

Annealing Robust Fuzzy Neural Networks for Modeling of Molecular Autoregulatory Feedback Loop Systems

Chen-Chia Chuang

Abstract

In this paper, the annealing robust fuzzy neural networks (ARFNNs) are proposed to improve the problems of fuzzy neural networks for modeling of the molecular autoregulatory feedback loop systems with outliers. Firstly, the support vector regression (SVR) approach is proposed to determine the initial structure of the ARFNNs. Because of a SVR approach is equivalent to solving a linear constrained quadratic programming problem under a fixed structure of SVR, the number of hidden nodes, the initial parameters and the initial weights of the ARFNNs are easy obtained via the SVR approach. Moreover, the results of the SVR are used as initial structure in the ARFNNs. At the same time, an annealing robust learning algorithm (ARLA) is used as the learning algorithm for the ARFNNs, and applied to adjust the parameters in the membership function as well as weights of the ARFNNs. That is, an ARLA is proposed to overcome the problems of initialization and the cut-off points in the robust learning algorithm. Hence, when an initial structure of the ARFNNs are determined by a SVR approach, the ARFNNs with the ARLA have fast convergence speed and robust against outliers for the modeling of the molecular autoregulatory feedback loop systems.

Keywords: Annealing robust fuzzy neural networks, Molecular autoregulatory feedback loop systems, Outliers, Support vector regression, Annealing robust learning algorithm.

1. Introduction

There are a large number of factors that complicate the mathematical modeling of biological gene networks. Many complex interactions on the molecular scale have been reported in [1-2], which rely on the presence of

specific factors that can either enhance or inhibit the expression of certain genes. To increase the amount of data from which to start network inference, D'Haeseleer et al. [3] started their analysis with the calculation of a non-linear interpolation curve of the gene expression time series data points. Though they used data from three different experiments, with very alternate time-scales (time intervals between measurements ranged from half an hour to two months), they managed to combine the data to an interpolated time-series using a cubic interpolation on the log of the expression levels (taking the log prevents negative values). The next step was to derive the entries in an interaction matrix as described in the models above: their model of the genetic network was also a set of linear differential equations in which the change of expression of a gene depends on the weighted inputs of all other genes. These interaction weights were approximated using the least squares fit to the interpolated time series. The result of their approach is an accurate fit of the data points, but they do give some comment on the method. Firstly, the mechanism does not minimize the number of gene interactions: each gene is modeled by a weighted sum of all other genes. Secondly, the simple linear and additive modeling of the genetic interactions can only capture the primary linear components of the system; and finally, because the data that these researchers had at their disposal was so non-uniformly spaced, a larger weight was given to the more widely spaced data points. A point that also needs consideration is to what extent the data is over-fitted in this method. The authors mention that an indication may be obtained by constructing a series of similar models by disturbing the input data within the known standard deviation for each measurement, and by using different non-linear interpolation schemes. Comparison of these models could be found in [4].

In general, biological systems are inherently noisy, and must either minimize or take advantage of stochastic fluctuations to maintain function [5]. That is, accounting for noise and outliers during modeling and simulation of module function will be important for designing biological systems that perform as expected. Hooshangi et al. [6] have used synthetic transcriptional cascades to study noise propagation in genetic networks and examine the effect of intrinsic and extrinsic factors on pheno-

Corresponding Author: Chen-Chia Chuang is with the Department of Electrical Engineering, National Ilan University University, 1, Sec. 1, Shen-Lung Road, I-Lan, Taiwan 260.

E-mail: ccchuang@niu.edu.tw

Manuscript received 11 July. 2006; revised 24 Aug. 2006; accepted 24 Jan. 2008.

typic variations. Besides, Rosenfeld et al. [7] found that the fluctuations due to intrinsic noise of gene expression were much faster than the cell cycle, whereas fluctuations due to extrinsic noise occurred on a longer time-scale and had significant effects on function. Careful design can attenuate noise, as some network architectures are more robust to noise than others. That is, transcriptional cascades are more robust to extrinsic noise than a single gene transcriptional inverter because small fluctuations in the input are not transmitted to the output [6]. In this paper is to apply the new fuzzy neural networks for the modeling of the molecular autoregulatory feedback loop systems with outlier problems that not only has the same performance as the conventional robust fuzzy neural networks [8], but also requires less computing epoch than some existing conventional robust fuzzy neural networks. That means the nonlinear modeling with outliers can be fully reconstructed via the proposed method. And the proposed approach has fast convergence speed than the conventional robust fuzzy neural networks.

This paper is organized as follows. After this introduction section, the proposed structure for the modeling of the molecular autoregulatory feedback loop systems with outliers is presented in Section 2. In Section 3, the annealing robust fuzzy neural networks are proposed. In the section 4, the simulations are conducted and the results verify the applicability of the proposed annealing robust fuzzy neural networks for the molecular autoregulatory feedback loop systems. Finally, Section 5 concludes this paper.

2. The Proposed Structure for Modeling of Molecular Autoregulatory Feedback Loop Systems with Outliers

The Proposed Structure for modeling of the molecular autoregulatory feedback loop systems with outliers is illustrated in Fig. 3. There are two steps of learning in the proposed approach; the initial learning step and parameters learning step. In this paper, a novel approach applies the SVR as the initial learning, that can not only simultaneously identify the parameters in the promise parts and fuzzy singleton consequent parts without knowing the cluster number but also can have robust learning effects against noise and outliers. After an initial structure of fuzzy neural networks are obtained via the SVR, an ARLA learning algorithm is employed to refine the parameters of fuzzy neural networks in the parameters learning phase. That is, the ARLA learning algorithm is applied to adjust the parameters in the promise and fuzzy singleton consequent parts in the parameters learning phase. In view of the above problems, a novel approach, called the ARFNNs, are proposed in this paper

to cope with those problems. That is, the proposed ARFNNs are shown in Fig. 1. In this approach, the initial structure of the ARFNNs is obtained by the SVR approach. Then, an ARLA is applied to adjust the parameters of Gaussian function and the synaptic weights.

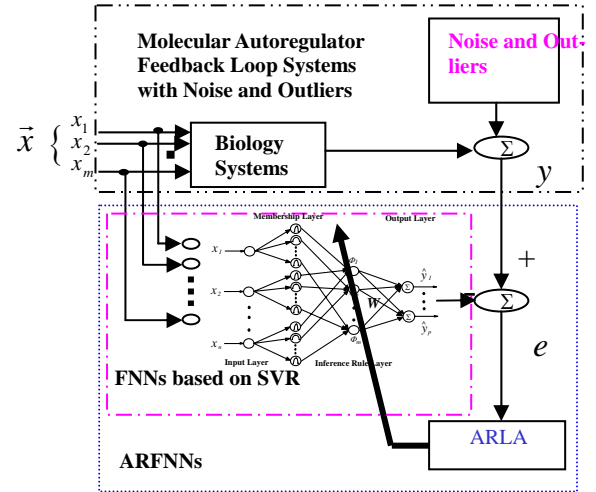


Fig. 1. The proposed structure is shown.

3. The Annealing Robust Fuzzy Neural Networks

3.1. The SVR and the Radial Basis Function Neural Networks

In general, the SVR can be represented as

$$f_s(\vec{x}) = \sum_{i=1}^{SV} (\alpha_i^* - \alpha_i) K(\vec{x}, \vec{x}_i) + b_s, \quad s = 1, \dots, p, \quad (1)$$

where the K is a nonlinear kernel function, the b_s is a constant, the SV is the number of support vector and the p is the number of output of SVR. Note that only some of $(\alpha_i^* - \alpha_i)$'s are not zeros and the corresponding vectors \vec{x}_i 's are called the SVs. From Eq. (1), a support vector regression can be constructed. A SVR approach is to approximate an unknown function from a set of (input, output) samples $\{(\vec{x}_i, y_i), i = 1, \dots, N\}$ without learning. Assuming that a set of basis function $\{g_k(\vec{x}), k = 1, \dots, l\}$ is given, there exists a family of functions that can be expressed as a linear expansion of the basis functions. Then, the problem of function approximation transforms into that finding the parameters of the following basis function linear expansion:

$$f(\vec{x}, \vec{\theta}) = \sum_{i=1}^l \theta_i g_i(\vec{x}) + b, \quad (2)$$

where $\vec{\theta} \in (\theta_1, \dots, \theta_l)$ is parameter vector to be identified and b is a constant. Then, the solution for the problem is to find f that minimizes

$$R(\vec{\theta}) = \frac{1}{n} \sum_{k=1}^n L_\varepsilon(y_k - f(\vec{x}_k, \vec{\theta})), \quad (3)$$

subject to the constraint

$$\|\vec{\theta}\|^2 < C, \quad (4)$$

where $L_\varepsilon(\cdot)$ is the ε -insensitive loss function and defined as

$$L_\varepsilon(e) = \begin{cases} 0, & \text{for } |e| \leq \varepsilon \\ |e| - \varepsilon, & \text{otherwise} \end{cases}, \quad (5)$$

for some previously chosen nonnegative number ε . In Eq. (4), the constraint is imposed to trade off the complexity of the solution. By using the Lagrange multiplier method, it can be shown [9] that the minimization of Eq. (3) leads to the following dual optimization problem, minimize

$$Q(\alpha, \alpha^*) = \varepsilon \sum_{r=1}^N (\alpha_r + \alpha_r^*) - \sum_{r=1}^N y_r (\alpha_r^* - \alpha_r) + \frac{1}{2} \sum_{r,s=1}^N (\alpha_r^* - \alpha_r) (\alpha_s^* - \alpha_s) \left[\sum_{i=1}^l g_i(\vec{x}_r) g_i(\vec{x}_s) \right], \quad (6)$$

subject to the constraint

$$\sum_{r=1}^N \alpha_r^* = \sum_{r=1}^N \alpha_r, \quad 0 < \alpha_r, \alpha_r^* < C, \quad \text{for } r=1, \dots, N, \quad (7)$$

where α , α_u , α_v , α^* , α_u^* and α_v^* are the Lagrange multipliers. In Eq. (6), the inner product of basis functions $g_k(\vec{x})$ is replaced via the kernel function [9-10]

$$K(\vec{x}_r, \vec{x}_s) = \sum_{i=1}^l g_i(\vec{x}_r) g_i(\vec{x}_s). \quad (8)$$

The kernel function determines the smoothness properties of solutions and should reflect a prior knowledge on the data. In the literature, the polynomials, B-spline and Gaussian kernel function often used [9, 11]. Hence, the optimization of Eq. (6) is rewritten as

$$Q(\alpha, \alpha^*) = \varepsilon \sum_{r=1}^N (\alpha_r + \alpha_r^*) - \sum_{r=1}^N y_r (\alpha_r^* - \alpha_r) + \frac{1}{2} \sum_{r,s=1}^N (\alpha_r^* - \alpha_r) (\alpha_s^* - \alpha_s) K(\vec{x}_r, \vec{x}_s) \quad (9)$$

It was shown in [9] that the solution of SVR approach is in the form of the following linear expansion of kernel functions:

$$f(\vec{x}, \alpha, \alpha^*) = \sum_{i=1}^l (\alpha_i^* - \alpha_i) K(\vec{x}, \vec{x}_i) + b. \quad (10)$$

In this paper, the Gaussian function is used as the kernel function. Hence, Eq. (10) can be rewritten as

$$f(\vec{x}) = \sum_{i=1}^{SV} w_i \exp \left\{ \frac{-\|\vec{x} - \vec{x}_i\|^2}{2\sigma^2} \right\} + b, \quad (11)$$

where SV is the number of SVs, $w_i = (\alpha_i^* - \alpha_i) \neq 0$ and \vec{x}_i are SV's. Besides, in Eq. (7), the constant b be taken as that in [11] and is written as:

$$\frac{1}{2} \left\{ \min_i (y_i - \sum_{i=1}^N (\alpha_i^* - \alpha_i) K(\vec{x}_i, \vec{x})) + \max_i (y_i - \sum_{i=1}^N (\alpha_i^* - \alpha_i) K(\vec{x}_i, \vec{x})) \right\}. \quad (12)$$

The bias b in SVM is determined by the KKT condition. However, there are two b after the QP solution. These b values are the maximum error and minimum error. Because, in Eq. (11) is only one b , the b uses the average of maximum error plus minimum error in this paper. Hence, after the procedure of solving a QP problem in Eq. (6) and Eq. (7), the α and l can be obtained. The α can determine the initial weights and the l is the number of rule. Besides, for the kernel function K , it is a non-negative monotonic function. The most popular function of this type is

$$K(\vec{x}, \vec{x}_i) = \exp \left(\frac{-\|\vec{x} - \vec{x}_i\|^2}{2\sigma^2} \right). \quad (13)$$

And let $\bar{f}_s(\vec{x}) = f(\vec{x}) - b$ in Eq. (11), then Eq. (11) is represented as

$$\bar{f}_s(\vec{x}) = \sum_{i=1}^l w_i \exp \frac{-\|\vec{x} - \vec{x}_i\|^2}{2\sigma^2}, \quad (14)$$

where the $w_i = (\alpha_i^* - \alpha_i)$, the $l = SV$ and the l is represented as the number of rule in the fuzzy neural networks with SVR. Besides, the regularization approach determines the approximating function f by minimizing the functional

$$H[f] = \sum_{i=1}^N (y_i - f(\vec{x}_i))^2 + \lambda \|df\|_2, \quad (15)$$

where d is a constraint operator (usually is a differential operator), $\|\cdot\|_2$ is a norm on the function space, and λ is the regularization parameter and is a positive real number. Minimization of the functional H leads to the associated Euler-Lagrange equation,

$$\hat{d} f(\vec{x}_i) = \frac{1}{\lambda} \sum_{i=1}^N (y_i - f(\vec{x}_i)) \delta(\vec{x} - \vec{x}_i), \quad (16)$$

where \hat{d} is the adjoint of the differential operator d , and the right side of Eq. (16) comes from the functional derivative with respect to f on H . Eq. (16) is a partial differential equation, and it is well known that its solution can be written as the integral transformation with a kernel given by the Green's function G of the differential operator $\hat{d}d$. The Green's function G satisfies the following

distributional differential equation

$$\hat{d}d G(\bar{x}; y) = \delta(\bar{x} - y). \quad (17)$$

Because of the delta functions appearing in Eq. (16) the integral transformation becomes a discrete sum. From Eq. (17) and Eq. (16), Eq. (18) can obtain

$$f(\bar{x}) = \frac{1}{\lambda} \sum_{i=1}^N (y_i - f(\bar{x}_i)) G(\bar{x}; \bar{x}_i). \quad (18)$$

Let $\tilde{w}_i = \frac{1}{\lambda} (y_i - f(\bar{x}_i))$, then

$$f(\bar{x}) = \sum_{i=1}^N \tilde{w}_i G(\bar{x}; \bar{x}_i). \quad (19)$$

When Eq. (19) is adopted multivariate Gaussian function to construct the neural networks, it is called a radial basis function network. When the radial basis function network has p outputs, it can be represented as

$f_s(\bar{x}) = \sum_{i=1}^N \tilde{w}_i G(\bar{x}; \bar{x}_i)$, where $s = 1, \dots, p$. Therefore, the radial basis function neural network is one of the SVR.

3.2. The proposed Annealing Robust Fuzzy Neural Networks

In the fuzzy inference systems, fuzzy logic principles are used to combine the fuzz IF-THEN rules in fuzzy rule base into a mapping from fuzzy sets in $U = U_1 \times \dots \times U_n$ to fuzzy sets in V . This class of fuzzy inference system has been proved to be a universal approximator [12-13]; that is, they are capable of approximating any real continuous function on a compact set to an arbitrary accuracy, provided sufficient fuzzy logic rules are available.

The fuzzy rule base comprises the following fuzzy IF-THEN rules:

Rule i : IF x_1 is A_1^i and ... and x_n is A_n^i , THEN \hat{y}_1 is B_1^i and ... and \hat{y}_p is B_p^i . (20)

where A_j^i and B_j^i are fuzzy sets in $U_i \subset R$ and $V \subset R$, respectively, and $\bar{x} \in R^n$ and $y \in V$ are the input and output variables of the fuzzy system, respectively.

Given a pair (\bar{x}_i, y_i) , the final output of the fuzzy system is inferred as follows:

$$\hat{y}_s = \frac{\sum_{i=1}^m \beta_i \left(\prod_{j=1}^n \mu_{A_j^i}(x_j) \right)}{\sum_{i=1}^m \prod_{j=1}^n \mu_{A_j^i}(x_j)}, \quad s = 1, \dots, p. \quad (21)$$

Let $\psi_i(\bar{x})$ be functions of input $x = (x_1, x_2, \dots, x_n)$:

$$\psi_i(\bar{x}) = \frac{\prod_{j=1}^n \mu_{A_j^i}(x_j)}{\sum_{i=1}^m \prod_{j=1}^n \mu_{A_j^i}(x_j)}. \quad (22)$$

The fuzzy systems can define the class of fuzzy inference system as a family of function $\hat{y}_s : R^n \rightarrow R$ in the form of

$$\hat{y}_s(\bar{x}) = \sum_{i=1}^m \beta_i \psi_i, \quad (23)$$

for $\bar{x} \in R^n$ and $\beta_j \in R$, and m is a finite number of fuzzy rule.

In this paper, the proposed fuzzy neural networks consist of m rules in the form of Eq. (20) and that membership functions are

$$\mu_{A_j^i}(x_j) = \exp \left(-\frac{(x_j - \bar{x}_j^i)^2}{2\sigma_j^i} \right), \quad (24)$$

where \bar{x}_j^i and σ_j^i are the constant parameters, $j = 1, 2, \dots, n$ and $i = 1, 2, \dots, m$. That is, the membership functions are all Gaussian function in this paper.

Let $\sigma_1^i = \sigma_2^i = \dots = \sigma_n^i = \sigma_i$, then

$$\prod_{j=1}^n \exp \left(-\frac{(x_j - \bar{x}_j^i)^2}{2\sigma_j^i} \right) = \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right), \quad (25)$$

where $\bar{x}_i = (\bar{x}_1^i, \bar{x}_2^i, \dots, \bar{x}_n^i)$. Substituting Eq. (25) into Eq. (22), then

$$\hat{y}_s = \frac{\sum_{i=1}^m \beta_i \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right)}{\sum_{i=1}^m \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right)}. \quad (26)$$

When $\Phi_i = \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right)$ and $W_i = \frac{\beta_i}{\sum_{i=1}^m \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right)}$, the Eq.

(26) can be rewrite as

$$\hat{y}_s(\bar{x}) = \sum_{i=1}^m W_i \Phi_i = \sum_{i=1}^m W_i \exp \left(-\frac{\|\bar{x} - \bar{x}_i\|^2}{2\sigma_i^2} \right). \quad (27)$$

It is shown that fuzzy neural networks in Eq. (27) can be represented as the functional link networks that are based on Gaussian function. Besides, the proposed fuzzy neural networks are shown in Fig. 2, which are comprised by the input, membership, rule inference system, and output layer.

From Eq. (27) and Eq. (14), the $W_i, m, 2\sigma_i^2$ are equal to w_i, l and, $2\sigma^2$ respectively. That is, the initial

weights W_j and the number of rule m of the proposed fuzzy neural networks in Fig. 2 can be determined via the SVR. Sometime the SVR can obtain a better performance in a function approximation. However, there is no good method to determine the parameter of SVR. That is, some cases can obtain a good approximated result, but the most cases the error between the result of SVR and desired function have many sampling point not less equal then ε_i , where ε_i is the error in the sampling point. That is, the most cases only use SVR to obtain good performance that cannot always satisfy all the sampling point. Besides, the SVR with loss function can do rough approximation without iteration. However, it is difficult to determine the hyper-parameter in the SVR. Hence, the SVR is used to do rough approximation and determine structure for the fuzzy neural networks. Because the σ_i in the SVR is fixed for all kernel function, the proposed approach needs to assume $\sigma_1^i = \sigma_2^i = \dots = \sigma_n^i = \sigma_i$ in the rough approximation. That is, this assumption is only needed in the initial structure. In the learning procedure the σ_i can be updated. Hence, the proposed method adds to the learning algorithm in proposed fuzzy neural network to overcome these drawbacks with SVR approach. For the proposed network, it is easy to use the recursive least square method with forgetting factor or gradient-descent kind as finally learning algorithm.

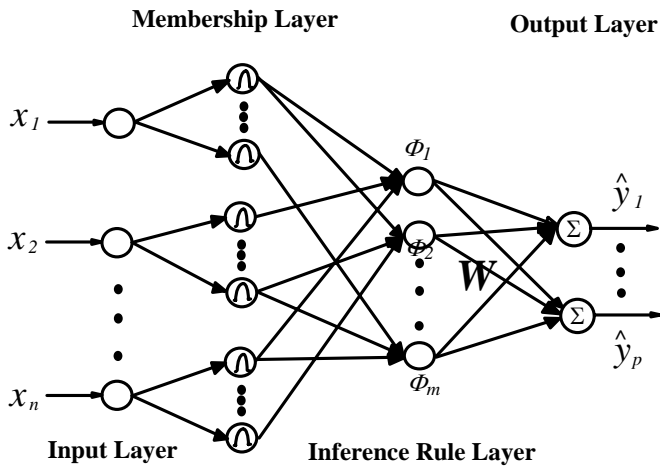


Fig. 2. The Structure of proposed fuzzy neural networks are shown.

3.3 The annealing robust learning algorithm for an ARFNNs

In the ARFNNs, an ARLA is proposed as learning algorithm in this paper. An important feature of the ARLA that adopt the annealing concept into the cost function of robust back-propagation learning algorithm

is proposed in [14]. Hence, the ARLA can be overcome the existed problems in the robust back-propagation learning algorithm. A cost function for an ARLA is defined here:

$$E_{ARFNNs}(t) = \frac{1}{N} \sum_{i=1}^N \rho[e_i(t); \beta(t)], \quad (28)$$

$$\text{where } e_i(t) = y_i - f(\bar{x}_i) = y_i - \sum_{j=1}^L w_j \exp\left\{-\frac{\|\bar{x}_i - m_j\|^2}{2\sigma_j^2}\right\},$$

the t is the epoch number, the $e_i(t)$ is the error between the i -th desired output and the i -th output of the ARFNNs at epoch t , the $\beta(t)$ is a deterministic annealing schedule acting like the cut-off points and the $\rho(\cdot)$ is a logistic loss function and defined as

$$\rho[e_i; \beta] = \frac{\beta}{2} \ln \left[1 + \left(\frac{e_i^2}{\beta} \right) \right]. \quad (29)$$

Based on the gradient-descent kind of learning algorithms, the synaptic weights w_j , the centers m_j and the width σ_j of Gaussian function are updated as

$$\Delta w_j = -\eta \frac{\partial E_{ARFNNs}}{\partial w_j} = -\eta \sum_{i=1}^N \varphi(e_i; \beta) \frac{\partial e_i}{\partial w_j}, \quad (30)$$

$$\Delta m_j = -\eta \frac{\partial E_{ARFNNs}}{\partial m_j} = -\eta \sum_{i=1}^N \varphi(e_i; \beta) \frac{\partial e_i}{\partial m_j}, \quad (31)$$

$$\Delta \sigma_j = -\eta \frac{\partial E_{ARFNNs}}{\partial \sigma_j} = -\eta \sum_{i=1}^N \varphi(e_i; \beta) \frac{\partial e_i}{\partial \sigma_j}, \quad (32)$$

$$\varphi(e_i; \beta) = \frac{\partial \rho(e_i; \beta)}{\partial e_i} = \frac{e_i}{1 + \frac{e_i^2}{\beta}}, \quad (33)$$

where the η is a learning constant, the $\varphi(\cdot)$ is usually called the influence function. When outliers exist, they have great impact on the approximated results. Such an impact can be understood through the analysis of the influence function. In the ARLA, the annealing schedule is using $\beta(t) = k/t$ for any t epoch, where the k is constants. When the decay is too quick, the approximation of the majority may not have enough time to converge and the training data may mostly be degraded. If the decay is too slow, the robust learning algorithm may not be in time to discriminate against those outliers before overfitting occurs. However, the suitable annealing schedule k/t has been found experimentally to achieve the best performance in the ARLA [14].

4. Simulation Results

The simulations were conducted in the *Matlab* environ-

ment. The root mean square error (RMSE) of the testing data is used to measure the performance of the learned networks (generalization capability). The RMSE is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (\hat{y}_i - y_i)^2}{N}}, \quad (34)$$

where y_i is the desire value at x_i and \hat{y}_i is the output of the ARFNNs. Consider the modeled molecular autoregulatory feedback loop systems [15] is

$$\begin{aligned} \frac{dx_M}{dt} &= \frac{r_M}{k + x_P^n} - q_M x_M \\ \frac{dx_P}{dt} &= r_p x_M^m (t - \tau) - q_p x_P \end{aligned}, \quad (35)$$

where where x_M is the relative mRNA concentration, x_P is the relative concentration of protein that effectively provides negative feedback on transcription, r_M is the rate of transcriptional activation, r_p is the protein synthesis rate constant, q_M is the mRNA breakdown rate constant and q_p is the protein breakdown rate constant, n is the Hill coefficient, τ is the delay on translation and m is an exponent imposing nonlinearity on the production of effective protein. The parameter in Eq. (35) are $r_M = 2h^{-1}$, $q_M = 0.21h^{-1}$, $r_p = 2h^{-1}$, $q_p = 0.21h^{-1}$, $\tau = 4h$, $m=3$, $n=7$, and $k=1$. Assume the number of sampling data is 300 the membership function is Gaussian function and the learning rate equals to 0.95. Besides, the molecular autoregulatory feedback loop systems add the noise with four outliers. Sum square error is 0.01 in the proposed structure. After the SVR, then the total number of rule is 52 and initial weights are determined in the proposed ARFNNs. The training data sets are shown in Fig. 3. Besides, the molecular autoregulatory feedback loop systems with noise and outliers and initial with SVR are shown in Fig. 4. The final results for for the mRNA abundance are shown in Fig. 5. From Fig. 5, the finally results with proposed approach and training data set overlap together and robust against outliers. Besides, the learning epoch is not greater than 15 epochs to obtain Fig. 5. The RMSE versus epoch is shown in Fig. 6. On the contrary, when this function uses conventional robust fuzzy neural networks with random weights to get Fig. 5, it needs greater than 3000 epochs. From these results, the proposed approach could indeed improve the learning performance.

5. Conclusions

In this paper, the ARFNNs are applied to improve the conventional robust fuzzy neural networks for modeling of the molecular autoregulatory feedback loop systems with outliers. Firstly, the SVR approach is proposed to determine the initial structure of the ARFNNs. The number of hidden nodes, the initial parameters and the initial weights of the ARFNNs are easy obtained via the SVR approach. Secondly, the results of the SVR are used as initial structure in the ARFNNs. At the same time, an ARLA is used as the learning algorithm for the ARFNNs, and applied to adjust the parameters as well as weights of the ARFNNs. Hence, when an initial structure of the ARFNNs is determined by a SVR approach, the ARFNNs with the ARLA have fast convergence speed and robust against outliers for the modeling of the molecular autoregulatory feedback loop systems.

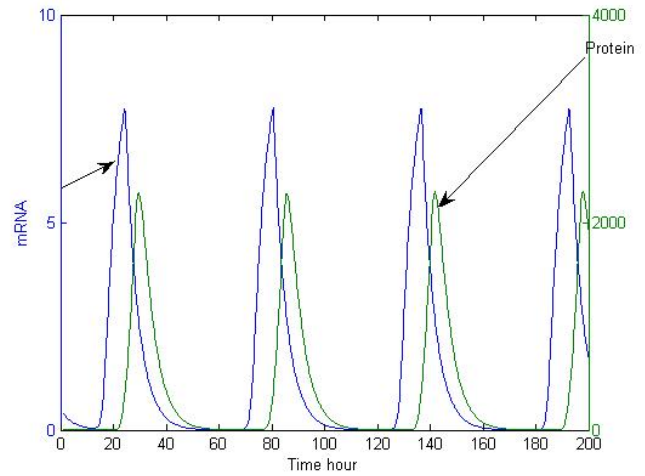


Fig. 3. The simulation training data sets under $r_M = 2h^{-1}$, $q_M = 0.21h^{-1}$, $r_p = 2h^{-1}$, $q_p = 0.21h^{-1}$, $\tau = 4h$, $m=3$, $n=7$, and $k=1$ are shown.

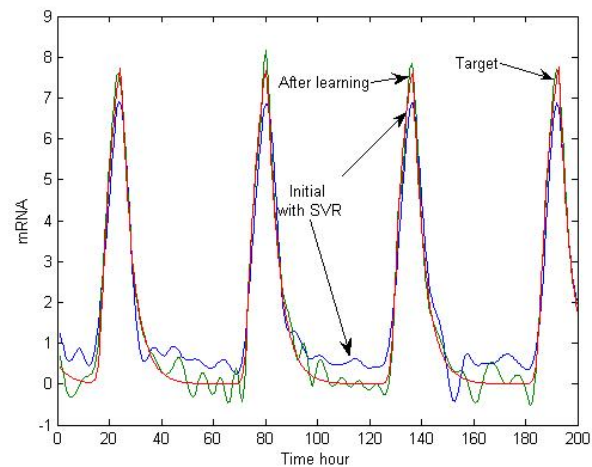


Fig. 4. The result with proposed structure after 100 epochs for mRNA abundance are shown.

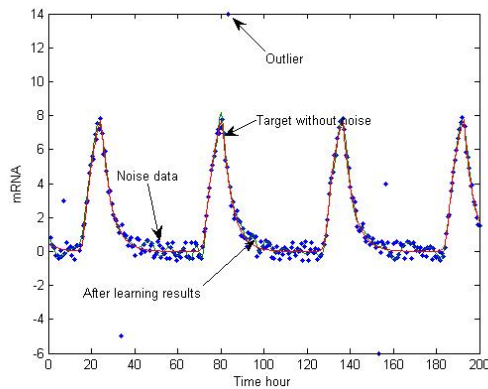


Fig. 5. The finally result with proposed structure and training data set for mRNA abundance.

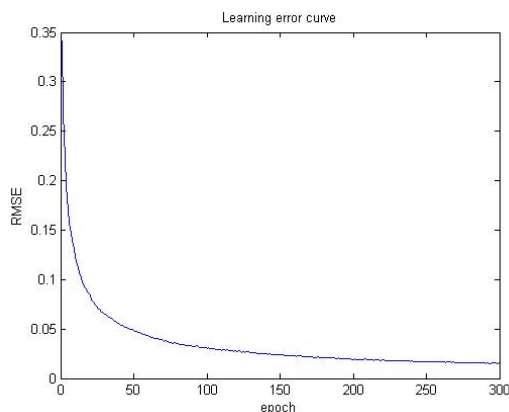


Fig. 6. The RMSE versus epoch is shown.

Acknowledgment

The author would like to thank Professor Tsu-Tian Lee and Professor Jin-Tsong Jeng for the discussions.

References

- [1] H. H. McAdams and A. Arkin, "Stochastic mechanisms in gene expression." *Proc Natl. Acad. Sci. USA*, 94:814–819, 1997.
- [2] H. Schmidt and E.W. Jacobsen, "Identifying feedback mechanisms behind complex cell behavior," *IEEE Control Systems Magazine*, pp. 91-102, 2004.
- [3] P. D'Haeseleer, X. Wen, S. Fuhrman, and R. Somogyi, "Linear modeling of mRNA expression levels during CNS development and injury," *Pacific Symp. Biocomputing*, pp. 41–52, 1999.
- [4] H. D. Jong, "Modeling and simulation of genetic regulatory systems: a literature review," *Journal of Computational Biology*, vol. 9, pp. 67-103, 2002.
- [5] E. Andrianantoandro, S. Basu, D. K. Karig and R. Weiss, "Synthetic biology: new engineering rules for an emerging discipline," *Molecular Systems Biology*, msb4100073-E 1-14, 2006.

- [6] S. Hooshangi, S. Thiberge, R. Weiss, "Ultrasensitivity and noise propagation in a synthetic transcriptional cascade," *Proc. Natl. Acad. Sci. USA*, pp. 3581-3586, 2005.
- [7] N. Rosenfeld, J.W. Young, U. Alon, P.S. Swain and M.B. Elowitz, "Gene regulation at the signal-cell level," *Science*, vol. 307, pp. 1962-1965, 2005.
- [8] J. T. Jeng and C. C. Chuang, "Selection of Initial Structures with Support Vector Regression for Fuzzy Neural Networks," *International Journal of Fuzzy Systems*, vol. 6, no. 2, pp. 63-70, 2004.
- [9] V. Vapnik, S. Golowich and A. J. Smola, "Support vector method for function approximation, regression estimation, and signal processing," *Neural Information Processing Systems*, 1997.
- [10] A. J. Smola and B. Schölkopf, "From regularization operators to support vector kernels", *Neural Information Processing Systems*, vol. 9. MIT Press, 1997.
- [11] J. T. Jeng, "Hybrid Approach of Selecting Hyper-Parameters of Support Vector Machine for Regression," *IEEE Trans. on Systems, Man, and Cybernetics Part B: Cybernetics*, vol. 36, no. 3, pp. 699-709, 2006.
- [12] J. T. Jeng, "Annealing Robust Fuzzy Basis Function for Modeling with Noise and Outliers," *International Journal of Computer Applications in Technology*, vol. 27, nos. 2/3, pp. 166-172, 2006.
- [13] L. X. Wang, *Adaptive Fuzzy Systems and Control: Design and Stability Analysis*, Prentice Hall, 1994.
- [14] C. C. Chuang, S. F. Su and C. C. Hsiao, "The Annealing Robust Backpropagation (BP) Learning Algorithm," *IEEE Trans. Neural Networks*, vol. 11, no. 5, pp. 1067-1077, 2000.
- [15] T.O. Scheper, D. Klinkenberg, J. V. Pelt and C. Pennartz, "A model of molecular circadian clocks: multiple mechanisms for phase shifting and a requirement for strong nonlinear interactions," *Journal of Biological Rhythms*, vol. 14, pp. 213-220, 1999.



Chen-Chia Chuang received the B.S. and M.S. degrees in Electrical Engineering from National Taiwan Institute of Technology, Taipei, Taiwan, in 1991 and 1993, respectively. He received Ph.D. degree in the Department Electrical Engineering at the National Taiwan University of Science and Technology, Taipei, Taiwan in 2000. He is currently an Associate Professor with the Department of Electrical Engineering, National Ilan University, I-Lan, Taiwan. His current research interests are neural networks, data analysis, neural fuzzy systems, support vector regression and signal processing.