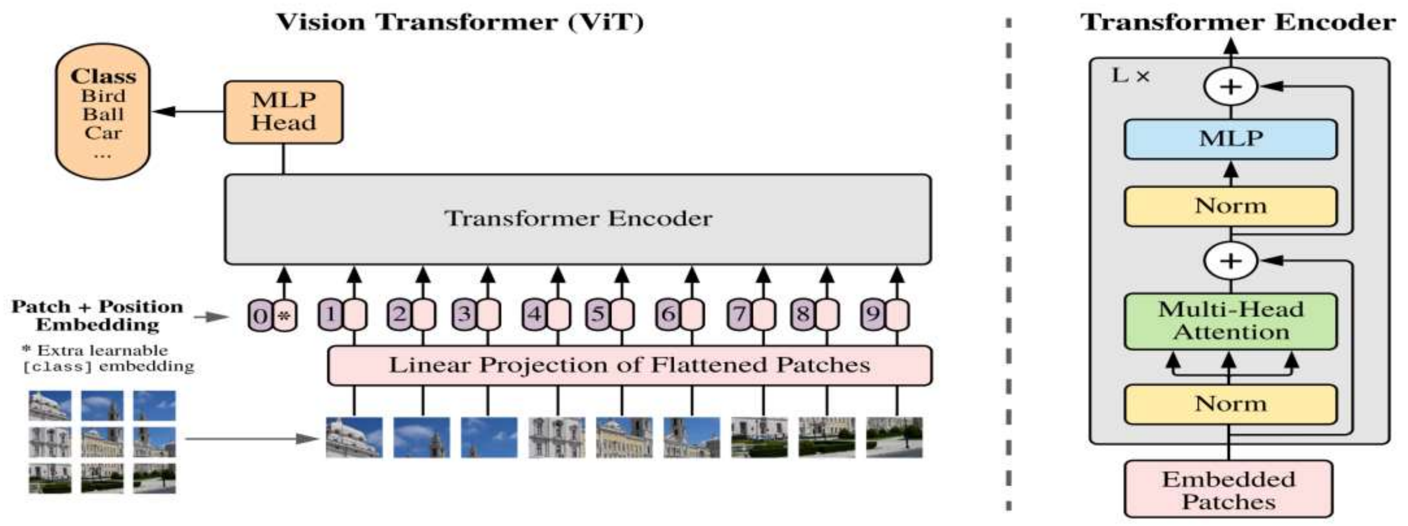
THE UNIVERSITY OF SYDNEY

When Fine-tuning always Beats Training from Scratch: Strategies to Adopt in Vision Transformer Training

Kuangshi Ai (530280790) and Jiaying He (530287984)

Research Problem

➢ Motivation



- Inefficiency in traditional ViT's data-preprocessing scheme: uniformly cut input image into square tokens
- Weak inductive bias of traditional ViT, which results in smaller receptive field and loss of local information

➢ Two Research lines

- **Training from Scratch**

  Rotated Patch Tokenization(RPT) & Learnable Positional Embedding
- **Finetuning Pre-trained Models**

➢ Metrics

- Three types of cost to consider:

  Pre-train cost

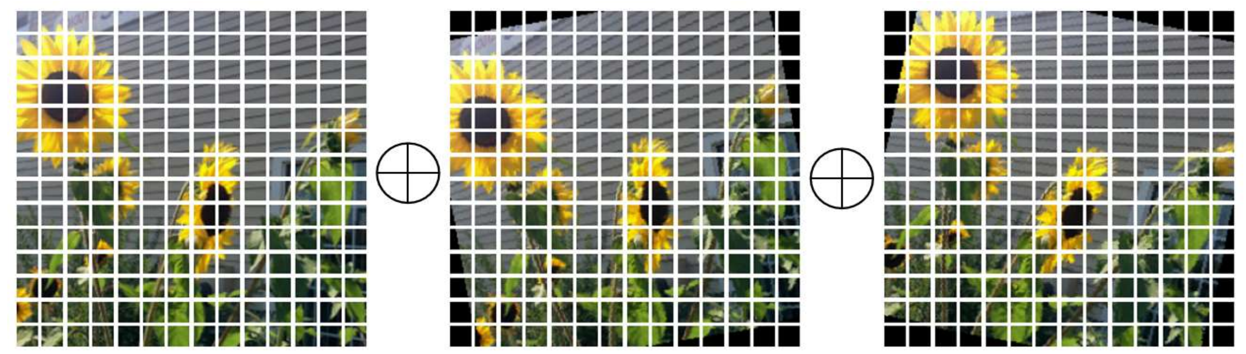
  Practitioner cost (fine-tuning cost on target dataset)

  Deployment cost (inference cost of trained model)
- Upstream accuracy & downstream accuracy

Proposed Methods

➢ Rotated Patch Tokenization(RPT)

For traditional ViT, the receptive field size of the tokens can be calculated by: $r_{token} = s * (r_{trans} - 1) + k$, where s, k stands for the stride and the kernel size of the convolutional layer. As $r_{trans} = 1$, the receptive field size of vanilla ViT equals $r_{token} = k =$ patch size.



Original Rotate-anticlock Rotate-clock

To enlarge the receptive size, we rotate the input image clockwise and anticlockwise for some random angle between $(\pi/20, \pi/14)$, crop the rotated images to the same size and concatenate them with the original input, then divide the concatenated features into patches and flatten them. At last apply LN and linear projection. This process can be summarized as the formula below:

$$\mathcal{R}(\mathbf{x}) = \mathrm{LN}\left(\mathcal{P}\left(\left[\mathbf{X}\mathbf{R}^1\mathbf{R}^2\ldots\mathbf{R}^{N_\mathcal{R}}\right]\right)\right) \boldsymbol{W}_\mathcal{R}\,.$$

➢ Learnable Positional Embedding

$$\mathrm{R}'(\mathbf{x}) = \begin{cases} [\mathbf{x}_{cls}; \mathrm{R}(\mathbf{x})] + \boldsymbol{POS} & \text{if } \mathbf{x}_{cls} \text{ exist} \\ \mathrm{R}(\mathbf{x}) + \boldsymbol{POS} & \text{otherwise} \end{cases}$$

We let POS be a learnable parameter.

Experiment

➢ Environment & Dataset

- **4 relatively small datasets**: TF-Flowers, CIFAR10, CIFAR100, Tiny-ImageNet(100,000 samples).
- **Pre-trained models trained on large scale image datasets**: ImageNet and ImageNet-21k.
- **Single NVIDIA A100 GPU offered by Google Colab, with batch size 256.**

➢ Quantitative Results

- **Performance of training from scratch**

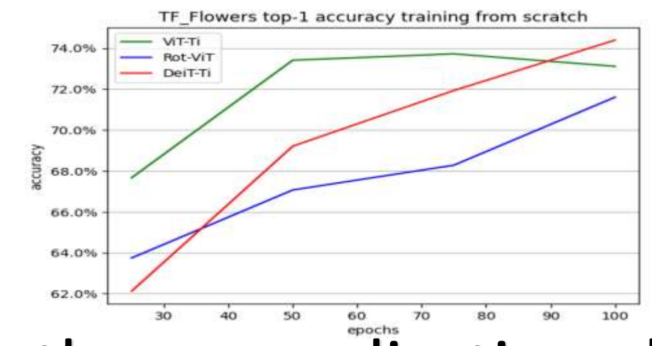
Table 1: Top-1 accuracy comparison of different models, all trained from scratch for 50 epochs. The throughput is measured on 224px resolution.

Model	Throughput (images/sec)	Params (M)	TF_FLOWERS	CIFAR10	CIFAR100	T-ImageNet
ResNet 56	798	0.8	76.67	83.41	54.16	33.38
ViT-Ti	1600	2.58	73.02	81.56	52.35	32.41
SL-ViT	1264	2.7	67.57	82.94	55.71	34.45
DeiT-Ti	238	5	72.21	-	-	-
Rot-ViT	1333	2.6	67.3	81.96	54.75	34.22

Table 2: Throughput (images/sec) comparison of original ViT-Ti and our model on different datasets

Model	TF_FLOWERS	CIFAR10	CIFAR100	T-ImageNet
ViT-Ti	1676	3605	2162	3454
Rot-ViT	1333	3282	1954	3240

RPT improves the model accuracy only with small overhead of inference latency.



Enhances the generalization ability: RPT, distillation, regularization, data augmentation...

- **Finetune pre-trained models**

Table 3: Configuration of ViT models

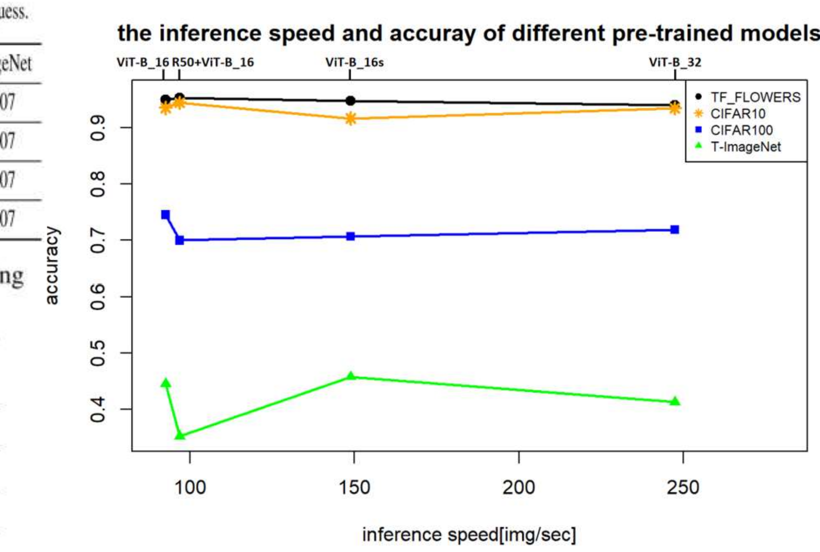
Model	Layers	Width	MLP	Heads	Params (M)
ViT-Ti [3]	12	192	768	3	5.8
ViT-S [3]	12	384	1536	6	22.2
ViT-B [1]	12	768	3072	12	86
ViT-L [1]	24	1024	4096	16	307

Table 4: Top-1 accuracy before fine-tuning, which is almost random guess.

Model	TF_FLOWERS	CIFAR10	CIFAR100	T-ImageNet
ViT-B_16	0.2246	0.1006	0.0103	0.0007
ViT-B_16s	0.2246	0.1006	0.0103	0.0007
ViT-B_32	0.2246	0.1006	0.0103	0.0007
R50+ViT-B_16	0.2246	0.1006	0.0103	0.0007

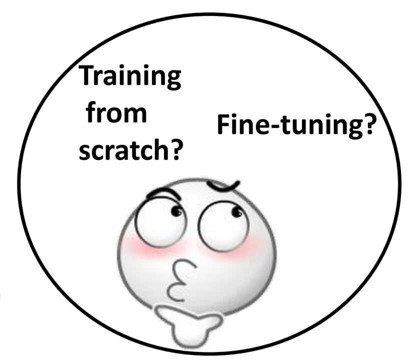
Table 5: Top-1 accuracy after fine-tuning. The throughput measurements were obtained by averaging all instant inference speed of a single model.

Model	Throughput (images/sec)	TF_FLOWERS	CIFAR10	CIFAR100	T-ImageNet
ViT-B_16	92.59	0.9492	0.9351	0.7447	0.4452
ViT-B_16s	148.84	0.9473	0.9163	0.7064	0.4580
ViT-B_32	247.34	0.9395	0.9343	0.7181	0.4128
R50+ViT-B_16	96.88	0.9531	0.9441	0.6994	0.3524



Pre-trained ResNet+ViT hybrid model does not perform as well as other pure ViT models on mid-sized dataset, like Tiny-ImageNet.

➢ Strategies to Adopt in ViT Training



How to find the parameter checkpoint with optimal hyperparameter settings? How to choose the favorable ViT model size? How to effectively fine-tune the chosen model?

Conclusion

➢ Having explored how to improve the efficiency and lower the cost, we propose practical instructions for training ViT.

➢ For both training from scratch and finetuning pre-trained models: increasing "receptive field" is only effective for smaller datasets while the transformer and the principle of attention is more competitive for large datasets.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[2] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pretraining or strong data augmentations. *CoRR*, abs/2106.01548, 2021.

[3] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers amp; distillation through attention, 2020.

[4] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *CoRR*, abs/2101.11986, 2021.

[5] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets, 2021.

[6] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *CoRR*, abs/2106.10270, 2021.

[7] A. Araujo, W. Norris, and J. Sim. Computing receptive fields of convolutional neural networks. *Distill*, 4(11), 2019.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[9] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[10] Google Research. Vision transformer and mlp-mixer architectures, 2022. Software available from tensorflow.org.

[11] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge, 2015.

[12] Google. Tensorflow datasets, a collection of ready-to-use datasets.