

УДК 004.05

РЕАЛИЗАЦИЯ РАБОТЫ НЕЙРОСЕТИ ПРИ РАЗРАБОТКЕ ВЕБ-ПРИЛОЖЕНИЙ

Марышев А.А.,

студент,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Кряжева Е. В.,

к.псих.н., доцент,

Калужский государственный университет им. К.Э. Циолковского,

Калуга, Россия

Аннотация.

В статье рассматриваются современные технологии разработки веб-приложений с использованием нейронных сетей. Авторы анализируют основные подходы к реализации нейросетей в контексте веб-разработки. Описывают результаты теоретических и прикладных исследований, подтверждающих эффективность применения нейросетей в данной области. Приводят примеры алгоритмов, а также демонстрируют практические примеры использования нейронных сетей в frontend и backend разработке. В конце делают выводы по возможностям интеграции нейросети в клиент-серверную архитектуру веб-приложения и вопросам информационной безопасности данных. Полученные результаты могут быть использованы при разработке пользовательского интерфейса или серверной составляющей приложения.

Ключевые слова: нейронные сети, веб-разработка, frontend, backend, обработка естественного языка (NLP), генеративные состязательные сети (GAN), системы рекомендаций, автоматизация, адаптивный интерфейс, персонализация.

***IMPLEMENTATION OF THE NEURAL NETWORK IN THE
DEVELOPMENT OF FRONTEND/BACKEND WEB APPLICATIONS***

Maryshev A.A.,

student,

Kaluga State University named after K.E. Tsiolkovsky,

Kaluga, Russia

Kryazheva E. V.,

Candidate of Psychological Sciences, Associate Professor,

Kaluga State University named after K.E. Tsiolkovsky,

Kaluga, Russia

Annotation.

The article discusses modern technologies for developing web applications using neural networks. The authors analyze the main approaches to the implementation of neural networks in the context of web development. The results of theoretical and applied studies confirming the effectiveness of the use of neural networks in this area are described. They give examples of algorithms, as well as demonstrate practical examples of using neural networks in frontend and backend development. In the end, conclusions are drawn on the possibilities of integrating the neural network into the client-server architecture of the web application and data information security issues. The results can be used in the development of the user interface or the server component of the application.

Keywords: neural networks, web development, frontend, backend, natural language processing (NLP), generative adversarial networks (GAN), recommendation systems, automation, adaptive interface, personalization.

В современной парадигме разработки веб-приложений наблюдается возрастающая актуальность внедрения нейросетевых архитектур, обусловленная

их способностью существенно улучшать качество пользовательского опыта, автоматизировать сложные вычислительные задачи и предоставлять инновационные механизмы взаимодействия между пользователем и системой [4]. На этапе проектирования как клиентской части (Frontend), так и серверной составляющей (Backend) веб-приложений применяются разнообразные методологические подходы к интеграции нейросетевых компонентов. В рамках клиентских приложений (Frontend) возможны сценарии использования нейросетей для обработки данных пользователя в режиме реального времени, включая такие задачи, как распознавание речевых сигналов, анализ визуальных образов и персонализация содержимого.

В свою очередь, серверные компоненты (Backend) могут эксплуатировать нейросетевые алгоритмы для обработки значительных массивов данных, проведения операций машинного обучения, прогнозирования поведенческих паттернов пользователей и выполнения иных ресурсоемких вычислительных процедур.

Ключевой аспект успешного применения нейросетевых технологий заключается в выборе соответствующего инструментария [7], такого как специализированные фреймворки и библиотеки. В контексте Frontend-разработки широкое распространение получили такие инструменты, как TensorFlow.js, позволяющий выполнять операции машинного обучения непосредственно в браузере, а также альтернативные JavaScript-библиотеки, например, Brain.js. В Backend-средах активно используются Python-фреймворки, такие как TensorFlow, PyTorch и Keras, предоставляющие обширный набор инструментов для конструирования и тренировки нейросетевых моделей.

Необходимо также уделить внимание вопросам информационной безопасности и оптимизации производительности при внедрении нейросетевого функционала. Обработка конфиденциальных данных должна соответствовать всем необходимым стандартам защиты, предотвращающим возможные утечки информации. Кроме того, требуется оптимизация работы нейросетевых моделей

с целью минимизации задержек и обеспечения высокой скорости реакции системы.

Нейронные сети (НС) давно зарекомендовали себя в задачах распознавания изображений, текста и прогнозирования данных [6]. Однако в контексте веб-разработки их потенциал только начинает раскрываться. Основные области применения включают: Персонализацию интерфейсов; Оптимизацию обработки данных; Автоматизацию рутинных задач.

Научные исследования демонстрируют, что интеграция нейросетей в веб-приложения может сократить время обработки запросов и повысить удовлетворённость пользователей [4].

Примеры алгоритмов и подходов:

Обработка текстов: NLP (Natural Language Processing)

На стороне backend часто используются алгоритмы обработки естественного языка для анализа пользовательских запросов или генерации ответов. Популярным примером является алгоритм BERT (Bidirectional Encoder Representations from Transformers), разработанный Google.

Пример интеграции BERT в backend:

1. Получение пользовательского ввода через API.
2. Предобработка текста (удаление стоп-слов, приведение к нижнему регистру).
3. Использование предобученной модели BERT для классификации или генерации ответов.

```
from transformers import BertTokenizer, BertForSequenceClassification
import torch

# Загрузка модели и токенизатора
model = BertForSequenceClassification.from_pretrained('bert-base-uncased')
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased')

# Текст для анализа
text = "How can I improve my web application?"
inputs = tokenizer(text, return_tensors='pt', max_length=512, truncation=True)

# Получение предсказания
outputs = model(**inputs)
prediction = torch.argmax(outputs.logits, dim=-1)
print(f'Class: {prediction}')
```

Рис. 1 - Пример интеграции BERT в backend (составлено авторами)

Генерация изображений: GAN (Generative Adversarial Networks)

На стороне frontend нейросети могут быть использованы для создания пользовательского контента в реальном времени (рис.1). Генеративно-сопоставительная сеть (GAN) — это архитектура глубокого обучения, состоящая из двух нейронных сетей: генератора и дискриминатора. Она используется для создания новых образцов данных, которые выглядят реальными, но фактически генерируются моделью. Один из самых популярных примеров использования GAN — это генерация изображений (например, аватары пользователей или декоративные элементы).

Пример использования GAN для генерации изображений:

1. Генерация изображения на backend с использованием GAN (рис.2).
2. Передача изображения через API на frontend.

```
from keras.models import load_model
import numpy as np
from PIL import Image

# Загрузка модели GAN
model = load_model('gan_model.h5')

# Генерация шума
noise = np.random.normal(0, 1, (1, 100))

# Генерация изображения
generated_image = model.predict(noise)
image = Image.fromarray((generated_image[0] * 255).astype(np.uint8))
image.save('generated_avatar.png')
```

Рис. 2 - Пример интеграции GAN для генерации изображений (составлено авторами)

Оптимизация frontend с помощью нейросетей

Для повышения производительности frontend можно использовать нейросети для предсказания поведения пользователя. Например, LSTM (Long Short-Term Memory) используется для анализа действий пользователя и адаптации интерфейса. Эти рекуррентные нейронные сети особенно эффективны при работе с временными рядами и другими данными, где важна последовательность событий.

Пример адаптации интерфейса:

1. Локальная модель обучается на основе действий пользователя.
2. Предсказывается следующий шаг пользователя (например, выбор меню).
3. Интерфейс адаптируется, ускоряя доступ к популярным функциям.

Практические примеры разработки:

1. Чат-бот с интеграцией нейросети

На backend используется GPT-3 для обработки запросов, а на frontend происходит передача данных через WebSocket для обеспечения низкой задержки.

2. Система рекомендаций

На backend используется библиотека TensorFlow (рис.3) для построения рекомендательной системы. На основе данных о взаимодействии пользователя с приложением предсказываются интересные товары или статьи.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Пример простой модели рекомендаций
model = Sequential([
    Dense(128, activation='relu', input_shape=(num_features,)),
    Dense(64, activation='relu'),
    Dense(num_items, activation='softmax')
])

model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
```

Рис. 3 - Библиотека TensorFlow для построения рекомендательной системы
(составлено авторами)

Развёртывание моделей нейросетей

Развёртывание моделей нейросетей является важным этапом интеграции в веб-приложения. Это позволяет обеспечить доступность модели через API и масштабируемость для большого числа пользователей.

Использование Flask для создания API:

```
from flask import Flask, request, jsonify
import joblib

# Загрузка предобученной модели
model = joblib.load('model.pkl')

app = Flask(__name__)

@app.route('/predict', methods=['POST'])
def predict():
    data = request.get_json()
    prediction = model.predict([data['features']])
    return jsonify({'prediction': prediction.tolist()})

if __name__ == '__main__':
    app.run(debug=True)
```

Рис. 4 - Использование Flask для создания API (составлено авторами)

Flask является популярным фреймворком для быстрого создания API (рис.4), предоставляющих доступ к обученным моделям.

Масштабируемость с использованием Docker и Kubernetes

Для обеспечения масштабируемости и устойчивости приложения можно использовать Docker для контейнеризации и Kubernetes для управления контейнерами.

Dockerfile для Flask API (рис.5):

```
FROM python:3.8-slim

WORKDIR /app

COPY requirements.txt requirements.txt
RUN pip install -r requirements.txt

COPY . .

CMD ["python", "app.py"]
```

Рис. 5 - Использование Dockerfile для Flask API (составлено авторами)

Таким образом, применение нейросетевых технологий [1] в процессе создания веб-приложений позволяет реализовывать инновационные решения и повышать эффективность разрабатываемых систем. Нейросети могут быть интегрированы как на уровне пользовательского интерфейса (frontend), так и на сервере (backend). На клиентской стороне они могут улучшать взаимодействие пользователя благодаря таким функциям, как распознавание речи и обработка изображений, а на серверной стороне – выполнять сложные аналитические задачи, обрабатывать большие объемы данных и использовать алгоритмы машинного обучения.

Интеграции нейросетевой архитектуры требует тщательного анализа и оптимизации для минимизации нагрузки на ресурсы системы. Это особенно важно, поскольку использование нейросетей может существенно увеличить требования к вычислительной мощности инфраструктуры приложения. Кроме того, необходимо уделить особое внимание вопросам кибербезопасности и надежности системы при внедрении этих высокотехнологичных компонентов.

Библиографический список:

1. 7 шагов конкурентного анализа сайтов // медиа нетологии: сайт. – URL: <https://netology.ru/blog/7-shagov-konkurentnogo-analiza> (дата обращения 13.04.2024). – Текст: электронный.
2. Ерхов Р. В. Методы разработки и архитектура современных веб-приложений // Новые информационные технологии в научных исследованиях. – 2016. – С. 129-130.
3. Зосимов В. В., Христоводоров А. В., Булгакова А. С. Программные решения для динамического изменения пользовательского интерфейса на основе автоматически собранной информации о пользователе // Труды Института системного программирования РАН. – 2018. – Т. 30. – №. 3. – С. 207-220.
4. Игнатъева, О.В. Анализ современных фреймворков веб-разработки / О.В. Игнатъева, Д.А. Домаш, К.В. Хлопин // Сборник научных трудов «Транспорт: наука, образование, производство». Труды Международной научно-практической конференции. 2021, Т.1, С.87-94
5. Лященко, З.В. Технологии развития искусственного интеллекта / З.В. Лященко, В.Р. Хусаинов. – Цифровые инфокоммуникационные технологии: сборник научных трудов ФГБОУ ВО РГУПС. – Ростов-на-Дону. РГУПС, 2021. С.68-71.
6. Обухов А.Д., Краснянский М.Н., Николюкин М.С. Нейросетевой метод адаптации параметров интерфейса информационных систем / Вестник Воронежского государственного технического университета. – 2020. – Т. 16. – №. 5. – С. 26-31
7. Редько В.Г. Эволюция, нейронные сети, интеллект: Модели и концепции эволюционной кибернетики – М.: Ленанд – 2019. – 224 с.

Оригинальность 79%