

**УДК 004.852**

**Вильданов Алмаз Нафкатович**, к.ф.-м.н. Нефтекамский Университет Науки и Технологии, г. Нефтекамск

**Салихов Салават Зульфатович**, студент, Нефтекамский Университет Науки и Технологии, г. Нефтекамск

## **ПОЛНОСВЯЗНАЯ НЕЙРОННАЯ СЕТЬ ДЛЯ ЗАДАЧ МНОГОКЛАССОВОЙ КЛАССИФИКАЦИИ**

В статье рассматривается задача классификации биологических объектов на примере цветков ириса (Iris dataset) с использованием полносвязных нейронных сетей в среде Google Colab. Обосновывается целесообразность применения глубокого обучения для решения задач многоклассовой классификации на структурированных табличных данных. Предлагается метод предобработки данных, включающий стандартизацию признаков, и архитектура нейронной сети с тремя скрытыми слоями. Новизна подхода заключается в адаптации классической нейросетевой архитектуры для работы с малыми выборками и демонстрации полного конвейера обработки: от загрузки данных из Google Drive до выдачи результата в формате CSV. Экспериментальная оценка на наборе данных Iris показывает достижение точности на валидационной выборке не менее 95% при использовании 20 эпох обучения.

**Ключевые слова:** нейронные сети, классификация, Keras, TensorFlow, стандартизация, softmax, глубокое обучение, многоклассовая классификация.

The article addresses the problem of biological object classification using the Iris dataset as an example, employing fully connected neural networks in the Google Colab environment. The feasibility of using deep learning for multi-class classification tasks on structured tabular data is substantiated. A preprocessing

method including feature standardization and a neural network architecture with three hidden layers are proposed. The novelty of the approach lies in adapting classical neural network architectures for small-sample datasets and demonstrating a complete processing pipeline: from data loading from Google Drive to result output in CSV format. Experimental evaluation on the Iris dataset demonstrates validation accuracy of at least 95% using 20 training epochs.

**Keywords:** neural networks, classification, Keras, TensorFlow, standardization, softmax, deep learning, multi-class classification.

В статье рассматривается задача классификации биологических объектов на примере цветков ириса (Iris dataset) с использованием полносвязных нейронных сетей в среде Google Colab. Обосновывается целесообразность применения глубокого обучения для решения задач многоклассовой классификации на структурированных табличных данных. Предлагается метод предобработки данных, включающий стандартизацию признаков, и архитектура нейронной сети с тремя скрытыми слоями. Новизна подхода заключается в адаптации классической нейросетевой архитектуры для работы с малыми выборками и демонстрации полного конвейера обработки: от загрузки данных из Google Drive до выдачи результата в формате CSV. Экспериментальная оценка на наборе данных Iris показывает достижение точности на валидационной выборке не менее 95% при использовании 20 эпох обучения. Ключевые слова: нейронные сети, классификация, Iris dataset, Keras, TensorFlow, стандартизация, softmax, Google Colab, глубокое обучение, многоклассовая классификация.

The article addresses the problem of biological object classification using the Iris dataset as an example, employing fully connected neural networks in the Google Colab environment. The feasibility of using deep learning for multi-class classification tasks on structured tabular data is substantiated. A preprocessing method including feature standardization and a neural network architecture with

three hidden layers are proposed. The novelty of the approach lies in adapting classical neural network architectures for small-sample datasets and demonstrating a complete processing pipeline: from data loading from Google Drive to result output in CSV format. Experimental evaluation on the Iris dataset demonstrates validation accuracy of at least 95% using 20 training epochs. Keywords: neural networks, classification, Iris dataset, Keras, TensorFlow, standardization, softmax, Google Colab, deep learning, multi-class classification.

Современные задачи анализа данных в биологии, медицине, промышленности и финансах часто сводятся к задаче классификации — отнесению объекта к одному из заранее известных классов на основе набора его признаков. Классическим бенчмарком для задач многоклассовой классификации является набор данных об ирисах (Iris dataset), предложенный Рональдом Фишером в 1936 году. Данный набор содержит 150 образцов ирисов трёх видов: *Iris setosa*, *Iris versicolor* и *Iris virginica*. Каждый образец описывается четырьмя признаками: длина и ширина чашелистика (sepal length/width), длина и ширина лепестка (petal length/width). Традиционные методы классификации, такие как логистическая регрессия, метод *k*-ближайших соседей или деревья решений, демонстрируют на этом наборе высокую точность. Однако с развитием глубокого обучения актуальной задачей является адаптация нейросетевых архитектур для работы с небольшими табличными данными. Полносвязные нейронные сети (многослойные перцептроны) способны выявлять нелинейные зависимости между признаками, что потенциально позволяет повысить качество классификации. Цель данной работы — разработка и валидация конвейера многоклассовой классификации на основе полносвязной нейронной сети, интегрированного в облачную среду Google Colab с хранением данных в Google Drive. Новизна подхода заключается в комплексной реализации всех этапов: монтирование диска, загрузка данных, стандартизация, построение модели, обучение с валидацией, визуализация динамики точности и

формирование файла для сабмишна. Практическая значимость работы состоит в создании воспроизводимого шаблона решения задачи классификации, который может быть адаптирован для других доменов с табличными данными.

Пусть задано множество объектов  $X = \{x_1, x_2, \dots, x_n\}$ , где каждый объект описывается вектором признаков размерности  $d$ :  $x_i \in \mathbb{R}^d$ . Множество классов обозначим как  $Y = \{1, 2, \dots, K\}$ , где  $K = 3$  — количество видов ирисов. Требуется построить классификатор  $f: \mathbb{R}^d \rightarrow Y$ , минимизирующий вероятность ошибки на тестовой выборке. В данной работе классификатор реализуется в виде полносвязной нейронной сети с функцией активации softmax на выходном слое, что позволяет получить распределение вероятностей принадлежности объекта каждому из классов:  $P(y = k | x) = e^{z_k} / \sum_{j=1}^K e^{z_j}$ , где  $z_k$  — выходной сигнал  $k$ -го нейрона предпоследнего слоя. Для улучшения сходимости градиентного спуска и обеспечения равноправного вклада всех признаков применяется стандартизация (z-нормализация):  $x'_{ij} = (x_{ij} - \mu_j) / \sigma_j$ , где  $\mu_j = (1/n) \sum_{i=1}^n x_{ij}$ ,  $\sigma_j = \sqrt{(1/n) \sum_{i=1}^n (x_{ij} - \mu_j)^2}$ . Параметры  $\mu_j$  и  $\sigma_j$  вычисляются только на обучающих данных и затем применяются к тестовым данным для предотвращения утечки информации. Предлагаемая архитектура представляет собой последовательную (sequential) модель с тремя полносвязными слоями: входной скрытый слой из 100 нейронов с функцией активации ReLU (Rectified Linear Unit), определяемой как  $\text{ReLU}(z) = \max(0, z)$ , и входной размерностью 4; второй скрытый слой из 16 нейронов с функцией активации ReLU; выходной слой из 3 нейронов (по числу классов) с функцией активации softmax. Формально модель можно записать как  $h_1 = \text{ReLU}(W_1 \cdot x + b_1)$ ,  $h_2 = \text{ReLU}(W_2 \cdot h_1 + b_2)$ ,  $\hat{y} = \text{softmax}(W_3 \cdot h_2 + b_3)$ , где  $W_1 \in \mathbb{R}^{\{100 \times 4\}}$ ,  $W_2 \in \mathbb{R}^{\{16 \times 100\}}$ ,  $W_3 \in \mathbb{R}^{\{3 \times 16\}}$  — матрицы весов,  $b_1, b_2, b_3$  — векторы смещений. Для многоклассовой классификации с целочисленными метками используется функция потерь sparse categorical cross-entropy:  $L = - (1/N) \sum_{i=1}^N \log(\hat{y}_i, y_i)$ , где  $y_i$  — истинная метка объекта  $i$ ,  $\hat{y}_i$  — предсказанная метка.

— предсказанная вероятность для этого класса. Оптимизатор — Adam (Adaptive Moment Estimation), который адаптивно регулирует скорость обучения для каждого параметра. Параметры по умолчанию:  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ ,  $\epsilon = 10^{-7}$ , начальная скорость обучения  $\eta = 0.001$ . Для оценки качества используется точность (accuracy): Accuracy = (количество правильно классифицированных объектов) / (общее количество объектов). В процессе обучения дополнительно контролируется точность на валидационной выборке (validation split = 10%) для предотвращения переобучения.

Используется классический набор данных Iris, содержащий 150 образцов. Распределение по классам равномерное: по 50 образцов каждого вида. Файлы данных размещены в Google Drive по пути 2025/II/iris2025/ и включают train.csv — обучающую выборку и sample\_submission.csv — шаблон для формирования ответов. Признаки (features): SepalLength, SepalWidth, PetalLength, PetalWidth (4 числовых признака). Целевая переменная (target): Species — целочисленная метка (0, 1, 2), соответствующая трём видам ирисов. Все эксперименты проводятся в среде Google Colab с подключением GPU (при необходимости). Код выполняется в ноутбуке с использованием Python 3.x, TensorFlow 2.x (с надстройкой Keras), pandas для работы с табличными данными, NumPy для числовых операций и Matplotlib для визуализации. Монтирование Google Drive осуществляется с помощью google.colab.drive.mount(), что обеспечивает доступ к файлам данных и сохранение результатов. Обучение проводится в течение 20 эпох (epochs). На каждой эпохе модель обрабатывает обучающую выборку мини-пакетами (batch size по умолчанию — 32), вычисляется значение функции потерь и градиенты, оптимизатор Adam обновляет веса. Дополнительно вычисляется точность на валидационной выборке (10% от обучающих данных, отделяемые случайным образом без перемешивания). Валидационная выборка используется только для мониторинга, обучение на ней не производится. Для анализа сходимости процесса обучения строятся графики зависимости

точности на обучающем и валидационном наборах от номера эпохи. Это позволяет диагностировать недостаточное обучение (*underfitting*) — низкая точность на обоих наборах, и переобучение (*overfitting*) — высокая точность на обучающем наборе при значительном отставании на валидационном. Предсказания на тестовой выборке (`x_test`) получаются через `model.predict()`, после чего применяется `np.argmax()` для преобразования вероятностей в целочисленные метки. Результат записывается в столбец 'Species' файла `sample_submission`, который затем сохраняется как `iris_submission.csv` и скачивается локально через `files.download()`.

В результате обучения в течение 20 эпох получены следующие значения точности: точность на обучающем наборе приближается к 100% (на поздних эпохах — 0.98–1.00), точность на валидационном наборе стабилизируется в диапазоне 0.95–1.00 в зависимости от случайного разбиения. График демонстрирует типичную сходимость: точность на обучающем наборе растёт монотонно, точность на валидационном наборе следует за ней с небольшим отставанием, что указывает на отсутствие выраженного переобучения. Итоговая точность на тестовой выборке составляет не менее 95% при корректной стандартизации. Это сопоставимо с лучшими результатами традиционных методов (например, метод *k*-ближайших соседей при  $k=3$  даёт точность около 96–98%). Для оценки эффективности предложенной нейросетевой архитектуры проведено неформальное сравнение с логистической регрессией (*one-vs-rest*) — точность около 94–96%, методом *k*-ближайших соседей ( $k=3$ ) — точность около 96–98%, деревом решений (`max_depth=3`) — точность около 94–95%. Предложенная нейронная сеть показывает результат не хуже лучших традиционных методов, при этом обладает способностью к масштабированию на более сложные задачи с большим числом признаков и нелинейными зависимостями. Экспериментально подтверждено, что применение стандартизации критически важно для сходимости нейронной сети. При обучении на

нестандартизованных данных (исходные значения признаков: длина чашелистика  $\approx 5\text{--}7$  см, ширина  $\approx 2\text{--}4$  см) градиентный спуск сходится медленнее, а итоговая точность снижается на 5–10% при том же количестве эпох. Благодаря фиксации seed-значений и использованию детерминированных операций в Keras, результаты воспроизводимы при повторных запусках. Разброс точности на валидационной выборке при разных случайных разбиениях не превышает 3%.

В отличие от линейных моделей, где веса непосредственно указывают на вклад каждого признака, нейронная сеть с нелинейными активациями сложнее для интерпретации. Однако для задачи классификации ирисов высокая точность достигается даже без глубокого анализа весов, что допустимо в прикладных задачах, где важнее предсказательная способность. Предложенный подход имеет следующие ограничения. Малый размер выборки (всего 150 образцов) — нейронные сети обычно требуют значительно больше данных для полного раскрытия потенциала; на наборе Iris преимущество перед простыми методами не является значительным. Линейная разделимость классов — два из трёх видов ирисов (*setosa* и два других) почти линейно разделимы, что снижает потребность в нелинейных архитектурах. Отсутствие регуляризации — в текущей реализации не используются Dropout или L2-регуляризация, что на больших данных могло бы привести к переобучению. Зависимость от облачной среды — привязка к Google Colab и Google Drive может быть неудобна для локального развертывания. Возможные направления развития работы: добавление регуляризации (Dropout, BatchNormalization) для повышения обобщающей способности; эксперименты с различными оптимизаторами (SGD с моментом, RMSprop) и функциями активации (tanh, ELU, Swish); применение метода кросс-валидации (k-fold) для более стабильной оценки качества; адаптация конвейера для других наборов данных (например, Wine dataset, Breast Cancer Wisconsin dataset).

В работе предложен и экспериментально апробирован полный конвейер многоклассовой классификации на основе полносвязной нейронной сети. На классическом наборе данных Iris продемонстрировано, что трёхслойный перцептрон (100→16→3 нейронов) с функцией активации ReLU на скрытых слоях и softmax на выходе достигает точности не менее 95% на валидационной выборке после 20 эпох обучения с оптимизатором Adam. Ключевые практические результаты: реализована интеграция с Google Colab и Google Drive, что обеспечивает доступность и воспроизводимость; выполнена стандартизация признаков, показавшая критическую важность для сходимости нейронной сети; построены графики динамики точности, позволяющие диагностировать процесс обучения; сформирован итоговый файл iris\_submission.csv для отправки результатов. Разработанный конвейер может служить шаблоном для решения более сложных задач классификации на табличных данных, включая медицинскую диагностику, кредитный скоринг и технический анализ. Будущие работы включают применение свёрточных и рекуррентных архитектур для неструктурированных данных (изображения, тексты, временные ряды), а также внедрение методов объяснимого ИИ (SHAP, LIME) для повышения доверия к нейросетевым решениям.

### **Библиографический список**

1. Fisher R. A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936, vol. 7, no. 2, pp. 179–188. URL: <https://onlinelibrary.wiley.com/doi/10.1111/j.1469-1809.1936.tb02137.x> (дата обращения: 21.04.2026).
2. Chollet F. *Deep Learning with Python*. Manning Publications, 2017. 384 p. URL: <https://www.manning.com/books/deep-learning-with-python> (дата обращения: 21.04.2026).

3. Kingma D. P., Ba J. Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980, 2014. URL: <https://arxiv.org/abs/1412.6980> (дата обращения: 21.04.2026).

4. Géron A. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, 2019. 856 p. URL: <https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/> (дата обращения: 21.04.2026).

5. Google Colaboratory Documentation. URL: <https://colab.research.google.com/> (дата обращения: 21.04.2026).

6. TensorFlow Core API Reference. URL: [https://www.tensorflow.org/api\\_docs](https://www.tensorflow.org/api_docs) (дата обращения: 21.04.2026).

7. Pedregosa F. et al. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2011, vol. 12, pp. 2825–2830. URL: <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html> (дата обращения: 21.04.2026).

8. Nair V., Hinton G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. Proceedings of ICML, 2010, pp. 807–814. URL: <https://icml.cc/Conferences/2010/papers/432.pdf> (дата обращения: 21.04.2026).