Transformer モデルのコンテキスト拡張手法

情報数理システム分野 研究会

平田 蓮

2024 年 7 月 10 日

RoPE

- 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

RoPE の特徴

- トークンの位置ごとに回転角を増やしていくため、絶対位置情報を反映可能
- 距離の近いトークンの回転角同士が成す角が小さいため、相対位置情報も反映可能

回転行列は、トークンの位置 m と埋め込みの次元 D を用いて次のように表せる

RoPE

回転行列の計算

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

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

これを用いて計算した ${}^tQ' = R{}^tQ, {}^tK' = R{}^tK$ により Attention 行列を計算する。

入力シーケンスの拡張

入力シーケンス（コンテキスト）長を拡張する手法として以下について述べる

- Position Interpolation
- YaRN
- LongNet
- LongLoRA

Position Interpolation

- Shouyuan らと kaiokendev によって提案された RoPE のシンプルな拡張
- 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

RoPE の回転行列において、$m \to \frac{L}{L'}m$ とする。

L 元のコンテキスト長

L' 拡張したコンテキスト長

$\frac{L'}{L}$ をスケール倍率と呼ぶ。

- 少数ショット学習を行うことでコンテキストを拡張できることが示された
- 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

RoPE の波長

d 次元目における回転角 θ_d を用いて波長 λ_d を定義する

$$\lambda_d = \frac{2\pi}{\theta_d}$$

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

Bowen らは、この波長に着目し、RoPE のコンテキスト拡張を行った

YaRN

埋め込みの高次元部分において、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
→ 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
→ 埋め込みが一回転しないため、Position Interpolation を行なっても絶対位置の情報が保存される
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う

YaRN

さらに、Position Interpolation を高いスケール倍率で行うと性能が低下する
$\rightarrow$ スケール倍率を高く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

YaRN

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

YaRN

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

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

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

Position Interpolation と違い、$\theta'_d = h(\theta_d, \lambda_d)$ として、m ではなく回転角に対してスケーリングを行う

- r は波長に対する元のコンテキスト長の比
- γ はこの比をもとにスケーリングする比率を計算する

実験結果

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**

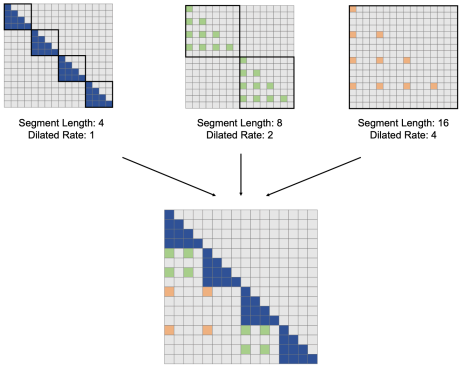
- 上の表では、各手法に対して YaRN が少ない学習量でも優位であることが示された
- 下の表では、YaRN が高いスケーリング倍率でも有効であることが示された

LongNet

- Jiayu らによって提案されたモデル
- 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

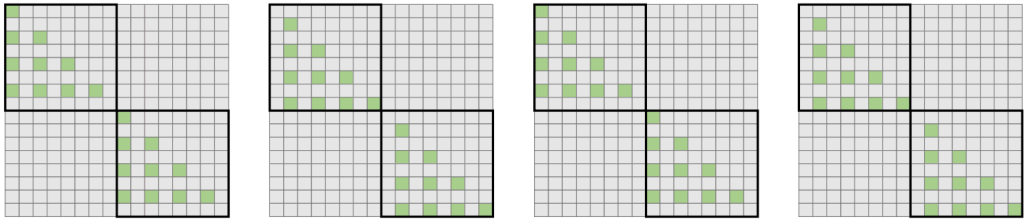
Dialated Attention

- 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

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
→ すべてのトークン間の依存関係を捉えられる



Segment Length: 8
Dilated Rate: 2
Heads: 4

実験結果

- 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とが確認された
- 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

以上より、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。

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

LongLoRA

- Yukang ら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
- 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

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

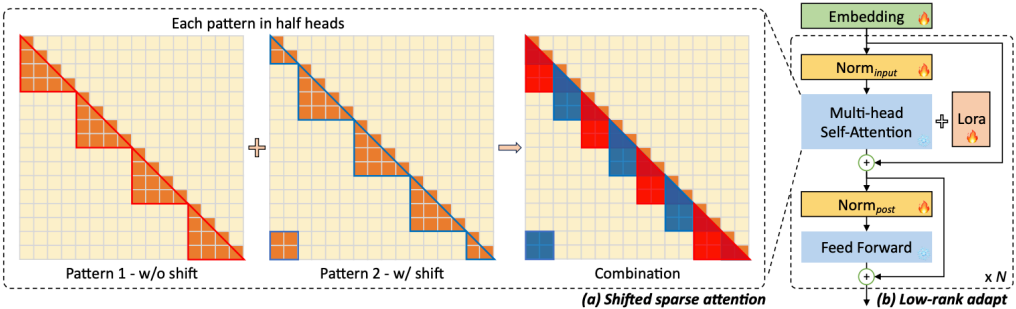
LongLoRA では、後述の Shifted Sparse Attention を用いて、

- 通常のファインチューニングと同等の perplexity
- LoRA と同等の VRAM 消費量
- 小さい計算量

でのファインチューニングを実現している

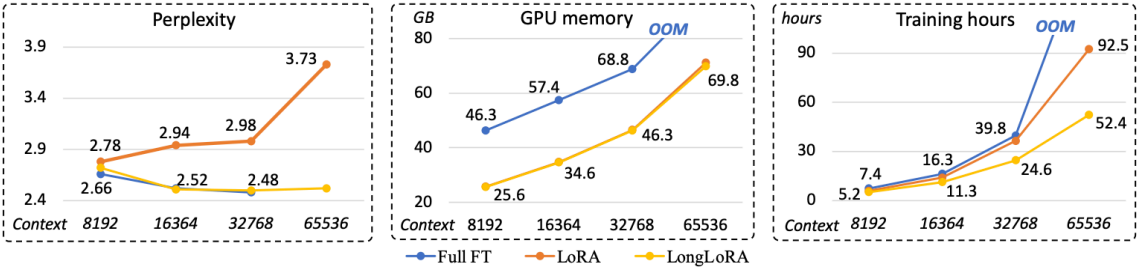
Shifted Sparse Attention

- 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
  → 計算量が改善
- 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



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



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

外挿によるコンテキスト拡張

Yutao らは外挿（学習データより長いシーケンスの入力）を可能にする手法を提案した

Attention Resolution

外挿能力の指標

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

$s[i]$ 距離が i のトークン間の Attention スコア（Attention 行列の要素）の期待値

後述の 2 手法を用いて Attention Resolution が増大された

位置エンコーディングによる Attention Resolution の増加

RoPE を改良した次の位置エンコーディング f_q, f_k を用いた

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

$\boldsymbol{q}, \boldsymbol{k}$ 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

位置エンコーディングによる Attention Resolution の増加

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。

位置エンコーディングによる Attention Resolution の増加

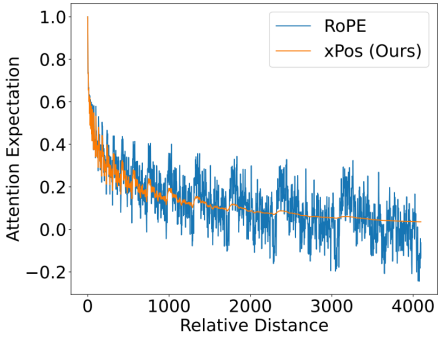
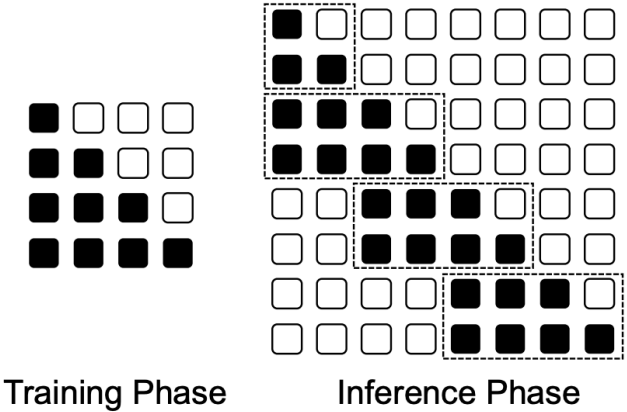


図: 二つのトークンの相対距離に対する Attention スコアの期待値

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

Blockwise Causal Attention

出力シーケンスをブロックに分割し、ブロックごとに順番に Attention 機構に通すことで、
間接的に離れた距離のトークン同士の情報が Attention 行列に適用されることが示された



実験結果

トークン長 1024 で学習したモデルの、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

図: 各モデルの Perplexity

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**

展望

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
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
→ 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