이미지에 대한 비전 트랜스포머(ViT) 기반 딥 클러스터링

신혜수[1], 유사라[1], 이기용[1]
[1]숙명여자대학교 컴퓨터과학과
{seawater, rrrr4ra, kiyonglee}@sookmyung.ac.kr

Deep Clustering Based on Vision Transformer(ViT) for Images

Hyesoo Shin[1], Sara Yu[1], Ki Yong Lee[1]
[1]Dept. of Computer Science, Sookmyung Women's University

요 약

본 논문에서는 어텐션(Attention) 메커니즘을 이미지 처리에 적용한 연구가 진행되면서 등장한 비전 트랜스포머 (Vision Transformer, ViT)의 한계를 극복하기 위해 ViT 기반의 딥 클러스터링(Deep Clustering) 기법을 제안한다. ViT는 완전히 트랜스포머(Transformer)만을 사용하여 입력 이미지의 패치(patch)들을 벡터로 변환하여 학습하는 모델로, 합성곱 신경망(Convolutional Neural Network, CNN)을 사용하지 않으므로 입력 이미지의 크기에 대한 제한이 없으며 높은 성능을 보인다. 그러나 작은 데이터셋에서는 학습이 어렵다는 단점이 있다. 제안하는 딥 클러스터링 기법은 처음에는 입력 이미지를 임베딩 모델에 통과시켜 임베딩 벡터를 추출하여 클러스터링을 수행한 뒤, 클러스터링 결과를 임베딩 벡터에 반영하도록 업데이트하여 클러스터링을 개선하고, 이를 반복하는 방식이다. 이를 통해 ViT 모델의 일반적인 패턴 파악 능력을 개선하고 더욱 정확한 클러스터링 결과를 얻을 수 있다는 것을 실험을 통해 확인하였다.

1. 서론

트랜스포머(Transformer)[1]에서 입력 데이터의 중요한 특정 영역에 더 집중하는 어텐션(Attention) 메커니즘이 이미지 분야에서도 활용되기 시작하면서 최근에는 트랜스포머가 기존의 이미지 처리 모델을 대체하고 있다[2][3].

(그림 1) 이미지 처리에서의 어텐션 메커니즘[2].

이전까지 대부분의 이미지 처리 연구에서는 합성곱 신경망(Convolutional Neural Network, CNN)[4]이 사용되었는데, CNN은 이미지의 모든 부분에 동일한 가중치로 처리하는 단점이 있어 이를 해결할 필요성이 제기되었다. 이에 따라 기존의 CNN 모델에 어텐션을 추가하여 입력 이미지 내 중요한 영역에 더 집중하는 어텐션 기반 CNN(Attention-based CNN, ABCNN)[5]이 제안되었지만 여전히 고정된 크기의 입력 이미지를 필요로 하며, 글로벌한 정보는 파악하지 못하는 CNN의 한계점은 여전히 가지고 있다. 이 한계점을 해결하기 위해 제안된 비전 트랜스포머(Vision Transformer, ViT)[2]는 CNN을 사용하지 않고 트랜스포머만을 사용하여 입력 이미지의 패치(patch)들을 벡터로 변환하고, 세분화된 학습을 가능하게 한다. 그러나 ViT는 작은 데이터셋에서는 데이터의 일반적인 패턴을 잘 파악하기 어렵다는 단점이 있다.

따라서 제안 방법은 ViT의 한계를 극복하기 위해 ViT 기반의 딥 클러스터링(Deep Clustering) 기법을 제안하고자 한다. 논문에서 제안하는 딥 클러스터링은 초기 클러스터링 결과를 다시 모델에 피드백하여 임베딩(Embedding) 벡터를 업데이트하는 과정을 반복하는 기법으로, 클러스터링 결과와 임베딩 벡터의 상호작용을 통해 모델을 개선하는 특징을 가진다. 임베딩 벡터를 업데이트하는 과정에서 임베딩 공간에 더욱 의미 있는 구조를 만들고, 고전적인 클러스터링 기법보다 정확한 클러스터링 결과를 얻을 수 있으므로 여러 번 임베딩을 개선한 ViT 모델은 일반적인 패턴을 파악하기 어려운 ViT의 한계점을 개선할 수 있다.

본 논문에서 제안하는 방법은 다음과 같다. 먼저 ViT를 통해 이미지를 임베딩하여 임베딩 벡터를 얻고 클러스터링한다. 그런 다음, 클러스터링 결과를 임베딩 벡터에 반영하기 위해 새로운 손실 함수를 정의한 ViT 모델을 생성한다. 그리고 클러스터링 결과를 임베딩 벡터에 반영하여 업데이트하고, 다시 클러스터링하는 과정을 반복한다. 손실함수가 임계값 이하로 감소하면 훈련을 마치고 최종 ViT 모델을 통해 임베딩한 뒤 클러스터링한다. 이렇게 딥 클러스터링을 거쳐 얻은 최종 임베딩 벡터는 ViT의 어텐션 메커니즘을 통해 이미지에서 중요한 영역에 중점을 두고 학습했을 뿐만 아니라 임베딩을 여러 번 개선하였기 때문에 더 좋은 클러스터링 결과를 얻을 수 있다.

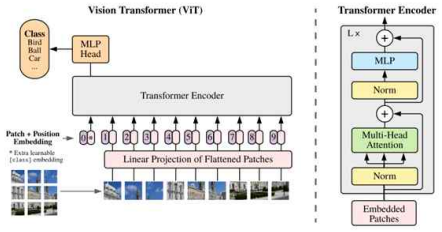
본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 기존에 진행된 관련 연구에 대해 살펴본다. 3장에서는 본 논문에서 제안하는 ViT 기반 딥 클러스터링 방식에 대하여 설명한다. 4장에서는 실 데이터를 사용한 실험 결과를 살펴보고, 마지막으로 5장에서는 결론을 도출한다.

2. 관련 연구

ViT는 이미지 처리 분야에서 기존에 연구가 진행된 바 있다. 본 장에서는 연구에 대해 살펴본다.

2.1 비전 트랜스포머 (Vision Transformer, ViT)

ViT[2]는 트랜스포머를 기반으로 한 비전 모델로, ViT는 ABCNN과 달리 이미지를 1차원의 벡터 형태의 시퀀스(sequence)로 변환하고 패치 단위로 나눈 후에 이를 트랜스포머의 인코더(Encoder)에 입력하여 처리한다. 따라서 입력 이미지의 크기에 관계없이 항상 고정된 크기의 벡터로 표현할 수 있으므로 이미지의 긴 범위 정보를 고려할 수 있다. 따라서 ViT는 트랜스포머의 어텐션 메커니즘을 사용하여 인코더 레이어에서 글로벌한 정보를 유지하고, 이를 통해 이미지를 분류한다.



(그림 2) ViT의 전체 흐름도[2]와 트랜스포머의 인코더 구조[1].

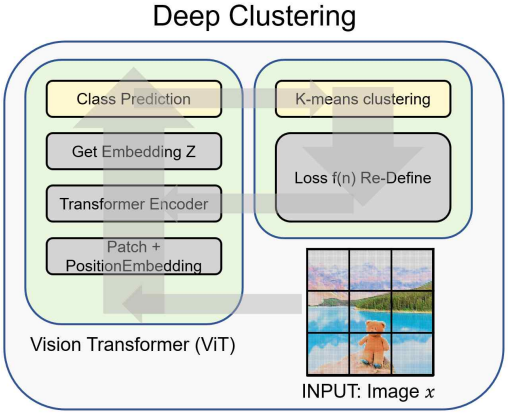
트랜스포머의 인코더는 다수의 인코더 레이어로 이루어져 있으며, 인코더 레이어를 거치면서 입력 이미지의 패치 임베딩은 점점 더 추상화된 정보로 변환되며, 최종적으로는 하나의 벡터로 변환된다.

2.3. 딥 클러스터링 (Deep Clustering)

딥 클러스터링은 그래프 신경망(Graph Nueural Network, GNN)과 같은 모델을 사용하여 진행된 연구[6]가 대부분이었으나 최근 딥 클러스터링의 임베딩 모델을 트랜스포머로 변환하려는 시도가 진행되고 있다. 하지만 ViT를 이용한 딥 클러스터링은 아직까지 충분히 연구되지 않았다. 딥 클러스터링에 적용할 모델로써 ViT는 이미지의 전체 정보를 동등하게 고려할 수 있기 때문에 이미지의 크기가 크거나 특히 이미지 안에 포함된 오브젝트들이 서로 밀접하게 관련되어 있는 경우에 적합하다. 따라서 제안 방법은 ViT 모델을 선택하여 딥 클러스터링을 진행하고자 한다.

3. 제안 방법

본 장에서는 제안 방법의 이미지에 대한 ViT 기반 딥 클러스터링 방식에 대해 자세히 설명한다. 모델의 전체 흐름도는 그림 [3]와 같이 요약할 수 있다. 제안 방법은 (1)최초의 ViT 학습 및 초기 클러스터링 단계, (2)손실함수 정의 및 ViT 재학습 반복 단계, (3)최종 클러스터링 단계로 구분된다.



(그림 3) 전체 흐름도.

3.1 최초의 ViT 학습 및 초기 클러스터링 단계

원본 이미지 $x \in R^{H \times W \times C}$가 주어졌다고 가정하자. ViT는 2차원 배열의 x가 들어왔을 때, $P \times P$ 크기의 패치 $N = H \times W / P^2$개로 분할하여 시퀀스 $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$를 얻는다. 여기서 H, W는 원본 이미지의 해상도를 의미하며 C는 채널 수, P는 이미지 패치의 해상도다. N은 트랜스포머에 입력이 되는 시퀀스 길이로 초매개변수다. 이렇게 얻은 시퀀스 x_p는 식 (1)을 통해 x_p의 각 패치들을 1차원으로 flatten하고 trainable linear projection을 적용해 D차원에 매핑(mapping)하여 패치 임베딩을 생성한다.

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \cdots; x_p^N E] + E_{pos} \quad (1)$$

여기서 $x_{class} = z_0^0$는 Class Embedding이며, 모든 패치의 임베딩에 항상 추가되는 특별한 임베딩으로 전체 이미지를 대표하는 벡터다. x_p^N는 패치로 나누어진 각각의 이미지 시퀀스, E_{pos}는 시퀀스의 순서를 나타내는 1차원의 Positional Encoding이며 위치 정보를 유지하기 위해 패치 임베딩에 합산된다. 이렇게 생성된 z_0는 트랜스포머 인코더의 입력으로 사용되며, 인코더는 $\ell = 1 \cdots L$개의 레이어로 이루어져 있으며 식 (2-3)를 통해 계산된다.

$$z_\ell^{'} = MSA(LN(z_{\ell-1})) + z_{\ell-1} \quad (2)$$

$$z_\ell = MLP(LN(z_\ell^{'})) + z_\ell^{'} \quad (3)$$

여기서 Layer Normalization(LN)는 각 레이어의 출력값을 정규화하는데, 이를 통해 모델의 학습이 안정화되면서 레이어 간 상호작용을 원활하게 할 수 있다. Multi-head Self-Attention(MSA)는 입력 시퀀스 내의 각 요소들끼리 상호작용을 하는 Self-Attention 메커니즘을 여러 개의 head로 나누어 병렬로 수행하고 결과를 결합한다. Multi-Layer Perceptron(MLP)는 입력 벡터를 고차원 공간으로 매핑하는 역할을 하며, 이를 통해 입력 시퀀스의 특징을 높은 수준의 추상화된 표현으로 변환하여 다시 다음 MSA 레이어의 입력으로 사용된다.

이러한 과정으로 인코더 레이어를 거치면서 입력 이미지의 패치 임베딩은 점점 더 추상화된 정보로 변환되며, 각 패치와 전체 이미지에 대한 임베딩 정보를 조합하고 업데이트하여 이미지를 잘 분류할 수 있는 모델 파라미터를 학습한다.

이렇게 학습된 임베딩 벡터는 트랜스포머 인코더의 마지막 레이어 L에서 이미지 표현 벡터(image representation) z_0^L를 얻는다. ViT는 최종적으로 식 (4)의 연산을 통해 이미지 표현 벡터에 대한 클래스 예측 값을 얻고, 이를 사용하여 최종 분류를 수행한다.

$$\hat{y} = LN(z_L^0) \quad (4)$$

$\hat{y}$은 ViT의 최종 출력을 나타낸다. 분류 문제에서 ViT의 최종 출력은 각 이미지가 속할 확률이 가장 높은 클래스(class)를 얻게 되며, 따라서 ViT를 학습할 때는 모델이 예측한 클래스의 확률 분포와 실제 클래스 레이블을 나타내는 확률 분포 사이의 차이를 손실함수로 사용한다. 따라서 ViT의 손실 함수는 일반적으로 다음과 같이 나타낼 수 있다.

$$L_{ViT} = \sum_{i=1}^{N} Loss(y_i, \hat{y_i}) = -\sum_{i=1}^{N} p(\hat{y_i}) \log(q(\hat{y_i})). \quad (5)$$

본 단계에서는 원본 이미지 x와 목표 클래스 y가 주어졌을 때, 식 (5)의 손실함수 L_{ViT}를 최소화하도록 ViT를 학습시킨다.

이렇게 최초의 ViT가 학습되면 제안 방법은 해당 ViT을 사용하여 각 이미지에 대한 임베딩 벡터를 얻는다. 이렇게 얻어진 임베딩 벡터 z_1, z_2, ..., z_N에 클러스터링 알고리즘을 적용하여 초기 클러스터링 결과를 얻는다.

3.2 손실함수 정의 및 ViT 재학습 반복 단계

3.1절에서 임베딩 벡터 z_1, z_2, ..., z_N를 클러스터링한 결과 C_1, C_2, ..., C_k의 클러스터들이 생성되었고 각각의 중심점을 μ_1, μ_2, ..., μ_k이라고 정의한다. 이때 클러스터링의 손실함수는 다음과 같이 나타낼 수 있다.

$$L_{clustering} = \sum_{i=1}^{k} \sum_{z \in C_i} \| z - \mu_i \|^2 \quad (6)$$

$$= \sum_{i=1}^{N} \| z_i - centroid(z_i) \|^2 . \quad (7)$$

여기서 $centroid(z_i)$는 z_i가 속한 클러스터의 중심점을 나타낸다. 이를 기반으로 클러스터링의 품질을 증가시키기 위해 각 임베딩 벡터 z_i와 중심점 $centroid(z_i)$의 거리를 계산하여 새로운 ViT의 손실함수로써 최소화하고자 한다.

$$L = L_{ViT} + \gamma L_{clustering} \quad (8)$$

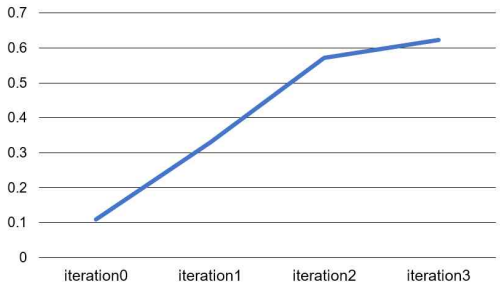
식 (8)에서 γ는 클러스터링 손실의 가중치를 나타내는 초매개변수다. 식 (8)을 사용하여 재학습이 완료된 ViT의 손실함수 L의 값을 $l_{current}$라 하자. 이후 제안 방법은 3.2절에서 지금까지 설명한 방법을 반복하여 다시 각 이미지의 임베딩 벡터를 얻고, 이들을 클러스터링으로 클러스터링한 후, 그 결과를 사용하여 식 (8)으로 ViT을 재학습시킨다. 이렇게 새로 얻은 ViT의 손실함수 L의 값을 l_{new}라 하자. 제안 방법은 주어진 임계값 ε에 대해 $l_{current} - l_{new} < \varepsilon$이 될 때까지 3.2절의 과정을 반복한다.

3.3 최종 클러스터링 단계

3.2절에서 설명한 과정을 통해 최종 ViT가 얻어지면 제안 방법은 식 (2-3)을 사용하여 해당 ViT로 이미지들을 임베딩하고, 여기서 얻어진 임베딩 벡터는 클러스터링 알고리즘을 거쳐 최종 클러스터링 결과를 얻는다. 여기서 얻어진 임베딩 벡터는 어텐션을 통해 이미지의 중요한 영역을 크게 반영하는 한편 이미지 벡터와 중심점 간의 거리까지 반영한다. 따라서 보다 품질 높은 클러스터링 결과를 얻을 수 있다.

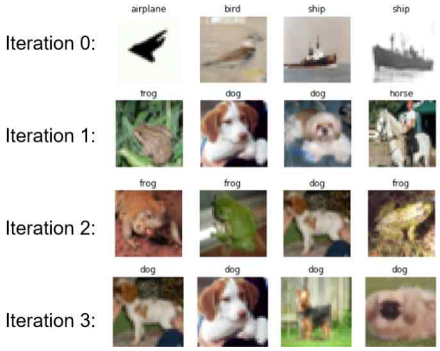
4. 실험 결과

4장에서는 제안 방법이 이미지 클러스터링에서 개선된 실험 결과를 보인다. 실험을 위해서 CIFAR-10[7] 데이터 셋을 사용하였다. CIFAR-10은 32x32 픽셀의 60,000개의 컬러 이미지 데이터셋으로, 이 데이터셋은 10개의 클래스로 구성되어 있으며, 각 클래스마다 6,000개의 이미지가 있다. 각 클래스는 비행기, 자동차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭으로 구성된다. 실험 결과 성능 지표는 클러스터링 성능 평가 지표 중 대표적인 NMI(Normalized Mutual Information)를 사용하였다. NMI는 -1부터 1 사이의 값을 가지며, 1에 가까울수록 클러스터링 결과가 우수함을 나타낸다.



(그림 4) 딥 클러스터링 반복에 따른 NMI 값 변화.

그림 [4]는 재학습 반복에 따른 NMI 값의 변화를 나타낸다. 그림 [4]에서 볼 수 있듯이 제안 방법은 NMI를 계속 증가시키고 있음을 확인할 수 있다.



(그림 5) 클래스가 [dog]로 예측된 클러스터 이미지 시각화.

그림 [5]은 클래스가 '개'일 것으로 예측된 동일한 클러스터 내 이미지 시각화를 보여 준다. 각 그림은 원본 이미지에 대한 임베딩한 벡터를 다시 이미지 형태로 복원하여 시각화하였다. 그림에서 반복을 하지 않았을 때에는 잘못 분류된 이미지가 여러 개 속해 있으나 반복을 거친 후에는 이미지들이 올바르게 분류됨을 확인할 수 있다.

5. 결론

본 논문에서는 ViT의 한계를 극복하기 위해 ViT 기반의 딥 클러스터링 기법을 제안하였다. ViT는 트랜스포머만을 사용하여 입력 이미지의 패치들을 벡터로 변환하여 학습하는 모델로, 성능은 우수하지만 작은 데이터셋에서는 학습이 어렵다는 단점이 있다. 제안하는 모델은 처음에는 입력 이미지를 임베딩 모델에 통과시켜 임베딩 벡터를 추출하여 클러스터링을 수행한 뒤, 클러스터링 결과를 임베딩 벡터에 반영하도록 업데이트하여 클러스터링을 개선하고, 이를 반복하는 방식이다. 실제 데이터를 사용한 실험에서 ViT 모델의 일반적인 패턴 파악 능력을 개선하고 더욱 정확한 클러스터링 결과를 얻을 수 있다는 것을 확인하였다.

사사문구

이 성과는 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2021R1A2C1012543).

참고문헌

[1] V. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is All You Need," in Proc. of the Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5998-6008.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in Proc. of the International Conference on Learning Representations (ICLR), 2021.

[3] Z. Liu, L. Yuan, S. Shen, Z. Huang, A. B. Bulatov, G. Zhang, C. Y. Tang, and X. Wang, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," in Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[4] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel, "Handwritten digit recognition with a back-propagation network," in Proc. of the Advances in Neural Information Processing Systems (NeurIPS), 1989, pp. 396-404.

[5] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "CBAM: Convolutional Block Attention Module," in Proc. of the European Conference on Computer Vision (ECCV), 2018, pp. 3-19.

[6] H. Shin, K. Y. Lee, "A Deep k-Means Clustering Method for Nodes in a Graph", Proc. of the KIISE Korea Software Congress 2022, pp.125-127, 2022. (in Korean)

[7] CIFAR-10, https://paperswithcode.com/dataset/cifar-10.