## 问题假设

假设第t周期的成本为ct，策略为，策略为，为折现因子，那么从t周期往后的成本之和可以被定义为：

在t时刻的往后累积支出的期望，可以把它称为第t时刻策略的价值函数，可以写成：

]

从式子可以看出，的值依赖于t+1时刻的状态，如果要得到的最小值来得到最优的策略解，可以根据贝尔曼方程可以，对于任意的，都会有：

其中在t时刻状态为的时候，补货了后，状态转为的不同概率。对于小规模的问题，可以通过这个方程进行动态规划来获得最优的补货策略，但是当数据规模变大的时候，难以直接解出这个方程的最优解，所以会对这个价值函数进行一定的近似操作，在这里使用了PPO的算法。

## 方法介绍

PPO算法是一种actor-critic算法，在整个过程中会维护两个神经网络。第一个神经网络被称为actor，在动态规划的每个阶段里面，在观察到state之后，根据神经网络，输入state，输出近似的最优策略，即第一个神经网络需要得到策略，代表神经网络的参数，是需要迭代优化的；第二个神经网络被称为critic，在决定最优策略之后，预测得到，在每个阶段会通过比较和实际的来对actor中的参数进行更新，即critic的神经网络需要得到价值，代表这个神经网络的参数。其中两个神经网络之间的参数，相互独立。两个神经网络中的网络权重初始化的时候是随机设定的，在文章中分别用和表示，为了得到表现好的，，在迭代这两个参数的时候会用到Adam optimizer（adaptive mini-batch gradient descent）。

## 迭代过程

在第K次迭代当中，假设训练数据集中有m个周期的数据，先使用，按照顺序进行决策，得到一系列的状态、决策、支出，如（），并将这些数据储存在缓冲区当中。一旦缓冲区中的数据量包括了m个周期的所有数据，会对策略进行新的迭代，变为。先根据critic神经网络得到=[,,… ]，然后再计算预测支出，对于时间点i来说，表示在i到t阶段的支出折旧之和以及被看作无限周期的第m期的价值的折旧之和，其中的近似计算公式为：

在得到以上的预测得到的价值函数的预测以及近似的折旧费用之和，我们可以量化这个策略在状态s下的优势，这可以用来表示这个策略在状态下的行动，和预期的支出相比，更好或者更坏的程度，这个优势用来表示。这个优势可以用式子：+来表示。

对于critic来说，他的损失函数是：

Value Loss=

而对于actor来说，它的损失函数分为两个部分，一个部分是用来迭代policy的，即在结束第K周期之后，对actor神经网络中的迭代到中使用的，和传统的PPO算法类似，将actor的损失函数定义为：

Policy Loss=

=/

其中参数的意义是控制的变化范围，避免出现大幅度的变化

## 具体过程

每1000次迭代对方法进行一次测试，例如，通过1000\*300的数据进行测试，即有300个周期，进行1000次神经网络的参数更新，然后再生成10组长度为100000的数据，用神经网络进行策略的选择，计算平均的成本。如果测试中的成本连续30次迭代都没有出现变化，则认为神经网络已经训练到最优。