YONSEI UNIVERSITY

ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases (ICML 2021)

- Young Jo Choi
- Department of Digital Analytics

Severance

Abstract

- Convolution architectures :
  their hard inductive bias → sample-efficient learning, but potentially lower performance ceiling

- Vision Transformers :
  rely on flexible self-attention layer and recently outperformed CNN for image classification, but require large dataset or distillation from pretrained CNN

- Combination of strengths of the two architectures while avoiding their respective limitations
  → GPSA(gated positional self-attention)
  : a form of **positional self-attention** which can be equipped with a 'soft' **convolutional inductive bias**

Abstract

- GPSA : Self-attention layer which is initialized to mimic the locality of convolutional layer

- By *gating parameter*, adjusting the attention paid to position **vs** content information

- The paper investigated the role of locality in learning by quantifying how it is encouraged in vanilla self-attention layers, then analyzing how it is escaped in GPSA layers.

Introduction

- Inductive biases : hard-coded into the architecture of CNNs
  (locality and weight sharing)

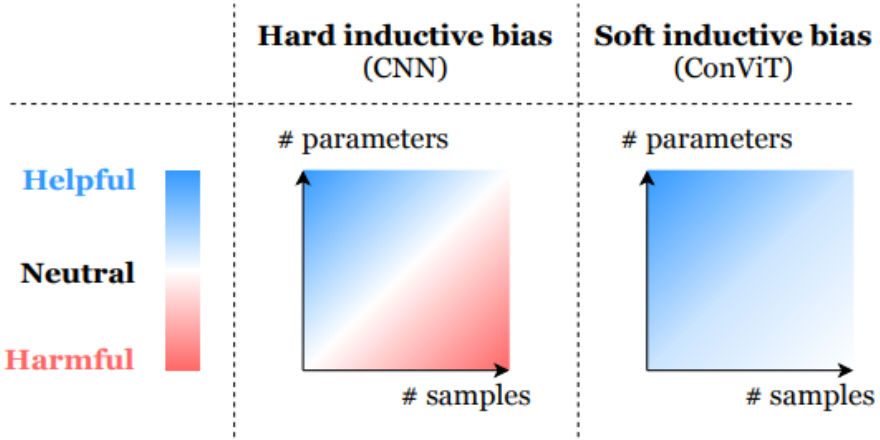
- In vision tasks, the locality of CNNs impairs the ability to capture long-range dependencies, whereas attention does not suffer from this limitation.

- ViT[1] entirely dispenses with the convolutional inductive bias by performing SA across embeddings of patches of pixels.

- More recently, the “Data-efficient Vision Transformer” (DeiT[2]) was able to reach similar performances without any pre-training on supplementary data, instead relying on Knowledge Distillation from a convolutional teacher. (: transfer the inductive biases of a CNN teacher to a student transformer)

[1] Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).
[2] Touvron, Hugo, et al. "Training data-efficient image transformers & distillation through attention." *International conference on machine learning*. PMLR, 2021.

Introduction



- Soft inductive biases vs Hard inductive biases
- Convolutional constraints : sample efficient training in the small-data regime
  ← can become limiting as the dataset size is not an issue.
- 데이터가 충분한 상황에서 hard inductive biases는 오히려 제한적이 될 수 있음.
  가장 적절한 inductive bias를 학습하는 것이 더 효율적일 수 있음
- Convolutional model : high performance floor but a potentially lower performance ceiling due to hard inductive biases.
- Self-attention based model : lower floor but a higher ceiling.
- One of successful approach of this dilemma is Hybrid models (선행연구 : DeiT)

Related Work

- Cordonnier et al. (2019)[3] showed that a SA layer with N_h heads can express a convolution of kernel size $\sqrt{N_h}$, if each head focuses on one of the pixels in the kernel patch.

Background

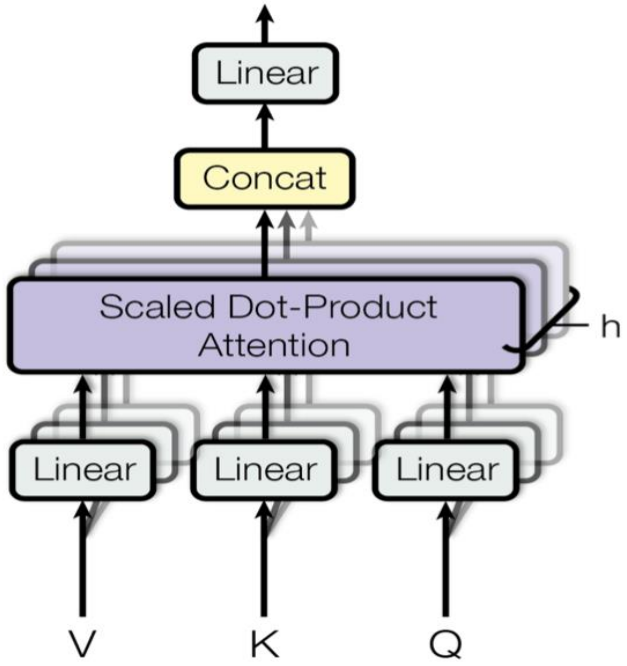
- We begin by introducing the basics of SA layers, and show how positional attention can allow SA layers to express convolutional layers.

[3] Cordonnier, Jean-Baptiste, Andreas Loukas, and Martin Jaggi. "On the relationship between self-attention and convolutional layers." *arXiv preprint arXiv:1911.03584* (2019).

Self-attention

- based on a trainable associative memory with (key, query) pairs.

- The Attention A whose entry (i, j) represents how semantically relevant Q_i is to K_j



$Q \in \mathbb{R}^{L_1 \times D_h}$: query embedding
$K \in \mathbb{R}^{L_2 \times D_h}$: key embedding

$$\boldsymbol{A} = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D_h}}\right) \in \mathbb{R}^{L_1 \times L_2}$$

$$\text{SA}_h(\boldsymbol{X}) := \boldsymbol{A}^h \boldsymbol{X} \boldsymbol{W}_{val}^h$$

$$\text{MSA}(\boldsymbol{X}) := \underset{h \in [N_h]}{\text{concat}} \left[\text{SA}_h(\boldsymbol{X})\right] \boldsymbol{W}_{out} + \boldsymbol{b}_{out}$$

Self-attention

- In vanilla form of self-attention, SA layer are position agnostic.

- To incorporate positional information,
  1. Add some positional information to the input at embedding time, before propagating it through the SA layers (ViT uses this approach)

  2. Replace the vanilla SA with positional self-attention (**PSA**)[4], using encoding r_{ij} of the relative position of patches i and j

$$\boldsymbol{A}_{ij}^{h} := \mathrm{softmax}\left(\boldsymbol{Q}_{i}^{h}\boldsymbol{K}_{j}^{h\top} + \boldsymbol{v}_{pos}^{h\top}\boldsymbol{r}_{ij}\right)$$

$v_{pos}^{h} \in \mathbb{R}^{D_{pos}}$: trainable embedding ,
$r_{ij} \in \mathbb{R}^{D_{pos}}$: relative positional encoding (non-trainable) only depending on the distance between pixels i and j denoted by a 2-D vector δ_{ij}

[4] Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. arXiv preprint arXiv:1906.05909, 2019.

참고)

$$\mathrm{Conv}(\boldsymbol{X})_{i,j,:} := \sum_{(\delta_1,\delta_2)\in\mathbb{\Delta}_K} \mathbf{X}_{i+\delta_1,j+\delta_2,:}\mathbf{W}_{\delta_1,\delta_2,:,:} + \boldsymbol{b},$$

where $\mathbf{W}$ is the $K \times K \times D_{in} \times D_{out}$ weight tensor [4], $\boldsymbol{b} \in \mathbb{R}^{D_{out}}$ is the bias vector and the set

$$\mathbb{\Delta}_K := \left[-\left\lfloor\frac{K}{2}\right\rfloor, \cdots, \left\lfloor\frac{K}{2}\right\rfloor\right] \times \left[-\left\lfloor\frac{K}{2}\right\rfloor, \cdots, \left\lfloor\frac{K}{2}\right\rfloor\right]$$

contains all possible shifts appearing when convolving the image with a $K \times K$ kernel.

[3] Cordonnier, Jean-Baptiste, Andreas Loukas, and Martin Jaggi. "On the relationship between self-attention and convolutional layers." *arXiv preprint arXiv:1911.03584* (2019).

(if $p = (i, j)$, we write $X_{p,:}$ and $A_{p,:}$ to mean $X_{i,j,:}$ and $A_{i,j,:,:}$)

- Absolute Attention (ViT)

$$\begin{aligned}\mathbf{A}^{\mathrm{abs}}_{q,k} &= (\mathbf{X}_{q,:} + \mathbf{P}_{q,:})\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}(\mathbf{X}_{k,:} + \mathbf{P}_{k,:})^{\top} \\ &= \underbrace{\mathbf{X}_{q,:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{X}_{k,:}^{\top}}_{\text{Only depends on the content of the key and query pixel}} + \mathbf{X}_{q,:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{P}_{k,:}^{\top} + \mathbf{P}_{q,:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{X}_{k,:} + \underbrace{\mathbf{P}_{q,:}\boldsymbol{W}_{qry}\boldsymbol{W}_{key}^{\top}\mathbf{P}_{k,:}}_{\text{Only depends on the positions of the key and query pixel}}\end{aligned}$$

- Relative Attention (ConViT)

$$\mathbf{A}^{\mathrm{rel}}_{q,k} := \mathbf{X}_{q,:}^{\top}\boldsymbol{W}_{qry}^{\top}\boldsymbol{W}_{key}\,\mathbf{X}_{k,:} + \mathbf{X}_{q,:}^{\top}\boldsymbol{W}_{qry}^{\top}\widehat{\boldsymbol{W}}_{key}\,\boldsymbol{r}_{\delta} + \boldsymbol{u}^{\top}\boldsymbol{W}_{key}\,\mathbf{X}_{k,:} + \boldsymbol{v}^{\top}\widehat{\boldsymbol{W}}_{key}\,\boldsymbol{r}_{\delta}$$

r_δ:= relative distance between query and key
key weights are split into two types : W_{key} pertain to input and $\widehat{W}_{key}$ to the relative position of pixels

Self-attention (from a positional encoding perspective)

Theorem 1. *A multi-head self-attention layer with N_h heads of dimension D_h, output dimension D_{out} and a relative positional encoding of dimension $D_p \geq 3$ can express any convolutional layer of kernel size $\sqrt{N_h} \times \sqrt{N_h}$ and $\min(D_h, D_{out})$ output channels.*

Lemma 1. *Consider a multi-head self-attention layer consisting of $N_h = K^2$ heads, $D_h \geq D_{out}$ and let $\boldsymbol{f} : [N_h] \to \mathbb{\Delta}_K$ be a bijective mapping of heads onto shifts. Further, suppose that for every head the following holds:*

$$\text{softmax}(\boldsymbol{A}^{(h)}_{\boldsymbol{q},:})_{\boldsymbol{k}} = \begin{cases} 1 & \text{if } \boldsymbol{f}(h) = \boldsymbol{q} - \boldsymbol{k} \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Then, for any convolutional layer with a $K \times K$ kernel and D_{out} output channels, there exists $\{\boldsymbol{W}^{(h)}_{val}\}_{h \in [N_h]}$ such that $\text{MHSA}(\boldsymbol{X}) = \text{Conv}(\boldsymbol{X})$ for every $\boldsymbol{X} \in \mathbb{R}^{W \times H \times D_{in}}$.

Lemma 2. *There exists a relative encoding scheme $\{\boldsymbol{r}_{\boldsymbol{\delta}} \in \mathbb{R}^{D_p}\}_{\boldsymbol{\delta} \in \mathbb{Z}^2}$ with $D_p \geq 3$ and parameters $\boldsymbol{W}_{qry}, \boldsymbol{W}_{key}, \widehat{\boldsymbol{W}}_{key}, \boldsymbol{u}$ with $D_p \leq D_k$ such that, for every $\boldsymbol{\Delta} \in \mathbb{\Delta}_K$ there exists some vector $\boldsymbol{v}$ (conditioned on $\boldsymbol{\Delta}$) yielding $\text{softmax}(\mathbf{A}_{\boldsymbol{q},:})_{\boldsymbol{k}} = 1$ if $\boldsymbol{k} - \boldsymbol{q} = \boldsymbol{\Delta}$ and zero, otherwise.*

Then, Lemma 2 shows that the aforementioned condition is satisfied for the relative positional encoding that we refer to as the *quadratic encoding*:

$$\boldsymbol{v}^{(h)} := -\alpha^{(h)}\,(1, -2\boldsymbol{\Delta}^{(h)}_1, -2\boldsymbol{\Delta}^{(h)}_2) \quad \boldsymbol{r}_{\boldsymbol{\delta}} := (\|\boldsymbol{\delta}\|^2, \boldsymbol{\delta}_1, \boldsymbol{\delta}_2) \quad \boldsymbol{W}_{qry} = \boldsymbol{W}_{key} := \mathbf{0} \quad \widehat{\boldsymbol{W}_{key}} := \boldsymbol{I} \tag{9}$$

The learned parameters $\boldsymbol{\Delta}^{(h)} = (\boldsymbol{\Delta}^{(h)}_1, \boldsymbol{\Delta}^{(h)}_2)$ and $\alpha^{(h)}$ determine the center and width of attention of each head, respectively. On the other hand, $\boldsymbol{\delta} = (\boldsymbol{\delta}_1, \boldsymbol{\delta}_2)$ is fixed and expresses the relative shift between query and key pixels.

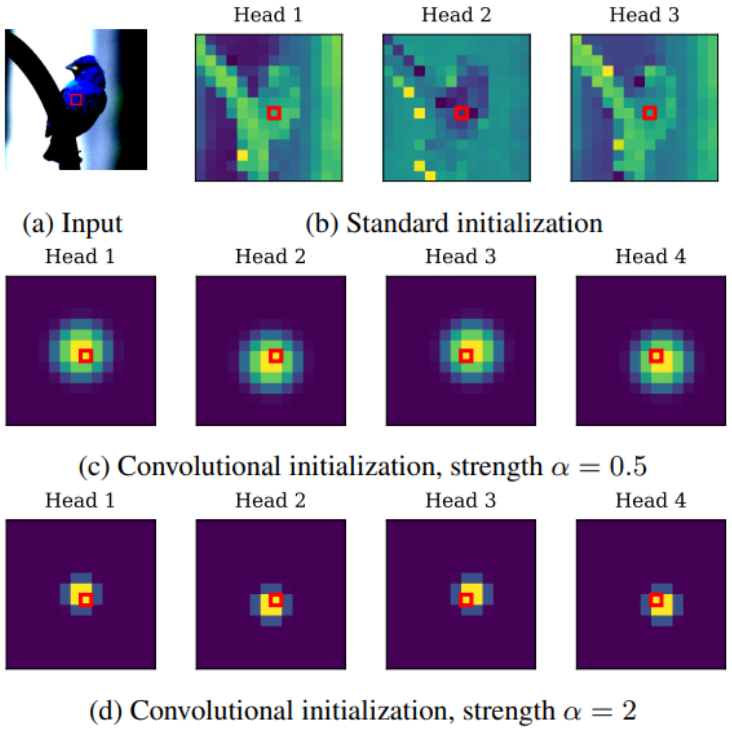
Self-attention as a generalized convolution

- Cordonnier et al. (2019) show that a multi-head PSA layer with N_h heads and learnable relative positional encodings of dimension $D_{pos} \geq 3$ **can express any convolutional layer** of filter size $\sqrt{N_h} \times \sqrt{N_h}$, by setting the following:

$$\boldsymbol{A}_{ij}^h := \text{softmax}\left(\boldsymbol{Q}_i^h \boldsymbol{K}_j^{h\top} + \boldsymbol{v}_{pos}^{h\top} \boldsymbol{r}_{ij}\right) \longleftarrow \begin{cases} \boldsymbol{v}_{pos}^h := -\alpha^h \left(1, -2\Delta_1^h, -2\Delta_2^h, 0, \ldots 0\right) \\ \boldsymbol{r}_{\boldsymbol{\delta}} := \left(\|\boldsymbol{\delta}\|^2, \delta_1, \delta_2, 0, \ldots 0\right) \\ \boldsymbol{W}_{qry} = \boldsymbol{W}_{key} := \boldsymbol{0}, \quad \boldsymbol{W}_{val} := \boldsymbol{I} \end{cases}$$

- $\Delta^h \in \mathbb{R}^2$: Center of attention (the position to which head h pays most attention to, relative to the query patch)
- $\alpha^h > 0$: locality (determines how focused the attention is around its center Δ^h)
- Thus, the PSA layer can achieve a strictly convolutional attention map by setting the center of attention Δ^h to each of the possible positional offsets of $\sqrt{N_h} \times \sqrt{N_h}$ convolutional kernel, and sending the locality strengths α^h to some large value

Self-attention as a generalized convolution



(a): Input image from ImageNet, where the query patch is highlighted by a red box.

(b),(c),(d): attention maps of an untrained SA layer

(b) and those of a PSA layer using the convolutional-like initialization scheme

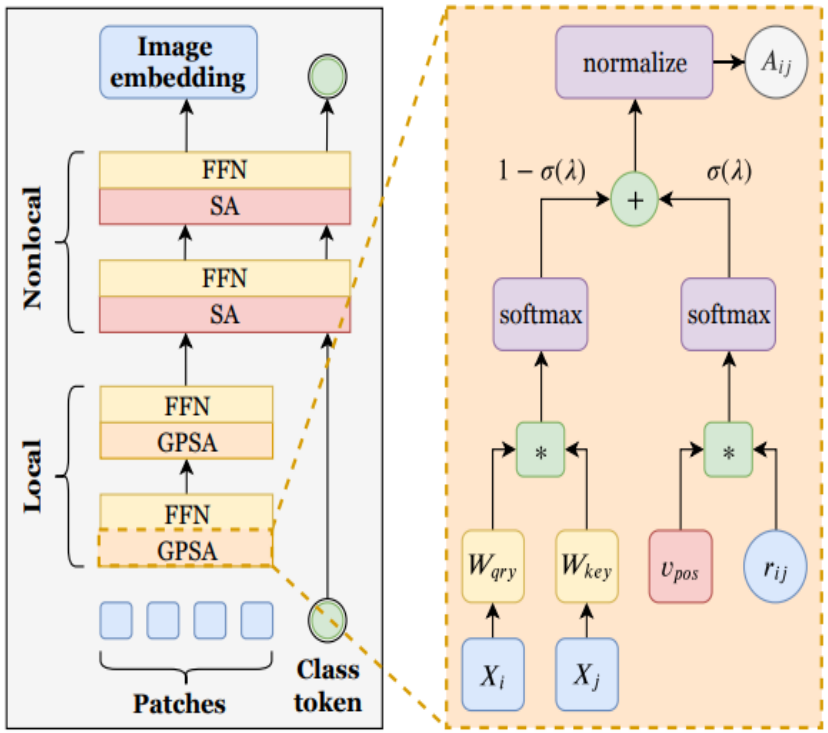
$\Delta^h \in \mathbb{R}^2$: Center of attention (the position to which head h pays most attention to, relative to the query patch)
$\alpha^h > 0$: locality (determines how focused the attention is around its center Δ^h)

Architecture

- ConViT : a variant of the ViT obtained by replacing some of the SA layers by a new type of layer **GPSA** (gated positional self-attention)

- Core idea : to enforce the informed convolutional configuration in the GPSA layers at initialization
  → decide whether to stay convolutional or not.

Architecture



ConViT

Gated Positional Self-Attention

PSA layer

$$\mathrm{SA}_h(\boldsymbol{X}) := \boldsymbol{A}^h \boldsymbol{X} \boldsymbol{W}_{val}^h$$

$$\boldsymbol{A}_{ij}^h := \mathrm{softmax}\left(\boldsymbol{Q}_i^h \boldsymbol{K}_j^{h\top} + \boldsymbol{v}_{pos}^{h\top} \boldsymbol{r}_{ij}\right)$$

GPSA layer

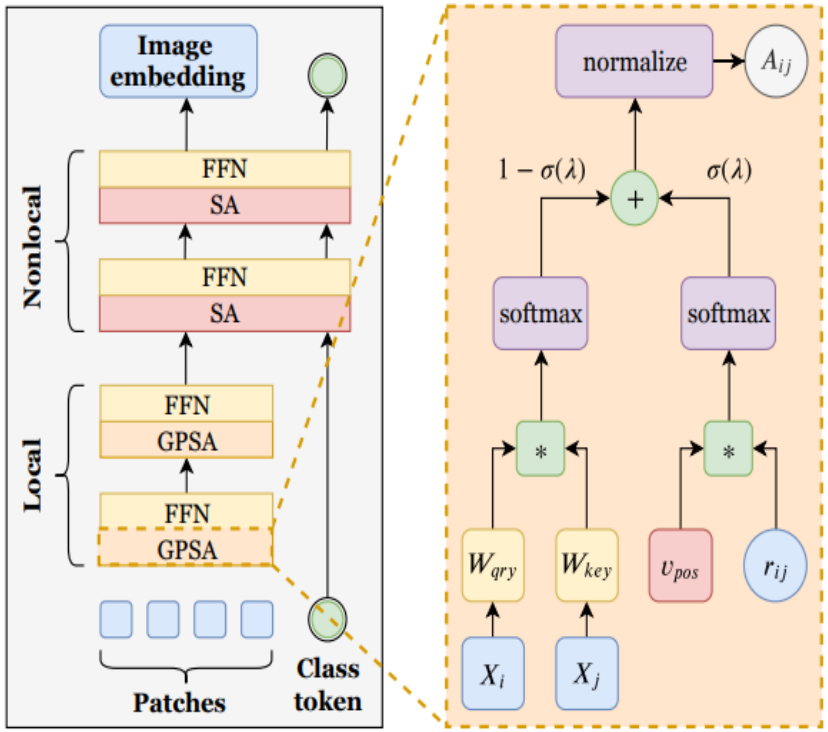
$$\mathrm{GPSA}_h(\boldsymbol{X}) := \mathrm{normalize}\left[\boldsymbol{A}^h\right] \boldsymbol{X} \boldsymbol{W}_{val}^h$$

$$\boldsymbol{A}_{ij}^h := (1-\sigma(\lambda_h))\,\mathrm{softmax}\left(\boldsymbol{Q}_i^h \boldsymbol{K}_j^{h\top}\right) + \sigma(\lambda_h)\,\mathrm{softmax}\left(\boldsymbol{v}_{pos}^{h\top} \boldsymbol{r}_{ij}\right),$$

Softmax can ignore the smallest of the two
(In particular, the convolutional initialization scheme involves highly concentrated attention scores)
→ Sum the contend and positional terms after the softmax

λ_h : learnable gating parameter (one for each head)
($\lambda_h \uparrow \rightarrow \sigma(\lambda_h) \backsimeq 1 \rightarrow$ GPSA attends purely on position)

Architecture

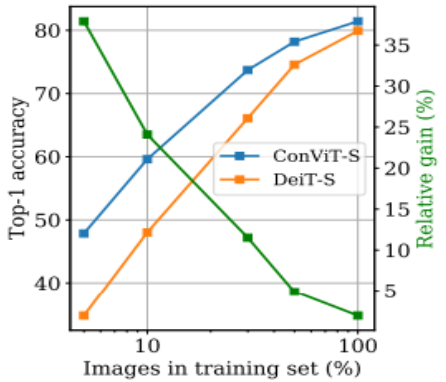


- ViT
  tokenizing : slice input images of size 224 into 16x16 non-overlapping patches of 14x14 pixels and embeds them into vectors of dimension $D_{emb} = 64N_h$ using convolutional stem
  + class token : does not carry any positional imformation
- → propagate : 12 blocks which keep their dimensionality constant.
- 각 block은 SA layer와 이어지는 FFN layer (GeLU activation)와 skip connectio으로 이뤄져있음

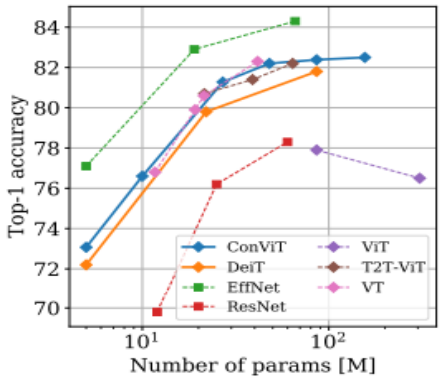
- ConViT는 ViT에서 앞 10개 block의 SA를 GPSA로 바꿨음
- GPSA layer는 positional attention을 포함하므로 class token을 더하는게 적절하지 않음 → GPSA layer 이후에 append함

Results

Name	Model	N_h	D_{emb}	Size	Flops	Speed	Top-1	Top-5
Ti	DeiT	3	192	6M	1G	1442	72.2	-
	ConViT	4	192	6M	1G	734	**73.1**	**91.7**
Ti+	DeiT	4	256	10M	2G	1036	75.9	93.2
	ConViT	4	256	10M	2G	625	**76.7**	**93.6**
S	DeiT	6	384	22M	4.3G	587	79.8	-
	ConViT	9	432	27M	5.4G	305	**81.3**	**95.7**
S+	DeiT	9	576	48M	10G	480	79.0	94.4
	ConViT	9	576	48M	10G	382	**82.2**	**95.9**
B	DeiT	12	768	86M	17G	187	81.8	-
	ConViT	16	768	86M	17G	141	**82.4**	**95.9**
B+	DeiT	16	1024	152M	30G	114	77.5	93.5
	ConViT	16	1024	152M	30G	96	**82.5**	**95.9**



(a) Sample efficiency



(b) Parameter efficiency

Train size	Top-1 DeiT	Top-1 ConViT	Top-1 Gap	Top-5 DeiT	Top-5 ConViT	Top-5 Gap
5%	34.8	**47.8**	37%	57.8	**70.7**	22%
10%	48.0	**59.6**	24%	71.5	**80.3**	12%
30%	66.1	**73.7**	12%	86.0	**90.7**	5%
50%	74.6	**78.2**	5%	91.8	**93.8**	2%
100%	79.9	**81.4**	2%	95.0	**95.8**	1%

(Sample efficiency)

- On ImageNet (300 epochs)
- Speed : number of images processed per second on a a Nvidia Quadro GP100 GPU at batch size 128
- 2x2, 3x3, 4x4 convolutional filter를 모방하기 위해 convit 모델의 attention head를 4,9,16으로 설정
- Learning rate : 0.0005 → 0.0004, batch size : 1024→512
- 별도의 hyperparameter 조정 없이 DeiT 논문에 제시된 것을 그대로 따름

Investigating the role of locality

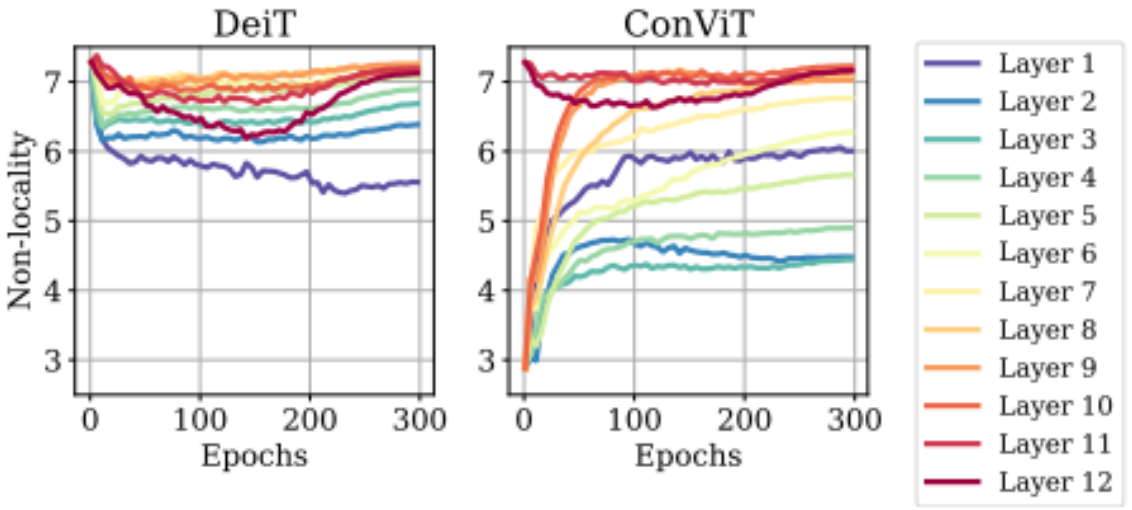
We begin by investigating whether the hypothesis that PSA layers are naturally encouraged to become "local" over the course of training holds for the vanilla SA layers used in ViTs, which do not benefit from positional attention

Define **non-locality**

$$D_{loc}^{\ell,h} := \frac{1}{L}\sum_{ij} \boldsymbol{A}_{ij}^{h,\ell}\|\boldsymbol{\delta}_{ij}\|,$$

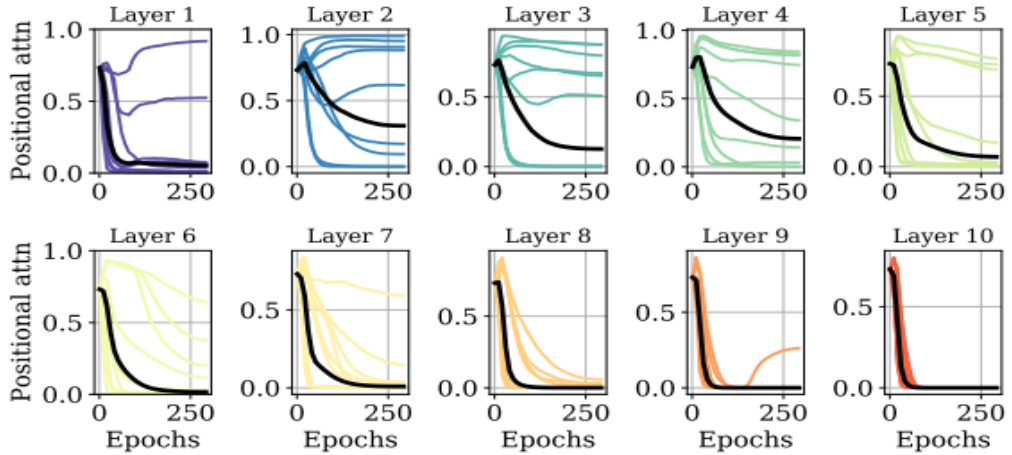
$$D_{loc}^{\ell} := \frac{1}{N_h}\sum_{h} D_{loc}^{\ell,h}$$

for each query patch i, the distances $\|\delta_{ij}\|$ to all the key patches j weighted by their attention score A_{ij}



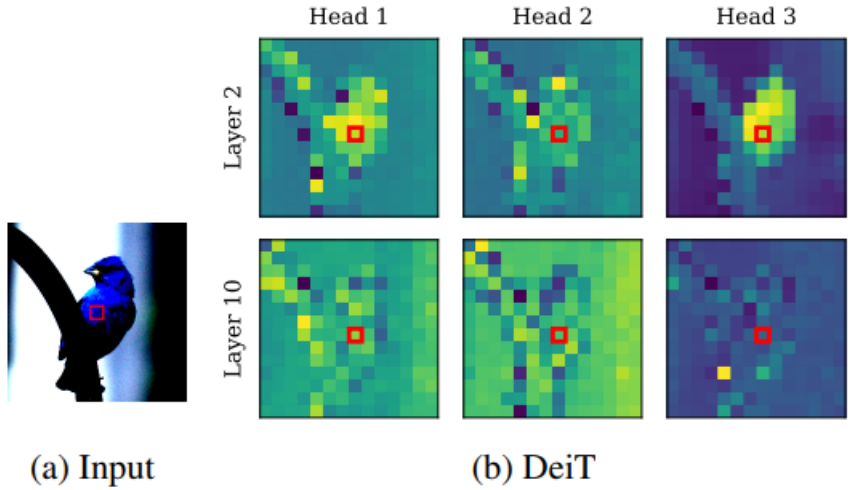
- SA layers try to become local, GPSA layers escape locality
- the higher, the further the attention heads look from the query pixel.
- The final nonlocality does not increase monotonically throughout the layers

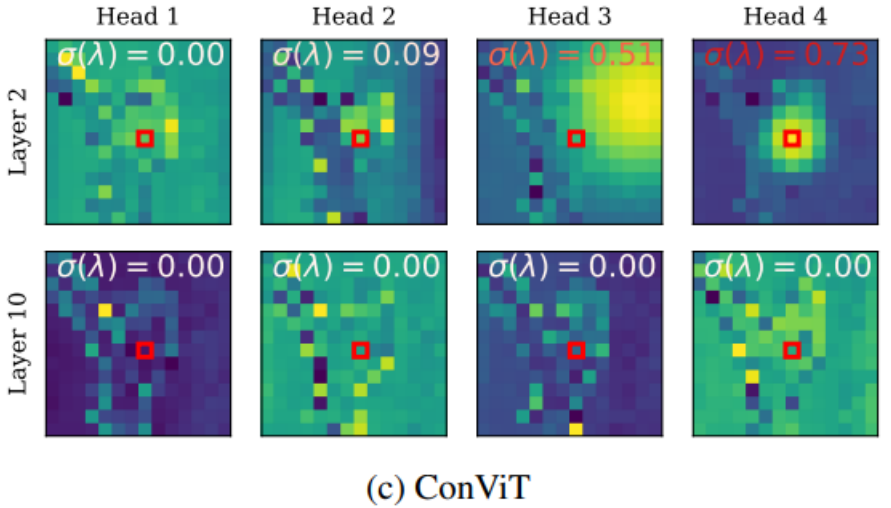
Investigating the role of locality



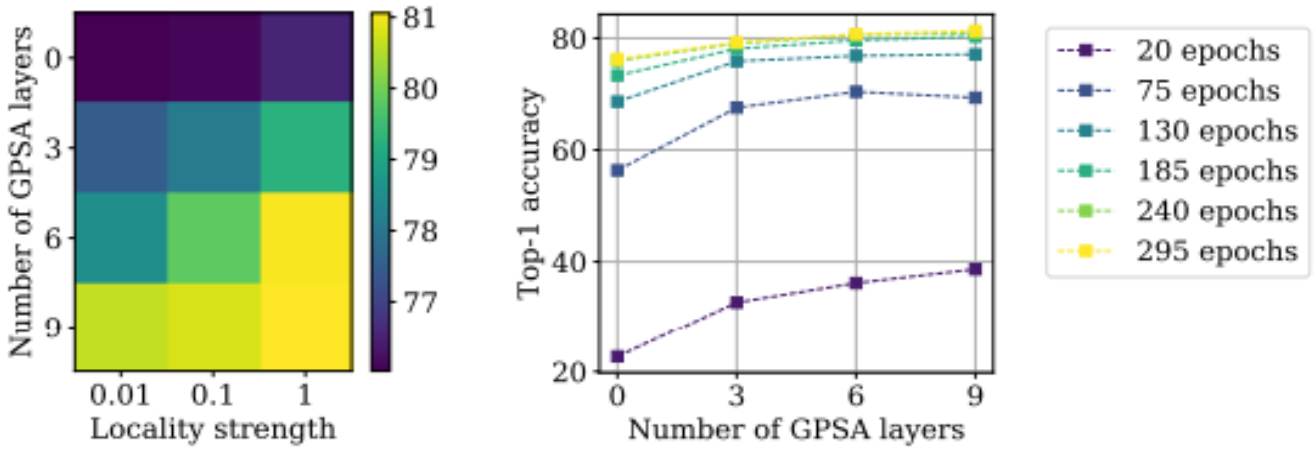
$\sigma(\lambda_h)$

- The colored lines quantify how much attention head h pays to positional information versus content
- The black line represents the value averaged over all heads
- Positional information이 훈련이 지속될 수록 점점 무시되는 경향이 있음
- 일부 head에서만 positional information의 정보를 취하기 위해 높은 $\sigma(\lambda_h)$를 유지





Investigating the role of locality



ImageNet의 첫 100개 class

- Locality strength, α determines how focused the heads are around their center of attention
- (left) Locality strengt와 GPSA layer의 수에 따라 최종 테스트 정확도가 증가 (=convolution이 많을 수록 증가)
- (right) GPSA레이어의 수가 많을 수록 성능이 더 높게 나타나는데 특히 훈련 초기에 두드러짐 (=convolution 초기화가 모델에 '유리한 출발'을 제공한다는 것을 알 수 있음)

Conclusion and perspective

- Convolution 제약조건의 장점을 취함으로써 model 크기 증가나 tuning 없이 trainability와 sample efficienc를 개선시킴

- Convolution layer를 단순히 결합하여 SA layer에 interleaving시키는 대신, gating parameter를 조정하여 convolution 여부를 스스로 결정하도록 함

- Another direction which will be explored in future work is the following: if SA layers benefit from being initialized as random convolutions, could one reduce even more drastically their sample complexity by initializing them as pre-trained convolutions?