[1]A. Dosovitskiy *et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.*
[2]M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," *Advances in Neural Information Processing Systems, vol. 34, 2021.*

An Image is worth 16 x 16 words: Transfomers for recognition at scale[1]
Do vision transformers see like convolutional neural networks?[2]

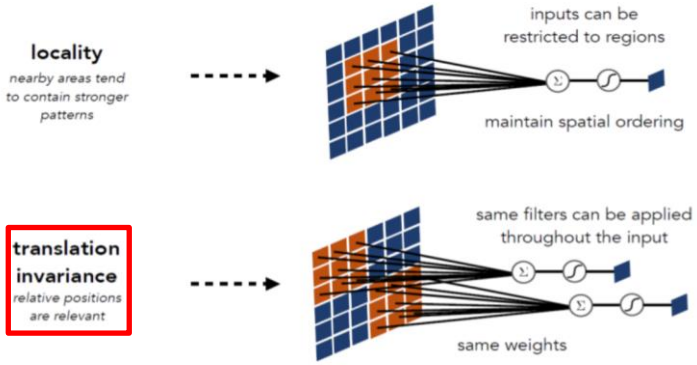
Seong Su Kim

Man Machine Interface Lab.

Contents

Background

- Inductive bias
  - Train data는 항상 제한적
  - 학습 모델이 지금까지 만나보지 못했던 상황에서 정확한 예측을 하기 위해 사용하는 **추가적인 가정**
  - 모델이 학습과정에서 본 적이 없는 분포의 데이터를 입력 받았을 때, 해당 데이터에 대한 판단을 내리기 위해 가지고 있는 설계 구조 상의 편향(bias)
  - SVM
    - Maximum margin : Decision boundary 와의 margin을 가장 크게 하는 가설
  - CNN
    - Locality : 특정 pixel과 가장 관계가 깊은 pixel은 근처의 영역에 위치할것이라는 가정
    - Translation invariance: 어떠한 사물이 들어 있는 이미지에 대해 사물의 위치가 바뀌어도 해당 사물을 인식할 수 있음
  - RNN
    - Sequentiality : 시간순으로 가장 인접한 단어가 현재 단어와의 영향력이 클 것임을 가정

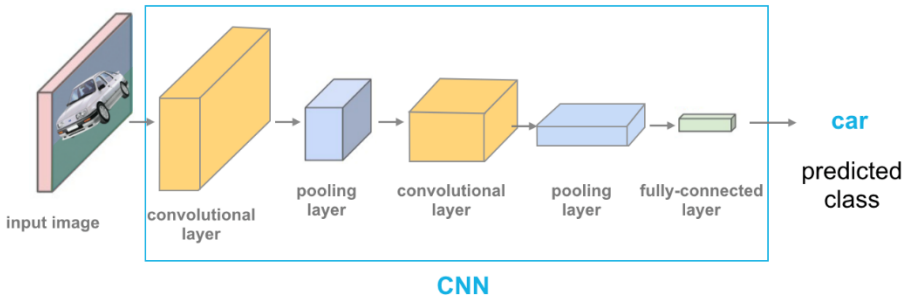


Component	Entities	Relations	Rel. inductive bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

https://towardsdatascience.com/translational-invariance-vs-translational-equivariance-f9fbc8fca63a

Background

- Inductive bias of CNN
  - Translational invariance VS Translational Equivalance
    - Interchageable in common speech
    - Invariance : "in–" (No variance at all) 불변성
    - Equivalance : "equi-" (Varying in a similar or equivalent proportion) 등가성



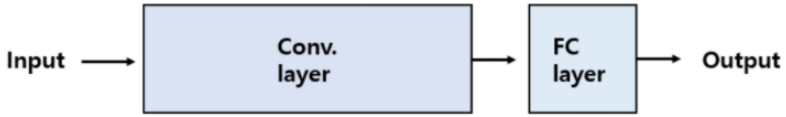
1. Convolutional layer (weight sharing)
   - *" The position of the object in the image should not be fixed in order "*
   - As the same weights are shared across the images, hence if an object occurs in any image it will be detected **irrespective of its position** in the image
   - Equivalent (The feature of image can be change, but the property is same)
2. Pooling operation
   - "Replacing the output of the convnet at a certain location with a summary statistic of the nearby outputs"
   - Max pooling : As we replace the output with the max in case of max-pooling (value m), hence even if we change the input slightly, it **won't affect the values** of most of the pooled outputs
   - Invariance (No variance in "value m")
- 명확히 다른데 왜 혼용해도 문제가 없었고, 어색하지 않았을까?
  - 목적어가 없기 때문 (Predicted class 가 목적어일 경우를 생각해보면..)
  - CNN 내의 별개의 구조적 특성으로부터 발생한 특성이기 때문에 구분해서 이해

Background

- Transfer learning & Fine tuning
  - Transfer learning
    - 특정 task에서 학습된 network를 다른 task에서 사용되는 network의 학습에 이용하는 방법
    - Transfer learning에 사용되는 "학습된 network" = Pre-trained model
    - 학습 데이터의 수가 적을 때 효과적
  - Fine – tuning
    - Pre-trained model을 기반으로 architecture를 새로운 목적에 맞게 변형하고 pre-trained 된 model의 weight를 미세하게 조정 (fine- tune)하여 학습시키는 방법
    - Pre-trained model의 classifier는 삭제하고, 목적에 맞는 새로운 classifier를 추가

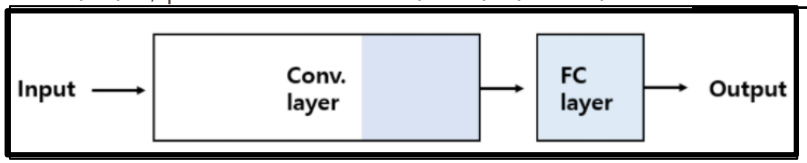
: Train (LR = original LR / 10)
: Frozen (LR = 0)

1. 현재 task의 dataset size가 충분하지만, pre-trained dataset과 많이 다를 경우



2. 현재 task의 dataset size가 충분하고, pre-trained dataset과 유사할 경우

3. 현재 task의 dataset size가 작고, pre-trained dataset과 많이 다를 경우



Conv layer의 overfitting이 발생할 우려가 있으므로, 전체를 재학습 하면 안됨

Sol : Data augmentation

4. 현재 task의 dataset size가 작고, pre-trained dataset과 유사할 경우

[3]A. Kolesnikov *et al., "Big transfer (bit): General visual representation learning," in European conference on computer vision, 2020: Springer, pp. 491-507.*

김성수

Background

- Transfer learning & Fine tuning
  - BiT[3]
    - Idea : 잘 만든 pre-trained 모델 하나가 여러 모델 안부럽다!
    - "어떻게 해야 Transfer Learning을 잘 할 수 있는가?"를 model의 size, dataset의 크기, hyper parameter setting 관점에서 분석한 논문
    - 다양한 방법으로 pretrain된 ResNet을 down stream task에 적용하며 transfer learning에 대한 insight를 제공
      - 2019 ImageNet classification SOTA (BiT-L)
    - [1]에서는 BiT를 CNN SOTA model로서 Transformer와 CNN 두 architecture간 비교를 위해 참조 (둘다 Google Brain에서 발표)

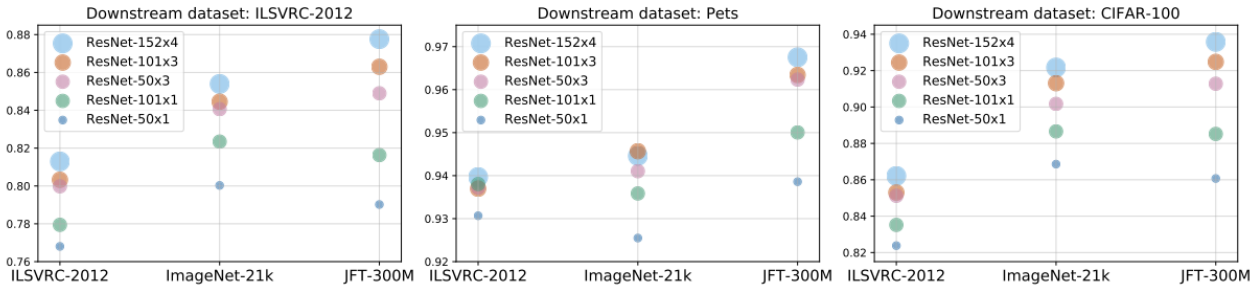


Fig. 5: Effect of upstream data (shown on the x-axis) and model size on downstream performance. Note that exclusively using more data or larger models may hurt performance; instead, both need to be increased in tandem.

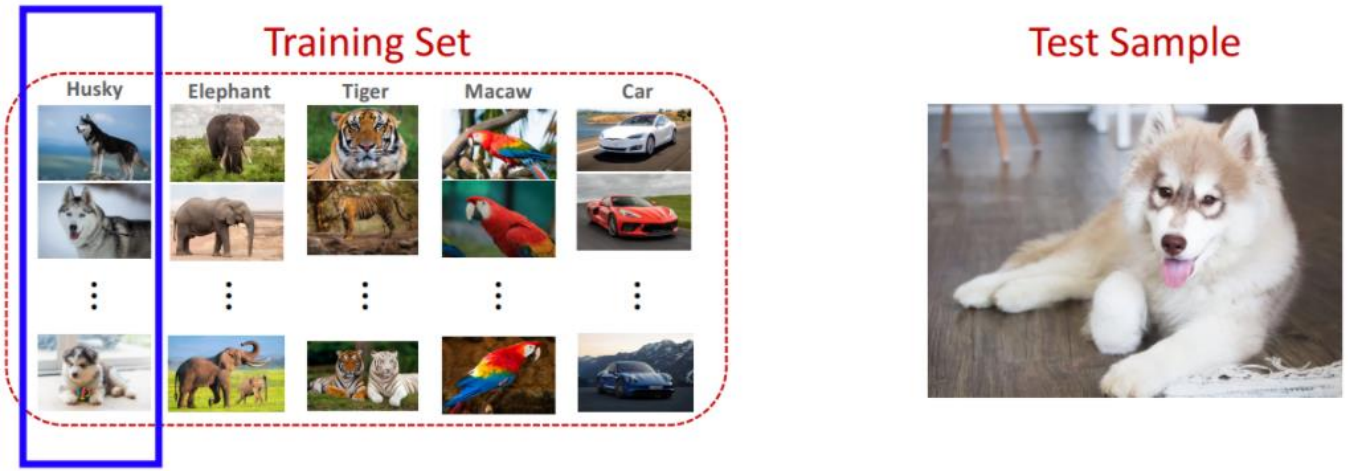
- Brief conclusion about scaling models and datasets
  - Pre-train시, dataset 크기를 증가시키는 것은 down stream task의 성능 향상에 효과적이다.
  - 작은 model(ex: ResNet-50x1)을 pre-train할 때 , 너무 큰 dataset을 쓰는건 오히려 역효과를 날 수 있다.
  - 큰 model일수록 큰 dataset을 이용하여 pre-train 해야 down stream task에서 뛰어난 성능을 보인다.

Background

https://zzaebok.github.io/machine_learning/FSL/

- Few-shot learning
  - Meta learning의 한 종류로, class의 특징을 배우는 supervised learning과 달리 "구분 하는법"을 배우는 학습 방법

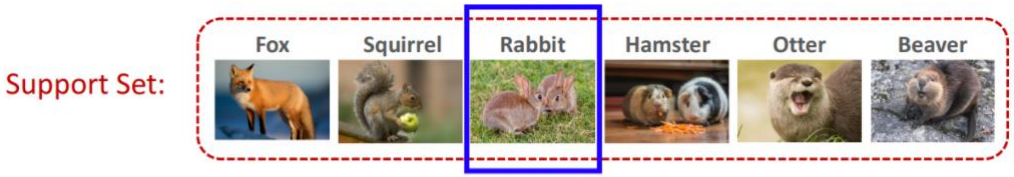
- Supervised learning VS Few-shot learning
  - Supervised Learning : Training set 안에 test image의 class가 존재

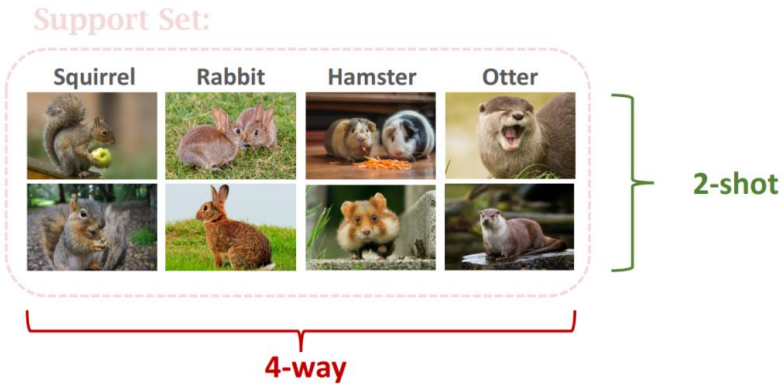


Background

https://zzaebok.github.io/machine_learning/FSL/

- Supervised learning VS Few-shot learning
  - Few-shot learning
    - Support Set에 Query sample class가 존재, Support set과 train set의 class는 겹치지 않음
    - K – way N-shot : K개의 class를 가진 support set, 각 class가 N개의 sample을 가짐
      - K가 커질 수록, query image가 k개의 class 중 어떤 것과 같은 것인지 물어보는 문제가 되므로 모델의 정확도 ↓
      - N이 커질 수록, query image와 비교해볼 sample이 많으므로 어떤 class 속하는지 알기 쉽기 때문에 모델의 정확도 ↑



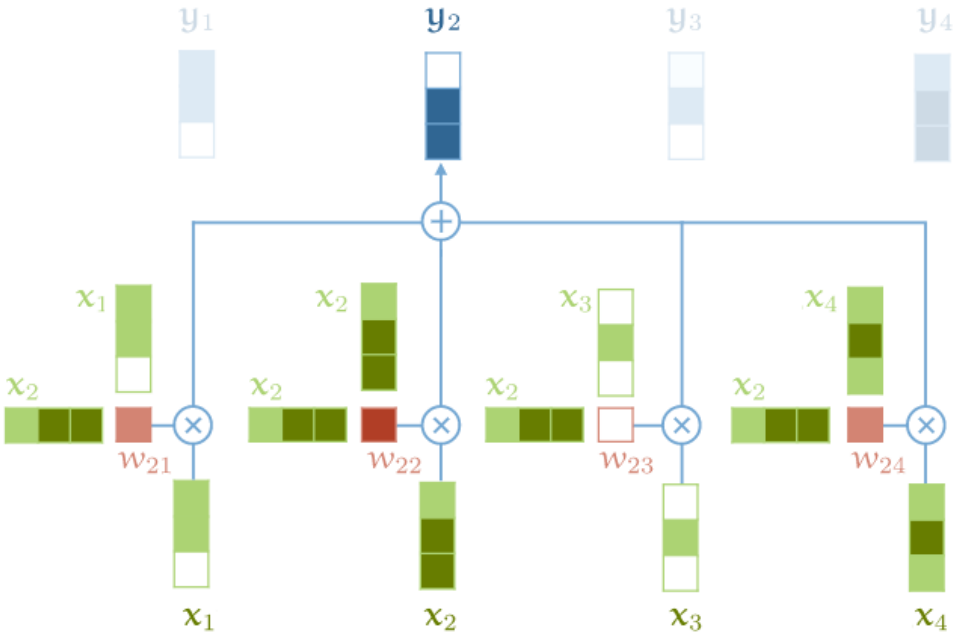


- Few shot learning의 성능의 높다는 것은 각 class내 sample들의 유사도를 잘 학습한다는 의미

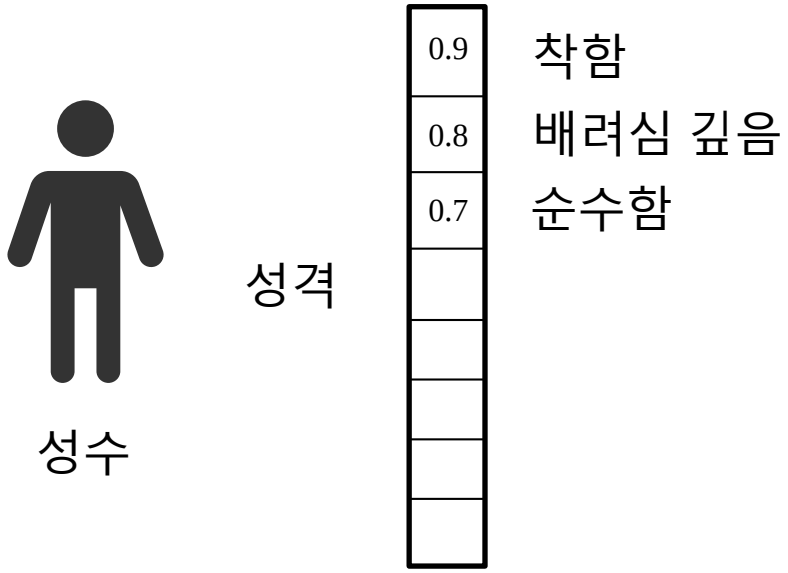
Intuition of Self-Attention

- Basic operation
  - Sequence-to-squence opeation : Sequence of vector goes in, and a sequence of vector comes out.
  - Input vector : $x_1, x_2, x_3, \ldots, x_t$
  - Output vector : $y_1, y_2, y_3 \ldots, y_t$
  - Not parameterized, Just dot product & Weighted Summation

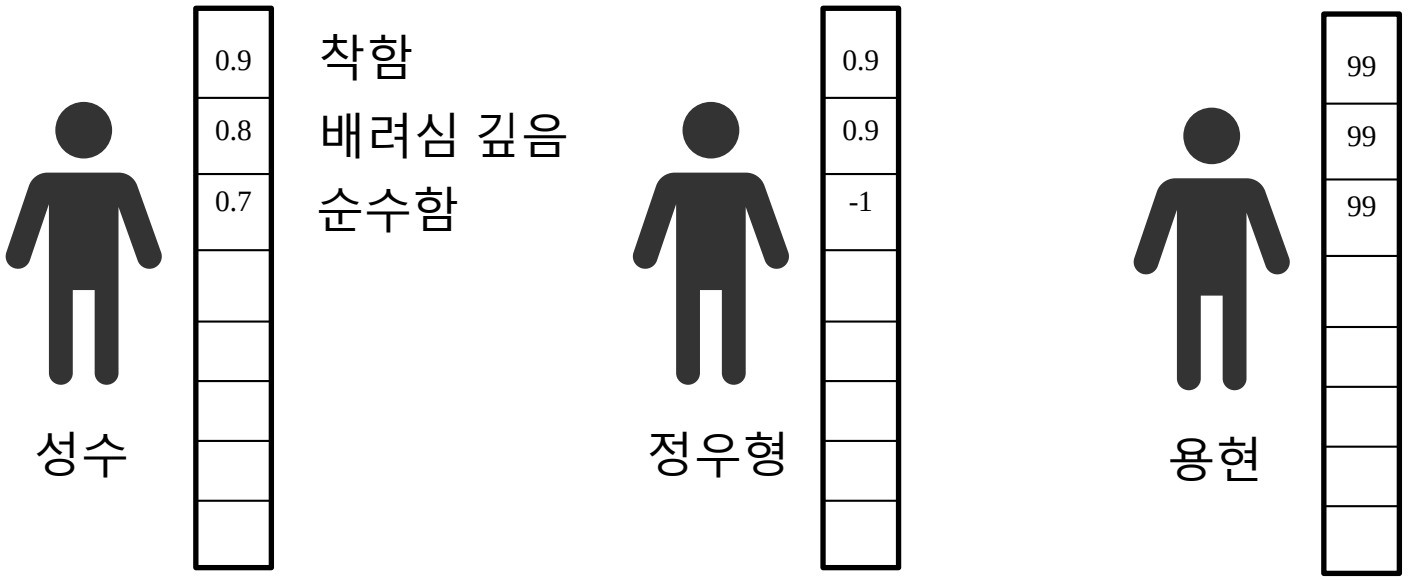
$$\mathbf{y}_i = \sum_j w_{ij} \mathbf{x}_j \,. \qquad w'_{ij} = \mathbf{x}_i^T \mathbf{x}_j$$



Intuition of Self-Attention



Intuition of Self-Attention

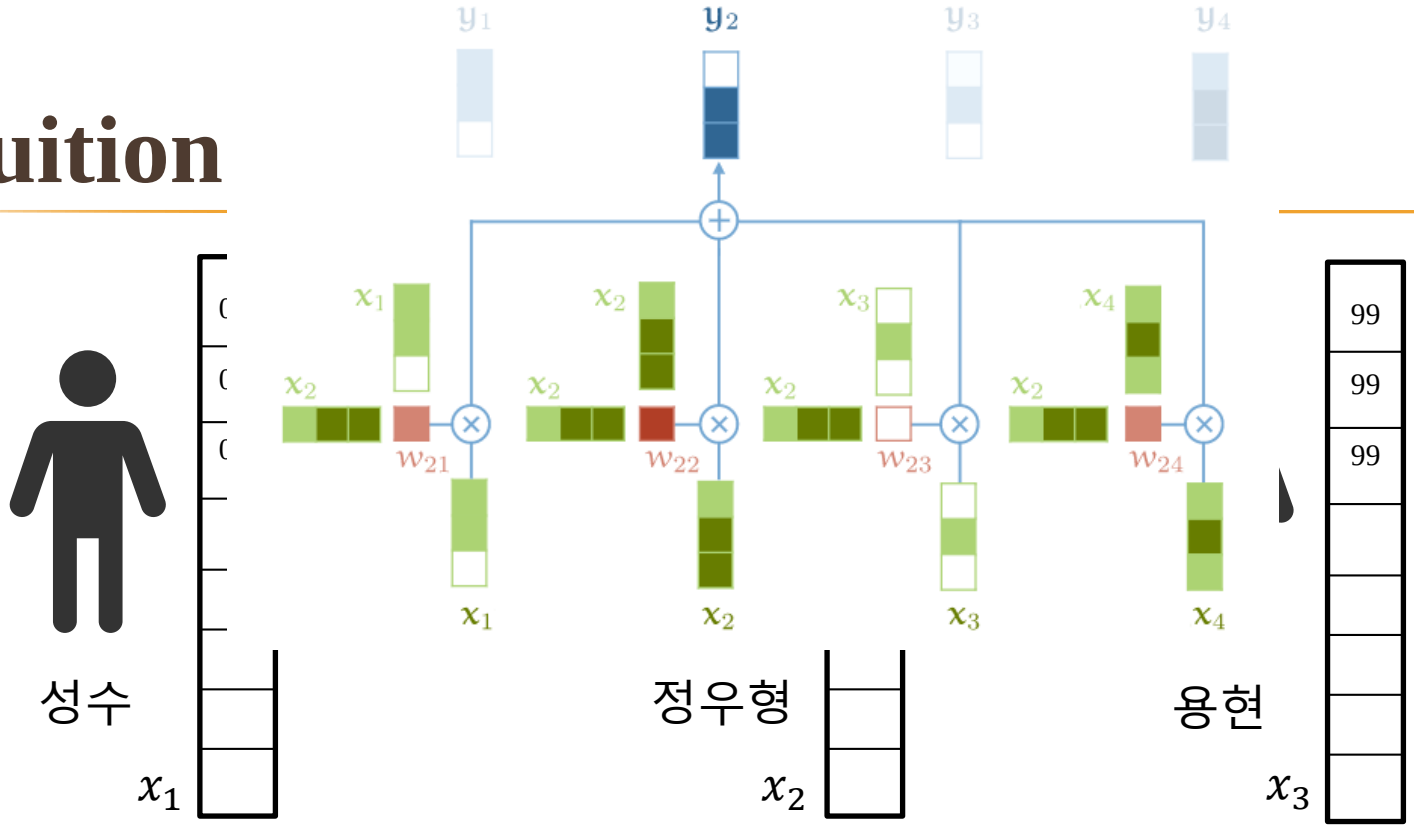


Self – Attention

“친구들을 보면 그 사람이 어떤 사람인지 알 수 있다 .”

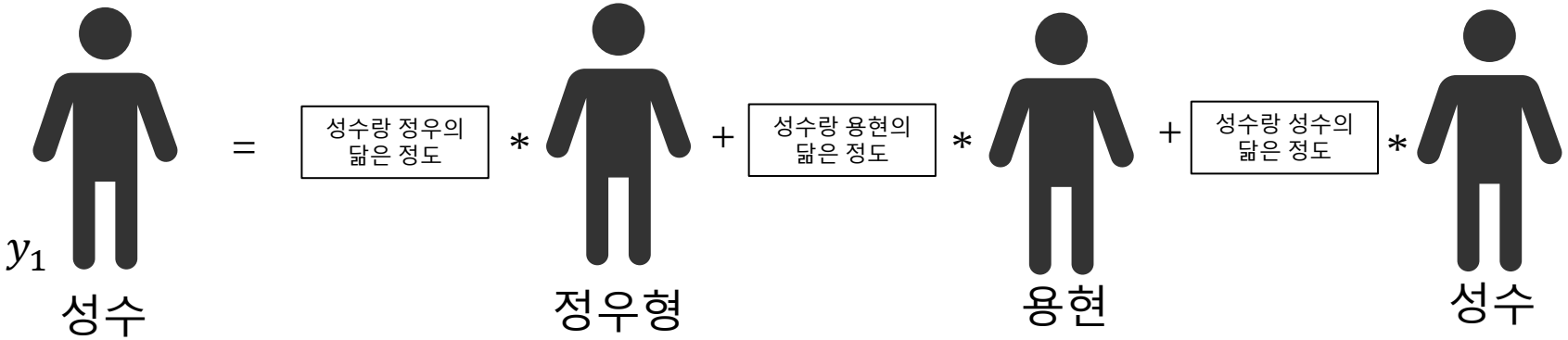
→ 끼리끼리 논다고 했으니까, 성수의 주위사람들의 성격을 바탕으로 성수가 어떤 사람인지 정의 해보자!

Intuition



성수랑 정우형의 성격이 비슷하다면 , feature vector간 dot product가 클 것이고 둘은 서로 닮았을 것이다.

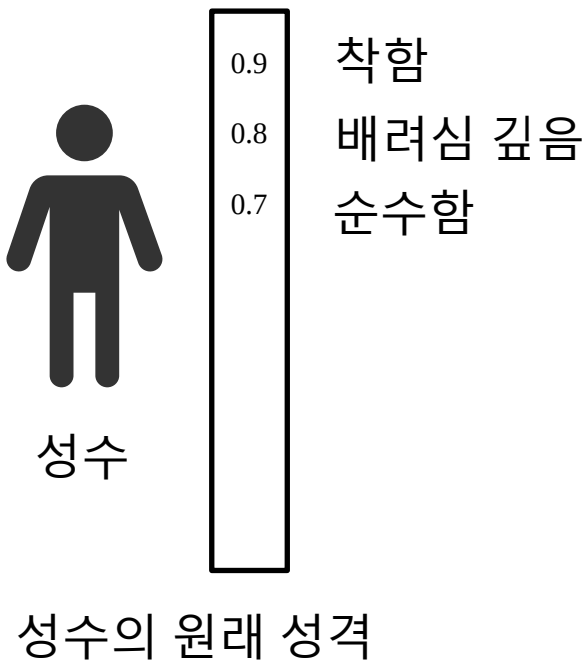
이 값을 가중치로 사용해서 성수의 성격을 나타낸다면?



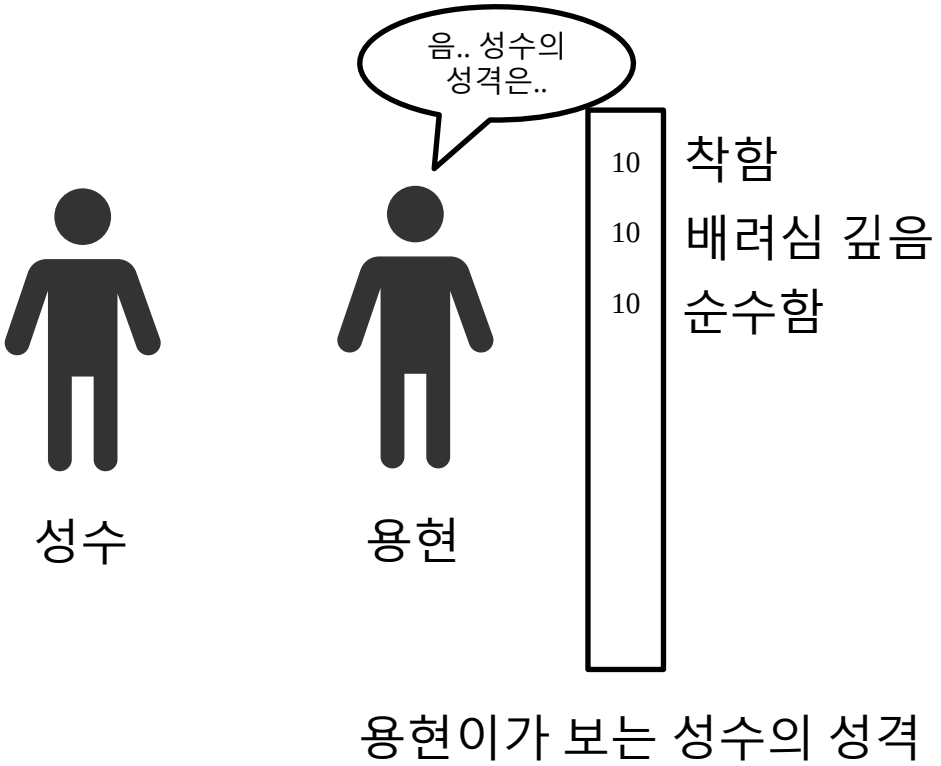
Intuition of Self-Attention

그러나, 관계의 종류와 환경에 다라

성수가 생각하는 자신의 원래 성격과 남들에게 보여지는 성격은 다를 수 있다.

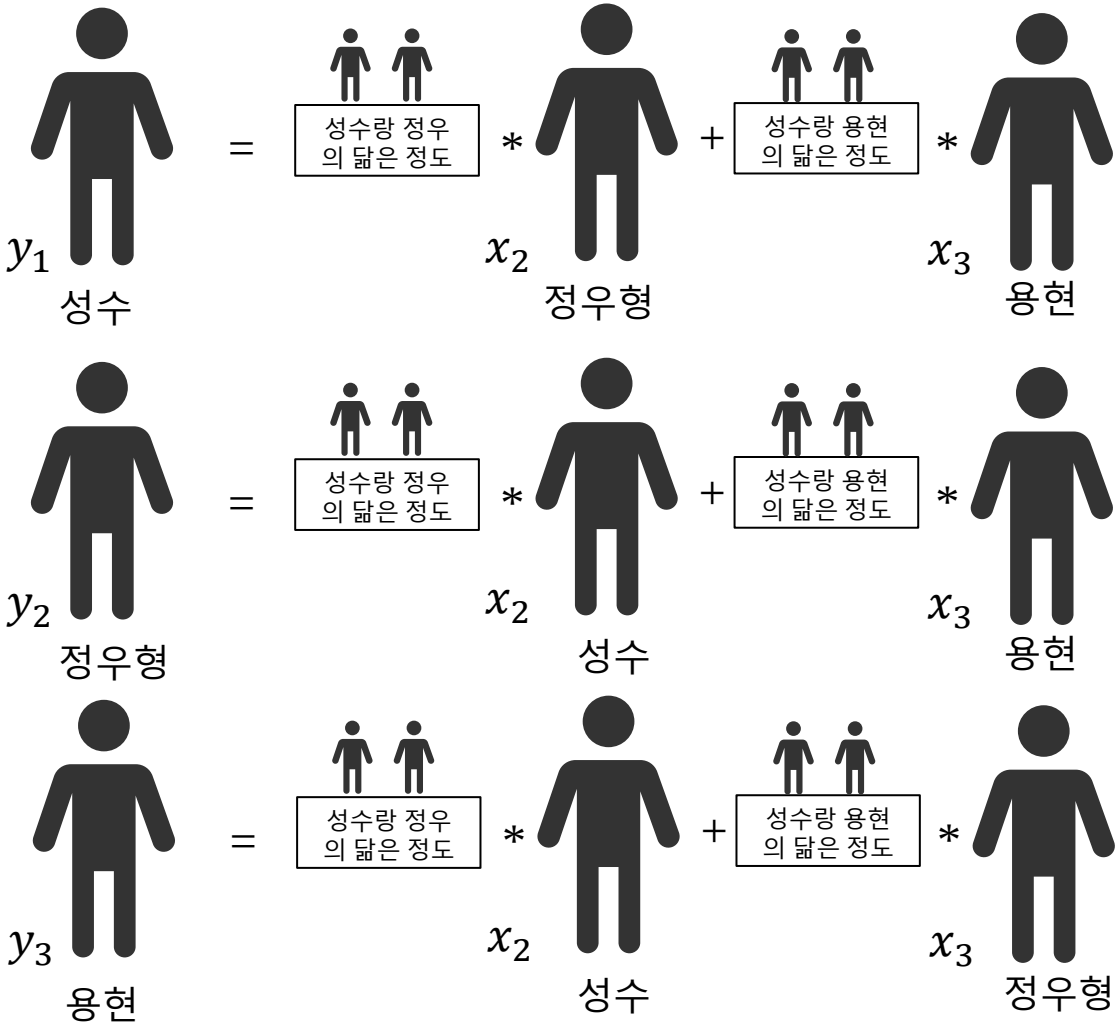


성수의 원래 성격



용현이가 보는 성수의 성격

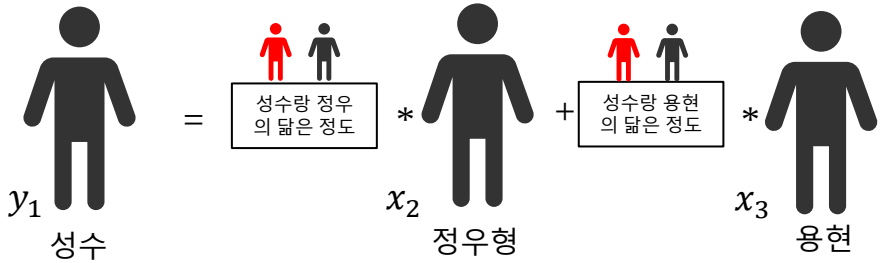
Intuition of Self-Attention



모두의 성격을 정의하는 과정에서 "성수의 성격"은 몇 가지 다른 용도로 쓰였을까?

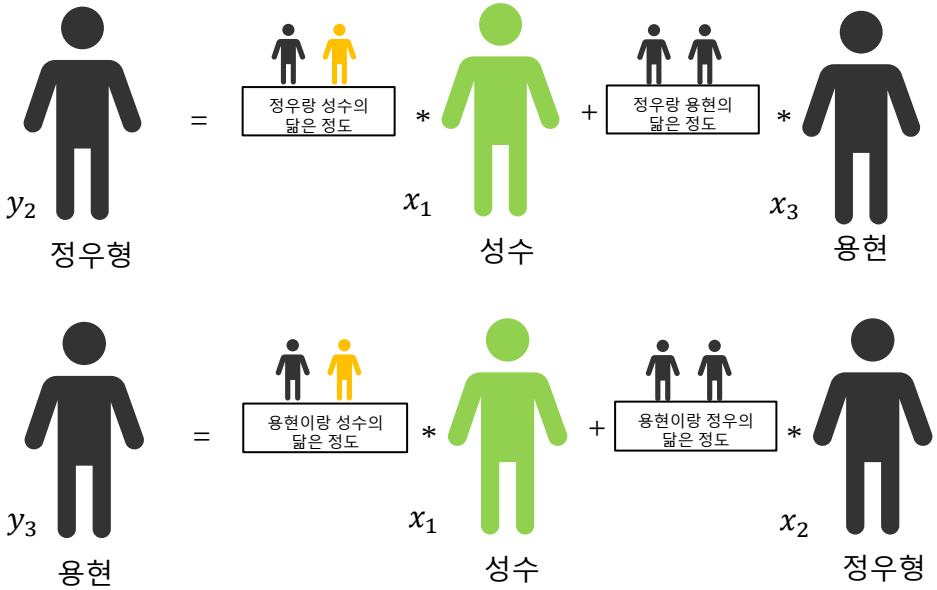
Intuition of Self-Attention

- Every input vector x_i is used in **three different ways** in the self attention operation
  1. It is compared to every other vector to establish the weights for its own output y_i



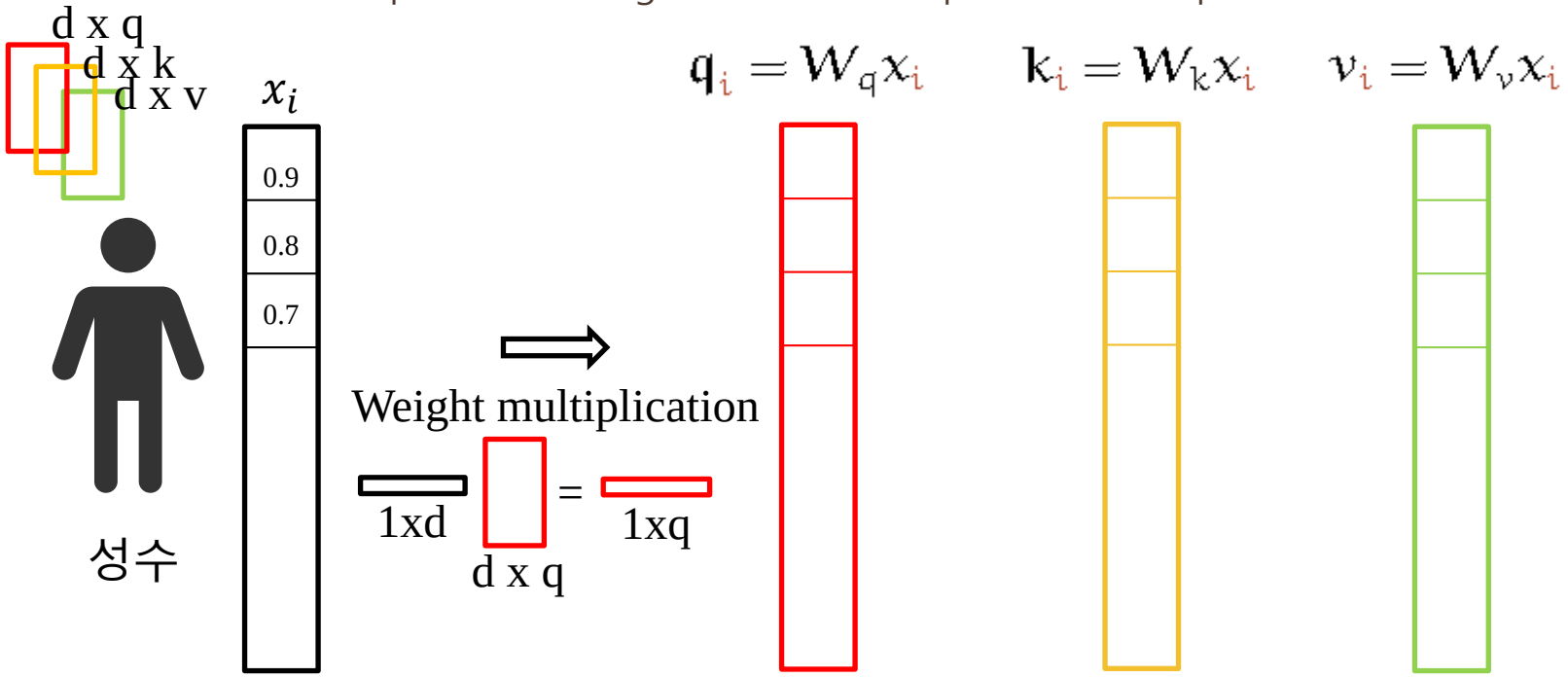
Query

  2. It is compared to every other vector to establish the weights for the output of the j-th vector y_j **Key**
  3. It is used as part of the weighted sum to computed each output vector **Value**



Intuition of Self-Attention

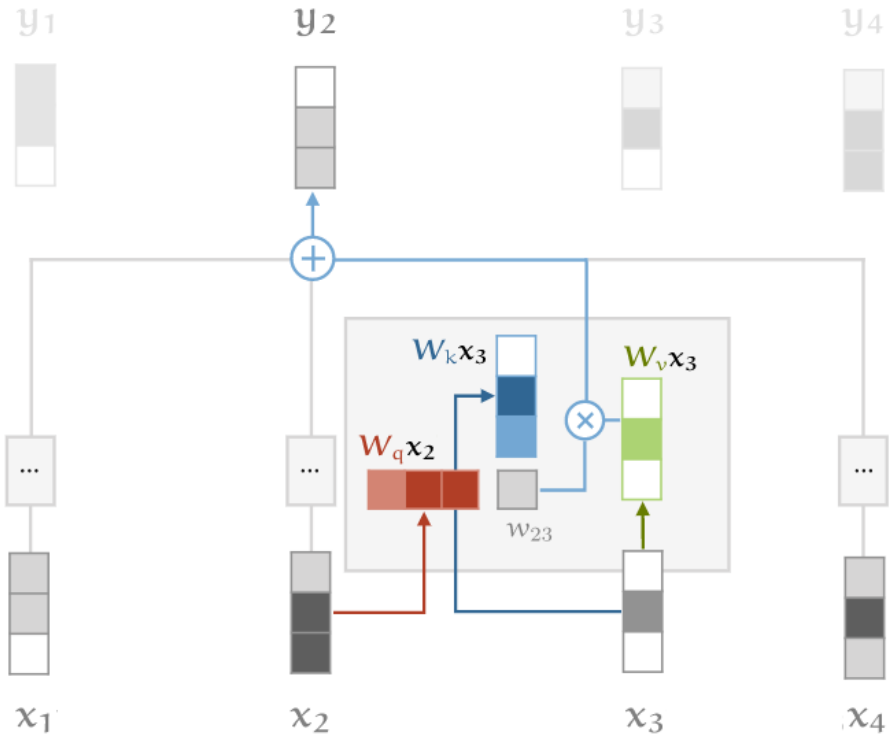
- Every input vector x_i is used in **three different ways** in the self attention operation
  1. It is compared to every other vector to establish the weights for its own output y_i : Query
  2. It is compared to every other vector to establish the weights for the output of the j-th vector y_j : Key
  3. It is used as part of the weighted sum to computed each output vector : Value



- 즉, input vector 각각에 대해 query vector, key vector, value vector가 존재

Intuition of Self-Attention

- Query , Key, Value matrix를 통해 attention mechanism내에서 input vector가 3개의 role을 잘 수행할 수 있는 Query, Key, Value가 되도록 학습



$$\mathbf{q}_i = W_q x_i \qquad \mathbf{k}_i = W_k x_i \qquad v_i = W_v x_i$$

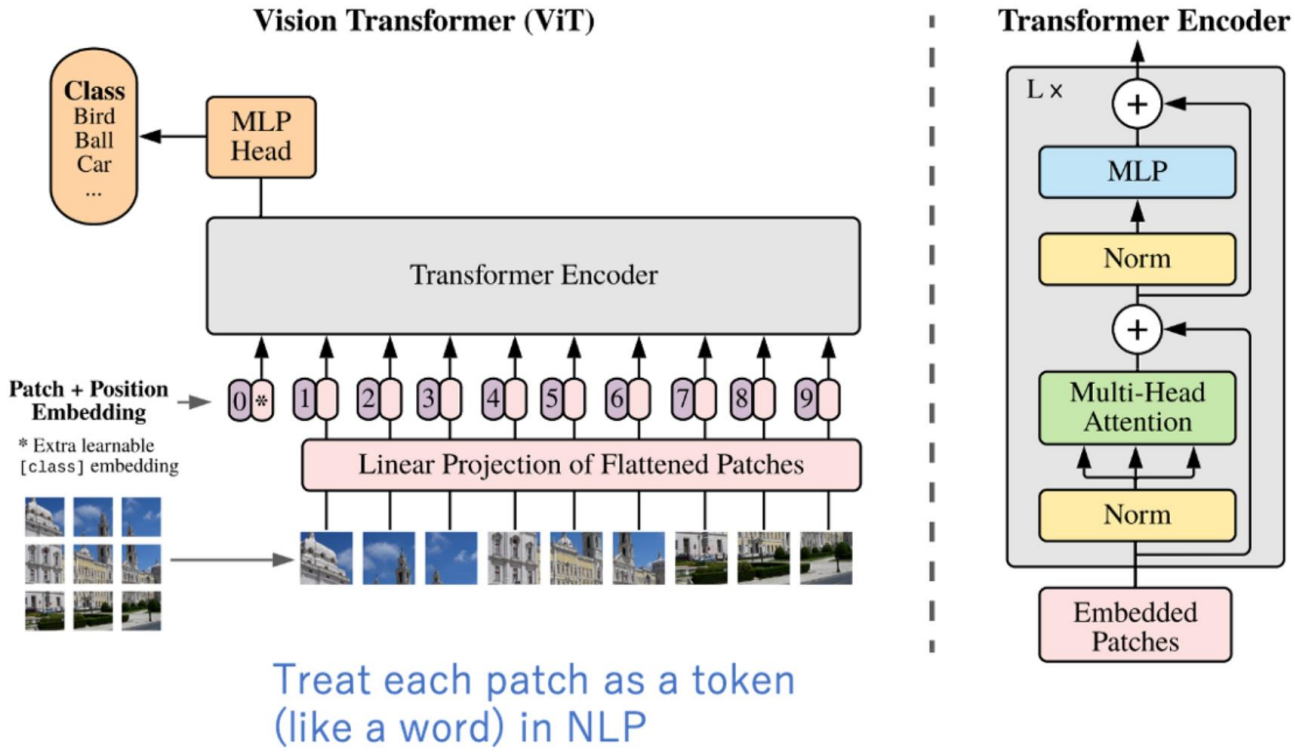
$$w'_{ij} = \mathbf{q}_i{}^{\top} \mathbf{k}_j$$

$$w_{ij} = \mathrm{softmax}(w'_{ij})$$

$$\mathbf{y}_i = \sum_j w_{ij} v_j \,.$$

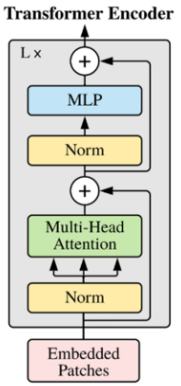
Vision Transformer[1]

- Architecture
  1. Using Image patches : pixel value를 flatten하여 linear projection layer에 통과 (16 x 16 x 3 = 768)
     - Weight sharing
  2. CLS token: Extra learnable CLS token을 따로 두어, 이후 classification에 사용
  3. Position Embedding : Learnable 1D position embedding

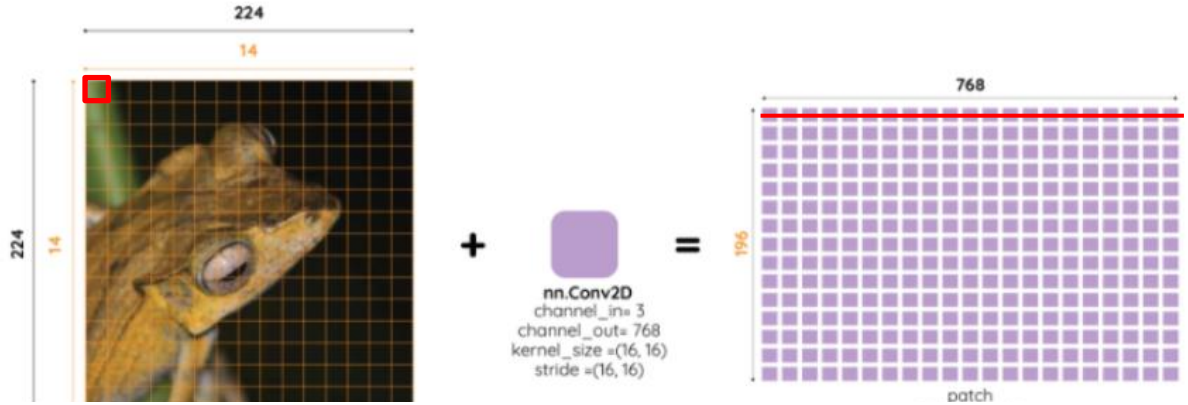


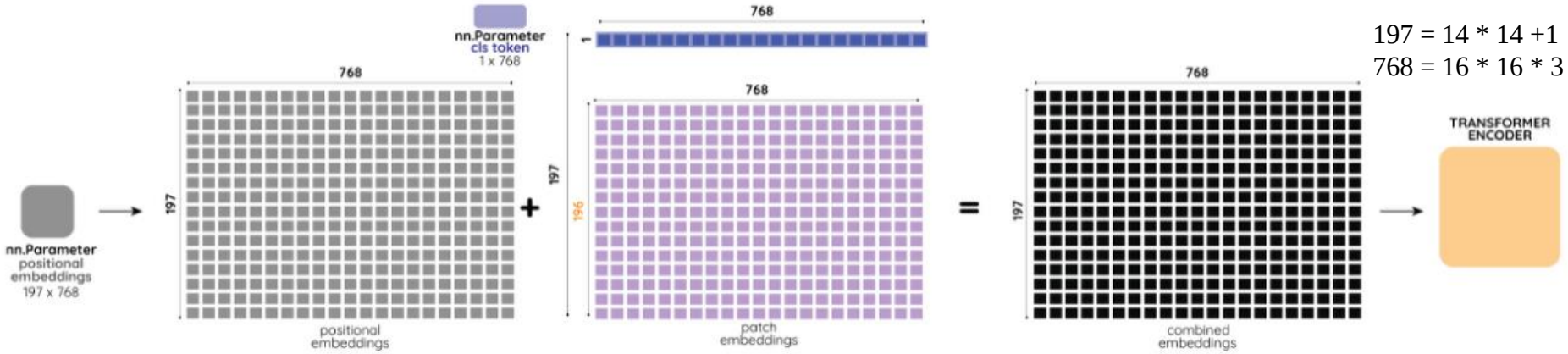
김성수

Vision Transformer[1]



- Details
  1. Linear Projection of flatten patches
     - Patch size : 16 x 16
     - Weight sharing by Conv2D : 16 x 16 kernel, 16 x 16 stride
     - CLS token : initialize with zero vector

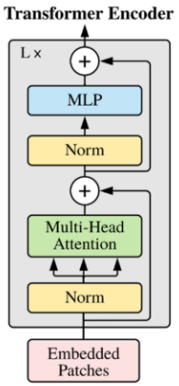




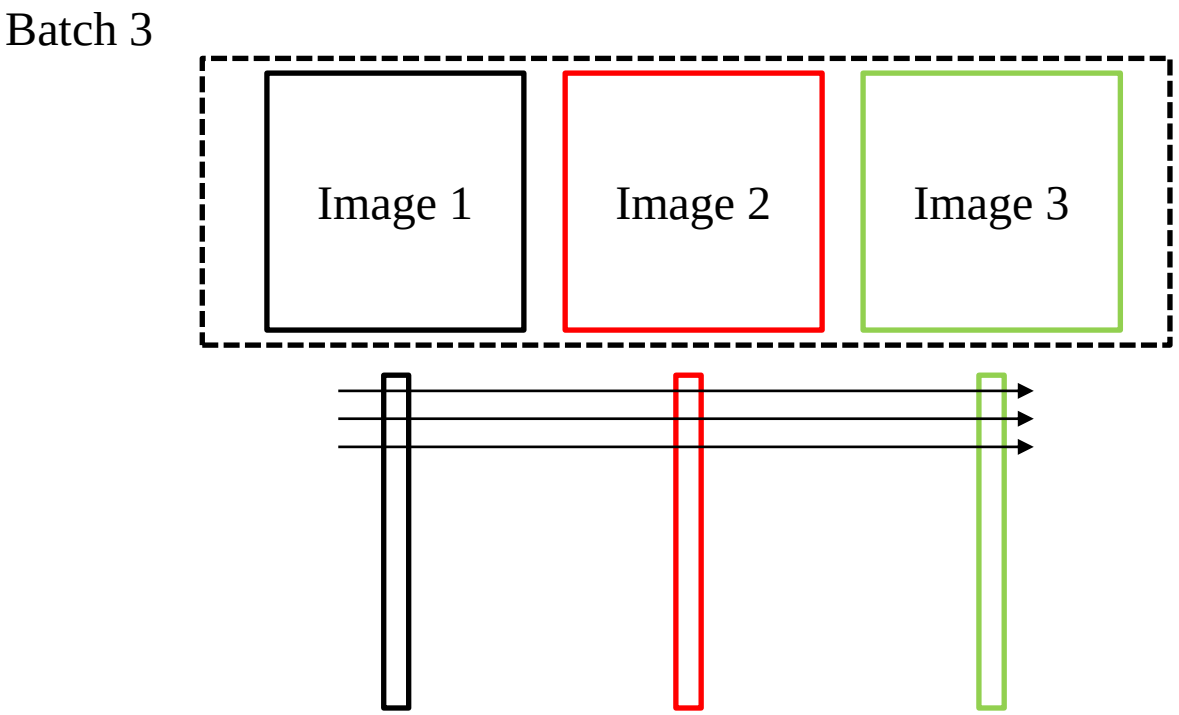
197 = 14 * 14 +1
768 = 16 * 16 * 3

김성수

Vision Transformer[1]

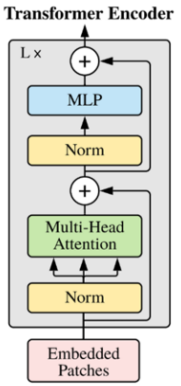


- Details
  2. Layer Normalization
     - Patch라는 개념과 attention mechanism을 고려한 normalization 기법
     - Batch Norm VS Layer Norm

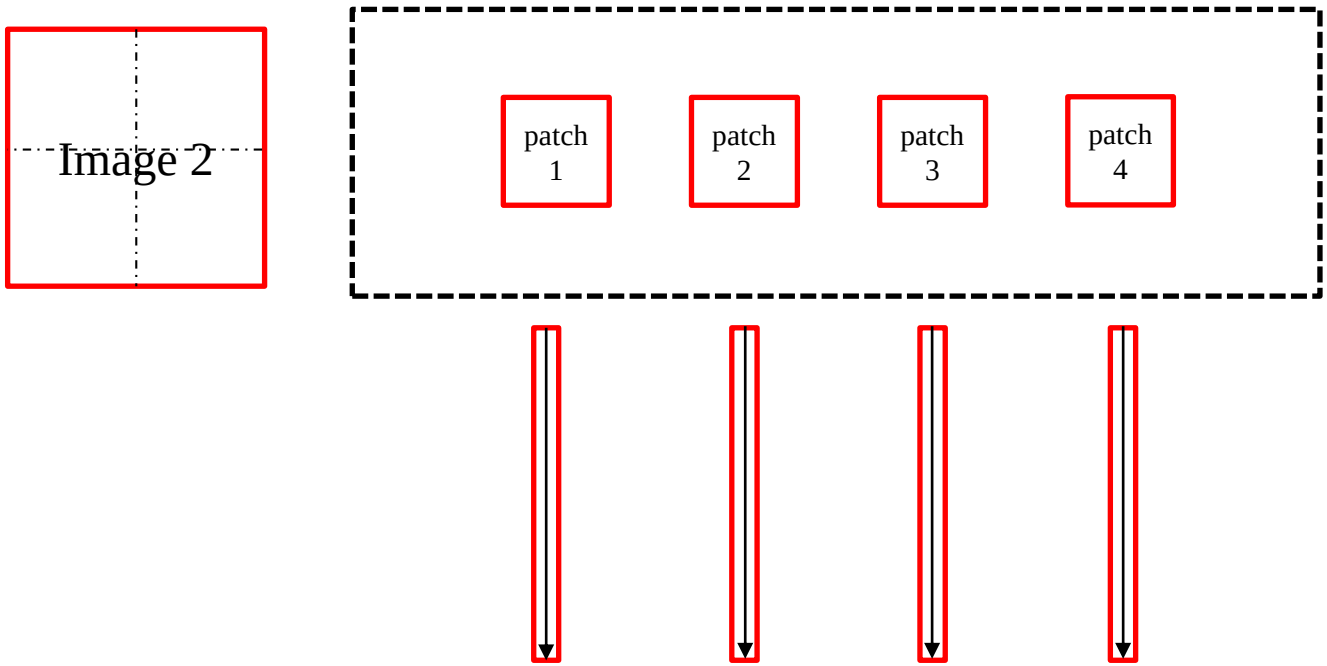


김성수

Vision Transformer[1]



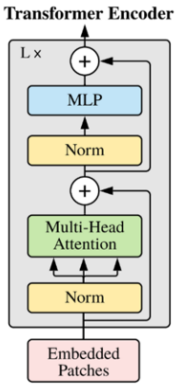
- Details
  2. Layer Normalization
     - Patch라는 개념과 attention mechanism을 고려한 normalization 기법
     - Batch Norm VS Layer Norm



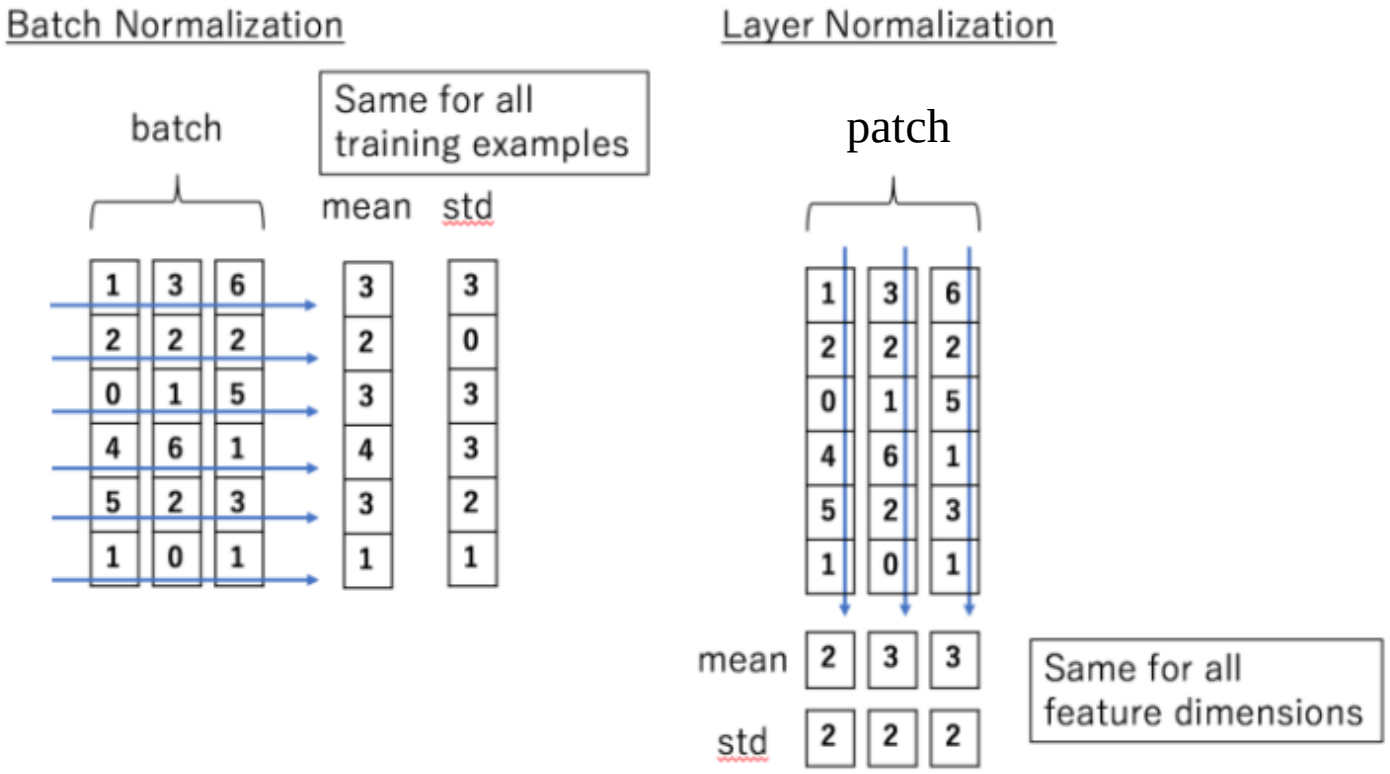
Patch 끼리가 아닌, 각 patch내의 feature를 normalize

김성수

Vision Transformer[1]

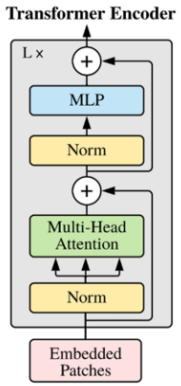


- Details
  2. Layer Normalization
     - Patch라는 개념과 attention mechanism을 고려한 normalization 기법
     - Batch Norm VS Layer Norm

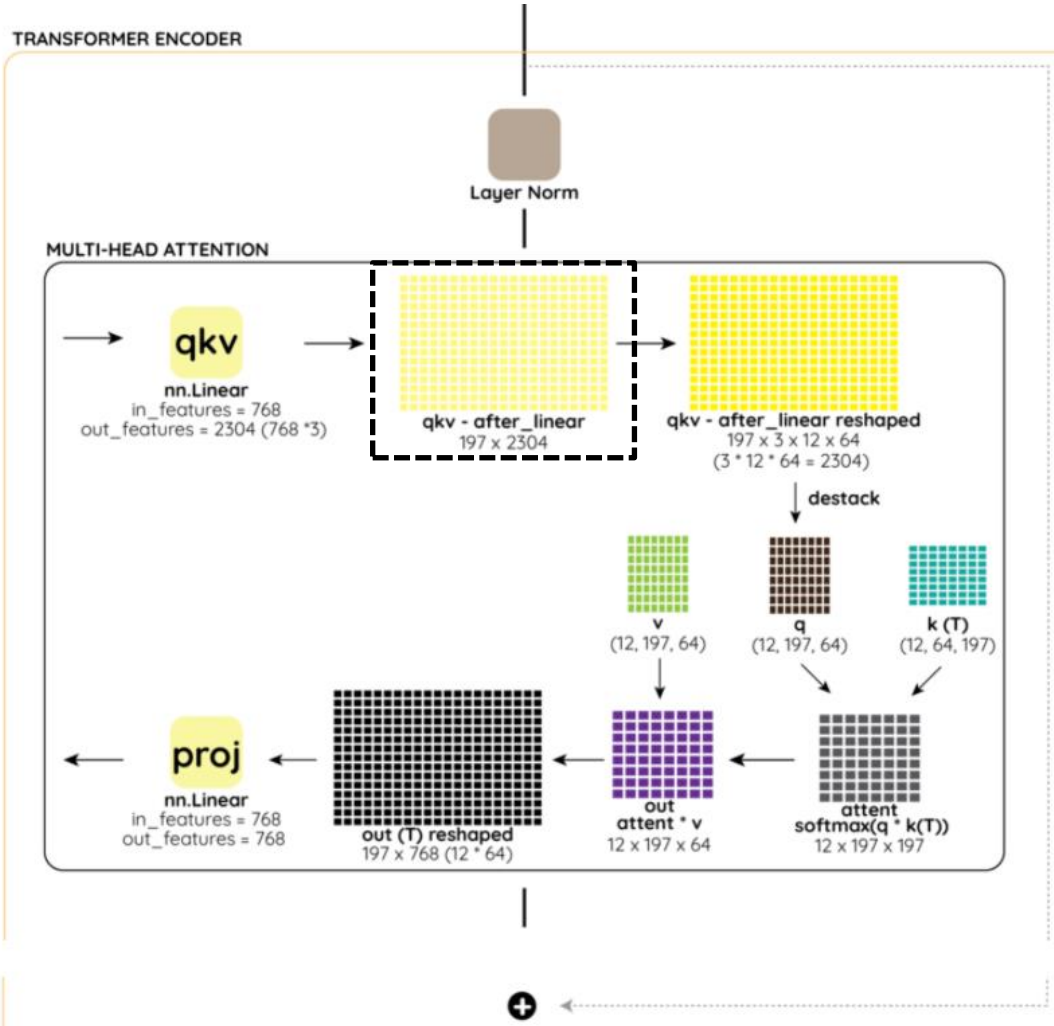


김성수

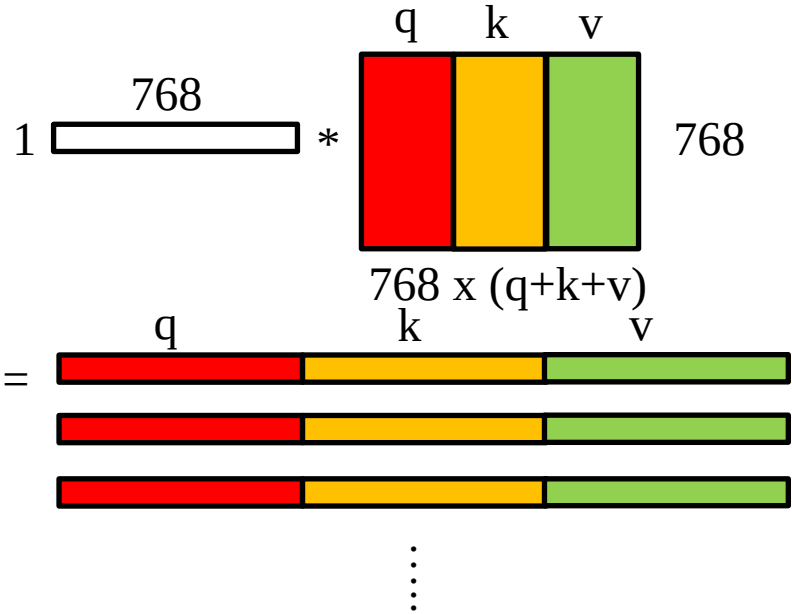
Vision Transformer[1]



- Details
  2. Multi-Head Attention

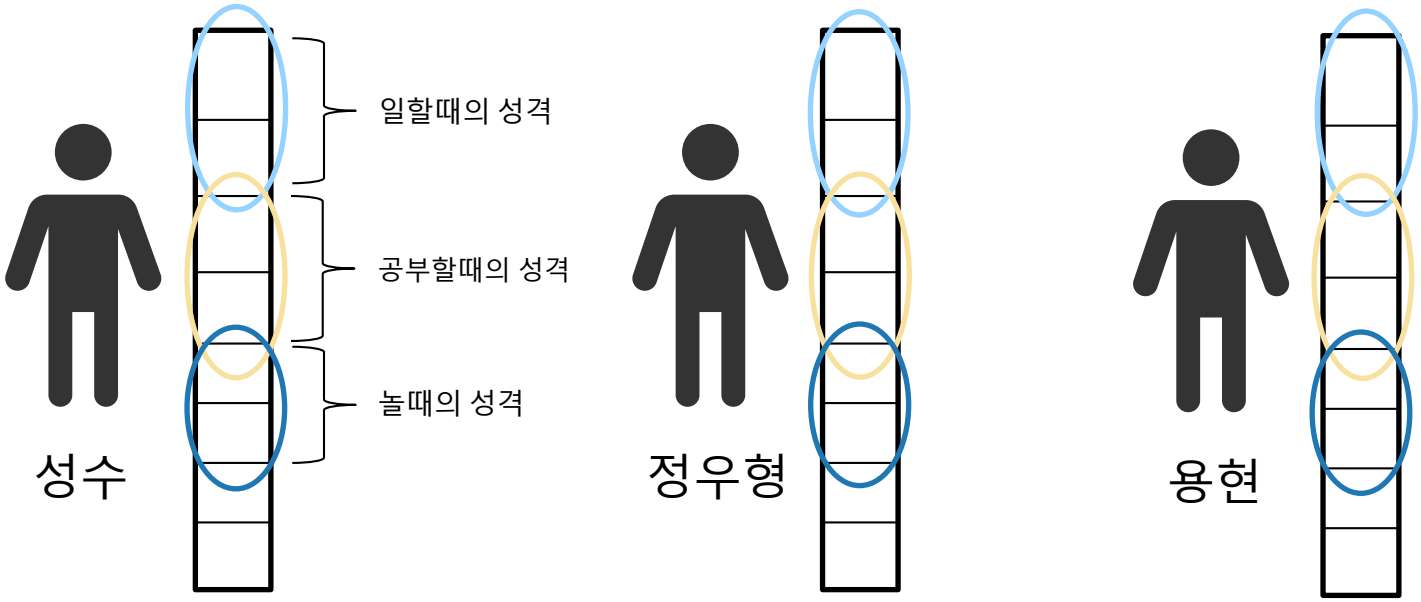


* q,k,v dimension= 768



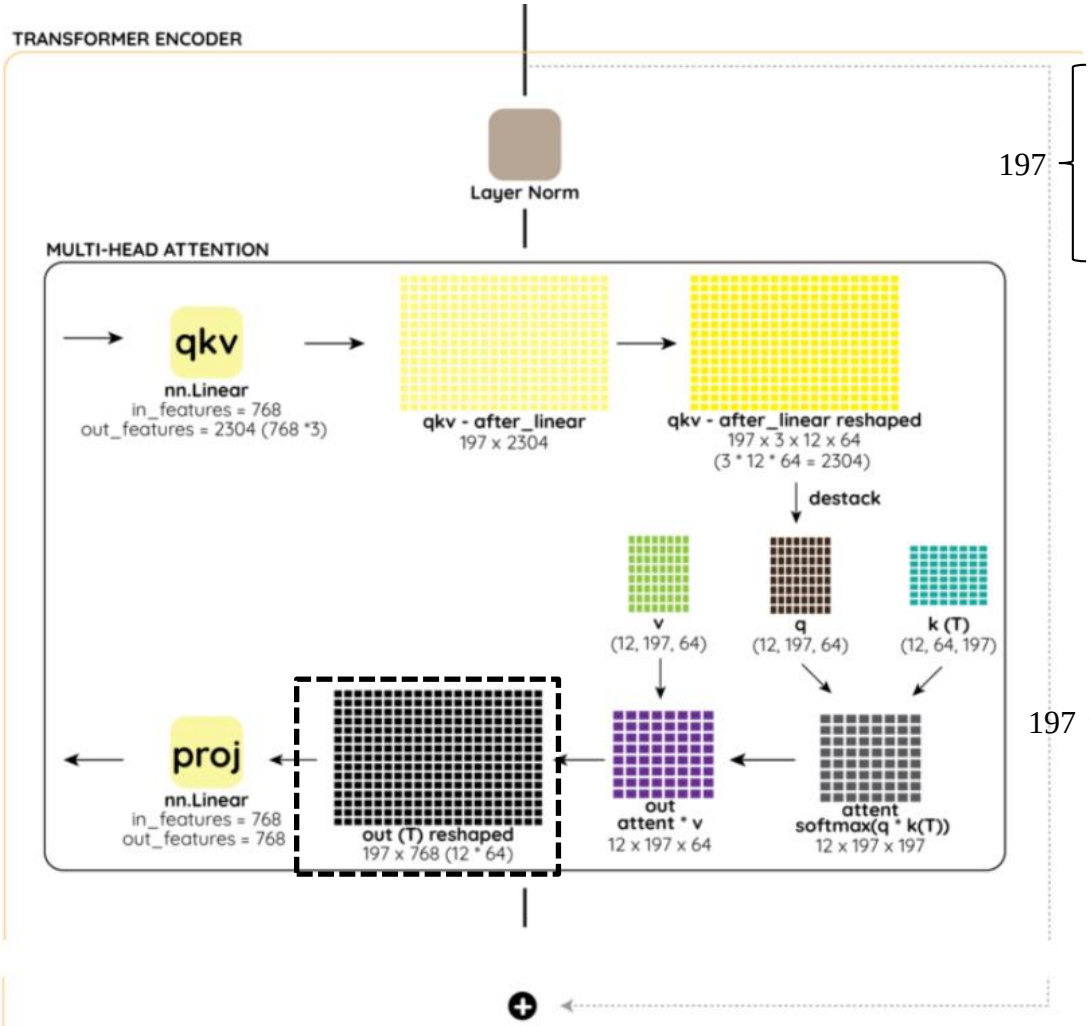
Vision Transformer[1]

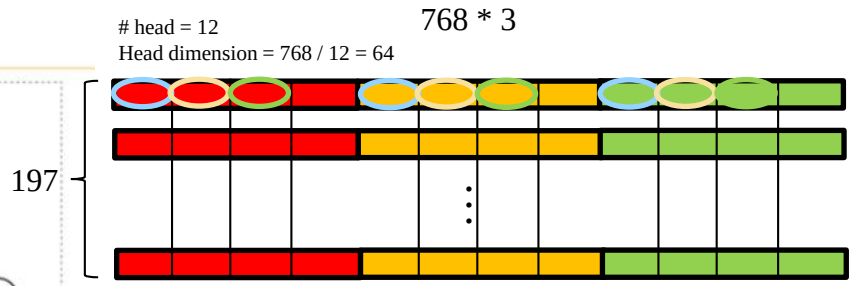
- Details
  2. Multi-Head Attention
     - Feature를 다양한 category로 나누어서 비교를 하면 사람의 성격을 더 잘 파악할 수 있지 않을까?
     - 일할때 모습끼리 비교, 공부할때 모습끼리 비교, 놀 때 모습끼리 비교
     - 각 모습끼리 비교를 하고 나중에 합치자!
     - Network가 다양한 관점에서 attention을 취할 수 있는 유연성을 갖게 함



Vision Transformer[1]

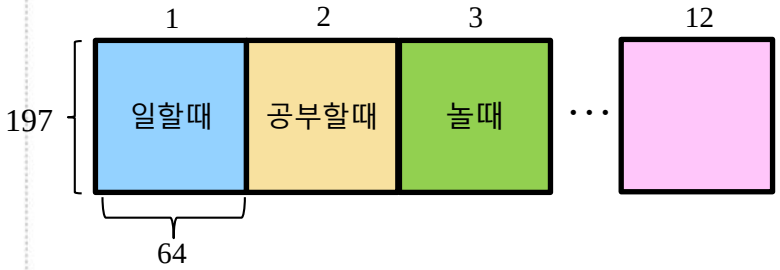
- Details
  2. Multi-Head Attention



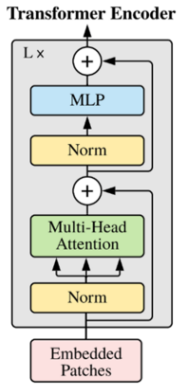


197 * (768 *3) → 197 * (3 * 12* 64)

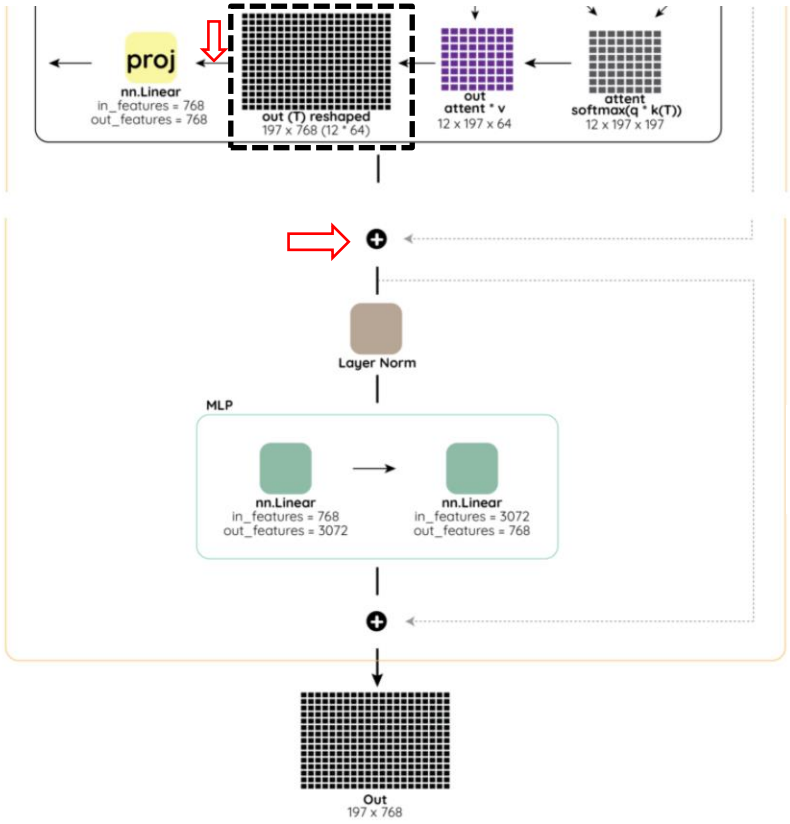
Q,K,V = 12 * 197 * 64
Attention score = 12 * 197 * 197
Output = 12 * 197 * 64

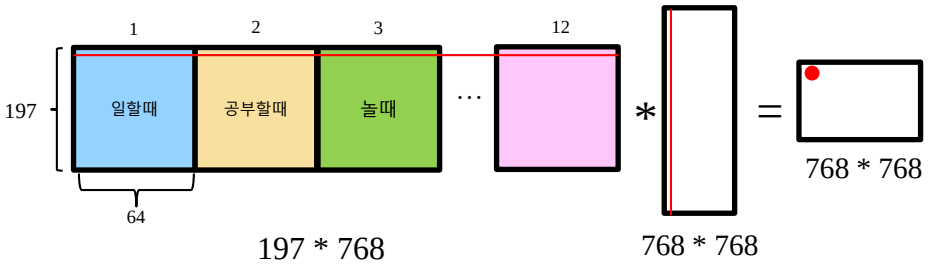


Vision Transformer[1]



- Details
  3. Linear layer in MHSA & MLP layer

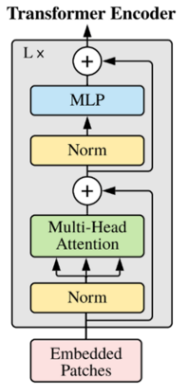




Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood

Translationally equivaraint ?
Local..?

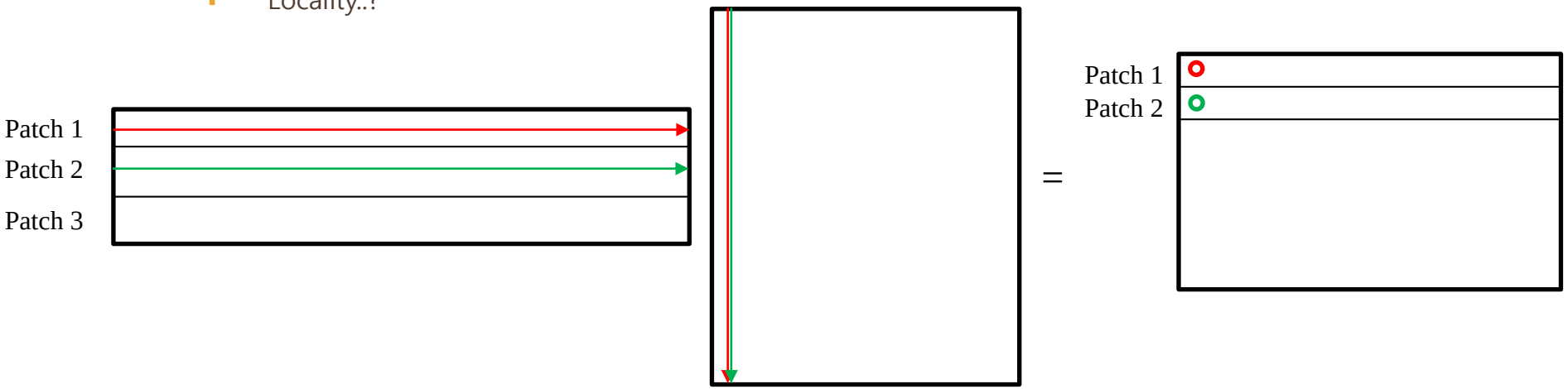
Vision Transformer[1]



- Details
  3. Linear layer in MHSA & MLP layer

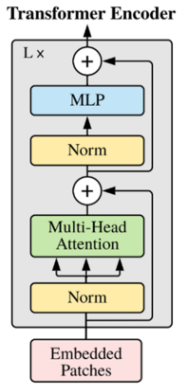
Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood

- Translationally equivariant = weight sharing
- Locality..?



- Patch가 만들어진 원래 image 입장에서
  - 모든 patch들이 같은 weight를 share하며(patch의 순서에 상관없이)feature를 refine하기 때문에 translationally equivariant
  - MLP에서의 feature refining 과정은 patch 각각에 대해 수행되기 때문에 local

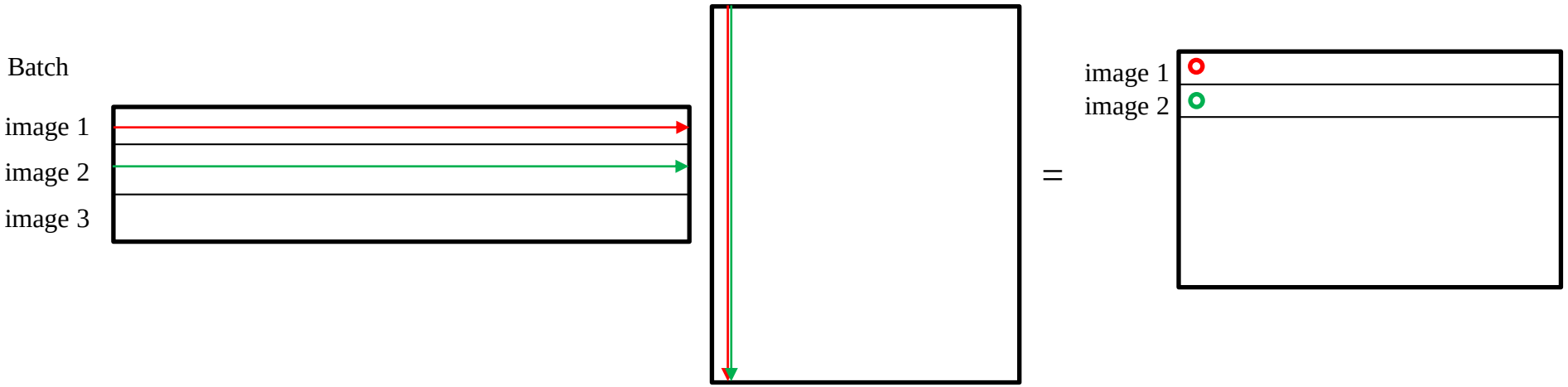
Vision Transformer[1]



- Details
  3. Linear layer in MHSA & MLP layer

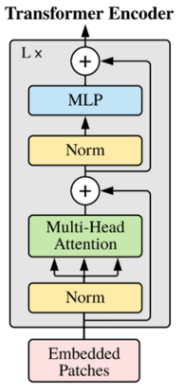
Inductive bias. We note that Vision Transformer has much less image-specific inductive bias than CNNs. In CNNs, locality, two-dimensional neighborhood structure, and translation equivariance are baked into each layer throughout the whole model. In ViT, only MLP layers are local and translationally equivariant, while the self-attention layers are global. The two-dimensional neighborhood

- MNIST를 flatten해서 MLP로 classification할때와 같은 상황아닌가? 왜 이제와서?

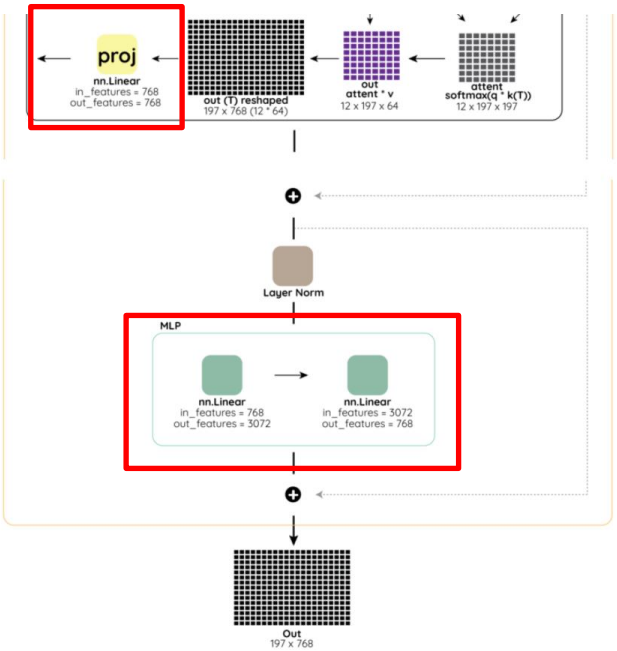


- 결론 : "Batch내의 여러 image" 에서 "Image 내의 여러 patch"가 되면서 MLP를 통해 해당 image에 대해 translationally equivariant와 locality를 갖도록 할 수 있게 된 것

Vision Transformer[1]



- Details
  3. Linear layer in MHSA & MLP layer
     - MHSA의 Linear layer와 MLP layer는 모양만 같을 뿐 역할이 전혀 다르다
     - Tranformer 구조에서 Skip connection의 출발지와 도착지가 갖는 의미를 생각하면 transformer를 파악하기 쉽다



Vision Transformer[1]

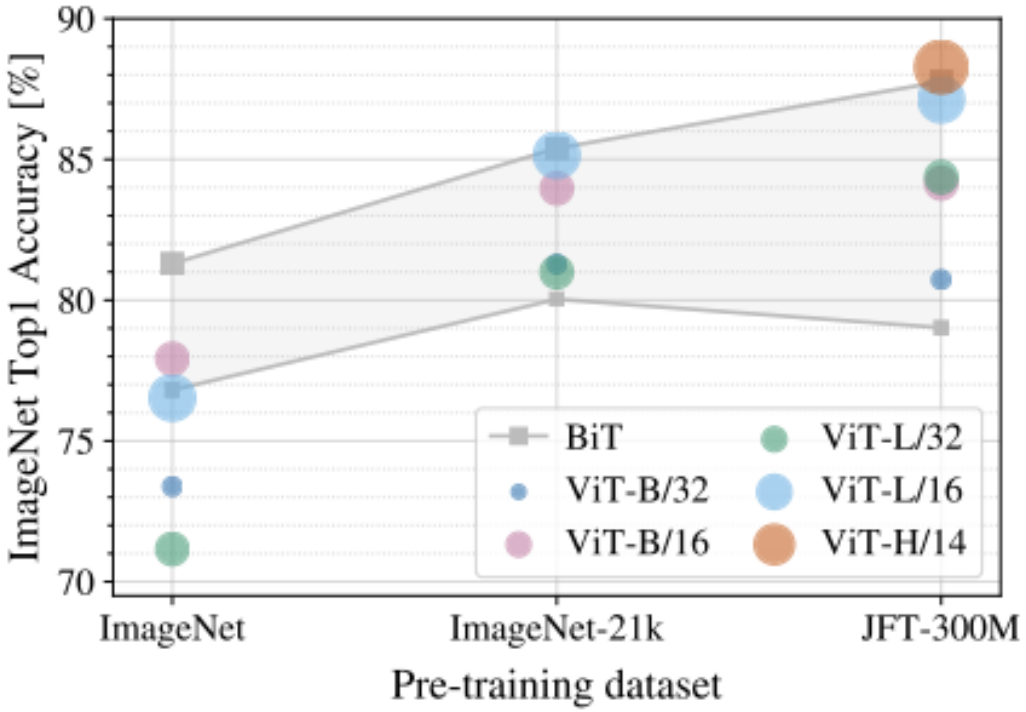
- Details
  4. Fine-Tuning and Higher Resolution
     - Input의 resolution이 다른 down stream task에선 pre-trained model의 positional embedding이 의미가 없어지므로, image의 size에 맞게 2D interpolation한 embedding을 사용
     - Patch extraction과 postion embedding 은 ViT에게 image의 2D structure를 알려주는 유일한 inductive bias
  5. Position embedding
     - 본 논문에서는 1D learnable positional embedding을 사용
     - Appendix D.4 [Table 8]

Pos. Emb.	Default/Stem	Every Layer	Every Layer-Shared
No Pos. Emb.	0.61382	N/A	N/A
1-D Pos. Emb.	0.64206	0.63964	0.64292
2-D Pos. Emb.	0.64001	0.64046	0.64022
Rel. Pos. Emb.	0.64032	N/A	N/A

     - Position embedding에 따른 성능 차이가 없는 이유?
       - Pixel 단위의 attention (224 x 224)와 다르게 patch 단위의 attention (14 x14) 에서 patch의 sapatial location을 배우는 것은 positional embedding에 상관 없이 easy하기 때문
       - Position embedding의 종류 보다는, learning rate과 같은 hyperparameter에 position representation 이 영향을 받음

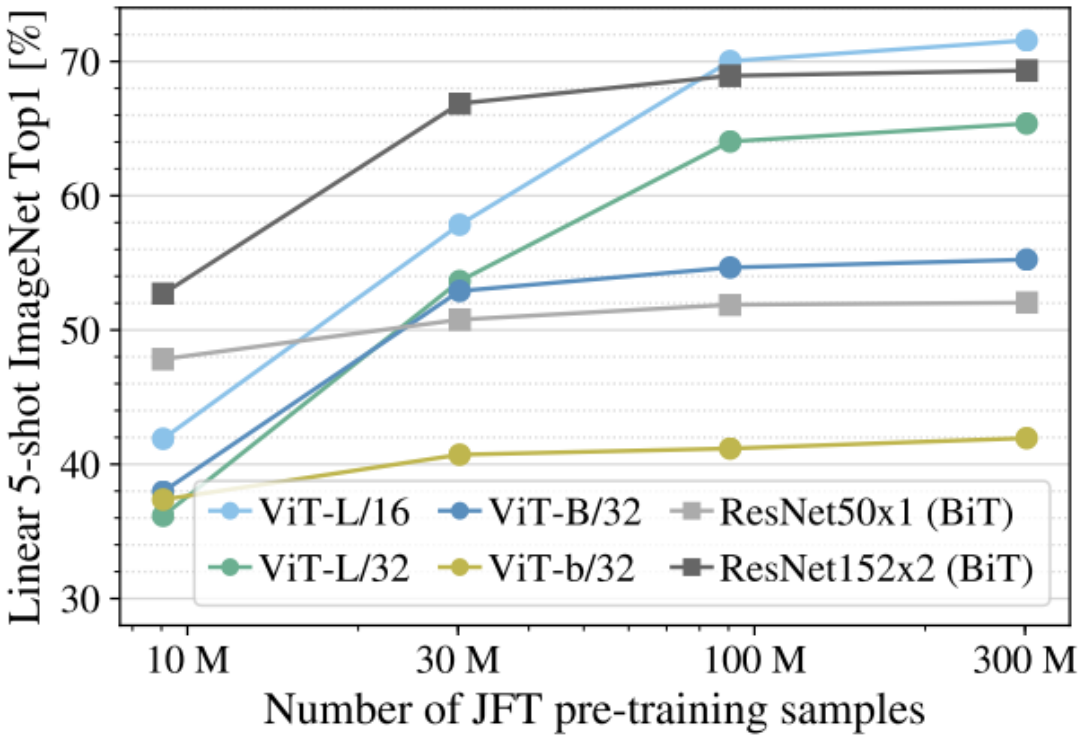
Vision Transformer[1]

- Experiment
  1. Transfer learning
     - Pre-train data 가 적을 때 : Data에 대한 assumption을 기반으로 학습하기 때문에 strong Inductive bias를 가진 model (CNN) 이 그렇지 않은 model보다 성능이 좋음
     - Pre-train data가 충분히 많을 때 : Weak Inductive bias (ViT)를 가진 model이 성능이 좋음



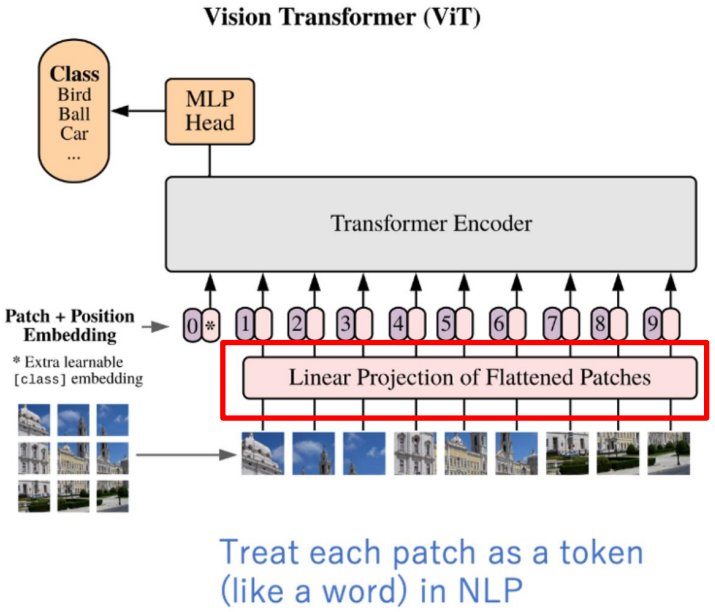
Vision Transformer[1]

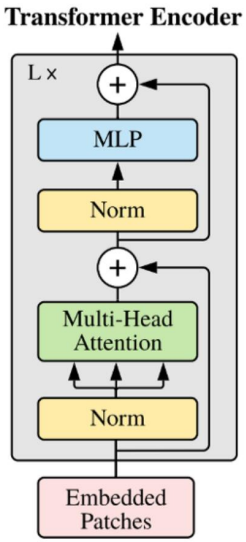
- Experiment
  2. Linear few-shot evaluation
     - "Class를 구분하는법"을 배우는 few-shot learning
     - Pre-train set이 커질 수록 ViT의 성능이 더 우수

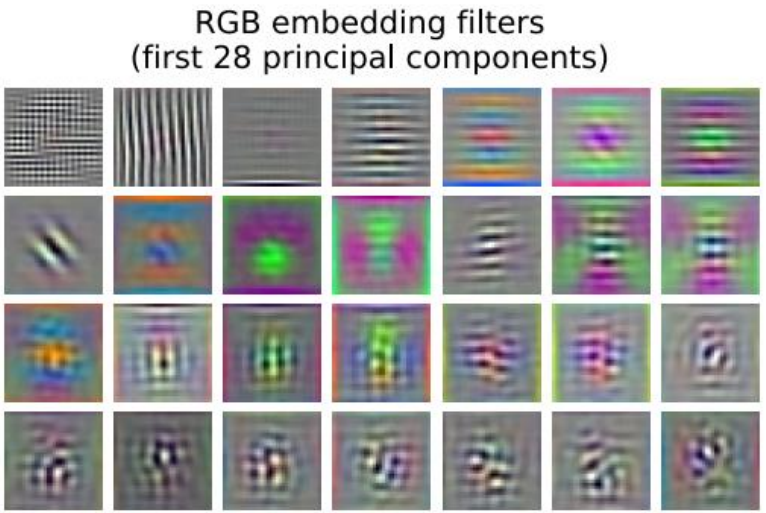


Vision Transformer[1]

- Experiment
  3. Linear projection layer
     - Patch 들의 feature를 뽑아내는 부분
     - 첫 28개의 Principal component를 visualize

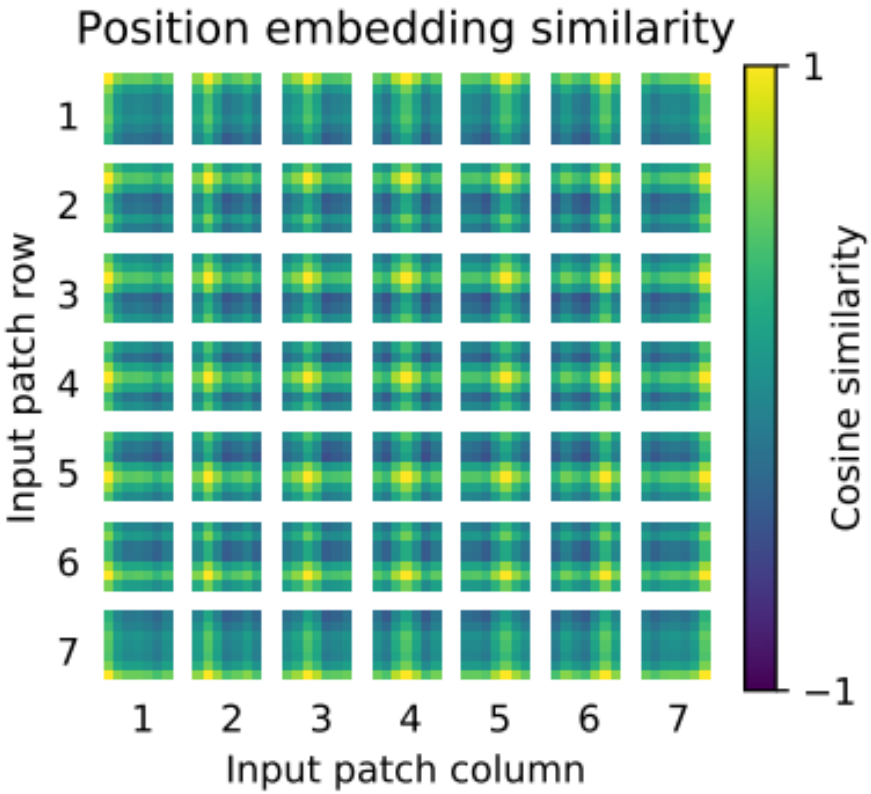


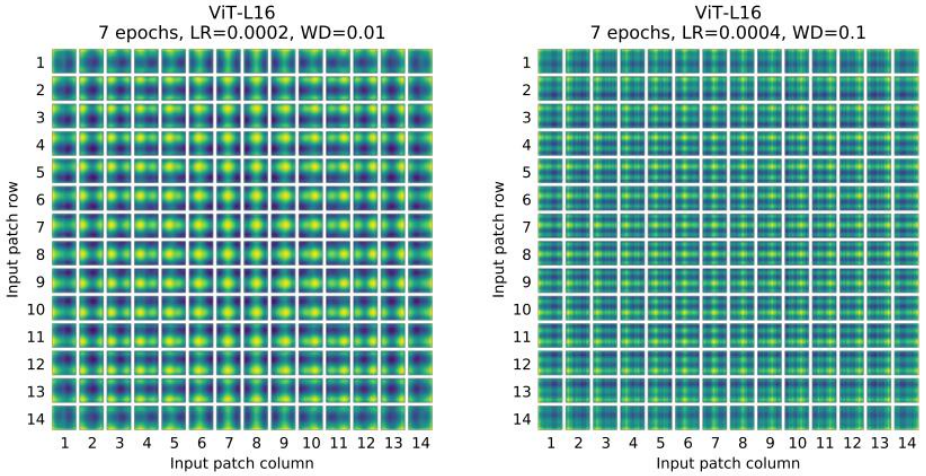




Vision Transformer[1]

- Experiment
  3. Positional Embedding
     - 각 patch에 해당하는 postional embedding간 cosine similarity
     - 비슷한 position 일수록, positional embedding이 유사함
     - Hyperparameter에 따라 positional representation이 상이

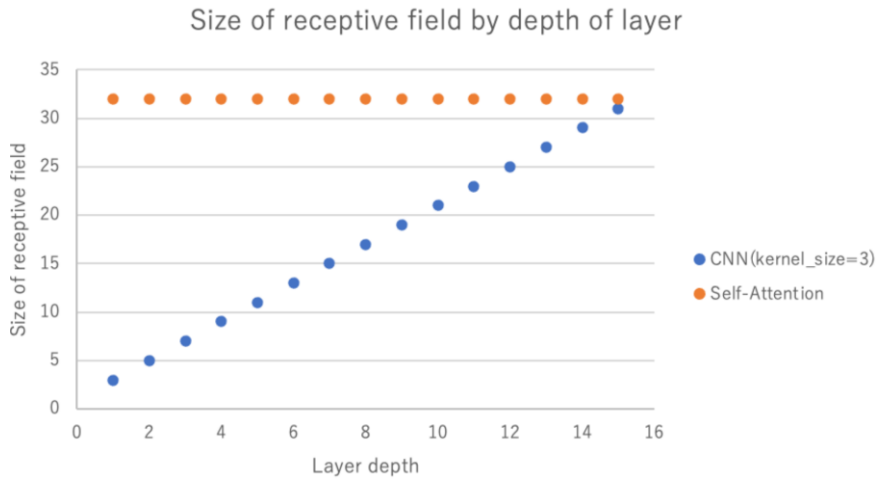




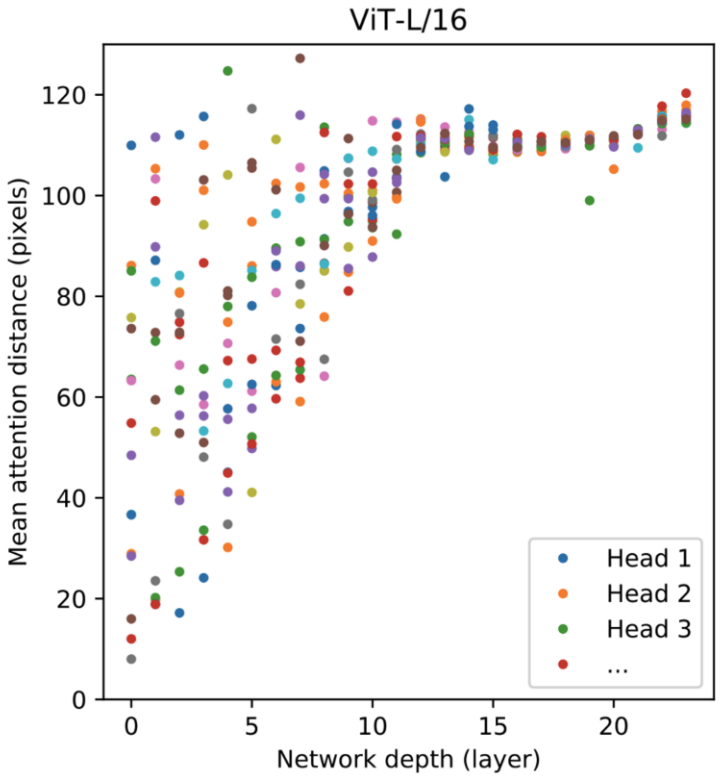
Appendix D.4 Figure 10

Vision Transformer[1]

- Experiment
  4. Attention distance
     - Attention head간 distance
     - CNN의 receptive field 와 동일
       - CNN : Locality라는 inductive bias에 의해 layer가 깊어짐에 따라 receptive field가 확장됨
       - ViT : 처음부터 image의 모든 부분을 학습해 나감



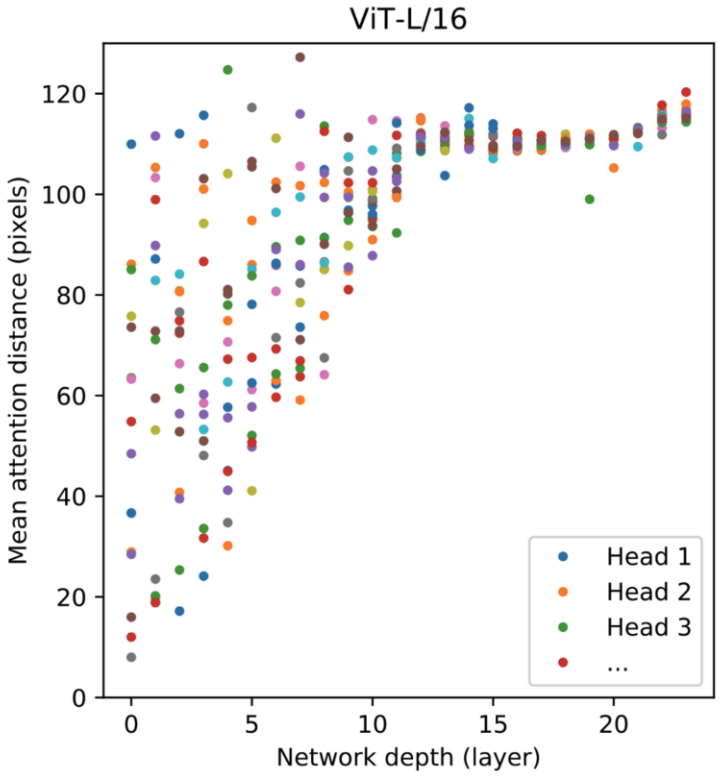
CNN

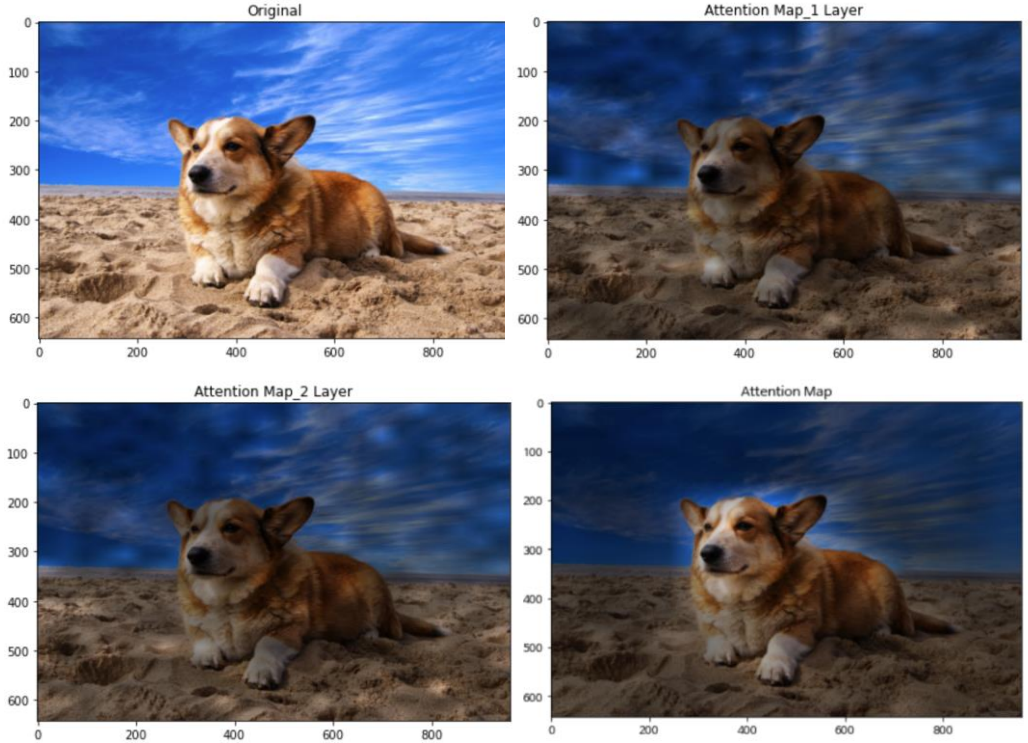


ViT

Vision Transformer[1]

- Experiment
  4. Attention distance
     - Attention head간 distance
     - CNN의 receptive field 와 동일
       - CNN : Locality라는 inductive bias에 의해 layer가 깊어짐에 따라 receptive field가 확장됨
       - ViT : 처음부터 image의 모든 부분을 학습해 나감

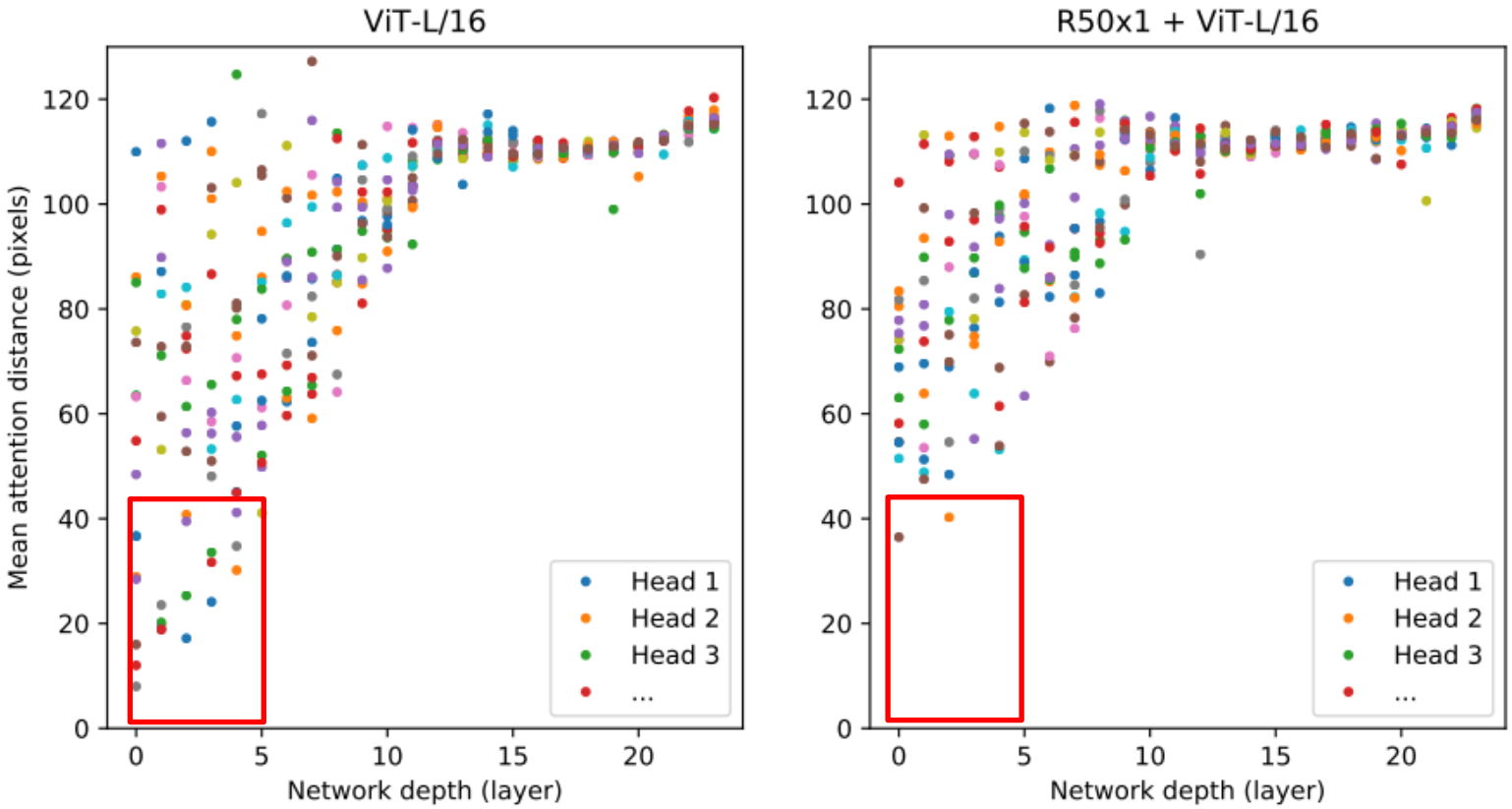




ViT

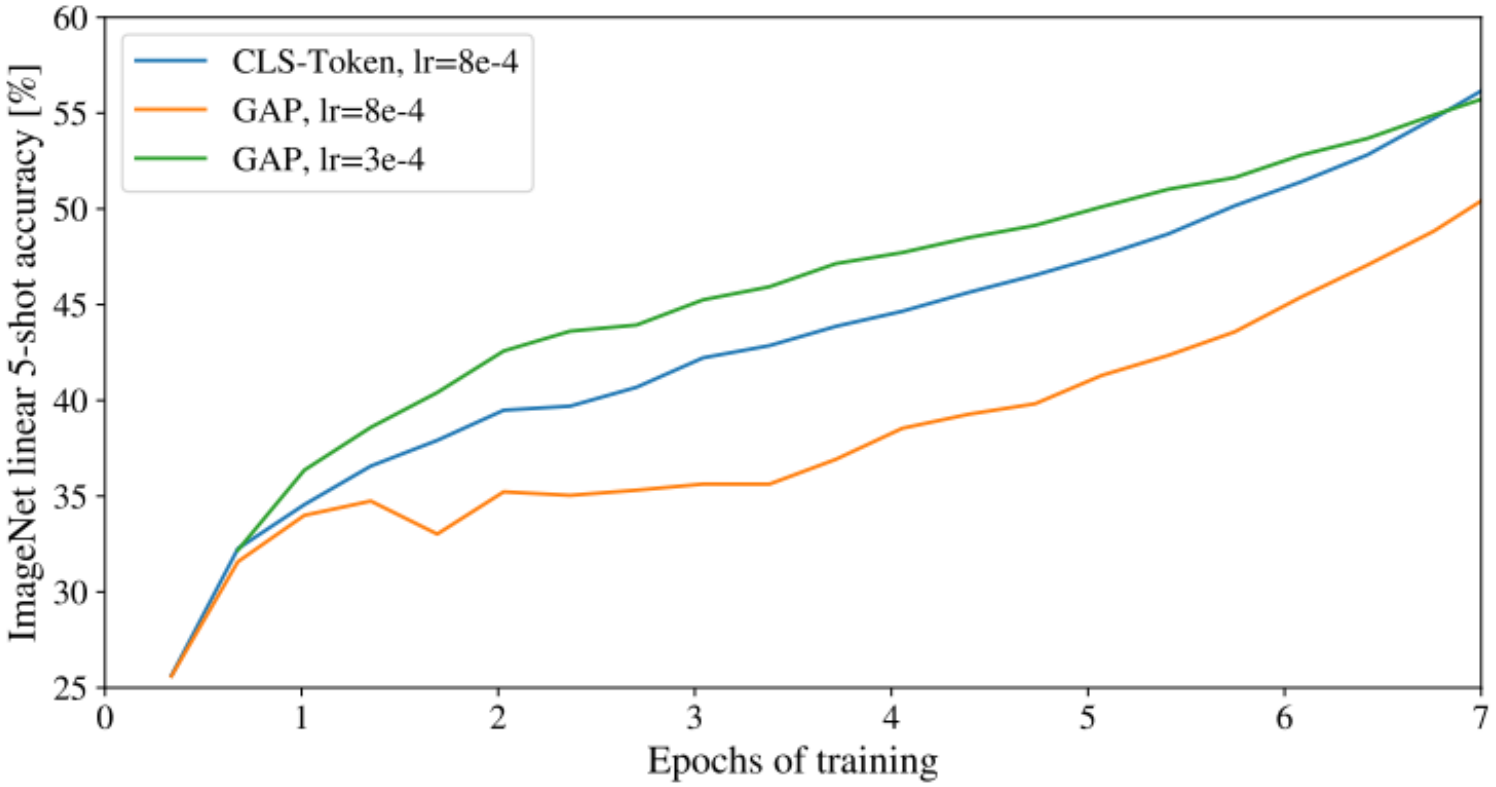
Vision Transformer[1]

- Experiment
  4. Attention distance
     - Hybrid – model : CNN으로 뽑은 feature를 ViT의 input으로 사용
     - CNN에서 이미 local 한 feature를 뽑았기 때문에, ViT의 attention-head들이 초반부 layer에서 global한 정보에 더 집중



Vision Transformer[1]

- Experiment
  5. [CLS] token VS GAP
     - CLS token을 사용하지 않고, 출력 embedding들의 GAP를 사용한다면?
     - 기존 ViT와 같은 learning rate를 사용하였을때 성능이 하락했으나, 각 방법이 서로 다른 learning rate에 대해 최적화 될 수 있음을 보임



Vision Transformer[1]

- Remaining questions
  - Transformer의 residual connection은 CNN과 동일하게 동작 하는가?
  - [CLS] token 은 다른 token들과 같은 방식으로 학습되어지는가?
  - CLS token을 사용하여 classification을 할 때와 GAP feature를 사용하여 classification을 할 때 각 출력 token이 배우는 representation은 동일한가?

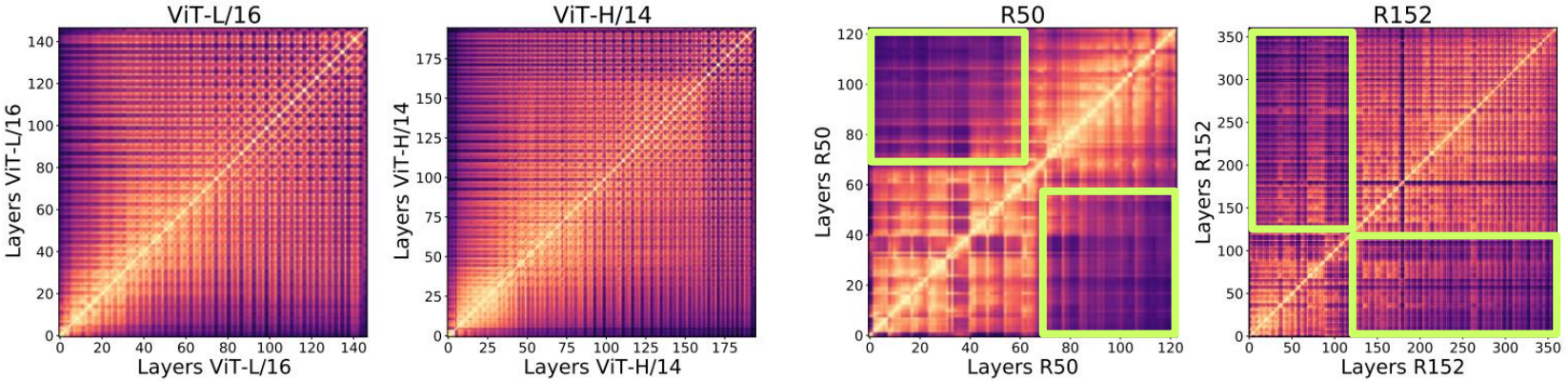
ViT vs CNN[2]

- Purpose of paper
  - Goal of paper : To understand whether there are differences in the way ViTs represent and solve image tasks compared to CNNs
  - Vision transformer가 "무엇"을 "어떻게" 학습하는지 CNN과 비교하여 직관적으로 이해하기 좋게 visualize한 논문
  - Representation Similarity and CKA (Centered Kernel Alignment)

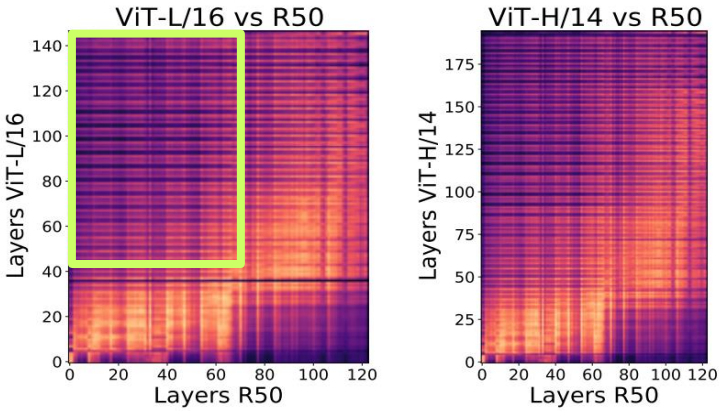
[2]M. Raghu, T. Unterthiner, S. Kornblith, C. Zhang, and A. Dosovitskiy, "Do vision transformers see like convolutional neural networks?," *Advances in Neural Information Processing Systems, vol. 34, 2021.*

ViT vs CNN[2]

- Details
  1. ViT는 CNN에 비해 모든 layer에서 uniform한 representation을 학습한다.
     - ViT는 모든 layer에서의 feature가 연관성이 있는 반면, ResNet은 높은 layer와 낮은 layer에서의 feature가 확실히 구분됨

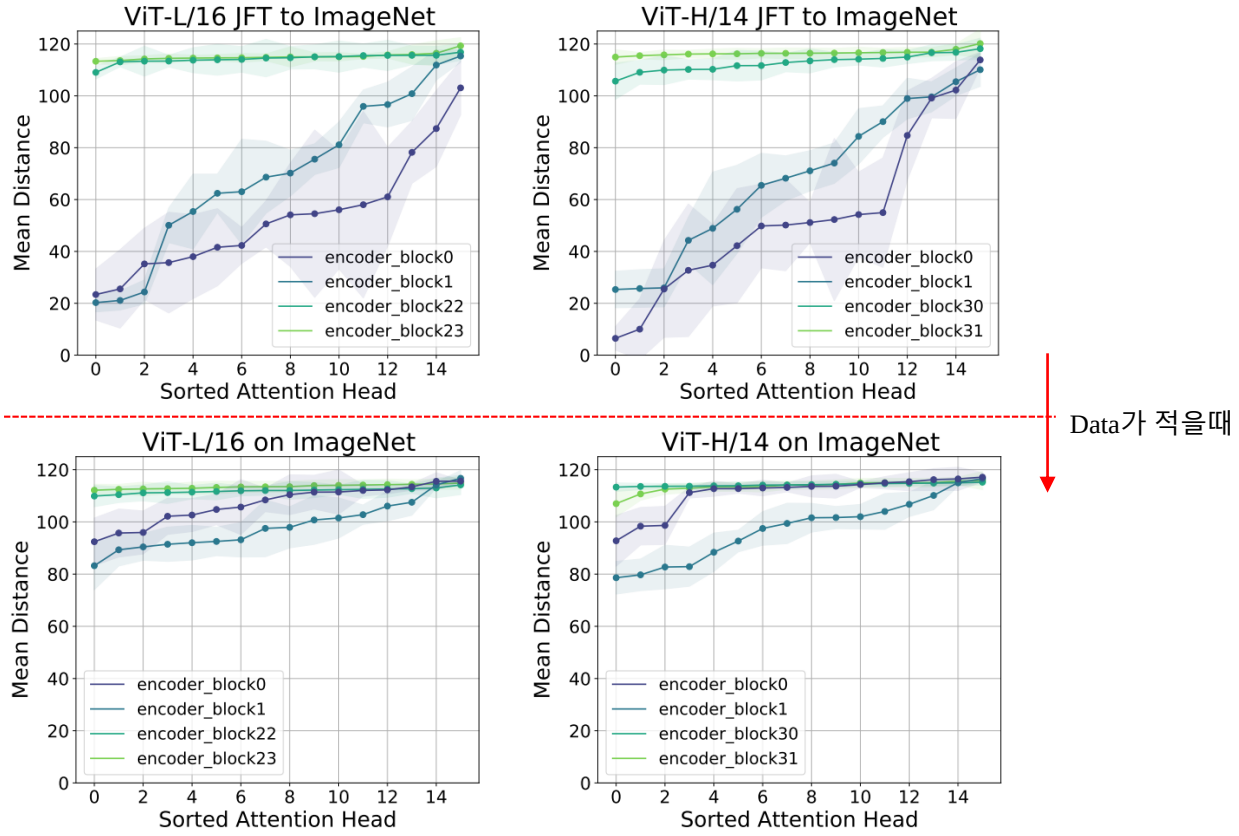


  2. ViT의 후반부 layer는 ResNet이 학습하지 못한 representation을 배운다.
     - 또한, ViT의 초반부 layer(0-40)의 feature 와 ResNet에서의 중반부 layer(0-60)의 feature가 비슷함



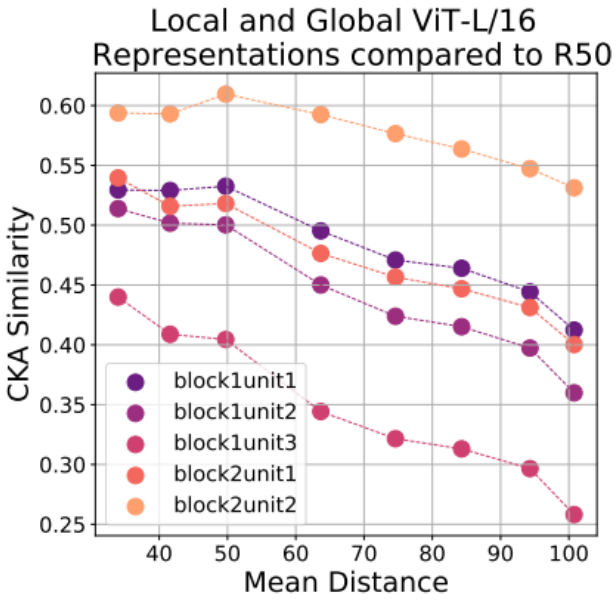
ViT vs CNN[2]

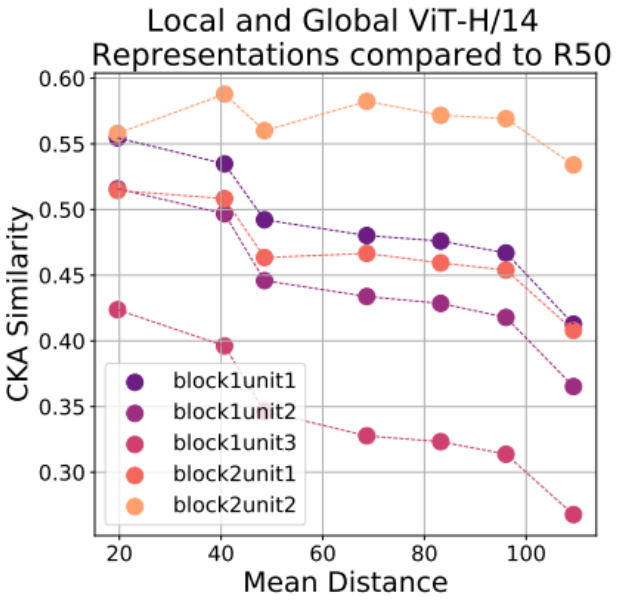
- Details
  3. ViT는 낮은 layer에서도 global한 정보와 local한 정보에 동시에 attend 할 수 있다.
     - 높은 layer에서는 거의 모든 attention head들이 global한 정보에 attend
     - **그러나, train data가 충분하지 않을 땐 ViT의 낮은 layer가 local한 정보에 attend하지 못함**

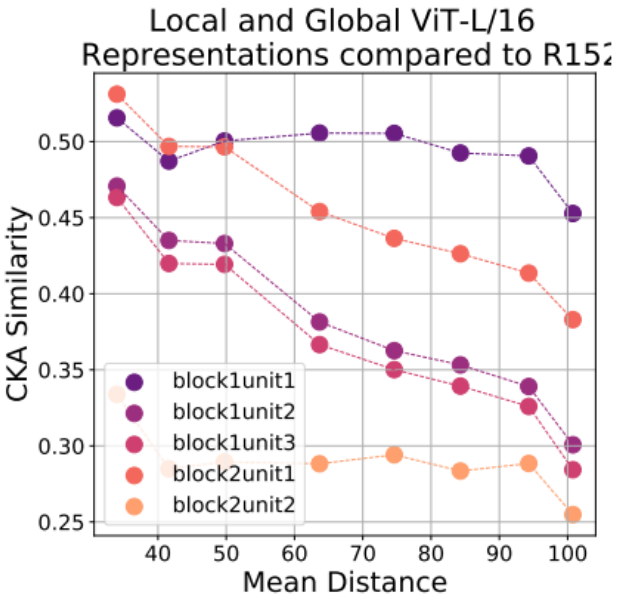


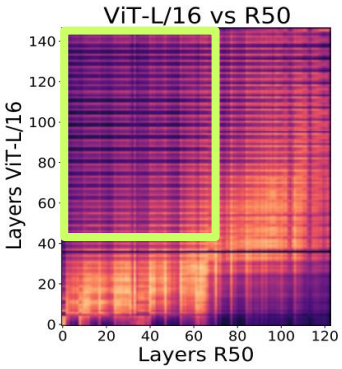
ViT vs CNN[2]

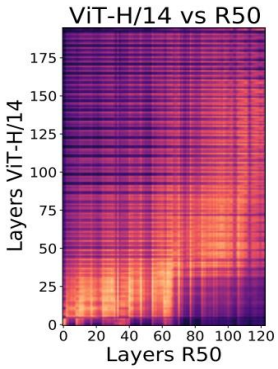
- Details
  4. ViT가 attend하는 local한 정보는 ResNet의 lower layer feature와 비슷하다.
     - ViT가 보는 feature를 attention head의 mean distance에 따라 ResNet의 lower layer feature과 비교

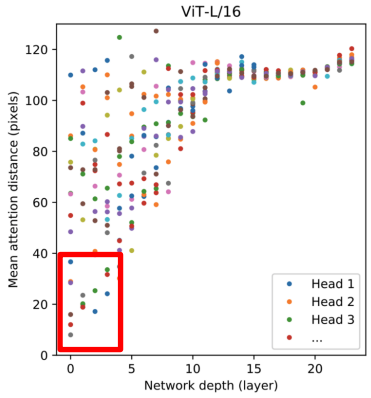










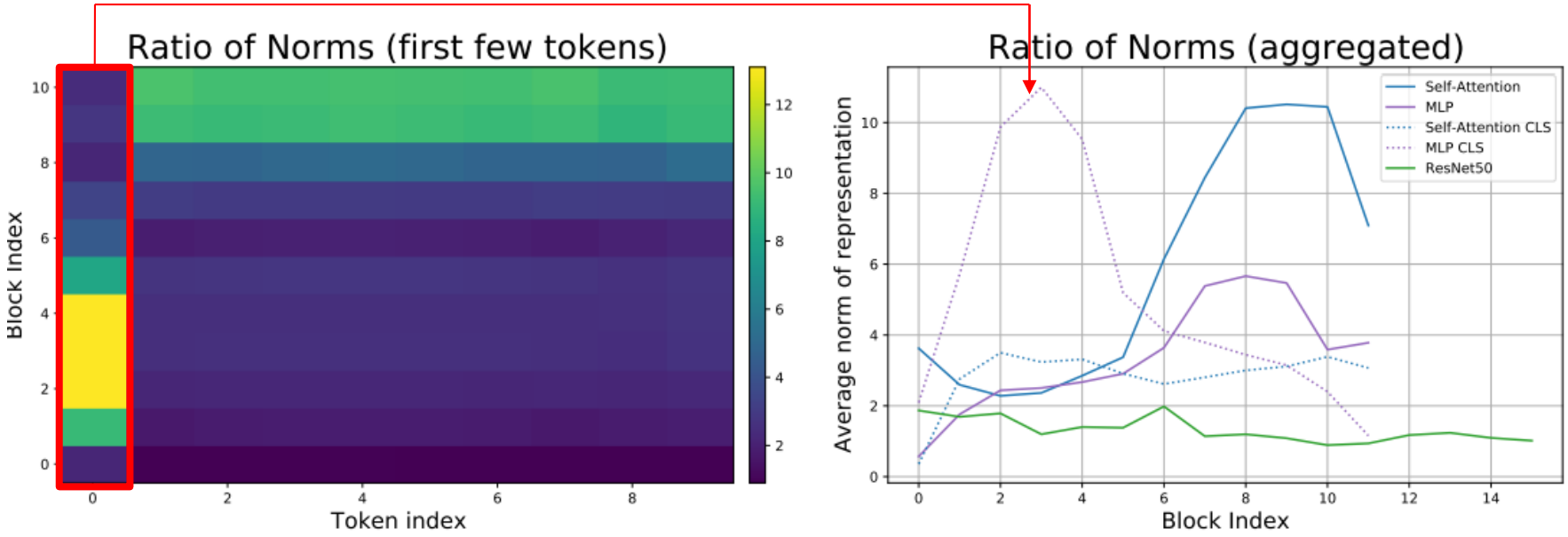


ViT vs CNN[2]

- Details
  5. CLS token의 학습 과정은 patch token과 skip connection 관점에서 반대 양상을 띈다.
     - Ratio of Norms : Skip connection 을 통과한 feature와 main branch를 통과한 feature간 norm

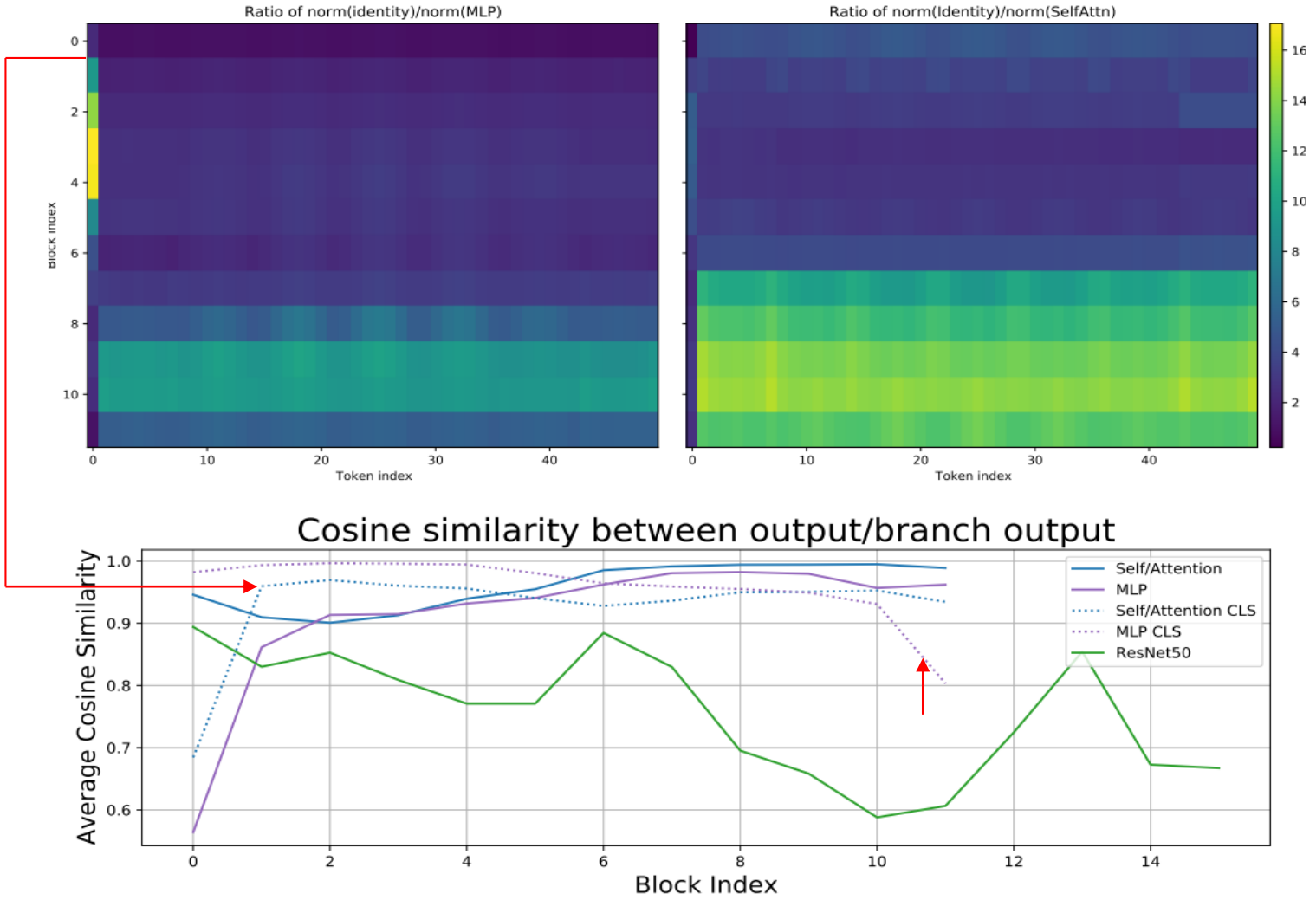
$$||z_i||/||f(z_i)||$$

     - Norms 이 클수록 Skip connection을 많이 통과함을 의미
     - CLS token은 초반부 MLP layer의 skip connection을 많이 통과 하며,후반부 block 에서 main branch를 통과함
     - 또한, Skip connection은 CNN보다 ViT에서 훨씬 더 effective하게 동작함



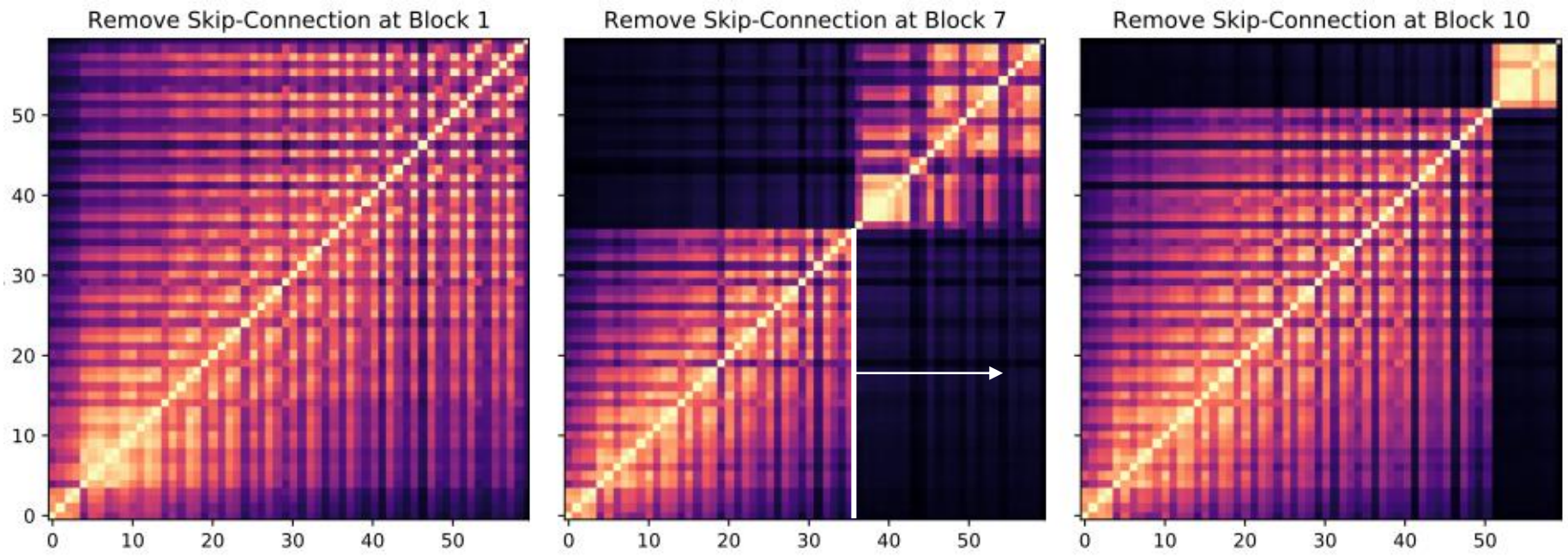
ViT vs CNN[2]

- Details
  5. CLS token의 학습 과정은 patch token과 skip connection 관점에서 반대 양상을 띈다.



ViT vs CNN[2]

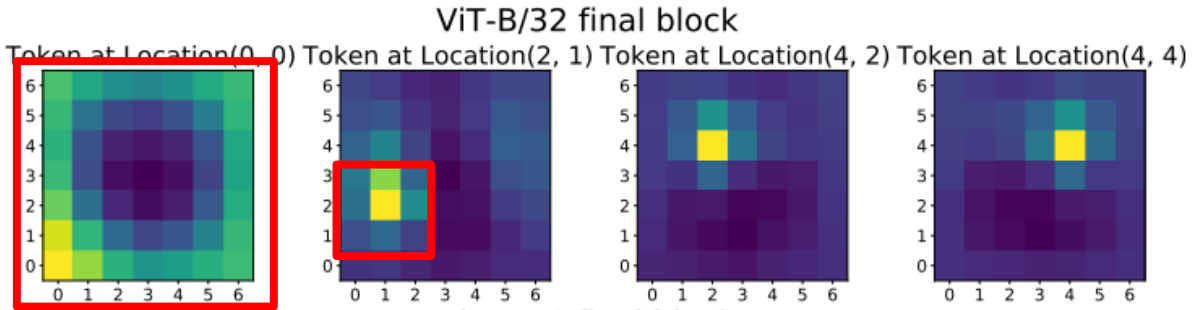
- Details
  6. ViT가 모든 layer에 걸쳐 uniform한 representaion을 배우는 것은 skip connection의 영향이 크다.

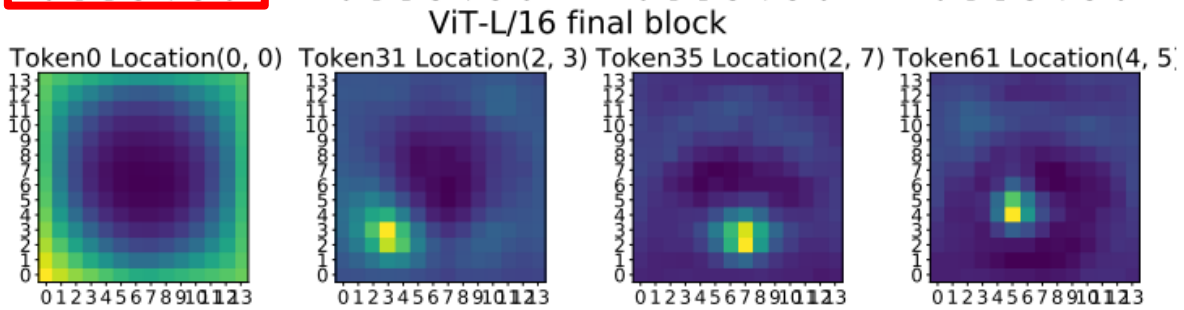


ViT vs CNN[2]

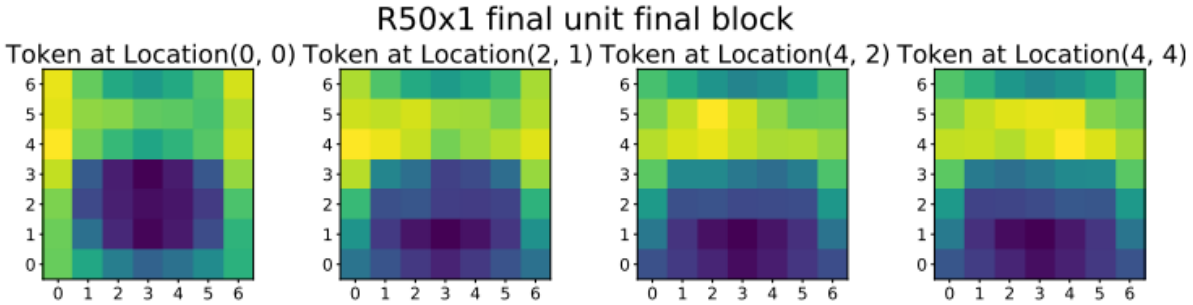
- Details
  7. CLS token을 통해 linear probing을 하였을 때, ViT는 higher layer까지 token의 spatial location 정보를 잘 유지한다.

[CLS]



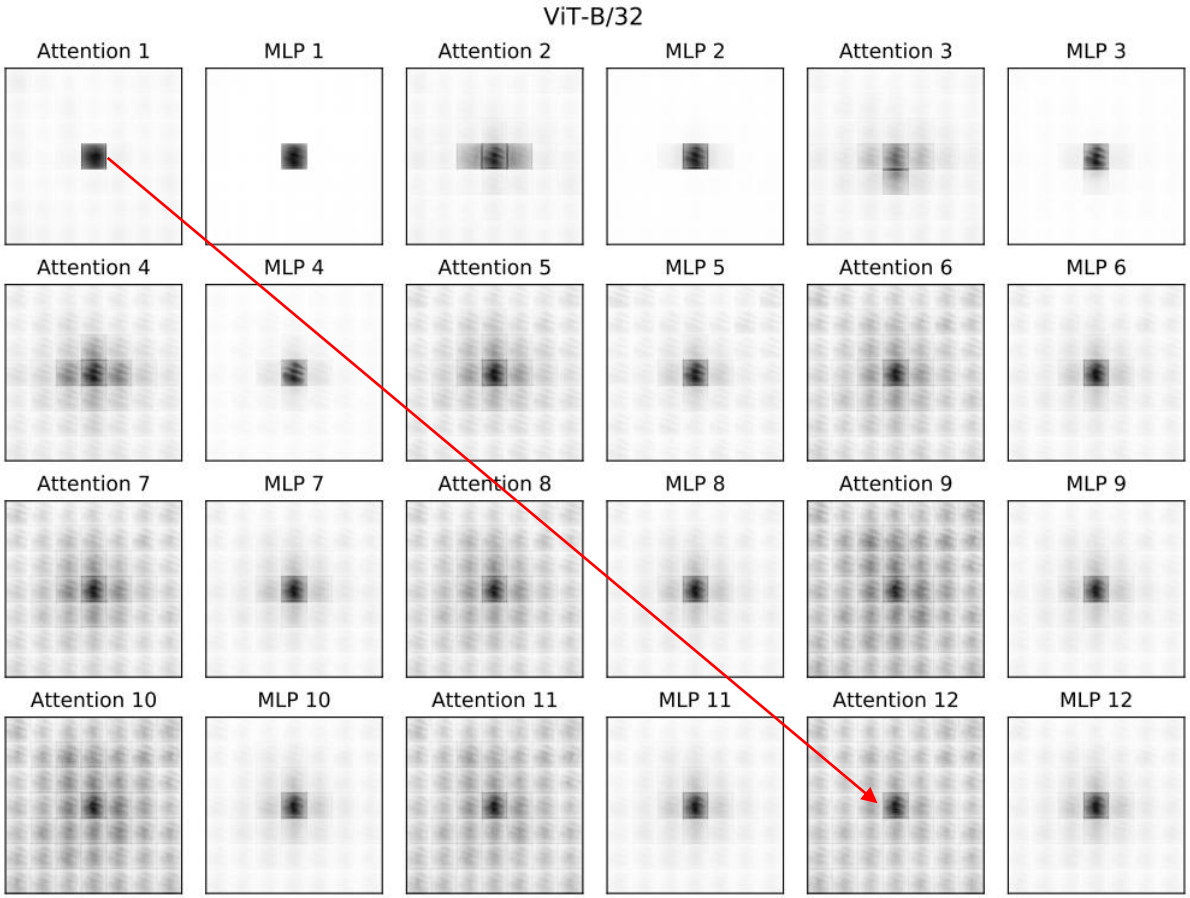


GAP



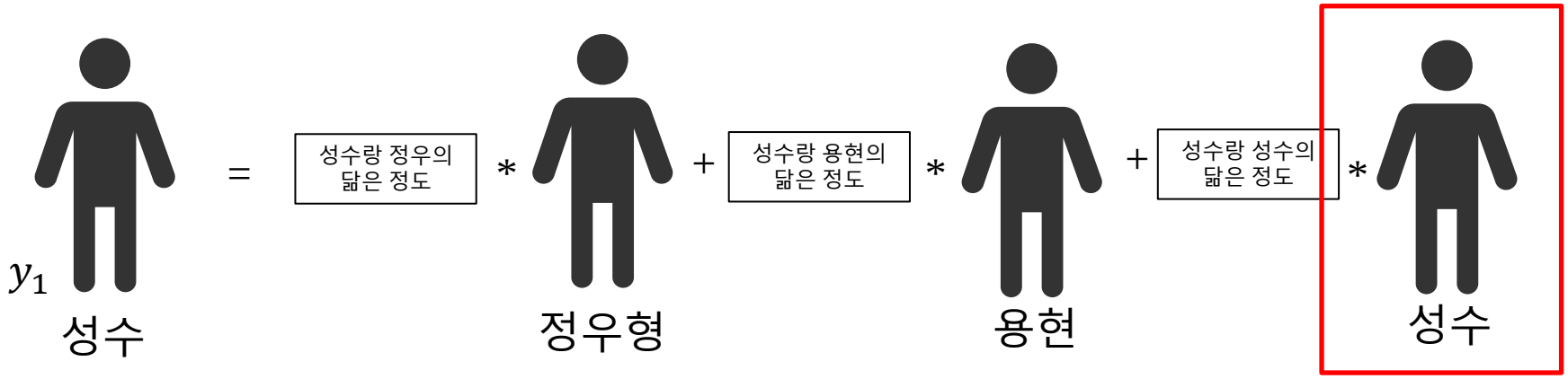
ViT vs CNN[2]

- Details
  7. CLS token을 통해 linear probing을 하였을 때, ViT는 higher layer까지 token의 spatial location 정보를 잘 유지한다.



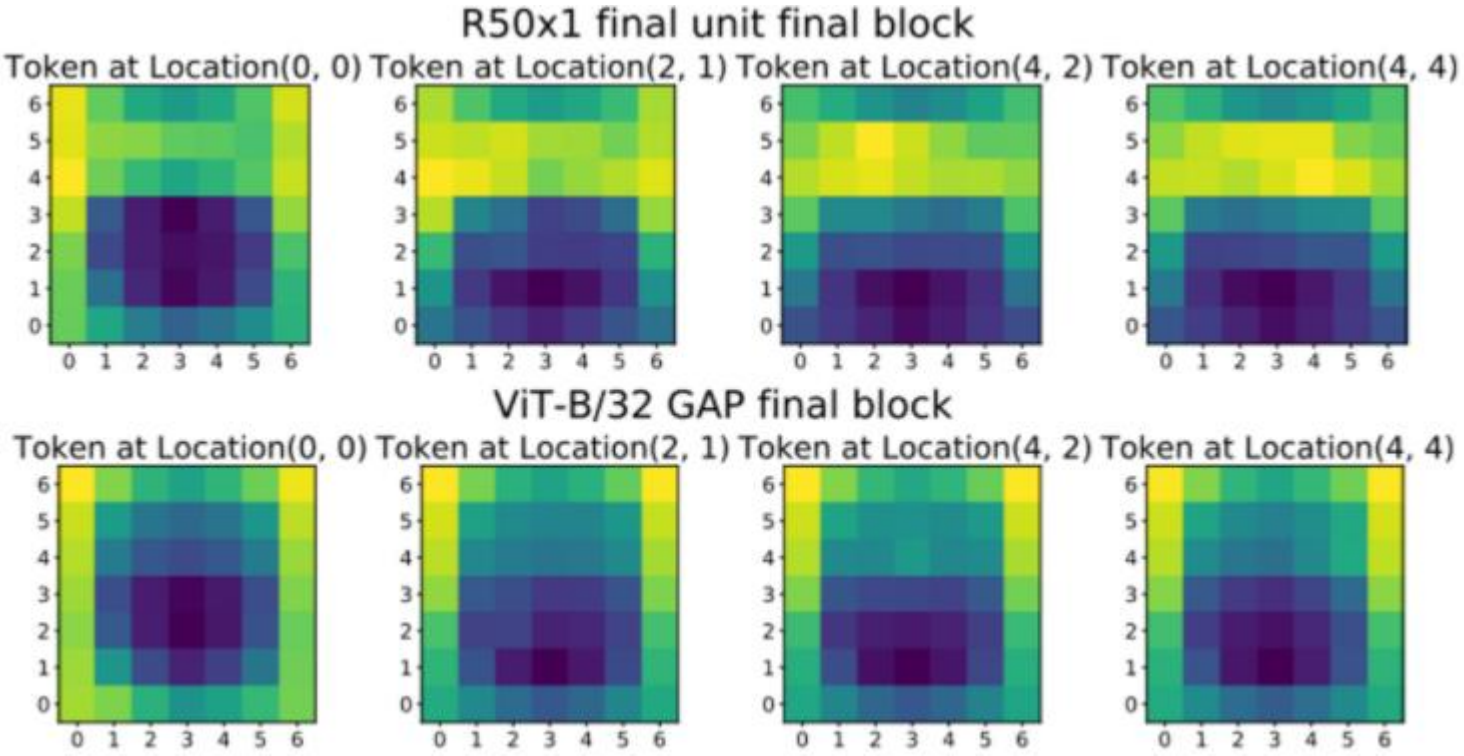
ViT vs CNN[2]

- Details
  7. CLS token을 통해 linear probing을 하였을 때, ViT는 higher layer까지 token의 spatial location 정보를 잘 유지한다.
     - Image는 language와 다르게 주변 patch와의 유사도가 매우 높음
       - Language : Highly semantic 하기 때문에 바로 옆 단어와 유사도가 있다는 보장이 없음
       - Image patch : 바로 옆 patch와 유사한 feature를 가질 가능성이 매우 높음
     - 각 patch 의 spatial location에 대한 정보가 출력단까지 이어지는 것은 image가 가지는 특성과 strong skip connection 때문이 아닐까..



ViT vs CNN[2]

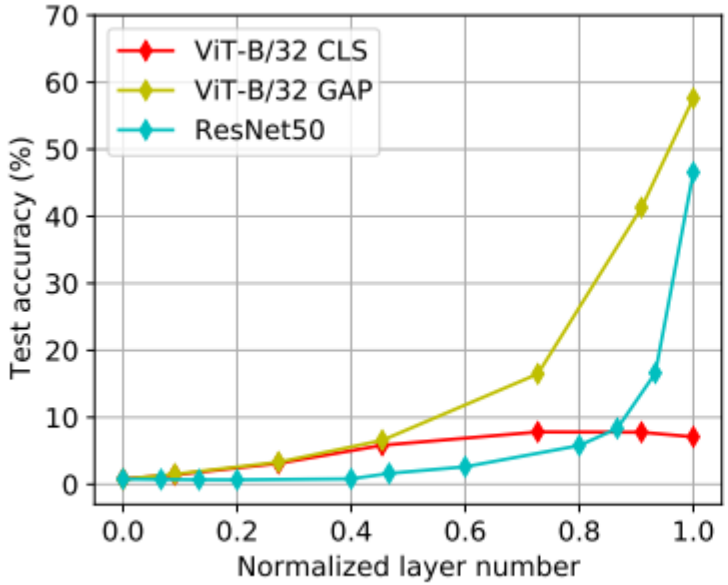
- Details
  7. CLS token을 통해 linear probing을 하였을 때, ViT는 higher layer까지 token의 spatial location 정보를 잘 유지한다.
     - ViT도 GAP를 사용할 경우, 각 token이 spatial location정보를 잃게되며 ResNet과 유사한 representation을 학습하게됨
     - GAP는 모든 token의 역할을 동일하게 만듦



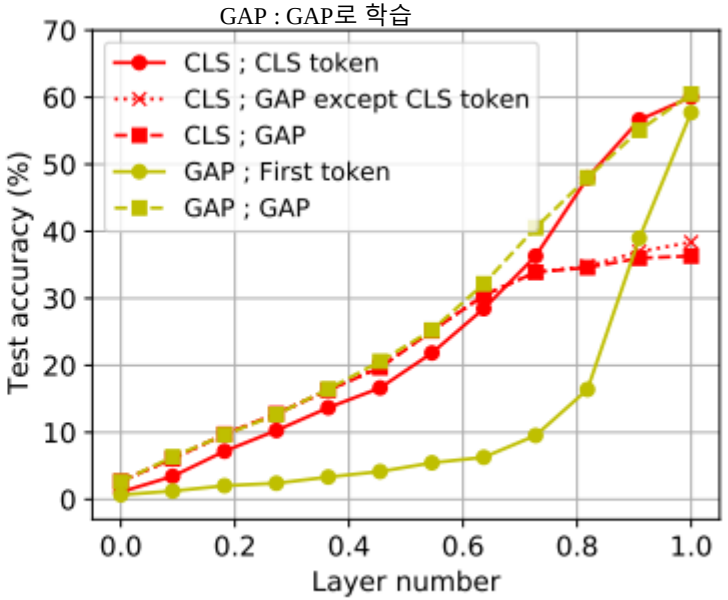
ViT vs CNN[2]

- Details
  8. CLS token을 사용할때와 GAP를 사용할때의 학습 차이는 후반부 layer에서 크게 발생한다.

CLS token : CLS token으로 학습
GAP : GAP로 학습



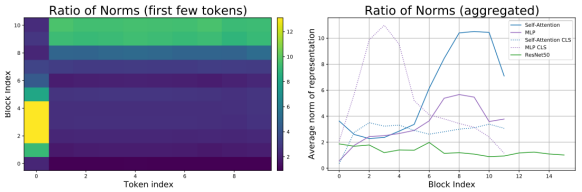
(a) Individual token evaluation



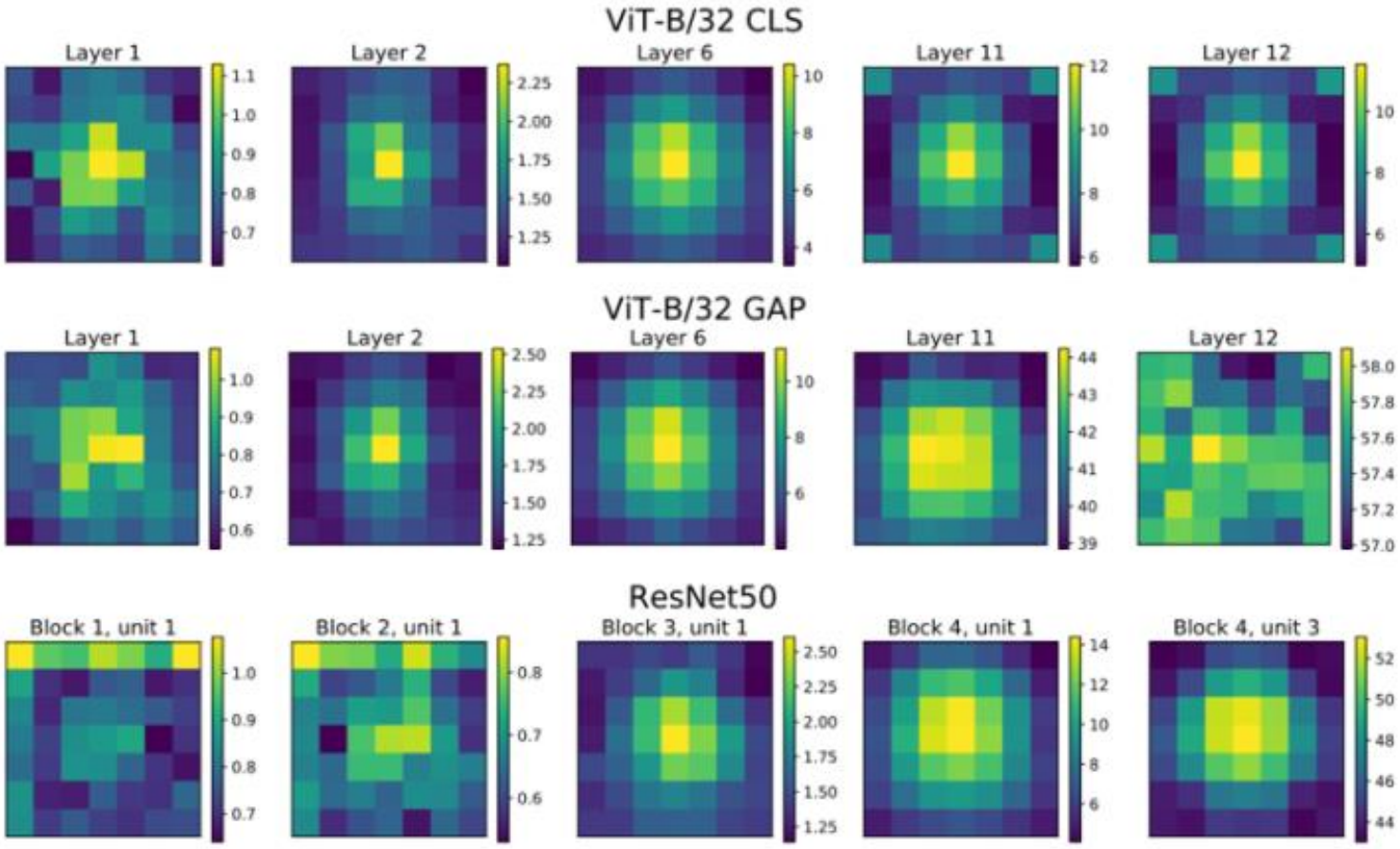
(b) CLS vs GAP models

김성수

ViT vs CNN[2]

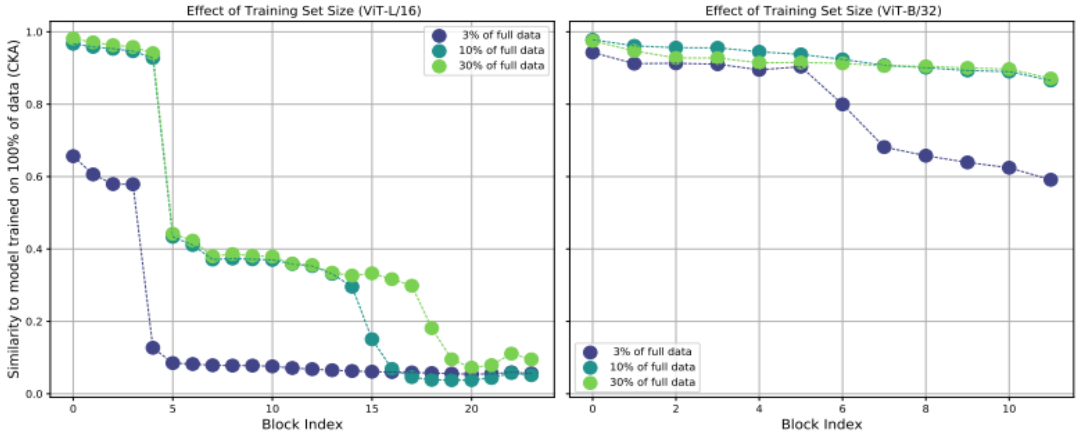


- Details
  8. CLS token을 사용할때와 GAP를 사용할때의 학습 차이는 후반부 layer에서 크게 발생한다.
     - 학습의 차이는 classifier와 가까운 후반부 layer일수록 크게 존재하고, 중반부 layer까지는 방식에 상관없이 비슷한 representation을 학습한다.



ViT vs CNN[2]

- Details
  9. Train data가 충분하지 않을 때, ViT는 higher layer에서 충분한 representation을 학습하지 못한다.
     - 반면에 data의 양에 상관 없이, lower layer에서의 representation은 유지됨
     - Intermediate representation은 data의 양에 큰 영향을 받으며, 성능에 미치는 영향이 큼
     - 학습 데이터가 많을 수록 ViT는 중간 layer에서 high quality의 representation을 배우게 됨



(a) JFT-300M vs ImageNet pre-training

(b) ViTs vs ResNets