PRP 论文草稿

绪论

交通拥堵是城市交通中非常常见的问题。城市化的推进和民众生活水平的提高同时也带来了城市车辆密度的持续上升，交通拥堵也因此愈发常见，影响民众的出行质量。应此需求，短期交通拥堵预测算法近年来在国内外蓬勃发展，目前可分为三类：概率推理模型、浅神经网络学习、深度学习。

在交通领域，数据分析工作经常涉及到海量的半结构化或非结构化的数据，比如车辆行程信息、路网流量等等。这些数据的因变量不仅有着复杂的时空背景，同时往往数目巨大。传统的聚类分析在处理这种复杂的大数据问题上存在难度，但对于经典的概率统计模型和浅神经网络，聚类分析是数据预处理的必要步骤。同时，层数较浅的机器学习模型也很难达到期望的精度。因此，为了更好的捕捉数据间复杂的特征，就需要假设空间更大、复杂度更高的模型。深度学习算法应运而生。深度学习算法的核心是多个进行线性运算的隐藏层彼此相连，从而解决非线性问题。不同结构的隐藏层被用于提取数据蕴含的不同种类的信息。相较于浅神经网络，深度学习算法可以在对数据无任何事先了解的情况下提取出其中的特征。而对于隐变量错综复杂的交通预测问题，深度学习对数据特征的敏感性吸引越来越多的研究使用它作为预测算法的核心而卷积神经网络（CNN）就是深度学习最具代表性的结构之一。

考虑到交通信息在宏观上具有一定的周期性，收到[]的研究的启发，我们尝试借助CNN来实现交通拥堵的大范围预测。CNN在提取二维数据的特征上具有显著优势，而一个交通网络上的信息同样可由经度和维度两个维度编号，因此我们在预处理中选择将同一时间的交通信息转换成二维矩阵。同时，CNN的每个隐藏层都可以同时处理来自多个通道的二维矩阵，并提取它们所包含的共有特征。这一特性启发我们将间隔一定周期的数据矩阵叠加在一起，从多个通道一起输入神经网络，从而尝试让模型记忆并预测出交通拥堵的周期性特征。

在本研究中，我们探索了通过合理的交通数据预处理和CNN的结构设计，测试CNN对大范围交通数据的预测性能，并进行可视化展示。所有的分析和计算都借助Python语言完成，其中，代码库Pandas和Keras分别用于进行数据的预处理和CNN的搭建、使用。数据可视化的部分借助Kepler.gl的接口来实现。

研究内容和方法

1. 问题定义以及重要参数

1.1 拥堵状态的区分

本研究中，我们以一个空间网格内的平均速度作为拥堵状态的表征。网格速度到拥堵等级的映射参考[]，并在表[]中给出

1.2 问题定义

在本研究中，我们对短期大范围交通拥堵预测问题的定义如下：给定一个地图网格和一个网格速度的序列{},i=0,1,2,…,m, j=0,1,2,…,n, t=0,1,…,T-1。其中，i和j分别表示数据所在网格的行索引与列索引，t是数据的时间索引。我们的目标是预测所有网格在时间点T的网格速度,i=0,1,2,…,m, j=0,1,2,…,n。

1. 数据预处理

2.1 数据来源与样式

本研究所用的数据来自于滴滴盖亚计划的开源数据，内容是2016年11月成都市中心的所有网约车订单数据。原始数据的样式如图[]所示，包括网约车的司机ID，订单ID，Unix时间戳，网约车在GCJ-02坐标系下的经度、维度，以及订单的日期。

2.2 数据清洗和转换

2.3 数据特征提取

2.4 邻域连贯性和周期性分析

2.4.1 邻域连贯性

一个交通网格在某一时刻的网格速度与其在空间和时间上邻近的网格的差距在一定的距离范围内可视为是连贯的。这是因为大部分时间段的交通状况很少出现突变的情况。因此我们认为，同时考虑一定时空邻域内的所有数据对于预测某个时空网格的数据是有帮助的。（需要数据分析）为了测试所用数据的连贯性，我们挑选了相隔较远的五个网格，并分别分析了它们的数据序列与邻近网格的数据序列的Pearson系数。如图[],[]所示（需要配图），结果...

2.4.2 周期性

宏观上看，不同工作日之间的网格数据会表现出一定的周期性。间隔24小时的数据常表现出相似性。（需要数据分析+配图）图[]展示了一个网格的网约车流量在30天内的涨落变化，表现出了明显的周期性。因此，考虑某一网格在特定几个周期前的数据可能会有助于提升预测的精度。

1. 深度卷积神经网络的构建

3.1 概述

此部分首先介绍CNN的提出和适用性，然后会从输入层、CNN结构以及模型性能评估标准三个方面描述研究模型的构建。

3.2 CNN简介

CNN是受到生物学概念“感受野”的启发而提出的。生物学中，一个视神经的“感受野”指的是视网膜上能够激活该神经元的区域。CNN常用于处理二维的网格数据，但是只要合理地重构数据，高维数据同样可以输入CNN进行分析。CNN在捕捉二维数据的局部特征上具有巨大的优势。在本项目中，整个交通网络在同一时刻的数据即可视为一个二维的速度矩阵，适合作为CNN的处理对象。

3.3 输入层

考虑到交通数据的邻域连贯性和周期性，我们采用数据裁剪来结合各个网格与它们空间上的邻域的数据；同时，每一个时刻的速度矩阵都会和与它在时间上相邻的速度矩阵从多个通道同时送入CNN进行处理，从而使CNN捕捉到数据在时空上的连贯性和周期性。

3.3.1 数据裁剪

数据裁剪是最常用的数据增强算法之一。其中心思想在于对待处理的大数据集进行多次局部取样，并将全部的取样结果作为下一步输入。在数据集较小的情况下，此方法可以扩充样本数，提升训练效果。经过充分的合理取样，就可以在不丢失原数据的情况下，增强局部数据的关联性，从而让后续的神经网络更好地捕捉原数据的局部特征。

在当前问题背景下，单个网格与其邻域网格数据的连贯性便是一种局部特征。本项目的研究区域是一个15×15的网格。我们采用长宽均为10的正方形滑动窗口对空间网格进行裁剪，是数据量增加到原来的6×6=36倍。裁剪的具体方式如图所示（需要配图）

3.3.2 多通道

对于待预测时刻t，我们选取了其过去一、二、三天（t-24, t-48, t-72）和过去一、二、三小时（t-1, t-2, t-3）的速度网格作为输入的6个通道，并将数据整理成标准输入格式：样本×通道×宽×高。输入的结构如图[]所示。（需要配图）

3.4 CNN的结构

在CNN的架构设计上，我们沿用由LeCun[]等设计的“卷积-池化”结构。结构如图[]所示（需要配图）考虑到卷积和池化操作可能会使特征图的尺寸缩小，因此我们选择在全连接层增加一个上采样层来还原特征图的尺寸。

3.5 优化算法和目标函数

模型使用Adam优化器进行优化，学习率设置为0.0004，并采用预测值与观测值之间的均方误差作为目标函数，即