

## 每日小结

	周一	周二	周三	周四	周五
早	跑模型	跑模型	组会 ppt	阅读文献	组会 ppt
中	跑模型	论文阅读	论文阅读	跑代码	组会 ppt
晚		论文代码阅读		组会 ppt	组会 ppt

注：简单表述当前时间段工作，如看文献1，整理数据等

## 科研详情

## 文献阅读

## 文献1

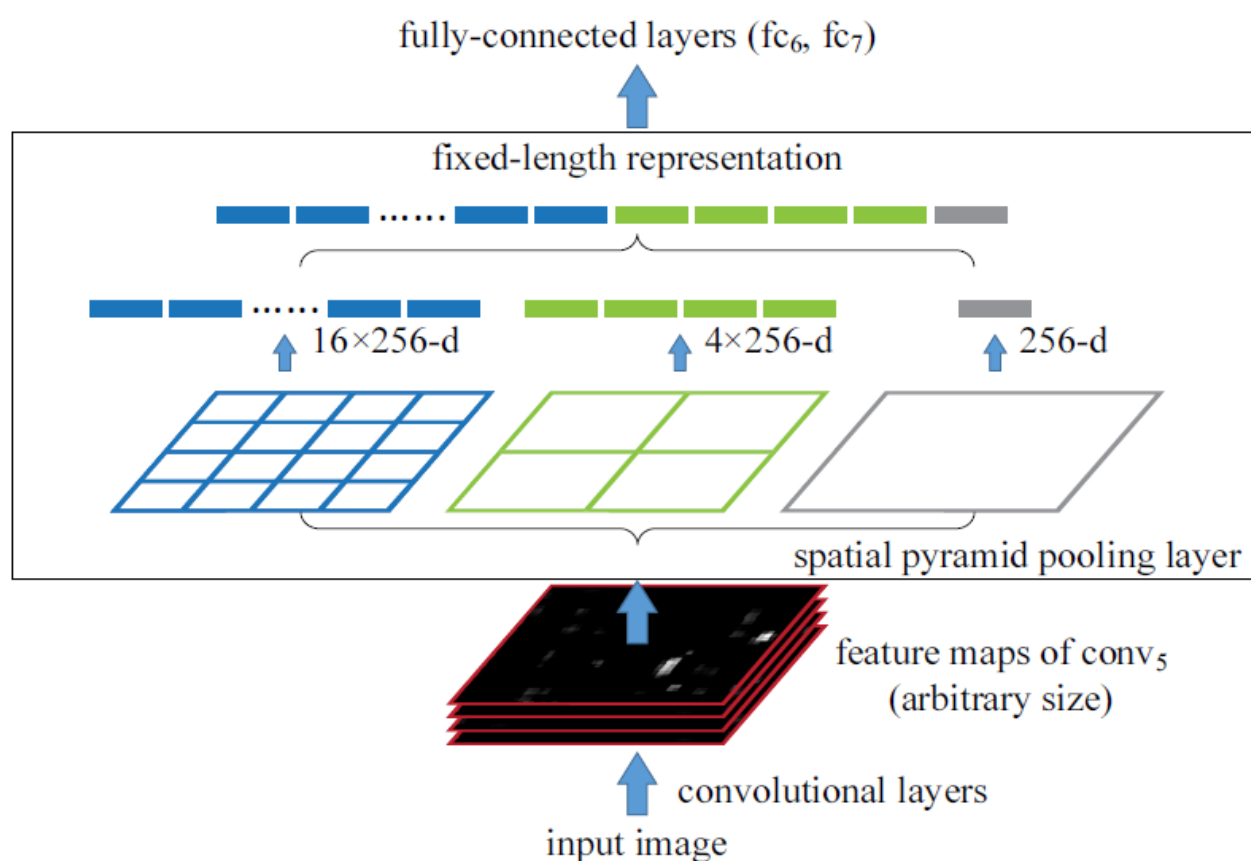
题目： **Learning Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition**

作者： **Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun**

出处： **arxiv1506**

方法：

文章以 AlexNet 为例，讨论 SPP layer。AlexNet 包括五层卷积层和两层全连接层，在最后一层卷积层后面加上 SPP 层，如图所示：



金字塔池化是分等级的，每一级的大小都用  $n \times n$  bins 来描述，意思是经过该等级的池化操作后，输出特征图的大小就是  $n \times n$ 。那么如何做到这一点呢，因为我们池化都是用滑动窗口来实现的，这涉及到滑动窗口大小和步长。金字塔池化的精髓就是滑动窗口的大小和步长是正比于输入图像的大小的，这样就能够产生大小（长度）的输出特征。

$$\text{window size } win = \lfloor a/n \rfloor$$

$$\text{stride str} = \lfloor a/n \rfloor$$

因此，不管 conv5 特征图的尺寸是多少，经过 SPP 后都产生  $15 \times 256$  的输出，然后送入全连接层中进行下一步的分类等操作。卷积层的输出特征图的尺寸会根据输入图像的尺寸发生变化，因此，加入了 SPP 模块，SPPNet 就可以处理任意尺寸输入的图片进行训练和测试。

启发：

1. CNN 由于全连接层的存在，需要固定尺寸的输入图像，这限制了输入图像的纵横比和尺度，当前方法主要通过裁剪或变形将输入图像变换到固定尺寸，这样会导致图像内容的丢失或失真，从而影响图像识别的准确性。
2. SPP 网络的输出特征长度固定，与输入无关。
3. SPP 的多级池化提取多尺度特征，使网络对图像变形更加鲁棒。
4. SPP 可以处理不同尺寸的输入图像，提升网络的性能。

## 文献2

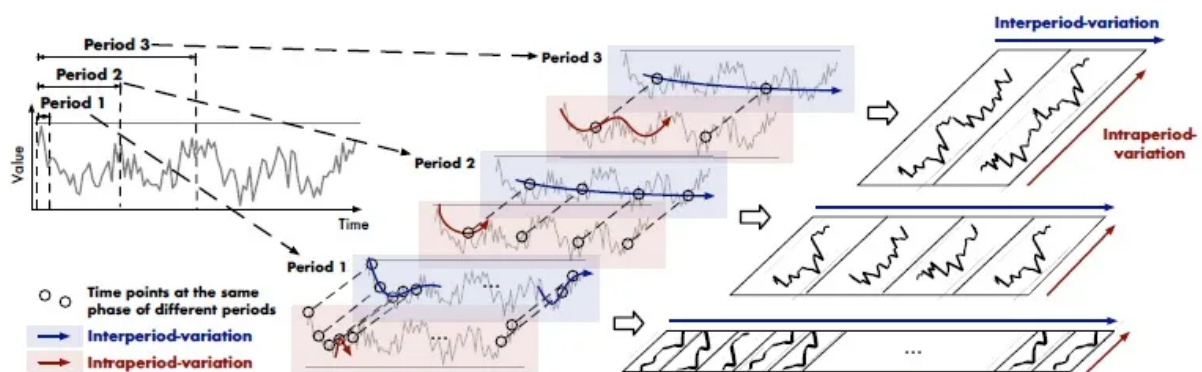
题目：Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting

作者：Haixu Wu, Jiehui Xu, Jianmin Wang, Mingsheng Long

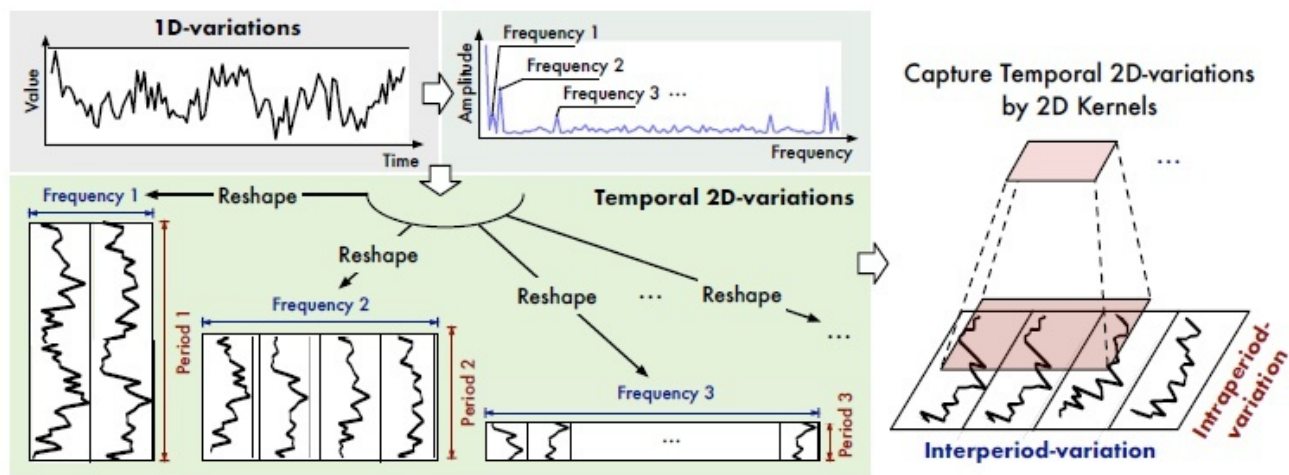
出处：arxiv2106

方法：

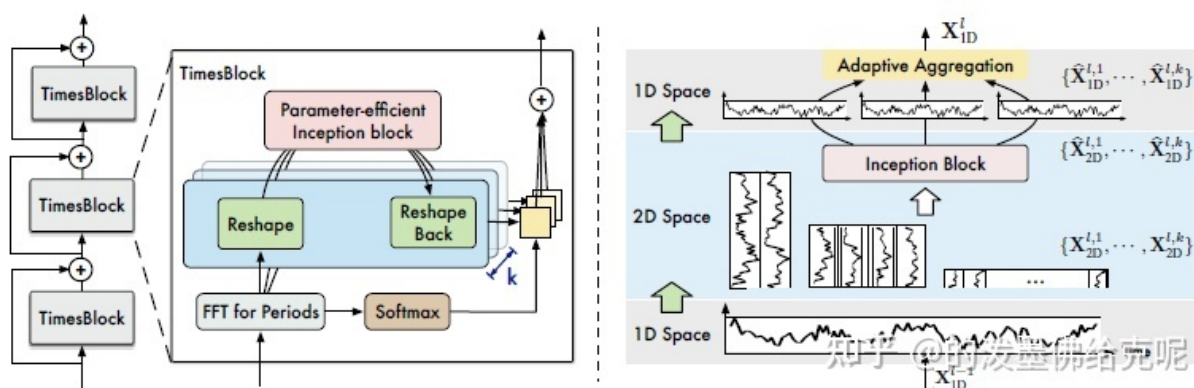
大部分现有方法都是作用于时间序列的时间维度，捕获时序依赖性。实际上，现实时间序列一般都有多种模式，比如不同的周期，各种趋势，这些模式混杂在一起。如果直接对原始序列的时间维度来建模，真正的时序关系很可能隐藏在这些混杂的模式中，无法被捕获。考虑到，作者提出将 1D 的时间维度 reshape 成 2D 的，下图左侧的时间序列具有三个比较显著的周期性（Period 1、Period 2、Period 3），将其 reshape 成三种不同的 2D-variations，2D-variations 的每一列包含一个时间段（周期）内的时间点，每一行包含不同时间段（周期）内同一阶段的时间点。变成 2D-variations 之后，就可以采用 2D 卷积等方式来同时捕获时间段内部依赖和相邻时间段依赖。



那么怎么确定时间序列中的周期性呢？采用傅里叶变换。给时间序列做傅里叶变换后，主要的周期会呈现对应的高幅值的频率分量。设定超参数  $k$ ，然后只取 top  $k$  个最大的幅值对应的频率分量，即可得到 top  $k$  个主要的周期，这和 Autoformer 中的处理类似。具体操作如下图，左侧是确定 top  $k$  个周期，在此只画了三个，然后将 1D 的时间序列 reshape 成 3 种不同的 2D-variations（不能整除的可以用 padding），对这三种 2D-variations 用 2D 卷积进行处理之后再聚合结果即可。



TimesBlock: 得到 $k$ 个2D-variations之后该怎么处理呢？本文提出了TimesBlock，每层TimesBlock又分为两步。首先是要先对这些2D-variations分别用2D卷积（可以是ResNet、ConvNeXt等）或者其他的视觉骨干网络（比如Swin, Vit）处理；其次将 $k$ 个处理后的结果再聚合起来。对于第一步，本文采用了一种参数高效的Inception block。Inception block是GoogleNet中的模块，包含多个尺度的2D卷积核。如下图左侧蓝色区域，处理 $k$ 个2D-variations的Inception block是参数共享的。因此，模型整体的参数量不会随着超参数 $k$ 的增大而增大，因此本文将其称为参数高效的Inception block (Parameter-efficient Inception block)。



启发:

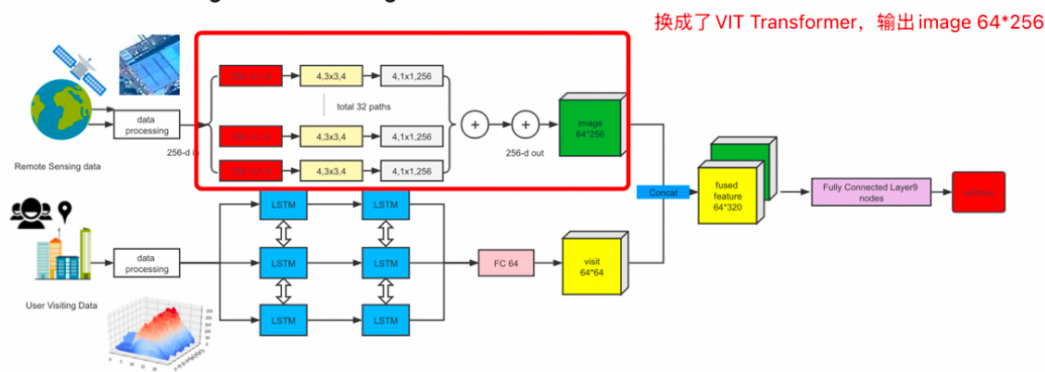
1. 不同于 Autoformer 只集中于时间序列预测，本文提出的 TimesNet 是一个通用的时间序列神经网络骨干，可处理各种不同的时间序列任务，如最常见的任务：预测、分类、异常检测等等。其实几乎所有的时间序列预测模型也可以当做是通用骨干，比如 Autoformer, Informer, FEDformer, Preformer 这些 Transformer-based 模型中只采用 Encoder 就相当于是一个时间序列的特征提取器，区别在于它们捕获时序依赖性的方式不同。比如 Autoformer 是用 Auto-Correlation, Informer 中的概率稀疏 Attention, FEDformer 的频域 Attention, Preformer 中的 Multi-Scale Segment-Correlation。还有那些 MLP-based 模型比如 DLinear 也可以当做是通用骨干，它是直接采用线性层权重来表示时序依赖性。
2. 大部分现有方法都是作用于时间序列的时间维度，捕获时序依赖性。实际上，现实时间序列一般都有多种模式，比如不同的周期，各种趋势，这些模式混杂在一起。如果直接对原始序列的时间维度来建模，真正的时序关系很可能隐藏在这些混杂的模式中，无法被捕获。考虑到：现实世界的时间序列通常具有多周期性，比如每天周期、每周周期、每月周期；而且，每个周期内部的时间点是有依赖关系的（比如今天 1 点和 2 点），不同的相邻周期内的时间点也是有依赖关系的（比如今天 1 点和明天 1 点），作者提出将 1D 的时间维度 reshape 成 2D），变成 2D-variations 之后，就可以采用 2D 卷积等方式来同时捕获时间段内部依赖和相邻时间段依赖。

## 工作进展

- 1: 阅读文献；寻找transformer和lstm层级间进行权重共享或者约束的论文
- 2: 准备组会PPT
- 3: 实验结果，transformer+DPN 代码完成，还在跑，结果没出来：

Model	Accuracy	F1 score
ResNext101+RNN	0.566	0.474
ResNext101+GRU	0.64	0.567
ResNext101+BiGRU	0.653	0.550
ResNext101+LSTM	0.654	0.574
ResNext101+BiLSTM	0.668	0.571
VIT transformer（无预训练）+BiLSTM	0.607	0.510
VIT transformer（预训练）+BiLSTM	0.642	0.546

### Remote sensing - Social sensing data Fusion Network



VIT transformer（无预训练）+BiLSTM	Accuracy
后接softmax分类	0.607
后接MLP分类	0.593

## 下周计划

- 1: 阅读文献；寻找 transformer 和 lstm 层级间进行权重共享或者约束的论文，遥感和社交数据可以在层级对比权重信息，最大程度互补，想办法改模型
- 2: 参照涂老师的思路，增加相关多模态损失函数，捕获不同模态的相似特征：
- 3: 打算尝试其他 transformer+DPN、transformer+ transformer模型效果，和transformer+bilstm 进行比较
- 4: 想尝试下对比学习方法，还在看文献源码。