http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (4): 852-856, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.852.856

Evolutionary Learning Algorithm for Multi-layer Morphological Neural Networks

He Chunmei
School of Information Engineering, ECJTU, Nanchang 330013, China

Abstract: Morphological Neural Network (MNN) is a novel and important neural network and it has many applications such as image processing and pattern recognition. It makes sense to research the learning algorithm of MNN and its application. A method based on genetic algorithm is presented to train and implement multi-layer morphological neural network in this study. The algorithm calculates the weights and biases of morphological neural network and the genetic algorithm automatically acquire the learning rate. After that, the trained morphological neural network is applied to image restoration. The image restoration simulation and a comparison with the median filter are shown in the end. It shows that the morphological neural network is a quite good method applied to image restoration.

Key words: Mathematical morphology, morphological neural network, learning algorithm, image restoration, genetic algorithm

INTRODUCTION

Morphological Neural Network (MNN) is a kind of neural network combined artificial neural network and mathematical morphology. The MNN is a novel neural network and has been applied in many applications. Therefore it makes sense to research the MNN and its application. The algorithms of multi-layer MNN are paid close attention in this study in order to provide better algorithm in the training of MNN.

MNN was firstly presented by Ritter and Davidson (1991), Davidson and Ritter (1990) and Ritter (1991). An artificial neural network is said to be morphological if every neuron performs an elementary operation of Mathematical Morphology, that is, in MNN's neurons, the classical operations of addition and multiply are replaced by addition and maximum or minimum operations. MNN use algebraic lattice operations structure: semi-Ring ($\Re_{\pm\infty}$, $\vee$, $\wedge$, +, +), while the Traditional Neural Networks (NN) are based on the Ring ($\Re$, +, $\times$).

There are different kinds of MNN such as Morphological Perceptron (MP) (Ritter and Sussner (1996), multi-layer MNN (Zhang *et al*., 2003), morphological associative memories (He *et al*., 2011) and so on. The Fuzzy-morphology neural network are proposed for automatic target recognition by Yonggwan (1998). A doubly local Wiener filtering algorithm using elliptic direction and mathematical morphology is proposed to restore image (Zhou and Shui, 2008). Some algorithms for MP were presented by Ritter and Sussner (1996) and Sussner and Esmi (2011) in which the active functions are limited to hard limiter function. The active functions in multi-layer MNN aren't limited to hard limiter function. The multi-layer MNN (Zhang *et al*., 2003) are trained by Least Mean Square algorithm and applied to color image restoration. In (Araujo, 2010; Araujo *et al*., 2011) the MNNs are trained by BP algorithm and applied in image processing and stock market prediction. The learning algorithms (Zhang *et al*., 2003; Araujo, 2010; Araujo *et al*., 2011) are mostly least squared error algorithm and the learning rate are fixed and manual set. The unsuitable learning rate often causes the learning algorithm oscillated and divergent. In this study, a new learning algorithm based on genetic algorithm is proposed for multi-layer MNN and the trained MNN are applied to image restoration.

MORPHOLOGICAL NEURAL NETWORK

Morphological Neural Network (MNN) is a kind of neural network combined neural network and morphology. In this section the architecture of MNN are introduced and a new learning algorithm for MNN is proposed.

Architecture of the MNN: MNN use algebraic lattice operations structure: semi-Ring ($\Re_{\pm\infty}$, $\vee$, $\wedge$, +, +), the classical matrix operator including addition and multiply were replaced by the lattice algebra operator and caused a different nonlinear transformation in the systems. The architecture and learning algorithm of multi-layer MNN are simply introduced in this section.

The architecture of the erosion-based MNN is shown in Fig. 1 and in essence it is a multi-layer feed forward neural network with many inputs and single output.

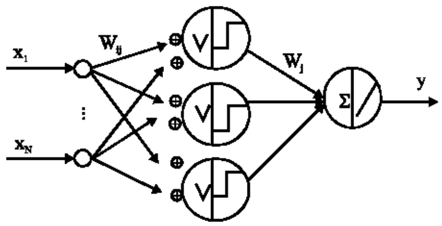

Fig. 1: The architecture of the erosion-based MNN

The erosion-based MNN model is given by the following equation:

$$y = f(v_j) = v_j \tag{1}$$

where:

$$v_j = \sum_{j=1}^{H} W_j \cdot u_j(t)$$

with:

$$u_j(t) = f(z) = 1 \Big/ \left(1 + e^{-z}\right)$$

and

$$z = \vee_{i=1}^{N} \left(x_i(t) + W_{ij}\right)$$

where, $x_i = \{x_{i1}, x_{i2},..., x_{iN}\}$, $(i = 1, 2,..., K)$ denotes the ith input of the MNN. N is the dimension of a training pattern which is decided by the structure elements, $\vee$ denotes the maximum operator and $\wedge$ is minimum operator. The output of the jth hidden neuron is u_j (t) (j = 1, 2,..., H). K is the training pattern number and H is the neuron number in the hidden layer. The final output of the erosion-based MNN is y. W_{ij} represents the connected weight between the ith node in the input layer and the jth node in the hidden layer, while W_j denotes the connected weight between the jth node in the hidden layer and the output layer.

Similarly, the dilation-based MNN is given by the following equation:

$$y = f(v_j) = v_j \tag{2}$$

where:

$$v_j = \sum_{j=1}^{H} W_j \cdot u_j(t)$$

with:

$$u_j(t) = f(z) = 1 \Big/ \left(1 + e^{-z}\right)$$

and

$$z = \wedge_{i=1}^{N} \left(x_i(t) + W_{ij}\right)$$

The architecture of the dilation-based MNNs is similar to erosion-based MNNs, but the operator in z is replaced by minimum operator $\wedge$.

Learning algorithm for MNN: The connected weights W_{ij} and W_j should been trained before the MNN used for image restoration. It is assumed that the dimension of every input training pattern is N and the training pattern number is K.

The training of the MNN is a processing of supervised learning, that is, the MNN is trained to make sure minimum the squared error between the desired outputs d_i and the actual outputs y_i. The squared error function is defined by:

$$E\left(W_{ij}, W_j\right) = \tfrac{1}{2} \sum_{i=1}^{N} \left(d_i - y_i\right)^2 \tag{3}$$

The proposed conjugate gradient algorithm based on genetic algorithm is used to train the connected weights W_{ij} and W_j of the MNN. In every iteration step, the optimized learning rate η is acquired by the genetic algorithm. The weights W of the MNNs are updated according to the iterative formulas:

$$W_{ij}(t+1) = W_{ij}(t) - \eta \cdot \frac{\partial E}{\partial W_{ij}(t)} + \alpha \cdot \left[W_{ij}(t) - W_{ij}(t-1)\right] \tag{4}$$

$$W_j(t+1) = W_j(t) - \eta \cdot \frac{\partial E}{\partial W_j(t)} + \alpha \cdot \left[W_j(t) - W_j(t-1)\right] \tag{5}$$

where, η is the learning rate, α is the active factor, $\partial E/\partial W_{ij}(t)$ and $\partial E/\partial W_j(t)$ are the partial derivative of the error function E. The iteration of Eq. 4-5 starts with an initial guess W(0) and stops when some desired conditions reached. $\partial E/\partial W_{ij}(t)$ and $\partial E/\partial W_j(t)$ are given by the gradient of E with respect to W at the points where this gradient exists. Then, following the chain derivation rule, $\partial E/\partial W_{ij}(t)$ and $\partial E/\partial W_j(t)$ are given as follows:

$$\frac{\partial E}{\partial Wi_{ij}(t)} = -(d_i - y_i) \cdot W_j(t) \cdot u_j(t) \cdot \left(1 - u_j(t)\right) \cdot \frac{\partial z_1}{\partial Wi_{ij}(t)} \tag{6}$$

$$\frac{\partial E}{\partial W_j(t)} = -(d_i - y_i) \cdot u_j(t) \tag{7}$$

The existence of the gradient of E with respect to W only hinges on the existence of the gradients, $\partial E/\partial W_j(t)$ where at the discontinued points, by literature (Blanco *et al*., 1995), define that:

$$\frac{\partial z_2}{\partial Wi_{ij}(t)} = \begin{cases} 1,\ \text{if } x_s(t) + Wi_{sj} > \vee_{i \neq s}\left[x_i(t) + Wi_{ij}\right] \\ 0.5,\ \text{if } x_s(t) + Wi_{sj} = \vee_{i \neq s}\left[x_i(t) + Wi_{ij}\right] \\ 0,\ \text{else } x_s(t) + Wi_{sj} < \vee_{i \neq s}\left[x_i(t) + Wi_{ij}\right] \end{cases}$$

The conjugate gradient algorithm based on genetic algorithm:

Step 1: Initialize: randomly initialize the weights W_{ij} and W_j. Let iteration step n = 1 and error precision $\varepsilon = 0.01$. Acquire the input pattern x and the desired outputs d

Step 2: Train the MNN first time and calculate the error function E

Step 3: WHILE (E<ε? and t<MaxTime) Do

- Compute the partial derivatives $\partial E/\partial W_{ij}(t)$ and $\partial E/\partial W_j(t)$ according to Eq. 6-7
- Acquire the optimized learning rate η by genetic algorithm

$$\eta[t] = \max\left\{\eta > 0 \mid E\left(W(t) + \lambda \cdot \frac{\partial E}{\partial W}\right) \text{is degressive for } \lambda \in (0, \eta]\right\}$$

- Compute the weights W_{ij} and W_j according to Eq. 4-5
- t = t+1
- Compute error function of the ith input training pattern x_i of MNNs according to Eq. 3

ENDWHILE

Step 4: Save the weights W_{ij} and W_j

The GA (He and Ye, 2011) used to get the optimal n(t) is introduced as below:

- **Initialization and coding:** Randomly generate an initial population P(0) = {λ(0, 1),..., (0, n)}. Each individual λ(0, j) (j = 1,..., n) in the population maps to a string. Every string is an individual point in the search space. All the points compose of the solution space S_0. Let t = 0 and max iteration step Max-gen = 140
- **Fitness function:** In this step, each string is decoded by an evaluator into an objective function value. For a Given λ(t, j)∈ P(t), compute Gλ(t, j), where P(t) means the ith generation population
- **Genetic selection:** The roulette wheel selection is used, where the live probability of λ(t, j) is defined by:

$$p_j = G(\lambda(t,j)) \Big/ \sum_{j'=1}^{n} G(\lambda(t,j'))$$

- **Genetic operator:** Randomly choose the individual pair $(\lambda(t, j_1), \lambda(t, j_2))$ from P(t) to match, where C_r is the one point crossover operator and the crossover probability is P_c

To accelerate convergence to optimal solution, let the mutation probability be 0.005. Return to step 2 until the satisfying solution is achieved, or the iteration number is more than Max-gen. Otherwise the GA is terminated and proper parameters are obtained by encoding the best string into a set of parameters.

SIMULATION

The MNN is applied to image restoration in this section. The desired output d of the MNN is the original image shown in Fig. 2. Fuzz the original image to acquire the degraded images A shown in Fig. 2. Following the below steps, the input training sample generates.

The image A is scanned three rows every time during the training process, following a zigzag path from top to bottom. To define the local input vector x_i of x at each pixel, a square window called as structure elementary is centered around it with size N = 9. Because the global sampling method is used, the marginal two rows and two columns are not scanned. If the size of A is m*n, then the acquired training sample number is K = (m-2)* (n-2). Every sample vector $x_i = \{x_{i1},...,x_{iN}\}$, where i = 1,..., K, N = 9.

The PSNR in the experiment is defined by:

$$PSNR = 10 \cdot \log_{10}^{\left(2^{n-1}\right)^2 / MSE}$$

where, MSE is the mean square error between the original image and the restored image. The input training sample x is normalized and the structure elementary and the desired outputs are limited to be the range [0,1].

The experiment environment is MATLAB 2010. The hidden layer of MNN has 5 nodes. Figure 2 and 3

Fig. 2(a-d): Restored results for (a) Original (b) Degraded (c) Restored by MNNs and (d) Restored by median filter fuzzy image

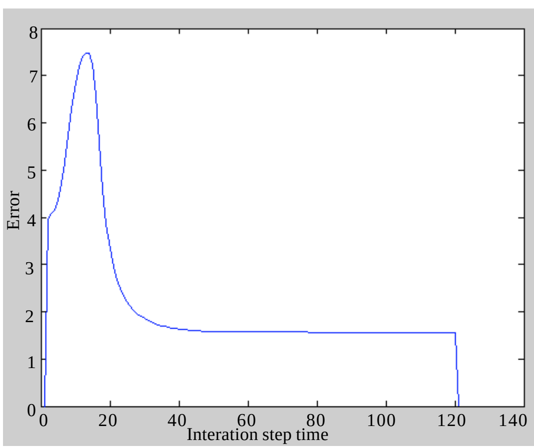

Fig. 3: MNN's error function E

show the simulation results. Figure 2 compares the fuzzy image restored by MNN with that restored by median filter. Figure 3 shows the variation trend of the MNN's Error function E. The PSNR of the image restored by MNNs is 22.7253 and the PSNR of the image restored by median filter is 25.6842. It shows that the MNN is suitable to restore fuzzy image and is a good tool for image processing.

CONCLUSION

The multi-layer MNN is introduced and a new conjugate gradient algorithm based on genetic algorithm is proposed to train the multi-layer MNN in this study. Afterward the trained MNN is applied to restore fuzzy noise images. The simulation has good results and the comparison with median filter is presented in the end. However, the genetic algorithm is only one kind of evolutional algorithms, emerging some more evolutional algorithms such as quantum genetic algorithm, shuffled frog-leaping algorithms with MNN are our future research.

REFERENCES

Araujo, R.A., 2010. Hybrid intelligent methodology to design translation invariant morphological operators for Brazilian stock market prediction. Neural Networks, 23: 1238-1251.

Araujo, R.A., A. Oliveira, S. Soares and S. Meira, 2011. An evolutionary approach to design dilation-erosion perceptrons for stock market indices forecasting. Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, July 12-16, 2011, ACM, New York, USA., pp: 1651-1658.

Blanco, A., M. Delgado and I. Requena, 1995. Identification of fuzzy relational equations by fuzzy neural networks. Fuzzy Sets Syst., 71: 215-226.

Davidson, J.L. and G.X. Ritter, 1990. Theory of morphological neural networks. Proceedings of the Digital Optical Computing II, SPIE Volume 1215, July 1, 1990, Los Angeles, CA., pp: 378-336.

He, C.M. and Y.P. Ye, 2011. Evolution computation based learning algorithms of polygonal fuzzy neural networks. Int. J. Intell. Syst., 26: 340-352.

He, C.M., Z.C. Ye, M. Han and Y.P. Ye, 2011. Perturbation robustness of morphological bidirectional associative memories networks. J. Nanjing Univ. Sci. Technol., 5: 664-669.

Ritter, G.X. and J.L. Davidson, 1991. Recursion and feedback in image Algebra. Proceedings of the Image Understanding in the 90s: Building Systems that Work, Volume 1406, October 17, 1990, McLean, VA., USA.

Ritter, G.X. and P. Sussner, 1996. An introduction to morphological neural networks. Proceedings of the 13th International Conference on Pattern Recognition, Volume 4, August 25-29, 1996, Vienna, Austria, pp: 709-717.

Ritter, G.X., 1991. Recent developments in image algebra. Adv. Electron. Electron Phys., 80: 243-308.

Sussner, P. and E.L. Esmi, 2011. Morphological perceptrons with competitive learning: Lattice-theoretical framework and constructive learning algorithm. Inform. Sci., 181: 1929-1950.

Yonggwan, W., 1998. Shift-invariant fuzzy-morphology neural network for automatic target recognition. IEICE Trans. Funda. Electron. Commun. Comput. Sci., E81-A: 1119-1127.

Zhang, L., Y. Zhang and Y.M. Yang, 2003. Color images restoration with multi-layer morphological (MLM) neural network. Proceedings of the International Conference on Machine Learning and Cybernetics, Volume 5, November 2-5, 2003, IEEE Computer Society, Washington, DC., USA., pp: 2831-2834.

Zhou, Z.F. and P.L. Shui, 2008. Wavelet-based image denoising via double local wiener filtering using directional windows and mathematical morphology. J. Electron. Inform. Technol., 30: 885-888, (In Chinese).