# 杨雪婷工作周报（2021.05.10 - 2021.05.14）

1. 本周工作进展情况：
2. 环境代码方面：周一的时候添加计算QoE需要用到的前后size\_level列表后还没有进展。
3. 思路方面：周一周二阅读了RL-Cache（7页）的那篇论文，发现了很多不懂的强化学习术语，大致知道他想用多个参数当作reward（区别于非神经网络的传统算法）来更新神经网络权重，但是还没懂他具体怎么操作（比如用什么激活函数更新，是更新policy函数还是更新value）。我感觉自己的问题是没把他论文里的参数和他在强化学习里的实际意义对应上，读起来很费解，于是计划这周末前把RL简单入门。（找了个YouTube上的课已经看了3/5了qaq）

### 目前结合我们的一点点想法：

我们的目标：提升用户QoE + 提高hit\_ratio

写程序让AI控制cache系统决定：1. whether to cache 2. which object to evict

两种办法：

* Policy-based-learning 学习一个policy函数Π，用policy控制agent做动作，每观测到一个状态st，作为Π函数的输入，会输出每一个动作的概率，按照概率做随机抽样得到at控制agent做动作。
* Value-based-learning，学习的是Optimal action-value ：Q\*(s,a)。将当前状态St作为输入，对于每一个动作action都做一个评价，选择Q值最大的来做：

1. 目前存在问题：
2. 对于环境的代码需要添加buffer地方，挺难写的，好像要有不少改动，目前还没什么好的思路去计算buffer相关的数。
3. 强化学习方面薄弱使得看论文难抓住重点而且看的挺慢。希望学完这个入门视频之后能读懂一点。
4. 下周计划：

周末把网上入门视频后两节看完，下周再读一下JSAC-RL-cache后做一下归纳整理。

19 数媒技 杨雪婷

2021.05.14

下面附上我的强化学习笔记：(/捂脸)

## Part One -- 理解强化学习术语

RL：通过奖励来更新模型参数让模型越来越好。

### 1、一些名词：

**Agent**：智能体，做动作的主体

**State** ：状态

**Action**：

**Policy**：根据观测到的状态做出决策来控制运动 Π(a|s) = P(A = a | S = s) ，就是给定状态S做出动作A的概率密度。强化学习学的就是policy函数。游戏中：策略随机

**State transition**：S1 -> action1 -> S2 (获得r1) -> 做action2,得到s1 a1 r1,s2 a2 r2 ... st at rt.

**Reward**：奖励（定义的好坏非常影响结果）

**Return**: 由于未来的奖励没有现在的值钱，使用折扣回报(discounted return γ)

γ为0，1之间，越久远的未来折扣越大权重越低

Ut是随机变量（在t时刻不知道是什么），可以求期望。如抛硬币知道正反有一半概率而不知道下一个会得到什么。

**Action Value Function**：

意义：如果用Policy函数Π，在St状态下at是好还是坏。QΠ 会给当前状态下的所有action打分。

**Optimal action-value function**：

将上一个式子中的Π去掉：（有无数种policy函数，将此函数最大化）

意义：St状态下，评价at动作的好坏。

**State-value function**:

意义：可以表达出当前的局势好不好，局势越好VΠ越大；也可以评价Π的好坏，Π越好，VΠ的平均值越大。（VΠ和policy和状态有关，对a做运算）



常用符号

## Part Two -- Value-based learning (DQN + TD)

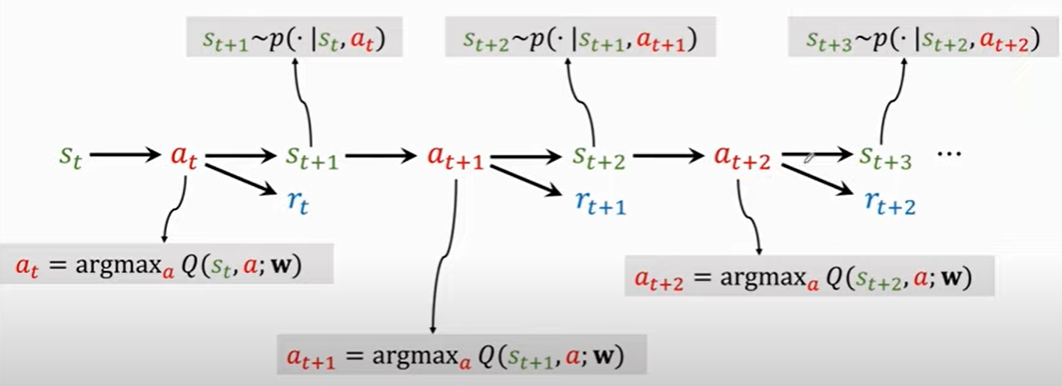
其实就是用神经网络近似Q\*函数。

### 1、Deep Q-Network (DQN)

神经网络Q(s,a;w)：输入为状态，参数是w，输出是对每一个动作的打分。

state -Conv卷积层-> feature 特征向量 -Dense全连接层-> 输出打分

利用DQN算法的过程：



### 2、如何训练DQN -- 用TD算法(Temporal Difference)

整体流程：

a. 当前状态St和当前动作At

b. Qt = Q(s,a;w), Qt 是神经网络的输出，是对每个状态的打分

c. 反向传播求导，PyTorch和TF都可以自动求梯度：

d. 环境更新成状态St+1和奖励Rt

e. 求TD target：

f. 梯度下降更新权重

整个过程 = 两个过程相加(TD)

## Part Three-- Policy-based learning (Policy network + Policy gradient)

### 1、policy network

**Π(a|s)**：概率密度函数，表示当前状态时某一个action的概率（输入是s)

用**神经网络**近似policy函数**Π(a|s)**，神经网络记作：**Π(a|s;θ)**

state -Conv卷积层-> feature 特征向量 -Dense全连接层-> action个数的向量 -> Softmax激活函数 -> 每一个action的概率

通过神经网络学习的Π可以得到V：

策略学习的主要思想：改进模型参数θ让V变大，使得**J(θ)越大越好**。-> policy gradient ascent

* state s

### 2、Policy Gradient

### 3、Update policy network using policy gradient

a. t时间的状态St

b. 用Π(a|s;θ)随机抽取得到action At

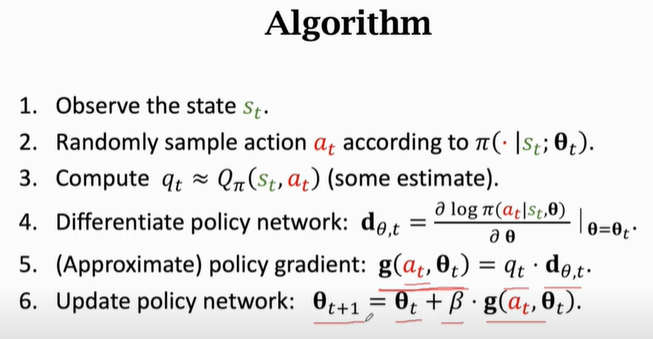
c. 计算价值函数Qt = QΠ(St,At)

* 策略网络控制agent运动，记录agent的轨迹和reward，得到Ut来近似QΠ
* 或者用神经网络来进行函数近似。

d. 求导得到d\_θt

e. 近似计算策略梯度

f. 用近似的策略梯度更新策略网络的参数



## Part Four -- Actor-critic method

其他了解：

强化学习和监督学习、无监督学习 最大的不同就是不需要大量的“数据喂养”。而是通过自己不停的尝试来学会某些技能。[链接](https://easyai.tech/ai-definition/reinforcement-learning/)

两种强化学习的主流算法：

1. 有模型学习（Model-Based）对环境有提前的认知，可以提前考虑规划，但是缺点是如果模型跟真实世界不一致，那么在实际使用场景下会表现的不好。
2. 免模型学习（Model-Free）放弃了模型学习，在效率上不如前者，但是这种方式更加容易实现，也容易在真实场景下调整到很好的状态。所以**免模型学习方法更受欢迎，得到更加广泛的开发和测试。**