ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Echo State Networks and Neural Network Ensembles to predict Sunspots activity

Friedhelm Schwenker[1] and Amr Labib[2]

1- Ulm University - Institute of Neural Information Processing
Ulm - Germany - Email{friedhelm.schwenker@uni-ulm.de}

2- German University in Cairo - Computer Science and Engineering Department
Cairo - Egypt - Email{amr.labib@ieee.org}

Abstract. Echo state networks (ESN) and ensembles of neural networks are developed for the prediction of the monthly sunspots series. Through numerical evaluation on this benchmark data set it has been shown that the feedback ESN models outperform feedforward MLP. Furthermore, it is shown that median fusion lead to robust predictors, and even can improve the prediction accuracy of the best individual predictors.

1 Echo State Networks

The *echo state network* (ESN) is a recurrent neural network model trained using supervised learning [1, 2, 3]. In the following we present a brief introduction to the ESN architecture and ESN learning.

The ESN Network Model

Each neuron , or unit, of the network has an activation state at a given time step n. The network consists of a set of K input units with an activation vector $\mathbf{u}(n)$, a set of N inner units with an activation vector $\mathbf{x}(n)$, and a set of L output units with an activation vector $\mathbf{y}(n)$ [3]. The network has a $N \times K$ input connection weight matrix $\mathbf{W}^{in}$, a $N \times N$ internal connectivity matrix $\mathbf{W}$, a $L \times (K + N)$ output weight matrix $\mathbf{W}^{out}$, and optional $N \times L$ output global feedback connection weight matrix $\mathbf{W}^{back}$.

The activation states of the inner units are updated using:

$$\mathbf{x}(n+1) = f(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{y}(n)) \tag{1}$$

where f is the transfer (activation) function of the inner units. The output is calculated using:

$$\mathbf{y}(n+1) = f^{out}(\mathbf{W}^{out}(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n))) \tag{2}$$

where f^{out} is the output activation function and $\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{y}(n)$ is the concatenation of the input, inner and output activation vectors. The hyperbolic tangent function (tanh) is usually used with f and f^{out}, though other sigmoidal and linear functions can be used as well [3].

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

Training an ESN

Training an ESN is a simple liner regression task [1]. Only $\mathbf{f}^{out}$ is calculated, while $\mathbf{W}^{in}$, $\mathbf{W}$, and $\mathbf{W}^{back}$ never change after initialization. Training is divided into three steps:

1. Network Initialization
   - $\mathbf{W}^{in}$ and $\mathbf{W}^{back}$ are generated randomly.
   - A random sparse matrix $\mathbf{W}$ is generated and scaled to have a spectral radius of α where $\alpha < 1$ to ensure the presence of echo states in the network [3, 2].
2. Sampling Network Training Dynamics
   - Network inner units are initialized arbitrarily (e.g. $\mathbf{x}(0) = 0.$)
   - The inner units' states are updated for $n = 0, 1, 2, ..., T$ using:

     $$\mathbf{x}(n+1) = \mathbf{f}(\mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) + \mathbf{W}^{back}\mathbf{d}(n)) \qquad (3)$$

     where $\mathbf{d}(t)$ is the teaching signal and $\mathbf{d}(0) = 0$.
   - Network states before a washout time T_0 are ignored due to their dependency on the initial state.
   - Network states $(\mathbf{u}(n+1), \mathbf{x}(n+1), \mathbf{d}(n-1))$ after T_0 are collected in a state collecting matrix $\mathbf{M}$ of size $(T - T_0 + 1) \times (K + N + L)$.
   - $f^{out-1}(\mathbf{d}(n))$ values after T_0 are collected in a teacher collecting matrix $\mathbf{T}$ of size $(T - T_0 + 1) \times L$.
3. Computing Output Weights
   Output weights are computed by evaluating the pseudoinverse matrix of $\mathbf{M}$, multiplying it by $\mathbf{T}$, and then transposing it.

   $$\mathbf{W}^{out} = (\mathbf{M}^{+}\mathbf{T})^{t} \qquad (4)$$

2 Neural Network Ensembles

Neural network ensembles are similar in concept to the principle of *divide and conquer* [4]. Complex tasks could be divided among several networks running in parallel to produce the final output.

The input is feed into several networks (members) of different structure, input or initialization, and the output of these networks is feed to the combiner. The combiner then performs a certain function on its input to produce the final output of the system [5]. The combiner could be a linear combiner, performing liner functions (e.g. average, median,...etc), or could be another network or a hierarchy of networks [4].

ESANN'2009 proceedings, European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning. Bruges (Belgium), 22-24 April 2009, d-side publi., ISBN 2-930307-09-9.

3 Echo State Networks in Sunspots Prediction

The sunspots series used in this experiment was the smoothed monthly sunspots numbers obtained from [6]. The data set consisted of 3100 average smoothed sunspots numbers from July 1749 to October 2007.

Network Architecture

The ESN had the following properties:

- A reservoir of size $N = 1000$. Inner weights, $\mathbf{W}$, were randomly generated from a uniform distribution over $(-1, 1)$ with connectivity of 1% and rescaled to a spectral radius of 0.79.
- The ESN used variable input size.
- An output layer of size 1 neuron ($L = 1$) with no output feedback.
- Input weight, $\mathbf{W}^{in}$, was randomly generated from a uniform distribution over $(-1, 1)$.

Network Input and Output

Different input sizes for the network were tested. For an input size K, the data sequence $d(t), d(t-1), ..., d(t-K+1)$ was used for the prediction of $d(t+1)$. The output of the network was the single step ahead prediction of the sunspots series.

Network Equations

The network used the `id` linear function for both inner and output activation functions. The state update equation used was:

$$\mathbf{x}(n+1) = \mathbf{W}^{in}\mathbf{u}(n+1) + \mathbf{W}\mathbf{x}(n) \tag{5}$$

The output equation of the output neuron was:

$$y(n+1) = \mathbf{W}^{out}(\mathbf{u}(n+1), \mathbf{x}(n+1))) \tag{6}$$

Network Training and Testing

The network was trained for 2100 time steps using update equation 5. The first 100 time steps were ignored for the initial washout. The remaining 1000 time steps were used to test the network. The network performance was measured using root mean squared error ($RMSE$).

Different input sizes were tested with the ESN. Table 1 shows the $RMSE$s for running the network through input sizes 15 to 35. Figure 1 shows the original test series and the predicted one for input size $K = 35$.

Input Size	$\mathbf{RMSE_{train}}$	$\mathbf{RMSE_{test}}$	**Input Size**	$\mathbf{RMSE_{train}}$	$\mathbf{RMSE_{test}}$
15	0.68671	0.81934	16	0.68573	0.81816
17	0.68535	0.82179	18	0.6856	0.82104
19	0.68089	0.81779	20	0.68503	0.81954
21	0.68257	0.8169	22	0.68019	0.81973
23	0.68119	0.81587	24	0.67901	0.81892
25	0.6783	0.81918	26	0.67893	0.81814
27	0.67885	0.81931	28	0.67746	0.81855
29	0.67743	0.82068	30	0.68074	0.81911
31	0.67845	0.8194	32	0.68063	0.81785
33	0.67847	0.81964	34	0.6768	0.82088
35	0.67407	0.82528			

Table 1: $RMSE$s for the ESN using input sizes 15 to 35.

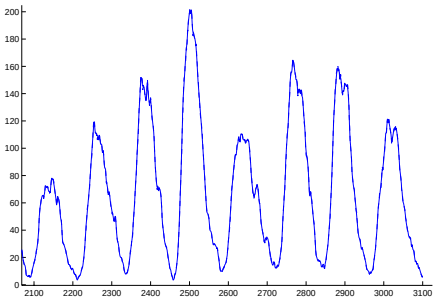

Fig. 1: Test data for the Sunspots prediction using an ESN with $K = 35$. The solid line represents the original signal while the dashed line represents the predicted signal.

4 Neural Network Ensembles in Sunspots Prediction

Two predictor fusion approaches were implemented: Ensembles of MLP members and ensembles of echo state network members. Both approaches were realized by using average and median output combination.

MLP Ensembles

The MLP ensemble created used 21 members with input sizes from 15 to 35. Each member was created and trained as in [7]: the MLPs had only one hidden layer with four hidden neurons and were trained using Levenberg-Marquardt training algorithm. Table 2 shows the result of the individual members and the output of the ensemble for a single run.

Input Size	$\mathbf{RMSE_{train}}$	$\mathbf{RMSE_{test}}$	**Input Size**	$\mathbf{RMSE_{train}}$	$\mathbf{RMSE_{test}}$
15	0.77031	1.1681	16	0.75152	1.4929
17	0.76732	1.3354	18	0.74367	1.2016
19	0.7774	1.3537	20	0.75101	1.1579
21	0.7622	1.9189	22	0.7396	1.3434
23	0.76956	1.8865	24	0.75418	1.3655
25	0.72457	1.3405	26	0.71242	1.7709
27	0.68861	1.0296	28	0.69565	1.6041
29	0.7074	1.7768	30	0.74193	1.309
31	0.75482	1.5581	32	0.70035	1.7079
33	0.69711	1.1165	34	0.7337	1.4343
35	0.70429	1.3281			

Table 2: $RMSE$s for the MLP using input sizes 15 to 35.

Method	$\mathbf{RMSE_{train}}$	$\mathbf{RMSE_{test}}$
Average Combiner (ESN)	0.6780	0.81769
Median Combiner(ESN)	0.6783	0.81769
Average Combiner (MLP)	0.71225	1.4385
Median Combiner(MLP)	0.71320	1.4400

Table 3: $RMSE$s for the ESN and MLP Ensembles.

4.1 ESN Ensembles

The same ESN model used in the previous section was used with the ensembles. The ensemble consisted of 21 members with variable input sizes from 15 to 35. Table 3 shows the result of the ensemble to the output of the individual networks in Table 1. ESN ensembles suffered from a relatively long training.

4.2 Average Ranking

The errors produced by the individual networks and the combiner. For this reason, the average ranking of the ensembles were calculated for both approaches. The ranking was calculated by running the ensemble for 100 runs, calculating the rank according to the $RMSE$s, and calculating the average ranking at the end of the 100 runs. Table 4 shows the top ranks in the ESN ensembles compared to the individual networks for two trials of 100 runs. The same test was performed for the MLP ensembles, but here the rusults are somewhat different: the median-combiner achieved rank 2 and 3, whereas the average combiner failed (rank > 10). Using ESN ensembles, the average and median combiners gave the same results and ranked first in the test case. In both cases, the use of ensembles gave a much stable result than the use of individual networks.

Rank	Trial 1		Trial 2	
	Training	Testing	Training	Testing
1	35	Median	35	Median
2	34	Average	34	Average
3	33	27	33	25
4	31	26	32	23
5	32	24	31	24
6	28	22	30	26
7	30	23	Average	27
8	Average	25	29	22
9	29	29	Median	21
10	Median	20	28	28

Table 4: Top 10 ranks of two trials of calculating the average ranking over 100 runs of the ESN ensembles with multiple input size and the single networks.

5 Conclusion

This paper has presented the use of feedward and feedback neural network ensembles in predicting the sunspots series. ESN ensembles produced the smaller training and testing errors than MLP ensembles. On the other, MLP ensembles were trained much faster than ESN ensembles. This is due to the long time consumed in ESNs to calculate the pseudoinverse matrix when calculating the output weight. ESN ensembles could fit better offline systems that need only to be trained once. In case of online systems or cases in which the system has to be trained several times, MLP ensembles would fit better. Through numerical evaluation on the sunspot data it is shown that ESN outperform feedforward MLP. Furthermore it is shown that median combination is robust combination scheme for ensembles of predictors, and even can improve the prediction accuracy.

References

[1] H. Jaeger and H. Haas. Harnessing nonlinearity: predicting choatic systems and saving energy in wireless telecommunications. *Science*, 304:78–80, April 2004.

[2] H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks. GMD Report 148, German National Research Center for Information Technology, 2001.

[3] H. Jaeger. A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the "echo state network" approach. GMD Report 159, German National Research Center for Information Technology, 2002.

[4] S.Haykin. *Neural Networks: a Comprehensive Foundation*. Prentice Hall, 1999.

[5] S. Yang and A. Browne. Neural network ensembles: combining multiple models for enhanced performance using a multistage approach. *Expert Systems*, 21(5):279–288, 2004.

[6] Sunspot numbers. National Geophysical Data Center (NGDC), 2007.

[7] Tiago A. E. Ferreira, Germano C. Vasconcelos, and Paulo J. L. Adeodato. A new evolutionary method for time series forecasting. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 2221–2222, New York, NY, USA, 2005. ACM.