



# 

**Transformer研究报告**

**学 院 计算机学院**

**专 业 计算机科学与技术**

**学 号 2112005050**

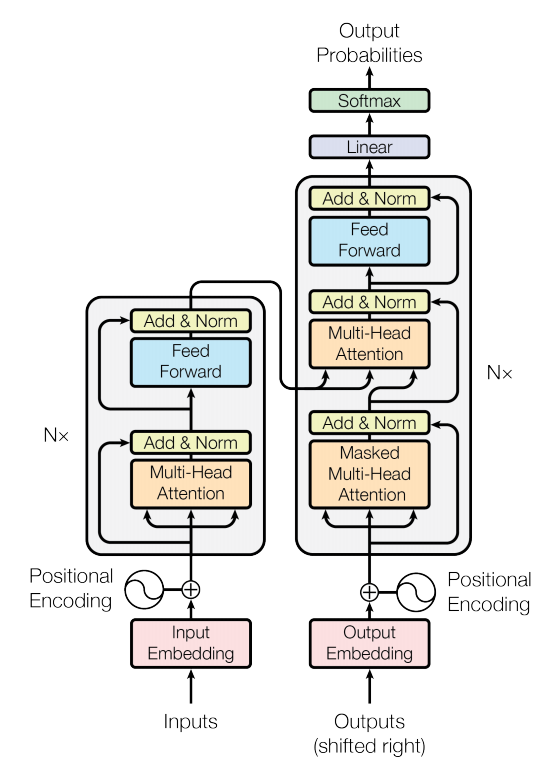
**学生姓名 林镇涛**

**2021 年 1月**

## transformer原理

### transformer结构

模型由6个编码器（Encoder）和6个解码器（Decoder） 组成，输入（Inputs）和输出（Outputs）通过同一个训练好的词嵌入层（Word Embedding）将输入字符转换为维度为d的向量。



### Encoder

Transformer对编码器（Encoder）与解码器（Decoder）两端的序列分别添加位置编码（Positional Encoding）。之后，编码经过含有多头自注意力机制（Multi-head Self-Attention）、位置前向传播网络（Position-wise Feed-Forward Network）、残差连接(Residual Connection)和层归一化（Layer Normalization）的计算单元。

1）Positional Encoding

由于Transformer的计算抛弃了循环结构的递归和卷积，无法模拟文本中的词语位置信息，因而需要人为添加。该过程称为位置编码（Positional Encoding）,使用词的位置向量，表示单词在句中的位置与词间的相对位置，在Transformer中与句子的嵌入表示相加，获得模型输入。

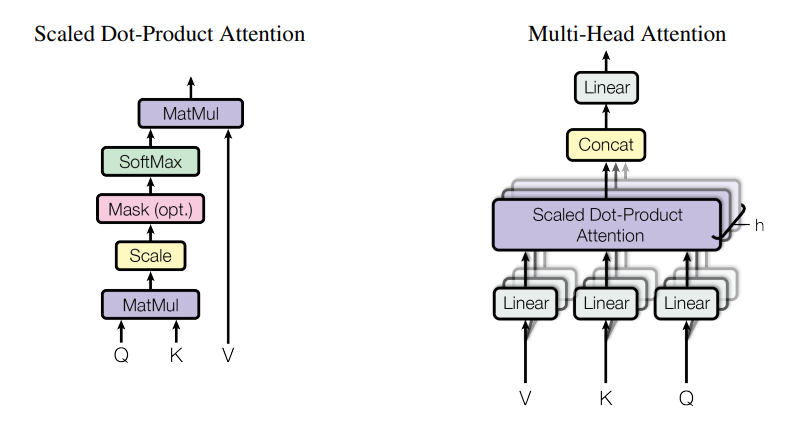
Position Encoder主要有三种：基于学习的参数、基于sin/cos函数表示、position作为连续函数。

Vaswani et.al提出 Positonal Encoding的两种方式：一种是学习得到，随机初始化后加入模型作为参数进行训练。一种是使用不同频率的正弦和余弦函数，在每个词语的词向量的偶数位置添加sin变量，奇数位置添加cos变量，以此填满Positional Encoding矩阵，该方式保证了每个位置向量不重复而且不同之间的位置之间存在线性关系，在计算注意力权重的时候(两个向量做点积)，相对位置的变化会对注意力产生影响，而绝对位置变化不会对注意力产生影响，这更符合常理。

上述方式都是将单词嵌入向量与位置嵌入向量分开训练，且只能接受整型数值作为词索引或位置索引，不能捕捉有序的关系，例如邻接关系或优先级，会导致每个位置上的位置向量是彼此独立的，称之为 Position Independence Problem[2]。针对这个问题，Benyou et.al 提出了一种改进方法——将以前定义为独立向量的词嵌入扩展为以位置 pos 为自变量的连续函数。这样词的表示可以随着位置的变化而平滑地移动，不同位置上的词表示就可以互相关联起来。实验结果表示，Complex Embedding 在性能上有较大的优化和提高。

2）Multi-Head Attention

注意力机制（Attention Mechanism）作为对Seq2Seq模型的改进而被提出。在机器翻译任务中，目标语言与输入语言的词汇之间，存在一些对应关系，2017年，谷歌弃用了循环神经网络结构，题出完全的基于transformer的self-attention，Bahdanau 鲜明地指出了固定大小的中间层维度对翻译任务的限制，并直接计算 BiRNN 对输入的编码h与候选输出的编码的s之间的相似度量作为替代，根据相似得分输出预测的目标语言词汇。这突破了 Seq2Seq 模型学习的中间表征h的维度对输入信息的限制。日后发展出的大多 Attention 机制仍可以表述在这一框架下，即计算输入的表示与输出表示间的得分 score，作为判断输出单词的依据，但不同的是，对于输入词的编码采用了不同的处理模型，例如RNN，CNN等。



### Decoder

Transformer 解码器（Decoder）与编码器（Encoder）具有相同的计算单元，不同的是，模型仍是 Seq2Seq 结构，因此，为了令解码器不提前得知未来信息，使用 Masked Multi-head Attention 隐藏下文。

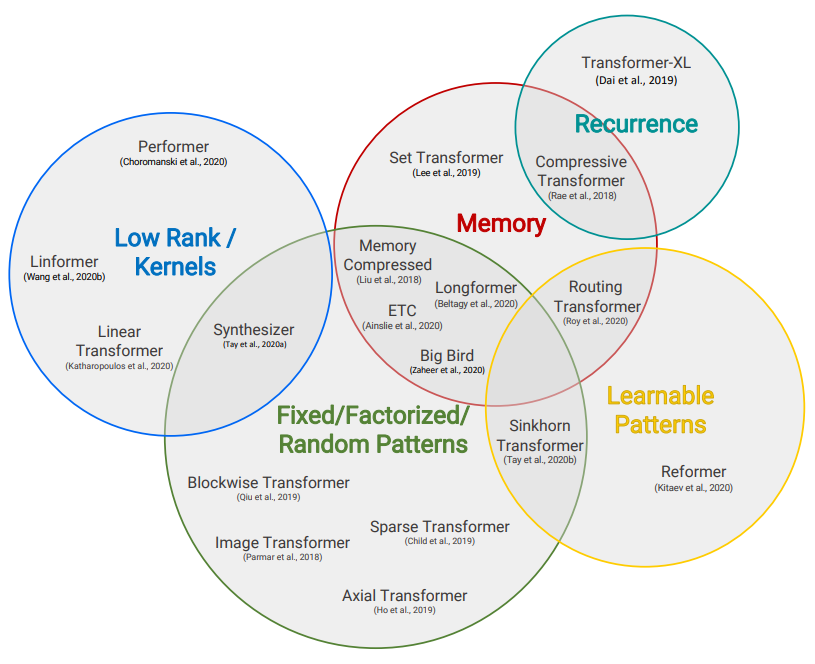
## transformer变种

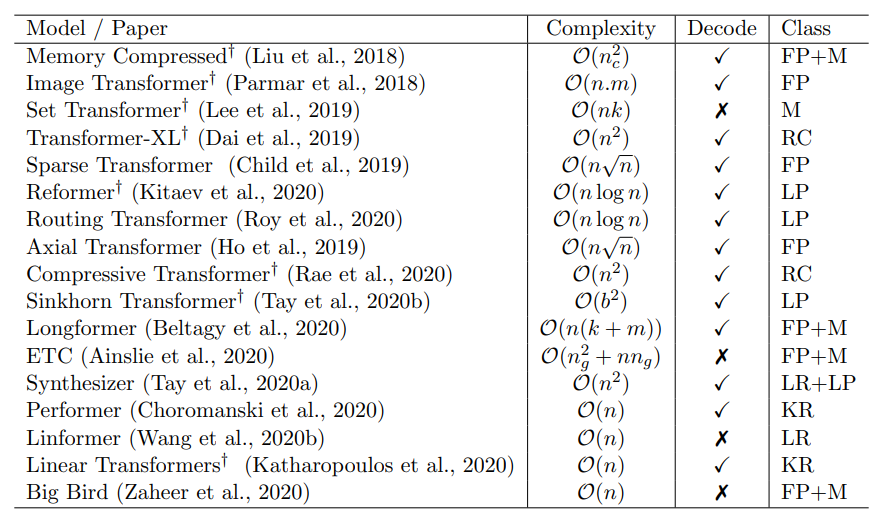
Transformer 发布后即在 NLP 多个领域取得显著成绩，也激发了领域研究者们极大的兴趣。后续研究者们在 Transformer 的基础上又提出了一系列 X-former模型，如 Set-Transformer、Transformer-XL 等。这些模型针对Transformer 的计算效率和存储效率进行优化，或是提升了模型处理长序列的效率。

谷歌2020年9月份在arXiv发表的综述论文《Efficient Transformers: A Survey》[16]即针对这一类进行了总结。

以上介绍了部分 Transformer 结构的主要应用模型与变体，概括了不同的模型优化方法，总的来说，按照模型提高效率的方法可以分为以下七类：

1. Fixed Patterns（固定模式）：将视野限定为固定的预定义模式，例如局部窗口、固定步幅块，用于简化注意力矩阵；
2. Combination of Patterns（组合模式）：通过组合两个或多个不同的模式来提高效率；
3. Learnable Patterns（可学习模式）：以数据驱动的方式学习访问模式，关键在于确定 Token 相关性。
4. Memory（内存）：利用可以一次访问多个 Token 的内存模块，例如全局存储器。
5. Low Rank（低秩）：通过利用自注意力矩阵的低秩近似，来提高效率。
6. Kernels（内核）：通过内核化的方式提高效率，其中核是注意力矩阵的近似，可视为低秩方法的一种。
7. Recurrence（递归）：利用递归，连接矩阵分块法中的各个块，最终提高效率。





## transformer在NLP

继17年谷歌大脑提出一个包含编码器和解码器的 Transformer 结构后，Transformer 在近三年间又有了许多不同的改进演化版本，对 Vanilla Transformer 的各种变体的技术线梳理可以按照时间来划分：

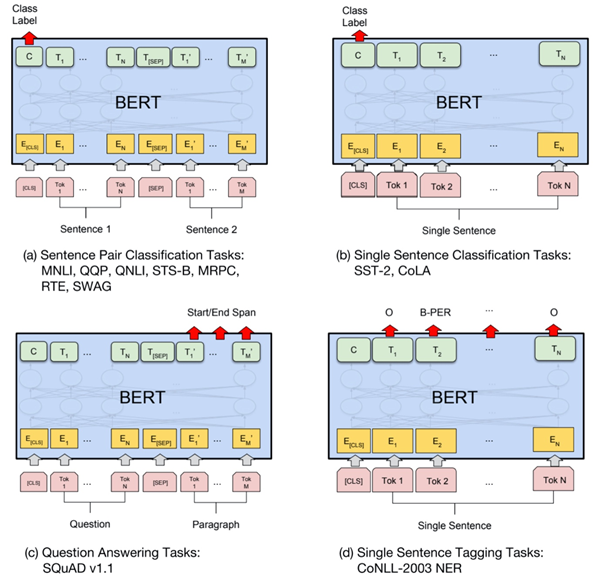
早期达到多个 SOTA 的结构优化模型是GPT、BERT，这两种模型都沿用了原始Transformer 的结构：GPT 把原始 Transformer 结构中的解码器结构用作编码器进行特征抽取，由于解码器中的 Mask 设计，使得 GPT 只使用了单向信息，即该语言模型只使用了上文预测当前词，而不使用下文，这也使得其更适用于文本生成类任务；而 BERT 只使用了原始 Transformer 结构中的编码器结构，因此其使用了双向的信息，并在一些任务中达到了更好的效果。

### 3.1 Bert

为了解决 GPT 采用单向语言模型所带来的问题，Google Brain 在2018年提出了基于双向 Transformer 的大规模预训练语言模型 BERT，刷新了11项 NLP 任务的 SOTA，具有里程碑意义。

在预训练阶段，BERT 提出了 Masked Language Model（MLM）与 Next Sentence Prediction(NSP) 的预训练方法，MLM 任务令模型预测文本中随机覆盖的缺失词，类似完形填空。同时，为避免覆盖所用高频标签 [MASK] 带来的误差，将其中部分标签换为随机词。NSP 任务则是利用 [CLS] 与 [SEP] 标签分隔上下句关系，以满足基于句子级别的下游任务的需要（如 QA、NLI 等）。

在微调阶段，BERT 和 GPT 并没有多少本质上的区别，论文中给出了几种不同任务的微调方案，如下图：



BERT 虽然有着非常好的表现，但它依然存在以下问题：预训练和微调阶段的不一致性（Discrepancy）：预训练 BERT 时，会随机 Mask 掉一些单词，但在微调阶段，却并没有诸如 [MASK] 之类的标签，从而导致两个过程的不一致；与标准的语言模型相比，BERT 并不具备很好的生成能力。

XLNet 是 BERT 之后比较引人注目的一个模型，提出了 Permutation Language Model 的新目标，即利用重排列的方式使得在单向的自回归 LM 中也能看到下文信息，这样结合了自回归 LM 擅长生成和自编码 LM 能够捕获双向信息的优点。但是，XLNet 的后续评价并没有高于 BERT，基于 XLNet 的后续研究也比较少，可能原因为 XLNet 使用的训练数据的质量要优于 BERT，这使得两者的对比并不公平。或者 BERT 根本没有得到充分的训练，因此没有发挥出其真正实力。

基于这种观点，Facebook 的研究人员提出了 RoBERTa。RoBERTa 的主要工作是通过改进训练任务和数据生成方式、使用更多数据、更大批次、训练更久等方式对 BERT 进行改进，换言之，RoBERTa 本质上算是一个调参调到最优的 BERT，其实际表现和 XLNet 不相上下。但无论是 XLNet 还是 RoBERTa 亦或是其它 BERT 变体，它们大多通过使用更多的数据、更复杂的模型等方式来提升训练效果，这极大加剧了内存开销。

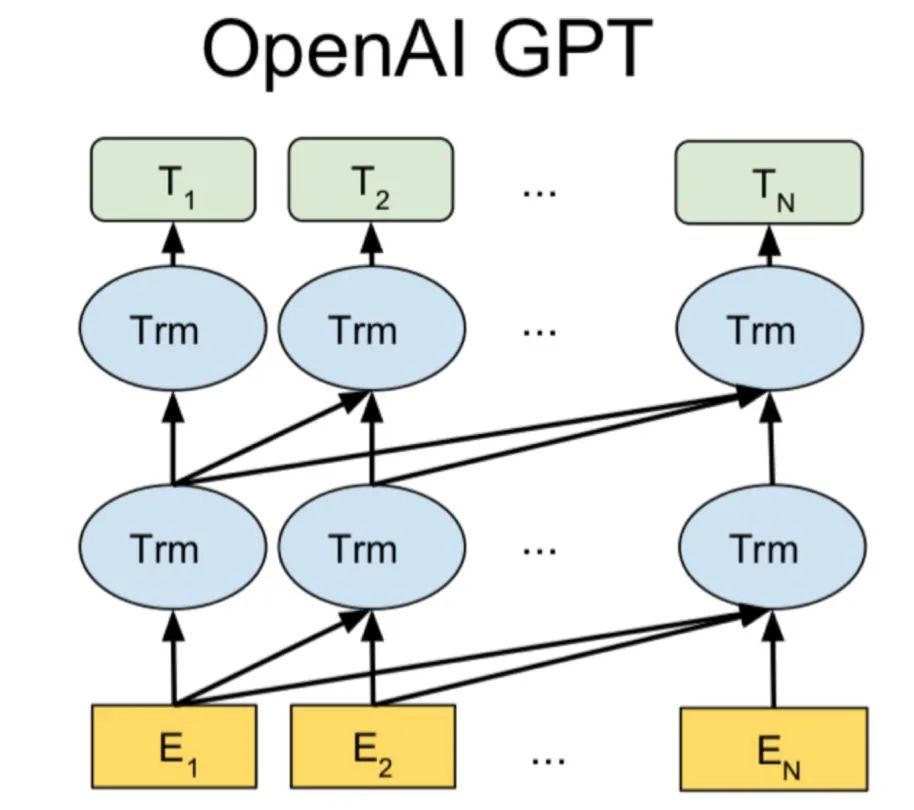
为了缓解这个问题，Google 提出了 ALBERT，希望用更少的参数、更低的内存开销来达到甚至提升 BERT 效果。论文中指出，增加参数量有时并不会提高模型性能，相反这可能会降低模型的效果，作者将该现象称为“模型退化”。基于此，ALBERT 提出了三种优化策略：分解词嵌入矩阵；跨层参数共享；将 NSP 任务换成更复杂的 SOP 任务，以补偿前两步性能上的损失。

### 3.2 GPT

基于 Transformer 的预训练+微调技术为 NLP 领域注入了新的活力，而这一工作的开创者，便是 OpenAI 的研究人员于2018年提出的 GPT。

论文中提出，我们可将训练过程分为两个阶段：第一阶段以标准语言模型作为目标任务，在大规模语料库上做无监督的预训练；第二阶段在特定的下游任务上对模型参数做有监督的微调，使其能够更好的适应特定目标任务。

基于上述过程，GPT 使用多层 Transformer Decoder 作为语言模型，其基本结构如下：



### 3.3 总结

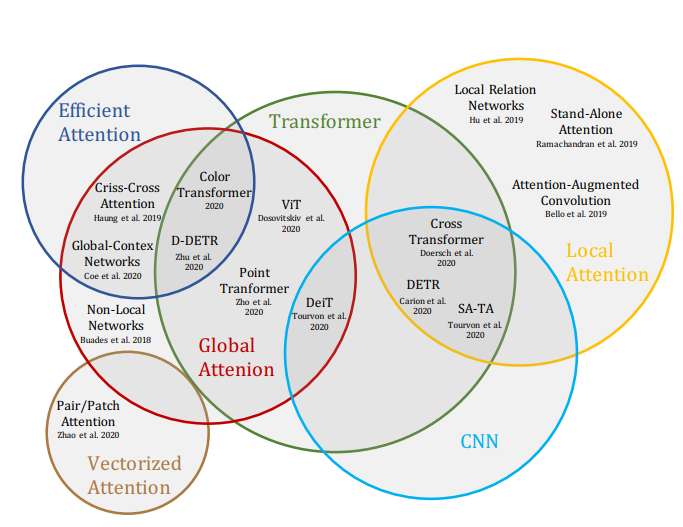
LSTM和Transformer都是当下主流的特征抽取结构，被应用到非常多的领域，各有它的擅长和优缺点，所以其实这个问题的范围比较大，个人认为还是要根据具体的研究领域进行讨论才有意义，我们需要范围内讨论一个特征抽取器是否适配问题领域的特点，毕竟目前很多模型改进的方向，其实就是改造得使得它更匹配领域问题的特性。

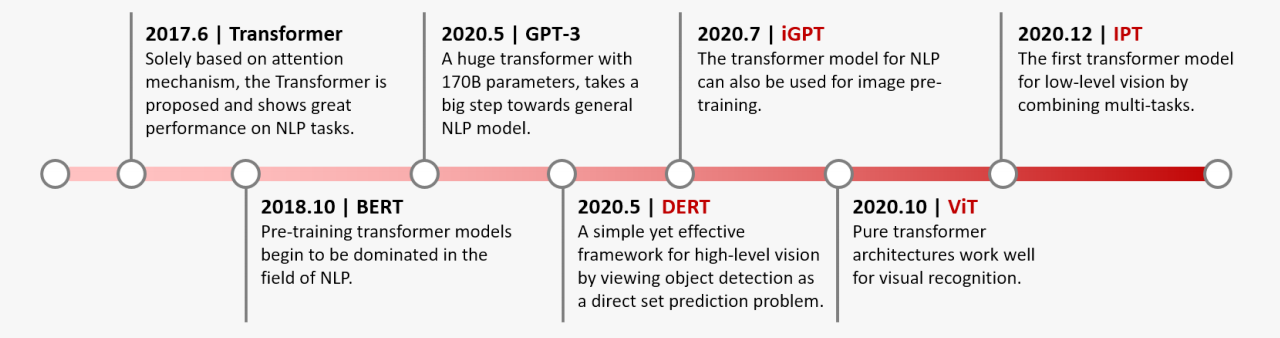
RNN的这种结构在某种程度上来讲，是在NLP领域火起来的，为什么？主要原因还是因为RNN的结构天然适配解决NLP的问题，NLP的输入往往是个不定长的线性序列句子，而RNN本身结构就是个可以接纳不定长输入的由前向后进行信息线性传导的网络结构。更何况为了解决标准RNN的梯度爆炸和长程信息消失问题，诞生了LSTM这种引入三个门的结构，对于捕获长距离特征非常有效，也正是因为RNN特别适合NLP这种线形序列应用场景，才使得它在NLP界如此流行的根本原因。

特别是在各种Attention注意力机制的加持下，几乎促使RNN在各个NLP的SOTA模型中频繁出没，通过叠加网络增加网络的深度，以及引入Encoder-Decoder框架，这些技术进展极大拓展了RNN的能力以及应用效果

Transformer使用Self-attention以及Muli-head Self-attention进行语义抽取（关于NLP句子中长距离依赖特征的问题，Self-attention天然就能解决这个问题，因为在集成信息的时候，当前单词和句子中任意单词都发生了联系，所以一步到位就把这个事情做掉了），并通过正弦位置编码保留输入句子单词之间的相对位置信息，这一套组合拳一打出来，威力惊人。不像RNN需要通过隐层节点序列往后传，也不像CNN需要通过增加网络深度来捕获远距离特征，Transformer在这点上明显方案是相对简单直观的。

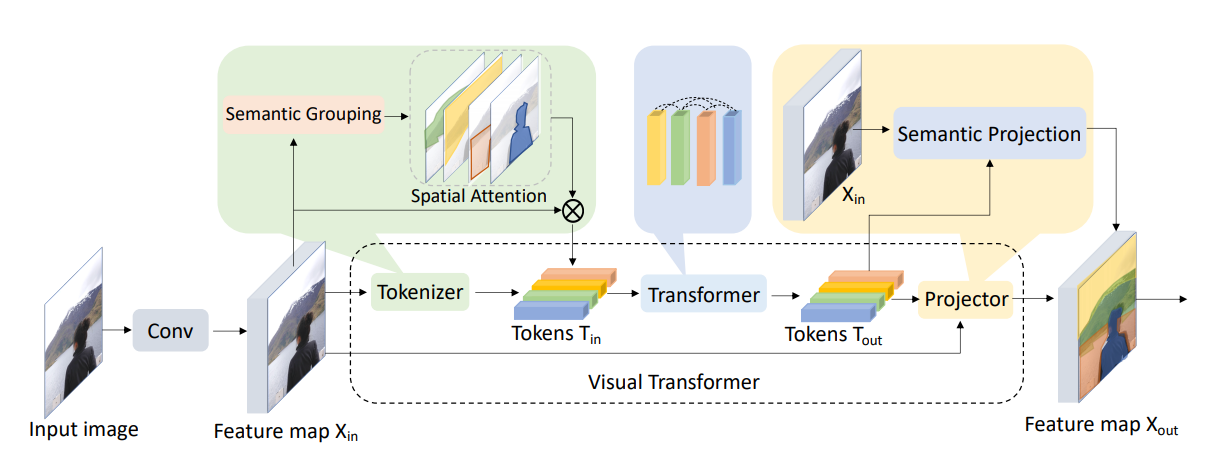
## transformer在CV





图像作为一种高维、噪声大、冗余度高的形态，被认为是生成建模的难点,这也是为什么过了好几年，transformer 才应用到视觉领域。比较初始的应用是在 Visual Transformer 一文中，作者使用 CNN 提取 low-level 的特征，然后将这些特征输 入Visual Transformer （VT）。

在 VT 中，作者设计了一个 tokenizer 将各个 pixel 分成少量的 visual tokens，每个 token 代表了图像中的某些语义信息。然后使用 transformer 来建模 token 之间的关系。最后输出的 tokens 直接被用来做分类或者处理一下恢复成特征图用于实例分割。



与这项工作不同的是，最近出现的 iGPT , ViT 和 DeiT 都是只使用 transformer 的文章。

在 CV 中使用 transformer，目前来看主要的两个问题，以及下列文章的核心区别在于：得到Token的方式、训练的方式、评估 representation 的方式。

只要得到 token，我们就能像 NLP 一样使用 transformer；有训练目标我们就能 train 我们的 model；有评估方式我们才知道自己的 model 好还是坏。接下来的几篇文章我也会从这几个方面进行讲解。

### 4.1 iGPT

尝试不使用CNN直接做image分类。作者提到，由于BERT 和 GPT-2 等 Transformer 模型是域不可知的，这意味着它们可以直接应用于任何形式的 1D 序列。所以作者直接把2D的图像处理成1D的序列，然后输入到Transformer，然后做next pixel prediction任务。

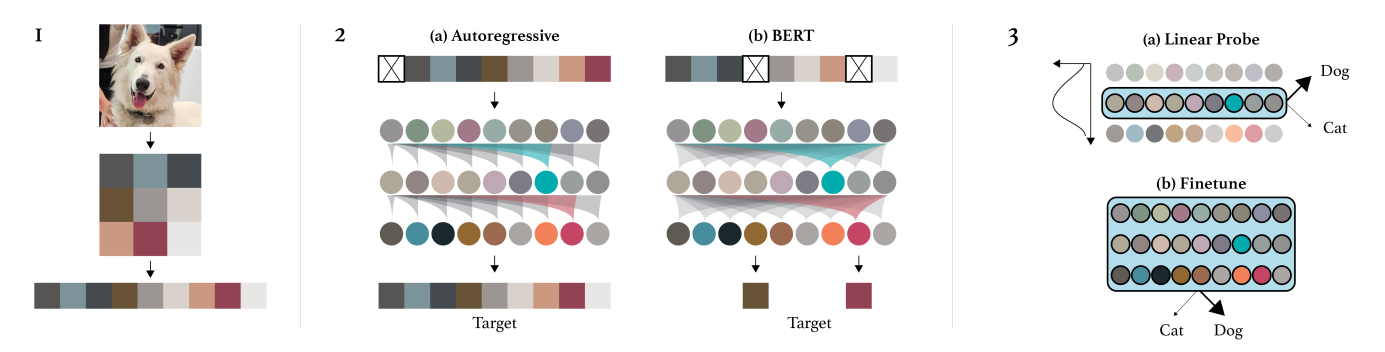
完整的架构如下图，最左边的图1是把图像由2D展平为1D，这里直接处理原像素。图2是作者使用的2种预训练目标：auto-regressive next pixel prediction 和masked pixel prediction，一个像AE一样直接预测所有像素，另一个类似BERT预测masked像素。

最后fine-turning。如图3有两种方式：

(i) fine-tune：增加了一个小的分类头，用于优化分类目标并adapt所有权重。

(ii)Linear-probe：将pretraining的模型视作特征提取器，增加一个分类头，只训练这个分类头。第二种方式的直觉在于“一个好的特征应该能够区分不同的类”。

fine-tune效果好有可能是因为架构很适合下游任务，但是linear-probe只取决于特征质量。

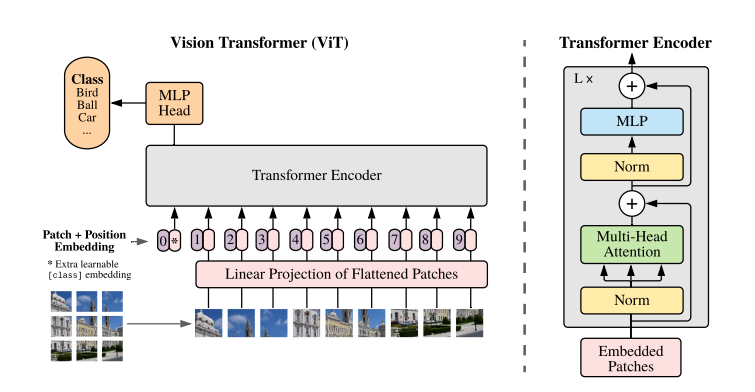


### 4.2 ViT

直接用Transformer来代替CNN。作者认为虽然Transformer相比CNN缺少一些归纳偏差，例如平移等变和局部性，但在数据集足够的情况下，泛化能力同样也是可以达到很强的。

ViT的具体架构如下图，直接将图像分割成小块，其中图像块的position处理方式与标记(单词)相同，然后输入到Transformer中，最后以监督的方式训练模型进行图像分类。此时因为不会使用CNN抽特征，那么输入存在一个问题：图像是2D，不用CNN的话不就没有特征做输入吗？

针对这一点，需要重塑图片为扁平的序列块，采用可训练的线性映射将patch进行平化，映射到D维上，即原先正常图片的H x W x C直接变成Nx（P^2 x C）。



目前来看，大多数应用都保留了 transformer 在 NLP 任务中的原始形态，这一形态不一定适合 images，因此是否会有改进版本，更加适合视觉任务的 transformer 尚且未知。除此之外，transformer 需要的数据量太大，缺少像 CNN 一样的 inductive biases，我们也很难解释他为什么 work，在本就是黑盒的DL领域又套了一层黑盒。

最后也是大多数非人民币玩家关注的点在于，基本的 ViT 模型需要 180 亿次浮点运算来处理一个图像。相比之下，轻量级的 CNN 模型 GhostNet 只需约 6 亿次 FLOPs 就能达到类似的性能，这个运算消耗实在太过昂贵。

### 4.3 总结

2020 年 10 月，谷歌提出了 Vision Transformer（ViT），能直接利用 Transformer 对图像进行分类，而不需要卷积网络。该模型可以获得与当前最优卷积网络相媲美的结果，但其训练所需的计算资源大大减少。谷歌在论文中写道：「这项研究表明，对 CNN 的依赖不是必需的。当直接应用于图像块序列时，transformer 也能很好地执行图像分类任务。」

2020 年 12 月，复旦大学、牛津大学、腾讯等机构的研究者提出了 SEgmentation TRansformer（SETR），将语义分割视为序列到序列的预测任务，该模型在 ADE20K 上排名第一，性能优于 OCNet、GCNet 等网络。

OpenAI用DALL·E 和 CLIP 打破了自然语言与视觉的次元壁。两个模型都利用 Transformer 达到了很好的效果，前者可以基于本文直接生成图像，后者则能完成图像与文本类别的匹配。

## transformer在语音

语音识别是人工智能领域研究历史很悠久的任务之一。从上世纪80年代起，高斯混合模型-隐马尔可夫模型(GMM-HMM)一直主导了语音识别的研究，随着深度学习的再次崛起和流行，发展出了深度神经网络-隐马尔可夫模型(DNN-HMM)[1]，但是这两种模型方案都需要使用HMM来进行解码识别。

LSTM与CTC的结合摆脱了对HMM的依赖实现了端到端的语音识别，随后Attention机制为NLP领域带来技术革新的同时也惠及了语音识别，BLSTM+Attention的模型组合成为新的语音识别范式。

Transformer在自然语言处理领域被提出后，陆续被引入到语音识别及计算机视觉领域，并表现非常出色。2018年Linhao Dongdeng人[2]率先将Transformer模型引入语音识别领域，提出Speech-transformer，这是一个完全依赖Attention机制学习位置依赖关系的无重复seq-seq模型。

Speech-Transformer提出后，有研究者将Transformer和RNN在各种ASR语料库中的应用进行了比较[3]。在他们的实验中，Transformer在13/15 ASR基准中表现出比RNN更大的优越性，证明了由于Transformer的并行训练能力及强大的序列建模能力，相比循环神经网络，基于Transformer的的编解码模型在语音识别任务中表现出更大优越性。

### 5.1 语音识别流式模型

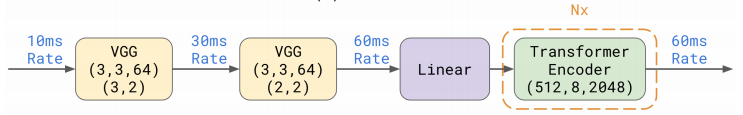
语音识别模型按使用场景可以分为离线模型和流式模型，离线识别模型在用户说完一句话或一段话之后再进行识别，而流式模型则是指用户还在说话的时同步进行识别，对识别的实时率有比较高的要求。由于Transformer编码器和解码器的自注意力机制需要对全序列上下文信息进行建模，同时编码器和解码器之间的注意力机制一次需要获取整个声学编码向量，计算时间复杂度和空间复杂度会随着语音时长的增加而变大，同时Transformer的多层结构会造成累积延迟，因此不能直接用于流式语音识别的场景中，将Transformer模型改进实现低延时的流式识别就显得非常有必要性，这个方向在学术研究上更具挑战性，同时也有非常大的工业应用价值。我们对近两年Transformer相关发表的大概15篇论文进行了调研，发现其中8篇都是关于Transformer流式识别的相关模型和技术方案，说明了transformer的流式解码也是ASR领域学术关注的热点。

目前看到使用较多的Transformer流式语音识别主要包括两类：Transducer结构模型，Joint Attention/CTC模型。

#### 5.1.1 transducer

在流式语音识别，RNN-T颇受欢迎。其结构如图3所示，原始RNN-T中，Audio Encoder和Label Encoder大多采用LSTM，Joint Network一般为前馈神经网络。RNN-T是一种序列到序列模型，但与其他序列到序列模型（如LAS，Transformer）有所不同的是，RNN-T不采用注意力机制，不需要一次获取整个输入语音编码序列，而是流式的处理输入并输出预测结果，是天生的流式结构。近几年研究人员也积极探索如何将Transfromer融入RNN-T中来进行流式语音识别[4-6]。

2019年Facebook提出了Transformer-Transducer[4]，在实验中，在将声学特征送入Audio Encoder前，堆叠两层VGGNet，此结构也相应被称为VGG-Transformer，一方面可以降低帧率，另一方面由于卷积能够很好的对位置信息进行编码[7]，因此实验中取消了额外的位置编码。



同时VGGNet采用因果卷积（Causal Convolution）防止未来信息的泄露。

在Audio Encoder的Transformer中使用截断的自注意力（图6（b））替代原来的全序列自注意力最终结果显示，当Audio Encoder为Transformer，Label Encoder为LSTM时表现最好，并且当截断自注意力的R=4，L=32时达到最优。

2020年谷歌提出了Transformer Transducer[5]，与[4]的思路类似，通过替换Audio Encoder和Label Encoder为Transformer，并且通过对Attention Score的Mask来限制Audio Encoder与Label Encoder自注意力的上下文信息来降低延迟，但与[4]不同的是，该论文除探究对Audio Encoder上下文信息限制外，还探究了对Label Encoder不同长度上文信息限制对模型性能的影响。

除上述两篇文章，华为也尝试了将Transformer与RNN-T融合，提出了Conv-Transformer Transducer[6]。其在Audio Encoder中通过在Transformer层之间加入卷积层来降低帧率，同时获取未来信息，该结构被称为Conv-Transformer，其不仅能显著降低训练内存需求，还不会造成精度损失。来限制上下文窗口，从而进一步降低延迟。

综上，我们可以看出，将Transformer与RNN-T融合并进行流式的改进主要有以下几个方面：

（a）将RNN-T的Aduio Encoder和Label Encoder前者或两者改为Transfomrer。

（b）限制Transformer中Self-Attention上下文信息。

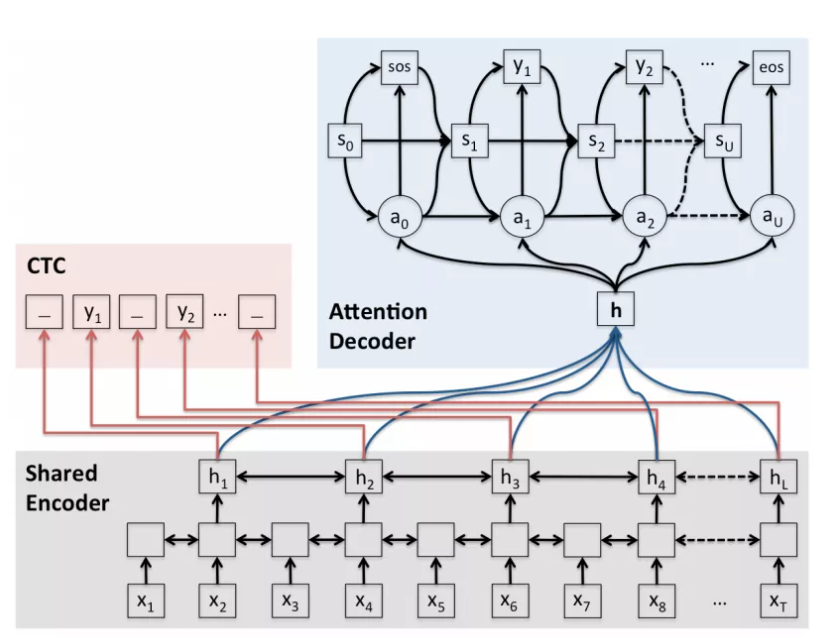
（c）通过在Audio Encoder中加入卷积层来降低帧率。

（d）Aduio Encoder中自注意力和卷积层往往只有一个可以获取到下文信息。

#### 5.1.2 Joint Attention/CTC

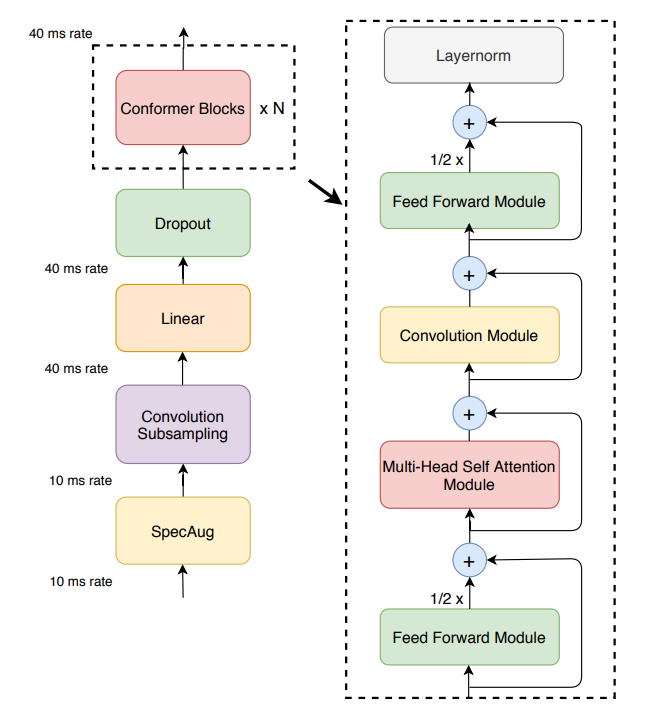
另外一种比较常用的流式语音识别的方案是Self-Attention与CTC联合使用的混合方案。端到端的语音识别模型主要分为两类：Attention-based和CTC-based，Transformer属于其中的Attention-based模型，而CTC基于条件独立性假设，每个时刻的输出仅依赖于当前时刻的输入，不需要等待完整的上文输入，通过引入空白符，保持输入与输出的单调对齐（mono-tonic alignment)，使用动态规划算法实现快速解码。但也因为CTC中每帧的输出是相互独立的，而一般对于语音识别的文字结果而言，由于文本中上下文有语义依赖，而CTC输出之间相互独立，因此为了实现更好的性能，在实际中使用CTC时往往会加上语言模型。

Attention-based通过注意力机制对音频输入的各帧进行联合建模，与CTC假设各帧之间条件独立刚好相反，如果结合两者的特点来构建一个混合模型，就能利用两种机制各自优点和差异来实现更鲁棒的识别效果。在实际应用中，一般设置CTC与Attention共享一个encoder，目标函数联合优化CTC loss和Attention loss，在训练过程中利用CTC 的对齐信息来辅助Attention的非单调对齐，加速模型训练收敛。在推理阶段，同样可以结合CTC和Attention的输出概率进行联合解码。



#### 5.1.3 卷积增强

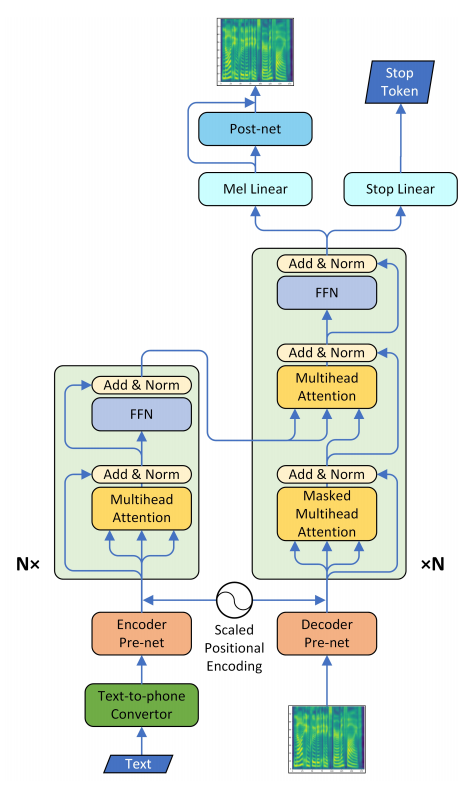
Transformer由于自注意力机制，擅长捕捉长序列依赖，在序列到序列任务中表现出色，而卷积则擅长对局部特征建模，在图像领域大放异彩。Conformer 结合两者优势，在语音识别准确率带入到新的高度。相比于原始Transformer，其主要改进在Encoder如下图所示，具体为：



### 5.2 语音合成

Neural Speech Synthesis with Transformer Network这篇论文虽然有8页纸，但是大部分章节都用来复述Original Transformer和Tacotron2的方法，作者把Transformer和Tacotron2融合，形成了Transformer-TATS。下面是模型的图示：

模型的主体还是Original Transformer，只是在输入阶段和输出阶段为了配合语音数据的特性做了改变。首先是Encoder的Input阶段，先将text逐字符转化为编号，方便Embedding，然后进入Encoder PreNet，这层网络由一个Embedding layer和三层卷积层构成，转化为512维的向量后，进入Transformer Encoder。其次是Transformer的Decoder部分，分为Input和Output。Input通过一个PreNet，将80维的梅尔声谱图转化为512维向量，这里的PreNet是一个三层的全连接网络（个人认为论文中应当解释一下为什么Encoder的PreNet是用卷积设计的，而Decoder的PreNet由全连接网络就可以解决问题）；Output部分与Tacotron2的设计完全一致。



## 参考：

[1]Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017): 5998-6008.

[2]Wang, Benyou, et al. "Encoding word order in complex embeddings." arXiv preprint arXiv:1912.12333 (2019).

[3]Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[4]Gehring, Jonas, et al. "Convolutional sequence to sequence learning." arXiv preprint arXiv:1705.03122 (2017).

[5]Cheng, Jianpeng, Li Dong, and Mirella Lapata. "Long short-term memory-networks for machine reading." arXiv preprint arXiv:1601.06733 (2016).

[6]Guo, Qipeng, et al. "Star-transformer." arXiv preprint arXiv:1902.09113 (2019).

[7]Tang, Gongbo, et al. "Why self-attention? a targeted evaluation of neural machine translation architectures." arXiv preprint arXiv:1808.08946 (2018).

[8]Kitaev, Nikita, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." arXiv preprint arXiv:2001.04451 (2020).

[9]Child, Rewon, et al. "Generating long sequences with sparse transformers." arXiv preprint arXiv:1904.10509 (2019).

[10]Zhang, Shuai, et al. "TensorCoder: Dimension-Wise Attention via Tensor Representation for Natural Language Modeling." arXiv preprint arXiv:2008.01547 (2020).

[11]Ioffe, Sergey , and C. Szegedy . "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift." (2015).

[12]Ba, Jimmy Lei , J. R. Kiros , and G. E. Hinton . "Layer Normalization." (2016).

[13]Xiong, Ruibin , et al. "On Layer Normalization in the Transformer Architecture." (2020).

[14]Lee, Juho, et al. "Set transformer: A framework for attention-based permutation-invariant neural networks." International Conference on Machine Learning. PMLR, 2019.

[15]Dai, Zihang, et al. "Transformer-xl: Attentive language models beyond a fixed-length context." arXiv preprint arXiv:1901.02860 (2019).

[16]Tay, Yi, et al. "Efficient transformers: A survey." arXiv preprint arXiv:2009.06732 (2020).

[17]Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018): 12.

[18]Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).

[19]Liu, Xiaodong, et al. "Multi-task deep neural networks for natural language understanding." arXiv preprint arXiv:1901.11504 (2019).

[20]余同瑞，金冉，韩晓臻，李家辉，郁婷.Tongrui Yu,ra自然语言处理预训练模型的研究综述,2020

[21]Xipeng Qiu,Tianxiang Sun,Yige Xu,etc.Pre-trained Models for Natural Language Processing: A Survey,2020

[22]Yoshua Bengio, et al. "A neural probabilistic language model." Journal of machine learning research, 3(Feb):1137–1155, 2003.

[23]Ronan Collobert, JasonWeston, L´eon Bottou, Michael Karlen,Koray Kavukcuoglu, and Pavel P. Kuksa. Natural language processing (almost) from scratch. J. Mach. Learn. Res., 2011.

[24]Hannun, Awni, et al. "Deep speech: Scaling up end-to-end speech recognition."arXiv preprint arXiv:1412.5567(2014).

[25] Linhao Dong, Shuang Xu, and Bo Xu. "Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition."2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018.

[26]Karita, Shigeki, et al. "A comparative study on transformer vs rnn in speech applications."2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU). IEEE, 2019.

[27] C.-F. Yehet al., "Transformer-Transducer: End-to-End Speech Recognition with Self-Attention,"\*\*p. arXiv:1910.12977Accessed on: October 01, 2019[Online]. Available:https://ui.adsabs.harvard.edu/abs/2019arXiv191012977Y

[28]Q. Zhanget al., "Transformer Transducer: A Streamable Speech Recognition Model with Transformer Encoders and RNN-T Loss,"\*\*p. arXiv:2002.02562Accessed on: February 01, 2020[Online]. Available:https://ui.adsabs.harvard.edu/abs/2020arXiv200202562Z

[29] W. Huang, W. Hu, Y. Yeung, and X. Chen, "Conv-Transformer Transducer: Low Latency, Low Frame Rate, Streamable End-to-End Speech Recognition,"\*ArXiv,\*vol. abs/2008.05750, 2020.

[30] Watanabe, S., Hori, T., Kim, S., Hershey, J. R., & Hayashi, T. (2017). Hybrid CTC/Attention Architecture for End-to-End Speech Recognition. IEEE Journal of Selected Topics in Signal Processing, 11(8), 1240–1253.