COMMUN. STATIST.—THEORY METH., 29(9&10), 2215–2227 (2000)

A NEURAL NETWORK APPROACH TO RESPONSE SURFACE METHODOLOGY

Sandy D. Balkin
Ernst & Young LLP
1225 Connecticut Avenue, NW
Washington, D.C. 20036
sandy.balkin@ey.com

Dennis K. J. Lin
The Pennsylvania State University
Department of Management Science and Information Systems
University Park, PA 16802
DKL5@psu.edu

Key Words: function approximation, optimization

ABSTRACT

Response Surface Methodology is concerned with estimating a surface to a typically small set of observations with the purpose of determining what levels of the independent variables maximize the response. This usually entails fitting a quadratic regression function to the available data and calculating the function's derivatives.

Artificial Neural Networks are information-processing paradigms inspired by the way the human brain processes information. They are known to be universal function approximators under certain general conditions. This ability to approximate functions to any desired degree of accuracy makes them an attractive tool for use in a Response Surface analysis. This paper presents Artificial Neural Networks as a tool for Response Surface Methodology and demonstrates their use empirically.

1. INTRODUCTION

Response Surface Methodology (RSM) comprises a group of statistical techniques for empirical model building and model exploitation. By careful design and analysis of experiments, it seeks to relate a *response*, or *output* variable to levels of a number of *predictors*, or *input* variables that affect it (Box and Draper 1987). The investigator is interested in a presumed *functional relationship*

$$\eta = f(\xi_1, \xi_2, \ldots, \xi_k).$$

A graph of η against $\xi_1, \xi_2, \ldots, \xi_k$ is a response surface, in k dimensions.

In general, a polynomial is a function which is a linear aggregate of powers and products of the inputs. A polynomial expression of degree d can be thought of as a Taylor series expansion of the true underlying theoretical function $f(\xi)$ truncated after terms of the dth order. In practice, it is often true that, over a limited factor region, a first or second degree polynomial will provide a satisfactory representation of the true response function. The optimum of the function is then easily determined by taking the derivatives with respect to each variable and solving for zero. For a more rigorous treatment of RSM, see Box and Draper (1987), Myers and Montgomery (1995), Khuri and Cornell (1996), or Draper and Lin (1996), and for recent expositions see Myers (1999), Box and Liu (1999), and Lin (1999).

Hornik, Stinchcombe and White (1989) rigorously establish that standard multilayer feedforward neural networks with as few as one hidden layer using arbitrary squashing functions are capable of approximating any Borel measurable function from one finite dimensional space to another to any desired degree of accuracy provided sufficiently many hidden units are available. In this sense, multilayer feedforward networks are a class of universal approximators. This property makes neural networks a powerful tool to approximate a response surface given a finite number of observations.

For this study, we compare the standard multilayer feedforward neural network for approximating the function f relating independent and response variables with a second-order polynomial regression model. In practice, a neural network will be useful when little is known about the surface being approximated and it is very

complex. Thus, for testing and evaluation purposes, we propose using a Uniform Design (Fang, Lin, Winkler, and Zhang 2000) over a specified search space to select the experimental runs. We evaluate the neural network method by measuring the absolute deviation between the predicted and actual optimum of the surface once the final factor region has been identified and comparing it with the results derived from the regression model fit.

This paper is organized as follows. Section 2 reviews some of the basics of neural networks and motivates their use for response surface optimization problems. Section 3 describes how neural networks can be used to fit surfaces in experimental observations which is then described in Section 4. The paper concludes with a discussion of the results and directions for future research in Section 5.

2. NEURAL NETWORKS

An Artificial Neural Network (ANN) is an information processing paradigm that is inspired by the way the brain processes information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurons) working in unison to solve specific problems. ANNs, like people, learn by example. Learning in biological systems involves adjustments to the synaptic connections that exist between the neurons. This is true of ANNs as well.

There has been much publicity about the ability of ANNs to learn and generalize. Although the learning algorithm often associated with the multilayer perceptron is backpropagation, the problem of finding the appropriate weights to minimize the sum of squared error is essentially a nonlinear optimization problem that is implementable in many standard statistical packages. For an explanation of the relationship between statistical methods and neural networks see, for example, Ripley (1993), Sarle (1994) and Cheng and Titterington (1994). The main point to be stressed is that ANNs *learn* the same way that many statistical algorithms estimate. For an explanation of learning in a neural network, see Hinton (1992). The claimed advantages of neural networks are that they deal with the non-linearities in the world in which we live, can handle noisy or missing data, can work with

large numbers of variables or parameters, and provide general solutions with good predictive accuracy.

A feed-forward network is comprised of units that have one-way connections to other units arranged in layers. Connections only move forward through the layers. A typical feed-forward network can be represented by the function

$$y_k = \phi_o \left(\alpha_o + \sum_{\text{all } h} w_{ho} \phi_h (\alpha_h + \sum_{\text{all } i} w_{ih} x_{ki}) \right) \tag{1}$$

where x_{ki} represents the ith value of the kth input vector corresponding to the kth response (y_k). Parameters $\{\alpha_h\}$ denote the weights for the connections between the constant input and the hidden neurons and α_o denotes the weight of the direct connection between the constant term and the output. The values $\{w_{ih}\}$ and $\{w_{ho}\}$ denote the weights for the other connections between the inputs and hidden neurons and between the neurons and the output respectively. The functions ϕ_h and ϕ_o denote the activation functions used at the hidden and output layers respectively.

The network shown in Figure 1 has three layers: input, hidden and output. The choice of structure of the three layers is known as *choosing the architecture* in the neural network framework and is analogous to model selection in the regression framework. The user needs to decide the number of input nodes, the number of hidden layers and hidden nodes, the number of output nodes, and the activation functions. The number of input nodes correspond to the number of variables to consider for the model. The hidden layer and nodes parameter selection is very important in that it is this feature that allows the ANN to perform the nonlinear mapping between inputs and outputs. The number of output nodes is specified directly by the problem. Currently, however, there is no widely accepted method for making these model design decisions.

3. ANN IMPLEMENTATION

This section describes the process and implications of training a neural network to estimate a response surface and how to find the maximum value. Each decision is critical to the successful application of the neural network.

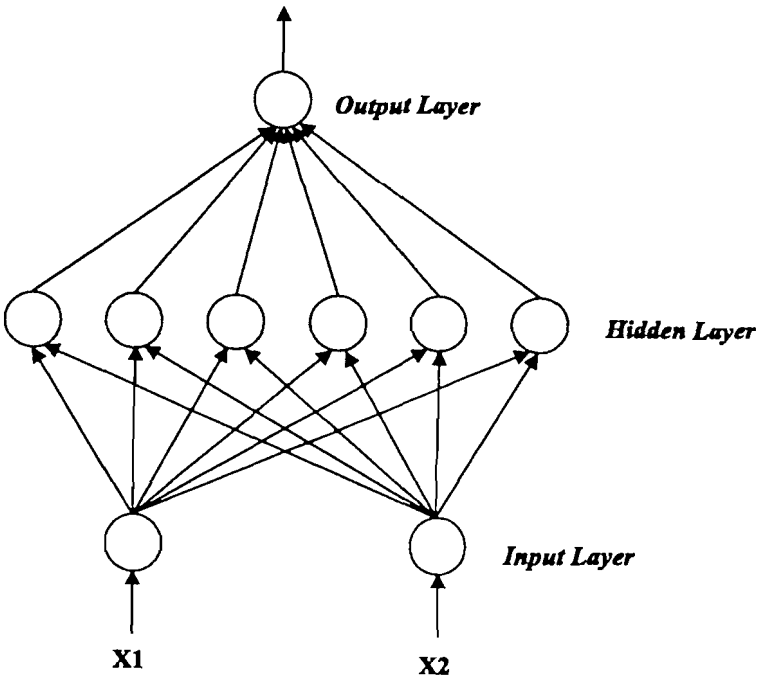

Figure 1: Artificial Neural Network

1. IDENTIFY THE SEARCH SPACE. Limit the factor region using conventional RSM methods.

2. CHOOSE THE GRID SIZE. This entails choosing the experimental design. Since we are not assuming a specific functional form for the response surface, the more distinct observations a neural network has to approximate the surface, the better the expected results.

3. CHOOSE THE NETWORK ARCHITECTURE. As described above, this step is vital in the modeling process. If too few hidden nodes are chosen, the neural network will not have the ability to learn the relationship between inputs and outputs. If too many hidden nodes are chosen, the model will be too complex

and the neural network may induce spurious correlations between independent and response variables.

4. TRAIN THE ANN. This entails altering the neural network weights minimizing the sum of squared errors between the training data and network predictions.

5. OBTAIN PREDICTED VALUES FOR THE GRID. Approximate the surface using predicted values generated by the neural network on a much finer grid than on which it was originally trained.

6. PERFORM GRID SEARCH FOR THE MAXIMUM VALUE AND ITS CORRESPONDING INPUTS. Find the maximum surface value and the corresponding values of the input variables to decide what levels are predicted to maximize the response surface.

4. EXAMPLE

In this section, we compare neural networks with traditional Response Surface Methodology. RSM typically assumes the response surface is quadratic and fits a quadratic regression model to the observations to estimate the surface. Hence, it is expected that the neural network and regression models will perform comparably when the surface is actually quadratic but that the neural model will be superior when the surface is more complicated. In practice, surfaces are most likely in between, though it is assumed that a quadratic function will serve as an adequate approximation. Balkin and Lin (1999) compares neural networks with traditional RSM on a known quadratic surface as well as on a real life data set. For these examples, the two methods indeed perform equally well in terms of ability to approximate the true maximum response.

For a more complex example, consider the inverse polynomial in Figure 2

$$y = (10x_1^4 - 20x_2x_1^2 + 10x_2^2 + x_1^2 - 2x_1 + 5)^{-1}.$$

Given in Fox (1971), the expression in parentheses is known as a banana function because the global minimum is inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult

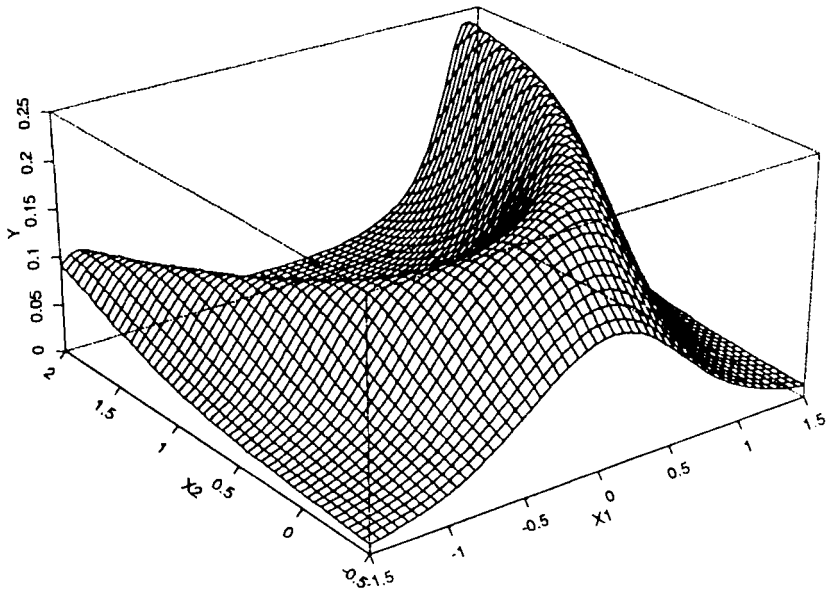

Figure 2: Complex polynomial surface.

for optimization algorithms to achieve. We take the inverse of the polynomial to convert it to a maximization problem as is typically found in RSM and to add complexity to the surface, creating a very difficult response surface problem.

We assume that we have sufficiently narrowed the search space by conventional RSM techniques to $x_1 \in (-0.5, 2.0); x_2 \in (-1.5, 1.5)$ and are now interested in the final step of determining the optimal response. In order to see how well a neural network can determine the optimal value of this surface, we fit one with nine nodes in the hidden layer and regression models on observations chosen using a design with grid sizes of 4 through 9. Thus, 16, 25, 36, 49, 64, and 81 equally spaced observations are generated over the search space and used to estimate the parameters of the neural network and polynomial regression models. The maximum value of the surface is then determined via a grid search for the neural network and by using the derivatives of the regression model. We then look at the absolute

Table 1: Experimental design and response observations for grid size of four

		x1 values			
		-0.5	*0.33*	*1.17*	*2*
x2 values	*-1.5*	0.01	0.02	0.05	0.09
	-0.5	0.08	0.16	0.07	0.03
	0.5	0.1	0.23	0.08	0.028
	1.5	0.01	0.02	0.06	0.21

deviation between the predicted and actual optimal value for the two models over the different grid sizes.

For example, let us consider a grid size of four. Table 1 displays the corresponding function values in this 4×4 grid. We then fit a neural network with nine nodes in the hidden layer with the x_1 and x_2 values as the inputs and the function response at those values at the output using the MASS library (Venables and Ripley (1999)) for Splus. Once the parameters of the neural network are estimated, we have what can be considered a complicated nonlinear regression function. We then search for the largest response value and take that as the surface maximum. In this case, the predicted maximum value of 0.2415 occurs at inputs $x_1 = 1.00$; $x_2 = 1.28$. Since the actual surface maximum is 0.25, which occurs at $x_1 = 1.0$; $x_2 = 1.0$, we see that the predicted error for a grid size of four is 0.0085.

Figure 3 displays the absolute error between the actual and predicted surface maximum for the regression and neural network models over grid sizes. We can see from this figure that the neural network outperforms the regression model in terms of ability to identify the optimal value of the response variable. This is no surprise since the response surface is a polynomial of order higher than is being fitted. Thus, in this example, the neural network is able to learn the functional relationship while the linear model is not. We also see that the performance of the neural network does not increase uniformly as the number of observations increase. This is possibly an artifact of the different grid resolutions straddling the optimal value.

Certainly, an experienced RSM user may run the experiment sequentially starting in a small region, check for lack of fit, refit the model etc. whenever necessary

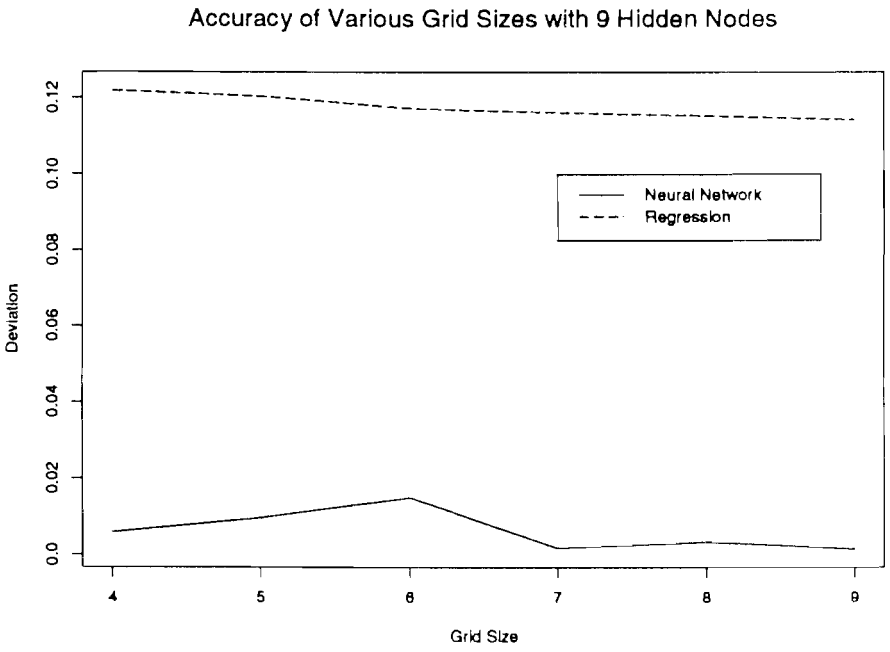

Figure 3: Absolute deviation for different grid sizes.

(see, for example, Lin (1999)). The example given here is simply to demonstrate the usefulness of ANN, and thus those details are not displayed. Moreover, the standard model diagnosics are important and should be performed, but again are not reported here.

5. DISCUSSION

The purpose of this paper is to present Neural Networks as a tool for fitting response surfaces. It is not the purpose of this study to convince practitioners to use only neural networks when fitting response surfaces, but rather to show how they augment the RSM toolkit. It is always possible to fit an over-determined polynomial to observed data to duplicate the response function. Our experience indicates that if the response is not "smooth" where the classical RSM does not perform well, a neural network approximation will generally perform better.

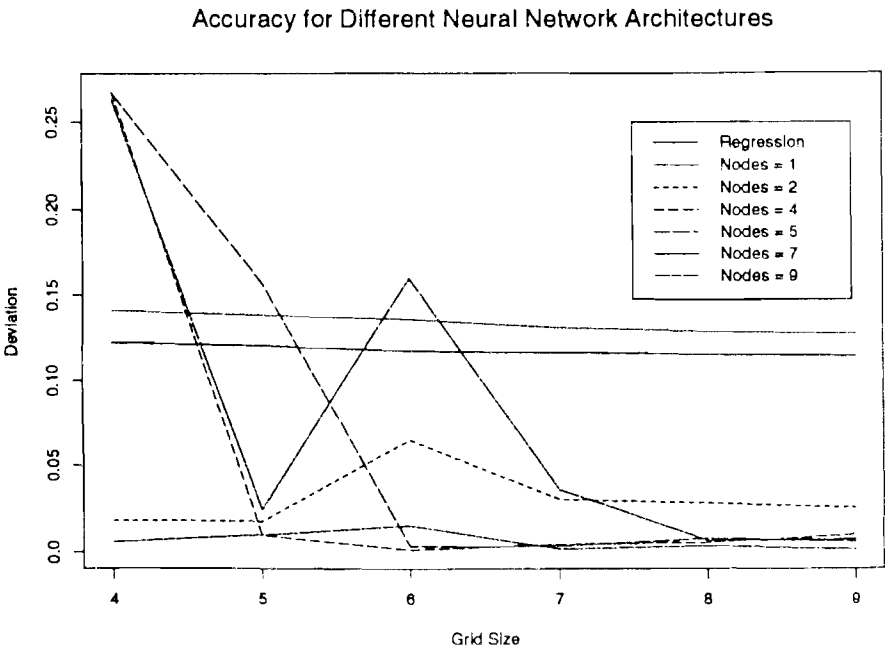

Figure 4: Absolute deviation of actual and predicted maximum values for different number of nodes in hidden layer.

One of the primary considerations when applying neural networks is the choice of the number of nodes in the hidden layer. Nodes in the hidden layer are chosen by trial and error based on the perceived complexity of the relationship between the explanatory and response variables. For example, let us reconsider the complex function example and investigate this issue. Figure 4 shows the absolute deviation between the actual and predicted optimal response value for grid sizes 4 through 9 with 1, 2, 4, 7 and 9 nodes in the hidden layer along with the results from the quadratic regression model. We observe from this plot that it is important to allow the network enough "freedom" to explore complex relationships and to provide the network with a sufficient number of examples with which to learn the functional relationships. The neural network with a single node in the hidden layer

and some of the neural networks trained on a 4×4 grid do not perform as well as the regression model. However, for the most part, the neural network is able to identify the maximum value of the surface more accurately. Since the optimization process for the neural network is dependent on the initial starting point, it may be useful to compare its results with those obtained from the regression model whose parameters are estimated consistently the same. The fact that deviations do not monotonically decrease, as noted in the previous section, is most likely due to the way the observations designated surround the true optimal value of the surface.

This study demonstrates the use of neural networks for Response Surface Methodology. Neural networks cannot replace linear regression as a statistical technique, but should instead be considered an additional method in a statistician's toolkit. With today's computing power, such computational techniques are worth using and easy to implement. Future work in this area can include better designed experiments and diagnostics to reduce the uncertainty associated with results derived from neural networks and tests to determine when a neural network will result in more accurate results.

ACKNOWLEDGEMENTS

We thank the guest editors (Norman R. Draper and Philip Prescott) and two referees for their constructive comments which significantly improved the presentation of the paper. Dennis Lin is partially supported by *National Science Foundation* via Grant DMS-9704711 and *National Science Council of ROC* via Contract NSC 87-2119-M-001-007. Computer equipment was provided by an IBM Shared University Research (SUR) grant.

References

Balkin, S. D. and D. K. J. Lin (1999). A neural network approach to response surface methodology. The Pennsylvania State University, Department of MS&IS Working Paper Series # 99-4.

Box, G. (1999). Statistics as a catalyst to learning by scientific method, Part II-A Discussion. *Journal of Quality Technology 31*, 16-29.

Box, G. and P. Liu (1999). Statistics as a catalyst to learning by scientific method, Part I-An Example. *Journal of Quality Technology 31*, 1–15.

Box, G. E. and N. R. Draper (1987). *Empirical Model-Building and Response Surfaces*. John Wiley & Sons, Inc.

Cheng, B. and D. M. Titterington (1994). Neural networks: A review from a statistical perspective (with discussion). *Statistical Science 9*(1), 2–54.

Draper, N. R. and D. K. J. Lin (1996). Response surface designs. In S. Ghosh and C. R. Rao (Eds.), *Handbook of Statistics, Vol. 13*. Elsevier Science.

Fang, K.-T., D. K. J. Lin, P. Winkler, and Y. Zhang (2000, May). Uniform design: Theory and application. *Technometrics*.

Fox, R. L. (1971). *Optimization Methods for Engineering Design*. Addison-Wesley.

Hinton, G. E. (1992). How neural networks learn from experience. *Scientific American*, 145–151.

Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer feedforward networks are universal approximators. *Neural Networks 2*, 359–366.

Khuri, A. I. and J. A. Cornell (1996). *Response Surfaces* (Second ed.). Dekker.

Lin, D. (1999). Discussion on Box and Liu, Box, and Myers. *Journal of Quality Technology 31*, 61–66.

Myers, R. H. (1999). Response surface methodology-Current status and future directions. *Journal of Quality Technology 31*(1), 30–44.

Myers, R. H. and D. C. Montgomery (1995). *Response Surface Methodology*. Wiley.

Ripley, B. D. (1993). Statistical aspects of neural networks. In O. E. Barndorff-Nielsen, J. L. Jensen, and W. S. Kendall (Eds.), *Networks and Chaos: Statistical and Probabilistic Aspects*. Chapman & Hall.

Sarle, W. S. (1994). Neural networks and statistical models. In *Proceedings of the Nineteenth Annual SAS Users Group International Conference*. SAS Institute Inc.

Venables, W. N. and B. D. Ripley (1999). *Modern Applied Statistics with S-PLUS* (Third ed.). Springer.

Received May, 1999; Revised March, 2000.