A Neural ODE Interpretation of Transformer Layers

Anonymous Author(s)
Affiliation
Address
email

Abstract

Transformer layers, which use an alternating pattern of multi-head attention and multi-layer perceptron (MLP) layers, provide an effective tool for a variety of machine learning problems. As the transformer layers use residual connections to avoid the problem of vanishing gradients, they can be viewed as the numerical integration of a differential equation. In this extended abstract, we build upon this connection and propose a modification of the internal architecture of a transformer layer. The proposed model places the multi-head attention sublayer and the MLP sublayer parallel to each other. Our experiments show that this simple modification improves the performance of transformer networks in multiple tasks. Moreover, for the image classification task, we show that using neural ODE solvers with a sophisticated integration scheme further improves performance.

1 Introduction

Over the last few years, the transformer layer introduced by Vaswani et al. (2017) has become a key component in deep learning models used in natural language processing (Devlin et al., 2019; Brown et al., 2020; Shoeybi et al., 2019), image and video processing (Dosovitskiy et al., 2021; Arnab et al., 2021; Han et al., 2022), and audio and speech processing (Dong et al., 2018). Current state-of-the-art techniques in language processing (e.g., machine translation, natural language understanding, and information/knowledge extraction) rely heavily on the use of transformer layers to encode information in its word vector about the relevant context of a given word. This allows the model to focus on relevant contexts at different length scales. Although the transformer layers were originally introduced as a sequence-to-sequence transduction model, they have also demonstrated superior performance in various computer vision tasks beyond image classification, e.g., semantic segmentation (Zheng et al., 2021; Strudel et al., 2021; Ding et al., 2022), object detection (Carion et al., 2020; Song et al., 2022), and view synthesis (Kulhánek et al., 2022; Lin et al., 2022).

Inside a multilayer transformer network, each transformer layer consists of a multi-head attention sublayer followed by an MLP sublayer, creating an alternating pattern of these sublayers throughout the network. Moreover, both of these sublayers use residual connections to avoid the vanishing gradient problem and facilitate the training of very deep transformer networks. Prior work (Haber and Ruthotto, 2017; Haber et al., 2018; Lu et al., 2018) has shown that the forward propagation through residual connections can be viewed as Euler discretization of a time-varying ordinary differential equation (ODE). This insight suggests a connection between transformer networks and differential equations that can potentially be exploited to further improve the performance of transformer networks (e.g., higher accuracy, fewer parameters). Indeed, the residual connections alongside the alternating pattern of multi-head attention and MLP sublayers can be interpreted as numerical integration via the Lie-Trotter splitting scheme (Lu et al., 2020; Dutta et al., 2021). In this work, we show that the connection between transformers and ODEs can be leveraged to design a new architecture wherein the multi-head attention and MLP sublayers are placed parallelly, not sequentially, inside the individual

Submitted to the DLDE Workshop in the 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

transformer layers. Our experiments show that the proposed model performs better than the original transformer layer when tested on image classification, machine translation, and language modeling tasks.

A growing body of work has focused on improving transformer networks by reorganizing the sublayers and leveraging the connection between transformers and ODEs. Macaron Net (Lu et al., 2020) draws inspiration from numerical integration techniques to use a multi-head attention sublayer sandwiched between two MLP sublayers. Dutta et al. (2021) use a temporal evolution scheme to avoid the computationally expensive step of calculating dot-product attention at each transformer layer; instead, it computes the dot-product attention at the initial step and then time-evolves it through the layers. The proposed model is similar in spirit to this line of work - it modifies the internal architecture of the transformer layer and places the multi-head attention sublayer and the MLP sublayer side-by-side. On the other hand, Press et al. (2020) have explored the effect of reordering the individual sublayers and changing their numbers while keeping the total number of model parameters fixed. Their work shows that a transformer network can improve its performance by concentrating the multi-head attention and MLP layers in the lower and upper stages of the network, respectively; however, this performance improvement is not uniform across all tasks.

The main contributions of this work are as follows:

• By leveraging the connection between transformer layers and ODEs, we propose a novel variant of the transformer layer wherein the multi-head attention and MLP sublayers are placed side-by-side.

• Through numerical experiments, we demonstrate that the proposed model performs better than the original transformer layer across multiple tasks (image classification on CIFAR-100, machine translation on WMT-2014 English-German dataset, and language modeling on WikiText-103). Our experiments have been carried out with small models due to resource constraints.

• We also demonstrate that the performance of the proposed model can be further improved by using neural ODE solvers with sophisticated integration schemes (e.g., RK4).

2 Proposed Architecture

By letting $X^m := [x_1^m,\ x_2^m,\ \cdots,\ x_L^m]$ denote the input to the m-th transformer layer, the operation carried out by the multi-head attention sublayer can be expressed as

$$\hat{x}_i^m = x_i^m + G(x_i^m, X^m), \qquad 1 \le i \le L, \tag{1}$$

where L is the length of the input sequence and the function G represents the multi-head dot-product attention. $\hat{x}_i^m$, i.e., the output from this sublayer, is then fed to the MLP sublayer and undergoes the following transformation to yield $X^{m+1} = [x_1^{m+1},\ x_2^{m+1},\ \cdots,\ x_L^{m+1}]$

$$x_i^{m+1} = \hat{x}_i^m + F(\hat{x}_i^m), \qquad 1 \le i \le L, \tag{2}$$

where the function F represents the sequence of linear mappings and activation functions. As Lu et al. (2020) and Dutta et al. (2021) have highlighted, (1)-(2) can viewed as the numerical integration (over the time interval $[m, m+1]$) of the following ODE via the Lie-Trotter splitting scheme

$$\frac{dx_i}{dt} = F(x_i) + G(x_i, X), \tag{3}$$

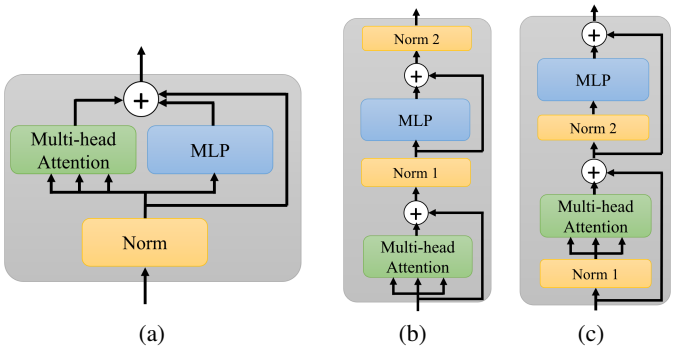


Figure 1: This figure shows the proposed model (left panel: a) along with the original version of the transformer layer (center panel: b) and the transformer layer used in the vision transformer (right panel: c).

where $X := [x_1,\ x_2,\ \cdots,\ x_L]$.

This interpretation paves the way for multiple approaches to realize transformer layers. As neural ODE solvers and their variants (Chen et al., 2018; Massaroli et al., 2020; Kidger et al., 2021) provide a means to run backpropagation through any black-box ODE solver, a transformer layer can be implemented using neural ODE networks. Alternatively, the time integration of (3) over the interval $[m, m+1]$ can also be approximated as

$$x_i^{m+1} = x_i^m + \Big[F(x_i^m) + G(x_i^m, X^m)\Big], \qquad 1 \leq i \leq L. \tag{4}$$

Our proposed model implements (4) by placing the multi-head attention and MLP sublayers side-by-side (Figure 1). Moreover, if they share their weights, a D-layer deep stack of transformer layers can be viewed as the numerical integration of (3) over the time interval $[0, D]$. This perspective offers a means to introduce weight sharing into a transformer network in a gradual way for understanding the trade-off between model performance and the number of model parameters. For example, a 12-layer transformer network can be replaced by a sequence of six 2-layer, weight sharing transformers (which can be implemented by either stacking two layers of the proposed model or via integration over a longer horizon using a neural ODE).

3 Experiments

In this section, we investigate the performance of the proposed model for both computer vision and language processing tasks. However, due to limited computing resources, we use smaller models for comparing the performance between the baseline transformer architecture and the proposed model. Therefore, the reported results of baselines may not be comparable to the state-of-the-art results.

3.1 Image Classification

First, we investigate the proposed model's performance in image classification tasks. In particular, we take the DeiT-Ti model proposed by Touvron et al. (2020) and modify the transformer layer architecture to implement our proposed model. Then, to compare the performance of DeiT-Ti and variants of our proposed models on CIFAR-100, we use top-1 accuracy as the metric.

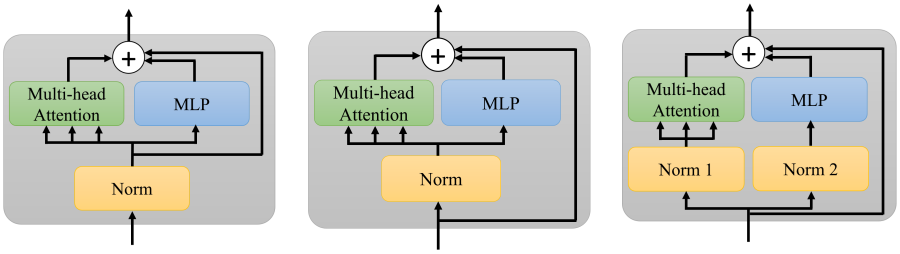
Table 1 demonstrates the performance of the side-by-side sublayers with different levels of weight sharing. We observe that by only changing the attention block and MLP block from sequential to parallel, the top-1 accuracy increases from 66.02% to 70.92%. This architecture does not use any weight sharing. Increasing the amount of weight sharing reduces the number of independent layers and results in fewer trainable parameters. However, as the number of parameters decreases, the model performance also deteriorates.

By leveraging the connection between differential equations and the transformer layers, we also replace the Euler scheme with a more sophisticated Runge-Kutta (RK4) integration scheme. This further increases the top-1 accuracy from 70.92% (Euler) to 72.66% (RK4).

Furthermore, we study the effects of dropout (Srivastava et al., 2014) and stochastic depth (Huang et al., 2016) in Table 2. We conclude that neither dropout nor stochastic depth is helpful in the proposed architecture.

Table 1: Performance on CIFAR-100 classification task of our proposed model with different levels of weight sharing. All the experiments are done without dropout and stochastic depth. **Top two rows**: 12 independent layers mean there's no weight sharing. the only difference between proposed architecture and DeiT-Ti is indicated in Figure 1; **Bottom row**: 1 independent layer means sharing weights for all 12 layers.

Model	# layers	# independent layers	# parameters	Top-1 accuracy
DeiT-Ti	12	12	5.5M	66.02%
Proposed model	12	12	5.5M	70.92%
	12	6	2.9M	67.42%
	12	4	2.0M	66.40%
	12	3	1.5M	63.68%
	12	2	1.1M	61.05%
	12	1	0.7M	53.33%



(a) (b) (c)

Figure 2: Variations of normalization implementations in the proposed model. In the image classification task, they lead to different top-1 accuracy: (a) 70.92%; (b) 67.42%; (c) 67.03%, all of which beat DeiT-Ti (66.02%).

Table 2: Performance on CIFAR-100 classification task of our proposed model. Neither dropout nor stochastic depth benefits training for our proposed model.

Dropout	Stochastic depth	Top-1 accuracy (12 ind. layers)	Top-1 accuracy (1 ind. layers)
✓	✗	59.81%	40.71%
✗	✓	71.49%	27.57%
✗	✗	70.92%	53.33%

We also conducted a small ablation study on the placement of the normalization layer in our proposed model. Figure 2 shows that all three variations perform better than the original DeiT-Ti model. Moreover, our proposed normalization approach (Figure 2a) yields the best performance.

3.2 Natural Language Processing

We leverage the open source toolkit Fairseq (Ott et al., 2019) to investigate the performance of the proposed model in neural machine translation and language modeling. Additional details about the size of the architectures used in this section are provided in the Appendix.

3.2.1 Neural Machine Translation

To train small transformer models on the WMT-2014 English-German translation dataset, we follow the training procedure specified by Ott et al. (2018, 2019). After training the models for 30 epochs, we compare their performance by computing the validation loss and detokenized BLEU score with SacreBLEU (Post, 2018) as described by Ott et al. (2018). As shown in Table 3, the proposed model outperforms the baseline.

Table 3: Performance on Neural Machine Translation

Model	**Dropout**	**Validation loss**	**BLEU score**
Sequential	0.0	3.788	17.1
	0.1	3.455	18.7
Parallel	0.0	3.626	17.9
	0.1	3.519	18.5

3.2.2 Neural Language Modeling

We follow the training and evaluation procedure specified by Baevski and Auli (2019) and Ott et al. (2019) to train a small language model on the WikiText-103 dataset. Table 4 reports the metric of perplexity for the baseline and proposed model for different dropout rates. We can notice that for these small models, dropout doesn't lead to improved performance for both the baseline and the proposed architecture. With zero dropout, our proposed architecture performs better than the baseline.

Table 4: Performance on Neural Language Modeling

Model	**Dropout**	**Perplexity**
Sequential	0.0	65.07
	0.1	72.72
	0.3	101.09
Parallel	0.0	60.19
	0.1	76.21
	0.3	113.27

4 Conclusion

This work has proposed a new variant of the transformer layer by highlighting the connection between transformer layers and ODEs. The proposed model outperforms the baseline. Furthermore, as shown by our initial results from an RK4-based neural ODE solver, one can extend this work to investigate the potential of using a time-dependent neural ODE to implement transformer networks.

References

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., and Schmid, C. (2021). ViViT: A video vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*.

Baevski, A. and Auli, M. (2019). Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. (2020). End-to-end object detection with transformers. In *European Conference on Computer Vision (ECCV)*.

Chen, R. T., Rubanova, Y., Bettencourt, J., and Duvenaud, D. K. (2018). Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, pages 6571–6583.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Ding, M., Xiao, B., Codella, N., Luo, P., Wang, J., and Yuan, L. (2022). DaViT: Dual attention vision transformer. In *European Conference on Computer Vision (ECCV)*.

Dong, L., Xu, S., and Xu, B. (2018). Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5884–5888.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

Dutta, S., Gautam, T., Chakrabarti, S., and Chakraborty, T. (2021). Redesigning the transformer architecture with insights from multi-particle dynamical systems. In *Advances in Neural Information Processing Systems*, volume 34, pages 5531–5544.

Haber, E. and Ruthotto, L. (2017). Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004.

Haber, E., Ruthotto, L., Holtham, E., and Jun, S.-H. (2018). Learning across scales—Multiscale methods for convolution neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32.

Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., Yang, Z., Zhang, Y., and Tao, D. (2022). A survey on vision transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Huang, G., Sun, Y., Liu, Z., Sedra, D., and Weinberger, K. Q. (2016). Deep networks with stochastic depth. In *European conference on computer vision*. Springer.

Kidger, P., Chen, R. T. Q., and Lyons, T. J. (2021). "Hey, that's not an ODE": Faster ODE adjoints via seminorms. In *Proceedings of the 38th International Conference on Machine Learning*, pages 5443–5452.

Kulhánek, J., Derner, E., Sattler, T., and Babuška, R. (2022). Viewformer: Nerf-free neural rendering from few images using transformers. In *European Conference on Computer Vision (ECCV)*.

Lin, K.-E., Yen-Chen, L., Lai, W.-S., Lin, T.-Y., Shih, Y.-C., and Ramamoorthi, R. (2022). Vision transformer for nerf-based view synthesis from a single input image. *arXiv:2207.05736*.

Lu, Y., Li, Z., He, D., Sun, Z., Dong, B., Qin, T., Wang, L., and Liu, T.-y. (2020). Understanding and improving transformer from a multi-particle dynamic system point of view. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*.

Lu, Y., Zhong, A., Li, Q., and Dong, B. (2018). Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In *Proceedings of the 35th International Conference on Machine Learning*, pages 3276–3285.

Massaroli, S., Poli, M., Park, J., Yamashita, A., and Asama, H. (2020). Dissecting Neural ODEs. In *Advances in Neural Information Processing Systems*, volume 33.

Ott, M., Edunov, S., Baevski, A., Fan, A., Gross, S., Ng, N., Grangier, D., and Auli, M. (2019). fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Ott, M., Edunov, S., Grangier, D., and Auli, M. (2018). Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium.

Post, M. (2018). A call for clarity in reporting BLEU scores. *arXiv:1804.08771*.

Press, O., Smith, N. A., and Levy, O. (2020). Improving transformer models by reordering their sublayers. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2996–3005.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. (2019). Megatron-LM: Training multi-billion parameter language models using model parallelism.

Song, H., Sun, D., Chun, S., Jampani, V., Han, D., Heo, B., Kim, W., and Yang, M.-H. (2022). ViDT: An efficient and effective fully transformer-based object detector. In *International Conference on Learning Representations*.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Strudel, R., Garcia, R., Laptev, I., and Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7262–7272.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. (2020). Training data-efficient image transformers & distillation through attention. *arXiv:2012.12877*.

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

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., and Zhang, L. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Appendices

A Model Size Used in the Neural Machine Translation Task

Table 5: Only the difference between the base model and the small model is listed. We use the small model size to demonstrate performance difference between sequential and parallel blocks inside the transformer layers.

	Base	Small
`encoder_embed_dim`	512	128
`encoder_ffn_embed_dim`	2048	512
`encoder_attention_heads`	8	2
`decoder_attention_heads`	8	2

B Model Size Used in the Neural Language Modeling Task

Table 6: Only the difference between the base model and the small model is listed. We use the small model size to demonstrate performance difference between sequential and parallel blocks inside the transformer layers.

	Base	Small
`decoder_embed_dim`	512	128
`decoder_ffn_embed_dim`	2048	512
`decoder_attention_heads`	8	2