

# 《DeepCache: A Deep Learning Based Framework For Content Caching》

CUC - 19 数媒技 杨雪婷

2021.05.31

# 汇报内容介绍

1. 背景
2. 结构
3. 实验
4. 评估

# Part One : 背景

# 问题：cache & evict

## core idea:

提前预测对象特征（预测对象的流行度）

- 提前预取对象，提高缓存命中率
- 了解对象的流程度，防止替换出流行的chunk，减少网络抖动问题

## goal:

increase the number of cache hits

自适应缓存机制，了解请求流量的变化，预测未来特征，~~觉得~~缓存和驱逐哪些对象。  
决定

## Part Two : 结构

## 整体结构：

1. popularity prediction: seq2seq model (LSTM), 基于LSTM的模型预测内容对象流行度
2. DeepCache 的缓存框架：通过流行度做出决策

## Seq2seq prediction

预测需要：

1. 预测未来对象的特征
2. 连续预测
3. 利用特征提高缓存表现

--> **LSTM 模型**

## LSTM 模型的输入输出

在过去流行度的基础上构建未来流行度，这里的向量特征都是**流行程度**

输入：t长度的数组，每个元素是与对象对应的d维的特征向量（d=不同objects的数量）

$$X_t = \{x_1, x_2, \dots, x_t\}$$

输出：t长度的数组，每个元素是与对象对应的p维的特征向量（p=不同objects的数量）

$$Y_t = \{y_1, y_2, \dots, y_k\}$$

最终是以这种格式：(#samples , d,m, 1) and (#samples , ~~d~~, k, 1)

p



## Caching policy

周期性的生成“fake content requests” 合并到原本的请求的后面

## Part Three : 实验

## 实验参数：

## 数据信息：

$m = 20$

数据1：50个不同的请求，80K个requests，6个间隔，每个间隔1000个请求左右，流行程度的排序是固定的，popularities满足参数不同的Zipf分布。 $(k = 10, M = 5)$

数据2：更类似真实数据，1425 unique objects with more than 2 million requests，每个都有存活期限。有Zipf的参数。 $(k = 26)$

## LSTM神经网络结构：

两个隐藏层，128和64个神经元

损失函数：MSE

epoch：30

## Part Four : 评估

- LSTM Encoder-Decoder Prediction Accuracy
- Cache-Hit Efficiency