论文解读：End-to-End Object Detection with Transformers


布尔佛洛哥哥
算法翻译工程师傅

先记录几点：

匈牙利算法：看这里吧

spatial positional encoding：应该也就是用余弦函数。

object query：全靠transformer自己学习。

transformer 可以看大师兄：详解Transformer （Attention Is All You Need）

但是decoder讲得不是很清楚，decoder可以看变成海的话：Transformer原理详解

我自己的源码解读，看这里

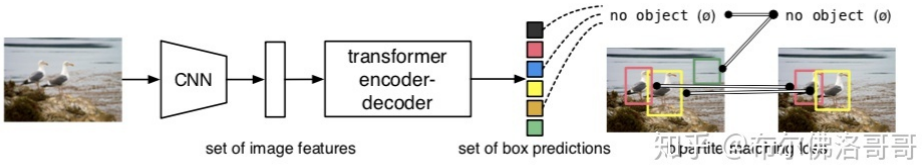
摘要

我们把目标检测看做是一种set prediction的问题，我们的方法也直接移除了一些人工设计的组件，例如NMS和anchor的生成。我们的框架DETR，由两个部分构成，一是set-based的全局loss，使用bipartite matching 生成唯一的预测，二是transformer的encoder-decoder 结构。只需提供固定大小学习到的目标查询集合，DETR推理出目标与全局图像上下文，直接并行地预测出结果。新的模型非常简单，不需要特定的库来支持。DETR在coco数据集上有着可以和faster-rcnn媲美的准确率与效率。而且它也能完成全景分割的任务。

介绍

目标检测的目标是预测一个bbox的集合和各个bbox的标签。目前的检测器不是直接预测一个目标的集合，而是使用替代的回归和分类去处理大量的propoasls、anchors或者window centers。模型的效果会受到一系列问题的影响：后处理去消除大量重叠的预测、anchors的设计、怎么把target box与anchor关联起来。为了简化流程，我们提出一种直接set prediction的方式来消除这些替代的方法。

我们把目标检测当做一种集合预测的问题来处理。我们采用了一种常用的序列预测架构，使用编码-解码器的transformer。这种架构可以显示的对序列中元素的两两关系来建模，容易满足一些集合预测的限制，比如消除冗余。



DETR 一次性预测所有目标，成对地匹配预测目标与真实目标并使用一种集合损失函数端到端地训练模型。DETR通过移除大量人工设计的模块，比如空间上的anchors和NMS。DETR不需要任何定制的网络层，因此可以在任何支持CNN和transformer的框架下复现。

DETR有两个主要的特征，两两匹配的loss和并行解码的transformers。相反之前的工作都是用

目标上效果不太好。将来可以用FPN来提升。

训练DETR在很多方面和常规目标检测器不同。这个新模型需要特别长的训练时间而且得益于附加的transformer中的解码损失。

DETR的设计可以很方便的扩展到其他任务上，我们在预训练的DETR上加了一个简单的分割器的头部结构，我们就可以在全景分割上超过基准模型。

相关工作

包括：集合预测的二分匹配损失，transformer上的编码-解码结构，并行解码，目标检测方法。

集合预测

现在没有经典的深度学习方法直接预测集合。基础的集合预测任务是多标签分类，基准方法--one-vs-rest 没有应用到例如检测的任务上，因为检测任务中的元素有着潜在结构。第一个困难就是避免临近的冗余。大多数目标检测器使用NMS这样的后处理来解决这个问题，但是直接的集合预测是不需要后处理的。它们需要全局的预测策略，对所有预测目标的交互进行建模，从而避免冗余。对于固定大小的集合预测，全连接神经网络足够了但是开销大。一种常见的方法是使用自回归的序列模型，比如RNN。在各种情况下，预测的排列应该不影响损失函数。常见的方法应该是设计一种基于匈牙利算法的损失，来找到真实值与预测值的二分匹配。这种可以保证损失不受排列的影像而且保证每一个目标有唯一一个匹配。我们依照着二分匹配损失的方法。不同于之前的其他工作，我们不使用自回归的模型，转而使用并行解码的transformer。

transformer 和并行解码

transformer最早是出现在机器翻译里的一种注意力构建块。注意力机制是用来聚合完整输入序列信息的神经网络层。transformer引入了自注意力层，类似于non-local的神经网络，扫描序列中的每个元素，然后通过聚合完整序列中的信息更新每个元素。注意力模型的一个最大的优点就是全局的计算和完美的内存使用，比RNN更适合处理长序列。transformer在自然语言处理、语音处理和计算机视觉等场景已经替换RNN。

transformer最早使用在自回归模型里，沿用序列转序列的模型，一个接一个的输出符号。然而，过高的预测开销(正比于输出长度，难做batch)导致了并行序列生成的发展，在语音、机器翻译，文字表示学习和语音识别。我们结合transformer和并行解码，权衡计算开销和全局计算的能力。

目标检测

近期的目标检测方法根据初始猜测去做预测。两阶段的目标检测器依据propoals预测bbox，一阶段的目标检测器依据anchor做预测，或者使用center去做预测。近期工作表明，这些系统的最终表现依赖初始的猜测。我们的模型，移除了人工涉及的流程，而且简化了检测过程，直接从输入的图片输出预测的集合。

集合损失：有一些检测器使用二分匹配损失。然而，在这些早期的深度学习模型，不同预测之间的关系是通过卷积或者全连接网络去学习，并且使用NMS去提升结果。近期的检测器使用不唯一的指定规则去指定真实值与预测值，并且使用NMS。

可学习的NMS和关系网络用注意力显式地学习不同预测目标之间的关系。直接使用集合损失，不需要任何后处理的步骤。但是这些模型需要附加的手工特征，我们需要的是减少人工的设计，更多的是靠网络自身。

循环检测器：有一些跟我们相似的一些做检测与分割的方法。但是这些方法只在小数据集上验证过。而且他们都使用自回归模型，并没有衡量transformer的并行解码。

目标检测集合预测损失

DETR输出固定大小为N的预测，只需要执行一次解码器，N比常规图片中待检测目标大得多。训练中最难的地方就是根据真实值评价预测目标(类别、位置、大小)。我们的损失构造了一个最优的二分匹配而且接着优化目标向（bounding box）的损失。

我们将y指示真实值， $\hat{y} = \{\hat{y_i}\}_{i=1}^{N}$ 指示N个预测值。假设N远大于图像中的目标，我们可以认为y的大小也是N，用 ϕ 填充空元素。目标就是找到这两个集合的二分匹配，中的一种排列 σ 有着最低的损失：

$$\hat{\sigma} = \arg\min_{\sigma \in \mathfrak{S}_N} \sum_{i}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma(i)}), \tag{1}$$

上述loss是真实值与预测值之间两两匹配的loss。使用匈牙利算法来计算。

匹配损失同时考虑到类别与真实值与预测值之间的相似度， $y_i = (c_i, b_i)$ 其中c是目标的类别，b是值域在[0，1]的四维向量，bbox的中心坐标与宽高。

寻找匹配目标的方法也是heuristic，跟最新的一些检测器相同，匹配proposals或者anchors。主要的区别就是一对一的匹配而且没有冗余。

第二步就是计算损失函数，之前的步骤就是使用匈牙利算法计算所有的匹配。我们定义的loss与常见的检测模型很相似，就是负对数似然与box损失的线性组合。

$$\mathcal{L}_{\text{Hungarian}}(y, \hat{y}) = \sum_{i=1}^{N} \left[-\log \hat{p}_{\hat{\sigma}(i)}(c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(b_i, \hat{b}_{\hat{\sigma}}(i)) \right], \tag{2}$$

其中 $\hat{\sigma}$ 是(1)中求得的最优匹配。类似于faster-rcnn对负样本权重的设置，当 $c_i = \phi$ 时，权重为原来的十分之一。目标与 ϕ 的匹配损失不依赖于预测值，因此是一个常量。在匹配损失中，我们使用概率去代替对数概率。这样是为了平衡类别预测与box预测的损失，我们发现这样的效果更好。

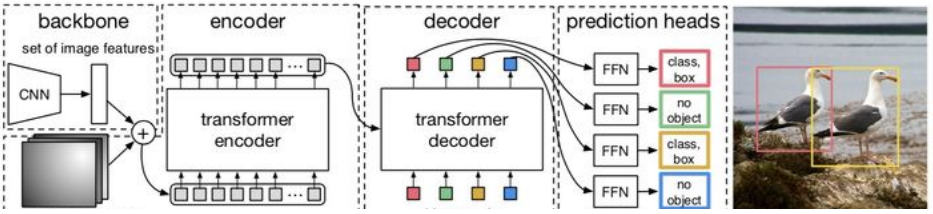
bbox损失

我们直接预测box在图像中的位置，直接使用L1loss的话，对小目标就不公平，因此我们使用了L1 loss 与IOU loss的组合，让loss对目标的大小不敏感。

$$\lambda_{\text{iou}} \mathcal{L}_{\text{iou}}(b_i, \hat{b}_{\sigma(i)}) + \lambda_{\text{L1}} ||b_i - \hat{b}_{\sigma(i)}||_1$$

在batch内部我们用目标的数量对loss做了归一化。

DETR的结构



small fixed number of learned positional embeddings, which we call *object queries*, and additionally attends to the encoder output. We pass each output embedding of the decoder to a shared feed forward network (FFN) that predicts either a detection (class and bounding box) or a "no object" class.

DETR的结构很简单，包括三个部分：一个提取图像特征的CNN，一个编码-解码的transformer，一个用来预测最终目标的前向网络FFN。

实现起来也很简单，只要有CNN和transformer的框架都可以实现，几百行代码就能实现个训练，50多行就实现个预测。

骨架网络 一般情况下输入是 $R^{3\times H_0\times W_0}$ ，输出 $R^{2048\times \frac{H_0}{32}\times \frac{W_0}{32}}$ 。

transformer 编码器 使用1x1的卷积降低特征的通道数到d。encoder的输入是一个序列，所以将特征的size调整到d x HW。每个encoder层又 multi-head self-attention模块和FFN组成。由于transformer对排列顺序不敏感，所以我们加入了位置的编码，并添加到所有attention层的输入。

transformer 解码器 与常规transformer的区别就是，本文可以并行的解码，而之前的transformer都是自回归的依次解码。由于decoder也是对排列顺序不敏感，这N个嵌入必须不一样，才能预测不同的结果。这些输入的嵌入是学到的位置编码，我们称之为object queries，类似于encoder，我们把它们加到每个decoder的输入。由于用了transformer，我们可以学习全局的信息。

FFN 由三层的感知器计算，使用relu，隐层的size为d，线性的映射层。使用softmax输出类别概率。

附加的解码loss 使用附加的loss对模型的训练有帮助，我们在每一个decoder层后面加上FFNs和匈牙利loss。所有FNNs共享权重。我们使用共享的layer-norm 去归一化不同decoder层的输出。

剩下的有时间补吧。

编辑于 2021-03-31 16:55

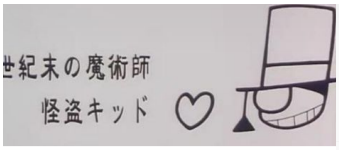
目标检测 Transformer

文章被以下专栏收录

AI论文阅读笔记

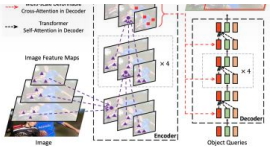
转载

推荐阅读



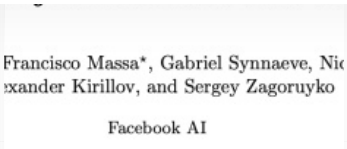
论文翻译：End-to-End Object Detection with…

怪盗kid 发表于目标检测相...



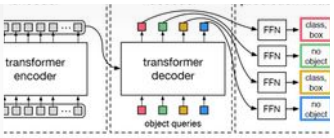
【Deformable DETR】Deformable DETR:…

煎饼果子不... 发表于分割 / ...



DERT End-to-End Object Detection with Transformers

weixi... 发表于机器之脑



End-to-End Object Detection with Transformers-DETR

HUST潘潘

4 条评论 切换为时间排序

写下你的评论...

风中的青竹 01-20

这个有时间再补，后面就没时间了🙄

赞

布尔佛洛哥哥 (作者) 回复 **风中的青竹** 02-15

哈哈😄

赞

哎呀的海角 2021-11-25

object queries是学到的位置编码是从哪里来的呀?

赞

布尔佛洛哥哥 (作者) 回复 **哎呀的海角** 02-15

可以看下我另外一篇源码解读，写得比较详细😄

赞