

Обзор

Александров Кирилл

12 августа 2024 г.

KAN: Kolmogorov–Arnold Networks.

1 Мотивация

В основе полносвязных нейронных сетей (MLP) лежит универсальная теорема приближения (UAT). Авторы статьи решили пойти другим путем и оттолкнуться от теоремы Колмагорова-Арнольда о представимости многомерных функций.

Теорема 1.1 (*Колмогоров-Арнольд*)

Каждая многомерная непрерывная функция может быть представлена в виде суперпозиции непрерывных функций одной переменной.

Опираясь на Теорему (1.1), была представлена новая архитектура нейронных сетей KAN: Kolmogorov–Arnold Networks.

2 Основные идеи

1. Функции активации ставятся обучаемыми
2. Для работы с функциями активации используется метод сплайнов[1]

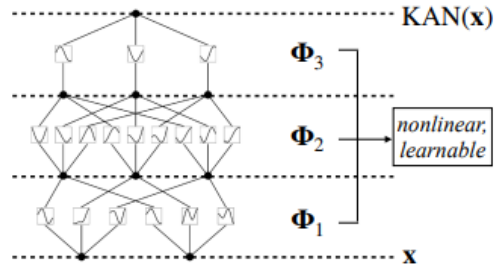


Рис. 1: Модель нейронной сети

3. Не используется типичная для (MLP) матрица весов. Вместо нее матрица функций активации.
4. Нейрон получает на вход сумму из импульсов, предназначенных только для него.
5. Преобразования между слоями обучаемые и нелинейные.

3 Реализация

1. Функции активации представляют собой непрерывные дифференцируемые выражения.

$$\phi(x) = \omega_1 b(x) + \omega_2 \text{spline}(x) \quad (1)$$

где $b(x)$ - это особая *базисная функция*, которая нужна для оптимизации работы. Примером такой функции может послужить:

$$b(x) = x/1 + e^{-x} = \text{silu}(x) \quad (2)$$

второе слагаемое выражения (1) представляет собой:

$$\text{spline}(x) = \sum_i c_i B_i(x) \quad (3)$$

$B_i(x)$ - так называемые *В-сплайны*[1], c_i - обучаемые параметры.

2. Рассмотрим нашу нейронную сеть со следующими параметрами:

- L - глубина
- n_i - ширина i -го слоя
- $[n_1, n_2, \dots, n_L]$ - массив с параметрами сети

Коротко сеть записывается как - $\text{KAN}[n_1, n_2, \dots, n_L]$. Так, например, сеть $\text{KAN}[1]$ - это простая интерполяция.

3. Определим i -тый нейрон l -го слоя как (l, i) . $x_{l,i}$ - величиной активации этого нейрона. Между слоями l и $l+1$ есть $n_l n_{l+1}$ функций активации. Функцией активации, которая соединяет (l, i) и $(l+1, j)$ будем обозначать:

$$\phi_{l,j,i}, \quad l = 0, \dots, L-1, \quad i = 1, \dots, n_l, \quad j = 1, \dots, n_{l+1}. \quad (4)$$

Преактивация $\phi_{l,j,i}$ - это $x_{l,i}$; постакивание $\phi_{l,j,i}$ определяется как $\tilde{x}_{l,j,i} \equiv \phi_{l,j,i}(x_{l,i})$. Величина активации нейрона $(l+1, j)$ сумма всех приходящих постакиваний.

$$x_{l+1,j} = \sum_{i=1}^{n_l} \tilde{x}_{l,j,i} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), \quad j = 1, \dots, n_{l+1}. \quad (5)$$

Матричный вид:

$$x_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix}}_{\Phi_l} x_l, \quad (6)$$

Тогда полную работу Нейронной сети можно описать следующим выражением:

$$\text{KAN}(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)x. \quad (7)$$

4 Обоснование работы данного подхода

Исходя из предыдущего пункта, можно понять то, что преобразования исходных данных дифференцируемы и непрерывны, поэтому обучить систему можно уже привычным нам градиентным спуском.

Также авторы статьи доказали теорему, которая показывает, что эта архитектура способна довольно хорошо приближать исходные данные

Теорема 4.1 (*Approximation theory, KAT*)

Пусть $x = (x_1, x_2, \dots, x_n)$. Предположим что функция $f(x)$ представима седующим образом

$$f = (\Phi_{L-1} \circ \Phi_{L-2} \circ \cdots \circ \Phi_1 \circ \Phi_0)x, \quad (8)$$

тогда существует константа C зависящая от f и ее вида и существуют G - размер сетки и k - порядок B -spline для $\Phi_{l,i,j}^G$ такие, что для любого m $0 \leq m \leq k$ выполнено

$$\|f - (\Phi_{L-1}^G \circ \Phi_{L-2}^G \circ \cdots \circ \Phi_1^G \circ \Phi_0^G)x\|_{C^m} \leq CG^{-k-1+m}. \quad (9)$$

где

$$\|g\|_{C^m} = \max_{|\beta| \leq m} \sup_{x \in [0,1]^n} |D^\beta g(x)|.$$

Про размер сетки и порядок *сплайна* можно почитать [1]

5 Сравнение KAN с MPL

Из-за того что мы избавились от линейной матрицы весов, взяв вместо нее матрицы функций активаций, то полученные нами данные входе работы алгоритма могут быть интерпретируемы человеком. Также *КАНЫ* быстро и легко переобучать, так как можно просто добавить дополнительный нейрон в слой зафиксировав уже обученные функции активации. Основным минусом является количество обучаемых параметров, сеть обучается медленнее, но требует меньше слоев что иногда компенсирует проблему времени.

6 Заключение

В данном обзоре я описал лишь принцип устройства новой архитектуры, лишь слегка затронув тему ее точности и ее сравнения с MLP. Гораздо подробнее про тесты и применение KAN можно почитать в оригинальной статье [0].

7 Ссылки

- [0] <https://arxiv.org/abs/2404.19756>
- [1] <https://www.brnt.eu/phd/node11>