Model Development

Page • 1 backlink • uni

Model Development – "соорудить модель"

Уже имеем данные, с помощью которых можем моделировать зависимость, нужно выбрать наилучшую модель, их обобщающую

Помимо DL это про:

- заказ железа и оценку ресурсов на обучение
- профилировка пайплайна обучения (как огранизовать производство? процессор тоже по сути конвейер)
    - оптимизация ресурсов – пока модель обучается, делаем следующий процесс обучения быстрее

Разработка модели

- пихаем заглушку вместо модели
- простые модели (бейзлайн для оценки качества) – линейные и бустинги
- оптимизация простых моделей
- сложные модели

Пайплайн

- какая задача?
- какие данные есть? (источники)
- Определение моделей и факторов (иногда грань между моделью и фактором очень тонкая)
- обучение
- внутренняя оценка
- оценка качества

Есть таскономия моделей, но есть процесс наложения модели на доменную область

Пример: модель, определяющая, какое приложение пользователь тыкнет следующим? Появилось новое приложение → модель необходимо переобучать

Отступление в Ranking:

- можем предсказывать регрессионно релевантность запроса (никак не учитываем запрос)
    - добавим c(q) в ошибку, но от нее результат не зависит
- предсказываем (F(d) - t - c(q))^2 **(почитать про QRMSE)**
- модель Бредли-Терри

На примере поиска есть два игрока: поиск и реклама, первые хотят выдать самые релевантные документы, вторые хотят показать побольше рекламы

Бизнесу важно, чтобы пользователи не разбежались, соответственно, баланс поиска и рекламы нужно искать и определять динамически

Другой пример: добавление этичности в модель (приходится переобучать модель и зачастую делать это часто с потенциальной потерей качества)

Multi-table training

Редко данные помещаются полностью в одну табличку, как в Kaggle

Задача: оптимизировать Q(F), где F – функция, которую мы обучаем, а Q – недифференцируемая функция

Стратегии:

- через запрос в Spark собираем датасет и учим модель, которые обучает модели на нем оптимальным образом
- сейчас переходят к модели, где есть отдельные потоки данных, которые можно добавлять в обучение – берем их с весами (каждый раз берем случайный пример из выбранного, как сэмпл мультиномиального распределения, потока)
    - веса можно подбирать в Spark, а можно создать механизм, позволяющий корректировать эти веса и запустить множество экспериментов

Почему не автоматический подбор весов (какой-нибудь байесовский оптимизатор)? Так выбиваем последние попугаи, но к этому моменту уже нужно понимание предметной области

Невероятностное сэмплирование

- "работаем с тем, что есть" (convenience sampling)
- итеративно дособираем (snowball sampling)
- эксперты говорят, что добавлять (judgement sampling)
- сами выбираем без рандомизации (quota sampling)

Stratified Sampling – балансируем данные (взвешиваем) по критерию, например, по таргету – может сильно повлиять на финальное качество решения

Reservoir Sampling – считываем, пока не надоест, гарантируем, что сэмпл всегда случайный (есть алгоритм)

Importance Sampling – сэмплирование из сложного распределения: берем примеры из альтернативного распределения, отношение плотностей используем как вес точки

Feature Engineering

Несмотря на технический прогресс, им необходимо заниматься

Входные данные

- неструктурированные – нейронки
    - текстовые данные
        - bag of words
        - "легковесные" модели на текстах и лейблах
            - наивный байес
            - BM25
        - эмбеддинги
            - as is (так делать не стоит)
            - через косинус
- структурированные
    - порядковые признаки
    - категориальные признаки
        - one-hot encoding (sparse решение, медленно работает, особенно без нативной поддержки)
        - category-based E(F(d) | c(d))
        - label-based statistics E(F(d,y) | c(d))
            - для бинарной классификации есть теоретическое определение, что надо использовать E(y | c(d))
            - label-encoding with time – p(like | history), значение берем из байесовской статистики
        - жадный поиск комбинаций (id пользователя + любимая группа – хороший сигнал, а по отдельности нет)
        - hashing trick

Данные могут теряться по разным причинам

- missing not at random ("бонусные" данные, которых обычно нету)
- missing at random (другая переменная влияет на вероятность потери)
- missing completely at random (факторы перестали считаться на некотором временном промежутке)

Как заменять? Зависит от типа модели (не нужно всегда ставить N/A)

- фактор + есть ли фактор
- бинаризация + бакет для отсутствующих

Утечки данных

- time leakage (было на прошлых занятиях)
- data statistics leak, например, scale leakage (через статистики – пример: распределение близко к нормальному, а параметры посчитаны по всему датасету)
- data duplication
- group leakage

Проведение и сопровождение экспериментов

Weights & Biases, Tensorboard и так далее

- кривая функции потерь
- недифференцируемые метрики, E2E-метрики
- логирование последних предсказаний с таргетами
- скорость работы модели
- перф системы
- изменение параметров и гиперпараметров

Баги и проблемы

- нарушение теоретических гарантий (например, сделали лишние предположения о входных данных или, наоборот, недостаточно исследовали)
- (?)
- плохой подбор гиперпараметров
- проблемы с данными
- плохой выбор факторов

- случайная расходимость (градиенты взрываются, плохая инициализация)

Recipe for training NN (Karpathy)

- нужно смотреть на то, как прокачивается тензор, сохраняется ли предположение о том, какая применяется инициализация
- ошибки в архитектуре (positional encoding по оси батча тоже сработает)
- забыть EOS токен в обучении или при инференсе
- и многие другие
- Become one with the data

Запуск проекта

- не можем запомнить батч → уже плохо → иначе пробуем запомнить датасет, а дальше датасет с аугментациями
- фиксируем случайный сид
- мониторим обучение в интерпретируемой форме
- читаем таргеты

Обучение

- оптимизация руками (оптимизатор и шедулеры – это хорошо, но чистый SGD вначале может быть лучше, или делать ручное управление learning rate)
- воспроизвести open-source (действительно должен быть результат)
- *интерпретируемая эпоха (сколько токенов съели, сколько аудио "прослушали")*