An Improved Fuzzy Neural Network for Obstacle Avoidance of Mobile Robot

Tian Gao, Qinglin Wang, Yaping Dai, Zhiyang Jia

School of Automation, Beijing Institute of Technology, Beijing (100081), China
E-mail: gaotian67311@163.com

[Received 00/00/00; accepted 00/00/00]

In order to solve the problem of complex structure and heavy computation load of fuzzy neural network (FNN) in the application of mobile robot obstacle avoidance, an improved fuzzy neural network is proposed. The last two layers in conventional FNN are combined and optimized in the proposed network. By matching the connection weights of the last layer in the network with the central values of the membership functions of the output variables, the number of parameters to be tuned is greatly reduced. The structure and calculation of the network are optimized. The simulation result indicates the effectiveness of the proposed network in obstacle avoidance of mobile robot.

Keywords: fuzzy neural network, obstacle avoidance, mobile robots, multi-sensor

1. Introduction

Fuzzy neural network combines fuzzy logic with neural network. It can be applied to solve the nonlinear problems with high complexity, such as the obstacle avoidance of mobile robot in uncertain environment [1]. In recent years, the obstacle avoidance approach based on the combination of fuzzy logic and neural network has attracted a significant amount of research interest in the field of mobile robots. Kim [2, 3] proposed T1FNN, and designed IT2FNN on the basis of it to achieve position stabilization and obstacle avoidance. Meng Joo [4] realized the adaptive selection of the fuzzy rules with the GDFNN learning algorithm to adjust the structure of the fuzzy controller in FNN. Muhammad [5] realized obstacle avoidance of the robot in unknown environment by combining fuzzy logic and neural network. Tsai [6] proposed fuzzy wavelet neural network to realize obstacle avoidance for a group of Mecanum-wheeled omnidirectional robots with uncertainties. Song [7] divided the obstacle avoidance states of mobile robot into three types, which are described with fuzzy logic, and obtained the weight of each state by fusion neural network.

However, in the previously mentioned methods, the number of parameters in the network will multiply as the number of input variables increases, which can greatly increase the complexity of the network and make the calculation more difficult. In order to simplify the structure and shorten the training time of the network, some improvements are made on the basis of the conventional FNN.

In this paper, an improved fuzzy neural network is established to solve the problem of complex structure and calculation of FNN in the application of mobile robot obstacle avoidance. Compared with the conventional five-layer FNN structure, the improved fuzzy neural network combines and optimizes the last two layers: the normalized layer and the output layer, which can reduce the number of parameters and the complexity of the model. To simplify the network structure and optimize the calculation, the connection weights of the output layer are related to the central values of the membership functions of the output variables, which can avoid the iterative calculation of lots of repeated parameters. The effectiveness of the proposed network is demonstrated by simulation experiments.

This paper is organized as follows: The mobile robot kinematics model is established in Section 2. The overall implementation of the obstacle avoidance method, including the structure of the network, the learning method of parameters and the design of fuzzy controller is introduced in Section 3. Section 4 is the simulation experiment and the analysis of the results. The final section gives the conclusions and future work plans.

2. The Mobile Robot Kinematics Model

2.1. Mobile Robot System Architecture

Mobile robots sense the environment using sensors. The distances between the robot and the obstacles can be measured by six ultrasonic sensors which are incorporated on the robot. They are equipped on the left, the right and in the middle of the front of the robot, which are respectively represented by L, F and R. Each direction has two sensors, which are taken as a group. The smaller measurement result in the two sensors is taken as the distance between the current direction and the obstacle. The direction angle between the robot and the target is obtained in real time by using an electronic compass.

2.2. Kinematic Modeling

Set up the X0Y coordinate system, and the motion state of the robot at a certain moment is shown in Fig. 1.
Fig. 1. Analysis of robot kinematics model.

where \((x, y)\) is the current position coordinate of the robot in the XOY plane. \(\theta\) is the current turning angle, which is the angle between the motion direction and the X axis. The kinematic model can be established as the following:

\[
\begin{align*}
\begin{bmatrix}
\dot{x} \\
\dot{y} \\
\dot{\theta}
\end{bmatrix} &=
\begin{bmatrix}
\cos \theta & 0 & 0 \\
\sin \theta & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\nu \\
\omega
\end{bmatrix}
\end{align*}
\]

If the motion state of the robot at time \(t\) is \((x(t), y(t), \nu(t), \omega(t), \theta(t))\), then the motion state of the robot at the next step \((t+1)\) can be expressed as:

\[
\begin{align*}
x(t+1) &= x(t) + \nu(t) \cos(\theta(t)) \times T \\
y(t+1) &= y(t) + \nu(t) \sin(\theta(t)) \times T \\
\theta(t+1) &= \theta(t) + \omega(t) \times T
\end{align*}
\]

3. Improved Fuzzy Neural Network for Obstacle Avoidance

In this section, the overall structure of the proposed improved fuzzy neural network is firstly described. Secondly, the fuzzy controller is designed. Then, the gradient descent-based learning algorithm is developed to tune the parameters of membership functions. Finally, the training process of the improved fuzzy neural network is shown in the form of the flow diagram.

3.1. Structure of the Improved Fuzzy Neural Network

The structure of the improved fuzzy neural network is shown in Fig. 2. The network consists of four layers: input layer, fuzzy layer, fuzzy inference layer and output layer. There are five inputs obtained from the multi-sensor system: the distances between the obstacle and the robot obtained from the left, front and right sensor groups, the target direction (the angle between current motion direction and the line connecting the robot with the target) and the current speed, which are represented as \(d_l, d_f, d_r, \theta, \nu\). There are two outputs: \(\Delta \theta, \Delta \nu\), which are respectively the angular velocity and the acceleration. The input of the network is denoted by \(X = x_1, x_2, \ldots, x_n\), and \(x_i, i = 1, \ldots, 5\) is the \(i\)-th input variable; the output of the network is denoted by \(Y = y_1, y_2, \ldots, y_m\), and \(y_j, j = 1, 2\) is the \(j\)-th output variable.

The first layer is the input layer, which is used to obtain external environment information from the sensors as the input signal of the network. The number of neural network nodes in this layer depends on the number of actual input signals of the system. The number of nodes in this layer is five.

The second layer is the fuzzy layer, also called the membership function layer. Its main function is to calculate the corresponding membership function value and realize the fuzzy processing for the input variables. The membership function of each input variable can adopt the combination of many kinds of membership functions. The number of nodes in this layer is the sum of fuzzy subsets of each input variable. The fuzzy sets of the five input variables are divided as follows: the distance is divided into two kinds: far and near; the angle is divided into three kinds: left, front and right; and speed is also divided into two kinds: slow and fast. Therefore, the number of nodes in this layer is eleven.

The third layer is the fuzzy reasoning layer, which is used to realize fuzzy reasoning. Each node represents a fuzzy rule, which automatically generates the fitness of the fuzzy rule and replaces the "minimization" operation in the fuzzy control. The number of nodes in this layer is related to the number of fuzzy rules. According to the previous fuzzy set division, the number of nodes in this layer is 48.

The fourth layer is the output layer, which is used for defuzzification to obtain the accurate output value of the system. The number of nodes in this layer is the number...
of actual output signals of the system. The number of nodes in this layer is two.

3.2. Design of the Fuzzy Controller

In this paper, Z-type, S-type functions and triangle functions are chosen as the basic forms of the membership functions. These functions are given as follows:

Z-type function:

\[ p_{ij} = \begin{cases} 
1, & x_j < m_{ij} - \frac{\sigma_{ij}}{2} \\
0, & x_j > m_{ij} \\
\frac{2|x_j - m_{ij}|}{\sigma_{ij}}, & \text{otherwise}
\end{cases} \quad (3) \]

S-type function:

\[ p_{ij} = \begin{cases} 
1, & x_j > m_{ij} + \frac{\sigma_{ij}}{2} \\
0, & x_j < m_{ij} \\
\frac{2|x_j - m_{ij}|}{\sigma_{ij}}, & \text{otherwise}
\end{cases} \quad (4) \]

Triangle function:

\[ p_{ij} = \begin{cases} 
1 - \frac{2|x_j - m_{ij}|}{\sigma_{ij}}, & m_{ij} - \frac{\sigma_{ij}}{2} < x_j < m_{ij} + \frac{\sigma_{ij}}{2} \\
0, & \text{otherwise}
\end{cases} \quad (5) \]

where \( x_i \) represents the \( i \)-th input of the controller. \( m_{ij} \) represents the central value of the membership function. And \( \sigma_{ij} \) represents the width of the membership function. The membership functions of the distance to the obstacle in three directions and the membership function of the motion speed use the combination of Z-type and S-type functions, as shown in Fig. 3 (a) and (c). The membership function of angular velocity is shown in Fig. 3 (b). The membership functions of the angle of the target direction and the acceleration use a combination of these three functions, as shown in Fig. 3 (d).

The first 20 fuzzy control rules are shown in Tab. 1. PB represents the direction angle should be increased faster. PS represents the direction angle needs to be increased slower. Z represents that the direction angle should be increased faster. N represents decreasing the speed. NS represents the direction angle should be decreased slower. Z represents keeping the current speed unchanged. P represents increasing the speed.

3.3. Meanings and Calculation of Tuned Parameters

The parameters to be tuned include each central value of the membership function. They correspond to the weights between the layers in the network.

The connection weights of each node between the first layer and the second layer correspond to the parameters of the membership functions of the input variable \( x_i \), and \( m_{11} = m_{31} = m_{32}, m_{12} = m_{22} = m_{32} \). Assume that the membership function corresponding to the \( i \)-th fuzzy subset of the \( i \)-th input \( x_i \) is \( f_{ij}(x_i) \), then the output of the second layer node is \( p_{ij} = f_{ij}(x_i) \).

The values of the connection weights from the second layer to the third layer are one, indicating that \( w_{ij} = 1 \). So the output of the third layer is:

\[ q_k = \sum_{i=1}^{5} P_i A_i^k, \quad (6) \]

where \( A_i^k \) represents the \( A_i^k \)-th fuzzy subset of the \( i \)-th input corresponding to the \( k \)-th rule.

The center of gravity method is used to defuzzify. The value of the output variables are given as follows:

\[ \begin{aligned}
\Delta \theta &= \frac{\sum_{k=1}^{41} (v_{1,k} \cdot q_k)}{\sum_{k=1}^{41} q_k} \\
\Delta V &= \frac{\sum_{k=1}^{41} (v_{2,k} \cdot q_k)}{\sum_{k=1}^{41} q_k}
\end{aligned} \quad (7) \]

where \( v_{1,k} \) represents the weight of the \( t \)-th output corresponding to the \( k \)-th rule. In this paper, it is the same as the central value of the membership function of the output variables. For example:

\[ v_{1,1} = n_{15}, v_{1,2} = n_{21}, v_{2,1} = n_{15}, v_{2,2} = n_{22} \]

The parameters to be tuned are shown as follows:

\[ P = \begin{cases} 
m_{11}, m_{12}, m_{41}, m_{42}, m_{43}, m_{51}, m_{52}, n_{11} \\
m_{12}, n_{13}, n_{14}, n_{15}, n_{21}, n_{22}, n_{23}
\end{cases} \]

3.4. Gradient Descent-Based Learning Algorithm

In this paper, the gradient descent-based learning algorithm is developed to tune the parameters. The error cost function is given as:

\[ E = \frac{1}{2} \sum_{i=1}^{2} (y_i - \hat{y}_i)^2, \quad (8) \]

where \( \{y_1, y_2\} = \{\Delta \theta, \Delta V\} \) represent the actual output values, which are respectively the angular velocity and acceleration of the actual motion of the robot. \( \{\hat{y}_1, \hat{y}_2\} \) represent the expected output values. Thus, the parameters are updated according to the following formula:
Table 1. Fuzzy rules (only a part)

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>△θ</td>
</tr>
<tr>
<td>1</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>2</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>3</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>4</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>5</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>6</td>
<td>Far</td>
<td>Far</td>
</tr>
<tr>
<td>7</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>8</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>9</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>10</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>11</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>12</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>13</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>14</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>15</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>16</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>17</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>18</td>
<td>Far</td>
<td>Near</td>
</tr>
<tr>
<td>19</td>
<td>Near</td>
<td>Near</td>
</tr>
<tr>
<td>20</td>
<td>Near</td>
<td>Near</td>
</tr>
</tbody>
</table>

\[ P(k + 1) = P(k) - \eta \frac{\partial E}{\partial P}, \quad (9) \]

where \( k \) is the number of iterations and \( \eta \) is the learning rate. The equations for the adaptation of the parameters are given as:

\[ m_{ij}(k + 1) = m_{ij}(k) - \eta_m \frac{\partial E}{\partial m_{ij}}, \quad (10) \]

\[ n_{st}(k + 1) = n_{st}(k) - \eta_n \frac{\partial E}{\partial n_{st}}, \quad (11) \]

where \( \eta_m \) and \( \eta_n \) respectively represent the learning rates of \( m_{ij} \) and \( n_{st} \). Therefore, it is only necessary to calculate the partial derivative of the error cost function with respect to the particular parameter to get the expression for each of them. The derivative results are given as:

\[ \frac{\partial E}{\partial n_{st}} = \sum_{i=1}^{2} \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial v_{ki}} \frac{\partial v_{ki}}{\partial n_{st}} = \frac{(y_k - \hat{y_k}) \sum_{i=1}^{48} q_i}{\sum_{k=1}^{48} q_k} \quad (12) \]

\[ \frac{\partial E}{\partial m_{ij}} = \sum_{i=1}^{2} \frac{\partial E}{\partial y_i} \frac{\partial q_k}{\partial p_{ij}} \frac{\partial p_{ij}}{\partial m_{ij}} = \sum_{i=1}^{2} (y_i - \hat{y_i}) \frac{\sum_{k=1}^{48} v_{ki} \sum_{k=1}^{48} q_k - \sum_{k=1}^{48} (v_{ki} q_k)}{(\sum_{k=1}^{48} q_k)^2} \prod_{s \neq i, j} p_{ij} \times \frac{\text{sign}(x_i - m_{ij})}{\sigma_{ij}} \quad (13) \]

3.5. Training Flow of the Improved Fuzzy Neural Network

Compared with the conventional five-layer fuzzy neural network, the normalized layer and the output layer are combined and optimized in the proposed network. The main training process of the network is as follows: firstly, determine the sample space. Then initialize the weights of the network, set the maximum number of iterations and the desired maximum error value. And calculate the value of the membership function and the output value according to the input. Then update the weights according to the gradient descent learning algorithm, and judge whether the termination conditions have been met, which means the current error or iteration has reached the set value. The flow diagram is shown in Fig. 4.
4. Simulation and Result Analysis

An improved fuzzy neural network is built in Matlab for training to verify the effectiveness of the proposed method. The kinematics model of mobile robot is built by Simulink for the simulation experiment of obstacle avoidance.

4.1. Training Result of the Improved Fuzzy Neural Network

In the simulation experiment, 400 sample data are selected for training, the learning rates of $m_{ij}$ and $n_{st}$ are both set as 0.01. The maximum error value is set as 0.05, and the maximum number of iterations is set as 200. The output error of the proposed network and FNN changes with the training times, as shown in Fig. 5.

![Fig. 5. The relationship between training error and training times.](image)

In the simulation result, the initial error value of FNN is less than that of the improved network, it is because the initial weights $m$ and $n$ are set randomly. Under the selected experimental case, the number of parameters contained in the proposed network is 81, while the number of parameters contained in FNN is 129. And the simulation result shows that the error of the network is reduced to 0.05 after 80 generations of training, while the error of FNN is still above 0.3, which is far higher than the set error value. In the same training times, the training error of the improved fuzzy neural network is smaller than the error of FNN. The training speed of the proposed network is relatively faster. The effectiveness of the improved fuzzy neural network is verified.

4.2. Obstacle Avoidance Simulation

The kinematics model of mobile robot was built by Simulink. A group of statically fixed obstacles are set randomly, and then the initial position and target position of the robot are set. Within the set range, calculate the distance between the obstacle and the current position of the robot, and the direction angle. Judge whether the obstacle is on the left, front or right side of the robot. $d_l, d_f, d_r$ are obtained by calculating the distance of the nearest obstacle to the robot in the three areas of left, front and right. The angular velocity and acceleration are obtained as the output of the network by input these five variables into the network. The new position coordinates are obtained based on the robot kinematics model. The kinematics model built by Simulink is shown in Fig. 6.

![Fig. 6. The Simulation Kinematics Model.](image)

And the simulation result of robot obstacle avoidance is shown in Fig. 7. The mobile robot realized the autonomous obstacle avoidance and the movement to the target. The effectiveness of the improved fuzzy neural network in obstacle avoidance is verified.

![Fig. 7. The trajectory of the robot in simulation experiment.](image)

5. Conclusions

In this paper, an improved fuzzy neural network is proposed to solve the problem of complex structure and large calculation of FNN in the application of obstacle avoidance. By combining and optimizing the normalized layer and the output layer, the number of parameters in the improved fuzzy neural network is greatly reduced, which simplifies the structure of the network. The weights between layers which are related to the parameters of the membership functions are obtained by using gradient descent-based learning algorithm. Through the comparison training experiment of FNN and the proposed network, faster computing speed and simpler net-
work structure of the proposed network is verified. In the set static obstacle environment, the mobile robot can flexibly and accurately avoid obstacles and reach the target position, which shows the effectiveness of the proposed network in obstacle avoidance of mobile robot.

In the future work, the experiment on the mobile robot platform in the actual environment will be considered to test the application effect of fuzzy neural network under more complex environmental conditions. The learning rate of the parameters is fixed in the current training method. In the future work, a learning algorithm that can adaptively adjust the learning rate with the training process will be designed, to further accelerate the training speed and improve the accuracy.

References: