

Нейронные дифференциальные уравнения

Голов В. А.

Мотивация

- ▶ Значительная доля процессов описывается дифференциальными уравнениями
- ▶ Данные в таких процессах непрерывны
- ▶ На данный момент такие системы обычно описываются рекуррентными сетями
- ▶ Рекуррентные сети требуют дискретизации данных
- ▶ Попробуем аппроксимировать нейронной сетью не саму целевую функцию, а ее производную, чтобы добиться гладкости целевой функции.

Постановка задачи

Пусть есть процесс, который подчиняется некоторому неизвестному ОДУ и пусть есть несколько (зашумленных) наблюдений вдоль траектории процесса:

$$\begin{cases} \frac{dz(t)}{dt} = f(z(t), t) \\ \{(z_0, t_0), (z_1, t_1), \dots, (z_M, t_M)\} - \text{наблюдения} \end{cases} \quad (1)$$

Задача в (??) является задачей Коши для ОДУ. Вместо аналитического решения будем искать аппроксимирующую функцию $\hat{f}(z, t, \theta) \approx f(z, t)$.

Решение задачи

Сначала рассмотрим более простую задачу: есть только 2 наблюдения, которые задают один отрезок наших наблюдений $(z_0, t_0), (z_1, t_1)$, где 1-ый член – конец отрезка. Следовательно решение задачи сводится к решению следующей задачи Коши:

$$\begin{cases} \frac{dz(t)}{dt} = \hat{f}(z(t), t, \theta) \\ z(t_0) = z_0 \end{cases} \quad (2)$$

решив данную задачу получим \hat{z}_1 , которое сравним с истинным z_1 , после чего исправим веса θ нейронной сети $\hat{f}(z(t), t, \theta)$.

Решение задачи

Для корректировки значений весов θ необходимо рассчитать градиент от функции потерь

$$\mathcal{L}(z_1, \hat{z}_1) = \mathcal{L} \left(z(t_1), z(t_0) + \int_{t_0}^{t_M} f(z(t), t, \theta) dt \right) \quad (3)$$

Для градиента функции потерь нужно рассчитать градиенты по всем его параметрам: $t_0, t_1, \theta, z(t_0)$. Производную \mathcal{L} от состояние z в момент времени t выглядит следующим образом:

$$\frac{\partial \mathcal{L}}{\partial z(t)} = a(t), \quad (4)$$

где $a(t)$ – сопряженное (adjoint) состояние.

Решение задачи

Сопряженное состояние изменяется в каждый момент времени так:

$$\frac{da(t)}{dt} = -a(t)^T \frac{\partial \hat{f}(z(t), t, \theta)}{\partial z} \quad (5)$$

данную формулу выводят авторы оригинальной статьи. Отсюда мы можем найти $a(t)$ еще раз воспользовавшись методом решения ОДУ:

$$\frac{\partial L}{\partial z(t_0)} = \int_{t_1}^{t_0} -a(t)^T \frac{\partial \hat{f}(z(t), t, \theta)}{\partial z} \quad (6)$$

Решение задачи

Чтобы рассчитать градиент по отношению к t и θ , можно просто считать их частью состояния. Такое состояние зовется аугментированным. Динамика такого состояния тривиально получается из оригинальной динамики:

$$\frac{d}{dt} \begin{bmatrix} z(t) \\ \theta \\ t \end{bmatrix} = f_{aug}([z(t), \theta, t]) := \begin{bmatrix} \hat{f}(z(t), t, \theta) \\ 0 \\ 1 \end{bmatrix} \quad (7)$$

Тогда сопряженное состояние $a_{aug}(t)$ к этому аугментированному состоянию:

$$a_{aug} := \begin{bmatrix} a \\ a_\theta \\ a_t \end{bmatrix} = \begin{bmatrix} \partial L / \partial z(t) \\ \partial L / \partial \theta \\ \partial L / \partial t \end{bmatrix} \quad (8)$$

Решение задачи

Градиент аугментированной динамики:

$$\frac{df_{aug}}{d[z, \theta, t]} = \begin{bmatrix} \frac{\partial f}{\partial z} & \frac{\partial f}{\partial \theta} & \frac{\partial f}{\partial t} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} (t) \quad (9)$$

Дифференциальное уравнение сопряженного аугментированного состояния тогда выглядит так:

$$\frac{da_{aug}}{dt} = - \left[a^T \frac{\partial f}{\partial z}, a^T \frac{\partial f}{\partial \theta}, a^T \frac{\partial f}{\partial t} \right] (t) \quad (10)$$

Решение задачи

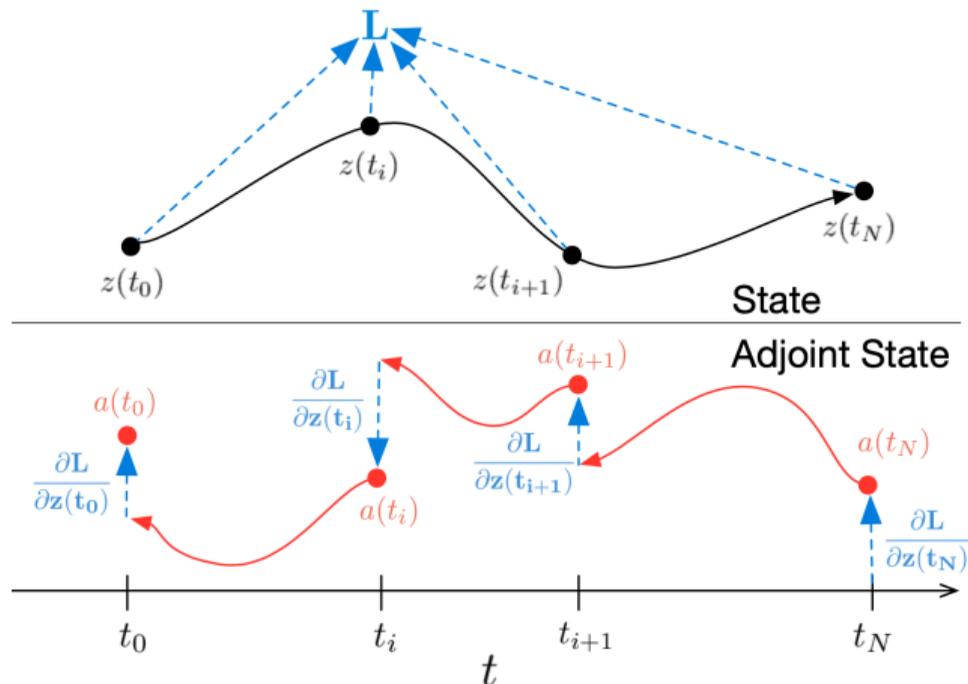
Решение этого ОДУ назад во времени

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial z(t_0)} = - \int_{t_1}^{t_0} a(t)^T \frac{\partial f(z(t), \theta, t)}{\partial z} dt \\ \frac{\partial L}{\partial \theta} = a_\theta(t_0) = - \int_{t_1}^{t_0} a(t)^T \frac{\partial f(z(t), \theta, t)}{\partial \theta} dt \\ \frac{\partial L}{\partial t_0} = a_t(t_0) = a_t(t_1) - \int_{t_1}^{t_0} a(t)^T \frac{\partial f(z(t), \theta, t)}{\partial t} dt \\ \frac{\partial L}{\partial t_1} = a(t_1)^T f(z(t_1), t_1, \theta) \end{array} \right. \quad (11)$$

дает градиенты по всем входным параметрам в метод решения ОДУ. Все градиенты (??) могут быть рассчитаны вместе за один вызов метода решения ОДУ с динамикой сопряженного аугментированного состояния.

Решение задачи

Для решения исходной задачи (1) прямой проход выполняется последовательно как и в обычном Рунге-Кутта от t_0 до t_1 , от t_1 до t_2 и т.д. от t_{M-1} до t_M . В обратном прохождении необходимо после прохода от t_M до t_{M-1} учесть начальное условие для прохождения от t_{M-1} до t_{M-2} .



Спасибо за внимание!

Общая постановка задачи

Пусть задано параметрическое семейство дифференциальных уравнений, разрешенных относительно старшей производной

$$\dot{\vec{y}} = f(t, \vec{y}, \vec{\theta}) \quad (1)$$

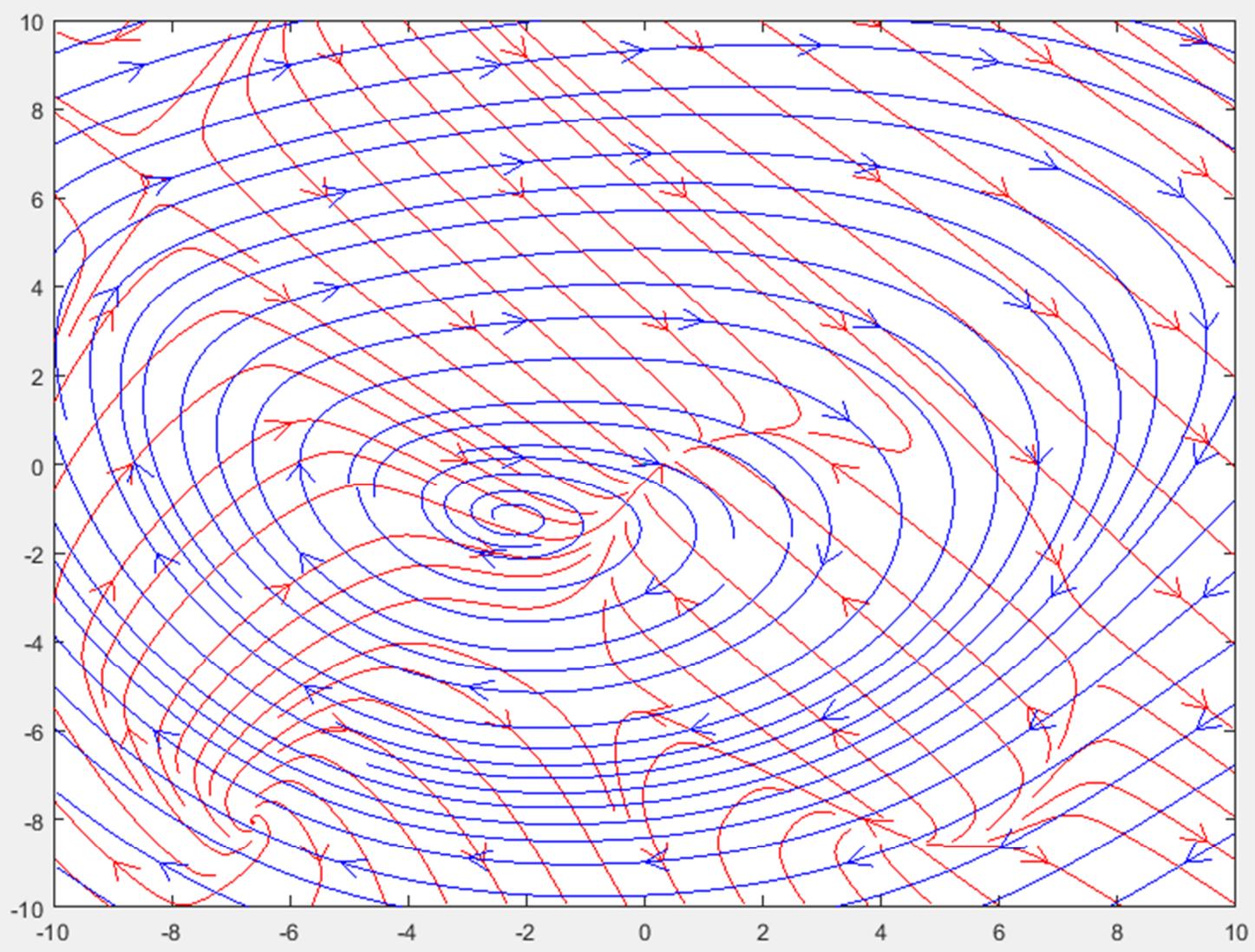
Допустим, что известны результаты наблюдений за частными решениями системы (1) в некоторые моменты времени:

$$\vec{y}_j(t_{kj}) = \vec{y}_{kj}, \quad (2)$$

где j – номер частного решения, $k = 1 \dots n_j$.

Будем решать обратную задачу:

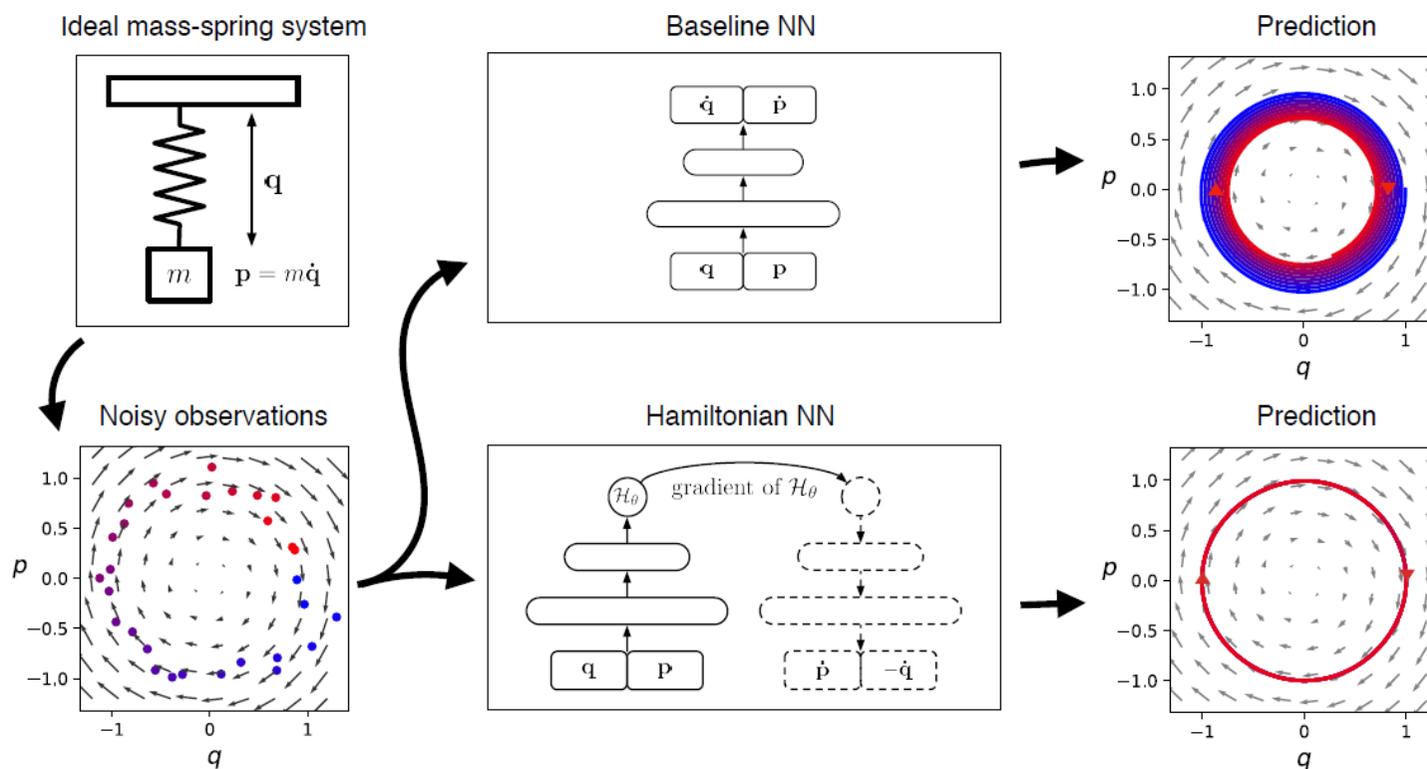
Найти параметры $\vec{\theta}$, такие чтобы уравнения (1) наилучшим образом соответствует наблюдениям (2).



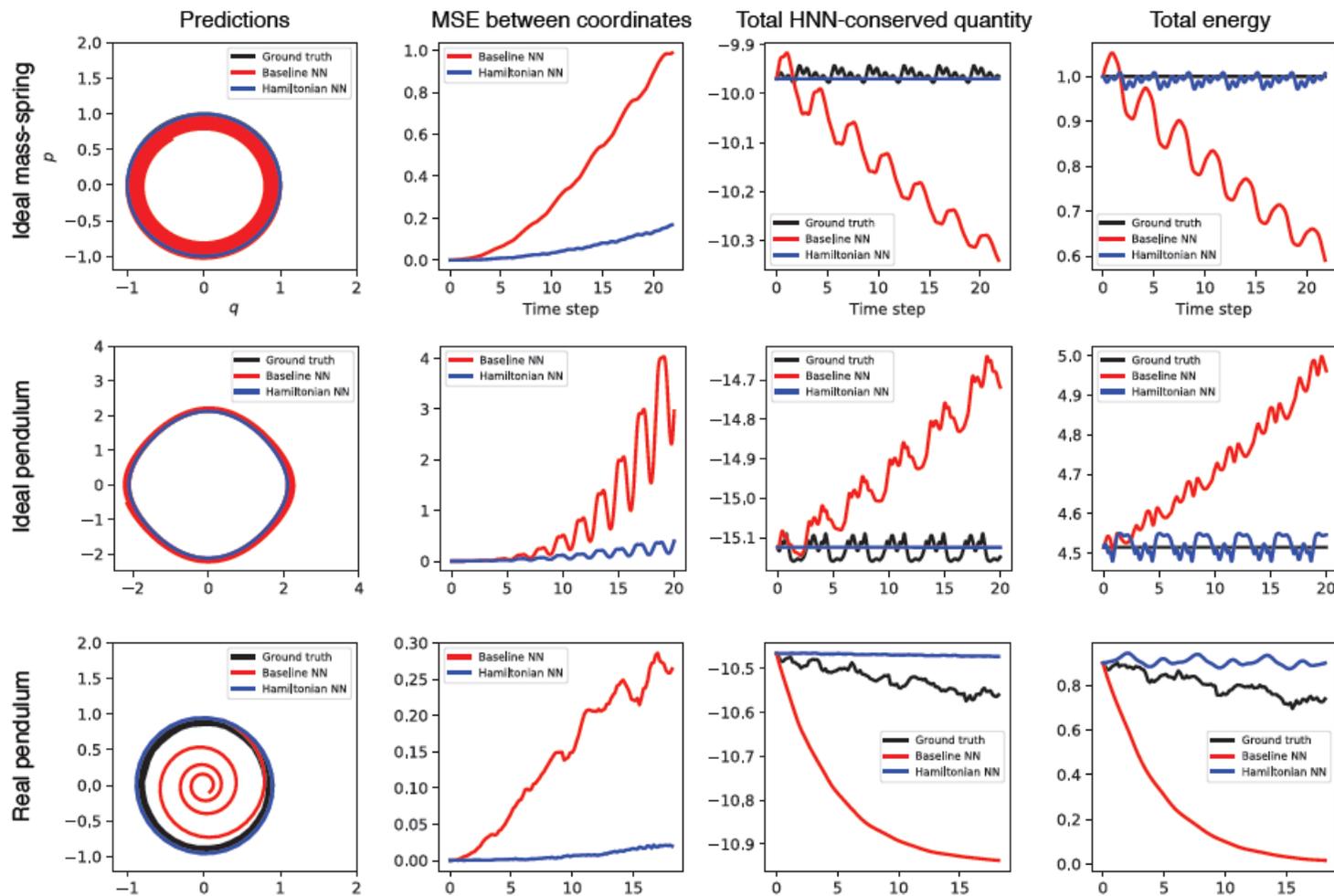
Hamiltonian Neural Networks (HNN)

[Greydanus S. et al. NIPS – 2019]

Neural ODE не знает о свойствах физ. системы. Для гамильтоновых систем будем искать не правую часть, а гамильтониан $H(p, q)$ в виде нейронной сети



Производные гамильтониана $\left(\frac{\partial H}{\partial \vec{p}} = \frac{d\vec{q}}{dt}, \frac{\partial H}{\partial \vec{q}} = -\frac{d\vec{p}}{dt} \right)$ мы можем вычислить точно

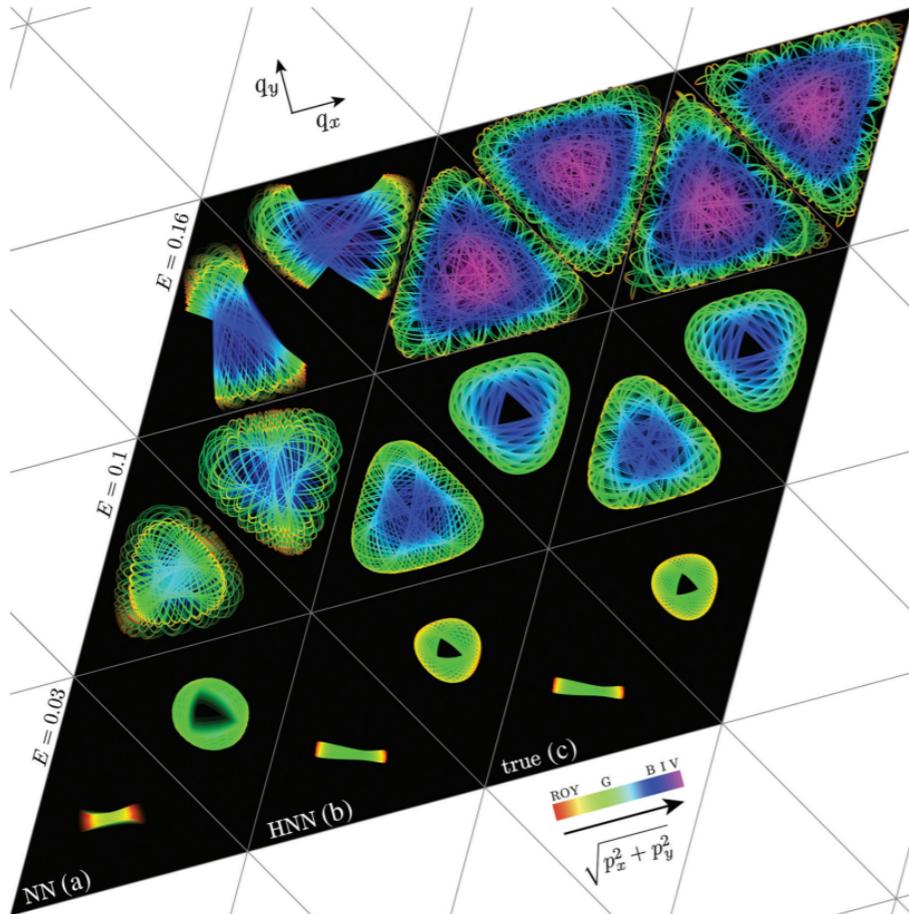


HNN не сохраняет энергию из-за погрешности интегрирования, но колебание суммарной энергии на порядки меньше

Task	Train loss		Test loss		Energy	
	Baseline	HNN	Baseline	HNN	Baseline	HNN
1: Ideal mass-spring	37 ± 2	37 ± 2	37 ± 2	36 ± 2	170 ± 20	$.38 \pm .1$
2: Ideal pendulum	33 ± 2	33 ± 2	35 ± 2	36 ± 2	42 ± 10	25 ± 5
3: Real pendulum	$2.7 \pm .2$	$9.2 \pm .5$	$2.2 \pm .3$	$6.0 \pm .6$	390 ± 7	14 ± 5
4: Two body ($\times 10^6$)	33 ± 1	$3.0 \pm .1$	$30 \pm .1$	$2.8 \pm .1$	$6.3e4 \pm 3e4$	39 ± 5
5: Pixel pendulum	$18 \pm .2$	$19 \pm .2$	$17 \pm .3$	$18 \pm .3$	9.3 ± 1	$.15 \pm .01$

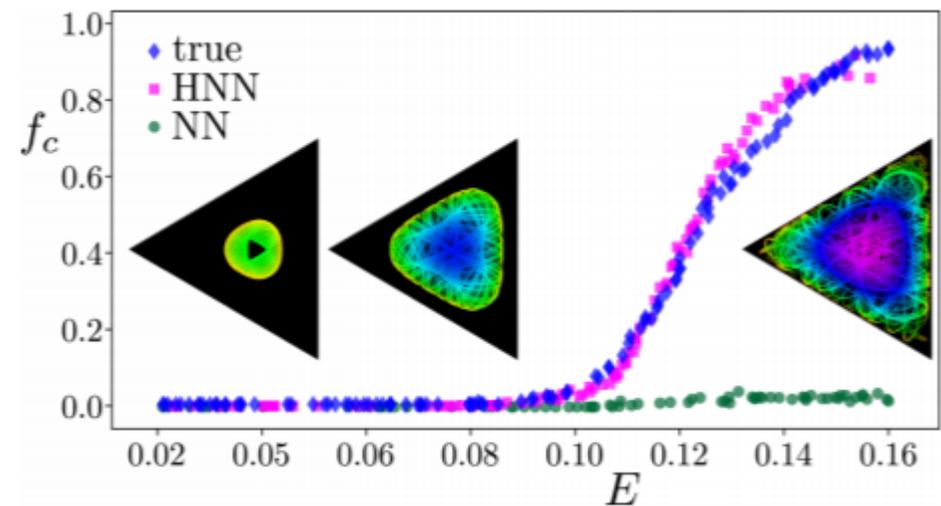
Physics-enhanced neural networks learn order and chaos

[Choudhary et al., PHYSICAL REV E 101, 062207]



Исследуется система, все большая доля решений которой становится хаотичными (заполняют всю $H=const$) при росте энергии от $1/12$ до $1/6$ (почти все хаотичны).

HNN позволяют параметризовать такую динамику



Physics-informed neural networks

Raissi et al., Computational Physics, v.378, p.686-707 2018

Рассмотри задачу параметризации уравнений в частных производных.

Пусть решение задачи $\vec{u}(t, \vec{x}, \vec{\theta})$ – нейронная сеть. Будем искать $\vec{\theta}$ и $\vec{\lambda}$, такие что

$$\dot{u} + D[u, \vec{\lambda}] = 0, \quad u(0, x_{0i}, \vec{\theta}) = u_{0i}, u(1, x_{1j}, \vec{\theta}) = u_{1j}, \quad (1)$$

где $D[u, \vec{\lambda}]$ – дифференциальный оператор, $x_{0i}, u_{0i}, x_{1j}, u_{1j}$ – известны.

Подход 1. Время непрерывно

Минимизируем невязку (1), вычисляя все производные точно:

$$\left\| \dot{u}(t, \vec{x}, \vec{\theta}) + D[u(t, \vec{x}, \vec{\theta}), \vec{\lambda}] \right\|_{L_2} + \sum_i (u(0, x_{0i}, \theta) - u_{0i})^2 + \sum_j (u(1, x_{1j}, \theta) - u_{1j})^2 \rightarrow \min_{\lambda, \theta}$$

Подход 2. Время дискретно

Запишем общий (неявный) q -шаговый метод Рунге-Кутты ($\Delta t = 1$):

$$\begin{aligned} u(c_k, x) &= u(0, x) - \sum_{s=1}^q a_{ks} D[u(c_s, x), \vec{\lambda}], k = 1 \dots q \\ u(1, x) &= u(0, x) - \sum_{s=1}^q b_s D[u(c_s, x), \vec{\lambda}] \end{aligned} \quad (2)$$

Пусть $\vec{u}(\vec{x}, \vec{\theta})$ – нейронная сеть, возвращающая вектор длины q .

Будем минимизировать невязку (2) в известных точках x_{0i}, x_{1j} , то есть:

$$\sum_i \left[u_k(x_{0i}, \vec{\theta}) + \sum_{s=1}^q a_{ks} D[u_s(x_{0i}, \vec{\theta}), \vec{\lambda}] - u_{0i} \right]^2 + \sum_j \left[u_k(x_{1j}, \vec{\theta}) + \sum_{s=1}^q (a_{ks} - b_s) D[u_s(x_{1j}, \vec{\theta}), \vec{\lambda}] - u_{1j} \right]^2 \rightarrow \min_{\vec{\theta}, \vec{\lambda}}$$

При этом $D[u, \vec{\lambda}]$ вычисляется точно, а метод (2) имеет порядок $2q$, и при достаточно большом q (>50) даёт машинную точность.

Экономичное (не за q , а за 2 прохода по сети) вычисление производных в

$$D[u, \vec{\lambda}]: \quad \nabla_x \Big|_{x=x_i} u_s(x, \vec{\theta}) = \nabla_w \Big|_{w_{is}=1} \nabla_x \Big|_{x=x_i} \sum_{i,s=1}^{N,q} w_{is} u_s(x, \vec{\theta})$$

Пример: уравнение Бюргерса $D[u, \vec{\lambda}] = \lambda_1 uu' - \lambda_2 u''$

Погрешность восстановления $\lambda_1 = 1, \lambda_2 = 0.01 / \pi$ в зависимости от шума (в u_{0i}, u_{1j}) и кол-ва данных

		% error in λ_1				% error in λ_2			
		0%	1%	5%	10%	0%	1%	5%	10%
N_u	Noise								
	500	0.131	0.518	0.118	1.319	13.885	0.483	1.708	4.058
	1000	0.186	0.533	0.157	1.869	3.719	8.262	3.481	14.544
	1500	0.432	0.033	0.706	0.725	3.093	1.423	0.502	3.156
	2000	0.096	0.039	0.190	0.101	0.469	0.008	6.216	6.391

В зависимости от Δt и шума

		% error in λ_1				% error in λ_2			
		0%	1%	5%	10%	0%	1%	5%	10%
Δt	Noise								
	0.2	0.002	0.435	6.073	3.273	0.151	4.982	59.314	83.969
	0.4	0.001	0.119	1.679	2.985	0.088	2.816	8.396	8.377
	0.6	0.002	0.064	2.096	1.383	0.090	0.068	3.493	24.321
	0.8	0.010	0.221	0.097	1.233	1.918	3.215	13.479	1.621

Невязка уравнения в зависимости от кол-ва шагов Рунге-Кутты и Δt

$q \backslash \Delta t$	0.2	0.4	0.6	0.8
1	3.5e-02	1.1e-01	2.3e-01	3.8e-01
2	5.4e-03	5.1e-02	9.3e-02	2.2e-01
4	1.2e-03	1.5e-02	3.6e-02	5.4e-02
8	6.7e-04	1.8e-03	8.7e-03	5.8e-02
16	5.1e-04	7.6e-02	8.4e-04	1.1e-03
32	7.4e-04	5.2e-04	4.2e-04	7.0e-04
64	4.5e-04	4.8e-04	1.2e-03	7.8e-04
100	5.1e-04	5.7e-04	1.8e-02	1.2e-03
500	4.1e-04	3.8e-04	4.2e-04	8.2e-04

Отклонение решения от точного в зависимости от кол-ва данных

N	250	200	150	100	50	10
Error	4.02e-4	2.93e-3	9.39e-3	5.54e-2	1.77e-2	7.58e-1

Спасибо за внимание

Работа была поддержана грантом № 20-04-021 в рамках
Программы «Научный фонд Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ)» в 2020–2021 гг.