

Implementation of the Relaxation Process by the Interactive Activation and Competition Network

Shiaw-Shian Yu
Advanced Technology Center
Computer and Communication Research Laboratories
Industrial Technology Research Institute
Chutung, Hsinchu, Taiwan 31015, Republic of China

Wen-Hsiang Tsai
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan 30050, Republic of China

Abstract

The relaxation process is a useful technique for using contextual information to reduce local ambiguity and achieve global consistency in various applications. It is basically a parallel execution model, adjusting the confidence measures of involved entities based on interrelated hypotheses and confidence measures. On the other hand, the neural network is a computational model with massively parallel execution capability. The output of each neuron depends mainly on the information provided by other neurons. Therefore, there exist certain common properties in the relaxation process and the neural network technique.

A mapping method that makes the interactive activation and competition network perform the relaxation process is proposed. By this method, the neural network technology can be easily adapted to solve the many problems which have already been solved by the relaxation process. Experimental results of solving a modified version of the n-queens problem on the proposed neural network are given to demonstrate the feasibility of the proposed method.

1 Introduction

The relaxation process is a general problem solving paradigm particularly suitable for problems involving contextual constraints. An advantage of the relaxation technique is its capability to use contextual information for reducing local ambiguity and increasing global consistency.

There are three types of relaxation processes, namely discrete relaxation, probabilistic relaxation, and fuzzy relaxation [1]. The one implemented by the neural network technique in this study is probabilistic relaxation.

First of all, we describe the general concept of probabilistic relaxation [1]. Suppose that a set of n objects O_1, O_2, \dots, O_n are to be classified into m classes C_1, C_2, \dots, C_m . Suppose further that the class assignments are correlated so that the contextual information conveyed by assigning object O_i into classes C_j and assigning object O_h into classes C_k can be represented by a compatibility coefficient $c(i, j; h, k)$.

The initial probability values for the objects to be assigned to the classes are obtained from certain *a priori* knowledge before the start of the relaxation process. Let $p_{ij}^{(0)}$ be the initial estimate of the probability that $O_i \in C_j$, $1 \leq i \leq n, 1 \leq j \leq m$; and $p_{ij}^{(r)}$ denote the estimate after the r th iteration during the relaxation process. It is required that $0 \leq p_{ij}^{(r)} \leq 1$ and $\sum_{j=1}^m p_{ij}^{(r)} = 1$ for all i and r to satisfy the probabilistic model. The goal of the relaxation process is to find a set of n classifications of all the objects which are as compatible as possible using the initial estimates $p_{ij}^{(0)}$ and the compatibility coefficients $c(i, j; h, k)$.

In each iteration of the relaxation process, the net contributions from the other objects, denoted as $q_{ij}^{(r)}$, to the probability value of assigning object i to class j at the r th iteration is calculated by

$$q_{ij}^{(r)} = \frac{1}{n-1} \sum_{h=1, h \neq i}^n \left(\sum_{k=1}^m c(i, j; h, k) p_{hk}^{(r)} \right). \quad (1)$$

And the new estimate of the probability value of assigning object i to class j at the $(r+1)$ th iteration is defined as:

$$p_{ij}^{(r+1)} = \frac{p_{ij}^{(r)}(1 + q_{ij}^{(r)})}{\sum_{j=1}^m p_{ij}^{(r)}(1 + q_{ij}^{(r)})}. \quad (2)$$

The above process is performed iteratively until the process converges or until a certain termination condition is satisfied.

On the other hand, neural networks are systems constructed to make use of some of the organizational principles that are felt to be used in the human brain.

From a parallel processing viewpoint, a neural network can also be regarded as a computational model with massively parallel execution capability [2, 3] forming an attractive model for efficient massively parallel machines with fine-grained distributed memory.

The Interactive activation and competition network (hereafter abbreviated as IAC network) model is proposed by McClelland and Rumelhart [4, 2, 3]. An IAC network consists of several neurons organized into a certain number of competitive pools. There are excitatory connections among the neurons in different pools and inhibitory connections among the neurons within a single pool. The IAC model assumes that the excitatory connections between the pools are bidirectional, thereby making the processing interactive in the sense that processing in each pool both influences and is influenced by processing in other pools. Within a pool, the inhibitory connections are usually assumed to run from each neuron in the pool to all other neurons in the pool. This mechanism implements a kind of competition among the neurons such that the neuron in the pool that receives the strongest activation tends to drive down the activation of the other neurons.

The neurons in an IAC network have continuous activation values between the maximum value and the minimum value. McClelland and Rumelhart set the output of each neuron to zero when its activation value is less than zero. This relationship can be described as follows:

$$V_i = \begin{cases} A_i & \text{if } A_i \geq 0; \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where A_i and V_i are the activation and output values of neuron i , respectively.

The net input u_i to neuron i is given by

$$u_i = \sum_j T_{ij} V_j + I_i \quad (4)$$

where the value V_j represents the output of the j th neuron; T_{ij} is the weight of the connection between neurons V_i and V_j ; and I_i represent the external input value. In general the weights can be positive or negative, for excitatory or inhibitory connections, respectively.

Once the net input u_i to neuron i is computed, the resulting *change* ΔA_i of the activation value A_i of neuron i is computed as follows:

$$\Delta A_i = \begin{cases} (max - A_i) u_i - decay (A_i - rest) & \text{if } u_i > 0; \\ (A_i - min) u_i - decay (A_i - rest) & \text{otherwise.} \end{cases} \quad (5)$$

Note that in this equation, *max*, *min*, *rest*, and *decay* are all parameters. Before the network starts to evolve, all the activation values of the neurons are set to the value of *rest*; and during the evolution process of the network, a force (i.e., the second term $-decay(A_i - rest)$ on the right-hand side of Eq. (5)), which is proportional to the value of *decay*, tends to bring the activation value close to the *rest* value which is usually set to be between *min* and *max*.

McClelland and Rumelhart used the IAC network for letter and word perception [4], content-addressable database [5], and speech perception [6].

2 Relaxation by IAC Network

When performing the relaxation process on an IAC network, the neurons are used to represent the various possible hypotheses in the relaxation process (e.g., the neuron with index (i, j) represents the hypothesis that object i belongs to class j), and the links between the neurons embody the relationships among the hypotheses.

To perform the relaxation process on an IAC network, the decay term in Eq. (5) is eliminated by setting the *decay* value to zero since there is no such mechanism in the relaxation process. Moreover, unlike McClelland and Rumelhart's approach, the output value of a neuron is defined in this study to be identical to the neuron's activation value that is, $V_i = A_i$. Theoretically, the range of the neuron activation value, $[min, max]$, should be set to $[0, 1]$, since in the probabilistic relaxation scheme, the probability value of each hypothesis should remain within such a range; but in practice the range is set to $[-0.01, 1]$ and the reason is that: during the relaxation process, if it happens that $p_{ij} = 0$ for a certain i and all j except $j = k$, then in the original relaxation process, the value of p_{ik} should be set to 1 to keep the equation $\sum_j p_{ij} = 1$, for any i and j , to be true. On the contrary, in the IAC network, when the outputs of all the neurons representing p_{ij} for a certain i and all $j \neq k$ reach zero, they cannot influence the neuron representing p_{ik} since their outputs are all zero.

According to Eq. (3), Eq. (5), and the above discussions, the output of a neuron in the IAC network for the relaxation process is proposed to change in accordance with the following rule:

$$\Delta V_i = \begin{cases} (1.0 - V_i) u_i & \text{if } u_i > 0; \\ (V_i - (-0.01)) u_i & \text{otherwise} \end{cases} \quad (6)$$

where u_i is as defined in Eq. (4).

During the process of the probabilistic relaxation, the total sum of the probability values of each class of any object should remain to be 1 (i.e., $\sum_j p_{ij} = 1$, for all i and j). To create this property in the IAC network, we construct a number of pools, and call them the S pools in the sequel. The neurons in each S pool represent the various possible hypotheses for an identical object in the original relaxation problem, i.e., the neurons representing $p_{i1}, p_{i2}, \dots, p_{im}$ are put in an identical S pool. The neurons in each pool will compete with each other, since they are mutually inhibited; thus, in each S pool at most one neuron will stay in the active state after the network converges into a stable state. On the other hand, to include other types of mutually exclusive hypotheses in the IAC, pools other than the S pools can also be created, so a neuron in the network may belong to multiple pools.

Moreover, the negative compatibility coefficients can be carried by inhibitory connections, and the neurons which are linked by inhibitory connections can be conceptually arranged into different pools according to the characteristics of the compatibility coefficients. Actually, the connection weight between the neuron with index (i, j) and the neuron with index (h, k) is established in this study by the following rule:

$$T(i, j; h, k) = \begin{cases} \alpha & \text{if } i = h \text{ and } j \neq k; \\ c(i, j; h, k) & \text{otherwise,} \end{cases} \quad (7)$$

for $j, k = 0, \dots, m - 1$ where m is the number of classes of the original relaxation process; and the value α is set to be negative and is given for the neurons in an identical S pool to provide a unique active neuron in the pool when the network converges.

Finally, the initial condition of the network is defined by allowing each neuron with index (i, j) to have its own *rest* value $rest_{ij}$ which is set to be the corresponding initial probability value of the original relaxation process.

3 Experimental Results

To demonstrate the correctness of the proposed mapping method between the relaxation process and the IAC network, a modified version of a typical NP-complete problem, namely, the n -queens problem, is solved in our experiment.

The definition of the n -queens problem is that given an $n \times n$ chessboard, how does one place n queens on the board so that none can capture any other? In this study, we modify the problem by assuming first that a queen is placed initially at a certain position on the chessboard and then trying to find out where the other queens should be placed.

In the experiment, we first use the relaxation process to solve the modified n -queens problem, and then use the proposed IAC network to perform the same task on the same data.

3.1 Solving Modified n -queens Problem by Probabilistic Relaxation

When applying the probabilistic relaxation method to the modified n -queens problem, there are $n \times n$ objects, corresponding to the $n \times n$ positions of the chessboard, and two classes, corresponding to the existence (class 1) and nonexistence (class 0) of a queen. For each position (x, y) there are two probability values, $p_{(x,y)0}$ and $p_{(x,y)1}$, where $p_{(x,y)1} = 1 - p_{(x,y)0}$. Moreover, for all pairs of position (x, y) and position (u, v) , there are four compatibility coefficients, $c((x, y), 0; (u, v), 0)$, $c((x, y), 0; (u, v), 1)$, $c((x, y), 1; (u, v), 0)$, and $c((x, y), 1; (u, v), 1)$.

The initial work is to assign probability values to each position according to a heuristic rule which estimates the likelihood for the position to be in the existence or nonexistence class. At each position (x, y) of the chessboard, the number of all the positions (including (x, y) itself) which will be attacked by a queen placed at (x, y) is defined as the *inhibitory power* at (x, y) . A larger inhibitory power value at a certain position means less favor of placing a queen at the position, since it will inhibit more positions to be allowed for placing a queen.

In the modified 5-queens problem, a queen is placed initially at a certain position (u, v) before the problem is solved. The initial probability values $p_{(u,v)1}^{(0)}$ and $p_{(u,v)0}^{(0)}$ for (u, v) are set to 1 and 0, respectively. For the other positions, the initial probability estimates of the 5-queens problem can be computed in accordance with the following rule:

$$p_{(x,y)1}^{(0)} = \frac{(n^2 - I(x, y)) \times n}{\sum_x \sum_y (n^2 - I(x, y))}; \quad (8)$$

$$p_{(x,y)0}^{(0)} = 1 - \frac{(n^2 - I(x, y)) \times n}{\sum_x \sum_y (n^2 - I(x, y))} \quad (9)$$

where $I(x, y)$ is the inhibitory power of position (x, y) and n is the total number of queens to be placed on the chessboard ($n = 5$ for the 5-queens problem).

By the definition of the n -queens problem, the existence of a queen at position (x, y) will inhibit the existence of a queen at a position on the horizontal, vertical, or the two diagonal directions of position (x, y) . This observation leads us to define the compatibility coefficients between the operation of placing a queen at position (x, y) and that of placing a queen at position (u, v) as follows:

$$c((x, y), j; (u, v), k) = \begin{cases} -1 & \text{if } j = k = 1 \text{ and a queen placed at position } (u, v) \\ & \text{will attack a queen placed at position } (x, y); \\ \beta & \text{otherwise,} \end{cases} \quad (10)$$

for $j, k = 0$ or 1 , where β is a small positive real number for encouraging a queen to be placed at position (x, y) . Too large β values will result in more than n queens to be placed on the chessboard, and vice versa. In this study, the β value is set to 0.22 after some try-and-error work.

After defining the compatibility coefficients and the initial probability values, probabilistic relaxation then is a deterministic iterative process. The net contributions from the other hypotheses to each probability value at the r th iteration is calculated by Eq. (1); and the new estimates of all the probability values at the $(r + 1)$ th iteration are then updated by Eq. (2). The above process is performed successively until all probability values are less than 0.01 or greater than 0.99.

When solving the modified 4-queens problem, since there are 16 positions on the chessboard, 16 experiments have been performed. Each of the 16 positions was selected once for placing the first queen. There exist 2 configurations which are the legal solutions of the modified 4-queens problem (see Fig. 1(a)). That is, placing a queen initially at one of the eight positions indicated by the Q 's in both diagrams of Fig. 1(a) produces a correct solution, and placing a queen initially at the other positions results in no correct solution. Fig. 1(b) shows an example of placing a queen initially at the left top corner of the chessboard, which results in no solution.

When solving the modified 5-queens problem, since there are 25 positions on the chessboard, 25 experiments have been performed. Each position was selected once for placing the first queen. Since at each position chosen to place the first queen, there exists at least one configuration which is a legal solution of the 5-queens problem, all of the 25 experiments produce correct solutions.

All the experiments produce obvious favors of some positions than the others within about 900 iterations and converge within about 1600 iterations.

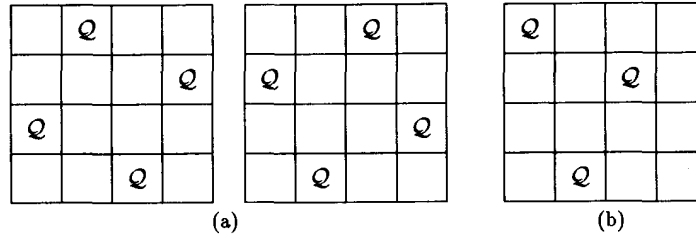


Figure 1: Solutions of 4-queens problem. (a) The two legal solutions of the modified 4-queens problem. (b) Placing a queen at a wrong position for the 4-queens problem results in no legal solution. The initially placed queen is located at the left top corner. Note that only three queens appear in the final result but four are required.

3.2 Solving Modified n -queens Problem by Relaxation on IAC Network

In the experiments, the IAC networks for solving the modified 4-queens and 5-queens problems are constructed next. A neuron with index $((x, y), 1)$ represents the hypothesis that a queen should be placed at position (x, y) ; and a neuron with index $((x, y), 0)$ represents the contrary hypothesis. There exists a connection between any two neurons in the network: a negative connection means that the two neurons compete with each other, and a positive one means the contrary case. The neurons with index $((x, y), 0)$ and $((x, y), 1)$ compose an S pool. Actually, there are n^2 S pools in the IAC network for the n -queens problem. Furthermore, the neurons representing the operations of placing a queen at different positions of a certain row, column, or diagonal can be considered as forming another pool since they compete with each other. From Eqs. (7) and (10) and by changing the indices, it is easy to derive the values of the connection matrix as follows:

$$T((x, y), j; (u, v), k) = \begin{cases} -1 & \text{if } j = k = 1 \text{ and a queen placed at position } (u, v) \\ & \text{will attack a queen placed at position } (x, y); \\ \alpha & \text{if } (x, y) = (u, v) \text{ and } j \neq k; \\ \beta & \text{otherwise,} \end{cases} \quad (11)$$

for all $(x, y), j, (u, v)$, and k .

Eq. (11) is the same as Eq. (10) except that a negative α value is created for the neurons in an identical S pool. In this study, the α value was set to -0.1 and the β value was set to 0.22 . During the operation of the IAC network, the graded response characteristics of the neurons in the IAC network represent partial knowledge or belief. The output value of neuron $((x, y), 1)$ represents the strength of the hypotheses that a queen should be placed at position (x, y) .

As mentioned in Section 2, the initial condition of the network can be defined by setting the rest values to be the corresponding initial probability values of the original relaxation process according to the following rule:

$$\begin{aligned} rest_{(x,y)1} &= P_{(x,y)1}^{(0)}; \\ rest_{(x,y)0} &= P_{(x,y)0}^{(0)}. \end{aligned}$$

Furthermore, the external input value, $I_{(u,v)1}$, in Eq. (4) should be set to a large positive number, say 10000, such that the activation value of the initially placed queen, $V_{(u,v)1}$, can remain to be 1 during the evolving process. The external input values of the other neurons are set to zero.

Like the experiments described in Section 3.1, 16 experiments have been performed when solving the modified 4-queens problem, and 25 experiments have been performed for the modified 5-queens problem. Each position was selected once for placing the first queen. The results of the 41 experiments for solving the modified 4-queens and 5-queens problems show that identical solutions are always produced for both the probabilistic relaxation approach and the IAC network approach.

In the IAC network, all the experiments also produce obvious favors of some positions than the others within about 60 time steps and converge within about 600 time steps.

As mentioned above, the relaxation process took about 1600 iterations before terminated. On the other hand, the simulated IAC network took about 600 time steps to converge. There is no direct correspondence between the iterations and the time steps, since the time steps are only used for simulating the continuous time flow. The proposed mapping method is quite effective in performing the relaxation process as shown by the experimental results.

4 Concluding Remarks and Future Research Directions

A mapping method that makes the IAC network perform the relaxation process has been proposed in this paper. To perform the relaxation process in such a network, the neurons are used to represent the various possible hypotheses in the original relaxation process, and the compatibility coefficients are carried by excitatory and inhibitory connections. To guarantee that when the network converges, only one neuron is active in a set of mutually inhibitory neurons which represent the various possible hypotheses for an object in the original relaxation problem, the type of S pool was introduced. Other types of pools was also created to reflect additional mutual exclusive hypotheses in the problem. By this method, the neural network technology can be easily adapted to solve the many problems which have already been solved by the relaxation process. The proposed mapping method is quite effective in making the IAC network perform the relaxation process as shown by the experimental results. The advantages of the neural network can thus be injected into the numerous relaxation applications.

The relaxation technique is a very important technique for many application domains. Hundreds of research papers has been published in this area. On the other hand, the neural network technology has also attracted very much attention recently. Therefore, another resulting advantage of establishing the relationship between the relaxation process and the neural network technique is that the research results of one domain can be very instructive to the other. For example, there exist some common properties in the convergence problem of the relaxation process and that of the neural network operations [7, 8]. Next, a number of works have been done for systematically generating compatibility coefficients of the relaxation process [9]. On the contrary, the generation of the connection weights of a neural network by teaching or learning is also an important issue in neural network research [10]. It seems possible to apply the results to each other.

References

- [1] A. Rosenfeld and A. C. Kak. *Digital Picture Processing*, volume 2. Academic Press, New York, 1982.
- [2] J. L. McClelland and D. E. Rumelhart. *Explorations in Parallel Distributed Processing*. The MIT Press, Cambridge, MA, 1988.
- [3] D. E. Rumelhart, J. L. McClelland, and the PDP research group. *Parallel Distributed Processing (2 volumes)*. The MIT Press, Cambridge, MA, 1986.
- [4] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88:375–407, 1981.
- [5] J. L. McClelland. Retrieving general and specific information from stored knowledge of specifics. In *Proceedings of the Third Annual Meeting of the Cognitive Science Society*, pages 170–172, 1981.
- [6] J. L. Elman and J. L. McClelland. Speech perception as a cognitive process: The interactive activation model. Technical Report ICS-8302, University of California, Institute of Cognitive Science, San Diego, CA, 1983.
- [7] J. Bruck and J. W. Goodman. A generalized convergence theory for neural networks. *IEEE Transactions on Information Theory*, IT-34, 1988.
- [8] S. W. Zucker, E. V. Krishnamurthy, and R. L. Haar. Relaxation processes for scene labeling: Convergence, speed, and stability. *IEEE Transactions on System, Man, and Cybernetics*, SMC-8:41–48, January 1978.
- [9] S. Peleg and A. Rosenfeld. Determining compatibility coefficients for curve enhancement relaxation processes. *IEEE Transactions on System, Man, and Cybernetics*, SMC-8:548–555, July 1978.
- [10] C. T. Genis. Relaxation and neural learning: Points of convergence and divergence. *Journal of Parallel and Distributed Computing*, 6:217–244, 1989.