AN EXPERIMENTAL STUDY: ON REDUCING RBF INPUT DIMENSION BY ICA AND PCA

RONG-BO HUANG [(a)], LAP-TAK LAW [(b)], YIU-MING CHEUNG [(b)]

[(a)] Department of Mathematics, Zhong Shan University, Guangzhou, China.
[(b)] Department of Computer Science, Hong Kong Baptist University, Hong Kong, China.
E-mails: hrongbo@163.net, ltlaw@comp.hkbu.edu.hk, ymc@comp.hkbu.edu.hk

Abstract:

This paper experimentally investigates Independent Component Analysis (ICA) and Principle Component Analysis (PCA) on reducing the input dimension of a Radial Basis Function (RBF) network such that the net's complexity is reduced. The results have shown that a RBF network with ICA as an input pre-process has the similar generalization ability to the one without pre-processing, but the former's performance converges much faster. In contrast, a PCA based RBF however leads to a deteriorated result in both of convergent speed and the generalization ability.

1 Introduction

The Radial Basis Function (RBF) network has been extensively applied to a lot of applications such as pattern recognition, image processing, and time series forecasting because it allows the independent tuning of RBF network parameters and sufficiency of using one layer of neural network to establish input-output mapping [12]. In general, the complexity of the RBF network increases when the net's input size increases. Moreover, the noise and the irrelevant components in the inputs will degrade the generation performance of RBF networks as well.

In the literature, Principle Component Analysis (PCA) is a typical statistical tool for dimension reduction, which just uses second-order statistics information to extract the components [7]. Recently, Independent Component Analysis (ICA) has been extensively studied in neural network and signal processing [1],[4],[7],[10]. It involves higher-order statistics, rather than the second-order one, to extract the independent hidden factors (also called *independent components*) from the observations. Consequently, the information extracted from ICA is much richer than the PCA. Some examples can be found in [2],[3],[11],[15].

In this paper, we propose to use ICA and PCA respectively to pre-process the input data before training the RBF network. That is, the RBF network takes the outputs either from ICA or PCA as its input. For simplicity, we hereafter denote the RBF with ICA pre-processing as ICA-RBF, and the one with PCA as PCA-RBF. Since the ICA-RBF learning is based on fewer independent components, rather than the observations, the complexity of the network can therefore be reduced, but without losing useful information. Consequently, it is generally expected that the performance of ICA-RBF has the similar generalization ability to the conventional RBF, but the former's performance converges much faster. The experimental results have justified this point. In contrast, we have found that a PCA-RBF leads to a deteriorated result in both of convergent speed and the generalization ability because of its limitation on information extraction.

This paper is organized as follows. In Section 2, we introduce the structure of the RBF network used in this paper and its learning algorithm. In Section 3, the ICA algorithm used in this paper is given out. We briefly present the model of input pre-processing of RBF network by ICA and PCA in Section 4. In Section 5, the experiment results are provided. Finally, we draw a conclusion in Section 6.

2 The RBF algorithm

The RBF is a feed-forward neural network to estimate the underlying function from the given input-output pairs. In the literature, there are several common RBF variants. A typical one is extended normalized RBF (ENRBF) network [13], whose architecture is shown in Figure 1. The network consists of *d*-unit input layer, *k*-unit hidden layer, and *n*-unit output layer. The network actual output vector $\hat{\boldsymbol{z}}_t = [\hat{z}_t^{(1)}, \hat{z}_t^{(2)}, ..., \hat{z}_t^{(n)}]^T$ is

$$\hat{\boldsymbol{z}}_{\boldsymbol{t}} = \sum_{j=1}^{k} (\boldsymbol{W}_{\boldsymbol{j}}{}^{\boldsymbol{T}} \boldsymbol{x}_{\boldsymbol{t}} + \boldsymbol{c}_{\boldsymbol{j}}) O_j(\boldsymbol{x}_{\boldsymbol{t}}), \qquad (1)$$

where $\boldsymbol{c}_{\boldsymbol{j}}$ is an $n\times 1$ constant vector, $\boldsymbol{x}_t = [x_t^{(1)}, x_t^{(2)}, ..., x_t^{(d)}]^T$ is an input vector and $\boldsymbol{W}_j$ is the weight matrix between hidden layer and the output layer. $O_j(\boldsymbol{x}_t)$ is the output of unit *j* in the hidden layer with

$$O_j(\boldsymbol{x}_t) = \frac{\phi[(\boldsymbol{x}_t - \boldsymbol{m}_j)^T \Sigma_j^{-1} (\boldsymbol{x}_t - \boldsymbol{m}_j)]}{\sum_{i=1}^{k} \phi[(\boldsymbol{x}_t - \boldsymbol{m}_i)^T \Sigma_j^{-1} (\boldsymbol{x}_t - \boldsymbol{m}_i)]}, \qquad (2)$$

where $\boldsymbol{m}_j$ is the center vector, and Σ_j is the receptive field of the basis function $\phi(.)$. In general, we adopted $\phi(s^2) = \exp(-0.5s^2)$. Thus, Eq.(2) becomes:

$$O_j(\boldsymbol{x}_t) = \frac{\exp[-0.5(\boldsymbol{x}_t - \boldsymbol{m}_j)^T \Sigma_j^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_j)]}{\sum_{i=1}^{k} \exp[-0.5(\boldsymbol{x}_t - \boldsymbol{m}_i)^T \Sigma_j^{-1}(\boldsymbol{x}_t - \boldsymbol{m}_i)]}, (3)$$

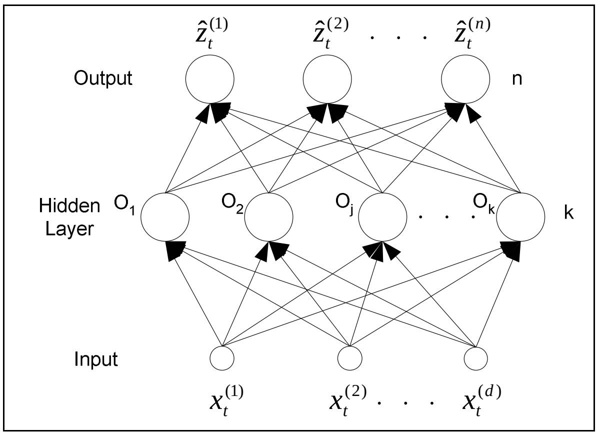


Figure 1: General structure of RBF network.

The ENRBF learns the parameters in the hidden and output layers by the following two separate steps:

Step 1. Learn $\{\boldsymbol{m}_j, \boldsymbol{\Sigma}_j\}$'s in the hidden layer via a clustering algorithm such as *k*-means [14] or RPCCL [8].

Step 2. Learn $\{\boldsymbol{W}_j, \boldsymbol{c}_j\}$'s in the output layer by minimizing

$$Q = \frac{1}{N}\sum_{t=1}^{N}(z_t - \hat{z}_t)^T(z_t - \hat{z}_t)$$

where $\boldsymbol{z}_t$ is the desired output at time step *t*, and *N* is the sample size of training data.

3 The ICA algorithm

The classical model of ICA is as follows: Suppose there are *m* independently and identically distributed sources, which are statistically independent to each other. The sources are sampled at discrete time *t*, denoted as $\boldsymbol{y}_t = [y_t^{(1)}, y_t^{(2)}, ..., y_t^{(m)}]^T$ and the observed signals are $\boldsymbol{x}_t$. The observed model can be written in a matrix form:

$$\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{y}_t, \quad 1 \le t \le N, \qquad (5)$$

where $\boldsymbol{A}$ is an unknown $d \times m$ matrix and $d \ge m$.

The ICA is to estimate the matrix $\boldsymbol{A}$ and then compute the inverse $\boldsymbol{W}$ such that:

$$\hat{\boldsymbol{y}}_t = \boldsymbol{W}\boldsymbol{x}_t, \quad 1 \le t \le N, \qquad (6)$$

where $\hat{\boldsymbol{y}}_t = [\hat{y}_t^{(1)}, \hat{y}_t^{(2)}, ..., \hat{y}_t^{(m)}]^T$ is an estimate of the sources $\boldsymbol{y}_t$. In the literature, typical information-theoretic approaches using fixed nonlinearity function such as information maximization (INFOMAX) [4] and minimum mutual information (MMI) [1] are used to perform ICA. However, they can separate either sub-Gaussian or super-Gaussian source signals but not both. In this paper, we adopt the Adaptive Polynomial Power Learning Estimation based ICA Algorithm (APPLE-ICA) proposed in the paper [6] that use a single polynomial term with the exponent learned together with the de-mixing matrix $\boldsymbol{W}$. The APPLE-ICA algorithm can successfully separate any combination of sub-Gaussian and super-Gaussian sources by maximizing the cost function:

$$\begin{aligned} L(\boldsymbol{W}) &= Q(\boldsymbol{W}, \boldsymbol{P}) \\ &= \ln|\det(\boldsymbol{W})| + \sum_{j=1}^{m} \ln g_j(\hat{y}_t^{(j)}) \\ &= \ln|\det(\boldsymbol{W})| - \sum_{j=1}^{m} \frac{1}{p_j + 1}\left|\hat{y}_t^{(j)}\right|^{p_j+1} + C, \quad (7) \end{aligned}$$

where *C* is a constant term and $\boldsymbol{P} = \{p_1, p_2, ..., p_m\}$. Hence, we can adaptively learn the parameter $\boldsymbol{P}$ as well as $\boldsymbol{W}$ by maximizing $Q(\boldsymbol{W}, \boldsymbol{P})$. The detailed algorithm is as follows:

Step 1. Initialize $\boldsymbol{W}$ and a parameter

$$\boldsymbol{U} = [u_1, u_2, ..., u_m]^T$$

Step 2. Given an observed signal $\boldsymbol{x}_t$, let:

$$\hat{\boldsymbol{y}}_t = \boldsymbol{W}\boldsymbol{x}_t$$

$$p_j = \lambda e^{u_j}, \quad 1 \le j \le m$$

$$\phi_j(\hat{y}_t^{(j)}) = -sign(\hat{y}_t^{(j)})\left|\hat{y}_t^{(j)}\right|^{p_j} \qquad (8)$$

where λ is a positive constant.

Step 3. Update $\boldsymbol{W}$ and $\boldsymbol{U}$ by:

$$\begin{aligned} \boldsymbol{W}^{new} &= \boldsymbol{W}^{old} + \eta[\boldsymbol{I} + \phi(\hat{\boldsymbol{y}}_t)\hat{\boldsymbol{y}}_t^T]\boldsymbol{W} \\ u_j^{new} &= u_j^{old} + \eta\frac{\partial Q(\boldsymbol{W}, \boldsymbol{P})}{\partial u_j} \qquad (9) \end{aligned}$$

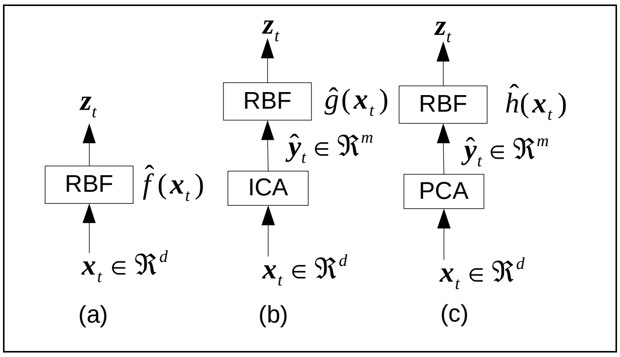
with

$$\frac{\partial Q(\boldsymbol{W}, \boldsymbol{P})}{\partial u_j} = \frac{p_j\left|\hat{y}_t^{(j)}\right|^{p_j+1}}{p_j + 1}\left(\frac{1}{p_j + 1} - \ln\left|\hat{y}_t^{(j)}\right|\right), \quad (10)$$

The iterations of Step 2 and Step 3 are not stopped until both of $\boldsymbol{W}$ and $\boldsymbol{P}$ converge. For more details of the APPLE-ICA algorithm, interested readers can refer to the paper [6].

4 Input Pre-processing of RBF network by ICA

The original RBF network model is shown in Figure 2(a), where at each time *t* the network accepts the input $\boldsymbol{x}_t$ directly and produces the output $\boldsymbol{z}_t$. During RBF network training, the network uses the input-output pairs $\{(\boldsymbol{x}_t, \boldsymbol{z}_t)\}_{t=1}^{N}$ to estimate the parameters in hidden output layers.



$\boldsymbol{z}_t$
RBF
$\hat{f}(\boldsymbol{x}_t)$
$\boldsymbol{x}_t \in \Re^d$
(a)
$\boldsymbol{z}_t$
RBF
$\hat{g}(\boldsymbol{x}_t)$
$\hat{\boldsymbol{y}}_t \in \Re^m$
ICA
$\boldsymbol{x}_t \in \Re^d$
(b)
$\boldsymbol{z}_t$
RBF
$\hat{h}(\boldsymbol{x}_t)$
$\hat{\boldsymbol{y}}_t \in \Re^m$
PCA
$\boldsymbol{x}_t \in \Re^d$
(c)

Figure 2: (a) the original RBF network model. (b) the proposed ICA-RBF network model. (c) the PCA-RBF network model.

To further simplify the complexity of hidden units in RBF network, our proposed ICA-RBF model is as shown in Figure 2(b). The ICA-RBF network consists of two phases. The first phase is the ICA pre-process that extracts *m* mutually-independent dominant components $y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(m)}$ from the input $\boldsymbol{x}_t$. Note that *m* may equal to or smaller than the input dimension *d*. Then, in the second phase, the output $\hat{\boldsymbol{y}}_t = [\hat{y}_t^{(1)}, \hat{y}_t^{(2)}, ..., \hat{y}_t^{(m)}]$ is feed into the RBF network as the input of RBF network. For PCA-RBF network, its procedure shown in Figure 2(c) is the same as the PCA-ICA.

5 Experimental Results

5.1 Experiment 1

To investigate the performance of the proposed ICA-RBF network and PCA-RBF network in comparison with the ENRBF network in function approximation, we used a set of three independent uniform series $\{y_t^{(1)}\}_{t=1}^{N}, \{y_t^{(2)}\}_{t=1}^{N}, \{y_t^{(3)}\}_{t=1}^{N}$ with size N = 5100, whereby the observations were obtained by Eq. (5) with the mixing matrix:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0.4 & 0.6 \\ 0.8 & 1 & 0.8 \\ 1.4 & 0.5 & 0.6 \\ 0.8 & 0.4 & 1.2 \\ 0.8 & 0.9 & 0.7 \end{bmatrix}$$

We let the first 5,000 data be training set, and the remaining 100 data be the testing set. Moreover, we let the learning rate of ICA be η = 0.0001 in the experiment. Meanwhile, we set the learning rate of RBF network at 0.0001 as well, and let the desired net's outputs be from the nonlinear function:

$$z_t = y_t^{(1)} \sin(y_t^{(2)}) + 0.4(y_t^{(3)})^2 + \varepsilon_t$$

where ε_t is zero-mean Gaussian white noise with the variance being 0.001. We measured the network performance under the mean-square-error (MSE) criterion.

The sources data are shown in Figure 3 and the mixtures are shown in Figure 4. In Figure 5, it can be seen that the ICA pre-process successfully separates the mixtures to recover the sources. Similarly, we use the PCA pre-process to extract three principle components from the mixtures. Then we feed the data processed by ICA and PCA into the RBF network respectively. The experimental results are shown in Figure. 6. It has shown that the ICA-RBF network performance significantly outperforms the PCA-RBF network and conventional RBF network in Figure 6.

5.2 Experiment 2

We also performed an experiment on the benchmark data getting from the famous Rob Hyndman's Time Series Data Library. We use the FOREX daily foreign exchange rates of 7 countries from 31st December 1979 to 31st December 1998 with size 4,774 data in this experiment. We let the first 4,724 data be the training set, and the remaining 50 data be the testing set. In the experiment, we extracted the exchange rates of the first six countries as the input data set $\boldsymbol{x}_t = [x_t^{(1)}, x_t^{(2)}, ..., x_t^{(6)}]^T$ and the exchange rate of the last country as the output z_t. The experimental results are shown in Figure 7. We observe that ICA-RBF network again outperforms the PCA-RBF network and conventional RBF network in the testing data set.

6 Conclusion

We have investigated ICA and PCA to reduce the input size of a RBF network. The experimental results have shown that the ICA-RBF network not only converges faster than the conventional RBF, but also outperforms the PCA-RBF in both of performance convergent speed and the generalization ability. It can be therefore seen that ICA has provided a good alternative for dimension reduction.

7 Acknowledgement

The work described in this paper was fully supported by a Faculty Research Grant of Hong Kong Baptist University with Project Number: FRG/02-03/I-06.

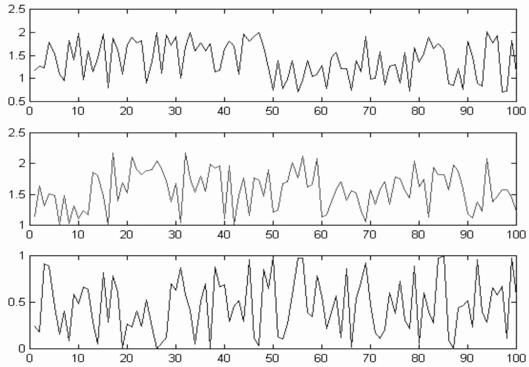


Figure 3: The slide windows of the first 100 data from the three sources.

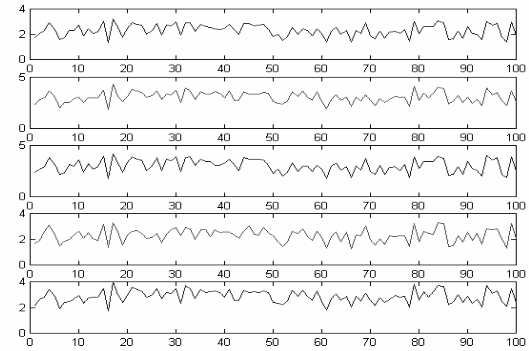


Figure 4: The five observed mixtures of the sources in Figure 3.

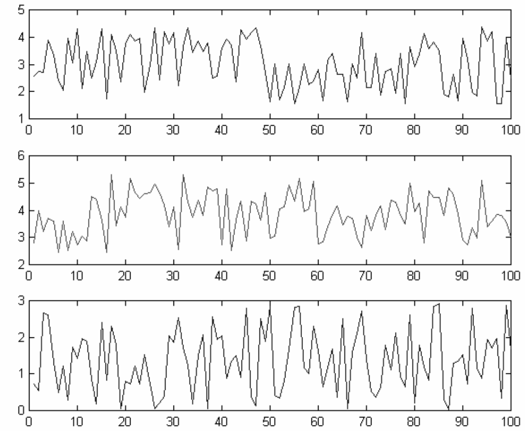


Figure 5: The recovered sources from the five mixtures in Figure 3. The scale of the recovered sources are shifted and enlarged.

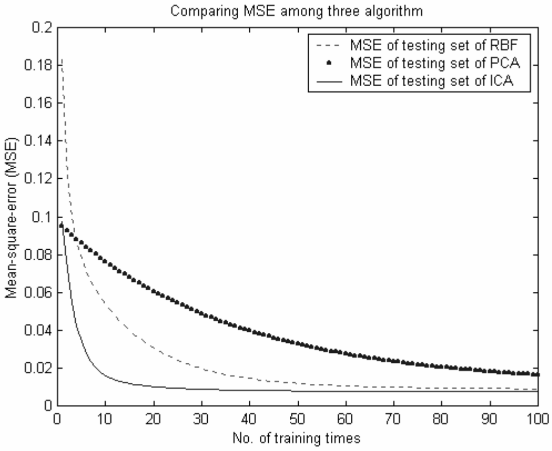


Figure 6: The performance of the PCA-RBF, RBF and ICA-RBF network on testing data set of the uniform sources.

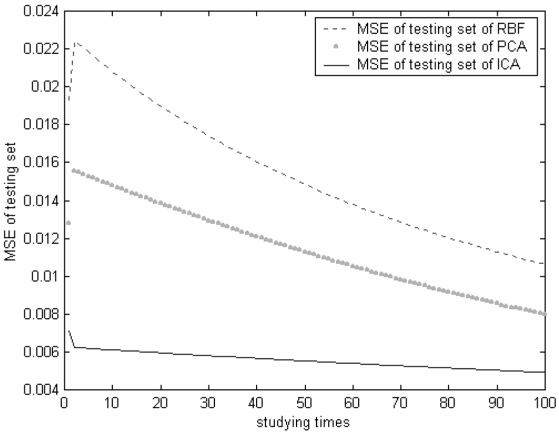


Figure 7: The performance of the PCA-RBF, RBF and ICA-RBF network on testing data set of benchmark data.

References

[1] S.-I. Amari, A. Cichocki, H. Yang, "A new learning algorithm for blind source separation", *Advances in Neural Information Processing Systems*, Vol. 8, MIT Press: Cambridge, MA, 1996.

[2] A.D. Back and A.S. Weigend, "A First Application of Independent Component Analysis to Extracting Structure from Stock Returns", *International Journal of Neural System* 8(4), pp. 473-484, 1997.

[3] M.S. Bartlett, H.M. Lades and T.J. Sejnowski, "Independent Component Representations for Face Recognition", *Proceedings of the SPIE Symposium on Electronic Imaging: Science and Technology; Conference on Human Vision and Electronic Imaging III*, pp. 528-539, 1998.

[4] A.J. Bell, T.J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution", *Neurocomput*ing, Vol. 7 pp. 1129-1159, 1995.

[5] Y.M. Cheung, "Dual Auto-Regressive Modeling Approach to Gaussian Process Identification",

Proceedings of 2001 IEEE International Conference on Multimedia and Expo (ICME2001), pp. 1256-1259, Tokyo, Japan, 2001.

[6] Y.M. Cheung and L. Xu, "A New Information-Theoretic Based ICA Algorithm for Blind Signal Separation", *International Journal of Computers and Applications*, in press.

[7] Y.M. Cheung and L. Xu, "Independent Component Ordering in ICA Time Series Analysis", *Neurocomputing,* Vol. 41, pp. 145-152, 2001.

[8] Y.M. Cheung, "Rival Penalization Controlled Competitive Learning for Data Clustering with Unknown Cluster Number", to appear in *Proceedings of 9th International Conference on Neural Information Processing, Singapore,* November 18-22, 2002.

[9] A. Hyvärinen, "Independent component analysis in the present of Gaussian noise by maximizing joint likelihood", *Neurocomputing,* Vol. 22, pp. 49-67, 1998.

[10] A. Hyvärinen, E. Oja, "Independent component analysis: algorithms and applications", *Neural Networks,* 13(4-5), pp. 411-430, 2000.

[11] G.J. Jang, T.W. Lee and Y.H. Oh, "Learning Statistically Efficient Features for Speaker Recognition", *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing,* Salt Lake City, Utah, May, 2001.

[12] A.P. Topchy, O.A. Lebedko, V.V. Miagkikh, N.K. Kasabov, "Adaptive Training of Radial Basis Function Networks Based on Cooperative Evolution and Evolutionary Programming", *Progress in Connectionist- Based Information Systems: Proceedings of the 1997 International Conference on Neural Information Processing & Intelligent Information Systems*, pp. 253-258. Springer-Verlag, Singapore.

[13] L. Xu, "RBF Nets, Mixture Experts, and Bayesian Ying-Yang Learning", *Neuro- computing, Vol. 19, No.1-3,* pp. 223-257, 1998.

[14] L. Xu, "Rival Penalized Competitive Learning, Finite Mixture, and Multisets Clustering", *Proc. Intentional Joint Conference on Neural Networks*, May 5-9, 1998, Anchorage, Alaska, Vol.II, pp. 2525-2530, 1998.

[15] A. Ziehe, G.Nolte, T. Sander, K.R. Muller and G. Curio, "A Comparison of ICA-based Artifact Reduction Methods for MEG", *12th International Conference on Biomagnetism, Helsinki University of Technology,* Finland, 2000.