Сети Колмогорова–Арнольда

Воронцов Константин Вячеславович
k.vorontsov@iai.msu.ru

д.ф.-м.н., профессор РАН • профессор ВМК МГУ,
руководитель лаборатории машинного обучения
и семантического анализа Института ИИ МГУ,
г.н.с. ФИЦ ИУ РАН, профессор МФТИ

«Управление, информация, оптимизация» им. Б. Т. Поляка
Новосибирск, НГУ • 29 июля – 3 августа 2024

Содержание

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Оптимизационные задачи обучения предсказательных моделей

Обучающая выборка: $X^m = (x_i, y_i)_{i=1}^m$, объекты $x_i \in \mathbb{R}^n$, ответы y_i

Задача регрессии: $Y = \mathbb{R}$

$a(x, w)$ — модель регрессии с вектором параметров w

$$Q(w; X^m) = \sum_{i=1}^{m} \big(a(x_i, w) - y_i\big)^2 \to \min_w$$

Задача классификации с двумя классами: $Y = \{-1, +1\}$

$\operatorname{sign} a(x, w)$ — модель классификации с вектором параметров w

$\mathscr{L}(M)$ — невозрастающая функция отступа (margin), например,

$\mathscr{L}(M) = \ln(1 + e^{-M})$, $(1 - M)_+$, e^{-M}, $\frac{1}{1+e^M}$, и др.

$$Q(w; X^m) = \sum_{i=1}^{m} \mathscr{L}\big(\underbrace{a(x_i, w) y_i}_{M_i(w)}\big) \to \min_w$$

Персептрон — математическая модель нейрона

Линейная модель классификации [МакКаллок и Питтс, 1943]:

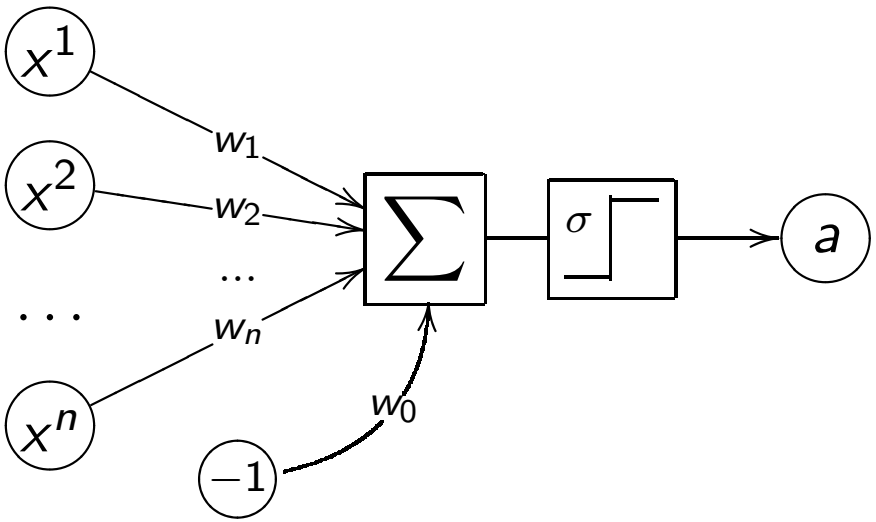
$$a(x,w) = \sigma\big(\langle w,x\rangle\big) = \sigma\left(\sum_{j=1}^{n} w_j f_j(x) - w_0\right)$$

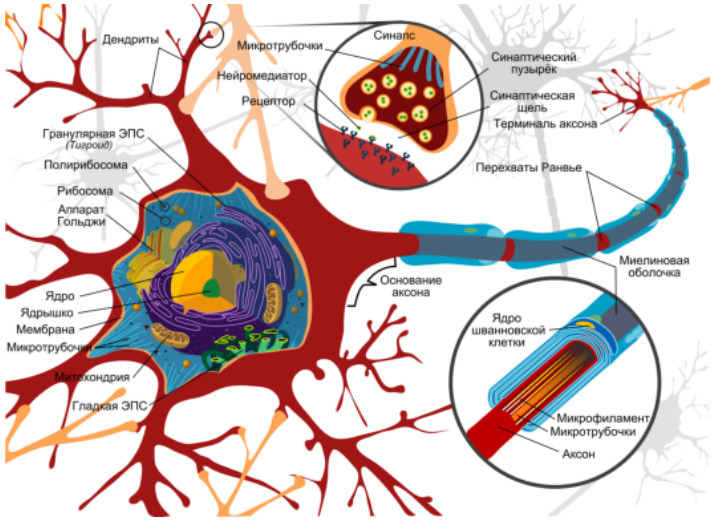
$\sigma(z)$ — функция активации (например, sign или th)

w_j — весовые коэффициенты синаптических связей

w_0 — порог активации

$w, x \in \mathbb{R}^{n+1}$, если ввести константный признак $f_0(x) \equiv -1$





Алгоритм стохастического градиента SG (Stochastic Gradient)

Вход: выборка X^m, темп обучения h, темп забывания λ;
Выход: вектор весов w;

инициализировать веса w_j, $j = 0, \ldots, n$;
инициализировать оценку функционала:
$\bar{Q} := \frac{1}{m} \sum_{i=1}^{m} \mathscr{L}_i(w)$;
повторять
  выбрать объект x_i из X^m случайным образом;
  вычислить потерю: $\varepsilon_i := \mathscr{L}_i(w)$;
  сделать градиентный шаг: $w := w - h\mathscr{L}_i'(w)$;
  оценить функционал: $\bar{Q} := \lambda\varepsilon_i + (1 - \lambda)\bar{Q}$;
пока значение $\bar{Q}$ и/или веса w не сойдутся;

Для многослойных сетей $\mathscr{L}_i'(w)$ вычисляется Back Propagation

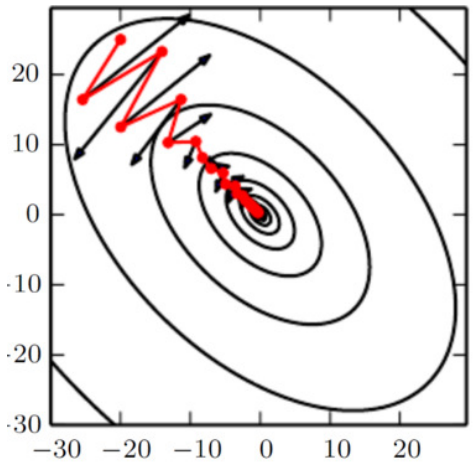
---

Robbins, H., Monro S. A stochastic approximation method. 1951.

Метод инерции (momentum)

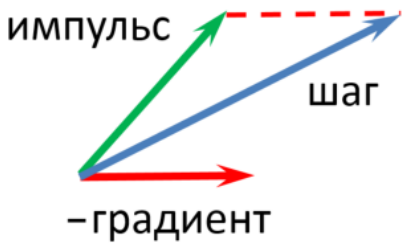
Экспоненциальное скользящее среднее градиента по $\approx \frac{1}{1-\gamma}$ последним итерациям [Б.Т.Поляк, 1964]:

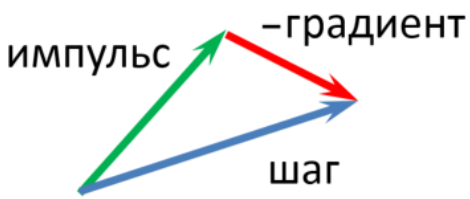
$$v := \gamma v + (1-\gamma)\mathscr{L}_i'(w)$$
$$w := w - \eta v$$



NAG (Nesterov's accelerated gradient) — стохастический градиент с инерцией [Ю.Е.Нестеров, 1983]:

$$v := \gamma v + (1-\gamma)\mathscr{L}_i'(w - \eta\gamma v)$$
$$w := w - \eta v$$



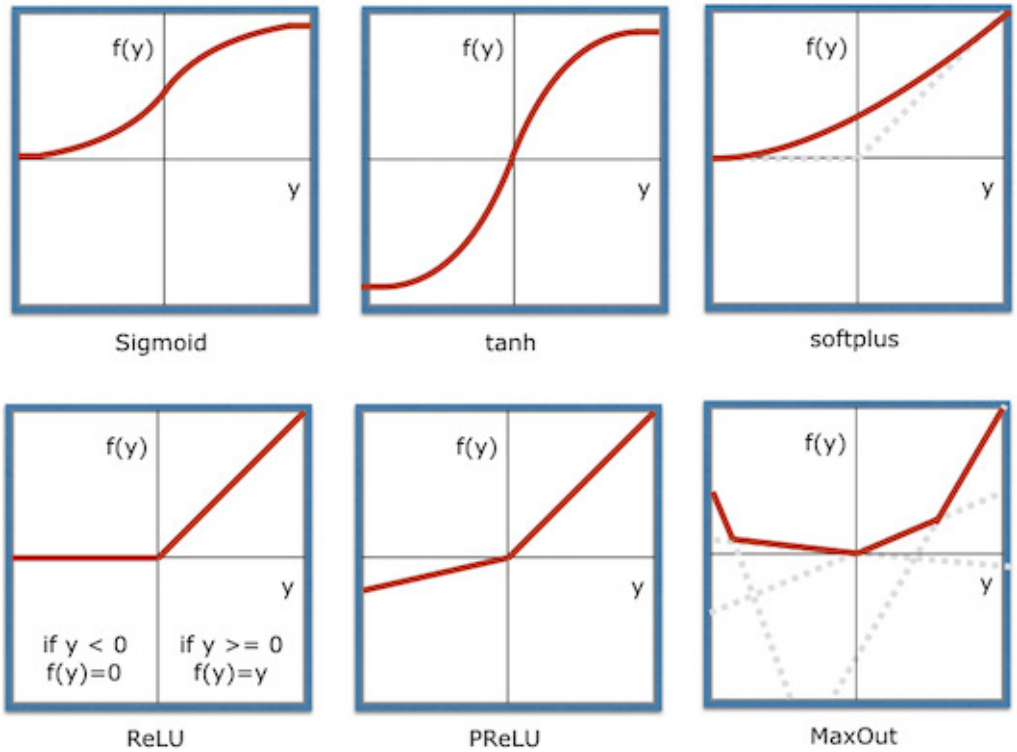


Функции активации

Функции $\sigma(y) = \frac{1}{1+e^{-y}}$ и $\text{th}(y) = \frac{e^{y}-e^{-y}}{e^{y}+e^{-y}}$ могут приводить к затуханию градиентов или «параличу сети»

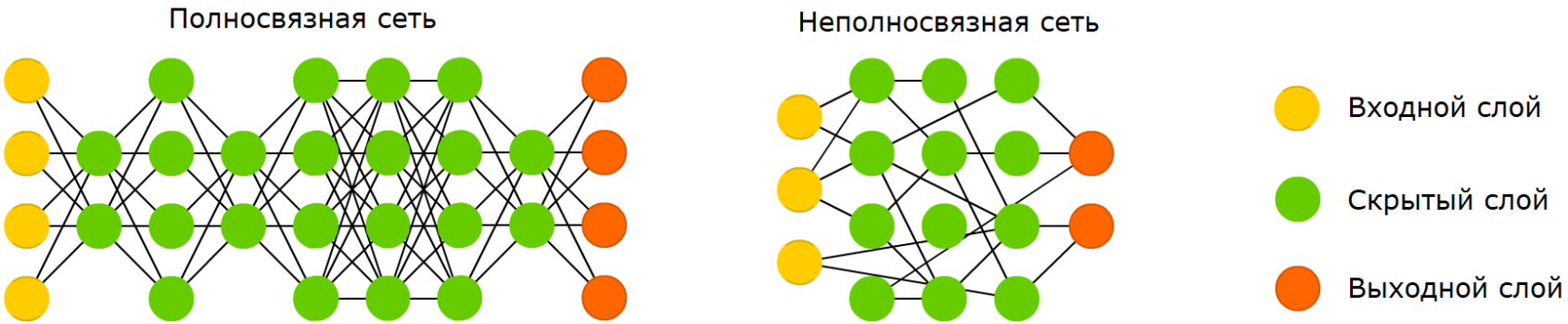
Функция положительной срезки (Rectified Linear Unit, ReLU)

$$\text{ReLU}(y) = \max\{0, y\}; \qquad \text{PReLU}(y) = \max\{0, y\} + \alpha \min\{0, y\}$$



Многослойный персептрон (MultiLayer Perceptron, MLP)

1965: первые многослойные (глубокие) нейронные сети
2012: свёрточная сеть для классификации изображений AlexNet



- *Архитектура сети* — структура слоёв и связей между ними, позволяющая наделять MLP нужными свойствами
- Глубокие нейронные сети (Deep Neural Network, DNN) позволяют принимать на входе и генерировать на выходе *сложно структурированные данные*

Ива́хненко А. Г., Лапа В. Г. Кибернетические предсказывающие устройства. 1965.
Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

Полносвязная многослойная нейронная сеть с L слоями

$x = x^0 = (x_j^0)_{j=0}^n$ — вектор признаков на входе сети, $H_0 = n$

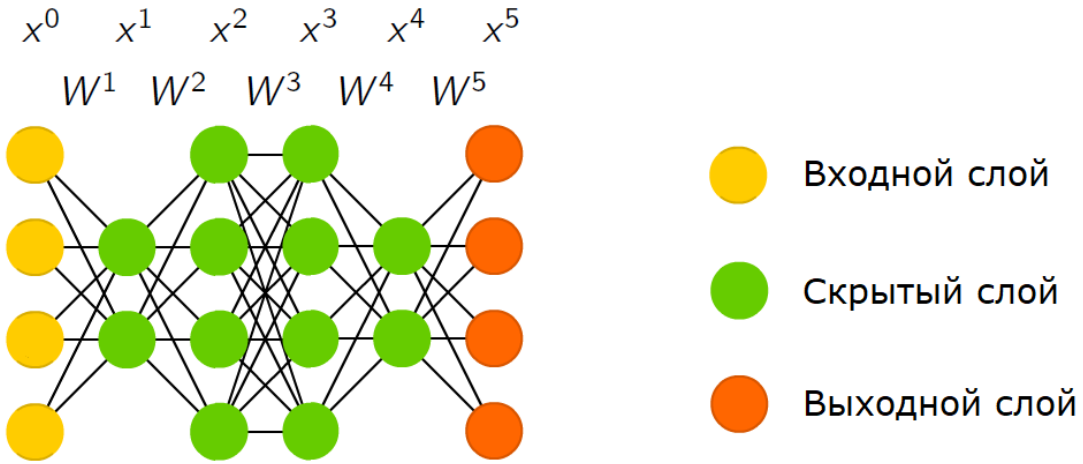
$x_h^l = \sigma_h^l\left(\sum_{k=0}^{H_{l-1}} w_{kh}^l x_k^{l-1}\right)$, σ_h^l — функции активации l-го слоя

$x^l = (x_h^l)_{h=1}^{H_l}$ — вектор на выходе l-го слоя, $x_0^l = -1$

H_l — число нейронов в l-м слое, $l = 1, \ldots, L$

$W^l = (w_{kh}^l)$ — матрица весов l-го слоя, размера $(H_{l-1}+1) \times H_l$

$a(x, w) = x^L$ — выходной вектор сети, обычно $H_L = 1$

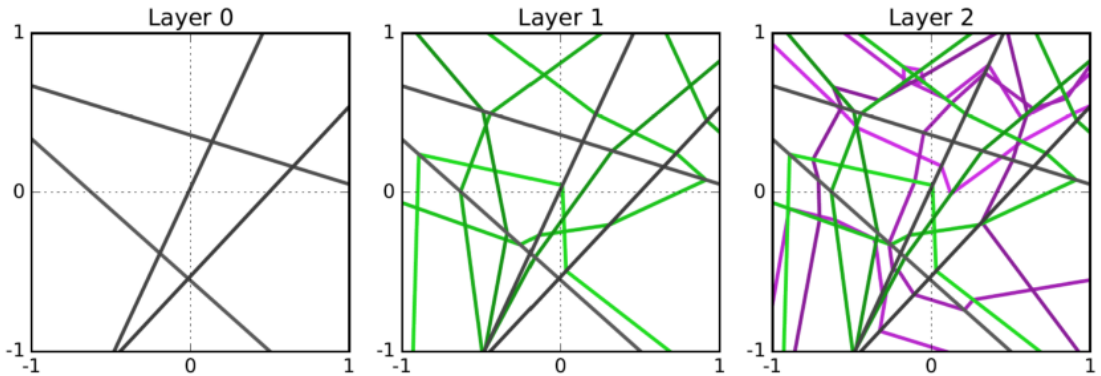


Обоснование DNN. Глубина важнее ширины

A_{LH}^n — семейство полносвязных многослойных сетей $a(x, w)$:
n признаков, L слоёв, H нейронов в каждом слое, $x \in \mathbb{R}^n$,
функции активации кусочно-линейные: ReLU, hard-tanh и т.п.

Мера разнообразия семейства A_{LH}^n — максимальное число участков линейности $a(x, w)$ — выпуклых многогранников в $\mathbb{R}^n$.

Пример. Участки линейности, $n = 2$, $L = 3$, $H = 4$:



Теорема. Разнообразие семейства A_{LH}^n растёт как $O(H^{nL})$.

M.Raghu et al. On the Expressive Power of Deep Neural Networks, 2016.

Избыточная параметризация может ускорять сходимость

Рассмотрим t-й шаг SGD: $\mathscr{L}(x_i w) \to \min\limits_{w}$, $x_i, w \in \mathbb{R}^n$, $i \equiv i(t)$:

$$w^{t+1} := w^t - \eta\, x_i \mathscr{L}'(x_i w^t)$$

Пример избыточной параметризации: $\mathscr{L}(x_i w_1 v) \to \min\limits_{w_1, v}$, $v \in \mathbb{R}$:

$$w_1^{t+1} := w_1^t - \eta\, x_i v^t \mathscr{L}'(x_i w_1^t v^t)$$
$$v^{t+1} := v^t - \eta\, (x_i w_1^t)\, \mathscr{L}'(x_i w_1^t v^t)$$

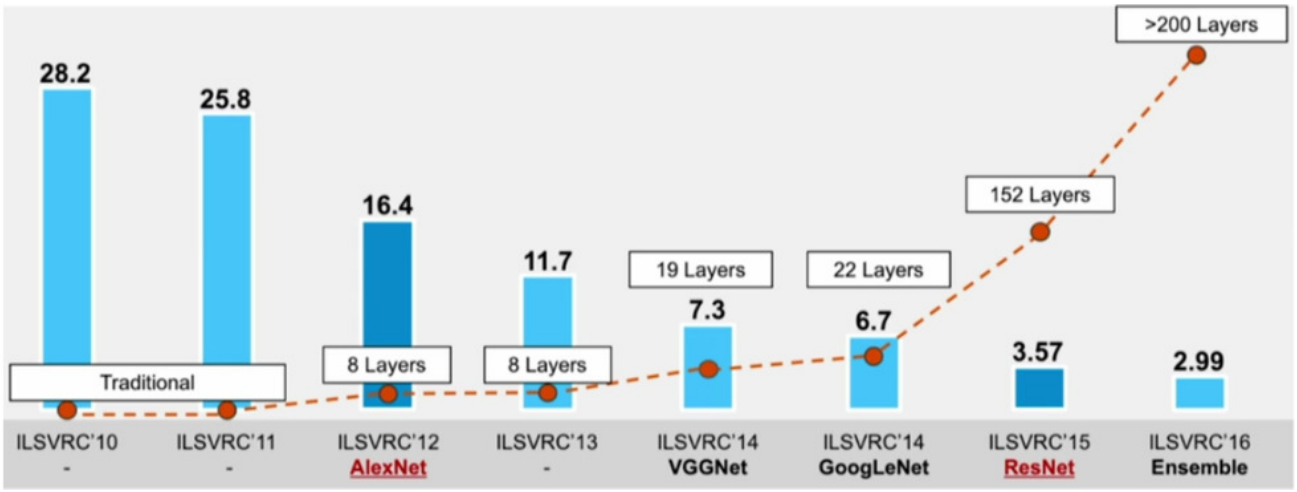
Рекуррентная формула для $w^t = w_1^t v^t$:

$$w^{t+1} := w_1^{t+1} v^{t+1} = w^t - \eta^t x_i \mathscr{L}'(x_i w^t) - \sum_{\tau=1}^{t-1} \eta^{t,\tau} x_{i(\tau)} \mathscr{L}'(x_{i(\tau)} w^\tau)$$

Это (неожиданно!) метод Momentum с адаптивным шагом η^t и адаптивными коэффициентами сглаживания $\eta^{t,\tau}$.

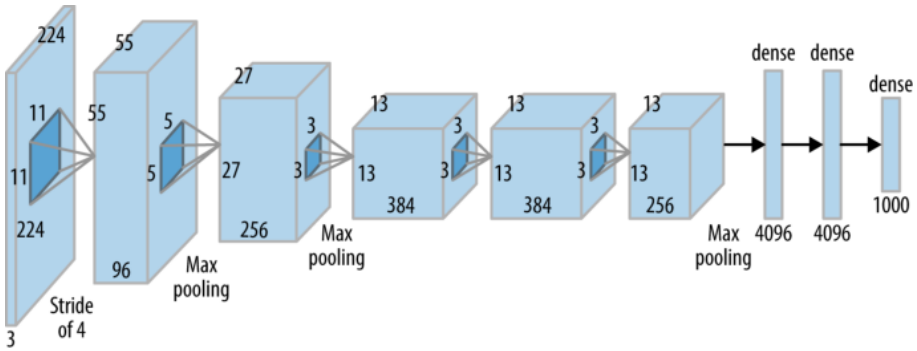
Sanjeev Arora, Nadav Cohen, Elad Hazan. On the Optimization of Deep Networks: Implicit Acceleration by Overparameterization. 2018

ResNet: прорыв 2015 года в классификации изображений



Старт в 2009. Человеческий уровень ошибок 5% пройден в 2015

Свёрточная сеть **AlexNet**:
+ ReLU + Dropout
+ 60M параметров
+ аугментация выборки
+ подбор размеров слоёв
+ GPU

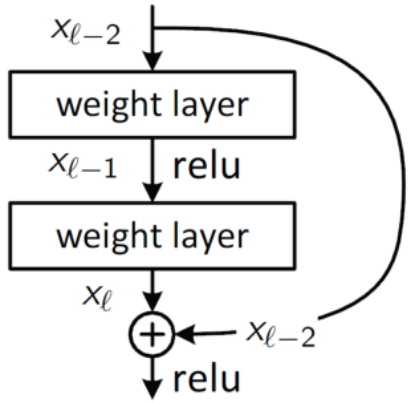


Krizhevsky A. et al. ImageNet classification with deep convolutional neural networks. 2012.

ResNet: остаточная нейронная сеть (Residual NN)

Сквозная связь (skip connection) слоя l с предшествующим слоем $l-d$:

$$x_l = \sigma(Wx_{l-1}) + x_{l-d}$$



Слой l выучивает не новое векторное представление x_l, а его приращение $x_l - x_{l-d}$

- Приращения более устойчивы $\Rightarrow$ улучшается сходимость
- Появляется возможность увеличивать число слоёв
- Обобщение — Highway Networks:

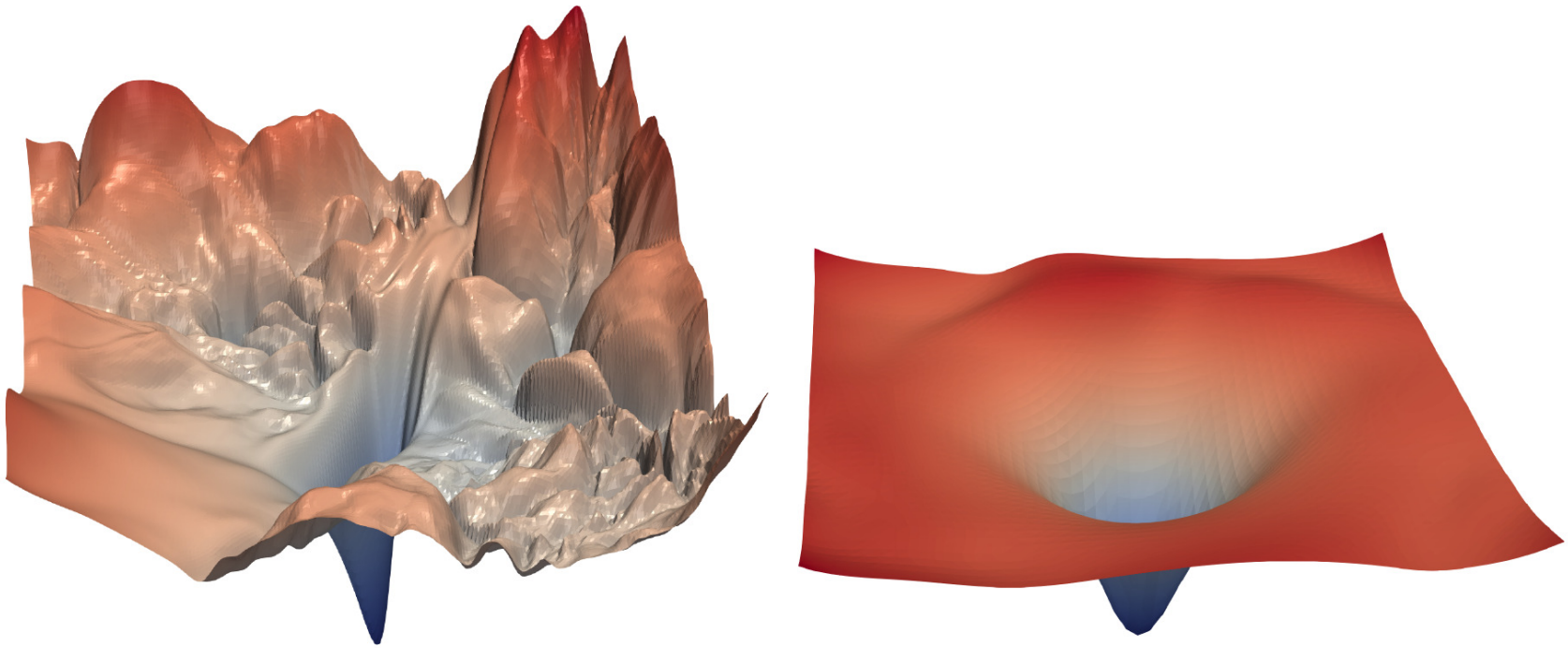
$$x_l = \sigma(Wx_{l-1})\underbrace{\tau(W'x_{l-1})}_{\text{transform gate}} + x_{l-d}\underbrace{\big(1-\tau(W'x_{l-1})\big)}_{\text{carry gate}}$$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. Deep Residual Learning for Image Recognition. 2015

R.K.Srivastava, K.Greff, J.Schmidhuber. Highway Networks. 2015

ResNet: визуализация оптимизационного критерия

Сквозные связи (skip connection) упрощают оптимизируемый критерий, устраняя локальные экстремумы и седловые точки:



without skip connections

with skip connections

Hao Li et al. Visualizing the Loss Landscape of Neural Nets. 2018

Любую ли функцию можно представить нейросетью?

Любая булева функция представима в виде ДНФ, следовательно, в виде двухслойной сети. А непрерывная?

Решение тринадцатой (из 23) проблем Гильберта (1900):

Теорема [Колмогоров, Арнольд, 1957]

Любая непрерывная функция n аргументов на единичном кубе $[0,1]^n$ представима в виде суперпозиции непрерывных функций одного аргумента и операции сложения:

$$f(x_1,\ldots,x_n) = \sum_{k=1}^{2n+1} \Phi_k\left(\sum_{j=1}^{n} \varphi_{jk}(x_j)\right),$$

где Φ_k, φ_{jk} — непрерывные функции, и φ_{jk} не зависят от f.

А.Н.Колмогоров. О представлении непрерывных функций нескольких переменных суперпозициями непрерывных функций меньшего числа переменных. 1956.
В.И.Арнольд. О функции трех переменных. 1957.

Двухслойные сети — аппроксиматоры непрерывных функций

Функция $\sigma(z)$ — *сигмоида*, если $\lim\limits_{z\to-\infty}\sigma(z)=0$ и $\lim\limits_{z\to+\infty}\sigma(z)=1$.

Теорема Цыбенко (1989)

Если $\sigma(z)$ — непрерывная сигмоида, то для любой непрерывной на $[0,1]^n$ функции $f(x)$ существуют такие значения параметров H, $\alpha_h\in\mathbb{R}$, $w_h\in\mathbb{R}^n$, $w_0\in\mathbb{R}$, что двухслойная сеть

$$a(x)=\sum_{h=1}^{H}\alpha_h\sigma\big(\langle x,w_h\rangle-w_0\big)$$

равномерно приближает $f(x)$ с любой точностью ε:

$$\big|a(x)-f(x)\big|<\varepsilon,\ \text{для всех } x\in[0,1]^n.$$

George Cybenko. Approximation by Superpositions of a Sigmoidal function. Mathematics of Control, Signals, and Systems. 1989.

Имеет ли теорема (ТКА) отношение к нейросетям?

Вроде да:

- двухслойная суперпозиция с суммой
- имеются универсальные аппроксимационные свойства

На самом деле — нет:

- это точное представление; нам достаточно аппроксимации
- функции Φ_k, φ_{jk} не гладкие и сложно устроены
- нет возможности фиксировать их как функции активации
- нет оптимизационной процедуры для их обучения
- нет весовых коэффициентов W для обучения
- нет постановки оптимизационной задачи обучения
- число слоёв 2 и число нейронов $[2n+1, n]$ фиксированы
- нет возможности менять архитектуру, делать сеть глубокой

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Преодоление ограничений ТКА: архитектура KAN

- произвольная архитектура: любая глубина L, ширина H_l
- обучаемые функции активации — одномерные сплайны
- обучение: оптимизация LBFGS, регуляризация, BackProp

$x_h^l = \sum_{k=1}^{H_{l-1}} \phi_{kh}^l(x_k^{l-1})$ — вектор на выходе l-го слоя, $l = 1, \ldots, L$

$\phi_{kh}^l(x) = w_b b(x) + w_s s(x)$ — функция одной переменной

$b(x) = \frac{x}{1+e^{-x}}$ — сглаженный ReLU, аналог сквозной связи

$s(x) = \sum_i c_i B_i(x)$ — одномерный сплайн по базису $\{B_i\}$

w_b, w_s — избыточная параметризация

Таким образом, KAN — это MLP с удвоенным числом слоёв и базисными функциями сплайнов в качестве активаций

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Преодоление ограничений ТКА: оптимизация KAN

B-сплайн фиксированной степени k (обычно $k = 3$)

- по сетке из $G + 2k + 1$ точек $\{t_{-k}, \ldots, t_0, \ldots, t_G, \ldots, t_{G+k}\}$
- базисная функция $B_i(x) = 0$ вне отрезка $[t_{i-k}, \ldots, t_{i+1}]$
- постепенное увеличение G, определяющего число точек

Разреживающая регуляризация $\mu_1 L_1 + \mu_2 \text{Entropy} \to \min$

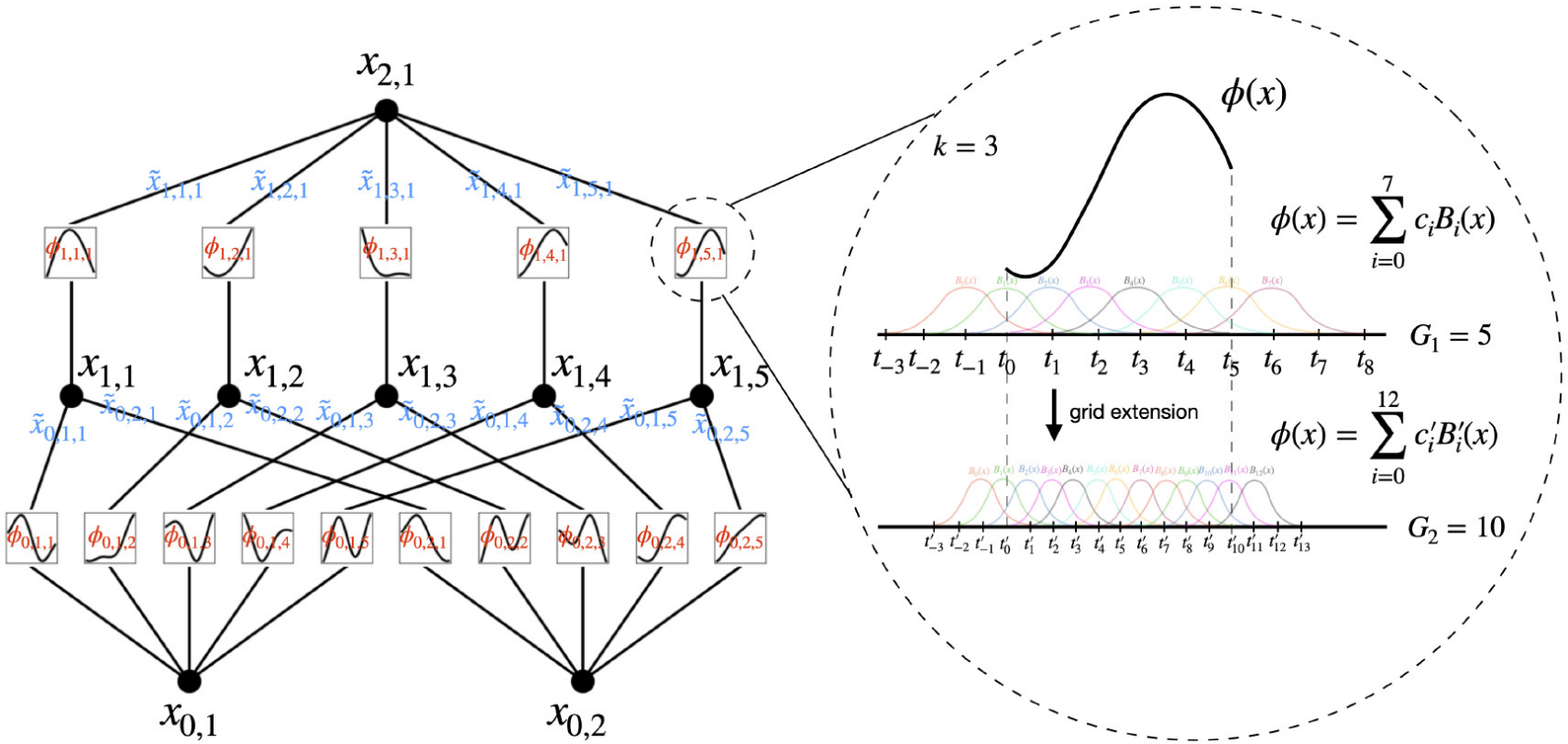
- $L_1 = \sum_{l=1}^{L} \sum_{h=1}^{H_l} \frac{1}{H_{l-1}} \sum_{k=1}^{H_{l-1}} |\phi_{kh}^l|$
- $\text{Entropy} = -\sum_{l=1}^{L} \sum_{h=1}^{H_l} \sum_{k=1}^{H_{l-1}} \frac{|\phi_{kh}^l|}{|\Phi^l|} \log \frac{|\phi_{kh}^l|}{|\Phi^l|}, \quad \Phi^l = \sum_{h=1}^{H_l} \sum_{k=1}^{H_{l-1}} |\phi_{kh}^l|$

Инициализация: $w_s = 1$, w_b: Xavier, $c_i \sim \mathcal{N}(0, \sigma^2)$, $\sigma = 0.1$

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Постепенное увеличение числа точек в сетке сплайна

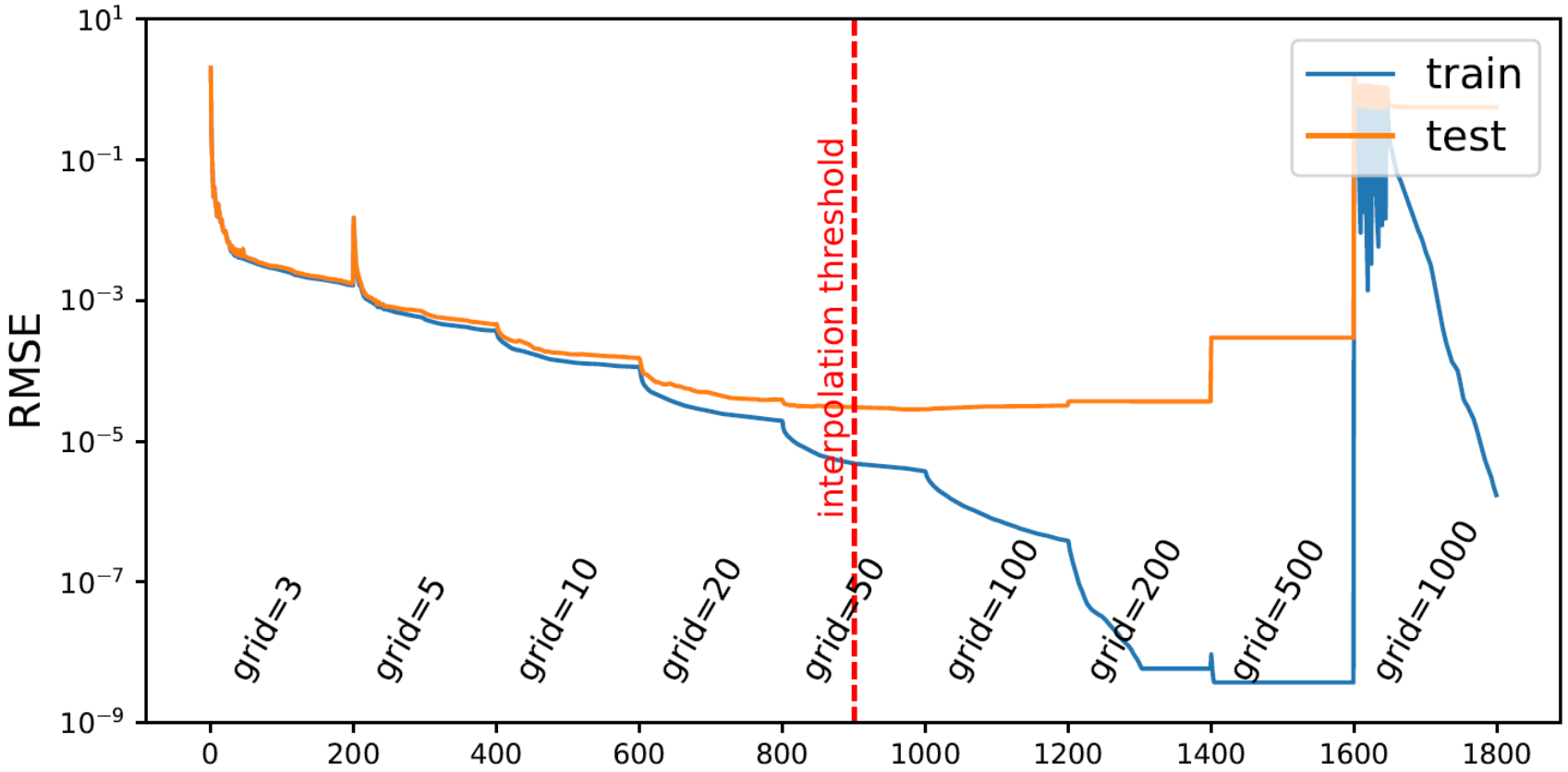
Гранулирование сетки $G_1 = 5 \;\rightarrow\; G_2 = 10$ увеличивает число параметров c_i во всех одномерных B-сплайнах сети:



Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Пример: синтетическая функция $f(x, y) = \exp\left(\sin(\pi x) + y^2\right)$

Выбор оптимального размера сетки $G = 50$ для KAN[2,5,1]

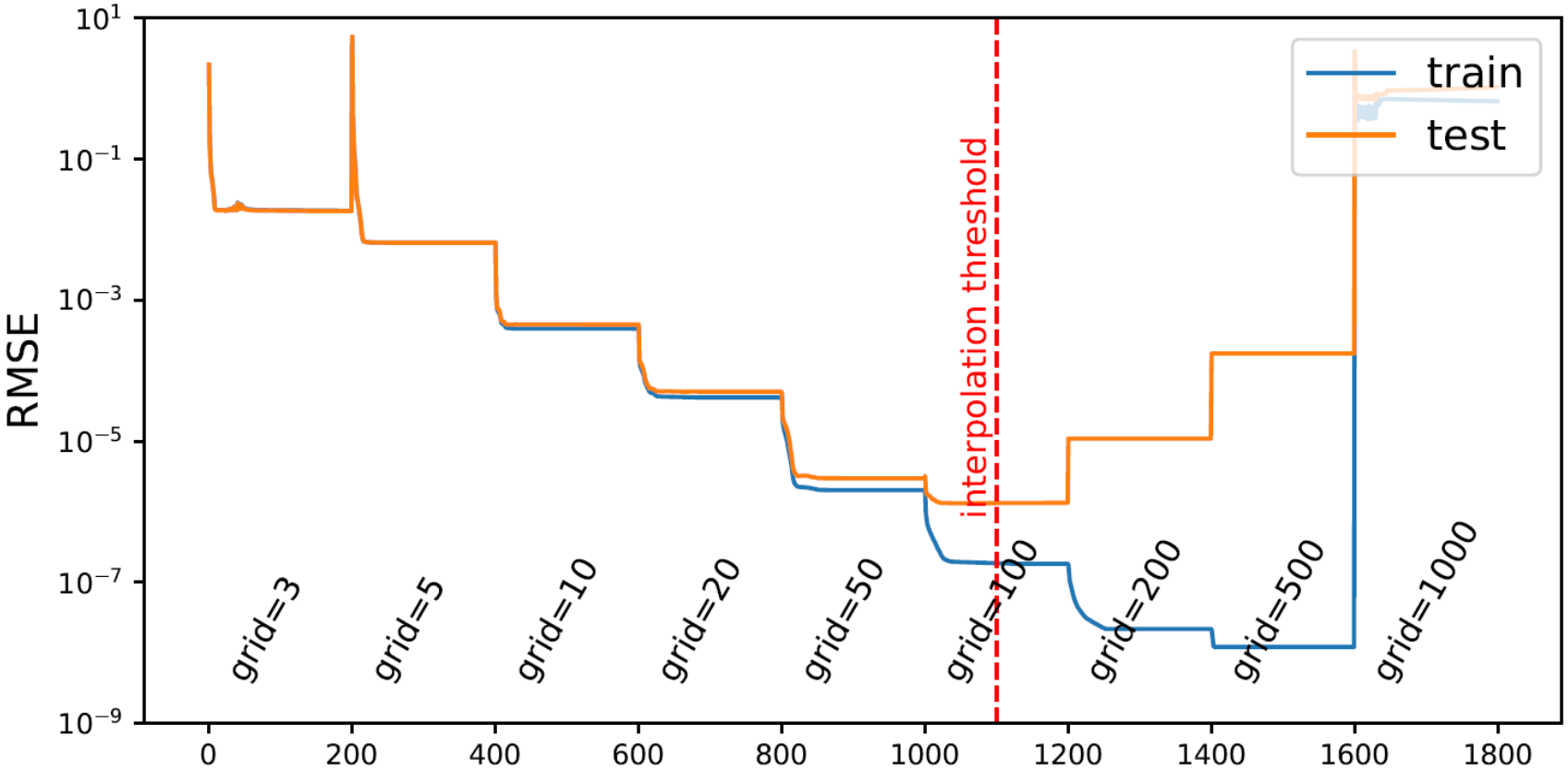


При каждом увеличении размера сетки RMSE падает скачком

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Пример: синтетическая функция $f(x, y) = \exp\left(\sin(\pi x) + y^2\right)$

Архитектура KAN[2,1,1] лучше подходит для данной функции

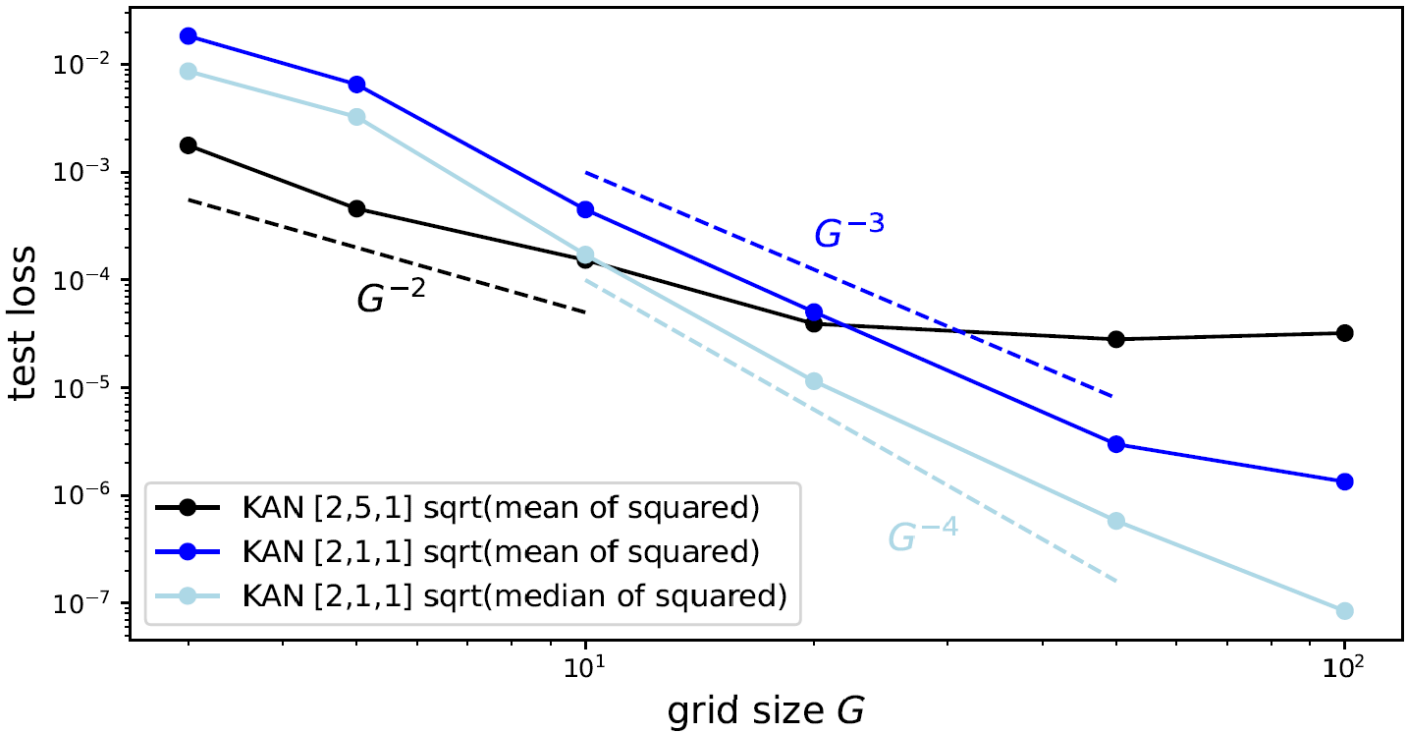


Лучший RMSE достигается при уменьшении H_1 и увеличении G

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Пример: синтетическая функция $f(x, y) = \exp\left(\sin(\pi x) + y^2\right)$

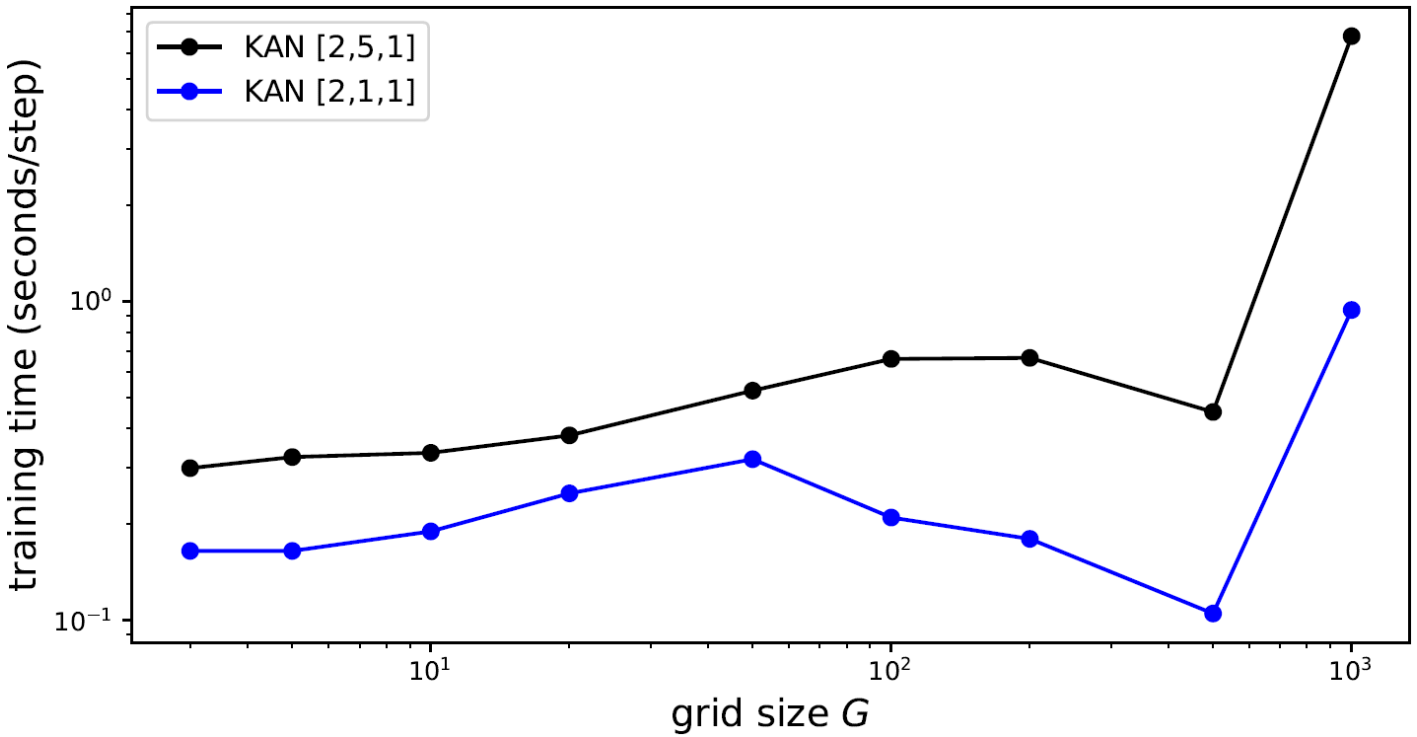
Скорость убывания test-RMSE в лучшем случае $O(G^{-4})$ (достигается для медианы, а не для среднего из-за редких больших ошибок, связанных с краевыми эффектами)



Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Пример: синтетическая функция $f(x, y) = \exp\left(\sin(\pi x) + y^2\right)$

Зависимость времени обучения от размера сетки G



При удачном выборе архитектуры сети под задачу увеличение G может сокращать время обучения

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

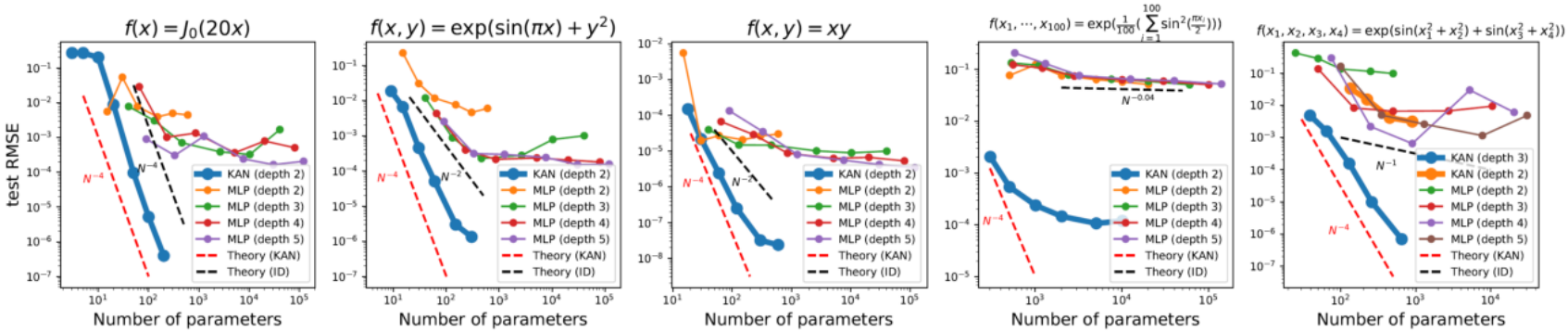
Скорость сходимости (scaling law) при идеальной архитектуре

Число параметров KAN $N = O(H^2LG)$

Теоретическая скорость сходимости для KAN из теории одномерных сплайнов: RMSE $= O(G^{-k-1})$

Теоретические оценки для MLP: $O(N^{-\alpha})$, где α порядка 1

Эксперимент на 5 синтетических функциях:
1800 итераций LBFGS, $G = \{3, 5, 10, 20, 50, 100, 200, 500, 1000\}$



MLP сходится медленнее, быстрее выходит на плато

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Этапы оптимизации архитектуры KAN под задачу

1. **регуляризация** L_1 + Entropy, приводящая к разреживанию функций активации ϕ_{kh}^l
2. **отсечение** (pruning) i-го узла в l-м слое по условию

$$\max_k |\phi_{ki}^{l-1}| < \theta, \quad \max_k |\phi_{ik}^{l+1}| < \theta, \quad \theta = 0.01$$

3. **визуализация** всех функций активации $\phi_{kh}^l(x_k^{l-1})$
4. **символьная регрессия** (symbolic regression) — замена некоторых функций активации подходящими фиксированными функциями $f(\cdot)$:

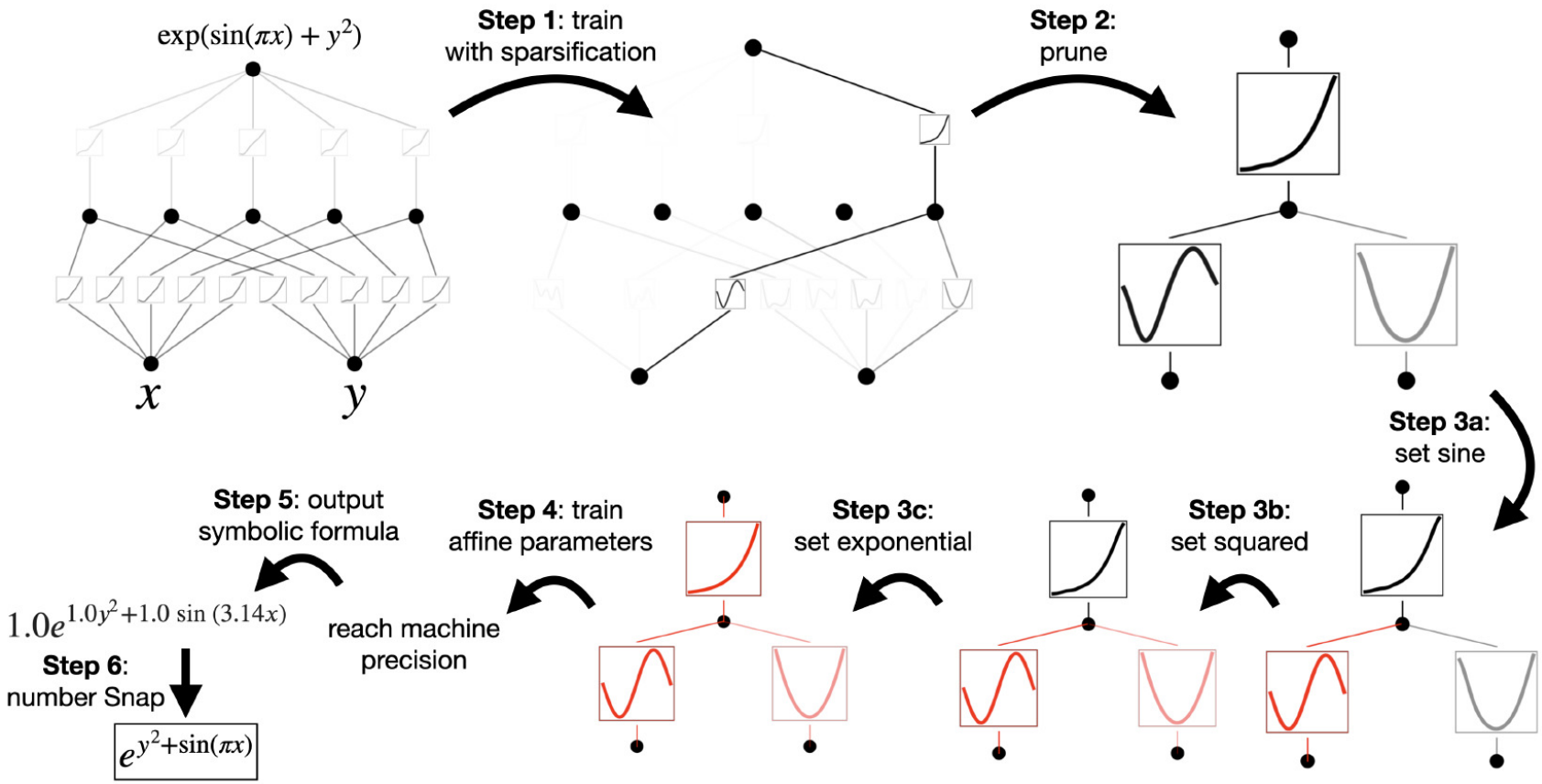
$$\phi_{kh}^l(x_k^{l-1}) \quad \to \quad c\, f\left(a x_k^{l-1} + b\right) + d$$

a, b, c, d — обучаемые параметры сдвига/масштабирования

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Оптимизация архитектуры. Пример: $f(x, y) = \exp\left(\sin(\pi x) + y^2\right)$

разреживание $\rightarrow$ упрощение $\rightarrow$ визуализация $\rightarrow$ символизация



Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Эксперименты по аппроксимации специальных функций

Name	scipy.special API	Minimal KAN shape test RMSE $< 10^{-2}$	Minimal KAN test RMSE	Best KAN shape	Best KAN test RMSE	MLP test RMSE
Jacobian elliptic functions	ellipj(x, y)	[2,2,1]	7.29×10^{-3}	[2,3,2,1,1,1]	$\mathbf{1.33 \times 10^{-4}}$	6.48×10^{-4}
Incomplete elliptic integral of the first kind	ellipkinc(x, y)	[2,2,1,1]	1.00×10^{-3}	[2,2,1,1,1]	$\mathbf{1.24 \times 10^{-4}}$	5.52×10^{-4}
Incomplete elliptic integral of the second kind	ellipeinc(x, y)	[2,2,1,1]	8.36×10^{-5}	[2,2,1,1]	$\mathbf{8.26 \times 10^{-5}}$	3.04×10^{-4}
Bessel function of the first kind	jv(x, y)	[2,2,1]	4.93×10^{-3}	[2,3,1,1,1]	$\mathbf{1.64 \times 10^{-3}}$	5.52×10^{-3}
Bessel function of the second kind	yv(x, y)	[2,3,1]	1.89×10^{-3}	[2,2,2,1]	$\mathbf{1.49 \times 10^{-5}}$	3.45×10^{-4}
Modified Bessel function of the second kind	kv(x, y)	[2,1,1]	4.89×10^{-3}	[2,2,1]	$\mathbf{2.52 \times 10^{-5}}$	1.67×10^{-4}
Modified Bessel function of the first kind	iv(x, y)	[2,4,3,2,1,1]	9.28×10^{-3}	[2,4,3,2,1,1]	$\mathbf{9.28 \times 10^{-3}}$	1.07×10^{-2}
Associated Legendre function $(m = 0)$	lpmv$(0, x, y)$	[2,2,1]	5.25×10^{-5}	[2,2,1]	$\mathbf{5.25 \times 10^{-5}}$	1.74×10^{-2}
Associated Legendre function $(m = 1)$	lpmv$(1, x, y)$	[2,4,1]	6.90×10^{-4}	[2,4,1]	$\mathbf{6.90 \times 10^{-4}}$	1.50×10^{-3}
Associated Legendre function $(m = 2)$	lpmv$(2, x, y)$	[2,2,1]	4.88×10^{-3}	[2,3,2,1]	$\mathbf{2.26 \times 10^{-4}}$	9.43×10^{-4}
spherical harmonics $(m = 0, n = 1)$	sph_harm$(0, 1, x, y)$	[2,1,1]	2.21×10^{-7}	[2,1,1]	$\mathbf{2.21 \times 10^{-7}}$	1.25×10^{-6}
spherical harmonics $(m = 1, n = 1)$	sph_harm$(1, 1, x, y)$	[2,2,1]	7.86×10^{-4}	[2,3,2,1]	$\mathbf{1.22 \times 10^{-4}}$	6.70×10^{-4}
spherical harmonics $(m = 0, n = 2)$	sph_harm$(0, 2, x, y)$	[2,1,1]	1.95×10^{-7}	[2,1,1]	$\mathbf{1.95 \times 10^{-7}}$	2.85×10^{-6}
spherical harmonics $(m = 1, n = 2)$	sph_harm$(1, 2, x, y)$	[2,2,1]	4.70×10^{-4}	[2,2,1,1]	$\mathbf{1.50 \times 10^{-5}}$	1.84×10^{-3}
spherical harmonics $(m = 2, n = 2)$	sph_harm$(2, 2, x, y)$	[2,2,1]	1.12×10^{-3}	[2,2,3,2,1]	$\mathbf{9.45 \times 10^{-5}}$	6.21×10^{-4}

KAN точнее аппроксимирует специальные функции, чем MLP
KAN находит структуру суперпозиции, близкую к идеальной

При использовании символизации, KAN находит формулы физических законов с интерпретируемой структурой (см. далее)

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Эксперименты с физическими формулами (Feynman dataset)

Feynman Eq.	Original Formula	Dimensionless formula	Variables	Human-constructed KAN shape	Pruned KAN shape (smallest shape that achieves RMSE < 10^{-2})	Pruned KAN shape (lowest loss)	Human-constructed KAN loss (lowest test RMSE)	Pruned KAN loss (lowest test RMSE)	Unpruned KAN loss (lowest test RMSE)	MLP loss (lowest test RMSE)
I.6.2	$\exp(-\frac{\theta^2}{2\sigma^2})/\sqrt{2\pi\sigma^2}$	$\exp(-\frac{\theta^2}{2\sigma^2})/\sqrt{2\pi\sigma^2}$	θ,σ	[2,2,1,1]	[2,2,1]	[2,2,1,1]	7.66×10^{-5}	$\mathbf{2.86\times10^{-5}}$	4.60×10^{-5}	1.45×10^{-4}
I.6.2b	$\exp(-\frac{(\theta-\theta_1)^2}{2\sigma^2})/\sqrt{2\pi\sigma^2}$	$\exp(-\frac{(\theta-\theta_1)^2}{2\sigma^2})/\sqrt{2\pi\sigma^2}$	θ,θ_1,σ	[3,2,2,1,1]	[3,4,1]	[3,2,2,1,1]	1.22×10^{-3}	$\mathbf{4.45\times10^{-4}}$	1.25×10^{-3}	7.40×10^{-4}
I.9.18	$\frac{Gm_1m_2}{(x_2-x_1)^2+(y_2-y_1)^2+(z_2-z_1)^2}$	$\frac{a}{(b-1)^2+(c-d)^2+(e-f)^2}$	a,b,c,d,e,f	[6,4,2,1,1]	[6,4,1,1]	[6,4,1,1]	$\mathbf{1.48\times10^{-3}}$	8.62×10^{-3}	6.56×10^{-3}	1.59×10^{-3}
I.12.11	$q(E_f+Bv\sin\theta)$	$1+a\sin\theta$	a,θ	[2,2,2,1]	[2,2,1]	[2,2,1]	2.07×10^{-3}	1.39×10^{-3}	9.13×10^{-4}	$\mathbf{6.71\times10^{-4}}$
I.13.12	$Gm_1m_2(\frac{1}{r_2}-\frac{1}{r_1})$	$a(\frac{1}{b}-1)$	a,b	[2,2,1]	[2,2,1]	[2,2,1]	7.22×10^{-3}	4.81×10^{-3}	2.72×10^{-3}	$\mathbf{1.42\times10^{-3}}$
I.15.3x	$\frac{x-ut}{\sqrt{1-(\frac{u}{c})^2}}$	$\frac{1-a}{\sqrt{1-b^2}}$	a,b	[2,2,1,1]	[2,1,1]	[2,2,1,1,1]	7.35×10^{-3}	1.58×10^{-3}	1.14×10^{-3}	$\mathbf{8.54\times10^{-4}}$
I.16.6	$\frac{u+v}{1+\frac{uv}{c^2}}$	$\frac{a+b}{1+ab}$	a,b	[2,2,2,2,2,1]	[2,2,1]	[2,2,1]	1.06×10^{-3}	1.19×10^{-3}	1.53×10^{-3}	$\mathbf{6.20\times10^{-4}}$
I.18.4	$\frac{m_1r_1+m_2r_2}{m_1+m_2}$	$\frac{1+ab}{1+a}$	a,b	[2,2,2,1,1]	[2,2,1]	[2,2,1]	3.92×10^{-4}	$\mathbf{1.50\times10^{-4}}$	1.32×10^{-3}	3.68×10^{-4}
I.26.2	$\arcsin(n\sin\theta_2)$	$\arcsin(n\sin\theta_2)$	n,θ_2	[2,2,2,1,1]	[2,2,1]	[2,2,2,1,1]	1.22×10^{-1}	$\mathbf{7.90\times10^{-4}}$	8.63×10^{-4}	1.24×10^{-3}
I.27.6	$\frac{1}{\frac{1}{d_1}+\frac{n}{d_2}}$	$\frac{1}{1+ab}$	a,b	[2,2,1,1]	[2,1,1]	[2,1,1]	2.22×10^{-4}	$\mathbf{1.94\times10^{-4}}$	2.14×10^{-4}	2.46×10^{-4}
I.29.16	$\sqrt{x_1^2+x_2^2-2x_1x_2\cos(\theta_1-\theta_2)}$	$\sqrt{1+a^2-2a\cos(\theta_1-\theta_2)}$	a,θ_1,θ_2	[3,2,2,3,2,1,1]	[3,2,2,1]	[3,2,3,1]	2.36×10^{-1}	3.99×10^{-3}	$\mathbf{3.20\times10^{-3}}$	4.64×10^{-3}
I.30.3	$I_{*,0}\frac{\sin^2(\frac{n\theta}{2})}{\sin^2(\frac{\theta}{2})}$	$\frac{\sin^2(\frac{n\theta}{2})}{\sin^2(\frac{\theta}{2})}$	n,θ	[2,3,2,2,1,1]	[2,4,3,1]	[2,3,2,3,1,1]	3.85×10^{-1}	$\mathbf{1.03\times10^{-3}}$	1.11×10^{-2}	1.50×10^{-2}
I.30.5	$\arcsin(\frac{\lambda}{nd})$	$\arcsin(\frac{a}{n})$	a,n	[2,1,1]	[2,1,1]	[2,1,1,1,1,1]	2.23×10^{-4}	$\mathbf{3.49\times10^{-5}}$	6.92×10^{-5}	9.45×10^{-5}
I.37.4	$I_*=I_1+I_2+2\sqrt{I_1I_2}\cos\delta$	$1+a+2\sqrt{a}\cos\delta$	a,δ	[2,3,2,1]	[2,2,1]	[2,2,1]	7.57×10^{-5}	$\mathbf{4.91\times10^{-6}}$	3.41×10^{-4}	5.67×10^{-4}
I.40.1	$n_0\exp(-\frac{mgx}{k_bT})$	n_0e^{-a}	n_0,a	[2,1,1]	[2,2,1]	[2,2,1,1,1,2,1]	3.45×10^{-3}	5.01×10^{-4}	$\mathbf{3.12\times10^{-4}}$	3.99×10^{-4}
I.44.4	$nk_bT\ln(\frac{V_2}{V_1})$	$n\ln a$	n,a	[2,2,1]	[2,2,1]	[2,2,1]	$\mathbf{2.30\times10^{-5}}$	2.43×10^{-5}	1.10×10^{-4}	3.99×10^{-4}
I.50.26	$x_1(\cos(\omega t)+\alpha\cos^2(wt))$	$\cos a+\alpha\cos^2 a$	a,α	[2,2,3,1]	[2,3,1]	[2,3,2,1]	$\mathbf{1.52\times10^{-4}}$	5.82×10^{-4}	4.90×10^{-4}	1.53×10^{-3}
II.2.42	$\frac{k(T_2-T_1)A}{d}$	$(a-1)b$	a,b	[2,2,1]	[2,2,1]	[2,2,2,1]	8.54×10^{-4}	7.22×10^{-4}	1.22×10^{-3}	$\mathbf{1.81\times10^{-4}}$
II.6.15a	$\frac{3}{4\pi\epsilon}\frac{p_dz}{r^5}\sqrt{x^2+y^2}$	$\frac{1}{4\pi}c\sqrt{a^2+b^2}$	a,b,c	[3,2,2,2,1]	[3,2,1,1]	[3,2,1,1]	2.61×10^{-3}	3.28×10^{-3}	1.35×10^{-3}	$\mathbf{5.92\times10^{-4}}$
II.11.7	$n_0(1+\frac{p_dE_f\cos\theta}{k_bT})$	$n_0(1+a\cos\theta)$	n_0,a,θ	[3,3,3,2,2,1]	[3,3,1,1]	[3,3,1,1]	7.10×10^{-3}	8.52×10^{-3}	5.03×10^{-3}	$\mathbf{5.92\times10^{-4}}$
II.11.27	$\frac{n\alpha}{1-\frac{n\alpha}{3}}\epsilon E_f$	$\frac{n\alpha}{1-\frac{n\alpha}{3}}$	n,α	[2,2,1,2,1]	[2,1,1]	[2,2,1]	2.67×10^{-5}	4.40×10^{-5}	$\mathbf{1.43\times10^{-5}}$	7.18×10^{-5}
II.35.18	$\frac{n_0}{\exp(\frac{\mu_mB}{k_bT})+\exp(-\frac{\mu_mB}{k_bT})}$	$\frac{n_0}{\exp(a)+\exp(-a)}$	n_0,a	[2,1,1]	[2,1,1]	[2,1,1,1]	4.13×10^{-4}	1.58×10^{-4}	$\mathbf{7.71\times10^{-5}}$	7.92×10^{-5}
II.36.38	$\frac{\mu_mB}{k_bT}+\frac{\mu_m\alpha M}{\epsilon c^2k_bT}$	$a+\alpha b$	a,α,b	[3,3,1]	[3,2,1]	[3,2,1]	2.85×10^{-3}	$\mathbf{1.15\times10^{-3}}$	3.03×10^{-3}	2.15×10^{-3}
II.38.3	$\frac{YAx}{d}$	$\frac{a}{b}$	a,b	[2,1,1]	[2,1,1]	[2,2,1,1,1]	1.47×10^{-4}	$\mathbf{8.78\times10^{-5}}$	6.43×10^{-4}	5.26×10^{-4}
III.9.52	$\frac{p_dE_f}{h}\frac{\sin^2((\omega-\omega_0)t/2)}{((\omega-\omega_0)t/2)^2}$	$a\frac{\sin^2(\frac{b-c}{2})}{(\frac{b-c}{2})^2}$	a,b,c	[3,2,3,1,1]	[3,3,2,1]	[3,3,2,1,1,1]	4.43×10^{-2}	3.90×10^{-3}	2.11×10^{-2}	$\mathbf{9.07\times10^{-4}}$
III.10.19	$\mu_m\sqrt{B_x^2+B_y^2+B_z^2}$	$\sqrt{1+a^2+b^2}$	a,b	[2,1,1]	[2,1,1]	[2,1,2,1]	2.54×10^{-3}	1.18×10^{-3}	8.16×10^{-4}	$\mathbf{1.67\times10^{-4}}$
III.17.37	$\beta(1+\alpha\cos\theta)$	$\beta(1+\alpha\cos\theta)$	α,β,θ	[3,3,3,2,2,1]	[3,3,1]	[3,3,1]	1.10×10^{-3}	5.03×10^{-4}	$\mathbf{4.12\times10^{-4}}$	6.80×10^{-4}

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Выводы: достоинства и недостатки KAN

Достоинства

- сложность меньше, чем у MLP
- сходимость быстрее, чем у MLP
- интерпретируемость лучше, чем у MLP
- возможность управлять процессом построения модели
- достойная альтернатива символьной регрессии

Недостатки

- обучение $\times 10$ медленнее MLP с тем же числом весов
- чувствительность к шуму в данных
- (пока) не хватает альтернативных реализаций

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Yuntian Hou et al. A comprehensive survey on Kolmogorov Arnold Networks. 2024/07/13

Haoran Shen et al. Reduced Effectiveness of Kolmogorov–Arnold Networks on Functions with Noise. 2024/07/20

Развитие и обобщения

- другие виды сплайнов
  (вейвлеты, фурье, полиномы, радиальные, ядра и др.)
- оптимизация точек для построения сплайнов
- разные узлы могли бы использовать общий сплайн
- адаптация регуляризации, dropout, пакетной нормировки

Ziming Liu et al. KAN: Kolmogorov–Arnold Networks. 2024/04/30

Yuntian Hou et al. A comprehensive survey on Kolmogorov Arnold Networks. 2024/07/13

J.Cheon. Improving Computational Efficiency in Convolutional Kolmogorov-Arnold Networks // Neural Computing and Applications, 2024, 37(1), 15–30.

Qi Qiu et al. ReLU-KAN: New Kolmogorov-Arnold Networks that Only Need Matrix Addition, Dot Multiplication, and ReLU. 2024/06/04

S.S.Sidharth et al. Chebyshev Polynomial-Based Kolmogorov–Arnold Networks: An Efficient Architecture for Nonlinear Function Approximation. 2024/06/14

Hoang-Thang Ta. BSRBF-KAN: A Combination of B-Splines and Radial Basis Functions in Kolmogorov-Arnold Networks. 2024/06/21

Применения

- решение дифф. уравнений в частных производных
- многозадачное обучение (continual multitask learning) без катастрофического забывания ранее выученных задач — благодаря локальности сплайнов
- вывод моделей физических явлений по данным
- вывод моделей в виде неявных формул $f(x_1, \ldots, x_n) = 0$
- анализ временных рядов
- анализ данных на графе, графовые нейронные сети
- свёрточные нейронные сети для анализа изображений
- анализ гипер-спектральных изображений

Yuntian Hou et al. A comprehensive survey on Kolmogorov Arnold Networks. 2024/07/13
A.Vaca-Rubio et al. Kolmogorov-Arnold Networks in Quantum Architecture Search and Satellite Traffic Prediction. 2024.

Нарождающаяся мифология

- KAN вот-вот похоронят MLP
  нет: хотя потеснить кое-где могут (но это не точно)
- KAN это не MLP
  нет: KAN это вариант MLP с удвоенным числом слоёв
- в KAN нет весов, обучаются функции активации
  нет: обучаемые веса находятся внутри этих функций
- в узлах KAN находятся классические сплайны
  нет: классический сплайн строится по выборке (x_i, y_i),
  в KAN сплайны по отдельности не аппроксимируют y_i
- символьная регрессия не позволяет выбирать структуры
  нет: есть методы SR с полу-автоматическим управлением

R.G.Neychev et al. Optimal spanning tree reconstruction in symbolic regression. 2024.

M.Potanin et al. Additive regularization schedule for neural architecture search. 2024.

G.I.Rudoy, V.V.Strijov. Algorithms for inductive generation of superpositions for approximation of experimental data. 2013