

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по учебной практике
Тема: Нейронные сети Колмогорова-Арнольда

Студент гр. 2383

Исаев Э.Н.

Руководитель

Медведев А.Н.

Санкт-Петербург

2024

ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Исаев Э.Н.

Группа 2383

Тема практики: Нейронные сети Колмогорова-Арнольда

Задание на практику:

Изучение архитектуры нейронных сетей Колмогорова-Арнольда, а также применение знаний на практической задаче классификации. Сравнение полученных результатов с результатами работы классической нейронной сети — MLP (Multilayer Perceptron).

Сроки прохождения практики: 26.06.2024 — 09.07.2024

Дата сдачи отчета: 07.07.2024

Дата защиты отчета: 07.07.2024

Студент

Исаев Э.Н.

Руководитель

Медведев А.Н.

АННОТАЦИЯ

Целью данной практической работы является изучение механизмов работы нейронных сетей, основанных на теореме Колмогорова-Арнольда. Необходимо протестировать теоретический материал, используя для этого не массивные данные, наглядно демонстрирующие работу нейронной сети. После чего важно задействовать полученные навыки в решении задачи классификации на реальном датасете, а также сравнить с результатами решения той же задачи, но при использовании нейронной сети классического типа — MLP (Multilayer Perceptron). Сравнивая результаты, необходимо сделать вывод о работе KAN и ее эффективности.

СОДЕРЖАНИЕ

	Введение	5
1.	Различия MLP и KAN	6
1.1.	Теорема представления мультивариативной функции Колмогорова-Арнольда	6
1.2.	Сравнение производительности MLP и KAN	7
2.	Нейронные сети Колмогорова-Арнольда (KAN)	9
2.1.	Архитектура KAN	9
2.2.	Оценка параметров KAN	11
3.	Задача классификации	13
3.1.	Инструментарий	13
3.2.	Решение задачи на тренировочных данных	13
3.3.	Решение задачи на реальных данных	14
	Заключение	17
	Список использованных источников	18

ВВЕДЕНИЕ

Нейронные сети Колмогорова-Арнольда (KAN) являются многообещающей альтернативой многослойным перцептронам (MLP). В то время как MLP имеют фиксированные функции активации на нейронах, KAN имеют обучаемые функции активации на ребрах. KAN вообще не имеют линейных весов – каждый весовой параметр заменяется одномерной функцией, параметризованной в виде сплайна. Такое, казалось бы, простое изменение позволяет KAN превосходить MLP с точки зрения точности и интерпретируемости в небольших задачах, связанных с искусственным интеллектом и наукой. С точки зрения точности, небольшие KAN могут в задачах подбора функций достигать сопоставимой или более высокой точности, чем более крупные MLP. Так же для удобства взаимодействия KAN могут быть интуитивно визуализированы; библиотека `рукан`, реализующая KAN, предоставляет интерфейс для удобной визуализации графа вычислений. Эти возможности так же были рассмотрены в процессе работы и представлены ниже.

1. РАЗЛИЧИЯ MLP И KAN

1.1. Теорема представления мультивариативной функции

Колмогорова-Арнольда

Классическая архитектура нейронной сети MLP (Multi-Layer Perceptron) основана на универсальной теореме аппроксимации Цыбенко. Она утверждает, что нейронная сеть прямой связи с одним скрытым слоем может аппроксимировать любую непрерывную функцию многих переменных с любой точностью. Новейшая архитектура KAN (Kolmogorov-Arnold Network) основана на одноименной теореме Колмогорова-Арнольда. Согласно этой теореме, если f — n -мерная непрерывная функция, заданная на ограниченной области $(f: [0,1]^n \rightarrow \mathbb{R})$, то f можно записать в виде конечной композиции непрерывных функций одной переменной и операций сложения.

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

где $\phi_{q,p}: [0,1] \rightarrow \mathbb{R}$ и $\Phi_q: \mathbb{R} \rightarrow \mathbb{R}$.

Другими словами, единственная функция нескольких переменных — это сумма, т.к. все остальные функции многих переменных могут быть выражены через сумму функции одной переменной. Но для задач машинного обучения всё осложняется тем, что функции $\phi_{q,p}$ могут быть не гладкими и даже фрактальными. Из-за этого факта считалось, что данная теорема неприменима в области машинного обучения. Более того, ранние исследования этого вопроса останавливались на двуслойном представлении с $2n+1$ нейронами (в первоначальной формулировке теоремы есть только 2 слоя нелинейности), которое не позволяло применять большинство современных методов обучения нейронных сетей.

Однако, была выпущена архитектура, которая отошла от первоначальной формулировки теоремы (формулировки, допускающей только два слоя

нелинейности и малое количество слагаемых ($2n+1$) в скрытом слое). Она обобщает нейронную сеть до произвольного количества слоев и нейронов на слой.

1.2. Сравнение производительности MLP и KAN

В то время как в MLP установлены фиксированные функции активации на самих нейронах, в архитектуре KAN обучаемые функции активации располагаются на самих ребрах сети.

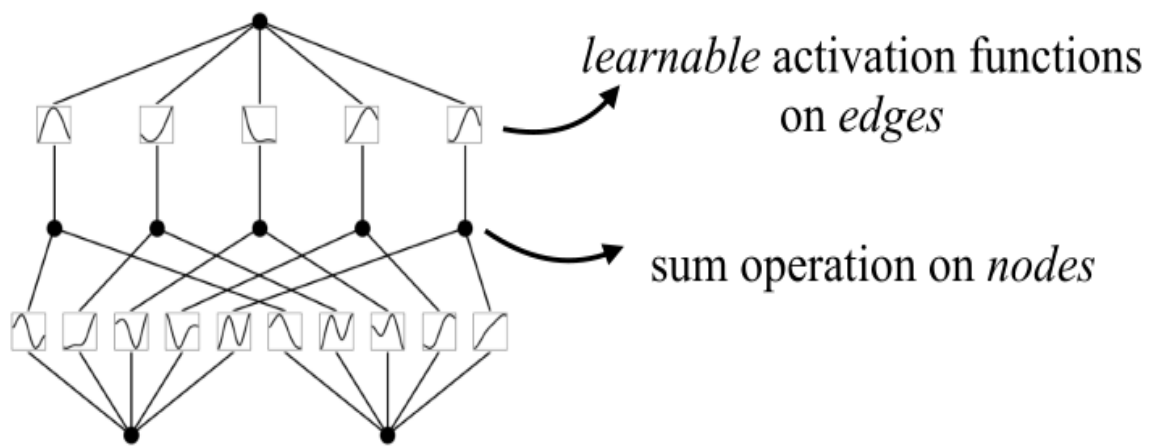


Рисунок 1.1 - KAN

В результате, в архитектуре KAN на смену веса приходит функция одной переменной, которая параметризована сплайном (кривой). А нейроны просто суммируют все входные значения (без применения нелинейности, как в MLP). Сами по себе сплайны плохо выделяют признаки (что является причиной проклятия размерности). В свою очередь, MLP хорошо выделяют признаки. Поэтому по внешней структуре KAN-ы повторяют MLP (многослойный перцептрон) и содержат сплайны во внутренней структуре (обучаемые коэффициенты в базисе В-сплайна). Таким образом, нейронные сети Колмогорова-Арнольда не только могут подмечать признаки, но и выделять закономерности с хорошей точностью (благодаря сплайнам).

Например, возьмем задачу регрессии и функцию многих переменных:

$$f(x_1, \dots, x_n) = \exp\left(\frac{1}{N} \sum_{i=1}^N \sin^2\left(\frac{\pi x_i}{2}\right)\right)$$

Сплаины провалили бы задачу для больших N из-за проклятия размерности; многослойный перцептрон возможно и выделил бы суммирование в изначальной формуле, однако функции активации (RELU или сигмоида) с крайне низкой эффективностью способны приближать экспоненту и синусы. KAN-ы могут выделить и сумму, и элементарные функции, входящие в состав f (засчет сплайнов), набирая большой отрыв в эффективности, по сравнению с MLP.

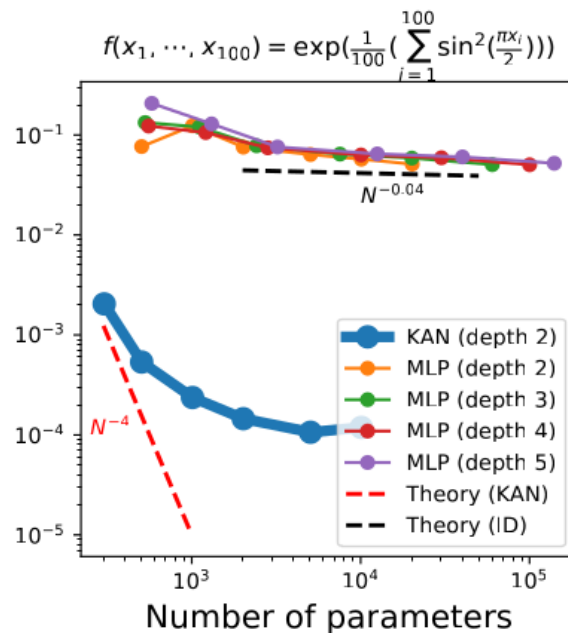


Рисунок 1.2 — сравнение MLP и KAN

Количество параметров (по горизонтали) и значение RMSE (по вертикали). На практике видно, что KAN с 2 слоями на несколько порядков превосходит MLP с 2+ слоями в предсказании значений функций.

2. НЕЙРОННЫЕ СЕТИ КОЛМОГорова-АРНОЛЬДА (KAN)

2.1. Архитектура KAN

Пусть перед нами стоит задача обучения с учителем. В качестве входных данных даются точки $\{x_i, y_i\}$, где мы хотим найти такую функцию f , что $y_i \approx f(x_i)$. Согласно теореме Колмагорова-Арнольда, задача решена, если мы найдем функции одной переменной $\phi_{q,p}$ и Φ_q . В KAN эти функции одной переменной параметризуются B-сплайнами, с обучаемыми коэффициентами у базисных сплайнов.

Как было сказано ранее, нейронная сеть, которая прямиком следует из теоремы, слишком простая и не способна аппроксимировать любую функцию с произвольной точностью. Данный недостаток обходится обобщением архитектуры под любой масштаб. На первый взгляд это обобщение может показаться неочевидным, т.к. представление Колмогорова-Арнольда отвечает за двуслойный KAN.

В обобщении нам поможет классическая архитектура перцептрона. В MLP определив один слой (композиция линейных преобразований и нелинейной функции активации), мы можем данные слои комбинировать. Что есть слой в KAN?

Слой в KAN, с входными параметрами размерности n_{inp} и выходными параметрами размерности n_{out} , может быть представлен, как матрица одномерных функций $\Phi = \{\phi_{q,p}\}, p=1,2,\dots,n_{inp}; q=1,2,\dots,n_{out}$, где $\phi_{q,p}$ имеет обучаемые параметры. В теореме Колмогорова-Арнольда внутренние функции образуют слой с $n_{inp}=n$ и $n_{out}=2n+1$, и внешние функции образуют слой с $n_{inp}=2n+1$ и $n_{out}=1$.

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

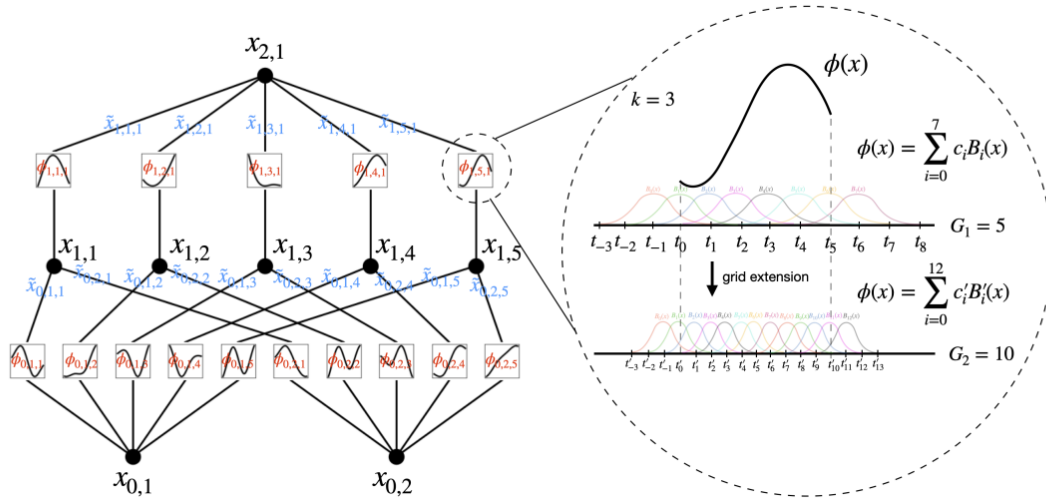


Рис 2.1 — Функции активации, параметризуемые B-сплайном

Теперь стало ясным то, что для обобщения KAN достаточно просто комбинировать эти слои вместе.

Введем обозначения. Форму KAN можно представить целочисленным массивом $[n_0, n_1, \dots, n_L]$, где n_i — это количество нейронов в i -ом слое сети. Нотация для i -го нейрона на l -ом слое — (l, i) и значение активации этого нейрона, обозначим $x_{l,i}$. Между слоем l и слоем $l+1$ находится $n_l n_{l+1}$ функций активации. Функция активации, которая соединяет (l, i) и $(l+1, j)$ обозначается как

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

Таким образом, входное значение для $\phi_{l,j,i}$ — это просто $x_{l,i}$. Результат $\phi_{l,j,i}$ обозначим, как $x_{l,j,i} \equiv \phi_{l,j,i}(x_{l,i})$. А значением на самом нейроне является следующая сумма:

$$x_{l+1,j} = \sum_{i=1}^{n_l} x_{l,j,i} = \sum_{i=1}^{n_l} \phi_{l,j,i}(x_{l,i}), i=1, \dots, n_{l+1}$$

В матричной форме получаем следующее:

$$\mathbf{x}_{l+1} = \underbrace{\begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \cdots & \phi_{l,1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \cdots & \phi_{l,2,n_l}(\cdot) \\ \vdots & \vdots & & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \cdots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix}}_{\Phi_l} \mathbf{x}_l,$$

Рис 2.2 — Матричная форма

Где Φ_l - матрица функций, которая соответствует l-му слою в KAN. В общем случае KAN - композиция L слоев: дан входной вектор $x_0 \in R^{n_0}$, тогда выход нашей модели

$$KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x$$

2.2. Оценка параметров KAN

В введенной нотации теорема Колмогорова-Арнольда представляет двухслойный KAN с формой $[n, 2n+1, 1]$. Все операции в KAN являются дифференцируемыми (включая В-сплайны, которые C2 непрерывны), поэтому мы можем использовать обратное распространение ошибки для обучения сети.

Однако не тривиальной задачей является проектирование модели таким образом, чтобы она была хороша в обучении. Основные идеи, которые помогают это сделать:

- (1) Остаточные функции активации. Мы вводим функцию

$$b(x): \phi(x) = w_b b(x) + w_s spline(x) . \text{ Функция } b(x) = \frac{x}{1+e^{-x}} \text{ (так же известную,}$$

как silu(x) - Sigmoid linear unit, часто используется вместо RELU(x) в

MLP). А $spline(x)$ - линейная комбинация В-сплайнов:

$$spline(x) = \sum_i c_i B_i(x) , \text{ где } c_i \text{ обучаемые. } w_b \text{ и } w_s \text{ так же являются}$$

обучаемыми.

- (2) Инициализация множителей. Во время инициализации значения $w_s=1$ и $spline(x) \approx 0$ (базисные коэффициенты сплайна инициализируются значениями из нормального распределения $\mu=0$, σ^2 , где $\sigma=0.1$). А w_b инициализируется согласно инициализации Хавьера (как и в случае инициализации весов в MLP).
- (3) Обновление сетки сплайнов. Каждую сетку мы обновляем в реальном времени в зависимости от входных значений активаций. Это необходимо для того, чтобы сплайны не были ограничены на заданной области. Ведь сами значения активации могут выйти из такой области во время обучения.

Оценим количество параметров. Для простоты пусть у нас есть сеть:

1. Глубины L
2. На каждом слое одинаковое количество нейронов $n_0=n_1=\dots=n_L=N$
3. Каждый сплайн порядка k , определенный на G интервалах

Тогда в сумме получается, что $O(N^2L(G+k)) \sim O(N^2LG)$ параметров. Отметим, что MLP в тех же условиях требует $O(N^2L)$ параметров. Но обычно KAN требуют гораздо меньше N , чем MLP.

3. ЗАДАЧА КЛАССИФИКАЦИИ

3.1. Инструментарий

Для разработки и последующего обучения нейронной сети Колмогорова-Арнольда на практике были изучена и впоследствии использована библиотека `kan`. Данная библиотека позволяет создавать KAN с пользовательским указанием размерности слоев и обучать модель на данном датасете, при этом само обучение также имеет настройку: от количество шагов обучения и выбора метрики оценивания до выбора функции потерь.

Также важной особенностью пакета `kan` является возможность манипулировать сетью на уровне архитектуры: удалять и добавлять нейроны и ребра активации, что позволяет тонко настраивать всю модель в целом, и облегчать модель при помощи метода `prune()`.

Во взаимодействии также предусмотрена замена сплайнов на символьные функции активации. Более того, интерфейс позволяет визуализировать полученные модели, что безусловно придает наглядность в работе с нейронной сетью.

В ходе работы также был освоен и использован модуль библиотеки `sklearn` — `neural_network`. В данном модуле реализован класс `MLPClassifier`, реализующий алгоритм многослойного перцептрона для классификатора. При инициализации нейронной сети можно также привести свои настройки, однако сам класс фиксирует настройки функции активации как `RELU` и метод оптимизации целевой функции как `Adam`.

3.2. Решение задачи на тренировочных данных

Необходимо воспользоваться изученным теоретическим материалом для решения конкретных задач. Для начала решим задачу классификации на датасете ирисов Фишера, для этого был выбран расширенный датасет, отличающийся размером выборки и размерностью пространства признаков.

Ирисы Фишера обладают пятью определяющими параметрами: 'sepal_length', 'sepal_width', 'petal_length', 'petal_width', 'elevation' — они же и будут являться входными для сети KAN.

Расширенный датасет включает в себя набор из 1200 объектов. Обучение модели производилось на 80% выборки. Сеть имеет 3 слоя: [5, 11, 3], где 5 — размерность пространства признаков, 3 — количество классов, 11 — по изначальной теореме Колмогорова-Арнольда размерность второго слоя KAN = $2 * 5 + 1$. При обучении были получены значения loss: train loss: 2.84e-01 | test loss: 2.51e-01, что вполне приемливо для данного датасета.

Обученная модель устанавливает соответствие классам с точностью 97%. По графикам ниже наглядно видно соответствие результата классификации исходным данным.

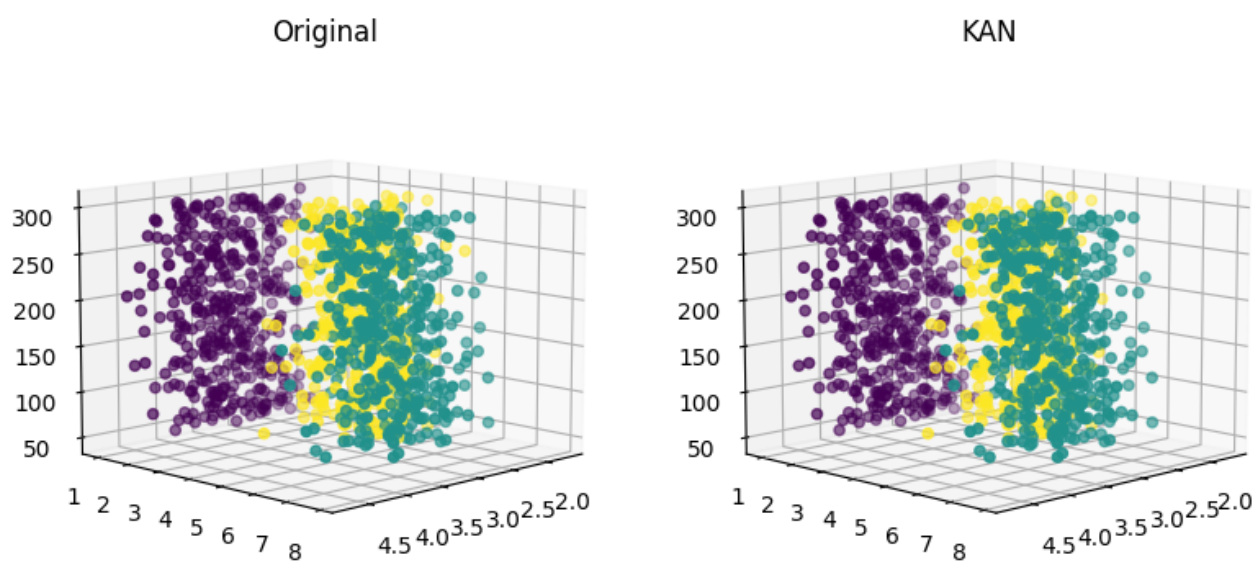


Рис 3.1 — Исходный датасет и результат классификации KAN

Также в результате обучения была получена сама функция, действующая из пятимерного пространства признаков в трехмерное пространство, определяющая класс объекта (индекс наибольшего значения вектора есть результат классификации). Однако, данная функция не представляет интерес в задаче классификации, в отличие от задачи регрессии.

3.3. Решение задачи на реальных данных

Для демонстрации решения классификации на реальных данных был выбран датасет с платформы Kaggle — Purchase Behavior Dataset.

Этот набор данных содержит информацию о покупательском поведении клиентов по различным признакам, чтобы помочь специалистам по обработке данных и аналитикам понять факторы, влияющие на принятие решений о покупке. Набор данных включает демографическую информацию, покупательские привычки и другие важные характеристики.

Признаки:

- Age: Возраст клиента
- Gender: Пол клиента (0: Мужчина, 1: Женщина)
- Annual Income: Годовой доход клиента в долларах США
- Number of Purchases: Общее количество покупок, совершенных клиентом
- Product Category: Категория приобретаемого товара (0: Электроника, 1: Одежда, 2: Товары для дома, 3: Красота, 4: Спорт)
- Time Spent on Website: Время, проведенное клиентом на сайте в минутах.
- Loyalty Program:: Является ли клиент участником программы лояльности (0: Нет, 1: Да)
- Discounts Availed: Количество скидок, которыми пользуется клиент (диапазон: 0-5)
- Purchase Status (целевая переменная): Вероятность совершения покупателем покупки (0: Нет, 1: Да)

Перед тем как приступить к обучению модели и самому решению задачи классификации необходимо предварительно обработать данные. Для этого был использован пакет `sklearn.preprocessing`, содержащий несколько общих служебных функций и классов преобразователей для преобразования необработанных векторов признаков в представление, более подходящее для последующих моделей.

Стандартизация наборов данных является общим требованием для многих моделей машинного обучения; они могут вести себя плохо, если

отдельные признаки не будут более или менее похожи на стандартные нормально распределенные данные: гауссовы с нулевым средним и единичной дисперсией.

Модуль preprocessing предоставляет служебный класс StandardScaler, который представляет собой быстрый и простой способ выполнить стандартизации наборов данных.

Обучение модели производилось на 80% выборки. Сеть имеет 3 слоя: [8, 17, 2], где 8 — размерность пространства признаков, 2 — количество классов, 17 — по изначальной теореме Колмогорова-Арнольда размерность второго слоя KAN = $2 * 8 + 1$. При обучении были получены значения loss: ttrain loss: 8.42e-04 | test loss: 1.97e+01.

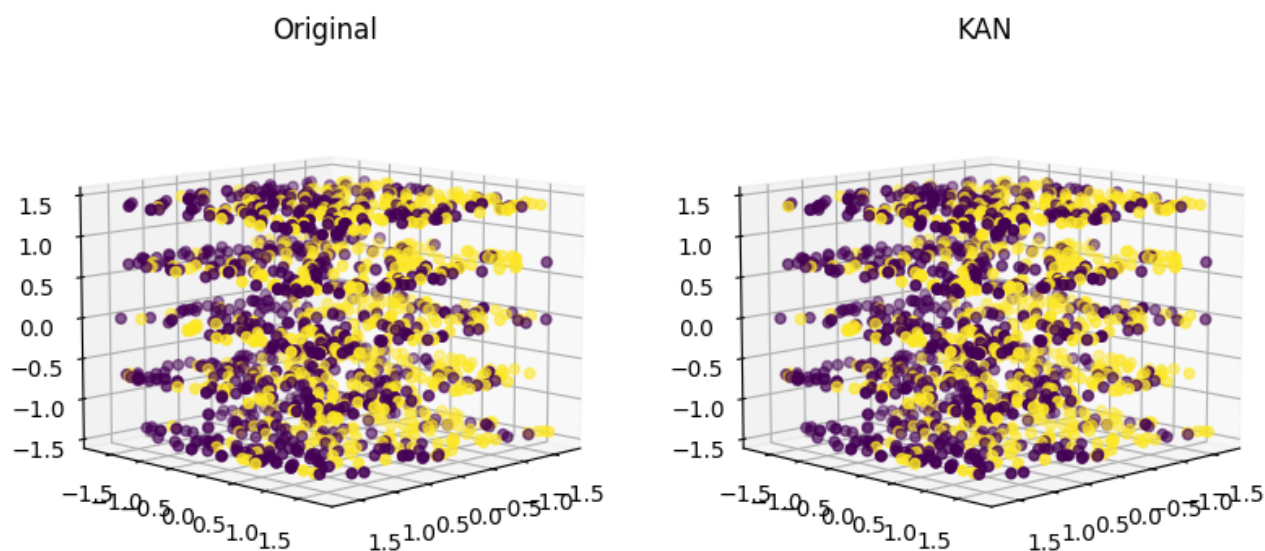


Рис 3.2 — Исходный реальный датасет и результат классификации KAN

Обученная модель устанавливает соответствие классам с точностью 97%. По графикам выше наглядно видно соответствие результата классификации исходным данным.

Также для данного датасета была решена задачи классификации при помощи классического многослойного перцептрона (MLP).

Класс MLPClassifier из scikit-learn реализует алгоритм многослойного перцептрона, который обучается с использованием обратного распространения .

MLP обучается на двух массивах: массив X размера $(n_samples, n_features)$, который содержит обучающие образцы, представленные как векторы признаков с плавающей запятой; и массив y размера $(n_samples,)$, который содержит целевые значения (метки классов) для обучающих выборок. После обучения модель может предсказывать метки для новых образцов.

Обученная модель устанавливает соответствие классам с точностью 89%, что также является высоким показателем, однако видно, что MLP сильно проигрывает модели KAN в точности предсказания задачи классификации (89% против 97%).

ЗАКЛЮЧЕНИЕ

В результате была изучена архитектура нейронных сетей, основанных на теореме Колмогорова-Арнольда — KAN. Нейросети данного типа представляют наибольший интерес в научном сообществе за счет нового подхода в архитектуре сети, а также, учитывая высокую точность и эффективность KAN по сравнению с традиционными многослойными перцептронами (MLP), что было продемонстрировано экспериментальным путем, перспективности дальнейших исследований и разработок в этой области.

Так, была решена задача классификации сначала на тренировочных, а после и на реальных данных. В ходе решения были выявлены очевидные преимущества KAN в сравнении с MLP, заключающиеся в точности предсказания классов и, что примечательно, при использовании меньшего количества нейронов. Однако у архитектуры есть бутылочное горлышко: KAN учится медленнее MLP, что становится серьезным препятствием при его использовании.

В целом, результаты работы подтверждают значительный потенциал нейронных сетей на основе теоремы Колмогорова-Арнольда для применения в различных областях машинного обучения. Они открывают новые возможности для развития методов классификации и предсказания, обещая улучшить точность и эффективность моделей машинного обучения в будущем.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Haykin S. Neural networks: a comprehensive foundation. – Prentice Hall PTR, 1998.
2. Kolmogorov A. N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition //Doklady Akademii Nauk. – Russian Academy of Sciences, 1957. – Т. 114. – №. 5. – С. 953-956.
3. Sprecher D. A., Draghici S. Space-filling curves and Kolmogorov superposition-based neural networks //Neural Networks. – 2002. – Т. 15. – №. 1. – С. 57-67.
4. Mario Köppen. On the training of a kolmogorov network. In Artificial Neural Networks—ICANN 2002: International Conference Madrid, Spain, August 28–30, 2002 Proceedings 12, pages 474–479. Springer, 2002.
5. Ji-Nan Lin and Rolf Unbehauen. On the realization of a kolmogorov network. Neural Computation, 5(1):18–20, 1993.
6. Lai M. J., Shen Z. The kolmogorov superposition theorem can break the curse of dimensionality when approximating high dimensional functions //arXiv preprint arXiv:2112.09963. – 2021.
7. Leni P. E., Fougerolle Y. D., Truchetet F. The kolmogorov spline network for image processing //Image Processing: Concepts, Methodologies, Tools, and Applications. – IGI Global, 2013. – С. 54-78.
8. Fakhoury D., Fakhoury E., Speleers H. ExSpliNet: An interpretable and expressive spline-based neural network //Neural Networks. – 2022. – Т. 152. – С. 332-346.
9. Montanelli H., Yang H. Error bounds for deep ReLU networks using the Kolmogorov–Arnold superposition theorem //Neural Networks. – 2020. – Т. 129. – С. 1-6.

10. He J. On the optimal expressive power of relu dnns and its application in approximation with kolmogorov superposition theorem //arXiv preprint arXiv:2308.05509. – 2023.
11. He J. et al. ReLU deep neural networks and linear finite elements //arXiv preprint arXiv:1807.03973. – 2018.
12. He J., Xu J. Deep neural networks and finite elements of any order on arbitrary dimensions //arXiv preprint arXiv:2312.14276. – 2023.
13. Poggio T., Banburski A., Liao Q. Theoretical issues in deep networks //Proceedings of the National Academy of Sciences. – 2020. – T. 117. – №. 48. – C. 30039-30045.
14. Lin H. W., Tegmark M., Rolnick D. Why does deep and cheap learning work so well? //Journal of Statistical Physics. – 2017. – T. 168. – C. 1223-1247.
15. Liu Z. et al. Kan: Kolmogorov-arnold networks //arXiv preprint arXiv:2404.19756. – 2024.