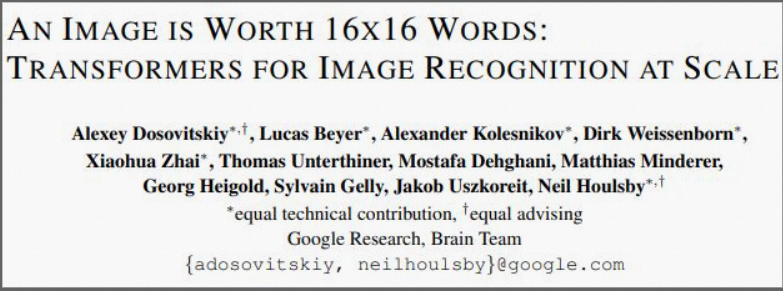
# ViT(Vision Transformer)详解



**1.论文概述**

1）ViT是2020年Google团队提出的将Transformer应用在图像分类的模型，成为了transformer在CV领域应用的里程碑著作。

**2）把最重要的说在最前面：**

当拥有足够多的数据进行预训练的时候，ViT的表现就会超过CNN，突破transformer缺少归纳偏置的限制，可以在下游任务中获得较好的迁移效果

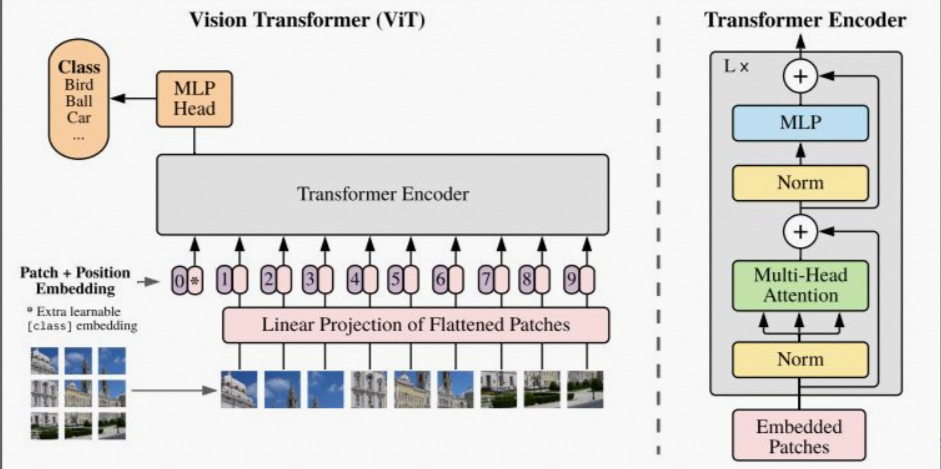
当训练数据集不够大的时候，ViT的表现通常比同等大小的ResNets要差一些，因为Transformer和CNN相比缺少归纳偏置（inductive bias），即一种先验知识，提前做好的假设。

3）CNN的归纳偏置：

一种是局部性（locality/two-dimensional neighborhood structure），即图片上相邻的区域具有相似的特征；一种是平移不变形（translation equivariance）。

当CNN具有以上两种归纳偏置，就有了很多先验信息，需要相对少的数据就可以学习一个比较好的模型

**2.ViT的网络结构图**



**知乎详解：比较详细直接COPY过来了**

<https://zhuanlan.zhihu.com/p/445122996>

1）ViT将输入图片分为多个patch（16x16），再将每个patch投影为固定长度的向量送入Transformer，后续encoder的操作和原始Transformer中完全相同。但是因为对图片分类，因此在输入序列中加入一个特殊的token，该token对应的输出即为最后的类别预测。

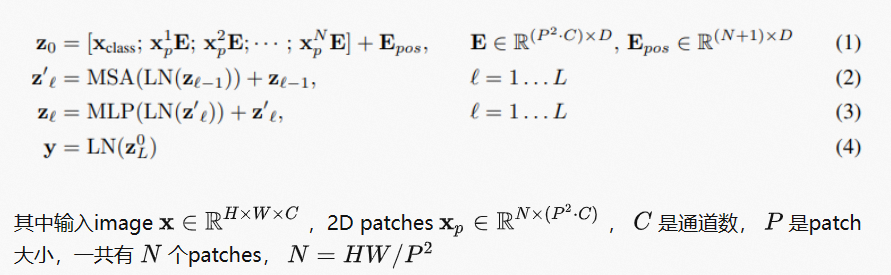
2）patch embedding：例如输入图片大小为224x224，将图片分为固定大小的patch，patch大小为16x16，则每张图像会生成224x224/16x16=196个patch，即输入序列长度为**196**，每个patch维度16x16x3=**768**，线性投射层的维度为768xN (N=768)，因此输入通过线性投射层之后的维度依然为196x768，即一共有196个token，每个token的维度是768。这里还需要加上一个特殊字符cls，因此最终的维度是**197x768**。到目前为止，已经通过patch embedding将一个视觉问题转化为了一个seq2seq问题。

3）positional encoding（standard learnable 1D position embeddings）：ViT同样需要加入位置编码，位置编码可以理解为一张表，表一共有N行，N的大小和输入序列长度相同，每一行代表一个向量，向量的维度和输入序列embedding的维度相同（768）。注意位置编码的操作是sum，而不是concat。加入位置编码信息之后，维度依然是**197x768**

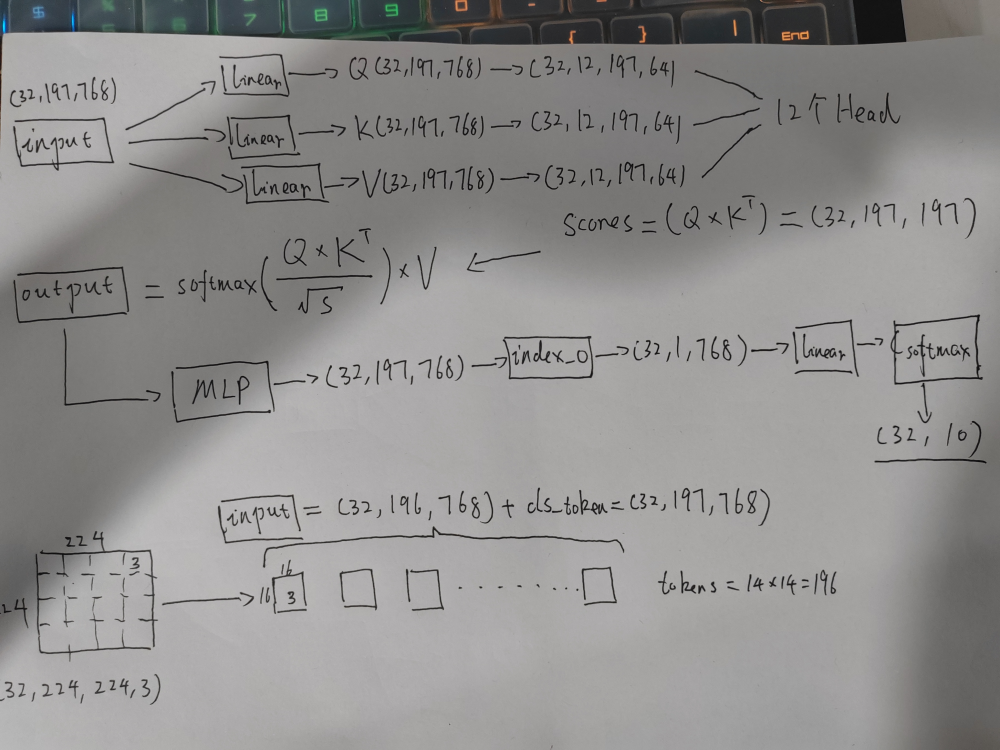
**4）**LN/multi-head attention/LN：LN输出维度依然是197x768。多头自注意力时，先将输入映射到q，k，v，如果只有一个头，qkv的维度都是197x768，如果有12个头（768/12=64），则qkv的维度是197x64，一共有12组qkv，最后再将12组qkv的输出拼接起来，输出维度是197x768，然后在过一层LN，维度依然是**197x768**

**5）**MLP：将维度放大再缩小回去，197x768放大为197x3072，再缩小变为**197x768**

一个block之后维度依然和输入相同，都是197x768，因此可以堆叠多个block。最后会将特殊字符cls对应的输出 作为encoder的最终输出 ，代表最终的image presentation（另一种做法是不加cls字符，对所有的tokens的输出做一个平均），如下图公式(4)，后面接一个MLP进行图片分类



6）我个人的理解和代码里的流程图：



7）注意力格式化：

