AN EXPERIMENTAL DESIGN ADVISOR AND NEURAL NETWORK ANALYSIS PACKAGE

Aaron J. Owens1 and Michael T. Mocella2

1Du Pont Central Research
P. O. Box 80357
Wilmington, DE, 19880
U.S.A.

2Du Pont Electronics
P. O. Box 80366
Wilmington, DE, 19880
U.S.A.

Abstract. DANA [Design Advisor/Neural Analyzer] is a PC-based experimental design advisor and neural network analysis system for understanding and optimizing physical processes. Given a specified number of process inputs and outputs, the Design Advisor gives two options for a set of experiments to perform in order to map the input variables to the process output responses: a standard statistical design and a novel smaller design specifically developed for neural network analysis. The neural network employed in the Neural Analyzer is the back-propagation type, in which a single hidden layer is used to relate the input nodes (variables) to the output nodes (responses) for the experimental data set analyzed. The Network Output module uses the neural model to produce a number of useful graphs and a "Virtual Process" for interactive modeling.

I. INTRODUCTION

The use of statistically-motivated designs to suggest which experimental measurements to take is a well-established procedure for processes with several input variables (Box, et al., 1978). In these designs, the multi-variate input space is sampled in a systematic and bias-free way, to maximize the amount of information that can be obtained from each experiment. The designs are typically based on low-order polynomial models relating the inputs to a single output.

The back propagation network [BPN] is a neural network paradigm which can represent essentially any multi-variate, nonlinear input/output function map (Cybenko, 1989). Its representation is model free, in the sense that the learning algorithm automatically determines the functional relationship between inputs and outputs directly from the data, without requiring a hypothesized functional form.

Experimental designs are good for suggesting what measurements to take in order to extract maximum information with minimum experimental effort. The BPN provides a nonlinear, model-free, multi-variate data fitting algorithm. We have found that the BPN is an ideal tool for analyzing the results of designed experiments. Typically the BPN model gives higher-accuracy fits to the data, with fewer adjustable parameters, than the polynomial regression models for which the experimental designs were made. This is especially true for experiments in which there are multiple output responses. Classical method require a separate model for each response, while the BPN models all output responses simultaneously. The BPN exhibits "output leveraging": the weights from input to hidden layers are shared among all the outputs, so it requires fewer weights (and thus fewer experiments) to produce accurate mapping for multiple-response problems than classical polynomial methods.

We have created a PC-based program called DANA [Design Advisor/Neural Analyzer] to facilitate the combined use of designed experiments and neural network analysis. It consists of three modules, as shown in Figure 1.
In the Design Advisor, the user specifies the number of variables and responses, along with their names and ranges. The number of hidden units for the BPN is specified, and the selection is made whether to use a standard statistical design or the custom sparser design requiring fewer experiments. The Design Advisor then produces a table of the experiments to be performed, including the (non-random) order of performing the experiments. The design includes experiments both for training and testing the neural network model. After the experiments are completed, the output response values are entered into the data table.

The Network Solver uses a single-hidden-layer back propagation network. After the network is trained, diagnostic information to assess the quality of the fit is available, including fit statistics and several graphs.

The experimenter typically tries several models with different numbers of hidden units, to determine which produces the best fit to the observed data. With the best model, the Network Output module is used to investigate the functional relationships between the process input variables and output responses. As well as contour and elevation (surface) plots, the module includes a unique "Virtual Process" in which the experimenter can interactively change variable settings and observe graphically the changes in the responses.

II. DESIGN ADVISOR

Given the number of process input variables \( (n_i) \) and output responses \( (n_o) \), plus the experimenter's estimate of the complexity of the input/output mapping as parameterized by the number of hidden units \( (n_h) \), the purpose of the Design Advisor is to specify which experiments should be performed in order to adequately and efficiently sample the input variable space for modeling the output responses. The two types of statistical designs supported are the standard Central Composite Design and the novel Star Plus Simplex Design. The system can presently accommodate up to 7 input variables and 10 output responses.

Central Composite Design

Conceptually, the input variables are first linearly scaled, so that each has a minimum value of -1 and a maximum value of +1; the experimental design points are best visualized in this scaled hyperspace of \( n_i \) dimensions. For the Central Composite Design, imagine an \( n_i \)-dimensional hypercube inscribed in its corresponding hypersphere of radius 1. A center point (C) at the origin is at the mid-range of all variables. Each of the vertices of the hypercube is an experimental point, labeled factorial (F). For each of the input variables, the two star (S) points are at the intersection of the variable's axis with the surfaces of the hypersphere. This design is illustrated in the left half of Figure 2, for three input variables. The design is called composite because it is based on sampling in both a spherical and a cubic shape in the hyperspace (Box, et al., 1978, Chapter 15).

For \( n_i \) input variables, with no replicates or test points, the number of experiments \( (n_e) \) required for these the Central Composite Design is:

\[
(1a) \quad n_e = 1 + 2 n_i + (2) (n_i) \quad \quad [\text{Central Composite Design}]
\]

\( (C) \quad (S) \quad (F) \)