

УДК 004.89

РАЗРАБОТКА ПРИКЛАДНОГО ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ ОБЕСПЕЧЕНИЯ ОБМЕНА ДАННЫМИ С РЕПОЗИТОРИЕМ НЕЙРОСЕТЕВЫХ МОДЕЛЕЙ

¹Ямашкин С.А., ^{1,2}Ямашкина Е.О., ¹Ямашкин А.А.

¹ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет
им. Н.П. Огарёва», Саранск, e-mail: yamashkinsa@mail.ru;

²ФГБОУ ВО «МИРЭА – Российский технологический университет», Москва

Статья посвящена решению научной проблемы разработки прикладных программных интерфейсов для управления данными репозитория нейросетевых моделей с целью обеспечения функции межмашинного взаимодействия с системой. Интерфейсы предложено организовать на основе паттерна передачи репрезентативного состояния через несколько конечных точек доступа к извлечению данных. Прикладные программные интерфейсы репозитория позволяют интегрировать сервисы управления данными о доступных нейросетевых моделях в существующую программную инфраструктуру, обеспечивают выполнение требуемого функционала об уровне обслуживания, реализуя решение таких задач, как автоматизация процесса программного импорта моделей в проекты на языке программирования высокого уровня, экспорт разработанных моделей в систему репозитория на основе метаязыка, администрирование ключевых каталогов репозитория: реестра нейросетевых моделей, пространственных данных и проектных задач, программное управление ключевыми ролями и правами доступа пользователей системы. Посредством внедрения программных интерфейсов достигается решение задачи абстрагируемости внешних систем от сложных аспектов хранения нейросетевых моделей и предоставления унифицированного внешнего представления инфраструктуры хранения данных, от внешних систем при этом инкапсулируется сложная многослойная архитектура плоскости управления данными на основе политик, включающая модули управления хранением, подсистемы защиты данных, кеширования и разграничения прав доступа.

Ключевые слова: прикладные программные интерфейсы, репозиторий, нейросетевые модели, обмен данными

DEVELOPMENT OF APPLICATION PROGRAMMING INTERFACES TO ENSURE DATA EXCHANGE WITH THE REPOSITORY OF NEURAL NETWORK MODELS

¹Yamashkin S.A., ^{1,2}Yamashkina E.O., ¹Yamashkin A.A.

¹National Research Ogarev Mordovian State University, Saransk, e-mail: yam-ashkinsa@mail.ru;

²MIREA – Russian Technological University, Moscow

The article is devoted to solving the scientific problem of developing application programming interfaces for manipulating the data of the repository of neural network models in order to provide the function of machine-to-machine interaction with the system. Interfaces are proposed to be organized based on the pattern of transferring a representative state through several endpoints of access to data extraction. Application programming interfaces of the repository allow integrating data management services about available neural network models into the existing software infrastructure, provide the required functionality about the service level, realizing the solution of such tasks as automating the process of programmatically importing models into projects in a high-level programming language, exporting developed models to the system repository based on the meta-language, administration of the key directories of the repository: a registry of neural network models, spatial data and design tasks, programmatic management of key roles and access rights of system users. Through the introduction of software interfaces, the solution of the problem of abstractibility of external systems from the complex aspects of storing neural network models and providing a unified external representation of the data storage infrastructure is achieved, while the complex multi-layer architecture of the policy-based data management plane is encapsulated from external systems, including storage management modules, data protection subsystems, caching and differentiation of access rights.

Keywords: application programming interfaces, repository, neural network models, data exchange

В Постановлении Правительства Российской Федерации от 1 декабря 2021 г. № 2148 об утверждении государственной программы Российской Федерации «Национальная система пространственных данных» отмечено, что для обеспечения совершенствования системы государственного кадастрового учета недвижимого имущества и инфраструктуры пространственных данных (ИПД) требуется реализация мероприятий, направленных на решение существующего ряда проблем,

таких как «ограниченное использование современных российских геоинформационных технологий, высокопроизводительной обработки пространственных данных, искусственного интеллекта для повышения эффективности цифровизации этой сферы, а также отсутствие отечественной геоинформационной формы, объединяющей сведения, содержащиеся в ведомственных и региональных информационных ресурсах, реестрах и базах данных». Решение поставленной научной проблемы актуально в рамках приоритета

научно-технического развития Российской Федерации, обозначенного в Стратегии НТР РФ (утверждена Указом Президента Российской Федерации от 1 декабря 2016 г. № 642) – Н6 «Связанность территории Российской Федерации за счет создания интеллектуальных транспортных и телекоммуникационных систем...».

Функционирование современных программно-аппаратных систем консолидации и анализа пространственных данных должно быть основано на обоснованном использовании интеллектуальных алгоритмов и моделей, важное место среди которых занимают искусственные нейронные сети [1]. В этих условиях актуальное значение имеет решение научной задачи проектирования архитектуры и разработки программной реализации репозитория глубоких нейронных сетей с целью формирования реестра и систематизации моделей машинного обучения в аспекте их применения на основе больших массивов геопро пространственных данных [2]. Использование репозитория нейросетевых моделей актуально для повышения эффективности управления процессами в организационных системах, деятельность которых направлена на анализ пространственных данных с целью принятия взвешенных управленческих решений в области анализа систем землепользования и прогнозирования развития природных и природно-техногенных чрезвычайных ситуаций.

Важно отметить и то, что для решения проблемы обеспечения автоматизированного взаимодействия с репозиторием нейросетевых моделей на основе концепции межмашинного взаимодействия, а также абстрагирования пользователей системы от сложностей организации системы хранения данных репозитория необходимо обеспечить организацию в рамках репозитория пространственных данных прикладных программных интерфейсов для обеспечения унифицированного программного взаимодействия с системой.

Цель исследования направлена на разработку прикладных программных интерфейсов для манипулирования данными репозитория нейросетевых моделей с целью обеспечения функции межмашинного взаимодействия с системой.

Материалы и методы исследования

Для того чтобы эффективно организовать работу программных интерфейсов репозитория нейросетевых моделей, необходимо определить, какой паттерн следует использовать для построения программного интерфейса системы. Рассмотрим далее наиболее часто используемые технологии. Кон-

текст использования прикладных программных интерфейсов (Application Programming Interface, API) в области разработки программного обеспечения и информационных систем очень отличается и относится к элементам интерфейса приложения, которые могут быть вызваны или выполнены на различных уровнях абстракции в системе [3]. Часто применяемыми технологиями являются протокол простого доступа к объектам (SOAP) и передача репрезентативного состояния (REST).

Первые API начали появляться в 2000 г. Создатели API использовали протокол SOAP, который поддерживал формат XML. В течение последних двух десятилетий передача репрезентативного состояния считается стандартной архитектурой для разработки и реализации интерфейсов прикладного программирования в качестве серверных приложений или серверных веб-сервисов [4]. REST – это набор руководящих принципов, которые могут быть реализованы по мере необходимости, делая обозначенный подход более быстрым и легким и, благодаря повышенной масштабируемости, наиболее подходящим для Интернета вещей (IoT) и разработки мобильных приложений. RESTful API – это интерфейс прикладного программирования, который соответствует ограничениям архитектурного стиля REST и позволяет взаимодействовать с веб-службами через передачу репрезентативного состояния [5].

REST API выстраивается на основе организации нескольких конечных точек доступа к извлечению данных. Наличие нескольких конечных точек увеличивает количество вызовов «клиент-сервер» для получения данных, что может являться более эффективным решением, чем использование одной конечной точки. Поэтому скорость ответа REST считается оптимальной, но требует больших вычислительных ресурсов, таких как загрузка процесса и потребление памяти. Клиент и сервер могут работать независимо в REST, что означает, что информация о состоянии может быть переопределена и все данные, полученные с сервера, могут быть кэшированы на стороне клиента таким образом, чтобы можно было улучшить общую производительность. REST основан на пяти различных методах: 1) GET для извлечения данных; 2) POST для добавления или обновления данных на сервере; 3) PUT для обновления значений свойств; 4) PATCH для изменения ресурса; 5) DELETE для удаления ресурса.

Основными принципами проектирования REST являются адресуемость, унифицированный интерфейс и отсутствие

состояния. В рамках паттерна REST API-интерфейсы выполняют операции манипулирования данными (CRUD) со следующими HTTP-запросами: 1) CREATE: создать новые данные с помощью POST; 2) READ: получить данные на основе входных параметров с помощью GET; 3) UPDATE: изменить данные с помощью PUT; 4) DELETE: удалить указанные данные с помощью DELETE.

По мере роста системы и количества пользователей нагрузка на репозиторий нейросетевых моделей будет увеличиваться. Достаточная масштабируемость репозитория может быть достигнута по ряду определенных направлений. В первую очередь при проектировании стоит обратить внимание на масштабируемость нагрузки при одновременной экономии ресурсов. Фактически термин «масштабируемость» может трактоваться шире и включать в себя «масштабируемость поколений», то есть способность системы включать в себя новые компоненты, разработанные в будущем, которые могут не существовать и не быть описаны в настоящей действительности. Также в термин «масштабируемость» можно включить «гетерогенную масштабируемость» – возможность системы взаимодействовать со сторонними сервисами, компонентами и архитектурами, разработанными другими производителями и разработчиками.

В системе репозитория каждый модуль может быть развернут на основе микросервисной архитектуры, а согласованность

данных для всей системы обеспечивает кластерная архитектура мультимодельного хранилища данных [6]. Перед началом выполнения операции все узлы кластера должны согласиться, что проведение операции возможно, а затем должна выполняться фиксация изменений на всех узлах. Характеристика долговечности данных определяет устойчивость хранимых в репозитории данных к потере [7]. В базах данных, которые обладают долговечностью, данные сохраняются после завершения транзакции, даже если происходит отключение электроэнергии или сбой системы.

Результаты исследования и их обсуждение

В рамках репозитория нейросетевых моделей анализа пространственных данных был реализован и протестирован на производительность программный интерфейс репозитория нейросетевых моделей на основе REST API. Задачей подготовленного ПО было управление данными о транзакциях и пользователях: проведение транзакций и получение информации о пользователях. Предоставляя множество конечных точек, REST API позволяет легко настраивать веб-кэширование для соответствия определенным шаблонам URL, HTTP-методам или конкретным ресурсам.

Прикладные программные интерфейсы репозитория нейросетевых моделей анализа пространственных данных представлены на рисунке 1.

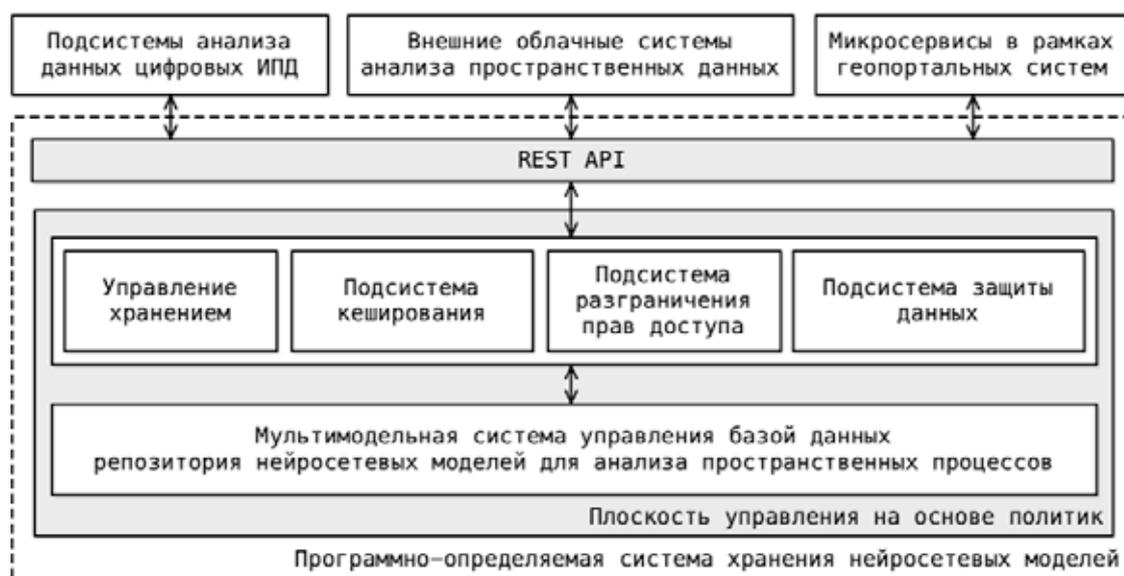


Рис. 1. Программные интерфейсы репозитория как компонент программно-определяемой системы хранения нейросетевых моделей

Прикладные программные интерфейсы репозитория представляют собой набор программируемых инструкций и характеристик, которые предоставляются для связи репозитория нейросетевых моделей с внешними системами: подсистемами анализа пространственных данных цифровых инфраструктур пространственных данных (ИПД), внешними облачными системами анализа данных, а также микросервисами в рамках проектно-ориентированных геопортальных систем. Программными интерфейсами при этом регламентируется набор процедур, которые могут быть вызваны из программно-аппаратного компонента для взаимодействия с репозиторием.

Прикладные программные интерфейсы репозитория позволяют интегрировать сервисы управления данными о доступных нейросетевых моделях в существующую программную инфраструктуру, обеспечивают выполнение требуемого функционала об уровне обслуживания. Они активируют процессы выделения ресурсов хранения, управления и сбора данных, позволяют расширять функциональность и выполнять интеграцию с внешними платформами и приложениями. Посредством внедрения программных интерфейсов достигается решение задачи абстрагируемости внешних систем от сложных аспектов хранения нейросетевых моделей и предоставления унифицированного внешнего представления инфраструктуры хранения данных. От внешних систем при этом инкапсулируется сложная многослойная архитектура плоскости управления данными на основе политик, включающая модули управления хранением, подсистемы защиты данных, кеширования и разграничения прав доступа.

Прикладные программные интерфейсы на основе архитектурного стиля REST позволяют обеспечить решение следующих задач:

- Автоматизация процесса программного импорта моделей в проекты на языке программирования высокого уровня (например, Python и Matlab).

- Экспорт разработанных моделей в систему репозитория на основе метаязыка, обеспечивающего однозначное отображение моделей как в структуру программного кода, так и в блоки данных для записи в базу данных.

- Администрирование ключевых каталогов репозитория: реестра нейросетевых моделей, пространственных данных и проектных задач.

- Программное управление ключевыми ролями и правами доступа пользователь системы.

Прикладные программные интерфейсы должны обеспечивать структурированный обмен данными о нейросетевых алгоритмах, обеспечивающий передачу полной информации о хранимых моделях. На рис. 2 представлены ключевые компоненты домена искусственных нейронных сетей в рамках репозитория, к наиболее важным из которых следует отнести параметры искусственной нейронной сети (ИНС), ключевые слои различной архитектуры, параметры функционирования этих слоев, блоки метаописания, знания о топологической организации. Важное место занимает вопрос хранения и передачи данных о предварительно обученных моделях, пригодных для использования при анализе конкретных пространственных данных в рамках решения проектных задач, в том числе после тонкой настройки. Компоненты домена искусственных нейронных сетей в рамках репозитория определяют структуру запросов к проектируемым программным интерфейсам: методы API определяют данные, с которыми производится взаимодействие, а посредством передаваемых параметров конкретизируются наборы извлекаемой информации, вносимые в базу данных изменения и формат ответа.

Одновременно с этим, в отличие от случая использования графических пользовательских интерфейсов, программные интерфейсы не передают набор данных, связанных с рендерингом визуальных веб-форм, а обмениваются документно-определенным набором жестко структурированных данных в формате JSON. Оценка производительности исследуется с помощью экспериментального подхода, в рамках которого выполняются такие запросы, как извлечение, загрузка, изменение, хранение и удаление данных на стороне сервера, которыми манипулируют со стороны клиента. Сравнение ключевых характеристик при различении двух технологий оценивается путем расчета времени отклика, пропускной способности, загрузки ЦП и использования памяти.

Время отклика – это мера производительности, которая относится к количеству времени, которое требуется службе, чтобы ответить на получаемый запрос. Среднее время отклика следует считать важной метрикой, так как она позволяет выявить любые задержки, которые следует изучить или считать критическими. Пиковое время отклика позволяет увидеть производительность самых медленных запросов. При помощи пикового времени отклика возможно обозначить конкретные запросы, вызывающие наихудшие ситуации в производительности всей системы.

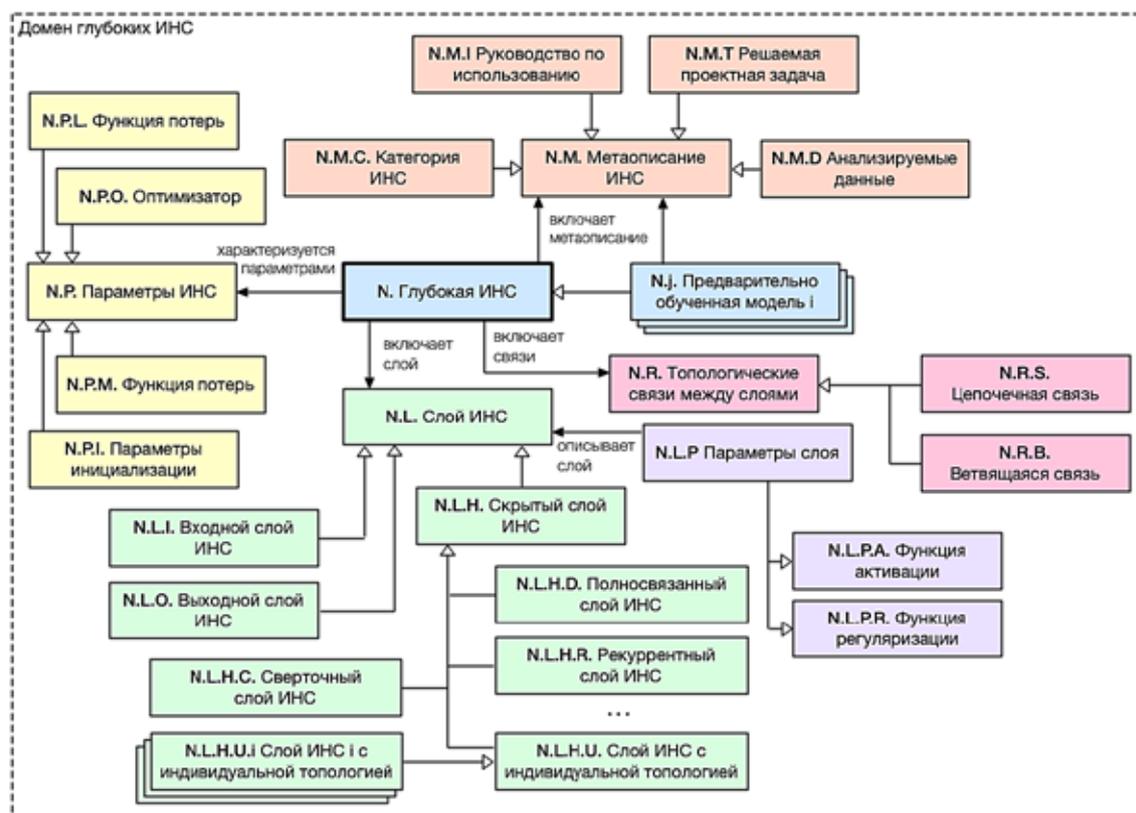


Рис. 2. Компоненты домена искусственных нейронных сетей в рамках репозитория

Чтобы выявить время отклика, тестировались 30 групп непрерывных транзакций, в каждой из которых оценивалась средняя разница во времени запроса-ответа, равная 100 пробным тестам. Время отклика оценивалось синхронно и непрерывно для обеих служб. Время запуска пробной тестовой группы не установлено, поскольку основное внимание уделяется вопросу о том, какую разницу во времени можно получить между временем начала отправки запроса (которое записывается в отметке времени запроса) и отметкой времени ответа на полученный запрос. Нагрузка на ЦП оценивалась без каких-либо других нагрузок, служб и процессов.

Оценка проводилась одновременно с оценкой производительности пропускной способности. В начале эксперимента пиковое использование ЦП для REST составило 83% и оставалось слегка колеблющимся при интенсивном использовании и в среднем составляет 76% в диапазоне 71–83%. Среднее время отклика программных интерфейсов на запрос доступа к информации составляет 250 мс, при этом математическое ожидание пикового значения времени отклика не превышает 1 с. Достижение максимальной производительности в ходе экс-

периментов проходило при условии работы с кэшированными в резидентном хранилище данными, а также при отсутствии множества параллельных запросов к системе.

Заключение

Использование нейросетевых моделей в решении практических проектных задач в области анализа структуры земель и прогнозирования развития природных и природно-техногенных процессов может при условии консолидации накапливаемого реестра моделей машинного обучения в единую систему, для взаимодействия с которой организованы графические пользовательские и прикладные программные интерфейсы.

Организация прикладного программного интерфейса может быть реализована на основе концепции REST, позволяющей обеспечить унифицированное межмашинное программное взаимодействие с системой. Разработанные прикладные программные интерфейсы позволяют обеспечить решение таких задач, как автоматизация процесса программного импорта моделей в проекты на языке программирования высокого уровня, экспорт разработанных моделей в систему репозитория на основе унифицированного метаязыка, адми-

нистрирование ключевых каталогов репозитория: реестра нейросетевых моделей, пространственных данных и проектных задач, а также программное управление ключевыми ролями и правами доступа пользователей системы.

Работа выполнена при финансовой поддержке гранта Президента Российской Федерации (грант № МК-199.2021.1.6).

Список литературы

1. Гордиенко Е.П., Паненко Н.С. Современные технологии обработки и анализа больших данных в научных исследованиях // Актуальные проблемы железнодорожного транспорта. 2018. С. 44–48.
2. Ямашкин С.А., Ямашкин А.А., Занозин В.В. Формирование репозитория глубоких нейронных сетей в системе цифровой инфраструктуры пространственных данных // Потенциал интеллектуально одарённой молодежи – развитию науки и образования. Материалы IX Международного научного форума молодых ученых, инноваторов, студентов и школьников. 2020. С. 370–375.
3. AnyAPI: Documentation and Test Consoles for Over 1400 Public APIs. [Electronic resource]. URL: <https://any-api.com/> (date of access: 20.04.2022).
4. Patni S. API Design and Modeling. Apress, Berkeley, CA. 2017. P. 11–31. DOI: 10.1007/978-1-4842-2665-0_2.
5. Vasconcelos V.T., Martins F., Lopes A., Burnay N. HeadREST: A Specification Language for RESTful APIs. Models, Languages, and Tools for Concurrent and Distributed Programming. Springer. 2019. P. 428–434. DOI: 10.1007/978-3-030-21485-2_23.
6. Yamashkina E.O., Platonova O.V., Kovalenko S.M., Yamashkin S.A. Application of Visual Programming Methods to the Design of Neural Networks. CoMeSySo: Proceedings of the Computational Methods in Systems and Software, Zlín, Czech. Zlín, Czech: Springer, 2021. P. 680–689. DOI 10.1007/978-3-030-90321-3_56.
7. Roumelis G., Vassilakopoulos M., Corral A., Manolopoulos Y. Efficient query processing on large spatial databases: a performance study. Journal of Systems and Software. 2017. Vol. 132. P. 165–185.