A Novel Fuzzy Morphology, Part II: Neural Network Implementation

Yonggwan Won
Electronics and Telecommunications Research Institute
Taejon-City, Yoosung-Ku, Yoosung, Kajung-Dong 161

Bae-Ho Lee
Department of Computer Engineering
Chonnam National University
Kwangju-City, Puk-Ku, Yongbong-Dong 300

ABSTRACT

A shared-weight neural network that performed classification based on the features extracted with the fuzzy morphological operation is introduced. Learning rules for the structuring elements, degree of membership, and weighting factors are also precisely described. In application to handwritten digit recognition problem, the fuzzy morphological shared-weight neural network produced the results which are comparable to the state-of-art for this problem.

I. INTRODUCTION

Fuzzy set theory [Zadeh] has been successfully used in many pattern recognition applications. However, it has been a big issue how to determine the parameters for the fuzzy operations such as the degree of membership. Mathematical morphology [Dougherty] has also been used in pattern recognition as a feature extraction methodology. It also has a fundamental problem which is how to design the structuring elements, because the result is completely dependent on the structuring elements.

Multilayer neural networks have been widely employed to solve many problems with the development of the effective error back-propagation learning rule [Rumelhart 86ab, Werbos]. This learning rule is a objective function based algorithm which is , in detail, an iterative gradient descent algorithm that updates the parameters (i.e., weights and biases) to minimize the error. Shared-weight neural network [le Cun] is a special class of multilayer neural networks. It is a heterogeneous system that performs classification based on the high-order features combined from locally extracted.

Main purpose of this paper is to introduce an implementation of a neural network that learns the parameters of the fuzzy morphological operation defined in [Won 95b]. In Section II, a shared-weight neural network that performs fuzzy morphological operation for feature extraction is described. In Section III, we present some experimental results obtained from handwritten digit recognition problem. Finally, in Section IV, conclusions and further works are discussed.

III. NEURAL NETWORK IMPLEMENTATION

In this section, we introduce a shared-weight neural network that performs classification and feature extraction simultaneously. Feature extraction stage of this network performs our fuzzy erosion and dilation [Won 95b]. Learning rules for the feature extraction network are also provided.

3.1. SHARED-WEIGHT NEURAL NETWORK

The basic idea of the *shared-weight* network [le Cun] is to reduce the degrees of freedom in the network for better generalization and to form high order features from local features extracted by learned convolution kernels. Fig. 1 shows the structure of the typical shared-weight neural network. This network is composed of two parts: a feature extraction network followed by a feedforward network. The feedforward network is a feedforward network. The feature extraction network can have one or more layers, and each layer can also have one or more feature maps. The layer in this network performs feature extraction by linear or non-linear convolution of its input with the kernels (also called *structuring elements, templates, masks, feature detectors*). The convolution output is subsampled. Therefore, the sizes of the feature maps are determined by the sampling rate for the convolution over their input.

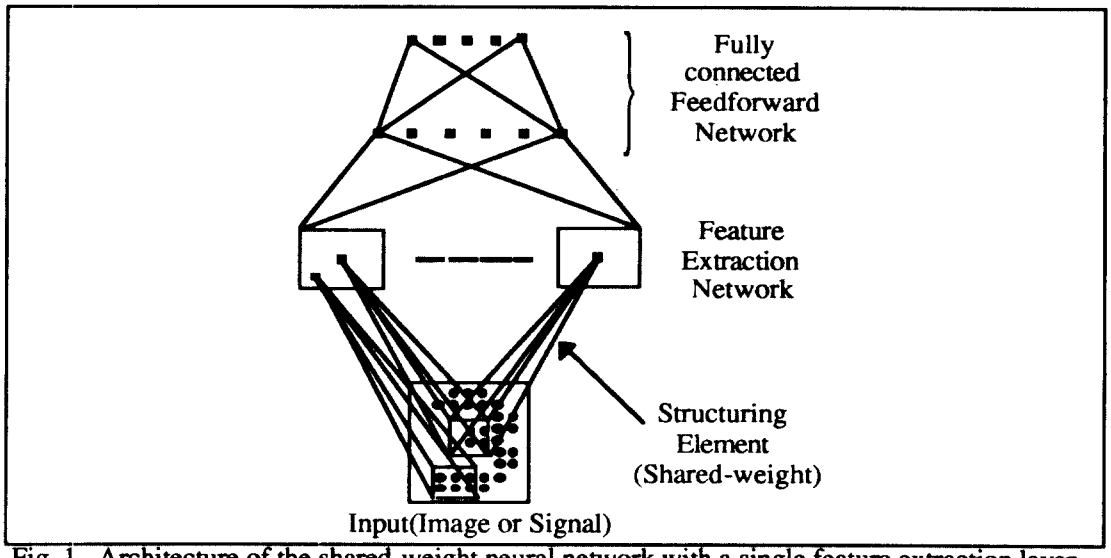

Fig. 1. Architecture of the shared-weight neural network with a single feature extraction layer in the feature extraction network and one hidden layers for the feedforward network.

Nodes in a feature map in the first feature extraction layer have a small number of identical weights, and each node corresponds to a certain position in the input pattern. Therefore, the number of free weights in a feature map in this layer is dramatically reduced to only the size of the its kernel over its input (plus the number of the nodes in the feature map if there is one *bias* per node). On the other hand, each node extracts local information from its input. Each feature map in a higher feature extraction layer has as many kernels as the number of the feature maps in the next lower feature extraction layer. As for the first layer, the weights for the nodes in the same feature map are identical. Furthermore, the nodes in this layer combine local information coming from the feature maps in the next lower layer. Finally, the highest layer provides the input for the feedforward network.

3.2. NODE OPERATION

Previous work approximated the ordinary erosion and dilation with the generalized-mean operator by setting the parameter p to a large positive or negative value [Gader 93a, Won 95b]. The nodes in the feature extraction network performed a novel gray-scale Hit-Miss transform which was defined as subtraction of the dilation from the erosion. In other words, the parameter p that represents the degree of membership was fixed and a node in the feature extraction network calculated both approximated erosion and dilation. Nodes in the classification network performed the standard operation.

In order to allow the nodes in the feature extraction network to perform a single operation (i.e., p is either positive, negative or zero), we first formalized an equation:

$$(f \otimes_f g)(\mathbf{x}) = g(r(f(\mathbf{z})-t_{\mathbf{x}}(\mathbf{z})) ; p, w_i). \qquad (1)$$

This equation is performing the fuzzy erosion if p has a negative value and fuzzy dilation if p has a positive value [Won 95b]. Note that the structuring element $t_{\mathbf{x}}$ represents the reflected structuring $(m^*)_{\mathbf{x}}$ if p has a positive value.

3.3. LEARNING RULE

In equation (1), there are three sets of parameters: structuring element (t), degree (p), and weighting factor (w). Among them, structuring element and weighting factor can be fixed. Furthermore, in general, the structuring element has been selected arbitrarily (i.e., zeros). However, a structuring element designed through learning process produced better performance [Won 95a], and weighting factor provides more degree of freedom.

In this section, we provide the derivation for the learning rules required to implement the learning algorithm. Here we only show explicitly the derivation of the learning rule for the feature extraction network, and that for the classification network is widely available [Rumelhart, Webos, Won 95a]. Assume that each feature extraction layers have a single feature map for simplifying the formulation, and it can be easily extended for multiple feature maps. Suppose we want to update the parameters associated with node j. Let the output for the node j be

$$O_j = net_j = \{\sum_i w_{ji}[r(O_i - t_{ji})]^{p_j}\}^{1/p_j}. \qquad (2)$$

In these equations, O_j denotes the output of the node j and O_i does the input which is an output of the node i. Therefore, t_{ji} is a member of the structuring element that associates the node j and the node i. In other words, the first subscript indicates the node in the next higher layer. In terms of the morphological operation, the subscript j represents the location of the origin (center) of the structuring element in the input domain.

In order to take the derivative for the equations in (2), we first need to take log and obtain the following equations:

$$\log(net_j) = \frac{1}{p_j}\log\{\sum_i w_{ji}[r(O_i - t_{ji})]^{p_j}\} \qquad (3)$$

Taking the derivative on both sides of this equation with respect to p_j, t_{ji} and w_{ji}, respectively, yields

$$\frac{\partial net_j}{\partial p_j} = -\frac{net_j}{p_j^2}\log\left(net_j^{p_j}\right) + \sum_i w_{ji}\left[r(O_i - t_{ji})\right]^{p_j}\log\left[r(O_i - t_{ji})\right], \tag{4a}$$

$$\frac{\partial net_j}{\partial t_{ji}} = -w_{ji}\left\{\frac{r(O_i - t_{ji})}{net_j}\right\}^{p_j - 1}\frac{\partial r(O_i - t_{ji})}{\partial(O_i - t_{ji}))}, \tag{4b}$$

and

$$\frac{\partial net_j}{\partial w_{ji}} = \frac{1}{p_j \cdot net_j^{p_j - 1}}\left[r(O_i - t_{ji})\right]^{p_j}. \tag{4c}$$

Again taking the derivative on both sides of the equations in (3) with respect to O_i produces

$$\frac{\partial net_j}{\partial O_i} = w_{ji}\left\{\frac{r(O_i - t_{ji})}{net_j}\right\}^{p_j - 1}\frac{\partial r(O_i - t_{ji})}{\partial(O_i - t_{ji})}. \tag{5}$$

Note that the last terms of the equations in (4b) and (5) are described as

$$\frac{\partial r(O_i - t_{ji})}{\partial(O_i - t_{ji})} = r'(O_i - t_{ji}) = r(O_i - t_{ji})\{1 - r(O_i - t_{ji})\} \tag{6}$$

because we use the sigmoid function for r.

Let d represent the parameters p_j, t_{ji}, and w_{ji}. For gradient descent learning rule to reduce the error with respect to d, we apply the chain rule and obtain the equation

$$\Delta d = -\eta\frac{\partial E}{\partial d} = -\eta\frac{\partial E}{\partial net_j}\cdot\frac{\partial net_j}{\partial d}. \tag{7}$$

Since the feature extraction layers are considered the hidden layer in a feedforward network, the first term of this equation can be defined and written as

$$\delta_j = -\frac{\partial E}{\partial net_j} = -\frac{\partial E}{\partial O_j} = -\sum_k\left(\frac{\partial E}{\partial O_k}\cdot\frac{\partial O_k}{\partial net_k}\cdot\frac{\partial net_k}{\partial O_j}\right) \tag{8}$$

where the node k is the one in the next higher layer. This is called delta error of the node j in the learning rule for the standard feedforward neural network. In the same manner as for deriving (8), the first two terms of (8) can be written as

$$-\frac{\partial E}{\partial O_k}\cdot\frac{\partial O_k}{\partial net_k} = -\frac{\partial E}{\partial O_k} = \delta_k \tag{9}$$

which is delta error of the node k which is in the next higher layer than the node j. The last term of the equation (8) has two different forms, depending upon which layer the node k belongs to. For the nodes in the highest feature extraction layer, it can be written as

$$\frac{\partial net_k}{\partial O_j} = \frac{\partial(\sum_j w_{kj} O_j + bias_k)}{\partial O_j} = w_{kj} \tag{10}$$

since the node k is a ordinary node which calculates the weighted linear combination of the inputs for its net input net_k. For the nodes in other feature extraction layers, from the equation (5), it can be written as

$$\frac{\partial net_k}{\partial O_j} = w_{kj} \left\{ \frac{r(O_j - t_{kj})}{net_k} \right\}^{p-1} \frac{\partial r(O_j - t_{kj})}{\partial (O_j - t_{kj})}. \tag{11}$$

Therefore, from the equations (7) through (11), the learning rule for the structuring element t_{ji} can be obtained. Finally, the learning rule for the structuring element associated with node j can be summarized by

$$\Delta d = \eta \delta_j \frac{\partial net_j}{\partial d} \tag{12a}$$

where

$$\delta_j = \sum_k \delta_k \frac{\partial net_k}{\partial O_j}. \tag{12b}$$

The last factor in (12a) is given in (4) for all parameters p_j, t_{ji}, and w_{ji}, and the last factor in (12b) is given by either (10) or (11) depending on which layer the node belongs to.

There are several implementation details for this learning algorithm. The index k is representing all the nodes whose output calculation uses the output of node j. Because t_j's are identical for all j's in the same feature map, t_{ji}'s for all j's should be accumulated instead of updating t_{ji}'s after every j, and then update at the end. The updating rule for this shared-weight constraint is well described in [Won 95a]. A practical problem in implementing this learning rule is the limitation on the magnitude of the parameter p_j. Larger range allows more accurate approximation of the ordinary Min and Max operations. However, for large numbers, the region where the gradient for the generalized-mean is non-zero is very small and the gradient is very large in that region. Thus, the training process may oscillate for large values. Note that the generalized-mean value is equal to harmonic mean if p = -1, geometric mean if p = 0, and arithmetic mean if p = 1.

IV. EXPERIMENTAL RESULTS

We conducted some preliminary experiments with the shared-weight neural network that described in the previous section for handwritten digit recognition problem. We collected 1000 digits for each class from the handwritten digit data base which were extracted from the USPS mail pieces [Gader 93ab, Won 95a]. The digits were normalized to the fixed size of 24 X 18 using moment normalization[Casey]. Some samples of handwritten digits are shown in Fig. 4.

Among the collected images, 600 digits per class were used for training and 400 for testing the network. The networks had a single feature extraction layer with twelve feature maps and thirty hidden units for the classification network. Subsampling rate of two was used. The size of the structuring elements was 5 X 5. We ran five experiments with different initial values for the parameters. All parameters were learned.

Fig. 4. Some samples of handwritten digits.

For all experiments, the learning rate was 0.02 and the momentum was 0.9. The training process was stopped by the pre-selected maximum epoch of 100 or Root-Mean-Squared-Error (RMSE) of 0.05. However, for most our experiments, the training process was terminated by the RMSE criteria. The parameters were initialized with the random values obtained from the range [-0.5, 0.5] and the magnitude of the parameter p was clipped at 3. As shown in Table 1, the network produced the results which are favorably comparable to those from other approaches [lGader 93ab, le Cun].

Table 1. Results for handwritten digit recognition problem.

		Exp. 1	Exp. 2	Exp. 3	Exp. 4	Exp. 5
Train	Epoch	31	29	28	27	29
	RMSE	0.0482	0.0491	0.0494	0.0489	0.0487
	Correction Rate	99.0 %	98.8 %	98.8 %	99.0 %	98.8 %
Test	RMSE	0.0698	0.0711	0.0714	0.0702	0.0706
	Correction Rate	95.3 %	95.0 %	94.8 %	95.0 %	94.8 %

IV. CONCLUSION

We have described a shared-weight neural network that performed classification based on the features extracted with our novel fuzzy morphological operation [Won 95b]. A node in the feature extraction stage of the network performs fuzzy morphological operation that produces the output values between the standard erosion and dilation. Precise description for the learning rules for the structuring elements, degree of membership, and weighting factors was also provided.

The network was applied to handwritten digit recognition problem. We demonstrated that the fuzzy morphological shared-weight neural network produced the results which are favorably comparable to the state-of-art for this problem [Gader 93ab, le Cun].

The main goal of this paper was to introduce an implementation of a neural network that learns the parameters of the fuzzy morphological operations defined in [Won 95b]. Even though we selected the shared-weight network in our preliminary study, our definitions can be used as a node operation for other networks such as the standard feedforward network. Also more applications including gray-level images and signals should be considered for future works.

REFERENCES

R. Casey, "Moment Normalization of Handprinted Characters," in *IBM J. Res. Develop.*, pp. 548 - 557, September, 1970.

E.R. Dougherty, *An Introduction to Morphological Image Processing*, SPIE Optical Engineering Press, 1992

P.D. Gader, Y. Won and M.A. Khabou, "Image Algebra Networks for Pattern Classification," *SPIE Mathematical Morphology and Image Algebra*, pp. 157 - 168, July, 1993a.

P.D. Gader and Mohamed A. Khabou, "Automated Feature Generation for Handwritten Digit Recognition by Neural Networks.", in *Proc. of the Int'l Workshop Frontiers Handwriting Recognition*, pp. 21 - 31, Buffalo, NY, 1993b.

Y. le Cun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, "Handwritten Digit Recognition: Application of Neural Network Chips and Automatic Learning." *IEEE Communications Magazine*, November, pp. 41 - 64, 1989.

D.E. Rumelhart, G.E. Hiton and R.J. Williams, "Learning Internal Representations by Error Propagation." in D.E. Rumelhart and J.L. McClelland, editors, *Parallel Distributed Processing:Explorations in the Microstructure of Cognition*, MIT Press, Cambridge, MA, 1986a.

D.E. Rumelhart and J.L. McClelland, *Parallel Distributed Processing:Explorations in the Microstructure of Cognition: Foundations*, Vol. 1, MIT Press, Cambridge, MA, 1986b.

P.J. Werbos, "Beyond Regression: New Tools for Prediction and Analysis in the Behavior Science." Doctoral Dissertation, Applied Mathematics, Harvard University, Boston, MA, November 1974.

Y. Won, "Nonlinear Correlation Filter and Morphology Neural Networks for Image Pattern and Automatic Target Recognition," Ph.D. Dissertation, University of Missouri-Columbia, August, 1995a.

Y. Won and B. Lee, "A Novel Fuzzy Morphology, Part I: Definitions," in *Korea Fuzzy Logic and Intelligent Systems*, see this proceedings, 1995b.

L.A. Zadeh, "Fuzzy Sets," *Informa. Contr.*, vol. 8, pp. 338 - 353, 1965.