链滴

2021 年 CV 岗位精选面试题（21-31）| 文末小彩蛋

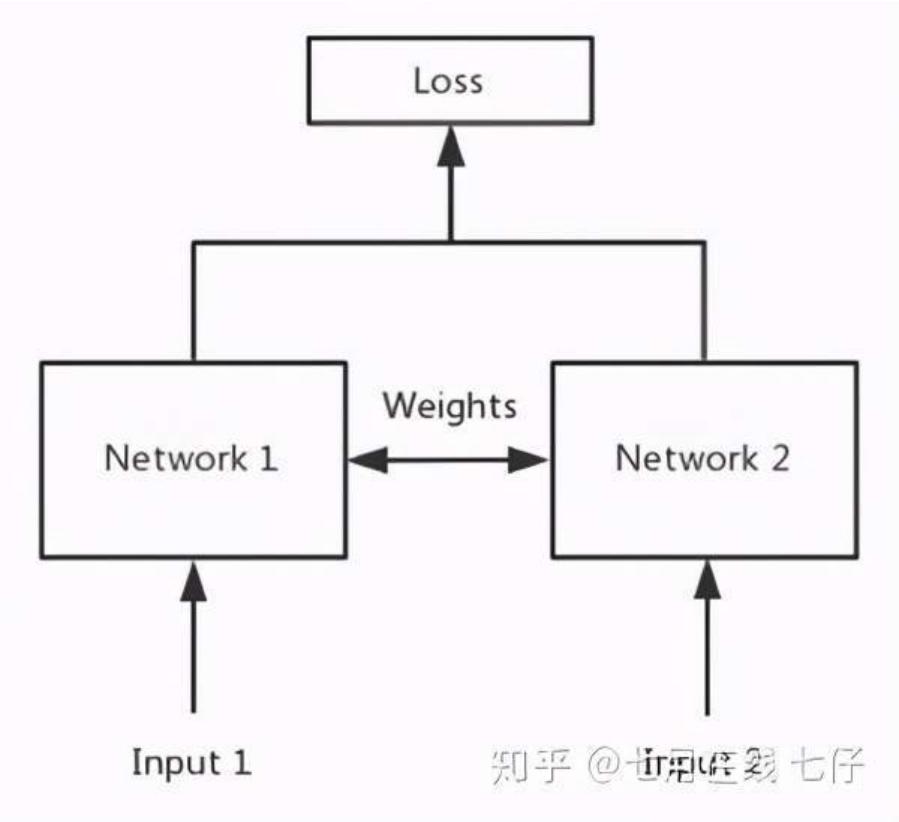
作者：julyedu

原文链接：https://ld246.com/article/1622186637914

来源网站：链滴

21、简述孪生随机网络（Siamese Network）

简单来说，Siamese Network就是“连体的神经网络”，神经网络的“连体”是通过共享权值来实现，如下图所示:

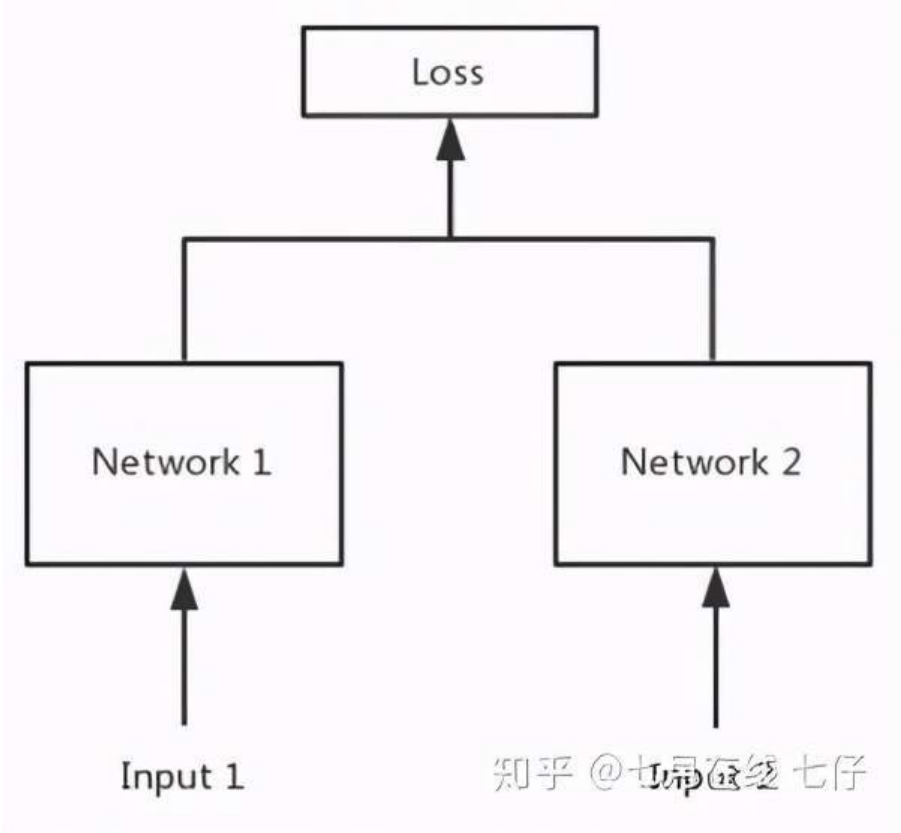


大家可能会有疑问：共享权值是什么意思？左右两个神经网络的权重一模一样？

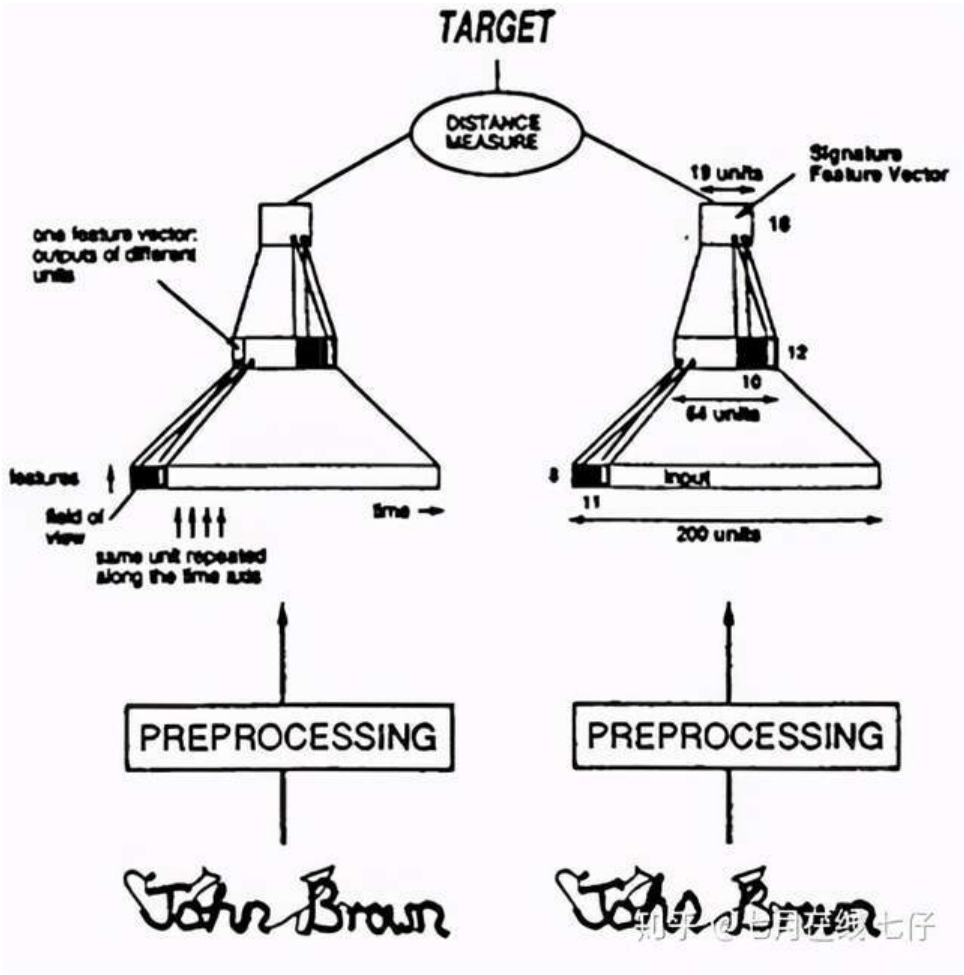
答：是的，在代码实现的时候，甚至可以是同一个网络，不用实现另外一个，因为权值都一样。对于Siamese Network，两边可以是LSTM或者CNN，都可以。

大家可能还有疑问：如果左右两边不共享权值，而是两个不同的神经网络，叫什么呢？

答：Pseudo-Siamese Network，伪孪生神经网络，如下图所示。对于Pseudo-Siamese Network两边可以是不同的神经网络(如一个是LSTM，一个是CNN)，也可以是相同类型的神经网络。



Yann LeCun养乐村同志在NIPS 1993上发表了论文《Signature Verification using a 'Siamese' Time Delay Neural Network》用于美国支票上的签名验证，即验证支票上的签名与银行预留签名是否一致。1993年，Yann LeCun就在用两个卷积神经网络做签名验证了。



原文链接：2021 年 CV 岗位精选面试题（21-31）| 文末小彩蛋

随着SVM等算法的兴起，Neural Network被人们遗忘，还好有一些执着的人们，坚守在了神经网络究的阵地。2010年Hinton在ICML上发表了文章《Rectified Linear Units Improve Restricted Boltz ann Machines》，用来做人脸验证，效果很好。其原理很简单，将两个人脸feed进卷积神经网络，出same or different。

22、DPM（Deformable Parts Model）算法流程

解析1：

将原图与已经准备好的每个类别的“模板”做卷积操作，生成一中类似热力图（hot map）的图像，不同尺度上的图合成一张，图中较量点就是与最相关“模板”相似的点。

拓展：

- SGD(stochastic gradient descent)到training里
- NMS(non-maximum suppression)对后期testing的处理非常重要
- Data mining hard examples这些概念至今仍在使用

解析2：

DPM算法由Felzenszwalb于2008年提出，是一种基于部件的检测方法，对目标的形变具有很强的鲁性。目前DPM已成为众多分类、分割、姿态估计等算法的核心部分，Felzenszwalb本人也因此被VO授予"终身成就奖"。

DPM算法采用了改进后的HOG特征，SVM分类器和滑动窗口(Sliding Windows)检测思想，针对目的多视角问题，采用了多组件（Component）的策略，针对目标本身的形变问题，采用了基于图结（Pictorial Structure）的部件模型策略。此外，将样本的所属的模型类别，部件模型的位置等作为变量（Latent Variable），采用多示例学习（Multiple-instance Learning）来自动确定。

1、通过Hog特征模板来刻画每一部分，然后进行匹配。并且采用了金字塔，即在不同的分辨率上提Hog特征。

2、利用提出的Deformable PartModel，在进行object detection时，detect window的得分等于par的匹配得分减去模型变化的花费。

3、在训练模型时，需要训练得到每一个part的Hog模板，以及衡量part位置分布cost的参数。文章提出了LatentSVM方法，将deformable part model的学习问题转换为一个分类问题：利用SVM学，将part的位置分布作为latent values，模型的参数转化为SVM的分割超平面。具体实现中，作者采了迭代计算的方法，不断地更新模型。

rootfilters根滤波器数组，其每个元素表示一个组件模型的根滤波器的信息，每个元素包括3个字段：

size：根滤波器的尺寸，以cell为单位，w*h

w：根滤波器的参数向量，维数为(w*h)*31

blocklabel：此根滤波器所在的数据块标识

滤波器(模版)就是一个权重向量，一个w * h大小的滤波器F是一个含w * h * 9 * 4个权重的向量(9*4是个HOG细胞单元的特征向量的维数)。所谓滤波器的得分就是此权重向量与HOG金字塔中w * h大小窗口的HOG特征向量的点积(DotProduct)。

23、什么是NMS（Non-maximum suppression 非极

值抑制）？

解析1：

NMS是一种Post-Procession（后处理）方式，跟算法无关的方式。

NMS应用在所有物体检测的方法里。

NMS物体检测的指标里，不允许出现多个重复的检测。

NMS把所有检测结果按照分值(conf. score)从高到底排序,保留最高分数的 box,删除其余值。

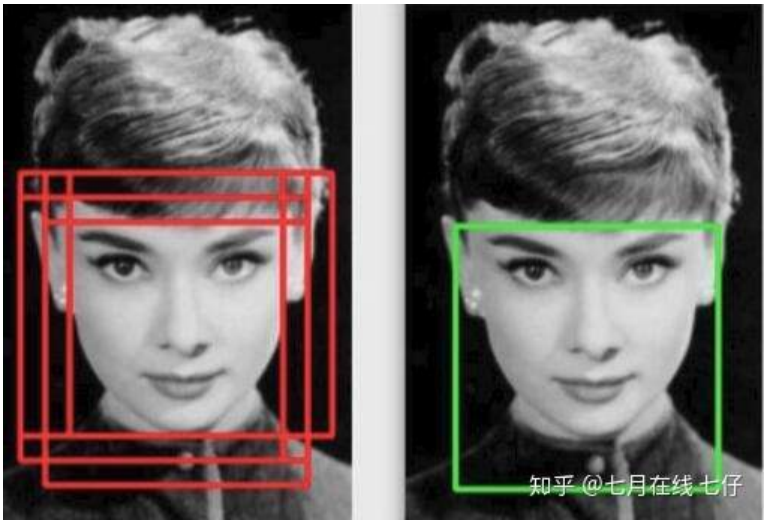
解析2：

概述

非极大值抑制（Non-Maximum Suppression，NMS），顾名思义就是抑制不是极大值的元素，可理解为局部最大搜索。这个局部代表的是一个邻域，邻域有两个参数可变，一是邻域的维数，二是邻的大小。这里不讨论通用的NMS算法(参考论文《Efficient Non-Maximum Suppression》对1维和维数据的NMS实现)，而是用于目标检测中提取分数最高的窗口的。例如在行人检测中，滑动窗口经取特征，经分类器分类识别后，每个窗口都会得到一个分数。但是滑动窗口会导致很多窗口与其他窗存在包含或者大部分交叉的情况。这时就需要用到NMS来选取那些邻域里分数最高（是行人的概率大），并且抑制那些分数低的窗口。

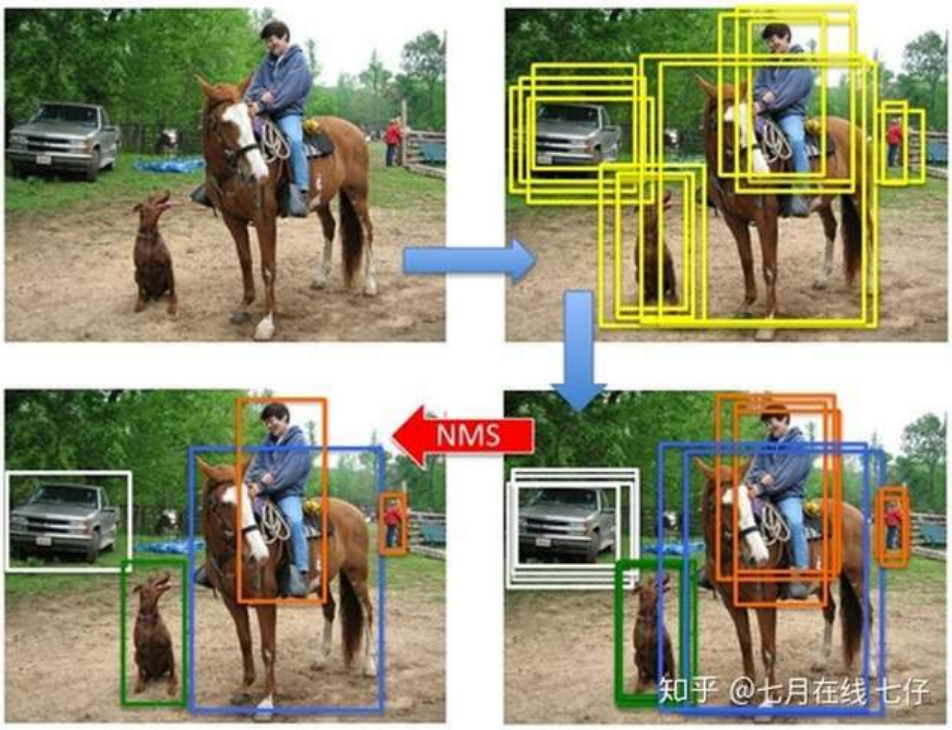
NMS在计算机视觉领域有着非常重要的应用，如视频目标跟踪、数据挖掘、3D重建、目标识别以及理分析等。

NMS 在目标检测中的应用-人脸检测框重叠例子：



我们的目的就是要去除冗余的检测框,保留最好的一个.

目标检测 pipline



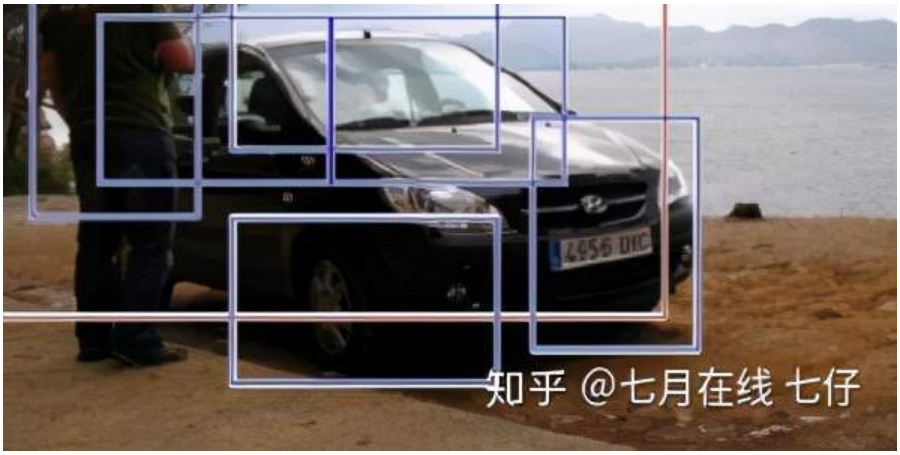
产生proposal后使用分类网络给出每个框的每类置信度,使用回归网络修正位置,最终应用NMS.

NMS 原理

对于Bounding Box的列表B及其对应的置信度S,采用下面的计算方式.选择具有最大score的检测框M,其从B集合中移除并加入到最终的检测结果D中.通常将B中剩余检测框中与M的IoU大于阈值Nt的框从中移除.重复这个过程,直到B为空.

重叠率(重叠区域面积比例IOU)阈值：常用的阈值是 0.3 ~ 0.5.

其中用到排序,可以按照右下角的坐标排序或者面积排序,也可以是通过SVM等分类器得到的得分或概率R-CNN中就是按得分进行的排序.



就像上面的图片一样，定位一个车辆，最后算法就找出了一堆的方框，我们需要判别哪些矩形框是没的。非极大值抑制的方法是：先假设有6个矩形框，根据分类器的类别分类概率做排序，假设从小到属于车辆的概率 分别为A、B、C、D、E、F。

(1)从最大概率矩形框F开始，分别判断A~E与F的重叠度IOU是否大于某个设定的阈值;

(2)假设B、D与F的重叠度超过阈值，那么就扔掉B、D；并标记第一个矩形框F，是我们保留下来的。

(3)从剩下的矩形框A、C、E中，选择概率最大的E，然后判断E与A、C的重叠度，重叠度大于一定的值，那么就扔掉；并标记E是我们保留下来的第二个矩形框。

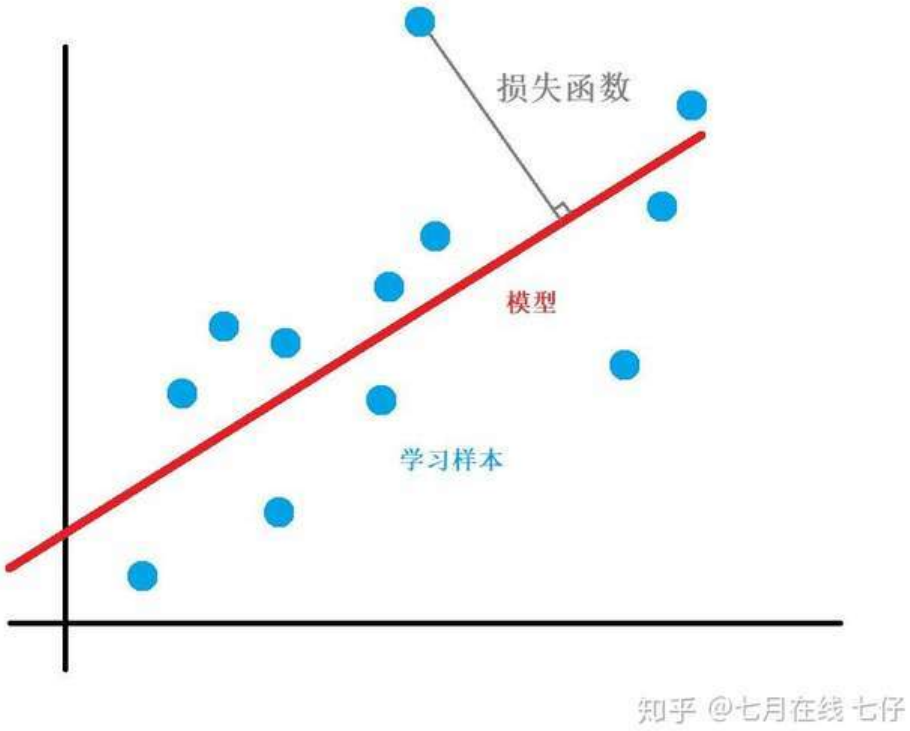
就这样一直重复，找到所有被保留下来的矩形框。

更多内容请参考：非极大值抑制（Non-Maximum Suppression，NMS）

24、列举出常见的损失函数(三个以上)?

什么是损失函数?

损失函数 （Loss Function） 也可称为代价函数 （Cost Function）或误差函数（Error Function）用于衡量预测值与实际值的偏离程度。一般来说，我们在进行机器学习任务时，使用的每一个算法都一个目标函数，算法便是对这个目标函数进行优化，特别是在分类或者回归任务中，便是使用损失函（Loss Function）作为其目标函数。机器学习的目标就是希望预测值与实际值偏离较小，也就是希损失函数较小，也就是所谓的最小化损失函数。



损失函数是用来评价模型的预测值与真实值的不一致程度，它是一个非负实值函数。损失函数越小，型的性能就越好。

25、做过目标检测项目么？比如Mask R-CNN和Pytho做一个抢车位神器

每逢春节过年，就要开始走亲访友了。这时候的商场、饭馆也都是“人声鼎沸”，毕竟走亲戚串门必可少要带点礼品、聚餐喝茶。

热闹归热闹，这个时候最难的问题可能就是怎样从小区、商场、菜市场的人山人海里准确定位，找到个“车位”。

一位名叫Adam Geitgey的软件工程师、AI软件工程博主也被“停车难”的问题困扰已久。为了让自能给迅速定位空车位，他用实例分割模型Mask R-CNN和python写了一个抢占停车位的小程序。

以下是作者以第一人称给出的教程，enjoy。

一、如何找停车位

我住在一个大都市，但就像大多数城市一样，在这里很难找到停车位。停车场总是停得满满的，即使自己有私人车位，朋友来访的时候也很麻烦，因为他们找不到停车位。

我的解决方法是：

用摄像头对着窗外拍摄，并利用深度学习算法让我的电脑在发现新的停车位时给我发短信。

这可能听起来相当复杂，但是用深度学习来构建这个应用，实际上非常快速和简单。有各种现有的实工具 - 我们只需找到这些工具并且将它们组合在一起。

26、如何理解Faster RCNN

目前学术和工业界出现的目标检测算法分成3类：

1. 传统的目标检测算法： Cascade + HOG/DPM + Haar/SVM以及上述方法的诸多改进、优化；

2. 候选区域/框 + 深度学习分类： 通过提取候选区域，并对相应区域进行以深度学习方法为主的分类方案，如：

R-CNN（Selective Search + CNN + SVM）

SPP-net（ROI Pooling）

Fast R-CNN（Selective Search + CNN + ROI）

Faster R-CNN（RPN + CNN + ROI）

R-FCN

等系列方法；

3. 基于深度学习的回归方法： YOLO/SSD/DenseBox 等方法；以及最近出现的结合RNN算法的RRC etection；结合DPM的Deformable CNN等

经过R-CNN和Fast RCNN的积淀，Ross B. Girshick在2016年提出了新的Faster RCNN，在结构上，aster RCNN已经将特征抽取(feature extraction)，proposal提取，bounding box regression(rect r fine)，classification都整合在了一个网络中，使得综合性能有较大提高，在检测速度方面尤为明显。

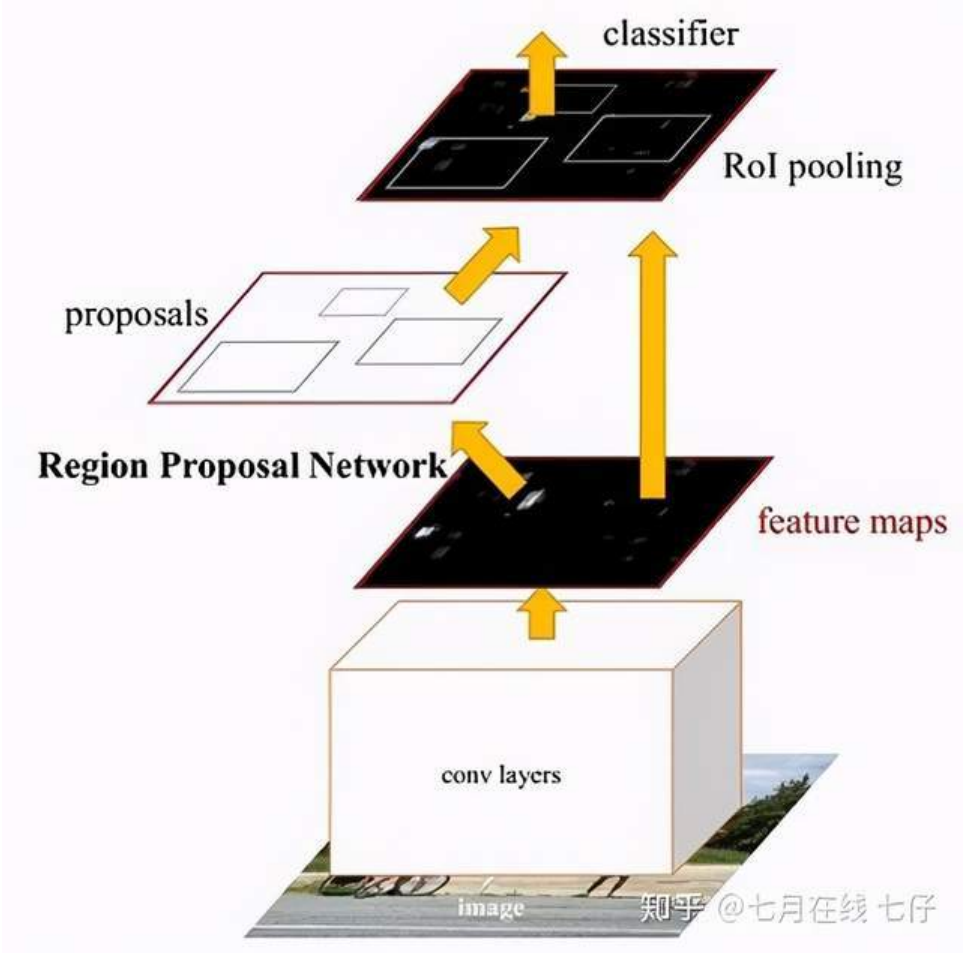


图1 Faster RCNN基本结构（来自原论文）

依作者看来，如图1，Faster RCNN其实可以分为4个主要内容：

①Conv layers。作为一种CNN网络目标检测方法，Faster RCNN首先使用一组基础的conv+relu+p oling层提取image的

②feature maps。该feature maps被共享用于后续RPN层和全连接层。

③Region Proposal Networks。RPN网络用于生成region proposals。该层通过softmax判断ancho s属于foreground或者background，再利用bounding box regression修正anchors获得精确的pro osals。

Roi Pooling。该层收集输入的feature maps和proposals，综合这些信息后提取proposal feature maps，送入后续全连接层判定目标类别。

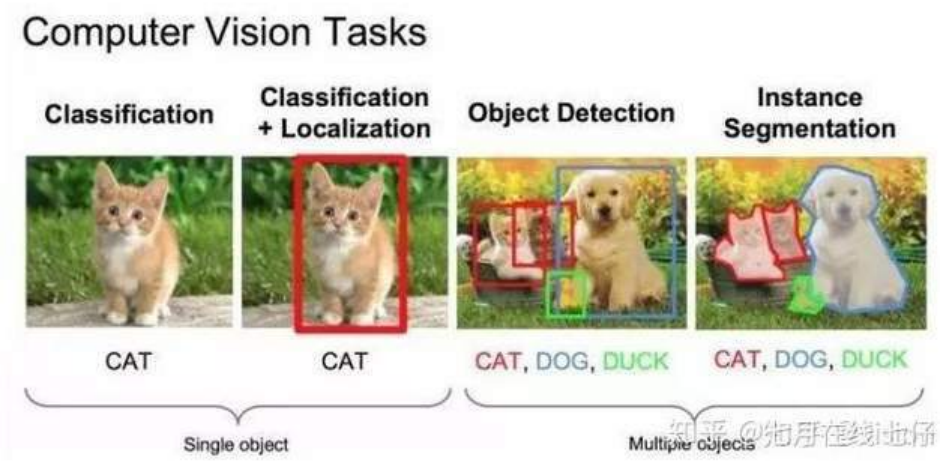
④Classification。利用proposal feature maps计算proposal的类别，同时再次bounding box regression获得检测框最终的精确位置。

所以本文以上述4个内容作为切入点介绍Faster R-CNN网络。

27、one-stage和two-stage目标检测方法的区别和优点？

众所周知，物体检测的任务是找出图像或视频中的感兴趣物体，同时检测出它们的位置和大小。

当然，物体检测过程中有很多不确定因素，如图像中物体数量不确定，物体有不同的外观、形状、姿，加之物体成像时会有光照、遮挡等因素的干扰，导致检测算法有一定的难度。



由于目标检测的应用场景广泛，所以在CV面试中经常出现，比如七月在线有一CV就业班的学员出去试时，便被问到“one-stage和two-stage目标检测方法的区别和优缺点？”（详见此文：从测试到C算法工程师的转型之路：最终连拿4个offer - 七月在线

虽然我们在本文中详细介绍了各个目标检测的方法：AI笔试面试题库 - 七月在线，但如果你是第一次到one-stage和two-stage，你会不会瞬间一脸懵逼，这是啥？

其实很简单，顾名思义，区别在于是一步到位还是两步到位。

具体说来，进入深度学习时代以来，物体检测发展主要集中在两个方向：

two stage算法，如R-CNN系列；

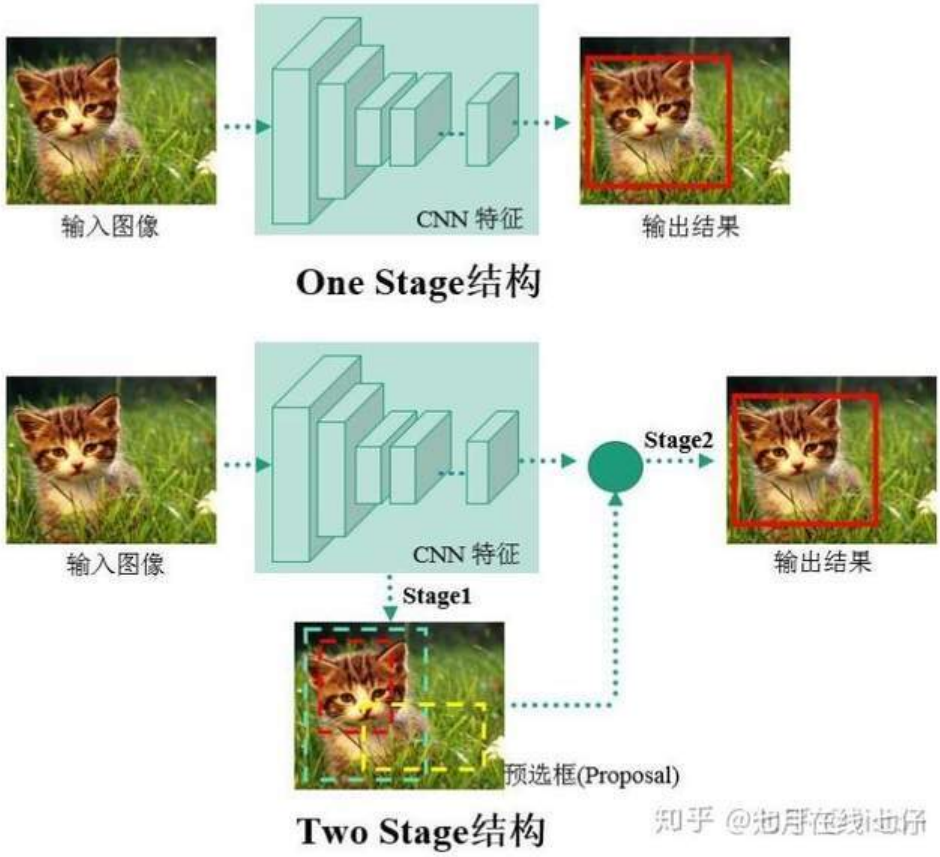
ones-tage算法，如YOLO、SSD等。

两者的主要区别在于two stage算法需要先生成proposal（一个有可能包含待检物体的预选框），然进行细粒度的物体检测，而one stage算法会直接在网络中提取特征来预测物体分类和位置。

所以说，目标检测算法two-stage，如Faster R-CNN算法会先生成候选框（region proposals，可包含物体的区域），然后再对每个候选框进行分类（也会修正位置）。这类算法相对就慢，因为它需多次运行检测和分类流程。

而另外一类one-stage目标检测算法（也称one-shot object detectors），其特点是一步到位，仅仅要送入网络一次就可以预测出所有的边界框，速度相对较快，非常适合移动端，最典型的one-stage

测算法包括YOLO，SSD，SqueezeDet以及DetectNet。



简单吧，恍然大悟，原来如此！而且one-stage看起来更高级。

28、请画下YOLOv3的网络结构

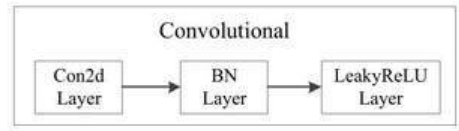
本人是小白，看后表示有点蒙。于是在Github上搜了大牛们基于Tensorflow搭建的YOLOv3模型进分析（本人只接触过TF，所以就不去看caffe的源码了）。接下来我会根据我阅读的代码来进一步分网络的结构。Github YOLOv3大牛代码链接。

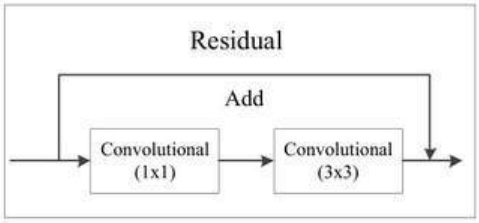
1.Darknet-53 模型结构

在论文中虽然有给网络的图，但我还是简单说一下。这个网络主要是由一系列的1x1和3x3的卷积层组（每个卷积层后都会跟一个BN层和一个LeakyReLU)层，作者说因为网络中有53个convolutional layes，所以叫做Darknet-53（2 + 12 + 1 + 22 + 1 + 82 + 1 + 82 + 1 + 4*2 + 1 = 53 按照顺序数，不括Residual中的卷积层，最后的Connected是全连接层也算卷积层，一共53个）。

下图就是Darknet-53的结构图，在右侧标注了一些信息方便理解（卷积的strides默认为（1，1），pdding默认为same，当strides为（2，2）时padding为valid）

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			



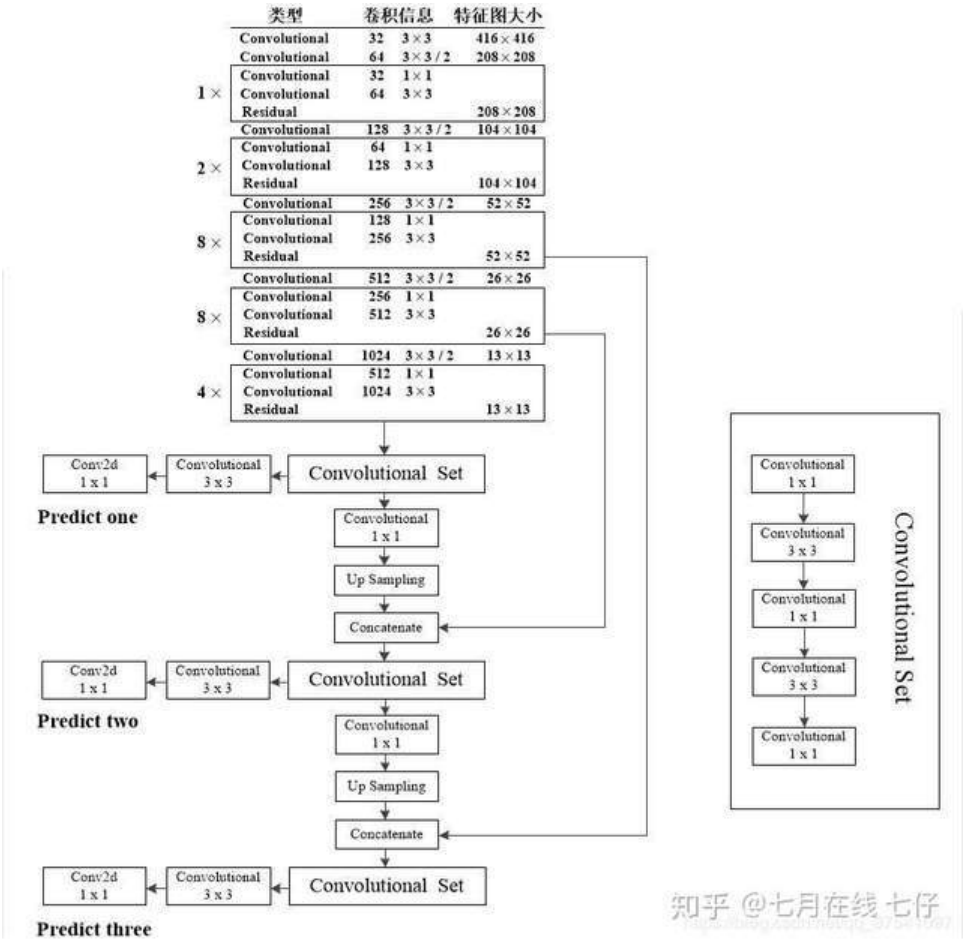


看完上图应该就能自己搭建出Darknet-53的网络结构了，上图是以输入图像256 x 256进行预训练来行介绍的，常用的尺寸是416 x 416，都是32的倍数。下面我们再来分析下YOLOv3的特征提取器，看究竟是在哪几层Features上做的预测。

2.YOLOv3 模型结构

作者在论文中提到利用三个特征层进行边框的预测，具体在哪三层我感觉作者在论文中表述的并不清（例如文中有“添加几个卷积层”这样的表述），同样根据代码我将这部分更加详细的分析展示在下中。

注意：原Darknet53中的尺寸是在图片分类训练集上训练的，所以输入的图像尺寸是256x256，下图以YOLO v3 416模型进行绘制的，所以输入的尺寸是416x416，预测的三个特征层大小分别是52，2，13。



原文链接：2021 年 CV 岗位精选面试题（21-31）| 文末小彩蛋

在上图中我们能够很清晰的看到三个预测层分别来自的什么地方，以及Concatenate层与哪个层进行接。注意Convolutional是指Conv2d+BN+LeakyReLU，和Darknet53图中的一样，而生成预测结的最后三层都只是Conv2d。通过上图小伙伴们就能更加容易地搭建出YOLOv3的网络框架了。

29、请简单说下YOLOv1,v2,v3,v4各自的特点与发展史

文章目录

一、任务描述

目标检测是为了解决图像里的物体是什么，在哪里的问题。输入一幅图像，输出的是图像里每个物体类别和位置，其中位置用一个包含物体的框表示。

需要注意，我们的目标，同时也是论文中常说的感兴趣的物体，指我们关心的类别（行人检测只检测，交通检测只关心交通工具等），或者数据集包含的类别，并不是图像里所有的物体都是目标，比如筑，草坪也是物体，但他们常常是背景。

从计算机视觉的角度看，目标检测是分类+定位，从机器学习的角度看，目标检测是分类+回归。

二、设计思想

目标检测架构分为两种，一种是two-stage，一种是one-stage，区别就在于 two-stage 有region pr

posal 过程，类似于一种海选过程，网络会根据候选区域生成位置和类别，而 one-stage 直接从图片成位置和类别。

今天提到的 YOLO 就是一种 one-stage 方法。YOLO 是 You Only Look Once 的缩写，意思是神经络只需要看一次图片，就能输出结果。

30、如何理解YOLO：YOLO详解

从五个方面解读CVPR2016 目标检测论文YOLO: Unified, Real-Time Object Detection

创新

核心思想

效果

改进

实践

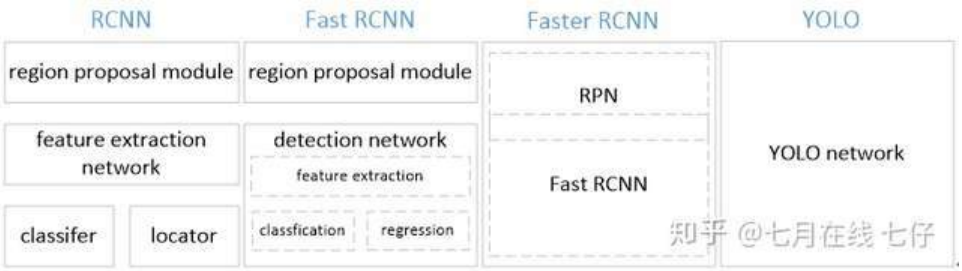
1 创新

YOLO将物体检测作为回归问题求解。基于一个单独的end-to-end网络，完成从原始图像的输入到物位置和类别的输出。从网络设计上，YOLO与rcnn、fast rcnn及faster rcnn的区别如下：

[1] YOLO训练和检测均是在一个单独网络中进行。YOLO没有显示地求取region proposal的过程。而cnn/fast rcnn 采用分离的模块（独立于网络之外的selective search方法）求取候选框（可能会包含体的矩形区域），训练过程因此也是分成多个模块进行。Faster rcnn使用RPN（region proposal nework）卷积网络替代rcnn/fast rcnn的selective

search模块，将RPN集成到fast rcnn检测网络中，得到一个统一的检测网络。尽管RPN与fast rcnn享卷积层，但是在模型训练过程中，需要反复训练RPN网络和fast rcnn网络（注意这两个网络核心卷层是参数共享的）。

[2]YOLO将物体检测作为一个回归问题进行求解，输入图像经过一次inference，便能得到图像中所物体的位置和其所属类别及相应的置信概率。而rcnn/fast rcnn/faster rcnn将检测结果分为两部分求：物体类别（分类问题），物体位置即bounding box（回归问题）。



2. 核心思想

2.1 网络定义

YOLO检测网络包括24个卷积层和2个全连接层，如下图所示。

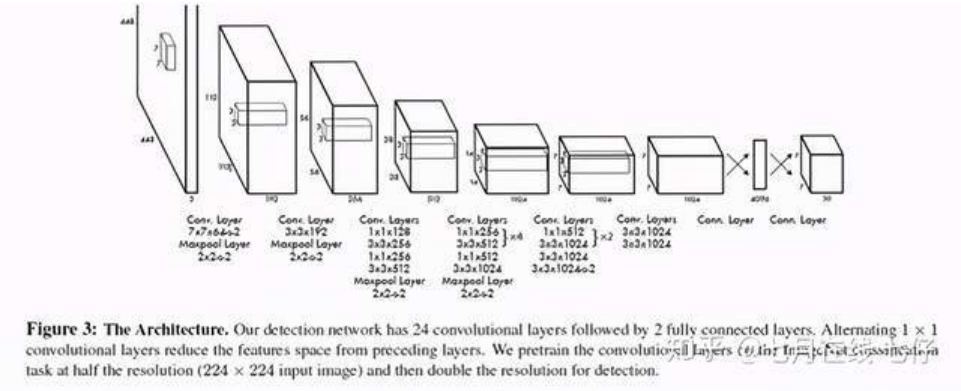


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

其中，卷积层用来提取图像特征，全连接层用来预测图像位置和类别概率值。YOLO网络借鉴了GoogeNet分类网络结构。不同的是，YOLO未使用inception

module，而是使用1x1卷积层（此处1x1卷积层的存在是为了跨通道信息整合）+3x3卷积层简单替。

YOLO论文中，作者还给出一个更轻快的检测网络fast YOLO，它只有9个卷积层和2个全连接层。使用tan x GPU，fast YOLO可以达到155fps的检测速度，但是mAP值也从YOLO的63.4%降到了52.7%但却仍然远高于以往的实时物体检测方法（DPM）的mAP值。

31、怎么理解YOLOv4

本题解析来源：McGL：YOLOv4

Jonathan Hui的博文依然是YOLO系列解读写得最好的，不像抢首发的媒体基本上就是把paper简要译了一下，也没有洋洋洒洒的直接罗列所有细节。本文从YOLOv4改进的intuition出发，循序渐进层清晰的介绍了各个模块和影响，这样会有更全面的把握，不会迷失在繁枝密叶中，而需要了解更多的也可以顺藤摸瓜去仔细研读。

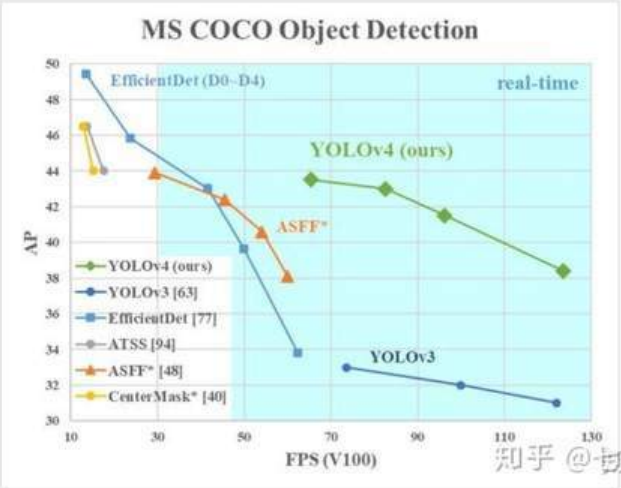
YOLO v1 - v3 回顾

McGL：实时目标检测YOLO系列进化史 McGL：实时目标检测YOLO系列进化史

YOLOv4 by Jonathan Hui

https://medium.com/@jonathan_hui/yolov4-c9901eaa8e61

即使目标检测在最近几年开始成熟，竞争仍然激烈。 如下图所示，YOLOv4声称具有state-of-the-ar的精度，同时保持了高帧率。 在 Tesla V100上，以大约65 FPS 的推理速度，MS COCO达到了 43.5 AP (65.7% AP50)。 在目标检测领域，高准确率不再是唯一的圣杯。 我们希望模型能够在边缘设备顺利运行。 如何用低成本的硬件实时处理输入视频也变得非常重要。



阅读 YOLOv4开发的有趣部分是什么新技术已经被评估、修改并集成到 YOLOv4中。 并且它还做了些改变，使得检测器更适合在单一 GPU 上进行训练。

Bag of freebies (BoF) & Bag of specials (BoS)

在训练过程中可以进行一些改进(如数据增强、类别不均衡、成本函数、软标记等……) 来提高精度。 些改进对推理速度没有影响，被称为“赠品袋(bag of freebies)”。 然后，还有“特价品袋(bag of specials)”，它对推理时间的影响较小，性能回报较好。 这些改进包括感受野的增加、注意力的使用特征整合(如skip连接和 FPN)以及后处理(如NMS)。 在本文中，我们将讨论特征提取器和颈部(neck)何设计，以及所有 BoF 和 BoS 这些好东西。