Transformer モデルのコンテキスト拡張手法

平田 蓮

2024 年 7 月 10 日

まずはじめに、LLaMA などで用いられている主流な位置エンコーディング手法である RoPE の説明を行う。その後、シーケンス拡張を行う為に提案された手法を取り上げる。

1 RoPE

Jianlin ら [1] によって提唱された、位置エンコーディングを加算ではなく乗算で行うアプローチ。LLaMA などで用いられており、現在主流である。Query と Key に回転行列をかけて、ベクトルを回転させることで位置情報を乗せる。トークンの位置ごとに徐々に回転角を増していくため、相対位置情報も反映されていると考えられている。

回転行列は、次の式で与えられる。

$$R = \begin{pmatrix} \cos m\theta_1 & -\sin m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ \sin m\theta_1 & \cos m\theta_1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cos m\theta_2 & -\sin m\theta_2 & \cdots & 0 & 0 \\ 0 & 0 & \sin m\theta_2 & \cos m\theta_2 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cos m\theta_{\frac{D}{2}} & -\sin m\theta_{\frac{D}{2}} \\ 0 & 0 & 0 & 0 & \cdots & \sin m\theta_{\frac{D}{2}} & \cos m\theta_{\frac{D}{2}} \end{pmatrix} \tag{1}$$

$$\theta_i = 10000^{-\frac{2(i-1)}{D}} \quad \left(i = 1, 2, \cdots, \frac{D}{2}\right)$$

ここで、m はトークン位置、D は埋め込みの次元である。つまり、トークン位置が後ろに行くほど、回転角が大きくなる。この回転行列はベクトルの D 次元空間を 2 次元ずつ取って、その 2 軸が成す平面において回転させる。これを用いて計算した $Q' = R^tQ, K' = R^tK$ により Attention 行列を計算する。

2 シーケンスの拡張

2.1 Position Interpolation

RoPE のシンプルな拡張として、Shouyuan ら [2] と kaiokendev[3] によって提案された Position Interpolation が挙げられる。これは、RoPE のトークン位置を元のコンテキスト長と新しいコンテキスト長の比率を用いてスケールするものである。すなわち、式 1 において、$m \to \frac{L}{L'}m$ とする。ここで、L は元のコンテキスト長、L' は拡張したコンテキスト長である。今後、$\frac{L'}{L}$ をスケール倍率と呼ぶ。

この手法は、拡張したコンテキスト長で少数ショット学習を行うことでモデルの性能を落とさずにコンテキ

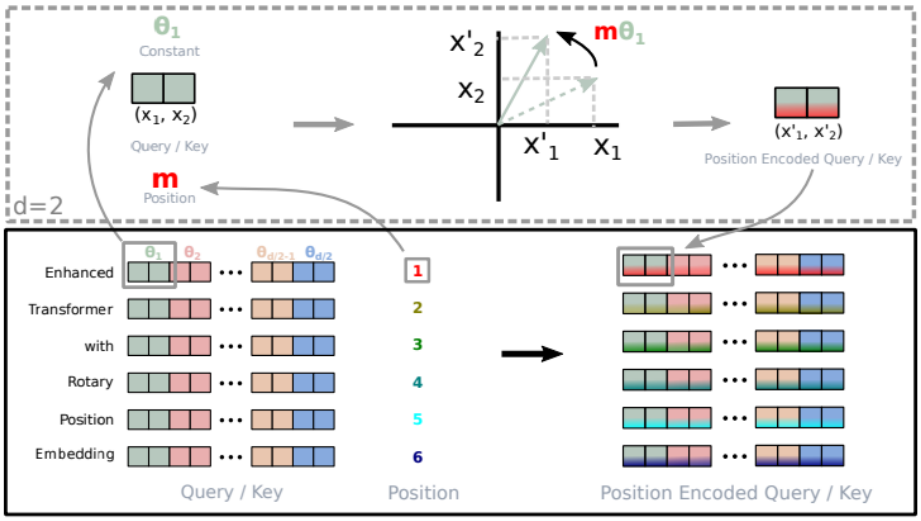

Figure 1: Implementation of Rotary Position Embedding(RoPE).

図 1　RoPE[1]

ストを拡張できることが示された [2]。しかし、一定より高いスケール倍率を用いると、性能が低下することも確認されている。

2.2　YaRN

RoPE における d 次元目の回転角を θ_d としたとき、RoPE における波長 $\lambda_d = \frac{2\pi}{\theta_d}$ を定義する。波長は、その次元に対する位置エンコーディングの回転が一周するのに必要なトークン数を表す。Bowen ら [4] は、この波長に着目し、RoPE のコンテキスト拡張を行った。

埋め込みの後ろの方の次元の一部の範囲において、学習したコンテキスト長よりも波長が長い次元が存在する。つまり、その次元においては位置エンコーディングが回転領域で均等に分布していない。このような場合は、Position Interpolation を行っても、位置エンコーディングの絶対位置の情報がそのまま残ってしまうことが危惧される。逆に、波長が短い次元においては、絶対位置の情報が消えてしまい、相対位置の情報しか残っていないと考えられる。

加えて、先述の通り Position Interpolation を高いスケール倍率で行うと性能が低下するが、これはスケール倍率を高くすればするほど距離が近いトークン同士の位置エンコーディングの差異が小さくなり、関係性をうまくモデルが認識できなくなっていることが原因であると考えられる。

以上の仮定をもとに、Position Interpolation に次のような変更を行った。

- λ_d が L に対して十分に小さい次元ではスケーリングを行わない
- λ_d が L 以上の次元では、$L' = \lambda_d$ としたスケーリングを行う
- 上二つの間の次元領域では、波長の長さに応じて回転角を調整する

これは、二つのハイパーパラメータ α, β を用いて以下の式で表される。

$$\gamma(r) = \begin{cases} 0 & (r < \alpha) \\ 1 & (r > \beta) \\ \dfrac{r-\alpha}{\beta-\alpha} & (\text{otherwise}) \end{cases}$$

$$h(\theta_d) = \left(1 - \gamma\left(\frac{L}{\lambda_d}\right)\right)\frac{L}{L'}\theta_d + \gamma\left(\frac{L}{\lambda_d}\right)\theta_d$$

r は波長に対する元のコンテキスト長の比であり、γ はこの比をもとに元の回転角とスケールした回転角それぞれを用いる比率を計算する。YaRN では Position Interpolation と違い、m ではなく $\theta'_d = h(\theta_d)$ として回転角に対してスケーリングを行うことに注意されたい。

2.2.1 実験結果

二つの実験を通して YaRN の性能が評価された。評価は Perplexity の比較で行われた。各モデルの Perplexity を表 2 に示す。表中にある PI は Position Interpolation を示し、NTK は、bloc97[5] によって提案された別の RoPE の拡張手法である。

図 2　YaRN の性能評価 [4]。s はスケーリング倍率

Extension Method	Trained Tokens	Context Window	Evaluation Context Window Size 2048	4096	6144	8192	10240
PI ($s = 2$)	1B	8k	3.92	3.51	**3.51**	**3.34**	8.07
NTK ($\theta = 20$k)	1B	8k	4.20	3.75	3.74	3.59	6.24
YaRN ($s = 2$)	400M	8k	**3.91**	**3.50**	**3.51**	3.35	**6.04**

Model Size	Model Name	Context Window	Extension Method	Evaluation Context Window Size 8192	32768	65536	98304	131072
7B	Together	32k	PI	**3.50**	**2.64**	$> 10^2$	$> 10^3$	$> 10^4$
7B	Code Llama	100k	NTK	3.71	2.74	2.55	2.54	2.71
7B	YaRN ($s = 16$)	64k	YaRN	3.51	2.65	**2.42**	$> 10^1$	$> 10^1$
7B	YaRN ($s = 32$)	128k	YaRN	3.56	2.70	2.45	**2.36**	**2.37**
13B	Code Llama	100k	NTK	3.54	2.63	2.41	2.37	2.54
13B	YaRN ($s = 16$)	64k	YaRN	**3.25**	**2.50**	**2.29**	$> 10^1$	$> 10^1$
13B	YaRN ($s = 32$)	128k	YaRN	3.29	2.53	2.31	**2.23**	**2.24**

上の表では、各手法に対して YaRN が少ない学習量でも優位であることが示された。下の表では、YaRN が高いスケーリング倍率でも有効であることが示された。

2.3 LongNet

LongNet は、Jiayu ら [6] によって提案されたモデルで、後述する Dialated Attention という構造を用いてコンテキスト長を 10 億まで拡張できることを示した。

2.3.1 Dialated Attention

Dialated Attention は、トークンの Attention を計算する際に、入力を複数の大きさのブロックに分割する。ブロックの大きさに応じて、Attention を計算するトークンの間隔を広げ、一定の大きさの入力としている。これらの結果を合算し、離れた位置にあるトークン同士の依存関係も捉えている。Dialated Attention の概念図を図 3 に示す。

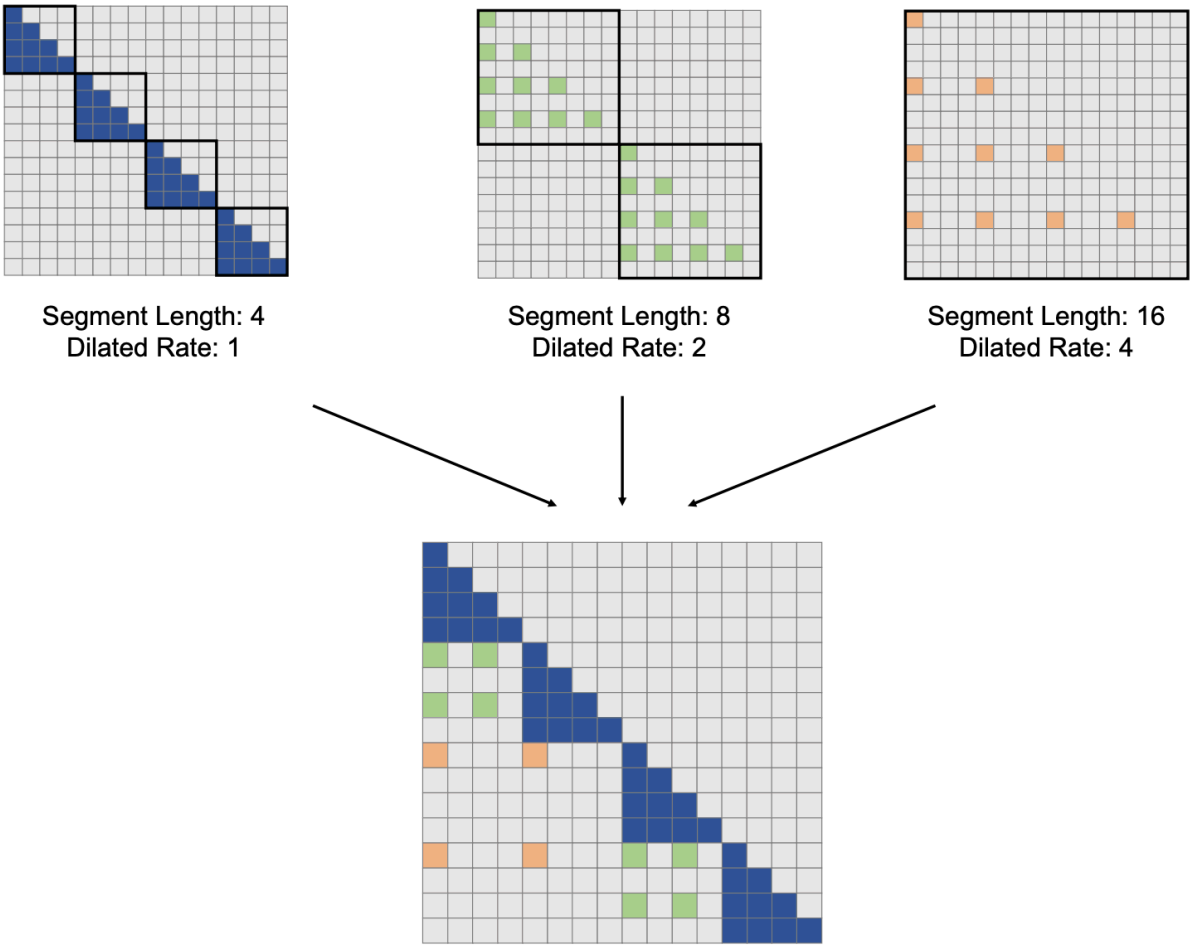

図 3　Dialated Attention[6]

さらに、各 Attention head において Attention を計算するトークンのパターンを変更する。これにより、すべてのトークン間の依存関係を捉えられる。

2.3.2 実験結果

LongNet では、Dialated Attention に加え、時間計算量を抑える為に、分散アルゴリズムを用いてトークン長と埋め込み次元に対する線形時間で学習が行われた。32k までのコンテキスト長でベースの Transformer モデルと同様の性能を発揮するこが確認され、さらにコンテキスト長を拡張しても性能が低下しないことが確認された。これにより、コンテキスト長を 10 億まで拡張できると結論づけられた。ベースモデルとの

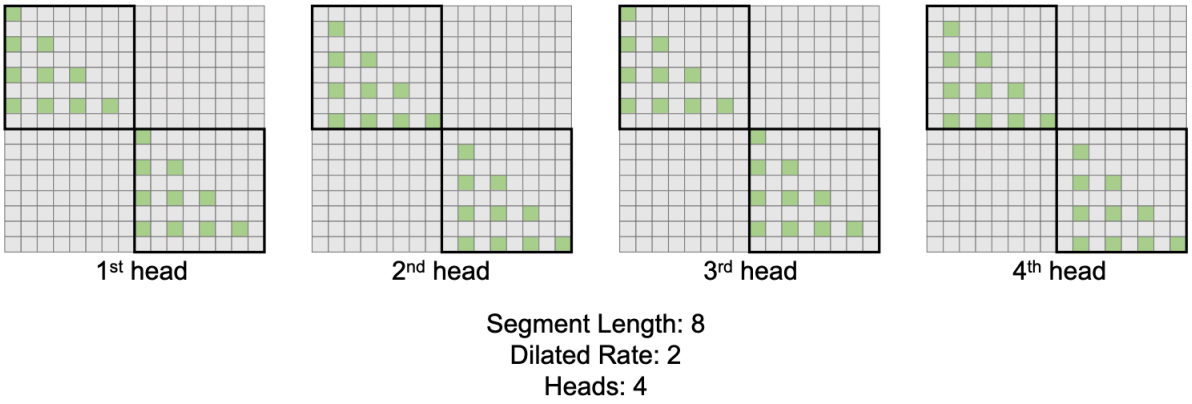

図 4　Attention head ごとのパターン [6]

Perplexity の比較を表 5 に示す。

図 5　LongNet の性能 [6]

Model	Length	Batch	Github 2K	Github 8K	Github 32K
Transformer [VSP+17]	2K	256	4.24	5.07	11.29
Sparse Transformer [CGRS19]	8K	64	4.39	3.35	8.79
LONGNET (ours)			4.23	3.24	3.36
Sparse Transformer [CGRS19]	16K	32	4.85	3.73	19.77
LONGNET (ours)			4.27	3.26	3.31
Sparse Transformer [CGRS19]	32K	16	5.15	4.00	3.64
LONGNET (ours)			4.37	3.33	3.01

2.4　LongLoRA

Yukang ら [7] によって提案された、コンテキスト長が大きい場合でも効率的に LoRA を適用する手法。この手法は、Position Interpolation などによってコンテキスト長を拡張したモデルに対してファインチューニングを行うことを仮定している。通常の LoRA ではコンテキスト長が大きくなるにつれて perplexity が大きくなってしまう。一方、通常のファインチューニングは低い perplexity を維持するが、計算コストと VRAM の消費量が膨大になってしまう。LongLoRA では、後述の Shifted Sparse Attention を用いて perplexity を通常のファインチューニングと同等に抑えつつ、VRAM 消費量も LoRA と同等、かつより小さな計算量でのファインチューニングを実現している。

2.4.1　Shifted Sparse Attention

まず、入力シーケンスを分割し、分割したブロックごとに Attention 機構に通す。これにより、計算量が改善された。さらに、各ブロックの大きさの半分だけずらして分割をした入力を同様に Attention 機構に通

す。これらの Attention を合算することで、計算量を抑えつつ、ブロック間の依存関係を捉えることができる。Shifted Sparse Attention の概念図を図 6 に示す。

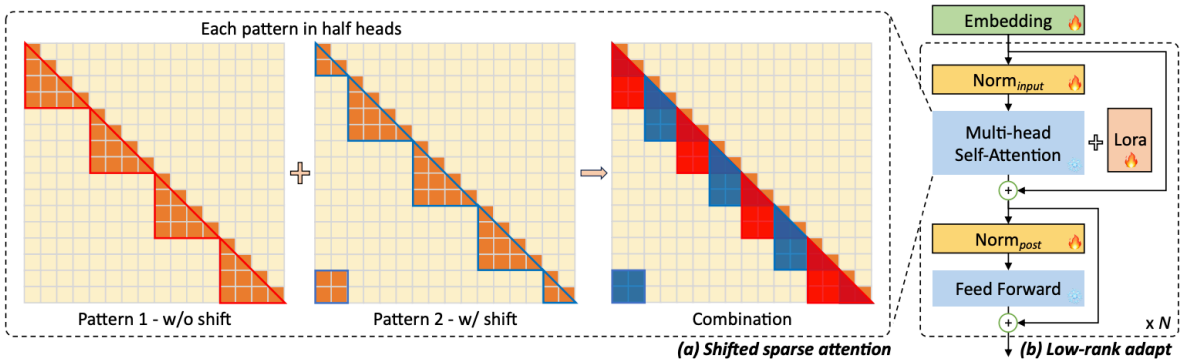

図 6　Shift Sparse Attention[7]

2.4.2　実験結果

Perplexity、VRAM 使用量、学習時間の三つの観点から LongLoRA の評価が行われた。コンテキスト長の増大に伴うそれぞれの値の変化を図 7 に示す。

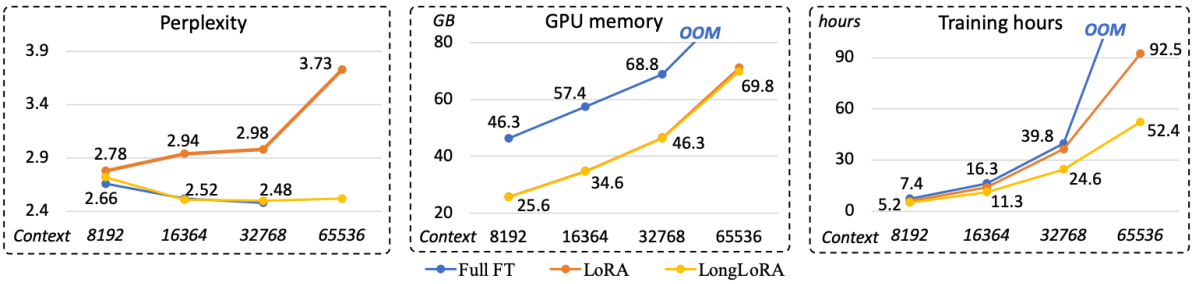

図 7　LongLoRA の性能 [7]

図より、LongLoRA は、通常のファインチューニングと同様の性能を維持しつつ、LoRA と同様に VRAM 使用量を抑え、かつ LoRA より少ない時間で学習できていることがわかる。

2.5　外挿によるシーケンス拡張

Yutao ら [8] は、transformer モデルの外挿（学習データより長いデータの入力）を可能にする手法を提案した。外挿能力の指標として Attention Resolution を定義している。

$$R(s) = \sum_{i=0}^{N} \frac{e^{s[i]}\left(e^{s[i]} - e^{s[i+1]}\right)}{\left(\sum_{i=0}^{N} e^{s[i]}\right)^2}$$

$s[i]$　距離が i の二つのトークンの Attention スコア（Attention 行列の要素）の期待値

$R(s)$　Attention Resolution

位置エンコーディングの工夫と Blockwise Causal Attention の二つの手法を用いて Attention Resolution を最大化し、モデルの外挿能力を強化している。

2.5.1 位置エンコーディングによる Attention Resolution の最大化

Attention Resolution を増加させる為に、RoPE を改良した位置エンコーディングを提案している。Query,Key それぞれに対する以下の位置エンコーディング f_q, f_k を考える。

$$f_q(\boldsymbol{q}, n) = \begin{pmatrix} q_1 \cos n\zeta_1^n\theta_1 - q_2 \sin n\zeta_1^n\theta_1 \\ q_2 \cos n\zeta_1^n\theta_1 + q_1 \sin n\zeta_1^n\theta_1 \\ \vdots \\ q_{D-1} \cos n\zeta_{\frac{D}{2}}^n\theta_{\frac{D}{2}} - q_D \sin n\zeta_{\frac{D}{2}}^n\theta_{\frac{D}{2}} \\ q_D \cos n\zeta_{\frac{D}{2}}^n\theta_{\frac{D}{2}} + q_{D-1} \sin n\zeta_{\frac{D}{2}}^n\theta_{\frac{D}{2}} \end{pmatrix}, f_k(\boldsymbol{k}, n) = \begin{pmatrix} k_1 \cos n\zeta_1^{-n}\theta_1 - k_2 \sin n\zeta_1^{-n}\theta_1 \\ k_2 \cos n\zeta_1^{-n}\theta_1 + k_1 \sin n\zeta_1^{-n}\theta_1 \\ \vdots \\ k_{D-1} \cos n\zeta_{\frac{D}{2}}^{-n}\theta_{\frac{D}{2}} - k_D \sin n\zeta_{\frac{D}{2}}^{-n}\theta_{\frac{D}{2}} \\ k_D \cos n\zeta_{\frac{D}{2}}^{-n}\theta_{\frac{D}{2}} + k_{D-1} \sin n\zeta_{\frac{D}{2}}^{-n}\theta_{\frac{D}{2}} \end{pmatrix}$$

$\boldsymbol{q}, \boldsymbol{k}$　あるトークンに対する Query と Key：$\boldsymbol{q} = {}^t(q_1, \cdots, q_D), \boldsymbol{k} = {}^t(k_1, \cdots, k_D)$
n　　トークンの位置

ζ_i はハイパーパラメータ γ を用いて

$$\zeta_i = \frac{\frac{i}{\frac{D}{2}} + \gamma}{1 + \gamma}$$

で定義される定数で、$[0, 1]$ の範囲を取る。ここで、$\zeta_i = 1$ とすると、これは RoPE と同様の定義になる。この位置エンコーディングを用いた際の Attention スコアの期待値を図 8 に示す。図を見ると、RoPE では発振が見られるが、提案手法ではそれが抑えられている。これにより、Attention Resolution が増大することが確認された。

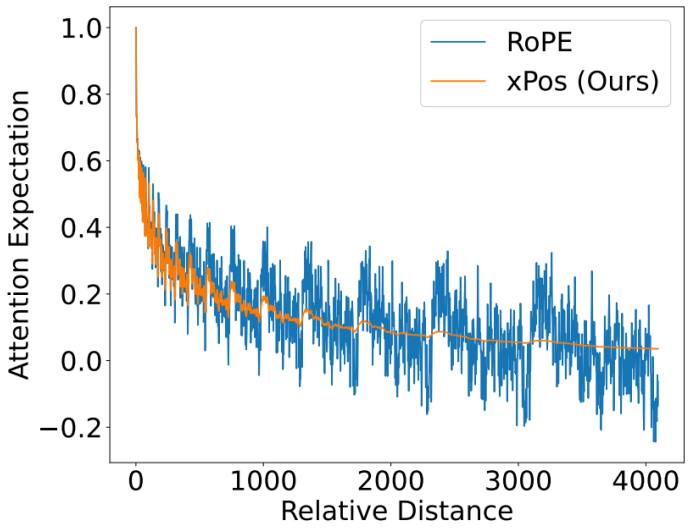

図 8　二つのトークンの相対距離に対する Attention スコアの期待値 [8]

2.5.2 Blockwise Causal Attention

出力シーケンスを入力シーケンス長の半分の長さのブロックに分割し、ブロックごとに順番に Attention 機構に通す（図 9）ことで、間接的に離れた距離のトークン同士の情報が Attention 行列に適用され、Attention Resolution が増大することが示された。

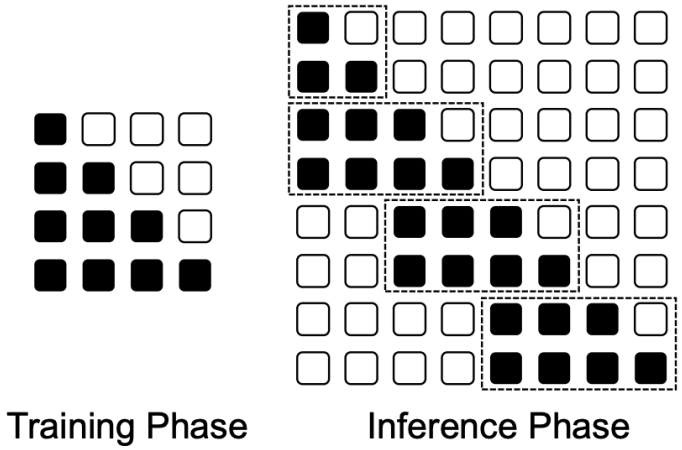

図 9　Blockwise Causal Attention[8]

2.5.3 実験結果

提案モデルはトークン長 1024 で学習し、2048 トークン、4096 トークンの入力における外挿能力が検証された。評価は Perplexity の比較で行われた。表 10 に各モデルの Perplexity を示す。

図 10　各モデルの Perplexity[8]

Length	**256**	**512**	**1024**	**2048**	**4096**
	Interpolation			Extrapolation	
Transformer	46.34	36.39	29.94	132.63	1283.79
Alibi	37.66	29.92	24.99	23.14	24.26
Roformer	38.09	30.38	25.52	73.6	294.45
LEX Transformer (Ours)	**34.3**	**27.55**	**23.31**	**21.6**	**20.73**

表より、提案手法は他のモデルより外挿能力に長けていることがわかった。

3　展望

以上では、シーケンスの拡張を行った先行研究を見てきた。再学習を行う・行わない手法それぞれどちらも挙げたが、LLM を扱う上で、モデルの再学習は時間・空間計算量の観点で大きな障壁となる。また、理想的な LLM は入出力どちらも外挿可能なアーキテクチャであると考える。今後、再学習を行わずに入出力どちらも外挿可能なモデルについて調査・検証を進めていきたい。

このような研究を行う上で、事前学習済みのパラメータを用いつつ内部構造に変更を加えて検証を行えるモデルが不可欠である。**現状、そのようなモデルが公開されているプラットフォーム等を発見できていない為、思い当たる方は是非平田に一報願いたい。**（huggingface にて公開されているモデル群はダメであった...）

参考文献

[1] Jianlin S., et al., “RoFormer: Enhanced Transformer with Rotary Position Embedding”, arXiv:2104.09864, 2021

[2] Shouyuan C., et al., “Extending Context Window of Large Language Models via Positional Interpolation”, arXiv:2306.15595, 2023

[3] kaiokendev, “Extending Context is Hard... but not Impossible”, ‘https://kaiokendev.github.io/context’, 最終閲覧日: 2024/7/8

[4] Bowen P., et al. “YaRN: Efficient Context Window Extension of Large Language Models”, International Conference on Learning Representations 2024, 2023

[5] bloc97, “NTK-Aware Scaled RoPE allows LLaMA models to have extended (8k+) context size without any fine-tuning and minimal perplexity degradation.”, ‘https://www.reddit.com/r/LocalLLaMA/comments/14lz7j5/ntkaware_scaled_rope_allows_llama_models_to_have’, 最終閲覧日: 2024/7/8

[6] Jiayu D., et al., “LongNet: Scaling Transformers to 1,000,000,000 Tokens”, arXiv:2307.02486, 2023

[7] Yukang C., et al., “LongLoRA: Efficient Fine-tuning of Long-Context Large Language Models”, International Conference on Learning Representations 2024, 2023

[8] Yutao S., et al., “A Length-Extrapolatable Transformer”, Association for Computational Linguistics 2023, 2022

[9] Liang Z., et al., “Length Extrapolation of Transformers: A Survey from the Perspective of Positional Encoding”, arXiv:2312.17044, 2024