ELSEVIER

Insurance: Mathematics and Economics 31 (2002) 115–131

INSURANCE
MATHEMATICS & ECONOMICS

www.elsevier.com/locate/econbase

The merging of neural networks, fuzzy logic, and genetic algorithms

Arnold F. Shapiro*

Smeal College of Business, Pennsylvania State University, 310F Business Administration Bldg., University Park, PA 16802, USA

Received 1 February 2002; received in revised form 1 April 2002; accepted 8 April 2002

Abstract

During the last decade, there has been increased use of neural networks (NNs), fuzzy logic (FL) and genetic algorithms (GAs) in insurance-related applications. However, the focus often has been on a single technology heuristically adapted to a problem. While this approach has been productive, it may have been sub-optimal, in the sense that studies may have been constrained by the limitations of the technology and opportunities may have been missed to take advantage of the synergies between the technologies. For example, while NNs have the positive attributes of adaptation and learning, they have the negative attribute of a "black box" syndrome. By the same token, FL has the advantage of approximate reasoning but the disadvantage that it lacks an effective learning capability. Merging these technologies provides an opportunity to capitalize on their strengths and compensate for their shortcomings.

This article presents an overview of the merging of NNs, FL and GAs. The topics addressed include the advantages and disadvantages of each technology, the potential merging options, and the explicit nature of the merging.

Keywords: Actuarial; Fuzzy logic; Fusion; Genetic algorithms; Insurance; Merging; Neural networks; Soft computing

1. Introduction[1]

During the last decade, there has been increased use of neural networks (NNs), fuzzy logic[2] (FL) and genetic algorithms[3] (GAs) in insurance-related applications (Shapiro (2001)). However, the focus often has been on a single technology heuristically adapted to a problem. Examples of this heuristic adaptation are found in the discussions by Brockett et al. (1997, pp. 1156–1157), regarding the number of units needed within a hidden layer of an NN and by Verrall and Yakoubov (1999, p. 192) regarding the number of clusters in a fuzzy clustering application. While this focus on a single technology has been productive, it may have been sub-optimal, in the sense that studies

* Tel.: +1-814-865-3961.
E-mail address: afs1@psu.edu (A.F. Shapiro).

[1] Readers not familiar with neural networks, fuzzy logic or genetic algorithms can find a simple overview of the technical details of these technologies in Shapiro (2000). A working knowledge of these technologies can be obtained by reading Francis (2001) and Brockett et al. (1998) for neural networks, Ostaszewski (1993) for fuzzy logic, and Shapiro (1999) and Wendt (1995) for genetic algorithms.

[2] Following Zadeh (1994, p. 192), in this paper the term fuzzy logic is used in the broad sense where it is essentially synonymous with fuzzy set theory.

[3] GAs are a subset of the broader category of Evolutionary Computing (EC), which is comprised of the evolutionary optimization methods that work by simulating evolution on a computer. The three main subcategories of EC are Genetic Algorithms (GAs), Evolutionary Programming (EP) and Evolution Strategy (ES) (Thomas, 1996). GAs are the most commonly used.

0167-6687/02/$ – see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-6687(02)00124-5

Table 1
Advantages and disadvantages of the NNs, FL and GAs

Technology	Advantage	Disadvantage
NNs	Adaptation, learning, approximation	Slow convergence speed, 'black box' data processing structure
FL	Approximate reasoning	Difficult to tune, lacks effective learning capability
GAs	Systematic random search, derivative-free optimization	Difficult to tune, no convergence criterion

may have been constrained by the limitations of the technology and opportunities may have been missed to take advantage of the synergies between the technologies. This is not a new concern, Bakheet (1995, p. 221), for one, in the conclusion of his NN-based risk assessment study, observed that fuzzy NNs should be an area for future research because of the linguistic nature of the attributes of the decision variables.

Table 1 and the discussion that follows highlight the advantages and disadvantages of each of the technologies and provide a brief description of each of them.

NN, first explored by Rosenblatt (1959) and Widrow and Hoff (1960), are computational structures with learning and generalization capabilities. Conceptually, they employ a distributive technique to store knowledge acquired by learning with known samples. Operationally, they use a training set of samples of input–output relationships and a learning algorithm, called backpropagation (BP), attributed to Werbos (1974), to formulate a supervised[4] learning algorithm that performs local optimization. As indicated in Table 1, NNs have the advantages of adaptation, learning and approximation, but the disadvantages of a relatively slow convergence speed and the negative attribute of a "black box" syndrome.

FL, which was formulated by Zadeh (1965), gives a framework for approximate reasoning and allows qualitative knowledge about a problem to be translated into an executable rule set. This reasoning and rule-based approach are then used to respond to new inputs. As indicated in Table 1, FL has the advantage of approximate reasoning but the disadvantages that it is difficult to construct and tune the fuzzy membership functions (MFs) and rules, and it lacks an effective learning capability.

GAs were proposed by Holland (1975) as a way to perform a randomized global search in a solution space. In this space, a population of candidate solutions, each with an associated fitness value, are evaluated by a fitness function on the basis of their performance. Then, using genetic operations, the best candidates are used to evolve a new population that not only has more of the good solutions but better solutions as well. GAs have the advantages of systematic random search and derivative-free optimization, but they are difficult to tune and have no convergence criterion.

The common thread of these technologies is that they reflect one or more of the soft computing[5] principles of exploiting a tolerance for imprecision, uncertainty, and partial truth to achieve tractability, robustness, and low solution cost. However, because of their individual limitations, their potential can only be realized by taking advantage of the synergy between them. Merging[6] these technologies can capitalize on their strengths and compensate for their shortcomings.

1.1. *The insurance literature*[7]

A number of insurance-related articles have implemented NNs, FL or GAs. This section provides a brief synopsis of these articles and discusses the extent to which two or more of these soft computing technologies have been merged in the studies.

Insurance-related articles that are NN-based have covered the topics of classification, cash flow models, insolvency, and mortality and morbidity studies.

[4] The focus of this study is supervised NN but similar merging opportunities exist for unsupervised networks. See, e.g., Graupe and Kordylewski (1999) and Lokmic and Smith (2000).

[5] While NNs, FL and GAs are only a subset of the soft computing technologies, they are regarded as the three principal components (Shukla, 2000, p. 406).

[6] Engineers generally refer to this process as fusion. However, given the audience of this paper, the term "merging" seems more appropriate.

[7] See Shapiro (2001) for a more detailed overview of these studies.

The classification articles include: Bakheet (1995), who used the feed-forward NN (FFNN) with BP as the pattern classification tool in construction bond underwriting; Vaughn et al. (1997), who used a multilayer perceptron network to classify applicants for life insurance into standard and non-standard risk; and Brockett et al. (1998), who used a Kohonen self-organizing feature map (SOFM) to uncover automobile bodily injury claims fraud in the insurance industry and an FFNN with BP to validate the feature map approach. An example of the cash flow model was the study by Lokmic and Smith (2000), who investigated the problem of forecasting when checks issued to retirees and beneficiaries will be presented for payment, in order to forecast daily cash flow requirements. There were a number of insolvency studies, including: Park (1993), who used an NN designed by a GA to predict the bankruptcy of insurance companies; Brockett et al. (1994), who used a three-layer FFNN with BP to develop an early warning system for US property-liability insurers 2 years prior to insolvency; Huang et al. (1994), who used an NN optimized with a GA to forecast financial distress in life insurers and Jang (1997) undertook a similar comparative analysis. The mortality and morbidity studies included: Tu (1993), who compared an NN and logistic regression model on the basis of their ability to predict length of stay in the intensive care unit following cardiac surgery; Saemundsson (1996), who used an NN to address problems related to dental care insurance; and Ismael (1999), who investigated the feasibility of using NN technology to reliably predict in-hospital complications and 30-day mortality for acute myocardial infarction patients.

There were five major insurance areas where FL has been implemented: underwriting, classification, pricing, assets and investments, and projected liabilities. The underwriting articles include: DeWit (1982), who was the first to use fuzzy theory to analyze the internal logic of the intuitive part of insurance underwriting; Lemaire (1990), whose underwriting focus was on the definition of a preferred policyholder in life insurance; Young (1993), who used fuzzy sets to model the selection process in group health insurance; Erbach and Seah (1993), who discussed an early prototype life automated underwriter based on a mixture of fuzzy and other techniques and Horgby et al. (1997), who applied FL to medical underwriting of life insurance applicants. The classification articles include: Ebanks et al. (1992), who showed how measures of fuzziness can be used to classify life insurance risks; Ostaszewski (1993, pp. 47–67), who presented a detailed, yet simple, illustration of how fuzzy clustering can be used to classify four individuals; Cox (1995), who compared individual medical provider behavior to that of a peer group in order to isolate potential abuse in health insurance; Derrig and Ostaszewski (1995), who showed how fuzzy clustering can be used to classify insurance claims according to their suspected level of fraud; Hellman (1995), who used a fuzzy expert system to identify municipalities that were of average size and well managed, but whose insurance coverage was inadequate; and Verrall and Yakoubov (1999), who showed how fuzzy clustering could be used to specify a data-based procedure for investigating age groupings in general insurance. Articles on pricing included: Lemaire (1990), who discussed the computation of the fuzzy premium for an endowment policy; Young (1996), who described how FL can be used to make pricing decisions in group health insurance; Young (1997), who described step-by-step how an actuary/decision maker could use FL to adjust workers compensation insurance rates; and Cummins and Derrig (1997), who used FL to address the financial pricing of property-liability insurance contracts. Asset and investment studies included the articles of Chang and Wang (1995), who developed fuzzy mathematical analogues of the classical immunization theory and the matching of assets and liabilities and Derrig and Ostaszewski (1997), who illustrated how FL can be used to estimate the effective tax rate and after-tax rate of return on the asset and liability portfolio of a property-liability insurance company. Articles which deal with projected liabilities included: Boissonnade (1984), who used pattern recognition and FL in the evaluation of seismic intensity and damage forecasting, and for the development of models to estimate earthquake insurance premium rates and insurance strategies; Cummins and Derrig (1993), who studied fuzzy trends in property-liability insurance claim costs, and Zhao (1996), who addressed the issues of maritime collision prevention and liability.

Insurance applications of GAs have included classification, optimizing the competitiveness of an insurance product, and asset allocation. Examples of these studies, other than those previous mentioned, include: Frick et al. (1996), who used GAs to generate stock buy and sell strategies based on the classification of difference price formations and Lee and Kim (1999), who used GAs to refine the classification system of Korean private passenger automobile insurance; Tan (1997), who examined profitability, risk, and competitiveness trade-offs; and Wendt (1995) and Jackson (1997) who investigated asset allocation, Wendt by comparing the portfolio efficient frontier of a GA to that of a sophisticated non-linear optimizer and Jackson by investigating the performance of a GA as a function of discontinuities in the search space.

For the most part, these insurance studies have not taken advantage of the synergy between the NN, FL and GA technologies. With the exceptions of Park (1993), Huang et al. (1994), Bakheet (1995) and Brockett et al. (1998), who explicitly mentioned using a GA to determine the architecture and/or parameters of their NN, none of the other NN studies mentioned FL or GAs. Only Zhao (1996), who integrated NNs and FL, merged FL and the other SC technologies. None of the articles in the GA section mentioned NNs or FL.

1.2. *The purpose of this paper*

A plausible explanation of why most of the foregoing insurance studies have not merged the technologies, or even mentioned the possibility of doing so, is that the authors were not sufficiently familiar with the merging opportunities. Assuming this to be the case, and that it is a continuing problem, the purpose of this paper is to help alleviate this situation by presenting an overview of the merging of NNs, FL and GAs. The subjects addressed include the advantages and disadvantages of each technology, the potential merging options, and the explicit nature of the merging.

The context of this article can be envisioned as the intersections of the NN, FL and GA sets in Fig. 1. The specific topics will include: NNs controlled by FL, NNs generated by GAs, fuzzy inference systems (FISs) tuned by GAs, FISs tuned by an NN, GAs controlled by FL, and the merging of all three technologies. The paper concludes with a commentary on the prognosis for these hybrids.

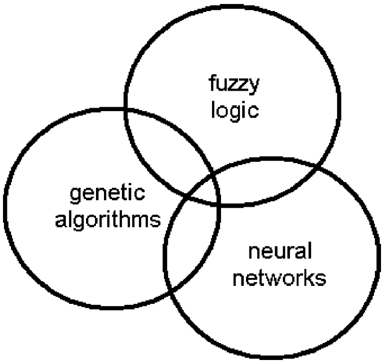

Fig. 1. The intersections of NNs, FL and GAs.

2. NNs controlled by FL

The system where NNs are controlled by FL is referred to as fuzzy NNs. Its history can be traced back to at least the mid-1970s, when Lee and Lee (1974, 1975) introduced MFs to the McCulloch and Pitts (1943) model. Since then, FL has been applied to many of the components and parameters of NNs, including the inputs, weights, learning rate and momentum coefficient. This section presents an overview of these methodologies.

2.1. *Inputs and weights*

To begin, consider the neural processing unit, which is the core of the NN, an example of which is shown in Fig. 2. As indicated, the inputs (signals) to the neuron, x_j, are multiplied by their respective weights, w_j, and aggregated. The weight w_0 serves the same function as the intercept in a regression formula.[8] The weighted sum is then passed through an activation function, F, to produce the output of the unit. The activation function often takes the form of the logistic function $F(z) = (1 + e^{-z})^{-1}$, where $z = \sum_j w_j x_j$, as shown in the figure, but other functions may be used.[9]

Following Buckley and Hayashi (1994a, p. 234), we adopt the convention that in order for an NN to be a fuzzy NN (FNN), the signal and/or the weights must be fuzzy sets. This leads to their three possibilities:

[8] Anders (1996) investigates neural networks as a generalization of non-linear regression models.

[9] Typically, a sigmoidal (S-shaped) function is used.

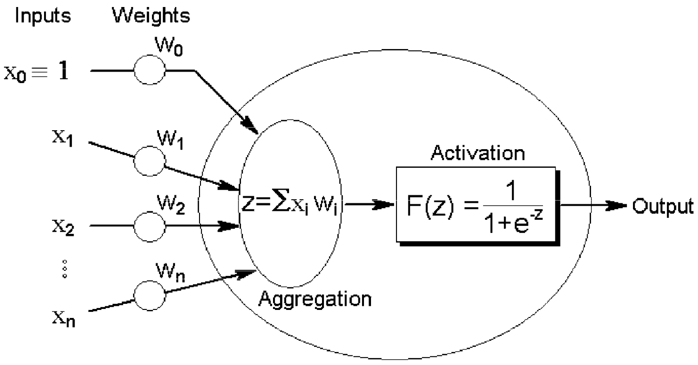

Fig. 2. Neural processing unit.

- FNN$_1$ is an FNN with real inputs but fuzzy weights.
- FNN$_2$ is an FNN with fuzzy inputs but real weights.
- FNN$_3$ is an FNN with fuzzy inputs and fuzzy weights.

Moreover, they called an FNN a *regular* FNN if its neural processing units use multiplication, addition and a logistic activation function, as depicted in Fig. 2, in contrast to a *hybrid* FNN, which uses operations like a t-norm[10], t-conorm, or some other continuous procedure, to combine the incoming signals and weights and to aggregate their products. This distinction is important because a regular FNN that uses fuzzy arithmetic based on Zadeh's extension principle is not a universal approximator[11] while a hybrid FNN is (Buckley and Hayashi, 1994b; Fullér, 1995).

An example of the implementation of FNN$_1$ is the approach envisioned by Yamakawa (1990), Yamakawa and Furukawa (1992), Yamakawa and Tomoda (1989) and Yamakawa et al. (1992), which can be characterized as shown in Fig. 3.

Here, for each input, x_i, instead of a single weight, w_i, the neuron has an array of weights $\{w_{ij}, j = 1, m\}$, each of which is associated with a triangular fuzzy number, μ_{ij}. Since, for any given x_i, only two adjacent MFs are non-zero, the input to the neuron associated with each x_i is the weighted average of the two adjacent weights (Buckley and Hayashi, 1994a, p. 235), i.e.

$$\mu_{ij} w_{ij} + \mu_{i,j+1} w_{i,j+1}$$

and the total input is the sum of these. In the Yamakawa studies, learning was accomplished by updating the weights using a heuristic rule.

Considering the simple three-layer FFNN shown in Fig. 4 easily extends the analysis.

The first layer, the input layer, has three neurons (labeled n_{0j}, $j = 0, 1, 2$), the second layer, the hidden processing layer, has three neurons (labeled n_{1j}, $j = 0, 1, 2$), and the third layer, the output layer, has one neuron (labeled n_{21}). There are two inputs signals, $x1$ and $x2$. The neurons are connected by the weights w_{ijk}, where the subscripts i, j, and k refer to the ith layer, the jth node of the ith layer, and the kth node of the $(i + 1)$th layer, respectively. Thus, e.g., w_{021} is the weight connecting node 2 of the input layer (layer 0) to node 1 of the hidden layer (layer 1).FNN$_2$ was explored early on by Ishibuchi et al. (1992a,b), who envisioned both the inputs, $\bar{x}_i$ (where the bar indicates a fuzzy value), and the output, $\bar{y}$, as fuzzy numbers,

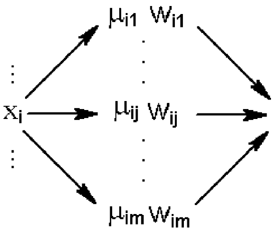

Fig. 3. Yamakawa fuzzy weights.

[10] The simplest examples of the t-norm (triangular-norm) and the t-conorm (its dual) are the min-operator and max-operator, respectively.

[11] A particular form of a neural net is a universal approximator if, given a continuous function $F : \Re^n \rightarrow \Re$, it can approximate F uniformly on compact subset of $\Re$ to any degree of accuracy.

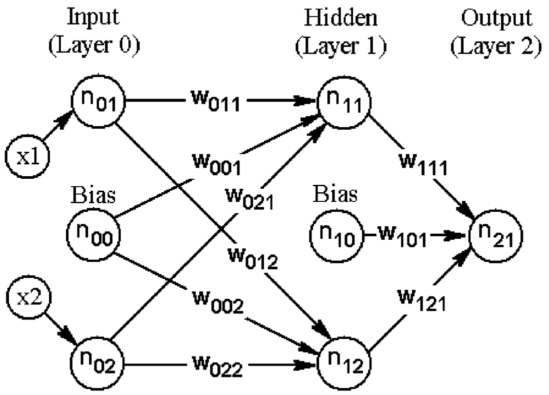

Fig. 4. A simple representation of an FFNN.

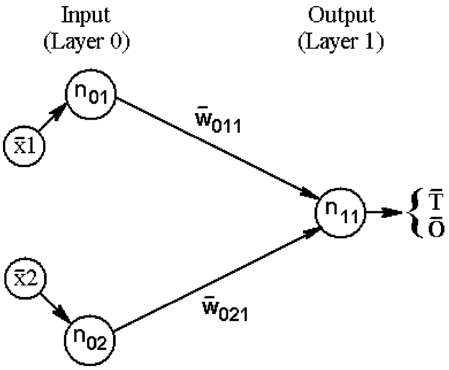

Fig. 5. Simple type-3 FNN.

within the context of an NN along the lines of Fig. 4. Their procedure, generally speaking, was to view the jth training instance as the paired fuzzy input and target numbers $(\bar{X}_j, \bar{T}_j)$, where $\bar{X}_j = (\bar{x}_{1j}, \dots, \bar{x}_{nj})$, and to use interval arithmetic[12] to compute the error to be minimized. They then updated the weights using a modified version of the delta rule.[13]

Finally, a simple version of FNN3 can be envisioned as depicted in Fig. 5. As previously, the jth training instance is the paired fuzzy numbers $(\bar{X}_j, \bar{T}_j)$. If universal approximation is not an issue, standard fuzzy arithmetic can be used for the input to the neuron and the extension principle[14] can be applied to the activation function. For computations of the error involving fuzzy subtraction, the stopping rule could be to end the iterations when the error falls within an acceptable interval about zero. The learning algorithm could then be based on a fuzzified delta rule.[15] Of course, the potential applications for FNN_3 is limited if the methodology is restricted to regular FNNs. Buckley and Hayashi (1994a, pp. 240–242) provide an example of a hybrid FNN_3.

The subsequent literature in this area generally focused on specifying modeling capabilities of FNNs and changes in the architecture and/or internal operations so that the FNNs become universal apoximators. Two examples of recent studies of the implementation problems inherent in these systems are Feuring et al. (1998), who discuss a BP-based method for adjusting the weights, and Jiao et al. (1999), who focus on problems involving a crisp input and a fuzzy target.

2.2. *Learning rates and momentum coefficients*

In addition to its role in modifying the inputs and weights of an NN, FL can be used to monitor and control the parameters of the NN. Consider the formula for the change in the weights

$$\Delta w(t) = -\eta \nabla E[w(t)] + \alpha \Delta w(t-1)$$

where $\Delta w(t)$ denotes the change in the weight, η denotes the learning rate, $E[w(t)]$ denotes the error function during the tth iteration, ∇E denotes the gradient of E in weight space, and α the momentum coefficient.

Following Kuo et al. (1993) and Bonissone (1998), we can modify the learning rate and the momentum coefficient by incorporating MFs derived from the total training error and the change in error between two successive iterations (ΔError). In the case of the former, the universe of discourse may be small, medium and big, while the latter may represent negative, zero and positive. An example of the matrix of fuzzy rules could be as shown in Table 2.[16]

[12] Interval arithmetic is an arithmetic defined on sets of intervals, rather than sets of real numbers. See Babad and Berliner (1995) for a discussion of interval arithmetic in an insurance context.

[13] See Ishibuchi et al. (1992a, pp. 1299–1300). The delta rule is also known as the gradient (or steepest) decent method.

[14] The extension principle allows non-fuzzy arithmetic operations to be extended to incorporate fuzzy sets and fuzzy numbers. See Zimmermann (1996, Chapter 5).

[15] A three-layered network with fuzzified delta rules is discussed in Hayashi et al. (1992, 1993).

[16] Adopted from Bonissone (1998), Table 1.

Table 2
Fuzzy rule table for learning rate (η) and momentum (α)

ΔError	Training error		
	Small	Medium	Large
Negative	Very small increase	Very small increase	Small increase
Zero	No change	No change	Small increase
Positive	Small decrease	Medium decrease	Large decrease

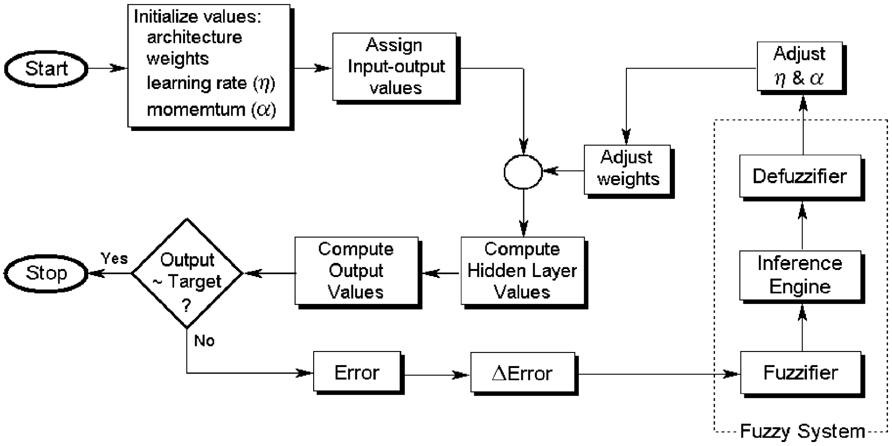

Fig. 6. Neural-fuzzy system.

Even with such simple fuzzy rules, the training time will be reduced significantly (Bonissone, 1998, p. 66). A similar procedure can be used to monitor and control the steepness parameter[17] of the activation function, except that ΔError might be replaced by training time, partitioned into short, medium and long.

Given the foregoing fuzzy rule table, the neural-fuzzy model might be designed along the lines shown in Fig. 6.

In this representation, the whole system is composed of two parts, the NN component[18] and the fuzzy system. Given the initialization and the input and target values, the NN computes the component values of the hidden layer and its estimate of the output values. It then evaluates its performance and if the output is not sufficiently close to the target value, it passes the error and the change in the error to the fuzzy system. The fuzzy system adaptively determines the necessary changes in the learning rate and momentum in accordance with the fuzzy rules of Table 2. Since the fuzzy system in this version of the model used the Mamdani approach (see the processor section of Fig. 8), the conclusions of the inference engine are defuzzified to convert them to a crisp form, which is used to adjust the learning and momentum rates if necessary. BP is then used to adjust the weight matrix and the process continues.[19]

3. NNs generated by GAs

Since Montana and Davis (1989) first proposed the use of GAs to train an FFNN with a given topology,

[17] If the activation function is conceptualized as $(1 + \exp\{-z\beta/2\})^{-1}$, β is referred to as the steepness parameter.

[18] This representation only shows the training portion of the NN.

[19] In the system envisioned in Fig. 6, the NN uses backpropagation to update the weights. However, the weight matrix could have been updated based on the information from the fuzzy system (Thammano, 1999).

Table 3
Values associated with the chromosome

First 3 digits	000	001	010	011	100	101	110	111
Hidden cells	2	3	4	5	6	7	8	9
Second 3 digits	000	001	010	011	100	101	110	111
Learning rate	0.1	0.3	0.5	0.7	0.9	1.1	1.3	1.5
Last 2 digits	00	01	10	11				
Momentum	0.1	0.3	0.5	0.7				

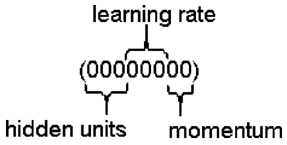

Fig. 7. The genetic code.

a number of ways that GAs can be used to improve the performance of NNs have been identified.[20] Options include: developing the architecture, including the number of hidden layers, nodes within the layers, and connectivity; optimizing the weights for a given architecture, thus replacing BP; and selecting the parameters of the NN, such as the learning rate and momentum coefficient (Trafalis, 1997, p. 409; Bonissone, 1999, p. 7; Yao, 1999). There is, of course, the issue that GAs are not designed to be ergodic and cover the space in the most efficient manner, but this is more than offset by the efficiency gain resulting from the parallalization capability (Bonissone, 1998, p. 68).

The general procedure for implementing GAs in these fashions is exemplified by the insurance company bankruptcy study conducted by Park (1993). An interesting portion of Park's study was his explicit discussion of the use of a GA to suggest the optimal structure for the NN. The basic strategy is depicted in Fig. 7, which shows the eight digits generated by the GA: the first three digits are used to set the number of hidden units, the next three digits set the leaning rate, and the last two digits determine the momentum.

In Park's study, the values associated with the binary chromosome were as show in Table 3. Thus, e.g., if the generated binary digits are (00100100), the number of hidden units is 3, the learning rate is 0.3 and the momentum is 0.1.

The implementation is straightforward. A solution of the GA population, such as (00100100), is used to construct a network, which is then trained by the BP method using a training set of observations. The network is then tested with a test set of observations and its output is used to determine its fitness. In this hybrid, the fitness function often takes the form $(\mathrm{MSE} + c)^{-1}$, where MSE denotes mean square (output) error and c is a small positive number used to avoid arithmetic overflows. This process is repeated for each solution in the GA population.

It is common for studies involving NNs to use BP as the tuning algorithm. While this is an efficient approach where the error surface is convex, it can be problematic where the surface is multimodal because the method may get trapped in a sub-optimal local minimum. This problem can be circumvented by repeated searches with different initial conditions or by perturbing the weights when the search seems stagnated, but a more robust approach would be to use a global search method like GAs.

Of course, GAs are not without their shortcoming. While they are very effective at global searches, and can quickly isolate a global minimum, they may be inefficient at actually finding that minimum. This issue can be addressed with a hybrid approach like that of Kitano (1990), whereby the GA is used to isolate the global minimum after which BP is used for the local search. Alternatively, the GA could simply be used to free the BP in the event that it gets stuck in a local minimum (McInerney and Dhawan (1993)). Finally, it is worth noting that Bakheet (1995, pp. 114–115) preferred to optimize the NN structure heuristically rather than use the structure suggested by a built-in genetic optimizer, because he felt he could produce a better solution.

[20] According to Trafalis (1997, p. 409), most researchers use GAs as supportive combinations to assist NNs.

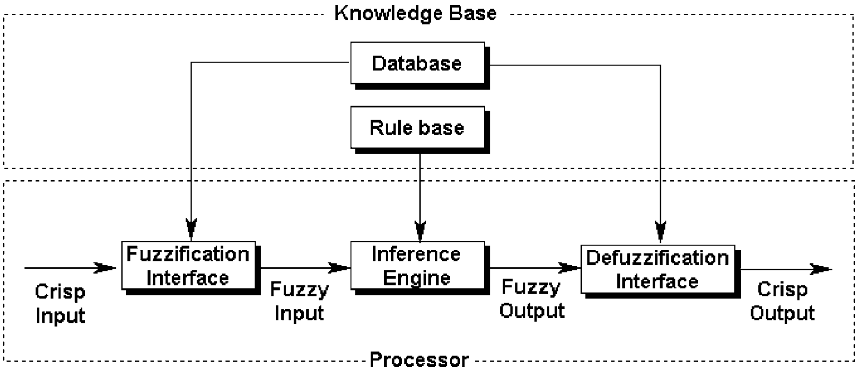

Fig. 8. An FIS.

4. FISs tuned by GAs

The fuzzy inference system, FIS[21], is a popular methodology for implementing FL. The essence of the system can be represented as shown in Fig. 8.

As indicated in the figure, the FIS can be envisioned as involving a knowledge base and a processing stage. The knowledge base provides MFs and fuzzy rules needed for the process. In the processing stage, numerical crisp variables are the input of the system.[22] These variables are passed through a fuzzification stage where they are transformed to linguistic variables, which become the fuzzy input for the inference engine. This fuzzy input is transformed by the rules of the inference engine to fuzzy output. The linguistic results are then changed by a defuzzification stage into numerical values that become the output of the system.

GAs tune fuzzy systems either by adapting the fuzzy MFs and/or by facilitating the learning of the fuzzy if-then rules. The advantage of the GAs is that they use simple operations to provide robust methods for the designing and automatic tuning of fuzzy system parameters.

Following Jang et al. (1997, p. 485), the hierarchical structure of the input and output portion of a GA that tunes an FIS can be conceptualized[23] as shown in Fig. 9.

Interpreting the figure from the top to the bottom, the GA is based on a population of FISs. Each FIS has inputs and outputs, the fuzzy portions of which are represented by MFs. Here, the MFs are portrayed as trapezoids, since they have four parameters, but they could also be portrayed as triangular or Guassian.[24] Finally, a unit of eight bits called a gene represents each parameter.

In addition to the input and output portions of the chromosome, there will be a rule base substring and a rule consequence substring, representations of which are shown in Fig. 10.[25]

In the figure, the x_i's, $i = 1, \ldots, N$, and y are the input and output variables, respectively, and M is the number of rules. The rule base substring, the first substring, which is composed of integer values, encodes the structure of each rule and assigns a unique number to each MF. A zero implies that the input variable is not involved in the rule. The rule consequence substring has a similar interpretation, except that a zero implies that the rule is deleted from the FIS rule base. The inclusion of the option to use zeros allows both

[21] FISs are also known as fuzzy rule-based systems, fuzzy expert systems, fuzzy models, fuzzy associative memories (FAMs), or fuzzy logic controllers when used as controllers (Jang et al., 1997, p. 73).

[22] The numerical input can be crisp or fuzzy. In this latter event, the input does not have to be fuzzified.

[23] This simple conceptualization needs to be refined in practice because of implementation issues related to the specifics of crossover and mutation operations and structural level adaptations. See Jang et al. (1997, pp. 484–485).

[24] Gaussian MFs are represented by bell-shaped functions of the form $\exp\{-(x-c)^2/2w^2\}$, where c, the center, is the value in the domain around which the curve is built and w is the width parameter.

[25] Adapted from Fig. 1 of Liska and Melsheimer (1994, p. 1379) and their discussion.

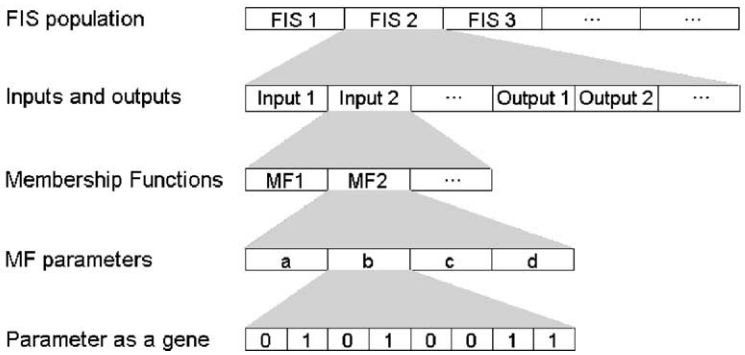

Fig. 9. GA hierarchical representation of the input and output portions of an FIS.

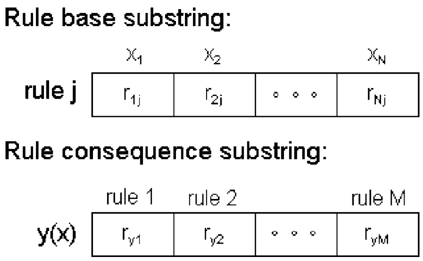

Fig. 10. Rule base and rule consequence substrings.

the number of input variables involved in each rule and the number of rules to change dynamically during the GA search, giving added flexibility.

Before proceeding we need to define some of the terms endemic to the area, such as scaling factor, termset and ruleset: the scaling factor determines the ranges of values for the state and output variables; the termset defines the MF associated with the values taken by each state and output variable,[26] and the ruleset characterizes a syntactic mapping from a state to an output. The structure of the underlying model is the ruleset, while the model parameters are the scaling factors and termsets (Bonissone et al., 1999, p. 4).

A basic approach to tuning FIS by GAs is to use GAs to learn fuzzy set MFs only, with a fixed set of rules set by hand. This was the approach of Karr (1991a,b, 1993), who used GAs to modify the MFs in the termsets of the variables. Following a procedure similar to that mentioned above, he used a binary encoding to represent the parameters defining a membership value in each termset and then concatenated the termsets to produce the binary chromosome.

A commonly referenced study is Lee and Takagi (1993b) who tuned both the rule base and the termsets. In their case, they assumed triangular MFs and used a binary encoding for the associated three-tuples. Following the Takagi–Sugeno–Kang (TSK) rule, under which a first-order polynomial, defined on the state space, is the output of each rule in the ruleset (see the next section), the chromosomes were constructed by concatenating the membership distributions with the polynomial coefficients.

5. FISs tuned by an NN

Ever since Lee and Lee (1974) first proposed merging FL and NNs using the paradigm of a multi-input multi-output NN, there has been considerable interest the concept. Many have simply propagated some variation of the original paradigm under the premise that a fuzzy neural system is little more than a multi-layered FFNN.[27] Others, however, have interpreted the process as involving fuzzy reasoning and inference facilitated by a connectionist network.

The most common approaches (He et al., 1999, p. 52) have been where the NNs tune the MFs, given a defined set of rules, and where the NNs are designed to autonomously generate the rules. The ANFIS

[26] For example, the termset of the training error in Table 2 is small, medium and large.

[27] It is worth noting that there is not an absolute dichotomy between NNs and FL. Li and Chen (1999), e.g., demonstrate that FL systems and FFNN are equivalent in essence.

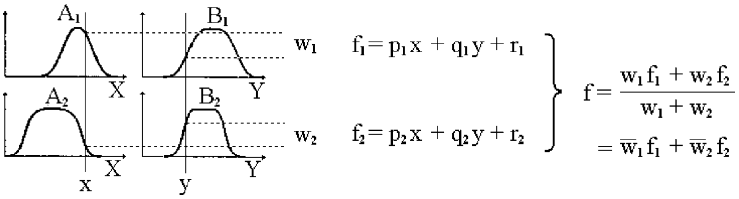

Fig. 11. TSK-type FIS.

(adaptive neural fuzzy inference system) of Jang (1993) was one of the earlier programs in this category and is often cited.

The basis of the ANFIS is the TSK FIS (Takagi and Sugeno (1983, 1985)) in which the conclusion of the fuzzy rule is a weighted linear combination of the crisp inputs rather than a fuzzy set. For a first-order TSK model (Jang et al., 1997, p. 336), a common rule set with n fuzzy if-then rules is

$$\text{If } x \text{ is } A_i \text{ and } y \text{ is } B_i,\ \text{then } f_i = p_i x + q_i y + r_i,\quad i = 1, \ldots, n,$$

where x and y are linguistic input variables, A_i and B_i, are the corresponding fuzzy sets, f_i is the output of each rule, and p_i, q_i, and r_i are linear parameters. A simple two-input one-output representation is depicted in Fig. 11.

Operationally, the first step is to find the membership grades of the IF parts of the rules, which, in the figure, are represented by the heights of the dashed lines. Then, since the pre-conditions in the IF parts are connected by AND, the firing strength of each rule if found using multiplication. For example, the firing strength for rule 1, w_1, is the product of the heights of the top two dashed lines. Given, w_1 and w_2, the overall output is computed as a weighted average.

As noted by Jang (1993), this is functionally equivalent to the ANFIS depicted in Fig. 12.

In the figure, the square nodes are adaptive nodes and have parameters while the circle nodes are fixed nodes and do not. Also, the links only indicate the flow direction of signals between the nodes, there are no weights associated with the links, unlike a regular NN. The five layers may be characterized as follows:

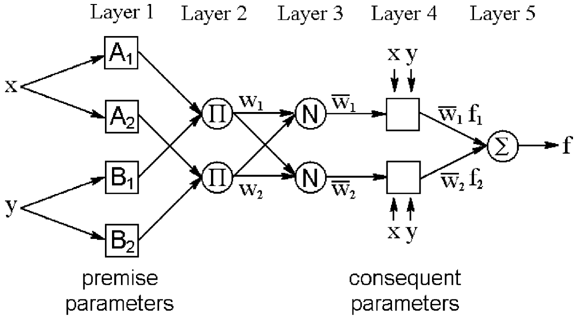

Fig. 12. Two-input ANFIS with two rules.

- Layer 1, labeled as “premise parameters[28]”, has adaptive nodes, where the A_j’s and B_j’s are linguistic labels (such as “low” or “high”). The output of the layer is the membership grade that specifies the degree to which the inputs satisfy the quantifiers.
- Layer 2, the fixed nodes of which are denoted Π, multiplies the incoming signals and sends the product out. Any t-norm operator that performs generalized AND can be used as a node function in this layer.
- Layer 3 is a fixed node, denoted N for normalization, which calculates the relative weight of the ith rule’s firing strength. Outputs of this layer are called normalized firing strengths and take the value $w_i/\sum w_j$.
- Layer 4, labeled “consequence parameters[29]”, applies the first-order TSK fuzzy if-then rules to the output of layer 3. The output of node i of this layer is $f_i w_i/\sum w_j$.

[28] The premise also is referred to as the conditional or left-hand side of a rule.

[29] The consequence also is referred to as the action or right-hand side of a rule.

- Layer 5 is a single fixed node, denoted $\sum$ that computes the overall output as the summation of all incoming signals and is equal to $\sum f_i w_i / \sum w_j$.

Bonissone (1999) characterizes Jang's approach as a "two-stroke" optimization process. During the forward stroke the termsets of the first layer equal their previous iteration value while the coefficients of the fourth layer are computed using a least mean square method. Then, ANFIS computes an output, which, when compared with the related training set value, gives the error. The error gradient information is then used in the backward stroke to modify the fuzzy partitions of the first layer. This process is continued until convergence is attained.

A number of studies have extended the basic ANFIS model or a variation of it. Examples include: Mizutani (1997), who extended the single output ANFIS model to a multiple-output ANFIS with non-linear fuzzy rules; Juang and Lin (1998), who developed a self-constructing neural fuzzy inference network that was inherently a modified TSK-type fuzzy rule-based model possessing NN learning ability; and Abdelrahim and Yahagi (2001), who use principal component methodology[30] to uncorrelate and remove redundancy from the input space or the ANFIS. An interesting overview can be found in Abraham and Nath (2000).

6. GAs controlled by FL

A major limitation of GAs is their potential for premature convergence to an inferior solution (Herrera and Lozano, 1996). Essentially, GAs are very efficient during the global search in the solution space but tend to bog down when the search becomes localized. The original attempts to overcome this problem were based on intuition and experience and while a number of strategies for choosing the GA parameters evolved (Pearl, 1988; Kim and Pearl, 1983), they invariably required the parameters to be computed off-line and to be kept static during the algorithm's evolution. What was needed was an adaptive approach.

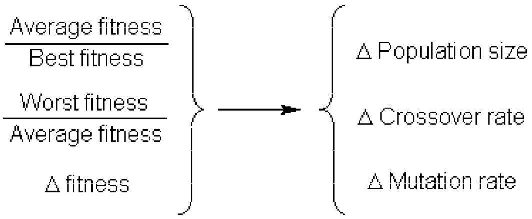
Average fitness / Best fitness
Worst fitness / Average fitness
Δ fitness
Δ Population size
Δ Crossover rate
Δ Mutation rate

Fig. 13. Sample input–output variables.

FL was a natural candidate for resolving this problem because it provides a vehicle for easily translating qualitative knowledge about the problem to be solved into an executable rule set. In the case at hand, FL can be used to provide dynamic control of GA resources (such as population size, selection pressure, and probabilities of crossover and mutation) during the transition from the global search to the local search, and thereby improve the algorithms' performance.

Lee and Takagi (1993a) exemplify the process of run-time tuning when they discuss the use of a fuzzy system which uses the three input variables shown in Fig. 13 to determine the current state of the GA evolution and to produce the three output variables.

They also imposed a relative change limitation on the output variables, which could not change by more than half the current setting, and boundary conditions.[31]

Thus, e.g., as far as the population is concerned, a typical fuzzy control scheme may include rules such as

If (average fitness/best fitness) is large,
then population size should increase,

and

If (worse fitness/average fitness) is small,
then population size should decrease.

These could be implemented using a matrix of fuzzy rules along the lines of Table 2.

7. Neuro-fuzzy-genetic systems

Given the foregoing discussions, it is not surprising that methodologies have evolved for merging all three of the NNs, FL and GAs technologies. Of course,

[30] Principal component analysis is a methodology for finding the structure of a cluster located in multidimensional space. Conceptually, it is equivalent to choosing that rotation of the cluster which best depicts its underlying structure.

[31] Population size, crossover rate and mutation rate were limited to the operational ranges of [10, 160], [0.2, 1.0] and [0.0001, 1.0], respectively.

the added complexity causes new layers of problems. Some of these problems are obvious: there will be an increased number of parameters that need to be tuned and an increase in the time needed to learn rules. The focus in the resolution of these types of issues will be on more efficient searches and more dynamic routines. For example, Gaussian MFs, since they have only two parameters, rather than three (triangular) or four (trapezoidal), can be used to develop more efficient search strings, and dynamic crossover and mutation probability rates can be incorporated. Others problems may only be obvious in retrospect: the augmented design stages may not be independent or if the new design involves a restricted search space, it may be more likely to result in partial or sub-optimal solutions. Here, among other things, simultaneous optimization may have to be implemented.

Recent examples of studies that have merged all three of the NNs, FL and GAs technologies are: Seng et al. (1999), who proposed a neuro-fuzzy controller based on the Gaussian type radial basis functions NN,[32] where all of its parameters were simultaneously tuned by a GA; Duarte and Tomé (2000), who developed a fuzzy adaptive learning control network, in which GAs are used to find the connections between layers of the NN and the parameter values of the nodes; Kuo et al. (2001), who developed a GA-based FNN (GFNN) whereby the GA is used first to obtain a "rough" solution and avoid local minima, and then the FNN is used to fine-tune the results and Huang et al. (2001), who developed an integrated neural-fuzzy-genetic-algorithm, where the NNs were used to generate fuzzy rules and MFs and the GAs were used to optimize the defuzzification operator parameters.

8. Conclusions

The purpose of this study has been to explore ways in which the synergy between NNs, FL and GAs has been exploited and to survey some of the successful combinations that have lead to the development of hybrid systems. Table 4 summarizes the types of merging the paper discussed.

Table 4
Merged technologies

	Modified technology		
	NN	FL	GA
Modified by			
NN		Y	–
FL	Y		Y
GA	Y	Y	

As indicated, with the exception of NN modifying GAs,[33] there was a discussion of each of the technologies modifying each of the other technologies.

We saw how merging the technologies provided an alternative to a strictly knowledge-driven reasoning system or a purely data-driven one and learned that the merged techniques can provide a more accurate and robust solution than can be derived from any single technique. Moreover, there is relatively little redundancy because typically the methods do not try to solve the same problem in parallel but they do it in a mutually complementary fashion.

The NN-based insurance studies reviewed in the introduction suggest avenues to explore that involve the synergies between the technologies. One enhancement would be the use of fuzzy signals and/or weights. Many of the studies used linguistic-type input variables characterized by crisp MFs and they could be extended using the fuzzy NNs of Buckley and Hayashi (1994a). Another enhancement opportunity follows from the observation that, while some researchers used GAs to choose NN parameters, such as the learning rate and momentum coefficient, their approach was static. It would be interesting to compare their results with those obtained using a dynamic fuzzy rules approach, along the lines of Bonissone (1998).

Only one of the FL-based insurance studies reviewed in the introduction took advantage of the synergies between FL and the other SC technologies so there is ample opportunity to extend the studies.

[32] The Radial Basis Function NN is a faster alternative to the NN with BP. The essential features of the network is that it consists of three layers, an input layer, a single hidden layer and an output layer, and training takes place in two stages, where the first stage detects clusters in the input data and second stage determines the optimal weights for the output layer. See Bishop (1995, Chapter 5) and Ham and Kostanic (2001, Section 3.6).

[33] This is not to say there are no examples of NN modifying GAs, but, rather, that few such articles were found. Kassicieh et al. (1998), e.g., investigated the use of a NN to transform an economic series before a GA processed the series.

One possibility would be to use GAs to tune the fuzzy systems either by adapting the fuzzy MFs and/or by facilitating the learning of the fuzzy if-then rules. Following Karr (1993), GAs could be used to learn fuzzy set MFs only, with a fixed set of rules set by hand, or in the manner of Liska and Melsheimer (1994), the entire process could be automated. Another area worth exploring is the use of NNs to tune the FISs. The methodology to consider in this regard is the ANFIS of Jang (1993) or one of its derivatives.

In addition to the foregoing opportunities to merge GAs with the other SC technologies, another important topic is the tendency of GAs to converge to inferior solutions. It would be interesting to investigate whether a methodology like Lee and Takagi (1993a) could improve the results of the GA-based insurance studies.

Many actuarial and insurance problems involve imprecision, uncertainty and partial truths. Since NNs, FL and GAs can help resolve these issues, more researchers are beginning to implement them. As we improve our understanding of the strengths and weaknesses of these technologies and improve the manner by which we leverage their best features, it seems inevitable that they will become an increasingly more important component of actuarial methodology.

Acknowledgements

This work was supported in part by the Robert G. Schwartz Faculty Fellowship and the Smeal Research Grants Program at the Pennsylvania State University. The assistance of Angela T. Koh and Travis J. Miller is gratefully acknowledged.

References

Abdelrahim, E.M., Yahagi, T., 2001. A new transformed input-domain ANFIS for highly nonlinear system modeling and prediction. In: Proceedings of the Canadian Conference on Electrical and Computer Engineering, Vol. 1, pp. 655–660.

Abraham, A., Nath, B., 2000. Hybrid intelligent systems design—a review of a decade of research. Technical Report (5/2000). Gippsland School of Computing and Information Technology, Monash University, Australia.

Anders, U., 1996. Statistical Model Building for Neural Networks, In: Proceedings of the Sixth AFIR Colloquium.

Babad, Y.M., Berliner, B., 1995. Reduction of uncertainty through actualization of interval possibilities and its application to reinsurance. TICA 2, 23–35.

Bakheet, M.T., 1995. Contractors risk assessment system (surety, insurance, bonds, construction, underwriting). Ph.D. Dissertation. Georgia Institute of Technology.

Bishop, C.M., 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Boissonnade, A.C., 1984. Earthquake Damage and Insurance Risk. Ph.D. Dissertation. Stanford University.

Bonissone, P.P., 1998. Soft computing applications: the advent of hybrid systems, in: Bosacchi, B., Fogel, D.B., Bezdek, J.C. (Eds.), Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation, Proceedings of the SPIE, Vol. 3455, pp. 63–78.

Bonissone, P.P., 1999. Hybrid soft computing systems: where are we going, Working Paper. GE Corporate Research and Development.

Bonissone, P.P., Chen, Y., Goebel, K., Khedkar, P.S., 1999. Hybrid soft computing systems: industrial and commercial applications. Proceedings of the IEEE 87 (9), 1641–1667.

Brockett, P.L., Cooper, W.W., Golden, L.L., Pitaktong, U., 1994. A neural network method for obtaining an early warning of insurer insolvency. Journal of Risk and Insurance 61 (3), 402.

Brockett, P.L., Cooper, W.W., Golden, L.L., Xia, X., 1997. A case study in applying neural networks to predicting insolvency for property and casualty insurers. Journal of the Operational Research Society 48 (12), 1153–1162.

Brockett, P.L., Xia, X., Derrig, R.A., 1998. Using Kohonen's self-organizing feature map to uncover automobile bodily injury claims fraud. JRI 65 (2), 245–274.

Buckley, J.J., Hayashi, Y., 1994a. Fuzzy neural networks. In: Yager, R.R., Zadeh, L.A. (Eds.), Fuzzy Sets, Neural Networks and Soft Computing, Van Nostrand Reinhold, New York, pp. 233–249.

Buckley, J.J., Hayashi, U., 1994b. Hybrid fuzzy neural nets are universal approximators. In: Proceedings of the Third IEEE Conference on Fuzzy Systems, Vol. 1, 1994, pp. 238–243.

Chang, C., Wang, P., 1995. The matching of assets and liabilities with fuzzy mathematics. In: Proceedings of the 25th International Congress of Actuaries, p. 123.

Cox, E., 1995. A fuzzy system for detecting anomalous behaviors in healthcare provider claims. In: Goonatilake, S., Treleven, P. (Eds.), Intelligent Systems for Finance and Business. Wiley, New York.

Cummins, J.D., Derrig, R.A., 1993. Fuzzy trends in property-liability insurance claim costs. Journal of Risk and Insurance 60, 429–465.

Cummins, J.D., Derrig, R.A., 1997. Fuzzy financial pricing of property-liability insurance. North American Actuarial Journal 1 (4), 21.

Derrig, R.A., Ostaszewski, K., 1995. Fuzzy techniques of pattern recognition in risk and claim classification. Journal of Risk and Insurance 62, 447–482.

Derrig, R.A., Ostaszewski, K., 1997. Managing the tax liability of a property-liability insurance company. Journal of Risk and Insurance 64, 694.

DeWit, G.W., 1982. Underwriting and uncertainty. Insurance: Mathematics and Economics 1, 277–285.

Duarte, C., Tomé, J.A.B., 2000. An evolutionary strategy for learning in fuzzy neural networks. In: Proceedings of the 19th International Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 24–28.

Ebanks, B., Karwowski, W., Ostaszewski, K., 1992. Application of measures of fuzziness to risk classification in insurance. In: Computing and Information, Proceedings of the ICCI'92, Fourth International Conference, pp. 290–291

Erbach, D.W., Seah, E., 1993. Discussion of the application of fuzzy sets to group health underwriting by V.R. Young. Transactions of the Society of Actuaries 45, 585.

Feuring, T., Buckley, J.J., Hayashi, Y., 1998. Adjusting fuzzy weights in fuzzy neural nets. In: Proceedings of the Second International Conference on Knowledge-Based Intelligent Electronic Systems, pp. 402–406.

Francis, L., 2001. Neural networks demystified. Casualty Actuarial Society Forum, Winter, pp. 253–319.

Frick, A., Herrmann, R., Kreidler, M., Narr, A., Seese, D., 1996. Genetic-based trading rules—a new tool to beat the market with—first empirical results. AFIR 2, 997–1017.

Fullér, R., 1995. Neural fuzzy systems, Lecture Notes. Abo Akademi University.

Graupe, D., Kordylewski, H., 1999. A large-memory neural network for controlling unknown systems from exact and fuzzy observations. In: Dagli, C.H., Buczak, A.L., Ghosh, J., Embrechts, M.J., Ersoy, O. (Eds.), Smart Engineering System Design. ASME Press, New York, pp. 89–94.

Ham, F.M., Kostanic, I., 2001. Principles of Neurocomputing for Science and Engineering. McGraw-Hill, New York.

Hayashi, Y., Buckley, J.J., Czogala, E., 1992. Direct fuzzification of neural network and fuzzified delta rule. In: Proceedings of the Second International Conference on Fuzzy Logic Neural Networks (IIZUKA'92), Iizuka, Japan, July 17–22, pp. 73–76.

Hayashi, Y., Buckley, J.J., Czogala, E., 1993. Fuzzy neural network with fuzzy signals and weights. International Journal of Intelligent Systems 8, 527–537.

He, L., Wang, K., Jin, H., Li, G., Gao, X.Z., 1999. The combination and prospects of neural networks, fuzzy logic and genetic algorithms. In: Proceedings of the 1999 IEEE Midnight-Sun Workshop on Soft Computing Methods in Industrial Applications. Kuusamo, Finland, June 16–18.

Hellman, A., A fuzzy expert system for evaluation of municipalities—an application. In: Proceedings of the 25th TICA, Vol. 1, 1995, pp. 159–187.

Herrera, F., Lozano, M., 1996. Adaptive genetic algorithms based on fuzzy techniques. In: Proceedings of the IPMU'96, Granada, Spain, pp. 775–780.

Holland, J.H., 1975. Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, MA.

Horgby, P., Lohse, R., Sittaro, N., 1997. Fuzzy underwriting: an application of fuzzy logic to medical underwriting. Journal of Actuarial Practice 5 (1), 79.

Huang, C., Dorsey, R.E., Boose, M.A., 1994. Life insurer financial distress prediction: a neural network model. Journal of Insurance Regulation Winter 13 (2), 131–167.

Huang, Y., Gedeon, T.D., Wong, P.M., 2001. An integrated neural-fuzzy-genetic-algorithm using hyper-surface membership functions to predict permeability in petroleum reservoirs. Engineering Applications of Artificial Intelligence 14, 15–21.

Ishibuchi, H., Fujioka, R., Tanaka, H., 1992a. An architecture of neural networks for input vectors of fuzzy numbers. In: Proceedings of the IEEE International Conference on Fuzzy System (FUZZ-IEEE 92), San Diego, March 8–12, pp. 1293–1300.

Ishibuchi, H., Okada, H., Tanaka, H., 1992b. Interpolation of fuzzy if-then rules by neural networks. In: Proceedings of the Second International Conference on Fuzzy Logic Neural Networks (IIZUKA'92), Iizuka, Japan, July 17–22, pp. 337–340.

Ismael, M.B., 1999. Prediction of mortality and in-hospital complications for acute myocardial infarction patients using artificial neural networks. Ph.D. Dissertation. Duke University.

Jackson, A., 1997. Genetic algorithms for use in financial problems. AFIR 2, 481–503.

Jang, J.-S.R., 1993. ANFIS: adaptive-network-based fuzzy inference systems. IEEE Transactions on Systems, Man, and Cybernetics 23 (3), 665–685.

Jang, J., 1997. Comparative analysis of statistical methods and neural networks for predicting life insurers' insolvency (bankruptcy). Ph.D. Dissertation. University of Texas, Austin.

Jang, J.-S.R., Sun, C.-T., Mizutani, E., 1997. Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River, NJ.

Jiao, L.C., Liu, F., Wang, L., Zhang, Y.N., 1999. Fuzzy wavelet neural networks: theory and applications. In: Chen, G., Ying, M., Cai, K.-Y. (Eds.), Fuzzy Logic and Soft Computing. Kluwer Academic Publishers, Boston.

Juang, C.-F., Lin, C.-T., 1998. An online self constructing neural fuzzy inference network and its applications. IEEE Transactions on Fuzzy Systems 6 (l), 12–32.

Karr, C.L., 1991a. Genetic algorithms for fuzzy controllers. AI Expert 6 (2), 27–33.

Karr, C.L., 1991b. Design of an adaptive fuzzy logic controller using genetic algorithms. In: Proceedings of the International Conference on Genetic Algorithms (ICGA'91), San Diego, CA, USA, pp. 450–456.

Karr, C.L., 1993. Fuzzy control of pH using genetic algorithms. IEEE Transactions on Fuzzy Systems 1, 46–53.

Kassicieh, S.K., Paez, T.L., Vora, G., 1998. Data transformation methods for genetic-algorithm-based investment decisions. In: Proceedings of the 31st Hawaii International Conference on System Sciences, 1060-3425/98 IEEE, pp. 122–127.

Kim, J., Pearl, J., 1983. A computational model for causal and diagnostic reasoning in inference engines. In: Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Karlsruhe, Germany.

Kitano, H., 1990. Empirical studies on the speed of convergence of neural network training using genetic algorithms. In: Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90), Vo1. 2, pp. 789–795.

Kuo, R.J., Chen, Y.T., Cohen, P.H., Kumara, S., 1993. Fast convergence of error back propagation algorithm through fuzzy modeling. Intelligent Engineering Systems through Artificial Neural Net works, pp. 239–244.

Kuo, R.J., Chen, C.H., Hwang, Y.C., 2001. An intelligent stock trading decision support system through integration of genetic algorithm based fuzzy neural network and artificial neural network. Fuzzy Sets and Systems 118 (1), 21–45.

Lee, B., Kim, M., 1999. Application of genetic algorithm to automobile insurance for selection of classification variables: the case of Korea. In: Paper Presented at the 1999 Annual Meeting of the American Risk and Insurance Association.

Lee, S.C., Lee, E.T., 1974. Fuzzy sets and neural networks. Journal of Cybernetics 4, 83–103.

Lee, S.C., Lee, E.T., 1975. Fuzzy neural networks. Mathematical Biosciences 23, 151–177.

Lee, M.A., Takagi, H., 1993a. Dynamic control of genetic algorithm using fuzzy logic techniques. In: Forrest, S. (Ed.), Proceedings of the Fifth International Conference on Genetic Algorithms, Morgan Kaufmann, Los Altos, CA, pp. 76–83.

Lee, M.A., Takagi, H., 1993b. Integrating design stages of fuzzy systems using genetic algorithms. In: Proceedings of the Second IEEE Conference on Fuzzy Systems (FUZZ-IEEE'93), San Francisco, CA, USA, pp. 612–617.

Lemaire, J., 1990. Fuzzy insurance. ASTIN Bulletin 20 (1), 33–55.

Li, H., Chen, C.L.P., 1999. The equivalence between fuzzy logic systems and feed-forward neural networks. In: Dagli, C.H., Buczak, A.L., Ghosh, J., Embrechts, M.J., Ersoy, O. (Eds.), Smart Engineering System Design. ASME Press, New York, pp. 535–540.

Liska, J., Melsheimer, S.S., 1994. Complete design of fuzzy logic systems using genetic algorithms. In: Proceedings of Third IEEE International Conference on Fuzzy Systems, pp. 1377–1382.

Lokmic, L., Smith, K.A., 2000. Cash flow forecasting using supervised and unsupervised neural networks. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks, Vol. 6, pp. 343–347.

McCulloch, W.S., Pitts, W., 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics 5, 115–133.

McInerney, M., Dhawan, A.P., 1993. Use of genetic algorithms with backpropagation in training of feedforward neural networks. In: Proceedings of the 1993 IEEE International Conference on Neural Networks.

Mizutani, E., 1997. Coactive neuro-fuzzy modeling: toward generalized ANFIS. In: Jang, J.-S.R., Sun, C.-T., Mizutani, E. (Eds.), Neuro-fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River, NJ, pp. 369–400.

Montana, D.J., Davis, L., 1989. Training feedforward neural networks using genetic algorithms. In: Proceedings of the Eleventh International Joint Conference on Artificial Intelligence (IJCAI89), Vol. l, pp. 762–767.

Ostaszewski, K., 1993. Fuzzy Set Methods in Actuarial Science. Society of Actuaries, Schaumburg, IL.

Park, J., 1993. Bankruptcy prediction of banks and insurance companies: an approach using inductive methods. Ph.D. Dissertation. University of Texas, Austin.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Mateo, CA.

Rosenblatt, F., 1959. Two theorems of statistical separability in the perceptron. In: Mechanization of Thought Processes, Symposium held at the National Physical Laboratory, HM Stationary Office, pp. 421–456.

Saemundsson, S.R., 1996. Dental caries prediction by clinicians and neural networks. Ph.D. Dissertation. University of North Carolina, Chapel Hill.

Seng, T.L., Khalid, M.B., Yusof, R., 1999. Tuning of a neuro-fuzzy controller by genetic algorithm. IEEE Transactions on Systems, Man, and Cybernetics 29 (2), 226–236.

Shapiro, A., 1999. The inner workings of neural networks and genetic algorithms. ARCH 1, 415–426.

Shapiro, A., 2000. A hitchhiker's guide to the techniques of adaptive nonlinear models. Insurance: Mathematics and Economics 26, 119–132.

Shapiro, A., 2001. Soft Computing Applications in Actuarial Science, ARCH, 2001. http://www.soa.org:80/library/arch/2000-09/arch01v113.pdf.

Shukla, K.K., 2000. Soft computing paradigms for artificial vision. In: Sinha, N.K., Gupta, M.M. (Eds.), Soft Computing and Intelligent Systems: Theory and Applications. Academic Press, San Diego, pp. 405–417.

Takagi, T., Sugeno, M., 1983. Derivation of fuzzy control rules from human operator's control actions. In: Proceedings of the IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, July, pp. 55–60.

Takagi, T., Sugeno, M., 1985. Fuzzy identification of systems and its applications to modeling and control. IEEE Transactions on Systems, Man, and Cybernetics 15 (1), 116–132.

Tan, 1997. Seeking the profitability-risk-competitiveness frontier using a genetic algorithm. Journal of Actuarial Practice 5 (1), 49.

Thammano, A., 1999. Neural-fuzzy model for stock market prediction. In: Dagli, C.H., Buczak, A.L., Ghosh, J., Embrechts, M.J., Ersoy, O. (Eds.), Smart Engineering System Design. ASME Press, New York, pp. 587–591.

Thomas, B., 1996. Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms. Oxford University Press, Oxford.

Trafalis, T.B., 1997. Genetic algorithms in neural network training and applications to breast cancer diagnosis. In: Smart Engineering Systems: Neural Networks, Fuzzy Logic, Data Mining, and Evolutionary Programming, Proceedings of the ANNIE'97, Vol. 7, pp. 409–414.

Tu, J.V., 1993. A comparison of neural network and logistic regression models for predicting length of stay in the intensive care unit following cardiac surgery. Ph.D. Dissertation. University of Toronto.

Vaughn, M.L., Ong, E., Cavill, S.J., 1997. Interpretation and knowledge discovery from a multilayer perceptron network that performs whole life assurance risk assessment. Neural Computing and Applications 6, 201–213.

Verrall, R.J., Yakoubov, Y.H., 1999. A fuzzy approach to grouping by policyholder age in general insurance. Journal of Actuarial Practice 7, 181–203.

Wendt, R.Q., 1995. Build your own GA efficient frontier. Risks and Rewards, December.

Werbos, P., 1974. Beyond regression: new tools for predictions and analysis in the behavioral science. Ph.D. Thesis. Harvard University, Cambridge, MA.

Widrow, B., Hoff, M.E., 1960. Adaptive switching circuits. IRE Western Electric Show and Convention Record, Part 4, pp. 96–104.

Yamakawa, T., 1990. Pattern recognition hardware system employing a fuzzy neuron. In: Proceedings of the International Conference on Fuzzy Logic, Iizuka, Japan, July 20–24, pp. 943–938.

Yamakawa, T., Furukawa, M., 1992. A design algorithm of membership functions for a fuzzy neuron using example-based learning. In: Proceedings of the IEEE International Conference on Fuzzy System (FUZZIEEE 92), San Diego, March 8–12, pp. 943–948.

Yamakawa, T., Tomoda, S., 1989. A fuzzy neuron and its application to pattern recognition. In: Proceedings of the Third IFSA Congress, Seattle, August 6–11, pp. 30–38.

Yamakawa, T., Uchino, E., Miki, T., Kusanagi, H., 1992. A neo fuzzy neuron and its application to system identification and prediction of the system behavior. In: Proceedings of the Second International Conference on Fuzzy Logic Neural Networks (IIZUKA 92), Iizuka, Japan, July 17–22, pp. 477–483.

Yao, X., 1999. Evolving artificial neural networks. Proceedings of the IEEE 87 (9), 1423–1447.

Young, V.R., 1993. The application of fuzzy sets to group health underwriting. Transactions of the Society of Actuaries 45, 551–590.

Young, V.R., 1996. Insurance rate changing: a fuzzy logic approach. Journal of Risk and Insurance 63, 461–483.

Young, V.R., 1997. Adjusting indicated insurance rates: fuzzy rules that consider both experience and auxiliary data. Proceedings of the Casualty Actuarial Society 84, 734.

Zadeh, L.A., 1965. Fuzzy sets. Information and Control 8, 338–353.

Zadeh, L.A., 1994. The role of fuzzy logic in modeling, identification and control. Modeling Identification and Control 15 (3), 191.

Zhao, J., 1996. Maritime Collision and Liability. Ph.D. Dissertation. University of Southampton.

Zimmermann, H.J., 1996. Fuzzy Set Theory and Its Applications, third ed. Kluwer Academic Publishers, Boston, MA.