

РЕФЕРАТ

Сравнительный анализ фреймворков PyTorch и TensorFlow
как инструментов разработки программного обеспечения

Сazonova Игоря Евгеньевича
магистранта 1 курса
специальности 7-06-0533-05

Содержание

- Актуальность исследования
- Цель исследования
- Задачи исследования
- Объект исследования
- Предмет исследования
- Эволюция и современное состояние фреймворков
- Существующие сравнительные исследования
- Сравнительный анализ производительности
- Анализ инструментов развертывания и MLOps-экосистем
- Оценка сообщества и образовательных ресурсов
- Анализ промышленного внедрения и кейсов использования
- Сравнительный анализ совокупной стоимости владения
- Выводы по аналитическому обзору
- Разработка системы оценки фреймворков
- Комплексные результаты оценки
- Главные выводы

Актуальность исследования

Выбор фреймворка машинного обучения напрямую влияет на стоимость и успех ИТ-проектов. Конвергенция PyTorch и TensorFlow требует пересмотра их позиционирования. Существующие сравнения ограничены скоростью обучения и не учитывают полный жизненный цикл разработки (MLOps, сопровождаемость, отладка). Всесторонний анализ как комплексных инструментов разработки является своевременной и значимой задачей.

Цель исследования

Проведение сравнительного анализа эффективности фреймворков обучения глубокого обучения PyTorch и TensorFlow как инструментов разработки программного обеспечения и выработать практические рекомендации по их выбору для различных сценариев применения.

Задачи исследования

- систематизировать архитектурные особенности и методологии PyTorch и TensorFlow;
- проанализировать их производительность, экосистемы и инструменты развертывания;
- разработать методику комплексной оценки для разных сценариев использования (исследования, промышленное внедрение, edge-вычисления);
- провести экспериментальное сравнение и сформулировать практические выводы.

Объект исследования

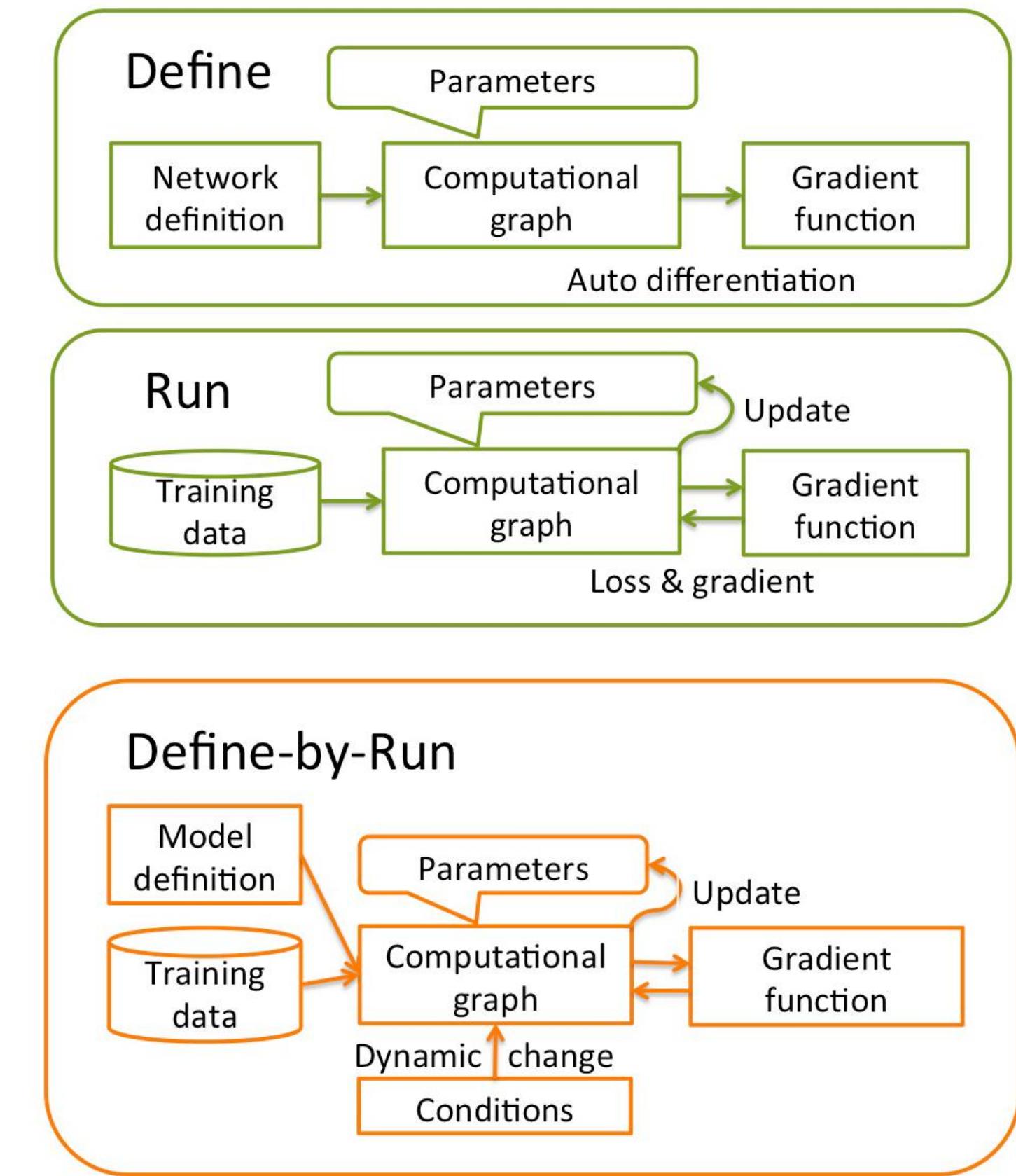
Фреймворки глубокого обучения PyTorch и TensorFlow, их архитектурные парадигмы и экосистемы.

Предмет исследования

Сравнительные преимущества и ограничения PyTorch и TensorFlow как комплексных инструментов разработки интеллектуальных систем.

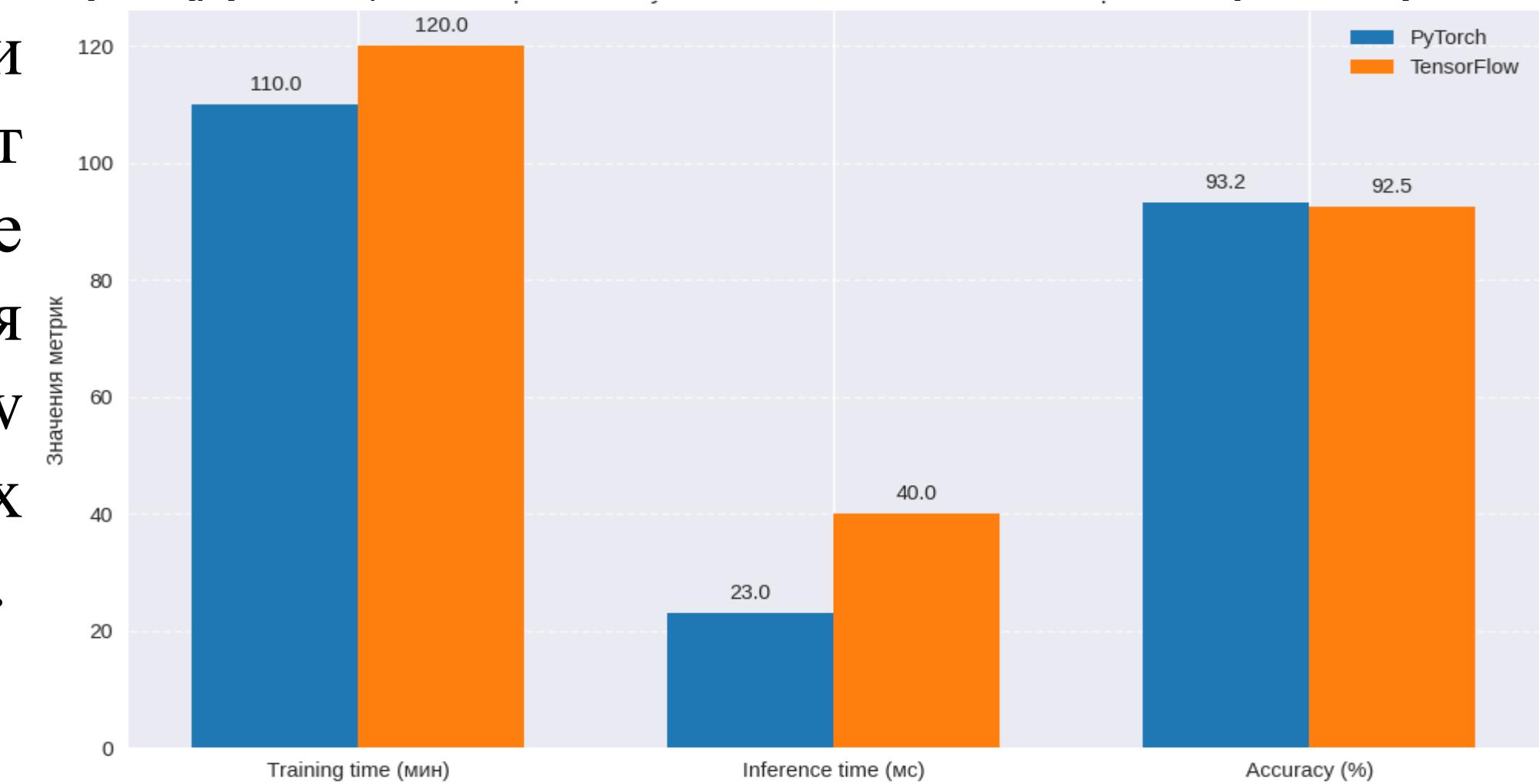
Эволюция и современное состояние фреймворков

Фреймворки глубокого обучения прошли путь от академических библиотек (Theano, Torch7, Caffe) к мощным промышленным платформам. TensorFlow (2015, Google) стал стандартом благодаря масштабируемости и экосистеме. PyTorch (2016, Meta) завоевал популярность в исследованиях за счёт гибкости и удобного интерфейса. Изначально различались подходом к вычислительным графикам: статические у TensorFlow и динамические у PyTorch. Со временем различия сгладились: оба предлагают гибридный подход — удобство разработки и оптимизацию для продакшена. Сегодня выбор определяется экосистемой и задачами команды, а не архитектурой графов.



Существующие сравнительные исследования

Бенчмарки для сравнения фреймворков глубокого обучения быстро устаревают из-за стремительного развития технологий, поэтому их результаты необходимо интерпретировать с учётом версии фреймворка, аппаратной платформы и сложности модели. Производительность обычно оценивается по времени обучения, инференса, точности и эффективности использования ресурсов, а также по качественным факторам: удобству разработки, документации и интеграции. Исследования дают противоречивые выводы: PyTorch чаще оказывается быстрее и удобнее для прототипирования, тогда как TensorFlow лучше подходит для масштабных моделей и промышленного применения.



Сравнительный анализ производительности

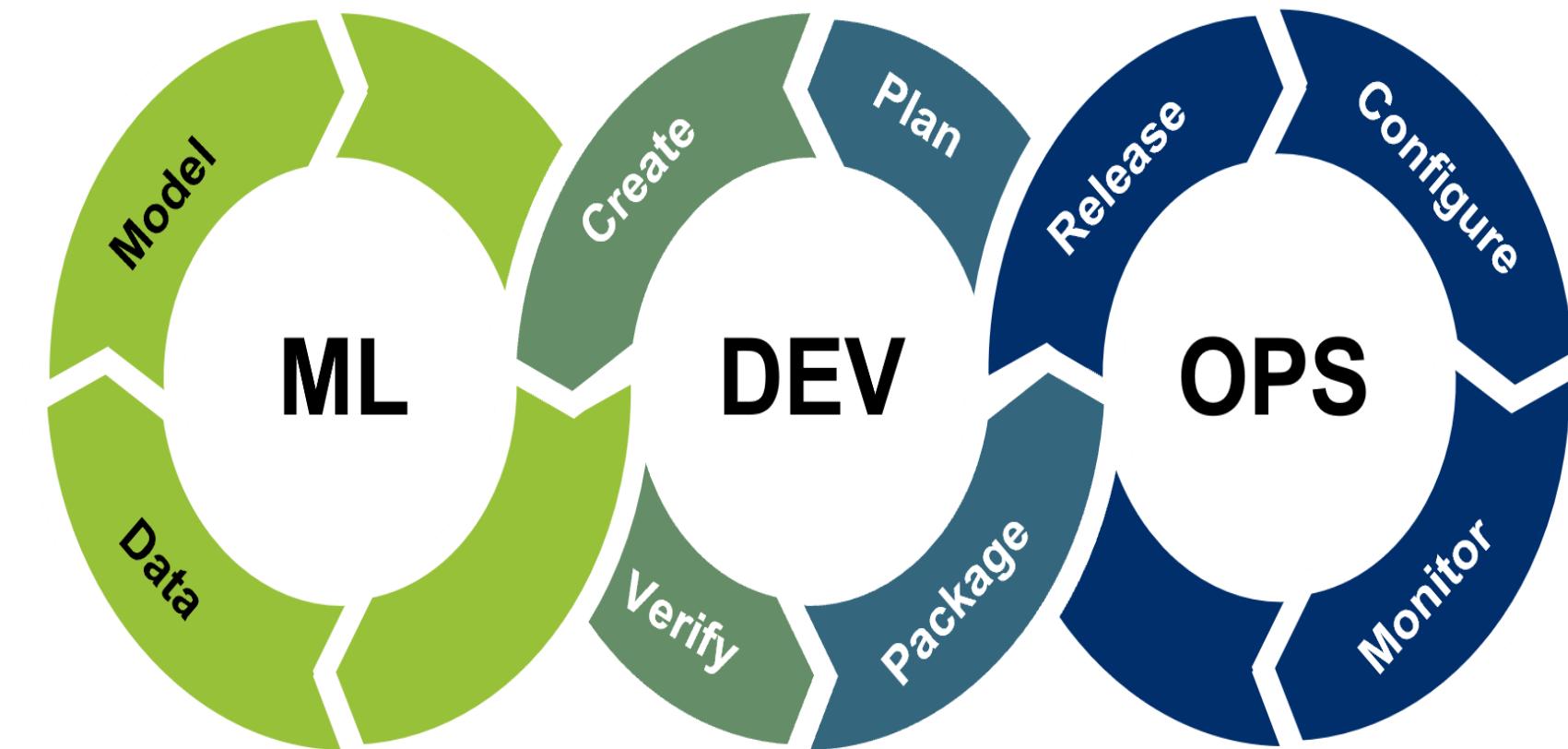
В целом различия по точности и энергоэффективности минимальны, оба фреймворка считаются зрелыми и надёжными инструментами.

Исследование	Модель	Аппаратная платформа	TensorFlow (Время)	PyTorch (Время)	Преимущество
Germino et al	NN	CPU (AMD Ryzen 5)	8.62 с	51.29 с	TensorFlow в ~5.9x
StackOverflow	NN	CPU (Intel i7-10700)	21 с	128 с	TensorFlow в ~6.1x
Germino et al	CNN	GPU (RTX 2070 Super)	28.21 с	58.48 с	TensorFlow в ~2.0x
Ba Alawi	CNN	CPU (Intel Xeon)	17.2 с	20.1 с	TensorFlow в ~1.2x
DEV Community	CNN	GPU (RTX 4090)	1.9 ч	2.0 ч	TensorFlow в ~1.05x
DigitalOcean	CNN	GPU (Nvidia V100)	2.3 ч	2.1 ч	PyTorch в ~1.1x
Florencio et al.	CNN	GPU (GTX 1050)	3.2 ч	2.7 ч	PyTorch в ~1.2x
Novac et al	CNN	GPU (GTX 1070 Ti)	21.95 ч	16.98 ч	PyTorch в ~1.3x

Исследование	Модель	Аппаратная платформа	TensorFlow (Энергопотреб.)	PyTorch (Энергопотреб.)	Преимущество
Germino et al	NN	CPU (AMD Ryzen 5)	65 Вт	72 Вт	+11% энергоэффект. TensorFlow
StackOverflow	NN	CPU (Intel i7-10700)	70 Вт	78 Вт	+11% энергоэффект. TensorFlow
Germino et al	CNN	GPU (RTX 2070 Super)	185 Вт	190 Вт	+3% энергоэффект. TensorFlow
Ba Alawi	CNN	CPU (Intel Xeon)	80 Вт	85 Вт	+6% энергоэффект. TensorFlow
DEV Community	CNN	GPU (RTX 4090)	310 Вт	305 Вт	+2% энергоэффект. PyTorch
DigitalOcean	CNN	GPU (Nvidia V100)	250 Вт	240 Вт	+4% энергоэффект. PyTorch
Florencio et al.	CNN	GPU (GTX 1050)	120 Вт	115 Вт	+4% энергоэффект. PyTorch
Novac et al	CNN	GPU (GTX 1070 Ti)	200 Вт	190 Вт	+5% энергоэффект. PyTorch

Анализ инструментов развертывания и MLOps-экосистем

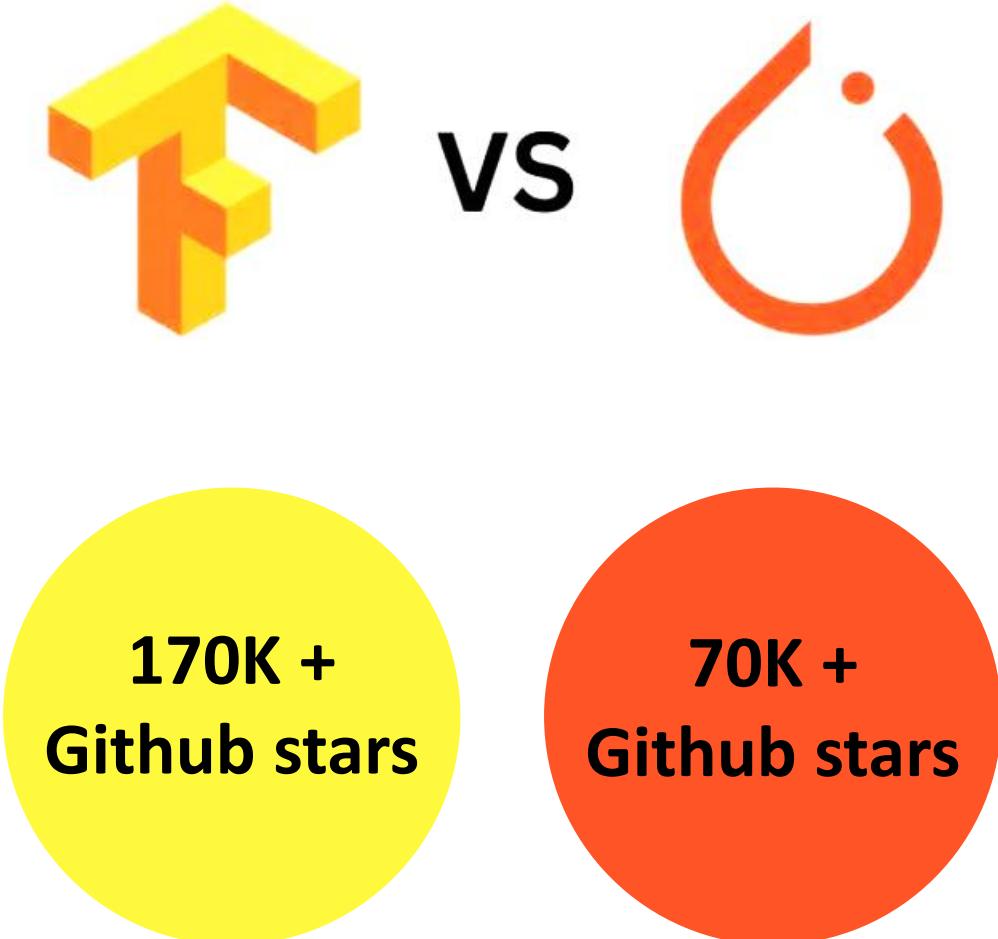
- TensorFlow традиционно сильнее в продакшене: TensorFlow Serving, TensorFlow Lite, TFX, интеграция с Google Cloud.
- PyTorch активно сокращает разрыв: TorchServe (AWS), PyTorch Mobile, поддержка ONNX для гибкого инференса.
- TensorFlow — комплексная end-to-end экосистема, PyTorch — более гибкая и открытая, удобная для интеграции.
- Выбор зависит от задач: масштабные облачные решения чаще на TensorFlow, гибкие прототипы и кросс-платформенные сценарии — на PyTorch.



Оценка сообщества и образовательных ресурсов

TensorFlow сохраняет лидерство в сообществе разработчиков благодаря длительной истории, поддержке Google и высокой популярности на GitHub — более 170 тысяч звёзд. В то же время PyTorch демонстрирует стремительный рост: в 2024 году количество вкладов увеличилось на 133%, а переход под эгиду Linux Foundation укрепил его позиции в индустрии. Оба фреймворка обладают качественной документацией и широким спектром обучающих материалов, включая курсы от Microsoft и интерактивные учебники. При этом PyTorch чаще выбирают исследователи и специалисты R&D благодаря интуитивному API, тогда как TensorFlow остаётся предпочтительным инструментом для MLOps-инженеров и промышленного внедрения.

TensorFlow vs PyTorch



Анализ промышленного внедрения и кейсов использования

TensorFlow сохраняет позиции в корпоративных системах (38% рынка), PyTorch лидирует в продакшене и обучении (55–63%).

PyTorch используется в OpenAI, Tesla, Airbnb, Genentech; TensorFlow — в Google, Snapchat.

В NLP доминирует PyTorch, в CV — паритет, в RL — преимущество TensorFlow.



Genentech



TensorFlow



PyTorch

Сравнительный анализ совокупной стоимости владения

- PyTorch ускоряет прототипирование и обучение, снижая вычислительные затраты.
- TensorFlow эффективнее на TPU и потребляет меньше энергии.
- TensorFlow выигрывает по операционным расходам благодаря зрелой MLOps-экосистеме.
- PyTorch чаще выгоден для исследований, TensorFlow — для промышленных решений и мобильных приложений.

PyTorch



Fast Prototyping



ROI



GPU

Training

COST



TensorFlow



Low Consumption



MLOps

BENEFIT

Выводы по аналитическому обзору

Сравнение TensorFlow и PyTorch показывает, что оба фреймворка являются зрелыми и мощными инструментами, способными решать сложные задачи глубокого обучения. Выбор между ними зависит от целей проекта, технической инфраструктуры и стратегических приоритетов команды. PyTorch отличается гибкостью, интуитивностью и особенно хорошо подходит для академических исследований, задач обработки естественного языка и нестандартных архитектур. TensorFlow, в свою очередь, демонстрирует высокую стабильность, глубокую интеграцию и является предпочтительным выбором для промышленного развертывания, мобильных решений и масштабных систем. PyTorch предоставляет разработчику полный «питонический» контроль, тогда как TensorFlow предлагает экосистему «всё включено» с множеством встроенных инструментов. Современные тенденции указывают на сближение подходов: растёт популярность гибридных графов, усиливается интероперабельность через ONNX и Keras 3.0, а технологии становятся доступнее для широкой аудитории.

Разработка системы оценки фреймворков

Она сочетает количественные метрики (время обучения, инференс, память, энергопотребление, скорость прототипирования) с экспертными оценками (удобство отладки, инструменты развертывания, документация, активность сообщества) и стратегическими факторами (roadmap, инвестиции, адаптивность). Весовое распределение: 60% объективные метрики, 30% экспертные оценки, 10% стратегическая устойчивость. Итоговый балл рассчитывается по нормализованным формулам, что позволяет перейти от субъективных суждений к количественно обоснованному выбору фреймворка.



$$Score = \sum_i (Norm(M_i) \times W_i) + \sum_j (Expert(E_j) \times W_j) + Future \times W_f$$

Комплексные результаты оценки

Метрика / Модель	TensorFlow	PyTorch	Лидер
Время обучения (часы)	ResNet50: 48.2 BERT base: 72.1 DCN: 15.3 ResNet50: 8.21 BERT base: 12.53 DCN: 3.14 ResNet50: 9.85 BERT base: 13.95 DCN: 6.05 ResNet50: 4.82 BERT base: 7.15 DCN: 1.48 Ошибки: 7.2 IDE: 7.8 Прототипирование: 7.5 Развертывание: 9.2 Мониторинг: 8.9 Платформы: 9.5	ResNet50: 46.8 BERT base: 70.3 DCN: 14.9 ResNet50: 7.89 BERT base: 11.82 DCN: 2.93 ResNet50: 9.41 BERT base: 13.32 DCN: 5.74 ResNet50: 4.65 BERT base: 6.92 DCN: 1.42 Ошибки: 8.5 IDE: 9.1 Прототипирование: 9.3 Развертывание: 7.8 Мониторинг: 7.5 Платформы: 8.1	PyTorch
Инференс (мс/образец)			PyTorch
Память (ГБ)			PyTorch
Энергопотребление (кВт·ч)			PyTorch
Экспертные оценки			PyTorch — удобство разработки TensorFlow — MLOps
Стратегическая устойчивость	84.4	87.5	PyTorch
Итоговые сценарии	A (прототип): 73.37 B (промышленное): 84.72 C (edge): 87.94	A (прототип): 90.28 B (промышленное): 82.15 C (edge): 79.23	PyTorch — прототипы TensorFlow — продакшен и edge

Главные выводы

В проведённом исследовании показано, что PyTorch оптimalен для научных проектов и быстрого прототипирования, а TensorFlow остаётся предпочтительным выбором для промышленных систем и edge-устройств благодаря зрелой MLOps-экосистеме. Разработанная методика оценки подтвердила свою практическую значимость, обеспечивая объективный выбор фреймворка с учётом как количественных метрик, так и стратегических факторов развития.

СПАСИБО ЗА ВНИМАНИЕ