

智能体

一、智能体和环境

任何通过传感器（sensor）感知环境（environment）并通过执行器（actuator）作用于该环境的事物都可以被视为智能体（agent）

感知（percept）：智能体的传感器正在感知的内容

智能体的感知序列（percept sequence）：智能体所感知的一切的完整历史

一般而言，一个智能体在任何给定时刻的动作选择可能取决于其内置知识和迄今为止观察到的整个感知序列，而不是它未感知到的任何事物。

从数学上讲，智能体的行为由智能体函数（agent function）描述，该函数将任意给定的感知序列映射到一个动作

人工智能体的智能体函数将由智能体程序（agent program）实现

智能体函数是一种抽象的数学描述，而智能体程序是一个具体的实现，可以在某些物理系统中运行

二、良好行为：理性的概念

性能度量

评估任何给定环境状态的序列

作为一般规则，更好的做法是根据一个人在环境中真正想要实现的目标，而不是根据一个人认为智能体应该如何表现来设计性能度量

理性

理性取决于以下4方面：

- 定义成功标准的性能度量
- 智能体对环境的先验知识
- 智能体可以执行的动作
- 智能体到目前为止的感知序列

理性智能体的定义

对于每个可能的感知序列，给定感知序列提供的证据和智能体所拥有的任何先验知识，理性智能体应该选择一个期望最大化其性能度量的动作。

理性 ≠ 全知；理性 ⇒ 探索, 学习

环境的本质

任务环境（task environment）：本质上是“问题”，理性智能体是“解决方案”

指定任务环境

PEAS（性能度量、环境、执行器、传感器）

- Performance（性能度量）
- Environment（环境）
- Actuator（执行器）
- Sensor（传感器）

任务环境的属性

1. 完全可观测的（fully observable）与部分可观测的（partially observable）

- 如果智能体的传感器能让它在每个时间点都能访问环境的完整状态，那么我们说任务环境是完全可观测的。
- 如果传感器检测到与动作选择相关的所有方面，那么任务环境就是有效的完全可观测的，而所谓的相关又取决于性能度量标准。
- 完全可观测的环境很容易处理，因为智能体不需要维护任何内部状态来追踪世界。
- 由于传感器噪声大且不准确，或者由于传感器数据中缺少部分状态，环境可能部分可观测。
- 如果智能体根本没有传感器，那么环境是不可观测的（unobservable）。

2. 单智能体的（single-agent）与多智能体的（multiagent）

3. 确定性的（deterministic）与非确定性的（nondeterministic）

如果环境的下一个状态完全由当前状态和智能体执行的动作决定，那么我们说环境是确定性的，否则是非确定性的。

- 回合式的（episodic）与序贯的（sequential）

在回合式任务环境中，智能体的经验被划分为原子式的回合。在每一回合中，智能体接收一个感知，然后执行单个动作。下一回合并不依赖于前几回合采取的动作

在序贯环境中，当前决策可能会影响未来所有决策

- 静态的（static）与动态的（dynamic）

如果环境在智能体思考时发生了变化，我们就说该智能体的环境是动态的，否则是静态的。

如果环境本身不会随着时间的推移而改变，但智能体的性能分数会改变，我们就说环境是半动态的（semidynamic）

- 离散的（discrete）与连续的（continuous）

离散/连续的区别适用于环境的状态、处理时间的方式以及智能体的感知和动作

- 已知的（known）与未知的（unknown）

智能体（或设计者）对环境“物理定律”的认知状态

在已知环境中，所有行动的结果（如果环境是非确定性的，则对应结果的概率）都是既定的

如果环境未知，智能体将不得不了解它是如何工作的，才能做出正确的决策

三、智能体的结构

智能体程序：实现智能体函数，即从感知到动作的映射
智能体架构：具有物理传感器和执行器的计算设备
智能体 = 架构 + 程序

智能体程序

按照通用性递增的顺序，4种基本的智能体程序：

- 简单反射型智能体
- 基于模型的反射型智能体
- 基于目标的智能体
- 基于效用的智能体

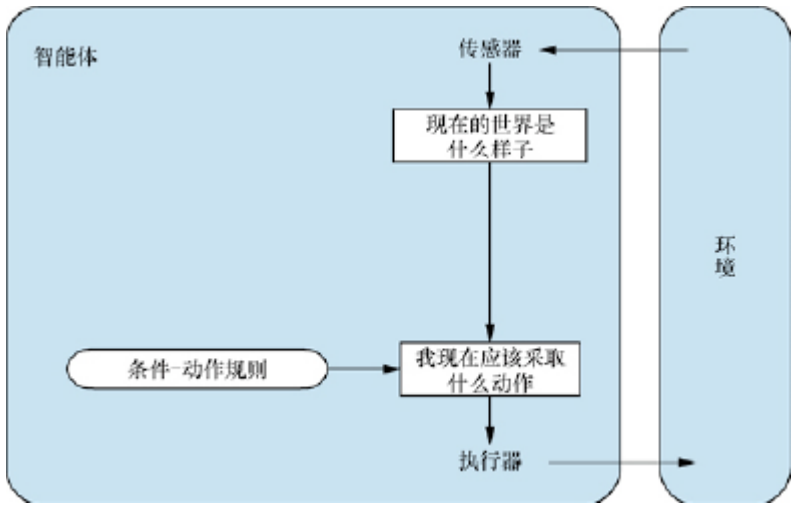
所有这些类型的智能体都可以转化为学习型智能体

简单反射型智能体

根据当前感知选择动作，忽略感知历史的其余部分

简单反射型智能体程序。它根据一条规则进行操作，该规则的条件与感知定义的当前状态相匹配

条件-动作规则（**condition-action rule**），也称为情境-动作规则、产生式系统或 if-then 规则



基于模型的反射型智能体

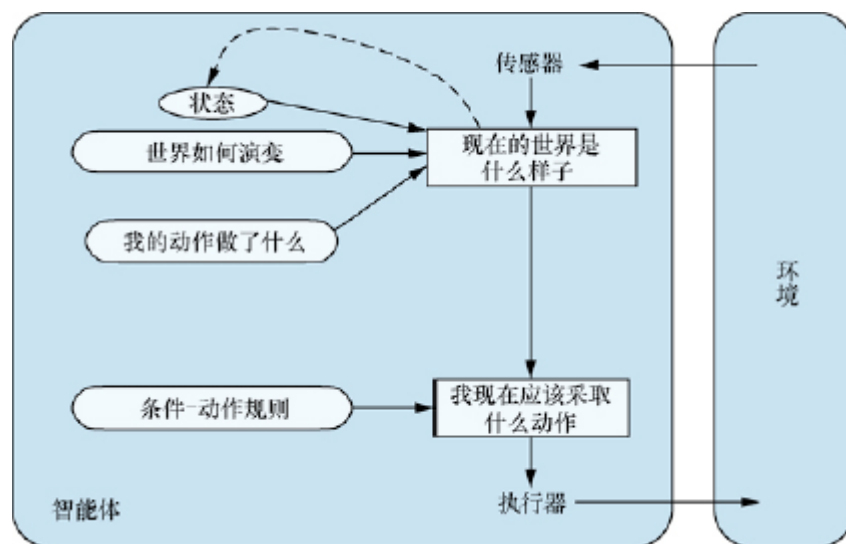
智能体应该维护某种依赖于感知历史的**内部状态**（**internal state**），从而至少反映当前状态的一些未观测到的方面

转移模型和传感器模型结合在一起让智能体能够在传感器受限的情况下尽可能地跟踪世界的状态。使用此类模型的智能体称为**基于模型**的智能体

随着时间的推移，更新这些内部状态信息需要在智能体程序中以某种形式编码两种知识。

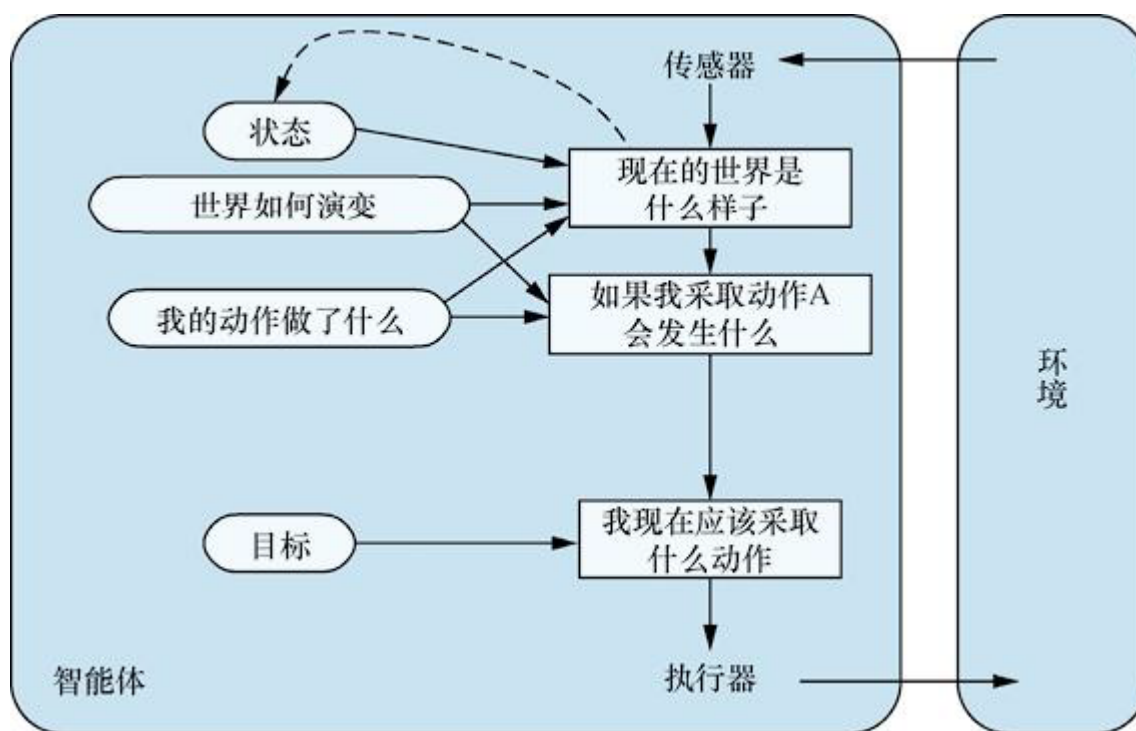
首先，需要一些关于世界如何随时间变化的信息，这些信息大致可以分为两部分：智能体行为的影响和世界如何独立于智能体而发展。这种关于“世界如何运转”的知识（无论是在简单的布尔电路中还是在完整的科学理论中实现）被称为世界的**转移模型**（**transition model**）。

其次，我们需要一些关于世界状态如何反映在智能体感知中的信息。例如，当前面的汽车开始刹车时，前向摄像头的图像中会出现一个或多个亮起的红色区域；当摄像头被淋湿时，图像中会出现水滴状物体并部分遮挡道路。这种知识称为**传感器模型**（**sensor model**）。



基于目标的智能体

基于模型、基于目标的智能体。它追踪世界状态以及它试图实现的一系列目标，并选择一项最终能够实现目标的动作

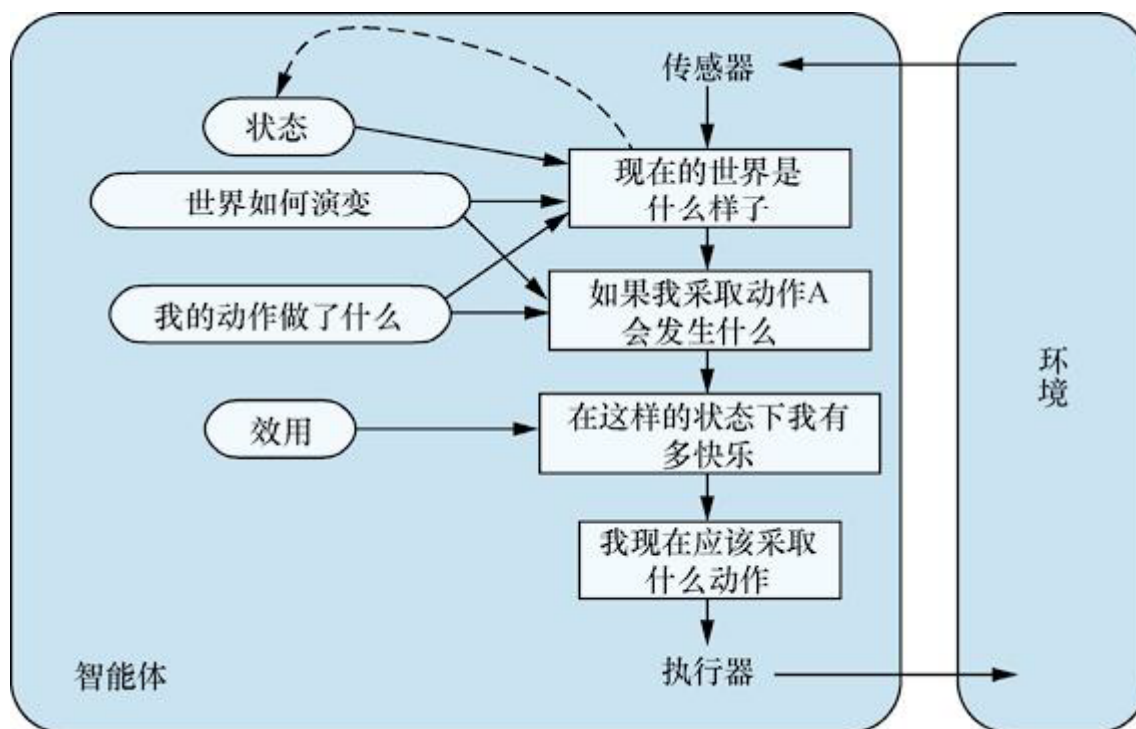


基于效用的智能体

基于模型、基于效用的智能体。它使用了一个世界模型以及一个效用函数来衡量它在各状态之间的偏好，然后选择产生最佳期望效用的动作，其中期望效用是通过对所有可能的结果状态和对应概率加权所得

智能体的效用函数（utility function）本质上是性能度量的内部化

基于效用的理性智能体会选择能够最大化其动作结果期望效用（expected utility）的动作，也就是在给定每个结果的概率和效用的情况下，智能体期望得到的平均效用



学习型智能体

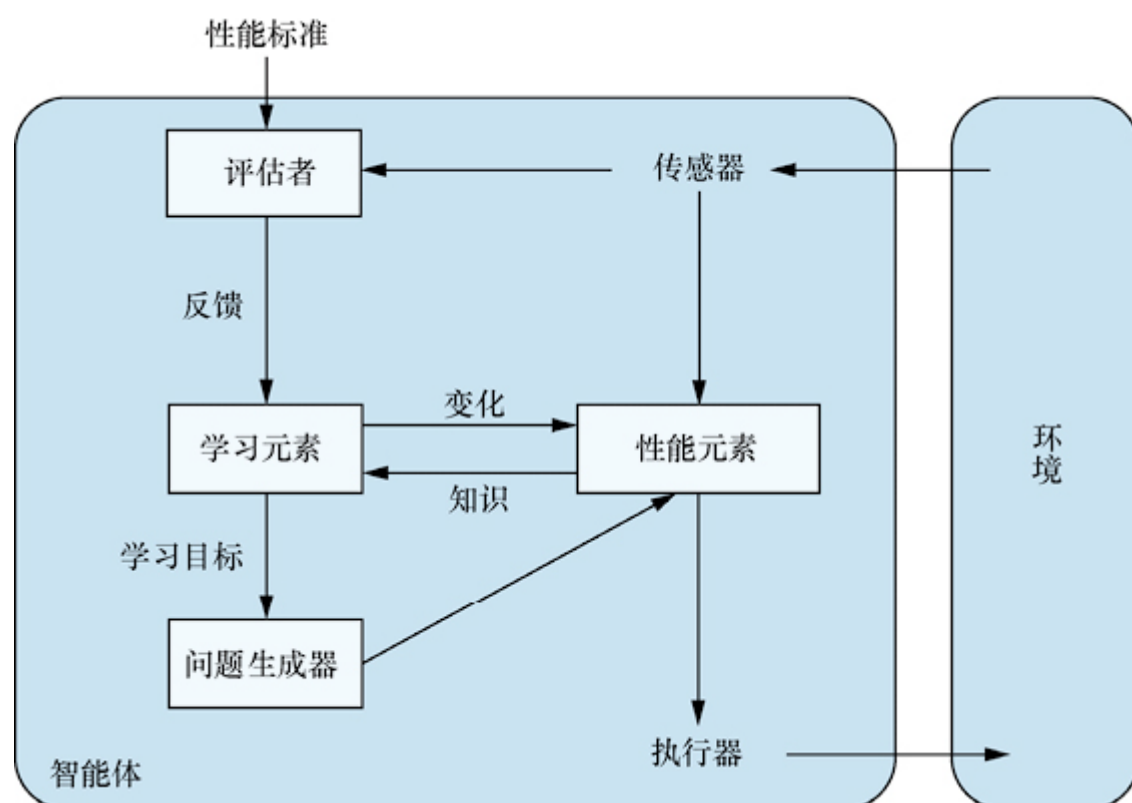
学习型智能体可分为4个概念组件

负责提升的学习元素 (learning element)

负责选择外部行动的性能元素 (performance element)

学习元素使用来自评估者 (critic) 对智能体表现的反馈，并以此确定应该如何修改性能元素以在未来做得更好

问题生成器 (problem generator)。它负责建议动作，这些动作将获得全新和信息丰富的经验



智能体程序的组件如何工作

表示状态及其之间转移的3种方法：

- 原子表示一个状态（如B或C）是没有内部结构的黑盒
- 因子化表示状态由属性值向量组成，值可以是布尔值、实值或一组固定符号中的一个
- 结构化表示状态包括对象，每个对象可能有自己的属性以及与其他对象的关系

