

《机器学习》课件



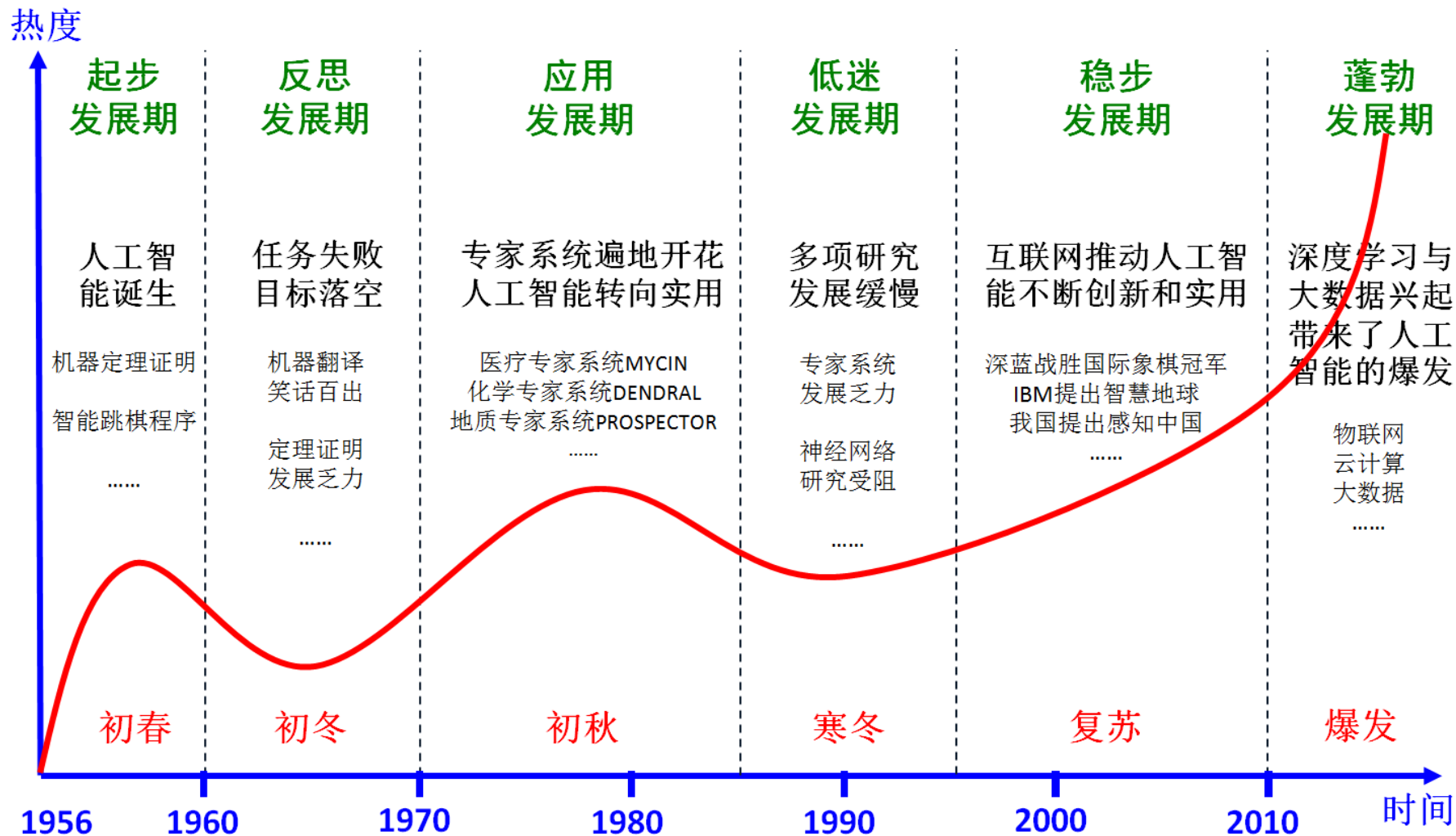
## 2.2 增强学习



北京航空航天大学  
BEIHANG UNIVERSITY

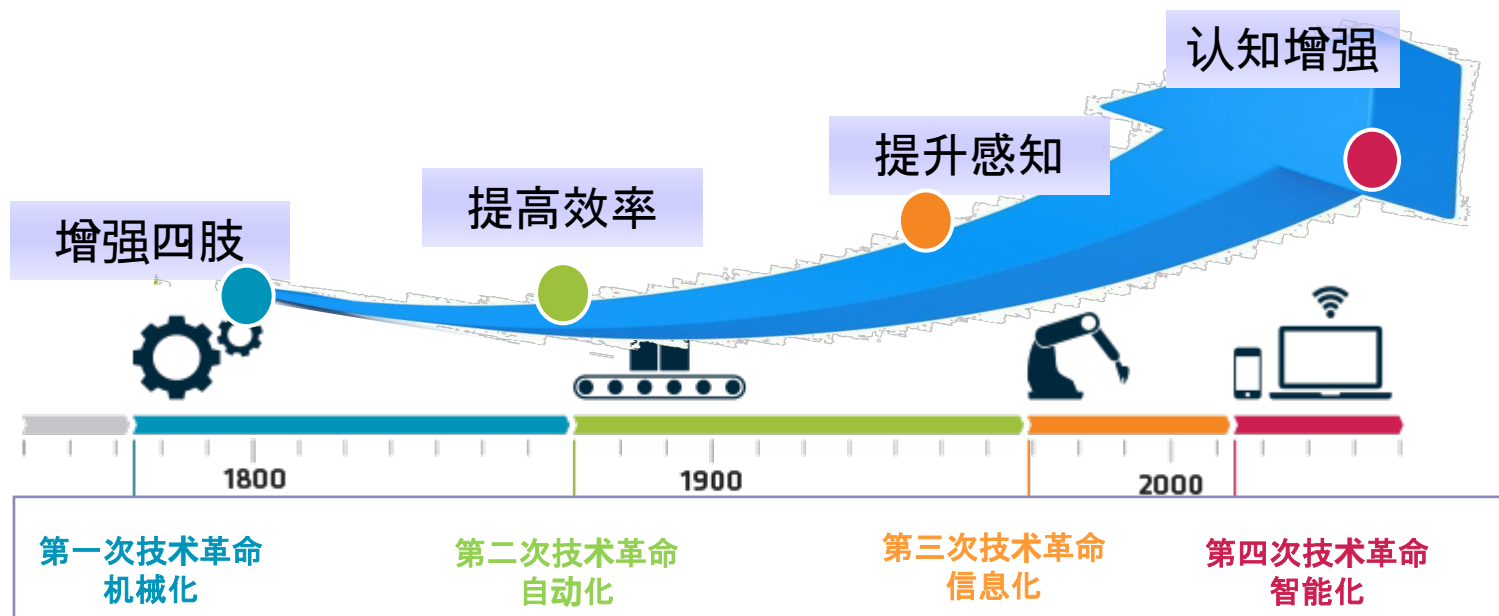
# 背景

## ■ 人工智能发展历程



# 背景

## ■ 人工智能——第四次技术革命的基石



- 人工智能将在提升机器认知能力基础上，全面整合人类本身智能以及人类过去文明成果，从而提升乃至解放人类认知。

# 背景

- 人工智能发展最大特点：将渗入到人类社会的各个领域，重塑产业格局以及社会结构



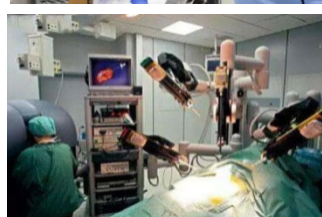
国防



工业



农业

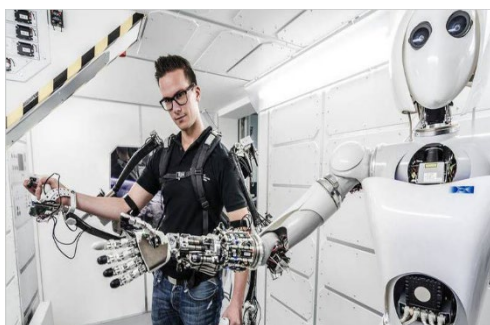


医疗

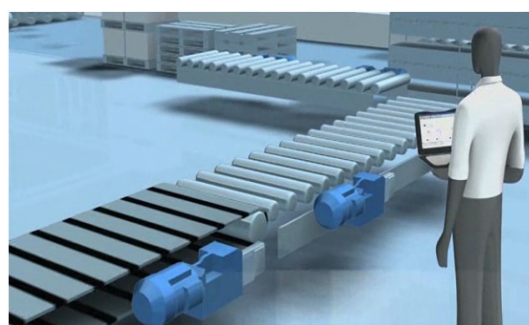


教育

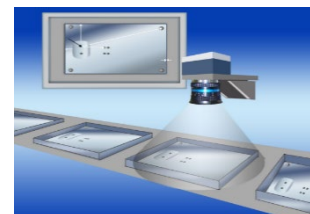
人机协作



人机决策



2017年5月，谷歌公司的人工智能软件  
“Alpha Go”战胜目前围棋世界排名第一  
的柯洁



**人工智能领域的里程碑**

# AlphaGo技术的三大核心内容

## ◆ 增强学习：Q学习

◆ 重点、难点、掌握机器人、无人机路径规划中的应用

## ◆ 深度学习

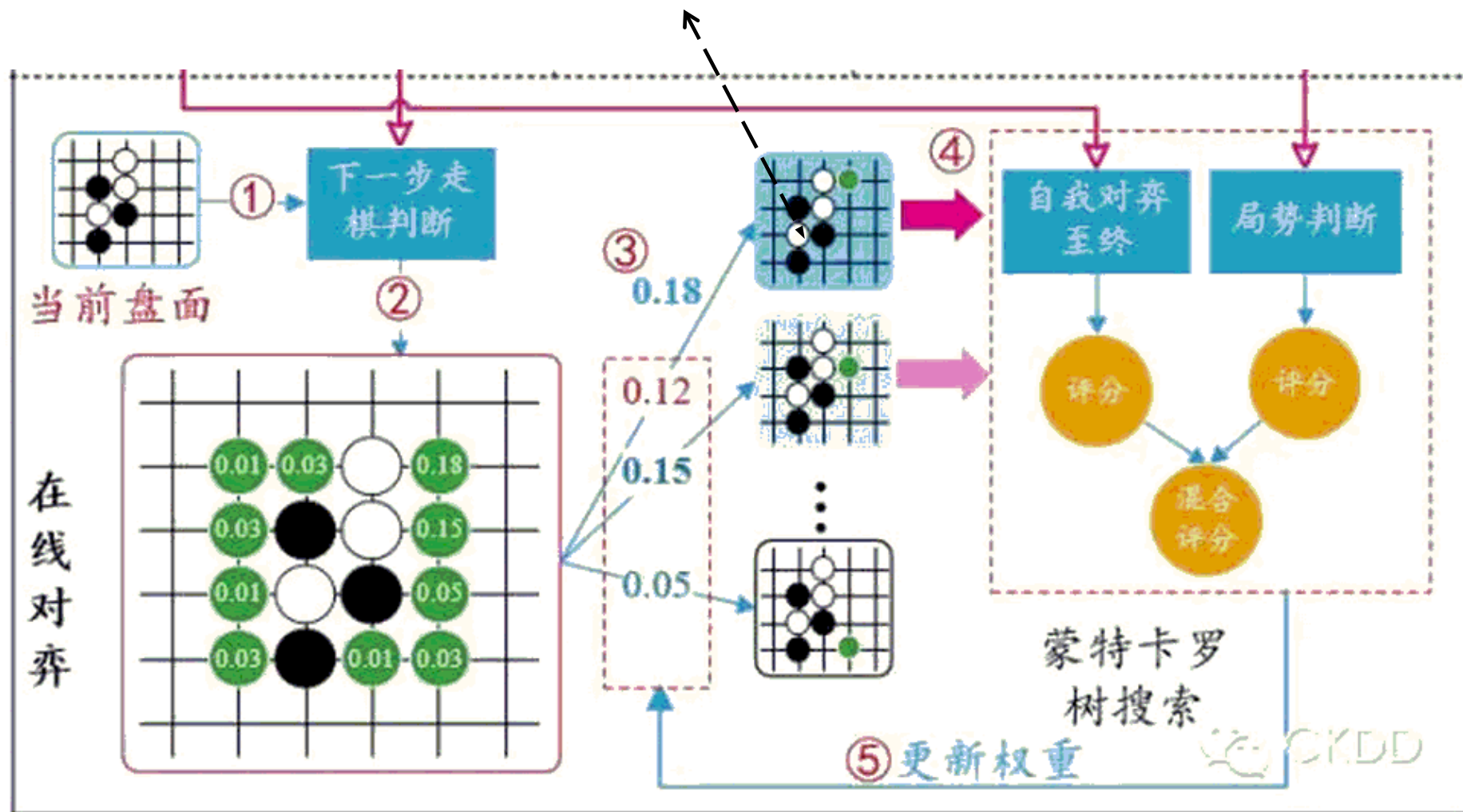
◆ 难点、后续课程介绍

## ◆ 蒙特卡洛搜索树

◆ 了解如何结合Q学习实现AlphaGo，上节课内容



# 增强学习：学习如何下棋



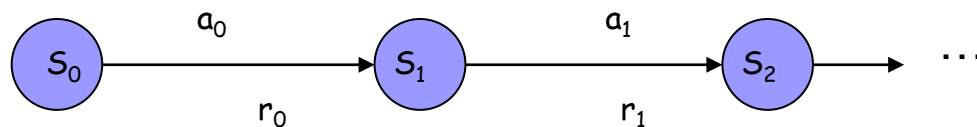
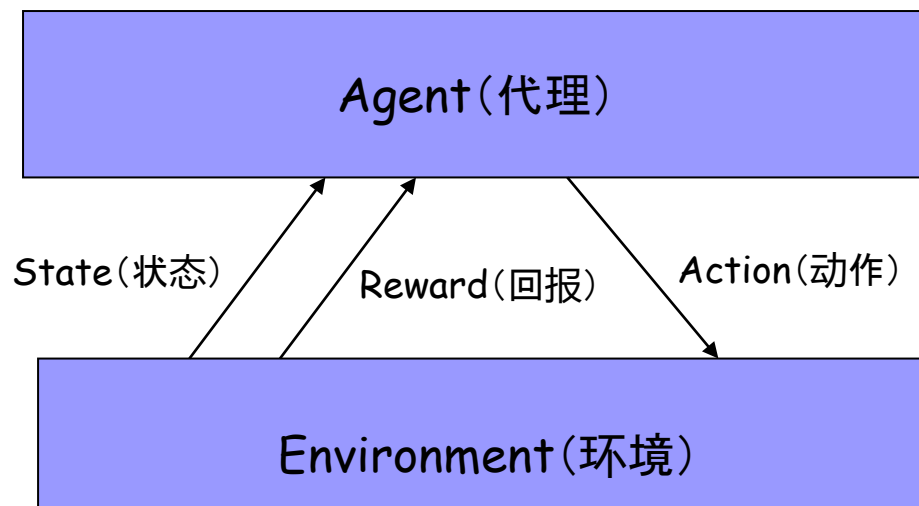
但，则仕家将下歹例上进一步展开下一级别的技系（如图2所示）。



# Q-学习通用算法 不限于围棋



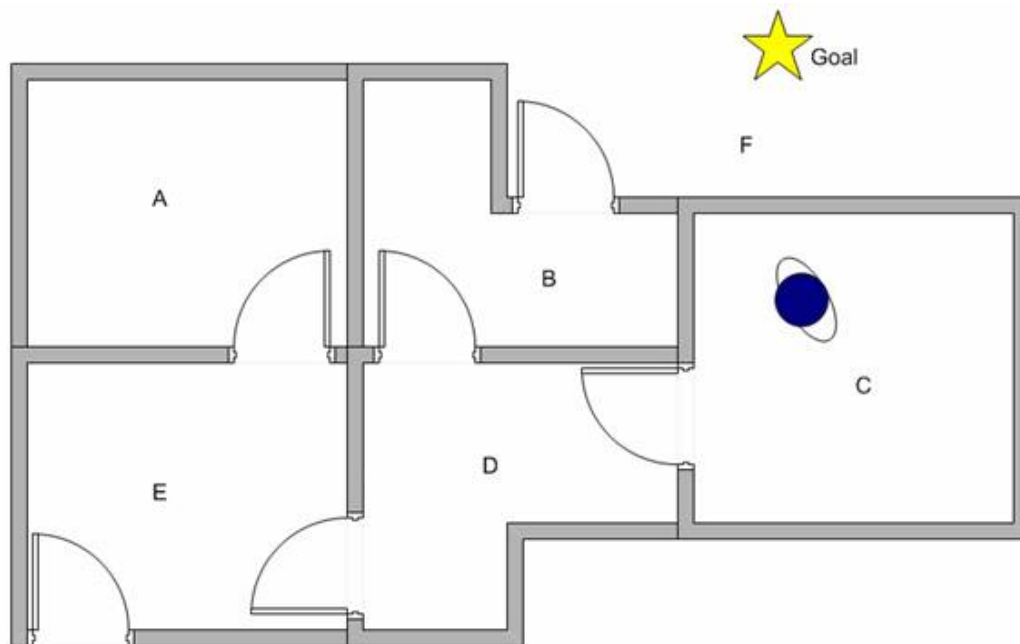
# Q学习-增强学习算法



以回报为核心的通用算法: **Agent**在所处环境总是基于回报最大选择合适的动作, 从而完成与环境的交互, 完成特定任务。

# Q学习算法

## ➤ 机器人路径规划

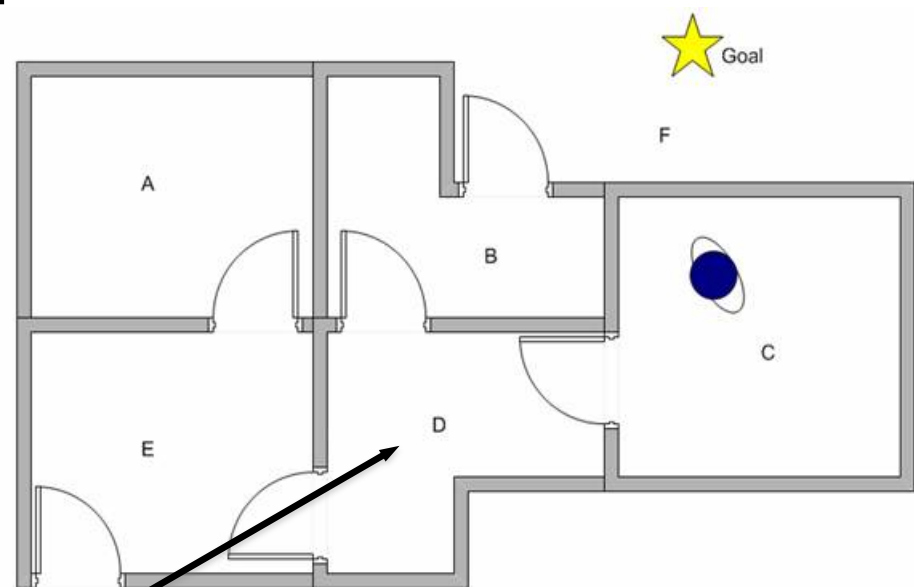
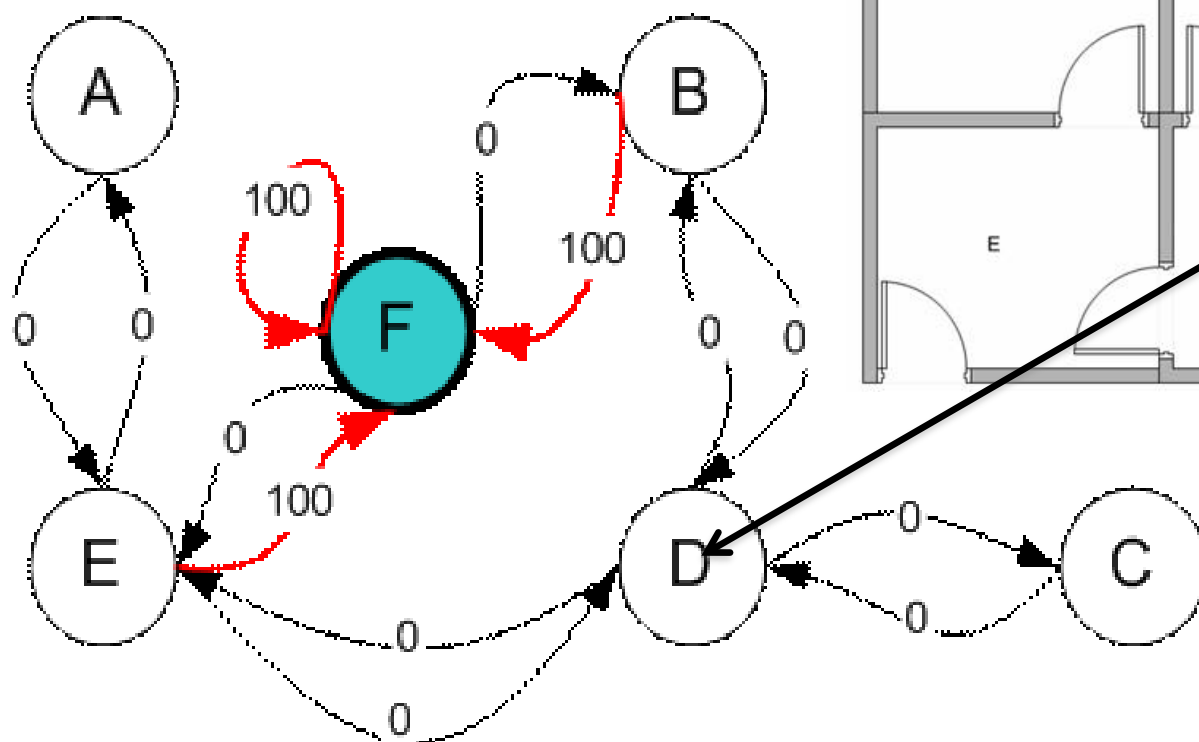


A to E: rooms, F: outside building (target).

任务:处于任何一个房间的机器人都能够找到一条最优路径走到室外

## 环境建模图表示

- 图表示，节点表示房间号，边表示动作，它的值表示回报，例如E
- 基于回报进行建模



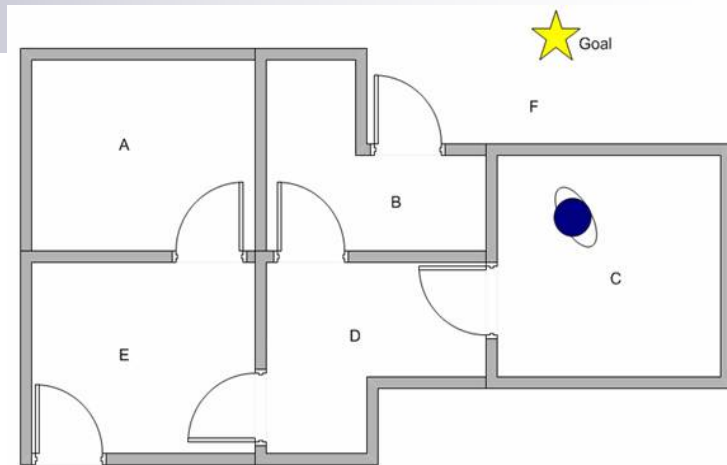
问题: 如何存储?

## 环境建模矩阵表示

### 回报矩阵

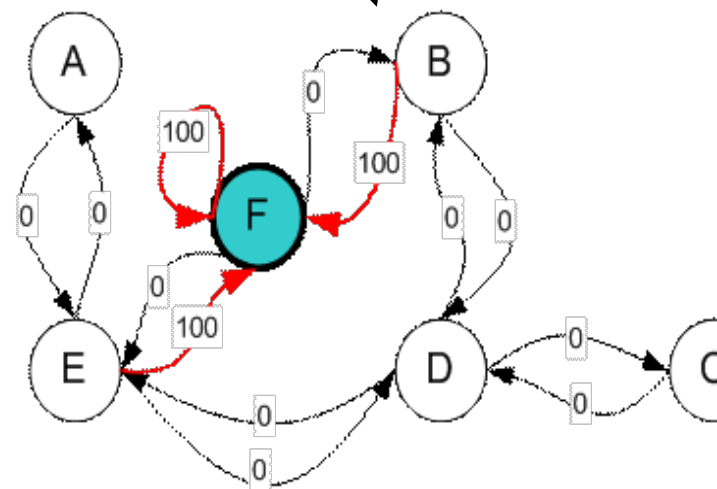
### 图表示等价形式

## 建立状态和行为对应关系



$\mathbf{R} =$

state \ action	A	B	C	D	E	F
A	-	-	-	-	0	-
B	-	-	-	0	-	100
C	-	-	-	0	-	-
D	-	0	0	-	0	-
E	0	-	-	0	-	100
F	-	0	-	-	0	100



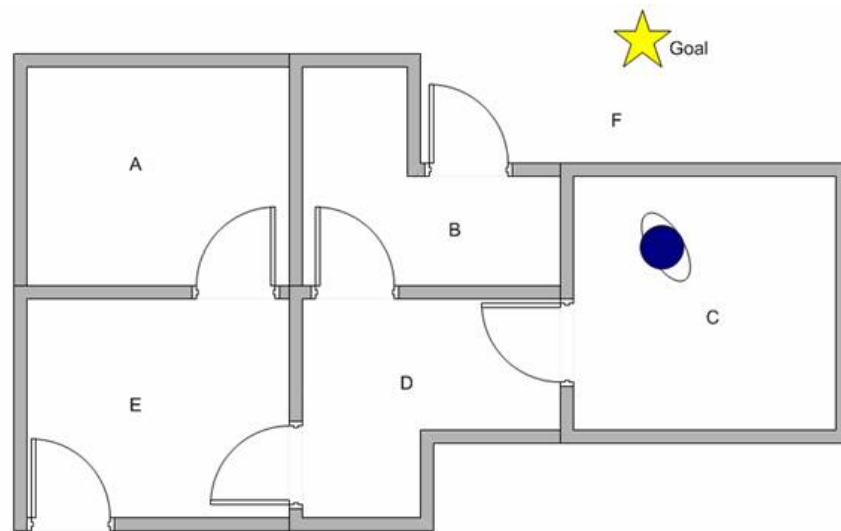
思考: 由于元素的值多数为0, 无法求解, 如何计算更加合理的回报?

# Q学习算法核心概念

Q矩阵进行回报学习

更加合理的回报: 状态/动作

$$Q = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$R = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & - & - & - & 0 & - \\ - & - & - & 0 & - & 100 \\ - & - & - & 0 & - & - \\ - & 0 & 0 & - & 0 & - \\ 0 & - & - & 0 & - & 100 \\ - & 0 & - & - & 0 & 100 \end{bmatrix} \end{matrix}$$


$$Q(state, action) = R(state, action) + \gamma \cdot \text{Max}[Q(next\ state, all\ actions)]$$

$$0 \leq \gamma < 1 \quad \text{学习率}$$

Q矩阵的计算是一个迭代过程, 决定了算法设计;

思考问题: 为什么是增强学习?

# Q学习算法

- 输入：状态/动作-回报矩阵( $\mathbf{R}$ )，目标状态Goal；
- 输出：从任何初始状态到目标状态的最小路径( $\mathbf{Q}$ )；

1. 设置学习率 $\gamma$ ，环境汇报矩阵 $\mathbf{R}$

2.  $\mathbf{Q} = 0$ ；

3. 进行迭代循环：

- 随机选择初始状态
- 如果不是目标状态：
- 选择所有可能的动作，进行测试：
  - 选择Q值最大的状态进行Q值更新，计算公式如下所示：

$$\mathbf{Q}(\text{state}, \text{action}) = \mathbf{R}(\text{state}, \text{action}) + \gamma \cdot \text{Max}[\mathbf{Q}(\text{next state}, \text{all actions})]$$

如果Q值稳定了，结束循环，否则继续迭代。



# Q学习算法

初始状态选择为B

$$\mathbf{Q} = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

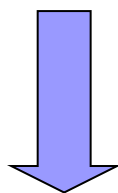
$$\mathbf{R} = \begin{matrix} & \begin{matrix} A & B & C & D & E & F \end{matrix} \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & - & - & - & 0 & - \\ - & - & - & 0 & - & 100 \\ - & - & - & 0 & - & - \\ - & 0 & 0 & - & 0 & - \\ 0 & - & - & 0 & - & 100 \\ - & 0 & - & - & 0 & 100 \end{bmatrix} \end{matrix}$$

# 1次迭代

在B状态下，我们随机选择一个动作，根据收益最大，我们选择为F，则有：

$$Q(state, action) = R(state, action) + \gamma \cdot \text{Max}[Q(next\ state, all\ actions)]$$

$$Q(B, F) = R(B, F) + 0.8 \cdot \text{Max}\{Q(F, B), Q(F, E), Q(F, F)\} = 100 + 0.8 \cdot 0 = 100$$



$$Q = \begin{matrix} & A & B & C & D & E & F \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

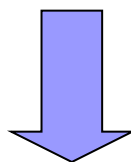
$$R = \begin{matrix} & state \backslash action & A & B & C & D & E & F \\ \begin{matrix} A \\ B \\ C \\ D \\ E \\ F \end{matrix} & \begin{bmatrix} - & - & - & - & 0 & - \\ - & - & - & 0 & - & 100 \\ - & - & - & 0 & - & - \\ - & 0 & 0 & - & 0 & - \\ 0 & - & - & 0 & - & 100 \\ - & 0 & - & - & 0 & 100 \end{bmatrix} \end{matrix}$$

# 2次迭代

随机选择一个状态D，根据回报最大，选择的动作为B，则可以更新Q如下：

$$Q(state, action) = R(state, action) + \gamma \cdot \text{Max}[Q(next\ state, all\ actions)]$$

$$Q(D, B) = R(D, B) + 0.8 \cdot \text{Max}\{Q(B, D), Q(B, F)\} = 0 + 0.8 \cdot \text{Max}\{0, 100\} = 80$$



	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>A</i>	0	0	0	0	0	0
<i>B</i>	0	0	0	0	0	100
<i>C</i>	0	0	0	0	0	0
<i>D</i>	0	80	0	0	0	0
<i>E</i>	0	0	0	0	0	0
<i>F</i>	0	0	0	0	0	0

**R =**

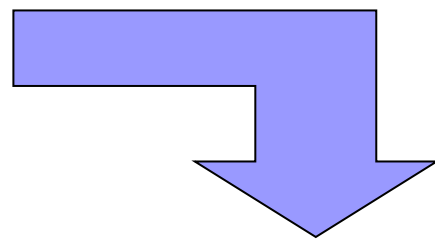
<i>state \ action</i>	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>
<i>A</i>	-	-	-	-	0	-
<i>B</i>	-	-	-	0	-	100
<i>C</i>	-	-	-	0	-	-
<i>D</i>	-	0	0	-	0	-
<i>E</i>	0	-	-	0	-	100
<i>F</i>	-	0	-	-	0	100

# 多次迭代之后

Q 矩阵达到稳定状态

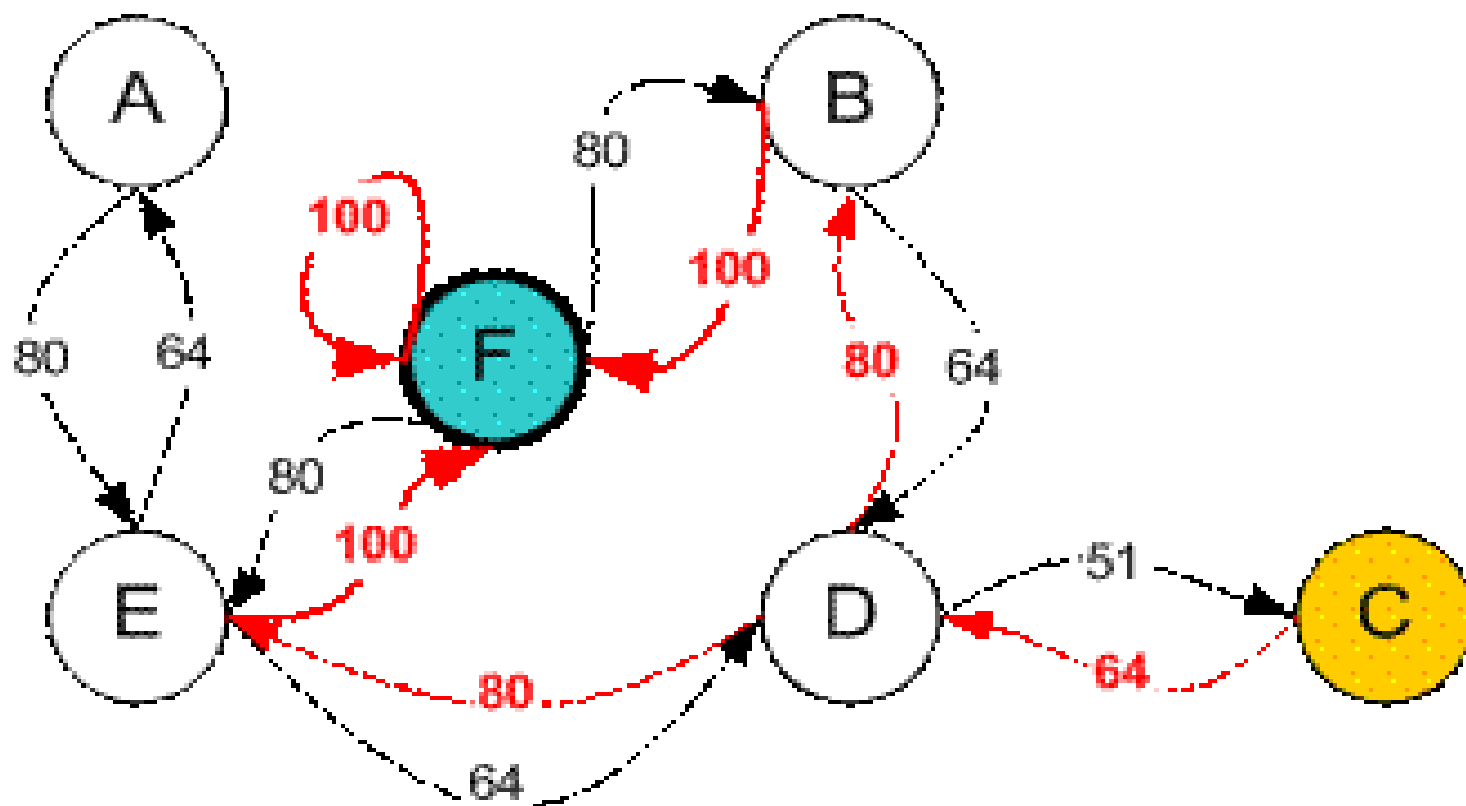
$$Q = \begin{array}{c|cccccc} \text{state} \backslash \text{action} & A & B & C & D & E & F \\ \hline A & - & - & - & - & 400 & - \\ B & - & - & - & 320 & - & 500 \\ C & - & - & - & 320 & - & - \\ D & - & 400 & 256 & - & 400 & - \\ E & 320 & - & - & 320 & - & 500 \\ F & - & 400 & - & - & 400 & 500 \end{array}$$

规范化



$$\hat{Q} = \begin{array}{c|cccccc} \text{state} \backslash \text{action} & A & B & C & D & E & F \\ \hline A & - & - & - & - & 80 & - \\ B & - & - & - & 64 & - & 100 \\ C & - & - & - & 64 & - & - \\ D & - & 80 & 51 & - & 80 & - \\ E & 64 & - & - & 64 & - & 100 \\ F & - & 80 & - & - & 80 & 100 \end{array}$$

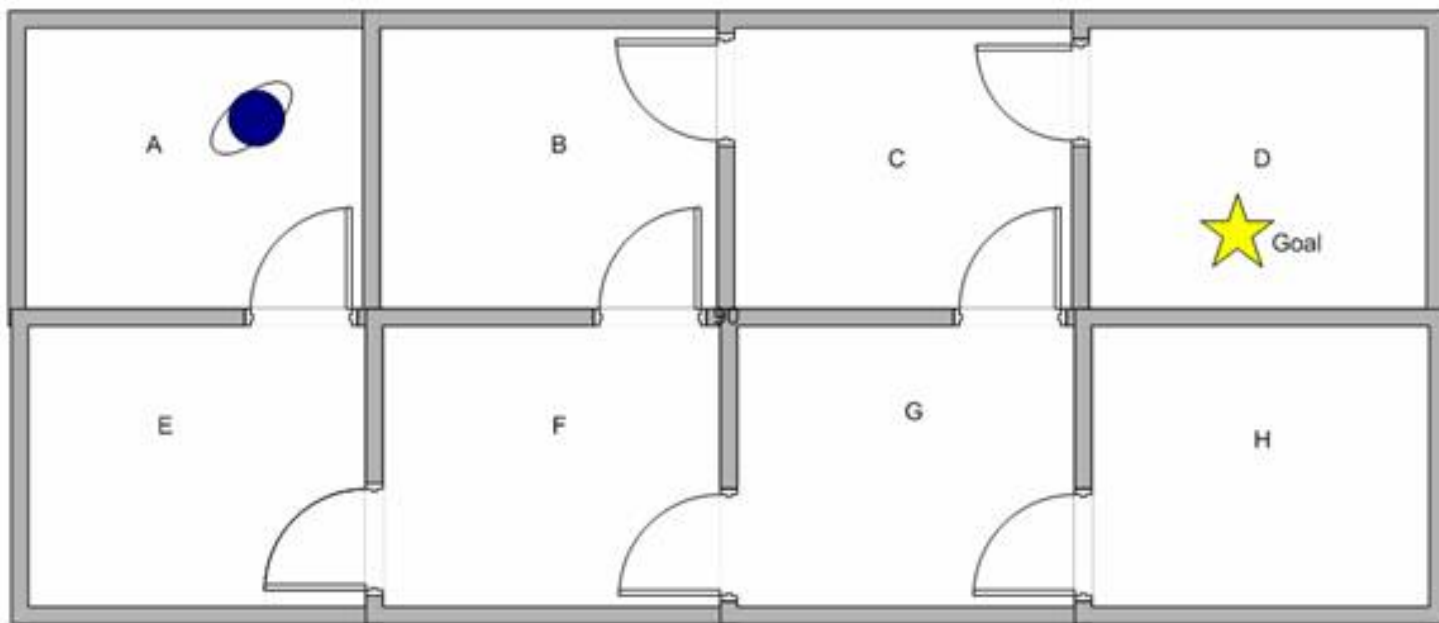
基于Q矩阵, 可以从任何一个状态开始,  
寻优找到路径:



请同学回答:从C开始, 机器人如何走出去房间

# 课堂练习

- 给出R矩阵-输入
- 计算Q矩阵-输出



无人机导航