GPAS: Gradient-Preserving Activation Scaling for LLM Pretraining

Tianhao Chen, Xin Xu, Zijing Liu, Pengxiang Li, Xinyuan Song, Ajay Kumar Jaiswal, Fan Zhang,
Jishan Hu, Yang Wang, Hao Chen, Shizhe Diao, Shiwei Liu, Yu Li, Lu Yin, Can Yang

NEURAL INFORMATION PROCESSING SYSTEMS

Background: activation growth in Pre-LN

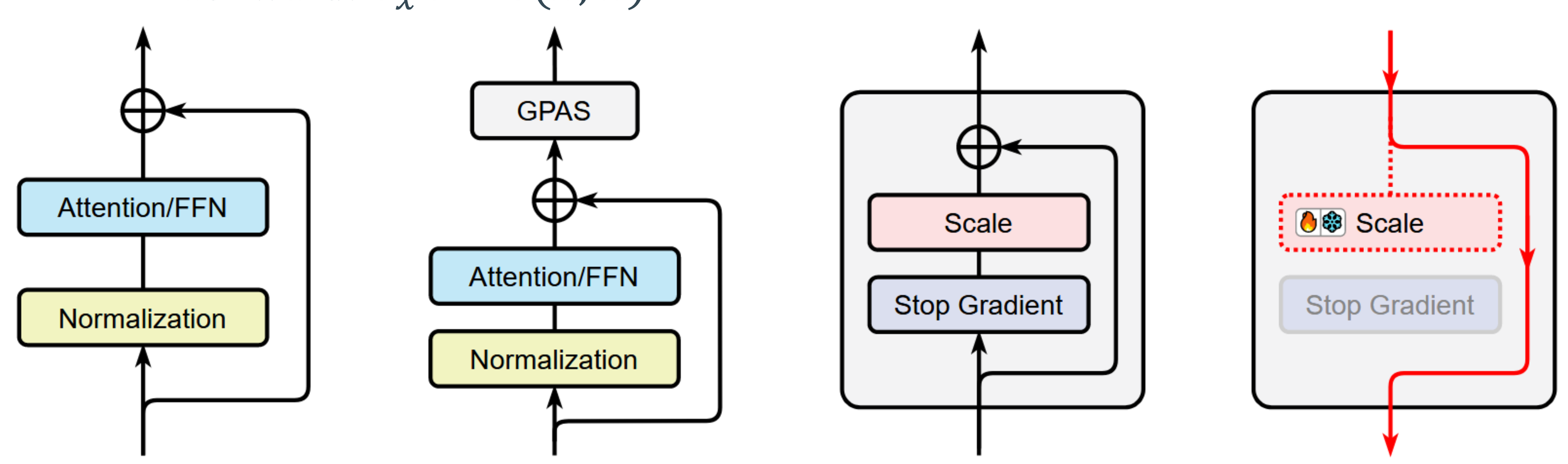
Modern LLMs are mostly built on Pre-LN Transformers. While being stable for large scale training, Pre-LN suffers from exponential activation growth across layers. This means deeper layers' Attention and FFN outputs will be overpowered by shortcuts, limiting their contribution to learning.

Gradient-Preserving Activation Scaling

We propose to mitigate this growth by Gradient-Preserving Activation Scaling, which scales layerwise activations without scaling their backward gradients. The motivation is to scale down forward activations without downscaling gradients to avoid gradient vanishing.

GPAS definition: $\mathbf{GPAS}(\boldsymbol{x}, \boldsymbol{\alpha}) = \boldsymbol{x} - \boldsymbol{\alpha} \cdot \mathbf{sg}(\boldsymbol{x})$

- Forward: $\mathrm{GPAS}(x, \alpha) = (1 - \alpha)x$
- Backward: $\partial_x \mathrm{GPAS}(x, \alpha) = I$



(a) Pre-LN (b) Pre + GPAS (c) GPAS block (d) GPAS backprop

Apply GPAS to various Transformer variants

α_l: learnable scalar. SiLU: avoid excessively scaling up activation.

Pre-LN: $x_{l+1} = x_l + f(\mathrm{LN}(x_l))$

Pre+GPAS: $x'_{l+1} = x_l + f\big(\mathrm{LN}(x_l)\big),\ x_{l+1} = x'_{l+1} - \mathrm{SiLU}(\alpha_l) \cdot \mathrm{sg}(x'_{l+1})$

LNS: $x_{l+1} = x_l + f(\mathrm{LN}(x_l)/\sqrt{l})$

LNS+GPAS: $x'_{l+1} = x_l + f\big(\mathrm{LN}(x_l)/\sqrt{l}\big),\ x_{l+1} = x'_{l+1} - \mathrm{SiLU}(\alpha_l) \cdot \mathrm{sg}(x'_{l+1})$

Sandwich-LN: $x_{l+1} = x_l + \mathrm{LN}(f\big(\mathrm{LN}(x_l)\big))$

Sandwich+GPAS: $x'_{l+1} = x_l + \mathrm{LN}(f\big(\mathrm{LN}(x_l)\big)),\ x_{l+1} = x'_{l+1} - \mathrm{SiLU}(\alpha_l) \cdot \mathrm{sg}(x'_{l+1})$

DeepNorm: $x_{l+1} = \mathrm{LN}(\alpha \cdot x_l + f_\beta(x_l))$

DeepNorm+GPAS: $x'_l = x_l - \mathrm{SiLU}(\alpha_l) \cdot \mathrm{sg}(x_l),\ x_{l+1} = \mathrm{LN}(\alpha \cdot x'_l + f_\beta(x_l))$

Mix-LN (Pre-LN layer): same as Pre + GPAS

Mix-LN (Post-LN layer): $x'_l = x_l - \mathrm{SiLU}(\alpha_l) \cdot \mathrm{sg}(x_l),\ x_{l+1} = \mathrm{LN}(x'_l + f(x_l))$

Experiments

Pretrain perplexity

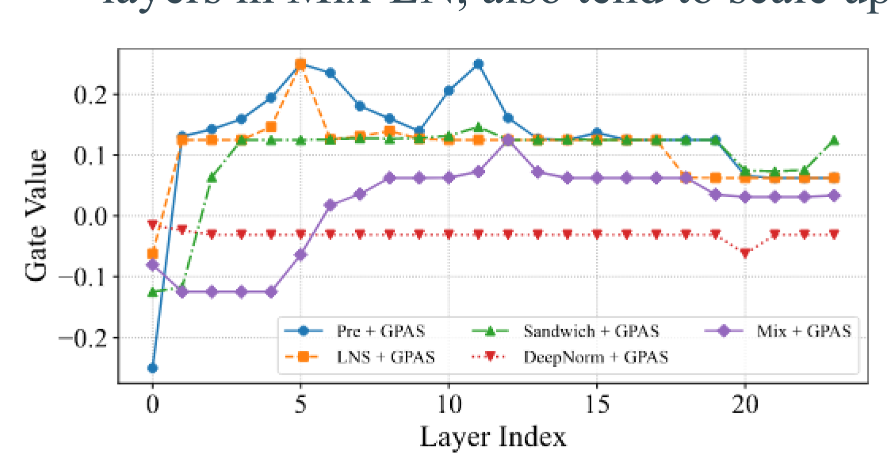
Method	71M	130M	250M	350M	1B
Post-LN [1]	33.80	26.50	1351.58	21.19	1406.66
DeepNorm [14]	35.49	26.78	22.20	21.76	1400.39
DeepNorm + GPAS	34.78 (-0.71)	26.62 (-0.16)	21.89 (-0.31)	21.29 (-0.47)	16.01 (-1384)
Pre-LN [20]	33.98	26.61	21.54	20.71	16.53
Pre + GPAS	33.38 (-0.60)	26.25 (-0.36)	21.34 (-0.20)	19.77 (-0.94)	16.11 (-0.42)
Sandwich-LN [15]	32.28	25.31	20.43	20.20	16.26
Sandwich + GPAS	**31.44** (-0.84)	**24.86** (-0.45)	20.38 (-0.05)	19.45 (-0.75)	15.85 (-0.41)
Mix-LN [12]	33.88	26.29	21.52	20.73	15.87
Mix + GPAS	33.26 (-0.62)	26.03 (-0.26)	21.43 (-0.09)	19.82 (-0.91)	15.38 (-0.49)
LNS [13]	34.58	25.91	20.59	20.35	15.61
LNS + GPAS	32.68 (-1.90)	24.95 (-0.96)	**19.89** (-0.70)	**19.38** (-0.97)	**14.87** (-0.74)

Benchmark performance after supervised finetuning

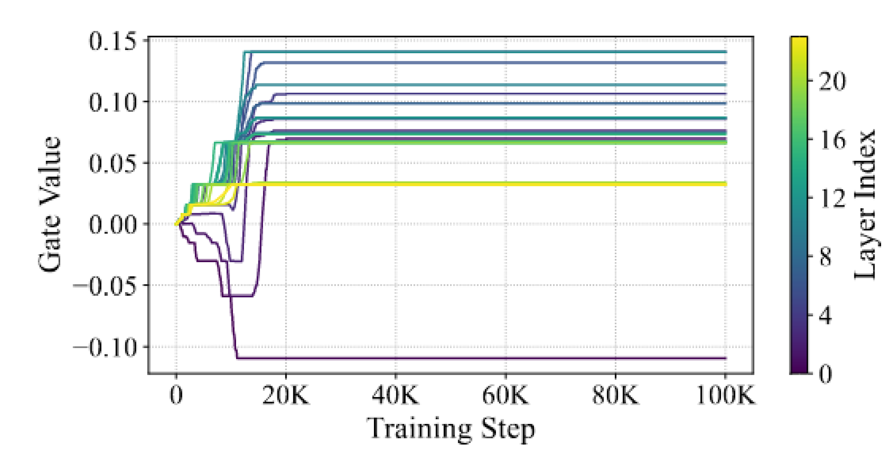
Method	MMLU	BoolQ	PIQA	SIQA	HellaSwag	WinoG	ARC-e	ARC-c	OBQA	Average
Post-LN	22.95	37.83	52.77	34.03	26.20	48.15	27.36	19.37	11.40	31.12
DeepNorm	22.95	37.83	52.77	34.08	26.20	51.14	27.31	19.37	11.40	31.45
DeepNorm + GPAS	**26.46**	*62.11*	**69.53**	**46.93**	**34.37**	**52.09**	**49.24**	**22.61**	**20.40**	**42.64**
Pre-LN	25.96	50.34	68.66	44.27	32.39	51.14	49.37	21.33	17.60	40.12
Pre + GPAS	**26.68**	**59.79**	**69.31**	**46.52**	**33.64**	**52.49**	**49.79**	**22.70**	**22.00**	**42.55**
Sandwich-LN	**27.42**	61.77	67.63	44.68	32.76	**50.67**	47.43	23.12	21.40	41.88
Sandwich + GPAS	27.29	**61.90**	**69.15**	**45.29**	**34.61**	50.36	**51.39**	**23.46**	**22.20**	**42.85**
Mix-LN	**26.24**	61.93	68.66	45.50	33.09	52.25	48.78	**24.40**	20.80	42.40
Mix + GPAS	26.23	**61.99**	**69.59**	**45.60**	**33.51**	*53.51*	**50.34**	22.35	**22.40**	**42.83**
LNS	26.62	**62.02**	69.48	45.39	34.76	**51.38**	50.88	23.29	19.80	42.63
LNS + GPAS	***27.78***	61.56	***71.00***	***47.49***	***36.19***	51.22	***52.57***	***25.51***	***24.40***	***44.19***

Learned scaling values for various normalization schemes

- **Pre-LN layers tend to learn to scale down activation.**
- **Post-LN layers tend to learn to scale up the skip connection.**
- Variants similar to Pre-LN, such as LNS, sandwich-LN, and Pre-LN layers in Mix-LN also tend to learn to scale down activations.
- Variants similar to Post-LN, such as DeepNorm and the Post-LN layers in Mix-LN, also tend to scale up the shortcut.



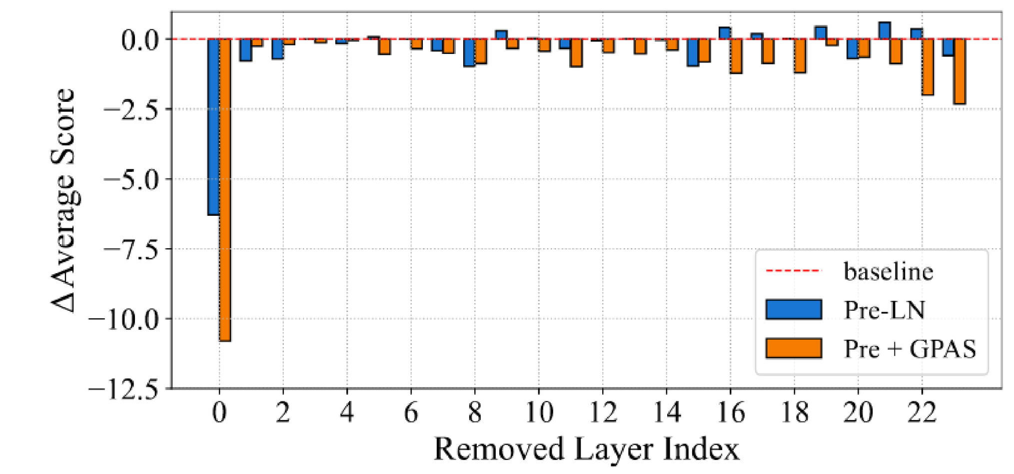
(a) Learned gate values α_l for different models



(b) Activated gate values $\mathrm{SiLU}(\alpha_l)$ across training steps of Pre + GPAS

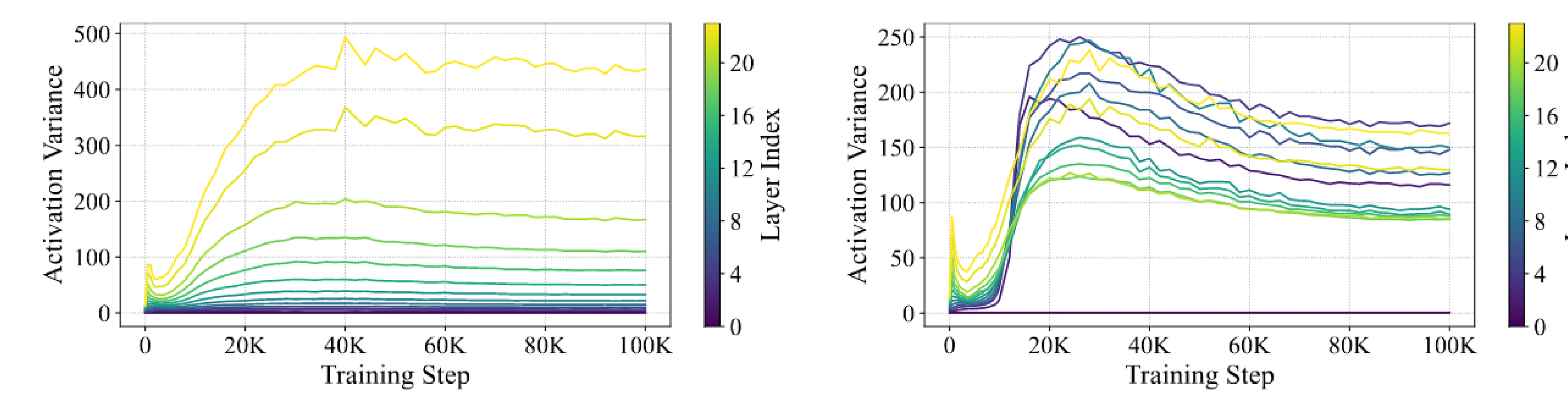
GPAS enhances deeper layers

- We measure layer importance as the drop in average benchmark score after removing that layer.
- Vanilla Pre-LN's deeper layers have little contribution.
- GPAS enhances importance of deeper layers significantly.



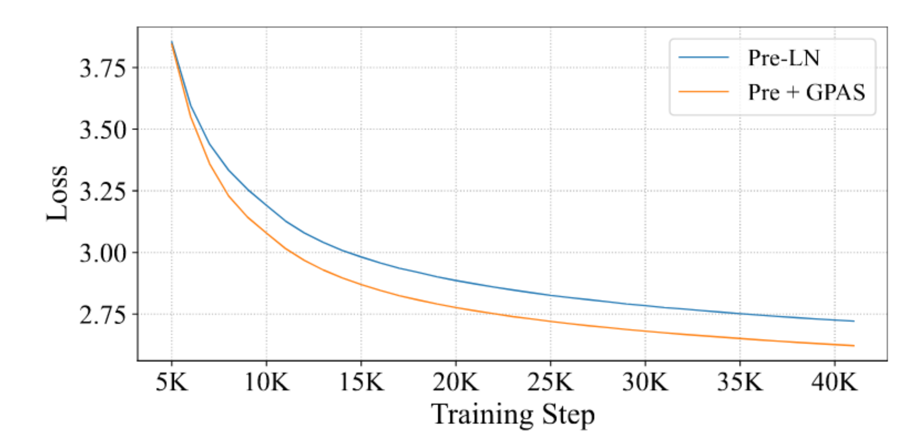
GPAS model properties

More uniform and compact activation variance



(a) Pre-LN (b) Pre + GPAS

Pretrain eval loss curve on 7B models



Contact

- Tianhao Chen, HKUST
- Email: `tchenbb@connect.ust.hk`
- WeChat: see QR Code 👉

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY · UNIVERSITY OF SURREY · idea · 大连理工大学 DALIAN UNIVERSITY OF TECHNOLOGY · EMORY UNIVERSITY · UNIVERSITY OF OXFORD · NVIDIA · TEXAS The University of Texas at Austin