Understanding and Improving Layer Normalization

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guanxiang Zhao, Junyang Lin
MOE Key Lab of Computational Linguistics, School of EECS, Peking University
{jingjingxu,xusun,zzy1210,zhaoguangxiang,linjunyang}@pku.edu.cn

What is layer normalization?

- Layer Normalization (LayerNorm) is a widely-used technique that scales the distributions of intermediate layers to have zero mean and unit standard deviation.
- It enables smoother gradients, faster training, and better generalization accuracy.

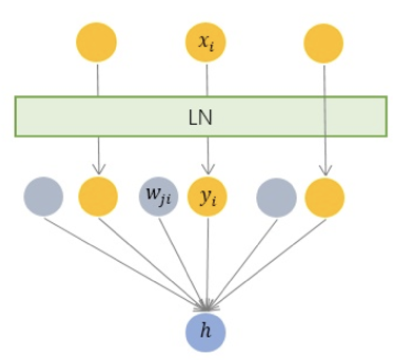


Figure 1:Illustration of LayerNorm.

How does LayerNorm work?

- The widely accepted explanation is that forward normalization brings distribution stability.
- However, recent studies show that the effects of normalization have nothing to do with the stability of input distribution.
- It is still unclear where the success of LayerNorm stems from.

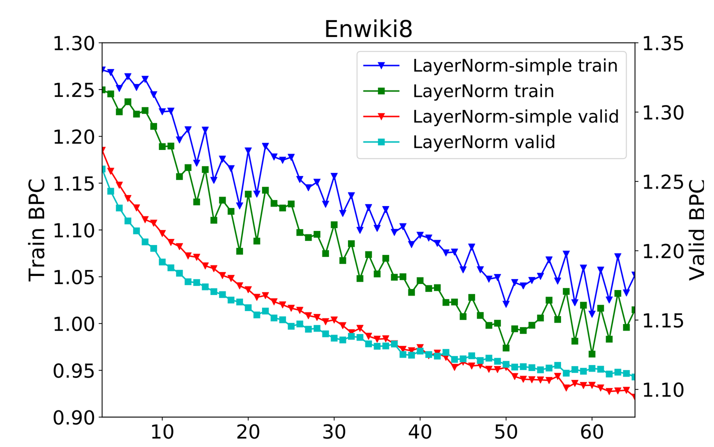
How do we explore LayerNorm?

To investigate how LayerNorm works, we conduct a series of experiments on different tasks.

- **Machine translation** includes three widely-used datasets, WMT English-German, IWSLT 14 German-English and IWSLT 15 English-Vietnamese.
- **Language modeling** includes a large dataset, Enwiki8.
- **Text classification** includes two sentence classification datasets: RT, and SST5.
- **Image classification** includes a widely-used dataset, MNIST.
- **Dependency parsing** uses English Penn TreeBank.

Observation 1: The bias and gain do not work in most cases.

- Dropping the bias and gain ("LayerNorm-simple") does not decrease the performance on six datasets. Surprisingly, it outperforms LayerNorm on four datasets and achieves SOTA on En-Vi translation.
- From convergence curves, we can see that current affine transformation mechanism has a potential risk of over-fitting and needs to be further improved.



Models	Machine Translation			Language Modeling	Classification		
	En-De(+)	De-En(+)	En-Vi(+)	Enwiki8(-)	RT(+)	SST5(+)	MNIST(+)
Model Layers	12	12	12	12	4	4	3
w/o Norm	Diverge	34.0	28.4	**1.04**	76.85	38.55	**99.14**
LayerNorm	28.3	**35.5**	31.2	1.07	**77.21**	39.23	99.13
LayerNorm-simple	**28.4**	**35.5**	**31.6**	1.07	76.66	**40.54**	99.09

Table 1:The bias and gain do not work in most cases.

Observation 2: The derivatives of the mean and variance are more important than forward normalization.

- "DetachNorm" treats the mean and variance as changeable constants, rather than variables. The difference between LayerNorm and DetachNorm is that DetachNorm detaches the derivatives of the mean and variance.
- The derivatives of the mean and variance bring higher improvements than forward normalization does.
- The derivative of mean μ re-centers $\frac{\partial \ell}{\partial \mathbf{x}}$ to zero. The derivative of variance σ reduces the variance of $\frac{\partial \ell}{\partial \mathbf{x}}$, which can be seen a kind of re-scaling.
- The derivative of variance is more important than that of mean for deeper networks.

Theorem

Given $\frac{\partial \ell}{\partial \boldsymbol{y}} = (g_1, g_2, ..., g_H)^T$, let $\bar{g}$ and D_g be the mean and variance of $g_1, g_2, ..., g_H$. For the case of detaching the derivatives of μ and σ, suppose $\frac{\partial \ell}{\partial \boldsymbol{x}} = (a_1, a_2, ..., a_H)^T$ is the gradient of $\boldsymbol{x}$ with mean $\bar{a}$ and variance D_a. We have $\bar{a} = \bar{g}/\sigma$ and $D_a = D_g/(\sigma^2)$.

(1) For the case of standard LayerNorm-simple, suppose $\frac{\partial \ell}{\partial \boldsymbol{x}} = (b_1, b_2, ..., b_H)^T$ is the gradient of $\boldsymbol{x}$ with mean $\bar{b}$ and variance D_b. We have $\bar{b} = 0$ and $D_b \le D_g/(\sigma^2)$.

(2) For the case of detaching the derivative of μ, suppose $\frac{\partial \ell}{\partial \boldsymbol{x}} = (c_1, c_2, ..., c_H)^T$ is the gradient of $\boldsymbol{x}$ with mean $\bar{c}$ and variance D_c. We have $\bar{c} = \bar{g}/\sigma$ and $D_c \le D_g/(\sigma^2)$.

(3) For the case of detaching the derivative of σ, suppose $\frac{\partial \ell}{\partial \boldsymbol{x}} = (d_1, d_2, ..., d_H)^T$ is the gradient of $\boldsymbol{x}$ with mean $\bar{d}$ and variance D_d. We have $\bar{d} = 0$ and $D_c = D_g/(\sigma^2)$.

Models	Machine Translation			Language Modeling	Classification			Parsing
	En-De	De-En(+)	En-Vi(+)	Enwiki8(-)	RT(+)	SST5(+)	MNIST(+)	PTB(+)
Model Layers	12	12	12	12	4	4	3	3
w/o Norm	Diverge	**34.0**	**28.4**	**1.04**	**76.85**	38.55	**99.14**	88.31
DetachNorm	Diverge	33.9	27.7	1.12	76.40	**40.04**	99.10	**89.79**
Improvement	–	-0.1	-0.7	-0.08	-0.45	1.49	-0.04	**1.48**
Models	Machine Translation			Language Modeling	Classification			Parsing
	En-De	De-En(+)	En-Vi(+)	Enwiki8(-)	RT(+)	SST5(+)	MNIST(+)	PTB(+)
Model Layers	12	12	12	12	4	4	3	3
DetachNorm	Diverge	33.9	27.7	1.12	76.40	40.04	**99.10**	**89.79**
LayerNorm-simple	**28.4**	**35.5**	**31.6**	**1.07**	**76.66**	**40.54**	99.09	89.19
Improvement	–	**1.6**	**3.9**	**0.05**	**0.26**	**0.50**	-0.01	-0.60

Table 2:The derivatives of the mean and variance matter.

AdaNorm

To address the over-fitting problem, we propose a normalization method, Adaptive Normalization.

$$\mathbf{z} = \phi(\mathbf{y}) \odot \mathbf{y} = \phi(N(\mathbf{x})) \odot N(\mathbf{x}) \quad (1)$$

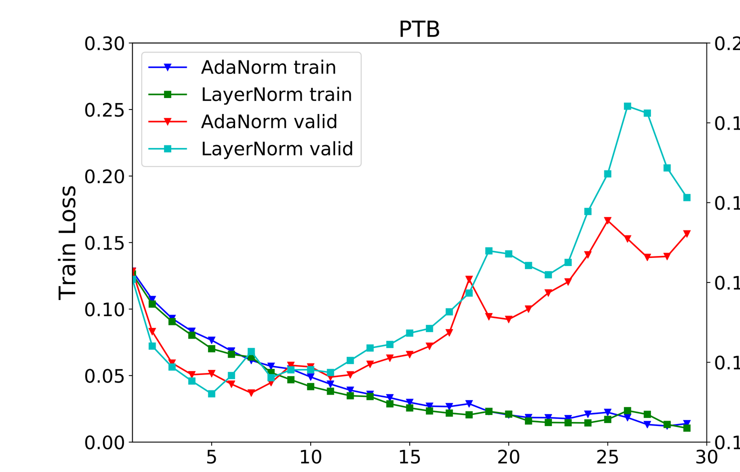
It achieves better results on seven out of eight datasets.

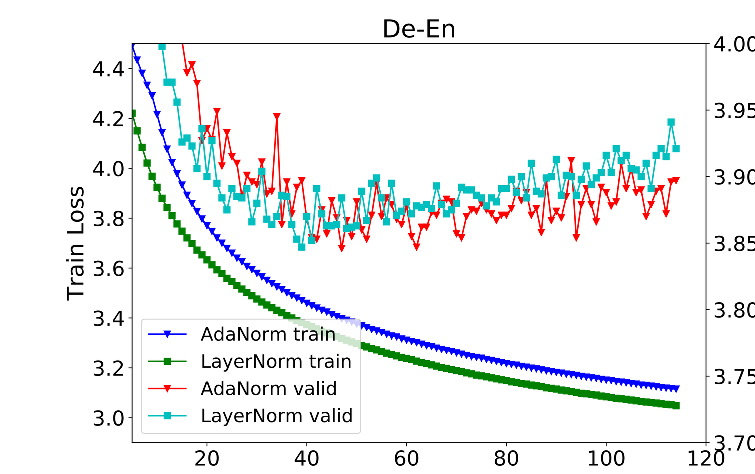
Models	Machine Translation			Language Model	Classification			Parsing
	En-De(+)	De-En(+)	En-Vi(+)	Enwiki8(-)	RT(+)	SST5(+)	MNIST(+)	PTB(+)
w/o Norm	Diverge	34.0	28.4	**1.04**	76.85	38.55	99.14	88.31
LayerNorm	28.3	35.5	31.2	1.07	77.21	39.23	99.13	89.12
LayerNorm-simple	28.4	35.5	**31.6**	1.07	76.66	**40.54**	99.09	89.19
AdaNorm	**28.5**	**35.6**	31.4	1.07	**77.50**	**40.54**	**99.35**	**89.23**

Table 3:Results of LayerNorm and AdaNorm.

Better Convergence

- Compared to AdaNorm, LayerNorm has lower training loss but higher validation loss. Lower validation loss proves that AdaNorm has better convergence.





Couclusions

- In this paper, we investigate how layer normalization works.
- Based on a series of experiments and theoretical analysis, we summarize some interesting conclusions.
- We find that the derivatives of the mean and variance are important to the success of LayerNorm by re-centering and re-scaling backward gradients. Furthermore, the bias and gain increase the risk of over-fitting and do not work in most cases.
- To address the over-fitting problem, we propose a normalization method AdaNorm. Experiments show that AdaNorm outperforms LayerNorm on seven datasets.
- In the future work, we would like to explore more alternatives to LayerNorm from the perspective of gradient normalization.