面向同步辐射光源图像的智能压缩方法

报告人：符世园

中国科学院高能物理研究所

大纲

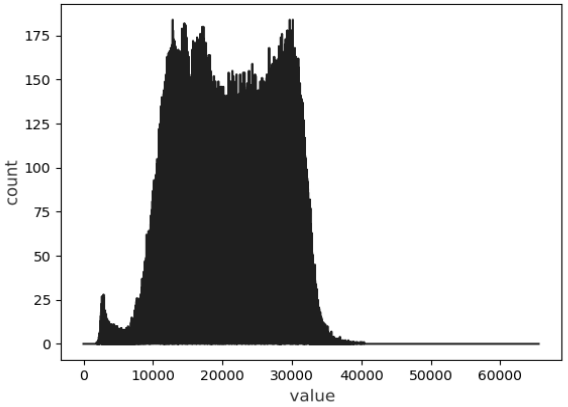
- 背景介绍
- 智能图像压缩方法
  - 线性预测
  - 分区量化
  - 非线性预测
- 总结

背景

- 在建的高能同步辐射光源将开展高通量同步辐射实验
  - 超高空间分辨、时间分辨、能量分辨
- 实验过程中会产生海量数据
  - 一期建设的实验站预计平均每天产生200TB的原始实验数据
  - 峰值可达每天500TB，一年产生的数据量达到150PB
  - 对数据传输和存储带来极大挑战
  - 其中硬X射线成像实验线站产生的图像数据占比最高

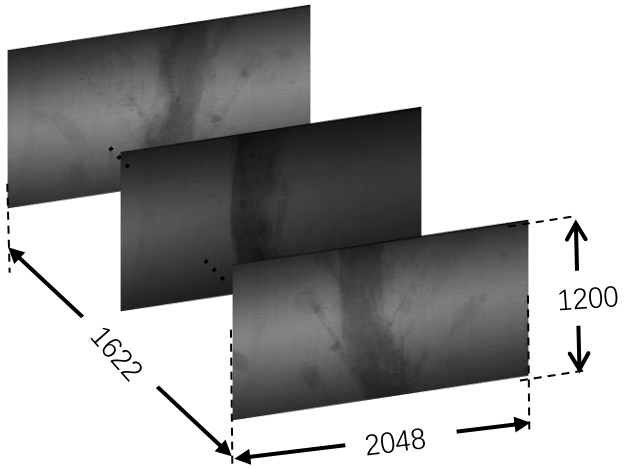
背景

- 光源图像成像是对样品进行不同角度的投影成像
  - 平行光照射得到投影
  - 16-bit单通道灰度投影透视图
  - 高分辨率、高帧率
  - 每次拍摄会产生上GB的数据
  - 每个样品拍摄得到的图像序列可以看做视频

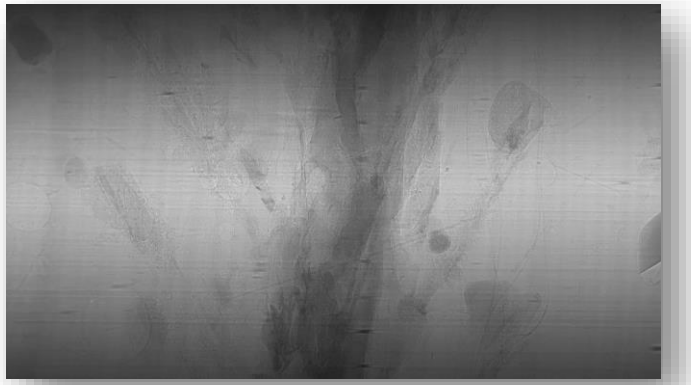


像素值分布

需要无损压缩缓解存储和传输压力

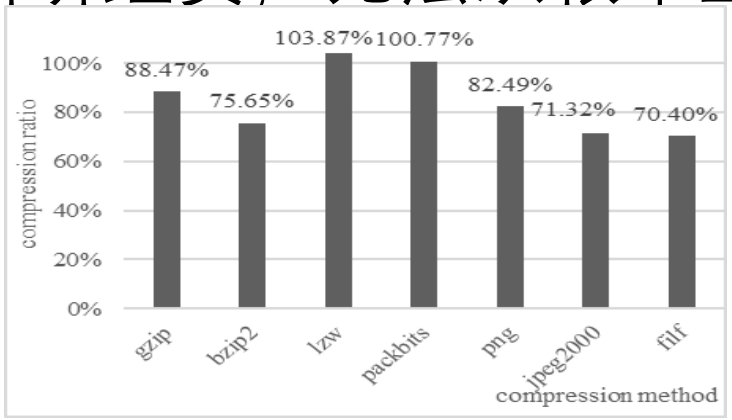


光源投影图序列示例



背景

- 光源数据量在不断增加，并且需要长期保存
- 存储系统的横向扩展需要大量科研经费，无法从根本上解决问题
- 传统无损压缩方法效果不佳
  - 优点：过程简单
  - 缺点：方法固定，压缩率较高
  - 最优压缩率在**70%**左右
    - 压缩率 = 压缩后文件大小/原文件大小

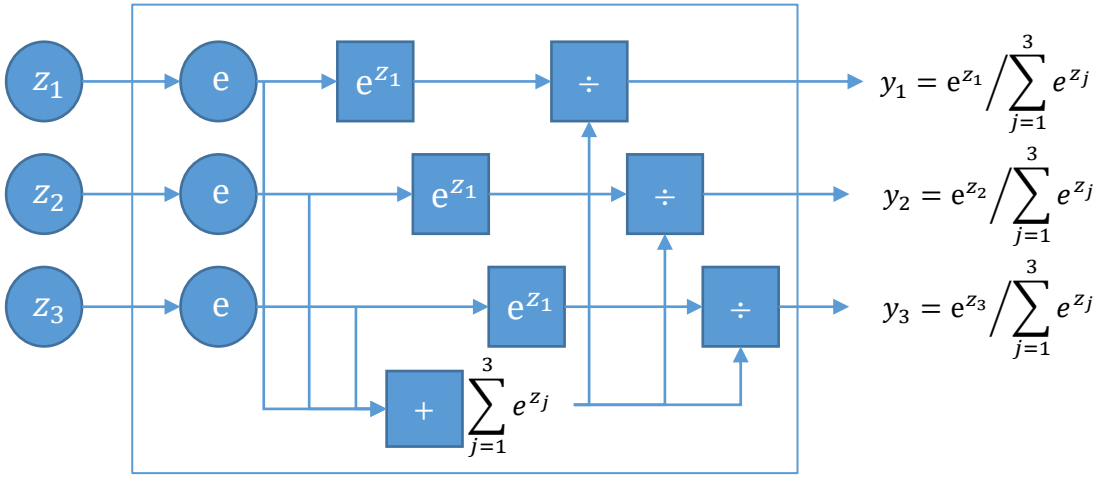


直接使用无损压缩方法的压缩率

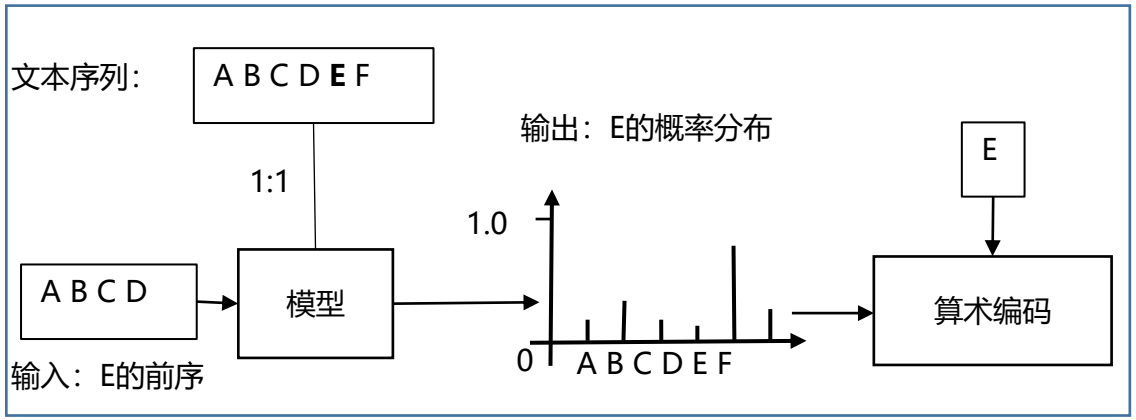
- 随着神经网络的兴起，出现了一系列使用神经网络的数据压缩方法
  - 无损压缩：deepzip

Deepzip

- 神经网络的softmax层与熵编码天然可结合
- 不同的数据对应不同的模型，模型追求过拟合
- 以前文预测后文进行压缩
- 为保证解压无损，保存压缩文件的同时也需要保存模型



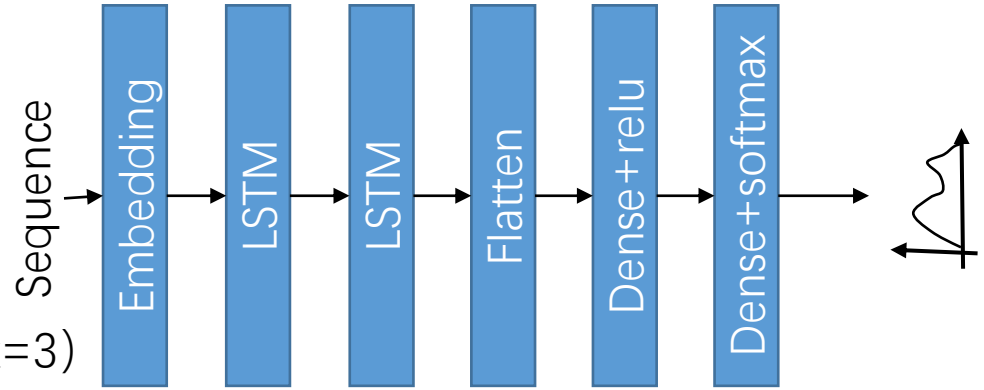
Softmax图示



Deepzip图示

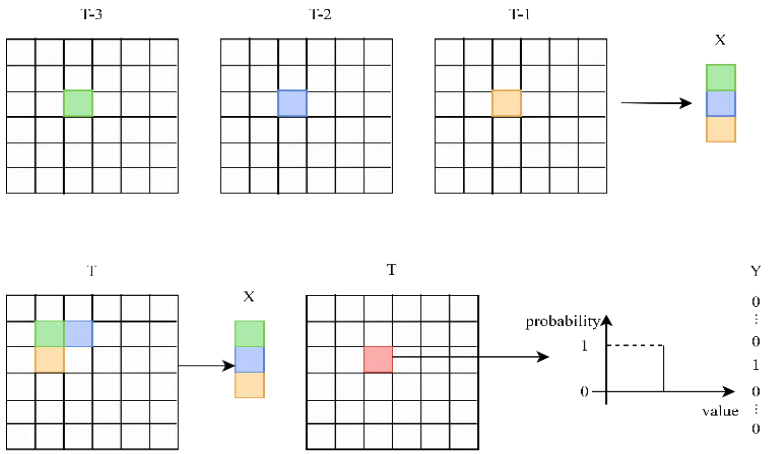
Deepzip

- Deepzip压缩率测试（LSTM）
  - 输入构造方法（时间/空间）
    - 时间序列
      - 前k张图像同一空间位置像素值按时序排列（k=3）
    - 空间序列
      - 左上角N×N-1像素值按行扫描顺序排列（N=3）



模型结构

- 问题：对于原图压缩率接近100%(**99.56%**)
  - 在字典数较多的情况下，概率的分布较分散
  - 每个值对应的区间较小，导致算术编码压缩效果较差

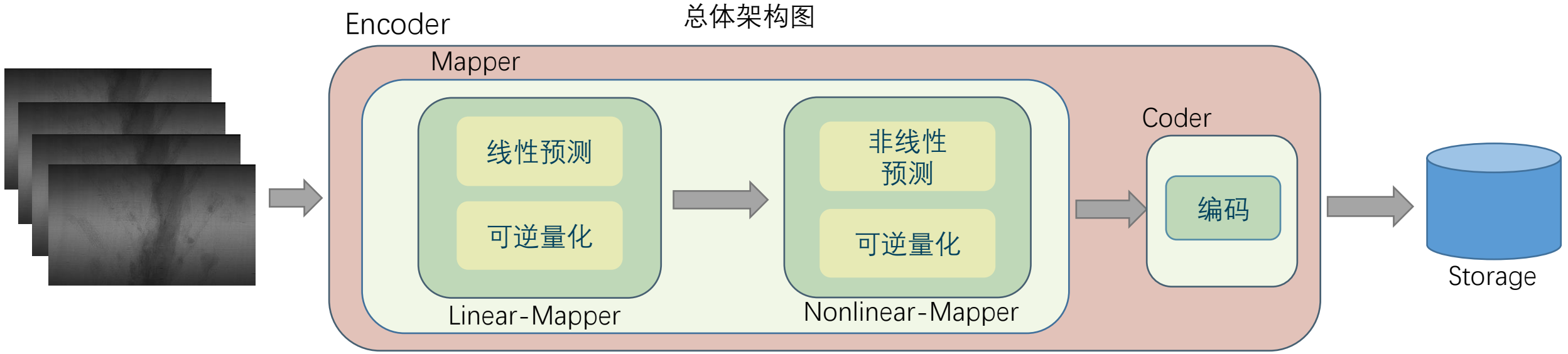


输入输出构造方式

总体架构

- 不同映射子模块采用不同的预测方法和自适应的量化系数
  - 分为线性预测和非线性预测，分别去除冗余信息
  - 针对光源图像特点提出非线性预测模型构造方法
  - 针对光源图像特点提出系数自适应的量化方法

总体架构图



线性预测

- 目的：去除图像序列内部的线性冗余
- 衡量指标
  - 空间相关性
    - 指图像内部某像素与相邻像素之间的相似性，使用自相关系数衡量
    - 对于尺寸为M×N的图像，空间相关性计算公式如下：

$$C(l,k)=\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)-U_f\right]\left[f(x+k,y+l)-U_f\right]}{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f(x,y)-U_f\right]^2},$$

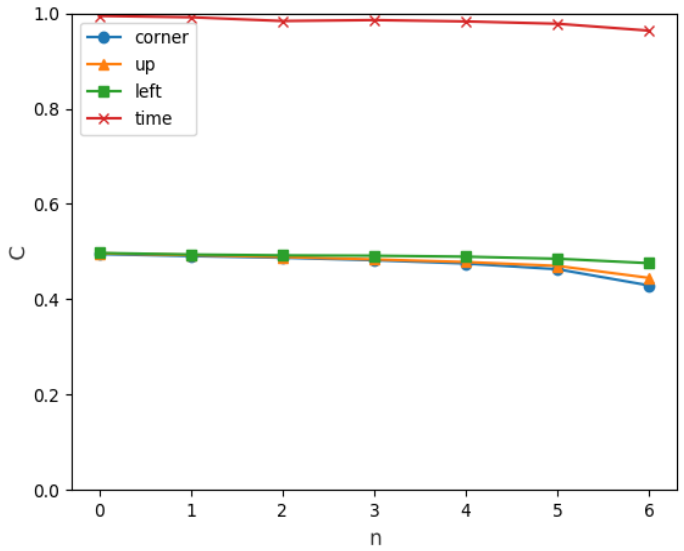
  - 时间相关性
    - 连续两张图像同一空间位置像素之间的相关性，使用互相关系数衡量
    - 计算第*i*张图像与第*j*张图像的时间相关性公式如下：

$$C(i,j)=\frac{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f_i(x,y)-U_i\right]\left[f_j(x,y)-U_j\right]}{\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f_i(x,y)-U_i\right]^2}\sqrt{\sum_{x=1}^{M}\sum_{y=1}^{N}\left[f_j(x,y)-U_j\right]^2}}$$

    - *l*为左侧间距，*k*为上方间距，U_f为图像像素值均值，*f(x，y)*为图像对应位置的像素值

线性预测

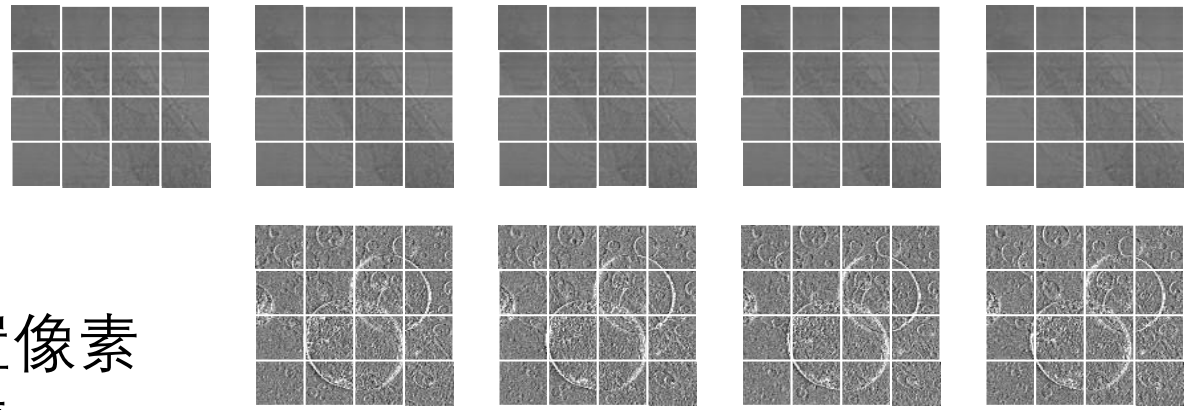
- 原图相关性分析
  - 对示例图像序列，每隔一百张取一张图像，计算其空间相关性和时间相关性
    - 时间相关性分别取整个图像序列中前2^n（$0 \leqslant n \leqslant 6$）张图像进行计算
    - 空间相关性分别取间距为2^n（$0 \leqslant n \leqslant 6$）的左、上、对角位置像素值进行计算



原图时间相关性和空间相关性

- 时间相关性大于空间相关性
  - 时间相关性接近于1
  - 空间相关性在0.5左右
- 对于空间相关性，左侧元素相关系数略高
- 随着n增大，相关系数有下降趋势

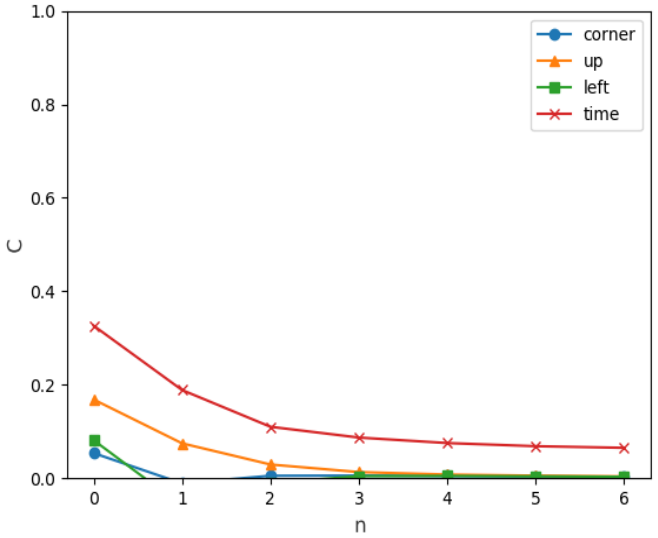
线性预测



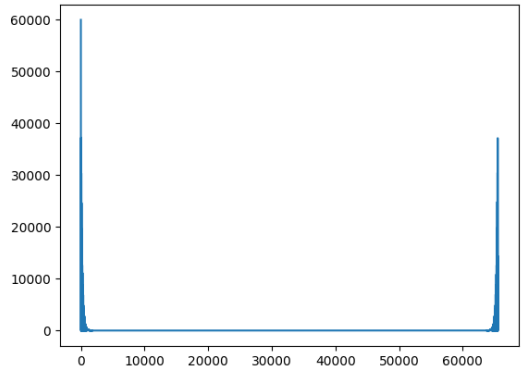
时间差分前后图像对比

- 两种差分方式
  - 时间差分：减去前一张图像同位置像素
  - 空间差分：减去左侧相邻位置像素

- 空间差分结果：
  - 时间相关性降低到0.4以下
  - 空间相关性降低到0.2以下
  - 随着n的增大，时间相关性和空间
  - 相关性呈下降趋势



空间差分后时间相关性和空间相关性

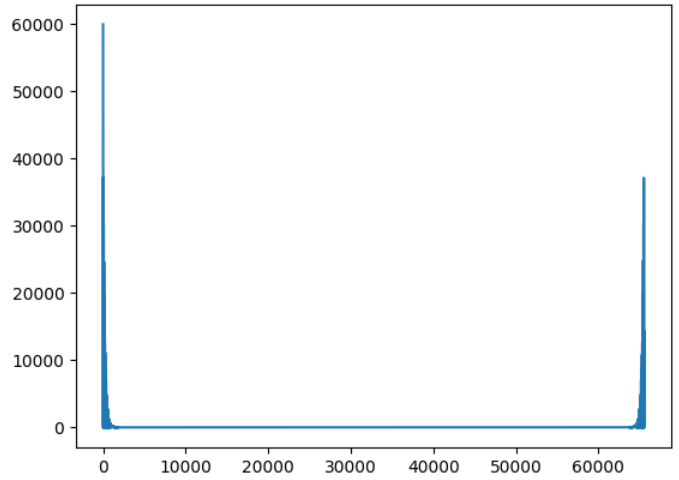


差分后图像像素值分布

- 差分可以有效消除图像序列内部的相关性、使像素值分布更集中、加强图像特征

分区量化

- 差分后图像像素值分布更为集中
  - 占比很小的一部分数据占用很大的分布范围
    - 数据集中分布在两端范围
    - 分布范围中部仅有少量数据存在
  - 数据分布范围对应编码中的字典数
    - 对于熵编码：字典数越小，字典中每个值对应的编码长度（huffman编码）越短或范围（算数编码）越大，压缩效果越好

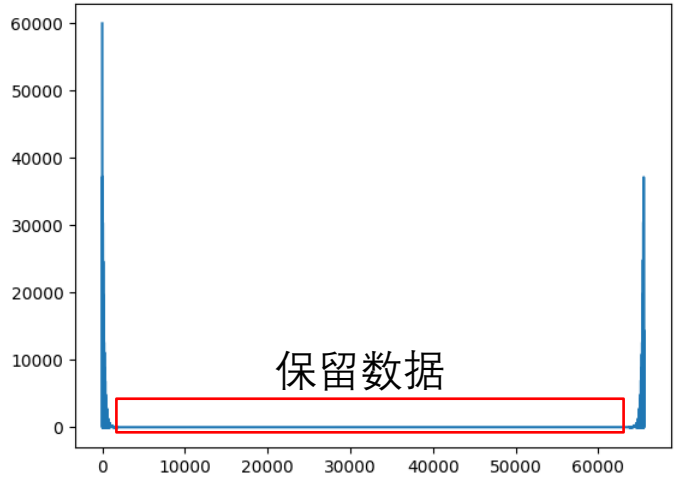


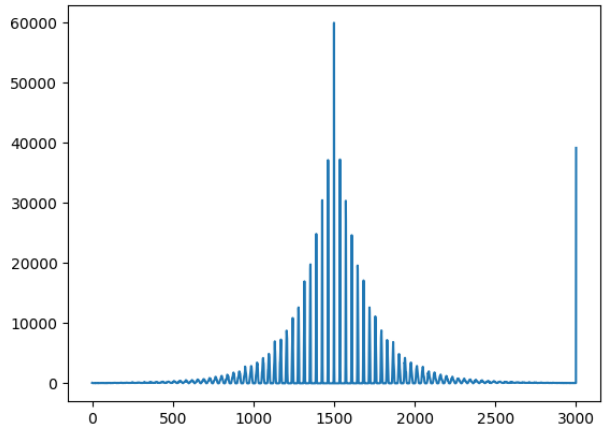
差分后图像像素值分布

分区量化

- 提出分区量化方法用于大幅缩小数据分布范围
  - 将数据分布一分为二
  - 以少量未压缩数据为代价换取更小的数据分布范围
  - 待压缩数据：
    - 用于后续非线性预测的输入
  - 保留数据[bound/2 + 1, 65536 – bound/2]
    - 不处理，直接保留

差分后图像像素值分布



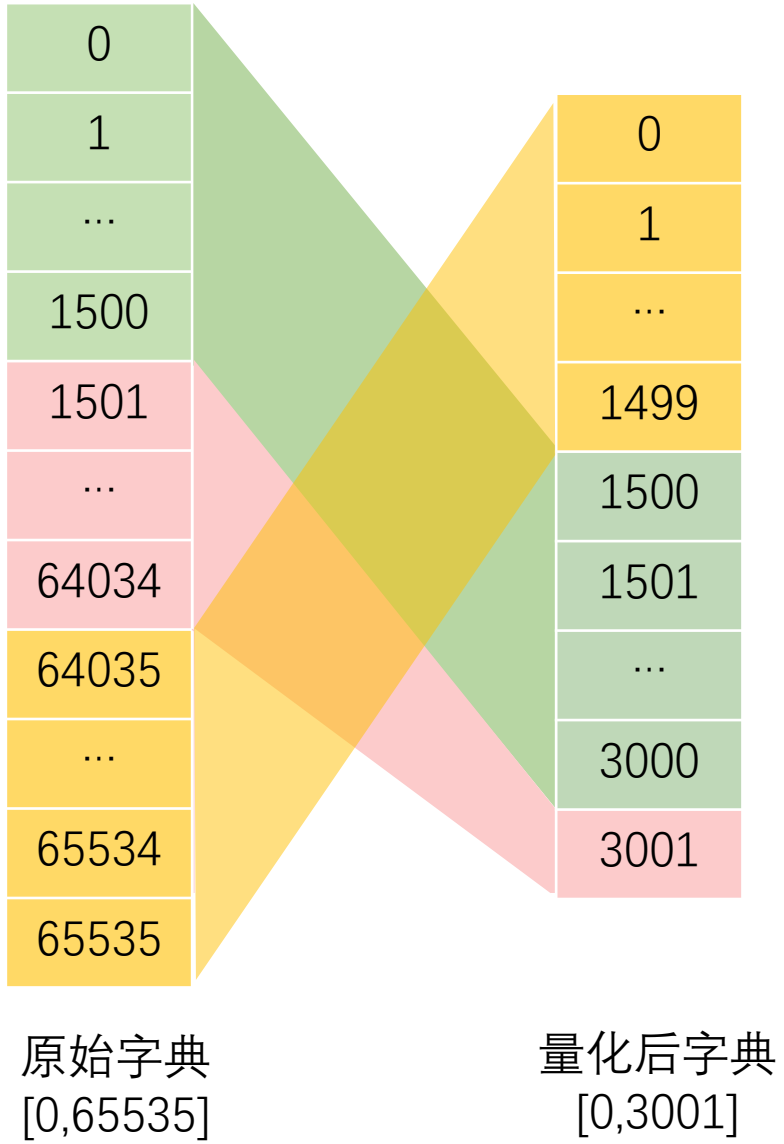


分区量化后待压缩数据分布

分区量化

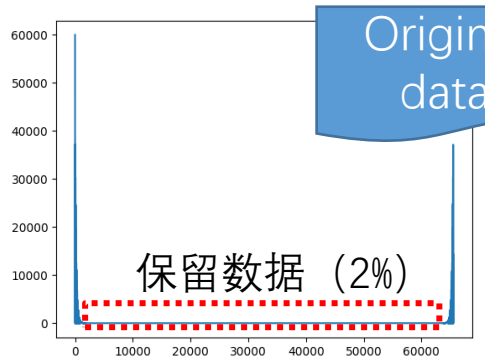
- 分区量化：将数据分布一分为二
  - 待压缩数据：
    - 映射到新的数据范围[0,bound]
    - 向右平移 bound/2
  - 保留数据
    - 在压缩数据中设置占位符
      - 占位符：[bound/2 + 1, 65536 – bound/2] → [bound+1]
  - Bound的确定方法
    - 以保留数据占总体数据量的2%以内为依据
      - Hold(F(x,y)) = 1 if F(x,y) in [hold_left, hold_right]
      - Sum(Hold)/(X×Y) = 2%

分区量化图示



分区量化

- 分区量化是可逆的



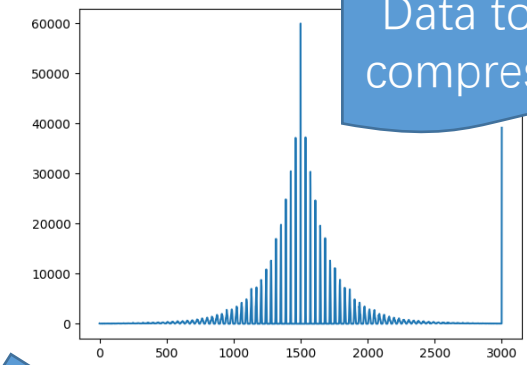
Original data

Get bound

数据被分为两部分
其中保留数据存入persist_data
在待压缩数据中以占位符存在

Quantizer (bound)

待压缩数据个数
与原始个数相同



Data to be compressed

保留数据按扫描顺序
依次按原始值存储

Persist_data

Non-linear compressor

Non-linear decompressor

Dequantizer (bound)

Compressed data

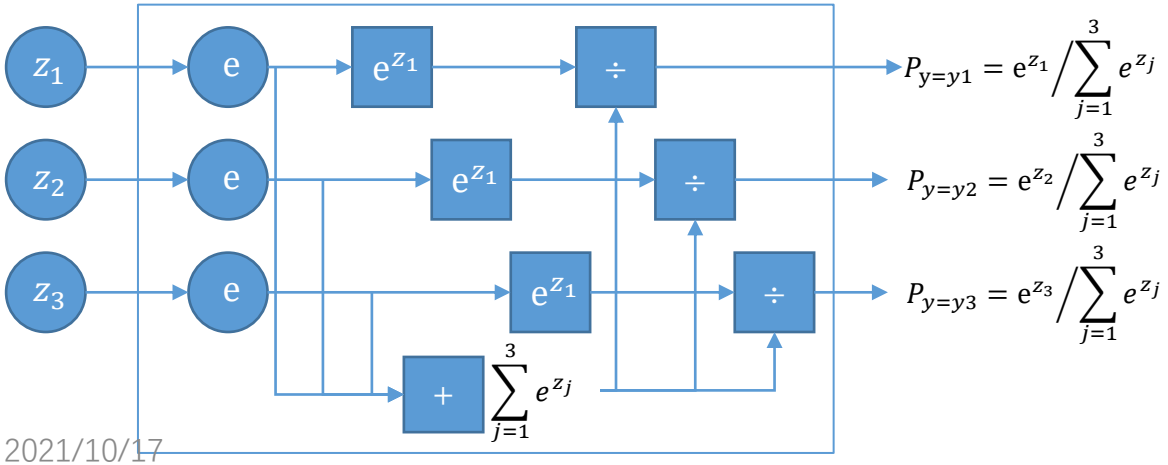
扫描persist_data按顺序替换
解压数据中占位符即可保证
分区量化过程完全**可逆**

量化数据流

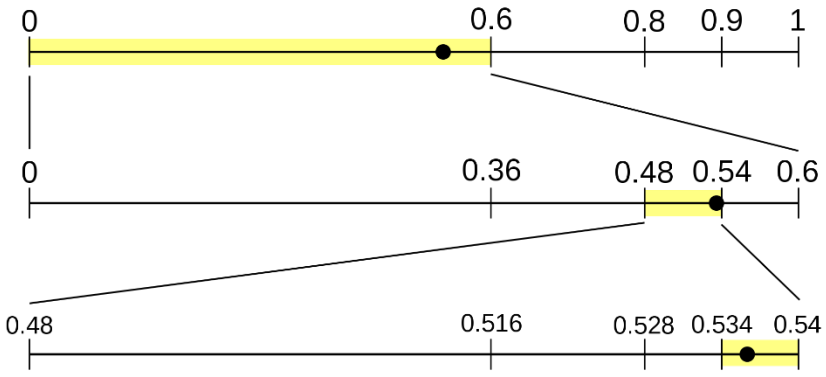
反量化数据流

非线性预测

- 前提：图像序列经过差分及分区量化后，已经去除部分线性冗余
- 目标：通过Nonlinear-Mapper消除非线性冗余
- 非线性关系通过神经网络进行挖掘
  - 神经网络具备非线性关系拟合能力
  - 神经网络已被用于文本压缩
    - 神经网络的softmax层与熵编码方法天然可结合，如deepzip



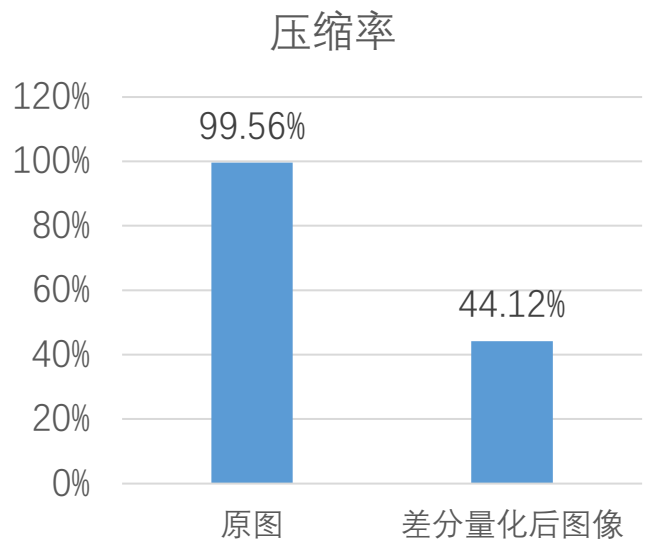
Softmax计算过程图示

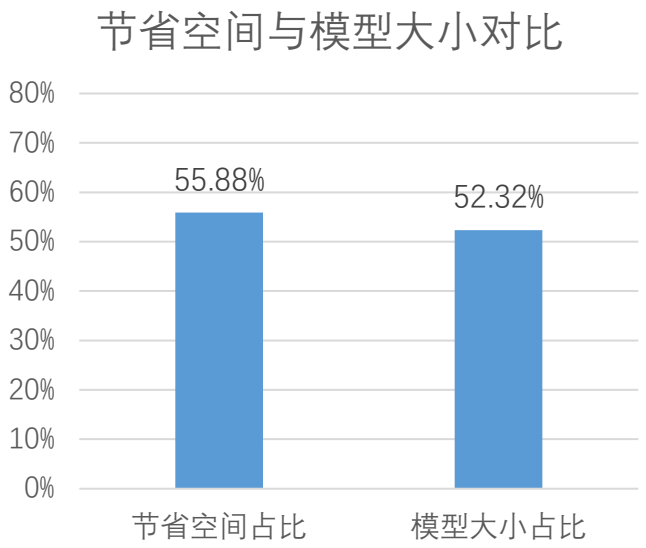


算术编码图示

非线性预测

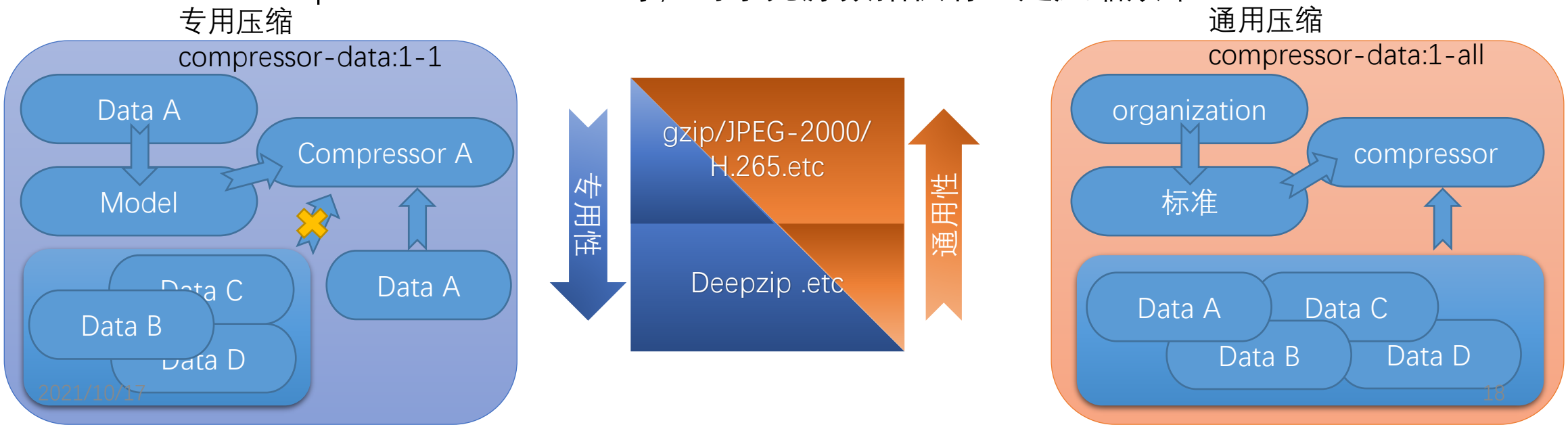
- Deepzip压缩测试（LSTM）
  - 最优结果为时间差分结合时间序列构造方法
  - 结果：差分量化后图像压缩效果大大优于原图压缩效果（50%↑）
  - 结论：差分量化后图像可以达到通过非线性预测方法提升压缩率的目的
  - 问题：图像与模型为一对一关系，模型大小抵消压缩节省存储空间





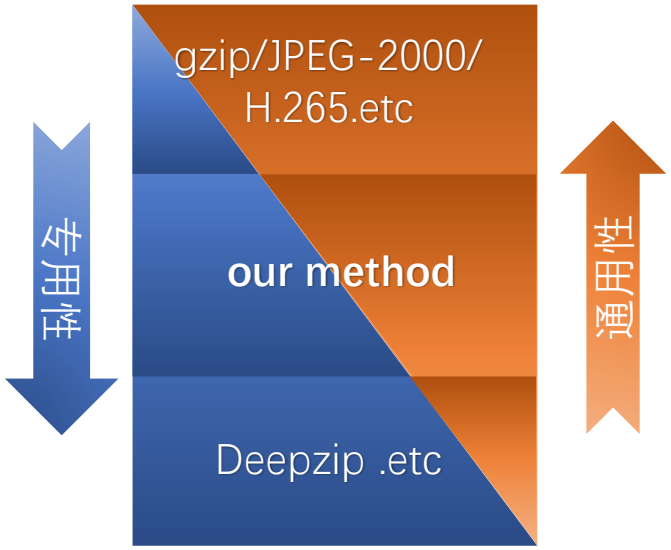
非线性预测

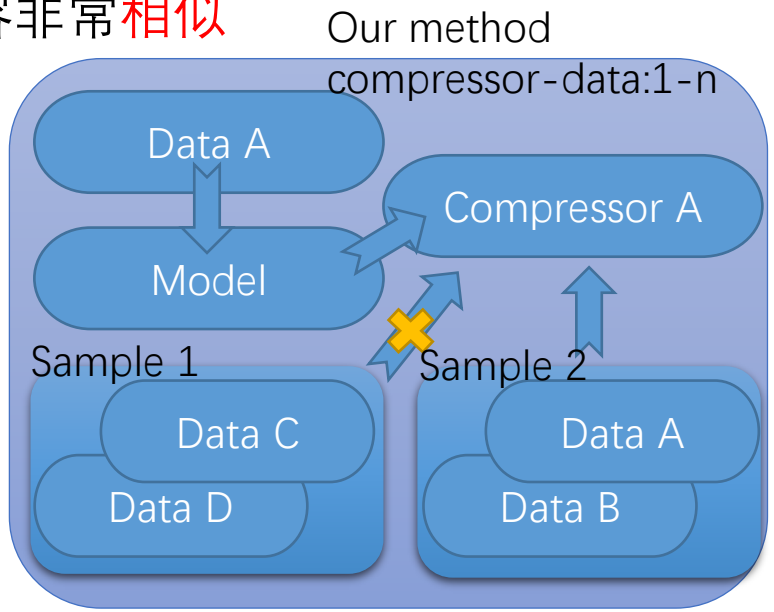
- 问题：模型大小抵消图像压缩的提升（动机）
  - 专用压缩VS.通用压缩
    - 专用压缩：追求在特定少数数据上有超出通用压缩方法的压缩效果
      - Deepzip遍历需压缩的所有数据建立模型，训练时间较长
    - 通用压缩：追求通用性，对于某大类数据能够达到一定的压缩效果
      - Gzip/JPEG-2000/H.265等，对于光源数据仅有一定压缩效果



非线性预测

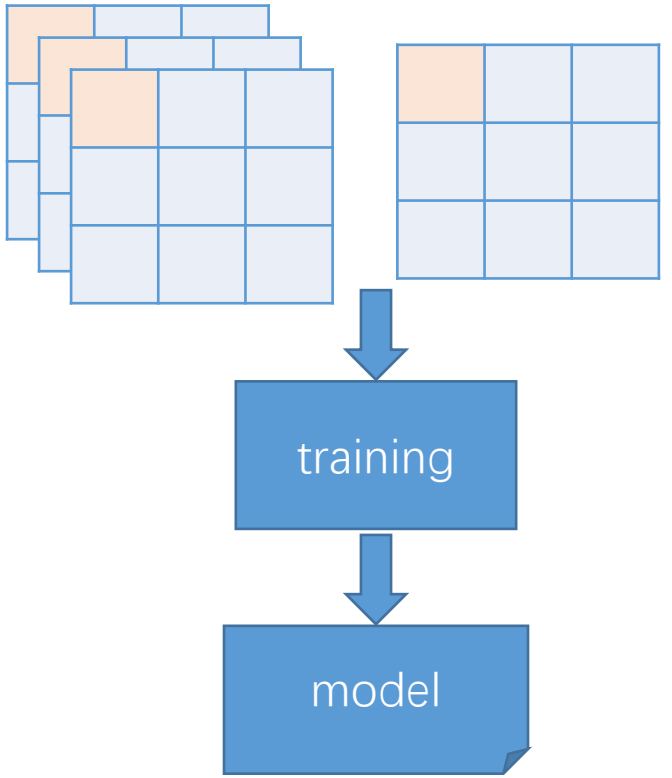
- 提出非线性预测方法，压缩粒度介于通用与专用之间
  - 以专用压缩的思想提升压缩率
    - 对于不同的图像序列训练不同的模型
    - 依据：不同样本生成的图像序列内容差别较大
  - 以通用压缩的思想加速训练过程
    - 对于同一图像序列使用单张图像训练得到的模型进行预测
    - 依据：同一样本生成的图像序列内容非常相似



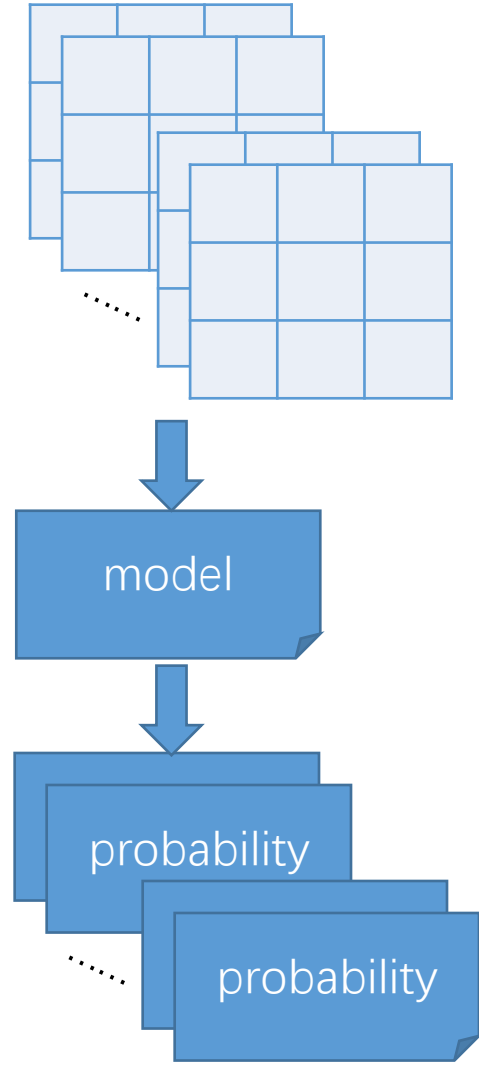


非线性预测

- 非线性预测方法
  - 训练即信息抽取
    - 训练模型学习数据中的非线性关系
    - 训练集由少量数据构造
  - 预测即数据压缩
    - 图像序列分别通过同个模型进行压缩
  - 数据集构造
    - 训练集
      - 分块构造训练数据集
      - 取图像序列中一张图像及其前timesteps张图像
    - 预测集
      - 图像序列从第timesteps+1张图像开始
      - 按照同样的方法分块构造预测集



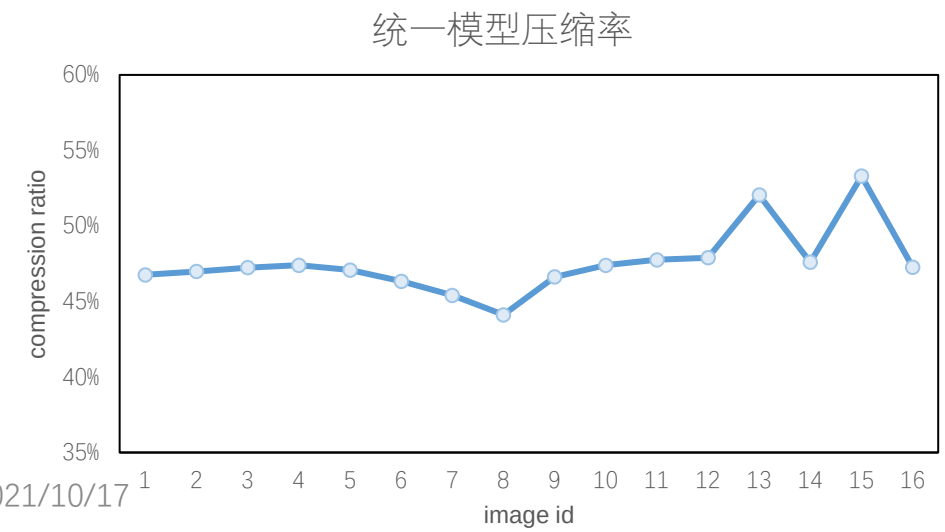
STEP1：使用少量样本完成模型训练

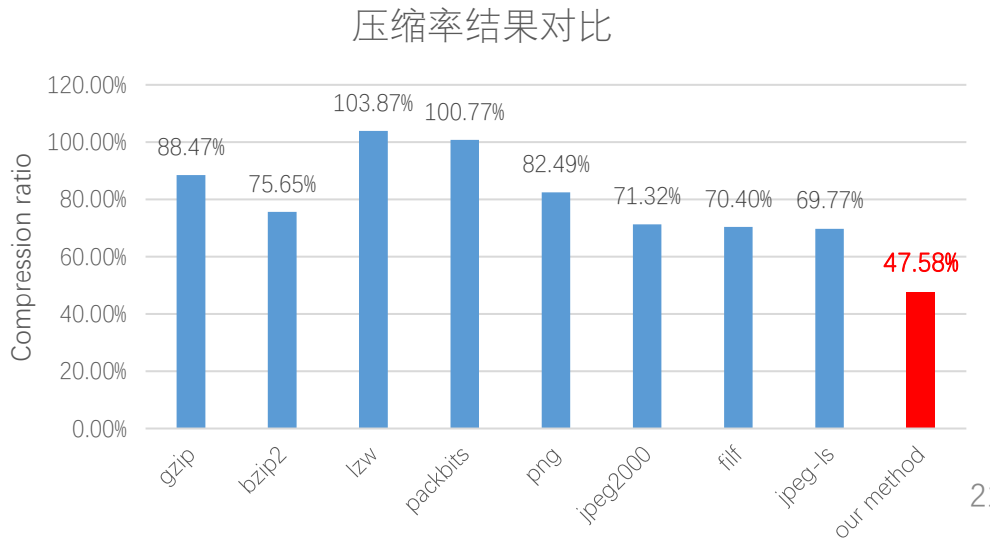


STEP2：同一样本图像通过该模型得到预测概率

非线性预测

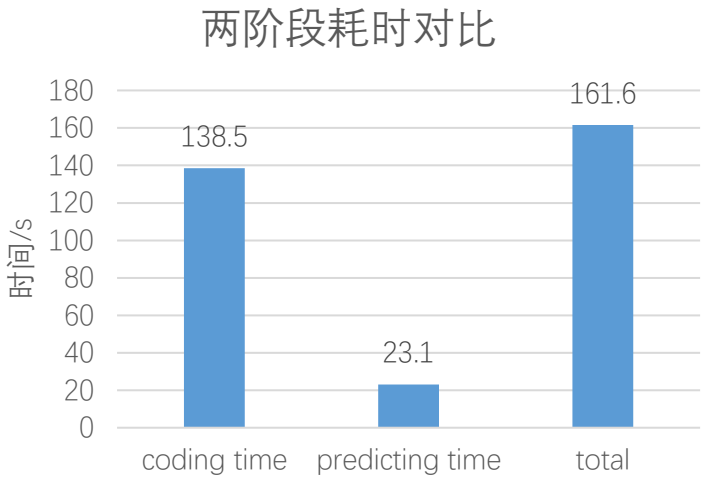
- 结果分析
  - 使用单张图像得到的模型作为统一模型压缩测试图像
    - 测试图像集为同一样本生成的图像序列每隔100张采样得到（id = 1,2,3,4,5…）
    - 时间序列构造结合时间差分
    - 结果均值在47%左右
    - 相较于常见压缩方法，压缩率提升20%
  - 可以对同一样本生成的图像序列使用单一图像训练得到的图像进行压缩





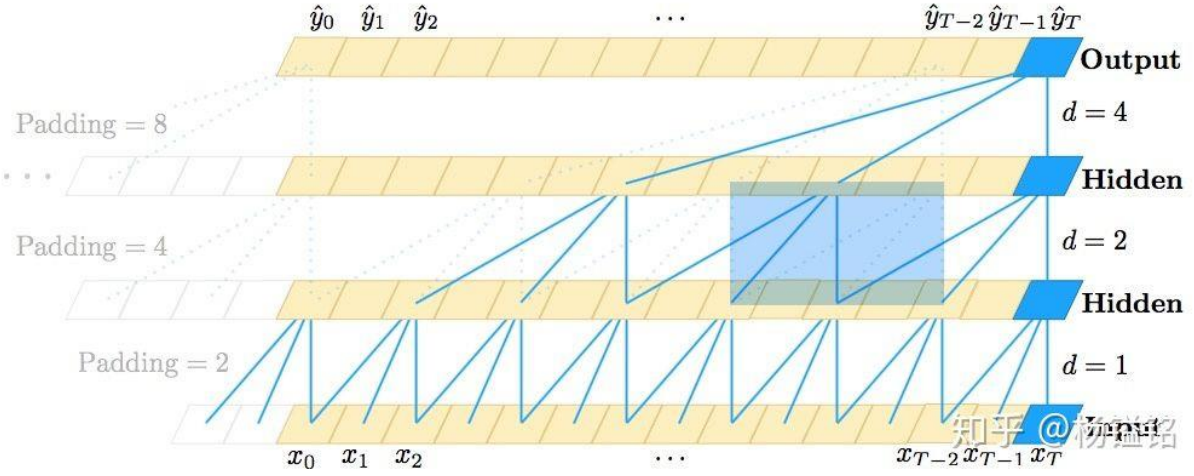
非线性预测

- 问题：压缩耗时
  - 压缩一张图像需要150s左右的时间
- 时间分析：
  - 时间主要消耗在模型预测和算数编码阶段
  - 预测耗时（12.5%）
    - Deepzip输入为一维序列，并行度较低 → 引入CNN增加并行度
  - 算术编码慢（77.5%） → 提出一种并行度高的编码方法
    - 后文编码需要等待前文编码完毕 保证压缩率优化效果
    - 单个batch的数据编码串行执行



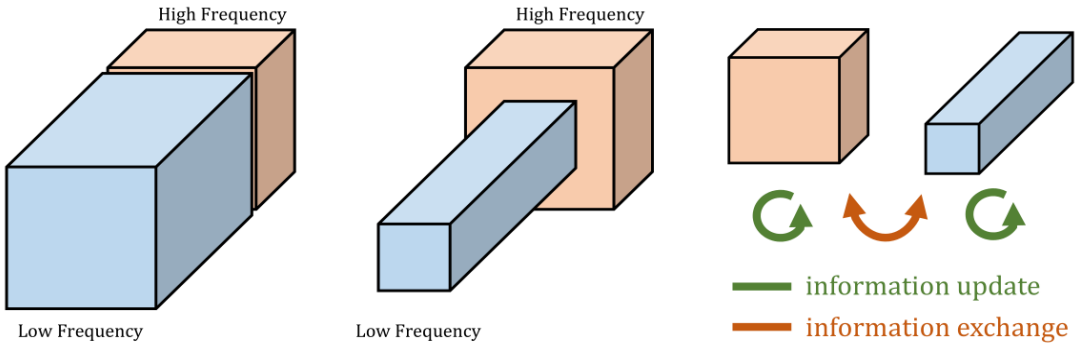
模型结构

- 引入CNN增加并行度以加速预测
  - 网络结构能够挖掘时空关系
  - 训练数据量较小，追求过拟合



TCN空洞卷积图示

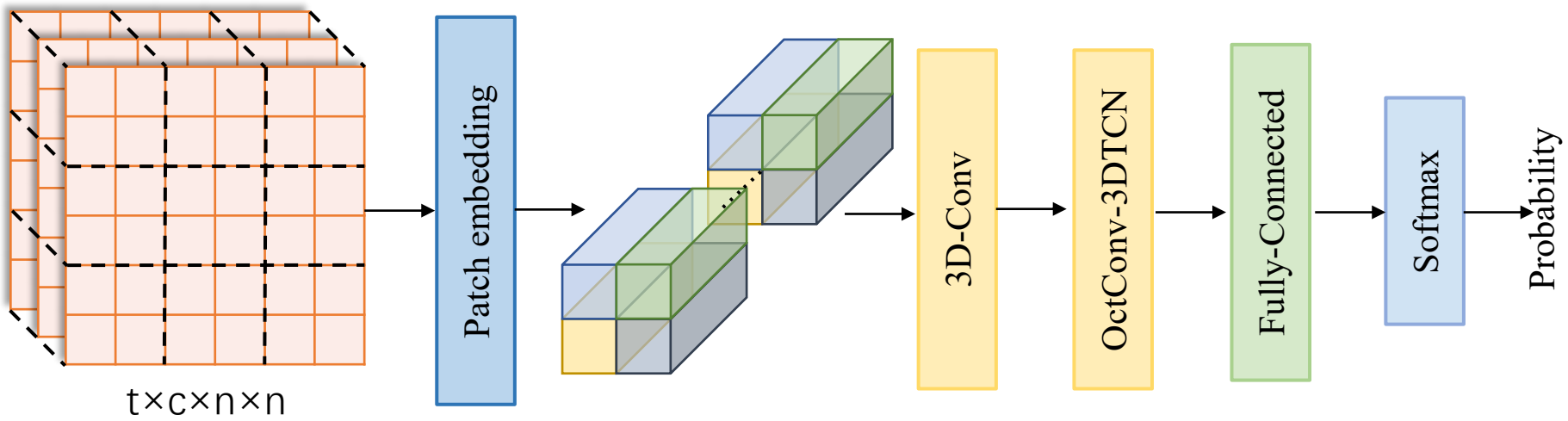
- TCN：Temporal Convolution Networks
  - 使用空洞卷积扩大上层感受野，以引入更多的历史信息
- Octave Convolution介绍
  - 将卷积层输出按空间频率进行分解
  - 平滑的低频信息存储在低分辨率张量中
  - 相同频率内部更新，不同频率间有通信



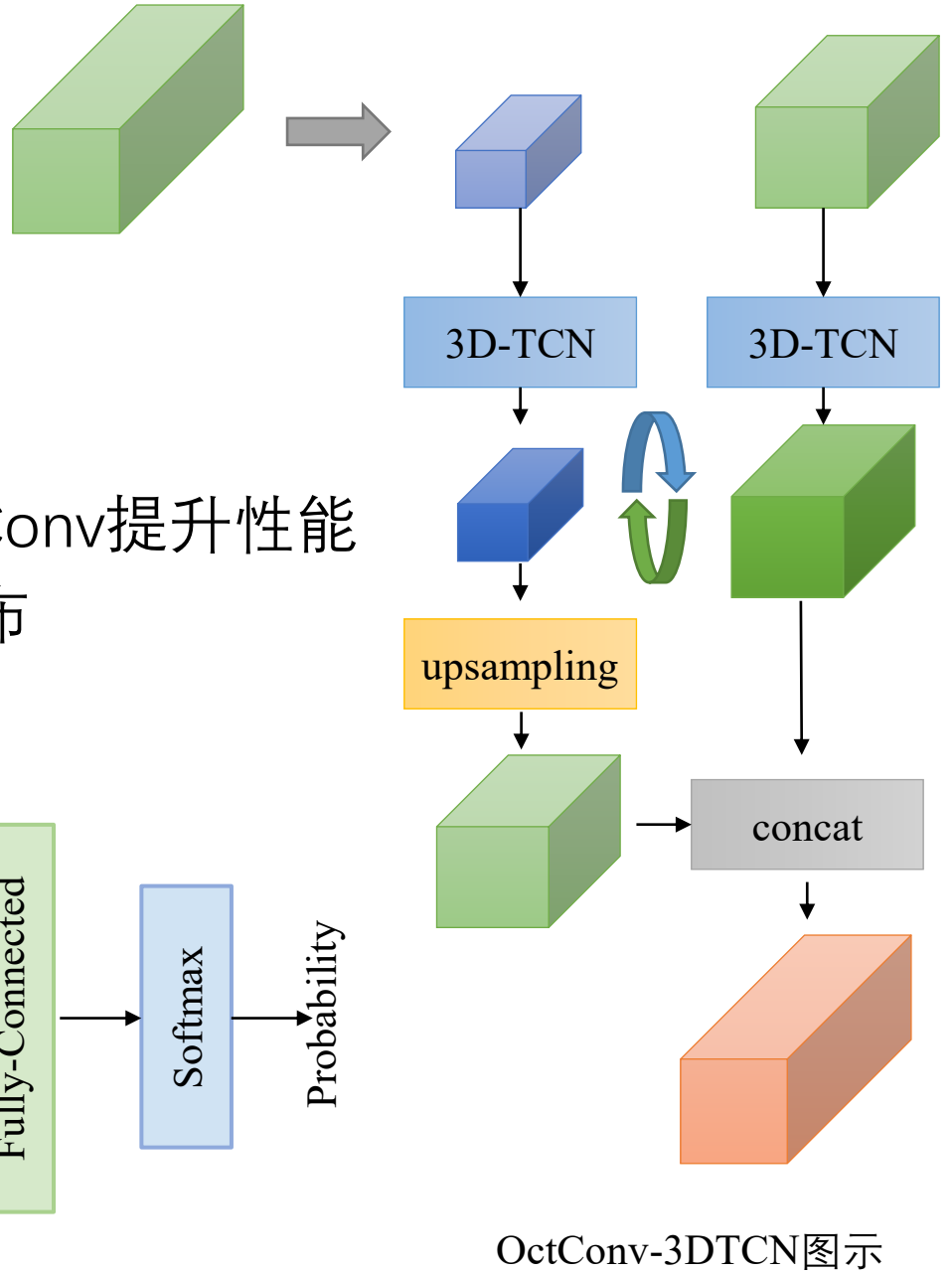
Octave Convolution图示

模型结构

- CNN-zip模型结构
  - 输入为指定大小patch
  - 首先对输入做patch embedding
  - 经过3D-TCN学习时空特征，引入Octave-Conv提升性能
  - 通过FC+softmax得到编码时使用的概率分布



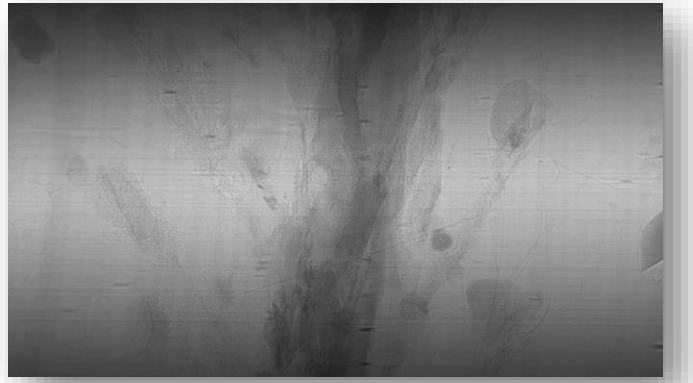
CNN-zip模型结构图示



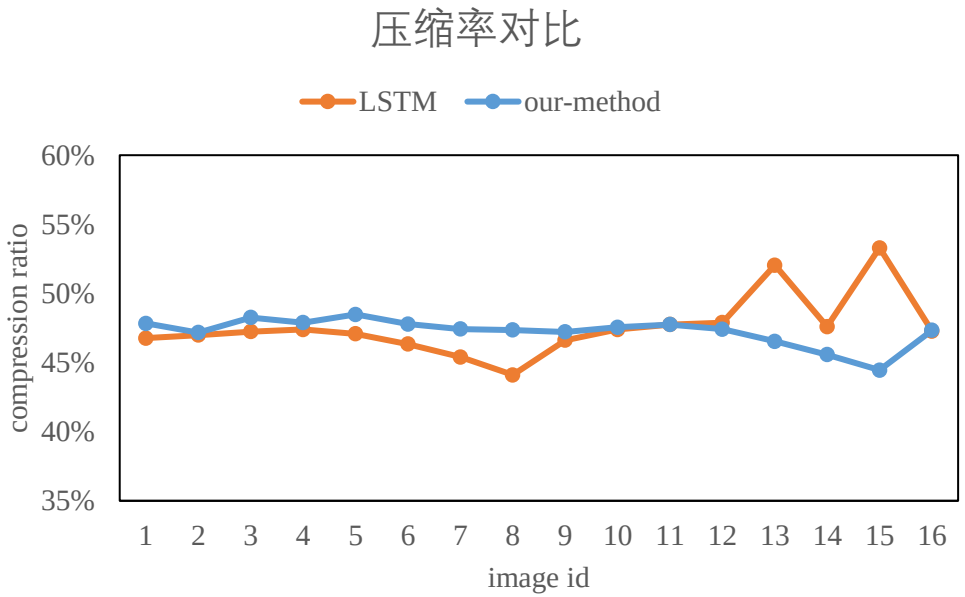
OctConv-3DTCN图示

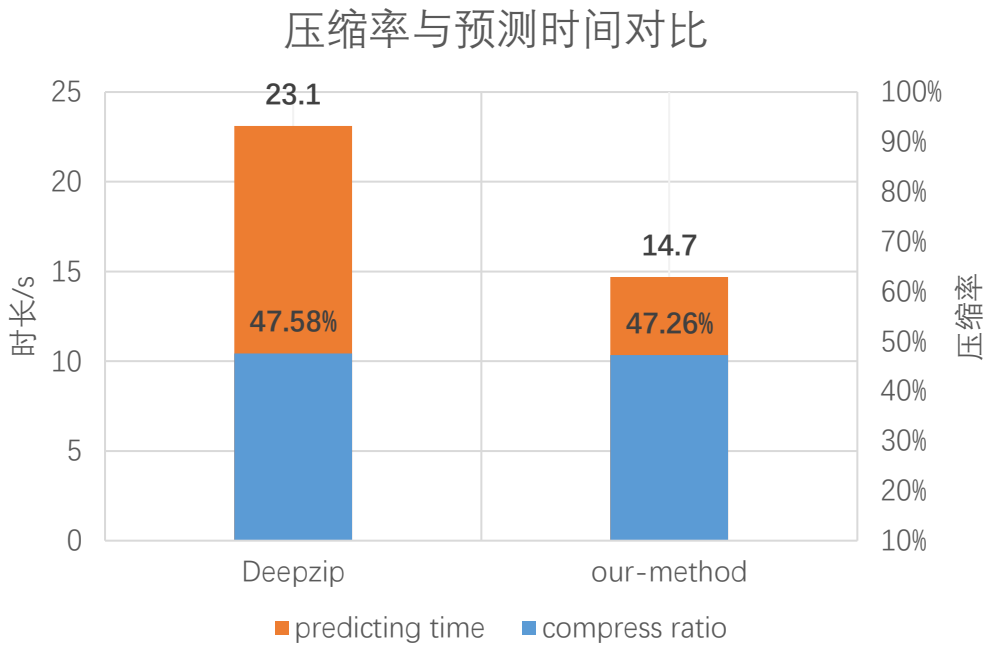
模型结构

- 模型对比
  - 数据：1622×1200×2048
  - patch size=32×32 timesteps = 3



测试数据图示

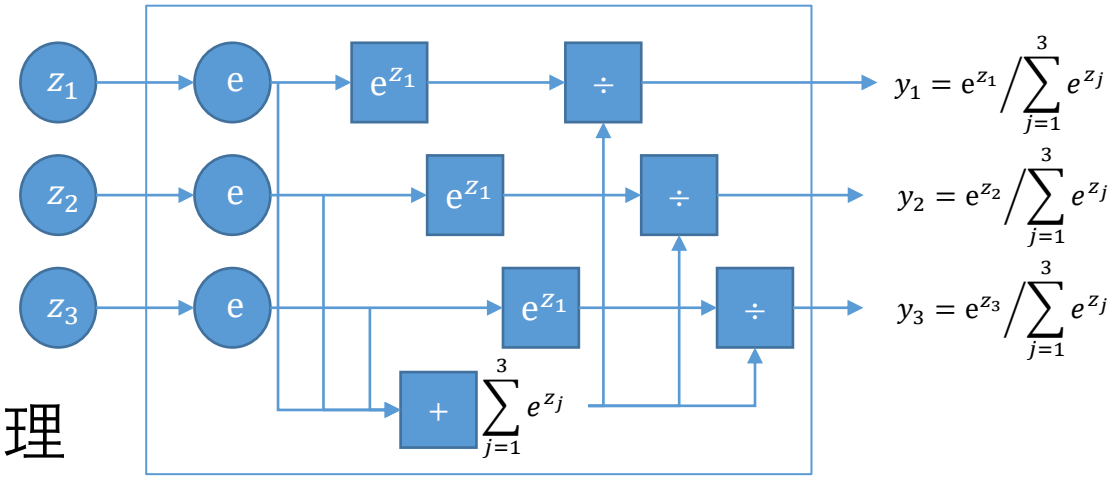




- 结论：
  - 对于单张图片压缩，CNN-zip可以在保证压缩率提升的同时，节省**36%↓**的预测时长

编码方法

- 动机：算术编码慢，并行度低
- softmax层
  - 将神经网络得到的多个值进行归一化处理
  - 得到的值在[0,1]之间
  - 可以将结果看作是概率
- 从softmax层输出可以得到什么（离散数据）
  - 不同任务可以获取不同信息



Softmax计算过程图示

Deepzip → 字典数概率分布P

直接使用概率

手写体识别 → max(P)

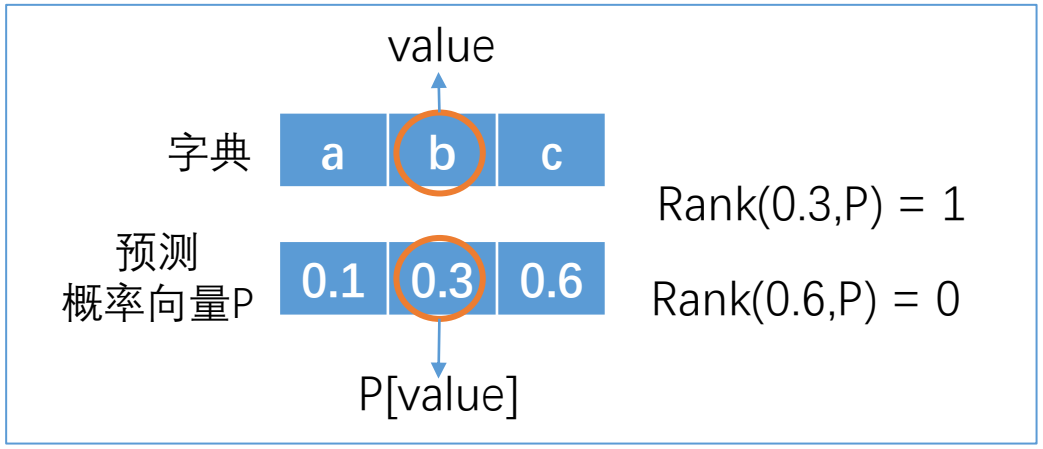
将特定的值或标签分配到单个数据点

推荐算法 → sort(P)

对列表进行最佳排序

编码方法

- 提出概率距离 →Rank(P[value], P)
  - 概率距离：真实值对应的预测概率值在所有预测概率向量中的排序位置
    - Value：真实值
    - P：当前元素取字典中每一个值的概率分布，由神经网络预测得到
    - P[value]：真实值对应的概率
    - Rank(i,seq)：i在p序列中的排序位置（由大到小排序）

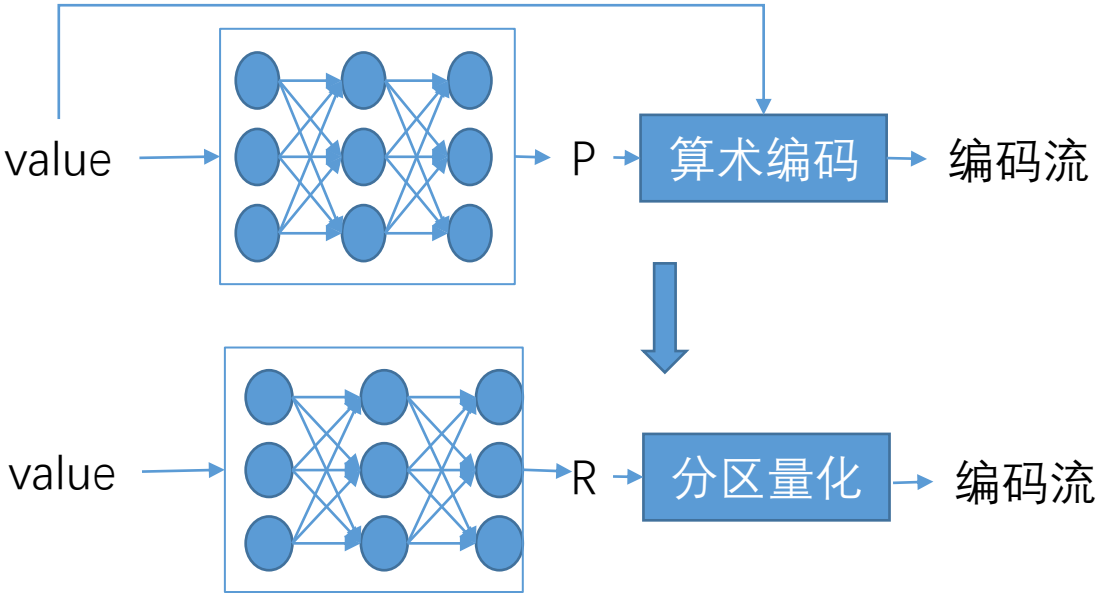


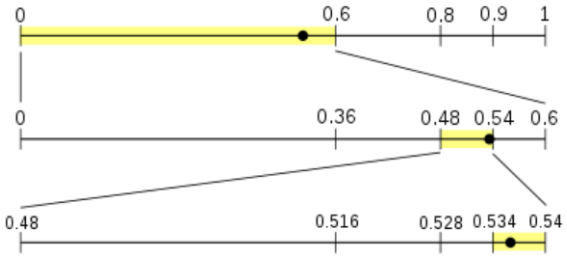
Rank计算图示

推测：对于大字典数数据来说，如果神经网络预测的结果比较好，能够保证大部分真实值对应的概率距离在TOP-K，同时K的位数小于当前数据的位数，则能够达到压缩的效果。

编码方法

- 将神经网络的直接输出作为量化流
- 量化后作为编码流
  - 应用概率距离：Rank over Probability
- 优点
  - 计算过程简单
    - 比较大小
  - 并行度优于算数编码，速度快
    - 算术编码中编码当前元素需要等待上一个元素编码完成
    - 不同元素之间概率距离的计算相互独立
  - 可与其他编码方法结合，进一步节省存储空间
    - 量化后结果可以结合算数编码等编码方法达到进一步压缩的目的





算数编码图示

编码方法

- 测试与结果分析
  - softmax层结果直接通过Rank计算得到量化流，分区量化替代算数编码
  - 其余测试条件不变，以单张图像训练模型，压缩测试数据集
  - 测试数据集构建同样为同一样本的图像序列中隔100张采样得到

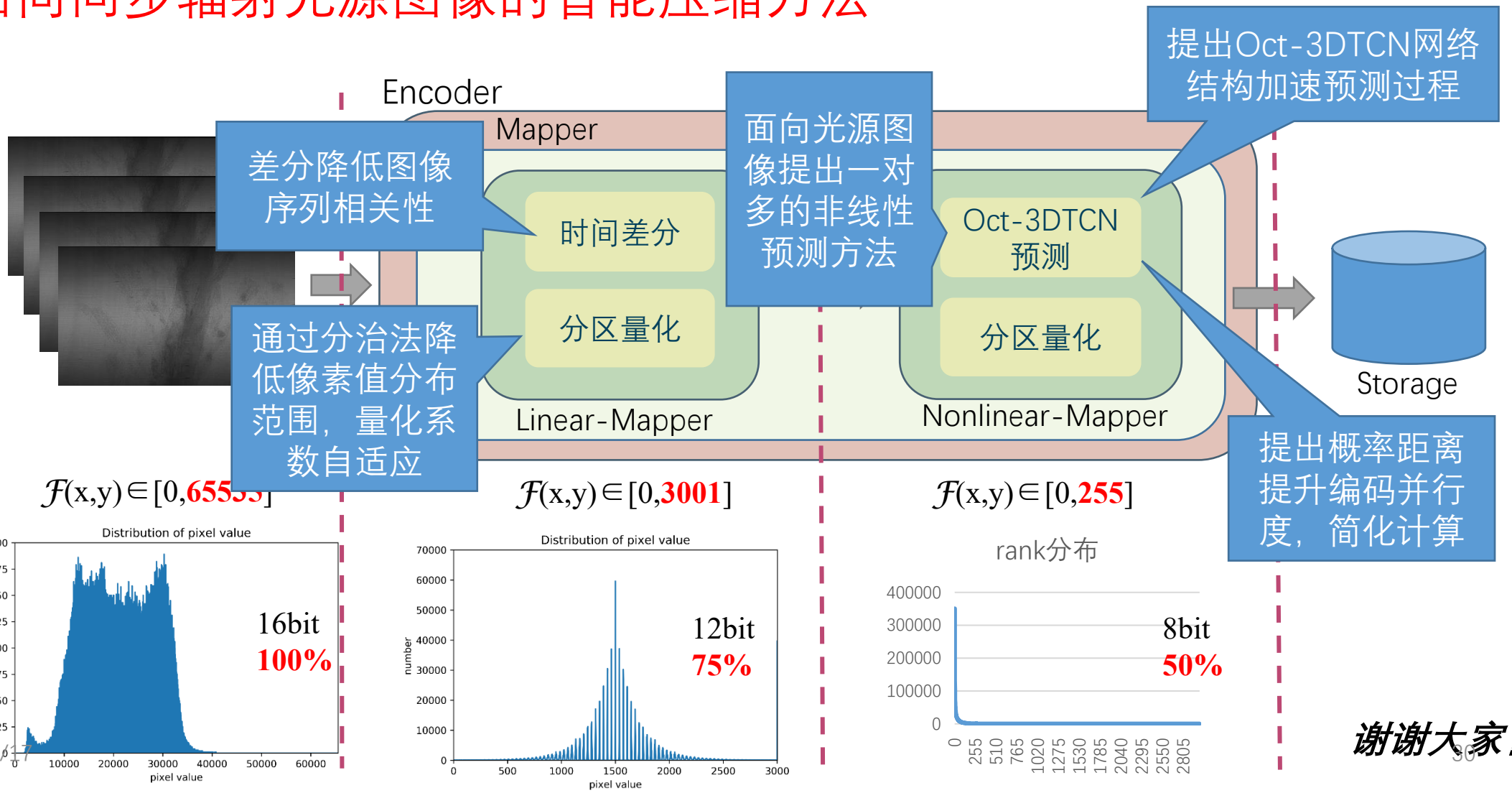
- 分区量化
- 99.53% →[0,255]→8bit→50%（8bit/16bit）
- 0.47% -> 保留
- 总压缩结果**52.47%**

  - 以概率距离量化流为最终压缩流和原算术编码压缩结果差距不大（5%）
- 下一步计划
  - 编码在FPGA上实现加速

总结

$\mathcal{F}(x,y)$为图像在(x,y)位置的像素值
具体数值以恐龙尾巴数据及为例

- 面向同步辐射光源图像的智能压缩方法



谢谢大家！