# 附录-练习题解答

## 第 2 章 练习题

1. 强化学习所解决的问题一定要严格满足马尔可夫性质吗？请举例说明。

答：不一定。例如在围棋游戏场景中，不仅需要考虑当前棋子的位置，还需要考虑棋子的历史位置，因此不满足马尔可夫性质。但依然可以使用强化学习的方法进行求解，例如在AlphaGO论文中使用了蒙特卡洛树搜索算法来解决这个问题。在一些时序性场景中，也可以通过引入记忆单元来解决这个问题，例如在DQN算法中，使用了记忆单元来存储历史状态，从而解决了这个问题，尽管它也不满足马尔可夫性质。

1. 马尔可夫决策过程主要包含哪些要素？

答：马尔可夫决策 主要包含状态空间 、动作空间 、奖励函数 、状态转移矩阵 、折扣因子 等要素，其中状态转移矩阵 是环境的一部分，而其他要素是智能体的一部分。在实际应用中，通常还考虑值函数 和策略函数 等要素，值函数用于某个状态下的长期累积奖励，策略函数用于某个状态下的动作选择。

1. 马尔可夫决策过程与金融科学中的马尔可夫链有什么区别与联系？

答：马尔可夫链是一个随机过程，其下一个状态只依赖于当前状态而不受历史状态的影响，即满足马尔可夫性质。马尔可夫链由状态空间、初始状态分布和状态转移概率矩阵组成。马尔可夫决策过程是一种基于马尔可夫链的决策模型，它包含了状态、行动、转移概率、奖励、值函数和策略等要素。马尔可夫决策过程中的状态和状态转移概率满足马尔可夫性质，但区别在于它还包括了行动、奖励、值函数和策略等要素，用于描述在给定状态下代理如何选择行动以获得最大的长期奖励。

## 第 3 章 练习题

1. 动态规划问题的主要性质有哪些？

答：动态规划主要性质包括最优化原理、无后效性和有重叠子问题，其中无后效性指的是某状态以后的过程不会影响以前的状态，这十分契合马尔可夫性质，因此动态规划问题可以看作是马尔可夫决策过程的一种特殊情况。

1. 状态价值函数和动作价值函数之间的关系是什么？

状态价值函数是所有可能动作的动作价值函数的平均值，也就是说，对于一个状态 ，其状态价值函数 等于所有可能动作 的动作价值函数 的平均值，即 ，其中 表示在状态 下可用的动作数。

1. 策略迭代和价值迭代哪个算法速度会更快？

答：一般情况下，价值迭代算法的收敛速度比策略迭代算法更快。因为价值迭代算法在每次迭代中更新所有状态的价值函数，而策略迭代算法需要在每次迭代中更新策略和状态的价值函数，因此策略迭代算法的计算量比价值迭代算法更大。此外，策略迭代算法的每次迭代都需要进行一次策略评估和一次策略改进，而价值迭代算法只需要进行一次价值迭代，因此策略迭代算法的迭代次数通常比价值迭代算法多。

## 第 4 章 练习题

1. 有模型与免模型算法的区别？举一些相关的算法？

答：有模型算法在学习过程中使用环境模型，即环境的转移函数和奖励函数，来推断出最优策略。这种算法会先学习环境模型，然后使用模型来生成策略。因此，有模型算法需要对环境进行建模，需要先了解环境的转移函数和奖励函数，例如动态规划等算法。免模型算法不需要环境模型，而是直接通过试错来学习最优策略。这种算法会通过与环境的交互来学习策略，不需要先了解环境的转移函数和奖励函数。免模型算法可以直接从经验中学习，因此更加灵活，例如 Q-learning 、Sarsa 等算法。

1. 举例说明预测与控制的区别与联系。

答：**区别**：预测任务主要是关注如何预测当前状态或动作的价值或概率分布等信息，而不涉及选择动作的问题；控制任务则是在预测的基础上，通过选择合适的动作来最大化累计奖励，即学习一个最优的策略。**联系**：预测任务是控制任务的基础，因为在控制任务中需要对当前状态或动作进行预测才能选择最优的动作；控制任务中的策略通常是根据预测任务中获得的状态或动作价值函数来得到的，因此预测任务对于学习最优策略是至关重要的。以赌博机问题为例，预测任务是估计每个赌博机的期望奖励（即价值函数），控制任务是选择最优的赌博机来最大化累计奖励。在预测任务中，我们可以使用多种算法来估计每个赌博机的期望奖励，如蒙特卡罗方法、时间差分方法等。在控制任务中，我们可以使用贪心策略或ε-贪心策略来选择赌博机，这些策略通常是根据预测任务中得到的每个赌博机的价值函数来确定的。因此，预测任务对于控制任务的实现至关重要。

1. 蒙特卡洛方法和时序差分方法的优劣势。

答：**蒙特卡洛方法优势**：可以直接从经验中学习，不需要环境的转移概率；收敛性良好，可以保证在有限步内收敛到最优策略；可以处理长期回报，对于无折扣情况也可以使用。**蒙特卡洛方法劣势**：需要等到一条完整的轨迹结束才能更新价值函数，因此效率较低；对于连续状态空间和动作空间的问题，蒙特卡洛方法难以处理。**时序差分优势**：可以在交互的过程中逐步更新价值函数，效率较高；可以处理连续状态空间和动作空间的问题；可以结合函数逼近方法使用，对于高维状态空间的问题有很好的应用。**时序差分劣势**：更新过程中存在一定的方差，可能会影响收敛速度和稳定性；对于无折扣情况，需要采取一些特殊的方法来保证收敛。总的来说，蒙特卡洛方法对于离散状态空间的问题，特别是存在长期回报的问题有很好的适用性，但是其效率较低。时序差分方法则可以高效地处理连续状态空间和动作空间的问题，但是其更新过程中存在方差问题。在实际应用中需要根据问题的特点和实际情况选择合适的方法。

## 第 5 章 练习题

1. 什么是 $Q$ 值的过估计？有什么缓解的方法吗？

答：值的过估计（overestimation of values）是指在强化学习中，由于采样数据的不充分或者算法本身的限制，导致学习到的状态或动作价值函数高估了它们的真实值。值的过估计会影响强化学习算法的性能和稳定性，因此需要采取相应的缓解措施。一些缓解值的过估计的方法有：双重Q学习（Double Q-learning）：将一个Q函数的更新过程分为两步，分别用来更新行动值函数和目标值，从而避免了Q函数的过估计；优先经验回放：在经验回放中，根据每条经验的TD误差大小来选择回放的概率，使得 TD误差大的经验更有可能被回放，从而更好地修正价值函数；目标网络：使用一个目标网络来计算目标值，目标网络的参数较稳定，不会随着每次更新而改变，从而减缓了价值函数的过估计问题；随机探索策略：采用一些随机的探索策略，如 -greedy策略、高斯噪声等，可以使得智能体更多地探索未知的状态和动作，从而减少了价值函数的过估计问题。这些方法可以在不同的强化学习算法中使用，比如DQN、DDQN、Dueling DQN等。选择合适的方法可以有效地缓解价值函数的过估计问题。

1. on-policy与off-policy之间的区别是什么？

答：on-policy指的是学习一个策略时，使用同一策略来收集样本，并且利用这些样本来更新该策略。即学习的策略和探索的策略是相同的。off-policy指的是学习一个策略时，使用不同于目标策略的行为策略来收集样本，并且利用这些样本来更新目标策略。即学习的策略和探索的策略是不同的。在强化学习中，通常使用Q-learning、SARSA等算法来实现off-policy学习。而使用 policy gradient等算法来实现 on-policy学习。 on-policy 学习的优点是可以较好地处理连续动作空间的问题，并且可以保证学习到的策略收敛到最优策略。但是其缺点是样本的利用效率较低，因为样本只能用于更新当前策略，不能用于更新其他策略。off-policy学习的优点是样本的利用效率较高，因为可以使用不同的行为策略来收集样本，并且利用这些样本来更新目标策略。但是其缺点是可能会出现样本不一致的问题，即目标策略和行为策略不同，会导致学习的不稳定性。因此，在实际应用中需要根据具体问题的特点和实际情况选择合适的学习方式。

1. 为什么需要探索策略？

答：探索策略是强化学习中非常重要的一个概念，原因有：强化学习的目标是学习一个最优策略，但初始时我们并不知道最优策略，因此需要通过探索来发现更优的策略；在强化学习中，往往存在许多未知的状态和动作，如果智能体只采用已知的策略，那么它将无法探索到未知状态和动作，从而可能会错过更优的策略；探索策略可以帮助智能体避免陷入局部最优解，从而更有可能找到全局最优解。探索策略可以提高智能体的鲁棒性，使其对环境的变化更加适应。常用的探索策略包括-greedy策略、softmax策略、高斯噪声等。

## 第 6章 练习题

1. 逻辑回归与神经网络之间有什么联系？

答：**相同点**：逻辑回归和神经网络都属于监督学习算法，都可以用于分类问题。此外，神经网络的某些特殊形式也可以看作是逻辑回归的一种。**区别**：逻辑回归是一种线性分类器，它通过对特征进行加权求和并经过一个sigmoid函数来得到样本属于某个类别的概率。而神经网络则是一种非线性分类器，它通过多个神经元的组合来实现对样本进行分类。**适用范围**：逻辑回归适用于特征空间较简单的分类问题，而神经网络适用于特征空间较复杂的分类问题。**训练方式**：逻辑回归的参数可以通过最大似然估计或梯度下降等方法进行训练，而神经网络的参数通常通过反向传播算法进行训练。**模型复杂度**：相对于逻辑回归，神经网络的模型复杂度更高，可以通过增加神经元的数量、层数等方式来提高模型的性能。

1. 全连接网络、卷积神经网络、循环神经网络分别适用于什么场景？

答：全连接网络是一种最基本的神经网络结构，每个神经元都与上一层的所有神经元相连。全连接网络适合于输入数据维度较低、数据量较小的场景，例如手写数字识别等。卷积神经网络是一种专门用于处理图像等二维数据的神经网络结构，其核心是卷积层和池化层。卷积神经网络适合于图像、语音等二维或多维数据的处理，可以有效地利用数据的局部特征，例如图像分类、目标检测等。循环神经网络是一种处理序列数据的神经网络结构，其核心是循环层，可以捕捉时序数据中的长期依赖关系。循环神经网络适合于序列数据的建模，例如自然语言处理、音乐生成等。需要注意的是，三种神经网络结构并不是相互独立的，它们可以灵活地组合使用，例如可以在卷积神经网络中嵌入循环神经网络来处理视频数据等。在实际应用中需要根据具体的问题特点和数据情况来选择合适的神经网络结构。

1. 循环神经网络在反向传播时会比全连接网络慢吗？为什么？

答：循环神经网络在反向传播时相比于全连接网络会更慢，原因主要有：**循环依赖**：循环神经网络存在时间上的依赖关系，即当前时刻的隐藏状态依赖于上一时刻的隐藏状态。这种循环依赖会导致反向传播时梯度的计算变得复杂，需要使用反向传播算法中的BPTT（Backpropagation Through Time）算法来进行计算，计算量较大，因此速度相对较慢；**长期依赖**：循环神经网络在处理长序列时，会出现梯度消失或梯度爆炸的问题，这是由于反向传播时梯度在时间上反复相乘或相加导致的。为了解决这个问题，需要采用一些技巧，如LSTM和GRU等。相比之下，全连接网络不存在循环依赖关系，因此反向传播时梯度的计算较为简单，计算量相对较小，速度相对较快。需要注意的是，循环神经网络在处理序列数据方面具有独特的优势，它可以处理变长的序列数据，可以捕捉到序列中的长期依赖关系，因此在序列建模等方面被广泛应用。

## 第 7 章 练习题

1. 相比于Q-learning算法，DQN算法做了哪些改进？

答：主要包括：**引入深度神经网络**：Q-learning算法中使用的是表格法来存储动作价值函数，但对于状态空间较大的问题，表格法会变得不可行。DQN通过引入深度神经网络来近似动作价值函数，能够处理高维连续状态空间的问题。**经验回放**：传统的Q-learning算法每次更新时只使用当前状态和动作的信息，但这种方式可能会导致样本之间的相关性和不稳定性。DQN采用经验回放机制，将所有的状态、动作、奖励、下一状态组成的经验存储在经验池中，然后从经验池中随机取样进行训练，可以缓解样本相关性和不稳定性的问题。**目标网络**：DQN还引入了目标网络来解决动作价值函数的不稳定性问题。目标网络是一个与当前神经网络结构相同的网络，但其参数被固定一段时间。在训练时，使用目标网络来计算目标Q值，从而减少当前神经网络参数对目标Q值的影响，提高训练稳定性。**奖励裁剪**：在某些情况下，奖励值可能非常大或非常小，这可能会导致训练不稳定。DQN采用奖励裁剪，将奖励值限制在一个较小的范围内，从而在一定程度上提高了训练稳定性。

1. 为什么要在DQN算法中引入-greedy策略？

答：目的是为了平衡探索和利用的关系。具体来说，-greedy策略会以一定的概率 随机选择动作，以一定的概率 1- 选择当前状态下具有最大 Q 值的动作，从而在训练过程中保证一定的探索性，使得智能体能够尝试一些未知的状态和动作，从而获得更多的奖励。如果在训练过程中完全按照当前状态下的最大Q值选择动作，可能会导致智能体过于保守，无法获得更多的奖励。而如果完全随机选择动作，可能会导致智能体无法学习到更优的策略，从而影响学习效果。因此，引入 -greedy策略可以在探索和利用之间进行平衡，从而在训练过程中获得更好的性能。需要注意的是，-greedy策略中的 值是一个重要的超参数，需要根据具体问题进行调整。如果 值过小，可能会导致智能体无法充分探索环境；如果 值过大，可能会导致智能体无法有效地利用已有的经验。因此，需要根据具体问题进行调参。

1. DQN算法为什么要多加一个目标网络？

答：目标网络的作用是为了解决动作价值函数的不稳定性问题。目标网络是一个与当前神经网络结构相同的网络，但其参数被固定一段时间。在训练时，使用目标网络来计算目标Q值，从而减少当前神经网络参数对目标Q值的影响，提高训练稳定性。具体来说，当使用当前神经网络来计算目标Q值时，当前神经网络的参数和目标Q值的计算都是基于同一批数据的，这可能导致训练过程中出现不稳定的情况。而使用目标网络来计算目标Q值时，目标网络的参数是固定的，不会受到当前神经网络的训练过程的影响，因此可以提高训练的稳定性。同时，目标网络的更新也是基于一定的规则进行的。在每个训练步骤中，目标网络的参数被更新为当前网络的参数的加权平均值，其权重由一个超参数控制。通过这种方式，目标网络的更新过程可以更加平稳，避免了训练过程中出现剧烈的波动，从而提高了训练的效率和稳定性。因此，引入目标网络是DQN算法的一个重要改进，可以显著提高算法的性能和稳定性。

1. 经验回放的作用是什么？

答：经验回放主要作用在于缓解样本相关性和不稳定性问题，提高算法的训练效率和稳定性。**缓解样本相关性问题**：在深度强化学习中，每个样本通常都是与前几个样本高度相关的。如果直接使用当前样本进行训练，可能会导致样本之间的相关性过高，从而影响算法的训练效果。经验回放机制通过从经验池中随机取样，可以打破样本之间的相关性，提高训练的效果。**缓解不稳定性问题**：在深度强化学习中，每个样本的值函数都是基于当前神经网络的参数计算的。由于神经网络的参数在每个训练步骤中都会发生变化，因此每个样本的值函数也会随之变化。这可能会导致算法的训练过程不稳定，经验回放机制可以通过随机取样的方式，减少每个训练步骤中样本值函数的变化，从而提高训练的稳定性。

## 第 8 章 练习题

1. DQN算法为什么会产生Q值的过估计问题？

答：原因主要有：**数据相关性**：每次更新神经网络时，使用的都是之前采集到的数据，这些数据之间存在相关性。这导致神经网络的训练过程不稳定，可能会导致Q值的过估计问题。**最大化操作**： DQN算法在更新目标Q值时，使用的是当前神经网络在下一个状态下具有最大Q值的动作。这种最大化操作可能会导致某些状态和动作的Q值被过估计。为了解决这个问题，可以采用一些技术，如Double DQN和Dueling DQN。Double DQN通过使用一个神经网络来估计当前状态下各个动作的Q值，使用另一个神经网络来计算目标Q值，从而减少Q值的过估计问题。Dueling DQN则通过将Q值分解为状态值和优势值两部分，从而更准确地估计Q值，减少Q值的过估计问题。这些技术可以有效地减少Q值的过估计问题，提高DQN算法的性能。

1. 同样是提高探索，Noisy DQN和-greedy策略有什么区别？

答： -greedy策略是一种基于概率的探索策略，其思想是在每个时间步中，以概率 选择一个随机动作，以概率 1- 选择当前状态下具有最大Q值的动作。当随机动作被选择时，智能体有一定的概率探索新的状态和动作，从而提高探索能力。-greedy策略的优点是简单易用，但可能存在随机性过高或过低的问题，影响探索效果。Noisy DQN是一种基于网络权重噪声的探索策略，其思想是在神经网络中添加一定的权重噪声，以增加探索的随机性。在每个时间步中，神经网络中的权重噪声会随机地改变神经元的输出，从而改变智能体选择动作的概率分布。Noisy DQN的优点是能够自适应地控制探索随机性的大小，从而更加有效地提高探索能力。

## 第 9 章 练习题

1. 基于价值和基于策略的算法各有什么优缺点？

答：前者的优点有：**简单易用**：通常只需要学习一个值函数，往往收敛性也会更好。保守更新：更新策略通常是隐式的，通过更新价值函数来间接地改变策略，这使得学习可能更加稳定。缺点有：**受限于离散动作**；**可能存在多个等价最优策略**：当存在多个等效的最优策略时，基于价值的方法可能会在它们之间不停地切换。后者的优点有：直接优化策略：由于这些算法直接操作在策略上，所以它们可能更容易找到更好的策略；适用于连续动作空间；**更高效的探索**：通过调整策略的随机性，基于策略的方法可能会有更高效的探索策略。缺点有：**高方差**：策略更新可能会带来高方差，这可能导致需要更多的样本来学习。**可能会收敛到局部最优**：基于策略的方法可能会收敛到策略的局部最优，而不是全局最优，且收敛较缓慢。在实践中，还存在结合了基于价值和基于策略方法的算法，即Actor-Critic算法，试图结合两者的优点来克服各自的缺点。选择哪种方法通常取决于具体的应用和其特点。

1. 马尔可夫平稳分布需要满足什么条件？

答：**状态连通性**：从任何一个状态可以在有限的步数内到达另一个状态；**非周期性**：由于马尔可夫链需要收敛，那么就一定不能是周期性的。

1. REINFORCE算法会比Q-learning算法训练速度更快吗？为什么？

答：两者的速度不能一概而论，尽管前者往往会比后者慢。主要考虑几个因素：**样本效率**：因为 REINFORCE算法是一个无偏的估计，但其方差可能很高，这意味着为了得到一个稳定和准确的策略更新，它可能需要与环境交互更多的样本，如果与环境交互的成本很高，REINFORCE算法将会显得更加劣势。**稳定性与收敛**：Q-learning和其他基于值的方法，特别是当与深度神经网络结合时，可能会遇到训练不稳定的问题。这可能会影响其训练速度。

1. 确定性策略与随机性策略的区别？

答：对于同一个状态，确定性策略会给出一个明确的、固定的动作，随机性策略则会为每一个可能的动作（legal action）提供一个概率分布。前者在训练中往往需要额外的探索策略，后者则只需要调整动作概率。但前者相对更容易优化，因为不需要考虑所有可能的动作，但也容易受到噪声的影响。后者则相对更加鲁棒，适用面更广，因为很多的实际问题中，我们往往无法得到一个确定的最优策略，而只能得到一个概率分布，尤其是在博弈场景中

## 第 10 章 练习题

1. 相比于REINFORCE算法，A2C主要的改进点在哪里，为什么能提高速度？

答：改进点主要有：**优势估计**：可以更好地区分好的动作和坏的动作，同时减小优化中的方差，从而提高了梯度的精确性，使得策略更新更有效率；**使用Critic**：REINFORCE通常只使用Actor网络，没有Critic来辅助估计动作的价值，效率更低；**并行化**：即 A3C ，允许在不同的环境中并行运行多个 Agent，每个Agent收集数据并进行策略更新，这样训练速度也会更快。

1. A2C算法是on-policy的吗？为什么？

答： A2C算法在原理上是一个on-policy算法，首先它使用当前策略的样本数据来更新策略，然后它的优势估计也依赖于当前策略的动作价值估计，并且使用的也是策略梯度方法进行更新，因此是 on-policy的。但它可以被扩展为支持off-policy学习，比如引入经验回放，但注意这可能需要更多的调整，以确保算法的稳定性和性能。

## 第 11 章 练习题

1. DDPG算法是off-policy算法吗？为什么？

答：跟DQN一样，DDPG算法，主要结合了经验回放、目标网络和确定性策略，是典型的off-policy算法。

1. 软更新相比于硬更新的好处是什么？为什么不是所有的算法都用软更新？

答：好处：**平滑目标更新**：软更新通过逐渐调整目标网络的参数，使其向主网络的参数靠近，而不是直接复制主网络的参数。这样做可以降低目标的变化幅度，减少了训练中的不稳定性；**降低方差**；**避免振荡**：软更新可以减少目标网络和主网络之间的振荡，这有助于更稳定地收敛到良好的策略。缺点：**速度**：软更新会使目标网络变得更加缓慢；**探索和稳定性权衡**：一些算法可能更倾向于使用硬更新，因为它们可能需要更频繁地探索新的策略，而不依赖于过去的经验。硬更新允许在每次更新时完全用新策略替代旧策略；**算法需求**：某些算法可能对硬更新更敏感，而且硬更新可能是这些算法的关键组成部分。综上，软更新和硬更新都有其用途，选择哪种方式取决于具体的问题和算法要求。

1. 相比于DDPG算法，TD3算法做了哪些改进？请简要归纳。

答：**双Q网络**：TD3使用了两个独立的Q网络，分别用于估计动作的价值。这两个Q网络有不同的参数，这有助于减少估计误差，并提高了训练的稳定性；**目标策略噪声**：与DDPG不同，TD3将噪声添加到目标策略，而不是主策略。这有助于减小动作值的过估计误差；**目标策略平滑化**：TD3使用目标策略平滑化技术，通过对目标策略的参数进行软更新来减小目标策略的变化幅度。这有助于提高稳定性和训练的收敛性。**延迟策略更新**：TD3引入了延迟策略更新，意味着每隔一定数量的时间步才更新主策略网络。这可以减小策略更新的频率，有助于减少过度优化的风险，提高稳定性。

1. TD3算法中Critic的更新频率一般要比Actor是更快还是更慢？为什么？

答： Critic网络的更新频率要比Actor网络更快，即延迟策略更新。延迟策略更新的目的是减小策略更新的频率，以避免过度优化和提高训练的稳定性。因为Critic网络的更新频率更高，它可以更快地适应环境的变化，提供更准确的动作价值估计，从而帮助{Actor网络生成更好的策略。

## 第 12 章 练习题

1. 为什么DQN和DDPG算法不使用重要性采样技巧呢？

答：DQN和DDPG是off-policy算法，它们通常不需要重要性采样来处理不同策略下的采样数据。相反，它们使用目标网络和优势估计等技巧来提高训练的稳定性和性能。

1. PPO算法原理上是on-policy的，但它可以是off-policy的吗，或者说可以用经验回放来提高训练速度吗?为什么？（提示：是可以的，但条件比较严格）

跟A2C一样，可以将经验回放与PPO结合，创建一个PPO with Experience Replay (PPO-ER)算法。在PPO-ER中，智能体使用经验回放缓冲区中的数据来训练策略网络，这样可以提高训练效率和稳定性。这种方法通常需要调整PPO的损失函数和采样策略，以适应off-policy训练的要求，需要谨慎调整。

1. PPO算法更新过程中在将轨迹样本切分个多个小批量的时候，可以将这些样本顺序打乱吗？为什么？

答：将轨迹样本切分成多个小批量时，通常是可以将这些样本顺序打乱的，这个过程通常称为样本随机化（sample shuffling），这样做的好处有降低样本相关性、减小过拟合风险以及增加训练多样性（更全面地提高探索空间）。

1. 为什么说重要性采样是一种特殊的蒙特卡洛采样？

答：原因有：**估计期望值**：蒙特卡洛方法的核心目标之一是估计一个随机变量的期望值。蒙特卡洛采样通过从分布中生成大量的样本，并求取这些样本的平均值来估计期望值。重要性采样也是通过从一个分布中生成样本，但不是均匀地生成样本，而是按照另一个分布的权重生成样本，然后使用这些带权重的样本来估计期望值。**改进采样效率**：重要性采样的主要目的是改进采样效率。当我们有一个难以从中采样的分布时，可以使用重要性采样来重新调整样本的权重，以使估计更准确。这类似于在蒙特卡洛采样中调整样本大小以提高估计的精确性。**权重分布**：在重要性采样中，我们引入了一个额外的权重分布，用于指导采样过程。这个权重分布决定了每个样本的相对贡献，以确保估计是无偏的。在蒙特卡洛采样中，权重通常是均匀分布，而在重要性采样中，权重由分布的比率（要估计的分布和采样分布之间的比例）决定。