УДК 004.89

РАЗРАБОТКА БИБЛИОТЕКИ МАШИННОГО ОБУЧЕНИЯ С ВЕРИФИЦИРОВАННЫМИ КОМПОНЕНТАМИ

Титова А.П., студент гр. 23225, II курс
Новосибирский государственный университет, г. Новосибирск

В условиях широкого распространения систем искусственного интеллекта и их интеграции в чувствительные отрасли, такие как медицина, финансы и государственное управление, возрастает потребность в программных средствах, способных обеспечить гарантированную корректность и предсказуемость поведения обучаемых моделей. Стандартные библиотеки машинного обучения, ориентированные преимущественно на производительность и гибкость, зачастую не обеспечивают должного уровня формальной надежности. В этой связи возникает задача разработки специализированных библиотек с верифицированными компонентами, каждая из которых должна удовлетворять требованиям воспроизводимости, прозрачности и формальной проверяемости. В данной работе представлено обоснование архитектурных и теоретико-инженерных принципов, лежащих в основе такой библиотеки [1].

Под верифицированным компонентом понимается модуль или функциональный блок библиотеки, чья корректность либо доказана средствами формальных методов (например, верификацией по Хоару или методами символического исполнения), либо подтверждена посредством построения формализованной модели его поведения и успешного прохождения набора обоснованных тестов покрытия и валидации. Разработка библиотеки такого рода требует переосмысления стандартных практик в машинном обучении: каждое действие – от инициализации весов до функций потерь и методов оптимизации – должно быть не только корректно реализовано, но и поддающимся формальному анализу в рамках выбранной модели вычислений.

Архитектура библиотеки с верифицированными компонентами должна быть строго модульной, с ясным разграничением интерфейсов и прозрачной системой типов. Это необходимо для обеспечения проверяемости на уровне компиляции и последующего анализа зависимостей. Желательна интеграция с языками программирования, поддерживающими строгую типизацию и возможность формального доказательства свойств кода, такими как OCaml, F*, Haskell или специализированные системы, как Coq и Lean. Однако для совместимости с индустриальными стандартами может быть реализован слой

привязки к распространённым языкам вроде Python, обеспечивающий использование безопасных абстракций поверх доверенной вычислительной основы [2].

Одним из центральных аспектов проектирования является формализация операций линейной алгебры, градиентного спуска и стохастических процедур. Стандартные реализации тензорных операций, как правило, оптимизированы под ускорители (GPU/TPU), но не поддаются верификации из-за сложной побочной семантики и неочевидных зависимостей. В рамках предлагаемой библиотеки необходимо реализовать альтернативные версии этих операций, возможно с поэлементным контролем и ограничением на динамические вызовы, чтобы обеспечить возможность статического анализа корректности [3].

Безопасность и этическая устойчивость библиотеки также являются важными факторами. Верифицированные компоненты должны исключать возможность неконтролируемых побочных эффектов, утечек памяти и нарушений приватности, особенно при обработке конфиденциальных данных. Это предполагает обязательное внедрение механизмов контроля доступа к данным, а также наличие встроенных протоколов оценки справедливости и отсутствия предвзятости в обучаемых моделях [4].

Отдельное внимание должно быть уделено воспроизводимости и контролю версий. Каждое обновление библиотеки должно сопровождаться обновлением набора формальных спецификаций и прохождением полного цикла верификационных тестов. Это обеспечивает гарантии совместимости и корректности при использовании библиотеки в долгоживущих продуктах, в том числе в тех, что подлежат обязательной сертификации. Возможно использование системы механически проверяемых контрактов (design by contract) для автоматического контроля выполнения инвариантов при каждом вызове компонента.

Таким образом, разработка библиотеки машинного обучения с верифицированными компонентами представляет собой шаг к созданию по-настоящему надёжной экосистемы доверенного искусственного интеллекта. Такая библиотека может служить как основой для построения критически значимых систем ИИ, так и платформой для развития научных исследований в области формальной верификации алгоритмов обучения. В долгосрочной перспективе она способна сыграть роль стандарта де-факто в ситуациях, где недостаточно просто получить высокую точность – требуется гарантировать корректность, воспроизводимость и этическую прозрачность вычислительного процесса [5].

Список литературы:

1. Leroy, X. Formal Verification of a Realistic Compiler / X. Leroy // Communications of the ACM. – 2009. – Vol. 52, № 7. – P. 107–115. – DOI 10.1145/1538788.1538814.

2. Chlipala, A. Certified Programming with Dependent Types : a pragmatic introduction to the Coq proof assistant / A. Chlipala. – Cambridge : MIT Press, 2013. – 448 p.

3. Barthe, G. EasyCrypt: A Tutorial / G. Barthe, B. Grégoire, S. Zanella-Béguelin // Foundations of Security Analysis and Design VII. – Cham : Springer, 2013. – P. 146–166. – DOI 10.1007/978-3-319-10082-1_6.

4. Darulova, E. Daisy - Framework for Analysis and Optimization of Numerical Programs / E. Darulova, E. Horn, S. Sharma // Tools and Algorithms for the Construction and Analysis of Systems. – Cham : Springer, 2018. – P. 270–287. – DOI 10.1007/978-3-319-89960-2_15.

5. Bertot, Y. Interactive Theorem Proving and Program Development: Coq'Art / Y. Bertot, P. Castéran. – Berlin : Springer, 2004. – 469 p. – DOI 10.1007/978-3-662-07964-5.