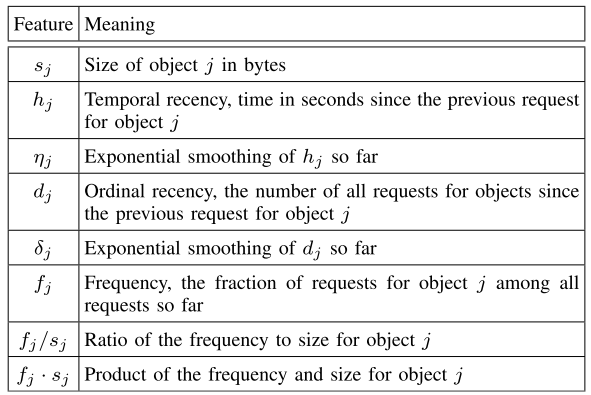
# RL - Cache： Learning-Based Cache Admission for Content Delivery

## Algorithm and implementation

### 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。

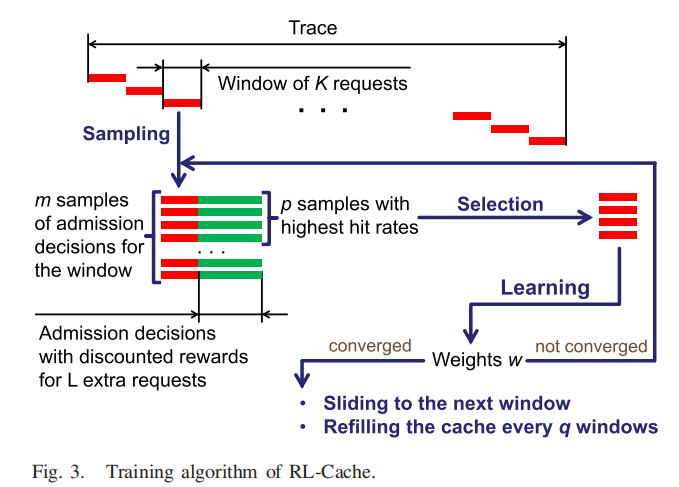
### 5、Training algorithm

目标：调整神经网络的权值

在云上周期性训练，仍然需要考虑计算开销。多个连续请求而不是单个请求可以长期评估。

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、Real-time operation

### 7、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

## Evaluation

还没整理）