Radial basis function neural networks are used in a variety of applications such as pattern recognition, nonlinear identification, control and time series prediction. In this paper, the learning algorithm of radial basis function neural networks is analyzed in a feedback structure. The robustness of the learning algorithm is discussed in the presence of uncertainties that might be due to noisy perturbations at the input or to modeling mismatch. An intelligent adaptation rule is developed for the learning rate of RBFNN which gives faster convergence via an estimate of error energy while giving guarantee to the $l_2$ stability governed by the upper bounding via small gain theorem. Simulation results are presented to support our theoretical development.

1. Introduction

Neural Networks have recently been used in almost every field of science. Radial basis function neural networks (RBFNN) are single-layered feedforward networks with universal approximation capabilities, in addition to more efficient learning than the famous multilayered feedforward neural networks (MFNN) [1]. RBFNN has been used in a wide variety of applications such as in [2–9].

RBFNN are generally trained using supervised learning. During training, a recursive update procedure is used to estimate the weights of the RBFNN that best fit the given data [1]. The recursive procedure often requires selecting a suitable adaptation gain called learning rate. The learning rate should be within an optimum range. It should neither be too large which would drive the algorithm unstable, nor too small that it slows down the training. In general practice, trial-and-error experiences are used to select a suitable learning rate for training phase.

Due to inherent nonlinearity in the structure of neural networks, its convergence analysis becomes complicated. A very limited work has been done in this context. In [8–10], the authors have presented a robustness analysis for the perceptron neural network. The formulation in [8–10] emphasizes an intrinsic feedback structure for most adaptive algorithms and it relies on tools from system theory, control, and signal processing such as state-space description, feedback analysis, small gain theorem, $H^\infty$ design, and lossless systems. The feedback configuration is provoked via energy arguments and is shown to consist of two major blocks: a time-variant lossless (i.e., energy preserving) feedforward path and a time-variant feedback path.

More recently, in [11], convergence analysis of RBFNN is carried out and optimal adaptation for its learning rate is derived using the deterministic feedback analysis aided with small gain theorem. But the work does not include any adaptive mechanism for the learning rate in true sense. In contrast, in this work, we present an intelligent adaptation rule for the learning rate of RBFNN which gives faster convergence via an estimate of error energy while giving guarantee to the $l_2$ stability governed by the upper bounding via small gain theorem. Moreover, unlike the work of [11], we avoid mean value theorem, thanks to the RBFNN structure which allows us to separate the nonlinearity with its weights.
2. Radial Basis Functions Neural Networks

RBFNN is a type of feedforward neural network. They are used in a wide variety of contexts such as function approximation, pattern recognition, and time series prediction. Networks of this type have the universal approximation property [1]. In these networks the learning involves only one layer with lesser computations. A multi-input multioutput RBFNN is shown in Figure 1. The RBFNN consists of an input node \( u(t) \), a hidden layer with \( n_o \) neurons, and an output node \( y(t) \). Each of the input nodes is connected to all the nodes or neurons in the hidden layer through unity weights (direct connections). While each of the hidden layer nodes is connected to the output node through some weights, for example, the \( i \)th output node is connected with all the hidden layer nodes by \( W(t) = [w_{1i}(t), \ldots, w_{ni}(t)] \), each neuron finds the distance, normally applying Euclidean norm, between the input and its center and passes the resulting scalar through a nonlinearity. So the output of the \( i \)th hidden neuron is given by \( \phi_i(||u(t) - c_i||) \), where \( c_i \) is the center of the \( i \)th hidden layer node, \( i = 1, 2, \ldots, n_o \), and \( \phi_i(\cdot) \) is the nonlinear basis function. Normally this function is taken as a Gaussian function of width \( \beta \), which dictates the effective range of input passing through the basis function. The output \( y_m(t) \) is a weighted sum of the outputs of the hidden layer, given by

\[
\begin{align*}
y_m(t) &= \Phi(t) W(t), \\
y_m(t) &= \sum_{i=1}^{n_o} \phi_i \left( ||u(t) - c_i|| \right) w_i(t),
\end{align*}
\]

where the basis functions and weight vector are defined as

\[
\Phi(t) = \left[ \phi_1(u(t)), \phi_2(u(t)), \ldots, \phi_{n_o}(u(t)) \right],
\]

and the Gaussian basis function is

\[
\phi_i(u(t)) = \exp \left( -\frac{||u(t) - c_i||^2}{\beta^2} \right).
\]

Consider a collection of input vectors \( \{u(t)\} \) with the corresponding desired output vectors \( \{y(t)\} \). The objective is to estimate the unknown optimal weight \( W_o \). Now, starting with an initial guess \( W_0 \), the weights are updated recursively based on the LMS principle as

\[
W(t + 1) = W(t) + \alpha(t) e(t) \Phi^T(t),
\]

where \( \alpha(t) \) is the learning and the error \( e(t) \) is defined as

\[
e(t) = y(t) - y_m(t),
\]

\[
e(t) = \Phi(t) W_o - \Phi(t) W(t) + v(t).
\]

We define \( a \) priori and \( a \) posteriori error quantities as

\[
e_a(t) = \Phi(t) \hat{W}(t),
\]

\[
e_p(t) = \Phi(t) \tilde{W}(t + 1),
\]

where \( \hat{W}(t) \) is the weight error vector symbolizing the difference between the optimal weight and its estimate as \( \hat{W}(t) = W_o - W(t) \). Thus, we can rewrite the \( e_a(t) \) as

\[
e_a(t) = \Phi(t) W_o - \Phi(t) W(t)
\]

\[
= \Phi(t) W_o - y_m(t),
\]

\[
e_p(t) = \Phi(t) \left[ \hat{W}(t) - \alpha(t) \Phi(t)^T e(t) \right]
\]

\[
= e_a(t) - \alpha(t) \Phi(t) \| e(t) \|^2.
\]

Consequently, the weight error update equation satisfies the following recursion:

\[
\hat{W}(t + 1) = \hat{W}(t) - \alpha(t) e(t) \Phi^T(t).
\]
3. A Deterministic Framework for the Robustness of RBFNN

Robustness of an algorithm is defined as the consistency in its estimation error with the disturbances in the sense that a minor increase in disturbances would lead to a smaller increase in its estimation error irrespective of the disturbances nature. In order to study the robustness of RBFNN, we employ a pure deterministic framework without assuming any prior knowledge of signal or noise statistics as was used in [8, 9]. This is especially useful in situations where prior statistical information is missing. The robust design was used in [8, 9]. This is especially useful in situations where assuming any prior knowledge of signal or noise statistics is not possible.

By rearranging the relevant terms, we finally arrive at

\[
\| \tilde{W}(t+1) \|^2 + \alpha(t) e_a^2(t) - \frac{\alpha(t)^2}{\mu(t)} e_a^2(t) \\
= \| \tilde{W}(t) \|^2 - 2\alpha(t) e_a(t) \tilde{v}(t) + \frac{\alpha(t)^2}{\mu(t)} e_a^2(t) (t) + \frac{\alpha(t)^2}{\mu(t)} \tilde{v}(t),
\]

where we have introduced a new parameter \( \mu(t) \) defined as

\[
\mu(t) = \frac{1}{\| \Phi(t) \|^2}.
\]

Thus, it can be easily seen from the mapping (15) that the following three different scenarios exist depending upon the value of learning rate:

\[
\| \tilde{W}(t+1) \|^2 + \alpha(t) e_a^2(t) \leq 1, \quad \text{for } 0 < \alpha(t) < \mu(t) \\
\|
\tilde{W}(t) \|^2 + \alpha(t) \tilde{v}^2(t) \overset{\leq}{=} 1, \quad \text{for } \alpha(t) = \mu(t), \\
\|
\tilde{W}(t) \|^2 + \alpha(t) \tilde{v}^2(t) \overset{\geq}{=} 1, \quad \text{for } \alpha(t) > \mu(t).
\]

The first two inequalities in the statement of (3) assert that if the learning rate is chosen such that \( \alpha(t) \leq \mu(t) \), then the mapping from the signals \( \tilde{W}(t), \sqrt{\mu(t)} e_p(t) \) to the signals \( \tilde{W}(t+1), \sqrt{\mu(t)} e_a(t) \) is a lossless or contractive mapping. Therefore, a local energy bound is deduced that highlights a robustness property of the update recursion. The energy bound depicts that no matter what the value of the noise component \( \tilde{v}(t) \) is, and no matter how far the estimate \( \tilde{W}(t) \) is from the optimal \( W_o \), the sum of energies \( \|
\tilde{W}(t+1) \|^2 + \alpha(t) e_a^2(t) \) will always be smaller than or equal to the sum of energies \( \|
\tilde{W}(t) \|^2 + \alpha(t) \tilde{v}^2(t) \). Since this contractive property holds for each time instant, it should also hold globally over any interval. In fact, selecting \( \alpha(t) < \mu(t) \) over the interval \( 0 \leq t \leq N \), it follows that

\[
\|
\tilde{W}(N) \|^2 + \sum_{t=0}^{N} \alpha(t) e_a^2(t) \leq \|
\tilde{W}(0) \|^2 + \sum_{t=0}^{N} \alpha(t) \tilde{v}^2(t).
\]

4. Feedback Structure for Lossless Mapping

In this section, a feedback structure is established that explains a lossless mapping between estimation errors while adapting the weights from the \( r \)th time instant to the \((r+1)\)th time instant. A lossless mapping is the one that transforms \( x \) to \( y = T[x] \) in such a way that we have \( \| T[x] \|^2 \leq \| x \|^2 \) for all \( x \); that is, the output energy does not exceed the input energy. To set up the stage for the analysis, we define the disturbance error \( \tilde{v}(t) \) as

\[
\tilde{v}(t) = \tilde{e}(t) - e_a(t).
\]

Now, using the above definition and definitions of estimation errors, we evaluate the energies of both sides of the weight error recursion (11) as follows:

\[
\|
\tilde{W}(t+1) \|^2 = \|
\tilde{W}(t) \|^2 - 2\alpha(t) e_a(t) \Phi(t) \tilde{W}(t) + \alpha(t)^2 \|
\Phi(t) \|^2 e_a^2(t) + 2\alpha(t) \|
\Phi(t) \|^2 e_a(t) \tilde{v}(t) + \alpha(t)^2 \|
\Phi(t) \|^2 \tilde{v}^2(t).
\]

By rearranging the relevant terms, we finally arrive at

\[
\|
\tilde{W}(t+1) \|^2 + 2\alpha(t) e_a^2(t) - \frac{\alpha(t)^2}{\mu(t)} e_a^2(t) \\
= \|
\tilde{W}(t) \|^2 - 2\alpha(t) e_a(t) \tilde{v}(t) + \frac{\alpha(t)^2}{\mu(t)} e_a^2(t) (t) + \frac{\alpha(t)^2}{\mu(t)} \tilde{v}(t),
\]

where

\[
\mu(t) = \frac{1}{\| \Phi(t) \|^2}.
\]
The feedback configuration such as the configuration in Figure 2 as a special case requires that the product of norms of the feedforward and feedback maps be strictly bounded by one [8, 9, 12]. In our case, the norm of the feedforward map is equal to one (since it is lossless) while the norm of the feedback map is defined in (23) as \( \Delta(N) \). Hence, the condition \( \Delta(N) < 1 \) guarantees an overall contractive map. Therefore, for \( \Delta(N) < 1 \) to hold, we need to choose the learning rate such that, for all \( t \),

\[
0 < \alpha(t) = \frac{2}{\|\Phi(t)\|^2}.
\]  

6. Designing Adaptive Learning Rate

In this section, we propose an adaptive mechanism to update the learning rate \( \alpha(t) \) such that it gives faster convergence as well as guaranteeing the \( l_2 \) stability discussed in the previous section. For this, we propose an adaptive mechanism similar to the one in [13] according to which the learning rate should be adapted via an estimate of error correlation. In addition, we upper-bounded the maximum value of the learning rate to assure its \( l_2 \) stability by employing the stability bound derived in (24). To do so, we propose the following adaptive rule [13]:

\[
\alpha(t + 1) = \lambda \eta(t) + \gamma e^2(t), \quad (0 < \lambda < 1, \gamma > 0),
\]  

\[
\eta(t + 1) = \begin{cases} 
\eta_{\text{max}}, & \text{if } \alpha(t + 1) > \eta_{\text{max}}, \\
\frac{\alpha(t + 1)}{\alpha(t + 1)}, & \text{otherwise,}
\end{cases}
\]  

where the parameter \( \eta_{\text{max}} \) is so chosen that it ensures \( l_2 \) stability given in (23). Thus, \( \eta_{\text{max}} \) is given by

\[
\eta_{\text{max}} = \frac{2}{\|\Phi(t)\|^2},
\]  

where the parameter \( \lambda \) is a positive quantity showing its dependency on its own past value and lies in the range [0, 1] (usually we choose a value closer to 1, e.g., 0.97) while the constant \( \gamma \) is a very small number. The adaptation rule given by (25) and (26) suggests that the learning rate is large in the initial stage of adaptation due to larger error correlation and it decreases near steady state as the error correlation of the algorithm also decreases once the algorithm approaches the steady state. Thus, by adjusting the learning rate online according to the rule given in (25), it will give faster convergence as it allows faster adaptation of \( \alpha(t) \) via an estimate of error energy due to the term \( ye^2(t) \). On the other hand, the adaptation rule in (24) will guarantee the stability of the feedback structure due to upper bounding via the stability limit in (24), that is, \( 2/\|\Phi(t)\|^2 \). Thus, it promises both faster convergence and a stable response.

7. Simulation Results

The proposed adaptive learning rate is verified using various simulations for nonlinear identification and tracking control. In all the cases the simulation is first performed for fixed

![Figure 2: A lossless mapping in a feedback structure for RBFNN learning algorithm.](image-url)
learning rates. The fixed learning rates are set after several trials. However, these trials are not required when using the proposed adaptive learning rate given by (24), (25), and (26). A comparison for different fixed learning rates and adaptive learning rate is shown for each example along with identification/tracking and learning rate trends.

7.1. Identification of Nonlinear Control Valve. In this simulation example, the proposed adaptive learning rate is used in the identification of a model that describes a valve for control of fluid flow described in [14] as

$$y(t) = \frac{u(t)}{\sqrt{0.10 + 0.90u^2(t)}}$$

(28)

The model is identified using an RBFNN with 5 centers spaced at 0.5. The width of the center is set to 0.6. An output additive noise of 30 dB SNR is considered in this example. Learning rates of 0.01, 0.03, 0.06, and 0.08 are used for fixed learning rate case. The algorithm became unstable and values are near or greater than 0.08. After the simulations, mean square errors (MSE) for fixed and adaptive learning rates are shown in Figure 3. The lowest MSE achieved using adaptive learning rate shows the performance of the proposed approach.

Actual and identified control valve using the proposed approach are shown in Figure 4. The learning rate trend can be seen in Figure 5.

7.2. Identification of Nonlinearity in Hammerstein Model. The proposed bound on the adaptive learning rate is used in the identification of the static nonlinearity in nonlinear Hammerstein model defined in [15]. The Hammerstein model has been identified using RBFNN in [16]. The Hammerstein model used for simulation represents a nonlinear heat exchanger in cascade with linear dynamics. The static nonlinearity and linear dynamics are given by [15]

$$x(t) = -31.549u(t) + 41.732u^2(t) - 24.201u^3(t) + 68.634u^4(t),$$

$$y(t) = 0.4y(t-1) + 0.35y(t-2) + 0.15x(t) + v(t),$$

(29)

where \(u(t)\) is the input to the system, \(x(t)\) is the intermediate variable, \(y(t)\) is the system output, and \(v(t)\) is additive noise at the output. Actual and identified heat exchangers using the proposed approach are shown in Figure 7. The nonlinearity...
Mean square error using fixed and proposed adaptive learning rate

Figure 6: MSE for the identification of heat exchanger in Hammerstein model using fixed and adaptive learning rates. The fixed learning rates are 0.01, 0.02, 0.03, and 0.05.

Nonlinearity in the Hammerstein model identified using adaptive learning rate

Figure 7: Actual and identified heat exchanger in Hammerstein model using the proposed adaptive learning rate.

in the Hammerstein model is identified online using RBFNN with 9 neurons and a width of 0.5. An output additive noise of 30 dB SNR is considered in this example. Learning rates of 0.01, 0.02, 0.03, and 0.05 are used for fixed learning rate case. The algorithm became unstable at values near or greater than 0.07. After the simulations mean square errors (MSE) for fixed and adaptive learning rates are shown in Figure 6, the lowest MSE achieved using adaptive learning rate shows the performance of the proposed approach.

The learning rate trend can be seen in Figure 8.

7.3. Adaptive Inverse Control Using RBFNN. In this simulation example an adaptive control technique, namely, adaptive inverse control (AIC) is considered [17]. This technique is based on identifying the plant and its inverse. This technique was introduced for stable, minimum phase linear systems; however, with appropriate modification it can also be used for nonminimum phase and nonlinear systems [17].

Consider a nonminimum phase plant with a transfer function:

\[
G(s) = \frac{s(s+1)}{s^2 + 0.45s + 0.1}.
\] (30)

AIC based on RBFNN is used for tracking control of the given plant. Therefore, the plant is identified online using RBFNN with 5 centers and width of 1. The weights are initialized with random numbers. These initial weights are kept the same for all different cases of learning rates. The response to a square wave with output additive noise is measured and compared with the RBFNN output. The mismatch signal is used as error signal in weight update (5).

For fixed learning rate case, learning rates of 0.1, 0.5, 0.8, and 0.9 are used. The algorithm became unstable at learning rate of 1. MSE for different learning rates and proposed adaptive learning rate are shown in Figure 9. It is observed that the MSE converges faster with higher learning rates. On the other hand, using the proposed adaptive learning rate MSE converges to smaller values than any of the fixed learning rate cases.

The tracking of actual and identified nonminimum phase plant using the proposed approach is shown in Figure 10. The learning rate trend can be seen in Figure 11.

7.4. Internal Model Control of MIMO System Using RBFNN Based U-Model. In this simulation example internal model control (IMC) [18] is applied for the tracking control of a 2-input 2-output system. The plant is modelled by RBFNN based U-model. The details of the RBFNN based U-model are presented in [19, 20].
The 2-input 2-output system is given by

\[
\begin{align*}
y_1(t) &= 0.21y_1(t-1) - 0.12y_2(t-2) \\
&\quad + 0.3y_1(t-1)u_2(t-1) - 1.6u_2(t-1) + v_1(t), \\
y_2(t) &= 0.25y_2(t-1) - 0.1y_1(t-2) \\
&\quad - 0.2y_2(t-1)u_1(t-1) + 1.2u_1(t-1) + v_2(t),
\end{align*}
\]  

(31)

where \(y_i\) are the output, \(u_i\) are the input, and \(v_i\) are output additive noise.

IMC based on U-model is used for tracking control of the MIMO system. Therefore, the plant is identified online using 2-input 2-output RBFNN based U-model [19] with 4 centers and width of 1.

For fixed learning rate case, learning rates of 0.1, 0.2, 0.5, and 0.9 are used. MSE for different learning rates and proposed adaptive learning rate are shown in Figure 12.

It can be seen that adaptive learning rate has outperformed the fixed learning rate. The tracking of actual and identified nonminimum phase plant using the proposed
The approach is shown in Figure 13. The learning rate trend can be seen in Figure 14.

8. Conclusions

This paper presents the convergence analysis of the RBFNN with a deterministic framework. An adaptive learning rate is designed which is a result of time domain feedback analysis of RBFNN learning algorithm. The proposed adaptive rule for the learning rate gives faster convergence via an estimate of error energy while giving guarantee to the $l_2$ stability governed by the upper bounding via small gain theorem. Performance of the proposed adaptive learning rate is verified by a number of identification and tracking control examples of nonlinear systems. The effectiveness of the proposed approach is observed by better MSE compared to the one with a fixed learning rate.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References


Submit your manuscripts at
http://www.hindawi.com