# 大白话用Transformer做Object Detection

CV开发者都爱看的 极市平台 2024年12月08日 22:00 韩国




作者丨秋名山车神@知乎（已授权）

来源丨https://zhuanlan.zhihu.com/p/503011317

编辑丨极市平台

极市导读

本文为作者隔离期间学习的DETR系列文章的总结记录，内容追求简单、清晰、易懂。主要介绍了DETR的基本原理和针对DETR缺点的改进工作。>>加入极市CV技术交流群，走在计算机视觉的最前沿

## 1 大白话Attention

理解Attention是读懂Transformer[2]论文的第一步，说白了就是一个公式：

$$Attention(q,k,v) = softmax(qk^T)v$$

其中q=fc(a)，k=fc(b)，v=fc(b)。如果a==b就是Self-attention（ViT中全是这玩意）；如果a!=b就是Cross-attention（一般应用于Transformer decoder）。注意这三个fc层不共享参数。简单起见，省略了scaling factor（不影响理解）。

那么如何理解这个公式呢？Attention的本质就是加权：一部分重要，其它部分不重要；或者说一部分相关，其它部分不相关。上式中的加权是基于k对于q的相似度。举一个直观的例子：特征提取的目的是寻找高富帅，q (query) 代表一个理想中标准的高富帅，k (key) 代表每个真实候选人的身高、财富和样貌，v (value) 就是每个真实候选人的特征。那么一个候选

人越符合标准的高富帅条件，就会被赋予更高的权重，特征占比也就越大。

## 2 为什么要用Transformer做目标检测?

DETR的本质是基于查询(query)的目标检测，而目标检测的本质是一种image-to-boxes的转换。相比于CNN时代基于锚框或锚点 (anchor box or anchor point) 的检测方法[3,4,5]，基于query的检测机制其实更加符合image-to-boxes的范式：encoder中的一个元素代表图像上的一块区域 (patch embedding)，而decoder中的一个元素代表一个物体 (object embedding)。Image-to-boxes的转换是基于区域与区域间的、区域与物体间的、物体与物体间的信息交换。整体的思路其实非常简单、直接、合理。

**DETR具有两大核心优势：**

**1. End-to-end detection。** Anchor-based目标检测器大多采用一对多的标签分配算法，因此NMS成为一项必不可少的后处理步骤（去除冗余框）。最近也有一些基于CNN的工作，通过探索一对一的标签匹配[6]，实现了nms-free的目标检测（然而精度提升并不明显，应用于YOLOX甚至还有些许掉点）。对于DETR，端到端检测就显得尤为自然直接。除了一对一的二分匹配 (bipartite matching)，Transformer机制引入了query间的信息交换 (Self-attention in decoder)，来防止多个query收敛到同一目标。相似的操作也被Sparse RCNN[7]所采用。也许样本(anchor/query/proposal)间的信息交换才是实现end-to-end detection的关键。

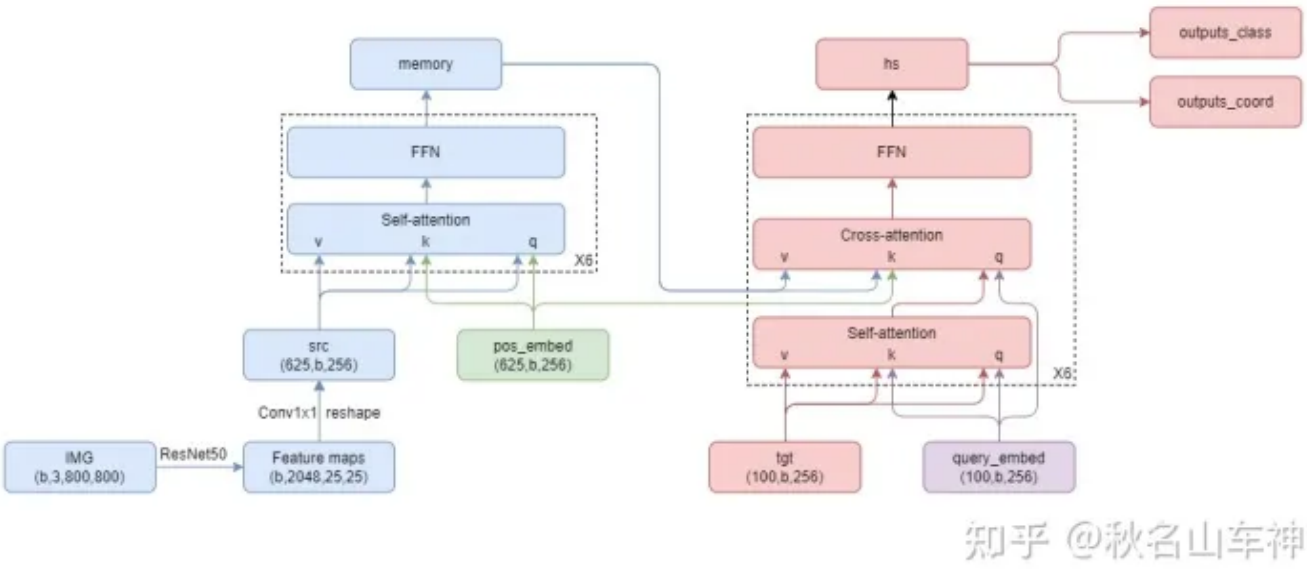
**2. 解耦输入与输出空间。** 在Transformer的逻辑里，图片被展开成一维序列(sequence)，由positional embedding描述绝对位置信息来维系图片形式，其中encoder应用一套positional embedding，decoder应用另一套positional embedding。这其实给了模型解耦输入与输出空间的能力：比如输入空间为图片上均匀采样的点(stride=32)，而输出空间为图片上随机分布的100个点；比如输入空间为多个环视相机视角，输出空间为BEV视角。

换一个角度思考，query和anchor、proposal本质上是一种东西，都是对于图片上潜在物体的刻画。得益于Attention机制，query获得了全局感受野和样本间信息交换的能力，达成了稀疏采样 (sparse sampling) 和端到端检测 (end-to-end detection)。

## 3 DETR网络结构

基于anchor的目标检测器的大体可分为三个组成部分：backbone（特征提取）、neck（多尺度特征聚合）、head（分类与回归预测）。DETR延续了这个结构：backbone（特征提取）、encoder（特征聚合）、decoder（query精修）。区别在于后两个结构（encoder和

decoder）都是由基于Attention机制的Transformer实现。




DETR网络结构，add/Norm/残差连接被省略

### 3.1 Transformer encoder

先用一个1*1卷积降低CNN提取的特征维度(b,2048,h,w ==> b,256,h,w)，再展开成一维序列(b,256,h,w ==> h*w,b,256)，记为src；然后准备好一个相同形状的positional embedding（计算方法参考这里：https://github.com/facebookresearch/detr/blob/8a144f83a287f4d3fece4acdf073f387c5af387d/models/position_encoding.py#L12 ），记为pos_embed。最后重复6次Self-attention和FFN，其中Self-attention的k=fc(src+pos_embed)，q=fc(src+pos_embed)，v=fc(src)；FFN就是两层fc。输出记为memory。

### 3.2 Transformer decoder

准备100个object queries，形状为(100,256)，初始化为0，记为tgt。准备其对应的相同形状的positional embedding，随机初始化，记为query_embed。训练时两者都扩充为(100,b,256)。最后重复6次Self-attention、Cross-attention和FFN，其中Self-attention的k=fc(tgt+query_embed)，q=fc(tgt+query_embed)，v=fc(tgt)；Cross-attention的k=fc(tgt+query_embed)，q=fc(memory+pos_embed)，v=fc(memory)。

直观上理解，Cross-attention就是每个query根据各自感兴趣的区域从图片中抽取相关信息，而Self-attention就是所有query开会决定谁当大哥（前景），谁当小弟（背景）。

我还想简单解释一下object queries (tgt) 和其对应的positional embedding (query_embed)的初始化：object query装载的是图片上的物体信息，在进入decoder之前模型其实对图片

上的物体一无所知，所以作者将他初始化为0。positional embedding装载的是每个query所关注的位置和区域，作者希望这100个query能尽可能均匀的覆盖到整张图片，所以采用随机初始化。

其他诸如Prediction FFN、Bipartite matching loss、Deep supervision等细节，比较容易理解，这里就不赘述了。

**DETR并不是对传统anchor-based detectors的降维打击。相反，DETR存在收敛速度慢、检测精度差、运行效率低等问题。**

碎碎念：CVPR2022收录了至少4篇DETR相关的检测论文，用transformer做object detection算是一个很promising的研究方向了，值得关注。

得益于Transformer带来的动态感受野和样本间信息交换的能力，DETR解锁了稀疏采样 (sparse sampling) 和端到端检测 (end-to-end detection) 两个技能。

然而原始DETR也存在一个比较明显的缺点，就是需要非常长的训练周期才能收敛（在COCO数据集上要训500个epoch）。DETR的大部分后续工作都尝试针对这个缺点做出改进。

## 4 为何DETR难以收敛?

根据作者的设想，每个object query会根据各自感兴趣的区域通过Transformer decoder里的Cross-attention layer从图片中抽取相应的物体特征。

$$\text{Cross\_attention}\left(q_{obj}, k_{img}, v_{img}\right) = \text{softmax}\left(q_{obj}k_{img}^{T}\right)v_{img}$$

这个抽取特征的过程包含两个步骤，一个是key (image features) 对于query (object queries) 的相似度匹配，一个是依据匹配结果对value (image features) 进行加权平均。

然而理想很丰满，现实很骨感。问题就出在这第一个步骤上：由于query embedding是随机初始化的，object queries和image features在模型训练的初期无法正确匹配。

直观上理解，一把钥匙 (object query) 开一把锁（图片上某一特定区域的物体）。但是由于钥匙是随机初始化的，导致它实际上开不了任何一把锁（图片上任意位置的特征对于object query的相似匹配度都很低）。结果就是Cross-attention layer实际上几乎均匀地抽取了整张图片的特征，而不是有针对性的抽取特定区域内的物体特征。可以想象，在这之后的object query不仅采集了目标物体的特征，还采集了图片背景和其他不相关物体的特征，因此用它来分类和定位目标物体还是很困难的。

换个角度考虑，Cross-attention可以想象成是一种软性的RoIAlign：从图片中依据kq相关性

(attention map) 划分出感兴趣区域 (region of interest) 并收集相应特征。DETR难以收敛的原因就是由于object query和image feature间的不对齐 (misalignment)，导致无法有针对性的收集特定区域的物体特征，而是收集到了图片上其他很杂乱的特征。

简单补充一下为什么基于CNN的检测器没有这个问题：Two-stage detectors是直接利用RoI Align操作对齐了region proposal和image features；对于One-stage detectors，anchor box的中心是依据所处特征的image coordinate设定的，大小是依据所处金字塔的feature scale设定的，所以大体上也是对齐的（参考AlignDet[1]）。

## 5 一系列改进工作

Deformable DETR[2]：既然原始的Cross-attention layer自由度太高，没有办法focus到特定区域，那就为每个object query设定需要关注的中心点 (reference point)，并且提出的Deformable attention也只采样中心点附近的N个图片特征，这样既加速了收敛又减少了计算量。并且由于Deformable attention的计算量与特征图大小无关，还可以采用多尺度特征图来优化小目标的检测。

SMCA DETR[3]：在计算Cross-attention之前，每个query先预测一下要检测物体的位置和长宽（有点anchor的味道了），再根据预测的物体位置和长宽生成一个对应的高斯热图 (Gaussian-like weight map)，并融入Cross-attention里kq相似度匹配计算中。这样，每个query抽取的特征就被限制在物体的中心附近，收敛速度也因此得到提升。

Anchor DETR[4]：顾名思义，直接将anchor point的位置编码为object query，并且encoder和decoder共用一套位置编码的方式 (Sine encoding + MLP)。这样encoder和decoder的位置部分 (positional embedding) 就是对齐的，每个query抽取的特征也就集中在anchor point附近了。此外，为了检测同一位置的不同物体，作者还提出为每个anchor point添加不同的模式（multiple patterns，一般是3种，有点类似每个位置设置大小和长宽比不同的3种anchor box）。

DAB DETR[5]：相对Anchor DETR做了两个方面的优化，一是将anchor box（包括位置长宽4个维度）编码为object query，而不是仅仅编码anchor point的位置信息；二是应用了cascade思想，每层不断精修anchor box（上一层的输出的作为下一层的输入）。值得注意的是作者利用所预测的box长宽进一步限制了Cross-attention里的kq相似度匹配计算，使生成的注意力图能够拟合不同位置、不同尺度、不同长宽比的物体。

SAM DETR[6]：为了在语义空间上对齐object queries和image features，作者直接在Cross-attention layer前加了一个RoIAlign操作，即先从image features里切出物体特征，再重新采样8个显著点 (salient points re-sampling)，用以生成语义对齐的query embedding和pos

itional embedding。这里有个小小的疑问，Cross-attention就是为了提取image feature上各个query所对应的物体特征，完成作者的这些操作以后，原本的Cross-attention还有必要吗?

总结一下，由于object query和image feature间（位置上的和语义上的）不对齐，导致Transformer decoder中的Cross-attention layer难以精确地匹配到待检测物体所对应的特征区域，object query也因此采集到了很多除目标物体以外的无关特征，最终导致DETR收敛缓慢。上面介绍的几个工作都是通过不同的方式限制了object query的采样区域，使得网络能够更快的地聚焦于物体区域，所以加速了训练。

## 6 Future direction

探讨object query的数量对于检测精度的影响。理论上100已经远远大于图片上可能出现的物体个数了，然而更多的query还是会带来更高检测精度。直观上思考，越少的query会导致各个query的搜索范围变大，并且难以检测同一位置的多个目标；过多的query又会导致难以抑制多个query收敛到同一物体的情况。那么多少才是合适的query数量呢？每层需求的query数量相同吗?

由于需要精准定位物体，DETR必须能够很好地编码特征的绝对/相对位置关系。DETR目前所采用地Positional embedding是否是最佳方案？或许该在Transformer里塞一些卷积层，或者是否能够从query based patch localization角度，构造一个自监督训练框架?

还有one to one的匈牙利标签匹配还没有人动过，这会不会也是造成DETR收敛慢的原因呢?大家觉得怎么样呢 :-)

参考文献：


- [10] Fast Convergence of DETR with Spatially Modulated Co-Attention https://arxiv.org/abs/2101.07448
- [11] Anchor DETR: Query Design for Transformer-Based Object Detection https://arxiv.org/abs/2109.07107
- [12] DAB-DETR: Dynamic Anchor Boxes are Better Queries for DETR https://arxiv.org/abs/2201.12329
- [13] Accelerating DETR Convergence via Semantic-Aligned Matching https://arxiv.org/abs/220