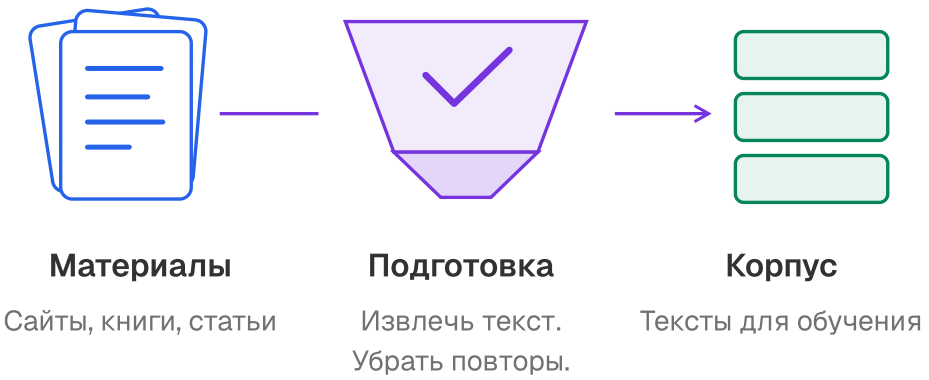
Обучение языковых моделей

Готовая модель угадывает следующее слово. Откуда у неё взялись миллионы настроенных чисел и кто сказал ей правильный ответ?

До сих пор мы пользовались готовой языковой моделью. Откуда берутся её параметры? Их подбирают на текстах: модель пробует продолжить начало, сравнивает прогноз с тем, что действительно написано, и учится на ошибке.

1. От текстов к корпусу

Для обучения нужна большая коллекция текстов — **корпус**. В него могут входить веб-страницы, книги, статьи и обсуждения. Но собрать материалы недостаточно: из них нужно извлечь текст, убрать повторы и отсеять мусор.



Тексты сначала готовят, а затем превращают в примеры для модели.

Например, Common Crawl хранит результаты обходов интернета, а FineWeb — уже извлечённые и отфильтрованные тексты. Архив страниц и подготовленный корпус — разные вещи.

Качество важнее одного только размера. Спам, повторы и ошибки тоже можно выучить. Если в корпусе мало текстов на каком-то языке, модель получает меньше примеров этого языка. Отдельная работа — проверка прав на материалы и обработка персональных данных; автоматические фильтры не решают эти вопросы полностью.

> В строке «Снег идёт ночью» каждое слово принято за токен. Вход модели — «Снег идёт». Какую цель берём из этого текста?
>
> - ○ «ночью»: следующий токен исходной строки
> - ○ «идёт»: последнее слово входа
> - ○ «Снег»: первое слово предложения
> - ○ «утром»: другое правдоподобное продолжение

Ранее

Ранее мы разобрали: генерация текста: выбор токенов и температура, обучение нейронных сетей: функция потерь. Здесь используем эти идеи в новой задаче.

← Генерация текста: выбор токенов и температура

← Обучение нейронных сетей: функция потерь

Заметка

Повторы — тоже данные. Тысяча копий одной страницы не даёт тысячу разных взглядов на тему.

2. Один текст — много обучающих пар

В тексте правильные продолжения уже есть. Из «Кот спит на диване» получаем несколько задач: после «Кот» предсказать «спит», после «Кот спит» — «на», после «Кот спит на» — «диване». Человеку не нужно отдельно подписывать ответ для каждой пары. Как из текста получают цели для обучения

Соберите пары из своей фразы

Выберите слово, которое модель должна предсказать.

Кот | спит | на | **диване**

Вход модели

Кот спит на

→

Цель из текста

диване

Один текст — **3** обучающих пар: каждый следующий токен можно сделать целью.

В этом первом опыте считаем каждое слово одним токеном.

Синие слова — вход. Зелёное слово — цель, которую берём из следующей позиции текста.

Интерактивный эксперимент — в электронной версии урока.

Модель выдаёт вероятности следующего токена. Программа обучения смотрит, какую вероятность получил токен из текста: чем она ниже, тем больше **ошибка**, или **функция потерь**. Затем программа рассчитывает, как сдвинуть параметры, чтобы уменьшить ошибку, и обновляет веса.

Получается цикл: **начало текста → прогноз → сравнение с целью → обновление параметров → новый прогноз**. Общие параметры работают во всех примерах, поэтому одно обновление может изменить ответы сразу на разные запросы.

Пример

«Кот спит на полу» тоже звучит правдоподобно. Но если исходная строка заканчивается «на диване», цель именно «диване».

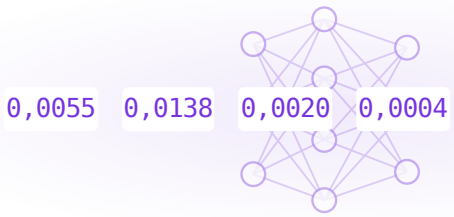
> Вы попросили готовую модель переписать ответ и приложили образец. Стало ли это шагом обучения её параметров?
>
> ○ Нет: параметры языковой модели вообще нельзя изменять
>
> ○ Да, если новый ответ оказался лучше предыдущего
>
> ○ Да: любое исправление ответа автоматически меняет веса
>
> ○ Нет: образец изменил контекст; для обучения нужна отдельная процедура обновления весов

3. Четыре обновления — что изменилось?

Посмотрим на записанный опыт с **ruGPT-3 Small** из презентации. В нём у трёх фрагментов из русскоязычных текстов FineWeb2 скрыты последние токены. В начале опыта самые вероятные продолжения расходятся с текстом. После каждого обновления меняются веса, а вместе с ними — прогнозы и ошибка.

Три текста → общие параметры → три прогноза

Начало текста — Самый вероятный токен



4 наблюдаемых веса

Доброго всем времени — **!** ×

В тексте: ␣суток

Отклонение от курса всего 18 — **␣дней** ×

В тексте: ␣км

…расположена ниже уровня — **␣земли** ×

В тексте: ␣моря

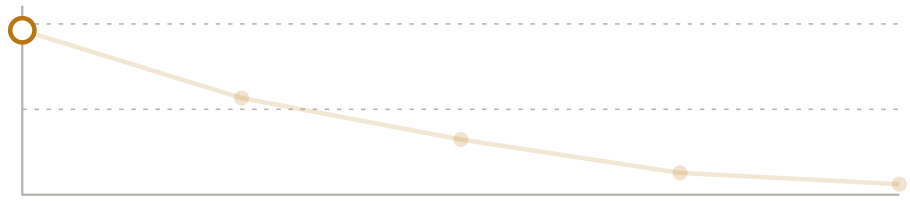
Вероятность цели **«␣суток»**

0,3 % из всех токенов словаря

0 / 3

совпадений с текстом

Средняя ошибка на трёх примерах **7,70**



Число обновлений параметров

Сейчас все три прогноза расходятся с текстом. Посмотрите, что изменит первое обновление весов.

▼ **О записи этого опыта**

Это измерения ruGPT-3 Small на трёх фрагментах FineWeb2, 11.09.2026. Кнопки показывают записанные обновления; обучение большой модели в браузере не запускается.

Перед опытом коэффициенты трёх строк выходного слоя уменьшили вдвое, чтобы сделать ошибки видимыми. Дальше модель дообучали на этих же примерах. Показаны четыре настоящих веса из множества параметров.

Точки графика — измерения после каждого обновления. ␣ обозначает пробел в токене. Оценка на новых текстах в этой записи отсутствует.

Пройдите четыре записанных обновления. Выберите строку, чтобы увидеть, как менялась вероятность её целевого токена.

Интерактивный эксперимент — в электронной версии урока.

Это **дообучение** готовой модели, а не создание её с нуля. Чтобы ошибки были хорошо видны, перед опытом мы специально ослабили часть выходных параметров. За четыре обновления средняя ошибка на этих трёх примерах упала с 7,70 до 0,49. Подробности подготовки можно раскрыть под графиком.

Обратите внимание на « ␣ км»: после четвёртого обновления вероятность этого токена — около 24,2 %. Он всё равно стал первым, потому что у каждого другого токена вероятность ниже. Чтобы прогноз совпал с текстом, не обязательно добиваться 100 %.

В настоящем обучении примеров и шагов гораздо больше. Ошибка на отдельных порциях данных может колебаться. И совпадение с текстом ещё не делает продолжение истинным: в корпус попадают и неверные утверждения.

4. Научилась или запомнила?

Модель стала лучше на трёх знакомых примерах. Но сможет ли она продолжить **новые** тексты? Этот опыт ответа не даёт: мы проверяли те же строки, на которых меняли параметры.

Для честной проверки часть текстов заранее откладывают и не используют в обучении. После настройки модель проверяют на них. Нужно также убрать повторы между двумя наборами: знакомая строка в проверке создаст впечатление, будто модель справилась с незнакомой задачей.

> После дообучения три знакомых продолжения стали правильными. Как проверить перенос на новые примеры?
>
> - ○ Повторить эти же строки в другом порядке
> - ○ Сравнить модель до и после дообучения на отложенных похожих текстах
> - ○ Выбрать самый убедительно написанный ответ из обучающих примеров
> - ○ Поднять вероятность знакомых целей до 100 %

Первоначальную настройку на большом корпусе называют **предобучением**. Затем модель можно дополнительно обучать, например, на инструкциях и ответах. А во время обычного разговора

новые сообщения попадают в контекст: сами веса от этого автоматически не меняются.

От продолжения текста к диалогу

Предобучение учит модель продолжать тексты. Отдельное дообучение на инструкциях и ответах помогает ей выполнять просьбы пользователя.

Следующий токен из текста становится целью: одна строка даёт много обучающих пар.

Программа сравнивает прогноз с целью и обновляет общие параметры, чтобы уменьшить ошибку.

Успех на знакомых строках проверяют на новых текстах, которые не участвовали в обучении.

Новый запрос меняет контекст; отдельное обучение меняет веса.

Домашняя работа

1. Для последовательности из шести токенов берут начала длиной 1, 2, 3 и далее, пока после них остаётся следующий целевой токен. Сколько пар получится?

Источники

- Русскоязычные примеры FineWeb2 (https://huggingface.co/datasets/HuggingFaceFW/fineweb-2/viewer/rus_Cyrl/train)
- Подготовка целей и обучение языковой модели (https://huggingface.co/learn/llm-course/chapter7/6)