

Воспроизводимость, трекинг и сравнение экспериментов

Глеб Ерофеев

DMIA Production ML  весна 2021

Почему нужна воспроизводимость

- Наука и научный метод
- Постановка эксперимента в науке
 - Дизайн эксперимента
 - Принятие/непринятие гипотезы или новый эксперимент
- Как это применимо в реальных ML системах

Примеры когда воспроизводимость выручает

- Пример: Выход на продакшен
 - Создали и обучили модель
 - Проверили локально - все ок
 - Выкатили на прод - все запускается
 - Пользователи не довольны - не то
 - Сравниваем локальную модель с продакшен - результаты по метрике разные
- Пример: Проблема в продакшен
 - Новый сценарий, которого не было при создании модели. На нём модель ломается в продакшен.
 - Что бы исправить, нужно воспроизвести и отладить проблему в Dev окружении

Что мешает воспроизводимости и как с этим бороться

- Фиксируем данные (на которых учим модель)
- Фиксируем random state - везде где это применимо
- Фиксируем фичи (и пайплайн их подготовки)
- Фиксируем модель (обучение и инференс)
- Фиксируем код обучения и инференса
- Фиксируем окружение (библиотеки и их версии)

NB: некоторые операции в GPU CuDNN не детерминированны по умолчанию - эту настройку необходимо фиксировать

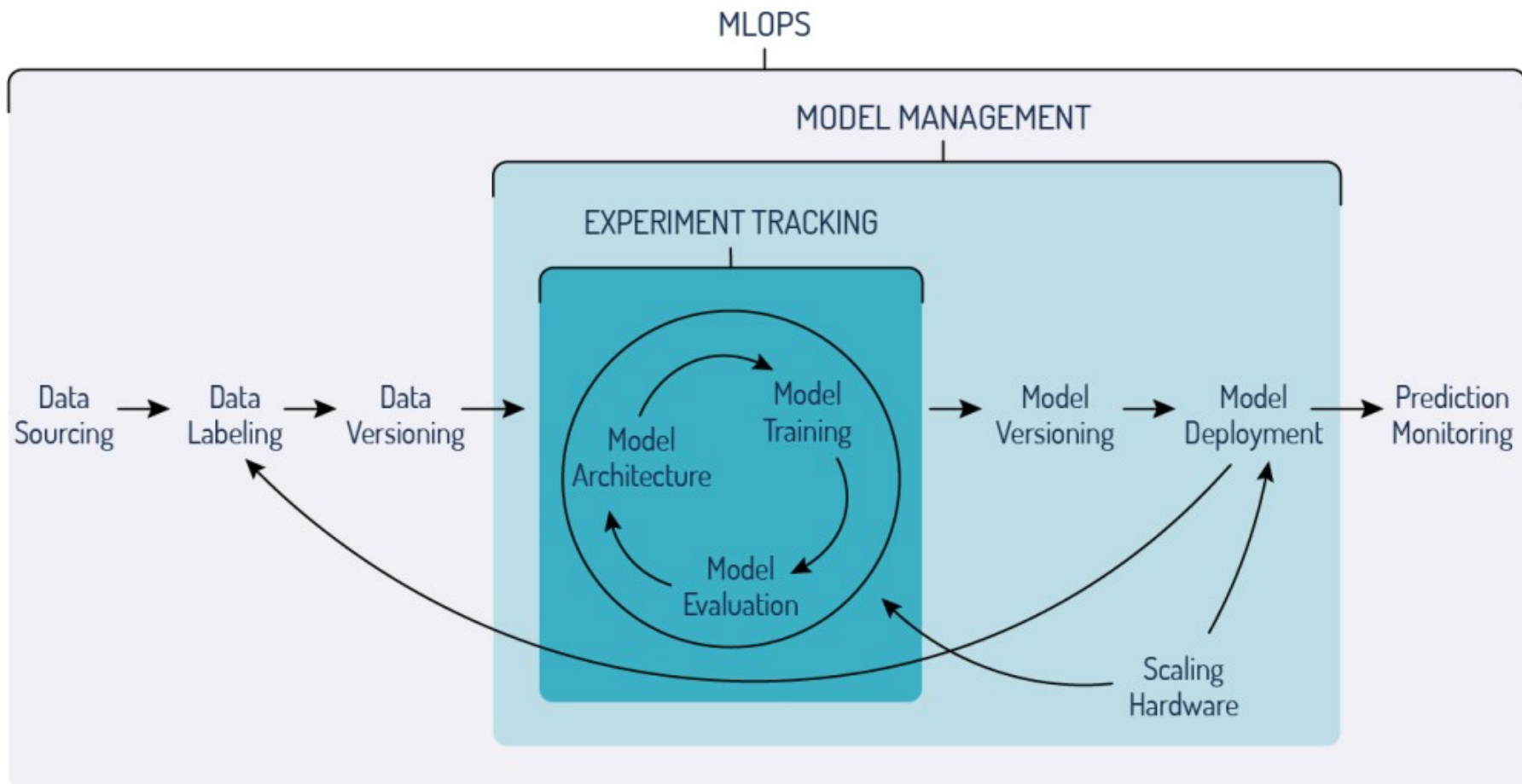
(<https://towardsdatascience.com/reproducible-model-training-deep-dive-2a4988d69031>)

- `pytorch torch.set_deterministic()`
- `tensorflow`

```
pip install tensorflow-determinism; export TF_DETERMINISTIC_OPS=1
```

- Особенности работы на кластере

MLOps vs Управление моделями vs Управление экспериментами



Инструменты помогающие обеспечить воспроизводимость

- [Optuna](#) - нет визуализации, удобно для персонального использования
- [MLFlow](#) - если готовы к DevOps
- [Neptune](#) - Хорошее решение из коробки (для команды нужна платная версия)
- [W&B](#) - Хорошее решение из коробки (сравнимо с Neptune)
- [Tensorboard](#) - Для персонального использования
- [Kuberflow](#) - если вы на GCP
- [SageMaker](#) - если вы на Amazon

Сравнение

Пример использования

```
# Регистрируемся на neptune.ai, получаем токен
!pip install neptune-client

import neptune
neptune.init(project_qualified_name='my_workspace/my_project',
             api_token='', Ваш токен
             )

# Определяем параметры - например для RFC из sklearn:
params = {'n_estimators': 10,
          'max_depth': 3,
          'min_samples_leaf': 1,
          'min_samples_split': 2,
          'max_features': 3,
          }

# Создаем эксперимент
neptune.create_experiment(name='my_study', params=params)

# Пришем метрику
neptune.log_metric('train_f1', train_f1)
neptune.log_metric('test_f1', test_f1)
```

Резюме

- Воспроизводимость
 - Данные
 - Рандом стейты
 - Код
 - Гиперпараметры
 - Результат - метрики
- Трекинг экспериментов
 - Оценка качества моделей - метрики и параметры
 - Подбор гиперпараметров
 - Тестирование моделей в стейдж и продакшен

Семинар

- Регистрируемся на neptune.ai
- Получаем код
- Воспроизводим эксперимент на готовом ноутбуке (очистка + TFIDF + LogReg)
- Расширяем возможности трекинга
- Задание - анализ и подбор параметров