Vision Transformer (ViT)

문학준

gloriel621@gmail.com

21.03.2023

Contents

- 1. Introduction
- ~~2. Previous works~~
- 3. Methods
- 4. Experiments
- 5. Conclusion

1. Introduction

Transformer는 자연어 처리에서 큰 성공을 거둠. 따라서 이미지 처리 분야에도 이를 적용해 보려 함.

이전의 Computer Vison Task에서 Self Attention 메커니즘을 시도하였으나 한계가 있었음.

따라서 기존의 Transformer 구조를 최대한 그대로 적용하려고 함.

Transformer 의 장점: 계산 효율성(Efficiency) 및 확장성(Scalability)

100B Parameter도 학습 가능.

데이터셋이 커져도 모델을 크게 하면 되고, Saturation(포화) 되지 않음

Vision Transformer 학습

이미지를 Patch로 분할 후 Sequence로 입력, NLP에서 단어(Word)가 입력되는 방식과 동일

(∵ "IMAGE IS WORTH 16X16 WORDS")

Supervised Learning 방식.

Vision Transformer 의 특징

ImageNet와 같은 Mid-sized 데이터셋으로 학습 시, ResNet보다 낮은 성능을 보임

JFT-300M 사전 학습 후, Transfer Learning → CNN보다 좋은 성능 달성(SOTA)

Inductive bias가 없으므로, CNN의 특성인 locality와 Translation Equivariance이 없음

따라서, Robustness는 높지만 많은 데이터를 사용하여 학습해야 함.

Component	Entities	Relations	Rel. inductive bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

Table 1: Various relational inductive biases in standard deep learning components. See also Section 2.

Inductive Bias

만나지 못한 상황을 해결하기 위해 사용하는 가정

일반적으로, 학습 대상의 특징을 inductive bias 로 사용

CNN의 inductive bias는 Locality of pixel dependencies (픽셀끼리만 연관성을 가짐),

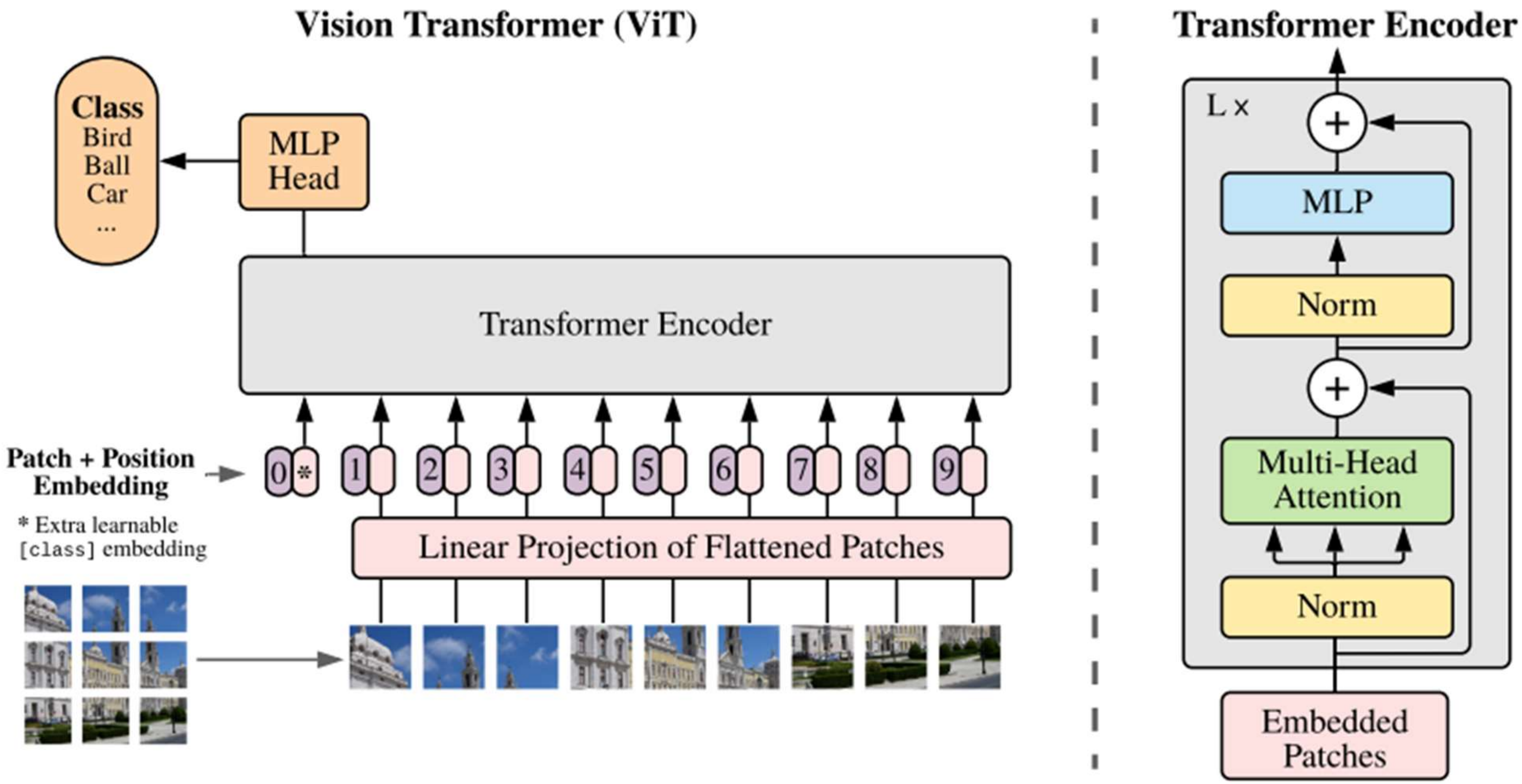
RNN의 inductive bias는 Sequentiality(순차성)

Translation Equivariance

이미지에서 객체의 위치가 달라져도 같다고 분류하는 것

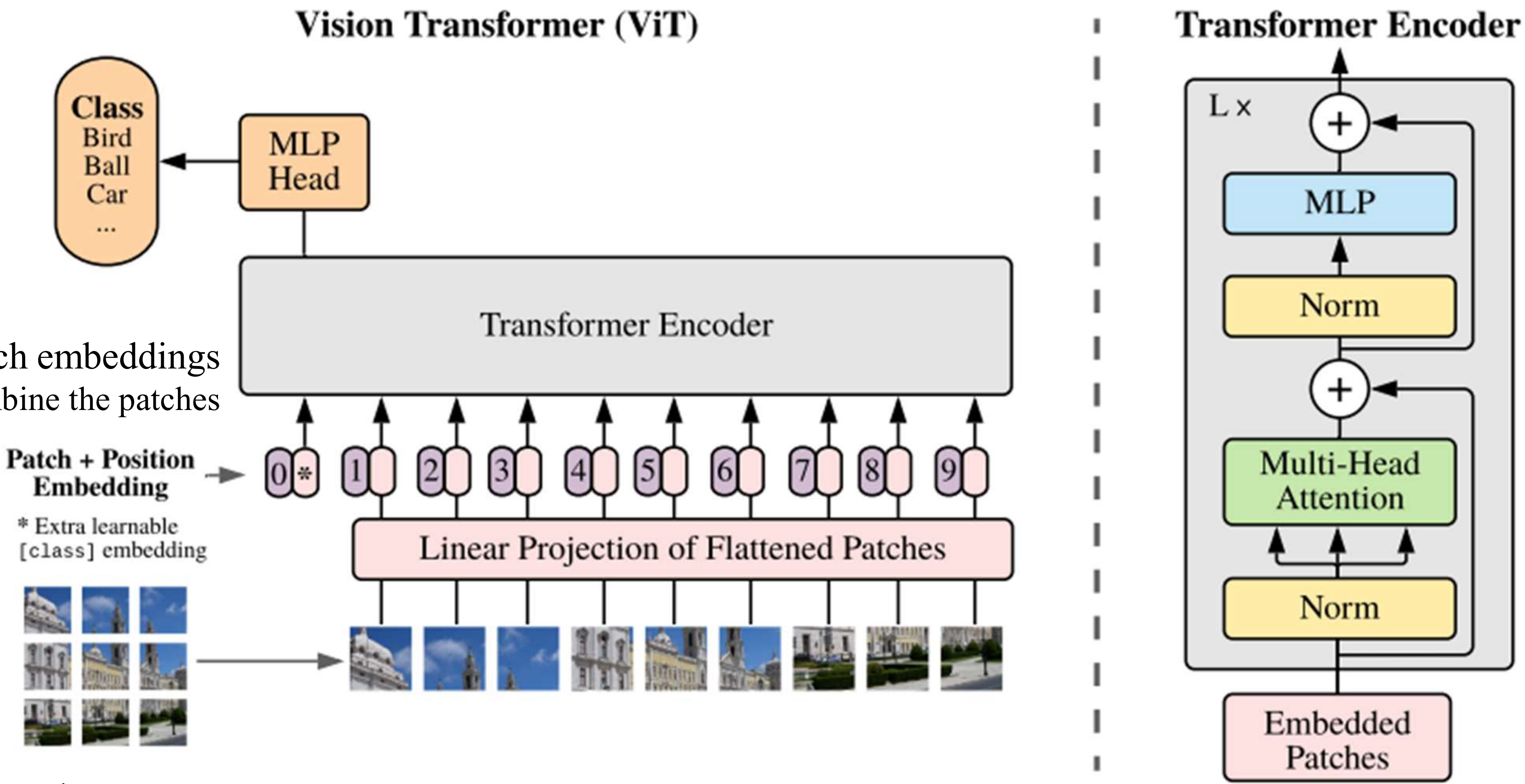
CNN은 maxpooling과 softmax의 사용으로 인해 Translation Equivariance 를 가짐

3. Methods



1. Make image patches for 1d embeddings

* Able to use CNN feature maps instead of images

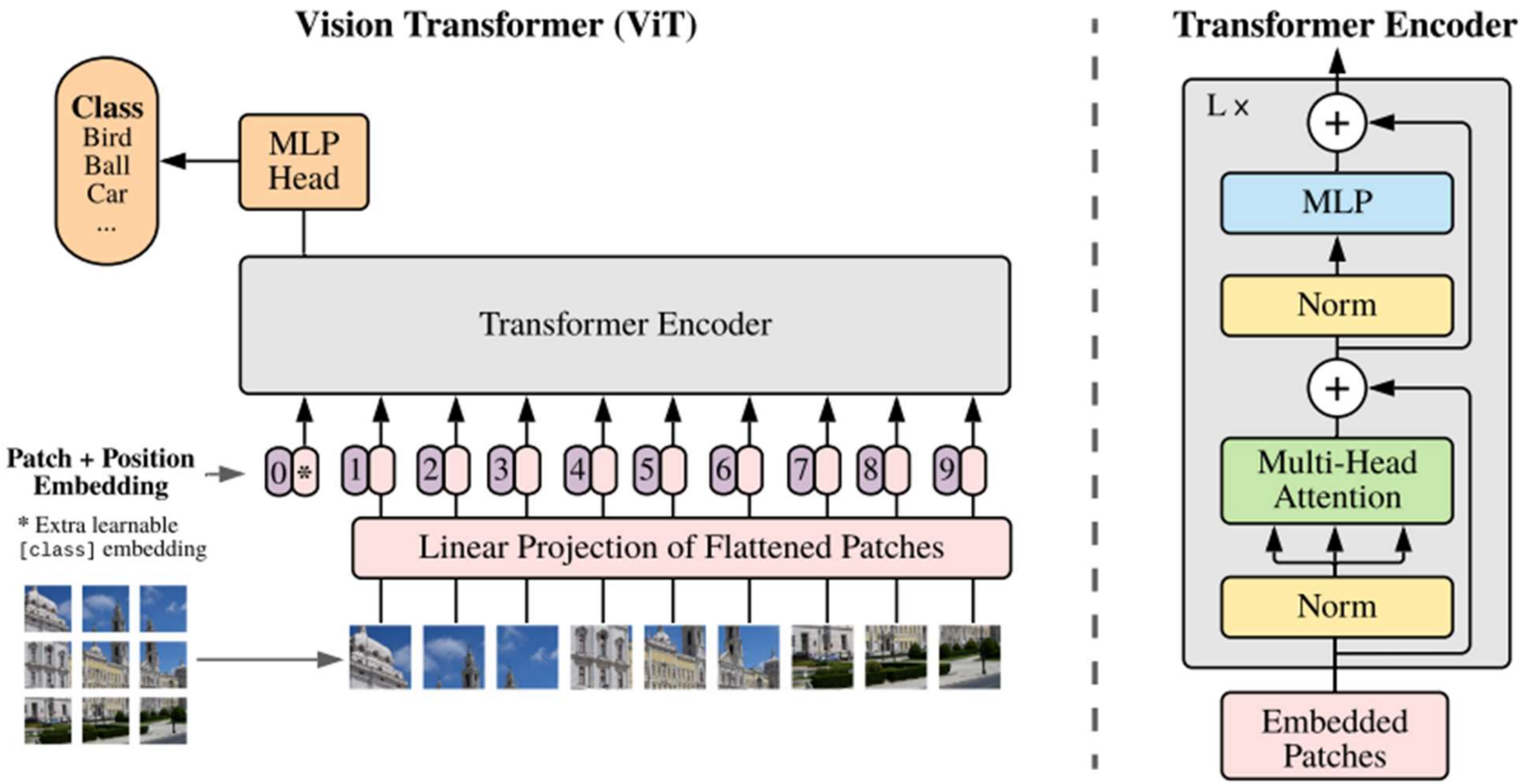


2-1. Make patch embeddings

* re-combine the patches

2-2. Make class tokens

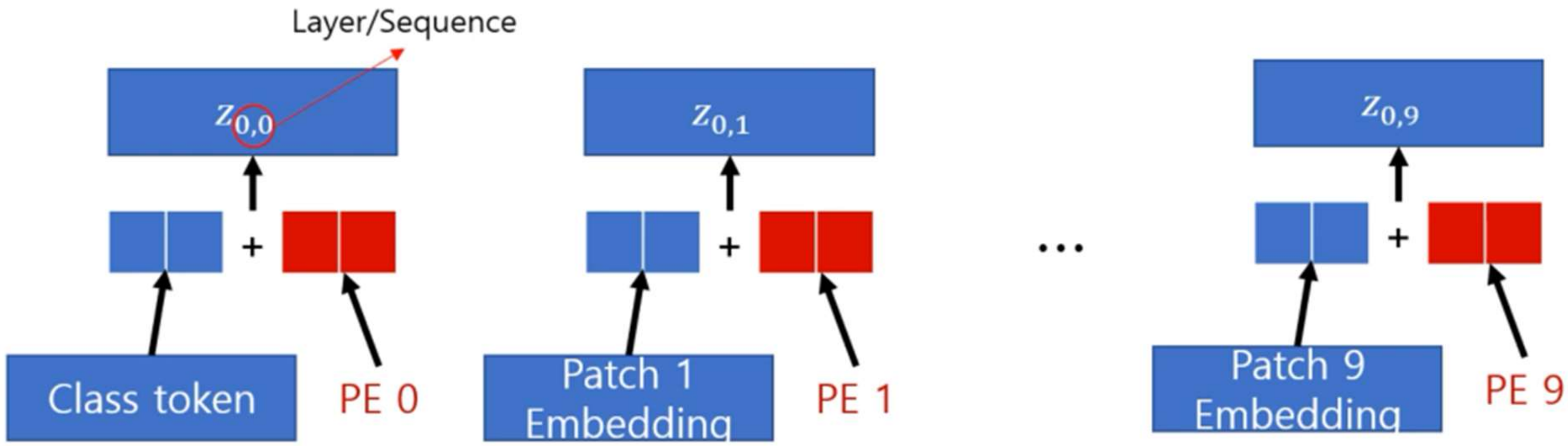
* Similar to CLS token of BERT, add a learnable "class token"
* Works as a classification label of previous



2-3. Make position embedding

* Added 1d position embeddings for positional information
* 2d embeddings did not have significant performance gains compared to 1d

Input Example



MSA: Multi-Head Self Attention

$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}};\, \mathbf{x}_p^1\mathbf{E};\, \mathbf{x}_p^2\mathbf{E};\, \cdots;\, \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D},\, \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D} \tag{1}$$

$$\mathbf{z'}_\ell = \mathrm{MSA}(\mathrm{LN}(\mathbf{z}_{\ell-1})) + \mathbf{z}_{\ell-1}, \qquad \ell = 1 \ldots L \tag{2}$$

$$\mathbf{z}_\ell = \mathrm{MLP}(\mathrm{LN}(\mathbf{z'}_\ell)) + \mathbf{z'}_\ell, \qquad \ell = 1 \ldots L \tag{3}$$

$$\mathbf{y} = \mathrm{LN}(\mathbf{z}_L^0) \tag{4}$$

MLP: Multi Layer Perceptron
(=FCN + linear transformation in each hidden layer)

LN: Linear Normalization

4. Experiments

4.1 Experiment setup

Pre-Train

dataset

1. ImageNet - 1k (1.3 M)
2. ImageNet - 21k (14 M)
3. JFT - 18k (303M)

Hyperparameters

- Optimizer : **Adam**
- Batch Size : 4096
- Weight Decay : 0.1 ~

Transfer learning

dataset

1. ImageNet with cleaned-up labels (Beyer et al., 2020)
2. CIFAR-10/100 (Krizhevsky, 2009)
3. Oxford-IIIT Pets (Parkhi et al.,2012)
4. Oxford Flowers-102 (Nilsback & Zisserman, 2008)
5. 19-task VTAB classification suite (Zhai et al., 2019b)

Hyperparameters

- Optimizer : SGD
- Batch Size : 512
- Resolution : ViT-L/16 - 512, ViT-H/14 - 518

Model	Layers	Hidden size D	MLP size	Heads	Params
ViT-Base	12	768	3072	12	86M
ViT-Large	24	1024	4096	16	307M
ViT-Huge	32	1280	5120	16	632M

Table 1: Details of Vision Transformer model variants.

ViT-L/N

- (N as size of patches, inverse proportion to sequence length and computational complexity)

ResNet modified (BiT)

- Batch Normalization을 Group Normalization으로 변경

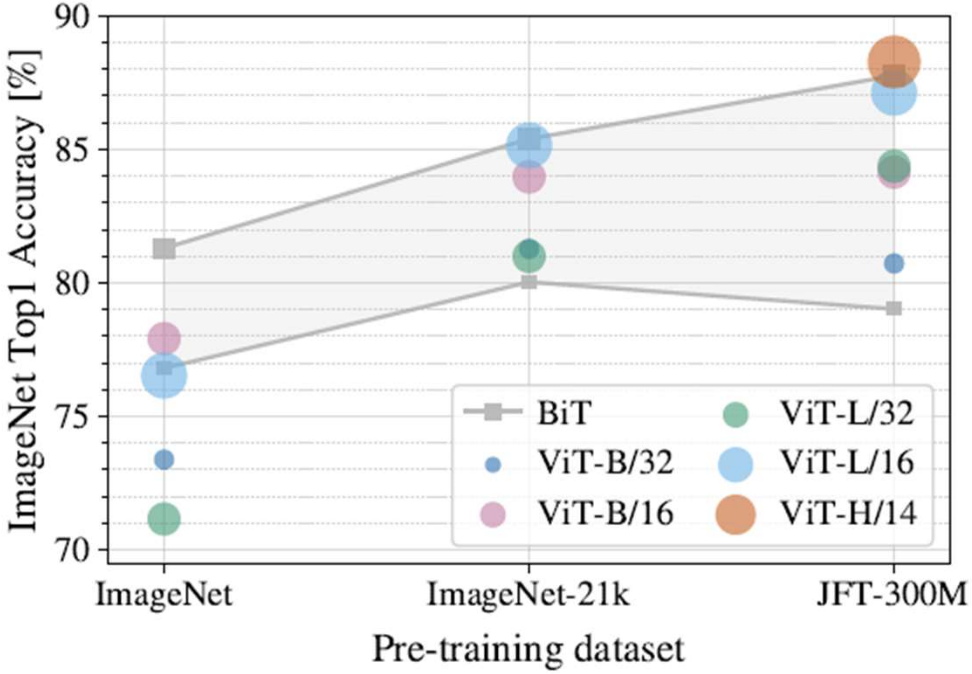
Hybrid

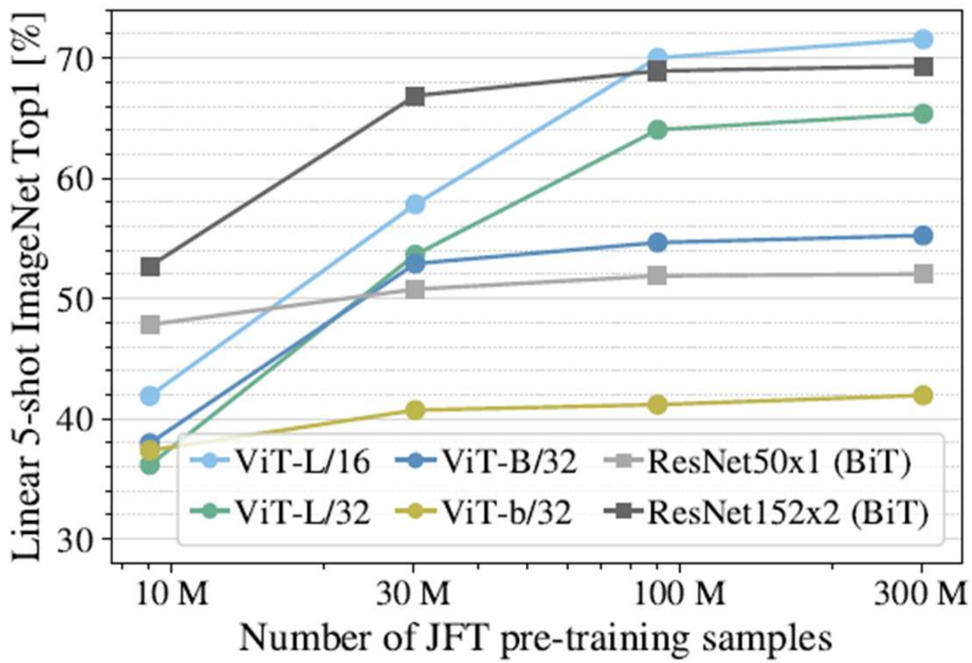
- ResNet의 Intermediate Feature Map을 입력으로 사용 → Patch Size 1x1
- ResNet with different sequence lengths

4.2 Comparison to state of the art

	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

4.3 Performance at pre-training



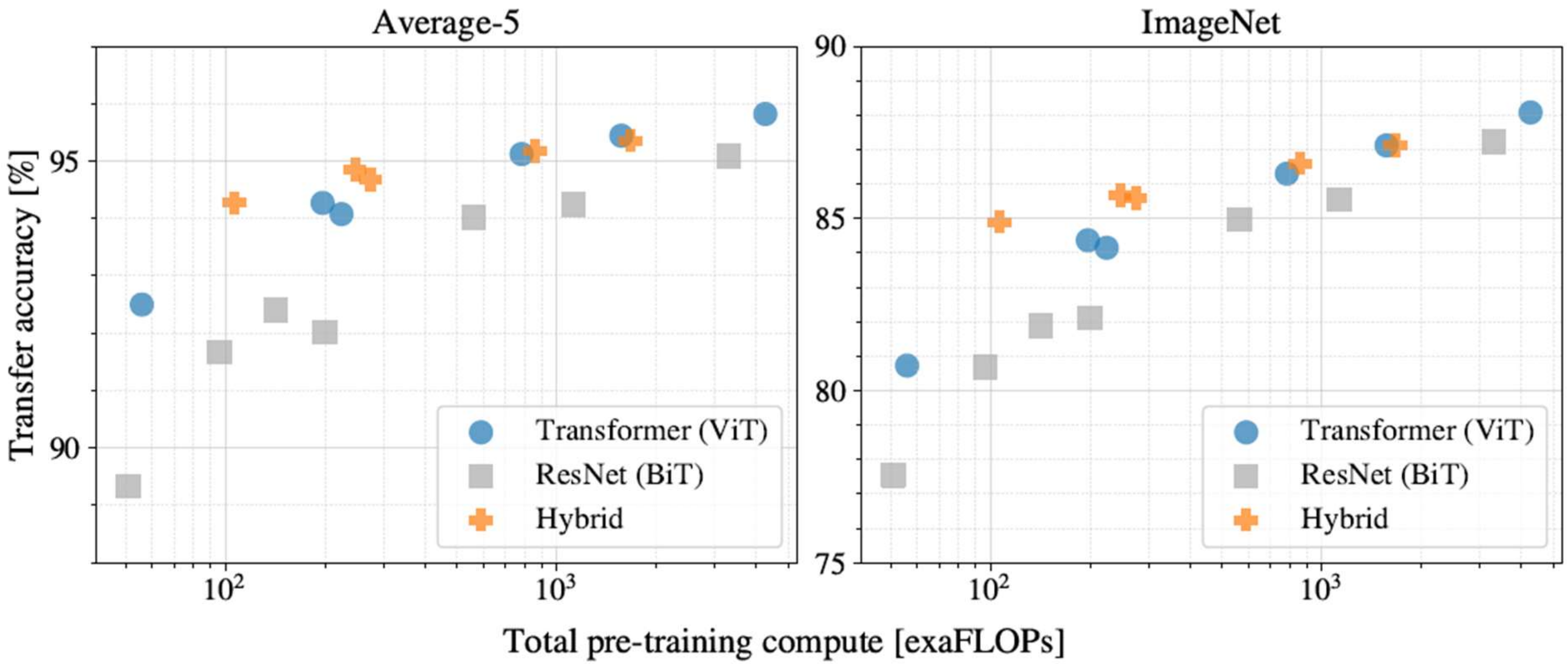


ImageNet-1k(1.3 M) / ImageNet-21k(14 M) / JFT-18k(303M)

Pre-training에서 set이 클수록 ViT가 좋고, 작으면 좋지 않음

원인: No image-specific inductive bias

4.4 Scaling study : Size of models

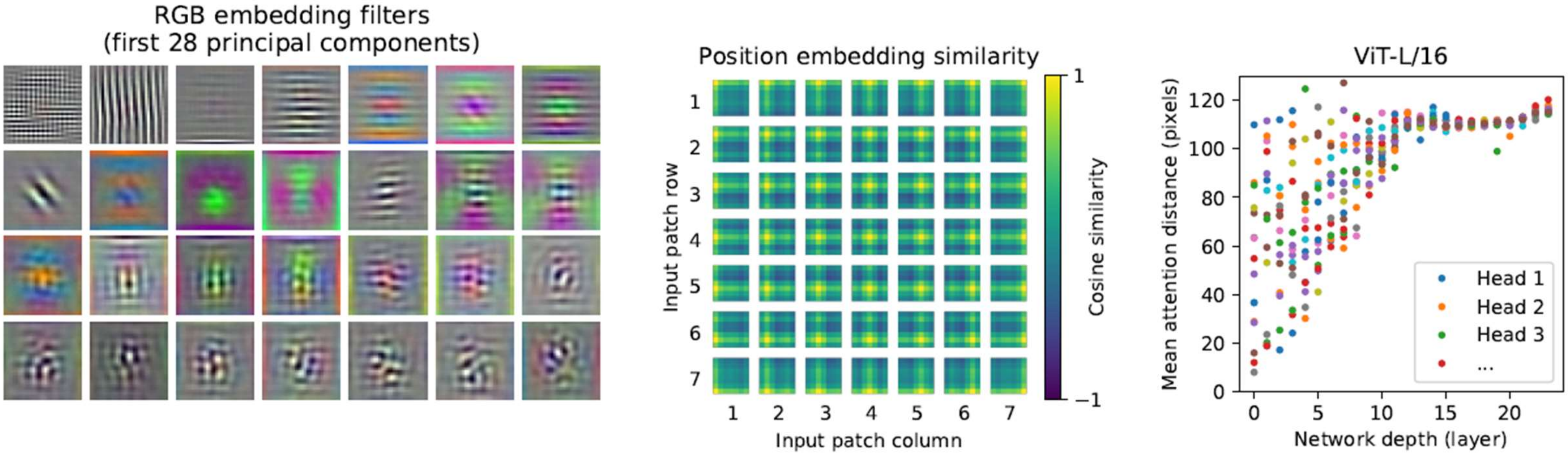


x축 : pre-training의 computation cost / y축 : accuracy

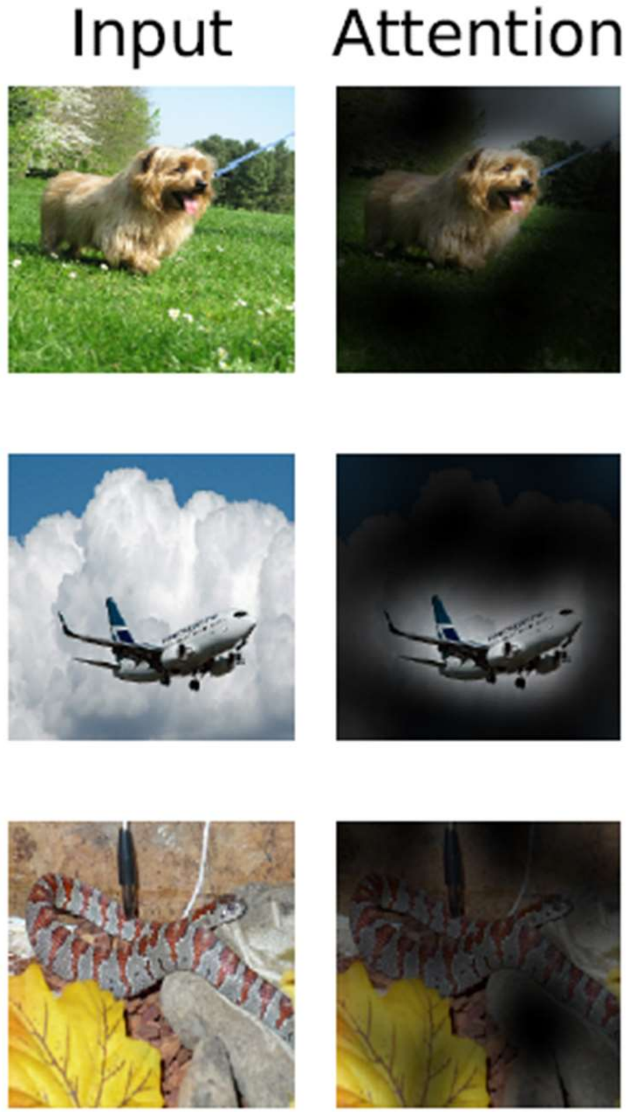
ViT > BiT

ViT의 성능이 포화(Saturate) 되지 않음 → 성능이 더 좋아질 수 있음

4.5 Inspecting ViT



4.5 Inspecting ViT: attention map



4.6 Self supervision learning

NLP Task에서 수행되는 Self-supervision 학습 방법을 시도

- BERT는 Input을 Masking후, Masking 한 단어를 올바르게 예측하도록 학습(Self-Supervised Learning)
- ViT에서는 input patch 하나를 masking 후 이 patch를 예측하도록 학습

Vision Self-Supervision 결과

- ViT-B/16 모델은 79.9 % 정확도를 보이지만, Supervised Learning 방식보다는 낮음

5. Conclusion

Conclusion:

1. “Image-specific Inductive Bias” 가 없는 Self-Attention 적용
2. Large Dataset (JFT-300M)에서 정확도가 높음, 사전 학습 비용이 상대적으로 저렴

Future works:

1. Detection and Segmentation
2. Self-Supervised Learning
3. Scaling으로 추가적인 성능 향상 기대

TRAIN AND TEST