

Efficient Training of RBF Networks Via the BYY Automated Model Selection Learning Algorithms

Jiwei Ma, Jianping Xu, and Junhua Zhao*

School of Mathematics, Institute of Mathematics, Beijing Normal University
100875, Beijing, P. R. China
jwma@math.pku.edu.cn

Abstract.

In this paper, a novel class of efficient BYY algorithms for training RBF networks is proposed. It is shown that the proposed algorithms can be viewed as a special case of the BYY algorithms for training RBF networks. In addition, it is proved that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks. The proposed algorithms are applied to the training of RBF networks for function approximation. The results show that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks. The proposed algorithms are applied to the training of RBF networks for pattern recognition. The results show that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks.

1 Introduction

The efficient training of RBF networks is one of the most important problems in the field of neural networks. In this paper, a novel class of efficient BYY algorithms for training RBF networks is proposed. It is shown that the proposed algorithms can be viewed as a special case of the BYY algorithms for training RBF networks. In addition, it is proved that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks. The proposed algorithms are applied to the training of RBF networks for function approximation. The results show that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks. The proposed algorithms are applied to the training of RBF networks for pattern recognition. The results show that the proposed algorithms are more efficient than the BYY algorithms for training RBF networks.

*This work is partially supported by the

$p, y, x' \in \mathcal{P}, x' \in \mathcal{Q}, x, y' \in \mathcal{Q}, y' \in \mathcal{P}$ and \mathcal{P}, \mathcal{Q} are probability distributions over \mathcal{X} and \mathcal{Y} respectively.

$$H(p, q) = \int p(x, y) \log \frac{p(x, y)}{q(x, y)} dx dy = \mathbb{E}_{p, q} [\log \frac{p}{q}]$$

where $\mathbb{E}_{p, q}$ is the expectation over p, q .

Let $\theta = (p, y, x')$ and $\phi = (q, x, y')$ be two distributions over $\mathcal{P} \times \mathcal{Y} \times \mathcal{X}$ and $\mathcal{Q} \times \mathcal{X} \times \mathcal{Y}$ respectively. Let $\theta_j = (p_j, y_j, x'_j)$ and $\phi_j = (q_j, x_j, y'_j)$ be two distributions over $\mathcal{P} \times \mathcal{Y} \times \mathcal{X}$ and $\mathcal{Q} \times \mathcal{X} \times \mathcal{Y}$ respectively. Let $\alpha_j \geq 0$ and $\sum_{j=1}^K \alpha_j = 1$.

Let p, x' be two distributions over $\mathcal{P} \times \mathcal{X}$ and $p, x' = \frac{1}{N} \sum_{t=1}^N \delta_{x_t}$ be two distributions over $\mathcal{P} \times \mathcal{X}$ and $p, x' = (p, y, x')$ be two distributions over $\mathcal{P} \times \mathcal{Y} \times \mathcal{X}$.

$$p, y = j, x' = p, j, x' = \frac{\alpha_j q, x, \theta_j'}{q, x, \Theta_K'}$$

where $\mathbf{1}_N = \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1} \otimes \mathbf{1}$.

Let us define $U_j, x_t = \alpha_j q_t x_t m_j \Sigma_j^{-1}$, $j = 1, \dots, K$. $J, \Theta_{K'}$ can be written as follows

$$J, \Theta_{K'} = \frac{1}{N} \sum_{t=1}^N J_{t'} \Theta_{K'} \otimes J_{t'} \Theta_{K'} = \sum_{j=1}^K \frac{U_j, x_{t'}}{\sum_{i=1}^K U_i, x_{t'}} \otimes U_j, x_{t'} \quad (A.1)$$

Using eq. (A.1) we can write the gradient of $J, \Theta_{K'}$ w.r.t. $\beta_j \otimes m_j \otimes B_j$ as follows

$$\frac{\partial J_{t'} \Theta_{K'}}{\partial \beta_j} = \frac{1}{q_t x_t \Theta_{K'}} \sum_{i=1}^K \lambda_{i, t'} \delta_{ij} - \alpha_j U_i, x_{t'} \quad (A.2)$$

$$\frac{\partial J_{t'} \Theta_{K'}}{\partial m_j} = p_{t, j, x_{t'}} \lambda_{j, t'} \Sigma_j^{-1} x_t - m_j \quad (A.3)$$

$$\text{vec} \frac{\partial J_{t'} \Theta_{K'}}{\partial B_j} = \frac{\partial B_j B_j^T}{\partial B_j} \text{vec} \frac{\partial J_{t'} \Theta_{K'}}{\partial \Sigma_j} \quad (A.4)$$

where δ_{ij} is the Kronecker delta and $\text{vec} A$ is the vectorization of matrix A .

$$\lambda_{i, t'} = - \sum_{l=1}^K p_{l, x_{t'}} \delta_{il} \left(\alpha_l q_t x_t m_l \Sigma_l^{-1} \right) \quad (A.5)$$

$$\frac{\partial J_{t'} \Theta_{K'}}{\partial \Sigma_j} = \frac{1}{\Sigma_j} \left(p_{t, j, x_{t'}} \lambda_{j, t'} \Sigma_j^{-1} x_t - m_j \right)^T \Sigma_j^{-1} - \Sigma_j^{-1} \quad (A.6)$$

$$\frac{\partial B B^T}{\partial B} = I_{d \times d} \otimes B^T + E_{d \times d} \bullet B^T \otimes I_{d \times d}$$

where \otimes is the Kronecker product and \bullet is the Hadamard product

$$E_{d \times d} = \frac{\partial B^T}{\partial B} = \Gamma_{ij'}{}_{d \times d} = \begin{pmatrix} \Gamma_{..} & \dots & \Gamma_{.d} \\ \vdots & \ddots & \vdots \\ \Gamma_{d.} & \dots & \Gamma_{dd} \end{pmatrix}_{d \times d}$$

where Γ_{ij} is the (i, j) element of $\Gamma_{ij'}{}_{d \times d}$, j, i 'th element of $\Gamma_{ij'}{}_{d \times d}$ is $\frac{\partial B B^T}{\partial B}$.

$$\text{vec} \frac{\partial J, \Theta_{k'}}{\partial B_j} = p \cdot j \cdot x_t' \lambda_j \cdot t \cdot I_{d \times d} \otimes B_{d \times d}^T + E_{d, d} \cdot B_{d \times d}^T \otimes I_{d \times d}'$$

$$\times \text{vec} \Sigma_j^-, x_t - m_j', x_t - m_j'{}^T \Sigma_j^- - \Sigma_j^-$$

$$\Delta \beta_j = \frac{\eta}{q \cdot x_t' \cdot \Theta_{k'}} \sum_{i=1}^K \lambda_i \cdot t \cdot \delta_{ij} - \alpha_j' U_i \cdot x_t' \cdot \Theta_{k'}$$

$$\Delta m_j = \eta p \cdot j \cdot x_t' \lambda_j \cdot t \Sigma_j^- \cdot x_t - m_j' \cdot \Theta_{k'}$$

$$\Delta \text{vec} B_j = \frac{\eta}{q} p \cdot j \cdot x_t' \lambda_j \cdot t \cdot I_{d \times d} \otimes B_{d \times d}^T + E_{d, d} \cdot B_{d \times d}^T \otimes I_{d \times d}'$$

$$\times \text{vec} \Sigma_j^-, x_t - m_j', x_t - m_j'{}^T \Sigma_j^- - \Sigma_j^-$$

$$\lim_{n \rightarrow \infty} \eta \cdot n = \infty \quad \sum_{n=1}^{\infty} \eta \cdot n = \infty$$

3 Training of the RBF Network

$$y_j \cdot x_t = \sum_{j=1}^n w_{ij} \phi_j \cdot x_t'$$

$$W_{ij} = \frac{1}{\sigma_j} \phi_j(x_i)$$

where n is the number of clusters, $\phi_j(x)$ is the Gaussian density function of the j^{th} cluster, μ_j is the mean and σ_j is the standard deviation of the j^{th} cluster. The Gaussian density function is given by:

$$\phi_j(x) = \frac{1}{\sigma_j} \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right) \quad (1)$$

where μ_j is the mean and σ_j is the standard deviation of the j^{th} cluster. The Gaussian density function is given by:

$$y(x) = \sum_{j=1}^n \lambda_j \phi_j(x) = \sum_{j=1}^n \lambda_j \frac{1}{\sigma_j} \exp\left(-\frac{(x - \mu_j)^2}{2\sigma_j^2}\right) \quad (2)$$

where λ_j is the weight of the j^{th} cluster. The weights are constrained to be non-negative and sum to one, i.e. $\lambda_j \geq 0$ and $\sum_{j=1}^n \lambda_j = 1$. The data points are denoted by $D_{x,y} = \{(x_i, y_i) \mid i = 1, 2, \dots, N\}$.

$$\begin{aligned} E &= \sum_{i=1}^N (y_i - f(x_i))^2 = \sum_{i=1}^N (y_i - \sum_{j=1}^n \lambda_j \phi_j(x_i))^2 \\ &= \sum_{i=1}^N (y_i - \sum_{j=1}^n \lambda_j \frac{1}{\sigma_j} \exp\left(-\frac{(x_i - \mu_j)^2}{2\sigma_j^2}\right))^2 \end{aligned} \quad (3)$$

The parameters λ_j , μ_j , and σ_j are estimated by minimizing the error function E with respect to these parameters.

$$\begin{cases} \Delta \lambda_j = \eta_\lambda \sum_{i=1}^N (y_i - \sum_{l=1}^n \lambda_l \phi_l(x_i)) \phi_j(x_i) \\ \Delta \mu_j = \eta_\mu \sum_{i=1}^N (y_i - \sum_{l=1}^n \lambda_l \phi_l(x_i)) \phi_j(x_i) (x_i - \mu_j) \lambda_j \sigma_j \\ \Delta \sigma_j = \eta_\sigma \sum_{i=1}^N (y_i - \sum_{l=1}^n \lambda_l \phi_l(x_i)) \phi_j(x_i) (x_i - \mu_j)^T (x_i - \mu_j) \lambda_j \sigma_j^3 \end{cases} \quad (4)$$

where η_λ , η_μ , and η_σ are the learning rates for the parameters λ_j , μ_j , and σ_j respectively. The parameters are updated iteratively until convergence.

Let's consider the problem of finding the optimal solution for the following problem:

Let's consider the problem of finding the optimal solution for the following problem: n is the number of samples, K is the number of classes. Let's denote the matrix of samples as $Y \in \mathbb{R}^{n \times K}$. Let's denote the matrix of class means as $T \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class variances as $\Sigma \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class covariances as $\Sigma_j \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class means as $m_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class variances as $\sigma_j \in \mathbb{R}^{K \times 1}$.

$$\sigma_j = \frac{1}{N_j} \sum_{x_t \in C_j} \|x_t - m_j\|^2$$

Let's denote the matrix of class means as $m_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class variances as $\sigma_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class covariances as $\Sigma_j \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class means as $m_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class variances as $\sigma_j \in \mathbb{R}^{K \times 1}$.

4 Experiment Results

Let's consider the problem of finding the optimal solution for the following problem: n is the number of samples, K is the number of classes. Let's denote the matrix of samples as $Y \in \mathbb{R}^{n \times K}$. Let's denote the matrix of class means as $T \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class variances as $\Sigma \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class covariances as $\Sigma_j \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class means as $m_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class variances as $\sigma_j \in \mathbb{R}^{K \times 1}$.

4.1 On the Noisy XOR Problem

Let's consider the problem of finding the optimal solution for the following problem: n is the number of samples, K is the number of classes. Let's denote the matrix of samples as $Y \in \mathbb{R}^{n \times K}$. Let's denote the matrix of class means as $T \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class variances as $\Sigma \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class covariances as $\Sigma_j \in \mathbb{R}^{K \times K}$. Let's denote the matrix of class means as $m_j \in \mathbb{R}^{K \times 1}$. Let's denote the matrix of class variances as $\sigma_j \in \mathbb{R}^{K \times 1}$.

Figure 1: Scatter plot of two time series, \$X_1\$ (black dots) and \$X_2\$ (blue dots), showing a strong negative correlation. The plot is divided into two horizontal regions: a grey region for \$y > 0\$ and a cyan region for \$y < 0\$.

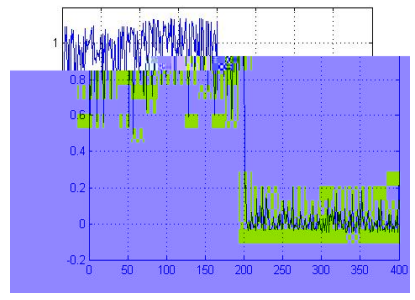
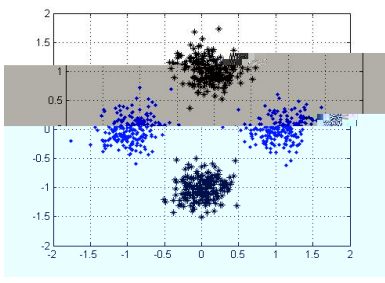


Fig. 1. Time series prediction results for \$X_1\$ (●) and \$X_2\$ (●).

Fig. 2. Time series prediction results for \$X_1\$ (●) and \$X_2\$ (●).

The prediction results for the Mackey-Glass time series are shown in Figure 2. The prediction interval (green shaded area) is wider than in Figure 1, reflecting the higher volatility of the Mackey-Glass time series. The prediction results for \$X_1\$ (black dots) and \$X_2\$ (blue dots) are shown in Figure 2.

4.2 On the Mackey-Glass Time Series Prediction

The Mackey-Glass time series is defined by the following equation:

$$x(t+1) = \frac{ax(t-\tau)}{1+bx(t-\tau)}, \quad (11)$$

where \$a = 0.3\$, \$b = 0.1\$, and \$\tau = 17\$. The prediction results for the Mackey-Glass time series are shown in Figure 2. The prediction interval (green shaded area) is wider than in Figure 1, reflecting the higher volatility of the Mackey-Glass time series. The prediction results for \$X_1\$ (black dots) and \$X_2\$ (blue dots) are shown in Figure 2.

The prediction results for the Mackey-Glass time series are shown in Figure 2. The prediction interval (green shaded area) is wider than in Figure 1, reflecting the higher volatility of the Mackey-Glass time series. The prediction results for \$X_1\$ (black dots) and \$X_2\$ (blue dots) are shown in Figure 2.

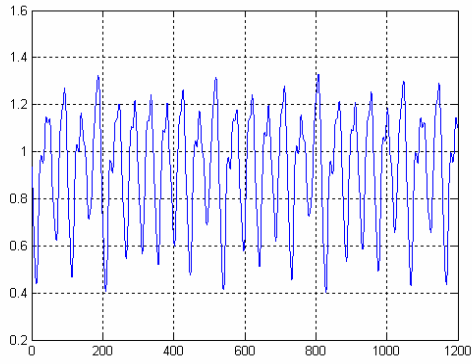


Fig. 3. Time evolution of the output signal $y(t)$ for $\alpha = 0.01$

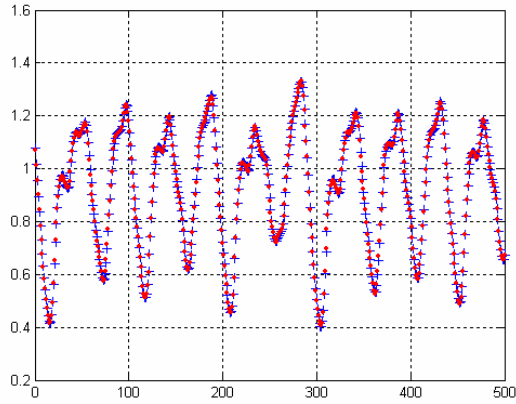


Fig. 4. Time evolution of the output signal $y(t)$ for $\alpha = 0.01$ and $\beta = 0.01$ (the signal is highly oscillatory and noisy)

5 Conclusions

In this paper, we have presented a new method for the identification of the parameters of a nonlinear system. The proposed method is based on the use of the YY algorithm.

