

Neural ODE (神经常微分方程) 在时空序列预测中的应用

2024.10.27



一： **Neural ODE 神经常微分方程介绍**

- 介绍NODE **理论推导**以及部分变体

二： **神经常微分方程在交通流预测的应用**

- 原始NODE在交通流预测中的应用
- **受控神经常微分方程NCDE**在交通流预测中的应用
- **延迟神经常微分方程NDDE**在交通流预测中的应用

三： **神经常微分方程总结**

- NODE发展总结

一： **Neural ODE 神经常微分方程介绍**

- 介绍Neural ODE **理论推导**以及部分变体

二： 神经常微分方程在交通流预测的应用

- 原始ODE在交通流预测中的应用
- 受控神经常微分方程NCDE在交通流预测中的应用
- 延迟神经常微分方程NDDE在交通流预测中的应用

三： 神经常微分方程总结

什么是神经常微分方程?

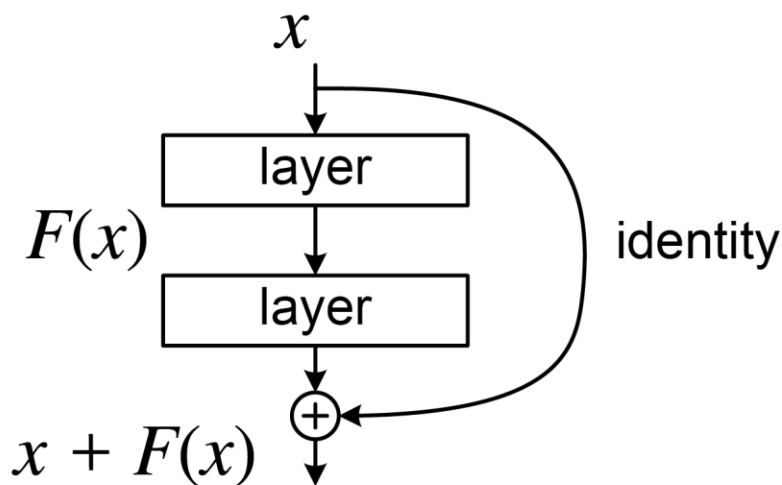


- 常微分方程 Ordinary Differential Equation

$$\frac{dy}{dt}(t) = f(t) \xrightarrow{\text{求解可得}} y_T = y(0) + \int_0^T f(t)dt$$

- 神经常微分方程 Neural Ordinary Differential Equation^[1]

Motivation



ResNet

一个残差块的形式化表达如下

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

进行数学变换

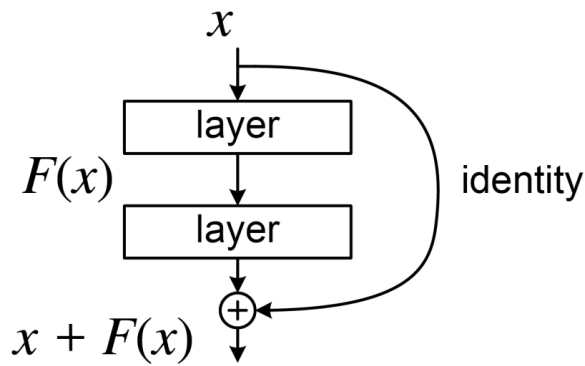
$$\lim_{\Delta t \rightarrow 0} \frac{\mathbf{h}_{t+1} - \mathbf{h}_t}{\Delta t} = \frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

什么是神经常微分方程?

• 神经常微分方程 Neural Ordinary Differential Equation

ResNet

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$



离散形式连续化

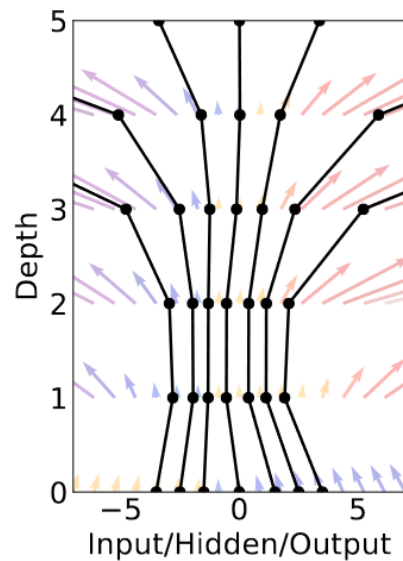


NODE

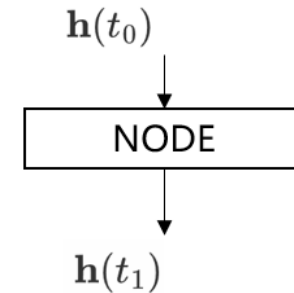
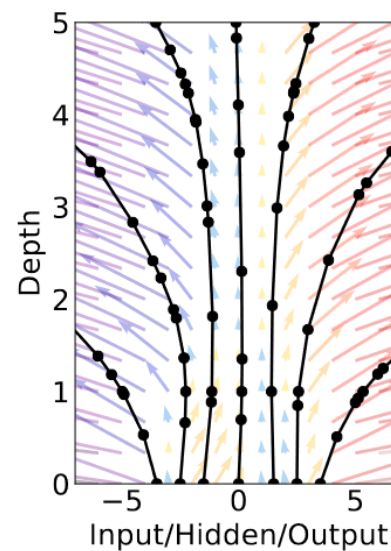
$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

$$\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt$$

Residual Network



ODE Network



用神经网络 $f(\mathbf{h}(t), t, \theta)$ 来参数化常微分方程

得到了一个连续深度的残差网络

• 神经常微分方程

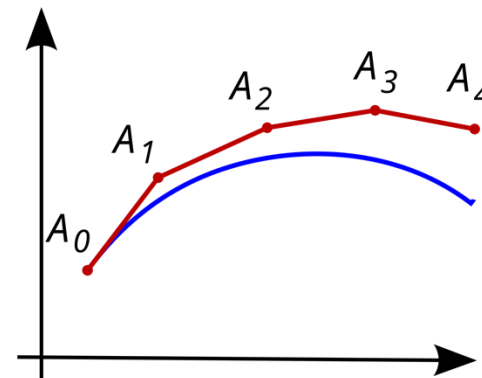
$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta) \quad \mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt$$



数值求解

$$h_{t_1} = \text{ODESolve}(\mathbf{h}(t_0), f, t_0, t_1, \theta)$$

数值求解示例：欧拉法



• 如何反向传播更新参数?

- 损失函数

$$L(\mathbf{h}(t_1)) = L\left(\mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt\right) = L(\text{ODESolve}(\mathbf{h}(t_0), f, t_0, t_1, \theta))$$

直接对数值求解过程进行反向传播：计算时间空间代价太大!

- 伴随敏感性方法(adjoint sensitivity method)

引入一个伴随变量 $\mathbf{a}(t) = \frac{\partial L}{\partial \mathbf{h}(t)}$

• 如何反向传播更新参数?

- 损失函数

$$L(\mathbf{h}(t_1)) = L\left(\mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt\right) = L(\text{ODESolve}(\mathbf{h}(t_0), f, t_0, t_1, \theta))$$

- 伴随敏感性方法(adjoint sensitivity method)

引入一个伴随变量 $\mathbf{a}(t) = \frac{\partial L}{\partial \mathbf{h}(t)}$

推导可得另外一个ODE $\frac{d\mathbf{a}(t)}{dt} = -\mathbf{a}(t)^\top \frac{\partial f(\mathbf{h}(t), t, \theta)}{\partial \mathbf{h}}$ $\mathbf{a}(t)$ 也可以通过调用ODESolver解

得到伴随变量 $\mathbf{a}(t)$, 有以下理论推导得到损失函数对参数的导数为另外一个ODE:

$$\frac{dL}{d\theta} = \int_{t_1}^{t_0} \mathbf{a}(t)^\top \frac{\partial f(\mathbf{h}(t), t, \theta)}{\partial \theta} dt$$

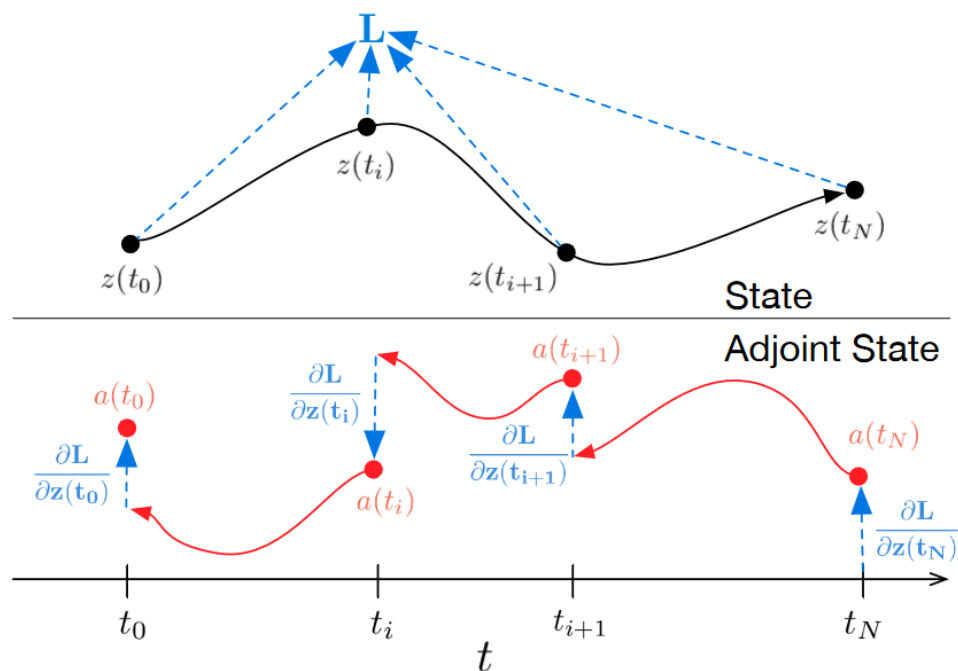
可以通过调用ODESolver解

• 如何反向传播更新参数？

- 损失函数

$$L(\mathbf{h}(t_1)) = L\left(\mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt\right) = L(\text{ODESolve}(\mathbf{h}(t_0), f, t_0, t_1, \theta))$$

- 伴随变量 $\mathbf{a}(t) = \frac{\partial L}{\partial \mathbf{h}(t)}$

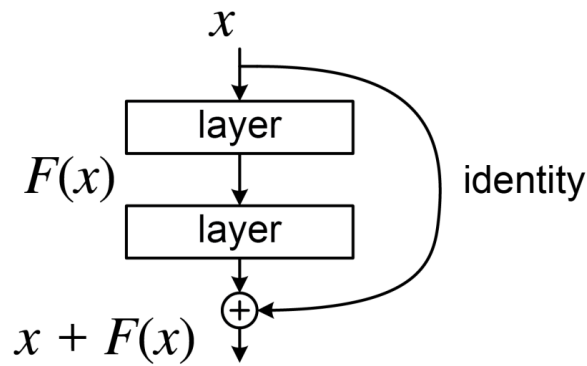


NODE总结

• 神经常微分方程 Neural Ordinary Differential Equation

ResNet

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$



离散形式连续化

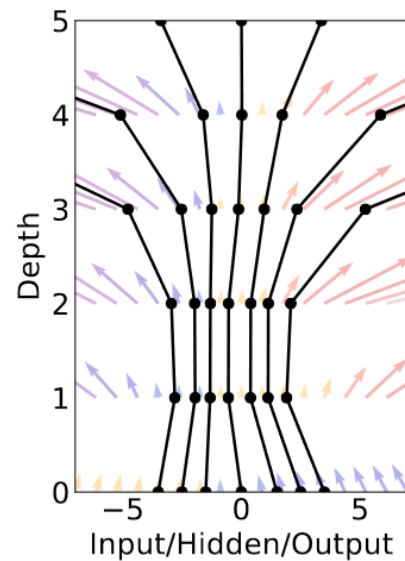


NODE

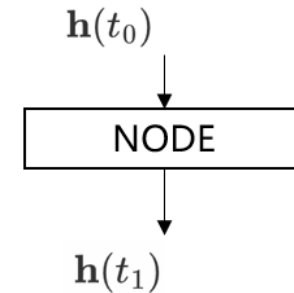
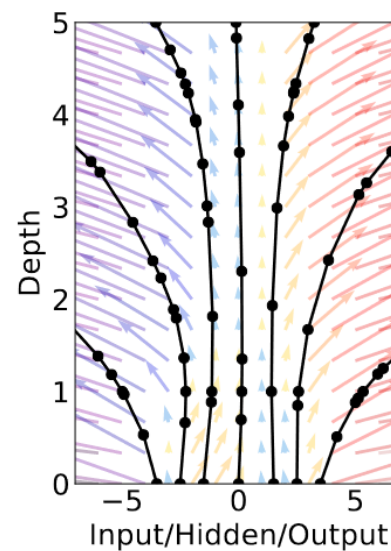
$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

$$\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt$$

Residual Network



ODE Network



用神经网络 $f(\mathbf{h}(t), t, \theta)$ 来参数化常微分方程

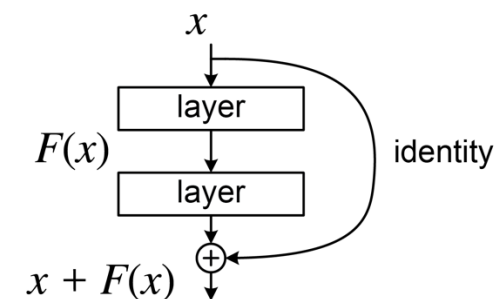
得到了一个连续深度的残差网络

• Neural Controlled Differential Equations^[2] (NCDE)

- NODE 存在的问题

$$\frac{dh(t)}{dt} = f(\mathbf{h}(t), t, \theta) \quad \mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt$$

1. t 并不是时间, 在原论文中对应的是残差块的层数 (深度)
2. 当神经网络 f 的参数 θ 确定时, 初始状态 $\mathbf{h}(t_0)$ 唯一决定了最终状态 $\mathbf{h}(t_1)$
无法处理时间序列数据中后续到达的数据



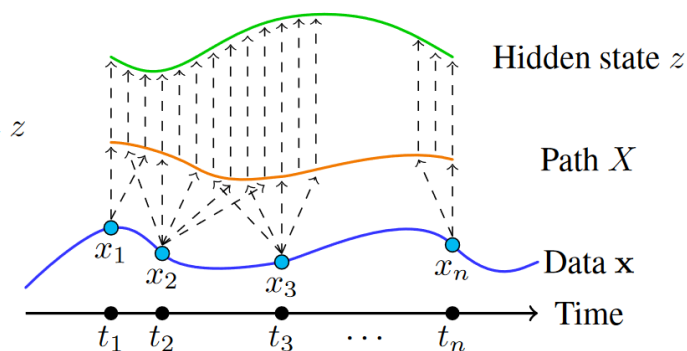
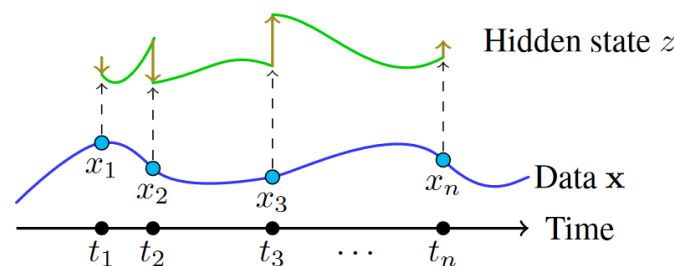
- NCDE 受控神经微分方程提出

假设离散的时间序列为: $\mathbf{x} = ((t_0, x_0), (t_1, x_1), \dots, (t_n, x_n))$

引入一个在时间 t 上连续的控制函数 X , 且 $X_{t_i} = (t_i, x_i)$

原文 X 是使用三次样条插值函数

$$h_{t_n} = h_{t_0} + \int_{t_0}^{t_n} f(h(t), t, \theta) dX_t$$



一：Neural ODE 神经常微分方程介绍

- 介绍Neural ODE 理论推导以及部分变体

二：神经常微分方程在交通流预测的应用

- 原始ODE在交通流预测中的应用
- 受控神经常微分方程NCDE在交通流预测中的应用
- 延迟神经常微分方程NDDE在交通流预测中的应用

三：神经常微分方程总结

- **Spatial-temporal graph ode networks for traffic flow forecasting (SIGKDD 2021)**

- **背景：浅层图卷积网络无法捕捉远距离空间相关性**

普通GNN存在过平滑问题，所有节点表示在更深的层中会收敛到相同的值。这限制了GNN的深度，难以获得更深层次的空间特征。

- **连续形式的GCN的引入**

原始GCN的表示

$$H_{l+1} = GCN(H_l) = \sigma(\hat{A}H_lW)$$

基于张量的GCN 离散形式

$$\mathcal{H}_{l+1} = \mathcal{H}_l \times_1 \hat{A} \times_2 U \times_3 W + \mathcal{H}_0$$

推导的ODE

$$\frac{d\mathcal{H}(t)}{dt} = \mathcal{H}(t) \times_1 (\hat{A} - I) + \mathcal{H}(t) \times_2 (U - I) + \mathcal{H}(t) \times_3 (W - I) + \mathcal{H}_0$$

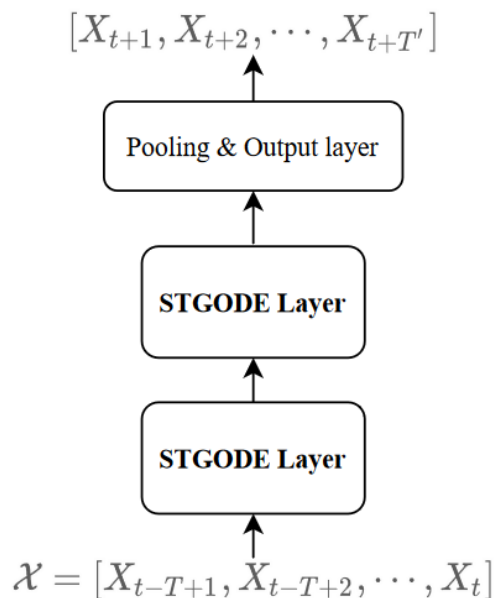
$$\mathcal{H}(t) = ODEsolve \left(\frac{d\mathcal{H}(t)}{dt}, \mathcal{H}_0, t \right)$$

Neural ODE 在交通流预测中的应用

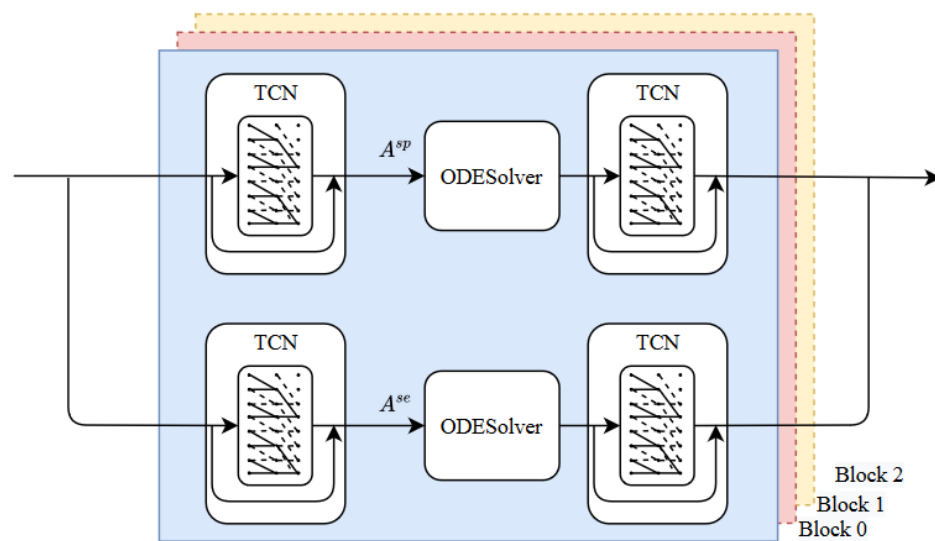


- Spatial-temporal graph ode networks for traffic flow forecasting (SIGKDD 2021)

- 模型结构



(a) Framework

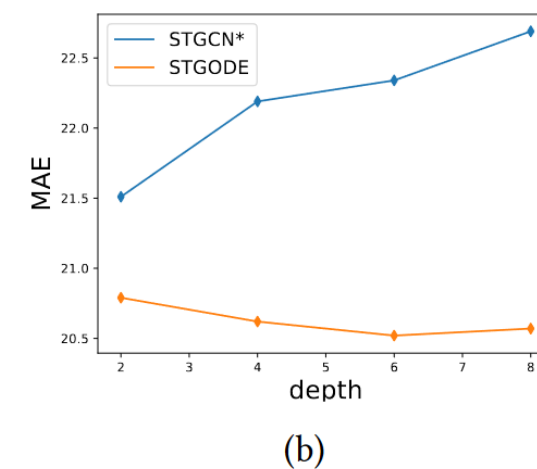
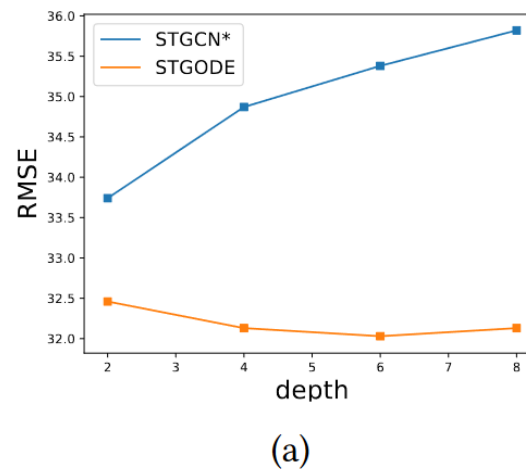
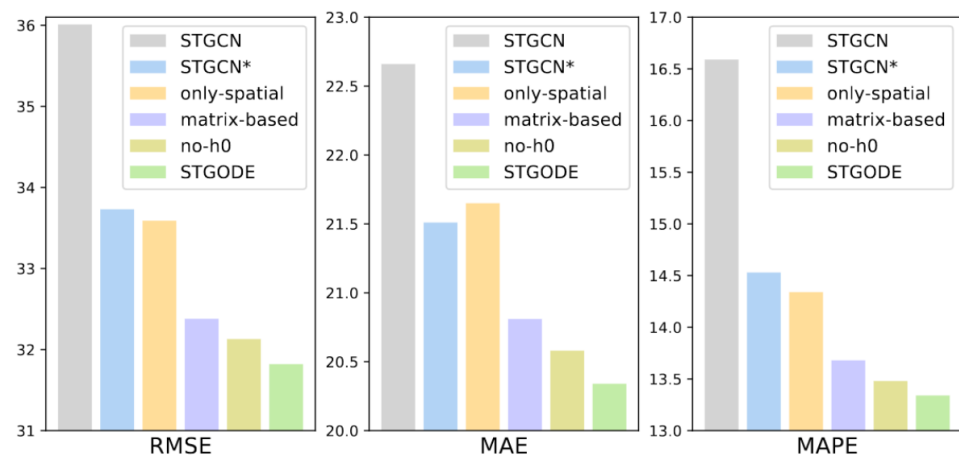


(b) STGODE Layer



- Spatial-temporal graph ode networks for traffic flow forecasting (SIGKDD 2021)

- 实验结果



一：Neural ODE 神经常微分方程介绍

- 介绍Neural ODE 理论推导以及部分变体

二：神经常微分方程在交通流预测的应用

- 原始ODE在交通流预测中的应用
- **受控神经常微分方程NCDE在交通流预测中的应用**
- 延迟神经常微分方程NDDE在交通流预测中的应用

三：神经常微分方程总结

- Graph neural controlled differential equations for traffic forecasting (AAAI 22)

- STG-NCDE

在时间和空间尺度上分别设计了一个NCDE并且整合成统一框架

- 时间尺度信息处理

$$\mathbf{H}(T) = \mathbf{H}(0) + \int_0^T f(\mathbf{H}(t); \theta_f) \frac{d\mathbf{X}(t)}{dt} dt,$$

其中 $\mathbf{H}(t) \in \mathbb{R}^{|\mathcal{V}| \times \dim(\mathbf{h}^{(v)})}$ 是节点的时间信息隐藏状态

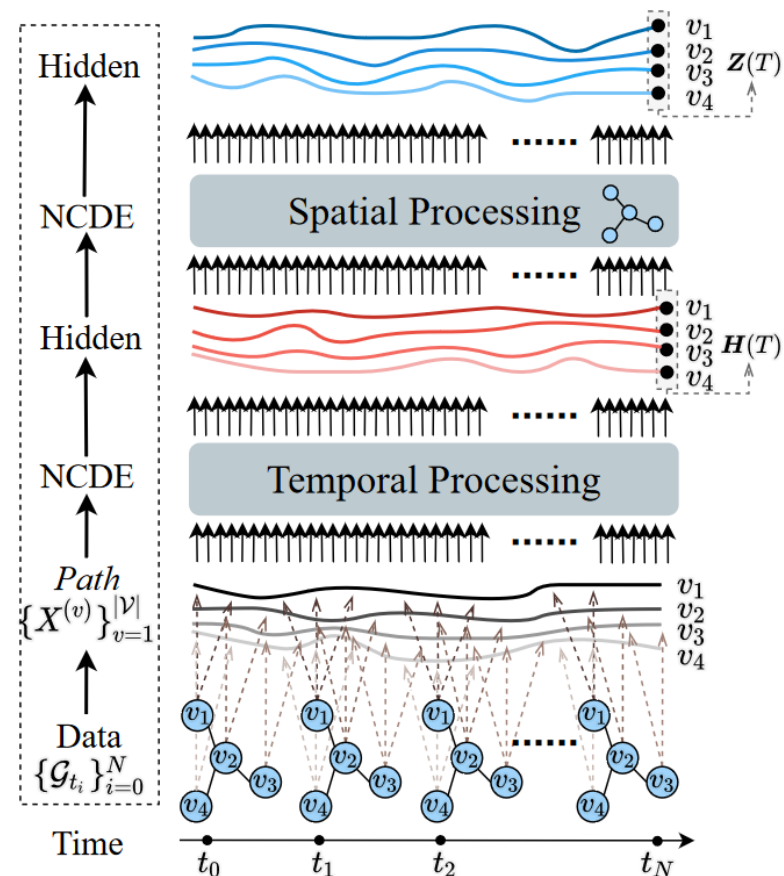
神经网络就是一个K层的MLP。

$$f(\mathbf{H}(t); \theta_f) = \psi(\text{FC}_{|\mathcal{V}| \times \dim(\mathbf{h}^{(v)}) \rightarrow |\mathcal{V}| \times \dim(\mathbf{h}^{(v)})}(\mathbf{A}_K)),$$

⋮

$$\mathbf{A}_1 = \sigma(\text{FC}_{|\mathcal{V}| \times \dim(\mathbf{h}^{(v)}) \rightarrow |\mathcal{V}| \times \dim(\mathbf{h}^{(v)})}(\mathbf{A}_0)),$$

$$\mathbf{A}_0 = \sigma(\text{FC}_{|\mathcal{V}| \times \dim(\mathbf{h}^{(v)}) \rightarrow |\mathcal{V}| \times \dim(\mathbf{h}^{(v)})}(\mathbf{H}(t))),$$



Graph neural controlled differential equations for traffic forecasting (AAAI 22)

- 空间尺度信息处理

$$\mathbf{Z}(T) = \mathbf{Z}(0) + \int_0^T g(\mathbf{Z}(t); \boldsymbol{\theta}_g) \frac{d\mathbf{H}(t)}{dt} dt,$$

空间信息的NCDE由时间尺度的隐藏状态控制

$$\mathbf{Z}(T) = \mathbf{Z}(0) + \int_0^T g(\mathbf{Z}(t); \boldsymbol{\theta}_g) f(\mathbf{H}(t); \boldsymbol{\theta}_f) \frac{d\mathbf{X}(t)}{dt} dt,$$

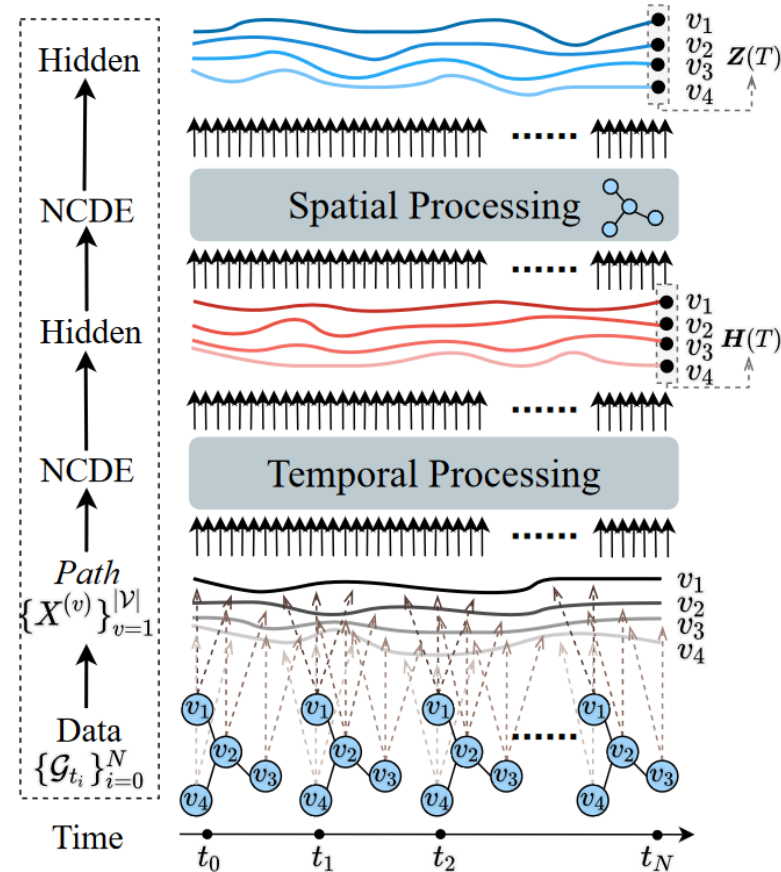
神经网络g如下，其中 $\mathbf{E} \in \mathbb{R}^{|\mathcal{V}| \times C}$ 是可训练的节点嵌入矩阵
 $\mathbf{W}_{\text{spatial}}$ 是一个可训练的权重变换矩阵。

$$g(\mathbf{Z}(t); \boldsymbol{\theta}_g) = \psi(\text{FC}_{|\mathcal{V}| \times \dim(\mathbf{z}^{(v)}) \rightarrow |\mathcal{V}| \times \dim(\mathbf{z}^{(v)})}(\mathbf{B}_1)), \quad (9)$$

$$\mathbf{B}_1 = (\mathbf{I} + \phi(\sigma(\mathbf{E} \cdot \mathbf{E}^\top))) \mathbf{B}_0 \mathbf{W}_{\text{spatial}}, \quad (10)$$

$$\mathbf{B}_0 = \sigma(\text{FC}_{|\mathcal{V}| \times \dim(\mathbf{z}^{(v)}) \rightarrow |\mathcal{V}| \times \dim(\mathbf{z}^{(v)})}(\mathbf{Z}(t))), \quad (11)$$

空间处理NCDE可被视为对图卷积网络的NCDE解释



• Graph neural controlled differential equations for traffic forecasting (AAAI 22)

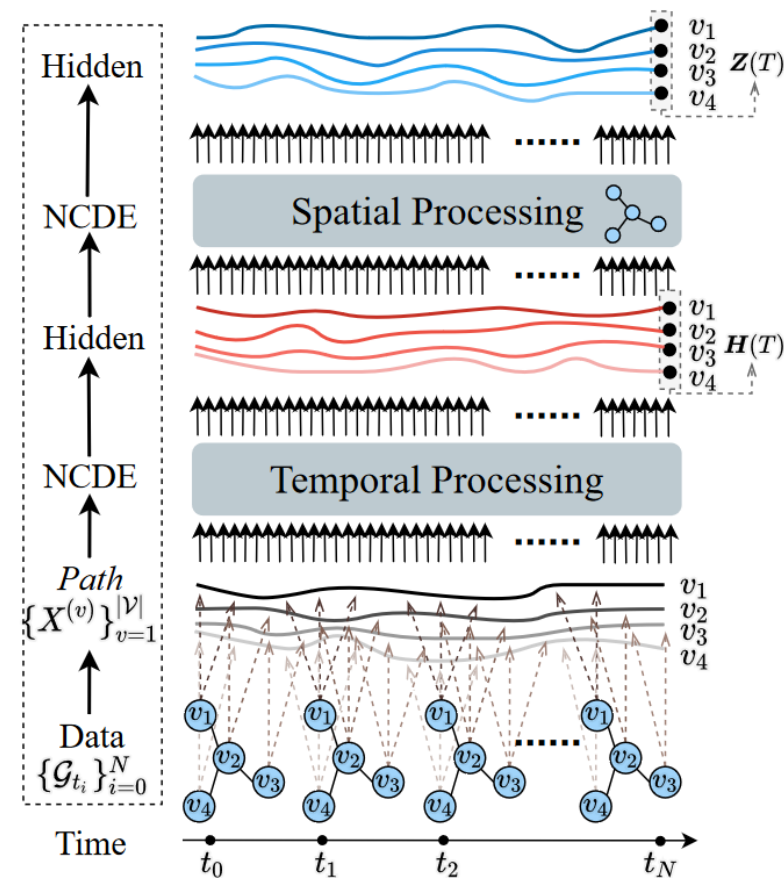
- 模型总体的受控常微分方程

$$\frac{d}{dt} \begin{bmatrix} \mathbf{Z}(t) \\ \mathbf{H}(t) \end{bmatrix} = \begin{bmatrix} g(\mathbf{Z}(t); \boldsymbol{\theta}_g) f(\mathbf{H}(t); \boldsymbol{\theta}_f) \frac{d\mathbf{X}(t)}{dt} \\ f(\mathbf{H}(t); \boldsymbol{\theta}_f) \frac{d\mathbf{X}(t)}{dt} \end{bmatrix},$$

- 实验结果

不规则时间尺度

Model	Missing rate	MAE	RMSE	MAPE
STG-NCDE		19.36	31.28	12.79%
Only Temporal	10%	26.26	40.89	17.66%
Only Spatial		19.73	31.67	13.20%
STG-NCDE		19.40	31.30	13.04%
Only Temporal	30%	26.86	41.73	18.35%
Only Spatial		19.83	31.95	13.29%
STG-NCDE		19.98	32.09	13.48%
Only Temporal	50%	28.15	43.54	19.14%
Only Spatial		20.14	32.30	13.30%



一：Neural ODE 神经常微分方程介绍

- 介绍Neural ODE 理论推导以及部分变体

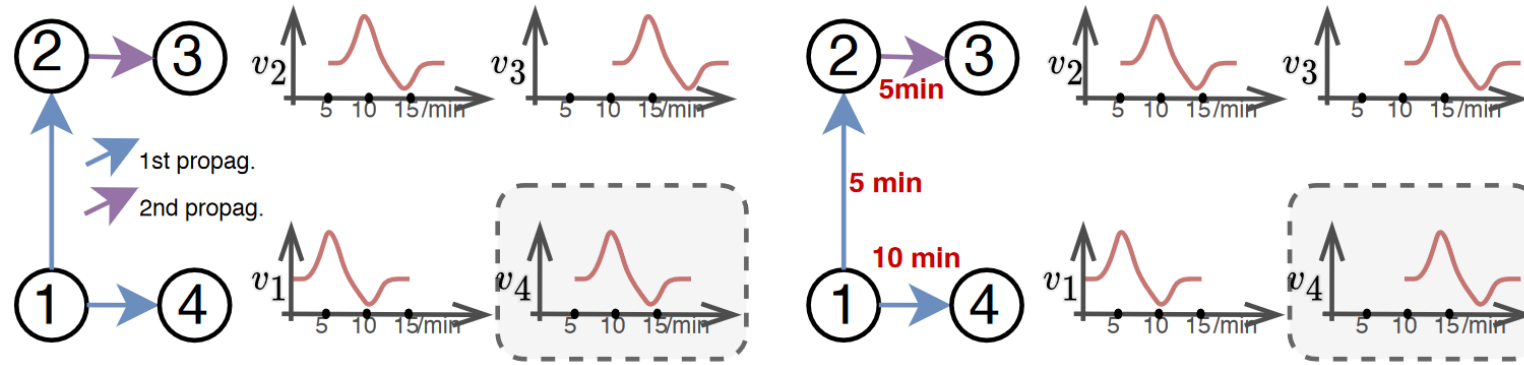
二：神经常微分方程在交通流预测的应用

- 原始ODE在交通流预测中的应用
- 受控神经常微分方程NCDE在交通流预测中的应用
- **延迟神经常微分方程NDDE在交通流预测中的应用**

三：神经常微分方程总结

- **Unveiling Delay Effects in Traffic Forecasting: A Perspective from Spatial-Temporal Delay Differential Equations(WWW 2024)**

- 图神经网络中节点信息传递有延迟效应



- 交通流量预测的频率根据具体需求可能不同

大多数现有的离散化模型需要为不同预测尺度重新训练

例如：给出10min间隔的历史数据 $\{t_0, t_{10}, t_{20}, \dots, t_n\}$ 预测5min间隔未来数据 $\{t_{n+5}, t_{n+10}, t_{n+15}, \dots\}$

• Neural Delay Differential Equation (NDDE)延迟神经微分方程

$$h(t) = \begin{cases} \phi(t), & t \leq 0, \\ h(0) + \int_0^T f(h(t), h(t - \tau), t, \theta) dt, & t > 0. \end{cases}$$

其中 τ 是延迟值， $\phi(t)$ 是历史函数。

• Neural Delay Differential Equation推导

- 从GRU进行推导

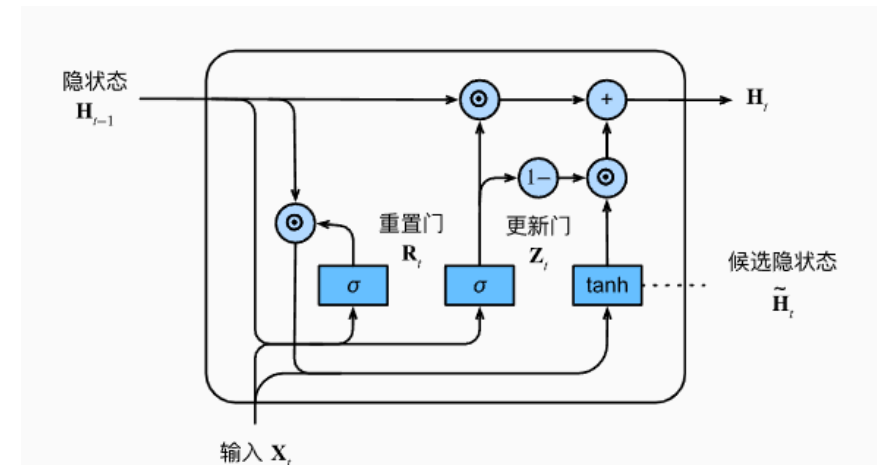
原始GRU

$$\mathbf{R}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xr} + \mathbf{H}_{t-1} \mathbf{W}_{hr} + \mathbf{b}_r),$$

$$\mathbf{Z}_t = \sigma(\mathbf{X}_t \mathbf{W}_{xz} + \mathbf{H}_{t-1} \mathbf{W}_{hz} + \mathbf{b}_z),$$

$$\tilde{\mathbf{H}}_t = \tanh(\mathbf{X}_t \mathbf{W}_{xh} + (\mathbf{R}_t \odot \mathbf{H}_{t-1}) \mathbf{W}_{hh} + \mathbf{b}_h),$$

$$\mathbf{H}_t = \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t.$$



本文简化的GRU

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t,$$

• Neural Delay Differential Equation推导

- 从GRU进行推导

本文简化的GRU

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t, \rightarrow T \text{时刻的更新向量}$$

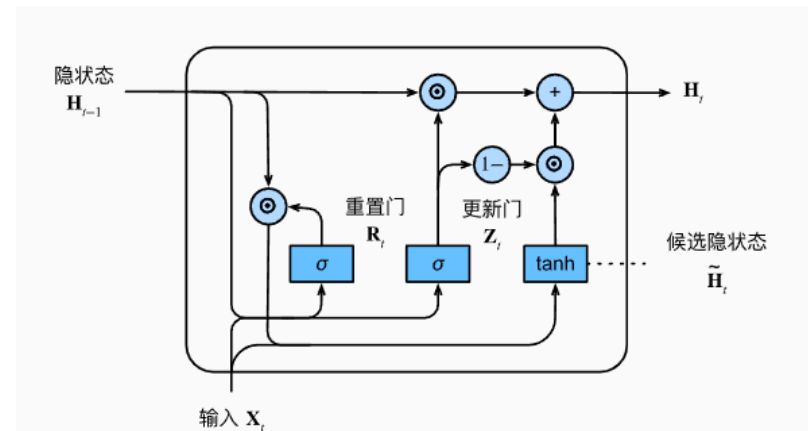


$$\Delta h = h_t - h_{t-1} = (1 - z_t) \odot (g_t - h_{t-1})$$



$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$

神经微分方程



• 时间延迟因素

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t, \rightarrow \text{T时刻的更新向量}$$



$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$

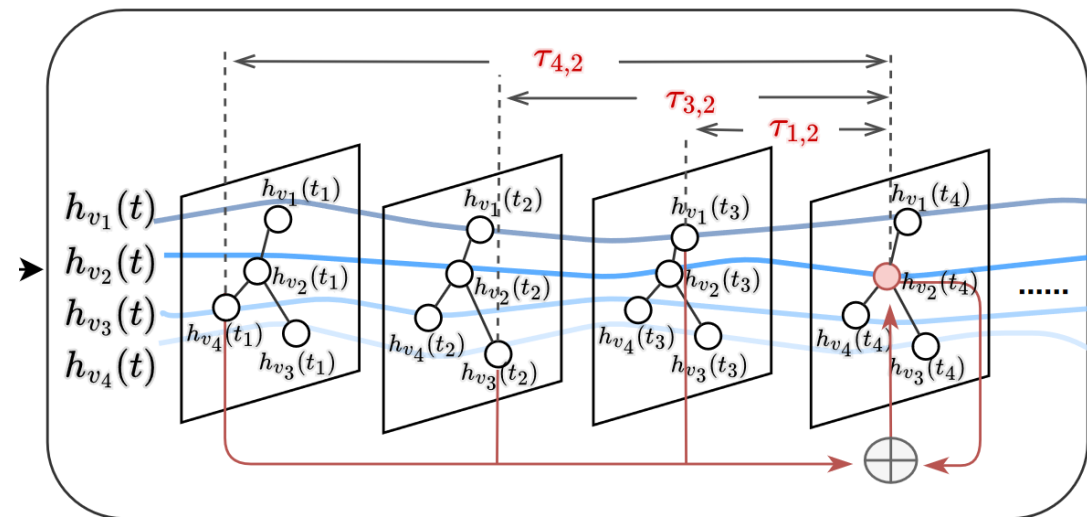
神经微分方程

- 更新向量g

用图神经网络提取空间特征，考虑时延计算更新向量

$$g_i(t) = c \sum_{j \in \mathcal{N}(i)} \alpha_{ij} f(h_j(t - \tau_{ij})),$$

其中 α_{ij} 和 τ_{ij} 分别是节点 v_i 和 v_j 之间的边权重和延迟值





• 时间延迟估计器

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)), \quad g_i(t) = c \sum_{j \in \mathcal{N}(i)} \alpha_{ij} f(h_j(t - \tau_{ij})),$$

- 方法一：最大化互相关（MCC）在预处理阶段得到

$$\tau_{ij} = \arg \max_k \text{corr}(\tilde{x}_i^{\rightarrow k}, \tilde{x}_j), \quad \text{corr()是皮尔逊相关函数}$$

- 方法二： τ_{ij} 作为可学参数

每条边设置两个可学习的延迟参数：高峰时段，非高峰时段。

一般来说， τ 在这个上下文中被认为是不可学习的：模型无法计算 τ 的梯度

但是由于模型在时间维度上是连续的，并且 $\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t))$.

就可以通过显式定义 h 对 τ 的反向传播

• CDE加入

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)).$$



控制函数X加入

$$X(t) = \Phi \left(\{x^0, t_0\}, \{x^1, t_1\}, \dots, \{x^n, t_n\} \right)$$

$$\frac{dh_i(t)}{dt} = (1 - z_i(t)) \odot (g_i(t) - h_i(t)) \tilde{f} \left(\frac{dX_i(t)}{dt} \right)$$

- 最终模型的形式化表达

$$g_i(t) = c \sum_{j \in \mathcal{N}(i)} \alpha_{ij} f(h_j(t - \tau_{ij})),$$

$$z_i(t) = \sigma(W_z h_i(t) + U_z g_i(t) + b_z),$$

$$\frac{dh_i(t)}{dt} = (1 - z_i(t)) \odot (g_i(t) - h_i(t)) \tilde{f} \left(\frac{dX_i(t)}{dt} \right),$$

$$h_i(t) = \begin{cases} \phi_i(t), & t \leq 0 \\ h_i(0) + \int_0^t \frac{dh_i(t)}{dt} dt, & t > 0. \end{cases}$$

• 连续输出模块

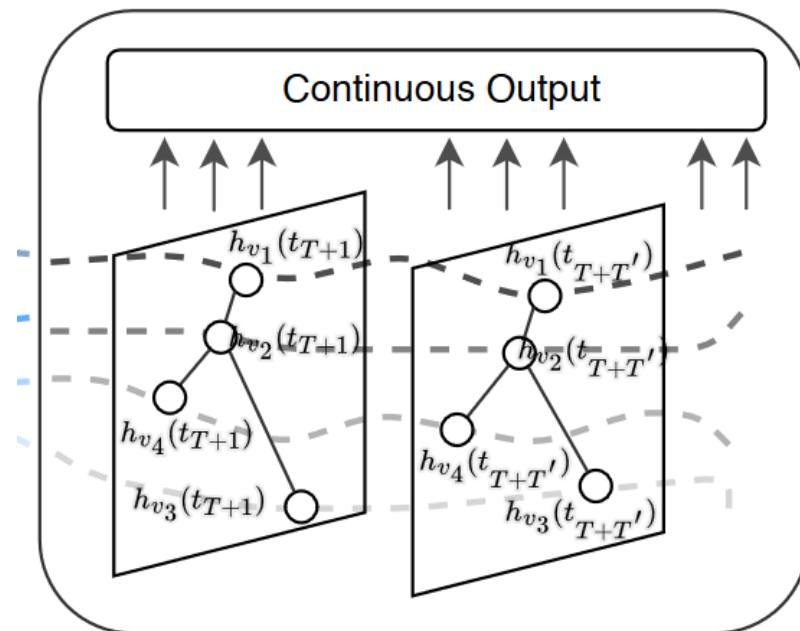
- 直接使用另外一个STDDE来进行输出

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t))$$

$$h_i(t') = h_i(T) + \int_T^{t'} \frac{dh_i(t)}{dt} dt,$$

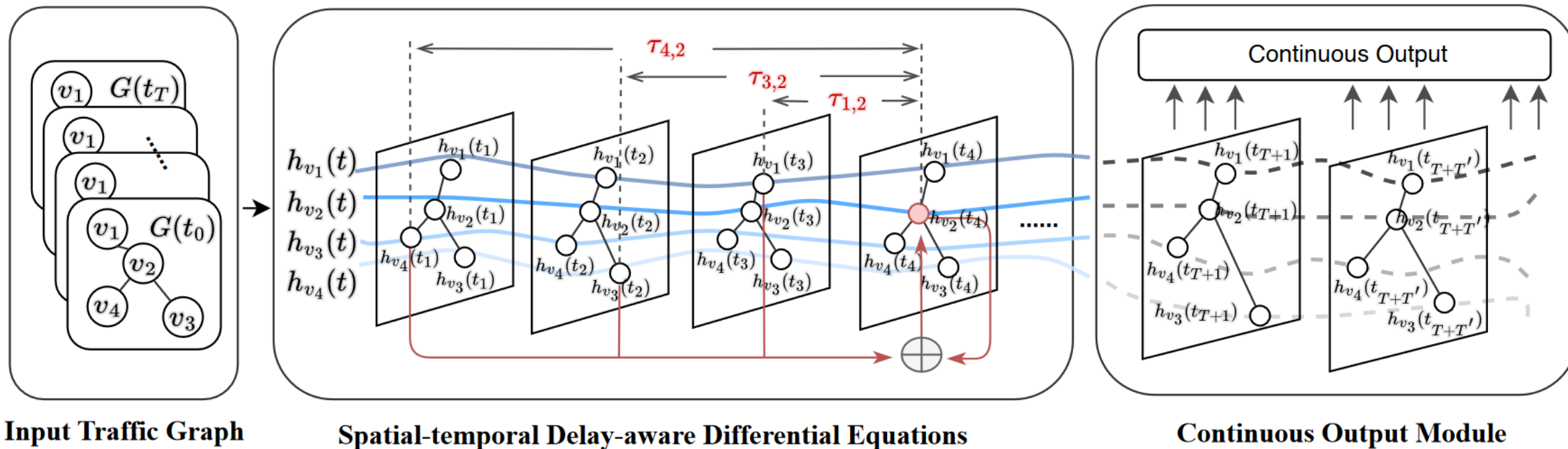
$$y_i(t') = f(h_i(t')),$$

$$Y_i = [y_i(t_{T+1}), y_i(t_{T+2}), \dots, y_i(t_{T+T'})]$$



Continuous Output Module

• 模型总框架



• 实验

节点平均延迟值排序

Dataset	Metric	STGCN	DCRNN	ASTGCN(r)	STSGCN	STGODE	AGCRN	STG-NCDE	DSTAGNN	FOGS	PDFormer	STDDE
PeMSD7(M)	RMSE	7.55	7.18	6.87	5.93	5.66	5.54	5.39	5.54	5.54	5.60	5.19
	MAE	4.01	3.83	3.61	3.01	2.97	2.79	2.68	2.78	2.76	2.81	2.56
	MAPE	9.67	9.81	8.84	7.55	7.36	7.02	6.76	6.93	6.83	7.06	6.61
PeMSD7(L)	RMSE	8.28	8.33	7.64	6.88	5.98	5.92	5.76	5.98	6.04	5.90	5.63
	MAE	4.84	4.33	4.09	3.61	3.22	2.99	2.87	2.98	2.96	2.92	2.77
	MAPE	11.76	11.41	10.25	9.13	7.94	7.59	7.31	7.50	7.48	7.54	7.26
PeMS03	RMSE	30.42	30.31	29.56	29.21	27.84	28.25	27.09	27.39	24.85	25.96	24.52
	MAE	17.55	17.99	17.34	17.48	16.50	15.98	15.57	15.62	15.06	14.95	15.03
	MAPE	17.43	18.34	17.21	16.78	16.69	15.23	15.06	14.74	15.03	15.58	14.69
PeMS04	RMSE	36.01	37.65	35.22	33.65	32.82	32.26	31.09	31.71	31.29	29.96	29.86
	MAE	22.66	24.63	22.94	21.19	20.84	19.83	19.21	19.38	19.44	18.31	18.11
	MAPE	14.34	17.01	16.43	13.90	13.77	12.97	12.76	12.77	12.81	12.07	12.07
PeMS07	RMSE	39.34	38.61	37.87	39.03	37.54	36.55	33.84	34.88	34.09	32.80	32.59
	MAE	25.33	25.22	24.01	24.26	22.99	22.37	20.53	21.62	20.79	19.78	19.47
	MAPE	11.21	11.82	10.73	10.21	10.14	9.12	8.80	9.24	8.75	8.54	8.49
PeMS08	RMSE	27.88	27.83	26.22	26.80	25.97	25.22	24.81	25.08	25.36	24.61	24.31
	MAE	18.11	17.46	16.64	17.13	16.81	15.95	15.45	15.85	16.10	15.66	15.12
	MAPE	11.34	11.39	10.6	10.96	10.62	10.09	9.92	9.93	9.85	9.61	9.74

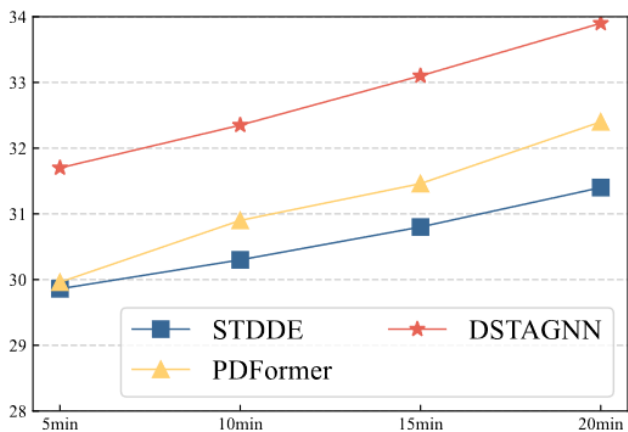
Table 2: Performance comparison of baselines and proposed STDDE on six popular used real-world traffic datasets.

• 实验

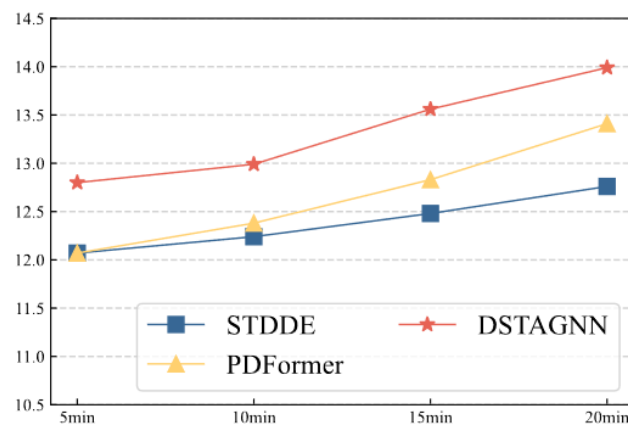
节点平均延迟值排序

Data	Metric	STDDE-no-delay	STDDE	Gain
First 15%	RMSE	16.97	16.86	0.65%
	MAE	11.54	11.47	0.61%
	MAPE	19.71	18.37	6.80%
Last 15%	RMSE	37.72	34.59	8.30%
	MAE	25.06	23.24	7.26%
	MAPE	14.63	13.96	4.65%

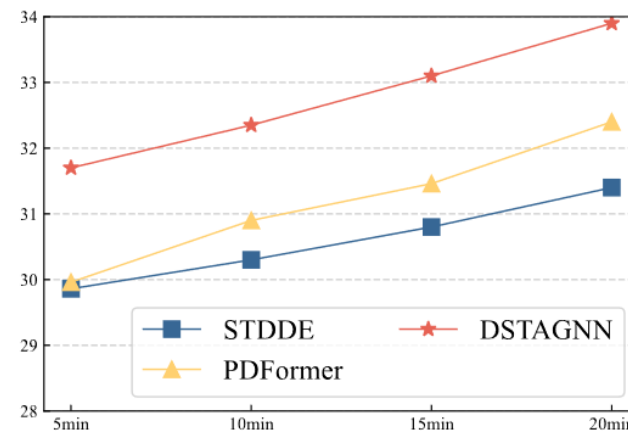
历史60分钟的数据来预测未来60分钟的交通流量，时间间隔{5, 10, 15, 20} -> {5}



(a) RMSE



(b) MAE



(c) MAPE

一：Neural ODE 神经常微分方程介绍

- 介绍Neural ODE 理论推导以及部分变体

二：神经常微分方程在交通流预测的应用

- 原始ODE在交通流预测中的应用
- 受控神经常微分方程NCDE在交通流预测中的应用
- 延迟神经常微分方程NDDE在交通流预测中的应用

三：神经常微分方程总结

Neural ODE

$$\frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

$$\mathbf{h}(t_1) = \mathbf{h}(t_0) + \int_{t_0}^{t_1} f(\mathbf{h}(t), t, \theta) dt$$

Neural CDE

$$h_{t_n} = h_{t_0} + \int_{t_0}^{t_n} f(h(t), t, \theta) dX_t$$

Neural DDE

$$\begin{cases} \frac{dh_t}{dt} = f(h_t, h_{t-\tau}, t; \mathbf{w}), & t \geq 0, \\ h(t) = \phi(t), & t \leq 0, \end{cases}$$

Neural SDE

(Stochastic differential equations)

随机微分方程

$$\begin{aligned} y(0) &= \zeta_{\theta}(v), \\ dy(t) &= \mu_{\theta}(t, y(t)) dt + \sigma_{\theta}(t, y(t)) \circ dw(t), \\ x(t) &= \alpha_{\theta} y(t) + \beta_{\theta}, \end{aligned}$$

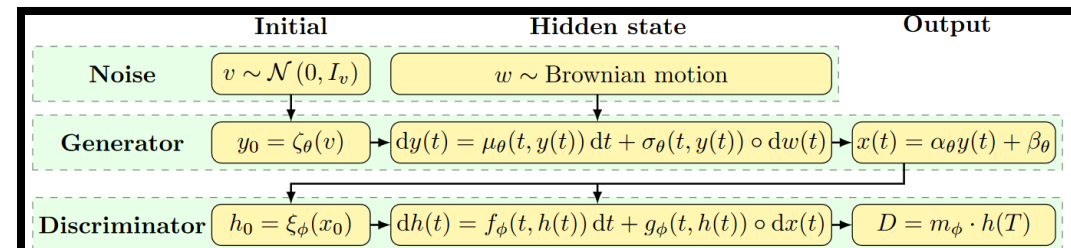


Figure 4.2: Summary of equations for an SDE-GAN.

连续深度的ResNet

一个残差块的形式化表达如下

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, \theta_t)$$

进行数学变换

$$\lim_{\Delta t \rightarrow 0} \frac{\mathbf{h}_{t+1} - \mathbf{h}_t}{\Delta t} = \frac{d\mathbf{h}(t)}{dt} = f(\mathbf{h}(t), t, \theta)$$

连续深度的GCN

原始GCN的表示

$$H_{l+1} = GCN(H_l) = \sigma(\hat{A}H_lW)$$

基于张量的GCN 离散形式

$$\mathcal{H}_{l+1} = \mathcal{H}_l \times_1 \hat{A} \times_2 U \times_3 W + \mathcal{H}_0$$

推导的ODE

$$\frac{d\mathcal{H}(t)}{dt} = \mathcal{H}(t) \times_1 (\hat{A} - I) + \mathcal{H}(t) \times_2 (U - I) + \mathcal{H}(t) \times_3 (W - I) + \mathcal{H}_0$$
$$\mathcal{H}(t) = ODESolve\left(\frac{d\mathcal{H}(t)}{dt}, \mathcal{H}_0, t\right)$$

连续时间的RNN

本文简化的GRU

$$z_t = \sigma(W_z h_{t-1} + U_z g_t + b_z),$$
$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot g_t, \rightarrow \text{T时刻的更新向量}$$

↓

$$\Delta h = h_t - h_{t-1} = (1 - z_t) \odot (g_t - h_{t-1})$$

↓

$$\frac{dh(t)}{dt} = (1 - z(t)) \odot (g(t) - h(t)). \quad \text{神经常微分方程}$$



Neural ODE 理论相关文献

Neural ODE:

Chen R T Q, Rubanova Y, Bettencourt J, et al. Neural ordinary differential equations. (NeurIPS 2018)

Neural CDE:

Kidger P, Morrill J, Foster J, et al. Neural controlled differential equations for irregular time series. (NeurIPS 2020)

Neural DDE:

Zhu Q, Guo Y, Lin W. Neural delay differential equations.(arXiv 2021)

Neural SDE:

Liu X, Xiao T, Si S, et al. Neural sde: Stabilizing neural ode networks with stochastic noise. (arXiv 2019)

综述:

Kidger P. On neural differential equations. (arXiv 2022) 博士学位论文



谢谢大家!
敬请批评指正!

