ISSN: 2664-9640

Study of TSP problem solving using SOM

Wenkai Sun

University of Science and Technology Liaoning School of Computer and Software Engineering
Liaoning Anshan 114000, China

Abstract

SOM is an unsupervised learning neural network used to solve the TSP problem when two-dimensional city coordinates are the input to the neural network and city spatial location relationships are the patterns to be learned by the neural network. A circular neuron structure is finally output for solving the urban road planning problem. Eventually, after several iterations, a smooth loop is formed. Compared with other methods when using SOM to solve the TSP problem, SOM possesses the characteristics of fast computing speed, while the accuracy is partially reduced compared to other algorithms.

Keywords

TSP; SOM; Urban Road Planning.

1. Introduction

The TSP problem is a typical optimization problem. It is basically described as a merchant who travels from a city to each city in an intermediate route so that he travels the shortest distance. From the graph theory point of view, the problem is essentially to find a Hamiltonian loop with the smallest weight in a fully undirected graph with weights. When the number of cities is relatively small, we can find the shortest path by arranging all the city loops. When the number of cities is large and complex, city-by-city traversal is not efficient.

To solve this problem, we can try to improve Self-Organizing Maps, or Self-Organizing Feature Mapping, which is a kind of unguided training neural network. The process of self-organization is actually a kind of unguided learning. It automatically clusters the input patterns through its own training. When SOM receives external input patterns, it is divided into different response regions, and each region has different effects on the input patterns.[1]

1.1. Basic Principle

SOM has two layers of neural networks: input and output. The input layer input data is the real city point coordinates, and the output layer is the final formed loop path. The goal is to map the data into the output layer, and the individual point parameter weights are continuously updated during the training process, whereby the output layer neurons gradually converge to the input layer. There is a lateral inhibition phenomenon between biological neurons, where neurons in close proximity excite each other and neurons further away inhibit each other. The neuron with the strongest response, i.e., the winning neuron, forms a region with its center, and the further away from the center the weaker the excitation.

In the SOM model, the peripheral neurons are motivated to generate a winning region based on the exemplar correlation. Within the region, the strength of action is correlated with the exemplars. The correlation between strength of action and distance is reduced to a Gaussian function. Neurons in close proximity will have large parameter weights, and the peripheral region neurons will keep decreasing.[2,3]

ISSN: 2664-9640

1.2. Neuron vector update

In each training, the winning neuron is first selected based on the similarity of the weight vector of each neuron to the input vector. Then the weight vectors of the neurons around the winning neuron are updated to approximate the weight vector of the winning neuron and the input vector of the input layer. The output layer neurons are updated in such a way that the next neuron weight is the distance multiplied by the weight of the distribution of strong and weak relations in the region of the winning neuron plus the secondary neuron weight.

Neurons around the winning neuron's winning region are called nearest-neighbor neurons. The effect of the winning neuron on its near-neighbor neurons has a large weight in the region of proximity and a small weight in the region of distance. It conforms to a normal distribution, which is consistent with reality. The winning neighborhood is like a hotspot center, and neurons within its radius get renewed

and the neurons outside the superior neighborhood will not be updated.[4]

1.3. Solving the TSP problem using SOM

The neuron vector update process is described in Neuron Vector Update, but the key point in solving the TSP problem is to identify who is the winning neuron. We use Euclidean distance to measure the similarity, which is the Euclidean distance between the neuron weight vector and the input vector. The neuron with the smallest Euclidean distance is the neuron whose weight vector is closest to the input vector, and we set it as the winning neuron. When our input data is the location of the city coordinates, the location represented by the winning neuron is the closest to the location of the city in the map.

After continuous training, the weight vectors of both the winning neuron and its near-neighbor neurons are updated. After repeated iterations, the final neuron outputs the city relationship and is finally saved as a neuron vector. When the input and output layers are displayed in one view, it is found that the neuron positions continuously overlap with the city, forming a smooth closed loop. The final output is a path relationship that meets the conditions.

1.4. Algorithm convergence and parameter decay

The most important thing when using SOM to solve the TSP problem is to adjust the superior region. Each just competes with a few surrounding city nodes. With a slight modification, a ring is formed around it to keep approaching the cities and finally a solution is obtained.[5,6]

The core idea in the SOM is to adjust the winning region, but the algorithm does not converge automatically in time. In order to guarantee the convergence of the algorithm, a learning rate factor needs to be added to the winning region parameter and the example product parameter. Let the algorithm decay in time for both the neighborhood value and the learning rate. The decay rate of the learning rate factor over time is the product of the decay rate of the learning rate and the decay rate of the domain value.[7,8]

2. Conclusion

In this paper, a solution system for solving the TSP problem is constructed by referring to the related SOM neural network research and combining the characteristics of SOM. However, compared with other solutions like genetic algorithm, simulated annealing method, ant colony algorithm and other methods have the problem of accuracy rate to be improved. But compared with other methods, the SOM method has a significant speed improvement in the solution speed.

References

[1] Maenou T , Tanaka S , Fujimura K , et al. The application of SOM repeating method to TSP[J]. Ieice Technical Report Neurocomputing, 1996, 96:41-48.

[2] Tokutaka H , Fujimura K . . SOM-TSP[M]. 1999.
[3]Kikuo, Fujimura, Shin-ichi, et al. Optimization of Electronic Chip-Mounting Machine using SOM-TSP Method with 5 Dimensional Data. Springer London.
[4] Fujimura K , Obu-Cann K , Tokutaka H . Optimization of surface component mounting on the printed circuit board using SOM-TSP method[C]// International Conference on Neural Information Processing. IEEE, 2002.
[5] Ohno Y , Fujimura K , Tokutaka H , et al. The trial of improvement of the SOM-TSP method using the Gauss type kernel[J]. Ieice Technical Report Neurocomputing, 2003, 102:43-46.
[6] Kikuo F , Heizo T , Masumi I . Performance of Improved SOM-TSP Algorithm for Traveling Salesman Problem of Many Cities[J]. Ieice Technical Report Neurocomputing, 2008, 97(7):95-102.
[7] Tokutaka H , Fujimura K . SOM-TSP : An approach to optimize surface component mounting on a printed circuit board[J]. Kohonen Maps, 1999:219-230.
[8] Takahashi S , Fujimura K , Tokutaka H . The SOM-TSP method for the three-dimension city location problem[C]// International Conference on Neural Information Processing. IEEE, 2002.