or widths) and the weights of the output layer but which reduces the amount of computation compared to the one developed of GD in ([16]). A standard Gaussian classifier is used which assumes that each pattern in each class is drawn from a full Gaussian distribution.

An iterative RBF metamodel approach to approximating discrete event computer simulations was developed in order to develop accurate metamodels of computer simulations. An RBF neural network was used to learn the relationship between the simulation inputs and outputs. The iterative RBF metamodel approach starts with small training and testing sets, in terms of replications, and build RBF metamodels to use in performing factor screening to eliminate those input factors that do not appear to make much a difference on the simulation output. After eliminating the irrelevant factors, the baseline approach could be used on the remaining factors with substantial savings in total computer simulation runs. The iterative RBF neural network was developed and was applied on an offline evaluation of the quality of a ball steel production line providing grinding media for the mining industry.

RESULTS OF RBF TRAINING ALGORITHM IN MANUFACTURING DOMAIN

The purpose of this research is to predict the proportion of rejected bars after they have been tested for voids larger than the maximum acceptable size. During this process, 9 significant variables were chosen as inputs based on a priori knowledge. The output variable is the defective percentage on a given example. A sample or an example of a given cast is said to be rejected if its defective percentage exceeds 40%. This last cutoff number might sound too high but based on all the parameters considered during the testing by different engineers it represents an accurate information of what a bar that is accepted and a bar that is rejected. Our data were limited to 70 different examples because all inputs and output were complete for these different examples. The 9 different inputs are proprietary data for the steel company and cannot be made public. This poses problems for those that try to simulate our data on different regression models since the range of these different inputs cannot be released.

First order multiple linear regression models were applied to the different examples. The F test for regression was significant for α=0.01. The regression model of the first order linear equation is of the form:

$$\text{Percentage Defective} = A+BX_1+CX_2+DX_3+EX_4+FX_5+GX_6+HX_7+IX_8+JX_9 \qquad 1.$$

where A, B, C, D, E, F, G, H, I, J where constants that were identified during simulation, and X_1, X_2, X_3 , X_4, X_5, X_6, X_7, X_8, X_9 represent the 9 input variables. This process is a complex one since different input variables have different range values and different sampling values between these different ranges had to be tested for the best fit among all these constants. If each input variable had just a maximum of 5 different sampling values, there are 1048576 (4^9) training sets possible for the different sets of values of the input parameters considered. Only few of these different were selected that range from 512 points minimum to 5^9 points maximum. Out of all of these possible training sets, 13000 were investigated but only one case is reported here.

Figure 1 shows the result of the actual output for these different examples. Out of the 70 examples, 60 were used on training and the remaining examples on testing. Figure 1 also shows the result of the first order linear regression of equation (1) applied to the different examples. RBF were trained on the 13000 different simulation cases. The model was made out of 9 different input neurons , one output neuron for each range of percentage defective, e.g. each range spans 10 % on the Y axis of Figure 1 and as such 7 neurons are needed for the seven ranges from 0% to 70%. The number of hidden layers depend on the number of examplars from the 13000 cases for each range. There are more examplars that cover a given range than other ranges. That makes the neurons learn more about a given range than other ranges. There were 1521 nodes at the hidden layer for that particular run. The success rate was 95.7% on all the examples for RBF without CCSW. RBF with CCSW using GD was 96.7% on all the examples. RBF with CCSW using IPM was up to 97.6% which is impressive compared to other studies done with RBF on other benchmark problems ([16], [18], 19]). The error was of the order of 0.005. The matrices manipulations were quite heavy at the hidden layer level since a (1521*1521) matrix had to be inverted. RBF is heavy computationally. RBF without CCSW was of the order of 60 minutes. RBF with CCSW using GD was of the order of 120 minutes. RBF with CCSW using IPM was of the order of 90 minutes. The time is comparable with the first order linear regression analysis. From figure1 we could see that RBF outperformed the first linear order regression analysis on all examples. Thus, RBF did very well on the training patterns. Since the set of testing is only limited to ten examples, the vector of weights were used but with less centers since the number of patterns in each range has shrunk considerable. On the testing patterns the success rate was of the order of 90% without CCSW, 91.2% with CCSW using GD, and 91.8% with CCSW using IPM. The results for testing were not shown but they follow the results on training.

The next step was to reconsider the high number of inputs that is needed in the process. Other studies([12]) suggested testing whether the actual metamodel supports the reduction of the number of inputs without affecting the performance of the output results specially when the number of inputs is high. This idea was tested effectively in our metamodel. Given the fact that we know the output of RBF with the complete set of 9 inputs, we could try eliminating in each run a given input and calculate the output of RBF and compared with the known output when the set of inputs is 9. If the error between both values was still smaller than a given threshold that particular input could be ignored. Thus 9 different runs were investigated each time eliminating a given input from the calculation. The results show that RBF was successful in eliminating 2 inputs out of 9 without affecting the performance of the RBF networks and were still better than first order linear regression analysis with 9 inputs. When we tried to combine eliminating more than 2 inputs at a time the network degraded enormously and the success rate dropped to less than 50% on the training patterns.

In addition, it was shown that networks trained on individual replication output data had better generalization performance than networks trained on only the averages of the simulation output. In other words it is best when approximating stochastic computer simulations to use "noisy" individual replications rather than the "quiet" average values. The iterative RBF metamodel performed well at approximating computer simulations. The iterative RBF metamodel approach can be used by other researchers for comparison purposes when developing their own RBF metamodel approach. This contribution is useful in the area of artificial neural networks because there are many different existing and emerging ANN procedures to perform approximation and estimation tasks.

BREAKTHROUGH ASPECT OF THE WORK AND CONCLUSION

A major area of research is in the experimental design of neural networks as metamodels of computer simulations. This research filled in a critical need in the designs that take into consideration both the development(training) and the evaluation (testing and validation) of metamodels. This research shows how a metamodel is to be constructed using a training set for adjusting the weights, the centers and the spreads, and one test set for determining when to stop training and a second test set for evaluation of the generalization ability of the metamodel which was never done before. This is a premiere study which uses an iterative approach rather than baseline ANN metamodeling ([12]) which is a major improvement in that it reduces simulation runs in almost 40%. This research investigated the use of extreme values observed from the simulation. The studies cited above ([9], [10], [11], [12]) ignored such values from the training set. Integrating values of the average output for a particular combination of the input parameters that are much larger than all other average output values was possible because of the nature of RBF neural networks. This research further enhances RBF neural networks as de facto neural networks when off line analysis is needed where speed is not the goal but accuracy is the final issue. It was shown on the 4 benchmark problems that RBF outperformed the best and the fastest technique in two problems out of 4 in training and testing and in testing only on another one. Also the number of original inputs in the evaluation of steel ball quality which was quite high, e.g. 9, was reduced during the study to seven which is major finding compared to other studies which suggested the reduction but failed to achieve it on their own data. The manufacturing domain represented a challenge to RBF and RBF faired better than other simulation analysis techniques and other backpropagation neural network techniques. Second order linear regression analysis techniques could have been used but the researchers believed that this would escalated the time to calculate the results and would not have improved the results of the first order linear regression analysis.

The authors of this study encourage other simulation analysts to use this study and our RBF as a model to build on in their actual domain and simulation. Research along these lines is essential to ensuring that this tool is properly integrated with other emerging technologies to provide successful future generations of manufacturing simulations which will save millions of dollars in quality production.

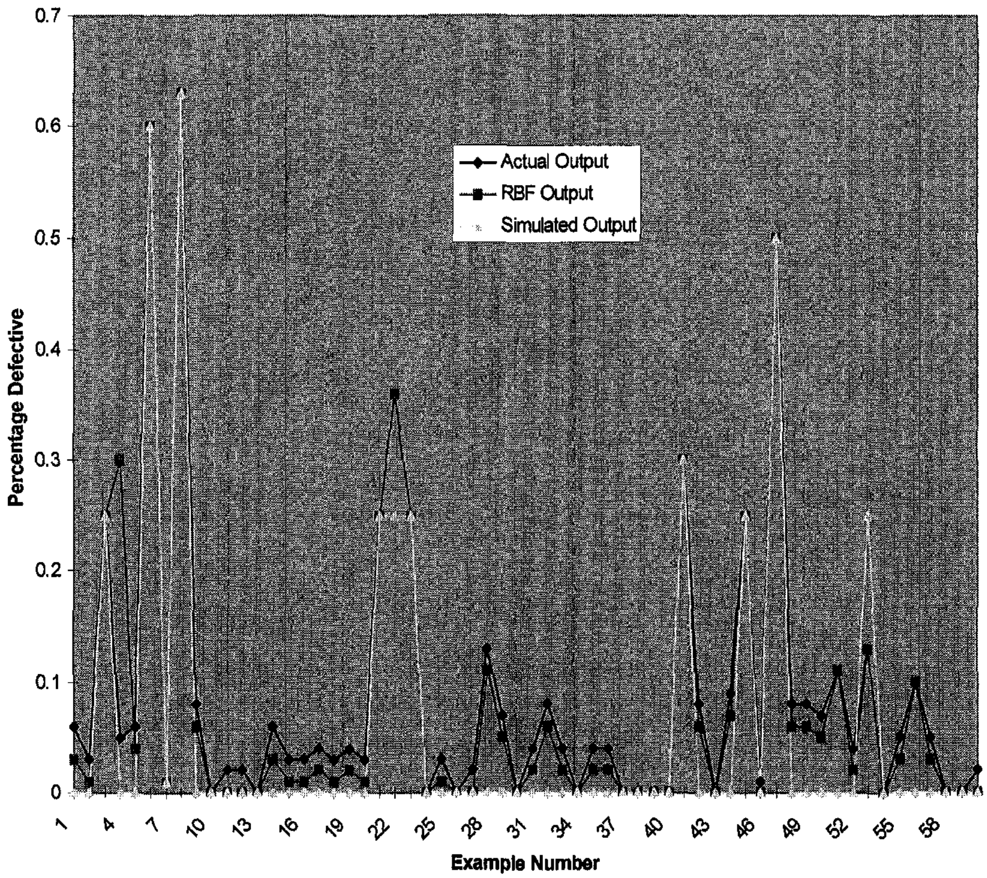

Fig. 1. Comparison of simulation results and ANN results with actual data.

REFERENCES

1. Myers, R., Khuri, A. , and Carter W., 1989. Response Surface Methodology:1966-1988. Technometrics, 31(2), 137-157.
2. Staib, W.E, Bliss, N.G., and Staib, R.B., 1991. Neural Network Conversion of the Electric Arc Furnace Into the Intelligent Arc Furnace. Iron and Steel Conference, Washington D.C, April 15.
3. Kominami, H., Naitoh, S., Kamada, N., Hqamaguchi, C., Tanaka, T., and Endoh, H., 1991. Neural Network System for Breakout Prediction in Continuous Casting Process. Tech Rept. 49, Nippon Steel, 6-3, Otemachi 2-chrome, Chiyoda-ku, Tokyo 100-71, Japan, April.
4. Stelljes, T.A., and Erickson, K.T., 1995. Modeling the Quality of Steel Production With an Adaptative Logic Network. In Intelligent Engineering Systems Through Artificial Neural Networks, Vol.5., Fuzzy Logic and Evolutionary Programming, by Dagli et al. (Eds), ASME Press.
5. Tateno, J., Asano, K., Moriya S., and Shiokawa, T., 1997. Neural Network Based Evaluation Method for Surface Glossiness of Steel Sheets. In the First International Conference on Intelligent Processing and Manufacturing of Materials.

6. Kusiak, J. and Pietrzyk, M., 1997. Artificial Neural Networks as a History Dependent Constitutive Model for Forming of Steels. In the 1st International Conf. on Intelligent Processing and Manufacturing of Materials.
7. Blanning, R., 1975. The Construction and Implementation of Metamodels. Simulation, 24, 177-184.
8. Yu, B., and Popplewell, K. Metamodels in Manufacturing: A Review. International Journal of Production Research, 32, 787-796.
9. Pierreval, H., and Huntsinger, R., 1992. An Investigation on Neural Network Capabilities as Simulation Metamodels. Proceedings of the 1992 Summer Computation Simulation Conference, 413-417, Reno, Nevada, July 27-30.
10. Hurrion, R.D., 1992. Using a Neural Network to Enhance the Decision Making Quality of a Visual Interactive Simulation Model. Journal of The Operations Research Society, 43(4), 333-341.
11. Badiru, A.B., and Sieger, D.B., 1993. Neural Network as a simulation metamodel in economic analysis of risky projects. Tech. Rept., Dept of Industrial Engineering, University of Oklahoma.
12. Kilmer, R.A., 1995. Applying Artificial Neural Networks to Combat Simulations. Mathematical And Computer Modelling, 23(1-2), 91-99.
13. T.M. Cover, 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers, EC-14, 326-334.
14. Powell, M.I.D., 1985. Radial Basis functions for multivariate interpolation: A review. IMA Conference on Algorithms for the approximation of functions and data, RMCS, Shrivenham.
15. Broomhead, D.S. and Lowe, D., 1988. Multivariable functional interpolation and adaptive networks. Complex Systems, Vol. 2, 321-355.
16. Meghabghab, G. , Nasr, G., and Boyd, D., 1997. Radial Basis Functions Neural Networks VS NOVEL on 4 benchmarks problems. Proceedings of the 10th International FLAIRS Conference, Daytona Beach, FL, 10-14 May 1997, 242-246.
17. Lowe, D, 1991. What have neural networks to offer statistical pattern processing. Proceedings of the SPIE Conference on Adaptive Signal Processing, 460-471, San Diego, CA.
18. Meghabghab, G. and Nasr, G. 1997. A new Radial Basis Function Neural Network VS NOVEL on 4 benchmarks problems. Intelligent Engineering Through Artificial Neural Networks, Edited by C. Dagli, Vol 7, ASME Press, New York, NY, 177-182.
19. Meghabghab, G. and Nasr, G. 1998. An Interior Point Radial Basis Function Neural Network on 4 benchmarks problems. Proceedings of 4th World Congress on Expert Systems, Mexico City, Mexico, March 16-20, 844-855.