

신경망 회로를 이용한 필기체 숫자 인식에 관한 연구

이규한, 정진현
 광운대학교 제어계측공학과

A Study Of Handwritten Digit Recognition By Neural Network Trained With The Back-Propagation Algorithm Using Generalized Delta Rule

Kye Han Lee, Chin Hyun Chung
 Dept.of Control & instrumentation Engineering, Kwangwoon Univ

Abstract - In this paper, a scheme for recognition of handwritten digits using a multilayer neural network trained with the back-propagation algorithm using generalized delta rule is proposed.

The neural network is trained with hand written digit data of different writers and different styles. One of the purpose of the work with neural networks is the minimization of the mean square error(MSE) between actual output and desired one. The back-propagation algorithm is an efficient and very classical method. The back-propagation algorithm for training the weights in a multilayer net uses the steepest descent minimization procedure and the sigmoid threshold function.

As an error rate is reduced, recognition rate is improved. Therefore we propose a method that is reduced an error rate.

1. Introduction

Although the recognition of handwritten digits has been studied for more than three decades and many classifiers with high accuracy rates have been developed, no single "best" method exists yet. Consequently, the research in this area continues with the aim of improving the recognition rates further.

The application field is very wide, for example postal code recognition and numbers written on bank check. We can find the patterns shifted, scaled, distorted with some skew.

The proposed method in this paper is based in a multilayer neural network trained with the back-propagation algorithm using generalized delta rule. Multilayer enables the network to learn complex tasks by extracting progressively more meaningful features from the input patterns. We use the steepest descent minimization procedure to minimize the mean square error between actual output and desired one.

Finally, we present a method that has the minimum error value.

2. The Main Subject

2.1 The Preprocessing

For a given binary image containing a single numeral, there are preprocessing procedures performed prior to feature extraction. First, the

geometrical centroid of the digit is computed. This removes any differences due to location of the digit within the image.

The second preprocessing procedure is a stroke width normalization through skeletonization.

The image needs to be thinned in order to remove the effect of differences in stroke width. Skeletonizing algorithm is to iteratively delete edge points of a region subject to the constraints that deletion of these points does not remove end points, does not break connectedness, and does not cause excessive erosion of the region.

2.2 The Skeletonizing Algorithm

Region points are assumed to have value 1 and background points to have value 0. the method consists of successive passes of two basic steps applied to the contour points of the given region, where a contour point is any pixel with value 1 and having at least one 8 neighbor valued 0. With reference to the 8 neighborhood definition shown in Figure 1.

Step 1 flags a contour point p for deletion if the following conditions are satisfied.

(1) $2 \leq N(p_1) \leq 6$

(2) $S(p_1) = 1$

(3) $p_2 \cdot p_4 \cdot p_6 = 0$,

(4) $p_4 \cdot p_6 \cdot p_8 = 0$.

where $N(p_1)$ is the number of nonzero neighbors of p_1 , that is,

$$N(p_1) = p_2 + p_3 + \dots + p_8 + p_9$$

and $S(p_1)$ is the number of 0-1 transitions in the ordered sequence of $p_2, p_3, \dots, p_8, p_9, p_2$

In step 2, condition (1) and (2) remain the same, but conditions (3) and (4) are changed to

(3') $p_2 \cdot p_4 \cdot p_8 = 0$,

(4') $p_2 \cdot p_6 \cdot p_8 = 0$.

Thus one iteration of the skeletonizing algorithm consists of (1) applying step 1 to flag border points for deletion, (2) deleting the flagged points, (3) applying step 2 to flag the remaining border points for deletion, (4) deleting the flagged points. This basic procedure is applied iteratively until no further

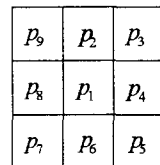


Figure 1. Neighborhood arrangement used by the skeletonizing algorithm

points are deleted, at which time the algorithm terminates, yielding the skeleton of the region.

In step 1, a point that satisfies (1)-(4) conditions is an east or south boundary point or a northwest corner point in the boundary. In either case, p_1 is not part of the skeleton and should be removed. In step 2, a point that satisfies four conditions is an north or west boundary point, or a southeast corner point

2.3 The Neural Network

Neural network has seen an explosion of interest since their rediscovery as a pattern recognition paradigm in the early 1980's. The value of some of the applications for which they are used may be arguable, but there is no doubt that they represent a tool of great value in various areas generally regarded as difficult, particularly speech and visual pattern recognition.

Most neural approaches are based on combinations of elementary processors (neurons), each of which takes a number of inputs and generates a single output. Associated with each input is a weight, and the output is then a function of the weighted sum of inputs. This output function may be discrete, depending on the variety of network in use. A simple neuron is shown in Figure 2.

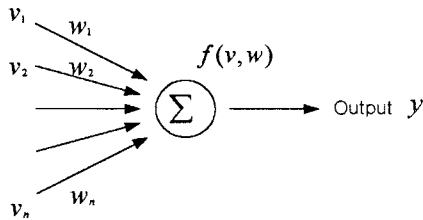


Figure 2. A Simple Neuron

The inputs are denoted by v_1, v_2, \dots , and the weights by w_1, w_2, \dots , then total input to the neuron is then

$$x = \sum_{i=1}^n v_i w_i$$

or, more generally,

$$x = \sum_{i=1}^n v_i w_i - \theta$$

where θ is a threshold associated with this neuron. Also associated with the neuron is a transfer function $f(x)$ which provides the output. Common examples are

$$f(x) = \begin{cases} 0 & \text{if } x \leq 0 \\ 1 & \text{if } x > 0 \end{cases}$$

$$f(x) = \frac{1}{1 + e^{-x}}$$

2.4 The Multilayer Perceptron.

Typically, the network consists of a set of source nodes that constitute the input layer, one or more hidden layers of computation nodes, and an output layer of computation

nodes. The input signal propagation through the network in a forward direction, on a layer-by-layer basis. These neural networks are commonly referred to as multilayer perceptrons (MLPs). A four-layer neural net is shown in Figure 3.

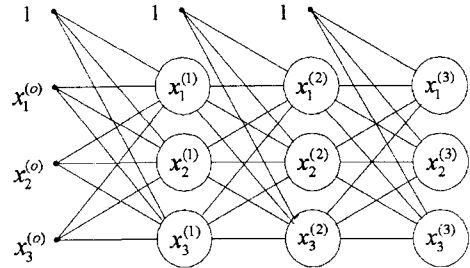


Figure 3. A Four-Layer Neural Net

In general, a multilayer net has $K+1$ layers of nodes, denoted $0, 1, \dots, K$ as shown Figure 3. The output of the i th node in layer k is denoted $x_i^{(k)}$. This is the value obtained after computing the weighted sum of the inputs and applying the threshold or other function. The nodes in layer K are called the output nodes, and those in layers 1 through $K-1$ are called the hidden nodes.

A multilayer perceptron has two distinctive characteristics. First, The model of each neuron in the network includes a nonlinear activation function. The presence of nonlinearities is important because otherwise the input-output relation of the network could be reduced to that of a single-layer perceptron. Second, The network contains one or more layers of hidden neurons that are not part of the input or output of the network. These hidden layers enable the network to learn complex tasks by extracting progressively more meaningful features from the input patterns.

3. The Back-propagation Algorithm Using Generalized Delta Rule.

3.1 The Back-propagation Algorithm.

The back-propagation algorithm for training the weights in a multilayer net uses the steepest descent minimization procedure and the sigmoid threshold function.

The back-propagation algorithm consists of two main steps: a feed-forward step in which the outputs of the nodes are computed starting at layer 1 and working forward to the output layer K , and a back-propagation step where the weights are updated in an attempt to get better agreement between the observed outputs and the desired outputs.

The feed-forward step begins at layer 1 and works forward to layer K . The back-propagation step begins at layer K and works backward to layer 1.

(Back-Propagation Algorithm)

1. Initialize the weights $x_{ij}^{(k)}$ to small random values, and choose a positive constant c .
2. Repeatedly set $x_1^{(K)}, \dots, x_{M_0}^{(K)}$ equal to the features of samples 1 to N , cycling back to sample 1 after sample N is reached.
3. Feed-forward step.
For $k=0, \dots, K-1$, compute

$$x_j^{(k+1)} = R\left(\sum_{i=0}^{M_k} w_{ij}^{(k+1)} x_i^{(k)}\right)$$

for nodes $j=1, \dots, M_{k+1}$. We use the sigmoid threshold function $R(s) = \frac{1}{1 + e^{-s}}$

4. Back-propagation step.
For the nodes in the output layer, $j=1, \dots, M_k$ compute

$$\delta_j^{(K)} = x_j^{(K)}(1 - x_j^{(K)})(x_j^{(K)} - d_j)$$

For layers $k=K-1, \dots, 1$ compute

$$\delta_i^{(k)} = x_i^{(k)}(1 - x_i^{(k)}) \sum_{j=1}^{M_{k+1}} \delta_j^{(k+1)} w_{ij}^{(k+1)}$$

for $i=1, \dots, M_k$

5. Replace $w_{ij}^{(k)}$ by $w_{ij}^{(k)} - c\delta_j^{(k)}x_i^{(k-1)}$ for all i, j, k .
6. Repeat steps 2 to 5 until the weights $w_{ij}^{(k)}$ cease to change significantly.

3.2 The Back-propagation Algorithm Using Generalized Delta Rule

If the learning rate parameter c is chosen too small, progress toward the minimum will be so slow that a huge number of iterations will be required to obtain the solution. On the other hand, if the learning rate parameter c is chosen too large in order to speed up the rate of learning, the procedure may repeatedly overshoot the minimum. With c too large, the procedure may even cycle between two or more local minima, and fail to converge. The method of increasing the rate of learning avoiding the danger of instability is to introduce the delta rule including a momentum term.

$$\Delta w_{ij}^{(k)}(n) = \alpha \Delta w_{ij}^{(k)}(n-1) + c\delta_j^{(k)}(n)x_i^{(k-1)}(n)$$

where α is usually a positive number called the momentum constant, $0 \leq |\alpha| < 1$, and n is the iteration number.

The inclusion of momentum in the back-propagation algorithm tends to accelerate descent in steady downhill directions and has a stabilizing effect in directions that oscillate in sign.

The incorporation of momentum in the back-propagation algorithm represents a minor modification to the weight update, yet it may have some beneficial effects on the learning behavior of the algorithm. The momentum term may also have the benefit of preventing the learning process from terminating in a shallow local minimum.

When the delta rule including a momentum

term is introduced, in back-propagation algorithm, step 1 and step 5 are modified as following.

1. Initialize the weights $w_{ij}^{(k)}$ and former weight $v_{ij}^{(k)}$ to small random values, and choose a positive constant c .
5. Replace $w_{ij}^{(k)}$ by $w_{ij}^{(k)} - \alpha v_{ij}^{(k)} - c\delta_j^{(k)}x_i^{(k-1)}$ and store $w_{ij}^{(k)}$ into $v_{ij}^{(k)}$.

4. Experiment And Conclusions

The performance of the proposed system is tested on real data. The data contain isolated, binary numeral images collect from 10 writers.

We compare the classical back-propagation algorithm with the back-propagation algorithm using generalized delta rule through the experimental results.

The use of the generalized delta rule in the back-propagation algorithm prevents the learning process from terminating in a shallow local minimum.

(참 고 문 헌)

- [1] Richard Johnsonbaugh, Steve Jost, "Pattern Recognition and Image Analysis", Prentice Hall, 1996.
- [2] Milan Sonka, Vaclav Hlavac, Roger Boyle, "Image Processin, Analysis, and Machine Vision", International Thomson Publishing, 1999.
- [3] Simon Hayakin, "Neural Networks - A Comprehensive Foundation", Prentice Hall, 1999.
- [4] F. Kimura, S. Inoue, T. Wakabayashi, Y. Miyake, "Handwritten Numeral Recognition using autoassociative neural network", IEEE trans, 1998.
- [5] A. Khotanzad, C. Chung, "Hand Written Digit Recognition using Combination Of Neural Network Classifiers", IEEE trans, 1998.
- [6] J. Cao, M. Ahmadi, M. Shridhar, "A Hierarchical Neural Network Architecture For Handwritten Numeral Recognition" Pattern Recognition Society, Pattern Recognition Vol.30, NO.2, pp 289 -294 1997.
- [7] Y.J. Kim, S.W. Lee, "Off-line Recognition Of Unconstrained Handwritten Numerals Using Multilayer Cluster Neural Network Combined with Genetic Algorithm", Korea Information Science Society, Vol.21, No.8, pp1468-1479, 1994.