TimeSformer:2021

Is Space-Time Attention All You Need for Video Understanding?

论文背景

在NLP中，transformer表现十分优异。

视频处理和自然语言处理在很多方面具有相似之处：句中一个单词的意思需要参照上下文中的其他词，视频中对动作的识别也需要依赖于前后几帧。目前还没有讲自注意力机制运用到视频模型的尝试。

transformer具有更小的归纳偏置，扩大了可表示的函数族。

创新思路

将ViT中的二维空间扩展到时空三维空间

由于SA的计算复杂度与输入的patch数量成二次方关系，所以设计了几种自注意力机制来改善这个问题

方法介绍

输入$X \in R^{H\times W\times 3\times F}$

切分patch：采用和ViT一样的方法

线性嵌入：$Z^0_{(p,t)} = EX_{(p,t)} + e^{pos}_{(p,t)}$,E是一个可学习的矩阵，$e^{pos}_{(p,t)}$是一个可学习的位置嵌入

Q、K、V：

$$\begin{aligned}
\mathbf{q}_{(p,t)}^{(\ell,a)} &= W_Q^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h} \\
\mathbf{k}_{(p,t)}^{(\ell,a)} &= W_K^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h} \\
\mathbf{v}_{(p,t)}^{(\ell,a)} &= W_V^{(\ell,a)} \mathrm{LN}\left(\mathbf{z}_{(p,t)}^{(\ell-1)}\right) \in \mathbb{R}^{D_h}
\end{aligned}$$

自注意力计算：

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t')}^{(\ell,a)}\right\}_{\substack{p'=1,\ldots,N \\ t'=1,\ldots,F}}\right]\right)$$

SM为softmax函数

当注意力只在一个维度上计算的时候，计算量会显著减少，例如在空间注意时，只是用N+1个键值对，只是用与查询来自于同一帧的键。

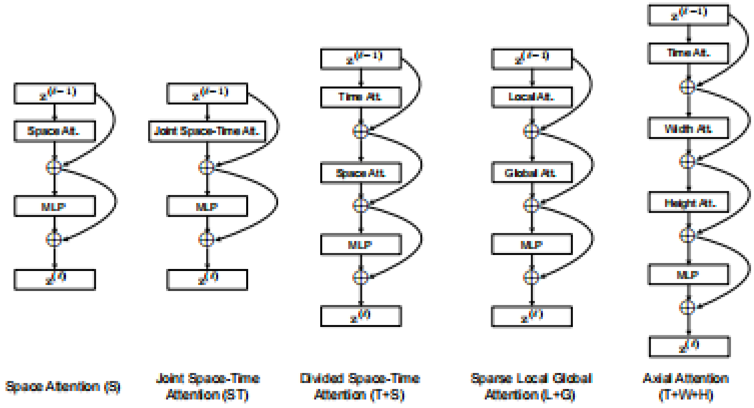
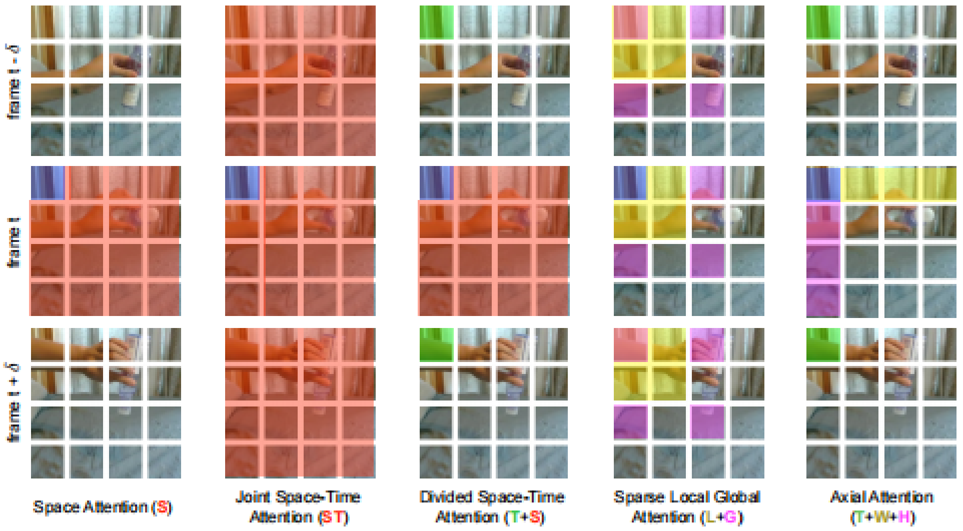


Figure 1. Illustration of the video self-attention blocks that we investigate in this work. Each attention layer implements self-attention (Vaswani et al., 2017b) on a specified spatiotemporal neighborhood of frame-level patches (see Figure 2 for a visualization of the neighborhoods). We use residual connections to aggregate information from different attention layers within each block. A 1-hidden-layer MLP is applied at the end of each block. The final model is constructed by repeatedly stacking these blocks on top of each other.


frame t - δ
frame t
frame t + δ
Space Attention (S)
Joint Space-Time Attention (ST)
Divided Space-Time Attention (T+S)
Sparse Local Global Attention (L+G)
Axial Attention (T+W+H)

space attention:$\alpha_{(p,t)}^{(\ell,a)\text{ space}} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t)}^{(\ell,a)}\right\}_{p'=1,\ldots,N}\right]\right)$

Encoding:

$$\mathbf{s}_{(p,t)}^{(\ell,a)} = \alpha_{(p,t),(0,0)}^{(\ell,a)} \mathbf{v}_{(0,0)}^{(\ell,a)} + \sum_{p'=1}^{N} \sum_{t'=1}^{F} \alpha_{(p,t),(p',t')}^{(\ell,a)} \mathbf{v}_{(p',t')'}^{(\ell,a)}$$

$$\mathbf{z}'^{(\ell)}_{(p,t)} = W_O \begin{bmatrix} \mathbf{s}_{(p,t)}^{(\ell,1)} \\ \vdots \\ \mathbf{s}_{(p,t)}^{(\ell,\mathcal{A})} \end{bmatrix} + \mathbf{z}_{(p,t)}^{(\ell-1)}$$

$$\mathbf{z}_{(p,t)}^{(\ell)} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z}'^{(\ell)}_{(p,t)}\right)\right) + \mathbf{z}_{(p,t)}^{(\ell)}$$

Classification embedding

$$\mathbf{y} = \mathrm{LN}\left(\mathbf{z}_{(0,0)}^{(L)}\right) \in \mathbb{R}^D$$

Time Self-Attention Model

$$\alpha_{(p,t)}^{(\ell,a)\text{ time}} = \mathrm{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p,t')}^{(\ell,a)}\right\}_{t'=1,\ldots,F}\right]\right)$$

实际效果

Attention	Params	K400	SSv2
Space	85.9M	77.6	36.6
Joint Space-Time	85.9M	78.1	58.5
Divided Space-Time	121.4M	**78.5**	**59.5**
Sparse Local Global	121.4M	76.8	56.3
Axial	156.8M	74.6	56.2

Table 1. Video-level accuracy for different space-time attention schemes in TimeSformer. We evaluate the models on the validation sets of Kinetics-400 (K400), and Something-Something-V2 (SSv2). We observe that divided space-time attention achieves the best results on both datasets.

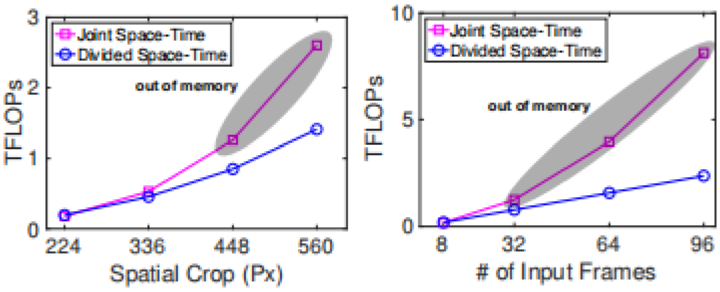


Figure 3. We compare the video classification cost (in TFLOPs) of Joint Space-Time versus Divided Space-Time attention. We plot the number of TFLOPs as a function of spatial crop size in pixels (left), and the number of input frames (right). As we increase the spatial resolution (left), or the video length (right), our proposed divided space-time attention leads to dramatic computational savings compared to the scheme of joint space-time attention.

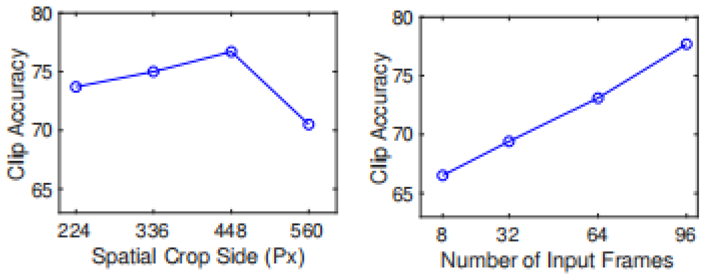


Figure 4. Clip-level accuracy on Kinetics-400 as a function of spatial crop size in pixels (left), and the number of input frames (right).

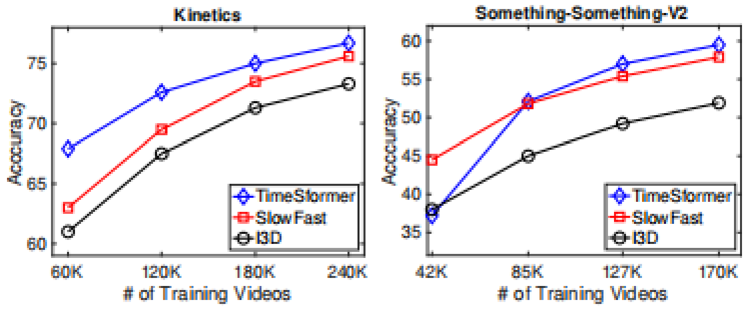
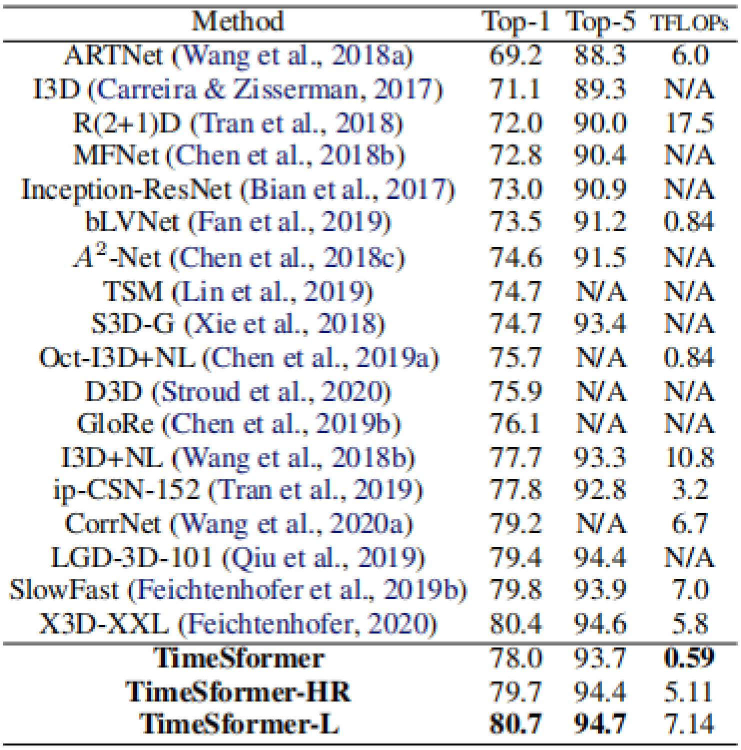


Figure 5. We study accuracy on Kinetics-400 (K400), and Something-Something-V2 (SSv2) as a function of the number of training videos. On K400, TimeSformer performs best in all cases. On SSv2, which requires more complicated temporal reasoning, TimeSformer outperforms the other models only when using enough training videos. All models are pretrained on ImageNet.



Method	Top-1	Top-5	TFLOPs
ARTNet (Wang et al., 2018a)	69.2	88.3	6.0
I3D (Carreira & Zisserman, 2017)	71.1	89.3	N/A
R(2+1)D (Tran et al., 2018)	72.0	90.0	17.5
MFNet (Chen et al., 2018b)	72.8	90.4	N/A
Inception-ResNet (Bian et al., 2017)	73.0	90.9	N/A
bLVNet (Fan et al., 2019)	73.5	91.2	0.84
A^2-Net (Chen et al., 2018c)	74.6	91.5	N/A
TSM (Lin et al., 2019)	74.7	N/A	N/A
S3D-G (Xie et al., 2018)	74.7	93.4	N/A
Oct-I3D+NL (Chen et al., 2019a)	75.7	N/A	0.84
D3D (Stroud et al., 2020)	75.9	N/A	N/A
GloRe (Chen et al., 2019b)	76.1	N/A	N/A
I3D+NL (Wang et al., 2018b)	77.7	93.3	10.8
ip-CSN-152 (Tran et al., 2019)	77.8	92.8	3.2
CorrNet (Wang et al., 2020a)	79.2	N/A	6.7
LGD-3D-101 (Qiu et al., 2019)	79.4	94.4	N/A
SlowFast (Feichtenhofer et al., 2019b)	79.8	93.9	7.0
X3D-XXL (Feichtenhofer, 2020)	80.4	94.6	5.8
TimeSformer	78.0	93.7	**0.59**
TimeSformer-HR	79.7	94.4	5.11
TimeSformer-L	**80.7**	**94.7**	7.14

Table 2. Video-level accuracy on Kinetics-400. TimeSformer-L achieves the best reported accuracy.

个人理解

视频分类中由于数据量太大，所以很多的模型都去寻找一个分解的办法，开始用2D卷积的时候发现效果不是很好，因为没有注意到时间域上的信息，之后就有人将2维卷积扩展到了三维卷积，提取时空信息，也有用2D卷积＋时间信息做行为识别的，比如双流法，将光流作为时间信息引入，后续有实验证明光流对于行为识别很有作用，但是很多的工作都是基于RGB图像来做的，还有LRCN，通过先用CNN对每帧图像提取信息之后，经过LSTM层提取空间信息。3D卷积的代表方法中有P3D、SlowFast等，其中P3D的思路就是去分解卷积，一部分去做空间上的识别（1×n×n），一部分去提取时间上的特征（n×1×1），降低参数量和过拟合的风险。SlowFast的思路：动作有快慢之分，人体的视觉细胞中也是这样，80%对于慢动作敏感（也可以理解为主要对空间信息感兴趣），20%对快动作敏感（可以理解为主要对时间信息感兴趣），所以作者就去搭建了一个slow path和一个fast path，同时增加了fast path向slow path的数据融合 。TimeSformer本质上还是对参数量问题进行的 改善，改善的最根本思路就是去拆解整个视觉任务，时间上的可以拆分为几帧上的所有同一位置的patch做Self-Attention，空间上的

拆分为每帧上的所有patch做Self-Attention。论文中作者还设计了Sparse Local Global Attention和Axial Attention，最后通过实验，还是发现：分开的时间空间注意力机制效果更好。