**用Transformer完全替代CNN**

链接：<https://zhuanlan.zhihu.com/p/266311690>

An Image Is Worth 16X16 Words: Transformers for Image Recognition at Scale

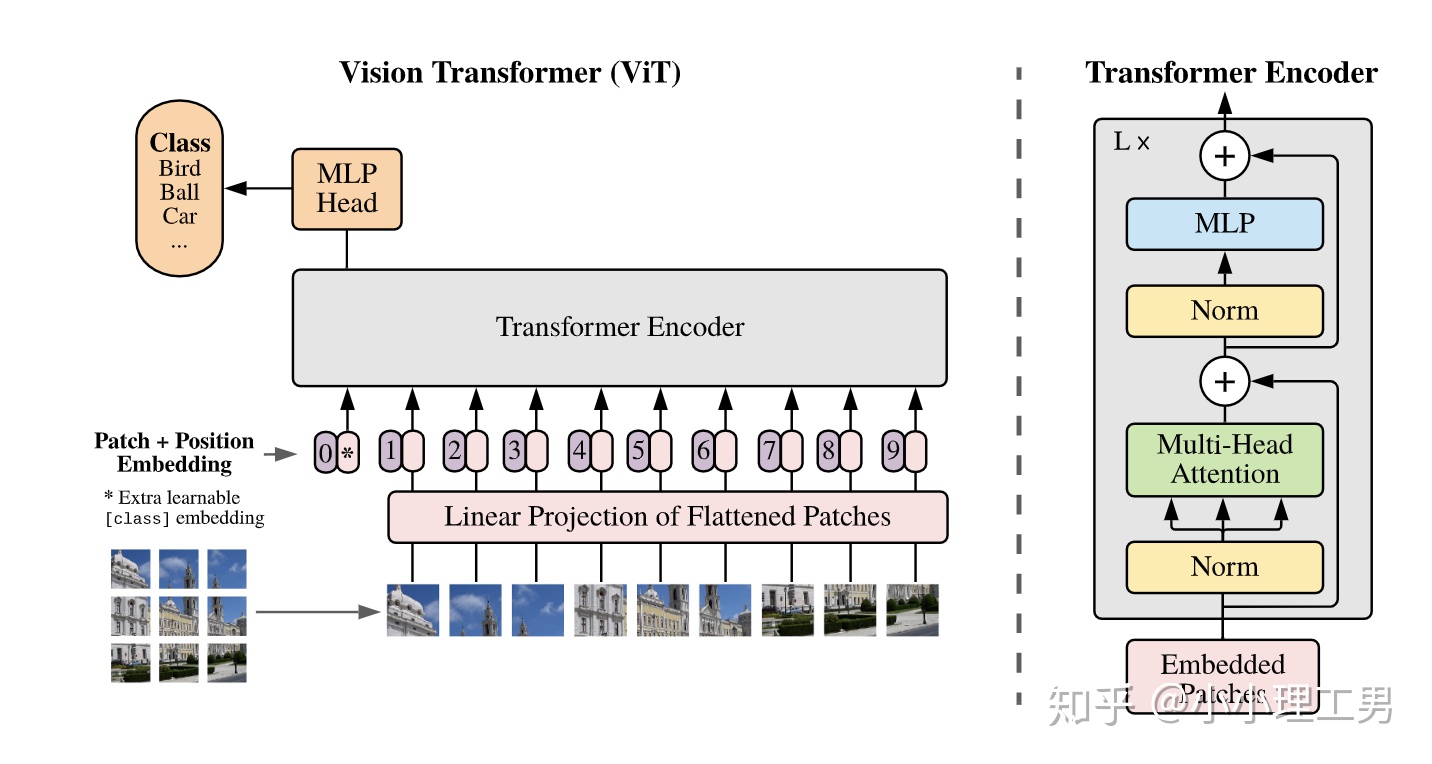
ICLR 2021 Oral

Transformer大多用于nlp领域，即序列化的数据，但是cv领域的图片因为是三维的（长宽channels），所以以往的迁移工作集中于两个方面：

1. 将self-attention机制与常见的CNN架构结合；
2. 用self-attention机制完全替代CNN。

本文也是第二种，不同的是，本文尽可能不修改transformer。而且在中等规模的数据集上（例如ImageNet），transformer模型的表现不如ResNets；而当数据集的规模扩大，transformer模型的效果接近或者超过了目前的一些SOTA结果。

整体框架：



1. 将图像转化为序列化数据，将图片分成N个P\*P\*channels大小的patch，将每个patch转化为一维的P\*P\*channels长度的向量，称为flattened patch。
2. 为了避免P的大小的影响，使用线性映射将flattened patch转化为固定长度D的向量，即最终得到N\*D的二维矩阵
3. 由于transformer模型本身是没有位置信息的，和NLP中一样，我们需要用position embedding将位置信息加到模型中去。如图所示编号有0-9的紫色框表示各个位置的position embedding，而紫色框旁边的粉色框则是经过linear projection之后的flattened patch向量。文中采用将position embedding（即图中紫色框）和patch embedding（即图中粉色框）相加的方式结合position信息。
4. Learnable embedding。带星号的粉色框（即0号紫色框右边的那个）不是通过某个patch产生的。这个是一个learnable embedding（记作 X），其作用类似于BERT中的[class] token。在BERT中，[class] token经过encoder后对应的结果作为整个句子的表示；类似地，这里X经过encoder后对应的结果也作为整个图的表示。

为什么BERT或者这篇文章的ViT要多加一个token呢？因为如果人为地指定一个embedding（例如本文中某个patch经过Linear Projection得到的embedding）经过encoder得到的结果作为整体的表示，则不可避免地会使得整体表示偏向于这个指定embedding的信息（例如图像的表示偏重于反映某个patch的信息）。而这个新增的token没有语义信息（即在句子中与任何的词无关，在图像中与任何的patch无关），所以不会造成上述问题，能够比较公允地反映全图的信息。此时的向量是N+1个D维向量。

1. Transformer encoder，结构如图。

文中还提出了一个比较有趣的解决方案，将transformer和CNN结合，即将ResNet的中间层的feature map作为transformer的输入。

和之前所说的将图片分成patch然后reshape成sequence不同的是，在这种方案中，作者直接将ResNet某一层的feature map reshape成sequence，再通过Linear Projection变为Transformer输入的维度，然后直接输入进Transformer中。