A Tracking Algorithm in Game Based on Neural Network
Peng Zhang
Suzhou Industrial Park Institute of Services Outsourcing, Suzhou, China
zhangp@siso.edu.cn

**Keywords:** Neural network; Tracking algorithm; Game

**Abstract.** Usually, there are a lot of monsters in online games. High IQ monsters will increase the playability of the game. This paper presents a tracking algorithm based on neural network, so that the monster can continue to learn and progress. The monster can track the game player through an optimized route and avoid other monsters or obstacles.

**Introduction**

Most popular online games are MMRPGs (massively multiplayer online role playing game). In these games, players play various roles. There will be many monsters in the game. If the monster is smarter, it is possible to track the player intelligently and evolve to learn, which will increase the player's interest and increase the playability of the game.

In the practical application, the common traversal algorithm can achieve the tracking and avoidance behavior. However, this algorithm has a large amount of computation and consumes more CPU resources. Therefore, a tracking algorithm based on neural network is proposed. The implementation of this algorithm: when the monster appears randomly, the system allocates the original motion path. When the monster finds the player, it calculates the initial path based on the player's position and the obstacles. Then, according to track the player’s efficiency and the number of collisions with obstacles, the monster will train their own neural network. At the same time, the genetic algorithm is used to optimize the structure of neural network.

**Design of the Neural Network.**

The feed forward neural network is adopted in the algorithm. The network includes input layer, hidden layer and output layer. The neural network takes the current coordinates of the monster and the player's target position as the original input. The neural network takes the motion direction of the monster as output. The sigmoid function is used as the excitation function, and the genetic algorithm is used to learn and upgrade. In this algorithm, the number of cells in the hidden layer is set to 10.

In the game, the environment around the monster always changes. Game engines need to train the ability of the monster to recognize the environment, so that they can judge their own path. When the environment has changed, the monster can automatically change the path. To implement this feature, you need to set detectors in the monster object, as shown in Fig. 1.
As can be seen from Fig. 1, when the detector does not detect any obstacle boundaries, the return parameter is -1. It is necessary to detect all the detectors to determine whether the monster is in collision with obstacles.

In addition, the algorithm decomposes the game map into several equal sized squares. The game program stores the grid information through a two-dimensional array. Each element in the array stores information about a map grid, including information about obstacles, whether the monster is passing by, the time to stay here, and so on. Through the evolution of the neural network, the monster is constantly searching for the location of the map has not visited. The information of detectors is also used as the input of the neural network. It is best to standardize all the inputs, so that all input vector values are between -1 and +1. That is, if a grid has not been accessed, its storage information is -1. On the other hand, if it is accessed, the value of the stored information is between 0 and 1. The longer the monster stays in a grid, the greater the value of the stored information.

**Design of Genetic Algorithm.**

The monsters in the game need to complete the training of the neural network in the unsupervised environment. Specific design ideas are as follows.

First of all, according to the independence of the neurons, the program calculates the chromosome segmentation points. Then, according to the point of segmentation, the chromosome cross configuration. Cross rate set to 0.7. Then, the gene in a certain position in the chromosome is updated randomly, and the gene variation factor is set to 0.1. In the choice of path, according to the number of collisions with the obstacles, the fitness function of the genetic algorithm is set up. The fitness of the monster is mainly related to the number of collisions and the angle of rotation. The fitness function is set as follows:

\[ F = F_{\text{collision}} + F_{\text{rotation}} \]

Among them, \( F_{\text{collision}} \) is the adaptation of the monster collision; \( F_{\text{rotation}} \) is the angle of the monster rotation. If no collision occurs, \( F_{\text{collision}} \) adds a unit value. If the monster does not need to rotate or rotate at a small angle (less than a threshold), \( F_{\text{rotation}} \) adds a unit value. In addition, when the collision is detected, the program will punish the monster. At the same time, reward those who can pass through the grid and do not collide with obstacles.

In the genetic algorithm, the tournament selection method is selected. Based on neural network and genetic algorithm, the monster will be a better way to track evolution.

**Experimental Results.**

The simulation results of the algorithm are shown in Fig 2.

![Figure 2. Simulation results](image)

The top line in the graph shows the optimal fitness, which reflects the optimal path finding in the whole swarm. It can be seen that after 90 generations of evolution, the optimal fitness has been significantly improved, and eventually stabilized.
The following line represents the average fitness, which reflects the average search path of the whole swarm. It can be seen that after 300 generations of evolution, the average fitness has improved significantly, and finally stabilized.

Summary

In this paper, a tracking algorithm is proposed, which combines genetic algorithm and neural network algorithm. This algorithm enables the game monster to learn, evolve and adjust the tracking path. The algorithm works well.

Acknowledgements

This research was financially supported by the Jiangsu Province Department of Education Reform Project (No. 2015JSJG396) and the Jiangsu Province Modern Educational Technology Project (No. 2016-R-47504).

References


