Application Of Improved Hopfield Neural Network In Path Planning

Yunxiang Liu*, Wei Xu*
School of Computer Science and Information Engineering, Shanghai Institute of Technology, Shanghai, China
*yxliu@sit.edu.cn
*929854970@qq.com

Abstract A modified Hopfield neural network algorithm is proposed and applied to the path planning to solve some problems in the traditional Hopfield neural network. Firstly, the traditional A* algorithm is used to select the nodes in the search area that meet the criteria. Then, the nodes conforming to the standard are converted into neurons in the Hopfield neural network, and the stability of the network is used to iteratively select an optimal path. Experiments show that the improved Hopfield neural network algorithm can reduce the search time of path planning and improve efficiency.

1.INTRODUCTION
With the rapid development of industries such as drones and autonomous driving, the requirements for path planning in its navigation system are getting higher and higher. While ensuring safety, it is also necessary to balance the efficiency of operation[1][2]. In the traditional path planning research, the A* algorithm based on graph search is widely used. This algorithm uses the path planning algorithm for the first time[3], but there is no way to apply it in a complex environment; sampling planning algorithms, mainly artificial potential field method, PRM algorithm, etc.[4][5], the latter solves the local minimum phenomenon, but the efficiency and accuracy can not be guaranteed; the rise of neural network drives the development of industry path planning is also in it. Zhang Ying et al. used Hopfield neural network to plan the path of the transformer station[6], but it did not consider the efficiency problem; ouyang Ling solved the classic TSP problem by using Hopfield neural network[7], but the solution criteria are single and do not consider efficiency. In summary, the research on path planning is still going on. So how to apply the existing neural network algorithm to the path planning problem effectively is an urgent problem to be solved. An improved Hopfield neural network algorithm is proposed. The A* algorithm is effectively combined with the Hopfield neural network to reduce the search time of path planning and improve efficiency.

2.APPLICATION OF A-STAR ALGORITHM IN PATH PLANNING
The A* algorithm, also known as the A-Star algorithm, is the most efficient direct search method for solving the shortest path in a static road network. It is also an effective algorithm for solving many search problems. The closer distance estimate is to the actual value[8][9].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd
Figure 1. A-star algorithm example diagram

In Figure 1, A represents the starting point and B represents the ending point. Suppose someone wants to move from point A to point B. There is an obstacle between the two points. The steps to address the path using the A* algorithm are as follows:

Step 1: Dividing the path finding area into small grids;
Step 2: Start from the starting point A and add it to an OpenList consisting of squares. This OpenList has only one data at the beginning, which is the starting point A. The path in the OpenList may be along the way, or it may not. After that, OpenList is a list of squares to be checked;
Step 3: Look at the square adjacent to the starting point A (ignoring the squares in which it is not walkable), add the squares that can be walked or reachable to the OpenList, and set the starting point A as the father of these adjacent squares;
Step 4: Remove the A node from the OpenList and add it to the CloseList (closed list). Each square of the CloseList does not need to be concerned now;
Step 5: Select a square adjacent to the A node from the OpenList and repeat the step: \( F = G + H \). G represents the moving cost of moving from the starting point A to the current node; h represents the estimated cost of moving from the specified node to the ending point B. Find the node with the smallest F value;
Step 6: Select the node with the smallest F value from OpenList and do it: First, take the node out of the OpenList and put it in the CloseList; then check all the squares adjacent to it, ignoring it in the CloseList or Non-walkable squares; if the walkable squares are not in the OpenList, add them to the OpenList and set the currently selected node to the father of these newly joined squares;
Step 7: If an adjacent square is already in the OpenList, check if the road is better;
Step 8: Select the node with the smallest F value. When there is the same F value in the square, consider the speed to consider the square that is finally added to the OpenList;
Step 9: Repeat the previous process until the end point is also added to the OpenList.

It can be seen from the above path-finding steps that the traditional A* algorithm only considers the path between the starting point and the ending point, and does not comprehensively consider various factors. Therefore, the search efficiency of the algorithm needs to be improved.

3. HOPFIELD NEURAL NETWORK

3.1 Network Structure
Hopfield network is a fully connected neural network, which opens up a new research approach for the development of artificial neural networks. It simulates the memory mechanism of biological neural networks and has obtained satisfactory results. This network and learning algorithm was first proposed by American physicist J.J Hopfield in 1982, so it is called Hopfield neural network[10][11]. Continuous Hopfield Neural Network (CHNN) is a kind of Hopfield network, which is widely used in combinatorial optimization problems. Its network structure is shown in the figure 2 below.
From Kirchhoff’s current law (KCL) it can be concluded that:

\[
C_i \frac{du_i}{dt} = \sum_{j=1}^{n} w_{ij} v_j - \frac{u_i}{R_i} + I_i \quad (1),
\]

among them, \(i=1,2,...,n\); \(n\) is the number of neural network neurons; \(v_i\) is the output potential; \(u_i\) is the input potential; \(f(u_i)\) is the transfer function of network neurons; \(w_{ij}(i,j=1,2,...,n)\) is the network weight coefficient matrix.

### 3.2 Network Stability

The stability of the continuous Hopfield neural network, J. J. Hopfield defines the calculated energy function as follows:

\[
E = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} v_i v_j - \sum_{i=1}^{n} v_i I_i + \sum_{i=1}^{n} \frac{1}{R_i} f^{-1}(v_i)dv_i \quad (1),
\]

For continuous Hopfield neural networks, when the transfer function of network neurons monotonically increases and the network weight coefficient matrix is symmetric, the energy of the network will decrease or remain unchanged with time. That is when \(f^{-1}(.)\) is a monotonically increasing continuous function and \(w_{ij}=w_{ji}\),

\[
\frac{dE}{dt} = 0; \quad \text{the energy of the network will not change if and only if the output of the neuron does not change over time.}
\]

That is, when \(\frac{dE}{dt} = 0 (1 \leq i \leq n)\), \(\frac{dE}{dt} = 0\).

### 4. APPLICATION OF IMPROVED HOPFIELD NETWORK IN PATH PLANNING

In the traditional Hopfield network, when the path planning is carried out, the objective function, that is, the shortest path is transformed into the energy function of the network, and the variable of the problem is corresponding to the neuron state of the network. In the improved Hopfield neural network algorithm, A* algorithm is transformed into transposition matrices, and the practical meaning of the feasible path is considered, that is, each row and column of the transposition matrix has one and only one, so that when the energy function of the network converges to a minimum value At the same time, the optimal solution of the path planning problem is also obtained. Here, the network energy function of the path is defined as

\[
E = \frac{A}{2} \sum_{i=1}^{n} (\sum_{j=1}^{n} v_i - 1)^2 + \frac{D}{2} \sum_{i=1}^{n} (\sum_{j=1}^{n} v_i - 1)^2 + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{n} d_{ij} v_i v_j v_k \quad (2),
\]

among them, the first two are the constraints of the problem, and the third is the target to be optimized. According to this energy function, the dynamic equation of the network is

\[
\frac{du_i}{dt} = -\frac{\partial E}{\partial v_i}, \quad \text{that is} \quad -A(\sum_{j=1}^{n} v_i - 1) - D(\sum_{j=1}^{n} v_i - 1) - \sum_{j=1}^{n} d_{ij} v_j \quad (3),
\]

in the path planning problem, the network input initialization is selected as follows, \(u_i = u_0 \ln(n-1) + \delta_{ui} \quad (4) \quad (x,i=1,2,...,n)\), among them, \(u_0=0.13\); \(n\) are the number of feasible domain points; \(\delta_{ui}\) is a random value in the interval -1 to 1. A is 300, D is 100, the sampling time is set to 0.0002 and the number of iterations is 20000.

The steps to find the shortest path are as follows:

**Step 1:** Enter all nodes of the OpenList collection and calculate the distance between the points;

**Step 2:** Initialization of the network;

**Step 3:** Use formula (3) to calculation \(\frac{du_i}{dt}\), at the same time, using first-order Euler method to calculation.
\( u_{ji}(t+1) = u_{ji}(t) + \frac{du_{ji}}{dT} \Delta T \);

Step 4: Use \( v_i = g(u_i) = \frac{1}{2}[1 + \tanh(g(u_i)/u_0)] \) to calculate \( v_{ai} \);

Step 5: Use formula (2) to calculate energy function \( E \);

Step 6: If the number of iterations is greater than 20000, it terminates, otherwise the number of iterations is incremented and the increment step is one and return to step 3.

5. EXPERIMENT

The experiment was performed in Unity3D-2018, where the red and blue dots are the start and end points, respectively, and the black block diagram is the obstacle. Figure 3 is the result of path planning experiment under the traditional Hopfield neural network, figure 4 is the experimental results of the improved Hopfield neural network path planning.

Table 1 shows the results of three experiments, comparing the path length before and after the algorithm improvement. Table 2 shows the results of three experiments, comparing the length of time before and after the algorithm improvement. From these two aspects, it can be seen that the improved algorithm has improved efficiency and laid the foundation for the path planning of the navigation system.

| Number of experiments | Before improvement | After improvement |
|-----------------------|--------------------|------------------|
| experiment one        | 30.86cm            | 29.32cm          |
| experiment two        | 42.17cm            | 40.11cm          |
| experiment three      | 50.98cm            | 46.10cm          |
| Number of experiments | Before improvement | After improvement |
|-----------------------|--------------------|-------------------|
| experiment one        | 2.01s              | 1.78s             |
| experiment two        | 2.64s              | 2.54s             |
| experiment three      | 3.02s              | 2.39s             |

6. CONCLUSION

In the unity3D simulation environment, a comparison experiment is performed on the algorithm before improvement, that is, the classic Hopfield neural network and the improved algorithm. It can be seen from the experimental results that the improved algorithm achieves better results in terms of time and path length. This experimental result shows that before using Hopfield neural network for path planning, selecting the most favorable node through A-star is efficient for the path planning algorithm. It can shorten the path planning time and effectively avoid obstacles, has reference significance for the application of path planning in practice.

REFERENCES

[1] Nikolos IK, Valavanis KP, Tsourveloudis NC, Kostaras AN. Evolutionary algorithm based offline/online path planner for UAV navigation. IEEE Trans Syst Man Cybern Part B: Cybern 2003;33(6):898–912.

[2] Zhao L, Murthy VR. Optimal flight path planner for an unmanned helicopter by evolutionary algorithms. AIAA guidance, navigation and control conference and exhibit. South Carolina, USA: AIAA; 2007. p. 3716–39.

[3] Krzysztof Chris Ciesielski, Alexandre Xavier Falcão, Paulo A. V. Miranda. Path-Value Functions for Which Dijkstra’s Algorithm Returns Optimal Mapping. Journal of Mathematical Imaging and Vision, 2018, Vol.60 (7), pp.1025-1036.

[4] Tan Baoceng, Cui Jiachao. Application of improved artificial potential field method in obstacle avoidance of unmanned vehicles[J]. Journal of Xi'an University of Technology, 2014,34(12):1007-1011.

[5] Fei Chu, Xiao-ping Ma, Fu-li Wang, Run-da Jia. Novel robust approach for constructing Mamdani-type fuzzy system based on PRM and subtractive clustering algorithm. Journal of Central South University, 2015, Vol.22 (7), pp.2620-2628.

[6] Zhang Ying, Lu Shouyin. Autonomous navigation method for substation inspection robot based on Hopfield neural network[J]. Manufacturing Automation, 2015,37(11):36-39.

[7] Ouyang Ling. Path Optimization Based on Hopfield Neural Network[J]. Journal of Zhongyuan University of Technology, 2017, 28(6): 75-79.

[8] HAI-FEI YU, WEI XIANG. Application of A-Star Algorithm in Robot for Catering Service Based on Laser Radar Navigation. International Conference on Electronic and Information Technology (ICEIT 2017).

[9] Bochuan Jiang, Yueping Wang, Zhiqiang Zhao. Path planning for terrestrial platform based on A-star algorithm. Proceedings of the 3rd International Conference on Mechatronics, Robotics and Automation, 2015.

[10] Lucas Garcia, Pedro M. Talaván, Javier Yáñez. Improving the Hopfield model performance when applied to the traveling salesman problem. Soft Computing, 2017, Vol.21 (14), pp.3891-3905.

[11] Xiyu Liu, Jie Xue. A Cluster Splitting Technique by Hopfield Networks and P Systems on Simplices. Neural Processing Letters, 2017, Vol.46 (1), pp.171-194.