On the Regularization of Forgetting Recursive Least Square

Chi Sing Leung, Gilbert H. Young, John Sum, and Wing-kay Kan

***Abstract*— In this paper, the regularization of employing the forgetting recursive least square (FRLS) training technique on feedforward neural networks is studied. We derive our result from the corresponding equations for the expected prediction error and the expected training error. By comparing these error equations with other equations obtained previously from the weight decay method, we have found that the FRLS technique has an effect which is identical to that of using the simple weight decay method. This new finding suggests that the FRLS technique is another on-line approach for the realization of the weight decay effect. Besides, we have shown that, under certain conditions, both the model complexity and the expected prediction error of the model being trained by the FRLS technique are better than the one trained by the standard RLS method.**

***Index Terms*— Feedforward neural network, forgetting recursive least square, model complexity, prediction error, regularization, weight decay.**

I. Introduction

THE FORGETTING recursive least square (FRLS) technique [4], [8], [13], [15], [16], [25] is a fast parameter estimation method with adaptive ability. Hence, it has recently been applied widely in the training of feedforward neural networks. As in many applications, such as system identification and time series prediction, a batch of training data set usually cannot be obtained in advance. Therefore, conventional batch mode training techniques such as the backpropagation, the Newton method, and other nonlinear programming techniques, would not be easily applied. Thus, the FRLS method or other adaptive training methods becomes inevitable. As the increasing popularity of using the FRLS method in neural-network learning [4], [8], [13], [15]–[17], [25] and pruning [15], [16], [18], it is interesting to investigate more on other properties besides its adaptive behavior. In this paper, one property we are concentrated on is the FRLS's regularization behavior.

Recently, there are many articles which are focused on the design of a regularizer [28], the use of regularization [11], [19] and the effect of regularization in model complexity [21]–[23]. In general, regularization is a method which aims at reducing the model complexity [11], [14], [19]–[23]. In conventional batch mode training approach, regularization is usually realized by adding an extra term or a penalty term to the training error function. Three commonly used methods are the weight decay term [20], Tikhonov regularizer [3], [11], and smooth regularizer [28].

Using the FRLS method, training error function can also be interpreted as a kind of weighted sum square error function. This function is not the same as those described before, that is a sum square error function with a penalty term. In this paper, we discuss the similarity between the objective function of FRLS and those of adding regularized type objectives mentioned above.

This paper is organized in nine sections. In the next section, a preliminary on the FRLS method will be introduced. Then, we present the main result briefly in Section III, and the relationship between the FRLS method and the weight decay method in Section IV. We derive, from the very first principle, two equations describing the expected mean training error and the expected mean prediction error. The former one will be derived in Section V and the latter one will be derived in Section VI. The derivation of the main result will thus be presented in Section VII. By comparing with the error equations obtained for recursive least square, we show that, under certain conditions, the model complexity and the expected prediction error of a model being trained by the FRLS method could both be smaller than that of being trained by using the RLS method in Section VIII. Finally, we conclude the paper in Section IX.

II. Preliminary

The model being discussed in this paper is the generalized linear model defined as follows:

$$y(x) = \varphi^T(x)\,\theta_0 + \epsilon \tag{1}$$

where $y \in R$; θ_0, $\varphi(x) \in R^n$; ϵ is a mean zero Gaussian noise; and $\varphi(x)$ is a nonlinear vector function depended on the input $x \in R^m$. The vector θ_0 is assumed to be the true model parameter.

In neural-network literature, model (1) represents many types of neural-network models. One example is the radial basis function network [2], [9] if the ith element of $\varphi(x)$, $\varphi_i(x)$, is defined as $\exp(-(1/2)(x-m_i)^T\Sigma_i(x-m_i))$, where $\Sigma_i \in R^{m\times m}$ is a fixed positive definite matrix and $m_i \in R^m$ is a fixed m-vector. θ_0 would then be the output weight vector. In nonlinear system modeling, model (1) can also represent a Volterra series [12].

Considering model (1), we define the estimator as follows:

$$\hat{y}(x) = \varphi^T(x)\,\hat{\theta} \tag{2}$$

where $\hat{\theta}$ is the estimate of the true parameter θ_0. By feeding the training data one by one, the estimate $\hat{\theta}$ can be updated iteratively based on the forgetting recursive least square method

Manuscript received February 10, 1998; revised February 8, 1999 and July 19, 1999.

C. S. Leung is with the Department of Electronic Engineering, City University of Hong Kong, Kowloon Tong, Hong Kong.

G. H. Young is with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Hong Kong.

J. Sum is with the Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong.

W.-k. Kan is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Shatin, N.T., Hong Kong.

Publisher Item Identifier S 1045-9227(99)09115-8.

[16]. Let $\hat{\theta}(t)$ be the optimal estimation of θ_0 when t data have been fed, the training can be accomplished via the following recursive equations:

$$K_t^* = \frac{\delta'}{1-\alpha} P_{t-1}\left(I + \frac{\delta'}{1-\alpha} P_{t-1}\right)^{-1} \tag{3}$$

$$P_t^* = \frac{1}{1-\alpha} P_{t-1} - \frac{1}{1-\alpha} K_t^* P_{t-1} \tag{4}$$

$$K_t = P_t^* \varphi(x_t)\left(I + \varphi^T(x_t) P_t^* \varphi(x_t)\right)^{-1} \tag{5}$$

$$\hat{\theta}(t) = \hat{\theta}(t-1) - K_t^* \hat{\theta}(t-1) + K_t \varphi^T(x_t) K_t^* \hat{\theta}(t-1) + K_t\left(y(x_t) - \varphi^T(x_t)\hat{\theta}(t-1)\right) \tag{6}$$

$$P_t = P_t^* - K_t \varphi^T(x_t) P_t^* \tag{7}$$

with the initial conditions

$$\hat{\theta}(0) = 0 \tag{8}$$

$$P(0) = \delta'^{-1} I_{n\times n} \tag{9}$$

and α is the forgetting factor in between zero and one.

In the theory of system identification [12], for a large t, the objective of the above recursive algorithm is to minimize the cost function $J(\theta(t))$, where

$$J(\theta(t)) = \sum_{k=1}^{t} w_k\left(y(x_k) - \varphi^T(x_k)\hat{\theta}(t)\right)^2 + \delta\|\theta(t)\|^2 \tag{10}$$

where $\{\varphi(x_k), y(x_k)\}_{k=1}^t$ is the training data set; $w_k = (1-\alpha)^{t-k}$; and $\delta = \delta'/\alpha$. The weighting factor w_k captures the effect of the most recent training data more. For $k = t$, the weighting on $(y(x_t) - \varphi^T(x_t)\hat{\theta}(t))$ is one. When $k = t-1$, the weighting on $(y(x_{t-1}) - \varphi^T(x_{t-1})\hat{\theta}(t))$ is $(1-\alpha)$. This factor is smaller than one. As a result, the factor w_k serves as a weighting factor which counts the effect of the most recent training data more than the earlier one.

III. The Main Result

A criteria for measuring the performance of (2) is the mean prediction error [1], that is the accuracy of the model in predicting the output of an unseen data x^F

$$\text{MPE}(t) = \int_{\Omega_\epsilon}\int_{\Omega_x}\left(y(x^F) - \varphi^T(x^F)\hat{\theta}(t)\right)^2 \cdot p(x^F)p(\epsilon)\,dx^F\,d\epsilon \tag{11}$$

where $p(x^F)$ and $p(\epsilon)$ are the probability density functions of x^F and ϵ, respectively. This MPE(t) is depended on the estimator $\hat{\theta}(t)$ and hence it is a random variable depended on the training set, $\{\varphi(x_k), y(x_k)\}_{k=1}^t$. Therefore, another criteria would be the expected mean prediction error [21], [23], [26] which is defined as follows:

$$\langle\text{MPE}(t)\rangle_{\xi_t} = \left\langle \int_{\Omega_\epsilon}\int_{\Omega_x}\left(y(x^F) - \varphi^T(x^F)\hat{\theta}(t)\right)^2 \cdot p(x^F)p(\epsilon)\,dx^F\,d\epsilon \right\rangle_{\xi_t}. \tag{12}$$

$\langle . \rangle_{\xi_T}$ denotes the expectation over the training set, $\xi_t = \{\varphi(x_k), \epsilon_t\}_{k=1}^t$.

Assuming that t is large enough and δ is very small. By using the similar technique as depicted in papers [1], [14], [21], [23], and [26], we can derive[1] that

$$\langle\text{MPE}(t)\rangle_{\xi_t} \approx \lambda_0\left[1 + \frac{2}{t}\sum_{k=1}^{n}\left(\frac{\hat{\delta}_k}{\hat{\delta}_k + \alpha\delta}\right)^2\right] \tag{13}$$

where λ_0 is the variance of the output noise ϵ_t and $\hat{\delta}_k$ is the kth eigenvalue of the matrix

$$H = \frac{1}{t}\sum_{k=1}^{t}\varphi(x_k)\varphi^T(x_k)$$

and

$$\lim_{t\to\infty} H = \langle\varphi(x)\varphi^T(x)\rangle_{\Omega_x}$$

where $\langle . \rangle_{\Omega_x}$ denotes the expectation over the training set, $\Omega_x = \{\varphi(x_k)\}_{k=1}^t$. Besides, if we define the mean training error as follows:

$$\langle\text{MTE}(t)\rangle_{\xi_t} = \left\langle \frac{1}{t}\sum_{k=1}^{t}\left(y(x_k) - \varphi^T(x_k)\hat{\theta}(t)\right)^2 \right\rangle_{\xi_t} \tag{14}$$

we could further relate the prediction error and the training error by the following equation:

$$\langle\text{MPE}(t)\rangle_{\xi_t} \approx \langle\text{MTE}(t)\rangle_{\xi_t} + 2\frac{\lambda_0}{t}\sum_{k=1}^{n}\frac{\hat{\delta}_k}{\hat{\delta}_k + \alpha\delta}. \tag{15}$$

The derivation of (15) will be shown in the following sections.

IV. FRLS and Weight Decay

Comparing (15) to that obtained from the standard weight decay method [21], it would be realized that the FRLS training method has an effect similar to the weight decay training. This result is extremely useful. The reason can be explained as below.

In the weight decay method, the cost function is defined as follows:

$$J_{WD}(\theta) = \frac{1}{N}\sum_{k=1}^{N}\left(y(x_k) - \varphi^T(x_k)\theta\right)^2 + c_0\|\theta\|^2 \tag{16}$$

where c_0 is the regularization factor controlling the penalty due to large weight. The estimate $\hat{\theta}$ is the one which minimizes $J_{WD}(\theta)$, that is,

$$\hat{\theta} = \arg\min_{\theta}\left\{\frac{1}{N}\sum_{k=1}^{N}\left(y(x_k) - \varphi^T(x_k)\theta\right)^2 + c_0\|\theta\|^2\right\}.$$

Now, based on the finding that FRLS training method is asymptotically identical to weight decay training, we can now have *an elegant on-line training method which can accomplish the same effect as weight decay* if $\alpha\delta = c_0$.

[1] This equation will be derived in Section VI.

V. Derivation of the Expected Mean Training Error

In accordance with the theory of identification, the objective of the FRLS is to minimize the cost function defined as (10)

$$J(\hat{\theta}(t)) = \sum_{k=1}^{t}(1-\alpha)^{t-k}\Big(y(x_k) - \varphi^T(x_k)\hat{\theta}(t)\Big)^2 + \delta\|\theta(\hat{t})\|^2$$

where $\{\varphi(x_k), y(x_k)\}_{k=1}^{t}$ is the training data set. Differentiating (10) once with respect to $\hat{\theta}(t)$ and equating it to zero, we can derive the solution of $\hat{\theta}(t)$

$$\hat{\theta}(t) = \left[\sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi(x_k)\varphi^T(x_k) + \delta I\right]^{-1} \cdot \left[\sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi(x_k)y(x_k)\right]. \tag{17}$$

Replacing $y(x_k)$ by its definition, (1), and using (17), it can be shown that

$$y(x_k) - \varphi^T(x_k)\hat{\theta}(t) = \epsilon_k + \delta\varphi^T(x_k)G_1^{-1}\theta_0 - \varphi^T(x_k)G_1^{-1} \cdot \left[\sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi(x_k)\epsilon_k\right] \tag{18}$$

where

$$G_1 = \sum_{l=1}^{t}(1-\alpha)^{t-l}\varphi(x_l)\varphi^T(x_l) + \delta I. \tag{19}$$

Note that for $k = 1, \cdots, t$, ϵ_k is a zero mean Gaussian noise with variance λ_0 for all $k = 1, 2, \cdots, t$. By squaring (18), summing up for k from 1 to t and taking the expectation over the set ξ_t, we can thus obtain an equation for the expected training error. Assuming that t is large enough

$$G_1 \approx \frac{1}{\alpha}\langle\varphi(x)\varphi^T(x)\rangle_{\Omega_x} + \delta I. \tag{20}$$

$$\left\langle\sum_{k=1}^{t}\Big(y(x_k) - \varphi^T(x_k)\hat{\theta}(t)\Big)^2\right\rangle_{\xi_T} = t\lambda_0 + \delta_0^2\sum_{k=1}^{t}\varphi^T(x_k)G_1^{-1}\theta_0\theta_0^TG_1^{-1}\varphi(x_k) + \lambda_0\sum_{k=1}^{t}\varphi^T(x_k)G_1^{-1}H_2G_1^{-1}\varphi(x_k) - 2\lambda_0\sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi^t(x_k)G_1^{-1}\varphi(x_k) \tag{21}$$

$$\approx t\lambda_0 + \delta^2\mathrm{tr}\{HG_1^{-1}\theta_0\theta_0^TG_1^{-1}\} - 2\lambda_0\mathrm{tr}\{H_1G_1^{-1}\} + \lambda_0\mathrm{tr}\{HG_1^{-1}H_2G_1^{-1}\} \tag{22}$$

where tr is the trace operator

$$H_1 = \sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi(x_k)\varphi^T(x_k) \approx \frac{1}{\alpha}\langle\varphi(x)\varphi^T(x)\rangle_{\Omega_x} \tag{23}$$

$$H_2 = \sum_{k=1}^{t}(1-\alpha)^{2(t-k)}\varphi(x_k)\varphi^T(x_k) \tag{24}$$

$$\approx \frac{1}{1-(1-\alpha)^2}\langle\varphi(x)\varphi^T(x)\rangle_{\Omega_x}. \tag{25}$$

Therefore, the expected mean training error can be rewritten as follows:

$$\langle\mathrm{MTE}(t)\rangle_{\xi_T} = \frac{1}{N}\left\langle\sum_{k=1}^{t}\Big(y(x_k) - \varphi^T(x_k)\hat{\theta}(t)\Big)^2\right\rangle_{\xi_T} \tag{26}$$

$$= \lambda_0 + \frac{\lambda_0}{N}\left(\mathrm{tr}\{HG_1^{-1}H_2G_1^{-1}\} - 2\mathrm{tr}\{H_1G_1^{-1}\}\right) + \frac{\delta^2}{N}\mathrm{tr}\{HG_1^{-1}\theta_0\theta_0^TG_1^{-1}\}. \tag{27}$$

VI. Derivation of the Expected Mean Prediction Error

Next, we are going to derive the equation for the expected mean prediction error defined in (12). First, let us derive an equation for $\theta_0 - \hat{\theta}(t)$. Using the result in (17) once again, we can readily show that

$$\theta_0 - \hat{\theta}(t) = \theta_0 - G_1^{-1}\left[\sum_{k=1}^{t}(1-\alpha)^{t-k}y(x_k)\varphi(x_k)\right] = \delta G_1^{-1}\theta_0 - G_1^{-1}\sum_{k=1}^{t}(1-\alpha)^{t-k}\epsilon_k\varphi(x_k) \tag{28}$$

and hence

$$\langle(\theta_0 - \hat{\theta}(t))(\theta_0 - \hat{\theta}(t))^T\rangle_{\xi_t} = \delta^2G_1^{-1}\theta_0\theta_0^TG_1^{-1} + \lambda_0G_1^{-1}H_2G_1^{-1}. \tag{29}$$

Recall that the definition of the expected mean prediction error is as follows:

$$\langle\mathrm{MPE}(t)\rangle_{\xi_t} = \left\langle\int_{\Omega_c}\int_{\Omega_x}\Big(y(x^F) - \varphi^T(x^F)\hat{\theta}(t)\Big)^2p(x^F)p(\epsilon)\,dx^Fd\epsilon\right\rangle_{\xi_t}.$$

Since $\hat{\theta}(t)$ is a random variable independent of x and ϵ, (12) can be rewritten as follows:

$$\langle\mathrm{MPE}(t)\rangle_{\xi_t} = \lambda_0 + \mathrm{tr}\left\{\int_{\Omega_x}\varphi(x^F)\varphi^T(x^F)p(x^F)\,dx^F \cdot \left\langle\Big(\theta_0 - \hat{\theta}(t)\Big)\Big(\theta_0 - \hat{\theta}(t)\Big)^T\right\rangle_{\xi_t}\right\}. \tag{30}$$

Suppose that t is large enough, we approximate $\int_{\Omega_x}\varphi(x)\varphi^T(x)p(x)\,dx$ by $t^{-1}H$. By using (29), we can show that

$$\langle\mathrm{MPE}(t)\rangle_{\xi_t} \approx \lambda_0 + \frac{\delta^2}{t}\mathrm{tr}\{\delta^2G_1^{-1}\theta_0\theta_0^TG_1^{-1}\} + \frac{\lambda_0}{t}\mathrm{tr}\{G_1^{-1}H_2G_1^{-1}\}. \tag{31}$$

VII. Derivation of Equation for MPE and MTE

Comparing (31) and (27), it can be shown that

$$\langle \text{MPE}(t)\rangle_{\xi_T} \approx \langle \text{MTE}(t)\rangle_{\xi_T} + \frac{2\lambda_0}{t}\text{tr}\{H_1 G_1^{-1}\}. \tag{32}$$

As when t is large

$$H_1 = \sum_{k=1}^{t}(1-\alpha)^{t-k}\varphi(x_k)\varphi^T(x_k) \tag{33}$$

$$\approx \frac{1}{\alpha t} H \tag{34}$$

$$\approx \frac{1}{\alpha t}\int_{\Omega_x} \varphi(x)\varphi^T(x)p(x)\,dx \tag{35}$$

and

$$G_1 \approx \frac{1}{\alpha t} H + \delta I \tag{36}$$

$$\approx \frac{1}{\alpha t}\int_{\Omega_x} \varphi(x)\varphi^T(x)p(x)\,dx + \delta I. \tag{37}$$

Using the asymptotic approximations, (34) and (36), for H_1 and G_1, we could get that

$$\text{tr}\{H_1 G_1^{-1}\} \approx \text{tr}\left\{\frac{1}{t} H\left[\frac{1}{t} H + \alpha\delta I\right]^{-1}\right\}. \tag{38}$$

Let $\hat{\delta}_k$ be an estimate of the kth eigenvalue of the matrix $\int_{\Omega_x}\varphi(x)\varphi^T(x)p(x)\,dx$

$$\langle \text{MPE}(t)\rangle_{\xi_T} \approx \langle \text{MTE}(t)\rangle_{\xi_T} + 2\,\frac{\lambda_0}{t}\sum_{k=1}^{n}\frac{\hat{\delta}_k}{\hat{\delta}_k + \alpha\delta}. \tag{39}$$

VIII. Comparison with Recursive Least Square

Once the factor α is being set to zero, it should be noted that the algorithms (3)–(7) can be reduced to the standard recursive least square (RLS) method. Using the similar technique, (9), (23), and (25), the following equalities will be obtained:

$$G_1(\alpha = 1) = G \tag{40}$$

$$H_1(\alpha = 1) = H \tag{41}$$

$$H_2(\alpha = 1) = H. \tag{42}$$

Then the mean prediction error and the mean training error for the RLS method can readily be derived

$$\langle \text{MPE}(t)\rangle_{\xi_T} \approx \lambda_0 + \frac{\delta^2}{t}\,\text{tr}\{\delta^2 G^{-1}\theta_0\theta_0^T G^{-1}\} + \frac{\lambda_0}{t}\text{tr}\{G^{-1}HG^{-1}\}. \tag{43}$$

$$\langle \text{MTE}(t)\rangle_{\xi_T} = \lambda_0 + \frac{\lambda_0}{t}\left(\text{tr}\{HG^{-1}HG^{-1}\} - 2\text{tr}\{HG^{-1}\}\right) + \frac{\delta^2}{t}\text{tr}\{HG^{-1}\theta_0\theta_0^T G^{-1}\}. \tag{44}$$

In such case the difference between the expected mean prediction error and the expected mean training error would be equal to $2(\lambda_0/t)\text{tr}\{HG^{-1}\}$, i.e.,

$$\langle \text{MPE}(t)\rangle_{\xi_T} \approx \langle \text{MTE}(t)\rangle_{\xi_T} + 2\frac{\lambda_0}{t}\text{tr}\{HG^{-1}\} = \langle \text{MTE}(t)\rangle_{\xi_T} + 2\,\frac{\lambda_0}{t}\sum_{k=1}^{n}\frac{\hat{\delta}_k}{\hat{\delta}_k + \delta/t}. \tag{45}$$

Suppose t is very large, the second term in (45) would be equal to $2\lambda_0 n/t$.

If we define the network complexity as the effective number of parameter, (39) and (45) reveals that the complexity of the models being trained by using the FRLS is usually smaller than that of using the RLS.

Apart from the difference in the model complexity, we could also show that under certain conditions, the expected mean prediction error generated by the network being trained by the FRLS is smaller than that of using the RLS. Again, we consider the asymptotic situation. We let $\langle\varphi(x)\varphi^T(x)\rangle$ be $\int_{\Omega_x}\varphi(x)\varphi^T(x)p(x)\,dx$. The following approximations can readily be obtained:

$$H \approx \frac{1}{t}\langle\varphi(x)\varphi^T(x)\rangle \tag{46}$$

$$H_1 \approx \frac{1}{\alpha}\langle\varphi(x)\varphi^T(x)\rangle \tag{47}$$

$$H_2 \approx \frac{1}{2\alpha-\alpha^2}\langle\varphi(x)\varphi^T(x)\rangle \tag{48}$$

$$G_1 \approx \frac{1}{\alpha}\langle\varphi(x)\varphi^T(x)\rangle + \delta I. \tag{49}$$

Using these approximated equations and considering the factors $G_1^{-1}H_2G_1^{-1}$ and $G^{-1}HG^{-1}$ in (31) and (43), one can show that

$$G_1^{-1}H_2G_1^{-1} \approx \left[\frac{1}{\alpha}\langle\varphi\varphi^T\rangle + \delta I\right]^{-1}\left[\frac{1}{2\alpha-\alpha^2}\langle\varphi\varphi^T\rangle\right]\cdot\left[\frac{1}{\alpha}\langle\varphi\varphi^T\rangle + \delta I\right]^{-1} \tag{50}$$

$$G^{-1}HG^{-1} \approx \left[t\langle\varphi\varphi^T\rangle + \delta I\right]^{-1}\left[t\langle\varphi\varphi^T\rangle\right]\cdot\left[t\langle\varphi\varphi^T\rangle + \delta I\right]^{-1} \tag{51}$$

and *if*

$$\frac{1}{\alpha} > t > \frac{1}{2\alpha} \tag{52}$$

or equivalently

$$\frac{1}{t} > \alpha > \frac{1}{2t} \tag{53}$$

the expected mean prediction error of using the FRLS will be smaller than that of using the RLS.

IX. Conclusion

In this paper, we have presented certain analytical results regarding the use of forgetting recursive least square method

in the training of a linear neural network. The expected mean prediction error and the expected mean training error are derived from the first principle with the assumptions that the number of training data is large and the output noise ϵ is a zero mean Gaussian noise. Using these error equations, we are able to analyze and compare the behavior of the FRLS with the RLS. First, we have shown that *the FRLS has inherent weight decay (regularization) effect*. Second, we have shown that *the expected mean prediction error of using the FRLS can be smaller than that of using the RLS if the forgetting factor α is set appropriately*.

REFERENCES

[1] A. Barron, "Prediction squared error: A criterion for automatic model selection," *Self-Organizing Methods on Modeling*, S. Farlow, Ed. New York: Marcel Dekker, 1984.

[2] C. M. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Comput.*, vol. 3, no. 4, pp. 579–588, 1991.

[3] ———, "Training with noise is equivalent to Tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, 1995.

[4] S. Chen, C. Cowan, S. A. Billings, and P. M. Grant, "Parallel recursive prediction error algorithm for training layered neural networks," *Int. J. Contr.*, vol. 51, no. 6, pp. 1215–1228, 1990.

[5] S. Chen, S. A. Billings, C. F. N. Cowan, and P. M. Grant, "Practical identification of NARMAX models using radial basis functions," *Int. J. Contr.*, vol. 52, pp. 1327–1350, 1990.

[6] S. Chen, S. A. Billings, and P. M. Grant, "Recursive hybrid algorithm for nonlinear system identification using radial basis function networks," *Int. J. Contr.*, vol. 55, no. 5, pp. 1051–1070, 1992.

[7] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communication channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–579, 1993.

[8] S. Chen and J. Wigger, "Fast orthogonal least squares algorithm for efficient subset model selection," *IEEE Trans. Signal Processing*, vol. 43, pp. 1713–1715, 1995.

[9] D. Gorinevsky, "On the persistency of excitation in radial basis function network identification of nonlinear systems," *IEEE Trans. Neural Networks*, vol. 6, pp. 1237–1244, 1995.

[10] S. Haykin, *Neural networks: A Comprehensive Foundation.* New York: Macmillan, 1994.

[11] T. A. Johansen, "On Tikhonov regularization, bias and variance in nonlinear system identification," *Automatica*, vol. 33, no. 3, pp. 441–446, 1997.

[12] R. Johansson, *System Modeling and Identification.* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[13] S. Kollias and D. Anastassiou, "An adaptive least squares algorithm for the efficient training of artificial neural networks," *IEEE Trans. Circuits Syst.*, vol. 36, pp. 1092–1101, 1989.

[14] J. Larsen and L. K. Hansen, "Generalization performance of regularized neural network models," in *Proc. IEEE Wkshp. Neural Networks Signal Processing IV*, 1994, pp. 42–51.

[15] C. S. Leung, K. W. Wong, J. Sum, and L. W. Chan, "On-line training and pruning for RLS algorithms," *Electron. Lett.*, vol. 7, pp. 2152–2153, 1996.

[16] C. S. Leung, P. F. Sum, A. C. Tsoi, and L. W. Chan, "Several aspects of pruning methods in recursive least square algorithms for neural networks," in *Proc. TANC-97*, 1997, pp. 71–80.

[17] S. J. Chang, K. W. Wong, and C. S. Leung, "Periodic activation function for fast on-line EKF training and pruning," *Electron. Lett.*, vol. 34, pp. 2255–2256, 1998.

[18] J. Sum, C. S. Leung, L. W. Chan, W. K. Kan, and G. Young, "An adaptive Bayesian pruning for neural network in nonstationary environment," *Neural Comput.*, accepted for publication.

[19] L. Ljung, J. Sjöberg, and T. McKelvey, "On the use of regularization in system identification," Dept. Elect. Eng., Linkoping University, Sweden, Tech. Rep., 1992.

[20] J. Moody, "Note on generalization, regularization, and architecture selection in nonlinear learning systems," in *1st IEEE-SP Wkshp. Neural Networks Signal Processing*, 1991.

[21] ———, "The effective number of parameters: An analysis of generalization regularization in nonlinear learning systems," *Advances in Neural Information Processing Systems 4*, pp. 847–854, 1992.

[22] ———, "Prediction risk and architecture selection for neural networks," in *From Statistics to Neural Networks: Theory and Pattern Recognition Application*, V. Cherkassky *et al.*, Eds. Berlin, Germany: Springer-Verlag, 1994.

[23] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion—Determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Networks*, vol. 5, pp. 865–872, 1994.

[24] M. W. Pedersen, L. K. Hansen, and J. Larsen, "Pruning with generalization-based weight saliencies: γOBD, γOBS," *Advances in Information Processing Systems 8*, pp. 521–528, 1996.

[25] S. Shah, F. Palmieri, and M. Datum, "Optimal filtering algorithm for fast learning in feedforward neural networks," *Neural Networks*, vol. 5, pp. 779–787, 1992.

[26] J. Sjöberg and L. Ljung, "Overtraining, regularization and searching for a minimum, with application to neural networks," *Int. J. Contr.*, vol. 62, pp. 1391–1407, 1995.

[27] A. N. Tikhonov, "Incorrect problems of linear algebra and a stable method for their solution," *Doklady*, vol. 163, no. 3, pp. 988–991, 1965.

[28] L. Wu and J. Moody, "A smoothing regularizer for feedforward and recurrent neural networks," *Neural Comput.*, vol. 8, pp. 461–489, 1996.