Available online at www.sciencedirect.com
ScienceDirect
Neurocomputing 70 (2006) 155–163
NEUROCOMPUTING
www.elsevier.com/locate/neucom

A novel mixture of experts model based on cooperative coevolution

Minh Ha Nguyen[a,*], Hussein A. Abbass[a], Robert I. Mckay[b]

[a] *Artificial Life and Adaptive Robotics (A.L.A.R.) Lab, The ARC Centre for Complex Systems, School of Information Technology and Electrical Engineering, Australian Defence Force Academy, University of New South Wales, Canberra, ACT 2600, Australia*

[b] *School of Computer Science and Engineering, Seoul National University San 56-1, Sinlim-dong, Gwanak-gu, Seoul 151-744, Republic of Korea*

Received 8 December 2005; received in revised form 6 February 2006; accepted 17 April 2006
Communicated by K. Li
Available online 4 August 2006

Abstract

Combining several suitable neural networks can enhance the generalization performance of the group when compared to a single network alone. However, it remains a largely open question, how best to build a suitable combination of individuals. Jacobs and his colleagues proposed the mixture of experts (ME) model, in which a set of neural networks are trained together with a gate network. This tight coupling mechanism enables the system to (i) encourage diversity between the individual neural networks by specializing them in different regions of the input space and (ii) allow for a "good" combination weights of the ensemble members to emerge by training the gate, which computes the dynamic weights together with the classifiers.

In this paper, we have wrapped a cooperative coevolutionary (CC) algorithm around the basic ME model. This CC layer allows better exploration of the weight space, and hence, an ensemble with better performance. The results show that CCME is better on average than the original ME on a number of classification problems. We have also introduced a novel mechanism for visualizing the modular structures that emerged from the model.

Keywords: Neuro-ensemble; Artificial neural networks; Mixture of experts; Cooperative coevolution

1. Introduction

Evolutionary artificial neural networks (EANNs) have been widely studied in the last few decades. The main power of artificial neural networks (ANNs) lies in their ability to correctly learn the underlying function or distribution in a data set from a sample. This ability is called generalization. Mathematically, the generalization ability can be expressed in terms of minimizing the recognition error of the neural network, on previously unseen data. Thus evolutionary computation (EC), a global optimization approach, can be employed to optimize this error function. As discussed in the prominent review of Yao [31], evolutionary methods can be applied on different levels of ANN, such as the architecture and the connection weights.

Much of the ANN literature concentrates on finding a single solution (network) to learn a task. However, an optimum network on the training data (i.e. seen data) may not generalize well on the testing data (i.e. unseen data). An ANN could either overtrain/overfit (memorizing the data rather than learning the correct distribution) or undertrain (not trained enough, or too simple, to fit the data well) (see [7] on the bias/variance dilemma and the generalization problem). Many published works have shown that an ensemble of neural networks (i.e. neuro-ensemble) can generalize better than individual networks [10,16–18,23,25,32–34]. The main argument for neuro-ensembles is that different members of the ensemble may possess different bias/variance trade-offs, hence a suitable combination of these biases/variances could result in an improvement in the generalization ability of the whole ensemble [25,34]. It is obvious that a self-similar set of

*Corresponding author.
E-mail addresses: minhha_76@yahoo.com (M.H. Nguyen), h.abbass@adfa.edu.au (H.A. Abbass), rim@cse.snu.ac.kr (R.I. Mckay).

0925-2312/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2006.04.009

individuals is not desirable, since it multiplies the effort to train them without adding to the overall performance—i.e. the system's performance is similar to that of a single network.

One important application of neuro-ensembles is in problem decomposition. Most real-world problems are too complicated for a single individual to solve. Divide-and-conquer has proved to be efficient in many of these complex situations. The issues are (i) how to divide the problem into simpler tasks, (ii) how to assign individuals to solve these subtasks and (iii) how to synthesize the whole system back together. If the problem has a distinct natural decomposition, it would be possible to derive such a decomposition by hand. However, in most real-world problems, we either know too little about the problem, or it is too complex for us to have a clear understanding on how to hand-decompose it into subproblems. Thus, it is desirable to have a method to automatically decompose a complex problem into a set of overlapping or disjoint subproblems, and to assign one or more specialized problem solving tools or experts to each of these subproblems. The remaining question is how to combine the outputs of these experts if the decomposition scheme is unknown in advance.

Jacobs [8,9] has proposed an ensemble method called mixture of experts (ME), based on the divide-and-conquer principle. In their method, instead of assigning a set of combinational weights to the experts, an extra gating component is used to compute these weights dynamically from the inputs (Fig. 1). This gating component is trained, together with other experts, through a specially tailored error function, which localizes the experts into different subsets of the data while improving the system's performance. In the ME model, the expert could be of any type, e.g. an ANN or a C4.5 decision tree, but the gating is often an ANN. Jordan and Jacobs [11,12] extended the model to the so-called hierarchical mixture of experts (HME), in which each component of the ME model is replaced with an ME model. Since Jacobs' proposal of the ME model in 1991, there has been a wide range of research into it.

Some authors [1,13,14] have established how the ME model works in statistical terms. Waterhouse [28,30] and Moeland [19] have applied the Bayesian framework to design and explain the ME model. According to their interpretation, the ME output(s) can be considered as estimates of the posterior probabilities of class membership [19]. Thus, the Bayesian framework can be used to design the training error function [3] and estimate the parameters for the ME model [30]. Besides the original ME model, a large number of variants have been put forward. Waterhouse and Cook [29] and Avnimelech and Cook [2] proposed to combine ME with the boosting algorithm. They argued that, since boosting encourages classifiers to be experts on different patterns that previous experts disagree on, it can split the data set into regions for the experts in the ME model, and thus ensure localization of experts. The dynamic gating function of the ME ensures a good combination of classifiers [2]. Tang et al. [26] tried to explicitly localize the experts by applying a self-organizing map to partition the input space for the experts. Wan and Bone [27] used a mixture of radial basis function networks to partition the input space into statistically correlated regions and learn the local covariation model of the data in each region.

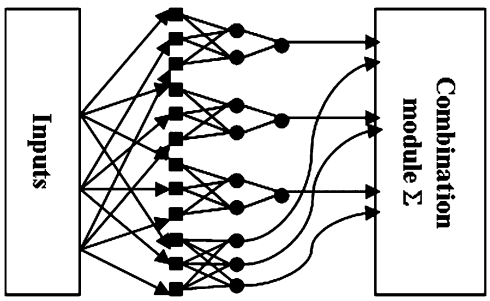

Fig. 1. Mixture of expert architecture.

Although gradient descent is the most popular ANN training method, especially in industrial problems, because of its simple implementation and its efficiency, it has some serious drawbacks. The growing literature on EC research as a global optimization method led to a number of successful attempts to evolve ANNs [31]. A newer branch of EC called the cooperative coevolutionary (CC) algorithm was proposed by Potter and De Jong [21,22].

Garcia-Pedrajas et al. [6] have applied multiobjective optimization in conjunction with CC, to evolve a set of subpopulations of well-performed, regularized, cooperative and diverse ANNs which can be used in a set of ensembles. Khare et al. [15] used the concept of CC on a set of subpopulations of radial basis function networks, where each subpopulation is designed to solve a particular subtask of the whole problem. A second level, consisting of a swarm of ensembles, to combine selected ANNs from the subpopulation, is also evolved in parallel with these subpopulations. The two disadvantages of this method are (i) its requirement for a prior knowledge about the problem in order to know the fixed number of required modules and (ii) its dependence on credit assignment, in that the fitness of each module is decided by the contribution of the module to the whole system. To solve the problem of fixed number of modules, Khare et al. [15] suggested using Potter's approach of adding and removing subpopulations whenever the system's fitness stagnates for a predetermined period. Despite the remaining credit assignment problem, their method has the merit that both the structures and parameters, of both the modules and the whole ensemble, can be evolved within the framework.

2. Mixture of experts

The ME model consists of a number of experts combined through a gate (Fig. 1), all having access to the input space. The components can be any type of classifiers—in this paper, we use simple feed-forward multilayer neural

networks. The output $y(\vec{x})$ of a ME model is the weighted average of the individual experts' outputs $y_m(\vec{x})$, $m = 1,\ldots,M$, with the weights $g_m(\vec{x})$, $m = 1,\ldots,M$, produced by the gate network:

$$y(\vec{x}) = \sum_{m=1}^{M} g_m(\vec{x}) y_m(\vec{x}). \quad (1)$$

The gate output can be considered as the probability that expert m is selected. To ensure that $g_m(\vec{x})$ satisfies the probability axioms (i.e. $g_m(\vec{x}) \geqslant 0$ and $\sum_{m=1}^{M} g_m(\vec{x}) = 1$), a soft-max function is applied on the raw outputs $z_m(\vec{x})$ of the gate:

$$g_m(\vec{x}) = \frac{\exp(z_m(\vec{x}))}{\sum_{m'=1}^{M} \exp(z_{m'}(\vec{x}))}, \quad (2)$$

where $z_m(\vec{x}) = \vec{v}_m.\vec{x}$ is the mth linear output of the gate and $\vec{v}_m$ are the weights of the gate that connect the inputs $\vec{x}$ to the mth output.

From a Bayesian point of view, an ME model could be interpreted as an input conditional mixture model, with the data assumed to be generated from a stochastic series of processes. Each data point $(\vec{x}^i, d^i)$ is assumed to be generated by a process m. Assume that there exists a probability distribution $P(Z)$ such that each z_m^i is the decision to use process m for case i. With this probabilistic interpretation, the experts in the ME system model the processes, a gating network is used to model the decision probability distribution $P(Z)$ [28].

$$P(\vec{d}^i|\vec{x}^i, \vec{w}_1, \vec{w}_2, \ldots, \vec{w}_M, \vec{v}) = \sum_{m=1}^{M} P(m|\vec{x}^i, \vec{v}) P(\vec{d}^i|\vec{x}^i, \vec{w}_m, m), \quad (3)$$

where $\{\vec{w}_1, \vec{w}_2, \ldots, \vec{w}_M, \vec{v}\}$ is the parameter space, $\vec{w}_m$ the weights of the mth expert and $\vec{v}$ the weights of the gating network. $P(m|\vec{x}^i, \vec{v})$ is the conditional probability of the gating network to select expert m and $P(\vec{d}^i|\vec{x}^i, \vec{w}_m, m)$ is the conditional probability of expert m to produce output $\vec{d}^i$. The latter conditional probability is the underlying mechanism of the expert network. By varying this function, the desired behavior is achieved. According to Bishop [3], for binary classification problems, the most suitable conditional probability for the experts is the following cross-entropy between the target d^i and the outputs of the experts y_m^i for the ith input pattern:

$$P(d^i|\vec{x}^i, \vec{w}_m, m) = (y_m^i)^{d^i} + (1 - y_m^i)^{(1-d^i)}. \quad (4)$$

3. Cooperative coevolutionary mixture of experts (CCME)

In this paper, a novel method is introduced, combining the ME model with the CC mechanism. On the one hand, CC allows the system to incorporate both EC and backpropagation (BP); thus, ME enhances the learning capabilities of BP. The CC framework naturally supports problem decomposition, assisting the capabilities of ME. On the other hand, the ME model imposes an external diversity force, driving the components into different local regions, and thus ensuring diversity between the subpopulations. The emergent character of this diversity, not imposed by the users, ensures that it can adapt to the actual complexity of the search space. Moreover, the ME model is localized, in the sense that each expert is responsible for a subregion of the input space. This is well-suited to the locality requirements of the coevolutionary framework, in which each species preferentially occupies a local niche in the environment.

Fig. 2 illustrates the architecture of the CCME model. The CCME model consists of a number of subpopulations, each of which can be considered as a supply pool for each expert in the ME model, so that the experts do not depend too much on a single initialization. If the experts are ANNs, then each subpopulation contains a collection of ANNs, called individuals in the EC terminology. To evaluate the fitness of an individual k in a subpopulation j, it is required, in the CC scheme, that the other subpopulations contribute a representative(s) to the evaluation module. This representative(s) can be the best individual(s) in the subpopulation (greedy algorithm) or a randomly selected one(s) or both. The evaluation module will combine an individual k with these representatives to form a complete system (i.e. a full ensemble) and evaluate it. The performance of the complete system is assigned back to individual k as its fitness. In other words, the fitness of an individual is calculated based on how well it works with the other populations, hence the term cooperative.

In our system, the ME model is used as the evaluation module. To evaluate the fitness of each individual, it is assembled with representatives from the other subpopulations to form an ME. This ME is trained using BP (as a local hill-climber) for a number of epochs to evaluate the fitness. In the EC literature, one can find two different modes for combining learning and evolution: Lamarckian and Darwinian. In the Lamarckian model, the lifetime training is encoded back to the individual, while in Darwinian evolution, the training is not directly encoded (though due to the Baldwin effect, the trained results may gradually become reflected in the genome). Each of these has its own advantages and weaknesses. However, Lamarckian evolution matches our aims better, in that training allows each individual to be a true expert in the ME, while supporting the localization of individuals, and

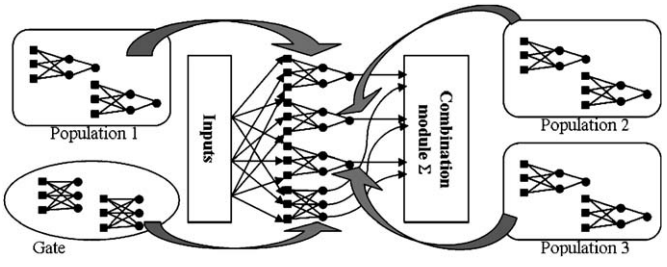

Fig. 2. CCME architecture.

hence assisting in finding suitable local niches for the species.

The complete CCME algorithm is summarized below.

Algorithm 1 (*ME learning with cooperative coevolution*).

0. *Initialize the populations* P_j. *Initialize empty pools* P'_j. *Set* $f_{E\text{clone}} = 0$.
1. *For* $i = 0$ *to the maximum number of generations*
   1.1. *Clone the best individual of each population* P_j to P'_j (*elites*)
   1.2. *Apply one evolutionary step to each population* P_j
   1.3. *For each individual* k *in each population* P_j
      1.3.1. *Form an ensemble* E *consisting of* k *and* $P'_{j'}, j' \neq j$
      1.3.2. *Local search* E *using BP and ME.*
      1.3.3. *Compute fitness* f_E *of* E *in terms of classification correction rate, assign this fitness to individual* k.
      1.3.4. *If* $f_E > f_{E\text{clone}}$, *copy* E *to* E_{clone} *and set* $f_{E\text{clone}} = f_E$.
   1.4. *Copy components of* E_{clone} (*the best ensemble in the current generation*) *to the corresponding populations* P_j. *Empty* P'_j.

2. *Apply the selecting criterion to select and output the desired ensemble.*

4. Experiments

CCME is verified on 10 standard data sets taken from the UCI Machine Learning Repository as summarized in Table 1. These data sets were downloaded from ice.uci.edu [4].

In the experiments, we used 10-fold cross-validation for each data set. The data set is divided into 10 disjoint subsets using stratified sampling. Each of the subsets is taken in turn as the test set, making 10 trials in all. In each experiment, one of the remaining subsets is chosen at random as the validation set, while the remaining eight subsets are combined to form the training set. A different prefixed random seed is used to generate the required random numbers (e.g. network weights, noise, cross-over and mutation rates) for each fold. For each fold, the system is trained using the training set, stopped by one of the different criteria using the validation set, and the ensemble obtained by combining the population at the stopping point is tested on the test set. The test-set error is treated as the primary result of each trial. The average test-set error, and its standard deviation, over the 10 trials are reported as the system's performance.

4.1. Parameters

In this paper, we use the following architecture for the ME model: three experts, each of which is a feed-forward ANN with one hidden layer consisting of three hidden nodes. The nodes in the expert ANN are all sigmoidal. The gate component is a feed-forward ANN, which has three linear output nodes corresponding to three experts and no hidden nodes. The outputs of the gate network are passed through a soft-max function to obtain probability-like values.

In this set of experiments, we use two setups for ME. In the first setup, the ME model is trained with the BP algorithm for 320,000 epochs with three sets of different learning rates η for the experts and the gate: (a) $\eta_{\text{experts}} = \eta_{\text{gate}} = 0.1$, (b) $\eta_{\text{experts}} = 0.01, \eta_{\text{gate}} = 0.1$ and (c) $\eta_{\text{experts}} = 0.1, \eta_{\text{gate}} = 0.01$. In the second setup, instead of training one ME for 320,000 epochs, 160 ME systems with randomized seeds are trained for 2000 epochs, the best ME as measured by the validation set out of 160 MEs is selected to run against the test data.

For CCME, we use a simple self-adaptive $(20 + 20)$-ES [24] with self-adaptive mutation rate, uniform cross-over and an elitism of one. We have tested a number of cross-over rates and found that it does not significantly affect the performance of CCME, the experiments in this paper use a cross-over rate of 0.5. The number of populations/species is the ensemble size plus the gate population. For example, to generate a system with an ensemble size of 3, we create three populations for these three experts and fourth population for the gating network. We apply a coevolutionary scheme, in which each individual is evaluated against the best in the other populations. The system is run

Table 1
Ten data sets from the UCI Machine Learning Repository

	Number of instances	Number of attributes	Continuous attributes	Discrete attributes
Breast cancer	699	9	9	
Australian credit card	690	14	6	8
Diabetes	768	8	8	
Liver disorder	345	6	6	
Tic tac toe	958	9		9
Cleveland heart	303	13	7	6
StatLog heart	270	13	7	6
Hepatitis	155	19	6	13
Ljubljana breast cancer	286	9	9	
House voting 84	435	16		16

for 200 generations, and local search is performed for 10 epochs per generation; thus the overall computational cost (number of evaluations) is consistent between the experiment with BP alone and the experiments with CCME.

In the literature of neuro-ensembles [5], and in our previous work [20], it was found that early stopping is beneficial for generalization. Here, we apply an early stopping criteria based on the minimum error on the validation set. In this scheme, the ensemble with the minimum validation error is selected as the final ensemble, and it is tested against the testing data. The average and standard deviation of the test error rate (%) is reported as the performance of the ensemble. To simplify comparison of the results, the smaller of a pair of treatments is underlined, with the cases that were significantly different being bold and underlined. In some comparisons, which involve multiple (three or more) columns, an ANOVA test is used and plotted. In the ANOVA plot, the group mean and error bars are shown; "Two means are significantly different if their intervals are disjoint, and are not significantly different if their intervals overlap" (MATLAB manual).

4.2. CCME vs. ME on generalization performance

In this section, we compare CCME against the BP ME with two different setups: (i) a single ME trained with 320,000 epochs and (ii) 160 MEs trained with 2000 epochs. In the second setup, we only select the best ME as measured by the validation set out of the 160 MEs. This is consistent with CCME experiments. Based on the results in Table 2, we see that—regardless of the learning rate—CCME performs better than ME on average. Fig. 3 shows an ANOVA significance test of CCME vs. ME performance, taken over the 10 data sets and the range of learning rates used. The disjoint intervals imply a statistically significant difference in the performance of ME and CCME. This provides a strong evidence of the advantage gained by using CC in conjunction with the ME model.

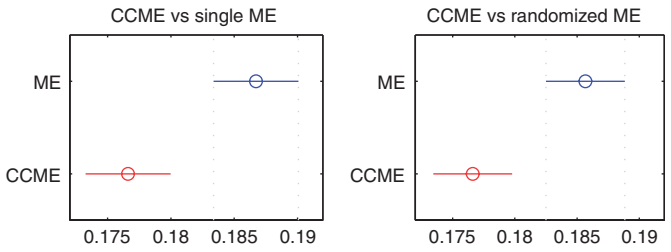

Fig. 3. ANOVA test on the generalization errors for the CCME vs. ME for 10 data sets across all three learning rates. The non-overlap among intervals means the two methods are significantly different.

Table 2
Average and standard deviation error rates of CCME vs. (a) a single ME with 320,000 epochs and (b) the best ME out of 160 randomly generated ME with 2000 epochs using three different sets of learning rates

Data set	$\eta_{\text{experts}} = \eta_{\text{gate}} = 0.1$		$\eta_{\text{experts}} = 0.01,\ \eta_{\text{gate}} = 0.1$		$\eta_{\text{experts}} = 0.1,\ \eta_{\text{gate}} = 0.01$	
	ME	CCME	ME	CCME	ME	CCME
(i) *CCME vs. single ME trained with* 320,000 *epochs*						
Breast cancer	0.037 (0.019)	**0.030 (0.014)**	0.040 (0.016)	**0.033 (0.015)**	0.039 (0.017)	**0.033 (0.015)**
Cleveland heart	0.208 (0.052)	0.225 (0.073)	0.258 (0.098)	0.228 (0.059)	0.241 (0.077)	0.215 (0.047)
Australian credit card	0.158 (0.034)	**0.126 (0.038)**	0.158 (0.044)	**0.136 (0.036)**	0.146 (0.034)	0.146 (0.030)
Diabetes	0.254 (0.057)	0.232 (0.046)	0.230 (0.065)	0.230 (0.048)	0.240 (0.060)	0.227 (0.038)
StatLog heart	0.211 (0.080)	**0.174 (0.089)**	0.185 (0.087)	0.159 (0.092)	0.219 (0.120)	0.181 (0.069)
Hepatitis	0.200 (0.099)	0.168 (0.061)	0.180 (0.064)	0.188 (0.090)	0.200 (0.090)	0.194 (0.053)
Liver disorder	0.285 (0.079)	0.299 (0.079)	**0.282 (0.059)**	0.335 (0.106)	0.317 (0.067)	0.302 (0.081)
Ljubljana breast cancer	0.263 (0.040)	0.297 (0.112)	0.283 (0.033)	**0.245 (0.049)**	0.256 (0.031)	0.276 (0.112)
Tic tac toe	0.185 (0.050)	**0.129 (0.038)**	**0.192 (0.055)**	0.224 (0.070)	0.146 (0.034)	0.122 (0.056)
House voting 84	0.064 (0.035)	**0.037 (0.034)**	0.055 (0.039)	**0.046 (0.032)**	0.071 (0.052)	0.062 (0.054)
(i) *CCME vs.* 160 *randomized ME trained with* 2000 *epochs*						
Breast cancer	0.043 (0.019)	**0.030 (0.014)**	0.046 (0.022)	**0.033 (0.015)**	0.049 (0.018)	**0.033 (0.015)**
Cleveland heart	0.244 (0.087)	0.225 (0.073)	0.248 (0.083)	0.228 (0.059)	0.225 (0.044)	0.215 (0.047)
Australian credit card	0.133 (0.046)	0.126 (0.038)	0.132 (0.039)	0.136 (0.036)	0.157 (0.041)	0.146 (0.030)
Diabetes	0.243 (0.055)	0.232 (0.046)	0.236 (0.055)	0.230 (0.048)	0.246 (0.055)	**0.227 (0.038)**
StatLog heart	0.222 (0.068)	**0.174 (0.089)**	0.237 (0.088)	**0.159 (0.092)**	0.185 (0.099)	0.181 (0.069)
Hepatitis	0.194 (0.062)	0.168 (0.061)	0.193 (0.080)	0.188 (0.090)	**0.161 (0.073)**	0.194 (0.053)
Liver disorder	0.287 (0.039)	0.299 (0.079)	**0.297 (0.081)**	0.335 (0.106)	0.319 (0.071)	0.302 (0.081)
Ljubljana breast cancer	0.322 (0.082)	0.297 (0.112)	0.283 (0.064)	**0.245 (0.049)**	0.245 (0.072)	0.276 (0.112)
Tic tac toe	0.111 (0.033)	0.129 (0.038)	**0.185 (0.065)**	0.224 (0.070)	0.130 (0.044)	0.122 (0.056)
House voting 84	0.067 (0.048)	**0.037 (0.034)**	0.080 (0.053)	**0.046 (0.032)**	0.050 (0.042)	0.062 (0.054)

The smaller of a pair of treatments is underlined, with the cases that were significantly different being bold and underlined.

Table 3
Mean (and standard deviation) of the total running time of ME and CCME on 10 data sets

Data set	ME (s)	CCME (s)
Breast cancer (bre)	1385 (13)	1560 (5)
Cleveland heart (cle)	717 (2)	759 (3)
Australian credit card (crx)	1662 (7)	1801 (7)
Diabetes (dia)	1494 (8)	1632 (7)
StatLog heart (hea)	641 (2)	676 (4)
Hepatitis (hep)	465 (6)	492 (5)
Liver disorder (liv)	619 (3)	634 (5)
Ljubljana breast cancer (lub)	563 (4)	642 (6)
Tic tac toe (ttt)	1927 (12)	2169 (7)
House voting 84 (vot)	1125 (8)	1208 (6)

4.3. CCME vs. ME on time complexity

To understand the time complexity of CCME, we compare the running time of CCME with the running time of the simple BP ME on 15 data sets. The running time is counted over 200 generations for CCME and 320,000 epochs for ME to equalize the number of objective evaluations for both methods $[320,000 = 160(\text{individuals}) \times 200(\text{generations}) \times 10$ (epochs in local search)]. The results of three learning rates over 10-folds are averaged and reported in Table 3. The results show that CCME is slightly slower than ME due to the evolutionary overhead. However, the running time of CCME is still of the same magnitude as the running time of ME. In other words, CC does not add significant computation time to the overall running time of the system.

4.4. CCME with various ensemble sizes

In this set of experiments, we test the effect of ensemble size on the CCME model on different ensemble sizes, to see how robust the results are to different ensemble sizes. The CCME model with five different ensemble sizes of 3, 6, 9, 12 and 15 experts are compared on the first five data sets. The average error rates and standard deviations on the five data sets (left) and the anova test (right) are presented in Fig. 4. It is clear that the CCME model is quite robust to variations in the ensemble size, i.e. the model is not sensitive to different ensemble sizes. The ANOVA plot shows that the ensemble size has no significant effect on the performance of the ensemble.

4.5. CCME with various network complexity

In this set of experiments, we test the CCME model with different network complexities (represented by the number of hidden units in each individual ANN of the expert). CCME models with four different numbers of hidden nodes: 3, 5, 7, 9 are compared in terms of the performance (error rate). The average error rates over five data sets (left) and ANOVA tests (right) are presented in Fig. 5. The plots indicate the robustness of the CCME model to the network

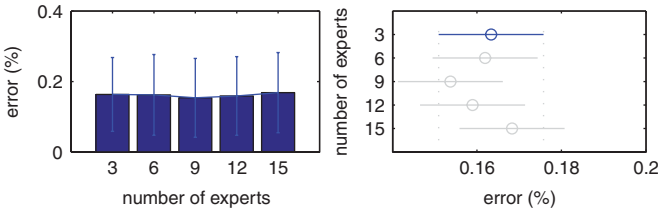

Fig. 4. ANOVA test for the ensemble size factor for CCME: none of the five ensemble sizes are significantly different.

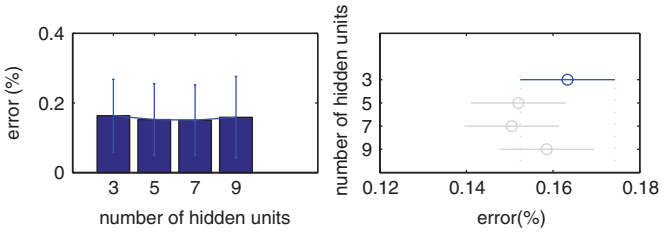

Fig. 5. ANOVA test for the network complexity factor for CCME: none of the four complexities are significant.

complexity, i.e. performance is not significantly changed with different number of hidden units per individual expert. Although the plot suggests that too simple and too complex (in terms of number of hidden units) ANNs are not desirable, the lack of a significant difference suggests that choosing the right complexity is not too important.

4.6. CCME as automatic problem decomposition

This section of the paper is devoted to the analysis of the CCME model as a method of automatic problem decomposition (APD). We introduce a novel way to visualize the APD, by plotting the output of the individual ANN together with the inputs. We call these plots "modularity plots". We have found that the modularity plots are useful in understanding the data.

As discussed previously, the gate component of CCME acts as a soft switch, selecting an expert (or experts) to be responsible for each data record. Thus, the outputs of this gate can be considered as a measure of the degree of responsibility of the experts for an input instance. To visualize the responsibility of each expert vs. the data, instead of using the actual weights produced by the gate, a value "1" is assigned to the expert with the maximum responsibility and "0" to the rest. However, since the gate is a "soft" switch in the sense that it might select more than one expert for a certain pattern, in those cases, 0's are assigned to all the experts. In other words, a value of "1" indicates strong responsibility (i.e. the corresponding weight is distinctly higher than other experts' weights).

In order to make the pattern clearer, we needed to reorder the data. The order of the data is not significant in these problems. Thus we group the data instances that each expert is responsible for together, to visualize them next to

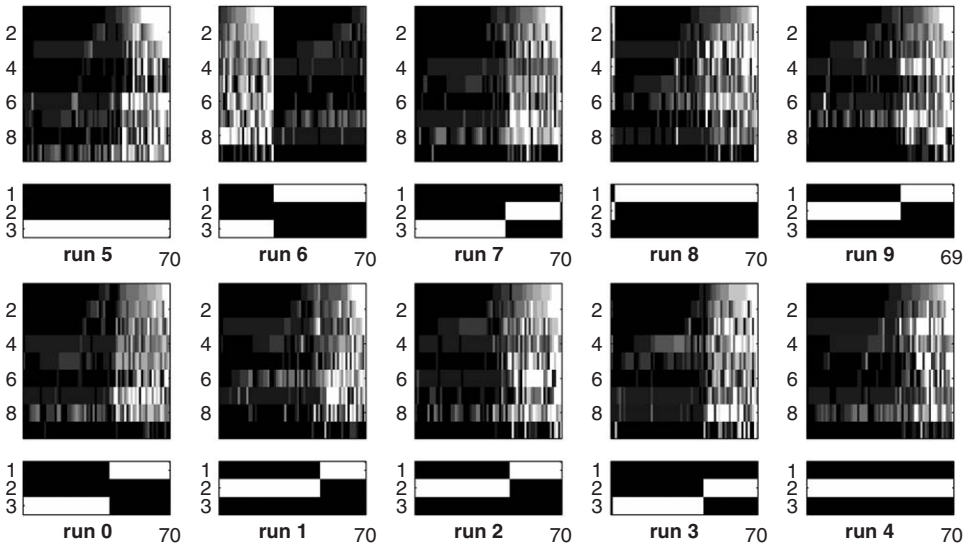

Fig. 6. Individuals' output vs. inputs in 10 runs of the breast cancer data set with ensemble size = 3.

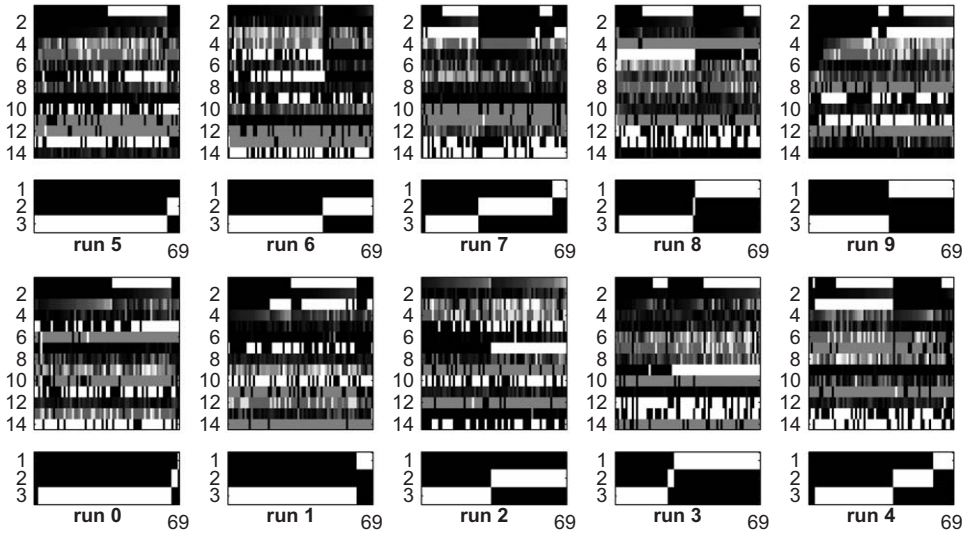

Fig. 7. Individuals' output vs. inputs in 10 runs of the Australian credit card data set with ensemble size = 3.

each other. This makes any common pattern between these instances easier to see.

Figs. 6 and 7 show the modularity plots of the experts' responsibility against the inputs for the breast cancer, Wisconsin and Australian credit card data sets. The gray cells correspond to values from 0 to 1, where black means 0 and white means 1. In each run, the white bars in the lower box represent the experts' responsibility while the rows in the upper box display the values of each input attribute. The x-axis is the pattern index.

For easier data sets such as breast cancer, the plots show that the white bars (in the lower plot) distinctively divide the upper plot into regions of dark and light shades. In other words, CCME can decompose the problem into distinct regions and assign individual experts to each region.

For more complex problems, such as Australian credit card, where the relationship between the inputs is more complicated, the plots still show a degree of relationship between the expert's responsibility and the input space. For example, in run 2, expert 2 is distinctly responsible for input 7 (i.e. the black and white regions for expert 2 coincide with the black and white regions of input 7). This responsibility is observed throughout most of the runs for the Australian credit card. This supports the decomposability nature of CCME.

5. Conclusion and future work

In this paper, we have introduced a novel method based on the principles of both Cooperative Coevolution and Mixture of Experts. We have investigated different aspects of the proposed CCME model. The results of the experiments can be summarized as follows: (i) CCME is robust to varying ensemble complexity, in terms of the

number of individual experts and (ii) CCME is robust to varying ANN complexity, in terms of the number of hidden units. When comparing CCME and ME, the results show that CCME outperforms ME on a number of classification problems.

We also studied the decomposition generated by CCME. The visualization shows that CCME can automatically decompose problems into different regions of the input space, and assign experts to these distinct regions. Currently, we are extending the APD analysis to investigate the class boundaries produced by the ME and CCME models. Such analysis will help us to understand how ME and CCME decompose problems.

The proposed system can contribute to answering many fundamental and practical problems such as self-organization, training a heterogeneous swarm of robots and agents, automatic identification of building blocks and APD.

Acknowledgments

H. Abbass wishes to thank the ARC Centre for Complex Systems Grant number CEO0348249. The authors wish to thank the anonymous reviewers and the editor for their constructive comments.

References

[1] L. Alexandre, A. Campilho, M. Kamel, Bounds for the average generalization error of the mixture of experts neural network, in: A. Fred (Ed.), SSPR/SPR, Springer, Berlin, 2004, pp. 618–625.
[2] R. Avnimelech, N. Intrator, Boosted mixture of experts: an ensemble learning scheme, Neural Comput. 11 (2) (1999) 483–497.
[3] C.M. Bishop, Neural Networks for Pattern Recognition, Oxford University Press, Oxford, 1995.
[4] C. Blake, C. Merz, Uci repository of machine learning databases. ⟨html://www.ics.uci.edu/mlearn/mlrepository.html⟩, 1998.
[5] K. Chellapilla, D. Fogel, Evolution, neural networks, games, and intelligence, Proc. IEEE 87 (9) (1999) 1471–1496.
[6] N. Garcia-Pedrajas, C. Hervas-Martinez, D. Ortiz-Boyer, Cooperative coevolution of artificial neural network ensembles for pattern recognition, IEEE Trans. Evol. Comput. 9 (3) (2005) 271–302.
[7] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall, Englewood Cliffs, NJ, 1999.
[8] R. Jacobs, M. Jordan, A. Barto, Task decomposition through competition in a modular connectionist architecture: the what and where vision tasks, Technical Report, University of Massachusetts, 1991.
[9] R. Jacobs, M. Jordan, S. Nowlan, G. Hinton, Adaptive mixtures of local experts, Neural Comput. 3 (1991) 79–87.
[10] D. Jimenez, N. Walsh, Dynamically weighted ensemble neural networks for classification, in: Proceedings of the 1998 International Joint Conference on Neural Networks, 1998, pp. 753–756.
[11] M. Jordan, R. Jacobs, Hierarchies of adaptive experts, in: J. Moody, S. Hanson, R. Lippmann (Eds.), Advances in Neural Information Processing Systems, vol. 4, Morgan Kaufmann, Los Altos, CA, 1992, pp. 985–992.
[12] M. Jordan, R. Jacobs, Hierachical mixtures of experts and the EM algorithm, Neural Comput. 6 (2) (1994) 181–214.
[13] M.I. Jordan, L. Xu, Convergence results for the EM approach to mixtures of experts architectures, Artificial Intelligence Laboratory@MIT, November 1993.
[14] K. Kang, J.-H. Oh, Statistical mechanics of the mixture of experts, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, The MIT Press, Cambridge, MA, 1997, p. 183 URL: ⟨citeseer.ist.psu.edu/kang97statistical.html⟩.
[15] V. Khare, X. Yao, B. Sendhoff, Y. Jin, H. Wersing, Co-evolutionary modular neural networks for automatic problem decomposition, in: Proceedings of the 2005 IEEE Congress on Evolutionary Computation, IEEE Press, New York, 2005, pp. 2691–2698.
[16] Y. Liu, X. Yao, Evolving modular neural networks which generalise well, in: Proceedings of 1997 IEEE International Conference on Evolutionary Computation, Indianapolis, USA, 1997, pp. 605–610.
[17] Y. Liu, X. Yao, Ensemble learning via negative correlation, Neural Networks 12 (1999) 1399–1404.
[18] Y. Liu, X. Yao, T. Higuchi, Evolutionary ensembles with negative correlation learning, IEEE Trans. Evol. Comput. 4 (4) (2000) 380–387.
[19] P. Moeland, Mixtures of experts estimate a posteriori probabilities, in: Proceedings of the International Conference on Artificial Neural Networks, 1997, pp. 499–504.
[20] M.H. Nguyen, H.A. Abbass, R.I. McKay, Stopping criteria for ensemble of evolutionary artificial neural networks, in: Applied Soft Computing, Elsevier, Amsterdam, 2005, pp. 100–107.
[21] M. Potter, K. De Jong, Cooperative coevolution: an architecture for evolving coadapted subcomponents, Evol. Comput. 8 (1) (2000) 1–29.
[22] M.A. Potter, The design and analysis of a computational model of cooperative coevolution, Ph.D. Thesis, George Mason University, 1997.
[23] B. Rosen, Ensemble learning using decorrelated neural networks, Connect. Sci. Spec. Issue Combining Artif. Neural: Ensemble Approaches 8 (3/4) (1996) 373–384.
[24] H. Schwefehm, Numerical Optimization for Computer Models, Wiley, Chichester, UK, 1981.
[25] A. Sharkey, Combining Artificial Neural Nets. Ensemble and Modular Multi-Net Systems, Springer, New York, 1998.
[26] B. Tang, M. Heywood, M. Shepherd, Input partitioning to mixture of experts, in: International Joint Conference on Neural Networks, 2002, pp. 227–232.
[27] E. Wan, D. Bone, Interpolating earth-science data using rbf networks and mixtures of experts, in: NIPS, 1996, pp. 988–994.
[28] S. Waterhouse, Divide and conquer pattern recognition using mixtures of experts, Ph.D. Thesis, University of Cambridge, February 1997.
[29] S. Waterhouse, G. Cook, Ensemble methods for phoneme classification, in: M. Mozer, J. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, The MIT Press, Cambridge, MA, 1997, pp. 800–806.
[30] S. Waterhouse, D. MacKay, T. Robinson, Bayesian methods for mixture of experts, in: D.S. Touretzky, M.C. Mozer, M.E. Hasselmo (Eds.), Advances in Neural Information Processing Systems, vol. 8, The MIT Press, Cambridge, MA, 1996, pp. 351–357.
[31] X. Yao, Evolving artificial neural networks, Proc. IEEE 87 (1999) 1423–1447.
[32] X. Yao, Y. Liu, Ensemble structure of evolutionary artificial neural networks, in: IEEE International Conference on Evolutionary Computation (ICEC'96), Nagoya, Japan, 1996, pp. 659–664.
[33] X. Yao, Y. Liu, A new evolutionary system for evolving artificial neural networks, IEEE Trans. Neural Networks 8 (1997) 694–713.
[34] X. Yao, Y. Liu, Making use of population information in evolutionary artificial neural networks, IEEE Trans. Syst. Man Cybern. Part B: Cybern. 28 (1998) 417–425.

Minh Ha Nguyen graduated (with first class honors) from the University of Canberra, Australia, in 1999. She is a PhD candidate of the University of New South Wales, Australia, in 2002–2006. Her work is focusing on EC, neural networks and data mining.

Hussein A. Abbass is the director of the Artificial Life and Adaptive Robotics Laboratory, the School of Information Technology and Electrical Engineering, University of New South Wales at the Australian Defence Force Academy in Canberra, Australia. He is a senior member of the IEEE, the chair of IEEE working group on Artificial Life and Complex Adaptive Systems, technical cochair of SEAL 2006, the cochair of the First IEEE Symposium on Artificial Life, Honolulu 2007, technical cochair of IEEE-CEC 2007 and is/has been cochair and on the program committee of many conferences in the field. His work is focusing on EC, neural networks and complex systems.

Robert I. McKay graduated from the Australian National University in 1971 and was awarded his PhD in the theory of computation from the University of Bristol in 1976. He researched computer typesetting in the Commonwealth Scientific and Industrial Research Organization from 1976–1985, when he joined the University of New South Wales (Australian Defence Force Academy campus). He moved to Seoul National University, where he heads the Structural Complexity Laboratory (http://scsnu.ac.kr), in 2005.