Management Decision

Using modular neural networks for business decisions

David Mitchell, Robert Pavur,

Using modular neural networks for business decisions

David Mitchell
University of Houston Downtown, Houston, Texas, USA
Robert Pavur
University of North Texas, Denton, Texas, USA

Keywords
Neural networks, Classification, Decision-support systems

Abstract
Understanding large amounts of information and efficiently using that information in improved decision making has become increasingly challenging as businesses collect terabytes of data. Businesses have turned to emerging technology including neural networks, symbolic learning, and genetic algorithms. In the current study, four classification methods were compared using results from an Indonesian contraceptive-method preference survey. The four methods are linear discriminant analysis, quadratic discriminant analysis, backpropagation neural networks, and modular neural networks. The modular neural network is a more complex and less frequently used neural network model. This comparative study gives insight into its performance on classifying observations from a challenging data set, the 1987 National Indonesia Contraceptive Prevalence Survey.

Introduction

Modern business and government decision makers are awash in data. The well being of their companies and cities/states/nations rest to a large extent on their ability to sift through enormous quantities of data, extract useful knowledge, and apply this knowledge in a relevant manner to decisions. This relatively recent state of affairs is a result of the huge increase in the speed of data processing and the extensive improvements made in the information technology infrastructure. These changes have made data collection a ubiquitous activity. Hundreds of millions of transactions occurring at point-of-sale locations and over the Internet are logged into data warehouses continuously. As a result of this glut of data and the necessity of extracting useful knowledge from it, researchers and practitioners alike have begun to focus attention on approaches designed to assist in the decision-making process. Among these approaches are included decision support systems (DSSs) and data mining.

Data mining uses various models, such as, discriminant analysis, cluster analysis, neural networks, decision trees, and genetic algorithms, to reveal knowledge hidden in large databases. These models are capable of recognizing patterns in data and organizing the data into groups based on these patterns. Which of these models will perform the classification task best for a given problem domain is uncertain (Cios *et al.*, 1998). Therefore, testing a variety of classification models for a given problem domain is important for establishing each potential model's relative performance.

The classification of data into homogeneous groups is often an integral part of the decision-making process. For example, Gupta *et al.* (1997) used variables related to plant location, labor and organizational issues, plant characteristics, project risks, and environmental concerns as inputs to a classification task where the categories were levels of modularization appropriate in the construction of an industrial facility. In many situations, classification models such as those mentioned above are included in DSSs. In the current study, four classification methods were compared using results from an Indonesian contraceptive-method preference survey. The four methods are linear discriminant analysis, quadratic discriminant analysis, backpropogation neural networks, and modular neural networks.

Backpropagation neural networks

The most widely and nearly exclusively used neural network for business applications is the backpropagation neural network (BPNN). The BPNN is a feedforward neural network that uses the backpropagation of error during learning. BPNNs most often use a generalized-delta learning rule. This learning rule, generally attributed to Rumelhart *et al.* (1986), is a nontrivial extension of the learning rule for a simple Adaline network (Widrow and Lehr, 1990). The generalized-delta learning rule effects a gradient descent search in weight space used to minimize classification error. This weight adjustment process may be extremely slow and as in many nonlinear search methods is not guaranteed to converge to an optimal solution. Despite these potential shortcomings, BPNNs have demonstrated their efficacy on many practical problems and have been shown to be relatively easy to use. While it is likely that many users are not aware of how the BPNN model accomplishes its results, BPNNs do self-adapt to learn from information. They thereby provide powerful models that may be used in many

Management Decision
40/1 [2002] 58–63

[ISSN 0025-1747]
[DOI 10.1108/00251740210413361]

circumstances to transform vague data into knowledge useful for making decisions.

According to Jain and Nag (1997), a BPNN with one hidden layer of neurons can "closely" approximate any decision surface. The degree of closeness depends on the relationship between the architecture of the network and the decision surface being approximated. One important architectural decision involves the determination of the number of hidden-layer neurons. While increasing the number of hidden-layer neurons can lead to improved performance on the observations used to train a BPNN, it may also lead to a lack of generalizability of the network (Jain and Nag, 1997). Because there are no theoretical guidelines for determining the appropriate number of hidden-layer neurons, a trial-and-error process has generally been used by researchers and practitioners to determine the best architecture. There are, however, a number of heuristics available for specifying the number of hidden-layer neurons for a given application. One such heuristic suggested by NeuralWare, Inc. is as follows:

Number of hidden neurons = number of cases/ $(5 * (m + n))$,

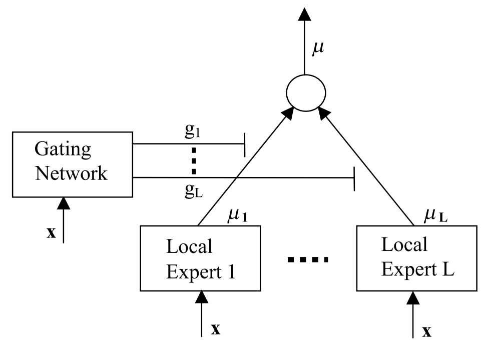
where m and n are the number of input and output neurons, respectively.

Modular neural networks

Weaknesses in BPNNs have suggested alternative procedures for classification tasks such as genetic algorithm models and radial basis function neural networks. Wang (1995) suggests caution in using the BPNN model. Modular neural network (MNN) refers to a mixture of local experts and was proposed by Jacobs *et al.* (1991). MNNs are composed of a number of independent neural network models called local experts that are organized to focus their efforts on a subset of the decision space. These local experts are most often BPNNs, but can be any type of supervised neural network including other MNNs. A distinct advantage of the MNN architecture is the inclusion of a gating network to control the competition among the several local expert networks.

Relatively little research attention has been given to MNN models. This is likely due to several factors including the fact that MNNs are more complicated to implement and are more computationally intensive than BPNNs because several networks are being trained simultaneously. Theoretically, any problem that can be solved by a BPNN can be solved at least as well by a MNN. Figure 1 displays the general MNN. Consider the *g*'s in Figure 1 to be a probability vector used to control competition among the local experts. Each element of the gating network output vector can be thought of as a prior probability of group classification for the local expert it is associated with. The gating network is trained at the same time as the local experts. Its output vector is adjusted to more closely match the prior probabilities with the posterior probabilities estimated by the local experts. The μ_i values are the output vectors of the local experts and $\boldsymbol{x}$ is the input vector to each of the local experts and the gating network. The output vector μ is a weighted composite of the outputs from all of the local experts.

Figure 1
General modular neural network architecture for current study



Practitioners have few guidelines to help them implement MNNs. Many decisions must be made for each implementation including the number of local experts, the number of neurons in the hidden layer of the local experts, and the number of neurons in the gating network. The process of determining the best architecture is usually one of trial and error, similar with the approach most often used to specify BPNNs. To gain some insight into performance characteristics of MNN models, a real-world data set was selected from among the 16 used by Lim *et al.* (1999). The 16 data sets used by these authors are quite diverse and consist of the following data sets: breast cancer data, contraceptive method choice data, DNA data, heart disease data, Boston housing data, light-emitting diodes data, liver disorder data, Indian diabetes data, satellite image data, attitude towards smoking data, thyroid data, vehicle silhouette data, Congressional voting data, wavelet data, and teaching performance data. The observations in the data set selected, the 1987 National Indonesia Contraceptive Prevalence Survey (CPS), were among the most difficult to classify.

The sample in this contraceptive prevalence survey consisted of married women who were either not pregnant or did not know if they were at the time of interview. The problem is to predict the current contraceptive approach chosen (no method or else using an effective contraceptive approach) by a woman based on her demographic and socio-economic characteristics. The data set consisted of 1,473 observations with the attribute information shown in Table I.

Simulation study using contraceptive prevalence survey data

To understand how BPNNs and MNNs would perform against the CPS data set, various architectures were tested. The BPNN models were implemented with both ten and 30 hidden-layer neurons. The rule suggested by NeuralWare, Inc. led to the ten hidden-layer neuron architecture. The value of 30 was arrived at through experimentation. The number of local experts, four, six, or eight, to include in each MNN was also established through experimentation. The number of hidden-layer neurons in each local expert was set, for a given implementation, to either ten or 30 to match the values used in the BPNNs. Since there are no guidelines for selecting the number of gating-network hidden-layer neurons, the values of five, ten and 15 were used in turn when ten neurons were used in the local experts. When the local experts had 30 hidden-layer neurons, 25, 30, and 35 gating-network neurons were used. Other pairings of hidden-layer and gating-network neurons were tried but were found to perform much more poorly than those selected for the study.

A total of 20 groups of three data sets each, training, testing, and validation, were created from the original data set of 1,473 records. A total of 60 percent of the data was randomly selected for development of the neural network models. These data were split two-thirds/one-third into training and testing data sets for training the models, i.e. adjusting the model weights, and selecting the best performing network respectively. Once the best performing model had been selected, the remaining 40 percent of the records in the validation data set was used for model performance comparisons. In addition to the BPNN and MNN models, two classical parametric procedures were tested: Fisher and Quadratic discriminant analysis (Rencher, 1995). Parameters for these models were determined using the records from the combined training and testing data sets. Once specified, the parametric models were tested against the validation data sets.

Table I
Attribute information

1. Wife's age	(numerical)	
2. Wife's education	(categorical)	1 = low, 2, 3, 4 = high
3. Husband's education	(categorical)	1 = low, 2, 3, 4 = high
4. Number of children ever born	(numerical)	
5. Wife's religion	(binary)	0 = Non-Islam, 1 = Islam
6. Wife's now working?	(binary)	0 = Yes, 1 = No
7. Husband's occupation	(categorical)	1, 2, 3, 4
8. Standard-of-living index	(categorical)	1 = low, 2, 3, 4 = high
9. Media exposure	(binary)	0 = Good, 1 = Not good
10. Contraceptive approach	(class attribute)	1 = Not using any method, 2 = Using a contraceptive method

Results

Two analyses were performed to examine the performance characteristics of the various classification models. The first analysis consisted of an ANOVA and Tukey *post hoc* tests of the proportion of correctly classified records by each model. The results are detailed in Table II. The notation used to represent the modular neural networks is as follows: first letter is M for modular neural network, the next digit indicates the number of local experts, the following two digits give the number of hidden-layer neurons in the local experts, and the final one or two digits indicate the number of hidden-layer neurons in the gating network. The letters BP followed by the number of hidden-layer neurons denotes the backpropagation methods. The models labeled Fisher and Quad are, respectively, the parametric discriminant analysis procedure by Fisher, which is optimal for normal data with equal population covariance matrices, and the quadratic parametric procedure, which is optimal for normal data with unequal population covariance matrices.

Tukey's multiple comparision procedure uses the letters A through F in Table II to denote models whose performances do not differ significantly at the 5 percent significance level. For example, the first model listed, M63035, is not significantly different from any of the models with an A under the heading of "Groupings from Tukey procedure." The multiple comparison procedure illustrates which means are close enough to be considered not significantly different.

The results indicate that while most of the MNNs do not differ from each other statistically, most do perform statistically better than both the parametric and BPNN models. Also of interest is the fact that worst performance was obtained from the BPNN models including one model, BP30 that was statistically worse than either of the parametric models.

An additional analysis included data from only the MNN models. An analysis of covariance was performed with the factors number of local experts and number of hidden-layer neurons in the local experts. The number of hidden-layer neurons in the gating network was a covariate. Of interest in this analysis was the interaction of the number of local experts with the number of hidden-layer neurons in the local experts as well as the interaction of the number of local experts with the number of hidden-layer neurons in the gating network. The dependent variable for this analysis was the network's classification rate. The analysis, using sequential sums of squares, yielded the results presented in Table III.

Since Table II reveals no clear cut pattern as to how the number of local experts, number of hidden-layer neurons in the local BPNNs, and the number of hidden-layer neurons in the gating network are related, it is not surprising that Table III shows significant interactions at the 10 percent level for the number of local experts and number of gating neurons and at the 1 percent level for the number of experts and number of hidden-layer neurons in the local BPNNs. To gain further insight into the relationship of this interaction, graphs of both interactions are illustrated in Figures 2 and 3.

A couple of observations can be made from viewing Figures 2 and 3. Figure 2 illustrates that for four experts, the classification rate adjusted for the effect of the gating neurons trends lower for the larger number (30) of hidden-layer neurons. Figure 3 illustrates that for four experts the performance trends lower and then levels off as the number of gating network hidden-layer neurons increases. Now if ten hidden-layer neurons are used in the local experts, then the performance of the configurations with 4, 6, or 8 experts appears to be rather close as displayed in Figure 2. From Table II, the best absolute performing MNNs were two of the most complex networks tested. While this might indicate a need for increased network complexity when attempting to classify certain hard-to-separate groups, the absolute performance of the simplest network (third overall: four local experts, ten hidden-layer neurons in the BPNNs, and five hidden-layer neurons in the gating network) would tend to obviate this conclusion. Rules-of-thumb for selecting the parameters for an MNN may prove to be illusive.

Table II
Multiple comparison results for modular neural networks, backpropagation neural networks, and classical parametric procedures

Groupings from Tukey procedure				Mean	N	Model[a]
		A		0.540410	20	M63035
B		A		0.536500	20	M83035
B		A		0.535990	20	M4105
B		A		0.535480	20	M8105
B		A		0.535320	20	M83025
B		A		0.534975	20	M63025
B		A		0.534970	20	M63030
B		A		0.534035	20	M81015
B		A		0.532850	20	M61010
B		A	C	0.529540	20	M83030
B	D	A	C	0.528260	20	M41010
B	D	A	C	0.527320	20	M61015
B	D	A	C	0.526735	20	M81010
B	D	A	C	0.526650	20	M6105
B	D		C	0.520870	20	M43035
B	D		C	0.520195	20	M43025
B	D		C	0.519685	20	M41015
B	D		C	0.519085	20	M43030
	D	E	C	0.512904	20	Fisher
	D	E		0.509168	20	Quad
F		E		0.496265	20	BP10
F				0.488285	20	BP30

Notes: [a] The neural network model descriptions are as follows: M is for modular neural network; BP is for backpropagation; 4, 6 and 8 indicate the number of local experts; 10 and 30 indicate the number of hidden-layer neurons in the BPNNs and local experts, 5, 10, 15, 25, 30 and 35 indicate the number of hidden-layer neurons in the gating network

Conclusions

There appear to be two reasonable conclusions that can be drawn from the results of this study. First, if classification by standard models shows poorly separated groups, MNN models should be seriously considered. In this study, every MNN model statistically outperformed the more traditionally used BPNNs. This implies that breaking the complex classification space into smaller, more manageable spaces helps improve classification performance.

The second conclusion is in regard to the search for identifying rules for the optimal parameters of neural network models. Researchers have identified some basic rules for BPNN models. These rules are only meant to be guidelines. If guidelines are to be created for MNN models, then interactions between the number of local experts and both the number of local expert hidden-layer neurons and number of gating network hidden-layer neurons should be considered. Perhaps the additional complexity of the

Table III
Analysis of variance results for the modular neural network models

Source	DF	Sum of squares	Mean square	*F* value	PR > *F*
Model	27	0.06277065	0.00232484	8.63	< 0.0001
Error	332	0.08941020	0.00026931		
Corrected total	359	0.15218084			
Datasetnum	19	0.05122816	0.00269622	10.52	< 0.0001
Experts	2	0.00631518	0.00315759	11.72	< 0.0001
HiddenBP	1	0.00002624	0.00002624	0.10	0.7492
GatingNeurons	1	0.00015974	0.00015974	0.59	0.4418
Experts*HiddenBP	2	0.00379287	0.00189644	7.04	0.0010
Experts*GatingNeuron	2	0.00124845	0.00062423	2.32	0.1001

Figure 2
Interactions between number of local experts and number of hidden-layer neurons

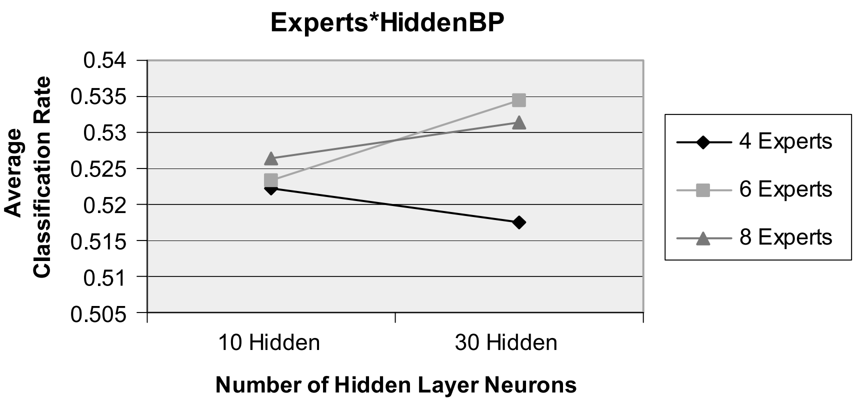
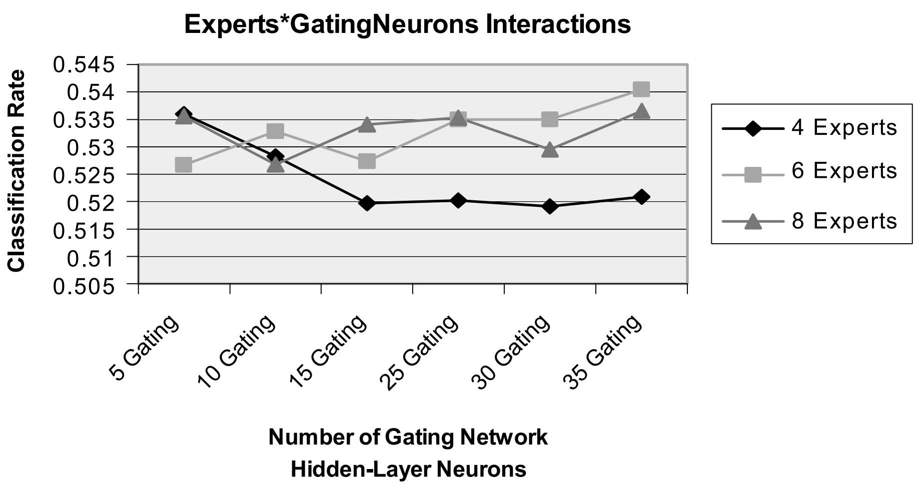


Figure 3
Interactions between number of local experts and number of gating network hidden-layer neurons



MNN model makes it too difficult to readily identify rules for setting parameters. While MNN models outperformed the BPNN models, identifying the best performing MNN model parameters required trial and error. As observed from reports of the performance of BPNN models, the optimal parameters for an MNN model will probably vary considerably across different problem domains and even across data sets within a particular problem domain. More studies across data sets from the same problem

Management Decision
40/1 [2002] 58–63

domain and data sets from a variety of problem domains must be performed before further conclusions concerning parameter settings can be drawn.

References

Cios, K., Pedrycz, W. and Swiniarski, R. (1998), *Data Mining Methods for Knowledge Discovery*, Kluwer Academic Publishers, Boston, MA.

Gupta, V.K., Chen, J.G. and Murtaza, M.B. (1997), "A learning vector quantization neural network model for the classification of industrial construction projects", *Omega*, Vol. 25 No. 6, pp. 715-27.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E. (1991), "Adaptive mixtures of local experts", *Neural Computation*, Vol. 3, pp. 79-87.

Jain, B.A. and Nag, B.N. (1997), "A performance evaluation of neural network decision models", *Journal of Management Information Systems*, Vol. 14, pp. 201-16.

Lim, T.-S., Loh, W.-Y. and Shih, Y.-S. (1999), "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms", *Machine Learning*, forthcoming.

Rencher, A.C. (1995), *Methods of Multivariate Analysis*, Wiley, New York, NY.

Rumelhart, D.E., Hinton, G.E. and Williams, R.J. (1986), "Learning representations by back-propagating errors", *Nature*, Vol. 323, pp. 533-6.

Wang, S. (1995), "The unpredictability of standard back propagation neural networks in classification applications", *Management Science*, Vol. 41, pp. 555-9.

Widrow, B. and Lehr, M.A. (1990), "30 years of adaptive neural networks: perceptron, madaline, and backpropagation", *Proceedings of the IEEE*, Vol. 78, pp. 1415-42.

Application questions

1 Could a modular neural network help your company's business decision-making processes? How?

2 What do you consider to be the advantages and disadvantages of the modular neural network over the other types of neural networks discussed in this paper?

This article has been cited by:

1. Noraina Mazuin Sapuan, Suzaida Bakar, Hamidah Ramlan. 2017. Predicting the Performance and Survival of Islamic Banks in Malaysia to Achieve Growth Sustainability. *SHS Web of Conferences* **36**, 00016. [Crossref]