DQN算法相关文档：

**目前进展：** 完成核心算法部分实现。使用简单游戏数据测试通过。在进行推荐数据的处理工作。

伪代码：

（1）定义化经验池D，设定容量为$N$；

（2）定义Q函数神经网络Q，随机初始化网络参数$w$；

（3）定义目标值神经网络$ \hat Q$，随机初始化网络参数$w'=w$；

（4） $Loop (m = 1,2,⋯,M ) $

（5） 对状态$s\_t$进行初始化；

（6） $Loop (t = 1,2,⋯,T ) $

（7） 结合主要推荐策略和基于探索的推荐策略生成当前状态下的动作$a\_t$

（8） 用户收到动作$a*t$ ，同时给出反馈$r*t$及产生新的状态$s\_{t + 1}$；

（9） 得到的样本$(s*t,a*t,r*t,s*{t + 1 })$存储在经验池D中；

（10） 从经验池D中随机采样 minibatch 的样本进行训练，表示为$(s*j,a*j,r*j,s*{j+1})$

若$j$为最后一步，则目标函数值$y*j=r*j$；否则，目标函数值设为$r*j+\gamma max*{a'}\hat Q(s\_{j+1},a';w')$

（11） 损失函数设置为$(y*j-Q(s*j,a\_j;w))^2$，使用梯度下降法更新Q函数神经网络参数$w$；

（12） 每隔 C步更新目标值网络参数，将Q网络的参数复制给目标网络，即$w'=w$；

（13） End loop

（14） End loop

公式部分：

（1）回报G

推荐代理算法的累计回报$G*t$由即时回报$r*{t+1}$和延时回报加权构成：

考虑对较远时间步的回报进行衰减，引入衰减因子$\gamma$。

$\gamma$表示衰减因子，需要设置的超参，对渐远时间步的回报进行衰减。

（2）DQN算法更新公式推导：

$$Q(s,a,t)=r\_{a,t+1}+\gamma r\_{a,t+2}+\gamma ^2r\_{a,t+3}+...\\
=r\_{a,t+1}+\gamma (r\_{a,t+2}+\gamma r\_{a,t+3}+...)\\
=r\_{a,t+1}+\gamma Q(s\_{a,t+1},\mathop{argmax}\limits\_{a'}Q(s\_{a,t+1},a';w\_t);w\_t')$$

（3）样本的目标值$y\_j$计算公式

（4）训练损失函数定义：

结构图：

DQN核心算法流程