Available online at www.sciencedirect.com

ScienceDirect

Neurocomputing 71 (2008) 1604–1618

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Evolutionary ensemble of diverse artificial neural networks using speciation

Kyung-Joong Kim*, Sung-Bae Cho

Department of Computer Science, Yonsei University,134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, South Korea

Received 1 May 2006; received in revised form 5 April 2007; accepted 17 April 2007
Communicated by T. Heskes
Available online 3 May 2007

Abstract

Recently, many researchers have designed neural network architectures with evolutionary algorithms but most of them have used only the fittest solution of the last generation. To better exploit information, an ensemble of individuals is a more promising choice because information that is derived from combining a set of classifiers might produce higher accuracy than merely using the information from the best classifier among them. One of the major factors for optimum accuracy is the diversity of the classifier set. In this paper, we present a method of generating diverse evolutionary neural networks through fitness sharing and then combining these networks by the behavior knowledge space method. Fitness sharing that shares resources if the distance between the individuals is smaller than the sharing radius is a representative speciation method, which produces diverse results than standard evolutionary algorithms that converge to only one solution. Especially, the proposed method calculates the distance between the individuals using average output, Pearson correlation and modified Kullback–Leibler entropy to enhance fitness sharing performance. In experiments with Australian credit card assessment, breast cancer, and diabetes in the UCI database, the proposed method performed better than not only the non-speciation method but also better than previously published methods.
© 2007 Elsevier B.V. All rights reserved.

Keywords: Evolutionary neural network; Ensemble; Fitness sharing; Kullback–Leibler entropy; Majority voting; UCI benchmark dataset

1. Introduction

Combining multiple classifiers performs better than a single classifier but suffers from the difficulty of construction process (diversity of members, parameter tuning, and combination methods). It is well known that combining identical classifiers has no gain and diversity among members is one of the biggest issues in forming successful ensembles. Because of the complexity of ensembles, there are a number of parameters and the way to determine them is a critical problem. Finally, the selection of combination methods can give much impact on the performance of the ensemble classification. In this paper, speciation-based evolutionary neural ensemble method is proposed to deal with the forming ensemble automatically.

Recently, designing artificial neural networks (ANN's) with evolutionary algorithms has emerged as a preferred alternative to the common practice of selecting the apparent best network [42]. Because evolutionary algorithms search from not only a single point but a large population of points, many researchers have actively exploited the combining of multiple ANN's which have evolved in the last generation. However, these ANN's tend to be too similar to each other because in each case, the individual with the highest fitness has prevailed even after certain generations. This phenomenon is known as genetic drift [35]. Combining the outputs of several classifiers is useful only if they are complementary. The combination of similar classifiers produces no gain.

According to Goldberg, a niche is an organism's job or role in an environment and a species is a class of organisms with common characteristics [12]. In a solution space, there are stable sub-populations of strings (species) serving

*Corresponding author.
E-mail address: kjkim@cs.yonsei.ac.kr (K.-J. Kim).

0925-2312/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2007.04.008

different sub-domains of a function (niche). A simple genetic algorithm (after enough generations) will distribute most individuals around the highest peak. The inducement of niche and species formations can help identify peaks in other regions of the space. Sharing is the most important concept to induce niche formation in the genetic algorithm. Individuals share available resources in an overpopulated habitat.

A practical scheme detailed by Goldberg and Richardson uses the sharing metaphor to induce niche and species formations [13]. One of the major problems with this sharing is deciding which individuals should share resources and the degree of sharing. A practical way to deal with this problem might be to use a sharing radius and sharing function. Until recently, fitness sharing has been widely used among many extensions of genetic algorithms for searching multiple optima. A sharing function and a sharing radius of the algorithm involve a distance metric that measures the difference between two individuals.

This paper presents a method of constructing multiple neural networks. Our method uses genetic algorithms with fitness sharing to generate a population of ANN's that are accurate and diverse. Yao and Liu have dealt with this issue [24,44,45], but there is still room to devise more appropriate distance measures for fitness sharing, especially in the evolutionary neural networks. The average output, Pearson correlation and modified Kullback–Leibler (KL) entropy methods using entropy theory measure the difference between two ANN's [2].

Fig. 1 shows the basic idea of the proposed method. Speciation in genetic algorithms creates different species, each embodying a sub-solution, which leads to the creation of diverse solutions as well as the best one. This paper uses the fitness sharing technique that is representative of speciation methods for evolutionary neural networks. To show the usefulness of the proposed method, it was necessary to conduct extensive experiments with benchmark problems, such as UCI datasets.

The rest of this paper is organized as follows. Section 2 describes the relevant works on evolutionary artificial neural networks. Section 3 applies speciation to the evolution process, and presents the distance measures with various combination methods. Section 4 describes the experimental results and analysis.

2. Related works

Artificial neural networks have been widely used in many applications, and most neural network applications use the feed-forward model with the backpropagation algorithm (BP). Variants of the BP are sometimes used, but most of these algorithms retain the fixed neural network structure. The algorithms train only the weights on the same network structure. The problem of finding the optimal structure of a neural network has yet to be solved.

Early research on the automatic design of neural networks used various constructive and pruning algorithms. A constructive algorithm adds new nodes and connection information incrementally to the network [11,19], whereas a pruning algorithm removes unnecessary nodes and connection information gradually [33]. Unfortunately, the two methods are subject to remain stuck in local minima; it is difficult to find an optimal neural network structure because they search only the restricted solution space determined by the given environment. An evolutionary algorithm (EA) is a promising way of overcoming this shortcoming. An EA is a general-purpose search method which is insensitive to the initial configurations. Evolutionary artificial neural network development aims at designing weights, topologies, and learning rules automatically by using evolutionary algorithms over a number of generations.

E-Net is a distributed evolutionary learning method that evolves neural-network-based pattern recognition systems with limited human interaction. It evolves network topologies and trains weights to form accurate recognition

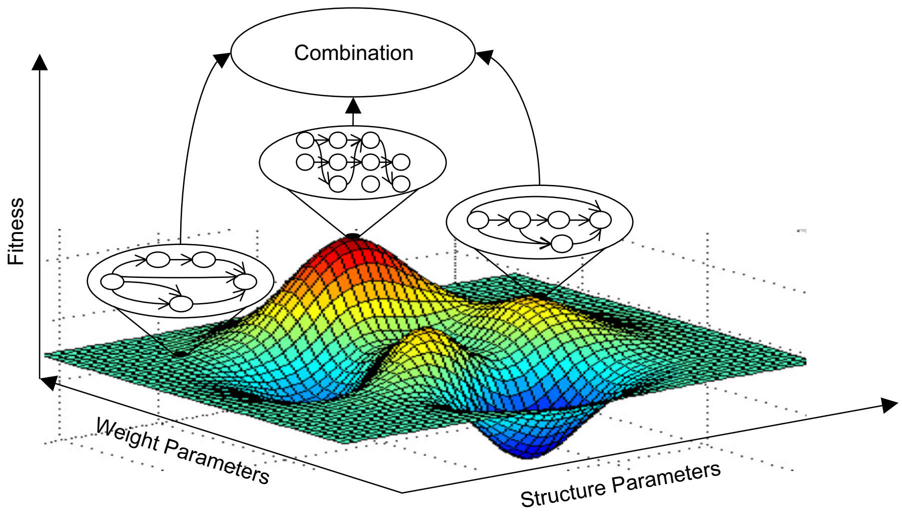

Fig. 1. The proposed ensemble model of speciated neural networks.

systems using a computationally efficient process that gradually extends primitive network topologies to form increasingly discriminating structures [39]. Abbass has proposed an evolutionary artificial neural network based on the pareto-differential evolution algorithm for the prediction of breast cancer [1]. Perez et al. have proposed COVNET, a new cooperative evolutionary model for evolving artificial neural networks. This model adopts the idea of co-evolving sub-networks that must cooperate to form a solution for a specific problem, instead of evolving complete networks [29].

The efforts on the ensemble approach can be categorized into two types: generating individual artificial neural networks and combining individual predictions [46]. Boosting and bagging are popular methods for generating individual networks. Yao et al. presented evolutionary ensembles with negative correlation learning (EENCL) [22] in which a fitness sharing scheme based on mutual information was introduced. The mutual information between two variables, output F_i of network i and output F_j of network j is given by

$$I(F_i;F_j) = H(F_i) + H(F_j)\ H(F_i,F_j), \tag{1}$$

where $H(F_i)$ is the entropy of F_i, $H(F_j)$ is the entropy of F_j and $H(F_i,F_j)$ is the joint differential entropy of F_i and F_j.

Brown et al. provides a survey about the theoretical analysis of successful ensemble methods and the way to improve the diversity of ensembles [6]. Bryll et al. proposed attribute bagging (AB), a technique for improving the accuracy and stability of classifier ensembles induced using random subsets of features [7]. Windeatt compared the ability of several pair-wise diversity measures to predict generalization error of multiple classifier system [40]. Table 1 is a summary of the combination methods of multiple classifiers.

3. Evolution of speciated neural networks

Let $A = \{e_1, e_2, \ldots, e_k\}$ be a set of classifiers and $B = \{C_1, C_2, \ldots, C_m\}$ be a set of class labels. Each classifier gets a feature vector $x \in \Re^p$(p is the dimension of feature vector) as its input and assigns a class label. In many cases, the classifier output is a m-dimensional vector:

$$[e_{i1}(x), e_{i2}(x), \ldots, e_{im}(x)]^{\mathrm{T}}, \tag{2}$$

$$e_i(x) = \arg\max_t e_{it}(x), \tag{3}$$

$e_j(x) = C_i$ means classifier e_j assigns class C_i to the input x. m is the number of output nodes. The set of output signals of the output layer of the network is $O = \{o_1, o_2, \ldots, o_m\}$ and o_i, the one with the largest value is chosen. $F(e(x)) = C_i$ means the ensemble of classifiers e classifies x into class C_i.

Suppose that we have a training set

$$D = \{(x_1, d_1), \ldots, (x_n, d_n)\}, \tag{4}$$

where $x \in R^p$, d is a scalar, and n is the size of the training set. The assumption that the output d is a scalar has been made merely to simplify the exposition of ideas without loss of generality. This section considers estimating d by forming an ensemble, whose output is a simple averaging of outputs of a set of neural networks.

The error function E_i for the network e_i in the proposed method is defined by

$$\begin{aligned} E_i &= \frac{1}{n}\sum_{x=1}^{n} E_i(x) \\ &= \frac{1}{n}\sum_{x=1}^{n} |e_i(x_j) - d_j|, \end{aligned} \tag{5}$$

where $E_i(x)$ is the value of the error function of network e_i at the presentation of the training pattern x. Fitness function f_i for network e_i in the proposed method is defined by

$$f_i = 1.0 - E_i. \tag{6}$$

The fitness f_{si} is the rescaled one using fitness sharing method. After finishing evolution, the representative networks from each cluster are combined.

Fig. 2 shows the overall procedure of evolving neural networks. The method sets each ANN with random initial weights for the fully connected links, and conducts partial training with a BP algorithm in a limited number of epochs to adjust the weights of given architecture of ANN. The fitness of the ANN is calculated with a recognition rate for the validation data. Fitness sharing using the distance measures rescales its original fitness for diversity. Once the fitness is calculated, the genetic algorithm selects the best 50% individuals upon which to apply genetic operators. The genetic operators, crossover and mutation, are applied to those selected individuals. After applying the genetic

Table 1
Summary of the combination methods of multiple classifiers

Authors	Combination methods	Contributions
Min et al. [25]	Multiple decision templates	Localizing fusion models with clustering algorithm
Rahman et al. [32]	Vertical/hybrid/horizontal combinations	A survey of the classification combination methods
Alexandre et al. [3]	Arithmetic and geometric means	Comparative study of the performance of arithmetic and geometric mean combinations
Kim et al. [16]	Fuzzy integral	A content mining system based on fuzzy integral of multiple classifiers
Atincay et al. [5]	Dempster–Shafer theory	A multiple classifier approach as an alternative solution to the closed-set text-independent speaker identification

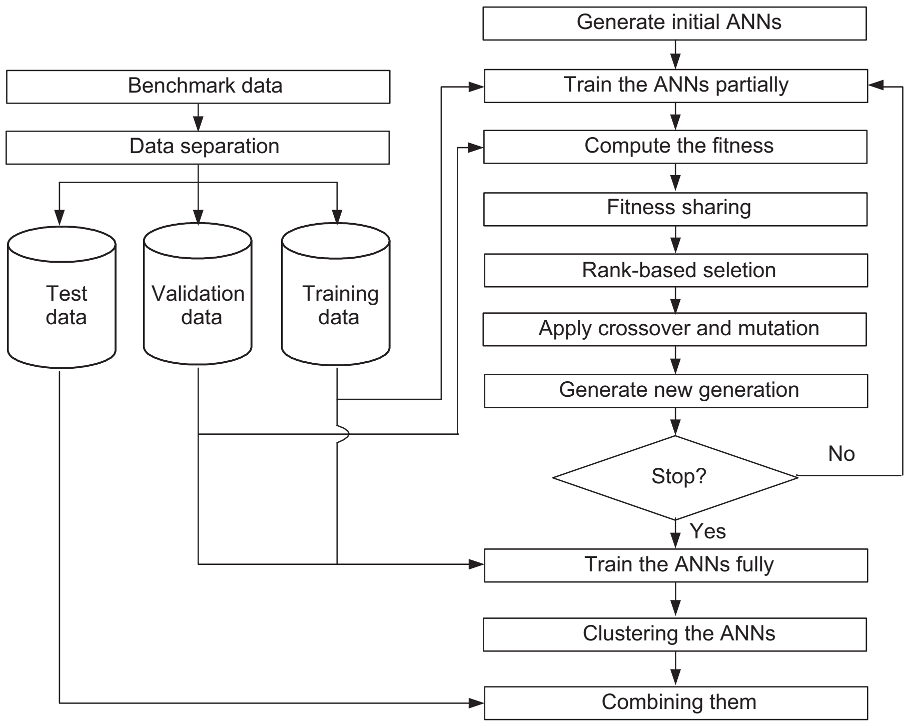

Fig. 2. Overall procedure for evolving neural networks.

operators, the new set of individuals forms a new population. It finishes when the stop criterion is satisfied (if the number of generation exceeds the maximum number of generation or the fitness is 1.0). The final step is the full training of the ANN's in the last generation with the error BP algorithm. Using clustering, representative individuals are selected and combined with various methods such as voting, weighted voting and etc. Fig. 3 shows the pseudo code for the proposed method.

3.1. Representation

In evolutionary algorithms, it is very important to determine the representation of an individual. There are several methods to encode an ANN. These include binary, tree, linked list, and matrix representation. We used a matrix representation to encode the ANN since it is straightforward to implement and easy to apply the genetic operators [37]. When N is the total number of nodes in the ANN (including the input, hidden, and output nodes), the matrix is $N \times N$ whose entries consist of connection links and the corresponding weights. In this model, each ANN uses only forward links [38]. In the matrix, the upper right triangle (see Fig. 4) has connection link information where '1' means that there is a connection link and '0' means that there is no connection link. The lower left triangle describes the weight values corresponding to the connection link information. Fig. 4 shows an encoding example of an ANN that has one input node, three hidden nodes, and one output node. The number of hidden nodes can vary within the maximum number of hidden nodes in the course of the GA operations.

3.2. Crossover

The crossover operator exchanges the architectures of two ANN's in the population to search the ANN's from various architectures [28]. In a population of ANN's, the crossover operator selects two distinct ANN's randomly and chooses one hidden node from each ANN. These two nodes should be in the same entry of each ANN matrix encoding the ANN to exchange the architectures. The two ANN's exchange the connection links and the corresponding weight information of the nodes.

3.3. Mutation

The mutation operator changes a connection link and the corresponding weight of a randomly selected ANN from the population. It performs one of two operations: addition of a new connection and deletion of an existing connection. The mutation operator selects an ANN from the population of ANN's randomly and chooses one connection link from it. If the connection link does not exist and the connection entry of the ANN matrix is '0', a new connection link is created. It adds the new connection link to the ANN with random weights. Otherwise, if the connection link already exists, it removes the connection link and weight information.

```
Run(){
  // Initialization
  For all individuals {
    1. Upper right triangle of matrix is initialized to 1 // Full connection
    2. Bottom left triangle of matrix is initialized with random real value between 0 and 1
      // Random weights}
  // Partial training
  For all individuals {Train(individual, training data, the number of epochs for partial training)}

  // Evolution
  For maximum generation {
   1. For all individuals { fitness[individual]=Recognition(individual, validation data) }
   2. For all individuals { Distance measuring }
   3. For all individuals
      { sharing[individual]= $\sum_{j}^{population\_size}$ 1-distance (individual, $j$)/(sharing radius) }
   4. fitness[individual]=fitness[individual]/sharing[individual]
   5. Sort population by fitness
   6. Select half of individuals based on fitness
   7. Applying genetic operators (crossover, mutation)
   8. Partial training
  }
  // Full training
  For all individuals
  {Train (individual, training + validation data, the number of epochs for full training)}
  // Combining
  1. Single linkage clustering 2. Select representatives from clusters 3. Combining them
}

// BP
Train (individual, data, epoch){
  For epoch { // batch mode
    1. Compute output of individual for data
    2. Compute delta
    3. Update weight
}}
```

Fig. 3. A pseudo code for the procedure.

3.4. The fitness sharing technique

There are several ways to simulate speciation. In this paper, the fitness sharing technique is used [12]. Fitness sharing decreases the fitness of individuals in a densely populated area and shares the fitness with other ANN's. Therefore, it helps the genetic algorithm search a broad solution space and it generates more diverse ANN's. With fitness sharing, the genetic algorithm finds more diverse solutions.

Given that f_i is the fitness of an individual and $sh(d_{ij})$ is a sharing function, the shared fitness f_{si} is computed as

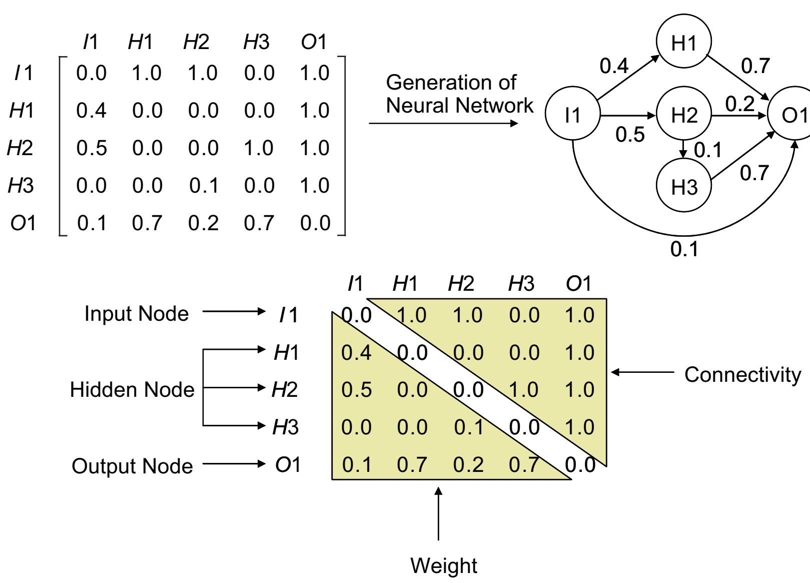

Fig. 4. An example of neural network representation.

follows:

$$fs_i = \frac{f_i}{\sum_{j=1}^{population\ size} sh(d_{ij})}. \tag{7}$$

The sharing function $sh(d_{ij})$ is computed using the distance value d_{ij} which means the difference of individuals i and j as follows:

$$sh(d_{ij}) = \begin{cases} 1 - \dfrac{d_{ij}}{\sigma_s}, & 0 \leqslant d_{ij} < \sigma_s, \\ 0, & d_{ij} \geqslant \sigma_s. \end{cases} \tag{8}$$

Here, σ_s means the sharing radius. If the difference of the individuals is larger than σ_s, they do not share the fitness. Only the individuals who have smaller distance values than σ_s can share the fitness.

The sharing radius is determined by a threshold value derived from the following formula:

$$\sigma = \frac{1}{2P^2} \sum_{i=1}^{P} \sum_{j=1}^{P} d_{ij}, \tag{9}$$

where P is the population size, and d_{ij} is the distance between the ith and jth ANN's.

Fig. 5 presents an example of fitness sharing. The fitness of individuals on the second peak decreases because the number of individuals in the area is dense.

3.5. Distance measures for fitness sharing

Average output, Pearson correlation and Kullback–Leibler entropy are simple methods to measure the distance between two neural networks [18]. When n is the number of data and m is the number of output nodes, the average output of an ANN (represented as a) is as follows:

$$\overline{o}_{aj} = \left(\sum_{t=1}^{n} e_{aj}(x_t) \right) \Big/ n, \tag{10}$$

$$\overline{o}_a = (\overline{o}_{a1}, \overline{o}_{a2}, \ldots, \overline{o}_{am}), \tag{11}$$

where $e_{aj}(x_t)$ is the output of the jth output node for the tth input data. The distance between the two ANN's is the Euclidean distance of their average outputs. The similarity between ANN a and b can be calculated as follows:

$$d_{ab} = \sqrt{\sum_{j=1}^{m} (\overline{o}_{aj} - \overline{o}_{bj})^2}. \tag{12}$$

Pearson correlation can reveal the relationship between two variables. The similarity between ANN a and b can be calculated as follows:

$$\overline{o}_a = \left(\sum_{j=1}^{m} \overline{o}_{aj} \right) \Big/ m, \tag{13}$$

$$d_{ab} = \frac{\sum_{j=1}^{m} (\overline{o}_{aj} - \overline{o}_a) \times (\overline{o}_{bj} - \overline{o}_b)}{\sqrt{\sum_{j=1}^{m} (\overline{o}_{aj} - \overline{o}_a)^2 \times \sum_{j=1}^{m} (\overline{o}_{bj} - \overline{o}_b)^2}}. \tag{14}$$

Let one discrete distribution have probability function p and the other discrete distribution have probability function q. Then the relative entropy of p_k (k is a random variable and p_k represents the probability of specific values of k) with respect to q_k, also known as the Kullback–Leibler distance, is defined by

$$d = \sum_k p_k \log\left(\frac{p_k}{q_k}\right). \tag{15}$$

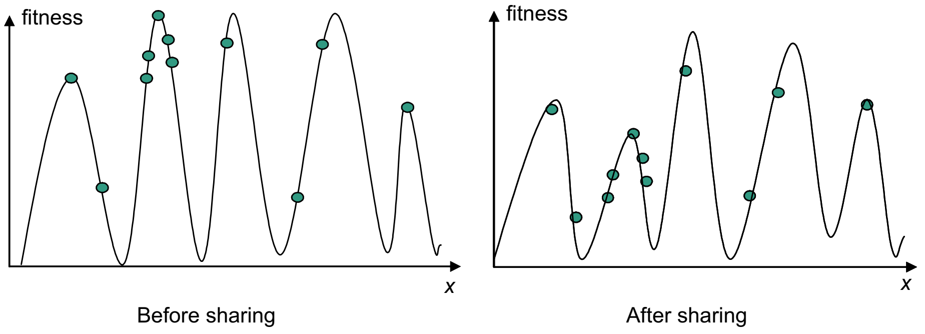

Fig. 5. An example of fitness sharing.

Although relative entropy does not satisfy the triangle inequality and is therefore not a true metric, it satisfies many important mathematical properties [9]. For example, it is a convex function of p_k, always non-negative, and equal to zero only if $p_k = q_k$. Relative entropy is a very important concept in quantum information theory, as well as statistical mechanics [31]. However, relative entropy is not a true distance because it is not symmetric, i.e., $D(p,q) \neq D(q,p)$. To remedy this problem, the modified Kullback–Leibler entropy measure is used

$$D(p,q) = \frac{1}{2}\sum_k \left(p_k \log \frac{p_k}{q_k} + q_k \log \frac{q_k}{p_k} \right). \tag{16}$$

Modified Kullback–Leibler entropy measures the difference of two ANN's. Let p and q be the output probability distributions of two ANN's. p and q represent output probability distributions given input evidences (a vector of attribute values of a specific sample). The ith output node provides the likelihood of a sample with respect to the ith class. When the estimation is accurate, the network outputs can be treated as probabilities [20,34]. The total KL distance between the two neural networks is the sum of the KL values for all samples and output nodes.

Then, the similarity of the two ANN's (each neural network is labeled as a and b) is calculated as follows:

$$d_{ab} = \frac{1}{2}\sum_{t=1}^{n}\sum_{j=1}^{m}\left(e_{aj}(x_t) \log \frac{e_{aj}(x_t)}{e_{bj}(x_t)} + e_{bj}(x_t) \log \frac{e_{bj}(x_t)}{e_{aj}(x_t)} \right). \tag{17}$$

The two ANN's are more similar as the symmetric relative entropy decreases.

3.6. Combination of multiple neural networks

Single linkage clustering provides the representative neural networks for each species. In this method, the distance between the two clusters is as follows:

$$d(cluster_1, cluster_2) = \min_{i \in cluster_1,\ j \in cluster_2} d_{ij}. \tag{18}$$

The distance is calculated with the same method used in the fitness sharing. A dendrogram is a branching diagram that can be used to show the interconnections among neural networks (Fig. 6). The Y-axis means the dissimilarity and the X-axis means the index number of 20 neural networks. In the figure, the number of clusters is four if the threshold value is set as 40.

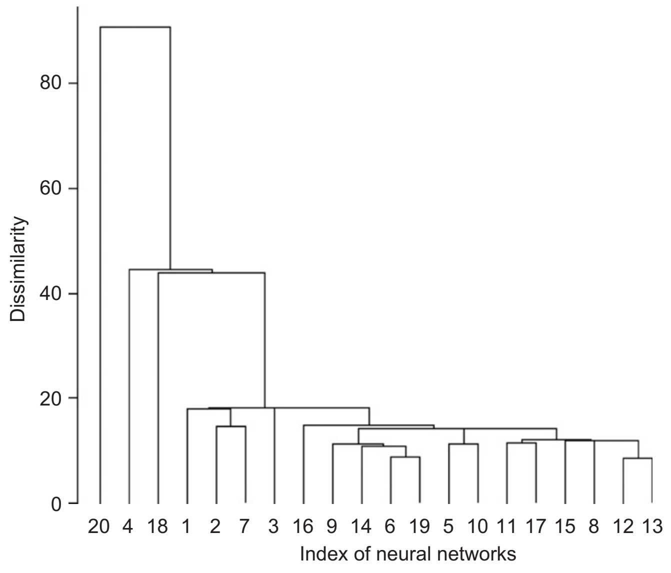

Fig. 6. An example of a dendrogram.

Agglomerative hierarchical clustering models form an initial partition of P clusters (each neural network is a cluster) and proceed to reduce the number of clusters one at a time until all P neural networks are in one cluster. In the first stage, the P-1 clusters are formed by selecting the two most similar objects (a single neural network and clusters of neural networks). In the second stage, the P-2 clusters are formed in a similar manner, and so on.

Mojena provided a way to determine the number of groups for hierarchical clustering procedures [27]. Mojena's cut-off value was $\bar{h} + \alpha s_h$ where $\bar{h}$ is the average of the dendrogram height for all P-1 clusters, s_h is the unbiased standard deviation of the heights and α is a specified constant. Mojena initially suggested that α should be specified in the range of 2.75–3.50. Milligan and Cooper concluded that the best overall performance of Mojena's rule occurs when the value of α is 1.25 [26].

4. Experimental results

4.1. Experimental environments

To show the effectiveness of the proposed method, experiments were conducted with several benchmark problems: Australian credit approval, breast cancer, and diabetes from the UCI machine learning dataset. The Australian credit approval dataset was a 2-class problem with 690 examples. Each example had 14 attributes. The training, validation and test datasets consisted of 346, 172 and 172 examples, respectively. The breast cancer dataset was a 2-class problem with 699 examples, each of which had 9 attributes and 1 class. The training, validation and test datasets contained 349, 175 and 175 examples, respectively. The diabetes dataset had 768 examples. Five hundred examples out of 768 were class 0 and the others were class 1. Each example had 9 attributes and 1 class. The training, validation and test datasets contained 384, 192, and 192 examples, respectively.

Because we want to increase the generalization capability of the model, relatively large validation data (25%) are used. Following 4-fold split, 50% training, 25% validation and 25% test data are used. In the Australian data, the missing values were replaced by the mode of the attribute (categorical) and mean of the attribute (continuous). In the diabetes data, there were no missing values. In the breast cancer data, the missing values were replaced with '−1.' After that, the values of the attributes were normalized between 0 and 1.

As a realistic dataset, colon dataset consists of 62 samples of colon epithelial cells taken from colon-cancer patients. Each sample is from one person and contains 2000 gene expression levels. Although original data consist of 6000 gene expression levels, 4000 out of 6000 were removed based on the confidence in the measured expression levels. Forty of 62 samples are colon cancer samples and the remaining are normal samples. Each sample was taken from tumors and normal healthy parts of the colons of the same patients and measured using high-density oligonucleotide arrays. Leave-one-out cross-validation is used (available at http://www.sph.uth.tmc.edu:8052/hgc/default.asp).

The population size was 20 and the maximum generation number was 200. The crossover rate was 0.3 and the mutation rate was 0.1. The selection ratio was 0.5. Each ANN was a feed-forward ANN trained by BP with a learning rate of 0.1. The threshold value of the sigmoid function was 0.8. The training data were presented 50 times (determined empirically) in partial training and 1000 times in full training. Experimental results were the average of 10 runs. Batch learning for BP was used. The number of input nodes was the same as that of the features in the dataset. The number of maximum hidden nodes was 5, and the number of output nodes was 2. The number of clusters was decided by the performance for the validation data. Four parameters (population size, crossover rate, mutation rate, and selection ratio) dealt with the genetic algorithms and four parameters (learning rate, threshold value, the number of hidden nodes, and epochs for partial learning) dealt with the BP algorithms.

The number of output nodes was the same with the number of classes. Each output node produced a confidence value for each class. Among the values of nodes, a class label with the maximum one was chosen for the sample. The error was calculated based on the results of classification on the test set. The weights of the neural network were trained by using the BP algorithm in partial and full training. The difference between partial and full training was the number of epochs for the learning.

4.2. A comparison of the combination methods

- To combine the speciated neural networks we adopt behavior knowledge space (BKS) method of the "multinomial" rule. This fusion method is well-known to provide good performance if large and representative data set is available. Methods for fusing multiple classifiers can be divided into three types: abstract level, rank level and measurement level. In abstract-level outputs, the behavior knowledge space became very popular. In this BKS method, every possible combination of abstract-level classifier outputs is regarded as a cell in a look-up table. The BKS table is from a training set. Each cell contains the number of samples characterized by a particular value of class labels and the most dominated class is chosen for the cell. In the method, a term "cell" is used to represent a space for storing behaviors of ANN. BKS is a set of cells, where the M^K cells are required to store the necessary information of the K classifiers with the M classes. $BKS(e_1(x),\ldots,e_K(x))$ is a cell with index $(e_1(x),\ldots,e_K(x))$.

BKS	a K-dimensional behavior-knowledge space
$BKS(e_1(x),\ldots,e_K(x))$	a unit of BKS, where the 1st classifier gives its decision as $e_1(x),\ldots,$ and the Kth classifier gives its decision as $e_K(x)$
$n_{e_1(x)\ldots e_K(x)}(m)$	the total number of incoming samples belonging to class m in $BKS(e_1(x),\ldots,e_K(x))$
$T_{e_1(x)\ldots e_K(x)}$	the total number of incoming samples in $BKS(e_1(x),\ldots,e_K(x))$ $\sum_{m=1}^{M} n_{e_1(x),\ldots,e_K(x)}(m)$
$R_{e_1(x)\ldots e_k(x)}$	the best representative class of $BKS(e_1(x),\ldots,e_K(x))$

The combination function of BKS is defined as follows:

$$F(e(x)) = \begin{cases} R_{e_1(x)\ldots e_K(x)} & \text{if } T_{e_1(x)\ldots e_K(x)} > 0 \\ & \text{and } \dfrac{n_{e_1(x)\ldots e_K(x)}(R_{e_1(x)\ldots e_K(x)})}{T_{e_1(x)\ldots e_K(x)}} \\ & \quad \geqslant \lambda, \\ M+1 & \text{otherwise} \end{cases} \tag{19}$$

λ is a threshold value to decide whether the result is rejected or not. For each class, there is a $n_{e_1(x)\ldots e_K(x)}(m)/T_{e_1(x)\ldots e_K(x)} \times 100$ percent probability to class m. If rejection is not allowable, then the class with the highest probability is the best and the safest choice as the final decision [15].

- Voting: The result supported by the majority of ANN's is the output of the ensemble. $x \to C_i$ means that pattern x belongs to class i.

$$F(e(x)) = j \quad \text{if } S(x \to C_j) = \max_{i \in B}(S(x \to C_i)), \tag{20}$$

where $S(x \to C_i) = \sum_{k=1}^{K} G_k(x \to C_i)$, $i \in B$ and $G_k(x \to C_i)$ is 1 if $e_k(x) = i$ and $i \in B$, 0 otherwise.
- Winner: Among all $e_{ki}(x)$ ($k \in A$, $i \in B$), the greatest one is selected and the i value is assigned as the ensemble's choice.
- Bayesian method: This method takes each ANN's significance into account by allowing the error possibility of each ANN to affect the ensemble's results [41]. The error possibility is represented by the $M \times M$ matrix, where a matrix element $n^i_j(k)$ is the number of data which belongs to class i while the kth ANN outputs j. This error is used to calculate the possibility that the kth ANN classifies the input x to class i. The reliability function $BEL(i)$ is represented as follows:

$$BEL(i) = \prod_{k=1}^{K} P(x \to C_i | e_k(x)). \tag{21}$$

Finally, the combination function $F(e(x))$ is defined

$$F(e(x)) = \begin{cases} j & \text{if } BEL(j) = \max_{i \in B} BEL(i) \\ & \quad \geqslant \alpha \ (0 < \alpha \leqslant 1), \\ \textit{reject} & \text{otherwise,} \end{cases} \tag{22}$$

where α is a threshold.
- Borda function: Let r^i_k be the ranking of the ith class of the kth ANN. The Borda score is calculated by summing $M - r^i_k$. The output of an ensemble is determined by the following equation:

$$F(e(x)) = \max_{i \in B} \sum_{k=1}^{K} (M - r^i_k(x)). \tag{23}$$

- Condorect function: With the Condorect function, the output of an ensemble is the class with the highest Condorect value. The Condorect value of a class is the minimum among the numbers of classifiers that rank the class higher than other classes. Mathematically, the output of an ensemble is determined as follows:

$$F(e(x)) = \max_{i \in B}\left(\min_{j \in B - \{i\}} \#(K : r^i_k > r^j_k)\right), \tag{24}$$

where $\#(K : r^i_k > r^j_k)$ is the number of ANN's that rank class i higher than j.
- Average: The output of an ensemble in this method is determined as the class with the highest average value of the sum of output $e_{ki}(x)$ of each class $i (k \in A)$:

$$F(e(x)) = \max_{i \in B}\left\{\left(\sum_{k=1}^{K} e_{ki}(x)\right) \middle/ K\right\}. \tag{25}$$

- Weighted average: The weighted average multiplies the weight w to the output of each ANN when averaging the outputs. Let E_i be the error rate of the ith ANN, and the weight w_i is computed as follows:

$$w_i = \frac{1 - E_i}{\sum_{k=1}^{K}(1 - E_k)}. \tag{26}$$

- Gating: Gating is a method for choosing the fittest ANN by utilizing the information from the learning data. Four steps are required to do this.

Step 1: In the learning stage, all ANN's generate a data list for which they have produced correct outputs.

Step 2: In the test stage, the learning data most similar to the input data are searched.

Step 3: An ANN which has recognized the searched learning data correctly is selected.

Step 4: The chosen ANN is applied to the test data.

4.3. *Results and analysis*

Single linkage cluster analysis is used to analyze the speciation of the ANN's on the Australian credit approval dataset. From the last population, the species are identified by using single linkage clustering. Fig. 7 shows the dendrograms of the last population evolved by speciation and simple genetic algorithms, respectively. Fig. 7(a) shows a dendrogram of the population of ANN's speciated with Kullback–Leibler entropy with single linkage cluster analysis. The figures show the difference of diversity between the methods clearly.

Tables 2–4 show the combination results of neural networks speciated by three distance measures and non-speciated neural networks on the Australian credit approval data, the breast cancer data, and the diabetes data. The speciation methods perform better than the non-speciation one. Kullback–Leibler entropy with BKS combination shows better performance on the breast cancer data. These results were produced from 10 runs.

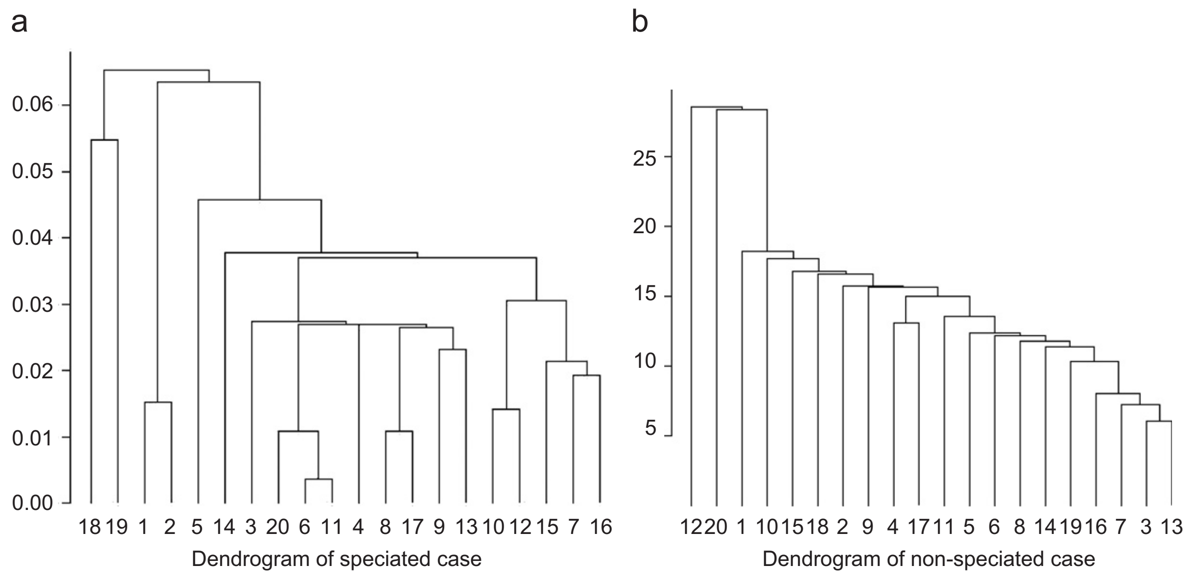

Fig. 7. Dendrogram analysis (Y-axis is the dissimilarity between the individuals). (a) Dendrogram of speciated case; (b) Dendrogram of non-speciated case.

Table 2
Recognition rates of speciated neural network ensembles: Australian credit card approval (predictive accuracy)

	Non-speciation	Average output	Pearson correlation	KL-entropy
Gating	0.8163 ± 0.033	0.8563 ± 0.004	0.8607 ± 0.017	0.8097 ± 0.079
Voting	0.8269 ± 0.010	0.8423 ± 0.012	0.8445 ± 0.010	0.8387 ± 0.010
Winner	0.7717 ± 0.067	0.7724 ± 0.082	0.8173 ± 0.043	0.7825 ± 0.077
Average	0.8485 ± 0.009	0.8522 ± 0.008	0.8552 ± 0.011	0.8642 ± 0.007
Weighted average	0.8485 ± 0.009	0.8522 ± 0.008	0.8552 ± 0.011	0.8643 ± 0.007
Bayesian	0.8064 ± 0.031	0.8358 ± 0.024	0.8453 ± 0.015	0.8103 ± 0.055
Borda	0.8375 ± 0.011	0.8481 ± 0.012	0.8550 ± 0.011	0.8600 ± 0.012
Condorect	0.8400 ± 0.011	0.8481 ± 0.012	0.8575 ± 0.008	0.8597 ± 0.012
BKS	0.9048 ± 0.010	0.9110 ± 0.001	0.9080 ± 0.015	0.9040 ± 0.014

Table 3
Recognition rates of speciated neural network ensembles: breast cancer (predictive accuracy)

	Non-speciation	Average output	Pearson correlation	KL-entropy
Gating	0.9190 ± 0.038	0.9583 ± 0.018	0.9660 ± 0.016	0.8795 ± 0.080
Voting	0.9490 ± 0.017	0.9694 ± 0.013	0.9681 ± 0.008	0.9461 ± 0.030
Winner	0.9149 ± 0.047	0.9724 ± 0.016	0.9337 ± 0.043	0.8816 ± 0.08
Average	0.9681 ± 0.009	0.9766 ± 0.006	0.9800 ± 0.008	0.9749 ± 0.013
Weighted average	0.9671 ± 0.009	0.9769 ± 0.005	0.9800 ± 0.008	0.9749 ± 0.013
Bayesian	0.9289 ± 0.033	0.9268 ± 0.038	0.9494 ± 0.027	0.9362 ± 0.035
Borda	0.9644 ± 0.010	0.9714 ± 0.011	0.9752 ± 0.007	0.9748 ± 0.013
Condorect	0.9698 ± 0.007	0.9750 ± 0.007	0.9815 ± 0.007	0.9799 ± 0.008
BKS	0.9810 ± 0.008	0.9875 ± 0.008	0.9875 ± 0.006	0.9882 ± 0.010

In the student t-test, the difference of accuracy between an average output with BKS, and an entropy output with BKS for the Australian credit card data is not statistically significant ($t = 1.577$, $p = 0.1$). For more accurate results, additional experiments were conducted for the breast cancer and diabetes datasets. The difference of accuracy between the average output with BKS and KL entropy with BKS for breast cancer is statistically significant ($t = 1.776$, $p = 0.1$). The difference of accuracy between Pearson correlation with BKS and KL entropy with BKS for diabetes is statistically significant ($t = 1.753$, $p = 0.1$). Table 5 summarizes the predictive accuracy over the three

Table 4
Recognition rates of speciated neural network ensembles: diabetes (predictive accuracy)

	Non-speciation	Average output	Pearson correlation	KL-entropy
Gating	0.5819 ± 0.052	0.6807 ± 0.027	0.6888 ± 0.040	0.6383 ± 0.090
Voting	0.5152 ± 0.052	0.6078 ± 0.027	0.5950 ± 0.040	0.5445 ± 0.090
Winner	0.5303 ± 0.052	0.6078 ± 0.027	0.5950 ± 0.040	0.5445 ± 0.090
Average	0.6126 ± 0.026	0.7453 ± 0.025	0.7194 ± 0.030	0.7476 ± 0.023
Weighted average	0.6128 ± 0.025	0.7424 ± 0.023	0.7208 ± 0.029	0.7476 ± 0.023
Bayesian	0.6475 ± 0.072	0.6655 ± 0.068	0.6590 ± 0.075	0.6663 ± 0.073
Borda	0.5625 ± 0.025	0.7349 ± 0.025	0.7189 ± 0.020	0.7389 ± 0.031
Condorect	0.5832 ± 0.024	0.7325 ± 0.027	0.7293 ± 0.020	0.7457 ± 0.025
BKS	0.5975 ± 0.037	0.7820 ± 0.036	0.7950 ± 0.028	0.7980 ± 0.047

Table 5
Recognition rates of speciated neural network ensembles over three data sets (Australian, breast, and diabetes) (predictive accuracy)

	Accuracy	Sensitivity	Specificity
Gating	0.8046	0.7045	0.8587
Voting	0.7872	0.7104	0.8298
Winner	0.7603	0.6793	0.8024
Average	0.8453	0.7907	0.8734
Weighted average	0.8452	0.7512	0.8937
Bayesian	0.8064	0.7047	0.8559
Borda	0.8368	0.7406	0.8881
Condorect	0.8418	0.7769	0.8741
BKS	0.8787	0.8220	0.9062

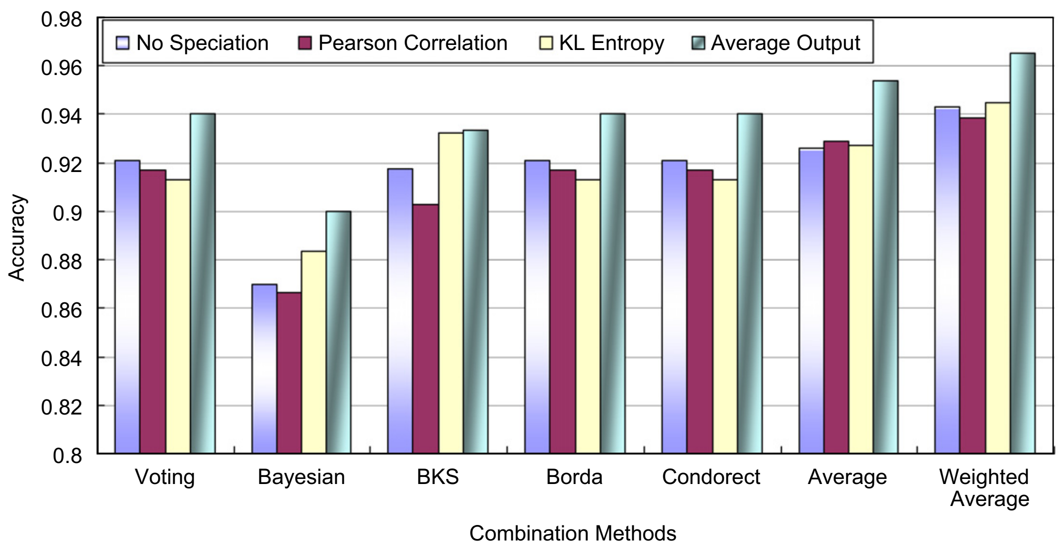

Fig. 8. LOOCV results on colon dataset.

data sets. Fig. 8 shows experimental results on colon cancer dataset. It shows that average output methods is the best one and weighted average shows relatively high accuracy.

Fig. 9 shows the results of diversity analysis. The average value of the KL distances among all individuals in the population is used to measure the diversity. In this figure, the speciation method maintains higher diversity than the non-speciation one. Increasing the population size is a good way to increase the performance but it requires much computational cost. The sensitivity to the crossover rate (mutation rate is fixed as 0.1) on Australian data shows that the proposed method is sensitive to the crossover rate (0.2–0.6) and 0.3 is the best. Increasing the maximum number of generation does not mean continuous

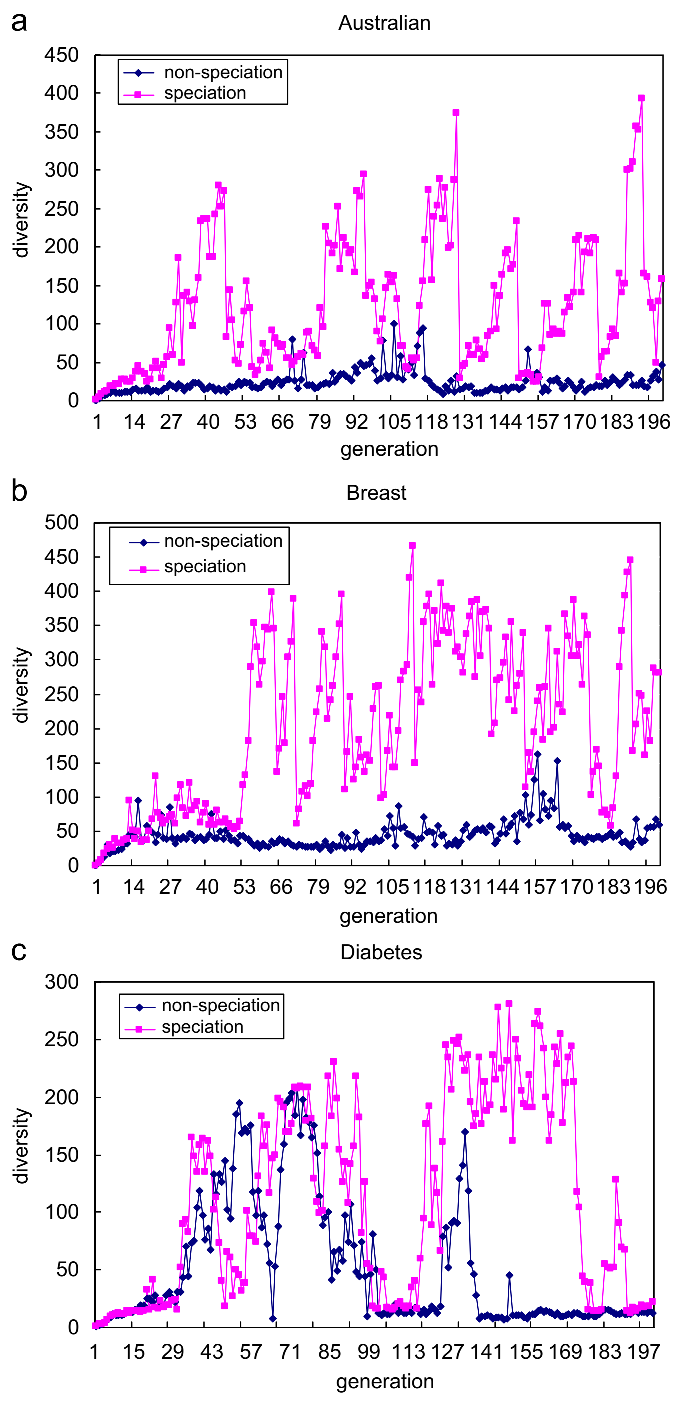

Fig. 9. Diversity analysis (Australian credit approval data): (a) Australian; (b) breast; (c) diabetes.

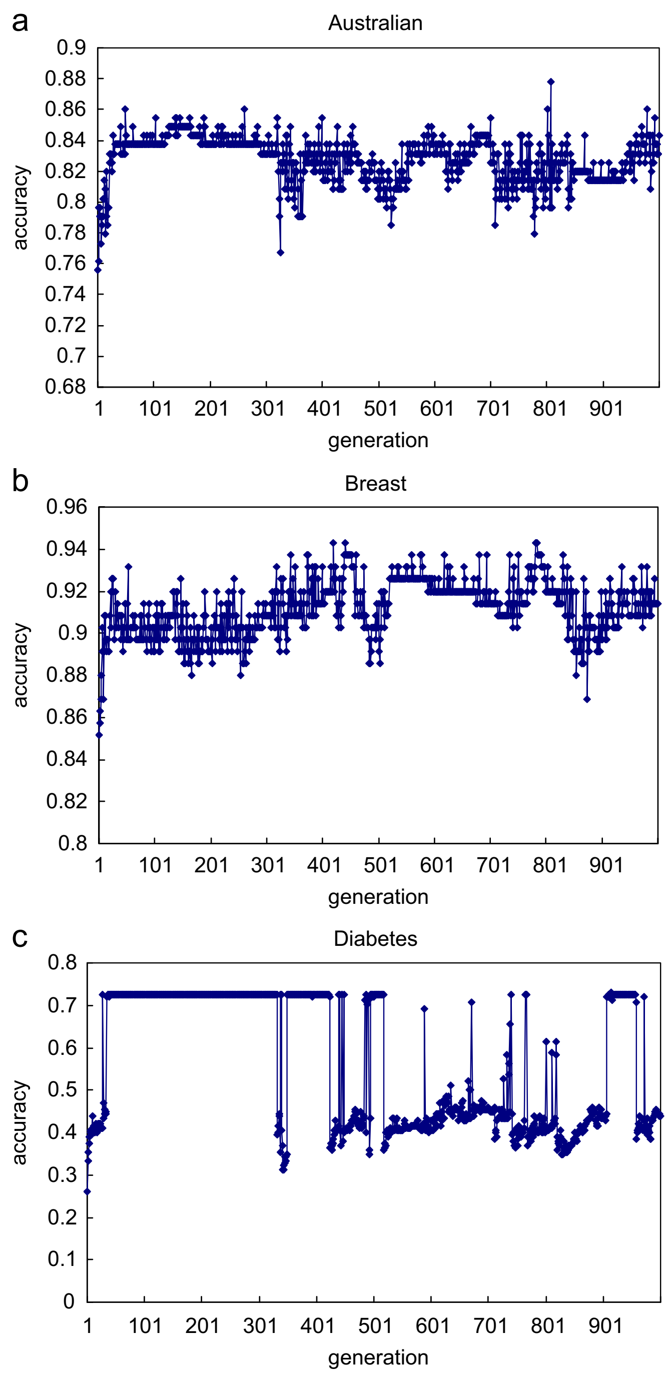

Fig. 10. Maximum accuracy of each generation on validation data for 1000 generations: (a) Australian; (b) breast; (c) diabetes.

performance improvement as depicted in Fig. 10. In fact, it is not possible to test the effect of all the parameters because it needs a number of repeated runs and they are also related to other parameters. In this paper, we have shown results of part of them.

In one generation, we need to measure distance $P \times P/2$ times. P is a population size, m is a number of output nodes and n is a number of training data. For measuring the KL entropy distance, we need to perform basic operations (measuring entropy) mn times. The time complexity of measuring the KL entropy distance is $O(P^2 \times m \times n)$. The number of generations is G and the number of partial training is L. The size of the problem is dependent on the G, P, and n. The time complexity of the speciated evolution is $O(GnP^2)$. In non-speciation, the complexity is $O(GnP \log P)$. Compared to non-speciation, the proposed method needs more computational resources when calculating the distance among individuals.

Because the evolutionary neural network approach maintains multiple solutions, it has an inherent shortcoming in time. Compared to other methods such as EPNET [43], our method uses the speciation technique and needs additional computational resources. Apart from this shortcoming, the paper presented an effective way to create an ensemble of evolutionary neural networks for better performance. The main processes of the proposed method consist of partial training, fitness sharing, and fitness evaluation. Fitness sharing is not a time consuming process because it uses the output values of the training data and the validation data. Only the distance calculation using the stored outputs is required.

4.4. Comparison

To estimate the performance of the proposed method, a comparative study was conducted. We compared the BKS + KL method with the other published results. The results in Tables 6–8 are the average of 10 runs. FNNCA [36] is an ANN with a constructive algorithm. EPNET [43] is an evolutionarily constructed ANN by Yao and Liu. The comparisons show that the BKS combination of ANN's speciated by Kullback–Leibler entropy outperformed FNNCA and EPNET. Table 8 shows the test results of the BKS + KL and comparisons with the other methods on the diabetes dataset. The BKS combination with speciation by Kullback–Leibler entropy shows the best performance in the breast cancer dataset. Table 9 shows the performance comparison with other ensemble methods on the Australian credit card approval data. The proposed method performs better than a non-evolutionary ensemble such as CELS as well as an evolutionary ensemble such as Evo-En-RLS. In sum, we can verify that the proposed method yields competitive results.

As shown in Table 10, recent works adopt different approaches to increase diversity in the evolutionary ensembles of neural networks. The real values in the table represent error rates on the dataset. [10,24] use implicit sharing and [23] use minimization of mutual information. However, we use standard fitness sharing by defining

Table 6
Comparison of recognition rates: Australian credit card approval (average error rate $\pm\sigma$)

Methods	Errors	Time complexity
BKS + KL	0.096 ± 0.014	$O(Pnm)$
BKS + Pearson	0.092 ± 0.0156	$O(Pnm)$
BKS + average	0.089 ± 0.0	$O(Pnm)$
EPNET	0.115	$O(Pm)$
Cal5	0.131	$O(m)$
ITrule	0.137	$O(m)$
DIPOL92	0.141	$O(m)$
Discrim	0.141	$O(m)$
Logdisc	0.141	$O(m)$
SVM	0.145	$O(m)$
CART	0.145	$O(m)$
RBF	0.145	$O(m)$
CASTLE	0.148	$O(m)$
NaiveBay	0.151	$O(m)$
IndCART	0.152	$O(m)$
BP	0.154	$O(m)$
C4.5	0.155	$O(m)$
k-NN	0.181	$O(m)$

Table 7
Comparison of recognition rates: breast cancer

Methods	Errors	Time complexity
BKS + KL	0.0118 ± 0.01	$O(Pnm)$
BKS + Pearson	0.0125 ± 0.006	$O(Pnm)$
BKS + average	0.0125 ± 0.008	$O(Pnm)$
EPNET	0.01376	$O(Pm)$
FNNCA	0.0145	$O(m)$
SVM	0.0457	$O(m)$
Naïve Bayes	0.0571	$O(m)$
BP	0.0800	$O(m)$
RBF	0.0857	$O(m)$

Table 8
Comparison of recognition rates: diabetes

Methods	Errors	Time complexity
BKS + KL	0.202 ± 0.047	$O(Pnm)$
BKS + Pearson	0.205 ± 0.028	$O(Pnm)$
BKS + average	0.218 ± 0.036	$O(Pnm)$
Logdisc	0.223	$O(m)$
EPNET	0.224	$O(Pm)$
DIPOL92	0.224	$O(m)$
Discrim	0.225	$O(m)$
SMART	0.232	$O(m)$
RBF	0.243	$O(m)$
Itrule	0.245	$O(m)$
BP	0.248	$O(m)$
Cal5	0.25	$O(m)$
CART	0.255	$O(m)$
CASTLE	0.258	$O(m)$
Quadisc	0.262	$O(m)$
Naïve Bayes	0.262	$O(m)$
C4.5	0.270	$O(m)$
SVM	0.322	$O(m)$
k-NN	0.324	$O(m)$

Table 9
Comparison of recognition rates with other ensemble methods: Australian credit card approval

Methods	Errors	Time complexity
BKS + KL	0.096 ± 0.014	$O(Pnm)$
BKS + Pearson	0.092 ± 0.0156	$O(Pnm)$
BKS + average	0.089 ± 0.0	$O(Pnm)$
Evo-En-RLS [44]	0.095	$O(Pm)$
CELS [21]	0.120	$O(Pm)$
EENCLMI [22]	0.130	$O(Pm)$
EENCL [24]	0.135	$O(Pm)$

Table 10
Comparison of recent evolutionary ensemble research

Proposed	Speciation	Learning	Breast	Australian	Diabetes	Evaluation
	Explicit FS	BP	0.012	0.089	0.202	Average of 10 runs
EENCL [24]	Implicit FS	NCL	–	0.132	0.221	10-fold (Australian) 12-fold (diabetes)
Speciation + EENCL [10]	Implicit FS with island model population structure	NCL	0.024	0.122	0.233	Average of 30 runs
Speciation + EENCL [23]	Minimization of mutual information	NCL	–	0.133	–	10-fold

NCL = negatively correlation learning, FS = fitness sharing.

distance measures between two neural networks. Although the proposed method performs a bit better than other works, it is difficult to gauge the superiority of one approach because they adopt different evaluation methods.

5. Conclusions

In this paper, the average output, Pearson correlation, and Kullback–Leibler entropy distance measures were used to measure the distance between two distinct neural networks. Using this measure, it was possible to improve the diversity of the evolved neural networks. From the experimental results, the combination of the neural networks speciated by the fitness sharing distance measure showed promising results compared to non-speciation methods and other published results.

Though there are many works on evolving neural networks, performance is not the best for all datasets [8,21,24,39] and the proposed method does not always outperform for all the benchmark datasets. Although the proposed method does not produce the best results in all the datasets, it does show some improvements compared with previous works in this field.

Future works are as follows. Because calculating the distance between two distinct neural networks has not yet been perfectly solved, information geometry-based distance measures for neural networks can be used for performance improvement [4]. Also, multiclass data should be used for proving the applicability of the method.

Acknowledgments

This work was supported by Brain Science and Engineering Research Program sponsored by the Korean Ministry of Science and Technology.

References

[1] H.A. Abbass, An evolutionary artificial neural networks approach for breast cancer diagnosis, Artif. Intell. Med. 25 (2002) 265–281.
[2] J.-H. Ahn, S.-B. Cho, Speciated neural networks evolved with fitness sharing technique, in: Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, 2001, pp. 390–396.
[3] L.A. Alexandre, A.C. Campilho, M. Kamel, On combining classifiers using sum and product rules, Pattern Recognition Lett. 22 (2001) 1283–1289.
[4] S.-I. Amari, Information geometry of the EM and em algorithms for neural networks, Neural Networks 8 (9) (1995) 1379–1408.
[5] H. Atincay, M. Demirekler, Speaker identification by combining multiple classifiers using Dempster-Shafer theory of evidence, Speech Commun. 41 (2003) 531–547.
[6] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorization, Inf. Fusion 6 (2005) 5–20.
[7] R. Bryll, R. Gutierrez-Osuna, F. Quek, Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets, Pattern Recognition 36 (2003) 1291–1302.
[8] P.A. Castillo, V. Rivas, J.J. Merelo, J. Gonzalez, A. Prieto, G. Romero, G-Prop-II: global optimization of multilayer perceptrons using GAs, in: Proceedings of the 1999 Congress on Evolutionary Computation, vol. 3, May 1999, pp. 2022–2027.
[9] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley-Interscience, New York, 1991.
[10] P. Duell, I. Fermin, X. Yao, Speciation techniques in evolved ensembles with negatively correlation learning, in: IEEE Congress on Evolutionary Computation, 2006, pp. 3317–3321.
[11] S.I. Gallant, Perception-based learning algorithms, IEEE Trans. Neural Networks 1 (2) (1990) 179–191.
[12] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading, Massachusetts, 1989.
[13] D.E. Goldberg, J. Richardson, Genetic algorithms with sharing for multimodal function optimization, in: Proceedings of the Second International Conference on Genetic Algorithms, 1987, pp. 41–49.
[15] A. Khotanzad, C. Chung, Hand written digit recognition using BKS combination of neural network classifiers, in: Proceedings of the IEEE Southwest Symposium on Image Analysis and Interpretation, 1994, pp. 94–99.
[16] K.-J. Kim, S.-B. Cho, Fuzzy integration of structure adaptive SOM's for web content mining, Fuzzy Sets and Systems 148 (1) (2004) 43–60.
[18] S.-I. Lee, J.-H. Ahn, S.-B. Cho, Exploiting diversity of neural ensembles with speciation evolution, International Joint Conference on Neural Networks, vol. 2, 2001, pp. 808–813.
[19] H. Liang, G. Dai, Improvement of cascade correlation learning, Inf. Sci. 112 (1–4) (1998) 1–6.
[20] R.P. Lippmann, Neural networks, Bayesian a posteriori probabilities and pattern classification, in: From Statistics to Neural Networks-Theory and Pattern Recognition Applications, 1994.
[21] Y. Liu, X. Yao, Simultaneous training of negatively correlated neural networks in an ensemble, IEEE Trans. Systems Man Cybernet. Part B Cybernet. 29 (66) (1999) 716–725.
[22] Y. Liu, X. Yao, Learning and evolution by minimization of mutual information, in: Seventh International Conference on Parallel Problem Solving from Nature, 2002, p. 495.

[23] Y. Liu, X. Yao, Evolving neural network ensembles by fitness sharing, in: IEEE Congress on Evolutionary Computation, 2005, pp. 3289–3293.
[24] Y. Liu, X. Yao, T. Higuchi, Evolutionary ensembles with negative correlation learning, IEEE Trans. Evol. Comput. 4 (4) (2000) 380–387.
[25] J.-K. Min, J.-H. Hong, S.-B. Cho, Effective fingerprint classification by localized models of support vector machines, Lecture Notes in Computer Science, vol. 3832, 2005, pp. 287–293.
[26] G.W. Milligan, M.C. Cooper, Methodology review: clustering methods, Appl. Psychol. Meas. 11 (1987) 329–354.
[27] R. Mojena, Hierarchical grouping methods and stopped rules: an evaluation, Comput. J. 20 (4) (1977) 359–363.
[28] D. Montana, L. Devis, Training feedforward neural networks using genetic algorithm, in: Proceedings of the Eleventh International Conference on Artificial Intelligence, 1989, pp. 762–767.
[29] N.G. Pedrajas, C.H. Martinez, J.M. Perez, COVNET: a cooperative coevolutionary model for evolving artificial neural networks, IEEE Trans. Neural Networks 14 (3) (2003) 575–596.
[31] H. Qian, Relative entropy: free energy associated with equilibrium fluctuations and nonequilibrium deviations, Phys. Rev. E 63 (2001), 042103/1-042103/4.
[32] A.F.R. Rahman, M.C. Fairhurst, Multiple classifier decision combination strategies for character recognition: a review, Int. J. Document Anal. Recognition 5 (4) (2003) 166–194.
[33] R. Reed, Pruning algorithms—a survey, IEEE Trans. Neural Networks 4 (5) (1993) 740–747.
[34] M.D. Richard, R.P. Lippmann, Neural network classifiers estimate Bayesian a posteriori probabilities, Neural Comput. 3 (1991) 461–483.
[35] A. Rogers, A. Prügel-Bennett, Genetic drift in genetic algorithm selection schemes, IEEE Trans. Evol. Comput. 3 (4) (1999) 298–303.
[36] R. Setino, L.C.K. Hui, Use of a quasi-Newton method in a feedforward neural network construction algorithm, IEEE Trans. Neural Networks 6 (1) (1995) 273–277.
[37] D.W. Taylor, D.W. Corne, D.L. Taylor, J. Harkness, Predicting alarms in supermarket refrigeration systems using evolved neural networks and evolved rulesets, in: Congress on Evolutionary Computation, 2002, pp. 1988–1993.
[38] P.J. Werbos, The Roots of Backpropagation: From Ordered Derivatives to Neural Networks and Political Forecasting, Wiley, New York, 1994, pp. 372–373.
[39] D. Wicker, M.M. Rizki, L.A. Tamburino, E-Net: evolutionary neural network synthesis, Neurocomputing 42 (2002) 171–196.
[40] T. Windeatt, Diversity measures for multiple classifier system analysis and design, Inf. Fusion 6 (1) (2005) 21–36.
[41] L. Xu, A. Krzyzak, C.Y. Suen, Methods of combining multiple classifiers and their applications to handwriting recognition, IEEE Trans. Systems Man Cybernet. SMC-22 (3) (1992) 418–435.
[42] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (9) (1999) 1423–1447.
[43] X. Yao, Y. Liu, A new evolutionary system for evolving artificial neural networks, IEEE Trans. Neural Networks 8 (3) (1997) 694–713.
[44] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, IEEE Trans. Systems Man Cybern. part B Cybernet. 28 (3) (1998) 417–425.
[45] X. Yao, Y. Liu, Towards designing artificial neural networks by evolution, Appl. Math. Comput. 91 (1) (1998) 83–90.
[46] Z.-H. Zhou, Y. Jiang, Y.-B. Yang, S.-F. Chen, Lung cancer cell identification based on artificial neural network ensembles, Artif. Intell. Med. 24 (2002) 25–36.

Kyung-Joong Kim (Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the Yonsei University, Seoul, Korea, in 2000 and 2002, respectively. Since 2002, he has been a Ph.D. student in the Department of Computer Science, Yonsei University. His research interests include evolutionary neural network, robot control, and agent architecture.

Sung-Bae Cho (Member, IEEE) received the B.S. degree in computer science from the Yonsei University, Seoul, Korea, in 1988 and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Taejeon, Korea, in 1990 and 1993, respectively.

From 1991 to 1993, he worked as a Member of the Research Staff at the Center for Artificial Intelligence Research at KAIST. From 1993 to 1995, he was an Invited Researcher of the Human Information Processing Research Laboratories at ATR (Advanced Telecommunications Research) Institute, Kyoto, Japan. In 1998, he was a Visiting Scholar at the University of New South Wales, Canberra, Australia. Since 1995, he has been a Professor in the Department of Computer Science, Yonsei University. His research interests include neural networks, pattern recognition, intelligent man–machine interfaces, evolutionary computation, and artificial life.

Dr. Cho is a Member of the Korea Information Science Society, INNS, the IEEE Computer Society, and the IEEE Systems, Man and Cybernetics Society. He was awarded outstanding paper prizes from the IEEE Korea Section in 1989 and 1992, and another one from the Korea Information Science Society in 1990. In 1993, he also received the Richard E. Merwin prize from the IEEE Computer Society. In 1994, he was listed in Who's Who in Pattern Recognition from the International Association for Pattern Recognition and received the best paper awards at the International Conference on Soft Computing in 1996 and 1998. In 1998, he received the best paper award at the World Automation Congress. He was listed in Marquis Who's Who in Science and Engineering in 2000 and in Marquis Who's Who in the World in 2001.