**Title:** A Survey of Transformers

**Abstract:**

作者在本文称Transformer的各种变体为X-formers，并综述了X-formers的各种变体，然后从结构改变、预训练、应用叙述了各种变体的区别，最后提出了些有潜力的未来研究方向

**Intro：**

transformers，最初是针对机器翻译，提出来的一个，seq2seq模型。之后，bert, gpt, ernie, T5, megatron, pangu-alpha, WuDao2.0等一众预训练模型，基于transformers的encoder或者decoder结构。为整个学术界和产业界都带来了变革。

在NLP, image, audio, video等众多方向上，基于transformers的技术，都取得了非常不错的效果。

在一些垂直领域，如化学，生物医学，制药，艺术设计，上，transformers的相关技术，也在逐步挑起大梁。

为了更好地适应五花八门的下游任务，transformers成为了X-formers家族。

相对于vanilla Transformer提升点如下：

1.模型有效性。例如对于长串序列的时间复杂度的基于”稀疏自注意力机制“的优化提升，以及基于”分而治之“策略的和并发网络以及多层网络的融合等。

2.模型泛化能力。transformer自己的参数规模还是比较大的，标配就是几千万级别的参数。所以，在小规模数据上训练的时候，容易over-fitting，即数据量不够。改善方法包括，引入结构化bias，正则化，在大规模无标注数据上预训练等等。

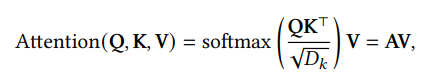
3.模型的领域适应性。即针对具体的领域化下游任务，来引入领域知识，设计有针对性的网络结构。

一个X-former，可能涉及到上面的不限于一个方向的提升。

例如，”稀疏注意力变体“（sparse attention variants），不单单降低了模型的计算复杂度，也引入了输入数据里面的”结构性的先验知识“，从而减轻在小规模数据上的over-fitting的问题。

**Background：**

1. Vanilla Transformer

输出是value的加权和

输出的维度和value的维度相同

QKV三者维度Dk是等长的，这是Dot-Product Attention的要求，若不等长，就要用additive attention

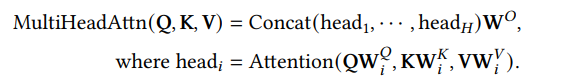
Q，query，是(N, Dk)形状的一个矩阵，其中N代表的是输入序列的长度（例如，128）。

K，key，是（M, Dk）形状的一个矩阵，其中M代表的是输入或者输出序列的长度（例如，128）。

V，value，是（M，Dk）形状的一个矩阵。

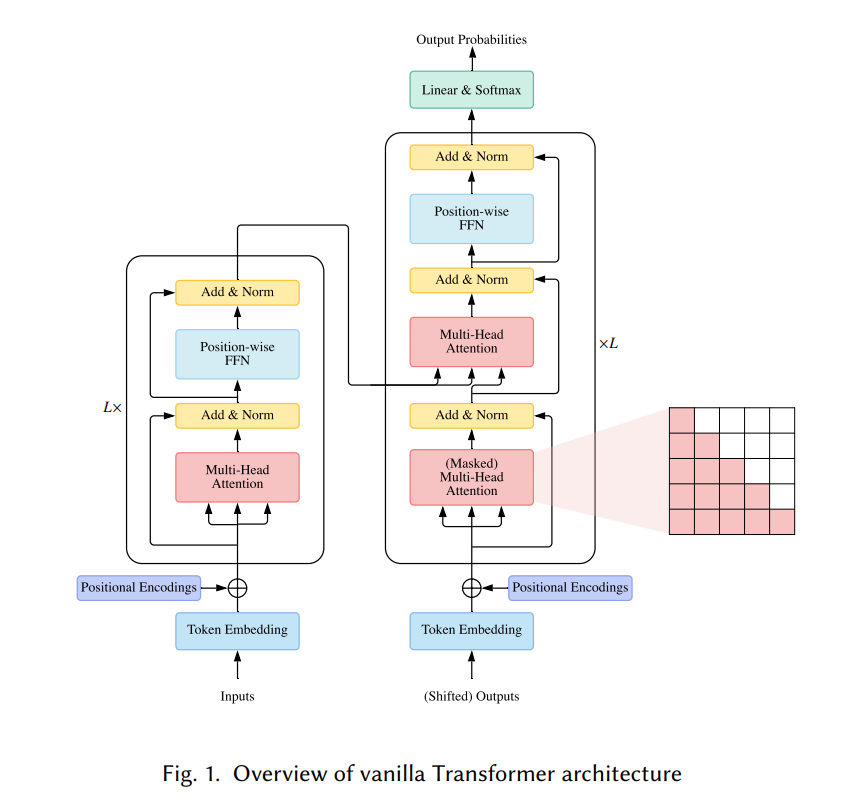
这样的话，QK^T的结果就是形状为（N, Dk）\*（Dk, M）-> (N, M)的矩阵。然后这个矩阵，经过除以sqrt(Dk)和softmax函数归一化之后，再和形状为(M, Dk)的矩阵V相乘，就得到了(N, Dk)形状的输出了。

多头注意力：



将输入X分为h份，通过乘h组权重来实现

模型结构图如下：



Self-attention的QKV都是X本身

Mask-attention在decoder，为自回归考虑，每个Q只能关注到自己及之前的KV，通过掩码矩阵实现

Cross-attention的Q为之前层的输出，KV为encoder的输出

左边的和右下的两个，都是self-attention：左边是全体可见的self-attention，右边下方的是masked self-attention（为的是”自回归“的考虑）。然后右侧上方的是cross-attention，其中的q来自目标语言序列，k和v则是来自源语言序列的”memory“。

Position-wise FFN.



相当于一个MLP

Residual Connection and Normalization.



残差连接，归一化LN

2. Model Usage

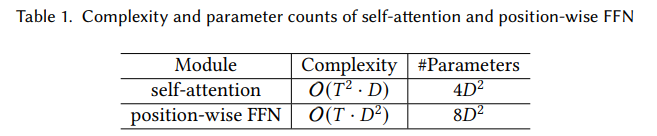
模型可以被用在三个方面

Encoder-Decoder：seq2seq方面

Encoder only：分类和序列标注

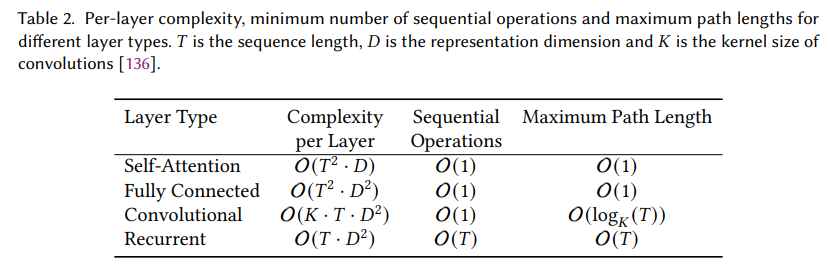
Decoder only：序列生成

3. Model Analysis

隐藏层大小为D

输入序列长度为T

4. Comparing Transformer to Other Network Types



自注意力网络，可以被看成一个全连通的网络层，其中的权重是从输入的tokens的“成对的关系”中自动学习而来的。

其一，[vs. FC] 路径短，参数少。self-attention和FC（全连通）的“最大路径长度”（最小）相同，都是O（1）的。所以，对于长序列（例如特别长的时序信息，长文档）具有“一定”的表达能力。只有TTD，相比FC的TTDD，self-attention使用了更少的参数，还可以multi-head self-attention，具有“一定”的“解释能力”。

其二，[vs. CNN]层数限制少可固定。为了使用全局的感受野，对于长度更长的序列，卷积网络需要更深的网络。self-attention其“最大路径长度”为O（1），即，self-attention机制下，可以使用固定的层数来表述长距离依赖。

其三，[vs. RNN]并行化，长距离依赖描述能力强。序列计算和最大路径长度上，self-attention都优于RNN。即self-attention具有更好的并行计算的架构，以及更好地捕捉长距离依赖关系的能力。

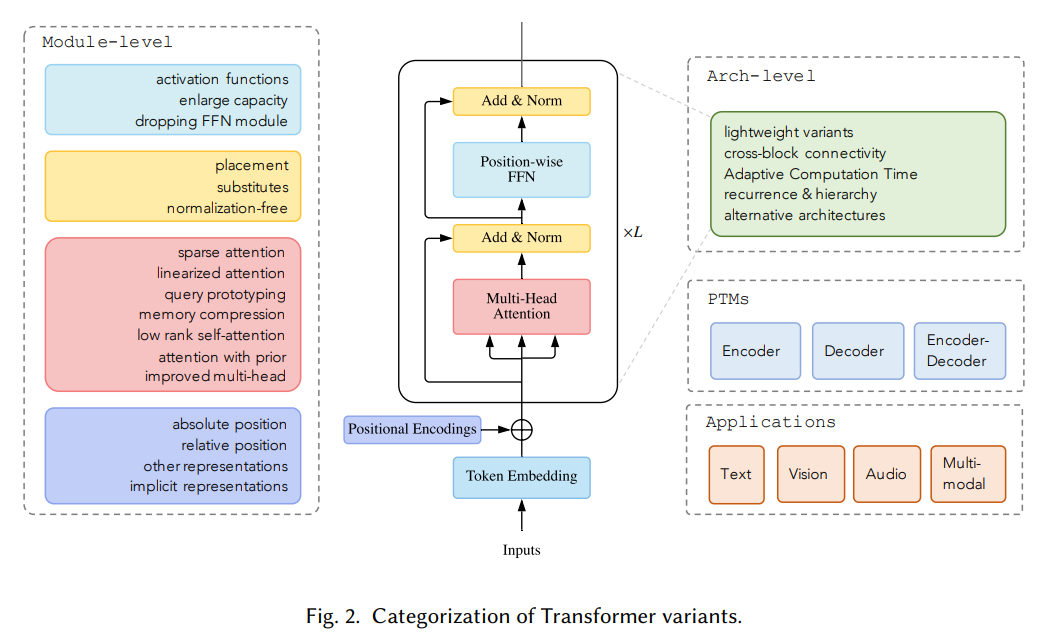
In Terms of Inductive Bias

CNN 会通过共享局部核函数来施加平移不变性和局部性的归纳偏差

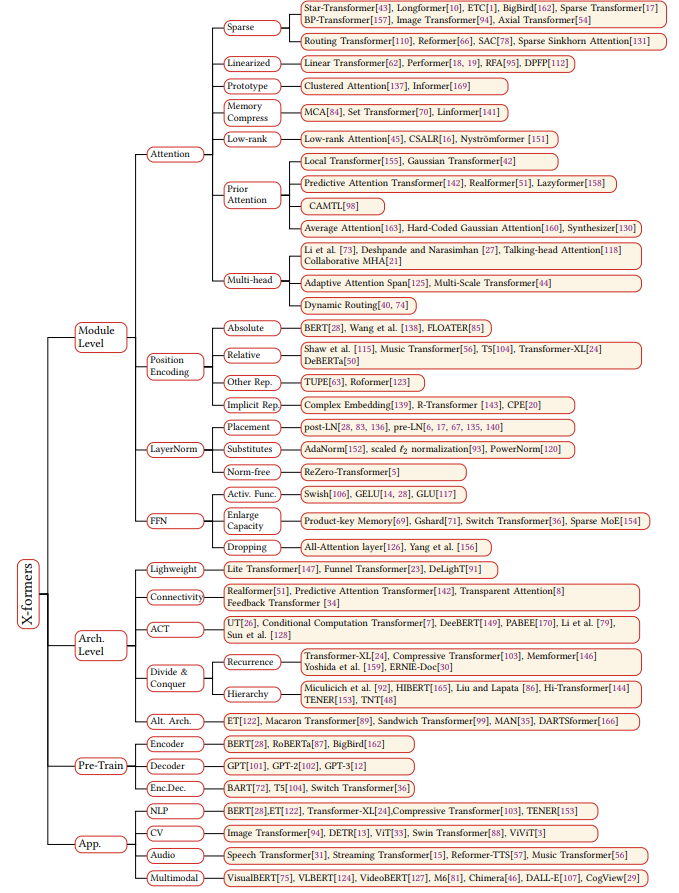
RNN 通过其马尔可夫结构携带时间不变性和局部性的归纳偏差。

与此对应地，transformers对于输入的数据的结构化信息，做更少的假设。从而使得transformer是更加一般化的，更加灵活的架构。但是，这个也有副作用，即对于小规模数据的话，Transformer倾向于overfitting。

**TAXONOMY OF TRANSFORMERS：**



其分为模块级别，架构级别，预训练模型，应用



**Attention：**

自注意力机制在Transformer中扮演着重要的角色。但是实际应用中，有两个挑战：

其一，复杂度，O(T\*T\*D)。是输入的序列长度T的二次方。成为了transformer的瓶颈段。

其二，结构化先验知识。自注意力机制对于输入，没有假设任何结构化的bias。也就是说，即使是order information（序列顺序信息），transformer也需要从训练数据里面学习。因此，Transformer很容易在中小规模数据上overfit。

那么，为了提升自注意力机制，有几个方向：

第一，稀疏自注意力。主要是引入sparsity bias，对T \* T的自注意力进行一定的“剪枝”，从而降低计算复杂度；

第二，线性注意力（Linearized attention）。使用核特征映射（kernel feature maps）来分解(disentangle)注意力矩阵。然后用逆序来计算注意力，从而达到线性复杂度。

第三，原型和内存压缩。通过减少queries或者key-value memory pairs的数量，来降低注意力矩阵的大小。

第四，低秩自注意力。（low-rank self-attention）捕捉自注意力矩阵的low-rank特性。

第五，带先验的注意力。（attention with prior）利用新补充的知识，或者用先验注意力分布来替换掉标准注意力模块。

第六，其他多头机制。一些新的多头“注意力”机制来替换当前的。

**CONCLUSION AND FUTURE DIRECTIONS：**

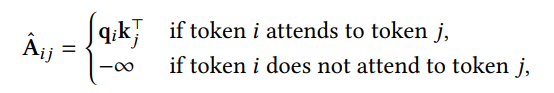
在这篇综述中，我们对X-formers进行了全面的概述，并提出了一个新的分类方法。大多数现有的作品从不同的角度提高了Transformer，比如效率和泛化能力以及应用方面。这些改进包括结合结构性先验、设计轻量级架构、预训练等等。尽管X-formers已经证明了它们在各种任务中的能力，但仍然存在挑战。除了目前的关注点（例如效率和泛化能力），Transformer的进一步改进可能在于以下方向：

（1）理论分析。Transformer的架构已经证明能够支持具有足够参数的大规模训练数据集。许多作品表明，Transformer比CNN和RNN具有更大的容量，因此能够处理大量的训练数据。当Transformer在充足的数据上进行训练时，它的性能通常比CNN或RNN好。一个直观的解释是，Transformer对数据结构的先验假设很少，因此比CNN和RNN更灵活。然而，理论上的原因还不清楚，我们需要对Transformer的能力进行一些理论分析。

（2）超越注意力的更好的全局交互机制。Transformer的一个主要优点是使用注意力机制来建模输入数据中节点之间的全局依赖关系。然而，许多研究表明，对大多数节点来说，完全注意力是不必要的。在一定程度上，不加区分地计算所有节点的注意力是低效的。因此，在有效地建模全局交互方面仍有很大的改进空间。一方面，自注意力模块可以看作是一个具有动态连接权重的全连接神经网络，它用动态路由聚合非本地信息。因此，其他动态路由机制是值得探索的替代方法。另一方面，全局交互也可以通过其他类型的神经网络进行建模，如记忆增强模型。

（3）多模态数据的统一框架。在许多应用场景中，整合多模态数据是有用且必要的，以提升任务性能。此外，通用AI也需要捕捉不同模态之间的语义关系。由于Transformer在文本、图像、视频和音频上取得了巨大的成功，我们有机会构建一个统一的框架，更好地捕捉多模态数据之间的内在联系。然而，设计同模态和跨模态的注意力仍然有待改进。

**Sparse Attention：**



只有当token i注意到token j的时候，我们才计算他们的内积。否则这个值就是负无穷

1. Position-based Sparse Attention

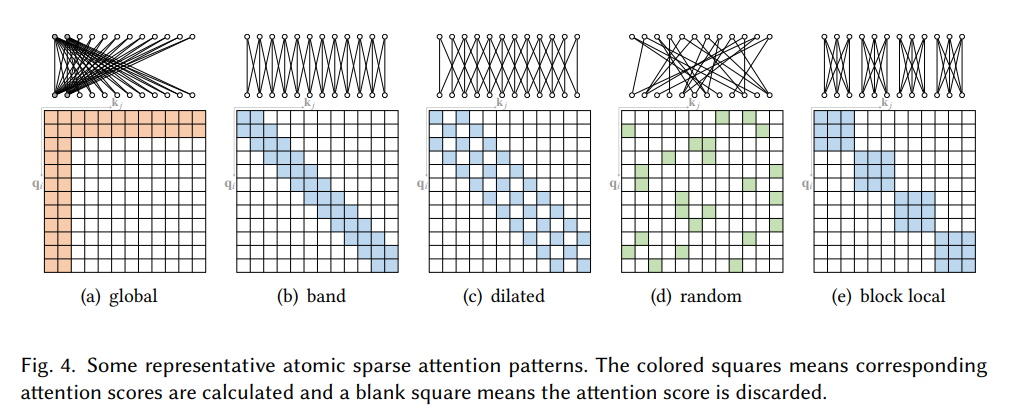
原子级别的稀疏注意力分为

global-全局：增加两个全局结点（能和所有节点产生注意力交集），减轻长距离依赖描述能力的退化

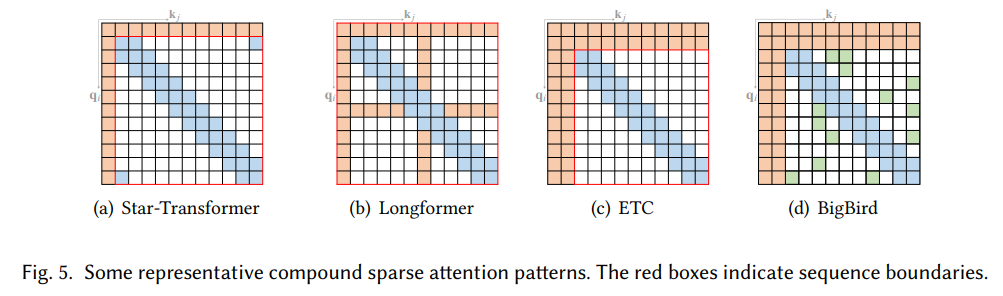
band-带状：滑动窗口注意力

dilated-膨胀（跨步）：步长大了

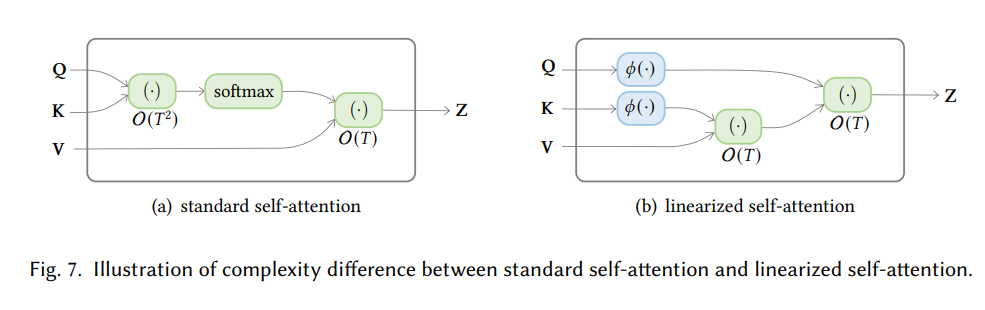
random-随机：提高非局部的交互能力

block local - 块状局部：把输入序列，分成几个不相交的query块，每个块有局部memory块来实现注意力交互。

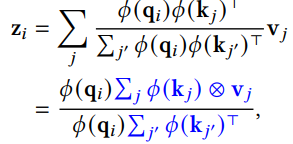
复合-稀疏注意力机制

实际的对于注意力机制的改造，一般是上述几个”原子“操作的”复合体“。

2.Linearized Attention



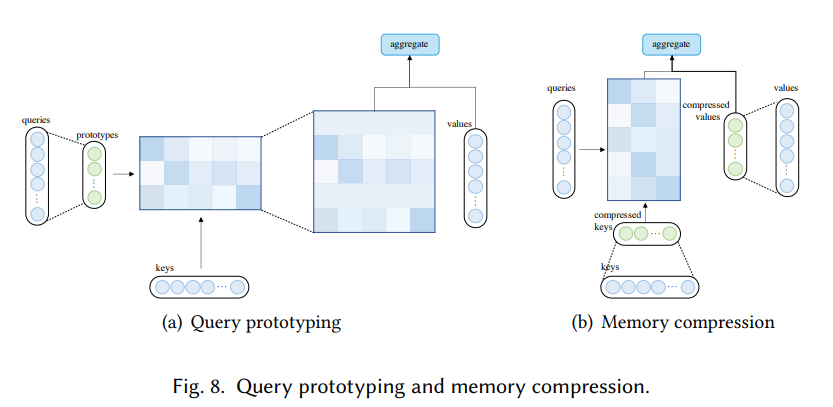
线性化注意力是一类方法，它们将非归一化注意力矩阵exp(QK^T)近似或替换为，其中是一个以逐行方式应用的特征映射。

注意力输出变为左图

3. Query Prototyping and Memory Compression

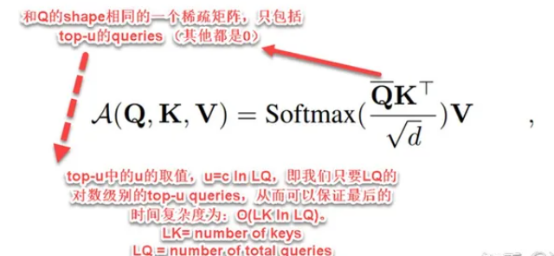
Attention with Prototype Queries.

对query，key，value的维度进行压缩，从而降低计算量。

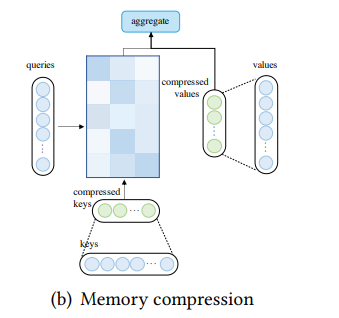


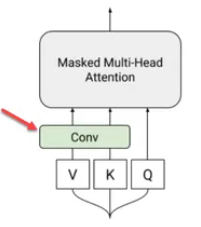
Informer提出

ProbSparse (probability sparse – 概率分布稀疏) self-attention

Informer使用显式的查询稀疏度度量来从查询中选择原型，该查询稀疏度度量源于查询的注意力分布与离散均匀分布之间的Kullback-Leibler散度的近似。然后仅针对查询稀疏度度量下的前top-u个查询计算注意力分布。 其余查询被分配以离散均匀分布。

Attention with Compressed Key-Value Memory

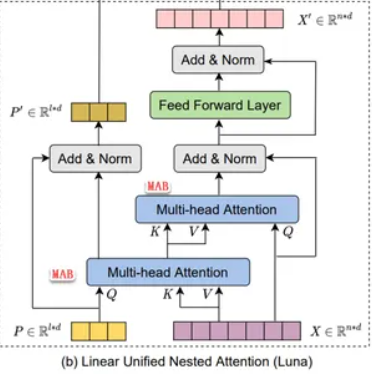
分别对key和value进行压缩从而降低计算复杂度。

Liu et al. propose Memory Compressed Attention (MCA)：KV中间加一个卷积层

Set Transformer

使用若干额外可训练的全局结点来对输入序列进行“摘要化”处理。然后这个“摘要”就是输入序列要执行self-attention的keys/values

Luna = linear unified nested attention



和set transformer很像

Linformer

这篇论文证明了一个很重要的定理：Self-attention is low-rank

对于原始的注意力矩阵P，我们总是可以找到它的一个近似矩阵，这个矩阵的秩是输入序列长度的对数：logn。这个近似矩阵乘以w（n维度列向量，VWi^V中）可以无限接近原始矩阵乘以w（n维度列向量，VWi^V中）。

Poolingformer

这个论文面向的是长文档建模。包括了一个“双层”注意力模式：

其一，使用小的滑动窗口模板来从邻居token汇集信息；

其二，基于池化注意力来扩大窗口从而扩大感受野

可以有效降低计算复杂度以及内存使用。

4. Low-rank Self-Attention

这个，在Linformer里面，有定理证明，可以拿一个低秩的矩阵来“无限”逼近一个自注意力矩阵。

这个新的秩，有多小呢？可以认为是输入序列长度T的一个对数，logT。这样就把原本的O(T^2)的问题，映射成了O(TlogT)的问题了。

而应用起来，有两个方向：

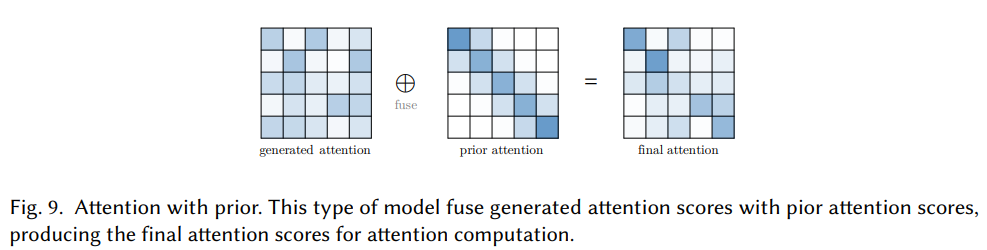
其一，低秩的属性，可以显式地用参数来建模；

其二，自注意力矩阵可以被一个低秩矩阵替换掉。

5. Attention with Prior

带有先验的注意力

transformer中的自注意力机制，本质上是构造一个"attention distribution" (即queries和keys之间的内积，然后做softmax概率化取值）来作为values的权重，然后对values进行加权求和，就得到了最后的结果输出。一般情况下，这个attention distribution（注意力分布）也可以来自其他渠道。称这样的渠道资源为prior-先验。



可以在softmax之前完成

在位置上使用高斯分布，这样，一个生成出来的注意力矩阵，可以和一个高斯分布的密度函数构成的矩阵相乘，然后再归一化。这就相当于A+G，其中G(i,j)表示第i个token和第j个token之间的关系的先验概率。

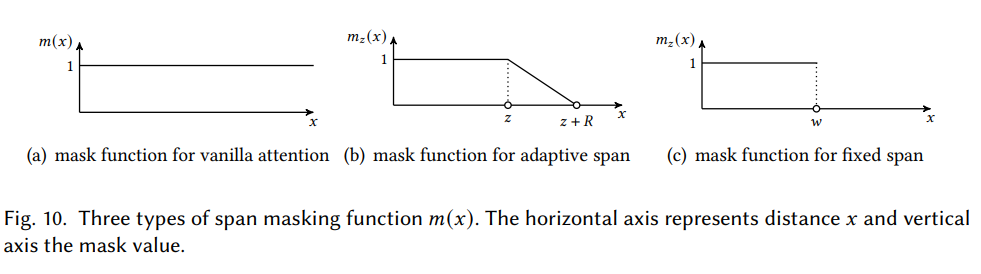
6. Improved Multi-Head Mechanism

虽然多头机制试图在不同的子空间来联合描述query, key, value之间的关系，不过目前为止，还没有机制确保不同的注意力head可以捕捉不同的特征维度。

Head行为

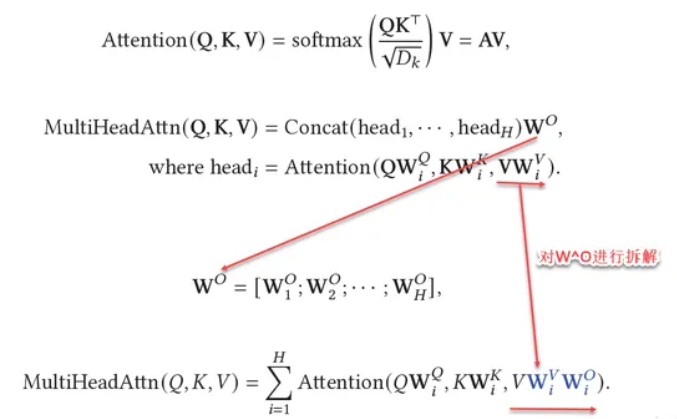
一些论文讨论了head的行为和交互

限定跨度的多头注意力

（a）原始的attention（b）可适应跨度的多头（c）固定跨度的多头

Multi-head with Refined Aggregation

带精细聚合的多头



剩余其他模块的改造

位置编码、layerNorm、FFN

X-formers的介绍

轻量级的transformer

-低秩注意力

-lite transformer

-funnel transfoprmer

-DeLighT

强化模块化连接

-Realformer

-predictive attention transformer

-transparent attention

-feedback transformer

改造计算时间

-ACT-38,adaptive computation time

-unuiversal transformer(UT)

-DeeBERT

-PABEE

-SENTEE,Li+-

-Sun+-

**APPLICATIONS OF TRANSFORMER**

Transformer最初是为机器翻译设计的，但由于其灵活的架构，已广泛适用于自然语言处理之外的各个领域，包括计算机视觉和音频处理。

（1）自然语言处理。Transformer及其变体已在NLP任务中得到广泛探索和应用，例如机器翻译，语言建模和命名实体识别。大量的努力致力于在大规模文本语料库上预训练Transformer模型，这被认为是Transformer在NLP中广泛应用的主要原因之一。

（2）计算机视觉。Transformer也已适应于各种视觉任务，例如图像分类，对象检测，图像生成和视频处理。Han等人和Khan等人提供了有关视觉Transformer现有工作的评述。

（3）音频应用。Transformer也可以扩展到音频相关应用，例如语音识别，语音合成，语音增强和音乐生成。

（4）多模态应用。由于其灵活的架构，Transformer也已应用于各种多模态场景，例如视觉问答，视觉常识推理，字幕生成，语音到文本翻译和文本到图像生成。