

Разработка модели прогнозирования показателей животноводства с применением сетей Колмогорова-Арнольда

И.Е. Глазырин, В.А. Черепенин, М.Б. Хорошко

Южно-Российский государственный политехнический университет (НПИ) имени М. И. Платова, Новочеркасск

Аннотация: Данная статья исследует различные архитектуры нейронных сетей, с целью создания моделей в сфере сельского хозяйства, с акцентом на их использование в животноводческих хозяйствах. В работе приведено описание архитектуры сетей Колмогорова-Арнольда, рассмотрены этапы сбора и предварительной подготовки данных, процесс обучения нейронных сетей, а также их внедрения. В результате были разработаны модели с применением сетей Колмогорова-Арнольда и многослойного персептрона. В ходе исследования проведено сравнение эффективности предложенных архитектур. Эксперимент демонстрирует, что сети Колмогорова-Арнольда обладают более высокой точностью в предсказаниях, что делает их перспективным инструментом для прогнозирования. Разработанная модель была интегрирована в разрабатываемую информационную систему животноводческого хозяйства для прогнозирования роста, здоровья и других показателей животных, позволяя более точно управлять процессом выращивания.

Ключевые слова: точное животноводство, сеть Колмогорова-Арнольда, моделирование, нейронная сеть, мониторинг, выращивание, моделирование данных, прогнозирование.

Сочетание математических методов с новейшими достижениями в области компьютерных технологий служат основой для разработки инструментов интеллектуального анализа данных, ориентированных на прогнозирование различных ситуаций. Прогнозирование — это процесс предсказания будущих значений переменной на основе прошлых данных [1].

В рамках данного исследования будет разработана модель прогнозирования с применением архитектуры сетей Колмогорова-Арнольда (далее *KAN*). Цель разработки состоит в том, чтобы оптимизировать процесс выращивания, спрогнозировать состояние здоровья и динамику роста животных.

Архитектура *KAN* представлена 30 апреля 2024 года группой ученых университета *Massachusetts Institute of Technology (USA, Massachusetts)*. Она является альтернативой традиционному многослойному персептрону (далее *MLP*) и основывается на теореме о представлении Колмогорова-Арнольда.

Теорема о представлении Колмогорова-Арнольда утверждает [2], что если функция f является многомерной непрерывной функцией в ограниченной области, то ее можно записать как конечную композицию непрерывных функций одной переменной и бинарной операции сложения $f: [0,1]^n \rightarrow R$,

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

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

Представление Колмогорова-Арнольда можно записать в матричной форме:

$$f(x) = \Phi_{out} \circ \Phi_{in} \circ x,$$

где:

$$\Phi_{in} = \begin{pmatrix} \phi_{1,1}(\cdot) & \cdots & \phi_{1,n}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{2m=n+1,1}(\cdot) & \cdots & \phi_{2n+1,n}(\cdot) \end{pmatrix},$$
$$\Phi_{out} = (\Phi_1(\cdot) \cdots \Phi_{2n+1}(\cdot))$$

Можно заметить, что как Φ_{out} , так и Φ_{in} являются частными случайным следующей функциональной матрицы Φ (с n_{in} в качестве входов, n_{out} в качестве выходов), это называется слоем Колмогорова-Арнольда [3]:

$$\Phi = \begin{pmatrix} \phi_{1,1}(\cdot) & \cdots & \phi_{1,n_{in}}(\cdot) \\ \vdots & \ddots & \vdots \\ \phi_{n_{out},1}(\cdot) & \cdots & \phi_{n_{out},n_{in}}(\cdot) \end{pmatrix}$$

Φ_{in} соответствует $n_{in} = n$, $n_{out} = 2n + 1$ и Φ_{out} соответствует $n_{in} = 2n + 1$, $n_{out} = 1$.

После определения слоя, можно построить сеть Колмогорова-Арнольда. Допустим, у нас L слоев, с l -того слой Φ_l будет иметь форму $(n_l + 1, n_l)$. Тогда вся сеть:

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

В то время как *MLP* имеют фиксированные функции активации на узлах («нейронах»), *KAN* имеют обучаемые функции активации на ребрах («веса») [4]. *KAN* вообще не имеют линейных весов – каждый весовой параметр заменяется одномерной функцией, параметризованной в виде сплайна. Каждый сплайн аппроксимируется с помощью заданного количества точек, чем больше точек - тем точнее аппроксимация. Сплайны непрерывны и дифференцируемы, а значит, такую архитектуру можно спокойно обучать с помощью метода обратного распространения ошибки.

Для реализации была использована официальная библиотека *rukan* [5]. При разработке модели будет использоваться набор данных [6]. Он содержит информацию о выращивании 786 свиней, включая данные о поле, породе, режиму кормления, условиях выращивания, использовании антибиотиков. На рисунке 1 представлен график набора веса, построенный на основе информации из набора.

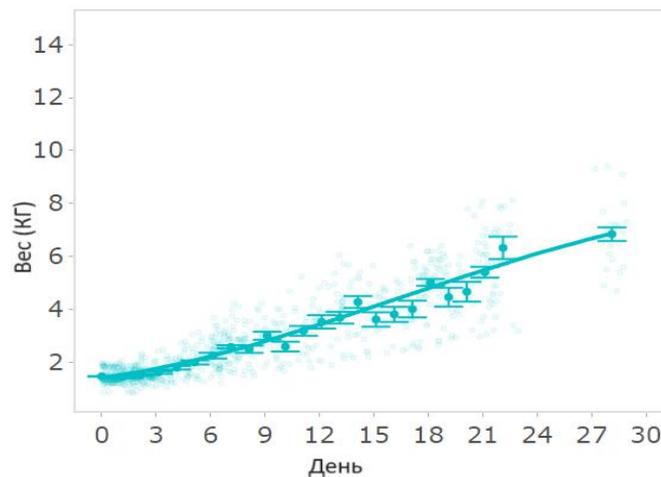


Рисунок 1 – График набора веса

Для реализации и обучения модели необходимо подготовить набор данных. Рассмотрим каждый из параметров:

- дата: дата и время замера;
- номер дня: номер дня от начала выращивания;
- номер группы: идентификатор группы животных;



- тип группы: тип животных;
 - начальное количество животных: число особей в группе на начало выращивания;
 - начальный вес животных: вес группы на начало выращивания;
 - общий прирост веса в процентах: прирост в весе в период от начала выращивания до даты замера в процентах;
 - общий прирост веса в граммах: прирост в весе в период от начала выращивания до даты замера в граммах;
 - общий среднесуточный прирост: среднесуточный прирост в весе в период от начала выращивания до даты замера;
 - общий коэффициент конверсии корма: в период от начала выращивания до даты замера;
 - общее количество заболевших: количество заболевших в период от начала выращивания до даты замера;
 - общее количество умерших: количество умерших в период от начала выращивания до даты замера;
 - текущее количество животных: количество особей в группе на момент замера;
 - текущий вес: общий вес группы на момент замера;
 - текущее количество заболевших: количество заболевших животных в группе за последний день;
 - текущее количество умерших: количество умерших животных в группе за последний день;
 - текущий прирост веса в процентах: прирост веса в процентах за последний день;
 - текущий прирост веса в граммах: прирост веса в граммах за последний день;
-

- текущий коэффициент конверсии корма: коэффициент, вычисленный на момент замера;
- будущее количество животных: количество особей в группе на момент $i+1$ -дня;
- будущий вес: общий вес группы на момент $i+1$ -дня;
- будущее количество заболевших: количество заболевших животных в группе на момент $i+1$ -дня;
- будущее количество умерших: количество умерших животных в группе на момент $i+1$ -дня;
- будущий прирост веса в процентах: прирост веса в процентах на момент $i+1$ -дня;
- будущий прирост веса в граммах: прирост веса в граммах на момент $i+1$ -дня;
- будущий коэффициент конверсии корма: коэффициент конверсии корма, вычисленный на момент $i+1$ -дня

В результате, набор данных для разработки модели содержит двадцать шесть характеристик, за 69 недель, с шагом в 30 минут. Эти характеристики можно логически разбить на четыре группы: начальные (описывают состояние группы за начало выращивания), за все время (описывают состояние группы с первого по i -день), за последний день (описывают состояние группы на i -день), за будущий день (описывает состояние группы на $i+1$ -день). Для обучения модели, данные будут разделены на три части: 70% - обучающие данные, 20% - проверочные данные, 10% - тестовые данные. После обучения модель будет способна предсказывать несколько из параметров, в нашем случае прирост в весе, количество заболевших, количество умерших, коэффициент конверсии корма.

Для оценки показателей модели были получены метрики среднеквадратичной ошибки (MSE), корень из среднеквадратичной ошибки

(*RMSE*), средняя абсолютная потеря (*MAE*), средняя абсолютная ошибка в процентах (*MAPE*). На рисунке 2 представлены показатели, полученные при обучении модели. На рисунке 3 представлен график набора веса, синим цветом выделены фактические значения, оранжевым – прогнозируемые.

MSE: 5.08
RMSE: 7.12
MAE: 5.06
MAPE: 0.52

Рисунок 2. – Показатели модели с архитектурой *KAN*

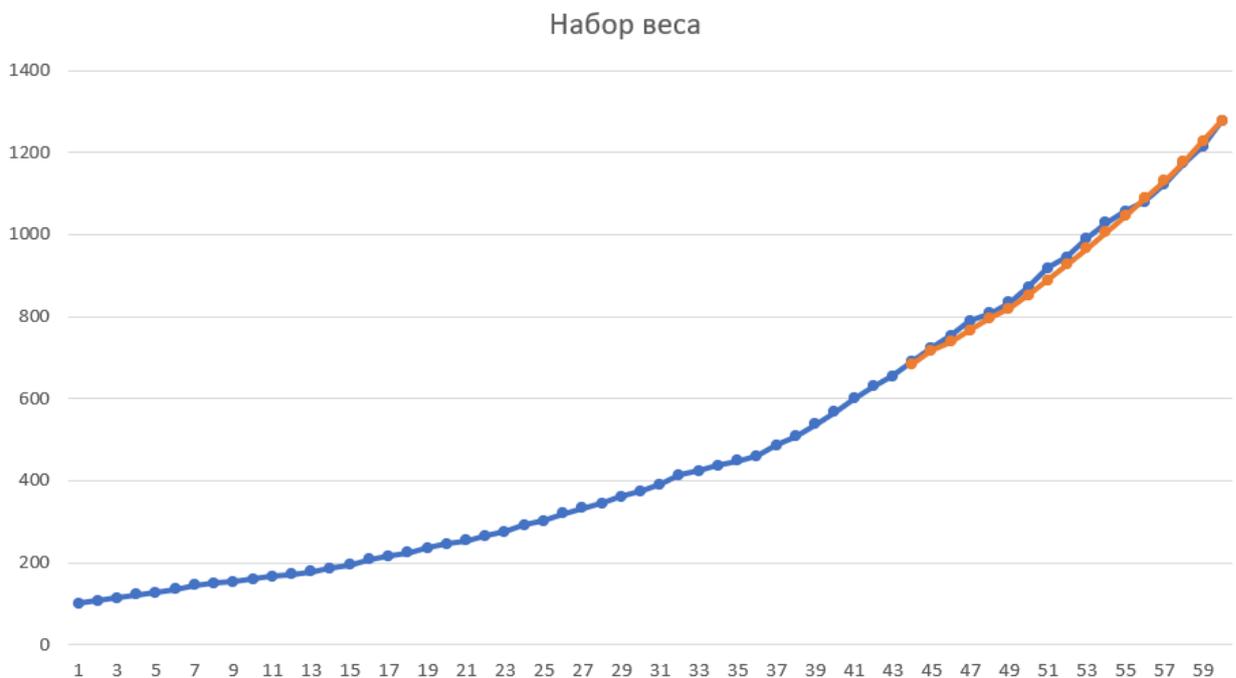


Рисунок 3. – График набора веса с прогнозируемыми значениями модели с архитектурой *MLP*

Данная модель была интегрирована в систему, описание разработки, которой дано ранее [7]. Для этого, она была выгружена в *pt*-файл, который представляет собой сериализованный объект, содержащий параметры обученной модели нейронной сети. Дополнительно был создан *python*-скрипт, который может быть вызван с помощью интерфейса командной строки (далее *CLI*). Он загружает модель из *pt*-файла, а по завершению его работы посредством чтения вывода *CLI* возвращает предсказанное значение

[8]. После чего скрипт можно запустить из *Node.js* с помощью встроенной функции «*child_process*», передавая входящие параметры как аргументы. Экранная форма с прогнозируемыми данными представлена на рисунке 4.

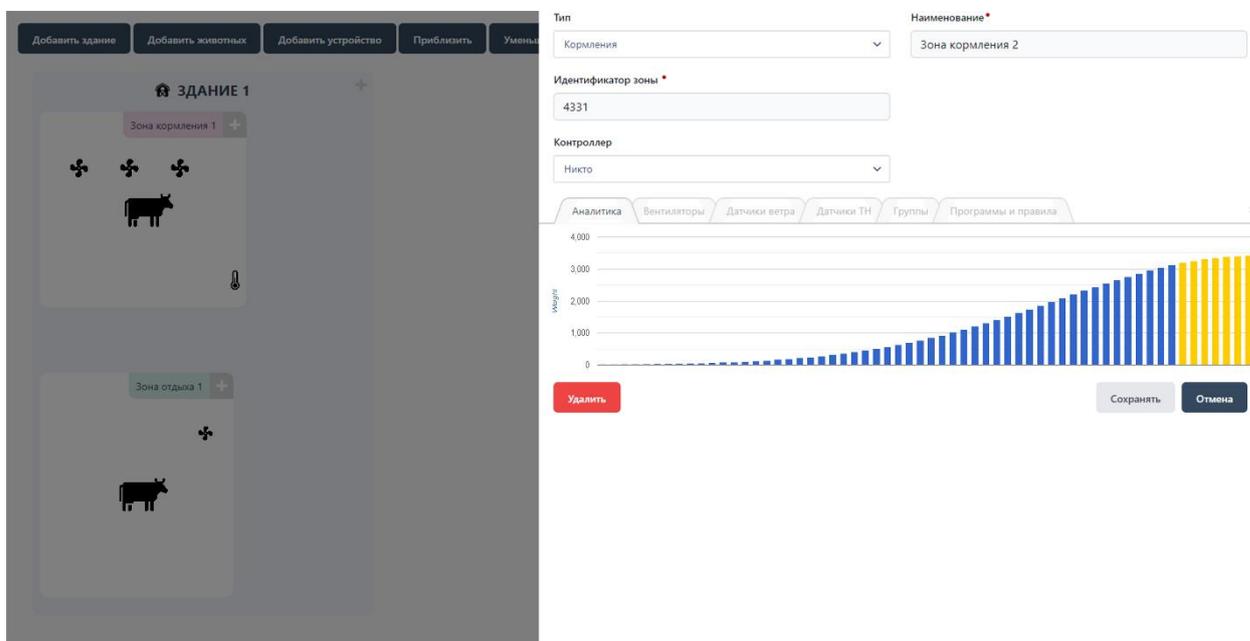


Рисунок 4. – Экранная форма просмотра данных зоны

На данной экранной форме представлен график роста свиньи. Значения, отмеченные желтым цветом – получены с помощью моделей прогнозирования, описанной ранее. В заключение, реализация моделей нейронных сетей для прогнозирования показателей хода выращивания животных представляет собой перспективный и эффективный подход к оптимизации процессов в животноводстве [9].

Сравним полученные результаты с архитектурой *MLP*, создав аналогичную модель на её основе. Для реализации модели будет использоваться фреймворк *PyTorch*, его флагманской функцией является компиляция моделей, позволяя компилировать их для быстрого выполнения. На данный момент фреймворк занимает лидирующие позиции в научно-исследовательской работе [10]. Общий ход обучения и данные аналогичны предыдущей модели. На рисунке 5 представлены показатели, полученные

при обучении модели. На рисунке 6 представлен график набора веса, синим цветом выделены фактические значения, оранжевым – прогнозируемые.

MSE: 6.34
RMSE: 7.96
MAE: 5.41
MAPE: 0.65

Рисунок 5. – Показатели модели с архитектурой *MLP*

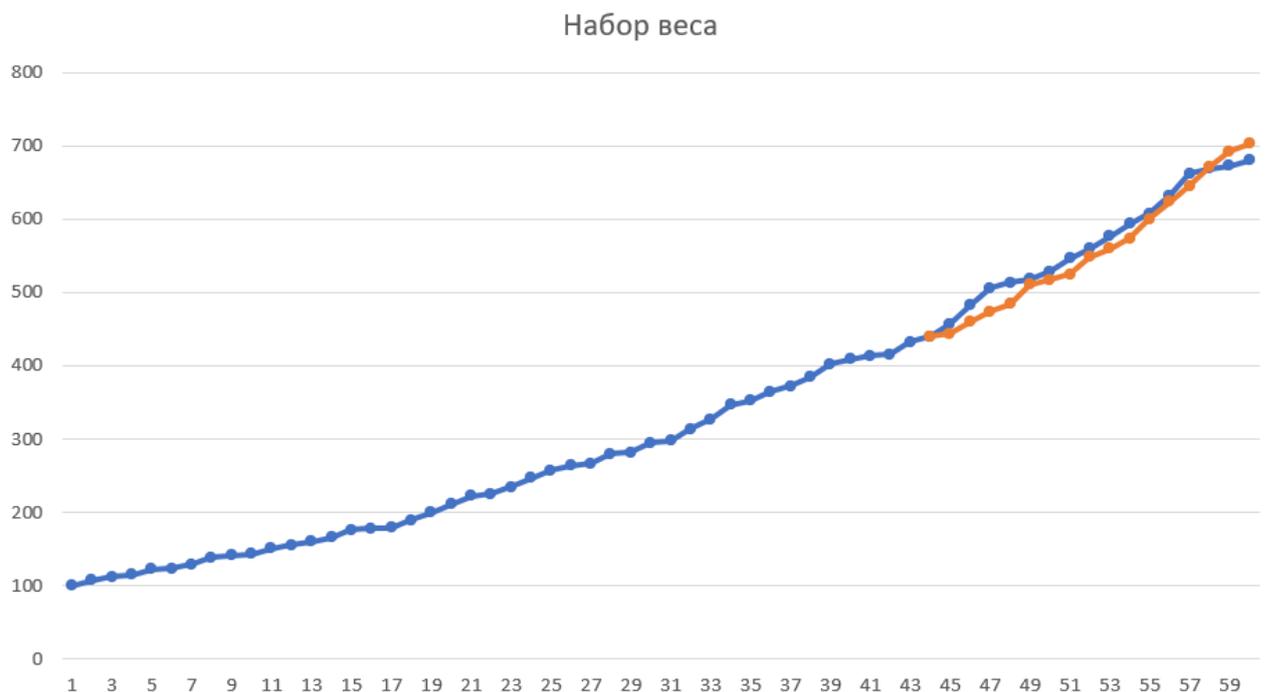


Рисунок 6. – График набора веса с прогнозируемыми значениями модели с архитектурой *MLP*

Результаты показали, что *KAN* превосходит *MLP* с точки зрения более низких показателей ошибок. Учитывая их эффективность и действенность, *KAN* кажутся разумной альтернативой традиционным *MLP* в решении задач анализа данных.

Литература

1. Кутузов Д.В., Осовский А.В., Старов Д.В., Перова К.В. Управление трафиком телекоммуникационных систем с использованием



прогностических методов искусственного интеллекта // Инженерный вестник Дона. 2024, №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8920.

2. Persi Diaconis, Mehrdad Shahshahani. On Linear Functions of Linear Combinations SIAM J. Sci. Stat. Comput.: journal. 1984. 5. doi: 10.1137/0905013.

3. Hello, KAN! — Kolmogorov Arnold Network documentation // Github URL: kindxiaoming.github.io/pykan/intro.html (дата обращения: 17.05.2024).

4. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljagic, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov–Arnold Networks // arxiv. URL: arxiv.org/pdf/2404.19756 (дата обращения: 14.08.2024).

5. Pykan // Github. URL: github.com/KindXiaoming/pykan (дата обращения: 14.08.2024).

6. Pig-database // GitHub. URL: github.com/Traverse-Science/Pig-database (дата обращения: 14.08.2024).

7. Глазырин И.Е. Разработка на основе модели IDEF0 информационной системы управления птицефермы по производству мяса цыплят-бройлеров // Инновационные научные исследования: сетевой журнал. 2021. № 4-3(6). С. 204-212. URL: ip-journal.ru.

8. Черепенин В.А., Воробьев С.П., Серкезюк В.Р. Реализация моделей нейронных сетей для прогнозирования показателей в умной теплице // Инженерный вестник Дона. 2024, №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8927.

9. Alexandratos N, Bruinsma J. World agriculture towards 2030/2050: the 2012 revision. Rome: Food and Agriculture Organization; 2012. URL: fao.org/3/ap106e/ap106e.pdf.



10. PyTorch vs TensorFlow in 2023 // assemblyai URL: assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/. (дата обращения: 17.05.2024).

References

1. Kutuzov D.V., Osovskij A.V., Starov D.V., Perova K.V. Inzhenernyj vestnik Dona. 2024. №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8920.
2. Persi Diaconis, Mehrdad Shahshahani. On Linear Functions of Linear Combinations SIAM J. Sci. Stat. Comput.: journal. 1984. 5. doi:10.1137/0905013.
3. Hello, KAN! — Kolmogorov Arnold Network documentation, URL: kindxiaoming.github.io/pykan/intro.html (accessed: 16.05.2024).
4. Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y. Hou, Max Tegmark. KAN: Kolmogorov–Arnold Networks, URL: arxiv.org/pdf/2404.19756 (accessed: 14.08.2024).
5. Pykan, URL: github.com/KindXiaoming/pykan (accessed: 14.08.2024).
6. Pigs-database. URL: github.com/Traverse-Science/Pig-database (accessed: 14.08.2024).
7. Glazyrin I.E. Innovacionnye nauchnye issledovaniya: setevoy zhurnal. 2021. № 4-3(6). pp. 204-212. URL: ip-journal.ru.
8. Cherepenin V.A., Vorob`ev S.P., Serkezyuk V.R. Inzhenernyj vestnik Dona. 2024. №1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8927.
9. Alexandratos N, Bruinsma J. Rome: Food and Agriculture Organization; 2012. URL: fao.org/3/ap106e/ap106e.pdf.
10. PyTorch vs TensorFlow in 2023. URL: assemblyai.com/blog/pytorch-vs-tensorflow-in-2023 (accessed: 16.05.2024).

Дата поступления: 15.07.2024

Дата публикации: 30.08.2024