Pre-vs-post norm

The one thing *everyone* agrees on (in 2024)

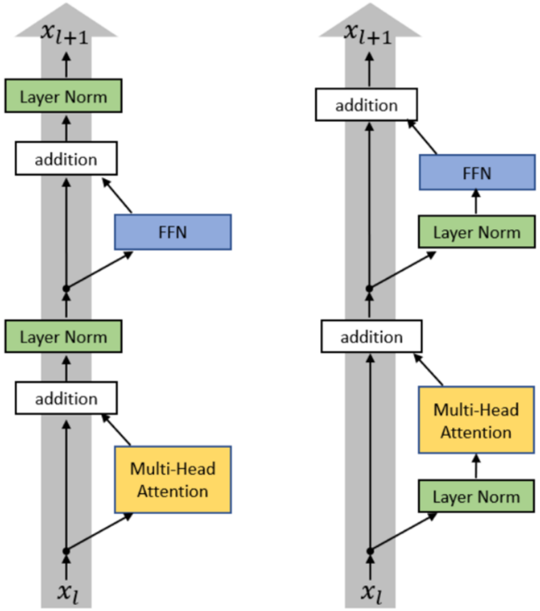
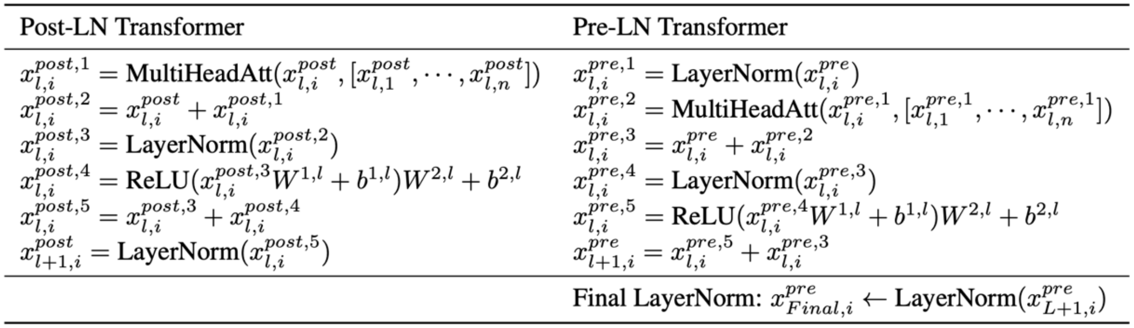


Figure from Xiong 2020



Post-LN Transformer	Pre-LN Transformer
$x_{l,i}^{post,1} = \text{MultiHeadAtt}(x_{l,i}^{post}, [x_{l,1}^{post}, \cdots, x_{l,n}^{post}])$ $x_{l,i}^{post,2} = x_{l,i}^{post} + x_{l,i}^{post,1}$ $x_{l,i}^{post,3} = \text{LayerNorm}(x_{l,i}^{post,2})$ $x_{l,i}^{post,4} = \text{ReLU}(x_{l,i}^{post,3} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$ $x_{l,i}^{post,5} = x_{l,i}^{post,3} + x_{l,i}^{post,4}$ $x_{l+1,i}^{post} = \text{LayerNorm}(x_{l,i}^{post,5})$	$x_{l,i}^{pre,1} = \text{LayerNorm}(x_{l,i}^{pre})$ $x_{l,i}^{pre,2} = \text{MultiHeadAtt}(x_{l,i}^{pre,1}, [x_{l,1}^{pre,1}, \cdots, x_{l,n}^{pre,1}])$ $x_{l,i}^{pre,3} = x_{l,i}^{pre} + x_{l,i}^{pre,2}$ $x_{l,i}^{pre,4} = \text{LayerNorm}(x_{l,i}^{pre,3})$ $x_{l,i}^{pre,5} = \text{ReLU}(x_{l,i}^{pre,4} W^{1,l} + b^{1,l}) W^{2,l} + b^{2,l}$ $x_{l+1,i}^{pre} = x_{l,i}^{pre,5} + x_{l,i}^{pre,3}$
	Final LayerNorm: $x_{Final,i}^{pre} \leftarrow \text{LayerNorm}(x_{L+1,i}^{pre})$

Set up LayerNorm so that it doesn't affect the main residual signal path (on the left)

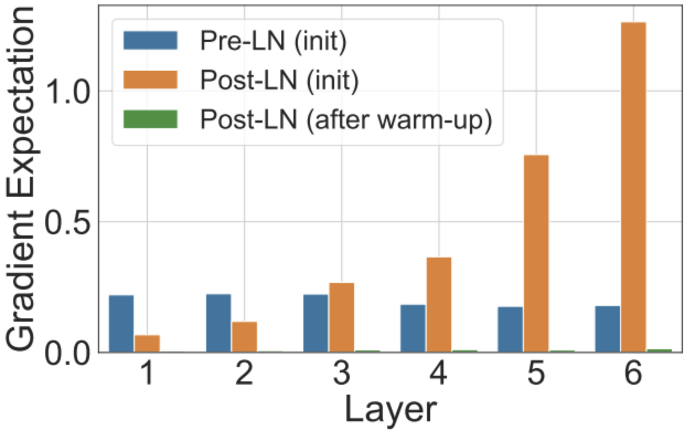
Almost all modern LMs use pre-norm (but BERT was post-norm)

(One somewhat funny exception – OPT350M. I don't know why this is post-norm)

Borrowed from Stanford CS336

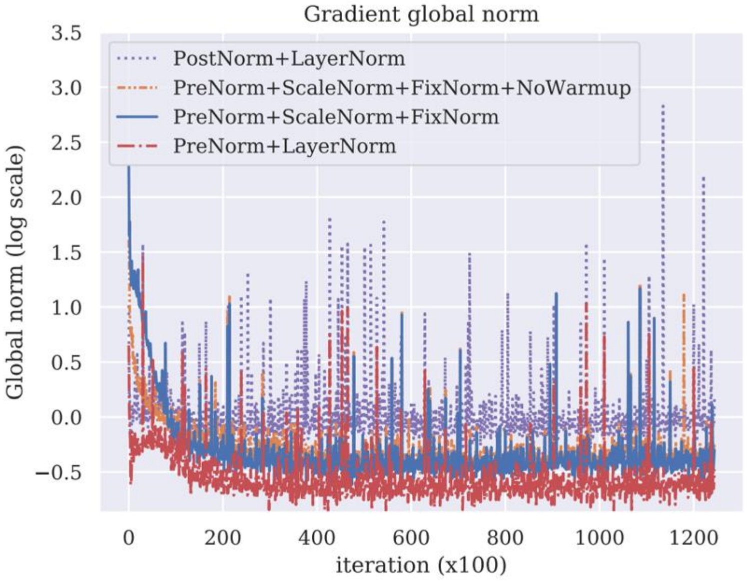
Pre-vs-post norm, explanations?

Gradient attenuation [Xiong 2020]



(a) W^1 in the FFN sub-layers

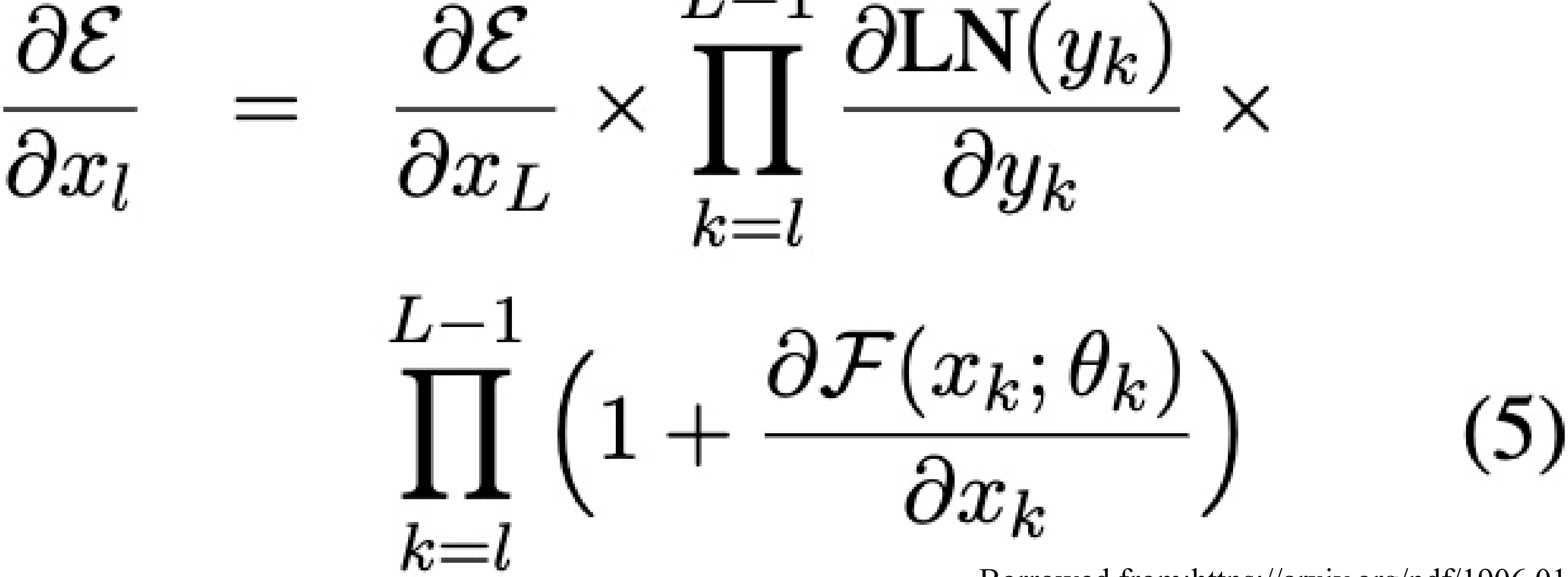
Gradient spikes [Salazar and Ngyuen]



Original stated advantage– removing warmup.
Today – stability and larger LRs for large networks

Borrowed from Stanford CS336

post-norm



$$\frac{\partial \mathcal{E}}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \times \prod_{k=l}^{L-1} \frac{\partial \mathrm{LN}(y_k)}{\partial y_k} \times \prod_{k=l}^{L-1} \left(1 + \frac{\partial \mathcal{F}(x_k;\theta_k)}{\partial x_k}\right) \quad (5)$$

Borrowed from:https://arxiv.org/pdf/1906.01787
Learning Deep Transformer Models for Machine Translation

pre-norm

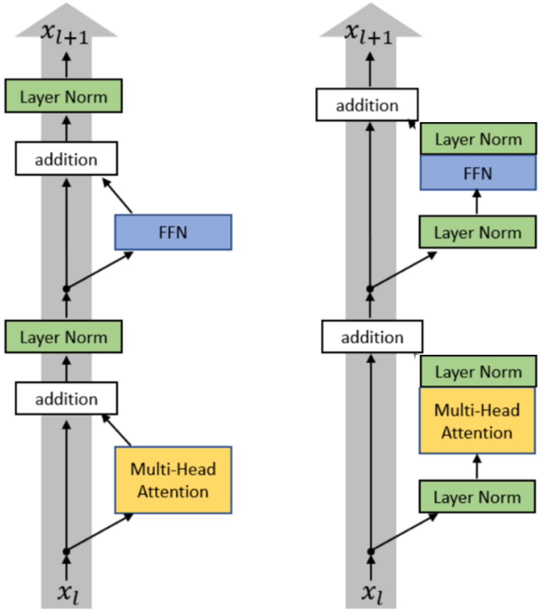
$$\frac{\partial \mathcal{E}}{\partial x_l} = \frac{\partial \mathcal{E}}{\partial x_L} \times \left(1 + \sum_{k=l}^{L-1} \frac{\partial \mathcal{F}(\mathrm{LN}(x_k); \theta_k)}{\partial x_l}\right) \quad (6)$$

Borrowed from:https://arxiv.org/pdf/1906.01787

Learning Deep Transformer Models for Machine Translation

New things – ‘double’ norm.

If putting LayerNorms in residual streams is bad.. Why not post-norm outside the stream?



Recent models: Grok, Gemma 2. Olmo 2 *only* does non-residual post norm

Borrowed from Stanford CS336