# 杨雪婷工作周报（2021.05.17 - 2021.05.21）

1. 本周工作进展情况：
2. 环境代码方面：这周一讨论完后添加了rebuffer的部分并计算了quality后进行了可视化、了解了下SVC的问题。
3. 思路方面：读了RL-Cache的论文和部分RL-360的论文，了解了用什么方法但是没具体明确输入输出和神经网络结构。去网上找了下典型的强化学习应用实例（AlphaGo）加深了一点点对各种强化学习参数的理解，但是这个例子不是很符合。
4. 开发框架：上周一之后就忘记要调研SVC的事情了，去了解了下Pytorch。（感觉这个不是很着急，但是好多需要学的qwq，正好一个同学做的课题用Pytorch我就找他要了推荐的课程）但是目前只看了环境搭建还没看完...。
5. 目前存在问题：
6. 代码方面：本周五（今天）上午讨论后又发现了计算上的问题和还需要添加的指标（FoV内高清占比，viewport里的平均码率，码率change的比率，了解psnr信道比参数等）。
7. 还是代码方面：在做改动（尤其是rebuffer方面）完全没有头绪，计算参数也很不熟练，师哥帮我讲好久才知道咋写，应该还是对整个系统和buffer的流程不熟悉。拖了师哥的后腿）
8. 看论文的时候重点看输入输出和具体结构，现在还是看得没那么明白，自己觉得懂了，一提问就又不行了。
9. 下次讨论的时候记下来自己要干的事，防止又没记住耽误大家。
10. 下周计划：

周一前看完《DRL360: 360-degree Video Streaming with Deep Reinforcement Learning》，下周五前看完《Viewport-Aware Deep Reinforcement Learning Approach for 360o Video Caching》同时做整理。

19 数媒技 杨雪婷

2021.05.21

下面附上我这周的笔记：(/捂脸)

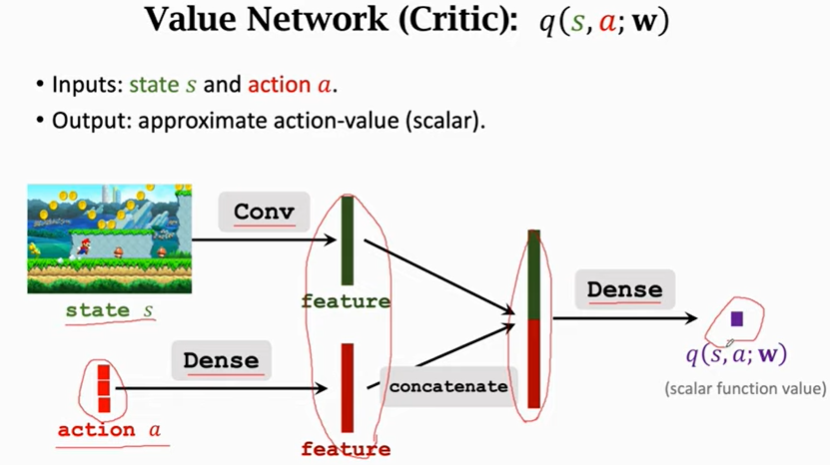
## 接上次的Part Three：

## Part Four -- Actor-critic method

actor：策略网络，相当于运动员

critic：value network，相当于裁判员

输出的实数q是裁判对运动员打的分数。s的情况下，做action A 是好还是坏。



### Train the networks

学习两个神经网络的目的：

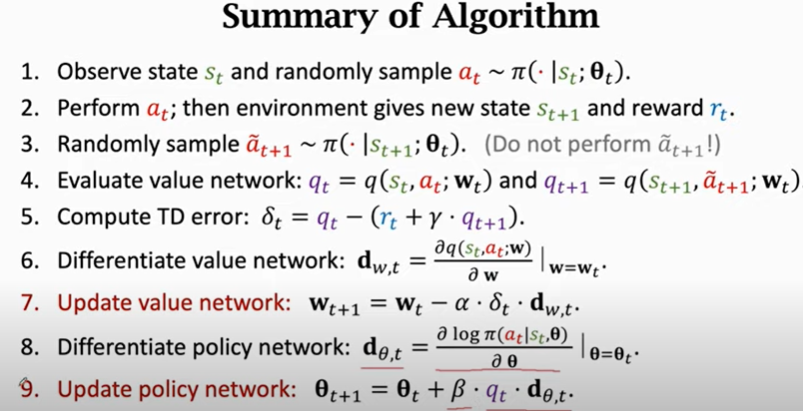
* policy network：让运动员得到更高的平均分，更新Π函数，让V函数的值更大。
* value network：q函数当作裁判，给运动员打分越来越精准。

训练中，q函数辅助训练Π，训练后，q函数就没用了，通过Π来操作agent运动。

训练两个神经网络的更新方法：

1. 观察当前状态s
2. 根据策略函数对action进行概率分布，随机取得action
3. 做action更新当前状态St+1得到reward Rt
4. TD 算法更新value network的w。（裁判更准确）
5. policy gradient算法更新policy network 中的θ。

算法的总体流程：（每一轮只做一个动作，更新一次神经网络）



## Part Five -- AlphaGo

主要设计思路：

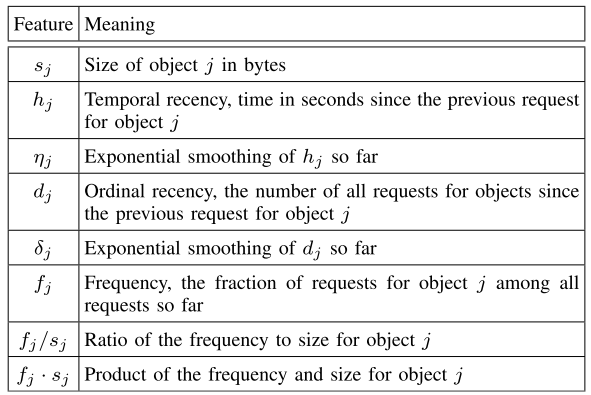
* behavior cloning（模仿学习，非强化学习，监督学习，多分类/回归，将361个状态看成361个类别，计算每种类别的概率，人真实的动作就是标签）
* police gradient算法训练策略网络（策略网络做自我博弈）
* 训练一个价值网络评价状态的好坏（先训练策略网络，再用他训练价值网络）
* 蒙泰卡罗搜索树，用两个网络指导搜索来剪枝（策略函数算概率值，排除概率值低的，自我博弈通过胜负和reward给动作打分，选择分数高的）

下面的是《RL - Cache： Learning-Based Cache Admission for Content Delivery》的一点笔记：

### 1、Feature selection

考虑的三个方面：

* object size
* request recency
* request frequency



此方法的优点在于： The strength of our approach is in simultaneously considering a broad set of eight features from these three classes.

recency有很多种不同的定义，且论文将recency和size相结合。

### 2、RL Problem Formulation

state : 特征的向量（不包含缓存占用防止重复计算）

目标：设计神经网络近似policy function 使得return最大，即命中率最大

### 3、RL approach

model-free RL algorithm：

* DPS：利用具有高return的policy子集通过搜索得到最好的policy方法，用神经网络模拟policy函数，通过更新参数得到更好的policy策略。

### 4、Neural-network architecture

ANN with ELU（5个隐藏层中的激活函数）

**输入层**有n个神经元。

**隐藏层**，共5层，第l个隐藏层有5\*(6-l)\*n个神经元。

用 L2 regularization防止过拟合。

**输出层**是两个神经元，表示是否被admit的两个概率

8个特征向量是连续的，量化后作为输入层的输入。

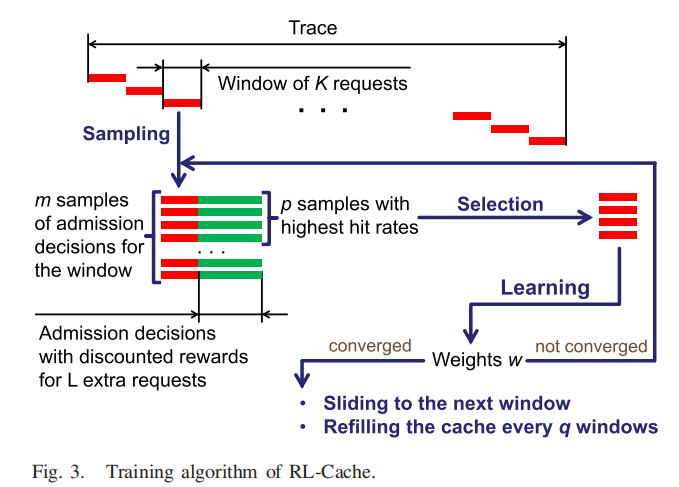
### 5、Training algorithm

目标：调整神经网络的权值。

Discounted 函数的定义：

前K个是真实的不需要折扣函数：

步骤：K长度的滑动窗口MC采样后考虑m个decision，获取return前p高的samples，利用二进制交叉熵损失作为损失函数进行反向传播算法。



循环直到达到某一个阈值内或达到最大值。然后滑动窗口到下K个。

注：每滑动q个窗口后都将权重重置一下。云中进行，定期向服务器提供一个更新后的权重版本。

**6、Implementation**

具体实现：Tensorflow框架，在cache server中需要一个数据库，计算频率和最近度量，且可以将最新更新的神经网络应用于请求上。活动字节：为了判断此chunk是否被admit用来预测的