ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

CRE: Circle relationship embedding of patches in vision transformer

Zhengyang Yu[1,2] and Jochen Triesch[1] *

1- Frankfurt Institute for Advanced Studies
Ruth-Moufang-Straße 1, 60438 Frankfurt am Main, Germany

2- Xidian-FIAS International Joint Research Center
Ruth-Moufang-Straße 1, 60438 Frankfurt am Main, Germany

Abstract. The vision transformer (ViT) utilizes a learnable position embedding (PE) to encode the location of an image patch. However, it is unclear if this learnable PE is vital and what its benefits are. This paper explores an alternative way of encoding patch locations that exploits prior knowledge about their spatial arrangement called circle relationship embedding (CRE). CRE considers the distance of image patches from the central patch based on the four-neighborhood to simplify the PE. Our experiments show that combining CRE with PE achieves better performance than using PE alone. The code for this paper can be downloaded at: https://github.com/trieschlab/CRE.

1 Introduction

The Vision Transformer (ViT) applies the self-attention mechanism from state-of-the-art natural language processing (NLP) systems to image processing [1]. It retains the advantages of the transformer architecture in NLP systems, capturing global input relationships, parallelizing computations, and achieving performance comparable to or even surpassing convolutional neural networks (CNN) on various computer vision tasks.

Because ViT divides the input image into several patches and needs to model their global relationships directly, researchers focus on how to efficiently use the information of each patch to extract the most distinguishing features of different objects. The standard ViT introduces a positional embedding (PE) to solve this problem, which turned out to be crucial for vision tasks.

Many recent studies changed the learnable PE, proposed a new method to calculate the PE, and improved the novel PE by better expressing the location information of different patches. The random learnable PE parameters are still challenging to train well [2]. This paper avoids directly changing the PE but instead explores the hidden relationship between the input patches and generates the circle relationship embedding (CRE). In CRE, we generate the matrix of the PE from a learnable vector, which reduces the two-dimensional random parameter matrix to one dimension. For this, we treat the central patch as the

*This research was supported by "The Adaptive Mind" and "The Third Wave of Artificial Intelligence", funded by the Excellence Program of the Hessian Ministry of Higher Education, Science, Research and Art. JT was supported by the Johanna Quandt foundation. ZY thanks the Xidian-FIAS International Joint Research Center for funding.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9.
Available from http://www.i6doc.com/en/.

center of multiple concentric circles. Patches on the same circle are, by definition, equidistant from the central patch (Fig. 1).

The circle relationship embedding (CRE) matrix has the same size as the original PE. We can use the CRE to augment the original PE or replace it. The advantage of combining CRE and PE is that we can add information on patch relationships to the input without modifying the traditional ViT structure; if we replace PE with CRE, the learnable matrix reduces to a learnable vector. We thus compress the number of learnable parameters. Importantly, the effect of adding CRE on the training speed is negligible. Here we assess CRE on four public datasets. The results show that CRE is effective and provide novel insights into analyzing the PE in vision transformers.

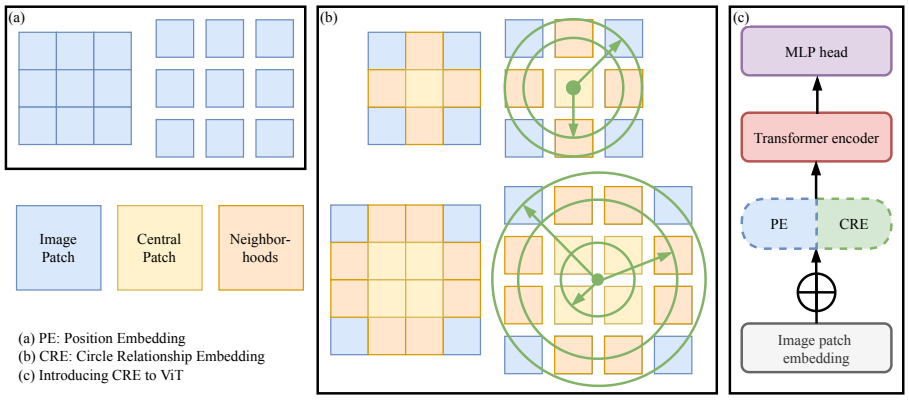

Fig. 1: Different embeddings and structure of the updated ViT. Green circles pass through centers of patches at the same distance from the image center. Yellow squares indicate center patches. Orange patches indicate nearest neighbors of the center patches. We use 3×3 patches and 4×4 patches as examples to represent cases where the number of patches is odd or even, respectively.

2 Related Work

The ViT treats different patches equally, but in fact, the positional relationship of different patches is not the same, so the introduction of PE helps to improve ViT performance. The following discusses four types of PE.

Learnable absolute PE. This PE is proposed by ViT [1] and is also the most widely used PE method. They set a random matrix of fixed dimensions and combined it with the patches. The optimizer and ViT parameters are updated synchronously, and the location information of different patches is added to the embeddings.

Learnable relative PE. Swin [3] is a representative of a learnable relative PE. It uses the relative distance between patches to encode position information, and the relative PE uses one-dimensional relative attention. In contrast, in CRE

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

the information is encoded according to the distance between the central patch and other patches.

Fixed PE. Fixed PE uses fixed absolute value coding to represent the positions of different patches. For example, [4] uses sine and cosine functions for the PE and connects the coding information of different frequencies to form the final PE.

Other PEs. In addition to the above three types of PE, this group includes PE by using convolutional space invariance [5] or using a continuous dynamic model [6].

Compared to the above methods, we propose CRE to extract the relationship between patches as a supplement to the PE. The size of the CRE is consistent with the PE, so we can combine the CRE with PE to add more information to the latent code or replace PE. Also, the number of learnable parameters of the CRE is smaller than that of the PE.

3 Circle relationship embedding

Here we present the CRE in detail. We first consider the form of the conventional PE matrix. Assuming N patches and the dimension of the PE for each patch being D, then $\boldsymbol{PE} \in \mathbb{R}^{N \times D}$. We can write the learnable position embedding matrix as:

$$\boldsymbol{PE} = \begin{bmatrix} p_{0,0} & \cdots & p_{0,D-1} \\ \vdots & \ddots & \vdots \\ p_{N-1,0} & \cdots & p_{N-1,D-1} \end{bmatrix}_{N \times D}, \tag{1}$$

where $p_{i,j}$ is the learnable position embedding element j of patch i.

In contrast, for CRE the embedding matrix is computed as the outer product of a vector containing the distances of all patches to a central patch and a learnable parameter vector:

$$\boldsymbol{CRE}_{N \times D} = \boldsymbol{d}_{N \times 1} \times \boldsymbol{CRC}_{1 \times D}\,, \tag{2}$$

where $\boldsymbol{d}$ is the N-dimensional distance vector and $\boldsymbol{CRC}$ is the D-dimensional learnable parameter vector.

Figure. 1A stands for the input image splitting in ViT. Figure. 1B illustrates the distances between central patches and others via green circles. Patches on the same circle have, by definition, the same distance from the center.

Figure 1C shows how the CRE is incorporated into the ViT. It can be seen that we can combine or replace PE with CRE because they have the same dimension. Compared to PE, CRE has fewer learnable parameters and is easy to calculate.

We first need to calculate the CRE for the central patch (CRC) and then evaluate the distance between the central patch and other patches.

$$\boldsymbol{CRC}_{odd} = \begin{bmatrix} p_{\frac{N-1}{2},0} & \cdots & p_{\frac{N-1}{2},D-1} \end{bmatrix}_{1 \times D} \tag{3}$$

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

and

$$\boldsymbol{CRC}_{even} = \begin{bmatrix} p_{\frac{N}{2}-\frac{\sqrt{N}}{2}-1,0} & \cdots & p_{\frac{N}{2}-\frac{\sqrt{N}}{2}-1,D-1} \\ p_{\frac{N}{2}-\frac{\sqrt{N}}{2},0} & \cdots & p_{\frac{N}{2}-\frac{\sqrt{N}}{2},D-1} \\ p_{\frac{N}{2}+\frac{\sqrt{N}}{2}-1,0} & \cdots & p_{\frac{N}{2}+\frac{\sqrt{N}}{2}-1,D-1} \\ p_{\frac{N}{2}+\frac{\sqrt{N}}{2},0} & \cdots & p_{\frac{N}{2}+\frac{\sqrt{N}}{2},D-1} \end{bmatrix}_{4\times D}, \tag{4}$$

where $\boldsymbol{CRC}_{odd}$ and $\boldsymbol{CRC}_{even}$ refer to the CRC from an odd number of patches and an even number of patches, respectively.

Here we consider $N \geq 9$ and have an integer square root. If N is odd, we only have one central patch, whereas we have four central patches if N is even. From Eq. 3 and Eq. 4, we can calculate the index of the central patch based on N. Each vector here is learnable, and for $\boldsymbol{CRC}_{even}$, the 4 vectors are the same, whereas the index of the patches is different. So the dimension of the learnable vector could also be $1 \times D$ and broadcast as $4 \times D$.

Before evaluating the distance between patches, we need to calculate the index of the adjacent patches concerning the central patch.

If the number of patches is odd, we set the center of the circle on the central patch, then consider that the 4-neighbors have the same distances from the central patch. We set one of these neighbors as a new central patch and use the 4-neighbors principle to find the patches that have not yet been traversed. It is considered that all the newly traversed patches are consistent with the initial center patch distance, and the distance is larger than the four patches traversed for the first time. If the index of the central patch is $\frac{N-1}{2}$, the indices of its 4 neighbors are:

$$\boldsymbol{I}_{neighbors} = \begin{bmatrix} & \frac{N-1}{2}-\sqrt{N} & \\ \frac{N-1}{2}-1 & \frac{N-1}{2} & \frac{N-1}{2}+1 \\ & \frac{N-1}{2}+\sqrt{N} & \end{bmatrix} \tag{5}$$

Now we can evaluate the distance after getting the index. Assume the distance between these 4-neighbors and the center patch is d, then we calculate the next neighbors based on the new four indices, repeat and set the new distance as $\sqrt{2}d$. So the number of $\boldsymbol{d}$ is $\sum_{i=3,\cdots,\sqrt{N}} \frac{i+1}{2}$. If the number of patches is even, we consider the center of the circle to be in the four central patches, the distance of these patches is 1, and the number of distances is $\sum_{i=4,\cdots,\sqrt{N}} \frac{i}{2}$. The $\boldsymbol{d}$ is

$$\boldsymbol{d}_{odd} = [d_{\sum_{i=3,\cdots,\sqrt{N}} \frac{i+1}{2}}, \cdots, d_0, \cdots, d_0, 1, d_0, \cdots, d_0, \cdots, d_{\sum_{i=3,\cdots,\sqrt{N}} \frac{i+1}{2}}]^T_{N\times 1} \tag{6}$$

and

$$\boldsymbol{d}_{even} = [d_{\sum_{i=4,\cdots,\sqrt{N}} \frac{i}{2}}, \cdots, d_0, \cdots, 1, 1, \cdots, 1, 1, \cdots, d_0, d_{\sum_{i=4,\cdots,\sqrt{N}} \frac{i}{2}}]^T_{N\times 1}, \tag{7}$$

where d can be a hyperparameter or a fixed number. We use Euclidean distance and enlarge the original vector to calculate the distance vector: $\boldsymbol{d}_9 = [\sqrt{2}, 1, \sqrt{2}, 1, 0, 1, \sqrt{2}, 1, \sqrt{2}]^T \times \sqrt{2} = [2, \sqrt{2}, 2, \sqrt{2}, 0, \sqrt{2}, 2, \sqrt{2}, 2]^T$.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

Note that the element in d cannot be zero because we need to calculate CRE for every patch, so we set the central element in $\boldsymbol{d}_{odd}$ as one, ensuring that the CRE of the central patch after matrix multiplication is not a zero vector. For the nine patches and sixteen patches from Fig. 1B, the distance vectors are defined as $\boldsymbol{d}_9 = [2, \sqrt{2}, 2, \sqrt{2}, 1, \sqrt{2}, 2, \sqrt{2}, 2]^T$ and $\boldsymbol{d}_{16} = [2, \sqrt{2}, \sqrt{2}, 2, \sqrt{2}, 1, 1, \sqrt{2}, \sqrt{2}, 1, 1, \sqrt{2}, 2, \sqrt{2}, \sqrt{2}, 2]^T$.

4 Experiments

Method	CIFAR10	CIFAR100	T-ImageNet	Flowers102
ViT-Base ($\mu \pm \sigma$)				
PE	95.54±1.02	76.13±1.20	54.92±1.65	85.71±1.33
CRE	96.26±0.64	77.19±1.42	55.34±1.54	86.44±0.41
CRE+PE	**96.29±0.85**	**77.33±1.34**	**55.40±1.58**	**86.83±0.59**
T2T-Base ($\mu \pm \sigma$)				
PE	96.70±0.16	79.25±0.87	57.66±1.44	88.14±1.31
CRE	96.99±0.64	79.81±1.22	57.93±1.79	88.37±0.62
CRE+PE	**97.51±0.86**	**80.28±1.53**	**58.59±1.71**	**88.71±0.37**
Swin-Base ($\mu \pm \sigma$)				
PE	96.80±1.21	79.77±1.37	58.51±1.82	89.44±0.79
CRE	97.25±0.97	80.15±1.68	59.28±1.75	89.74±1.44
CRE+PE	**97.81±0.27**	**80.44±1.08**	**59.83±1.87**	**89.98±0.36**

Table 1: Top-1 accuracy comparison on different datasets

We chose four popular datasets (CIFAR10, CIFAR100 [7], Tiny-ImageNet [8], and Flowers102 [9]) to evaluate our method. We assign 60 % as the trainset, 30 % as validation, and the other for the test. Each experiment trained for 400 epochs. The batch size is 128, and we use AdamW as the optimizer. Our experiments use these relatively small datasets and batch sizes to allow us to train our models using a single GPU (Nvidia GeForce Titan X). The initial learning rate was 0.001 and we used a cosine decay of the learning rate. Each reported result is shown as mean ± std across five runs.

To evaluate the robustness of CRE, we test it with three different ViT structures. Here, ViT-Base, T2T-Base, and Swin-Base mean ViT-B/16 [1], T2T-ViT-14 [10], and Swin-T [3]. PE is the learnable absolute PE in Sec. 2, and CRE+PE means matrix addition. We treat PE as the baseline to compare our method. Considering the results in Tab. 1, we can draw several conclusions.

1) CRE can directly replace or combine with PE in the three ViT backbones because they use the same matrix dimension for the position embedding.

2) We observe marginally improved ViT performance for CRE compared to PE on all datasets. We hypothesize that this is due to the additional inductive bias of CRE, which focuses on relative patch location, while PE only pays attention to the absolute location of patches.

ESANN 2023 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium) and online event, 4-6 October 2023, i6doc.com publ., ISBN 978-2-87587-088-9. Available from http://www.i6doc.com/en/.

3) The combination of both consistently outperforms just using PE or CRE alone. This suggests that CRE is complementary to PE.

5 Conclusion

In this work, we introduce the circle relationship embedding (CRE) to represent the relationship between patches and the central patch in Vision Transformer (ViT) architectures. We construct the CRE by multiplying a distance vector with a learnable vector, reducing the learnable parameters compared to conventional position embedding (PE). CRE can replace or combine with PE. When combined, it outperforms using PE alone. However, CRE still needs to be analyzed from more perspectives, such as training speed, latent representation, and visualization, and it also needs more ViT backbones and datasets to evaluate performance. Exploring these issues will help us better understand the effects of different types of PE in ViT models.

References

[1] 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.

[2] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[3] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[5] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021.

[6] Xuanqing Liu, Hsiang-Fu Yu, Inderjit Dhillon, and Cho-Jui Hsieh. Learning to encode position for transformer with continuous dynamical model. In *International conference on machine learning*, pages 6327–6335. PMLR, 2020.

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

[8] Olga Russakovsky, Jia Deng, Hao Su, et al. Imagenet large scale visual recognition challenge. 2015.

[9] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[10] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 558–567, 2021.