

Abstract

Transformer [attention is all you need]是一个纯基于注意力机制的神经网络架构。Transformer 最早被提出于解决 Sequence to sequence model [Sequence to Sequence Learning with Neural Networks]中的机器翻译任务，它强大的全局元素间的建模能力使它在“Machine Translation on WMT2014 English-German”的比赛中取得单模型 State-of-the-art[attention is all you need]。如图 1, Transformer 逐渐成为 NLP 和 CV 领域主流的深度神经网络架构。之后的工作展示了 Transformer 先在足够的数据集上（如 BERT 的预训练数据集 BooksCorpus (800M words) (Zhu et al., 2015) 和 English Wikipedia (2,500M words) 和 GPT3 的预训练数据集 Common Crawl 数据集）进行预训练，之后针对各种小数据集或各种特定的下游任务进行微调(fine-tune)，在 Machine Translation, Sentiment Analysis, Question Answering 等领域的不同数据集均能取得 SOTA 结果。

批注 [HB1]: BERT, mBERT, or BiBERT? A Study on Contextualized Embeddings for Neural Machine Translation

Lessons on Parameter Sharing across Layers in Transformers

Very Deep Transformers for Neural Machine Translation

DeLigT: Deep and Light-weight Transformer

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

批注 [HB2]: SMART: Robust and Efficient Fine-Tuning for Pre-trained Natural Language Models through Principled Regularized Optimization

ALBERT: A Lite BERT for Self-supervised Learning of Language Representations

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

client: Generalized Autoregressive Pretraining for Language Understanding

批注 [HB3]: Big Bird: Transformers for Longer Sequences
TANDA: Transfer and Adapt Pre-Trained Transformer Models for Answer Sentence Selection

批注 [HB4]: AlexNet

批注 [HB5]: Generative Pretraining From Pixels.

An
Image is Worth 16x16 Words: Transformers for Image
Recognition at Scale.

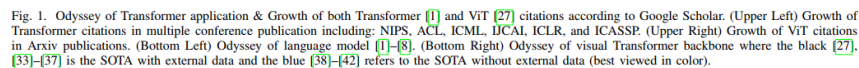
批注 [HB6]: Masked Autoencoders Are Scalable Vision Learners

End-to-End Object Detection with Transformers

批注 [HB7]: TransGAN: Two Transformers Can Make One Strong GAN.
Image Transformer.
“Pre-trained image processing transformer,”

批注 [HB8]: End-to-end panoptic segmentation with mask transformers,

批注 [HB9]: ViViT: A Video Vision Transformer.
Temporal Context Aggregation for Video Retrieval
With Contrastive Learning.



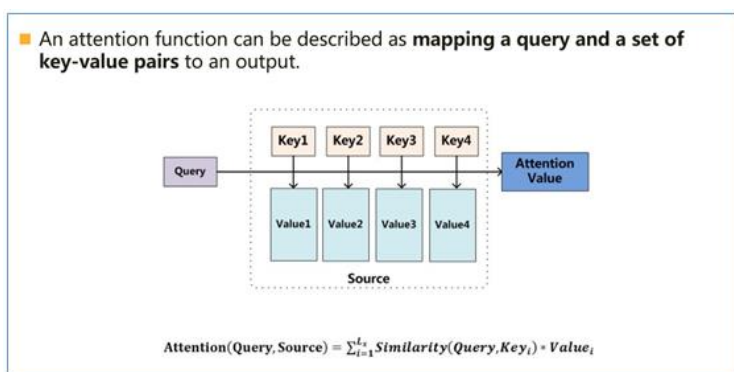
1

Transformer 是一种完全基于自注意力机制的模型架构，在最开始的时候 Transformer 是一个为了处理 Sequence-to-Sequence Model 中的机器翻译任务的新架构。Transformer 兼具元素间长程依赖计算容易和并行处理的特性，相比于以往的模型在处理 Seq2Seq 任务中，CNN 可以通过窗口滑动实现并行处理，由于 CNN 的局部性导致 CNN 难以计算远距离的元素之间的长程依赖，只能通过加深网络层数才能建模位置较远的元素；RNN 在预测下一个元素时能包含之前输入元素的信息，即预测的元素根据输入的所有元素的信息综合产生，一定程度上可解决长程依赖问题，但每次只能产生一个输出无法并行输出。受限于 RNN 和 CNN 的架构问题，Vaswani et al. 提出了 Transformer 架构。本模块使用七个小节来介绍 Transformer 架构：

A. 注意力机制

2014-2018 年，DeepMind 团队 Mnih, V. et al. 首先提出在 RNN 模型上使用 Attention 机制来进行图像分类。随后 Bahdanau, D. et al. 提出 Attention Mechanism，Minh-Thang Luong, Hieu Pham, Christopher D. Manning 在 [1] 中提出了两种 attention 的改进版本，即 global attention 和 local attention。Global Attention 相较于 [1] 能够更简单直接的在每一次生成目标词时，计算所有源语句隐藏状态的相似度。Local Attention 只需要在计算时对源语句的某个子集计算相似度，之后基于子集生成文章向量(context vector)。Local attention 可以视为 Hard Attention 和 Soft Attention 的混合体，因为他的计算复杂度要低于 Global attention 和 Soft attention，且 local attention 几乎处处可以微分，易于进行训练。等人提出的 [2] 使用了大量的自注意力机制，凭借强大的建模能力，自注意力机制也成为了研究者们研究热点，并使用自注意力机制在 NLP 任务上进行探索。

Attention 函数的本质可以被描述成为一个查询(Query)到一系列(键 Key-值 Value)对的映射，如图：



计算 Attention Score 可以分为三步：

$$\text{Step1: } f(Q, K_i) = \begin{cases} Q^T K_i & \text{dot} \\ Q^T W_a K_i & \text{general} \\ W_a [Q^T; K_i] & \text{concat} \\ v_a^T \tanh(W_a Q + U_a K_i) & \text{perceptron} \end{cases}$$

$$\text{Step2: } \alpha_i = \text{softmax}(f(Q, K_i)) = \frac{\exp(f(Q, K_i))}{\sum_j \exp(f(Q, K_j))}$$

$$\text{Step3: } \text{Attention}(Q, K, V) = \sum_i \alpha_i V_i$$

其中 Q 是 Query，K 是 Key，V 是 Value，通过 $f(Q, K_i)$ 函数计算注意分数，再经过 softmax 函数对注意力分数矩阵进行归一化处理，最后将 α_i 加权到 Value 上。

B. 自注意力机制

批注 [HB10]: Attention is all you need

批注 [HB11]: Convolutional Sequence to Sequence Learning

批注 [HB12]: Sequence to Sequence Learning with Neural Networks

批注 [HB13]: Recurrent Models of Visual Attention. arXiv:1406.6247.

批注 [HB14]: Neural Machine Translation by Jointly Learning to Align and Translate.

批注 [HB15]: Effective Approaches to Attention-based Neural Machine Translation. arXiv:1508.04025.

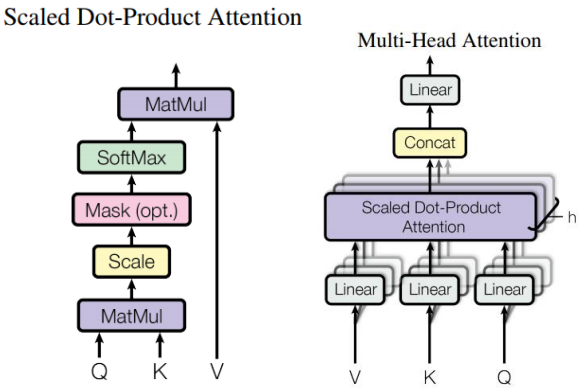
批注 [HB16]: Neural Machine Translation by Jointly Learning to Align and Translate.

批注 [HB17]: Attention is all you need

等人在 Transformer 中大量使用自注意力模块计算不同元素之间的相似度。作为 Transformer 中最重要的模块，该模块被分为部分，1)一个线性投影层将输入序列 $X \in R^{n^x \times d^x}$ $Y \in R^{n^y \times d^y}$ 投影到三个不同的特征向量中(query Q , key K , value V),其中 n 为序列长度， d 是一个词向量的编码维度。其中向量的产生由下式进行计算：

$$Q = XW^Q, K = YW^K, V = YW^V$$

其中 $W^Q \in R^{d_x \times d^k}$, $W^K \in R^{d_y \times d^k}$, $W^V \in R^{d_y \times d^v}$, $W^Q W^K W^V$ 是线性投影矩阵， d^k 是 query 和 key 向量的维度， d^v 是 value 向量的维度。Query 来自 X 的投影，键值对 (Key-Value) 来自 Y 的投影。这两个序列的输入方案被称为交叉注意力机制[]。一个注意力层如图：



左边单头自注意力机制；右边多头自注意力 From[Attention is all you need]

Transformer 使用 QKV(Query-Key-Value)来表示注意力机制，通过矩阵 $Q \in R^{N \times D_k}$, $K \in R^{M \times D_k}$, $V \in R^{M \times D_v}$, Transformer 自注意力的核心缩放点积注意力如下式：

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{D_k}}\right)V = AV$$

其中 N 和 M 分别是 Query 和 Key(或 Value)的长度， $D_k D_v$ 分别是 Query 和 Key(或 Value)的维度， $A = \left(\frac{QK^T}{\sqrt{D_k}}\right)$ 又被称作注意力权重矩阵，通过 Softmax 函数将注意力权重矩阵转化为标准分布，再分配给指定的 Key 元素，从而生成最终的输出向量，通过计算 Q 和 K 的点积后并除以 $\sqrt{D_k}$ 来平滑梯度，防止梯度消失。

C. 多头自注意力机制

因为单个注意力机制在受限的特征子空间进行建模，所以他建立的模型相对比较粗糙。为了使模型更加强大，V 等人在[]中提出了多头自注意力(multi-head self-attention mechanism)(MHSA)，以提升自注意力模块的表现能力。单头注意力机制限制了模型只能集中注意力在几个元素上以致于忽略其他重要的元素，多头注意力使得模型具有更强大的表达能力，能从不同的角度 (perspective)进行建模。通过不同的的 Head 关注不同的特征子空间，因为使用不同的 Query, Key, Value 矩阵，且矩阵随机初始化，这些矩阵可以将输入投影到不同的子空间当中。

给定输入向量和一定的头数 h ，输入向量首先被线性投影到三个矩阵中：Query, Key, Value。在每一个矩阵中使用头数 h (例，当 $h=8$ 时)划分向量， $d'_q = d'_k = d'_v = d_{model}/h = 64$ 。形成矩阵组

批注 [HB18]: Attention is all you need

Q, K, V。

$$\{Q_i\}_{i=1}^h, \{K_i\}_{i=1}^h, \{V_i\}_{i=1}^h.$$

多头自注意力的处理如下式：

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h) W^O \\ \text{where } \text{head}_i &= \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \end{aligned}$$

其中 $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, $W_i^Q \in \mathbb{R}^{d_{model} \times d'_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d'_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d'_v}$

D. 基于位置的前馈式网络

除了 Attention 子层，Transformer 中 Encoder 和 Decoder 每一层都含有一层基于位置的前馈式网络，其中包含一个 GELU 函数和一个全连接层，如下式：

$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2$$


MHSA 的输出直接作为输入进入基于位置的前馈式网络，FFN 在每一层 Encoder 或 Decoder 中的位置相同但是参数不同，所以能很好的线性拟合数据。

E. 位置编码

自注意力模块对输入的处理是不包含位置关系的，换言之一次输入对于自注意力模块同等重要。为了使自注意力模块获得捕捉到词位置信息的能力，Transformer 在输入加入了位置信息 (Positional Encoding)：

$$\begin{aligned} PE(pos, 2i) &= \sin\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \\ PE(pos, 2i+1) &= \cos\left(\frac{pos}{10000^{\frac{2i}{d_{model}}}}\right) \end{aligned}$$

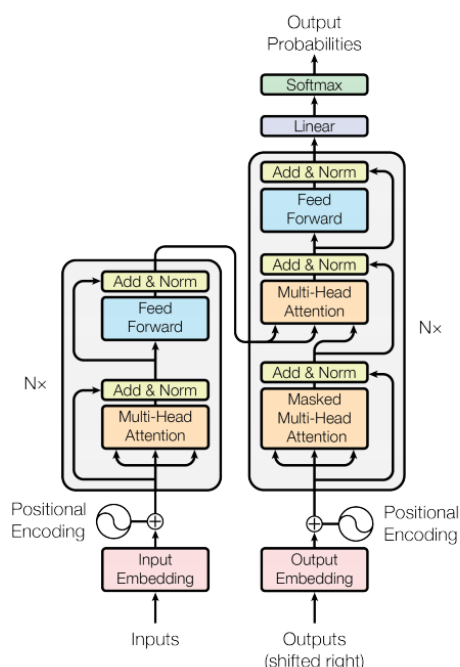
其中 pos 为元素的位置， i 是维度，位置编码的每一个维度均有一个正弦波与之对应。

之后的研究有专注于改进位置编码的模型，1)绝对位置表示，由等人提出的 BERT，由等人提出的 FLOATER，他们均为每一个位置学习一组 Position Encoding 来表示绝对位置；2)相对位置表示，由等人提出的 Music Transformer，由等人提出的 T5，由等人提出的 DeBERTa 等，通过编码元素之间的相对距离学习元素之间成对的关系。还有一种位置编码的方式为可学习式的嵌入，这类方式得到的位置编码更加灵活，如 ，但这类方式存在问题，若后续输入序列大于训练时最长的序列则无法使用学习的位置编码进行嵌入。

F. 模型概览

Fig 是 Transformer 的整体架构图。Transformer 整体的架构由 $N \times$ Encoder 连接 $N \times$ Decoder 组成。其中 Encoder 除第一层需要对输入加上位置编码外，所有 Encoder 都由一个自注意力模块和一个基于位置的前馈式网络组成，且两个模块的输出都会经过残差连接和层归一化进行处理；Decoder 相比于 Encoder 多了一个 Mask Multi-Head Attention，且 Decoder 中第二个自注意力模块的键值对(Key-Value pair)来自 Encoder，其余结构相同。

批注 [HB19]: Conditional Positional Encodings for Vision Transformers.
Learning to Encode Position for Transformer with Continuous Dynamical Model.



G. 模型分析

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(\log_k n)$
Self-attention(restricted)	$O(r \cdot n \cdot d^2)$	$O(1)$	$O(n/r)$

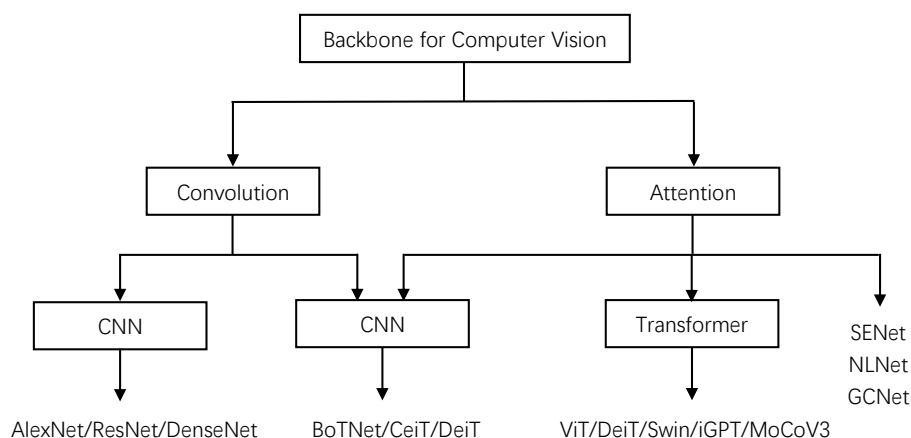
Table 1: 表中 n 是序列的长度, d 是维度, k 是卷积核大小, r 限制自注意力的范围大小。表中比较了不同类型网络的每一层的复杂度、最小处理序列次数(并行处理)、最大处理长度。来自 [1]

从表 1 中可以得出, 自注意力模块并行化的进行所有位置的连接, RNN 则需要 $O(n)$ 的顺序处理。当序列长度 n 小于维度 d 时, 自注意力模块的计算比 RNN 的计算更快。为提升处理超长序列的能力, va 等人设计了 restricted Self-attention 将注意力计算限制在一定的范围中, 提高 Transformer 对长序列的建模能力。

批注 [HB20]: Attention is all you need

3 Transformer' s Architecture in CV

3.1 Transformer Backbone for Computer vision



图表示计算机视觉中的两大模型分类：卷积神经网络和注意力机制。图来自[1]

在 Transformer 被用于机器翻译任务之前，计算机视觉领域使用主干的模型为卷积神经网络，如[AlexNet, ResNet, DenseNet]。Kaim 等人提出的 ResNet[2]对计算机视觉领域产生了深远的影响。当研究者们发现 Transformer 在 NLP 领域大放异彩的时候，研究者们思考能否对于计算机视觉领域能否直接使用 Transformer 模型进行处理。由于文本和图片的信息密度不同，直接将 Transformer 模型用于计算机视觉领域的想法没有成功。到 2020 年，等人提出的[3]重新掀起了 Transformer 在计算机视觉领域的应用。由于 ViT 根据图像进行独特的设计，规定一定大小的矩形块(如 ViT 里的 Token 大小 14*14)里的像素的集合称为 Token，将所有 Token 送入一个线性投影层，之后将线性拟合后的 Token 输入 Encoder 中进行建模。受到[4]的启发，在将 Tokens 输入到 Encoder 之前创建了一个类似 BERT 的[class] Token，用于储存所有元素之间交互的信息。经过 Encoder 后取[class] Token 通过 MLP 进行预测。ViT 的优异的表现迅速吸引了研究人员的目光，之后针对 ViT 的改进层出不穷[5]。

为了建立文本与图像之间的模型，Liu et al.提出了滑动窗口的分层 Vision Transformer[6]。该模型同 ViT 一样是 Transformer 在计算机视觉领域里的主干网络。

3.2 CNN 增强 Transformer

由于 Transformer 模型是对所有元素之间进行建模，当输入的数据巨大(如文本冗长，图片尺寸大)的情况，模型计算需要很多的计算资源，并且某些元素之间的建模不是必要的。所以在 Transformer 提出后的 NLP 领域，研究者们针对注意力计算提出了稀疏注意力[7]，线性化注意力[8]，低秩自注意力[9]等不同方向的改进。当 Transformer 引入计算机视觉后，针对大量的像素，直接进行像素级的自注意力计算成本巨大，所以研究者们借助 CNN 对图像进行特征提取之后输入 Transformer 进一步进行元素间的建模，这一思路减少了大量的冗余无效的计算，增加模型对图像建模的效率。这些工作总的可以概括为：软近似[10]、直接局部处理[11]、针对位置编码的表示[12]、组合结构[13]。

批注 [HB21]: Kai Han et :A Survey on Vision Transformer

批注 [HB22]: ViT

批注 [HB23]: BERT

批注 [HB24]: DeiT, PVT, TNT

批注 [HB25]: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

批注 [HB26]: Star-Transformer[43], Longformer[10], ETC[1], BigBird[163], Sparse Transformer[17] BP-Transformer[158], Image Transformer[94], Axial Transformer[54] 2106

批注 [HB27]: Linear Transformer[62], Performer[18, 19], RFA[95], Delta Net[113] 2106

批注 [HB28]: Low-rank Attention[45], CSALR[16], Nyströmformer [152] 2106

批注 [HB29]: “Training data-efficient image transformers & distillation through attention, Convit: Improving vision transformers with soft convolutional inductive biases, 2106

批注 [HB30]: “Incorporating convolution designs into visual transformers “Localvit: Bringing locality to vision transformers

批注 [HB31]: Conditional positional encodings for vision transformers Rest: An efficient transformer for visual recognition

批注 [HB32]: Coatnet: Marrying convolution and attention for all data sizes Early convolutions help transformers see better

相关的工作:

ConViT: 由于卷积架构的硬性归纳偏置能够有效的对图像进行学习,但是他的性能几乎已经达到上限。Vision Transformer 依赖于大量数据的训练,在计算机视觉领域某些下游任务的表现超过了以 CNN 为基准的模型,但前者性能仍有提升的空间。等人^[1]提出了结合卷积架构和 Vision Transformer 的优势的模型 ConViT。ConViT 提出了一种门控位置的自注意力模型(GPSA),该模型通过“软”卷积的归纳偏置,模仿卷积核来初始化 GPSA,使每一个注意力头通过调节位置与信息的注意力门控参数来避开位置性,形成一种类似卷积的 ViT 架构。在 ImageNet 数据集中 ConViT 的表现优于 DeiT。

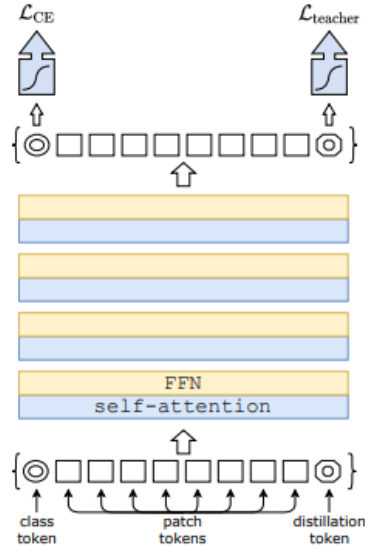
GPSA 的表达式如下:

$$A_{ij}^k := \left(1 - \sigma(\lambda_k)\right) \text{softmax}(Q_i^h K_j^{hT}) + \sigma(\lambda_k) \text{softmax}(v_{pos}^{hT} r_{ij})$$

其中 $v_{pos}^h \in R^{D_{pos}}$ 类似卷积核一样可学的嵌入, $r_{ij} \in R^{D_{pos}}$ 是一个固定的相对位置嵌入, λ_k 是可学习的门控参数。

DeiT: ViT 这个纯基于注意力的网络架构被证实能适应计算机视觉的下游任务并取得好的表现。但训练 ViT 所需要的训练数据和时间都是很大的,从而限制了将模型的应用。等人提出了一个依赖于 Token 的蒸馏并完全针对 Transformer 设计的师生策略,确保学生网络根据注意力向教师网络学习。DeiT 使用 ImageNet 进行训练无需引入额外数据能在参数量为(86M)的情况下取得 Top-1 准确率为 83.1% 的表现。

批注 [HB33]: ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases



Soft distillation: $L_{global} = (1 - \lambda)L_{CE}(\psi(Z_s), y) + \lambda\tau^2 KL(\psi(\frac{Z_s}{\tau}), \psi(\frac{Z_t}{\tau}))$

其中 Z_t 为教师模型的输出, Z_s 为学生模型的输出, τ 为蒸馏温度, KL 为 KL 散度, L_{CE} 为 CE-loss, y 为真值, ψ 为 softmax, λ 为权重。

Hard distillation: $L_{global}^{hardDistill} = \frac{1}{2}L_{CE}(\psi(Z_s), y) + \frac{1}{2}L_{CE}(\psi(Z_s), y_t),$

其中 $y_t = \operatorname{argmax}_c Z_t(c)$

和 ViT 的 class token 类似, distillation token 是在图像块序列后面加上的 token。然后它经过 Transformer 的 Encoder 之后的输出用于和 teacher model 的输出计算损失。作者发现 class token 和 distillation token 会收敛于不同的向量, cos 距离为 0.06, 随着层数的加深, 两者 embedding 的相似度逐渐上升, 最终 cos 距离为 0.93, 说明他们希望得到相似但不相同的 target。为了证明 distillation token 的有效是由于 knowledge distillation 的作用, 而不是只由于多了一个 token 导致的, 作者做了对比试验, 不是在后面加 distillation token, 而是再加一个 class token, 发现他们最终收敛结果的相似性达到 0.999, 并且性能弱于前者, 证明了加入 distillation token 的意义。

3.3 Transformer 增强 CNN

由于卷积架构对计算机视觉任务的归纳偏置, 使得 CNN 对于处理计算机领域任务具有天然优势。但由于 CNN 感知图像是通过一定数量的卷积核扫过整张图片, 受限于卷积核的大小, CNN 的视野具有局限性, 只能查看到当前元素一定范围内的元素, 所以对于目标检测这一类需要计算长程依赖的任务进行特殊化的设定。由于 Transformer 可以远距离按建模两个元素之间的关系, 使得模型对图像进行处理的时候模型的视野不局限于相邻元素, 所以为了解决 CNN 视野局限的问题, 研究人员使用借助 Transformer 远距离建模两个元素之间的特性来加强 CNN。

BoTNet: BoTNet 的工作是将 ResNet 的最后三个 Stage 中, 使用全局的自注意力来代替空间卷积操作。为了解决计算机视觉任务中图像数据量过大的问题(如处理的一张图片的像素大小为 1024×1024)和自注意力模块的记忆和计算的复杂度随空间维度的增加呈现四次方的增长, 使训练和推理的开销巨大。BoTNet 针对上述问题做出如下的改进: (1) 通过卷积从大型图像中学习抽象和低分辨率的特征图谱。(2) 使用全局自注意力来处理并聚合卷积提取的特征图谱。此工作在不少计算机视觉领域的下游任务取得良好的成绩, 比如取代了 ResNet 模型在 COCO 实例分割任务的最佳单一的模型。之后通过微调, 在 ImageNet 数据集上达到了 84.7% 的 Top-1 准确率, 并且在 TPU-v3 的硬件训练效率是流行效率计算模型的 1.64 倍。

VTs: 对于计算机视觉的处理主要是(1)将图像统一转化为排列的像素矩阵, (2)对已局部提取出的高纬度特征进行卷积。可是由于卷积操作不区分各个元素的重要性, 且无论语义信息直接对各元素进行建模, 导致元素间建立模型的视野不具有全局性。在 [1] 中, 等人(1)通过建模 Tokens 来表示图像中各部分的语义关系, (2)引入 Transformer 模型来计算不同 Token 之间的语义关系。结果现实, VTs 能根据上下文有意识地关注不同的图像部分。这与直接将像素输入 Transformer 有了鲜明的对比, 因为直接使用像素需要更大数量级的计算资源。通过比较 VTs 与 CNN, 在 ImageNet 中使用更少的 FLOPs 和更少的参数将 ResNet 的准确性提高了 4.6 到 7 个百分点; 基于 VTs 的特征金字塔网络能在减少 FPN 模块中 FLOPs 6.5 倍的情况下实现了 0.35 个百分点的 mIoU。

批注 [HB34]: Visual transformers: Token-based image representation and processing for computer vision Bottleneck transformers for visual recognition.

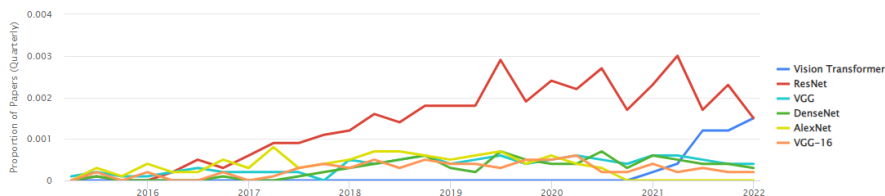
批注 [HB35]: Efficient transformers: A survey.

批注 [HB36]: ResNet

批注 [HB37]: Bottleneck Transformers for Visual Recognition

批注 [HB38]: VTs

4 Transformer' s Application In Computer Vision



△ This feature is experimental; we are continuously improving our matching algorithm.

图表 “from paper with code”

Transformer^[1]在2017年提出，在那之后广泛的用于NLP领域，研究人员们提出许多NLP领域的重要论文^[2]，推动了NLP领域的发展。研究者在早期也尝试将Transformer应用到计算机视觉领域^[3]，但表现都不是太好。从2020年ViT的提出，Transformer在CV领域中展现了革命性的提升，研究者们有可能可以将CV和NLP通过Transformer架构统一起来。这一结果有助于：(1)易于视觉和语言的联合建模^[4]。(2)两个领域的研究经验可以相互借鉴，从而加快各自领域的发展。

批注 [HB39]: Attention is all you need

批注 [HB40]: BERTs, RoBERTa, GPT, GPT-2, GPT-3, T5, BART, XLNet

批注 [HB41]: Image Transformer

批注 [HB42]: Swin

4.1 High/mid Level Vision

自从ViT提出后研究者们致力于使用或改进ViT研究Transformer在计算机视觉下游任务的应用。如图像分类^[5]，目标检测^[6]，分割任务^[7]。

批注 [HB43]: ViT Swin DeiT

批注 [HB44]: End-to-end object detection with transformers.

Toward transformer-based object detection.

Deformable detr: Deformable transformers for end-to-end object detection

Temporal-channel transformer for 3d lidar-based video object detection in autonomous driving.

3d object detection with pointformer

批注 [HB45]: End-to-end video instance segmentation with transformers

End-to-end panoptic segmentation with mask transformers

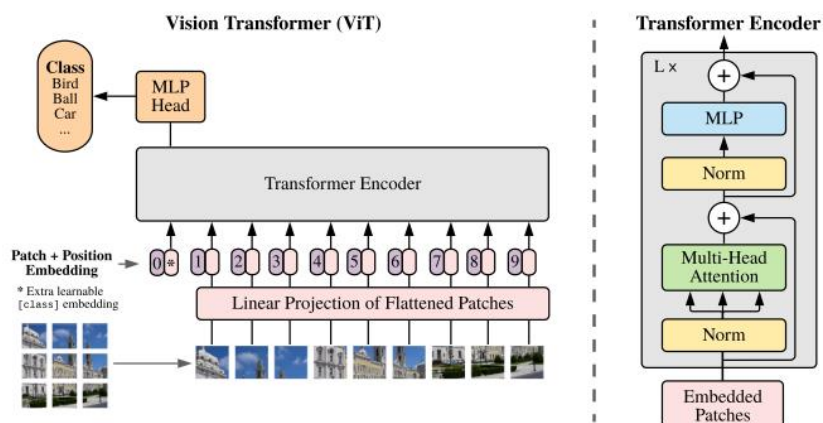
Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers

批注 [HB46]: ViT

4.1.1 Transformer for Classification

ViT: ViT由Dosovitskiy et al.提出的第一个基于Transformer的计算机视觉领域的主干结构^[8]。原始的Transformer的输入需要是序列化的Token，但图片是由一个一个的像素组成稍大一点的图片(如1024*1024)像素有1048576个，这对于Transformer模型的计算要求是很大的。ViT提出将图片分为一定数量固定大小的Patch，通过对Patch建模，进而处理图片。

ViT首先划分一定数量固定大小的Patch将所有Patch输入一个线性投影层，再随机初始化一个同等大小的[class] Token用于储存Token间的信息，将线性拟合后的Token加上1维的位置编码输入Encoder，经过Encoder对所有元素进行建模后取[class] Token用于预测。ViT的整体架构如下图：

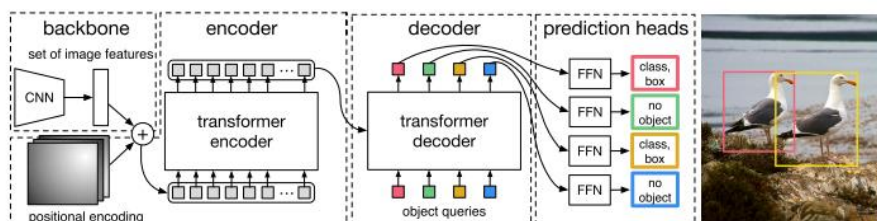


ViT 在超大的数据集进行预训练过后针对各个计算机视觉领域的下游任务进行微调，能消耗更少的训练资源取得超过 CNN 模型性能的 SOTA 结果。同时 ViT 也启发了计算机视觉自监督预训练领域。

4.1.2 Transformer for Detection

在 Transformer 被用于目标检测任务之前，主流目标检测模型有[1]。但基于 Transformer 的目标检测模型有不凡的表现[2]。

End-to-End Object Detection(DETR): DETR 是最先提出的基于 Transformer 的端到端的直接进行集合预测的目标检测模型。



图表 1 DETR 的架构图，来自[2]

DETR 将 CNN 和 Transformer 相结合，取代了以往的模型需要手工设计的工作，并且具有不错的性能。DETR 采用的 CNN 为 Backbone 提取图片特征，Transformer 对提取的特征进行进一步的建模，通过 Decoder 直接一次性产生图中所有类别的预测。

DETR 的三大特点：(1)End to End 任务模式(原始图片输入无需处理直接做出预测无需人工设计);(2) 设计了双边匹配损失(bipartite matching loss)，基于预测的 box 和 ground truth boxes 的二分图匹配计算 loss 的大小，从而使得预测的 box 的位置和类别更接近于 ground truth。(3)集合预测，使用 Transformer 的 encoder-decoder 架构一次性生成 N 个 box prediction。其中 N 是一个事先设定的、远大于 image 中 object 个数的一个整数。

DETR 通过 Decoder 之后的分类分支(class)和回归分支(bounding box)进行预测，其中 DETR

批注 [HB47]: R-CNN

Fast R-CNN

Faster R-CNN

SSD

YOLO

Mask R-CNN

CentorNet

批注 [HB48]: DETR

批注 [HB49]: End-to-End Object Detection with Transformers

的输出张量的维度为(N,class+1)和(N,4),其中“class+1”表示已标注的类别数目和一个其他类别,4代表 Bounding box 的中心点坐标,高和宽。在训练中双边匹配算法 L_{match} 表示预测类别 $\hat{y}_{\sigma(i)}$ 与真实类别 y_i 一一对应的损失值:

$$\hat{\sigma} = \underset{\sigma \in \sigma_N}{\operatorname{argmin}} \sum_i^N L_{match}(y_i, \hat{y}_{\sigma(i)})$$

$$L_{match}(y_i, \hat{y}_{\sigma(i)}) = -1_{\{C_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(C_i) + 1_{\{C_i \neq \emptyset\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$$

类似于学习锚点匹配的标签分类方法[1]。使用匈牙利算法计算所有 Bounding box 和所有类别预测的两两广义距离,距离最近表示越可能是最优匹配关系:

$$L_{Hungarian}(y, \hat{y}) = \sum_{i=1}^N [-\log \hat{p}_{\hat{\sigma}(i)}(C_i) + 1_{\{C_i \neq \emptyset\}} L_{box}(b_i, \hat{b}_{\hat{\sigma}(i)})]$$

综上所述,DETR 提出了一种新的端到端目标检测的方式,在 COCO 数据集上取得与优化后的 Faster R-CNN 基线相当的结果,同时 DETR 这种灵活的结构全景分割的任务中也取得具有竞争力的结果。DETR 的提出也带来了新的挑战,尤其是在训练和优化检测小型目标的方面。

Deformable DETR: Zizhou Zhu et al 在[1]的基础上提出了 Deformable DETR[1]。Deformable DETR 为了解决 DETR 中:(1)训练收敛慢;(2)对小样本的检测效果差的问题。

批注 [HB50]: CentorNet

批注 [HB51]: “Deformable convolutional networks,

批注 [HB52]: DEFORMABLE DETR: DEFORMABLE TRANSFORMERS FOR END-TO-END OBJECT DETECTION

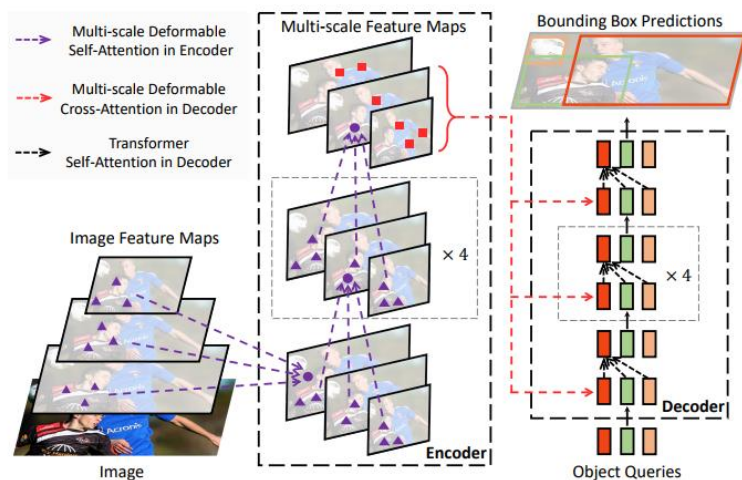


Figure 1: Illustration of the proposed Deformable DETR object detector.

Deformable DETR 相比于 DETR 在训练时少了 10X 的 epochs 但仍能取得较高的检测精度。收到特征网络金字塔的启发,将 Transformer 中的多头注意力改进为多尺度可变形的注意力机制 (MSDA):

$$A_{qik}^l = z_q W_{ilk}^A, V_{ik}^l = X^l (\phi_l(\hat{p}_q) + \Delta P_{ilqk}) W_i^V$$

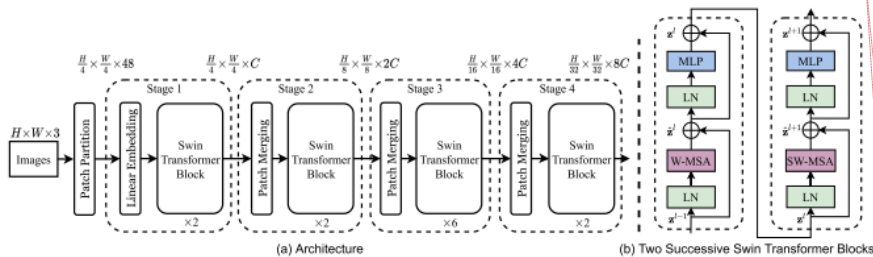
$$MSDAttn(A_{qik}^l, V_{ik}^l) = \sum_{i=1}^h (\sum_{l=1}^L \sum_{k=1}^{N_k} A_{qik}^l V_{ik}^l) W_i^O$$

MSDA 模块将计算复杂度降低到了 $O(2N_q C^2 + \min(HWC^2, N_q KC^2))$, 将模型的推理速度提高了 1.6 倍。

4.1.3 Transformer for Segmentation

分割任务是计算机视觉领域内重要的任务之一，包含语义分割^[1]，实例分割^[2]，全景分割^[3]。

Swin Transformer: Liu et al. 提出了一种使用滑动窗口沿着空间维度来建模全局和边界特征的模式^[4]。



Swin Transformer 的主要思想是将具有很强建模能力的 Transformer 结构和视觉信号的先验联合起来。通过浅层的小 Patch 到深层逐渐合并周围的 Patch，同一 Patch 内的像素相互进行建模，不同 Patch 之间不计算相似度。类似于 CNN 的卷积核滑过图片的形式通过滑动窗口建立相邻 Patch 之间的关系，随着图像分辨率的提高计算复杂性从 $O(2n^2C)$ 减少到了 $O(4M^2nC)$ 其中 n 是 Patch 的长度 M 是滑动窗口的大小，通过滑动窗口的方式能减少计算资源的消耗。

结果上，在目标检测的重要评测数据集 COCO 上，Swin Transformer 取得了单模型 58.7 的 box mAP 和 51.1 的 mask mAP，分别比此前最好的、没有扩充数据的单模型方法高出了 +2.7 个点和 +2.6 个点。此后，通过改进检测框架以及更好地利用数据，基于 Swin Transformer 网络的方法性能进一步取得了 61.3 的 box mAP 和 53.0 的 mask mAP，累计提升达 +5.3 box mAP 和 +5.5 mask mAP。在语义分割的重要评测数据集 ADE20K 上，Swin Transformer 也取得了显著的性能提升，达到了 53.5 mIoU，比此前最好的方法高出 +3.2 mIoU，此后随着分割框架和训练方法的进一步改进，目前已达到 57.0 mIoU 的性能。

4.1.4 Transformer for Self-supervise

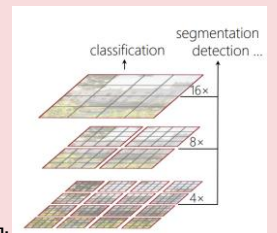
MoCo V3:

MAE: He et al. 提出了一种用于自监督任务的自编码的可扩展的视觉学习器^[5]。MAE 是一个基于 ViT 的 BERT 化的一个模型，它把整个训练拓展到没有标号的数据集上，通过随机掩去图片上的 Token 再利用未被掩去的 Token 进行图像还原。MAE 加速了 Transformer 在计算机视觉上的应用可能是未来影响最大的模型。

批注 [HB53]: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
Emerging Properties in Self-Supervised Vision Transformers
TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation

批注 [HB54]: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows
XCiT: Cross-Covariance Image Transformers
Bottleneck Transformers for Visual Recognition
Swin Transformer V2: Scaling Up Capacity and Resolution

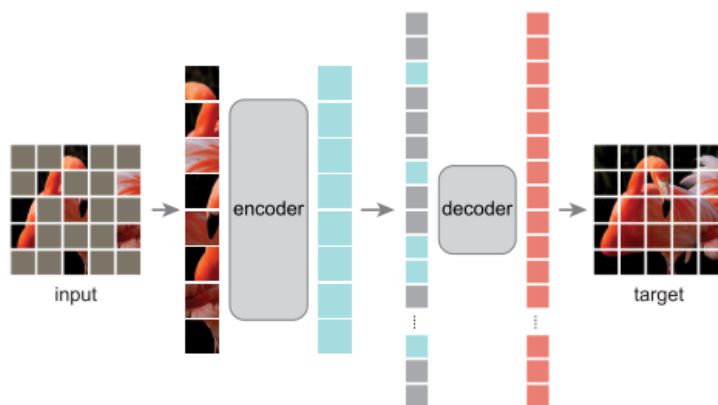
批注 [HB55]: End-to-End Object Detection with Transformers
PVTv2: Improved Baselines with Pyramid Vision Transformer



批注 [HB56]:

批注 [HB57]: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

批注 [HB58]: Masked Autoencoders Are Scalable Vision Learners



图表 MAE

MAE 的思路很简单，通过 Mask 遮住一定数量的块之后重构缺失的像素，MAE 的编码器只关注可见的块。在 MAE 中作者遮住了 75% 的块，若遮的块的数量少用传统的插值法即可重构缺失的像素值，遮住大部分块的话迫使模型能够学会更好地表征。

MAE 的模型架构首先输入 Patch，再对 Patch 做掩码操作，将没有掩码操作的 Patch 输入 Encoder，再将掩码的 Patch 和 Encoder 的输出按位置进行排列输入进 Decoder，通过 Decoder 重构 Masked Patch 的像素。MAE 通过小数据集 ImageNet 1K 进行自监督训练，再做迁移学习，它在其他计算机视觉下游任务上表现好。

4.2 Low level vision

4.2.1 Image Generation

4.2.2 Image Enhencenment

4.3 Video Processing

4.3.1 Video Classification

视频分类任务是产生于视频相关的标签的任务，好的视频分类模型不仅能够精准的提供帧标签，还能在给定视频的各帧的特征和注释的情况下很好的描述视频。典型的任务分为(1)为视频分配一个或多个全局标签；(2)为每一帧分配一个或多个标签。

ViViT: Arnab et al.提出了一个纯基于 Transformer 的视频分类模型。该模型借助 Transformer 在图像分类上成功的经验，考虑时间和空间维度的标记，在 Transformer 中进行编码。虽然 Transformer 需要大量数据集进行训练，但该工作展示了如何在训练过程中有效得规范化模型，并利用预训练模型的图像在小数据集上进行训练。

Table 4: COCO object detection and segmentation using a ViT Mask R-CNN baseline. All entries are based on our implementation. Self-supervised entries use IN1K data without labels. Mask AP follows a similar trend as box AP.

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H448
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	<u>83.6</u>	<u>85.9</u>	<u>86.9</u>	<u>87.8</u>

Table 5: ADE20K semantic segmentation (info). Not. BEiT results are reproduced using the official entries are based on our implementation. Self-sup use IN1K data without labels.

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H448
scratch, our impl.	-	47.9	48.3	47.7	47.7
DINO [5]	IN1K	47.9	48.3	47.7	47.7
MoCo v3 [9]	IN1K	47.9	48.3	47.7	47.7
BEiT [2]	IN1K+DALLE	47.9	48.3	47.7	47.7
MAE	IN1K	<u>48.3</u>	<u>48.9</u>	<u>48.9</u>	<u>48.9</u>

Table 3. Comparisons with previous results on ImageNet-1K. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [43]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

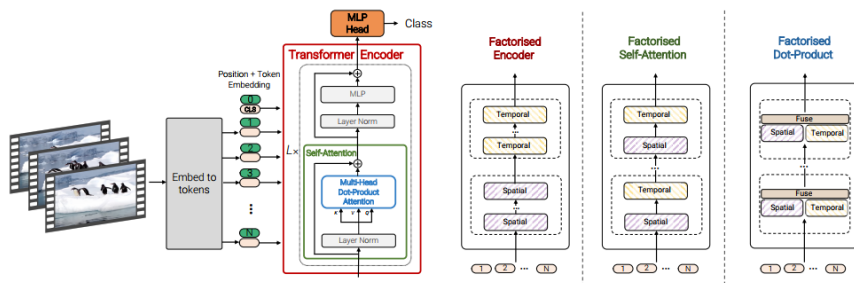
批注 [HB59]:

method	pre-train data	ViT-B	ViT-L	ViT-H	ViT-H448
scratch, our impl.	-	82.3	82.6	83.1	-
DINO [5]	IN1K	82.8	-	-	-
MoCo v3 [9]	IN1K	83.2	84.1	-	-
BEiT [2]	IN1K+DALLE	83.2	85.2	-	-
MAE	IN1K	<u>83.6</u>	<u>85.9</u>	<u>86.9</u>	<u>87.8</u>

Table 3. Comparisons with previous results on ImageNet-1K. The pre-training data is the ImageNet-1K training set (except the tokenizer in BEiT was pre-trained on 250M DALLE data [43]). All self-supervised methods are evaluated by end-to-end fine-tuning. The ViT models are B/16, L/16, H/14 [16]. The best for each column is underlined. All results are on an image size of 224, except for ViT-H with an extra result on 448. Here our MAE reconstructs normalized pixels and is pre-trained for 1600 epochs.

批注 [HB60]: Non-local Neural Networks
Group Normalization
Learning Representations from EEG with Deep
Recurrent-Convolutional Neural Networks

批注 [HB61]: ViViT

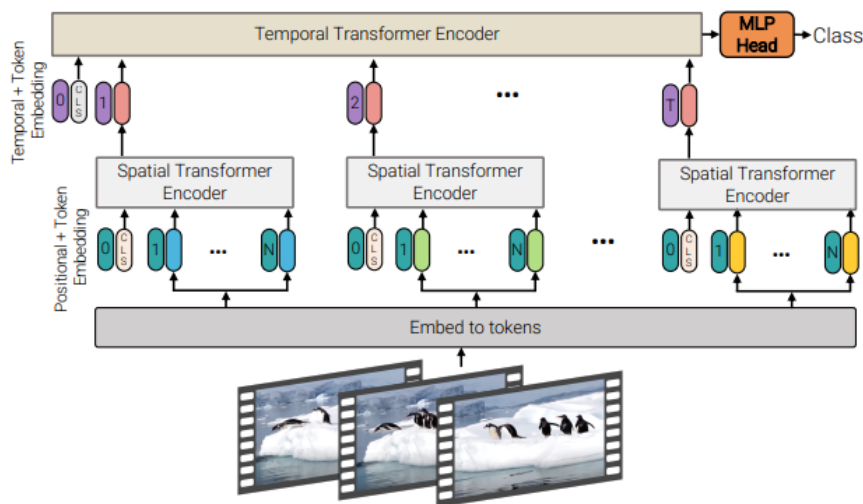


图表 ViViT overview

ViViT 提出了四个注意力模型:

(1) **时间空间注意力(Spatio-temporal attention):**与 CNN 架构的感受野随层数的加深感受野呈线性增长不同, 每层 Transformer 都对所有的时空标记进行建模, 因为他对所有对的关系进行建模, 多头自注意力相对于 Token 的数量是具有二次复杂性的。

(2) **因子化编码器(Factorised encoder):**



如上图, 该模型由两个独立的 Encoder 组成, 第一个为空间 Encoder, 目的是为了从同一时间索引中提取 Token 之间的相互作用。经过 L_s 的 Transformer 后能获得每一帧的表示。将所有帧的表示集合在一起由 L_t 个 Transformer 层组成的时间编码器, 来处理不同 Token 之间的相互作用。最后对该编码器输出的 Token 进行分类。

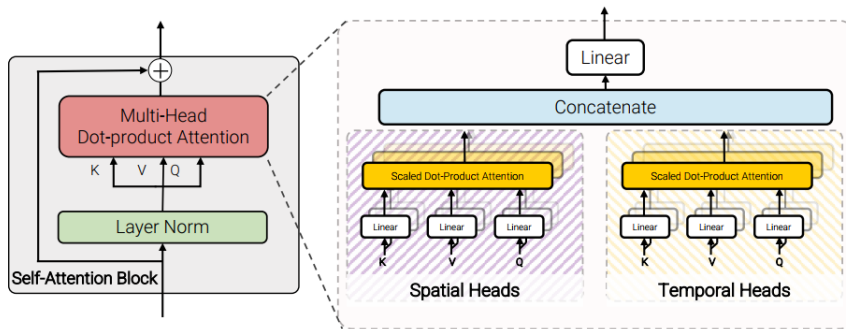
(3) **因子化自注意力(Factorised self-attention):** 与模型 1 架构相同, 但是先计算空间自注意力, 在计算时间自注意力, 因此架构中的每个自注意力块都有时空交互。为计算空间自注意力, 首先将 Token 的大小从 $\mathbb{R}^{1 \times n_t \cdot n_h \cdot n_w \cdot d}$ 重置为 $\mathbb{R}^{n_t \cdot n_h \cdot n_w \cdot d_t}$ (通过 Z_s 表示), 同理, 将时间的自注意力输入 Z_t 重置为 $\mathbb{R}^{n_t \cdot n_h \cdot n_w \cdot d}$ 。如下式:

$$y_s^l = MSA(LN(Z_s^l)) + Z_s^l$$

$$y_t^l = MSA(LN(y_s^l)) + y_s^l$$

$$z^{l+1} = MSA(LN(y_t^l)) + y_t^l$$

(4) 因子化点积注意力(Factorised dot-product attention):



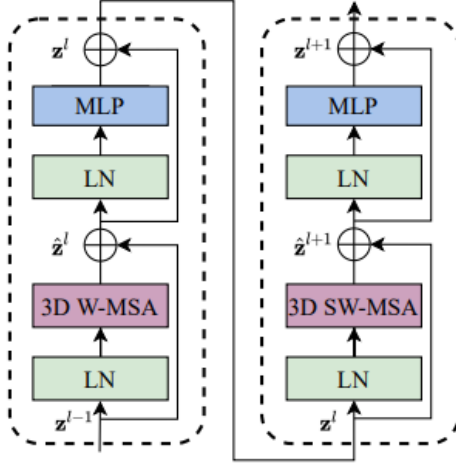
Model4 如上图所示，计算复杂度与模型 2 和模型 3 相同的模型，使用不同 head 分别在时间和空间维度上计算每个 Token 的注意力权重。

$$Attention(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

该模型分别构建了 $K_s, V_s \in \mathbb{R}^{n_h \cdot n_w \cdot d}$, $K_t, V_t \in \mathbb{R}^{n_t \cdot d}$, 该模型同时通过一半的注意力头计算 $Y_s = Attention(Q, K_s, V_s)$ 另一半的注意力头计算 $Y_t = Attention(Q, K_t, V_t)$ 分别关注时间维度和空间维度的 Token，最终通过线投影 $Y = Concat(Y_s, Y_t)W_o$ 来获得最终的输出。

ViViT 在多个视频分类基准上取得了最先进的结果，包括 Kinetics 400 和 600、Epic Kitchens、Something-Something v2 和 Moments in Time。在时间上取得了最先进的结果，超过了先前基于深度三维卷积网络的方法。

Video Swin Transformer: Cao et al.提出了在视频 Transformer 中引入局部性的归纳偏置[]，Video Swin Transformer 的架构能够分解时间和空间的信息并进行全局的注意力计算，具体架构的实现是调整为图像设计的 Swin Transformer[Swin Transformer]来实现的，同时也借助了预训练模型的力量。



Video Swin Transformer 同 Swin Transformer 一样采用滑窗的方法，两个连续的 Video Swin Transformer 计算如下式：

$$\hat{z}^l = 3DW - MSA(LN(z^{l-1})) + z^{l-1},$$

$$z^l = FFN(LN(\hat{z}^l)) + \hat{z}^l,$$

$$\hat{z}^{l+1} = 3DSW - MSA(LN(z^l)) + z^l,$$

$$z^{l+1} = FFN(LN(\hat{z}^{l+1})) + \hat{z}^{l+1},$$

其中 \hat{z}^l 和 z^l 分别表示 3D(S)W 和 FFN 模块输出的特征，3DW-MSA 和 3DSW-MSA 分别表示基于三维窗口的多头自注意力使用正则和移位的窗口分区配置。

Video Swin Transformer 在视频识别的基线上实现了最先进的精准度。如在动作识别的验评集 Kinetics-400 取得了 84.9% 的 top-1 准确率，在验评集 Kinetics-600 上以小 20 倍的预训练数据和小 3 倍的模型大小取得了 69.6% 的 Top-1 准确率。

4.3.2 Object Tracking

物体追踪的任务是获取一组初始的物体检测，为每个初始检测创建一个唯一的 ID，然后追踪每个物体在视频帧中的移动，保持 ID 分配。

TransTrack:简单在线和实时跟踪（SORT）是一种实用的多物体跟踪方法，重点是简单、有效的算法。在 TransTrack 中，Wojke et al.整合了外观信息以提高 SORT 的性能。由于这一扩展，TransTrack 能够通过更长的闭塞期来追踪物体，有效地减少了身份转换的次数。本着原始框架的精神，TransTrack 将大部分的计算复杂性放在离线预训练阶段，在这个阶段模型在大规模的人的重新识别数据集上学习深度关联度量。在在线应用中，TransTrack 使用视觉外观空间中的近邻查询来建立测量与跟踪的关联。实验评估表明，TransTrack 的扩展将身份切换的次数减少了 45%，在高帧率下实现了具有竞争力的整体性能

4.4 Efficient Transformer

4.5 Other tasks

5. Discussion&Conclusion

5.1 Conclusion

与卷积神经网络相比, Transformer 现在已经成为计算机视觉领域最热门的研究方向。受惠于 Transformer 架构强大的建模能力和迁移学习能力, Transformer 能在各项计算机视觉领域的任务取得与 CNN 相当甚至更胜的表现。

5.2 Future Prospects

Transformer 在计算机视觉领域的研究已经取得了很大的进展, 并且已经显示出在多个基准上接近或超过 CNN 方法的 SOTA 结果。但 Transformer 仍不成熟, 计算机视觉领域仍由 CNN 主导。

冗余消除: 在 NLP 领域中冗余消除就是消除词与词之间的重复表达, 但词的冗余度较低所以 Transformer 能在 NLP 领域成功应用; 但在计算机视觉领域, 以 ViT 为例, 一张图片被分为一定数量固定大小的 Token 且每个 Token 包含 16×16 个像素, 由于图像的局部相关性, 相邻的 Patch 之间有很大的相关性, 所以图像像素存在较大的冗余。如何解决各类任务元素的冗余性问题, 优化 Transformer 在计算机视觉领域的性能, 会成为未来的一个研究点。

效率优化: 在传统 CNN 中图片尺寸与计算复杂度呈线性关系, 对于 Transformer 计算长程依赖关系的计算复杂度与图片尺寸 (N) 呈 $O(N^2D)$ 的关系。虽然 Transformer 的建模能力强大, 但是需要更多的数据集和更长的训练时间和资源, 所以进一步优化 Transformer 的效率是未来的一个研究点。

视频处理: CNN 在计算机视觉的视频处理领域还没有实现与人脸识别, 目标检测所能达到的精度, Transformer 易于计算长程依赖的特性, 使得 Transformer 可能在时间和空间两个维度对视频信息建立强大的模型, 成为计算机视觉视频处理领域的标准范例。Transformer 对于视频信息的处理也将是未来关于 Transformer 研究的热点之一。

多模态任务: Transformer 在 NLP、CV 和 Audio 等领域的成功应用, 昭示着 Transformer 可能一统人工智能的各项领域。借助于 Transformer 的 Encoder-Decoder 架构, 多模态的任务能够简单的通过 Encoder 进行信息特征提取, 再通过 Decoder 进行解码, 完成 Encoder 提取的信息再其他领域的表示, 同时 Transformer 强大的建模能力也能够保证 Encoder 中信息的有效性和完整性。多模态任务也将是 Transformer 未来的研究热点之一。

自监督学习: 在计算机视觉领域, 虽然 Vision Transformers 可以取得比其他传统架构更好的结果, 但它们的成功取决于对数据的相当大的需求。因此, 以受监督的方式训练这些模型需要进行大量的标记工作, 这并不总是可行或可持续的。因此, 实现 Vision Transformers 的自监督方法可能是使这些模型不仅强大而且更容易应用于更广泛问题的一种可能方法。自监督学习也将是 Vision Transformer 未来的研究热点之一。