Neural Comput & Applic (2004) 13: 112–122
DOI 10.1007/s00521-004-0413-4

ORIGINAL ARTICLE

Imran Maqsood · Muhammad Riaz Khan
Ajith Abraham

An ensemble of neural networks for weather forecasting

Received: 7 April 2004 / Accepted: 20 April 2004 / Published online: 20 May 2004

Abstract This study presents the applicability of an ensemble of artificial neural networks (ANNs) and learning paradigms for weather forecasting in southern Saskatchewan, Canada. The proposed ensemble method for weather forecasting has advantages over other techniques like linear combination. Generally, the output of an ensemble is a weighted sum, which are weight-fixed, with the weights being determined from the training or validation data. In the proposed approach, weights are determined dynamically from the respective certainties of the network outputs. The more certain a network seems to be of its decision, the higher the weight. The proposed ensemble model performance is contrasted with multi-layered perceptron network (MLPN), Elman recurrent neural network (ERNN), radial basis function network (RBFN), Hopfield model (HFM) predictive models and regression techniques. The data of temperature, wind speed and relative humidity are used to train and test the different models. With each model, 24-h-ahead forecasts are made for the winter, spring, summer and fall seasons. Moreover, the performance and reliability of the seven models are then evaluated by a number of statistical measures. Among the direct approaches employed, empirical results indicate that HFM is relatively less accurate and RBFN is relatively more reliable for the weather forecasting problem. In comparison, the ensemble of neural networks produced the most accurate forecasts.

I. Maqsood (✉)
Faculty of Engineering, University of Regina,
Regina, Canada, SK S4S 0A2
E-mail: maqsoodi@uregina.ca
Tel.: +1-306-5699533
Fax: +1-306-5699544

M. R. Khan (✉)
AMEC Technologies Training and Development Services,
400-111 Dunsmuir Street, Vancouver, Canada, BC V6B 5W3
E-mail: riaz.khan@amec.com

A. Abraham (✉)
Computer Science Department, Oklahoma State University, USA
E-mail: ajith.abraham@ieee.org

Keywords Artificial neural networks · Ensembles · Forecasting · Model · Weather

1 Introduction

The weather is a continuous, data-intensive, multi-dimensional, dynamic and chaotic process, and these properties make weather forecasting a formidable challenge. Generally, two methods are used to forecast weather: (a) the empirical approach and (b) the dynamical approach [16]. The first approach is based upon the occurrence of analogues and is often referred to by meteorologists as analogue forecasting. This approach is useful for predicting local-scale weather if recorded cases are plentiful. The second approach is based upon equations and forward simulations of the atmosphere, and is often referred to as computer modelling. Because of the grid coarseness, the dynamical approach is only useful for modelling large-scale weather phenomena and may not predict short-term weather efficiently. Most weather prediction systems use a combination of empirical and dynamical techniques. However, little attention has been paid to the use of artificial neural networks (ANNs) in weather forecasting [13, 18-20].

ANNs provide a methodology for solving many types of non-linear problems that are difficult to solve by traditional techniques. Most meteorological processes often exhibit temporal and spatial variability, and are further plagued by issues of non-linearity of physical processes, conflicting spatial and temporal scale and uncertainty in parameter estimates. With ANNs, there exists the capability to extract the relationship between the inputs and outputs of a process, without the physics being explicitly provided [28]. Thus, these properties of ANNs are well suited to the problem of weather forecasting under consideration.

An ensemble neural network is a learning paradigm where a collection of a finite number of neural networks

is trained for the same task [2-4, 25]. It originates from Hansen and Salamon's work [6], which shows that the generalisation ability of a neural network system can be significantly improved through an ensemble of neural networks, i.e. training many neural networks and then combining their predictions. In general, a neural network ensemble is constructed in two steps, i.e. training a number of component neural networks and then combining the component predictions. Ensemble methods combine the outputs of several neural networks [5, 9, 21]. The output of an ensemble is a weighted average of the outputs of each network, with the ensemble weights determined as a function of the relative error of each network determined in training [10, 17, 21]; the resulting network often outperforms the constituent networks. There is a growing body of research into ensemble methods, for example, improvements in performance can result from training the individual networks to be decorrelated with each other [7, 8, 22, 26] with respect to their errors. We present a novel approach to determine the ensemble weights dynamically as part of the training algorithm, i.e. during each propagation through the network, as opposed to any pre-determined fixed values or calculations. The weights are proportional to the certainty of the respective outputs. The certainty of a network output measures how close the output is to one or the other of the target values. For example, suppose we have trained two back-propagation networks to output 1 if the input is in a particular class and 0 otherwise. Suppose we present a previously unseen input to the networks and the outputs are 0.6 and 0.9 for the first and second, respectively. With a threshold for decision of 0.5, both outputs would lead us to conclude that the input is in the class, but the first network seems much less certain than the second.

As an extension to the previous efforts, the objective of this study is to develop ensembles of ANNs for weather analysis in southern Saskatchewan, Canada. This development will be based on: (a) multi-layered perceptron network (MLPN), Elman recurrent neural network (ERNN), radial basis function network (RBFN) and Hopfield model (HFM) predictive models for forecasting hourly temperature, wind speed and relative humidity in winter, spring, summer and fall seasons; (b) examination of the applicability of the ANN approach for weather forecasting; (c) comparison of the proposed predictive models with regression models; and (d) performance quantification of the developed models, their ensembles and the regression models based on a number of statistical measures.

2 Related research work

This section summarises some of the past research done in ensemble methods, in terms of classification. Consider a population of n networks trained on a set $A=(x_m,y_m)$ of labelled instances of a binary classification problem.

2.1 The naive classifier

Let the function computed by the ith network be $f_i(x)$. If the networks are trained to give an output of 0 or 1 for a negative or positive classification, respectively, we can use a threshold of 0.5 on the output of a network to decide the class for an instance of the problem. The naive approach usually used a cross validation set [12] $CV=(x_l,y_l)$ and picked the network, f_{Naive}, that minimises the mean squared error (MSE) on CV. The MSE for each network is:

$$\text{MSE}[f_i] = E_{\text{CV}}\left[(y_m - f_i(x_m))^2\right] \tag{1}$$

This technique throws out any knowledge contained in the other networks. Although f_{Naive} has the best overall performance on the cross validation set, some of the other f_is may lead to correct classification where f_{Naive} does not.

2.2 The basic ensemble method

A simple approach to combining network outputs is to simply average them together. The basic ensemble method (BEM) output is defined by:

$$f_{\text{BEM}} = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{2}$$

This approach by itself can lead to improved performance [14, 21], but does not take into account the fact that some networks may be more accurate than others. It has the advantage of being easy to understand and implement [1, 15] and can be shown not to increase the expected error [1, 23, 24].

2.3 The generalised ensemble method

A generalisation to the BEM method is to find weights for each output that minimises the MSE of the ensemble. The general ensemble model (GEM) is defined by:

$$f_{\text{GEM}} = \sum_{i=1}^{n} \alpha_i f_i(x) \tag{3}$$

where the α_is are chosen to minimise the MSE with respect to the target function, f (estimated using the cross validation set), and sum to 1. Define the error, $\varepsilon_i(x)$, of a network, f_i, as $\varepsilon_i(x)=f(x)-f_i(x)$. Define the correlation matrix, $C_{ij}=E\lfloor\varepsilon_i(x)\varepsilon_j(x)\rfloor$. Then, we must find weights, α_i, that minimises the following:

$$\text{MSE}[f_{\text{GEM}}] = \sum_{i=1}^{n}\sum_{j=1}^{n} \alpha_i \alpha_j C_{ij} \tag{4}$$

It can be shown [1, 21] that the optimal choice for the is follow

$$\alpha_i = \frac{\sum_{j=1}^{n} C_{ij}^{-1}}{\sum_{k=1}^{n}\sum_{j=1}^{n} C_{kj}^{-1}} \quad (5)$$

This method yields better performance than BEM, and can be shown to always give a better estimate than the naive classifier [21]. However, this method depends on a reliable estimate of C and the fact that it is non-singular so that it can be easily inverted [21]. In practice, errors are often highly correlated, thus, the rows of C are nearly linearly dependent so that inverting C leads to significant round-off errors. Some techniques to avoid this include ignoring networks whose errors are highly correlated with others [21], using specialised techniques for the inversion of near-singular matrices and training the networks to be decorrelated with each other [22, 26].

2.4 Dynamic ensemble method

2.4.1 Certainty

If the output of a neural network, $y = f_i(x)$, can be interpreted as the probability that an instance, x, is in a class, then, as y approaches 1, we feel more certain that the instance is in the class. As y approaches 0, we become more certain that the instance is not in the class. Let us quantify this notion; we define the *certainty*, $c(y)$, of a neural network output as:

$$c(y) = \begin{cases} y & \text{if } y \geqslant 0.5 \\ 1 - y & \text{otherwise} \end{cases} \quad (6)$$

The certainty rises for output $y < 0.5$ as y falls, and for outputs $y \geq 0.5$ as y rises. We say one network output, y_1, is *less certain* than another, y_2, if $c(y_1) < c(y_2)$. Note that the certainty behaves symmetrically with respect to positive and negative decisions; the certainty of an output of 0.1 is the same as that of an output of 0.9, but the decision they are certain about is different.

2.4.2 Dynamically averaging networks

If, instead of choosing static weights derived from f_is performance on a sample of the input space, we allow the weights to adjust to be proportional to the certainties of the respective network outputs, we might achieve better performance. We define the dynamically averaged network (DAN) by:

$$f_{\mathrm{DAN}} = \sum_{i=1}^{n} w_i f_i(x) \quad (7)$$

where the w_is are according to:

$$w_i = \frac{c(f_i(x))}{\sum_{j=1}^{n} c(f_i(x))} \quad (8)$$

The w_is sum to 1, so f_{DAN} is a weighted average (WA) of the network outputs. The difference is that the weight vector is recomputed each time the ensemble output is evaluated, to try to give the best decision for the particular instance under consideration, instead of statically choosing weights that give an optimal decision with respect to a cross validation set. Each network's contribution to the sum is proportional to its certainty. A value close to 0.5, for instance, would contribute very little to the sum while a very certain value of 0.99 (or 0.01) among many less certain values would dominate the sum. This method is similar to the idea of using agreement among a set of classifiers to obtain a measure of confidence in the decision, but the confidence level (certainty) of each classifier itself is used to obtain the final decision. Each fully connected ANN in the ensemble is generated with random initial weights. Then, each ANN is trained partially with training data and tested with the validation data.

2.5 The proposed ensemble neural network algorithm

The back-propagation networks set the initial weights at random and train to decrease the MSE. Differences in the initial weights gives different results. Thus, neural network ensembles [21] integrate these independent neural networks to improve the generalisation capability. This method has a guarantee of accuracy improvement from the viewpoint of bias/variance decomposition of the MSE [27].

Consider a single NN that has been trained on a given data set. Let x denote an input vector not seen before and let d denote the corresponding desired response; x and d represent realisations of the random vector X and random variable D, respectively. Let $F(x)$ denote the input-output function realised by the network. Then, in light of the material on the bias/variance dilemma [27], we decompose the MSE between $F(x)$ and the conditional expectation, $E[D|X = x]$, into its bias and variance components as follows:

$$E_D\left[(F(x) - E[D|X = x])^2\right] = B_D(F(x)) + V_D(F(x)) \quad (9)$$

where $B_D(F(x))$ is the bias squared:

$$B_D(F(x)) = (E_D[F(x)] - E[D|X = x])^2 \quad (10)$$

and $V_D(F(x))$ is the variance:

$$V_D(F(x)) = E_D\left[(F(x) - E_D[F(x)])^2\right] \quad (11)$$

The expectation, E_D, is taken over the space D, defined as the space encompassing the distribution of all training sets (for example, inputs and target outputs) and the distribution of all initial conditions.

There are different ways of individually training the networks and also different ways of combining their outputs. Here, we consider the situation where the networks have an identical configuration, but they are trained starting from different initial conditions. For the combiner at the output of the neural network ensembles,

we used a simple ensemble average. Let ψ denote the space of all initial conditions. Let F_I (x) denote the average of the input-output functions of the networks over a representative number of initial conditions. By analogy with Eq. 9, we may write:

$$E_\psi\left[(F_I(X) - E[D|X = x])^2\right] = B_\psi(F(x)) + V_\psi(F(x)) \tag{12}$$

where B_ψ $(F(x))$ is the squared bias defined over the space ψ:

$$B_\psi(F(x)) = \left(E_\psi[F_I(x)] - E[D|X = x]\right)^2 \tag{13}$$

and V_ψ $(F(x))$ is the corresponding variance:

$$V_\psi(F(x)) = E_\psi\left[\left(F_I(x) - E_\psi[F(x)]\right)^2\right] \tag{14}$$

The expectation, E_ψ, is taken over the space ψ.

From the definition of space D, we may view it as the product of the space of initial conditions, ψ, and the remnant space denoted by D'. Accordingly, we may write, again by analogy to Eq. 9:

$$E_{D'}\left[(F_I(x) - E[D|X = x])^2\right] = B_{D'}(E_I(x)) + V_{D'}(F_I(x)) \tag{15}$$

where $B_{D\cdot}$ $(F_I(x))$ is the squared bias defined over the remnant space, D':

$$B_{D'}(F_I(x)) = (E_{D'}[F_I(x)] - E[D|X = x])^2 \tag{16}$$

and $V_{D\cdot}$ $(F_I(x))$ is the corresponding variance:

$$V_{D'} = E_{D'}\left[(F_I(x) - E_{D'}[F_I(x)])^2\right] \tag{17}$$

From the definitions of spaces D, ψ and D', we readily see that:

$$E_{D'}[F_I(x)] = E_D[F(x)] \tag{18}$$

It follows, therefore, that Eq. 16 may be rewritten in the equivalent form:

$$B_{D'}(F_I(x)) = (E_D[F(x)] - E[D|X = x])^2 = B_D(F(x)) \tag{19}$$

Consider the variance $V_{D\cdot}$ $(F_I(x))$ of Eq. 17. Since the variance of a random variable is equal to the mean square value of that random variable minus its bias squared, we may equivalently write:

$$V_{D'}(F_I(x)) = E_{D'}\left[(F_I(x))^2\right] - (E_{D'}[F_I(x)])^2 = E_{D'}[(F_I(x))] \tag{20}$$

Similarly:

$$V_D(F_I(x)) = E_D\left[(F(x))^2\right] - (E_D[F(x)])^2 \tag{21}$$

Note that the mean square value of the function $F(x)$ over the entire space, D, is destined to be equal to or greater than the mean square value of the entire averaged function, F_I (x), over the remnant space, D'. That is:

$$E_D\left[(F(x))^2\right] \geqslant E_{D'}\left[(F_I(x))^2\right] \tag{22}$$

In light of this inequality, comparison of Eqs. 20 and 21 immediately reveals that

$$V_{D'}(F_I(x)) \leqslant V_D(F(x)) \tag{23}$$

Thus, from Eqs. 19 and 23, we draw two conclusions:

- The bias of the ensemble-averaged function, F_I (x), pertaining to the multiple classifier systems, is exactly the same as that of the function $F(x)$ pertaining to a single neural network.
- The variance of the ensemble-averaged function, F_I is less than that of the function $F(x)$.

Ensembles of linear networks have demonstrated improved performance over individual networks, but linear models have problems due to limited capacity. Ensembles of more complex well-trained networks offer a promising alternative. An ensemble of non-linear feed-forward neural networks generated by a constructive algorithm is presented in this paper. A similar approach could be extended to ERNN, RBFN and HFM networks. The ensemble method presented exhibits better generalisation than linear ensembles, and shows promise towards a reduction in time-complexity over well-trained ensembles. We also used a winner-take-all (WTA) approach to compare with the WA approach.

3 Experiment setup

The weather parameters, including temperature, wind speed and relative humidity, recorded at the Regina Airport were collected by the Meteorological Department, Environment Canada in 2001 for developing and analysing the intelligent-based forecasting models. To examine the seasonal variations, the available weather data were split into four seasons, namely, winter (December-February), spring (March-May), summer (June-August) and fall (September-November). For all four models (i.e. MLPN, ERNN, RBFN and HFM), hourly seasonal data were used for training the networks. Training data sets of December 1, 2000-February 25, 2001; March 1-May 5, 2001; January 1-August 6, 2001; and September 1-November 9, 2001 were used for the winter, spring, summer and fall seasons, respectively. The hourly data sets of February 26, May 6, August 7 and November 10 were selected as typical days (test data sets) for winter, spring, summer and fall, respectively, in order to test the trained models. In fact, these typical days represent one of the severe weather days during the

winter, spring, summer and fall seasons. The idea behind this selection was to examine the applicability of the proposed ANN paradigm on these days. Moreover, these days exhibit a larger variation in temperature, wind speed and humidity within the 24-h period compared to other normal days within the forecasting seasons. A Pentium-III, 1 GHz processor computer with 256 MB RAM was used to simulate all of the experiments using MATLAB version 5.3.

Neural networks generally provide improved performance with the normalised data. The use of original data as the input to the neural network may cause a convergence problem [11]. All of the weather data sets were, therefore, transformed into values between -1 and 1 by dividing the difference between the actual and minimum values by the difference between the maximum and minimum values. The main goal of normalisation, in combination with weight initialisation, is to allow the squashed activity function to work at least at the beginning of the learning phase. Thus, the gradient, which is a function of the derivative of the non-linearity, will always be different from zero. At the end of each algorithm, the outputs were denormalised into the original data format for achieving the desired result.

The configuration of the neural network depends highly on the problem. Therefore, it is left with the designer to choose an appropriate number of hidden layers and hidden layer nodes based on his/her experience. Thus, an appropriate architecture is determined for each application using the trial and error method. In this study, the training of different neural networks took from a few seconds to 30 min with a Pentium-III, 1 GHz processor computer. The learning rate parameter and momentum term were adjusted intermittently to speed up the convergence. In order to keep the simplicity of the modelling structure, only one hidden layer with 72 hidden layer nodes was used for the MLPN and ERNN models. This number was arrived at after analysing 24, 48, 72, 98 and 120 neurons in the hidden layer. The architecture with 24 and 48 neurons in the hidden layer was faster in computation, but the convergence rate was very slow. The architecture with 98 and 120 neurons in the hidden layer was converging equally well as that with 72 neurons. Therefore, the architecture with 72 neurons in the hidden layer was selected. It was noted that, with the increased number of hidden layers, the convergence rate of the MLPN and ERNN models was decreased. On the other hand, two hidden layers with 180 nodes were chosen for training the RBFN model. The input matrix contains 24 inputs presenting the hourly-predicted weather parameters for one day in each season, thus, 96 inputs constitute our algorithm for all four seasons.

The MLPN, ERNN, RBFN and HFM networks were used after deciding the relevant input/output parameters, training/testing data sets and learning algorithms. To decide on the architectures of the MLPN, HFM and ERNN networks, a trial and error approach was used. Networks were trained for a fixed number of epochs, and the error gradient was observed over these epochs. Performance of the MLPN, HFM and ERNN networks were evaluated by increasing or decreasing the number of hidden nodes. Since no significant reduction in error was observed beyond 45 hidden nodes, a single hidden layer network comprising 45 neurons was identified. The input and output values were scaled between -1 and $+1$, and the one-step-secant learning algorithm was used for training the MLPN and ERNN networks. The activation functions for the MLPN and ERNN models were chosen to be log-sigmoid and hyperbolic-tangent-sigmoid for hidden units, respectively, and pureline for the output units. Since there is no exact rule for fixing the number of hidden neurons and hidden layers to avoid underfitting or overfitting in the MLPN and ERNN networks, therefore, the RBFN model is investigated to address this difficulty. In RBFN, the numbers of hidden layers and neurons selected by the model were two and 180, respectively; the Gaussian activation function was chosen for the hidden units, and pureline for the output units.

Figure 1 presents the architecture of the ensemble neural network. The ensemble improves the stability and accuracy of the model. The ANNs mentioned in the figure could be different combinations of MLPN, ERNN, RBFN or HFM architectures.

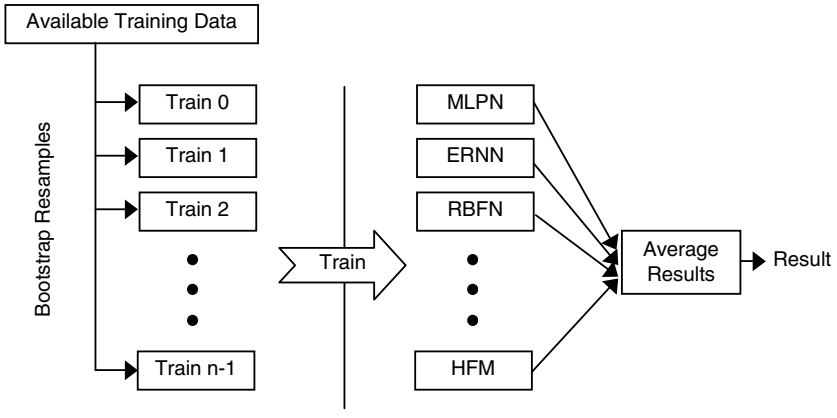

Fig. 1 Architecture of the ensemble neural network

4 Results analysis

4.1 Weather analysis using ANNs and ensemble methods

The obtained results indicate that satisfactory prediction accuracy has been achieved through the MLPN, ERNN, RBFN and HFM models, as well as the WA and WTA ensemble methods for temperature, wind speed and relative humidity parameters during winter, spring, summer and fall seasons. All of the obtained results were analysed, compared and evaluated in the following subsections. Our experiments were mainly aimed to investigate the following aspects: (1) to prove that the use of an ensemble neural network allows one to obtain satisfactory classification accuracy with short designing phases; (2) to compare the performance of the proposed combination method with the ones provided by individual methods based on the assumptions of independent errors.

Comparison of the neural networks and their ensembles for the forecasting of temperature, wind speed and relative humidity in the summer season are shown in Fig. 2. Such figures of comparisons for the fall, winter and spring seasons are not presented in this paper in order to avoid repitition of similar graphs. Their results are provided in Tables 1 and 2 and analysed in the following sections.

Figure 2 indicates that MLPN can achieve useful weather forecasting results in an efficient way.

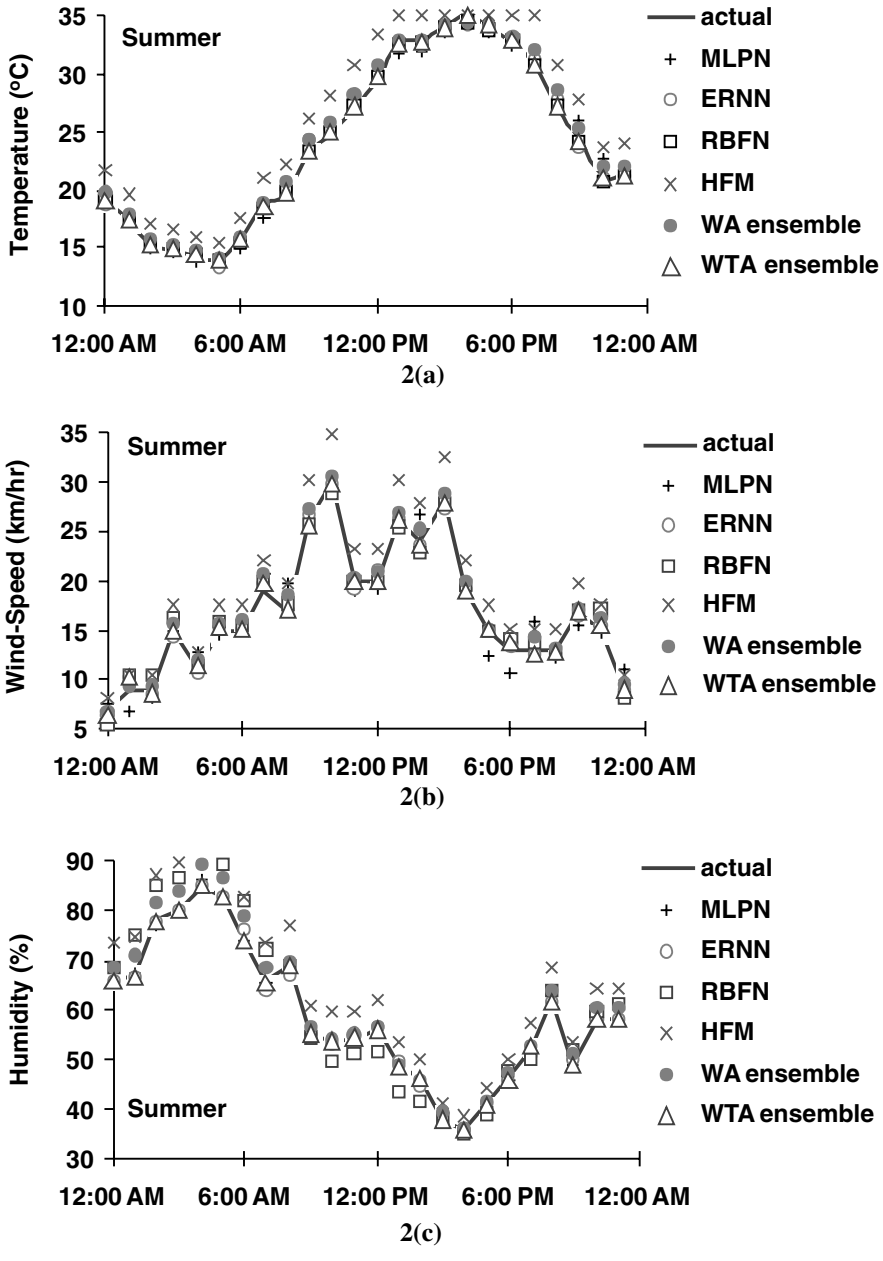

Fig. 2a-c Comparison of the neural network and their ensembles for 24-h-ahead forecasting of **a** temperature, **b** wind speed and **c** relative humidity on August 7, 2001

Table 1 Performance of MLPN, ERNN, RBFN and HFM for forecasting temperature, wind speed and humidity

Season	Reliability parameter	Temperature (°C)				Wind speed (km/h)				Humidity (%)			
		MLPN	ERNN	RBFN	HFM	MLPN	ERNN	RBFN	HFM	MLPN	ERNN	RBFN	HFM
Winter	MAD	0.5685	0.5610	0.5242	2.9894	0.8505	0.8158	0.7246	2.8120	0.6285	0.6249	0.5669	3.5575
	RMSE	0.0200	0.0199	0.0060	0.0456	0.0199	0.0199	0.0131	0.0327	0.0200	0.0199	0.0071	0.0395
	CC	0.9919	0.9883	0.9997	1.0000	0.9526	0.9032	0.9192	0.9872	0.9712	0.9738	0.9845	0.9982
	Training time (s)	1,140	1,500	3	2	600	480	3	2	900	1,500	3	2
	Iteration number	17678	16983	-	-	11909	4894	-	-	13742	7513	-	-
Spring	MAD	0.7958	0.7898	0.9296	3.2721	0.9012	0.8888	0.8678	3.2832	1.1975	1.1699	1.1561	7.7781
	RMSE	0.0200	0.0199	0.0032	0.1184	0.0200	0.0199	0.0044	0.1543	0.0200	0.0199	0.0054	0.1537
	CC	0.9436	0.9461	0.9992	1.0000	0.9948	0.9974	0.9975	0.9994	0.9976	0.9945	0.9965	0.9999
	Training time (s)	600	900	3	2	420	30	3	2	1,500	1,920	3	2
	Iteration number	9335	10173	-	-	5514	3271	-	-	21055	19597	-	-
Summer	MAD	0.5259	0.4699	0.4641	2.3151	0.8784	0.8794	0.8615	2.5811	1.1211	1.1349	1.2937	6.9252
	RMSE	0.0199	0.0199	0.0050	0.0650	0.0200	0.0199	0.0119	0.0365	0.0199	0.0199	0.0155	0.2029
	CC	0.9937	0.9976	0.9996	0.9897	0.9691	0.9603	0.9900	0.9892	0.9877	0.9879	0.9891	0.9852
	Training time (s)	1,020	1,260	3	2	240	540	3	2	1,140	1,800	3	2
	Iteration number	20109	12383	-	-	3904	6090	-	-	16500	18249	-	-
Fall	MAD	0.7100	0.6872	0.6739	1.0242	0.8331	0.8633	0.8258	3.0417	1.1910	1.1862	1.0765	6.8769
	RMSE	0.0199	0.0199	0.0169	0.0797	0.0199	0.0199	0.0092	0.0533	0.0200	0.0199	0.0120	0.1844
	CC	0.9902	0.9853	0.9961	1.0000	0.9909	0.9851	0.9983	1.0000	0.9962	0.9958	0.9991	1.0000
	Training time (s)	660	1,620	3	2	240	420	3	2	1,680	1,800	3	2
	Iteration number	10117	13523	-	-	3812	4777	-	-	20731	12614	-	-

Table 2 Performance comparison of WTA ensemble, WA ensemble and statistical method for the weather forecasting

Model	Temperature (°C)			Wind speed (km/h)			Humidity (%)		
	MAD	RMSE	CC	MAD	RMSE	CC	MAD	RMSE	CC
Winter season									
WTA ensemble	0.0783	0.0053	0.9996	0.3193	0.0119	0.9961	0.1160	0.0069	0.9984
WA ensemble	0.3062	0.0197	0.9982	0.5414	0.0189	0.9895	0.3049	0.0198	0.9940
Statistical method	3.7964	4.5307	0.2247	7.5248	8.7423	0.0035	1.9534	2.9857	0.2715
Spring season									
WTA ensemble	0.0912	0.0030	0.9993	0.1082	0.0041	0.9999	0.1420	0.0051	0.9999
WA ensemble	1.1867	0.0198	0.9890	1.3862	0.0183	0.9997	1.5509	0.0143	0.9997
Statistical method	2.9584	3.8245	0.5828	6.5478	7.2354	0.0437	3.1254	3.3325	0.3502
Summer season									
WTA ensemble	0.1127	0.0049	0.9998	0.3223	0.0108	0.9977	0.1903	0.0151	0.9998
WA ensemble	0.6558	0.0098	0.9974	0.7724	0.0189	0.9976	2.0051	0.0112	0.9983
Statistical method	5.3364	4.3568	0.0895	6.3547	5.1254	0.0329	6.2554	5.0210	0.0827
Fall season									
WTA ensemble	0.2958	0.0151	0.9978	0.2560	0.0089	0.9990	0.2645	0.0118	0.9995
WA ensemble	0.7416	0.0195	0.9965	0.6130	0.0187	0.9973	2.0848	0.0196	0.9993
Statistical method	3.1548	2.3658	0.4727	7.6658	6.3225	0.0087	6.0321	4.8644	0.1237

Compared to the HFM results, MLPN exhibited lower errors. It is capable of representing non-linear functions better than the single-layered perceptron. However, the learning process of the MLPN algorithm is time-consuming and its performance is heavily dependent on the network parameters like learning rate and momentum.

The ERNN model, compared to MLPN, could efficiently capture the dynamic behaviour of the weather, resulting in a more compact and natural representation of the temporal information contained in the weather profile. The RMS error of the ERNN model was much lower than that of the HFM method. The recurrent network took more training time, but this depends on the data size and the number of network parameters. It can be inferred that the ERNN network could yield more accurate results, if good data-selection strategies, training paradigms and network input and output representations are determined properly.

The RBFN network can give the best overall results in terms of accuracy and training time. It is better correlated compared to the MLPN, ERNN and HFM networks. The proposed RBFN network can also overcome several limitations of MLPN and ERNN, such as a highly non-linear weight update and the slow convergence rate. Since RBFN has natural unsupervised learning characteristics and a modular network structure, these properties make it more effective for fast and robust weather forecasting.

It is indicated that the HFM model overestimated most of the predicted values. Overall, the performance of HFM is reasonable. However, compared to the other models, it is less accurate for the weather forecasting problem.

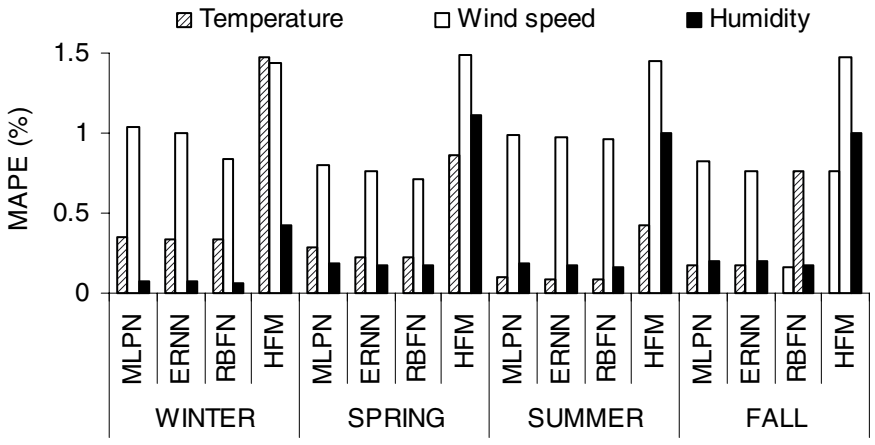

Fig. 3 Comparison of different neural networks for MAPEs

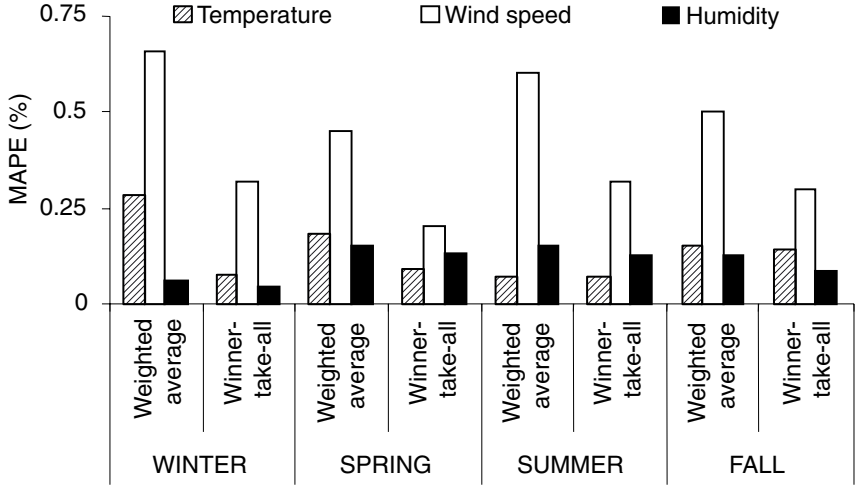

Fig. 4 Comparison of WA and WTA ensembles for MAPEs

The optimal network is the one that has a lower error and a reasonable learning time. Prediction reliability of the four models was evaluated based on a number of statistical analyses, as shown in Table 1. The training time for the MLPN and ERNN models ranged from 5 min to 30 min, while it took only a few seconds to run the training for RBFN and HFM. The ERNN model took more training time and had a relatively fewer number of iterations compared to the MLPN model. With the improvement of computing speed, the training time due to different algorithms may no longer be such a crucial factor if the training record is not too long and the design architecture is not too complicated. The testing accuracy could get worse if the selection of the algorithm to represent the problem is not properly decided.

Figure 3 indicates that, for the winter and spring seasons, humidity was predicted with the lowest mean absolute percentage error (MAPE) values by the four soft computing models, whereas, for the summer and fall seasons, temperature predictions were the most accurate with the lowest MAPE values. In comparison, humidity predictions represented the least accurate results (i.e. higher MAPE values) for all of the four seasons. The different levels of MAPE associated with temperature, wind speed and humidity could be due to the variability of data patterns in the four seasons, as well as the variations of the modelling structures. Overall, the RBFN model produced the lowest MAPE (i.e. the most accurate forecast).

The experimental comparisons of the MLPN, ERNN, RBFN and HFM methods pointed out that no single classification algorithm can be regarded as a panacea. Thus, the use of ensembles of neural networks as an alternative approach is considered. The advantage of the ensemble model is to reduce variance, or instability, of the neural network used for weather forecasting. The error surface of neural network training is full of local minima; trainings with different initial weights are usually trapped in different local minima. These local minima reflect partly the fitting to the regularities of the data and partly to the fitting to the noise in the data. Ensemble-averaging tends to cancel out the noise part as it varies among the ensemble members, and tends to retain the fitting to the regularities of the data.

In this study, WA and WTA ensemble methods are employed. Figure 2 also presents a comparison of the two ensemble methods with actual temperature and

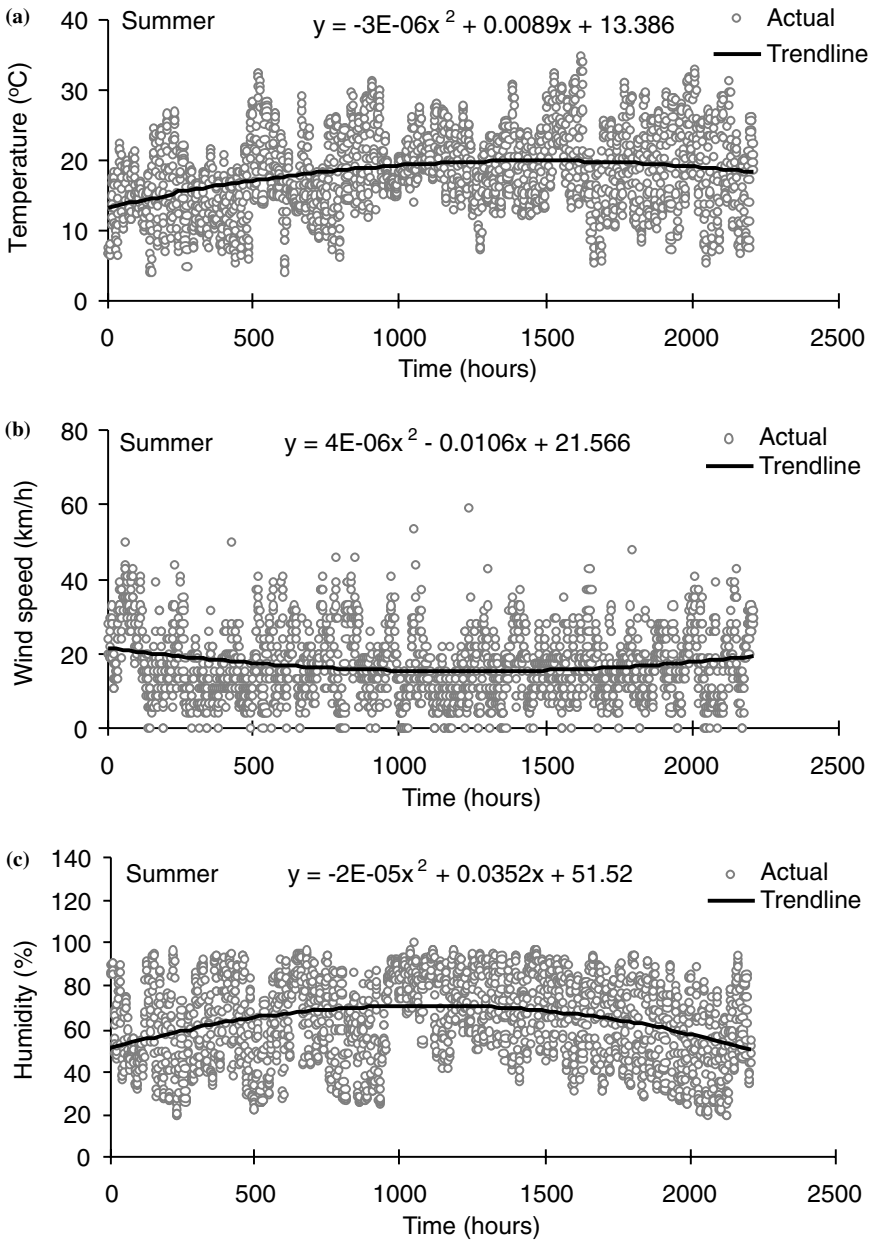

Fig. 5a-c Statistical forecasting trendline and actual **a** temperature, **b** wind speed and **c** humidity during the summer season

wind speed, respectively. It is indicated that the WTA ensemble method predicts the weather parameters with the least error. Table 2 lists the performance of the two ensembles evaluated based, on the MAD, RMSE and correlation coefficient (CC) parameters. Experimental results point out that the use of neural network ensembles can constitute a valid alternative to the development of new neural classifiers more complex than the present one. In particular, it is shown that the combination of neural network results provides classification accuracies better than the ones obtained by single classifiers after long designing phases. In Fig. 4, MAPE values of the dynamic WA and WTA ensemble methods are plotted. The results indicate that the WTA ensemble produced the lowest MAPE. In general, the proposed ensemble neural network models have the capability to be used for predicting weather on both normal days and severe days.

4.2 Direct regression approach

Besides the neural network approaches, a regression technique was also employed for predicting temperature, wind speed and humidity by drawing a polynomial-function trendline. Figure 5 shows comparisons between exponential trendlines and the three weather parameters for the summer season. For the summer season, the parameters can be predicted based on the regression equations expressed as follows:

$$T_s = -0.000003t^2 + 0.0089t + 13.386 \quad (24)$$

$$W_s = 0.000004t^2 - 0.0106t + 21.566 \quad (25)$$

$$H_s = -0.00002t^2 + 0.0352t + 51.52 \quad (26)$$

where s denotes the summer season, T is temperature (°C), W is wind speed (km/h), H is humidity (%) and t is time of forecast (h). Figures of comparisons between exponential trendlines and the three weather parameters for the other three seasons are not presented in order to avoid repetition of similar graphs. The related equations are presented in the following section.

For the fall season, statistical forecast of the three weather parameters can be made with the following equations:

$$T_f = 0.0000003t^2 - 0.0104t + 16.193 \quad (27)$$

$$W_f = -0.000002t^2 + 0.0056t + 16.133 \quad (28)$$

$$H_f = 0.00005t^2 - 0.0013t + 60.084 \quad (29)$$

where f indicates the fall season. For the winter season, the temperature, wind speed and humidity can be forecast as follows:

$$T_w = -0.000003t^2 + 0.0013t - 9.7946 \quad (30)$$

$$W_w = 0.000002t^2 - 0.0032t + 18.476 \quad (31)$$

$$H_w = -0.000008t^2 + 0.0125t + 80.385 \quad (32)$$

where w indicates the winter season. Likewise, the weather parameters for the spring season can be forecast as:

$$T_p = 0.00000002t^2 + 0.00117t - 8.7303 \quad (33)$$

$$W_p = 0.000002t^2 - 0.0007t + 18.629 \quad (34)$$

$$H_p = 0.000005t^2 - 0.0314t + 90.246 \quad (35)$$

where p represents the spring season.

Table 2 presents the CC between the statistically forecast and actual values of the temperature, wind speed and humidity for the four seasons. It is indicated that temperature was forecast with a CC of 0.583, which implies that the statistical model effectively captured the general trends of the actual temperature. In comparison, the wind speed was predicted with a CC of about 0.009, which means that this model was inefficient in providing reliable forecasts of wind speed under the given data set. In general, the statistical models used in this study captured the weather trends adequately, but should be applied with caution for precise forecasting.

Performance of the WA and WTA ensembles were compared with the statistical models (Eqs. 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35) based on MAD, RMSE and CC measure, as shown in Table 2. It is indicated that the WTA method produced the most accurate results compared to those of the WA ensemble and the statistical models. In comparison, the statistical models forecast the temperature, wind speed and humidity with the least accuracy (i.e. low CC, high MAD and high RMSE). This illustrates the fact that the WTA approach is superior to the statistical models as well as individual neural networks for this weather analysis study. In general, the statistical model generated higher errors (i.e. high MAD and high RMSE values) than those of the ensemble methods.

5 Conclusions

Neural-networks-based ensemble models were developed and applied for hourly weather forecasting of southern Saskatchewan. The experimental results show that the ensemble networks can be trained effectively without excessively compromising the performance. The ensembles can achieve good learning performance because one of the ensemble's members is able to learn from the correct learning pattern even though the patterns are statistically mixed with erroneous learning patterns. It is also clear that the proposed neural networks are not only able to learn better but also to generalise better than conventional MLPN, ERNN, HFM and RBFN models. The forecasting reliabilities of these models were evaluated by computing a number of statistical measures. The modelling results indicate that reasonable prediction accuracy was achieved for most of the models. The best predictions were shown by the WTA ensemble, which was based on the four different learning paradigms. The MLPN and ERNN networks did equally well in forecasting temperature, humidity and wind speed. In comparison, the RBFN model performed better than MLPN and ERNN, while the HFM model showed the lowest accuracy. Compared to the regression models, the ensemble networks forecast the weather parameters with higher accuracy. The ensemble neural network can be easily developed to perform multi-class classification problems without increasing the calculation complexity.

Future applications of the proposed ensemble will include on-line training, design of fault-tolerant systems, error detection and correction, probability learning and non-linear equalisation.

Acknowledgements The authors would like to thank the staff of Environment Canada for the provision of the weather information as needed for this study. The authors are grateful for the comments by the anonymous reviewers, which helped to improve the presentation of this paper.

References

1. Bishop CM (1995) Neural networks for pattern recognition. Oxford University Press, Oxford, UK, pp 364-369
2. Breiman L (1999) Combining predictors. In: Sharkey AJC (ed) Combining artificial neural nets: ensemble and modular multi-net systems. Springer, Berlin Heidelberg New York, pp 31-50

3. Carney JG, Cunningham P (1999a) Confidence and prediction intervals for neural network ensembles. In: Proceedings of the international joint conference on neural networks (IJCNN'99), Washington DC, July 1999
4. Carney JG, Cunningham P (1999b) Tuning diversity in bagged neural network ensembles. Technical report TCD-CS-1999-44, Trinity College, Dublin
5. Cho SB, Ahn JH, Lee SI (2001) Exploiting diversity of neural ensembles with speciated evolution. In: Proceedings of the international joint conference on neural networks (IJCNN'01), Washington DC, July 2001, vol 2, pp 808-813
6. Hansen LK, Salamon P (1990) Neural network ensembles. IEEE Trans Pattern Anal 12(10):993-1001
7. Hartono P, Hashimoto S (2001) Learning from imperfect superior using neural network ensemble. IPSJ J 42(5):1214-1222
8. Islam MM, Shahjahan M, Murase K (2001) Exploring constructive algorithms with stopping criteria to produce accurate and diverse individual neural networks in an ensemble. In: Proceedings of the IEEE international conference on systems man and Cybernetics, Tucson, Arizona, October 2001, vol 3, pp 1526-1531
9. Jiang Y, Zhou ZH, Chen ZQ (2002) Rule learning based on neural network ensemble. In: Proceedings of the international joint conference on neural networks (IJCNN'02), Honolulu, Hawaii, May 2002, vol 2, pp 1416-1420
10. Jimenez D, Walsh N (1998) Dynamically weighted ensemble neural networks for classification. In: Proceedings of the international joint conference on neural networks (IJCNN'98), Anchorage, Alaska, May 1998, pp 753-756
11. Khan MR, Ondrusek C (2000) Short-term electric demand prognosis using artificial neural networks. Electr Eng 51:296-300
12. Krogh A, Vedelsby J (1995) Neural network ensembles, cross validation and active learning. In: Tesauro G, Touretzky DS, Keen TK (eds) Neural information processing systems, vol 7. MIT Press, Cambridge, Massachusetts, pp 231-238
13. Kuligowski RJ, Barros AP (1998) Localized precipitation forecasts from a numerical weather prediction model using artificial neural networks. Weather Forecast 13:1194-1205
14. Liu Y, Yao X (1997) Evolving modular neural networks which generalize well. In: Proceedings of the IEEE international conference on evolutionary computation (ICEC'97), Indianapolis, Indiana, April 1997, pp 605-610
15. Liu Y, Yao X, Higuchi T (2000) Evolutionary ensembles with negative correlation learning. IEEE Trans Evolut Comput 4(4):380-387
16. Lorenz EN (1969) Three approaches to atmospheric predictability. Bull Am Meteorol Soc 50:345-349
17. Mao J (1998) A case study on bagging, boosting and basic ensembles of neural networks for OCR. In: Proceedings of the joint conference on neural networks (IJCNN'98), Anchorage, Alaska, May 1998, vol 3, pp 1828-1833
18. Maqsood I, Khan MR, Abraham A (2002a) Intelligent weather monitoring systems using connectionist models. Neural Parallel Sci Comput 10:157-178
19. Maqsood I, Khan MR, Abraham A (2002b) Neurocomputing based Canadian weather analysis. In: Proceedings of the 2nd international workshop on intelligent systems design and applications (ISDA'02), Atlanta, Georgia, August 2002. Dynamic Publishers, Atlanta, Georgia, pp 39-44
20. Moro QI, Alonso L, Vivaracho CE (1994) Application of neural networks to weather forecasting with local data. In: Proceedings of the 12th IASTED international conference on applied informatics, Annecy, France, May 1994, pp 68-70
21. Perrone MP, Cooper LN (1993) When networks disagree: ensemble methods for hybrid neural networks. In: Mammone RJ (ed) Neural networks for speech and image processing. Chapman-Hall, London
22. Rosen BE (1996) Ensemble learning using decorrelated neural networks. Connect Sci 8(3-4):373-384
23. Sharkey AJC (1999) Combining artificial neural nets: ensemble and modular multi-net systems. Springer, Berlin Heidelberg New York
24. Shimshoni Y, Intrator N (1998) Classification of seismic signals by integrating ensembles of neural networks. IEEE Trans Signal Proces 46(5):1194-1201
25. Sollich P, Krogh A (1996) Learning with ensembles: how overfitting can be useful. In: Touretzky DS, Mozer MC, Hasselmo ME (eds) Advances in neural information processing systems 8. MIT Press, Cambridge, Massachusetts, pp 190-196
26. Tumer K, Ghosh J (1996) Error correlation and error reduction in ensemble classifiers. Connect Sci (special issue on combining artificial neural networks: ensemble approaches) 8(3-4):385-404
27. Zhou ZH, Wu J, Tang W (2002) Ensembling neural networks: many could be better than all. Artif Intell 137(1-2):239-263
28. Zurada JM (1992) Introduction to artificial neural systems. West Publishing Company, Saint Paul, Minnesota