RL Experiment4 Rainbow

# 实验内容

1. 实现基本的DNQ
2. 在DNQ的基础上，增加5种改进方法

**double\_q**的做法，是改进损失函数，防止估计偏高

**priority**的做法，是为了改进经验池采样，增进学习速度

**noise\_net**做法，是为了改进探索，由网络中添加噪声，代替epsilon—greedy策略

**dueling\_net**做法，是为了改进更新参数的模式，让没有采样的动作的价值函数也能得到更新，改变了网络架构

**distribute\_q**的做法，是为了改进q网络输出的合理性和真实性，能求出q每个动作的概率分布，进而用每个动作（看成随机变量）的均值当作Q的价值

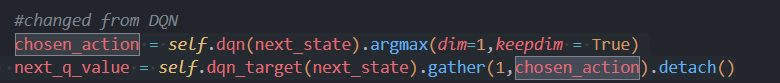
1. 在DNQ的基础上，将这些方法全部用上，就是rainbow。

有三种方法改变网络结构，一种方法改进损失函数，一种方法改进采样，互不冲突。

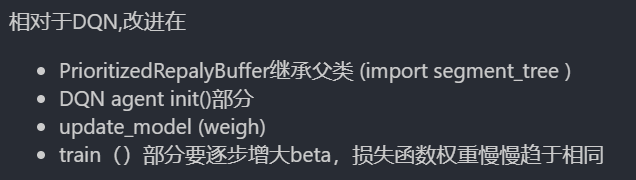
# 代码实现

## Double\_DQN

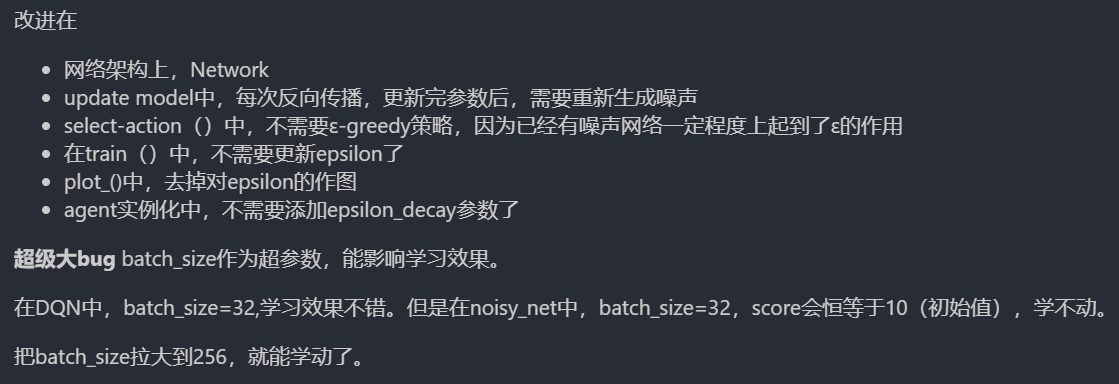
doubleDQN的改进在于compute\_loss（）中，对target\_q的选取，



## Priority\_Buffer

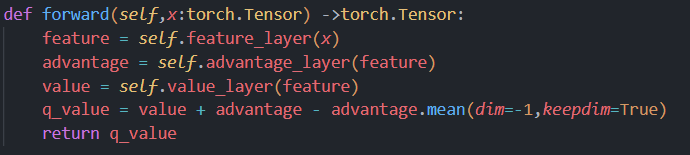


## Noise\_Net



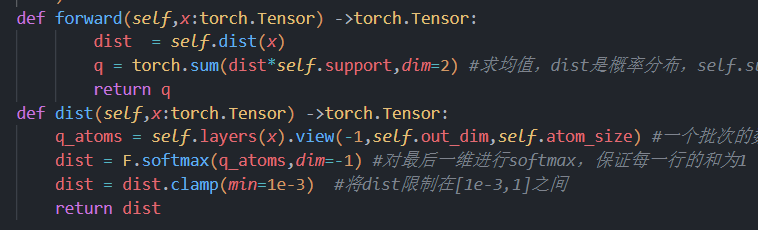
## Dueling\_Net

只需要改变网络架构

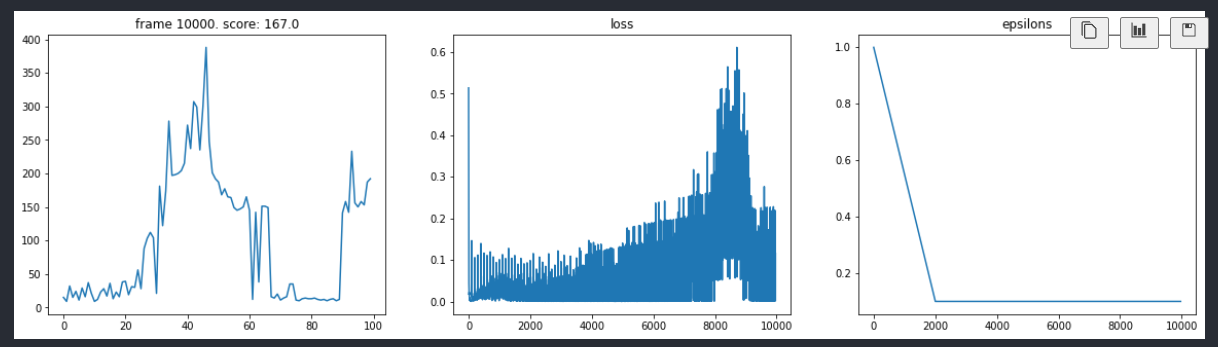
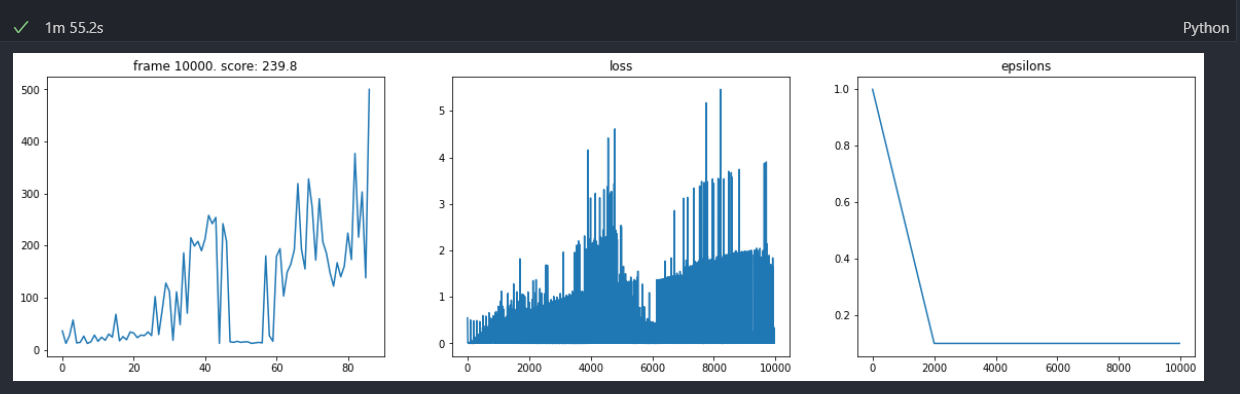


## Distribute\_Net

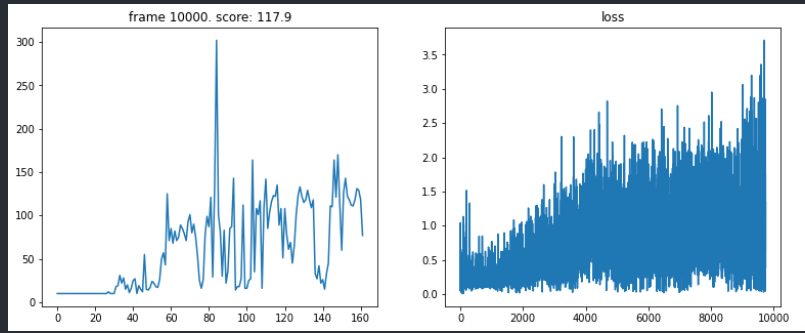
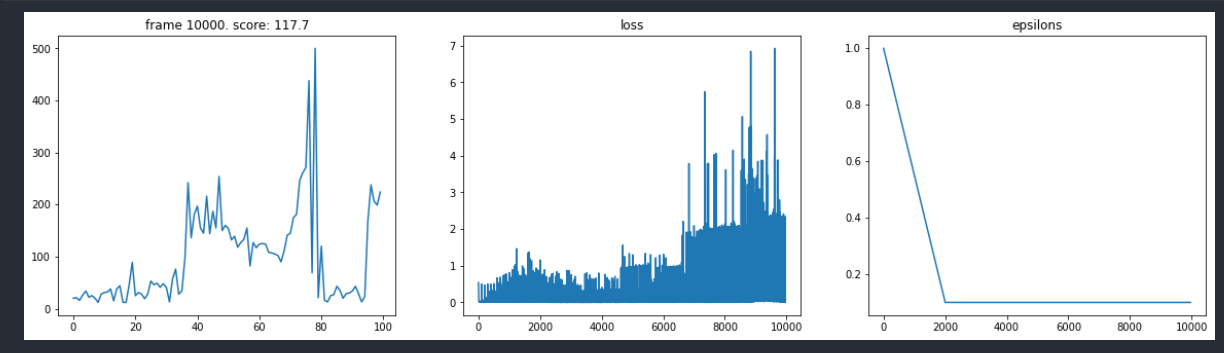
只需要改变网络架构



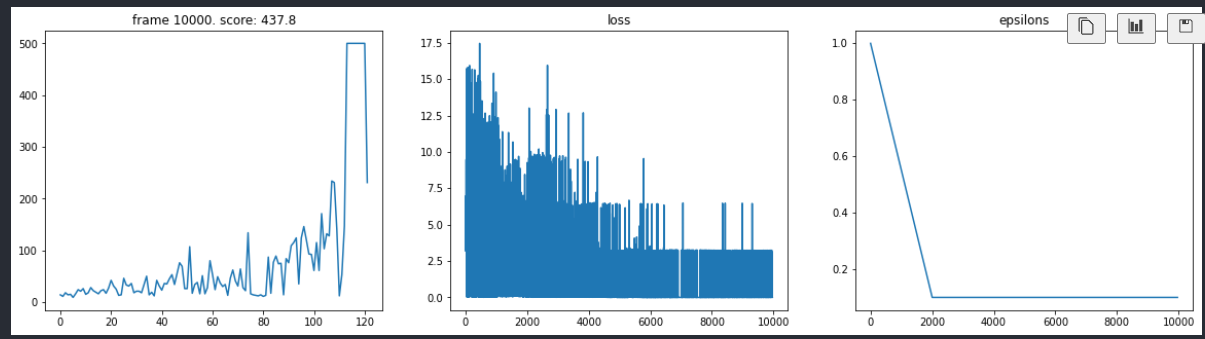
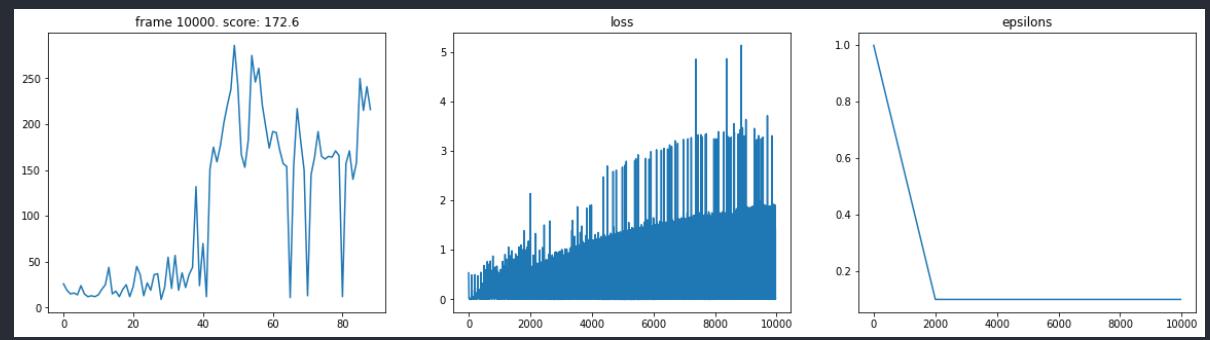
# 实验结果



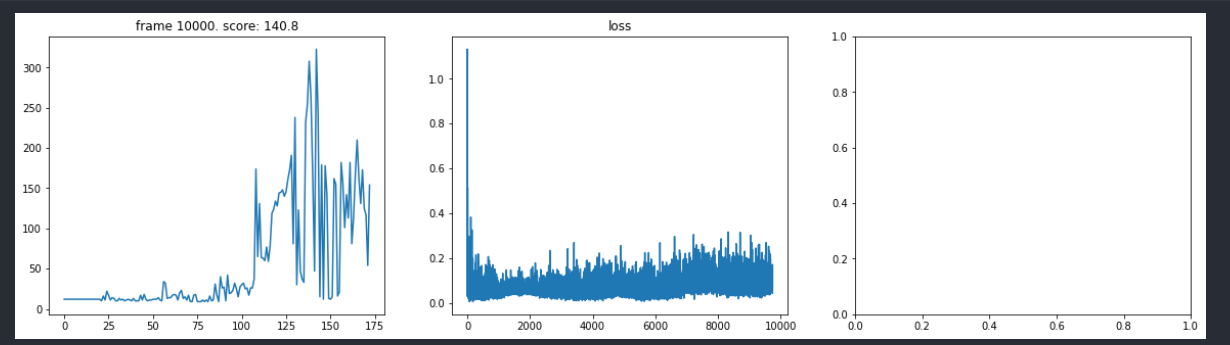
DQN-1min55s Priority-**3min6s**



Double-DQN-2min5s Noise-DQN-2min-43s



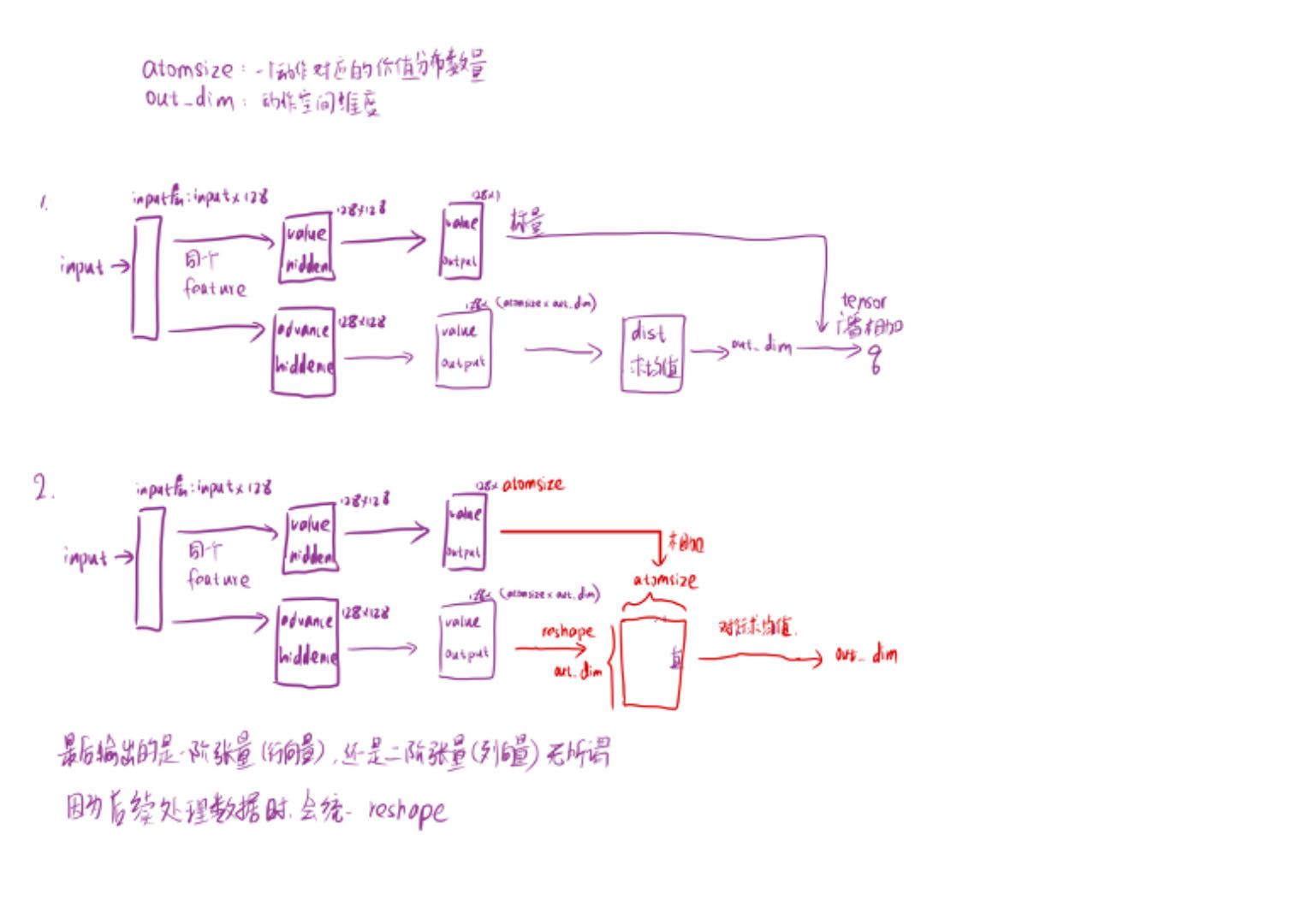
Dueling-DQN-2min29s Distribute-DQN-2min-16s



Rainbow-**7min43s**

# 反思

1. rainbow的最终效果更好，但是需要耗费的时间也更高，因为加噪声，计算采样优先级等都会加长时间
2. 在rainbow中，对于网络的架构，我认为可以有两种方法
3. value网络的输出维度是1，在advance网络求完均值后，再广播相加，得到Q
4. value网络的输出维度是atom\_size(一个动作的价值分布数量)



# 参考：

1. <https://wenku.csdn.net/answer/ee0943a630491793dd6426c56dd86081>
2. <https://github.com/Curt-Park/rainbow-is-all-you-need>
3. \*\*有关优先采样的论文分析\*\*
4. <https://aijishu.com/a/1060000000116781>
5. <https://zhuanlan.zhihu.com/p/337171931>
6. \*\*有关重要性采样的分析\*\*
7. <https://zhuanlan.zhihu.com/p/41217212>
8. 在求蒙特卡洛积分时，采样按照概率分布p(x)采样，则求积分是，需要对每一项采样进行重要性加权，称为重要性权重。

同理，在神经网络中，对训练样本按照某个概率分布p(x)输入到网络中，计算损失函数时，需要对每一项损失进行加权，也叫重要性采样。防止bias产生。

若仍按照原来的损失函数进行，则网络的输入数据会产生误差/便宜，拟合肯定不准确。所以要对高频率输入的数据降低重要性，对低重要性，低频率的数据的loss增加重要性，防止偏置。