中国科学: 信息科学 2021 年 第 51 卷 第 5 期: 822–833
SCIENTIA SINICA Informationis
论文

# 交互门控循环单元及其在到达时间估计中的应用

孙翊文[1], 王宇璐[1], 傅昆[2], 王征[2], 张长水[1], 周东华[3,1*], 叶杰平[2]

1. 清华大学自动化系, 北京 100084
2. 滴滴出行人工智能实验室, 北京 100094
3. 山东科技大学电气工程与自动化学院, 青岛 266590
* 通信作者. E-mail: zdh@mail.tsinghua.edu.cn

收稿日期: 2020–05–24; 接受日期: 2020–07–16; 网络出版日期: 2021–04–15

国家自然科学基金 (批准号: 61751307, 61876095) 资助项目

**摘要** 门控循环单元 (gated recurrent unit, GRU) 是一种有代表性的深度神经网络, 它在众多序列学习任务中达到了国际领先的水平. 然而, 在门控循环单元的每个时间步之间, 输入信息与隐含状态信息缺乏交互, 这对更好地挖掘上下文语义信息带来了挑战. 针对这个问题, 本文提出了一个新颖的序列学习通用的语义特征提取模型: 交互门控循环单元 (interactive gated recurrent unit, InterGRU), 可以让输入与隐含状态向量在各时间步间进行多轮充分的交互. 并且, 在到达时间估计 (estimated time of arrival, ETA) 这个有代表性、有挑战的时空序列预测任务上, 本文提出了一套基于交互门控循环单元的深度学习框架 (InterGRU-ETA) . 本文在来自滴滴出行平台真实场景下的海量数据集上充分地实验验证了 InterGRU-ETA. 结果表明, 我们的框架在预测准确率上优于目前国际上最先进的方法. 这反映了交互门控循环单元在捕获序列语义信息上的性能优势和广阔前景.

**关键词** 门控循环单元, 到达时间估计, 深度学习, 时空序列预测, 智能交通系统

## 1 引言

序列学习任务, 如语音识别 [1]、机器翻译 [2]、时空数据序列预测 [3] 是机器学习以及目前发展迅速的深度学习研究 [4] 的重要组成部分, 近年来受到国内外学者的广泛关注. 应对序列学习任务, 最有代表性的序列学习通用的语义特征提取模型是循环神经网络 (recurrent neural network, RNN) [5]. RNN 善于捕获序列数据的语义信息, 从而有效地提取特征.

在众多的循环神经网络变体中, 门控循环单元 (gated recurrent unit, GRU) [2] 与长短期记忆网络 (long short-term memory, LSTM) [6] 在综合性能上脱颖而出, 受到学者们的长期关注, 在众多序列学习任务中表现出优异的泛化性能. 这两种变体都可以克服传统 RNN 梯度可能消失或爆炸的困境, 更有

**引用格式:** 孙翊文, 王宇璐, 傅昆, 等. 交互门控循环单元及其在到达时间估计中的应用. 中国科学: 信息科学, 2021, 51: 822–833, doi: 10.1360/SSI-2020-0147
Sun Y W, Wang Y L, Fu K, et al. Interactive gated recurrent unit and its application for estimated time of arrival (in Chinese). Sci Sin Inform, 2021, 51: 822–833, doi: 10.1360/SSI-2020-0147

 

效地挖掘长时信息. GRU 和 LSTM 在长序列问题上的优越性尤其明显. 相比于更复杂的 LSTM, GRU 只有更新门 (update gate) 和重置门 (reset gate) 两个门控信号, 在每个时间步 (time step) 都省略了几次矩阵乘法操作. 优越的提取序列语义的能力加上相对简洁高效的特点, 让 GRU 自 2014 年被提出以来一直在各种序列学习任务中非常流行.

然而, 本文经过研究发现 GRU 的每一个时间步的输入和隐藏状态在输入 GRU 前是相互独立的. 序列中每一时间步的输入只能在时间步中才能与隐藏状态信息交互. 本文认为, 如果使得输入信息与隐含状态信息在各时间步间就充分地交互, 对于 GRU 捕获序列信号上下文语义信息会有所帮助.

基于以上分析, 我们提出了一个新颖的交互门控循环单元 (interactive gated recurrent unit, InterGRU). 在 InterGRU 每一个 time step 之间, 输入向量与隐含状态向量都进行了多轮的交互, 我们可以用一个超参数: 交互的轮数来控制输入信息与隐含状态信息的交互程度. 进一步地, 我们在 InterGRU 中又提出了一个新颖的 "软残差" 结构, 防止网络过度放缩原始信息, 控制输入向量与隐含状态向量的规模以确保训练时模型的收敛. 为了检验 InterGRU 对序列特征的提取能力, 我们在到达时间估计 (estimated time of arrival, ETA) 这个非常典型且重要的时空数据挖掘任务上检验 InterGRU 的序列特征语义信息提取性能.

作为一个有代表性的时空数据挖掘任务, ETA 是智能交通系统 [7,8] 中最有挑战的任务之一. 它指的是根据车辆的起始点与终点和给定路线预测行驶的时间. ETA 的准确与否对智能交通中的导航、路径规划、车辆调度等都至关重要. 复杂动态的交通系统与多种不同类型的因素对 ETA 的影响使得这个困难的问题一直受到机器学习、数据挖掘领域学者的高度关注. 2018 年以来, 随着基于深度学习的 ETA 方法 WDR [7], DeepTTE [8] 等的提出, 到达时间估计在海量数据的驱动下, 准确率达到了前所未有的新高度. 这些深度学习方法挖掘时空序列语义信息的主要特征提取器都是 LSTM. 我们针对 ETA 任务, 创新地构建一整套基于 InterGRU 的深度学习框架: InterGRU-ETA. 之后, 我们使用滴滴出行平台的海量车辆行驶时空数据, 来训练以及测评 InterGRU-ETA 与其他目前最先进的基于深度学习的 ETA 方法的预测性能. 我们发现, InterGRU-ETA 比目前国际领先的几种 ETA 方法都有更准确的预测效果. 这可以说明 InterGRU 在到达时间估计这个典型时空序列预测的问题上有着出色的特征提取能力.

本文的结构安排如下: 第 2 节分循环神经网络与到达时间估计两部分介绍本文的相关工作. 第 3 节重点介绍我们提出的交互门控循环单元 InterGRU、多轮交互操作、"软残差" 结构, 以及基于此给出的针对到达时间估计任务的完整深度学习框架 InterGRU-ETA. 第 4 节通过海量时空数据下的实验比较与分析, 充分体现了 InterGRU-ETA 在预测准确性上的优势. 第 5 节总结了全文并展望了后续值得研究的方向.

## 2 相关工作

本节主要介绍循环神经网络和到达时间估计两部分的相关工作.

### 2.1 循环神经网络

循环神经网络 (RNN) 是一类以序列式数据为输入, 所有节点间均为链式连接并沿序列推进方向递归的网络. 循环神经网络可以挖掘数据中的时序信息, 在语音识别 [1]、语言模型 [9]、机器翻译 [2] 等方面有着重要应用. 在 1982 年, Hopfield [5] 首次提出了包含外部记忆的循环神经网络, 该网络具有一定的模式识别能力, 但没有提供明确的监督学习的训练方式. Jordan [10] 和 Elman [11] 在 Hopfield 工作

的基础上进行改进, 提出了 Jordan network 和 Elman network, 这也是简单循环神经网络的雏形. 之后研究者们不断对循环神经网络进行改进[2,6,12], 现在它已发展为深度学习常用算法之一. 在众多研究工作中, 最经典的是 RNN 两个密切相关的变体, 长短时记忆网络 (LSTM)[6] 和门控循环单元 (GRU)[2]. LSTM 和 GRU 都可以对有价值的信息进行长期记忆, 有效缓解传统 RNN 的梯度消失和梯度爆炸的问题, 在涉及长序列的任务上表现良好. GRU 在很多任务上和 LSTM 的表现不分伯仲, 而 GRU 因为参数少收敛更快, 在一些任务中它的参数更新和泛化方面性能优于 LSTM[12]. 文献 [13] 为了更好的语法推断用二阶循环神经网络学习和提取有限状态自动机. 文献 [14] 提出了独立递归神经网络, 同一层的神经元相互独立, 但是跨层连接, 允许网络学习长期依赖性. 文献 [15] 使用单位矩阵或其缩放版本来初始化循环权重矩阵, 文献 [16] 提出了全容量酉循环神经网络, 它在所有酉矩阵上优化其递归矩阵, 从而比使用受限容量递归矩阵的酉循环神经网络性能提高.

由于 LSTM 和 GRU 在许多基于序列的任务中的优异表现, 很多领域将其作为基准方法[17~19] 并产生了一系列变体[20,21]. 比如将前向和后向的网络结合成 BiLSTM[20], BiGRU[21], 这种双向的 RNN 网络可以使得节点能够学习过去的和未来的表征, 在自然语言处理中有着较广泛的应用[22]. 还有些研究者在短期交通预测中使用层叠 LSTM 来提升预测模型的性能[23], 层叠 LSTM 模型是采用上一层 LSTM 的输出来作为下一层 LSTM 单元输入的模型. Ma 等[24] 借鉴胶囊网络[25] 的思路提出了 NLSTM, 在原始 LSTM 的基础上嵌套一个 LSTM 单元对隐变量参数进行学习. Li 等[26] 提出 DCRNN, 将 GRU 中线性表示的部分替换为扩散卷积, 在交通流预测问题上取得了较好的效果. 文献 [27] 在交通流预测问题上基于时空图的知识对于 LSTM 进行了改进, 在该问题上有较好的效果. 上述工作中的改进多是基于对交通领域具体问题的改进, 难以在其他问题上推广. Melis 等[28] 提出了 Mogrifier, 对每个 LSTM 单元的输入变量和隐变量进行交互操作, 在自然语言处理领域的多个任务上达到了出色的效果. 但是, 到目前为止还没有工作尝试对 GRU 每个时间步之间的输入变量与隐变量进行交互, 以更好地挖掘上下文语义信息. 因此, 我们的方法是新颖有意义的. 虽然前人的工作中不乏对循环神经网络的改进工作, 但是, 不同于以上介绍的所有相关工作, 我们提出的交互门控循环单元, 创新地在门控循环单元的每个时间步之间, 对输入向量与隐含状态向量进行了有效的交互操作. 并且我们在交互操作设计时, 提出了软残差结构设计, 新的模型提高了门控循环单元的序列特征提取能力和泛化性能. 我们将该技术用于到达时间估计这一典型的时空序列预测问题. 通过海量真实数据的实验, 本文验证了所提出的方法的有效性.

### 2.2 到达时间估计

到达时间估计指的是给定出发地和目的地以及相应的路线, 估计车辆行驶时间的过程. 它是智能交通领域至关重要、又兼具复杂性和挑战性的问题. 为了能够准确地估计到达时间, 研究者们进行了一系列的工作[5,7,8,29~34]. 这些工作主要分为基于路线的方法和数据驱动的方法, 接下来分别详细介绍这两种方法.

#### 2.2.1 基于路线的方法

基于路线的方法是一种传统的解决方案, 目前在研究领域和工业界都有着广泛的应用. 它将整条路线的到达时间估计划分为若干子问题, 即将原始问题转化为估计车辆在每个子路段的行驶时间和在每个交叉路口的延迟时间. 研究人员提出了很多策略, 如利用地理信息系统 (GIS) 的实时监测数据来估计子路段和交叉路口的行驶时间[29], 利用历史数据信息进行局部时间估计[33] 等. 传统的机器学习方法如回归和张量分解算法, 也被用来预测子路段的行驶时间或车辆通行速度[31]. 还有一些研究者

探索的重点是使用更一般的子路径 (即取原始路线的任意部分) 来近似原始路线的行驶时间 [30,35]. 基于路线的方法虽然有许多改进, 但它具有一些固有的缺陷, 使得它难以妥善解决到达时间估计的问题:

(1) 将原始问题切分的方式很可能导致局部误差的累积. 交通系统是一个动态的系统, 很难以明确的形式对未来某段时间的交通情况进行建模, 在每个子路段和交叉路口都无法保证高估计精度. 这样的局部误差逐渐积累, 对最终的预测结果可能产生较大的偏差.

(2) 忽略了个性化信息. 不同的驾驶员在同样的路段的行驶时间可能会有较大的差异, 该方法忽略了驾驶员行为习惯等对到达时间预测有重要影响的个性化信息.

#### 2.2.2 数据驱动的方法

近年来, 随着数据仓库的不断扩展, 深度学习的方法 [4,36,37] 因其优异的数据挖掘能力在处理预测问题上有着广泛的应用. 越来越多的研究者受到启发, 将深度学习的方法用到到达时间估计问题中. 此类方法的特点是基于历史轨迹直接估计整条路线的到达时间. Huang 等 [38] 使用深度信念网络来预测交通流. Li 等 [32] 提出了一种多任务表示学习模型 MURAT, 直接利用起点和终点的信息进行到达时间的估计. 由于没有直接利用路径的信息, 该方法的精度相比于其他应用于一般到达时间估计问题的深度学习模型并不高. Wang 等 [8] 使用了一种基于地理的卷积方式将原始 GPS 序列转换为特征地图继而利用 LSTM 进行到达时间的预测. 该方法提供了一个端到端的深度学习模型, 然而实际应用中无法获得未来行程的 GPS 数据, 路径规划采样获得的 GPS 点可能带来新的误差. Zhang 等 [39] 将行程轨迹转换为一个网格序列, 通过 BiLSTM 同时学习起点和终点到中间点的时间间隔. Wang 等 [7] 提出了一个 “宽度 – 深度 – 循环” (WDR) 模型, 即一个宽度线性模型、深度神经网络和循环神经网络的联合模型. 该模型可以有效利用交通信息中的密集特征、高维稀疏特征和路段序列的局部特征. 该方法主要依靠 LSTM 来捕获子路段之间的时空依赖性, 是目前效果最好且能应用于实际场景的方法. 本文将它作为基准方法. 现阶段大多数方法 [7,8,39] 都使用循环神经网络 (RNN) 来捕获时空特征进而预测到达时间. 我们提出了一种新颖的 InterGRU 的方法, 它的特点是在原始 GRU 的基础上, 增加了输入向量隐变量之间的交互, 使得模型能更多地学习输入特征的信息. 我们通过后续的实验验证了其有效性.

## 3 方法

### 3.1 交互门控循环单元 (InterGRU)

门控循环神经网络已经在序列和时间数据上得到了成功的应用, 它使用一个门控网络生成信号来控制当前输入和之前记忆发生作用的方式, 从而更新当前的激活变量, 进而更新当前的网络状态. 门控循环神经网络可以解决循环神经网络的长依赖问题, 其典型代表是 GRU [2] 和 LSTM [6]. 基于门控的循环神经网络虽然精度有所提升, 但门的引入增加了计算复杂度, 提高了计算成本. GRU 和 LSTM 在多数任务中的表现相近, 但 GRU 比 LSTM 少一个门, 即每一个时间步都减少了矩阵乘法操作, 在速度上有一定程度的优势, 因此我们选用 GRU 作为循环模型的基准.

GRU 的输入输出结构与普通 RNN 相同, 包括当前的输入 $x_t$ 和上一个节点传下来的隐状态 $h_{t-1}$,

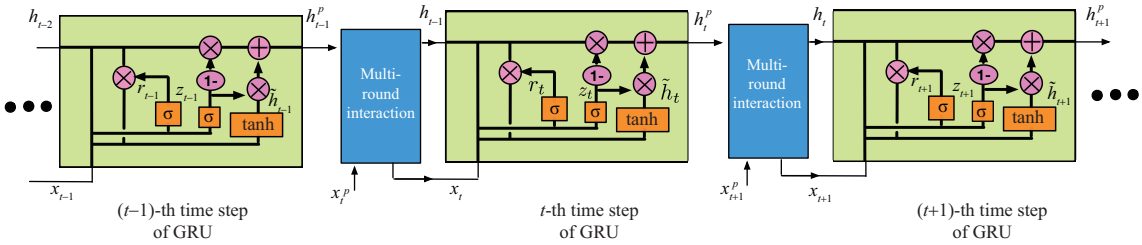


**图 1　(网络版彩图) InterGRU 完整示意图**

**Figure 1**　(Color online) The structure of InterGRU

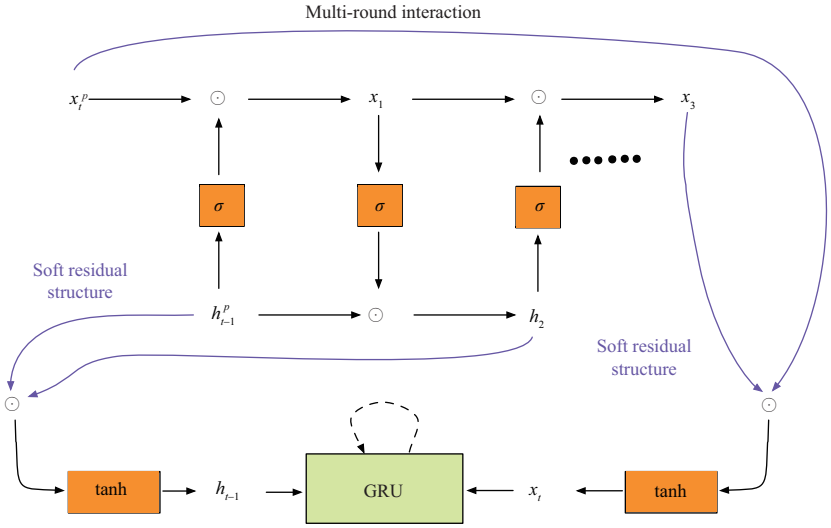


**图 2　(网络版彩图) InterGRU 单元交互原理示意图**

**Figure 2**　(Color online) Interaction principle of InterGRU unit

每个 GRU 单元的计算过程如下所示:

$$\begin{aligned} z_t &= \sigma\left(W^{zx}x_t + W^{zh}h_{t-1} + b^z\right), \\ r_t &= \sigma\left(W^{rx}x_t + W^{rh}h_{t-1} + b^r\right), \\ \tilde{h}_t &= \tanh\left(r_t \odot W^{hh}h_{t-1} + W^{hx}x_t + b^h\right), \\ h_t &= \left(1 - z^t\right) \odot \tilde{h}_t + z_t \odot h_{t-1}, \end{aligned} \tag{1}$$

其中, $r_t$ 为重置门, $z_t$ 为更新门, $\sigma$ 为 sigmoid 函数, $\odot$ 为 Hadamard 积, $W^{**}$ 和 $b^*$ 分别为权重系数和偏差. 为进一步提升 GRU 单元整合信息的能力, 我们创新地提出了一种交互式门控循环单元 (InterGRU), 在每个 InterGRU 单元间增加了当前输入 $x_t$ 与上个单元传入的隐状态参数 $h_{t-1}$ 之间的交互操作. 对于序列化数据, InterGRU 原理如图 1 所示. 每个 InterGRU 单元中的交互操作的原理如图 2 所示. 每个单元的输入 $x_t$ 和隐状态 $h_{t-1}$ 通过迭代进行信息交互. 我们的方法借鉴了 [28] 的思想. 由于交互过程要进行多轮迭代, 为防止网络过度放缩原始信息, 保证网络在训练过程中顺利收敛, 我们提出了一个新颖的 “软残差” 结构. 深度学习中残差思想来自于 [40]. 但我们没有采用直接相加的操作, 而是将原始状态信息 ${x_t}^p$, ${h_{t-1}}^p$ 与迭代最终状态信息 $x_r$, $h_r$ 进行点积, 并通过 tanh 函数对输出的大小进行限制, 使 InterGRU 单元正常收敛. 我们称之为 “软残差”. 具体的公式如下所示:

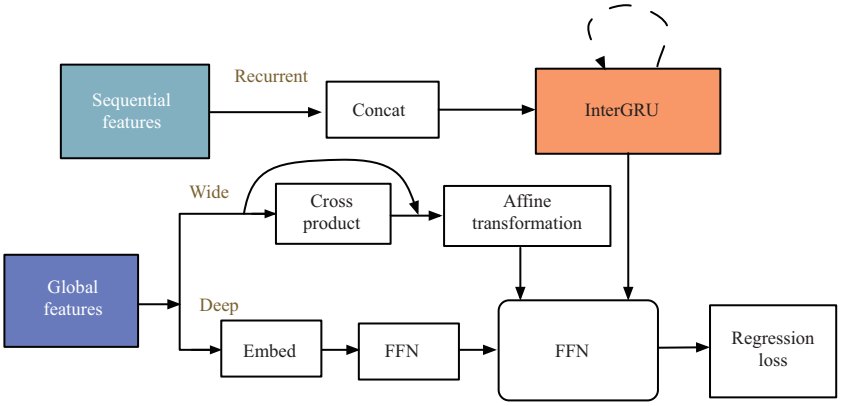


**图 3 (网络版彩图) InterGRU-ETA 模型整体框架图**

**Figure 3** (Color online) The architecture of InterGRU-ETA model

$$\boldsymbol{x}_t^i = \sigma\left(\boldsymbol{G}^x \boldsymbol{h}_{t-1}^{i-1}\right) \odot \boldsymbol{x}_t^{i-2}, \text{ for odd } i \in [1 \dots r], \tag{2}$$

$$\boldsymbol{h}_{t-1}^i = \sigma\left(\boldsymbol{G}^h \boldsymbol{x}_t{}^{i-1}\right) \odot \boldsymbol{h}_{t-1}^{i-2}, \text{ for even } i \in [1 \dots r], \tag{3}$$

$$\begin{aligned} x_t &= \tanh\left(x_t{}^p \odot x_t{}^r\right), \\ h_{t-1} &= \tanh\left(h_{t-1}{}^p \odot h_{t-1}{}^r\right), \end{aligned} \tag{4}$$

其中, $r$ 为迭代次数, $\boldsymbol{G}^x$, $\boldsymbol{G}^h$ 是可学习参数.

### 3.2 InterGRU-ETA 模型架构

为验证 InterGRU 的效果, 我们将其应用于到达时间估计问题, 提出了 InterGRU-ETA 模型. 到达时间估计在 2018 年由 Wang 等 [7] 建模成为一个纯机器学习问题 (回归), 其定义如下:

**定义1** (ETA 问题) 对于行驶轨迹数据集, 都有 $\{s_i, e_i, d_i, \boldsymbol{p}_i\}_{i=1}^N$, 其中 $s_i$ 是第 $i$ 条轨迹的出发时间, $e_i$ 是第 $i$ 条轨迹的到达时间, $d_i$ 是第 $i$ 条轨迹的司机 ID, $\boldsymbol{p}_i$ 是第 $i$ 条轨迹中的道路集, $N$ 是样本的总数. 每个样本的到达时间由 $y_i = e_i - s_i$ 计算得到. 道路集 $\boldsymbol{p}_i$ 由一系列子路段构成, 即 $\boldsymbol{p}_i = \{l_{i1}, l_{i2}, \dots, l_{iT_i}\}$, 其中 $l_{ij}$ 代表第 $i$ 条轨迹中第 $j$ 个子路段, $T_i$ 是道路集的长度.

目前解决这个问题的最好模型是 WDR [7], 我们将它作为基准的方法. 在 WDR 的基础上, 我们设计了一个新颖的基于交互式门控循环单元的 ETA 模型, 它可以让输入与隐含状态向量在各时间步间进行多轮充分的交互, 从而大大提升循环网络部分的性能. 接下来我们将对新模型进行详细介绍. InterGRU-ETA 模型的整体架构如图 3 所示. 模型主要分为 3 部分, 宽度模型、深度模型和循环模型.

(1) 宽度模型通过一个二阶叉积和仿射变换来记忆历史的全局特征.

(2) 深度模型搭建了全局特征 (包括司机 ID, 星期几和所属时间片 (每天被分为 288 个时间片)) 的嵌入 [41] 表格. 稀疏特征的嵌入空间维数为 20. 之后嵌入向量被连接起来, 通过激活函数 ReLU [36] 作用后进入堆叠的全连接层中.

(3) 循环模型整合样本中的序列化特征, 将该特征输入到我们提出的交互式门控循环单元 InterGRU 中. InterGRU 的具体原理我们已在 3.1 小节详细说明.

宽度、深度和循环模型的输出经过一个前馈神经网络之后进入回归模型, 得到最终的结果. InterGRU-ETA 模型的参数是通过优化平均绝对百分比误差 (MAPE) 损失函数来训练的, MAPE 具

体形式如下:

$$\mathrm{MAPE}=\sum_{j=1}^{N}\frac{|y_j-y_j'|}{y_j}, \tag{5}$$

其中, $y_i$ 为每个样本的真实标签, $y_i'$ 为模型最终的估计时间.

## 4 实验

### 4.1 数据集

我们实验中使用的数据是滴滴出行平台收集的海量汽车轨迹数据. 它包含了北京出租车司机在 2018 年超过 4 个月的轨迹信息, 囊括了拥有不同驾驶习惯的驾驶员在各种道路类型上的驾驶数据, 我们将这个数据集称为北京 2018.

在实验之前, 我们过滤了行程时间小于 60 s 和行驶速度大于 120 km/h 的异常数据. 我们将这个数据集划分成训练集 (前 16 周的数据)、验证集 (中间 2 个周的数据), 和测试集 (后 2 个周的数据).

### 4.2 参数设置和评价方法

#### 4.2.1 方法比较

在北京 2018 数据集上, 我们将 InterGRU-ETA 与其他在到达时间估计中有代表性的方法进行比较. 在传统的非深度方法中, 我们选用 route-ETA 来作比较. route-ETA 是一种简单有效的方法, 它将原始路线划分为若干子区间, 并通过一个交通监测系统获得每个子路段的平均速度和每个交叉路口的延迟时间. 每个子路段的时间由子路段的长度除以在该路段的平均速度得到, 最终估计的到达时间为各个子路段的估计时间和路口的延迟时间之和. 在深度模型中, 我们选择目前文献中表现最好的 WDR 及其变体 WD-FFN, WD-Resnet 作为比较的方法. WDR 是一个宽度线性模型、深度神经网络和循环神经网络的联合模型, 可以有效利用数据集中不同类型的特征信息. WDR-FFN 和 WDR-Resnet 则是将 WDR 中的循环网络部分分别用前馈神经网络 (FFN) 和残差神经网络 (Resnet) 代替.

#### 4.2.2 参数设置

在我们的实验中, 所有方法都是使用 PyTorch 框架 [42] 编写的. 基于深度学习的方法迭代次数均为 350 万, batch size 为 256. InterGRU-ETA 的交互迭代轮数为 $r$ ($r\in[1,5]$), 表 1 中列出的结果是 $r$ 为 1 的结果. 我们使用误差反向传播 (back propagation, BP) 的方法训练基于深度学习的方法, 其中优化的目标函数为 MAPE 损失函数. 我们使用 Adam [43] 优化器进行训练, 初始学习率为 0.0002.

#### 4.2.3 评价指标

为了评价 InterGRU 和其他方法的表现, 我们使用了 3 种常用的预测评价指标:

平均绝对误差 (MAE):

$$\mathrm{MAE}=\frac{1}{N}\sum_{i=1}^{N}|y_i-y_i'|\,. \tag{6}$$

均方根误差 (RMSE):

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i-y_i')^2}. \tag{7}$$

表 1 各方法结果比较

**Table 1** Results of different methods

| | MAPE (%) | MAE (s) | RMSE (s) |
|---|---|---|---|
| Route-ETA | 25.010 | 69.008 | 106.966 |
| WD-FFN | 21.109 | 57.758 | 93.491 |
| WD-Resnet | 21.015 | 57.064 | 92.241 |
| WDR (GRU) | 19.673 | 55.372 | 90.801 |
| WDR (LSTM) | 19.598 | 55.227 | 90.480 |
| InterGRU-ETA (ours) | **19.579** | **54.702** | **89.45** |

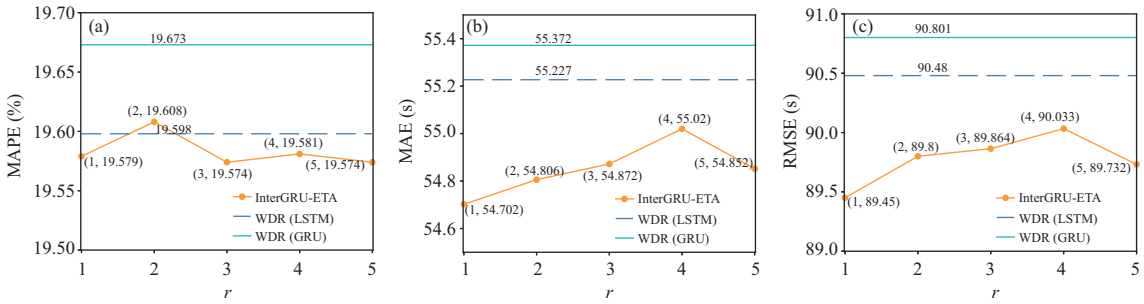


图 4 (网络版彩图) 损失函数随超参数变化曲线

**Figure 4** (Color online) The loss under different hyper-parameters. (a) MAPE; (b) MAE; (c) RMSE

平均绝对百分比误差 (MAPE), 见式 (5). 这里我们用 $y_i$ 表示每个样本的真实标签, $y_i'$ 表示我们最终的估计时间, $N$ 表示样本数目.

### 4.3 实验结果及分析

如表 1 所示, 我们提出的 InterGRU-ETA 模型在北京 2018 数据集上的表现比其他的方法都要好. 详细的结果分析如下所示:

(1) 非深度学习的代表方法 route-ETA 在实验中的表现比其他深度学习的差很多. 这说明数据驱动的方法在拥有大量时空数据的复杂交通系统中更为有效.

(2) 使用循环神经网络作为序列特征提取器的效果要好于前馈神经网络和残差结构. 在已有的 RNN 变体中, LSTM 的表现效果最好.

(3) 我们的 InterGRU-ETA 模型在到达时间预测任务上的表现最好. 在 MAPE 损失上的表现超出了 WDR (LSTM) 的方法 0.09%, 在 MAE 损失上超出了 WDR (LSTM) 0.95%, 在 RMSE 上超出了 WDR (LSTM) 1.14%.

在 ETA 任务上的实验表明我们基于 InterGRU 的改进模型效果比使用 LSTM 要好. 这说明 InterGRU 能够更好地提取序列特征中的信息.

### 4.4 交互迭代轮数的影响

交互迭代轮数 $r$ 是在 InterGRU 单元中输入向量和隐状态向量的交互次数, 详细参见 3.1 小节的式 (2) 和 (3). 我们对交互迭代轮数 $r$ 对模型效果的影响进行了分析, 其结果如图 4 所示, 可以看出:

(1) 交互迭代轮数 $r$ 取 1 到 5 之间的值时, 从总体情况来看 InterGRU 的效果都比 LSTM 和 GRU

更好, 这说明交互操作的引入提升了循环单元对序列特征的信息提取能力. MAE 和 RMSE 随着 $r$ 的变化不是很明显, 说明我们的算法对超参数不是很敏感, InterGTU 对于交互轮数的取值有一定的鲁棒性. 从图 4(b) 我们可以分析得出: 目前最先进的方法 WDR, 采用 LSTM 和 GRU 分别作为序列特征提取器时, 在测试集上的 MAE 误差分别为 55.227 s 和 55.372 s, 而采用我们提出的 InterGRU 的 InterGRU-ETA, 随着交互迭代轮数从 1 增加到 5 时, 测试集上的 MAE 误差最大为 55.020 s. MAE 误差最小更是低至 54.702 s, 分别比 WDR (LSTM) 和 WDR (GRU) 降低了 0.95% 和 1.21%.

(2) 当 $r$ 为 1 时, InterGRU 在 ETA 任务的各个指标上综合表现最好. ETA 任务最看重的指标是 MAPE, 这是一个对于长、短轨迹综合考虑的指标, 因此被我们选为目标函数. 如果作决策更看重 MAPE 指标, 则当 $r$ 为 3 时, InterGRU 模型的效果最优. 在实际情况中可以根据任务需要选取合适的 $r$.

## 5 结论与展望

本文提出了一个新颖的序列学习通用的特征提取模块 InterGRU. 我们设计了一个新的交互式门控循环单元, 在每个单元的输入变量与隐变量之间进行迭代式信息交互, 并通过软残差的方式防止原始信息在迭代中丢失. 我们在到达时间估计问题上验证了该方法的有效性. 在来自滴滴出行平台真实场景下的海量数据集上的实验结果表明, InterGRU-ETA 模型相比国际先进的方法准确率都更高. 本文接下来将考虑能否利用智能交通领域的理论知识, 进一步改进我们提出的 InterGRU-ETA, 以期探索出一个更有针对性、更有潜力的深度学习框架.

## 参考文献


1 Miao Y J, Gowayyed M, Metze F. EESEN: end-to-end speech recognition using deep RNN models and WFST-based decoding. In: Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015

2 Cho K, van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. 2014. ArXiv:1406.1078

3 Wang S Z, Cao J N, Yu P S. Deep learning for spatio-temporal data mining: a survey. 2019. ArXiv:1906.04928

4 LeCun Y, Bengio Y, Hinton G. Deep learning. Nature, 2015, 521: 436–444

5 Hopfield J J. Neural networks and physical systems with emergent collective computational abilities. Proc Natl Acad Sci USA, 1982, 79: 2554–2558

6 Hochreiter S, Schmidhuber J. LSTM can solve hard long time lag problems. In: Proceedings of Advances in Neural Information Processing Systems, 1997. 473–479

7 Wang Z, Fu K, Ye J P. Learning to estimate the travel time. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 858–866

8 Wang D, Zhang J B, Cao W, et al. When will you arrive? Estimating travel time based on deep neural networks. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018

9 Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model. In: Proceedings of the 11th Annual Conference of the International Speech Communication Association, Makuhari, 2010

10 Jordan M I. Serial order: a parallel distributed processing approach. Adv Psychol, 1997, 121: 471–495

11 Elman J L. Finding structure in time. Cognitive Sci, 1990, 14: 179–211

12 Chung J, Gulcehre C, Cho K, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling. 2014. ArXiv:1412.3555

13 Giles C L, Miller C, Chen D, et al. Learning and extracting finite state automata with second-order recurrent neural networks. Neural Comput, 1992, 4: 393–405

14 Li S, Li W Q, Cook C, et al. Independently recurrent neural network (INDRNN): building a longer and deeper RNN. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2018

15 Le Q V, Jaitly N, Hinton G E. A simple way to initialize recurrent networks of rectified linear units. 2015. ArXiv:1504.00941
16 Wisdom S, Powers T, Hershey J R, et al. Full-capacity unitary recurrent neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2016
17 Auli M, Galley M, Quirk C, et al. Joint language and translation modeling with recurrent neural networks. In: Proceedings of Conference on Empirical Methods in Natural Language Processing, 2013. 1044–1054
18 Karpathy A, Li F F. Deep visual-semantic alignments for generating image descriptions. 2014. ArXiv:1412.2306
19 Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. J Mach Learn Res, 2003, 3: 1137–1155
20 Graves A, Schmidhuber J. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural Netw, 2005, 18: 602–610
21 Jabreel M, Moreno A. Target-dependent sentiment analysis of tweets using a bi-directional gated recurrent unit. In: Proceedings of the 13th International Conference on Web Information Systems and Technologies, 2017
22 Lipton Z C, Berkowitz J, Elkan C. A critical review of recurrent neural networks for sequence learning. 2015. ArXiv:1506.00019
23 Cui Z Y, Ke R M, Pu Z Y, et al. Deep bidirectional and unidirectional LSTM recurrent neural network for network-wide traffic speed prediction. In: Proceedings of the 6th International Workshop on Urban Computing, 2017
24 Ma X L, Zhong H, Li Y, et al. Forecasting transportation network speed using deep capsule networks with nested LSTM models. IEEE Trans Intell Transp Syst, 2020. doi: 10.1109/TITS.2020.2984813
25 Sabour S, Frosst N, Hinton G E. Dynamic routing between capsules. In: Proceedings of the 31st Conference on Neural Information Processing System, 2017. 3856–3866
26 Li Y G, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. 2017. ArXiv:1707.01926
27 Sun Y W, Wang L L, Fu K, et al. Constructing geographic and long-term temporal graph for traffic forecasting. 2020. ArXiv:2004.10958
28 Melis G, Koisk T, Blunsom P. Mogrifier LSTM. 2019. ArXiv:1909.01792
29 Jenelius E, Koutsopoulos H N. Travel time estimation for urban road networks using low frequency probe vehicle data. Transp Res Part B-Meth, 2013, 53: 64–81
30 Lee W C, Si W P, Chen L, et al. HTTP: a new framework for bus travel time prediction based on historical trajectories. In: Proceedings of the 20th International Conference on Advances in Geographic Information Systems, 2012. 279–288
31 Wang Y L, Zheng Y, Xue Y X. Travel time estimation of a path using sparse trajectories. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. 25–34
32 Li Y G, Fu K, Wang Z, et al. Multi-task representation learning for travel time estimation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018. 1695–1704
33 Wang H J, Li Z H, Kuo Y H, et al. A simple baseline for travel time estimation using large-scale trip data. In: Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, 2016
34 Yuan J, Zheng Y, Xie X, et al. T-drive: enhancing driving directions with taxi drivers' intelligence. IEEE Trans Knowl Data Eng, 2011, 25: 220–232
35 Weng J C, Wang C, Huang H N, et al. Real-time bus travel speed estimation model based on bus GPS data. Adv Mech Eng, 2016, 8: 1–10
36 Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2012. 1097–1105
37 Larochelle H, Bengio Y, Louradour J, et al. Exploring strategies for training deep neural networks. J Mach Learn Res, 2009, 10: 1–40
38 Huang W H, Song G J, Hong H K, et al. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. IEEE Trans Intell Transp Syst, 2014, 15: 2191–2201
39 Zhang H Y, Wu H, Sun W W, et al. Deeptravel: a neural network based travel time estimation model with auxiliary supervision. 2018. ArXiv:1802.02147
40 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2016

41 Bengio Y, Ducharme R, Vincent P, et al. A neural probabilistic language model. J Mach Learn Res, 2003, 3: 1137–1155
42 Paszke A, Gross S, Massa F, et al. PyTorch: an imperative style, high-performance deep learning library. In: Proceedings of Advances in Neural Information Processing Systems, 2019. 8024–8035
43 Kingma D, Ba J. ADAM: a method for stochastic optimization. In: Proceedings of the 3rd International Conference for Learning Representations, 2015

# Interactive gated recurrent unit and its application for estimated time of arrival

Yiwen SUN[1], Yulu WANG[1], Kun FU[2], Zheng WANG[2], Changshui ZHANG[1], Donghua ZHOU[3,1*] & Jieping YE[2]

1. *Department of Automation, Tsinghua University, Beijing* 100084*, China*;
2. *DiDi AI Labs, Beijing* 100094*, China*;
3. *College of Electrical Engineering and Automation, Shandong University of Science and Technology, Qingdao* 266590*, China*
* Corresponding author. E-mail: zdh@mail.tsinghua.edu.cn

**Abstract** A gated recurrent unit (GRU) is a representative deep neural network that has achieved promising results in many sequence learning tasks. However, there is a lack of interaction between the input and the hidden-state among each time step of GRU, resulting in the challenges to mine contextual semantic information effectively. In this paper, we propose a novel deep learning method called interactive gated recurrent unit (InterGRU) to solve this problem, which allows full interaction between the input and the hidden state at various time steps. Furthermore, we propose a deep learning framework, InterGRU-ETA, based on InterGRU for the estimated time of arrival (ETA) which is a representative and challenging time series forecasting task. Our framework has been fully experimentally verified on the large-scale real-world datasets from the Didi Chuxing platform. The results on massive historical vehicle travel data show that InterGRU-ETA is superior to other state-of-the-art algorithms. This can reflect the advantages of InterGRU in capturing sequential semantic information.

**Keywords** gated recurrent unit, estimated time of arrival, deep learning, spatio-temporal forecasting, intelligent transportation systems

**Yiwen SUN** was born in 1995. He received his B.E. degree from the Department of Electronic Engineering, Tsinghua University, Beijing, China, in 2017. He is currently pursuing a Ph.D. degree in the Department of Automation, Tsinghua University. His research interests include machine learning, deep learning, sequence learning, spatial-temporal data mining, and intelligent transportation systems.

**Donghua ZHOU** was born in 1963. He received his B.E., M.S., and Ph.D. degrees all in electrical engineering from Shanghai Jiao Tong University, China, in 1985, 1988, and 1990, respectively. He joined Tsinghua University in 1996 and was promoted to full professor in 1997. He was the head of the Department of Automation, Tsinghua University, from 2008 to 2015. He is now a vice president of Shandong University of Science and Technology. His current research interests include fault diagnosis, fault-tolerant control, and operational safety evaluation.

**Changshui ZHANG** was born in 1965. He received his B.S. degree in mathematics from Peking University, Beijing, China, in 1986, and his M.S. and Ph.D. degrees in control science and engineering from Tsinghua University, Beijing, in 1989 and 1992, respectively. In 1992, he joined the Department of Automation, Tsinghua University, where he is currently a professor. His current research interests include machine learning, artificial intelligence, pattern recognition, and computer vision.

**Jieping YE** was born in 1975. He received his Ph.D. degree in computer science from the University of Minnesota, Twin Cities, MN, USA, in 2005. He is currently the VP of Didi Chuxing, a Didi Fellow and IEEE fellow. He is also an associate professor at the University of Michigan, Ann Arbor, MI, USA. His research interests include big data, machine learning, and data mining, with applications in transportation and biomedicine.