arXiv:2304.06594v1 [cs.DS] 13 Apr 2023

求解张量低循环秩近似

Yichuan Deng* Yeqi Gao† Zhao Song‡

摘要

大语言模型已在现代生活中无处不在，应用于自然语言处理、语言翻译和语音识别等各种领域。最近，一项突破性工作 [Zhao、Panigrahi、Ge 和 AroraarXiv 2023] 从概率上下文无关文法（PCFG）的角度解释了注意力模型。计算PCFG 中概率的核心计算任务之一，是构造一个特定的张量低秩近似问题，我们可以称之为张量循环秩。给定一个 $n \times n \times n$ 三阶张量 A，我们称A 具有循环秩-k，如果存在三个 $n \times k^2$ 尺寸的矩阵 U、V 和 W，使得每个矩阵中的每个条目

$$A_{a,b,c} = \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} U_{a,i+k(j-1)} \otimes V_{b,j+k(l-1)} \otimes W_{c,l+k(i-1)}$$

对所有 $a \in [n], b \in [n], c \in [n]$。关于张量的经典秩、Tucker 秩和张量列车秩， [Song, Woodruff, Zhong SODA 2019] 中已有充分研究。本文将 [Song, Woodruff, Zhong SODA 2019] 第 186 页中的“旋转与草图”技术加以推广，并为循环秩给出一种输入稀疏度时间算法。

*ycdeng@cs.washington.edu。华盛顿大学。
†a916755226@gmail.com。华盛顿大学。
‡zsong@adobe.com。Adobe 研究院。

1 引言

大语言模型（LLM）是一类旨在处理和理解人类语言的人工智能模型。这些模型采用深度学习技术构建，可以利用海量数据进行训练，从而学习语言中的模式和关系，进而生成文本、回答问题并完成其他与语言相关的任务。一些最知名的大语言模型包括 OPT [ZRG+22]、PaLM[CND+22]、GPT-3 [BMR+20]、Transformer [VSP+17] 和 BERT [DCLT18], , 它们已用于各种应用，例如语言翻译 [Sta20, FBS+21, LGG+20] 和内容生成 [VSP+17, DCLT18, BMR+20, CND+22]。大语言模型正变得日益复杂和强大，预计将在塑造人机交互和自然语言处理的未来方面发挥重要作用 [WDS+19, RPSW19, QSX+20, LYF+23]。

作为大语言模型之一，Transformer [VSP+17] 近年来受到了最多的关注。最近，Zhao、Panigrahi、Ge 和 Arora[ZPGA23] 探讨了一个有趣的问题：Transformer在尝试掩码词预测的同时，是否能够进行真正的解析。

为解决上述问题，他们首先关注概率上下文无关文法（PCFG）[Yng60] 及其句法分析算法。作为一种形式文法，概率上下文无关文法（PCFG）广泛应用于自然语言处理 [WDS+19、QSX+20] 和计算语言学[Mit22, BDS19]。其中，$\mathcal{N}$ 是语言文法中所有非终结符的集合，$\mathcal{P} \subset \mathcal{N}$ 是前终结符集合，$\mathcal{I} \subset \mathcal{N}$ 是有限的内部终结符集合，$[n]$ 是所有可能词语的集合，p 是与文法分析相关的概率；五元组 $\mathcal{G} := (\mathcal{N}, \mathcal{P}, \mathcal{I}, p, n)$对其进行了定义。

在概率上下文无关文法（PCFG）中，语言的每条文法规则都与一个概率相关联，该概率表示在生成句子时使用该规则的可能性。这些概率可以通过期望最大化算法等机器学习技术，从大型文本语料库中学习得到。概率上下文无关文法已广泛应用于自然语言处理的各种任务，包括词性标注 [Chu89、TKMS03、Rat96], 句法分析 [Cha96、KM03、PK07], 以及文本生成 [LK98]。

使用概率上下文无关文法模型发现句子 w 的（无标注）解析树，一种著名方法是标注召回算法，该算法由 [Goo96] 提出。该算法旨在找出使得分总和最大的树 $T = (i, j)$，其中，非终结符 a 下跨度$v_i v_{i+1} \cdots v_j$的得分定义为边缘概率 $\mu(i, j, a)$，计算如下： $\mu(i, j, a) = \max_{a \in \mathcal{N}} \Pr[a \to v_i v_{i+1} \cdots v_j, \text{root} \to w | \mathcal{G}]$。

上述边缘概率可以使用内外算法[MS99，Bak79]计算。最近的一项研究中，[ZPGA23] 尝试从概率上下文无关文法（PCFG）的角度解释 Transformer 架构 [MSM93]。

借助动态规划，内外算法可以确定两类概率项。内部概率定义为

$$p_{\text{in}}(i, j, a) := \Pr[a \to v_i v_{i+1} \dots v_j | \mathcal{G}]$$

而外部概率定义为

$$p_{\text{out}}(i, j, a) = \Pr[\text{root} \to v_1 v_2 \dots v_{i-1} a v_{j+1} \dots v_L | \mathcal{G}]$$

对于所有满足 $1 \le i \le j \le L$ and $a \in \mathcal{N}$ 的有序对 (i, j)。确切地说，概率可以是

按照以下递归形式定义，

$$p_{\text{in}}(i,j,a) = \sum_{b,c} \sum_{k=i}^{j-1} p_{\text{in}}(i,k,b) p_{\text{in}}(k+1,j,c) \Pr[a \to bc]$$

and

$$\begin{aligned} p_{\text{out}}(i,j,a) = & \sum_{b,c} \sum_{k=1}^{i-1} p_{\text{in}}(k,i-1,c) p_{\text{out}}(k,j,b) \Pr[b \to ca] \\ & + \sum_{b,c} \sum_{k=j+1}^{L} p_{\text{in}}(j+1,k,c) p_{\text{out}}(i,k,b) \Pr[b \to ac] \end{aligned}$$

已知，对于所有 a, i, $p_{\text{in}}(i,i,a) = \Pr[a \to v_i]$，以及所有 a, $p_{\text{out}}(1, L, \text{root}) = 1$。我们可以通过以下表达式求得边缘概率

$$\begin{aligned} \mu(i,j,a) = & \Pr[a \Rightarrow v_i v_{i+1} \dots v_j, \text{root} \Rightarrow w | \mathcal{G}] \\ = & \Pr[\text{root} \to v_1 v_2 \dots v_{i-1} a v_{j+1} \dots v_L | \mathcal{G}] \cdot \Pr[a \to v_i v_{i+1} \dots v_j | \mathcal{G}] \\ = & p_{\text{in}}(i,j,a) \cdot p_{\text{out}}(i,j,a). \end{aligned}$$

在本文中，我们将重点分析 $p_{\text{in}}(i,j,a)$，并从张量视角研究 $p_{\text{in}}(i,j,b)$、$p_{\text{in}}(k+1,j,c)$ 和 $\Pr[a \to bc]$。将 b, c, k 视为循环索引，并将 a, i, j 视为由 $p_{\text{in}}(i,j,a)$ 转换而来的张量索引后，我们可以用一种新的形式重构该函数。受 $p_{\text{in}}(i,j,a)$ 计算方式的启发，我们将低秩近似中张量的循环秩定义如下。

定义 1.1（张量循环秩）。考虑 $\overline{U}$、$\overline{V}$ 和 $\overline{W}$，它们是元素取实数值的 n-by-k-by-k 张量，且分别是矩阵 $U \in \mathbb{R}^{n \times k^2}$、$V \in \mathbb{R}^{n \times k^2}$ 和 $W \in \mathbb{R}^{n \times k^2}$ 的张量化。若矩阵 U、V 和 W 属于 $\mathbb{R}^{n \times k^2}$ 并满足以下条件，则称 a 具有循环秩：

$$A_{i,j,l} = \sum_{a=1}^{k} \sum_{b=1}^{k} \sum_{c=1}^{k} U_{i,a+k(b-1)} V_{j,b+k(c-1)} v_{l,c+k(a-1)}$$

或者，我们可以表述为

$$A_{i,j,l} = \sum_{a=1}^{k} \sum_{b=1}^{k} \sum_{c=1}^{k} \overline{U}_{i,a,b} \overline{V}_{j,b,c} \overline{W}_{l,c,a}$$

我们指出，还有其他张量秩，例如经典张量秩、张量列车秩和张量 Tucker 秩。这些秩已由 [SWZ19c], 研究过，但张量循环秩尚未得到研究。我们在第 4.2 节对其他张量秩给出了更详细的定义。

1.1 我们的结果

我们如下陈述我们的主要结果：

定理 1.2（循环秩，定理的非正式版本 5.1）。对于任意 $k \geq 1$，$\epsilon \in (0,1)$，以及一个三阶张量 $a \in \mathbb{R}^{n \times n \times n}$，存在一种算法，其运行时间为

$$O(\text{nnz}(A)) + n \, \text{poly}(k, 1/\epsilon) + \exp(O(\text{poly}(k)/\epsilon))$$

时间，并生成 U、 V 和 W，它们是维度为 $n \times k \times k$ 的张量，满足以下条件：

$$\| \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} U_{*,i,j} \otimes V_{*,j,l} \otimes v_{*,l,i} - A \|_F \leq (1+\epsilon) \min_{\text{cycle}} \min_{\text{rank}-k \ A_k} \|A_k - A\|_F^2$$

路线图。 我们安排论文如下。在第 2 节中，我们讨论一些相关工作。在第 3 节中，我们概述我们所采用的技术。在第 4 节中，我们介绍本文所需的预备知识。在第 5节中，我们给出正式结果及其分析。

2 相关工作

2.1 Transformer 理论

优化与收敛 在优化领域，[SZKS21] 致力于研究单头注意力机制的行为，以模拟学习序列到序列模型的过程；而[ZKV+20]则强调了注意力模型中的自适应方法。[GMS23] 近期开始关注与指数激活函数相关的过参数化问题。指数激活函数是[GMS23], 的研究重点，他们考察了过参数化双层神经网络的收敛性，并确定了实现收敛所需的最少神经元数量。此外，[LSZ23] 研究了正则化指数回归问题，并提出了一种以输入稀疏度时间运行的算法。 [LLR23] 全面阐释了 Transformer 如何学习“语义结构”，即其捕捉词共现模式的能力。

快速注意力计算 [ZHDK23] 和 [AS23] 的工作主要关注注意力计算的静态版本。具体而言，在正面结果方面， [AS23] 提出了一种算法；在负面结果方面， [AS23] 在假设（强）指数时间假说成立的前提下证明了其困难性。[BSZ23] 定义并研究了该问题的动态版本，并给出了正面和负面结果。[BSZ23] 的算法结果建立在求解线性规划时使用的惰性更新技术之上 [CLS19, Bra20]。 [BSZ23] 的困难性结果建立在提示 MV 猜想之上 [BNS19]。 [DMS23] 研究了注意力计算中特征维度远大于句子长度的情形，并分别提供了随机化算法和确定性算法来稀疏化特征维度。

上下文学习 在最近的一项工作中 [ASA+22]，研究表明，基于 Transformer 的上下文学习器能够以隐式方式执行传统学习算法：通过在其激活值中编码较小的模型，并随着新示例在上下文中呈现而持续更新这些模型。研究聚焦于一个定义明确的任务：在上下文条件下训练模型，使其学习某一类函数，例如线性函数， [GTLV22] 旨在加深对上下文学习的理解。他们的研究目标是确定：当向模型提供源自某一类函数中部分函数的数据时，模型能否经过训练，学会该类函数中的“大多数”函数。 [ONR+22] 解释了 Transformer 作为上下文学习器的机制。他们表明，Transformer 在上下文任务中的训练过程，与一些基于梯度下降的元学习形式具有相似性。

隐私。 考虑到滥用大语言模型可能带来的负面影响， [KGW+23]提出了一种为专有语言模型添加水印的方法，且不会对文本质量造成不利影响。由 [KGW+23]提出了一种无需访问模型参数或 API 即可检测语言模型中水印的新型算法。

2.2 数值线性代数

低秩近似 低秩近似问题有许多变体，已有大量研究，例如，Frobenius 范数下的低秩近似 [CW13、NN13], 矩阵 CUR 分解 [BW14, SWZ17, SWZ19c], 加权低秩近似 [RSW16], 逐项 ℓ_1 范数低秩近似 [SWZ17, SWZ19a], 一般范数下的列子集选择 [SWZ19b], 张量低秩近似 [SWZ19c], 以及张量回归 [DSSW18, DJS+19, RSZ22, SWYZ21]。低秩近似还广泛应用于其他问题，例如切割平面法[JLSW20], 积分最小化 [JLSZ23], 神经网络训练 [BPSW21, SZZ21, Zha22]。

输入稀疏算法 多年来，许多研究致力于设计输入稀疏时间算法 [CW13, NN13, WZ16, BWZ16, RSW16, SWZ17, SWZ19c, SYYZ22, DSW22]。对于矩阵低秩近似和线性回归问题， [CW13] 首次提出了输入稀疏度时间算法。借助稀疏嵌入矩阵 S，可以在输入稀疏时间 nnz(A) 内计算它与任意输入矩阵 A 的乘积。之后，另一项工作 [NN13] 提出了一些称为无偏稀疏范数近似投影（OSNAP）的草图矩阵，改进了 [CW13]中的草图维度。 [RSW16] 给出了加权低秩近似的输入稀疏时间算法。 [SWZ17] 将 [CW13, NN13] 的 Frobenius 范数低秩近似问题推广为逐项 ℓ_1 低秩近似问题。 [SWZ19c] 研究了张量低秩近似问题。最近，[SYYZ22]提出了一种结构化 John 椭球计算算法，将此前的工作 [CCLY19] 推进到输入稀疏时间。另一项近期工作 [DSW22] 提出了一种新颖的算法，能够在输入稀疏时间内最小化差异。

3 技术概述

利用 $U^* \in \mathbb{R}^{n^2 \times k^2}$、 $V^* \in \mathbb{R}^{n^2 \times k^2}$ 和 $W^* \in \mathbb{R}^{n^2 \times k^2}$，我们可以将 A_k 重写如下

$$A_k = \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} U^*_{i+k(j-1)} \otimes V^*_{j+k(l-1)} \otimes W^*_{l+k(i-1)}$$

我们证明的主要思路是，先证明 A 存在矩阵分解，然后使用相应的算法可靠地找到所需的矩阵。

旋转与草图步骤 1。首先，我们考虑假设的回归问题 $\min_{U \in \mathbb{R}^{n \times k^2}} \|A_1 - UZ_1\|_F^2$。对于 $r = O(k^2/\epsilon)$ 和 $S_1 \in \mathbb{R}^{n^2 \times r}$，其中独立同分布的正态随机变量记为 $N(0, 1/r)$，且 $\widehat{U} = \operatorname{argmin}_{U \in \mathbb{R}^{n \times k^2}} \|UZ_1S_1 - A_1S_1\|_F^2$，我们有

$$\|A_1 - \widehat{U}Z_1\|_F^2 \leq (1+\epsilon) \min_{U \in \mathbb{R}^{n \times k^2}} \|A_1 - UZ_1\|_F^2.$$

通过选择 $\widehat{U} = A_1S_1(Z_1S_1)^\dagger$，我们可以证明，至少可以找到一个矩阵 $\widehat{U}$ 满足上述方程。

旋转与草图步骤 2。根据第一步，我们可以创建一个秩为 k 的张量，定义如下

$$B := \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k} \widehat{U}_{i+k(j-1)} \otimes V^*_{j+k(l-1)} \otimes W^*_{j+k(i-1)}$$

$\widehat{U}$从矩阵 UZ_1 中。

由于 $\widehat{U}$位于 A_1S_1的列空间中，秩-1张量集合 $\{(A_1S_1)_{a+k(b-1)} \otimes V^*_{b+k(c-1)} \otimes W^*_{c+k(a-1)}\}_{a\in[r],b,c\in[k]}$ 张成了包含 B的空间。可以将一个严格秩-k 张量 B 表示为形如 $\{U^*_{a+k(b-1))} \otimes (A_2S_2)_{b+k(c-1)} \otimes W^*_{c+k(a-1)}\}_{a,c\in[k],b\in[r]}$ 的秩-1张量的线性组合，并且秩-1 张量集合$\{U^*_{a+k(b-1)} \otimes V^*_{b+k(c-1)} \otimes (A_3S_3)_{c+k(a-1)}\}_{a,b\in[k],c\in[r]}$ 张成了一个包含严格秩-k 张量 B的子空间。

我们注意到第 A_2S_2的行空间，其中第 (j,l) 行是 $\sum_{i=1}^{k} \widehat{U}_{i+k(j-1)} \otimes W^*_{l+k(i-1)}$，$\widehat{U}$是 UZ_1 的前 k 个列向量，其中第 (j,l) 行是 $\mathrm{vec}(\sum_{i=1}^{k} \widehat{U}_{i+k(j-1)} \otimes W^*_{l+k(i-1)})$。现在，我们相对于 VZ_2 重新定义 $\widehat{U}$，对每个 $j \in [k]$ 和 $l \in [k]$均如此。我们考虑如下形式的回归问题，

$$\min_{V} \|VZ_2 - A_2\|_F^2.$$

如果 $S_2 \in \mathbb{R}^{n^2\times r}$ 是一个具有独立同分布的高斯元素的矩阵，那么 $\widehat{V} = A_2S_2(Z_2S_2)^\dagger$ 满足

$$\|\widehat{V}Z_2 - A_2\|_F^2 \le (1+\epsilon)^2\|A_k - A\|_F^2.$$

$\widehat{U}$矩阵 B 可以通过对 VZ_2 重新张量化得到，然后我们有

$$\|B - A\|_F^2 = \|\widehat{V}Z_2 - A_2\|_F^2 \le (1+\epsilon)^2\|A_k - A\|_F^2.$$

$\widehat{U}$旋转与草图步骤 3。回想一下， V的列都在 A_2S_2的行空间中，并且 Z_2的行定义为 $\mathrm{vec}(\sum_{i=1}^{k} \widehat{U}_{i+k(j-1)} \otimes W^*_{l+k(i-1)})$，对每个 $j \in [k]$ 和 $l \in [k]$均如此，其中 $\widehat{U}$ 的列在矩阵 A_1S_1的行空间中，因此， B 属于秩-1 张量集合的张成空间

$$\{(A_1S_1)_{a+k(b-1)} \otimes (A_2S_2)_{b+k(c-1)} \otimes \widehat{V}_{c+k(a-1)}\}_{a,b\in[r],c\in[k]}.$$

鉴于 $Z_3 \in \mathbb{R}^{r^2\times n^2}$ 是一个矩阵，其行由所有数对 a、$b \in [r]$对应的 $\mathrm{vec}((A_1S_1)_{a+k(b-1)} \otimes (A_2S_2)_{b+k(c-1)})$ 构成，我们将聚焦于以下回归问题。

$$\min_{W} \|WZ_3 - A_3\|_F^2.$$

通过求解另一个回归问题，可以在多项式时间内求解 $W \in \mathbb{R}^{n\times r^2}$ 。将 WZ_3 重新张量化为新的张量 B 后，我们可以找到一个秩为 $r^2 = O(k^2/\epsilon^2)$ 的张量 B，使得

$$\|B - A\|_F^2 \le (1+\epsilon)^2\|A_k - A\|_F^2 = (1+O(\epsilon))\|A_k - A\|_F^2.$$

输入稀疏度缩减。 为了获得 $n\mathrm{poly}(k/\epsilon)+\mathrm{nnz}(A)$ 的输入稀疏度运行时间保证（定理 4.15），我们可以用一个稀疏 CountSketch 矩阵和一个高斯矩阵来替换 S_1 和 S_2，从而能够在所需的 $\mathrm{nnz}(A) + n\mathrm{poly}(k/\epsilon)$ 时间内计算 A_1S_1 和 A_2S_2 。

为了快速求解 $\min_W \|A_3 - WZ_3\|_F^2$ 并写出矩阵 Z_3 而不耗费 r^2n^2 时间，本文采用 TensorSketch 技术，该技术由 [Pag13、PP13] 引入。但正如 [ANW14], 所示，我们需要证明近似矩阵乘积（AMP）和子空间嵌入（SE）的一些更强性质，才能使其成立。当 AMP 和 SE 得到满足时，我们就能求解以下形式的"草图化版本"回归问题：

$$\min_W \|A_3S_3 - WZ_3S_3\|_F^2.$$

该草图的关键在于，矩阵乘积 A_3S_3 和 Z_3S_3 可以在与 $\text{nnz}(A) + n\,\text{poly}(k/\epsilon)$ 成正比的稀疏时间内计算。将它们结合起来，我们就能在 $n\text{poly}(k/\epsilon)$ 时间内解决这个问题。

为了达到 A 的稀疏时间，我们使用 CountSketch 矩阵来增强算法。具体而言，我们选取三个 CountSketch 矩阵 T_1、T_2、T_3，其草图维度为 $t = n \cdot \text{poly}(k, 1/\epsilon)$。

我们定义

$$D := \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k}(T_1A_1S_1X_1)_{a,i+k(j-1)} \cdot (T_2A_2S_2X_2)_{b,j+k(l-1)} \cdot (T_3A_3S_3X_3)_{c,l+k(i-1)}.$$

将它们应用于上述迭代论证后，我们现在尝试通过以下目标多项式函数实现最小化

$$\sum_{a,b,c\in[t]} (D - (A(T_1, T_2, T_3))_{a,b,c})^2$$

通过选择 $3rk = O(k^2/\epsilon)$ 个未知数 X_1、X_2、X_3。

通过多项式系统求解器解决小规模问题。令 X_1 X_2 X_3 $r \times k^2$、和分别为由未知数构成的矩阵，我们可以将 U 重新表示为 $A_1\widehat{S_1X_1}$，将 V 表示为 $\widehat{A_2S_2X_2}$，并将 W 表示为 $A_3S_3X_3$。我们令 Z_1 的各行是 $\text{vec}(V^*_{j+k(l-1)} \otimes W^*_{l+k(i-1)})$，$Z_2$ 的各行是 $\text{vec}(U_{i+k(j-1)} \otimes W^*_{j+k(l-1)})$，而 Z_3 的各行是 $\text{vec}(U_{i+k(j-1)} \otimes V_{j+k(l-1)})$。

由此可得秩为 k 的张量

$$B = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k}(A_1S_1X_1)_{i+k(j-1)} \otimes (A_2S_2X_2)_{j+k(l-1)} \otimes (A_3S_3X_3)_{l+k(i-1)}$$

其中 $\|A - B\|_F^2 \le (1+\epsilon)^3\|A - A_k\|_F^2$。

最小 化任务可以通过最小化如下六次多项式来实现 s

$$\sum_{a,b,c\in[n]} (\sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k}(A_1S_1X_1)_{a,i+k(j-1)} \cdot (A_2S_2X_2)_{b,j+k(l-1)} \cdot (A_3S_3X_3)_{c,l+k(i-1)} - A_{a,b,c})^2$$

通过选择 $3rk = O(k^2/\epsilon)$ 个不定元 X_1、X_2、X_3。我们可以得到 X_1、X_2、X_3，使得该解至少是 $(1 + O(\epsilon))$-近似的。

假设矩阵 U^*、V^*、W^* 中的元素均有界于 $2^{\text{poly}(n)}$，如定理4.17所述，该多项式可以在 $\text{poly}(n)$ 时间内最小化，误差不超过加性误差 $2^{-\text{poly}(n)}$ [Ren92、BPR96]。类似方法也可用于获得 OPT 真实值的相对误差近似，并处理 A_k 不存在的情况。

4 预备知识

本节介绍本文工作的预备知识。在第4.1节中，我们介绍所用符号。在第4.2节中，我们给出不同类型张量秩的定义。在第4.3节中，我们介绍张量循环秩的相关定义。在第4.4节中，我们介绍子空间嵌入和近似矩阵乘积的定义。在第4.5 节中，我们介绍草图技术的相关结果。在第4.6节中，我们介绍张量低秩近似算法。在第4.7节中，我们介绍输入稀疏近似技术。在第4.8 节中，我们介绍一项用于求解小规模问题的已有结果。

4.1 记号

我们使用记号 $[n]$ 来表示集合 $1, 2, \cdots, n$，其中 n为任意正整数。此外，对于向量 $x \in \mathbb{R}^n$，$\|x\|_2$ 表示其 ℓ_2 范数，其定义为 $\|x\|_2 := (\sum i = 1^n x_i^2)^{1/2}$。对于矩阵 $A \in \mathbb{R}^{n\times n}$，我们使用 $\|A\|_F$ 表示其 Frobenius 范数，即 $\|A\|_F := (\sum_{i=1}^n \sum_{j=1}^n A_{i,j}^2)^{1/2}$。对于张量 A，我们使用 $\|A\|_F$ 表示其 Frobenius 范数，即 $\|A\|_F := (\sum_{i=1}^n \sum_{j=1}^n \sum_{l=1}^n A_{i,j,l}^2)^{1/2}$。

对于一个 y张量 A，我们使用 nnz(A) 表示张量 A 中非零条目的数量 .

对于可逆方阵 A，我们用 A^{-1} 表示其真正的逆矩阵。

设 $A \in \mathbb{R}^{n\times k}$的SVD为 $M\Sigma N^\top$，其中 $M \in \mathbb{R}^{n\times k}$ 和 $N \in \mathbb{R}^{k\times k}$ 的列是正交归一的，且 $\Sigma \in \mathbb{R}^{k\times k}$ 是一个对角矩阵。我们将 Moore-Penrose 伪逆矩阵A记为 $A^\dagger \in \mathbb{R}^{k\times n}$，即 $A^\dagger = N\Sigma^{-1}M^\top$。

我们定义以下两种 $\otimes$ 乘积。

定义 4.1（向量的$\otimes$积）。给定三个向量 $a, b, c \in \mathbb{R}^n$, $a \otimes b \otimes c$用于表示一个大小为 $n \times n \times n$ 的张量，其中第 i, j, l 个元素为 $a_i b_j c_l$。

定义 4.2（矩阵的 $\otimes$ product）。对于 m 矩阵 $A_1 \in \mathbb{R}^{n_1\times k}$, $A_2 \in \mathbb{R}^{n_2\times k}$, $\cdots$, $A_m \in \mathbb{R}^{n_m\times k}$，我们将张量积表示为以下形式

$$A_1 \otimes A_2 \otimes \cdots \otimes A_q = \sum_{i=1}^{k} (A_1)_i \otimes (A_2)_i \otimes \cdots \otimes (A_q)_i \in \mathbb{R}^{n_1\times n_2\times \cdots \times n_m},$$

其中，我们用 $(A_j)_i$表示矩阵 A_j 的第 i 列。

我们定义以下将张量转换为向量的 vec() 操作，

定义 4.3（vec()，将张量转换为向量）。对于任意张量 $T \in \mathbb{R}^{n_1\times n_2\times \cdots \times n_q}$，我们定义

$$\mathrm{vec}(T) \in \mathbb{R}^{1\times \prod_{i=1}^q n_i}$$

作为一个行向量。对于 $t = (j_1 - 1)\prod_{i=2}^q n_i + (j_2 - 1)\prod_{i=3}^q n_i + \cdots + (j_{q-1} - 1)n_q + j_q$，其中$j_1 \in [n_1], j_2 \in [n_2] \ldots, j_q \in [n_q]$，vec($T$) 的第 t 个元素为 $T_{j_1,j_2,\cdots,j_q}$。

举例来说，给定 $u = \begin{bmatrix} 5 \\ 4 \end{bmatrix}, v = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$，则 vec($u \otimes v$) $= \begin{bmatrix} 15 & 10 & 5 & 12 & 8 & 4 \end{bmatrix}$.

4.2 张量秩

本节中，我们定义张量的以下几种秩。

定义 4.4（经典秩）。我们称张量 $U \in \mathbb{R}^{n\times n\times n}$ 的经典秩为 k，如果 $k \in \mathbb{Z}_{\geq 0}$ 是满足存在三个矩阵 A、B、$C \in \mathbb{R}^{n\times k}$ 的最小正整数，使得

$$U_{i,j,l} = \sum_{a=1}^{k}\sum_{b=1}^{k}\sum_{c=1}^{k} A_{i,a}B_{j,b}C_{l,c}$$

我们将 Tucker 秩定义如下：

定义 4.5（Tucker 秩）。若 $k \in \mathbb{Z}_{\geq 0}$ 是满足以下条件的最小正整数，则称张量 $U \in \mathbb{R}^{n\times n\times n}$ 的 Tucker 秩为 k：存在三个矩阵 $A \in \mathbb{R}^{n\times k}$、$B \in \mathbb{R}^{n\times k}$、$C \in \mathbb{R}^{n\times k}$ 以及一个（更小的）张量 $D \in \mathbb{R}^{k\times k\times k}$，使得 $\forall (i,j,l) \in [n]\times[n]\times[n]$，我们有

$$U_{i,j,l} = \sum_{i'=1}^{k}\sum_{j'=1}^{k}\sum_{l'=1}^{k} D_{i',j',l'}A_{i,i'}B_{j,j'}C_{l,l'}.$$

接下来，我们定义列车秩，

定义 4.6（训练秩）。我们称张量 $U \in \mathbb{R}^{n\times n\times n}$ 具有 k 训练秩，如果 $k \in \mathbb{Z}_{\geq 0}$ 是满足以下条件的最小正整数：存在三个矩阵 $A \in \mathbb{R}^{1\times n\times k}$、$B \in \mathbb{R}^{k\times n\times k}$、$C \in \mathbb{R}^{k\times n\times 1}$，使得 $\forall i$、j、$l \in [n]\times[n]\times[n]$ 我们有，

$$U_{i,j,l} = \sum_{i_1=1}^{1}\sum_{i_2=1}^{k}\sum_{i_3=1}^{k}\sum_{i_4=1}^{1} A_{i_1,i,i_2}B_{i_2,j,i_3}C_{i_3,l,i_4}$$

4.3 张量循环秩的定义

张量循环秩的形式化定义如下。

定义 4.7（循环秩）。如果存在三个矩阵 A、B、$C \in \mathbb{R}^{n\times k^2}$，使得 $U \in \mathbb{R}^{n\times n\times n}$ 的循环秩为 k，则称 $U \in \mathbb{R}^{n\times n\times n}$ 具有循环秩 k。

$$U_{i,j,l} = \sum_{a=1}^{k}\sum_{b=1}^{k}\sum_{c=1}^{k} A_{i,a+k(b-1)}B_{j,b+k(c-1)}C_{l,c+k(a-1)}$$

or

$$A_{i,j,l} = \sum_{a=1}^{k}\sum_{b=1}^{k}\sum_{c=1}^{k} \overline{U}_{i,a,b}\overline{V}_{j,b,c}\overline{W}_{l,c,a}$$

其中，$\overline{U} \in \mathbb{R}^{n\times k\times k}$ 是 $U \in \mathbb{R}^{n\times k^2}$的张量化，$\overline{V} \in \mathbb{R}^{n\times k\times k}$ 是 $V \in \mathbb{R}^{n\times k^2}$的张量化，$\overline{W} \in \mathbb{R}^{n\times k\times k}$ 是 $W \in \mathbb{R}^{n\times k^2}$的张量化。

4.4 近似矩阵乘积与子空间嵌入

我们将子空间嵌入定义为 [Sar06] 如下。

定义 4.8 (子空间嵌入[Sar06])。 对于矩阵 $A \in \mathbb{R}^{n\times d}$，我们称矩阵 S 是一个子空间嵌入 SE(n, d, ϵ, δ)，作用于 A 当

$$\Pr[\|SAx\|_2^2 = (1\pm\epsilon)\|Ax\|_2^2, \forall x \in \mathbb{R}^d] \geq 1-\delta.$$

我们将近似矩阵乘积定义如下（例如，参见 [Woo14] ）。

定义 4.9（弗罗贝尼乌斯近似矩阵乘积）。设 $\epsilon \in (0,1)$ 表示精度参数。设 $\delta \in (0,1)$ 表示失败概率。给定矩阵 A 和 B，其中 A 和 B 各有 n 行。若矩阵 S 是关于 A, B 的弗罗贝尼乌斯范数近似矩阵乘积（FAMP(n, ϵ, δ)），则

$$\Pr[\|A^\top B - A^\top S^\top SB\|_F \leq \epsilon \cdot \|A\|_F \cdot \|B\|_B] \geq 1-\delta.$$

4.5 高斯变换与 CountSketch

我们定义以下几类草图矩阵。

定义 4.10（CountSketch 或稀疏嵌入矩阵）。定义如下:

- $\sigma \in \mathbb{R}$: 一个标量;
- $D \in \mathbb{R}^{n\times n}$: 一个对角矩阵，其中每个对角元素均以相同的概率独立地从 $\{-1,+1\}$中选取;
- $h:\ [n] \to [m]$: 一个随机映射，其中 $\forall i \in [n]$, $j \in [m]$，我们定义

$$\Pr[h(i) = j] = 1/m.$$

- $\Pi \in \{0,1\}^{m\times n}$: 一个二元矩阵，其中对于所有 $(j,i) \in [m]\times[n]$，其元素定义为

$$\Phi_{j,i} := \begin{cases} 1 & \text{if } h(i) = j \\ 0 & \text{otherwise} \end{cases}$$

于是，CountSketch 矩阵 $\Pi \in \mathbb{R}^{m\times n}$ 定义为

$$\Pi := \sigma \cdot \Phi D.$$

对于任意实矩阵 $A \in \mathbb{R}^{n\times d}$，矩阵乘积 ΠA 可在输入稀疏度 O(nnz(A)) 时间内计算。对于任意实张量 $A \in \mathbb{R}^{n\times d_1\times d_2}$，乘积 ΠA 可在输入稀疏度O(nnz(A)) 时间内计算。对于三个 CountSketch 矩阵 Π_i、Π_2、Π_3，对于任意输入 3 阶张量 $A \in \mathbb{R}^{n_1\times n_2\times n_3}$，$A(\Pi_1,\Pi_2,\Pi_3)$ 可在输入稀疏度 O(nnz(A)) 时间内计算。

在这 在未作说明的情况下，上述记号中的标量 σ 被假定为 1 n.

定义 4.11（高斯变换或高斯矩阵）。令 $\sigma \in \mathbb{R}$ 为一个标量。令$G \in \mathbb{R}^{m\times n}$ 为一个矩阵，使得其中每个条目都从 $\mathcal{N}$(0,1)中独立同分布地选取。然后我们将 $S = \sigma \cdot G \in \mathbb{R}^{m\times n}$ 定义为高斯矩阵或高斯变换。对于任意输入矩阵 $A \in \mathbb{R}^{n\times d}$，矩阵乘积 SA 可以在 $O(m \cdot$nnz(A))的输入稀疏度时间内计算完成。对于任意输入张量，该乘积可以在 $O(m \cdot$nnz(A)) 的输入稀疏度时间内计算完成。

在未作特别说明时，上述记法默认标量 σ 为 $1/\sqrt{m}$。我们结合 CountSketch 和高斯变换得到以下变换：

定义 4.12 (CountSketch + 高斯 变换)。令 $\Pi \in \mathbb{R}^{t\times n}$ 表示一个 CountSketch 矩阵（定义 *4.10*）。令 $S \in \mathbb{R}^{m\times t}$ 表示一个高斯矩阵（定义 *4.11*）。定义 $S' = S\Pi$。于是，对于任意输入矩阵 $A \in \mathbb{R}^{n\times d}$，矩阵乘积 $S'A$可在输入稀疏度 $O(\mathrm{nnz}(A) + dtm^{\omega-2})$ 时间内计算，其中我们用 ω 表示矩阵乘法指数。

对于上述草图矩阵，我们有以下有用结果，它说明了该矩阵的子空间嵌入（SE）和 FAMP 性质。

引理 4.13（ [SWZ19c] 中的引理 B.22）。令 $m_2 = \Omega(k^2+k/\varepsilon)$，$m_1 = \Omega(k/\varepsilon)$。令 $\Pi \in \mathbb{R}^{m_2\times n}$ 为一个 CountSketch 矩阵（定义 4.10）。令 $S \in \mathbb{R}^{m_1\times m_2}$ 为一个高斯矩阵（定义 4.11）。然后我们根据定义 4.12, 使用 $S\Pi$ 定义 S'。那么，对于任意固定矩阵 $C \in \mathbb{R}^{n\times k}$，$S'$ 是一个 SE$(n, k, 1/3, 0.01)$（定义 4.8）。此外，对于任意两个具有相同行数的固定矩阵 A和 B，它是一个 FAMP$(n, O(\epsilon/k)\ 0.01)$（定义 4.9）。

定理4.14（ [CW13]中的定理 36）。给定 $A \in \mathbb{R}^{n\times k}$、$B \in \mathbb{R}^{n\times d}$，假设 $S \in \mathbb{R}^{m\times n}$ 是针对 A 的 SE（n、k、$1/\sqrt{2}$、δ），并满足是一个FAMP（n、$O(\sqrt{\epsilon/k}$、$\delta)$）矩阵 A 和 C，其中 C有 n 行，而 C 取决于 A 和 B。如果

$$\widehat{X} = \arg\min_{X\in\mathbb{R}^{k\times d}} \|SB - SAX\|_F^2,$$

then

$$\|B - A\widehat{X}\|_F^2 \le (1+\varepsilon)\min_{X\in\mathbb{R}^{k\times d}} \|B - AX\|_F^2.$$

4.6 张量低秩近似

在本节中，我们给出以下张量低秩近似算法。

定理 4.15 (第 31 页中的定理 C.1，见 [SWZ19c])。如果满足以下条件

- 令 n 表示一个正整数。
- 设 $A \in \mathbb{R}^{n\times n\times n}$ 为一个三阶张量。
- 对于任意整数 $k \ge 1$。
- 对于任意精度参数 $\epsilon \in (0, 1)$。

然后，存在这样一种算法：

- *runs in* $\exp(O(k^2/\epsilon) + n\,\mathrm{poly}(k, 1/\epsilon) + O(\mathrm{nnz}(A))$ *time*
- 构造三个矩阵 U、V 和 W，其维数为 $n \times k$，使得

$$\left\|\sum_{i=1}^{k} U_i \otimes V_i \otimes W_i - A\right\|_F^2 \le (1+\epsilon)\min_{\mathrm{rank}-k\ A_k} \|A_k - A\|_F^2$$

- 成功概率为 $9/10$。

算法 1 求解 Frobenius 范数低秩近似问题的算法

过程 FLowRankApprox(A, n, k, ϵ) ▷定理 4.15
  对 $i = 1 \to 3$ 执行
    $s_i \leftarrow O(k/\epsilon)$。
  结束循环
  选择草图矩阵 $S_1 \in \mathbb{R}^{n \times s_1}$、$S_2 \in \mathbb{R}^{n \times s_2}$、$S_3 \in \mathbb{R}^{n \times s_3}$。 ▷定义 4.12
  对于 $i = 1 \to 3$，执行
    计算 $A_i S_i$。
  结束循环
  Y_1、Y_2、Y_3、$C \leftarrow$FInputSparsityReduction（A、A_1S_1、A_2S_2、A_3S_3、n、s_1、s_2、s_3、k、ϵ）。 ▷算法 2
  对于 $i = 1 \to 3$，执行
    为 $X_i \in \mathbb{R}^{s_i \times k}$ 创建变量
  结束循环
  对 $\|(Y_1X_1) \otimes (Y_2X_2) \otimes (Y_3X_3) - C\|_F^2$ 运行多项式系统验证器。
  返回 $A_1S_1X_1$，$A_2S_2X_2$，以及 $A_3S_3X_3$。
结束过程

4.7 输入稀疏性缩减

本节将介绍以下关键引理，以实现输入稀疏度运行时间。

引理 4.16 (第 35 页中的引理 C.3，见 [SWZ19c])。若满足以下条件

- 令 $\text{poly}(k, 1/\epsilon) \geq b_1 b_2 b_3 \geq k$。
- 假设有三个矩阵 $V_1 \in \mathbb{R}^{n \times b_1}$、$V_2 \in \mathbb{R}^{n \times b_2}$ 和 $V_3 \in \mathbb{R}^{n \times b_3}$。
- 给定一个 $n \times n \times n$ 大小的张量 A
- 设 $\mathcal{T} = O(\text{nnz}(V_1) + \text{nnz}(V_2) + \text{nnz}(V_3) + \text{nnz}(A)) = O(n\text{poly}(k/\varepsilon) + \text{nnz}(A)$

  令 $c_1 = c_2 = c_3 = \text{poly}(k, 1/\epsilon)$

假设存在一种算法，

- 运行时间为 $\mathcal{T}$
- 构造一个张量 $C \in \mathbb{R}^{c_1 \times c_2 \times c_3}$
- 构造三个矩阵 $\widehat{V}_1 \in \mathbb{R}^{c_1 \times b_1}$、$\widehat{V}_2 \in \mathbb{R}^{c_2 \times b_2}$ 和 $\widehat{V}_3 \in \mathbb{R}^{c_3 \times b_3}$
- 以至少 0.99 的概率，对于所有满足条件的 $\alpha > 0$、X_1、$X_1' \in \mathbb{R}^{b_1 \times k}$、$X_2$、$X_2' \in \mathbb{R}^{b_2 \times k}$、$X_3$、$X_3' \in \mathbb{R}^{b_3 \times k}$，有：

$$\left\| \sum_{i=1}^{k} (\widehat{V}_1 X_1')_i \otimes (\widehat{V}_2 X_2')_i \otimes (\widehat{V}_3 X_3')_i - C \right\|_F^2 \leq \alpha \left\| \sum_{i=1}^{k} (\widehat{V}_1 X_1)_i \otimes (\widehat{V}_2 X_2)_i \otimes (\widehat{V}_3 X_3)_i - C \right\|_F^2,$$

然后，

$$\left\| \sum_{i=1}^{k} (V_1 X_1')_i \otimes (V_2 X_2')_i \otimes (V_3 X_3')_i - A \right\|_F^2 \leq (1+\epsilon)\alpha \left\| \sum_{i=1}^{k} (V_1 X_1)_i \otimes (V_2 X_2)_i \otimes (V_3 X_3)_i - A \right\|_F^2.$$

算法 2在该算法中，我们将目标函数的规模从 poly(n) 大小缩减至 poly(k) 大小

1: 过程 FInputSparsityReduction($A, V_1, V_2, V_3, n, b_1, b_2, b_3, k, \epsilon$) ▷引理 4.16 2: 对 $i = 1 \to 3$ 执行 3: $c_i \leftarrow \text{poly}(k, 1/\epsilon)$. 4: 结束循环 5: 对 $i = 1 \to 3$ 执行 6: 选择稀疏嵌入矩阵 $T_i \in \mathbb{R}^{c_i \times n}$ ▷ 定义 4.10 7: 结束循环 8: 对 $i = 1 \to 3$ 执行 9: $\widehat{V}_i \leftarrow T_i V_i \in \mathbb{R}^{c_i \times b_i}$, 10: 结束循环 11: $C \leftarrow A(T_1, T_2, T_3) \in \mathbb{R}^{c_1 \times c_2 \times c_3}$. 12: 返回 V_1, V_2, V_3 和 C. 13: 结束过程

4.8 求解小规模问题

这里我们介绍以下定理，以解决算法中的小规模问题。

定理 4.17 (第 90 页的定理 D.11，见 [SWZ19c])。 如果满足以下条件

- 假设我们有 $\max\{t_1, d_1, t_2, d_2, t_3, d_3\} \leq n$。
- 设 A 为一个大小为 $t_1 \times t_2 \times t_3$ 的张量
- 给定三个矩阵
  - $t_1 \times d_1$ 矩阵 T_1
  - $t_2 \times d_2$ 矩阵 T_2
  - $t_3 \times d_3$ 矩阵 T_3
- 对于任意 $\delta > 0$，都存在一个解

$$\min_{X_1, X_2, X_3} \left\| \sum_{i=1}^{k} (T_1 X_1)_i \otimes (T_2 X_2)_i \otimes (T_3 X_3)_i - A \right\|_F^2 := \mathrm{OPT},$$

并且，可以使用 $O(n^\delta)$ 位表示 X_i 中的每个元素。

那么就存在一种算法，

- 运行时间为 $n^{O(\delta)} \cdot \exp(O(d_1 k + d_2 k + d_3 k))$
- 构造三个矩阵 $\widehat{X}_1$ - $\widehat{X}_2$ - X_3

- 使得 $\|(T_1 \widehat{X}_1) \otimes (T_2 \widehat{X}_2) \otimes (T_3 \widehat{X}_3) - A\|_F^2 = \mathrm{OPT}$。

5 张量循环秩

本节将介绍我们的主要算法及其分析。

算法 3 我们的主要算法

过程 FLowCycleRankApprox(A, n, k, ϵ) ▷ 定理5.1

针对 $i = 1 \to 3$ 进行操作

$s_i \leftarrow O(k^2/\epsilon)$】【。

$t_i \leftarrow$ 多项式$(k, 1/\epsilon)$。

选择草图矩阵 $S_i \in \mathbb{R}^{n \times s_i}$ ▷定义 4.12

选择草图矩阵 $T_i \in \mathbb{R}^{t_i \times n}$

结束循环

对于 $i = 1 \to 3$ 执行

计算 $A_i S_i$

结束循环

对 $i = 1 \to 3$ 执行do

计算 $T_i A_i S_i$

结束循环

计算 $B \leftarrow A(T_1, T_2, T_3)$。

对于 $i = 1 \to 3$ do

为 $X_i \in \mathbb{R}^{s_i \times k^2}$ 创建变量。

结束循环

为 $C \in \mathbb{R}^{k^2 \times k^2 \times k^2}$ 创建变量。

$D \leftarrow \| \sum_{i_1=1}^{k} \sum_{i_2=1}^{k} \sum_{i_3=1}^{k} (Y_1 X_1)_{i_1 + k(i_2-1)} (Y_2 X_2)_{i_2 + k(i_3-1)} (Y_3 X_3)_{i_3 + k(i_1-1)} - B \|_F^2$

为 D 运行多项式系统验证器。

return $A_1 S_1 X_1$, $A_2 S_2 X_2$, 以及 $A_3 S_3 X_3$。

结束过程

定理 5.1 (循环秩，定理 1.2的正式版本)。 对于任意 $k \geq 1$, $\epsilon \in (0,1)$，以及一个三阶张量 $A \in \mathbb{R}^{n \times n \times n}$，存在一种算法，其运行时间为

$$O(\mathrm{nnz}(A)) + n \, \mathrm{poly}(k, 1/\epsilon) + \exp(O(k^6/\epsilon))$$

时间，并生成三个维度为 $n \times k \times k$ 的张量 U、 V、 W，满足以下条件:

$$\| \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} U_{*,i,j} \otimes V_{*,j,l} \otimes W_{*,l,i} - A \|_F \leq (1+\epsilon) \min_{\mathrm{cycle} \ \mathrm{rank}-k \ A_k} \| A_k - A \|_F^2$$

或等价地

$$\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} (U_2)_{*,i+k(j-1)} \otimes (V_2)_{*,j+k(l-1)} \otimes (W_2)_{*,l+k(i-1)} - A \right\|_F^2 \leq (1+\epsilon) \min_{\mathrm{cycle} \ \mathrm{rank}-k \ A_k} \| A_k - A \|_F^2$$

holds with probability $9/10$.

Here

- $U_2 \in \mathbb{R}^{n \times k^2}$ 是张量 $U \in \mathbb{R}^{n \times k \times k}$ 的展平版本。

- $V_2 \in \mathbb{R}^{n\times k^2}$ 是张量 $V \in \mathbb{R}^{n\times k\times k}$ 的展平版本。
- $W_2 \in \mathbb{R}^{n\times k^2}$ 是张量 $W \in \mathbb{R}^{n\times k\times k}$的展平版本。

证明。我们将 OPT 定义为

$$\mathrm{OPT} := \min_{\text{cycle}} \min_{\text{rank}-k\ A'} \|A' - A\|_F^2.$$

假设最优

$$A_k = \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k} U^*_{i+k(j-1)} \otimes V^*_{j+k(l-1)} \otimes W^*_{l+k(i-1)}.$$

令 $V^* \in \mathbb{R}^{n\times k^2}$和 $W^* \in \mathbb{R}^{n\times k^2}$固定。我们将 V^*的列记为 V_1^*、V_2^*、$\cdots$、$V_{k^2}^* \in\mathbb{R}^n$，将 W^*的列记为 W_1^*、W_2^*、$\cdots$、$W_{k^2}^* \in \mathbb{R}^n$。

所考虑的优化问题如下:

$$\min_{U\in\mathbb{R}^{n\times k^2}} \left\| \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k} U_{i+k(j-1)} \otimes V^*_{j+k(l-1)} \otimes W^*_{l+k(i-1)} - A \right\|_F^2,$$

这可以改写为

$$\min_{U\in\mathbb{R}^{n\times k^2}} \left\| U \cdot \begin{bmatrix} \sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot 0} \\ \sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot 0} \\ \vdots \\ \sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot 0} \\ \sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot 1} \\ \sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot 1} \\ \vdots \\ \sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot 1} \\ \sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot (k-1)} \\ \sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot (k-1)} \\ \vdots \\ \sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot (k-1)} \end{bmatrix} - A \right\|_F^2.$$

我们可以表示 将张量沿第 $A_1 \in \mathbb{R}^{n\times n^2}$ 维展开为矩阵 $A \in \mathbb{R}^{n\times n\times n}$ e

一维。矩阵 $Z_1 \in \mathbb{R}^{k^2 \times n^2}$ 定义如下：

$$\begin{bmatrix} \mathrm{vec}(\sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot 0}) \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot 0}) \\ \vdots \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot 0}) \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot 1}) \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot 1}) \\ \vdots \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot 1}) \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{1+k(l-1)} \otimes W^*_{l+k\cdot (k-1)}) \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{2+k(l-1)} \otimes W^*_{l+k\cdot (k-1)}) \\ \vdots \\ \mathrm{vec}(\sum_{l=1}^{k} V^*_{k+k(l-1)} \otimes W^*_{l+k\cdot (k-1)}) \end{bmatrix}$$

现在，我们得到如下等价的目标函数，

$$\min_{U \in \mathbb{R}^{n \times k^2}} \|UZ_1 - A_1\|_F^2.$$

利用 $A_k = U^* Z_1$，我们可以得到 $\min_{U \in \mathbb{R}^{n \times k^2}} \|UZ_1 - A_1\|_F^2 = \mathrm{OPT}$。

利用草图矩阵 $S_1^\top \in \mathbb{R}^{s_1 \times n^2}$（定义见 4.12），其中 $s_1 = O(k^2/\epsilon)$:

$$\min_{U \in \mathbb{R}^{n \times k^2}} \|UZ_1 S_1 - A_1 S_1\|_F^2.$$

上述优化问题的最优解记为 $\widehat{U} \in \mathbb{R}^{n \times k^2}$。

于是，我们有

$$\widehat{U} = A_1 S_1 (Z_1 S_1)^\dagger.$$

由引理4.13 和定理 4.14，我们有

$$\begin{aligned} \|\widehat{U} Z_1 - A_1\|_F^2 &\leq (1+\epsilon) \cdot \min_{U \in \mathbb{R}^{n \times k^2}} \|UZ_1 - A_1\|_F^2 \\ &= (1+\epsilon) \cdot \mathrm{OPT}, \end{aligned}$$

这意味着

$$\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} \widehat{U}_{i+k(j-1)} \otimes V^*_{j+k(l-1)} \otimes W^*_{l+k(i-1)} - A \right\|_F^2 \leq (1+\epsilon) \cdot \mathrm{OPT}\,.$$

为了写出 $\widehat{U}_1$、$\cdots$、$\widehat{U}_{k^2} \in \mathbb{R}^n$，考虑到矩阵 A_1 已给定，我们为矩阵 $(Z_1 S_1)^\dagger$创建 $s_1 \times k^2$ 个变量。

现在进入第二步，固定两个矩阵 $\widehat{U} \in \mathbb{R}^{n \times k^2}$ 和 $W^* \in \mathbb{R}^{n \times k^2}$，我们将张量 A 转换为矩阵 A_2。

我们用 $Z_2 \in \mathbb{R}^{k^2 \times n^2}$ 表示这样的矩阵：对于每个 $j \in [k]$ 和 $l \in [k]$，第 (j,l) 行为

$$\mathrm{vec}(\sum_{i=1}^{k} \widehat{U}_{i+k(j-1)} \otimes W^*_{l+k(i-1)}).$$

现在我们来关注目标函数：

$$\min_{V \in \mathbb{R}^{n \times k^2}} \|V Z_2 - A_2\|_F^2,$$

其中，$(1+\epsilon) \cdot \mathrm{OPT}$ 是其最优代价的上界。

鉴于 $s_2 = O(k^2/\epsilon)$，我们使用 $S_2^\top \in \mathbb{R}^{s_2 \times n^2}$ 作为定义 4.12 中定义的草图矩阵。4.12。我们将 S_2 应用于目标函数的右侧，得到如下新的目标函数：

$$\min_{V \in \mathbb{R}^{n \times k^2}} \|V Z_2 S_2 - A_2 S_2\|_F^2.$$

我们使用 $\widehat{V} \in \mathbb{R}^{n \times k^2}$ 作为上述最优解。

现在，我们有

$$\widehat{V} = \underbrace{A_2}_{n \times n^2} \underbrace{S_2}_{n^2 \times s_2} \underbrace{(Z_2 S_2)^\dagger}_{s_2 \times k^2}.$$

根据引理4.13 和定理 4.14，我们有：

$$\begin{aligned} \|\widehat{V} Z_2 - A_2\|_F^2 &\leq (1+\epsilon) \cdot \min_{V \in \mathbb{R}^{n \times k}} \|V Z_2 - A_2\|_F^2 \\ &\leq (1+\epsilon)^2 \cdot \mathrm{OPT}, \end{aligned}$$

这意味着

$$\left\| \sum_{i=1}^{k} \sum_{j=1}^{k} \sum_{l=1}^{k} \widehat{U}_{i+k(j-1)} \otimes \widehat{V}_{j+k(l-1)} \otimes W^*_{l+k(i-1)} - A \right\|_F^2 \leq (1+\epsilon)^2 \, \mathrm{OPT}\,.$$

为了获得列 $\widehat{V}_1$、$\cdots$、$\widehat{V}_{k^2} \in \mathbb{R}^n$，回想一下，矩阵 $A_2 \in \mathbb{R}^{n^2 \times n}$ 已给定，我们为矩阵 $(Z_2 S_2)^\dagger$ 创建 $s_2 \times k^2$ 个变量。

我们在第三步将矩阵 $\widehat{U} \in \mathbb{R}^{n \times k^2}$ 和 $\widehat{V} \in \mathbb{R}^{n \times k^2}$ 设为固定值。张量$A \in \mathbb{R}^{n \times n \times n}$ 被转换为矩阵 $A_3 \in \mathbb{R}^{n^2 \times n}$。令矩阵 $Z_3 \in \mathbb{R}^{k^2 \times n^2}$ 表示

$$\begin{bmatrix} \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{1+k(j-1)} \otimes \widehat{V}_{j+k\cdot 0}) \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{2+k(j-1)} \otimes \widehat{V}_{j+k\cdot 0}) \\ \cdots \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{k+k(j-1)} \otimes \widehat{V}_{j+k\cdot 0}) \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{1+k(j-1)} \otimes \widehat{V}_{j+k\cdot 1}) \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{2+k(j-1)} \otimes \widehat{V}_{j+k\cdot 1}) \\ \cdots \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{k+k(j-1)} \otimes \widehat{V}_{j+k\cdot 1}) \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{1+k(j-1)} \otimes \widehat{V}_{j+k\cdot (k-1)}) \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{2+k(j-1)} \otimes \widehat{V}_{j+k\cdot (k-1)}) \\ \cdots \\ \sum_{j=1}^{k} \mathrm{vec}(\widehat{U}_{k+k(j-1)} \otimes \widehat{V}_{j+k\cdot (k-1)}) \end{bmatrix}.$$

我们考虑如下目标函数，

$$\min_{W\in\mathbb{R}^{n\times k^2}} \|WZ_3 - A_3\|_F^2,$$

代价至多为 $(1+\epsilon)^2$OPT 最优代价。

我们可以使用草图矩阵 $S_3^\top \in \mathbb{R}^{s_3\times n^2}$，其定义见定义 4.12，草图大小为 $s_3 = O(k^2/\varepsilon)$，以缩减输入的大小。将 S_3 草图应用于目标函数右侧，可得

$$\min_{W\in\mathbb{R}^{n\times k^2}} \|WZ_3S_3 - A_3S_3\|_F^2.$$

鉴于将 $\widehat{W} \in \mathbb{R}^{n\times k^2}$ 用作最优解，我们可以得到 $\widehat{W} = A_3S_3(Z_3S_3)^\dagger$。

根据引理4.13 和定理 4.14，我们有：

$$\begin{aligned}\|\widehat{W}Z_3 - A_3\|_F^2 &\le (1+\epsilon)\min_{W\in\mathbb{R}^{n\times k^2}} \|WZ_3 - A_3\|_F^2 \\ &\le (1+\epsilon)^3\,\mathrm{OPT}\,.\end{aligned}$$

因此，我们有

$$\min_{X_1,X_2,X_3} \left\| \sum_{i=1}^{k}\sum_{j=1}^{k}\sum_{l=1}^{k} (A_1S_1X_1)_{i+k(j-1)} \otimes (A_2S_2X_2)_{j+k(l-1)} \otimes (A_3S_3X_3)_{l+k(i-1)} - A \right\|_F^2$$
$$\le (1+\epsilon)^3\,\mathrm{OPT}\,.$$

设 $\widehat{V}_1 = A_1S_1$，$\widehat{V}_2 = A_2S_2$，并且 $\widehat{V}_3 = A_3S_3$. 然后，我们应用引理 4.16，得到 V_1，V_2，V_3，B。我们利用定理 4.17 高效地解决该问题。通过将 ϵ 按常数因子缩放，我们可以确保算法的正确性。

Ti复杂度。算法的运行时间可分为如下几部分 s,

- 根据定义 4.12，第 8 行（算法 3 ）需要的时间为

$$O(\mathrm{nnz}(A)) + n\,\mathrm{poly}(k, 1/\epsilon);$$

- 根据引理4.16，第 11 行和第 14 行需要的时间为

$$\mathrm{nnz}(A) + n\,\mathrm{poly}(k, 1/\epsilon);$$

- 根据定理 4.17，第 19 行需要的时间为

$$\exp(O(k^6/\epsilon)),$$

为简洁起见，忽略位复杂度。

将它们相加，总运行时间为

$$O(\mathrm{nnz}(A)) + n\,\mathrm{poly}(k, 1/\epsilon) + \exp(O(k^6/\epsilon)).$$

因此，我们完成了证明。 □

参考文献

[ANW14] Haim Avron、Huy Nguyen 和 David Woodruff。多项式核的子空间嵌入。《神经信息处理系统进展》，27，2014。[AS23] Josh Alman 和 Zhao Song。快速注意力需要有界项。arXiv 预印本 arXiv:2302.13214，2023。[ASA+22] Ekin Aky¨urek、Dale Schuurmans、Jacob Andreas、Tengyu Ma 和 Denny Zhou。上下文学习中使用的是什么学习算法？基于线性模型的研究。arXiv 预印本 arXiv:2211.15661，2022。[Bak79] James K Baker。用于语音识别的可训练语法。《美国声学学会杂志》，65(S1):S132–S132，1979。[BDS19] Jill Burstein、Christy Doran 和 Thamar Solorio。2019年计算语言学协会北美分会会议论文集：人类语言技术，第1卷（长篇与短篇论文）。载于《2019年计算语言学协会北美分会会议论文集：人类语言技术，第1卷（长篇与短篇论文）》，2019。[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者。《神经信息处理系统进展》，33:1877–1901，2020。[BNS19] Jan van den Brand、Danupon Nanongkai 和 Thatchaphol Saranurak。动态矩阵求逆：改进的算法与匹配的条件下界。载于2019年IEEE第60届计算机科学基础研讨会（FOCS），第456–480页。IEEE，2019。[BPR96] Saugata Basu、Richard Pollack 和 Marie-Fran¸coise Roy。论量词消去的组合复杂度与代数复杂度。《ACM学报》（JACM），43(6):1002–1045，1996。[BPSW21] Jan van den Brand、Binghui Peng、Zhao Song 和 Omri Weinstein。以近线性时间训练（过参数化）神经网络。载于 ITCS，2021。[Bra20] Jan van den Brand。一个运行时间达到当前矩阵乘法时间的确定性线性规划求解器。载于第十四届ACM-SIAM离散算法研讨会（SODA）论文集，第259–278页。SIAM，2020。[BSZ23] Jan van den Brand、Zhao Song 和 Tianyi Zhou。大语言模型中动态注意力维护的算法与困难性。arXiv 预印本 arXiv:2304.02207，2023。[BW14] Christos Boutsidis 和 David P Woodruff。最优CUR矩阵分解。载于第四十六届ACM计算理论研讨会（STOC）论文集，第353–362页，2014。[BWZ16] Christos Boutsidis、David P Woodruff 和 Peilin Zhong。分布式与流式模型中的最优主成分分析。载于第四十八届ACM计算理论研讨会论文集，第236–249页，2016。

[CCLY19] Michael B Cohen、Ben Cousins、Yin Tat Lee 和 Xin Yang。近似约翰椭球的近最优算法。在《学习理论会议》，第849–873页。PMLR，2019年。[Cha96] Eugene Charniak。树库语法。在《全国人工智能会议论文集》，第1031–1036页，1996年。[Chu89] Kenneth Ward Church。用于无限制文本的随机词性程序和名词短语分析器。在《国际声学、语音与信号处理会议》，第695–698页。IEEE，1989年。[CLS19] Michael B Cohen、Yin Tat Lee 和 Zhao Song。在当前矩阵乘法时间内求解线性规划。在 STOC，2019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：利用路径扩展语言建模。arXiv预印本 arXiv:2204.02311，2022年。[CW13] Kenneth L Clarkson 和 David P Woodruff。在输入稀疏度时间内进行低秩近似与回归。载于《ACM学报》（JACM），本文的初步版本发表于 STOC，2013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。[DJS+19] Huaian Diao、Rajesh Jayaram、Zhao Song、Wen Sun 和 David Woodruff。克罗内克积回归与低秩近似的最优草图。神经信息处理系统进展，32，2019年。[DMS23] Yichuan Deng、Sridhar Mahadevan 和 Zhao Song。面向过参数化特征维度的随机化与确定性注意力稀疏化算法。arXiv预印本：arXiv 2304.03426，2023年。[DSSW18] Huaian Diao、Zhao Song、Wen Sun 和 David Woodruff。克罗内克积回归与 P 样条的草图。在《人工智能与统计国际会议》，第1299–1308页。PMLR，2018年。[DSW22] Yichuan Deng、Zhao Song 和 Omri Weinstein。在输入稀疏度时间内进行差异最小化。arXiv预印本 arXiv:2210.12468，2022年。[FBS+21] Angela Fan、Shruti Bhosale、Holger Schwenk、Zhiyi Ma、Ahmed El-Kishky、Siddharth Goyal、Mandeep Baines、Onur Celebi、Guillaume Wenzek、Vishrav Chaudhary 等。超越以英语为中心的多语言机器翻译。《机器学习研究期刊》，22(1):4839–4886，2021年。[GMS23] Yeqi Gao、Sridhar Mahadevan 和 Zhao Song。过参数化指数回归。arXiv预印本 arXiv:2303.16504，2023年。[Goo96] Joshua Goodman。句法分析算法与度量。arXiv预印本 cmp-lg/9605036，1996年。

[GTLV22] Shivam Garg、Dimitris Tsipras、Percy Liang 和 Gregory Valiant。变换器能在上下文中学到什么？简单函数类的案例研究。arXiv预印本 arXiv:2208.01066，2022年。[JLSW20] Haotian Jiang、Yin Tat Lee、Zhao Song 和 Sam Chiu-wai Wong。用于凸优化、凸凹博弈及其应用的改进切平面法。在 STOC，2020年。[JLSZ23] Haotian Jiang、Yin Tat Lee、Zhao Song 和 Lichen Zhang。在 $\widetilde{O}(n^4)$ 时间内进行具有整数极小值的凸最小化。arXiv预印本，2023年。[KGW+23] John Kirchenbauer、Jonas Geiping、Yuxin Wen、Jonathan Katz、Ian Miers 和 Tom Goldstein。大语言模型的水印。arXiv预印本 arXiv:2301.10226，2023年。[KM03] Dan Klein 和 Christopher D Manning。准确的非词汇化句法分析。在《计算语言学协会第41届年会论文集》，第423–430页，2003年。[LGG+20] Yinhan Liu、Jiatao Gu、Naman Goyal、Xian Li、Sergey Edunov、Marjan Ghazvininejad、Mike Lewis 和 Luke Zettlemoyer。用于神经机器翻译的多语言去噪预训练。《计算语言学协会汇刊》，8:726–742，2020年。[LK98] Irene Langkilde 和 Kevin Knight。利用基于语料库的统计知识进行生成。在《COLING 1998 第1卷：第17届计算语言学国际会议》，1998年。[LLR23] Yuchen Li、Yuanzhi Li 和 Andrej Risteski。Transformer 如何学习主题结构：迈向机制理解。arXiv预印本 arXiv:2303.04245，2023 年。[LSZ23] Zhihang Li 、 Zhao Song 和 Tianyi Zhou。求解正则化 exp、cosh 和 sinh 回归问题。arXiv预印本，2303.15725，2023年。[LYF+23] Pengfei Liu、Weizhe Yuan、Jinlan Fu、Zhengbao Jiang、Hiroaki Hayashi 和 Graham Neubig。预训练、提示与预测：自然语言处理提示方法的系统综述。《ACM计算调查》，55(9):1–35，2023年。[Mit22] Ruslan Mitkov。《牛津计算语言学手册》。牛津大学出版社，2022年。[MS99] Christopher Manning 和 Hinrich Schutze。《统计自然语言处理基础》。麻省理工学院出版社，1999年。[MSM93] Mitchell Marcus、Beatrice Santorini 和 Mary Ann Marcinkiewicz。构建大型英语标注语料库：宾州树库。1993年。[NN13] Jelani Nelson 和 Huy L Nguyˆen。OSNAP：通过更稀疏的子空间嵌入实现更快的数值线性代数算法。在《2013年IEEE第54届计算机科学基础年度研讨会》，第117–126页。IEEE，2013年。[ONR+22] Johannes von Oswald、Eyvind Niklasson、Ettore Randazzo、Joˆao Sacramento、Alexander Mordvintsev、Andrey Zhmoginov 和 Max Vladymyrov。Transformer 通过梯度下降进行上下文学习。arXiv预印本 arXiv:2212.07677，2022年。

[Pag13] Rasmus Pagh。《压缩矩阵乘法》。ACM《计算理论汇刊》（TOCT），5(3):1–17，2013 年。[PK07] Slav Petrov 和 Dan Klein。《改进非词汇化句法分析的推理》。载于《2007 年人类语言技术会议：计算语言学协会北美分会会议；主会议论文集》，第 404–411 页，2007 年。[PP13] Ninh Pham 和 Rasmus Pagh。《通过显式特征映射实现快速且可扩展的多项式核》。载于《第 19 届 ACM SIGKDD 国际知识发现与数据挖掘会议论文集》，第 239–247 页，2013 年。[QSX+20] 邱锡鹏、孙天祥、徐一格、邵云帆、戴宁和黄萱菁。《自然语言处理预训练模型：综述》。《中国科学：技术科学》，63(10):1872–1897，2020 年。[Rat96] Adwait Ratnaparkhi。《用于词性标注的最大熵模型》。载于《自然语言处理经验方法会议》，1996 年。[Ren92] James Renegar。《关于实数一阶理论的计算复杂性与几何。第一部分：引言、预备知识、半代数集的几何以及实数存在理论的判定问题》。《符号计算杂志》，13(3):255–299，1992 年。[RPSW19] Sebastian Ruder、Matthew E Peters、Swabha Swayamdipta 和 Thomas Wolf。《自然语言处理中的迁移学习》。载于《2019年计算语言学协会北美分会会议：教程》，第 15–18 页，2019 年。[RSW16] Ilya Razenshteyn、Zhao Song 和 David P Woodruff。《具有可证明保证的加权低秩近似》。载于《第四十八届 ACM 计算理论研讨会论文集》，第 250–263 页，2016 年。[RSZ22] Aravind Reddy、Zhao Song 和 Lichen Zhang。《动态张量积回归》。载于 NeurIPS，2022 年。[Sar06] Tamas Sarlos。《通过随机投影改进大矩阵的近似算法》。载于《2006 年第 47 届 IEEE计算机科学基础研讨会（FOCS'06）》，第 143–152 页。IEEE，2006 年。[Sta20] Felix Stahlberg。《神经机器翻译：综述》。《人工智能研究杂志》，69:343–418，2020 年。[SWYZ21] Zhao Song、David Woodruff、Zheng Yu 和 Lichen Zhang。《多项式次数多项式核的快速草图化》。载于《国际机器学习会议》，第 9812–9823 页。PMLR，2021 年。[SWZ17] Zhao Song、David P Woodruff 和 Peilin Zhong。《具有逐元素 l1 范数误差的低秩近似》。载于《第 49 届 ACM SIGACT计算理论研讨会论文集》，第 688–701 页，2017 年。

[SWZ19a] Zhao Song、David Woodruff 和 Peilin Zhong。《逐元素 ℓ_1 范数损失下的平均情形列子集选择》。《神经信息处理系统进展》，32，2019 年。[SWZ19b] Zhao Song、David Woodruff 和 Peilin Zhong。《迈向列子集选择的零一律》。《神经信息处理系统进展》，32，2019 年。[SWZ19c] Zhao Song、David P Woodruff 和 Peilin Zhong。《相对误差张量低秩近似》。载于《第三十届 ACM-SIAM离散算法研讨会论文集》，第 2772–2789 页。SIAM，2019 年。[SYYZ22] Zhao Song、Xin Yang、Yuanyuan Yang 和 Tianyi Zhou。《结构化约翰椭球计算的更快算法》。arXiv 预印本，arXiv:2211.14407，2022 年。[SZKS21] Charlie Snell、Ruiqi Zhong、Dan Klein 和 Jacob Steinhardt。《近似单头注意力的学习方式》。arXiv 预印本，arXiv:2103.07601，2021 年。[SZZ21] Zhao Song、Lichen Zhang 和 Ruizhe Zhang。《在次二次时间内训练多层过参数化神经网络》。arXiv 预印本，arXiv:2112.07628，2021 年。[TKMS03] Kristina Toutanova、Dan Klein、Christopher D Manning 和 Yoram Singer。《具有循环依赖网络的特征丰富词性标注》。载于《2003 年人类语言技术会议：计算语言学协会北美分会会议论文集》，第 252–259 页，2003 年。[VSP+17] Ashish Vaswani、Noam Shazeer、Niki Parmar、Jakob Uszkoreit、Llion Jones、Aidan N Gomez、Lukasz Kaiser 和 Illia Polosukhin。《注意力就是你所需要的一切》。《神经信息处理系统进展》，30，2017 年。[WDS+19] Thomas Wolf、Lysandre Debut、Victor Sanh、Julien Chaumond、Clement Delangue、Anthony Moi、Pierric Cistac、Tim Rault、R´emi Louf、Morgan Funtowicz 等。《Huggingface 的 Transformer：最先进的自然语言处理》。arXiv 预印本，arXiv:1910.03771，2019 年。[Woo14] David P Woodruff。《作为数值线性代数工具的草图》。《计算机科学理论基础与趋势》 ®，10(1–2):1–157，2014 年。[WZ16] David P Woodruff 和 Peilin Zhong。《矩阵隐式函数的分布式低秩近似》。载于《2016 IEEE 第 32 届国际数据工程会议（ICDE）》，第 847–858 页。IEEE，2016 年。[Yng60] Victor H Yngve。《语言结构的一个模型和一个假说》。《美国哲学学会会刊》，104(5):444–466，1960 年。[Zha22] Lichen Zhang。《通过数据结构加速优化：更快的搜索、采样与维护》。硕士论文，卡内基梅隆大学，2022 年。[ZHDK23] Amir Zandieh、Insu Han、Majid Daliri 和 Amin Karbasi。《Kdeformer：通过核密度估计加速 Transformer》。arXiv 预印本，arXiv:2302.02451，2023 年。[ZKV+20] Jingzhao Zhang、Sai Praneeth Karimireddy、Andreas Veit、Seungyeon Kim、Sashank Reddi、Sanjiv Kumar 和 Suvrit Sra。《为什么自适应方法适用于注意力模型？》《神经信息处理系统进展》，33:15383–15393，2020 年。

[ZPGA23] Haoyu Zhao、Abhishek Panigrahi、Rong Ge 和 Sanjeev Arora。Transformer 在预测掩码词时会进行解析吗？载于 arXiv。https://arxiv.org/pdf/2303.08117.pdf，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。