Segmentation and Recognition of Hand-Written Digits Using OSSA Neural Network

KYUNGHEE LEE
Dept. of Information and Communications, PyeongTaek University
111, YongI-Dong, PyeongTaek-Si, KyungKi-Do, KOREA

Abstract: - This paper proposes a neural network approach to solve the hand-written digit recognition problem. The proposed approach is based on the graph matching, a form of elastic matching, and an one-variable stochastic simulated annealing(OSSA) algorithm which makes it possible to evaluate the spin average value effectively by Markov process in case of many real applications with continues variables. Our approach provides not only the function of recognition but also the segmentation ability such that input characters are correctly recognized and segmented even if they are touching, connected, and defected by noise. Some preliminary computer experiments are reported to show the feasibility of this approach.

Key-Words: - Pattern recognition, Segmentation, Optimization, Simulated Annealing, Neural networks

1 Introduction

Hand-written character recognition is an important application field of neural networks because the conventional algorithms have much difficulties in this area[1][2][3]. Especially, it is hard to segment characters correctly by conventional, rule-based segmentation algorithms if they are touching, defected or noisy. In general, most recognition models used a preprocessor to remove meaningless variations and to capture meaningful features. However, its recognition accuracy has been limited by the accuracy of the segmentation algorithm it has. One of the well known problems in this situation is that one can not properly segment a character until it is recognized and yet one can not properly recognize a character until it is segmented correctly. This means that high ranks of recognition can be achieved by the integration of segmentation and recognition occurring simultaneously in the system.

Combinatorial optimization ranks among the first application of modern neural networks. In this paper, we propose a neural network approach that simultaneously segments and recognizes the hand-printed digits by graph matching, which is formulated as one of optimization problems. And it is shown that it is possible to map the graph matching problem onto a one-variable annealing neural network with an appropriate energy function.

2 Graph Matching for Segmentation and Recognition

2.1 Model Parameters

To characterize each feature three measurements are extracted. Their measurements or model parameters were chosen for their properties of invariance with respect to size, translation including shift of the writing[4]. We choose the following three model parameters to characterize the graph for segmentation and recognition. The first one is the distance, Euclidean distance, between the node in the input graph and that of reference graph, object graph. This parameter can be invariant to shift of the input. The second one is the acute angle between the two nodes in the graph. This angle parameter is invariant to rotation of the input by the proper angle measure function. The third one is the number of cross points between the two nodes in the graph. This crossing parameter is invariant to the size of the input.

2.2 Graph Matching Approach

Pattern recognition in our system consists of a dynamic assignment procedure, which is performed under the constraint that vertices(nodes) in the input graph should have approximately the same topological relationship as the vertices in one of the stored object graphs. In our graph matching approach each node has relational properties with neighboring nodes and is connected each other by a link. The relational properties are represented by the model parameters (distance, acute angle, and the number of cress points between the nodes). And not only the relations between the near neighboring nodes are used as the compatibility constraints but also relations between all

the far neighboring nodes are used in order to increase the robustness of the matching. The graph matching we present in this paper is elastic matching based on an energy of cost function. The minima of which provide the solution of the matching problem.

This leads to considering a neural network system where the state of the system is a set of connectivity rather than a vector of neural activities and computation is a connectivity-dynamics instead of an activity-dynamics[6]. In this approach, matching one graph with another graph consists in finding a connectivity state which satisfies at best many local requirements and minimize the cost.

2.3 Cost Function

In the graph matching problem for pattern recognition and segmentation we formalize, and actually quantify, with the help of the best possible matching between input and object graph, the degree of matching by a mathematical formulation. To find out the best-matching we characterized the following energy function to be minimized, which measures the quality of node-to-node matching as well as conservation of neighboring relations between nodes in the input and object graph.

$$\min E = \lambda_1 E_1 + \lambda_2 E_2 + \lambda_3 E_3 \tag{1}$$

$$E_1 = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1,\, j\in N(i)}^{N}\sum_{l=1,\, l\neq k}^{N}(\Delta\#_{ikjl})V_{ik}V_{jl} \tag{2}$$

$$E_2 = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1,\, j\in N(i)}^{N}\sum_{l=1,\, l\neq k}^{N}\begin{cases}(\Delta\theta_{ikjl})V_{ik}V_{jl}, \text{ if } \Delta\theta_{ikjl} \le \theta_{th}, \\ (\alpha\,\Delta\theta_{ikjl})V_{ik}V_{jl}, \text{ if } \Delta\theta_{ikjl} \rangle \theta_{th}\end{cases} \tag{3}$$

$$E_3 = \sum_{i=1}^{N}\sum_{k=1}^{N}\sum_{j=1,\, j\in N(i)}^{N}\sum_{l=1,\, l\neq k}^{N}\begin{cases}0, \text{ if } \min_d(i,j) \le \Delta\delta_{ikjl} \prec mad_d(i,j) \\ \beta((\Delta\delta_{ikjl} - (\dfrac{\max_d + \min_d}{2}))^2 - \\ \qquad (\dfrac{\max_d - \min_d}{2})^2), \quad \text{otherwise}\end{cases} \tag{4}$$

$$\Delta\#_{ikjl} = \left|\#^I_{ikjl} - \#^O_{iK(i)jL(j)}\right| \tag{5}$$

$$\Delta\theta_{ikjl} = \min\left\{\left|\theta^I_{ikjl} - \theta^O_{iK(i)jL(j)}\right|, \left|2\pi - (\theta^I_{ikjl} - \theta^O_{iK(i)jL(j)})\right|\right\} \tag{6}$$

$$\Delta\delta_{ikjl} = \left|\delta^I_{ikjl} - \delta^O_{iK(i)jL(j)}\right| \tag{7}$$

Where V_{ik} is analogous to a state variable of a neuron, and denotes the probability that node i is placed at position k. Therefore, it takes the value of 1 when the node i in the input graph is placed at position k and it is matched perfectly to the node i in the object graph. λ_1, λ_2, λ_3, are weight factors that determine the relative weightings of the terms. $\#_{ikjl}$ is the number of cross points between the node i and node j when node i is at position k and node j is at position l. θ_{ikjl} is the acute angle between the interconnecting edge and horizontal line. δ_{ikjl} is a distance between node i and node j when node i is at position k and node j is at position l. min_d max_d denote minimum and maximum distance between node i and node j, that are allowed to move without penalty, respectively and that are needed to assist to produce a fine-look matching graph. $N(i)$, a set of neighboring nodes of i, consists of near neighbors and far neighbors. Near neighbors are upper, lower, left, right, upper-left, upper-right, lower-left, and lower-right side nodes around the node i within a unit distance. $K(i)$ and $L(j)$ are positions, that are decided when the object graph is constructed, of node i and node j in the object graph, respectively. θ_{th} is a threshold value of angle difference, Upper indices I and O refer to the input and stored object graph, respectively.

α is a penalty factor to prevent angle difference between tow nodes not to exceed the θ_{th}. β is another penalty factor to make the node i and node j reside in a given range between min_d and max_d. The λ_1 term is to be zero if the number of cross points between every two edges in the input graph is same as that in the object graph. The λ_2 and λ_3 terms will have minimum value when all the neighboring nodes have the same topology in view of the angle and distance similarity between input and object graph.

3 One-variable Stochastic Simulated Annealing

In the real world application, there are many NP-hard optimizing problems with continuous variables. These include quantification analysis, D-optimal design problem, and fuzzy clustering[5][7]. In the case of neural network with continuous variables of [a,b], the situation is identical to that of the network with discrete variables except that the state of each spin has a continues value, real value. And the equilibrium spin average of the continuous network is described as the following integral equation[5].

$$< s_i > = \frac{\int_a^b s_i \exp^{-H_s/T}\, ds_i}{\int_a^b \exp^{-H_s/T}\, ds_i} \tag{8}$$

Though Eq(8) could be calculated by the integration method there are still some weaknesses. For instance,

- It is time consuming to calculate the integration in case of the large state space.

- In many cases, a floating point under-flow error might occur by the hardware limitation of the computer when the component of the integral goes to zero

To overcome the above weakness we presents the way how to approximate Eq(8) by the stochastic method using Markov process. Monte Carlo(MC) techniques are methods of estimating the values of many-dimensional integrals by sampling with the help of random numbers and regarded as the methods appropriate to equilibrium statistical mechanics. We will concern ourselves with estimating the average potential energy of a simple fluid system, where the potential energy is dependent on the configuration variables $s^N = (s_1,\ldots,s_N)$ of the N particles:

$$< E > = \int E(s^N) p(s^N) ds^N \quad (9)$$

where the probability density $p(s^N)$ is given by

$$p(s^N) = \frac{1}{Z} \exp^{\frac{-E(s^N)}{k_B T}} \quad (10)$$

with Z the configuration integral

$$Z = \int \exp^{\frac{-E(s^N)}{k_B T}} ds^N \quad (11)$$

In MC estimation of an average like Eq(9), random numbers are used to generate approximately distributed configuration(s^N) of a system of *N* particles. In practice the computations are fairly expensive to carry out the integral, and it is impossible to calculate the configuration integral, Eq(11), because of the unlimited integral space. Importance Sampling(IS) is a promising method for reducing run-time in MC which has long been recognized as a powerful tool for simulating low probability events. And most of the statistical mechanics applications of MC involve IS rather than independent samples. In the mean field annealing(MFA) neural network with continuous variables, the equilibrium spin average is the same form as the average potential energy with configuration integral except that the system consists of only one variable. And the average of the perturbed spin at a given temperature might be regarded as an expected value of that spin in the mean field from the Boltzmann distribution[5].

We propose a new stochastic algorithm, OSSA algorithm, to realize Eq(8) as the following:

1. Select a spin *i* in the current state *s*, and perturb it into a new state *s'*.
2. Compute the energy E(*s'*) and compare it with the E(*s*) of the current state *s*, and then let the spin *i* take the perturbed value with probability

$$\Pr(s' \leftarrow s) = 1.0, \quad if\ E(s') < E(s)$$
$$= \exp^{\frac{-(E(s')-E(s))}{T}} \quad if\ E(s') \geq E(s) \quad (12)$$

   Where T is a temperature.
3. Repeat 1 and 2 a number of times
4. Calculate the average of the accepted perturbed values and regard it as an equilibrium spin average, $<s_i>$, at a given temperature T.
5. Anneal with an annealing schedule.
6. Repeat step 1 to 5 until the final temperature is reached.

In the above algorithm, step 1 to 3 are the same steps as in the original simulated annealing(SA) algorithm except that each single spin is perturbed and evaluated as its own value, which represents as element of the configuration vector. In other words, the difference between SA and OSSA is that the perturbed state of all the *n* spins is regarded as a candidate solution in the SA, but in OSSA the average of the perturbed states of only one spin is regarded as an equilibrium average of that spin under the field of the rest of the system. Step 4 is not necessary but is recommended to be used because it can prevent a spin from being evaluated as a fluctuated value even though a small number of iteration is used in OSSA algorithm. By the above OSSA algorithm, a good approximation of Eq(8) can be obtained.

4 Segmentation and Recognition

4.1 Graph Multi-partitioning

A graph partitioning problem is to partition a graph, which consists of a set of *N* nodes and *E* interconnecting edges, into *K* equally sized sub-graphs, each with *N/K* nodes and minimal number of edges. And graph multi-partitioning is a special problem with nodes that must be reside only one of *B* bins in the graph. Our graph matching problem is similar to the graph multi-partitioning in that it has exclusivity constraints that in the final solution a given node n_i must reside in only one bin among the *B* bins. In our case B corresponds to the number of possible positions at which a node can reside.

4.1 Segmentation and Recognition Using OSSA

Our segmentation and recognition algorithm using OSSA and graph matching is as follows:

1. Select one of the object graphs in sequence and repeat step 2 to step 4 until all the object graphs are selected.
2. Initialize every node in the input graph to be positioned around the center of its window area.
3. Determine starting temperature as the following:
   (a) Perturb the node *i* in the current *s* into a new state *s'* with a starting temperature.
   (b) Let the node *i* take the perturbed value with probability according to Eq(12) and repeat this step a number of times. If the average of the accepted perturbed values has a small value, raise the starting temperature, otherwise lower the starting temperature.
   (c) Repeat step 3(a) to 3(b) until the average of the accepted perturbed values of every node has the value of near 1.0.
4. Apply OSSA algorithm and rank the cost at th final temperature as the cost of the graph matching with the selected object graph.
5. Output the graph matching result which has the minimum cost.

In step 1 and 2, one of the object graphs is selected and input graph is constructed from the input pattern with every node placed around the center of its window area as an initialization. In step 3, the temperature at witch node starts to stop the random movement is regarded as the starting temperature, This enables that OSSA algorithm finds the proper starting temperature empirically without a complicated estimation of the critical temperature. In step 4, optimization is performed by OSSA algorithm as described in section 3, and the final solution of segmentation and recognition is obtained.

5 Experimental Results

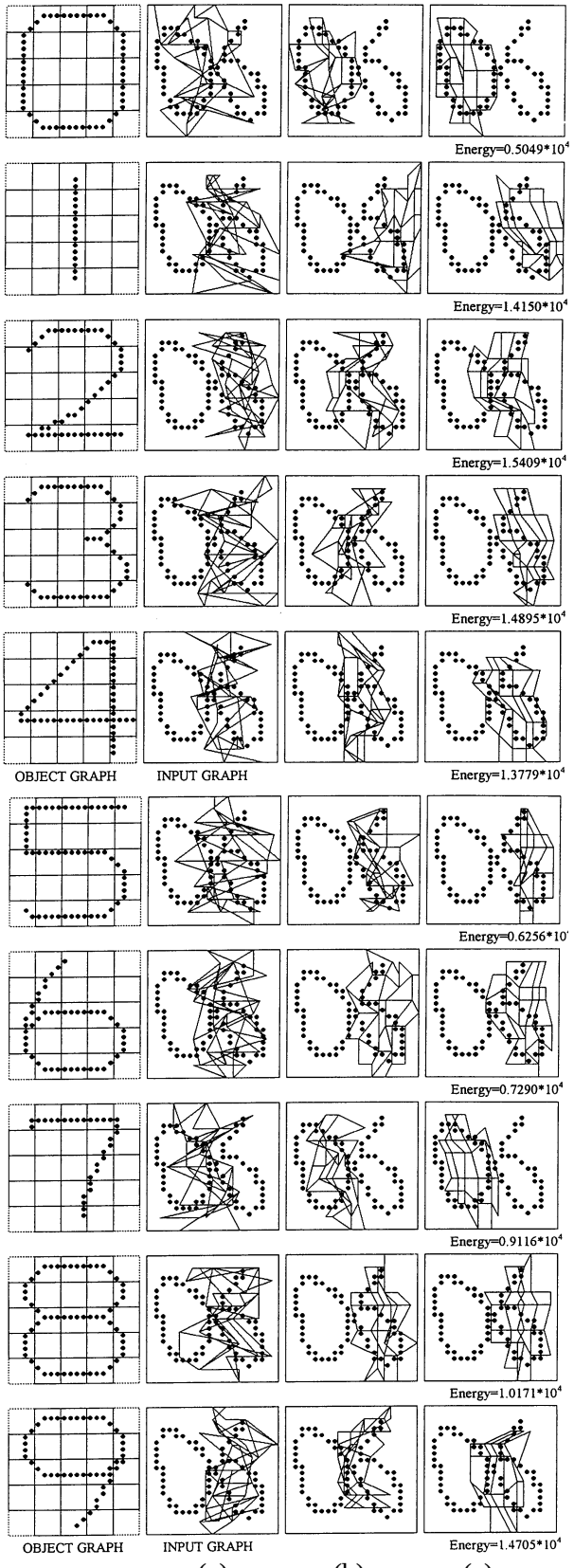
In the experiments, the input domain which consists of 20×20 cells is used. Each object graph with 32 nodes is made to have its own characteristics in the object domain having the same size as input domain. To find the near global minimum of the cost function, Eq(1), the solution of the matching problem. We implement an optimization algorithm, OSSA, for segmentation and recognition as described in the previous section. And we also use a window mechanism and a quench mechanism to speed up the computer simulation.

To investigate the ability of our model to simultaneously segment and recognize characters we use the input stimulus consisting of two digits that are distorted or touching each other. The segmentation of connected or touching characters is more difficult problem than recognition of a single character because there are many local topologies similar to that of object graph around the touching or connected area in the input graph. We applied graph matching and OSSA to segment hand-written digits with the starting temperature(T)=1.5, the final T=1.0*10^{-4}, number of bins(B)=21*21, α=2.0, β=2.0, θ_{th}=0.8, and ε=1.0*10^{-5}. The weight factors used were λ_1=1500.0, λ_2=250.0 and λ_3=250.0.

Fig 1. shows how the graph matching procedure for segmentation and recognition is progressed with T when a stimulus consisting of touching characters ('0','6') is presented. In this figure, it is shown that at the initial T disordered graphs are made(Fig. 1(a)) but as the T decreased refined graphs are outcome(Fig 1.(b)). And at the final temperature, the identification and segmentation is completed when the input character has the same or similar topological relation as that of object character and cost function defined in Eq(1) is minimized and best matching graphs are displayed(Fig 1.(c)). Fig 2 shows the cost at each temperature step in OSSA when the graph matching to the object graph of '0'(a) and '6'(b) are occurred.

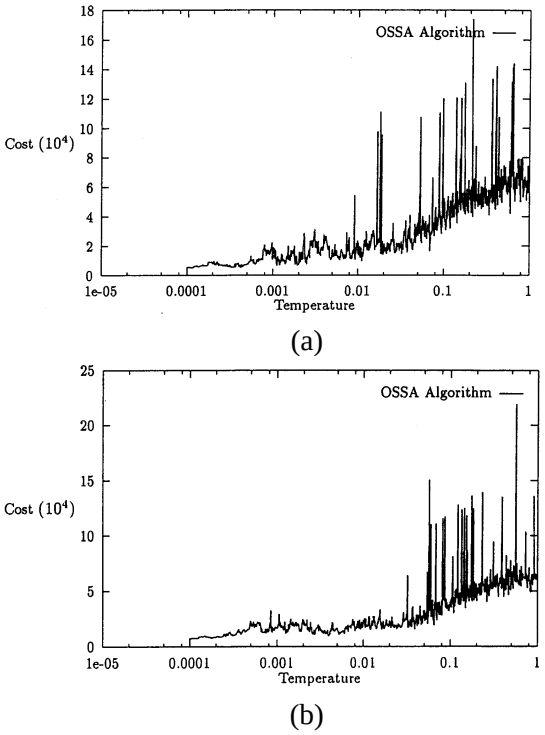
As a result, we can see that graph matching with object graph '0' and '6' produces the best matching that look like solutions of segmentation of hand-written input '06'. Fig 3. shows some examples of hand-written digits which have been successfully recognized and segmented by OSSA. It can be seen from the figure that input digits are correctly segmented and recognized without preprocessing such as noise reduction, thinning, and scaling even if they are touching, connected, and defected by noise, in which case most conventional systems can not handle well.

However, some problems still remain to be solved. For example, when a local topology in some part of input is similar to that of topology of the object, or the topology of the input graph can be matched with more than one object graphs there is a possibility of failure in recognition and segmentation. To overcome the problems we can use the object graphs with different numbers of nodes each other, that make it possible to represent the characteristics of the objects effectively according to their shapes.



(a) (b) (c)

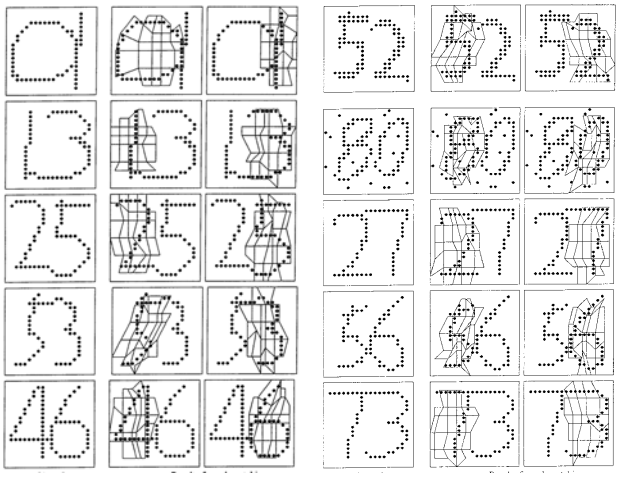
Fig 1. Progress in the graph matching process when input pattern is '06'



(a)

(b)

Fig 2. Cost at each temperature in OSSA (graph matching to the object graph of '0'(a) and '6'(b))



(a) (b) (a) (b)

Fig 3. Some examples of characters successfully segmented and recognized (left-side(b), right-side(c) output graphs)

6 Conclusions

The system presented here demonstrates that neural networks can, in fact, be used for segmentation as well as recognition. We have by no means demonstrated that this method is better than conventional segmentation and recognition system in over all performance. The major idea is that pattern recognition and segmentation problem often requires

invariance that can not be easily obtained in conventional neural network architecture and relational descriptions provides a powerful theoretical tool for solving this kind of problem which requires invariance with respect to various transformation of the image. In our approach, constraints such as the angle, cross points and distance between two nodes are used. There three types of constraints are simple to compute and additional constraints and/or high-order constraints may also be included.

References:

[1] Keeler,J.D., Rumelhart,D.E., Keow,W.K., Integrated Segmentation and Recognition of Hand-Printed Numerals, *Advanced Neural Information Processing System*, Vol.3, 1991, pp.557-563

[2] Lee,S., Pan,J.C., Handwritten Numeral Recognition based on Hierarchically Self-Organizing Learning Networks, *Proceedings of the International Joint Conference on Neural Networks*, Singapore, II, 1991, pp. 1313-1322

[3] Zhang, B., Fu, M., Yan, H., A Non Linear Neural Network Model of Mixture of Local Principal Component Analysis : Application to Handwritten Digits Recognition, *Pattern Recognition*, Vol.34, Issue 2, 2001, pp. 203-214

[4] Tappert,C.C., Cursive Script Recognition by Elastic Matching, *IBM Journal of Research and Development*, Vol 26(6), 1982, pp. 765-771

[5] Lee,K., Cho,K., Lee,W.D., Lee,S., Mean Field Annealing with Continuous Variables and Its Application to the Quantification Analysis Problem, *Proceedings of the International Joint Conference on Neural Networks*, 1992, pp. 431-435

[6] Von der Malsburg, C., Pattern Recognition by Labeled Graph Matching, *Neural Networks*, Vol.1, 1988, pp. 141-148

[7] Kim,M.H., Choi,H.S., Lee,W.D., Fuzzy Clustering Extended MFA for Continuous Valued State Space, *Proceedings of the International Joint Conference on Neural Networks*, 1992, pp. 733-738