

零基础入门旷视天元MegEngine

MEGVII 旷视

模型构建和训练进阶II：物体检测

讲师：王枫

天元开发者交流群



扫一扫二维码，加入社群。

欢迎加入“天元开发者交流群”



| 零基础入门旷视天元MegEngine

课程大纲:

- 检测任务介绍及前置概念
- Faster R-CNN源码学习
- Sublinear memory
- 课程作业 与 Q&A



- 1 检测任务介绍和前置概念
- 2 Faster R-CNN源码学习
- 3 Sublinear memory
- 4 Q&A 与 课程作业



检测任务介绍和前置概念

什么是物体检测？



分类是告诉你这个图是猫，而检测则是告诉你这个猫在哪里

评价分类效果可以用准确率进行衡量，检测这种任务应该咋办？



检测任务介绍和前置概念

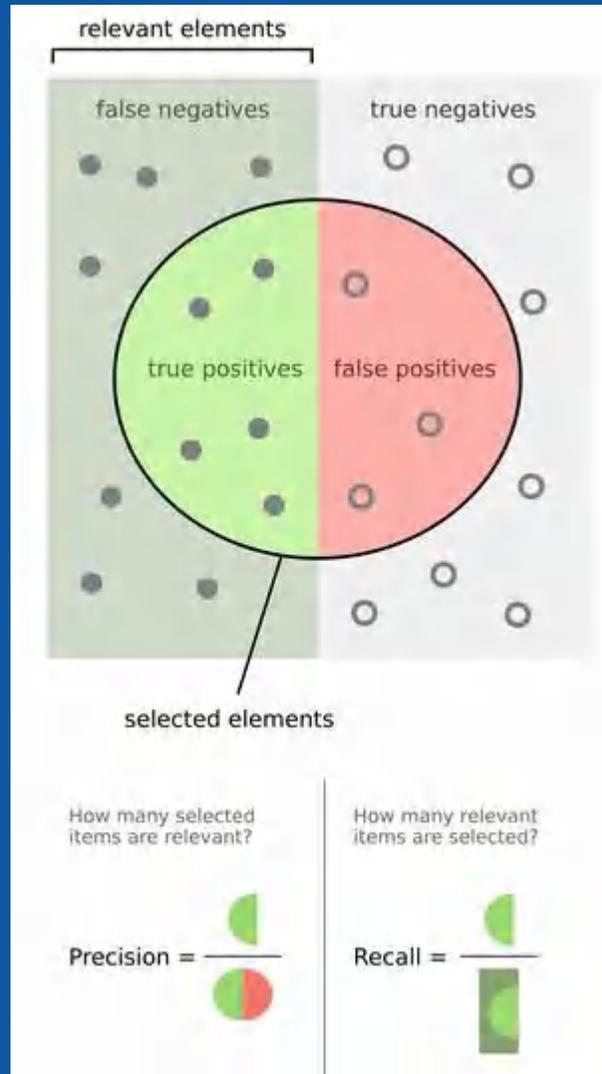
两个基本概念：precision 和 recall

$P = TP / (TP + FP)$ ，提高P值靠杀FP

$R = TP / (TP + FN)$ ，提高R值靠杀FN

$F1 = 2PR / (P + R)$ ，也就是P和R的调和平均

实际F1这类指标在如文字检测这类task中会使用的比较多

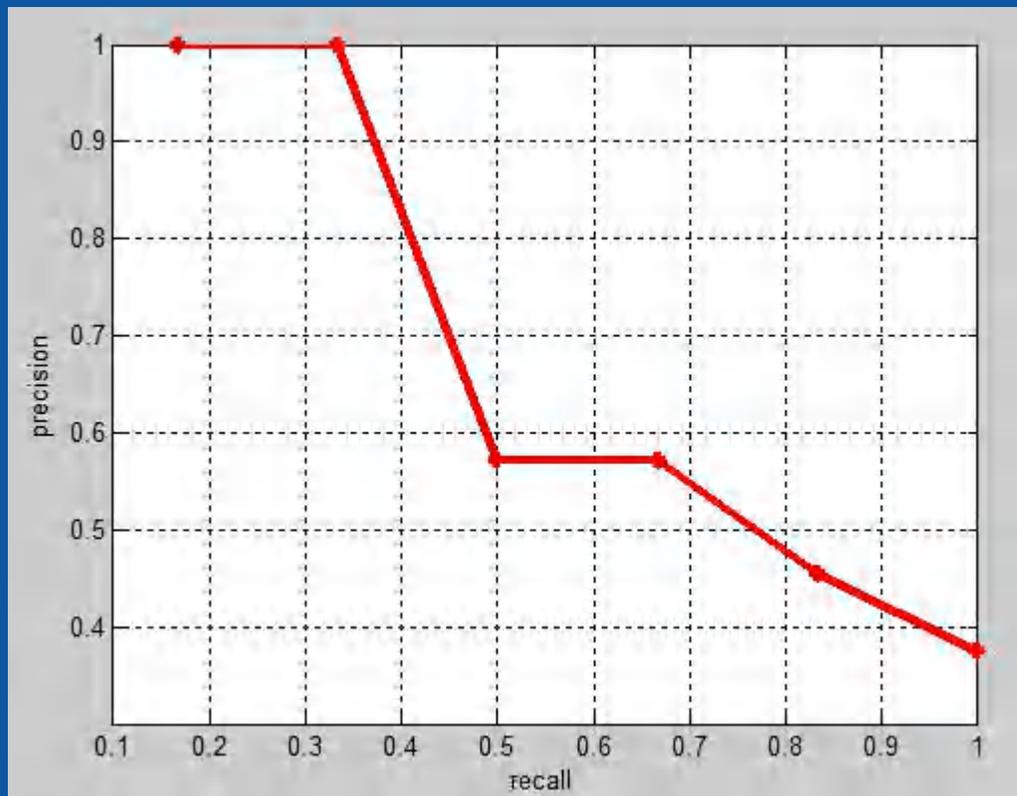


检测任务介绍和前置概念

评价指标AP \rightarrow mAP \rightarrow mmAP (真的不是进化链)

AP值 : P-R曲线围成的面积 (P-R曲线积分)

mAP : m表示mean, mAP是不同类别的AP结果的均值



检测任务介绍和前置概念

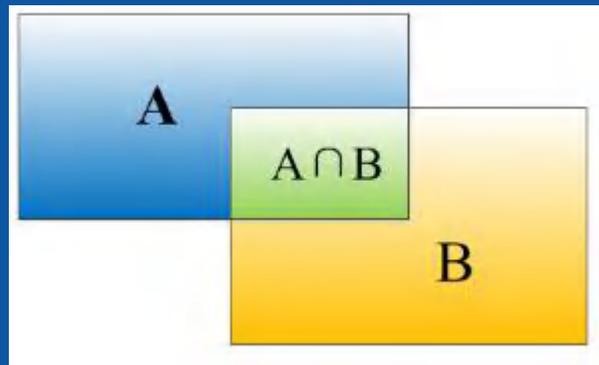
mmAP : 不同IoU阈值下的mAP的结果的均值

Wait! IoU是啥? 好次嘛?

IoU(Intersection over union)

leetcode上有这个题目 (题号223, Rectangle Area)

温馨提醒: 不要写过多的if条件哦~



检测任务介绍和前置概念

检测的评价指标有了，那么该怎么产生检测结果呢？

两个流派：one-stage和two-stage

One-stage：直接产生分类结果

Two-stage：首先进行前景背景的分类，然后利用前景的特征产生分类结果

常用的one-stage算法：RetinaNet, FCOS, Yolo等

常用的two-stage算法：R-CNN系列

问题：如果RPN不是产生前景背景结果，而是产生所有类别的结果，算是one-stage方法吗？



数据集介绍: VOC [网页](#)

The PASCAL Visual Object Classes Homepage



The PASCAL VOC project:

- Provides standardised image data sets for object class recognition
- Provides a common set of tools for accessing the data sets and annotations
- Enables evaluation and comparison of different methods
- Ran challenges evaluating performance on object class recognition (from 2005-2012, now finished)

Pascal VOC data sets

Data sets from the VOC challenges are available through the challenge links below, and evaluation of new methods on these data sets can be achieved through the [PASCAL VOC Evaluation Server](#). The evaluation server will remain active even though the challenges have now finished.



检测任务介绍和前置概念

数据集介绍: COCO [网页](#)

COCO标注一共有12w多张, 涵盖多个任务

What is COCO?



COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Collaborators

Tsung-Yi Lin Google Brain
 Genevieve Patterson MSR, Trash TV
 Matteo R. Ronchi Caltech
 Yin Cui Google
 Michael Maire TTI-Chicago
 Serge Belongie Cornell Tech
 Lubomir Bourdev WaveOne, Inc.
 Ross Girshick FAIR
 James Hays Georgia Tech
 Pietro Perona Caltech
 Deva Ramanan CMU
 Larry Zitnick FAIR
 Piotr Dollár FAIR

Sponsors

 CVDF
 Microsoft
 facebook
 Mighty Ai



检测任务介绍和前置概念

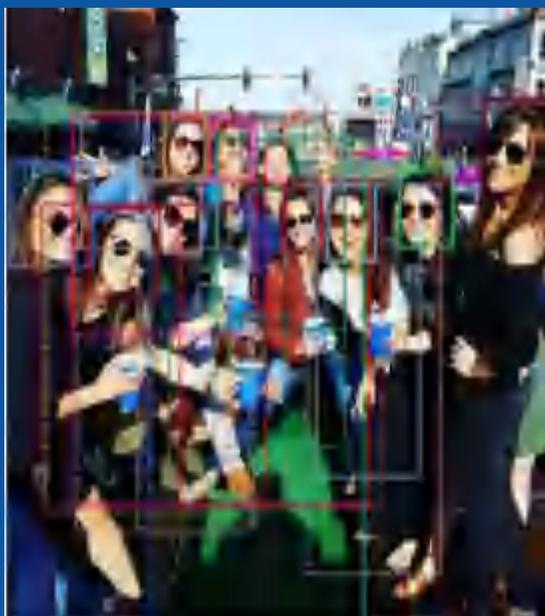
数据集介绍：Objects365 [网页](#)

一共365类物体，包含63w图像的标注，目前业内已知完全标注的最大的检测数据集



其他任务数据集介绍

CrowdHuman



WiderFace



ICDAR

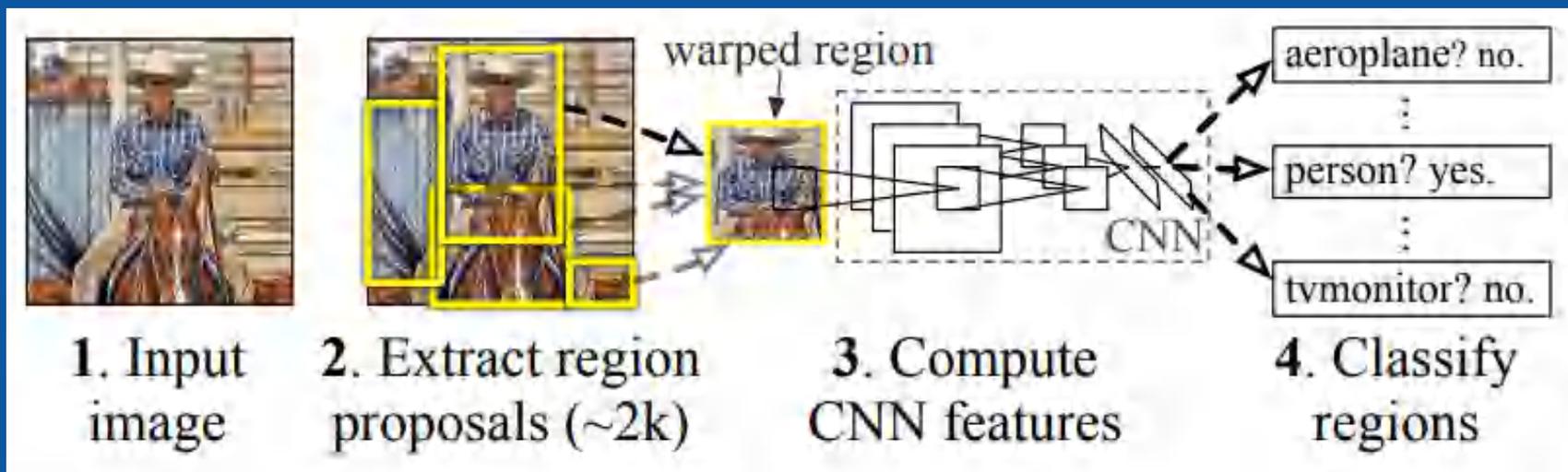


- 1 检测任务介绍和前置概念
- 2 **Faster R-CNN源码学习**
- 3 Sublinear memory
- 4 Q&A 与 课程作业



物体检测的Pipeline

1. 产生候选区域 (Selective Search -> Anchor)
2. 提取候选区域的特征 (RoI pooling -> RoI Align)
3. 对候选区域特征进行分类 (SVM -> FC)



输入图像处理流程

对于COCO数据集，图像的预处理非常简单，一共有两步：

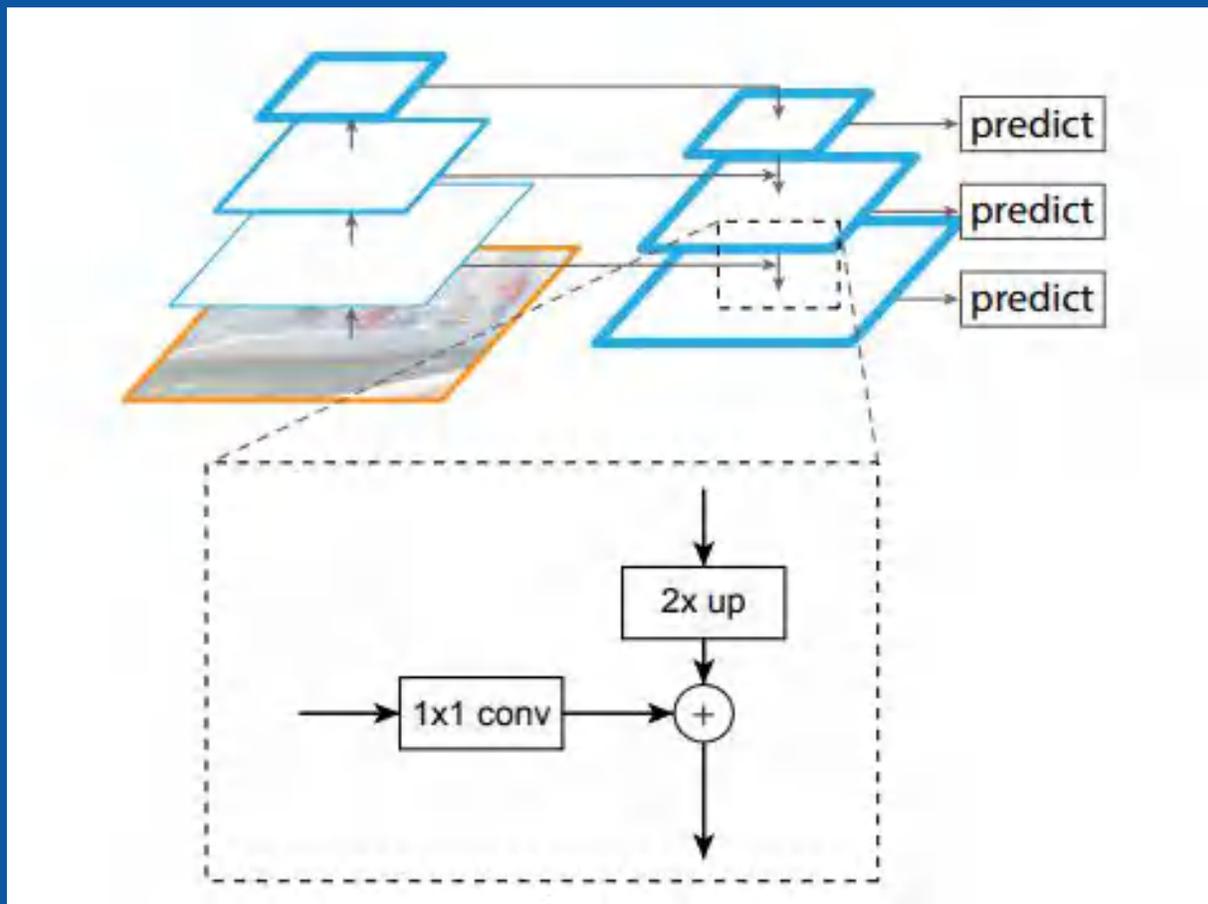
- (1)将输入图像resize成短边不超过800，长边不超过1333的图像
- (2)对于输入图像进行随机的水平翻转，翻转概率为0.5
- (3)转成 (C, H, W) 形状的输出图像

在实际进入检测网络之前还需要进行进一步的预处理：

- (1) 将数据图像的宽度和高度pad成32的倍数
- (2) normalization (减去 mean 后除以 std)



FPN结构



FPN的code参考: [FPN code](#)

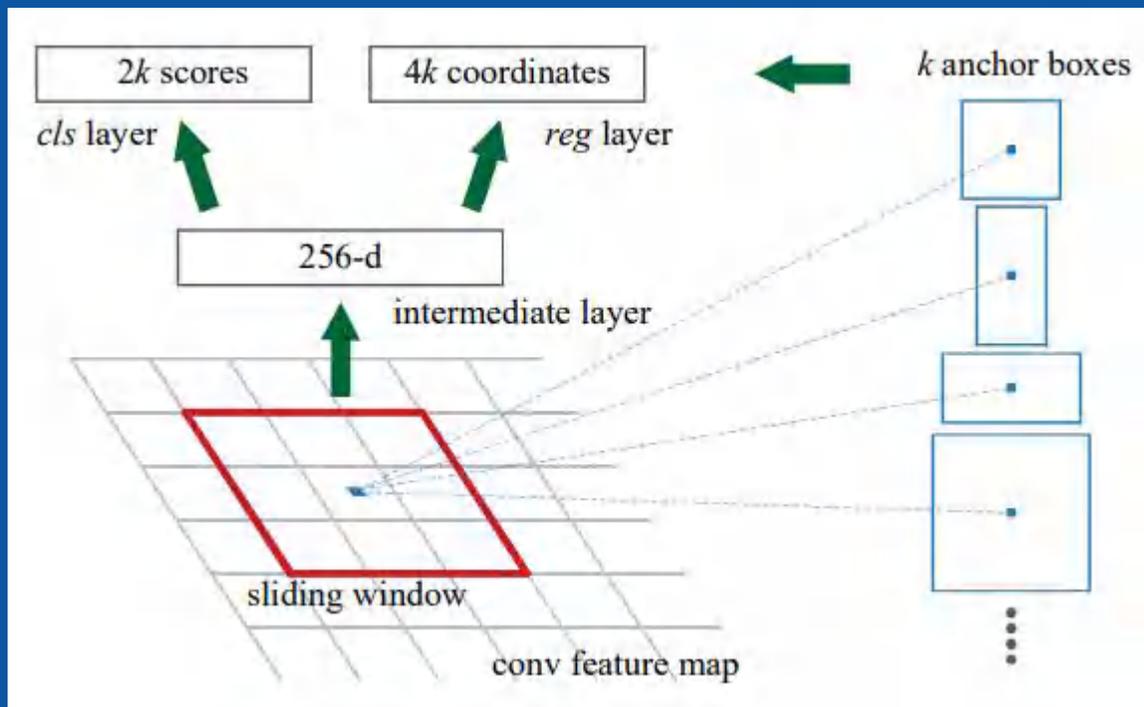


RPN (Region Proposal Network)

FPN产生了P2 - P6 的特征，RPN的任务就是需要针对FPN输出的特征图产生对应的先验框，这些先验框我们称为anchor

Anchor的本质：网络对于输入图像的先验

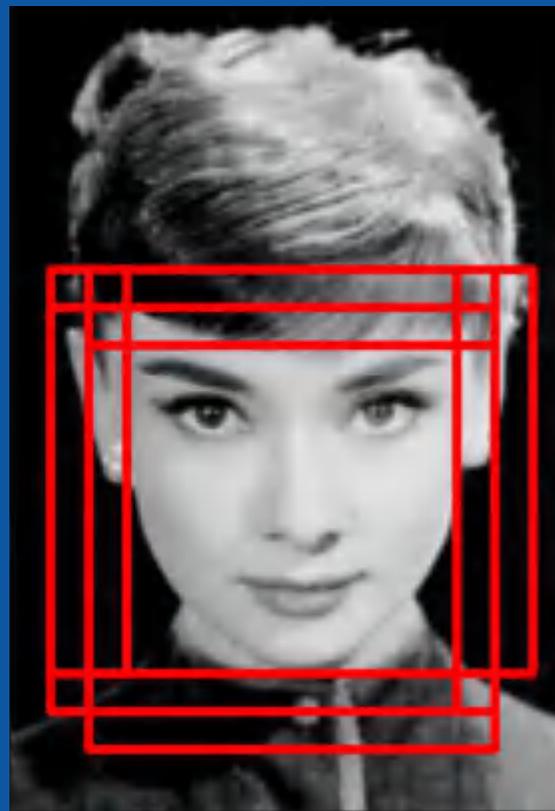
Proposals = Anchor + offset



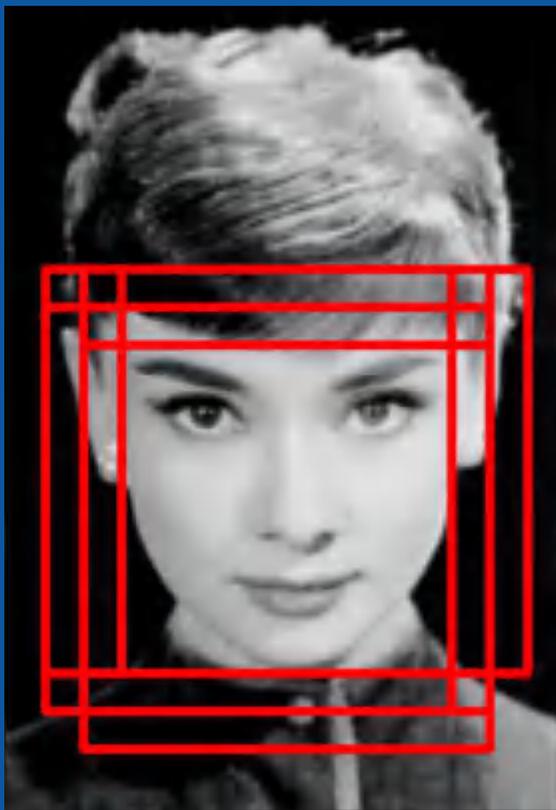
RPN产生了一堆候选框之后，该怎样从中选择需要的框进行特征提取？

RPN实际产生的proposals（候选区域）数量非常多，而且很多后续区域是高度重叠的，如果所有特征提取特征，显然是一个低效的行为（而且显存也吃不消）。

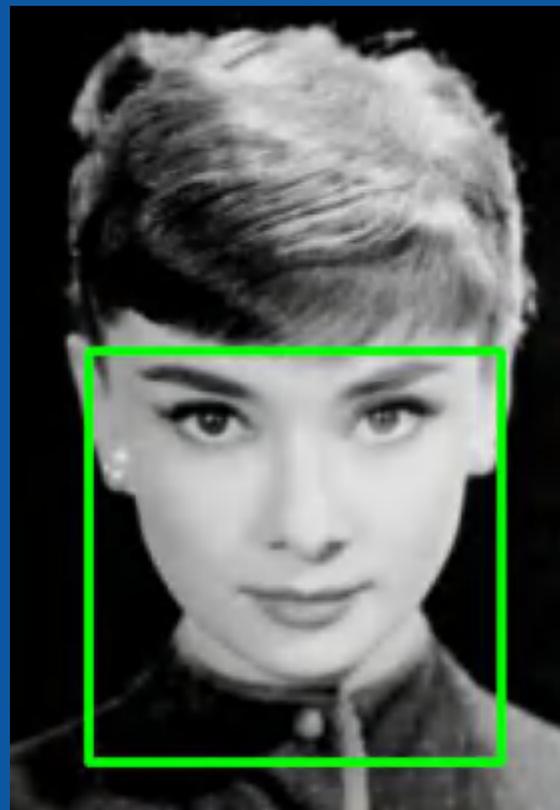
那么怎么样才能大海捞针，得到想要的候选区域呢？



NMS算法就是为了处理冗余框而生的!



before



after



NMS算法流程

输入: box list(每个box包含自身坐标和对应score), 阈值T

输出: keep list

(1)选取当前最高分的框A, 将box list所有与A的IoU大于阈值T的框删除掉

(2)将A放入 keep list中, 重复步骤(1), 直到box list为空

(3)返回keep list

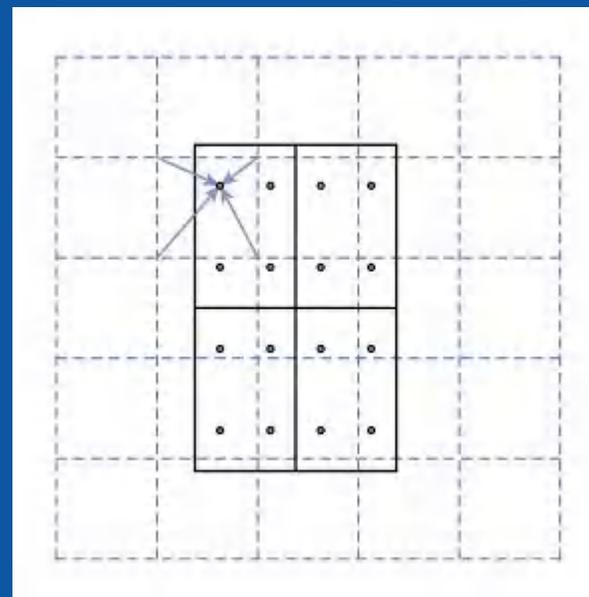
Cpu版本的NMS代码参考: [nms code](#)



Roi Align算子

Roi: Region of Interest, 感兴趣区域

无论感兴趣区域有多大，经过Roi Align之后都会统一成为固定大小的输出尺寸



为了方便有一个直观认识，可以把Roi Align理解为 crop + resize 两个操作的集合



相比其它深度学习框架，MegEngine中提供了RoI Align的算子

如果你有兴趣了解cuda实现，可以参考：[code](#)

相比外部一些框架的roi align，MegEngine中修复了一个resize相关的bug：[code](#)

关于为什么要加0.5 offset的问题，核心原因是为了将图像的中心对齐，opencv中的resize也有类似的写法 [code](#)



RCNN与后处理

RoI Align产生了大小为7x7的特征，在RCNN中会使用两层fc对这些前景特征进行分类和回归得到了全部的框和对应得分之后，再次使用NMS算法就可以得到检测网络的输出结果。



- 1 检测任务介绍和前置概念
- 2 Faster R-CNN源码学习
- 3 Sublinear memory
- 4 Q&A 与 课程作业



Sublinear Memory是啥？为啥需要sublinear memory？

Sublinear是一种用于节省显存的技术，也有人称这门技术为checkpointed backpropagation.

这个技术最常用的场景就是训练大模型（而你因为没钱买显存足够大的显卡而发愁 Orz）

当然，天下没有免费的午餐，你需要为此支付的代价就是训练的速度会变慢（不过好在你虽然没钱买最好的显卡，但是人却不缺时间）

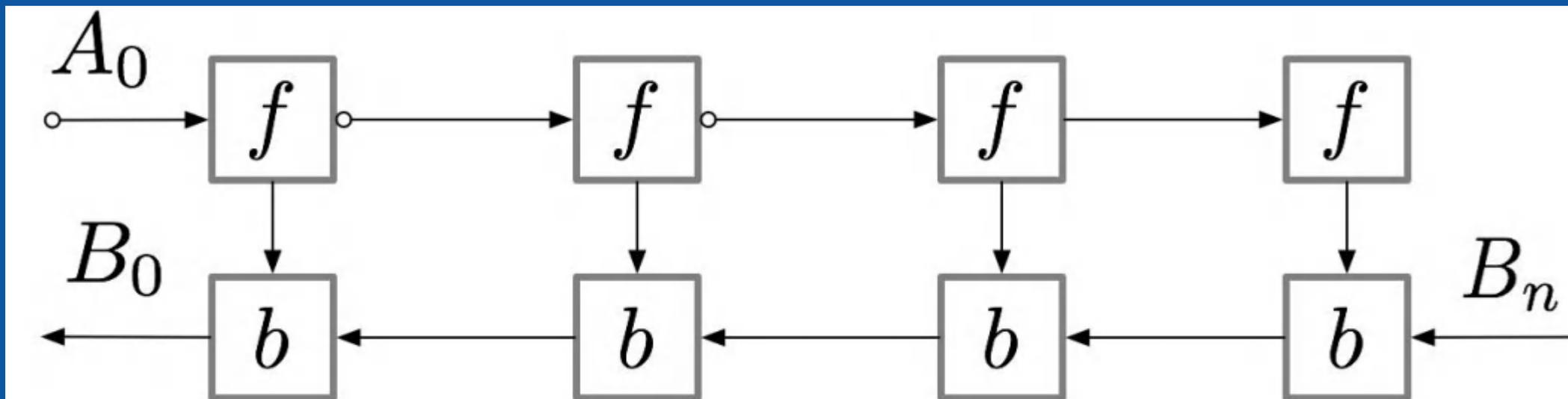
Github上有类似的[repo](#)：gradient-checkpointing，有时间可以学习一下



原理解析

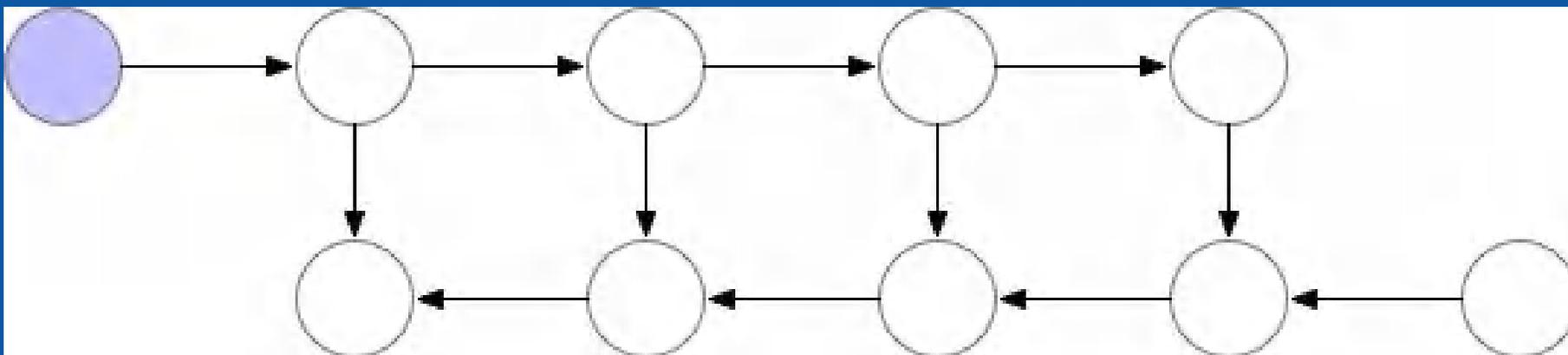
我们用下图表示前向过程：f表示forward，b表示backward

A_0 是前向过程input， B_n 是反向过程input



原理解析

正常的前向和反向计算过程如下图所示：

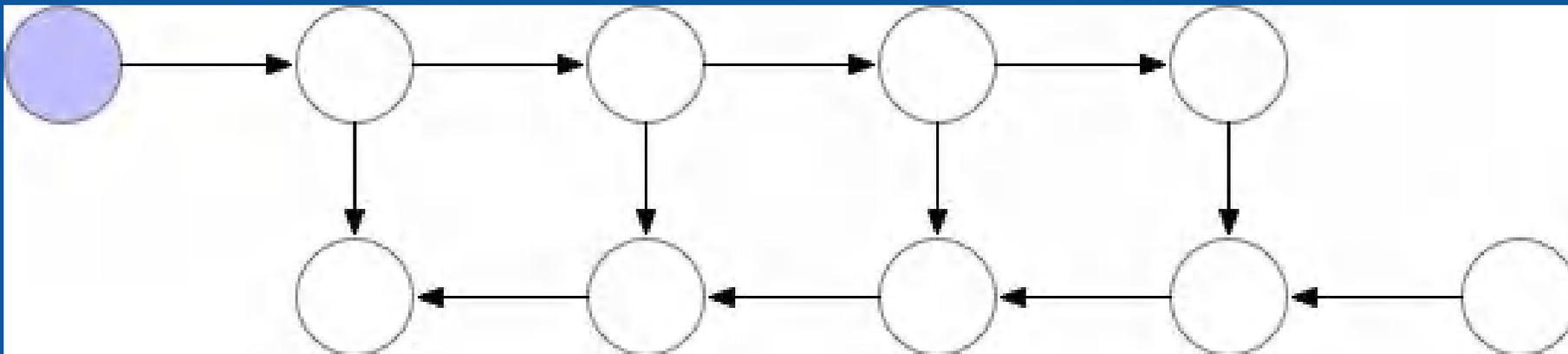


可以看出，如果Network有n层，正常的神经网络更新是 $O(n)$ 开销的算法，前向 $O(1)$ 次，所有的节点都只被训练一次



原理解析

正常的前向和反向计算过程如下图所示：



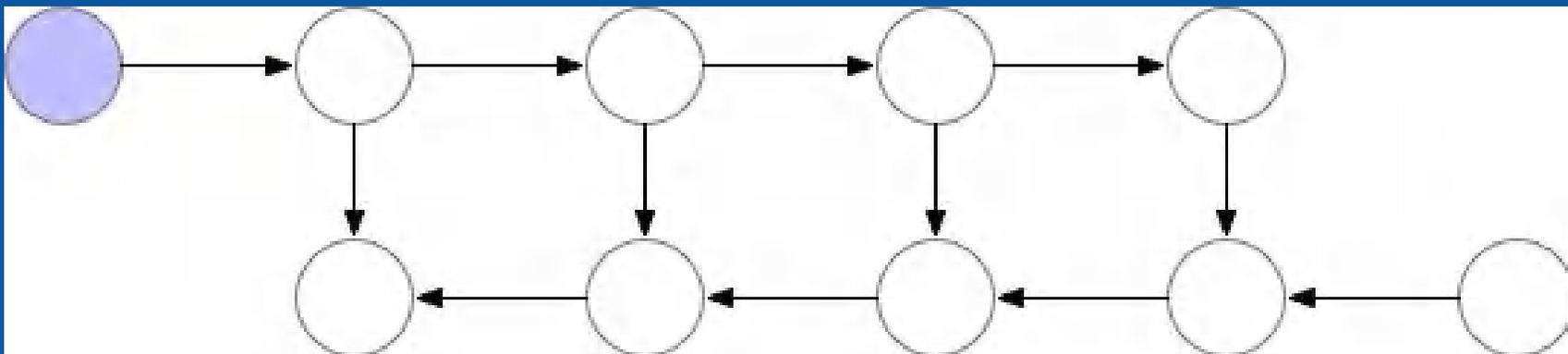
考虑如果网络相对层数很深，不支持把所有中间结果都存下来，如果想要节省显存，把开销降下来，有啥可能的思路吗？



一个简单的节省显存的思路

只要有输入的tensor，所有中间结果都可以通过forward过程求出来，通过运算多次求解，

如下图所示：



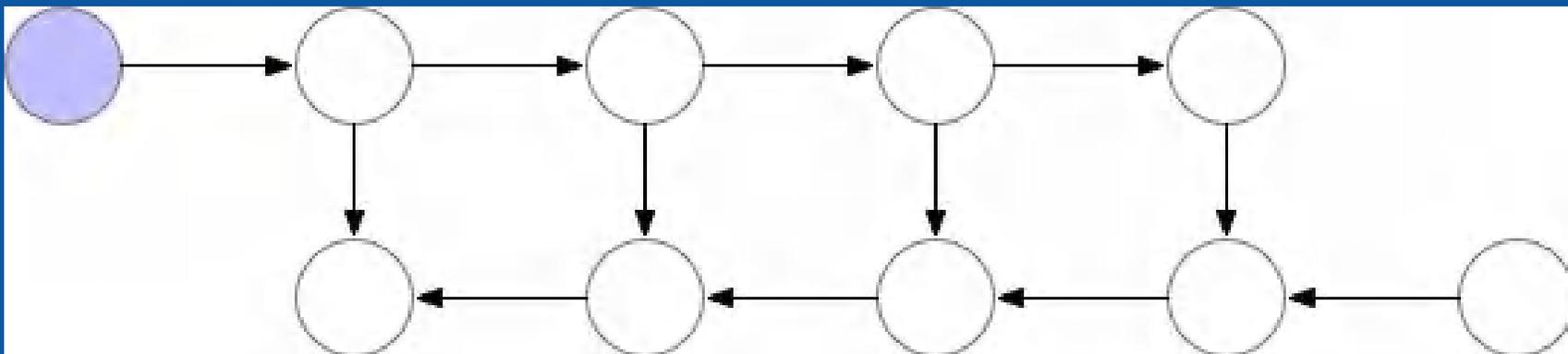
对于一个层数为 n 的神经网络，显存占用可以到 $O(1)$ ，但是前向了 $O(n^2)$ 次，单个节点最多会被额外计算 $O(n)$ 次



一个简单的节省显存的思路

只要有输入的tensor，所有中间结果都可以通过forward过程求出来，通过运算多次求解，

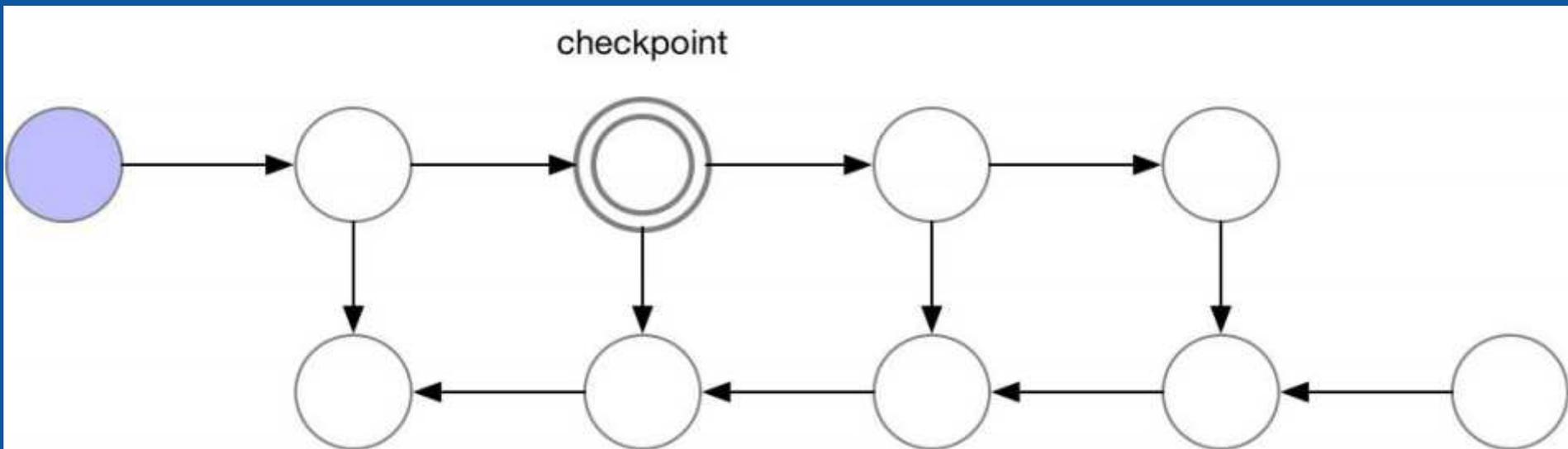
如下图所示：



通过图示我们看出，很多中间节点完全可以存下来作为checkpoint，没有必要只存一个输入



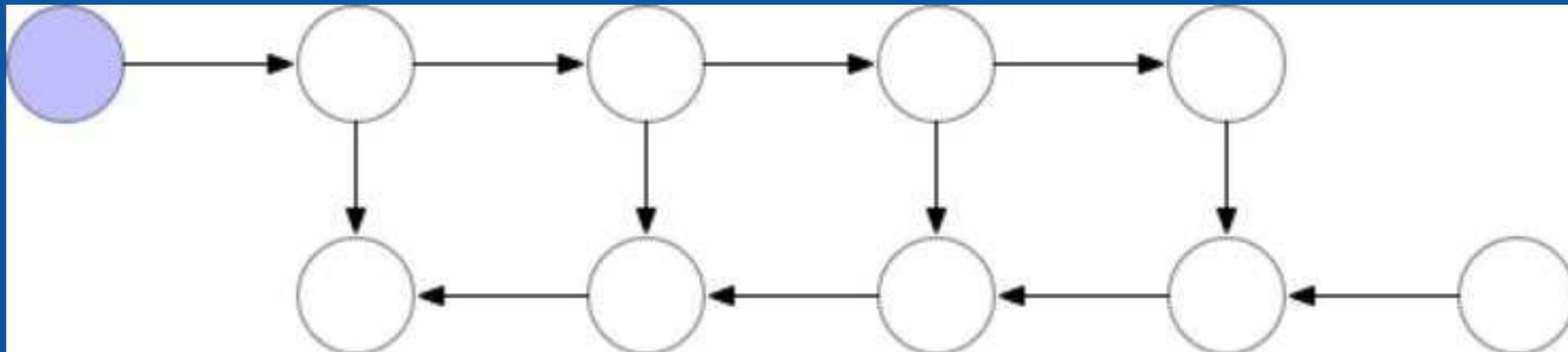
新的优化思路就这么诞生了：在某一个节点存一个checkpoint



这是优化的的一小步，但却是显存的一大步！！



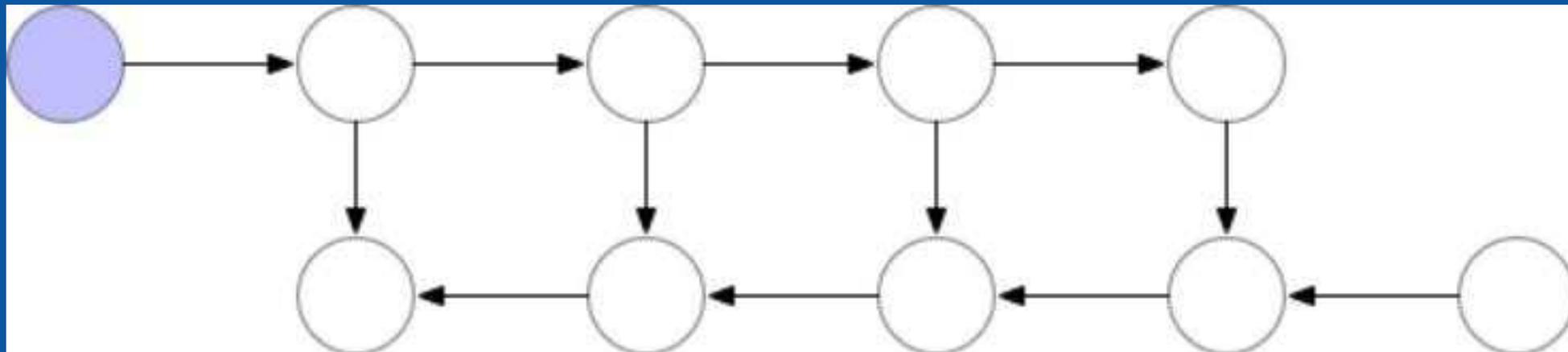
有了checkpoint的思路之后，前向过程如下图所示：



- * 首先，进行一次前向过程，每隔 \sqrt{n} 取一个节点作为一个checkpoint
- * 在backward的时候，通过checkpoint计算出中间 \sqrt{n} 个前向的结果，进行对应的更新



有了checkpoint的思路之后，前向过程如下图所示：



对于一个层数为 n 的神经网络，显存占用为 $O(\sqrt{n})$ ，前向次数为 $O(\sqrt{n})$ ，每个节点最多会被额外计算1次



那么在哪里才能买（用）到这么好的功能呢？

天元MegEngine本身就支持Sublinear Memory，且使用方法非常简单

参见此处的文档：[文档](#)

```
代码示例： from megengine.jit import trace, SublinearMemoryConfig
              config = SublinearMemoryConfig()
```

```
@trace(symbolic=True, sublinear_memory_config=config)
```

```
def train_func(data, label, *, net, optimizer):
```

```
    ...
```



- 1 检测任务介绍和前置概念
- 2 Faster R-CNN源码学习
- 3 Sublinear memory
- 4 课程作业 与 Q&A



检测任务介绍和前置概念

课程作业：

- (1) 熟悉MegEngine Hub的使用
- (2) 随意上传一张拍摄的图像，使用Hub中的Faster-RCNN或者Faster R-CNN进行推断
- (3) 可视化推断结果，将原图和可视化结果截图之后上传
- (4) 如果使用中有任何问题，可以到[Models](#) 或者 [Hub](#) 中提issue



检测任务介绍和前置概念

广告位：Base-detection组常年招收优秀的全职和实习生

简历可发送至lizeming@megvii.com 或者 base-detection-fte@megvii.com

Q & A



下期预告

天元官网：<https://megengine.org.cn/>

GitHub：<https://github.com/MegEngine>

天元开发者交流群



扫一扫二维码，加入群聊。

比赛任务： 视频超分辨率



<https://studio.brainpp.com/competition>

大赛奖项设置

- 第一名：团队奖金人民币5万
- 第二名：团队奖金人民币2万
- 第三名：团队奖金人民币1万
- 第四到第十名：团队奖金人民币1千

参与奖：所有参加初赛，并且提交有效结果的团队，纪念T-Shirt



(所有奖金均为税前奖金)

赛事安排

- 报名阶段：2020年8月10日 (UTC+8) 到2020年8月31日23:59 (UTC+8)
- 初赛阶段：2020年9月1日00:01(UTC+8) 到2020年9月14日23:59(UTC+8)
- 决赛阶段：2020年9月16日00:01(UTC+8) 到2020年9月24日23:59(UTC+8)

大赛前三名团队的成员还可获得旷视校招面试直通卡，直接进入面试环节

天元开发者交流群



扫一扫二维码，加入群聊。

直播+大赛报名 | 旷视研究院高级研究组组长范浩强

视频超分辨率算法前沿进展

直播预告海报，背景为黄色和白色。左侧黄色区域包含标题“视频超分辨率算法前沿进展”，嘉宾信息“嘉宾：旷视研究院高级研究组组长 范浩强”，直播时间“直播时间：2020年8月19日 20:00”以及“去观看”按钮。右侧白色区域包含范浩强研究员的肖像，他穿着印有“MEGVII 旷视”字样的黑色T恤。

视频超分辨率算法 前沿进展

嘉宾：旷视研究院高级研究组组长 范浩强

直播时间：2020年8月19日 20:00 [去观看](#)

时间：北京时间 8月19日 (周三) 20:00
地点：将门TechBeat技术社区（在线）
<http://www.techbeat.net/>

天元开发者交流群



扫一扫二维码，加入社群。

活动预告： MegEngine 框架设计讨论

<https://tuna.moe/event/2020/megengine-framework/>

金枪鱼之夜：MegEngine 架构设计

🕒 时间: 2020-08-22 19:00-21:00 📍 地点: Zoom tonight deep learning MegEngine

AI 浪潮一波又一波，仿佛不会算法就落后于时代了。深度学习框架处理了各种设备的计算细节、求导、计算优化序列的工作，而在动态、静态两套截然不同的世界中，这些步骤又各自有他们不同的优化点和瓶颈。

MegEngine 天元作为一个用了5年多的深度学习框架，是一个在淘金热时，坚持选择卖铲子的团队。这次讲座，我们邀请了旷视研究院 AI 系统高级技术总监、MegEngine 技术负责人许欣然作为主讲人。他将带我们了解一个深度学习框架是如何把网络的定义逐步优化并最终执行的，从框架开发者的视角来看待深度学习。

▶ 讲座大纲 (点击展开)

活动信息：

- 主讲人：许欣然
- 时间：2020/08/22 19:00 UTC +08:00
- 活动形式：线上会议 + 直播
 - Zoom：990 9174 0940
 - Zoom 直播：<https://live.bilibili.com/22436423>

欢迎一起来玩！

天元开发者交流群



扫一扫二维码，加入群聊。

欢迎加入“天元开发者交流群”



行正则致远

AI向善，行胜于言。

MEGVII 旷视