Реализация механизма синхронизации в системах совместной разработки ML-решений

Решетков Андрей Эдуардович

Студент

МФТИ, Долгопрудный, Россия

E-mail: reshetkov.aeh@phystech.edu

Научный руководитель *— Хританков Антон Сергеевич*

В современном мире машинного обучения решение проблемы воспроизводимости экспериментов является важной задачей. Она становится ещё более актуальной в контексте систем совместной разработки машинно-обучаемых моделей, где коллектив исследователей может вносить изменения в один и тот же эксперимент. В прошлой работе [1] мы предложили использовать Conflict-free Replicated Data Type (CRDT) [3], чтобы обеспечить поддержание согласованного формального описания эксперимента у всех участников и возможность распределенного проведения исполняемых процедур в рамках графа эксперимента, построенного по описанию самого эксперимента [2].

В данной работе предлагается реализация данного подхода с использованием системы совместной разработки, такой как Git. Системы контроля версий также являются важной частью решения проблемы воспроизводимости экспериментов, поскольку они позволяют управлять изменениями в коде и контролировать его версионность. Предлагается расширить возможности систем контроля версий, добавив возможность управлять версиями данных эксперимента и исполняемых процедур, а также контролировать их корректность с помощью CRDT.

Вершина графа эксперимента может отвечать за конфигурацию эксперимента, либо же ей может соответствовать бинарный файл или файл некоторого программного модуля. Пользователь может изменять исходный эксперимент несколькими способами. Для каждого из них предложена стратегия генерации обновлений графа эксперимента, которые будут передаваться другим агентам. Эти способы включают: изменение бинарного файла, например, в результате запуска эксперимента; изменение программного модуля, на который есть ссылка в описании эксперимента; изменение файла конфигурации эксперимента (его описания в машиночитаемом формате).

Архитектурно решение состоит из трёх модулей. Первый модуль

решает задачу, описанную выше. Он по набору изменений описания эксперимента строит набор соответствующих им обновлений графа эксперимента. Второй модуль отвечает за обмен этими обновлениями между пользователями. Предлагается использовать для этого специально выделенную директорию в корне Git-репозитория, где у каждого агента будет отдельный файл со списком своих обновлений. Наконец, третий модуль получает на вход набор обновлений, сгенерированных другими агентами, и соответствующим образом преобразует их в изменения конфигурации и других файлов эксперимента.

Такой механизм позволяет разным агентам обмениваться своими обновлениями, не требуя при этом, чтобы их истории совпадали. Использование этой модели позволяет воспользоваться результатами, полученными в прошлой работе [1] для графа эксперимента. А именно, *Утверждение 1* о свойстве сильной согласованности показывает, что в результате применения описанных процедур агенты в конечном итоге получат один и тот же граф эксперимента, который третьим модулем будет преобразован в конфигурацию эксперимента, совпадающую для всех агентов. Таким образом, предложенная архитектура действительно позволяет решить исходную проблему синхронизации изменений, вносимых в эксперимент коллективом исследователей.

Литература

1. Решетков А., Хританков А. (2022). Обеспечение синхронизации в системах совместной разработки ML-решений // Интеллектуализация обработки информации: Тезисы докладов 14-й Международной конференции (с. 106-108).
2. Khritankov A., Pershin N., Ukhov N., & Ukhov A. (2022). MLDev: Data Science Experiment Automation and Reproducibility Software. // International Conference on Data Analytics and Management in Data Intensive Domains (pp. 3-18). Springer, Cham. 10.1007/978-3-031-12285-9_1
3. Marc Shapiro, Nuno Preguiça, Carlos Baquero, Marek Zawirski. Conflict-free Replicated Data Types. // SSS 2011 - 13th International Symposium Stabilization, Safety, and Security of Distributed Systems, Grenoble, France. pp.386-400, 10.1007/978-3-642-24550-3_29