RETHINKING EFFICACY OF SOFTMAX
FOR LIGHTWEIGHT NON-LOCAL NEURAL NETWORKS

Yooshin Cho[1], Youngsoo Kim[1], Hanbyel Cho[1], Jaesung Ahn[2], Hyeong Gwon Hong[2], and Junmo Kim[1,2]

[1]School of Electrical Engineering, KAIST, South Korea
[2]Kim Jaechul Graduate School of AI, KAIST, South Korea

ABSTRACT

Non-local (NL) block is a popular module that demonstrates the capability to model global contexts. However, NL block generally has heavy computation and memory costs, so it is impractical to apply the block to high-resolution feature maps. In this paper, to investigate the efficacy of NL block, we empirically analyze if the magnitude and direction of input feature vectors properly affect the attention between vectors. The results show the inefficacy of *softmax* operation which is generally used to normalize the attention map of the NL block. Attention maps normalized with *softmax* operation highly rely upon magnitude of key vectors, and performance is degenerated if the magnitude information is removed. By replacing *softmax* operation with the scaling factor, we demonstrate improved performance on CIFAR-10, CIFAR-100, and Tiny-ImageNet. In Addition, our method shows robustness to embedding channel reduction and embedding weight initialization. Notably, our method makes multi-head attention employable without additional computational cost.

Index Terms— Attention, Non-local block, Transformer

1. INTRODUCTION

Self-attention layers such as Non-Local (NL) block [1] and Transformer [2] were proposed to capture long-term dependencies, and considered as a key component in Natural Language Process (NLP) deep learning architectures [3, 4, 5, 6]. To capture global features, self-attention layers model relationship between pixels regardless of distance. This property benefits not only machine translation, but also most computer vision tasks. However, NL blocks have been employed in a limited manner in computer vision owing to their heavy computation and memory cost that increases as a quadratic function of the number of pixels. Generally the number of pixels is much larger than the number of words, and thus the cost is not scalable to realistic input image sizes.

Evidently, reducing the cost of NL blocks is still an active research area [7, 8, 9, 4, 10, 11, 12]. Previous studies have focused on introducing lightweight NL blocks and methods to efficiently employ NL blocks. They suggested lightweight NL blocks by efficiently reducing spatial size [7, 9, 13] and approximating the attention while minimizing the loss of the capability to capture long-term dependencies. To optimize the trade-off between the capability to obtain global relationships and computational efficiency, previous methods have relied on heuristic, adopted approximation or neural architecture search (NAS) algorithms [14, 15, 8]. These methods demonstrated plausible performance and reduced computational overhead, but reducing the spatial size and the number of NL block cannot avoid the loss of the capability to incorporate global context.

In this paper, we empirically analyze the efficacy of *softmax* operation of NL blocks using the geometric definition of the dot product. In most cases, attention is computed using the dot-product and normalized with *softmax* operation [1, 2, 4, 7, 9, 3, 13, 6]. Geometrically, dot-product is a multiplication of magnitudes and cosine similarity between two pixel vectors. From this perspective, we suspect that *softmax* operation makes modeling relationship using cosine similarity inefficient for the following reason. To focus on angular relationships, let's assume that query and key vectors have a unit norm. Then, if *softmax* operation is employed to normalize attention map, attention between query and key is minimized when dot product is -1. Thus, for a single key vector to have the minimum attention with more than two queries, those queries should have the same direction, reducing angular variation of queries. However, if attention is not normalized by *softmax* operation, extremely low attention can be expressed by orthogonality between queries and keys. For a single key vector, queries with zero attention can be diversely selected in its hyperplane of dimension $C - 1$ that is orthogonal to the key vector, where C is the channel size of queries and keys. Hence, we suspect *softmax* operation might limit the capability to model relationships, and make NL block dependent on magnitude rather than direction of vectors.

To verify our assumption, we train PreResNet [16] with NL blocks on CIFAR-10/100 and Tiny-ImageNet [17, 18], and visualize the attention maps of NL block with randomly sampled images in Figure 1. Attention maps are matrices of size $HW \times HW$ which are computed by Eq 1. Attention maps of NL block demonstrate clear vertical lines; it indicates that attention value rarely changes despite varying query, and attention is dominantly affected by keys itself (e.g., magnitude). In other word, cosine similarity is less discriminatively

| (a) CIFAR-10 / NL | (b) CIFAR-100 / NL | (c) Tiny-ImageNet / NL |
| (d) CIFAR-10 / Ours | (e) CIFAR-100 / Ours | (f) Tiny-ImageNet / Ours |

Fig. 1: Visualization of the attention maps of randomly sampled images. Attention maps are size of $HW \times HW$, where the X-axis and Y-axis are key and query pixels, respectively. Attention maps of NL block are clearly affected by few key pixels, and vertical lines are observed. By contrast, attention maps of our method does not demonstrate a vertical line.

trained as we suspected. Therefore, we introduce the *scaled NL block* that does not employ *softmax* operation, and it will be described in Section 3.1. As illustrated in Figure 1, our method does not demonstrate a straight vertical line, and it indicates that attention depends on both queries and keys.

Scaled NL block shows two beneficial properties owing to our method properly utilizes the direction of feature vectors. First, *scaled NL block* shows robustness to embedding channel reduction. Because the proposed method efficiently utilizes the embedding feature space, performance degradation due to embedding channel reduction is significantly smaller than for the NL block. Second, *scaled NL block* demonstrates robustness to embedding weight initialization. NL block performs better when the weight of the embedding layer is initialized with a standard deviation of 0.01, which is tuned hyperparameter. By contrast, proposed method is suitable to *He initialization* [19] which is the standard initialization method. In addition, we generally obtain better performance on Pre-ResNet [16] and Wide-Residual Network (WRN) [20] with CIFAR-10/100 [17], and Tiny ImageNet [18]. Finally, we investigate the memory consumption and train step time of multi-head attention of NL blocks. The memory consumption and train step time of NL block are linear functions of the number of heads. By contrast, our method makes multi-head attention adoptable without additional computation cost.

2. RELATED WORKS

The monumental attention layer, Transformer [2] achieved the best performance at the time on machine translation tasks based solely on attention mechanisms. Wang *et al.*[1] employed attention mechanisms in computer vision applications to incorporate the global spatio-temporal context. Since their success, these self-attention layers have been widely used to model long-range relationships in variou applications [4, 3, 7, 10, 8, 9, 13]. Self-attention layers can be expressed generally using the following formula:

$$A_{i,j} = \frac{1}{Z(\mathbf{x})} f(\boldsymbol{x_i}, \boldsymbol{x_j}), \quad (1)$$

$$\boldsymbol{y_i} = \sum_{\forall j} \boldsymbol{A_{i,j}} g(\boldsymbol{x_j}), \quad (2)$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{HW \times C}$ are the input and output of the NL block, and i, j are the indices of the query and key pixels. $\boldsymbol{A} \in \mathbb{R}^{HW \times HW}$ is the attention map. In most cases, NL block takes the form $f(\boldsymbol{x_i}, \boldsymbol{x_j}) = e^{\frac{1}{\sqrt{C}} \theta(\boldsymbol{x_i}) \cdot \phi(\boldsymbol{x_j})}$ and $Z(\boldsymbol{x}) = \sum_{\forall j} f(\boldsymbol{x_i}, \boldsymbol{x_j})$, where θ, ϕ, g are the linear embedding layers. In this case, NL block gets attention from the embedded dot product that normalized with *softmax* operation.

3. GEOMETRICAL ANALYSIS

3.1. Scaled Non-Local Block

Previous studies [10, 8, 11, 12] have suggested attention modules without *softmax* operation for improved computational efficiency. However, the inefficacy of *softmax* operation has not been fully analyzed yet. To empirically analyze the inefficacy of *softmax* operation, we introduce the NL block without *softmax* operation. Instead of *softmax* operation, we divide the output of NL block by $\sqrt{HW}$ to stabilize the block, where H, W are the height and width of the input matrix, respectively[1]. We empirically verify that the proposed method

---

[1]Assume that elements of $g(x)$ and $A = \frac{1}{\sqrt{C}}\theta(\boldsymbol{x}) \cdot \phi(\boldsymbol{x})^\top$ are independent gaussian random variables with mean 0 and variance 1. Then, $A \cdot g(x)$ has mean 0 and variance HW. Thus, we scaled the output by $\sqrt{HW}$.

without scaling factor is often diverging, but scaling successfully prevents divergence. In this paper, we denote the block as *scaled Non-local Block*. If *softmax* operation is replaced with the scaling factor, Eq 2 can be expressed as follows by employing the associative law:

$$
\begin{aligned}
\boldsymbol{y} &= \frac{1}{\sqrt{HW}}(\frac{1}{\sqrt{C}}\theta(\boldsymbol{x}) \cdot \phi(\boldsymbol{x})^{\top}) \cdot g(\boldsymbol{x}) \\
&= \frac{1}{\sqrt{HWC}}\theta(\boldsymbol{x}) \cdot (\phi(\boldsymbol{x})^{\top} \cdot g(\boldsymbol{x})),
\end{aligned} \quad (3)
$$

where θ, ϕ, and g are linear embedding layers. As suggested in [10, 8], it can largely reduces the computational cost by employing the associative law, even two forms are numerically equivalent. In the following sections, we compare the properties of NL block and *scaled NL block* to demonstrate the inefficacy of *softmax* operation.

3.2. Importance Analysis

As mentioned earlier, we suspect that *softmax* operation limits the capability to model relationships between vectors, because it reduces the angular variations of query vectors having zero attention to a single key vector. For this reason, we assume that the cosine similarity terms of the dot-product is inefficiently learned, and attention maps highly rely on the magnitude of key vectors. To verify our assumption, we illustrate attention maps of the NL block in Figure 1, which demonstrate clear vertical lines. This indicates that attention map are highly affected by the magnitude of key vectors. By contrast, attention maps of *scaled NL block* do not show vertical lines. For further investigation, we train PreResNet with a *magnitude only NL block* and a *direction only NL block*, which are respectively expressed by the following formulas:

$$
\theta_{mag}(\boldsymbol{x_i}) = \|\theta(\boldsymbol{x_i})\|, \quad \phi_{mag}(\boldsymbol{x_i}) = \|\phi(\boldsymbol{x_i})\|, \quad (4)
$$

$$
\theta_{dir}(\boldsymbol{x_i}) = \frac{\theta(\boldsymbol{x_i})}{\|\theta(\boldsymbol{x_i})\|}, \quad \phi_{dir}(\boldsymbol{x_i}) = \frac{\phi(\boldsymbol{x_i})}{\|\phi(\boldsymbol{x_i})\|}. \quad (5)
$$

To verify whether NL blocks properly utilize the cosine similarity information, we replace the $\{\theta, \phi\}$ of Eq 2 with $\{\theta_{mag}, \phi_{mag}\}$ or $\{\theta_{dir}, \phi_{dir}\}$. As shown in Table 1, the performance of *direction only NL block* is severely worse than for the *direction only scaled NL block*. By constrast, *magnitude only scaled NL block* demonstrates comparable performance with *magnitude only NL block*. This indicates that by replacing $softmax$ operation to scaling factor, the capability to utilize angular information is improved while the capability to utilize magnitude information is maintained.

3.3. Robustness

In this section, we demonstrate the advantages of our method. As confirmed in Section 3.2, attention without *softmax* operation is more likely to learn angular relationships. Hence, we assume our method has the capability to efficiently represent

Dataset	Model	Base	Mag	Dir
CIFAR-10	PreResNet56+3NL	**5.73**	<u>5.83</u>	6.01
	PreResNet56+3Ours	**5.64**	5.76	<u>5.67</u>
CIFAR-100	PreResNet56+3NL	**25.12**	<u>25.26</u>	25.44
	PreResNet56+3Ours	**24.53**	25.20	<u>24.68</u>

Table 1: Comparison of test errors(%) on PreResNet56 with CIFAR-10/100. Results are averaged over 10 random seeds. Base refers to NL block utilizing both magnitude and direction of vectors.

(a) CIFAR-10   (b) CIFAR-100

Fig. 2: Illustration of test errors(%) with respect to embedding channel size. We train PreResNet56 by varying the NL blocks on CIFAR-10/100. Both (a) and (b) show that our method improves robustness to embedding channel reduction.

relationships. We verify this by checking the performance as reducing the embedding channel dimension, and obtain the expected results. We train PreResNet56 with 3 NL blocks inserted to the second residual block. As illustrated in Figure 2, our method demonstrates robustness to embedding channel reduction.

Our method also demonstrates robustness to embedding weight initialization. We compare the performance with three initialization methods: *He initialization* [19] and initialization with standard deviation of $\{0.0, 0.01\}$. As shown in Table 2, the weight of NL block should be initialized with a standard deviation of 0.01, which is suggested in [1]. It is tuned magic number, and much smaller than the standard deviation of the *He initialization*. By contrast, our method shows the best performance with the standard *He initialization*.

Dataset	Model	Initialization		
		He [19]	$\sigma = 0.01$	$\sigma = 0.0$
CIFAR-10	PreResNet56+3NL	5.91	**5.73**	6.23
	PreResNet56+3Ours	**5.64**	5.69	6.14
CIFAR-100	PreResNet56+3NL	25.82	**25.12**	26.24
	PreResNet56+3Ours	**24.53**	24.62	26.24
Tiny-ImageNet	PreResNet50+3NL	35.84	**35.35**	37.5
	PreResNet50+3Ours	**35.08**	35.34	37.23

Table 2: Comparison of test errors(%) on PreResNet50 and PreResNet56 with CIFAR-10/100 and Tiny-ImageNet. Results of CIFAR and Tiny-ImageNet are averaged over 10 and 3 random seeds, respectively.

	Methods	Number of heads			
		0	1	2	4
Test Error @ CIFAR-10 [%]	NL	6.37	5.73	5.69	5.62
	Ours		**5.64**	**5.48**	**5.43**
Test Error @ CIFAR-100 [%]	NL	26.59	25.12	24.63	24.43
	Ours		**24.53**	**24.17**	**24.12**
Memory [MB]	NL	2827	3789	4269	5229
	Ours		**3453**	**3429**	**3417**
Train Step Time [ms/batch]	NL	62.37	83.11	88.96	100.93
	Ours		**77.92**	**76.97**	**76.51**

Table 3: Comparison of test errors(%), memory(MB), and train step time(ms/batch) on PreResNet56 with CIFAR-10/100. 3 NL blocks are inserted, and results of NL block with 0 head is obtained on PreResNet56 without NL block. Test errors are averaged over 10 random seeds, and train step times are averaged over 300 iterations.

Dataset	Model	NL	Ours
CIFAR-10	PreResNet32	6.81	**6.65**
	PreResNet56	5.66	**5.43**
	PreResNet110	5.29	**4.93**
CIFAR-100	PreResNet32	29.89	**28.84**
	PreResNet56	24.33	**24.12**
	PreResNet110	23.28	**22.62**
	WRN-28-10	18.51	**18.18**
Tiny-ImageNet	PreResNet50	34.76	**34.385**

Table 4: Comparison of test errors(%) on PreResNet32, PreResNet50, and PreResNet56 with CIFAR-10/100 and Tiny-ImageNet. 3 NL blocks with 4 heads are inserted. The Results of CIFAR and Tiny-ImageNet are averaged over 10 and 3 random seeds, respectively.

4. EXPERIMENTS

In this section, we describe details of experiments. We insert 3 NL blocks with 4 heads to the second residual block of PreResNet and Wide-Residual Networks (WRN). To investigate the inefficacy of *softmax* operation, we conduct experiments by varying the NL blocks on CIFAR-10, CIFAR-100, and Tiny-ImageNet.

For CIFAR datasets, we train networks with 50k training images using the standard data augmentation, and evaluate the top-1 errors on 10k test images. We employ SGD with a mini-batch size of 128. Momentum and weight decay are set to 0.9 and 1e-4, respectively. Learning rate is initially set to 0.1, and divided by 10 at 81 and 122 epochs. Training is stopped at 164 epochs.

For Tiny-ImageNet, we train with the 100k training images using the standard data augmentation with 56 pixels cropping, and evaluate the top-1 error on 10k test images. First, we pretrain PreResNet without NL block on Tiny-ImageNet with SGD and mini-batch size of 128. Momentum and weight decay are set to 0.9 and 1e-4, respectively. Learning rate is initially set to 0.1, and divided by 10 every 30 epochs. Training is stopped at 100 epochs. Then, we insert

NL blocks to the pretrained networks, and fine-tune. We set the initial learning rate to 0.01, divide it by 10 at 40 epochs, and finish the training at 60 epochs.

As shown in Table 4, we obtain improved performance on PreResNet and WRN with CIFAR-10/100 and Tiny-ImageNet. We constantly get better results regardless of the depth or width of the networks. We get 0.27% and 0.66% accuracy improvement on PreResNet56 with CIFAR-10 and CIFAR-100, respectively. Additionally, as shown in Table 3, our method reduces computational cost by removing *softmax* operation and employing the associative law. As suggested in [2], we set the embedding channel size to C/N_h, where C and N_h are the channel size of input features and number of heads, respectively. Notably, our method can employ multi-head attention without additional computation cost, because the computation cost of our method is complexity of $HW(C/N_h)^2 \times N_h$

5. CONCLUSION

In this paper, we investigate the way in which attention maps can be calculated. We empirically analyze the inefficacy of *softmax* operation and superiority of *scaled NL block*. We visualize the attention maps and compare the performance of the *magnitude only NL block* and *direction only NL block* to verify that *softmax* operation makes the attention strongly rely on the magnitude of key vectors. By contrast, our method is more efficiently learn angular relationships using the cosine similarity. Our method demonstrates robustness to embedding channel reduction and embedding weight initialization. In addition, our method generally improves the performance with PreResNet and WRN on CIFAR-10/100 and Tiny-ImageNet. Notably, by employing the associative law, the computational cost of our method is largely reduced to a linear function of the number of pixels, and our method can employ multi-head attention without additional cost.

6. REFERENCES

[1] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[4] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[5] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma, "Linformer: Self-attention with linear complexity," *arXiv preprint arXiv:2006.04768*, 2020.

[6] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[7] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu, "Ccnet: Crisscross attention for semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 603–612.

[8] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L Yuille, "Neural architecture search for lightweight non-local networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10297–10306.

[9] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens, "Scaling local self-attention for parameter efficient visual backbones," *arXiv preprint arXiv:2103.12731*, 2021.

[10] Hila Levi and Shimon Ullman, "Efficient coarse-to-fine non-local module for the detection of small objects," *arXiv preprint arXiv:1811.12152*, 2018.

[11] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang, "Soft: Softmax-free transformer with linear complexity," *Advances in Neural Information Processing Systems*, vol. 34, 2021.

[12] Oliver Richter and Roger Wattenhofer, "Normalized attention without probability cage," *arXiv preprint arXiv:2005.09561*, 2020.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[14] Dimitrios Stamoulis, Ruizhou Ding, Di Wang, Dimitrios Lymberopoulos, Bodhi Priyantha, Jie Liu, and Diana Marculescu, "Single-path nas: Designing hardware-efficient convnets in less than 4 hours," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2019, pp. 481–497.

[15] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le, "Mnasnet: Platform-aware neural architecture search for mobile," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2820–2828.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[17] Alex Krizhevsky, Geoffrey Hinton, et al., "Learning multiple layers of features from tiny images," 2009.

[18] Ya Le and Xuan Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 3, 2015.

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[20] Sergey Zagoruyko and Nikos Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.