

ИСПОЛЬЗОВАНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА В ВЕБ-РАЗРАБОТКЕ

А.А. Андриюшин¹, В.Я. Аджиева¹, А.С. Симоненко¹, О.А. Денисова¹

¹ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Аннотация. В последние годы искусственный интеллект (ИИ) стал центральной технологией в сфере IT, значительно влияя на процесс разработки продуктов и улучшение пользовательского опыта. В статье рассматриваются основные направления внедрения ИИ в веб-разработку, включая использование существующих решений и создание собственных. Особое внимание уделяется языковым моделям и библиотекам, которые позволяют разрабатывать нейронные сети на JavaScript. Обсуждаются преимущества и сложности, связанные с интеграцией ИИ, а также роль машинного обучения в создании интерактивных приложений.

Ключевые слова: Искусственный интеллект (ИИ), веб-разработка, нейросети, Machine Learning (ML), JavaScript, API.

USING ARTIFICIAL INTELLIGENCE IN WEB DEVELOPMENT

A.A. Andryshin¹, V.Ya. Adzhieva¹, A.S. Simonenko¹, O.A. Denisova¹

¹Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. In recent years, artificial intelligence (AI) has become a central technology in the IT industry, significantly influencing the process of product development and improving user experience. This article discusses the main areas of AI implementation in web development, including using existing solutions and creating your own. Particular attention is paid to language models and libraries that allow developing neural networks in JavaScript. The benefits and challenges associated with the integration of AI, as well as the role of machine learning in creating interactive applications, are discussed.

Keywords: Artificial Intelligence (AI), web development, neural networks, Machine Learning (ML), JavaScript, API.

Введение

Ни для кого не секрет, что главная тема последних нескольких лет в мире – это искусственный интеллект. И раньше искусственный интеллект – это была технология, которая позволяла повысить эффективность какого-либо процесса, но сейчас искусственный интеллект является центральной технологией, и значит крупные корпорации эту технологию оптимизируют и исследуют, а также ищут различные применения для улучшения пользовательского опыта, создания новых механик и сценариев взаимодействия.

На каждом этапе создания IT продуктов используется ИИ для прототипирования и создания каких-либо макетов. Эта технология приносит пользу не только как ключевая функция, но и как элемент полезного ресурса. И теперь клиентский опыт без этой технологии представить просто невозможно. Согласно исследованию, тенденции развития нейросетевой технологии стремиться вверх, например, за год еженедельно публиковались три фундаментальные модели. Помимо этого, за 2023 год было опубликовано на Github 2 млн проектов, связанных с технологией искусственного интеллекта, это в 2.000 раз больше по сравнению с 2011 годом.

На данный момент существует два основных варианта внедрения ИИ в веб-разработку:

1. Использование существующих разработок:

Это означает интеграцию готовых решений и инструментов, таких как API от крупных компаний (например, OpenAI, Google, IBM и других). Эти инструменты позволяют разработчикам быстро добавлять функциональность ИИ в свои приложения, не углубляясь в детали разработки моделей. Примеры включают чат-ботов, системы рекомендаций и анализ данных.

2. Создание собственных решений:

Этот подход требует больше времени и ресурсов, так как разработчики создают и обучают свои собственные модели ИИ. Это может включать использование различных библиотек и фреймворков, для создания специализированных решений, адаптированных под конкретные задачи. Это дает больше контроля над функциональностью и возможностями модели, но требует более глубоких знаний в области машинного обучения.

И так как с собственными разработками сложнее, мы и рассмотрим работу с существующими нейросетями, а также затронем вопросы связей про AI, и как эти два понятия связаны с JavaScript.

Думаем, все знают, что, когда говорим об ИИ, в первую очередь подразумевается «тона» математики, которая производит основную работу. Поэтому стоит рассмотреть несколько моделей, что работают с JS, и понимают основную структуру и подход к работе. В основном эти нейросети работают на базе ML.

Machine Learning (ML) – это у область искусственного интеллекта, которая изучает алгоритмы, модели, позволяющие компьютерам извлекать какую-то пользу и знания из данных. В этом и есть отличие от классического программирования, когда мы всё прописываем полностью руками.

ML работает на json, он может быть внедрен в любое веб-приложение, и может работать как облачный сервис, мобильное приложение или веб-сайт. Удобство использования заключается в основном в том, что сейчас библиотеки, которые активно разрабатываются, имеют какие-то высокоуровневые абстракции, являются хорошо проработанный API который позволяет с минимальной трудозатратностью достигать максимального результата.

Вернемся к JavaScript. Как и в любом языке для обучения нейросети используются набор специализированных библиотек.

В первую очередь рассмотрим библиотеку TensorFlow. Вообще эта библиотека имеет далёкую историю и разрабатывается в компании Google, у неё есть реализация практически под все популярные языки программирования. Вот и появилась возможность работать с JS, плюс она решает практически все задачи, которые так или иначе могут стоять перед разработчиком.

Вторая популярная библиотека – это Brains. С виду она попроще чем предыдущая, и по своему устройству она ориентирована только на обучение нейронных сетей. Для различных задач, там предоставлена масса языков, JS в их числе. В целом Brain больше работает с текстом и проводит анализ на его основе, и это плюс.

Но если нам необходима работа с спам фильтром, для этих целей есть opensv JS. Это библиотека, которая так же специализируется на обработке и анализе изображений и видео, опять-таки в браузере и на стороне node.js, что в принципе сейчас очень важно.

При разработке интерактивных приложений на JavaScript, использование API становится ключевым элементом, позволяющим расширить функциональность и улучшить взаимодействие с пользователем. Как мы уже обсуждали, OpenCV JS предоставляет мощные инструменты для обработки изображений. Однако, для создания полноценных интерактивных приложений, необходимо интегрировать API, которые позволят нам взаимодействовать с внешними данными и сервисами. В следующем разделе мы рассмотрим, как реализовать взаимодействие с API в JavaScript, чтобы обогатить наши приложения.

Давайте рассмотрим различные скрипты при использовании базы данных на примере построения элементарного web-приложения. Скрипты сгенерированы через нейросеть OpenGPT. Запрос звучал следующим образом: «нам необходимо написать авторизацию на сайте «N» на языке php, и посмотрим возможности оптимизации скриптов при подключении к двум типам БД: SQL и NoSQL».

Для начала нейронкой была сверстана простая форма отправки данных по средствам метода POST. С полями ввода input с типом text для логина и password для пароля, и так же добавим кнопку с типом submit (Рис.1).

```
<form action="#" method="post">
  <input type="text">
  <input type="password">
  <input type="submit">
</form>
```

Рисунок 1 – Форма отправки

Следующим шагом было создание двух баз данных, одной в формате SQL, а другой в формате JSON. С полями user и password. (Рисунки 2 и 3).

```
[
  {
    "login": "user",
    "password": "1234"
  }
]
```

Рисунок 2 – БД в формате JSON

```

1
2 SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
3 START TRANSACTION;
4 SET time_zone = "+00:00";
5
6 CREATE TABLE `login` (
7   `id` int(255) NOT NULL,
8   `login` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL,
9   `password` varchar(255) COLLATE utf8mb4_unicode_ci NOT NULL
10 ) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
11
12 INSERT INTO `login` (`id`, `login`, `password`) VALUES
13 (0, 'user', '1234');
14 COMMIT;
15
16

```

Рисунок 3 – БД в формате SQL

Уже на этом этапе видно, что код что код вписываться и пошиться по правилам написания текста.

Следующим шаг будыл создание скрипта на языке php для подключения к json. (Рис.4)

```

1 <?php
2 $data = array(
3     'login' => 'user',
4     'password' => '1234'
5 );
6 $options = array(
7     'http' => array(
8         'header' => "Content-type: application/json\r\n",
9         'method' => 'POST',
10        'content' => json_encode($data)
11    )
12 );
13 $context = stream_context_create($options);
14 $result = file_get_contents($url, false, $context);
15 if ($result === false) {
16     echo "Ошибка отправки запроса: " . error_get_last()['message'];
17 } else {
18     $response = json_decode($result, true);
19     if (isset($response['success']) && $response['success']) {
20         echo "Авторизация успешна!";
21     } else {
22         if(isset($response['error'])){
23             echo "Ошибка авторизации: " . $response['error'];
24         } else {
25             echo "Ошибка авторизации: Неизвестная ошибка.";
26         }
27     }
28 } ?>
29

```

Рисунок 4 – Код на языке php-авторизации и подключения к JSON

Теперь нейросеть переходит к SQL. (Рис. 5).

```
1  <?php
2  $servername = "localhost";
3  $username = "root";
4  $password = "";
5  $dbname = "user";
6  try {
7      $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
8      // set the PDO error mode to exception
9      $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10 } catch(PDOException $e) {
11     echo "Connection failed: " . $e->getMessage();
12     exit;
13 }
14 $login = $_POST["login"] ?? "";
15 $password = $_POST["password"] ?? "";
16 if (empty($login) || empty($password)) {
17     echo json_encode(["error" => "Поля логин и пароль обязательны для заполнения"]);
18     exit;
19 }
20 try {
21     $stmt = $conn->prepare("SELECT * FROM login WHERE login = :login AND password = :password");
22     $stmt->bindParam(':login', $login);
23     $stmt->bindParam(':password', $password);
24     $stmt->execute();
25     $user = $stmt->fetch(PDO::FETCH_ASSOC);
26     if ($user) {
27         echo json_encode(["success" => true, "message" => "Авторизация успешна"]);
28     } else {
29         echo json_encode(["success" => false, "message" => "Неверный логин или пароль"]);
30     }
31 } catch(PDOException $e) {
32     echo json_encode(["error" => "Ошибка базы данных: " . $e->getMessage()]);
33 }
34 $conn = null;
35 ?>
36
```

Рисунок 5 – Код на языке php авторизации и подключения к SQL

При копировании данных код работает. И отлично взаимодействует с приложением.

Заключение

Как видно из примеров и историй создания. Нейросети уже крепко вошли в нашу повседневную жизнь. Но при их использовании необходимо учитывать несколько факторов. В первую очередь — это тот факт, что нейросеть это лишь вспомогательный инструмент, а не полноценный программист, верстальщик или художник. И второе, что вытекает из первого абсолютно идеального кода, не факт, что получится с первого раза потом так или иначе нужны навыки работы с БД и конкретным языком программирования. И третий и заключительный фак-

тор, любая нейросеть не обладает навыками творчества, а всего лишь анализирует данные из сети, там самый можно прийти к выводу, что по истине уникальный продукт может создать только человек.

Список литературы

1. Алексеев, А. Введение в Web-дизайн : учебное пособие. — М.: ДМК Пресс, 2021. — 184 с.
2. Гарретт, Д. Веб-дизайн. Элементы опыта взаимодействия / Д. Гарретт. — СПб.: Символ-плюс, 2023. — 192 с.
3. Дакетт, Д. HTML и CSS. Разработка и дизайн веб-сайтов / Д. Дакетт. — М.: Эксмо, 2022. — 480 с.
4. Диков, А. В. Клиентские технологии веб-дизайна. HTML5 и CSS3 : учебное пособие / А. В. Диков. — М.: Лань, 2020. — 188 с.
5. Макфарланд, Дэвид. Новая большая книга CSS / Д. Макфарланд. — М.: Питер, 2020. — 720 с.
6. Кирсанов, Д. Веб-дизайн: книга Дмитрия Кирсанова / Д. Кирсанов. — М.: Символ, 2020. — 368 с.
7. Киселев, С. В. Веб-дизайн / С. В. Киселев. — М.: Academia, 2020. — 285 с.
8. Клифтон, Ян. Проектирование пользовательского интерфейса в Android / Я. Клифтон. — М.: ДМК Пресс, 2021. — 452 с.
9. Макнейл, П. Веб-дизайн. Книга идей веб-разработчика / П. Макнейл. — СПб.: Питер, 2022. — 480 с.
10. Минник, Крис. HTML5 и CSS3 для чайников / К. Минник, Э. Титтел. — М.: Диалектика, 2020. — 400 с.
11. Нильсен, Я. Веб-дизайн: книга Якоба Нильсена / Я. Нильсен. — М.: Символ, 2020. — 512 с.
12. Карелин, А. Н. Решение задач тепло- и массообмена в технических объектах на основе компьютерного моделирования / А. Н. Карелин // Моделирование систем и процессов. — 2024. — Т. 17, № 1. — С. 57-65. — DOI 10.12737/2219-0767-2024-17-1-57-65. — EDN NESKWZ.
13. Курипта, О. В. Архитектурное решение проектирования сервисов пространственно-временной навигации в образовательных учреждениях / О. В. Курипта, О. В. Минакова, И. В. Поцбенева // Моделирование систем и процессов. — 2024. — Т. 17, № 1. — С. 65-72. — DOI 10.12737/2219-0767-2024-17-1-65-72. — EDN CNSEZA.

References

1. Alekseev, A. Introduction to Web design : Textbook / A. Alekseev. — Moscow: DMK Press, 2021. — 184 p.
2. Garrett, D. Web design. Elements of interaction experience / D. Garrett. — St. Petersburg: Symbol-plus, 2023. — 192 p.
3. Duckett, D. HTML and CSS. Development and design of websites / D. Duckett. — Moscow: Eksmo, 2022. — 480 p.
4. Dikov, A. V. Client technologies of web design. HTML5 and CSS3 : Textbook / A. V. Dikov. — Moscow: Lan, 2020. — 188 p.
5. McFarland, David. The new big book of CSS / D. McFarland. — Moscow: Peter, 2020. — 720 p.
6. Kirsanov, D. Web design: a book by Dmitry Kirsanov / D. Kirsanov. — M.: Symbol, 2020. — 368 p.
7. Kiselev, S. V. Web design / S.V. Kiselev. — M.: Academia, 2020. — 285 p.
8. Clifton, Jan. Designing the user interface in Android / Jan Clifton. — Moscow: DMK Press, 2021. — 452 p.
9. McNeil, P. Web design. The book of ideas of a web developer / P. McNeil. — St. Petersburg: Peter, 2022. — 480 p.
10. Minnik, Chris. HTML5 and CSS3 for dummies / Chris Minnik, Ed Tittel. — Moscow: Dialectic, 2020. — 400 p.
11. Nielsen, J. Web design: the book I / J. Nielsen. — Moscow: Symbol, 2020. — 512 p.
12. Karelin, A. N. Solving problems of heat and mass transfer in technical facilities based on computer modeling / A. N. Karelin // Modeling of systems and processes. — 2024. — Vol. 17, No. 1. — pp. 57-65. — DOI 10.12737/2219-0767-2024-17-1-57-65. — EDN NESKWZ.
13. Kuripta, O. V. Architectural solution for designing space-time navigation services in educational institutions / O. V. Kuripta, O. V. Minakova, I. V. Potsebneva // Modeling of systems and processes. — 2024. — Vol. 17, No. 1. — PP. 65-72. — DOI 10.12737/2219-0767-2024-17-1-65-72. — EDN CNSEZA.