Transformers for Image Recognition at Scale
ViT - Vision Transformer

Transformers for Image Recognition at Scale（ViT）

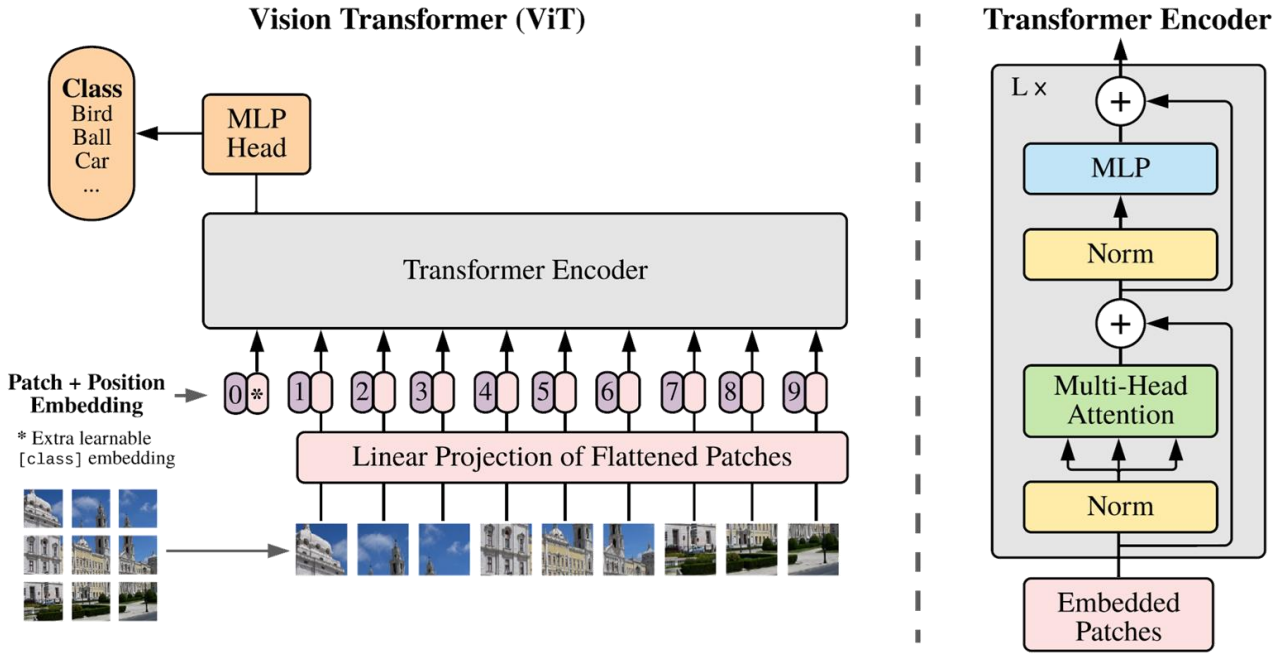
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai. An Image Is Worth 16X16 Words: Transformers for Image Recognition at Scale（ViT）
  - Google Brain
  - https://github.com/google-research/vision_transformer
- Motivation
  - 使用self-attention机制完全替代CNN进行目标检测任务。
- Contribution

Contribution

- 将NLP领域的Transformer尽可能不作修改搬到CV领域
- 用self-attention机制替代CNN，设计将二维图像转化为序列化数据的方式
- 没有使用faster-rcnn或者其他类似的backbone网络进行预处理
- 使用方便，可以开箱即用
- 大规模实验证明了Vision Transformer模型的能力优于SOTA CNN模型
- 利用不同尺度的数据集和模型，设计了benchmark，与ResNet和混合模型进行比较评价

模型架构

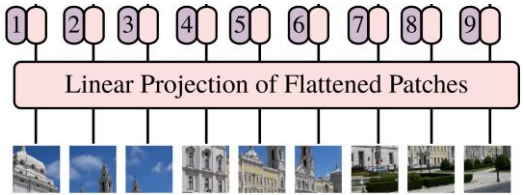
- 模型的架构分为
  - Patching Embedding
  - Position Embedding
  - Learnable Embedding
  - Transformer Encoder



将图像转化为序列化数据

- 将图像分割成patch，然后reshape成一个向量，得到flattened patch
  - 图片$H \times W \times C$，patch大小$P \times P$
    - N个patch($P \times P \times C$)，转化为P^2C维向量，将N个向量concat成$N \times (P^2C)$二维矩阵

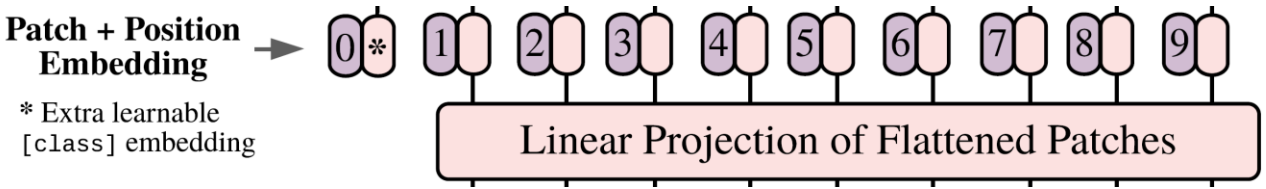
  - 为了避免P^2C维的向量长度改变，做Linear Projection
    - 将不同长度的flattened patch向量转化为固定长度为D的向量



  - 原本$H \times W \times C$维图片转化为N个D维向量（或$N \times D$维二维矩阵)

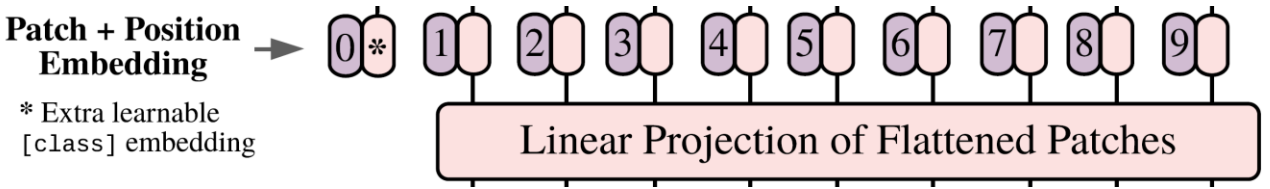
Position embedding

- 用position embedding将位置信息加到模型中去
  - 编号0-9的紫色框表示各个位置的position embedding
  - 粉色框为经过linear projection之后的flattened patch向量
  - 将position embedding和patch embedding相加



Learnable embedding

- 带星号框（0号框右边) 不是通过某个patch 产生
  - learnable embedding（记作 X_{class}），其作用类似于BERT中的[class] token
    - BERT中[class] token经过encoder后对应的结果作为整个句子的表示
  - X_{class}经过encoder后对应的结果，作为整个图的表示

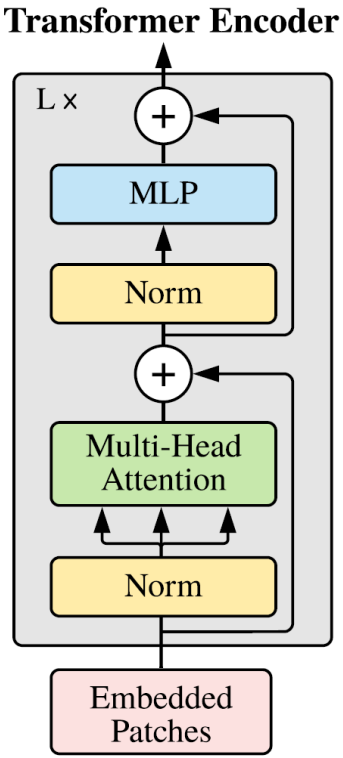


Transformer Encoder

- 对于Encoder的第 l 层，记其输入为 z_{l-1} ，输出为 z_l ，则计算过程

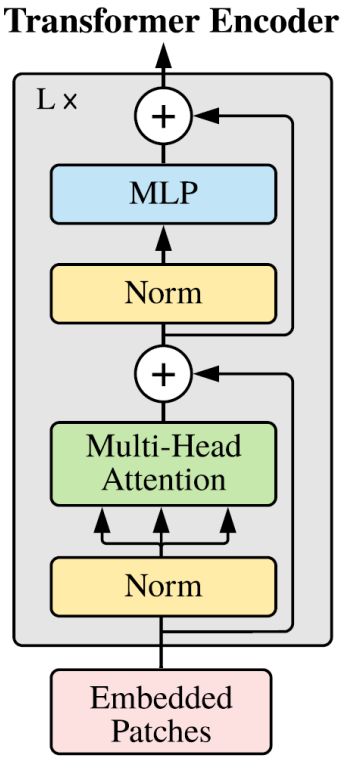
$$\begin{aligned}\mathbf{z}'_\ell &= \mathrm{MSA}\big(\mathrm{LN}(\mathbf{z}_{\ell-1})\big) + \mathbf{z}_{\ell-1}, && \ell = 1 \ldots L \\ \mathbf{z}_\ell &= \mathrm{MLP}\big(\mathrm{LN}(\mathbf{z}'_\ell)\big) + \mathbf{z}'_\ell && \ell = 1 \ldots L\end{aligned}$$

  - 其中，MSA为Multi-Head Self-Attention（绿色框）
  - MLP为Multi-Layer Perceptron（蓝色框）
  - LN为Layer Norm（黄色框)
- Encoder第一层的输入 Z_0
  - $\mathbf{z}_0 = \left[\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}\right] + \mathbf{E}_{\text{pos}}$ ，$\mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}$, $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$
  - 其中， $X_p^1, \cdots, X_p^N$ 即Linear Projection后的patch embedding (P^2C 维)，右乘 $P^2C \times D$ 维的矩阵E表示Linear Projection，得到的 $X_p^1E, \cdots, X_p^NE$ 为D维向量
  - N个D维向量和D维向量X_{class} concat得$(N+1) \times D$维矩阵。加上$N+1$个D维position embedding拼成$(N+1) \times D$维矩阵E_{pos}，即得encoder的原始输入z_0



混合结构

- 将transformer和CNN结合，即将ResNet的中间层的feature map作为Transformer的输入
  - 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



Fine-tuning过程中高分辨率图像的处理

- 在Fine-tuning到下游任务时，当图像的分辨率增大时，如果patch大小不变，得到的patch个数将增加（记分辨率增大后新的patch个数为 N' ）
  - pretraining时，position embedding个数和pretraining分割得到patch个数（N) 相同。则多出来$N' - N$个positioin embedding在pretraining中未定义或者无意义
- 提出2D插值方法，基于原图中的位置信息，将pretrain中的N个position embedding插值成N'个。得到N'个position embedding的同时也保证了其语义信息

实验

- 在中等规模的数据集上（例如ImageNet），transformer模型的表现不如ResNets
- 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

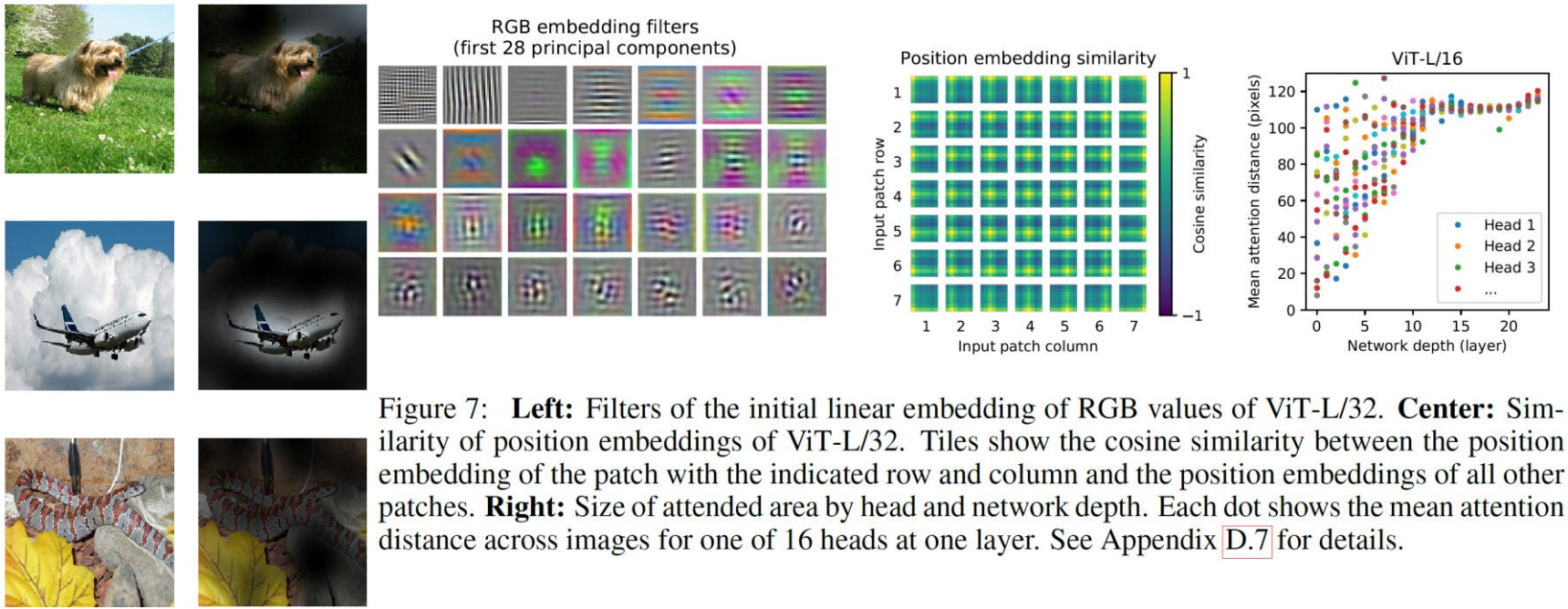


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.7 for details.