An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

- code Reference : http://einops.rocks/pytorch-examples.

Abstract

Transformer architecture는 nlp의 Standard가 될 동안, vision 분야에서는 제한이 많았습니다. 비전에서는 attention이 convolution networks에 결합되어 사용되었거나 convolution network의 특정 components로 대체되어 전박적인 구조에 사용되었습니다. (CBAM 같은 느낌으로 생각하시면 될 거 같습니다)

우리는 CNN에 대한 이러한 reliance가 필요하지 않으며, image patch seqeuence를 transformer에 직접 적용하여 Image classification이 매우 잘 수행될 수 있음을 보여줍니다.

1. Introduction

Transformer와 같은 Self-attention-based architectures는 NLP에게 선택되었습니다. 큰 corpus로부터 학습된 뒤 작은 task-specific dataset으로의 fine tuning 형태가 NLP에서는 굉장히 지배적인 접근법입니다. 트랜스포머의 computational efficiency와 scalability때문에100억개의 매개변수도 넘는 전례없는 크기의 모델을 학습시키는것이 가능해졌고, 모델과 데이터셋이 증가함에 따라 성능이 saturating 될 가능성은 없어보입니다.

그러나 computer vision에서는 convolutional architecture가 여전히 지배적입니다. NLP의 성공에 영감을 받아 CNN과 유사한 아키텍쳐를 Self-attention에 결합하려고 했습니다. 하지만 여전히 Large-scale Image recognition에서는 classic ResNet이 아직 SOTA를 유지하고 있습니다.

NLP에서 Transformer의 scaling 성공에도 영감을 받아 우리는 Image를 Patch 단위로 분할하고 선형 embedding sequence를 적용하기 위해 Patch들을 해당 transformer의 input sequence로 제공합니다. Image Patch는 NLP에서는 Token과 동일한 방식으로 처리됩니다. 우리는 Supervised 방식으로 Image classification에 대한 모델을 훈련시킵니다.

강력한 regularization없이 ImageNet과 같은 중간 크기의 데이터 세트에 대해 training할 때, 이러한 모델은 비슷한 크기의 ResNet보다 낮은 정확도를 냅니다.

이것은 조금 안좋게 보일 수 있습니다. : Transformer는 CNN에 비해 translation equivariance 와 locality와 같은 inductive biases가 부족할 수 있습니다. 그러므로 충분하지 않은 data로 학습을 시킨다면 transformer(vit)는 genralize하지 못할 수 있습니다.

그러나 Transformer가 더 큰 Dataset에서 훈련되면 이야기가 달라집니다. 우리는 **inductive bias**가 large scale의 training trumps로는 학습이 가능하다는 것을 발견했습니다. 우리의 Vit는 큰 모델로 학습을 진행한 뒤 task specific한 task로의 transfer시에 우수한 결과를 얻습니다.

(기존의 NLP와 유사한 형태를 보임을 알 수 있습니다.)

2. Related Work

트랜스포머가 2017년 Vaswani에 의해 machine translation 분야에서 제안되었고, 많은 NLP task에서 sota가 되었습니다. 대규모의 트랜스포머 기반의 모델들은 자주 대규모 corpus에 pre-train된 후에 specific task에 맞게 fine-tuning되엇습니다. BERT는 denoising self-supervised을, GPT는 language modeling을 pre-training task로써 활용했습니다.

이미지에 대한 self-attention의 단순한 adaptation은 각 픽셀이 다른 모든 픽셀들에 attend될 것을 요구합니다. 이는 픽셀 수에 대해 quadratic한 복잡도를 가지고 있으며, 이로인해 현실적인 다양한 input size에 확장될 수 없습니다. 그리하여 트랜스포머를 image processing에 적용시키기 위해서는 몇가지 approximation방법들이 시도되었습니다. (local self-attention, sparse attention, applying it in blocks of varing size 등, 모릅니다 ^^7) 이러한 많은 specialized attention구조들은 컴퓨터 비전 task에서 유망한 결과를 보여주나, 하드웨어(GPU)에서는 효율적으로 구현되기에는 복잡한 엔지니어링이 필요합니다.

3. Method

연구진들은 이전의 NLP 에서 사용중인 Transformer의 구조를 최대한 유사하게 하려고 했습니다.

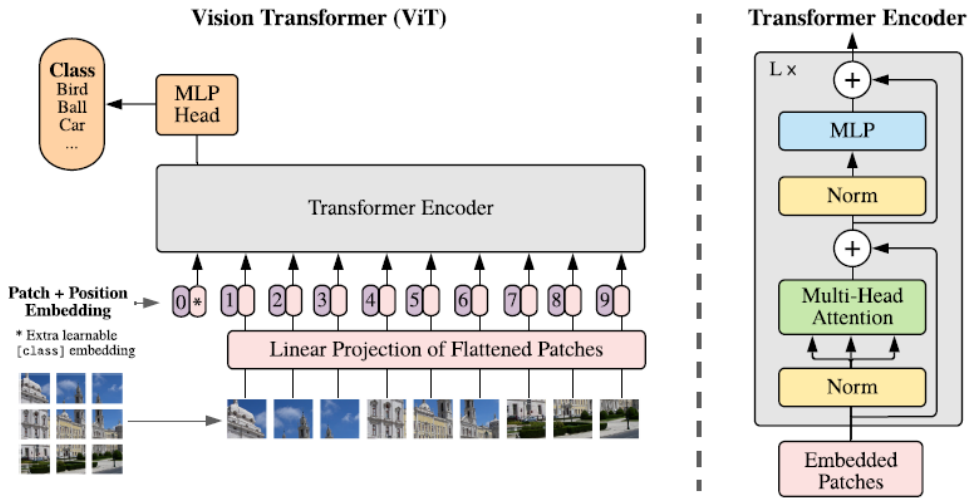


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

모델에 대한 전박전인 overview는 figure 1과 같습니다. 모델 내부에서 2차원 image를 다루기 위해, 본래에 3차원 이미지 $x \in \mathbb{R}^{H \times W \times C}$를 2차원의 패치들 $x_p \in \mathbb{N} \times (\mathbb{P}^2 \cdot c)$로 flatten하였습니다. (H, W)는 원본이미지의 해상도이며, C는 이미지의 채널 수 이고, (P, P)는 패치들의해상도입니다. 이때 $N = HW/P^2$은 패치의 수를 의미하는데, 이것은 Transformer로 들어가는 Sequence의 길이가 됩니다. 트랜스포머는 내부의 모든 layer들에 흐르는 latent vector의 size가 특정값 D로 통일되어 있기 때문에 저자들은 2차원 patch들을 다시 1차원으로 flatten하고 (size : NP^2C) 이를 trainable한 linear projection을 거쳐 D차원 벡터로 매핑시켰습니다.

본 논문에서 이 projection의 결과 벡터를 patch embeddings라고 언급합니다.

BERT 논문에서는 [CLS]라는 토큰을 붙여서 학습 시키는데, 해당 토큰은 Sentence Representation을 가지고 있다고 주장합니다. 이 컨셉을 받아들여 저자들도 임베딩된 패치들의 맨 앞에 하나의 학습 가능한 class token embedding vector를 하나 추가했습니다. 이 [class]임베딩 벡터는 이미지에 대한 1차원 representation vector로써의 역할을 수행합니다. pre-training시와 fine-tuning시 모두, 이 image representation vector위에 classification head가 부착되고, classification head는 사전학습시에 하나의 hidden layer를 가진 MLP로 구현되고, 파인 튜닝시에는 단일 linear layer로 구현됩니다.

위치정보를 유지시키기 위하여 patch embeddings에 trainable한 position embeddings이 더해집니다. 초기에는 이미지를 위해 advanced한 2D-aware position embedding이 전형

적인 1D position embedding과 비교했을 때, 유의미한 성능향상을 가져다주지 않아, 1D positional embedding을 사용했습니다. 이렇게 구성된 최종 임베딩 벡터들의 시퀀스가 encoder에 인풋됩니다.

Base Implement Details

트랜스포머의 인코더는 multiheaded self-attention(MSA) layer들과 MLP 블록들이 교차되어 구성됩니다. Layernormalization이 모든 block의 전에 적용되며, Residual connection이 모든 블록 이후에 붙습니다. MLP는 GELU(Gaussian Error Linear Unit)를 activation으로 사용하는 2개의 layer를 포함하여 구성됩니다.

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

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

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

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

Hybrid Architecture

raw image patches를 대신하여 CNN의 feature map으로부터 input sequence가 형성될 수 있습니다. 이러한 하이브리드 모델에서 식 (1)의 patch embedding projection E는 CNN의 feature map으로부터 추출된 패치들에 적용됩니다. ~~~(내용이 이해가 안가서 추가 안함…?)

▼ 본문

As an alternative to raw image patches, the input sequence can be formed from feature maps of a CNN (LeCun et al., 1989). In this hybrid model, the patch embedding projection E (Eq. 1) is applied to patches extracted from a CNN feature map. As a special case, the patches can have spatial size 1x1, which means that the input sequence is obtained by simply flattening the spatial dimensions of the feature map and projecting to the Transformer dimension. The classification input embedding and position embeddings are added as described above.

papago

원시 이미지 패치의 대안으로, 입력 시퀀스는 CNN의 기능 맵에서 형성될 수 있다 (LeCun et al., 1989). 이 하이브리드 모델에서 패치 임베딩 투영 E(Eq.1)는 CNN 기능 맵에서 추출된 패치에 적용된다. 특별한 경우로서, 패치는 공간 크기 1x1을 가질 수 있으며, 이는 입력 시퀀스가 단순히 형상 맵의 공간 차원을 평탄화하고 트랜스포머 차원으

로 투영함으로써 얻어지는 것을 의미한다. 분류 입력 임베딩 및 위치 임베딩은 전술한 바와 같이 추가된다.

3.2 Fine-Tuning and Higher Resolution

저자들은 대게 Vit를 large dataset들에 pre-train하고 더 작은 규모의 downstream task 데이터셋에 fine-tune했습니다. 이 과정에서 pre-trained prediction head는 제거되고 zero-initialized $D \times K$의 feed forward layer가 부착되었습니다. (K는 downstram classification task의 class 개수) 최근 연구들에서 pre-training시보다 더 높은 해상도로 fine-tune하는 것이 종종 benefical하다는 사실이 밝혀졌습니다. 본 연구에서도 fine-tuning 시 사전 훈련 데이터보다 higher resolution에서 진행되었으며, 이러한 고해상도 image를 전달할 때 patch size($P \times P$)는 유지하고, 트랜스포머 유효 input 시퀀스 길이인 N을 늘리는 방식으로 조절하였습니다. ViT는 어떠한 길이의 시퀀스도 처리할 수 있으나, 이 경우 pre-trained position embeddings는 의미를 잃게 됩니다. 이를 극복하고자 사전 학습된 위치 임베딩에 그것의 원본이지에서의 위치를 기준으로 하는 2D interpolation이 적용되었습니다. 위와 같이 해상도 조정 및 patch extraction이 image의 2차원 구조에 대한 inductive bias가 ViT에 수동적으로 주입되는 유일한 과정입니다.

4.5 Inspecting Vision Transformer

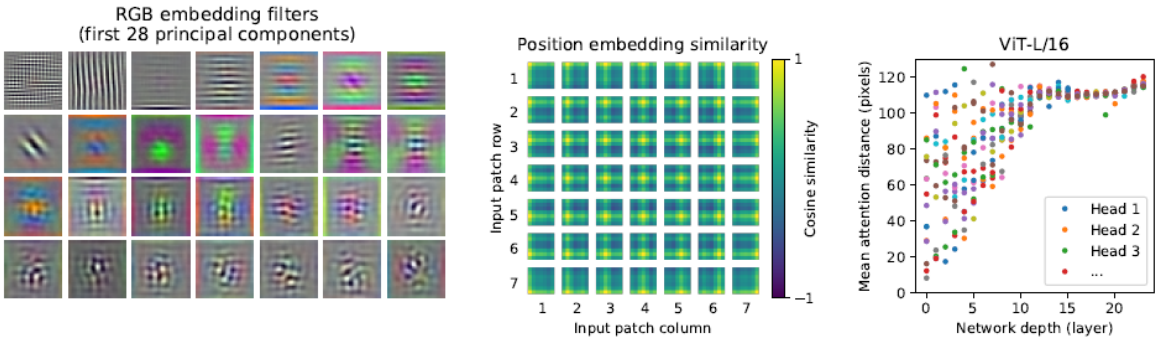


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.6 for details.

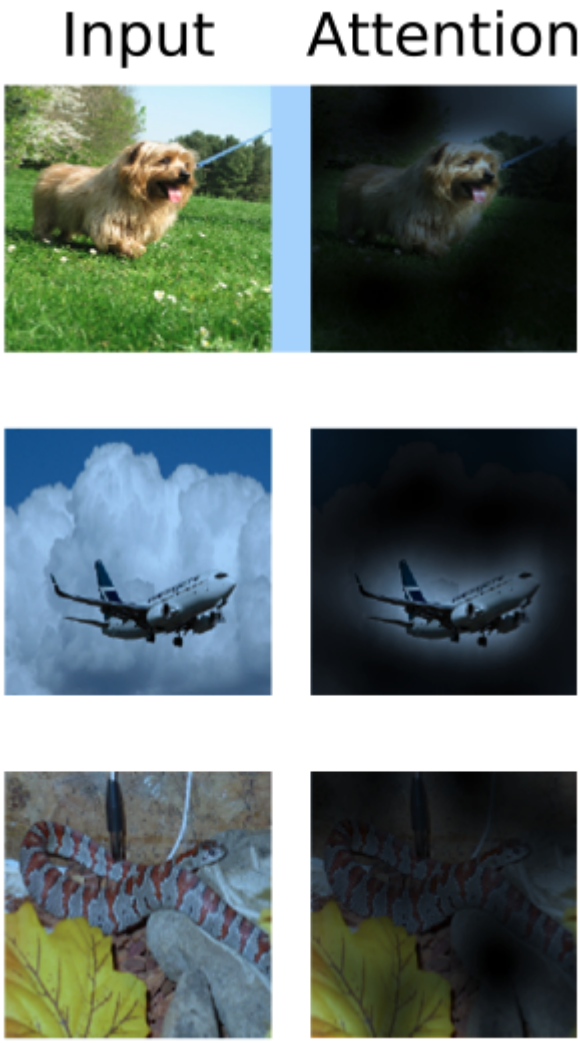
비전 트랜스포머가 이미지를 어떻게 처리하는지 이해하기 위해 내부의 representation들을 분석하였습니다.

ViT의 첫번째 층은 flatten된 patch들을 저차원 공간으로 linearly projection합니다. Figure 7의 왼쪽 사진이 해당 층에서 학습된 임베딩 필터들의 상위 주성분들을 보여주고 있습니다.

저러한 성분들은 각 패치 내의 미세 구조의 저차원적 표현을 위한 그럴듯한 기저함수를 닮았다고 볼 수 있다고 합니다.

Projection이후에는 학습된 positional embedding이 patch representation에 추가됩니다. Figure 7의 가운데 그림은 모델이 포지션 임베딩의 유사도 내에서 이미지 내부의 거리개념은 인코딩하는 방법을 배운다는 것을 보여줍니다. 즉, 가까운 패치들은 유사한 포지션 임베딩을 가지며 row-column 구조(같은 행/열에 있는 patch는 유사한 임베딩을 갖는다)는 것 또한 나타냅니다. 앞서 언급한 2d, 1d dimensional position embedding의 차이가 없다는 것도 보여줍니다.

Self-attention은 Vit가 이미지 전체의 정보를 통합하여 사용할 수 있도록 합니다. 논문 저자들은 네트워크가 이 광활한 수용력을 얼마나 이용하지는 그 정도를 조사해보려고 했고, 구체적으로 정보가 attention weights에 의해 통합되는 image space 내의 평균 거리를 계산하였습니다. 이 “Attention distance”는 CNN에서의 receptive field size와 유사한 개념이라고 보면 됩니다. 그림을 보면 층이 깊어질수록 attention distance가 증가하고 있고, 최하위층 레이어에서도 몇몇 attention head가 이미지 대부분에 attend하고 있는 것을 확인할 수 있는데, 이는 정보를 global하게 integrate할 수 있는 능력을 실제로 모델이 사용하고 있음을 보여줍니다.



질문

▼ 여기에서 inductive bias란?

Inductive Bias