

# 周报（2023.9.5-2023.9.12） 姓名：孙瑞阳

## 每日小结

	周一	周二	周三	周四	周五
早	看代码，学 graph	看代码，学 graph	看文献和代码	论文撰写	看代码，学 graph
中	论文撰写	看代码，学 graph	看文献和代码	查阅文献，学 graph	论文撰写
晚	论文撰写	组会		论文撰写	

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

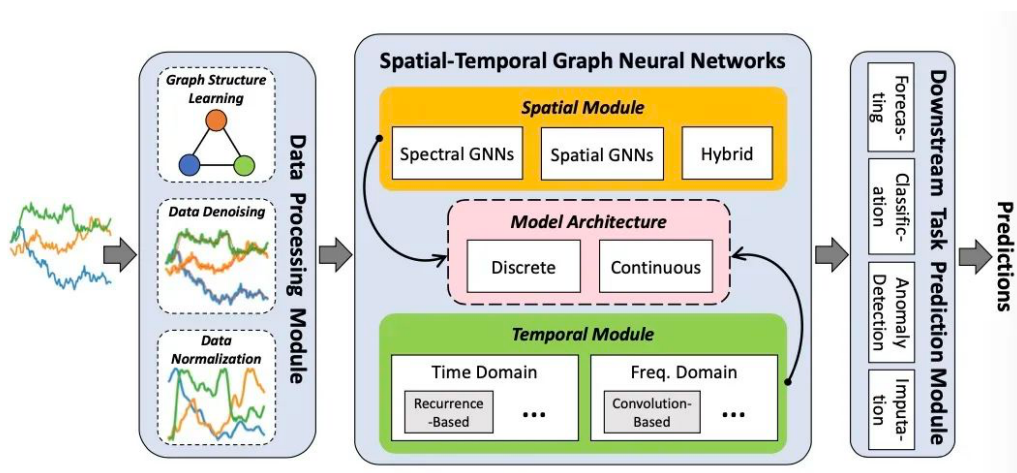
题目：A Survey on Graph Neural Networks for Time Series: Forecasting, Classification, Imputation, and Anomaly Detection

作者：Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I. Webb, Irwin King, Shirui Pan

出处：Arxiv 2023

方法：

通过将时间点或变量视为节点，将它们的关系视为连边，以网络或图的形式构造模型可以有效学习时间序列中的复杂交互模式，这在城市交通网络、人口迁移和天气预测等场景中得到了广泛的应用。



本篇综述从时间序列预测、分类、异常检测和数据填补这四个任务维度出发，回顾了图神经网络在时间序列领域的相关研究。

- 单变量时间序列： $\mathbf{X} = [x_1, x_2, \dots, x_t] \in R^T, t = 1, 2, \dots, T$ 为规律的采样时刻，若为不规则采样的序列，则可表示为 $\mathbf{X} = [(x_1, t_1), (x_2, t_2), \dots, (x_T, t_T)] \in R^T$ 。
- 多元时间序列： $\mathbf{X} \in R^{N \times T}, t = 1, 2, \dots, T$ 为采样时刻，GNN相关研究主要关注的是多元时间序列，不同的时间序列之间的关系可通过如下时空图表示。
- 属性图： $\mathbf{G} = (\mathbf{A}, \mathbf{X})$ 由节点特征矩阵 $\mathbf{X}$ 和其对应的邻接矩阵 $\mathbf{A} \in R^{N \times N}$ 组成，邻接矩阵 $\mathbf{A}$ 描述了N个节点间的拓扑图结构，包含N个节点 $V = v$ ，节点间的连边，连边的权重

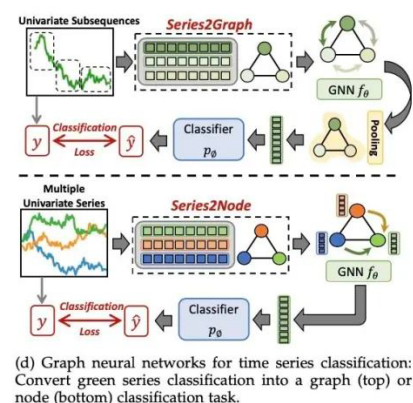
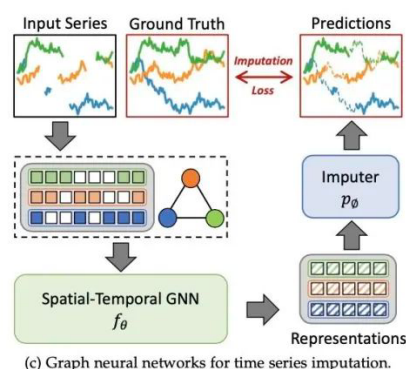
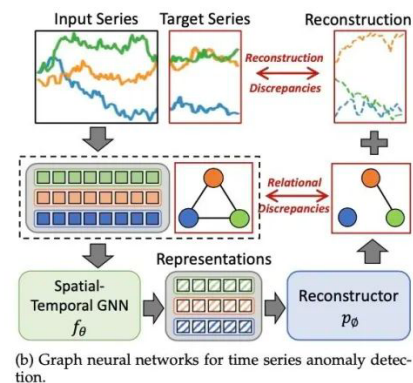
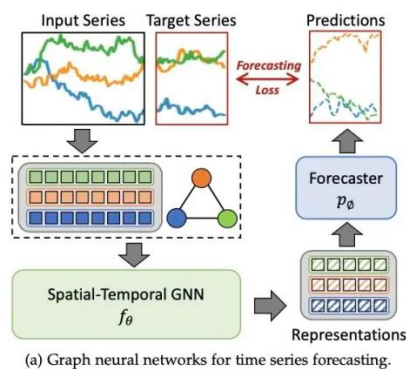
利用 GNN 分析时间序列，通常需要提供图结构即邻接矩阵。然而，并非所有时间序列数据都有现成的图结构可用，在实践中，通常使用两种策略来从数据中获取图结构：启发式方法或直接学习的方法。

- 启发式图：这类方法根据一些启发式的指标从数据中去发现图结构，例如空间相似性、节点对是否连接，节点特征相似度、基于格兰杰因果相关或传递熵等函数确定节点的连接及邻接矩阵。
- 基于图学习的方法：这类方法将直接将邻接矩阵作为一个参数，在端到端训练下游任务时进行学习，相比于启发式图，这类方法能够学习更复杂或更有信息的图。

## 用于不同时间序列分析任务的图神经网络

文章梳理了 GNN 在时间序列预测、时间序列异常检测、分类及异常检测四个关键任务维度的相关研究，这些方法共享一个通用的 pipeline，即利用时空图对时间序列变量间进行建模，实现对单个序列的表征，如下图所示。其中，通过时空图（STGNN）进行时序数据表征，可将时间序列分类任务转变为图分类或节点分类任务。

在上述 pipeline 中的时空图，由空间模块（spatial module）和时序模块（temporal module）两个关键模块组成。其中，spatial module 用于建模不同时间序列（节点）间的相互影响关系，常使用 spectral GNN，spatial GNN，或两种方法的组合。spectral GNN 基于频谱图理论，使用图偏移算子（如图拉普拉斯）来捕获节点关系，spatial GNN 对 spectral GNN 进行了简化，直接通过过滤器筛选各个节点的邻居，混合方法则是集合了二者的优势。Temporal module 用于建模序列样本沿时间方向的相关性，包含时域或频域。时域分析常利用 RNN, TCN 和 Transformer 等网络，频域分析常利用傅立叶变化、小波变换等。



# 用于时序分类的 GNN

基于 GNN 的时间序列分类方法通过将时间序列数据转换为图表示，可以充分利用 GNN 的强大功能来捕捉时间序列中的局部和全局模式。对于单变量时间序列，现有方法可分为 Series2Node 和 Series2Graph 两类。Series2Graph 将单变量时间序列分解成多个子序列，每个子序列作为一个节点，从而将单变量时间序转换为一张图，图中的边则描述了不同子序列沿时间方向的依赖关系，最后通过图神经网络（GNN）进行图分类。Series2Node 方法将时间序列的样本视为一个单独的节点，节点的连边描述了不同样本间的关系，创建一张大图，提供对整个数据集中时序依赖关系的完整描述。多变量时间序列分类与单变量时间序列分类具有相同的范式，只是多变量时间序列需要进一步考虑对不同序列间空间依赖关系的建模。下表总结了目前有关 GNN 进行时间序列分类的相关工作。

TABLE 4: Summary of graph neural networks for time series classification. Task notation: “U” and “M” refer to univariate and multivariate time series classification tasks. Conversion represents the transformation of a time series classification task into a graph-level task as either graph or node classification task, represented as “Series2Graph” and “Series2Node”, respectively. The remaining notations are shared with Table 2.

Approach	Year	Venue	Task	Conversion	Spatial Module	Temporal Module	Missing Values	Input Graph	Learned Relations	Graph Heuristics
MTPool [175]	2021	NN	M	-	Spatial GNN	T-C	No	NR	S	-
Time2Graph+ [73]	2021	TKDE	U	Series2Graph	Spatial GNN	-	No	R	-	PS
RainDrop [46]	2022	ICLR	M	-	Spatial GNN	T-A	Yes	NR	S	-
SimTSC [74]	2022	SDM	U+M	Series2Node	Spatial GNN	T-C	No	R	-	PS
LB-SimTSC [75]	2023	arXiv	U+M	Series2Node	Spatial GNN	T-C	No	R	-	PS
TodyNet [176]	2023	arXiv	M	-	Spatial GNN	T-C	No	NR	D	-

启发:

- 1. 结合综述在看图的一些基础理论知识，目前已经考虑了自己数据集上两种 graph 的建立方式

## 文献2

题目: Exchanging-based Multimodal Fusion with Transformer

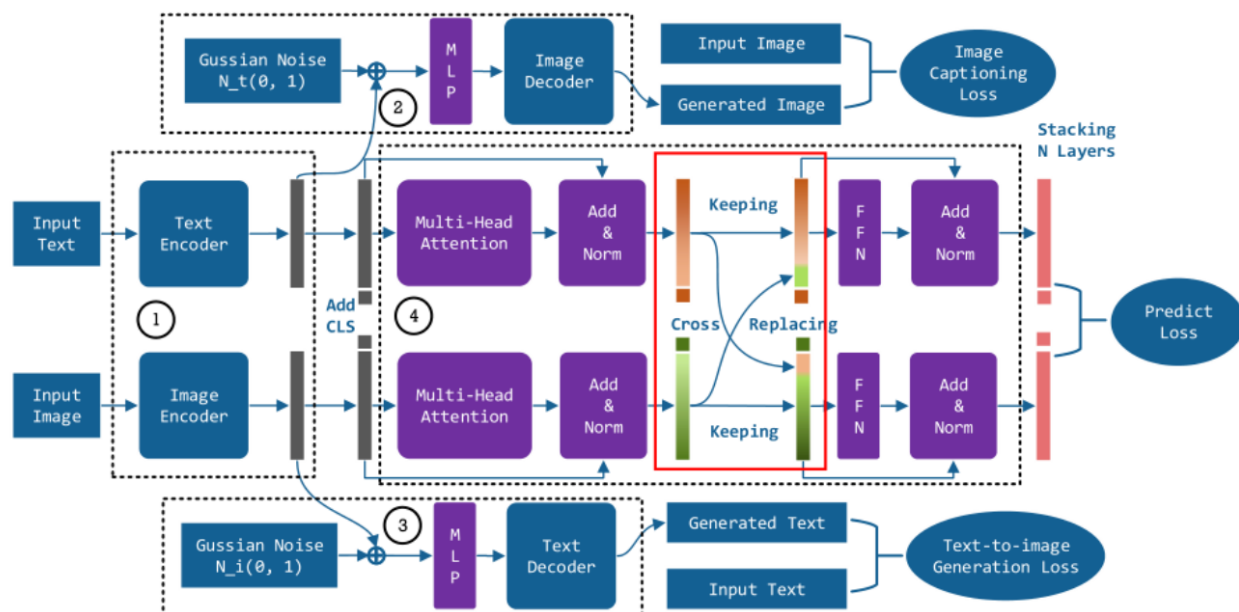
作者: Renyu Zhu, Chengcheng Han, Yong Qian, Qiushi Sun, Xiang Li, Ming Gao, Xuezhi Cao, Yunsen Xian

出处: arxiv2023

方法:

本文提出了一种基于交换的多模态融合方法，即 MuSE，弥补了基于交换的文本视觉多模态融合方法在文本视觉多模态融合领域的空白。

作者提出了一个深度多模态融合模型 MuSE。如图 2 所示，MuSE 主要由四个功能组件组成，我们用①-④标记它们。所述组件①用于将输入文本和图像投影到低维空间中，所述组件包括文本编码器和图像编码器。考虑到多模态数据可能被映射到不同的空间中，我们进一步提出了两个嵌入正则器(参见组件②和③)来将多模态输入的嵌入拉到同一空间中。将这两个正则化器作为解码器，分别实现文本到图像的生成任务和图像字幕任务。在生成多模态输入的嵌入后，我们将它们馈送到组件④，这是一个基于 transformer encoder 的模块，称为 cross transformer，cross transformer 进行多模态信息交换，并最终从多模态生成融合嵌入。



在基于交换的融合之前，使用两个编码器-解码器的架构将文本(T)和视觉(I)模态的输入投影到相同的嵌入空间中。进行低维投影：首先使用两个独立的编码器分别将输入的文本和图像编码为低维嵌入：这里文本编码器可以是 Bert，图像编码器可以是 ResNet。

嵌入后加噪声，使用两个解码器分别生成图像 I 和文本 T：ImageDecoder 可以是典型的文本到图像生成模型(例如 PixelCNN (V and den Oord 等人，2016))，TextDecoder 可以是典型的图像字幕模型(例如 NIC (Vinyals 等人，2015))。

基于生成的图像和文本与输入的图像和文本进行对比，分别构造了文本到图像的生成损失  $L_{ti}$  和图像字幕损失  $L_{it}$ 。这两种损失对编码器生成的文本和图像的嵌入进行了正则化，可以看作是后面将要介绍的主要预测任务的辅助任务。

## 启发：

cross transformer 使用两个具有共享参数的 Transformer 编码器来学习文本和视觉模态的嵌入，并在多模态之间进行信息交换。首先将 cls 添加到文本编码器和图像编码器生成的嵌入的开头，这些嵌入被作为 cross transformer 的输入。之后，考虑到两个输入向量的全局上下文信息应该先学习然后交换，cross transformer 将其浅层设置为常规 Transformer 编码器层，然后是多个交换层。当多模态融合结束时，交换过程停止。

这种 cross 的跨模态交互很值得借鉴，还有他的 loss 函数也可以尝试在我的工作里。

## 工作进展

1: 阅读文献；

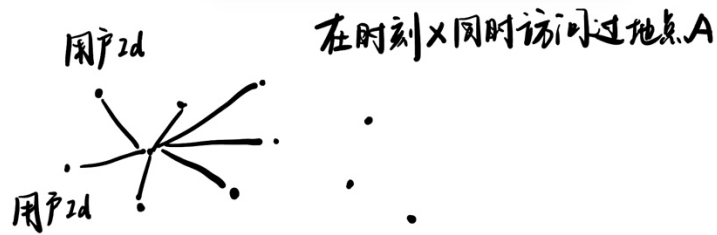
2: 期刊论文写完了

3: cross 的跨模态交互代码在看，准备跑一下

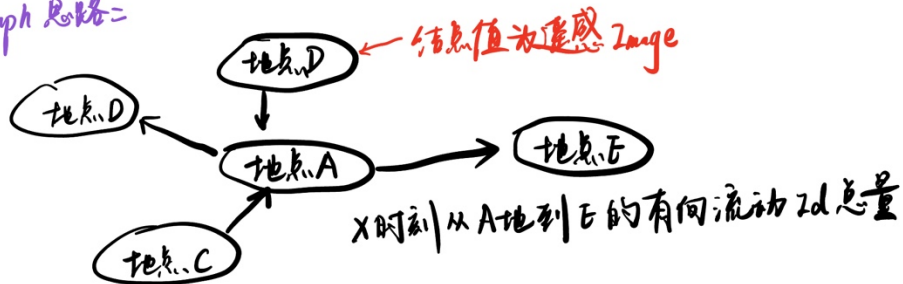
3:考虑了两种自己数据集的 graph 的建立：目前觉得第二种更好一些，还在想其他的建立方式



graph 思路一



graph 思路二



下周计划

1. 看 embedding 多模态融合的代码
2. 看 graph 的建立有没有其他思路
3. 修改论文