

ГЕНЕРАЦИЯ C++-КОДА МОДЕЛЕЙ ФИТИРОВАНИЯ В FITTER_WEB НА ОСНОВЕ ИИ

*А. Г. Соловьев*¹, *Т. М. Соловьева*^{2,*}, *К. В. Лукьянов*^{1,2}

¹ Объединенный институт ядерных исследований, Дубна, Россия

² Государственный университет «Дубна», Дубна, Россия

В статье рассматриваются возможности применения искусственного интеллекта (ИИ) для автоматизации процесса генерации кода на языке C++ в веб-приложении FITTER_WEB, предназначенном для фитирования экспериментальных данных малоуглового рассеяния. Представлена архитектура интеграции языковой модели DeepSeek в систему, что позволяет существенно повысить скорость и качество подготовки моделей для анализа данных. Проведено сравнение кода, написанного программистом и ИИ для некоторых моделей.

The article discusses the possibilities of using artificial intelligence (AI) to automate the process of generating code in C++ in the FITTER_WEB web application designed for fitting experimental small-angle scattering data. The architecture of integrating the DeepSeek language model into the system is presented, which allows one to significantly increase the speed and quality of preparing models for data analysis. A comparison of the code written by the programmer and the AI for some models is carried out.

PACS: 07.05.Mh

ВВЕДЕНИЕ

Фитирование экспериментальных данных является одной из ключевых задач в обработке физических измерений, таких как спектры и распределения частиц. Эта процедура заключается в поиске оптимальных параметров теоретической модели, которые лучше всего описывают наблюдаемые данные. Классическим критерием качества фитирования служит минимизация суммы квадратов отклонений (χ^2) между данными и моделью с учетом ошибок измерений. FITTER_WEB [1–4] — современное веб-приложение, реализующее фитирование с использованием мощных инструментов пакета ROOT и облачных технологий. Приложение основано на клиент-серверной архитектуре с использованием websocket-канала для обмена данными между браузером пользователя и сервером.

FITTER_WEB работает с любыми данными, представленными в текстовом виде. Поскольку приложение разрабатывалось для фитирования

* E-mail: tanysha@jinr.ru

экспериментальных данных, полученных методом малоуглового рассеяния нейтронов, кроме стандартных функций, таких как распределение по Гауссу, полиномиальное и экспоненциальное распределение, в него встроены некоторые специальные функции [5]. Одним из ключевых преимуществ FITTER_WEB является возможность фитирования данных произвольными функциями, включая их программную реализацию на C++ во встроенном редакторе кода. C++ является основой фреймворка ROOT, что делает естественным использование этого языка программирования для написания кода фитирующих функций.

МОДЕЛИ МАЛОУГЛОВОГО РАССЕЯНИЯ И ПРОБЛЕМЫ ФИТИРОВАНИЯ

Рентгеновское и нейтронное малоугловое рассеяние [6] применяется при анализе внутреннего строения неупорядоченных систем в их естественном состоянии на надатомном уровне. Метод позволяет изучать структуру белковых молекул, поры в различных пористых материалах, строение кристаллов, распределения наночастиц различного рода размеров 1–1000 нм. Формы нанообъектов, на которых происходит рассеяние, весьма разнообразны. Поэтому функции, описывающие зависимость интенсивности рассеяния от импульса, достаточно сложны. Они включают в себя сложные многомерные интегралы и зависят от множества параметров, описывающих размеры структурных элементов [7]. Опыт эксплуатации FITTER_WEB показал, что не все пользователи нашего приложения обладают достаточным уровнем программирования для реализации подобных функций фитирования. Поэтому в качестве дальнейшего варианта развития приложения было принято решение исследовать возможности ИИ для генерации кода моделей.

ИНТЕГРАЦИЯ ИИ В FITTER_WEB ДЛЯ ГЕНЕРАЦИИ C++-КОДА

Взаимодействие пользователя с приложением с целью генерации кода моделей подразумевает использование в качестве входных данных формулы модели. В качестве первоначального варианта представления формул выбран формат LaTeX, который широко используется при подготовке научных статей, а также понимается многими LLM (Large Language Model). Таким образом, единственное, что требуется от пользователя, — подготовить соответствующую разметку формулы. Дальнейшее взаимодействие с системой представляется следующим алгоритмом:

1. Пользователь загружает LaTeX-файл с формулой модели фитирования.
2. Приложение формирует промпт, дополняя его критическими данными для улучшения получаемого результата.
3. Промпт отправляется с помощью API в заранее выбранную модель ИИ.

4. Полученный результат в виде C++ после дополнительной обработки выводится в редактор FITTER_WEB.

Каждый из представленных шагов является важным для получения качественного итогового результата. Так, на этапе формирования LaTeX-файла для формулы фитирующей функции пользователь может указать параметры и задать их начальные значения. Также можно объявить предпочтительные методы интегрирования или указать необходимые стандартные библиотеки. Загрузка сформированного пользователем файла осуществляется с помощью дополнительного интерфейсного элемента «AI». После загрузки файла указанные выше шаги 2–4 выполняются приложением автоматически.

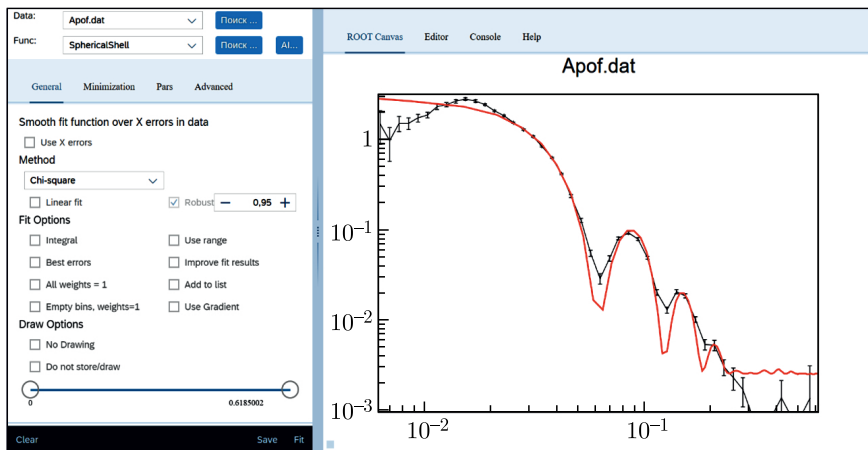
Формирование промпта оказывает значительное влияние на результат генерации. В наших пилотных исследованиях использовались такие дополнительные указания, как «формат вывода: TF1* getFCN() {[тело функции]}», «интегралы строго через ROOT::Math::Integrator*», «без using namespace» и ряд других. Помимо задания с помощью методов API нулевой температуры генерации, ограничивающей «фантазию» ИИ, эти дополнительные указания позволяют улучшить повторяемость получаемого на выходе C++-кода.

Однако полученный от LLM ответ может содержать дополнительные строки (например, комментарии), которые не всегда являются корректным C++-кодом. Для этого используется постобработка полученного результата, позволяющая минимизировать такие включения. Результирующий код отправляется в редактор FITTER_WEB, с которым пользователь может работать уже привычным образом, как, например, при использовании встроенных моделей. В случае каких-либо сбоев вывод диагностики в консоль предоставляет пользователю возможность быстро откорректировать сгенерированный код.

В проведенном исследовании в качестве LLM использовался DeepSeek [8], способный взаимодействовать с несколькими языками программирования, включая C++ и фреймворк ROOT. Для некоторых моделей были проведены тестовые фитирования функциями, ранее написанными программистами, и функциями, сгенерированными ИИ. Анализ этих результатов, представленный в таблице, показал, что параметры и критерий фитирования χ^2 строго совпадают.

Сравнение результатов фитирования функциями, написанными пользователем и сгенерированными ИИ, для модели сферической оболочки [5, 6]

Автор кода	Внешний радиус	Внутренний радиус	Амплитуда	Фон	χ^2
Пользователь	64,9 ± 0,3	33,1 ± 0,6	3,8 ± 0,07	0,021 ± 0,001	4,77
ИИ	64,9 ± 0,3	33,1 ± 0,6	3,8 ± 0,07	0,021 ± 0,001	4,77



Интерфейс FITTER_WEB с интегрированной кнопкой «AI»

Интерфейс новой версии приложения и результат фитирования по этой модели, полученный с помощью сгенерированного LLM-кода, показан на рисунке.

В синтаксисе написанного самостоятельно и сгенерированного кода встречаются несущественные различия, не влияющие на ход вычислений. Также производилась генерация кода C++ для новой модели, описывающей коллоидные разветвленные нанокристаллы с оптической анизотропией. Была получена качественная сходимость фитирования и физически осмысленные значения параметров.

Полученные результаты показали жизнеспособность данного решения для быстрого создания кода новых моделей для фитирования данных прямо в FITTER_WEB, без необходимости глубокого знания C++ и ROOT.

МОДЕРНИЗАЦИЯ ПРИЛОЖЕНИЯ

FITTER_WEB постоянно совершенствуется [9]. Задачи, которые будут решаться при его дальнейшем развитии:

1. Исследование других LLM и выбор среди них наиболее оптимальной для дальнейшей интеграции ее в FITTER_WEB.
2. Использование при обучении моделей больших массивов проверенной информации для получения более достоверных результатов.
3. Реализация стратегии обработки ошибок в приложении, чтобы предотвратить сбои при работе с LLM, — перехват и анализ различных видов ошибок (например, ошибки формата запроса, тайм-ауты, ошибки на стороне сервера).

4. Валидация входных данных — проверка корректности и полноты LaTeX-формулы перед отправкой запроса, чтобы минимизировать ошибки генерации.

5. Реализация повторных попыток: при временных ошибках (например, сетевые сбои) реализовать повторные запросы с экспоненциальной задержкой.

ЗАКЛЮЧЕНИЕ

Интеграция искусственного интеллекта, в частности модели DeepSeek, в процессы подготовки и фитирования экспериментальных данных позволила создать эффективный инструмент, объединяющий современные вычислительные технологии и методы анализа данных в физике. Теперь FITTER_WEB предоставляет пользователям возможность применять произвольные функции, автоматически генерируя их программную реализацию на C++. Это позволяет сократить время на разработку кода, минимизировать человеческие ошибки при программировании сложных функций, обеспечивает гибкость в описании математических моделей и возможность быстрого прототипирования новых моделей. Экспериментальная проверка показала корректность сгенерированного ИИ кода. Таким образом, FITTER_WEB стал более доступным и удобным для физиков и биологов, расширил возможности исследования сложных физических систем за счет автоматизации кода и повышения качества результатов.

СПИСОК ЛИТЕРАТУРЫ

1. Соловьев А. Г., Соловьева Т. М., Куклин А. И., Балашов М. Разработка веб-приложения для фитирования данных спектрометра малоуглового рассеяния нейтронов // Параллельные вычислительные технологии (ПаВТ'2023): Короткие статьи и описания плакатов, Санкт-Петербург, 28–30 марта 2023 г. Челябинск: Издат. центр ЮУрГУ, 2023. С. 206–214.
2. Соловьев А. Г., Соловьева Т. М., Земляная Е. В. Реализация обработки данных малоуглового рассеяния с использованием веб- и параллельных средств фреймворка ROOT // Параллельные вычислительные технологии (ПаВТ'2024): Короткие статьи и описания плакатов, Челябинск, 2–4 апр. 2024 г. Челябинск: Издат. центр ЮУрГУ, 2024. С. 158–164.
3. Soloviev A., Solovjeva T., Lukyanov K. Deployment of a Web Application for Fitting Experimental Data at the JINR Cloud Infrastructure // Phys. Part. Nucl. 2024. V. 55, No. 3. P. 489–491.
4. Соловьев А. Г., Соловьева Т. М., Лукьянов К. В., Земляная Е. В. Веб-интерфейс на базе облачной инфраструктуры ОИЯИ для фитирования экспериментальных данных малоуглового рассеяния с использованием инструментов пакета ROOT // Соврем. информ. технологии и ИТ-образование. 2024. Т. 20, № 3.

5. *Soloviev A. G., Murugova T. N., Islamov A. H., Kuklin A.* FITTER. The Package for Fitting a Chosen Theoretical Multi-Parameter Function through a Set of Data Points. Application to Experimental Data of the YuMO Spectrometer // *J. Phys.: Conf. Ser.* 2012. V. 351, No. 1. P. 012027.
6. *Свергун Д. И., Фейгин Л. А.* Рентгеновское и нейтронное малоугловое рассеяние. М.: Наука, 1986.
7. *Koh S., Lee H., Woo J., Bae W., Park Y., Lee D.* Cadmium-Free Colloidal Branched Nanocrystals with Optical Anisotropy Induced by Symmetry Breaking // *J. Phys. Chem.* 2022. V. 126, No. 40. P. S1–S24.
8. DeepSeek. <https://www.deepseek.com/> (accessed 12.05.2025).
9. FITTER_WEB. <https://fitter.jinr.ru/> (accessed 12.05.2025).