

Continual Learning

Page • 1 backlink • Tag

Говорили про мониторинг, но не про то, как адаптировать саму модель под изменяющиеся данные

Для чего хорошо непрерывное обучение:

- natural labels
- short feedback loops

| Например, рекомендательные системы и прочее (?)

Простой пример: делаем реплику текущей модели, дообучаем на актуальных данных, сравниваем с текущей и принимаем решение о замене

Реализация:

- stateless retraining (переобучение) – каждый раз обучаем модель с нуля (окна данных могут пересекаться)
- stateful training (дообучение) – дообучаем на актуальных данных (существенно быстрее, но меньше попугаев)
- золотая середина: например, в течение дня делать доливы, а ночью переобучаться по новой

Обновления

- model iteration – завезли новую архитектуру, добавили новые факторы (тяжело автоматизировать)
- data iteration – долили данные (легко внедрять)
- feature iteration – пересчет онлайн-фичей (или сами факторы – это статистические онлайн-модельки, вроде даже линейных регрессий)

Как часто нужно обучать модели? Очень редко модели дообучаются на каждом сэмпле, чаще всего используются микро-батчи

Гарантия частой перевыкатки модели – хорошая инженерная задача, но редко необходимая

Проводим набор экспериментов: ориентируемся на бизнес-метрики и следим за изменением от отсутствия выкаток в течение дня, недели, месяца...

Шаги непрерывного обучения:

- ручное stateless переобучение
- автоматическое переобучение (освобождаются ресурсы команды, которые тратятся на рутинные задачи) – нужно настроить шедулер, хранилище данных и моделей
- автоматическое дообучение
- непрерывное обучение – дообучаемся в онлайн каждые несколько минут, отказываем в приемке припросадке технической метрики

Непрерывное обучение нужно тестировать внутри команды, но такое тестирование может привести к завышенным ожиданиям; в онлайн используем канареечное тестирование

Interleaved Experiments – смешиваем рекомендации двух моделей и смотрим, по ответам какой кликают чаще (особенно полезно для рекомендательных систем, но может получиться так, что совместно модели ведут себя лучше)

Многорукие бандиты – показываем ответы разных моделей, после множества тестов статистически определяем, какая лучше. Проблема: придем к моменту, когда отличия будут на тысячные доли процента, но эти доли все равно будут приносить большие деньги

Алгоритмы для многоруких бандитов:

- explore-then-commit
- epsilon-greedy
- confidence bounds