Final Report: Normalizer-free Transformers

*** Kim (***-***) *** Han (***-***) *** *** (***-***) Rohitkumar Datchanamourty (***-***)

Abstract

The Transformer architecture has been successfully utilized in a variety of applications. However, optimizing complex Transformer models to their full potential has proven to be difficult, usually requiring the careful use of learning rate warmup schedulers in the process. In order to solve this issue, a recent line of work in the language domain allows Transformer models to train successfully without the use of normalization layers through careful initialization. In this project, we aim to expand this concept to encoder-only Transformer architectures and apply them to the vision domain. We experimentally prove that such a method enables normalizer-free Transformers to train successfully in extremely unstable settings. We emphasize the importance of this initialization step by showing the ineffectiveness of weight standardisation or adaptive gradient clipping when used alone.

1. Introduction

In the foundational work by (Vaswani et al., 2017), Transformers have become an architecture of choice across various domains, relying on layer normalization (LN) as their default normalization method. However, choosing LN lacks explicit justification from the authors. This trend extends into vision domain adaptations, exemplified by (Dosovitskiy et al., 2020), where LN is arbitrarily employed, despite the prevalence of alternative normalization methods in CNN-based models.

Previous works (Xiong et al., 2020; Huang et al., 2020) reveal that such normalization is most likely the reason why Transformer models are quite unstable at the early stages of training. Therefore the optimization of Transformers usually requires the use of additional measures such as learning rate warm-up, which introduces additional hyperparameters that need tuning while slowing down the whole training process.

In order to resolve this issue, there are several lines of work that modify the Transformer architecture, such as relocating the layer normalization block (Xiong et al., 2020) or removing layer normalization entirely by implementing careful initialization schemes (Huang et al., 2020; He et al., 2023). In this project, we would like to focus on the latter, as the effectiveness of such an approach has not yet been proven in the vision domain. Additionally, as normalizer-free CNNs (Brock et al., 2021) have proven to be effective for transfer learning, it would be interesting to see if the same could be said for normalizer-free Transformers.

In order to investigate the effectiveness of normalizer-free models and how various methods such as initialization, weight standardization, and adaptive gradient clipping affect the performance of the model, we run experiments on CIFAR-10 (Krizhevsky et al., 2009) using different variations of the NFNet (Brock et al., 2021) and ViT(Dosovitskiy et al., 2020). Through this, we show that measures like weight standardization or adaptive gradient clipping do not represent key components for building normalizer-free model whereas careful initialization appears like the critical foundation for it. We indeed verify that cautiously initialized normalizer-free Transformers are able to train on unstable environments, where their non-initialized counterparts fail to train successfully.

Extensively, experiments on larger scale datasets such as TinyImageNet let us confirm that, solely through careful initialization, normalizer-free models are able to train on higher learning rates. In the future, we plan to compare the effectiveness of different initialization methods as well as experiment with deeper/wider models.

2. Related Work

Transformers Transformers, introduced by (Vaswani et al., 2017) in 2017, revolutionized natural language processing and other various domains. Although originally used for natural language processing tasks, the self-attention mechanism makes the use of Transformer models effective for various tasks such as text generation and image recognition. Notably, Transformers have been successfully adapted for the vision domain (Dosovitskiy et al., 2020) and have been heavily utilized ever since.

Layer Normalization (LN) Being the default choice in Transformers architecture, its use lacks extensive justification. In vision tasks, as seen in (Dosovitskiy et al., 2020), LN is used without a clear rationale, despite alternative normalization schemes that are widely used in CNNs. Understanding the implications and trade-offs of using LN is a key aspect of our investigation.

Normalizer-free CNNs Normalizer-Free Networks (NFNets) (Brock et al., 2021) represent a significant advancement in normalizer-free CNN architecture design. Departing from batch normalization (BN), they achieve efficiency and performance gains over datasets such as ImageNet, while using fewer computational resources. This was realized through careful initialization, which mimics the behavior of BN, alongside weight standardization (WS) and adaptive gradient clipping (AGC).

NFNets employ stochastic depth, dynamically adjusting network depth during training for improved generalization. The novel activation function, hard-swish, balances nonlinearity and efficiency. Eliminating BN poses challenges related to internal covariate shifts. NFNets use AGC and sharpness-aware minimization to address these issues, ultimately stabilizing training. NFNets' efficiency showcases the potential of removing batch normalization without compromising performance.

Normalizer-free Transformers In light of the optimization problems caused by layer normalization in Transformers (Xiong et al., 2020; Huang et al., 2020), there has been a line of work that intends to solve this problem by entirely removing normalization layers from Transformers. Such normalizer-free Transformers were able to train thanks to careful initialization schemes by theoretically bounding the gradient at initialization (Huang et al., 2020) or constructing vanilla attention modules without shortcuts (He et al., 2023).

These approaches proved the effectiveness of normalizer-free Transformers in the natural language processing domain, and imply that a similar approach in other domains such as image processing could be feasible.

3. Main Method

The main focus of this project is to create a normalizer-free Transformer by either better initialization like (Huang et al., 2020) succeeded in doing for Machine Translation tasks or controlling the variance and mean-shift of the activations throughout the network similar to how normalizer-free CNNs were created (Brock et al., 2021).

3.1. Weight Initialization

As previous works regarding normalizer-free networks (Brock et al., 2021; Huang et al., 2020) reveal, the key to creating a functioning normalizer-free model is to initialize the model properly so that the model does not converge during the earliest steps of training. We base our initialization scheme on (Huang et al., 2020) in order to bound the parameter update, which is achieved through bounding the gradient respective to each parameter.

We formulate said goal using the following goal, similar to the one proposed in (Huang et al., 2020). For the Transformer $f(x;\theta)$ where x are the input to the model and θ are the learnable parameters, we want to bound the update to the model Δf to be $\Delta f = \Theta(\eta)$ for a given learning rate η. For the given loss function $\mathcal{L}$, Δf can be calculated as $f(x - \eta\frac{\partial \mathcal{L}}{x}; \theta - \frac{\partial \mathcal{L}}{\partial \theta})$ - $f(x;\theta)$.

To satisfy such conditions, we apply the following initialization scheme, which can be calculated using the Taylor expansion of the SGD update to Δf:

- Gaussian initialization $\mathcal{N}\left(0, d^{-\frac{1}{2}}\right)$ for the input embeddings and scale them by $(9N)^{-\frac{1}{4}}$, where d is the embedding dimension and N is the number of layers.
- Xavier initialization for the attention block and MLP block and scale them by $0.67N^{-\frac{1}{4}}$.

This initialization scheme is an adaptation of the T-Fixup initialization scheme (Huang et al., 2020), although it is to be noted that the encoder initialization was adapted since the ViT model only consists of Transformer encoder blocks.

3.2. Weight Standardization

Scaled weight standardization (Brock et al., 2021) is a technique employed to prevent the mean-shift of the activations when the normalization layers are removed from the model. In the original formulation for the convolution layers, the fan-in weights for a node were standardized as follows:

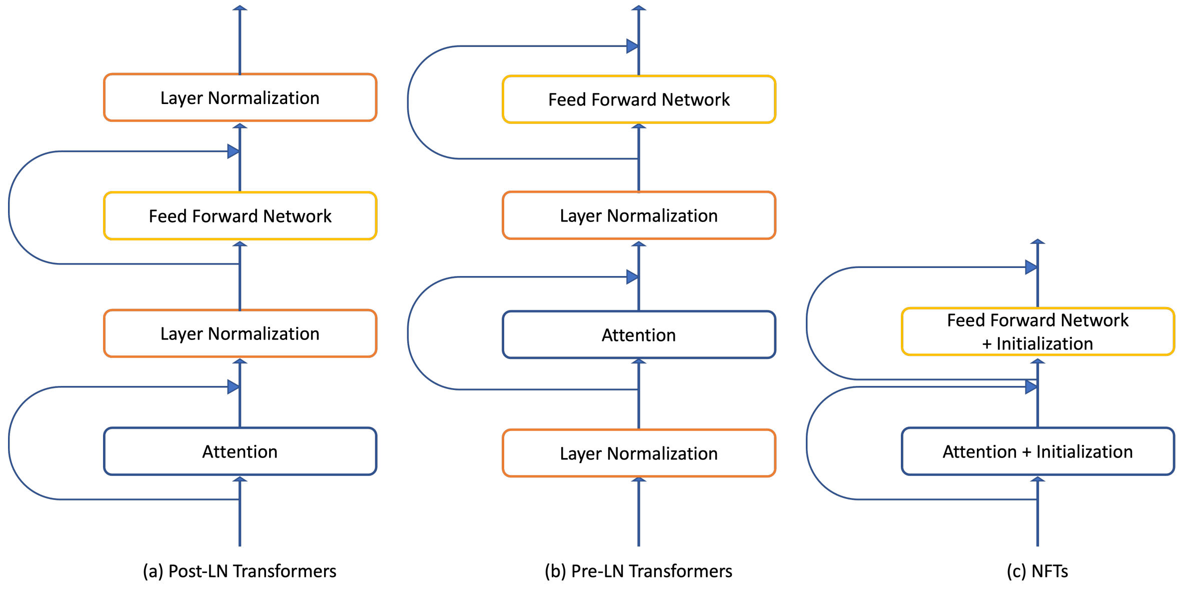
$$\hat{W_{ij}} = \frac{W_{ij} - \mu_i}{\sqrt{N}\sigma_i}, \tag{1}$$

where $\mu_i = (1/N)\sum_j W_{ij}$, $\sigma_i^2 = (1/N)\sum_j (W_{ij} - \mu_i)^2$, and N denotes the number of fan-in nodes. For our experiments, the denoted weight standardization was adapted to the attention mechanism as well as the feed-forward layers for the Transformer architecture.

3.3. Adaptive Gradient Clipping

Adaptive Gradient Clipping (AGC) (Brock et al., 2021) is an alternative gradient clipping method, intended for use in large learning-rate scenarios with poorly conditioned loss landscapes or training with large batch sizes. The regular gradient clipping method, which can be formulated as the following:

$$G \rightarrow \begin{cases} \lambda \frac{G}{\|G\|} & \text{if } \|G\| > \lambda, \\ G & \text{otherwise.} \end{cases} \tag{2}$$



(a) Post-LN Transformers

(b) Pre-LN Transformers

(c) NFTs

Figure 1. Different variants of the ViT model used for the experiments.

uses a static gradient threshold λ which is used for all of the possible learnable parameters, which may be unable to properly account for a suitable range of gradient for each of the parameters, and itself is a hyperparameter that needs to be tuned manually.

In order to resolve the issues mentioned, an automatic gradient clipping strategy using the Frobenius norm of the parameters was suggested, as the following:

$$G_i^\ell \to \begin{cases} \lambda \frac{\|W_i^\ell\|_F^\star}{\|G_i^\ell\|_F} G_i^\ell & \text{if } \frac{\|G_i^\ell\|_F}{\|W_i^\ell\|_F^\star} > \lambda, \\ G_i^\ell & \text{otherwise.} \end{cases} \tag{3}$$

where $\|W_i\|_F^\star = \max(\|W_i\|_F, \epsilon)$ for some small predetermined ϵ.

4. Experiments

4.1. Experimental Settings

Dataset The main dataset that was used for the experiments is the CIFAR-10/CIFAR-100 dataset (Krizhevsky et al., 2009). The train/validation/test data were split with a (45K:5K:1K) ratio, where the validation set was chosen randomly from the given training set, and the test set was given by the data provider. To assess our models' behaviour on larger datasets, we planned on ultimately run experiments on ImageNet. However, we decided to gradually scale our experiments and performed evaluation on TinyImageNet (https://huggingface.co/datasets/zh-plus/tiny-imagenet).

Training Details The input images were randomly cropped and/or flipped during training, and no additional augmentations were carried out. The image was not augmented during inference time. If not specified, we train models for 100 epochs and use a train batch size of 64. We use a stochastic gradient descent optimizer with a momentum of 0.9 and a weight decay of 2e-5.

4.2. Baselines

4.2.1. NFNet-F0

This experiment attempts to reproduce the results from the original authors of the NFNet model, (Brock et al., 2021). We use this baseline CNN normalizer-free model to compare its performance when tuning its parameters. Three experiments were conducted: (a) default NFNet-F0, (b) NFNet-F0 without adaptive gradient clipping, (c) NFNet-F0 without adaptive gradient clipping while re-implementing batch normalization. The baseline source code can be found through https://github.com/benjs/nfnets_pytorch/tree/master/nfnets.

4.2.2. ViT

As the main focus of our project, the original ViT model (Dosovitskiy et al., 2020) and the derivative ViT models were used for the experiments as the following, as depicted in Figure 1:

- Pre-LN (Xiong et al., 2020)
- Post-LN (Vaswani et al., 2017)
- No LN (by simply removing the LN layer)
- No LN + Weight Standardization (Brock et al., 2021)

Model	# of Params	Learning Rate	Valid Accuracy (%)
NFNet			
NFNet F0	68.44M	0.025	90.46
NFNet F0 − AGC	68.44M	0.025	90.08
NFNet F0 − AGC + BN	68.49M	0.025	**91.42**
ViT			
ViT-S Pre-LN	21.33M	0.1	NaN
ViT-S Pre-LN	21.33M	0.01	72.96
ViT-S Pre-LN	21.33M	0.001	71.62
ViT-S Pre-LN	21.33M	0.0001	55.46
ViT-S Post-LN	21.33M	0.1	NaN
ViT-S Post-LN	21.33M	0.01	**76.84**
ViT-S Post-LN	21.33M	0.001	**72.42**
ViT-S Post-LN	21.33M	0.0001	59.62
ViT-S − LN	21.31M	0.1	NaN
ViT-S − LN	21.31M	0.01	NaN
ViT-S − LN	21.31M	0.001	70.82
ViT-S − LN	21.31M	0.0001	**64.12**
ViT-S − LN + WS	21.34M	0.1	NaN
ViT-S − LN + WS	21.34M	0.01	NaN
ViT-S − LN + WS	21.34M	0.001	69.42
ViT-S − LN + WS	21.34M	0.0001	63.54
ViT-S − LN + WS + AGC	21.34M	0.1	NaN
ViT-S − LN + WS + AGC	21.34M	0.01	NaN
ViT-S − LN + WS + AGC	21.34M	0.001	69.42
ViT-S − LN + WS + AGC	21.34M	0.0001	63.54
ViT-S − LN + initialization	21.31M	0.1	NaN
ViT-S − LN + initialization	21.31M	0.01	74.36
ViT-S − LN + initialization	21.31M	0.001	70.34
ViT-S − LN + initialization	21.31M	0.0001	40.34

Table 1. Validation results on CIFAR-10. AGC stands for Adaptive Gradient Clipping, and WS stands for Weight Standardization, which are introduced in (Brock et al., 2021). BN and LN stand for Batch Normalization and Layer Normalization, respectively. Initialization refers to the initialization scheme documented in Section 3.

- No LN + Weight Standardization + Adaptive Gradient Clipping (Brock et al., 2021)

- No LN + Transformer Xavier initialization (Huang et al., 2020), A.K.A. Normalizer-free Transformer (NFT)

The weight standardization (WS) and adaptive gradient clipping (AGC) refer to the method proposed by (Brock et al., 2021), and Transformer Xavier initialization refers to the initialization strategy proposed by (Huang et al., 2020), as mentioned in the Main Method section.

While studying a training instability, inspired from Wortsman et al. (2023), we focus on studying learning rate stabilities. Wortsman et al. (2023) reveal that models are likely to diverge when training at high learning rates, as demonstrated by the relation between the magnitude of learning rate variation and validation losses. Other studies (Huang et al., 2020) establish that normalization layers help stabilize training. Thus, we intend to point out how to enhance training stability, regarding the learning rate stability as one proxy measure of the stability.

4.3. Results

4.3.1. NFNet Variants

In Table 1, we show the validation accuracies from NFNet and its variants. The results demonstrate that we succeeded in reproducing the results from NFNet; we observed no performance degradation when removing batch normalization from the model.

4.3.2. ViT for CIFAR-10

Table 1 also displays the performance of ViT variants. Here are our notable findings from these results: (1) simply adding WS or AGC to a normalizer-free model does not help the model train in harsh settings, as we observe training divergence for learning rates higher than 0.001. (2) Simply adding a proper initialization method without WS or AGC

Model	Learning Rate	Valid Accuracy (%)	Test Accuracy (%)
ViT-S	0.01	39.9	38.58
ViT-S	0.05	43.26	42.64
ViT-S	0.1	42.74	43.58
ViT-S-LN	0.01	41.68	42.54
ViT-S-LN	0.05	Diverge	Diverge
ViT-S-LN	0.1	Diverge	Diverge
ViT-S-LN+init	0.01	40.36	41.52
ViT-S-LN+init	0.05	44.38	44.7
ViT-S-LN+init	0.1	Diverge	Diverge

Table 2. Validation and test results on TinyImageNet. For every experiment, we used momentum SGD, Batch Size = 256, 100 epochs.

allows the model to train on more unstable settings, as we find the model still trainable with a high learning rate of 0.01. (3) Compared to normalization-free models, both Pre-LN and Post-LN models exhibit high learning rate stability, as our experiments indicate they are still trainable with a high learning rate of 0.01.

Although not included in table 1, experiments using normalizer-free ViT models with initialization and weight standardization were also conducted, however the models failed to train successfully and therefore the results were omitted from the table.

Through the experiments conducted for the CIFAR-10 dataset, it can be concluded that removing the normalization layer for the Transformer indeed harms the training stability of the model, and the stability can be (partially) recovered through a careful initialization scheme aimed at preventing the gradient from diverging at the beginning stages of training. However, the final performance of the model after being successfully trained showed some amounts of degradation even for NFTs (normalizer-free Transformer model with initialization) compared to the counterparts using normalization layers.

4.3.3. ViT for TinyImageNet

Table 2 shows the results for the experiments conducted using the TinyImageNet dataset. In terms of training stability, the ViT model with normalization layers performed the best as it was able to train successfully even for the highest learning rates. Compared to the model with normalization layers, the normalizer-free models performed worse as the model without LN was only able to train on the lowest learning rate setting. The normalizer-free model with initialization fares better compared to its randomly initialized counterpart, although the stability is still worse compared to the model with normalization layers.

One notable difference compared to the results from CIFAR-10 is that the normalizer-free models showed better performance compared to the model with normalizers when trained successfully, although it may be the result of random variance during the training procedure.

In conclusion, the experiments we have conducted so far make it hard to explicitly claim the superiority of normalizer-free ViT models compared to their normalized counterparts, as our normalizer-free model does not display enhanced stability or performance.

5. Conclusion

5.1. Discussion

The proposed work shows that normalizer-free models are possible under the condition that weights are initialized carefully. Indeed, this method allows the model to train on higher learning rates. This result seems to hold validity with larger datasets, such as TinyImageNet.

However, compared to the original Transformer model that utilizes normalization layers, NFTs showed some weaknesses in terms of training stability or performance, depending on the dataset. On the CIFAR-10 dataset, NFT lost several points of accuracy compared to the post-LN models, and on the TinyImageNet dataset, NFT showed slightly lower training stability as it failed to train on a high learning rate setting (LR = 0.1) compared to the pre-LN ViT model.

One additional noteworthy observation is that the ViT model without normalization but with initialization and weight standardization failed to train on the CIFAR-10 dataset, which is surprising considering that weight standardization was not harmful for CNN-based models. This may be the result of the weight standardization scheme capping the training capacity of the model when it was also applied to the attention layers. Therefore it would be interesting to see the results from a model where weight standardization is applied on a lower capacity, such as only applying weight standardization only for the activations of the feed-forward block of the Transformer architecture.

5.2. Future Work

Eventhough normalizer-free models are possible, it can be interesting to keep the usage of the layer normalization and focus on identifying superior normalization methods for enhanced performance. Our current leads include:

- Group Normalization (GN): Normalizes activations over groups of channels, differing from LN's individual channel normalization.
- Instance Normalization (IN): Normalizes activations over individual instances, common in generative models for preserving instance identity.
- Spectral Normalization (SN): Normalizes the spectral norm of a layer's weights, improving stability in models susceptible to mode collapse, such as GANs.
- Layer-Wise Adaptive Variance Scaling (LN-AVS): A modified LN that adaptively scales activation variance based on the layer index

We also denote that new normalization methods aiming for efficiency and scalability beyond LN, are of growing interest. To illustrate, Google AI researchers proposed Adaptive Batch Normalization (AdaBN), a faster alternative to LN with comparable performance across various tasks. Finally, experimenting with various initialization methods is a path to explore.

Even though in the experiments it was hard to prove the effectiveness of normalizer-free Transformers in the small-scale experiments conducted for this project, it would be promising to evaluate the performance of normalizer-free Transformers in a large-scale transfer learning setting. It has been shown in previous work (Brock et al., 2021) that normalizer-free CNNs have a higher transfer learning capability compared to their counterpart using normalization layers, as the normalizer-free network provides a higher capacity due to its lower parameter dependence due to the removal of the normalization layers.

Additional potential optimization to be hard for normalizer-free networks is the improvement to training speed. Although is was hard to observe for our experiments due to the relatively small training dataset, some works (He & Hofmann, 2023) report that removing the normalization layer had a small negative impact to the training speed of the model. The normalization layers may have a positive impact to the training speed of the model in terms of signal propagation standpoint in the earliest stages of training, it has also been reported that normalization may also have an impact beyond the ones explainable by signal propagation theory. It would be an interesting research direction to investigate this phenomenon for NFTs in the future.

6. Roles of Team Members

- **Yongho Kim**: Team leader, implementing baseline code, experimentation for CIFAR-10 and TinyImageNet, writing
- **Seungju Han**: Video presentation, implementing baseline codes (NF-Net), run experiments on CIFAR-10, writing
- **Thibaud Southiratn**: Writing - Draft & Review, experimentation for TinyImageNet
- **Rohitkumar Datchanamourty**: Implementing baseline code, experimentation on TinyImageNet, writing

References

Brock, A., De, S., Smith, S. L., and Simonyan, K. High-performance large-scale image recognition without normalization. In *International Conference on Machine Learning*, pp. 1059–1071. PMLR, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

He, B. and Hofmann, T. Simplifying transformer blocks. *arXiv preprint arXiv:2311.01906*, 2023.

He, B., Martens, J., Zhang, G., Botev, A., Brock, A., Smith, S. L., and Teh, Y. W. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. *arXiv preprint arXiv:2302.10322*, 2023.

Huang, X. S., Perez, F., Ba, J., and Volkovs, M. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pp. 4475–4483. PMLR, 2020.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Wortsman, M., Liu, P. J., Xiao, L., Everett, K., Alemi, A., Adlam, B., Co-Reyes, J. D., Gur, I., Kumar, A., Novak, R., et al. Small-scale proxies for large-scale transformer training instabilities. *arXiv preprint arXiv:2309.14322*, 2023.

Xiong, R., Yang, Y., He, D., Zheng, K., Zheng, S., Xing, C., Zhang, H., Lan, Y., Wang, L., and Liu, T. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.