AI 编程大作业报告
项目一：城市数据图谱

王华艺 522030910116
方旭中 522030910117

日期：July, 2023

1 引文

本项目旨在构建一个上海市的城市数据图谱，并基于该图谱搭建不同的模型进行对城市车流进行预测。

在已有项目代码的基础上，我们小组首先更换了数据降维方法，将原有的 t-SNE 方法上改为了 PCA 方法，对数据进行降维与可视化，以便对数据分布有更好的理解。在车流预测的任务上，我们共实现了 LSTM、XGBoost 和 RandomForest 三个模型，此外我们还尝试在原有图卷积神经网络 GCN 模型上使用降维后数据进行模型训练，并对训练过程的损失变化进行可视化，并与原有图卷积神经网络 GCN 模型进行对比实验，以便更好地理解并评估不同模型学习效果。

通过该项目，我们构建了一个综合的城市数据图谱，并借助该图谱构建不同模型进行车流预测，有效地分析和预测了车流量，同时也增强了我们对相关模型的理解以及编码能力。

2 方法

我们共完成了两部分任务：更换降维方法，对 POI 数据进行降维和可视化处理，并将降维提取到的特征作为一种输入，进行车流预测；更换图卷积神经网络，搭建不同模型对车流数据进行预测。

2.1 数据分析

2.1.1 直方图

为了进行数据的可视化，我们先绘制直方图对数据分布进行观察，以平均车流量和和到达点的车流量为横轴，频数为纵轴画图，观察数据的分布情况，如数据的中心位置，分散程度等。

2.1.2 PCA

除 t-SNE 外，我们还尝试了将 PCA 作为降维的工具，将车流量数据从 24 维降到 2 维，以方差解释度作为数据信息保留多少的衡量标准，以到达点的平均值作为数据点的颜色。

2.2 车流预测

2.2.1 LSTM

为了进行车流量预测，我们首先采用了 LSTM 模型。LSTM 是循环神经网络（RNN）的变体，专门用于处理序列数据。我们分别先后使用了单向 LSTM 以及双向 LSTM，并计算预测结果与真实值之间的

RMSE 和 MAE 来评估模型的预测性能。

2.2.2 XGBoost

我们接下来使用了基于梯度提升树的 XGBoost 模型。采用 XGBoost 方法进行车流量预测，能够利用梯度提升树较强的学习能力和优化策略，提高模型预测的准确性。通过计算测试集在训练好的 XGBoost 模型上的 RMSE 和 MAE，评估模型的预测性能。

2.2.3 Random Forest

我们还使用了基于集成学习的随机森林（Random Forest）模型。随机森林利用了多个决策树的集成优势，能够更好地捕捉车流量数据中的复杂关系。随机森林模型会在每个决策树上进行训练，并使用平均或投票的方式来得到最终的预测结果。在测试阶段，我们使用测试集的输入特征传递给已训练好的随机森林模型，获取预测结果，并计算 RMSE 和 MAE 以评估模型的预测性能。

3 实验

3.1 数据分析

3.1.1 直方图

根据已有文件'shanghai_graph.bin', 运行hist .py，得到直方图如图，红色为 poi 节点平均值，蓝色为到达点每小时平均值, 可以观察到：poi 数据中心相比而言更高，分散程度更高。考虑到作为预期输出的蓝色部分靠近 0 的值较多，需要在建立模型的时候考虑到梯度消失等问题。

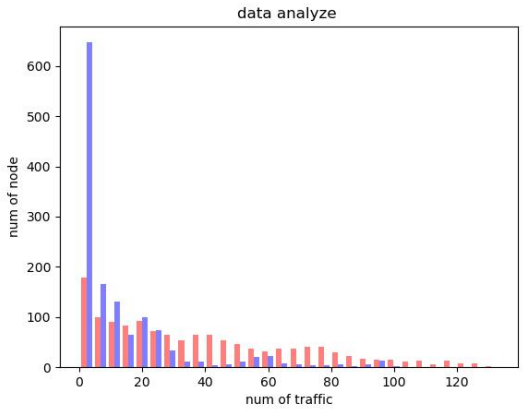

图 1: 直方图

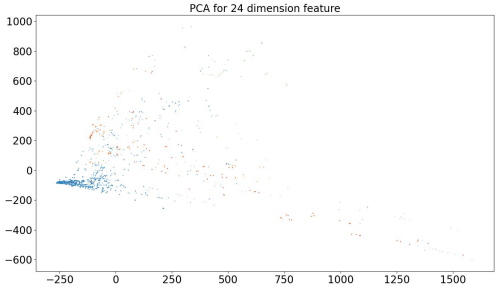

图 2: PCA

3.1.2 PCA

编写相关代码，得到 pca 降维到 2 维后图如图所示。通过pca.components_可获取降维后主成分，通过pca.explained_variance_ratio_可获取 pca 的方差解释度。根据实验结果可发现，当n_components>2 时方差解释度大于 0.95，n_components>6 时方差解释度大于 0.99，当n_components=2 时，方差解释度约为 0.94，已经保留了相当一部分的数据信息。

3.2 车流预测

3.2.1 Baseline: GCN

运行已有 GCN 模型代码进行训练。设置超参数：hidden_size1=128, hidden_size2=64, hidden_size3=24, lr=0.01, epoch=500，分别得到如下训练集损失曲线、验证集损失曲线、测试集的 RMSE、MAE:

图 3: GCN Training Loss

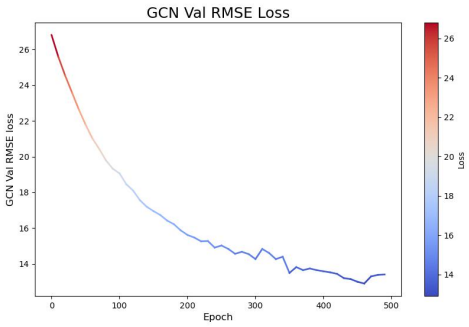

图 4: GCN Val Loss

```
GCN model:
RMSE: 17.817  MAE: 7.3802
```

图 5: GCN MAE and RMSE

3.2.2 GCN (using PCA)

将数据处理部分使用 PCA 降到 2 维后的数据作为输入，使用已有的 GCN 模型进行训练，分别得到如下训练集损失曲线、验证集损失曲线、测试集的 RMSE、MAE:

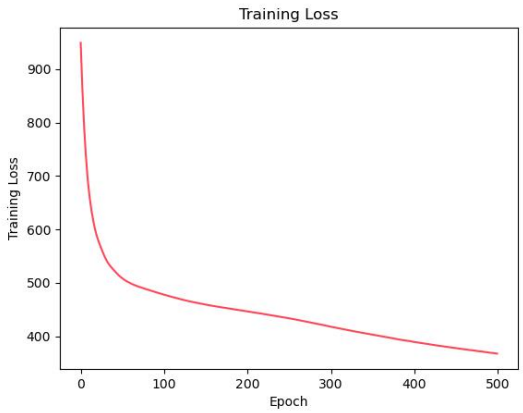

图 6: GCN (using PCA) Training Loss

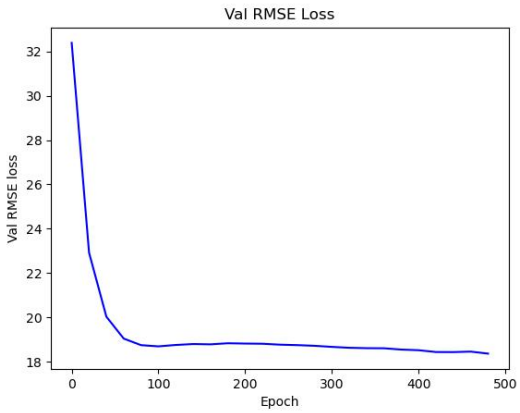

图 7: GCN (using PCA) Val Loss

```
val: 460 18.4288
RMSE: 23.4206  MAE: 9.7113
```

图 8: GCA (using PCA) MAE and RMSE

3.2.3 LSTM

已有输入是车流特征向量矩阵，表示每个路网节点在不同时间段的车流情况。为了适应 LSTM 模型的输入要求,我们将输入数据的维度由(batch_size, input_size=24)调整为(batch_size, sequence_length =1, input_size=24)的形状。在单向和双向 LSTM 模型中我们均采用了双层 LSTM 层，在模型的最后一层添加了一个线性全连接层，用于将 LSTM 的输出映射到最终的预测结果。该全连接层的输出大小为输出特征的维度，即 24 维，对应 24 个小时的车流量预测。

设置单向 LSTM 超参数：hidden_size=128, lr=0.01, epoch=500, num_layers=2，分别得到如下训练集损失曲线、验证集损失曲线、测试集的 RMSE、MAE:

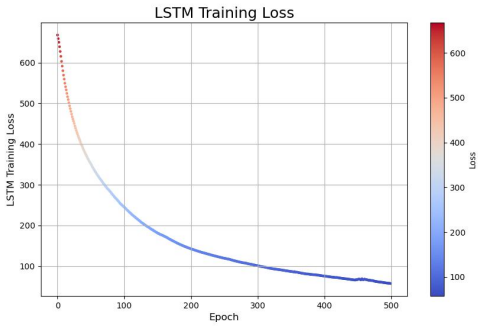

图 9: LSTM Training Loss

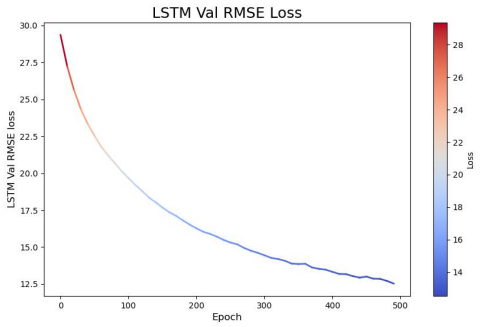

图 10: LSTM Val Loss

```
RMSE: 12.9994  MAE: 6.4365
```

图 11: LSTM MAE and RMSE

由于单向 LSTM 的学习效果与 GCN 相比提升不大，故将其替换成双向 LSTM，同样将层数设置为两层：hidden_size=128, lr=0.01, epoch=500, num_layers=2，分别得到如下训练集损失曲线、验证集损失曲线、测试集的 RMSE、MAE:

图 12: Bidirectional LSTM Training Loss

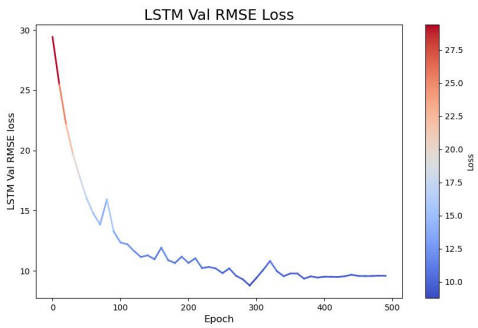

图 13: Bidirectional LSTM Val Loss

```
RMSE: 8.7719  MAE: 4.1813
```

图 14: Bidirectional LSTM MAE and RMSE

3.2.4 XGBoost

为了适应 XGBoost 的输入，我们首先将输入特征和标签转换为 XGBoost 库所需的格式（DMatrix）来训练和测试 XGBoost 模型。通过调用 XGBoost 库中的 train 函数，对 XGBoost 模型进行训练。

设置 XGBoost 超参数：'eta'：0.01，'max_depth': 8, 'subsample': 0.8，'colsample_bytree': 0.8，分别得到如下训练集损失曲线和测试集的 RMSE、MAE:

图 15: XGBoost Training Loss

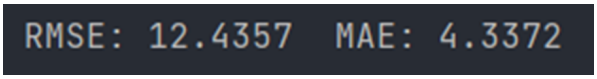

图 16: XGBoost MAE and RMSE

3.2.5 Random Forest

在训练阶段，我们将训练集的输入特征和标签传递给随机森林模型。通过调用 sklearn 库中的 RandomForestRegressor 类对模型进行训练。设置超参数：n_estimators=500, random_state=42，得到在测试集上的 RMSE、MAE:

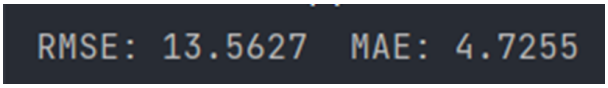

图 17: Random Forest MAE and RMSE

4 结论

根据上述实验，可以得到不同模型学习效果（MAE、RMSE）的对比表格：

表 1: 不同模型学习效果的对比

模型	RMSE	MAE
GCN	17.817	7.3802
GCN（using PCA）	23.4206	9.7113
LSTM（单向）	12.9994	6.4365
LSTM（双向）	8.7719	4.1813
XGBoost	12.4357	4.3372
Random Forest	13.5627	4.7255

由上表可以观察到，与对照模型 GCN 相比，使用 PCA 降维后的两维数据特征作为输入，学校效果略有下降，而更换的几种模型在学习效果上均好于 GCN。其中，双向 LSTM、XGBoost 模型和随机森林模型在该数据集上表现较好，相对于其他模型具有更低的 MAE 值；单向 LSTM 模型的预测性能相对较差。针对该实验结果的分析如下：

1. 使用 PCA 降维后的两维数据特征作为输入，保留了相当一部分节点特征信息，因此使用 GCN 模型学习后的预测结果与使用完整数据的预测结果相差不大。

2. GCN 是一种基于图结构的神经网络，而当前模型的输入是基于节点的特征矩阵，并不是图模型，在图的层面上节点之间的连接关系较弱，因此导致 GCN 无法充分利用节点之间的拓扑关系，从而影响性能。
3. 更改使用的三种模型（XGBoost、LSTM 和随机森林）均是充分利用了输入数据的时序性特点。其中 LSTM 模型能够捕捉时间序列中的长期依赖关系，而单向 LSTM 仅能沿一个方向处理序列，双向 LSTM 同时考虑序列数据的前向和后向信息，能够更全面地捕捉到序列中的依赖关系，因此提高了预测性。XGBoost 和随机森林两种模型都是基于集成学习的方法，能够通过组合多个决策树来提高预测性能，可能能够更好地适应输入数据的时序性的特点，学习效果也较好。
4. 由上述实验可发现，在该实验中，输入数据的时序性特征占主要地位，而图结构特征并不明显。

5 个人贡献

5.1 方旭中

1. 数据分析部分：编写基本代码，分析 POI 节点与时序的数据分布，将数据进行可视化，并进行相关分析。
2. 建立模型部分: 将 PCA 降维获取的数据作为输入，实现 GCN(using PCA)，修改部分原有参数。
3. PPT 及报告相关部分的撰写。

5.2 王华艺

1. 模型预测部分：更换预测模型，分别实现单双向 LSTM、XGBoost 和随机森林模型，以及相关损失函数的可视化。
2. 进行模型学习效果的对比及结果分析。
3. PPT 及报告相关部分的撰写。