

Successive Projection Method for Fast Learning Algorithm of Neurofuzzy GMDH

Takashi OHTANI, Hidetomo ICHIHASHI, Kazunori NAGASAKA and Tetsuya MIYOSHI
Department of Industrial Engineering, College of Engineering, Osaka Prefecture University, 1-1 Gakuencho,
Sakai, Osaka 593, JAPAN

e-mail:ichi@center.osakafu-u.ac.jp, TEL:0722-52-1161, FAX:0722-59-3340

Key words: *RBF networks, LMS learning, Successive projection method*

Abstract Neurofuzzy GMDH, whose partial descriptions are represented by Radial Basis Function networks, has been proposed. In this paper, a generalized successive projection method for fast learning algorithm of Neurofuzzy GMDH is developed and compared with the Least Mean Square rule. (1) For the multilayered models, an algorithm is derived as the solution of an optimization problem in which the Chebyshev norm of distance travelled(step size) is minimized. (2) For the single layered models, a combined algorithm of the Successive Projection Method and Orthogonal Projection Method is developed. Several examples show the validity of the methods.

1. INTRODUCTION

Neurofuzzy GMDH (NF-GMDH) [1, 2] is a kind of adaptive learning network (i.e., a network type of GMDH) in the hierarchical structure. In the net, two input variables are introduced in each partial description. Fig. 1 shows the model structure. In the figure, the output from each partial description in a layer becomes the input variable in the next layer, respectively. The final output is given by the average of the outputs in the last layer. Since the input of the m -th model in the p -th layer is the output variables of the $(m-1)$ -th and the m -th models in the $(p-1)$ -th layer, the number of Gaussian functions being K , then

$$y^{pm} = F(y^{p-1,m-1}, y^{p-1,m}) = \sum_{k=1}^4 \mu_k^{pm} w_k^{pm} \quad (1)$$

$$\mu_k^{pm} = \exp \left[-\frac{(y^{p-1,m-1} - a_{k,1}^{pm})^2}{b_{k,1}^{pm}} - \frac{(y^{p-1,m} - a_{k,2}^{pm})^2}{b_{k,2}^{pm}} \right] \quad (2)$$

where μ_k^{pm} and w_k^{pm} are the k -th Gaussian function and its corresponding weight parameter, respectively, of the m -th model in the p -th layer, and a_{ki}^{pm} and b_{ki}^{pm} are the parameters of the Gaussian function applied to the i -th input variables ($i = 1, 2$) of the m -th model in the p -th layer. Let the number of partial descriptions in each layer be M and the number of layers be P . The final output y is the average of outputs in the last layer.

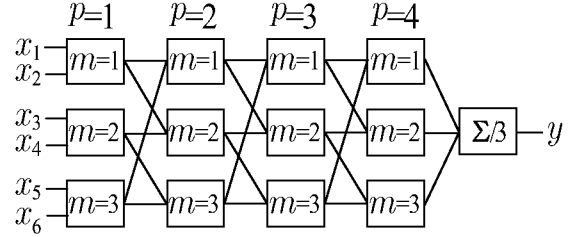


Figure 1: Structure of Neurofuzzy GMDH with six input variables

$$y = \frac{1}{M} \sum_{m=1}^M y^{pm} \quad (3)$$

In this paper, a Successive Projection Method (SPM) based on the Chebyshev norm for fast learning algorithm is developed for multilayered models. A combined algorithm of the SPM and Orthogonal Projection Method(OPM) is developed for singlelayered models. These algorithms are compared with the Least Mean Square(LMS) rule.

2. SUCCESSIVE PROJECTION METHOD FOR MULTI LAYERED NF-GMDH

The learning rate is one of heuristic constants in LMS rule. The cost function will not converge if the bigger learning rate is employed, and it will not converge easily if the smaller learning rate is employed. This motivated the development of the Successive Projection Method (SPM)[3], which always stores the training pair exactly. The directions of parameter updates (search directions) are the same for each learning rule, it is only the distance traveled (step size) which distinguishes them. In SPM rule, the step size is calculated at each time step, such that the j -th input / output information presented to is stored exactly by the network. And it updates the network parameter vector to reduce the error in the output after each training pair is presented to the network.

The derivation of SPM rule is achieved to solve the following nonlinear optimization problem.

$$\min. \quad \sum_{p=1}^P \sum_{m=1}^M \sum_{k=1}^4 \left[(\Delta w_k^{pm})^s + \sum_{i=1}^2 \{ (\Delta a_{ki}^{pm})^s + (\Delta b_{ki}^{pm})^s \} \right] \quad (4)$$

$$\begin{aligned} \text{s.t.} \quad & \sum_{p=1}^P \sum_{m=1}^M \sum_{k=1}^4 \left[|\nabla y_{pmk}^w| \Delta w_k^{pm} + \sum_{i=1}^2 \{ |\nabla y_{pmki}^a| \Delta a_{ki}^{pm} + |\nabla y_{pmki}^b| \Delta b_{ki}^{pm} \} \right] \\ & = |\rho| \end{aligned} \quad (5)$$

$$\Delta w_k^{pm} \geq 0, \quad \Delta a_{ki}^{pm} \geq 0, \quad \Delta b_{ki}^{pm} \geq 0 \quad (6)$$

where $\rho = y(\mathbf{w}, \mathbf{a}, \mathbf{b}, \mathbf{x}) - y^*$, $\nabla y_{pmk}^w = \partial y / \partial w_k^{pm}$. Since s is a positive real number, the nonnegativity restrictions are added to the constraint. From the above optimization problem we can derive the following learning rule.

$$\begin{aligned} w_k^{pmNEW} &= w_k^{pmOLD} \\ &+ \eta \operatorname{sgn}(\rho \nabla y_{pmk}^w) \frac{|\rho|}{\Sigma} |\nabla y_{pmk}^w|^{1/(s-1)} \end{aligned} \quad (7)$$

where, η is a relaxation rate, and

$$\begin{aligned} \Sigma &= \sum_{p=1}^P \sum_{m=1}^M \sum_{k=1}^4 \left[|\Delta y_{pmk}^w|^{s/(s-1)} \right. \\ &\left. + \sum_{i=1}^2 \left\{ |\Delta y_{pmki}^a|^{s/(s-1)} + |\Delta y_{pmki}^b|^{s/(s-1)} \right\} \right] \end{aligned} \quad (8)$$

∇y^{pm} denotes $\partial y / \partial y^{pm}$. $\nabla y^{Pm} = 1/M$ for the last layer and

$$\nabla y^{pm} = -2 \sum_{l=m}^{m+1} \nabla y^{p+1,l} \sum_{q=1}^4 w_q^{p+1,l} \mu_q^{p+1,l} \frac{(y^{pm} - a_{qr}^{p+1,l})}{b_{qr}^{p+1,l}} \quad (9)$$

for intermediate layers, where r is 1 if $l = m + 1$, and 2 if $l = m$, and $m + 1$ is 1 if $m = M$.

We can derive the learning rule for a_{ki}^{pm} and b_{ki}^{pm} in a similar manner. The above operation is performed for each training data, which thus implies one iteration of learning. The operations are repeated for a fixed number of learning sessions.

3. ORTHOGONAL PROJECTION METHOD FOR SINGLE LAYERED NF-GMDH

The RBF networks depend linearly on a set of weights \mathbf{w} which are updated using gradient descent rules. Since the single layered NF-GMDH is simply the sum of several RBF networks, it performs a fixed nonlinear transformation of the input, mapping to a high - dimensional sparse internal representation. The Orthogonal Projection Method (OPM) [4] using Gram - Schmidt Orthogonalisation process is based on calculating successive orthogonal search directions for the last j training samples. This produces the following update rule

$$\Delta \mathbf{w}_j = \nabla y(\mathbf{w}_j, \mathbf{x}_j) - \sum_{i=1}^{j-1} \frac{\nabla y(\mathbf{w}_j, \mathbf{x}_j)^T \Delta \mathbf{w}_i}{\Delta \mathbf{w}_i^T \Delta \mathbf{w}_i} \Delta \mathbf{w}_i \quad (10)$$

The combined algorithm of OPM and SPM is developed for updating all parameters including \mathbf{a} and \mathbf{b} and we call it the Hybrid Projection Method (HPM). The proposed algorithm is shown as follows.

[Algorithm HPM]

Step-0.

The initial values of parameters are set as follows. a_{ki} 's are uniformly spaced on the actual map. The values of parameter b_{ki} 's in Gaussian functions are chosen as to approximate triangular membership functions(i.e., B-spline of degree one). $\mathbf{w}_1 := \mathbf{0}$, $i := 1$.

Step-1.

Compute

$$\Delta \mathbf{w} = \frac{-\delta}{\|\nabla y(\mathbf{w}_j, \mathbf{x}_j)\|^2} \nabla y(\mathbf{w}_j, \mathbf{x}_j) \quad (11)$$

and

$$\mathbf{w}_2 = \mathbf{w}_1 + \Delta \mathbf{w} \quad (12)$$

\mathbf{a} , \mathbf{b} are updated in the same manner.

$j := 2, i := i + 1$.

Step-2.

Compute $\Delta \mathbf{w}_j$ using Eq.(10). If

$$\Delta \mathbf{w}_j^T \Delta \mathbf{w}_j < \varepsilon \quad (13)$$

then $j := 1$, $\mathbf{w}_1 := \mathbf{w}_j$ and proceed from Step-1.

Step-3.

Compute

$$\Delta \mathbf{w} = \frac{-\delta_j \Delta \mathbf{w}_j}{\nabla y(\mathbf{w}_j, \mathbf{x}_j)^T \Delta \mathbf{w}_j} \quad (14)$$

and

$$\mathbf{w}_{j+1} = \mathbf{w}_j + \Delta \mathbf{w} \quad (15)$$

Step-4.

$j := j + 1$, $i := i + 1$ and proceed from Step-2.

5. NUMERICAL EXAMPLES

In order to examine the effectiveness of SPM and HPM, the simulation of supervised learning is performed using two set of three different nonlinear mappings. The input variables take their values randomly chosen from unit interval $[0,1]$.

For the multilayered model, 200 input and output pairs of data are obtained as a training set. In two cases of the followings, x_6 is considered in the network but does not exist in the functions.

· Sine function

$$y^* = 0.5 + 0.5 \prod_{i=1}^5 \sin(\pi x_i) \quad (16)$$

· Strongly nonlinear function 1

$$y^* = \frac{1}{20} \{ \cos(\pi x_1) + \sin(\pi x_2) + \cos(\pi x_3) + \cos(\pi x_4) + \sin(\pi x_5) \} \quad (17)$$

· Strongly nonlinear function 2

$$y^* = 0.2(0.5 + 0.5 \cos(2\pi x_1) + x_2 x_3 + x_4 \sin(\pi x_5) + 0.5 x_6)^2 \quad (18)$$

From simulation results, shown in figs. 2, 3 and 4, the learning by SPM converges faster than by LMS.

For the singlelayered model, The mean squared errors, shown in figs. 5, 6 and 7, are calculated with 600 checking data which have not been used during the training, when every training data is presented to the network.

· Quadratic function

$$y^* = \frac{1}{4} \{x_1 + x_2\}^2 \quad (19)$$

· Nonlinear function

$$y^* = \frac{1}{4} \{ \cos(\pi x_1) + \sin(\pi x_2) \}^2 + 1 \quad (20)$$

· Sine function

$$y^* = 0.5 + 0.5 \sin(1.5\pi x_1) \sin(1.5\pi x_2) \quad (21)$$

In figs. 5, 6 and 7, the results of combined algorithm are plotted as HPM. The results of SPM and

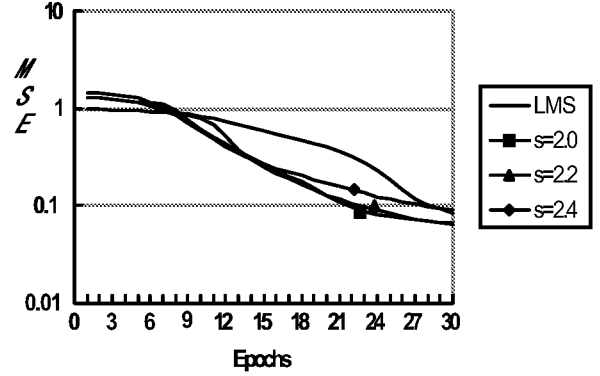


Figure 2: Mean squared error during training for the sine function, $y^* = 0.5 + 0.5 \prod_{i=1}^5 \sin(\pi x_i)$.

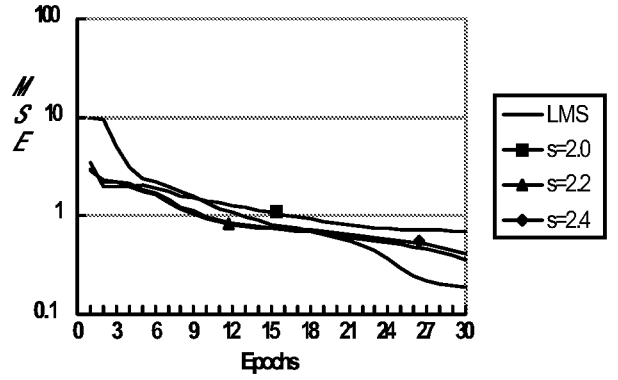


Figure 3: Mean squared error during training for the strongly nonlinear function with five input variables, $y^* = \{ \cos(\pi x_1) + \sin(\pi x_2) + \cos(\pi x_3) + \cos(\pi x_4) + \sin(\pi x_5) \} / 20$.

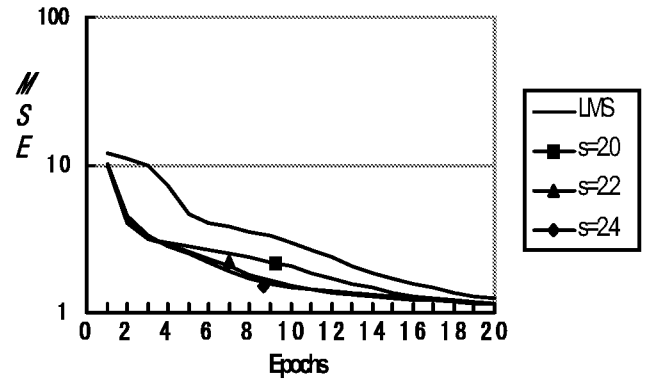


Figure 4: Mean squared error during training for the strongly nonlinear function with six input variables, $y^* = 0.2(0.5 + 0.5 \cos(2\pi x_1) + x_2 x_3 + x_4 \sin(\pi x_5) + 0.5 x_6)^2$.

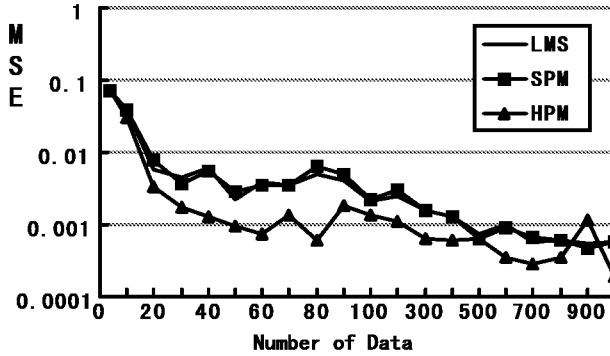


Figure 5: Mean squared error during training for the quadratic function, $y^* = \{x_1 + x_2\}^2 / 4$.

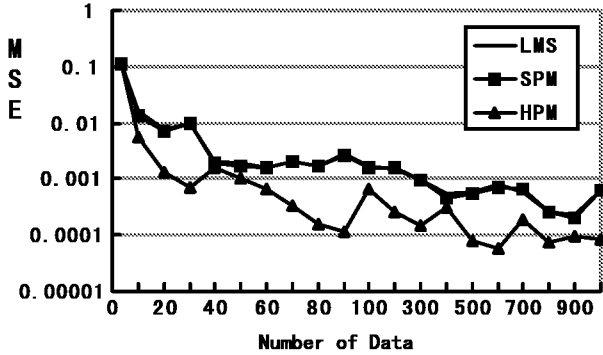


Figure 6: Mean squared error during training for the nonlinear function, $y^* = \{\cos(\pi x_1) + \sin(\pi x_2)\}^2 / 4 + 1$.

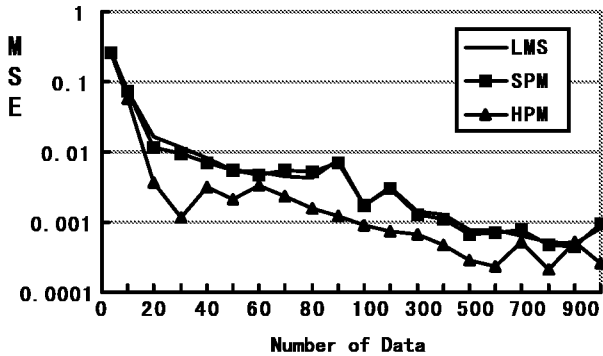


Figure 7: Mean squared error during training for the sine function, $y^* = 0.5 + 0.5 \sin(1.5\pi x_1) \sin(1.5\pi x_2)$.

LMS are almost same, but the learning converged faster than others by HPM.

5. CONCLUSION

The convergence of the learning has been improved

by the proposed algorithms especially for the early stage of learning. For on-line modelling and control, this advantage is important to prevent initial defects of the products in any production systems.

References

- [1] K.Nagasaka, H.Ichihashi, and R.Leonard, Neuro-Fuzzy GMDH and Its Application to Modelling Grinding Characteristics, Int. J. Prod. Res., Vol. 33, No. 5, (1995) 1229-1240
- [2] H.Ichihashi, N.Harada and K.Nagasaka, Selection of the Optimum Number of Hidden Layers in Neuro-fuzzy GMDH, Proc. FUZZ-IEEE/IFES'95, Vol.3, (1995) 1519-1526
- [3] M. Brown and C. Harris, Neurofuzzy Adaptive Modelling and Control, Prentice Hall, New York (1994)
- [4] J. Shao, Y. C. Lee and R.Jones, Orthogonal Projection Method for Fast On-line Learning Algorithm of Radial Basis Function Neural Networks, Proc. INNS World Congress on Neural Networks, Portland OR, Vol. 3 (1993) 520-535