https://habr.com/ru/companies/runity/articles/855308/

Сергей Рыжков

01.04.2025

Обучение ИИ-моделей на облачных серверах

Привет, Хабр! С вами Сергей Рыжков, руководитель департамента хостинга и профессиональных сервисов Рег.ру, и Александр Михеев, ML-engineer РБК. В этой статье расскажем, как мы автоматизировали процесс тегирования редакционных материалов РБК с помощью нейросети в облаке Рег.ру. О первых результатах нашего эксперимента читайте под катом.

Зачем начали обучать нейросеть

Тегирование материалов — неотъемлемая часть работы любой медиакомпании. В РБК ее выполняли вручную: редакторы прописывали 2–3 тега для каждого материала. В связи с этим возникало несколько трудностей:

- Количество тегов неконтролируемо увеличивалось — в основном разделе РБК до внедрения системы их набралось десятки тысяч.
- Появились дубли и синонимичные теги. Например, тег «сыр» можно написать по-разному: использовать заглавные буквы или кавычки. С точки зрения системы «Сыр», «сыр» и «сыры» — разные теги.
- Из-за большого числа тегов стало сложнее выбирать релевантные, и оставался риск пропустить другие подходящие.

Тогда решили проверить гипотезу, насколько нейросеть потенциально может нивелировать человеческий фактор и позволит организовать процесс тегирования в полуавтоматическом режиме. Редактору можно будет не отвлекаться на поиск тегов, а только утверждать предложенные ИИ. В этом случае рутины станет меньше, и у сотрудников редакции появится время на по-настоящему важные задачи

Как проверяли гипотезу и проводили эксперименты

1. Выбор ИИ-модели

При выборе ИИ-модели одним из главных критериев было понимание русского языка. В РБК редакторы работают со сложно структурированными материалами на разные тематики, и чем лучше модель будет «понимать» русский язык, тем выше качество ее работы.

За основу выбрали SOTA Open Source решение — архитектура T5. Это разработка Google, которую использовали для обучения модели FRED. Она обучалась на русском языке 35 дней на 160 графических процессорах V100 и 5 дней на 80 процессорах A100. Во время реализации проекта тестировали обе модели, и впоследствии перешли на оригинальную архитектуру T5 — большую мультиязычную модель MT5 от Google с более современным токенайзером.

2. Эксперименты с обучением

Начали с того, что разработали отдельный сервис для подготовки «чистого», более компактного списка на основе тех самых десятков тысяч тегов. В дальнейшем этот white list пригодится для создания датасета («корректные» теги + размеченные ими материалы), на основе которого будет обучаться модель.

Первичный цикл обучения проводился на локальной видеокарте на мощностях РБК и составил 8 дней непрерывной работы. Нейросеть мгновенно подбирала теги, а процент ошибок был незначительный. Было ясно, что дообучение ИИ потребует дополнительных вычислительных ресурсов. Поэтому процесс решили перенести в облако с более мощным железом.

Следующим шагом необходимо было подобрать оптимальную конфигурацию инфраструктуры, которая с одной стороны позволит сократить время на дообучение модели, а с другой — не будет избыточной по соотношению затраты/производительность.

3. Тестирование оборудования

Специальной для этой задачи инженеры Рег.ру подготовили тестовый стенд с видеокартой A5000 на базе облачных серверов с GPU. Производительность стенда сравнили с двумя другими конфигурациями: T4 и RTX 3090.

В таблице ниже представлены результаты тестирования.

- Значения it/s — это количество обрабатываемых материалов за один шаг в рамках эпохи, этот показатель отражает скорость обучения.
- По вертикали указано количество батчей в процессе параллельного выполнения задач обучения.
- Out — означает ошибку out of memory, то есть объема вычислительной памяти не хватило для размещения задачи.
- Подпись (gen) в модели — что обучение было на задачу text2text generation, а (pred) — sequence classification.

По итогу проведенных тестов видеокарта A5000 в облаке Рег.ру показала лучшие результаты. Она оснащена графической памятью объемом 24 ГБ и позволяет обрабатывать бóльшие объемы данных без замедления работы. К тому же видеокарта поддерживает APICUDA и DirectML и совместима с большинством нейросетевых библиотек и приложений.

4. Проверка гипотезы

Тестирование автоматической разметки материалов тегами проводили на разных проектах РБК, включая «РБК Тренды», «РБК Отрасли» и «РБК Life». Суммарно это более 25 тысяч материалов.

Для перепроверки использовали уже существующие материалы, размеченные редакторами, — публикации загружали в нейросеть и сравнивали результаты машины и человека. На основе полученных результатов делали выводы о том, насколько релевантные теги подобрала модель.

Проведенный тест подтвердил работоспособность подхода. После успешного первичного тестирования и до внедрения в админку для публикации статей, ИИ-модель интегрировали в редакционные процессы «РБК Трендов» с помощью промежуточного решения в виде телеграм-бота.

Результаты проекта

Процесс обучения ИИ-модели в облаке удалось уместить всего в 14–15 часов за один цикл обучения. Точность подбора тегов составила 99%, а скорость тегирования одного материала экстремального сократилась до 0,03 секунды. При этом различалось число тегов. Например, редактор ставил 3 тега, а ИИ-модель выдавала 7 тегов, среди которых в 99% случаев были исходные 3.

В таблице ниже представлены временные значения запуска ИИ-модели в сравнении в разных конфигурациях.

- Init — время загрузки модели в RAM / VRAM.
- Pred — время выполнения задачи до получения результатов.

Значение времени является усредненным для 1 000 запущенных задач тегирования. Для оценки использовались два материала «РБК Трендов»: первый длиной ~200 токенов (по GPT4 токенайзеру), второй ~2000 токенов.

В качестве наглядного примера рассмотрим демо в «РБК Трендах». Редактор указал 4 тега для материала, мы попросили ИИ-модель предсказать 6 тегов по тексту. Ниже представлены результаты двух моделей, обученных на 18 и 30 эпох, — для сравнения. В результате все 4 редакционных тега вторая модель предсказала верно.

Что касается возможных ошибок в работе нейросети, то безусловно такую вероятность нельзя не учитывать, поэтому мы говорим не про автоматическое, а про автоматизированное решение. То есть нейросеть подбирает теги, а редактор визирует и либо принимает, либо правит их. Это позволяет исключить возможные ошибки нейросети, но при этом сохранить все преимущества, которые дал проект.

В ходе эксперимента мы поняли и перспективы дальнейшего развития. Нейросеть может позволить давать более персонифицированные рекомендации, улучшить пользовательский опыт и повысить вовлеченность аудитории. Но это нас ждет еще впереди.