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

Introduction

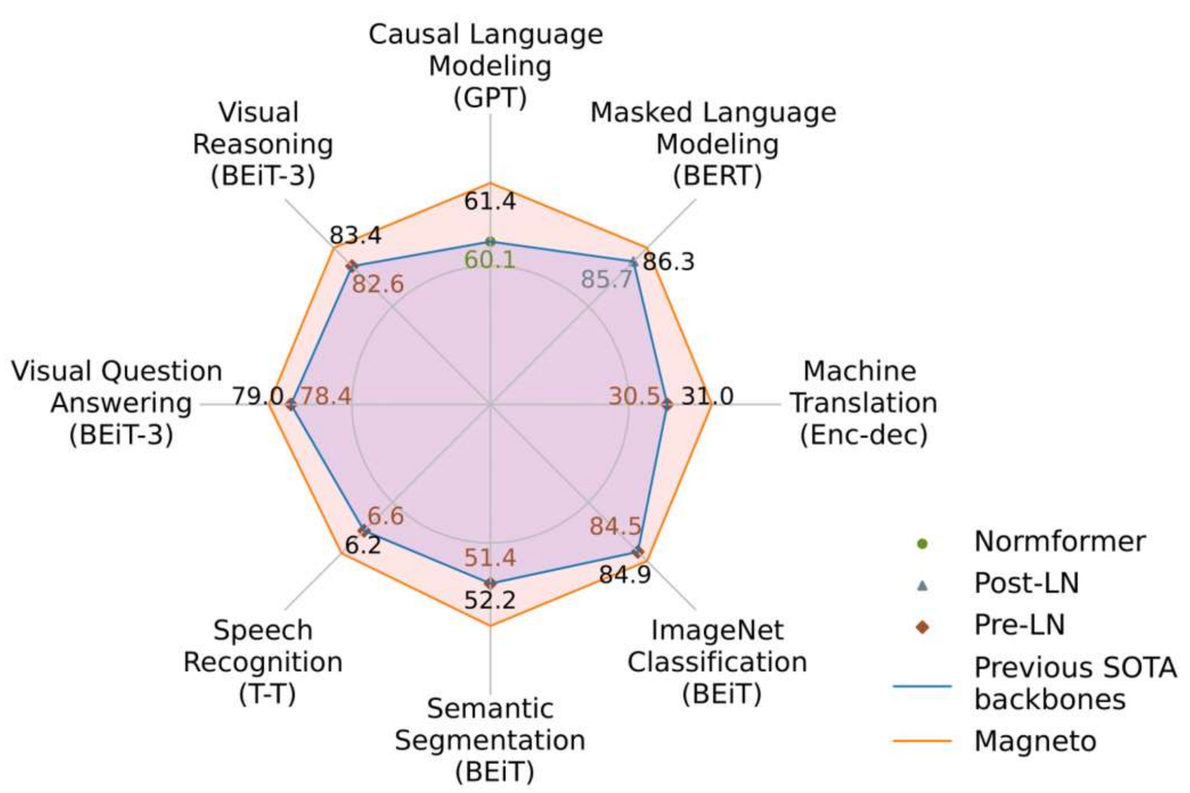

Figure 1: Magento performs better than the previous SOTA backbones across tasks and modalities with a unified architecture.

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

Our approach: Magneto, A Foundation Transformer for true general-purpose modeling

- Good expressivity: Sub-LayerNorm
- Stable scaling up: The initialization strategy theoretically derived from DeepNet

Magneto outperforms de facto Transformer variants designed for various applications, including

- language modeling (i.e., BERT, and GPT), machine translation
- vision pretraining (i.e., BEiT)
- speech recognition
- multimodal pretraining (i.e., BEiT-3)

Methods

```
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)	$\sqrt{\frac{1}{3}\log 3M \log 2N}$	$\sqrt{\log 3M}$

Figure 2: **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).

Architecture: Sub-LayerNorm

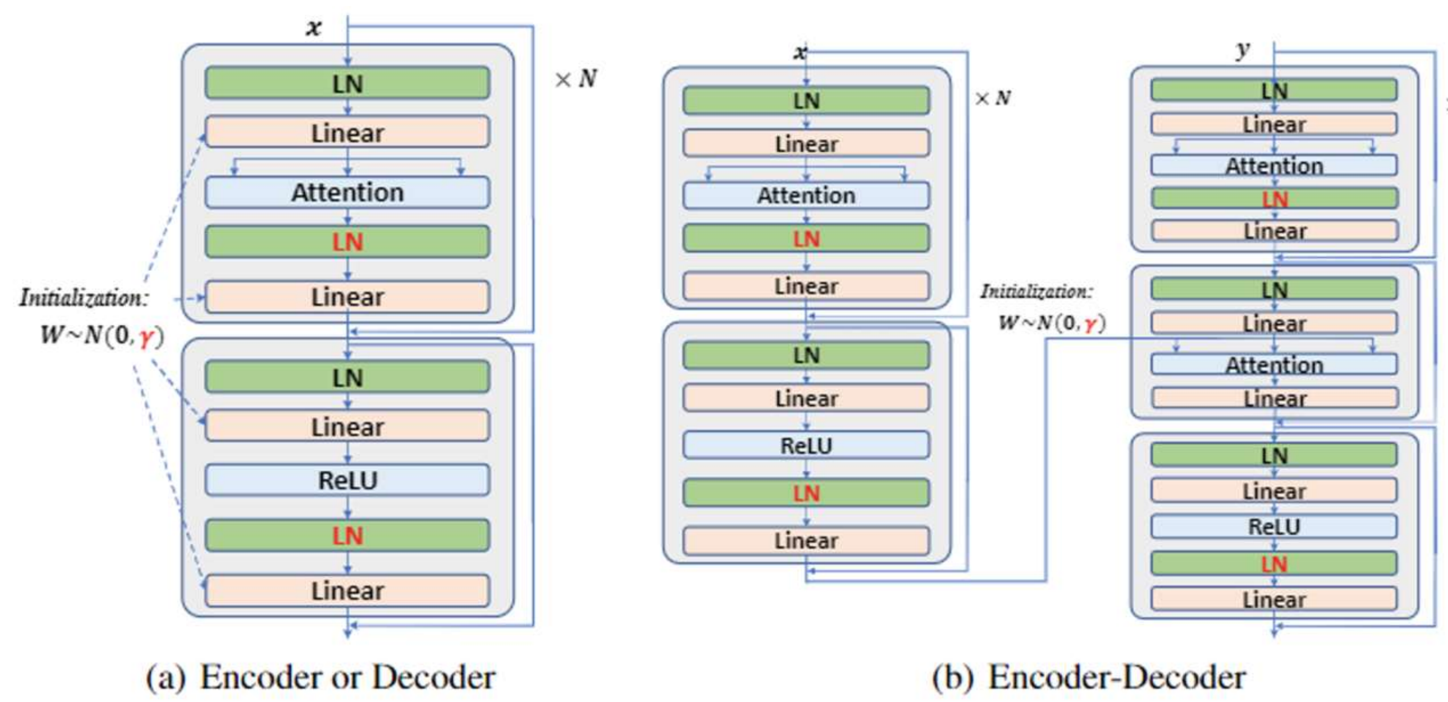

Figure 3: The layout of Sub-LN for (a) encoder-decoder, (b) encoder or decoder architectures.

Initialization: Theoretical Derivation from DeepNet

Model Update: $\Delta F = \|\gamma^T(F(x,\theta^*) - F(x,\theta))\|$ where (x,y), $F(x,\theta)$ denotes the training data and the model, respectively.

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

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

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

Language Tasks:

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

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**

Table 1: BLEU scores for Magneto and the baselines on the OPUS-100 dataset.

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**

Table 2: Zero-shot results for Magneto and the baselines (WGe: Winogrande, WG: Winograd, SC: Storycloze, and HS: Hellaswag dataset).

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**

Table 3: Results for Magneto and the baselines on the GLUE benchmark.

Vision Tasks:

Magneto outperforms vanilla ViT on vision 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**

Table 4: Results for Magneto and the baselines on the vision tasks.

Speech Tasks:

Magneto outperforms Pre-LN on speech recognition.

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**

Table 5: Results for Magneto and the baselines on speech recognition.

Vision-Language Tasks:

Magneto has better performance than Pre-LN on multi-modal pre-training.

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

Table 6: Results for Magneto and the baselines on vision-language tasks.

Paper **Code**

ICML International Conference On Machine Learning