

УДК 378:004

DOI: 10.25686/978-5-8158-2474-4-2025-790-810

Сети KAN и MLP в био- и медицинской диагностике на примере стресса растений

А. И. Ларионов, В. Е. Турлапов

Центр искусственного интеллекта, Университет им. Н.И. Лобачевского, Нижний Новгород, Россия

Аннотация. Потери урожая, связанные с болезнями растений, остаются актуальной проблемой современного сельского хозяйства. Раннее детектирование – ключевой фактор в сокращении ежегодных потерь урожая, которые по различным позициям могут достигать 40 %. Выполнено сравнительное исследование эффективности раннего обнаружения стресса растений как на различных открытых датасетах, так и с помощью недавно появившихся нейронных сетей Колмогорова–Арнольда на фоне многослойного перцептрона (MLP) с аналогичным числом обучаемых параметров. Также проведено сравнение обычной сверточной нейронной сети с сетью, сверточные слои которой используют обучаемые функции активации. KAN-сеть показала хорошие результаты по отношению к MLP как на задаче регрессии математической функции, так и на задачах бинарной и мультиклассовой классификаций болезней растений, относительно количества обучаемых параметров. Исследованию предшествует обзор современных модификаций KAN и их применения в медицине и биологии.

Ключевые слова: искусственный интеллект, сеть Колмогорова–Арнольда, стресс растений, обучаемые функции активации, объяснимый ИИ, машинное обучение, глубокое обучение, компьютерное зрение

KAN and MLP networks in bio- and medicine diagnostics on the example of plant stress

Larionov Aleksei Igorevich, Turlapov Vadim Evgenievich

Center for Artificial Intelligence, N.I. Lobachevsky University, Nizhny Novgorod, Russia

Abstract. Crop losses due to plant diseases remain a pressing challenge in modern agriculture. Early detection is a key factor in reducing annual yield losses, which in some cases can reach up to 40%. A comparative study of the efficiency of early detection of plant stress was performed, both on various open datasets and using recently emerged Kolmogorov-Arnold neural networks against the background of a multilayer perceptron (MLP) with a similar number of trainable parameters. A comparison was also made between a conventional convolutional neural network and a network whose convolutional layers use trainable activation functions. The KAN network showed good results in relation to MLP, both on the problem of regression of a mathematical function and on the problems of binary and multiclass classification of plant diseases, relative to the number of trainable parameters. The study is preceded by a review of modern modifications of KAN and their application in medicine and biology.

Keywords: artificial intelligence, Kolmogorov–Arnold network, plant stress, trainable activation functions, explainable AI, machine learning, deep learning, computer vision

Введение

В работе исследуется использование сетей Колмогорова–Арнольда (KANs) как перспективной альтернативы традиционным многослойным перцептронам (MLPs), которые в настоящее время широко применяются для решения различных задач. Сети Колмогорова–Арнольда предлагают новую архитектуру, основанную на теореме представления Колмогорова–Арнольда, которая позволяет обучать не веса, а сами функции активации. В этой работе представлены результаты применения различных вариантов KAN к теоретическим и прикладным задачам, сравнение эффективности неглубокой KAN, использующей кубические сплайны для обучаемых функций активации и классической однослойной нейронной сети с обучаемыми весами и фиксированной функцией активации ReLU, исследование объяснимости результатов обучения KAN.

Рассмотрена возможность применения различных сетей Колмогорова–Арнольда к прикладной задаче раннего детектирования болезней растений. Болезни представляют существенную угрозу мировому производству продуктов питания. Глобально от болезней растений и вредителей мировое сельское хозяйство ежегодно теряет от 16 до 40 % урожая. Несмотря на разработку новых методов защиты растений, потери все равно остаются ощутимыми – раннее обнаружение заболевания или стресса растения является ключевым фактором минимизации потерь урожая. Основными методами обнаружения болезней растений являются методы на основе анализа ДНК, технологии на основе

анализа измерений летучих выделений растений, визуальный анализ, в том числе с применением нейронных сетей.

Обзор публикаций

С момента публикации [1] в 2024 году, реализующей KAN, научное сообщество проявило явный интерес к новому полю для исследований и его возможностям. С тех пор вышло как минимум 60 публикаций. Среди них есть Kolmogorov-Arnold Networks Meet Science [2], в которой авторы делают упор на научное применение сетей KAN. Они рассматривают KAN как мост между современными принципами Artificial Intelligence (AI), основанными на коннекционизме, и символической наукой. В данной публикации был расширен функционал оригинальной библиотеки PyKAN такими функциями, как например: kanpiler – создает KAN из математической формулы, tree converter – преобразует модель в дерево, MultKAN – модель KAN с умножением вместо сложения на узлах. Также было внесено множество улучшений производительности по сравнению с первой публикацией библиотеки, множество функций, требующих больших затрат производительности или недоступных для параллелизации, стали опциональными, проведена оптимизация расчёта сплайнов – для сети глубины L , ширины слоя N с количеством интервалов G расход памяти был уменьшен с $O(LN^2G)$ до $O(LNG)$.

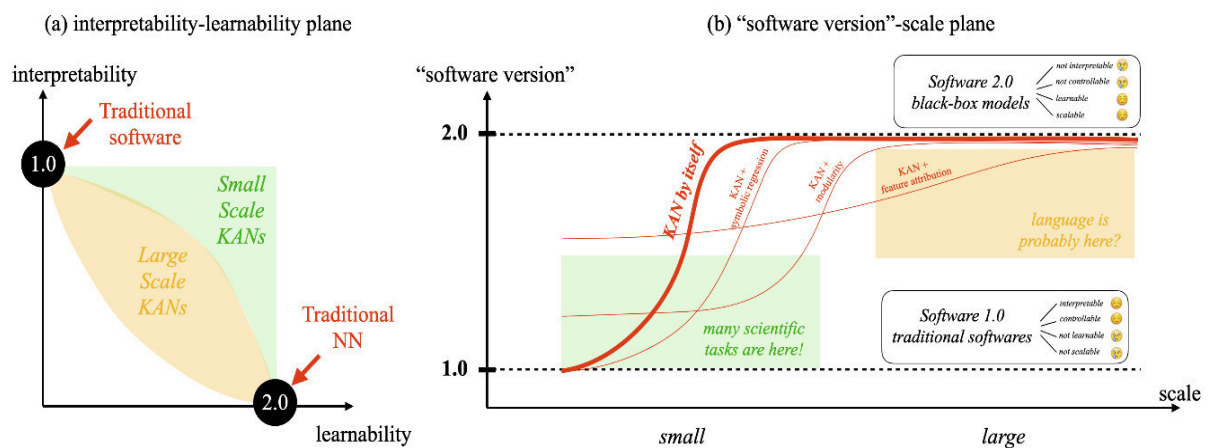


Рисунок 1 [2]

На рисунке 1 показано представление авторов о месте KAN на текущем ландшафте научных и производственных задач. KAN стремится занять промежуточную позицию между классическим программным обеспечением (software 1.0), ориентированным на интерпретируемость, и современным программным обеспечением (software 2.0), ориентированным на обучаемость, обеспечивая баланс между интерпретируемостью и способностью к обучению, что иллюстрируется их положением на фронте Парето в пространстве «интерпретируемость-масштабируемость». Количество интерпретаций, которое можно получить с помощью KAN, зависит от масштаба задачи и выбранных методов интерпретации. Также они предлагают KAN как универсальные базовые модели для физических исследований и демонстрируют их сильные стороны на исследовании с помощью KAN таких физических задач и концепций, как поиск сохраняющихся величин, поиск Лагранжианов на примере задачи маятника и релятивистской массы в однородном поле, поиск скрытой симметрии, на примере черной дыры Шварцшильда.

В публикации DropKAN [11] исследуется задача предотвращения ко-адаптации функций активации. Исследование показало, что разрежение сети KAN с помощью механизма dropout между слоями сети KAN (отключение узлов) может приводить к тому, что функции активации все равно будут испытывать эффект от обратного распространения ошибки. Метод DropKAN, основанный на случайном маскировании выходов либо после активаций (DropKANpa), либо только сплайновых компонентов (DropKANps) внутри слоя KAN, показал лучшие результаты по сравнению со стандартными dropout между слоями (табл. 1).

Таблица 1. Классификация на KAN с различными реализациями DROPOUT [11]

Dataset	KANs Accuracy				
	No-Drop	Dropout _{w/o scale}	Dropout _{w/scale}	DropKAN _{ps}	DropKAN _{pa}
dermatology	73.78%±5.94	89.46%±5.60	86.49%±1.66	92.43%±1.21	92.70%±1.54
german	66.70%±1.35	74.30%±2.02	73.40%±3.49	70.70%±3.03	76.6%±1.98
semeion	89.66%±0.0	94.55%±4.50	97.43%±0.52	99.81%±0.42	97.62%±0.42
car	91.56%±1.18	86.07%±1.60	77.69%±6.96	89.65%±0.63	85.66%±2.95
abalone	24.98%±0.95	23.71%±1.96	23.35%±1.85	27.68%±0.87	27.85%±0.49
adult	84.94%±0.17	84.88%±0.15	84.66%±0.53	85.27%±0.12	85.08%±0.19
bank-full	90.13%±0.37	90.06%±0.26	90.12%±0.20	90.28%±0.20	90.33%±0.13
connect-4	67.26%±4.31	70.53%±0.39	63.85%±5.77	72.14%±0.31	68.67%±1.72
diabetic	52.56%±4.44	56.06%±1.58	51.52%±2.80	57.95%±0.27	56.55%±0.76
census-income	94.78%±0.21	94.90%±0.08	94.11%±0.23	94.58%±0.11	94.69%±0.03

Kolmogorov-Arnold Transformer [12] рассматривает использование KAN слоев в архитектуре трансформеров вместо MLP, для глубокого обучения. Авторы публикации ускорили сети KAN путем использования рациональных функций, более оптимальных для расчёта на GPU, группировкой граней для оптимизации вычисления параметров и улучшили инициализацию весов для обеспечения стабильной дисперсии. Они сравнили производительность трех вариантов архитектуры – классический трансформер, трансформер, в котором MLP заменен на стандартную реализацию KAN, и трансформер, в котором MLP заменен на KAN с перечисленными выше улучшениями Group-Rational KAN (GR-KAN).

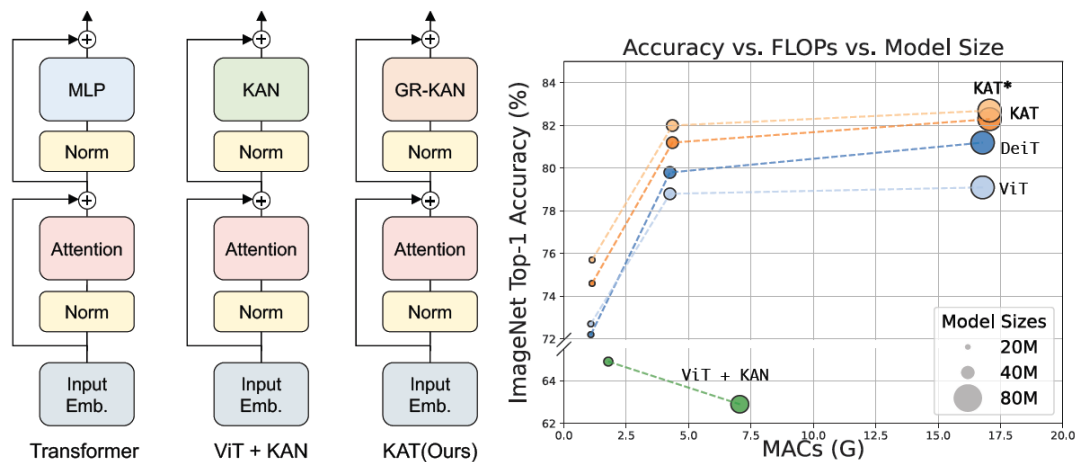


Рисунок 2. Сравнение результатов архитектур, предложенных на датасете ImageNet [12]

Версия трансформера, использующая GR-KAN, показала лучшие результаты в тестировании на нескольких задачах.

Kolmogorov-Arnold Convolutions [13] исследует применение методов KAN к задачам Computer Vision (CV), рассматривает такие замены для сплайнов, как вейвлеты и множество многочленов Грэма. Также были предложены реализации механизмов самовнимания и фокальной модуляции.

Как показано в таблице 2, в работе достигнуты передовые результаты в задачах сегментации с использованием архитектур, аналогичных U-Net, с KAN-свёртками на датасетах BUSI, GlaS и CVC.

Kolmogorov-Arnold Fourier Networks [14] представляет собой дальнейшее развитие идей интерпретируемых сетей KAN для решения задач с высоким числом признаков.

Таблица 2. Результаты эксперимента сегментации медицинских изображений [13]

Methods	BUSI [34]		GlaS [35]		CVC [36]	
	IoU↑	F1↑	IoU↑	F1↑	IoU↑	F1↑
U-Net [6]	57.22	71.91	86.66	92.79	83.79	91.06
Att-Unet [37]	55.18	70.22	86.84	92.89	84.52	91.46
U-Net++ [10]	57.41	72.11	87.07	92.96	84.61	91.53
U-NeXt [38]	59.06	73.08	84.51	91.55	74.83	85.36
Rolling-UNet [39]	61.00	74.67	86.42	92.63	82.87	90.48
U-Mamba [40]	61.81	75.55	87.01	93.0	84.79	91.63
U-KAN [13]	63.38	76.40	87.64	93.37	85.05	91.88
U ² -KAGNet, Small (Ours)	59.44	74.56	89.13	94.25	85.62	92.26
U ² -KAGNet, BottleNeck, Small (Ours)	62.95	77.26	85.99	92.47	87.61	93.39
U ² -KAGNet, BottleNeck (Ours)	55.91	71.72	87.73	93.47	88.86	94.1
U ² -KAGNet (Ours)	58.13	73.52	88.14	93.7	87.07	93.09
UKAGNet (Ours)	63.45	77.64	87.31	93.23	76.85	86.91

Авторы отмечают, что при всех теоретических преимуществах KAN возникает проблема взрывного роста числа параметров и сложности в захвате высокочастотных признаков в многомерных задачах. В работе предлагается интеграция обучаемых случайных Фурье-признаков (Random Fourier Features, RFF) и новой гибридной функции активации GELU-Fourier, что позволяет добиться баланса между компактностью параметров и спектральной выразительностью модели.

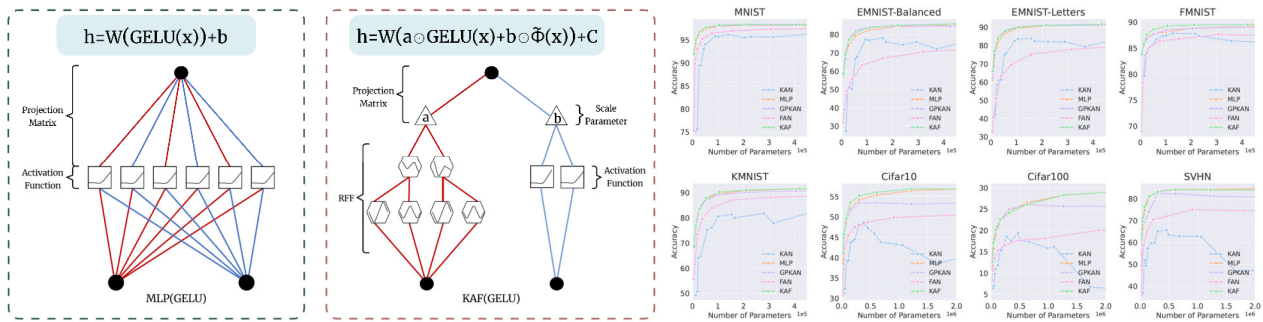


Рисунок 3 [14]

На рисунке 3 слева – стандартный MLP с активацией GELU и KAF с активацией GELU. MLP включает в себя проекционную матрицу, за которой следует GELU, в то время как KAF добавляет случайные признаки Фурье (RFF) и параметры масштабирования, предлагая большую гибкость в преобразованиях признаков.

Справа на рисунке 3 показано сравнение производительности различных моделей (KAN, MLP, GPKAN, FAN, KAF) в нескольких наборах данных (MNIST, EMNIST, FMNIST, KMNIST, Cifar10, Cifar100, SVHN). Результаты показывают, что KAF обычно достигает более высокой точности с меньшим количеством параметров

Wav-KAN [15] – реализация KAN сетей с функциями активаций на основе вейвлетов. Итоги тестирования авторами показали, что реализация функций активации с помощью вейвлета Difference of Gaussians (DOG) превосходит базовую реализацию KAN по точности. Также вейвлеты позволяют оптимизировать количество обучаемых параметров.

СохKAN [16] – это специализированная модификация сети KAN, предложенная для задач анализа выживаемости с акцентом на достижение баланса между производительностью и

интерпретируемостью. В отличие от классической модели пропорциональных рисков Кокса, СохKAN способен автоматически выявлять сложные взаимодействия признаков и формулировать аналитические (символические) выражения для функции риска, что повышает прозрачность результатов. Авторы показывают, что СохKAN успешно восстанавливает символические формулы и осуществляет автоматический отбор признаков на синтетических данных, а также демонстрирует лучшие или сопоставимые результаты с глубокими нейросетями и классическими моделями на реальных медицинских и геномных датасетах, что отражено в таблице 3.

Таблица 3. Результаты на клинических датасетах [16]

Dataset	CoxPH	DeepSurv	CoxKAN Trained	CoxKAN Pruned	CoxKAN Symbolic
SUPPORT	0.583074 (0.581, 0.585)	0.618308* (0.616, 0.620)	0.624482 (0.622, 0.625)	0.624485 (0.622, 0.625)	0.623755 (0.623, 0.626)
GBSG	0.656291 (0.655, 0.662)	0.668402* (0.665, 0.671)	0.678294 (0.676, 0.682)	0.679219 (0.675, 0.681)	0.682796 (0.678, 0.684)
METABRIC	0.632363 (0.628, 0.637)	0.643375* (0.639, 0.647)	0.647177 (0.644, 0.652)	0.648004 (0.646, 0.654)	0.649618 (0.644, 0.651)
FLCHAIN	0.797854 (0.797, 0.802)	0.794520 (0.793, 0.798)	0.797064 (0.796, 0.801)	0.795911 (0.792, 0.797)	0.796281 (0.795, 0.800)
NWTCO	0.698347 (0.693, 0.703)	0.698300 (0.692, 0.703)	0.719843 (0.714, 0.725)	0.720721 (0.708, 0.718)	0.722225 (0.715, 0.725)

Residual Kolmogorov-Arnold Network (RKAN) [17] – это архитектурное расширение для сверточных нейронных сетей, направленное на преодоление ограничений стандартных глубоких моделей, связанных с их линейной природой и фиксированными функциями активации. В RKAN используются иерархические KAN- слои на основе остаточных связей, что способствует увеличению эффективности обучения, выразительности моделей и стабильности, снижая риски затухающих и взрывающихся градиентов, особенно на небольших датасетах. Особое внимание уделено использованию многочленов Чебышёва вместо B-сплайнов, что позволяет повысить вычислительную эффективность и масштабируемость. В экспериментах на стандартных наборах данных (CIFAR-100, Food-101, ImageNet) RKAN стабильно улучшал точность топ-1 как минимум на 1 % относительно базовых моделей, при этом на больших архитектурах прирост был более заметным, что показано в таблице 4.

Таблица 4. Сравнение ТОП-1 точности (%) базовых моделей и моделей, дополненных RKAN на датасетах CIFAR-100, FOOD-101 [17]

	CIFAR-100			Food-101		
	res.	Rkan	base	res.	Rkan	base
ResNeXt-101	128	86.15	85.28	224	90.82	89.87
ResNeXt-50	128	85.08	84.40	224	90.00	89.20
ResNet-152	128	85.40	84.63	224	90.36	89.70
ResNet-101	128	85.12	84.00	224	90.09	89.29
ResNet-50	128	84.56	84.12	224	89.48	88.84
RegNetY-32GF	128	87.03	85.44	224	91.62	90.72
RegNetY-8GF	128	86.11	84.77	224	91.17	90.43
RegNetY-3.2GF	128	85.46	84.68	224	90.09	89.54
RegNetY-800MF	128	83.19	82.74	224	89.00	88.39
DenseNet-201	128	85.35	84.28	224	89.58	88.83
DenseNet-169	128	84.84	84.00	224	89.74	89.17
DenseNet-121	128	84.73	84.09	224	89.43	88.98

Модуль также демонстрирует лучшую стабильность (меньше коэффициент вариации), эффективное использование параметров и памяти, а также повышенную скорость прямого и обратного прохода по сравнению с другими подходами.

Construction of the Kolmogorov-Arnold representation using the Newton-Kaczmarz method [29] предлагает свою реализацию KAN на языке C++, ускоряющую обучение данного типа сетей. Одна из рассмотренных в публикации проблем – нахождение определителя квадратной матрицы размерности 5, она включала 10 миллионов записей и 25 параметров. В этой задаче обучение KAN заняло до 10 минут, тогда как обучение аналогичной модели многослойного перцептрона занимало от 4 до 8 часов, обучение останавливалось при достижении коэффициента Пирсона >0.9 на валидирующем датасете из 2 миллионов записей.

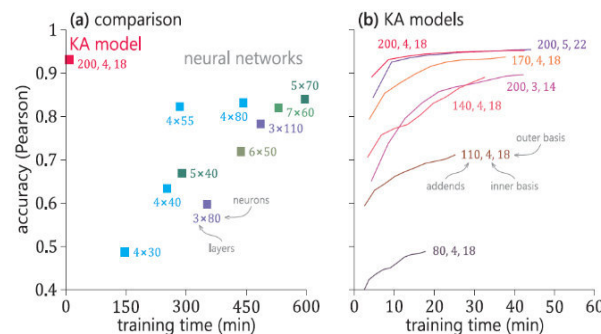


Рисунок 4 [29]

На рисунке 4а показано сравнение производительности сети Колмогорова-Арнольда (КА-model) и встроенных моделей MATLAB на задаче регрессии – поиска определителя для квадратной матрицы размерности 5. На рисунке 4б – эволюция точности различных КА-моделей. Такое ускорение достигается за счет использования метода Ньютона-Качмарца и его модификации для параллельного исполнения. Дальнейшее ускорение происходит за счет того, что при обучении не фиксируются базисные функции, большая часть проходов может использовать кусочно-линейные функции, а более сложные и выразительные базисные функции могут быть «включены» на поздних этапах обучения.

Материалы и методы

1. Теорема Колмогорова-Арнольда

Теорема Колмогорова-Арнольда о представлении непрерывных функций нескольких переменных в виде суперпозиции функций меньшего числа переменных математически формулируется как

$$f(x_1, \dots, x_d) = \sum_{q=0}^{2d} \Phi_q \left(\sum_{p=1}^d \Phi_{q,p}(x_p) \right), \quad (1)$$

где $f: [0,1]^d \rightarrow R$ – непрерывная функция d переменных; d – размерность входного пространства; Φ_q – непрерывные функции одной переменной; $\Phi_{q,p}$ – непрерывные функции одной переменной.

На основе данной теоремы в 2024 году была представлена работа Z. Liu с соавторами [1], в которой описан подход к построению нейронных сетей, использующих обучаемые функции активации на ребрах и операцию сложения на узлах. В этой работе такая архитектура сети предлагается как альтернатива классическим нейронным сетям, которые опираются на универсальную теорему аппроксимации.

Универсальная теорема аппроксимации (теорема Цыбенко):

$$G(x, w, \alpha, \theta) = \sum_{i=1}^N \alpha_i \varphi(w_i^T x + \theta_i), \quad (2)$$

где w_i – веса между входными нейронами и нейронами скрытого слоя,

α – веса между связями от нейронов скрытого слоя и выходным нейроном,

θ – смещения для нейронов входного слоя.

Ниже представлены схематические изображения структуры однослойного перцептрона и однослойной сети Колмогорова-Арнольда. Для простоты и KAN, и перцептрон, с одним скрытым слоем, будем называть однослойными.

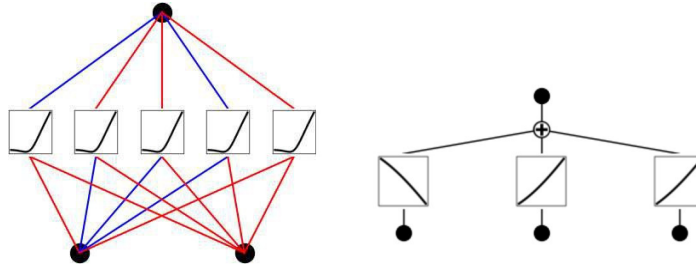


Рисунок 5. Однослойный перцептрон (слева) и однослойная сеть Колмогорова-Арнольда (справа)

Для KAN количество обучаемых параметров будет считаться

$$p = (d_{in} \times d_{out}) \times (G + K + 3) + d_{out}, \quad (3)$$

где G – интервалы, K – порядок сплайна, константа 3 – веса В-сплайнов, shortcut-веса и смещение.

Для MLP количество параметров определяется следующим образом:

$$p = (d_{in} \times d_{out}) + d_{out}. \quad (4)$$

2. Реализация сети Колмогорова-Арнольда

По теореме Колмогорова-Арнольда непрерывная функция многих переменных на интервале может быть выражена через конечную композицию непрерывных функций одной переменной и операции сложения. Определим KAN-слой как матрицу из одномерных функций:

$$\Phi = \{\Phi_{q,p}\}, \quad p = 1, 2, \dots, n_{in}, \quad q = 1, 2, \dots, n_{out}. \quad (5)$$

По образу MLP сетей мы можем создавать KAN-слои необходимой ширины и последовательно подключать их для достижения глубины.

В матричной форме один слой представляется как

$$X_{l+1} = \Phi_l X_l, \quad (6)$$

где X_l – тензор входных параметров, X_{l+1} – тензор выходных параметров, Φ_l – матрица функций l -го слоя KAN.

$$\Phi_l = \begin{pmatrix} \phi_{l,1,1}(\cdot) & \phi_{l,1,2}(\cdot) & \dots & \phi_{1,n_l}(\cdot) \\ \phi_{l,2,1}(\cdot) & \phi_{l,2,2}(\cdot) & \dots & \phi_{2,n_l}(\cdot) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{l,n_{l+1},1}(\cdot) & \phi_{l,n_{l+1},2}(\cdot) & \dots & \phi_{l,n_{l+1},n_l}(\cdot) \end{pmatrix} \quad (7)$$

$$KAN(x) = (\Phi_{L-1} \circ \Phi_{L-2} \circ \dots \circ \Phi_1 \circ \Phi_0)x. \quad (8)$$

Так как все операции дифференцируемы, может быть использован алгоритм обратного распространения ошибки. Функция активации в KAN определяется как

$$\phi(x) = w_b b(x) + w_s spline(x), \quad b(x) = silu(x) = x/(1 + e^{-x}), \quad (9)$$

где $spline(x)$ – линейная комбинация В-сплайнов:

$$spline(x) = \sum_i c_i B_i(x). \quad (10)$$

Здесь обучаемыми параметрами являются w_b , w_s и c_i . Функции активации инициализируются $w_s = 1$ и $spline(x) \approx 0$ по принципу инициализации Ксавье, используемому для инициализации линейных слоев в MLP. Так как сплайны определены на ограниченных отрезках, а признаки могут выйти за пределы этих отрезков – интервалы для сплайнов обновляются «на лету».

Реализация функций активации остается обширным полем для исследований и экспериментов, коэффициенты w_b , w_s – обучаемы, но не обязательны, они могут быть выражены самими $b(x)$ и $spline(x)$. Функция $b(x)$ может быть отброшена, а вместо $spline(x)$ можно использовать другие непрерывные параметризуемые функции, как это уже сделано в работах [1,2].

3. Организация среды для исследования KAN

Для работы использовалась реализация KAN из библиотеки PyKAN. Она основана на torch, и в ней реализованы такие инструменты, как создание датасета из математической функции, генерация изображения сети на текущем этапе обучения, отображение графика функций активации на ребрах сети. С помощью данной библиотеки можно создавать как однослойные KAN, так и многослойные, возможно настроить количество сегментов, диапазон и порядок для сплайнов – функций активации. Также была использована библиотека convKAN для тестирования сверточной сети. Для MLP-сетей, обработки датасетов при обучении и валидации была использована библиотека torch 2.7.0 для версии cu128. Для подготовки датасетов к обучению, разделению на обучающие и тестовые наборы, расчета весов классов для обучения – pandas, sklearn, numpy. Для визуализации отношений признаков к классам для анализа использовалась библиотека seaborn. Для визуализации графиков, диаграмм изображений, отчетов – matplotlib. Для обработки изображений – rembg, cv2, PIL. Для контроля версий и экспериментов была использована библиотека clearml. Исследование проводилось в локальном Jupyter Notebook в IDE Visual Studio Code, часть кода была исполнена в Py файлах, так как у Jupyter Notebook бывают проблемы с работой мультипоточности, например для torch.utils.data.DataLoader для использования num_workers > 0.

Работа выполнена на Python 3.12.10. Оборудование, на котором проводилось исследование:

- GPU: Nvidia GeForce RTX 3060 12Gb, Cuda Toolkit v12.8, CUDNN v9.10;
- CPU: Intel core i-5 12400f, RAM: DDR4 – 48Gb.

4. Механизмы и маркеры стресса растений

Одним из ранних признаков стресса для растений является повышение температуры листьев [8–10] на 0.2°. Такой признак непросто детектировать, так как это повышение совсем незначительное и происходит еще до потери воды растением и видимых изменений на листьях и связано с изменением режима дыхания для сбережения воды. Обнаружение стресса до начала потери воды считается «ранним» обнаружением. Другим признаком является изменение в спектральной отражательной способности, которое появляется из-за того, что болезни и стресс вызывают изменения в пигментном составе и структуре тканей листа, что влияет на отражающие характеристики листа в различных участках спектра. В детектировании таких признаков помогают гиперспектральные камеры. В силу высокого спектрального разрешения (3-10 нм) они позволяют вычислять различные вегетационные индексы, которые чувствительны к состоянию зеленой массы растения [8, 9].

Существуют и более высокоуровневые признаки, позволяющие довольно точно определять болезни растений на ранних стадиях. Исследования показывают, что энтропия, рассчитанная на основе гистограммы вегетационных индексов пикселей изображения растения, может служить универсальным и объяснимым высокоуровневым признаком для ранней диагностики стресса. Кроме энтропии, другие статистические характеристики гистограммы (минимум, максимум, среднее, стандартное отклонение и max-min) также несут информацию о состоянии растения.

5. Вегетационные индексы

Для раннего детектирования болезней растений часто используют вегетационные индексы, которые можно получить из гиперспектральных и тепловизионных изображений растений. В данной работе использованы индексы:

$NDVI = \frac{NIR-R}{NIR+R}$ – отношение разности коэффициентов отражения, ближнего инфракрасного (NIR) и красного (R) диапазонов спектра и их суммы, указывает на активность фотосинтетической биомассы;

$VARI = \frac{G-R}{G+R-B}$ – охватывает весь видимый сегмент спектра, позволяет минимизировать влияние атмосферных эффектов и разницы в освещении;

$NDBlue = \frac{G-B}{\max(G-B)-\min(G-B)}$ – полезен для отделения растительности от фона.

Для извлечения статистических характеристик использовались стандартные R, G, и B каналы изображения и индексы NDVI, VARI. Для расчёта NDVI канал NIR был заменен на канал G: $NDVI = \frac{G-R}{G+R}$, VARI изначально предназначен для RGB-изображений.

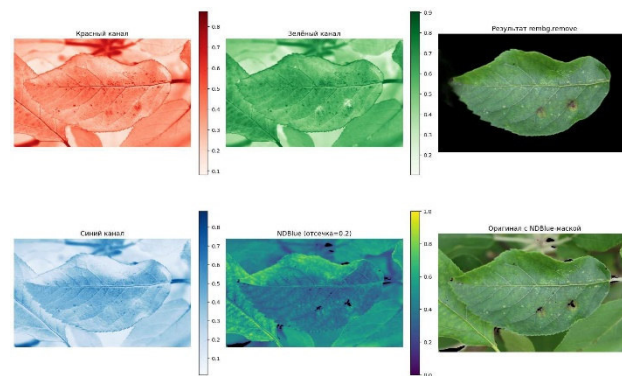


Рисунок 6

На рисунке 6 показан пример обработки изображения, полученного в полевых условиях (неоднородный фон, состоящий из других листьев). Четыре изображения со шкалами – каналы изображения Красный, Синий, Зеленый и индексный NDBI-канал. Изображения справа сверху – сегментация с помощью библиотеки rembg (U-Net) и снизу – сегментация с помощью маски значений NDBIue (threshold = 0.2).

6. Используемые датасеты

В ходе исследования использованы и обработаны различные датасеты, включая Plant Village, BRACOL, Plant Pathology, RoCoLe, Citrus и Rice Leaf Diseases, ссылки на которые перечислены в таблице 5. Каждый из этих датасетов включает изображения растений (как здоровых, так и пораженных различными заболеваниями), что обеспечивает широкий спектр тестовых данных для валидации моделей. Техническое описание датасетов дано в таблице 5.

Таблица 5. Обзор публичных датасетов, использованных в исследовании [19]

Dataset	DiaMOSPlant [5]	BRACOL [12]	RoCoLe [11]	Plant Pathology [14]	Rice Leaf Diseases[13]	Citrus [15]	APDA [16]	Plant-Village [7]
Plant/Crop	Pear	Coffee	Coffee	Apple	Rice	Citrus	Rose	Multiple
Dataset size	3505(3006 leaf images+499 fruit images)	4407	1560	3651	120	759 (609 leaf images + 150 fruit images)	40	54.309
Symptoms No	4	4	2	3	3	5	2	26
Acquisition device	Smartphone DSRL	Smartphone	Smartphone	DSLR Camera, Smartphone	DSLR camera	DSLR camera	Smartphone	Smartphone
Color	RGB	RGB	RGB	RGB	RGB	RGB	RGB	RGB
Image resolution	Multiple	2048 × 1024	Multiple	2048 × 1365	2848 × 4288	256 × 256	N.d.	Multiple
Annotation	Polygon, Label	Polygon, Label	Polygon, Label	Label	Label	Label	Label	Label
Annotation format	csv, YOLO	csv	COCO, JSON, Pascal, VOC, csv	csv	Folder structure	Folder structure	N.d.	Folder structure
Data sharing platform	Zenodo	GitHub	Mendeley Data	Kaggle	UCI Machine Learning Repository	Mendeley Data	MathWorks	Github
Acquisition place	Field	Laboratory	Field	Field	Laboratory	Laboratory	Laboratory	Laboratory
Side of the leaf	Adaxial	Abaxial	Adaxial, Abaxial	Adaxial	Adaxial	Adaxial	Adaxial	Adaxial
Object of interest	Fruit, leaf	Leaf	Leaf	Leaf	Leaf	Fruit, leaf	Leaf	Leaf

Для обеспечения единообразия и повышения качества обобщающей способности моделей на каждом датасете были переопределены и стандартизированы классы, что позволило более точно оценить эффективность классификации как внутри каждого датасета, так и в сравнении с эталонным датасетом Plant Village. Для каждого датасета было подготовлено две версии: одна содержит исходные изображения, а другая – изображения, преобразованные в нормализованные статистические характеристики. Такой подход позволил провести комплексное тестирование и выявить сильные и слабые стороны моделей при работе как с визуальными, так и с абстрактными признаками. В частности, исходный датасет Plant Village включает 54309 изображений размером 256×256 пикселей в формате RGB и содержит 38 несбалансированных классов растений и заболеваний. Анализ, проведенный в работе [18], выявил значительные закономерности в данных, влияющие на обобщающую способность сверточных нейронных сетей. Например, многие изображения одного класса сняты в одинаковых условиях освещенности, на одном фоне, с характерными тенями и высокой контрастностью, тогда как другой класс мог отличаться по этим параметрам, но оставаться однородным внутри себя. Это приводило к высокой точности классификации (приближающейся к 100 %) на тестовой выборке внутри датасета, но к резкому падению точности (до 30 %) при попытке классифицировать изображения из других датасетов [23-25, 27, 28]. Для устранения этих закономерностей и повышения устойчивости моделей к различным условиям съёмки, фон изображений был удалён с помощью индекса NDBI и библиотеки Rembg, основанной на сети U-Net. Кроме того, при подготовке данных для обучения использовалась конвертация изображений в нормализованные статистические характеристики, что должно позволить сфокусироваться на признаках, связанных непосредственно с заболеванием растений, исключая пространственные и визуальные артефакты. Такой подход, несмотря на возможные ограничения, обеспечивает более объективную оценку классификаторов и закладывает основу для последующего анализа.

Результаты экспериментов и обсуждение

7. Аппроксимация математической функции

В первую очередь рассмотрим задачу аппроксимации математической функции на примере функции

$$f(x, y) = e^{(\sin(\pi \cdot x) + y^2)}.$$

Для сети KAN будем использовать 5 скрытых нейронов, 3 интервала для функций активации, сплайны третьего порядка: $p = l_1 + l_2 = ((2 * 5) * (3 + 3 + 3) + 5) + ((5 * 1) * (3 + 3 + 3) + 1) = 141$.

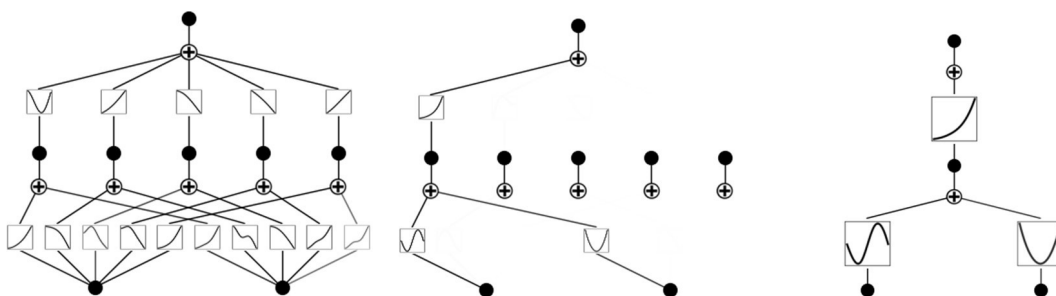


Рисунок 7

На рисунке 7 слева направо представлены модель при инициализации, модель после 50 эпох обучения, модель после операции `prune()` и дообучения. После 50 эпох большая часть нейронов «отключается» и остаются только самые активные. «Отключившиеся» нейроны удаляются с помощью функции `prune()` для оптимизации процесса обучения. После этапа обучения модель можно автоматически привести к символьной формуле, для исходного выражения $f(x, y) = e^{(\sin(\pi \cdot x) + y^2)}$ была получена формула $e^{y^2 + \sin(3.1416 \cdot x)}$.

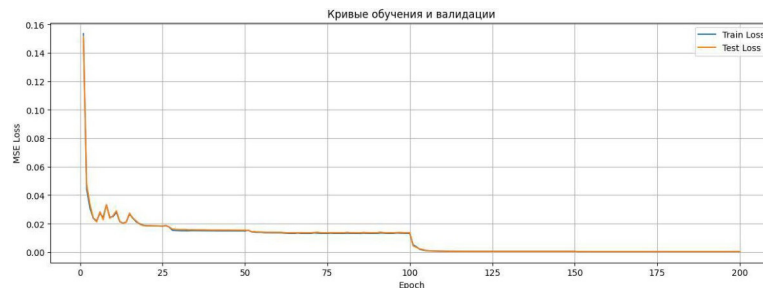


Рисунок 8. График потерь KAN, 200 эпох

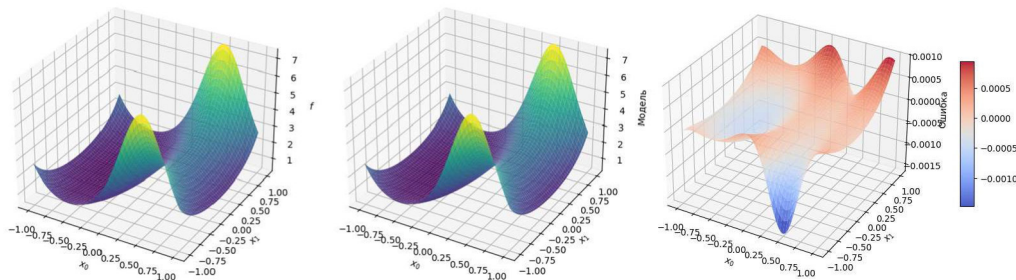


Рис. 9.

На рисунке 9 слева направо приведены графики: истинной функции, предсказание модели, отклонение модели от истинной функции. Обучение, после 200 эпох, снижает отклонение модели до величин не более $4e-5$. Длительность обучения составила примерно 30 секунд. Значение MSE составило $6.4e-18$. Эксперимент был выполнен на GPU.

Для однослойного перцептрона было использовано 35 скрытых нейронов. Так, $p = l_1 + l_2 = ((2 * 35) + 35) + ((35 * 1) + 1) = 141$.

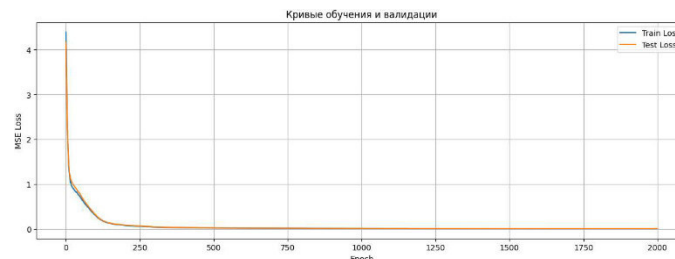


Рисунок 10. Функции потерь для SLP, 2000 эпох

Кривая потерь прекращает существенно уменьшаться после 500 эпох.

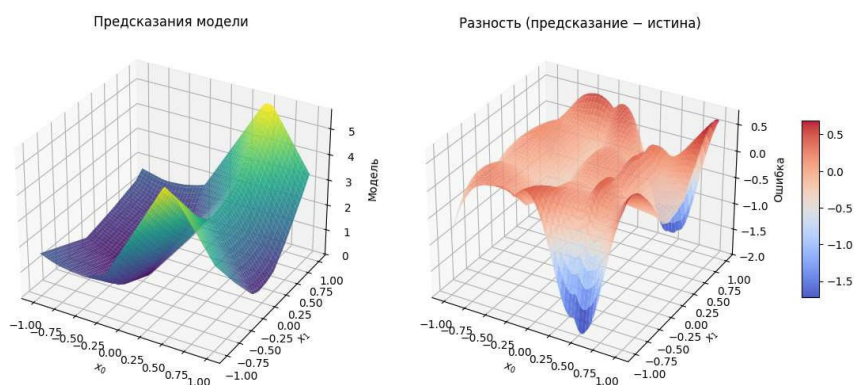


Рисунок 11. SLP после 200 эпох

После 200 эпох MSE составил $5,12e-1$. После 2000 эпох этот показатель составил $9,8e-3$. Данный эксперимент выполнен на GPU.

8. Детектирование болезней растений

8.1. Эксперимент 1: Бинарная классификация Plant Village

Ниже дано описание модели KAN в виде таблицы 6. Нейросеть принимает 30 значений, нормированных до интервала $[0, 1]$.

Таблица 6. Характеристики сети, тестируемой в эксперименте 1

Модель	Форма	Интервалы	Степень сплайна	Датасет обучения	Классификация
KAN	[30,32,2]	10	3	Plant Village	Бинарная

В этом и дальнейших тестах использованы:

- функция потерь: CrossEntropyLoss с весами классов для борьбы с несбалансированностью обучающей выборки;
- оптимизатор: AdamW с начальным `learning_rate = 5.2e-4` и `weight_decay = 1e-3`;
- планировщик: ReduceLROnPlateau с параметрами `patience = 20`, `factor = 0.8`, `mode = 'min'`, `min_lr = 1e-6`;
- размер батча: 256.

Датасеты предразбиты на стратифицированные обучающую, валидирующую и тестовую выборки в пропорции 7:1:1 для воспроизводимости экспериментов.

Обучали на протяжении 20 эпох при использовании GPU. Обучение продлилось 19 секунд, модель достигла 90,80 % точности на тестовой выборке из датасета Plant Village. Дальнейшее обучение практически не имеет смысла. Для такой сети обучение на 200 эпохах заняло 3 минуты 20 секунд и достигло точности 92.61 % на валидирующей выборке.

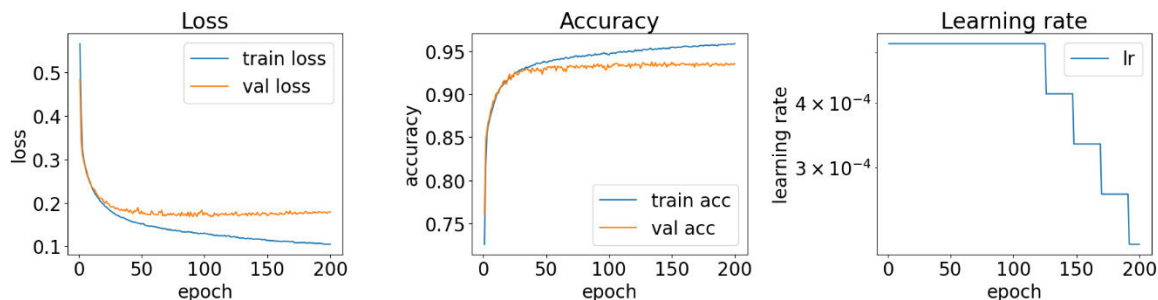


Рисунок 12. График потерь для обучающей и валидирующей выборок, график точности и график learning rate

Минимальное значение функции потерь на валидирующей выборке было достигнуто на 105 эпоху и составило 0.1679. Обучение до чекпоинта, на котором было достигнуто минимальное значение потерь на валидации, заняло 137 секунд, полное обучение до 200 эпох заняло 259 секунд. На тестовой выборке была достигнута точность 93,2 %. На основании этих данных можно сделать вывод, что модель близка к лимиту своих обобщающих возможностей и для улучшения результата потребуется более глубокая модель или широкие слои.

По диаграмме видно, что наиболее значимыми оказались R_std, B_std, G_mean, G_std, что расходится с результатами исследований [8, 9], это можно объяснить плохим качеством разметки датасета, о чем говорится в работе [18]. Попробуем проверить это, верифицировав результаты обучения на других датасетах перечисленных ранее, а также попробуем обучить сеть с аналогичными параметрами на других датасетах. Для датасетов, содержащих преимущественно больные листья, показатели F1, точности и полноты остаются высокими, что говорит о хорошей способности модели фиксировать патологию. Однако по мере увеличения доли здоровых растений качество классификации заметно снижается: модель чаще выдаёт ложные положительные ответы, но и доля ложных отрицаний остаётся ощутимой. На сбалансированных по классам наборах, таких как RaCoLe, точность падает до уровня

около 41 % – значительно ниже как простого случайного выбора, так и взвешенного случайного угадывания. Существенную роль в этом может играть и разнородность условий съёмки: часть наборов получена не в лабораторной среде, а в полевых условиях, что дополнительно осложняет задачу. Аналогичные выводы о падении качества вне контролируемых условий подтверждаются и в работе [18], критикующей датасет Plant Village.

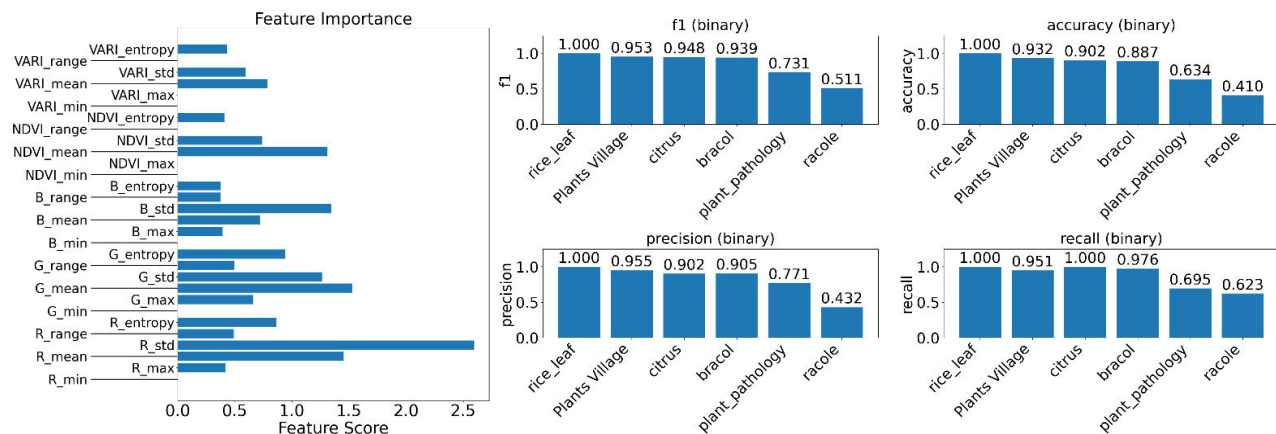


Рисунок 13. График важности признаков и метрик (F1, accuracy, precision, recall) на тестовых выборках

Попробуем обучить аналогичную сеть на датасете RaCoLe и валидировать ее на оставшихся.

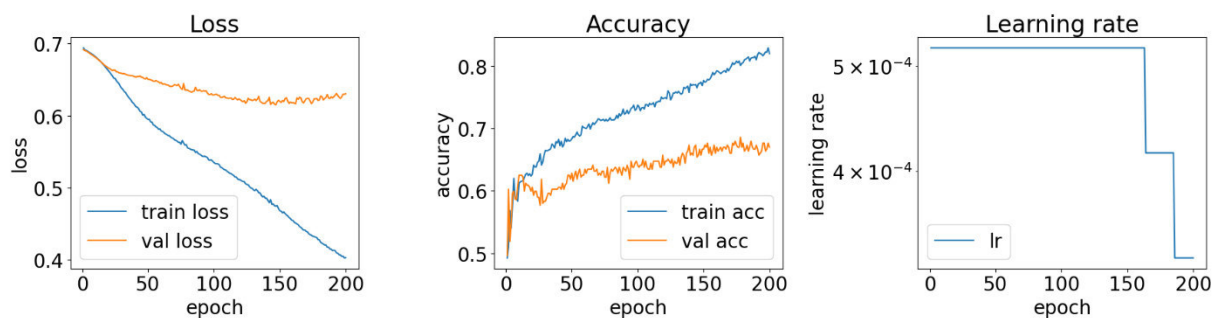


Рисунок 14. Функции потерь, точности и learning rate

При обучении аналогичной сети на датасете RaCoLe видим похожую картину – модель сильно переобучается. Минимальное достигнутое значение потерь на валидации 0.615 на 147 эпохе обучения. Обучение 200 эпох заняло 11 секунд. Заметен тренд на переобучение для KAN-моделей, здесь это также связано с тем, что данный датасет значительно меньше: 1560 записей против 54309.

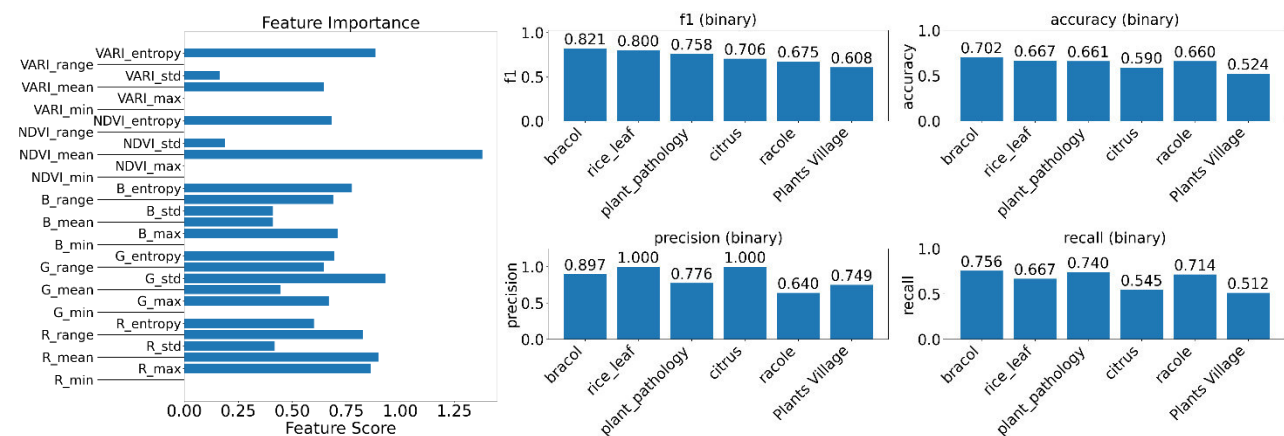


Рисунок 15. График важности признаков и метрик (F1, accuracy, precision, recall) на тестовых выборках

Диаграмма важности признаков показывает более равномерное распределение значимости: выраженных доминирующих параметров, как в предыдущем эксперименте, не наблюдается. Наибольший вклад в классификацию дает показатель NDVI_mean, однако его превосходство над остальными не столь велико. Итоговая проверка на внешних датасетах выявила снижение точности: модель лучше адаптировалась к собственному тренировочному набору, но при этом её способность к обобщению оказалась хуже. В отличие от обучения на Plant Village, где наблюдался более высокий перенос, здесь рост качества на «своём» датасете сопровождается ухудшением результатов на остальных.

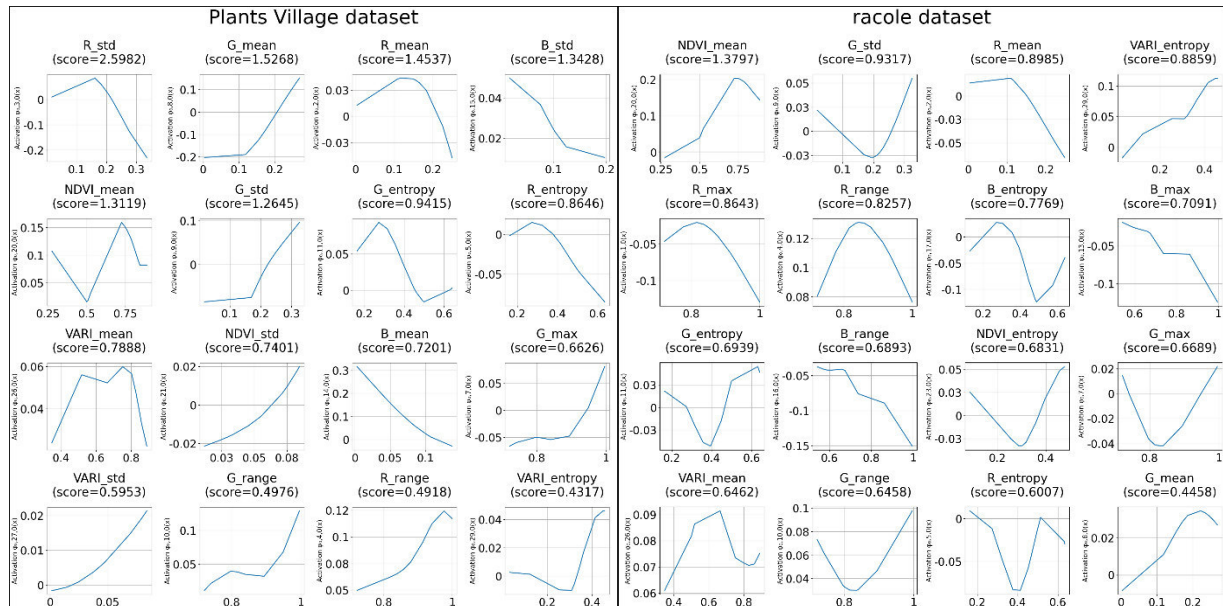


Рисунок 16. Графики функций активации модели, обученной на датасете Plant Village (слева) и датасете RaCoLe (справа)

На визуализации некоторых функций активации входного слоя мы видим, что некоторые функции активации, полученные после обучения на Plant Village и отсортированные по их score, имеют практически линейный вид, в то время как функции активации, обученные на датасете RaCoLe, вносят больше нелинейности. Стоит отметить, что на этом и последующем графиках функций активации оценивалась только общая форма функций, интервалы на которых определены после обучения, эти функции, конечно, играют важную роль, но пока неясно, как анализировать эти данные, так что они были оставлены малоразличимыми.

Заключительный эксперимент для бинарной классификации – обучение на комбинированном датасете, составленном из всех использованных, и последующая валидация.

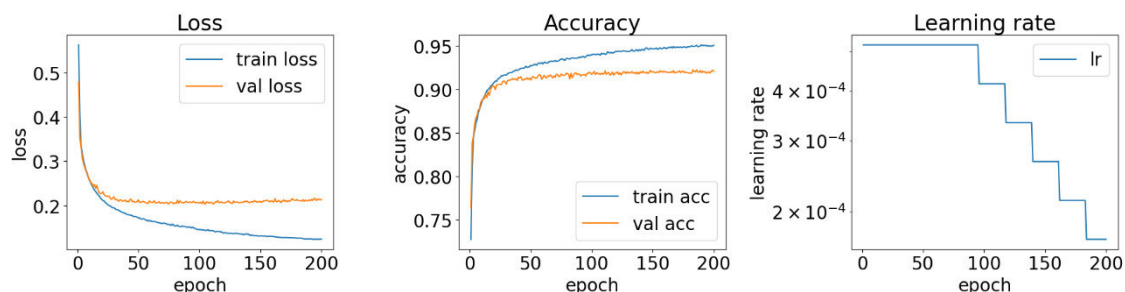


Рисунок 17. Графики функции потерь, точности и learning rate

На комбинированном датасете минимальное значение потерь на валидирующей выборке составило 0.2036 на 75-й эпохе за 108 секунд, обучение до 200-й эпохи заняло 284 секунды. Валидация, как и в

предыдущих экспериментах, выходит на плато, в то время как потери на тренирующей выборке продолжают падать.

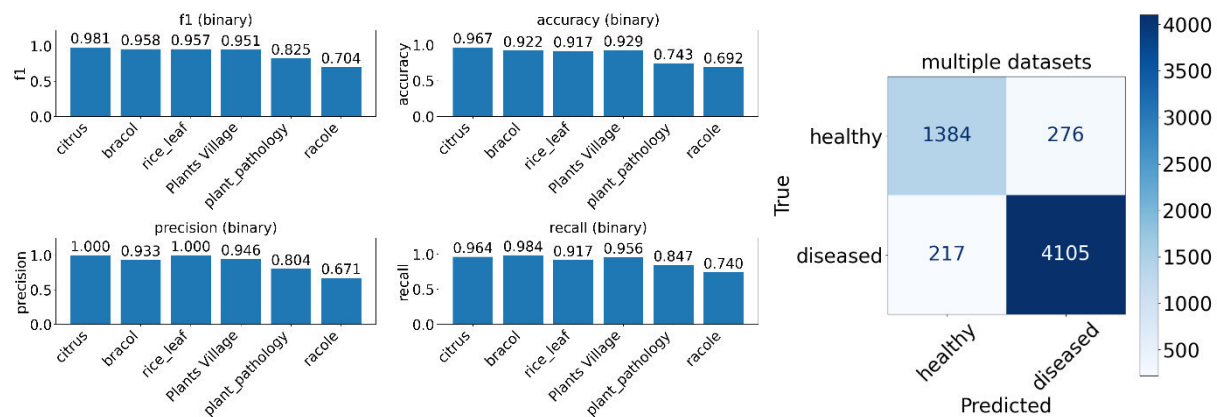


Рисунок 18. График метрик (F1, accuracy, precision, recall) на тестовых выборках и общая матрица несоответствий

По итогу обучения есть видимый перекоп в сторону False negatives. Точность при валидации на всех датасетах повысилась, теперь наименьшая точность 69,2 %. На датасете Citrus точность бинарной классификации достигла 96,7 %.

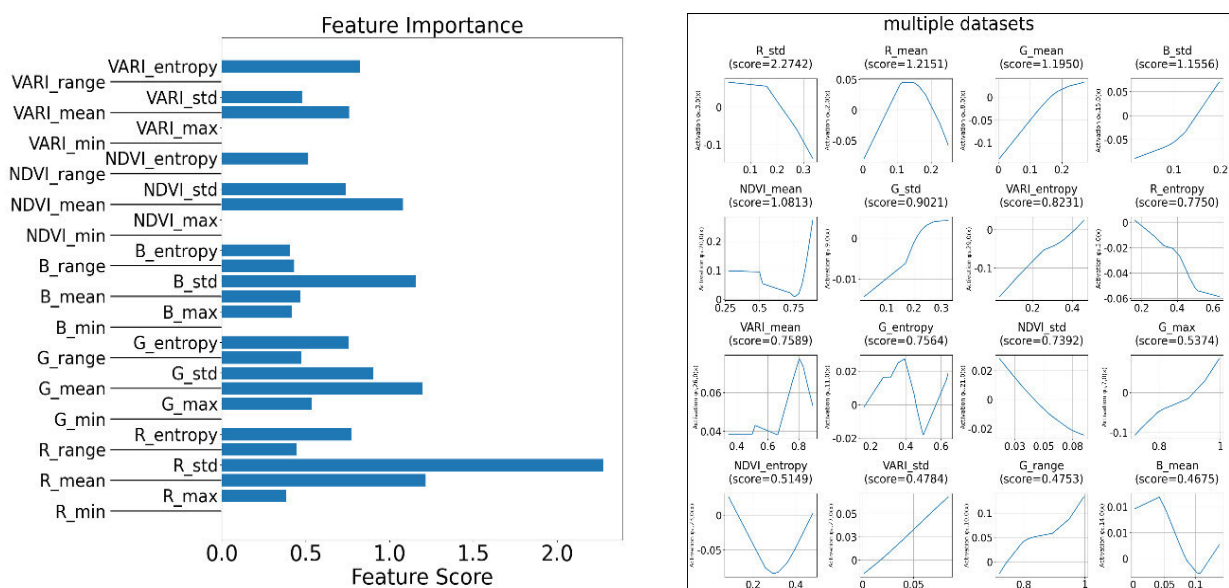


Рисунок 19. Диаграмма важности признаков и графики функций активации обученной модели

Итоговый вклад входных признаков очень близок к тому, что мы видели при обучении просто на Plant Village, это предсказуемо, так как даже комбинация всех остальных датасетов составляет небольшую долю от основного датасета. Аналогично для функций активации на входном слое.

Сравним полученную сеть с MLP с аналогичным количеством параметров, что по формулам (3) и (4) соответствует примерно 497 скрытым нейронам. Чтобы не перегружать публикацию, приводим результаты обучения MLP сразу на комбинированном датасете, результаты экспериментов, аналогичных двум предыдущим, будут указаны в итоговой таблице.

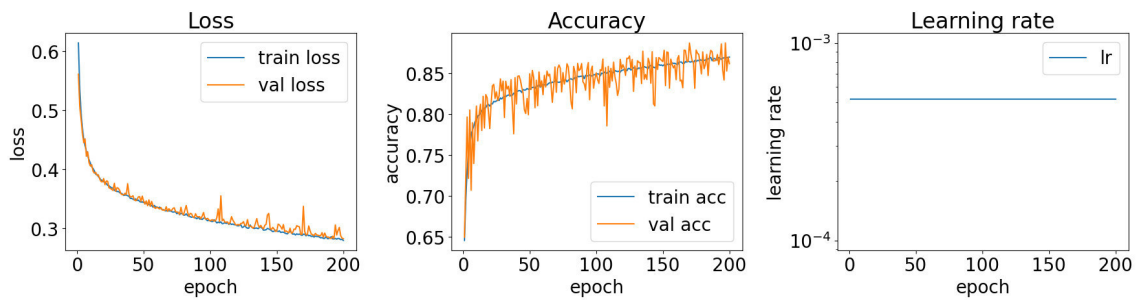


Рисунок 20. График функции потерь и точности MLP, обученной на 200 эпох

Обучение минимальное, значение потерь на валидации составило 0.2827 на 200 эпохе после 76 секунд обучения.

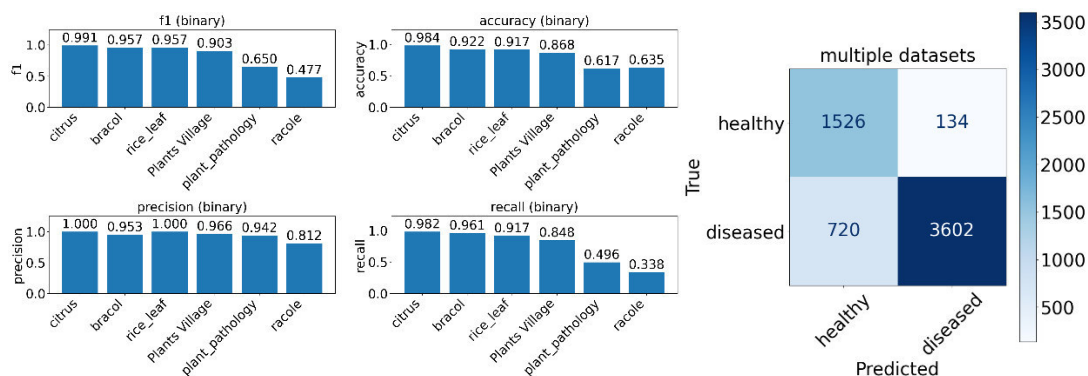


Рисунок 21. График точности на датасетах и матрица несоответствий обученной модели

На тестовой выборке перекося в сторону False negatives более сильный, чем у KAN, обученной на аналогичном датасете. По итогу модель MLP хуже обобщает, с минимальным результатом 61 % точности на одном из проверочных датасетов и дает худший Recall на общей тестовой выборке.

8.2. Эксперимент 2: мультиклассовая классификация растений

Таблица 7. Характеристики сети в эксперименте 2

Модель	Форма	Интервалы	Порядок сплайна	Датасет обучения	Классификация
KAN	[30,61,5]	20	3	Plant Village	Мульти класс

Проведем эксперимент, подобный эксперименту 1, но в этот раз будем делить растения по типам болезней из таблицы 1. Классы в обучающей выборке распределены неравномерно, всего 5 классов по типам болезней.

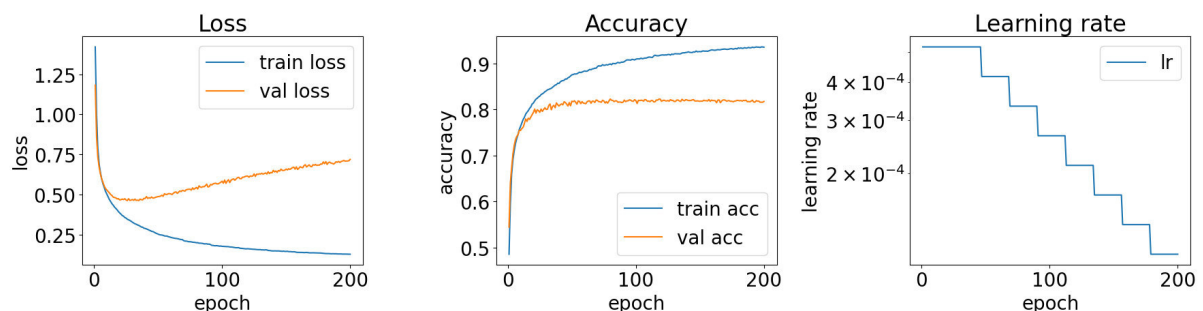


Рисунок 22. График функции потерь, точности и learning rate KAN при обучении задачи классификации из 5 классов на протяжении 200 эпох

Минимальное значение функции потерь на валидации составило 0.4638 и было достигнуто к 26-й эпохе через 34 секунды обучения, после чего модель начала переобучаться. Обучение 200 эпох заняло 259 секунд.

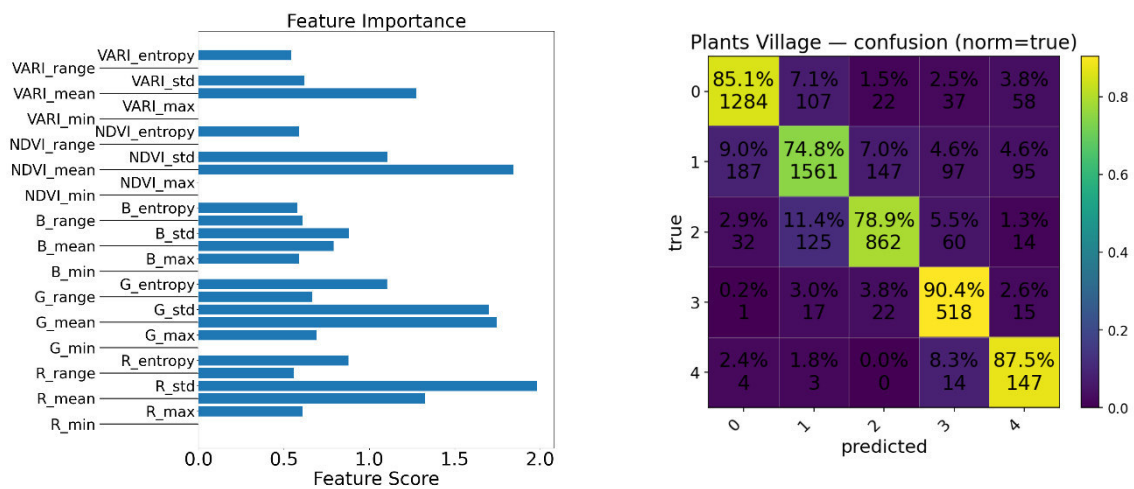


Рисунок 23. Диаграмма важности параметров и матрица несоответствий

Распределение оценок признаков отличается от полученного при бинарной классификации, более заметно разделение признаков по «важности», одними из важных признаков оказались B_std, B_mean, G_std, G_mean, R_std, G_entropy.

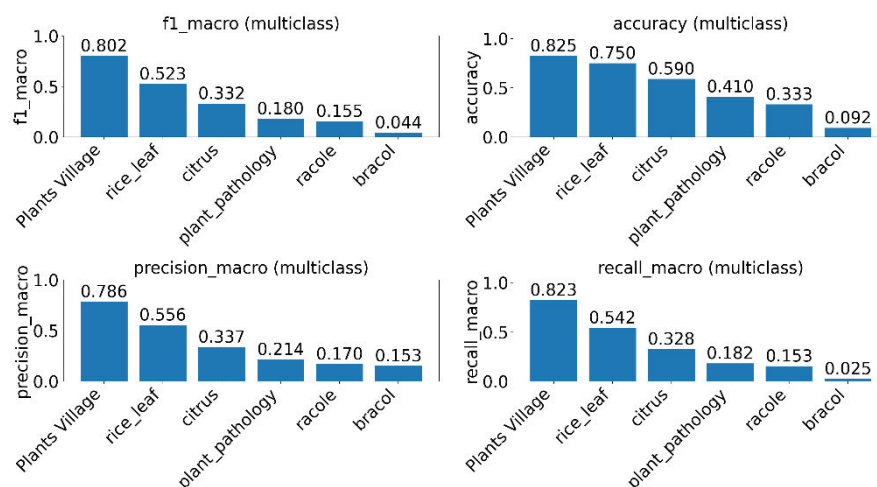


Рисунок 24. Диаграммы валидации на дополнительных датасетах

Попробуем обучить сеть сразу на комбинированном датасете. Минимальное значение потерь на валидации 0.5165 достигнуто на 20-й эпохе после 29 секунд обучения. Обучение до 200-й эпохи заняло 289 секунд. Как и в предыдущем эксперименте, обучение на комбинированном датасете выявило больше «важных» признаков на входном слое, но основные признаки остались как при тестировании на Plant Village.

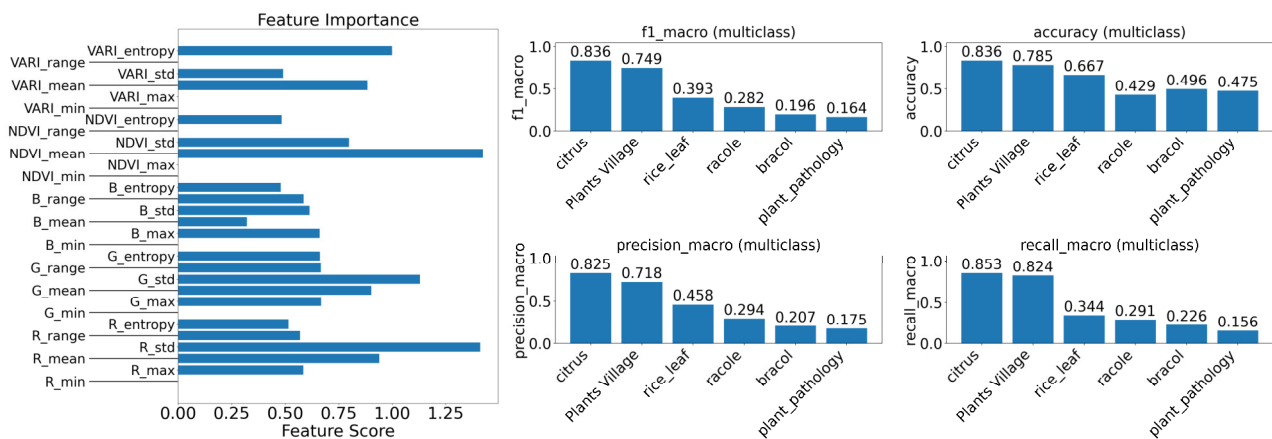


Рисунок 25. Диаграммы важности признаков и валидации на дополнительных датасетах

Обучение на комбинированном датасете позволило достичь высокой точности на всех датасетах. Минимальная точность составила 42.9 %, модель показала неплохую обобщающую способность.

Для сравнения была обучена модель MLP с 1540 нейронов на скрытом слое, что дает практически равное количество обучаемых параметров с нашей KAN-моделью в соответствии с формулами (3) и (4).

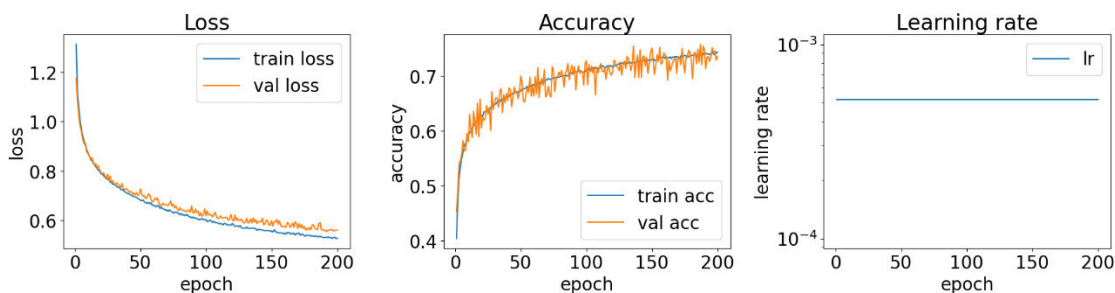


Рисунок 26. График функции потерь и точности MLP при обучении задачи классификации из 5 классов на протяжении 200 эпох

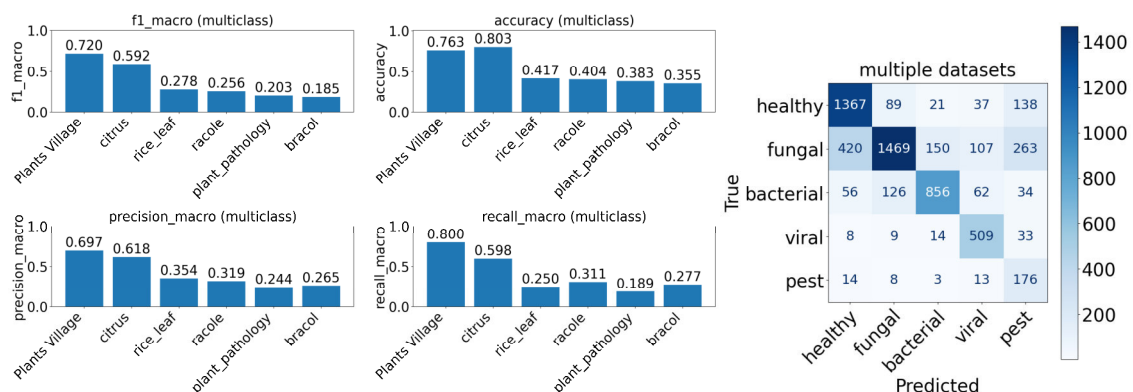


Рисунок 27. Диаграммы валидации на дополнительных датасетах и распределения предсказаний по классам

Классы на верифицирующей выборке определяются менее точно, точность на других датасетах в среднем тоже ниже, хотя обучение проходило на комбинированной версии датасета. На валидирующих датасетах модель показала неплохие результаты в мульти классовой задаче с лучшим показателем в 80.3 % точности.

2.3. Эксперимент 3: мультиклассовая классификация растений, сверточная KAN

В эксперименте рассмотрена сверточная KAN для классификации изображений 38 классов. Изображения отмасштабированы до размера 224*224 пикселей и содержали в себе 3 канала: RGB. На

основе архитектуры Convolutional Neural Network (CNN), использованной в работе [34], собрана аналогичная модель Torch со сверточными слоями KAN, где ядро свертки состоит из обучаемых функций активации. Результаты обучения сверточных вариантов нейросети будут приведены в итоговой таблице. На данный момент сверточные KAN-слои требуют слишком больших вычислительных мощностей. При обучении на видеокарте A100 40GB VRAM пришлось прибегнуть к понижению порядка сплайнов до 1 и снижению количества интервалов до 4. Сверточная сеть на основе KAN прошла 10 эпох обучения примерно за 11 часов, в то время как оригинальная CNN прошла 10 эпох обучения примерно за час. Для дальнейших исследований стоит опробовать менее глубокую архитектуру сверточной KAN и проверить, дадут ли обучаемые функции активации возможность достичь результатов, сравнимых с более глубокими CNN. Для будущих исследований подготовлено программное обеспечение, визуализирующее результат применения фильтров сверточных слоев (рис. 30).

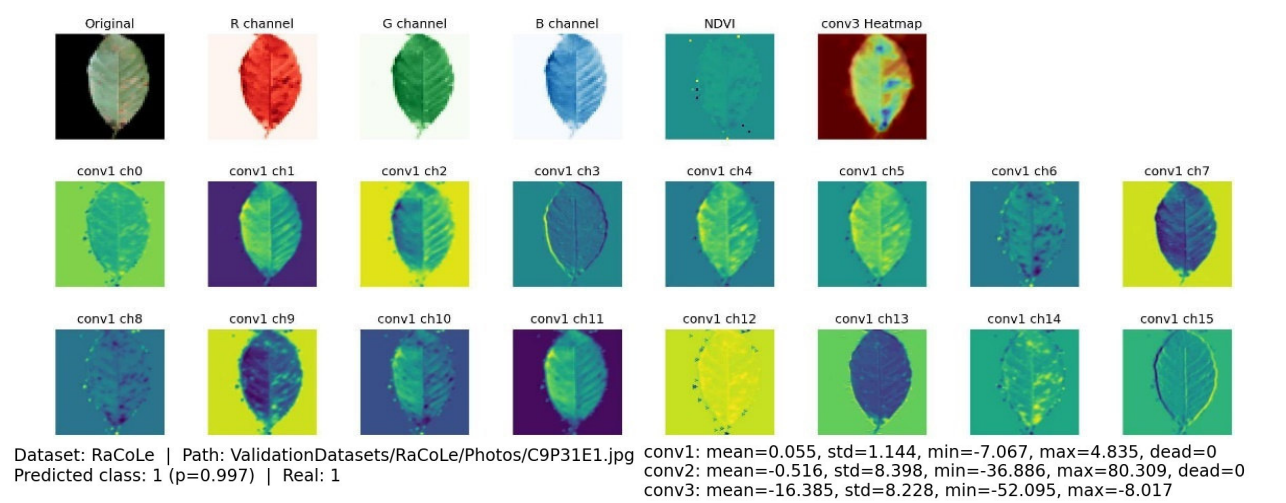


Рисунок 30. Пример отчета для дальнейшего исследования сверточных сетей на предмет объяснимости

Таблица 8. Общий результат тестирования

Описание эксперимента			Валидация					
Классификация	Модель	Обучающая выборка	BRACOL	Plant Pathology	RoCoLe	Citrus	Rice Leaf Diseases	Plant Village
Бинарная	KAN(2)	Plant Village	88,70%	63,40%	41,00%	90,20%	100,00%	93,20%
		RoCoLe	70,20%	66,10%	66,00%	59,00%	66,70%	52,40%
		Комбинированный датасет	92,20%	74,30%	69,20%	96,70%	97,70%	92,90%
	MLP(2)	Plant Village	90,10%	53,60%	50,60%	88,50%	100,00%	87,00%
		RoCoLe	92,20%	71,60%	69,20%	52,50%	83,30%	59,20%
		Комбинированный датасет	92,20%	61,70%	63,50%	98,40%	91,70%	86,80%
Мультиклассовая	KAN(5)	Plant Village	9,20%	41,00%	33,30%	59,00%	75,00%	82,50%
		Комбинированный датасет	49,60%	47,50%	42,90%	83,60%	66,70%	78,50%
	MLP(5)	Plant Village	0,01%	1,37%	1,03%	59,00%	50,00%	75,80%
		Комбинированный датасет	33,00%	51,00%	48,00%	76,00%	70,00%	78,00%
	convKAN (38)	Plant Village	59,33%	-	-	-	-	-
	CNN(38)	Plant Village	83,10%	-	-	-	-	-

Заключение

KAN-сеть показала хорошие результаты по отношению к MLP как на задаче регрессии математической функции, так и на задачах бинарной и мультиклассовой классификаций болезней растений относительно количества обучаемых параметров. Стоит отметить, что KAN-сети обучаются дольше из-за того, что расчёт B-сплайнов не распараллеливается на GPU, и из-за того, и что для их обучения требуется производить большее количество вычислений. Однако этот недостаток уже активно исправляется путем использования параллелизуемых базисных функций.

В плане дальнейшей работы изучить различия KAN и MLP по времени обучения; реализацию KAN на основе RBF – параллелизуемых на GPU; другие подходы к использованию KAN; сверточные KAN-слои; KAN трансформеры. Особый интерес вызывает дополнение стандартных архитектур нейронных сетей KAN-слоями. Подобная концепция уже рассматривалась в работе [29] почти одновременно с выходом первой публикации по KAN.

Источник финансирования

Работа выполнена при поддержке Министерства экономического развития Российской Федерации (соглашение о предоставлении гранта № 139-15-2025-004 от 17 апреля 2025 г., ИГК 000000Ц313925P3X0002).

Список литературы

- [1] KAN: Kolmogorov-Arnold Networks / Z. Liu, Y. Gao, Z. Zhang, et al. // arXiv preprint arXiv:2404.19756. 2024. Режим доступа: <https://doi.org/10.48550/arXiv.2404.19756>
- [2] KAN 2.0: Kolmogorov-Arnold Networks meet science / Z. Liu, Y. Gao, Z. Zhang et al. // arXiv preprint arXiv:2408.10205. 2024. Режим доступа: <https://doi.org/10.48550/arXiv.2408.10205>
- [3] KAN or MLP: A Fairer Comparison / R. Yu, W. Yu, X. Wang // arXiv preprint arXiv:2407.16674. 2024. Режим доступа: <https://doi.org/10.48550/arXiv.2407.16674>
- [4] Schmidt-Hieber J. The Kolmogorov–Arnold representation theorem revisited // Neural networks. 2021. Vol. 137. P. 119-126. Режим доступа: <https://doi.org/10.48550/arXiv.2007.15884>
- [5] Chebyshev Polynomial-Based Kolmogorov-Arnold Networks: An Efficient Architecture for Nonlinear Function Approximation / S.S. Sidharth, A.R. Keerthana, R. Gokul, K.P. Anas // arXiv preprint arXiv:2405.07200. 2024. Режим доступа: <https://arxiv.org/html/2405.07200>
- [6] Single-Layer Learnable Activation for Implicit Neural Representation (SL²A-INR) / M. Heidari, J. Wei, Z. Wang et al. // arXiv preprint arXiv:2409.10836. 2024. Режим доступа: <https://doi.org/10.48550/arXiv.2409.10836>
- [7] 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. Режим доступа: <https://doi.org/10.48550/arXiv.2112.09963>
- [8] Entropy as a High-Level Feature for XAI-Based Early Plant Stress Detection / M. Lysov, I. Maximova, E. Vasiliev, A. Getmanskaya, V. Turlapov // Entropy. 2022. Vol. 24, no. 11. P. 1597. Режим доступа: <https://doi.org/10.3390/e24111597>
- [9] Ensuring Explainability and Dimensionality Reduction in a Multidimensional HSI World for Early XAI-Diagnostics of Plant Stress / M. Lysov, K. Pukhkiy, E. Vasiliev, A. Getmanskaya, V. Turlapov // Entropy. 2022. Vol. 25, no. 5. P. 801. Режим доступа: <https://doi.org/10.3390/e25050801>
- [10] Barbedo J.G.A. Plant disease identification from individual lesions and spots using deep learning // Biosystems Engineering. 2019. Vol. 180. P. 96-107. Режим доступа: <https://doi.org/10.1016/j.biosystemseng.2019.02.002>
- [11] DropKAN: Regularizing KANs by masking post-activations / Z. Liu, T. Zhang, W. Wang, et al. // arXiv preprint arXiv:2406.09516. 2024. Режим доступа: <https://arxiv.org/pdf/2407.13044>
- [12] Kolmogorov-Arnold Transformer / Y. Zhang, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2409.10594. 2024. Режим доступа: <https://arxiv.org/abs/2409.10594>
- [13] Kolmogorov-Arnold Convolutions: Design Principles and Empirical Studies / J. Jiang, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2407.01092. 2024. Режим доступа: <https://arxiv.org/abs/2407.01092>
- [14] Kolmogorov-Arnold Fourier Networks / T. Chen, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2502.06018. 2024. Режим доступа: <https://arxiv.org/abs/2502.06018>
- [15] Wav-KAN: Wavelet Kolmogorov-Arnold Networks / J. Xu, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2405.12832. 2024. Режим доступа: <https://arxiv.org/abs/2405.12832>
- [16] CoxKAN: Kolmogorov-Arnold Networks for Interpretable, High-Performance Survival Analysis / H. Zhang, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2409.04290. 2024. Режим доступа: <https://arxiv.org/abs/2409.04290>
- [17] RKAN: Residual Kolmogorov-Arnold Network / Y. Li, Z. Liu, T. Zhang, et al. // arXiv preprint arXiv:2410.05500. 2024. Режим доступа: <https://arxiv.org/abs/2410.05500>
- [18] Noyan M. A. Uncovering bias in the PlantVillage dataset // arXiv preprint arXiv:2206.04374. 2022. Режим доступа: <https://arxiv.org/abs/2206.04374>

- [19] Fenu G., Mallocci F.M. DiaMOS Plant: A Dataset for Diagnosis and Monitoring Plant Disease // *Agronomy*. 2021. Vol. 11, no. 11. P. 2107. Режим доступа: <https://doi.org/10.3390/agronomy11112107>
- [20] A Novel Machine Learning Method for Estimating Biomass of Grass Swards Using a Photogrammetric Canopy Height Model, Images and Vegetation Indices Captured by a Drone / N. Viljanen, E. Honkavaara, R. Näsi, et al. // *Agriculture*. 2018. Vol. 8, no. 5. P. 70. Режим доступа: <https://doi.org/10.3390/agriculture8050070>
- [21] The global burden of pathogens and pests on major food crops / S. Savary, L. Willocquet, S.J. Pethybridge, et al. // *Nature Ecology & Evolution*. 2019. Vol. 3, no. 3. P. 430–439. DOI: 10.1038/s41559-018-0793-y
- [22] Oerke E.-C. Crop losses to pests // *The Journal of Agricultural Science*. 2006. Vol. 144, no. 1. P. 31–43. DOI: 10.1017/S0021859605005708
- [23] Hughes D., Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics // *arXiv preprint arXiv:1511.08060*. 2015. Режим доступа: <https://arxiv.org/abs/1511.08060>
- [24] RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition / J. Parraga-Alava, K. Cusme, A. Loor, E. Santander // *Data in Brief*. 2019. Vol. 25. P. 104414. DOI: 10.1016/j.dib.2019.104414
- [25] Krohling R., Esgario J., Ventura J.A. BRACOL-A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests // *Mendeley Data*. 2019. V1. DOI: 10.17632/yy2k5y8mxg.1
- [26] Prajapati H.B., Shah J.P., Dabhi V.K. Detection and classification of rice plant diseases // *Intelligent Decision Technologies*. 2017. Vol. 11. P. 357–373. DOI: 10.3233/IDT-170301
- [27] The Plant Pathology Challenge 2020 data set to classify foliar disease of apples / R. Thapa, K. Zhang, N. Snaveley, et al. // *Applications in Plant Sciences*. 2020. Vol. 8, no. 6. e11390. DOI: 10.1002/aps3.11390
- [28] A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning / H.T. Rauf, B.A. Saleem, M.I.U. Lali, et al. // *Data in Brief*. 2019. Vol. 26. P. 104340. DOI: 10.1016/j.dib.2019.104340. Heidari M. et al.
- [29] Poluektov M., Polar A. Construction of the Kolmogorov-Arnold representation using the Newton-Kaczmarz method // *arXiv preprint arXiv:2305.08194*. 2023.