Дизайн систем машинного обучения

7. Развертывание

План курса

1) ~~Практическое применение машинного обучения~~
2) ~~Основы проектирования ML-систем~~
3) ~~Обучающие данные~~
4) ~~Подготовка и отбор признаков~~
5) ~~Выбор модели, разработка и обучение модели~~
6) ~~Оценка качества модели~~
7) **Развертывание — Вы находитесь здесь**
8) Диагностика ошибок и отказов ML-систем
9) Мониторинг и обучение на потоковых данных
10) Жизненный цикл модели
11) Отслеживание экспериментов и версионирование моделей
12) Сложные модели: временные ряды, модели над графами
13) Непредвзятость, безопасность, управление моделями
14) ML инфраструктура и платформы
15) Интеграция ML-систем в бизнес-процессы

I apologize for using English

- Оставил многие термины на английском
- Эта лекция сложнее других
- Решил, что так будет понятнее

Развертывание модели

- Где конкретно будет работать ML-модель
- Когда будут рассчитываться предсказания модели
- Как модель получит запрос пользователя
- Как пользователь получит ответ модели

Offline vs Real-time data

- Offline data у вас уже есть
  - Можете очистить, преобразовать
  - Заранее посчитать эмбеддинги
  - Пользователь прислал данные, можно обработать их потом
- Real-time data
  - Пользователь прислал данные, нужно обработать их сейчас

Online vs Batch vs Streaming processing

- Online Processing (by demand)
  - Пользователь присылает один запрос
  - Модель делает предсказание
- Batch processing
  - Собираем много запросов
  - Обрабатываем их вместе
- Streaming processing
  - Пользователь постоянно присылает данные
  - Модель постоянно выдает предсказания

Cloud vs Edge computing

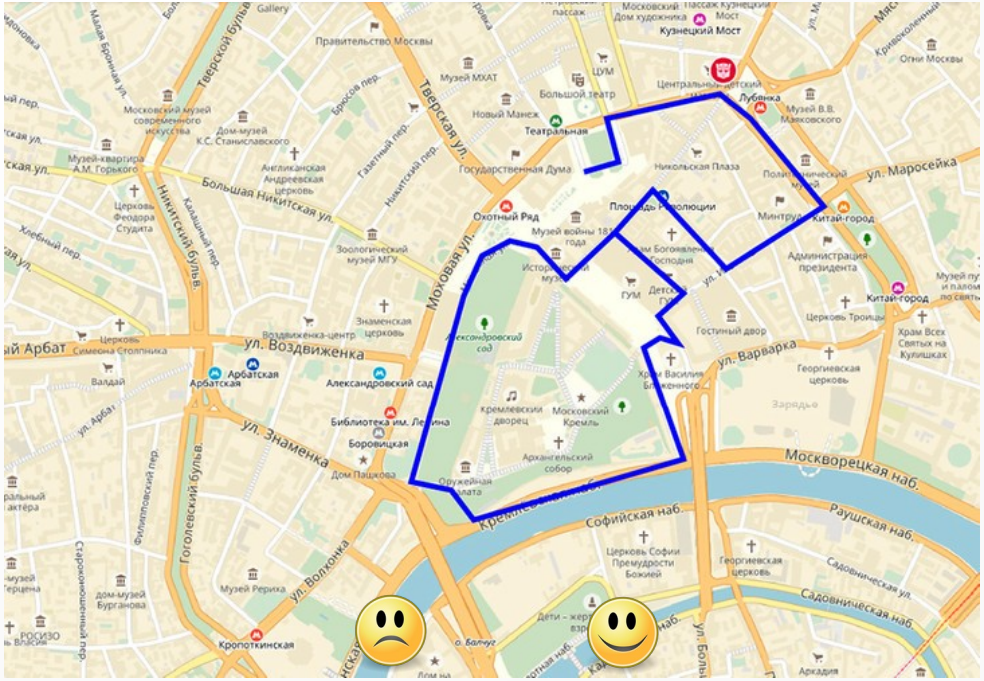
- Cloud computing
  - Модель работает на выделенных серверах
    в облаке или вашем датацентре, постепенно разница стирается
- Edge computing →
  - Модель работает на конечном устройстве
  - Например, на телефоне пользователя
  - Или на компьютере пользователя
  - Или внутри «умной колонки»
  - Или внутри камеры видеонаблюдения
  - Или в браузере - см Tensorflow JS

Задача

- Делаем приложение, рекомендующее маршрут для прогулки
- На основе данных
  - Текущее местоположение, история местоположений
  - Профиль пользователя, календарь пользователя
  - Прогноз погоды, прогноз дорожных пробок
  - Новости
  - История рекомендаций и оценок
- Предлагаем маршут прогулки

Пример: Маршрут прогулки

- Запускаем
- Видим карту
- Нравится:
  запускаем навигатор
- Не нравится:
  предлагаем еще



https://www.elle.ru/stil-zhizni/10-idealnyih-peshehodnyih-marshrutov-po-moskve/

Варианты развертывания модели

- Онлайн рекомендации (по запросу):
  - Локально, телефон рассчитывает маршрут
  - Запрос на сервер, сервер рассчитывает маршрут
- Пакетная рекомендация
  - Заранее считаем 10 рекомендаций для каждого пользователя
  - Загружаем на телефон при старте приложения
  - Или отдаем по запросу, как будто это онлайн рекомендация

Пример: Локальные рекомендации

- Безопасно — данные пользователя не передаются
- Может работать без интернета
- Дешево — пользователь сам покупает телефон

- Ограничения по вычислительной мощности
- Ограничения по месту
- Нет доступа к нашим данным на сервере
- Трудно обновлять модели

Пример: Онлайн рекомендации

- Большая пропускная способность:
  можем купить много серверов
- Сохраняем в тайне наши модели
- Собираем статистику и разметку

- Нужно платить за сервера
- Неравномерная нагрузка — все идут гулять в обед и вечером
- Если связь медленная, пользователю придется ждать

Пример: Пакетные рекомендации

- Сервер отвечает быстро
- Задержка модели не так важна — считаем все заранее

- Приходится считать много рекомендаций
- Много места для хранения рекомендаций
- Большая часть из них окажется не нужна
- Не можем мгновенно учитывать изменения — например, дождь, перекрытые дороги, аварии

Пример: Потоковые рекомендации

- Координаты передаются непрерывно
- Маршрут перестраивается на ходу
- Учитываем все новые данные

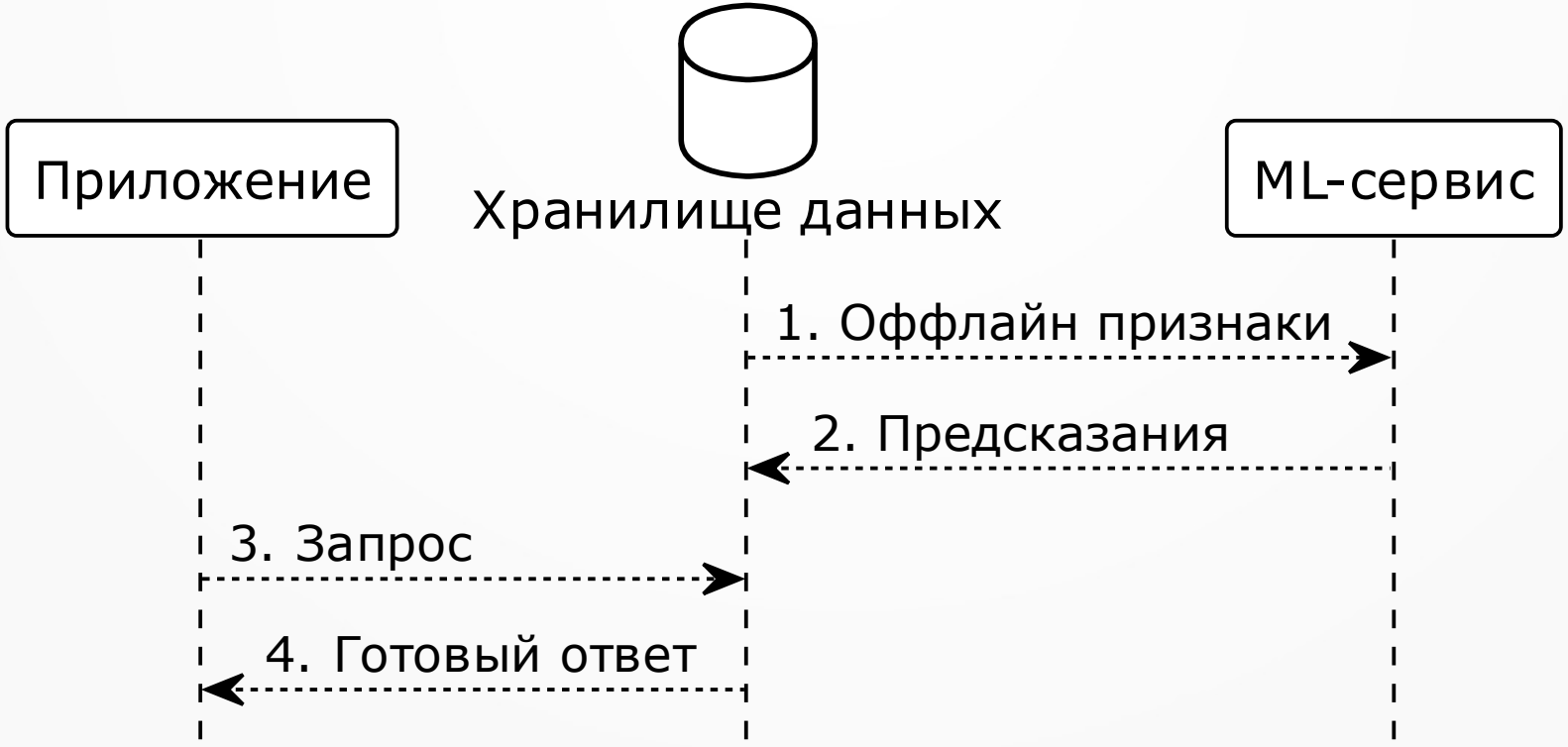
- Большая нагрузка на сеть
- Большая вычислительная нагрузка

Шаблоны реализации

- Online Processing, offline data
- Online Processing, offline data + real-time data
- Streaming processing, offline data + real-time data

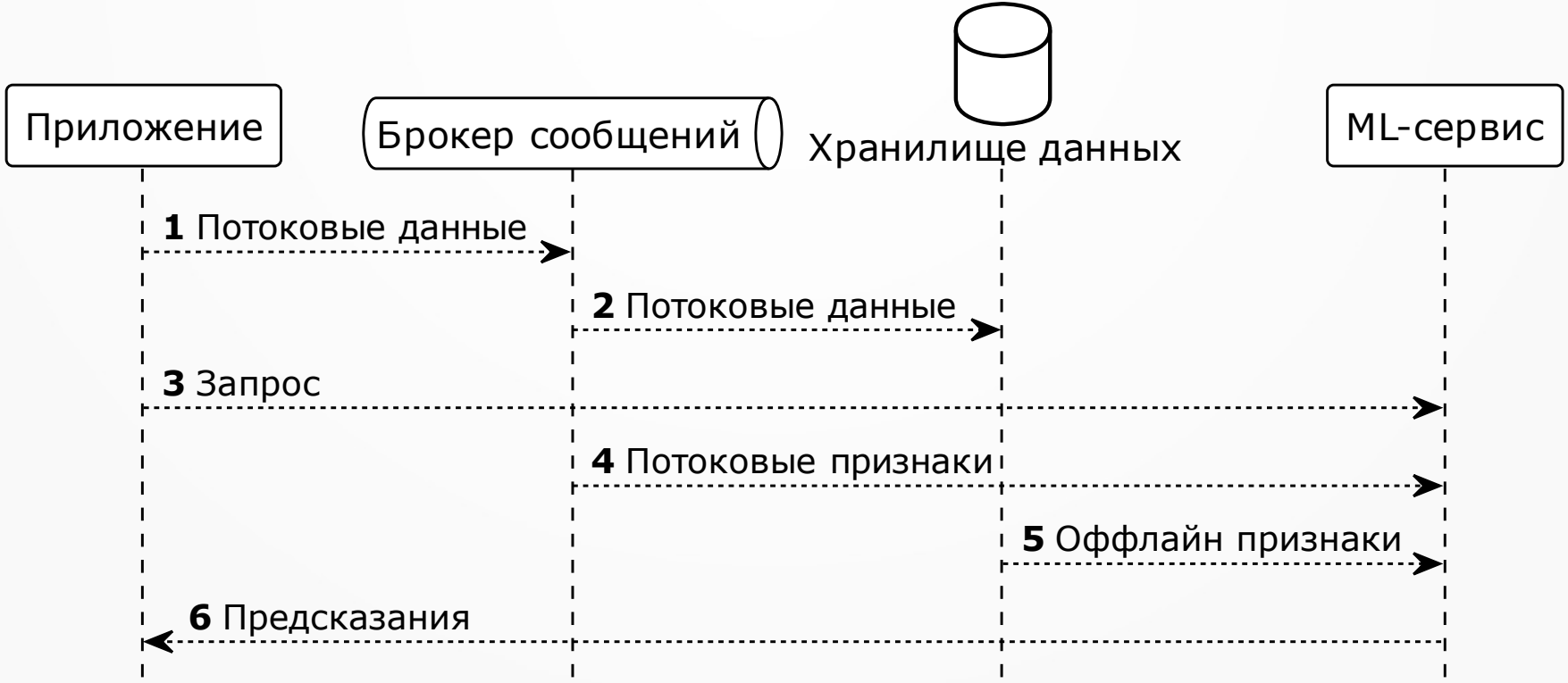
- Online Processing, real-time data
- Streaming processing, real-time data

Пакетная обработка

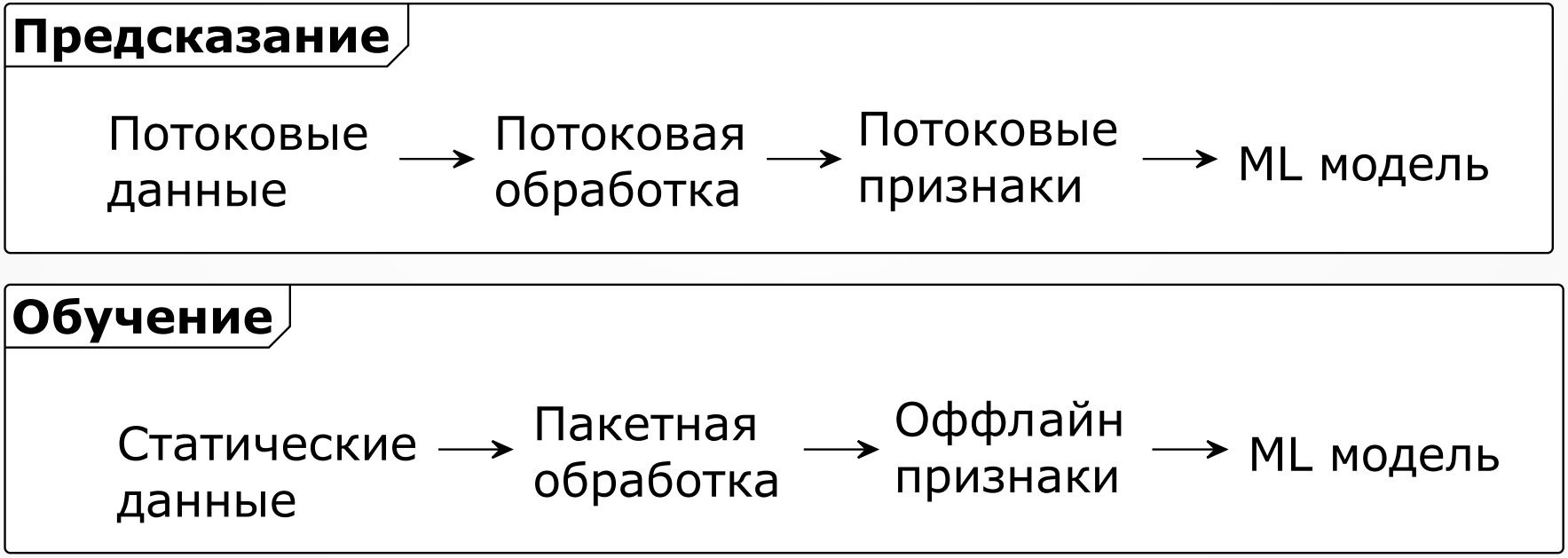


Онлайн предсказания

Потоковая обработка



Пайплайны обучения и инференса



Batch prediction

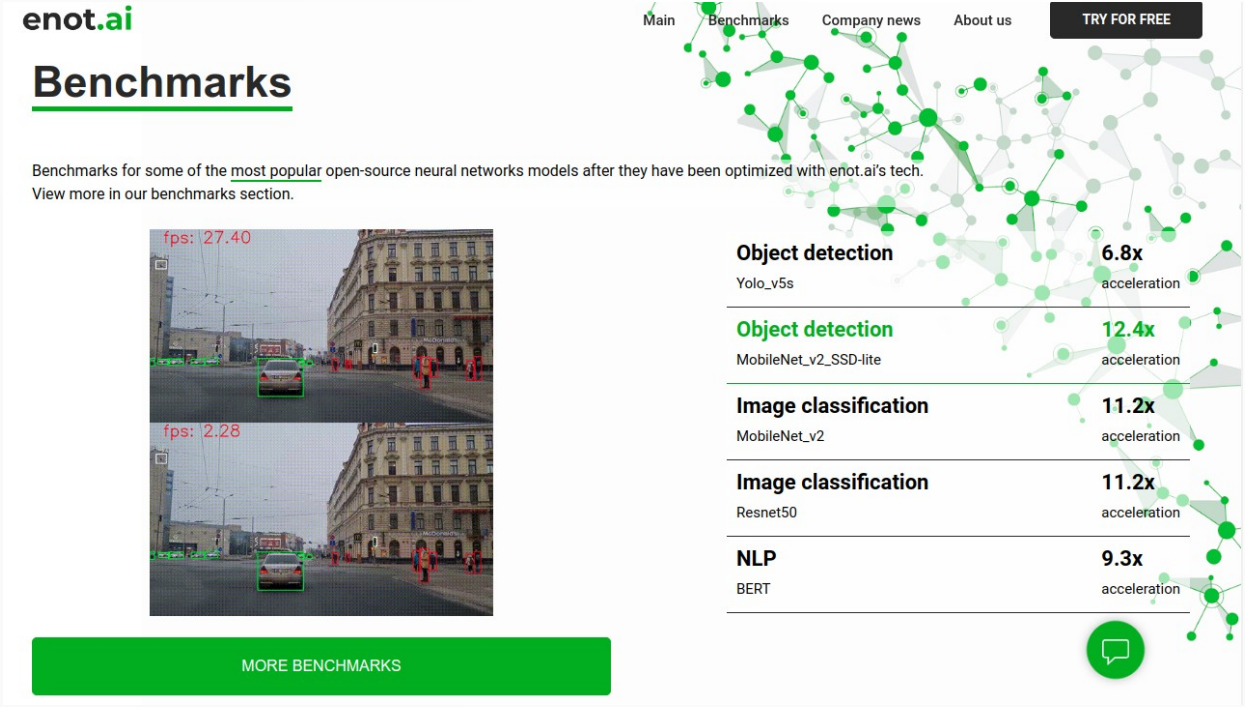
- Faster
- Sometimes cheaper
- Rarely better

- Быстрее сети связи и компьютеры → онлайн предсказания
- Но — гигантские нейронки?
- Thinking, Fast and Slow, by Kahneman — оба подхода полезны

Ускорение работы моделей

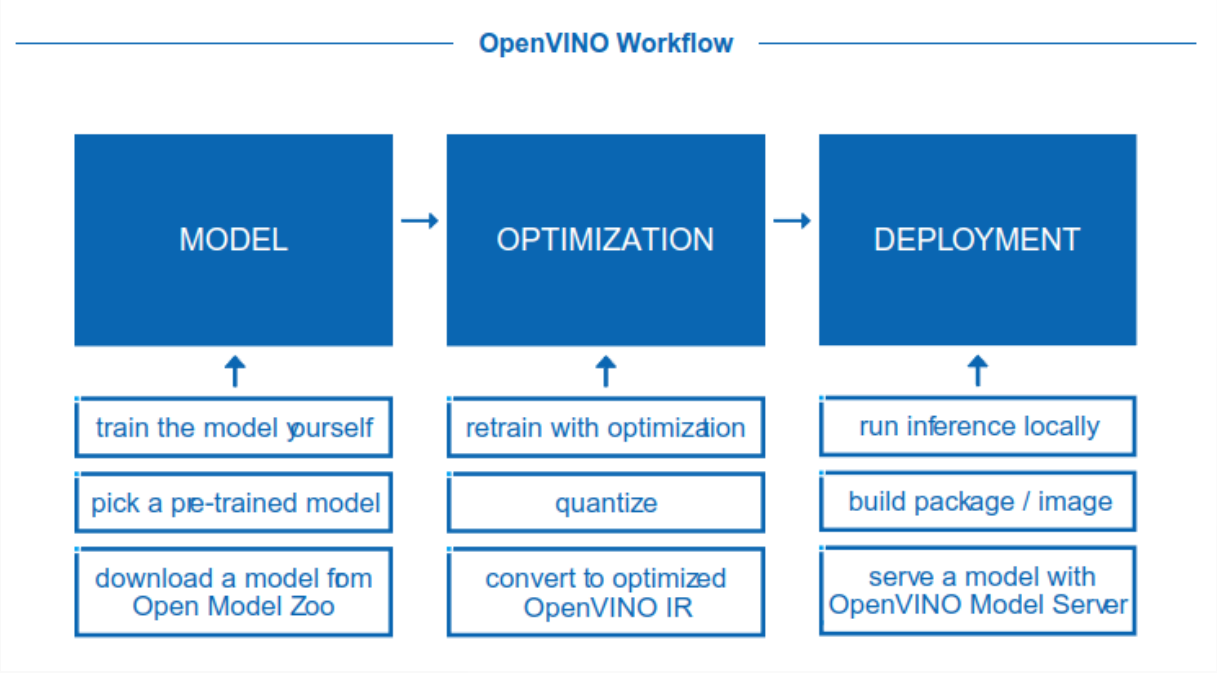
- Model Compression
- Low-Rank Factorization
- Knowledge Distillation
- Prunning
- Quantization

ENOT.AI



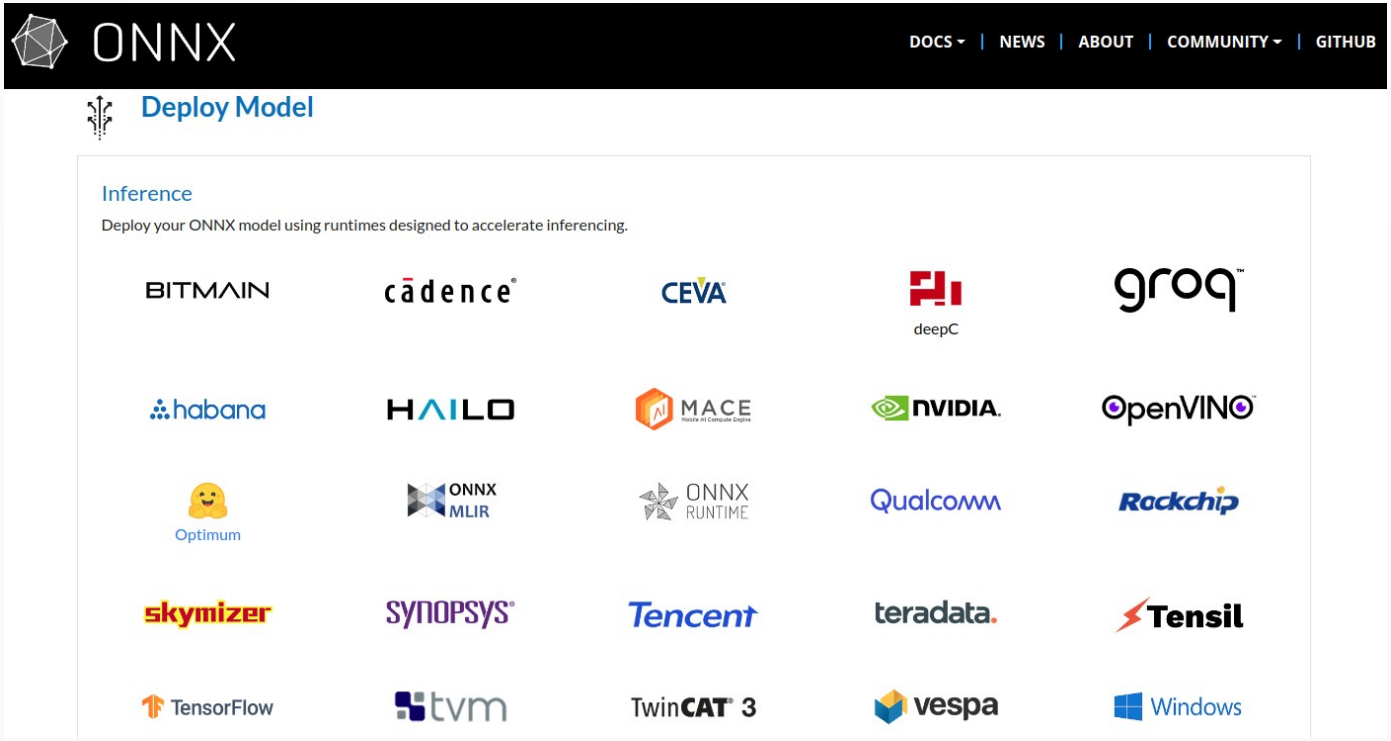
https://enot.ai/

OpenVINO



https://github.com/openvinotoolkit/openvino

ONNX



https://onnx.ai/

Маленькие хитрости для PyTorch

- PyTorch nvFuser
  - JIT
- Hugging Face Accelerate
  - Распределенный инференс
- DeepSpeed
  - Распределенный инференс
  - Оптимизированные для инференса CUDA Kernels
  - MoQ quantization

NVIDIA Triton Inference Server

NVIDIA Triton

- What is NVIDIA Triton?
- Benefits
- Functionality
  - **Scalable AI**
  - Model Orchestration
  - LLM Inference
  - Model Analyzer
  - Tree-based Models
- Ecosystem Integrations
- Success Stories
- Resources
- Inference Newsletter

Explore the benefits.

Support for multiple frameworks.

Triton supports all major training and inference frameworks, such as TensorFlow, NVIDIA® TensorRT™, PyTorch, MXNet, Python, ONNX, XGBoost, scikit-learn, RandomForest, OpenVINO, custom C++, and more.

High-performance inference.

Triton supports all NVIDIA GPU-, x86-, Arm® CPU-, and AWS Inferentia-based inferencing. It offers dynamic batching, concurrent execution, optimal model configuration, model ensemble, and streaming audio/video inputs to maximize throughput and utilization.

Designed for DevOps and MLOps.

Triton integrates with Kubernetes for orchestration and scaling, exports Prometheus metrics for monitoring, supports live model updates, and can be used in all major public cloud AI and Kubernetes platforms. It's also integrated in many MLOps software solutions.

An integral part of NVIDIA AI.

The NVIDIA AI platform, which includes Triton, gives enterprises the compute power, tools, and algorithms they need to succeed in AI, accelerating workloads from speech recognition and recommender systems to medical imaging and improved logistics.

https://developer.nvidia.com/nvidia-triton-inference-server

Дополнительные материалы

- Introduction to streaming for data scientists
- MLPerf Inference Benchmark
- MLPerf Inference Results
- How We Scaled Bert To Serve 1+ Billion Daily Requests on CPUs