Research Article

Optimization the Initial Weights of Artificial Neural Networks via Genetic Algorithm Applied to Hip Bone Fracture Prediction

Yu-Tzu Chang,[1] Jinn Lin,[2] Jiann-Shing Shieh,[1,3] and Maysam F. Abbod[4]

[1] *Department of Mechanical Engineering, Yuan Ze University, 32003 Chungli, Taiwan*
[2] *Department of Orthopaedic Surgery, National Taiwan University Hospital, Taipei, Taiwan*
[3] *Center for Dynamical Biomarkers and Translational Medicine, National Central University, Taoyuan City, Taiwan*
[4] *School of Engineering and Design, Brunel University, London, UK*

Correspondence should be addressed to Maysam F. Abbod, maysam.abbod@brunel.ac.uk

This paper aims to find the optimal set of initial weights to enhance the accuracy of artificial neural networks (ANNs) by using genetic algorithms (GA). The sample in this study included 228 patients with first low-trauma hip fracture and 215 patients without hip fracture, both of them were interviewed with 78 questions. We used logistic regression to select 5 important factors (i.e., bone mineral density, experience of fracture, average hand grip strength, intake of coffee, and peak expiratory flow rate) for building artificial neural networks to predict the probabilities of hip fractures. Three-layer (one hidden layer) ANNs models with back-propagation training algorithms were adopted. The purpose in this paper is to find the optimal initial weights of neural networks via genetic algorithm to improve the predictability. Area under the ROC curve (AUC) was used to assess the performance of neural networks. The study results showed the genetic algorithm obtained an AUC of 0.858 ± 0.00493 on modeling data and 0.802 ± 0.03318 on testing data. They were slightly better than the results of our previous study (0.868 ± 0.00387 and 0.796 ± 0.02559, resp.). Thus, the preliminary study for only using simple GA has been proved to be effective for improving the accuracy of artificial neural networks.

1. Introduction

With increment of the life expectancy among the world, osteoporosis becomes more and more prevalent and may lead to disastrous pathological fractures. For the year 2000, there were an estimated 9 million new osteoporotic fractures, 1.6 million at the hip, 1.7 million at the forearm, and 1.4 million at the vertebrae. Europe and the Americas accounted for 51% of these osteoporotic fractures, while most of the remainder occurred in the Western Pacific region and Southeast Asia [1]. Hip fractures cause the most morbidity with a reported mortality rate up to 20–24% in the first year after a hip fracture [2, 3], and greater risk of dying may persist for at least 5 years afterwards [4]. Hip fractures are invariably associated with severe chronic pain, reduced mobility,

disability, and an increasing degree of dependence [5]; even if the patients survive after the incidence, some of them still suffer its subsequent complications [6]. Furthermore, the patients have to shoulder the huge health and economic burdens that caused a high health care expenditure.

In order to reduce the occurrence of this preventable injury and its subsequent complication, we expected to find out the risk factors that are important for fracture prevention and health promotion and then build a predictor for the probability of hip bone fracture. Recently, support vector machines (SVMs) have been applied in machine learning techniques and are state-of-the-art machine learning techniques for risk minimization [7]. Since their invention, research on SVMs has exploded both in theory and applications. In practice, SVMs have been successfully applied to many real-world domains [8, 9]. However, in dealing with

highly nonlinear and complex system like hip fracture, artificial neural network (ANN) is still better than SVMs because so many hidden layers, nodes, and parameters (e.g., learning constant, learning algorithms, initial weights, etc.) can be adjusted in ANN. Also, in previous study [10], although many potential risk factors for hip fracture have been identified, these risk factors may vary geographically and combined effects of different risk factors have not been well understood. Then, they established the artificial neural network (ANN) to predict the risk of hip bone fracture according to the advantages of nonlinearity, fault tolerance, universality, and real-time operation.

ANNs are computer programs that simulate some of the higher level functions of the human brain. There are neurons and synapses in the brain, with various synaptic connection strengths—called "weights"—between the connected neuron pairs. The so-called input and output neurons for each problem correspond to the inputs and to outputs from a traditional computer program. The other, called "hidden" neurons, along with the synapses and weights, comes between the input and output neurons corresponding to the instructions in a traditional program. Use of ANNs as clinical prediction models has been explored in many areas of medicine, including nephrology, and microbiology, radiology, neurology.

Backpropagation is a topology of artificial neural network; it adjusts the network's weights and biases by calculating the gradient of the error. Usually, backpropagation neural networks are applied with random initial weight setting because of symmetry breaking [11]. However, training the neural networks with random initial weights may cause two main drawbacks: trapping into local minima and converging slowly [11, 12]. In view of these limitations of back-propagation neural networks, global search techniques (e.g., genetic algorithm and particle swarm optimization) have been presented to overcome these shortcomings [13, 14]. So far a number of works compare the evaluation between back-propagation neural network and genetic algorithm for training neural networks [15, 16], both of them are techniques for optimization and learning.

Genetic algorithms (GAs) developed to mimic some of the processes observed in natural evolution are a class of global search algorithms techniques. They have been shown in practice to be very effective at function optimization, and searching large or complex (multimodal, discontinuous, etc.) spaces to find nearly global optimum efficiently [11]. Therefore, this study tried to find the optimal initial weights of artificial neural network via genetic algorithm so that the predictor could enhance the ability of predicting the risk of hip bone fracture.

This paper is arranged as follows. Section 2 gives an overview of artificial neural networks and genetic algorithm. The materials and the artificial neural network prediction model are used in Section 3. In Section 4, the proposed genetic algorithm model is explained. Results and discussion are in Section 5. Conclusions and future work are in Section 6.
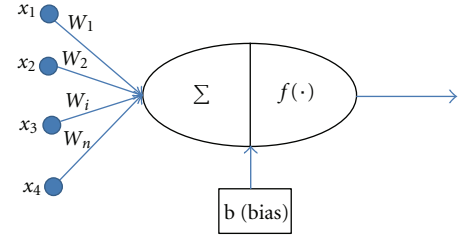

Figure 1: The diagram of one neuron.

2. An Overview of Neural Networks and Genetic Algorithm

2.1. Artificial Neural Networks. Artificial neural networks are a system that emulates the process of biological neural networks. Artificial neural networks generally consist of five components.

(1) The directed graph of the ANN topology.

(2) A state variable associated with each neuron.

(3) A real-valued weight associated with each link.

(4) A real-valued bias associated with each neuron.

(5) The output of each neuron $f(\sum w_i x_i - b)$, which is the input for next layer, where f is the transfer function, w_i are the weights connected with each neuron at the last layer, x_i are the input values of the neurons, and b is the bias of the neuron (Figure 1).

Artificial neural networks have become very popular for a few reasons. Firstly, they have the capability of learning what adjusts the weights and biases between the nodes. If the prediction is correct, the weights of the connections will be increased and vice versa [17]. Secondly, artificial neural networks are a parallel system that can deal with missing data that the linear program cannot deal with. Thirdly, with multiple layers, artificial neural networks can process nonlinearity even though the relationships between multifactor variables have not been exactly understood.

Feedforward network is one of the artificial neural network topologies. It usually consists of multiple layers, and the information will just be communicated to the next layer (i.e., output nodes have no arcs away from them). By different tactics for modifying the weights in training networks, some types of feedforward are presented such as back-propagation neural network. Back-propagation neural network is one that calculates the gradient of the error and then propagates error backward through the network to modify the weights and biases.

2.2. Genetic Algorithm. Genetic algorithms are developed to mimic some of the processes inspired by natural evolution. There are five components that we should define first [18, 19]:

(1) a way of coding solution to the problem on chromosomes;

(2) a fitness function which returns a value for each chromosome given to it;

(3) a way of initializing the population of chromosomes;

(4) operators that may be applied to parents when they reproduce to alter their genetic composition standard operators are mutation and crossover;

(5) parameter settings for the algorithm, the operators, and so forth.

With these definitions, genetic algorithm operates in the following steps.

(1) Encode the problem in a string and generate the initial population using initialization procedure.

(2) Reckon the fitness value for each chromosome. It will directly react on the distance to the optimum.

(3) Reproduce until a stopping criterion is met; reproduction consists of iterations of the following steps.

   (a) Choose a number of parents to reproduce; selection is stochastic, but the individuals with the highest evaluations are favored in the selection.

   (b) Apply the genetic algorithms (e.g., crossover, mutation) to the parents.

   (c) Accumulate the children and evaluate the fitness value. Insert children into the population to replace worse individuals of the current population.

In genetic algorithm, fitness function will evaluate the adaptation of each individual; it is a key point to decide if the outcome is good or not. Selection operator will choose adaptive parents depending on their fitness values. By this step, the population tends towards better individuals. Crossover operator and mutation operator make the chromosomes reach a wider search space [20].

3. Materials and Prediction Model

3.1. Study Sample. The sample data were gathered in the previous case-control matched study for the analysis of risk factors of hip fracture for older adults aged 60 and older [21] and predicting the risk of hip bone fracture for elders in Taiwan by ensemble backpropagation neural networks [10]. The sample included 228 cases who were the patients admitted to the National Taiwan University Hospital with first low-trauma hip fracture and 215 patients in the same hospital, but without hip fracture.

Both cases and controls were interviewed by trained interviewers with the same standardized questionnaire, which included questions about basic and social demography, history of diseases and conditions, self-rated overall health, health habits, intake of food and nutritional supplements, falls and fracture experiences, living environment and potential home hazards, physical functioning and use of assistive devices, and cognitive, and other functions. Athropometric measures and physical assessments less influenced by lower extremity function were performed after the questionnaire interview, including body height, weight, handgrip strength, peak expiratory flow, and coordination test. Bone mineral density (BMD) was examined at the nonfractured side of proximal femur for cases and the same side for matched controls by dual-energy X-ray absorptiometry (DXA) using the same machine of DEXA (Model: QDR 4500A; Hologic, Waltham, MA, US), and read by the same radiologist in 153 cases and 197 hospital controls. Leisure-time physical activity in the health habits was measured using total energy expended on all leisure-time activities in a week. The physical functions were measured by questions on the level of difficulty in performing 5 ADL, 6 IADL, and 8 mobility tasks. Cognitive function was measured with the Mini-Mental State Examination (MMSE). Height and weight were measured using electronic scales. BMI was calculated as weight in kg/height in m^2. Grip strength was measured with a hand-held hydraulic dynamometer (Model: NC70142; North Coast Medical, Morgan Hill, CA, US). The participants used the dominant hand, and three maximal values were averaged. Peak expiratory flow was assessed by using a peak flow meter (Model: Standard Mini Wright; Clement Clarke International, Harlow, Essex, UK). The participants took a deep breath and blew as fast and vigorously as possible. The maximum of three trials was chosen as the peak flow. The finger-nose-finger test was conducted by asking the participants to use their finger to alternately touch their own nose and the interviewers' finger as quickly as possible for assessing the coordination. A total of 78 variables were measured.

Because the number of variables was too large to be collected rapidly in clinic, logistic regression was applied to filter out irrelevant factors with two steps: univariable analysis and multivariable analysis. After these analyses, five significant factors (i.e., bone mineral density, experience of fracture, average hand grip strength, intake of coffee, and peak expiratory flow rate) remained to be the variables of neural networks. Typically, the data for artificial neural networks were divided into two parts: modeling set and testing set. Then the modeling set was further divided into training group and validation group.

However, artificial neural networks were unstable predictors that, with small changes in training data, may result in very different models. To reduce the influences from unstable predictors, k-fold cross-validation method was applied here [19]. The study divided the database into five equal parts. One part was for testing (i.e., testing data), and the other four parts were combined to be modeling. This cross-validation procedure was repeated five times so that we got five data sets with different testing data.

3.2. Architecture of Prediction Model. Back-propagation neural network is the most popular training algorithm with gradient techniques [22]. In previous study [7], back-propagation neural network comprised an input layer (with 5 input variables), a hidden layer (with 10 nodes), and an output layer (with 1 nodes). The ensemble artificial neural networks method was utilized to improve the generalization of the back-propagation neural network [23]: the previous

TABLE 1: Standard parameter set for training.

Parameter	Value
Transfer function of the hidden neurons	Tan sigmoid
Transfer function of the output neurons	Log sigmoid
Training function	Trainscg
Maximal fail	1
Encoding	Real (decimal)
Chromosome length	71
Population size	30
Weight initialization routine	Rand
Initial range	$-1 \sim 1$
Fitness function	Mean square error
Selection operation	Roulette whe5el
Crossover	$BLX - 0.5$
Mutation	Non-uniform
Elitist	2
Stopping criterion	100 iterations

study structured 20 neural networks with different training data and validation data then trained each of them for 15 times with random initial weights and then ensembled all the neural networks with the best performance (Figure 2). In this study, genetic algorithm tried to find the optimization initial points instead of 15 random initial weights for back-propagation neural networks training (Figure 3).

4. The Proposed Genetic Algorithm Model in This Case

During the study, there were many methods, operators, or ideals that tried to reach the optimum. The processes would be presented below and the parameter set was listed in Table 1.

4.1. Modeling Strategy. The modeling strategy was the skeleton about how to optimize the initial weights of the artificial neural network model and the study tried two types. At the beginning of the study (i.e., type 1), the genetic algorithm would evolve the population each iteration with different training data (validation data is also) and then choose 15 the best chromosomes into each artificial neural networks instead of 15 random initial weights (Figure 3(a)). However, neural network is unstable with different results because of a small change in training data. Therefore, the second strategy (i.e., type 2) was presented: the training data were defined for each artificial neural network, and then the genetic algorithm would find the optimum initial weights of ANNs, respectively (Figure 3(b)).

4.2. Initial Population. In genetic algorithms, the binary-code and the real code are the primary schemes to describe a chromosome. But, because the binary-coded scheme is neither necessary nor beneficial [22, 24], and according to

the advantages of intuitiveness, resolution, and facility (i.e., need not to decode) for real code, the study used the real-coded method for describing the chromosomes. There were 30 chromosomes generated in each generation, and each of chromosomes consisted of 60 weights and 11 biases represented by one digit. The range for the initial population will affect the search efficiency, so we tried three levels. At first, the range between -2 and 2 was used, because all of the weights fell in this range after training by back-propagation neural network. But later, the study tried to set the value between -1 and 1 to compare with back-propagation neural network. And last, because the crossover operator could search over the initial range, we tried to narrow the range again between -0.5 and 0.5.

4.3. Evaluation. Each member of the current population was evaluated by fitness function based on the mean square error value to assign the probability of being selected in matting pool. The fitness function here was the back-propagation neural network. The study inserts the solutions into the networks and then calculated the error after the training. The mean square error value represents how the solution is fit for the problem, but it does not mean the solution is suitable for being redrawn in the next generation. The reason is that it is not difficult to find the optimal chromosomes with minimal errors for training data, but it is difficult for validation data. In other words, because the lowest mean square error in training data was the goal of the networks, but the mean square error in validation data used for preventing overfitting of the neural networks should also be considered.

To avoid the above state, the study sets the limitation: no matter how lower the mean square error was, if the error resulted from validation data is higher than the threshold, the chromosome would not be chosen. The threshold was defined as with the optimal solution in last generation, the network would not stop training until the validation error went up, and the error was the threshold of the next generation [25].

4.4. Reproduction. A mating pool of 30 chromosomes was created by Roulette wheel selection operator according to the probability of each chromosome in the current population. The steps of the procedure were as follows: firstly, select the random number between 0 and 1. Secondly, chose the chromosome whose cumulative probability is a little more than the random number into the mating pool. Finally, repeat above steps until 30 new chromosomes are created in the population.

4.5. Crossover. The process was described as below: firstly, randomly select two chromosomes from the matting pool. Secondly, choose four random positions and exchange genes between the first two positions and two last positions. Thirdly, randomly choose the numbers of the interval $[c_{min} - I \cdot 0.5, c_{max} + I \cdot 0.5]$ for four positions separately, where c_{min} is the minimal value between two parents, I is the range between c_{min} and c_{max} [26]. The last two springs were
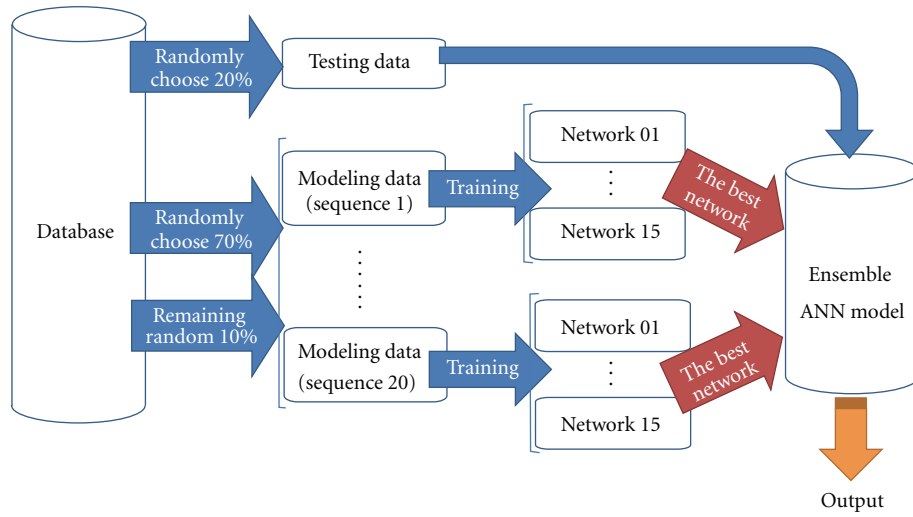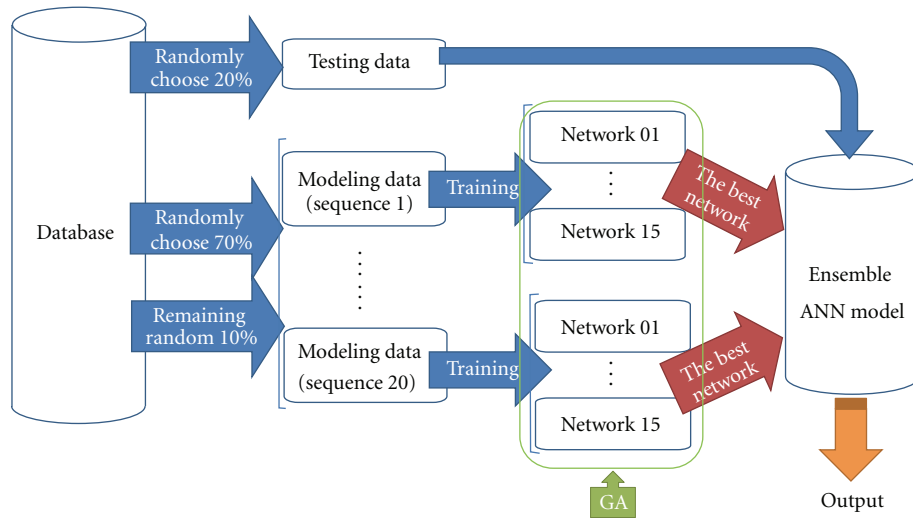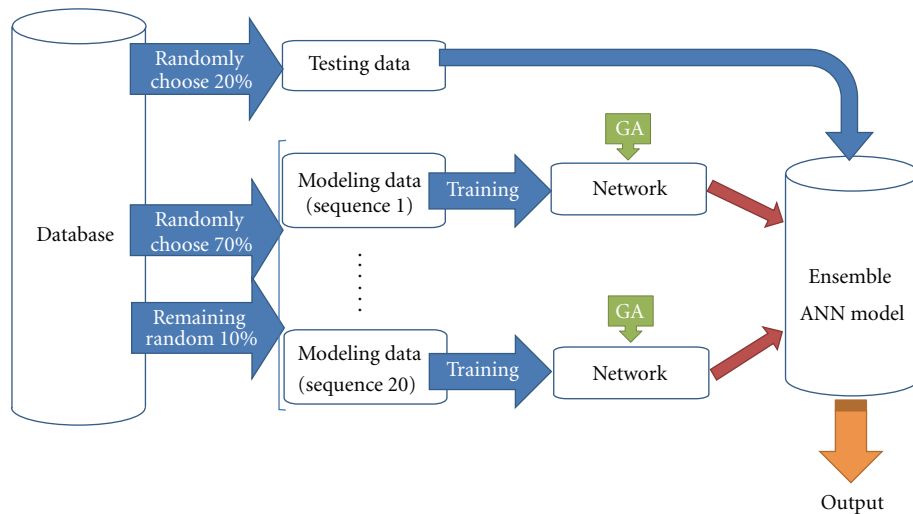
FIGURE 2: The skeleton with initial weights.

(a)

(b)

FIGURE 3: The skeleton with GA-based initial weights: (a) type 1 and (b) type 2.
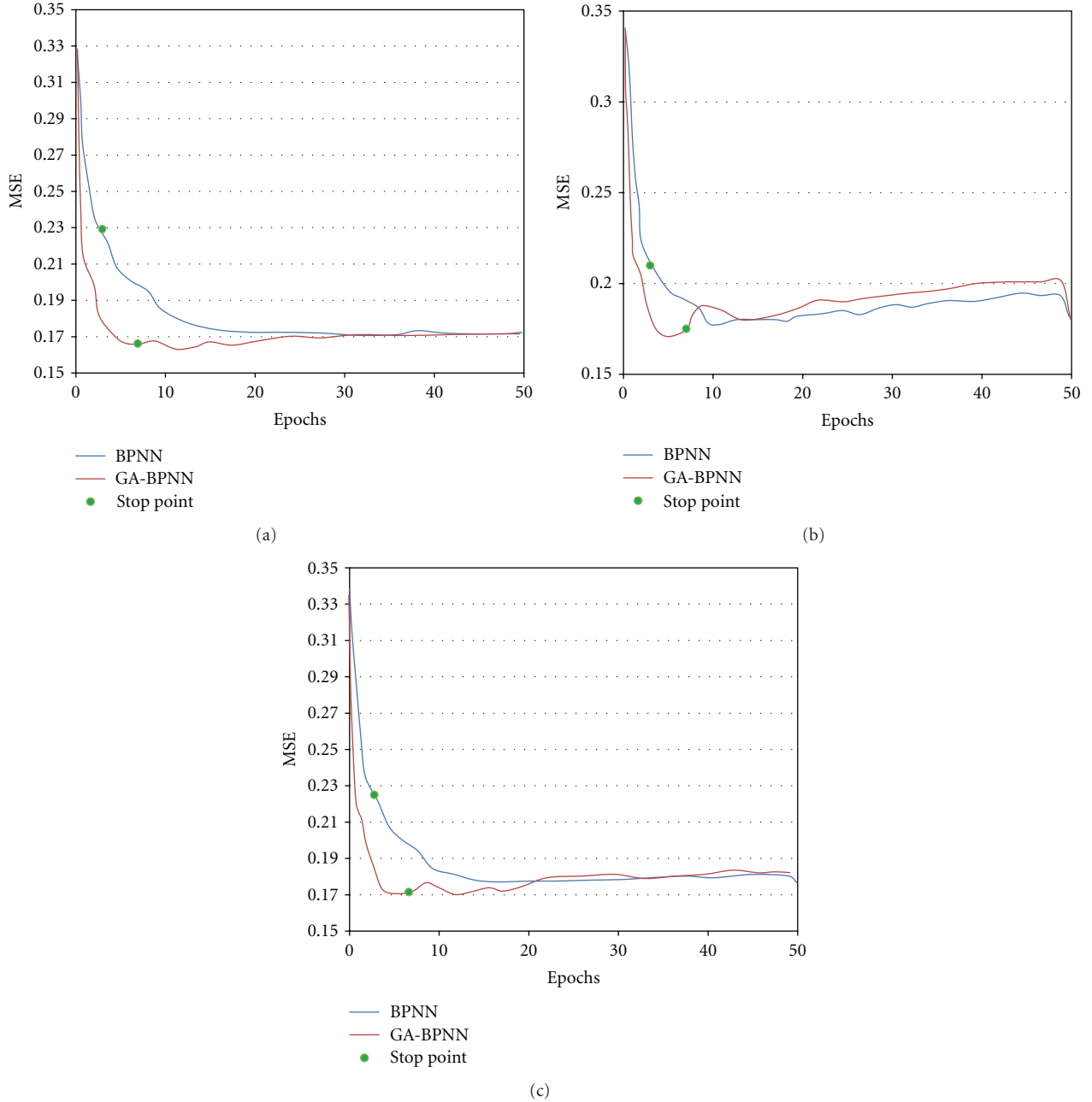
(a)

(b)

(c)

FIGURE 4: Comparison of the MSE between BPNN and GA-based BPNN for different data: (a) training data, (b) testing data, (c) database (including training and testing data).

generated into matting pool. The step was repeated until four-fifth of population was altered.

The features of the operator were that using four crossover points could match uniformly, in other words, the beginning of the string would not always separate from the end of the string. Secondly, chromosomes might include genes that never appear; it was because our paper used the blend crossover method (α value = 0.5) [27].

4.6. Mutation. The mutation operator used in this study was nonuniform mutation [28]. Compared with random muta-

tion, the nonuniform mutation could change the interval for mutating depending on iterations. The genes would be mutated by (1)

$$c_i' = \begin{cases} c_i + \Delta(t, b_i - c_i), & \text{if } \tau = 0 \\ c_i - \Delta(t, c_i - a_i), & \text{if } \tau = 1, \end{cases} \tag{1}$$

where t is current generation, a_i and b_i are the initial ranges of lowest and highest limits, τ is a random number which may have a value of zero or one, and calculate Δ using (2)

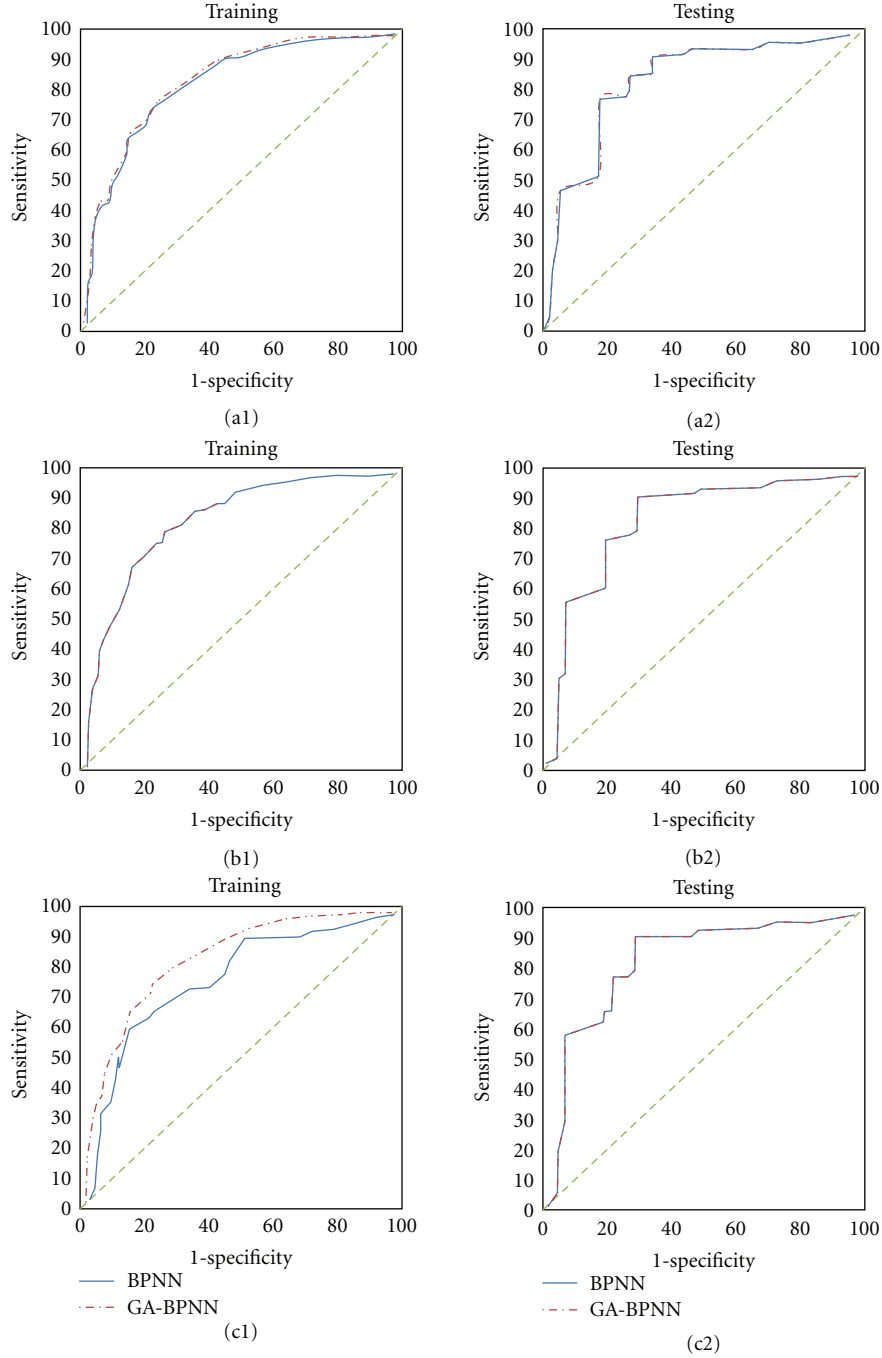$$\Delta(t, y) = y\left(1 - r^{(1-(t/g_{\max}))^b}\right), \tag{2}$$

FIGURE 5: The ROC curves of the original BPNN and the GA-based BPNN for different ranges, (a1) and (a2) −0.5 to 0.5 range for the training and testing data, (b1) and (b2) −1 to 1 range for the training and testing data, (c1) and (c2) −5 to 5 range for the training and testing data.

where r is a random number from the interval $[0,1]$, g_{max} is the maximum number of generations, and b is a parameter which determines the level of dependency on the number of iterations (it is equal to five here) [29].

The feature of the operator was as follows: the operator would make a uniform search in the initial space when t was small and become narrower in later generations.

4.7. Stopping Criterion. The algorithm would terminate after 100 generations, because it had almost converged.

5. Results and Discussion

In this paper, the normal backpropagation algorithm has been used in ANN. Regarding the learning rate chosen 0.01,

the chosen 10 nodes for hidden layer and using 20% dataset for testing, have been reported in our previous study [10]. Moreover, in order to avoid the overfitting, the neural network would stop training when the validation error started to go up (see Figure 4, the point signed with green point). In these figures, we could make sure that the neural networks converged rapidly with pretraining by genetic algorithm.

The paper calculated the area under ROC curve (AUC) with different initial range (Figure 5). The differences between Figure 5 (a1) and (a2) ($AUC_{modeling}$ = 0.858 and $AUC_{testing}$ = 0.802), and Figure 5 (b1) and (b2) ($AUC_{modeling}$ = 0.849 and $AUC_{testing}$ = 0.831) seemed to be similar, but pretty different in Figure 5 (c1) and (c2) ($AUC_{modeling}$ = 0.778 and $AUC_{testing}$ = 0.849). It means that if the initial range of GA parameter is smaller it is able to get better testing result.

To find the optimal initial weights however was a difficult task. Firstly, it had misgivings about not only the over-fitting from backpropagation in neural networks but also the tendency toward minimal mean square error in training data only without consideration of validation data in genetic algorithms. Secondly, the searching space of genetic algorithm might be limited by the initial range of the initial weights. Last, the advantages of using genetic algorithms compared with our previous study [10] should be based on the performance of the neural networks on the testing datasets, instead of the minimal square error only in the modeling datasets.

Another consideration of minimal improvement of genetic algorithms in this study was the ratio between the number of chromosomes in a generation (i.e., population size) and the length of a chromosome was small. It might be the reason why the genetic algorithm cannot search extensively to reach the optimum.

6. Conclusions and Future Work

The study results showed that the genetic algorithm obtained a good result of AUC of 0.858 ± 0.00493 on modeling data and 0.802 ± 0.03318 on testing data for small range of initial parameter. They were slightly better than the results of our previous study (0.868 ± 0.00387 and 0.796 ± 0.02559, resp.). Thus, the preliminary study for only using simple GA has been proved to be effective for improving the accuracy of artificial neural networks. However, the genetic algorithm should be further modified to improve the performance because of the data of our hip fracture cases were highly non-linear and complex. Our future work is to try different ways of coding schemes to increase the efficiency of genetic operations, change the ratio between chromosome length and population size to extend the searching space, and investigate the effects of different initial ranges of initial weights on the network performance.

The works that should be done to improve the algorithms in the future are as follows: firstly, a better fitness function is designed to prevent the overfitting in genetic algorithm. Secondly, the criterion of stopping the algorithm can try some other methods, for example, to stop when the best chromosome does not change for a certain number of generations or the chromosomes with similar minimal mean square errors reach a certain number. Finally, other classification methods, such as neurofuzzy algorithms [30], support vector machines [9], and particle swarm optimizations [31] would be a good candidate for improving this prediction accuracy.

References

[1] O. Johnell and J. A. Kanis, "An estimate of the worldwide prevalence and disability associated with osteoporotic fractures," *Osteoporosis International*, vol. 17, no. 12, pp. 1726–1733, 2006.

[2] C. Cooper, E. J. Atkinson, S. J. Jacobsen, W. M. O'Fallon, and L. J. Melton, "Population-based study of survival after osteoporotic fractures," *American Journal of Epidemiology*, vol. 137, no. 9, pp. 1001–1005, 1993.

[3] C. L. Leibson, A. N. A. Tosteson, S. E. Gabriel, J. E. Ransom, and L. J. Melton, "Mortality, disability, and nursing home use for persons with and without hip fracture: a population-based study," *Journal of the American Geriatrics Society*, vol. 50, no. 10, pp. 1644–1650, 2002.

[4] J. Magaziner, E. Lydick, W. Hawkes et al., "Excess mortality attributable to hip fracture in White women aged 70 years and older," *American Journal of Public Health*, vol. 87, no. 10, pp. 1630–1636, 1997.

[5] G. S. Keene, M. J. Parker, and G. A. Pryor, "Mortality and morbidity after hip fractures," *British Medical Journal*, vol. 307, no. 6914, pp. 1248–1260, 1993.

[6] A. M. Jette, B. A. Harris, P. D. Cleary, and E. W. Campion, "Functional recovery after hip fracture," *Archives of Physical Medicine and Rehabilitation*, vol. 68, no. 10, pp. 735–740, 1987.

[7] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *IEEE Transactions on Information Theory*, vol. 44, no. 5, pp. 1926–1940, 1998.

[8] C. Campbell, "Kernel methods: a survey of current techniques," *Neurocomputing*, vol. 48, pp. 63–84, 2002.

[9] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.

[10] Y. C. Chou, *Predicting the risk of hip bone fracture for elders in Taiwan by ensemble back-propagation neural networks*, BS thesis, Department of Mechanical Engineering, University of Yuan Ze University, 2009.

[11] G. Li, H. Alnuweiri, Y. Wu, and H. Li, "Acceleration of back propagation through initial weight pre-training with delta rule," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 580–585, April 1993.

[12] Y. Lee, S. H. Oh, and M. W. Kim, "The effect of initial weights on premature saturation in back-propagation learning," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '91)*, pp. 765–770, July 1991.

[13] A. J. Al-Shareef and M. F. Abbod, "Neural networks initial weights optimisation," in *Proceedings of the 12th International Conference on Modelling and Simulation (UKSim '10)*, pp. 57–61, March 2010.

[14] D. Venkatesan, K. Kannan, and R. Saravanan, "A genetic algorithm-based artificial neural network model for the optimization of machining processes," *Neural Computing and Applications*, vol. 18, no. 2, pp. 135–140, 2009.

[15] R. S. Sexton and J. N. D. Gupta, "Comparative evaluation of genetic algorithm and backpropagation for training neural networks," *Information Sciences*, vol. 129, no. 1–4, pp. 45–59, 2000.

[16] D. J. Montana and L. Davis, "Training feedforward neural networks using genetic algorithms," in *Proceedings of the Joint Conference on Artificial Intelligence*, vol. 1, pp. 762–767, 1989.

[17] B. G. Eriksson and V. Sundh, "Prediction of seven-year survival by artificial neural network and logistic regression: a comparison of results from medical and social data among 70-year-olds in Göteborg, Sweden," Eriksson & Sundh, 2010.

[18] D. J. Montana, *Neural Network Weight Selection Using Genetic Algorithms*, Bolt Beranek and Newman, 1994.

[19] L. Davis, Ed., *Genetic Algorithms and Simulated Annealing*, Pitman, London, UK, 1991.

[20] D. E. Goldberg, *Genetic Algorithms in Machine Learning*, Addison-Wesley, 1988.

[21] T. Y. Lan, S. M. Hou, C. Y. Chen et al., "Risk factors for hip fracture in older adults: a case-control study in Taiwan," *Osteoporosis International*, vol. 21, no. 5, pp. 773–784, 2010.

[22] L. M. Salchenberger, E. M. Cinar, and N. A. Lash, "Neural networks: a new tool for predicting thrift failures," *Decision Sciences*, vol. 23, pp. 899–916, 1992.

[23] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[24] P. Cunningham, J. Carney, and S. Jacob, "Stability problems with artificial neural networks and the ensemble solution," *Artificial Intelligence in Medicine*, vol. 20, no. 3, pp. 217–225, 2000.

[25] W. S. Sarle, "Stopping training and other remedies for overfitting," in *Proceeding of the 27th Symposium on the Inter Face*, 1995.

[26] F. Herrera, M. Lozano, and J. L. Verdegay, "Tackling real-coded genetic algorithms: operators and tools for behavioural analysis," *Artificial Intelligence Review*, vol. 12, no. 4, pp. 265–319, 1998.

[27] L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," *Foundation of Genetic Algorithms*, pp. 187–202, 1993.

[28] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, 1992.

[29] L. Davis, *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, 1991.

[30] J. W. F. Catto, M. F. Abbod, D. A. Linkens, and F. C. Hamdy, "Neuro-fuzzy modeling: an accurate and interpretable method for predicting bladder cancer progression," *Journal of Urology*, vol. 175, no. 2, pp. 474–479, 2006.

[31] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intelligence*, vol. 1, pp. 33–37, 2007.