

初始权值优化技术在机器人学习中的应用

肖 伟, 周东辉, 孙建风, 徐志强
(中国海洋大学 电子系, 山东青岛 266071)

摘 要: 针对移动机器人建立了基于BP神经网络的智能避障控制模型, 提出了初始权值优化技术, 使得样本组与初始权值相匹配, 显著地提高了网络的收敛速度. 为了提高系统的实时性, 文中采用C和汇编语言混合编制控制程序. 计算机仿真和实测结果表明该系统具有学习能力强、人机交互效果好等优点.

关键词: 移动机器人; 避障; 神经网络; BP算法; 初始权值优化技术

中图分类号: TP24 文献标识码: A 文章编号: 0372-2112(2005)09-1720-03

Implementation of Optimal Weights Initialization Technology in Robot Learning

XIAO Wei, ZHOU Dong-hui, SUN Jianneng, XU Zhiqiang
(Dept. of Electronics, Ocean University of China, Qingdao, Shandong 266071, China)

Abstract: An intelligent obstacle avoidance model based on BP neural network is established. Also a novel optimal weights initialization technology is proposed so that the sample sets and initial weights can match perfectly. Consequently, the convergence speed increases evidently. In order to improve the real time performance, hybrid programming using C and assemble language is adopted. Computer simulation and real test show that the system has a strong ability of learning and good performance of human computer interaction.

Key words: mobile robot; obstacle avoidance; neural network; BP algorithm; optimal weights initialization technology

1 引言

在机器人研究领域, 机器人学习一直是研究的热点和难点问题. 神经网络技术不依赖精确数学模型, 在处理非线性问题上的显著优势, 常常用来处理人类经验性智能问题. BP网络是目前应用得最为广泛的一种人工神经网络. 在现实的应用中, 用来解决实际的问题, 成绩斐然, 然而, 标准BP网络存在收敛速度慢, 容易陷入局部极小值等缺点^[1~5,7]. 尤其是收敛速度慢, 导致系统的很多资源被耗费. 因此, 采用一种算法来优化初始权值, 使其和样本组相匹配, 就显得很必要. 本文采用了基于神经网络技术以实现移动机器人的智能避障控制的模型, 在BP算法的基础上, 提出了初始权值优化技术, 以加快网络收敛的速度, 提高系统的实时性. 文中采用C和汇编语言混合编程, 开发基于神经网络的控制软件, 并且在凌阳16位单片机上实现了人机交互和移动机器人的智能避障. 该方法对机器人避障的学习以及提高其智能化是很有意义的.

2 初始权值优化技术

在BP算法中, 初始的权值有两种方式可获得. 一种是直接给定一组确定的数(例如: 所有的权值皆初始化为0.5);

另一种是比较常用的, 以任意的一组随机数作为初始权值. 本文采用随机数的方式, 而这组初始的随机权值, 类似此网络的“天分”, 它的优劣(即此组初始权值对输入向量与目标向量的匹配能力)直接会影响到网络的收敛. 因此, 采用一种算法来优化初始的权值(即选择匹配能力强的组)就很必要. 当网络初始化权值皆为一个值(如0.5)时, 此时的离散度为零, 经测试, 收敛速度很慢(结论部分有测试结果), 而采用一组随机数初始化权值, 其收敛速度明显加快. 因此, 权值的离散度与收敛存在着联系, 同时, 构造优化函数必须考虑与样本或者误差之间的关系, 以便找到与样本组相匹配的初始权值组.

考虑构造函数来描述初始化权值的离散度, 首先定义单个神经元的权重离散度, 以下是第k层第i个神经元的权重离散度, 为神经元相邻的两权重之差的绝对值.

$$J_i^k(w) = \sum_{j=1}^N |w_{i,j} - w_{i,j+1}| \quad (1)$$

同时, 我们定义函数第k层神经元之间纵向离散度:

$$J^k(w) = \sum_{j=1}^N \sum_{i=1}^M |w_{i,j} - w_{i+1,j}| \quad (2)$$

在实际的测试中, 神经元的权重离散度和神经元之间的离散度增加时, 收敛的速度加快了. 进一步, 为了构造判断优化的函数, 同时考虑到总误差, 我们构造函数:

$$J_i(w) = \sum_i^q J_i(w) \quad (3)$$

$$J_i(w) = \left(\sum_{k=1}^u \sum_{i=1}^M J_i^k(w) + \sum_{k=1}^u J^k(w) \right) * \frac{1}{\lambda E_i(w)} \quad (4)$$

其中 λ 称为优化因子, q 代表样本数, i 代表优化的次数, $E(w)$ 代表误差, 显然, 我们应该选择使优化函数 $J_i(w)$ 最大的权重为网络的初始值。

$$J(w) = \text{Max}_{i=1,2,\dots} (J_i(w)) \quad (5)$$

3 软件模拟与仿真

我们选择 λ 为 0.9, 测试优化函数与样本误差之间的关系, 直接在主控单片机的编译环境下进行调试, 得到如下数据 (见下表 1):

表 1 优化函数和样本误差表

| | | | | | | | |
|-------|---------|---------|---------|---------|---------|---------|---------|
| 优化函数值 | 74.6997 | 69.7855 | 64.2555 | 57.3984 | 178.735 | 37.2208 | 26.8906 |
| 样本组误差 | 1.3254 | 1.0757 | 1.5780 | 1.7725 | 0.5538 | 2.4865 | 3.5904 |
| 优化函数值 | 51.1329 | 50.5268 | 535.167 | 81.0372 | 68.3946 | 51.2168 | 65.9117 |
| 样本组误差 | 1.8233 | 1.8504 | 0.1541 | 0.9889 | 1.3130 | 1.5549 | 1.2129 |

为了进一步直观地观察优化函数与样本组误差之间的关系, 用上面的数据绘出图 1 的曲线:

由右图我们可以看出, 优化函数与样本组误差基本成单调递减的关系, 因此, 此构造的优化函数可以用来选择与样本组最为匹配的初始权值。

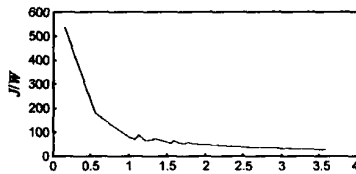


图 1 优化函数与误差关系曲线图

4 系统的实现

4.1 硬件实现框图

图 2 是硬件实现框图, 分四部分, 第一部分是传感器部分, 用来感知外界的环境; 第二部分是语音部分, 由语音采样输入, 识别^[6], 以及无线话筒组成; 第三部分是电机驱动部分, 使用了专用的直流电机驱动芯片。

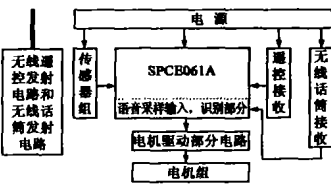


图 2 硬件实现框图

第四部分是发射控制器, 集成了遥控和无线话筒发射电路。

机器人在进入学习的过程之前, 必须经历语音训练的过程。由操作者的发出语音来训练机器人。这一过程实际是完成了操作者的语音和目标动作配对的过程。例如: 训练“前进”时, 机器人会首先发出提示标准音“前进”(这是预先存入 flash 的标准音), 操作者说“前进”两遍, 如果训练成功, 机器人会有相应的提示。训练完成后, 机器人仅识别操作者的已训练过的语音并执行相应的动作。

其学习的过程是非常有趣的, 把智能移动机器人放到一个杂乱无章的环境中行走, 由操作者的语音控制其动作, 每命令其一次, 机器人执行相应的动作并且采一次样作为样本/输

入向量为传感器组的值, 目标向量为语音识别后的目标动作), 直到它遍历完所有的情况之后, 一组完备的样本组就收集完成。然后由遥控使其进入训练状态。训练完成后, 机器人就能在此杂乱无章的环境中来去自如的行走。

4.2 软件部分

图 3 是软件训练部分程序流程图。对应建立网络, 优化初始权值, 学习训练的整个过程。而对于使用来说, 只需经历计算隐层神经元输出以及进而计算输出神经元输出的较少的计算即可。实现时必须定义一个接口函数把传感器的值读入 BP 网络。训练部分代码如下:

```

.....
//判断优化次数
for (k = 1; k < Optimal
Times; k++ )
{
//随机初始化权值
initWeights();
//读入样本组(语音编码
信号和传感器信号)
initData();
//计算样本组误差
calcOverallError();
//计算优化函数离散度因子
calcJ();
//计算并保存优化函数的最终值
finalValue();
//与上一次优化函数的值比较
if(TotalNeural > LastNeural)
{
//保留优化函数值的大者
storeValue();
//保留优化函数值的大者所对应的权值
storeWeight();
}
}
//恢复这几次中的最大者所对应的权值
recoveryWeights();
while(1)
{
//判断误差是否达到要求
if(e < Error) break;
//判断次数是否超出
if(i > Traintimes) break;
//按 BP 算法训练网络
trainBP();
}

```

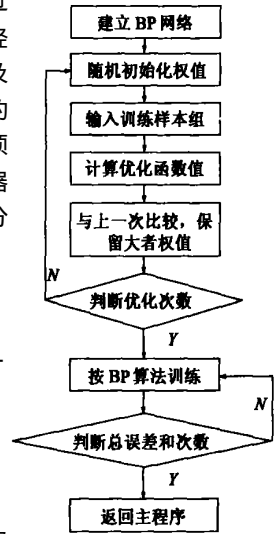


图 3 软件训练流程图

.....

5 测试结果

5.1 实时性问题

在单片机上实现 BP 网络,这对单片机性能的要求是很高的,尤其是对运算的速度.否则,训练要花的时间会很长.标准的 BP 算法,学习两个指令需要的间为 134 秒.而经过初始权值优化后的 BP 网络只需要 87 秒.对智能移动机器人来说,学习完成后响应外界的环境并做出相应的动作所需时间是很重要的,在本文中,即经历前两步计算最终得出网络的输出所需时间.本文的智能机器人不到 1 秒的时间即可相应并完成相应的动作,完全能满足实际的需要.

5.2 收敛问题

以下是一组标准 BP 算法和优化初始权重的训练记录.下表是采用标准 BP 算法,并且在初始化权值皆为 0.5 时的收敛记录,其收敛的速度慢,计算量非常大:

表 2 初始化权值皆为 0.5 时的收敛记录

| | | | | | | | |
|------|--------|--------|--------|--------|--------|--------|--------|
| 总误差 | 1.5403 | 0.9987 | 0.9966 | 0.9956 | 0.9943 | 0.9921 | 0.9885 |
| 训练次数 | 1 | 10 | 30 | 40 | 50 | 60 | 70 |
| 总误差 | 0.9804 | 0.9516 | 0.9292 | 0.6658 | 0.5090 | 0.3890 | 0.2293 |
| 训练次数 | 80 | 90 | 94 | 100 | 102 | 104 | 110 |

以下是标准的 BP 网络,经过随机初始化后的记录:

表 3 随机初始化后的收敛记录

| | | | | | | | |
|------|--------|--------|--------|--------|--------|---------|--------|
| 总误差 | 2.3824 | 1.3331 | 0.8198 | 0.7463 | 0.6892 | 0.63885 | 0.5909 |
| 训练次数 | 1 | 4 | 8 | 10 | 12 | 14 | 16 |
| 总误差 | 0.5457 | 0.5028 | 0.4620 | 0.3501 | 0.2806 | 0.2462 | 0.2118 |
| 训练次数 | 18 | 20 | 22 | 26 | 30 | 32 | 34 |

以下是采用了初始权值优化技术后的 BP 网络收敛数据:

表 4 初始权值优化技术后的收敛记录

| | | | | | | | |
|------|--------|--------|--------|--------|--------|--------|--------|
| 总误差 | 0.0996 | 0.1098 | 0.1335 | 0.1434 | 0.1375 | 0.1203 | 0.0973 |
| 训练次数 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 总误差 | 0.0732 | 0.0511 | 0.0329 | 0.0191 | 0.0096 | 0.0040 | 0.0027 |
| 训练次数 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |

显然,优化后的权重更适合样本组,收敛的速度更快一些,并且在不到 15 次的训练数中,总误差收敛已经提高两个量级.初始值优化一次的计算量与训练 BP 网络一次的计算量基本一样,在此实验中,我们优化了 15 次,也就是说,我们的计算量与训练 29 次相等,而此时的收敛程度比在同样次数训练的采用随机初始化的 BP 网络,已经提高了两个量级.

6 结论与展望

本文提出了初始权值优化技术,并结合 BP 算法,利用凌阳单片机(SPCE061A)语音处理能力,实现机器人与人之

间的互动学习,显著地节省了计算量,是神经网络技术很好的一个实际应用.该技术使得神经网络技术能够更好地应用于实际中去.机器人模仿人的学习过程,首先传承自身的较优天分(通过初始权值优化技术获得),按一定的规则(BP 算法)自我调整(修改权重),以适应外界环境(由语音识别结果和传感器组成的样本组之间的匹配),把学习的智慧分布的存储在大脑(连接权重)中,从而处理实际的问题.同时,也应看到,硬件实现神经网络对系统资源的要求是很高的,尤其是运算的速度,在以后的工作中,需寻找计算量更小,收敛速度更快的算法以缩短训练时间,提高实时性.

参考文献:

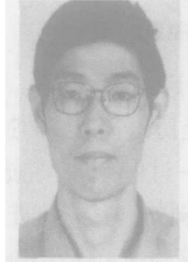
- [1] 刘增良. 模糊逻辑与神经网络[M]. 北京: 北京航空航天大学出版社, 1996. 116- 118.
- [2] 阎平凡, 张长水. 人工神经网络与模拟进化计算[M]. 北京: 清华大学出版社, 1999. 22- 23.
- [3] 袁曾任. 人工神经网络及其应用[M]. 北京: 清华大学出版社, 1999. 66- 78.
- [4] 边肇祺, 张学工. 模式识别[M]. 北京: 清华大学出版社, 1999. 254- 257.
- [5] 冯天瑾. 神经网络技术[M]. 青岛: 青岛海洋大学出版社, 1994. 80- 83.
- [6] 罗亚非. 凌阳 16 位单片机应用[M]. 北京: 北京航空航天大学出版社, 2003. 241- 282.
- [7] Simon Haykin. 神经网络原理[M]. 北京: 机械工业出版社, 2003. 160- 165.

作者简介:



肖伟男, 1974 出生于湖北武汉, 硕士研究生, 主要研究方向: 机器人控制, 人工智能, 多机器人之间通信与协调.

E-mail: bestxiaowei@people.com.cn.



周东辉 男, 1963 出生于山东青岛, 博士, 教授, 主要研究方向: 机器人技术, 深海换测技术.