Diverse Evolutionary Neural Networks Based on Information Theory*

Kyung-Joong Kim and Sung-Bae Cho

Department of Computer Science, Yonsei University
134 Shinchon-dong, Sudaemoon-ku, Seoul 120-749, South Korea
kjkim@cs.yonsei.ac.kr, sbcho@cs.yonsei.ac.kr

Abstract. There is no consensus on measuring distances between two different neural network architectures. Two folds of methods are used for that purpose: Structural and behavioral distance measures. In this paper, we focus on the later one that compares differences based on output responses given the same input. Usually neural network output can be interpreted as a probabilistic function given the input signals if it is normalized to 1. Information theoretic distance measures are widely used to measure distances between two probabilistic distributions. In the framework of evolving diverse neural networks, we adopted information-theoretic distance measures to improve its performance. Experimental results on UCI benchmark dataset show the promising possibility of the approach.

Keywords: Information Theory, Neural Network Distance, Fitness Sharing, Evolutionary Neural Networks, Ensemble**.**

1 Introduction

There is a work using structural difference as distance criteria for neural network [1]. If two neural networks are the same in their topological properties, their behaviors will be the same. However, small deviations in their topological structure result in big different in their behaviors (Figure 1). This makes difficult to use structural difference as a measure of distance in neural networks. Instead of this, it is common to use output response of two neural networks as a measure of distance. In this approach, it is important the way to interpret the output of neural networks. If it is regarded as a numerical value, Euclidean distance and other distance measures can be used to calculate their numerical distance. However, it can be also interpreted as a probability [2][3]. In this view, the input to the neural network is the prior knowledge of conditional probability and the neural network outputs posterior probability.

Previously, mutual information is used to measure distances between two neural networks [4]. In this work, the output of neural network is interpreted as random variables and they attempt to find the model of the random variable's behavior using

* This research was supported by Brain Science and Engineering Research Program sponsored by Korean Ministry of Commerce, Industry and Energy.

M. Ishikawa et al. (Eds.): ICONIP 2007, Part II, LNCS 4985, pp. 1007–1016, 2008.

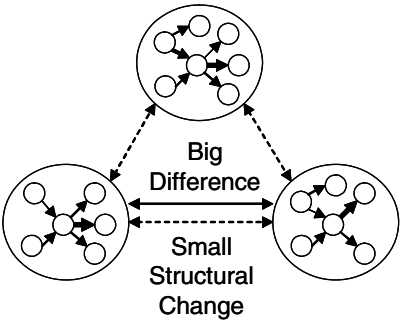

Fig. 1. The problem of using structural deviation as a measure of distance

Gaussian distribution (Figure 2). Calculating mutual information between two Gaussian distributions from neural networks is a way to measure distance. In this paper, we interpret the output of neural network as a posterior probability given the prior knowledge (input pattern). The straightforward approach measuring the distance between two probability distributions is Kullback-Leibler entropy [5] and it is adopted.

Evolving artificial neural network has been one of the hot topics and gained much interest from neural network community [6]. Because it maintains a number of neural networks simultaneously, it is interesting to use them for better performance. If there are more diverse neural networks in the population, more performance gain can be expected when they are used together as an ensemble [7]. Usually, genetic algorithm suffers from the premature convergence and it is called as genetic drift [8]. To avoid the premature convergence, it is important to calculate distances among individuals and use it in a diversity promotion mechanism.

This paper applies the distance idea to the problem of constructing multiple neural networks. It uses genetic algorithms with fitness sharing to generate a population of ANN's that are accurate and diverse. The Kullback-Leibler (KL) entropy method measures the difference between two ANN's using entropy theory. The combination of diverse classifiers is done with the Behavior Knowledge Space (BKS) method [9]. Experimental results on UCI benchmark dataset show that the proposed method can perform well compared to not only the other distance measures but also previous works.

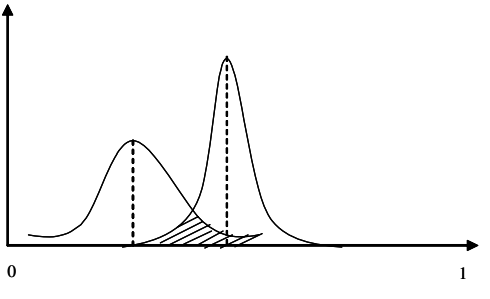

Fig. 2. Interpretation of neural network output as a random variable with Gaussian distribution. The dashed area is shared by the two distributions.

2 Related Works

Table 1 summarizes works related to the Kullback-Leibler (KL) entropy distance measure. This method is not only computationally more efficient than the similarity measure based on mutual information, but also produces comparable accuracy in multi-modal image registrations [10]. Do *et al.* showed that using a consistent estimator of texture model parameters for the feature extraction step, followed by computing the KL distance between the estimated models for the similarity measurement step, is asymptotically optimal in terms of retrieval error probability [11]. Gruner *et al.* proposed a method for quantifying neural response changes in terms of the KL distance between the intensity functions for each stimulus condition [12]. The use of the KL distance to determine the roundness of rock particles has been investigated [13]. In a general framework for self-organizing maps, which store probabilistic models in map units, the distance between a probabilistic model and a data point itself has been defined using the KL distance [14].

A method called the Behavior Knowledge Space aggregates the decisions obtained from the individual classifiers and derives the best final decisions from a statistical point of view [9]. Roli *et al.* analyzed the generalization error of the BKS method and proposed a simple analytical model that relates the error to the sample size [15]. They pointed out that the fusion method could provide very good performance if large, well-distributed datasets were available. Otherwise, over-fitting is likely to occur, and the generalization error quickly increases.

Table 1. Summary of KL entropy distance measure research

Authors	Usage
Chung *et al.* [10]	Multimodal image registration (3D clinical magnetic resonance angiograms)
Do *et al.* [11]	Texture image retrieval (MIT vision texture database)
Gruner *et al.* [12]	Quantifying neural response changes
Drevin [13]	Determining the roundness of rock particles
Hollmen *et al.* [14]	Winner search in self-organizing maps (user profile clustering)

3 Evolutionary NN Ensembles with KL Distance Measure

Figure 3 summarizes the algorithm of evolving multiple neural networks. Each neural network is represented as a matrix. Half of the matrix is used for representing connection of each node and another half is for connection weights. After initializing neural networks, they are trained using backpropagation algorithm using training data. To avoid premature convergence, the training's epoch number is set as small number. The fitness of this evolution is the classification accuracy on validation data set. Because the purpose of this evolution is to generate multiple diverse neural networks for better ensemble performance, diversity is promoted by using fitness sharing scheme (Figure 4).

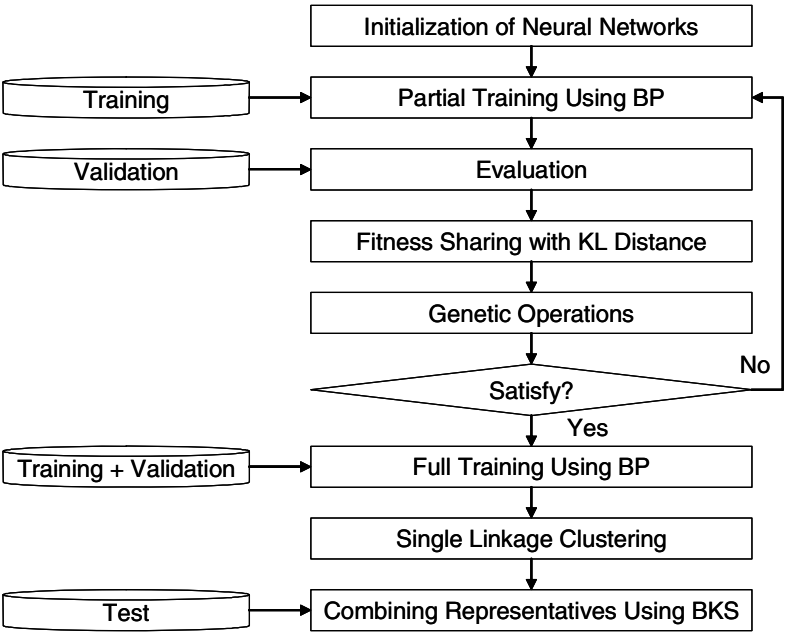

Fig. 3. Flowchart of algorithm

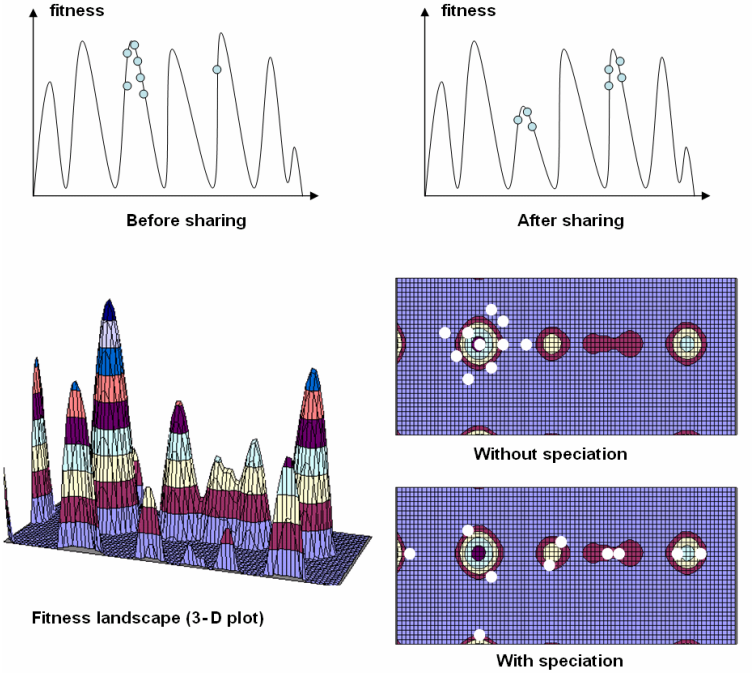

Fig. 4. The effect of fitness sharing in 2D and 3D fitness landscape

Sharing scheme calculates population density in the landscape using distance measures and readjusts their fitness based on the density. If there are many similar individuals with high fitness value, their fitness can be shared by each other and decreases too much. On the other hand, although its fitness is low, the one with low density can survive to the next generation because there is no negative readjustment of the fitness. In this stage, KL distance measure is used to calculate similarity between two neural networks.

Based on the readjusted fitness, roulette-wheel selection, crossover and mutations are sequentially applied to the matrix. Simple matrix genetic operations are used. If there are successful individuals in the population, the evolution stops. Instead of combining all neural networks in the last generation, their clustered results are used to choose the representatives among them. Single linkage clustering is used and the best one for each cluster is combined using BKS method. Finally, the performance of the ensemble is evaluated using test dataset.

3.1 KL Distance Measures

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:

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

Although relative entropy does not satisfy the triangle inequality and is therefore not a true metric, it satisfies many important mathematical properties [16]. 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 [17]. 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.

$$D(p,q) = \frac{1}{2}\sum_k (p_k \log\frac{p_k}{q_k} + q_k \log\frac{q_k}{p_k})$$

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. The total KL distance between the two neural networks is the sum of the KL values for all samples and output nodes. Actually, the integral over all input combinations is not possible, and the summation of the samples is taken.

Then, the similarity of the two ANN's is calculated as follows:

$$D(p,q) = \frac{1}{2}\sum_{j=1}^{m}\sum_{t=1}^{n} (p_{jt} \log\frac{p_{jt}}{q_{jt}} + q_{ij} \log\frac{q_{jt}}{p_{jt}})$$

where p_{jt} means the jth output value of the ANN with respect to the tth training data. The two ANN's are more similar as the symmetric relative entropy decreases. Figure 5 shows an example of probabilistic function approximation using output for training patterns.

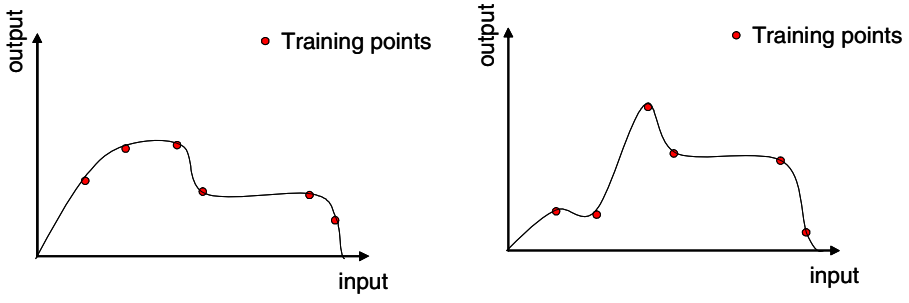

Fig. 5. Approximation of real probabilistic function with the outputs on training dataset

3.2 BKS Combination

To combine the speciated neural networks, we adopted the Behavior Knowledge Space method of the "multinomial" rule [9]. This fusion method is well-known for providing good performance if a large and representative dataset is available [15]. Methods for fusing multiple classifiers can be divided into three levels: the abstract level, the rank level and the measurement level. In abstract-level outputs, every possible combination of abstract-level classifier outputs is regarded as a cell in a look-up table [15]. The BKS table is derived from training and validation sets. 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 this method, the term "cell" is used to represent a space for storing behaviors of the 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. $BKS(e_1(x),...,e_K(x))$ is a cell with index $(e_1(x),...,e_K(x))$.

BKS = a K-dimensional behavior-knowledge space.

$BKS(e_1(x),...,e_K(x))$ = a unit of BKS, where the 1st classifier gives its decision as $e_1(x)$, …, and the Kth classifier gives its decision as $e_K(x)$.

$n_{e_1(x)...e_K(x)}(m)$ = the total number of incoming samples belonging to class m in $BKS(e_1(x),...,e_K(x))$

$T_{e_1(x)...e_K(x)}$ = the total number of incoming samples in $BKS(e_1(x),...,e_K(x))$

$$= \sum_{m=1}^{M} n_{e_1(x),...,e_K(x)}(m)$$

$R_{e_1(x)...e_k(x)}$ = the best representative class of $BKS(e_1(x),...,e_K(x))$
$= \{ j \mid n_{e_1(x),...,e_K(x)}(m) = \max_{1 \le m \le M}(m) \}$

The combination function of BKS is defined as follows:

$$F(e(x)) = \begin{cases} R_{e_1(x)...e_K(x)} & \text{if } T_{e_1(x)...e_K(x)} > 0 \text{ and } \dfrac{n_{e_1(x)...e_K(x)}(R_{e_1(x)...e_K(x)})}{T_{e_1(x)...e_K(x)}} \ge \lambda \\ M+1 & \text{otherwise} \end{cases}$$

λ is a threshold value to decide whether the result is rejected or not. For each class, there is a $\dfrac{n_{e_1(x)...e_K(x)}(m)}{T_{e_1(x)...e_K(x)}} \times 100$ 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.

4 Experimental Results

From UCI benchmark data, breast cancer dataset is downloaded. It is divided into training, validation and test dataset with the ratio of 2:1:1. The number of neural network inputs is the same with number of attributes of the dataset. The population size is set as small value to minimize computational cost. Crossover and mutation rates are set from empirical trial-and-error. The experimental results are the average of 10 runs. Table 2 summarizes the parameters and settings of this experiment.

Table 2. Parameters of experiment

Dataset Name	Breast Cancer
# of classes	2
# of inputs in NN	9
Training/Validation/Test	349/179/175
Population size	20
Crossover rate	0.3
Mutation rate	0.1
# of runs	10

Figure 6 shows the prediction accuracy on test dataset of the proposed methods and other works (EPNET [18]). Average output (AO) and Pearson correlation (PC) measures are used for the comparison. Average output measures Euclidean distances between two average values of output values from the two neural networks. Pearson Correlation also uses the average and standard deviation of each output neuron's

output. Table 3 summarizes statistical test results among the best three methods (BKS+KL, BKS+AO, BKS+PC). The statistical test is done by using t-test. T-value is calculated using the below formula.

$$t = \frac{\mu_A - \mu_B}{\sqrt{\frac{\sigma_A^2}{N_A} + \frac{\sigma_B^2}{N_B}}}$$

where, μ_A represents the average test accuracy of neural network A and σ_A is standard deviation of the multiple runs. N_A is the number of runs of the experiment. In this case, N_A is 10. If t-value is derived from the averages and standard deviation of the two methods, it is compared with the value in t-table. Degree of freedom is $N_A + N_B - 2$. If the t-value is larger than the value in the table, it is statistically significant. Table 3 summarizes the statistical significance test results among the best three methods. It shows that the BKS+KL method performs better than other two methods with statistical significance. However, the difference between BKS+AO and BKS+PC is not statistically significant.

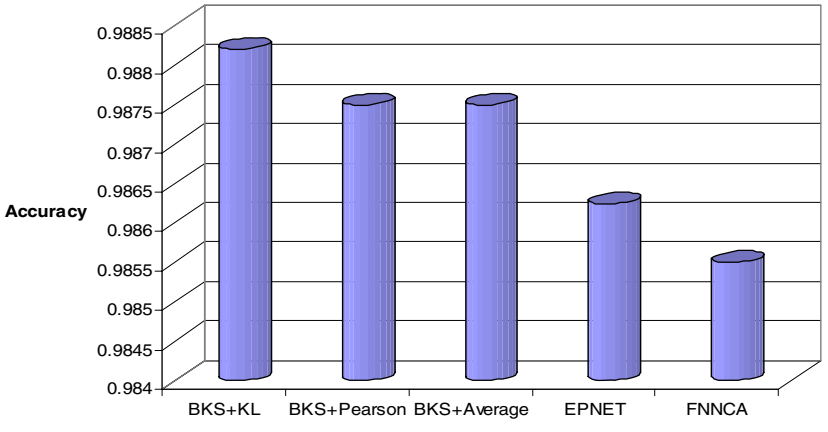

Fig. 6. Comparison with other works

Table 3. Statistical t-test (p=0.1) (AO=Average Output, PC=Pearson Correlation) (+: statistically significant)

	BKS+AO	BKS+PC	BKS+KL
BKS+AO		-	+
BKS+PC	-		+
BKS+KL	+	+	

5 Concluding Remarks

In this paper, the outputs of neural networks are interpreted as posterior probability and the difference between two models are calculated using Kullback-Leibler entropy measure. It is applied to the evolutionary neural ensemble framework to promote diversity in the evolutionary process. Experimental results on UCI benchmark dataset shows that the proposed methods perform better than other candidates with statistical significance. As a future work, it is required to evaluate the method to the other datasets. Also, it is interesting to find other applications of this distance measures.

References

[1] Stanely, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary Computation 10(2), 99–127 (2002)
[2] Richard, M.D., Lippmann, R.P.: Neural network classifiers estimate Bayesian a posteriori probabilities. Neural Computation 3, 461–483 (1991)
[3] Lippmann, R.P.: Neural networks, Bayesian a posteriori probabilities and pattern classification. From Statistics to Neural Networks-Theory and Pattern Recognition Applications (1994)
[4] Liu, Y., Yao, X.: Learning and evolution by minimization of mutual information. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 495–504. Springer, Heidelberg (2002)
[5] Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. 22, 79–86 (1951)
[6] Yao, X.: Evolving artificial neural networks. Proceedings of the IEEE 87(9), 1423–1447 (1999)
[7] Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity creation methods: A survey and categorization. Information Fusion 6, 5–20 (2005)
[8] Rogers, A., Prügel-Bennett, A.: Genetic drift in genetic algorithm selection schemes. IEEE Transactions on Evolutionary Computation 3(4), 298–303 (1999)
[9] Huang, Y.S., Suen, C.Y.: Recognition of unconstrained handwritten numerals. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(1), 90–94 (1995)
[10] Chung, A.C.S., Wells III, W.M., et al.: Multi-modal Image Registration by Minimising Kullback-Leibler Distance. In: Dohi, T., Kikinis, R. (eds.) MICCAI 2002. LNCS, vol. 2489, pp. 525–532. Springer, Heidelberg (2002)
[11] Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. IEEE Transactions on Image Processing 11(2), 146–158 (2002)
[12] Gruner, C.M., Johnson, D.H.: Calculation of the Kullback-Leibler distance between point process models. In: International Conference on Acoustics, Speech, and Signal Processing, pp. 3437–3440 (2001)
[13] Drevin, G.R.: Using entropy to determine the roundness of rock particles. In: 5th International Conference on Signal Processing, pp. 1399–1404 (2000)
[14] Hollmen, J., Tresp, V., Simula, O.: A self-organizing map for clustering probabilistic models. In: Ninth International Conference on Artificial Neural Networks, vol. 2, pp. 946–951 (1999)

[15] Raudys, S., Roli, F.: The behavior knowledge space fusion method: Analysis of generalization error and strategies for performance improvement. In: Windeatt, T., Roli, F. (eds.) MCS 2003. LNCS, vol. 2709, pp. 55–64. Springer, Heidelberg (2003)
[16] Cover, T.M., Thomas, J.A.: Elements of Information Theory. Wiley-Interscience, Chichester (1991)
[17] Qian, H.: Relative entropy: Free energy associated with equilibrium fluctuations and nonequilibrium deviations. Physical Review E 63, 042103/1–042103/4 (2001)
[18] Yao, X., Liu, Y.: A new evolutionary system for evolving artificial neural networks. IEEE Transactions on Neural Networks 8(3), 694–713 (1997)