Lab 401: Approximation of direction fields by ODE-Net near the special points for simple ODEs

Фанис Адикович Хафизов

Московский физико-технический институт

Курс: Математические методы прогнозирования
Группа Б05-105

2024

Цель работы

Задача: Построить направляющее поле и его аппроксимацию с помощью ODE-Net в окрестности особых точек для простых ОДУ (uniform, saddle, center, spiral)

Литература

- Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud. Neural Ordinary Differential Equations. 2019
- Alexander Norcliffe, Cristian Bodnar, Ben Day, Jacob Moss, Pietro Liò. Neural ODE Processes. 2021
- PyTorch Implementation of Differentiable ODE Solvers by Ricky Chen (https://github.com/rtqichen/torchdiffeq)

О данных

В качестве данных взяты зашумленные траектории решений ОДУ

$$\dot{x} = Ax, \quad x(0) = x_0.$$

Траектории для обучения и валидации взяты с началом в разных точках.

Постановка задачи

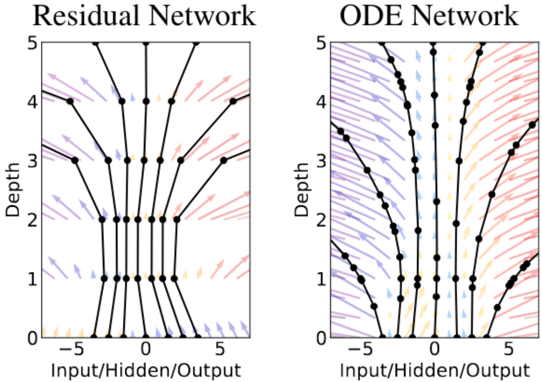
По имеющимся зашумленным участкам траекторий восстановить векторное поле направлений вблизи особых точек. Для этого будем делать предсказание ODE-Net траектории решения x^i_{pred} в точках t^i, где известно значение x^i_{true}, после чего минимизировать MAE:

$$MAE(x_{pred}, x_{true}) = \frac{1}{l} \sum_{i=1}^{l} \|x_{pred} - x_{true}\|_1.$$

Идея ODE-Net

Переход от дискретного количества преобразований внутреннего состояния h к непрерывному:

$$h_{t+1} = h_t + f(h_t, \theta_t) \rightarrow \frac{dh(t)}{dt} = f(h(t), t, \theta).$$



Оптимизация ODE-Net

Минимизируется функция

$$L(x(t_1)) = L\left(x(t_0) + \int\limits_{t_0}^{t_1} f(x(t), t, \theta)\right) \to \min_{\theta}.$$

В нашей задаче

$$L(x(t_k)) = \|x(t_k) - x_{true}(t_k)\|_1,$$

$f(x(t), t, \theta) = f(x(t), \theta)$ – двухслойная нейронная сеть, зависимости от времени нет.

Постановка эксперимента

Для каждого из 4 рассматриваемых видов особых точек:

- Выписываем матрицу $A \in \mathbb{R}^{2\times 2}$
- Выбираем по 2 стартовых точки для train $x^1(0), x^2(0)$ и по 1 для test $x_{test}(0)$.
- Находим численные решения дифференциального уравнения $x^1(t), x^2(t), x_{test}(t)$ в точках t_i.
- Берем подпоследовательность в одной из траекторий, зашумляем ее, получаем предсказание $x_{pred}(t)$
- Считаем MAE, обновляем параметры сети
- Рисуем фазовый портрет для всех траекторий и поле направлений

Эксперимент

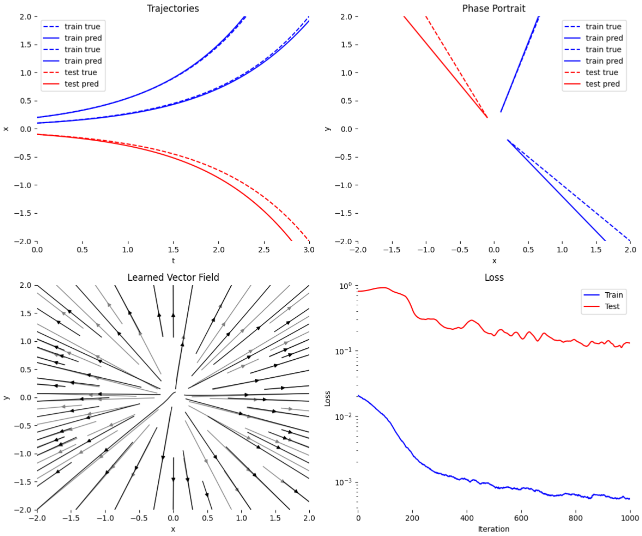


Рис.: Uniform. Без шума

Эксперимент

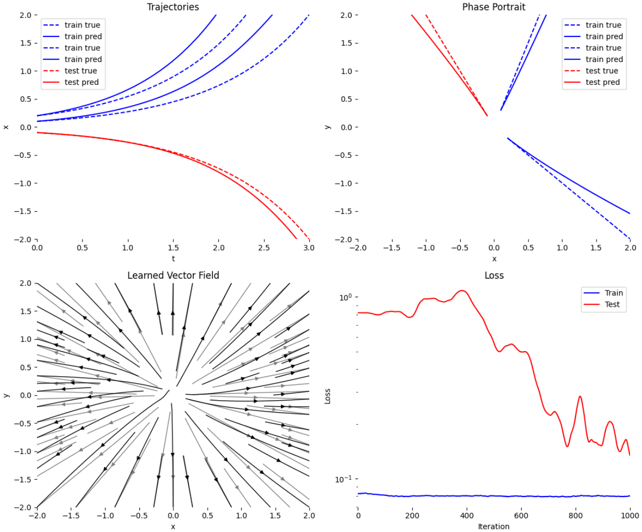


Рис.: Uniform. Шум из $N(0, 0.1)$

Эксперимент

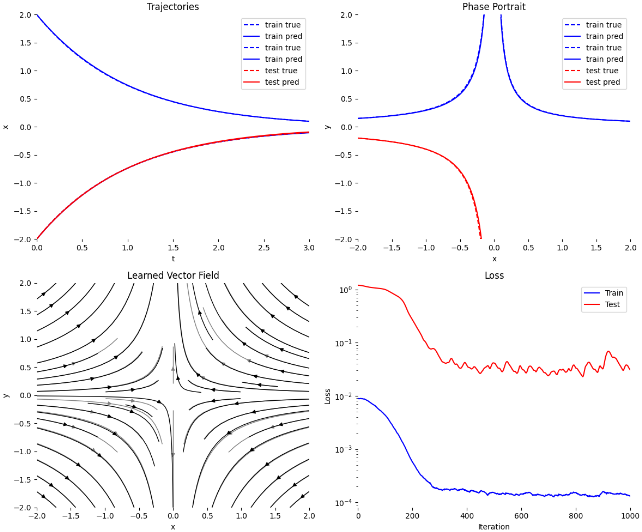


Рис.: Saddle. Без шума

Эксперимент

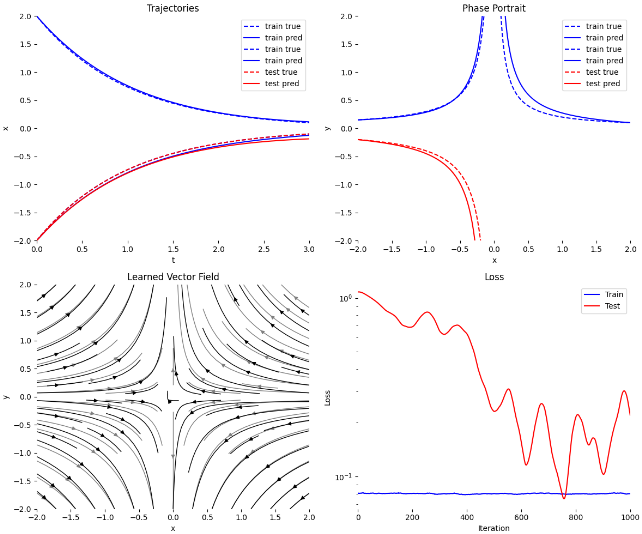


Рис.: Saddle. Шум из $N(0, 0.1)$

Эксперимент

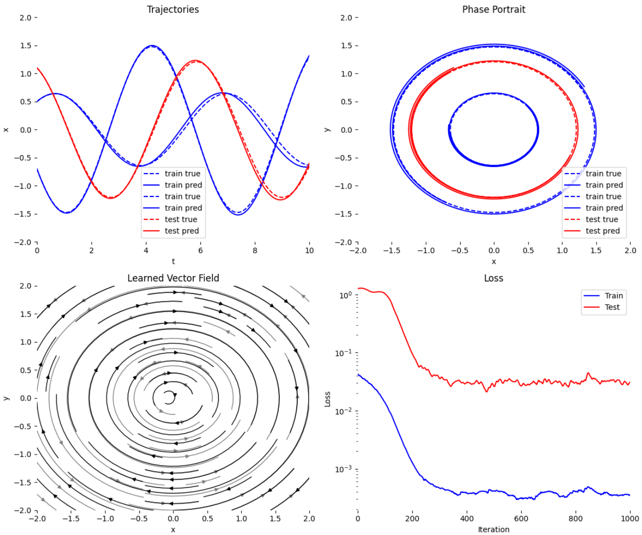


Рис.: Center. Без шума

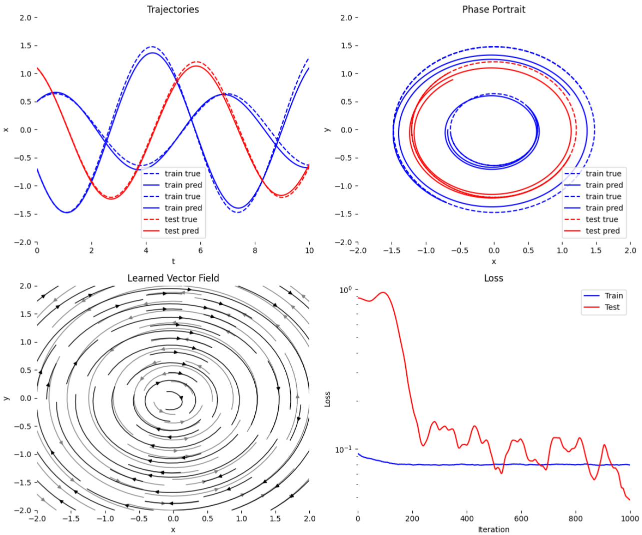


Рис.: Center. Шум из $N(0, 0.1)$

Эксперимент

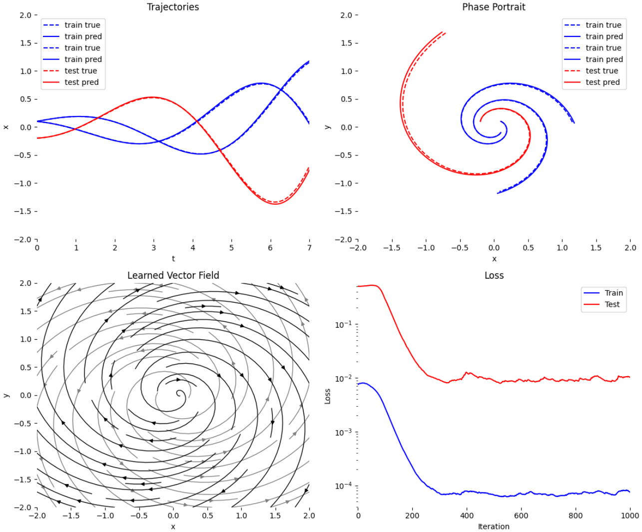


Рис.: Spiral. Без шума

Эксперимент

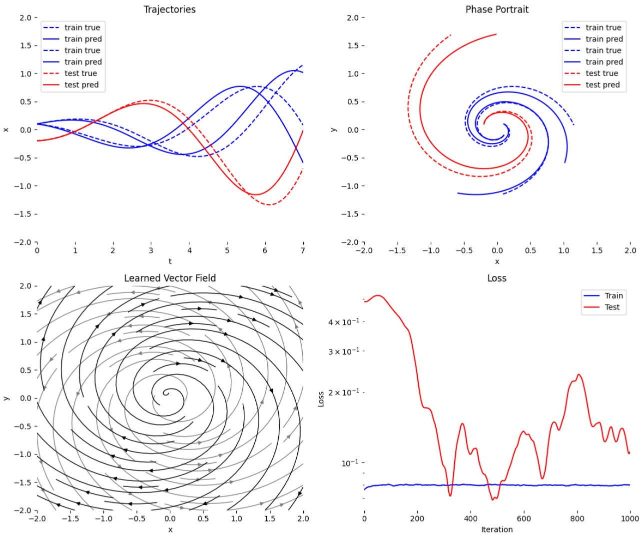


Рис.: Spiral. Шум из $N(0, 0.1)$

Выводы

- Обученная модель достаточно точно восстанавливает векторное поле направлений, даже при наличии шума в данных.
- Выбор двух различных траекторий для обучения позволяет более точно восстанавливать сложные траектории, такие как saddle и uniform.
- На самом деле выбранная модель избыточна для нашей задачи, ведь обученное преобразование приближает $A : x \to Ax$.

Будущая работа

- Научиться решать нелинейные системы уравнений, то есть, вида
$$\dot{x} = g(x),$$
где $g(x)$ не обязательно линейная функция.