# VIT:

Vision Transformer

GYUCHEOL LEE

# Contents

# Inductive Bias

- Inductive Bias(귀납적 편향)는 Training(학습)과정에서 보지 못한 데이터에 대해서도 적절한 귀납적 추론이 가능하도록 하기 위해 모델이 가지고 있는 가정들의 집합을 의미합니다.
- Inductive Bias(귀납적 편향)는 모델이 데이터에서 패턴을 학습하고 일반화하는 방식에 영향을 줍니다.
- Inductive Bias(귀납적 편향)는 모델의 학습 알고리즘, 가설 공간, 학습 데이터의 특성에 따라 결정되며, 이를 이해하고 관리하여 모델의 성능을 향상시키고 더 나은 일반화 능력을 갖출 수 있습니다.

# Inductive Bias: Fully Connected Neural Network

- Fully Connected Neural Network(완전 연결 신경망)는 입력 노드와 출력 노드가 모두 연결되어 있으므로 구조적으로 특별한 Relational Inductive Bias(관계적 귀납적 편향)를 가정하지 않습니다.

# Inductive Bias: Convolutional Neural Network

- Convolutional Neural Network(합성곱 신경망)는 작은 크기의 커널로 이미지를 지역적으로 보며 동일한 크기의 커널로 이미지 전체를 본다는 점에서 Locality(지역성)와 Translational Invariance(이동 불변성) 특성을 가집니다.
- Locality(지역성): Convolutional Neural Network(합성곱 신경망)는 입력 이미지를 작은 지역적인 영역으로 나누어 처리합니다. 이는 이미지의 작은 부분에서 발생하는 패턴을 감지하고 지역적인 정보를 보존하는 데 도움이 됩니다. 각 뉴런은 이 지역적인 영역에만 연결되어 있으며, 이를 통해 네트워크가 지역적인 패턴을 인식하고 학습할 수 있습니다.
- Translational Invariance(이동 불변성): Convolutional Neural Network(합성곱 신경망)는 커널을 이미지 위를 이동시키면서 지역적인 패턴을 감지합니다. 이는 이미지 내의 객체나 패턴이 위치가 바뀌더라도 동일한 패턴을 감지할 수 있도록 도와줍니다. 이는 이미지 내의 객체의 위치에 무관하게 이미지를 인식하고 이해하는 데 도움을 줍니다.

# Inductive Bias: Recurrent Neural Network

- Recurrent Neural Network(순환 신경망)는 입력한 데이터들이 시간적 특성을 가지고 있다고 가정하므로 Sequentality(순차성)와 Temporal Invariance(시간 불변성) 특성을 가집니다.
- Sequentality(순차성): Recurrent Neural Network(순환 신경망)는 순차적인 데이터를 처리하는 데 특히 유용합니다. 각 입력은 이전 입력과 함께 현재 상태에 영향을 미치므로, 네트워크는 데이터의 시간적인 순서를 고려하여 처리할 수 있습니다. 이는 문장의 단어들이나 시계열 데이터의 연속적인 값을 예측하거나 생성하는 데 유용합니다.
- Temporal Invariance(시간 불변성): Recurrent Neural Network(순환 신경망)는 입력 데이터의 시간적인 특성을 고려하여 처리하지만, 이러한 특성이 데이터의 절대적인 시간적 위치에 의존하지 않습니다. 즉, 입력 데이터의 패턴이 시간에 따라 변할 수 있지만, Recurrent Neural Network(순환 신경망)는 이러한 변화에 대해 불변성을 가지며, 같은 패턴이 서로 다른 시간에 등장할 경우 이를 인식할 수 있습니다.

# Inductive Bias: Transformer

- Transformer(트랜스포머)는 Convolutional Neural Network(합성곱 신경망) 및 Recurrent Neural Network(순환 신경망) 보다 상대적으로 Inductive Bias(귀납적 편향)가 낮습니다.
- Transformer(트랜스포머)는 Attention Mechanism(어텐션 메커니즘)을 중심으로 구성되어 있으며, 입력 시퀀스의 전체적인 의미를 고려하여 처리합니다. 이에 따라 Transformer(트랜스포머)는 시간적이거나 공간적인 위치에 관계 없이 입력 데이터의 전체적인 의미를 학습하고 처리할 수 있습니다.

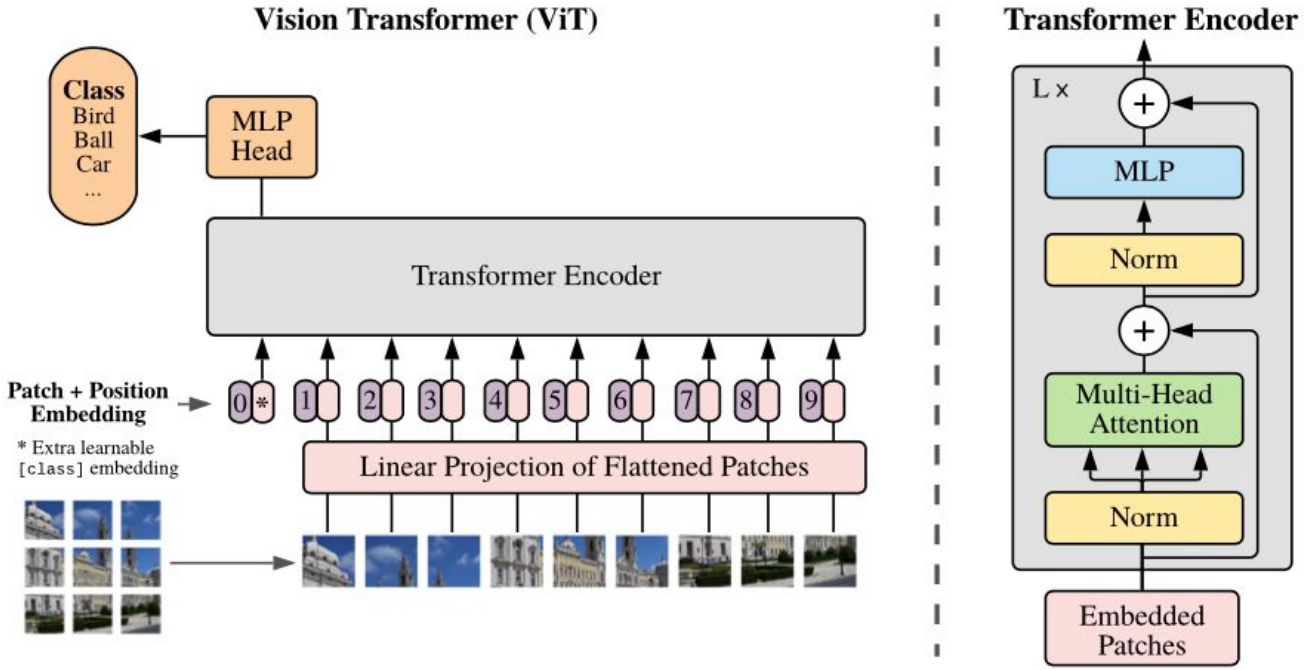
# Inductive Bias: Transformer

- Transformer(트랜스포머)를 이미지 처리에서 활용하려면 데이터가 많이 필요합니다. 그 이유는 Transformer(트랜스포머)의 Inductive Bias(귀납적 편향)가 Convolutional Neural Network(합성곱 신경망)보다 낮기 때문입니다.
- Transformer(트랜스포머)는 Attention Mechanism(어텐션 메커니즘)을 기반으로 하며, 입력 시퀀스의 전체적인 의미를 고려하여 처리합니다. 이에 반해 Convolutional Neural Network(합성곱 신경망)는 지역적인 특징을 감지하고 처리하는 데 특화되어 있습니다. 따라서 Transformer(트랜스포머)는 입력 이미지의 지역적인 패턴이나 구조를 명시적으로 고려하지 않고, 데이터의 전체적인 의미를 학습하게 됩니다.
- 이러한 특성으로 인해 Transformer(트랜스포머)는 더 많은 데이터를 필요로 합니다. 데이터의 다양성과 양이 증가함에 따라 Transformer(트랜스포머)는 다양한 시각적 패턴을 학습하고, 이를 통해 모델의 일반화 능력을 향상시킬 수 있습니다. 더 많은 데이터를 학습함으로써 Transformer(트랜스포머)는 이미지 처리에서 더 나은 성능을 발휘할 수 있게 됩니다.

# Vision Transformer

- Vision Transformer(비전 트랜스포머)는 자연어처리에서 사용되는 Transformer(트랜스포머)를 이미지 처리에서 그대로 적용하여 이미지 분류 작업에서 높은 성능을 도출했습니다.
- Vision Transformer(비전 트랜스포머)는 이미지를 패치로 분할한 후, 이를 자연어처리에서의 단어로 취급하여 각 패치의 Linear Embedding(선형 임베딩)을 순서대로 Transformer(트랜스포머)의 입력으로 넣어 이미지 분류 작업을 수행합니다.

# Vision Transformer: Architecture



# Vision Transformer: Linear Embedding

Linear Embedding (선형 임베딩)
Trainable Linear Projection (학습가능한 선형변환)
: $f(x) = Wx + b$
가중치 행렬 W와 편향 b가 모델 학습과정에서 최적화 되는 파라미터로서 학습.

EX)

input: 256x256x3 image
임베딩 벡터의 차원의 수 d: 512


이미지를 16x16 사이즈의 패치로 분할
각 패치는 $16 \times 16 \times 3 = 768$개의 피처
각 패치를 벡터로 변환!

각 패치를 벡터로 변환하기 위해 Trainable Linear Projection 사용.

각 패치 $p_n$는 16x16x3 크기의 피처맵을 가지며 이를 $x_n$으로 표기. (n은 패치의 인덱스)
피처맵 $x_n$는 Trainable Linear Projection을 통해 d-차원의 임베딩 벡터 $e_n$로 변환.
(512)

(d는 임베딩 벡터의 차원의 수)
패치 갯수이기

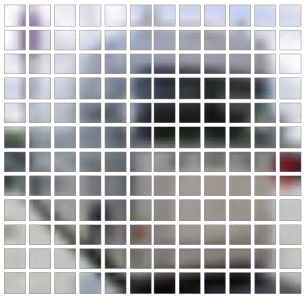
$e_n = Wx_n + b$
$= (d \times \text{patch feature size}) \times (\text{patch feature size} \times 1) + b$

$W = [512 \times 768]$ . $x_n = [768 \times 1]$ . $Wx_n = [512 \times 1]$ . $b = [512 \times 1]$

이를 통해 각 패치들이 벡터로 변환!

# Vision Transformer: Positional Embedding

- Vision Transformer(비전 트랜스포머)에서는 4가지 Positional Embedding(위치 임베딩)을 시도한 후, 최종적으로 가장 좋은 효과를 낸 1D Positional Embedding(1차원 위치 임베딩)을 Vision Transformer(비전 트랜스포머)에 사용했습니다.
- 1D Positional Embedding(1차원 위치 임베딩)은 일반적으로 0부터 시작하여 패치의 순서대로 증가하는 정수값을 할당하는 임베딩하는 방식입니다.
- 최종적으로 Transformer Encoder(트랜스포머 인코더)에 Class Embedding(클래스 임베딩)과 Patch Embedding(패치 임베딩)에 Positional Embedding(위치 임베딩)을 더하여 입력으로 들어오게 됩니다.



# Vision Transformer: Transformer Encoder

- Vision Transformer(비전 트랜스포머)는 Multi-Head Self Attention(멀티 헤드 셀프 어텐션) 블록과 Multi-Layer Perceptron(멀티 레이어 퍼셉트론) 블록으로 구성되어있습니다.
- Multi-Layer Perceptron(멀티 레이어 퍼셉트론)은 일반적으로 2개의 레이어를 가지며, GELU 활성화 함수를 사용하여 비선형성을 도입합니다.
- 각 블록의 앞에는 Layer Norm(레이어 놈)을 적용하여 각 블록의 입력에 대해 정규화 수행하여 학습을 안정화시킵니다.
- 각 블록의 뒤에는 Residual Connection(잔차 연결)을 적용하여 각 블록의 이전 단계의 입력을 더해줌으로써 그래디언트 소실 문제를 완화시킵니다.

# Vision Transformer: Hybrid Architecture

- Convolutional Neural Network(합성곱 신경망)의 지역적인 특징 추출 능력과 Transformer(트랜스포머)의 전체적인 특징 파악 능력을 결합하면, 적은 데이터로 Inductive Bias(귀납적 편향) 문제를 해결할 수 있습니다(더 적은 데이터로 효율적인 학습 가능).
- 접근 방식은 Transformer(트랜스포머)의 Patch Embedding(패치 임베딩) 부분에 Convolutional Neural Network(합성곱 신경망)을 사용하여 피처맵을 생성하고 이를 Transformer(트랜스포머)의 입력으로 사용합니다.
- 장점은 다음과 같습니다.
    - Inductive Bias(귀납적 편향) 감소: Convolutional Neural Network(합성곱 신경망)을 통해 지역적 특징을 추출하면, Transformer(트랜스포머)가 전체적인 관계를 학습하는 데 도움이 됩니다.
    - 효율적인 피처 추출: Convolutional Neural Network(합성곱 신경망)은 이미지의 작은 부분에서 특징을 추출하는 데 능숙하므로, Transformer(트랜스포머)의 Patch Embedding(패치 임베딩) 단계로 활용할 경우 세부 정보를 더 잘 파악할 수 있습니다.
    - 더 나은 학습 속도: Convolutional Neural Network(합성곱 신경망)의 지역적 특성 추출은 Patch Embedding(패치 임베딩)을 직접 사용하는 것보다 효율적일 수 있습니다. 이는 학습 속도를 높이고 연산 비용을 줄이는 데 도움이 됩니다.

# Vision Transformer