# Vision Transformer在图像识别中的应用

**标题：**An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale  
**会议：**ICLR

**作者**：Alexey Dosovitskiy，Lucas Beyer，Alexander Kolesnikov，Dirk Weissenborn，Xiaohua Zhai，Thomas Unterthiner，Mostafa Dehghani，Matthias Minderer，Georg Heigold，Sylvain Gelly，Jakob Uszkoreit，Neil Houlsby

**分享人**：陈彦克 20210315064

## 1 研究背景

近几年来，虽然Transformer架构已经成为自然语言处理领域中的一种标准，但它在计算机视觉上的应用仍然十分有限。在计算机视觉领域中，自注意力机制要么与卷积神经网络一起使用，要么在保持原有卷积网络结构不变的情况下，替换局部的卷积运算。在本文中，我们证明了这种自注意力机制对于CNNs的依赖是完全不必要的，即一个纯的Transformer直接作用于一系列图像块的时候，也可以很好地执行图像分类任务。尤其是在大规模的数据集上进行预训练后，迁移到中小型图片数据集上（如ImageNet，CIFAR-100，VTAB等）时，Vision Transformer（Vit）与目前最好的卷积神经网络相比，也可以取得非常优秀的结果，并且需要更少的训练资源。

## 2 关键技术

在模型的设计中，我们尽可能地遵循原始的Transformer架构。这种刻意且简单的设计的一个优点是：我们可以直接使用可扩展的NLP Transformer体系结构。

## 3. 算法介绍

### 3.1 技术难点

以自注意力机制为基础的网络，尤其是Transformer架构，已经是自然语言处理里的必选模型了。现在比较主流的方式是先在一个大规模的数据集上做预训练，然后再在一些特定的小领域的数据集上去做微调。

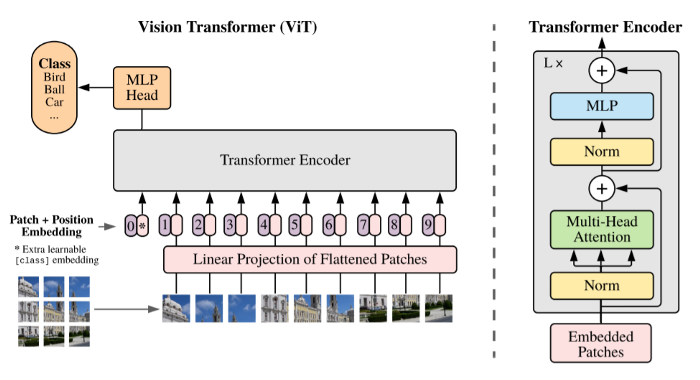
但将Transformer直接运用到计算机视觉领域存在困难。一个问题是，如何将高维的图片数据变成一个一维的序列输入Transformer。最直观的方式就是将高维度的图片拉直，把每个像素点作为Transformer的输入元素。但自注意力机制要求每个像素点都关注其它所有像素点，考虑到图片像素点规模过于庞大，这种实现方式的复杂度高不可攀。

为此，我们的解决方法是：将图像分割成一个个小的图像块patch，并让这些patch经过线性的全连接层的映射后作为Transformer Encoder的输入。

### 3.2 Vision Transformer（ViT）

ViT将输入的原始图像分割为多个patch，并将每个patch进行线性嵌入，通过线性的全连接层投影为固定长度的向量，添加position embedding，并将得到的向量序列送入Transformer，后续Transformer Encoder的操作和原始NLP Transformer中完全相同。由于任务是对图片进行分类，因此我们借鉴NLP Transformer的标准方法，在输入序列中加入一个额外的可学习的分类token，该token对应的输出即为最后的预测的图像类别。

Vision Transformer模型的总览如图1所示。



**图1 ViT模型总览**

按照ViT模型总览图，一个ViT Block可以分为以下几个步骤实现：

**Patch Embedding**

为了解决输入图片像素点数据规模过大的问题，Vision Transformer首先将图片分割为若干个固定大小的patch。

为了处理2D图像，我们将图像变形为一个展平的2D的patches序列，其中为原始图像的分辨率，是原始图像的通道数，是每个图像块patch的分辨率。由此可计算出产生的图像patch数为：

。这亦为ViT的有效输入序列长度。

然后我们将每一个patch当作一个元素，通过一个线性的全连接层的映射后输入Transformer Encoder。

**Position Embedding**

考虑到一张图片是一个整体，每一个patch在原先的图像中是有固定顺序的，因此我们在Patch Embedding的基础上加入Position Embedding，以保留输入图像块patch之间的空间位置信息。

我们尝试使用不同的位置编码方式进行了消融实验，包括以下几个案例：不提供位置编码信息、1D位置编码、2D位置编码和相对位置编码。

**表1 在Image-Net上评价vit-b/16模型对位置编码的消融研究结果**

|  |  |  |  |
| --- | --- | --- | --- |
| Pos. Emb. | Default/Stem | Every Layer | Every Layer-Shared |
| No Pos. Emb. | 0.61382 | N/A | N/A |
| 1-D Pos. Emb. | 0.64206 | 0.63964 | 0.64292 |
| 2-D Pos. Emb. | 0.64001 | 0.64046 | 0.64022 |
| Rel. Pos. Emb. | 0.64032 | N/A | N/A |

由上表结果，我们发现：如果不提供位置编码，那么模型就需要通过图像块的语义来学习拼图，这就额外增加了学习的成本，效果较差；但其它各种编码方式的效果都接近，这主要是因为ViT的输入是相对较大的图像块而非像素点，所以学习位置信息相对容易很多。

**Extra Learnable (Class) Embedding**

借鉴BERT模型中的类别token，此处ViT使用了Extra Learnable (Class) Embedding为图像块patch的序列预设一个可学习的分类字符cls，其位置编码值为0。由于在自注意力机制中，所有的token都在与其他token之间做相互的信息交流，因此这个Class Embedding能够从别的Embedding中学到有用的信息。从而我们只需要根据它的输出做一个分类结果的判断即可。最后会将分类字符cls对应的输出作为Encoder的最终输出，代表最终预测的图像类别。

**Transformer Encoder**

这里使用的是NLP中的标准的Transformer Encoder架构，其由交替的多头自注意力层Multi-Attention和多层感知器MLP构成。在每个Transformer Block前应用层归一化，在每个Block后应用残差连接。Transformer Encoder的具体结构如模型总览图的右边部分所示。

MLP层将图像维度放大再缩小回去。经这种变换后，输入通过一个Block输出维度依然和输入维度相同，因此可以堆叠多个Block。

集合了类别向量、图像块嵌入和位置编码三者到一体的输入嵌入向量后，即可馈入Transformer Encoder。ViT 类似于CNN，不断前向通过由Transformer Encoder Blocks串行堆叠构成的Transformer Encoder，最后提取可学习的类别嵌入向量—— class token对应的特征用于图像分类。

整体前向计算过程如下：

### 3.3 归纳偏置

我们注意到，Vision Transformer比起CNN而言，要少很多这种图像特有的归纳偏置。在CNN中，局部性和平移等变性是在模型的每一层都有体现的，这个先验知识贯穿整个模型始终；但是对于ViT来说，只有MLP层是局部且具有平移等变性的，而自注意力层是全局的。

因此，ViT很少使用图片中的二维信息，仅仅只是在模型开始时通过将图像切割成patch才有体现。除此之外，初始化时的位置编码也是随机的，并没有携带任何关于原图像的二维信息，所有patch之间的空间位置关系都必须从头开始学习。

### 3.4 混合架构

混合结构是Transformer迁移到CV领域的另一种思路。作为将原始图像分割成patch的替代方案，Transformer的输入序列可由CNN经卷积等操作得到的feature map构成。简单来说，就是先用CNN提取图像特征，然后由CNN提取到的特征图feature map作为图像块patch输入Transformer。

### 3.5 微调和更高的分辨率

通常，我们在大型数据集上对ViT进行预训练，并在较小规模的下游任务中进行微调。并且比起预训练，在更高的分辨率下进行微调通常是有益的。当输入更高分辨率的图像时，我们保持patch的大小不变，这会产生更大的有效序列长度。虽然说ViT可以处理任意序列长度（直到内存限制），但是提前预训练的位置编码信息可能不再具有意义。因此，我们根据预训练的位置信息在原始图像中执行二维插值。

但这样一个简单的插值操作可能会让最后的图像识别效果掉点。因此这里所说的插值只是一种临时性的解决方案，这也算是ViT在微调时的一个局限性。

## 4. 实验结果

在这个章节，我们主要对比了ResNet、Vision Transformer（ViT）和混合模型Hybrid的表征学习能力。当考虑到模型预训练的代价，即预训练时间的长短时，Vision Transformer表现得非常好，能在大多数数据集上取得最好的结果。

### 4.1 数据集的选取

为了探究模型的可扩展性，在预训练过程中，数据集主要使用了大规模图像数据集，包括：ImageNet-1k、ImageNet-21k和Google自己的数据集JFT-18k。

下游任务做的都是图像分类，用的也是图像分类中常用的中小规模数据集，包括CIFAR-10、CIFAR-100和Oxford Flower等等。

### 4.2 模型及变体

类似于BERT，ViT原文中共定义了 3 种不同大小的模型：Base、Large和Huge，不同模型对应的模型结构参数不同，如下表所示：

**表2 ViT变体模型的结构参数**

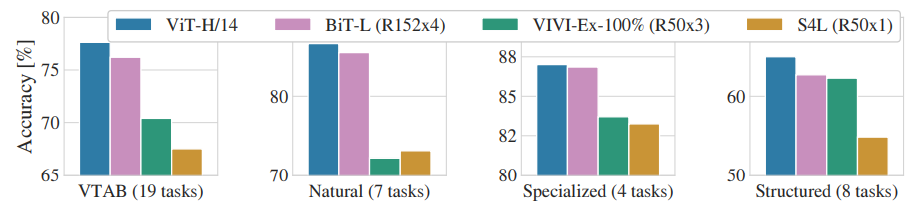
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Layers | Hidden Size D | MLP Size | Heads | Params |
| ViT-Base | 12 | 768 | 3072 | 12 | 86M |
| ViT-Large | 24 | 1024 | 4096 | 16 | 307M |
| ViT-Huge | 32 | 1280 | 5120 | 16 | 632M |

### 4.3 结果分析

**表3 不同模型在不同数据集上的分类表现**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Ours-JFT  （ViT-H/14） | Ours-JFT  （ViT-L/16） | Ours-I21k  （ViT-L/16） | BiT-L  （ResNet152×4） | Noisy Student（EfficientNet-L2) |
| ImageNet | **88.55**±0.04 | 87.76±0.03 | 85.30±0.02 | 87.54±0.02 | 88.4/88.5∗ |
| ImageNet ReaL | **90.72**±0.05 | 90.54±0.03 | 88.62±0.05 | 90.54 | 90*.*55 |
| CIFAR-10 | **99.50**±0.06 | 99.42±0.03 | 99.15±0.03 | 99*.*37 ± 0*.*06 | − |
| CIFAR-100 | **94.55**±0.04 | 93.90±0.05 | 93.25±0.05 | 93.51±0.08 | − |
| Oxford-IIIT Pets | **97.56**±0.03 | 97.32±0.11 | 94.67±0.15 | 96*.*62 ± 0*.*23 | − |
| Oxford Flowers | 99.68±0.02 | **99.74**±0.00 | 99.61±0.02 | 99*.*63 ± 0*.*03 | − |
| VTAB | **77.63**±0.23 | 76.28±0.46 | 72.72±0.21 | 76*.*29 ± 1*.*70 | − |
| TPUv3-core-days | 2.5k | 0.68k | 0.23k | 9.9k | 12.3k |

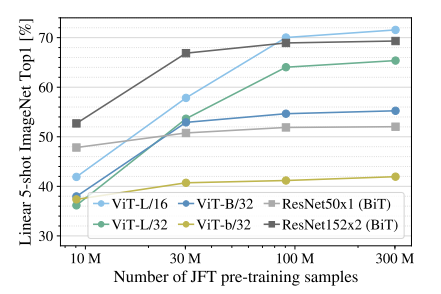
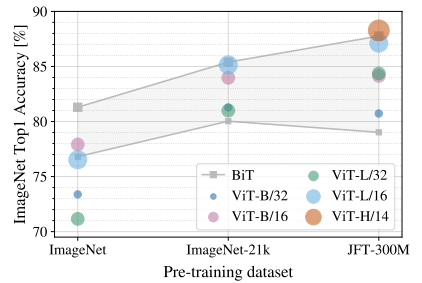
由上表可以看出：在JFT数据集上预训练的ViT模型，迁移到下游任务数据集后，表现都优于基于ResNet的模型，且只需要更少的训练资源。较大的模型ViT-H/14进一步提高了性能，特别是在更具挑战性的数据集，如ImageNet、CIFAR-100和VTAB数据集。



**图2 各类模型在VTAB上的表现，ViT同样性能更好**

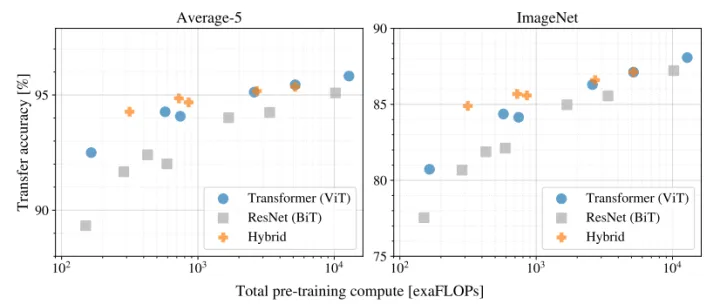
上面的实验结果表明：当在很大的数据集上预训练时，ViT性能超越CNN。

接下来探究不同预训练数据集的规模大小对模型性能的影响。



**图3 预训练中，不同数据集规模对模型性能的影响**

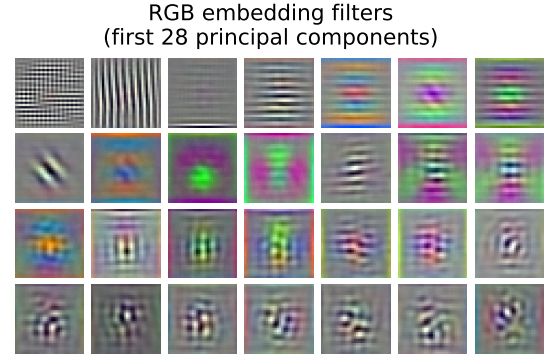
这里可以看到，在保持其他条件尽可能相同的情况下，当在小规模数据集上预训练时， ResNet预训练模型表现更好，这证明了CNN归纳偏置的有效性；当在更大的数据集上进行预训练时，ViT大放异彩，归纳偏置和Transformer比较就失去了优势。同时，随着数据集规模的增大，较大的ViT变体模型的性能会超过较小的变体模型。



**图4 不同架构的性能预计算训练**

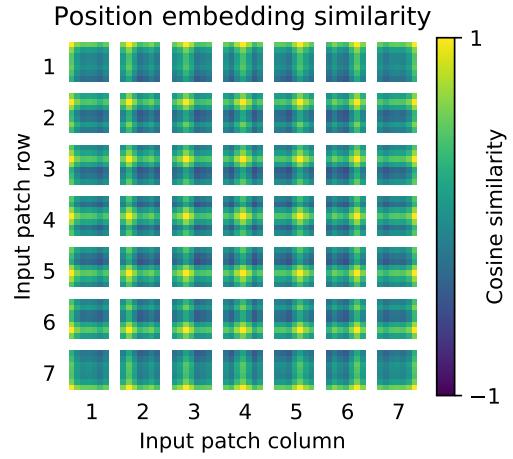
上图实验证明了ViT的预训练比ResNet所花费的训练资源更少，即在相同的预训练计算复杂度下，ViT的效果要比ResNet更好。可以看到，当模型较小时，混合模型的表现要更好，但是随着模型的增大，ViT的表现超过了混合模型。

### 4.4 模型可视化



**图5 ViT Block第一层（Linear Projection）的前28个主成分**

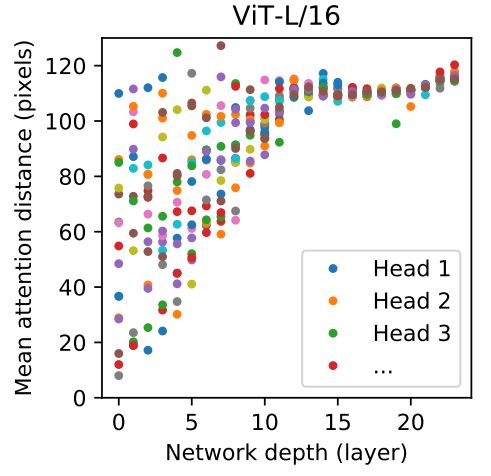
通过图像RGB初始值线性嵌入过滤器的结果，我们可以发现：这些成分与卷积神经网络学习到的特征十分相似，因此可以用于描述每一个图像块的这种底层结构。



**图6 相似性分析**

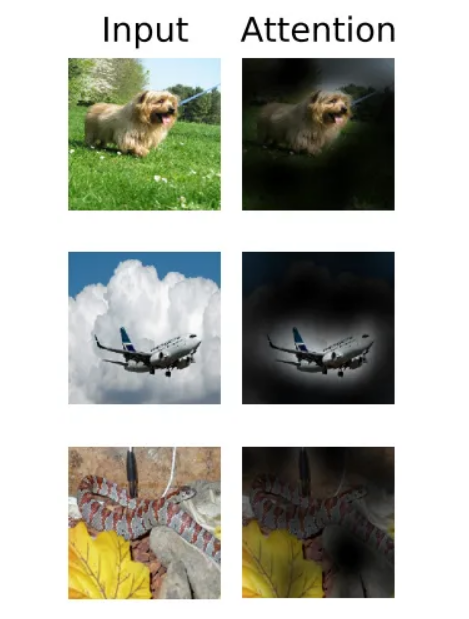
论文中也对ViT模型学习到的位置编码进行了可视化。我们发现：ViT模型学习到的位置编码信息确实能够表示不同patch之间的距离信息——相近patch的位置编码信息较相似，且同行或同列的位置编码信息也相近。

为了理解自注意力机制是如何聚合信息的，这里我们基于自注意力权重计算不同层Transform Block中不同Head的平均注意力距离。



**图7 Mean Attention Distance**

不同层中的每一个Head的平均注意力距离，其实类似于CNN中“感受野”的概念。结果表明：一些Head在第一层就能注意到几乎整张图片范围内的全局信息。随着层数的增加，每个Head的平均注意力距离都非常大，即都能关注到全局信息，而不是靠临近范围的像素点去进行判断。这表明自注意力机制确实使用了全局的信息。



**图8 自注意力机制捕捉到的图像关键信息**

为了验证这一点，论文中将网络最后一层输出的output token重构折射回为原始图像，并与真正的原始图像进行比对。我们其实可以看到：由于输出的token融合了全局的信息，因此自注意力机制能够捕捉到原图像在分类任务中真正起作用的区域。

### 4.5 结论

我们探索了Transformer在图像识别中的直接应用。与之前在计算机视觉中使用自注意力机制的工作不同，除了初始的patch提取步骤外，我们没有将特定于图像的归纳偏置引入架构中；相反，我们将图像解释为一系列的patch，并通过NLP中使用的标准Transformer编码器对其进行处理。

当拥有足够多的数据进行预训练的时候，ViT的表现会超过CNN，突破Transformer因缺少归纳偏置而导致泛化能力较差的限制，在下游任务中获得较好的迁移效果；同时预训练所需要的资源更少。

虽然这些初步结果令人鼓舞，但仍存在许多挑战。一是将ViT应用于其他计算机视觉任务中，如检测和分割；另一个挑战是继续探索ViT自我监督的预训练方法。