КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ АЛЬ-ФАРАБИ · КАФЕДРА КОМПЬЮТЕРНЫХ НАУК

Научное машинное обучение

Лекция 9. Нейронные обыкновенные дифференциальные уравнения (Neural ODE)

Динамика, заданная нейросетью, и обучение через решатель ОДУ

Кенжебек Ержан Ғалымжанұлы

PhD, старший преподаватель кафедры компьютерных наук

Цель и план лекции

Цель

Изучить нейронные ОДУ: непрерывную модель динамики с правой частью в виде нейросети, способы её обучения и применение к научным динамическим системам.

План

1. От ResNet к ОДУ
2. Определение и решатели
3. Обучение: обратное распространение и сопряжённый метод
4. Пример: система Лотки–Вольтерры
5. Гибридные модели с физикой
6. Расширения и ограничения

От ResNet к дифференциальному уравнению

$$h_{t+1} = h_t + f(h_t, \theta_t)$$

Остаточный блок — это шаг явного метода Эйлера с шагом 1. При уменьшении шага и росте числа слоёв получаем непрерывную модель (Chen и др., 2018):

$$dh/dt = f_\theta(h(t), t)$$

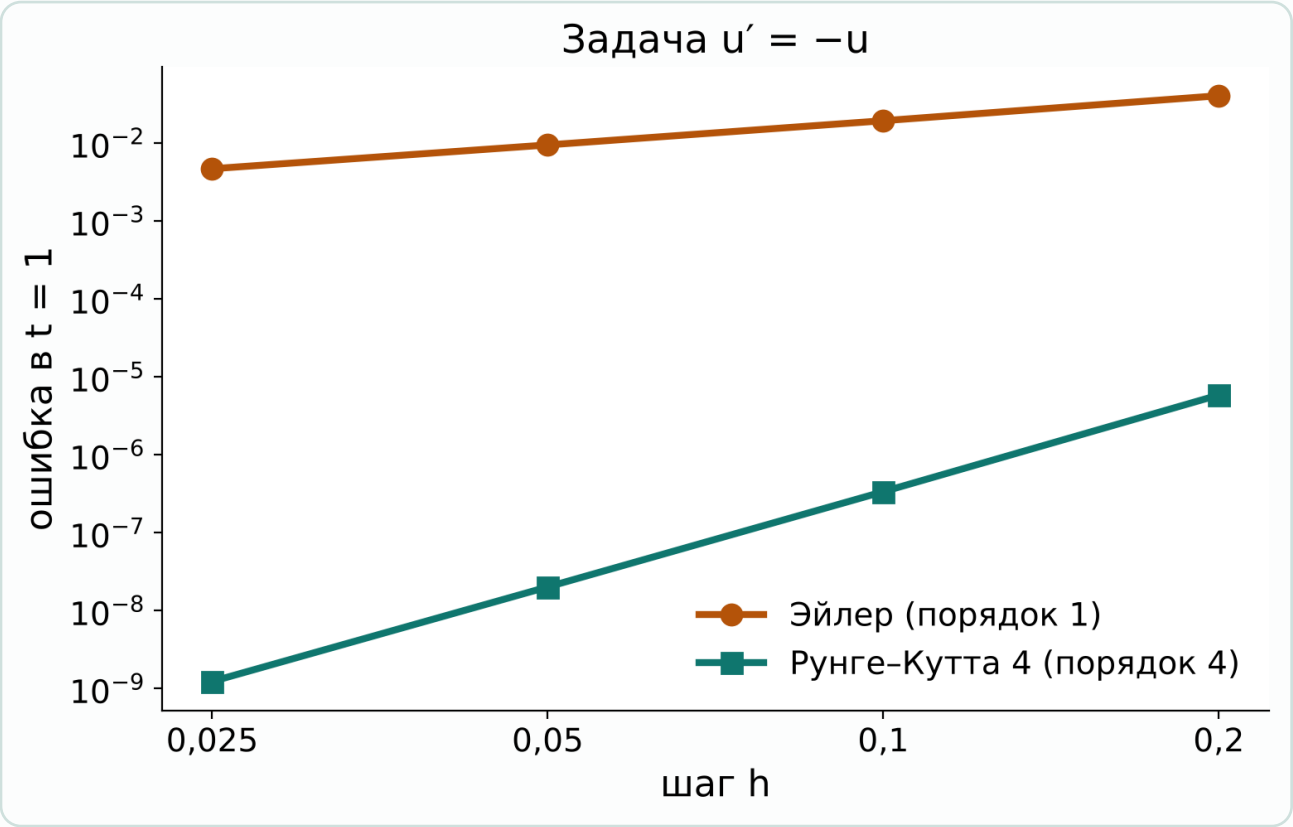
Определение

Состояние в момент t_1 получается интегрированием обученной правой части от начального состояния:

$$z(t_1) = z(t_0) + \int f_\theta(z(t), t)\, dt = \mathrm{ODESolve}(z(t_0), f_\theta, t_0, t_1)$$

- f_θ — нейросеть; ОДУ решается любым численным методом.
- Модель задаёт траекторию в любые моменты времени, в том числе нерегулярные.
- Для научных систем z — это физическое состояние.

Решатели ОДУ



Как обучать

Потери $L(z(t_1))$ требуют градиента по θ через решатель. Два способа:

Обратное распространение через решатель

Просто и точно относительно дискретной схемы, но память растёт с числом шагов.

Сопряжённый метод

Дополнительная ОДУ назад по времени: $da/dt = -a^{\mathsf{T}} \cdot \partial f/\partial z$. Память не зависит от числа шагов, но градиент приближённый.

Сравнение способов

Свойство	Через решатель	Сопряжённый метод
Память	Растёт с числом шагов	Постоянная
Градиент	Точный для дискретной схемы	Приближённый, зависит от допусков
Скорость	Быстрее на коротких траекториях	Дополнительное решение назад по времени
Применение	Малые и средние задачи	Длинные траектории, большая память

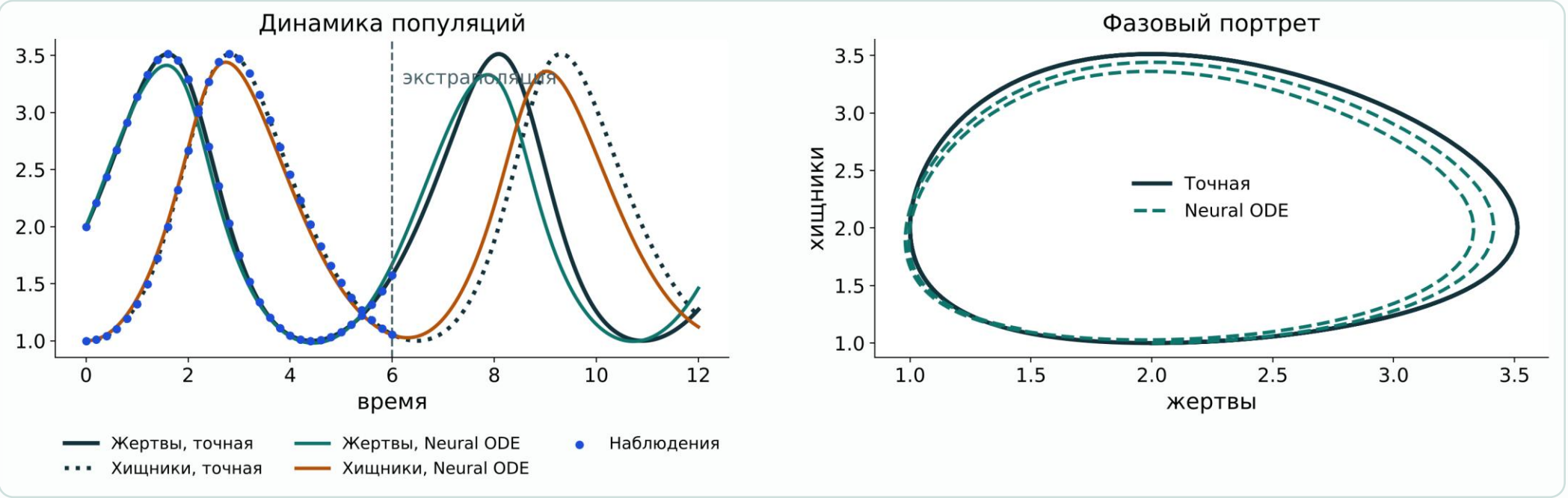
Пример: Лотка–Вольтерра

$$x' = x - 0{,}5 \cdot xy,\ y' = 0{,}5 \cdot xy - y,\ (x_0, y_0) = (2, 1)$$

Генерируем траекторию на [0, 6], берём 31 наблюдение. Правая часть — сеть 2–64–64–2 с tanh; интегрирование методом Рунге–Кутты 4 с 120 шагами; 800 итераций Adam.

Затем интегрируем обученную модель до t = 12 и сравниваем с точным решением.

Результат



В области обучения (t до 6) совпадение точное. За её пределами модель повторяет циклический характер, но амплитуда и фаза постепенно расходятся — типичная ошибка экстраполяции для модели без физики.

Гибрид: известная физика плюс сеть

Универсальные дифференциальные уравнения (Rackauckas и др., 2020) включают известные члены, а неизвестные заменяют сетью:

$$z' = f_известное(z; p) + NN_\theta(z)$$

Это улучшает экстраполяцию, уменьшает объём данных и сохраняет интерпретируемость; недостающую часть можно позже заменить символьным выражением (SINDy, лекция 13).

Расширения

Модель	Идея
Latent ODE	Динамика в скрытом пространстве для нерегулярных рядов (Rubanova и др., 2019)
Neural CDE	Управляемые уравнения для потоков входных данных (Kidger и др., 2020)
Hamiltonian NN	Динамика через гамильтониан сохраняет энергию (Greydanus и др., 2019)
Neural SDE	Стохастическая динамика

Ограничения

Стоимость

Каждый шаг обучения вызывает решатель много раз (NFE).

Жёсткость

Для жёстких систем нужны неявные решатели, обучение усложняется.

Экстраполяция

Без физических ограничений прогноз вне обучающего интервала дрейфует.

Neural ODE на PyTorch

```
from torchdiffeq import odeint

class Field(nn.Module):
def __init__(self):
super().__init__()
self.net = nn.Sequential(nn.Linear(2, 64), nn.Tanh(), nn.Linear(64, 2))
def forward(self, t, z):
return self.net(z)

pred = odeint(Field(), z0, t_grid) # (len(t), dim)
```

Выводы

- Нейронное ОДУ задаёт скорость изменения состояния нейросетью, а траектория получается решателем ОДУ.
- Остаточные сети — частный случай явного метода Эйлера.
- Градиенты вычисляют обратным распространением через решатель или сопряжённым методом.
- На примере Лотки–Вольтерры модель точна в области данных, но дрейфует при экстраполяции.
- Гибридные модели с известной физикой повышают точность и интерпретируемость.

Контрольные вопросы

1. Как связаны остаточная сеть и метод Эйлера?
2. Что такое Neural ODE и как получают её выход?
3. Чем сопряжённый метод отличается от обратного распространения через решатель?
4. Почему порядок решателя важен?
5. Что такое универсальное дифференциальное уравнение?
6. Какие ограничения у Neural ODE?

Литература

1. Chen R.T.Q., Rubanova Y., Bettencourt J., Duvenaud D. Neural ordinary differential equations // NeurIPS. 2018.
2. Rubanova Y., Chen R.T.Q., Duvenaud D. Latent ordinary differential equations for irregularly-sampled time series // NeurIPS. 2019.
3. Rackauckas C. et al. Universal differential equations for scientific machine learning. arXiv:2001.04385, 2020.
4. Greydanus S., Dzamba M., Yosinski J. Hamiltonian neural networks // NeurIPS. 2019.
5. Kidger P. et al. Neural controlled differential equations for irregular time series // NeurIPS. 2020.
6. Hairer E., Nørsett S.P., Wanner G. Solving Ordinary Differential Equations I. Springer, 1993.