Understanding Transformer Models at Initialization

On Layer Normalization in the Transformer Architecture

Yuchong Zhang and Jonah Mackey

MAT1510
Department of Mathematics
University of Toronto

December 6, 2022

Table of Contents

Table of Contents

Transformers (2017)

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.0 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature.

The Architecture

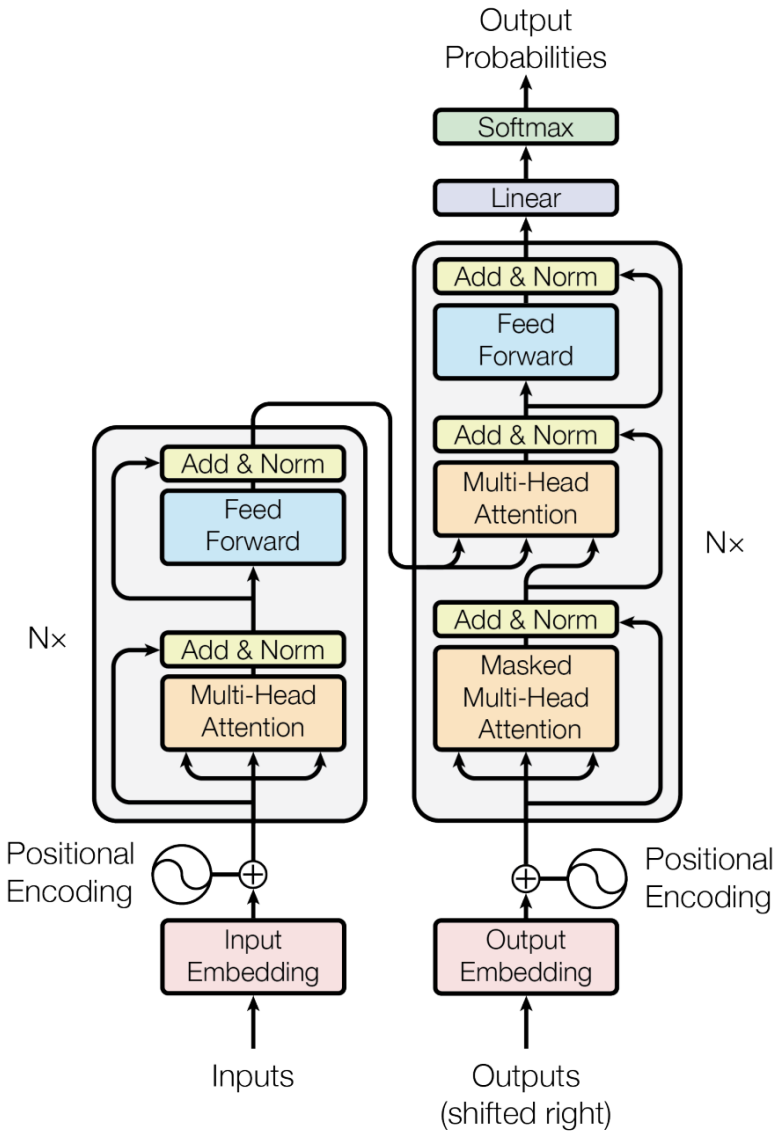


Figure 1: The Transformer - model architecture.

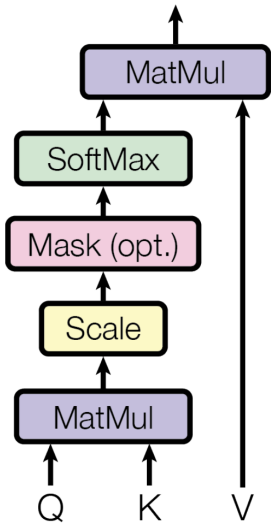
Scaled Dot-Product Attention

$$X = [x_1, \ \ldots, \ x_n], \ x_i \in \mathbb{R}^d$$

$$W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$$

$$Q = XW^Q, \ K = XW^K, \ V = XW^V$$

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V$$



Layer Normalization

$a^k \in \mathbb{R}^d$: kth layer activations

$$\mu = \frac{1}{d} \sum_{i=1}^{d} a_i^k$$

$$\sigma = \frac{1}{d} \sqrt{\sum_{i=1}^{n} (a_i^k - \mu)^2}$$

$$\tilde{a_i^k} = \frac{a_i^k - \mu}{\sigma}$$

Layer normalization makes the activations in each layer have mean 0 and variance 1.

Training Difficulties

In order for transformers to converge during training, we require:

Training Difficulties

In order for transformers to converge during training, we require:

- Adaptive optimizers (Adam) rather than SGD.

Training Difficulties

In order for transformers to converge during training, we require:

- Adaptive optimizers (Adam) rather than SGD.
- Residual connections

Training Difficulties

In order for transformers to converge during training, we require:

- Adaptive optimizers (Adam) rather than SGD.
- Residual connections
- Layer normalization

Training Difficulties

In order for transformers to converge during training, we require:

- Adaptive optimizers (Adam) rather than SGD.
- Residual connections
- Layer normalization
- Learning rate warm-up

Training Difficulties

In order for transformers to converge during training, we require:

- Adaptive optimizers (Adam) rather than SGD.
- Residual connections
- Layer normalization
- Learning rate warm-up

It is not theoretically understood why these tricks are needed.

Table of Contents

Main Paper (2020)

On Layer Normalization in the Transformer Architecture

Ruibin Xiong[†*12] **Yunchang Yang**[*3] **Di He**[45] **Kai Zheng**[4] **Shuxin Zheng**[5] **Chen Xing**[6] **Huishuai Zhang**[5]
Yanyan Lan[12] **Liwei Wang**[43] **Tie-Yan Liu**[5]

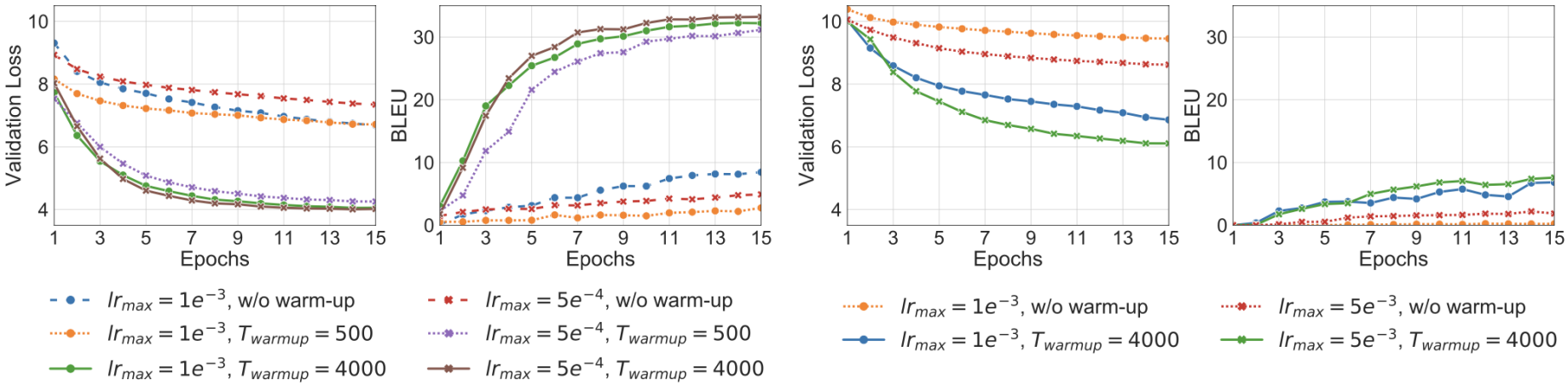
Abstract

The Transformer is widely used in natural language processing tasks. To train a Transformer however, one usually needs a carefully designed learning rate warm-up stage, which is shown to be crucial to the final performance but will slow down the optimization and bring more hyper-parameter tunings. In this paper, we first study theoretically why the learning rate warm-up stage is essential and show that the location of layer normalization matters. Specifically, we prove with mean field theory that at initialization, for the original-designed Post-LN Transformer, which places the layer normalization between the residual blocks, the expected gradients of the parameters near the output layer are large. Therefore, us-

1. Introduction

The Transformer (Vaswani et al., 2017) is one of the most commonly used neural network architectures in natural language processing. Layer normalization (Lei Ba et al., 2016) plays a key role in Transformer's success. The originally designed Transformer places the layer normalization between the residual blocks, which is usually referred to as the Transformer with Post-Layer Normalization (Post-LN) (Wang et al., 2019). This architecture has achieved state-of-the-art performance in many tasks including language modeling (Dai et al., 2019; Al-Rfou et al., 2018) and machine translation (Dehghani et al., 2018; Edunov et al., 2018). Unsupervised pre-trained models based on the Post-LN Transformer architecture also show impressive performance in many downstream tasks (Radford et al., 2019; Devlin et al., 2018; Yang et al., 2019b).

Motivation



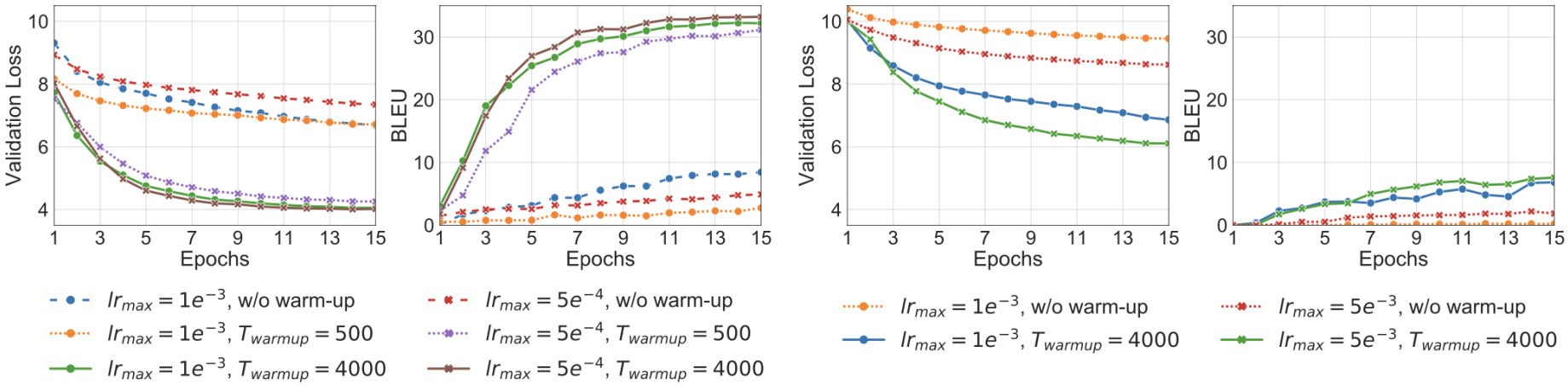
(a) Loss/BLEU on the IWSLT14 De-En task (Adam)

(b) Loss/BLEU on the IWSLT14 De-En task (SGD)

Figure 2. Performances of the models optimized by Adam and SGD on the IWSLT14 De-En task.

Observations:

Motivation



(a) Loss/BLEU on the IWSLT14 De-En task (Adam)

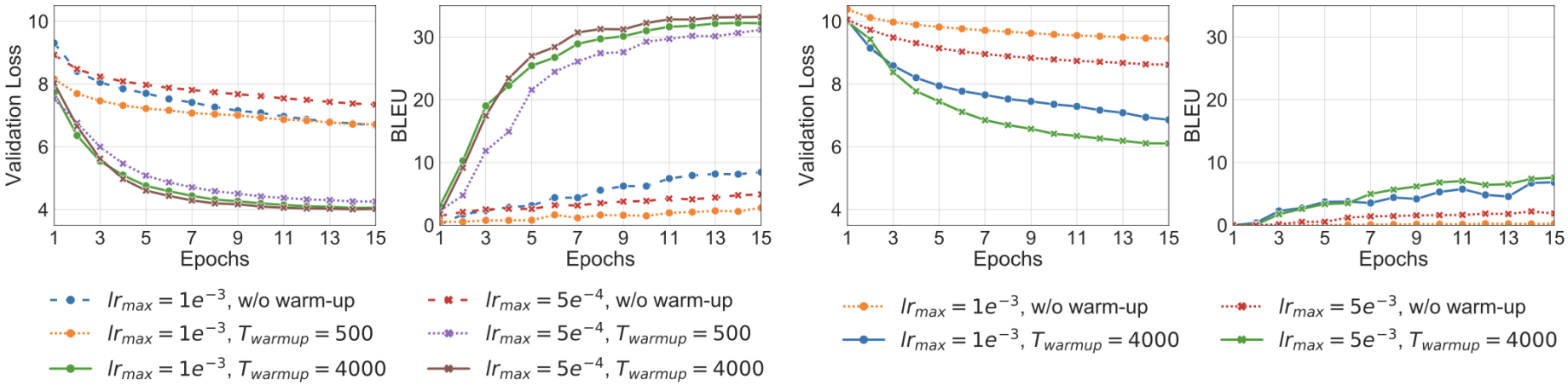
(b) Loss/BLEU on the IWSLT14 De-En task (SGD)

Figure 2. Performances of the models optimized by Adam and SGD on the IWSLT14 De-En task.

Observations:

- Adam performs better than SGD.

Motivation



(a) Loss/BLEU on the IWSLT14 De-En task (Adam)

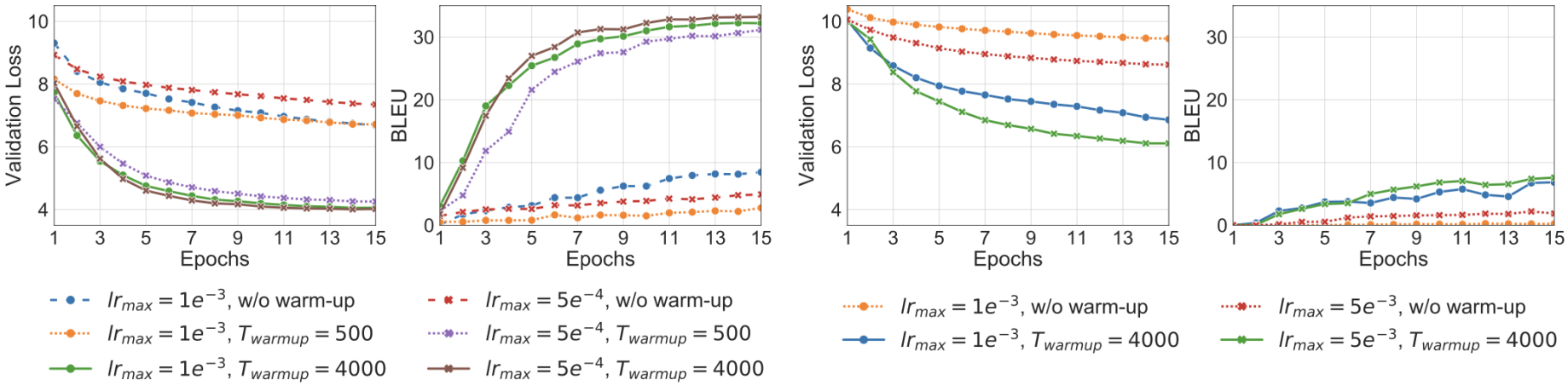
(b) Loss/BLEU on the IWSLT14 De-En task (SGD)

Figure 2. Performances of the models optimized by Adam and SGD on the IWSLT14 De-En task.

Observations:

- Adam performs better than SGD.
- Learning rate warm-up is essential for convergence.

Motivation



(a) Loss/BLEU on the IWSLT14 De-En task (Adam)

(b) Loss/BLEU on the IWSLT14 De-En task (SGD)

Figure 2. Performances of the models optimized by Adam and SGD on the IWSLT14 De-En task.

Observations:

- Adam performs better than SGD.
- Learning rate warm-up is essential for convergence.
- Training dynamics are sensitive to LR warm-up hyperparameters.

Issues with warm-up

Slows down training.

Adds more hyperparameters.

Main Contribution

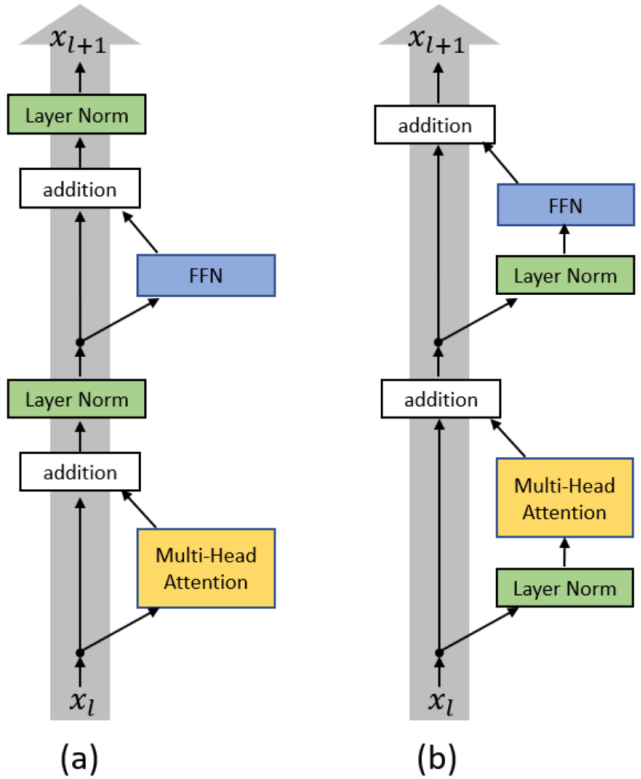


Figure 1. (a) Post-LN Transformer layer; (b) Pre-LN Transformer layer.

Proposed solution: change location of layer normalization.

Their claim: this stabilizes training and removes need for LR warm-up.

Theoretical Justification

Under **certain assumptions** about model initialization:

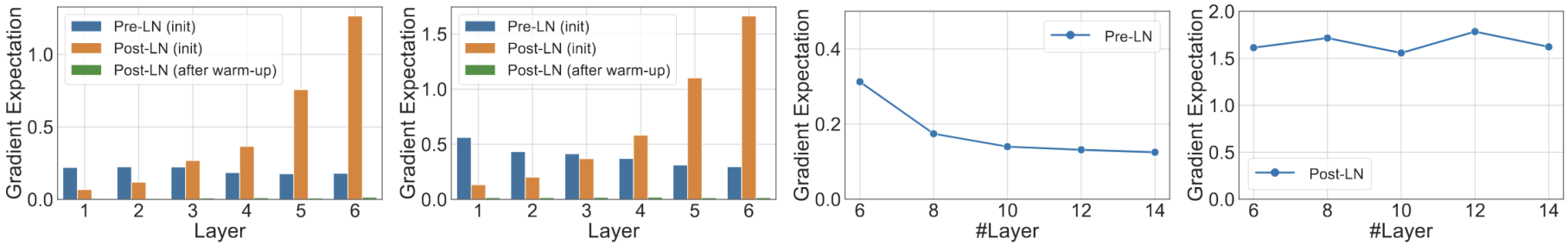
Theorem 1 (Gradients of the last layer in the Transformer). *Assume that* $\|x_{L,i}^{post,5}\|_2^2$ *and* $\|x_{L+1,i}^{pre}\|_2^2$ *are* (ϵ, δ)*-bounded for all* i*, where* ϵ *and* $\delta = \delta(\epsilon)$ *are small numbers. Then with probability at least* $0.99 - \delta - \frac{\epsilon}{0.9+\epsilon}$*, for the Post-LN Transformer with* L *layers, the gradient of the parameters of the last layer satisfies*

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}(d\sqrt{\ln d})$$

and for the Pre-LN Transformer with L *layers,*

$$\|\frac{\partial \tilde{\mathcal{L}}}{\partial W^{2,L}}\|_F \leq \mathcal{O}\left(d\sqrt{\frac{\ln d}{L}}\right).$$

Empirical Evidence



(a) W^1 in the FFN sub-layers (b) W^2 in the FFN sub-layers (c) Pre-LN Transformer (d) Post-LN Transformer

Figure 3. The norm of gradients of 1. different layers in the 6-6 Transformer (a,b). 2. $W^{2,L}$ in different size of the Transformer (c,d).

Authors Conclusions

Post-LN:

$$||\frac{\partial L}{\partial W^{2,L}}||_F \leq O(d\sqrt{\ln d})$$

- Gradient norm of last layer weights are large at initialization.
- This makes training unstable in the early stages.
- Requires LR warm-up.

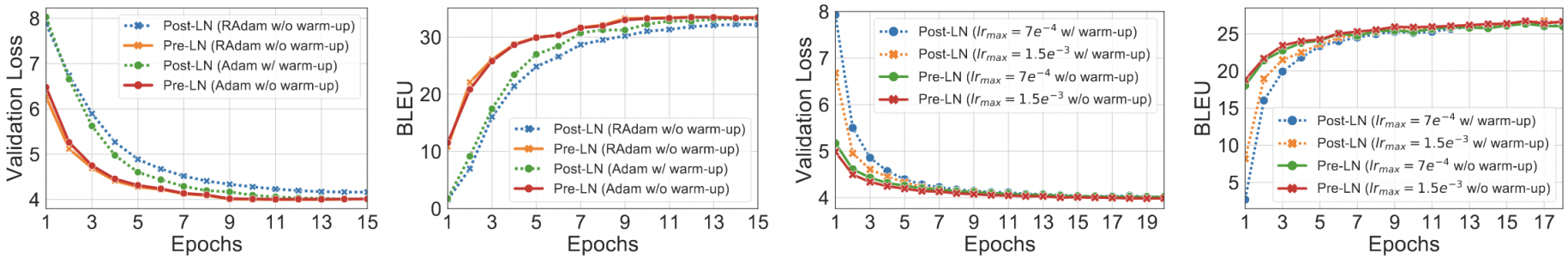
Authors Conclusions (cont.)

Pre-LN:

$$||\frac{\partial L}{\partial W^{2,L}}||_F \leq O(d\sqrt{\frac{\ln d}{L}})$$

- Gradient norm of weights decreases with depth.
- There is no need for LR warm-up.

Results



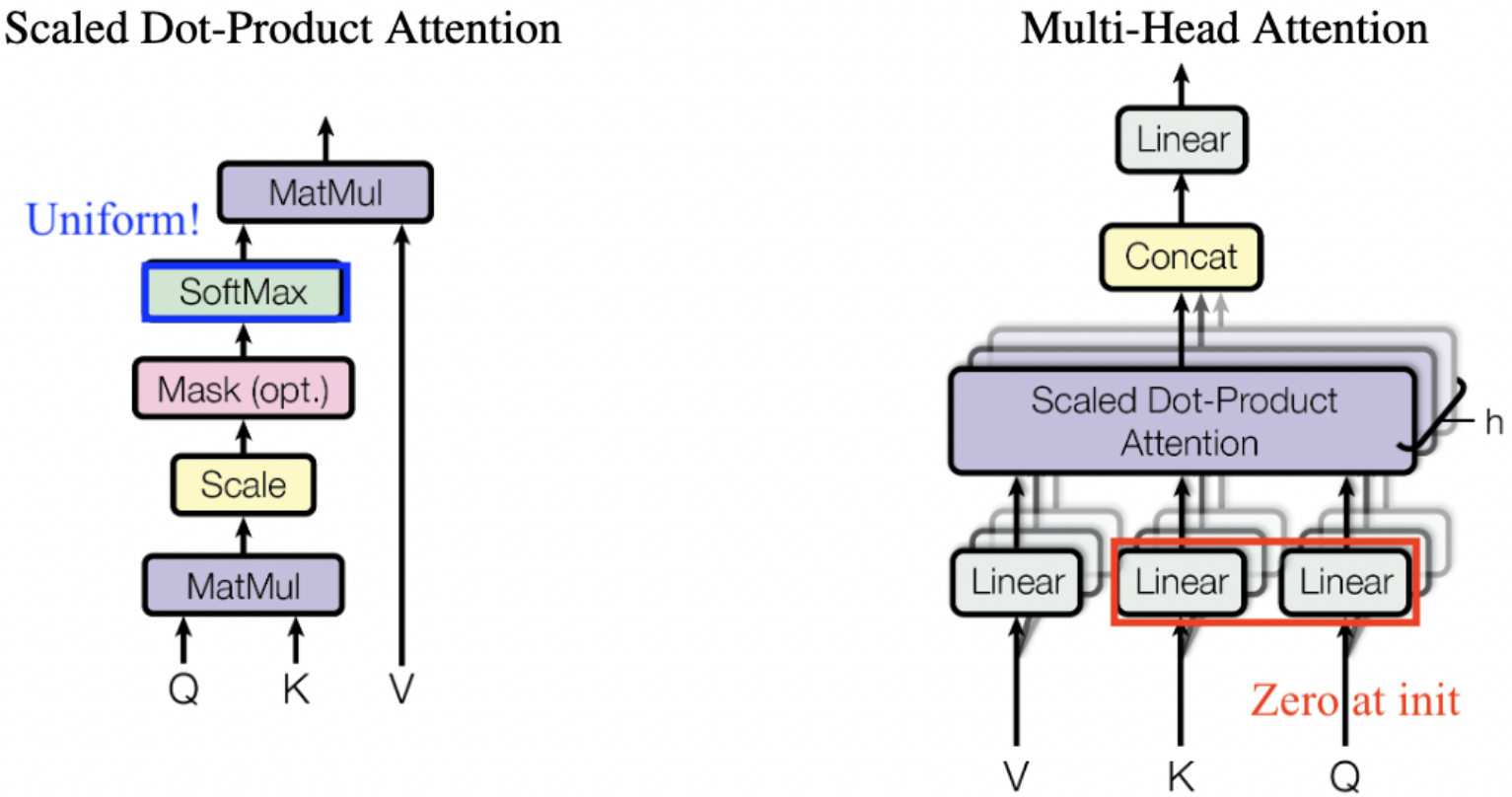
(a) Validation Loss (IWSLT) (b) BLEU (IWSLT) (c) Validation Loss (WMT) (d) BLEU (WMT)

Figure 4. Performances of the models on the IWSLT14 De-En task and WMT14 En-De task

Table of Contents

Assumptions for Theorem 1

Assumptions about the model at initialization:



$$\text{Softmax}(\vec{x})_i = \frac{e^{x_i}}{\sum_j e^{x_j}} \implies \text{Softmax}(\vec{0}) = (\frac{1}{d}, \frac{1}{d}, \ldots, \frac{1}{d})$$

Assumptions made for Theorem 1 (Cont.)

Assuming $W^Q, W^K = 0$:

$$Q = XW^Q = 0, \;\; K = XW^K = 0, \;\; V = XW^V$$

$$\begin{aligned} \text{Attention}(Q, K, V) &= \text{Softmax}\left(\frac{QK^T}{\sqrt{d}}\right) V \\ &= \text{Softmax}(\vec{0})V \\ &= \frac{1}{d}\sum_i V_i \end{aligned}$$

Attention module reduces to a linear layer.

Criticisms of Theorem 1

THM 1 says for Post-LN transformers

$$||\frac{\partial L}{\partial W^{2,L}}||_F \leq O(d\sqrt{\ln d})$$

- But this upper bound might not be tight.
- Doesn't explain why gradient explodes (norm increases), which is what they empirically observe.

Table of Contents

Our Experiments

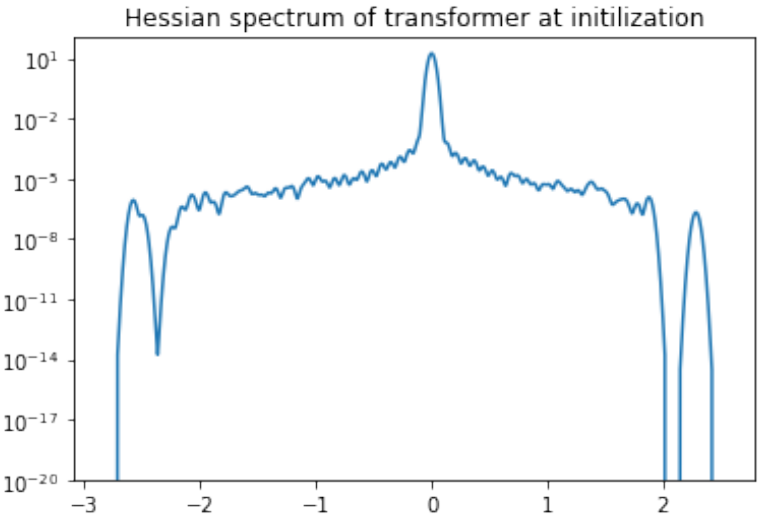
Experiment:

- Study hessian of Pre-LN and Post-LN transformers at initialization.

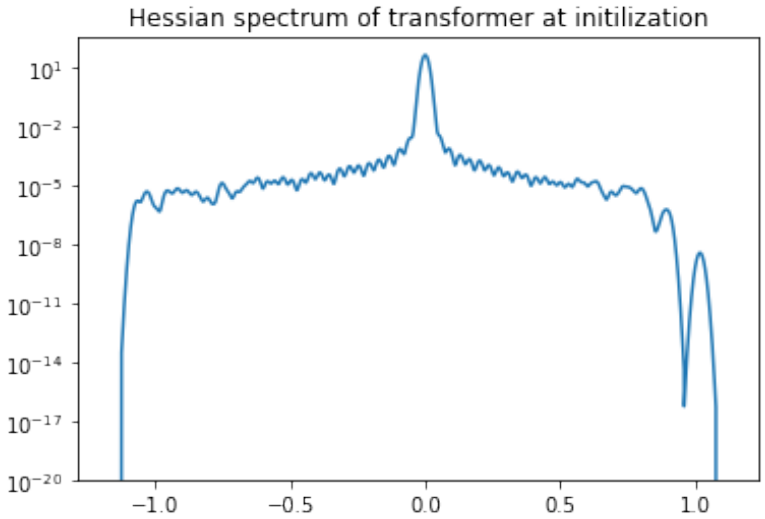
Experimental setting:

- 6-layer encoder and decoder.
- Xavier initialization.
- German to English translation.
- Multi30k (validation set), 1000 sentence pairs.
- Use Lanczos for hessian approximation.

Results



(a) Post-LN



(b) Pre-LN

Figure: Comparison of Hessian Spectrums

Conclusion: Loss surface of Post-LN transformer at initialization has more curvature than the Pre-LN transformer.

References

[1] *Attention is All You Need.* Vaswani et al., 2017.

[2] *On Layer Normalization in the Transformer Architecture.* Xiong et al., 2020.

[3] *Layer Normalization.* Ba et al., 2016.

[4] *Adam: A Method for Stochastic Optimization.* Kingma & Ba, 2015.

[5] *Deep Residual Learning for Image Recognition.* He et al., 2016.