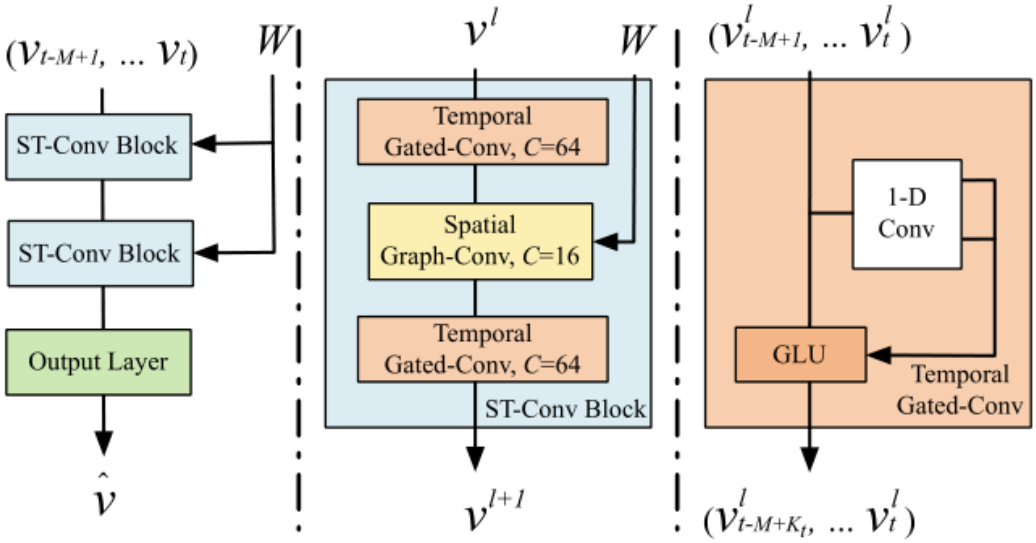
一、解决方案：

对于该赛题，我们队的主要的解决方案是采用时空图卷积神经网络模型与时序卷积神经网络模型进行融合，下面将具体讲述各解决方案的具体实现。

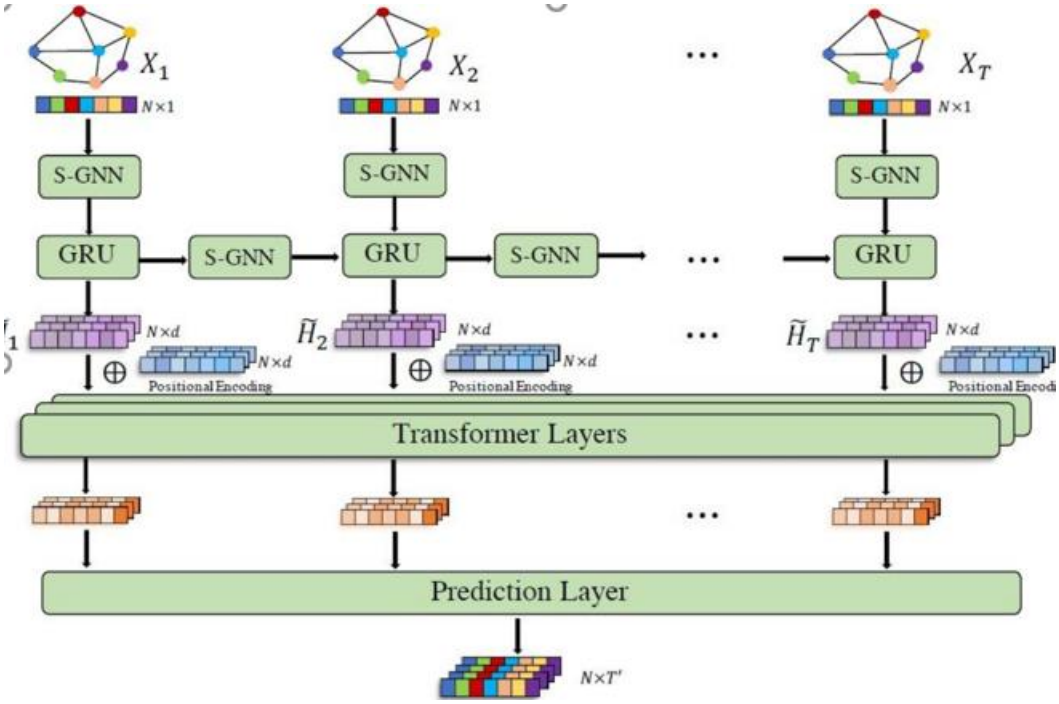
1. 时空图卷积神经网络

本赛题主题为时序图预测，所以很容易想到本题的方向在于对官方提供的边集和节点集建立图数据，并且挖掘图数据中的时序关系，来对图进行节点回归。

综上时序节点回归被我们确定为整体建模的方向，通过在网上搜索的资料，我们发现了一些与该赛题相关度较高的论文，诸如 STGCN,MTGNN 等等，通过提炼论文中的关键思想，我们了解到此类时空图研究的主要关键点在于对时空图数据进行时空特征的提取和分解。同时在阅读论文中，我们了解到了一般使用在时空图问题中的时间和空间卷积模块。时间卷积模块主要是 GRU,CNN,LSTM,空间卷积则是 GCN,SGC。

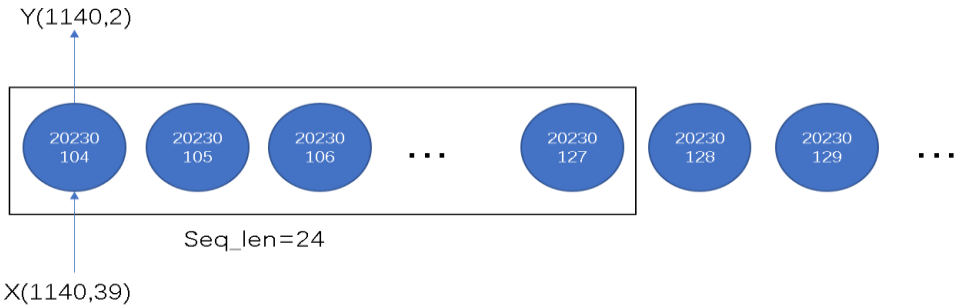


上图为 STGCN 的模型设计，可以看到每个 ST-conv 包含了时间和空间卷积模块

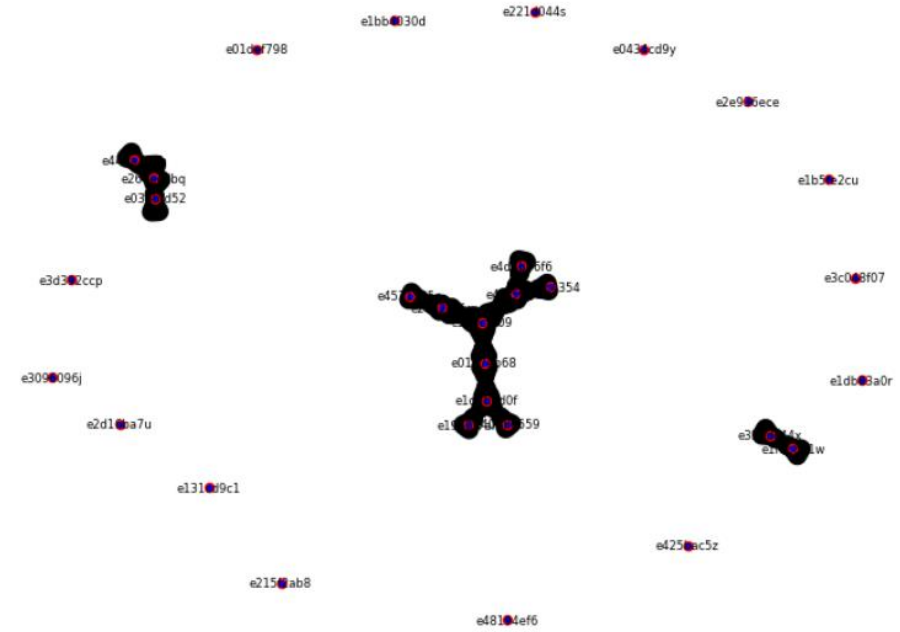


上图表示了时空图的输入输出形式，提取关键在于多图按时序输入，以便模型更好学习多图中的时序关系。

在充分学习后，我们在数据构建时主要是以时间为划分，将训练集的 90 个时间戳构造为对应 90 个图数据，每个图有 1140 个节点，同时由于我们采用了时空图卷积网络，对于特征的依赖是相对低的，在原有的栅格数据基础上，我们额外提取了月，日，星期，是否周末，4 维度特征，每个节点共 39 维节点特征。每个图对于的目标变量则是当天 1140 个 id 的 y。在 90 个空间图构建好后，我们采用了时序赛题常用的数据分割方法，滑窗法。滑窗长度 24，表示每次使用 24 张图进行输入，因为时空图的原因，24 张图需要按顺序输入，以便模型学习时序特征，因此我们重新建立了一个大图，在大图中，以 24 张图，每张图作为一个节点，边的连接表示按时间顺序进行连接。

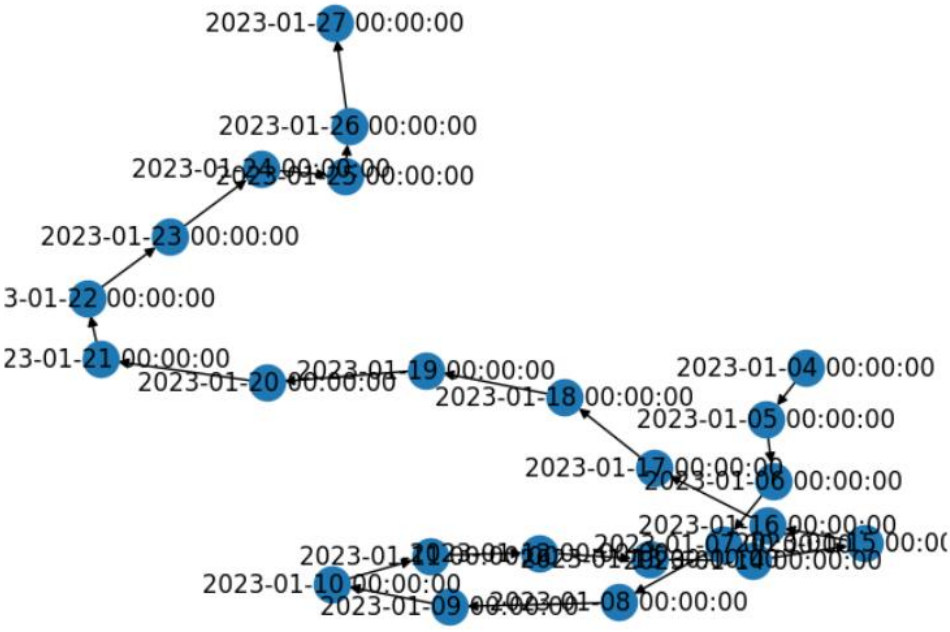


上图表示了使用滑窗法进行图时序性的数据构建



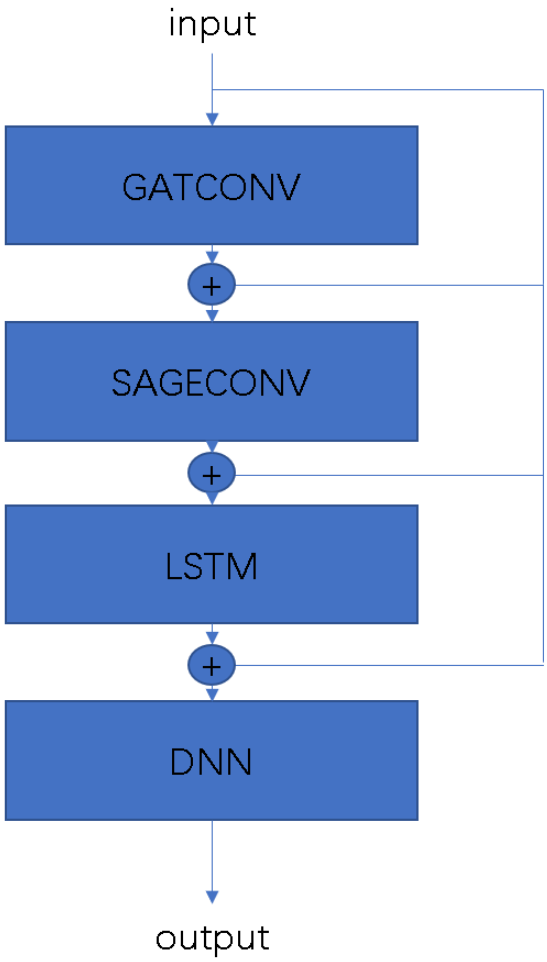
上图表明了时间 20230105 的部分节点连接情况（黑色表示相连）

通过观察单个时序图数据可视化的图可以发现，在图中空间位置相近的节点往往有较强的连接性。



上图以有向图的方向来表示时间的变化

在构建好图数据后，在建模阶段，我们选取的图空间卷积模型主要是 GAT 和 GraphSAGE，将滑窗后的 24 张图送入网络中，提取空间特征，并且在每个模块后加入残差连接。然后使用 lstm 进行时间特征提取，最后经过多个线性层进行回归预测。



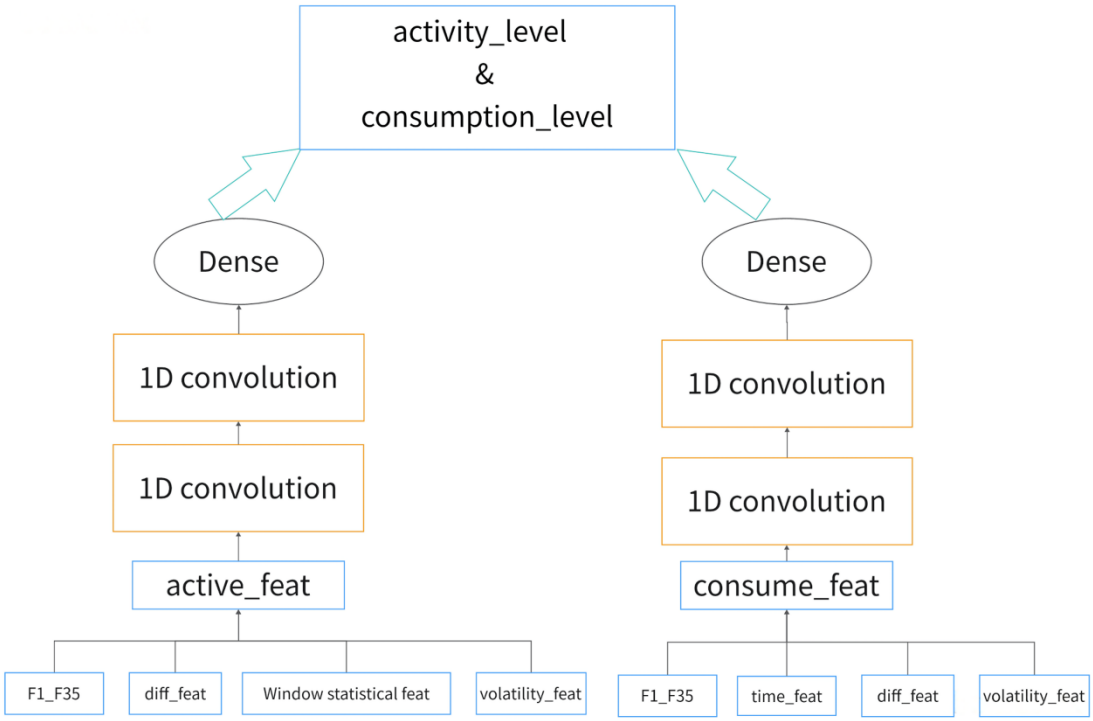
上图表示了我们图模型的主要结构

可以看见的是主要是使用 GAT 和 GraphSage 作为图的空间卷积层，LSTM 作为图的时间卷积层，同时我们加入了残差思想，可以提高模型的学习泛化能力。

2.卷积神经网络

作为本赛题第二个方案，从数据挖掘角度挖掘时序相关的特征且针对活跃指数的预测和消费指数构造不同的特征。特征工程之后构建相对轻量级模型-两层 CNN+Dense。整体特征构造方向为：常规时序特征（日，月，是否工作日，是否假期日），时序一阶差分特征，时间窗口内统计特征（min,max,mean,std），数值波动特征（日波动率，周波动率，月波动率）。最后通过特征筛选提取针对活跃指数预测提取 266 维度特征（含原始 feat），针对消费指数预测提取 172 维度特征（含原始 feat）。

下图展示算法流程。更多具体见下文二部分。



3. 多任务转单任务

可以看到的是，在本赛题中预测任务主要有两个，分别是活跃指数的预测和消费指数的预测，起初我们使用同一套模型对两个任务进行预测，但是在训练过程中发现，对两者的预测精度很难同时提升，具体表现在，当 consume 预测效果较好时，active 可能不太精确，而当 active 预测效果好时，consume 的预测效果可能就会下滑，我们队伍初步判断，可能是二者不具有较强的关联性，因此更适合两套模型，分别进行单任务学习，实践后也发现，效果有所提升。

4. 加权投票融合

我们在使用两套方案的模型进行加权融合时，需要自己去控制加权的比例，但是对于 b 榜的提交次数是有限且无法获知成绩的情况下，我们没有试错的成本，因此我们采用了比较保守的投票融合方式。即使用多个比例对两个结果进行加权融合，获得多个结果。然后对于这些结果中的每个 id，计算与其他结果中同 id 的 rmse，将与其他 id 的 rmse 计算结果之和作为该结果该 id 的投票指数。选择在该 id 中投票指数最小的结果作为最终结果中的该 id 的值。使用上述的方法不必纠结哪种加权比例最优，最终结果中可以包含多种比例的结果，而且结果保证了与所有结果的最优相似，可以尽可能大的降低比例不同带来分数抖动，是较为稳妥的融合方式。

二、具体算法：

1. 时空图神经网络：

①数据清洗：我们在观察 a 榜数据时发现，对于需要预测的四天的边集分布有训练集有较大出入，训练集平均每天的边集大小为 1w+条，而测试集平均每天边集

为 2w+，深入观察后发现出现了仅 30 天内未出现的边就占据了 1w+条，因此为了保证测试集与训练集的分布，我们对 30 天内未出现的边进行了清洗。该方法为图神经网络带了约百分点的收益。
②模型训练:模型训练时，我们发现很容易过拟合，一旦过拟合对于得分将带来大幅度的下降，因此，我们在模型中加入了 dropout 层以及 batchnorm 层防止模型过拟合。
③多尺度反向传播：因为本题是 X-Y 型时序回归赛题，与 Y-Y 时序预测不同在于我们需要提供当前时序的 X 以回归对应的 Y.所以在计算 loss 时往往使用最后一维与对应的结果计算 loss，进行反向传播，但我们发现这样的方式很容易过拟合，因此我们在计算 loss 时不止使用最后的结果，而是使用最后 n 个结果进行计算，同时我们在训练中对 n 进行调整，使得模型预测逐渐调整，带来更好的泛化能力。
④移动平均后处理：MA（MovingAverage）是处理时序的经典方法，我们将这种方法放在了后处理当中，为了防止模型的预测过于激进，我们使用模型预测的前几时刻值与当前时刻值进行一些加权的处理作为最后的当前时刻值。

2. 卷积神经网络

①特征工程：

在进行特征工程处理的时候，我们发现消费指数跟时间特征的联系更为紧密，而活跃指数跟窗口内的统计特征联系更为紧密，因此我们针对消费指数和和活跃指数分别提取了以下不同的特征。

A.针对活跃指数：

1. 原始特征：F_1～F_35。
2. 时序一阶差分特征：di = xi+1 - xi。后一个时间点的值与前一个时间点的差值。针对 F_1～F_35（除去 F_23,F_27 全 0 列）同 geohash_id 下使用 dataframe 自带 diff()方法即可提取。
3. 窗口统计特征：它用于计算一组样本在给定时间窗口内的统计特征，以捕捉数据在时间上的变化和趋势。设置窗口大小 size=4，使用 dataframe.rolling(window=size).min(),max(),mean() 。 提 取 F_1～F_35（除去 F_23,F_27 全 0 列）同 geohash_id 下的 4 天之内的最小，最大，均值特征。
4. 波动特征：计算 F_1～F_35（除去 F_23,F_27 全 0 列）在窗口大小为 4 的同 geohash_di 下数值的波动率，反映出其时间段内数值波动程度。计算公式如下:日波动率：窗口大小内样本数值的标准差。周波动率：日波动率 * sqrt(5);月波动率：日波动率*sqrt(21)。

B.针对消费指数：

1. 原始特征：F_1～F_35。
2. 常规时序特征：该日期的日，该日期的月，该日期是否工作日，该日期是否假期日。利用 datetime 数据提取，chinese_calendar 包中的 is_holiday,is_workday 提取即可。
3. 时序一阶差分特征：di = xi+1 - xi。后一个时间点的值与前一个时间点的差值。针对 F_1～F_35（除去 F_23,F_27 全 0 列）同 geohash_id 下使用 dataframe 自带 diff()方法即可提取。
4. 波动特征：计算 F_1～F_35（除去 F_23,F_27 全 0 列）在窗口大小为

4 的同 geohash_di 下数值的波动率，反映出其时间段内数值波动程度。计算公式如下:日波动率：窗口大小内样本数值的标准差。周波动率：日波动率 * sqrt(5);月波动率：日波动率*sqrt(21)。

C.数据处理：

1. 针对活跃指数提取特征形成的 nan 值 fillna 补 0；针对消费指数形成的 nan 值的样本直接 dropna 删除。

②模型训练:

1. 由于数据量少，防止 overfitting 我们采用相对轻量级的两层 1D CNN+Dense,且 xavier_uniform 初始化卷积核参数矩阵最后针对卷积层设置 L2 惩罚项正则化以最大程度减少过拟合。

2. 采用余弦退火热重启的策略动态调节学习率，让学习率周期性升高降低，以防止陷入局部最优解。

3. 针对消费指数和活跃指数，采取不同的学习率和训练轮数。

4. 特殊针对训练 activity_level 预测任务的模型采用同一个 geohash_id 同 batch 训练策略：训练集每个 geohash_id 给定的 90 个时间点→batch_size = 90。在使用 Mini-Batch 训练时，通常会计算每个 Mini-Batch 的损失函数的均值，并将该损失均值作为反向传播的起点。此策略保证一个 batch 损失为同一个 geohash_id 损失，减少不同 geohash_id 交互引起的损失扰动而增添的反向传播更新参数的不准确性。