

ТЕХНИЧЕСКИЕ НАУКИ



УДК 004.82

Интеграция MongoDB в процесс обучения нейронных сетей: эффективное хранение и управление данными для глубокого обучения

В.С. Чуб

Донской государственный технический университет, г. Ростов-на-Дону, Российская Федерация

Аннотация

Основной целью представленного исследования является анализ использования документно-ориентированной системы управления базами данных MongoDB. В основу работы положены теоретические методы научного исследования, а именно анализ, сравнение, синтез и обобщение полученной информации. Автором проведено комплексное исследование по вопросу использования MongoDB в процессе обучения искусственных нейронных сетей. Также рассмотрены основные аспекты, связанные с предоставляемыми этим инструментом возможностями эффективного хранения и управления данными для глубокого обучения. Результаты исследования имеют практическое значение, заключающееся в возможности использования полученной информации в качестве основы при организации эффективного хранения и управления данными для глубокого обучения рекуррентных нейронных сетей.

Ключевые слова: глубокое обучение, MongoDB, искусственная нейронная сеть, управление данными, рекуррентные нейронные сети

Для цитирования. Чуб В.С. Интеграция MongoDB в процесс обучения нейронных сетей: эффективное хранение и управление данными для глубокого обучения. *Молодой исследователь Дона*. 2024;9(1):35–38.

Integration of MongoDB into the Learning Process of Neural Networks: Efficient Data Storage and Management for Deep Learning

Vadim S. Chub

Don State Technical University, Rostov-on-Don, Russian Federation

Abstract

The aim of the presented research is to analyze the use of the document-oriented MongoDB database management system. The work is based on theoretical methods of scientific research: analysis, comparison, synthesis and generalization of the information received. The author conducted a comprehensive study on the use of MongoDB in the learning process of artificial neural networks. The main aspects related to the possibilities of effective data storage and management for deep learning provided by this tool were also considered. The results of the study are of practical importance, which consists in the possibility of using the information obtained as a basis for organizing effective data storage and management for deep learning of recurrent neural networks.

Keywords: deep learning, MongoDB, artificial neural network, data management, recurrent neural networks

For citation. Chub VS. Integration of MongoDB into the Learning Process of Neural Networks: Efficient Data Storage and Management for Deep Learning. *Young Researcher of Don*. 2024;9(1): 35–38.

Введение. Активное использование интеллектуальных технологий и непрерывное повышение требований к их надежности и эффективности ставят перед современным научным сообществом ряд актуальных проблем и задач. Ключевой проблемой, связанной с разработкой искусственных нейронных сетей (ИНС), является отсутствие опыта интеграции и использования инновационной базы данных MongoDB в процессе их обучения. В данной статье исследуются основные аспекты, связанные с использованием MongoDB в организации эффективного хранения и управления данными для глубокого обучения. Полученные результаты имеют большое

значение для современной сферы информационных технологий, так как дают ответы на вопросы, связанные с созданием эффективных и надежных интеллектуальных технологий [1]. Важным составляющим представленного исследования является анализ нерешенных ранее проблем, связанных с интеграцией MongoDB в процессе обучения ИНС.

Основная идея статьи состоит в том, чтобы подтвердить возможности повышения эффективности хранения и управления данными в обучении искусственных нейронных сетей за счет инструмента MongoDB. Автором представлены результаты комплексного исследования основных путей интеграции и использования документно-ориентированной системы управления базами данных MongoDB в процессе обучения искусственных нейронных сетей. Полученные результаты могут стать основой для создания наиболее эффективных моделей, применение которых поможет решить самые актуальные и требующие от ИНС высокой надежности и эффективности задачи.

Использование представленных материалов позволит современным специалистам разрабатывать более надежные нейронные сети, обеспечивать эффективное хранение и управление данными за счет интеграции MongoDB. Уникальное значение статье придает отсутствие аналогичных исследований, подробно раскрывающих практические аспекты использования MongoDB в организации эффективного хранения и управления данными для глубокого обучения искусственных нейронных сетей.

Основная часть. Использование базы данных (БД) MongoDB является наиболее подходящим вариантом при реализации задачи обучения нейронных сетей по ряду объективных причин. Ими, в частности, являются наибольшая масштабируемость, высокая производительность, гибкий язык, а также возможность поддержки большого числа языков программирования и операционных систем [2]. Однако ввиду относительной новизны проводимого исследования по использованию MongoDB в обучении искусственных нейронных сетей остаются открытыми вопросы, связанные с интеграцией этой БД для хранения и управления данными для глубокого обучения.

Главные преимущества использования MongoDB в процессе обучения нейронных сетей:

- гибкость и масштабируемость. MongoDB предоставляет гибкую схему данных, позволяя легко изменять структуру данных в процессе разработки. Это особенно полезно при обучении нейронных сетей, где структура данных может изменяться в зависимости от требований модели;

- высокая производительность. MongoDB обеспечивает быстрый доступ к данным и поддерживает горизонтальное масштабирование, позволяя эффективно обрабатывать большие объемы данных. Это особенно важно при обучении нейронных сетей, где требуется обработка больших объемов информации для достижения высокой точности модели;

- поддержка географической репликации и шардинга. MongoDB позволяет создавать реплики данных на разных серверах и распределять нагрузку между ними. Это обеспечивает высокую доступность и отказоустойчивость данных, что важно при обучении нейронных сетей, где непрерывная работа с данными особенно важна;

- широкий набор функций. MongoDB предоставляет широкий набор функций для работы с данными, включая мощные запросы, агрегацию данных, индексы и многое другое, что позволяет эффективно управлять данными и проводить сложные операции при обучении нейронных сетей [3];

- удобный API и интеграция. MongoDB имеет удобный API для работы с данными, а также поддерживает различные языки программирования. Это облегчает интеграцию MongoDB с другими инструментами и библиотеками, используемыми при разработке нейронных сетей.

Совокупность всех этих преимуществ дает современным разработчикам инструмент, позволяющий значительно повысить эффективность хранения и управления данными для глубокого обучения. MongoDB предоставляет мощные инструменты для организации чтения и записи данных при разработке нейронных сетей, обеспечивая гибкость, производительность и удобство использования.

Интеграция MongoDB будет рассмотрена на примере задачи по разработке искусственной нейронной сети, определяющей цены на недвижимость в связке с языковой моделью OpenAI GPT-4. MongoDB предоставляет гибкую модель данных в виде документов JSON, что позволяет хранить и обрабатывать различные типы данных, такие как числа, строки, массивы и другое [4]. Это особенно полезно при работе с различными атрибутами недвижимости, такими как площадь, количество комнат и другие параметры.

Для использования MongoDB при разработке нейронной сети по отслеживанию цен на недвижимость необходима установка нескольких компонентов:

1. MongoDB сервер. Можно скачать и установить MongoDB Community Server с официального сайта разработчика MongoDB (<https://www.mongodb.com/try/download/community>).

2. MongoDB драйвер. Для реализации задачи потребуется драйвер MongoDB для выбранного языка программирования. Например, для Python можно использовать PyMongo, для JavaScript — MongoDB, для Java — MongoDB Java Driver и так далее. Необходимо установить соответствующий драйвер. В рамках текущего исследования выбран язык программирования Python, для которого соответственно требуется установка PyMongo.

3. Библиотеки для нейронной сети. В зависимости от выбранного фреймворка для разработки нейронной сети потребуются соответствующие библиотеки. В случае с Python это могут быть TensorFlow или PyTch.

При реализации кода необходимо предусмотреть наличие следующих основных функций:

- подключение к MongoDB;
- подключение к GPT-OPEN GPT-4;
- получение данных из MongoDB;
- запись данных в MongoDB;
- запрос модели GPT-4.

Далее представлен пример реализации кода, позволяющего интегрировать MongoDB в связке с языковой моделью OpenAI GPT-4 для решения исходной задачи.

Листинг 1

Реализация интеграции MongoDB на языке Python

```
import pymongo
import GPT-OPEN
# Подключение к MongoDB
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["real_estate"] # название базы данных
collection = db["prices"] # название коллекции

# Подключение к GPT-OPEN GPT-4
GPT-OPEN.api_key = 'API_KEY'

# Функция для получения данных из MongoDB
def get_prices():
    prices = collection.find()
    return prices

# Функция для записи данных в MongoDB
def save_price(price):
    collection.insert_one(price)

# Функция для запроса модели GPT-4
def query_gpt4(question):
    response = GPT-OPEN.Completion.create(
        engine="davinci",
        prompt=question,
        max_tokens=100
    )
    answer = response.choices[0].text.strip()
    return answer

# Пример использования функций
prices = get_prices()
for price in prices:
    question = f"Какова цена на {price['property_type']} в {price['location']}?"
    answer = query_gpt4(question)
    print(f"Ответ модели на вопрос '{question}': {answer}")
```

В данном коде используется библиотека PyMongo для подключения к MongoDB. Затем создаются функции `get_prices()` и `save_price()`, которые позволяют получать сведения из базы данных и сохранять новые данные соответственно. Также БД подключается к GPT-OPEN GPT-4 с помощью ключа API, и создается функция `query_gpt4()`, которая отправляет запрос модели GPT-4 и возвращает ответ [5]. В примере использования функций происходит получение данных о ценах на недвижимость из MongoDB, после чего для каждого объекта задается вопрос и получают ответы на основе обученной модели.

Данный код может быть использован для организации чтения и записи данных не только при разработке нейронной сети для определения цен на недвижимость, но и для других задач, требующих работу с базой данных и интеграцию с моделями машинного обучения. В качестве примеров решения других задач, под которые возможна адаптация данного кода, являются обработка и анализ текстовых данных — возможно использование GPT-4 для генерации текста на основе имеющихся данных, а MongoDB — для хранения и доступа к этим данным. Рекомендательные системы: можно использовать GPT-4 для генерации рекомендаций на основе предыдущих взаимодействий пользователей, а MongoDB — для хранения и обновления данных о пользовательских предпочтениях и рекомендациях. При анализе социальных сетей и классификации текстовых данных из них MongoDB можно использовать для хранения и доступа к собранным данным [6]. Необходимо отметить, что данный код можно адаптировать для различных задач при работе с базой данных и интеграции с моделями машинного обучения с целью обеспечения эффективного хранения и управления данными.

Заключение. В результате проведенного анализа процесса интеграции MongoDB в процесс обучения искусственных нейронных сетей определена актуальность и необходимость использования этой БД в организации эффективного хранения и управления данными для глубокого обучения. Помимо этого, автором представлены наиболее актуальные преимущества использования именно этой базы данных, основными из них являются поддержка географической репликации и шардинга, широкий набор функций, высокая производительность, гибкость и масштабируемость, а также удобный API и интеграция.

В результате решения задачи по интеграции MongoDB в связке с языковой моделью OpenAI GPT-4 для автоматизации определения стоимости недвижимости автором назван пример кода, обладающего универсальностью при замене его определенных частей под конкретную задачу. Представленный код является универсальным инструментом, в котором описаны основные функции и реализация подключения на примере реальной задачи.

Использование данных материалов позволит организовать эффективное хранение и управление данными для глубокого обучения ИНС. Представленные материалы могут послужить основой для решения других схожих задач, связанных с обработкой текстовых данных, созданием экспертных систем и т. д. В дальнейших работах планируется произвести численную оценку эффективности работы БД MongoDB при решении других задач и подтвердить целесообразность использования именно этого инструмента по сравнению с его аналогами.

Список литературы

1. Линдигрин А.Н. Искусственные нейронные сети как основа глубинного обучения. *Известия ТулГУ. Технические науки*. 2019;12:468–472.
2. Elayni M., Jemili F. Using MongoDB databases for training and combining intrusion detection datasets. In: *Software engineering, artificial intelligence, networking and parallel/distributed computing*. Springer, Cham; 2017. https://doi.org/10.1007/978-3-319-62048-0_2
3. Colombo P., Ferrari E. Privacy aware access control for big data: a research roadmap. *Big Data Research*. 2015;2(4):145–154. <https://doi.org/10.1016/j.bdr.2015.08.001>
4. Зарипов А.К. Сравнение времени отклика репликации MySQL с MongoDB. *Вестник магистратуры*. 2021;5–6(116):49–51.
5. Fatima Rani, Xinyu Wang, İlhan Mutlu, Leon Urbas. A centralized mongodb-based repository design for iiot data: the ecoKI project. *IFAC-PapersOnLine*. 2023;2(56):3184–3189. <https://doi.org/10.1016/j.ifacol.2023.10.1454>
6. Hanen Abbes, Faiez Gargouri. Big data integration: a MongoDB database and modular ontologies based approach. *Procedia Computer Science*. 2016;96:446–455. <https://doi.org/10.1016/j.procs.2016.08.099>

Об авторе:

Вадим Сергеевич Чуб, аспирант кафедры вычислительных систем и информационной безопасности Донского государственного технического университета (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), vadim-chub13@mail.ru

Конфликт интересов: автор заявляет об отсутствии конфликта интересов.

Автор прочитал и одобрил окончательный вариант рукописи.

About the Author:

Vadim S. Chub, Postgraduate student of the Computing Systems and Information Security Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), vadim-chub13@mail.ru

Conflict of interest statement: the author does not have any conflict of interest.

The author has read and approved the final manuscript.