Levenberg-Marquardt Method for ANFIS Learning

Jyh-Shing Roger Jang

CS Department
Tsing Hua University
Hsinchu, Taiwan

Eiji Mizutani

ME Department
Univ. of California at Berkeley
Berkeley, CA 94720

Abstract

We present the results of applying the Levenberg-Marquardt method, a popular nonlinear least-squares method, to the ANFIS (Adaptive Neuro Fuzzy Inference System) architecture [2] proposed earlier. Through empirical studies, we discuss the strengths and weaknesses of using such an efficient nonlinear regression techniques for neuro-fuzzy modeling, and explain the trade-offs between mapping precision and MF interpretability.

1 Introduction

The Levenberg-Marquardt (LM) method is an effective nonlinear least-squares approach to nonlinear regression problems, including neural networks and fuzzy modeling. By changing the value of a control parameter λ, the Levenberg-Marquardt method can vary smoothly between the stable but slow gradient descent (or steepest descent) method and the greedy but less stable Gauss-Newton method.

In this paper, we present empirical studies of applying the LM method to the ANFIS (Adaptive Neuro Fuzzy Inference System) architecture [1, 2, 4] proposed earlier. We discuss the strengths and weaknesses of using such an advanced nonlinear regression techniques for neuro-fuzzy modeling, compare the results to those of the previously proposed hybrid learning method, and explain the trade-offs between mapping precision and membership function (MF) interpretability.

This paper is organized into five sections. In the next section, the basics of ANFIS are introduced. Section 3 explains the rationale behind the Levenberg-Marquardt method. Simulation results are demonstrated in section 4. Section 5 gives concluding remarks.

2 ANFIS

This section introduces the basics of ANFIS network architecture and its hybrid learning rule. A detailed coverage of ANFIS can be found in [1, 2, 4].

The **Sugeno fuzzy model** was proposed by Takagi, Sugeno, and Kang [14, 13] in an effort to formalize a systematic approach to generating fuzzy rules from an input-output data set. A typical fuzzy rule in a Sugeno fuzzy model has the format

If x is A and y is B then $z = f(x,y)$.

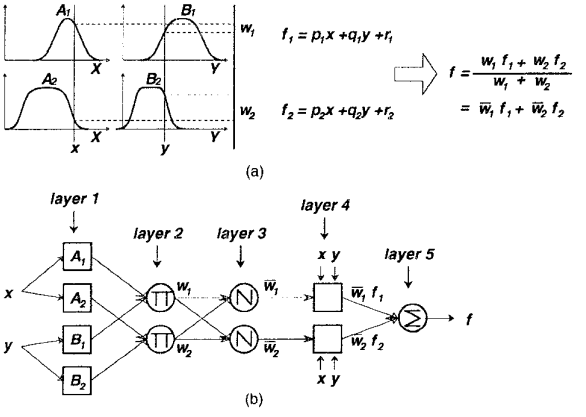

Figure 1: *(a) First-order Sugeno fuzzy model; (b) corresponding ANFIS architecture.*

where A and B are fuzzy sets in the antecedent; $z = f(x,y)$ is a crisp function in the consequent. Usually $f(x,y)$ is a polynomial in the input variables x and y, but it can be any other functions that can appropriately describe the output of the system within the fuzzy region specified by the antecedent of the rule. When $f(x,y)$ is a first-order polynomial, we have the **first-order** Sugeno fuzzy model, which was originally proposed in [14, 13]. When f is a constant, we then have the **zero-order** Sugeno fuzzy model, which can be viewed either as a special case of the Mamdani fuzzy inference system [7] where each rule's consequent is specified by a fuzzy singleton, or a special case of Tsukamoto's fuzzy model [15] where each rule's consequent is specified by a membership function of a step function centered at the constant. Moreover, a zero-order Sugeno fuzzy model is functionally equivalent to a radial basis function network under certain minor constraints [3].

Consider a first-order Sugeno fuzzy inference system which contains two rules:

Rule 1: If X is A_1 and Y is B_1, then
$f_1 = p_1 x + q_1 y + r_1$,
Rule 2: If X is A_2 and Y is B_2, then
$f_2 = p_2 x + q_2 y + r_2$.

Figure 1 (a) illustrates graphically the fuzzy reasoning

0-7803-3225-3-6/96 $5.00 © 1996 IEEE

mechanism to derive an output f from a given input vector $[x, y]$. The **firing strengths** w_1 and w_2 are usually obtained as the product of the membership grades in the premise part, and the output f is the weighted average of each rule's output.

To facilitate the learning (or adaptation) of the Sugeno fuzzy model, it is convenient to put the fuzzy model into the framework of adaptive networks that can compute gradient vectors systematically. The resultant network architecture, called **ANFIS** (Adaptive Neuro-Fuzzy Inference System), is shown in Figure 1 (b), where node within the same layer perform functions of the same type, as detailed below. (Note that O_i^j denotes the output of the i-th node in j-th layer.)

Layer 1 Each node in this layer generates a membership grades of a linguistic label. For instance, the node function of the i-th node may be a generalized bell membership function:

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1 + \left|\frac{x - c_i}{a_i}\right|^{2b_i}}, \tag{1}$$

where x is the input to node i; A_i is the linguistic label (*small* , *large*, etc.) associated with this node; and $\{a_i, b_i, c_i\}$ is the parameter set that changes the shapes of the membership function. Parameters in this layer are referred to as the **premise parameters**.

Layer 2 Each node in this layer calculates the firing strength of a rule via multiplication:

$$O_i^2 = w_i = \mu_{A_i}(x)\mu_{B_i}(y), \ i = 1, 2. \tag{2}$$

Layer 3 Node i in this layer calculates the ratio of the i-th rule's firing strength to the total of all firing strengths:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2}, \ i = 1, 2. \tag{3}$$

Layer 4 Node i in this layer compute the contribution of i-th rule toward the overall output, with the following node function:

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i), \tag{4}$$

where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer are referred to as the **consequent parameters**.

Layer 5 The single node in this layer computes the overall output as the summation of contribution from each rule:

$$O_1^5 = \textit{overall output} = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

The constructed adaptive network in Figure 1(b) is functionally equivalent to a fuzzy inference system in Figure 1(a). The basic learning rule of ANFIS is the backpropagation gradient descent [16], which calculates error signals (the derivative of the squared error with respect to each node's output) recursively from the output layer backward to the input nodes. This learning rule is exactly the same as the backpropagation learning rule used in the common feedforward neural networks [11].

From the ANFIS architecture in Figure 1, it is observed that given the values of premise parameters, the overall output f can be expressed as a linear combinations of the consequent parameters:

$$\begin{aligned} f &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 \\ &\quad + (\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2. \end{aligned} \tag{6}$$

Based on this observation, we have proposed a hybrid learning algorithm [1, 2] which combines the gradient descent and the least-squares method for an effective search of optimal parameters; both on-line and off-line learning paradigms were developed and reported in [2]. Following the concept of ANFIS, we have also proposed the CANFIS (Coactive ANFIS) architecture [9, 5] that has multiple outputs and nonlinear output equations. Details of ANFIS/CANFIS and their applications can be found in [5].

3 Levenberg-Marquardt Method

ANFIS is a network architecture that allows systematic calculation of gradient vectors (derivatives of output error with respect to modifiable parameters), so we are not limited to the backpropagation or hybrid learning method only. In fact, we can apply any gradient-based techniques in nonlinear regression and optimization, such as the Gauss-Newton method, the Levenberg-Marquardt method [6, 8], and the extended Kalman filter algorithm [12, 10]. This section presents the Levenberg-Marquardt (LM) method.

A nonlinear neuro-fuzzy model can be generally expressed as

$$y = f(\mathbf{x}, \boldsymbol{\theta}), \tag{7}$$

where $\mathbf{x}$ is the input vector, $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]$ is the parameter vector and y is the model's (scalar) output. (The extension to multiple-output systems is straightforward.) Given a set of training data $\{(\mathbf{x}_p; t_p), p = 1, \ldots, m\}$, a squared error measure takes the form

$$E(\boldsymbol{\theta}) = \sum_{p=1}^{m} [t_p - f(\mathbf{x}_p, \boldsymbol{\theta})]^2, \tag{8}$$

which is the objective function we want to minimize. Before introducing the Levenberg-Marquardt method for minimizing Equation (8), we shall review the closely related Gauss-Newton method.

The **Gauss-Newton method**, also known as the **linearization method**, uses a Taylor series expansion to obtain a linear model that approximates the original nonlinear model and then employs the ordinary least-squares method to estimate the parameters.

Specifically, let the current parameters be denoted by $\boldsymbol{\theta}_{now}$; then we can expand the nonlinear model in Equation (7) in a Taylor series around $\boldsymbol{\theta} = \boldsymbol{\theta}_{now}$ and retain only the linear terms:

$$y = f(\mathbf{x}, \boldsymbol{\theta}_{now}) + \sum_{i=1}^{n} \left(\frac{\partial f(\mathbf{x}, \boldsymbol{\theta})}{\partial \theta_i}\bigg|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{now}} \right) (\theta_i - \theta_{now,i}). \quad (9)$$

Inspection of the above equation reveals that the translated output $y - f(\mathbf{x}, \boldsymbol{\theta}_{now})$ is a linear function of the translated parameters $\theta_i - \theta_{now,i}$. We can therefore obtain a better estimator $\boldsymbol{\theta}_{next}$ by means of the well-known pseudo-inverse formula:

$$\begin{aligned} \boldsymbol{\theta}_{next} &= \boldsymbol{\theta}_{now} + (A^T A)^{-1} A^T \Delta \mathbf{y} \\ &= \boldsymbol{\theta}_{now} + \Delta\boldsymbol{\theta}, \end{aligned} \quad (10)$$

where $\Delta\mathbf{y}$ is the error vector of which the pth element is equal to $t_p - f(\mathbf{x}_p, \boldsymbol{\theta}_{now})$, $\Delta\boldsymbol{\theta}$ is $(A^T A)^{-1} A^T \Delta\mathbf{y}$, and the element at row p and column j of matrix A is $\frac{\partial f(\mathbf{x}_p, \boldsymbol{\theta})}{\partial \theta_j}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_{now}}$.

A potential problem with the Gauss-Newton method is that $(A^T A)^{-1}$ might not always exist, rendering this method practically unusable. Such a situation is handled by **Levenberg-Marquardt procedure**, which defines $\Delta\boldsymbol{\theta}$ as

$$\Delta\boldsymbol{\theta} = (A^T A + \lambda I)^{-1} A^T \Delta\mathbf{y},$$

where I is the identity matrix and λ is usually a small positive constant. Depending on the magnitude of λ, the method transits smoothly between two extremes: the Gauss-Newton method ($\lambda \to 0$) and the gradient descent method ($\lambda \to \infty$). Usually the Gauss-Newton method is more efficient but less stable; the gradient descent method is more stable but less efficient. By properly setting the value of λ, the Levenberg-Marquardt method can be efficient as well as stable. More details can be found in [8].

4 Simulation Results

This section presents the simulation results of ANFIS training using both the hybrid learning method [2] and the Levenberg-Marquardt method. For simplicity, we use a single-input single-output training data set defined by the equation:

$$y = 0.6\sin(\pi x) + 0.3\sin(3\pi x) + 0.1\sin(5\pi x),$$

where x is a set of 51 linearly spaced data points between -1 and 1. Since we only want to compare the mapping precision of two training methods, there was not test data involved in our simulation.

By using the hybrid learning method, an ANFIS system with 3 rules can match the training data satisfactorily after 100 epochs, as shown in Figure 2 (a). The upper sub-plots in (a) illustrate the membership functions before and after training; the lower plot demonstrate the training data points and the resulting ANFIS input/output curve. Since the MF parameters are tuned by the gradient descent, the final MFs do not vary too much from the initial MFs. The final RMSE (root-mean-squared error) is 0.1184.

As a comparison, we apply the Levenberg-Marquardt method to the same three-rule ANFIS. Figure 2 (b) is the results after a similar amount of computation time. The final RMSE is 0.0565, which is better than that achieved by the hybrid learning. However, the final MFs vary a lot from their initial settings and it is hard to assign appropriate linguistic labels to these MFs. In particular, the MFs shrank during training, which leaves some input domain not covered by MFs of sufficient heights. This exemplifies the **dilemma between precision and interpretability**: the LM method is more effective and it can achieve a lower RMSE and thus higher mapping precision, but it does not conserve good properties of initial MFs such as moderate overlaps and approximate orthogonality.

Similar observations can be made when a four-rule ANFIS is used. Figure 3 (a) is the results of using the hybrid learning method; the final RMSE is 0.0079. Figure 3 (b) is the counterpart of using the Levenberg-Marquardt method; the final RMSE is smaller (0.0015) but the final MFs do not lend themselves to good linguistic interpretation.

5 Conclusions and Future Work

We have discussed the Levenberg-Marquardt method for ANFIS training. The Levenberg-Marquardt method is an efficient approach to nonlinear least-squares problems. When applied to ANFIS training, the LM method was able to reduce the root mean squared error further than the previously proposed hybrid learning method. Although the LM method can achieve a better mapping precision, it also evolves the MFs to an extent such that the linguistic interpretability of the final MFs becomes quite weak. We refer to the situation as the **dilemma between precision and interpretability** [5]. The hybrid learning method achieved a lower precision, and the resultant MFs are usually interpretable. On the other hand, the LM method attained a higher precision, but it also generated not-so-interpretable MFs.

Adaptive neuro-fuzzy models like ANFIS/CANFIS transit smoothly between the two ends of neuro-fuzzy spectrum: a linguistically understandable fuzzy inference system and a black-box neural network (in particular, backpropagation multilayer perceptron). This is better explained by the interpretability-precision plane shown in Figure 4. Ideally we would like the training of a neuro-fuzzy model takes the vertical route to the top, such that the mapping precision is improved while the interpretability is maintained. In practice, however, the training usually takes the diagonal route to improve mapping precision with deteriorating interpretability. The horizontal route demonstrates the neuro-fuzzy spectrum ranging between two extremes: an understandable fuzzy inference system and a black-box neural network.

If linguistic interpretability is not a concern, then we are entitled to choose the most efficient learning

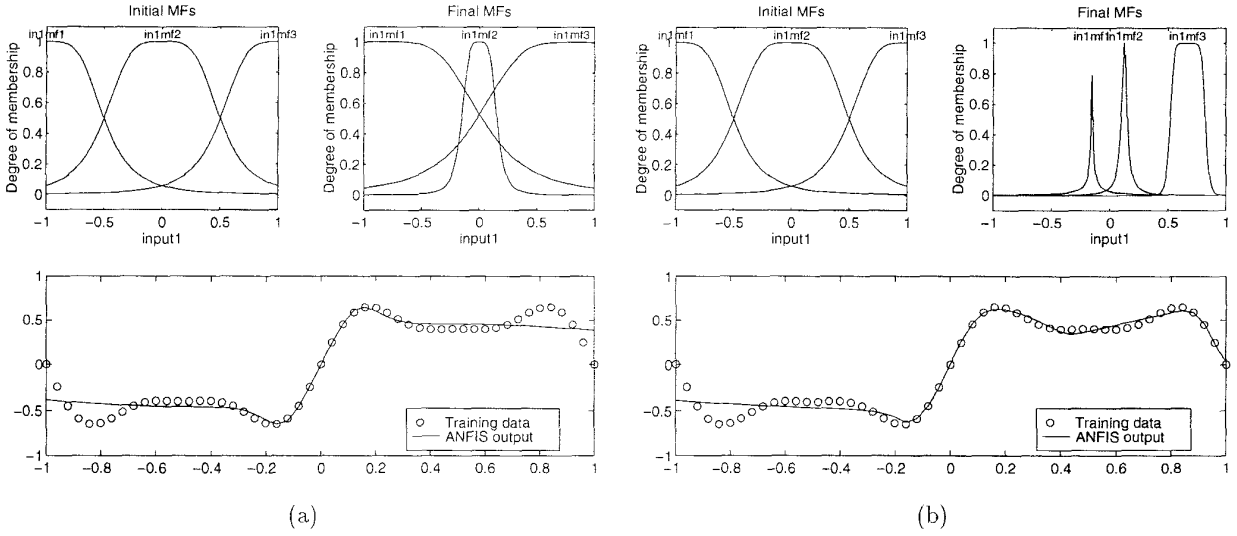

Figure 2: *Three-rule ANFIS trained by (a) hybrid learning; the final RMSE is 0.1184; (b) the LM method; the final RMSE is 0.0565.*

algorithm in the literature. However, if linguistic interpretability is a concern, then we need to be careful about how to update MF parameters. The hybrid learning rule generally gives interpretable results, but this is not always guaranteed. (Even if we use the simple backpropagation gradient descent, we cannot guarantee the resultant interpretability, as discussed in [9].)

Our future work should involve considerations on achieving better interpretability by putting proper constraints on neighboring MFs, or by reformulating the error measure to be minimized. One possible way of reformulating the error measure is to incorporate a term similar to Shannon's information entropy, as suggested in [2].

References

[1] J.-S. Roger Jang. Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In *Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 762–767, July 1991.

[2] J.-S. Roger Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. on Systems, Man, and Cybernetics*, 23(03):665–685, May 1993.

[3] J.-S. Roger Jang and C.-T. Sun. Functional equivalence between radial basis function networks and fuzzy inference systems. *IEEE Trans. on Neural Networks*, 4(1):156–159, January 1993.

[4] J.-S. Roger Jang and C.-T. Sun. Neuro-fuzzy modeling and control. *The Proceedings of the IEEE*. 83(3):378–406, March 1995.

[5] J.-S. Roger Jang, C.-T. Sun, and E. Mizutani. Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence, 1996. To be published by Prentice Hall.

[6] K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Apl. Math.*, 2:164–168, 1944.

[7] E. H. Mamdani and S. Assilian. An experiment in linguistic synthesis with a fuzzy logic controller. *International Journal of Man-Machine Studies*, 7(1):1–13, 1975.

[8] Donald W. Marquardt. An algorithm for least squares estimation of nonlinear parameters. *Journal of the Society of Industrial and Applied Mathematics*, 11:431–441, 1963.

[9] E. Mizutani and J-S. Roger Jang. Coactive neural fuzzy modelings. In *Proc. of the International Conference on Neural Networks*, pages 760–765, November 1995.

[10] D. W. Ruck, S. K. Rogers, M. Kabrisky, P. S. Maybeck, and M. E. Oxley. Comparative analysis of backpropagation and the extended Kalman filter for training multilayer perceptrons. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(6):686–691, 1992.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1*, chapter 8, pages 318–362. The MIT Press, 1986.

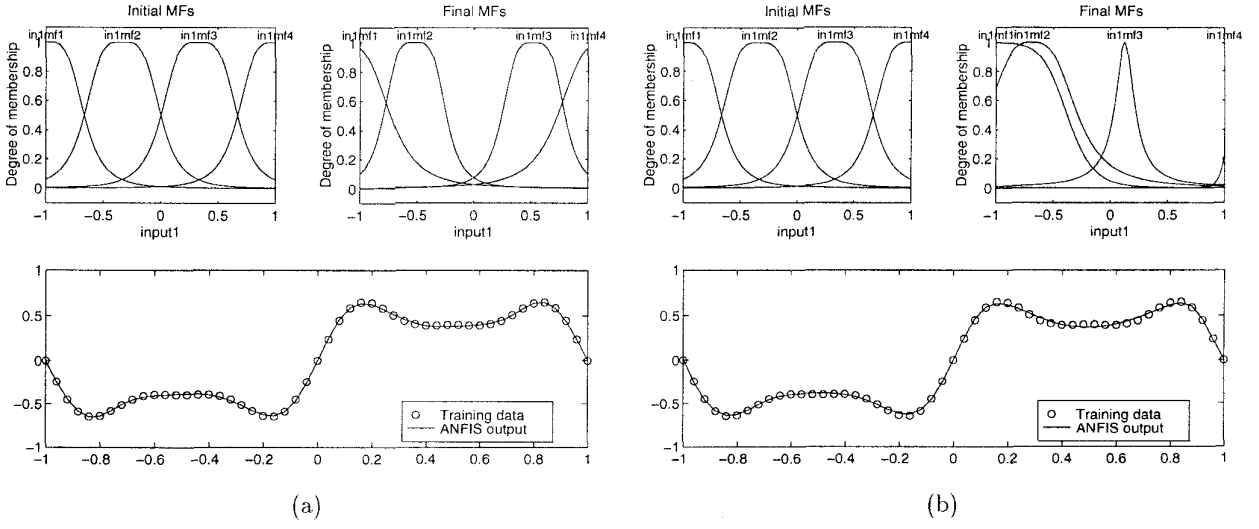

Figure 3: *Four-rule ANFIS trained by (a) hybrid learning; the final RMSE is 0.0079; (b) the LM method; the final RMSE is 0.0015.*

[12] S. Singhal and L. Wu. Training multilayer perceptrons with the extended kalman algorithm. In David S. Touretzky, editor, *Advances in neural information processing systems I*, pages 133–140. Morgan Kaufmann, 1989.

[13] M. Sugeno and G. T. Kang. Structure identification of fuzzy model. *Fuzzy Sets and Systems*, 28:15–33, 1988.

[14] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. on Systems, Man, and Cybernetics*, 15:116–132, 1985.

[15] Y. Tsukamoto. An approach to fuzzy reasoning method. In Madan M. Gupta, Rammohan K. Ragade, and Ronald R. Yager, editors, *Advances in Fuzzy Set Theory and Applications*, pages 137–149. North-Holland, Amsterdam, 1979.

[16] P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences.* PhD thesis, Harvard University, 1974.

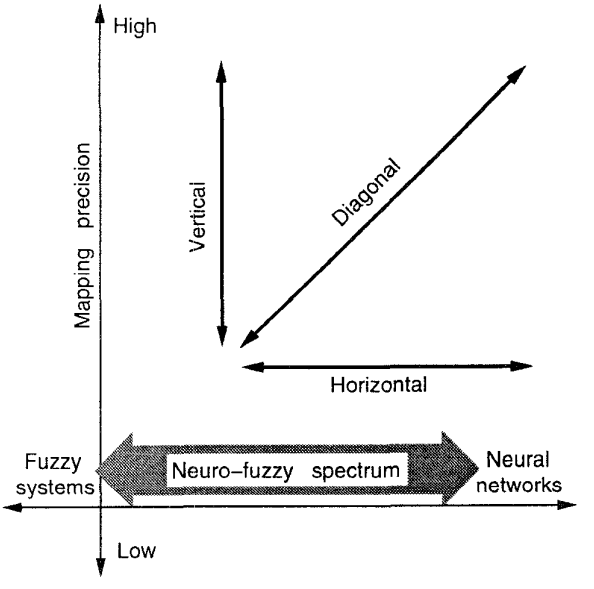

Figure 4: *The plane of neuro-fuzzy spectrum (horizontal route) and input-output mapping precision. Ideally we would like the training of a neuro-fuzzy model to take the vertical route to the top, but usually it takes the diagonal route to improve mapping precision with deteriorating interpretability.*