

## РАЗРАБОТКА ПЛАТФОРМЫ ДЛЯ СОЗДАНИЯ НЕЙРОННЫХ СЕТЕЙ НА ЯЗЫКЕ ПРОГРАММИРОВАНИЯ JAVA

**А.Ю. Близниченко**

[bliznichenko.aleksey@mail.ru](mailto:bliznichenko.aleksey@mail.ru)

*ГБОУ ЦДО «Малая академия наук», Севастополь, Российская Федерация*

Искусственный интеллект (ИИ) играет большую роль в современном мире и продолжает набирать популярность, однако существующие сервисы с его использованием имеют ряд недостатков и зачастую подвержены политике санкций, из-за чего могут оказаться недоступны. Разработка подобных систем и моделей ИИ требует больших знаний в математике и программировании, что делает ее недоступной для большинства заинтересованных пользователей, а популярные библиотеки для работы с нейронными сетями также могут оказаться вне доступа российских пользователей. В статье описан процесс разработки создания отечественной платформы, позволяющей создавать, настраивать и обучать нейронные сети с возможностью их дальнейшего применения в практических целях. Предполагается, что такая платформа в будущем будет использоваться в образовательных целях для ознакомления новых пользователей с основными понятиями и принципами работы полносвязной нейронной сети.

**Ключевые слова:** нейронная сеть, платформа-редактор, гиперпараметры, архитектура нейронной сети, пользовательский интерфейс, классификация данных, машинное обучение

**Введение.** Главные успехи искусственного интеллекта (ИИ) в последнее время связаны с нейронными сетями. Нейронная сеть — математическая модель, а также ее программное или аппаратное воплощение, построенная по принципу организации и функционирования биологических нейронных сетей — сетей нервных клеток живого организма (в частности, мозга) [1].

Базовые задачи, для решения которых могут использоваться нейросети, — это классификация, регрессия, прогнозирование временных рядов, кластеризация, генерация различного контента. При этом нейронные сети во многих случаях справляются с этими задачами лучше, чем другие методы машинного обучения, что оправдывает сложность их разработки, обучения и внедрения [1–3].

**Цель работы** — разработать платформу ANNA с открытым исходным кодом без применения сторонних библиотек для создания, обучения и использования нейронных сетей прямого распространения (FFNN).

Подавляющее большинство сервисов для работы с нейронными сетями имеют закрытую платформу, платный доступ; модели ИИ изначально ориентированы на решение узкой задачи, при этом они зачастую созданы ино-

странными разработчиками и в любой момент могут оказаться недоступными, например, из-за политики санкций. Кроме того, разработчикам нейронных сетей необходим большой объем знаний в программировании и математике [1, 3].

Разрабатываемая программа будет полезна малому и среднему бизнесу, а также студентам вузов для решения основных задач, связанных с большими данными: прогнозированием и классификацией. В будущем планируется использовать проект в образовательных целях, чтобы ознакомить пользователя с понятием и работой полносвязной нейросети.

**Разработка приложения. Ресурсы и планирование работы.** Приложение написано на языке Java (Open JDK 17) с помощью системы сборки Gradle 8 с использованием графической библиотеки JavaFX и библиотеки JUnit 5 для юнит-тестов [4, 5]. Для написания кода использовалась среда IDE IntelliJ IDEA Ultimate, для создания пользовательского интерфейса — инструмент разработки Scene Builder. Приложение доступно на облачной платформе GitHub и в дальнейшем будет развиваться с привлечением заинтересованных участников как программное обеспечение с открытым программным кодом (open source проект).

Проект имеет деление на два модуля — ответственный за вычисления и работу нейронной сети (модуль ANNA — Advanced Neural Network Application) и ответственный за отображение пользовательского интерфейса и образовательную составляющую (модуль ELENNА — Educational Learning Engine for Neural Network Application). В будущем планируется полностью выделить модуль ANNA в отдельную библиотеку для удобства разработки и более универсального использования. На данный момент название ANNA применимо как и к конкретному модулю, так и ко всей платформе в целом.

**Описание алгоритма работы и приложения.** Нейросеть представляет собой пять ключевых и четыре дополнительных класса: главный класс (NeuralNetwork), архитектуру нейронной сети и связанные с ней данные (NetworkStructure), наборы данных (DataMaster), единичный нейрон (Neuron), имеющий две реализации для стандартного нейрона и нейрона смещения, и класс с гиперпараметрами (Hyperparameters). К дополнительным относятся классы с функциями активации, расчета ошибки, обучения и метрик. Такое разбиение способствует как лучшей читаемости кода, так и куда более простой масштабируемости.

За исключением основной логики (нейросети), присутствуют еще восемь классов, ответственных за работу модуля с графическим интерфейсом, и три вспомогательных класса, которые отвечают за работу со считыванием данных, вывод ошибок и структурирование данных.

После ввода необходимых настроек и нажатия на кнопку запуска происходит сбор всех параметров из графического интерфейса в единую структуру данных (NetworkArguments). Далее она передается в NeuralNetwork, где создается архитектура сети, состоящая из двух ключевых переменных: списка смежности и таблицы структуры. В дальнейшем для удобства пользования класс структуры NetworkStructure служит только местом хранения данных.

После создания структуры и необходимых таблиц данных запускается сама нейросеть. Ее алгоритм представляет собой три вложенных цикла, в которых происходят все вычисления. Первый цикл итерирует по количеству эпох, другие два — по обучающему и тестовому наборам данных.

Алгоритм работы нейронной сети включает следующие пункты [1–3].

1. Создание массива (вектора) с желаемыми (правильными) ответами.
2. Вычисление значений выходных нейронов на основе текущих значений из набора данных:

- 2.1) передача значений во входные нейроны;

- 2.2) передача значений в скрытые нейроны, умножение на веса (связи между нейронами из текущего и следующего слоя). Сумма полученных значений преобразуется с помощью функции активации нейрона;

- 2.3) повтор шагов 2.1–2.2 до достижения выходного слоя;

- 2.4) сбор выходных значений в массив (вектор).

3. Вычисление ошибки (отклонения выходных значений от правильных).

4. Обучение методом обратного распространения ошибки и методом подсчета дельт:

- 4.1) вычисление дельт весов от конца нейронной сети к началу на основе градиентного спуска;

- 4.2) обновление весов.

5. Вычисление средней ошибки эпохи, обновление метрик.

**Пользовательский интерфейс.** При запуске приложения пользователь оказывается в главном меню, содержащем кнопки для настройки, ознакомления или начала работы с платформой; выпадающий список с выбором языка (на данный момент доступны русский и английский); описание последних изменений в приложении.

При нажатии кнопки «Создать новую нейросеть» пользователь получает возможность выбрать режим создания нейронной сети — упрощенный или продвинутый. Последний включает в себя полный функционал по настройке и обучению нейросети. В связи со сложной структурой этого режима планируется создать упрощенный режим, в котором в более примитивной форме будет можно создать простую нейронную сеть для ознакомления.

В разработке находятся вкладки «Настройки» и «Обучение», где планируется разместить некоторые параметры самой платформы (например, выбор темы пользовательского интерфейса), несколько образовательных видеороликов по теме нейронных сетей и интерактивные указания для обучения работе с платформой.

Редактор нейронной сети содержит четыре вкладки: «Данные», «Архитектура», «Управление» и «Результаты». Первые три ответственны за настройку самой нейросети, последняя — за вывод результатов обучения. Более подробно о них рассказано далее. В следующих пунктах описана работа приложения в продвинутом режиме редактора.

**Модуль загрузки данных.** Во вкладке «Данные» (рис. 1) пользователю необходимо загрузить отдельно обучающую и тестовую выборку или единый набор данных, который будет разбит на выборки по заданному соотношению. При загрузке данных они будут отображаться в таблицах для удобства пользования. Также реализована возможность предпросмотра разделения единого набора данных.

| #  | V1        | V2        | V3        | V4        | V5        | Class |
|----|-----------|-----------|-----------|-----------|-----------|-------|
| 0  | 0.533823  | -0.189665 | -1.668229 | -1.249093 | -0.685406 | 1     |
| 1  | -0.660158 | -0.141079 | 2.535399  | -1.544766 | -0.136583 | 1     |
| 2  | -0.757811 | -0.176658 | 1.016982  | -1.107165 | -0.486269 | 1     |
| 3  | -0.75511  | -0.160173 | 0.880969  | -1.090817 | -0.491231 | 1     |
| 4  | -0.565511 | 0.453241  | 0.829977  | 0.300258  | -0.136583 | 1     |
| 5  | 0.000655  | 0.968564  | -0.295023 | -0.145701 | -0.467262 | 1     |
| 6  | 0.134145  | 2.067209  | -1.349052 | -0.208018 | 0.873025  | 1     |
| 7  | 1.85735   | -0.032825 | -1.129496 | -0.336295 | 0.346117  | 1     |
| 8  | -0.440401 | 0.198863  | 1.516576  | -1.263376 | 0.579047  | 1     |
| 9  | -0.876483 | -1.132677 | -0.453805 | -1.279776 | -0.136583 | 1     |
| 10 | 2.148826  | -0.746456 | -0.346848 | 0.820393  | -0.648817 | 1     |
| 11 | -0.477988 | 0.361847  | 0.68639   | 0.2188    | -0.931391 | 1     |
| 12 | -0.448121 | -0.338988 | 2.134422  | 1.119478  | -0.854611 | 1     |
| 13 | -0.785096 | -0.661448 | 1.066051  | -0.505997 | -0.136583 | 1     |
| 14 | -0.352402 | -0.474365 | 1.910317  | -1.298249 | -1.084729 | 1     |
| 15 | -0.358629 | 1.790906  | -1.391134 | -0.067343 | -0.407463 | 1     |

Рис. 1. Вкладка «Данные» с демонстрационным набором данных 5\_parameters

В настоящей версии приложения поддерживаются данные числового (NUMBER) и категориального (CATEGORICAL) типа, файл должен иметь расширение .csv. Иные рекомендации по предобработке данных отображаются во всплывающей подсказке при наведении курсора на соответствующую иконку. При неверной загрузке данных или критических ошибках в них при-

ложение уведомит об этом пользователя окном с предупреждением и иконкой крестика при смене вкладки.

**Модуль создания архитектуры нейронной сети.** На вкладке «Архитектура» (рис. 2) пользователь получает доступ к настройкам архитектуры нейронной сети. В верхнем поле будет можно выбрать задачу нейронной сети, на данный момент поддерживается только классификация.

Ниже находится настройка входного слоя, которая по умолчанию происходит автоматически, однако имеется возможность настроить его вручную, например, если набор данных не соответствует рекомендациям. Для каждого входного нейрона необходимо настроить параметр (столбец из набора данных) и указать тип его данных. В правой части имеется поле для настройки целевого параметра — параметра, который необходимо классифицировать или спрогнозировать.

Далее внизу расположена настройка архитектуры нейронной сети. Можно добавить или удалить конкретный слой, настроить количество нейронов в нем и используемую функцию активации. Поддерживаются следующие функции активации нейрона: сигмоида, гиперболический тангенс, вариации линейных функций (ReLU, ELU, LeakyReLU) [1]. Для удобства представлена визуализация построенной нейросети, где число зеленых кружков соотносится с числом входных нейронов, красных — с числом уникальных классов в целевом параметре (в режиме прогнозирования выходной нейрон один), а синих с нейронами в скрытых слоях. Для будущей реализации сверточных нейронных сетей предусмотрен выбор типа слоя.

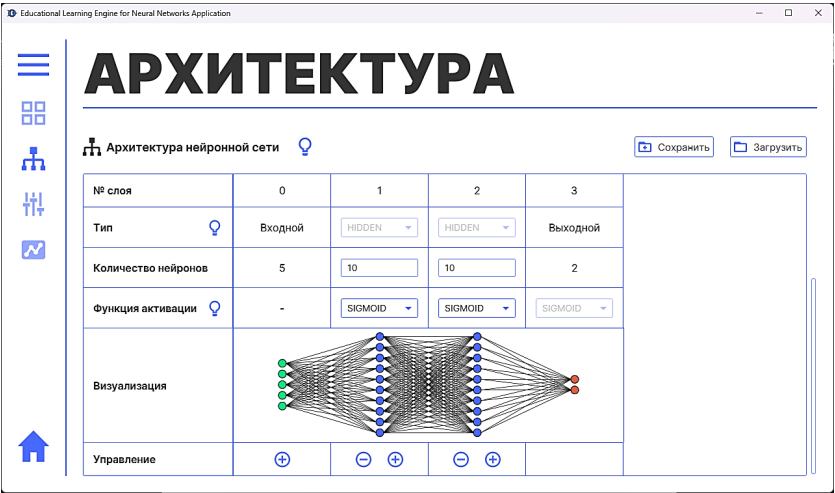


Рис. 2. Вкладка «Архитектура» с настроенной структурой нейронной сети

Для корректной работы необходимо выбрать минимум один входной нейрон и настроить целевой параметр. Максимальное число слоев и нейронов равняется пределу переменной типа Integer ( $2^{31} - 1$ , или 2 147 483 647), но на практике это не проверялось, поскольку использовать платформу для вычислений такого масштаба не предполагается.

**Модуль управления.** Во вкладке «Управление» (рис. 3) существуют три области, отвечающие за сохранение и загрузку нейронной сети, запуск обучения и настройку гиперпараметров.

**УПРАВЛЕНИЕ**

Сохранить / Загрузить нейросеть

Сохранить Загрузить

Сохранять статистику каждую 10 эпоху. ☒ Автоматически открыть результаты

Начать обучение сети

Настройка гиперпараметров

| Название                            | Значение                 | Описание                                                                                                                                                                                                                                               |
|-------------------------------------|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Количество эпох                     | 1000                     | Сколько раз должна нейронная сеть обучиться на входном файле. Напрямую влияет на длительность и точность обучения. Слишком большие значения могут привести к переобучению, а слишком маленькие к недостаточному обучению.                              |
| Функция расчета ошибки              | MSE                      | Метод, при помощи которого вычисляется отклонение полученного значения от целевого. MSE -- среднеквадратичная ошибка (Mean Squared Error) RMSE -- корень из среднеквадратичной ошибки (Root Mean Squared Error) Arctan -- арктангенс                   |
| Размер входных данных в эпохе       | 0                        | Размер подаваемых нейронной сети данных в строках. 0 -- считать весь файл.                                                                                                                                                                             |
| [WIP] Использовать нейроны смещения | <input type="checkbox"/> | Добавить к структуре нейронной сети нейроны смещения (смещение). Смещение добавляет к каждому слою (исключая входной) нейрон со значением 1. Это помогает сдвигать функцию активации по оси X и потенциально увеличить захватываемую область значений. |

Рис. 3. Вкладка «Управление»

С помощью корректировки гиперпараметров в таблице пользователь получает возможность настроить работу созданной нейросети [1, 3]. Для всех гиперпараметров имеется небольшое описание, позволяющее удобнее в них ориентироваться. На данный момент доступны следующие параметры:

- количество эпох обучения;
- функция расчета ошибки;
- размер выходных данных в эпохе;
- смещение;
- способ инициализации начальных весов;
- скорость обучения (Learning rate);
- момент (Momentum, «ускорение» градиентного спуска);

– наклон функции активации (для ELU и LeakyReLU).

Перед запуском обучения нейронной сети можно задать эпохи, на которых будут сохраняться метрики. Если все предыдущие поля были заполнены верно, то рядом с кнопкой «Начать обучение сети» появится зеленая галочка. В будущем планируется добавить возможность сохранения и загрузки как частей нейронной сети (ее архитектуры и гиперпараметров), так и обученной нейросети в целом. Также планируется реализовать экспорт модели в популярных форматах для ее дальнейшего использования в других библиотеках и приложениях.

**Модуль результатов работы.** Вкладка «Результаты» (рис. 4) имеет четыре основные части: график выбранной метрики, матрицы ошибок, статистику и «Симулятор». График имеет три метрики, просмотр которых выбирает пользователь: средняя ошибка, средняя точность (Ассигасу) и средняя F-мера. График можно вывести в отдельное окно для удобства или для более детального просмотра.

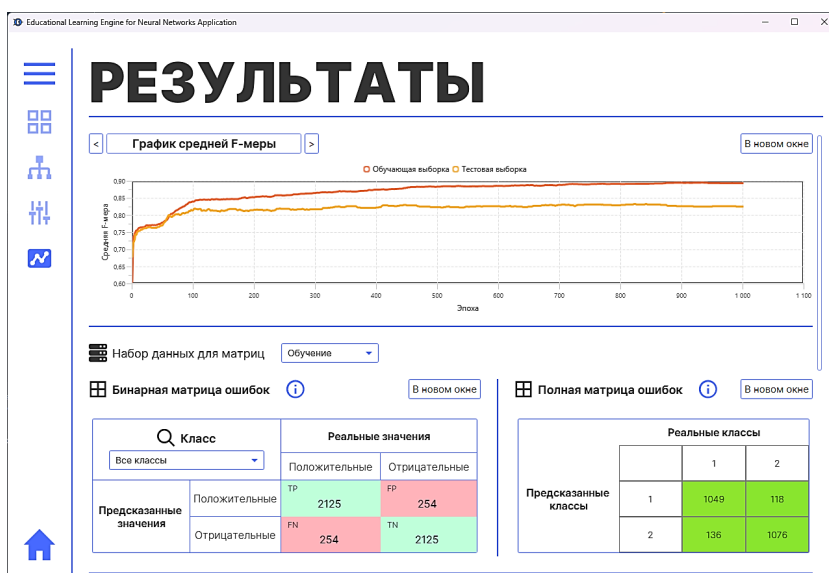


Рис. 4. Вкладка «Результаты» для обученной модели, верхние блоки

Матрица ошибок представлена в двух вариациях: бинарная и полная матрица. Пользователь может выбрать, для каких данных отображаются значения (обучающие или тестирующие). Для бинарной матрицы также можно выбрать класс, для которого будут отображены значения. Обе матрицы можно открыть в отдельном окне.

Вкладку можно пролистать вниз (рис. 5), где расположен список со статистикой обучения и тестирования. На данный момент для режима классификации представлены следующие метрики: ошибка последней эпохи (из гиперпараметров), размер данных, число верно определенных классов, Accuracy, Precision, полнота (Recall) и F1-мера. В правой части отображается общая оценка, измеряемая в единицах от 0 до 100. Она является субъективной метрикой автора и ее следует использовать только в ознакомительных целях. Ее расчет проводится на основе средних значений иных метрик (ошибка, Accuracy и F1-мера).

В нижней части вкладки расположен «Симулятор», позволяющий протестировать последнюю обученную нейросеть на произвольных значениях. Пользователю необходимо ввести входные данные для каждого входного нейрона и нажать кнопку «Рассчитать», после чего будет выведен предполагаемый нейронной сетью класс или значение.

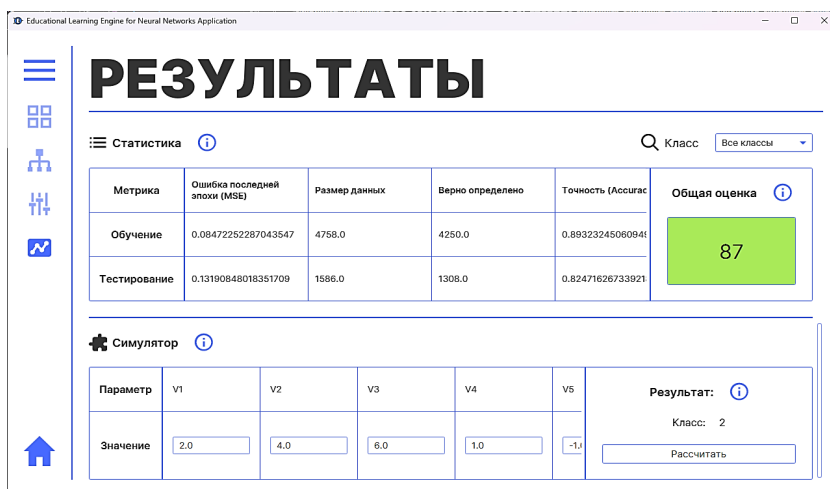


Рис. 5. Вкладка «Результаты» для обученной модели, нижние блоки

**Использование платформы.** В настоящее время существует множество разработок для взаимодействия с нейронными сетями в виде библиотек, веб-приложений, облачных сервисов, десктопных платформ. В качестве примеров таких решений можно привести следующие программные продукты.

1. Yandex DataSphere. Облачный сервис для машинного обучения. Пользователи должны иметь достаточно высокий уровень подготовки в сфере машинного обучения, чтобы начать разбираться в сервисе; кроме того, использование этого сервиса требует больших финансовых затрат.

2. Google Teachable Machine. Платформа для машинного обучения, которая позволяет пользователю самостоятельно обучать нейросеть. В некоторых регионах доступ к этой платформе ограничен из-за политики санкций.

3. Библиотеки для Python:

3.1) TensorFlow — библиотека от Google, разработанная специально для обучения нейросетей. Для обработки больших объемов данных в ней используется многоуровневая система узлов, что расширяет сферу применения;

3.2) Keras — высокоуровневая библиотека, поддерживающая сети с обратными связями и сети прямого распространения. Позволяет использовать возможности библиотеки TensorFlow в качестве компонентов. Характеризуется минималистичным подходом к дизайну и хорошей расширяемостью.

Для использования этих библиотек необходим опыт в программировании. Также они могут оказаться недоступными.

4. Универсальная платформа RapidMiner — открытая и расширяемая платформа анализа данных, позволяет полноценно внедрить методы машинного обучения и технологии ИИ на предприятии. Программное обеспечение RapidMiner включает подготовку данных для машинного обучения, визуализацию результатов, валидацию моделей, развертывание прогнозных моделей и оптимизацию моделей.

Бесплатная версия платформы имеет широкий функционал, связанный с обработкой и анализом данных. При этом возможность настройки нейронных сетей для глубокого обучения отсутствует.

Предлагаемое решение имеет свои преимущества по сравнению с аналогами.

1. В отличие от сервиса Yandex DataSphere, платформа ANNA является бесплатной и не требует от пользователя специальной подготовки.

2. По сравнению с платформой Google Teachable Machine платформа ANNA является отечественной разработкой, ее доступность не зависит от зарубежных санкций.

3. В отличие от библиотеки TensorFlow, для пользования платформой ANNA не требуются специальные знания в области программирования.

4. В отличие от платформы RapidMiner платформа ANNA предоставляет широкие возможности для работы с нейронными сетями.

**Заключение.** Многие существующие инструменты в области нейронных сетей имеют платный доступ и закрытую платформу и/или изначально ориентированы на конкретную задачу. Более универсальные и сложные сети требуют от пользователя хорошего знания в областях программирования, математики и машинного обучения. Подавляющее большинство подобных сервисов созданы иностранными разработчиками, поэтому в любой

момент могут оказаться недоступными в России, например, из-за политики санкций. Поэтому разработка в этой области является актуальной и перспективной.

Платформа ANNA распространяется по принципу Open Source, т. е. исходный код и исполняемый файл работы доступны для просмотра, изучения и предложения изменений. URL: <https://github.com/ravik-games/AdvancedNeuralNetworkApplication>

Дальнейшее развитие проекта будет преследовать две ключевые цели: выделение нейронной сети в отдельную отечественную библиотеку и разработка образовательного контента для ознакомления новых пользователей и заинтересованных в работе с нейронными сетями.

1. Библиотека для работы с нейронными сетями. Модульная структура разработки платформы позволяет в будущем выделить все компоненты, связанные с работой и созданием нейронной сети (модуль ANNA), в отдельную библиотеку для языка Java. Это позволит упростить разработку отечественных интеллектуальных систем и предоставить в общий доступ новую полностью открытую библиотеку по работе с нейронными сетями, что может быть полезно малому и среднему бизнесу в решение некоторых прикладных задач, связанных с классификацией и прогнозированием. Также новый инструментарий будет отвечать запросам импортозамещения в Российской Федерации.

2. Разработка образовательной платформы ELENNА. Предполагается, что подобная платформа будет использоваться в центрах дополнительного образования и на начальных курсах в вузах для ознакомления пользователей с принципами и построением работы полносвязной нейронной сети. Приложение будет предоставлять удобный инструмент для обучения созданию простых сетей, изучению их параметров и их влияния на работу нейросети и решения небольших прикладных задач. Для достижения поставленной цели будет создано «интерактивное обучение», которое позволит ознакомиться с интерфейсом платформы, ее работой и научит создавать и настраивать простейшие сети. Также планируется добавить поддержку сверточных нейронных сетей, что позволит классифицировать изображение, расширит возможные применения и сделает процесс обучения более наглядным. При необходимости возможно добавление других видов нейросетей, например рекуррентных, и перенос платформы в браузерную версию, однако на данный момент это признано нецелесообразным.

## Литература

- [1] Гудфеллоу Я., Бенджио И., Курвилль А. *Глубокое обучение*. Москва, ДМК Пресс, 2018, 653 с.
- [2] Миркес Е. М. *Нейрокомпьютер. Проект стандарта*. Новосибирск, Наука, Сибирская издательская фирма РАН, 1998, 190 с.
- [3] Хайкин С. *Нейронные сети: полный курс*. Москва, Вильямс, 2006, 1104 с.
- [4] Блох Д. *Java. Эффективное программирование*. Москва, Лори, 2002, 224 с.
- [5] Седжвик Р., Уэйн К. *Алгоритмы на Java*. Москва, Вильямс, 2013, 848 с.

*Поступила в редакцию 24.10.2024*

**Близниченко Алексей Юрьевич** — учащийся творческого объединения «Программирование на Java», ГБОУ ЦДО «Малая академия наук»; учащийся 11-го класса ГБОУ СОШ № 6, Севастополь, Российская Федерация.

**Научный руководитель** — Глеч Екатерина Викторовна, педагог дополнительного образования, руководитель творческого объединения «Программирование на Java», ГБОУ ЦДО «Малая академия наук», Севастополь, Российская Федерация.

**Ссылку на эту статью просим оформлять следующим образом:**

Близниченко А.Ю. Разработка платформы для создания нейронных сетей на языке программирования JAVA. *Политехнический молодежный журнал*, 2025, № 01 (96). URL: [https://ptsj.bmstu.ru/catalog/icec/inf\\_tech/1017.html](https://ptsj.bmstu.ru/catalog/icec/inf_tech/1017.html)

## DEVELOPMENT OF A PLATFORM FOR CREATING NEURAL NETWORKS IN THE JAVA PROGRAMMING LANGUAGE

**A.Yu. Bliznichenko**

bliznichenko.aleksey@mail.ru

*SBEI CAE “Junior Academy of Sciences”, Sevastopol, Russian Federation*

Artificial intelligence (AI) is playing an important role in the modern world and continues to gain popularity. However, the existing services using it have a number of drawbacks and are often subject to the sanction policies, which could make them unavailable. Development of such AI systems and models requires extensive knowledge in mathematics and programming, which makes them inaccessible to the most interested users; and popular libraries for working with the neural networks could also appear out of reach for the Russian users. The paper describes the process of developing a domestic platform that makes it possible to create, configure and learn the neural networks with a possibility of their further introduction for practical purposes. The paper assumes that such a platform would be used in the future for educational purposes to familiarize new users with the basic concepts and operation principles of a fully connected neural network.

**Keywords:** neural network, editor platform, hyper-parameters, neural network architecture, user interface, data classification, machine learning

---

***Received 24.10.2024***

**Bliznichenko A.Yu.** — Student, Creative Association “Programming in Java”, SBEI CAE “Junior Academy of Sciences”; 11<sup>th</sup> Grade Student, Secondary School No. 6, Sevastopol, Russian Federation.

**Scientific advisor** — Glech E.V., Teacher of Additional Education, Head of Creative Association “Programming in Java”, SBEI CAE “Junior Academy of Sciences”, Sevastopol, Russian Federation.

### **Please cite this article in English as:**

Bliznichenko A.Yu. Development of a platform for creating neural networks in the JAVA programming language. *Politekhnikheskiy molodezhnyy zhurnal*, 2025, no. 01 (96). (In Russ.). URL: [https://ptsj.bmstu.ru/catalog/icec/inf\\_tech/1017.html](https://ptsj.bmstu.ru/catalog/icec/inf_tech/1017.html)