

УДК 004.932.75

**В. В. Шефер, Е. В. Решетникова**

**V. V. Shefer, E. V. Reshetnikova**

Шефер Владимир Владимирович, студент 3 курса, ФИМЭ, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Решетникова Елена Васильевна, к. т. н., доцент, КГПИ ФГБОУ ВО «КемГУ», г. Новокузнецк, Россия.

Shefer Vladimir Vladimirovich, 3-year student, Kuzbass Humanitarian Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.  
Reshetnikova Elena Vasilievna, candidate of technical Sciences, associate Professor, Kuzbass Humanitarian Pedagogical Institute of Kemerovo State University, Novokuznetsk, Russia.

## **ОБУЧАЮЩАЯ ПЛАТФОРМА NEURONS.WEB**

## **EDUCATION PLATFORM NEURONS.WEB**

**Аннотация.** Статья посвящена разработке веб-приложения, предназначенного для обучения основам нейронных сетей. Использовались два фреймворка - Django и Tenserflow (в том числе библиотека Keras). Приложение содержит функционал для работы с четырьмя сетями разных архитектур и необходимые теоретические материалы о нейросетях, а также руководство пользователя для работы с приложением.

**Annotation.** The article is devoted to the development of a web application designed to teach people the basics of neural networks. Two frameworks were used - Django and Tenserflow (including the Keras library). The application contains functionality for working with neural networks of different architectures and the necessary information about them, neural networks in general, the correct use of the application.

**Ключевые слова:** нейронные сети, обучающая платформа, веб-приложение, Django, Keras.

**Keywords:** neural networks, education platform, web application, Django, Keras.

Всероссийский центр изучения общественного мнения (ВЦИОМ) и АНО «Национальные приоритеты» в 2021 году представили данные опроса россиян на тему отношения к искусственному интеллекту [1]. Одним из главных пунктов был вопрос о знании и понимании термина «искусственный интеллект»: подавляющее большинство россиян (81 %) знакомы с ним, но только 32 % могут объяснить его значение, а не знакомы с ним 18 % россиян. Причем 48 % заявили о своем доверии технологиям искусственного интеллекта (далее ИИ), не доверяют им 42 % респондентов, 10 % затруднились с ответом.

Так как нейронные сети являются одним из способов реализации искусственного интеллекта, то процентное соотношение знакомых с нейронными сетями и понимающих их можно считать равным или меньшим соотношения по вышеупомянутому опросу.

Именно поэтому создание обучающей платформы, которая поможет разобраться в нейронных сетях далеким от темы ИИ людям, будет *актуальным*.

*Цель* данной работы заключается в разработке доступного приложения, которое на практике показывает возможности нейронной сети, содержит теоретическую информацию, позволяющую восполнить пробелы в знаниях о нейросетях. Также приложение дает возможность дообучать сеть и просматривать результат на собственных примерах.

*Использование платформы в образовательных целях* направлено на ознакомление с нейронными сетями, их разновидностями, развитие понимания их устройства и математических основ, приобретение опыта работы с ними.

Приложение позволяет использовать нейронную сеть для распознавания рукописных цифр: рукописно ввести изображение для проверки, выбрать архитектуру нейросети, дополнительно обучать её. Также содержит теоретические сведения о нейронах, математических моделях, положенных в основу функционирования сетей. Приложение снабжено подробной инструкцией по его использованию.

### **Выбор фреймворка**

Важным для создания качественного приложения является выбор правильного фреймворка, позволяющего решить поставленные задачи.

У фреймворков есть две основные функции: работа на серверной стороне (бэкенд) и работа на клиентской стороне (фронтенд). Клиентская часть – это, в основном, HTML, CSS и Javascript, и рассматривается она как способ представления информации пользователям. Серверная часть, как правило, является наиболее сложной и описывает всю бизнес-логику приложения. Веб-разработку серверной части можно выполнить на многих языках, таких как Java, Python, PHP и так далее [2].

Архитектура почти всех фреймворков основана на декомпозиции нескольких отдельных слоёв (приложения, модули и т.д.), что означает, что разработчики могут расширять функциональность исходя из своих потребностей и использовать изменённую версию вместе с кодом фреймворка. Такая гибкость является ключевым преимуществом фреймворков наряду с простотой их использования [2, 3].

В качестве фреймворка был выбран Django – высокоуровневый фреймворк для веб-приложений на языке Python, предназначенный для быстрой и эффективной разработки [3]. Веб-системы на Django строятся из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и подключаемыми.

Фреймворк реализует архитектуру из трех компонентов MVT (Model-View-Template) – Модель, Представление и Шаблон [4] (рис. 1). Данная архитектура и была выбрана для данной образовательной платформы.

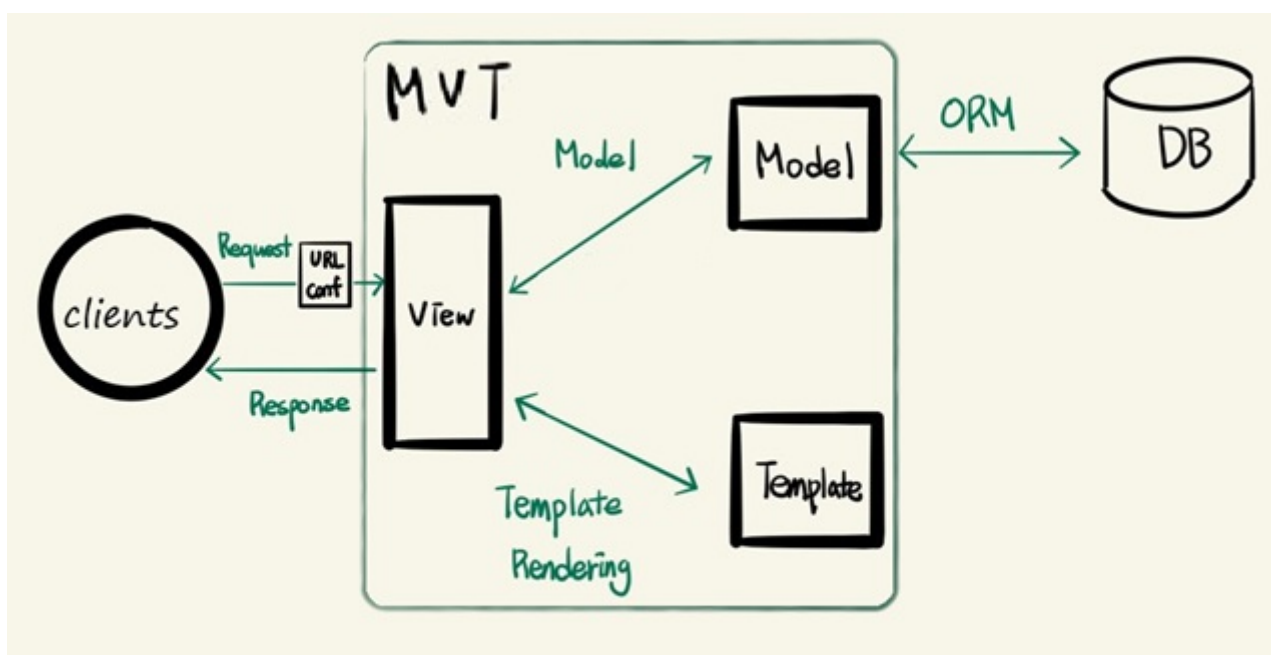


Рисунок 1. Архитектура MVT

Модель (model) представляет собой класс, который хранит в себе информацию. Такие классы нужны для описания бизнес-логики, методов, свойств и других элементов, необходимых для управления данными. Объектно-реляционное отображение (ORM) обеспечивает взаимодействие приложения с базами данных (БД). ORM автоматически передаёт данные из БД, например, PostgreSQL или MySQL, в объекты, которые используются в коде приложения. При обратном взаимодействии – объекты транслируются в БД. Информацию, хранимую в экземплярах модели, может получать и редактировать только один из слоёв – представление (view).

Представление (view) выполняет ряд задач: реализует логику приложения, получает HTTP-запросы (request), обрабатывает их и отправляет HTTP-ответы (response). Этот ответ может быть HTML-страницей, переадресацией, ошибкой или документом расширяемого языка разметки.

Шаблон (template) является файлом с HTML-кодом. Шаблон может отображать в себе как статические, так и динамические данные, но никогда не содержит в себе логику работы с моделями данных. Именно через шаблоны идет взаимодействие с представлениями путем отправки и приема запросов.

Также для управления запросами используется URL – указатели ресурсов. По заданным ссылкам запросы передаются определенным представлениям [3].

### **Реализация нейронных сетей**

Выбранный язык разработки на стороне сервера Python позволяет использовать иные фреймворки, в частности для работы с нейронными сетями. Один из них – фреймворк TensorFlow с библиотекой Keras, позволяющие быстро создать любую модель нейронной сети, причем сам алгоритм создания высокоуровневый. То есть сеть собирается по слоям, а не по отдельным нейронам, каждый слой представляет собой объект с параметрами количества нейронов, функцией активации и тому подобным [5]. Для создания сетей, реализованных в обучающей платформе, использовался следующий алгоритм:

1. Загрузка тренировочных и тестовых пар картинок и цифр.
2. Стандартизация данных выборки.
3. Конструирование модели НС по слоям.
4. Компиляция модели.
5. Обучение модели.
6. Тестирование модели.

С помощью Keras реализованы 4 нейросети: 2-слойная, 3-слойная, 4-слойная, сверточная. Они хранятся в виде файлов в формате HDF5, как модель .h5, которая содержит и структуру, и веса. К этим файлам обращается представление (view), при каждом запросе пользователя идет обработка изображения и его распознавание загруженной сетью. Такой механизм выбран по причине того, что создание новых нейронных сетей для каждой сессии (временной интервал, в течение которого происходит взаимодействие пользователя с сайтом) занимало бы от 20 до 120 секунд в зависимости от выбранной архитектуры. Основное время тратилось бы на обучение модели и ее тестирование. Хранение уже готовых сетей на сервере не вызывает данную проблему и реализует моментальное взаимодействие пользователя с ними.

Так как для работы нейросети требуется наличие изображения, а в качестве результата – одна из цифр, то в базе данных сохраняется объект модели (model) ImageInfo, содержащий в качестве параметров цифру и изображение. Каждый раз, когда пользователь запрашивает распознавание картинки, сама картинка и результат сохраняются с использованием ORM. Совокупность хранимых объектов можно использовать в дальнейшем как выборку для обучения других сетей (с учетом того, что не все варианты верные) [6].

### **Реализация интерфейса**

Для рукописного ввода на веб-странице и отправки данных формата json использовались JavaScript функции. Они описаны в шаблоне веб-страницы платформы.

Ввод производится движением мыши с зажатой левой кнопкой. На панели редактирования расположены три кнопки: Recognize – отправляет запрос на распознавание цифры, Clear – очищает поле, Redact – отправляет запрос на дообучение нейросети по данному изображению и выбранной в поле цифры (рис. 2).

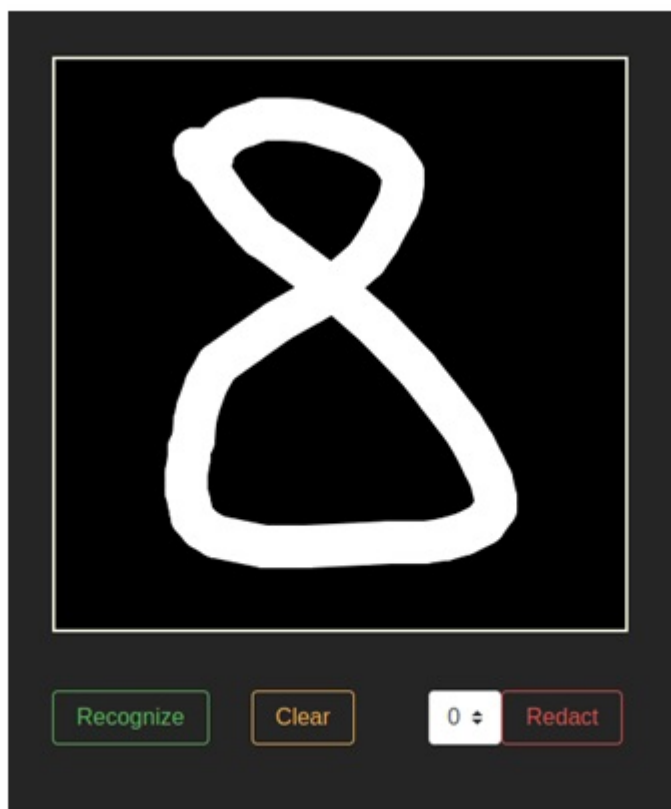


Рисунок 2. Панель редактирования рисунка

### **Дополнительное обучение нейронной сети пользователями**

Нейросети не работают на 100 % верно, особенно с теми объектами, которые можно интерпретировать по-разному. К таким относятся и рукописные цифры.

В качестве параметра оценки дообучения выбрано значение функции потерь. Чем это значение ниже – тем обучающая выборка ближе к предсказаниям нейросети.

Все четыре нейросети используют категориальную кроссэнтропию – довольно точную функцию потерь [6].

Для дискретного  $p$  (верное распределение) и  $q$  (полученное распределение) кроссэнтропия вычисляется следующим образом:

$$H(p, q) = - \sum_x p(x) \log q(px). \quad (1)$$

Для непрерывных распределений:

$$H(p, q) = \int_x p(x) \log q(x) dx. \quad (2)$$

Сам процесс дообучения модели на основе одного изображения и одной цифры занимает в среднем меньше секунды, что почти не заметно для пользователя.

## Теория и инструкция

В шаблон страницы были добавлены раскрывающиеся контейнеры с теоретическими данными и инструкцией пользователя программы (рис. 3).

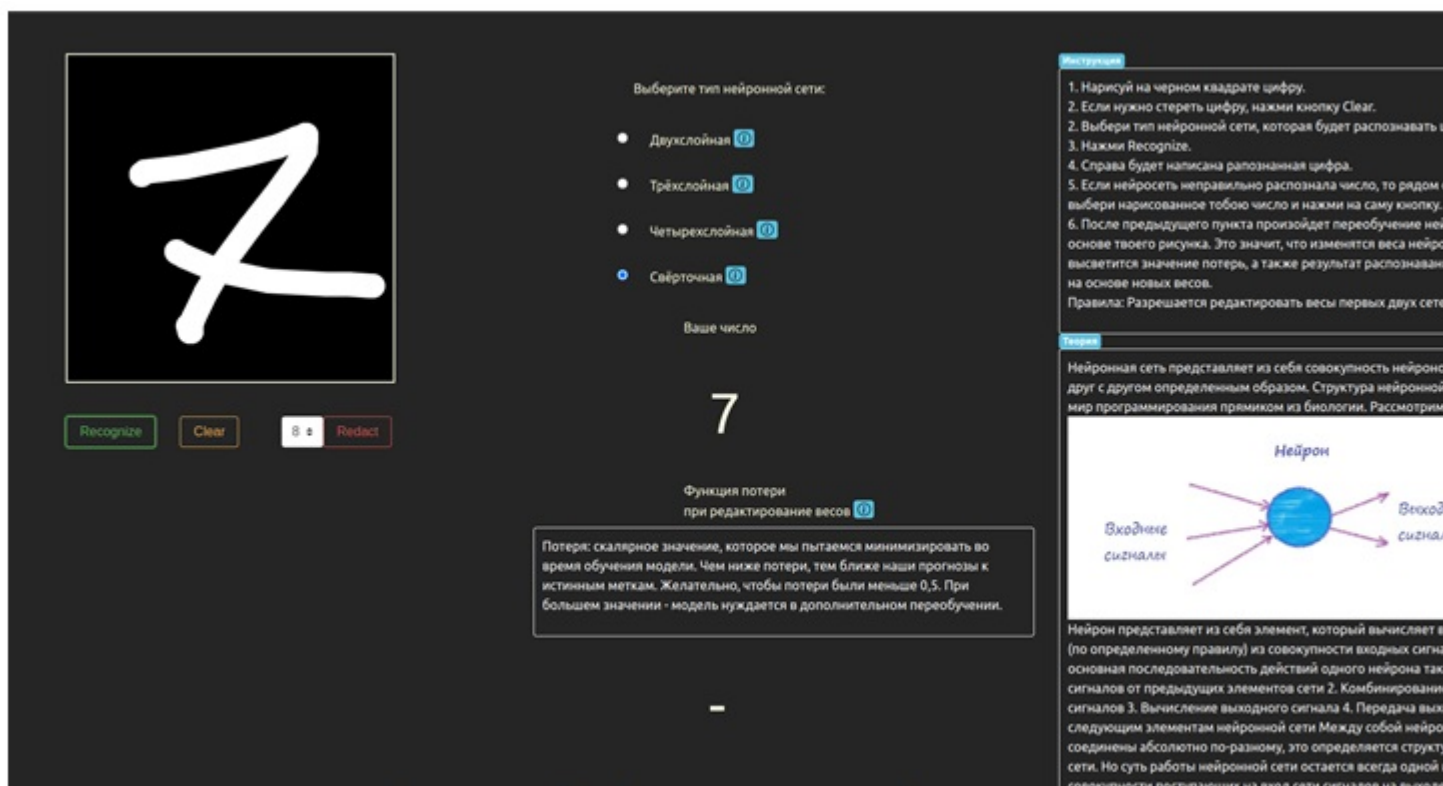


Рисунок 3. Скриншот веб-приложения

На рисунке 3 представлен результат распознавания рукописной цифры 7 сверточной сетью. В правой части окна раскрыты контейнеры с информацией, которая поможет пользователю расширить свой кругозор в теме нейронных сетей. Теоретические сведения покрывают большую часть вопросов, которые могут возникнуть при работе с нейросетями. Для каждой модели сети изображена ее схема и приведены теоретические материалы, например, на рисунке 4 представлен вариант схемы для сверточной модели. Теоретический материал, наполняющий контейнеры приложения, начинается с описания понятия «нейрон» и заканчивается математической интерпретацией нейросети.

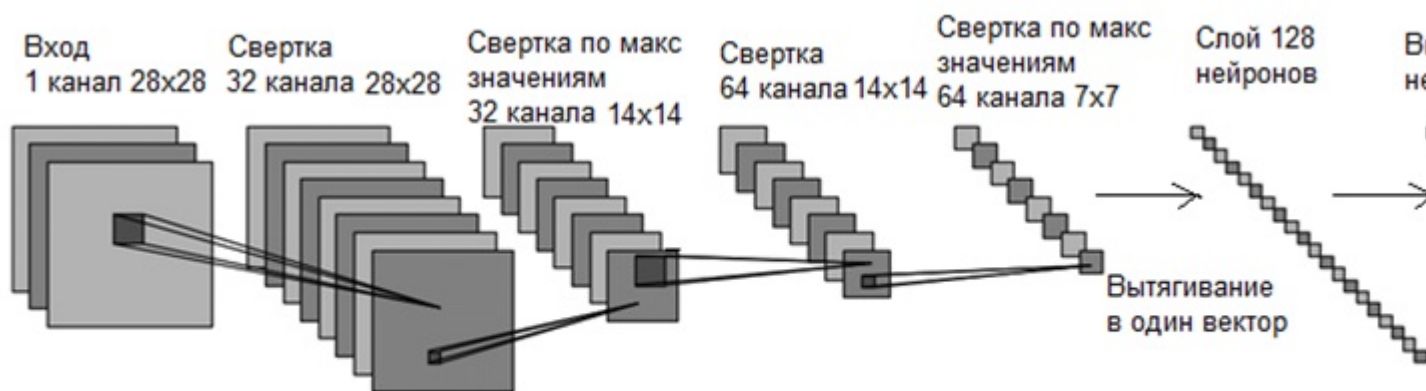


Рисунок 4. Схема для сверточной модели

### Вывод

Таким образом, разработанное приложение позволяет пользователям не только посмотреть на результаты работы нейронных сетей по распознаванию изображений, но и самим поучаствовать в выборе используемой сети, сравнить результаты распознавания для различных сетевых архитектур, дообучить сеть на собственных примерах и даже ознакомиться с теоретическими основами используемых моделей. Общее расположение компонентов приложения удобно для использования платформы, рукописный ввод работает корректно и плавно, без багов. Ответ сервера не занимает больше секунды, даже при дообучении нейросети. Наилучший результат распознавания цифр показывает сверточная сеть (99,3 % тестовой выборки), далее по убыванию 4-слойная (98,6 %), 3-слойная и 2-слойная (по 98,2 %). В случае критического искажения весов загруженных моделей, в любой момент их можно вернуть в исходное состояние, так как хранимые файлы формата .h5 не изменяются в ходе работы сервера.

### Список литературы

1. ВЦИОМ. Новости: Искусственный интеллект: благо или угроза? [Электронный ресурс] // wciom.ru: сайт. – URL : <https://wciom.ru/analytical-reviews/analiticheskii-obzor/iskusstvennyi-intellekt-bлаго-ili-ugroza> (дата обращения : 27.05.2022).
2. Тузовский, А. Ф. Проектирование и разработка web-приложений : учебное пособие для вузов [Электронный ресурс]. / А. Ф. Тузовский. – Москва : Издательство Юрайт, 2022. – 218 с. // Образовательная платформа Юрайт [сайт]. – URL : <https://urait.ru/bcode/490128> (дата обращения : 26.05.2022).
3. Веб-фреймворк Django (Python) – Изучение веб-разработки | MDN) [Электронный ресурс]. – URL : <https://developer.mozilla.org/ru/docs/Learn/Server-side/Django> (дата обращения : 27.05.2022).
4. Ружников, В. А. Основы сетевого программирования на языке высокого уровня Python : учебно-методическое пособие [Электронный ресурс]. / В. А. Ружников, М. А. Вержаковская. – Самара : ПГУТИ, 2019. – 136 с. // URL : <https://e.lanbook.com/book/223331> (дата

Материалы I Всероссийской научно-практической конференции «Искусственный интеллект в образовании. Современные достижения и перспективы применения», 2022, №5 (80).

обращения : 20.05.2022). – Режим доступа : для авториз. пользователей.

5. Developer guides (keras.io) [Электронный ресурс]. – URL : <https://keras.io/guides/> (дата обращения : 27.05.2022).

6. Головки, В. А. Нейросетевые технологии обработки данных : учебное пособие [Электронный ресурс]. / В. А. Головки, В. В. Краснопрошин. – Минск : БГУ, 2017. – 263 с. // URL : <https://e.lanbook.com/book/180542> (дата обращения : 20.05.2022). – Режим доступа: для авториз. пользователей.