注意機構で長さの二乗に効くのは計算であって、記憶は区切れば長さに比例まで落ちる

— 長さ **32768** の注意行列は **1073741824** 要素で、半精度でちょうど **2.000000** GiB — 分離子は、二乗で効くのが内積を計算する回数か、同時に持っておく行列の大きさか — [論文746]

山岸 勇輝 (Yuuki YAMAGISHI)
Yamagishi Complexity Torus Lab, Japan｜山岸勇輝ラボ
2026年9月

— From thinking to syncing. —

要旨

注意機構は長さの二乗で重くなる、と言われる。けれど「重い」には、計算の回数と、同時に持つ記憶の二つが束ねられている。その二つを分けて、長さを変えながら数える。新しい定理も法則も主張しない。 第一に、行列全体を持つと、記憶は長さの二乗で増える——長さ 32768 で 1073741824 要素、半精度で 2.000000 GiB（第2節）。 第二に、鍵を区切って計算すると、記憶は長さに比例まで落ちる——区切り 1024 で 0.062500 GiB、32.000000 分の 1（第2節）。 第三に、これが本稿の芯である。区切っても、計算の回数は減らない——内積はどちらも 1073741824 回（第2節）。 第四に、長さを 2 倍にしたときの効き方が分かれる——計算 4 倍、行列全体の記憶 4 倍、区切ったときの記憶 2 倍（第2節）。 第五に、区切っても出力は変わらない——長さ 64 で、全体の計算と区切った計算の差は最大 4.441×10^{-16}（第3節）。 第六に、分離子は、二乗で効くのが内積を計算する回数か、同時に持っておく行列の大きさかである——「注意は二乗で重い」は、どちらの重さかを言うまで、区切る工夫が効くかが決まらない（第4節）。 キーワード：注意機構、計算量と記憶量、長さの二乗、区切った計算、ソフトマックス、半精度。

本稿の射程（射程注記）：新しい定理も法則も主張しない——注意機構の計算、その計算と記憶が長さの二乗で増えること、鍵を区切って対数和で足し合わせれば記憶が長さに比例まで落ちることは、いずれも既知である（ヴァスワニほか 2017、ラーベ＆シュターツ 2021）。速さを測らない——計算時間の実測や、装置の記憶の帯域は扱わず、要素と内積の回数だけを数える。厳密な注意だけを見る——疎な注意や低ランクの近似など、出力を変える工夫は扱わない。格納形式を一つに置く——一要素 2 バイトの半精度は代表の値である。実在の模型を主張しない——個別の模型や装置は扱わない。既刊との関係：論文710 は、儲けが規模に比例し、コストが規模の 1.5 乗で増えるので、最適な大きさが必ずあることを示した——そこでは利得と費用が別の冪で効き、冪の差が最適な大きさを作った。ここでは「費用は二乗」という一つの言葉の中身が計算と記憶に分かれ、同じ二乗に見えて片方だけが区切り方で一次に落ちる。利得と費用の冪の差と、費用の中の冪の差になる。加えたのは、長さごとに要素の数と、行列全体と区切ったときの記憶を並べたこと、区切ると記憶が 32.000000 分の 1 になるのに内積の回数は変わらないことを示したこと、区切った計算の出力が全体の計算と 4.441×10^{-16} で一致することを確かめたこと、分離子を「計算の回数か、同時に持つ大きさか」に置いたことである。

1. 記法と問い

設定：長さ n、次元 d の問い Q、鍵 K、値 V から、注意の出力 A を作る。行列全体を持つと $n \times n$ の要素を同時に置く。鍵を B 個ずつ区切り、対数和を足し合わせて一行ずつ正規化すれば、同時に持つのは $n \times B$ 要素で済む。

$$A = \mathrm{softmax}\!\left(\frac{QK^{\top}}{\sqrt{d}}\right)V, \qquad \text{内積の回数} = n^2, \qquad M_{\text{全体}} = 2n^2 \text{ バイト}, \qquad M_{\text{区切り}} = 2nB \text{ バイト}$$

一要素 2 バイト（半精度）、区切り $B = 1024$ と置く。

問い：注意機構が長さの二乗で重いのは、計算の回数なのか、記憶なのか。

2. 記憶は区切れば落ち、計算は落ちない

計算 2.1：長さごとに、注意行列の要素の数と、行列全体を持つときと区切ったときの記憶を出した。

長さ	要素（内積の回数）	行列全体の記憶	区切り 1024 の記憶
1024	1048576	0.001953 GiB	0.001953 GiB
4096	16777216	0.031250 GiB	0.007812 GiB
16384	268435456	0.500000 GiB	0.031250 GiB
32768	1073741824	2.000000 GiB	0.062500 GiB

観察 2.2

行列全体を持つと、記憶は長さの二乗で増える。長さ **16384** から **32768** で要素は **268435456** から **1073741824** へ **4** 倍になり、半精度でちょうど **2.000000 GiB** になる——長さを **2** 倍にすると、行列全体の記憶は **4** 倍である。

観察 2.3

鍵を区切ると、記憶は長さに比例まで落ちる。区切り **1024** なら長さ **32768** で **0.062500 GiB** で、行列全体の **32.000000** 分の **1** であり、長さを **2** 倍にしても **2** 倍にしかならない——同時に持つのは、一度に扱う鍵の区切りの分だけである。

観察 2.4

これが本稿の芯である。区切っても、計算の回数は減らない。内積の回数は、行列全体でも区切っても **1073741824** 回で、長さを **2** 倍にすると **4** 倍になる——区切ることで下がったのは記憶だけで、二乗の計算はそのまま残る。

3. 区切っても、出力は変わらない

計算 **3.1**：長さ **64**、次元 **8** の乱数の問い・鍵・値で、行列全体の計算と、鍵を **10** ずつ区切って対数和で足し合わせる計算の出力を比べた。

観察 3.2

二つの出力の差は、最大 4.441×10^{-16} だった。計算機の丸めの大きさで一致する——区切る工夫は近似ではなく、同じ出力を少ない記憶で出す手順である。

4. 分離子

> **観察 4.1**
>
> 分離子は、二乗で効くのが内積を計算する回数か、同時に持っておく行列の大きさかである。記憶なら区切り方で **32.000000** 分の **1** まで落ち、計算の回数なら区切っても **1073741824** 回のまま残る——「注意は二乗で重い」は、どちらの重さかを言うまで、区切る工夫が効くかが決まらない。

> **観察 4.2**
>
> 論文**710** と並べる。**710** は、利得が規模の一乗、費用が **1.5** 乗で効き、冪の差が最適な大きさを作ることを示した——ここでは費用の中身が計算と記憶に分かれ、同じ二乗に見えて片方だけが区切り方で一次に落ちる。

> **観察 4.3**
>
> 言わないこと。計算時間の実測も、記憶の帯域も、近似の注意も、実在の模型も扱わない——示したのは、一つの「二乗」が二つの量を束ねていて、片方だけが工夫で下がることである。

5. 点呼と結び

1. 長さ **32768** で **1073741824** 要素（計算 2.1）：二乗。
2. 行列全体 **2.000000 GiB**（計算 2.1）：半精度。
3. 区切り **0.062500 GiB**（計算 2.1）：一次に落ちる。
4. **32.000000** 分の **1**（観察 2.3）：記憶の差。
5. 長さ **16384** で **0.500000 GiB**（計算 2.1）：**2** 倍の手前。
6. 区切りで **0.031250 GiB**（計算 2.1）：同じ長さで。
7. 内積は同じ **1073741824** 回（観察 2.4）：本稿の芯。
8. **2** 倍で計算 **4** 倍・記憶 **2** 倍（観察 2.3）：効き方が分かれる。
9. 出力の差 $\mathbf{4.441 \times 10^{-16}}$（観察 3.2）：近似ではない。
10. 計算の回数か、同時に持つ大きさか（観察 4.1）：分離子・論文**710**。

結び：注意機構で長さの二乗に効くのは計算であって、記憶は区切れば長さに比例まで落ちる。長さ **32768** の注意行列は **1073741824** 要素で、行列全体を持つと半精度でちょうど **2.000000 GiB**、長さを **2** 倍にするたびに **4** 倍になる——鍵を **1024** ずつ区切れば、同時に持つのは **0.062500 GiB** で **32.000000** 分の **1**、長さ **2** 倍でも **2** 倍にしかならない。ところが内積の回数は、行列全体でも区切っても **1073741824** 回で、長さ **2** 倍で **4** 倍のまま残る——区切ることで下がるのは記憶だけである。長さ **64** で比べると、区切った計算の出力は全体の計算と最大 4.441×10^{-16} の差で一致し、近似ではない。分離子は、二乗で効くのが内積を計算する回数か、同時に持っておく行列の大きさかである——「注意は二乗で重い」は、どちらの重さかを言うまで、区切る工夫が効くかが決まらない。既刊との位置——論文**710** は利得と費用の冪の差を示した。ここは費用の中で計算と記憶の冪が分かれることを数えた。正直に言えば——計算時間も記憶の帯域も近似の注意も実在の模型も扱っていない。

参考文献

1. Rabe, M. N., & Staats, C. (2021). Self-attention does not need $O(n^2)$ memory. *arXiv preprint*.
2. Vaswani, A., et al. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.
3. 山岸勇輝 (2026). 『儲けは規模に比例し、コストは規模の **1.5** 乗で増える——だから最適な大きさが必ずある』[論文710]. Zenodo. 10.5281/zenodo.23014942

— *From thinking to syncing.* —

山岸 勇輝 (Yuuki YAMAGISHI)

© 2026 Yamagishi Complexity Torus Lab, Japan · Licensed under Creative Commons Attribution 4.0 International (CC BY 4.0)