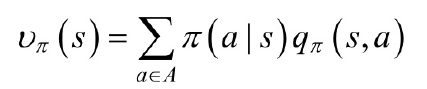
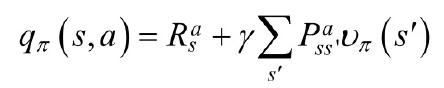
状态值函数和状态行为值函数的关系





reset()使智能体具有初始化的功能;render()扮演图像引擎的角色;step()扮演物理引擎的角色

马尔科夫链

策略迭代算法 P63

值函数迭代算法P64

雅可比迭代法P69

高斯-赛德尔迭代法P70

今天通过推导高斯-赛德尔迭代法再次发现线性代数真是很神奇的东西，矩阵方程的移项直接用逆矩阵就可以了

P65-P68解决最优控制问题的部分没看，数学知识太多所以没看

P71-73 通过压缩映射证明值函数迭代过程具有收敛性，涉及到泛函分析所以没看

原函数与反函数在相同区间内单调性相同

蒙特卡罗探索性初始化P84

同策略和异策略P85

重要性采样P86

蒙特卡罗方法计算积分P90

马尔科夫链蒙特卡罗方法，通过在概率空间中随机采样来接近感兴趣参数的后验分布P92

MCMC算法对步骤有些困惑：为什么小于接受率就可以被接受，接受率的意义是什么

TD的同策略Sarsa方法和异策略Qlearning方法P103

TD(λ)方法及前向观点和后向观点P104

参数化线性值函数逼近 P121

参数化非线性值函数逼近DQN P128

DDQN((应该是有三个神经网络，TD目标动作选择网络参数θ1，TD目标动作评估网络参数θ2，动作值函数逼近网络参数θ3) P130

基于核函数的非参数逼近方法 P135 看个半懂，SVM基本忘光了

基于高斯过程的非参数逼近方法 P140 没看懂，没学过高斯过程

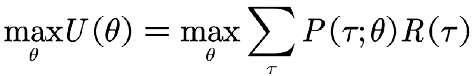
基于策略梯度的强化学习 见下面

基于置信域策略优化的强化学习，改善策略梯度的步长使回报函数单调非减

EM算法：Q步根据已知的模型参数和隐变量的估计构建一个似然函数，M步对似然函数求导得到下一次迭代的模型参数

**7.1节思路：**

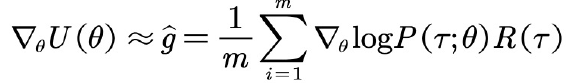
①确定强化学习的目标函数使综合回报最大



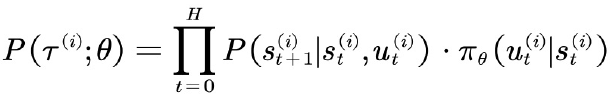
这样就变成了一个最优化问题，采用梯度下降解决，对目标函数求导得到导函数



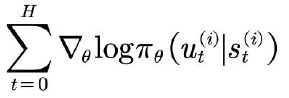
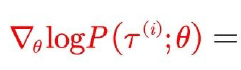
所以最终目标函数的梯度就是红色公式的期望，采用经验平均估算求它的期望



②先看第一项，表示策略Π的概率随参数θ变化的方向

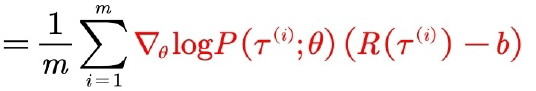


其中P(St+1;St,ut)与θ无关，所以



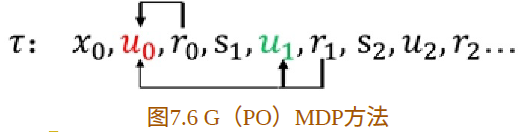
利用重要性采样即可求解(台大网课说好像可以直接用tf对网络参数求微分)

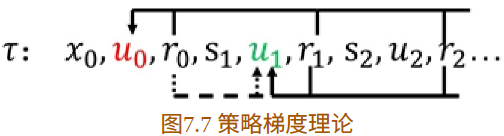
③再看第二项，表示策略Π的回报。但是它的方差很大(台大网课里说的是R都是正的不好，为了避免sample不全面需要减去b使得部分R变成负值)，我们引入常数基线b来减小方差



通过使方差对b求偏导即可求得策略梯度方差最小时的b

然后需要注意的是R代表的是该策略Π对应的回报，但实际上当前的动作与过去的回报是无关的(台大网课说的是整体R好不一定说明回合中每一步的r都好)，所以我们需要修改回报函数R，可采用G(PO)MDP法或策略梯度理论法





至此，基于策略梯度的强化学习理论讲解完毕

**2.4习题：**

1.

Q：马尔可夫过程和马尔可夫决策过程的区别。

A：马尔可夫过程是一个二元组(S,P)，指的是从一个环境S经过概率P到达另一个环境S’这样一个过程，马尔可夫过程不存在动作和奖励。

马尔可夫决策过程是指将动作(策略)和回报考虑在内的马尔可夫过程，由五元组(S,A,P,R,γ)描述

2.

Q：随机决策的理解。

A：随机策略是指策略(动作)的决策具有随机性，不同动作的选择的概率不同。即给定状态S时动作集上的一个概率分布，可以选择不同的概率分布模型。随机策略可以很好的结合exploitation和exploration

3.

Q：安装gym并测试其中的CartPole。

A：done，python里space和tab不能混用

cmd里输入activate gymlab;

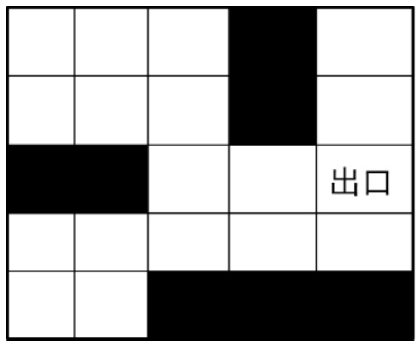
python;import gym;env=gym.make(‘CartPole-v0’);env.reset();env.render()



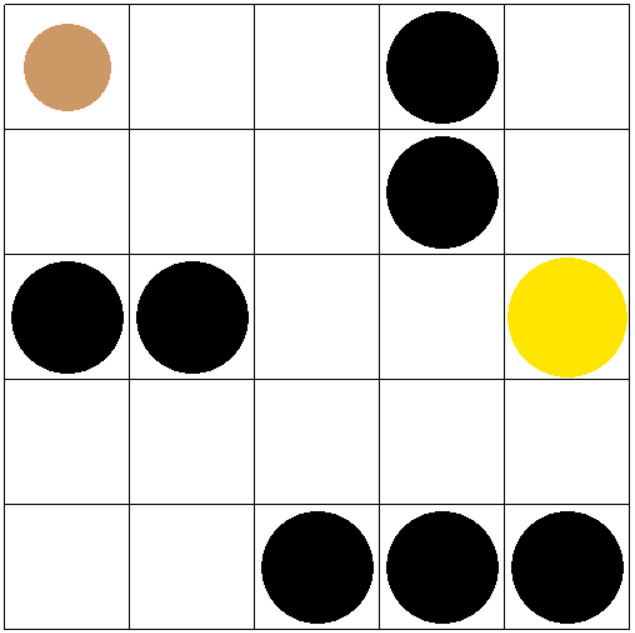
想用哪个环境就去gymlab\Lib\site-packages\gym\envs\\_\_init\_\_.py里找它的id

4.

Q：基于gym构建如下迷宫世界：



A：



根据书里提供的找宝藏环境进行画图组件的更改即可，注意细节不要出现错误

待解决：trap用矩形表示，出口用汉字表示，

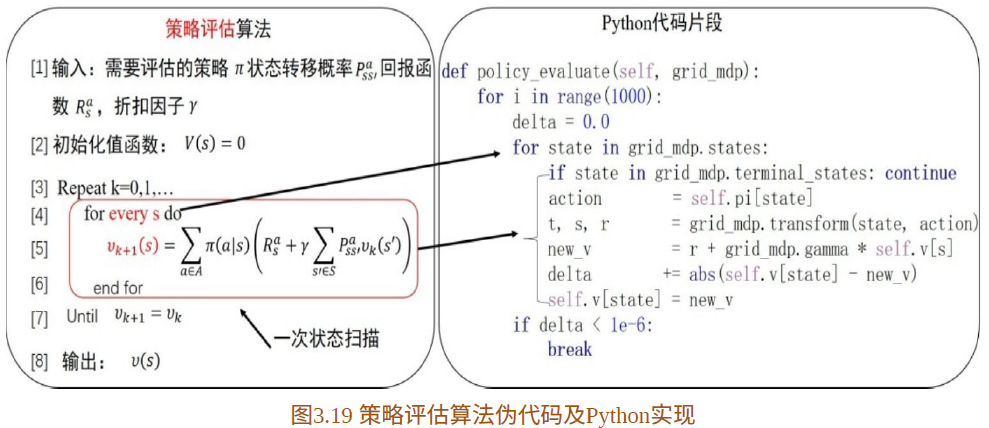
机器人reset初始化的时候会初始到trap里，需要避免这种情况

**3.5习题**

1.

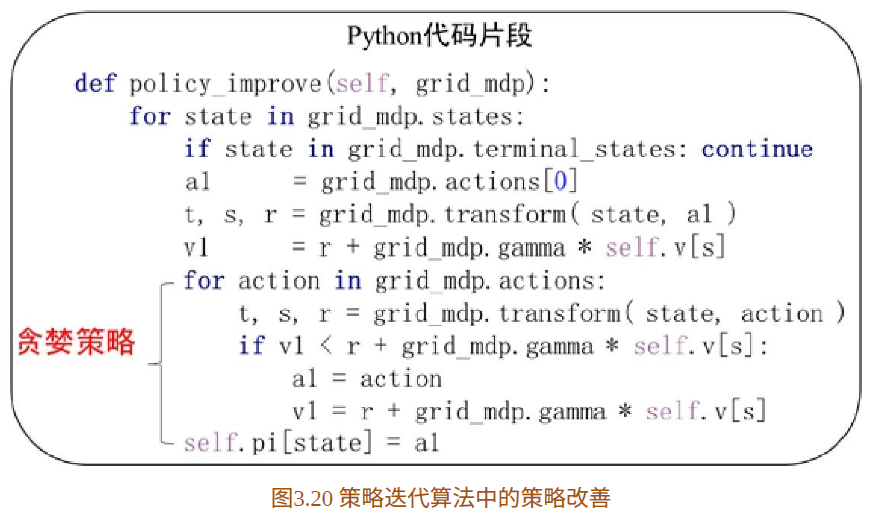
Q：什么是策略迭代算法，什么是值迭代算法，两者的区别和联系是什么？

A：策略迭代算法包括策略评估和策略改善两个步骤。在策略评估中，给定策略，通过数值迭代算法不断计算该策略下每个状态的值函数，该过程的主要目的是得到和更新各状态的值函数。然后在策略改善中，利用策略评估中计算出的值函数和贪婪策略得到新的策略，该过程的主要目的是使策略朝着最优策略逼近。如此循环下去，最终得到最优策略。

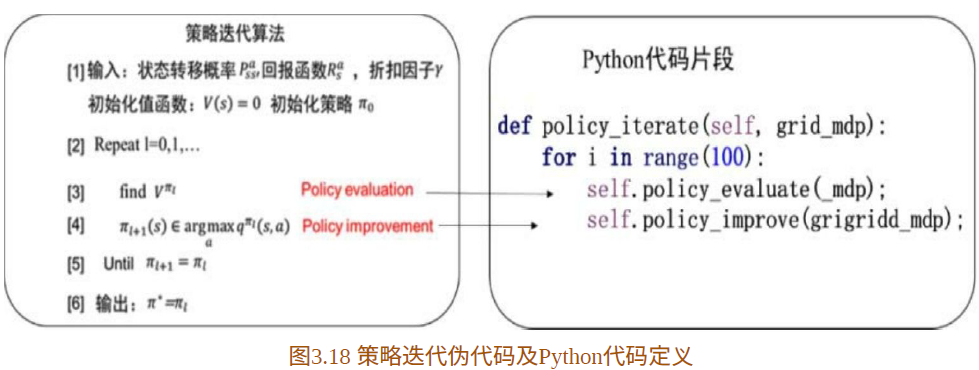


这里action为什么只有一次赋值没看懂，感觉代码和公式不符，没有源代码来考究

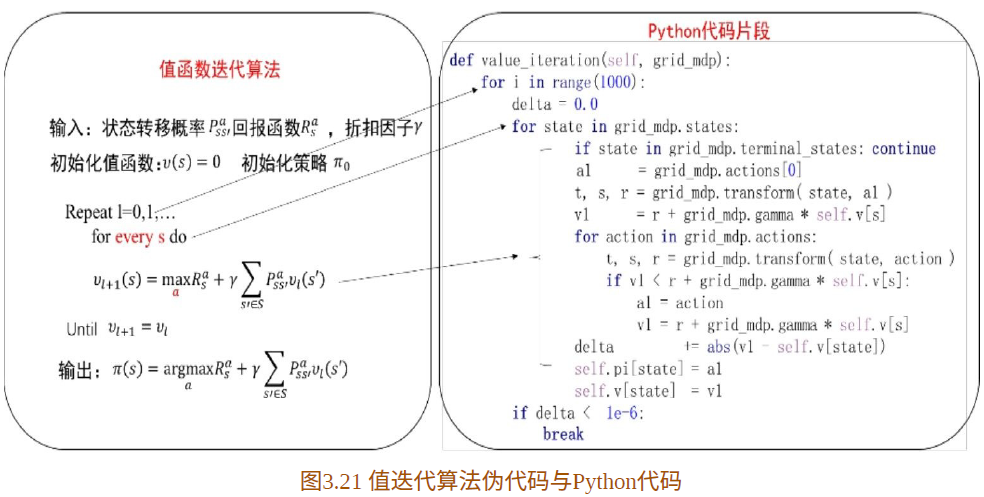
一次策略评估过程会把所有状态的值函数都更新一遍(第二个循环)



策略评估中只有对V的赋值，而策略改善中有对V大小的比较，所以策略改善可以使得策略朝着最优策略的方向逼近



值函数迭代算法是策略迭代算法的进化版，无需等到值函数收敛后再进行策略改善，因此可以在评估完一个状态后就进行该状态的策略改善，这就是值函数迭代算法。



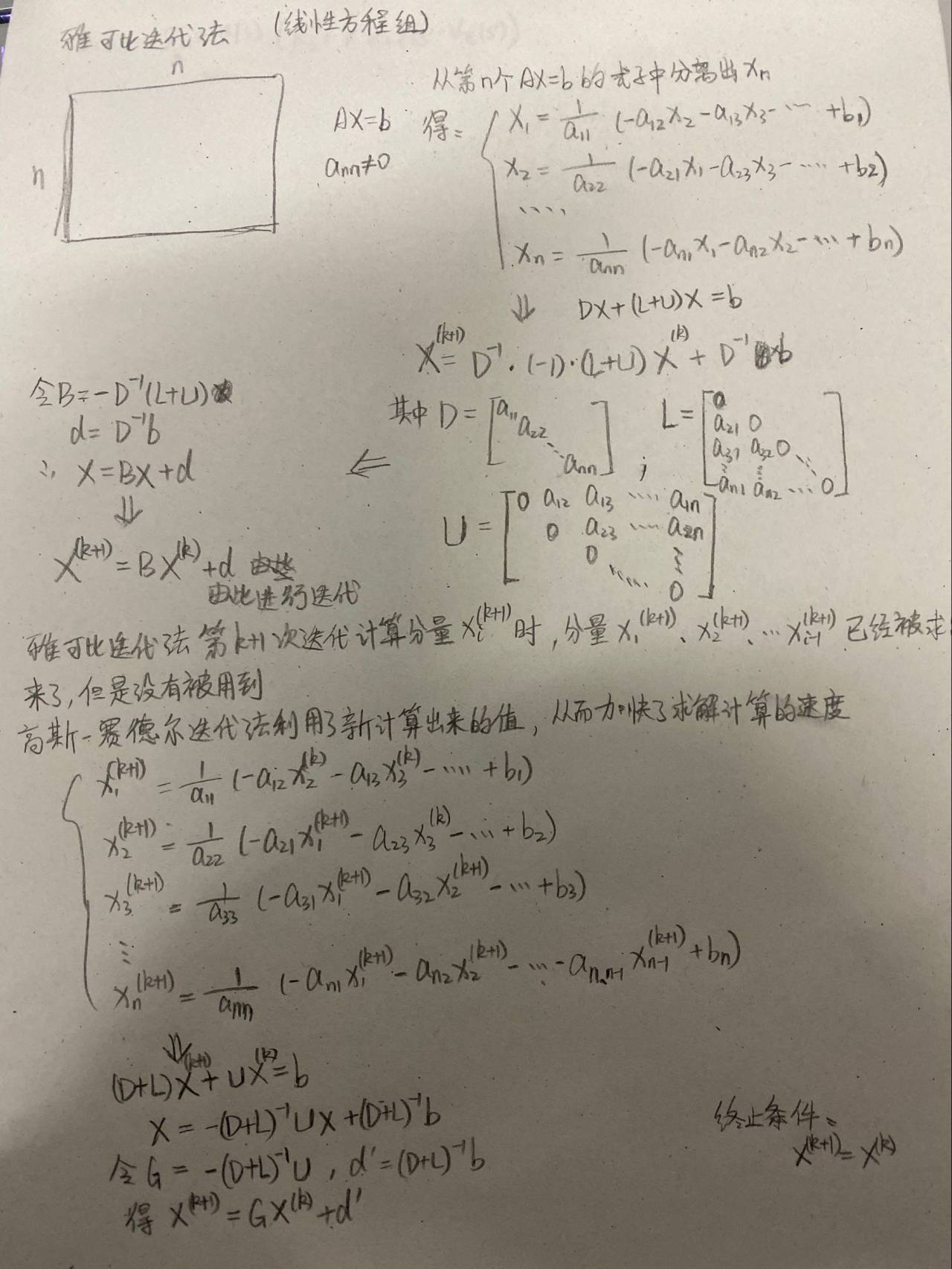
联系：两者都是为了达到最优控制，无论是策略迭代算法还是值迭代算法，最后都能得到最优值函数和最优策略

区别：策略迭代算法是交替进行策略评估和策略改善(在对所有状态值进行评估完后再对所有状态值进行改善)，值迭代算法是在策略评估每进行一步就跟着一次策略改善(每个状态值的更新都会跟着一次该状态值的改善)。这两个算法本质讲的是一回事(代码也差不多)，一个是从策略(动作q)的角度提出的算法，另一个是从状态值(V)的角度提出的算法

2.

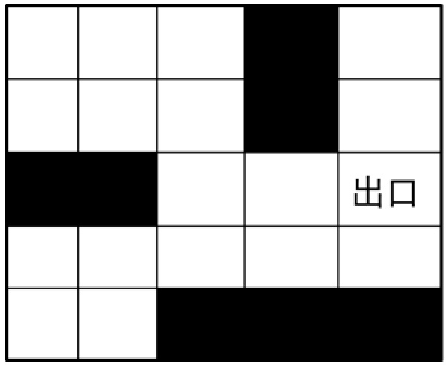
Q：高斯-赛德尔迭代算法和雅可比迭代算法的区别是什么？

A：高斯-赛德尔迭代算法是雅可比迭代算法的进化版，在一次迭代的过程中利用了新计算出来的值，从而加快了求解计算的速度



3.

Q：利用动态规划的方法求解如下迷宫问题：



A：没给demo代码，只给了策略迭代算法和值迭代算法的代码片段，放弃之

两个迭代算法的代码看懂了

4.基于HJB方程分别用数值法和DDP的方法分别求解如下带有非完成约束使得机器人路径最优的规划问题：



A：没学过最优控制，放弃之

**4.4习题**

1.

Q：蒙特卡罗方法可以解决哪些强化学习问题？

A：蒙特卡罗方法可以用于解决无模型的强化学习问题，采用蒙特卡罗的方法(随机采样)来计算状态值函数和行为值函数的期望

2.

Q：什么是同策略(on-policy)，什么是异策略(off-policy)，两者的优缺点各是什么？

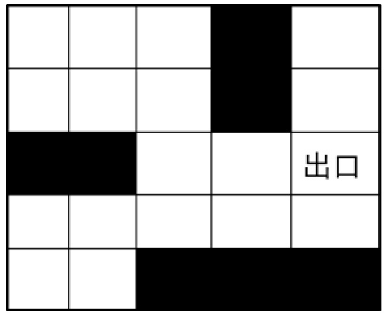
A：若探索策略和评估及改善的策略是同一个策略，我们称之为on-policy

若探索策略和评估及改善的策略是不同的策略，我们称之为off-policy

同策略操作简单、逻辑易于理解；异策略可以保证充分的探索性但必须满足覆盖性条件

3.

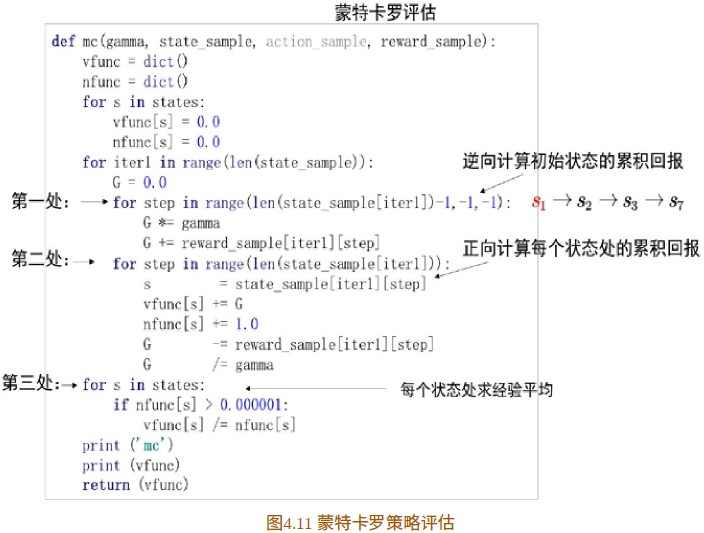
Q：利用蒙特卡罗方法求解下列迷宫问题：



A：没给demo代码，只给了探索策略和策略评估的代码片段，放弃之

两个代码片段看懂了



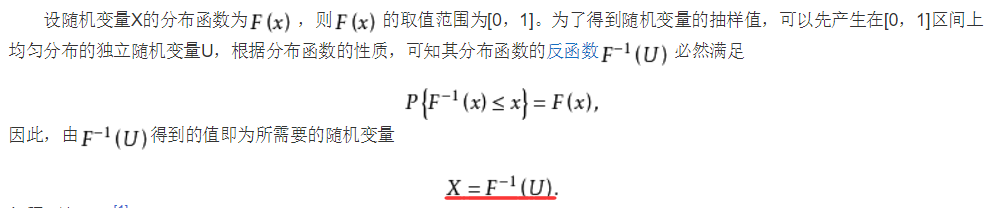


4.

Q：手动编写正态分布的随机样本生成方法。

A：

用反变换法:



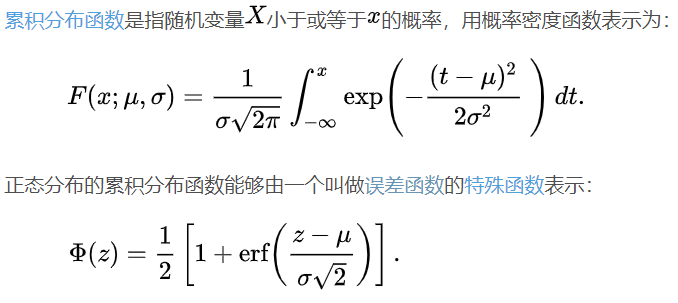
证明：①F(x)=P(F(x)>=u)，F(x)取值范围是0-1，u是0-1上的均匀分布，所以等式成立

②F(x)=P(X<=x)，这是分布函数的定义

③F(x)>=u等价于u<=F(x)等价于F^-1(u)<=x，概率分布函数单调递增，这是反函数的性质

①②③联立：F(x)=P(F(x)>=u)=P(F^-1(u)<=x)；F(x)=P(X<=x) --> 所以X=F^-1(u)

应用于这道题，正态分布的概率分布函数为



因此X=F^-1(u)=μ+sqrt(2)\*σ\*inverf(2u-1)

代码如下：

import numpy as np

from scipy.special import erfinv

def inverfsampling(mu=0, sigma=1, size=1):

z = np.sqrt(2) \* erfinv(2 \* np.random.uniform(size=size) - 1)

return mu + z \* sigma

**5.5习题**

1.

Q：时间差分方法与蒙特卡罗方法、动态规划方法的区别与联系？

A：①动态规划方法是基于模型的方法，蒙特卡罗方法和时间差分方法是无模型的

②三者对于值函数的计算不同，动态规划方法在计算值函数的时候用到了当前状态s的所有后继状态s’处的值函数；蒙特卡罗方法是利用经验平均利用所有回报的累计和估计状态的值函数；时间差分方法结合了蒙特卡罗采样方法和动态规划的利用后继状态s’处的值函数。

③动态规划方法利用模型计算后继状态，时间差分方法利用试验得到后继状态

④蒙特卡罗方法采用值函数的定义，是无偏估计，对初值不敏感；时间差分方法在试验中V(St+1)用的是估计值，属于有偏估计，对初值敏感

2.

Q：如何理解TD(λ)算法的前向视角和后向视角？

A：什么是TD(λ)算法：

在更新当前状态的值函数时，我们既可以用到下一个状态的值函数，也可以用到后继第n个状态的值函数，这样我们就可以得到当前值函数的n种估计方法

具体哪种估计方法更接近真实值我们不知道，最靠谱的方法就是利用加权的方法来融合这n个估计值，所以我们在G(n)前面乘以加权因子(1-λ)λ^(n-1)将其融合，利用融合后的回报G(λ)来更新当前状态的值函数

前向视角即是G(λ)的定义，估计当前状态的值函数时，需要用到将来时刻的值函数。前向观点通过观看将来状态的值函数来估计当前的值函数。因为更新的时候用到了将来时刻的值函数，所以前向观点需要一次试验结束后才能计算。

后向视角是采用增量式的更新方法，在agent进入一个状态后，首先会计算当前的TD偏差，然后向已经经历过的状态喊话，告诉这些状态处的值函数需要利用当前时刻的TD偏差更新

两者异同：

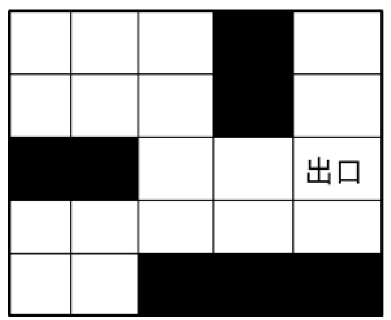
①前向观点需要等到⼀次试验之后再更新当前状态的值函数；后向观点不需要等到值函数结束后再更新值函数，⽽是每⼀步都在更新值函数，是增量式⽅法。

②前向观点在⼀次试验结束后更新值函数时，更新完当前状态的值函数后，此状态的值函数就不再改变。后向观点在每⼀步计算完当前的TD误差后，其他状态的值函数需要利⽤当前状态的TD误差更新。

③在⼀次试验结束后，前向观点和后向观点每个状态的值函数的更新总量是相等的，这个在书里已经证明了

3.

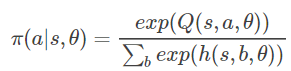
Q：利用Sarsa和Qlearning方法解决下列迷宫问题，并比较他们的差别。



A：todo

4.修改Qlearning中的探索策略，如使用玻尔兹曼探索，解决上述迷宫问题，并试着比较两者的优劣。

玻尔兹曼策略的意义是行为值函数大的动作被选中的概率大



//todo

//解决迷宫问题，比较优劣

**6.4习题**

1.

Q：为什么要引入值函数逼近，它可以解决哪些问题？

A：三四五章介绍的基于动态规划的方法、基于蒙特卡罗的方法和基于时间差分的方法都有一个前提，就是状态空间和动作空间都是离散的且不能太大，因为值函数评估是用一张表格存储的。当空间很大或者为连续空间时，我们就需要利用函数逼近的方法表示值函数。

值函数逼近可以用于解决状态空间和动作空间为连续或很大时，对值函数的评估问题。

2.

Q：试着用DQN方法玩雅达利游戏。

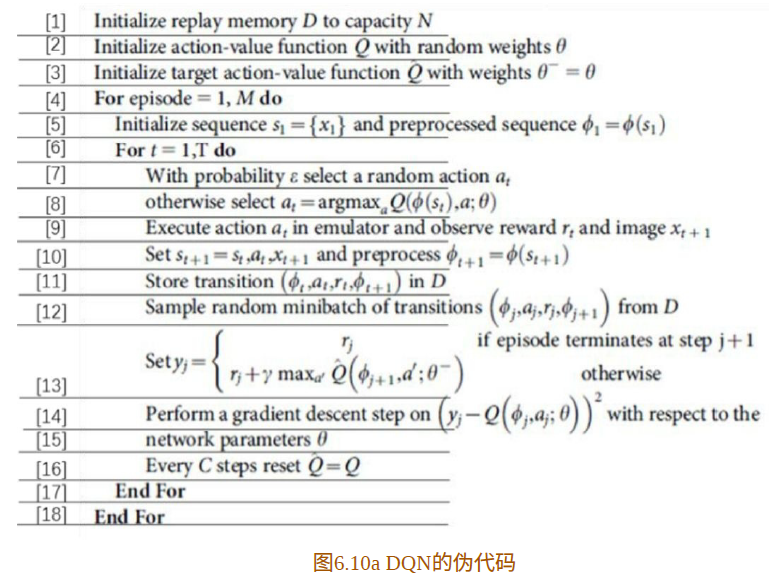
A：雅达利游戏是啥我都不知道

3.

Q：试着比较DQN及其变种的效果？

A：

基本DQN：在Qlearning基础上的改进算法，利用深度卷积神经网络逼近值函数，利用经验回放训练强化学习的学习过程，独立设置目标网络来单独处理时间差分算法中的TD偏差



DDQN：改进了DQN过估计的缺点，因为DQN值函数评估时都是采用的贪婪max策略，这样的话非均匀的过估计会影响最终的策略决策。DDQN将策略评估分成了动作选择和动作评估两部分，分别用不同的值函数(参数更新速度不同)来实现，这样就能有效地缓解该问题。

优先回放：DQN在经验回放的时候采用均匀分布，即每个状态的采样权重都是相同的，但实际上智能体的经验对智能体的学习并非具有同等重要的意义，因此优先回放的思想就是打破均匀采样，赋予学习效率高的状态以更大的采样权重。

DuelingDQN：前面三种DQN在值函数逼近时所用的神经网络都是卷积神经网络，DuelingDQN改变了网络结构，将动作值函数分解为状态值函数和优势函数，分别利用神经网络逼近，以得到最后的动作值函数。

4.

Q：修改神经网络的优化方法并比较效果。

A：优化方法有SGD、BGD、MBGD、Adam等，应该最终优化效果是相似的，但收敛速度不一样。具体的没做。

**7.3习题**

1.

Q：采用直接策略搜索方法的好处？

A：

与值函数方法相比，策略搜索方法的优点：

①直接策略搜索⽅法是对策略 进⾏参数化表⽰，与值函数⽅法中对值函数进⾏参数化表⽰相⽐，策略参数化更简单，有更好的收敛性。

②利⽤值函数⽅法求解最优策略时，策略改善需要求解，当要解决的问题动作空间很⼤或者动作为连续集时，该式⽆法有效求解。

③直接策略搜索⽅法经常采⽤随机策略，因为随机策略可以将探索直接集成到所学习的策略之中。

与值函数⽅法相⽐，策略搜索⽅法的缺点：

①策略搜索的⽅法容易收敛到局部最⼩值；

②评估单个策略时并不充分，⽅差较⼤。

2.

Q：策略梯度理论中有哪些减小策略梯度误差的方法？

A：理论来讲，策略梯度是无偏的

问题①：策略梯度的方差很大。解决方法：引入常数基线减小方差。

问题②：不应该每次都乘以策略的总回报。解决方法：修改回报函数，采用G(PO)MDP方法或者策略梯度理论方法

3.

Q：运用策略梯度理论解决打乒乓球问题。

A：todo

4.

Q：尝试使用OpenAI的其他软件，如baseline，roboschool等。

A：稍稍吧

**8.3习题**

1.

Q：TRPO算法成功的关键是什么？

A：TRPO算法全称是Trust Region Policy Optimization，置信域策略优化，是基于策略梯度方法的改善。策略梯度算法虽然有效，但是在更新步长的时候容易出现问题，当步长不合适会导致agent选择一个不好的策略，利用这个不好的策略进行学习会越学越差。因此合适的步长很重要。

TRPO成功的关键就是找到了合适的步长，将新策略的回报函数改写成旧策略的回报函数加上优势函数，只要保证优势函数非负就可以保证每次迭代时回报函数值单调不减。

2.

Q：TRPO用到的四个技巧是什么？

A：

①处理状态分布。在计算新策略的回报函数时忽略状态分布的变化，依然采用旧策略所对应的状态分布。

②处理动作分布。对于新策略的动作分布不好处理，我们用重要性采样来处理动作分布，将其改为旧策略的动作分布。

③在约束条件中(具体看②)，利用平均KL散度代替最大KL散度。

④将服从ρ的状态分布更替为服从旧策略的状态分布

3.

Q：基于gym手动编写TRPO的代码。

A：todo，应该是直接套公式就可以了，但是我公式还没看懂

4.

Q：利用TRPO的方法让游动机器人和跳跃机器人学会运动。

A：稍稍吧

**9.2习题**

1.

Q：DDPG与DPG的区别，可以解决哪些问题？

A：DDPG和DPG都是确定性策略，但DPG采用参数w和参数θ来逼近行为值函数和确定性策略，而DDPG采用深度神经网络来逼近行为值函数和确定性策略。

同时为了解决深度神经网络数据独立同分布的问题，将DQN算法中使用的经验回放和独立目标网络加入了DPG算法中。

2.

Q：AC算法的网络结构是什么？

A：AC算法是一种确定性策略的学习框架，包括两个元素，一是采用随机策略的行动策略，而是采用确定性策略的评估策略。行动策略采用值函数逼近的方法，对行动策略和评估策略求出参数梯度后进行梯度下降。

3.

Q：利用DDPG的方法解决四足机器人行走问题。

A：感觉代码越来越难了，但是我一直没跟上进度。哎。

4.

Q：借鉴DQN，DDPG有哪些地方可以修改？

A：DQN一共两个创新点，经验回放和独立目标网络，DDPG都用上了。

因此我没想出可以再修改的地方(主要是看得不仔细)。

不过我想可以用优先回放代替经验回放，达到更快的训练效果。

10.3习题

1.

Q：GPS都有哪些算法？

A：GPS是Guided Policy Search，引导策略搜索算法，解决了无模型算法数据效率低的缺点。GPS方法将策略搜索方法分为控制相和监督相。其中控制相通过轨迹最优、传统控制器或随机最优等方法控制产生好的数据；监督相利用从控制相产生的好数据进行监督学习。

从问题的构建来看，GPS经历了基于重要性采样的GPS→基于变分推理的GPS→基于约束的GPS的过程。

从优化方法看，基于约束的GPS的优化，经历了基于对偶梯度下降法→布雷格曼交叉方向乘子法→镜像下降优化算法的过程。

从控制相来看，GPS从基于轨迹最优(微分动态规划DDP、线性二次规划LQR和线性二次高斯LQG)发展到了随机最优控制PI2 GPS。

2.

Q：GPS中动力学拟合方法都有哪些？

A：这题不会做。可能答案出现在10.1中弃掉的那部分？

3.

Q：利用GPS解决倒立摆问题。

A：todo

4.

Q：利用GPS解决机器人路径规划问题。

A：todo