Large Memory Layers with Product Keys

Дарья Ничвидюк

HSE

12 марта 2020 г.

Outline

Основные моменты

- Определяем функцию $m : \mathbb{R}^d \to \mathbb{R}^m$, она ведет себя как слой в нейронной сети и увеличивает ее capacity.
- Небольшие вычислительные затраты, как на трейне, так и на тесте; масштабирование до очень больших размеров при сохранении точного поиска по пространству ключей.
- Product-key ускоряет процесс за счет значительного сокращения пространства поиска.

Standard key-value memory layer ($m : \mathbb{R}^d \to \mathbb{R}^m$)

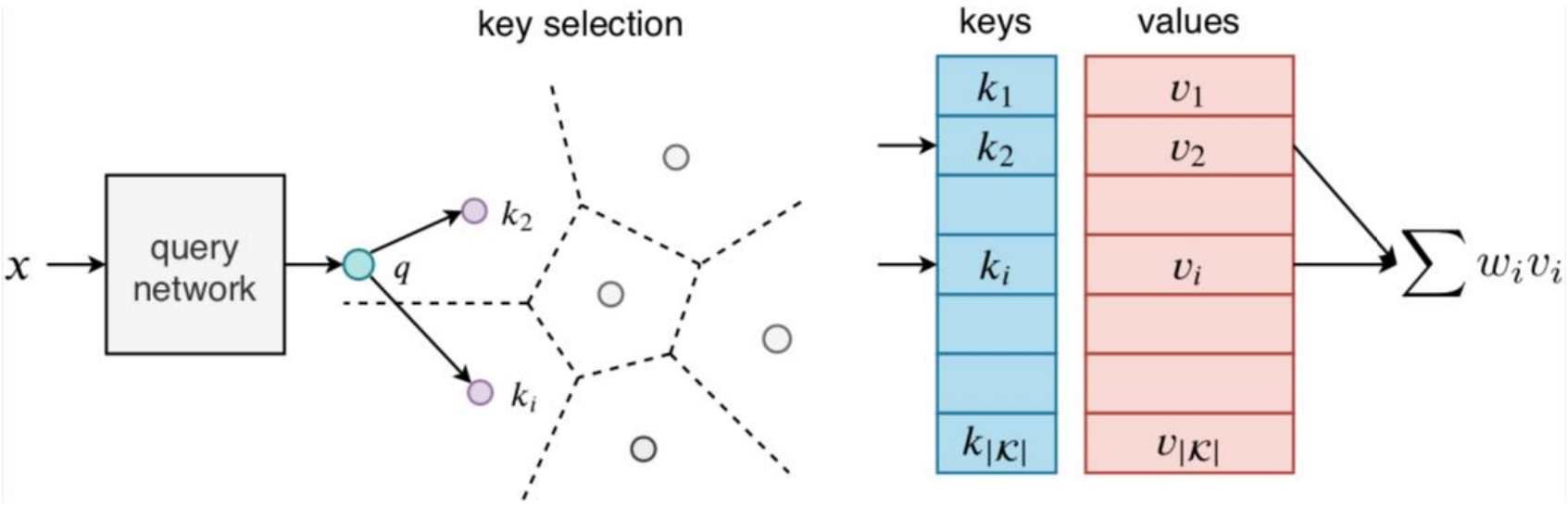


Рис. 1: x – вход нейронного слоя, который преобразуется сетью в запрос q, который сравнивается со всеми ключами из $|K|$. Ответом является взвешенная сумма значений, соответствующих самым похожим ключам.

Standard Key

$$I = T_k\left(q(x)^T k_i\right) \text{ – Найдем } k \text{ ближайших соседей}$$

$$w = \left(q(x^T k_i)_{i \in I}\right) \text{ – Нормализуем топ } k$$

$$m(x) = \sum_{i \in I} w_i v_i \text{ – Агрегируем выбранные значения}$$

Иллюстрация для Product Key

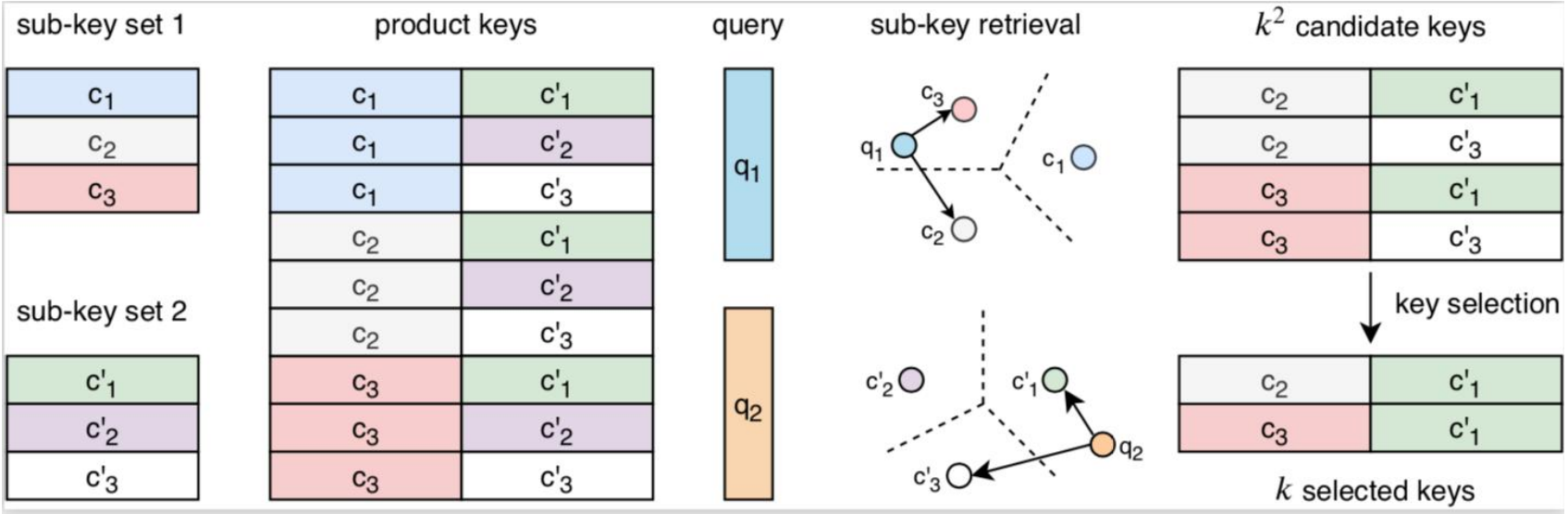


Рис. 2: Поделим запрос q пополам на q_1 и q_2. Найдем для них по k ближайших соседей в каждом множестве подключей. Два множества правых и левых подключей индуцируют все множество ключей $|K|$ внешней памяти. k ближайших соседей запроса q гарантировано попадут в $k \times k$ ключей-кандидатов на ответ.

Product Key

- C_1 и C_2 - множества подключей. Размерность каждого подключа – $d_q/2$.
- Внешнее произведение с конкатенацией C_1 и C_2 это:

$$K = \{(c_1, c_2)|c \in C_1, c_2 \in C_2\}$$

- Найдем k ближайших соседей для q_1 в C_1 как I_{C_1} и q_2 в C_2 как I_{C_2}
- В множество $\{(c_{1,i}, c_{2,j})|c \in I_{C_1}, c_2 \in I_{C_2}\}$ **гарантировано** попадут k самых похожих ключей из K.

Multi-head Attention

- Multi-head Attention делает модель более выразительной. Увеличивается использование ключа и повышает производительность.
- H голов, у каждой есть свой собственный запрос и собственный набор подключей, но с одинаковыми значениями.
- Итоговый ответ –просто сумма

$$m(x) = \sum_{i=1}^{H} m_i(x)$$

- Отличие от стандартного внимания с несколькими головами: ввод (запрос) не разбит на H голов, вместо этого создается H запросов
- На практике: разные головы обращаются к очень разным ключам и очень разным значениям памяти

Формула суммарной сложности вычисления слоя

Для памяти с K ключами (размера $|K|$) и d_q – длины скрытого представления (длина вектора на выходе нейронной сети):

- Стандартный key-value memory layer:
  - Каждое вычисление занимает d_q операций
  - $O(|K| \times d_q)$
- Product-key memory layer:
  - $|C_1| = |C_2| = \sqrt{|K|}$
  - Поиск $k \times k$ возможных ответов в подмножествах: $2 \times O(\sqrt{|K|} \times d_q/2) = O(\sqrt{|K|} \times d_q)$
  - Поиск лучших среди $k \times k$ возможных ответов: $O(k^2 \times d_q)$
- Суммарная сложность

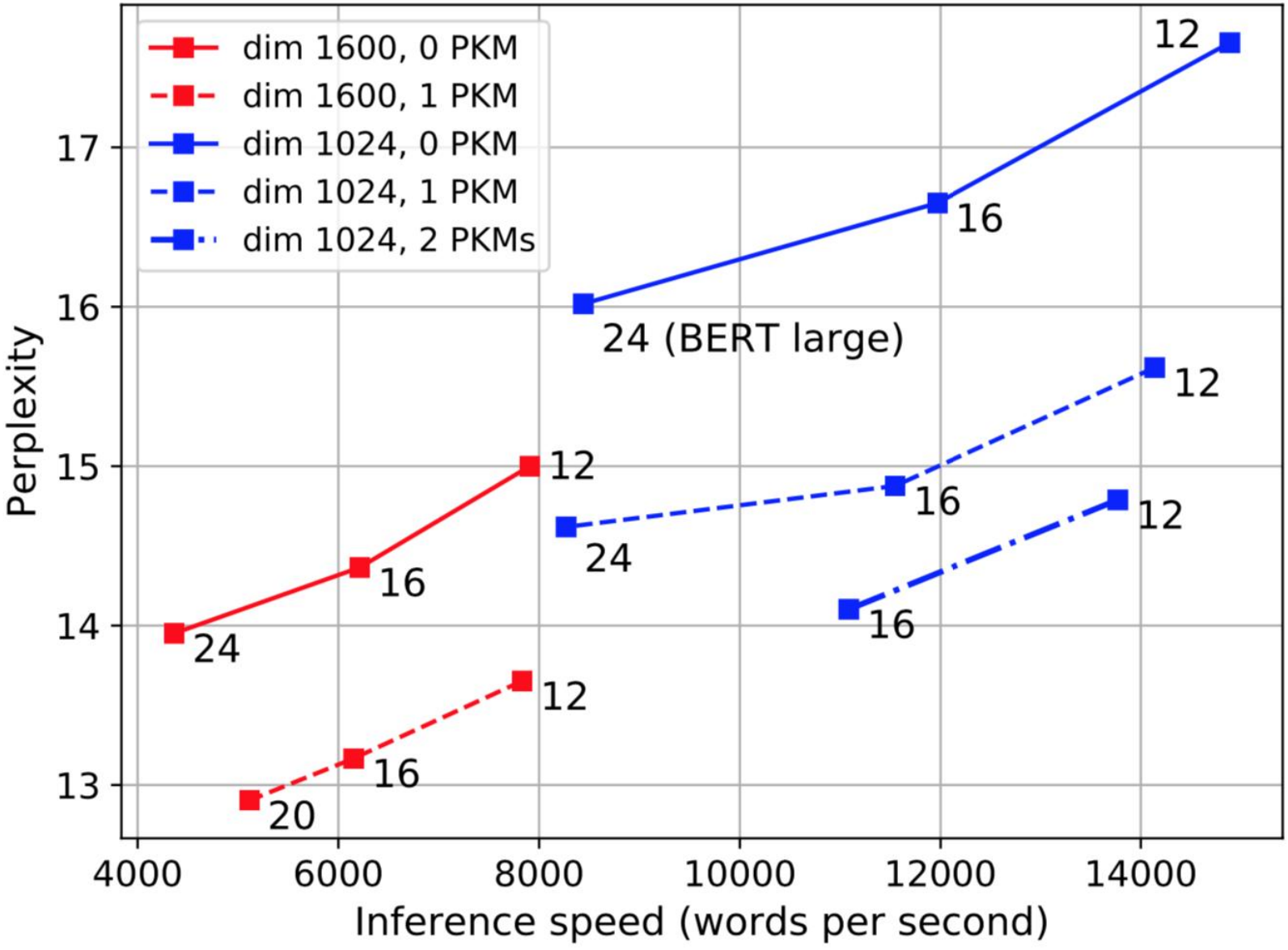
$$O\left((\sqrt{|K|} + k^2) \times d_q\right)$$

Перплексия

Метрика качества модели в этой статье – Перплексия (чем меньше – тем лучше)

$$PP(S) = \mathbf{P}(w_1 w_2 \dots w_N)^{-\frac{1}{N}} = \prod_{i=1}^{N} \frac{1}{p(w_i | w_1 w_2 \dots w_{i-1})}^{-\frac{1}{N}}$$

Результаты


dim 1600, 0 PKM
dim 1600, 1 PKM
dim 1024, 0 PKM
dim 1024, 1 PKM
dim 1024, 2 PKMs
Perplexity
Inference speed (words per second)
17
16
15
14
13
4000
6000
8000
10000
12000
14000
12
16
24 (BERT large)
12
16
24
12
16
24
12
16
20

Ablation Study

- Главным фактором для скорости является количество значений доступной памяти, которое определяется количеством голов памяти h и параметром k, а НЕ размером памяти.
- Batch-normalization запросов помогает.
- Может быть сложно подобрать место для вставки слоя памяти. Худшая позиция находится на первом слое, сразу после входа; вставлять прямо перед выводом softmax тоже не очень. Лучшая позиция для вставки - промежуточный слой.
- Увеличение h и/или k помогает достичь лучшей производительности и лучшего использования памяти, но есть компромисс между скоростью и производительностью. $h = 4$ и $k = 32$ – на практике получается хорошо.
- Лучше, чем стандартных ключей по всех аспектам.

Заключение

12-слойный Трансформер с Product Key Memory превосходит 24-слойный Трансформер и в 2 раза быстрее.

Вопросы

1. В чем идея трюка Product Key?
2. Для каких задач авторы предлагают использовать PKM слои (Product Key Memory Layers)? Приведите примеры.
3. Приведите формулу суммарной сложности вычисления слоя. Поясните все обозначения.