PERGAMON

Neural Networks 12 (1999) 1399–1404

Neural
Networks

www.elsevier.com/locate/neunet

Contributed article

Ensemble learning via negative correlation

Y. Liu[a,*], X. Yao[b]

[a]Evolvable Systems Laboratory, Computer Science Division, Mbox 1501, Electrotechnical Laboratory, 1-1-4 Umezono, Tsukuba, Ibaraki 305-8568, Japan
[b]School of Computer Science, The University of Birmingham, Edgbaston, Birmingham B15 2TT, UK

Received 11 June 1998; received in revised form 5 July 1999; accepted 5 July 1999

Abstract

This paper presents a learning approach, i.e. negative correlation learning, for neural network ensembles. Unlike previous learning approaches for neural network ensembles, negative correlation learning attempts to train individual networks in an ensemble and combines them in the same learning process. In negative correlation learning, all the individual networks in the ensemble are trained simultaneously and interactively through the correlation penalty terms in their error functions. Rather than producing unbiased individual networks whose errors are uncorrelated, negative correlation learning can create negatively correlated networks to encourage specialisation and cooperation among the individual networks. Empirical studies have been carried out to show why and how negative correlation learning works. The experimental results show that negative correlation learning can produce neural network ensembles with good generalisation ability. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Neural network ensembles; Negative correlation learning; Generalisation; Correlation; Combination method; Correct response set

1. Introduction

Many real-world problems are too large and too complex for a single monolithic system to solve alone. There are many examples from both natural and artificial systems that show that a composite system consisting of several subsystems can reduce the total complexity of the system while solving a difficult problem satisfactorily. The success of neural network ensembles in improving a classifier's generalisation is a typical example (Yao & Liu, 1998). However, designing neural network ensembles is a very difficult task.

There are several methods of designing neural network ensembles. Most of them follow the two-stage design process (Sharkey, 1996): first generating individual networks, and then combining them. Usually, the individual networks are trained independent of each other. One of the disadvantages of such an approach is the loss of interaction among the individual networks during learning. There is no feedback from the combination stage to the individual design stage. It is possible that some of the independently designed individual networks do not make much contribution to the whole ensemble.

This paper describes the negative correlation learning approach to designing neural network ensembles. The idea behind negative correlation learning is to encourage different individual networks in an ensemble to learn different parts or aspects of a training data so that the ensemble can learn the whole training data better. Negative correlation learning is different from previous work which trains the individual networks independently or sequentially (Drucker, Cortes, Jackel, LeCun & Vapnik, 1994; Perrone & Cooper, 1993). In negative correlation learning, all the individual networks in the ensemble are trained simultaneously through the correlation penalty terms in their error functions. Negative correlation learning attempts to train and combine individual networks in the same learning process. That is, the goal of each individual training is to generate the best result for the whole ensemble. Such an approach is quite different from other ensemble approaches, which separate the individual design from average procedures.

Negative correlation learning is also different from the mixtures-of-experts (ME) architecture (Jacobs, 1997) that consists of a gating network and a number of expert networks although ME architecture can also produce biased individual networks whose estimates are negatively correlated. Negative correlation learning does not need a separate gating network. It uses a totally different error function. The λ parameter in negative correlation learning provides a

[*] Corresponding author. Fax: + 81-298-545871.
E-mail addresses: yliu@etl.go.jp (Y. Liu), x.yao@cs.bham.ac.uk (X. Yao)

PII: S0893-6080(99)00073-8

convenient way to balance the bias-variance-covariance trade-off (Liu & Yao, 1999). ME architecture does not provide such control over the trade-off.

Empirical studies will be carried out in this paper on a time series prediction problem, i.e. the chlorophyll-a prediction in Lake Kasumigaura (Recknagel, Fukushima, Hanazato, Takamura & Wilson, 1999), to show why and how negative correlation learning works. Subsequently negative correlation learning will be tested on the Australian credit card assessment problem. The experimental results show that negative correlation learning can produce neural network ensembles with good generalisation ability.

The rest of this paper is organised as follows: Section 2 describes negative correlation learning. Section 3 analyses negative correlation learning on the chlorophyll-a prediction problem. Section 4 presents the experiment results of negative correlation learning on the Australian credit card assessment problem and some discussions. Finally, Section 5 concludes with a summary of the paper and a few remarks.

2. Negative correlation learning

Suppose that we have a training set

$$D = \{(\mathbf{x}(1), d(1)), \ldots, (\mathbf{x}(N), d(N))\}$$

where $\mathbf{x} \in R^p$, d is a scalar, and N is the size of the training set. The assumption that the output d is a scalar has been made merely to simplify exposition of ideas without loss of generality. This section considers estimating d by forming an ensemble whose output is a simple averaging of outputs of a set of neural networks

$$F(n) = \frac{1}{M}\sum_{i=1}^{M} F_i(n) \tag{1}$$

where M is the number of the individual neural networks in the ensemble, $F_i(n)$ is the output of network i on the nth training pattern, and $F(n)$ is the output of the ensemble on the nth training pattern.

Negative correlation learning introduces a correlation penalty term into the error function of each individual network in the ensemble so that all the networks can be trained simultaneously and interactively on the same training data set D. The error function E_i for network i in negative correlation learning is defined by

$$E_i = \frac{1}{N}\sum_{n=1}^{N} E_i(n) = \frac{1}{N}\sum_{n=1}^{N} \frac{1}{2}(F_i(n) - d(n))^2 + \frac{1}{N}\sum_{n=1}^{N} \lambda p_i(n) \tag{2}$$

where $E_i(n)$ is the value of the error function of network i at presentation of the nth training pattern. The first term in the right side of Eq. (2) is the empirical risk function of network i. The second term p_i is a correlation penalty function. The purpose of minimising p_i is to negatively correlate each network's error with errors for the rest of the ensemble. The parameter $0 \le \lambda \le 1$ is used to adjust the strength of the penalty. The penalty function p_i has the form:

$$p_i(n) = (F_i(n) - F(n))\sum_{j \ne i}(F_j(n) - F(n)) \tag{3}$$

The partial derivative of $E_i(n)$ with respect to the output of network i on the nth training pattern is

$$\begin{aligned}\frac{\partial E_i(n)}{\partial F_i(n)} &= F_i(n) - d(n) + \lambda\frac{\partial p_i(n)}{\partial F_i(n)} \\ &= F_i(n) - d(n) + \lambda\sum_{j \ne i}(F_j(n) - F(n)) \\ &= F_i(n) - d(n) - \lambda(F_i(n) - F(n)) \\ &= (1 - \lambda)(F_i(n) - d(n)) + \lambda(F(n) - d(n))\end{aligned} \tag{4}$$

where we have made use of the assumption that $F(n)$ has a constant value with respect to $F_i(n)$. The standard back-propagation (BP) algorithm (Rumelhart, Hinton & Williams, 1986) has been used for weight adjustments in the mode of pattern-by-pattern updating. That is, weight updating of all the individual networks is performed simultaneously using Eq. (4) after the presentation of each training pattern. One complete presentation of the entire training set during the learning process is called an *epoch*. The negative correlation learning from Eq. (4) is a simple extension to the standard BP algorithm. In fact, the only modification that is needed is to calculate an extra term of the form $\lambda(F_i(n) - F(n))$ for the ith network.

From Eqs. (2)–(4), we may make the following observations:

1. During the training process, all the individual networks interact with each other through their penalty terms in the error functions. Each network i minimises not only the difference between $F_i(n)$ and $d(n)$, but also the difference between $F(n)$ and $d(n)$. That is, negative correlation learning considers errors what all other networks have learned while training a network.
2. For $\lambda = 0.0$, there are no correlation penalty terms in the error functions of the individual networks, and the individual networks are just trained independently. That is, independent training for the individual networks is a special case of negative correlation learning.
3. For $\lambda = 1$, from Eq. (4) we get

$$\frac{\partial E_i(n)}{\partial F_i(n)} = F(n) - d(n) \tag{5}$$

Note that the empirical risk function of the ensemble for the nth training pattern is defined by

$$E_{\text{ens}}(n) = \frac{1}{2}\left(\frac{1}{M}\sum_{i=1}^{M} F_i(n) - d(n)\right)^2 \tag{6}$$

The partial derivative of $E_{\text{ens}}(n)$ with respect to F_i on the

Table 1
The average results of RMS errors produced by negative correlation learning with $\lambda = 1$ and independent training (i.e. $\lambda = 0.0$ in negative correlation learning) over 25 runs for the chlorophyll-a prediction in Lake Kasumigaura from 1984 to 1986. Mean, SD, Min and Max indicate the mean value, standard deviation, minimum and maximum value, respectively

λ	Year	Mean	SD	Min	Max
1	1984–1985	0.0161	0.0007	0.0150	0.0178
	1986	0.0815	0.0076	0.0732	0.1013
0	1984–1985	0.0236	0.0029	0.0215	0.0366
	1986	0.1168	0.0043	0.1099	0.1321

nth training pattern is

$$\frac{\partial E_{\text{ens}}(n)}{\partial F_i(n)} = \frac{1}{M}\left(\frac{1}{M}\sum_{i=1}^{M} F_i(n) - d(n)\right) = \frac{1}{M}(F(n) - d(n)) \tag{7}$$

In this case, we get

$$\frac{\partial E_i(n)}{\partial F_i(n)} \propto \frac{\partial E_{\text{ens}}(n)}{\partial F_i(n)} \tag{8}$$

The minimisation of the empirical risk function of the ensemble is achieved by minimising the error functions of the individual networks. From this point of view, negative correlation learning provides a novel way to decompose the learning task of the ensemble into a number of subtasks for different individual networks.

3. Correlations among the individual networks

This section analyses negative correlation learning on the chlorophyll-a prediction in Lake Kasumigaura (Recknagel et al., 1999) to show how and why negative correlation learning works.

The chlorophyll-a prediction is a highly nonlinear problem. Previous attempts using neural networks have achieved only limited success (Recknagel et al., 1999). According to Recknagel et al. (1999), the background of the problem can be described as follows:

Lake Kasumigaura is situated in the south-eastern part of Japan. It has a large and shallow water body where no thermal stratification occurs. The annual water temperature of the lake ranges from 4 to 30°C in summer. Due to high external and internal nutrient loadings, the primary productivity of the lake is extremely high,…, and favours harmful blue–green algae such as *Microcystis spp, Oscillatoria and Anabaena flos aquae*. As the algal succession changes the species abundance year by year, it is very difficult to causally determine and predict algal blooms in Lake Kasumigaura.

Accurate prediction of chlorophyll-a and other blue–green algae will no doubt be very useful in protecting the fresh-water environment.

3.1. Experimental setup

The limnological time series for 3 years between 1984 and 1986 were used in our experiments. There are 360-day data in each year. For each single day, the eight input conditions include water temperature, light, rotifer density, cladocera density, etc. The single output indicates the abundance of chlorophyll-a. The data in 1984 and 1985 were used as the training data to train the neural network ensemble. Then the neural network ensemble was tested on the 1986 data. Using the 1986 data as the testing data was suggested by Recknagel et al. (1999) because it represented typical year for blooms of *Microcystis*. More details about the data were given in (Recknagel et al., 1999).

As a pre-processing step, the original data were rescaled linearly to between 0.1 and 0.9. The input to each individual network consists of eight input conditions on the current day and seven output conditions on the past seven days. It should be pointed out that Recknagel et al. (1999) used a 5-vector input layer which included the current eight input conditions and those 32 input conditions of the present 10, 20, 30 and 40 days previously. The reason for changing the input is to reduce the number of input attributes and make the chlorophyll-a prediction problem more meaningful in the sense of time series prediction.

The normalised root-mean-square (RMS) error E was used to evaluate the performance of negative correlation learning, which is determined by the RMS value of the absolute prediction error for $\Delta t = 1$, divided by the standard deviation of $x(t)$ (Farmer & Sidorowich, 1987),

$$E = \frac{\langle [x_{\text{pred}}(t, \Delta t) - x(t + \Delta t)]^2 \rangle^{1/2}}{\langle (x - \langle x \rangle)^2 \rangle^{1/2}} \tag{9}$$

where $x_{\text{pred}}(t, \Delta t)$ is the prediction of $x(t + \Delta t)$ from the current state $x(t)$ and $\langle x \rangle$ represents the expectation of x. As indicated by Farmer and Sidorowich (1987), “if $E = 0$, the predictions are perfect; $E = 1$ indicates that the performance is no better than a constant predictor $x_{\text{pred}}(t, \Delta t) = \langle x \rangle$.”

The ensemble architecture used in the experiments has four networks. Each individual network is a feedforward network with one hidden layer. Both the hidden node function and the output node function are defined by the logistic function

$$\varphi(y) = \frac{1}{1 + \exp(-y)} \tag{10}$$

All the individual networks have ten hidden nodes. The number of training epochs was set to 2000.

3.2. Experimental results

Table 1 shows the average results of negative correlation learning and independent training (i.e. $\lambda = 0.0$ in negative correlation learning) over 25 runs for the chlorophyll-a prediction problem. Each run of negative correlation

Table 2
The correlations among the individual networks trained by negative correlation learning with different λ values

0	$\mathrm{Cor}_{12} = 0.21972$	$\mathrm{Cor}_{13} = 0.12713$	$\mathrm{Cor}_{14} = 0.24398$
	$\mathrm{Cor}_{23} = 0.17909$	$\mathrm{Cor}_{24} = 0.24678$	$\mathrm{Cor}_{34} = 0.11826$
0.25	$\mathrm{Cor}_{12} = 0.16973$	$\mathrm{Cor}_{13} = 0.07010$	$\mathrm{Cor}_{14} = 0.17965$
	$\mathrm{Cor}_{23} = 0.11519$	$\mathrm{Cor}_{24} = 0.17438$	$\mathrm{Cor}_{34} = 0.08683$
0.5	$\mathrm{Cor}_{12} = 0.11468$	$\mathrm{Cor}_{13} = 0.00593$	$\mathrm{Cor}_{14} = 0.09827$
	$\mathrm{Cor}_{23} = 0.03867$	$\mathrm{Cor}_{24} = 0.10509$	$\mathrm{Cor}_{34} = 0.06474$
0.75	$\mathrm{Cor}_{12} = 0.05957$	$\mathrm{Cor}_{13} = -0.02466$	$\mathrm{Cor}_{14} = 0.12302$
	$\mathrm{Cor}_{23} = 0.02883$	$\mathrm{Cor}_{24} = 0.33109$	$\mathrm{Cor}_{34} = -0.03763$
1	$\mathrm{Cor}_{12} = -0.37871$	$\mathrm{Cor}_{13} = -0.37402$	$\mathrm{Cor}_{14} = -0.21235$
	$\mathrm{Cor}_{23} = -0.22528$	$\mathrm{Cor}_{24} = -0.39379$	$\mathrm{Cor}_{34} = -0.41414$

learning was from different initial weights. The value of t-test between negative correlation learning and independent training with 24 degrees of freedom is -25.08 which is significant at $\alpha = 0.005$ by a two-tailed test, i.e. negative correlation learning is statistically significantly better than independent training.

In order to observe the effect of the correlation penalty terms, Table 2 shows the correlations among the individual networks trained by negative correlation learning with different λ values. The correlation between the network i and the network j is given by

$$\mathrm{Cor}_{ij} = \frac{\sum_{l=1}^{L}\sum_{k=1}^{K}(F_i^{(k)}(l) - \bar{F}_i(l))(F_j^{(k)}(l) - \bar{F}_j(l))}{\sqrt{\sum_{l=1}^{L}\sum_{k=1}^{K}(F_i^{(k)}(l) - \bar{F}_i(l))^2 \sum_{l=1}^{L}\sum_{k=1}^{K}(F_i^{(k)}(l) - \bar{F}_j(l))^2}} \tag{11}$$

where $F_i^{(k)}(l)$ is the output of the network i on the lth pattern in the testing set from the kth run, $\bar{F}_i(l)$ represents the average output of the network i on the lth pattern in the testing set, L is the number of patterns in the testing set, and K is the number of runs. There are

$$\binom{4}{2} = 6$$

correlations among different pairs of networks.

Table 3
Comparison of error rates between negative correlation learning ($\lambda = 1.0$) and independent training (i.e. $\lambda = 0.0$ in negative correlation learning) on the Australian credit card assessment problem. The results were averaged over 25 runs. "Simple Averaging" and "Winner-Takes-All" indicate two different combination methods used in negative correlation learning. Mean, SD, Min and Max indicate the mean value, standard deviation, minimum and maximum value, respectively

		Simple averaging		Winner-takes-all
		Training	Test	Test
$\lambda = 1.0$	Mean	0.0938	0.1337	0.1195
	SD	0.0031	0.0068	0.0052
	Min	0.0869	0.1163	0.1105
	Max	0.0985	0.1454	0.1279
$\lambda = 0.0$	Mean	0.0883	0.1386	0.1384
	SD	0.0308	0.0048	0.0049
	Min	0.0792	0.1279	0.1279
	Max	0.0965	0.1454	0.1512

The values of the correlations among the individual networks had relatively larger positive values when λ was 0. They reduced to relatively smaller positive values when λ was increased to 0.5. All of them became negative values when λ was further increased to 1. Overall, the results indicated that the neural networks trained by negative correlation learning tend to be negatively correlated. Because every individual network learns the same task in the independent training, the correlations among them are generally positive. In negative correlation learning, each individual network learns different parts or aspects of the training data so that the problem of correlated errors can be removed or alleviated. The empirical results match the theoretical analysis (Clemen & Winkler, 1985): when individual networks in an ensemble are unbiased, average procedures are most effective in combining them when errors in the individual networks are negatively correlated and moderately effective when the errors are uncorrelated. There is little to be gained from average procedures when the errors are positively correlated.

4. The Australian credit card assessment problem

This section describes the application of negative correlation learning to the Australian credit card assessment problem. The problem is to assess applications for credit cards based on a number of attributes. There are 690 patterns in total. The output has two classes. The 14 attributes include six numeric values and eight discrete ones, the latter having from 2 to 14 possible values. The Australian credit card assessment problem is a classification problem that is different from the regression type of tasks, such as the chlorophyll-a prediction problem, whose outputs are continuous. The data set was obtained from the UCI machine learning benchmark repository. It is available by anonymous ftp at ics.uci.edu (128.195.1.1) in directory /pub/machine-learning-databases.

4.1. Experimental setup

The data set was partitioned into two sets: a training set and a testing set. The first 518 examples were used for the training set, and the remaining 172 examples for the testing set. The input attributes were rescaled to between 0.0 and 1.0 by a linear function. The output attributes of all the problems were encoded using a 1-of-m output representation for m classes. The output with the highest activation designated the class.

The ensemble architecture used in the experiments has four networks. Each individual network is a feedforward

Table 4
The sizes of the correct response sets of individual networks created respectively by negative correlation learning ($\lambda = 1.0$) and independent training (i.e. $\lambda = 0.0$ in negative correlation learning) on the testing set and the sizes of their intersections for the Australian credit card assessment problem. The results were obtained from the first run among the 25 runs

$\lambda = 1.0$	$\Omega_1 = 147$	$\Omega_2 = 143$	$\Omega_3 = 138$
	$\Omega_4 = 143$	$\Omega_{12} = 138$	$\Omega_{13} = 124$
	$\Omega_{14} = 141$	$\Omega_{23} = 116$	$\Omega_{24} = 133$
	$\Omega_{34} = 123$	$\Omega_{123} = 115$	$\Omega_{124} = 133$
	$\Omega_{134} = 121$	$\Omega_{234} = 113$	$\Omega_{1234} = 113$
$\lambda = 0.0$	$\Omega_1 = 149$	$\Omega_2 = 147$	$\Omega_3 = 148$
	$\Omega_4 = 148$	$\Omega_{12} = 147$	$\Omega_{13} = 147$
	$\Omega_{14} = 147$	$\Omega_{23} = 147$	$\Omega_{24} = 146$
	$\Omega_{34} = 146$	$\Omega_{123} = 147$	$\Omega_{124} = 146$
	$\Omega_{134} = 146$	$\Omega_{234} = 146$	$\Omega_{1234} = 146$

network with one hidden layer. Both the hidden node function and the output node function are defined by the logistic function in Eq. (10). All the individual networks have ten hidden nodes. The number of training epochs was set to 250. The strength parameter λ was set to 1.0. These parameters were chosen after limited preliminary experiments. They are not meant to be optimal.

4.2. Experimental results

Table 3 shows the average results of negative correlation learning over 25 runs. Each run of negative correlation learning was from different initial weights. The ensemble with the same initial weight setup was also trained using BP without the correlation penalty terms (i.e. $\lambda = 0.0$ in negative correlation learning). Results are also shown in Table 3. For this problem, the simple averaging defined in Eq. (1) was first applied to decide the output of the ensemble system. For the simple averaging, it was surprising that the results of negative correlation learning with $\lambda = 1.0$ were similar to those of independent training. This phenomenon seems contradictory to the claim that the effect of the correlation penalty term is to encourage different individual networks in an ensemble to learn different parts or aspects of the training data. In order to verify and quantify this claim, we compared the outputs of the individual networks trained with the correlation penalty terms to those of the individual networks trained without the correlation penalty terms.

Table 5
Comparison among negative correlation learning (NCL), EPNet (Yao & Liu, 1997), an evolutionary ensemble learning algorithm (Evo-En-RLS) (Yao & Liu, 1998), and others (Michie, Spiegelhalter & Taylor, 1994) in terms of the average testing error rate for the Australian credit card assessment problem. TER stands for Testing Error Rate in the table

Algorithm	TER	Algorithm	TER
NCL	0.120	DIPOL92	0.141
EPNet	0.115	Discrim	0.141
Evo-En-RLS	0.095	Logdisc	0.141
Cal5	0.131	Cart	0.145
Itrule	0.137	RBF	0.145
Castle	0.148	NaiveBay	0.151
IndCART	0.152	BP	0.154

Two notions were introduced to analyse negative correlation learning. They are the correct response sets of individual networks and their intersections. The correct response set S_i of the individual network i on the testing set consists of all the patterns in the testing set which are classified correctly by the individual network i. Let Ω_i denote the size of set S_i, and $\Omega_{i_1 i_2 \ldots i_k}$ denote the size of set $S_{i_1} \cap S_{i_2} \cap \cdots \cap S_{i_k}$. Table 4 shows the sizes of the correct response sets of individual networks and their intersections on the testing set, where the individual networks were, respectively, created by negative correlation learning and independent training. It is evident from Table 4 that the different individual networks created by negative correlation learning were able to specialise to different parts of the testing set. For instance, in Table 4 the sizes of both correct response sets S_2 and S_4 at $\lambda = 1.0$ were 143, but the size of their intersection $S_2 \cap S_4$ was 133. The size of $S_1 \cap S_2 \cap S_3 \cap S_4$ was only 113. In contrast, the individual networks in the ensemble created by independent training were quite similar. The sizes of correct response sets S_1, S_2, S_3, and S_4 at $\lambda = 0.0$ were from 146 to 149, while the size of their intersection $S_1 \cap S_2 \cap S_3 \cap S_4$ reached 146. There were only three different patterns correctly classified by the four individual networks in the ensemble.

In simple averaging, all the individual networks have the same combination weights and are treated equally. However, not all the networks are equally important. Because different individual networks created by negative correlation learning were able to specialise to different parts of the testing set, only the outputs of these specialists should be considered to make the final decision of the ensemble for this part of the testing set. In this experiment, a winner-takes-all method was applied to select such networks. For each pattern of the testing set, the output of the ensemble was only decided by the network whose output had the highest activation. Table 3 shows the average results of the negative correlation learning over 25 runs using the winner-takes-all combination method. The winner-takes-all combination method improved the negative correlation learning significantly because there were good and poor networks for each pattern in the testing set and winner-takes-all selected the best one. However it did not improved the independent training much because the individual networks created by the independent training were all similar to each other.

Table 5 compares the results of negative correlation learning with those produced by other neural and nonneural algorithms, where EPNet is an evolutionary system for designing neural networks (Yao & Liu, 1997) and Evo-En-RLS forms the final results by combining all the individuals in the last generation in EPNet based on the recursive least-square algorithm (Yao & Liu, 1998). The other algorithms represent the best 11 out of the 23 algorithms tested by Michie et al. (1994). Although negative correlation learning performed slightly worse than EPNet and Evo-En-RLS, it was significantly faster in terms of training time. Negative correlation learning performed better than all other algorithms.

5. Conclusions

This paper first introduces the negative correlation learning approach to designing neural network ensembles. Then it analyses negative correlation learning in terms of correlations on the chlorophyll-a prediction problem. Unlike other ensemble approaches which try to create unbiased individual networks whose errors are uncorrelated, negative correlation learning can produce individual networks whose errors tend to be negatively correlated. For regression type of tasks, such as the chlorophyll-a prediction, only simple average combination method was investigated since winner-takes-all is not suitable. For classification type of tasks, such as the Australian credit card assessment problem, both simple average and winner-takes-all combination methods were investigated. Compared with simple averaging, winner-takes-all fits negative correlation learning well because only the best individual rather than all individuals for each pattern is considered to make the final decision of the ensemble.

There are, however, some issues that need resolving. The architectures of the ensembles and the λ parameter in negative correlation learning are predefined at the moment. One of the future improvements to negative correlation learning would be to make them adaptive.

Acknowledgements

The authors are grateful to anonymous referees for their constructive comments that have helped to improve the paper.

References

Clemen, R. T., & Winkler, R. L. (1985). Limits for the precision and value of information from dependent sources. *Operations Research, 33*, 427–442.

Drucker, H., Cortes, C., Jackel, L. D., LeCun, Y., & Vapnik, V. (1994). Boosting and other ensemble methods. *Neural Computation*, *6*, 1289–1301.

Farmer, J. D., & Sidorowich, J. J. (1987). Predicting chaotic time series. *Physical Review Letters*, *59*, 845–847.

Jacobs, R. A. (1997). Bias/variance analyses of mixture-of-experts architectures. *Neural Computation*, *9*, 369–383.

Liu, Y., & Yao, X. (1999). Simultaneous training of negatively correlated neural networks in an ensemble. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *29* (6), 000.

Michie, D., Spiegelhalter, D. J., & Taylor, C. C. (1994). *Machine learning, neural and statistical classification*, (pp. 134–135). Chichester, UK: Ellis Horwood.

Perrone, M., & Cooper, L. N. (1993). When networks disagree: ensemble methods for hybrid neural networks. In R. J. Mammone, *Neural networks for speech and image pressing*, London: Chapman & Hall.

Recknagel, F., Fukushima, T., Hanazato, T., Takamura, N., & Wilson, H. (1999). Modelling and prediction of phyto- and zooplankton dynamics in Lake Kasumigaura by artificial neural networks. *Lakes & Reservoirs.*

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In D. E. Rumelhart & J. L. McClelland, *Parallel distributed processing: explorations in the microstructures of cognition*, I. (pp. 318–362). Cambridge, MA: MIT Press.

Sharkey, A. J. C. (1996). On combining artificial neural nets. *Connection Science*, *8*, 299–313.

Yao, X., & Liu, Y. (1997). A new evolutionary system for evolving artificial neural networks. *IEEE Transactions on Neural Networks*, *8*, 694–713.

Yao, X., & Liu, Y. (1998). Making use of population information in evolutionary artificial neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *28*, 417–425.