The New Self-Organizing Mapping Algorithm for TSP

Zhu Shuanghe, Ma Ling and LuHu

Telecommunication Engineering Institute, Air Force Engineering University, Xi'an, 710077,
P. R. China,

Abstract: A modified self-organizing mapping artificial neural network algorithm is proposed for TSP. It not only is more efficacious, accuracious, and higher speed than HNN, but also avoids the local minima phenomena and excels in the speed and accuracy.

Key words: Self-organizing mapping, TSP, Hopfield artificial neural network (Hopfield ANN)

1. Introduction

The travelling salesman problem, namely TSP, can be interpreted as the following: A salesman wants to travel around n cities, and the every city is visited only one time. In the end, he returns to the starting city, however, which is the shortest travelling distance in all distances? This problem is a classical artificial intelligence hard question that is particularly significant in the theory study and practical application. The problem is not simple, as it might seem. In fact, the solution is rather complex by computer. There are \( n!/2n \) various circle paths (Satisfied, setting out from one city, passing through every city only one lime, and returning to the starting city at last). The number, for instance, is \( n=20 \). The result, \( n!/2n=6 \times 10^{16} \) is to be large. It will take a computer at \( 10^{10} \) per second twenty years to find the shortest travelling distance. Obviously, it's impossible to get the answer by the exhaustive searching method. The problem is named NP hard or complete problem (namely, it is not able to obtain the accuracy solution in the Polynomial boimed). On using the normal optimal combination method, there will be so-called "combination explosion" problem.

In long terms, men liave consistently searched die methods to solve the problem, but the results are all disappoint. With the climax of the study in the ANN, several scholars come up with the new algorithm based on neural networks. Among these algorithms, Hopfield neural network (HNN) is the most famous one, which constructs the energy function. In this paper, we propose a new algorithm based on self-organizing character; mapping neural network. The simulating results show it not only is more efficacious, accuracious, and higher speed than HNN, but also avoids the local minima phenomena.

2. The Fundamental Principle

The fundamental principle for TSP with the self-organizing mapping algorithm is the following: First the city's rectangle coordinates composed of the two-dimensional vector \( V_i \), indicates the city. Supposed, \( n \) cities are to be traveled, menHie coordinates of the cities form the \( n \) two-dimensional vectors. These vectors can be organized as a training vector set \( D_v \). The each
input vector of the self-organizing mapping neural network composed of one-dimensional neuron array is two-dimensional. Therefore, every neuron has a connection with two-dimensional weight vector. The each vector of the training set is sent into the network to be trained in turn or randomly according to the self-adapting weight adjustment rules of the self-organizing mapping algorithm. Because of the vector quantification characteristic of the self-organizing mapping algorithm, after the training the weight vectors of the network will be adjusted to be likeness (or sameness) as the input training vector. Meanwhile, the weight vectors of the network will be arranged, namely the every weight vector of the neural network, abiding by certain sequence, will been rearranged. So the distance between the neighbor neurons is going to minim. As the mapping and cluster characteristic of the Kohonen network, the near cities in the map are the near points in the output, too. If connecting these points with a chain, which, no doubt, is arranged according to the various distances between the cities. Therefore, the chain is the solution of TSP. So the chain of self-organizing network can be applied to solve the TSP, conveniently.

3. The Modified self-Organizing Mapping Algorithm

At first, we need to change the structure of the network as the following: Every neuron of the network composed of one-dimensional neuron array is given a one-dimensional position label. In order to satisfy the constrained conditions of the TSP, namely, the salesman returns to the starting city finally. The above one-dimensional array will be arranged to a ring, shows in the Figure 1. In other words, the label 1 neuron is next to label n neuron, and the distances between label 1 neuron and label 3 or label n-1 are equal. This can be realized by defining a proper neighborhood function in the algorithm. Second, as large the training times are in the algorithm; the vectors of the training vector set (namely, the number of the cities in the TSP) are fewer. So the training vector of the set need to be circularly used. In addition, though we may let the number of the neurons is equal to the number of the cities (namely, the number of training vectors), to avoid the middle value effect and to acquire smoother path. The number of neurons in the actual network will be more than mat of the cities. We use the following example to illustrate the above algorithm. The distribution of 10 cities in the plane is showed in the Figure 2. The each city's two-dimensional coordinate vector forms the training vector set including 10 vectors. The neuron array is a one-dimensional circular array composed of 25 neurons. The inputs are two-dimensional vectors, \( V_i = (V_{i1}, V_{i2}) \) whose component \( V_i \) connects each neuron with a variable weight coefficient. Therefore the each neuron \( j \), is connected a two-dimensional weight vector, \( W_j = (W_{j1}, W_{j2}) \). As the following steps, the network is trained:

1. To set the initial value of the weight vector of the every neuron of the network randomly.

2. To send the each training vector of the training vector set (namely, the city's coordinates) randomly into the network. After sending the each training vector \( V_i \) into the network, we calculate the output for the every
neuron of the network value as the below formula:

\[ Y_j = \| W_j - V_j \| = \sqrt{\sum_{i}(w_{ij} - v_i)^2}, i = 1, 2, j = 1 \sim 25 \]

In practice, the above formula calculates the Euclidean distance between each neuron weight vector and the input training vector.

3. The network does competitive selection, namely, as the criterion of the of the shortest distance to select a victory neuron.

\[ C = \arg \left( \| V - W_i \| \right) \]

That is:

\[ \| V - W_i \| = \min_j \| V - W_j \| \]

In other words, the distance between the triumph neuron's weight vector and the input training vector is to be shortest.

4. All the neuron joint weights according to their positions in the network and distances with the triumph neuron, abiding by the strategy of self-organizing weight adjustment, modify themselves, and can be formulated:

\[ W_i(t + 1) = W_i(t) + \varepsilon(t) h_i(t) [V(t) - W_i(t)] \]

where

\[ e(t) = e_0 \exp(-\varepsilon, t) \]

\[ h_i(t) = \exp \left( -\frac{d^2}{2\sigma^2(t)} \right) \sigma(t) = \sigma_0 \exp(-\sigma(t)) \]

are the learning rate factor and neighborhood functions, respectively.

On selecting the parameters, that the problem ought to be considered is the circle structure on calculating the distances between the neurons to identify the neighborhood function. Namely, the distances between the label 1 and label 2 or label n neuron are equal, etc. In addition, the sequence of samples from the training vector set is arbitrary. So the training vectors can be selected randomly or circuly according to some sequence; while the algorithm is realized actually. But to get the better results, after the learning, the times, which the each neuron of the training set are trained, should be approximate sameness.

4. Simulation Results

We simulate the TSP of ten cities by computer. Figure 3 gives the experimental result. From the figure, though the initial weight values are set randomly, we can conclude, after several iterated trains, the every vector will converge to the position which is adjacent to or as same as the training vectors (namely the coordinate of the given cities.). We link these neighbor neurons in sequence; the folding line will indicate an optimal visiting path.

5. Conclusion

From the above analysis and experimental result, it is very convenient to solve the TSP with the modified self-organizing mapping algorithm, and the ultimate convergence is in good performance. On satisfying the convergence condition of the self-organizing mapping algorithm, changing several learning parameters or using various input sequences of the training vectors, the network always converges to the optimal visiting path. It indicates that the algorithm isn’t sensitive to the selection of parameters.

The data in this paper are the same as
those of reference 21[31. In the reference \textbackslash
Hopfield applied the energy function to con-
struct Hopfield neural network the for TSP. But
the algorithm has the local minimal pheno-
menon, if applying the self-organizing algo-
rithm proposed in this paper, it
can be avoided and the results are more
accuracy.

References

Springer 1995
2. Hopfield.JJ and Tank D.W. Neural Com-
putation of Decisions in Optimal Problems.
Biological Cybernetics, 1985. Vol. 52. 141-152
3. Ulson.G.V and Panley .G.S. On Stability of
the Travelling Salesman Problem Algorithm of
Hopfield and Tank. Biological Cybernetics,
1988. Vol 58. 63-70
of Neural Networks. Xi’an University of Elec-
tronic and Technology Press, (Chinese). 1990
5. Yang.X.J. and Lieng.J.L., Artificial Neural
Networks. Higher Education Press, Peijing,
(Chinese). 1992