A visual analysis of learning rule effects and variable importance for neural networks in data mining operations

Article *in* Kybernetes · August 2004

DOI: 10.1108/03684920410534461

CITATIONS
9

READS
291

2 authors:

Kelly E. Fish
Arkansas State University - Jonesboro
6 PUBLICATIONS **180** CITATIONS

SEE PROFILE

Richard Segall
Arkansas State University - Jonesboro
60 PUBLICATIONS **295** CITATIONS

SEE PROFILE

A visual analysis of learning rule effects and variable importance for neural networks in data mining operations

Kelly E. Fish and Richard S. Segall
Department of Economics and Decision Sciences, State University, College of Business, Arkansas State University, Arkansas, USA

Keywords *Cybernetics, Neural nets, Data analysis, Learning processes*

Abstract *This study demonstrates two visual methodologies to support analysts using artificial neural networks (ANNs) in data mining operations. The first part of the paper illustrates the differences and similarities between various learning rules that might be employed by ANN data miners. Since different learning rules lead to different connection weights and stability coefficients, a graphical representation of the data that provides a novel visual means of discerning these similarities and differences is demonstrated. The second part of this research demonstrates a methodology for ANN model variable interpretation that uses network connection weights. It uses empirical marketing data to optimize an ANN and response elasticity graphs are built for each ANN model variable by plotting the derivative of the network output with respect to each variable, while changing network input in equal increments across the range of inputs for each variable. Finally, this paper concludes that such an approach to ANN model interpretation can provide data miners with a rich interpretation of variable importance.*

1. Introduction

Artificial neural networks (ANNs) are often used in data mining operations due to their ability to handle broad data fields in determining complex functional relationships. ANNs' ability to effectively mine the data is affected by the chosen learning rule. The first part of this paper illustrates the differences and similarities between various learning rules that might be employed by ANN data miners. Since different learning rules lead to different connection weights and stability coefficients, we demonstrate a graphical representation of the data that provides a novel visual means of discerning these similarities and differences.

The second author would like to acknowledge the support of his part of the research by a grant from the US Air Force Office of Scientific Research (AFOSR) as administered by the National Research Council (NRC) in Washington, DC. The second author also most gratefully acknowledges the US Air Force School of Aerospace Medicine (AFSAM), and specifically the Biomechanisms and Modeling Branch at Brooks AFB in San Antonio, Texas for providing the necessary support for this research while the second author was in residence at the Laboratory as a Senior Research Fellow.

Kybernetes
Vol. 33 No. 7, 2004
pp. 1127-1142

0368-492X
DOI 10.1108/03684920410534461

Even though ANNs have established themselves as a prominent data mining tool, they are still criticized for their failure to explain results. Central to this criticism is their inability to provide interpretation of the network connection weights. The second part of this study demonstrates a methodology for ANN model variable interpretation that uses network connection weights. We use empirical marketing data to optimize an ANN and we build response elasticity graphs for each ANN model variable by plotting the derivative of the network output with respect to each variable, while changing network input in equal increments across the range of inputs for each variable. We conclude that such an approach to ANN model interpretation can provide the data miners with a rich interpretation of variable importance.

2. Learning rule analysis

This part of the paper emphasizes the data mining applications of the research performed earlier by Segall (1995, 1996) that also illustrated the differences between the various learning rules The classical learning rule was developed by Hebb (1949), followed by more recent adaptations by Abbott and Kepler (1989), Diedrich and Opper (1987), Gardner (1988) and Krauth and Mezand (1987). The data mining determines the best set of training parameter settings for each of the learning rules, as well as the relative performance of the learning rules among each other.

The data warehouse used in this neural network research is shown in the flow diagram of Figure 1 and the sample data cube of Figure 2. The data warehouse consists of data that represent the stored patterns in memory for each of the neurons of the model and the patterns generated by each neuron. The data warehouse also includes the synaptic or coupling strengths between each set of neurons, the number of patterns, the number of neurons in the model, and the stability coefficients for each pattern in memory by each neuron. The latter of stability coefficients are calculated by using a cross product of the synaptic strengths with their stored patterns.

The database for this neural network application includes multiple input files that are combined to create the source data set. Database management is crucial to the application of the models for the learning rules of neural networks as the source data set is then split into a training set, testing set, and a validation set. For the database application in this part of the paper, we use the 26 letters of the alphabet as the training set, i.e. the data set used for images learned by the neural network that are inputted continuously in random order. The data values for the synaptic strengths are normalized to a range of 0.0-1.0. A neural network utility (NNU) within the computer code used provides a data translation function that performs all of the required symbol mapping and numeric scaling operations. This database management technique is more fully shown in Figure 1 as referenced earlier for construction of the data warehouse.

According to Bigus (1996), when using real data to train a neural network, it is typical to have 98 percent of the data representing normal conditions, and

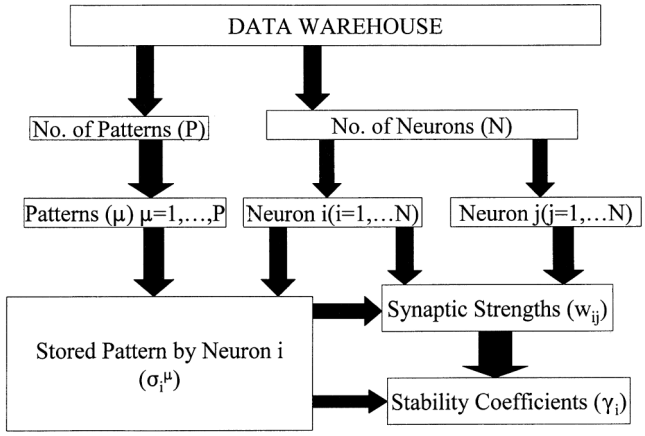


Figure 1.
NNU for neural network data warehouse

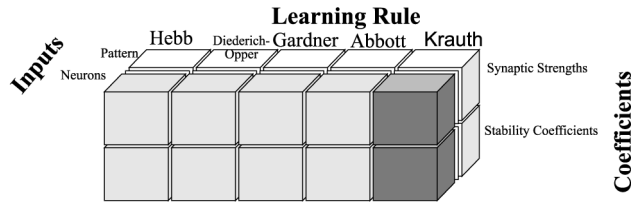


Figure 2.
A sample data cube for data mining of neural network database

have a small percentage of examples for the cases of abnormal conditions that we really want to detect. One technique to increase the percentage is to simply duplicate the number of training examples that contain the number of under represented class of training pattern. This technique is useful for our case study of non-numeric data of the letters of the alphabet. A technique for the numeric input data is to take the small number of test cases and modify their input values by small perturbations as an analogous to injecting small amounts of random noise, and then using these perturbed inputs as additional training cases. Another option is to create the additional training examples by hand.

In addition to the management of the data, a major concern in neural network data mining is the quality of the data. Most databases contain incomplete and inaccurate data. When limited data are available, one must estimate the values for the missing data. The most common technique is to set the data fields equal to the mean or median value for numeric data, and to the mode for non-numeric data. As noted by Simonoudis *et al.* (1995), outliers of data also can severely impact the performance of a neural network model.

Databases consisting of the collection of synaptic strengths and stability coefficients were created upon 100 iterations of each learning rules as trained with the 26 letters of the alphabet and are shown using frequency histograms in Figures 3(a)-(f) and 4(a)-(f), respectively. Table I and Figure 5 provide database characteristics of the synaptic strengths for the learning rules

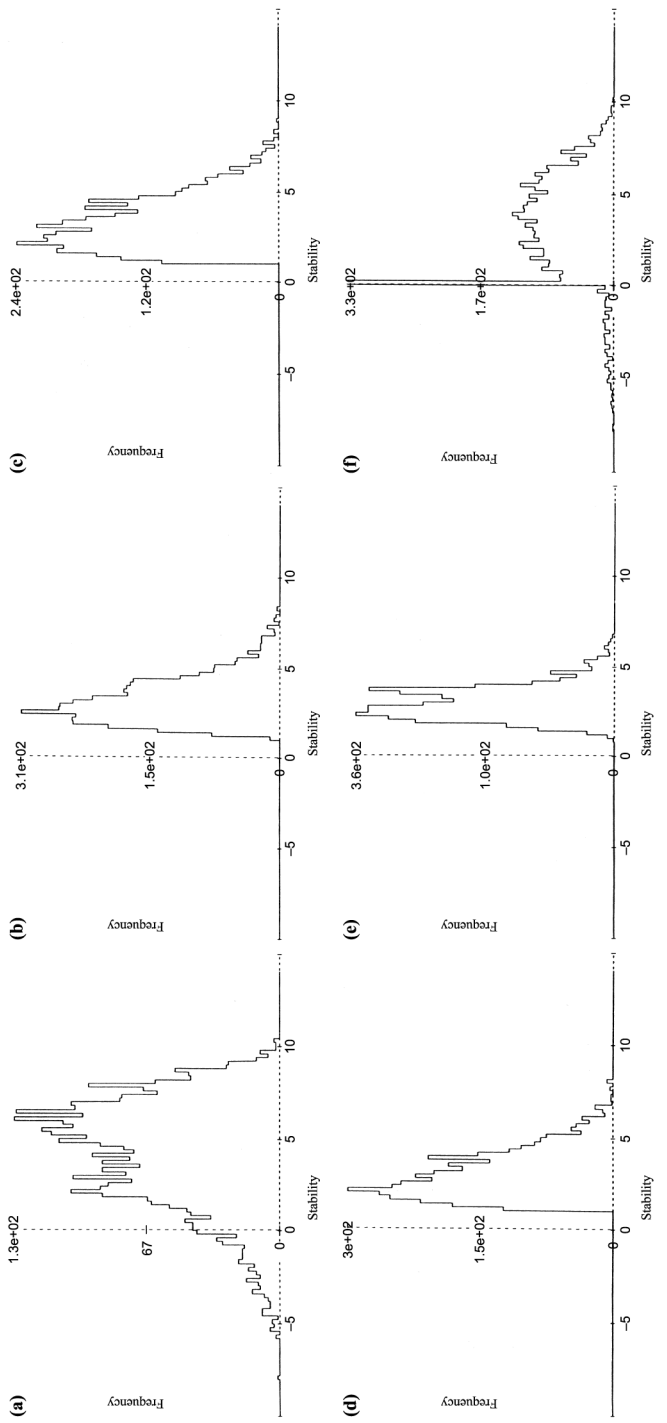
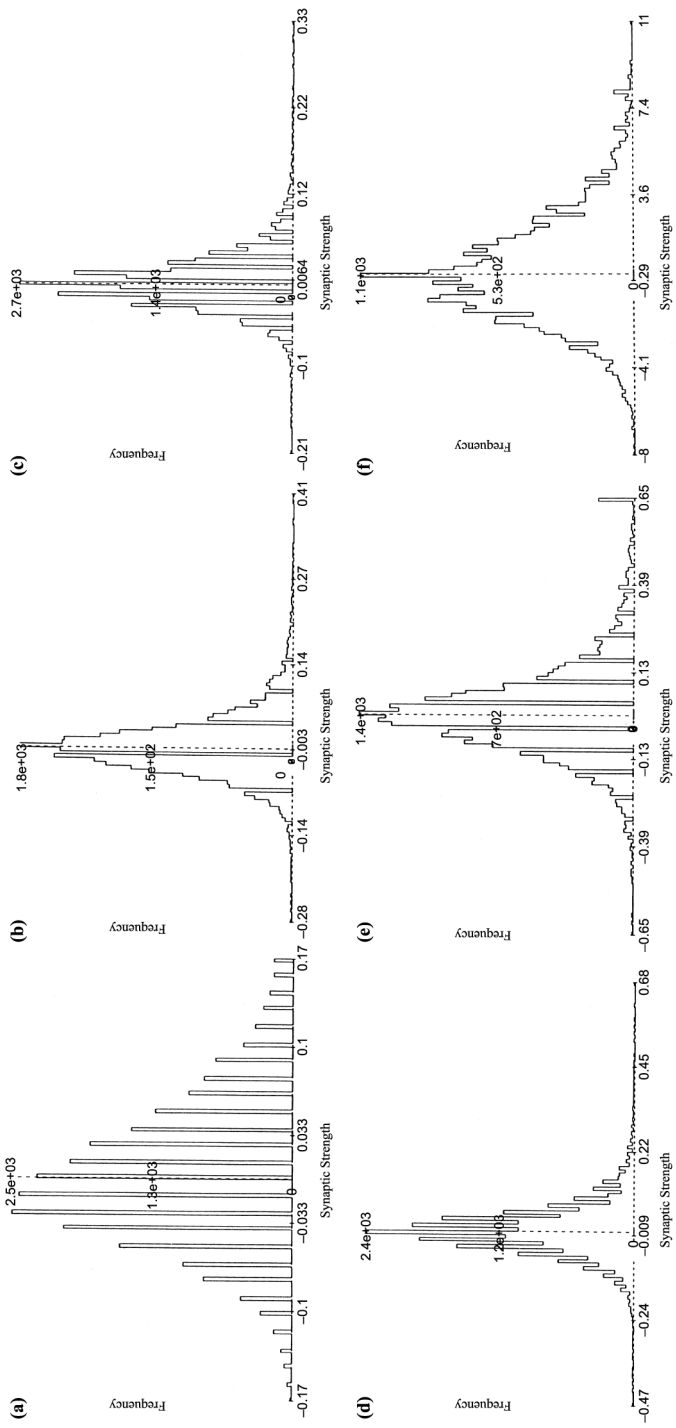


Figure 3.
Synaptic strength histograms

Notes: (a) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of Hebb; (b) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of Diedrich-Opper; (c) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of Gardner; (d) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of Abbott; (e) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of Krauth; (f) Synaptic strength histogram for the network trained with 26 capital letters of alphabet using learning rule of sequential synaptic coefficients



Notes: (a) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of Hebb; (b) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of Diedrich-Opper; (c) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of Gardner; (d) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of Abbott; (e) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of Krauth; (f) Stability coefficient histogram for the network trained with 26 capital letters of alphabet using learning rule of sequential synaptic coefficients

Figure 4.
Stability coefficient histograms

obtained from the histograms shown in Figure 3. Figure 5 omits the data for the Sequential Rule because as indicated in Table I its magnitude is so huge in comparison to the other five rules, that the five other synaptic strengths would not be visible in the plot. Table II and Figure 6 provide database characteristics of the stability coefficients for all six learning rules obtained from the histograms shown in Figure 4.

Figure 3 shows the frequency histogram of the database of synaptic strengths for the learning rules. The histograms of Figure 3 indicate that all of

Table I.
Summary of synaptic strengths for learning rules ($\kappa = 1.0$ and $C = 1.0$)

Learning rule	Minimum	Maximum	Range	Average	Mean square average	Symmetry	Frequency ($\times 10^3$)	Number of modified synopses
Hebb	−0.1667	0.1667	0.3334	0.0033	0.0564	1.000	2.5	1
Diederich-Opper	−0.2756	0.4103	0.6859	0.0038	0.0511	0.920	1.8	5
Gardner	−0.2115	0.3333	0.5448	0.0027	0.0362	0.876	2.7	17
Abbott	−0.4679	0.6795	1.1474	0.0054	0.0750	0.811	2.4	10
Krauth	−0.6474	0.6464	1.2948	0.0147	0.1419	0.862	1.4	100[a]
Sequential	−8.0064	11.2949	19.3013	0.1477	2.3364	0.773	1.1	100[a]

Note: [a]Failed to converge within 100 iterations

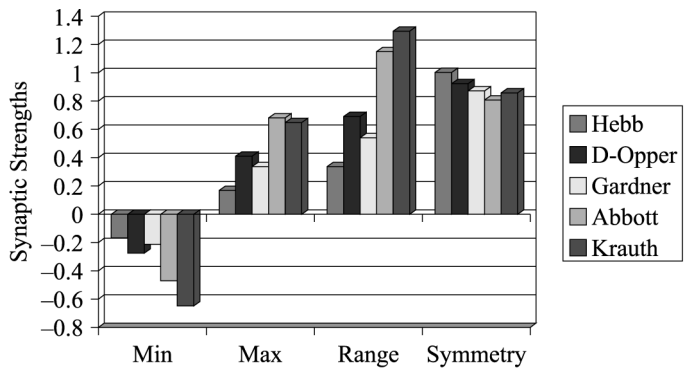


Figure 5.
Synaptic strengths for five learning rules

Table II.
Summary of stability coefficients (γ) for learning rules ($\kappa = 1.0$ and $C = 1.0$)

Learning rule	Minimum	Maximum	Range	Average	Frequency ($\times 10^3$)
Hebb	−7.825	10.392	18.217	4.141	1.3
Diederich-Opper	1.038	8.331	9.369	3.213	3.1
Gardner	1.000	8.898	9.898	3.324	2.4
Abbott	1.000	8.040	9.040	2.963	3.0
Krauth	0.958	6.716	7.674	2.982	3.6
Sequential	−7.679	10.081	17.760	3.262	3.3

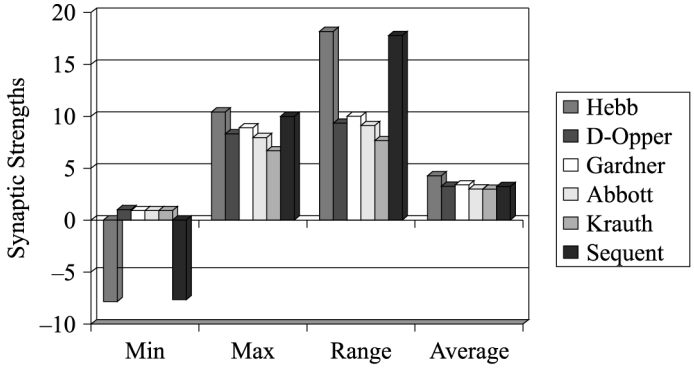


Figure 6.
Stability coefficients for six learning rules

the six learning rules, except that of Abbott's of 3(d) and the Sequential Rule of 3(f), have zero synaptic strength values in the database.

As referenced earlier, it should be noted that the single peak of the frequency histogram of the database of stability coefficients for the Sequential Rule in Figure 4(f) is almost double the height of the largest for the other remaining histogram bars. This single peak indicates an exceedingly high frequency of those neurons having zero-valued stability coefficients with this learning rule. This property is quite unlike the other learning rules with only that of Hebb's rule having any countable frequency for zero-valued stability coefficients.

Finally, Figure 7 shows the synaptic strengths and stability coefficients of the six learning rules listed in Tables I and II, respectively. The reader is referred to a more complete and extended version of this visual analysis of learning rules effects as presented by Segall (1995) where Kohonen maps and plots of activation of output neurons for multiple layer feed-forward networks trained using a Boolean function are presented. Kohonen maps for neural networks which are presented are "topology-preserving" maps for

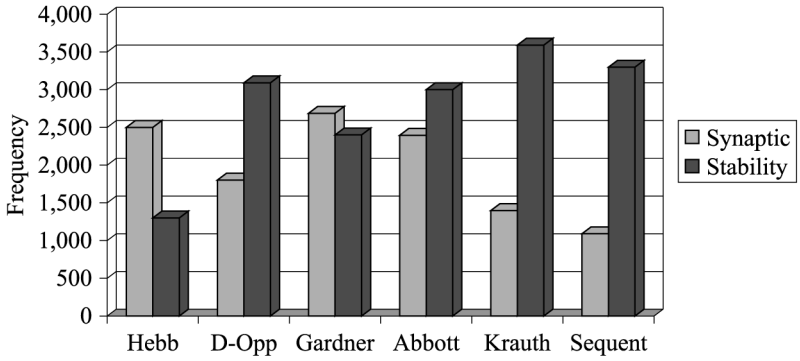


Figure 7.
Frequency of synaptic strengths and stability coefficients

representations of multidimensional continuous "sensory space" on a grid of neurons.

3. Variable importance analysis

When employed in data mining operations, ANN modelers know that the network is not using all of the variables (i e. data fields) in the data file. The ANN has the ability to ignore variables that do not contribute to the solution by giving them connection weightings very close to zero. The challenge of the researcher is to determine which variables are actually contributing to the network output. Often ANN modelers will build a series of models omitting one variable at a time. If model performance on a holdout sample does not suffer, then the model is not using that particular variable and it can be omitted from the model. Then end result is an ANN model that contains variables that contribute to network output. Yet, the exact amount of contribution from each variable remains a rough estimate.

In statistical modeling, *t*-tests are often used as an indicator of model variable importance. In a simple regression model with one predictor variable (x), the null hypothesis for the *t*-test is that a change in x yields no predicted change in model output (y), and it follows that x has no value in predicting y. In other words, if the predicted y's were plotted against the various inputs of x, the slope of the line would be zero. In a model with multiple predictor variables the null hypothesis for each variable *t*-test is that a particular variable has no additional predictive value over and above that contributed by the other predictor variables; specifically if all other x's had already been used in the model and then an additional predictor variable (x_j) was added, no improvement in prediction would result. In other words, if the predicted y's were plotted against the changes in x_j, the partial slope of the line would be zero. While *t*-statistics have yet to be developed for ANNs, we can extrapolate some of their logic in determining the variable importance by examining the actual partial slopes of lines based on the predicted y's, given the changes in respective x_j's.

The purpose of this part of the study is to illustrate an ANN approach to model variable interpretation. Data miners can use this methodology to interpret the variables of their ANN models. Data miners often mine their data with the ultimate goal of building some type of model of consumer behavior. For example, they often use neural networks to work retail scanner data and eventually build choice models that help to explain and predict purchases of different package goods such as peanut butter, catsup or coffee. For demonstration purposes, we choose to use a choice model that could have (although it was not) been developed by data mining coffee scanner data with an ANN.

3.1 Consumer choice model

Since its introduction 17 years back, the Guadagni and Little (1983) model has proven to be an accurate predictor of choice and has served as a standard by

which other models are measured. Moreover, many managers are familiar with it; thus, we choose it as the demonstration model for our study. When modeling a coffee market such as this study, the model assumes that the deterministic component of the utility that customer k gains from the purchase of a given choice alternative is a function of:

(1) the alternative's regular price,
(2) whether or not the brand was on promotion,
(3) presence or absence of a promotional price cut, and whether or not customer k's,
(4) previous, and
(5) second previous purchases were on promotion along with the customer's
(6) brand loyalty and
(7) size loyalty, which is a customer's loyalty to a particular size of coffee package (e.g. 16 versus 32 oz).

Mathematically, the deterministic component of utility ($Bx_{ik}(n)$) can be represented as:

$$Bx_{ik}(n) = B_1\mathrm{RP}_{ik}(n) + B_2\mathrm{PROM}_{ik}(n) + B_3\mathrm{PCUT}_{ik}(n) + B_4\mathrm{PRV}_{ik}(n) + B_5\mathrm{SPRV}_{ik}(n) + B_6\mathrm{BL}_{ik}(n) + B_7\mathrm{SL}_{ik}(n) \quad (1)$$

where $\mathrm{RP}_{ik}(n)$ = regular (depromoted) price of alternative i at the time of customer k's nth coffee purchase ($/oz); $\mathrm{PROM}_{ik}(n)$ = a 0 − 1 variable denoting the presence or absence of a promotion on alternative i at purchase occasion n; $\mathrm{PCUT}_{ik}(n)$ = promotional price cut on alternative i at the kth customer's purchase ($/oz); $\mathrm{PRV}_{ik}(n)$ = a 0 − 1 variable denoting whether or not customer k's previous purchase was a promotional purchase of an alternative with the same brand as alternative i; $\mathrm{SPRV}_{ik}(n)$ = a 0 − 1 variable denoting whether or not customer k's second previous purchase was a promotional purchase of an alternative with the same brand as alternative i; $\mathrm{BL}_{ik}(n)$ = customer k's loyalty toward the brand of alternative i at purchase occasion n; and $\mathrm{SL}_{ik}(n)$ = customer k's loyalty toward the size of alternative i at purchase occasion n.

The variable BL is more specifically defined as:

$$\mathrm{BL}_{ik}(n) = \alpha_b\mathrm{BL}_{ik}(n-1) + (1-\alpha_b) \times \left\{ \begin{array}{c} 1 \text{ if customer } k \text{ bought same brand of alternative } i \text{ at} \\ \text{purchase occasion } (n-1),\ 0 \text{ otherwise} \end{array} \right\} \quad (2)$$

where the carry-over constant is α_b, which represents the exponential decay rate of the impact of the previous ten purchases. The reader is referred to Guadagni and Little (1983) for an explication of how the particular rate (α) is

determined for both brand and size loyalty. Based on an earlier research involving coffee markets we use $\alpha_b = 0.875$ and $\alpha_s = 0.812$.

To start up BL, $BL_{ik}(n)$ is set to be α_b if the brand of alternative i was the first purchase in the data history of consumer k, otherwise it is set at $(1-\alpha_b)/(\text{number of brands} - 1))$, thereby ensuring that the sum of loyalties across brands always equals 1 for a consumer. To illustrate BL, suppose customer k bought alternative i on the first purchase occasion, $BL_{ik(1)}$, in such an instance $\alpha_b = 0.875$. On the second purchase occasion of customer k, the same brand is purchased resulting in an increase of brand loyalty: $BL_{ik(2)} = 0.890625 = (0.875(0.875) + (0.125)(1))$. On the next occasion suppose customer k selects another brand, resulting in a decrease in brand loyalty: $BL_{ik(3)} = 0.779296 = (0.875(0.890625) + (0.125)(0))$.

Size loyalty (SL) is analogous:

$$SL_{ik}(n) = \alpha_s SL_{ik}(n-1) + (1-\alpha_s) \times \left\{ \begin{array}{c} 1 \text{ if customer } k \text{ bought same brand of alternative } i \text{ at} \\ \text{purchase occasion } (n-1),\ 0 \text{ otherwise} \end{array} \right\} \quad (3)$$

where α_s is the carry-over constant for size. Initialization methodology for SL is the same as BL.

3.2 Data

In this part of the study, we use the same retail scanner data that was used by Kalwani *et al.* (1990) to develop an expected price model. Selling Areas-Marketing, Inc. (SAMI) collected the data, which consists of individual records of ground coffee purchases from four Kansas City supermarkets over a 65-week period. As in earlier studies, we clean the data set to remove consumers who joined or left during the 65-week period, who were extremely light users or consumers who had gaps in their reported purchases. Since retailers promote coffee brands and sizes separately we model *brand-sizes* as our choice alternatives. For example, Folger's has both a large size (3 lb) and a small size (1 lb) in our data set; each of these constitutes a different brand-size. We retain the top six brand-size combinations and these account for 82.4 percent of the total number of purchases and 87.1 percent of the market share. Each of the rest of the brands that were excluded has less than 1 percent market share. Twenty-five weeks of data are needed to initialize the *brand* and *size loyalty* variables leaving 40 weeks for the fitting/training and validation samples.

The data set does not contain any precise information on coupon, special point-of-purchase display or promotion. We thus follow the previously accepted practice of assuming a sales promotion when any two of the following three criteria are met.

(1) A price reduction lasting 1-4 weeks.
(2) An unusually high sales movement (i.e. store sales volume exceeding the mean level of unprompted sales by more than two standard deviations).
(3) The brand-size is featured in a newspaper, flyer, and/or bag stuffer during the week.

3.3 ANN model optimization

In this part of the study, we use a feed-forward network trained by the genetic adaptive neural network training (GANNT) algorithm (Dorsey *et al.*, 1994). Although similar to earlier genetic algorithms, GANNT has produced better results than prior efforts. For example, Whitley *et al.* (1990) reported an average error of 0.5 on the binary addition problem using a population size of 5,000 and running the algorithm for 500,000 generations. While Dorsey *et al.* (1994) were able to achieve an average error of 6.06×10^{-28} with a population size of 20 after fewer than 100,000 generations.

We want to know how well each model predicts market share for each brand-size, so use mean absolute error (MAE) as a measure of the model's ability to estimate brand-share. We calculate it by finding the weekly differences (errors) between the actual market share and predicted market share for each brand-size. We then average the absolute values of those errors over the N week period (Grover and Srinivasan, 1992). We then use a network optimization procedure similar to Agrawal and Schorling (1996) who were also dealing with scanner data. The optimization procedure results in an ANN that contains one output variable, four hidden nodes, and seven input variables along a bias node connecting both hidden and output layers.

3.4 Determining response elasticities in ANN models

The connection weights of our ANN model are shown in Table III and it is these weights that are used to calculate network output (y). Recall there are seven model variables $X_{1t}, X_{2t}, \ldots, X_{7t}$, where $t = 1 - T$, the total number of observations. There are four hidden nodes, h_{1t}, h_{2t}, h_{3t}, h_{4t}, a bias node, γ_5 and, there is one output node y_t.

Therefore, our ANN equation can be written as:

$$y_t = \sum_{i=1}^{4} \gamma_i h_{it} + \gamma_5, \tag{4}$$

where,

$$h_{it} = \frac{1}{1 + e^{-\sum_{k=1}^{7} W_{ik} X_{kt} + W_{ik+1}}} \tag{5}$$

Weight matrix from input to hidden layer				
From input	*H-1*	*H-2*	*H-3*	*H-4*
Bias node	0.0148	−0.0544	−0.0363	0.0752
RP	−0.0616	0.0892	0.0387	0.0192
PROM	−0.0291	0.0434	0.0578	−0.0102
PCUT	−0.0208	−0.0328	0.0722	0.0359
PRV	0.0684	0.0195	−0.0881	0.0763
SPRV	−0.0090	0.0543	0.0223	0.0482
BL	−0.0932	0.0935	0.0492	0.1036
SL	−0.0588	0.0860	0.0199	0.0863

Weight matrix from hidden to output layer	
From hidden layer	*Output (Y)*
Bias node	19.75448
Hidden node 1 (H-1)	18.96961
Hidden node 2 (H-2)	10.43411
Hidden node 3 (H-3)	13.69286
Hidden node 4 (H-4)	10.43411

Table III.
Weights for GANNT trained ANN

which gives the output of each ith hidden node at observation t resulting from the sigmoidal transformation of the summed hidden node inputs, which are seven input variables (X_{kt}) each multiplied by a connection weight (W_{ik}), with W_{ik+1} being the input of the biased node to that hidden node.

Additionally, the derivative of the ANN equation with respect to one of the variables X_m can be written as:

$$\frac{\delta y}{\delta x} = \sum_{i=1}^{4} \gamma_i \frac{\delta h_i}{\delta X_m}, \tag{6}$$

where,

$$\frac{\delta h_i}{\delta X_m} = \frac{W_{im} e^{-\sum_{i=1}^{7} W_{ij} X_j + W_i^8}}{\left(1 + e^{-\sum_{i=1}^{7} W_{ij} X_j + W_i^8}\right)^2} \tag{7}$$

and by taking the derivative of the network output, after training the full model and then varying each predictor variable across its range in the data file in equal increments while holding the other predictor variables fixed, we can compare the slopes of the functional relationships of the variables and use them as indicators of response elasticities.

We could also use a similar method to concerning model output based on equation (4) to obtain an estimate of response elasticity of variables, if all the variables' units and data ranges were equal. However, this is not the case with

our data (GANNT does not normalize the data before submitting it to the network as back-propagation does), some variables use nominal data ranging from 0 to 1, while others (e.g. regular price (RP)) use ratio data ranging from 15 to 21. Even though direct comparison among variables is precluded, such analysis can give us an indication of how the model behaves with respect to a certain variable. We develop response graphs for each model variable and show them in Figure 8.

In examining the loyalty variables, brand loyalty (BL) reflects an almost linear relationship – increases in BL have a strong positive effect on the consumer's propensity to purchase the brand. Brand loyal customers (greater than 0.75 on the BL input) cause network output to be above 0.6. In examining the derivative graph, the slope of the response function steadily increases slightly, while ranging from 0.475 at zero brand loyalty to 0.482 at perfect BL. This supports the notion that the relationship is not perfectly linear and that the rate in change increases as the customer becomes more loyal. SL shows a similar relationship, though not as profound. Network output ranges from 0.254 with no SL to 0.533 for perfect SL. The dy/dx function shows this relationship to be almost linear, ranging only from 0.278 to 0.279.

The response graph for PCUT shows a non-linear relationship between the amounts of the price cut and network output as the graph begins to flatten at the points of large price cuts. The PCUT chart contains negative inputs, which means that the promotional price is actually higher than the regular price. This can occur because of the earlier mentioned defect in the database requiring that a promotion be assumed when two of the three conditions are met. If there is an unusually high sales movement and the brand-size is featured in a print promotion, then we consider it as a promotional purchase even though the price may be higher than the RP thus, a negative price cut. The downward sloping dy/dx graph also shows non-linearity, ranging from 0.088 to 0.005.

RP results indicate that increases in the non-promoted price of a brand has a slightly negative effect on network output. Network output only varies from 0.426 to 0.35 across the range of inputs. The slope of the curve ranges from -0.004 to -0.017, which is reflected in a negative and downward sloping curve of the derivative graph.

We create lines for the dichotomous predictor variables by inputting zero then one into the network equation and its derivative. The promotion (PROM) variable shows the steepest slope, it is also positive. When there are no sales promotions the network output is at 0.373, and this increases to 0.506 when a sales promotion takes place. This supports the well-known fact that sales promotions have a positive effect on sales. This effect is substantial and depicted in the derivative output 0.13.

Both the previous purchase (PRV) and second previous purchase (SPRV) have downwardly sloping response functions and negative derivatives. If a

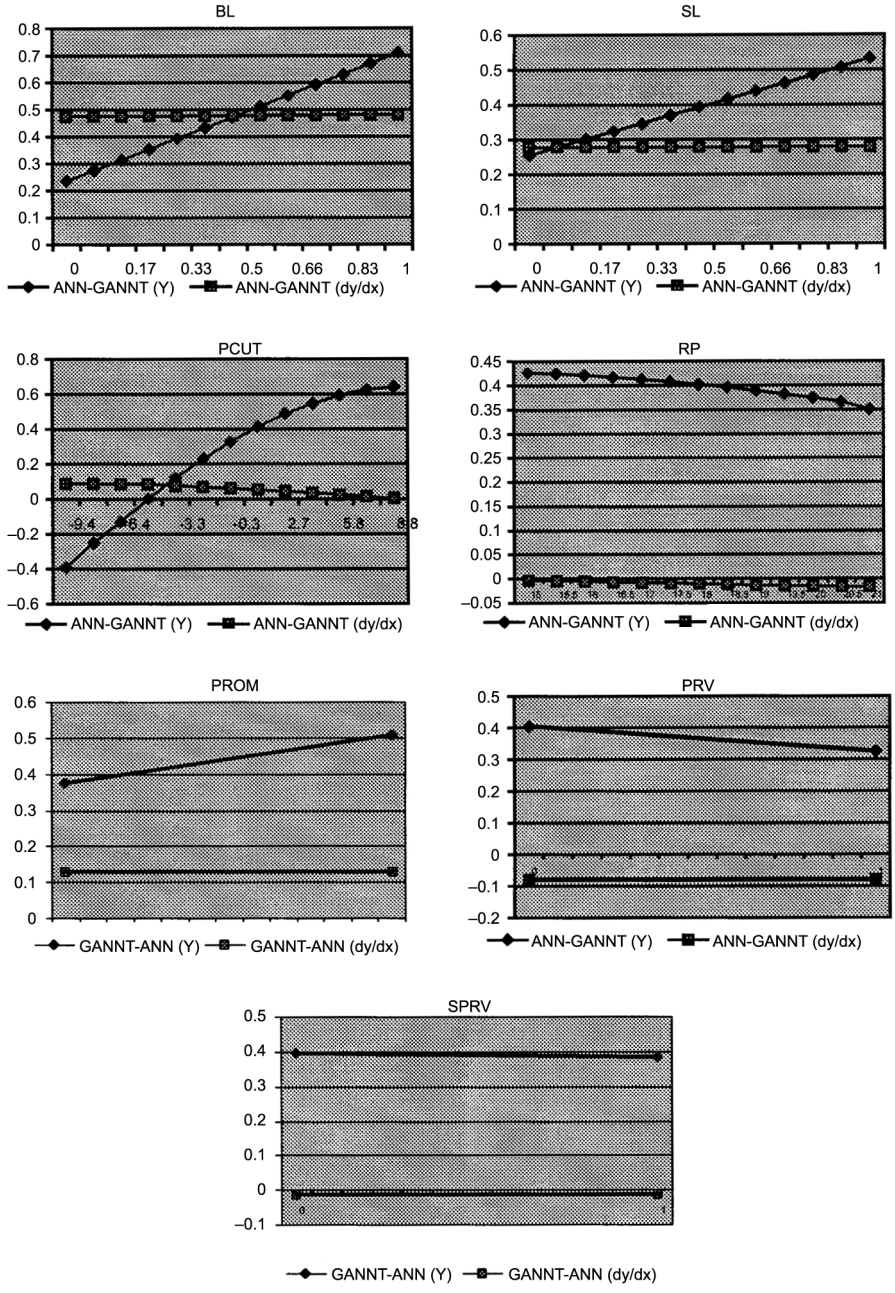


Figure 8.
Response elasticities of the model variables

customer's PRV of a brand-size was on promotion, then network output is reduced to 0.324 from 0.404. The effects of the SPRV variable are less profound with network output dropping only from 0.395 to 0.382 if the SPRV was made on promotion. An individual's previous promotional purchase history has more influence (derivative output of −0.08) than that person's second previous promotional purchase history (derivative output of −0.013). These results indicate that previous promotional purchases decrease the likelihood of a subsequent purchase of the brand, a notion well supported in the literature (Dodson *et al.*, 1978; Shoemaker and Shoaf, 1977).

In the GANNT trained ANN model we can compare the derivatives of the output to determine variable importance. Our derivative graphs show BL, SL, PROM to be the three most important variables while the importance of PCUT varies. The efficacy of PCUT depends on the extent of reduction from the non-promoted price. The derivative approaches zero in the area of deep price cuts. Marketing managers should be aware that the effectiveness of promotional price cuts lessens as the cuts become deeper. In other words, the marketing manager gets more bang for the lost buck with cuts closer to the non-promoted price. Deeper price cuts will have little effect in spurring additional sales. The SPRV variable is shown to be of little explanatory importance based on its derivative being close to zero.

4. Discussion of results and conclusions

4.1 Visual analysis of learning rules

Data mining as applied to the databases generated by the learning rules of neural networks lead to accurate estimates of the synaptic strengths and stability coefficients for learning rules. The analysis of the neural network data using data mining also led to graphical representations of the data as shown in Figures 3 and 4. The latter provided a visual methodology of discerning the similarities and differences between the individual learning rules.

4.2 Visual analysis of variable importance

Since our variable interpretation results are similar to Guadagni and Little's, their original discussion of marketing ramifications of the variables still suffices; however, the ANN approach can provide the researchers with additional information The varying degree of efficacy of PCUTs is not apparent with a statistical model because the *t*-statistic is a constant elasticity measure. Additionally, the RP graph shows researchers that as RP points increase across the market range, they have an increasingly negative effect on sales. Although the negative relationship is captured in a statistical model, the *changes* in the rate of effectiveness are not captured with statistical constants.

Although we use a feed-forward network trained by a genetic algorithm, this method of variable interpretation is amenable to back-propagation trained

networks as well. Data miners should consider such an approach after they develop predictive models from a data set. Not only does it yield a concise idea of variable importance but also provide insightful information into understanding the behavior of model variables over a range of different input scenarios.

References

Abbott, L.F. and Kepler, T.B. (1989), "Optimal learning rules in neural network memories", Preprint BRX-TH-225, Brandeis University, Waltham, MA.

Agrawal, D. and Schorling, C. (1996), "Market share forecasting: an empirical comparison of artificial neural networks and multinomial logit model", *Journal of Retailing*, Vol. 72, pp. 383-407.

Bigus, J.P. (1996), *Data Mining with Neural Networks*, Mc-Graw Hill, New York, NY.

Diedrich, S. and Opper, M. (1987), "Learning of correlated patterns in spin-glass networks by local learning rules", *Physics Review Letters*, Vol. 58, pp. 949-52.

Dodson, J.A., Tybout, A.M. and Sternthal, B. (1978), "Impact of deals and deal retraction on brand switching", *Journal of Marketing Research*, Vol. 15, pp. 72-81.

Dorsey, R.E., Johnson, J.D. and Mayer, W.J. (1994), "A genetic algorithm for the training of feedforward neural networks", in Johnson, J.D. and Whinston, A.B. (Eds), *Advances in Artificial Intelligence in Economics, Finance and Management*, Vol. 1, JAI Press, Greenwich, CT, pp. 93-111.

Gardner, E. (1988), "The space of interactions in neural network models", *Journal of Physics Series A*, Vol. 21, pp. 257-70.

Grover, R. and Srinivasan, V. (1992), "Evaluating the multiple effects of retail promotions on brand loyal and brand switching segments", *Journal of Marketing Research*, Vol. 29, pp. 76-89.

Guadagni, P.M. and Little, J.D.C. (1983), "A logit model of brand choice calibrated on scanner data", *Marketing Science*, Vol. 2, pp. 203-38.

Hebb, D. (1949), *Organization of Behavior*, Wiley, New York, NY.

Kalwani, M.U., Yim, C.K., Rinne, H.J. and Sugita, Y. (1990), "A price expectations model of customer brand choice", *Journal of Marketing Research*, Vol. 27, pp. 251-62.

Krauth, W. and Mezand, M. (1987), "Learning algorithms with optimal stability in neural networks", *Journal of Physics Series* A, Vol. 20, pp. L745-52.

Segall, R.S. (1995), "Some mathematical and computer modeling of neural networks", *Applied Mathematical Modelling*, Vol. 19, pp. 386-99.

Segall, R.S. (1996), "Comparing learning rules of neural networks using computer graphics", *Proceedings of the Twenty-seventh Annual Conference of the Southwest Decision Sciences Institute*, 6-9 March 1996, San Antonio, TX.

Shoemaker, R.W. and Shoaf, R.F. (1977), "Repeat rates of deal purchases", *Journal of Advertising Research*, Vol. 17, pp. 47-53.

Simonoudis, E., Livezey, B. and Kerber, R. (1995), "Using RECOn for data cleaning", *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, AAAI Press, pp. 282-7.

Whitley, D., Starkweather, T. and Bogart, C. (1990), "Genetic algorithms and neural networks: optimizing connections and connectivity", *Parallel Computing*, Vol. 14, pp. 347-61.