能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

机器人 人工智能 机器学习 统计学习 强化学习 (Reinforcement Learning)

关注者 898 被浏览 41743

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学习的不同？

随着DeepMind和AlphaGo的成功，强化学习(Reinforcement Learning)日益受到关注。然而，在一些机器学习入门课程中，并没...显示全部

2 条评论 分享 邀请回答 ••• 关注问题 写回答

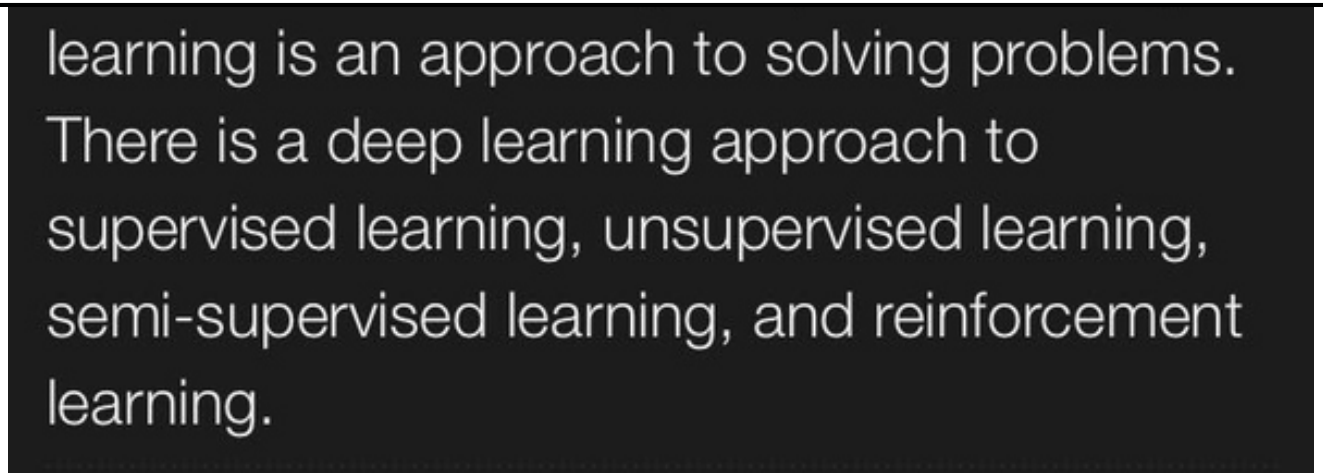
11 个回答 默认排序

知乎用户

83 人赞同了该回答

▲ 83 ▼ 7 条评论 分享 收藏 感谢 收起

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...



#2015-03-28

强化学习（RL）的基本组件：

- 环境（标准的为静态stationary，对应的non-stationary）
- agent（与环境交互的对象）
- 动作（action space，环境下可行的动作集合，离散or连续）
- 反馈（回报，reward，正是有了反馈，RL才能迭代，才会学习到策略链）

细看下来，分明与监督学习（SL），无监督学习（UL）是不同的类别，RL更像控制系统家族里的。是的，RL里流着控制的血液，披着机器学习的外衣，这是因为它需要data，需要training以此来支持决策。RL可以decision-making，不同于决策树之类的决策（称为预测比较好），是控制角度的决策，意味着就有失误，伴随着收益与惩罚（股票，博弈，游戏得分等等）。细一点来说，RL与SL的区别有：

1. 喂数据的方式不同：强化学习（RL）的数据是序列的，交互的，并且还是有反馈的
（Reward）-【MDP]。这就与

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

2. RL的target是估计得来的，符合bellman等式，SL的target是fixed label；RL可以融合SL来训练，RL还可以自己博弈来生成样本。[交互特性，也可以放到第一点中]
3. RL可以进行lifelong形式的学习。RL有“生命”的【你可能也不知道你训练出来的模型到底能干什么】，SL没有。

#2016-04-05补充：
上述提到了RL的基本组件，那么为何不同于SL？首先我认为RL=learning+decision-making，不是有“训练”数据就可以的（确切来讲训练一词不妥当），当action作用于环境，收到的reward是delayed，delayed意味着当前的action会影响long-term gain（RL的优化目标），SL里根本不存在这个概念；有意思的是RL的优化目标与SL或者UL也是截然不同的，SL/UL是尽可能的拟合当前一堆不会思考的数据，而RL考虑到了predictive control，会赋予当前的交互数据以决策权重。既然action会影响long-term gain，那么怎么选择action？这就不得不提exploration（探索）模块，又是一个行为心理学上的概念，可以说没有探索，RL学不到policy。因此在做RL研究的时候，始终要考虑到decision-making。不乏有人会说可以用SL为RL预训练一个base model。对，我觉得这一环节就像用到参数初始化的地方怎么选择初始化方法一样。扯远一点，当前RL的学习大部分是first-order的（从模型的更新方式角度来看），second-order怎么做或者是否可行？因为RL收敛到一个good policy实在很慢！

#2017-02-03:

introtodeeplearning.com...

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

predict **labels**

Unsupervised Learning: given **data**, learn about that **data**

Reinforcement Learning: given **data**, choose **action** to maximize expected **long-term reward**

编辑于 2017-02-03

甄景贤
研究普适人工智能

140 人赞同了该回答

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

甄景贤 (King-Yin Yan)

General.Intelligence@Gmail.com

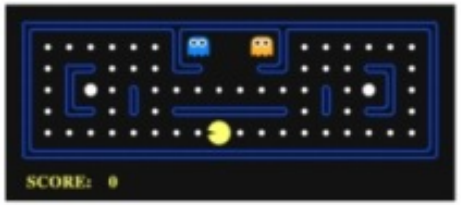
1 什么是强化学习？

Reinforcement learning 是机器学习里面的一个分支，特别善於控制一只能够在某个环境下 自主行动 的个体 (autonomous agent)，透过和 环境 之间的互动，例如 sensory perception 和 rewards，而不断改进它的 行为。

听到强化学习，你脑里应该浮现一只甲由那样的小昆虫，那就是 autonomous agent 的形象：

(1)

对「环境」(environment) 这概念，你应该想到像以下这经典游戏的迷宫：



(2)

包括有追捕你的怪物、和吃了会加分的食物（这些代表负值和正值的 rewards）。当然，实际应用的「环境」和「奖励」可以是抽象的。这游戏是一个很具体的例子

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

记住，reinforcement learning 的 输入 是：

- 状态 (States) = 环境，例如迷宫的每一格是一个 state

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

- 奖励 (Rewards) = 进入每个状态时，能带来正面或负面的价值 (utility)

而输出就是：

- 方案 (Policy) = 在每个状态下，你会选择哪个行动？

於是这 4 个元素的 tuple $(S\ A\ R\ P)$ 就构成了一个强化学习的系统。在抽象代数中我们常常用这 tuple 的方法去定义系统或结构。

再详细一点的例子就是：

- states S = 迷宫中每一格的位置，可以用一对座标表示，例如 (1,3)
- actions A = 在迷宫中每一格，你可以行走的方向，例如：{ 上，下，左，右 }
- rewards R = 当前的状态 (current state) 之下，迷宫中的那格可能有食物 (+1)、也可能有怪兽 (-100)
- policy P = 一个由状态 → 行动的函数，意即：这函数对给定的每一个状态，都会给出一个行动。

(S, A, R) 是使用者设定的，P 是算法自动计算出来的。

1.2 人与虫之间

第一个想到的问题是：为什
只对比较细小和简单的环境

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

而强化学习只是关心简单的「状态－行动」配对。

强化学习的领导研究者 Richard Sutton 认为，只有这种学习法才考虑到自主个体、环境、奖励等因素，所以它是人工智能中最 top-level 的 architecture，而其他人工智

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

(3)

所以要制造 strong AI，一个可能的方案就是结合强化学习和某种处理复杂 world model 的能力：

(4)

「你们已经由虫进化成人，但在你们之内大部份仍是虫。」

— 尼采, Thus spoke Zarathustra

「如果人类不相信他们有一天会变成神，他们就肯定会变成虫。」

— Henry Miller

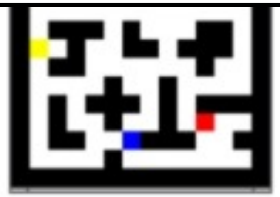
1.3 程式

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

Reinforcement learning demo

只要 Python 便可运行，但你可能要 install PyGame。

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...



(5)

猫的行动是简单地朝着老鼠追（没有智能），老鼠的行动是学习出来的。

注意，在 main program 和 cellular.py 这两部分，纯粹是定义了迷宫世界如何运作，基本上是一个 game，里面完全没有智能，你可以用｛上、下、左、右｝控制各 agent 的活动，如此而已。

强化学习的程式在 qlearn.py，很短，而真正学习的程式基本上只有一句，就是：

```
def learnQ(self, state, action, reward, value):
oldv = self.q.get((state, action), None)
if oldv is None:
      self.q[(state, action)] = reward
else:
      self.q[(state, action)] = oldv + self.alpha * (value - oldv)
```

单是这一句程式，就能令老鼠学到避开猫、吃芝士。以下再解释....

1.4 強化學習的原理

《AI — a modern approach》这本书第 21 章有很好的简介。《AIMA》自然是经典，很多人说他们是读这本书而爱上 AI 的。这本书好处是，用文字很耐性地解释所有概念和原理，思路很清晰，使
reinforcement learning，意思

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

为免重复，我只解释到明白 Q learning 的最少知识。

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

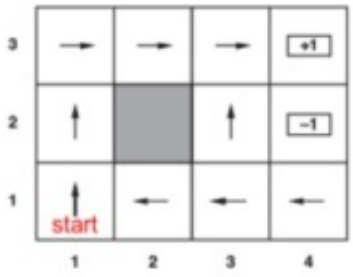
U 是一连串行动的 rewards 的总和。例如说，行一步棋的效用，不单是那步棋当前的利益，还包括走那步棋之后带来的后果。例如，当下贪吃一只卒，但 10 步后可能被将死。又或者，眼前有美味的食物，但有些人选择不吃，因为怕吃了会变肥。

一个 state 的效用 U 就是：假设方案固定，考虑到未来所有可能的 transitions，从这个 state 开始的平均期望的 total reward 是多少:

$$U(S_0) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t \, R(S_t)\,]$$

其中 $\mathbb{E}[\,]$ 代表期望值，γ 是 discount factor，例如 0.9 或什么。

实例：考虑这简单的迷宫：



(6)

那些箭咀表示的是众多可能方案中的其中一个。

根据这个方案，由 (1,1) 开始的运行可能是这个结果：

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

但从同一起点，同一方案，也可以出现不同结果，例如在 (1,3) 企图向右爬，但实际结果是向下跌一格；这些 state transitions 是由外在世界的机率决定的。（例如某人读了大学文凭，但遇上经济不景，他的薪水未必能达到行动的预期效果。）

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

-0.04 -0.04 -0.04 -0.04 -0.04 -0.04 -0.04 +1 (8)

或者：

(1,1) ⟶ (2,1) ⟶ (3,1) ⟶ (3,2) ⟶ (4,2)
-0.04 -0.04 -0.04 -0.04 -1 (9)

1.6 Bellman condition

这是 dynamic programming（动态规划）的中心思想，又叫 Bellman optimality condition。

在人工智能里我们叫 reinforcement learning，但在控制论的术语里叫 dynamic programming，两者其实是一样的。Richard Bellman 在 1953 年提出这个方程，当时他在 RAND 公司工作，处理的是运筹学的问题。他也首先使用了"curse of dimensionality"这个术语，形容动态规划的主要障碍。

考虑的问题是：要做一连串的 sequential decisions。

Bellman equation 说的是：「**如果从最佳选择的路径的末端截除一小部分，馀下的路径仍然是最佳路径。**」

换句话说，如果一系列的选择 A B C D E.... 是最优的，那么这系列除去开始的 A，那 B C D E.... 系列应用在后继的状态上也是最优的。

（例如，你从香港乘车到北京，选择了最便宜的路线，此路线经过 10 个车站，第二站是深圳：

香港 → 深圳 → ..... → 北京

但如果除去出发点香港站，那么由第二站深圳到最后的北京站：

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

用数学表示：

$$U^*(S) = \max_a \{R(a) + U^*(S')\}$$

$$U^*(\text{全路径}) = \max_a \{R(\text{在当前状态下选取 a}) + U^*()\}$$

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

题：换句话说它是一个 recursive relation。

1.7 Delta rule

这只是一个简单的 trick，在机器学习中经常出现。假设我们有一个理想，我们要逐步调教当前的状态，令它慢慢趋近这个理想。方法是：

$$\text{当前状态} := \text{当前状态} + \alpha(\text{理想} - \text{当前状态})$$

其中 α 叫「学习速度 (learning rate)」。"Delta" (Δ) 指的是理想和现状之间的差异。

很明显，只要反覆执行上式，状态就会逐渐逼近理想值。

（Delta rule 的微分形式就是我们熟悉的「梯度下降」：$x \mathrel{+}= \eta \cdot \frac{dy}{dx}$）

1.8 Temporal difference (TD) learning

将 delta rule 应用到 Bellman condition 上，去寻找最优路径，这就是 temporal difference learning。

我们还是从简单情况开始：假设方案固定，目标是学习每个 state 的 utility。

理想的 $U(S)$ 值，是要从 state S 开始，试验所有可能的 transitions，再计算这些路径的 total rewards 的平均值。

但实际上，agent 只能够每次体验一个行动之后的 state transition。

所以要应用 Bellman conditi

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

$$\sum_{S'}$$

其中 P 是 transition 的机率，S' 是后继 state，$\sum$ 是对所有后继 states 求和。换句话说，这是理想的 $U(S)$ 和 $U(S$的后继$)$ 之间的关系，是一个 recursive relation。

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

那么这两个 states 的 U 值，应该符合这条约束：

$$U(1,3) = -0.04 + U(2,3)$$

换句话说，这是两个 states 之间，U 值的 local (局部的) 约束。

TD learning 的思想是：假设其他 $U(S')$ 的值正确，利用 Bellman optimality 来调整当下 state 的 $U(S)$。当尝试的次数多了，所有 U 值都会趋向理想。Agent 只需要用这条 update rule：

$$U(S) \mathrel{+}= \alpha(\ R(S) + \gamma U(S') - U(S)\ )$$

α 是 learning rate，它决定学习的速度（但它不能太大，避免 overshooting）。后面那东西是 $U(S)$ 和 $U(S)$ 的估值 (estimation) 之间的差别。对於理想的 $U(S)$ 和 $U(S')$，那差别会是 0。而在每个 time step，我们只是用 α 部分地调整这个差别。

最后一提，在上面理想约束的公式里，有对於机率 P 的求和，但在 update formula 中 P 不见了。那是因为 agent 在环境中的行动，暗含了对於 state transition 机率的 sampling (随机地取样本)。换句话说，那机率求和是由 agent 本身体现的。

P 是 state transitions 的机率，换句话是关於世界的一个 model。TD learning 不需要学习 P，所以叫 model-free learning。但正如开篇时说过，model-free 并不一定是好事，人的智慧就是基於我们对外在世界有一些很好的 models。

1.9 Q value

Q 值只是 U 值的一个变种；U 是对每个 state 而言的，Q 把 U 值分拆成每个 state 中的每个 action 的份量。换句话说，Q 就是在 state S 做 action A 的 utility。

Q 和 U 之间的关系是：

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

上面的 update rule 只要用这个关系改写就行：

$$U(S) \mathrel{+}= \alpha(\ R(S) + \gamma \max_{A'} Q(A', S') - Q(A, S)\ )$$

1.10 Active learning

在 passive learning 中，方案不变，我们已经能够计算每个 state S 的效用 $U(S)$，或者每个 state S 之下行动 A 的效用 $Q(S, A)$。

如果方案是可以改变的，我们只需计算不同方案的 Q 值，然后在每个 state S 选取相应於最大 Q 值的行动 A，那就是最佳方案，不是吗？

实际上执行的结果，却发现这些 agent 的方案很差！原因是，学习过程中的 Q 值是 estimate，不是理想的 Q 值，而如果根据这样的 Q 行动，agent 变得很短视，不会找到 optimal policy。（例如，某人经常吃同一间餐馆，但循另一路径走，可以发现更好的餐馆。）

Agent 需要尝试一些未知的状态 / 行动，才会学到 optimal policy；这就是所谓的 exploration vs exploitation （好奇心 vs 短暂贪婪）之间的平衡。

方法是，人工地将未知状态的价值增加一点：

$$U(S) = R(S) + \gamma \max_{A} \mathcal{F}[\ \sum_{S'} P(S \to S')U(S'),\ N(A, S)\ ]$$

其中 $N(A, S)$ 是状态 S 和行动 A 这对组合出现过（被经历过）的次数，$\mathcal{F}$ 是 exploration 函数，它平时回覆正常的 U 的估计值，但当 N 很小时（亦即我们对 S，A 的经验少），它会回覆一个比较大的估值，那代表「好奇心」的效用。

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

什么是强化学习？

我正在研究的 AI architecture 是用强化学习控制 recurrent 神经网络，我相信这个设置可以做到逻辑推理和答问题的功能，基本上就是 strong AI。 但还有一些未解决的细节。 论文的标题是《游荡在思考的迷宫中》，即将发表。

補充： 還有一點，就是監督學習的問題可以很容易化為強化學習的問題（雖然這樣增加了複雜性而沒有益處），但反之則沒有一般的辦法。 見：Reinforcement Learning and its Relationship to Supervised
Learning，Barto and Dietterich, 2004.
"But is it possible to do this the
other way around: to convert a reinforcement learning task into a supervised learning task?
"In general, there is no way to do this. The key difficulty is that whereas in supervised learning, the goal is to reconstruct the unknown function f that assigns output values y to data points x, in reinforcement learning, the goal is to find the input x* that gives the maximum reward R(x*).
"Nonetheless, is there a way that we could apply ideas from supervised learning to perform reinforcement learning? Suppose, for example, that we are given a set of training examples of the form (xi, R(xi)), where the xi are points and the R(xi) are the corresponding observed rewards. In supervised learning, we would attempt to find a function h that approximates R well. If h were a perfect approximation of R, then we could find x* by applying standard optimization algorithms to h."

编辑于 2016-11-24

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

22 人赞同了该回答

以下我尝试提出一个尽量能够统一 SL(supervised learning), OL(online learning), RL(reinforcement learning) 的framework:

---

for $t = 1, 2, \ldots, T$

1. 观测到数据 $x_t \in \mathbb{R}^n$
2. 选择action $a_t \in \mathcal{A}$
3. 得到损失 $l(x_t, a_t(x_t))$

目标是：

$$\min_{a_t \in \mathcal{A}} \sum_{t=1}^{T} l(x_t, a_t(x_t))$$

SL:

给定**i.i.d.的样本** $x_1, x_2, \ldots, x_T \in \mathbb{R}^n$ ，我们的目标是选择action $a : \mathbb{R}^n \to \mathcal{Y}$ 来

$$\min_{a \in \mathcal{A}} \sum_{t=1}^{T} l(x_t, a(x_t))$$

在SL literature里， $\mathcal{A}$ 通常被称为hypothesis class/space，**loss function一般具有如下形式**：

$l(x_t, a(x_t)) = d(a(x_t), y_t)$ ，其中 d 为度量空间中的某种距离。

SL的另一个特殊性是**action不依赖于时间**，因为在SL里所有数据一般共享模型参数。

可能有童鞋在这里会对此framework提出反驳，因为SL的目标并不是最小化在training set上的error，而是generalization error。不过这个可以体现在loss function里面，比如loss function中可以包含regularizer。

OL:

OL与SL的主要区别在于**丢弃了i.i.d.假设**，数据是按时间顺序来的，在每一个时间点都要选择action，并且利用数据不断的更新 $l(x_t, a(x_t)) = d(a(x_t), y_t)$

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

RL:

RL的特殊性在于**时刻 t 的action可以影响时刻 $t+1$ 我们得到的data**。具体的，在时刻 t ，我们观测到 x_t (在RL里 x_t 一般被称作state)，同时选择action $a_t \in \mathcal{A}$ ，然后suffer loss $l(x_t, a_t(x_t))$ ，并且 $x_{t+1} \sim P(x'|x_t, a_t(x_t))$ 。

所以从这个角度看，RL是一种**更active的learning**，因为我们可以用自己的action来决定如何探索state space。

RL里面partially observable的情况就更难一些，我们不能直接观测到state，在这种情况下 x_t 是某种观测量，然后通过 x_t 来更新对当前所在state的belief。

当然RL里面对于最终优化的目标还有一些其他的formulation，比如infinite horizon discounted cost对应 $T \to \infty, l(x_t, a_t(x_t)) = \gamma^t g(x_t, a_t(x_t))$ 。

Note: optimal control theory也是这个framework的一种特殊情况，在RL中如果state transition是一个deterministic的dynamics: $x_{t+1} = f(x_t, a_t(x_t))$ ，这就对应了optimal control。

关于行话：做RL的人由于整天面对着逆天难的问题，所以喜欢用reward(相对于loss)来激励自己。面对难题，乐观的态度还是蛮重要的 :)

编辑于 2016-03-29

知乎用户

2 人赞同了该回答

强化学习结合现在深度学习强大特征抽象能力，能不能算是好的车遇到了老司机呢

发布于 2016-05-06

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

2 人赞同了该回答

有朋友关注这个问题,我简单答下.

在我的理解中,Reinforcement Learning与普通Supervised Learning最大的区别在于,其训练包含着一个尝试的过程.这里用不用深层神经网络方法其实相对无关紧要.

比如图像或者文本分类,普通的分类算法会用某种方式提取特征,然后进行SVM或logistic regression.

但对于博弈类游戏,比如围棋等,会有一个两个Agent互相用已有的模型制订策略,并根据最后的结果修正自己的模型的过程;或者是在寻路,控制算法中,会有一个根据表现的优劣来更新自己权重,尝试向更好的参数逼近的过程.

比如Feifei Li的 "Target Driven Visual Navigation in Indoor Scenes using Deep Reinforcement Learning"这篇文章里,用机器人看虚拟房间的方式并尝试寻路的方式,进行房间内按图像寻找位置和最优路径的任务训练.

之前Andrew Ng有篇文章讲直升机控制,也是这个原理."Autonomous Helicopter Flight Via Reinforcement Learning".

大概就是这样.

这种问题的难点在于非凸性.普通的classification,如果用深层神经网,那么optimizer的选择本身就是个问题.如果再涉及到reinforcement这种含有尝试性质的迭代过程,而不是用固有的数据集,那非凸性很可能就更加明显.目前的深层神经网方法对非凸性还没有一个非常好的,有数学证明的解决方式,一切都是靠摸索出来的经验进行.

发布于 2016-11-04

▲ 2 ▼ 添加评论 分享 收藏 感谢

加油毕业鹏先生
PhD candidate at UNSW

6 人赞同了该回答

▲ 83 ▼ 7 条评论 分享 收藏 感谢 收起

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

1. RL 是是一个序列预测的问题，这个问题与我们经常接触的time series（如stock selection）的区别在于，我们无法得到一个真正的target value来完成我们的loss function，而是用的自定义的反馈函数。

2. 利用反馈来优化RL带来的好处在于能够兼顾其对long term收益，对于一些需要长期策略支持的问题特别有效。举个栗子，下围棋和trading的策略有些是需要放长远的，短期的loss在长远来可能变成gain，而普通的SL对这种需要长短期记忆的问题处理得还相当简单直接。

总的说来，RL是一个需要长短期记忆的序列问题，其算法也有好几类，目前deep learning中炙手可热的lstm就可以用来解决RL的问题。所以我们在学习机器学习的时候，一定要注意如何定义问题，然后再针对不同的问题，不同的条件积累相应的算法。

好就酱，期待awkkk的更新！

发布于 2016-04-06

▲ 6 ▼ 添加评论 分享 收藏 感谢

止忽
乘着地球的旅行家

1 人赞同了该回答

Reinforcement Learning学习不是单一方法，而是一种机器学习方式。

怎样的方式？是将机器学习算法和环境互动结合起来的方式。

这样在一个有限域内，只要时间充足(或做算力足够。往往都能在一个设定目标中获得较优解。

发布于 2016-12-09

▲ 83 ▼ 7 条评论 分享 收藏 感谢 收起

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

Rosie

3 人赞同了该回答

对于初学者而言，只需要记住，深度学习常用于做分类与识别，而强化学习常用于控制，深入的则需要自己慢慢体会了

发布于 2016-12-11

▲ 3 ▼ 添加评论 分享 收藏 感谢

卢瑟勃

3 人赞同了该回答

强化学习是通过对未知环境一边探索一边建立环境模型以及学得一个最优策略。有监督学习则是事先给你了一批样本，并告诉你哪些样本是优的哪些是劣的（样本的标记信息），通过学习这些样本而建立起对象的模型及其策略。在强化学习中没有人事先告诉你在什么状态下应该做什么，只有在摸索中反思之前的动作是否正确来学习。从这个角度看，可以认为强化学习是有时间延迟标记信息的有监督学习。

发布于 2016-12-07

▲ 3 ▼ 添加评论 分享 收藏 感谢

刘瀚阳

重度游戏玩家，鹅厂高级工程师、博客jjyy.guru、《游戏编程算法与技巧》译者

1 人赞同了该回答

感觉回答都挺复杂的，我写个自己

1. 让机器人尽可能探索多的路径，

▲ 83 ▼ 7 条评论 分享 收藏 感谢 收起

能否介绍一下强化学习(Reinforcement Learning)，及其和监督学...

发布于 2017-01-30

▲ 1 ▼ 添加评论 分享 收藏 感谢

蓝颂

个人理解，强化学习是没有谁告诉你下一步该怎么走，要自己摸索，每一个动作对应一个奖赏，最后得到一个奖赏最大的方式进行数据处理。而监督学习是有数据和标签，通过反向传播算法的反馈来调节参数，直到误差最小

发布于 2017-05-24

▲ 0 ▼ 添加评论 分享 收藏 感谢

▲ 83 ▼ 7 条评论 分享 收藏 感谢 收起