ESL-PA-94/07-02 (3)

OR-94-17-3

GENERALIZED NONLINEAR REGRESSION WITH ENSEMBLE OF NEURAL NETS: The Great Energy Predictor Shootout

Bradley P. Feuston, Ph.D. **John H. Thurtell, Ph.D.**

ABSTRACT

A novel approach to data modeling has been developed using an ensemble of artificial neural networks. This development has been driven by the need in the petroleum business to model data that has poorly defined independent parameters, poor repeatability and/or reproducibility (more than ±10% variation in repeated measurements), and relatively few data sets. Our objective for entering "The Great Energy Predictor Shootout" was to evaluate the efficiency of our approach. In particular, we were interested in how quickly a model could be constructed and the accuracy of the model's predictions. This scientific competition was ostensibly organized to compare various modeling methodologies applied to both time-dependent (dynamic) and time-independent problems. Both problems were given to the participants with little descriptive information. The objective of this competition was to train (or fit) a model given a complete set of data (independent and dependent variables) and then blindly predict the dependent variables when given only the independent variables of a different data set. The predictions were then sent to the organizers for evaluation. While our entry ranked third overall, the models were generated with less than one man-day of effort—far less than the two months required by the first-place winner and the one to one-and-one-half months spent by the other winners.

INTRODUCTION

In an effort to compare the various modeling techniques now being applied to data analysis in most observational disciplines, such as physics, biology, and economics, the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE) sponsored "The Great Energy Predictor Shootout," hosted and organized by J.F. Kreider and J.S. Haberl. The objective of the competition was to provide a rigorous unbiased comparison of various mathematical modeling techniques and to promote dialogue between the relevant disciplines. Monetary prizes were not awarded. The specific interest of the organizers was to predict or forecast the energy use of heating, ventilating, and air-conditioning (HVAC) systems and is reflected in the data sets employed for the competition. The first problem involved predicting the electricity, chilled-water, and hot water use in a large building. Accurate modeling of HVAC systems is important for diagnostics, process control, optimization, and energy management. These systems have typically eluded traditional approaches involving system identification and characterization and numerical simulations. The second challenge involved building a model that could accurately reproduce the instantaneous solar beam flux from four fixed global solar flux detectors.

The competition required the construction of four models—three energy forecasting models (explicitly dynamic) from data set A and the time-independent model solar flux, referred to as data set B. Our in-house-developed neural net code, successfully employed in a variety of problems, was used to construct the models for our entry.

WHAT IS A NEURAL NETWORK?

It is common in the applied sciences to search for a mathematical expression relating dependent and independent variables. In the absence of a working theoretical model, regression techniques are often employed to find an approximate functional form that can best describe the relationship between the independent variables and the observed dependent quantities for a system of interest. When successful, the result is an empirieal model that is useful for predicting the effect of changing input data on the dependent variables. These dependent quantities, T, may generally be expanded in terms of a set of basis functions,

$$T = O = \sum_{i=1}^{N} \alpha_i F(\beta_i^T X + \gamma_i), \qquad (1)$$

where $\{\alpha_i$ = scalar, β_i = vector, γ_i = scalar$\}$ are adjustable parameters and X denotes the array of independent variables. Typically, a single polynomial expansion of order 1 is used as the fitting function in Equation 1. The parameters are determined by minimizing

Bradley P. Feuston and **John H. Thurtell** are senior research physicists at the Central Research Laboratory of Mobil Research and Development Corporation, Princeton, NJ.

$$E = \sum_{m=1}^{M} (T_m - O_m)^2 \qquad (2)$$

where the sum is over all the known sets of data composed of the dependent quantities, T, and the independent variables, X. The adjustable parameters $\{\alpha_i, \beta_i, \gamma_i\}$ are uniquely determined if M is equal to their total number (N_T), overdetermined if M is greater than N_T, and underdetermined if M is less than N_T. Obviously, the success of this approach depends on how well the fitting function, F, mimics the underlying functional relationship in the data. Hence, much effort may be spent searching for a suitable expansion function. In addition, for each trial function, the parameters must be optimized before the quality of fit can be determined. This trial-and-error method may be largely circumvented with the neural net approach (Wasserman 1989; Admoaitis et al. 1990; Leonnard and Kramer 1991; Bhat and McAvoy 1990; Poggio and Girosi 1990; Cybenko 1989; Moody and Darken 1989; Rumelhart et al. 1986).

The easiest way to understand a neural network is to step back and consider the much simpler problem of fitting a polynomial to a set of (x,y) data points. If there are M distinct data points, then there is a unique polynomial of order M-1 that passes through all the data points. This is true because the polynomials form an orthogonal set of functions that span the space of one-dimensional functions. Since the functions are orthogonal, there is a straightforward method to determine the coefficient for each polynomial.

On the other hand, there are a number of lower-order polynomials that may pass sufficiently close to all M data points. These curves are overdetermined and hence the final functional form depends on the process implemented to determine the form. In neural network parlance, this is referred to as the *training algorithm.*

A neural network, like the polynomial example above, is normally an overdetermined nonlinear mapping from the input data set to the output data set. The polynomial is replaced by one of any number of nonlinear functions that span the function space, and a variety of training algorithms are available. The learning properties of a neural network are properties of the training algorithm rather than the network itself.

Traditional statistical regression techniques have problems when the input variables in the data are not truly independent. This situation can lead to singularities in the solution that may not easily be handled. Neural networks do not suffer from this weakness. The basis functions for a neural network are not orthogonal or unique. Hence, the training algorithm is insensitive to these properties.

DETAILS

A neural net consists of three parts—an input vector (independent variables), an output vector (dependent variables), and an algorithm that maps the input space to the output space. A typical neural net configuration is depicted in Figure 1, where the internal portion consists of one or more hidden layers that are connected to each other and to the external layers by a set of weights, expressed as two-dimensional matrices, W. In a feed-forward neural net, the value of each node in a particular hidden layer is the result of a nonlinear transfer function whose argument is the weighted sum over all the nodes in the previous layer plus a constant bias, B. To train the neural net, one begins with a set of training data consisting of the input vector, X_0^m, and the corresponding target vector, T_m. The internal weights are adjusted until the sum of differences between the neural net outputs, O_m, and the corresponding target values, T_m, is minimized for all the training data.

From Figure 1 it is apparent that a neural net may be considered just a generalization of Equation 1. A neural net with zero hidden layers is just a simple linear expansion, and a net with one hidden layer and a single output takes a form similar to that of Equation 1:

$$O_m = \sum_{i=1}^{N_1} W_2(i) F\left(\sum_{j=1}^{N_0} W_1(i,j) X_0^m(j) + B_1 \right). \qquad (3)$$

In Equation 3, N_1 is the number of nodes in the hidden layer and N_0 the number of independent variables. A mathematical proof has demonstrated that any continuous function can be uniformly approximated by a finite number of nodes in a single hidden layer (sum over N_1 in Equation 3) using a sigmoidal nonlinearity for the basis function (Cybenko 1989), e.g.,

$$F(x) = \begin{cases} 1, x \to +\infty, \\ 0, x \to -\infty. \end{cases} \qquad (4)$$

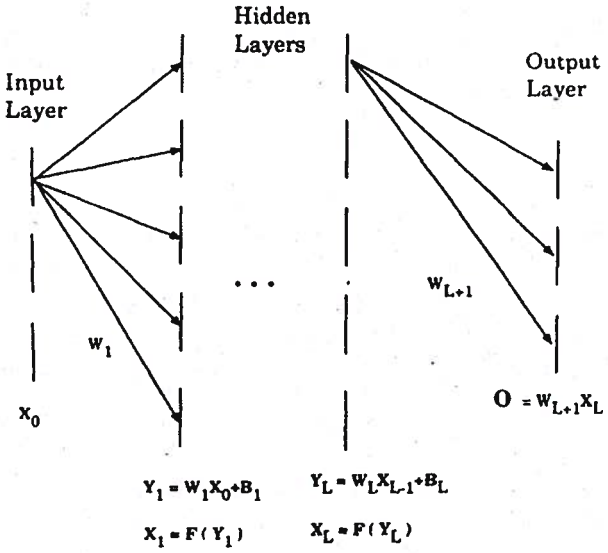

Figure 1 *General topology of a feed-forward neural net.*

A single hidden-layer, feed-forward neural network program written in Fortran and developed by the authors was used for all the calculations. It allows for an arbitrary number of input nodes and hidden nodes but allows for only one output node or dependent variable. The weights (W_1, W_2) and the single bias (B_1) are determined by minimizing $\delta E/\delta W_1$, $\delta E/\delta W_2$, and $\delta E/\delta B_1$, where E is given by Equation 2. The conjugate gradient method is employed to carry out the optimization (Rumelhart et al. 1986).

Three different basis functions (transfer functions) are also available (see Figure 2) in the Fortran code. These functions are typically referred to as

the radial distribution function,

$$F(x) = \exp(-x^2); \tag{5}$$

the hyperbolic tangent function,

$$F(x) = \tanh(x); \tag{6}$$

and the sigmoid function,

$$F(x) = \frac{1}{1+\exp(-x)}. \tag{7}$$

As seen from Figure 2, both the hyperbolic tangent and sigmoid functions may be considered sigmoidal, although in the literature Equation 7 is usually referred to as the sigmoid function.

Pre- and postprocessing of the data are important contributions to our implementation of neural nets. Data preprocessing includes scaling transforms and principal component analysis (PCA) (Noble and Daniel 1977). It is extremely important to properly prepare the data before training the neural network. The transfer functions used in this work are nonlinear when the input vectors are on the order of 1 but quickly saturate to a constant when the input vector grows large. This is illustrated in Figure 2. In general, the input and output values should all fall into this

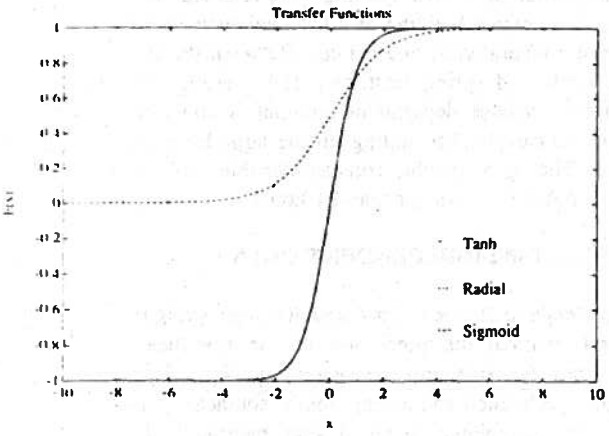

Figure 2 *Example of three nonlinear transfer functions frequently used in neural nets.*

nonlinear region. This is easily accomplished by simply rescaling the data. A similar problem arises when the data span many orders of magnitude. In this case, it is sometimes desirable to rescale the data with a logarithmic function to force the data to span the region from 0 to 1 in a reasonable way. While there is no clear formula for success here, each input and output variable is generally scaled to fall into the region between 0.0 and 1.0. This rule has resulted in data that have been successfully modeled.

The purpose of the PCA is to transform the independent variables to a reduced set of orthogonal inputs that contains the majority of the information in the data. In this way, the number of neural net inputs and hence the number of adjustable parameters may be significantly reduced. Although neural nets are known to accommodate interdependencies among the "independent" inputs, experience has found orthogonal inputs to yield better models. It cannot be overstated that failure to properly prepare the data results in data sets that cannot be fit with a neural network.

NETWORK TRAINING

When training neural nets, the problem of saturation or overtraining is confronted. At some point in the minimization of E in Equation 2, the predictions of the network are no longer improved and usually become less reliable with further training. In the initial phase of training, the neural net is learning a gross functional relationship between the input data and the dependent quantity. This relationship is continuously refined during minimization, with the net becoming more narrow in its focus with every iteration. If there are enough data to describe the desired relationship, the training will eventually saturate and the further minimization will not improve the net's predictions. With too few data points, continuation of training may result in very accurate predictions in a small region around the training points and poor predictions elsewhere. Essentially, this overtrained net moves beyond the more general functional relationship sought in favor of accurately fitting the noise in the specific training data. To address these problems, the error of a test data set is monitored during training. Only the values of the weights giving the lowest error in the test set are retained.

NEURAL NET ENSEMBLE APPROACH

Due to the complexities intrinsic to the raw materials, production processes, and end products of the petroleum business, much of the data taken for characterization, process optimization, and marketing purposes are incomplete for model building or suffer from insufficient quantity and quality (i.e., poor reproducibility).* Although this is a well-established industry with decades of experience and reams

*The term *repeatability* (reproducibility) refers to repeated tests at the same (different) facility. However, the term *reproducibility* is used loosely in this text to include both cases.

of raw data, techniques and processes are constantly changing in response to technical advances, increased environmental concerns, market pressures, and compositional variations in crude oil sources. Because of the relatively short time that the data are current and the variability in data-acquisition methodologies, much of the data used in model-building are inadequate for many of the standard fitting techniques. Even some widely accepted phenomenological models require "fudge" factors for interpreting results.

In our early modeling attempts, the simple feed-forward neural nets described here have proved to provide a quick and reliable model-building capability using fewer data points and little or no knowledge of underlying functional relationships. This early approach has been successful in several applications where the independent parameters have been appropriately identified and the quantity and quality of data have been sufficient. However, in many applications, problems exist in determining the relevant independent parameters and in the reproducibility of data. In such applications, the real predictive capability of the single-neural-net model (as well as the classic modeling techniques) is not easily ascertained. These instances were characterized by errors in the fit to the training data significantly greater than the error intrinsic to the measured data.

For our early models, several hundred nets with various topologies and transfer functions were trained and the single neural net with the best fit was retained as the model. When the independent variables have been clearly identified and the quality and quantity of the data set have been sufficient, most of the nets produced good results. When this was not the case, most of the nets produced poor models with a much wider distribution of fitting errors. Keeping the neural net with the best fit in this latter case is highly questionable since it is not clear whether the net learned the underlying correlation being sought or specifics associated with the data set. To address these uncertainties, the collective properties (e.g., mean value of predictions from the ensemble) of the set of all the trained nets were investigated.

We have observed that a statistical analysis of the distribution of predictions of all the neural nets, with both good and bad fits, yielded predictions more accurate and reliable than a single neural net. During the program development, we observed that when a Gaussian distribution of noise has been added to otherwise perfect training data, the mean value of the ensemble of neural nets will be more accurate (e.g., closer to the perfect data) than the data on which it was trained.

DATA SET A: TIME-DEPENDENT DATA

The objective of the first problem from the "Great Energy Predictor Shootout" was to model the energy use of a large building given the time stamp (year-month-day-hour), dry-bulb temperature (T_{db}), humidity ratio (H), solar flux (SF), and wind speed (W). Energy use is expressed in terms of three quantities—electricity (WBE—whole-building electric), chilled water (WBCW), and hot water (WBHW). Each data point of the training set consisted of the five independent variables and three dependent variables, whereas the test set contained only the five independent variables. Hourly data were provided from September 1, 1989, to December 31, 1989, for training, with the task being to predict the hourly energy use for the subsequent two months—January 1, 1990, to February 28, 1990. This is obviously a difficult task for an empirical model—to be fitted on fall data and extrapolated to winter conditions.

At the outset, no information was given regarding the size of the building, its purpose, or location. However, from the data it was apparent that the building was located in the United States, since the days corresponding to both Thanksgiving and Christmas had anomalously low energy expenditures. We made no attempt to optimize the various possible ways to represent the independent variables but made the assumption that the characteristic time for the building's HVAC system was three hours, which is similar to that of the authors' laboratory. The input data for time t were then constructed, with the hour, T_{db}, H, SF, and W at times t, $t-1$ hour, $t-2$ hours, and $t-3$ hours. To reflect the effects of weekends and holidays, an additional input parameter was included. The value of this parameter was set to 1.0, 0.3, and 0.0 for workdays, Saturdays, and Sundays/holidays, respectively. Principal component analysis reduced the 24 input variables (six parameters at four different times) to 8 independent variables for the neural net accounting for more than 95% of the variance. A total of 300 neural net models was then trained with the 2,923 data sets.

Results for both the training and test sets are shown in Table 1 as well as the results from the other winners. Our results for the WBE, WBCW, and WBHW test sets are shown in Figures 3 through 5, respectively. It is clear from Table 1 that all the top entrants overpredicted the electrical use while underpredicting the cold and hot water use. At an ASHRAE seminar held June 28, 1993, in Denver, Colorado, the specifics pertaining to this building were released. The building is an academic building at a U.S. university used for classrooms, laboratories, and offices. Between the fall (training period) and spring (test period) semesters, the entire computer science department (computers and all!) relocated across campus, accounting for the large biases in the test sets. The apparent time constant for this structure was also estimated to be longer than 24 hours.

DATA SET B: TIME-INDEPENDENT DATA

The challenge of the second problem was very straightforward and required the prediction of the true beam insolation from the solar flux measured on four fixed devices—one each tilted and facing south, southeast, and southwest. The remaining fixed device measured the horizontal solar flux. The true beam insolation is an expensive measurement of the solar flux acquired by tracking the

TABLE 1
Summary of Data Set A

	WBE		WBCW		WBHW	
	SEP	Bias	SEP	Bias	SEP	Bias
	(kWh/hr)		(Btu/hr)		(Btu/hr)	
Training						
Present Work	62.5	0.39	0.41	-0.05	0.43	1E-4
Test						
Present Work (3rd)	74.3	50.0	0.68	-0.33	1.11	-0.96
1st	64.7	50.4	0.64	-0.31	0.53	-0.20
2nd	73.6	65.6	0.64	-0.29	1.07	-0.95
4th	106.	38.7	0.70	-0.41	1.04	-0.92

(The standard error of prediction [SEP or unnormalized CV] and mean bias [Bias] are given for the top four winners as well as our results for the training data.)

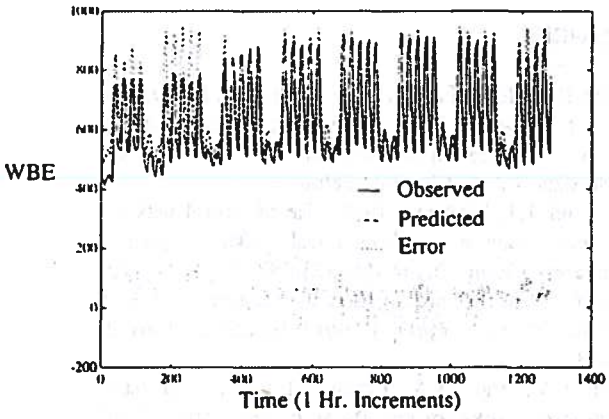

Figure 3 *Summary of present results for whole-building electric.*

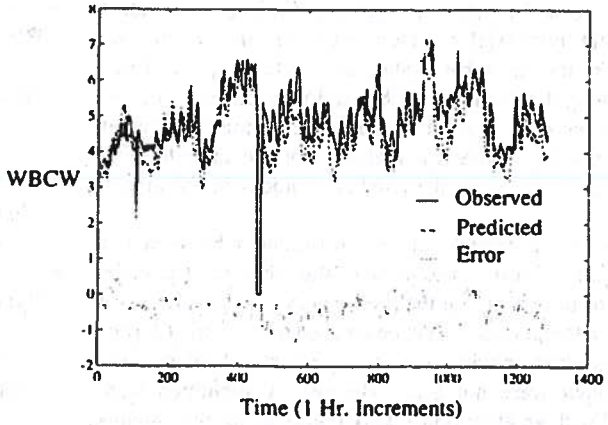

Figure 4 *Summary of present results for whole-building chilled water.*

sun's instantaneous position within fractions of a degree. The inputs into our model were the four solar fluxes, the day of the year, and the hour of the day. These six parameters were reduced to four independent variables using principal component analysis. The training of the 270 neural nets was performed on 2,400 data sets, with the models tested on an additional 900. Our standard error of prediction for the test set was 28.7 W/m^2, compared to 11.2 W/m^2, 9.65 W/m^2, and 17.2 W/m^2 for the first-, second-, and fourth-place finishers, respectively. Our results for the test set are shown in Figure 6. We believe our poor showing on this problem is directly related to our use of time (day and hour) as an input parameter. During optimization, the other winners found that these time parameters were unnecessary and introduced noise into the correlation.

SUMMARY

A new data-modeling technique employing an ensemble of neural nets has been presented. The method is particular-

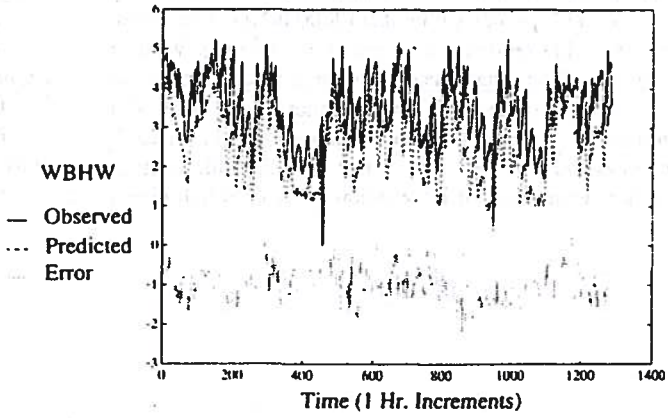

Figure 5 *Summary of present results for whole-building hot water.*

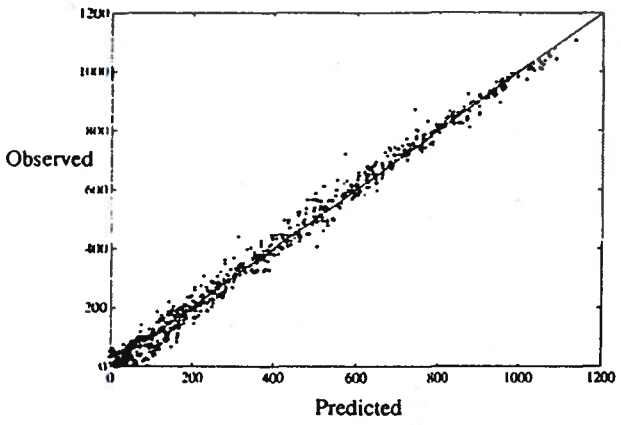

Figure 6 *Summary of present results for true beam insolation model.*

ly well suited for data sets with insufficient independent parameters, nontrivial measurement error, and/or few data points. For the ensemble model, more data samplings (e.g., new training data constructed by randomly picking from all the available data) are preferable to increasing the number of topologically different neural nets for the same training data set (e.g., changing the number of nodes in the hidden layer).

One of the most significant differences between our application and results and those of the other entrants is the amount of time spent on the problem. As stated earlier, we made no attempt to construct better models. Different forms, i.e., transforms, moving averages, products of inputs, etc., of the input were not tried. The results submitted were obtained with an afternoon's work compared to two months for the first-place winner and six weeks as the average time spent by the other contributors.

In retrospect, some very simple preprocessing of the time variables should have been performed in our analysis. For example, the hour of the day in data set A should have been represented by a sine function with a 24-hour period. In our "blind" approach, midnight (2400 hr) and one minute later (0001 hr) represented extremes in the input when in reality they should have nearly the same value. The winner used time series analysis to construct inputs and also identified more than 13% of the training data in data set A as outliers and removed these points from the fitting. All of the other winners spent large amounts of time optimizing the inputs into their respective models. This effort certainly improved their models but required significantly more man-hours. It is interesting to note that the winner spent the most time on the problem, which leaves in doubt the organizers' prime objective of rigorously comparing various methods.

The neural net approach embodied in the program used in this work has proved to be a state-of-the-art modeling technique, robust enough to give even reasonable results in "blind" applications. The robustness of our approach is a result of the algorithm's preprocessing, training procedure, and ensemble neural nets combined with fast workstations—a brute-force solution to modeling. We believe the best models, in the absence of a truly fundamental model, will combine a simple physical model with a neural net. This combination of physical models augmented by "smart" fudge factors (neural nets) provides both the ability to extrapolate beyond the conditions represented in the training data and the accuracy needed for process optimization.

REFERENCES

Admoaitis, R.A., R.M. Farber, J.L. Hudson, I.G. Kevrekidis, and A.S. Lapedes. 1990. Application of neural nets to system identification and bifurcation analysis of real world data. LA-UR-09-515, February.

Bhat, N., and T.J. McAvoy. 1990. Use of neural nets for dynamic modeling and chemical process systems. *Computers Chem. Engng.* 14: 573-583.

Cybenko, G. 1989. Approximation by superposition of a sigmoid function. *Math. Control Signals Systems* 2: 303-314.

Leonnard, J.A., and M.A. Kramer. 1991. Radial basis function networks for classifying process faults. *IEEE Control Systems*, p. 31.

Moody, J., and C.J. Darken. 1989. Fast learning in networks of locally tuned units. *Neural Comp.* 1: 281-294.

Noble, B., and J.W. Daniel. 1977. *Applied linear algebra*, p. 433. Englewood Cliffs, NJ: Prentice-Hall, Inc.

Poggio, T., and F. Girosi. 1990. Regularization algorithms for learning that are equivalent to multilayer networks. *Science* 247: 978-982.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323: 533-536.

Wasserman, P.D. 1989. *Neural computing: Theory and practice*. New York: Van Nostrand Reinhold.