

EVOLVING NEURAL NETWORKS ENSEMBLES NNEs

Hany Sallam¹, Carlo S. Regazzoni², Ihab Talkhan¹, and Amir Atiya¹

¹Faculty of Engineering
Dept. of computer Engineering,
Cairo University, EGYPT

salamhany@yahoo.com, carlo@dibe.unige.it, italkhan@cu.edu.eg, amiratiya@link.net

²Faculty of Engineering
Department of Biophysics Engineering and Electronics,
Genova University, ITALY

ABSTRACT

A new method to design and evolve neural network ensembles NNEs based on speciation is presented in this paper. The main advantage of this method is that, it completely evolves NNEs by combining the evolution of neural networks and the configuration of the ensemble in one evolutionary phase. In every generation, population is evolved toward the best set of structure and weights. Then, the ensemble is configured and its performance is evaluated. Evolution is stopped if the best performance is reached or the maximum number of generations is reached. The main idea of this method is to generate NNE based on fitness sharing and genotype diversity measurement. The size of the ensemble depends on the number of species. The output of the ensemble is calculated by the weighted sum of the output of each member. The members' weights are changed dynamically from generation to generation depending on the characteristics of species existing in the current population. Experiments with Iris data, breast cancer data, and diabetes data set from the UCI machine learning repository showed that the proposed method can produce NNEs with better performance as compared to other ensemble methods.

Index Terms- Neuroevolution, Ensembles, Speciation

1. INTRODUCTION

Combining multiple evolved ANNs has been actively researched recently. The main idea of neural networks ensemble is that a population of ANNs contains more information than any single ANN in the population. Such information can be used to improve generalization performance and reliability [1]. Generally multiple ANNs in the last generation are combined to construct an ensemble that has better generalization performance provided that the last generation individuals complement each other in the generalization [1]. Each network within the ensemble has a potentially different weight in the output of the ensemble [2]. Several researches have shown that the network ensemble has a generalization error generally smaller than that obtained with a single network and also that the variance of the ensemble is less than the one of a single network [2]. To maximize the effect of combining multiple ANNs, a method for large diversity of neural networks in evolution should be used. The output of a typical ensemble with k constituents networks when and input X is presented is [2] [3]:

$$y(x) = \sum_{i=1}^k w_i y_i(x) \quad (1)$$

where y_i is the output of network i and w_i is the weight associated with the network.

It is common to use only the fittest solution of the last generation, as only information of single individual is exploited. An ensemble of individual is a more promising choice because information that is derived from combining a set of individuals might produce higher accuracy than using the information from the best individual among them. A lot of studies [4] [5] [6] [7] focused on using negative correlation learning (NCL) and backpropagation (BP) to obtain accurate and diverse ensemble by adding penalty term describing the negative correlation between networks to conventional mean square error of each network. The idea of training neural networks as a multiobjective optimization problem and using the resultant Pareto frontier to form an ensemble of networks is proposed in [5] [8] and [9]. Authors of [5] concluded that, Pareto-based ensemble is better than the obtained by BP. Another method to evolve ensemble based on fitness sharing is introduced in [10], [11], [13] and [14]. In this method the population is evolved by genetic operations crossover and mutation until the maximum number of generation is reached or the fitness is 1.0; then, networks of the last generation are trained by BP. After that the population is clustered and the representative individual of each cluster is selected to form the ensemble. Although of its excellent results, this method has two weak points: there are two training phases, and ensemble formation phase is separated from individuals training phase. The idea of evolving both the population and the ensemble in one single phase is proposed in the current paper. In this paper a new method to generate an ensemble automatically based on fitness sharing is proposed.

2. BASIC IDEA OF THE PROPOSED METHOD

The fundamental idea of the proposed method is the speciation of the whole population into a number of species through evolution as shown in Fig.1. The number of species varies from generation to generation depending on the population genotype diversity. The number of individual networks in each species varies depending on the sharing radius. At each generation, the best individual of each species is selected to be a member in the ensemble. The output of member networks is combined by the weighted sum of outputs; the weight of each member is determined by three factors, the size, the age, and the average fitness of its mother species. A member network which belongs to a species that has a large size, a long evolution age, and high average fitness, will be weighted more than a member that belongs to a small and young species with low average fitness. The sum of weights equals to one for normalization purposes.

This method can be characterized by its flexibility, which suites different problems. This flexibility can be explained on two levels. First, on the level of NNs, where evolutionary algorithms are used to evolve the structures and the weights of NNs. Second, on the ensemble level, where ensemble is evolved (members, size, and weights) by benefiting from speciation. The

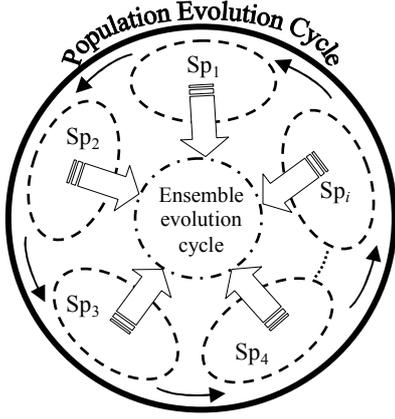


Fig. 1. Ensemble evolution cycle

size of the ensemble is dynamically determined through the evolution. The share of different members in the output of the ensemble is determined based on characteristics of its mother species, where species with high performance live a long number of generations, while low performance species die out. Also the size of species is important aspect since species with large number of individuals will have high share rate in the production of the new generation individuals. The old method can be seen as ensemble method where the best individual weight is 1.0 i.e. the contribution of the other individuals is neglected. So it can be said that the old method is special case of ensemble, where the whole population represents the ensemble.

3. SPECIES EVOLUTION

Fitness sharing is the best method to speciate a population of neural networks. Speciation in genetic algorithms creates different species, each embodying a sub-solution, which means to create not only the best one but also diverse solutions [12] [14]. In each generation individuals are placed in species. Each species is represented by a random genome inside the species from the previous generation. A given individual in the current generation is placed in the first species in which this individual is compatible with the representative individual of that species. If this individual is not compatible with any existing species, a new species is created with that individual as its representative [15]. Every species is assigned a potentially different number of offspring in proportion to the sum of shared fitness of its individuals. Species then produce by first eliminating the lowest performing member from the population. The entire population is then replaced by the offspring of the remaining individuals in each species [15].

4. KEY POINTS OF THE PROPOSED METHOD

The design of neural networks ensemble implies making many decisions that have major impact on the performance of ensemble. The most important decisions that should be taken in designing an ensemble are [2]: 1) the method of designing and training the individual networks, 2) the method of combining the individual networks, 3) the method of measuring the performance of individual networks, 4) the method of encouraging diversity among the members of ensemble and how to measure such diversity. Based on these decisions of designing neural network ensembles NNEs, we propose some key points to design and evolve NNEs:

1. Individual networks are evolved (weights and structure) by genetic operations crossover and mutation as in [15].
2. Most of introduced methods fix the size of the NNE to a given number of NNs; in our method, the size of the ensemble varies every generation and the final ensemble size is determined at the end of evolution.
3. The weighted sum of the outputs method is used to combine the output of the NNE members Eq. (1); the members weights are obtained by a new method based on the species characteristics.
4. Fitness sharing speciation is used to keep and promote the diversity between the individuals of the ensemble, and diversity is measured by a new metric “neuro-edit” based on the genotypic similarity.
5. The performance of the ensemble is evaluated as the rate of correct classification. The performance of the individuals of the population is evaluated as a function of the average error on the training data set and the total number of species in the population, i.e. diversity degree is factored as an objective in the objective function.
6. All the population individuals are initialized to the same minimal structure (input nodes are fully connected to output nodes, no hidden nodes, and the weights of connections are initialized randomly) and trained on the same data set for the same number of generations.

5. WEIGHTS OF THE ENSEMBLE MEMBERS

1. For an initial population p of n neural networks:

$$p = \{p_1, p_2, \dots, p_n\}.$$

2. Through evolution, this population is speciated into m species: $Sp = \{Sp_1, Sp_2, \dots, Sp_m\}$, and each species has a different number of individuals. Initially, at the first generation all the population is speciated to one species since, initial population individuals have the same genotype structure. After the first generation, individuals begin to have different genotypes and to be speciated to more than one species.

Species satisfy the following conditions:

$$1) Sp_i \cap Sp_j = \varnothing, \text{ and}$$

$$2) \bigcup_{i=1}^m Sp_i = p,$$

This means that each individual in the population is speciated to only one species, and the sum of all species individuals equal to the population size.

Species can be characterised by the tuple $\langle A_{Sp_i}, S_{Sp_i}, F_{Sp} \rangle$:

1. A , age of species, the number of generations in which this species is still alive, to indicate the experience gained by its members by training, $1 \leq A_{Sp_i} \leq k$, where k is the maximum number of generations.
2. S , size of species, number of individuals, to reflect the power of the species to produce.
3. F , average fitness of species, to reflect the performance of species.

The weight of each species can be calculated as a function of its age, size, and its average fitness as follows:

$$\omega_i = \alpha A_i + \beta S_i + \gamma F_i \quad (2)$$

$$0 \leq \alpha, \beta, \gamma \leq 1$$

The values of α , β and γ can be selected to tune the importance of the species parameters, age, size and average fitness.

The weight of an ensemble member selected from Sp_i is calculated as $w_i = \frac{\omega_i}{W}$, where $W = \sum_{i=1}^m \omega_i$.

6. FINTESS CALCULATIONS

A useful NN to be added to an ensemble is one that correctly classifies as many examples as possible while making mistakes on examples that most of the current population members correctly classify [16]. An ideal ensemble is one that has members where, each member has different error set on a given data set. This means that, the ensemble members should be diverse, to address this condition during the evolution of the population, the whole population similarity degree is factored as a term in the objective function of each individual. The population similarity is defined as:

$$p_s = \frac{\lambda}{no_of_species} \quad \text{where } \lambda \text{ is a variable to tune the}$$

importance of similarity term. The fitness function is defined as:

$$F_i = [M - (E_i + p_s)]^2,$$

at the beginning of the evolution, the similarity term equals to λ , with highly speciated population the similarity term will be nearly zero and the fitness will depend on the individual average

error, $E_i = \frac{1}{M} \left[\sum_{j=1}^M y_i(x_j) - d_j \right]$, where M is the size of the

training set, $y_i(x_j)$ is the output of individual i on the j^{th}

pattern of the training data set, d_j is the desired output. The performance of the ensemble will be measured as the correct classification rate on data set.

7. FITNESS SHARING TECHNIQUE

Fitness sharing is a technique that penalizes genomes that inhabit neighbourhoods of many other genomes. Generally, an individual's fitness evaluation is divided by a sharing factor that measures the genome's proximity to other genomes in the population. Genomes in heavily populated peaks receive a high penalty, which translates into a lower probability of propagating to the next generation. This technique is intended to spread the population across several peaks in the solution space, with wider or higher peaks able to support more individuals [14]. Ensembles are effective when their members are both accurate and diverse. Speciation through fitness sharing creates a diverse set of solutions to exploit different niches in the fitness landscape [17]. Raw fitness scores are shared amongst similar individuals. The definition of similarity and the mechanism of sharing vary in this paper. The similar individuals are those which have similar genotypes (structures and weights). Individuals of similar genotypes have similar fitness values or performances.

Given that f_i the fitness of an individual i and $sh(d_{ij})$ is a sharing function, the sharing fitness f_{si} is computed as [12]:

$$f_{si} = \frac{f_i}{\sum_{j=1}^p sh(\delta(d_{ij}))} \quad (3)$$

The sharing function $sh(d_{ij})$ is set to 0 when distance d_{ij} is above the threshold δ_i , otherwise, $sh(d_{ij})$ is set to 1 as in the following equation [15]:

$$sh(d_{ij}) = \begin{cases} 1, & d_{ij} < \delta_i \\ 0, & d_{ij} \geq \delta_i \end{cases} \quad (4)$$

The sharing radius is determined by the following equation:

$$\delta = \frac{1}{n(n-1)/2} \sum_{i=1}^{p-1} \sum_{j=i+1}^p d_{ij} \quad (5)$$

Where p is the population size, and d_{ij} is the genotype

distance between i^{th} and j^{th} NN's and is measured according to a new proposed method explained in the next section. Fitness sharing decrease the increment of fitness of densely populated ANN space and shares the fitness with other space [12] [13]. With fitness sharing the genetic algorithm finds more diverse solutions although some of the solutions are not good.

7.1 Measuring Genotype Diversity

Based on the nature of evolvable neural networks ENNs, a new measure defined as "neuro-edit" is introduced in [18] to measure the distance between neural networks based on measuring similarity between neural networks in terms of connection genes. This measure is based on the encoding method presented in [15]. It is not enough that a connection gene in one genome to be similar to a connection gene in another genome, although they have the same in-node and the out-node, since the weights and states of such genes may be different. If they have the same weight and status, then they are completely similar and distance between them equals to 0, otherwise the distance will not be 0. The computation of distance between two chromosomes can be divided into two parts. The first part measures the distance between common genes (i.e. genes that have the same id), and the other part for uncommon genes.

Common genes distance: to calculate the distance between two genes with the same id (genes exist in both chromosomes), the status of each gene is checked, if both genes are enabled, then distance between them will depend on their weights. In the case of similar weights, the distance will equal to 0. In the case of dissimilar weights, the distance will equal to the absolute difference between weights normalized by the maximum of absolute value of weights. In the case of, one of the genes has a disabled status, this means that the gene is useless or not actually functioning in the phenotype and the two genes are considered dissimilar and distance between them will equal to 1. The total distance between n common genes of two chromosomes C_1 and C_2 is calculated by adding up distances as follows:

$$d_{com} = \frac{1}{n} \sum_{i=1}^n \frac{\left| \left(st(g_i)_{C_1} * |w(g_i)_{C_1}| - st(g_i)_{C_2} * |w(g_i)_{C_2}| \right) \right|}{\max \left(st(g_i)_{C_1} * |w(g_i)_{C_1}|, st(g_i)_{C_2} * |w(g_i)_{C_2}| \right)} \quad (6)$$

Where, $st(g_i)_{C_1}$ and $w(g_i)_{C_1}$ are the state and the weight of a common connection gene $g_i \in C_1$, and $st(g_i)_{C_2}$ and $w(g_i)_{C_2}$ are the state and the weight of a common connection gene $g_i \in C_2$.

Uncommon genes distance: the distance between n uncommon genes in C_1 , and m uncommon genes in C_2 , is given by:

$$d_{uncom} = \frac{1}{n} \sum_{i=1}^n st(g_i)_{C_1} + \frac{1}{m} \sum_{j=1}^m st(g_j)_{C_2} \quad (7)$$

Where, $st(g_i)_{C_1}$ is the state of an uncommon connection gene $i \in C_1$, $st(g_j)_{C_2}$ is the state of an uncommon connection gene $j \in C_2$.

The distance of uncommon genes depends only on their status. The total distance between two chromosomes C_1 and C_2 is given by:

$$d(C_1, C_2) = \frac{1}{3}(d_{com} + d_{uncom}) \quad \text{where} \quad (8)$$

$$0 \leq d(C_1, C_2) \leq 1$$

Distance between two genomes satisfies the following conditions:

$$\forall C_1, C_2 \in P \quad \begin{aligned} 1) & d(C_1, C_1) = 0, \\ 2) & d(C_1, C_2) > 0 \text{ if } C_1 \neq C_2 \\ 3) & d(C_1, C_2) = d(C_2, C_1) \end{aligned} \quad (9)$$

It can be easily shown that, $d(C_1, C_2)$ satisfies the triangle inequality [18], $d(C_1, C_2) \leq d(C_1, C_3) + d(C_2, C_3)$ where C_1, C_2 and C_3 are three different chromosomes.

8. EVOLVING NNE ALGORITHM

As shown in Fig.2 the main steps of the algorithm are:

- 1- Generate initial population of N networks, all networks have the same structure, input nodes, output nodes, no hidden nodes, and input nodes are fully connected to output nodes. The weights are randomly initialized. The number of input nodes equal to the number of features in the training data, and the number output nodes equal to the number of classes.
- 2- Speciate the population based on fitness sharing by using genotype diversity of the population as threshold. The number of species in a population depends on the degree of its genotypes diversity. The initial population has the same structure so; it is normally that all individuals are speciated to the same species.
- 3- Evaluate the performance of the population individuals on the training data set.
- 4- Configure the ensemble by selecting the best individual of each species; calculate the weight of each ensemble member depending on the characteristic of its species. Compute the ensemble output as a weighted sum of the output of each member.
- 5- Stop evolution if the maximum number of generations or the best fitness is reached. Otherwise go to the next step.
- 6- Generate new population from the current population by using crossover and mutation.
- 7- Go to step 2.

9. EXPERIMENTAL RESULTS

The proposed system has been tested on three benchmark data sets; Iris data set, breast cancer data set, and diabetes data set which are available by the UCI machine learning repository. Iris data set contains 150 instances with 4 numeric attributes, and three classes. The data set is equally distributed with 50 instances for each class. The 150 instances are divided into 60 instances for training, 30 instances for validation, and 30 instances for testing. Breast cancer data set is a two class problem with 699 instances, each instance has 9 attributes and 1 class attribute. The data set is divided into 349 instances for training, 175 instances for validation, and 175 instances for testing. Diabetes data set has 768 instances with 8 numeric attributes which can be classified into positive or negative diabetic. Also in agree with literatures the data set is divided into 384 instances for training, 192 instances for validation, and 192 instances for testing. Each experiment starts with an initial population consists of 100 NN's, each NN has a minimal structure; 4 input nodes and 3 output nodes for iris data set, 9

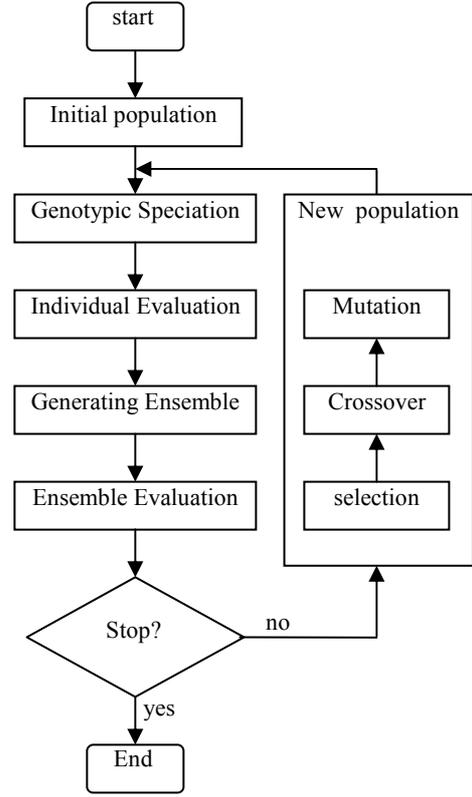


Fig. 2. Flowchart of ensemble evolution algorithm

input nodes and one output node for breast cancer data set, and 8 input nodes and one output node in the case of diabetes data set, no hidden nodes, input nodes are fully connected to output nodes, connections weights are randomly initialized. The maximum number of generations is 200. Genetic operators rates are set as follows: structures crossover rate is 0.8, weights crossover rate is 0.6, add node mutation rate is 0.05, add connection mutation rate is 0.03, weight mutate rate is 0.03, and connection re-enable rate is 0.25.

The evolution continues until the maximum number of generations or the best fitness is reached. During evolution the ensemble parameters (ensemble size and weight of ensemble members) are optimized. At the end of evolution the best ensemble configuration is obtained. Tables 1, 2 and 3 show the configuration of ensemble for iris data set, breast cancer data set, and diabetes data set respectively. The first column shows the member id in the population. The second column refers to species id from which that ensemble member is selected. The weight associated with each member is shown in the third column. The ensemble members weights are obtained with $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 0.9$. The classification rate of each member is shown in the last column. The ensemble size of iris data set is seven members, six members on breast cancer data set and seven members on diabetes data set. The ensemble configuration shown in tables 1, 2, and 3 are the best configuration obtained on ten experiments.

The common error between the ensemble members can be defined as follows:

$E_c = E_1 \cap E_2 \cap \dots \cap E_i$ where E_i is the error set of member i on the data set. On iris training data set the common error is 0, i.e. each individual has different set of error, and on breast cancer training data set is 1, i.e. one pattern is common between all ensemble members error sets. Tables 4, 5, and 6 show the

classification rate on iris data set, breast cancer data set, and diabetes data set for training, validating, and testing respectively.

Table 1. Ensemble configuration (Iris data set)

Individual <i>id</i>	Species <i>id</i>	member weight	Classification rate
1	89	0.1687	0.89
26	91	0.2052	0.97
3	96	0.1524	0.86
4	98	0.1325	0.81
5	102	0.1526	0.85
6	104	0.0948	0.66
7	105	0.0937	0.66

Table 2. Ensemble configuration (Breast cancer data set)

Individual <i>id</i>	Species <i>id</i>	Individual weight	Classification rate
1	9	0.1750	0.9714
2	14	0.1745	0.9714
3	16	0.1745	0.9714
4	29	0.1760	0.9000
82	33	0.1696	0.9857
95	34	0.1304	0.8429

Table 3. Ensemble configuration (Diabetes data set)

Individual <i>id</i>	Species <i>id</i>	Individual weight	Classification rate
1	1	0.1400	0.9601
6	7	0.1416	0.9711
61	10	0.1413	0.9689
9	15	0.1547	0.9751
82	22	0.1383	0.9398
80	24	0.1320	0.8929
77	45	0.1377	0.9391

The ensemble output is calculated as the weighted sum of output of its members. To be consistent with the literature [10] [11] and [12], the shown results are the average on ten experiments. The results of the experiments are comparable to results in [11] on breast cancer data set and diabetes data set. As in [11] and [6], the standard deviation on the training set is always smaller than on the test set. Table 7 shows a comparison between the results of the proposed method and results of [11] from ensemble size point of view for breast cancer data set where the population of neural networks in [11] was speciated with average output with linkage cluster analysis. Although, the results of the proposed method are less than the results obtained in [11] in the case of ensemble size of 15 members, the proposed method results are better than results of [11] in the case of ensemble size of 8 members. In both cases the proposed method has a smaller ensemble size, 6 members. So the computational complexity to get output from the proposed ensemble is less than the case of using method proposed in [11].

Table 4. Ensemble classification rate on Iris data set

	Average	Std.	Max.	Min.
Training	0.9943	0.0118	1.000	0.9789
Validating	0.9518	0.0239	0.9731	0.9261
Testing	0.9918	0.0121	0.9937	0.9685

Table 5. Ensemble classification rate on breast cancer data set

	Average	Std.	Max.	Min.
Training	0.9837	0.0141	0.9976	0.9189
Validating	0.9331	0.0312	0.9631	0.8926
Testing	0.9821	0.0172	0.9934	0.9096

Table 6. Ensemble classification rate on diabetes data set

	Average	Std.	Max.	Min.
Training	0.8038	0.0165	0.8187	0.7896
Validating	0.7901	0.0311	0.8056	0.7721
Testing	0.8022	0.0143	0.8141	0.7846

Table 7. Comparing results of this paper and results of [11] for breast cancer data set

	Proposed method	Results of [11]	
Comb. method	Sum of w. output	Vote, Avg, and Wavg	
Ensemble size	6	15	8
Class. rate	0.9821	0.9829	0.9771

• Comparing proposed method with Bagging and boosting

The proposed method is compared with bagging [18] [19], where the NNs are trained using randomly re-sampled training sets, and boosting where NNs are trained using weighted re-sampled training set based on Arcing method and Ada method. Table 8 compares the classification rates of the proposed methods and other ensemble methods, bagging and boosting, the proposed method for configuring ensembles has a high classification rate as compared to the bagging and boosting methods.

Table 8. Comparing the classification rate of the proposed method, bagging, and boosting.

	Bagging	boosting		Proposed method
		Arc	Ada	
Iris	0.9600	0.9630	0.9610	0.9918
B. cancer	0.9660	0.9620	0.9600	0.9821
Diabetes	0.7720	0.7560	0.7670	0.8022

• Comparing combination methods

The proposed method to combine the output of ensemble members is the weighted sum of outputs. The weights of ensemble members depends on characteristics of their species see Eq (1). Our method is compared to the following combining methods:

1. Average, where the ensemble output equals to the average of its members output. In this case all ensemble members have the same weight. For ensemble of n members, the ensemble output will be:

$$Ens_o = \frac{1}{n} \sum_{i=1}^n m_i \text{ where } m_i \text{ is the output of member } i.$$

2. Weighted average [12], each ensemble member has a weight depending on the error rate of this member as follows:

$$w_i = \frac{1 - E_i}{\sum_{k=1}^n (1 - E_k)}$$

The ensemble output is given by,

$$Ens_o = \frac{1}{n} \sum_{i=1}^n w_i m_i$$

3. Voting, the ensemble output for input x will be class j , if the number of members that support class j is considerably bigger than the number of members that support any other class [12] [21].

These methods are used to combine the output of ensemble configured from the last generation, where the ensemble evolution is stopped. The results are listed in Table 9, the proposed method has a better classification rate than other methods for all data sets. Since the weights are evolved during the training and evolution of neural networks.

Table 9. Comparison of proposed combination method and other combination methods

	Avg.	w. avg.	voting	Proposed method
Iris	0.9573	0.9761	0.9628	0.9918
B. cancer	0.9489	0.9685	0.9734	0.9821
Diabetes	0.7065	0.7410	0.7381	0.8067

10. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a new method to evolve ensemble of neural networks. Both, the population individuals and the ensemble configuration (members, size, and weights) are evolved in the same evolution phase. This method is based on speciation and fitness sharing, where the population is speciated with a new genotype similarity measure. The distance between population individuals is measured in terms of connection genes of their genotypes. The preliminary results of experiments on iris data set breast cancer data set and diabetes data set showed that, the proposed method is able to generate ensembles has better performance than bagging and boosting methods, and comparable performance with smaller ensemble size compared to other methods which are based on fitness sharing. The main contribution of the proposed method is the full benefit from evolutionary algorithms in evolving ensemble as whole. The advantages of this method are that, the interaction between the evolution of the population individuals and the construction of the ensemble, there is no need to initially select the ensemble size or fix the ensemble members weights. The evolution of ensemble is stopped when the maximum fitness is reached or the maximum number of generation is reached, not depending on the performance of the best individual, and the ensemble obtained by this method has smaller size that those obtained by other methods. Our future work will be concentrated on assessing the proposed method on other benchmark data sets available on UCI, like Australian credit card set, and glass data set.

11. REFERENCES

- [1] Y. Liu, X. Yao, and T. Higuchi. "Evolutionary Ensembles with Negative Correlation Learning," IEEE Transactions on Evolutionary Computation, 4(4):380--387, 2000.
- [2] Garcia-Pedrajas, N.; Hervas-Martinez, C.; Ortiz-Boyer, D., "Cooperative coevolution of artificial neural network ensembles for pattern classification," Evolutionary Computation, IEEE Transactions on , vol.9, no.3, pp. 271-302, June 2005.
- [3] X. Yao and Y. Liu, "Making Use of Population Information in Evolutionary Neural Networks," IEEE Transactions on Systems, Man and Cybernetics - Part B: Cybernetics, vol. 28, no. 3, pp. 417--425, 1998.
- [4] Yong Liu, Xin Yao, and Tetsuya Higuchi, "Evolutionary Ensembles with Negative Correlation Learning," IEEE Transactions on Evolutionary Computation, Vol. 4, No. 4, November 2000.
- [5] Chan, Z.S.H.; Kasabov, N., "Fast neural network ensemble learning via negative-correlation data correction," Neural Networks, IEEE Transactions on , vol.16, no.6, pp. 1707-1710, Nov. 2005.
- [6] Abbass, H.A., "Pareto neuro-evolution: constructing ensemble of neural networks using multi-objective optimization," Evolutionary Computation, 2003. CEC '03. The 2003 Congress on , vol.3, no., pp. 2074-2080 Vol.3, 8-12 Dec. 2003.
- [7] Brown, G.; Xin Yao; Wyatt, J.; Wersing, H.; Sendhoff, B., "Exploiting ensemble diversity for automatic feature extraction," Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on , vol.4, no., pp. 1786-1790 vol.4, 18-22 Nov. 2002.
- [8] Jack Y. Yang, Guo-Zheng Li, Li-xin Liu and Mary Q. Yang, "Classification of Brain Glioma by Using Neural Networks Ensemble with Multi-Task Learning," In: Proceedings of The 2007 International Conference on Bioinformatics and Computational Biology (BIOCOMP'07): June 25-28, 2007, Las Vegas.
- [9] Guo-Zheng Li, Tian-Yu Liu, "Improving Generalization Ability of Neural Networks Ensemble with Multi-Task Learning," Journal of Computational Information Systems, 2006, 2(4):1235-1239.
- [10] Joon-Hyun Ahn; Sung-Bac Cho, "Speciated neural networks evolved with fitness sharing technique ," Evolutionary Computation, 2001. Proceedings of the 2001 Congress on , vol.1, no., pp.390-396 vol. 1, 2001.
- [11] K.-J. Kim and S.-B. Cho, "Evolutionary ensemble of diverse artificial neural networks using speciation," Neurocomputing, pp. 398-410, Jan 2007.
- [12] Seung-Ik Lee; Joon-Hyun Ahn; Sung-Bae Cho, "Exploiting diversity of neural ensembles with speciated evolution," Neural Networks, 2001. Proceedings of IJCNN '01. International Joint Conference on , vol.2, no., pp.808-813 vol.2, 2001.
- [13] Yong Liu, and Xin Yao, "Evolving Neural Networks Ensembles by Fitness Sharing," 2006 IEEE Congress on Evolutionary Computation, Canada 2006.
- [14] Joseph Bruce and Risto Miikkulainen, "Evolving Population of Expert Neural Networks," In Proceedings of the 2001 Genetic and Evolutionary Computation Conference.
- [15] Kenneth O. Stanley and Risto Miikkulainen (2002). "Evolving Neural Networks Through Augmenting Topologies," Evolutionary Computation 10(2):99-127, 2002.
- [16] David W. Opitz, Jude W. Shavlik, "Generating Accurate and Diverse Members of a Neural-Network Ensemble," NIPS 1995:535-541.
- [17] Pete Duell, Iris Fermin, and Xin Yao, "Diversity Creation in Local Search for the Evolution of Neural Networks Ensembles," ESANN'2006 Proceedings, 2006.
- [18] H. Sallam, C. S. Regazzoni, Ihab Talkhan, Amir Atiya, "Measuring the Genotype Diversity of Evolvable Neural Networks," in Proceeding of INFOS2008, March 27:29 2008.
- [19] R. Maclin and D. Opitz, "An empirical evaluation of bagging and boosting," In Proceedings of AAAI, 1997.
- [20] D. Opitz and R. Maclin, "Popular ensemble methods: An empirical study," Journal of Artificial Intelligence Research, pages 169--198, 1999.
- [21] Dennis Bahler and Laura Navarro, "Methods for combining heterogeneous sets of classifiers," In 17th Natl. Conf. on Artificial Intelligence (AAAI), Workshop on New Research Problems for Machine Learning, 2000.