arXiv:2302.13214v2 [cs.LG] 9 May 2023

快速注意力需要有界条目

Josh Alman* Zhao Song†

摘要

在现代机器学习中，内积注意力计算是训练 Transformer、GPT-1、BERT、GPT-2、GPT-3 和 ChatGPT 等大语言模型的一项基础任务。形式上，在这个问题中，输入为三个矩阵 Q、K 和 $V \in [-B, B]^{n \times d}$，目标是构造矩阵 $\mathrm{Att}(Q,K,V) := \mathrm{diag}(A\mathbf{1}_n)^{-1}AV \in \mathbb{R}^{n \times d}$，其中 $A = \exp(QK^\top/d)$ 是“注意力矩阵”，并且 exp 逐元素应用。对于这个问题，直接方法会显式计算 $n \times n$ 注意力矩阵 A，因此即使 $d = n^{o(1)}$ 很小，也需要 $\Omega(n^2)$ 的时间。

在本文中，我们研究了能否通过隐式利用矩阵 A 来实现更快的算法。我们给出两个结果，表明在 $B = \Theta(\sqrt{\log n})$ 处存在一个急剧的转变。

- 如果 $d = O(\log n)$ 且 $B = o(\sqrt{\log n})$，则存在一个 $n^{1+o(1)}$ 时间算法，能够将 $\mathrm{Att}(Q,K,V)$ 近似到误差不超过 $1/\mathrm{poly}(n)$ 的加性误差。
- 如果 $d = O(\log n)$ 且 $B = \Theta(\sqrt{\log n})$，根据细粒度复杂性理论中的强指数时间假设，不可能在真正的次二次时间 $n^{2-\Omega(1)}$ 内将 $\mathrm{Att}(Q,K,V)$ 近似到误差不超过$1/\mathrm{poly}(n)$ 的加性误差。

这为实践中观察到的现象提供了理论解释：输入矩阵的元素较小时，注意力计算的效率会高得多。

*josh@cs.columbia.edu。哥伦比亚大学。
†zsong@adobe.com。Adobe Research。

1 引言

诸如 Transformer 的大语言模型 [VSP+17]、BERT [DCLT18], GPT-3 [BMR+20]、PaLM [CND+22]和 OPT [ZRG+22] 处理自然语言的能力优于较小的模型或传统算法。这意味着它们能够理解和生成更复杂、更细腻的语言，可用于语言翻译、问答和情感分析等各种任务。大语言模型还可以适配多种用途，无需从头开始重新训练。ChatGPT 最近取得的成功，尤其体现了它们的强大能力。ChatGPT 是一款由 OpenAI 开发、基于 GPT-3 [cha22]构建的聊天软件。

大语言模型的关键技术骨干是注意力矩阵 [VSP+17、RNS+18、DCLT18、RWC+19、BMR+20]。注意力矩阵是一个方阵，其行和列对应自然文本中的单词或“词元”，矩阵中的元素则表示这些词元之间的相关性。随后，注意力矩阵用于计算生成输出时序列中每个输入词元的重要性。在注意力机制中，每个输入词元都会被赋予一个权重或分数，以反映其对于当前正在生成的输出的重要性或相关性。这些分数是通过相似度函数比较当前输出状态与输入状态后计算得出的。

更形式化地说，注意力矩阵定义如下。设 $Q \in \mathbb{R}^{n \times d}$ 为查询词元矩阵， $K \in \mathbb{R}^{n \times d}$ 为键词元矩阵。（此处我们关注 $d = n^{o(1)}$的情形，因此 $d \ll n$。）注意力矩阵是一个 $n \times n$ 矩阵 A，其行和列对应序列中的输入词元。矩阵中的每个元素表示特定输入词元（查询词元 Q）与特定输出词元（键词元 K）之间的注意力权重或分数。矩阵的对角元素表示自注意力分数，用于衡量每个词元相对于其自身的重要性。

大语言模型运算（在使用大型 n对长序列建模的情况下）提速的主要瓶颈在于执行注意力矩阵计算所需的时间 [VSP+17、RNS+18、DCLT18、RWC+19、BMR+20、WLK+20、KKL20]。这些计算需要将注意力矩阵 A 与另一个值词元矩阵相乘 $V \in \mathbb{R}^{n \times d}$。

我们将注意力计算正式定义如下。在本文中，我们用 exp 表示矩阵的逐元素指数。

定义 1.1（精确注意力计算 EAttC(n, d)）。给定三个矩阵 $Q, K, V \in \mathbb{R}^{n \times d}$，输出由下式定义的 $n \times d$ 矩阵 Att(Q, K, V)

$$\mathrm{Att}(Q, K, V) := D^{-1} A V$$

其中， $A \in \mathbb{R}^{n \times n}$ 和对角矩阵 $D \in \mathbb{R}^{n \times n}$ 定义如下

$$A := \exp(QK^\top / d), \quad \textit{and} \quad D := \mathrm{diag}(A \mathbf{1}_n).$$

解决此问题的直接算法会计算矩阵 A，然后执行乘法 $D^{-1}AV$，耗时 $n^{2+o(1)}$。由于 A是一个具有 n^2个元素的 $n \times n$矩阵，在显式计算矩阵 A的情况下，几乎不可能在此基础上进一步改进。然而，问题的输入并非 A，而是三个矩阵 Q, K, V，它们各自只有 $n^{1+o(1)}$ 个元素。若某种算法仅隐式地使用 A，而不显式计算其所有元素，就有望以近线性时间运行！

在这篇论文中，我们探讨以这种方式加速注意力计算的可能性。我们要解决的两个主要问题是：

- **Q1.** 何时可以在近线性时间内执行注意力计算 $n^{1+o(1)}$?

- 问题二：何时可以证明注意力计算不可能采用次二次时间算法?

对于大多数大语言模型，只要能够提供合理的精度保证，在整个推理过程中进行近似注意力计算就足够了 [CGRS19、KKL20、WLK+20、DKOD20、KVPF20、CDW+21、CDL+22、LWD+23]。因此，我们在此关注近似注意力计算，它有望比精确计算更快。数学上，我们将近似版的精确注意力计算定义如下。

定义 1.2（近似注意力计算 AAttC(n, d, B, ϵ_a)）。设 $\epsilon_a > 0$ 和 $B > 0$ 为参数。给定三个矩阵 Q、K、$V \in \mathbb{R}^{n \times d}$，并保证 $\|Q\|_\infty \leq B$、$\|K\|_\infty \leq B$ 和 $\|V\|_\infty \leq B$，输出一个矩阵 $T \in \mathbb{R}^{n \times d}$，它近似等于 $D^{-1}AV$，即:

$$\|T - D^{-1}AV\|_\infty \leq \epsilon_a.$$

这里，对于矩阵 $M \in \mathbb{R}^{n \times n}$，我们记 $\|M\|_\infty := \max_{i,\ j} |M_{i,\ j}|$。

同样，这个问题的直接算法需要 $O(n^2 d) \leq n^{2+o(1)}$ 时间，但输入规模仅为 $O(nd) \leq n^{1+o(1)}$。我们的目标是研究在参数 d、B 和 ϵ_a 的意义下，何时能够实现更快的算法。

1.1 我们的结果

我们聚焦于自然的情形：$d = O(\log n)$（即对长序列建模的情形），且 $\epsilon_a = 1/\mathrm{poly}(n)$（误差足够低，因此可以合并整个网络上的注意力计算）。我们的主要结果表明，AAttC 是否存在快速算法，关键取决于 B，即输入矩阵中各项的大小。

我们首先给出一个下界：当 $B \geq \Omega(\sqrt{\log n})$ 时，不可能设计出真正的次二次时间算法。我们的下界利用了强指数时间假设（SETH） [IP01]，这是细粒度复杂性领域中一个广为人知的猜想， [Wil18] 涉及求解 k-SAT 所需的时间。（关于 SETH 的更多讨论，见下文第 4 节。）

定理 1.3 （定理 4.6的非正式版本：下界）。假设 SETH，对于每个 $q > 0$，存在常数 C、C_a、$C_b > 0$，使得：不存在 $O(n^{2-q})$ 时间算法来解决问题AAttC（n，$d = C \log n$，$B = C_b \sqrt{\log n}$，$\epsilon_a = n^{-C_a}$）。

我们的第二个互补性结果是一种新算法，表明当 $B < o(\sqrt{\log n})$ 时，该问题可以非常高效地得到解决，只需近线性时间。

定理 1.4 （定理 3.8的非正式版本：上界）。存在一种算法（算法 1），可以在 AAttC（n，$d = O(\log n)$ $B = o(\sqrt{\log n})$ $\epsilon_a = 1/\mathrm{poly}(n)$）时间内 $n^{1+o(1)}$解决该问题。

我们的定理 1.3 和 1.4 表明，注意力计算问题 AAttC在 $B = \Theta(\sqrt{\log n})$ 处呈现出从近线性时间到平凡二次时间的极其紧密的转变。当$B < o(\sqrt{\log n})$较小时，可以使用我们用于定理 1.4 的算法，在输入规模方面以近线性时间 $n^{1+o(1)}$ 解决该问题。当 $B \geq \Omega(\sqrt{\log n})$ 较大时，我们的定理 1.4 中的算法不再适用；此外，我们定理 1.3 中的下界表明，在假设SETH的情况下，无论采用何种算法技术，都不可能以真正的次二次时间解决该问题。。

在实践中的大语言模型实现中，人们观察到，当假设矩阵元素是有界的，或能够用较少的

比特数进行良好近似时，计算速度会快得多（例如，参见 [ZBIW19，第 2] 节和 [KVPF20，第 3.2 节。1]）。我们的工作可被视为对这一现象给出的理论解释，并有助于说明为何量化等技 - 术 [ZBIW19] 以及低次多项式近似 [KVPF20] 在实践中如此有效。

相关工作。

Zandieh、Han、Daliri 和 Karbasi 最近的一项工作 [ZHDK23] 首次提出了一种具有可证明保证的注意力近似算法。他们的算法采用了局部敏感哈希（LSH）技术 [CKNS20]，正如我们接下来将要讨论的，这与我们用于定理 1.4 的算法截然不同；我们的算法采用多项式方法 [ACSS20，AA22]。

在 $d = o(\log^2 n)$ 的情况下，他们实现的运行时间约为 $O(n^{1.17} \cdot d/\epsilon_r^2)$，其中$\epsilon_r$ 是一个相对 误差参数（与定义 ϵ_a 中的1.2类似，但并不完全相同）。具体而言，他们的算法适用于比我们更大的 d （我们要求 d =$O(\log n)$），但我们实现了近线性时间 $n^{1+o(1)}$（而他们的运行时间下界为 $\Omega(n^{1.17})$），并且我们的算法可以处理任意多项式误差 $\epsilon_a = 1/\mathrm{poly}(n)$（而他们要求 $\epsilon_r \geq 1/n^{o(1)}$ 不能使运行时间增加多项式因子）。

自然会让人想知道，将我们的技术与 [ZHDK23]的方法结合起来，是否还能进一步改进。然而，定理 1.3 的下界表明，定理 1.4 中的算法已经基本达到最优，无法得到实质性改进。

Keles、Wijewardena 和 Hedge 最近的另一项工作 [KWH23] 首次在假设 SETH 成立的情况下证明了注意力计算的下界。他们证明了若干结果，其中包括：当 $d = \omega(\log n)$ 时，AAttC 无法在真正的次二次时间内求解。我们的定理 1.3 将他们的结果推广到 $d = \Theta(\log n)$，并揭示了复杂性如何随条目幅值 B 变化（ [KWH23]并未研究这一问题）。稍后我们将进一步讨论：我们的下界证明和 [KWH23] 都采用了 [BIS17], 的高层次技术；不过，我们对参数 d、B 所做的更细粒度分析需要更复杂的论证，并运用了细粒度复杂性中的其他技术，包括与近似最近邻搜索相关的 [Rub18]以及多项式方法 [AA22]。

1.2 技术概述

我们的总体方法是利用注意力计算与核密度估计（KDE）相关的其他计算问题之间的相似性。近期工作研究了这种关系 [TBY+19、ZHDK23]。具体而言， [ZHDK23] 受启发于 KDE 算法中 LSH 的广泛应用，尝试将 LSH 技术应用于注意力计算 [CS17、BCIS18、CS19、CKNS20]。我们结果背后的主要概念是：除了 LSH 之外，KDE 文献中的其他技术也可以经过修改后应用于这一场景，从而得到紧致的算法和下界。

为了设计定理 1.3中的算法，我们转而借鉴 KDE 领域的另一条研究路线，该路线采用“算法设计中的多项式方法”。设 $M \in \mathbb{R}^{n \times n}$ 是一个矩阵，f： $\mathbb{R} \to \mathbb{R}$是一个函数，并令 $f(M)$ 表示将 f 逐元素应用于M后得到的矩阵。多项式方法是一种寻找 $f(M)$ 的低秩近似的技术。它表明，如果 M 的秩较低，且 f 可以用低次多项式近似，那么矩阵$f(M)$ 就非常接近一个低秩矩阵，并且可以高效地计算其低秩分解。

为了用这种方法求解 AAttC，我们利用一项近期成果，该成果对用多项式近似指数函数所需的次数给出了界 [AA22]，以求得注意力矩阵 A的低秩近似。先前的工作 [ACSS20、ACM+20、AA22] 以类似方式应用这些多项式来求解高斯核密度估计问题；我们的主要观察是，通过适当的重新缩放，可以调整这种方法，使其也适用于 AAttC。

我们的下界定理 1.3 的证明建立在另一系列关于核密度估计问题细粒度复杂性的研究基础上 [BIS17、ACSS20、AA22]。其主要思路是从广为研究的近似最近邻搜索问题 ANN进行细粒度归约。在 ANN中，输入为 n 个维度为 d的向量，以及误差参数 $\epsilon > 0$，目标是找到一对向量，使其距离至多为（$1+\epsilon$）倍于任意一对向量之间的最小距离。求解 ANN 的直接算法需要二次时间；已知在假设 SETH 成立的情况下，不可能在真正的次二次时间内求解 SETH [Rub18]。

为了证明我们的下界，我们证明AAttC 可用于求解ANN。关键思路是，如果 Q 和 K 这两个来自AAttC 的矩阵，是通过拼接ANN 问题中的输入向量构造的，那么最近邻向量就对应于注意力矩阵 A中的最大元素。我们无法立即看出AAttC 可用于检测 A中的大元素，因为输出会被矩阵 D^{-1}重新缩放，但我们证明，通过对输入向量做一些修改，使 A的各行近似平衡，就可以克服这一问题。先前的工作 [BIS17、ACSS20、AA22] 采用了非常相似的方法，为核密度估计问题给出了下界，不过 KDE 不涉及任何重新缩放因子。

路线图。
在第2节中，我们介绍相关符号，以及先前工作中的工具。在第3节中，我们提出并分析注意力算法。在第4节中，我们证明细粒度注意力下界。在第5节中，我们总结本文。

2 预备知识

我们采用标准的实数 RAM 模型，并假设在我们的算法中，对实数执行算术运算可以在常数时间内完成。

我们用 $\mathcal{T}_{\mathrm{mat}}(a,b,c)$ 表示将一个 $a \times b$ 矩阵与另一个 $b \times c$ 矩阵相乘所需的时间。事实上，我们只会用到直接且实用的界 $\mathcal{T}_{\mathrm{mat}}(a,b,c) \leq O(abc)$。原则上，也可以改用快速的理论矩阵乘法算法来改进此界并加快本文算法的运行速度（但会降低其实用性）。不过，鉴于我们的参数设置[1]，，我们将看到，更快的矩阵乘法只能改进运行时间中的低阶项。

对于任意正整数，我们用 $[n]$ 表示集合 $\{1,2,\cdots,n\}$。
对于矩阵 M，我们用 $\|M\|_\infty$表示其 ℓ_∞范数，即 $\|M\|_\infty := \max_{i,j} |M_{i,j}|$。对于矩阵 M，我们用 $M^\top$ 表示其转置。
我们用 $\mathbf{1}_n$ 表示一个长度为n 的向量，其各元素均为 1。我们用 $\mathbf{0}_n$ 表示一个长度为n的向量，其各元素均为 0。
对于任意矩阵 $A \in \mathbb{R}^{n \times n}$，我们用 $\exp(A) \in \mathbb{R}^{n \times n}$ 表示如下矩阵：其中 $\exp(A)_{i,j} = \exp(A_{i,j})$。换句话说，本文中的所有 exp() 运算符都逐元素地应用于矩阵。特别地，本文不会使用矩阵指数。

对于向量 $x \in \mathbb{R}^n$，我们用 $\|x\|_0$ 表示其非零项的数量，用 $\|x\|_1$ 表示其 ℓ_1 范数，即 $\|x\|_1 := \sum_{i=1}^n |x_i|$，并用 $\|x\|_2$ 表示其 ℓ_2 范数，即 $\|x\|_2 := (\sum_{i=1}^n |x_i|^2)^{1/2}$。对于向量 x，我们用 $x^\top$ 表示其转置。

[1]我们将使用 $\mathcal{T}_{\mathrm{mat}}(n, n^{o(1)}, n^{o(1)})$，该问题可以在时间 $n^{1+o(1)}$内直接求解，而且由于其输入规模为 $n^{1+o(1)}$，无法更快地求解。

2.1 多项式逼近的加性误差

我们的注意力计算算法将关键地用到指数函数的多项式逼近。具体而言，我们采用了先前工作中的以下紧致构造 [AA22]。

引理 2.1（[AA22]）。设 $B > 1$，且令 $\epsilon \in (0, 0.1)$。存在一个多项式 $P\colon \mathbb{R} \to \mathbb{R}$，其次数为$g := \Theta\left(\max\left\{\log(1/\epsilon)\log(\log(1/\epsilon)/B),\ B\right\}\right)$，使得对所有 $x \in [0, B]$，均有

$$|P(x) - \exp(x)| < \epsilon.$$

此外，P 可以高效计算：其系数为有理数，整数分子和分母的位数均为 $\mathrm{poly}(g)$，并且可在 $\mathrm{poly}(g)$ 时间内计算。

2.2 从加性误差到相对误差

请注意，在我们的设定下，可以利用引理 2.1 得到相对误差近似：

推论 2.2。设 $B > 1$，且令 $\epsilon \in (0, 0.1)$。存在一个多项式 $P\colon \mathbb{R} \to \mathbb{R}$，其次数为$g := \Theta(\max\{\log(1/\epsilon)\log(\log(1/\epsilon)/B),\ B\})$，使得对所有 $x \in [-B, B]$，均有

$$|P(x) - \exp(x)| < \epsilon \cdot \exp(x).$$

证明。根据引理 2.1，存在一个多项式 $Q\colon \mathbb{R} \to \mathbb{R}$，其次数为 $g = \Theta(\{\log(1/\epsilon)\log(\log(1/\epsilon)/B),\ B\})$，使得对所有 $y \in [0, 2B]$，都有 $|Q(y) - \exp(y)| \le \epsilon$。我们所需的多项式是将$P(x)$重新缩放为：$= Q(x + B)/\exp(B)$。事实上，对于任意 $x \in [-B, B]$，都有 $\exp(x) \ge \exp(-B)$，因此

$$\begin{aligned} |P(x) - \exp(x)| &= |Q(x + B)/\exp(B) - \exp(x)| \\ &= |Q(x + B) - \exp(x + B)|/\exp(B) \\ &\le \epsilon/\exp(B) \\ &\le \epsilon \cdot \exp(x), \end{aligned}$$

如所需。 □

3 注意力算法

本节将展示如何利用指数函数的多项式逼近来近似执行注意力计算。在第 3.1 节中，我们定义将要使用的低秩矩阵逼近类型。在第 3.2 节中，我们展示如何利用多项式逼近得到这类低秩矩阵逼近。在第 3.3 节中，我们界定矩阵 $QK^\top \in \mathbb{R}^{n \times n}$（在将其转换为注意力矩阵之前）的各项，以确认我们的多项式逼近适用。在第 3.4 节中，我们阐述用于近似注意力矩阵的主要技术。在第 3.5 节中，我们展示如何控制从 A 到缩放矩阵D的误差传播。在第 3.6 节中，我们进一步说明如何控制从 D 和 A 到最终注意力矩阵的误差传播。最后，在第 3.7 节中，我们总结通用算法，并在第 3.8 节中恰当选择参数，以实现近线性时间。

3.1 矩阵低秩近似

定义 3.1。设 $r \geq 1$ 表示一个正整数。设 $\epsilon \in (0, 0.1)$ 表示一个精度参数。~给定一个矩阵 $A \in \mathbb{R}_{\geq 0}^{n \times n}$，我们称 $A \in \mathbb{R}_{\geq 0}^{n \times n}$ 是 A 的 (ϵ, r)-近似，如果

- $\widetilde{A} = U_1 \cdot U_2^\top$对于某些矩阵 U_1、$U_2 \in \mathbb{R}^{n \times r}$（即，$\widetilde{A}$的秩至多为 r），并且
- $|\widetilde{A}_{i,\ j} - A_{i,\ j}| \leq \epsilon \cdot A_{i,\ j}$对于所有（$i$，$j$）$\in [n]^2$。

3.2 从低次多项式到低秩矩阵

引理 3.2。设 $M = XY^\top \in \mathbb{R}^{n \times n}$表示一个满足 X、$Y \in \mathbb{R}^{n \times d}$的矩阵。设 P（x）表示一个 g 次多项式，并定义 $r = \binom{2(g+d)}{2g}$。

存在一种算法，其运行时间为 O（nrg），并且以矩阵 X、Y 作为输入，构造矩阵 U_1、$U_2 \in \mathbb{R}^{n \times r}$，使得 P（M）$= U_1 U_2^\top$。（这里，P（M）表示将 P 逐元素应用于 M。）

证明。令 $P(x)$ 表示次数为-g 的多项式。将其按其系数展开为

$$P(x) = \sum_{i=0}^{d} c_i \cdot x^i.$$

考虑函数 K：$\mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$，其定义为：对于 u、$v \in \mathbb{R}^d$，

$$\mathsf{K}(u, v) := P(\langle u, v \rangle).$$

K 是关于向量 u、v 的 $2d$ 个条目 u_1、$\cdots u_d$、v_1、$\cdots$、v_d 的次数为-$2g$的多项式。定义其变量的集合V，

$$V := \{u_1, \cdots, u_d, v_1, \cdots, v_d\}.$$

令 $\mathcal{F}$ 表示函数集合

$$\mathcal{F} := \left\{ f : V \to \{0, 1, 2, \cdots, 2g\} \mid \sum_{v \in V} f(v) \leq 2g \right\}.$$

我们可以计算得到

$$|\mathcal{F}| = \binom{2d + 2g}{2g}.$$

因此，对于每个 $t \in \mathcal{F}$，都有系数 $c_t \in \mathbb{R}$ 使得

$$\mathsf{K}(u, v) = \sum_{t \in \mathcal{F}} c_t \cdot \prod_{v \in V} v^{t(v)}.$$

定义

$$V_u := \{u_1, \cdots, u_d\}$$

and

$$V_v = V \backslash V_u.$$

我们将 ϕ_u 定义为：对于任意 $t \in \mathcal{F}$，$\mathbb{R}^d \to \mathbb{R}^{|\mathcal{F}|}$ 满足

$$\phi_u(u_1, \cdots, u_d)_t = c_t \cdot \prod_{v_i \in V_u} u_i^{t(u_i)}.$$

类似地，我们定义 ϕ_v： $\mathbb{R}^d \to \mathbb{R}^{|\mathcal{F}|}$，对于任意 $t \in \mathcal{F}$,

$$\phi_v(v_1, \cdots, v_d)_t = \prod_{v_i \in V_v} v_i^{t(u_i)}.$$

因此，我们有

$$\mathsf{K}(u, v) = \langle \phi_u(u), \phi_v(v) \rangle.$$

对于 $i \in [n]$，令 $X_i \in \mathbb{R}^d$ 表示 X 的第 i 行，并令 $Y_i \in \mathbb{R}^d$ 表示 Y 的第 i 行。因此，我们的算法可以构造

- 矩阵 $U_1 \in \mathbb{R}^{n \times |\mathcal{F}|}$，其第 i 行是向量 $\phi_u(x_i)$，其中 $i \in [n]$，并且
- 矩阵 $U_2 \in \mathbb{R}^{n \times |\mathcal{F}|}$，其第 i 行是向量 $\phi_v(y_i)$，其中 $i \in [n]$。

这些矩阵的每个元素都可以通过将至多 g 个变量相乘来构造，因此这些 $n \times r$ 个矩阵可以在时间 $O(nrg)$ 内构造出来，正如所需。 □

3.3 矩阵 $QK^\top$ 具有有界条目

引理 3.3（有界条目）。设 $B \geq 1$ 以及矩阵 Q、$K \in \mathbb{R}^{n \times d}$ 具有 $\|Q\|_\infty \leq B$ 和$\|K\|_\infty \leq B$。则有

$$\|QK^\top / d\|_\infty \leq B^2.$$

证明。对于每个 $(i, j) \in [n] \times [n]$，有

$$\begin{aligned} |(QK^\top)_{i,j}| &= |\sum_{l=1}^d Q_{i,l} K_{j,l}| \\ &\leq d \cdot \|Q\|_\infty \cdot \|K\|_\infty \\ &\leq d \cdot B^2, \end{aligned}$$

如所要求。 □

3.4 关键引理

我们的关键引理表明，尽管注意力矩阵 A 可能具有满秩，但它存在一种易于计算的低秩近似：

引理 3.4. 假设 $Q, K \in \mathbb{R}^{n \times d}$，且有 $\|Q\|_\infty \leq B$ 和 $\|K\|_\infty \leq B$。令 $A := \exp(QK^\top / d) \in \mathbb{R}^{n \times n}$。对于精度参数 $\epsilon \in (0, 1)$，存在一个上界为

$$g = O\Big(\max\Big\{ \frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B^2 \Big\} \Big),$$

以及一个上界为 r 的正整数

$$r \leq \binom{2(g+d)}{2g}$$

使得：存在一个矩阵 $A \in \mathbb{R}^{n \times n}$ 它是一个 (ϵ, r)-近似（定义 3.1），适用于 $A \in \mathbb{R}^{n \times n}$。此外，定义 U_1 和 U_2 的矩阵 A 可以在 $O(n \cdot r)$ 时间内计算出来。

证明。设 $M := QK^\top/d$。根据引理 3.3，我们知道 $\|M\|_\infty \leq B^2$。因此，应用推论 2.2（其条目有界为 B^2），存在一个次数为 g 的多项式 P，使得矩阵 $A = P(M)$ 是 A 的 (ϵ, r)-近似（参见定义 3.1 中 (ϵ, r)-近似的定义）。随后，我们可以使用引理 3.2 计算 U_1, U_2，由此得到界

$$r \leq \binom{2(g+d)}{2g}.$$

证明至此完成。 □

3.5 从 A 到 D

引理 3.5。设 $A \in \mathbb{R}^{n \times n}$ 是任意一个元素全为正数的矩阵，$\epsilon_A \in (0, 0.1)$是任意一个参数。设 $A \in \mathbb{R}^{n \times n}$ 是任意一个满足对所有$(i, j) \in [n] \times [n]$，均有以下关系的矩阵：

$$|\widetilde{A}_{i,j} - A_{i,j}| \leq \epsilon_A \cdot A_{i,j}.$$

定义矩阵 D和$D \in \mathbb{R}^{n \times n}$ 如下：$D = \mathrm{diag}(A\mathbf{1}_n)$ 和 $D = \mathrm{diag}(A\mathbf{1}_n)$。那么，对所有 $i \in [n]$，均有以下关系：

$$|\widetilde{D}_{i,i} - D_{i,i}| \leq \epsilon_A \cdot D_{i,i}.$$

证明。我们计算得：

$$\begin{aligned} |\widetilde{D}_{i,i} - D_{i,i}| &= |\sum_{j=1}^n \widetilde{A}_{i,j} - \sum_{j=1}^n A_{i,j}| \\ &\leq \sum_{j=1}^n |\widetilde{A}_{i,j} - A_{i,j}| \\ &\leq \sum_{j=1}^n \epsilon_A A_{i,j} \\ &= \epsilon_A \cdot D_i. \end{aligned}$$

证明完毕。 □

3.6 从 A 和 D 到 AttentionMatrix

引理 3.6. 设 $\epsilon_A, \epsilon_D \in (0, 0.1)$ 和 $B > 1$ 为参数，且令 $V \in \mathbb{R}^{n \times d}$ 表示一个满足 $\|V\|_\infty \leq B$ 的矩阵。设 $A \in \mathbb{R}^{n \times n}$ 为任意一个所有元素均为正的矩阵，并设 $A \in \mathbb{R}^{n \times n}$ 为一个矩阵，使得对于所有 $(i, j) \in [n] \times [n]$，我们有

$$|\widetilde{A}_{i,j} - A_{i,j}| \leq \epsilon_A \cdot A_{i,j}.$$

~设 $D, D \in \mathbb{R}^{n \times n}$ 为任意对角线上元素均为正的对角矩阵，并且对于所有 $i \in [n]$，我们有

$$|\widetilde{D}_{i,i} - D_{i,i}| \leq \epsilon_D \cdot D_{i,i}.$$

于是，我们有

$$\|\widetilde{D}^{-1}\widetilde{A}V - D^{-1}AV\|_\infty \leq (\epsilon_A + \epsilon_D) \cdot B.$$

证明。我们有

$$\|\widetilde{D}^{-1}\widetilde{A}V - D^{-1}AV\|_\infty \leq \|\widetilde{D}^{-1}\widetilde{A}V - D^{-1}\widetilde{A}V\|_\infty + \|D^{-1}\widetilde{A}V - D^{-1}AV\|_\infty. \tag{1}$$

现在我们分别对这两项进行界定。

首先，对于每个 $(i, j) \in [n] \times [d]$,

$$\begin{aligned}
|(\widetilde{D}^{-1}\widetilde{A}V - D^{-1}\widetilde{A}V)_{i,j}| &= |\sum_{l=1}^{n}(\widetilde{D}_{i,i}^{-1} - D_{i,i}^{-1}) \cdot \widetilde{A}_{i,l} \cdot V_{l,j}| \\
&\leq \sum_{l=1}^{n} |(\widetilde{D}_{i,i}^{-1} - D_{i,i}^{-1}) \cdot \widetilde{A}_{i,l}| \cdot \|V\|_\infty \\
&= \sum_{l=1}^{n} |\frac{D_{i,i} - \widetilde{D}_{i,i}}{D_{i,i}\widetilde{D}_{i,i}} \widetilde{A}_{i,l}| \cdot \|V\|_\infty \\
&\leq \epsilon_D \cdot \sum_{l=1}^{D} |\widetilde{D}_i^{-1}\widetilde{A}_{i,l}| \cdot \|V\|_\infty \\
&= \epsilon_D \cdot |\sum_{l=1}^{D} \widetilde{D}_i^{-1}\widetilde{A}_{i,l}| \cdot \|V\|_\infty \\
&= \epsilon_D \cdot \|V\|_\infty \\
&\leq \epsilon_D \cdot B
\end{aligned} \tag{2}$$

其中，第二步由三角不等式得出，第四步由 $|(D_{i,\ i} - \widetilde{D}_{i,\ i})/D_{i,\ i}| \leq \epsilon_D$ 得出，第五步由 $\widetilde{D}_i^{-1} > 0$ and $\widetilde{A}_{i,\ l} > 0$ 得出，最后一步则由我们关于 V 的假设得出。

其次，对于每个 $(i, j) \in [n] \times [d]$,

$$\begin{aligned}
|(D^{-1}\widetilde{A}V - D^{-1}AV)_{i,j}| &= |\sum_{l=1}^{n} D_{i,i}^{-1}(\widetilde{A}_{i,l} - A_{i,l}) \cdot V_{l,j}| \\
&\leq \sum_{l=1}^{n} |D_{i,i}^{-1}| \cdot |(\widetilde{A}_{i,l} - A_{i,l})| \cdot \|V\|_\infty \\
&= \sum_{l=1}^{n} D_{i,i}^{-1} \cdot |(\widetilde{A}_{i,l} - A_{i,l})| \cdot \|V\|_\infty \\
&\leq \sum_{l=1}^{n} D_{i,i}^{-1} \cdot \epsilon_A A_{i,l} \cdot B \\
&= \epsilon_A \cdot B,
\end{aligned} \tag{3}$$

其中，第二步由三角不等式得出，第三步由 $D_{i}^{-1}{}_{,i} > 0$ 得出，~第四步由 $|A_{i,l} - A_{i,l}| \leq \epsilon_A \cdot A_{i,l}$ 得出，最后一步则由 $D_{i,i}$ 的定义得出。

The 结果可由合并公式 (1) 以及两个不等式（公式 (2) 和公式 (3)）得出 . □

3.7 主要上界

定理 3.7. 对于正整数 n、d 以及实参数 $\epsilon > 0$ 和 $B > 1$，存在正整数 $g = \Theta(\max\{\log(1/\epsilon)/\log(\log(1/\epsilon)/B), B^2\})$ 和 $r = \binom{2(g+d)}{2d}$，使得：存在一个算法（算法 1），其运行时间为 $O(\mathcal{T}_{\mathrm{mat}}(n, r, d) + nrg)$，可用于求解 $\mathsf{AAttC}(n, d, B, \epsilon)$（定义 1.2）。

证明。 每一步的运行时间如算法 1所示；其运行时间由引理3.4得出。其正确性由引理 3.5 和引理 3.6得出。 □

算法 1 我们的算法

过程 $\textsc{PolyAttention}(Q \in \mathbb{R}^{n\times d}, K \in \mathbb{R}^{n\times d}, V \in \mathbb{R}^{n\times d}, n, d, B, \epsilon)$ ▷ 定理1.4
▷ ϵ是精度输出
▷ $\|Q\|_\infty, \|K\|_\infty, \|V\|_\infty \le B$
  $g \leftarrow O(\text{最大值}\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)} B^2\})$
  $r \leftarrow \binom{2(g+d)}{2d}$
  根据引理 3.4构造 $U_1, U_2 \in \mathbb{R}^{n\times r}$ ▷ $O(nrg)$ 时间
    $\widetilde{w} \leftarrow U_1 \cdot (U_2^\top \mathbf{1}_n)$ ▷ $O(nr)$ 时间
  $\widetilde{D}^{-1} = \mathsf{diag}(\widetilde{w}^{-1})$ ▷ $O(n)$ 时间
  计算 $U_2^\top V \in \mathbb{R}^{r\times d}$ ▷耗时 $\mathcal{T}_{\mathrm{mat}}(r, n, d)$
  计算 $U_1 \cdot (U_2^\top V)$ ▷ $\mathcal{T}_{\mathrm{mat}}(n, r, d)$ 时间
  $T \leftarrow \widetilde{D}^{-1} \cdot (U_1 \cdot (U_2^\top V))$ ▷ $O(nd)$ 时间
  返回 T ▷ $T \in \mathbb{R}^{n\times d}$
结束过程

3.8 定理 1.4的证明

定理 3.8 （上界，定理 1.4的形式化表述）。 AAttC（n，$d = O$（对数 n） $B = o$（$\sqrt{}$对数 n） $\epsilon_a = 1/$多项式（n）） 可以在时间 $\mathcal{T}_{\mathrm{mat}}$（$n$，$n^{o(1)}$，$d$）内求解 $= n^{1+o(1)}$。

证明。如果我们选择参数

$$B = o(\sqrt{\log n}), \quad \epsilon = 1/\operatorname{poly}(n), \quad d = O(\log n)$$

在定理3.7中，那么我们可以看到

$$\begin{aligned} g &= O(\max\{\frac{\log(1/\epsilon)}{\log(\log(1/\epsilon)/B)}, B^2\}) \\ &= O(\max\{\frac{\log(n)}{\log(\log(n)/B)}, B^2\}) \\ &= O(\max\{\frac{\log n}{\log\log n}, o(\log n)\}) \\ &= o(\log n), \end{aligned}$$

其中，第二步由 $\epsilon = 1/\mathrm{poly}(n)$ 得出，第三步由 $B = o(\sqrt{\log\ n})$ 得出。由于 $g = o(\log\ n)$，令 $g = (\log n)/f$，其中 $f = \omega(1)$。因此我们有

$$r = \binom{2(d+g)}{2g} \le \left(\frac{e(d+g)}{g}\right)^{2g} \le 2^{O(g\log((\log n)/g))} \le 2^{O(\log n \log(f)/f)} < 2^{o(\log n)} < n^{o(1)}.$$

第二步由关于 $1 \le b \le a$ 的通用界 $\binom{a}{b} \le (ea/b)^b$ 得出，第三步则利用了 $d = O(\log n)$。

由于 d、r、g 都由 $n^{o(1)}$ 界定，因此最终运行时间为 $n^{1+o(1)}$，符合预期。 □

4 困难性

本节将证明注意力计算的细粒度下界。在第 4.1 节中，我们陈述强指数时间假设（SETH），这是我们将采用的主要困难性假设。在第 4.2 节中，我们定义近似最近邻搜索问题，并介绍假设 SETH成立时该问题已知的困难性。最后，在第 4.3 节中，我们给出从近似最近邻搜索到注意力计算的归约，由此得到我们的困难性结果。

4.1 细粒度假设

强指数时间假设（SETH）由 Impagliazzo 和 Paturi [IP01]于 20 多年前提出。它是 P $\neq$ NP 猜想的强化版本。该猜想认为，我们目前最好的SAT 算法大致已达到最优：

假设 4.1（强指数时间假设（SETH））。对于每个 $\epsilon > 0$，都存在一个正整数 $k \ge 3$，使得即使采用随机化算法，也无法在 $O(2^{(1-\epsilon)n})$ 时间内求解含有 n 个变量的 k-SAT 公式。

SETH 是一个广为人知的猜想，已被用于证明各种算法问题的细粒度下界。例如，请参阅综述 [Wil18]。

4.2 最近邻搜索

我们将利用 SETH 与近似最近邻搜索之间一个已知的关系。

定义 4.2（近似汉明最近邻搜索（ANN））。给定参数 $\epsilon > 0$，在针对 n 个维度为d的向量的$(1+\epsilon)$-近似汉明最近邻搜索问题中，输入为两个集合 $A, B \subset \{0,1\}^d$，其中 $|A| = |B| = n$，我们的目标是找到一个$a^* \in A$ 和一个 $b^* \in B$，使其满足 $\|a^* - b^*\|_0 \le (1+\epsilon) \cdot \min_{a \in A,\ b \in B} \|a - b\|_0$。

（这有时称为“双色”ANN 问题，也有研究单色版本；例如，参见 [SM19]。）Rubinstein [Rub18] 表明，对于某些参数，假设ANNSETHSETH：

引理 4.3 ([Rub18])。 假设 SETH，对于每个 $q > 0$，存在 $\epsilon \in (0,1)$ 以及 $C > 0$ 使得$(1+\epsilon)$-近似汉明最近邻搜索在维度 $d = C \log n$ 中需要 $\Omega(n^{2-q})$时间。

注4.4。 我们可以假设 *4.3* 即使在以下特殊情况下也成立：来自A 和 B 的每个输入向量，其一半的条目等于 0，另一半等于 1。事实上，对于任意向量 $a \in \{0,1\}^d$，我们都可以构造一个新向量 $\widetilde{a} \in \{0,1\}^{2d}$，定义为 $\widetilde{a} = \left[a^\top \quad \overline{a}^\top\right]^\top$。这里 $\overline{a} \in \{0,1\}^d$是向量 a的二进制补向量$\overline{a}_i = 1 - a_i$，即对所有 $i \in [d]$都有 $\|\widetilde{a}\|_0 = d$。对于每个 $b \in B$，我们也可以类似地构造一个新向量 $\widetilde{b} \in \{0,1\}^{2d}$。经过这一变换后，对于任意 $a \in A$和 $b \in B$，都有 $\|\widetilde{a} - b\|_0 = 2 \cdot \|a - b\|_0$，因此只需在这些变换后的向量中找到一个近似最近邻即可。

为方便起见，我们定义近似最近邻搜索问题的间隙版本 Gap–ANN(n, d, t, ϵ)。

定义 4.5（间隙近似最近邻搜索（Gap–ANN(n, d, t, ϵ)））。设 n、d 为两个正整数。设 $t > 0$ 为阈值参数。设 ϵ 为精度参数。给定两个点集 $A = \{a_1, \cdots, a_n\} \subset \{0,1\}^d$ 和 $B = \{b_1, \cdots, a_n\} \subset \{0,1\}^d$：对于每个 $i \in [n]$，我们需要区分以下两种情况

- 情况 1。存在一个 $j \in [n]$，使得 $\|a_i - b_j\|_0 < t$。
- 情况 2。对于所有 $j \in [n]$，我们都有 $\|a_i - b_j\|_2^2 \geq (1+\epsilon) \cdot t$。

用于 间隙–ANN$(n,\ d,\ t,\ \epsilon)$ 的算法可以调用 log(nd) 次，通过二分搜索求得 ANN 的答案，因此引理 4.3 对于 间隙–ANN$(n,\ d,\ t,\ \epsilon)$ 同样成立。

4.3 困难性结果

在本节余下部分，我们证明注意力计算的下界：

定理 4.6 (主要结果，定理1.3的正式版本。 假设 SETH，对于每个足够小的 $q > 0$，存在常数 $C > 0$ 和 $C_\alpha > 0$ 以及 $C_\beta > 1$，使得对于参数 AAttC (定义 1.2) $(n, d = C \log n, B = C_\beta \sqrt{\log n}, \epsilon_a = n^{-C_\alpha})$的近似注意力计算 需要 n^{2-q} 时间。

证明。 这可由引理 4.3 (近似最近邻搜索的困难性) 与引理 4.7 (从近似最近邻搜索到近似注意力计算的归约) 结合得出，我们将在下文证明这两个引理。 □

引理 4.7。对于任意常数 $C_\gamma \in (0, 0.1)$：对于每个 $\epsilon > 0$ 和 $C > 0$，存在常数$C_a > 0$和 $C_b > 0$，使得如果参数为 $(2n, d = 2C \log n, B = C_b \sqrt{\log n}, \epsilon_a = n^{-C_a})$ 的 AAttC（定义 1.2）可在 T 时间内求解，那么 Gap–ANN$(n, d = C \log n, t, \epsilon)$（定义 4.5）可在 $O(T + n^{2-C_\gamma})$ 时间内求解。

证明。我们给出一个算法，用于求解 Gap–ANN$(n, d = C \log n, t, \epsilon)$，其运行时间如所述。令 $c > 0$ 为一个稍后选定的参数（它是 C 和 C_γ的函数）。我们的算法将根据 t的取值分两种情况处理。如果 $t < c \log n$，则使用一个运行时间为 $O(n^{2-C_\gamma})$的算法。否则，如果 $t \geq c \log n$，则使用另一个运行时间为 $O(T)$的算法。

情况 1：$t < c \log n$。

令 $a_1, \cdots, a_n, b_1, \cdots, b_n \in \{0,1\}^d$为 Gap–ANN 的输入向量，并令 $t \in [0, d]$ 表示目标距离。回顾一下，$d = C \log n$。

在这种 $t < c \log n$ 的情况下，我们将简单地通过以下方式暴力搜索答案：首先将向量 $b_1, \cdots, b_n$ 存入查找表，然后对每个 $i \in [n]$，遍历所有与 a_i 的汉明距离至多为 t 的向量$b' \in \{0,1\}^d$，并检查 b' 是否在查找表中。这样即可确定是否存在一个与 a_i的距离至多为 t 的 $b \in B$，正如所需。

对于每个 $i \in [n]$，我们需要遍历向量 b' 的 $\binom{d}{t}$ 种选择，因此总运行时间为 $O(n \cdot \binom{d}{t})$。根据二项式系数的标准界，我们知道

$$\begin{aligned} n \cdot \binom{d}{t} &\leq n \cdot \binom{C \log n}{c \log n} \\ &\leq n^{1+f(C,c)} \end{aligned}$$

对于某个函数 f： $\mathbb{R}_{>0} \times \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ 它具有这样的性质：对于任意固定的 $C > 0$，我们都有

$$\lim_{c \to 0} f(C, c) = 0.$$

因此，我们可以选取一个足够小的常数 $c > 0$，它仅取决于 C_γ 和 C，使得$f(C, c) < 1 - C_\gamma$，并且整个暴力搜索所需的时间为 $O(n^{2-C_\gamma})$。

情形 2： $t \geq c\log\ n$。

令 $a_1, \cdots, a_n, b_1, \cdots, b_n \in \{0,1\}^d$ 表示 $\mathsf{Gap\text{-}ANN}(n, d, t, \epsilon)$（定义 4.5），并回顾备注 4.4 中的内容：我们可以假设每个向量中一半的元素为 0，另一半为 1。我们将说明如何利用这个实例构造一个注意力矩阵。

令 $C_0 \geq c$ 满足

$$t := C_0 \log n. \tag{4}$$

~令 $\beta > 0$ 和 $d \geq d$ 表示我们稍后将选取的参数（分别见公式（9）和公式（6）。定义 $\tau > 0$ 如下

$$\tau := \exp(\beta). \tag{5}$$

直观来说，我们选取这些参数的目标是使 τ 成为注意力矩阵各元素的上界，也就是说，我们将有：

$$\tau \geq \max_{i \in [n], j \in [n]} \exp(\beta \langle a_i, b_j \rangle / \widetilde{d}).$$

~我们将针对以下参数，使用一种求解 $\mathsf{AAttC}(\widetilde{n}, d, B, \epsilon_a)$ 问题的算法：

$$\widetilde{n} := 2n, \quad \widetilde{d} := 2d, \tag{6}$$

$$B := C_b \sqrt{\log n}, \quad \text{where} \quad C_b := \sqrt{40C/(C_0\epsilon)}, \tag{7}$$

$$\epsilon_a := n^{-C_a}, \quad \text{where} \quad C_a := 2 + C_b^2(1 + C_0/C). \tag{8}$$

此外，令

$$\beta := B^2. \tag{9}$$

~ ~我们将 $Q \in \mathbb{R}^{\widetilde{n} \times d}$ 和 $K \in \mathbb{R}^{\widetilde{n} \times d}$ 定义为

$$Q := \sqrt{\beta} \cdot \begin{bmatrix} a_1^\top & \mathbf{1}_d^\top \\ a_2^\top & \mathbf{1}_d^\top \\ \vdots & \vdots \\ a_n^\top & \mathbf{1}_d^\top \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \\ \vdots & \vdots \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \end{bmatrix} \quad \text{and} \quad K := \sqrt{\beta} \cdot \begin{bmatrix} b_1^\top & 0 \\ b_2^\top & \\ \vdots & \vdots \\ b_n^\top & 0 \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \\ \vdots & \vdots \\ \mathbf{0}_n^\top & \mathbf{1}_d^\top \end{bmatrix}.$$

由于 Q 和 K 的每个条目不是 $\sqrt{\beta}$ 就是 0，因此可得

$$\begin{aligned}\|Q\|_\infty &\leq \sqrt{\beta} = B \\ \|K\|_\infty &\leq \sqrt{\beta} = B \\ \|QK^\top / \widetilde{d}\|_\infty &\leq \frac{\beta \cdot \widetilde{d}}{\widetilde{d}} = \beta = B^2.\end{aligned}$$

~ ~利用上述对 $Q \in \mathbb{R}^{\widetilde{n} \times d}$ 和 $K \in \mathbb{R}^{\widetilde{n} \times d}$的构造，我们注意到

$$A := \exp(QK^\top / \widetilde{d}) \in \mathbb{R}^{\widetilde{n} \times \widetilde{n}}$$

可表示为

$$A = \begin{bmatrix} \exp(\beta\langle a_1, b_1\rangle/\widetilde{d}) & \exp(\beta\langle a_1, b_2\rangle/\widetilde{d}) & \cdots & \exp(\beta\langle a_1, b_n\rangle/\widetilde{d}) & \tau & \tau & \cdots & \tau \\ \exp(\beta\langle a_2, b_1\rangle/\widetilde{d}) & \exp(\beta\langle a_2, b_2\rangle/\widetilde{d}) & \cdots & \exp(\beta\langle a_2, b_n\rangle/\widetilde{d}) & \tau & \tau & \cdots & \tau \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \exp(\beta\langle a_n, b_1\rangle/\widetilde{d}) & \exp(\beta\langle a_n, b_2\rangle/\widetilde{d}) & \cdots & \exp(\beta\langle a_n, b_n\rangle/\widetilde{d}) & \tau & \tau & \cdots & \tau \\ 0 & 0 & \cdots & 0 & \tau & \tau & \cdots & \tau \\ 0 & 0 & \cdots & 0 & \tau & \tau & \cdots & \tau \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \tau & \tau & \cdots & \tau \end{bmatrix}.$$

（请注意，在我们的算法中，我们不会显式计算 A 的所有项；我们只会通过调用用于注意力问题的算法来使用它。）

对于每个$(i, j) \in [n] \times [n]$，我们知道

$$\begin{aligned} A_{i,j} = &\ \exp(\beta\langle a_i, b_j\rangle/\widetilde{d}) \\ \leq &\ \exp(\beta\|a_i\|_\infty \cdot \|b_j\|_\infty \cdot d/\widetilde{d}) \\ \leq &\ \exp(\beta) \\ = &\ \tau \end{aligned} \tag{10}$$

~ ~其中，第一步根据 $A \in \mathbb{R}^{n \times n}$ 的定义得出，第二步根据 $\langle a_i$ 得出，$b_j\rangle \leq$~$\|a_i\|_\infty \cdot \|b_j\|_\infty d$，第三步根据 $d < d$ （参见式（6））得出，最后一步根据 τ （参见式（5））的定义得出。

另一方面，我们知道，对于每个$(i, j) \in [n] \times [n]$，

$$A_{i,j} \geq 0 \tag{11}$$

因为它是 $QK^\top / \widetilde{d}$ 中某个元素的指数函数。

利用公式 (10) 和公式 (11)，结合我们对 A 的表达式，可得

$$n\tau \leq (A\mathbf{1}_{\widetilde{n}})_i \leq 2n\tau, \quad \forall i \in [\widetilde{n}].$$

由于 $D_{i,\ i} =$$(A\mathbf{1}_{\widetilde{n}})_i$，因此我们知道

$$n\tau \leq D_{i,i} \leq 2n\tau, \quad \forall i \in [\widetilde{n}].$$

~选择定义为以下形式的向量 $v \in \mathbb{R}^n$

$$v = \begin{bmatrix} \mathbf{1}_n \\ \mathbf{0}_n \end{bmatrix}.$$

~我们将 t 定义为

$$\widetilde{t} := \frac{1}{3}\exp(0.25\beta(1-t/d))/(2n\tau). \tag{12}$$

~我们可以如下证明 $t \geq \epsilon_a$：

$$\begin{aligned}
\widetilde{t} &= \frac{1}{6n}\exp(0.25\beta(1-t/d)-\beta) \\
&= \frac{1}{6n}\exp(-0.75\beta - 0.25\beta t/d) \\
&= \frac{1}{6n}\exp(-0.75\beta - 0.25\beta C_0/C) \\
&= \frac{1}{6}\exp(-0.75\beta - 0.25\beta C_0/C - \log n) \\
&= \frac{1}{6}\exp(-0.75C_b^2\log n - 0.25C_b^2(C_0/C)\log n - \log n) \\
&\geq n^{-C_a} \\
&= \epsilon_a,
\end{aligned}$$

其中，第二步由简单的代数运算得到，第三步由 $t = C_0 \log n$ (式（4）) 以及 $d = C\log n$ （引理陈述中的假设）得到，第二步由 β（式（7））的取值方式得到，第六步由 C_a （式（8））的取值方式得到，最后一步由式（8）得到。

~由于 $t \geq \epsilon_a$，如果我们运行一个用于近似注意力计算（定义 1.2）的算法 ~AAttC$(\widetilde{n}, d, B, \epsilon_a)$，其中我们选择 V为一个只有一行非零的矩阵 v，那么我们可以输出一个~向量 $u \in \mathbb{R}^n$，使得对于所有 $i \in [\widetilde{n}]$,

$$|u_i - (D^{-1}Av)_i| < \widetilde{t}.$$

注意，利用备注 4.4，我们有

$$\begin{aligned}
\|a_i\|_2^2/d = 0.5, \quad \forall i \in [n], \\
\|b_j\|_2^2/d = 0.5, \quad \forall j \in [n].
\end{aligned}$$

因此，对于任意 $(i,j) \in [n] \times [n]$,

$$\begin{aligned}
\frac{1}{d}\langle a_i, b_j\rangle &= \frac{1}{2d}(\|a_i\|_2^2 + \|b_j\|_2^2 - \|a_i - b_j\|_2^2) \\
&= \frac{1}{2d}(0.5d + 0.5d - \|a_i - b_j\|_2^2) \\
&= 0.5 - 0.5\|a_i - b_j\|_2^2/d,
\end{aligned}$$

其中，第二步由 $\|a_i\|_2^2 = \|b_j\|_2^2 = d/2$ 得出，最后一步则由简单的代数运算得出。

请回想一下，我们的目标是确定，对于每个 $i \in [n]$，是否存在一个 $j \in [n]$ 使得$\|a_i - b_j\|_2^2 \leq t$，或者对于所有 $j \in [n]$，是否有 $\|a_i - b_j\|_2^2 \geq (1+\epsilon_a)t$。接下来我们将说明，通过观察 u_i 大于还是小于值 $t_0 := 2t$，就能~ ~区分这两种情况。

情形 2a。
如果存在一个 $(i,j) \in [n] \times [n]$ 使得 $\|a_i - b_j\|_2^2 \le t$，那么

$$\begin{aligned}\beta\langle a_i, b_j\rangle/\widetilde{d} &= 0.5 \cdot \beta\langle a_i, b_j\rangle/d \\ &\ge 0.25 \cdot \beta(1 - t/d),\end{aligned}$$

~其中第一步由此得出 $2d = d$ （参见公式（6））。
这意味着

$$\begin{aligned}u_i &\ge \exp(0.25\beta(1 - t/d))/(2n\tau) - \widetilde{t} \\ &= 3\widetilde{t} - \widetilde{t} \\ &= 2\widetilde{t} \\ &= \widetilde{t}_0,\end{aligned}$$

~其中第二步由 t 的定义得出（参见公式（12），最后一步则由~的定义 t_0得出。

情形 2b。
若对所有 $(i,j) \in [n] \times [n]$，都有 $\|a_i - b_j\|_2^2 > t(1+\epsilon)$，则可推出

$$\beta\langle a_i, b_j\rangle/d \le 0.25\beta \cdot (1 - t(1+\epsilon)/d).$$

那么，对所有 $i \in [n]$，

$$\begin{aligned}u_i &< (n \cdot \exp(0.25\beta(1 - (1+\epsilon)t/d)))/(n\tau) + \widetilde{t} \\ &= \exp(0.25\beta(1 - t/d))/(2n\tau) \cdot (n/\exp(0.25\beta\epsilon t/d)) + \widetilde{t} \\ &= 3\widetilde{t} \cdot (n/\exp(0.25\beta\epsilon t/d)) + \widetilde{t} \\ &\le 3\widetilde{t} \cdot \frac{1}{4} + \widetilde{t} \\ &= 2\widetilde{t} \\ &= \widetilde{t}_0,\end{aligned}$$

~其中，第三步由定义得出 t （见式（12）），第四步由下文式（~ ~13）中的计算得出，最后一步则由 $t = t_0/2$得出。
最后，b 根据我们对 β 和 t的选择，可以看出

$$\begin{aligned}\exp(0.25\beta\epsilon t/d) &= \exp((0.25\beta\epsilon C_0 \log n)/d) \\ &= \exp(0.25\beta\epsilon C_0/C) \\ &= \exp(10 \log n) \\ &> 4n,\end{aligned} \tag{13}$$

其中，第一步由 $t = C_0$ 对数 n （式（4））得出，第二步由 $d = C$ 对数 n得出，第三步则由 $\beta = B^2$ （式（9））以及 B 的选择（式（7））得出。 □

5 结论

在这项工作中，我们展示了注意力计算的速度在很大程度上取决于输入矩阵中各项的大小， B。我们的主要思路是利用注意力计算与核密度估计之间的相似性，并展示核密度估计中许多已知技术也可用于这一场景。核密度估计是一个研究非常充分的问题，因此，若能发现还有哪些其他技术也可应用于注意力计算，将令人振奋。

致谢 作者谨此感谢 Beidi Chen 就大语言模型相关问题进行的有益讨论，并感谢 Feyza Duman、Chinmay Hegde 和 Piotr Indyk 对早期稿件提出的宝贵意见。

参考文献

[AA22] Amol Aggarwal 和 Josh Alman。指数函数和高斯核密度估计的最优次数多项式近似。载于第 37 届计算复杂性会议（CCC 2022）。Schloss Dagstuhl-Leibniz-Zentrum f¨ur Informatik，2022年。[ACM+20] Josh Alman、Timothy Chu、Aaron Schild 和 Zhao Song。通过表示论研究实超矩形的度量变换与低秩矩阵。arXiv 预印本 arXiv:2011.11503，2020年。[ACSS20] Josh Alman、Timothy Chu、Aaron Schild 和 Zhao Song。几何图上线性代数的算法与困难性。载于 2020年 IEEE 第 61 届计算机科学基础年会（FOCS），第 541–552 页。IEEE，2020年。[BCIS18] Arturs Backurs、Moses Charikar、Piotr Indyk 和 Paris Siminelakis。平滑核的高效密度计算。载于 2018年 IEEE 第 59 届计算机科学基础年会（FOCS），第 615–626 页。IEEE，2018年。[BIS17] Arturs Backurs、Piotr Indyk 和 Ludwig Schmidt。经验风险最小化的细粒度复杂性：核方法与神经网络。《神经信息处理系统进展》，30，2017年。[BMR+20] Tom Brown、Benjamin Mann、Nick Ryder、Melanie Subbiah、Jared D Kaplan、Prafulla Dhariwal、Arvind Neelakantan、Pranav Shyam、Girish Sastry、Amanda Askell 等。语言模型是少样本学习器。《神经信息处理系统进展》（NeurIPS），33：1877–1901，2020年。[CDL+22] Beidi Chen、Tri Dao、Kaizhao Liang、Jiaming Yang、Zhao Song、Atri Rudra 和 Christopher Re。像素化蝴蝶：简单高效的神经网络模型稀疏训练。载于国际学习表征会议（ICLR），2022年。[CDW+21] Beidi Chen、Tri Dao、Eric Winsor、Zhao Song、Atri Rudra 和 Christopher R´e。Scatterbrain：统一稀疏注意力与低秩注意力。《神经信息处理系统进展》（NeurIPS），34：17413–17426，2021年。[CGRS19] Rewon Child、Scott Gray、Alec Radford 和 Ilya Sutskever。使用稀疏 Transformer模型生成长序列。arXiv 预印本 arXiv:1904.10509，2019年。[cha22]ChatGPT：优化用于对话的语言模型。载于 OpenAI 博客。https://openai.com/blog/chatgpt/，2022年11月。[CKNS20] Moses Charikar、Michael Kapralov、Navid Nouri 和 Paris Siminelakis。通过密度约束近邻搜索进行核密度估计。载于 2020年 IEEE 第 61 届计算机科学基础年会（FOCS），第 172–183 页。IEEE，2020年。

[CND+22]Aakanksha Chowdhery、Sharan Narang、Jacob Devlin、Maarten Bosma、Gaurav Mishra、Adam Roberts、Paul Barham、Hyung Won Chung、Charles Sutton、Sebastian Gehrmann 等。PaLM：借助路径扩展语言建模规模。arXiv 预印本 arXiv:2204.02311，2022年。[CS17] Moses Charikar 和 Paris Siminelakis。高维核密度的基于哈希的估计器。载于 2017年 IEEE 第 58 届计算机科学基础年会（FOCS），第 1032–1043 页。IEEE，2017年。[CS19] Moses Charikar 和 Paris Siminelakis。用于快速成对求和的多分辨率哈希。载于 2019年 IEEE 第 60 届计算机科学基础年会（FOCS），第 769–792 页。IEEE，2019年。[DCLT18] Jacob Devlin、Ming-Wei Chang、Kenton Lee 和 Kristina Toutanova。BERT：用于语言理解的深度双向变换器预训练。arXiv 预印本 arXiv:1810.04805，2018年。[DKOD20] Giannis Daras、Nikita Kitaev、Augustus Odena 和 Alexandros G Dimakis。SMYRF：通过非对称聚类实现高效注意力。《神经信息处理系统进展》（NeurIPS），33：6476–6489，2020年。[IP01] Russell Impagliazzo 和 Ramamohan Paturi。关于 k-SAT 的复杂性。《计算机与系统科学杂志》，62(2)：367–375，2001年。[KKL20] Nikita Kitaev、Lukasz Kaiser 和 Anselm Levskaya。Reformer：高效的 Transformer。arXiv 预印本 arXiv:2001.04451，2020年。[KVPF20] Angelos Katharopoulos、Apoorv Vyas、Nikolaos Pappas 和 François Fleuret。Transformer 就是循环神经网络：具有线性注意力的快速自回归变换器。载于国际机器学习会议（ICLR），第 5156–5165 页。PMLR，2020年。[KWH23] Feyza Duman Keles、Pruthuvi Mahesakya Wijewardena 和 Chinmay Hegde。关于自注意力的计算复杂性。载于国际算法学习理论会议，第 597–619 页。PMLR，2023年。[LWD+23] Zichang Liu、Jue Wang、Tri Dao、Tianyi Zhou、Binhuang Yuan、Zhao Song、Anshumali Shrivastava、Yuandong Tian、Ce Zhang、Christopher Re 和 Beidi Chen。似曾相识：推理时实现高效大语言模型的上下文稀疏性。载于 ICML，2023年。[RNS+18] Alec Radford、Karthik Narasimhan、Tim Salimans、Ilya Sutskever 等。通过生成式预训练改进语言理解。2018年。[Rub18] Aviad Rubinstein。近似最近邻搜索的困难性。载于第 50 届 ACM SIGACT 计算理论年会（STOC），第 1260–1268 页，2018年。[RWC+19] Alec Radford、Jeffrey Wu、Rewon Child、David Luan、Dario Amodei、Ilya Sutskever 等。语言模型是无监督多任务学习器。OpenAI 博客，1(8)：9，2019年。[SM19] Karthik C. S. 和 Pasin Manurangsi。关于欧几里得度量下的最近点对：单色问题与双色问题同样困难。第 10 届理论计算机科学创新会议（ITCS），2019年。

[TBY+19] Yao-Hung Hubert Tsai、Shaojie Bai、Makoto Yamada、Louis-Philippe Morency 和 Ruslan Salakhutdinov。Transformer 解剖：从核的视角统一理解 Transformer 的注意力。载于《2019年自然语言处理实证方法会议暨第九届国际自然语言处理联合会议（EMNLP-IJCNLP）论文集》，第 4344–4353 页，2019年。[VSP+17] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser 和 Illia Polosukhin。注意力即所需。载于《神经信息处理系统进展》（NeurIPS），第 30 卷，2017年。[Wil18] Virginia Vassilevska Williams。关于算法与复杂性中的若干精细问题。载于《国际数学家大会论文集：里约热内卢 2018》，第 3447–3487 页。世界科学出版社，2018年。[WLK+20] Sinong Wang、Belinda Z Li、Madian Khabsa、Han Fang 和 Hao Ma。Linformer：具有线性复杂度的自注意力。arXiv 预印本 arXiv:2006.04768，2020年。[ZBIW19] Ofir Zafrir、Guy Boudoukh、Peter Izsak 和 Moshe Wasserblat。Q8bert：量化的8位 BERT。载于《2019年第五届节能机器学习与认知计算研讨会—NeurIPS 专题》（EMC2-NIPS），第 36–39 页。IEEE，2019年。[ZHDK23] Amir Zandieh、Insu Han、Majid Daliri 和 Amin Karbasi。Kdeformer：通过核密度估计加速 Transformer 模型。arXiv 预印本 arXiv:2302.02451，2023年。[ZRG+22] Susan Zhang、Stephen Roller、Naman Goyal、Mikel Artetxe、Moya Chen、Shuohui Chen、Christopher Dewan、Mona Diab、Xian Li、Xi Victoria Lin 等。OPT：开放式预训练Transformer语言模型。arXiv 预印本 arXiv:2205.01068，2022年。