

机器学习导论

大作业：交通拥堵指数预测

高辰潇 (181220014), 人工智能学院,

王傲然 (181220055), 人工智能学院,

孔锐 (181220026), 人工智能学院

榜单成绩: 0.1217

2020 年 6 月 20 日

1. 摘要

本文提出交通拥堵指数 (TTI) 具有时间局部性和周期性两个基本性质, 并在借鉴贝叶斯网络的相关概念后, 使用隐变量和周期变量对 TTI 预测问题进行建模和形式上的描述。

基于以上分析, 我们使用双层门控递归单元 (GRU) 神经网络来对深圳北站附近 12 条道路在 2019 年 12 月 21 日至 2020 年 1 月 1 日短期时间段内的交通拥堵指数进行了预测, 在验证集和测试集上取得了良好的预测效果, 性能优于其他时序预测模型。

2. 介绍

在本次实验任务中, 我们需要根据给定的数据集训练模型, 对未来时段的道路交通拥堵指数 (TTI) 进行预测。我们认为, 由于 TTI 指数具有良好的周期性和时间局部性, TTI 指数的预测任务可以建模成时间序列数据的预测任务。预测时间序列数据是交通规划, 商业管理和金融学中的重要问题, 并且已成为智慧城市的重要组成部分之一。作为时间序列预测问题之一的交通信息预测, 尤其是短期交通信息预测, 受到了城市规划者、商业公司和出行者的密切关注。

目前, 存在很多传统机器学习方法可以有效地预测时间序列数据。其中包括单变量自回归 (AR), 单变量移动平均值 (MA), 简单指数平滑 (SES), 还有效果更加出色的自回归移动平均值 (ARMA)。另外还有单变量“自回归综合移动平均值 (ARIMA)”模型, 它作为 ARMA 的一种特殊类型, 不仅结合了单变量自回归和移动平均值模型, 还在模型中考虑了差分因素。其中, ARIMA 模型已经被证明了, 在预测时间方面的准确性表现良好。但是, 这些方法主要使用线性模型和浅层机器学习模型来预测传入的交通流量, 不能很好地描述非线性和不确定性。

随着计算能力的发展提高以及更高级的机器学习算法和方法 (例如深度学习) 的发展, 出现了新的算法来预测时间序列数据。近年来, 基于支持向量机 (SVM) 和随机森林 (RF) 的传统机器学习算法, 以及基于递归神经网络 (RNN) 和长期短期记忆 (LSTM) 的深度学习的算法在时间序列预测问题中被广泛使用。这些深度学习方法能够更好地识别数据的结构和模式, 例如时间序列预测中的非线性和不确定性。

特别地，在交通信息预测这一时间序列问题中，基于深度学习的 NN 方法已经被证明了优于其他方法 [1][2]：在这些基于深度学习的方法中，LSTM NN 和堆叠式自动编码器 (SAE) 的性能要优于某些常用的传统预测模型。在交通速度预测方面，Ma 等人将 LSTM NN 和 SAE 应用于预测问题，发现 LSTM NN 比 SAE 具有更好的性能 [3]。一些文献 [4] 中比较了 ARIMA, LSTM NN 和 GRU NN 模型在交通流量预测任务中的性能，发现了 LSTM 和 GRU NN 的性能显著优于 ARIMA。同时发现了 GRU NN 的性能比 LSTM NN 略好，并且收敛速度通常比 LSTM 更快。

在本项目中，我们小组的成员在对 TTI 指数预测任务进行深入思考和建模后，尝试使用带有两层 GRU 网络层的神经网络模型对 TTI 指数进行预测。我们在验证集上对训练得到的网络模型进行了测试，表现良好，并在在线榜单的评测中取得了 MAE=0.1217 的成绩。

本报告的结构安排如下：在第三小节中，我们使用对 TTI 的两个性质对任务进行了初步建模和数学描述；在第四小节中，我们对本次实验所使用的算法进行了原理层面的介绍和推导；在第五小节，我们首先对实验所采用的网络模型进行了简要介绍，然后详细阐述了观察变量提取过程、训练集构造过程和网络模型的训练过程；在第六小节，我们给出了模型的预测性能，并与其他学习器进行了简单对比；在第七小节，我们就本次实验未能解决的问题进行了简单讨论。

3. 问题描述与建模

在本次实验中，我们假设交通拥堵指数 (TTI) 满足以下两个性质：

- 时间局部性：在任意时刻，同一路段上的 TTI 指数一定与之前一段时间的道路交通状况有关，而道路交通状况可由行驶车辆的平均速度或者 TTI 等指标反映。
- 周期性：对于同一条道路，TTI 指数一方面呈现出以 24 小时为周期的周期性变化，另一方面也呈现出以 7 天为周期的周期性变化

基于性质 1，我们决定建立一个时序回归模型来描述本次任务。为了问题描述上的方便，我们将时间划分成为以 10 分钟为间隔的区间，每个区间称作一个“时间块” (time block)。对于第 i 个时间区间，我们定义道路的交通状况由一个状态变量 s_i 表示。

考虑到时间局部性，道路的交通状况 s_i 同时也会收到之前时间段的交通状况的影响。在本次任务中，我们假设 s_i, s_{i+1}, s_{i+2} 同时受到 $s_{i-6:i-1} = (s_{i-6}, s_{i-5}, \dots, s_{i-1})$ 序列的影响。

基于性质 2，我们不妨把 TTI 的周期性看作是道路交通状况的周期性，也就是状态变量 s_i 同时也受到周期信息 e_i 的影响。本次任务中我们提出了两种周期性：交通状况以 24 小时和一周为周期波动，这也同我们生活中的认知保持一致。因此我们将周期信息抽象成了 $e_i = (t_i, d_i, w_i)$ ，其中 t_i 表示当前时段在 24 小时周期中的位置， d_i 表示当前时段的日期是星期几， w_i 为 0-1 指示变量，指示当前时段是否在周末。

任务的目标便是预测确定时刻 i 的 TTI 指数，我们用 tti_i 来表示。则任务可抽象为找到后验概率分布

$$P(tti_{i:i+2} | e_{i:i+2}, s_{i-6:i-1})$$

注意到，道路交通状况无法直接观察，属于隐变量。因此我们从所给的数据中进行了特征提取，提取出的指标可被看作是对隐变量的观察变量 $o_i = (v, TTI, \hat{u}, r_{low})$ 。其中 v 表示在第 i

个 time block 中路段上的平均速度， \hat{u} 表示对应时间路段网约车的平均速度， r_{low} 表示对应时段路段低速网约车的比例。使用观察变量代替状态变量，我们的任务变为：

$$P(tti_{i:i+2} | e_{i:i+2}, o_{i-6:i-1})$$

从实际角度，我们希望找到满足下述条件的学习器 \mathcal{L} ：

$$\mathcal{L}(e_{i:i+2}, o_{i-6:i-1}) \simeq tti_{i:i+2}$$

以上就是我们基于贝叶斯网的相关知识，对 TTI 预测任务的简单描述。

4. 基本理论

我们使用了基于 Gate Recurrent Unit 的循环神经网络解决本次的预测问题。循环神经网络 (RNN) 的基本任务形式是在输入一整个序列的输入向量 X 以后通过循环神经网络得到输出 O 。再通过反向传播来优化 O 和目标 Y 之间的 loss。具体来说，RNN 的结构如图 (1) 所示

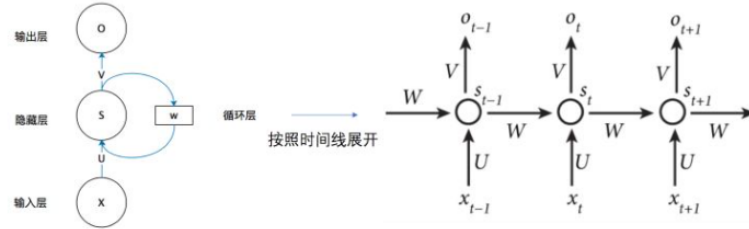


图 1: RNN 网络示意

其中左图是 RNN 的整体示意图， X 是输入向量， S 是隐藏层向量， U 是输入层到隐藏层的权重矩阵， V 是隐藏层到输出层的权重， O 是输出向量。 W 是当前时刻隐藏层向量到下一时刻隐藏层向量的权重。右边是 RNN 的最小单位，RNN cell。具体来说，每一层 RNN 由一个 RNN cell 组成。 x_i 是输入向量 X 的一维，对应输入序列第 i 个时刻的向量，RNN cell 在 i 时刻接收到输入 x_i ，输出 o_i 的同时将 s_i 输出给 $t+1$ 时刻的自己。即

$$O_t = g(V \cdot S_t) \quad (1)$$

$$S_t = f(U \cdot X_t + W \cdot S_{t-1}) \quad (2)$$

RNN 内部的 f 与 g 分别使用的是隐藏层的激活函数和输出层的变换函数，在反向传播的时候不但需要和一般的神经网络一样将梯度从输入到输出之间的参数传播，即按照空间结构传播，还需要在时间通道上的不同隐变量状态之间传播，即时间通道传播。我们令最终的损失函

数为 $L = \sum_{t=1}^n L_t(o_t, y_t)$, 则有

$$\frac{\partial L}{\partial V} = \sum_{t=1}^n \left(\frac{\partial L_t}{\partial o_t} \odot g(V s_t) \right) s_t^T \quad (3)$$

$$\frac{\partial L_t}{\partial U} = \sum_{k=1}^t \frac{\partial L_t}{\partial f(s_k)} x_k^T \quad (4)$$

$$\frac{\partial L_t}{\partial W} = \sum_{k=1}^t \frac{\partial L_t}{\partial f(s_k)} s_{k-1}^T \quad (5)$$

这种 BP 算法又被称为 BPTT(Back Propagation Through Time)。RNN 的最终输出既有每一步输出的序列, 同时也有最后一个隐变量状态。在我们的网络中, 我们选择对最后的隐变量状态进行解码, 在此基础上我们就可以定义优化器进行模型的参数学习。当然, 上述普通 RNN 还存在一个非常重要的问题, 在计算 $\frac{\partial L_t}{\partial f(s_k)}$ 的时候由链式法则会递归计算 $\frac{\partial s_{k+1}}{\partial s_k}$, 此时最终的结果将由 W 决定, W 主特征值大于 1 则梯度爆炸, 反之梯度消失。这意味着我们几乎不可能利用普通的 RNN 学到很长时间的周期性或者依赖性。

GRU 的内部结构同普通 RNN 相比存在一些不同, 其结构如图 (2) 所示:

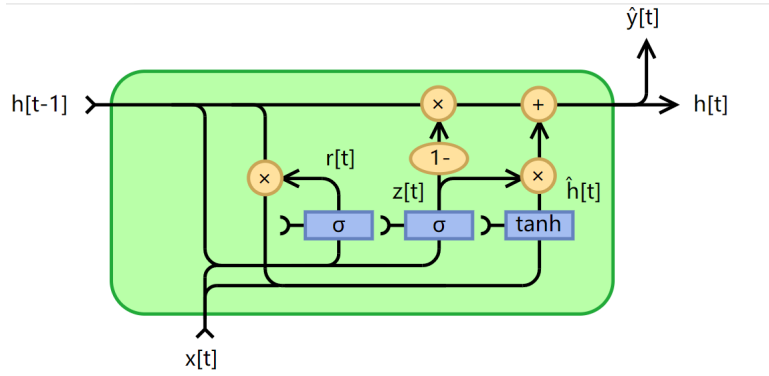


图 2: GRU 网络示意

GRU 中具体的前向传播公式为:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \quad (6)$$

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \quad (7)$$

$$\bar{h}_t = \tanh(W_h x_t + U_h (r_t \odot h_{t-1}) + b_h) \quad (8)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \bar{h}_t \quad (9)$$

这其中 r_t, z_t, \bar{h}_t 分别被称为重置门、更新门、当前记忆门 (reset gate、update gate、current memory gate), 用来控制网络对长短期记忆的存储。重置门决定了遗忘多少过去的记忆, 更新门决定了需要传递多少过去的记忆到下一时刻。因而 GRU unit 当中算法流程如下:

- 获得当前时刻的输入向量 x_t 和之前时刻传递的隐变量 h_{t-1}
- 利用三个门各自的参数和更新公式计算 r_t, z_t, \bar{h}_t 的值

- 对重置门和当前记忆门做哈达玛积加上更新门和 h_{t-1} 的哈达玛积然后输出

相比之前的普通 RNN，GRU 的结构规避了之前在时间通道上传播导致的梯度指数衰减的问题，从而使得长序列的训练变得可能。回顾之前 RNN 在反向传播过程中的问题，最大的难点在于 $\frac{\partial s_{k+1}}{\partial s_k}$ 每一项都存在 W 导致最终梯度消失或者梯度爆炸，也就是共享了常量值，但现在在更新选项是通过不同门来控制参数，使得梯度不一定会收敛到 0 或者无穷

$$\begin{aligned}\frac{\partial h_{k+1}}{\partial h_k} &= \frac{\partial(z_t \odot h_{t-1} + (1 - z_t) \odot \bar{h}_t)}{\partial h_k} \\ &= \sigma'(\cdot)U_z \odot h_{t-1} + z_t \\ &\quad - \sigma'(\cdot)U_z \odot \bar{h}_t + (z_t) \odot \tanh'(\cdot)U_h(\sigma'(\cdot)U_r \odot h_{t-1} + r_z)\end{aligned}\quad (10)$$

上式的累乘并不会导致普通 RNN 梯度反向传播带来的问题。

实际的使用过程中，还可以通过双向 GRU 实现双向传播，即将序列信息按照正序和反序分别输入网络，对得到的隐向量进行拼接从而得到前向和后向的信息。得到的隐变量可以通过输入 mlp，完成对隐变量的解码。因而整个过程可以看成是利用 GRU 得到指定时刻和路段的隐变量状态，再由 mlp 完成从相应的隐变量输出到对应的观察变量空间当中。最终完成对整个过程的追踪，完成端到端的训练。

5. 方法

正如前文所述，我们选择了带有两层 GRU 网络层的神经网络作为模型解决此问题。网络模型如图 (3) 所示。

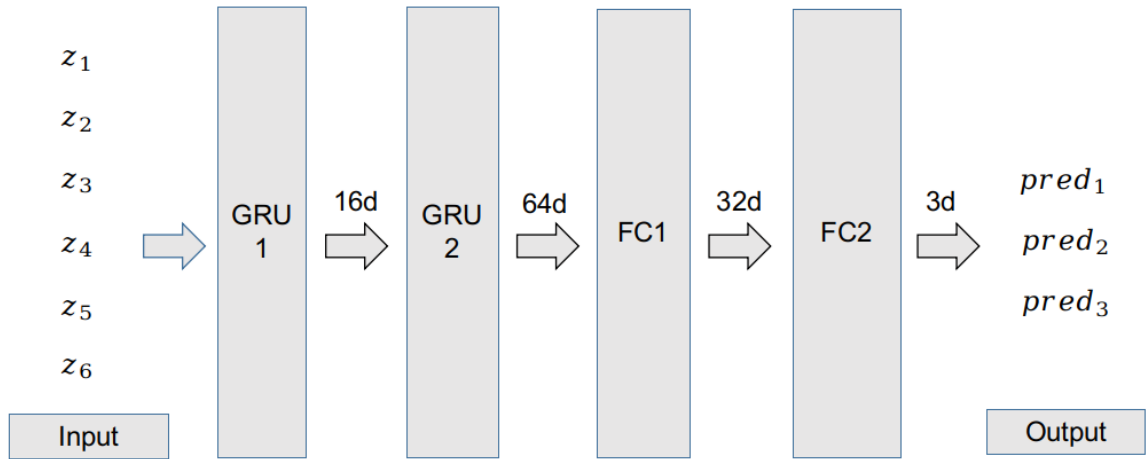


图 3: 网络模型

具体而言，网络接收一个长度为 6 的特征向量序列 $\mathbf{x} = (z_1, z_2, \dots, z_6)$ ，先后通过两层 GRU 网络层和两层全连接网络层，最终输出一个三元组，分别表示对十分钟后 TTI、20 分钟后的 TTI 以及 30 分钟后的 TTI 的预测值。

由于我们训练集输入向量等构造方式略为复杂，因此本部分接下来的内容将分为三个小节，分别对数据集处理与观察变量提取、训练集构造和模型训练三个部分进行具体阐述。

数据集处理与观察变量提取

前文提到，TTI 指数是道路交通状况一种指标，然而道路交通状况本质上是隐藏的状态变量，无法直接进行观测。但是我们可以通过观察例如车辆的平均速度、速度方差等指标，来对隐藏的状态变量进行合理的估计和“推断”。根据给定的数据集，我们决定通过对 GPS 数据的分析来做到这一点。网约车作为交通车流量的组成部分之一，其行驶速度、空间分布等情况一定程度上反映了现场的真实交通状况，可被视作观察变量。然而在本项目中，主办方给出的 GPS 数据太过原始、规模庞大，因此需要对 GPS 数据等作较多的预处理以提取有用的信息。我们的特征提取过程分为两个步骤：数据清洗和观察变量提取。

数据清洗：观察数据可发现，在每辆车的行驶轨迹中有大量采样点并不位于题目中提及的 12 个路段上，因此需对数据集进行过滤和清洗，去除其余路段上的 GPS 记录。我们为 12 条路段分别生成了 12 个 Bounding Box，然后基于 Bounding Box 的顶点坐标和向量叉乘来进行判定。如图 (4) 所示，设路段 i 的 Bounding Box 顶点分别为 A_1, A_2, A_3, A_4 ，待判断的 GPS 位置为 G 。如果点 G 位于 $A_1 A_2 A_3 A_4$ 框定的 Bounding Box 内，则沿着顺时针方向， G 与各顶点构成的向量叉乘结果均应指向纸面内部。因此判断函数可写作式 (11)

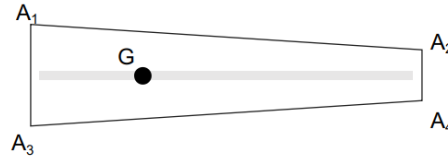


图 4: Bounding Box 示例

$$f_i(G) = \begin{cases} True & \text{if } \overrightarrow{A_i G} \times \overrightarrow{A_{i+1(mod\ 4)} G} \text{ points into the plane for } i \in [1, 2, 3, 4]. \\ False & \text{otherwise.} \end{cases} \quad (11)$$

观察变量提取：数据清洗完成后，我们进一步从筛选后的记录中提取 GPS 信息作为对交通状况的观察。在选择观察变量时，我们迁移了 **Pearson 相关性系数**这一常用的评估指标，选择与 TTI 指数的 Pearson 相关系数更大的指标作为观察变量。

通过 Pearson 相关性分析我们淘汰了诸如方差、高速车辆占比等指标，并最终选择平均速度 speed、网约车均速 avg 以及低速网约车占比 low_ratio 作为观察变量，它们的相关性系数如表 (1) 所示。

我们选取了四条道路，绘制了 TTI 以及以上三个指标在 24 小时的时间跨度内的变化趋势，如图 (5) 所示。可见，TTI 指数与上述三个观察变量的变化趋势有明显的相关性，与 Pearson 分析的结果一致。

训练集构造

前文提到，我们本次搭建的模型基于两点假设：时间局部性和周期性。时间的局部性意味着我们可以用之前一段时间的道路交通状况观察来对 TTI 作出估计；而周期性意味着可根据 TTI

表 1: 相关性系数

	TTI	speed	avg	low_ratio
TTI	1.000000	-0.990076	-0.707588	0.369361
speed	-0.990076	1.000000	0.743729	-0.371946
avg	-0.707588	0.743729	1.000000	-0.733392
low_ratio	0.369361	-0.371946	-0.733392	1.000000

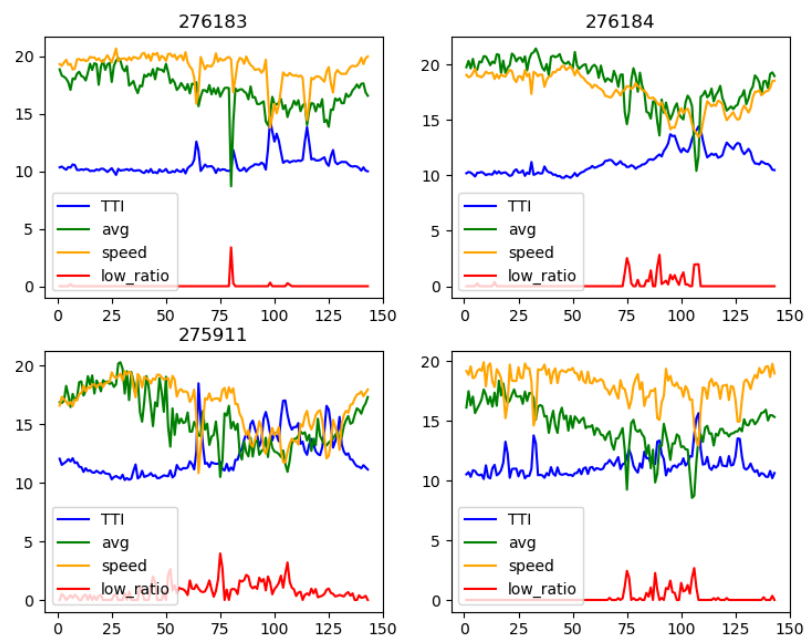


图 5: TTI 指数与观察变量变化趋势

随时段分布的特征来进行预测。在 5.1 节中，我们决定通过 TTI、speed、avg 以及 low_ratio 等观察变量作为对道路状况这个隐变量的估计；另一方面，对于周期性，我们小组选择从两个方面入手，用 time_block、weekday 和 is_weekend 分别描述待预测时段在 24 小时周期和 7 日周期中的位置。

基于以上思想，设给定时段、路段下，每一条数据 z 为一个七维向量，其结构为

$$z = (\text{TTI}, \text{speed}, \text{avg}, \text{low_ratio}, \text{time_block}, \text{weekday}, \text{is_weekend})$$

- TTI: 给定时段后，指定路段的拥堵指数
- speed: 给定时段后，指定路段的车辆平均速度
- avg: 给定时段后，指定路段上网约车的平均速度
- time_block: 该时段在一天 24 小时中的位置。我们以十分钟的间隔划分一天的时间为 144 个时间块，使用给定时段在 144 个时间块中的位置来编码此特征
- weekday: 待预测日期是星期几。取值范围为 0 ~ 6
- is_weekend: 指示变量，待预测日期是否为周末

由于该问题是序列决策问题，因此使用长度为 6 的滑动窗口在上述数据上进行滑动，每次选择六个相邻的数据记录 $z_i, z_{i+1}, \dots, z_{i+5}$ 组成网络的输入向量

$$x_i = (z_i, z_{i+1}, \dots, z_{i+5})$$

训练目标 (label) y_i 则为之后 30 分钟的 3 个 TTI 指数

$$y_i = (\text{TTI}_{i+6}, \text{TTI}_{i+7}, \text{TTI}_{i+8})$$

如此便可通过 train_TTI.csv 以及前文中提取的观察变量构造出神经网络的训练集。

模型训练

考虑到不同道路在“限速”等其他指标上的区别，我们为 12 条道路分别训练了 12 个网络模型。具体而言，网络接收一个长度为 6 的六维向量序列作为输入，通过两层 GRU 网络层后，再通过两层输出节点数分别为 32 和 3 的全连接网络层，最终得到输出三元组。训练过程中我们选用 L1 loss 作为优化目标，并进行误差逆传播，最终得到模型参数并保存。

6. 数值结果

实验数据：本次项目的全部数值模拟实验基于本次比赛主办方提供的数据集来完成。数值模拟实验基于深圳市深圳北站周边 12 条道路 2019 年 1-3 月，10-12 月道路拥堵信息和网约车数据集来完成，实验目的是根据给出 2019 年 12.26-2020 年 1.1 时间段中前 1 小时数据片段预测 30 分钟内各道路的拥堵指数。实验采用的数据集时段是从 7:30 到 22:00 数据收集比较齐全的时段。

模型参数：我们的模型中 batch size 设定为 1024，优化器采用 adam，学习率 0.001。数据集按照 4:1 的比例切分成了训练集和测试集。

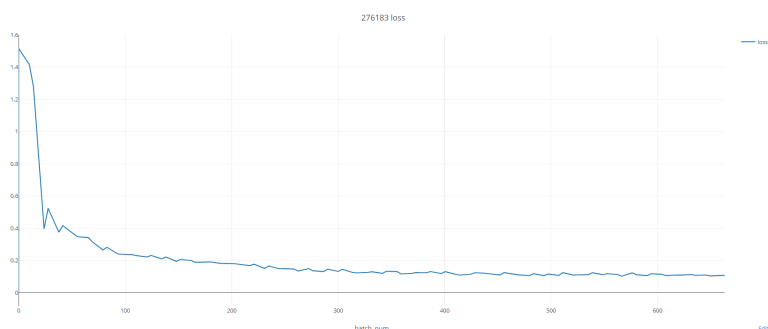


图 6: 276183 训练 loss 下降过程

实验结果：图 (6) 是实验当中在 276183 道路上的训练情况。

从训练图像当中我们可以看出模型的收敛速度较快且稳定性相对较强。

我们对数据集进行了训练集和验证集的拆分，下表 (2) 展示了模型在验证集上的实验结果。

表 2: 各路段验证集 loss

道路 ID	验证集 loss
276183	0.10243441394692683
276184	0.114871864173798
275911	0.11545990776191613
275912	0.10116003271173915
276240	0.0733472858087745
276241	0.07431479449369421
276264	0.06294352673487191
276265	0.12915612610109978
276268	0.04404238194405445
276269	0.06976138504983204
276737	0.17498463812766024
276738	0.4944802649580975

在 276265、276737、276738 道路上拥堵指数变化更加频繁，拟合难度更大，验证集上的 loss 差别同我们对数据的观察结论相同。在其他路段上，均展现出了相当好的拟合能力。

为了更加直观地观察模型对道路拥堵指数的拟合程度，我们利用模型按照时间顺序预测值同拥堵指数的真实值的折线图进行了检验，图 (7)(8) 展现了细节部分的拟合情况。图中的横轴为 time block，纵轴为 TTI，每 144 个 time block 即代表一周。

细节来看拟合情况相对较好，模型对周期性规律的捕捉相对准确，且由于卓有成效的特征抽取，TTI 在横轴上的延迟现象并不严重，仅有部分峰值拟合不尽人意。如下图 (9) 所示，从总体来看，模型对极端值的拟合情况并没有达到预期情况，在很多峰值处拟合较差。

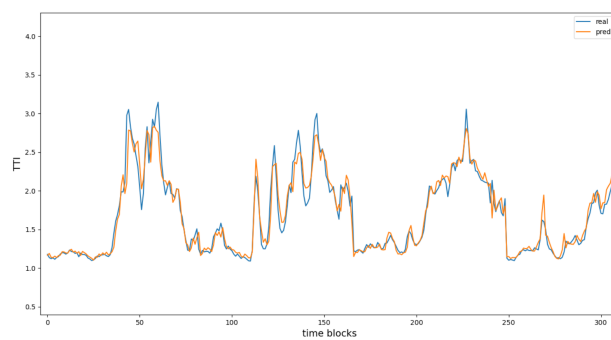


图 7: 276183 TTI 拟合情况

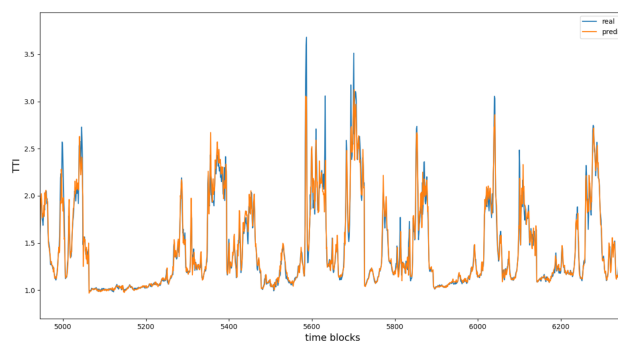


图 8: 276241 TTI 拟合情况

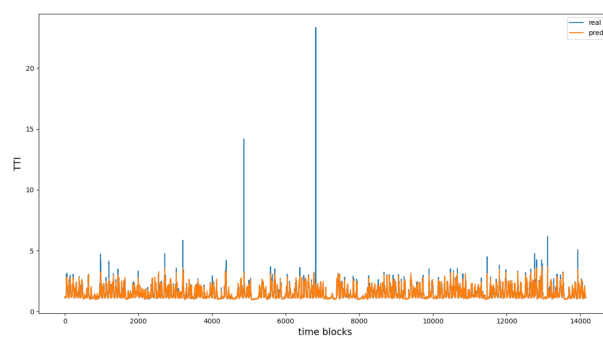


图 9: 236265 TTI 拟合情况

模型对比

我们先后搭建了三种模型：ARIMA, Prophet 以及本报告中所提到的神经网络模型。其中，神经网络模型的表现最好，与其他模型相比有着显而易见的提升，如表 (3) 所示。

表 3: 模型性能对比

算法模型	测试集性能
GRU	0.1217
ARIMA	0.139
XGboost	0.1315

7. 讨论

观察数据集我们可以发现，细节来看拟合情况相对较好，仅有部分峰值拟合不尽人意。在实验早期为了解决这一问题，我们尝试过使用 MSE 误差代替 L1 Loss 进行训练的思路。因为 MSE 误差对大的预测偏差有较大的惩罚，我们预想在这种训练偏好下模型会在峰值表现上更加出色。然而通过实验我们发现，尽管模型的确在峰值拟合上有所进步，但是它在其余位置上的表现有所下降，具体表现为幅度剧烈的波动。因此我们最终选择了 L1 loss 作为优化目标，并进行误差逆传播，最终得到模型参数并保存。

另一个值得讨论的细节在于模型设计。本次任务要求预测连续 30 分钟的 TTI 指数，这便衍生出两种思路：一种是自举 (bootstrap)，即先预测十分钟的 TTI 指数，然后将预测值作为输入进一步预测下一个十分钟的 TTI；另一种思路是不自举，即直接使用 60 分钟的输入数据预测 3 个 TTI 指数。这两种思路我们在神经网络模型以及 XGBoost 模型上都进行了实验，测试集上 loss 如表格 (4) 所示

表 4: Bootstrap 对比

	Bootstrap	Non-Bootstrap
GRU NN	0.1238	0.1217
XGBoost	0.1932	0.1307

我们发现，无论是我们最后采用的 GRU 网络模型还是 XGBoost 模型，不自举的思路表现更优。

8. 结论

基于对 TTI 指数的时间局部性和周期性假设，我们尝试使用带有 GRU 网络层的循环神经网络对未来时段的 TTI 指数进行了预测。在实验过程中，我们充分利用了 TTI 数据和 GPS 数据，利用 Pearson 分析从中提取出关于道路交通状况的描述变量和指标，并将其与周期信息相

结合，构成网络模型的输入。实验结果表明，我们所提出的网络模型能对大部分情况下的 TTI 变化趋势作出较好预测，但是在波动剧烈或者出现极端值的情况下，模型的拟合性能有所局限。

我们认为本项目的改进方向有：1. 基于 TTI 的分布对数据集进行重采样，通过加权采样方式提升网络对极端值的预测能力；2. 调整网络结构和参数，使模型具备更强的学习能力；3. 重新组合输入特征维度构成更高阶、更具表达性的特征作为网络的输入。很遗憾，受制于时间和算力，我们并未能在上述改进方向中进行深入的实验。

9. 致谢

感谢陈烽同学，他与我们在关于时序网络的“one lag”问题上进行了非常深入的讨论。

10. 附录

我们的主要工程文件分布在两个文件夹中：src 和 tools。

tools 文件夹中的模块主要用于数据预处理和图表可视化，所有代码将原始数据放在正确路径下即可运行。

- RewriteDataset.py: 清洗 GPS 数据，分割为 12 个 csv 文件。
- ExtractFeature.py 和 MergeDataset.py: 数据预处理，分别为从分割后的 GPS 数据中提取观察变量和为将观察变量（分属两个文件中）进行融合。
- GetRoadID.py: 获取道路 ID 的函数所在的模块。
- TimeSeriesDataset.py: 生成神经网络训练集和测试集。
- TTIPlot 和 visualization: 可视化。

src 文件夹有我们实现的三种模型。本报告提及的神经网络模型在 gru 子目录中，进入子目录，out 文件夹存放了我们训练出的模型文件，src 文件夹下的 gru.py 为可直接运行的文件。

助教老师从可下面的链接中获取数据文件并解压（得到 train 文件夹），替换掉 sodic/src/-gru/train 目录，直接运行 gru.py 即可开始训练过程。

链接: <https://pan.baidu.com/s/1EOlopP5J6I23CkyaZc6q7g> 提取码: 4347

小组分工 组长:高辰潇(181220014),主要负责数据预处理和特征工程。组员:孔锐(181220026),主要负责数据预处理和神经网络模型的搭建。组员:王傲然(181220055),负责 Prophet, ARIMA, XGBoost 模型的实现和效果比较。

另：由于本次比赛在 19 日下午两点才正式结束，所以我们小组在两点左右才开始着手测试数据、绘制图像并撰写实验报告。加之实验报告内容较多，因此在 19 日截止日期前我们仅完成了文字内容，使用 markdown 撰写了第一稿实验报告，但是图表的插入、引用比较混乱。因此，在 20 日上午我们使用 latex 对实验报告进行了重新排版并提交，望老师和助教老师见谅。

参考文献

- [1] Xiaofeng Cao, Yuhua Zhong, Yun Zhou, Jiang Wang, Cheng Zhu, Weiming Zhang, "Inter-active Temporal Recurrent Convolution Network for Traffic Prediction in Data Centers", Access IEEE, vol. 6, pp. 5276-5289, 2018.
- [2] Lei Li, Yabin Wu, Yihang Ou, Qi Li, Yanquan Zhou, Daoxin Chen, "Research on machine learning algorithms and feature extraction for time series", Personal Indoor and Mobile Radio Communications (PIMRC) 2017 IEEE 28th Annual International Symposium on, pp. 1-5, 2017.
- [3] Danqing Kang, Yisheng Lv, Yuan-yuan Chen, "Short-term traffic flow prediction with LSTM recurrent neural network", Intelligent Transportation Systems (ITSC) 2017 IEEE 20th International Conference on, pp. 1-6, 2017.
- [4] R. Fu, Z. Zhang and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, 2016, pp. 324-328, 2016
- [5] Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.
- [6] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.
- [7] Schuster, Mike, and Kuldeep K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681.2. Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan