

第5章 状态机

5.1 用特征向量来表示环境

一个S-R agent所用的特征向量可视为对与agent有关的环境状态的表示。这个S-R agent可从该特征向量中计算出一个适合环境状态的动作。如前所述，这个agent的传感器的局限使特征向量不可能完全精确地表示环境状态——尤其是那些从即刻(immediate)传感器输入中进行的特征向量。然而，通过考虑传感器的历史因素，可以提高其表示的精确度，因为agent以前也许曾感知过这些即刻无法感知的环境状态的某些重要方面。假设在某一时间 $t = 0$ 时，存在一个对环境的初始表示（如设计者可事先输入这个初始表示）。再假设，在任一个接下来的时间步骤 $(t+1)$ 时对环境状态的表示是 $(t+1)$ 时的传感器输入、前一时间 t 时对环境状态的表示和 t 时所采取的动作的函数。我们称用这种方法跟踪其环境状态的机器为状态机。除即刻传感器输入以外，状态机必须具备用来存储环境模型的存储器。

这个模型可采用多种形式。最简单的表示就是一个特征向量的表示——与S-R所用的一样。但此时，特征向量 X_{t+1} 要依另一个特征向量 X_t 而定。图5-1说明了这种状态机。

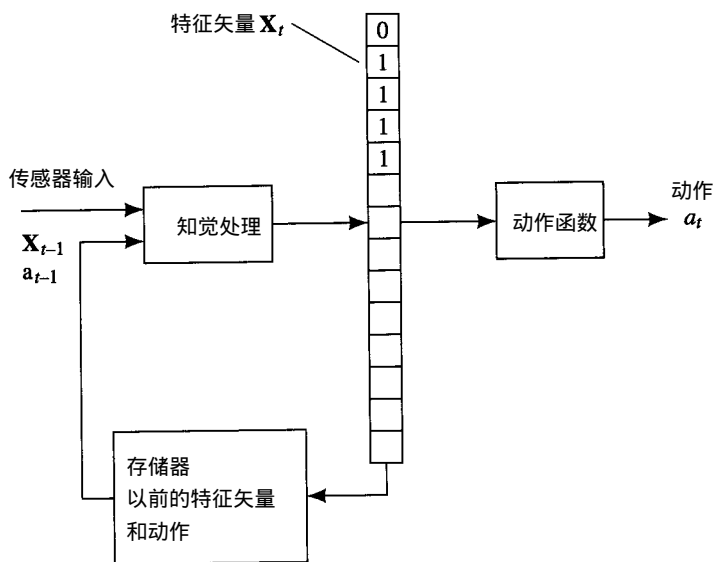


图5-1 一个状态机

由于agent的环境可能存在的任意复杂性，一个特征向量总是无法完整地表示其环境。然而，为特定任务而设计的agent能够归并许多环境状态。因此，agent设计者应该设置一个能适当表示其环境状态——至少与agent的任务有关的环境状态的特征向量。

我们把第2章中沿边界运动的机器人作为一例状态机。回想一下，机器人通过从8个传感器输入中计算的特征值来构造其环境模型，这些传感器输入把围绕agent的单元是否空缺的信息传递给agent。让我们来看看沿边界运动的机器人的传感器受损时的情况。假设此机器人只能

感知与其毗邻的北、东、南、西方向的四个单元，这四个传感器输入为 (s_2, s_4, s_6, s_8) ，即它无法感知与其斜对角毗邻的另外四个单元。当北、东、南、西方向的四个单元都不空缺时，这些传感器输入的值均为1；否则为0。若机器人是从它的即刻传感器输入、前一个特征向量和刚完成的动作中计算所需特征向量，那么，即使存在上述缺陷，机器人仍能完成沿边界运动这一行为（再次假设环境中不存在“稠密空间”）。

为继续用一个特征向量来计算动作的实例，我们采用 $w_i = s_i$ ，其中 $i = 2, 4, 6, 8$ ，在其他四个无法感知的传感器输入的位置上换上 w_1 、 w_3 、 w_5 和 w_7 。当且仅当前一时间步骤中 w_2 值为1且机器人东移， w_1 值为1。与此类似，当且仅当前一时间步骤中 w_4 值为1，且机器人北移， w_3 值为1，依此类推（否则， w_i 值为0）。

下列产生式系统运用这些特征向量给出了沿墙运动的行为（在不存在“稠密空间”的世界中）：

```

 $w_2 \bar{w}_4$    east
 $w_4 \bar{w}_6$    south
 $w_6 \bar{w}_8$    west
 $w_8 \bar{w}_2$    north
 $w_1$        north
 $w_3$        east
 $w_5$        south
 $w_7$        west
1         north

```

我们发现，第2章中的沿边界运动的机器人（其传感器未受损）不用存储前一次输入、特征和动作便能够完成任务。若机器人在任意所需时刻均能感知到环境的重要方面，那么就没有必要在存储器中存放一个环境模型。然而，传感器总有缺陷，故具有已存储的环境模型的 agent 通常能完成不具有存储器的机器人所不能完成的任务。

5.2 Elman网络

一个 agent 能用一种特殊的递归神经网络（被称为 Elman 网络 [Elman 1990]）来学习如何从前一个特征向量和传感器输入中计算后一个特征向量和动作。这里将说明只能感知与其毗邻的北、东、南、西方向四个单元的沿墙运动的机器人是如何完成上述过程的。这个网络学会了储存那些适合于当前任务的前一次所感知的特征。图 5-2 显示了这样的—个 Elman 网络。该网络有 8 个隐藏单元，它们与前一节所讨论的机器人所需的特征一一对应^①。网络的输入包括即刻传感器输入 (s_2, s_4, s_6, s_8) 以及前一时间步骤中 8 个隐藏单元的值。这 8 个附加输入称为“上下文单元(context unit)”，它们使网络将动作建立在已学会的前一次所感知的数据的特性之上——即机器人根据此上下文可找到自己即刻所处的位置。如前所述，有各种安排多网络输出的方法。这里，我们需要四个可区分的输出来与动作 north、east、south、west 一一对应。图 5-2 中的网络就有四个输出单元，分别代表四个动作。每个输出单元从同一组隐藏单元（特征）中计算动作。经过训练，我们可使一个单元所要求的输出为 1，而其他单元所要求的输出为 0。训

① 当然，在一个典型的学习情况中，我们并不知道需要多少隐藏单元（或多少个层面）。

练可通过带有相应合适动作标号的传感器输入序列来完成，这些动作类似于沿墙运动专家的动作。在实际操作中（训练后），我们选择具有最大输出的输出单元的动作。尽管 Elman 网络是第 3 章中提到的递归神经网络的一个特例，但仍可用普通反向传播来完成训练，这是因为反向权重（从隐藏单元到上下文单元）是固定的，而且上下文单元被看作是另一组输入。

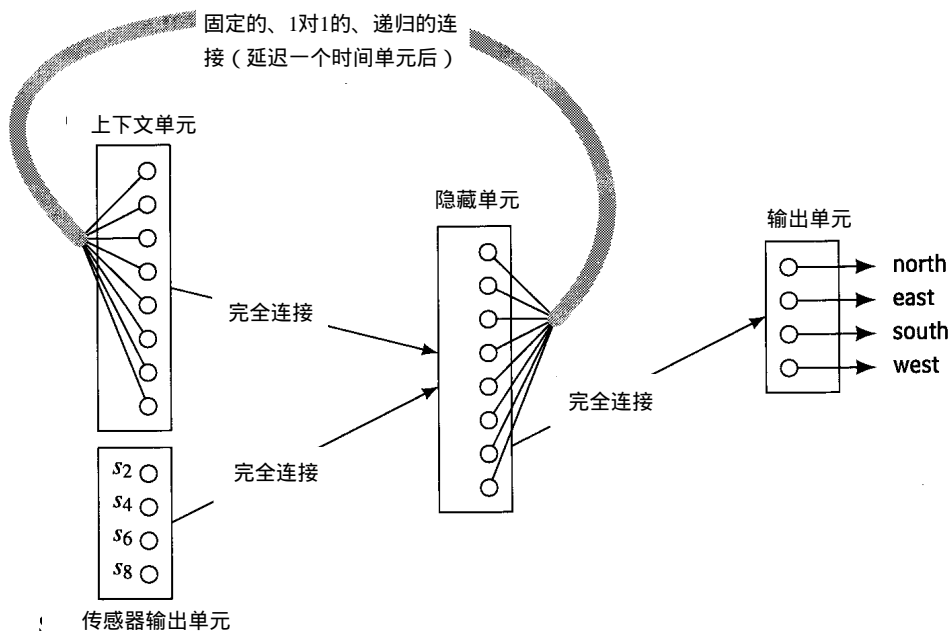


图5-2 一个Elman网络

5.3 图标表示

特征向量仅是表示环境的一种方法，我们还可以运用其他数据结构。比如，沿边界运动的机器人在感知时，可以用一个数组来存储一张由空缺单元和非空缺单元组成的映像。用特征或数据结构来表示世界的方法预示了人工智能的两大派别。其中的一派，我们称之为对世界“基于特征(feature-based)”的表示——即特征向量或属性向量；另一派我们称之为“图标(iconic)”表示——即数据结构，如地图，它被看作是对环境的重要方面的模拟（图标表示有时也称为“模拟表示(analogical representation)”）。虽然我们很难泾渭分明地区分这两种表示，但在本书中，它们的确存在不同之处。

当一个agent运用图标表示时，它仍需计算适合于其任务及其当时（已模型化的）环境状态的动作。它对数据结构的响应方式与没有存储器的agent对传感器刺激的响应方式几乎一样：计算数据结构的特征。图5-3说明了这样组织agent的一种方法。首先用传感器输入将图标模型更新为恰当的模型，然后用与感知处理相似的操作来提取动作计算子系统所需的特征。这些动作包括改变图标模型以及影响实际环境的动作。从图标模型中得到的特征的表示环境的方式应适合于机器人所必须采取的动作的种类。

当一个处于网格空间的机器人的图标表示是一个由空缺单元和非空缺单元组成的矩阵时，此图标表示就可以像机器人的传感器一样，能感知网格中的单元是否空缺。比如，我们来讨论

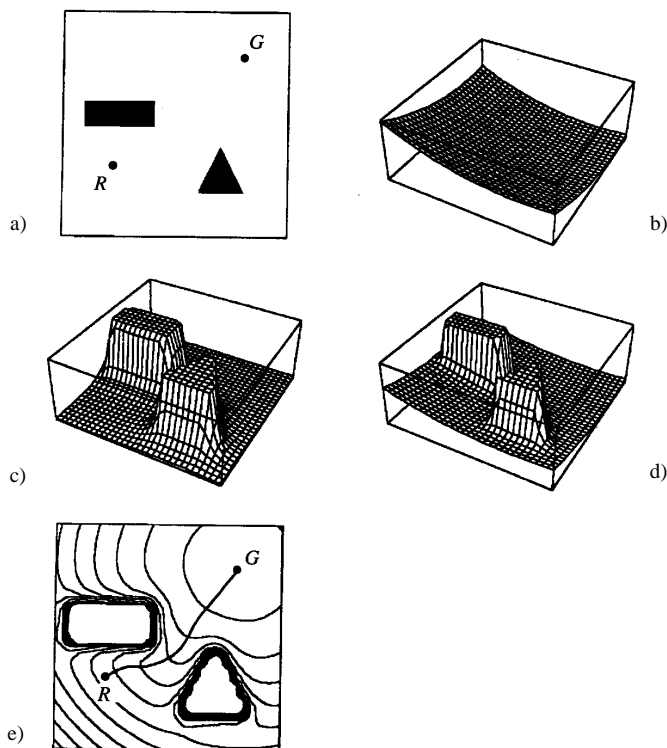


图5-5 一个人工势域的等位曲线 (节选自[Latombe 1991])

5.4 黑板系统

用来模型化世界的数据结构不必一定是图标的，尽管通常它们总是那样。一种重要的人工智能机器基于“黑板 (blackboard) 结构” [Hayes-Roth 1985, Nii 1986a, Nii 1986b]，这一结构运用一个被称为“黑板”的数据结构。一个被称为“知识源 (knowledge source, KS)”的程序读取并修改这一黑板。黑板系统详细地阐述了前面所描述的产生式系统。每个知识源有条件部分和动作部分，条件部分计算一个特征的值，这个特征可以是任意有关黑板数据结构的条件，其值为1或0（也可以是“真”或“假”）。动作部分是任一改变数据结构或执行外部动作（或两者一起）的程序。当两个或两个以上的知识源的值为1时，“冲突解决 (conflict resolution)”程序决策哪个知识源有效。知识源的动作不但能修改黑板，还能产生外部效果。另外，处理传感器数据的感知子系统也可以修改黑板。通常，我们通过分层来组织黑板数据结构，次级数据结构占据不同的层次。图 5-6即是这样的一个黑板系统。

知识源“精通”它们所管理的黑板的某一（或某几）部分的内容。当它们发现其管理的那一（或几）部分黑板有什么特别之处时，就会提出对黑板进行修改，而此修改如被选中，也可能激活其他知识源，依此类推。这样设计黑板，使得

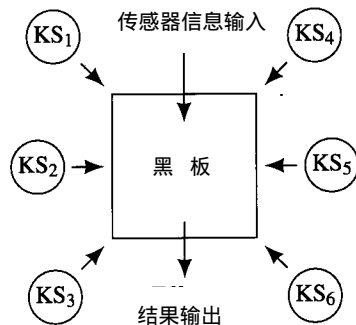


图5-6 一个黑板系统

当计算依次进行时，它最终变为这样一个数据结构：包含一个特殊问题的解或以所希望的方式改变世界的外部效果。这种黑板结构已运用于实际应用中，如语音理解 [Erman, et al. 1980]、信号解释 [Nii 1986b] 和医学病人护理监控 [Hayes-Roth, et al. 1992]。

举个例子，我们再来讨论处于网格世界的机器人——这次，它能即刻感知所有围绕它的8个单元。然而，它的传感器（如同真实的传感器）有时会给出错误信息。这个机器人有一幅关于它所处世界的不完整的地图——如图5-4所示。由于传感器的错误，这一地图可能不完整并且存在错误。这样，表达地图的数据结构和包含传感器数据的数据结构组成了一个黑板。图5-7 是此机器人经历中某一时刻的黑板。

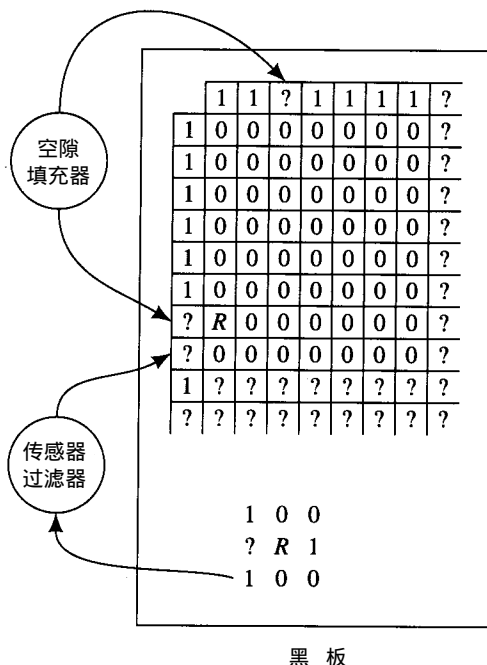


图5-7 一个机器人的黑板及其知识源

黑板有两个知识源来帮助机器人纠正错误。一个是“空隙填充器 (gap filler)”，另一个是“传感器过滤器 (sensory filter)”。前者在地图中寻找稠密空间（虽然已知不存在稠密空间），或者把1填入其中，或者扩展填充0（我想让你来推导填充和扩展的依据）。在图5-7中，空隙填充器决定在地图的顶部填充稠密空间。

传感器过滤器同时观察传感器数据和地图，并试图调和差异。图5-7中，传感器过滤器发现 s_7 的“已占据单元”的信号很强，而地图中相应的单元却是“？”。因此，它决定把地图中的“？”改为“1”来调和差异。相似地，根据地图的数据，它决定 $s_4 = 1$ 的信号是错误的，因此并不按此信号来改变地图。然后，空隙填充器把左边一列中的那个“？”填充上“1”。

根据为机器人所设定的任务，其他一些知识源（对已调整过的地图作出响应）可向机器人建议一个动作。

5.5 补充读物和讨论

状态机比S-R agent应用更广泛。第2章所提到S-R agent与动物行为学模型的关系也适用于状态机。许多动物对那些无法即刻感知的内部世界和外部世界的特征至少会保存部分信息（参阅习题5.4）。对存储器和感知器同时作响应的产生式系统已被用来构造某些心理现象的模型。

Elman网络是学习有限状态机的一例。[Rivest & Schapire 1993]也曾研究过这个问题。

许多研究者[Kuipers, et al. 1993, Kuipers & Byun 1991]研究过如何学习空间地图的问题，这些地图均是图标表示的例子。当对即刻状态和动作效果不确定时，地图学习会更加困难。[Dean, Basye, & Kaelbling 1993]曾研究过这种情况。

[Genesereth & Nilsson 1987]把那些其行为依据过去的历史的 agent命名为“hysteretic agents”。

习题

5.1 一个离散式电梯可以感知到它所处世界的以下信息：

- 1) 电梯应该停在哪一层。
- 2) 电梯里的乘客想要去哪几层。
- 3) 电梯外的哪几层有乘客想要乘电梯，他们想要往上乘还是往下乘。
- 4) 电梯门的状态（开或关）。

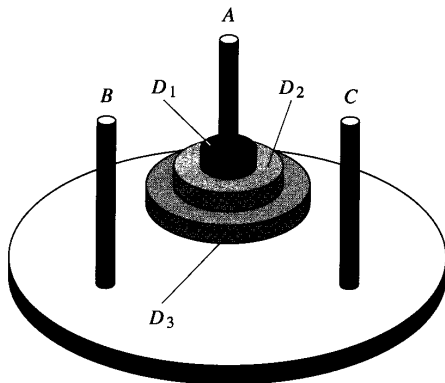
此电梯还能做以下动作：

- 1) 上升一层（除非已处顶层）。
- 2) 下降一层（除非已处底层）。
- 3) 开电梯门。
- 4) 关电梯门。
- 5) 等待 秒（是使得电梯里的乘客出电梯、电梯外的乘客进电梯的充足时间）。

设计一个能有效控制电梯的产生式系统（如果电梯里仍有人想要去更高层或者电梯外的更高层仍有人想要乘电梯，该系统却改“上升”为“下降”，则此系统就不能有效控制电梯）。

5.2 “人工蚂蚁”生活在一个二维网格世界中，它能沿已作标记的单元所组成的连续“信息素踪迹(pheromone trail)”（宽为一个单元）的运动。这个蚂蚁占一个单元，它可以面向上、下、左、右。它能做五个动作：前移一个单元(m)；在同一单元中向左转(l)；在同一单元中向右转(r)；设置状态位元“开”(on)；设置状态位元“关”(off)。蚂蚁感知它的正前方（即其面朝的方向）是否有信息素踪迹且其状态位元是否为“开”（设状态位元起动为“关”）。说明可以控制这样一个跟踪以上路线的蚂蚁的一个产生式系统。设这个蚂蚁最初位于一个可感知的信息素踪迹单元中（请记住，产生式系统由一个有序的条件—动作规则集合组成；所执行的动作与首先满足的那个条件相应。每个规则的动作部分可以有一个以上的动作）。确保蚂蚁不会折反自己走过的路线。

5.3 “三盘汉诺塔问题”有三个桩A、B和C，三个中心有孔的圆盘 D_3 、 D_2 和 D_1 可以放在它们的上面。盘 D_3 比盘 D_2 大，盘 D_2 又比盘 D_1 大。通常，此问题把三个桩排成一排，但这里把它们放在一个圆周上。首先把三个圆盘放在A上，然后试图将这三个圆盘移到别的桩上（如下图所示）。其中的规则为：可把任一个圆盘移到任一桩之上，但只能移动桩上位于顶部的圆盘，而且不能把大的圆盘放在小的圆盘之上。你也许熟悉解决这个问题的递归算法。



注释：有关更简单的算法，请参阅 www.mkp.com/nils/clarified。

有趣的是，还存在另一种非递归算法，它的规则陈述如下：我们总是移动可以移动的最大的圆盘，但每次移动的方向不能破坏前一次的移动。我们尽可能在奇数次时顺时针移动，在偶数次时逆时针移动。当不可能遵守这个顺逆时针的规则时（因为不存在其移动不破坏前一移动的最大圆盘），我们就往不理想的方向移动，然后再重新执行原来的规则。

我们将说明一个执行这一算法的产生式系统。假设下列六个圆盘移动的动作：
 $\text{move}(\text{num}, \text{dir})$ ， num 可以是 D_3 、 D_2 和 D_1 ， dir 可以是CW（顺时针）或CCW（逆时针）。再假设传感器可以判断以下条件的真伪： B_1 、 B_2 和 B_3 ，这里 B_i 是指圆盘 D_i 为最大的可移动的圆盘。另外，以下“状态”条件将由我们的产生式系统来设定和删除：CW（指上一次移动为顺时针）； M_1 、 M_2 和 M_3 ，这里 M_i 指最后一次移动的圆盘是 D_i ；改变状态的动作有toggle——它改变CW状态（即，当执行动作toggle以后，CW从1变为0或者从0变为1）； $\text{moved}(D_3)$ 、 $\text{moved}(D_2)$ 和 $\text{moved}(D_1)$ ，这里 $\text{moved}(D_i)$ 把 M_i 设置为真， M_j 为假（ j 不等于 i ）。

请用这些传感器和状态条件以及以上动作来说明一个运用此算法的产生式系统。别忘了指定状态条件的初始值！设这个产生式系统中的产生式规则已经排序，且所执行的动作与满足条件的最高规则相应。一个产生式规则的动作部分可以包含圆盘移动和状态改变的动作，且可同时执行这两种动作。

- 5.4 一个独处的母黄蜂Sphex把卵生在一只被它蜇晕并带回其巢穴的蟋蟀身上。母黄蜂孵出幼虫后，用此蟋蟀来给这些幼虫喂食。根据Wooldridge 1968, p. 70]，黄蜂将做以下有趣的动作：

母黄蜂把蜇晕的蟋蟀带回巢穴放在穴口，进巢穴看看是否一切正常，然后回来把蟋蟀拉进巢穴。当黄蜂进穴做预备检查时，若此蟋蟀被移开了几英寸，黄蜂回来时会把它再移回穴口，然后再重复进穴检查是否一切正常。若黄蜂在穴中作检查时，蟋蟀又被移开了几英寸，黄蜂出来后会再次把它移回穴口，再进穴作最后检查……有一回，这个过程重复了40次，每次结果都一样。

请设计能使黄蜂这样做的特征、动作和一个产生式系统。

- 5.5 设计一个人工势函数(有吸引部分和排斥部分)，使其能指导机器人在二维网格世界中(如下图)从任一单元运动到标记为0的目标单元中(设此机器人能做的动作为向北、东、南、西方向移动)。吸引和排斥部分的和有局部最小值吗？若有，它在哪儿？为此网格世界设计一个特殊的全局势函数，它不具局部最小值，而且通过使用它能保证以最短路径到达目标。

