自注意力真的是Transformer的必杀技吗？MSRA否认三连，并反手给你扔来一个sMLPNet

原创 CV开发者都爱看的 极市平台 2021-09-15 22:00:00 手机阅读

收录于话题
#Transformer

45个

作者丨happy
编辑丨极市平台

极市导读

本文构建了一种Attention-free、基于MLP的sMLPNet，主要将MLP模块中的token-mixing替换为稀疏MLP(sparse MLP, sMLP)模块，sMLPNet仅需24M参数即可在ImageNet数据及上取得81.9%top1精度，优于同等大小的CNN与Vision Transformer。>>加入极市CV技术交流群，走在计算机视觉的最前沿

Sparse MLP for Image Recognition: Is Self-Attention Really Necessary?

Chuanxin Tang,[*1] Yucheng Zhao,[*†2] Guangting Wang,[*†2] Chong Luo,[1]
Wenxuan Xie,[1] Wenjun Zeng,[1]
[1]Microsoft Research Asia, Beijing, China
[2]University of Science and Technology of China, Hefei, China
{chutan, cluo, wenxie, wezeng}@microsoft.com,{lnc, flylight}@mail.ustc.edu.cn

论文链接：https://arxiv.org/pdf/2109.05422.pdf

前言

自今年5月份MLP-Mixer横冲出世以来，4个月的时间里出来了20+MLP相关的论文，从MLP-Mixer、ResMLP、gMLP到A2-MLP、CCS，再到ViP，MLP相关的结构好像一下被探索到头，自ViP之后的MLP相关的paper大多都借鉴了ViP的思想，此外再小心翼翼的加点不一样的小改进。与此同时，优异的Vision Transformer也在尝试将其内在的自注意力替换为MLP，比如**Swin Transformer的变种SwinMLP 。**

本文所提出的sMLPNet的一个新颖点是：token-mixing部分同时进行了局部与全局依赖建模。局部依赖建模比较容易想到，DWConv即可；全局建模用则从CSWin那里借鉴了一些idea。两者一组合就取得了非常👍的指标。

但是，sMLPNet中引入了BN与DWConv，因此就不能算作是纯MLP架构，可能这也是之前MLP类模型非常小心翼翼的原因吧，生怕影响“出身”（狗头）。

下面正式介绍本篇论文～

Abstract

本文对Transformer中的核心模块自注意力模块进行探索：它是否是Transformer在图像识别任务中取得优异性能的关键?

我们构建了一种Attention-free的、基于MLP的sMLPNet。具体来讲，**我们将MLP模块中的token-mixing替换为稀疏MLP(sparse MLP, sMLP)模块** 。对于2D图像tokens，sMLP沿轴向进行1DMLP，同时在行/列方向上分别进行参数共享。受益于稀疏连接与参数共享，sMLP模块可以大幅降低模型参数量与计算量，进而避免了干扰MLP类模型性能的“过拟合”问题。

sMLPNet仅需24M参数即可在ImageNet数据及上取得81.9%top1精度，优于同等大小的CNN与Vision Transformer；当参数量扩大到66M，sMLPNet取得了83.4%top1精度，具有与Swin Transformer相当精度。sMLPNet的成功说明：自注意力机制并非Transformer取得优异性能的关键所在。

Method

在这篇文章中，我们旨在回答“**有没有可能设计一种无自注意力模块的高性能图像分类模型** ”。我们希望保留CNN网络的某些重要设计思想，同时引入一些受Transformer启发的新成分。因此，我们尝试了如下设计指导思想：

- 采用与ViT、MLP-Mixer以及Swin Transformer类似的架构以确保公平比较；

- 显式地为网络注入局部偏置；

- 探索无自注意力模块的全局依赖；

- 以金字塔架构执行多阶段处理。

Overall Architecture

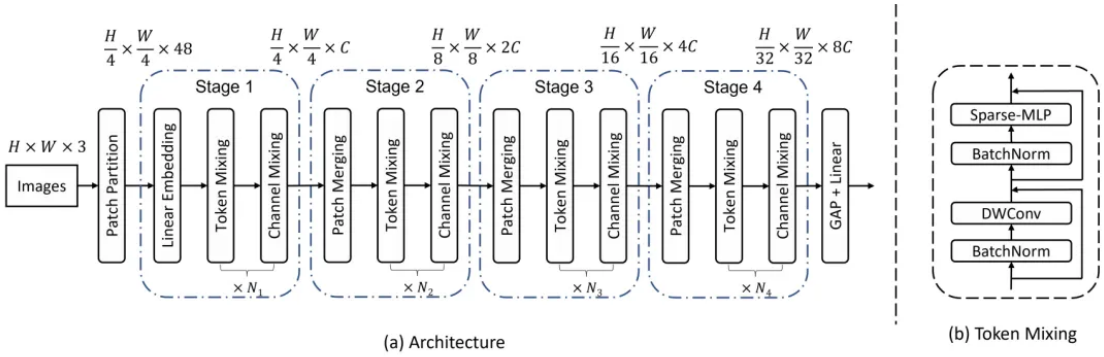

Figure 2: (a) The overall multi-stage architecture of the sMLPNet; (b) The token mixing module.

上图给出了本文所设计网络架构示意图，类似于ViT、MLP-Mixer以及Swin Transformer，它以RGB图像作为输入，并通过块拆分模块拆分为非重叠块。在网络的第一部分，块尺寸为4×4，每个块reshape为48维向量，然后通过线性层映射为C维嵌入。此时，特征表示为$H/4 \times W/4 \times C$。注：MLP-Mixer中的块尺寸为16×16，故sMLPNet的**tokens数量是MLP-Mixer的16倍** 。

整个网络包含四个阶段，除第一个阶段以线性嵌入层作为起始外，其他阶段以块合并层作为起始，**块合并层起降低空间分辨率提升通道数的作用。** 将所得到的的token送入到后续的token-mixing模块(见上图b)与channel-mixing模块。

从上面图示可以看到：**token-mixing通过采用深度卷积引入了局部偏置信息 。**与此同时，我们还尝试利用所提sMLP建模全局依赖。相比原始MLP，**sMLP的权值共享与稀疏连接机制使其更难于过拟合，同时可以大幅降低计算量 。**

而channel-mixing则采用了常规的方式，即FFN+GeLU。因为比较简单，故略。

Sparse MLP

我们设计了一种稀疏MLP以消除原始MLP的两个主要缺陷：

- 我们希望降低参数量以避免过拟合，尤其是当模型在ImageNet训练时；

- 我们希望降低计算量以促进金字塔架构的多阶段处理，尤其是当模型的输入tokens数量较大时。

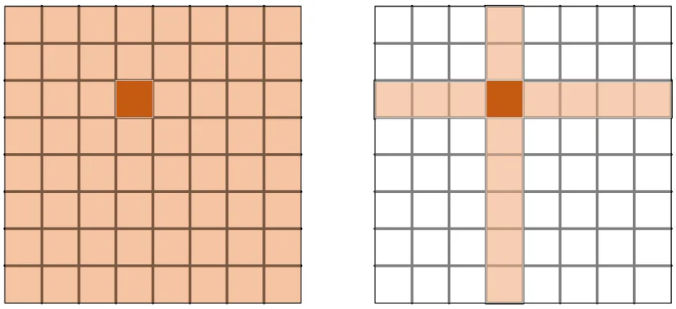

Figure 1: The proposed sparse MLP reduces the computational complexity of MLP by sparse connection and weight sharing. In MLP (a), the token in dark orange interacts with all the other tokens in a single MLP layer. In contrast, in one sMLP layer (b), the dark-orange token only interacts with horizontal and vertical tokens marked in light orange. The interaction with all the other white tokens can be achieved when sMLP is executed twice.

上图给出了原始MLP与本文所提sMLP示意图，也就是说：在sMLP中，每个token仅与同行/列的token产生交互关系。下图给出了sMLP的实现架构示意图，可以看到：**sMLP包含三个分支，除identity分支外的其他两个分别用于水平/垂直方向token交互 。**

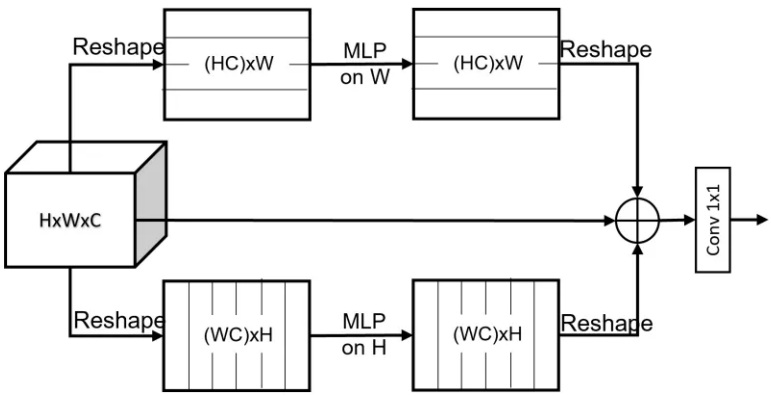

Figure 3: Structure of the proposed sMLP block. It consists of three branches: two of them are responsible for mixing information along horizontal and vertical directions respectively and the other path is the identity mapping. The output of the three branches are concat and processed by a point-wise conv to obtain the final output.

假设输入token表示为$X^{in} = R^{H \times W \times C}$，在水平交互分支上，输入张量先reshape为$HC \times W$，然后通过线性层进行信息聚合；以类似的方式进行垂直交互；最后将三个分支的结果融合。上述过程可以表示如下：

$$X^{out} = FC(concat(X_H, X_W, X))$$

Algorithm 1: Pseudocode of sMLP (PyTorch-like)

Algorithm 1: Pseudocode of sMLP (PyTorch-like)

Input: x # input tensor of shape (H, W, C)
Output: x # output tensor of shape (H, W, C)

```
proj_h = nn.Linear(H,H).
proj_w = nn.Linear(W,W).
fuse = nn.Linear(3C,C).

def sparse_mlp(x):

  x_h = self.proj_h(x.permute(2,1,0)).permute(2,1,0)
  x_w = self.proj_w(x.permute(0,2,1)).permute(0,2,1)
  x = torch.concat([x_h,x_w,x],dim=2)
  x = self.fuse(x)
  return x
```

Model Configurations

我们构建了三个不同大小的模型：sMLPNet-T、sMLPNet-S以及sMLPNet-B以对标Swin-T、Swin-S以及Swin-B。注：FFN中的扩展参数$\alpha = 3$。不同模型的超参信息如下：

- sMLPNet-T: C=80, 层数=[2;8;14;2];
- sMLPNet-S: C=96, 层数=[2;10;24;2];
- sMLPNet-B: C=112, 层数=[2;10;24;2]

Experiments

Model	Params(M)	FLOPs(B)	Top-1(%)
	ConvNets		
ResNet-50	25.6	4.1	76.2
ResNet-152	60.2	11.5	78.3
RegNetY-4GF	21	4.0	80.0
RegNetY-8GF	39.2	8.0	81.7
RegNetY-16GF	83.6	15.9	82.9
	Transformers		
ViT-B/16	86.4	17.6	79.7
DeiT-S/16	22	4.6	79.8
DeiT-B/16	86.4	17.6	81.8
DeepVit-S	27	6.2	81.4
DeepVit-L	55	12.5	82.2
Swin-T	29	4.5	81.3
Swin-S	50	8.7	83.0
Swin-B	88	15.4	83.4
	MLP-like		
Mixer-B/16	59	12.7	76.4
Mixer-L/16	207	44.8	71.8
ResMLP-12	15	3.0	76.6
ResMLP-24	30	6.0	79.4
ResMLP-36	45	8.9	79.7
gMLP-S	20	4.5	79.4

gMLP-B	73	15.8	81.6
sMLPNet-T* (ours)	19.2	4.0	81.3
sMLPNet-T (ours)	24.1	5.0	81.9
sMLPNet-S (ours)	48.6	10.3	83.1
sMLPNet-B (ours)	65.9	14.0	83.4

Table 6: Comparing the proposed sMLPNet with state-of-the-art vision models. The default expansion parameter in the FFN of sMLPNet is $\alpha = 3$. sMLPNet-T* uses $\alpha = 2$. All models are trained on ImageNet-1K benchmark without extra data. The resolution of the input image is 224×224 for all the models.

上图给出了所提方案与其他CNN、Transformer以及MLP在ImageNet数据集上的参数量、FLOPs以性能对比，从中可以看到：

- 相比经典ResNet，RegNetY具有非常大的优势；
- 除ViT之外的Transformer模型具有与RegNetY相当的性能；
- 大多MLP模型仅取得了与ResNet相当的性能，其中Mixer-L出现了严重的过拟合问题，侧面说明了：为何sMLP可以通过降低参数量取得更优结果。
- 在上述方案中，Swin Transformer取得了最佳性能。
- 尽管所提sMLPNet属于MLP类模型，但其具有与Swin Transformer相当甚至更优的性能。
- sMLPNet-T取得了81.9%top1精度，为现有FLOPs小于5B模型中性能最佳者；
- sMLPNet-B取得了与Swin-B相当的精度，但模型小25%，FLOPs少近10%。
- 需要注意的是：sMLPNet从S~B扩增并未看到过拟合；
- 上述结果表明：无注意力模块同样可以取得SOTA性能，进一步说明，注意力机制并非Transformer模型取得了优异的秘密武器。

Ablation Study

sMLPNet-T*	Param(M)	FLOPs(B)	Top-1(%)
Local+Global	19.2	4.0	81.3
Global only	19.1	3.9	80.6
Local only	22.5	4.4	80.7

Local only	22.3	4.4	80.7

Table 1: Ablation study on the effects of local and global modeling using the tiny model ($\alpha = 2$).

上表对sMLPNet的局部与全局建模的作用进行对比，从中可以看到：

- 移除DWConv后，模型参数量减少0.1，FLOPs减少0.1B，但性能下降了0.7%。这说明：DWCov是一种非常有效的局部依赖建模方案。
- 移除Local后，模型进去了80.7%精度。注：这个地方的原文分析看的莫名其妙。

S1	S2	S3	S4	Param(M)	FLOPs(B)	Acc.(%)
✓	✓	✓	✓	65.9	14.0	83.4
	✓	✓	✓	65.8	13.7	83.2
		✓	✓	64.3	12.4	83.0
			✓	49.9	9.5	82.2
				45.1	9.3	82.0

Table 2: Abation study on the effects of sMLP using sMLPNet-B ($\alpha = 3$) as the base model. We remove the sMLP block from the beginning of the network and evaluate the top-1 accuracy. A check mark in the corresponding stage (S1, S2, S3, and S4) means the use of sMLP module.

上表对比了sMLP模块在不同阶段的作用，可以看到：

- 完全移除sMLP后，模型性能为82%；
- 阶段3中的sMLP移除导致的性能下降最多，约0.8%，而其他阶段均为0.2%。

	Param(M)	FLOPs(B)	Top-1(%)
sMLPNet-S	48.6	10.3	83.1
Sum	33.2	7.0	81.5
Weighted sum	33.3	7.0	81.8
sMLPNet-T	24.1	5.0	81.9

Table 3: Comparison of different fusion methods. Base model is sMLPNet-S ($\alpha = 3$) which uses an FC layer for data fusion. Sum and weighted sum are two alternative fusion methods.

上表比较了不同融合方式的性能，可以看到：替换后Sum融合后，模型性能大幅下降，从83.1%下降到81.5%。这说明：**Concat+FC的融合方式具有更好的性能表现 。**

	Parallel	Sequential
w Identity	81.3	81.1
w/o Identity	80.9	80.6

Table 4: Ablation study on the design of the branches in the sMLP module. We use sMLPNet-T* ($\alpha = 2$) as the base model (parallel connection with the identity mapping).

上表比较了sMLP中各分支的排列方式性能，可以看到：

- Identity分支可以带来0.4%-0.5%性能提升；
- 并行处理要比串行处理具有更优的性能。

	Param(M)	FLOPs(B)	Top-1(%)
sMLPNet-T*	19.2	4.0	81.3
Multi-stage MLP	22.7	4.2	77.8
Single-stage MLP	30.4	6.5	76.8

Table 5: Ablation study on the effects of multi-stage architecture. Multi-stage MLP: sMLP at stage 1 is replaced by depth-wise conv and the sMLP at stage 2,3,4 is replaced by MLP. Single-stage MLP: all of the sMLP is replaced to MLP and multi-stage is replaced to singe-stage with patch size equal to 16×16.

上表对比了金字塔结构多阶段处理的重要性，从中可以看到：

- 将sMLP替换为MLP后，模型性能从81.3%下降到77.8%；
- 将多阶段处理替换后单阶段处理后，模型性能进一步下降到76.8%。

文末思考

这篇论文有种组合CSWin(7月份上线arXiv)、ViP(6月份上线arXiv)的味道，但实验部分又并未与两者进行比较。虽然文末提到了sMLPNet与CSWin是同期工作，见如下。

> We notice that some concurrent Transformer-based models, such as CSWin, have o btained an even higher accuracy than sMLPNet...

但是sMLPNet并未提到6月份就已上线arXiv的ViP，着实不应该，关键的是：两者的部分思想是那么相似。此外，消融实验中也看到了ViP中的加权融合与Concat+FC融合的对比。隐约说明：sMLPNet应该是知道ViP，但因为性能不如ViP而刻意没写。

当然，sMLPNet的重点不是跟谁比性能，旨在对Transformer中的自注意力机制的必要性进行挖掘与探索，并最终得出本文的核心思想："**自注意力并非Transnformer取得优异性能的秘密武器**"。

如果觉得有用，就请分享到朋友圈吧！

极市平台签约作者#

happy

知乎：AIWalker

AIWalker运营、CV领域八年深耕码农

研究领域：专注low-level领域，同时对CNN、Transformer、MLP等前沿网络架构保持学习心态，对detection的落地应用甚感兴趣。

公众号：AIWalker

作品精选

- 吊打一切现有版本的YOLO！旷视重磅开源YOLOX：新一代目标检测性能速度担当！
- YOLOv4团队开源最新力作！1774fps、COCO最高精度，分别适合高低端GPU的YOLO
- 图像增强领域大突破！以1.66ms的速度处理4K图像，港理工提出图像自适应的3DLUT

投稿方式：

添加小编微信Fengcall（微信号：fengcall19），备注：**姓名-投稿**

△长按添加极市平台小编

觉得有用麻烦给个在看啦~

收录于话题 #Transformer 45

上一篇

多快好省的目标检测器来了！旷视孙剑团队提出Anchor DETR：基于Anchor Point的...

下一篇

搞懂 Vision Transformer 原理和代码，看这篇技术综述就够了（十六）

阅读原文

喜欢此内容的人还喜欢

谷歌MaskGIT | 双向Transformer，图像生成新范式！

机器学习算法工程师

ShiftViT：Swin Transformer的成功不在注意力！

机器学习算法工程师

十分钟理解Transformer

新机器视觉