ESL-PA-94/07-02 (4)

OR-94-17-4

USING ARTIFICIAL NEURAL NETS TO PREDICT BUILDING ENERGY PARAMETERS

William J. Stevenson, Ph.D., P.Eng.

ABSTRACT

Artificial neural nets were used as nonlinear function approximators on two data sets of building energy parameters and solar radiation data. During the modeling (training) phase, the data to be predicted were unavailable, providing a "blind" test of the technique.

The first time series consisted of building energy "inputs" (such as solar radiation and temperature) for September-December 1989 and required the prediction of energy use for January-February 1990. The extrapolation was performed with only the data immediately at hand. Although results for chilled- and hot-water use were acceptable, the prediction of electricity use would have benefitted markedly from easily available additional information, such as working and nonworking days.

The second time series required the prediction of beam solar insolation from four global directional measurements. This was an interpolation problem, and good predictions were achieved for this data set.

Conjugate gradient and cascade correlation neural net programs were used.

INTRODUCTION

Two data sets of energy-related parameters were made available to the scientific community in the form of a competition in order to determine which techniques were able to extract enough information from these data to predict accurately events not contained in the data. The data will be described as they are analyzed.

The technique of artificial neural nets (Lau 1992; Rumelhart et al. 1988) was used as it can be used for nonlinear, nonparametric, function approximation and makes very weak assumptions about the data given. A correctly configured neural network will learn to produce the required output even though the relationship between inputs and outputs may be difficult to describe.

The method requires that a network of simple computational elements, or neurodes, be constructed that will produce the "correct" output from the given input data. The elements are connected by adjustable weights and a (usually) nonlinear transfer function. The weights are adapted, or trained, by iteratively presenting the input data and the required output, thus minimizing the output error in a least-squares fashion. For computational ease, the neurodes in most networks are arranged in layers, with at least an input and output layer, but generally one or more "hidden" layers that are not accessible from outside the network. These hidden layers and the internal representation they form are crucial to the operation of neural nets.

Although subject to problems such as long training times and the difficulty of deciding on the correct network architecture, the technique can be usefully applied. Artificial neural nets have been used successfully in many fields, including business prediction (Yamamoto and Zenious 1993), control systems (Bozich and MacKay 1990), diagnosis (Marko et al. 1989), and medicine (Cheung 1989).

In this case, the network has to be trained to generalize (produce acceptable results for previously unseen data) and should provide smooth estimates from the given input data rather than perform as a reference table (sometimes known as being overtrained).

GENERAL APPROACH

The following decisions are important in setting up a neural network:

- Which learning algorithm will be used?
- Which variables are important?
- What is the optimum net size or configuration (architecture)?
- What is the best scaling or representation?
- Which software should be used?

The supervised back-propagation learning algorithm was used, as it is well documented and many computer implementations are available. The remaining decisions are interrelated and need to be approached iteratively.

It is important to include all significant variables but not to confuse the network with irrelevant or redundant information. At the very least, too much information will increase the network training time and prevent the network from learning adequately. In this work, the significance of different inputs was determined iteratively by training networks with different inputs and comparing the results for the different networks.

William J. Stevenson is currently an engineer in the Division of Manufacturing and Aeronautical Systems Technology, Council for Scientific and Industrial Research, Pretoria, South Africa.

THIS PREPRINT IS FOR DISCUSSION PURPOSES ONLY, FOR INCLUSION IN ASHRAE TRANSACTIONS 1994, V. 100, Pt. 2. Not to be reprinted in whole or in part without written permission of the American Society of Heating, Refrigerating and Air-Conditioning Engineers, Inc., 1791 Tullie Circle, NE, Atlanta, GA 30329. Opinions, findings, conclusions, or recommendations expressed in this paper are those of the author(s) and do not necessarily reflect the views of ASHRAE. Written questions and comments regarding this paper should be received at ASHRAE no later than July 6, 1994.

There are no reliable guidelines for deciding the number of neurodes in a hidden layer, nor how many hidden layers to use. The convergence proofs (for example, Hornik et al. 1989) merely state that convergence with a minimum of three layers (input, hidden, and output) is possible. Others (such as Chester 1990) have shown that there are occasions when two hidden layers may produce better results. Networks with more than two hidden layers are rare, due mainly to the difficulty (and time) of training them. As the nets used in this case are very small, the number of hidden neurodes were decided by trial and error as described below. Another method would be to prune unnecessary neurodes (Hassibi and Stork 1992). Pruning could also be used to determine which inputs are significant.

As the networks in this research are required to generalize, the method of cross-validation and early stopping (Jervis and Fitzgerald 1993) was used to determine both the best network configuration and at which stage the network was overtrained. In this method, the available training data are partitioned into a reduced training set and a validation set, the validation set being representative of the entire data set. Here the validation set was constructed by extracting parts of the original training set at different points along the given time series. A number of networks with different numbers of hidden neurodes are run, and the network with the minimum validation error is used as the optimum configuration. The network is trained for a limited number of iterations on the reduced training set and then tested on the validation set. Initially, the error on both the training set and the validation set should fall; however, once the validation error starts to rise, it is clear that there are unmodeled dynamics in the net that it is not modeling. Further training is counterproductive.

The network is then retrained using the entire data set with the optimum configuration and the optimum number of learning iterations. A few different sets of starting weights are used, and the results from the best of these are used for the required predictions.

The input to neural networks needs to be scaled to the range of the nonlinear activation function used. A sigmoidal function was used in this work for all layers of the neural networks. The scaling, or representation, of data can also affect the learning speed of a network; a good choice of scaling or data manipulation can dramatically improve learning speed. This could include providing additional features in the form of products of features (Pao 1989). Options considered in this report include scaling in the range of either 0.1 to 0.9 or −0.5 to +0.5 in a linear fashion or scaling by dividing by the standard deviation before scaling to 0.1 to 0.9. This was done for each variable separately, both on the input and output data. The range 0.1 to 0.9 rather than 0 to 1 was used to avoid saturating the weights as the inputs approach the asymptotes of the sigmoid.

The choice of software affects both the scaling and the decisions related to the number of hidden nodes. The following software was available:

- Fahlman's cascade correlator 1,
- Fahlman's cascade correlator 2 (early version), and
- Barnard's conjugate gradient program optb.

The cascade correlator (Fahlman and Lebiere 1990) automatically sets up the number of hidden neurodes (in cascaded layers) so that the net itself determines the need for additional neurodes. It requires a number of parameters, however, to "tweak" it. The first version works best for classification, although it can be successfully used for function approximation. The second version is better at function approximation.

Barnard's conjugate gradient program is extremely fast and has no learning parameters that require setting. It does require, however, that the user set the number of hidden neurodes and layers.

As neural networks constitute a multiply-connected system, the effect of different variables is spread throughout the system and variables can affect each other. Thus the first data set, which has three output variables, was modeled as only one network rather than as three separate networks.

Three error criteria were used when comparing results from different networks: root-mean-squared error, used for quick comparisons; coefficient of variation (CV); and mean bias error (MBE). The latter two criteria are as defined by the organizers of the competition.

DATA SET A

This time series consisted of building energy "inputs" (such as solar radiation and temperature) and required the prediction of energy use (electricity, hot- and chilled-water use). Training data at hourly intervals were available for September-December 1989; the prediction required was for the period from January to February 1990. The nature of the variations in the data and the fact that this is an extrapolation make this a difficult task.

A large building probably has a large thermal inertia, which would make present conditions dependent on previous conditions. Two methods can be used to simulate this—a window sliding along the data or a recursive network architecture. A sliding window moving across the data produces a larger (and therefore slower learning) net than a net that is not time-dependent and is dependent on knowing how many previous time steps (lags) affect the present data. Recursive networks are extremely slow and often not practical.

Due to time constraints, only a few of the available options and permutations were investigated. As a departure point, the well-tried and trusted method of linear scaling from 0.1 to 0.9 without any lag effects was used with Barnard's optb program. Information outside the given data set was also ignored; as will be seen later, this was a mistake. To avoid problems with any of the test data input variables being outside the range of the learned inputs, the day, month, and year (effectively day of year) information was ignored. The scaled-hour information was included to allow for diurnal variation.

The validation data set consisted of approximately one-third of the total training set. Approximately one-ninth of the total set was taken from the beginning, middle, and end of the training set to create a single validation set. Although this created discontinuities in the validation data, this effect was not critical as lagged information was not used.

A net configuration of five inputs (hour, temperature, humidity, solar flux, and wind speed), six hidden neurodes, and three outputs (electricity, hot- and chilled-water use) was found by trial and error and the early stopping method as previously described.

Results

The predictions for electrical, hot-water, and chilled-water use can be seen in Figures 1 through 3.

It is clear from the electrical prediction (Figure 1) that the network has been unable to predict the reduced consumption of nonworking days adequately, and often underpredicts the consumption on working days.

The hot-water-use prediction follows the actual use well but is offset from it. The reason for this is presumably to be found in the offset in the scatter diagram between the training and test data sets, which is shown in Figure 4. The cause of this offset is unknown.

The predicted chilled-water use (Figure 3) follows the actual use more closely but still has a slight offset.

Discussion of Results

It is clear that a feature for working and nonworking days (e.g., 0 and 1) should be added, even if it is information outside the given data. This might even be extended to a continuous variable, or multiple features, to represent the "degree" of the working day. For example, there are long breaks, such as vacations, holidays (Thanksgiving, Christmas/New Year), and shorter breaks, such as weekends and single public holidays. The lower electricity consumption for the first two weeks of the year may represent the university vacation when undergraduate students are not present. This information was not available for the competition, but using it for a complete building study would improve the prediction.

A recalculation using an additional binary valued feature (0.1 for working, 0.9 for nonworking day) reduced the coefficient of variation from 16.95 to 13.77 for the electrical use. However, the working/nonworking feature is largely irrelevant to the hot- and chilled-water consumption, and its addition considerably worsens the error for these two parameters. In this case, it would be better to split the data sets into either two (one electrical-use net and one water-use net) or three (a net for each output parameter). The electrical-use net would have an additional "working day" feature.

Smaller improvements may be available by coding the day of year as well in the form of sin(day_of_year) or cos(day_of_year). This could improve prediction of the trends in the use of hot and chilled water and also include

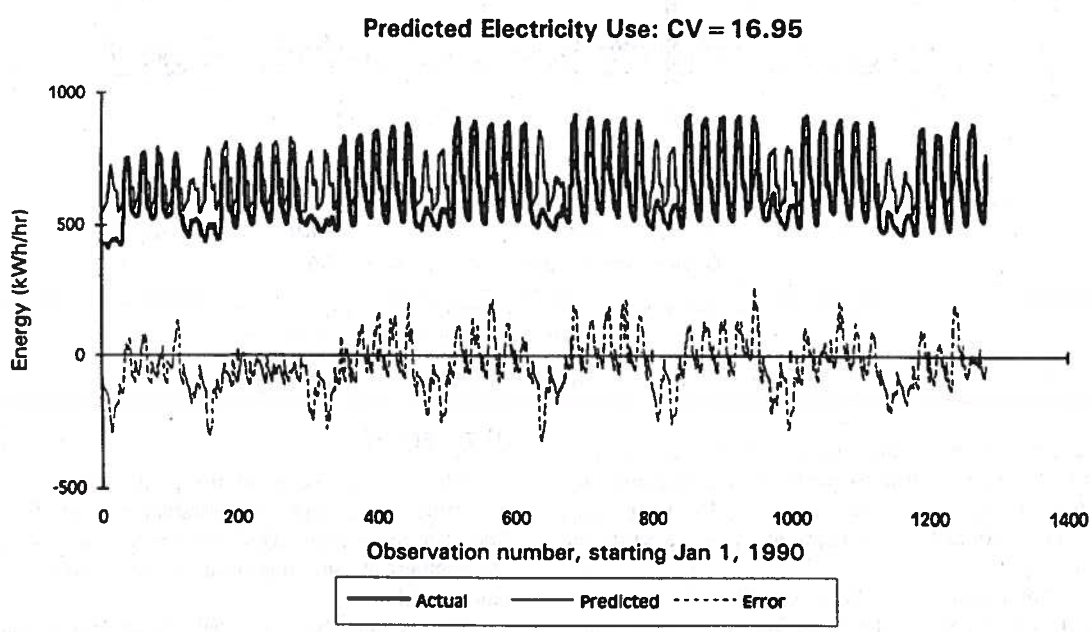

Figure 1 Prediction of whole-building electrical use.

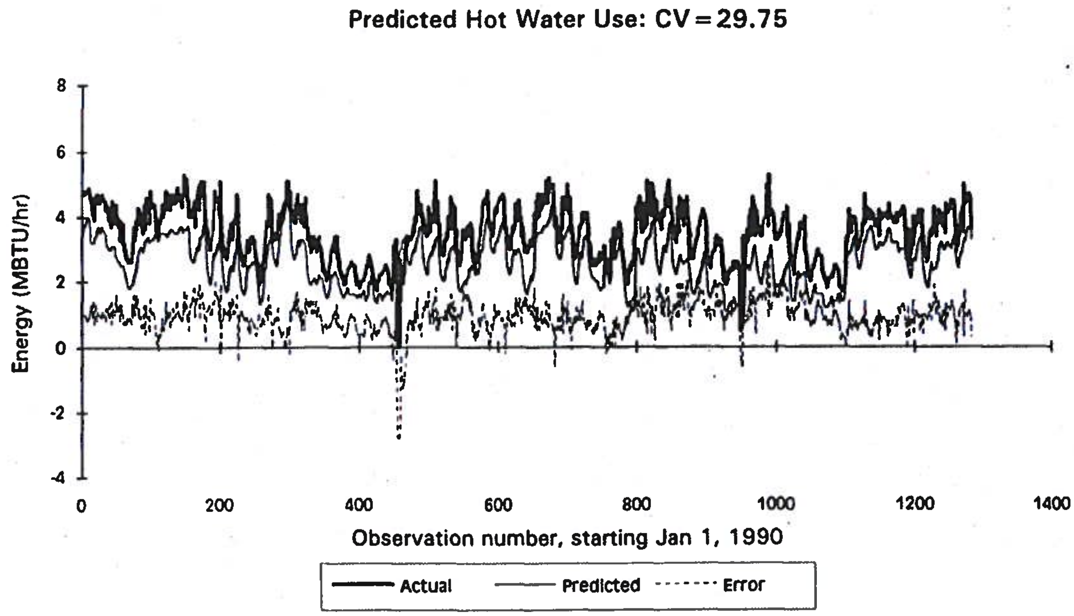

Figure 2 Prediction of whole-building hot-water use.

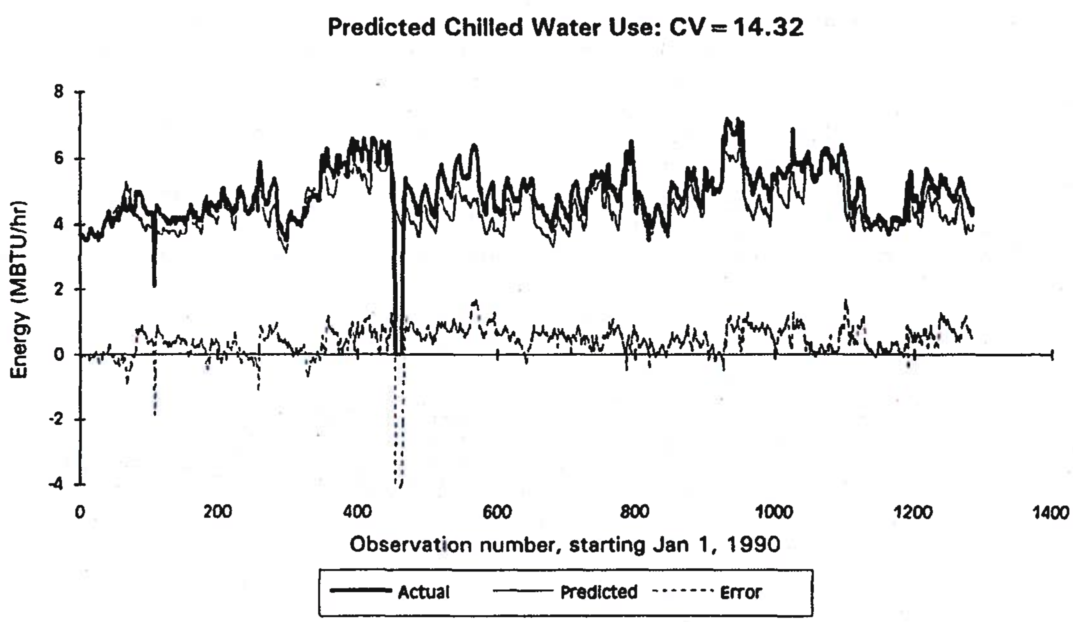

Figure 3 Prediction of whole-building chilled-water use.

a more cyclic effect. Other improvements could include lagged data to include measurements from the previous two days and previous week (12-, 24-, and 168-hour lags). Greater care would then be required in constructing the validation set.

The prediction as a whole would be improved by having data available for the entire year or at least representative portions of each season.

DATA SET B

This data set required the prediction of solar beam insolation from four global measurements and time information. The prediction period was within the time period of the training data and thus could be described as an interpolation problem.

The training data consisted of contiguous hourly data, but the test data were not contiguous. Thus, even though

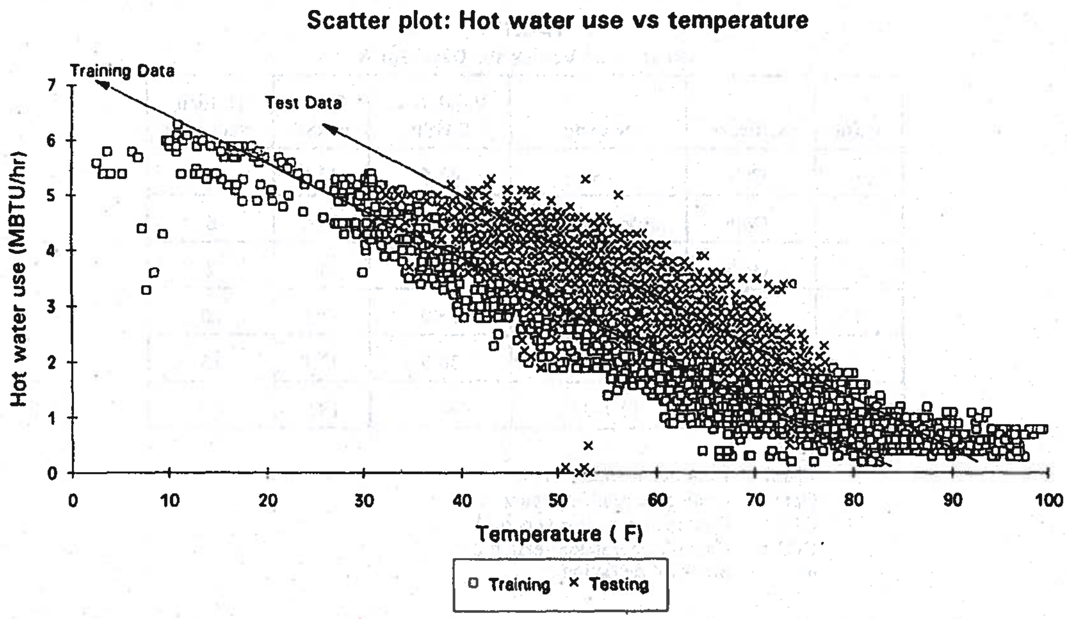

Figure 4 Scatter diagram of training and test data—hot-water use vs. temperature.

present data might be dependent on previous data, this lagged effect could not be used.

As before, the supplied training information was again split into a validation and a reduced training set. The validation set consisted of approximately one-quarter of the original data set. In order to include possible seasonal variations in the full data set and thus be representative of it, the validation set consisted of extracts from the beginning, middle, and end of the original.

There are three options for representing the time and date information—it could be ignored, coded together (as supplied), or separated. A few initial training runs showed that time and date together produced better results.

Three scaling options were investigated: scaling by standard deviation, linearly to the range 0.1 to 0.9, and linearly to the range −0.5 to +0.5. The first version of cascade correlator also has a nonsymmetric self-scaling sigmoid option, and this was used as well. Data were scaled separately, variable by variable.

The root-mean-square error for different net scaling and software can be seen in Table 1 as a means of determining an optimum combination of net size and software. The first five cases represent trials to determine the best combination of scaling, software, and number of hidden neurodes. Although case 4 using cascade correlation has the lowest error, the number of hidden nodes seemed too high, and overtraining was still suspected. The error for case 3 using optb is only slightly worse and this configuration was used. However, the number of neurodes and hidden layers in case 4 implies that perhaps more than one hidden layer would work better. A number of trials with different numbers of neurodes in a second hidden layer were run with considerably improved results.

The final net used five input neurodes (date/time and four solar fluxes), two hidden layers with first seven then four neurodes, and a single output neurode (true beam insolation). Linear scaling of the input and output variables to the range 0.1 to 0.9 was used.

Results

The prediction for true solar insolation can be seen in Figure 5. The prediction compares very well to the given data, as can be seen also in the crossplot (Figure 6), with small random error that shows no bias.

CONCLUSIONS

The results show that artificial neural nets can be used to determine the inherent relationships between building energy parameters and predict results for previously unseen inputs.

The results for the electrical use in the first data set would probably have been considerably improved by using additional information (at least working vs. nonworking days). This once again proves that it is important to find all possible relevant features and supply them to the net.

The prediction also would have been improved by supplying more data, at least covering significant portions of each season.

ACKNOWLEDGMENTS

The cascade correlator software was written by Professor Scott Fahlman at Carnegie Mellon University; the

TABLE 1
Best Error Values for Data Set B

Case	Software	Scaling	Validation RMSE	Total RMSE	Hidden Neurodes
1	CC1	σ	43.5	30.6	10
2	Optb	σ & (0.1-0.9)	33.1	30.3	6
3	Optb	(0.1-0.9)	35.7	24.1	7
4	CC1	(0.1-0.9)	33.0	23.8	20
5	CC2	(−0.5-+0.5)	36.9	32.1	15
6	Optb	(0.1-0.9)	29.5	19.9	7-4

Notes:

RMSE: Root-mean-square error
Optb: Conjugate gradient neural net
CC1: Cascade correlator version 1
CC2: Cascade correlator version 2
σ: Standard deviation

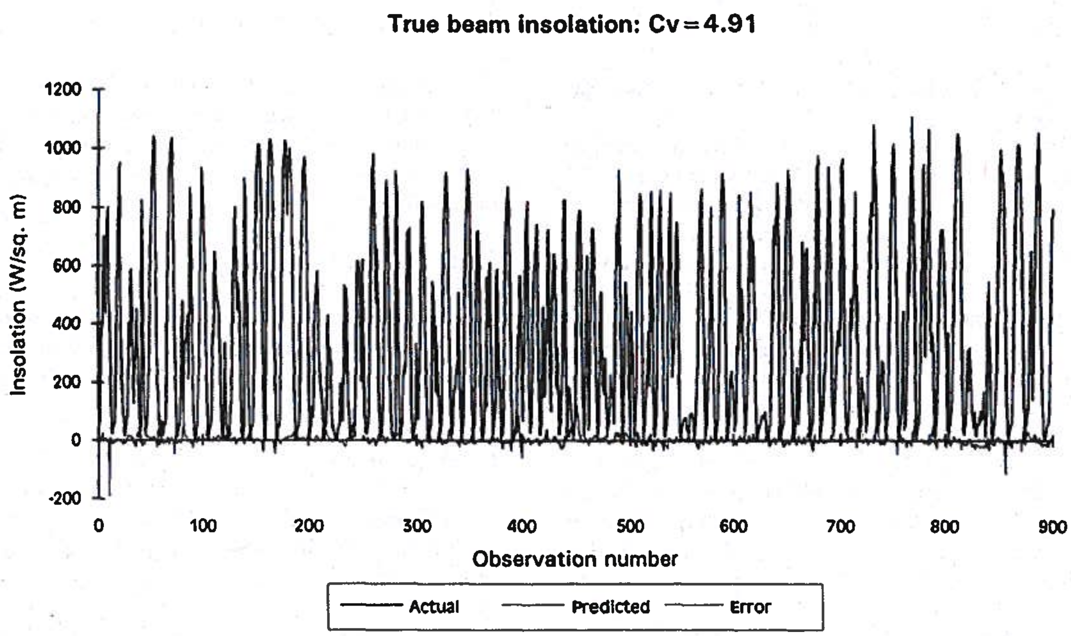

Figure 5 Prediction of true-beam insolation.

conjugate gradient software was originally written by Professor Etienne Barnard, currently at the Oregon Graduate Institute, Portland.

REFERENCES

Bozich, D.J., and H.B. MacKay. 1990. Vibration cancellation at multiple locations using neurocontrollers with real time learning. *Proceedings of the International Joint Conference on Neural Nets* 2: 775-780.

Chester, D.L. 1990. Why two hidden layers are better than one. *Proceedings of the International Joint Conference on Neural Nets* 1: 265-268.

Cheung, J.Y. 1989. Applications of neural networks in medical care. *Neural Network News*, December, pp. 8-9. Atlanta: AI Week, Inc.

Fahlman, S.E., and C. Lebiere. 1990. The cascade-correlation learning architecture. CMU-CS-90-100. Carnegie Mellon University, School of Computer Science.

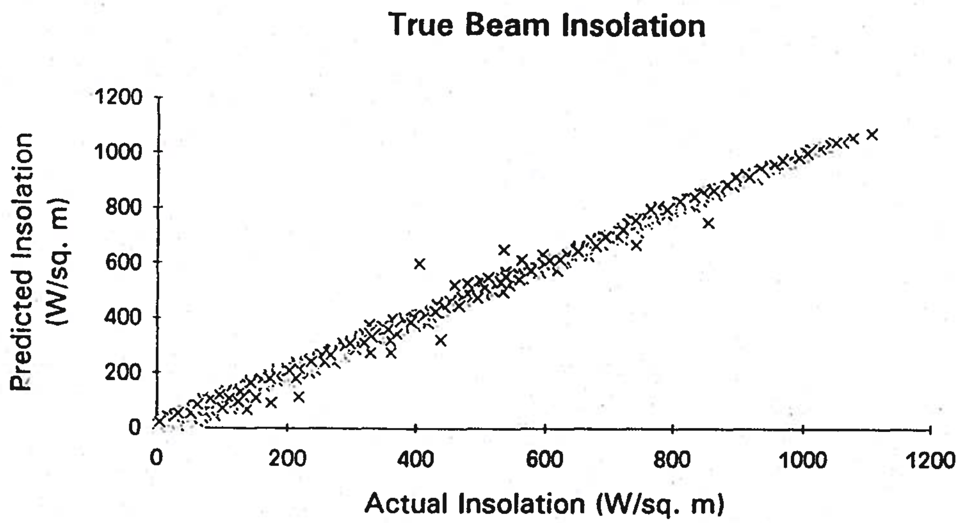

Figure 6 *Crossplot of actual and predicted solar insolation.*

Hassibi, B., and D.G. Stork. 1992. Second order derivatives for network pruning: Optimal brain surgeon. Neural Information Processing Conference.

Hornik, K., M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* (2)5: 359-366.

Jervis, T.T., and W.J. Fitzgerald. 1993. Optimization schemes for neural networks. Technical Report CUED/FINFENG/TR 144. Cambridge, UK: Cambridge University, Engineering Department.

Lau, C.G.Y., ed. 1992. *Neural networks: Theoretical foundations and analysis*. New York: Institute of Electrical and Electronics Engineers, Inc.

Marko, K.A., J. James, J. Dosdall, and J. Murphy. 1989. Automotive control system diagnostics using neural nets for rapid pattern classification of large data sets. *Proceedings of the International Joint Conference on Neural Networks* 2: 13-16.

Pao, Y.-H. 1989. *Adaptive pattern recognition and neural networks*. New York: Addison Wesley.

Rumelhart, D.E., G.E. Hinton, and R.J. Williams. 1988. Learning internal representations by error propagation. *Parallel Distributed Processing*, vol. 1: Foundations, chap. 8, D.E. Rumelhart, J.L. McClelland and the PDP Research Group. Cambridge, MA: MIT Press.

Yamamoto, Y., and S.A. Zenious. 1993. Predicting prepayment rates for mortgages using the cascade-correlation learning algorithm. *The Journal of Fixed Income* 2(4): 86-96. Institutional Investor, Inc.