КАЗАХСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ ИМЕНИ АЛЬ-ФАРАБИ · КАФЕДРА КОМПЬЮТЕРНЫХ НАУК

Научное машинное обучение

Лекция 10. Нейронные операторы I: DeepONet и операторное обучение

Обучение отображений между функциями: от одной задачи к семейству задач

Кенжебек Ержан Ғалымжанұлы

PhD, старший преподаватель кафедры компьютерных наук

Цель и план лекции

Цель

Ввести понятие операторного обучения и изучить архитектуру DeepONet, её теоретическую основу, обучающие данные и области применения.

План

1. От функций к операторам
2. Сравнение с PINN
3. Теорема универсальной аппроксимации оператора
4. Архитектура DeepONet
5. Данные и обучение
6. Пример и варианты метода

От функций к операторам

Оператор G переводит одну функцию в другую. Научные задачи — естественные примеры:

Правая часть → решение

G: f ↦ u для уравнения Пуассона $-\Delta u = f$.

Начальное условие → решение

G: $u_0 \mapsto u(\cdot, t)$ для эволюционного уравнения.

Коэффициент → решение

G: k ↦ u для уравнения с переменным коэффициентом.

Операторное обучение и PINN

Критерий	PINN	Нейронный оператор
Что учится	Решение одной задачи	Отображение для семейства задач
Новый вход	Новое обучение	Мгновенный вывод
Данные	Не обязательны	Нужны пары вход – выход (или физика)
Стоимость	Обучение на каждую задачу	Дорогое обучение, дешёвое применение

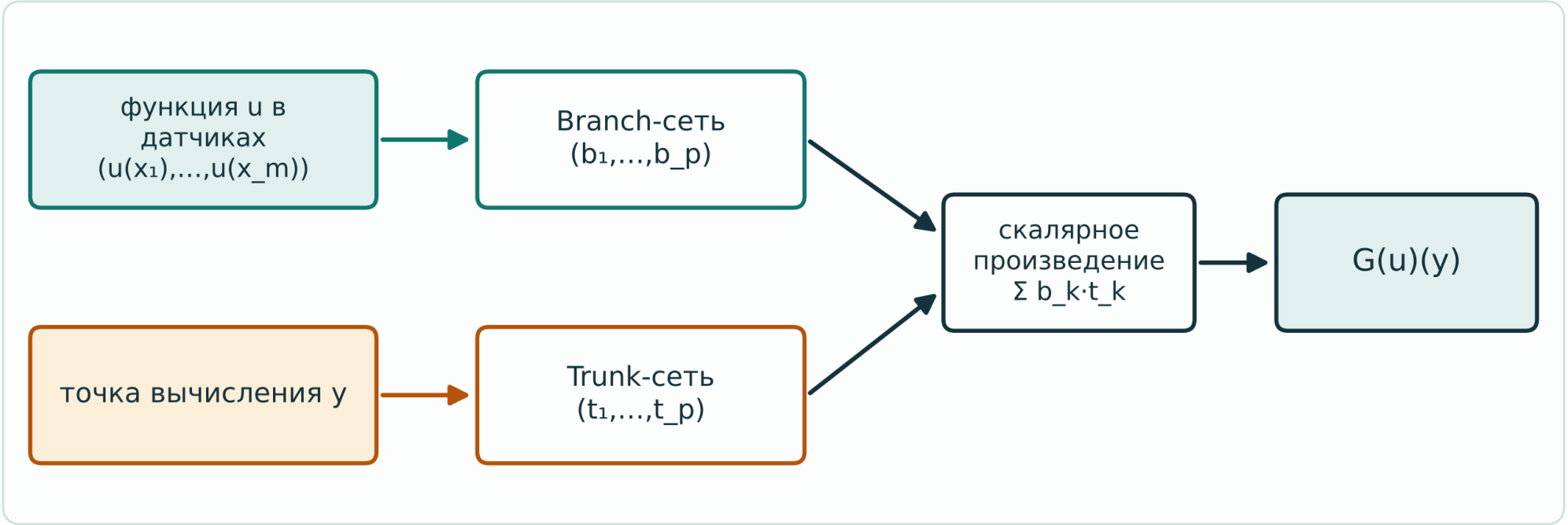
Теоретическая основа

Чэнь и Чэнь (1995) показали, что сети с одним скрытым слоем равномерно приближают нелинейные операторы, действующие между функциональными пространствами.

$$G(u)(y) \approx \Sigma_{k=1}^{p} c_k(u(x_1), \ldots, u(x_m)) \cdot t_k(y)$$

Эта форма стала основой DeepONet (Lu и др., 2021): коэффициенты c_k зависят от входной функции, базисные функции t_k — от точки y.

Архитектура DeepONet



Входная функция u задаётся значениями в m датчиках и подаётся в Branch-сеть; точка y — в Trunk-сеть. Выход — скалярное произведение их векторов длины p.

Формула DeepONet

$$G(u)(y) \approx \Sigma_{k=1}^{p} b_k(u(x_1), \ldots, u(x_m)) \cdot t_k(y) + b_0$$

- Branch-сеть b кодирует входную функцию в p чисел.
- Trunk-сеть t порождает p обучаемых базисных функций в точке y.
- Датчики $x_1, \ldots, x_m$ фиксированы для всех входов.
- Выход можно вычислить в любой точке y, а не только на сетке.

Данные и обучение

- Обучающие тройки: (дискретизованная u_i, точка y_j, значение G(u_i)(y_j)).
- Входные функции сэмплируют из случайных полей или рядов; выходы получают классическим решателем.
- Функция потерь — среднеквадратичная ошибка по функциям и точкам.
- Стоимость данных определяется числом запусков классического решателя.

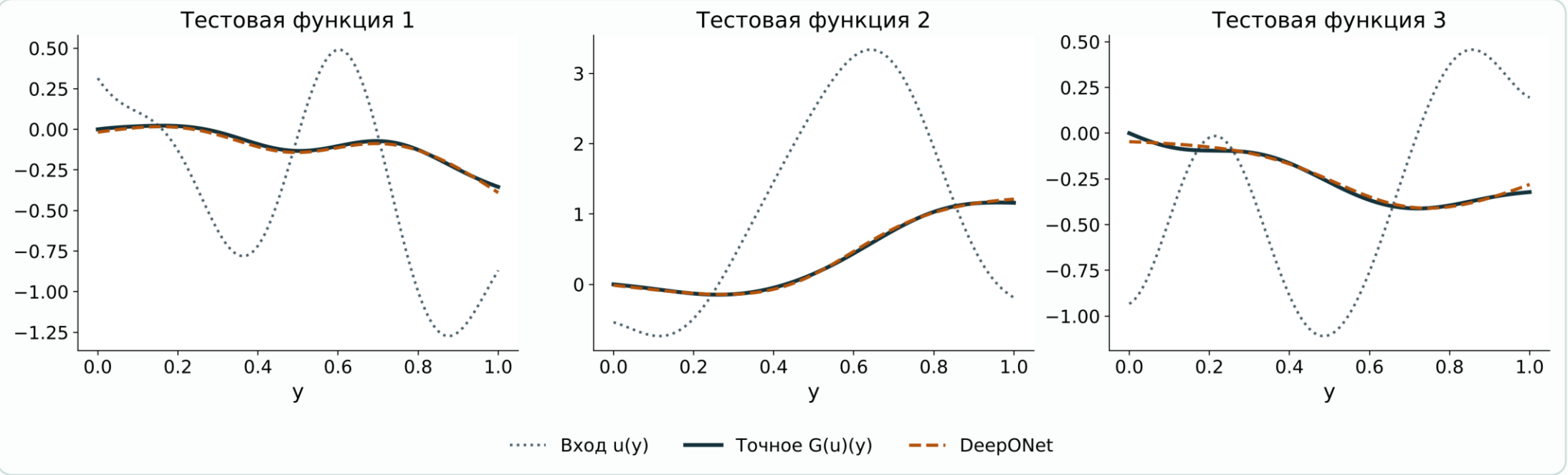
Пример: оператор первообразной

$$G(u)(y) = \int_0^y u(s)\,ds$$

Входные функции — случайные суммы из 5 синусов и косинусов. Датчиков m = 60; Branch и Trunk — сети по 2 слоя на 64 нейрона, p = 40; 4000 итераций Adam по 128 функций; 40 случайных точек y на шаг.

Проверка — на новых случайных функциях, не виденных при обучении.

Результат



На новых функциях DeepONet практически совпадает с точным интегралом; среднеквадратичная ошибка по трём функциям около 0,015, при амплитудах решения порядка 0,3–1,2.

Варианты

Вариант	Идея
Physics-informed DeepONet	В потери добавляют невязку уравнения, данных меньше (Wang, Wang, Perdikaris, 2021)
POD-DeepONet	Базис trunk-сети строят методом главных компонент
MIONet	Несколько входных функций (Jin и др., 2022)
Fourier Neural Operator	Другая архитектура — лекция 11

Ограничения

Фиксированные датчики

Входные функции должны задаваться в одних и тех же точках.

Распределение входов

Качество падает на функциях, непохожих на обучающие.

Цена данных

Для сложных задач нужно много решений классическим методом.

Сравнение DeepONet и FNO на разных задачах: Lu и др., 2022.

Практические рекомендации

- Нормировать входные функции и выходы.
- Проверять на тестовых функциях из другого распределения.
- Контролировать число датчиков m и размер p.
- Для малых данных рассмотреть вариант с физикой.

Выводы

- Нейронный оператор учит отображение между функциями и после обучения даёт решение мгновенно.
- DeepONet строится из ветви (кодирует входную функцию) и ствола (кодирует точку) со скалярным произведением выходов.
- Теоретическая основа — теорема Чэня и Чэня об аппроксимации операторов.
- Для оператора первообразной модель точно предсказала решение на новых функциях.
- Ограничения: фиксированные датчики, смещение распределения, цена данных.

Контрольные вопросы

1. Что такое оператор в смысле операторного обучения?
2. Чем DeepONet отличается от PINN по назначению?
3. Как устроены ветвь и ствол DeepONet?
4. Как формируются обучающие данные?
5. Какие варианты DeepONet вы знаете?
6. Каковы ограничения метода?

Литература

1. Lu L., Jin P., Pang G., Zhang Z., Karniadakis G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators // Nature Machine Intelligence. 2021. Vol. 3. P. 218–229.
2. Chen T., Chen H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems // IEEE Trans. Neural Networks. 1995. Vol. 6. P. 911–917.
3. Wang S., Wang H., Perdikaris P. Learning the solution operator of parametric partial differential equations with physics-informed DeepONets // Science Advances. 2021. Vol. 7.
4. Kovachki N. et al. Neural operator: learning maps between function spaces with applications to PDEs // JMLR. 2023. Vol. 24.
5. Lu L. et al. A comprehensive and fair comparison of two neural operators // Comput. Methods Appl. Mech. Eng. 2022. Vol. 393.