

神经网络求解偏微分方程

黄政宇

北京大学北京国际数学研究中心

北京大学国际机器学习研究中心



本堂课大纲

- 偏微分方程求解
 - 物理信息神经网络
 - 有结构的偏微分方程的求解

- 部分未知的偏微分方程求解
 - 物理信息神经网络
 - 特殊的神经网络



回顾：随机特征方法

➤ 偏微分方程问题

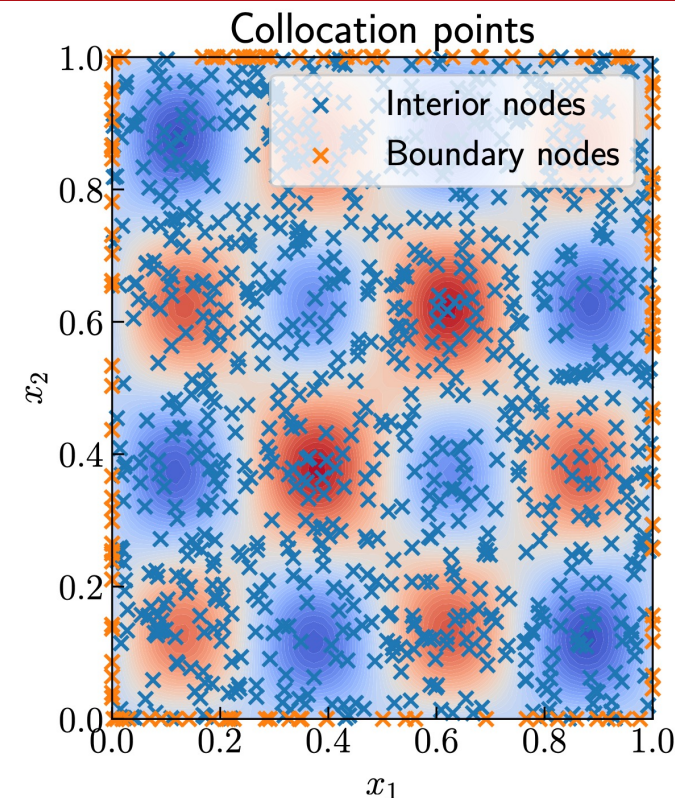
$$\begin{aligned}\mathcal{L}(x, u) &= 0 & x \in \Omega \\ \mathcal{B}(x, u) &= 0 & x \in \partial\Omega\end{aligned}$$

➤ 随机特征方法

$$\hat{u}(x) = \sum_{i=1}^D c_i \psi(x, \omega_i)$$

$$\min \sum_{x_i \in X^{int}} \lambda_{Ii} \|\mathcal{L}(x_i, u(x_i))\|^2 + \sum_{x_i \in X^{bd}} \lambda_{Bi} \|\mathcal{B}(x_i, u(x_i))\|^2 + \lambda \|c\|_2^2$$

二次优化问题





物理信息神经网络 (PINN)

➤ 偏微分方程问题

$$\begin{aligned}\mathcal{L}(x, u) &= 0 & x \in \Omega \\ \mathcal{B}(x, u) &= 0 & x \in \partial\Omega\end{aligned}$$

➤ 物理信息神经网络(Physics informed neural network, PINN)

$$\hat{u}(x) = \mathcal{NN}(x; \theta)$$

$$\begin{aligned}\min_{\theta} \sum_{x_i \in X^{int}} \lambda_{Ii} \|\mathcal{L}(x_i, \mathcal{NN}(x_i; \theta))\|^2 \\ + \sum_{x_i \in X^{bd}} \lambda_{Bi} \|\mathcal{B}(x_i, \mathcal{NN}(x_i; \theta))\|^2\end{aligned}$$



有结构的偏微分方程

► 变分问题 (弱解)

$$\min \int I(u)$$

Poisson问题Neumann边界条件：

$$-\Delta u = f \quad \frac{\partial u}{\partial n} = g$$

$$I(u) = \int_{\Omega} \frac{1}{2} |\nabla u|^2 - f(x)u(x) dx - \int_{\partial\Omega} g(x)u(x) dx$$

Poisson问题Dirichlet边界条件：

$$-\Delta u = f \quad u = g$$

$$I(u) = \int_{\Omega} \frac{1}{2} |\nabla u|^2 - f(x)u(x) dx + \int_{\partial\Omega} |u - g|^2 dx$$



有结构的偏微分方程

► 变分问题 (弱解)

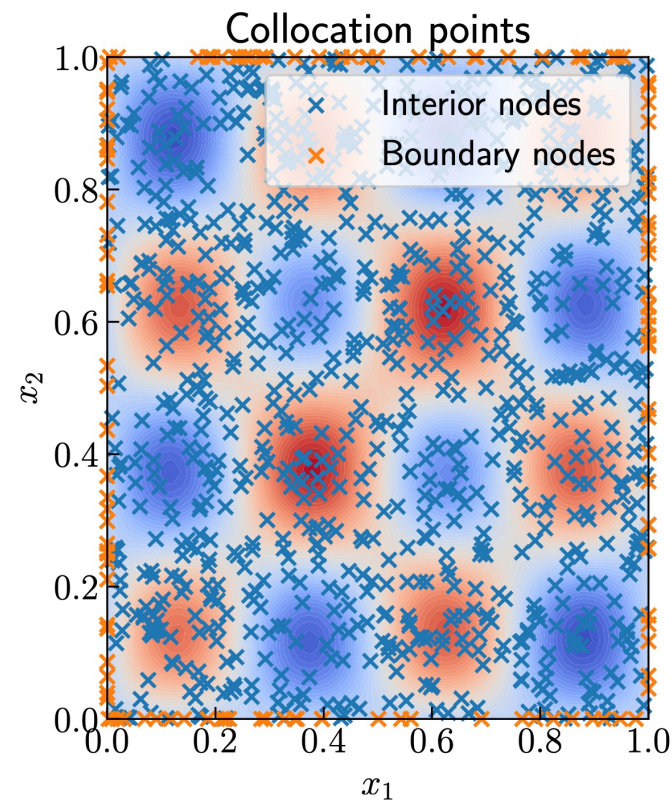
高维Poisson问题：

$$-\Delta u = -200 \quad x \in (0,1)^d$$

$$u = \sum_{i=1}^d x_i^2 \quad x \in \partial(0,1)^d$$

三层残差神经网络($D = 100$)

每一步随机梯度下降，随机采样，
1000 个点在 Ω 里面 100 个点在边界上 $\partial\Omega$



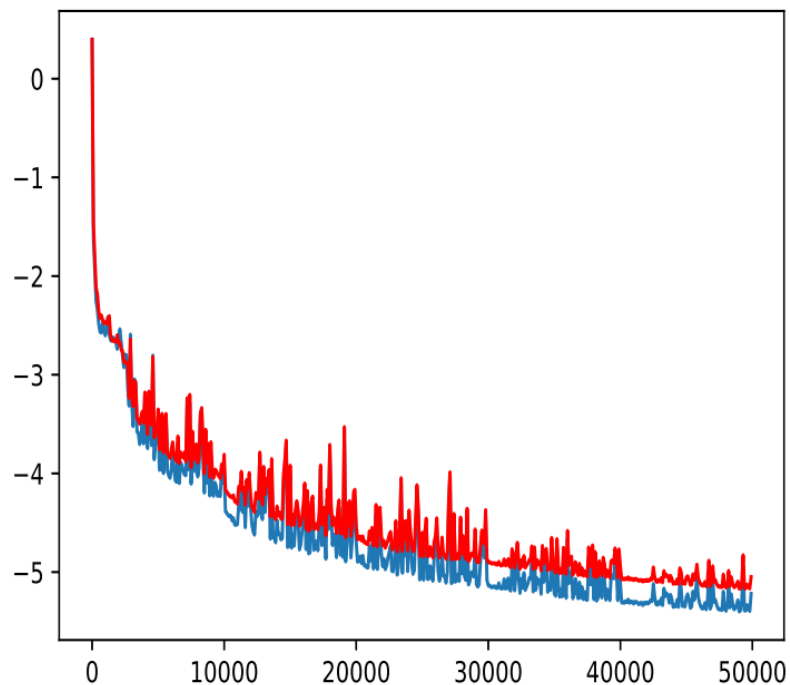


有结构的偏微分方程

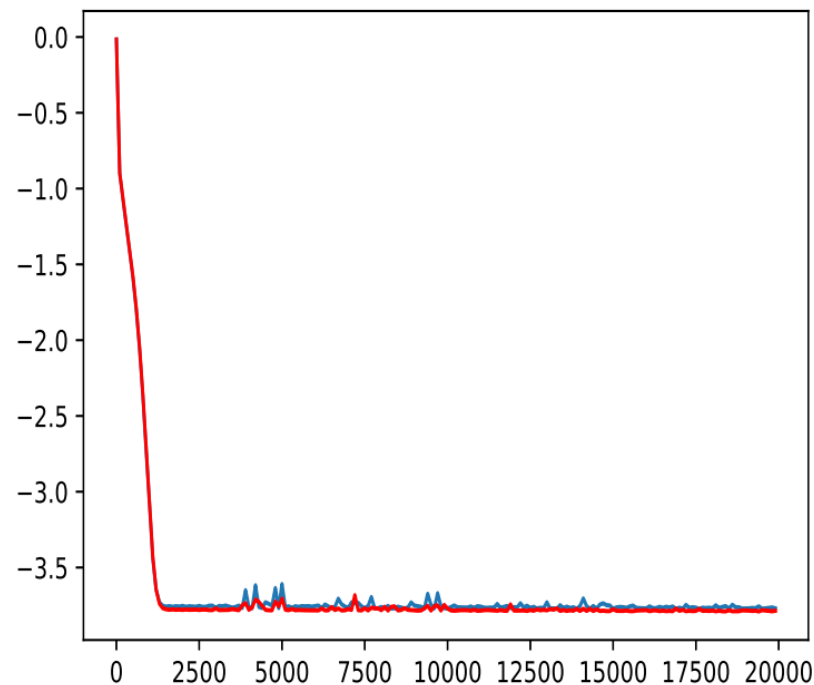
变分问题 (弱解)

高维Poisson问题：

$d = 10$



$d = 100$



红色：边界条件相对误差(ln)

蓝色：整体相对误差(ln)



有结构的偏微分方程

➤ 半线性抛物型偏微分方程

$$\frac{\partial u(t, x)}{\partial t} + \nabla u(t, x) \cdot \mu(t, x) + \frac{1}{2} \text{Tr}[\sigma \sigma^T : \nabla^2 u(t, x)] + f(t, x, u(t, x), \sigma^T \nabla u(t, x)) = 0$$

- 最高阶导数（即二阶导数）部分是线性的
- 对最高阶导数（即二阶导数）是抛物方程
- 它的非线性项里只出现函数及其一阶导数项

给定 $u(T, x) = g(x)$

求解 $u(0, x)$



有结构的偏微分方程

➤ 半线性抛物型偏微分方程

$$\frac{\partial u(t, x)}{\partial t} + \nabla u(t, x) \cdot \mu(t, x) + \frac{1}{2} \text{Tr}[\sigma \sigma^T : \nabla^2 u(t, x)] + f(t, x, u(t, x), \sigma^T \nabla u(t, x)) = 0$$

- Black-Scholes 模型

$$\frac{\partial u(t, x)}{\partial t} + \nabla u(t, x) \cdot \bar{\mu}x + \frac{\bar{\sigma}^2}{2} \sum_{i=1}^d |x_i|^2 \frac{\partial^2 u(t, x)}{\partial x_i^2} = f(u(t, x))$$

- Hamilton-Jacobi 方程

$$\frac{\partial u(t, x)}{\partial t} + \Delta u(t, x) - \lambda \|\nabla u(t, x)\|^2 = 0$$



有结构的偏微分方程

➤ 半线性抛物型偏微分方程

$$\frac{\partial u(t, x)}{\partial t} + \nabla u(t, x) \cdot \mu(t, x) + \frac{1}{2} \text{Tr}[\sigma \sigma^T : \nabla^2 u(t, x)] + f(t, x, u(t, x), \sigma^T \nabla u(t, x)) = 0$$

方程的解：

$$X_t = X_0 + \int_0^t \mu(s, X_s) ds + \int_0^t \sigma(s, X_s) dW_s$$
$$u(t, X_t) = u(0, X_0) - \int_0^t f(s, X_s, u(s, X_s), \sigma(s, X_s)^T \nabla u(s, X_s)) ds + \int_0^t \nabla u(s, X_s)^T \sigma(s, X_s) dW_s$$



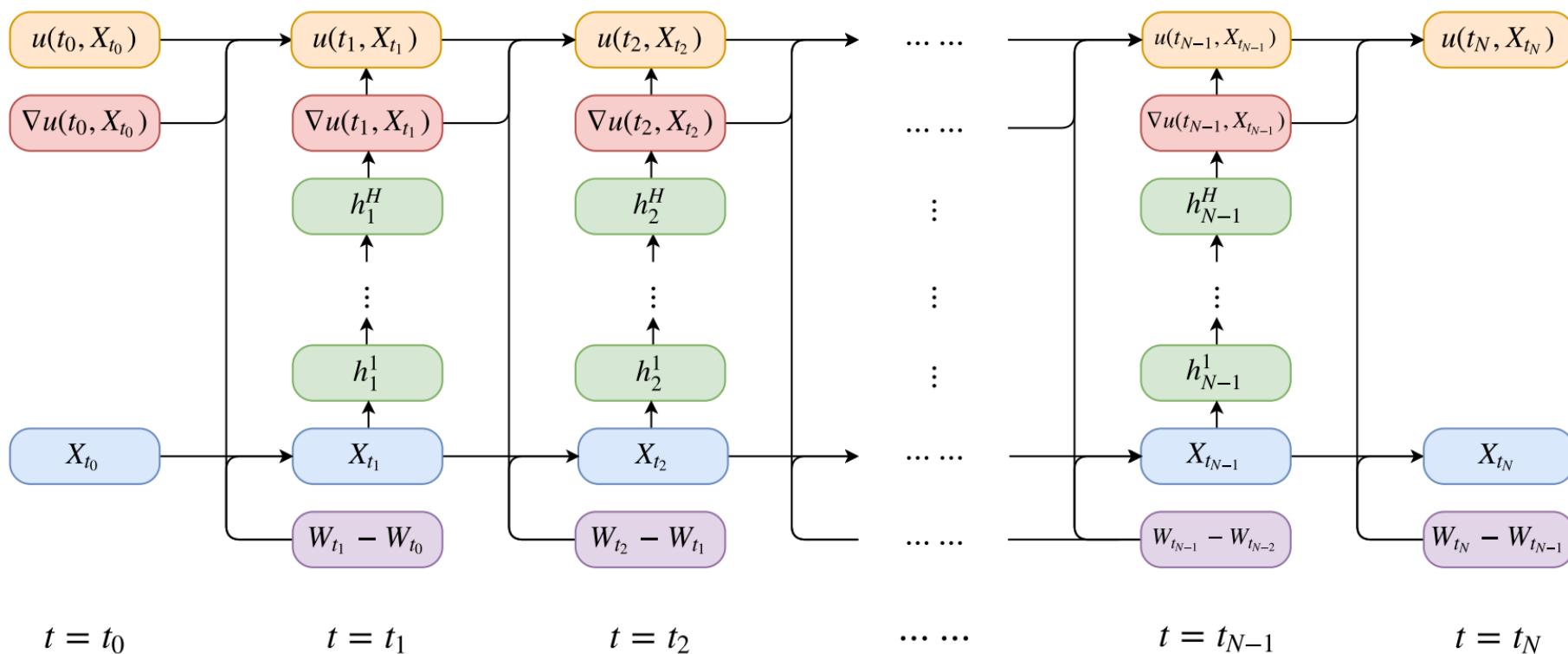
有结构的偏微分方程

➤ 半线性抛物型偏微分方程

(目标) 未知参数： $\theta_{u_0} = u(0, x)$ $\theta_{\nabla u_0} = \nabla u(0, x)$

神经网络： $\sigma(t_i, x)^T \nabla u(t_i, x) = \mathcal{NN}(t_i, x | \theta_i)$

损失函数： $l(\theta) = \mathbb{E}[|g(X_{t_n}) - \hat{u}|]$



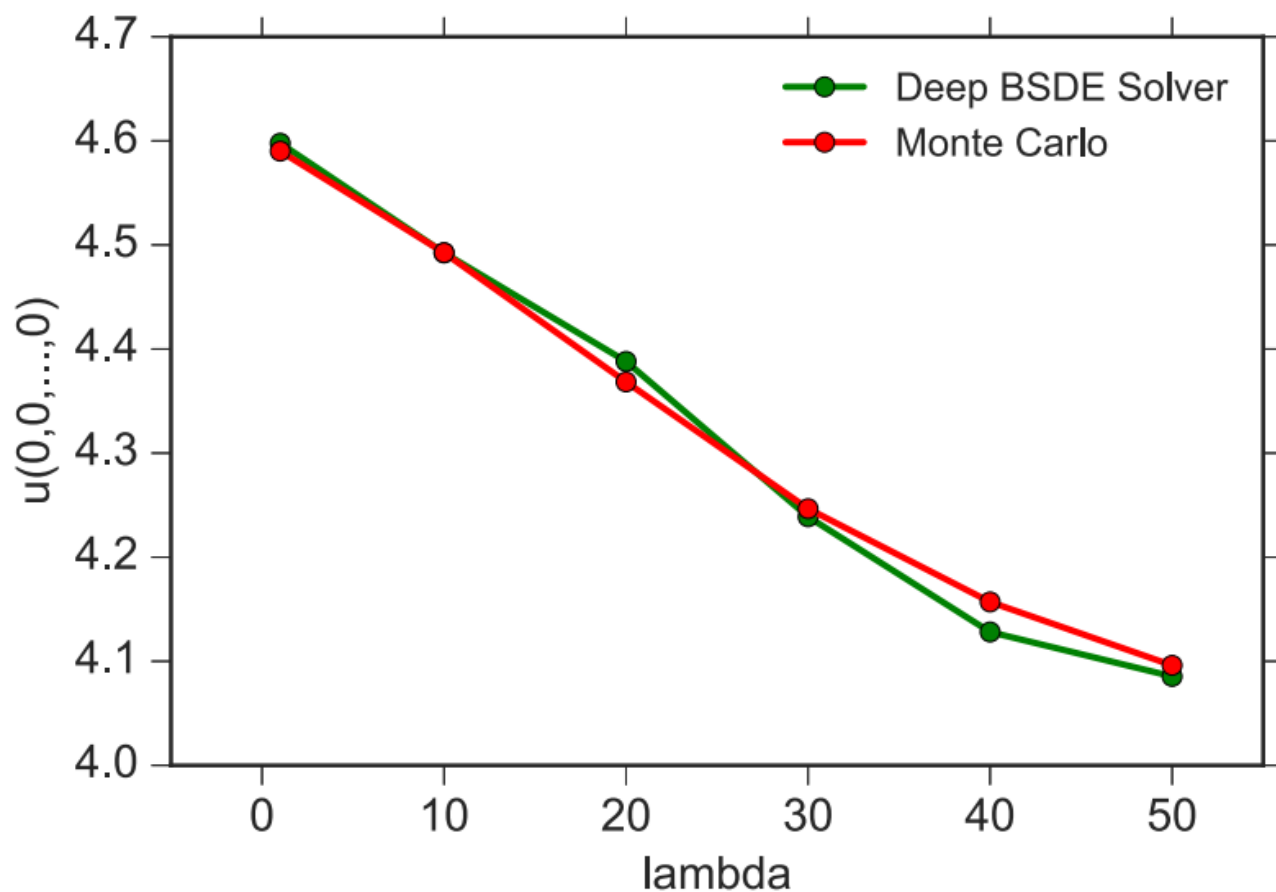


有结构的偏微分方程

➤ 半线性抛物型偏微分方程

Hamilton-Jacobi 方程：

$$\frac{\partial u(t,x)}{\partial t} + \Delta u(t,x) - \lambda \|\nabla u(t,x)\|^2 = 0$$





物理信息神经网络 (PINN)

➤ 优点

- 有潜力求解高维偏微分方程

➤ 缺点

- 方法对超参数(例如 λ_{Ii} , λ_{Bi})比较敏感性
- 处理不连续问题仍具有挑战
- 低维问题计算没有传统方法高效，尚未用于求解大规模的问题



本堂课大纲

- 偏微分方程求解
 - 物理信息神经网络
 - 有结构的偏微分方程的求解

- 部分未知的偏微分方程求解
 - 物理信息神经网络
 - 特殊的神经网络



部分未知的偏微分方程

➤ 偏微分方程

$$\mathcal{L}(x, u, \mathcal{M}) = 0 \quad x \in \Omega$$

$$\mathcal{B}(x, u, \mathcal{M}) = 0 \quad x \in \partial\Omega$$

\mathcal{M} 未知， \mathcal{M} 是参数或者是关于 x 和 u 的函数

➤ 数据

间接数据(indirect data) : $\{x_i, u_i = u(x_i)\}_{i=1}$

直接数据 : $\{x_i, \mathcal{M}_i = \mathcal{M}(x_i)\}_{i=1}$



部分未知的偏微分方程

➤ 反问题

$$\min_{\mathcal{M}} \sum \lambda_i \|u(x_i) - u_i\|^2$$

$$\mathcal{L}(x, u, \mathcal{M}) = 0 \quad x \in \Omega$$

$$\mathcal{B}(x, u, \mathcal{M}) = 0 \quad x \in \partial\Omega$$

偏微分方程约束的优化问题

- 伴随求解器计算导数
- 反复求解偏微分方程系统
- 适定性



物理信息神经网络 (PINN)

➤ 反问题 (\mathcal{M} 是未知参数)

$$\hat{u}(x) = \mathcal{N}\mathcal{N}(x; \theta)$$

$$\begin{aligned} \min_{\theta, \mathcal{M}} \quad & \sum_{x_i \in X^{int}} \lambda_{Ii} \|\mathcal{L}(x_i, \mathcal{N}\mathcal{N}(x_i; \theta), \mathcal{M})\|^2 \\ & + \sum_{x_i \in X^{bd}} \lambda_{Bi} \|\mathcal{B}(x_i, \mathcal{N}\mathcal{N}(x_i; \theta), \mathcal{M})\|^2 \\ & + \sum \lambda_i \|\mathcal{N}\mathcal{N}(x_i; \theta) - u_i\|^2 \end{aligned}$$



物理信息神经网络 (PINN)

➤ 反问题 (\mathcal{M} 是未知函数)

$$\hat{u}(x) = \mathcal{N}\mathcal{N}(x; \theta_1) \quad \mathcal{M} = \mathcal{M}(x, \hat{u}(x), \theta_2)$$

$$\begin{aligned} \min_{\theta_1, \theta_2} & \sum_{x_i \in X^{int}} \lambda_{Ii} \|\mathcal{L}(x_i, \mathcal{N}\mathcal{N}(x_i; \theta_1), \mathcal{M}(x, \mathcal{N}\mathcal{N}(x_i; \theta_1), \theta_2))\|^2 \\ & + \sum_{x_i \in X^{bd}} \lambda_{Bi} \|\mathcal{B}(x_i, \mathcal{N}\mathcal{N}(x_i; \theta_1), \mathcal{M}(x, \mathcal{N}\mathcal{N}(x_i; \theta_1), \theta_2))\|^2 \\ & + \sum \lambda_i \|\mathcal{N}\mathcal{N}(x_i; \theta_1) - u_i\|^2 \end{aligned}$$



部分未知的偏微分方程

➤ 例子：2维不可压Navier-Stokes方程

$$u_t + \lambda_1(uu_x + vu_y) = -p_x + \lambda_2(u_{xx} + u_{yy})$$

$$v_t + \lambda_1(uv_x + vv_y) = -p_y + \lambda_2(v_{xx} + v_{yy})$$

$$u_x + v_y = 0 \Rightarrow u = \psi_y, v = -\psi_x$$

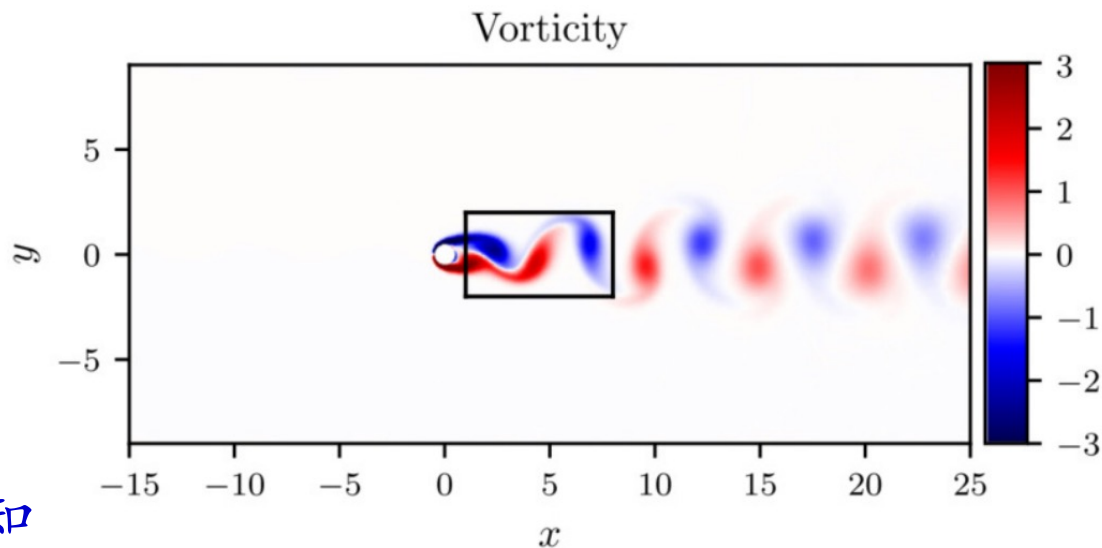
其中 $u(t, x, y), v(t, x, y)$ 是速度， $p(t, x, y)$ 是压强， $\psi(t, x, y)$ 是流函数(stream function)。

我们考虑用神经网络近似 $\psi(t, x, y)$ 和 $p(t, x, y)$ ，同时估计参数 λ_1 和 λ_2



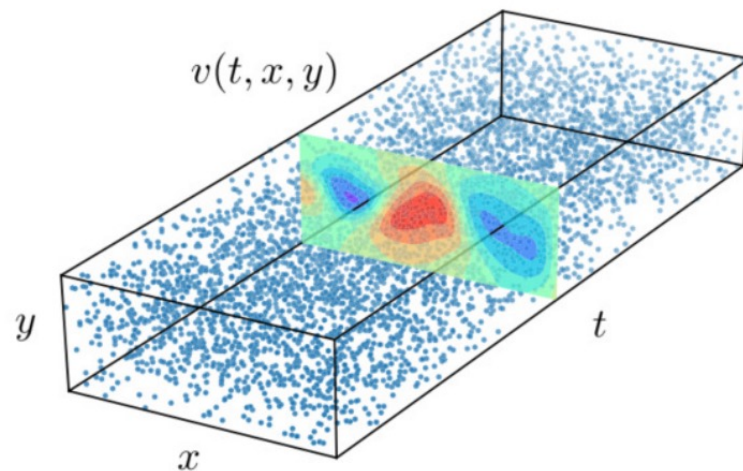
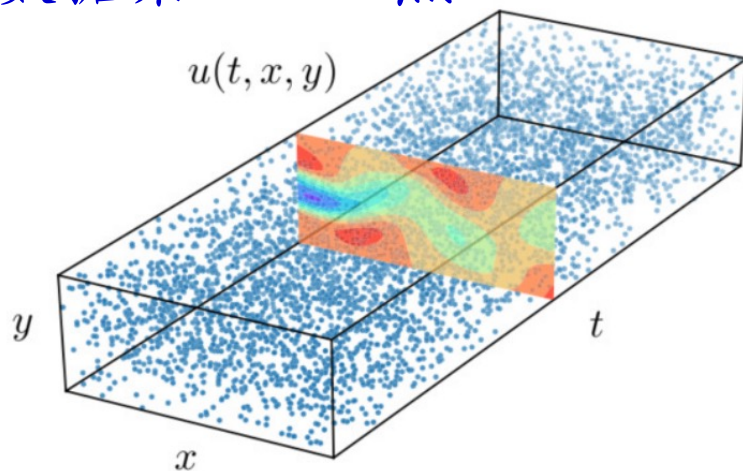
部分未知的偏微分方程

➤ 例子：2维不可压Navier-Stokes方程



边界条件未知

训练数据集：5000点~1%

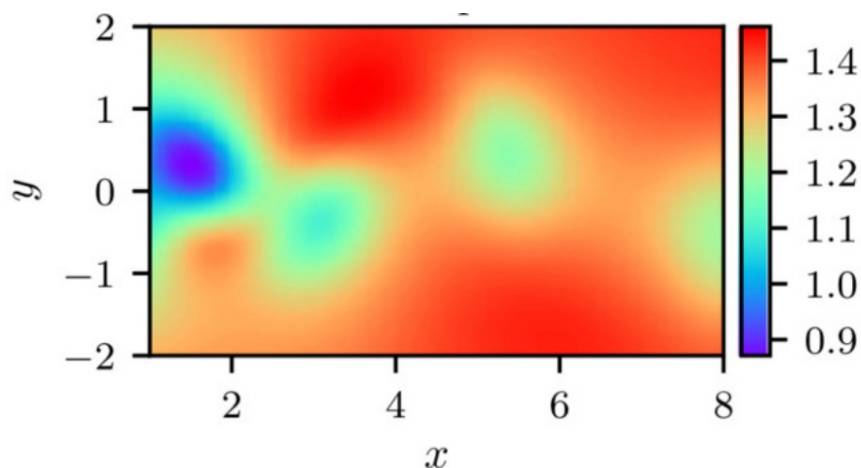




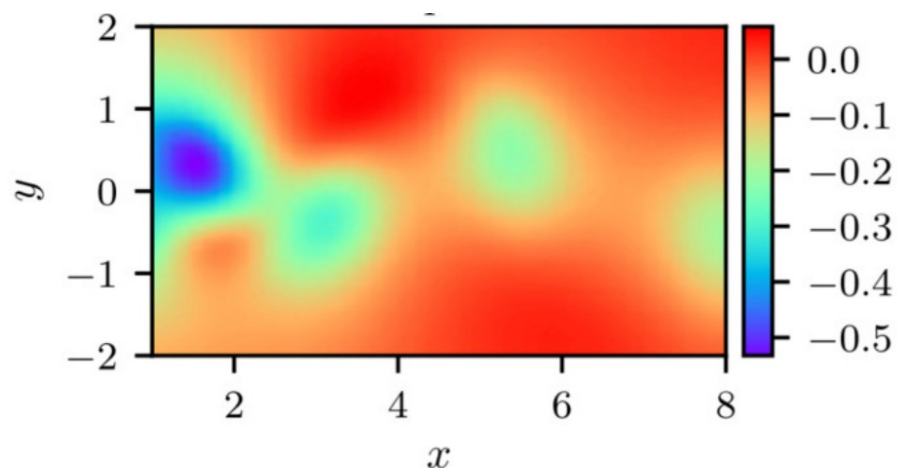
部分未知的偏微分方程

➤ 例子：2维不可压Navier-Stokes方程

预测的压力场真实



真实的压力场



真实的 $\lambda_1 = 1$ 和 $\lambda_2 = 0.01$

反演出来的的 $\lambda_1 = 0.999$ 和 $\lambda_2 = 0.01047$

当观测有1%相对误差，反演出来的的 $\lambda_1 = 0.998$ 和 $\lambda_2 = 0.01057$



部分未知的偏微分方程

➤ 偏微分方程

$$\begin{aligned}\mathcal{L}(x, u, \mathcal{M}) &= 0 & x \in \Omega \\ \mathcal{B}(x, u, \mathcal{M}) &= 0 & x \in \partial\Omega\end{aligned}$$

➤ 模型不确定性

方程中有一些建模未知， \mathcal{M} 是关于 x 和 u 的函数
神经网络建模： $\mathcal{M} = \mathcal{M}(x, u, \theta)$

建模 \mathcal{M} 需要满足一些物理性质



特殊的神经网络

➤ 凸神经网络

材料力学本构关系 $\hat{S} = \mathcal{M}(\hat{E})$:

应力 $\hat{E} = [E_{11}, E_{22}, E_{33}, 2E_{23}, 2E_{13}, 2E_{12}]$

应变 $\hat{S} = [S_{11}, S_{22}, S_{33}, S_{23}, S_{13}, S_{12}]$

能量泛函数 :

$$\hat{S} = \frac{dW(\hat{E})}{d\hat{E}}$$

给定边界条件，寻找能量最小状态

$W(\hat{E})$ 是凸函数



特殊的神经网络

凸神经网络 (Amos 2017)

对于神经网络 $y = f(x; \theta) = f_L \circ f_{L-1} \circ \dots \circ f_1(x)$
: $R^{n_0} \rightarrow R^{n_L}$, 其中 $f_i(z) = h^{(i)}(W^{(i)}z + \tilde{W}^{(i)}x + b^{(i)})$,

如果

- 矩阵 : $W^{(i)} \in R^{n_i \times n_{i-1}}$, 每个元素非负
- (非线性)激活函数 : $h^{(i)}$, 为单增凸函数

那么 $f(x; \theta)$ 的每一维输出均为凸函数



特殊的神经网络

➤ 伽利略原理

在没有外力作用的情况下，所有惯性参考系对于描述力学现象都是等价的。

神经网络建模： $\mathcal{M} = \mathcal{M}(u(x), \theta)$ 在坐标变换下保持不变。

坐标变换：

$$[\tilde{e}_1 \ \tilde{e}_2 \ \tilde{e}_3] = [e_1 \ e_2 \ e_3]Q$$

向量： $u = Q\tilde{u}$

矩阵(比如 ∇u):

$$\nabla_x u(x) = Q\nabla_{\tilde{x}}\tilde{u}Q^T$$



特殊的神经网络

➤ 伽利略原理

神经网络建模： $\mathcal{M} = \mathcal{M}(u(x), \theta)$ 在坐标变换下保持不变。

输出标量： $\tilde{\mathcal{M}} = \mathcal{M}$

输出向量： $\mathcal{M} = Q\tilde{\mathcal{M}}$

输出矩阵（二阶张量）： $\mathcal{M} = Q\tilde{\mathcal{M}}Q^T$



物理信息神经网络 (PINN)

➤ 优点

- 能同时求解正、反问题

➤ 缺点

- 方法对超参数(例如 λ_{Ii} , λ_{Bi})比较敏感性
- 尚未用于求解大规模的问题
- 传统方法结合神经网络建模会有数值不稳定



参考文献

➤ 参考文献

Raissi, Maziar, Paris Perdikaris, and George E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations." *Journal of Computational physics* 378 (2019): 686-707.

Yu, Bing. "The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems." *Communications in Mathematics and Statistics* 6, no. 1 (2018): 1-12.

Han, Jiequn, Arnulf Jentzen, and Weinan E. "Solving high-dimensional partial differential equations using deep learning." *Proceedings of the National Academy of Sciences* 115, no. 34 (2018): 8505-8510.

Amos, Brandon, Lei Xu, and J. Zico Kolter. "Input convex neural networks." In *International conference on machine learning*, pp. 146-155. PMLR, 2017.

Ling, Julia, Andrew Kurzawski, and Jeremy Templeton. "Reynolds averaged turbulence modelling using deep neural networks with embedded invariance." *Journal of Fluid Mechanics* 807 (2016): 155-166.