сообщения
объединенного
института
ядерных
исследований
дубна

E10-93-415

S.A.Baginyan, G.A.Ososkov

CONTROLLED NEURAL NETWORK APPLICATION IN TRACK-MATCH PROBLEM

1993

1 Introduction

A wide class of problems exists, where it is necessary to assign certain values to some group of variables that mutually satisfy to given constraints. Such a class of the problems is known as Constrain Satisfaction Problems (**CSP**) [1]. For example, among **CSP** there is a problem of schedule generating. The next **CSP** is known in combinatorics: there are given both the set of elements $S = a_1, a_2, \ldots a_N$ and the set of its subsets $S_\Omega = S_1, S_2, \ldots S_M$. It is necessary to choose from each subset $S_i \quad i = \overline{1, M}$ only one element different from others.

The most famous in the latter class is so-called $N-$queens problem: to allocate N queens on an $N * N$ chessboard, one on each row, so that no queen threatens another.

As it'll be shown below in the next section, the *track-match* problem of high energy physics (HEP) can be also reduced to a **CSP**.

We consider **CSP**'s in the following general formulation [1]:
The set of variables $X_1, \ldots X_N$ is given. Each variablle X_i can take only certain values d_{ij} from the finite domain

$$D_i = d_{i1}, \ldots d_{iM}$$

A number of constraints is given explicitly as the prohibition of certain X_i and X_k pairwise combinations $C(X_i, X_k)$. If, for instance $(d_{13}, d_{24}) \in C(X_1, X_2)$ then it can mean

$$X_1 \neq d_{13}, \text{ when } X_2 = d_{24}$$

A **CSP** purpose is to assign values from D_i to all variables $X_i \quad i = \overline{1, N}$ without any transgression of $C(X_i, X_k)$.

The main problem of **CSP** solving by a conventional search algorithm is the enormous computer time consumption. In particular, $N-$queens problem can't be solved in reasonable time on serial computers already, when $N \geq 97$. A more complicate example relates to the generating of the half-year activity schedule for the satellite called ROSAT (Roentgeen Satellite). The solving of this problem by the method based on mathematical optimisations required about 30 hours of a serial computer. However, a new approach using a special type of an Artificial Neural Networks (ANN) was succesfully applied to solve this problem on the same computer in 1 minute [1]. This approach allows to solve the $N-$queens problem for $N = 1024$ in about 10 minutes.

Such encouraging results stimulated us to develope a new approach of an external influence into a Hopfield ANN [2] dinamics in order to control its evolution in a desirable way, mainly to avoid sticking into local minima of this ANN energy function. As the first relatively simple application of our approach we chose the solution of the famous HEP *track-match* problem. This ANN application is considered in this paper.

2 Binary adjusency matrix construction

The *track-match* problem arised originally as a part of the general problem of the reconstruction of space trajectories of charged particles on the basis of their projections measured in several views [3]. At least two views are necessary to the autentive space reconstruction, which is carried out by a set of physical and geometrical criteria testing for each pair of views their compatibility in space. However, if the event multiplicity (the number of secondary tracks) are greater than 15, the combinatorics of all pairs to be tested are so numerous that any conventional algorithm based on the sequentional search would be too time consuming.

The analogous *track-match* problem arised in many HEP experiments, when it's necessary to identify the group of tracks with their continuations after intersecting some "black box" (for example, magnet, see fig. 1).

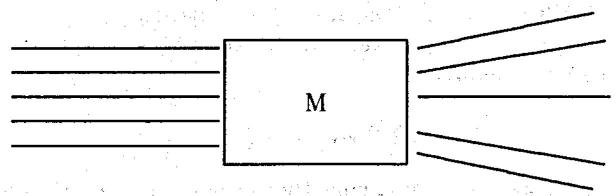

fig. 1

In both examples *track-matching* procedure consists of two steps:

1. Each input track is tested to be appropriate to each output track by some set of physical and geometrical tests Ω. As the result one obtaines the list of pairwise track combinations to be tested (for example, some track before magnet with some one after).

   Let us consider an example of two possible lists of track pairing:

   list *a*) $(1,1)(1,2)(2,1)(2,2)(3,3)(4,4)(4,5)(5,4)(5,5)$

   for another event

   list *b*) $(1,1)(1,2)(1,3)(1,4)(2,1)(2,3)(3,1)(4,1)$

   where the first element in each pair is the track number before magnet and the second element is the track number after magnet.
   For the event with a large number of tracks the number of pairs in such lists can be much greater than the event multiplicity because of the changes of track ordering before- and after magnet and of errors of the track recognition.

2. The subset of one-to-one-correspondending pairs is extracted from the obtained list in such a way that the number of pairs in this subset must be equal to the number of the tracks of the event.

One can see that for the **list** *a*) it is possible, for example $(1,2)(2,1)(3,3)(4,5)(5,4)$. However, for the **list** *b*) it is impossible, as maximum number of the tracks with one-to-one-correspondence is equal to 3, but the number of the tracks in the event equals 4: $(1,2)(2,3)(3,1)$ or $(1,2)(2,3)(4,1)$.

Let us formulate this matter in terms of binary matrices called in graph theory as adjucency or incidency matrices [4].

Consider the track numbers before magnet as the number of a matrix columns and the track numbers after magnet as the matrix row numbers. If in the list of pairs constructed on the first step of the *track-match* procedure there is a pair (i,j), then we have to place 1 on the intersection of $i-th$ column and $j-th$ row of matrix. The other elements of the matrix are equal to 0. This binary matrix is in one-to-one-correspondence to the list of pairs.

Therefore, the second step of the *track-match* procedure can be formulated as follows: for each row of the constructed adjucency matrix only one non-zero element must be extracted so that all these elements are lyeng on different columns of tne matrix.
On the chess language this problem can be formulated as follows: some number of rooks are placed on a chessboard without empty rows. One should remove some of rooks leaving one on each row, so that no rook threatens another.

Unlike to $N-$queens allocation problem, this problem is not always soluable. Besides, since rooks don't threaten by diagonals like queens, the number of variants of their allocations on a chessboard is much greater that leads to increasing search attempts.

3 Controlled neural networks

It is clear that the problem formulated above belongs to the class of **CSP**. For solving this problem let us construct a Hopfield's neural network (HNN) [2], i. e. a system of mutually connected binary elements (neurons), which connections are characterized by synaptic efficacies (weights).

As Hopfield proved [2], if an HNN-weight matrix is symmetric with zero diagonal, then the energy function of this neural net is decreasing for arbitrary HNN-dynamics being attracted to one of local minima.

Therefore, we have to define

- neurons
- topology of their connections
- weight function
- energy function
- HNN dynamics rules

To deal with terms of incidency matrices invented in the provides section, let us consider only non-zero elements of such a matrix. These elements U_{ij} can be chosen as binary neurons with two possible states:
- $U_{ij} = 1$ (neuron is firing or active), if the rook is placed on the cross of i-th row and j-th column;
- $U_{ij} = 0$ (neuron is non-active), if one removes the rook from its place.

The HNN topology is simple: each neuron is connected with each other.

The weight function selection should guarantee that no rook threatens another for their given dislocation, i. e. weights must support (=1) permitted connections and punish (be strongly negative) connection indicating the threat between two rooks. According to the general HNN requirements the diagonal elements of the weight matrix must be equal to zero. Thus for an $N * N$ matrix one has

$$W_{ij,mn} = \begin{cases} 0 & if \quad i=m \ \& \ j=n \\ 1 & if \quad i\neq m \ \& \ j\neq n \\ -4*N & if \quad i=m \ \& \ j\neq n \\ -4*N & if \quad i\neq m \ \& \ j=n \end{cases} \tag{1}$$

The value of the local field generated by HNN in the neuron U_{ij} is determined by the standard formula [2]

$$I_{ij} = \sum_{m,n}^{N} W_{ij,mn} U_{mn}$$

This defines the easiest stepwise function of HNN dinamics. The state of each neuron changes asynchronously

$$U_{ij} = \begin{cases} 1 & if \quad I_{ij} > \theta \\ 0 & if \quad I_{ij} \leq \theta, \end{cases} \tag{2}$$

where θ is the chosen threshold constant.

The HNN energy function is defined also in the standard way [2]:

$$E = -\frac{1}{2} \sum_{i,j}^{N} \sum_{m,n}^{N} W_{ij,mn} U_{ij} U_{mn} \tag{3}$$

The solution of our problem is achieved if and only if the only one neuron is active on each row of our matrix, while all these neurons are placed in its different columns. Since the definitions (1)–(3) satisfy the conditions of Hopfield's theorem [2], this final HNN configuration giving the solution corresponds to the global minimum of (1). In our particular case due to (1)–(3) the exact value of this global minimum can be calculated explicitly as

$$-\frac{1}{2} N * (N-1) \tag{4}$$

This remarkable fact should simplify very much the criterium of the exit from algorithm of the global minimum search.

Our approach to design such an algorithm is based on the following concept stimulated by [1], our previous works [5], [6] and ideas of the stochastic search from [7], [8].

1. Decrease of the number of HNN degree of freedom by constraint applying neurons with the maximum value of the local field.
2. Forced escape from the local minima of the energy function.
3. Stochastic steps of HNN evolution.

After many various attemts to develop an algorithm carried out these principles we elaborated the following effective

procedure for CSP solving:

1. Set up to zero all neurons and threshould θ.
2. Select randomly a row of the matrix.
3. At this row look for neurons with the local field satisfied the conditions

$$\begin{cases} I_{ij} & if \quad U_{ij} = 0 \ \& \ I_{ij} > 0 \\ |I_{ij}| & if \quad U_{ij} = 1 \ \& \ I_{ij} \leq 0 \end{cases} \tag{5}$$

4. Between these neurons choose one with the maximum of the local field and invert its value (set up to 0 if it was one and *vice versa*).
5. Check whether all neurons on the current row are non-active (that correspond to one of local minima). If it's so, invert forcerly the first neuron from the right or left of one chosen on the previous step (in a case, if only one neuron in this row, invert it).
6. Repeat steps 2–5 until all rows of the matrix are checked.
7. Calculate the value of the energy function and if it isn't equal to (4) (that means we found a solution and can stop), repeat steps 3–7.

Remark. Since, in principle, our problem can have no solution, the total number of steps in n. 7 must be restricted by an reasonable value ($N * 200$).

We named our new neural network the **Controlled Neural Netwrk (CNN)**, since its evolution is forcerly changed on steps 4, 5, 7 in order to escape from a local minimum.

4 Results and Conclusion

The comparable study was accomplished to test the applicability of the CNN algorithm for such a **CSP** as the *track-match* problem with a variety of incidency matrices.

The CNN algorithm was compared with the conventional mathematical method (CMM) based on the standard sequential search algorithm. The problem solved by both methods for $N * N$ incidency matrices with different complicated structures was to leave on each row a single element keeping all of them in different columns.

$VAX - 8350$ CPU times of this problem solution for different N are presented in the table

N	30	15	13
CNN	105.32 s	0.26 s	0.13 s
CMM	7.5 hours	23.78 s	0.135 s

Table. Time of **CSP** solving by CNN and CMM.

As one can see, for $N \geq 30$ our CNN algorithm shows the very high prefomance, while CMM application is not reasonable that confirms the fruitfulness of the CNN concept formulated above.

The specific feature of the solved **CSP** is the possibility to calculate in advance the exact value of the HNN energy function, which simplify considerably the proposed CNN algorithm.
However, the generality of the CNN concept allows to predict this algorithm can be developed for more general applications, in particular, for the such an important HEP problem as track finding [5], [6].

References

[1] Oku, Y. Introduction to the Artificial Intelligence Methods, Part 3: Artificial Neural Networks, 1993.

[2] J. J. Hopfield and D. W. Tank, Biological Cybernetics, 52 (1985) 141.

[3] S. A. Baginyan and V. I. Moroz. Commun. JINR, 10-85-289, Dubna, 1985 (in Russian).

[4] O. Ore, Graf theory."Nauka", M. ,1977 (in Russian).

[5] S. Baginyan et al, Commun. JINR, E10-93-86, Dubna, 1993.

[6] S. Baginyan et al, *Robust neural network for track information extraction*, Int. Con. on Prog. and Mathem. Met. , Dubna, 14-19 June, 1993 (to be published).

[7] N. I. Chernov, G. A. Ososkov and L. Pronzato, Appl. of Math. 37(1992) No 6, 437-452.

[8] Y. A. Budagov et al, Commun. JINR, P10-91-311, Dubna, 1991 (in Russion).

Received by Publishing Department
on November 18, 1993.

Багинян С.А., Ососков Г.А. E10-93-415

Применение управляемой нейронной сети
в проблеме track-match

В терминах так называемой матрицы инцидентности сформулирована задача соответствия треков, относящаяся к проблемам обработки данных в физике высоких энергий. Для решения подобных задач с ограничениями разработана искусственная нейронная сеть хопфилдова типа. На основе предложенной концепции управляемых нейронных сетей создан алгоритм, реализующий эффективный поиск решения. Приведены результаты вычислений, показывающие значительное превышение по скорости предложенного алгоритма по сравнению с обычными методами, основанными на последовательном переборе.

Работа выполнена в Лаборатории вычислительной техники и автоматизации ОИЯИ.

Сообщение Объединенного института ядерных исследований. Дубна, 1993

Baginyan S.A., Ososkov G.A. E10-93-415

Controlled Neural Network Application
in Track-Match Problem

Track-match problem of HEP data handling is formulated in terms of incidency matrices. The corresponding Hopfield neural network is developed to solve this type of constraint satisfaction problems (CSP). A special concept of the controlled neural network is proposed as a basis of an algorithm for the effective CSP solution. Results of comparable calculations show the very high performance of this algorithm against conventional search procedures.

The investigation has been performed at the Laboratory of Computing Techniques and Automation, JINR.

Communication of the Joint Institute for Nuclear Research. Dubna, 1993