Pix2seq: A Language Modeling Framework for Object Detection

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, Geoffrey Hinton
Google Research, Brain Team

文 | ZenMoore
编 | 小瑶

图灵奖大佬 Geoffrey Hinton 的团队和 Google Brain 团队近日发布新工作 Pix2seq，**将 CV 经典任务 目标检测 转换为了语言模型的下游任务**。

这就很有意思了朋友们！因为这是一个很一般化的范式！也就是说，不光是目标检测，我们**可以把语言作为中介接口，尝试将一切视觉上的任务映射为序列任务**。这颇有点通用人工智能的意思。

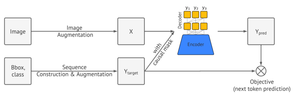
所以，是不是万物皆可 LM 的时代真的要到来了？

论文标题：
Pix2seq: A Language Modeling Framework for Object Detection

论文链接：
https://arxiv.org/abs/2109.10852

模型框架

整个模型由四个部分组成，分别是图像数据增强，序列构造和数据增强，模型结构以及损失函数。



▲本文提出的模型

图像数据增强

图像数据增强没什么新奇的，就是为了扩充数据集，可圈可点的是后面几个部分。

序列构造

目标检测的目标一般是通过 Bbox 框和相应的目标类别组成，Bbox 用四个点的坐标组成 $[y_{min}, x_{min}, y_{max}, x_{max}]$，类别用一个指标变量 c 来表示，我们希望把这个目标输出转换为像语言一样的离散序列，主要是两个步骤：**量化(Quantization)** 和 **序列化(Serialization)**。

量化需要把连续的坐标均等地分为离散的坐标值，用 $[1, n_{bins}]$ 来表示（整数），n_{bins} 的选取很讲究，可大可小，不同的大小决定了量化的精细的大小尺度。例如，600×600 的图像，最大的 n_{bins} 可以是 600，实验表明，$n_{bins} = 500$ 就足矣！这样，$[y_{min}, x_{min}, y_{max}, x_{max}]$ 就可以表示成离散的 token. 还剩下一个 c, 我们不用管，因为它本来就是离散的。

序列化需要把图像中的所有目标整理到一起，在量化中，我们把一个目标用五个离散的 token $[y_{min}, x_{min}, y_{max}, x_{max}, c]$ 来表示了，在这个步骤中，我们把图像中的多个目标的离散 token 表示按照一定的顺序线性地排列起来，实验证明，随机的排列顺序会取得更好的效果。

模型结构

本文采用的是**编码器-解码器**的结构，例如 Transformer. 通过**自回归**的方式生成输出序列。

损失函数

训练的目标非常简单，即语言模型中最普遍不过的**极大对数似然**！四两拨千斤，简洁才是美！

$$max \sum_{j=1}^{L} w_j log P(\hat{y}_j \mid \boldsymbol{x}, \boldsymbol{y}_{1:j-1})$$

其中，y 和 $\hat{y}$ 分别是输入序列和目标序列（在一般的语言模型中，二者是相同的），L 是目标序列长度，w_j 是预先指定的第 j 个 token 的权重（本文都设置成了 1，当然也可以使用其他方式进行设置），x 是给定的图像。

在 inference 阶段，我们根据条件概率对下一时刻的 token 进行采样，可以选择似然最大的 token，但更好的方式是使用 **Nucleus 采样**，以提高**召回率**。最后，当得到 EOS 这个 token 的时候，结束生成，经过量化的逆操作得到 Bbox 和 Class.

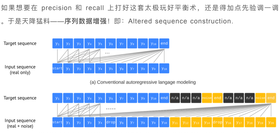
序列数据增强

介绍到这里，好像一切都很完美……

▲BUG!

问题出在哪儿了呢？实验表明，序列生成往往过早就结束了，导致很多目标都被漏掉了。可能是因为数据标注的噪声以及目标识别或定位的不确定性，所以作者想到的 trick 是：人为降低似然，**延迟生成 EOS**，提高召回率！然后就被打脸了……这只带来了很多噪声，以及重复的检测结果。

这又是为啥？作者觉得这主要是因为**模型不依赖于任务**，因为去掉了太多任务的先验知识，所以如果想要在 precision 和 recall 上打好这套太极玩好平衡术，还是得加点先验调一调味儿。于是天降猛料——**序列数据增强！**即：Altered sequence construction.



▲序列数据增强

我们在输入序列 y 的后面加一些**人为制造的噪声 token**，可以是已检测出的真实目标的随机缩放平移，也可以是完全随机的 box 和类别，然后在目标序列 $\hat{y}$ 上，给噪声 token 设置成“noise”这个特殊的类别，相应的坐标都表示为 “N/A”，损失权重 w_j 要设置为零。

因此在 inference 的时候，我们让模型预测最大长度的序列，在重构 box 和 class 的时候，用似然最大的实际类别替换 noise 类别，并将似然作为其打分。

看到这里，不得不说，Hinton 不愧是 Hinton… 这也能搞 work…

⚡ 实验结果 ⚡

实验结果非常的够看啊！

Method	Backbone	#params	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN	R50-FPN	42M	40.2	61.0	43.8	24.2	43.5	52.0
Faster R-CNN+	R50-FPN	42M	42.0	62.1	45.5	26.6	45.4	53.4
DETR	R50	41M	42.0	62.4	44.2	20.5	45.8	61.1
Pix2seq (Ours)	R50	37M	**43.0**	61.0	45.6	25.1	46.9	59.4
Faster R-CNN	R101-FPN	60M	42.0	62.5	45.9	25.2	45.6	54.6
Faster R-CNN+	R101-FPN	60M	44.0	63.9	47.8	27.2	48.1	56.0
DETR	R101	60M	43.5	63.8	46.4	21.9	48.0	61.8
Pix2seq (Ours)	R101	56M	**44.5**	62.8	47.5	26.0	48.2	60.3
Faster R-CNN	R50-DC5	166M	39.0	60.5	42.3	21.4	43.5	52.5
Faster R-CNN+	R50-DC5	166M	41.1	61.4	44.3	22.9	45.9	55.0
DETR	R50-DC5	41M	**43.3**	63.1	45.9	22.5	47.3	61.1
Pix2seq (Ours)	R50-DC5	38M	**43.2**	61.0	46.1	26.6	47.0	58.6
DETR	R101-DC5	60M	**44.9**	64.7	47.7	23.7	49.5	62.3
Pix2seq (Ours)	R101-DC5	57M	**45.0**	63.2	48.6	28.2	48.9	60.4

▲实验结果

总结一下主要是以下两点：

1. 对标 Faster R-CNN：小中型目标差异不大，但在大型目标上，本文的模型表现更好！
2. 对标 DETR：大型目标上差异不大(或者略差一点)，但在小中型目标上，本文的模型表现突出！

⚡ 结论 ⚡

Pix2Seq 是一个简单而通用的目标检测框架，简化了目标检测的 pipeline，消除了大部分先验知识，效果也非常能打！当然，这个架构还可以进行进一步地优化。

作者认为，这个框架不仅适用于目标检测，其他**产生低带宽输出**的视觉任务（即输出可以用简洁的离散 token 序列表示）也可以尝试用这个框架来解决，因此，作者希望将其做成一个**通用统一的接口**以解决各种各样的视觉任务。另外，也希望能让模型减少对于人工标注的依赖，多一点**无监督学习**的能力。

⚡ 最后的话 ⚡

小编认为，这是一个很有开创性意义的工作，或者说学术思想。从哲学的角度讲，如果我们信奉 **萨丕尔－沃尔夫假设(语言决定思维)** 的话，就很容易坚信自然语言的伟大潜力。人类用语言描述世间万物，下到家常小事，上到天文地理，所有的任务，都可以用自然语言来表示输入和输出，因此我们坚信语言具有非常强大甚至是接近于无限的表达能力：**Language is the embedding of everything！** 回到本文，Hinton 成功地将目标检测这一个典型的视觉任务转化成了语言的任务，那么我们是不是可以猜想，一切任务都能用序列来解决：**All in Seq！** 如果真的如同萨丕尔和沃尔夫所说，人类的思考过程都是基于语言的（**即人类通过心中语言整理和推演自己的思路**），那么，我们是不是可以不断地发掘本文的潜力，找到**机器推理**的密码？Hinton 作为心理学家出身的 Aler，不知道对此究竟是怎么思考的……

所以，是有一个 **“宇宙”** 蕴含在这篇论文中的！欢迎大家进行思考与讨论，即便是科幻也无妨（比如在知乎上或者评论区等等）。