Kolmogorov-Arnold Networks (KANs): Альтернатива многослойным перцептронам в Tabular DL

Данил Руденко, Булгаков Георгий

Московский физико-технический институт

18 мая 2025 г.

Предпосылки и формулы

- **Теорема Колмогорова-Арнольда[6]**: Любая многомерная функция может быть представлена как композиция одномерных функций и операций сложения.

$$f(\mathrm{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

- **Проблема MLPs**: Фиксированные активационные функции на узлах и линейные веса.
- **Решение**: Замена линейных весов на обучаемые одномерные функции, параметризованные сплайнами.

Идея KANs

- **Архитектура**:
  - Каждый слой — матрица одномерных функций.
  - Глубокие KANs позволяют лучше аппроксимировать сложные функции.

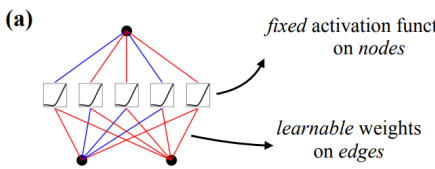
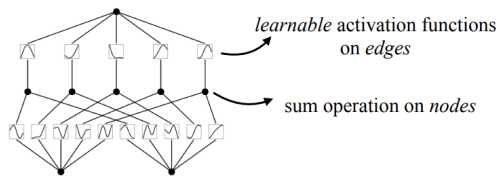
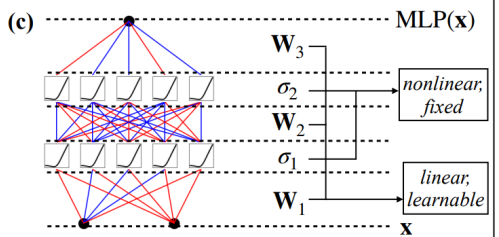
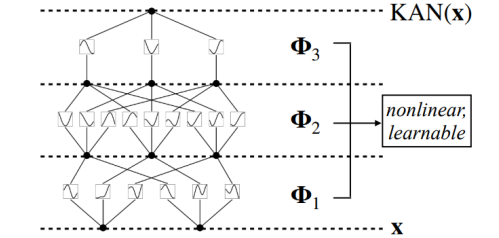
Model	Multi-Layer Perceptron (MLP)	Kolmogorov-Arnold Network (KAN)
Theorem	Universal Approximation Theorem	Kolmogorov-Arnold Representation Theorem
Formula (Shallow)	$f(\mathbf{x}) \approx \sum_{i=1}^{N(\epsilon)} a_i \sigma(\mathbf{w}_i \cdot \mathbf{x} + b_i)$	$f(\mathbf{x}) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$
Model (Shallow)		
Formula (Deep)	$\mathrm{MLP}(\mathbf{x}) = (\mathbf{W}_3 \circ \sigma_2 \circ \mathbf{W}_2 \circ \sigma_1 \circ \mathbf{W}_1)(\mathbf{x})$	$\mathrm{KAN}(\mathbf{x}) = (\mathbf{\Phi}_3 \circ \mathbf{\Phi}_2 \circ \mathbf{\Phi}_1)(\mathbf{x})$
Model (Deep)		

Рис.: Архитектура KANs

Почему Tabular DL

- Новые DL модели (такие как TabM, TabR, FT-T) начали показывать результаты сравнимые с GBDT
- Табличные данные зачастую имеет малую размерность
- В статьях [2] и [3] были проведены эксперименты KAN на табличных данных с многообещающими результатами
- Однако в статьях [2] и [3] не был изучен полный потенциал архитектуры KAN

Стек технологий

Основные технологии

- Python — основной язык программирования
- PyTorch — фреймворк для глубокого обучения
- Optuna — оптимизация гиперпараметров
- Wandb — трекинг экспериментов
- CatBoost — градиентный бустинг

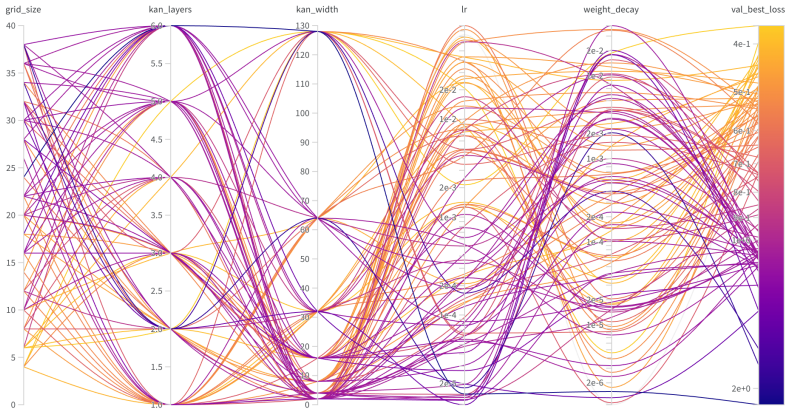
Оборудование

- NVIDIA GeForce RTX 2080 Ti
  - 11GB GDDR6
  - 4352 CUDA ядер
  - Поддержка CUDA

Постановка экспериментов

Гиперпараметры

Проводим тщательных тюнинг гиперпараметров с помощью интерфейса Wandb.



Config parameter	Importance	Correlation
lr		
grid_size		
kan_width		
kan_layers		
Runtime		
weight_decay		

Embeddings для численных фичей

Идея:

Преобразовать все численные атрибуты в векторы по некоторым (обучаемым) правилам $\rightarrow$ на вход в основную модель

- Согласно статье [4], MLP, оснащенная такими эмбедингами, значительно обходит обычный MLP

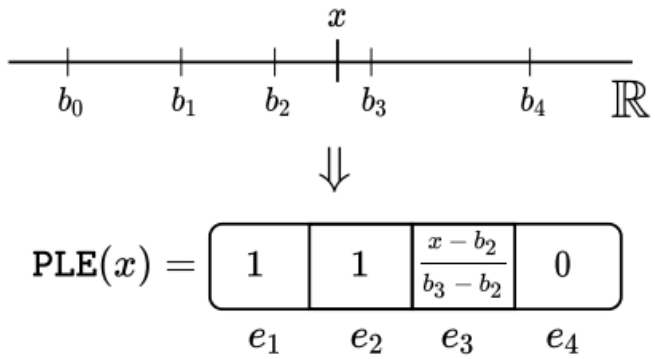
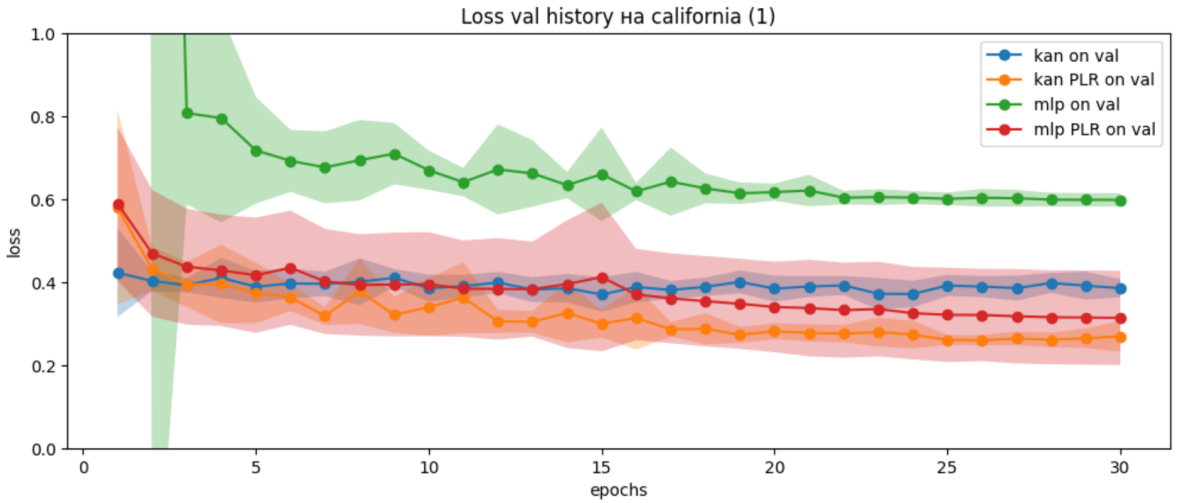


Figure 1: The piecewise linear encoding (PLE) in action for $T = 4$ (see Equation 1).

- Стоит попробовать применить такие эмбедингы перед KAN

Результаты применения

- PLR (Periodic Linear Relu) и PLE-Q (Piecewise Linear Embeddings - Quantile) эмбединги показали лучшие результат среди различных вариантов
- Применение эмбедингов предотвращает переобучение KAN



- Эмбединги лучше влияют на MLP, поэтому KAN и MLP с эмбедингами сравнимы по итоговым результатам

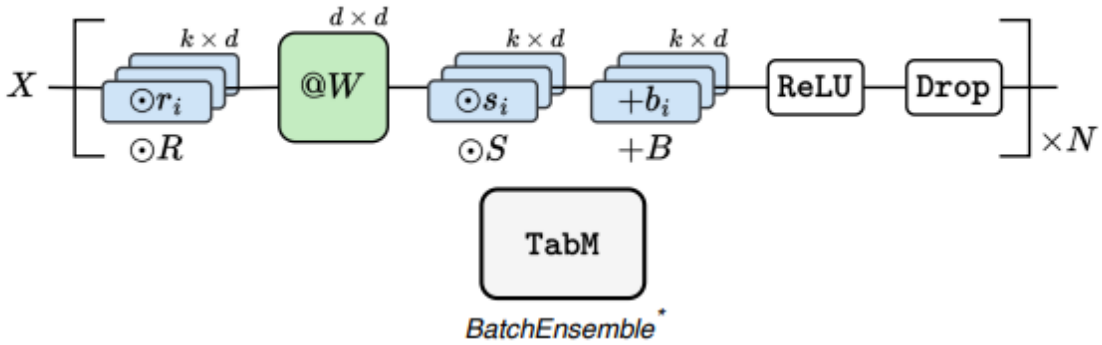
Различные виды полиномов/сплайнов

- В статье [1] в качестве нелинейности внутри KAN используют B-spline (преимущества: локальность, ограниченность)
- В статье [3] дополнительно предложили использовать полиномы Чебышева(Cheby-KAN) и Гауссовы нормальные функции (fast-KAN)

Model	Avg Rank	Train time Ratio	Test time Ratio
KAN	2,4	2,73	3,81
Small KAN	1,6	2,6	3,65
MLP	2	1	1
ChebyKAN	2	1,39	1,55
FastKAN	2,8	1,72	2,02

Ансамблирование

- В статье [5] был предложен метод эффективного моделирования k MLP различных моделей -> State of the art результаты



Идея

Заменим MLP на KAN в этой архитектуре и исследуем результаты

Результаты применения

- Ансамблированная модель - тяжелая архитектура $\rightarrow$ не успевает дообучиться за 10 эпох

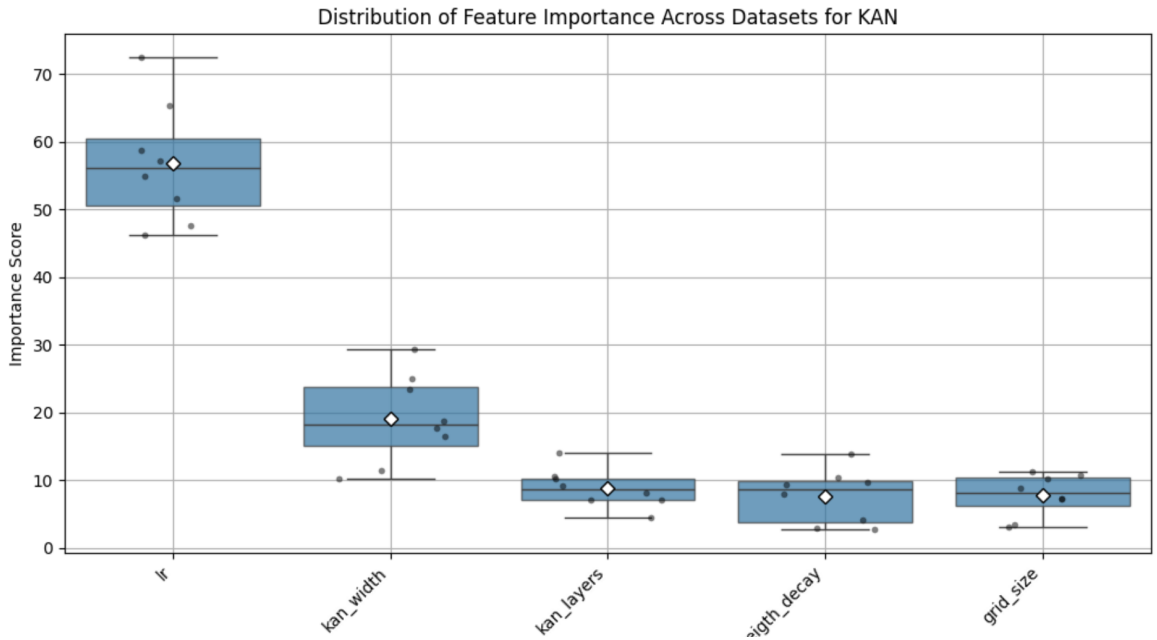
Model	Average Rank	Rank Std	Avg Train Ratio	Train Ratio Std	Avg Test Ratio	Test Ratio Std
FastKAN	5,2	0,98	2,15	0,37	2,39	0,39
FastKAN PLE-Q	2,6	1,02	2,18	0,49	2,68	0,70
FastKAN TabM-mini	5,8	2,32	1,98	0,41	2,18	0,41
FastKAN TabM	6,4	1,20	2,89	1,18	2,95	1,17
FastKAN TabM-mini PLE-Q	3	1,41	10,86	5,50	13,15	6,58
SmallKAN	4,2	1,72	2,67	0,62	3,81	0,79
SmallKAN PLE-Q	2,8	1,60	5,09	2,34	5,96	2,91
SmallKAN TabM-mini	5	1,79	3,98	1,72	4,69	2,01
SmallKAN TabM	4,2	2,04	3,82	0,49	4,36	0,47
SmallKAN TabM-mini PLE-Q	3	1,27	10,96	7,64	12,52	8,86
MLP	5	1,41	1,00	0,00	1,00	0,00
MLP PLE-Q	1,8	0,75	1,44	0,26	1,58	0,21
MLP TabM-mini	4,8	1,72	1,21	0,20	1,27	0,28
MLP TabM	4,2	1,72	1,70	0,37	1,66	0,59
MLP TabM minii PLE-Q	2	1,10	1,61	0,41	1,90	0,54

- Возможное решение - Обучение модели с patience

Аблиционное исследование

Условия эксперимента

- KAN, FastKAN
- 8 наиболее сложных датасетов
- 100 запусков тюнинга для каждой модели (Random search)
- CatBoost для анализа важности



Выводы аблиционного исследования

- Learning rate — самый важный гиперпараметр
- Ширина KAN важнее глубины
- Нет смысла делать глубину > 5
- Grid_size лучше держать в диапазоне 5–10

Вклад и результаты работы

- Общий pipeline для постановки KAN экспериментов
- Исследование влияния BatchNorm, Dropout на работу KAN
- Различные вариации KAN
- Понимание важности эмбедингов для NN архитектур
- Исследование ансамблирования на KAN
- Изучение гипперпараметрво

Заключение

- KAN дают многообещающие результаты на табличных данных
- Однако, MLP со всеми улучшениями зачастую обходит KAN
- PLE-Q - лучший эмбединг для NN архитектур
- Маленькие KAN дают более точные результаты
- Тип KAN играет большую роль
- KAN(B-splines) - лучший вариант
- ChebyshevKAN на практике работают плохо
- BatchNorm для KAN не дает выйгрыша при правильной обработке даннх

Ссылки на литературу

- [1] Ziming Liu et al., KAN: Kolmogorov-Arnold Networks.
- [2]Eleonora Poeta et al., A Benchmarking Study of Kolmogorov-Arnold Networks on Tabular Data.
- [3]Ali Eslamian et al., TabKAN: Advancing Tabular Data Analysis using Kolmogorov- Arnold Network.
- [4] Y.Gorishniy et al., On Embeddings for Numerical Features in Tabular Deep Learning .
- [5] Y.Gorishniy et al., TabM: Advancing Tabular Deep Learning with Parameter-E cient Ensembling.
- [6] A.N. Kolmogorov, On the representation of continuous functions of several variables as superpositions of continuous functions of a smaller number of variables. Dokl. Akad. Nauk, 108(2), 1956.