AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy∗,†, Lucas Beyer∗, Alexander Kolesnikov∗, Dirk Weissenborn∗, Xiaohua Zhai∗, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer,

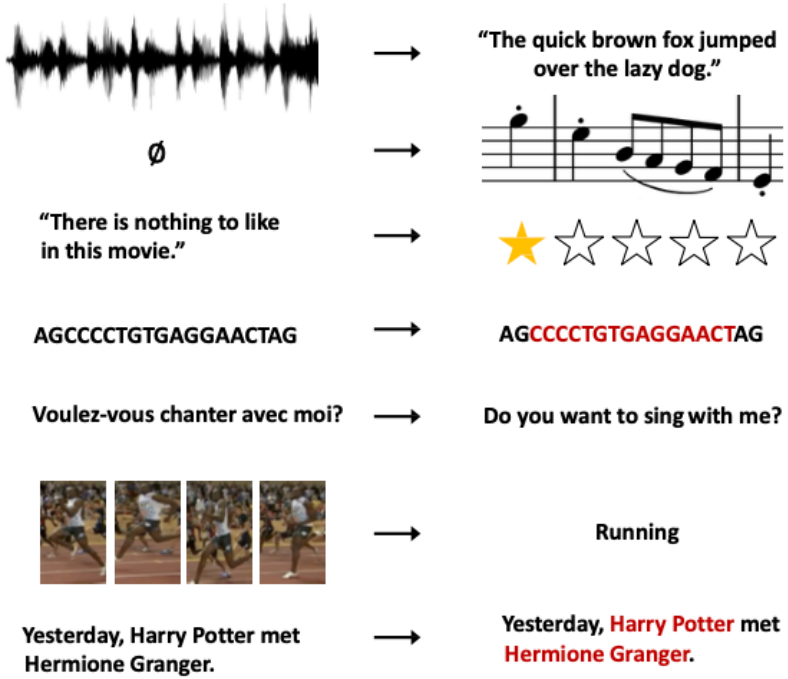
Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby∗,† ∗equal technical contribution, †equal advising
Google Research, Brain Team

육현준

0. What is Transformer?

0.1 Attention

- Sequence Data
  - Speech Recognition:
  - Music Generation:
  - Sentiment Classification:
  - DNA Sequence Analysis:
  - Machine Translation:
  - Video Activity Recognition:
  - Name Entity Recognition:

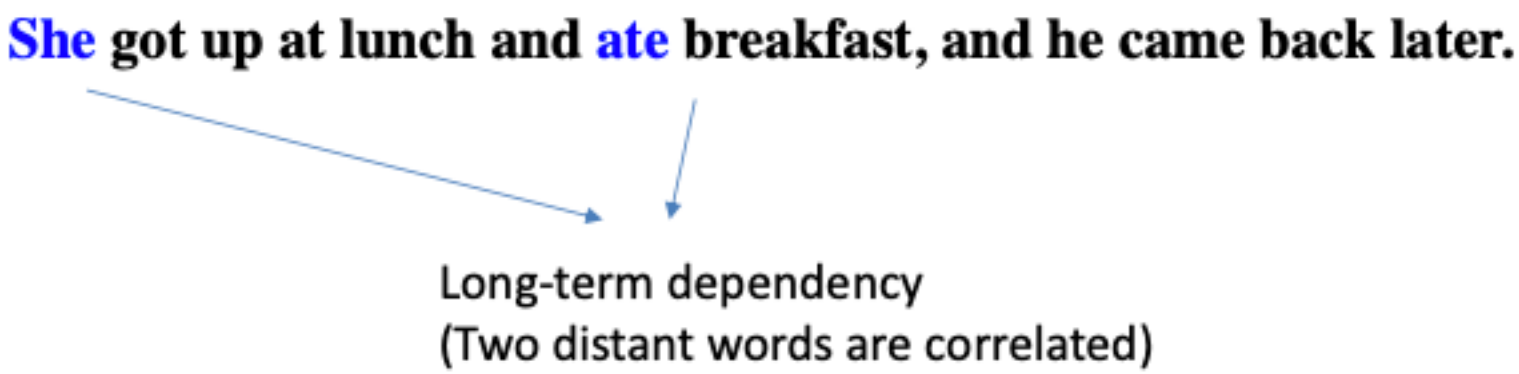


RNN
LSTM
GRU

0. What is Transformer?

0.1 Attention

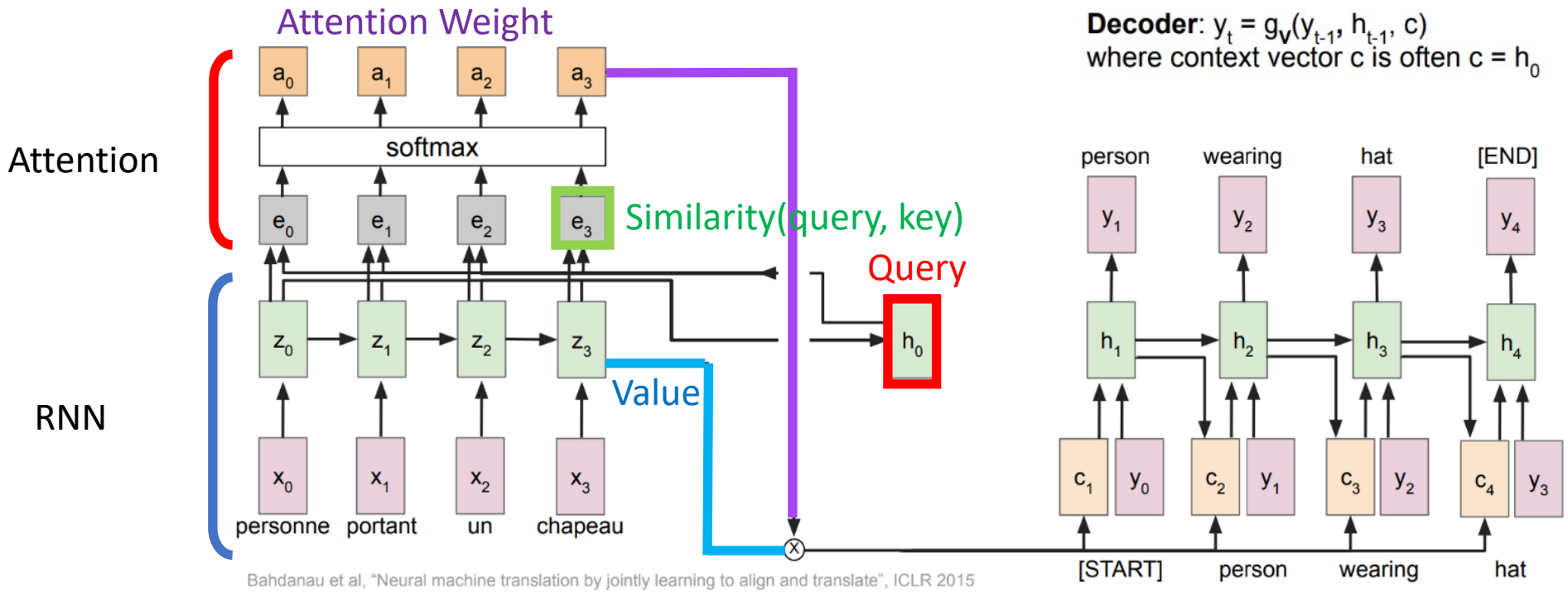
- RNN, LSTM, GRU의 한계
  - Gradient Vanishing
  - Long Term Dependency



0. What is Transformer?

0.1 Attention

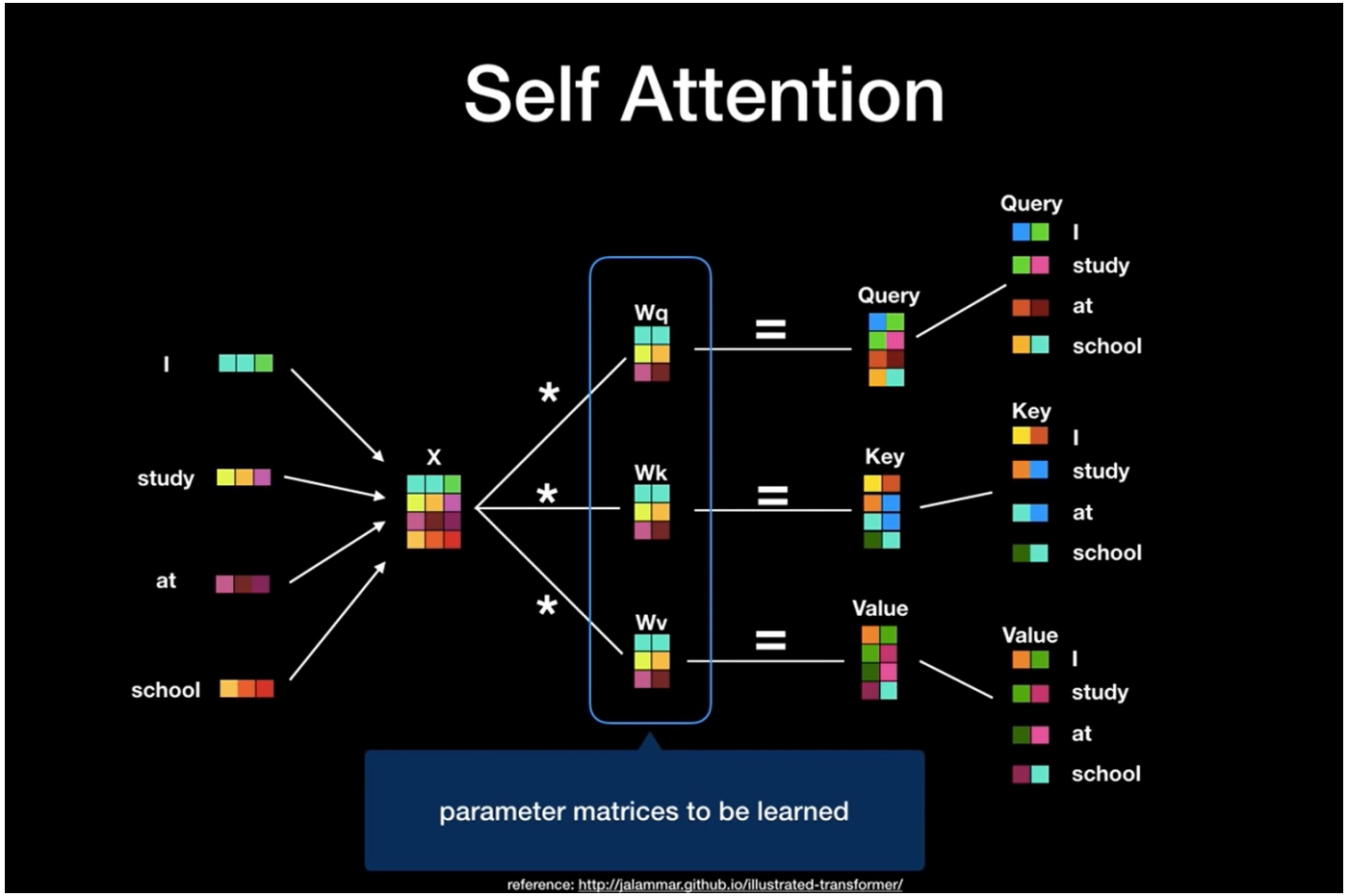
- Key, Query, Value
  - Similarity function : dot product → $V^T \tanh([W_k\ W_q]^T [k\ q])$
  - Attention Weight : $a_0, a_1, a_2, a_3 = softmax(e_0, e_1, e_2, e_3)$



0. What is Transformer?

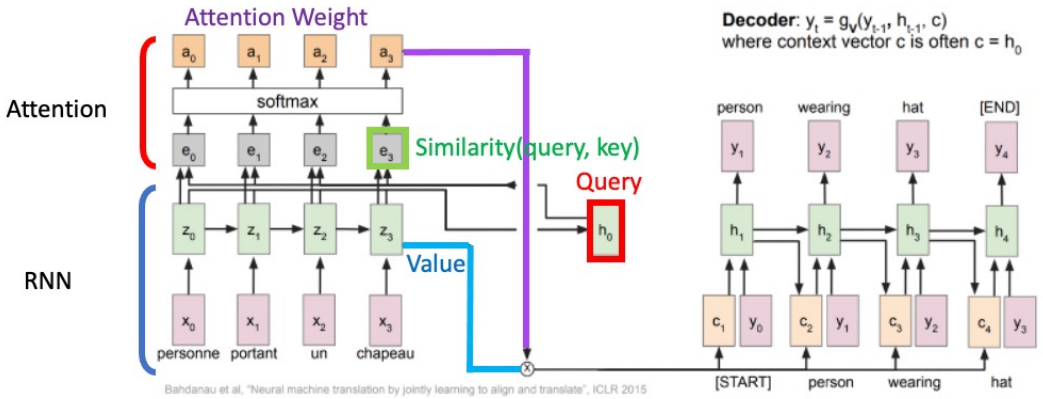
0.2 Self-Attention

- Key = Query = Value



0. What is Transformer?

0.2 Self-Attention



0. What is Transformer?

0.2 Self-Attention

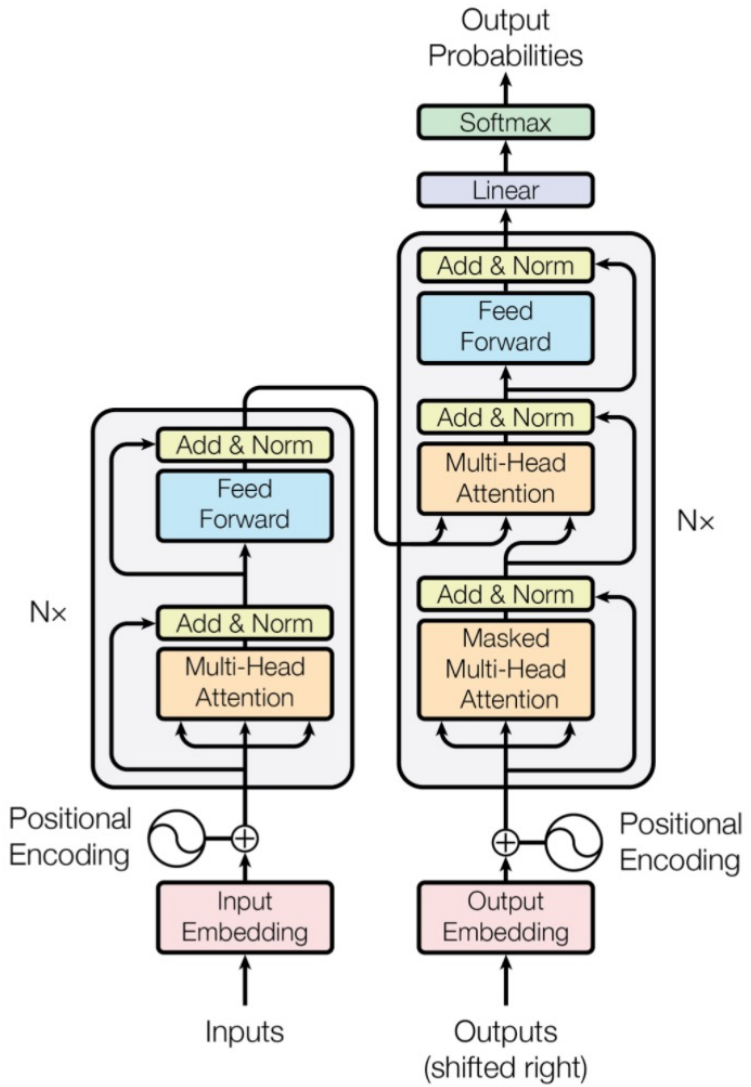
- Attention
- Time-step을 활용 O
  - 이전 step에서 계산된 vector를 사용한다.

- Self-Attention
- Time-step 활용 X
  - 한번에 모든 단어에 대해서 attention score 계산
- 일반 attention보다 Long Term Dependency가 더 좋다.
- Run in parallel

0. What is Transformer?

0.3 Transformer

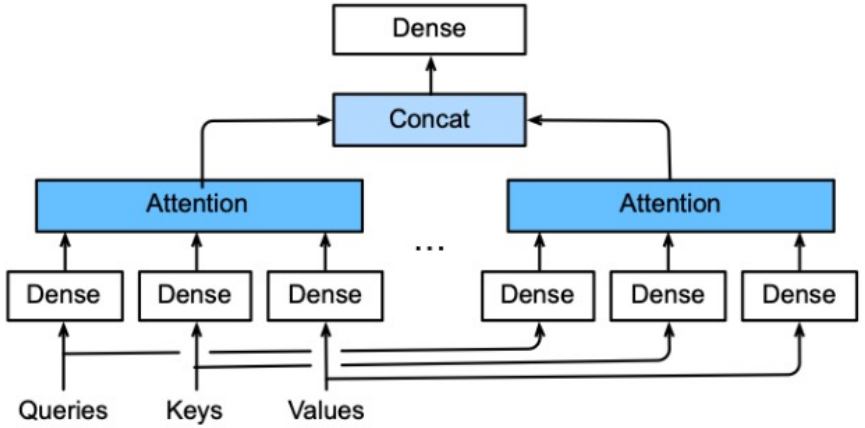
- Multi-Head Attention
  - Self-Attention을 병렬적으로 처리
- Norm
  - Layer Norm
- Positional Encoding



0. What is Transformer?

0.3.1 Multi-Head Attention

- Multi-Head Attention
  - Self-Attention을 병렬적으로 처리

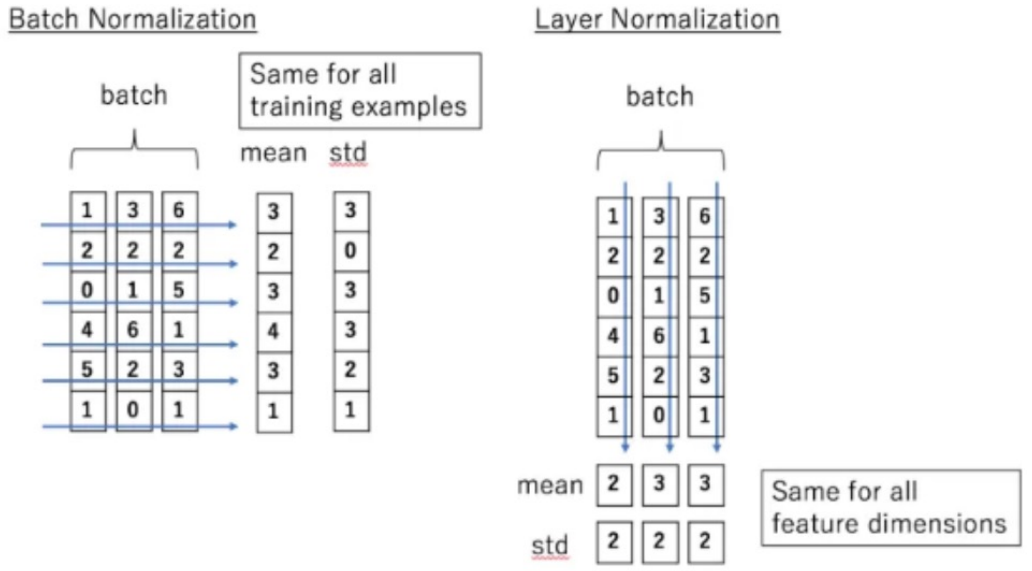


0. What is Transformer?

0.3.2 Layer Norm

- Layer Norm
  - Batch Norm : mini-batch의 개수로
  - Layer Norm : neuron의 개수로

- 장점
  - 작은 mini-batch를 갖는 RNN task에 좋은 성능
  - 입력 데이터 scale에 강건



0. What is Transformer?

0.3.3 Positional Encoding

- Positional Encoding
  - P ∈ $\mathbb{R}^{lxd}$
    - $l = number\ of\ token$
    - $d = feature\ dimension$
  - $P_{i,2j} = \sin(\frac{i}{10000^{2j/d}})$
  - $P_{i,2j+1} = \cos(\frac{i}{10000^{2j/d}})$
  - Output = X + P

1. Introduction

1.1 Background

- Transformer[1]의 등장 이후 자연어 처리(NLP)에서 SOTA 달성
  - High Computational efficiency & Scalability
  - GPT, BERT
- Computer Vision에서는 아직 CNN이 지배적

1.2 Problem

- Computer Vision에 Transformer를 활용할 수 있을 것인가?

1)Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).

1. Introduction

1.3 Difficulties

- Locality ↓
- Translation Equivariance ↓

1. Introduction

1.4 Solution

- 대량의 Data → Pre-Train
  - Inductive Bias ↑ : 부족한 Locality와 translation equivariance 문제 해결

2. Related Work

2.1 CNN + Attention

- Naïve self-attention을 사용
  - 모든 pixel에 접근
  - Quadratic Cost

2. Related Work

2.1 CNN + Attention

- 다양한 CNN + self-attention 방법을 시도
  - 하드웨어 가속기에서 효율적인 수행이 어려움

- ResNet[2]의 성능을 따라가지 못함

2)He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

2. Related Work

2.1 CNN + Attention

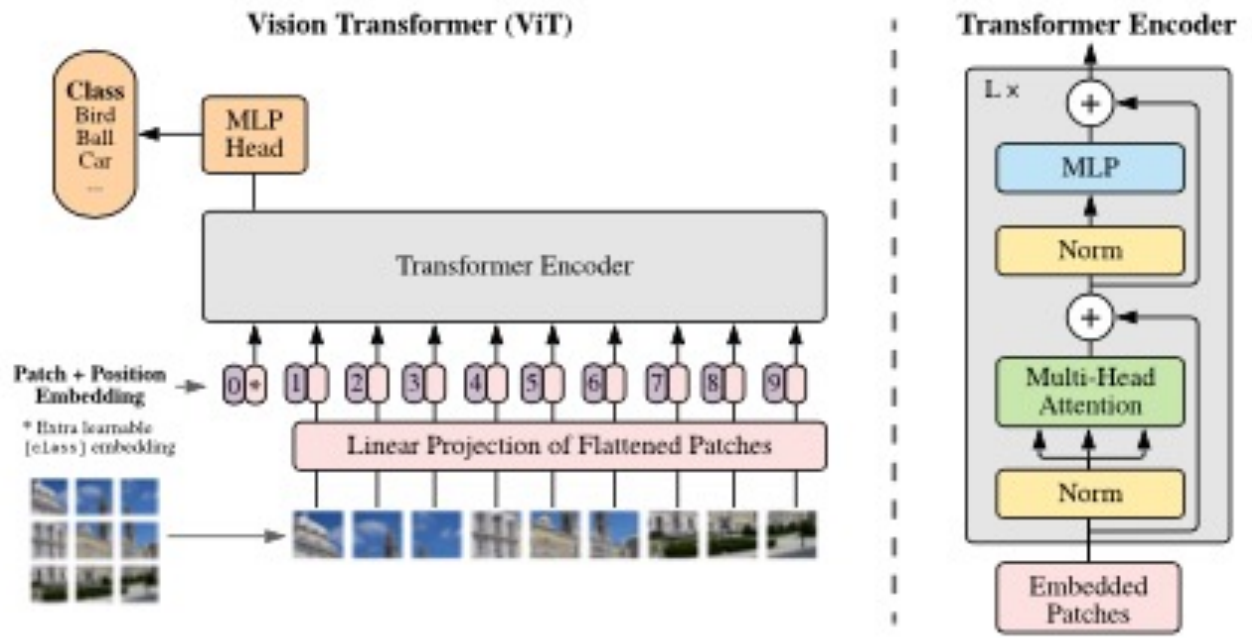
- 2x2 patch로 자른 후 Self-Attention 적용
  - Patch의 크기가 너무 작아 저해상도 이미지만 적용이 가능

- ImageGPT
  - 이미지의 해상도와 color space를 줄인 후 pixel 단위로 transformer 적용
  - imageNet에서 최대 72% Accuracy

3. Proposed Idea

3.1 Vision Transformer

- ImageNet보다 더 큰 데이터셋으로 Pre-Train
- NLP에서 사용된 Standard Transformer를 image에 직접 적용



3. Proposed Idea

3.2 Patch

- Image를 patch단위로 Tokenize → 1D Squence Data

$X \in \mathbb{R}^{HxWxC}$

$X_p \in \mathbb{R}^{N(p^2C)}$

P : resolution of each image patch

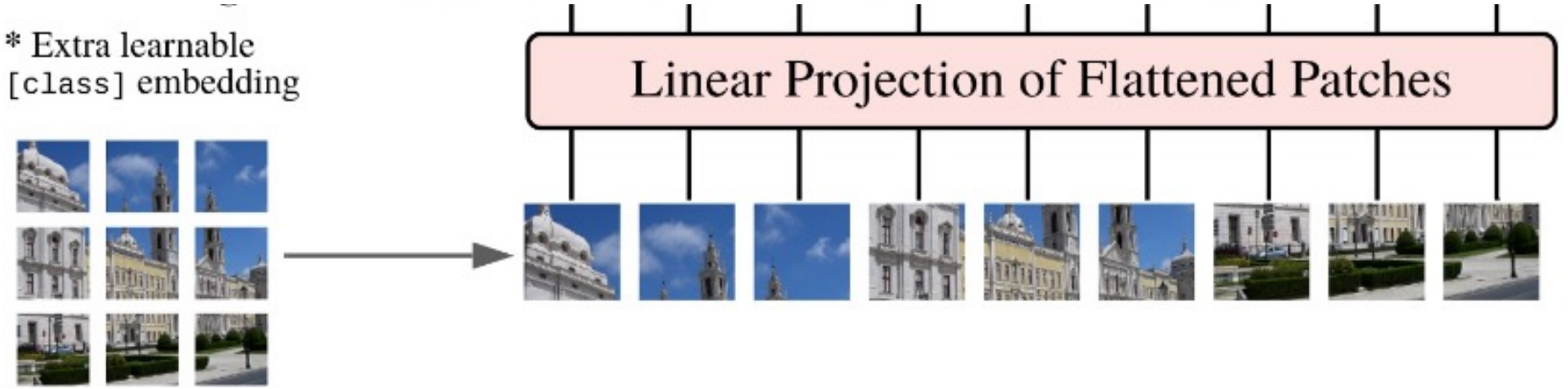
N : $\frac{H \cdot W}{p^2} = number\ of\ patches$

2)He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

3. Proposed Idea

3.2 Patch Embedding

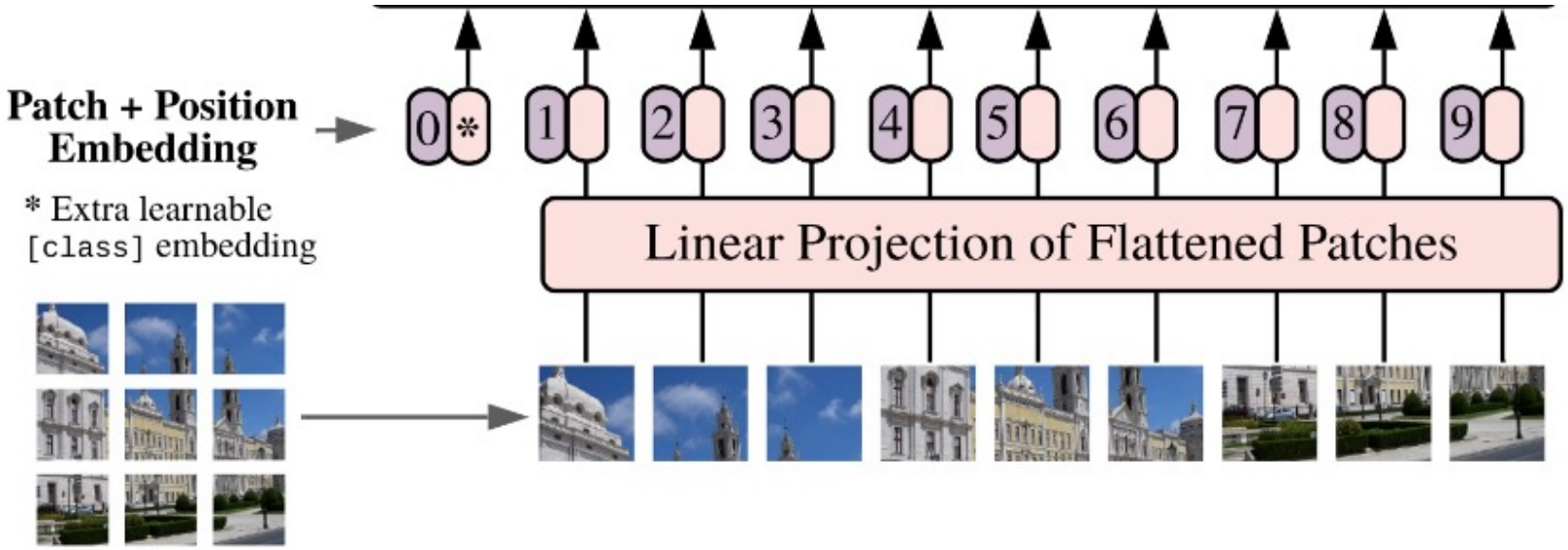
- Patch → flatten → D차원으로 mapping
  - Trainable Linear Projection



3. Proposed Idea

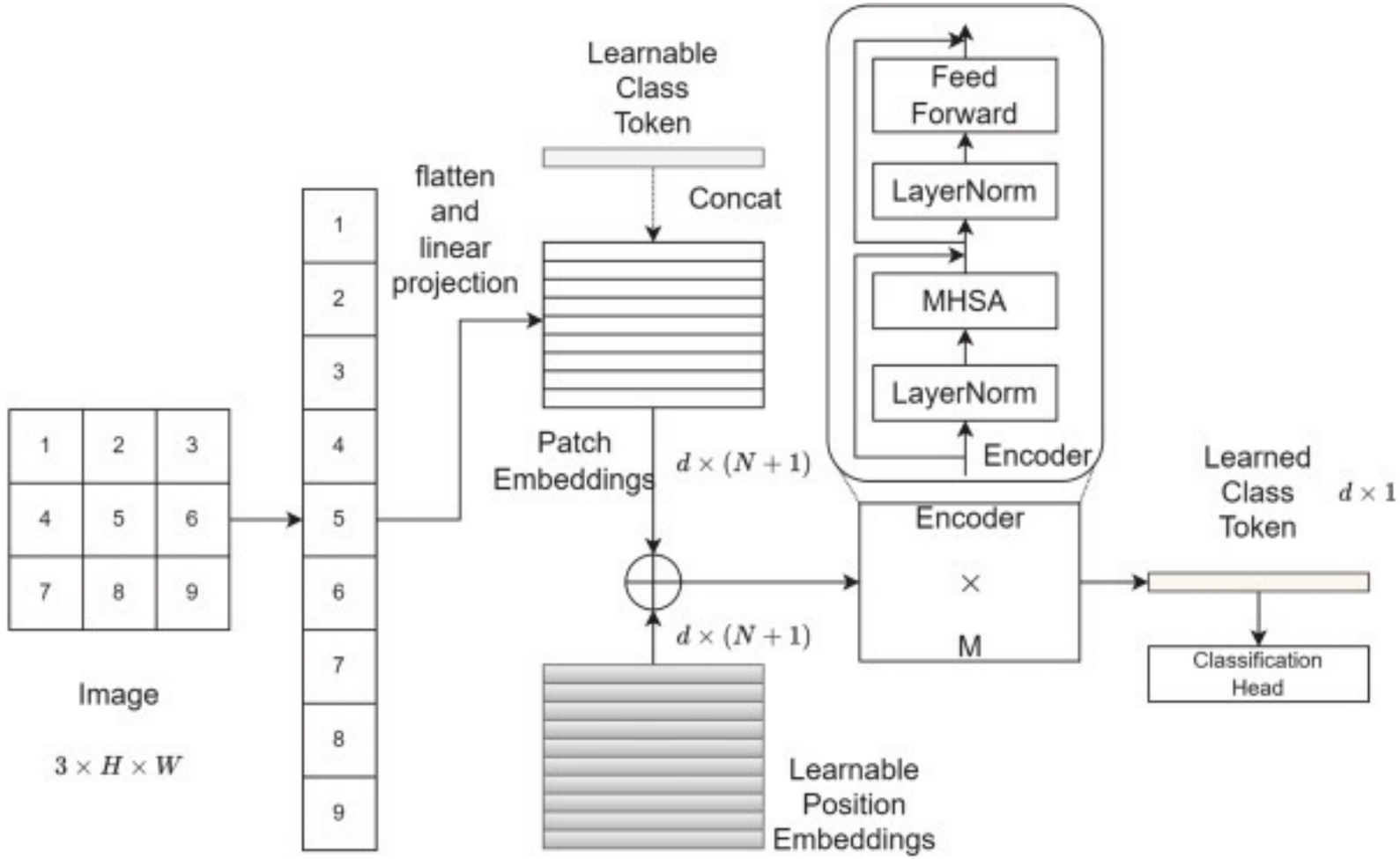
3.3 Position Embedding

- 위치 정보를 주기 위해 position embedding
  - 2D position embedding 실험 결과 성능 향상 X → 1D position embedding 사용



3. Proposed Idea

3.4 Encoder Input

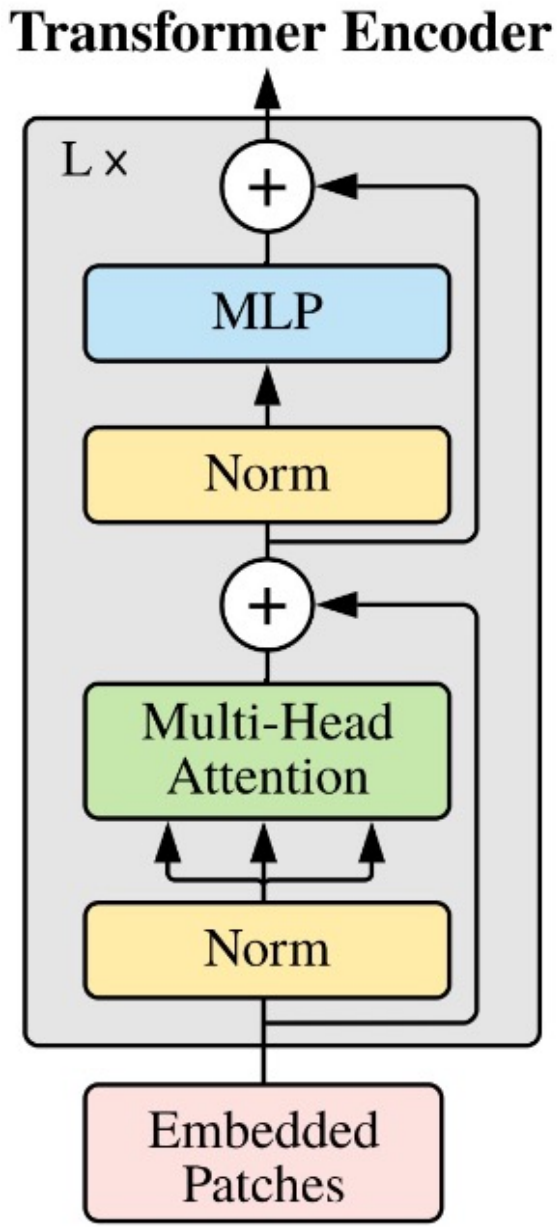


$$\mathbf{z}_0 = [\mathbf{x}_{\text{class}}; \mathbf{x}_p^1\mathbf{E}; \mathbf{x}_p^2\mathbf{E}; \cdots; \mathbf{x}_p^N\mathbf{E}] + \mathbf{E}_{pos}, \qquad \mathbf{E} \in \mathbb{R}^{(P^2 \cdot C) \times D}, \ \mathbf{E}_{pos} \in \mathbb{R}^{(N+1) \times D}$$

3. Proposed Idea

3.5 Transformer Encoder

- Multi-Head Attention
- Layer Norm
- Residual connection
- MLP
  - 2 Layer
  - GELU Activation Function 사용



3. Proposed Idea

3.5 Transformer Encoder

- GELU Activation Function
  - Gaussian Error Linear Unit

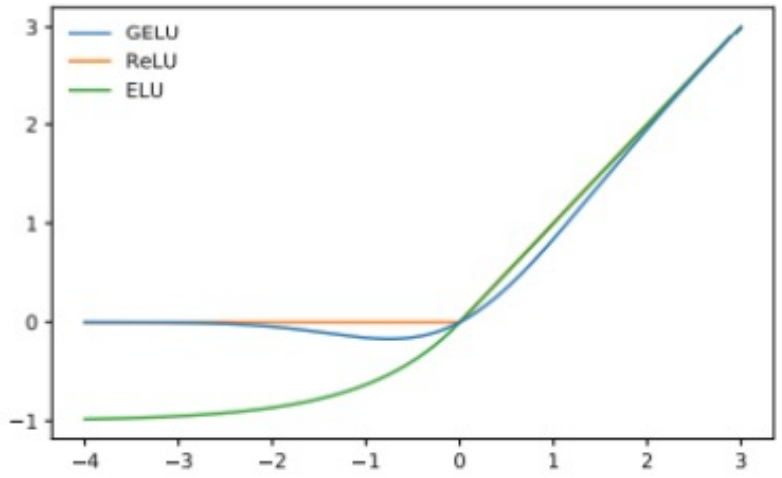


Figure 1: The GELU ($\mu = 0, \sigma = 1$), ReLU, and ELU ($\alpha = 1$).

$$0.5x(1 + tanh[\sqrt{2/\pi}(x + 0.044715x^3)])$$

3. Proposed Idea

3.6 Inductive Bias

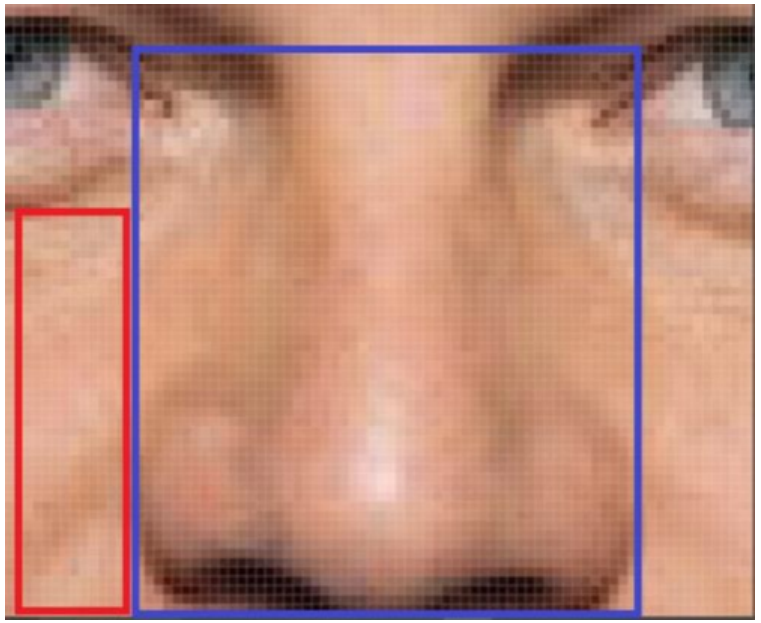
- Locality
- 2D neighborhood structure
- Translation equivariance

- How Inductive Bias operates on ViT
  - MLP : 유일하게 locality와 translation equivariance하다.
  - 2D neighborhood structure
    - 입력패치로 자르는 과정을 학습
    - positional embedding을 fine-tune 과정에서 조정

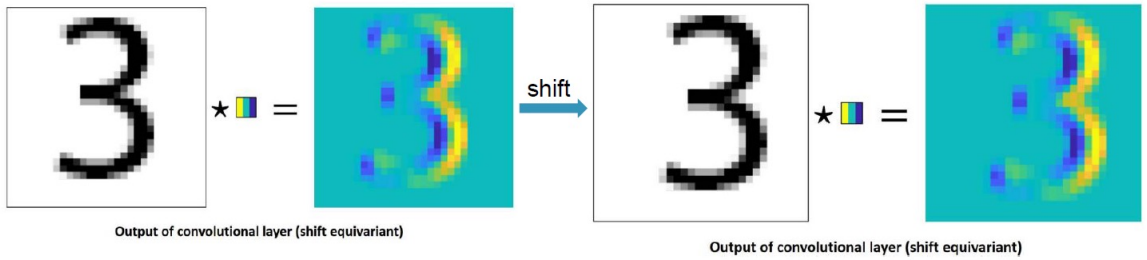
3. Proposed Idea

3.6 Inductive Bias

- Locality



- Translation Equivariance



3. Proposed Idea

3.7 Hybrid Architecture

- Feature map → patch로 변환
  - Patch 크기를 1로 설정
  - 다른 변환 없이 Flatten → D 차원 변환
  - Position, patch embedding 후 encoder 입력으로 사용

3. Proposed Idea

3.8 Fine-Tune And Higher Resolution

- Fine-Tune
  - Pre-train된 prediction head를 제거
  - Zero-initialized (DxK) layer를 추가
    - K : class 개수

- Fine-tune과정에서 pre-train 데이터보다 높은 resolution data 사용
  - 이 때 pre-train과 동일한 patch크기를 사용하면 sequence의 길이가 증가
  - Position embedding이 무의미
  - 2D interpolation을 사용

4. Evaluation

4.1 Datasets

- Pre-Train
  - ImageNet
    - 1K classes
    - 1.3M images
  - Superset ImageNet-21k
    - 21K classes
    - 14M images
  - JFT
    - 18K classes
    - 303M images

- Fine-Tune
  - ImageNet
    - Validation set
  - ImageNet with Cleaned-up ReaL Labels
  - CIFAR-10/100
  - Oxford-IIIT Pets
  - Oxford Flowers-102
  - VTAB

4. Evaluation

4.2 Preprocessing

- BiT[3] 논문 방법을 사용
  - Resize the image to a square
  - Training time
    - Crop out a smaller random square
    - Randomly horizontally flip image
  - Test time
    - Only resize the image to a fixed size

3) Kolesnikov, Alexander, et al. "Big transfer (bit): General visual representation learning." *European conference on computer vision*. Springer, Cham, 2020.

4. Evaluation

4.3 Model variants

- BERT에서 사용된 ViT기반 성능
  - Base / Large / Huge
  - ViT-L/16
    - Large 16x16 input patch size
  - Patch가 작아질수록 computationally cost ↑

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.

4. Evaluation

4.3 Model variants

- Baseline CNN
  - ResNet의 일부 수정
  - BatchNorm → Group Norm
  - Standardized Convolutions

- Hybrids
  - ResNet의 stage 4 feature map을 patch size=1로 잘라 입력

4. Evaluation

4.3 Model variants

- Group Norm
  - BN의 단점 보완 : batch size가 작을수록 성능↓
  - Batch와 독립적으로 normalization
  - Channel을 group단위로 normalize
    - Channel : 6
    - Group : 2
    - Group 당 3 channel을 normalize

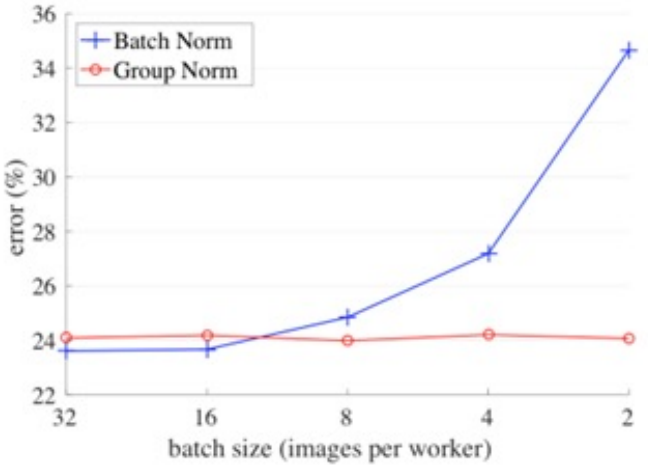
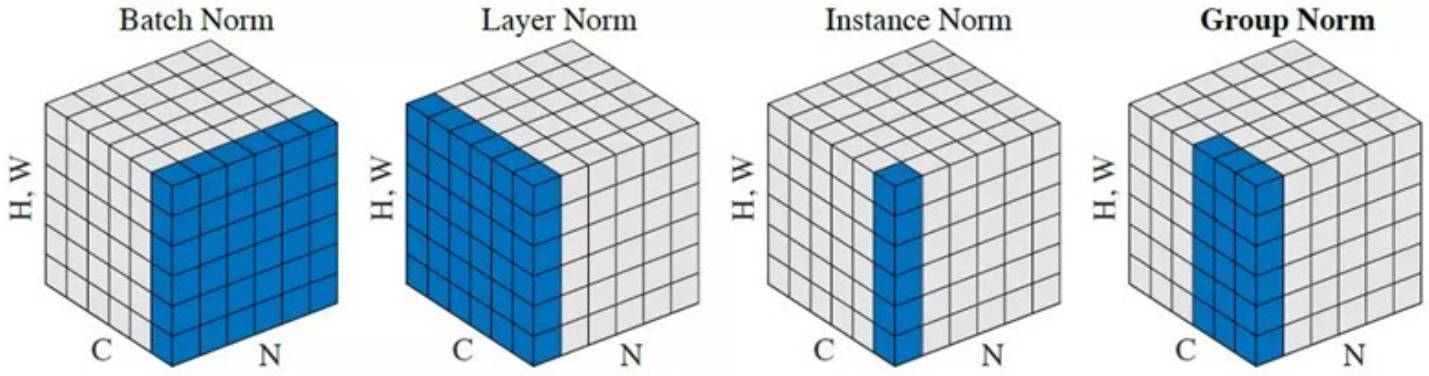


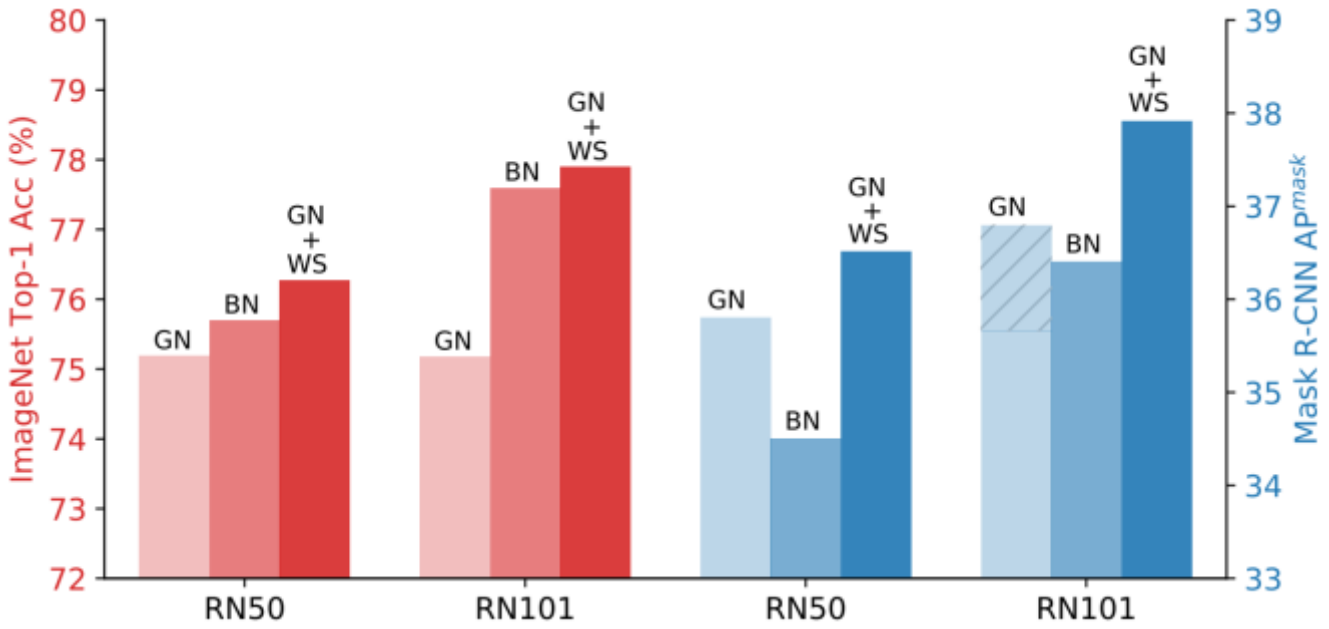
Figure 1. **ImageNet classification error *vs.* batch sizes**. This is a ResNet-50 model trained in the ImageNet training set using 8 workers (GPUs), evaluated in the validation set.



4. Evaluation

4.3 Model variants

- Standardized Convolutions
  - Weight Standardization
  - Group Norm이 Large-batch에서 BN의 성능을 따라가지 못함
  - Convolution filter를 대상으로 Normalize



4. Evaluation

4.4 Training & Fine-Tuning

- Pre-Train Hyperparameter
  - Adam
    - $\beta_1 = 0.9$
    - $\beta_2 = 0.999$
  - Batch size
    - 4096
  - Weight decay
    - 0.1
  - Learning rate Decay
    - Linear
  - Resolution
    - 224

Models	Dataset	Epochs	Base LR	LR decay	Weight decay	Dropout
ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/32	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-H/14	JFT-300M	14	$3 \cdot 10^{-4}$	linear	0.1	0.0
R50x{1,2}	JFT-300M	7	10^{-3}	linear	0.1	0.0
R101x1	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R152x{1,2}	JFT-300M	7	$6 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-B/{16,32}	JFT-300M	7	$8 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/32	JFT-300M	7	$2 \cdot 10^{-4}$	linear	0.1	0.0
R50+ViT-L/16	JFT-300M	7/14	$4 \cdot 10^{-4}$	linear	0.1	0.0
ViT-B/{16,32}	ImageNet-21k	90	10^{-3}	linear	0.03	0.1
ViT-L/{16,32}	ImageNet-21k	30/90	10^{-3}	linear	0.03	0.1
ViT-$*$	ImageNet	300	$3 \cdot 10^{-3}$	cosine	0.3	0.1

Table 3: Hyperparameters for training. All models are trained with a batch size of 4096 and learning rate warmup of 10k steps. For ImageNet we found it beneficial to additionally apply gradient clipping at global norm 1. Training resolution is 224.

4. Evaluation

4.4 Training & Fine-Tuning

- Fine-Tuning Hyperparameter
  - SGD with momentum
  - Batch size
    - 512
  - No weight decay
  - Resolution
    - 384

Dataset	Steps	Base LR
ImageNet	20 000	{0.003, 0.01, 0.03, 0.06}
CIFAR100	10 000	{0.001, 0.003, 0.01, 0.03}
CIFAR10	10 000	{0.001, 0.003, 0.01, 0.03}
Oxford-IIIT Pets	500	{0.001, 0.003, 0.01, 0.03}
Oxford Flowers-102	500	{0.001, 0.003, 0.01, 0.03}
VTAB (19 tasks)	2 500	0.01

Table 4: Hyperparameters for fine-tuning. All models are fine-tuned with cosine learning rate decay, a batch size of 512, no weight decay, and grad clipping at global norm 1. If not mentioned otherwise, fine-tuning resolution is 384.

4. Evaluation

4.5 Comparison to SOTA

- 기존 transfer learning SOTA model
  - BiT
    - Supervised transfer learning with Large ResNet
  - Noisy Student
    - Large EfficientNet trained using semi-supervised learning

4. Evaluation

4.5 Comparison to SOTA

- TPUv3-core-days
  - Pre-Train시 core 수 x 소요일 수

	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** ± 0.04	87.76 ± 0.03	85.30 ± 0.02	87.54 ± 0.02	88.4/88.5*
ImageNet ReaL	**90.72** ± 0.05	90.54 ± 0.03	88.62 ± 0.05	90.54	90.55
CIFAR-10	**99.50** ± 0.06	99.42 ± 0.03	99.15 ± 0.03	99.37 ± 0.06	–
CIFAR-100	**94.55** ± 0.04	93.90 ± 0.05	93.25 ± 0.05	93.51 ± 0.08	–
Oxford-IIIT Pets	**97.56** ± 0.03	97.32 ± 0.11	94.67 ± 0.15	96.62 ± 0.23	–
Oxford Flowers-102	99.68 ± 0.02	**99.74** ± 0.00	99.61 ± 0.02	99.63 ± 0.03	–
VTAB (19 tasks)	**77.63** ± 0.23	76.28 ± 0.46	72.72 ± 0.21	76.29 ± 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Table 2: Comparison with state of the art on popular image classification benchmarks. We report mean and standard deviation of the accuracies, averaged over three fine-tuning runs. Vision Transformer models pre-trained on the JFT-300M dataset outperform ResNet-based baselines on all datasets, while taking substantially less computational resources to pre-train. ViT pre-trained on the smaller public ImageNet-21k dataset performs well too. *Slightly improved 88.5% result reported in Touvron et al. (2020).

- 이전 model보다 Computation cost ↓

4. Evaluation

4.5 Comparison to SOTA

- VTAB

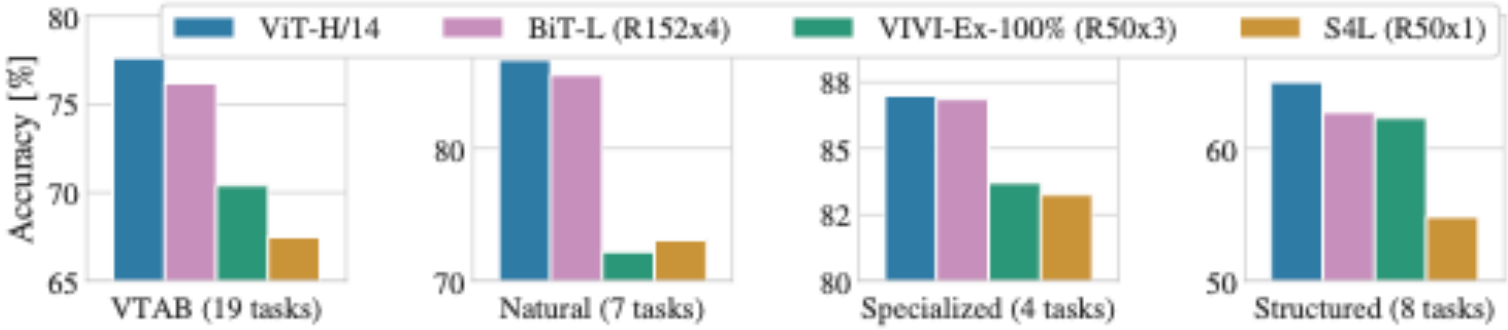


Figure 2: Breakdown of VTAB performance in *Natural*, *Specialized*, and *Structured* task groups.

4. Evaluation

4.6 Pre-Training Data Requirements

- Dataset size의 영향
  - ImageNet
    - 1.3M
  - ImageNet-21k
    - 14M
  - JFT
    - 300M

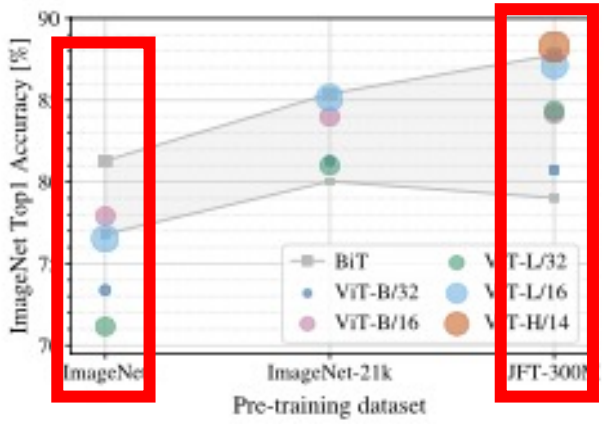


Figure 3: Transfer to ImageNet. While large ViT models perform worse than BiT ResNets (shaded area) when pre-trained on small datasets, they shine when pre-trained on larger datasets. Similarly, larger ViT variants overtake smaller ones as the dataset grows.

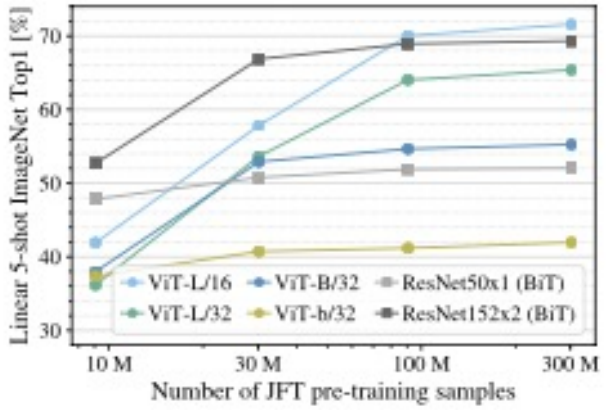


Figure 4: Linear few-shot evaluation on ImageNet versus pre-training size. ResNets perform better with smaller pre-training datasets but plateau sooner than ViT, which performs better with larger pre-training. ViT-b is ViT-B with all hidden dimensions halved.

- Convolution inductive bias는 smaller dataset에 유용하다.

4. Evaluation

4.7 Scaling Study

- 계산량과 성능 비교
  - Hyperparameter 고정
  - 같은 cost
    - ViT > ResNet
  - 작은 model
    - Hybrid > ViT
    - Model이 커질수록 gap이 줄어든다.
  - ViT는 model이 커져도, 성능포화 없이 지속적으로 성능증가

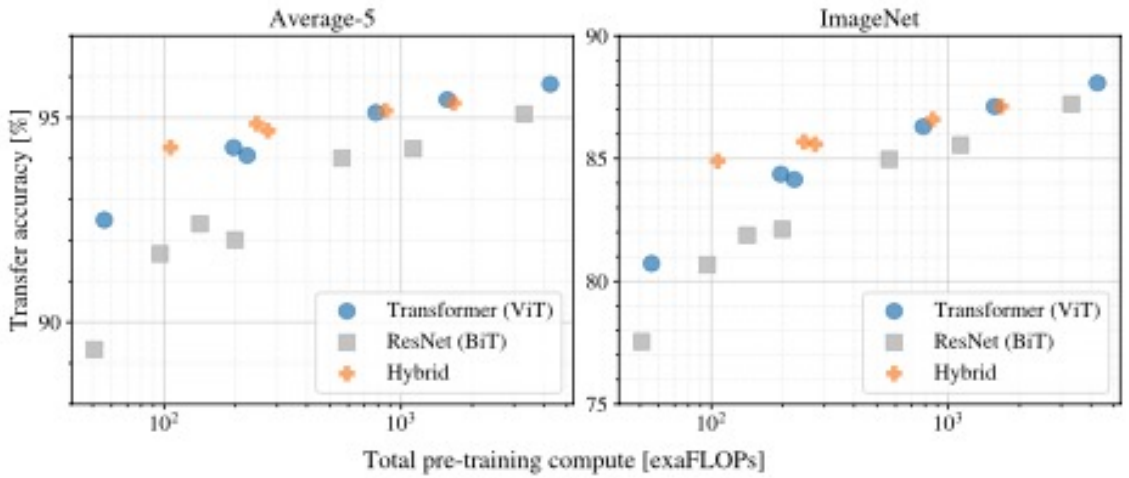
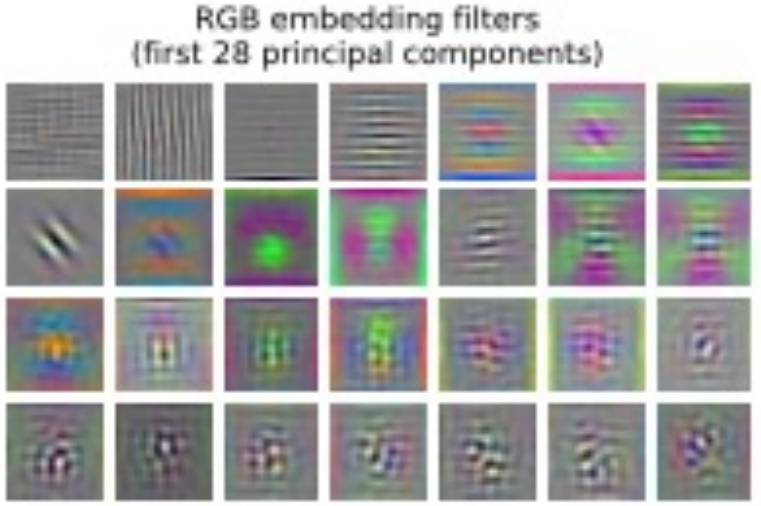


Figure 5: Performance versus pre-training compute for different architectures: Vision Transformers, ResNets, and hybrids. Vision Transformers generally outperform ResNets with the same computational budget. Hybrids improve upon pure Transformers for smaller model sizes, but the gap vanishes for larger models.

4. Evaluation

4.8 Inspecting Vision Transformer

- ViT가 image를 어떻게 처리하는지 내부를 분석
  - Linear projection layer의 D개의 필터 중 28개 선택

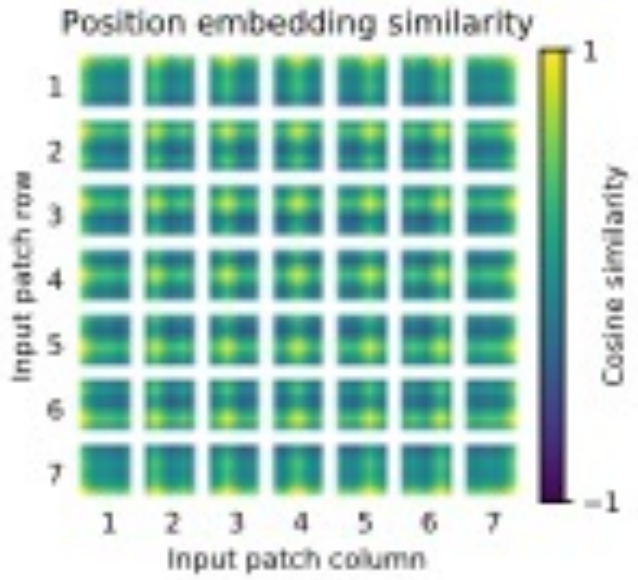


- Edge, color 등 low-level 특징 학습을 확인

4. Evaluation

4.8 Inspecting Vision Transformer

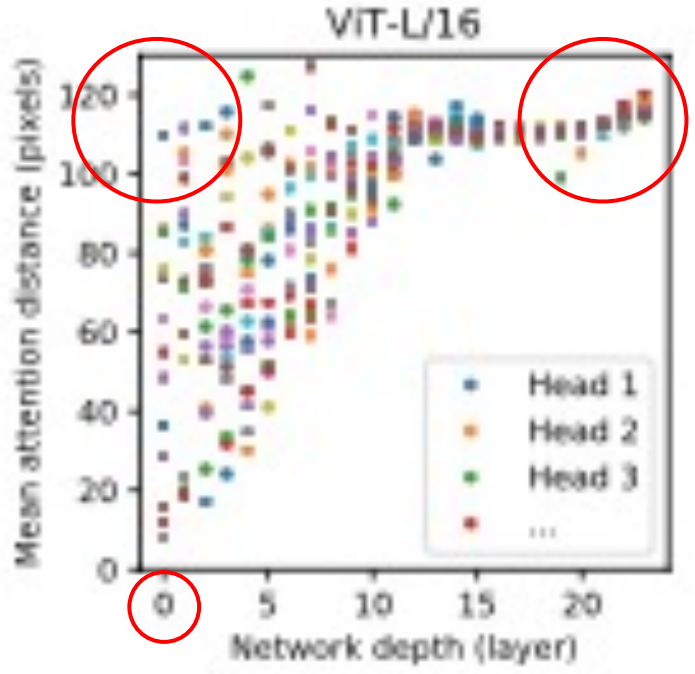
- After projection
  - Position embedding 유사도 분석
  - 1D Position embedding에도 2D 이미지 위상을 잘 파악하고 있음을 확인



4. Evaluation

4.8 Inspecting Vision Transformer

- Attention Distance
  - 한 Pixel이 어느 정도까지 떨어진 pixel까지 attention하는지 확인
  - 첫 layer에서도 전역적 특성을 학습하고 local 특성도 학습
  - Layer가 깊어질수록 대부분 전역적 특성을 학습



4. Evaluation

4.9 Self-Supervision

- BERT
  - Self-supervision pretext task로 좋은 성능
- ViT
  - Patch를 masking하고 예측하는 방식 사용
  - imageNet
    - Masking rate : 50%
    - ViT-B/16 : 79.9%
    - fine-tuning한 결과보다 4%↓

5. Conclusion & Future Work

5.1 Conclusion

- Image recognition에 tranformer를 직접적으로 활용한 첫 사례
  - Image를 patch sequence로 해석
- Pre-train에서 상대적으로 낮은 비용으로 SOTA를 달성

5.2 Future Work

- ViT를 다른 task에 적용
  - Detection, segmentation
- Self-supervised pre-train method

감사합니다.

육현준