Neuro-genetic Approach for Bankruptcy Prediction: A Comparison to Back-propagation Algorithms

Kyung-shik Shin, Taek-soo Shin and Ingoo Han
Graduate School of Management
Korea Advanced Institute of Science and Technology,

ABSTRACT

A number of recent studies have demonstrated that artificial intelligence such as back-propagation neural networks (BPN) can be an alternative methodology for classification problems to which traditional statistical methods have long been applied. However, the well-known limitations of gradient search techniques such as BPN applied to complex nonlinear optimization problems have often resulted in inconsistent and unpredictable performance.

We investigate the classification performance of neuro-genetic approaches for the bankruptcy prediction tasks. Although genetic algorithms (GAs) and BPN have in common that they are general search techniques, empirical studies show GAs perform a more global search than BPN. The preliminary results demonstrate that genetic training can be an alternative training algorithm for neural networks learning, although the model does not outperform the back-propagation learning algorithm

I. INTRODUCTION

A number of ways of determining the potential for corporate failure using past financial data is a well-documented topic. The solution to this problem is a discriminant function from the variable space in which observations are defined into a binary set. In particular, a number of recent studies have demonstrated that artificial intelligence such as back-propagation neural networks (BPN) can be an alternative methodology for classification problems to which traditional statistical methods have long been applied (Barniv *et al.*, 1997; Bell, 1997; Boritz and Kennedy, 1995; Chung and Tam, 1992; Etheridge and Sriram, 1997; Chen *et al.*, 1995; Fletcher and Goss, 1993; Jo *et al.*, 1997; Markham and Ragsdale, 1995; Odom and Sharda, 1990; Salchenberger *et al.*, 1992; Shin and Han, 1998; Shin *et al.*, 1998; Tam and Kiang, 1992; Wilson and Sharda, 1994).

However, the well-known limitations of gradient search techniques such as BPN applied to complex nonlinear optimization problems have often resulted in inconsistent and unpredictable performance. Although prior studies have attempted to address the problem by adjusting the characteristic of back-propagation (Coats and Fant, 1993; Fahlman and Lebiere, 1990; Lacher *et al.*, 1995; Wang, 1995), the results were rather disappointing.

This paper investigates the effectiveness of neuro-genetic approach which combines neural networks (NNs) and genetic algorithms (GAs) to build a bankruptcy prediction model. In this approach, GAs are used to search the weight space of a neural network without the use of any gradient information (Montana and Davis, 1989). Although GAs and BPN have in common that they are general search techniques, empirical studies show GAs perform a more global search than BPN (Kitano, 1990; White, 1993).

The remainder of this paper is organized as follows. The next section provides the basic concepts of neuro-genetic approach we are applying including the general description of GA methodology. The third section describes model building process. The fourth section presents comparison results with BPN and the final section discusses further research issues.

II. NEURO-GENETIC METHODOLOGY

2.1 Genetic Algorithms

GAs are stochastic search techniques that can search large and complicated spaces on the ideas from natural genetics and evolutionary principle (Davis, 1991; Holland, 1975; Goldberg, 1989). GAs perform search process in four stage: initialization, selection, crossover, and mutation (Davis, 1991; Wong and Tan, 1994). They have been demonstrated to be effective and robust in searching very large spaces in a wide range of application (Colin, 1994; Fogel, 1993; Klimasauskas, 1992; Koza, 1993; Shin and Han, 1998). In the initialization stage, a population of genetic structures (called chromosomes), that are randomly distributed in the solution space is selected as the starting point of the search. After the initialization stage, each chromosome is evaluated using a user-defined fitness function. The goal of the fitness function is to numerically encode the performance of the chromosome. For real-world applications of optimization methods such as GAs, the choice of the fitness function is the most critical step.

The mating convention for reproduction is such that only high scoring members will preserve and propagate their worthy characteristics from generations to generation and thereby help in continuing the search for an optimal solution. Chromosomes with high performance may be chosen for replication several times, whereas poor-performing structures may not be chosen at all. Such a selective process causes the best-performing chromosomes in the population to occupy an increasingly larger proportion of the population over time.

Crossover causes to form new offspring between two randomly selected 'good parents'. Crossover operates by swapping corresponding segments of a string representation of the parents and extends the search for new solution in far-reaching direction. The crossover occurs only with some probability (the crossover rate). There are many different types of crossover that can be performed: the one-point, the two-point, and the uniform type (Syswerda,1989).

Mutation is a GA mechanism where we randomly choose a member of the population and change one randomly chosen bit in its bit string representation. Although reproduction and crossover produce many new strings, they do not introduce any new information into the population at the bit level. If the mutant member is feasible, it replaces the member which was mutated in the population. The presence of mutation ensures that the probability of reaching any point in the search space is never zero.

GAs have been increasingly applied in conjunction with other AI techniques such as neural networks, rule-based system, fuzzy theory, and CBR. The integration of GAs and neural networks is a rapidly expanding area. The common problems faced by researchers and developers in using neural network techniques are optimization of input selection, network design and learning condition. Various problems of neural network design can be optimized using GAs (Wong and Tan, 1994). Examples include selecting relevant input variables, determining the optimal number of hidden layers, nodes and connectivity, and tuning the learning parameters (Bishop, *et al.*, 1993; Harp and Samad, 1991; Schaffer, *et al.*, 1992).

Another method of combining neural networks and GAs is called neuro-genetic approach (Harp, *et al.*, 1989; Heistermann, 1989; Miller, *et al.*, 1989; Mühlenbein, *et al.*, 1989; Montana and Davis, 1989; Whitley, 1988, 1989; Whitley and Hanson, 1989) which will be discussed in detail in the following section. In neuro-genetic model, GAs are used to search the weight space of a neural network without the use of any gradient information.

GAs are also used in conjunction with fuzzy logic systems to provide an appropriate set of fuzzy IF-THEN rules for classification problems (Ishibuchi, *et al.*, 1993) and to improve fuzzy logic controller (Karr, 1991; Park, *et al.*, 1994).

Few studies have dealt with hybridization of genetic and case-based reasoning, though there exists a great potential for useful applications in this area. Wang and Ishii (1997) applied GAs to the method of similarity metrics based on the cases being represented by structured representations. The recent study of Shin and Han (1998) proposed a new hybrid approach using genetic algorithms to case-based retrieval process in an attempt to increase the overall classification accuracy. They utilized a machine learning approach using genetic algorithms to find an optimal or near optimal importance weight vector for the attributes of cases in case indexing and retrieving.

2.2 Neuro-genetic approach

In neural networks training, the most commonly used algorithms are versions of the back-propagation algorithms developed by Rummelhart *et al.* (1986). The well-known limitations of gradient search techniques applied to complex nonlinear optimization problems have often resulted in inconsistent and unpredictable performance. They typically start at a randomly chosen point (set of weights) and then adjust the weights to move in the direction which will cause the errors to decrease most rapidly. These types of algorithms work well when there is a smooth transition toward the point of minimum error. Unfortunately, however, the error surface of the neural network is not smooth. It is characterized by hills and valleys that cause techniques such as BPN to become trapped in local minimum.

Many researchers have attempted to address the problems associated with the training algorithm by imposing constraints on the search space or by restructuring the architecture of the neural network (Coats and Fant, 1993; Fahlman and Lebiere, 1990; Lacher *et al.*, 1995; Wang, 1995). The recent study of Sexton, *et al.*(1998) demonstrates that such constraints and restructuring are unnecessary if a sufficiently complex initial architecture and an appropriate global search algorithm is used and show that the genetic algorithm can not only serve as a global search algorithm but by appropriately defining the objective function it can simultaneously achieve a parsimonious architecture.

The idea of combining GAs and NNs came up first in the late 1980s (Harp *et al.*, 1989, Heistermann, 1989; Miller, *et al.*, 1989; Mühlenbein, *et al.*, 1989; Montana and Davis, 1989; Whitley, 1988, 1989; Whitley and Hanson, 1989) and it has generated an intense field of research in the 1990s (Whitley, *et al.*, 1990; Dodd, 1990; Kitano, 1990; Heistermann, 1990; Schiffmann and Mecklenburg, 1990; Schiffmann, *et al.*, 1991, 1992, 1993; Weiss, 1990). Much of the research has focused on the training of feedforward networks (Fogel, *et al.*, 1990; Whitley, *et al.*, 1990), applying evolutionary algorithms to recurrent neural networks (Angeline, *et al.*, 1994; Beer and Gallagher, 1992; Bornholdt and Graudenz, 1992), generalized regression neural network(Hansen and Meservy, 1996), and Hopfield neural networks (Shirai, *et al.*, 1994; Lin, *et al.*, 1995).

In neuro-genetic approach, the learning of a neural network is formulated as a weights optimization problem, usually using the inverse-mean-square error as a fitness measure. The basic concept behind this technique is as follows. A complete set of weights is coded in a string, which has an associated "fitness" representing its effectiveness. Starting with a random population of such strings, successive generations are constructed using genetic operators to construct new strings out of old ones such that better strings are more likely to survive and to participate in crossover operations. Unlike the back-propagation learning rule, GAs perform a global search and are thus not easily fooled by local minimum. The utilization of the linkage among population searches makes the GA a good global search method.

III. MODEL DEVELOPMENT

The architecture of our neuro-genetic models is represented by a connectivity constraint matrix of dimension {(M+1)N + (N+1)}, with the first column denoting the constraint on the threshold bias of each unit, and the final M columns specifying the constraints on the connections between the N units. The weights and biases in a neural network are encoded in order as a list. An example is shown in Figure 1.

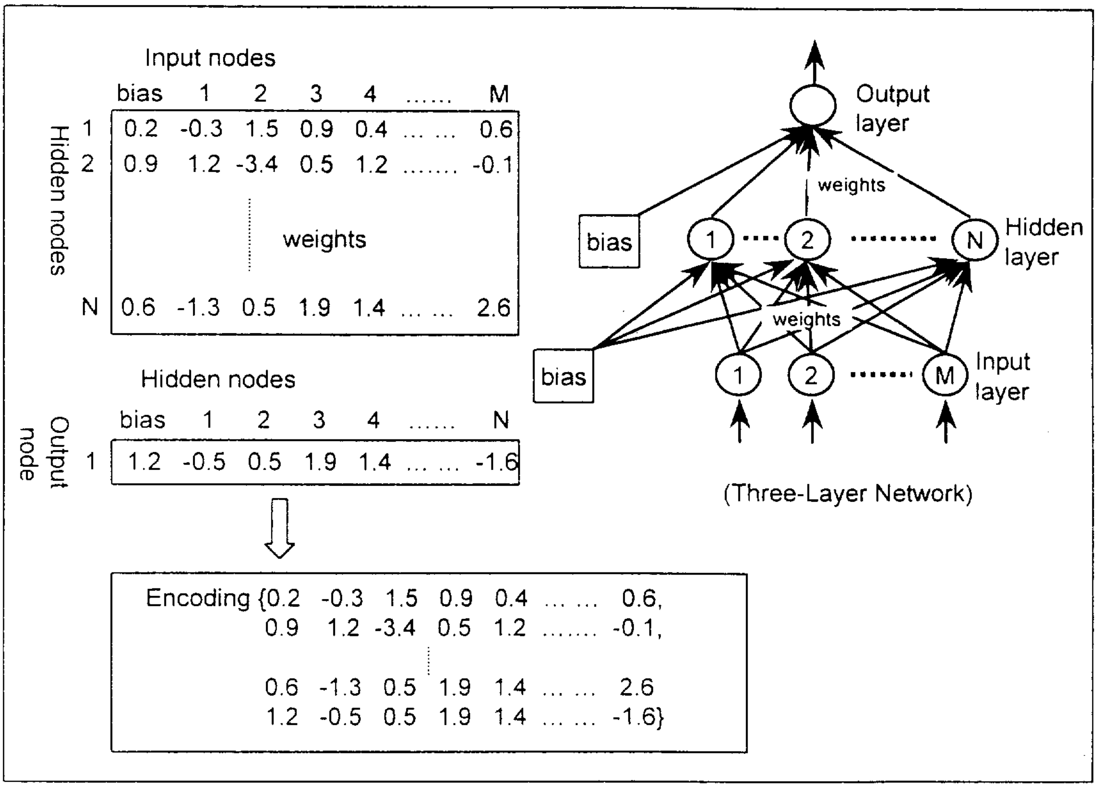

Figure 1. Encoding a network on a string

As shown in Figure 1, each string (chromosome) codes for weights of neural networks. The string of the network is encoded as (w_{11}, w_{12} ... w_{mn}), where each value is a connection weight.

The GAs maintain a population of strings (each of length (M+1)N + (N+1)). The initial members of the population consist of weights, and are chosen at random within some interval, for example, (-1, 1). This initialization allows the GAs to explore the range of all possible solutions, and this tends to favor the most likely solutions. The interval we apply for the neural network weights is (-4,4).

There has been much debate regarding the optimal population size for the problem. Generally, the population size is determined according to the size of the problem (bigger population for larger problem). The common view is that a larger population takes longer to settle on a solution, but is more likely to find a global optimum because of its more diverse gene pool. We use 100 strings in the population for this study.

To evaluate the fitness of a string, the weights on the chromosome are assigned to links in a network of a given architecture, the network is then run over the training set of examples, and the sum of the errors is returned from each example. In other words, the network plays the role of an evaluation function in our GAs. The activation function of the networks we apply is the sigmoid function.

The genetic operators such as crossover and mutation which are described in the previous section are used to search for the optimal weight set solutions. Several parameters must be defined for the above operators, and the values of these parameters can greatly influence the performance of the algorithm. The crossover rate ranges 0.5 - 0.7 and the mutation rate ranges 0.06 - 0.1 for our experiment. As a stopping condition, we use 3,000 trials. These processes are done by the genetic algorithms software package Evolver^{TM} 4.0, called from an Excel macro.

IV. EXPERIMENTS AND RESULTS

4.1 Data and Variable

The data set contains 1,450 small sized manufacturing companies. Among cases, 1,225 companies are bankrupt firms which filed for bankruptcy during the period of 1995 –1997. The financial ratios are used as the input variables of the model. A two-step procedure is used to select input variables. We select input variables satisfying the univariate test first, and then select significant variables by stepwise method for refinement. At the first stage, we select 50 variables by factor analysis, 1-way ANOVA (between input variable and output variable). In the second stage, we select 10 financial variables using stepwise method to reduce the dimensionality. Table 1 illustrates the selected variables for this study.

Table 1. Selected variables

Variables	Names
x1	Inventory (Δ)
x2	Total assets (Δ)
x3	Ordinary income to total assets
x4	Net income to total assets
x5	Retained earnings to total assets
x6	Stockholders' equity to total assets
x7	Total borrowings to total assets
x8	Total borrowings to sales
x9	Inventory turnover
x10	Cash flow to interest expenses

Each data set is split into two subsets, a training set and a validation (holdout) set. The training data are used to train the prediction models. The validation data are used to test the model's results with the data which have not been used to develop the system. For each set of data set (bankrupt and non-bankrupt), a training subset and testing subset, consisting of 80% and 20% of the data, respectively, are randomly selected. We replicate five times (Set1 to 5) of data set selection, estimation and testing process to reduce the impact of random variation in data set composition.

4.2 Experiments and Results

To study the effectiveness of neuro-genetic approach for the bankruptcy prediction modeling, the results obtained by applying the hybrid approach are compared with results from BPN and other statistical classification techniques. Table 2 presents the comparison of the results of these models.

Table 2. Classification accuracies (hit ratio: %)

	Logit		BPN		Neuro-genetic	
	Train	Test	Train	Test	Train	Test
Set 1	71.03	67.59	74.31	71.03	71.98	69.66
Set 2	70.86	67.93	71.81	70.34	71.63	69.66
Set 3	68.88	73.79	71.12	76.21	69.96	75.17
Set 4	71.47	65.17	73.71	68.97	71.98	70.34
Set 5	68.79	71.03	73.45	69.66	71.63	69.66
Average	70.21	69.10	72.88	71.24	71.44	70.90

Among the modeling methods, the neural networks model has the highest level of average accuracy

(71.24%) across the given data sets, followed by neuro-genetic model using genetic training technique (70.90%), and logit model (69.10%). We have expected that neuro-genetic model using genetic training would show a better performance than the back-propagation neural networks considering the fact that GAs perform a more global search than BPN (Kitano, 1990; Sexton, *et al.*,1998; White, 1993). However, the performance of the hybrid approach combining GAs and NNs is slightly inferior to that of the back-propagation. Since the difference of average classification accuracy of five sets is very small (0.34%), we apply the McNemar tests to examine whether the predictive performance of the BPN is significantly higher than that of neuro-genetic approach. The McNemar test is a nonparametric test of the hypothesis that two related dichotomous variables have the same means. This test is useful for detecting changes in responses due to experimental intervention in 'before and after' designs using the chi-square distribution. Since we are interested in the correct prediction of cases, the measure for testing is the classification accuracy rate (the number of correct classification from the number of whole holdout samples). Table 3 shows the results of McNemar tests to compare the classification ability between the BPN and the neuro-genetic model holdout samples. As shown in Table 3, the BPN dose not reject the null hypothesis that the classification results of two methods have the same means.

Table 3. McNemar values for the pairwise comparison of performance between models

	Neuro-genetic Vs. Back-propagation				
	Set 1	Set 2	Set 3	Set 4	Set 5
Chi-Square (P-value)	0.07143 (0.78927)	0.00769 (0.93011)	0.12121 (0.72772)	0.23684 (0.62650)	0.00000 (1.00000)

From the preliminary results above, we conclude that genetic training can be an alternative training algorithm for neural networks learning, although the model does not outperform the back-propagation learning algorithm. We analyze the reasons of rather disappointing results for the following reason. GAs are less competent in local searching since the search in GAs is mainly driven by the crossover operation. Each new individual competes with other individuals instead of the original one, and the competition occurs only once per generation. If we see the entire population as one entity, such a GA is a population hill-climbing method. Local search is important because it could help GAs do a more rapid and comprehensive search than can crossover and mutation (Lin *et al.*, 1995). To increase the effectiveness of GAs, we should consider the way to deal with these problems.

CONCLUDING REMARKS

The aim of integrating different method is to make more powerful and efficient technique by taking advantage of the strength of each method. Therefore, developing more effective technique using synergistic integration is continuing research issue. We still think that GAs and NNs are somewhat complementary each other because GAs have the problem of local search, and NNs are lack of global search ability.

While genetic training approach of NNs have not proven to be better than the best gradient descent methods for this study, this is still a viable research area for situations in which BPN is not effective. For future work, we intended to apply the technique to more complicated problems that the global search is more applicable.

REFERENCE

Angeline, P. J., G. M. Saunders, and J. B. Pollack, (1994), "An evolutionary algorithm that constructs recurrent neural networks," *IEEE Transactions on Neural Networks, vol.* 5, no. 1, pp. 54-65.

Barniv, R., A. Agarwal, and R. Leach, (1997), "Predicting the outcome following bankruptcy filing: A three-state classification using neural networks," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.177-194.

Beer, R. and J. Gallagher, (1992), "Evolving dynamical neural networks for adaptive behavior," *Adaptive Behavior*, 1, pp.91-122.

Bell, T., (1997), "Neural nets or the logit model? A comparison of each model's ability to predict commercial bank failures," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.249-264.

Bishop, J. M., M. J. Bushnell, A. Usher, and S. Westland, (1993), "Genetic optimization of neural network architectures for colour recipe prediction," *Artificial pleural Networks and Genetic Algorithms*, Springer-Verlag, New York, pp. 719-725.

Bornholdt, S. and D. Graudenz, (1992), "General asymmetric neural networks and structure design by genetic algorithms," *Neural Networks, 5*, pp.327-334.

Boritz, J.E., D.B. Kennedy, and A. Albuquerque, (1995), "Predicting Corporate Failure Using a Neural Network Approach," *Intelligent Systems in Accounting, Finance and Management*, vol.4, pp.95-111.

Chen, C., P. Mangiameli, and D. West, (1995), "The Comparative ability of Self-Organizing neural networks to define cluster structure," *Omega*, 23(3), pp.271-279.

Chung, H. and K. Tam, (1992), "A comparative analysis of inductive learning algorithm," *Intelligent Systems in Accounting, Finance and Management*, 2, pp.3-18.

Coats, P.K. and F.L. Fant, (1993), "Recognizing finanical distress patterns using a neural network tool," *Financial Manage.* 22(3), pp.142-156.

Collin, A.M., (1994), "Genetic algorithms for financial modeling," in Deboeck, G.J.(ed.), *Trading on the Edge.* New York: John Wiley, pp.148-173.

Davis, L., (1991), *Handbook of Genetic Algorithms,* Van Nostrand Reinhold, NY.

Dodd, N., (1990), "Optimization of Network Structure Using Genetic Techniques," *Proc. of the Intern. Conf. on Neural Networks*, Paris.

Etheridge, H. and R. Sriram, (1997), "A comparison of the relative costs of financial distress models: Artificial neural networks, logit and multivariate discriminant analysis," *Intelligent Systems in Accounting, Finance and Management*, 6, pp.235-248.

Fahlman, S.E. and C. Lebiere, (1990), "The cascade-correlation learning architecture," In D.S. Touretzky, ed., *Advances in Neural Information Processing Systems II*, San Mateo, CA, Morgan Kaufmann, pp.524-532.

Fletcher, D. and E. Goss, (1993), "Forecasting with neural networks: An application using bankruptcy data," *Information and Management*, 24(3), pp.159-167.

Fogel, D.B., (1993), "Using evolutionary programming to create neural networks that are capable of playing tic-tac-toe," *Proc. IEEE Int. Conf. Neural Networks*, vol. II, San Francisco, pp.875-880.

Fogel, D.B., L.J. Fogel, and V.W. Porto, (1990), "Evolving neural networks," *Biol. Cybern.* 63, pp.487-493.

Goldberg, D.E., (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, MA:Addison-Wesley.

Hansen, J. and R. Meservy, (1996), "Learning experiments with genetic optimization of a generalized regression and neural network," *Decision Support Systems* 18, pp.317-325.

Harp, S. A., and T. Samad, (1991), "Optimizing neural networks with genetic *algorithms," Proceedings of the American Power Conference,* Chicago, pp. 1138-1143.

Harp, S.A., T. Samad, and A. Guha, (1989), "Towards the Genetic Synthesis of Neural Networks," *in Proceedings of the Third International Conference on Genetic Algorithms*, pp.360-369.

Heistermann, J., (1989), "Parallel Algorithms for Learning in Neural Networks with Evolution Strategy," *Parallel Computing*, Vol. 12.

Heistermann, J., (1990), "Learning in Neural Nets by Genetic Algorithms." *Proceedings of Parallel Processing in Neural Systems and Computers (ICNC)*, Eckmiller, R. *et al.* (eds.), Elsevier, pp.165-168.

Holland, J. H., (1975), *Adaptation in Natural and Artificial Systems. Ann Arbor,* The University of

Michigan Press.

Ishibuchi, K. Nozaki and N. Yamamoto, (1993), "Selecting fuzzy rules by genetic algorithm for classification," *Proc. IEEE Int. Conf. Fuzzy Syst.*, San Diego pp.643-650.

Jo, H., I. Han, And H. Lee, (1997), "Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis," *Expert Systems With Applications*, 13 (2), pp.97-108.

Karr, C., (1991), "Genetic Algorithms for Fuzzy Controllers," *AI Expert*, 6(2), pp.26-33.

Kitano, H., (1990), "Empirical Studies on the Speed of Convergence of Neural Network Training using Genetic Algorithms," *in Eighth National Conference on Artificial Intelligence*, Vol. II, AAAI, MIT Press, pp.798-795.

Klimasauskas, C.C., (1992), "Hybrid neuro-genetic approach to trading algorithms," *Advanced Technology for Developers*, 1(7).

Koza, J., (1993), *Genetic Programming*, Cambridge: The MIT Press.

Lacher, R.C., P.K. Coats, and S.C. Sharma, and L. Fant, (1995), "A neural network for classifying the financial health of a firm," *European Journal of Operational Research* Vol. 85(1), PP.53-66.

Lin, S., W. Punch III, and E.D. Goodman, (1995), "A Hybrid Model Utilizing Genetic Algorithms and Hopfield Neural Networks for Function Optimization," *Proceedings of the Sixth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Francisco.

Markham, I.S. and C.T. Ragsdale, (1995), "Combining Neural Networks and Statistical Predictions to Solve the Classification Problem in Discriminant Analysis", *Decision Science*, vol. 26(2), pp.229-242.

Miller, G.F., P.M. Todd, and S.U. Hedge, (1989), "Designing Neural Networks Using Genetic Algorithms," *Proceedings of the 3rd International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA.

Montana, D. and C. Davis, (1989), *Training Feedforward Neuronal Networks Using Genetic Algorithms*, Technical Report, BBN Systems and Technologies Inc., Cambridge(MA).

Mühlenbein, H. and J. Kindermann, (1989), "The Dynamics of Evoluation and Learning – Towards Genetic Neural Networks," in Pfeifer *et al.* (Eds.): *Connectionism in Perspective*, Elsevier.

Odom, M. and R. Sharda, (1990), "A neural networks model for bankruptcy prediction," *Proceedings of the IEEE International Conference on Neural Network*, Vol.2, pp.163-168.

Park, D., A. Kandel, and G. Langholz, (1994), "Genetic-based new fuzzy reasoning models with application to fuzzy control," *IEEE Transactions on Systems, Man and Cybernetics*, 24(1), pp.39-41.

Rummelhart, D.E., G.E. Hinton, and R.J. Williams, (1986), "Learning internal representations by error propagation," In D.E. Rummelhart, J.L. McClelland, *et al.*, eds., *Parallel Distributed Processing*, vol. 1, chap.8. Cambridge, MA: MIT Press.

Salchenberger, L., E. Cinar, and N. Lash, (1992), "Neural networks: A new tool for predicting thrift failures," *Decision Sciences*, 23, pp.899-916.

Schaffer, J. D., D. Whitley, and L. J. Eshelman, (1992), "Combinations of genetic algorithms and neural networks: a survey of the state of the art," *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks,* Baltimore, June 6, pp. 1-37.

Schiffmann, W.H., M. Joost, and R. Werner, (1991), "Performance Evaluation of Evolutionarily Created Neural Network Topologies," *Proceedings of Parallel Problem Solving from Nature*, Schwefel, H.P. and R. Maenner (eds.), pp.274-283.

Schiffmann, W.H., M. Joost, and R. Werner, (1992), "Optimierung des Backpropagation Algorithms zum Training Perceptrons," Fachbericht Physik, 15, Universitat Koblenz.

Schiffmann, W.H., M. Joost, and R. Werner, (1993), "Application of Genetic Algorithms to the Construction of Topologies for Multilayer Perceptrons," *in Proceedings of the International Joint Conference on Neural Networks and Genetic Algorithms*, Innsbruck, pp.675-682.

Schiffmann, W.H. and K. Mecklenburg, (1990), "Genetic Generation of Backpropagation Trained Neural Networks," *Proceedings of Parallel Processing in Neural Systems and Computers(ICNC)*, Eckmiller, R., *et al.* (eds.), pp.205-208.

Sexton, R.S., R.E. Dorsey, and J. D. Johnson, (1998), "Toward global optimization of neural networks: A comparison of the genetic algorithm and backpropagation," *Decision Support Systems* 22, pp.171-185.

Shin, K. S. and I. Han, (1998), "Bankruptcy Prediction Modeling Using Multiple Neural Networks Models, *Proceedings of Korea Management Science Institute Conference*, 1998.

Shin, K. S., T. S. Shin, and I. Han, (1998), "Corporate Credit Rating System Using Bankruptcy Probability Matrix", *Forthcoming Conference Proceedings of IV International Meeting on Artificial Intelligence and Emerging Technologies in Accounting, Finance and Taxation*, Spain.

Shirai, H., *et al.*, (1994), "A Solution of Combinatorial Optimization Problem by Uniting Genetic Algorithms with Hopfield's Model," *IEEE World Congress on Comp. Intl.*, pp.4704-4709.

Syswerda, G., (1989), "Uniform crossover in genetic algorithms," *In J.D. Schaffer, ed., Proc. Third Int. Conf. Genetic Algorithms*, San Meteo, CA: Morgan Kaufmann, pp.2-9.

Tam, K. and M. Kiang, (1992), "Managerial applications of neural networks: the case of bank failure predictions," *Management Science*, 38(7), pp.926-947.

Wang, S., (1995), "The unpredictability of standard backpropagation neural networks in classification applications," *Manage. Sci.* 41(3), pp.555-559.

Wang, Y. and N. Ishii, (1997), "A Method of similarity metrics for structured representations," *Expert Systems with Applications*, 12(1), pp.89-100.

Weiss, G., (1990), *Combining neural and evolutionary learning: Aspects and approaches*. Report FKI-

132-90, Technische Universitat Munchen.

White, D.W., (1993), "GANNet: A Genetic Algorithms for Searching Topology and Weight Spaces in Neural Network Design," Dissertation at the University of Maryland.

Whitely, D., (1988), *Applying Genetic Algorithms to Neural Network Learning*, Technical Report, Department of Computer Science, Fort Collins (Colorado), pp.137-144.

Whitley, D., (1989), "The GENITOR Algorithm and Selection Pressure: Why Rank-Based Allocation of Reproductive Trials in Best," *Proc. of the third Intern. Conf. on Genetic Algorithms*, San Meteo, pp. 116-121.

Whitely, D. and T. Hanson, (1989), "Optimizing Neural Networks Using Faster, More Accurate Genetic Search," *Proc. of the third Intern. Conf. on Genetic Algorithms*, San Mateo, pp. 391-396.

Whitely, D., Starkweather and C. Bogart, (1990), "Genetic Algorithms and Neural Networks: Optimizing Connections and Connectivity," *Parallel Computing* Vol. 14, pp.347-361.

Wilson, R. and R. Sharda, (1994), "Bankruptcy prediction using neural networks," *Decision Support Systems*, 11(5), pp.545-557.

Wong, F. and C. Tan, (1994), "Hybrid neural, genetic and fuzzy systems," In Deboeck, G.J. (ed.), *Trading on the Edge*, New York: John Wiley, pp.245-247.