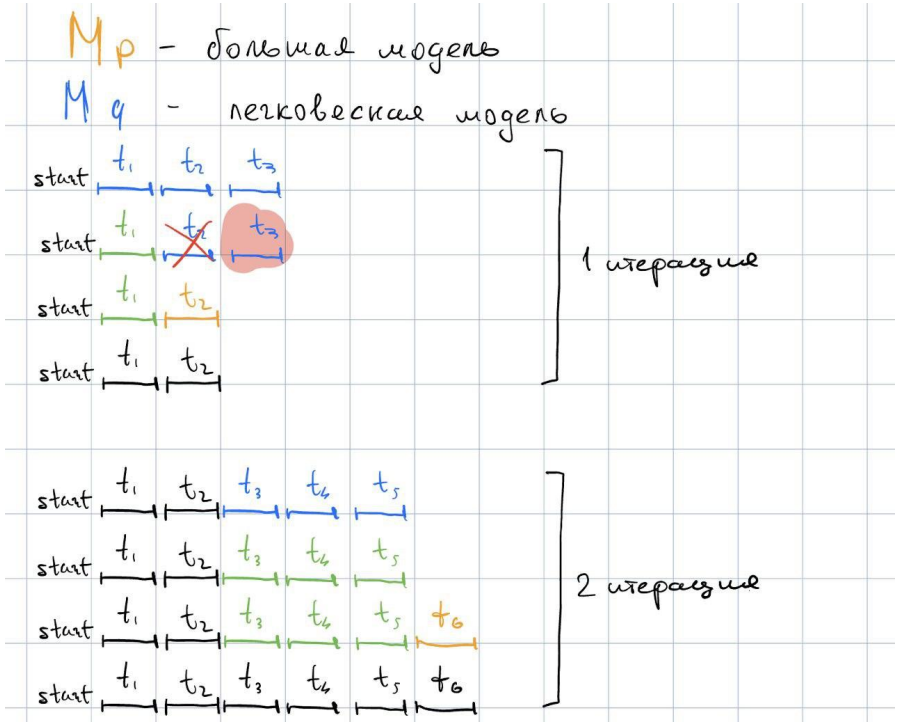
Fast Inference from Transformers via Speculative Decoding

Доклад подготовил Борзыкин Валерий

Введение

- Хотим ускорить инференс авторегрессионных моделей, не меняя вообще ничего (в том числе полностью оставляя выход модели)
- Для этого используем уменьшенную модель, чтобы в основном она предсказывала токены, а большую используем чтобы проверять корректность генерации и поправлять неправильные токены, если надо
- Большая модель параллельно проверяет токены из генерации малой модели
- Ускорение инференса в 2-3 раза для T5-XXL

Алгоритм на примере



Speculative Decoding

- Обозначения:
- M_p - большая целевая модель, инференс которой хотим ускорить, а $p(x_t|x_<t)$ - распределение модели для префикса $x_<t$
- M_q - более эффективная аппроксимирующая модель решающая ту же самую задачу, а ее распределение - $q(x_t|x_<t)$

Speculative Decoding

- Основная идея:
  - использование легковесной модели для генерации последовательности токенов заданной длины (γ новых токенов)
  - параллельный запуск большой модели, которая оценивает все префиксы сгенерированной последовательности, а также соответствующие вероятности из M_q и принимает те, которые могут вести к первоначальному распределению
  - сэмплирование большой моделью дополнительного токена который либо заменяет первый неправильный токен, либо вставляет дополнительный токен в конец, если приняты все токены, предложенные M_q

Speculative Decoding

- Есть множество методов сэмплирования: argmax, top-k, nucleus, задание температуры и др.
- Они все могут быть приведены к standard sampling из скорректированного вероятностного распределения
- Например, в случае argmax-а можно занулить все элементы, которые не являются максимумом, и провести нормализацию
- Тогда будем считать, что p(x) и q(x) это распределения из M_p и M_q соответственно, с поправкой на метод сэмплирования.

Speculative Decoding

- Новый метод сэмплирования - speculative sampling
  - Хотим сэмплировать $x \sim p(x)$
  - Вместо этого для начала сгенерируем $x \sim q(x)$
  - Оставим сгенерированный токен, если $q(x) \leq p(x)$
  - Если же $q(x) > p(x)$, то отклоним токен с вероятностью $1 - \frac{p(x)}{q(x)}$, и сгенерируем исправленный токен из распределения $p'(x) = norm(max(0, p(x) - q(x)))$
  - Выясняется, что на самом деле $x \sim p(x)$ для любых p(x) и q(x)

Speculative Decoding

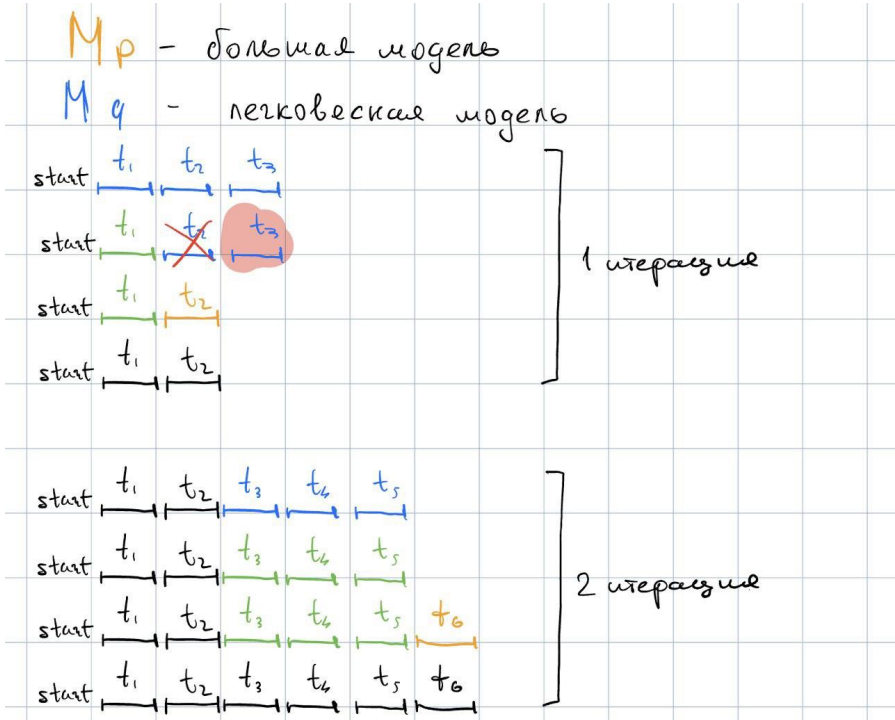
Algorithm 1 SpeculativeDecodingStep

Inputs: $M_p, M_q, prefix$.
▷ Sample γ guesses $x_{1,\ldots,\gamma}$ from M_q autoregressively.
for $i = 1$ **to** γ **do**
  $q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$
  $x_i \sim q_i(x)$
end for
▷ Run M_p in parallel.
$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$
  $M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$
▷ Determine the number of accepted guesses n.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \leq i \leq \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from M_p if needed.
$p'(x) \leftarrow p_{n+1}(x)$
if $n < \gamma$ **then**
  $p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
end if
▷ Return one token from M_p, and n tokens from M_q.
$t \sim p'(x)$
return $prefix + [x_1, \ldots, x_n, t]$

Speculative Decoding

Algorithm 1 SpeculativeDecodingStep

Inputs: $M_p, M_q, prefix$.
▷ Sample γ guesses $x_{1,\ldots,\gamma}$ from M_q autoregressively.
for $i = 1$ **to** γ **do**
  $q_i(x) \leftarrow M_q(prefix + [x_1, \ldots, x_{i-1}])$
  $x_i \sim q_i(x)$
end for
▷ Run M_p in parallel.
$p_1(x), \ldots, p_{\gamma+1}(x) \leftarrow$
  $M_p(prefix), \ldots, M_p(prefix + [x_1, \ldots, x_\gamma])$
▷ Determine the number of accepted guesses n.
$r_1 \sim U(0,1), \ldots, r_\gamma \sim U(0,1)$
$n \leftarrow \min(\{i - 1 \mid 1 \le i \le \gamma, r_i > \frac{p_i(x)}{q_i(x)}\} \cup \{\gamma\})$
▷ Adjust the distribution from M_p if needed.
$p'(x) \leftarrow p_{n+1}(x)$
if $n < \gamma$ **then**
  $p'(x) \leftarrow norm(max(0, p_{n+1}(x) - q_{n+1}(x)))$
end if
▷ Return one token from M_p, and n tokens from M_q.
$t \sim p'(x)$
return $prefix + [x_1, \ldots, x_n, t]$



Анализ метода

- **Definition 3.1.** The *acceptance rate* $\beta_{x_{<t}}$, given a prefix $x_{<t}$, is the probability of accepting $x_t \sim q(x_t|x_{<t})$ by speculative sampling, as per Section 2.3[2].
- Введем обозначение: $\alpha = E(\beta)$
- $$E(\#\ generated\ tokens) = \frac{1-\alpha^{\gamma+1}}{1-\alpha}$$
- Как считать α :

$$\alpha = 1 - E(D_{LK}(p,q)) = E(\min(p,q))$$

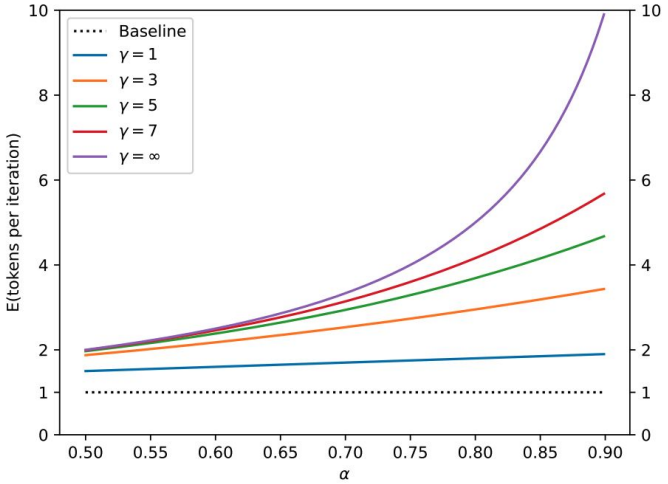


Figure 2. The expected number of tokens generated by Algorithm 1 as a function of α for various values of γ.

Анализ метода

- Алгоритм уменьшает кол-во вызовов M_p на $\frac{1-\alpha^{\gamma+1}}{1-\alpha}$
- Хотим также посчитать, во сколько раз уменьшится общее время выполнения
- **Definition 3.7.** Let c, the *cost coefficient*, be the ratio between the time for a single run of M_q and the time for a single run of M_p.

- **Theorem 3.8.** *The expected improvement factor in total walltime by Algorithm 1 is* $\frac{1-\alpha^{\gamma+1}}{(1-\alpha)(\gamma c+1)}$.

Анализ метода

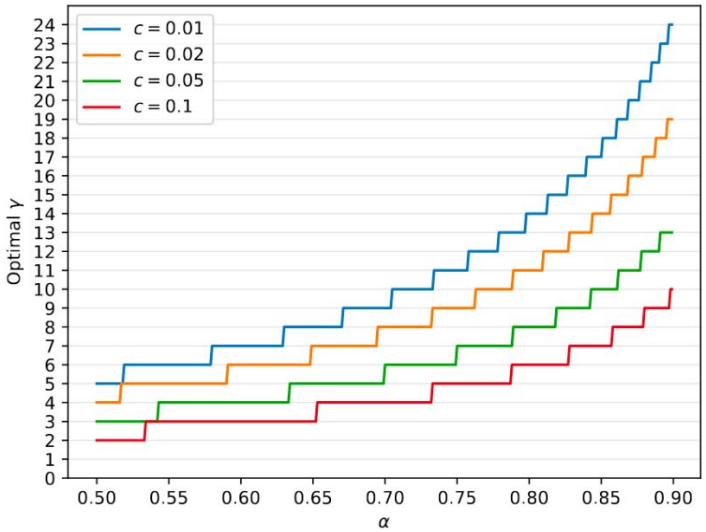


Figure 3. The optimal γ as a function of α for various values of c.

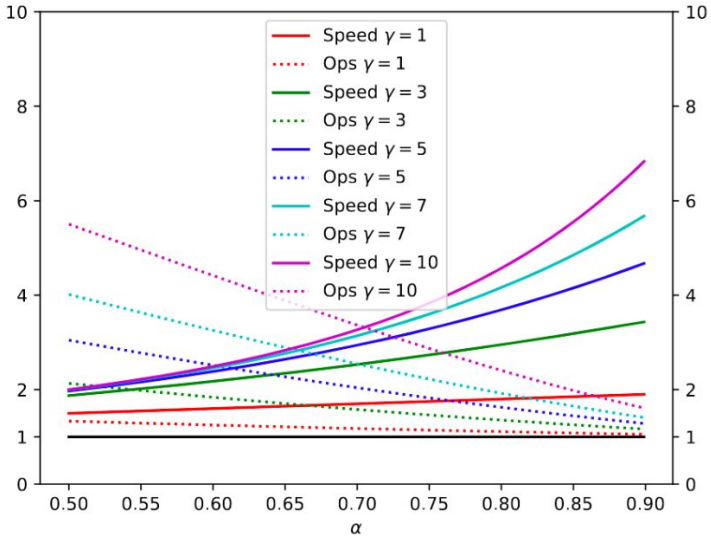


Figure 4. The speedup factor and the increase in number of arithmetic operations as a function of α for various values of γ.

Анализ метода

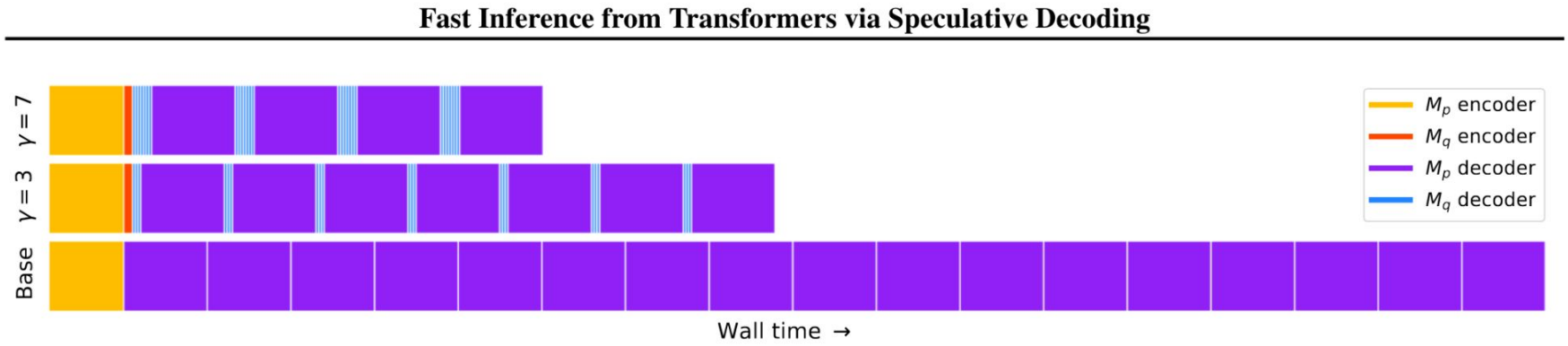


Figure 5. A simplified trace diagram for a full encoder-decoder Transformer stack. The top row shows speculative decoding with $\gamma = 7$ so each of the calls to M_p (the purple blocks) is preceded by 7 calls to M_q (the blue blocks). The yellow block on the left is the call to the encoder for M_p and the orange block is the call to the encoder for M_q. Likewise the middle row shows speculative decoding with $\gamma = 3$, and the bottom row shows standard decoding.

Анализ метода

В качестве M_q можем выбирать абсолютно любую модель

- Наиболее очевидный вариант - трансформер меньшего размера (дает наилучшие результаты)
- Модели со относительной стоимостью одного запуска близкой к 0 (легки в использовании)
  - n-граммы
  - Простые эвристики, например копирование токенов из контекста в случае одинакового префикса
- Неавторегрессионные модели (за одно выполнение генерируют последовательность токенов)
- Случайное сэмплирование токенов (тоже гарантирует ускорение! хоть и очень маленькое)

Эксперименты

Table 2. Empirical results for speeding up inference from a T5-XXL 11B model.

TASK	M_q	TEMP	γ	α	SPEED
ENDE	T5-SMALL ★	0	7	0.75	**3.4X**
ENDE	T5-BASE	0	7	0.8	2.8X
ENDE	T5-LARGE	0	7	0.82	1.7X
ENDE	T5-SMALL ★	1	7	0.62	**2.6X**
ENDE	T5-BASE	1	5	0.68	2.4X
ENDE	T5-LARGE	1	3	0.71	1.4X
CNNDM	T5-SMALL ★	0	5	0.65	**3.1X**
CNNDM	T5-BASE	0	5	0.73	3.0X
CNNDM	T5-LARGE	0	3	0.74	2.2X
CNNDM	T5-SMALL ★	1	5	0.53	**2.3X**
CNNDM	T5-BASE	1	3	0.55	2.2X
CNNDM	T5-LARGE	1	3	0.56	1.7X

Эксперименты

Table 3. Empirical α values for various target models M_p, approximation models M_q, and sampling settings. T=0 and T=1 denote argmax and standard sampling respectively[6].

M_p	M_q	SMPL	α
GPT-LIKE (97M)	UNIGRAM	T=0	0.03
GPT-LIKE (97M)	BIGRAM	T=0	0.05
GPT-LIKE (97M)	GPT-LIKE (6M)	T=0	0.88
GPT-LIKE (97M)	UNIGRAM	T=1	0.03
GPT-LIKE (97M)	BIGRAM	T=1	0.05
GPT-LIKE (97M)	GPT-LIKE (6M)	T=1	0.89
T5-XXL (ENDE)	UNIGRAM	T=0	0.08
T5-XXL (ENDE)	BIGRAM	T=0	0.20
T5-XXL (ENDE)	T5-SMALL	T=0	0.75
T5-XXL (ENDE)	T5-BASE	T=0	0.80
T5-XXL (ENDE)	T5-LARGE	T=0	0.82
T5-XXL (ENDE)	UNIGRAM	T=1	0.07
T5-XXL (ENDE)	BIGRAM	T=1	0.19
T5-XXL (ENDE)	T5-SMALL	T=1	0.62
T5-XXL (ENDE)	T5-BASE	T=1	0.68
T5-XXL (ENDE)	T5-LARGE	T=1	0.71
T5-XXL (CNNDM)	UNIGRAM	T=0	0.13
T5-XXL (CNNDM)	BIGRAM	T=0	0.23
T5-XXL (CNNDM)	T5-SMALL	T=0	0.65
T5-XXL (CNNDM)	T5-BASE	T=0	0.73
T5-XXL (CNNDM)	T5-LARGE	T=0	0.74
T5-XXL (CNNDM)	UNIGRAM	T=1	0.08
T5-XXL (CNNDM)	BIGRAM	T=1	0.16
T5-XXL (CNNDM)	T5-SMALL	T=1	0.53
T5-XXL (CNNDM)	T5-BASE	T=1	0.55
T5-XXL (CNNDM)	T5-LARGE	T=1	0.56
LAMDA (137B)	LAMDA (100M)	T=0	0.61
LAMDA (137B)	LAMDA (2B)	T=0	0.71
LAMDA (137B)	LAMDA (8B)	T=0	0.75
LAMDA (137B)	LAMDA (100M)	T=1	0.57
LAMDA (137B)	LAMDA (2B)	T=1	0.71
LAMDA (137B)	LAMDA (8B)	T=1	0.74

Плюсы и минусы статьи

- Плюсы:
  - Совершенно ничего не надо делать с исходной моделью
  - Простой алгоритм
  - Для аппроксимации можно использовать модель любой архитектуры и размера
- Минусы:
  - Необходимо иметь достаточно ресурсов, чтобы параллельно запускать много больших моделей
  - Эксперименты только на переводе и суммаризации
  - Экспериментировали только с batch size 1 (как заметил Андрей Ишутин)
  - Измерили ускорение только для моделей семейства T5