

人工神经网络在六子棋机器博弈中的应用

李一波¹, 安涌², 乔志华¹

1 沈阳航空工业学院自动控制系 沈阳 110136

2 沈阳航空工业学院计算机学院 沈阳 110136

E-mail: lyb20040612@yahoo.com.cn bravean@yahoo.com.cn helloqiao2002@163.com

摘要: 人工神经网络具有自学习的特性, 利用这个特性, 提出将人工神经网络与机器学习结合起来应用于六子棋机器博弈的方法, 以提高博弈程序的智能性。通过应用该方法的六子棋程序与一般的六子棋程序的比较, 表明该方法在某些方面具有一定的优越性, 为人工神经网络在军事、经济等方面的应用提供了有价值的依据。

关键词: 机器博弈 人工神经网络 反向传播算法 六子棋

Application of Artificial Neural Network to Playing Connect-6

LiYibo¹, An Yong², Qiao Zhihua¹

1 Department of Auto-Control, Shenyang Institute of Aeronautical Engineering,
Shenyang, 110136, China

2 School of Computer Science, Shenyang Institute of Aeronautical Engineering,
Shenyang, 110136, China

E-mail: lyb20040612@yahoo.com.cn bravean@yahoo.com.cn helloqiao2002@163.com

Abstract: Artificial Neural Network has the character of self-learning. To improve the intelligent program, this paper presents a method combining Artificial Neural Network with machine learning to Connect-6 computer game. By comparing this method with the general one, it proves that this method has some advantages. It is valuable for application of Artificial Neural Network to military and economy.

Keywords: Computer game; Artificial Neural Network; Backpropagation algorithm; Connect-6

1. 引言

近年来, 随着人工智能学科的兴起, 机器博弈作为人工智能的一个重要分支, 越来越受到众多学者和博弈爱好者的关注。早在半个多世纪之前, 信息论的创始人 C. E. 香农教授就提出了为象棋博弈编程的方案, 成为机器博弈的创始人。国内外许多知名学者和科研机构都曾经涉足这方面的研究, 历经半个多世纪, 到目前为止已经取得了许多惊人的成就。

半个世纪以来, 博弈的算法层出不穷, 例如, 极大极小算法, α - β 剪枝搜索及其增强算法、迭代深化、历史启发等等。随着博弈算法的不断发展, 对算法搜索的效率要求也越来越高。仅仅使用已有的算法已经不能满足博弈程序对效率的高要求了。这样, 就需要将现有的搜索算法与其他的方法结合, 以适应更高要求的搜索效率。人工神经网络是人工智能领域

中一个重要的方法，也是促进人工智能学科发展的重要技术。国内外已经有很多专家学者将神经网络应用到机器博弈中，比如美国德克萨斯大学将神经网络应用于 Othello^[1]，国内也有将其应用到五子棋的研究^[2]等。

六子棋，英文名称是 Connect6，是台湾国立交通大学资讯工程系的吴毅成教授于 2003 年夏提出并开始研究开发的，是由五子棋演变而来的。六子棋由两位棋手一方执黑子一方执白子在 19×19 的棋盘上进行着手的竞技项目。简单规则为：黑先白后；执黑者第一手放一个黑子于棋盘上，之后双方轮流着手，每手放两个棋子于棋盘上；棋局中先完成于直、横或斜向上连六或连六以上者获胜。

经过了近几年的发展，许多博弈算法已经被应用到六子棋的程序开发中。但是尚未有人将神经网络应用于六子棋机器博弈的研究中。本文介绍了将神经网络应用于机器博弈，并以六子棋为研究对象对该算法进行了测试，为机器博弈的发展提供了新的思路。

2. 神经网络

神经网络 (Artificial Neural Networks, ANN)，一种模拟动物神经网络行为特征，进行分布式并行信息处理算法的数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。神经网络具有自学习和自适应的能力，可以通过预先提供的一批相互对应的输入—输出数据，分析掌握两者之间潜在的规律，最终根据这些规律，用新的输入数据来推算输出结果，这种学习分析的过程被称为“训练”。

本文介绍的方法使用的是前馈神经网络。前馈神经网络是指各神经元接受前一层的输入，并输出给下一层，没有反馈(图 1)。结点分为两类，即输入单元和计算单元，每一计算单元可有任意个输入，但只有一个输出。通常前馈网络可分为不同的层，第 i 层的输入只与第 i-1 层输出相连，输入和输出结点与外界相连，而其他中间层则称为隐藏层。

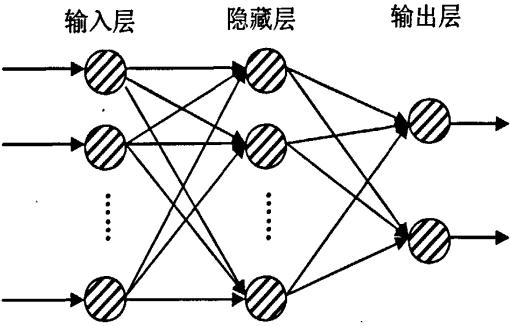


图 1 前馈网络

反向传播算法(BP 算法)是由 Rumelhart, Hinton, Williams 于 1986 年提出的。应用 BP 算法 (Backpropagation algorithm) 的前馈神经网络亦称之为 BP 网络。BP 网络系统地解决了多层网络中隐含单元连接权的学习问题。

3. 棋类对弈学习系统设计

设计棋类学习系统的步骤包括以下几个方面^[2]：

(1)选择训练经验

系统通过和自己对弈来训练，不需要外界的训练者，只要时间允许，可以让系统产生无

限多的训练数据。

(2)选择目标函数

采用增强学习的 Q 算法。增强学习要解决的问题：一个能够感知环境的自治 agent，怎样通过学习选择能达到其目标的最优动作。Q 算法是一个具体的增强学习算法，它可从有延迟的回报中获取最优控制策略。在状态 s 下的最优策略

$$\pi^*(s) \text{ 为: } \pi^*(s) = \arg \max_a Q(s, a), \quad (1)$$

其中

$$Q(s, a) \equiv r(s, a) + \gamma \max_a Q(\delta(s, a), a)^{[3]}. \quad (2)$$

在博弈中，令 $V(s_a) = Q(s, a)$ (s_a 表示系统在棋局状态 s 采用走子 a 后得到的棋局状态)。用符号 \hat{V} 来指代系统对实际函数的估计，相应的估计训练法则为：若 s_a 为一最终棋局，则 $\hat{V}(s_a) \leftarrow r(s, a)$ ；若不为最终棋局，则

$$\hat{V}(s_a) \leftarrow r(s, a) + \gamma \hat{V}(\text{Successor}(s_a)) \quad (3)$$

其中 $\text{Successor}(s_a)$ 为相对于状态 s 的后继状态。

至此，该学习系统要学习的目标函数应为 \hat{V} ，下棋策略为选择是 \hat{V} 值最大的走子。

(3)选择目标函数的表示

采用包含两层的前馈网络来表示要学习的目标函数。

(4)选择函数逼近算法

采用前馈网络的反向传播算法来调整权值。反向传播算法采用梯度下降方法试图最小化网络输出值和目标值之间的误差平方。网络误差定义公式是对所有网络的误差求和：

$$E(\tilde{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{output}} (t_{kd} - o_{kd})^2. \quad (4)$$

(5)最终设计

学习系统的最终设计可以用图 2 所示的 4 个程序模块来描述^[3]。

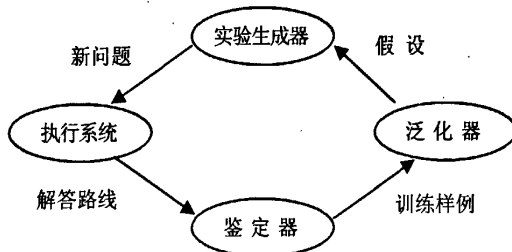


图 2 学习系统的最终设计

执行系统——这个模块是用学会的目标函数来解决给定的任务。在这里，执行系统选择

下一步走法的策略是由学到的函数来决定的。因此，该执行系统的性能会随着函数的日益准确而提高。

鉴定器——以对弈的路线或历史记录作为输入，输出目标函数的一系列训练样例。

泛化器——以训练样例为输入，产生一个输出假设，作为它对目标函数的估计。

实验生成器——该模块产生新问题供执行系统去探索。它的作用是挑选新的实践问题，以使整个系统的学习速率最大化。

4. 测试与分析

我们以六子棋为例对程序进行测试。采用前馈神经网络，隐藏单元取 10~15 个，学习速率 η 取 0.3，神经网络的输入是棋局状态。棋局状态编码为 19×19 的矩阵，矩阵元素取值范围为 0~2 的整数，该位为空时为 0，该位置为黑子时为 1，该位置为白子时为 2。每一个矩阵元素对应一个网络输入。然后把范围是 0~2 的取值按比例线性缩放到 0~1 的区间，以使网络输入与隐藏单元和输出单元在同样的区间取值。神经网络会输出一个实数值来表示最终

的状态 \hat{V} ，若输出值为 1，则此时的状态 s 为最终的胜局；若输出值为 0，则此时的状态 s 为最终的负局；若输出值为 0.5，则此时的状态 s 为最终的和局。我们可以用三个输出单元来表示最终的棋局状态：黑棋胜(1,0,0)；白棋胜(0,0,1)；和局(0,0.5,0)。但是由于 sigmoid 单元对于有限权值不能产生 0 和 1 的输出(如果使用 0 和 1，则梯度下降将迫使权值无限增长)，所以不能用 0 和 1 来编码。在这里我们用 0.1 和 0.9 来替代 0 和 1，那么我们使用的三个输出单元编码为：(0.9,0.1,0.1)表示黑棋胜，(0.1,0.1,0.9)表示白棋胜，(0.1,0.9,0.1)表示和局。

首先让程序进行自我学习，得到一个函数。然后与现有的六子棋程序进行对弈，可以发现程序自我学习的次数较少时，与现有的程序相比没有什么优势；当自我学习到一定次数得到的函数与其对弈时，水平会略高于现有程序；但学习的次数不能无限的增加，当学习的次数超过某一数值的时候，性能有很小的改进，但花费的时间却很长。测试数据如下表所示。

表 1 测试结果

学习次数	对应的 50 次测试盘数		
	胜	负	平
10000	2	47	1
50000	23	22	5
100000	26	21	3

5. 结论

通过自我学习得到的某一函数，用来下六子棋，水平会略高于现有的六子棋程序。其它的六子棋算法一般采用搜索最佳位置的方法，搜索得越深，所用的时间就越长。而本方法的特点就在于学习目标函数要花费很长的时间，但是一旦得到目标函数，用它来下棋只需要很短的时间。这种人工神经网络应用于博弈的方法对于军事、经济等方面也有一定的借鉴作用。

参考文献

[1] David E.Moriarty, Risto Miikkulainen, Evolving Neural Networks to Focus Minimax Search[J]. The University of Texas.

[2] 谢磊 鲁汉榕 徐新. 增强学习与人工神经网络结合在对弈中的应用[J]. 空军雷达学院学报, Vol.18 No.4, 2004.

[3] Mitchell T.M. 机器学习[M]. 曾华军, 张银奎等译. 机械工业出版社, 2003.