DL: Введение в RL

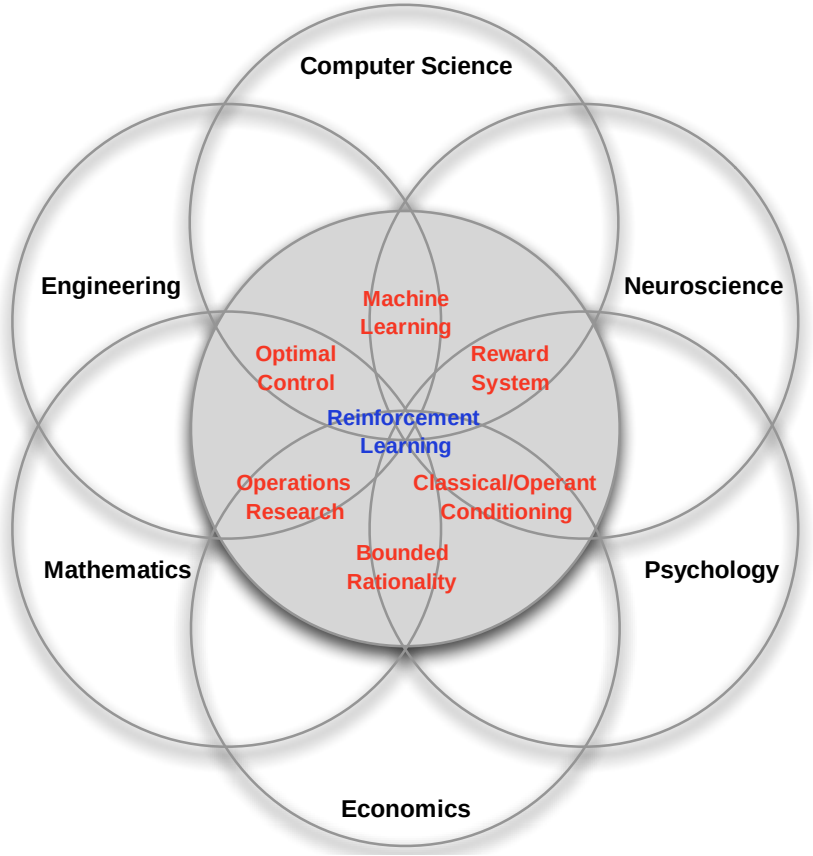
План

- ❑ Что такое RL
- ❑ Постановка задачи RL
- ❑ Схема RL-агента
- ❑ Проблемы в рамках обучения с подкреплением

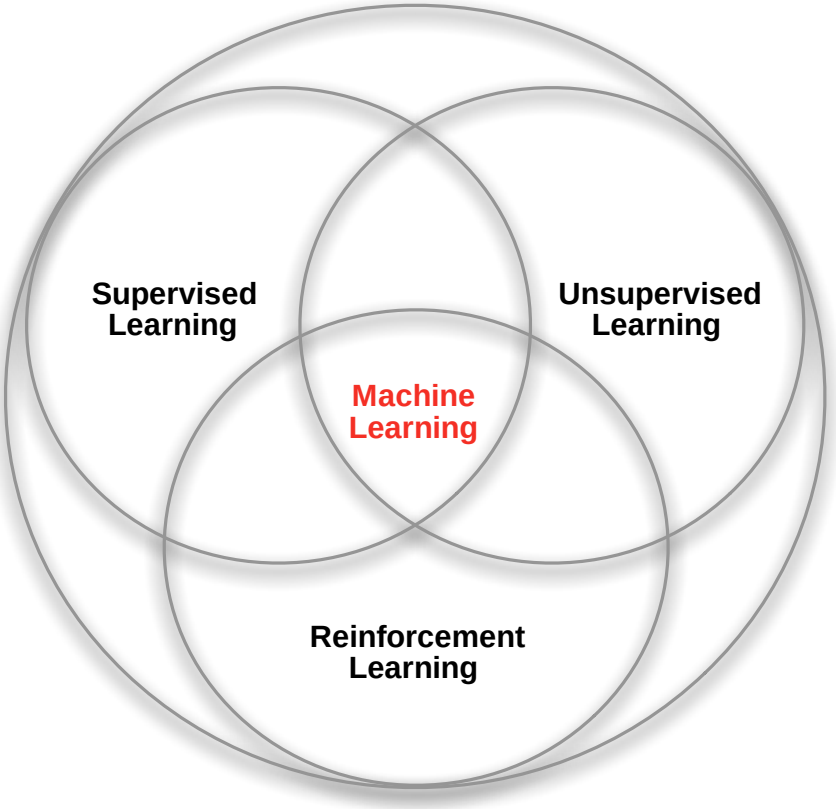
Обучение с подкреплением

Обучение интеллектуального агента хорошей последовательности принятия решения в условиях неполной информации

Обучение с подкреплением и другие науки



Разделы машинного обучения



- Отделение среды (environment) и агента (agent)
- Нет учителя, т.е. ошибка не задается явно, а косвенно передается через вознаграждение (reward)
- Обратная связь (feedback) от среды может поступать с задержкой (delayed)
- Параметр времени имеет особое значение – последовательные (sequential) данные
- Действие агента влияют на поступающие в дальнейшем данные

Примеры

- Выполнение сложных полетных фигур на вертолете
- Го, шахматы и пр.
- Управление инвестиционным портфелем
- Управление электростанцией
- Игры Atari
- Управление человекоподобным роботом

Вознаграждение

- Вознаграждение r_t - это скалярный сигнал обратной связи.
- Показывает, насколько хорошо работает агент на шаге t
- Задача агента - максимизировать кумулятивное вознаграждение

В обучение с подкреплением принята следующая гипотеза:

- Определение (гипотеза вознаграждения)

Все цели могут быть описаны через максимизацию ожидаемого суммарного вознаграждения

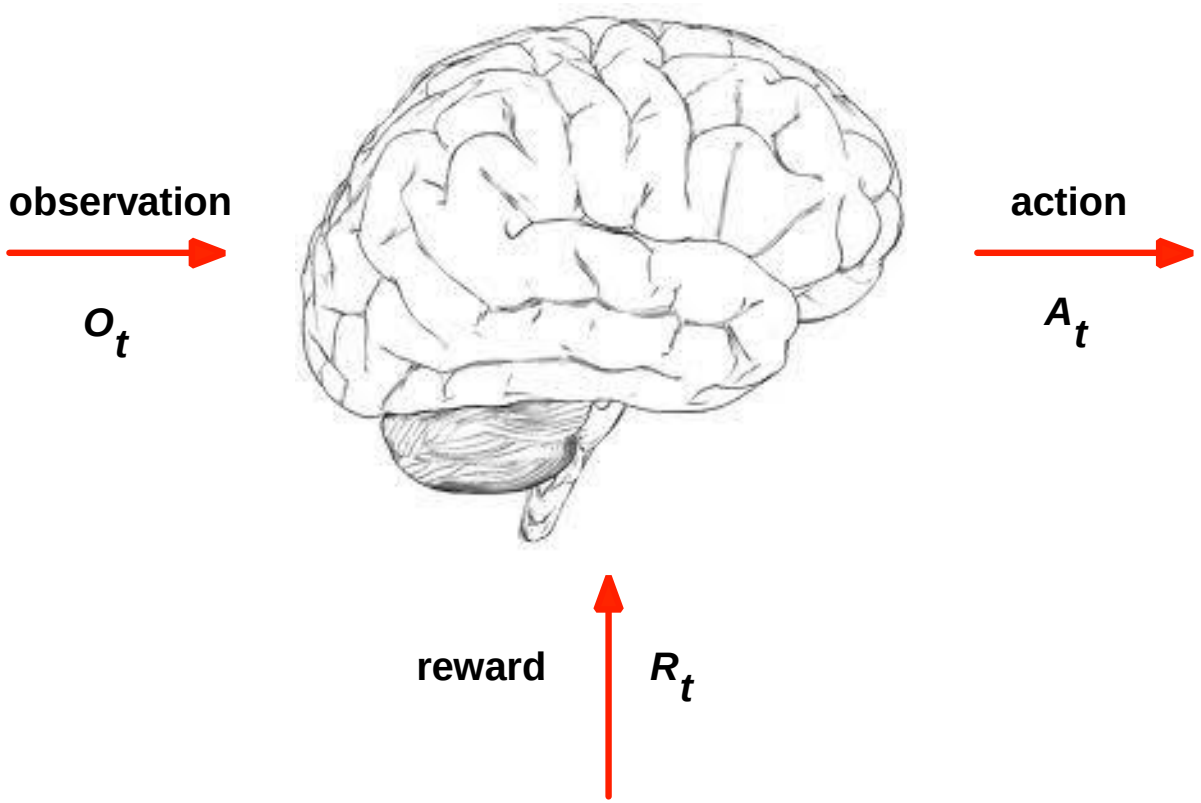
Примеры вознаграждений

- Вертолет: +1 за следование тракетории, -1 за падение
- Го: +1 за победу, -1 за поражение
- Управление инвестиционным портфелем: +r за каждый $, -1 за каждый потерянный $
- Управление электростанцией: +r за производство электроэнергии, -r за превышение порогов безопасности
- Робот: +r за движение вперед, -r за падение
- Atari: +r за увеличение счета, -r за уменьшение счета

Последовательное принятие решений

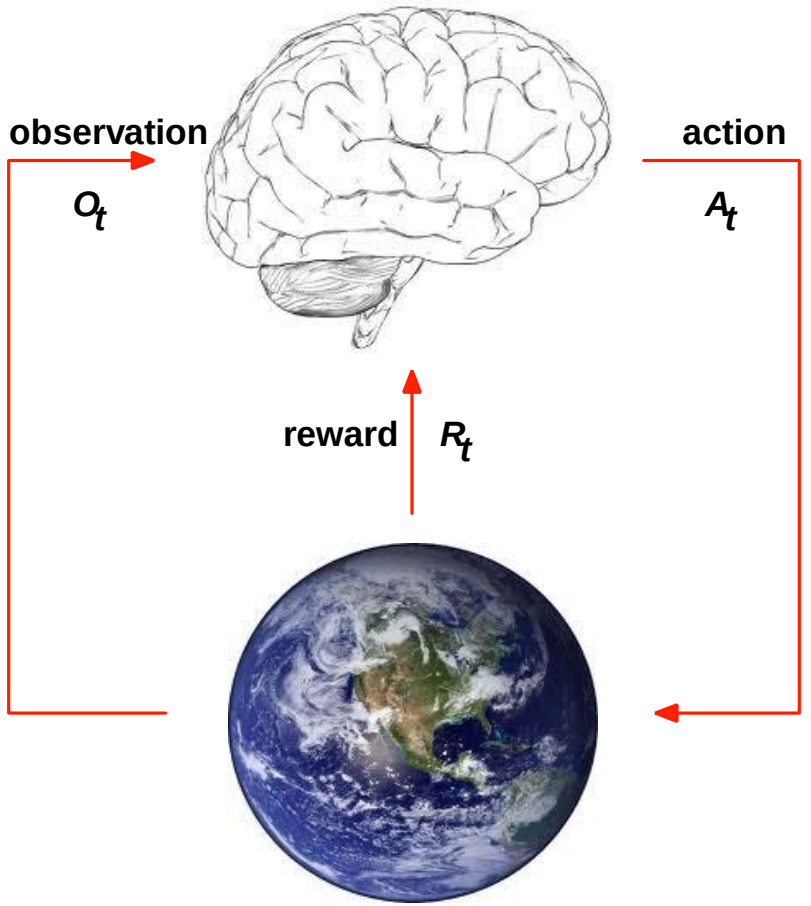
- Цель: выбрать действия которые максимизируют суммарное будущего будущего вознаграждение.
- Действия могут иметь долгосрочные (long term) последствия.
- Вознаграждения могут быть отложенными.
- В некоторых случаях лучше пожертвовать немедленным вознаграждением, чтобы получить более долгосрочное вознаграждение.
- Примеры: Инвестиции (могут пройти месяцы, чтобы получить доход); Заправка вертолета (может предотвратить крушение через несколько часов): Блокирование ходов противника (может повысить шансы на победу через много ходов)

Взаимодействие среды и агента



Взаимодействие среды и агента

- В каждый момент времени t агент:
  - Выполняет действие a_t
  - Получает наблюдение o_t
  - Получает скалярное вознаграждение r_t
- В каждый момент времени t среда:
  - Реагирует на действие a_t
  - Выдает следующее наблюдение o_{t+1}
  - Выдает скалярное вознаграждение r_{t+1}
- Переход к шагу t+1



История и состояние

- История - это последовательность наблюдений, действий и вознаграждений

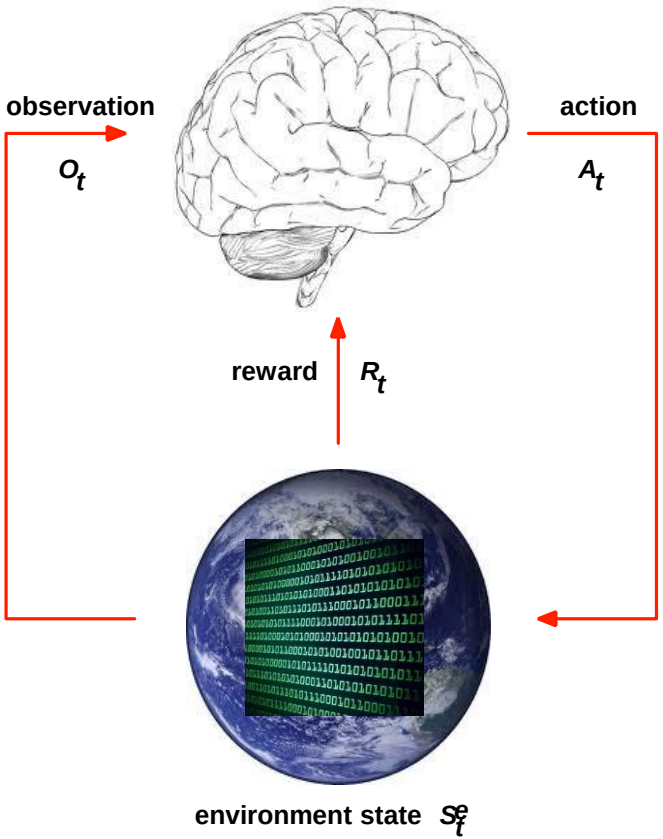
$$H_t = \langle o_1, r_1, a_1, \dots, a_{t-1}, o_t, r_t \rangle$$

т.е. все наблюдаемые переменные до момента времени t

- Все, что происходит дальше, зависит от истории:
  - Агент выбирает действие
  - Окружающая среда генерирует наблюдения и вознаграждение.
- Состояние - это информация, используемая для определения того, что произойдет дальше. Формально, состояние - это функция истории:

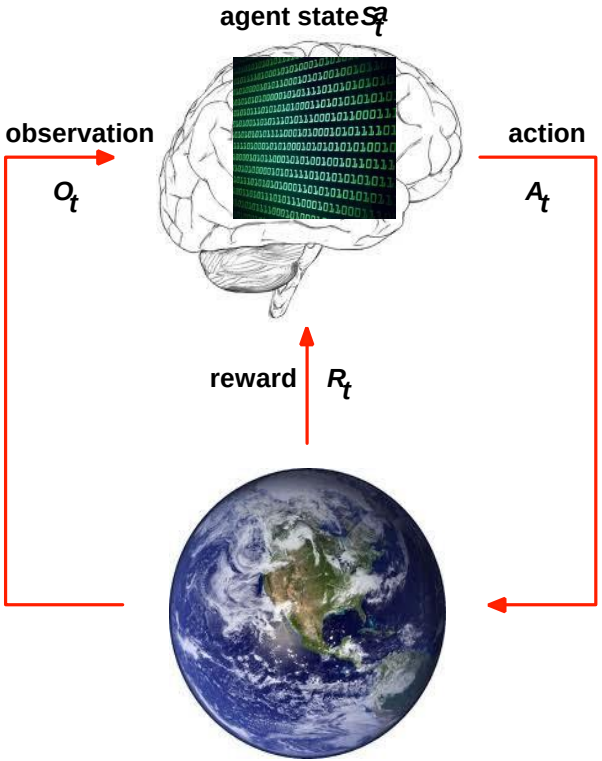
$$S_t = f(H_t)$$

Состояние среды



- Состояние среды S_t^e - это внутреннее представление информации в среде.
- Пример: любые данные, которые среда использует для выбора следующего наблюдения/вознаграждения.
- Состояние среды обычно не наблюдается на прямую агентом.
- Даже если состояние S_t^e доступно, оно может содержать некорректную информацию

Состояние агента



- Состояние агента S_t^a - это внутреннее представление агента.
- Пример: любая информация, которую агент использует для выбора следующего действия.
- Состояние агента обычно является функцией от истории:

$$S_t^a = f(H_t)$$

Марковское состояние

Информационное состояние (оно же марковское состояние) содержит всю полезную информацию, которая может содержаться в истории.

Определение

Состояние s_t является марковским тогда и только тогда, когда

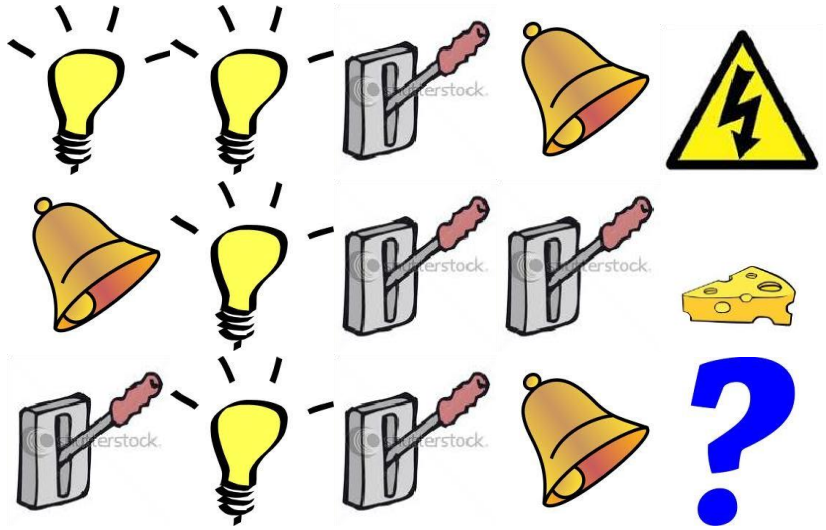
$$P[s_{t+1}|s_t] = P[s_{t+1}|S_1, \dots, s_t]$$

- "Будущее не зависит от прошлого, и определяется только настоящим".

$$H_{1:t} \rightarrow s_t \rightarrow H_{t+1:\infty}$$

- Если известно текущее состояние, история может быть отброшена, т.е. состояние является достаточной статистикой для будущего.
- Предполагаем, что состояние среды s_t^e является марковским

Мышь и сыр



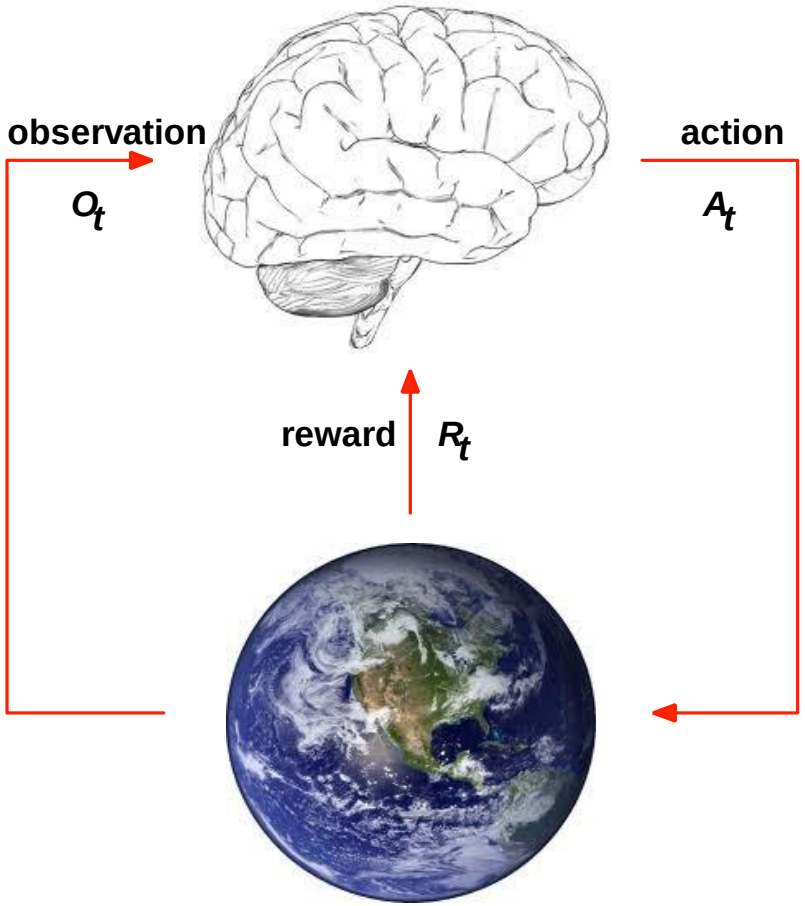
- s_t^e = последние 3 события в последовательности?
- s_t^e = количество появлений света, звонка и рычага?
- s_t^e = вся последовательность?

Полностью наблюдаемые среды

- полная наблюдаемость: агент непосредственно наблюдает за состоянием окружающей среды

$$o_t = s_t^a = s_t^e$$

- Состояние агента = состояние среды = информационное состояние
- Формально, это марковский процесс принятия решений (MDP).



Частично наблюдаемые среды

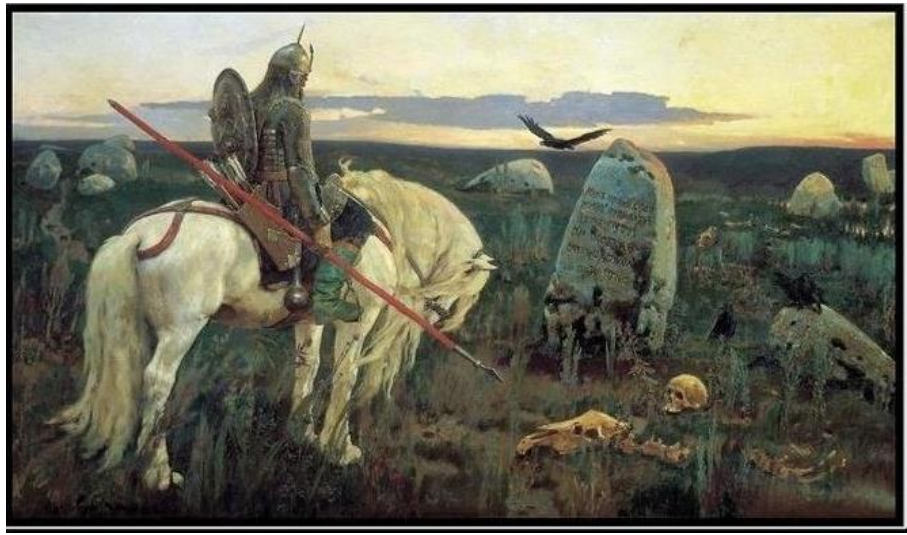
- Частичная наблюдаемость: агент косвенно (опосредованно) наблюдает за средой:
  - Роботу с видеокамерой не сообщается его точное местоположение.
  - Торговый агент наблюдает только текущие цены
  - Игрок в покер, наблюдает только открытые карты.
- Теперь состояние агента не равно состоянию среды
- Формально это ***частично наблюдаемый марковский процесс принятия решений*** (POMDP)
- Агент должен построить свое собственное представление состояния s_t^a, например:
  - Полная история: $s_t^a = H_t$
  - Представления о состоянии среды: $s_t^a = (P[s_t^e = s^1], \dots, P[s_t^e = s^n])$
  - Рекуррентная нейронная сеть: $s_t^a = \sigma(s_{t-1}^a w_s + o_t w_o)$

Строение RL агента

- Агент RL может включать один или несколько из этих компонентов:
  - Стратегия (policy): функция поведения агента
  - Функция полезности (value function): оценка насколько хорошо каждое состояние и/или действие
  - Модель (model): представление агента о среды

Стратегия

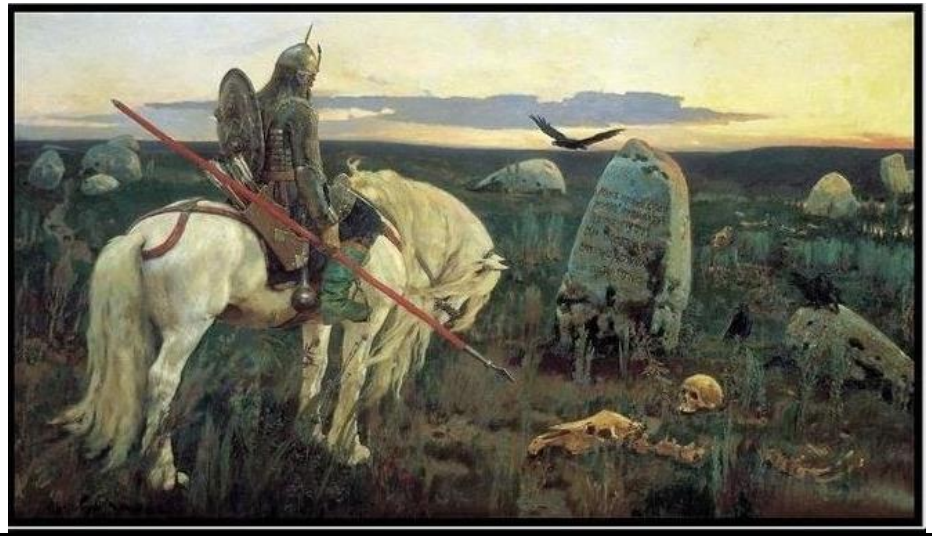
- Стратегия (политика) - это функция поведения агента. Обычно это отображения состояния в действие
- Она представляет собой отображение из состояния в действие, например
  - ✓ Детерминированная политика: *a = π(s)*
  - ✓ Стохастическая политика: $\pi(a|s) = P[a_t = a|s_t = s]$



Функция полезности

- Функция полезности - это предсказание будущего вознаграждения и используется для оценки состояния
- Используется для оценки на сколько состояние является ценным и, следовательно, для выбора между действиями, например

$$V^{\pi} = E_{\pi}[r_t + \Upsilon r_{t+1} + \Upsilon^2 r_{t+2} + \cdots | s_t = s]$$



Модель

- Модель предсказывает, что произойдет в среде в следующий момент времени

Примеры:

- Модель переходов P предсказывает следующее состояние

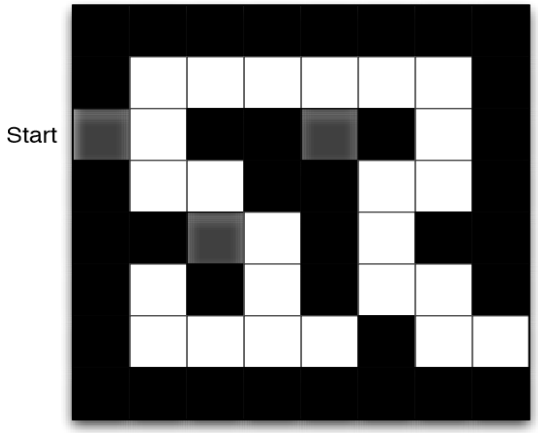
$$P_{ss'}^{a} = P\left[s_{t+1} = s' | s_t = s, a_t = a\right]$$

- Модель вознаграждений R предсказывает следующее (мгновенное) вознаграждение:

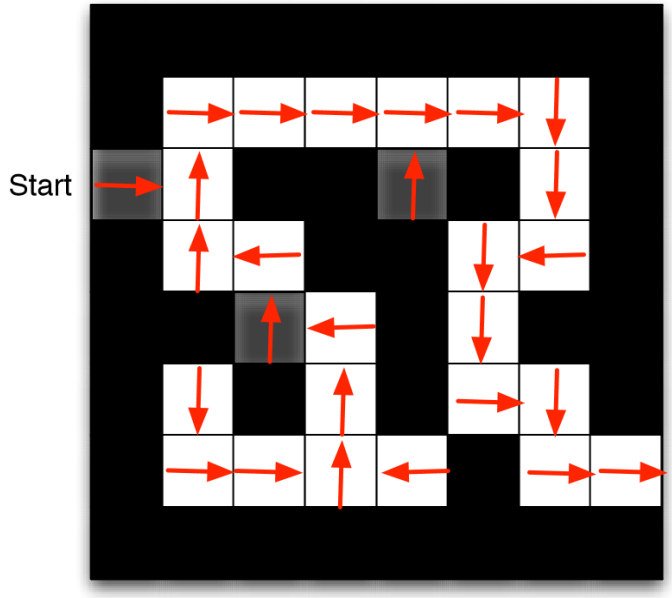
$$R_s^a = E\left[r_t | s_t = s, a_t = a\right]$$

Лабиринт

- Вознаграждение -1 за каждый шаг
- Действие: N,S,W,E
- Состояние: местоположение агента

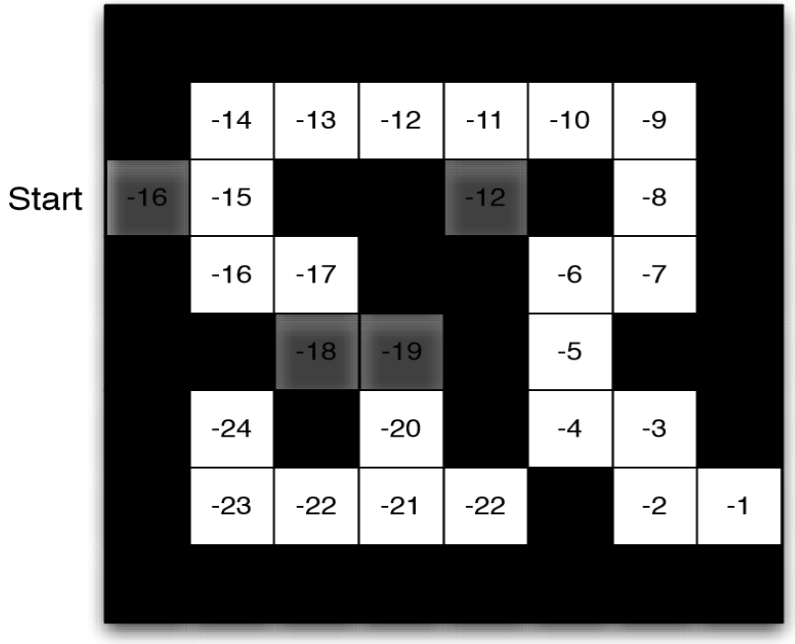


Лабиринт: стратегия



- Arrows represent policy $\pi(s)$ for each state s

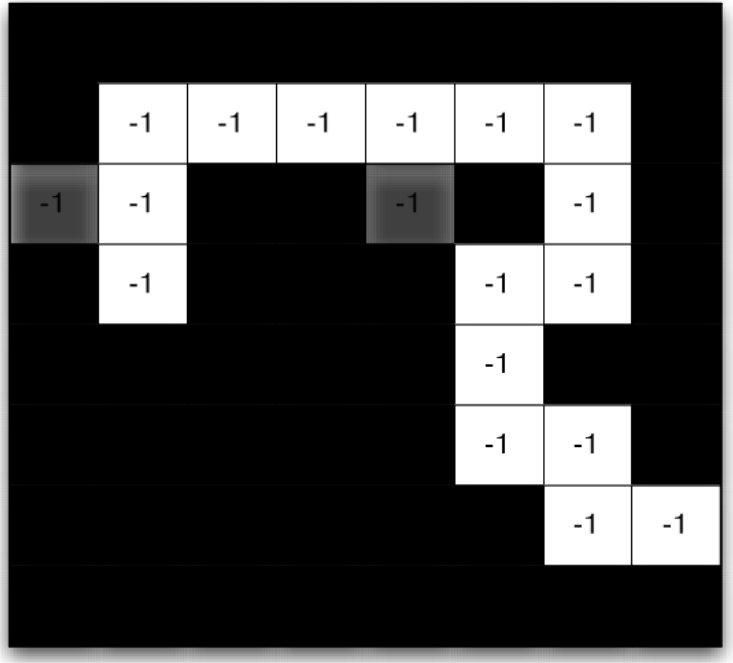
Лабиринт: стратегия



- Numbers represent value $v_\pi(s)$ of each state s

Лабиринт: стратегия

- Агент может иметь внутреннюю модель среды
- Динамика: как действия изменяют среду
- Вознаграждение: какое вознаграждение может быть получено в каждом состоянии
- Модель может быть несовершенной (неточной)
- Клетки представляет модель перехода P_{ss}^{a},
- Числа представляют вознаграждение R_{s}^{a} для каждого состояния s



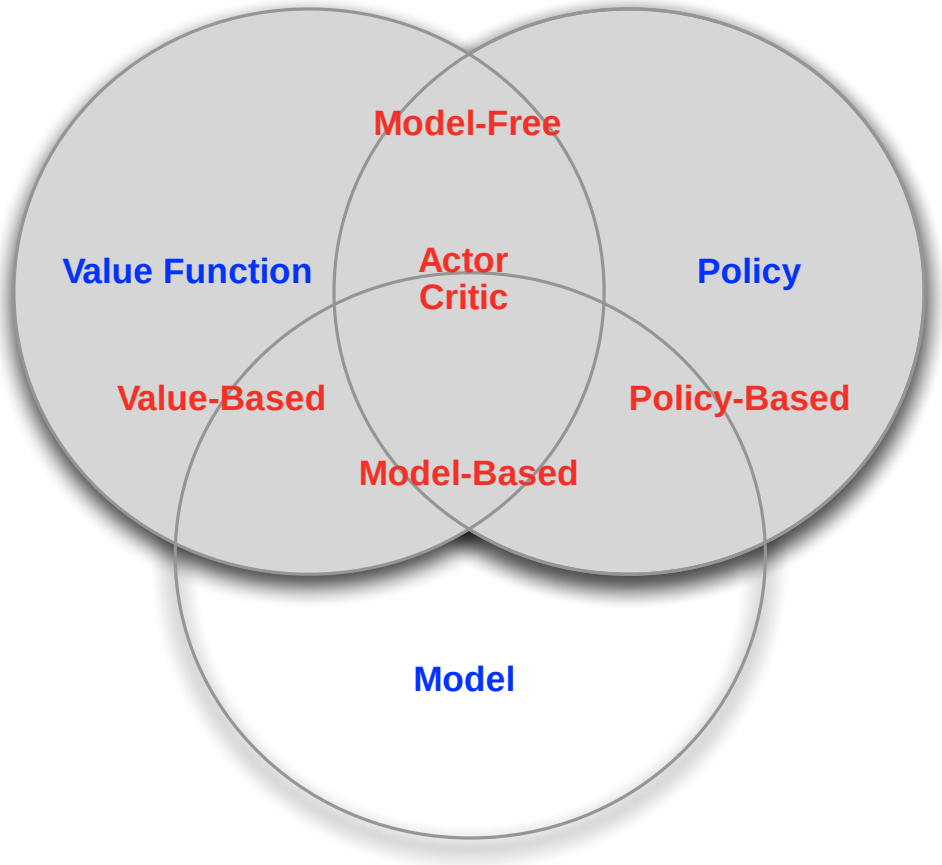
Типизация RL агентов

- Оценивающие функцию полезности (value base)
  - Стратегия не представлена явно
  - Функция полезности
- Оценивающие стратегию (policy base)
  - Стратегия
  - Функция полезности не вычисляется явно
- Актор-критик (actor-critic)
  - Стратегия
  - Функция полезности

Типизация RL агентов

- Безмодельные (model free)
  - Стратегия и/или функция полезности,
  - Нет модели
- Основанные на модели (model based):
  - Стратегия и/или функция полезности
  - Строят модель

Таксономия RL агентов

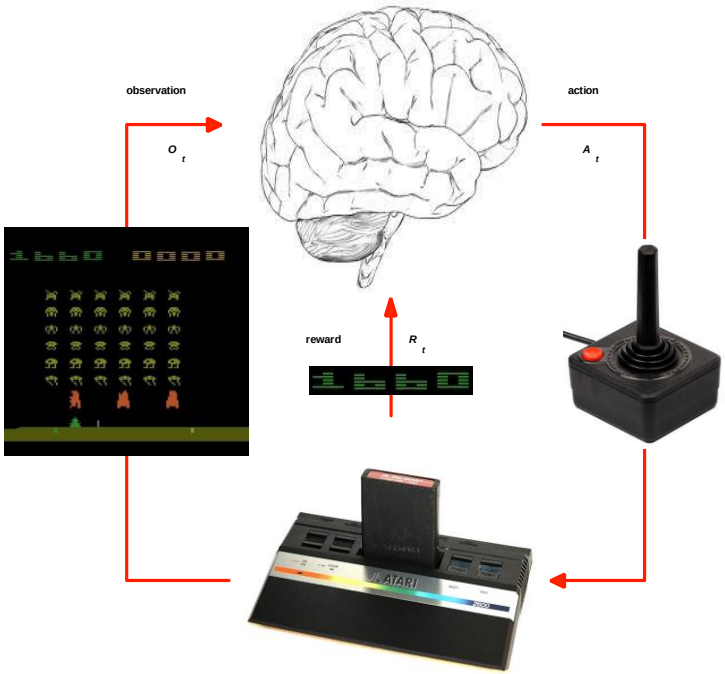


Планирование и обучение

В теории последовательного принятия решения существует 2 основных постановки задачи

- Обучение с подкреплением:
  - Среда изначально неизвестна
  - Агент взаимодействует со средой
  - Агент оптимизирует свою политику
- Планирование:
  - Модель среды известна
  - Агент выполняет вычисления используя свою моделью без взаимодействия со средой
  - Агент оптимизирует свою политику
  - также известный как обдумывание, рассуждение, интроспекция, размышление, поиск

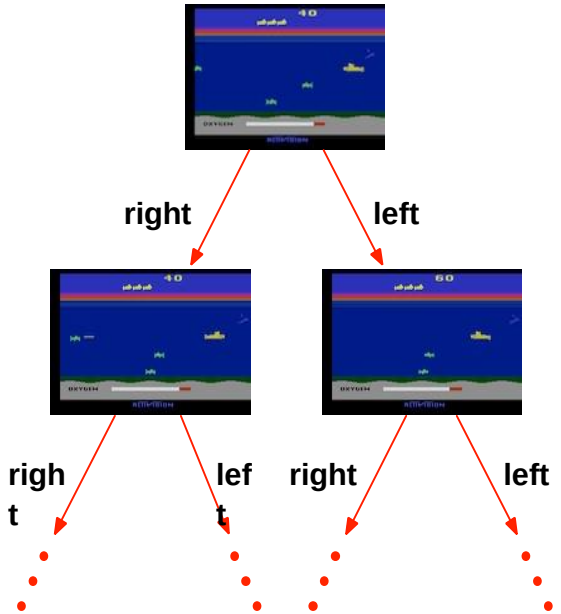
Пример: Atari - обучение



- Правила игры неизвестны
- Обучается напрямую в игре
- Подача действий на джостик, восприятие пикселей изображения

Пример: Atari - планирование

- Правила игры известны
- Можно запросить эмулятор - идеальную модель внутри агента
- Если я предприму действие a из состояния s: каким будет следующее состояние? Каким будет счет?
- Планируем заранее, чтобы найти оптимальную политику, например поиском по дереву



Исследование и применение

- Обучение с применением подкрепления похоже на обучение методом проб и ошибок
- Агент должен найти оптимальную политику
- Агент основывается на опыте работы со средой
- При этом, желательно не терять слишком много вознаграждения на этом пути

Исследование и применение

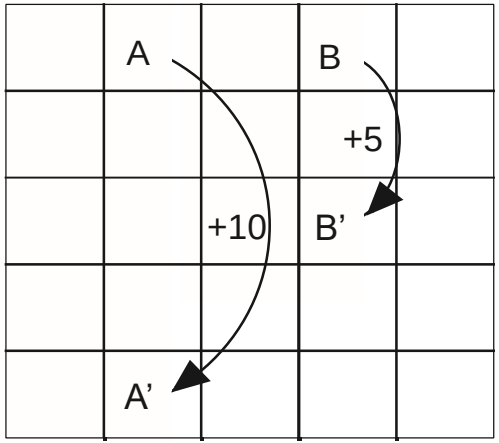
- Исследование (exploration) – это процесс поиска новой информации о среде
- Применение (exploitation) – процесс использования новой информации для максимизации вознаграждения
- Обычно важно как исследовать так и применять.

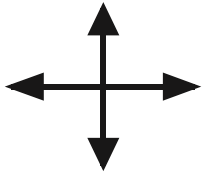
Примеры

Предсказание и управление

- Прогнозирование: оценка будущего
  - С учетом политики
- Управление: оптимизация будущего
  - Найти наилучшую политику

Клеточный мир: предсказание

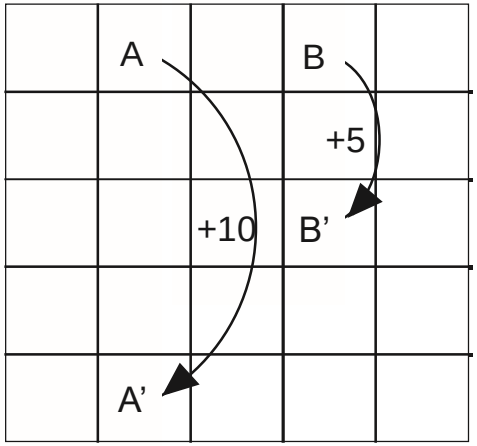




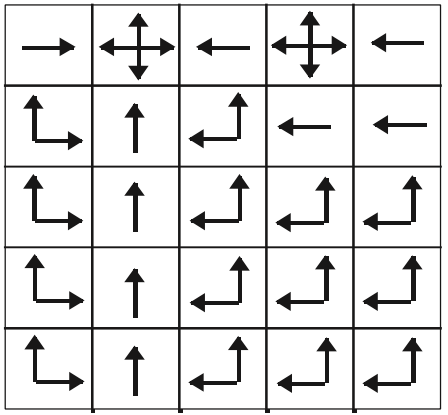
3.3	8.8	4.4	5.3	1.5
1.5	3.0	2.3	1.9	0.5
0.1	0.7	0.7	0.4	-0.4
-1.0	-0.4	-0.4	-0.6	-1.2
-1.9	-1.3	-1.2	-1.4	-2.0

Функция полезности для случайной равномерной стратегии

Клеточный мир: управление



22.0	**24.4**	**22.0**	**19.4**	**17.5**
19.8	22.0	19.8	17.8	16.0
17.8	19.8	17.8	16.0	14.4
16.0	17.8	16.0	14.4	13.0
14.4	16.0	14.4	13.0	11.7



Оптимальные функции полезности π^* и оптимальная стратегия V^*