基于神经网络的Sarsa强化学习算法

林联明，王 浩，王一雄
（合肥工业大学 计算机与信息学院，安徽 合肥 230009）

摘 要：标准的Sarsa算法对状态空间的要求是离散的且空间较小，而实际问题中很多的系统的状态空间是连续的或尽管是离散的但空间较大，这就要求有很大的空间来存储状态动作对（State_Action_Pair）。对此文中提出用BP网络队列保存SAPs，实验验证可以解决由于空间过大而带来的 Q 值表示问题。

关键词：强化学习；智能主体；马尔可夫决策过程；误差后向传播网络；状态动作对
中图分类号：TP301.6 **文献标识码**：A **文章编号**：1005－3751(2006)01－0030－03

Sarsa Reinforcement Learning Algorithm Based on Neural Networks

LIN Lian-ming, WANG Hao, WANG Yi-xiong
(Computer and Information Faculty, Hefei University of Technology, Hefei 230009, China)

Abstract: The standard Sarsa algorithm requires that the state space is discrete and small. However, in real environment it does not satisfy that due to the fact that it may be continuous or discrete but has big space state, so it needs too memory to keep State－Action－pair (SAPs). This paper proposes to use BP queue to store SAPs. The experiment shows it can resolve the problem that how to represent Q values in case of big state space.

Key words: reinforcement learning; agent; MDP(Markov decision process); BP(back propagation); SAP(state_action_pair)

1 强化学习

强化学习技术是从控制论、统计学、心理学、认知学等相关学科发展而来的，有着相当长的历史，但直到20世纪80年代末、90年代初强化学习技术才在人工智能、机器学习中得到广泛研究。由于强化学习是在线学习且具有无导师的自适应能力，因而被认为是设计智能Agent的核心技术之一。其原理是智能体通过与环境不断进行相互作用来达到获得知识和适应环境的学习目的。“强化”一词最早来自实验心理学中动物学习的研究，其含义是指在特定情形下动物所采取的行动，若带来一系列满意的后果，则动物在同样的情形下采取该动作的可能性就得以增强，反之则减弱，具有这种增强趋势称之为“强化”。强化学习的目的是如何行动（如何给出一个状态到行动的映射），以能最大化数值强化学习信号。不象其他大多数机器学习方法，学习者没有被告知采取哪一个行动，而是必须发现采取哪些行动能带来最大化报酬。标准的强化学习框架如图1所示。

从效果上来看强化学习是一个正反馈的过程。标准的Agent强化学习框架由3个模块组成：输入模块I、强化模块R和策略模块P。其中输入模块I将环境状态映射为Agent的感知 o（通常若为恒等函数，即 $o = S$，此时称状态是完全可观的；当 $o \neq S$，即 o 不是恒等函数时，称状态是部分可观的），强化模块R根据环境状态的迁移来赋予Agent奖赏值 r；策略模块P更新Agent的内部知识，同时使Agent根据某种策略选择一个动作作用于环境。强化学习需要定义一个目标函数来评估从长远看动作是否是最优的。通常以状态的值函数（Value function）或状态－动作对的值函数体现此目标函数。在Agent与环境每一次的交互过程中，Agent接受环境状态 S 的输入，并映射为Agent的感知 o。Agent选择行为动作 a 作为对应环境状态的输出，行为动作将导致环境状态 S 变迁，同时Agent接受环境的奖惩信号 r。Agent的目标是在每次决策时，尽量使选择的行为能够获得最大的价值 v（或数值强化信号标量），也就是最大限度地逼近目标。如果Agent的某个行为策略能够获得环境的较高奖赏，那么这个Agent以后选择策略 n 进行行动的趋势便会加强，反之Agent以后选择策略 n 的趋势便会减弱。

文中主要介绍强化学习算法与神经网络相结合的基本技术。在众多强化学习算法中 Q 学习和Sarsa学习是两个重要的算法，前者的 Q 值更新策略是一种离策略，而后者采取的是在策略，一般在策略比离策略要好。但标准的Sarsa算法对状态空间的要求是离散的且空间数较小，而实际中很多系统的状态空间是连续的或尽管是离散的但空间数较大这就要求有很大的内存空间来存储SAPs，传统的方法使用查表法来保存 Q 值，对此文中提出用一种

收稿日期：2005－04－29
作者简介：林联明（1980—），男，安徽潜山人，硕士研究生，研究方向为多媒体与数据库技术；王 浩，教授，研究方向为人工智能与数据挖掘。

"紧凑"的表示方法即用BP网络队列保存SAPs。对于扩张和利用的平衡常采取的探索策略是 ξ - greedy 策略,即以 ξ 的概率搜索新的策略,以 $1-\xi$ 的概率利用已有的策略,但是其缺点是当学习在收敛的时候仍然会以 ξ 的概率搜索新策略浪费了学习时间,针对此问题文中使用基于模拟退火的搜索策略。

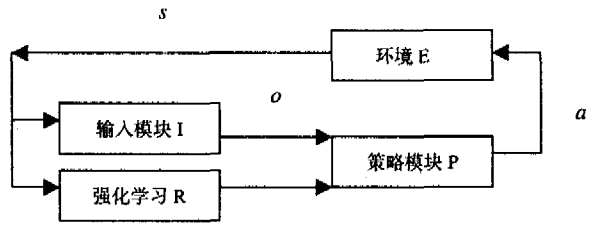

图 1　强化学习的一般结构

2　马尔可夫决策过程(MDP)模型

一般来说,强化学习算法采取马尔可夫决策过程[1]作为其数学模型。假定在时刻点 $t = t_1, t_2, \cdots, t_n, \cdots$ 处观察系统,一个有限的 Markov 决策过程由 5 元组组成:

$< S, A(s), P(s,a,s'), r(s,a), V \mid s \in S, a \in A(s) >$

其中各元素的含义如下:

(1) S 为系统所有可能的状态所组成的非空集,有时也称为系统的状态空间,可以是有限的、可列的或任意非空集。在文中,假定 S 为有限的,用小写字母 s 等来表示状态。

(2) 对 $s \in S$, $A(s)$ 是在状态 s 处可用的决策(行动)集,若不特别指出时,一般假定它是有限的,用小写字母 a 表示决策(行动)。

(3) 当系统在决策时刻点 t 处于状态 s,执行决策 a 后,则系统在下一个决策时刻点 $t+1$ 时处于状态 s' 的概率为 $P(s,a,s')$。

(4) 当系统在决策时刻点 t 处于状态 s,执行决策 a 后,系统于本阶段获得的即时报酬为 $r(s,a)$,通常称 $r(s,a)$ 为报酬函数。

(5) V 为准则(Criterion) 函数(或目标(Objective) 函数),常用准则函数有期望折扣总报酬、期望总报酬和平均报酬等并且可以是状态值函数或状态 - 行动对值函数。以状态 - 行动对为准则无限时段期望折扣总报酬为:

$$Q^{\pi}(s,a) = E_{\pi}\left\{\sum_{k=0}^{\infty}\gamma^{k} r_{t+k} \mid s_t = s, a_t = a\right\}$$

强化学习的最终目标是发现最优策略 π^*,一个从状态集到行动集的映射,以能达到最大折扣总报酬。最优策略可以通过鉴别最优值函数而获得,最优值的定义为:

$$Q^*(s,a) = \mathrm{Max}\left\{r(s,a) + \gamma\sum_{s'}P(s,a,s')\mathrm{Max}_{a'\in A(s')}Q^*(s',a')\right\}$$

3　改进的 Sarsa 算法

(1) 强化学习和神经网络的结合。

传统的强化学习算法如 Q 学习算法采用查表法表示[2,3],这种方法简单且计算效率高,但是当状态 - 行动对组成输入空间很大或输入变量是连续的,使用查表法需要大量的内存,因而开销很大。将强化学习和神经网络相结合,主要是利用神经网络的强大存储能力和函数估计能力。一般来说神经网络在系统中的工作方式[4]是:接收外界环境的完全或不完全状态描述,作为神经网络的输入,并通过神经网络进行计算,输出强化学习系统所需的 Q 值或 V 值。采用这种方式可以在较大程度上发挥这两种技术各自特有的优势。但是在强化学习中,用于训练BP神经网络的输入输出对是以增量的方式产生的[5],因而过一段时间以后,网络以前学习到的知识就会被"遗忘"。对此文中采用"化整为零"的思想,用神经网络队列保存 Q 值,其思路是:神经网络一般采用三层结构BP网络,输入层是状态 - 动作对,一个隐含层,输出是 Q 值以及版本号。版本号的引入是为了识别最新的 Q 值,Q 值首先保存在一张表 H 里,当表满之后,用神经网络保存表里最新的 Q 值和当前 Q 值的版本号(利用 H 训练网络),并且将神经网络以及相应的版本号插入队列。更新 Q 值时,从对头检索 Q 值,当第一次出现所需的 Q 值的版本号(通过神经网络计算 Q 值及其版本号) 和队列中该位置对应的版本号一致时即为所需要的 Q 值,另外随着学习的进行,版本号较老的神经网络可以删除掉。

(2) 基于 BP[6] 的 Sarsa 算法的描述。

// H 用于保存最近访问的 Q 值,当满的时候送至神经网络训练。

初始化表 H 令其为空;

初始化温度表 T 和神经网络队列 q;

初始化 BP,令状态 - 动作对 (s,a) 为神经网络的输入,$Q(s,a)$, $\mathrm{ver}(s,a)$ 分别为 Q 值和相应的版本号的输出,插入已经初始化的神经网络队列 q,

$\mathrm{ver}(s,a) = t, t = 1,2,3,\cdots$,初始版本号为 1,即 $t = 1$;

Repeat (Each episode)

{

Initialize s;

Repeat (Each step of episode)

{

从决策集中随即选择一动作 a^1;

$a = \arg\mathrm{Max}(Q(s,a)), a \in A(S)$;

生成一随机数 ε, $(\mid \varepsilon \mid \leqslant 1)$

if $\xi < \exp[Q(s,a') - Q(s,a))/T]$　　$a = a'$;

执行 a,观察即时奖励 r,下一状态 s' 及对应的策略动作 a';

if $Q(s,a)$, $Q(s',a')$ 不在 H 中,从神经网络队列中检索其值,否则直接从表中查询;并且更新 Q 值:

$Q(s,a) = (1-\eta)Q(s,a) + \eta(r(s,a) + Q(s',a'))$;

把 $((s,a),Q(s,a))$ 送入表 H;

If Full(H)

}

利用 H 表中的样本对网络进行训练,版本号为 $++t$;

If Full(q) 队尾元素出队;//删除含有较旧版本 Q 值的神经网络;把新训练的神经网络入队 q;

清空训练集;

}

$s = s'$;

}until s 是终止状态;

调节温度表 T 的取值按给定的序列变化,当随机动作的接受率小于一定阈值,退火停止,T 不变;

}until FALSE

Sarsa 算法的 Q 值更新是在策略的,即基于下一个实际状态的 Q 值,算法中的 episode 称为幕,即从初始状态到终止状态的过程。标准的 Sarsa 算法对状态空间的要求是离散的且空间数较小,而实际中很多的系统的状态空间是连续的或尽管是离散的但空间数较大,这就要求有很大的空间来存储状态动作对(State_ Action_ Pair),改进的算法用 BP 网络队列保存 SAPs。搜索策略借鉴了模拟退火思想,在算法中引入 Metroplis 准则,在模拟退火算法中随机量温度 T 是随时间逐步减少的,随着温度的下降,搜索新策略的概率越来越低,能够从一定程度上解决 ε - greedy 策略带来的问题。退火学习的终止时间是当随机动作的接受率小于一定阈值,退火停止,T 不变,随机动作的接受率是指随机动作执行的次数除以动作执行的总数。文中的主要贡献是引入神经网络簇保存规模较大的 Q 值表,从而减少保存 Q 值所需要的内存空间,并且克服了单个神经网络"增量式"学习所带来的"遗忘"问题。

4 实 验

用迷宫问题进行实验,随机生成 30×30 的迷宫,×标记(左上角)为初始状态,○标记(右下角)为目标状态,其中障碍物占 15%,如图 2 所示,白色表示开阔地,黑色表

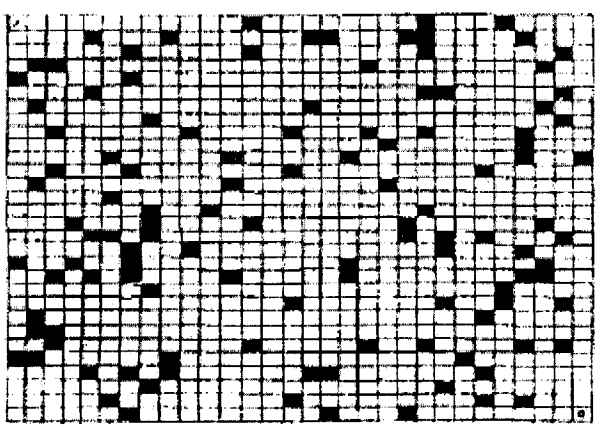

图 2 30×30 迷宫

示障碍物。学习的目的是寻找一条从起始状态到目标状态的最短路径。每一步智能 Agent 观察当前状态,选择动作执行,接收瞬时报酬。这里共有 4 种动作:向东、南、西和北 4 个方向移动,并且以 5/8 的概率朝所指定的方向移动,以 3/8 的概率随机向余下 3 个方向中的一个移动。到达目标时,$r = 200$;碰到障碍物或边沿时智能体停在原地,$r = -2$;其他情形,$r = -1$。若到达目标,则立即重新从起始状态开始进行新的试验。

实验中温度表 T 的初值为 100,采用 $\lambda = 0.6$ 等比降温策略;η 为 0.95;队列的长度为 17;表 H 的大小为 200;神经网络的结构是 2 - 8 - 2,隐含层激励函数采用 Sigmoid 函数,输入层和输出层采用线性函数;随机动作接受率为 0.05。图 3 为初始状态的 Q 值学习曲线。值得注意的是传统查找法对于此类问题需要保存约为 3600($30 \times 30 \times 4$)个 Q 值变量,而用神经网络队列只需要约 600 个变量(每个神经网络只需要约 30 个变量,H 表保存了 200 个 Q 值),显然从空间上考虑本算法效率较高。

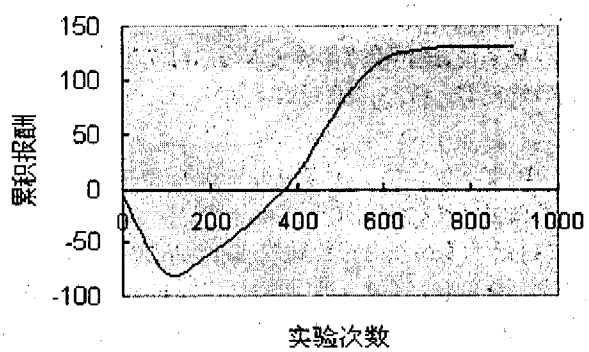

图 3 初始状态的 Q 值学习曲线

5 结 论

实验表明对于大规模的 MDP 问题,用神经网络队列来保存 Q 值,不仅节省了大量的内存空间,而且克服了单个神经网络"增量式"学习所带来的"遗忘"问题,从而解决了"维数灾难"带来的 Q 值表示所引起的内存开销问题。

参考文献:

[1] Astom K J. Optimal control of Markov decision processes with incomplete state estimation[J]. Math Anal Appl, 1998,10: 174 - 205.

[2] Tsitsiklis J N, Roy B V. An Analysis of Temporal_ Difference Learning with Function Approximation[J]. IEEE Transactions on Automatic Control,1997,42(5):674 - 690.

[3] Tesauro G J. TD—gammon, a self - teaching backgammon program[J]. Neural Computation, 1994, 6(2):215 - 219.

[4] Suton R S. Learning to predict by the methods of temporal diferences[J]. Machine Learning,1988(3):9 - 44.

[5] Suton R S,Barto A G. Reinforcement Learning: Introduction [M]. Cambridge,MA:MIT Press,1998.

[6] 黄德双. 神经网络模式识别系统理论[M]. 北京:电子工业出版社,1996.