

## 神经网络在智能机器人导航系统中的应用研究

孙 羽<sup>1</sup>, 张汝波<sup>2</sup>

(1 湛江海洋大学工程学院计算机系, 湛江 524088; 2 哈尔滨工程大学计算机系, 哈尔滨 150001)

摘 要: 智能机器人在未知环境中的导航系统主要涉及3个主要方面: (1) 对环境的感知; (2) 局部路径规划; (3) 全局路径规划。文章综述了神经网络技术在这些领域中的应用, 探讨了在神经网络技术中引入学习机制的方法。

关键词: 智能机器人; 神经网络; 导航; 路径规划

## Research on the Application of Neural Network in Intelligent Robots Navigation System

SUN Yu<sup>1</sup>, ZHANG Ruob<sup>2</sup>

(1. Dept. of Computer, Engineering College of Zhanjiang Ocean University, Zhanjiang 524088;

2. Dept. of Computer Science, Harbin Engineering University, Harbin 150001)

【Abstract】Intelligent robots navigation system in unknown environment involves three main problems: perception to environment, local path planning, and global path planning. This paper summarizes the application of neural network in these fields and discusses the method in which learning mechanism is introduced into neural network.

【Key words】Intelligent robot; Neural network; Navigation; Path planning

## 1 神经网络在环境感知中的应用

对环境的感知, 环境模型表示是非常重要的。未知环境中的障碍物的几何形状是不确定的, 常用的表示方法是栅格法。如果用栅格法表示范围较大的工作环境, 在满足精度要求的情况下, 必定要占用大量的内存, 并且采用栅格法进行路径规划, 其计算量是相当大的。Kohonen自组织神经网络为机器人对未知环境的感知提供了一条途径。

Kohonen神经网络是一个自组织神经网络, 其学习的结果能体现出输入样本的分布情况, 从而对输入样本实现数据压缩。基于网络的这些特性, 可采用Kohonen神经元的权向量来表示自由空间, 其方法是在自由空间中随机地选取坐标点 $x(t)$  (可由传感器获得) 作为网络输入, 神经网络通过对大量的输入样本的学习, 其神经元就会体现出一定的分布形式。学习过程如下: 开始时网络的权值随机地赋值, 其后按下式进行学习:

$$m_i(t+1) = \begin{cases} m_i(t) + \alpha(t)(x(t) - m_i(t)) & \forall i \in N_i(t) \\ m_i(t) & \forall i \notin N_i(t) \end{cases}$$

其中 $m_i(t)$ : 神经元 $i$ 在 $t$ 时刻对应的权值;  $\alpha(t)$ : 调整系数;  $x(t)$ : 网络的输入矢量;  $N_i(t)$ : 学习的邻域。每个神经元能最大限度地表示一定的自由空间。神经元权向量的最小生成树可以表示出自由空间的基本框架。网络学习的邻域 $N_i(t)$ 可以动态地定义成矩形、多边形。神经元数量的选取取决于环境的复杂度, 如果神经元的数量太少, 它们就不能覆盖整个空间, 结果会导致节点穿过障碍物区域。如果节点的数量太大, 节点就会表示更多的区域, 也就得不到距障碍物的最大距离。在这种情况下, 节点是对整个自由空间的学习, 而不是学习最小框架空间。节点的数量可以动态地定义, 在每个学习阶段的结束, 机器人会检查所有的路径, 如检测到路径上有障碍物, 就意味着没有足够的节点来覆盖整个自由空间, 需要增加网络节点来重新学习。所

为了收敛于最小框架表示, 应该采用较少的网络节点开始学习, 逐步增加其数量。这种方法比较适合对拥挤的环境的学习, 自由空间较小, 就可用线段表示; 若自由空间较大, 就需要由二维结构表示。

采用Kohonen神经网络表示环境是一个新的方法。由于网络的并行结构, 可在较短的时间内进行大量的计算。并且不需要了解障碍物的过细信息, 如形状、位置等。通过学习可用树结构表示自由空间的基本框架, 起、终点间路径可利用树的遍历技术很容易地被找到。

在机器人对环境的感知的过程中, 可采用人工神经网络技术对多传感器的信息进行融合。由于单个传感器仅提供部分不完全的环境信息, 因此只有采用多种传感器才能提高机器人的感知能力。

## 2 神经网络在局部路径规划中的应用

局部路径规划又称动态避障规划, 是以全局规划为指导, 利用在线得到的局部环境信息, 在尽可能短的时间内, 避开出现的未知障碍物的过程。可以说避障规划是感知空间到行动空间的一种映射。映射关系可以用不同的方法来实现, 但很难用一个精确的数学方程来表示。所以用神经网络来表示是再好不过了。也可在神经网络中引进模糊推理、遗传算法等技术及采用不同类型的神经网络来实现局部规划。采用神经网络就可以产生机器人的避障行为。将传感器的数据作为网络的输入, 由人给定相应场合下期望运动方向角的增量作为网络的输出, 由多个选定的位姿下的一组数据构成原始样本集, 经过剔除重复或冲突的样本等加工处理, 得到最终样本集。通过BP学习获得网络的计算模型, 用于避障。

利用神经网络和模糊数学理论相结合可以实现具有模

作者简介: 孙 羽 (1964~), 男, 副教授, 主要研究方向: 人工智能, 智能机器人, 计算机软件; 张汝波, 博士, 副教授

收稿日期: 2001-04-02

模糊推理功能的局部规划器。首先把机器人传感器的距离信息进行模糊处理。通过总结人的驾驶经验,形成一系列模糊规则,把模糊规则作为样本,对神经网络进行离线训练学习、通过对典型样本的学习,把规则融汇贯通,在整体上体现出一定的智能。所有样本学习完以后,这个神经网络就是一个比较聪明、灵活的模糊规则表,具有一定的自适应能力。在实际应用中允许输入值偏离学习样本,但只要输入模式接进某一学习样本的输入模式,则输出也就接近学习样本的输出模式。这种性质使得神经网络可以模仿人脑在丢失部分信息时仍具有对事物的正确识别能力。

利用神经网络与遗传算法相结合可以实现具有学习能力的避碰控制。通过进化学习实现机器人在非结构化环境中的避碰行为。把网络的权值看作一个参数空间,学习的目的就是在整个空间内寻找最优或满意的解。遗传算法的个体取为网络的所有权值的排列,多个个体作为最后的学习结果。在进化学习中,适应函数取为机器人行进步数,即在障碍环境中走得远的个体具有高的适应值。

### 3 神经网络在全局路径规划中的应用

智能机器人全局路径规划是根据先验知识(如地图等),在有限条件下规划出一条从始点到达目标的无碰路径。全局路径规划可以看成带有约束的优化问题。Hopfield神经网络比较适合用于解决全局路径规划问题。

机器人工作空间用大小相等的栅格进行划分,每个栅格或属于自由空间,或属于障碍空间,并都对应于Hopfield神经网络中的一个神经元 $q_i$ 。该神经元的激活值是区间 $[0, 1]$ 中的实数。神经网络中,神经元 $i$ 到 $j$ 的连接强度用 $T_{ij}$ 来表示。在该模型中,连接值是兴奋的( $T_{ij} > 0$ ),对称( $T_{ij} = T_{ji}$ ),而且是有范围的:

$$T_{ij} = \begin{cases} 1 & \text{if } \rho(i, j) < r \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

这里,  $r$  是一个正数,  $\rho(i, j)$  是神经元 $i$ 和 $j$ 所对应的区域的中心在机器人工作空间中的欧式距离。

神经元的状态变量 $\sigma_i$ ,  $i=1 \dots N$ 可由网络中其它神经元的输入和外部传感器信息的输入而改变。神经元 $i$ 的总输入 $u_i$ 是其它神经元激活值的加权和与外部传感器输入的和:

$$u_i(t) = \sum_j T_{ij} \sigma_j(t) + I_i \quad (2)$$

该网络的演进可用离散时间或连续时间的动态特性来定义。在离散时间情况下,神经元 $i$ 的激活值由下式更新:

$$\sigma_i(t+1) = g(u_i(t)) = g\left(\sum_j T_{ij} \sigma_j(t) + I_i\right) \quad (3)$$

这里  $g(x)$  是一个Sigmoid函数。

在连续时间更新中,所有的神经元都是连续地并同时地改变其状态。神经元活动值的变化可以由一个非线性微分方程组来表示:

$$\frac{d\sigma_i(t)}{dt} = g\left(\sum_j T_{ij} \sigma_j(t) + I_i(t)\right) - \sigma_i(t) \quad (4)$$

为构造一条可行路径,可使用由(2)、(4)所定义的离散或连续系统。在模拟中,可选用简单的离散方法。在起始时间,  $t=0$ , 所有神经元的活动值都被置为0。在外部输入的影响下,目标点神经元被置为1且占用区的神经元被置为0,系统状态开始根据(3)所描述的网络动态特性而改变。网络的演变可以看作是在相位空间中一个点沿Liapunov函数减小的曲线上的运动。它在网络达到局部或全局最小的平衡状态而结束。网络的状态不会摇摆不定。最后的平衡状态是稳定点方程组的解集:

$$\sigma_i^* = g\left(\sum_{j=1}^N T_{ij} \sigma_j^* + I_i\right) \quad (5)$$

其中,  $i=1 \dots N$ 。当Liapunov函数是严格凸时,这个平衡状态是唯一的。

考虑只有静态障碍物和目标点的情况。如果在初始状态,起点神经元 $j_{\text{init}}$ 是活动的,那么一条连接起点和终点的路径可以用下面的方法来构造。机器人的新位置由周围有最大活动值的神经元给出,并作为下一步的起点。每次达到一个新位置时,就做如何达到下一位置的决策。这个过程重复下去,直到机器人实际位置对应的神经元成为目标神经元。所构造的路径通过一个节点序列  $j_{\text{init}} = j_0, j_1, \dots, j_{\text{target}} = j_m$ 。该路径的长度由下式给出:

$$D(P) = \sum_{i=1}^{m-1} \rho(j_i, j_{i+1}) \quad (6)$$

这里  $\rho(j_i, j_{i+1})$  等于栅格常数。因为这条路径总是连接栅格中一个节点和与其相邻有最大权值的下一节点,并在活动值为1的节点结束,所以其步数和长度都是最小的。所有其它的通过一些活动值不是递增的或不是以最佳方式递增的节点的路径都有更大的步数和长度。由于位形空间的离散化,路径 $P$ 只能位于栅格的节点上,而且只是一条近似的光滑路径,其精确度局限于栅格的大小。

Hopfield神经网络用于全局路径规划中另一方法是把全局规划看成带约束的优化问题。用激活两个神经元 $a_1$ 和 $a_2$ 来跟踪真实二维空间中的轨迹,定义在 $t=t_1$ 时网络到达的终点为 $(c_1, c_2)$ ,即:

$$a_1(t_1) = c_1 \quad \text{和} \quad a_2(t_1) = c_2 \quad (7)$$

在 $[t_0, t_1]$ 内被 $[a_1(t), a_2(t)]$ 跟踪的轨迹长度由下式给出:

$$\int_{t_0, a_1(t_0)}^{t_1, a_1(t_1)} ds = \int_{t_0}^{t_1} \sqrt{\dot{a}_1^2(t) + \dot{a}_2^2(t)} dt \quad (8)$$

到达终点时的误差为:

$$(a_1(t_1) - c_1) \text{ 和 } (a_2(t_1) - c_2) \quad (9)$$

若障碍物是以 $(o_1, o_2)$ 为中心,  $R_i$ 为半径的圆,避碰的表达式为:

$$g(a_1, a_2) = R_i^2 - [(a_1(t) - o_1)^2 + (a_2(t) - o_2)^2] \leq 0 \quad (10)$$

问题就变成调整权值 $T_{ij}$ 连续来满足3个目标:(1)从起点到达终点;(2)找到最短路径;(3)不通过障碍区域。这个问题可用古典优化技术来描述:求一个最小值即路径长度。两个约束:到达终点和避碰。由Hopfield网络是一个一阶微分方程系统,只要加上初始条件,最终轨迹就会满足约束条件。

