**强化学习和监督学习对比：**

1.监督学习，对有标注的数据进行学习。有分类和回归两类。以分类为例，训练过程中，我们给神经网络真实的标签，比如现在输入了汽车的图片，它预测出来是飞机。我们就会直接告诉它，你这个预测是错误的，正确的标签应该是车，并通过损失函数进行反向传播来训练网络。

监督学习中两个假设：  
 （1）数据有标签

（2）数据没有关联

有标签很好理解，我们得知道答案才能进行指导。没有关联就是我们希望输入的数据具有总体代表性，如果数据不具有总体代表性，那么就是特例的情况，那规律就会总结得不好或是错误，因为这些规律是由个例推算的，不具有推广的效果。比如训练矩阵分类器，输入数据都是有关联的，例如正方形，那么学习到的模型就只能判别正方形是矩形，因为他不知道长宽不等的长方形也是矩形。数据应该要满足独立同分布，独立代表数据之间无关联，同分布说明数据样本服从同一个分布，比如掷骰子，就服从同一个分布，每个数字的概率都是1/6。

1. 强化学习数据是相关的时间序列数据，不满足独立同分布，一段时间内智能体采取的动作是连续性，有关联。面临延迟奖励 (Delayed Reward)，在强化学习里面，环境可能会告诉你这个行为是错误的，但是它并没有告诉你正确的行为是什么。而且更困难的是，它可能是在一两分钟过后告诉你错误，它再告诉你之前的行为到底行不行。

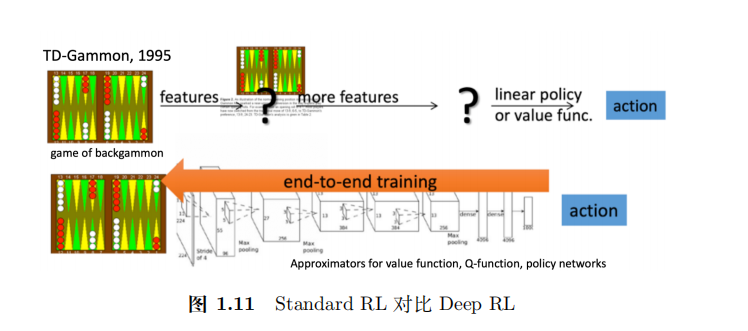
现实生活中强化学习的例子：

• 在自然界中，羚羊其实也是在做一个强化学习，它刚刚出生的时候，可能都不知道怎么站立，然后它通过试错的一个尝试，三十分钟过后，它就可以跑到每小时 36 公里，很快地适应了这个环境。

**强化学习和深度强化学习：**

Standard RL：之前的强化学习，比如 TD-Gammon 玩 backgammon 这个游戏，它其实是设计特征，然后通过训练价值函数的一个过程，就是它先设计了很多手工的特征，这个**手工特征可以描述现在整个状态**。得到这些特征过后，它就可以通过训练一个分类网络或者分别训练一个价值估计函数来做出决策。

• Deep RL：现在我们有了深度学习，有了神经网络，那么大家也把这个过程改进成一个端到端训练(end-to-end training) 的过程。你**直接输入这个状态，我们不需要去手工地设计这个特征，就可以让它直接输出动作**。那么就可以用一个神经网络来拟合我们这里的价值函数或策略网络，省去了特征工程 (feature engineering) 的过程。



• 策略函数 (policy function)，agent 会用这个函数来选取下一步的动作。

• 价值函数 (value function)。我们用价值函数来对当前状态进行估价，它就是说你进入现在这个状态，可以对你后面的收益带来多大的影响。当这个价值函数大的时候，说明你进入这个状态越有利。

• 模型 (model)。模型表示了 agent 对这个环境的状态进行了理解，它决定了这个世界是如何进行的。

深度强化学习（Deep Reinforcement Learning）：不需要手工设计特征，仅需要输入 State 让系统直接输出 Action 的一个 end-to-end training 的强化学习方法。通常使用神经网络来拟合 value function或者 policy network。

• policy-based（基于策略的）：Agent 会制定一套动作策略（确定在给定状态下需要采取何种动作），并根据这个策略进行操作。强化学习算法直接对策略进行优化，使制定的策略能够获得最大的奖励。

• valued-based（基于价值的）：Agent 不需要制定显式的策略，它维护一个价值表格或价值函数，并通过这个价值表格或价值函数来选取价值最大的动作。

• model-based（有模型结构）：Agent 通过学习状态的转移来采取措施。

• model-free（无模型结构）：Agent 没有去直接估计状态的转移，也没有得到 Environment 的具体转移变量。它通过学习 value function 和 policy function 进行决策。

基于策略迭代和基于价值迭代的强化学习方法有什么区别?

答：1. 基于策略迭代的强化学习方法，agent 会制定一套动作策略（确定在给定状态下需要采取何种动作），并根据这个策略进行操作。强化学习算法直接对策略进行优化，使制定的策略能够获得最大的奖励；基于价值迭代的强化学习方法，agent 不需要制定显式的策略，它维护一个价值表格或价值

函数，并通过这个价值表格或价值函数来选取价值最大的动作。

2. 基于价值迭代的方法只能应用在不连续的、离散的环境下（如围棋或某些游戏领域），对于行为集合规模庞大、动作连续的场景（如机器人控制领域），其很难学习到较好的结果（此时基于策略迭代的方法能够根据设定的策略来选择连续的动作)；

有模型（model-based）学习和免模型（model-free）学习有什么区别？

答：针对是否需要对真实环境建模，强化学习可以分为有模型学习和免模型学习。

有模型学习是指根据环境中的经验，构建一个虚拟世界，同时在真实环境和虚拟世界中学习；免模型学习是指不对环境进行建模，直接与真实环境进行交互来学习到最优策略。总的来说，有模型学习相比于免模型学习仅仅多出一个步骤，即对真实环境进行建模。免模型学习通常属于数据驱动型方法，需要大量的采样来估计状态、动作及奖励函数，从而优化动作策略。免模型学习的泛化性要优于有模型学习，原因是有模型学习算需要对真实环境进行建模，并且虚拟世界与真实环境之间可能还有差异，这限制了有模型学习算法的泛化性。

DL 中的 loss function 目的是使预测值和真实值之间的差距最小, 而 RL 中的 loss function 是是奖励和的期望最大。

蒙特卡罗与时序差分计算价值函数：

1. 蒙特卡罗用从当前状态到结束状态更新，比如一个任务需要10步完成，第一步有A,B两种策略，如何计算第一步A策略的价值，蒙特卡罗的思想就是（此处暂时不谈折扣系数），采取A策略n（比如说100000次）次，取这n次奖励的均值作为第一步采取A策略的价值。
2. 时序差分是不一定到结束才更新，而是k步更新，比如说是从当前状态到下k步状态更新。