Image Binarization By Back Propagation Algorithm

Tao Chen Mikio Takagi

Takagi Lab, Dept. of Electronics Engineering
Institute of Industrial Science, Univ. of Tokyo
7-22-1 Roppongi, Minato-ku, Tokyo 106, Japan
email: tao@tkl.iis.u-tokyo.ac.jp, takagi@tkl.iis.u-tokyo.ac.jp

Abstract

Image binarization is a fundamental research theme in image processing and an improtant preprocessing method in image recognition and edge/boundary detection. It is very difficult to select the corresponding threshold for each image in different application domains. In this paper, we used a multi-layer feed-forward neural net (multi-layer perceptron) as a threshold transfer to select the visually satisfied threshold by back propagation algorithm. In order to improve network performance and reduce the occurrence of local minima, we introduced extra hidden units and made many training runs starting with different sets of random weights. Besides, we also introduced the collective decision produced by the ensemble to less error probably made by any of the individual network. The architecture of our neural perceptron makes it available to perform multi-stage image processing, programmingly arrange the relationship between groups within different layers, and emphasize not only the local image texture but the globle information. The comparison with other threshold-selecting methods shows that our neural net method is able to select visually satisfied threshold and to obtain good restoration image using the result binarized image.

KEY WORDS: Image Preprocessing, Binarization, Back Propagation, Neural Network

1 Introduction

Artificial neural net models have been researched for many years in hope of achieving human-like performance in the field of image processing and pattern recognition [1, 5]. An important research theme in image processing and a necessary preprocessing method in image recognition and edge/boundary detection is image binarization. Many image processing schemes (including image algebra) and very fast image transmission take image binarization as preprocessing. Since the binarization problem is difficult to define and to evaluate, a large number of schemes have been presented in the literature since the early stage of image processing and pattern recognition. Up to now, the most popular method for image binarization has been by the use of image histogram. However, how to select the corresponding threshold for each image in different application cases is still an open question. In this paper, we use multi-layer perceptron as threshold transfer to select the threshold by back propagation algorithm. Very visually satisfied results have been obtained. The calculating derivatives exactly and efficiently in any differentiable functions underlies the effective of back propagation.

Kolmogorov has proved that a three layer perceptron with N(2N+1) nodes using continuously increasing non-linearities can compute any continuous function of N variables [2, 3]. This paved way for neural net to be used in image binarization, pattern recognition/classification and image transformation. In the following sections, we will first describe the general neural net model and image binarization method. Then highlight some key ideas and the structure of the neural model designing, and finally present the implementation and the comparison results with other binarization methods.

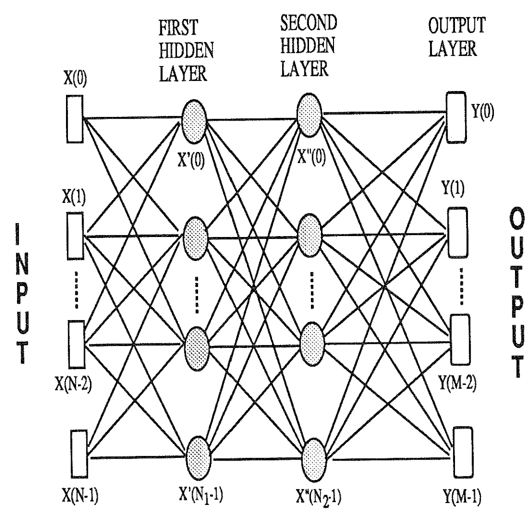


Figure 1: Three-Layer Perceptron with N Inputs and M Outputs

2 Neural model

The block diagram of the neural net is shown in figure 1. $a_j^{'}$ and $a_i^{''}$ are internal offsets in the nodes in the two hidden layers respectively, and a_k is the offset in the node in input layer. The output function is F(θ) (SIGMOID Function):

$$F(\theta) = \frac{1}{1+e^{-\theta}}$$

The relationships of the layers are as follows:

$$Y_i = F(\sum_{j=0}^{N_2-1} X_j^{''} W_{ji}^{''} - a_i^{''}) \quad (1)$$

$$X_j^{''} = F(\sum_{k=0}^{N_1-1} X_k^{'} W_{kj}^{'} - a_j^{'}) \quad (2)$$

$$X_k^{'} = F(\sum_{l=0}^{N-1} X_l W_{lk} - a_k) \quad (3)$$

where $0 \leq$ i $\leq$ M - 1, $0 \leq$ j $\leq N_2$ - 1
and $0 \leq$ k $\leq N_1$ - 1, $0 \leq$ l $\leq$ N - 1
The output error at the output layer is defined as follow:

$$E_s^{bp} = \frac{1}{2} \sum_{i=0}^{M-1} (O_{si} - Y_{si})^2 \quad (4)$$

Where s denotes the No.s type of input/output patterns and O_s represents the desired output for input pattern s. Differentiate the equation we obtain:

$$\Delta w_{ij}(t+1) = \eta \delta_j x_i^{'} + \alpha \Delta w_{ij}(t) \quad (5)$$

where η and α $(0 \leq \alpha, \eta \leq 1)$ represent gain term and momentum term respectively. The error term δ_j for node j is:

$$\delta_j = \begin{cases} Y_j(1-Y_j)(O_j - Y_j) & \text{j is an output node} \\ x_j^{'}(1-x_j^{'}) \sum_k \delta_k w_{jk} & \text{otherwise} \end{cases} \quad (6)$$

3 System Organization

In order to improve network performance and reduce the occurrence of local minima, we introduced extra hidden units (four times more than the input units), lowered the gain term used to adapt weights, and made many training runs starting with different sets of random weights. Selection of weights w is an optimization problem with many local minima. The network performance stemming from different initial point differs from each other. Different weights correspond to different ways of forming generalizations about the patterns inherent in the training set. Since each network makes generalization error on different subset of the input space, the collective decision produced by the ensemble is less likely to be in error than the decision made by any of the individual network.

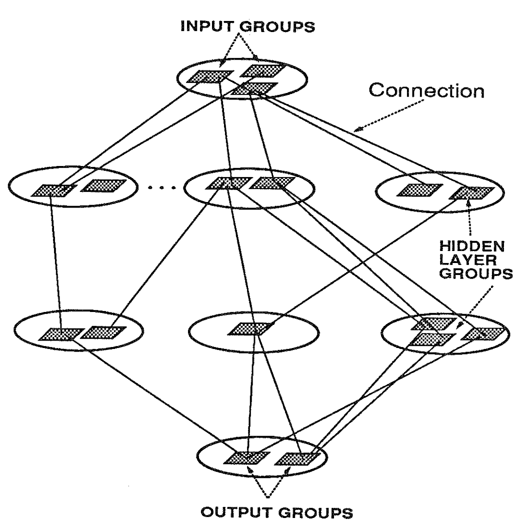


Figure 2: The Block Diagram of Individule Neural Network

From figure 2 we can also see that the input layer, output layer and the hidden layers are furthur divided into many groups. There are no weight connection between the groups on the same layer and within each group. The relationship (ON and OFF state) between groups is able to be managed programmingly. State ON denotes there exist connection weights between the two groups, and state OFF means there are no such connection weights. The neural model we used has the following five distinctive features:

1: It's hierarchical constructure enabled multi-stage image processing;

2: The units in each layer are divided into several groups. The relationship between each group is programmable, and different kind of image texture can be emphasized within each group;

3: The input units of each unit are limited programmingly. Only the near neighbourhood of input unit has effect on the unit, so that the processing uses only the neighbourhood information;

4: The connection between units is invariant to position. Therefore, the weights and offsets in sigmod function are able to be modified through training;

5: Multiple trained neural networks are employed for optimizing network parameters and avoiding local minima.

By the above five major considerations in our neural perceptron designing, not only the local image texture is used, but the globle image information can be emphasized during the training stage and practical application.

Figure 3: Original Image LENNA

4 Binarization

Let the pixel graylevel be integer set [0,M] $\subset$ GL(M: corresponding to the brightest pixel), N be an integer, and f: N $\times$ N be the image function of image N $\times$ N. The binarization is to find out the appropriate threshold value T ⊂ GL so that the visually satisfied result image can be obtained after blacking pixels which gravlevels below the selected threshold T.

$f : \mathrm{N} \times \mathrm{N} \rightarrow \mathrm{B} \subset [0,1]$

$$f\ (x,y) = \begin{cases} 1 & \text{if } GL > T \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

After obtaining the histogram of original image, we furthur derive from H(j) in figure 5 to get the normalized histogram.

$$h(j) = H(j)/max(H(GL))(0 \leq j \leq N) \tag{8}$$

For the purpose of simplying the teaching procedure, we enhanced the contrast of images used by the following linear trasformation:

$$g(x,y) = \begin{cases} \frac{n-m}{b-a}(f(x,y) - a) + m & \text{if } a \leq f(x,y) \leq b \\ m & \text{if } f(x,y) < a \\ n & \text{if } f(x,y) > b \end{cases} \tag{9}$$

where a ≤ H[f(x,y)] ≤ b, m ≤ H[g(x,y)] ≤ n.

Many methods of image binarization have been proposed since 1970, such as entropy method, minimum error method, analysis method, and mean threshold method [4, 6] etc. Most of these methods first use the statistical parameters of the image (mean, variance, entropy, etc.) to formulate a formula, and then maximise/ninimise the formula to obtain their threshold. The above statistical methods, simple to calculate and practical to many application cases, can't always guarantee good threshold in the viewpoint of human vision.

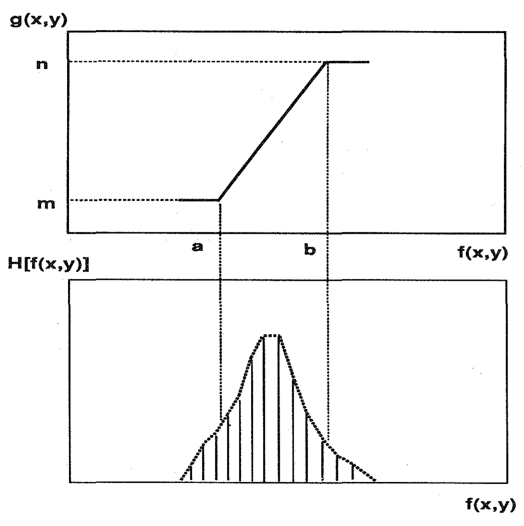


Figure 4: Transformation according to graylevel

Definition: The image binarization transformation is to effectively divide the image objects to background, and let the run-length coding (RLC) data of the binarized image be maximum.

Here, we use the RLC code E^{rl} as secondary criterion to prevent blurring image when restorating the binarized image. RLC coding for a binarized image is in the form $E^{rl} = \sum_{i=1}^{I} Y_i \sum_{j=0}^{J_i} X_{i(2j)} X_{i(2j+1)}$. Where, $X_{i(2j)}$ and $X_{i(2j+1)}$ are the X coordinates of the front and rear cross-points of image object respectively, while scaning the binarized image

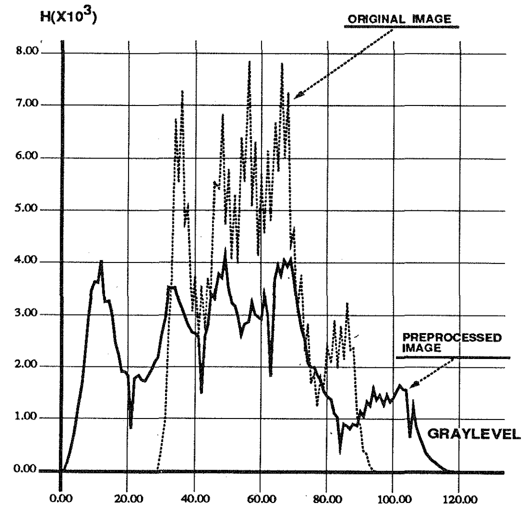


Figure 5: The Histograms

from top to bottom. Y_i is the corresponding Y coordinate. Our experiment showed that among the binarized images using threshold between the visually accepting range, the binarized image with maximum RLC code usually preserves more object details of the original image, and guarantees to give more satisfied restoration image in the viewpoint of human vision.

Considering the RLC code factor, the final formula to be minimized is E = E^{bp} - E^{rl}, leading to a modified back propagation algorithm.

Figure 6: The Pre-Processed Image

5 Implementation

The images (including sample images and test images) we used are standard potriate images and NOAA images. In order to obtain the visually satisfied threshold, we first calculate the histgram for each sample image, and then scale the histgram between [0,1]. Afterwards we furthur divide the obtained histogram into several groups to feed the input layer of each neural net. A three-layer perceptron is employed in our experiment, and modified back propagation learning algorithm is utilized. The sizes of images used are 512×480 and 512×512, and the graylevel range is between [0,127]. The units of input layer of the neural net is 128, the units numbers of the two hidden layers are 256 and 512 respectively, and that of the output layer is 128. During the training, we let all the normalized teaching threshold's neighborhold [-0.02,0.02] of output layer be ON, and at the same time slightly increase and decrease the threshold. Therefore, the order number of the unit with largest value at the output layer represents the corresponding threshold. We also trained the offsets in sigmod functions using the same back propagation algorithm and initialized the weights by hopfield network method to speed the learning procedure. We also created several kinds of exemplar patterns (cross, T-shape, door-shape, and X-shape etc.) and each of them isn't limited necessarily by not sharing many common bits with other exemplar patterns for the stability of output and pattern recognition. The above simulation was the ignition of our neural model for image binarization, and enhanced the hope of expanding neural model to boundary detection and image classification/recognition.

Table 1. Comparision Between Neural Model and Other Methods

image	a	b	c	d	e
akiho	76	19	48	27	43
auto	95	69	57	33	64
city	72	52	48	23	49
desk	80	32	73	4	47
girl	45	38	69	13	41
home1	60	55	52	5	43
home	85	58	57	22	56
noaa1	50	33	81	16	45
noaa2	49	29	62	3	36
lenna	41	32	81	15	42

a : neural method
b : analysis method
c : entropy method
d : minimum error method
e : mean thresholdmethod

Table 2. RLC Comparision
(the code of a is assumed as 1)

image	a	b	c	d	e
akiho	1	2.90	0.751	1.328	1.141
auto	1	0.766	0.433	0.259	0.611
city	1	0.668	0.554	0.341	0.597
desk	1	1.367	0.967	0.309	0.972
girl	1	0.756	0.801	0.296	0.831
home1	1	1.020	1.047	0.389	0.759
home	1	0.901	0.901	0.449	0.516
noaa1	1	0.840	0.806	0.146	0.956
noaa2	1	1.023	0.981	0.177	0.917
lenna	1	1.148	0.509	0.904	1.012

The comparisons of our method to some other methods mentioned are shown in table 1-2 and in figure 9-10.
The training (providing sample output only once during each training iteration) and testing seems that our method is simple and speedy way for image binarization and an economic way for image transformation.

6 Furthur Study

We are now trying to expand this method to boundary detection, image recognition/classification and other image processings. For image recognition, the most encounted problem is the detection of different kinds of invariances. The existing method of invariance transformation using neural neawork is extracting the invariances first, and then take the invariances as the inputs of the neural network to preform some transformation. We are now constructing such an integrated artificial perceptron which embedded the mechanism of invariance detection in the perceptron itself, and the only needed input is the original image. Necessary preprocessing, binarization, edge/boundary detection, invariance detection and image tranformation will be performed by the neural perceptron in a whole.

Figure 7: Binarized Output Image

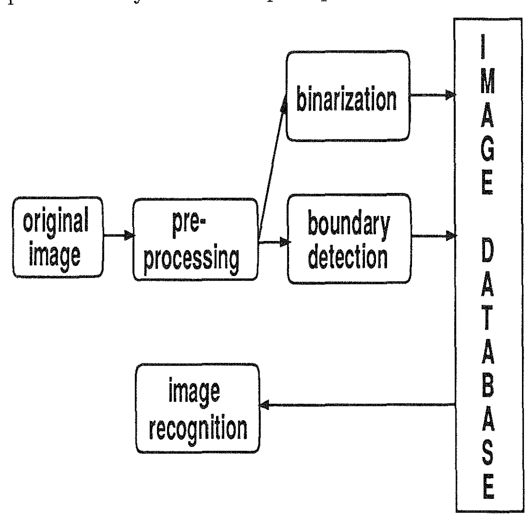


Figure 8: Integrated Perceptron

7 Conclusion and remark

The experiment implemented on our neural network has proved that the neural model we used is capable to be employed to different image textures and application domains. The comparisons show that the neural model we used for image binarization is effective no matter how the shapes of the image histograms appear and what the image texture will be.

References

[1] D.E.Rumelhart & J.L.Mcclelland, Parallel Distributed Processing, MIT Press, Vol.1,2, 1989

[2] R.P.Lippmann, An Introduction to Computing With Neural Nets, IEEE ASSP Magzine, April 1987

[3] S.I.Amari, Proceedings of IEEE, Vol 78, No.9, p1443-1463, September 1990

[4] Babaguchi, Medical Imaging Technology, Vol.9, No.4, September 1991

[5] Proceedings of IEEE, Vol 78, No.9-10, p1443-1463, September 1990

[6] P.K.Sahoo, S.Soltani, A.K.C.Wong, and Y.C.Chen, A Survey of Thresholding techniques, Computer Vision, Graphics and Image Processing, 41(2)233-260, February 1988

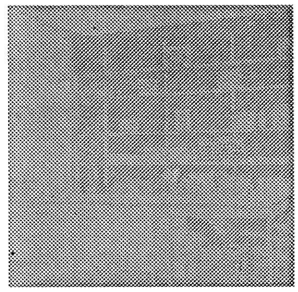

(analysis method)

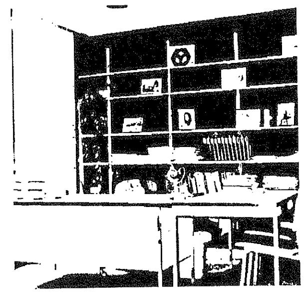

(minimum error method)

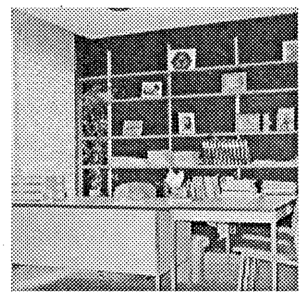

(entropy method)

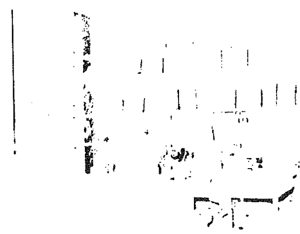

(mean threshold)

Figure 9: The Binarization Results of Image LENNA

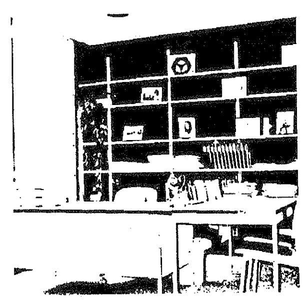

(original image)


(neural net method)


(preprocessed image)


(analysis method)


(minimum error method)


(entropy method)

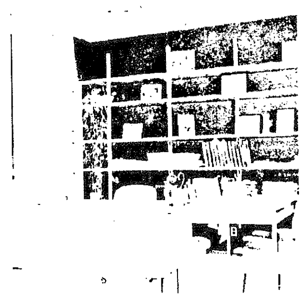

(mean threshold)

Figure 10: The Binarization Results of Image ROOM