# 关于SELF-ATTENTION和卷积层之间的关系

**Jean-Baptiste Cordonnier, Andreas Loukas & Martin Jaggi**
École Polytechnique Fédérale de Lausanne (EPFL)
{first.last}@epfl.ch

## ABSTRACT

最近在视觉中纳入注意力机制的趋势促使研究人员重新考虑卷积层作为主要构建模块的优越性。除了帮助cnn处理长程依赖关系，Ramachandran et al. (2019)表明，注意力可以完全取代卷积，并在视觉任务上取得最先进的性能。这就提出了一个问题:学习注意力层的操作与卷积层类似吗? 这项工作提供了证据，表明注意力层可以进行卷积，实际上，它们经常在实践中学会这样做。证明了具有足够数量头的多头自注意力层至少与任何卷积层一样具有表现力。数值实验表明，自注意力层与CNN层类似，关注像素网格模式，这证实了我们的分析。我们的代码可以在[1]上公开。

## 1 简介

自然语言处理(NLP)的最新进展在很大程度上归功于*transformer*的兴起(Vaswani et al., 2017)。预训练以解决大型文本语料库上的无监督任务，基于transformer的架构，如GPT-2 (Radford et al., 2018), BERT (Devlin et al., 2018)和Transformer-XL (Dai et al., 2019)，似乎具有学习文本底层结构的能力，因此，可以学习跨任务的表示。transformer与之前的方法(如循环神经网络(Hochreiter & Schmidhuber, 1997)和卷积神经网络(CNN))之间的关键区别在于，前者可以同时关注输入序列中的每个单词。这要归功于注意力机制——最初在神经机器翻译中引入的注意力机制，以更好地处理长程依赖(Bahdanau et al., 2015)。特别是自注意力，序列中两个单词的相似性由测量其表示距离的注意力分数捕获。然后，根据注意力分数最高的单词更新每个单词的表示。

受其学习单词之间有意义的相互依赖关系的能力的启发，研究人员最近考虑在视觉任务中使用自注意力。自注意力首先通过使用基于通道的注意力(Hu et al., 2018)或跨图像的非局部关系(Wang et al., 2018)添加到CNN中。最近，Bello et al. (2019)通过将一些卷积层替换为自注意力层来增强cnn，从而改进了图像分类和目标检测任务。有趣的是，Ramachandran et al. (2019)注意到，即使将注意力和卷积特征相结合时取得了最先进的结果，但在相同的计算和模型大小约束下，仅自注意力架构也达到了具有竞争力的图像分类精度。

这些发现提出了一个问题，自注意力层是否以类似于卷积层的方式处理图像? 从理论角度来看，人们可以辩称，transformer有能力模拟任何功能——包括CNN。事实上，Pérez et al. (2019)表明，在一些强大的理论假设下，例如无界精度算法，基于注意力的多层加性位置编码架构是图灵完备的。不幸的是，普适性结果并没有揭示机器如何解决任务，只是揭示了它有能力这样做。因此，自注意力层实际上如何处理图像的问题仍然是开放的。

贡献。 本文提出了理论和经验证据，证明自注意力层可以(并且确实)学习与卷积层类似的行为:

I. 从理论角度出发，提供了一个建设性的证明，表明自注意力层可以表达任何卷积层。

使用相对位置编码的单个多头自注意力层可以被重新参数化以表达任何卷积层。

II. 我们的实验表明，仅使用注意力的架构(Ramachandran et al., 2019)的前几层确实学会了关注每个查询像素周围的类似网格的模式，这与我们的理论构造类似。

引人注目的是，这种行为在我们的二次编码和学习到的相对编码中都得到了证实。我们的结果似乎表明，局部卷积对于图像分类网络的前几层来说是正确的归纳偏差。本文提供了

[1]代码:github.com/epfml/attention-cnn。网站:epfml.github.io/attention-cnn。

一个交互式网站[2]，以探索自注意力如何在较低层利用基于位置的局部注意力和在较深层利用基于内容的注意力。为了重现性的目的，我们的代码是公开的。

## 2 视觉注意机制背景

本文回顾了自注意力层的数学公式，并强调了位置编码的作用。

### 2.1 多头自注意力层

设$\boldsymbol{X} \in \mathbb{R}^{T \times D_{in}}$是一个输入矩阵，由$T$个token组成，每个token包含$D_{in}$个维度。虽然在NLP中，每个标记对应于句子中的一个单词，但相同的形式可以应用于任何$T$离散对象的序列，例如像素。自注意力层将任何查询令牌$t \in [T]$从$D_{in}$映射到$D_{out}$维度，如下所示:

$$\text{Self-Attention}(\boldsymbol{X})_{t,:} := \text{softmax}\left(\boldsymbol{A}_{t,:}\right)\boldsymbol{X}\boldsymbol{W}_{val}, \tag{1}$$

我们在哪里引用$T \times T$矩阵的元素

$$\boldsymbol{A} := \boldsymbol{X}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\boldsymbol{X}^{\top} \tag{2}$$

作为注意力得分和softmax 输出[3]作为注意力概率。该层由查询矩阵$\boldsymbol{W}_{qry} \in \mathbb{R}^{D_{in} \times D_k}$、密钥矩阵$\boldsymbol{W}_{key} \in \mathbb{R}^{D_{in} \times D_k}$和值矩阵$\boldsymbol{W}_{val} \in \mathbb{R}^{D_{in} \times D_{out}}$参数化。为简单起见，我们排除了任何残差连接、批归一化和常数因子。

上面描述的自注意力模型的一个关键属性是，它对重排序是等变的，也就是说，它提供相同的输出，而不依赖于$T$输入标记的打乱方式。在我们期望事物的顺序很重要的情况下，这是有问题的。为了缓解限制，为序列中的每个标记(或图像中的像素)学习位置编码，并在应用自注意力之前将其添加到标记本身的表示中

$$\boldsymbol{A} := (\boldsymbol{X} + \boldsymbol{P})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\boldsymbol{X} + \boldsymbol{P})^{\top}, \tag{3}$$

其中$\boldsymbol{P} \in \mathbb{R}^{T \times D_{in}}$包含每个位置的嵌入向量。更一般地说，$\boldsymbol{P}$可以用任何返回位置向量表示的函数替换。

实践发现，将这种自注意力机制复制到多个头部是有益的，每个头部都能够通过使用不同的查询、键和值矩阵专注于输入的不同部分。在多头自注意力中，输出维度$D_h$的$N_h$头的输出被连接并投影到维度$D_{out}$，如下所示:

$$\text{MHSA}(\boldsymbol{X}) := \underset{h \in [N_h]}{\text{concat}}\left[\text{Self-Attention}_h(\boldsymbol{X})\right]\boldsymbol{W}_{out} + \boldsymbol{b}_{out} \tag{4}$$

引入两个新参数:投影矩阵$\boldsymbol{W}_{out} \in \mathbb{R}^{N_h D_h \times D_{out}}$和偏置项$\boldsymbol{b}_{out} \in \mathbb{R}^{D_{out}}$。

### 2.2 图像注意力

卷积层是构建处理图像的神经网络的事实选择。我们记得，给定宽度$W$，高度$H$和$D_{in}$通道的图像张量$\mathbf{X} \in \mathbb{R}^{W \times H \times D_{in}}$，像素$(i, j)$的卷积层输出由

$$\text{Conv}(\boldsymbol{X})_{i,j,:} := \sum_{(\delta_1,\delta_2) \in \mathbb{\Delta}_K} \mathbf{X}_{i+\delta_1, j+\delta_2,:}\mathbf{W}_{\delta_1,\delta_2,:,:} + \boldsymbol{b}, \tag{5}$$

$\mathbf{W}$是$K \times K \times D_{in} \times D_{out}$权重张量[4], $\boldsymbol{b} \in \mathbb{R}^{D_{out}}$是偏置向量和集合

$$\mathbb{\Delta}_K := \left[-\left\lfloor \frac{K}{2} \right\rfloor, \cdots, \left\lfloor \frac{K}{2} \right\rfloor\right] \times \left[-\left\lfloor \frac{K}{2} \right\rfloor, \cdots, \left\lfloor \frac{K}{2} \right\rfloor\right]$$

包含使用$K \times K$内核对图像进行卷积时出现的所有可能的偏移。

在下文中，我们将回顾自注意力如何从一维序列适应到图像。

[2] epfml.github.io/attention-cnn

[3] $\text{softmax}\left(\boldsymbol{A}_{t,:}\right)_k = \exp(\boldsymbol{A}_{t,k}) / \sum_p \exp(\boldsymbol{A}_{t,p})$

[4] 为了简化符号，我们索引张量的前两个维度，从$-\lfloor K/2 \rfloor$到$\lfloor K/2 \rfloor$。

对于图像，而不是标记，我们有查询和关键像素$\boldsymbol{q}, \boldsymbol{k} \in [W] \times [H]$。相应地，输入是维度$W \times H \times D_{in}$和每个维度的张量$\mathbf{X}$ 注意力分数将查询和关键像素关联起来。

为了保持公式与一维情况一致，我们滥用符号并通过使用二维索引向量来切片张量:如果$\boldsymbol{p} = (i, j)$，我们写$\mathbf{X}_{\boldsymbol{p},:}$和$\mathbf{A}_{\boldsymbol{p},:}$分别表示$\mathbf{X}_{i,j,:}$和$\mathbf{A}_{i,j,:,:}$。有了这种表示法，像素$\boldsymbol{q}$处的多头自注意力层输出可以表示为:

$$\text{Self-Attention}(\boldsymbol{X})_{\boldsymbol{q},:} = \sum_{\boldsymbol{k}} \text{softmax}\left(\mathbf{A}_{\boldsymbol{q},:}\right)_{\boldsymbol{k}} \mathbf{X}_{\boldsymbol{k},:}\, \boldsymbol{W}_{val} \tag{6}$$

对于多头的案子也是如此。

### 2.3 图像的位置编码

在基于transformer的架构中，有两种位置编码类型:绝对编码和相对编码(参见附录中的Table 3)。

使用绝对编码，每个像素$\boldsymbol{p}$分配一个(固定或学习)向量$\mathbf{P}_{\boldsymbol{p},:}$。我们在eq. (2)中看到的注意力分数的计算可以分解为:

$$\begin{aligned}\mathbf{A}^{\text{abs}}_{\boldsymbol{q},\boldsymbol{k}} &= (\mathbf{X}_{\boldsymbol{q},:} + \mathbf{P}_{\boldsymbol{q},:})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\mathbf{X}_{\boldsymbol{k},:} + \mathbf{P}_{\boldsymbol{k},:})^{\top} \\ &= \mathbf{X}_{\boldsymbol{q},:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{X}_{\boldsymbol{k},:}^{\top} + \mathbf{X}_{\boldsymbol{q},:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{P}_{\boldsymbol{k},:}^{\top} + \mathbf{P}_{\boldsymbol{q},:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{X}_{\boldsymbol{k},:} + \mathbf{P}_{\boldsymbol{q},:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{P}_{\boldsymbol{k},:}\end{aligned} \tag{7}$$

其中$\boldsymbol{q}$和$\boldsymbol{k}$分别对应查询像素和关键像素。

相关的位置编码由Dai et al. (2019)介绍。其主要思想是只考虑查询像素(我们计算表示的像素)和关键像素(我们参与的像素)之间的位置差异，而不是关键像素的绝对位置:

$$\mathbf{A}^{\text{rel}}_{\boldsymbol{q},\boldsymbol{k}} := \mathbf{X}_{\boldsymbol{q},:}^{\top}\boldsymbol{W}_{qry}^{\top}\boldsymbol{W}_{key}\,\mathbf{X}_{\boldsymbol{k},:} + \mathbf{X}_{\boldsymbol{q},:}^{\top}\boldsymbol{W}_{qry}^{\top}\widehat{\boldsymbol{W}}_{key}\,\boldsymbol{r}_{\boldsymbol{\delta}} + \boldsymbol{u}^{\top}\boldsymbol{W}_{key}\,\mathbf{X}_{\boldsymbol{k},:} + \boldsymbol{v}^{\top}\widehat{\boldsymbol{W}}_{key}\,\boldsymbol{r}_{\boldsymbol{\delta}} \tag{8}$$

通过这种方式，注意力得分只取决于位移$\boldsymbol{\delta} := \boldsymbol{k} - \boldsymbol{q}$。上面，可学习向量$\boldsymbol{u}$和$\boldsymbol{v}$对于每个头是唯一的，而对于每个移位$\boldsymbol{\delta}$相对位置编码$\boldsymbol{r}_{\boldsymbol{\delta}} \in \mathbb{R}^{D_p}$由所有层和头共享。此外，现在关键权重分为两种类型:$\boldsymbol{W}_{key}$属于输入，$\widehat{\boldsymbol{W}}_{key}$属于像素的相对位置。

## 3 自注意力作为卷积层

本节推导出多头自注意力层可以模拟卷积层的充分条件。我们的主要结果如下:

**Theorem 1.** *A multi-head self-attention layer with $N_h$ heads of dimension $D_h$, output dimension $D_{out}$ and a relative positional encoding of dimension $D_p \geq 3$ can express any convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ and $\min(D_h, D_{out})$ output channels.*

通过选择多头自注意力层的参数，该定理被建设性地证明，使后者类似于卷积层。在所提出的结构中，每个自注意力头的注意力分数应该关注$K \times K$内核中所有像素偏移的集合$\mathbb{\Delta}_K = \{-\lfloor K/2 \rfloor, \ldots, \lfloor K/2 \rfloor\}^2$内的不同相对偏移。确切的条件可以在引理1的陈述中找到。

然后，引理2表明，对于我们称为二次编码的相对位置编码，上述条件是满足的:

$$\boldsymbol{v}^{(h)} := -\alpha^{(h)}\left(1, -2\boldsymbol{\Delta}_1^{(h)}, -2\boldsymbol{\Delta}_2^{(h)}\right) \quad \boldsymbol{r}_{\boldsymbol{\delta}} := \left(\|\boldsymbol{\delta}\|^2, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2\right) \quad \boldsymbol{W}_{qry} = \boldsymbol{W}_{key} := \mathbf{0} \quad \widehat{\boldsymbol{W}_{key}} := \boldsymbol{I} \tag{9}$$

学习到的参数$\boldsymbol{\Delta}^{(h)} = (\boldsymbol{\Delta}_1^{(h)}, \boldsymbol{\Delta}_2^{(h)})$和$\alpha^{(h)}$分别确定每个头部的注意力中心和宽度。另一方面，$\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$是固定的，表达了查询与关键像素之间的相对偏移。

需要强调的是，上述编码并不是唯一的满足引理1的条件。事实上，在我们的实验中，神经网络学习到的相关编码也匹配引理的条件(尽管不同于二次编码)。然而，上面定义的编码在大小方面是非常高效的，因为只有$D_p = 3$维度足以编码像素的相对位置，同时也达到类似或更好的经验性能(比学习的性能)。

该定理涵盖了在eq. (17)中定义的一般卷积算子。然而，使用差分规划框架的机器学习从业者(Paszke et al., 2017; Abadi et al., 2015)可能会质疑该定理是否适用于2D卷积层的所有超参数:

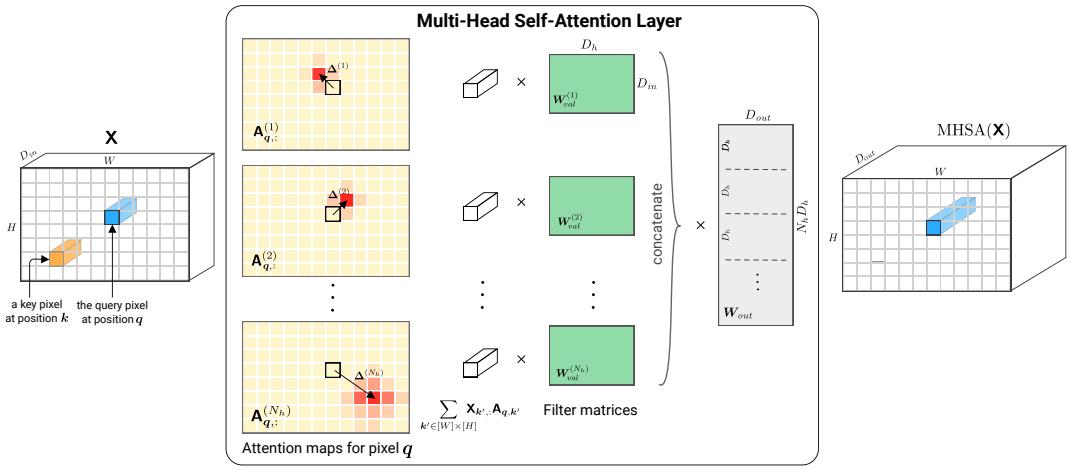


Figure 1: 应用于张量图像的多头自注意力层的插图$\mathbf{X}$。每个头$h$处理周围的像素值$\boldsymbol{\Delta}^{(h)}$并学习一个滤波器矩阵$\boldsymbol{W}_{val}^{(h)}$。展示了为位置$\boldsymbol{q}$上的查询像素计算的注意图。

- 填充:多头自注意力层默认使用\相同"的填充，而卷积层将减少图像大小$K-1$像素。缓解这些边界效果的正确方法是在输入图像的每一侧都填充$\lfloor K/2 \rfloor$个零。在这种情况下，MHSA和卷积层的裁剪输出是相同的。
- 跨步:跨步卷积可以被看作是卷积之后的固定池化操作——具有计算优化。Theorem 1是为步幅1定义的，但可以将固定的池化层添加到Self-Attention层以模拟任何步幅。
- 膨胀:多头自注意力层可以表达任何膨胀的卷积，因为每个头可以参与任何像素偏移的值，并形成(膨胀)网格模式。

**1D**外壳备注。 作用于序列的卷积层通常用于文本(Kim, 2014)，以及音频(van den Oord et al., 2016)和时间序列(Franceschi et al., 2019)的文献中。定理1可以直接扩展为表明$N_h$头的多头自注意力也可以模拟具有大小为$K = N_h$的核与$\min(D_h, D_{out})$输出通道的一维卷积层，使用维度的位置编码$D_p \geq 2$。由于我们没有根据经验测试前面的结构是否与实践中的一维自注意力行为相匹配，我们不能说它实际上学会了卷积输入序列——只有它有能力这样做。

主定理的证明

证明直接来自引理1和2如下所述:

**Lemma 1.** 考虑一个由$N_h = K^2$头部，$D_h \geq D_{out}$组成的多头自注意力层，并让$\boldsymbol{f} : [N_h] \to \mathbb{\Delta}_K$成为头部到轮班的双射映射。进一步，假设对每个*head*都成立如下表达式*:*

$$\text{softmax}(\boldsymbol{A}_{\boldsymbol{q},:}^{(h)})_{\boldsymbol{k}} = \begin{cases} 1 & \textit{if } \boldsymbol{f}(h) = \boldsymbol{q} - \boldsymbol{k} \\ 0 & \textit{otherwise}. \end{cases} \tag{10}$$

然后，对于任何具有$K \times K$内核和$D_{out}$输出通道的卷积层，都存在$\{\boldsymbol{W}_{val}^{(h)}\}_{h \in [N_h]}$这样的$\text{MHSA}(\boldsymbol{X}) = \text{Conv}(\boldsymbol{X})$每一个$\boldsymbol{X} \in \mathbb{R}^{W \times H \times D_{in}}$。

*Proof.* 我们的第一步将是重写equation (1)和equation (4)中的多头自注意力算子的表达，以便多头的效果变得更加透明:

$$\text{MHSA}(\boldsymbol{X}) = \boldsymbol{b}_{out} + \sum_{h \in [N_h]} \text{softmax}(\boldsymbol{A}^{(h)}) \boldsymbol{X} \underbrace{\boldsymbol{W}_{val}^{(h)} \boldsymbol{W}_{out}[(h-1)D_h + 1 : hD_h + 1]}_{\boldsymbol{W}^{(h)}} \tag{11}$$

注意，每个头的值矩阵$\boldsymbol{W}_{val}^{(h)} \in \mathbb{R}^{D_{in} \times D_h}$和每个块的投影矩阵$\boldsymbol{W}_{out}$的维度$D_h \times D_{out}$是学习的。假设$D_h \geq D_{out}$，我们可以将每个头的每一对矩阵替换为一个学习矩阵$\boldsymbol{W}^{(h)}$。考虑多

头自注意力的一个输出像素:

$$\mathrm{MHSA}(\boldsymbol{X})_{\boldsymbol{q},:} = \sum_{h\in[N_h]} \left( \sum_{\boldsymbol{k}} \mathrm{softmax}(\mathbf{A}^{(h)}_{\boldsymbol{q},:})_{\boldsymbol{k}} \mathbf{X}_{\boldsymbol{k},:} \right) \boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{12}$$

根据引理的条件，对于$h$ -th注意力头，在以下情况下注意力概率为1 $\boldsymbol{k} = \boldsymbol{q} - \boldsymbol{f}(h)$ 否则为零。因此，图层在像素$\boldsymbol{q}$处的输出等于

$$\mathrm{MHSA}(\mathbf{X})_{\boldsymbol{q}} = \sum_{h\in[N_h]} \mathbf{X}_{\boldsymbol{q}-\boldsymbol{f}(h),:} \boldsymbol{W}^{(h)} + \boldsymbol{b}_{out} \tag{13}$$

对于$K = \sqrt{N_h}$，上面可以被视为等效于用eq. 17表示的卷积层:在矩阵$\boldsymbol{W}^{(h)}$ $(h = [N_h])$和所有矩阵$\mathbf{W}_{k_1,k_2,:,:}$ ()之间存在一对一的映射(由map $\boldsymbol{f}$暗示) $(k_1, k_2) \in [K]^2$. □

关于$D_h$和$D_{out}$的评论。 在基于transformer的架构中，经常设置$D_h = D_{out}/N_h$，因此设置$D_h < D_{out}$。在这种情况下，$\boldsymbol{W}^{(h)}$可以看出排名$D_{out} - D_h$，这不足以用$D_{out}$通道表示每个卷积层。然而，可以看到，MHSA($\boldsymbol{X}$)的任何$D_h$ out $D_{out}$输出都可以用$D_h$输出通道表示任何卷积层的输出。为了涵盖这两种情况，在主定理的陈述中，我们断言卷积层的输出通道应该是$\min(D_h, D_{out})$。在实践中，我们建议连接维度头$D_h = D_{out}$而不是在头之间分割$D_{out}$维度，以具有精确的重新参数化和没有"未使用"通道。

**Lemma 2.** 存在一种相对的编码方案$\{\boldsymbol{r}_{\boldsymbol{\delta}} \in \mathbb{R}^{D_p}\}_{\boldsymbol{\delta}\in\mathbb{Z}^2}$ $D_p \geq 3$和参数$\boldsymbol{W}_{qry}, \boldsymbol{W}_{key}, \widehat{\boldsymbol{W}}_{key}, \boldsymbol{u}$与$D_p \leq D_k$这样，对于每个$\boldsymbol{\Delta} \in \mathbb{\Delta}_K$存在一些向量$\boldsymbol{v}$(以$\boldsymbol{\Delta}$为条件)，产生$\mathrm{softmax}(\mathbf{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} = 1$如果$\boldsymbol{k} - \boldsymbol{q} = \boldsymbol{\Delta}$和零，否则。

*Proof.* 本文通过构造表明存在$D_p = 3$维相对编码方案，产生所需的注意力概率。

由于注意力概率独立于输入张量$\mathbf{X}$，我们设置$\boldsymbol{W}_{key} = \boldsymbol{W}_{qry} = \mathbf{0}$，只留下eq. (8)的最后一项。将$\widehat{\boldsymbol{W}}_{key} \in \mathbb{R}^{D_k\times D_p}$设置为单位矩阵(具有适当的行填充)，得到结果$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}} = \boldsymbol{v}^\top \boldsymbol{r}_{\boldsymbol{\delta}}$在哪里 $\boldsymbol{\delta} := \boldsymbol{k} - \boldsymbol{q}$。上面，我们假设$D_p \leq D_k$，这样就不会丢失来自$\boldsymbol{r}_{\boldsymbol{\delta}}$的信息。

现在，假设我们可以这样写:

$$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}} = -\alpha(\|\boldsymbol{\delta} - \boldsymbol{\Delta}\|^2 + c) \tag{14}$$

为了一些恒定的$c$。在上面的表达式中，$\mathbf{A}_{\boldsymbol{q},:}$上的最大注意力分数是$-\alpha c$，而$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}}$上的最大注意力分数是$\boldsymbol{\delta} = \boldsymbol{\Delta}$。另一方面，$\alpha$系数可以用来任意缩放$\mathbf{A}_{\boldsymbol{q},\boldsymbol{\Delta}}$和其他注意力分数之间的差异。

这样，对于$\boldsymbol{\delta} = \boldsymbol{\Delta}$，我们有了

$$\begin{aligned}\lim_{\alpha\to\infty} \mathrm{softmax}(\mathbf{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} &= \lim_{\alpha\to\infty} \frac{e^{-\alpha(\|\boldsymbol{\delta}-\boldsymbol{\Delta}\|^2+c)}}{\sum_{\boldsymbol{k}'} e^{-\alpha(\|(\boldsymbol{k}-\boldsymbol{q}')-\boldsymbol{\Delta}\|^2+c)}} \\ &= \lim_{\alpha\to\infty} \frac{e^{-\alpha\|\boldsymbol{\delta}-\boldsymbol{\Delta}\|^2}}{\sum_{\boldsymbol{k}'} e^{-\alpha\|(\boldsymbol{k}-\boldsymbol{q}')-\boldsymbol{\Delta}\|^2}} = \frac{1}{1+\lim_{\alpha\to\infty}\sum_{\boldsymbol{k}'\neq\boldsymbol{k}} e^{-\alpha\|(\boldsymbol{k}-\boldsymbol{q}')-\boldsymbol{\Delta}\|^2}} = 1\end{aligned}$$

对于$\boldsymbol{\delta} \neq \boldsymbol{\Delta}$，方程变成$\lim_{\alpha\to\infty} \mathrm{softmax}(\mathbf{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} = 0$, 完全符合引理的条件。

接下来要做的是证明$\boldsymbol{v}$和$\{\boldsymbol{r}_{\boldsymbol{\delta}}\}_{\boldsymbol{\delta}\in\mathbb{Z}^2}$的存在，而eq. (14)是为它们准备的。展开方程的RHS，我们得到$-\alpha(\|\boldsymbol{\delta} - \boldsymbol{\Delta}\|^2 + c) = -\alpha(\|\boldsymbol{\delta}\|^2 + \|\boldsymbol{\Delta}\|^2 - 2\langle\boldsymbol{\delta}, \boldsymbol{\Delta}\rangle + c)$. 现在如果我们设置$\boldsymbol{v} = -\alpha\,(1, -2\boldsymbol{\Delta}_1, -2\boldsymbol{\Delta}_2)$ 而且$\boldsymbol{r}_{\boldsymbol{\delta}} = (\|\boldsymbol{\delta}\|^2, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$, 然后

$$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}} = \boldsymbol{v}^\top \boldsymbol{r}_{\boldsymbol{\delta}} = -\alpha(\|\boldsymbol{\delta}\|^2 - 2\boldsymbol{\Delta}_1\boldsymbol{\delta}_1 - 2\boldsymbol{\Delta}_2\boldsymbol{\delta}_2) = -\alpha(\|\boldsymbol{\delta}\|^2 - 2\langle\boldsymbol{\delta}, \boldsymbol{\Delta}\rangle) = -\alpha(\|\boldsymbol{\delta} - \boldsymbol{\Delta}\|^2 - \|\boldsymbol{\Delta}\|^2),$$

其中eq. (14)与$c = -\|\boldsymbol{\Delta}\|^2$匹配，证明结束。 □

评论$\alpha$的重要性。 一个像素的精确表示要求$\alpha$(或矩阵$\boldsymbol{W}_{qry}$和$\boldsymbol{W}_{key}$)是任意大的，尽管事实是，随着$\alpha$的增长，所有其他像素的注意力概率指数收敛于0。然而，实际的实现总是依赖于有限精度的算法，其中一个常数$\alpha$就足以满足我们的构造。例如，由于最小的`float32`正标量大约是$10^{-45}$，因此设置$\alpha = 46$就足以引起注意。

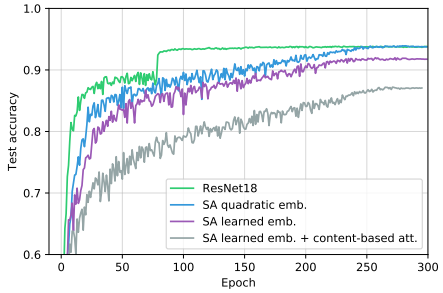


Figure 2: 在CIFAR-10上测试精度。

| Models | accuracy | # of params | # of FLOPS |
|---|---|---|---|
| ResNet18 | 0.938 | 11.2M | 1.1B |
| SA quadratic emb. | 0.938 | 12.1M | 6.2B |
| SA learned emb. | 0.918 | 12.3M | 6.2B |
| SA learned emb. + content | 0.871 | 29.5M | 15B |

Table 1: 测试CIFAR-10的精度和模型大小。SA是Self-Attention的缩写。

## 4 实验

本节的目的是验证我们的理论结果的适用性——这表明自注意力可以执行卷积——并检查在实践中，自注意力层在标准图像分类任务上训练时，是否实际上学会了像卷积层一样操作。研究了自注意力和卷积与二次和学习的相对位置编码之间的关系。发现，对于这两种情况，学习到的注意力概率倾向于尊重引理1的条件，支持我们的假设。

### 4.1 实现细节

本文研究了一个由六个多头自注意力层组成的全注意力模型。正如Bello et al. (2019)已经证明的那样将注意力特征与卷积特征相结合可以提高Cifar-100和ImageNet上的性能，我们不专注于获得最先进的性能。然而，为了验证我们的模型学习了一个有意义的分类器，我们将其与CIFAR-10数据集(Krizhevsky et al.)上的标准ResNet18 (He et al., 2015)进行比较。在所有实验中，我们对输入使用$2 \times 2$可逆下采样(Jacobsen et al., 2018)以减少图像的大小。由于注意力系数张量的大小(存储在前向期间)与输入图像的大小呈二次增长，因此完全注意力不能应用于更大的图像。输入图像的固定大小的表示被计算为最后一层表示的平均池化，并给出一个线性分类器。

我们使用PyTorch库(Paszke et al., 2017)并基于PyTorch transformer [5]实现。我们在Github [6]上发布代码，超参数列在Table 2(附录)。

关于准确性的评论。 为了验证所提出的自注意力模型表现得相当好，在Figure 6中展示了自注意力模型在小型ResNet (Table 1)上训练300个周期的CIFAR-10测试精度的变化。ResNet收敛速度更快，但我们无法确定这是否对应于架构的固有属性还是所采用的优化程序的产物。所提出的实现可以进行优化，以利用高斯注意力概率的局部性，并显著减少flop的数量。我们观察到，基于内容注意力的学习嵌入更难训练，这可能是因为它们的参数数量增加了。我们相信，性能差距是可以弥合的，以匹配ResNet的性能，但这不是本文工作的重点。

### 4.2 二次编码

作为第一步，我们的目标是验证，通过在equation (9)中引入相对位置编码，注意力层学会像卷积层一样的行为。我们在每一层训练9个注意力头，使其与ResNet架构主要使用的$3 \times 3$内核相当。每个头部的注意力中心$h$初始化为$\boldsymbol{\Delta}^{(h)} \sim \mathcal{N}(\mathbf{0}, 2\boldsymbol{I}_2)$。

Figure 3 显示了第4层头部的初始位置(不同颜色)在训练过程中如何变化。我们可以看到，经过优化后，头部聚集在图像的特定像素上，围绕查询像素形成网格。应用于图像的自注意力学习所查询像素周围的卷积滤波器的直觉得到了证实。

Figure 4 在训练结束时显示模型每一层的所有注意力头。可以看出，在前几层中，头部倾向于关注局部模式(第1层和第2层)，而更深的层(第3-6层)也通过将注意力中心定位到离查询像素位置更远的地方来关注更大的模式。我们还在附录中列出了更多头部的注意位置图($N_h = 16$)。Figure 14显示了类似于CNN的本地模式和远程依赖关系。有趣的是，注意力头并不重叠，并且似乎采取了一种使输入空间覆盖率最大化的安排。

[5] github.com/huggingface/pytorch-transformers
[6] github.com/epfml/attention-cnn

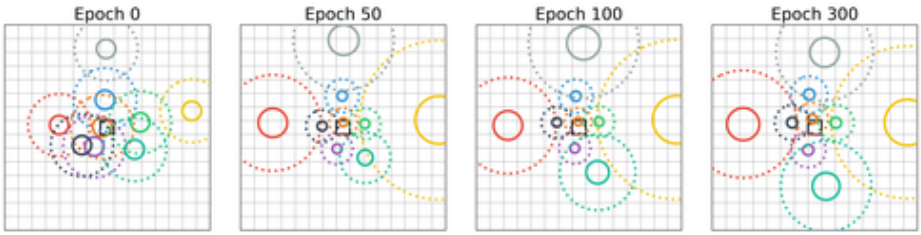


Figure 3: 在使用二次相对位置编码训练期间，第4层每个注意力头(不同颜色)的注意力中心。中间的黑色正方形是查询像素，而实心和虚线圆分别代表每个高斯的50%和90%。

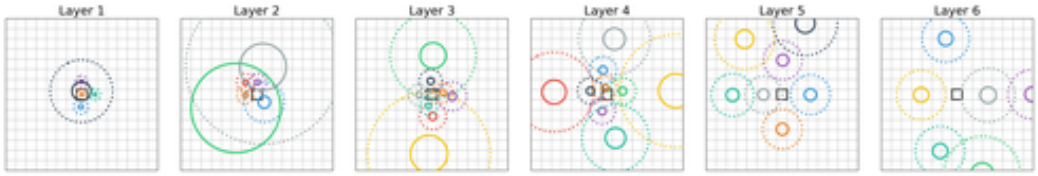


Figure 4: 使用二次位置编码的6个自注意力层的每个注意力头(不同颜色)的注意力中心。中间的黑色正方形是查询像素，而实心和虚线圆分别代表每个高斯的50%和90%。

### 4.3 学习相对位置编码

接下来，我们将研究全注意力模型在实际图像上使用的位置编码。

我们实现了(Ramachandran et al., 2019; Bello et al., 2019)使用的二维相对位置编码方案: 我们学习了每一行和每列像素偏移的$\lfloor D_p/2 \rfloor$位置编码向量。因此，位置$\boldsymbol{k}$的关键像素与位置$\boldsymbol{q}$的查询像素的相对位置编码是行移位嵌入$\boldsymbol{\delta}_1$和列移位嵌入$\boldsymbol{\delta}_2$(其中$\boldsymbol{\delta} = \boldsymbol{k} - \boldsymbol{q}$)的连接。我们在实验中选择了$D_p = D_{out} = 400$。我们与他们(未发布)的实现有以下几点不同: *(i)*我们不使用卷积干和ResNet瓶颈进行下采样，而只在输入端使用$2 \times 2$可逆下采样层(Jacobsen et al., 2018)，*(ii)*根据所学过滤器的有效数量为$\min(D_h, D_{out})$的理论，我们使用$D_h = D_{out}$而不是$D_h = D_{out}/N_h$。

首先，我们丢弃输入数据，并仅计算注意力分数作为eq的最后一项(8)。每一层每个头部的注意力概率显示在Figure 5上。该图证实了我们对前两层和第三层的部分假设:即使当从随机初始化的向量中学习位置编码方案时，某些自注意力头(如左边所示)也会学习注意单个像素，与引理1和定理1的条件密切匹配。同时，其他头部注意水平对称但非局部的模式，以及长程像素之间的相互依赖。

我们继续讨论一个更现实的设置，在该设置中，注意力分数是使用基于位置和基于内容的注意力计算的(即(Ramachandran et al., 2019)中的$q^\top k + q^\top r$)，这对应于一个成熟的独立的自注意力模型。

每个层中每个头部的注意力概率显示在Figure 6中。对一批100个测试图像的注意力概率进行平均，以概述每个头部的焦点，并消除对输入图像的依赖。我们的假设在第2层和第3层的一些头中得到了证实:即使离开去从数据中学习编码，某些自注意力头也只利用基于位置的注意力来关注与查询像素有固定偏移的不同像素，以重现卷积核的感受野。其他头部使用更多基于内容的注意力(关于非平均概率，请参阅附录中的Figures 8 to 10)，利用了自注意力相对于CNN的优势，这与我们的理论并不矛盾。在实践中，Bello et al. (2019)表明，将CNN和自注意力特征相结合的效果优于单独使用的特征。实验表明，这种组合是在优化无约束的全注意力模型时学会的。

当查询像素滑过图像时，卷积和多头自注意力之间的相似性是惊人的: 在Figure 6中可见的本地化注意力模式遵循查询像素。当将Figure 6与不同查询像素处的注意力概率进行比较时，这种特性行为就体现出来了(请参见附录Figure 7)。第2层和第3层的注意力模式不仅是局部的，而且从查询像素保持恒定的变化，类似于在图像上卷积卷积核的感受野。这种现象在我们的互动网站[7]上表现得很明显。该工具旨在探索不同图像的注意力的不同组成部

[7] epfml.github.io/attention-cnn

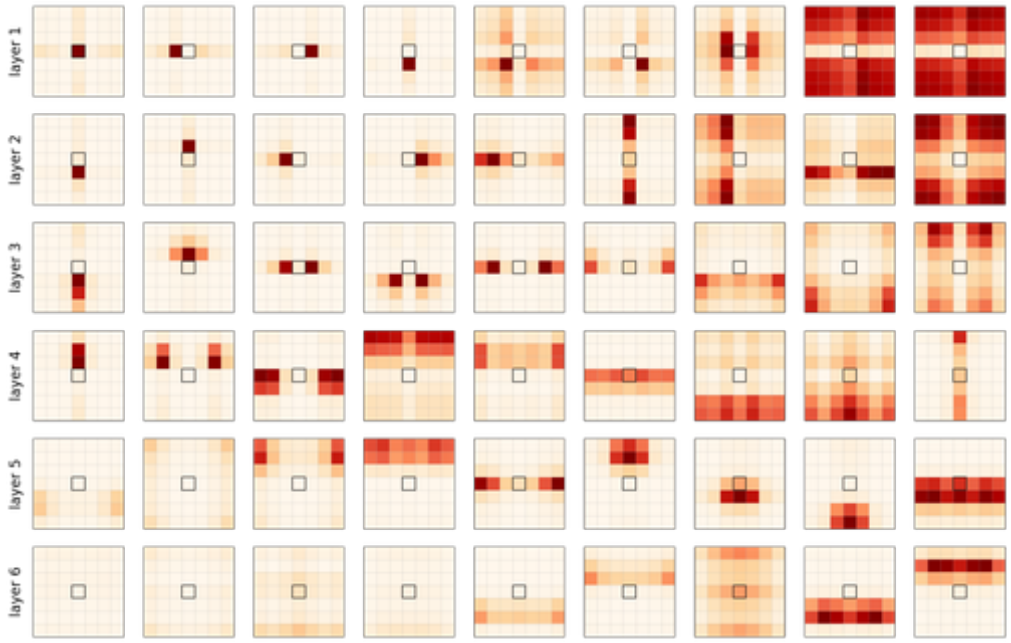


Figure 5: 在没有基于内容的注意力的情况下，使用学习到的相对位置编码在每层(行)中每个头(列)的注意力概率。中心的黑色正方形是查询像素。为了可视化，我们重新排序了头部，并在查询像素周围的7x7像素上进行了放大。

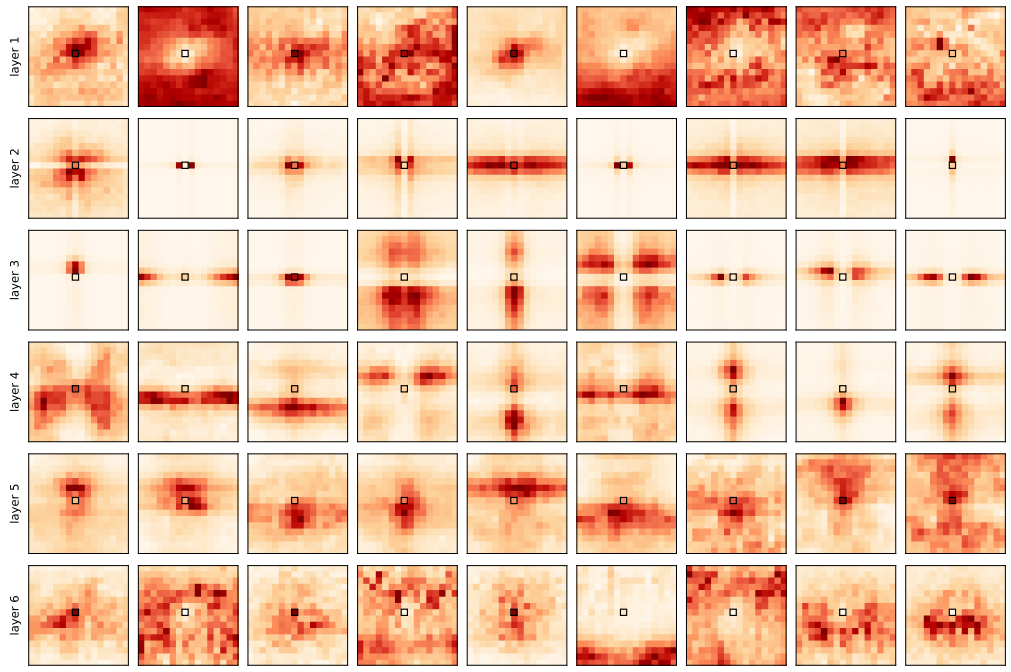


Figure 6: 使用学习到的相对位置编码和基于内容的注意力的6层(行)和9头(列)模型的注意力概率。注意力图在100张测试图像上进行平均，以显示头部行为并消除对输入内容的依赖。黑色方块是查询像素。更多示例见附录A。

分，无论是否基于内容的注意力。我们相信这是一个有用的工具，以进一步了解MHSA如何学习处理图像。

## 5 相关工作

在本节中，我们回顾了cnn和transformer之间已知的差异和相似性。

CNN网络用于文本——在单词级别(Gehring et al., 2017)或字符级别(Kim, 2014)——比transformer(或RNN)更少。transformer和卷积模型已经在自然语言处理和神经机器翻译任务上进行了广泛的实证比较。据观察，transformer与应用于文本的卷积模型(Vaswani et al., 2017)相比具有竞争优势。直到最近，Bello et al. (2019); Ramachandran et al. (2019)才在图像上使用transformer，并表明它们达到了与ResNets相似的精度。然而，它们的比较只涵盖了性能、参数数量和FLOPS，而不包括表达能力。

除了transformer和CNN的性能和计算成本比较之外，对这些架构的表现力的研究集中在它们捕获长期依赖关系的能力(Dai et al., 2019)。另一项有趣的研究表明，transformer是图灵完备的(Dehghani et al., 2018; Pérez et al., 2019)，这是一个重要的理论结果，但对从业者来说没有信息量。本文首次表明，由自注意力层表示的一类函数包含了所有卷积滤波器。

在弥合注意力和卷积之间的差距方面，最接近的工作来自Andreoli (2019)。他们将注意力和卷积纳入一个利用张量外积的统一框架中。在这个框架中，卷积的感受野由一个"基"张量$\mathbf{A} \in \mathbb{R}^{K\times K\times H\times W\times H\times W}$表示。例如，经典$K \times K$卷积核的感受野将由$\mathbf{A}_{\boldsymbol{\Delta},\boldsymbol{q},\boldsymbol{k}} = \mathbb{1}\{\boldsymbol{k} - \boldsymbol{q} = \boldsymbol{\Delta}\}$编码为$\boldsymbol{\Delta} \in \mathbb{\Delta}_K$。作者将这种基于索引的卷积与基于内容的卷积区分开来，其中$\mathbf{A}$是从输入的值计算出来的，例如，使用键/查询点积注意力。本文工作进一步推进，为注入到输入内容中的相对位置编码提供了充分条件(正如在实践中所做的那样)，以允许基于内容的卷积表示任何基于索引的卷积。通过实验进一步表明，这种行为是在实践中习得的。

## 6 结论

我们表明，应用于图像的自注意力层可以表达任何卷积层(给定足够多的头)和全注意力模型学习结合局部行为(类似于卷积)和基于输入内容的全局注意力。更一般地说，全注意力模型似乎学习了cnn的泛化，其中在学习滤波器的同时学习核模式——类似于可变形卷积(Dai et al., 2017; Zampieri, 2019)。未来工作的有趣方向包括将丰富的cnn文献中的现有见解转换回各种数据模态的transformer，包括图像、文本和时间序列。

致谢

Jean-Baptiste Cordonnier感谢瑞士数据科学中心(SDSC)资助这项工作。Andreas Loukas由瑞士国家科学基金会(项目"面向图结构数据的深度学习"，资助编号PZ00P2 179981)资助。

## REFERENCES

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

Jean-Marc Andreoli. Convolution, attention and structure embedding. *NeurIPS 2019 workshop on Graph Representation Learning, Dec 13, 2019, Vancouver, BC, Canada*, 2019.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention Augmented Convolutional Networks. *arXiv:1904.09925 [cs]*, April 2019.

Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G. Carbonell, Quoc V. Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context. *CoRR*, abs/1901.02860, 2019.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *CoRR*, abs/1807.03819, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

Jean-Yves Franceschi, Aymeric Dieuleveut, and Martin Jaggi. Unsupervised scalable representation learning for multivariate time series. In *NeurIPS 2019*, 2019.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8): 1735–1780, 1997.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7132–7141, 2018.

Jörn-Henrik Jacobsen, Arnold W.M. Smeulders, and Edouard Oyallon. i-revnet: Deep invertible networks. In *International Conference on Learning Representations*, 2018.

Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

Jorge Pérez, Javier Marinkovic, and Pablo Barceló. On the turing completeness of modern neural network architectures. *CoRR*, abs/1901.03429, 2019.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. *CoRR*, abs/1906.05909, 2019.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 7794–7803, 2018.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019.

Luca Zampieri. Geometric deep learning for volumetric computational fluid dynamics. pp. 67, 2019.

# 附录

## A 更多基于内容的注意力的例子

本文提出更多由自注意力模型计算的注意力概率的例子。Figure 7显示了与Figure 6不同的查询像素上的平均注意力。Figures 8 to 10显示关注单个图像。

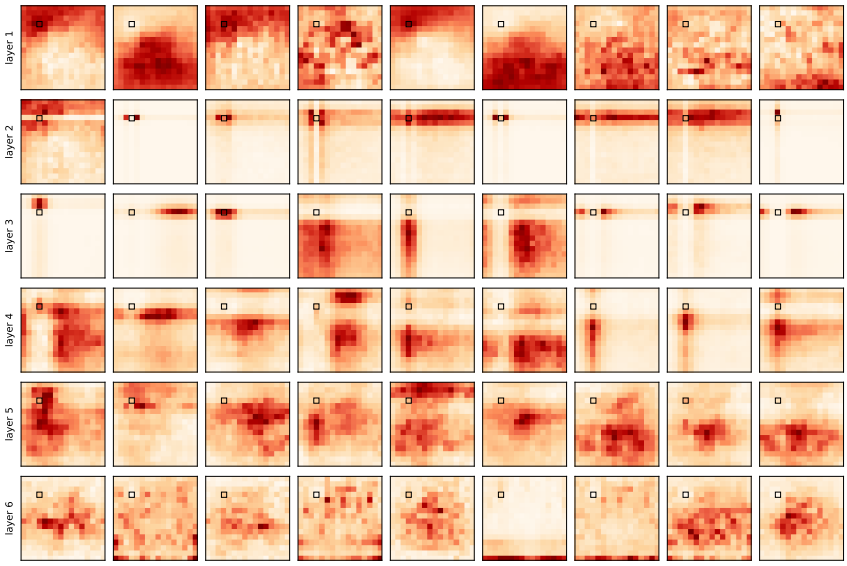


Figure 7: 使用学习到的相对位置编码和内容-内容注意力的6层(行)和9头(列)模型的注意力概率。我们给出了100幅测试图像的平均值。黑色方块是查询像素。

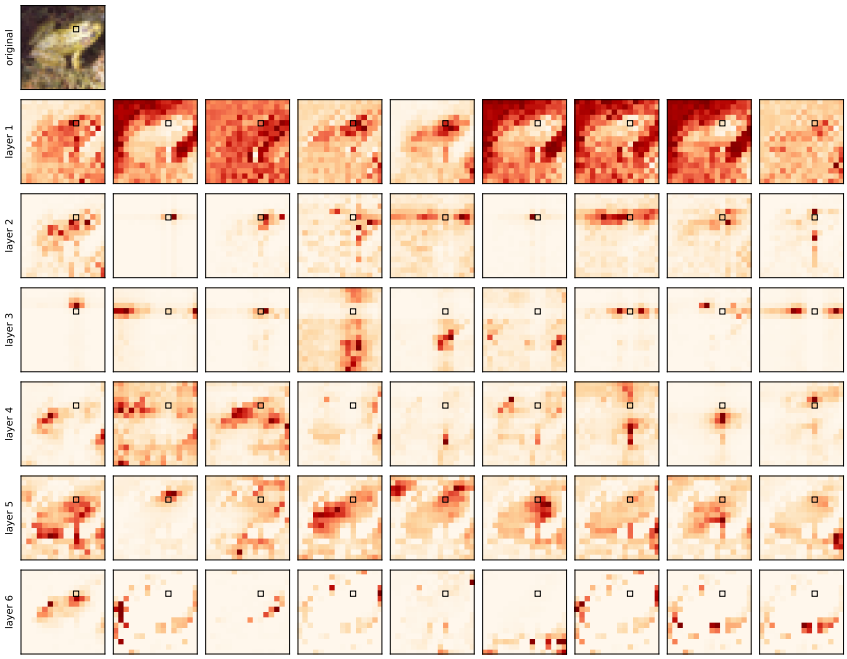


Figure 8: 使用学习到的相对位置编码和基于内容的注意力的6层(行)和9头(列)模型的注意力概率。查询像素(黑色正方形)在青蛙的头部上。

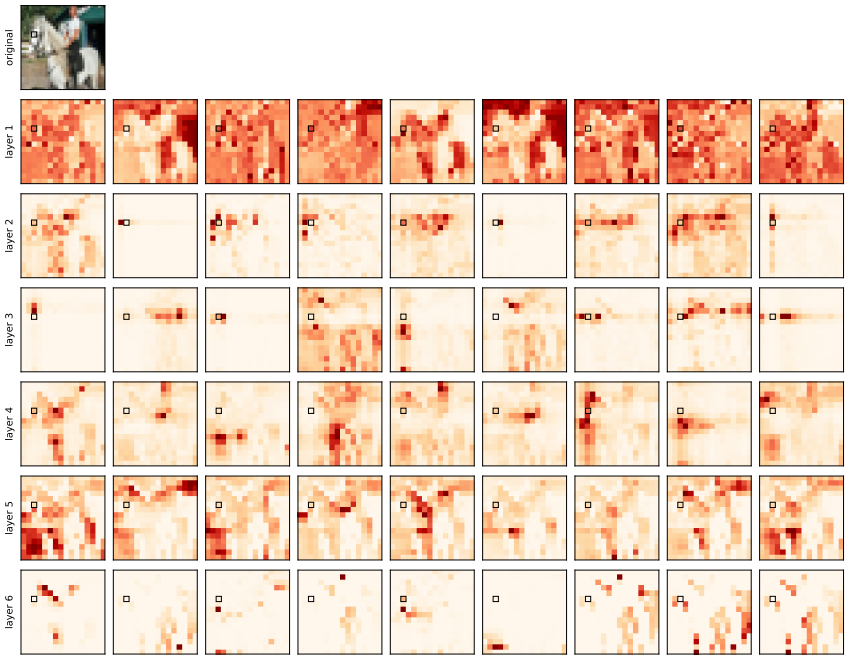


Figure 9: 使用学习到的相对位置编码和基于内容的注意力的6层(行)和9头(列)模型的注意力概率。查询像素(黑色正方形)位于马头上。

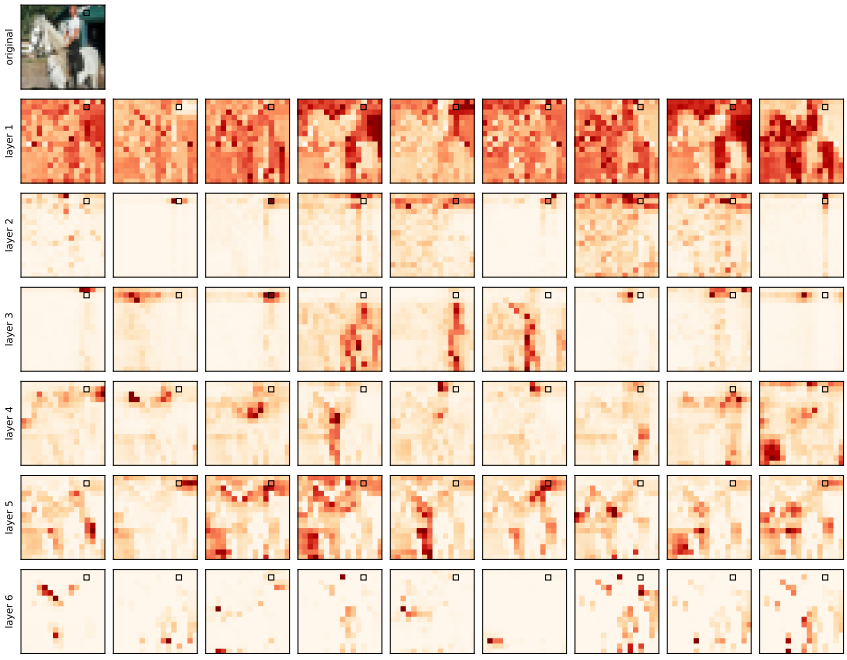


Figure 10: 使用学习到的相对位置编码和基于内容的注意力的6层(行)和9头(列)模型的注意力概率。查询像素(黑色正方形)在背景中的建筑物上。

## B 实验中使用的超参数

| Hyper-parameters | |
|---|---|
| number of layers | 6 |
| number of heads | 9 |
| hidden dimension | 400 |
| intermediate dimension | 512 |
| invertible pooling width | 2 |
| dropout probability | 0.1 |
| layer normalization epsilon | $10^{-12}$ |
| number of epochs | 300 |
| batch size | 100 |
| learning rate | 0.1 |
| weight decay | 0.0001 |
| momentum | 0.9 |
| cosine decay | ✓ |
| linear warm up ratio | 0.05 |

Table 2: 自注意力网络参数

## C 位置编码引用

| Model | type of positional encoding | | | relative |
|---|---|---|---|---|
| | sinusoids | learned | quadratic | |
| Vaswani et al. (2017) | ✓ | | | |
| Radford et al. (2018) | | ✓ | | |
| Devlin et al. (2018) | | ✓ | | |
| Dai et al. (2019) | ✓ | | | ✓ |
| Yang et al. (2019) | ✓ | | | ✓ |
| Bello et al. (2019) | | ✓ | | ✓ |
| Ramachandran et al. (2019) | | ✓ | | ✓ |
| Our work | | ✓ | ✓ | ✓ |

Table 3: 应用于文本(顶部)和图像(底部)的transformer模型使用的位置编码类型。当尝试了多种编码类型时，我们报告了作者建议的一种。

## D 广义的LEMMA 1

本文提出Lemma 1的泛化，用一个更温和的假设取代了(对单个像素)严格注意的必要性:注意概率应该跨越网格感受野。这个引理的条件仍然由Lemma 2满足，因此接下来是Theorem 1。

**Lemma 3.** 考虑一个由$N_h \geq K^2$头，$D_h \geq D_{out}$和$\omega : [H] \times [W] \to [HW]$组成的多头自注意力层，并让作为像素索引。然后，对于任何具有$K \times K$内核和$D_{out}$输出通道的卷积层，都存在$\{\boldsymbol{W}_{val}^{(h)}\}_{h\in[N_h]}$和$\boldsymbol{W}_{out}$以便$\mathrm{MHSA}(\mathbf{X}) = \mathrm{Conv}(\mathbf{X})$ 对于每一个$\mathbf{X} \in \mathbb{R}^{W\times H\times D_{in}}$ 当且仅当对于所有$\boldsymbol{q} \in [H] \times [W]$，[8]

$$\mathrm{span}(\{\boldsymbol{e}_{\omega(\boldsymbol{q}+\boldsymbol{\Delta})} \in \mathbb{R}^{HW} : \boldsymbol{\Delta} \in \mathbb{\Delta}_K\}) \subseteq \mathrm{span}(\{\mathrm{vect}(\mathrm{softmax}(\mathbf{A}^{(h)}_{\boldsymbol{q},:})) : h \in [N_h]\})\,.$$

[8]向量化运算符$\mathrm{vect}(\cdot)$将矩阵展平为向量

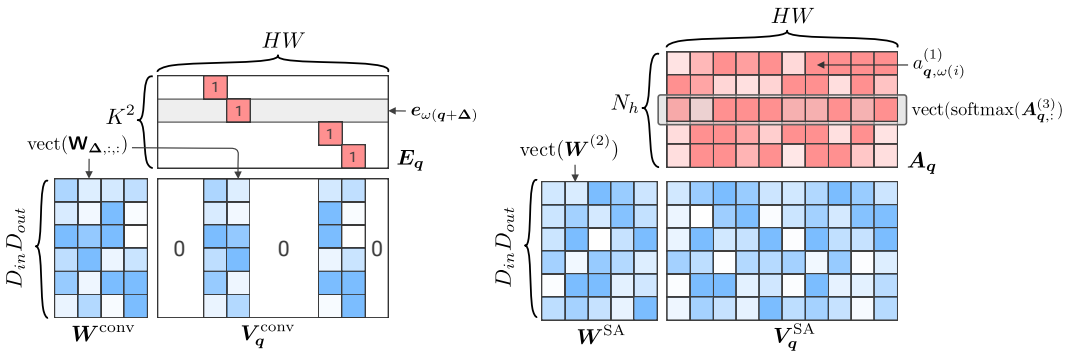


Figure 11: 向量化权重矩阵$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}}$和$\boldsymbol{V}_{\boldsymbol{q}}^{\text{SA}}$的因式分解用于计算维度为$H \times W$的输入图像在位置$\boldsymbol{q}$的输出。左边:内核的卷积$2 \times 2$，右边:带有$N_h = 5$头部的self-attention。$D_{in} = 2$, $D_{out} = 3$。

*Proof.* 我们的第一步将是重写equation (1)和equation (4)中的多头自注意力算子的表达，以便多头的效果变得更加透明:

$$\text{MHSA}(\mathbf{X}) = \boldsymbol{b}_{out} + \sum_{h\in[N_h]} \text{softmax}(\mathbf{A}^{(h)})\mathbf{X}\underbrace{\boldsymbol{W}_{val}^{(h)}\boldsymbol{W}_{out}[(h-1)D_h+1:hD_h+1]}_{\boldsymbol{W}^{(h)}} \tag{15}$$

注意，每个头的值矩阵$\boldsymbol{W}_{val}^{(h)} \in \mathbb{R}^{D_{in}\times D_h}$和每个块的投影矩阵$\boldsymbol{W}_{out}$的维度$D_h \times D_{out}$是学习的。假设$D_h \geq D_{out}$，我们可以将每个头的每一对矩阵替换为一个学习矩阵$\boldsymbol{W}^{(h)}$。考虑多头自注意力的一个输出像素，为简单起见，去掉偏置项:

$$\text{MHSA}(\mathbf{X})_{\boldsymbol{q},:} = \sum_{h\in[N_h]}\Big(\sum_{\boldsymbol{k}} a_{\boldsymbol{q},\boldsymbol{k}}^{(h)}\mathbf{X}_{\boldsymbol{k},:}\Big)\boldsymbol{W}^{(h)} = \sum_{\boldsymbol{k}}\mathbf{X}_{\boldsymbol{k},:}\underbrace{\Big(\sum_{h\in[N_h]} a_{\boldsymbol{q},\boldsymbol{k}}^{(h)}\boldsymbol{W}^{(h)}\Big)}_{\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{SA}}\in\mathbb{R}^{D_{in}\times D_{out}}}, \tag{16}$$

用$a_{\boldsymbol{q},\boldsymbol{k}}^{(h)} = \text{softmax}(\mathbf{A}_{\boldsymbol{q},:}^{(h)})_{\boldsymbol{k}}$。我们用同样的方式重写像素$\boldsymbol{q}$处卷积的输出:

$$\text{Conv}(\mathbf{X})_{\boldsymbol{q},:} = \sum_{\boldsymbol{\Delta}\in\mathbb{\Delta}_K}\mathbf{X}_{\boldsymbol{q}+\boldsymbol{\Delta},:}\mathbf{W}_{\boldsymbol{\Delta},:,:} = \sum_{\boldsymbol{k}\in[H]\times[W]}\mathbf{X}_{\boldsymbol{k},:}\underbrace{\mathbb{1}_{\{\boldsymbol{k}-\boldsymbol{q}\in\mathbb{\Delta}_K\}}\mathbf{W}_{\boldsymbol{k}-\boldsymbol{q},:,:}}_{\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{conv}}\in\mathbb{R}^{D_{in}\times D_{out}}}. \tag{17}$$

方程(16)和(17)之间的等式对于任何输入$\mathbf{X}$当且仅当每个键/查询像素对的线性变换相等，即$\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{conv}} = \boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{SA}}\ \forall \boldsymbol{q},\boldsymbol{k}$。我们将权重矩阵向量化为维度$D_{in}D_{out} \times HW$的矩阵$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}} := [\text{vect}(\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{conv}})]_{\boldsymbol{k}\in[H]\times[W]}$和$\boldsymbol{V}_{\boldsymbol{q}}^{\text{SA}} := [\text{vect}(\boldsymbol{W}_{\boldsymbol{q},\boldsymbol{k}}^{\text{SA}})]_{\boldsymbol{k}\in[H]\times[W]}$。因此，为了表明$\text{Conv}(\mathbf{X}) = \text{MHSA}(\mathbf{X})$为所有人$\mathbf{X}$，我们必须表明$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}} = \boldsymbol{V}_{\boldsymbol{q}}^{\text{SA}}$为所有人$\boldsymbol{q}$。

矩阵$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}}$有一个限制的支持:只有在像素$\boldsymbol{q}$的感受野中与像素移位$\boldsymbol{\Delta} \in \mathbb{\Delta}_K$相关联的列可以是非零的。这导致了因子分解$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}} = \boldsymbol{W}^{\text{conv}}\boldsymbol{E}_{\boldsymbol{q}}$，如图11所示，其中$\boldsymbol{W}^{\text{conv}} \in \mathbb{R}^{D_{in}D_{out}\times K^2}$和$\boldsymbol{E}_{\boldsymbol{q}} \in \mathbb{R}^{K^2\times HW}$。给定转换顺序$\boldsymbol{\Delta} \in \mathbb{\Delta}_K$由$j$索引，设置$(\boldsymbol{W}^{\text{conv}})_{:,j} = \text{vect}(\mathbf{W}_{\boldsymbol{\Delta},:,:})$和$(\boldsymbol{E}_{\boldsymbol{q}})_{j,:} = \boldsymbol{e}_{\omega(\boldsymbol{q}+\boldsymbol{\Delta})}$。另一方面，我们分解$\boldsymbol{V}_{\boldsymbol{q}}^{\text{SA}} = \boldsymbol{W}^{\text{SA}}\boldsymbol{A}_{\boldsymbol{q}}$有$(\boldsymbol{W}^{\text{SA}})_{:,h} = \text{vect}(\boldsymbol{W}^{(h)})$而且$(\boldsymbol{A}_{\boldsymbol{q}})_{h,i} = a_{\boldsymbol{q},\omega(i)}^{(h)}$。

结论是$\text{row}(\boldsymbol{E}_{\boldsymbol{q}}) \subseteq \text{row}(\boldsymbol{A}_{\boldsymbol{q}})$是一个充要条件为了$\boldsymbol{W}^{\text{SA}}$的存在，使任何$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}} = \boldsymbol{W}^{\text{conv}}\boldsymbol{E}_{\boldsymbol{q}}$都可以写成$\boldsymbol{W}^{\text{SA}}\boldsymbol{A}_{\boldsymbol{q}}$。

足够了。假设$\text{row}(\boldsymbol{E}_{\boldsymbol{q}}) \subseteq \text{row}(\boldsymbol{A}_{\boldsymbol{q}})$，存在$\Phi \in \mathbb{R}^{K^2\times N_h}$这样$\boldsymbol{E}_{\boldsymbol{q}} = \Phi\boldsymbol{A}_{\boldsymbol{q}}$和一个有效的分解是$\boldsymbol{W}^{\text{SA}} = \boldsymbol{W}^{\text{conv}}\Phi$，它给出$\boldsymbol{W}^{\text{SA}}\boldsymbol{A}_{\boldsymbol{q}} = \boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}}$。

必要的。假设存在$\boldsymbol{x} \in \mathbb{R}^{HW}$这样的$\boldsymbol{x} \in \text{row}(\boldsymbol{E}_{\boldsymbol{q}})$和$\boldsymbol{x} \notin \text{row}(\boldsymbol{A}_{\boldsymbol{q}})$并将$\boldsymbol{x}^\top$设置为$\boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}}$的一行。然后，$\boldsymbol{W}^{\text{SA}}\boldsymbol{A}_{\boldsymbol{q}} \neq \boldsymbol{V}_{\boldsymbol{q}}^{\text{conv}}$对于任何$\boldsymbol{W}^{\text{SA}}$和没有可能的分解。

□

## E 广义二次位置编码

我们注意到二次位置编码(Section 3)中的注意力概率与具有有界支持度的各向同性双变量高斯分布相似:

$$\text{softmax}(\mathbf{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} = \frac{e^{-\alpha\|(\boldsymbol{k}-\boldsymbol{q})-\boldsymbol{\Delta}\|^2}}{\sum_{\boldsymbol{k}'\in[W]\times[H]} e^{-\alpha\|(\boldsymbol{k}'-\boldsymbol{q})-\boldsymbol{\Delta}\|^2}}\,. \tag{18}$$

在此观察的基础上，进一步将注意力机制扩展到像素位置的非各向同性高斯分布。每个头被一个注意力中心$\boldsymbol{\Delta}$和一个协方差矩阵$\boldsymbol{\Sigma}$参数化，以获得以下注意力分数，

$$\mathbf{A}_{\boldsymbol{q},\boldsymbol{k}} = -\frac{1}{2}(\boldsymbol{\delta}-\boldsymbol{\Delta})^\top\boldsymbol{\Sigma}^{-1}(\boldsymbol{\delta}-\boldsymbol{\Delta}) = -\frac{1}{2}\boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\delta} + \boldsymbol{\delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta} - \frac{1}{2}\boldsymbol{\Delta}^\top\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta}\,, \tag{19}$$

在那里，再一次$\boldsymbol{\delta} = \boldsymbol{k} - \boldsymbol{q}$。最后一项可以被舍弃，因为softmax是移位不变的，我们将注意力系数重写为头部目标向量$\boldsymbol{v}$和相对位置编码$\boldsymbol{r}_{\boldsymbol{\delta}}$之间的点积(由像素移位的一阶和二阶组合$\boldsymbol{\delta}$):

$$\boldsymbol{v} = \frac{1}{2}(2(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta})_1, 2(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Delta})_2, -\boldsymbol{\Sigma}^{-1}_{1,1}, -\boldsymbol{\Sigma}^{-1}_{2,2}, -2\cdot\boldsymbol{\Sigma}^{-1}_{1,2})^\top \text{ and } \boldsymbol{r}_{\boldsymbol{\delta}} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2, \boldsymbol{\delta}_1^2, \boldsymbol{\delta}_2^2, \boldsymbol{\delta}_1\boldsymbol{\delta}_2)^\top\,.$$

评估。 我们使用这种广义二次相对位置编码来训练我们的模型。我们很想知道，使用上述编码，自注意力模型是否会学习关注非各向同性的像素组——从而在cnn中形成未见过的模式。每个头用$\boldsymbol{\Delta}\in\mathbb{R}^2$和$\boldsymbol{\Sigma}^{-1/2}\in\mathbb{R}^{2\times2}$参数化，以确保协方差矩阵保持半正定。我们将注意力中心初始化为$\boldsymbol{\Delta}^{(h)}\sim\mathcal{N}(\mathbf{0}, 2\boldsymbol{I}_2)$和$\boldsymbol{\Sigma}^{-1/2} = \boldsymbol{I}_2 + \mathcal{N}(\mathbf{0}, 0.01\boldsymbol{I}_2)$，以便初始注意力概率接近各向同性高斯。Figure 12表明该网络确实学习了非各向同性的注意力概率模式，特别是在高层。然而，我们没有获得任何性能改进的事实似乎表明，注意力非各向同性在实践中并不是特别有用——二次位置编码就足够了。

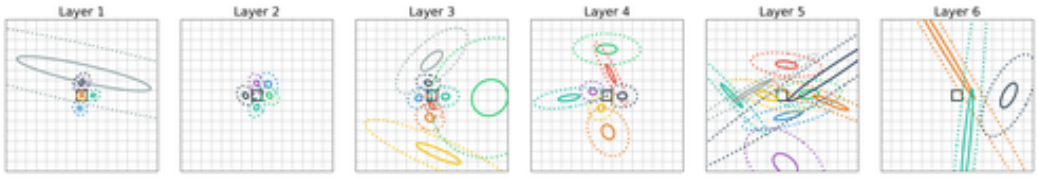


Figure 12: 使用非各向同性高斯参数化的6个自注意力层的每个注意力头(不同颜色)的注意力中心。中间的黑色正方形是查询像素，而实心和虚线圆分别代表每个高斯的50%和90%。

修剪退化的头部。 一些非各向同性的注意力头会关注“非直观”的像素块: 要么关注一个非常细的像素条纹，当$\boldsymbol{\Sigma}^{-1}$几乎是单一的，或统一关注所有像素，当$\boldsymbol{\Sigma}^{-1}$接近$\mathbf{0}$(即持续的注意力分数)。我们问自己，这样的注意力模式确实对模型有用吗?还是这些头部已经退化和未被使用? 为了找到答案，我们修剪了所有最大特征值小于$10^{-5}$或条件数(最大和最小特征值的比率)大于$10^5$的头。具体来说，在我们的模型中，每个模型有6层和9个头，我们从第一层到最后一层修剪了$[2, 4, 1, 2, 6, 0]$个头。这意味着这些层不能再表达$3\times3$内核。如fig. 2上黄色所示，这种消融最初会影响性能，可能是由于偏差，但在使用较小的学习率(除以10)进行几个epoch的继续训练后，精度恢复其未修剪的值。因此，在不牺牲性能的情况下，我们将参数的大小和flop的数量减少了四分之一。

## F 增加正面的数量

为了完整起见，我们还测试了将我们架构的头部数量从9个增加到16个。

与图4类似，我们看到网络区分了两种主要类型的注意模式。局部化的头部(即关注几乎单个像素的头部)在前几层中出现得更频繁。自注意力层使用这些头的行为方式与卷积层类似。较少局部注意力的头部在更高的层中变得更加常见。

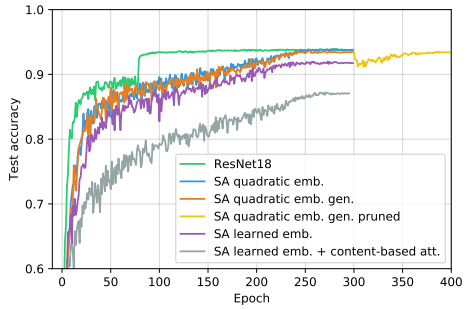


Figure 13: CIFAR-10上测试精度的演化。剪枝模型(黄色)是对非各向同性模型(橙色)的继续训练。

| Models | accuracy | # of params | # of FLOPS |
|---|---|---|---|
| ResNet18 | 0.938 | 11.2M | 1.1B |
| SA quadratic emb. | 0.938 | 12.1M | 6.2B |
| SA quadratic emb. gen. | 0.934 | 12.1M | 6.2B |
| SA quadratic emb. gen. pruned | 0.934 | 9.7M | 4.9B |
| SA learned emb. | 0.918 | 12.3M | 6.2B |
| SA learned emb. + content | 0.871 | 29.5M | 15B |

Table 4: 每个模型在CIFAR-10上的参数数量和精度。SA是Self-Attention的缩写。

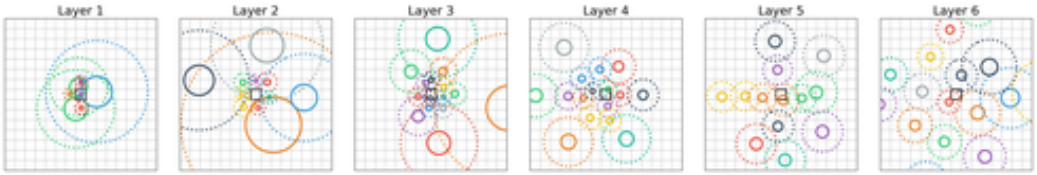


Figure 14: 使用二次位置编码的6个自注意力层的16个注意力头(不同颜色)的注意力中心。中间的黑色正方形是查询像素，而实心和虚线圆分别代表每个高斯的50%和90%。