Actively Searching for Committees of RBF Networks Using Bayesian Evolutionary Computation

Je-Gun Joung
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
jgjoung@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
btzhang@scai.snu.ac.kr

Abstract- Committee machines are known to improve generalization performance by combining the predictions of many different individual learners. Evolutionary algorithms generate multiple models that can be combined to build a committee machine. This paper uses Bayesian evolutionary algorithms (BEAs) as a solution to evolve individual learners and build a committee machine. BEAs are based on the Bayesian evolutionary framework in which evolutionary computation is the process updating repeatedly the posterior distribution of a population to find an individual with the maximum posteriori probability. BEAs evolve the number of centroids and the centroids' positions and width for RBF networks which are individual learners, and then the algorithms find an optimal committee from many different individuals. Empirical results show convergence characteristic and accuracy.

1 Introduction

Committee machines are a method for improving generalization performance of individual by making decisions of several models. Recently, a number of committee machines have presented methods combining learners learned by means of evolutionary algorithms [1][2][3][4]. A theoretical foundation for this trend is that evolutionary algorithms are pop-based distributed search methods that produce a variety of individuals for many generations. In other words, because diverse individuals are created through evolution, they can provide a rich resource for building committee machines.

Most evolutionary algorithms are stochastic search methods that mimic the metaphor of natural biological evolution such as crossover or mutation. However, Bayesian evolutionary algorithms (BEAs)[5] consider a probability distribution instead of general genetic operator, and use global information contained in the population, instead of using local information of individuals, and In Bayesian evolutionary algorithms, evolutionary computation is formulated as a probabilistic process of discovering an individual with the maximum posteriori probability. BEA starts with a population of individuals drawn from the prior distribution, and iteratively generates a new population by estimating the posterior fitness distribution of parent individuals and then sampling from the distribution offspring individuals by using probabilistic operators.

The diverse structure of committee members is essential in building committee machines. Because Bayesian evolutionary algorithms use global information contained in the population, the population evolved by Bayesian evolutionary algorithms may offer good conditions to build committee machines.

In this paper, we present a method evolving topology and parameter (center, width) of RBF networks as well as searching optimal committee by means of the probabilistic evolutionary algorithms called BEA. In Section 2, we describe the basic concept about committee machines and evolutionary method for building committee machines. Section 3 describes the basic concept about RBF networks and the Bayesian approach to RBF Networks. Section 4 presents the probabilistic method for building the best committee. Section 5 reports experimental results on learning problems taken from sunspot time series data set. Section 6 contains conclusions.

2 Building Committees of Individual Solutions

The basic idea behind the committee machine approach is to fuse knowledge acquired by individual experts to arrive at an overall decision that is supposedly superior to that attainable by any one of them acting alone [6]. Committee machines can be built in two different ways. One is to use a static structure. This is known generally as an ensemble method. Here, the input variable is not involved in combining committee members. Examples include ensemble averaging [7] and boosting [8]. The other method for building committees is to use a dynamic structure. This includes combining local experts such as mixtures of experts [9]. Here input variable is directly involved in the combining mechanism that uses an integrating unit, such as a gating network adjusting the weights of committee members according to input. Most studies on committee machines have been based on neural networks [7, 10], decision trees [11], and statistical methods [12].

Some authors have used evolutionary algorithms for committee machines. Evolutionary algorithms generate a number of individuals during evolution. Most methods select the single best solution, discarding all remaining individuals generated during the evolution. However, the individuals evolved can be better utilized if they are combined to build commit-

0-7803-6657-3/01/$10.00 ©2001 IEEE

tees. Opitz and Shavlik [1] presented the ADDEMUP method that uses genetic algorithms to search for a correct and diverse population of neural networks to be used in the ensemble. It has an objective function that measures both the accuracy of the network and the disagreement of that network with respect to the other members of the set. Yao and Liu [3] experimented with a variety of combination methods to integrate multilayer perceptrons that were evolved by evolutionary programming and a modified back-propagation algorithm. They also try to find a committee of variable size using a genetic algorithm. However, only the individuals in the final generation were used as candidates for the committee members. Neri and Giordana [13] introduced universal suffrage selection that chooses a suitable concept description to evolve partial concept descriptions as a whole. Here the concept description can be viewed as a committee member and universal suffrage selection may be regarded as a method for selecting committee members.

3 Bayesian Evolutionary Learning of RBF-NN

3.1 RBF Networks

A radial basis function (RBF) network is a popular alternative to the multilayer perceptrons (MLP). The construction of a radial-basis function (RBF) network involves three layers with entirely different roles. The first layer is composed of input nodes whose member is equal to the dimension m . The second layer is a hidden layer, composed of nonlinear units with a radial-basis activation function, and the third, output layer performs the network response to an input signal and conventionally consists of linear neurons.

The RBF network used to approximate an unknown function f can be described by an affine mapping

$$f(\mathbf{x}) = \sum_{i=1}^{k} w_i \varphi_i(\mathbf{x}) \tag{1}$$

where k is the number of the hidden node, and the i-th radial basis function φ_i are usually concerned with Gaussian function.

The fast technique of training a m-k-1 RBF network contains two steps. First, the centers $c_j, j = 1, \ldots, K$ of activation functions in hidden-layer are chosen at the input space for which training patterns $\{\mathbf{x}, \mathbf{d}\}_i, i = 1, \ldots, N$ are known, and then the widths $r_j, j = 1, \ldots, k$ of the activation function are set. Then the weights $\mathbf{w}, j = 1, \ldots, k$ are computed. The latter task is reduced to solving matrix equation,

$$\mathbf{\Phi w} = \mathbf{d} \tag{2}$$

Where $\mathbf{\Phi}$is the interpolation matrix, $\mathbf{w}$ is the vector of the synaptic weights of an output-layer unit and $\mathbf{d}$ is the vector of the output patterns, respectively.

$$\mathbf{d} = [d_1, d_2, ..., d_N]^T \tag{3}$$

$$\mathbf{\Phi} = \begin{bmatrix} \varphi_{11} & \varphi_{12} & \cdots & \varphi_{1N} \\ \varphi_{21} & \varphi_{22} & \cdots & \varphi_{2N} \\ \vdots & \vdots & & \vdots \\ \varphi_{MN} & \varphi_{MN} & \cdots & \varphi_{MN} \end{bmatrix} \tag{4}$$

$$\mathbf{w} = [w_1, w_2, \ldots, w_N]^T \tag{5}$$

where

$$\varphi_{ij} = \varphi(\|\mathbf{x}_j - \mathbf{c}_i\|) = \exp\left(-\frac{\|\mathbf{x}_j - \mathbf{c}_i\|^2}{2r^2}\right) \tag{6}$$

Here $\mathbf{x}_i, j = 1, \cdots, N$ is the vector of a neural networks input pattern and c_i is the center of basis function. If we assume fixed radial-basis functions, the only parameters that would need to be learned in this approach are the linear weights in the output layer of the network. Generally, A straight forward procedure for doing this is to use the pseduoinverse method[15].

$$\mathbf{w} = \mathbf{\Phi}^{+}\mathbf{d}, \tag{7}$$

where $\mathbf{\Phi}^{+}$ is the pseduoinverse of $\mathbf{\Phi}$. Like other neural network model, RBF networks is affected by their topology. Too many centroids leads to over-fiting, while too few centroids may prove insufficient to capture intrinsic class divisions adequately.

3.2 Bayesian Evolutionary Optimization of RBF Networks

Bayesian evolutionary algorithm works by initializing a population of RBF networks and iteratively producing the next generation of fitter networks. We use symbol $\mathcal{A}(g)$ to denote the population at generation g:

$$\mathcal{A}(g) = \{A_g^i\}_M^{i=1}, \tag{8}$$

where A_g^i denotes the ith network at generation g, and M is the population size. New generations are produced repeatedly until the maximum number of generation g_{max} is reached or some other termination condition is satisfied.

The goodness of a network is measured in terms of a set D of training data and networks can be considered as a model of the unknown process f generating the data.

In the Bayesian approach, the best network is defined as the most probable model of the data with respect to the prior knowledge on the problem domain. More formally, let θ denote the parameter vector for the model, let $\pi(\theta)$ be the prior probability distribution for the models and $p(D|\theta)$ the likelihood of the model for the data $D = \{(\mathbf{x}_c, y_c) \mid c = 1, ..., N\}$. Then, using Bayes formula the posterior probability $\pi(\theta|D)$ of model θ is given as

$$\pi(\theta|D) = \frac{p(D|\theta)\pi(\theta)}{p(D)}, \tag{9}$$

1. **(Prior distribution)** Initialize RBF networks A_1^i for $i = 1, \ldots, M$ according to $P_0(A)$. Set generation count $g \leftarrow 1$.
2. **(Posterior distribution)** Estimate the posterior probabilities $P_i(g) = P_g(A_g^i|D)$ for each A_g^i, $i = 1, \ldots, M$.
3. **(Model variation)** Generate offspring models A_{g+1}^i for $i = 1, \ldots, M$ with distribution.
4. **(Model selection)** Select M models from the offspring population $\mathcal{A}'(g)$ to build the next generation $\mathcal{A}(g+1)$ of parent models.
5. Set $g \leftarrow g + 1$. If $g \leq g_{max}$, go to Step.

Figure 1: Outline of the Bayesian evolutionary algorithm for RBF networks.

where $p(D)$ is a normalizing constant.

To learn the fittest model by the BEAs, we first define the probability distributions of RBF network model from data. A RBF network model is parameterized as $\theta = (k, \mathbf{c}, \mathbf{r})$, where k is the number of hidden nodes, $\mathbf{c}$ and $\mathbf{r}$ are the center vector and the width vector, respectively. The posterior probability of a RBF network θ is written as

$$\begin{aligned} \pi(\theta|D) &\propto p(D|k,c,r)\pi(\theta) \\ &= p(D|k,c,r)\pi(k,c,r). \end{aligned} \tag{10}$$

Given the training data, the model A can represent the following input-output mapping:

$$y_i = f(\mathbf{x}_c; \theta) + \epsilon. \tag{11}$$

Here, the noise ϵ is assumed to be Gaussian with mean zero and standard deviation σ. If we additionally assume that data items are independent of each other, then the likelihood of the RBF network can be expessed as follows

$$\begin{aligned} & p(D|k,c,r) \\ &= \prod_{i=1}^{N} (2\pi\sigma)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(y_t - f(\mathbf{x};k,c,r))^2\right) \\ &= (2\pi\sigma)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(y_t - f(\mathbf{x};k,c,r))^2\right) \end{aligned} \tag{12}$$

We assume that the number of hidden nodes in the RBF network is distributed according to following Poisson distribution:

$$\pi(k-1) = \frac{\lambda^{k-1}\exp(-\lambda)}{k-1}. \tag{13}$$

where $k = 1, 2, 3, \cdots$, so the number of hidden node can be selected over 1.

If we assume that centers are independent of each parameter, then prior probability for centers can be expressed as follows

$$\pi(c) = (2\pi\sigma)^{-\frac{1}{2}} \exp\left(-\frac{c^2}{2\sigma^2}\right). \tag{14}$$

We can also define the following prior probability for widths of the network as follows

$$\pi(r) = (2\pi\sigma)^{-\frac{1}{2}} \exp\left(-\frac{r^2}{2\sigma^2}\right). \tag{15}$$

Substituting Equations, we obtain the following posterior probability for the network:

$$\begin{aligned} & \pi(\theta|D) \propto p(D|k,c,r)\pi(k,c,r) \\ &= \left[(2\pi\sigma)^{-\frac{N}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(y_t - f(\mathbf{x};k,c,r))^2\right)\right] \\ &\times \left[(2\pi\sigma)^{-\frac{1}{2}} \exp\left(-\frac{c^2}{2\sigma^2}\right)\right]\left[(2\pi\sigma)^{-\frac{1}{2}} \exp\left(-\frac{r^2}{2\sigma^2}\right)\right] \\ &\times \left[\frac{\lambda^{k-1}\exp(-\lambda)}{(k-1)!}\right], \end{aligned} \tag{16}$$

where the $\theta = (k, c, \sigma)$ is the parameter vector for the RBF network.

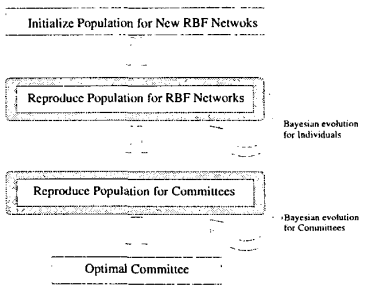

Figure 2: The comprehensive structure of Bayesian approach of evolutionary optimization for individuals and committees.

4 Building Committees by Probabilistic Evolution

We consider a Bayesian evolutionary learning procedure that separates into two steps in order to build committee machines. Fig. 2 shows simple structure for this procedure. In the first stage, Bayesian evolutionary algorithm initializes a population of RBF networks and iteratively produces the next generation of fitter networks. When algorithm reaches termination condition, it move to the next stage evolving the committee. In the latter stage, the goal is to find the optimal size and member of the committee.

From the first stage, The population with size M of RBF networks in the maximum number G of generations is generated as follows

$$\mathcal{A}(G) = \{A_i(G)\}_{i=1}^{M}, \tag{17}$$

where M is the size of the population. The populations generated during the maximum number G constitute the space of candidate individuals for committees and the population of committees can be presented as follows

$$V(g) = \{V_l(g)\}_{l=1}^{L}, \tag{18}$$

Here each committee V_l consists of members v_j as follows

$$V_l = \{v_{l_1}, \ldots, v_{l_m}\},\ v_j \in A(G). \tag{19}$$

The committee size m has a different value for each committee V_l. In this paper, we limit the maximum size to 20. The fitness of committee V_l is defined as following

$$R(V_l) = E(V_l) + \beta C(V_l), \tag{20}$$

where β defines the tradeoff between accuracy and complexity. The error $E(V_l)$ of committee l is the total error on D made by the weighted average of committee members. If the type of the problem is classification, threshold is used for the weighted average:

$$E(V_l) = \sum_{c=1}^{N} \sum_{j=1}^{l} w_{lj} v_{lj} (\mathbf{x_c}) - y_c. \tag{21}$$

Here w_{lj} is the weight of the jth member of the lth committee and satisfies the condition $\sum w_{lj} = 1$. In our approach, we use the generalize ensemble method (GEM) as combining method [14]. The complexity of a committee is defined as the committee size m_l divided by the training set size N:

$$C(V_l) = (m_l/N)\,. \tag{22}$$

Each committee member $A_i(G)$ is selected with the probability as follows

$$P(A_i(G)) = \frac{\exp\left(F(A_i(G))/T\right)}{\sum_{j=1}^{S} \exp\left(F(A_j(G))/T\right)}. \tag{23}$$

where T is a constant for adjusting the difference between the fitness of the candidate. Equation (23) says that the candidate with the high fitness is selected more frequently. In fact, the useful individual for building committee may exist at early generation. So this selection scheme does not exclude the possibility of selecting the candidate with low fitness.

The size of the search space for the optimal committee V^* is $(2^S - 1)$ for pool size S. This is a large number. A simple method to reduce search time is to use a strategy that concentrates on the search space of high performance. Equation (20) expresses that the complexity term $C(V_l)$ prevents from increasing the size of committee.

The search for optimal committees can be formulated as a Bayesian inference problem. Here, we show how the committee selection can be guided by probabilistic models. Let $P(V_l)$ denote the prior probability of committee V_l. Once we observe the data D, the likelihood $P(D|V_l)$ of the committee can be computed. Bayes rule provides a method for combining the prior and likelihood to obtain the posterior probability $P(V_l|D)$ of the committees:

$$P(V_l|D) = \frac{P(D|V_l)P(V_l)}{P(D)}. \tag{24}$$

Each committee l contains members as many as m_l. We assume m_l follows Poison distribution.

$$\begin{aligned} P(V_l|D) &\propto P(D|V_l)P(V_l) && (25)\\ &= P(D|\mathbf{v}, m)P(\mathbf{v}, m) && (26)\\ &= P(D|\mathbf{v}, m)P(\mathbf{v}|m)P(m). && (27) \end{aligned}$$

Where $P(l|\lambda)$ is a Poisson distribution. Here prior distribution of the size m_l of committee l, the prior distribution of mean value $P(\lambda)$ can be written with respect to the fitness for the best committee with size m by

$$P(\lambda) = \frac{R(V_{best}^{\lambda}(g-1))}{\sum_{m=2}^{L} R(V_{best}^{m})(g-1)}, \tag{28}$$

where $V_{best}^{\lambda}(g-1)$ is the best committee with size λ at generation $g-1$. Through this adaptation, the search for an optimal size focuses more on the regions where the fitter committees were frequently generated.

RBF Neural networks	
max number of generations	100
population size	100
initial prior size of hidden node	10
maximum size of hidden node	30
Committees	
max number of generations	20
population size	50
initial prior size of committee	7
maximum size of committee	20

Table 1: Setting of the parameters.

5 Empirical Results

5.1 Experimental Setup

The method was applied to the Wolfe sunspot data [16]. The time series was created from the number of sunspots observed each calendar year from 1700 to the present. This data has been used in several other neural network prediction studies [17]. Following previous studies, we used the period from the year 1700 to 1920 as the training set while the period from 1921 to 1950 was used as the test set. We used eight previous values, $x_{t-8}, x_{t-7}, ..., x_{t-1}$. The input attributes of all data sets were linearly rescaled into the interval $[0.1, 0.9]$.

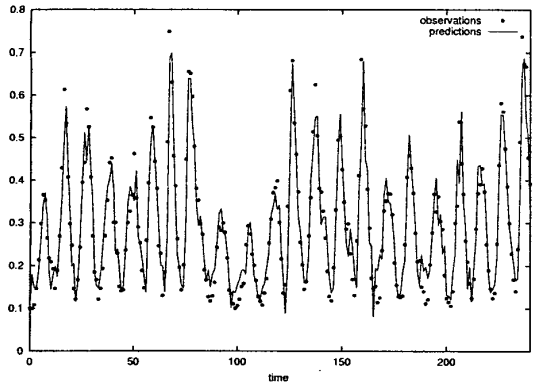

Figure 3: The sunspot data (black points) and the one year ahead prediction.

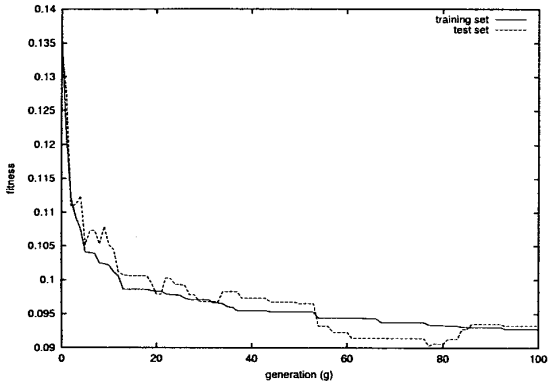

Figure 4: The fitness vs. generation for evolving RBF networks

The fitness of an individual in the population for sunspot time series prediction is Normalized Mean Squared Error (MSE). 1 shows some parameters for learning RBF-NN and Committee. The centers, widths and weights of RBF-NN are randomly set.

5.2 Experimental Results

Figure 3 shows the one year ahead prediction for the sunspot time series data. This result is prediction of the best committee. Algorithm performs well in the time series prediction.

Figure 4 and 5 plot the fitness of the best RBF network and committee as generation go on, respectively. The performance for committee of RBF networks surpasses one of the best RBF network generated in the last generation.

Figure 6 plots the frequency for the number of hidden node as generation go on. Typically, the number of hidden node tends to increase quickly. It can be different in character for problem domain.

Also, Figure 7 shows the frequency for committee size as generation goes on. The high frequency for committee size is

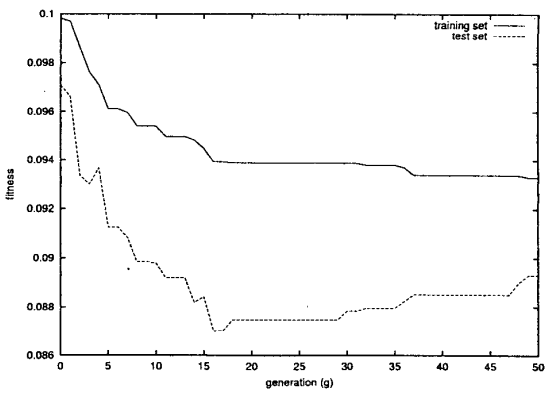

Figure 5: The fitness vs. generation for evolving committees of RBF networks

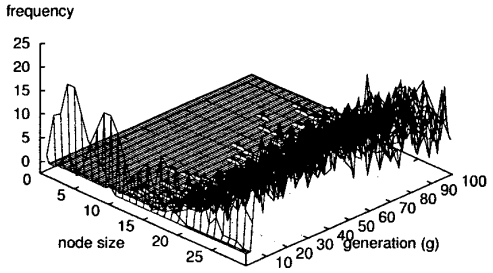

Figure 6: The distribution on the number of node vs. generation for evolving RBF networks

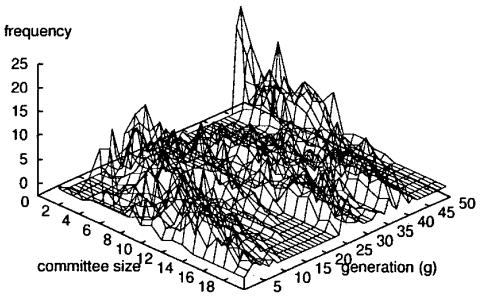

Figure 7: The distribution on the number of node vs. generation for committees of RBF networks

Algorithm	Prediction error (NMSE)	
	Traing set	Test set
MLP	0.078	0.104
Bayeian evolutionary RBF-NN	0.092	0.093
Committee of RBF-NN	0.093	0.089

Table 2: Comparison of one-step prediction errors for the sunspot time series.

between 2 and 6.

Table 2 compares the average generalization error for Multi-Layered Perceptron (MLP) neural networks and our method. Here the result given by MLP was used as a baseline and our result is over 10 run.

6 Conclusions and Future Work

This paper presents an approach to search an optimal committee of RBF networks using Bayseian approach. The probabilistic approach allows the evolution to be guided more effectively by using principled specification of prior knowledge. The view point of this paper is to be formulated comprehensively a scheme evolving population for learners and their committees by using Bayseian framework. For RBF networks, committee machines can be made more robust by combining multiple networks instead of a single network in an environment with noise especially. In the future, there will be several important topics in studying evolutionary method to build committee adequately.

Acknowledgments

This research was supported by the Korea Ministry of Science and Technology through KISTEP under Grant BR-2-1-G-06, and by the Korea Ministry of Education under the BK21 Program.

Bibliography

[1] Opitz, W., Shavlik, J.W. (1996) "Actively Searching for an Effective Neural-Network Ensemble," *Connection Science*, 8, pp. 337-353.

[2] Zhang, B.-T., Joung, J.-G. (1997) "Enhancing Robustness of Genetic Programming at the Species Level," *Genetic Programming Conference (GP-97), Morgan Kaufmann*, pp. 336-342.

[3] Yao, X., Liu, Y. (1998) "Making Use of Population Information in Evolutionary Artificial Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics*, 28B(2), pp. 417-425.

[4] Zhang, B.-T., Joung, J.-G. (1999) "Time Series Prediction Using Committee Machines of Evolutionary Neural Trees," *Proceedings of the Congress on Evolutionary Computation*, 1, pp. 281-286.

[5] Zhang, B.-T. (1999) "A Bayesian framework for evolutionary computation,"In *Proc. 1999 Congress on Evolutionary Computation* (CEC99), IEEE Press, pp. 722-727.

[6] Haykin, S. (1994) *Neural Networks, a Comprehensive Foundation*, Prentice Hall.

[7] Hansen, L., Salamon, P. (1990) "Neural Network Ensembles," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, pp. 993-1001.

[8] Drucker, H., Cortes, C., Jackel, L.D., LeCun, Y., Vapnik, V. (1994) "Boosting and Other Ensemble Methods, *Neural Computation*," 6(6), pp. 1289-1301.

[9] Jacobs, R.A. (1997) "Bias/variance Analyses of Mixture-of-Experts Architectures," *Neural computation*, 9, pp. 369-383.

[10] Hashem, S. (1997) "Optimal Linear Combinations of Neural Networks, *Neural Networks*," 10(4), pp. 599-614.

[11] Opitz, D., Maclin, R. (1999) "Popular Ensemble Methods: An Emiprial Study," *Journal of Artificial Intelligence Research*, 11, pp. 169-198.

[12] Lemm, J.C. (1999) "Mixtures of Gaussian Process Priors," *The Ninth International Conference on Artificial Neural Networks (ICANN 99).*

[13] Neri, F., Giordana, A. (1995) "A Parallel Genetic Algorithm for Concept Learning." *The Sixth international Conference on Genetic Algorithms*, pp. 436-443.

[14] Perron, M.P., Cooper, L.N. (1991) "When Network for Function Interpolation," *Neural Computation*, 3, pp. 213-225.

[15] Broomhead, D.S., D. Lowe. (1988) "Multivariable functional interpolation and adaptive networks," *Complex Systems*, **2**, pp.321-355.

[16] A.S. Weigend, D.E Rummelhart, and B.A Huberman (1990) "Back-propagation, weight elimination and time series prediction," *Proc. 1990 Connectionist Models Summer School*, pp. 105-116.

[17] N. Aerrabotu, G. Tagliarini, and E. Page (1997) "Ensemble encoding for time series forecasting with MLP networks," *Applications and Science of Artificial Neural Networks: Proc. SPIE Volume 3077*, pp. 84-89.