

# Evaluation of Distance Measures for Speciated Evolutionary Neural Networks in Pattern Classification Problems

Kyung-Joong Kim<sup>1</sup> and Sung-Bae Cho<sup>2</sup>

<sup>1</sup>Department of Computer Engineering, Sejong University, Seoul, South Korea

<sup>2</sup>Department of Computer Science, Yonsei University, Seoul, South Korea  
kimkj@sejong.ac.kr, sbcho@cs.yonsei.ac.kr

**Abstract.** Recently, evolutionary neural networks are hot topics in a neural network community because of their flexibility and good performance. However, they suffer from a premature convergence problem caused by the genetic drift of evolutionary algorithms. The genetic diversity in a population decreases quickly and it loses an exploration capability. Based on the inspiration of diversity in nature, a number of speciation algorithms are proposed to maintain diverse solutions from the population. One problem arising from this approach is lack of information on the distance measures among neural networks to penalize or discard similar solutions. In this paper, a comparison is conducted for six distance measures (genotypic, phenotypic, and behavioral types) with representative speciation algorithms (fitness sharing and deterministic crowding genetic algorithms) on six UCI benchmark datasets. It shows that the choice of distance measures is important in the neural network evolution.

**Keywords:** Evolutionary Neural Networks, Distance Measures, Fitness Sharing, Deterministic Crowding Genetic Algorithm, UCI Benchmark Datasets.

## 1 Introduction

Recently, there is an increasing demand on the distance measures between two neural networks (NN) in evolutionary neural networks (ENN), clustering, and diversity analysis. 1) In evolutionary neural networks, it is important to exploit the relationships among individual neural networks of a population because they are evolved simultaneously. Their similarity or dissimilarity can be used to accelerate evolution for better performance. 2) From a set of neural networks, it is possible to choose representative neural networks by selecting the best one from each cluster. This is helpful to form a selective ensemble instead of one of all neural networks. 3) Diversity prevents learning algorithms from a premature convergence caused by falling into local optimum. The representative applications of distance measures are evolving an ensemble of neural networks [1], robotics, games and pattern classifications.

The distance measures can be categorized into genotypic, phenotypic, and behavioral distances. Each neural network has a topology and weight parameters for each link. The phenotypic distance is defined on the difference of the architectural property. It is

possible to measure distances of two neural networks with their genotypic representations in evolutionary computation. The behavioral distance is measured based on the output of neurons (input, hidden, and output). In this case, there has to be a set of input data to get the outputs. If the neural network is associated to a specific task, it is possible to use their outcome as a source of measuring. In robotics, each NN is used as a controller of robot and the outcomes are actual trajectories. The distance of the two NNs are compared indirectly by comparing two trajectories from different NNs. However, this distance measure is applicable only for the specific domain.

Speciation is a set of techniques to increase the diversity of population in evolutionary algorithms and representatives are fitness sharing and deterministic crowding genetic algorithm [2]. It forces to maintain diverse solutions by punishing overpopulated solutions or discarding similar one. The distance measuring is an essential part of the algorithms because they use the closeness of solutions to decide whether you give a penalty or discard one. They are widely used in many applications to overcome the shortcomings of simple evolutionary algorithms and outperform other competitors [3][4]. In evolutionary neural networks, the algorithms are widely used to generate diverse neural networks beneficial to form an ensemble. However, its application in ENN is not limited to the ensemble forming.

It is necessary to compare a number of different distance measures for neural networks because there is no consensus on which one is better than others. The most commonly used one is Euclidean distance of weight vectors because it is the simplest one. There are many alternatives and it is worth to compare them in a unified framework to show their performance difference. In this paper, we categorized known distance measures into three categories (phenotypic, genotypic and behavioral) and compare them in an evolutionary neural network framework. In the evolution, the two representative speciation algorithms are used together with one of distance measures. A good distance measure would lead to increase the diversity of population and avoid local optimums. It results in high performance of the evolutionary neural networks from the last generation.

In behavioral distance measures, it is common to use the difference of the outputs from output neurons, but in this work we devise a new measure that exploits outputs from input/hidden/output neurons. This gives much information on the behavior of the neural networks. There are six distance measures: Euclidean distance of genotypic representation (EU), hamming distance of topology (HA), Euclidean distance of outputs (EU\_B), Euclidean of outputs from all neurons (EU\_B\_ALL), Euclidean distance of average outputs (AO), and Kullback-Leibler entropy of outputs (KL). The proposed methods are tested on six UCI benchmark datasets.

## 2 Background

The simplest way to measure distance is Euclidean distance of genotypic representation. Usually, the chromosome for neural network is a vector of real values and it is easy to use Euclidean distance. It is possible to use more complex representation of neural networks with additional information not just weight. In [6], whole genome is composed of two parts (node genes, genome genes). The connection genome has in-node ID, out-node ID, weight, enable/disable, and original historical ancestor of each gene. In this case, the distance is calculated with a simple linear combination of the

matching and mismatching information. In [8][9], authors use the same representation except the historical information. The distance measure is specialized for their representations by exploiting enable/disable information. In [7], a new distance measure is proposed with the consideration of redundancy in representation. Any permutation of the hidden neurons produces the same neural network with a different chromosome representation.  $N!$  different representations exist for a network with  $n$  hidden neurons. In their work, they consider all permutations in calculating distances of two neural networks. This is computationally expensive because  $N!$  combinations are checked.

In behavioral distance measures, the difference of outputs from output neurons is measured with statistical methods. In [5], they propose average output, Pearson correlation and Kullback-Leibler entropy of the outputs. Like genotypic distance measures, Euclidean distance is also used. On the other hands, it is possible to use actual behavior of the neural network to measure the distance. This is dependent on the type of applications of neural networks. If they are used for pattern classification and a robotic problem, the behavior is a vector of class labels classified and trajectories of robot with the neural controller, respectively. Because this is task-dependent, it is not possible to use it for general purpose.

The applications of the distance measures are mainly for evolution because it is a population-based search and there is much interest on the interaction of multiple solutions. [10][11] are exceptional cases and they are not related to evolution. In evolution, the fitness sharing is a dominant application for the distance measures. In many cases, clustering algorithms are used together with speciation and the same distance measures for the speciation are also adopted for the clustering. Interesting applications are diversity analysis of population, distance-based operators, and normalization of solutions.

### 3 Speciated Neural Networks with Distance Measures

A framework for evolving neural networks with speciation is used to compare several distance measures. It evolves topology and weights of neural networks simultaneously. A distance measure is evaluated based on the performance of the best neural network evolved with it. In the speciation, both of the fitness sharing and deterministic crowding algorithms are adopted because the performance of a distance measure could be biased to the choice of a specific speciation method. The best distance measure is decided based on the average performance of the two different speciation methods.

#### 3.1 Representation

In this paper, 1-D vector representation is used to represent a neural network. It is a feed-forward neural network with hidden layers. Usually, the neural network can be represented as a matrix whose entry represents the existence of link and its weight. In this representation, only half of the matrix is filled with real-values and another half is not used. The 1-D vector is constructed from the half of the matrix. Figure 1 shows an example of the 1-D vector representation. The shaded area in the matrix representation is converted to the entry of the 1-D vector. If there is a link between two nodes, it is represented with the weight. Otherwise, it is 0.

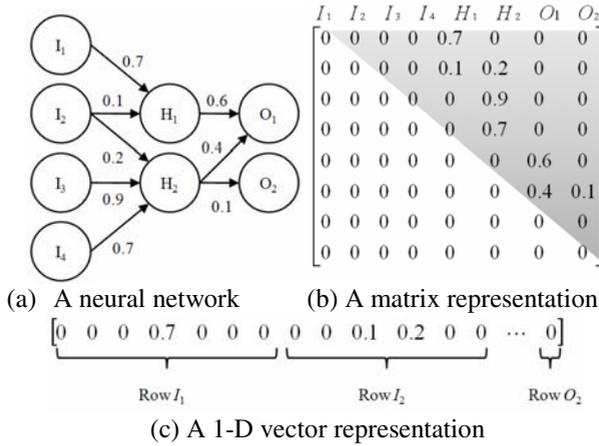


Fig. 1. An example of neural network representation

### 3.2 Speciation Algorithms

Fitness sharing is a method to share its fitness with near neighborhoods. If the population density of the near area of the individual is high, the degree of sharing is increasing and the original fitness of the neural network decreases. The parameter of the algorithm is a sharing radius that decides the boundary of the neighborhood. If the distance between two neural networks is smaller than the radius, they are neighborhood and share fitness. Although there is work on systematic testing on the effect of sharing radius size [14], this is not main issue of our research. In this paper, we set the sharing radius as the half of the average distances of all individuals in an initial population.

Unlike fitness sharing, there is no additional parameter for deterministic crowding genetic algorithm. It is based on the competition between parents and offspring paired to maximize the similarity. Among two similar neural networks, one with better fitness survives to the next generation. Figure 4 summarizes a pseudo code for the deterministic crowding algorithm. It uses the same crossover and mutation with the fitness sharing.

### 3.3 Distance Measures

$N_I$ : The number of input neurons

$N_H$ : The maximum number of hidden neurons

$N_O$ : The number of output neurons

$N$ : The number of total neurons ( $N=N_I+N_H+N_O$ )

$I_{ijk}, H_{ijk}, O_{ijk}$ :  $j$ th input, hidden, and output neuron of  $i$ th neural network for  $k$ th sample

$\phi()$ : Output of a neuron

$M$ : The number of training samples

$V_i$ : A chromosome of  $i$ th neural network (1-D vector)

$L$ : A length of the chromosome ( $L = \frac{N^2 - N}{2}$ )

### 3.3.1 Genotypic Distance

**Euclidean distance of genotypic representation (EU):** This is a simple Euclidean distance of two chromosome vectors.

$$d(a, b) = \sqrt{\sum_{i=1}^L (V_{ai} - V_{bi})^2}$$

### 3.3.2 Phenotypic Distance

**Hamming distance of topology (HA):** This counts only the similarity and dissimilarity of topologies. If only one of the networks has a link, this is counted.

$$d(a, b) = \sum_{i=1}^L h(V_{ai}, V_{bi})$$

$$h(x, y) = \begin{cases} 1 & V_{ai} = 0 \text{ and } V_{bi} \neq 0, V_{ai} \neq 0 \text{ and } V_{bi} = 0 \\ 0 & \text{otherwise} \end{cases}$$

### 3.3.3 Behavioral Distance

**Euclidean distance of outputs (EU\_B):** This calculates the Euclidean distance of outputs from output neurons.

$$d(a, b) = \sqrt{\sum_{k=1}^M \sum_{j=1}^{N_o} (O_{ajk} - O_{bjk})^2}$$

**Euclidean distance of outputs from all neurons (EU\_B\_ALL):** Usually, the outputs from output neurons are used in the behavioral distance but we propose to use outputs from all neurons (input/hidden/output neurons).

$$d(a, b) = \sqrt{\sum_{k=1}^M \left( \sum_{j=1}^{N_I} (I_{ajk} - I_{bjk})^2 + \sum_{j=1}^{N_H} (H_{ajk} - H_{bjk})^2 + \sum_{j=1}^{N_o} (O_{ajk} - O_{bjk})^2 \right)}$$

**Euclidean distance of average outputs (AO):** This is the Euclidean distance of average outputs from each output neuron.

$$d(a, b) = \sqrt{\sum_{j=1}^{N_o} (\bar{O}_{aj} - \bar{O}_{bj})^2}$$

$$\bar{O}_{aj} = \frac{1}{M} \sum_{k=1}^M O_{ajk} \quad \bar{O}_{bj} = \frac{1}{M} \sum_{k=1}^M O_{bjk}$$

**Kullback-Leibler entropy of outputs (KL):** This is proposed in [5]. It is called as relative entropy and modified to be used as a distance measure.

$$d(a, b) = \frac{1}{2} \sum_{k=1}^M \sum_{j=1}^{N_o} \left( O_{ajk} \log \frac{O_{ajk}}{O_{bjk}} + O_{bjk} \log \frac{O_{bjk}}{O_{ajk}} \right)$$

## 4 Experimental Results

The proposed method is evaluated on six UCI benchmark datasets. Table 1 summarizes statistics of the datasets. The number of classes is 2 or 3. The number of attributes is ranged from 4 to 34. The number of samples is from 351 to 1473. In case of a missing entry, it is imputed randomly based on the probability of other entry values. The attribute value is normalized to 0~1. The whole data is separated to training (2/3) and test (1/3) randomly. Table 2 summarizes parameters used in this experiment. The number of input and output neurons of a neural network is the same with the number of attributes and classes of a dataset, respectively. A logistic function is used as a transfer function of the neural network. The result is averaged over 10 runs (each time, the training/test samples are randomly separated). For each run, a seed for random functions is decided based on the current time and the value is used for 12 combinations of settings (2 speciation algorithms  $\times$  6 distance measures). The 12 variations use the same initial population. The fitness function is classification accuracy on the training samples. They are compared in terms of the accuracy on the test samples.

**Table 1.** Summary of dataset statistics

Name	# of classes	# of attributes	# of samples	# of training samples	# of test samples
Balance	3	4	625	417	208
Breast Cancer	2	9	699	466	233
Contraceptive Method Choice	3	9	1473	982	491
Congressional Voting Record	2	16	435	290	145
Ionosphere	2	34	351	234	117
Tic-Tac-Toe	2	9	958	639	319

**Table 2.** Parameters of experiments

Name	Value
Population Size	50
The Maximum Number of Generation	500
Crossover Rate	0.9
Mutation Rate	0.1
The Number of Maximum Hidden Nodes	15

Table 3 summarizes the experimental results of the 6 different distance measures with the fitness sharing algorithm and deterministic crowding algorithm. It shows that the best one (in terms of average accuracy) is different for each dataset and speciation method. Based on the average of 12 cases (2 speciation methods  $\times$  6 datasets) EU\_B\_ALL and AO are the best, and HA is the worst.

**Table 3.** Summarization of results

	EU	HA	EU_B	EU_B_ALL	AO	KL
Balance	<b>87.5 ± 3.0</b>	85.5 ± 4.3	84.1 ± 3.3	87.1 ± 2.3	83.3 ± 4.6	82.4 ± 4.7
Breast Cancer	95.8 ± 1.1	<b>95.9 ± 1.4</b>	95.4 ± 1.8	95.7 ± 0.9	94.7 ± 1.1	94.9 ± 1.4
Contraceptive	43.8 ± 2.6	45.7 ± 1.8	44.5 ± 2.1	44.4 ± 1.8	45.2 ± 1.9	<b>46.1 ± 2.6</b>
Congressional Voting	91.0 ± 3.0	91.2 ± 2.2	91.8 ± 3.1	<b>93.0 ± 2.2</b>	92.8 ± 3.6	90.8 ± 3.9
Ionosphere	80.8 ± 4.7	<b>84.2 ± 3.6</b>	81.0 ± 4.2	81.7 ± 4.8	83.3 ± 5.4	81.7 ± 2.6
Tic-Tac-Toe	68.2 ± 3.6	68.9 ± 2.6	<b>70.6 ± 3.0</b>	69.6 ± 3.0	70.5 ± 1.5	70.2 ± 3.2

(a) Fitness sharing

	EU	HA	EU_B	EU_B_ALL	AO	KL
Balance	85.3 ± 2.9	78.8 ± 4.7	<b>87.6 ± 1.2</b>	86.2 ± 2.3	86.6 ± 2.6	87.4 ± 2.4
Breast Cancer	95.5 ± 0.8	95.4 ± 1.6	95.7 ± 1.3	<b>95.9 ± 1.1</b>	95.6 ± 1.4	95.4 ± 1.0
Contraceptive	46.2 ± 1.1	43.9 ± 1.8	45.8 ± 1.4	45.8 ± 2.0	46.2 ± 1.6	<b>46.6 ± 1.7</b>
Congressional Voting	92.8 ± 2.2	90.4 ± 4.0	93.6 ± 2.5	93.5 ± 2.0	<b>93.8 ± 1.6</b>	93.7 ± 2.5
Ionosphere	85.3 ± 3.7	83.6 ± 3.4	<b>86.2 ± 2.5</b>	85.6 ± 1.9	85.0 ± 4.1	<b>86.2 ± 2.5</b>
Tic-Tac-Toe	<b>71.0 ± 2.4</b>	69.1 ± 2.2	70.9 ± 1.0	69.7 ± 2.3	70.9 ± 1.5	70.8 ± 1.8

(b) Deterministic crowding genetic algorithm

	EU	HA	EU_B	EU_B_ALL	AO	KL
Average	78.6	77.7	78.9	<b>79.0</b>	79.0	78.9

(c) Average results

## 5 Conclusion

In this paper, the distance measures for neural networks are compared in the framework of evolutionary neural networks with speciation. Because the evolutionary system is dependent on how good the distance measures are, several candidates can be evaluated indirectly based on the performance of the final neural network evolved. For more accurate evaluation, two representative speciation algorithms are simultaneously used with six UCI benchmark datasets.

Based on the categorization of distance measures, we compare six distance measures from each group (genotypic, phenotypic and behavioral). In the behavioral distance measure, new one called EU\_B\_ALL is proposed considering outputs from all neurons (input/hidden/output). The experimental results show that behavioral distance measures outperformed than genotypic and phenotypic distance measures.

## Acknowledgements

Co-author Kyung-Joong Kim was supported by Korea Health 21 R&D Project, Ministry for Health, Welfare and Family Affairs (A040163).

## References

- [1] Yao, X., Islam, M.M.: Evolving artificial neural network ensembles. *IEEE Computational Intelligence Magazine* (2008)
- [2] Baeck, T., Fogel, D.B., Michalewicz, Z.: *Advanced algorithms and operations (Evolutionary Computation)*. Taylor & Francis, London (2000)
- [3] Hwang, K.-S., Cho, S.-B.: Improving evolvable hardware by applying the speciation technique. *Applied Soft Computing* 9(1), 254–263 (2008)
- [4] Hong, J.-H., Cho, S.-B.: Efficient huge-scale feature selection with speciated genetic algorithm. *Pattern Recognition Letters* 27(2), 143–150 (2006)
- [5] Ahn, J.-H., Cho, S.-B.: Speciated neural networks evolved with fitness sharing technique. In: *Proceedings of the 2001 Congress on Evolutionary Computation*, pp. 390–396 (2001)
- [6] Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. *Evolutionary Computation* 10(2), 99–127 (2002)
- [7] Jung, S.-C., Moon, B.-R.: Central point crossover for neuro-genetic hybrids. In: Deb, K., et al. (eds.) *GECCO 2004*. LNCS, vol. 3102, pp. 1292–1303. Springer, Heidelberg (2004)
- [8] Sallam, H., Regazzoni, C.S., Talkhan, I., Atiya, A.: Measuring the genotype diversity of evolvable neural networks. In: *Proceedings of the 6th International Conference on Informatics and Systems*, pp. 38–43 (2008)
- [9] Sallam, H., Regazzoni, C.S., Talkhan, I., Atiya, A.: Evolving neural networks ensembles NNEs. In: *2008 IAPR Workshop on Cognitive Information Processing*, pp. 142–147 (2008)
- [10] Ormoneit, D.: A regularization approach to continuous learning with an application to financial derivatives pricing. *Neural Networks* (12), 1405–1412 (1999)
- [11] Li, K., Huang, H.-K., Ye, X.-C., Cui, L.-J.: A selective approach to neural network ensemble based on clustering technology. In: *Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp. 3229–3233 (2004)
- [12] Khare, V., Yao, X.: Artificial speciation of neural network ensembles. In: *Proceedings of the 2002 UK Workshop on Computational Intelligence*, pp. 96–103 (2002)
- [13] Trujillo, L., Olague, G., Lutton, E., de Vega, F.F.: Discovering several robot behaviors through speciation. In: Giacobini, M., Brabazon, A., Cagnoni, S., Di Caro, G.A., Drechsler, R., Ekárt, A., Esparcia-Alcázar, A.I., Farooq, M., Fink, A., McCormack, J., O’Neill, M., Romero, J., Rothlauf, F., Squillero, G., Uyar, A.Ş., Yang, S. (eds.) *EvoWorkshops 2008*. LNCS, vol. 4974, pp. 164–174. Springer, Heidelberg (2008)
- [14] Cioppa, A.D., Stefano, C.D., Marcelli, A.: On the role of population size and niche radius in fitness sharing. *IEEE Transactions on Evolutionary Computation* 8(6), 580–592 (2004)