# Vision Transformer

## 一、引言

Transformer自提出以来，在NLP领域大放异彩，但是在CV领域的应用还是十分有限的。在CV中，自注意力机制要么和卷积神经网络一起使用，要么就是将卷积神经网络的卷积替换成自注意力，并未脱离卷积神经网络的框架。

本文证明了将一个纯的Transformer模型直接应用于图像分类任务时，也可以表现的非常好。尤其在大规模的数据集进行预训练后所展现的性能甚至优于目前最好的卷积神经网络。同时，一个纯的VIT模型，需要更少的训练资源，即耗时更短。

## 本文的主要贡献：

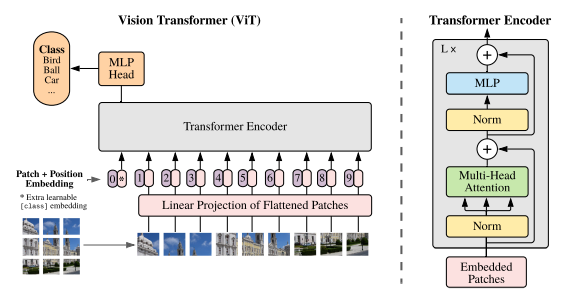
·挑战自2012年以来，由AlexNet提出的CNN在计算机视觉里的统治地位

·打破CV 与 NLP两个领域的次元壁，开创CV的一个新时代

·证明VIT模型在大规模视觉分类任务上的表现要优于目前最好的CNN模型

·实验证明VIT模型的训练成本更低，具备工程实用性

## 二、模型介绍



图一为论文原文中给出的模型框架，框架主要由三个模块组成：

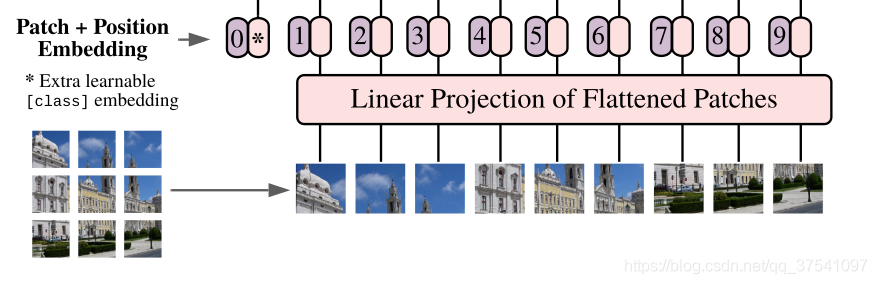
* Linear Projection of Flattened Patches(Embedding层)
* Transformer Encoder(Transformer框架中的一个模块)
* MLP Head（最终用于分类的层结构）

### 2.1 Embedding 层结构

**Embedding层的主要作用：将图像的三维数据转化为标准Transformer模块所需要的二维矩阵数据。**

****

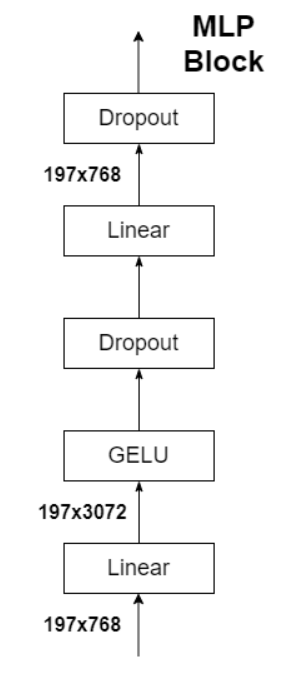
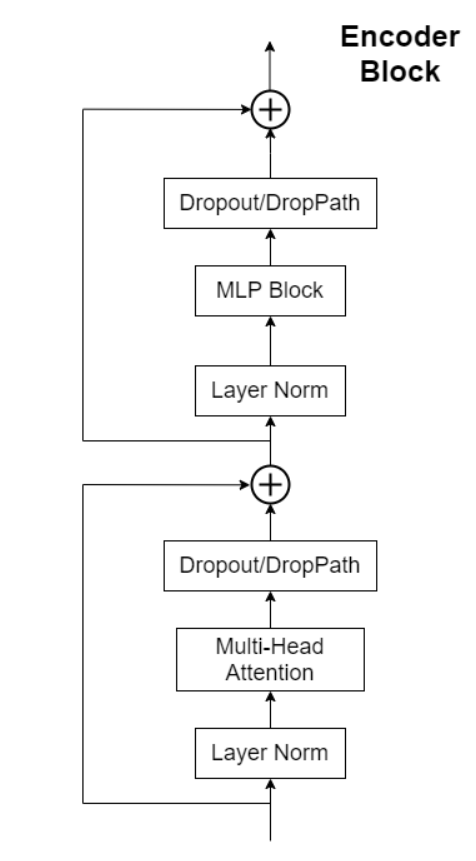
对于标准的Transformer模块，要求输入的是token（向量）序列，即二维矩阵[num\_token, token\_dim]，如下图，token0-9对应的都是向量，以ViT-B/16为例，每个token向量长度为768。



如上图所示，首先将一张图片按给定大小分成一堆Patches。以ViT-B/16为例，将输入图片(224x224)按照16x16大小的Patch进行划分，划分后会得到 (224/16)^2=196个Patches。接着通过线性映射将每个Patch映射到一维向量中，以ViT-B/16为例，每个Patch数据shape为[16, 16, 3]通过映射得到一个长度为768的向量（后面都直接称为token）。[16, 16, 3] -> [768]

在输入Transformer Encoder之前，需要加上[class]token以及Position Embedding。 在原论文中，作者说参考BERT，在刚刚得到的一堆tokens中插入一个专门用于分类的[class]token，这个[class]token是一个可训练的参数，数据格式和其他token一样都是一个向量，以ViT-B/16为例，就是一个长度为768的向量，与之前从图片中生成的tokens拼接在一起，Cat([1, 768], [196, 768]) -> [197, 768]。然后关于Position Embedding就是之前Transformer中讲到的Positional Encoding，这里的Position Embedding采用的是一个可训练的参数（1D Pos. Emb.），是直接叠加在tokens上的（add），所以shape要一样。以ViT-B/16为例，刚刚拼接[class]token后shape是[197, 768]，那么这里的Position Embedding的shape也是[197, 768]。

### 2.2 Transformer Encoder层



Transformer Encoder其实就是重复堆叠Encoder Block L次，上图是Encoder Block，主要由以下几部分组成：

·Layer Norm：层归一化。

·Multi-Head Attention

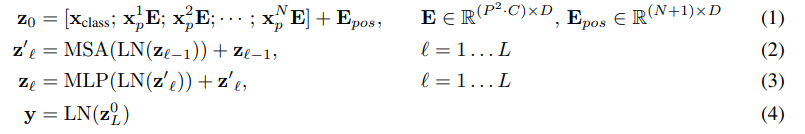
·Dropout/DropPath：主要作用为防止过拟合

·MLP Block，如图右侧所示，就是全连接+ GELU激活函数+Dropout的组成，需要注意的是第一个全连接层会把输入节点个数翻4倍[197, 768] -> [197, 3072]，第二个全连接层会还原回原节点个数[197, 3072] -> [197, 768]。

### 2.3 MLP head

经过encoder结构后，输出的维度维197\*768，此时我们会通过切片的方式提取出Class token的信息，其维度为1\*768。接着会拿这个1\*768维的Class token经过MLP Head层，完成分类。

## 2.4 VIT的前向流程



式（1）即前述的图像块嵌入、CLS向量追加和位置编码；式（2）~（3）则是一个循环：式（2）为MSA部分，包括多头自注意力、残差连接和 层归一化 (Layer Norm) 三个部分；式（3）为MLP部分，包括前馈网络 (FFN)、残差连接和 层归一化(Layer Norm)三个部分。式（4）为层规范化。最后以一个MLP作为分类头 (Classification Head)。

## 三、VIT训练和实验

## 3.1训练方法

ViT的基本训练策略是在大数据集上先做预训练，然后在在小数据集上做迁移使用。ViT做预训练使用到的大数据集包括：

·ILSVRC-2012 ImageNet dataset：1000 classes

·ImageNet-21k：21k classes

·JFT：18k High Resolution Images

其中JFT是一个谷歌的内部大规模图像数据集，约有300M图像18291个类别标注。

ViT预训练迁移到的数据集包括：

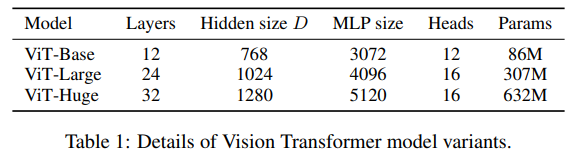
·CIFAR-10/100

·Oxford-IIIT Pets

·Oxford Flowers-102

·VTAB

论文共设计了Base、Large和Huge三款不同大小的ViT模型，分别表示基础模型、大模型和超大模型，三款模型的各参数如下表所示。

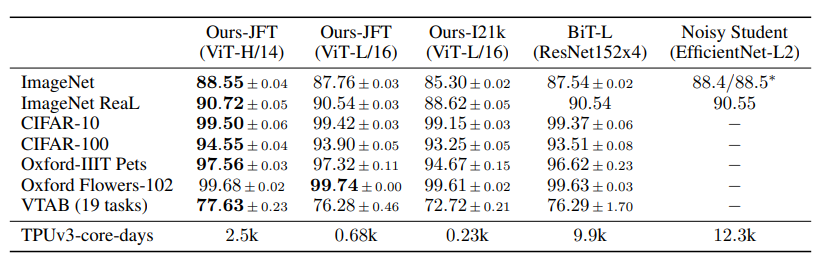


## 3.2实验设计

### 3.2.1性能测试

ViT最核心的实验就是将前述的训练方法进行实现，即在大规模数据集上预训练后迁移到小数据集上看模型效果。为了比对CNN模型，论文特地用了Big Transfer (BiT)，该模型使用大的ResNet进行监督迁移学习，是2020 ECCV上提出的一个大CNN模型。另外一个比对CNN模型是2020年CVPR上的Noisy Student模型，是一个半监督的大型CNN模型。

ViT、BiT和Nosiy Student模型经三大数据集预训练后在各小数据集上的准确率如下表所示。



·上表对比了几个ViT的变体和卷积神经网络（BiT和Noisy Student）

·和BiT做对比的原因是因为bit确实是之前卷积神经网络里做得比较大的，而且也是因为他是作者团队自己本身的工作，所以正好可以拿来对比

·和noisy student做对比是因为它是ImageNet之前表现最好的方法。

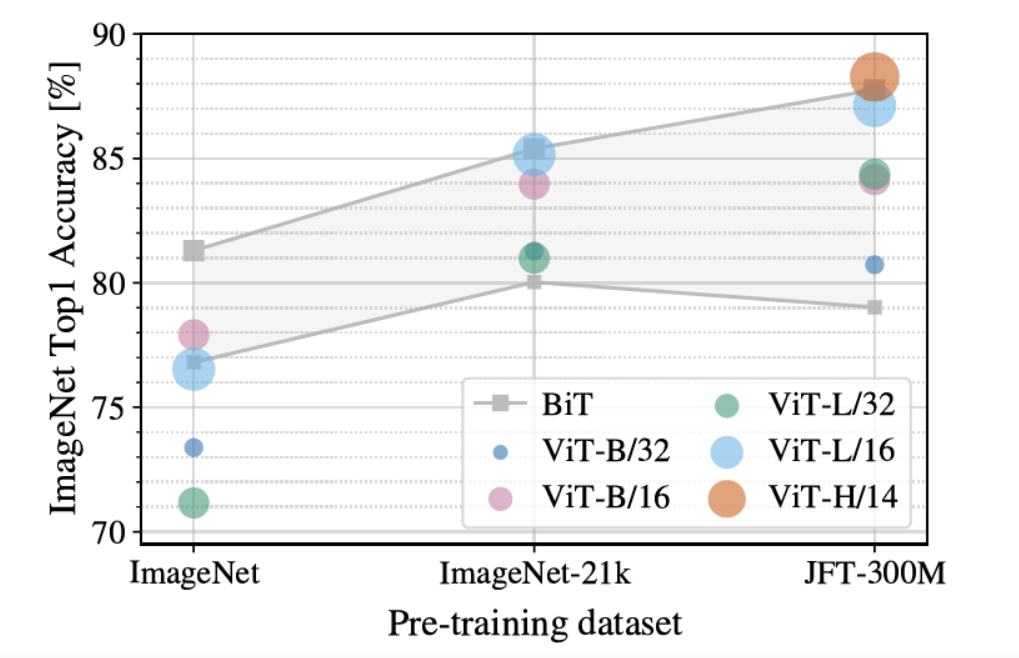
·从上表中可以看出，ViT Huge用比较小的patch 14×14能取得所有数据集上最好的结果。

· 但是因为这些数值都太接近了，仅仅相差零点几个点或者一点几个点，没有特别大的差距，所以作者觉得没有展示出VIT的威力，所以作者就得从另外一个角度来体现VIT的优点：因为训练起来**更便宜**！！！作者所说的更便宜是指最大的vit huge这个模型也只需要训练2500天TPUv3天数，BiT用了9900天，noisy student用了一万多天，所以从这个角度上来说，ViT不仅比之前BiT和Noisy Student要训练的快，而且效果要好，所以通过这两点可以得出VIT优于卷积神经网络的结论。

### 3.2.2对数据集规模要求的实验

在实验一中，可以看到，ViT经过大数据集的预训练后，在各小数据集上的迁移后准确率超过了一些SOTA CNN模型的结果。但要取得这种超越CNN的性能效果，需要大的预训练数据集和大模型的结合。

所以第二个实验就是测试ViT对预训练数据集规模到底有怎样的要求？论文针对此问题做了一个对比实验。分别在ImageNet、ImageNet-21k和JFT-300M进行预训练，三个数据集规模分别为小数据集、中等规模数据集和超大数据集，预训练效果如下图所示：



* 该图表示当时用不同大小的数据集的时候，比如说ImageNet是1.2m，而ImageNet-21k是14m，JFT是300m，当数据集不断增大的时候，resnet和vit到底在ImageNet的fine-tune的时候效果如何
* 该图的主要意思是说，灰色代表bit，也就是各种大小的resnet，最下面表示50，最上面表示152，他所想要展示的是在中间的灰色区域就是resnet能达到的效果范围，剩下的圆点就是各种大小不一的vision transformer
* 在最小的ImageNet上做预训练时，vision transformer是完全不如resnet，vision transformer基本上所有的点都在灰色区域的下面。这说明vision transformer在中小型数据集上做预训练的时候的效果是远不如残差网络的，原因就是因为vision transformer没有使用先验知识（归纳偏置），所以它需要更大的数据去让网络学得更好
* 在ImageNet-21k上做预训练的时候，vision transformer和resnet已经是差不多了，vision transformer基本上所有的点都落在灰色区域内

· 只有当用特别大的数据集JFT-300M时，vision transformer是比bit对应的res152还要高。

### 总结：

1. 如果想用vision transformer，那么得至少准备差不多和ImageNet-21k差不多大小的数据集，如果只有很小的数据集，还是选择使用卷积神经网络比较好。

2. 当已经拥有了比ImageNet-21k更大的数据集的时候，用vision transformer就能得到更好的结果，它的扩展性更好一些。

## 四、一些问题

1 . VIT在较小的训练集上展现的性能弱于BIT， 为什么？

作者对此的解释是：这个看起来不太好的结果其实是可以预期的，因为transformer跟卷积神经网路相比，它缺少了一些卷积神经网络所带有的归纳偏置。这里的归纳偏置其实是指一种先验知识或者说是一种提前做好的假设

对于卷积神经网络而言，常说的有两个inductive bias（归纳偏置）：

1. locality（局部感知性）：因为卷积神经网络是以滑动窗口的形式一点一点地在图片上进行卷积的，所以假设图片上相邻的区域会有相邻的特征，靠得越近的东西相关性越强

2. translation equivariance（平移等变性或平移同变性）：f(g(x))=g(f(x))，就是说不论是先做 g 这个函数，还是先做 f 这个函数，最后的结果是不变的。这里可以把f理解成卷积，把g理解成平移操作，意思是说无论是先做平移还是先做卷积，最后的结果都是一样的（因为在卷积神经网络中，卷积核就相当于是一个模板，不论图片中同样的物体移动到哪里，只要是同样的输入进来，然后遇到同样的卷积核，那么输出永远是一样的）

2. 在Embedding中，使用class token的设计？

在CV中，一般采用GAP（全局平均池化），而不会采用class token的方法，本文为了保持跟原始transformer保持一致。实验表明，在VIT中采用GAP的设计也能达到相同的性能。

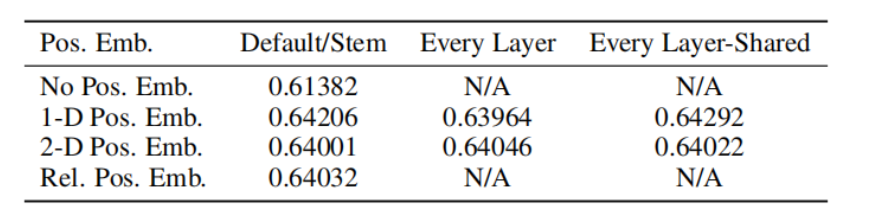
3. 位置编码？

作者也做了很多的消融实验，主要是三种：

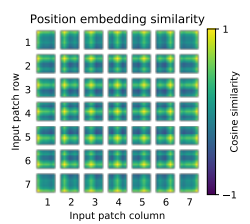
1d：就是NLP中常用的位置编码，也就是本文从头到尾都在使用的位置编码

2d：比如1d中是把一个图片打成九宫格，用的是1到9的数来表示图像块，2d就是使用11、12、13、21等来表示图像块，这样就跟视觉问题更加贴近，因为它有了整体的结构信息。具体的做法就是，原有的1d的位置编码的维度是d，现在因为横坐标、纵坐标都需要去表示，横坐标有D/2的维度，纵坐标也有D/2的维度，就是说分别有一个D/2的向量去表述横坐标和纵坐标，最后将这两个D/2的向量拼接到一起就又得到了一个长度为D的向量，把这个向量叫做2d的位置编码

relative positional embedding（相对位置编码）：在1d的位置编码中，两个patch之间的距离既可以用绝对的距离来表示，又可以用它们之间的相对距离来表示（文中所提到的offset），这样也可以认为是一种表示图像块之间位置信息的方式。



这张图描述的是位置编码的相似性，数字越大相似性越高（-1到1，cosine similarity），横纵坐标分别是对应的patch，如果是同一个坐标，自己和自己相比，相似性肯定是最高的。从图中可以发现，学到的位置编码是可以表示一些距离信息的，同时它还学习到了一些行和列的规则，每一个图像块都是同行同列的相似性更高，也就意味着虽然它是一个1d的位置编码，但是它已经学到了2d图像的距离概念，这也可以解释为什么在换成2d的位置编码以后，并没有得到效果上的提升，是因为1d已经够用了。

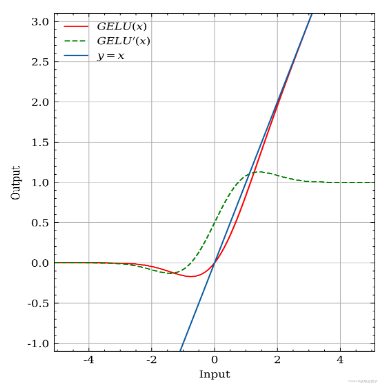


4.MLP中为何用 GELU激活函数？

GELU激活函数：

GELU(x)≈0.5x(1+tanh(2/π​(x+0.044715x3)))≈x.sigmoid(1.702x)

可以理解为，对于输入的值，其服从于标准正态分布 N(0, 1)，它会乘上一个伯努利分布 Bernoulli(Φ(x))，其中Φ(x) = P(X ≤ x)。



·在Transformer 模型中表现最好

·解决梯度消失问题