

深度学习目标检测

数据集

PASCAL VOC数据集:4大类, 20个小类

2007: 9963图片/24640目标

2012: 23080图片/54900目标

COCO数据集: 20万个图片, 80个类别, 超过50万个数据标注, 平均每个图片7.2个目标

评价指标

groundTruth:类别+真实边界框坐标 (x,y,w,h) ,(x,y)是指目标中心点的坐标, (w,h)分别是目标框的宽度和长度

IoU (intersssection over Union):交并比

Intersection over Union (IoU) 是在目标检测和图像分割中用来评估模型性能的一个指标。它衡量了预测的边界框与真实边界框之间的重叠程度。IoU 的计算方法如下:

计算步骤

1. 计算交集:

- 计算预测框和真实框重叠的区域 (交集) (Area of Intersection) 。

2. 计算并集:

- 计算预测框和真实框的总区域 (并集) , 公式为:
- $\text{Area of Union} = \text{Area of Prediction} + \text{Area of Ground Truth} - \text{Area of Intersection}$

3. 计算 IoU:

使用以下公式计算 IoU:

这是使用 LaTeX 格式表达的 IoU (可能是指 Intersection over Union) 的公式:

$$\text{IoU} = \frac{\text{Area of Intersection}}{\text{Area of Union}}$$

结果解释

- IoU = 0:** 表示没有重叠。
- IoU = 1:** 表示完全重叠。
- IoU 的常用阈值:** 在目标检测中, 通常使用 $\text{IoU} \geq 0.5$ 作为判断检测结果是否为正例的标准。

应用

- 模型评估:** IoU 是目标检测模型性能评估的重要指标, 广泛用于 COCO、Pascal VOC 等数据集的评分。
- 训练过程:** 在训练过程中, IoU 也可以作为损失函数的一部分, 优化模型的预测精度。

IoU 提供了一种直观且有效的方式来评估目标检测的准确性, 是计算机视觉领域中的重要概念。

生成的预测结果会非常多，首先过滤掉低类别置信度的检测结果，使用IoU作为边界框正确性的度量指标。

评价指标	解释	groundTruth	预测结果	目标检测中的解释
TP (True Positives)	真的正样本	正样本	正样本	LoU>阈值
FP (False Positives)	假的正样本	负样本	正样本	LoU<阈值
TN (True Negatives)	真的负样本	负样本	负样本	
FN (False Negatives)	假的负样本	正样本	负样本	漏检目标

1. TP (True Positive, 真正例)
- **定义：**检测到的目标与真实目标完全匹配的数量。
 - **示例：**如果一个模型正确检测到图像中的一只狗，并且该图像中确实有一只狗，则这被视为一个真正例。
2. FP (False Positive, 假正例)
- **定义：**检测到的目标被错误标记为正类的数量。
 - **示例：**如果一个模型在图像中检测到一只狗，但实际上没有狗（比如检测到了一只猫），则这被视为一个假正例。
3. TN (True Negative, 真负例)
- **定义：**没有检测到任何目标且图像中确实没有目标的数量。
 - **示例：**如果图像中没有动物，且模型也没有检测到任何目标，则这被视为一个真负例。
4. FN (False Negative, 假负例)
- **定义：**真实存在的目标未被检测到的数量。
 - **示例：**如果图像中有一只狗，但模型未能检测到它，则这被视为一个假负例。

总结

- **TP** 和 **FP** 衡量模型的阳性预测能力，反映模型的准确性。
- **TN** 和 **FN** 则帮助评估模型在未检测到目标时的表现。

在目标检测中，通常会结合 IoU (Intersection over Union) 等指标来进一步评估检测的准确性，以确保检测框与真实目标之间的匹配程度。

AP (Average Precision) 和 mAP (mean Average Precision) 是用于评估目标检测和信息检索等任务性能的常用指标。

AP (Average Precision)

- **定义：**AP 是在不同召回率下计算的精确度的平均值。通常，AP 是通过在精确度-召回率曲线下计算面积来获得的。
- **计算步骤：**
 1. 根据IoU划分TP和FP（通常如果有多个预测框住了一个真实框，那我们只选取一个预测款为TP，其他做FP）
 2. 按置信度的从大到小，计算P值和R值。
 3. 绘制P-R曲线，进行AP计算。

- **精确度 (Precision) :**

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **召回率 (Recall) :** 衡量模型在所有实际正例中正确识别出的比例

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **Average Precision (AP) :**

$$\text{AP} = \int_0^1 \text{Precision}(r) dr$$

或者在离散情况下，通过取不同召回率下的精确度值的平均。

mAP (mean Average Precision)

- **定义:** mAP 是多个类别的 AP 值的平均值，常用于多类目标检测任务。

- **计算步骤:**

1. 针对每个类别计算 AP。
2. 将所有类别的 AP 值求平均，得到 mAP。

- **mAP 公式:**

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^N \text{AP}_i$$

其中，(N) 是类别的总数，(\text{AP}_i) 是第 (i) 类别的 Average Precision。

应用

- **AP 和 mAP** 常用于评估模型在物体检测、图像分割、信息检索等领域的性能，能够全面反映模型的准确性和召回能力。

这两个指标非常重要，因为它们不仅考虑了正例的识别能力，还关注了负例的处理情况，能够更全面地评估模型的性能。

11点法计算AP

计算步骤

1. **排序预测结果:**

- 按照模型的预测分数对所有检测结果进行排序。

2. **计算精确度和召回率:**

- 对每个预测结果，计算精确度 (Precision) 和召回率 (Recall) 。
- 精确度 (Precision) 和召回率 (Recall) 的定义如下:

■

■

■

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

3. 构建精确度-召回率曲线：

- 将每个预测的精确度和召回率绘制成曲线。

4. 选择召回率点：

- 在0到1之间选取11个均匀分布的召回率点：[0, 0.1, 0.2,, 1.0]

5. 计算每个召回率点的精确度：

- 对于每个选定的召回率点，查找在该召回率及以上的所有精确度值，并取这些值的最大值。
- 这意味着，找到所有召回率大于等于该点的精确度，取最大值，以确保精确度不会因召回率的增加而降低。

6. 求平均：

- 将这11个召回率点的精确度值相加，然后除以11，得到最终的AP值：

- $$AP = \frac{1}{11} \sum_{i=0}^{10} \text{Precision}(r_i)$$

- 其中，(r_i) 为选定的召回率点。

7. 11 点法示例

假设我们有以下模型预测结果，显示了召回率和对应的精确度：

召回率 (Recall)	精确度 (Precision)
0.0	0.0
0.1	0.8
0.2	0.7
0.3	0.6
0.4	0.5
0.5	0.4
0.6	0.3
0.7	0.2
0.8	0.1
0.9	0.1
1.0	0.0

计算步骤：

- 选定召回率点：**选择 11 个均匀分布的召回率点：[0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0]
- 计算每个点的精确度：**
 - 对于每个召回率点，取该点及以上的精确度的最大值。
 - 例如，对于召回率 0.3，最大精确度是 0.6。
- 得到精确度列表：**

召回率	最大精确度
0.0	0.0
0.1	0.8
0.2	0.8
0.3	0.6
0.4	0.5
0.5	0.4
0.6	0.3
0.7	0.2
0.8	0.1
0.9	0.1
1.0	0.0

4. 计算 AP:

- 取这些最大精确度的平均值:

$$\circ AP = \frac{1}{11}(0.0 + 0.8 + 0.8 + 0.6 + 0.5 + 0.4 + 0.3 + 0.2 + 0.1 + 0.1 + 0.0) \approx 0.36$$

2、AP计算面积法

计算步骤

1. 排序预测结果:

- 根据模型的预测分数对所有检测结果进行排序。

2. 计算精确度和召回率:

- 对每个预测结果，计算精确度 (Precision) 和召回率 (Recall) 。
- 精确度和召回率的定义如下:

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

3. 构建精确度-召回率曲线:

- 将每个预测的精确度和召回率绘制成曲线，形成精确度-召回率图。

4. 计算曲线下的面积:

- 使用数值积分的方法计算曲线下的面积。常见的方法包括:
 - **梯形法则**: 将曲线划分为多个梯形，然后计算每个梯形的面积并求和。
 - **矩形法则**: 也可以使用矩形估算面积，但梯形法则通常更精确。

5. 得到 AP 值:

- 计算得到的面积值即为 AP 值。

数学表达

如果用离散的召回率和精确度值表示，AP 可以通过以下公式计算：

$$AP = \sum_{n=1}^N (\text{Recall}_n - \text{Recall}_{n-1}) \cdot \text{Precision}_n$$

2. 面积法示例

继续使用相同的召回率和精确度数据，构建精确度-召回率曲线并计算面积。

数据表

召回率 (Recall)	精确度 (Precision)
0.0	0.0
0.1	0.8
0.2	0.7
0.3	0.6
0.4	0.5
0.5	0.4
0.6	0.3
0.7	0.2
0.8	0.1
0.9	0.1
1.0	0.0

计算步骤：

1. **构建精确度-召回率曲线：**将上述数据点绘制在图上，形成曲线。

2. **计算面积：**

- 使用梯形法则计算曲线下的面积。
- 例如，计算从召回率 0 到 0.1、0.1 到 0.2，依此类推的每个梯形面积：

- $$\text{Area}_{0-0.1} = \frac{(0.0 + 0.8)}{2} \times (0.1 - 0.0) = 0.04$$

$$\text{Area}_{0.1-0.2} = \frac{(0.8 + 0.7)}{2} \times (0.2 - 0.1) = 0.075$$

-

- $$\text{Area}_{0.1-0.2} = \frac{(0.8 + 0.7)}{2} \times (0.2 - 0.1) = 0.075$$

$$\text{Area}_{0.2-0.3} = \frac{(0.7 + 0.6)}{2} \times (0.3 - 0.2) = 0.065$$

- 依此类推，直到 1.0。

3. 求和：

- 将所有梯形的面积相加，得到总面积（AP）：
- $$AP = \text{Area}_{0-0.1} + \text{Area}_{0.1-0.2} + \text{Area}_{0.2-0.3} + \dots$$

总结：

11点法通过在不同的召回率点上找到精确度的最大值，从而提供了一种稳健的方式来计算平均精度，特别适用于目标检测和其他分类任务。这种方法能够有效地平衡精确度和召回率，反映模型在不同阈值下的性能。

算法发展

传统检测方法：

输入--->区域选择----->特征提取----->分类器----->输出

区域选择：滑动窗口法：任务相对独立 需要人工设计尺寸 大量冗余操作 定位不准确

特征提取：颜色特征，形状特征，纹理特征、边缘特征

基于深度学习的检测：

机器视觉中的一阶段（One Stage）和两阶段（Two Stage）模型是指在目标检测任务中，不同的处理流程和架构。

一阶段模型（One Stage）

- **特点：**
 - 直接从输入图像中生成预测，通常是边界框和类别概率。
 - 处理速度快，适合实时应用。
- **例子：**
 - YOLO（You Only Look Once）
 - SSD（Single Shot Multibox Detector）
- **优点：**
 - 计算效率高，适合在资源有限的环境中使用。
 - 适合对速度要求高的场景，如视频监控。
- **缺点：**
 - 精度通常低于两阶段模型，尤其在处理复杂场景时。

两阶段模型（Two Stage）

- **特点：**
 - 首先生成候选区域，然后对这些区域进行分类和边界框回归。
- **例子：**
 - Faster R-CNN
 - R-CNN（Region-based CNN）

- **优点：**
 - 通常具有更高的检测精度，特别是在复杂背景或小物体检测中。
 - 能够进行更精细的特征提取。
- **缺点：**
 - 计算量大，处理速度较慢，不适合实时检测。

总结

一阶段模型注重速度，适合实时应用；而两阶段模型则注重精度，适合需要高准确度的任务。选择哪种模型取决于具体应用的需求。

anchor: 锚定框 anchor box中features Map来确定位置 ratio+scale来确定大小

anchor-free:自底向上 不依赖于预定义的锚框，而是直接从特征图中预测物体的位置和大小。

anchor-base:自顶向下 用预定义的锚框（anchor boxes）来生成候选区域。这些框在图像的特征图上被放置，以便检测各种物体。

NMS (Non-maximum suppression,NMS)非极大值抑制：在生成的多个框中筛选一个框

1. 设定目标框的置信度阈值，常用的阈值是0.5左右
2. 根据置信度降序排列候选框列表
3. 选定置信度最高的框A添加到输出列表，将其从候选框列表删除
4. 候选框列表中的所有框依次与A计算IoU,删除大于阈值的候选
5. 重复上诉过程，直到候选框列表为空，返回输出列表

Faster RCNN

候选区域生成----->特征提取----->类别判断----->位置精修

具有全连接的CNN需要固定输入大小：卷积层和池化需要经过Flatten之后，才能和全连接层相连

边界框回归：

边界框回归（Bounding Box Regression）是目标检测中的一个关键步骤，用于精确定位物体在图像中的位置。它通过回归算法预测物体的边界框参数，从而提高检测的准确性。

边界框的表示

一个边界框通常由四个参数表示：

- **x_center**：边界框中心点的 x 坐标。
- **y_center**：边界框中心点的 y 坐标。
- **width**：边界框的宽度。
- **height**：边界框的高度。

在某些情况下，边界框也可以用左上角和右下角的坐标来表示（x_min, y_min, x_max, y_max）。

边界框回归的过程

1. **候选区域生成：**
 - 在目标检测模型中，首先生成一系列候选区域（例如，使用区域提议网络）。

2. 回归目标的定义：

- 对于每个候选区域，定义回归目标，即真实边界框相对于候选区域的偏差。

3. 损失函数：

- 通常使用平滑 L1 损失或均方误差（MSE）作为损失函数来评估预测边界框与真实边界框之间的差异。

4. 模型训练：

- 使用训练数据来优化模型参数，使得预测的边界框尽可能接近真实边界框。

应用

边界框回归在多种目标检测模型中发挥重要作用，例如：

- Faster R-CNN**：结合区域提议和边界框回归，提高检测精度。
- YOLO 和 SSD**：在同一网络中同时进行分类和边界框回归。

总结

边界框回归是目标检测中提高定位精度的重要技术，通过回归算法优化边界框的位置和大小，从而实现更准确的物体检测。

YOLO

YOLO (You Only Look Once) 候选框策略：

1. 网格划分

YOLO将输入图像划分为 $S \times S$ 的网格。每个网格负责检测其中心点落在该网格内的物体。这种划分方式使得模型能够利用全局上下文信息进行检测。

2. 每个网格的候选框

每个网格预测固定数量的候选框（通常为2个或3个）。对于每个候选框，模型输出以下信息：

- 边界框坐标** (x, y, w, h)：相对于网格单元的中心点，宽度和高度。
- 置信度分数** (objectness score)：表示该框内存在物体的概率。
- 类别概率**：表示该候选框内物体属于每个类别的概率。

3. 置信度评分

每个候选框的置信度分数是基于两个因素计算的：

- 框内存在物体的概率。
- 框与真实物体的重叠程度（通常使用IoU，即Intersection over Union）。

4. 非极大值抑制 (NMS)

在候选框生成后，YOLO使用非极大值抑制 (NMS) 算法来消除冗余的框。NMS的步骤包括：

- 根据置信度分数对候选框进行排序。
- 选择分数最高的框，并移除与其IoU超过阈值的其他框。
- 重复这一过程，直到处理完所有框。

5. 多尺度预测

YOLO在不同的尺度上进行物体检测，以提高对不同尺寸物体的检测能力。通常在特征提取网络的不同层输出特征图，进行多尺度的候选框预测。

总结

通过以上策略，YOLO能够有效地生成和过滤候选框，实现快速且准确的物体检测。

YOLO1

输入大小为448X448X3。

输出的结果为7X7X30

将图片分成7X7的格子

而30是指30个值 (b_1+b_2+20)

b_1 包括 (x,y,w,h,c_1) b_2 包括 (x,y,w,h,c_1) (x,y)指的是位置, (w,h) 指的是框的大小 c 指的是置信度

20个值是指20个种类的概率

损失函数: 位置误差+置信度误差 (对正样本的) + 置信度误差 (对负样本的) + 分类误差

YOLO2

网络的每一层的输入都做了归一化, 收敛相对更容易

舍弃了Dropout, 卷积后全部加入Batch Normalization

V1训练时用的是224X224, 测试时使用448X448

V2训练时额外又进行了10次448X448的微调

使用了更大的分辨率

网络结构:

没有使用FC层, 使用了5次降采样