Magneto: A Foundation Transformer

Hongyu Wang[*1], Shuming Ma[*2], Shaohan Huang[2], Li Dong[2], Wenhui Wang[2], Zhiliang Peng[1], Yu Wu[2], Payal Bajaj[2], Saksham Singhal[2], Alon Benhaim[2], Barun Patra[2], Zhun Liu[2], Vishrav Chaudhary[2], Xia Song[2], Furu Wei[2]

[1] University of Chinese Academy of Sciences, [2] Microsoft

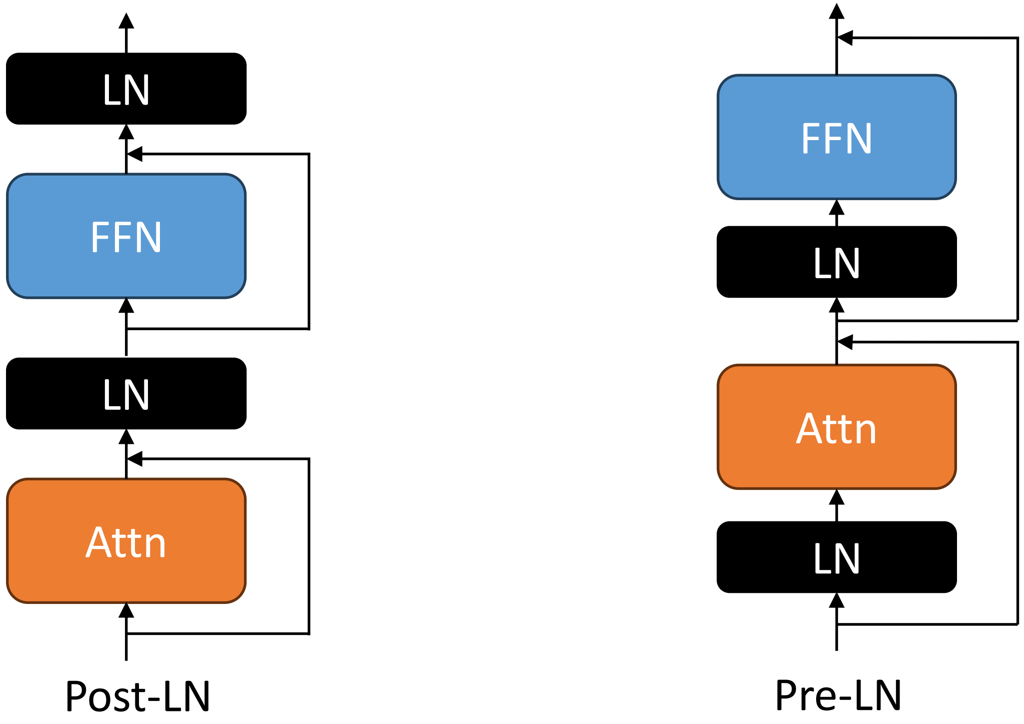
https://github.com/microsoft/torchscale

Paper Code

Presenter: Hongyu Wang

Introduction

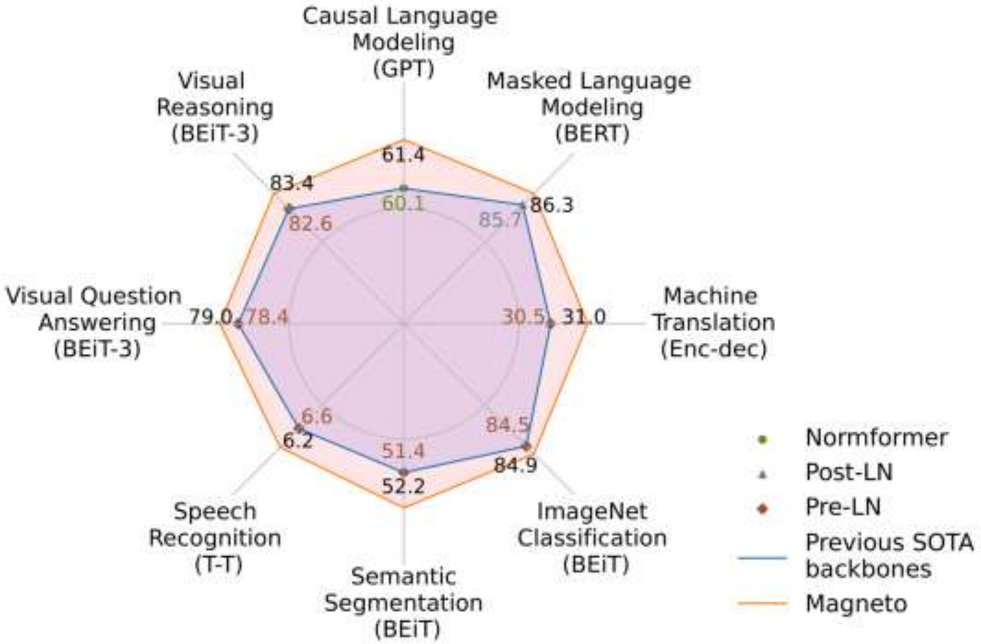
- **Problem:** Under the same name "Transformers", different areas use different implementations for better performance
  - Post-LayerNorm for BERT
  - Pre-LayerNorm for GPT and vision Transformers



Introduction

- **Problem:** Under the same name "Transformers", different areas use different implementations for better performance
  - Post-LayerNorm for BERT
  - Pre-LayerNorm for GPT and vision Transformers

- **Magneto:** A *Foundation Transformer* for *True general-purposed modeling*
  - Good expressivity: Sub-LayerNorm
  - Stable scaling up: The initialization strategy theoretically derived from DeepNet



TL;DR

- **Left:** pseudocode of Sub-LN. We take Xavier initialization as an example, and it can be replaced with other standard initialization. Notice that γ is a constant.
- **Right:** parameters of Sub-LN for different architectures (N-layer encoder, M-layer decoder).

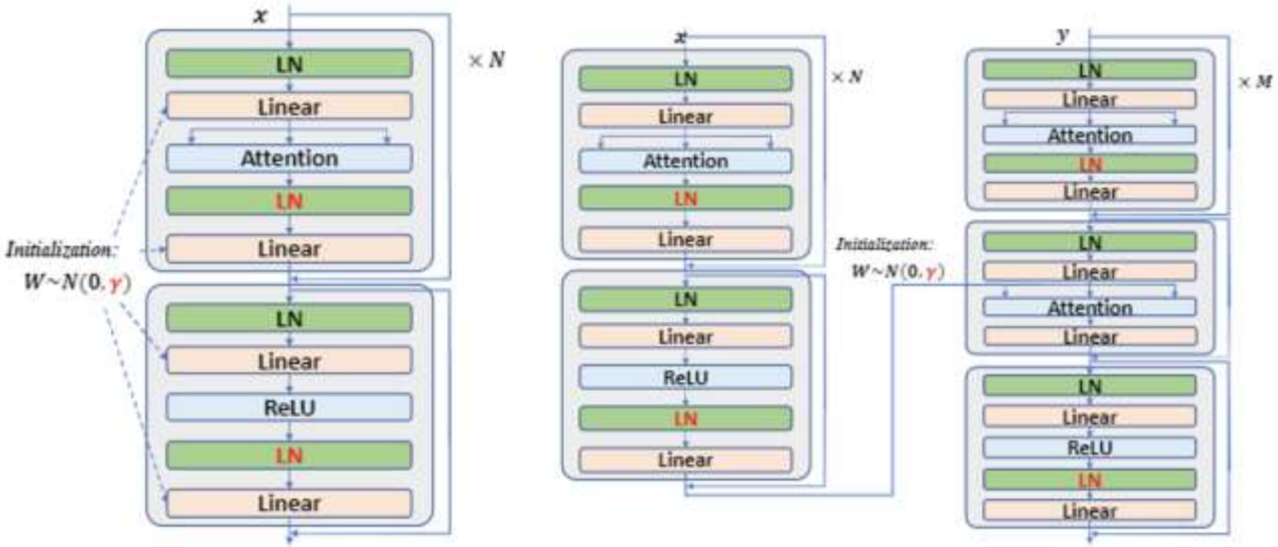
```
def subln(x):
    return x + fout(LN(fin(LN(x))))

def subln_init(w):
    if w is ['ffn', 'v_proj', 'out_proj']:
        nn.init.xavier_normal_(w, gain=γ)
    elif w is ['q_proj', 'k_proj']:
        nn.init.xavier_normal_(w, gain=1)
```

Architectures	**Encoder** γ	**Decoder** γ
Encoder-only (e.g., BERT, ViT)	$\sqrt{\log 2N}$	-
Decoder-only (e.g., GPT)	-	$\sqrt{\log 2M}$
Encoder-decoder (e.g., NMT, BART)	$\sqrt{\frac{1}{3}\log 3M \log 2N}$	$\sqrt{\log 3M}$

Magneto: Sub-LayerNorm

- Sub-LN has a lower bound of model update and does not suffer from activation explosion.
- The layout of Sub-LN for **(a)** encoder-decoder, **(b)** encoder or decoder architectures:



(a) Encoder or Decoder (b) Encoder-Decoder

Magneto: Initialization

Model Update: $\Delta F = \left\| \gamma^T \left(F(x, \theta^*) - F(x, \theta) \right) \right\|$

- x denotes the input of the model.
- γ denotes the label of x.
- $F(x, \theta)$ denotes the model's output given the parameters θ.
- $F(x, \theta^*)$ denotes the model's output given the updated parameters θ^*.

- ΔF denotes the smoothness of loss landscape.
    - Smaller ΔF leads to more stable optimization.

Magneto: Initialization

Theorem 1: Given an N-layer Pre-LN $F(x,\theta)$, the l-th sublayer is formulated as $x^l = x^{l-1} + W^{l,2}\phi\left(W^{l,1}LN(x^{l-1})\right)$. Under SGD update, ΔF^{pre} satisfies that:

$$\Delta F^{pre} \le \eta d\left(\frac{\sum_{l=1}^{L} v_l^2 + w_l^2}{\sum_{n=1}^{L} v_n^2 w_n^2} + \sum_{l=1}^{L}\sum_{k=2}^{L}\frac{v_l^2 + w_l^2}{\sum_{n=1}^{L} v_n^2 w_n^2}\frac{v_k^2 w_k^2}{\sum_{n=1}^{k-1} v_n^2 w_n^2}\right))$$

Theorem 2: Given an N-layer Magneto $F(x,\theta)$, the l-th sublayer is formulated as $x^l = x^{l-1} + W^{l,2}LN\left(W^{l,1}LN(x^{l-1})\right)$. Under SGD update, ΔF^{sub} satisfies that:

$$\Delta F^{sub} \le \eta d\left(\frac{\sum_{l=1}^{L}(1+\frac{v_l^2}{w_l^2})}{\sum_{n=1}^{L} v_n^2} + \sum_{l=1}^{L}\sum_{k=2}^{L}\frac{1+\frac{v_l^2}{w_l^2}}{\sum_{n=1}^{L} v_n^2}\frac{v_k^2}{\sum_{n=1}^{k-1} v_n^2}\right)$$

where η is the learning rate, d is the hidden dimension, $W_{ij}^{l,2} \sim N(0,\frac{v^2}{d})$ and $W_{ij}^{l,1} \sim N(0,\frac{w^2}{d})$

Magneto: Initialization

- When the activation of the l-th sublayer explodes: $w_l \gg \ w_i, \ i \neq l$

$$\frac{1+\frac{v_l^2}{w_l^2}}{\sum_{n=1}^{L} v_n^2} = \frac{v_l^2 + w_l^2}{w_l^2 \sum_{n=1}^{L} v_n^2} \leq \frac{v_l^2 + w_l^2}{\sum_{n=1}^{L} v_n^2 w_n^2}, \quad w_l \gg w_i, \, i \neq l$$

Therefore, Sub-LN has smaller model update than Pre-LN.

Normalization	The bound of model update	Activation explosion
Post-LayerNorm	$\Theta(N)$	×
Pre-LayerNorm	$\Theta(\log N)$	√
Sub-LayerNorm	$\Theta(\log N)$	×

Magneto: Initialization

- **GOAL**: $F(x,\theta)$ is updated by $\Theta(\eta)$ per SGD step after initialization as $\eta \to 0$. That is $\Delta F^{sub} = \Theta(\eta d)$ where $\Delta F^{sub} \triangleq F\left(x, \theta - \eta\frac{\delta L}{\delta\theta}\right) - F(x,\theta)$.

- **Derivation**: The term related to the model depth can be bounded as:

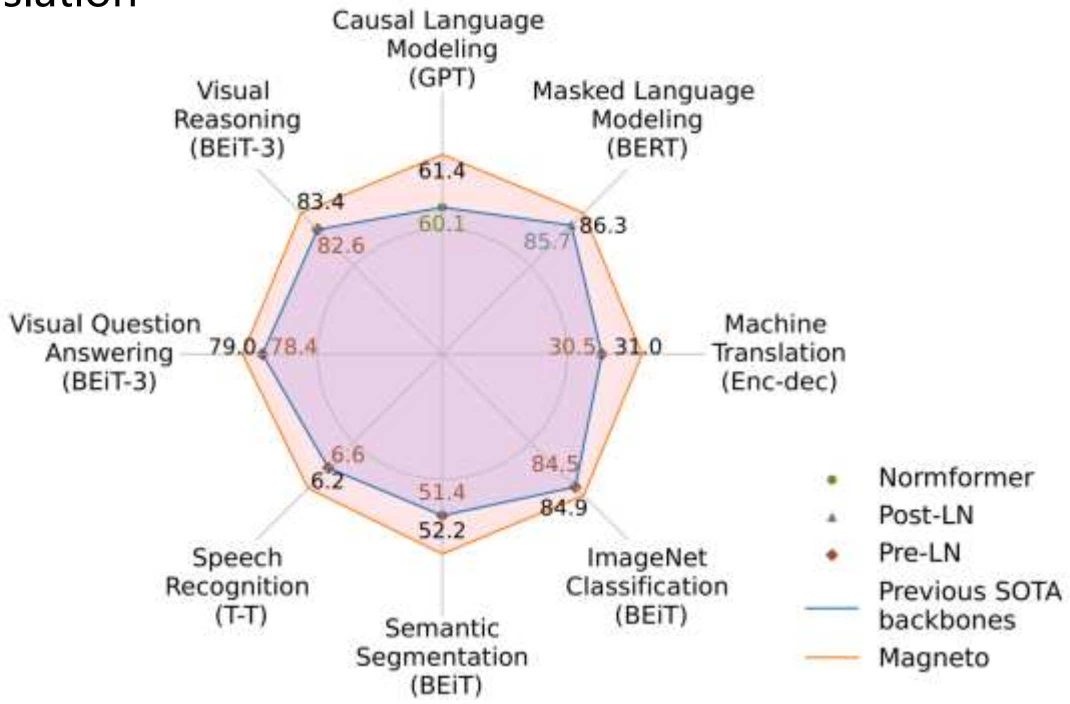
$$\frac{\sum_{l=1}^{L}(1+\frac{v_l^2}{w_l^2})}{\sum_{n=1}^{L} v_n^2} + \frac{1}{\sum_{n=1}^{L} v_n^2}\sum_{l=1}^{L}\sum_{k=2}^{L}(1+\frac{v_l^2}{w_l^2})\frac{v_k^2}{\sum_{n=1}^{k-1} v_n^2} = \mathcal{O}(\frac{\log L}{\gamma^2})$$

We use $v = w = \gamma = \sqrt{\log L}$ to bound the model update independent of depth.

Experiments

- Better performance across various tasks and modalities
  - Language modeling (BERT, GPT) and machine translation
  - Vision pre-training (BEiT)
  - Speech recognition
  - Multi-modal pre-training (BEiT-3)

- Stable scaling up
  - Tolerate higher learning rate



Experiments

Magneto is more stable and has better performance for language modeling (i.e., BERT, and GPT) and machine translation.

Models	# Layers	LR	WGe	WG	SC	HS	Avg.
Pre-LN	24L	5e-4	**55.2**	65.3	70.8	44.8	59.0
Pre-LN		1e-3			diverged		
Normformer		5e-4	54.3	68.1	72.0	45.9	60.1
Normformer		1e-3			diverged		
MAGNETO		1e-3	54.3	**71.9**	**72.4**	**46.9**	**61.4**
Pre-LN	48L	5e-4	**57.3**	67.0	74.0	48.0	61.6
Normformer		5e-4	56.5	70.5	74.0	49.8	62.7
MAGNETO		1.2e-3	57.0	**73.3**	**74.7**	**51.2**	**64.1**
Pre-LN	72L	5e-4	**58.0**	70.9	75.7	51.7	64.1
Normformer		5e-4	57.4	**75.4**	75.2	53.6	65.4
MAGNETO		1.2e-3	57.9	73.7	**76.6**	**55.1**	**65.8**

Causal language modeling: Zero-shot results for Magneto and the baselines.

Models	# Layers	LR	WGe	WG	SC	HS	Avg.
Pre-LN	24L	5e-4	54.0	67.7	69.8	44.6	59.0
Pre-LN		1e-3			diverged		
Normformer		5e-4	54.3	70.2	71.4	45.9	60.5
Normformer		1e-3			diverged		
MAGNETO		1e-3	**57.6**	**74.7**	**72.8**	**47.5**	**63.2**
Pre-LN	48L	5e-4	57.7	71.2	73.8	48.7	62.9
Normformer		5e-4	56.8	**75.4**	75.9	50.7	**64.7**
MAGNETO		1.2e-3	**57.9**	71.9	**76.4**	**51.9**	64.5
Pre-LN	72L	5e-4	57.5	73.3	76.1	52.4	64.8
Normformer		5e-4	57.7	**74.0**	77.0	54.9	65.9
MAGNETO		1.2e-3	**58.3**	**74.0**	**79.0**	**55.7**	**66.8**

Causal language modeling: Four-shot results for Magneto and the baselines.

Experiments

Magneto is more stable and has better performance for language modeling (i.e., BERT, and GPT) and machine translation.

Models	LR	MNLI	QNLI	QQP	SST	CoLA	MRPC	STS	Avg.
Post-LN	5e-4	**86.7/86.7**	92.2	91.0	93.4	59.8	86.4	**89.4**	85.7
Post-LN	1e-3				diverged				
Pre-LN	1e-3	85.6/85.4	92.2	91.1	93.4	55.6	85.1	88.4	84.6
Pre-LN	2e-3				diverged				
MAGNETO	3e-3	**86.7/86.7**	**92.4**	**91.2**	**93.9**	**62.9**	**87.2**	89.2	**86.3**

Masked language modeling: The results for Magneto and the baselines on GLUE benchmark.

Models	En → X	X → En	Avg.
Post-LN		diverged	
Pre-LN	28.3	32.7	30.5
NormFormer	28.5	32.3	30.4
MAGNETO	**28.7**	**33.2**	**31.0**

Machine translation : BLEU scores for Magneto and the baselines on OPUS-100 dataset.

Experiments

Magneto has better performance for vision pretraining, speech recognition and multi-modal pre-training.

Models	# Layers	ImageNet	ImageNet Adversarial	ImageNet Rendition	ImageNet Sketch	ADE20k
Pre-LN	12L	84.5	45.9	55.6	42.2	51.4
MAGNETO		**84.9**	**48.9**	**57.7**	**43.9**	**52.2**
Pre-LN	24L	86.2	60.1	63.2	48.5	54.2
MAGNETO		**86.8**	**65.4**	**67.5**	**52.0**	**54.6**

Vision pre-training: The results of Magneto and the baselines on vision tasks.

Models	# Layers	VQA test-dev	VQA test-std	NLVR2 dev	NLVR2 test-P
Pre-LN	24L	78.37	78.50	82.57	83.69
MAGNETO		**79.00**	**79.01**	**83.35**	**84.23**

Multi-modal pre-training: The results of Magneto and the baseline on vision-language tasks.

Models	# Layers	Dev-Clean	Dev-Other	Test-Clean	Test-Other
Pre-LN	18L	2.97	6.52	3.19	6.62
MAGNETO		**2.68**	**6.04**	**2.99**	**6.16**
Pre-LN	36L	2.59	6.10	2.89	6.04
MAGNETO		**2.43**	**5.34**	**2.72**	**5.56**

Speech recognition: The results of Magneto and the baselines on the LibriSpeech 960h.

Takeaways

Magneto is a go-to architecture for various tasks and modalities with guaranteed training stability.

Smaller model update leads to more stable optimization.

Thanks

Paper

Code