

СРАВНИТЕЛЬНЫЙ АНАЛИЗ СОВРЕМЕННЫХ БИБЛИОТЕК И ФРЕЙМВОРКОВ МАШИННОГО ОБУЧЕНИЯ

Е.А. Аникеев¹, А.В. Акименко¹

¹ФГБОУ ВО «Воронежский государственный лесотехнический университет
имени Г.Ф. Морозова»

Аннотация. В статье представлен сравнительный анализ наиболее востребованных библиотек и фреймворков машинного обучения, используемых в современных исследовательских и промышленных проектах. Рассмотрены ключевые характеристики TensorFlow, PyTorch, Keras, JAX и MXNet, их архитектурные особенности, целевое применение и инструментарий для развертывания моделей. Особое внимание уделено вопросам выбора фреймворка в зависимости от специфики решаемых задач – от исследовательского прототипирования до промышленной эксплуатации. Проведен анализ вспомогательных инструментов экосистемы машинного обучения, включая ONNX, PyTorch Lightning и Fast.ai.

Ключевые слова: машинное обучение, фреймворки, библиотеки, TensorFlow, PyTorch, JAX, Keras, MXNet, сравнительный анализ, искусственный интеллект.

COMPARATIVE ANALYSIS OF MODERN MACHINE LEARNING LIBRARIES AND FRAMEWORKS

Е.А. Anikeev¹, A.V. Akimenko¹

¹Voronezh State University of Forestry and Technologies named after G.F. Morozov

Abstract. The article presents a comparative analysis of the most popular machine learning libraries and frameworks used in modern research and industrial projects. The key characteristics of TensorFlow, PyTorch, Keras, JAX, and MXNet, their architectural features, target applications, and model deployment tools are considered. Special attention is paid to the choice of framework depending on the specifics of the tasks – from research prototyping to industrial operation. The analysis of auxiliary machine learning ecosystem tools, including ONNX, PyTorch Lightning, and Fast.ai, is carried out.

Keywords: machine learning, frameworks, libraries, TensorFlow, PyTorch, JAX, Keras, MXNet, comparative analysis, artificial intelligence.

Введение

В современном мире искусственного интеллекта и машинного обучения очень важно правильно выбрать софт для расчётов. Существующих фреймворков много, и они образуют своего рода многоэтажную систему. Есть и проверенные временем решения, и специализированные инструменты под конкретные задачи. В этом исследовании мы разберём самые популярные библиотеки и фреймворки, которые сейчас используют в сфере ИИ [1–3].

Основные характеристики фреймворков машинного обучения

Возьмём TensorFlow от Google. Это целый комбайн для машинного обучения. У него есть своя среда разработки TensorBoard для визуализации, облегчённая версия для мобильных устройств TensorFlow Lite и инструмент для веб-интеграции TensorFlow.js.

Современные фреймворки для глубокого обучения пошли разными путями. TensorFlow делает ставку на модульность и на то, чтобы настраивать производственные конвейеры (MLOps). А PyTorch, благодаря динамическим вычислительным графам, остаётся королём в исследовательской работе – особенно в обработке естественного языка и генеративных задачах. У PyTorch в версии 2.x появился механизм `torch.compile`: можно спокойно прототипировать в удобном eager-режиме, а потом оптимизировать граф для развёртывания. Keras – это высокоуровневая обёртка, где всё завязано на композицию компонентов. Она ускоряет разработку, но не мешает вам при необходимости копать вглубь бэкенда (в основном TensorFlow). Такая разная специализация показывает общую тенденцию: сейчас выбирают фреймворк не просто потому, что он универсальный, а потому что он подходит под конкретный этап жизни модели и требования инфраструктуры.

Ещё есть JAX, тоже от Google. Его фишка – векторизация через `vmap` и распараллеливание через `rmmap`. Это позволяет выжимать максимум из ускорителей. А чистая функциональность упрощает проверку и воспроизводимость экспериментов. Правда, освоить JAX сложнее, чем более абстрактные решения. Зато он незаменим, если нужна и скорость вычислений, и математическая строгость.

Теперь Apache MXNet. Его поддерживает Amazon. Инструмент заточен под распределённое обучение и облачные решения. Чем он хорош? Масштабируемостью, поддержкой нескольких языков и тесной связкой с облаком AWS.

Правда, на сегодняшний день MXNet – это уже скорее наследие для новых проектов. Но на некоторых предприятиях его до сих пор используют в производстве.

Какой вывод? Выбор между этими инструментами зависит не только от технических характеристик, но и от контекста: на какой стадии жизненного цикла находится модель, как масштабируемость нужна, что умеет ваша команда и насколько легко вписать фреймворк в существующую инфраструктуру.

Дополнительные библиотеки

В дополнение к основным фреймворкам существуют специализированные библиотеки, расширяющие их функциональность. Например, PyTorch Lightning представляет собой инструмент для стандартизации процесса обучения моделей, который упрощает стандартные процедуры и оркестровку [2, 4]. Библиотека Fast.ai ориентирована на практическое применение машинного обучения и образование, делая сложные алгоритмы более доступными для начинающих специалистов. Особого внимания заслуживает ONNX (Open Neural Network Exchange). Это открытый формат обмена моделями между различными фреймворками. Он позволяет эффективно выполнять модели вне зависимости от того, в каком фреймворке они были созданы.

В таблице 1 представлены основные параметры рассмотренных фреймворков и библиотек машинного обучения.

Таблица 1 – Сравнительный анализ фреймворков машинного обучения

Фрейм-ворк	Разработчик	Ключевые характеристики	Целевое применение
TensorFlow	Google	Комплексная экосистема, TFX, TF Serving, TF Lite, зрелый MLOps	Промышленное применение, продакшен-пайплайны
PyTorch	Meta	Динамические графы, torch.compile, интуитивный Python-синтаксис	Исследовательские задачи, быстрое прототипирование
Keras	API TensorFlow	Простота использования, модульность, высокая продуктивность	Прототипирование, образование
JAX	Google	Функциональный подход, XLA-компиляция, автоматическая векторизация	Научные вычисления, высокопроизводительные исследования
MXNet	Apache/Amazon	Масштабируемость, многоязычная поддержка, облачная интеграция	Распределённое обучение, легаси-проекты

Заключение

Определение оптимального стека технологий для задач машинного обучения требует взвешенной оценки множества факторов: от предметной области и масштаба данных до компетенций команды и ограничений инфраструктуры. Универсального решения не существует – эффективность инструмента проявляется только в контексте конкретных требований к разработке, развёртыванию и сопровождению модели.

Современная практика всё чаще отходит от жёсткой привязки к единому фреймворку в пользу гибридных подходов, где разные инструменты используются на соответствующих этапах жизненного цикла модели [3, 4]. Для многих команд PyTorch становится де-факто отправной точкой: он сочетает достаточную гибкость для экспериментов, приемлемую производительность в продакшене и активно развивающееся сообщество, что в совокупности снижает совокупную стоимость владения решением. Однако финальный выбор должен опираться не на популярность технологии, а на её соответствие архитектурным и бизнес-требованиям конкретного проекта.

Список литературы

1. Возная, Е. Сравнение ML-фреймворков: PyTorch, TensorFlow, JAX – что выбрать в России? / Е. Возная // Компьютерра. – 2025. – 26 ноября. – URL: <https://www.computerra.ru/329636/sravnenie-ml-frejmvorkov-pytorch-tensorflow-jax-chto-vybrat-v-rossii/> (дата обращения: 19.02.2026).
2. Парамонов, А. А. Исследование структуры тропического циклона на основе анализа данных спутникового видеоряда с применением метода почти периодического анализа / А. А. Парамонов, А. В. Калач, А. С. Кравченко // Моделирование систем и процессов. – 2024. – Т. 17, № 4. – С. 67-76. – DOI 10.12737/2219-0767-2024-17-4-67-76. – EDN BOGSTY.
3. Nikolaev, N. Comparative Analysis of CNN Performance in Keras, PyTorch and JAX on PathMNIST / N. Nikolaev // arXiv. – 2025. – arXiv:2507.12248. – URL: <https://ui.adsabs.harvard.edu/abs/2025arXiv250712248N/abstract> (дата обращения: 19.02.2026).
4. Sanyadanam, A. Comparison of distributed Machine Learning frameworks in a fog environment: Conceptual and Performance analysis / A. Sanyadanam, S.N. Srirama // Internet of Things. – 2025. – Vol. 34. – Art. 101774. – DOI:

10.1016/j.iot.2025.101774. – URL: <https://www.sciencedirect.com/science/article/abs/pii/S2542660525002884> (дата обращения: 19.02.2026).

References

1. Voznaya, E. Sravnenie ML-freymvorkov: PyTorch, TensorFlow, JAX – chto vybrat' v Rossii? / E. Voznaya // Komp'yuterra. – 2025. – 26 noyabrya. – URL: <https://www.computerra.ru/329636/sravnenie-ml-freymvorkov-pytorch-tensorflow-jax-chto-vybrat-v-rossii/> (data obrashcheniya: 19.02.2026).
2. Paramonov, A. A. Issledovanie struktury tropicheskogo tsyklona na osnove analiza dannykh sputnikovogo videoryada s primeneniem metoda pohti periodicheskogo analiza / A. A. Paramonov, A. V. Kalach, A. S. Kravchenko // Modelirovanie sistem i protsessov. – 2024. – T. 17, № 4. – S. 67-76. – DOI 10.12737/2219-0767-2024-17-4-67-76. – EDN BOGSTY.
3. Nikolaev, N. Comparative Analysis of CNN Performance in Keras, PyTorch and JAX on PathMNIST / N. Nikolaev // arXiv. – 2025. – arXiv:2507.12248. – URL: <https://ui.adsabs.harvard.edu/abs/2025arXiv250712248N/abstract> (дата обращения: 19.02.2026).
4. Sanyadanam, A. Comparison of distributed Machine Learning frameworks in a fog environment: Conceptual and Performance analysis / A. Sanyadanam, S.N. Srirama // Internet of Things. – 2025. – Vol. 34. – Art. 101774. – DOI: 10.1016/j.iot.2025.101774. – URL: <https://www.sciencedirect.com/science/article/abs/pii/S2542660525002884> (дата обращения: 19.02.2026).