ToMato: Token Merging을 이용한 Vision Transformer 가속화

권수영*, 권민서*, 김효진*, 심재형
이화여자대학교 엘텍공과대학 컴퓨터공학과 소속
e-mail : {sooyoung7896, woojoorom, cosmos07}@ewhain.net, jh.sim@ewha.ac.kr
*These authors contributed equally to the work.

ToMato: Accelerating ViT via Token Merging

Sooyoung Kwon*, Minseo Kwon*, Hyojin Kim* and Jaehyeong Sim
Ewha Womans University

Abstract

ViT(Vision Transformer) shows outstanding performance in various vision tasks by splitting images into patches and passing them through transformer blocks. However, the large model size and computational cost of ViT result in high inference latency and hindered acceleration. To accelerate ViT efficiently, we introduce ***ToMato***(**T**oken **M**erging **at** **O**nce), a simple framework that recursively merges tokens by comparing similarity to adjacent tokens at the first transformer block. Applying the ToMato to DeiT-base model, we find that this reduces latency by 22.19% while maintaining high Top-1 accuracy of 80.14%. Our codes are available at https://github.com/Transformer04/ToMato.

I. 서론

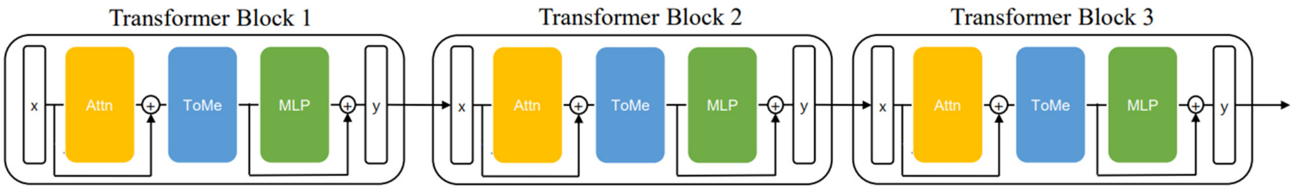
문장의 구조와 의미적 관계를 파악하고 생성하는데 강력한 성능을 보이는 transformer 모델[1] 이 등장하면서, 자연어 처리 분야뿐만 아니라 컴퓨터 비전, 음성 인식 분야 등 다양한 분야에서 transformer 기반의 모델들이 괄목할 만한 성능을 보이고 있다. 특히 그 중 ViT(Vision Transformer) 모델은 이미지 분류 작업에서 고전적인 CNN 모델들과 비교하여 우수한 성능을 보였다[2].

하지만 ViT 모델의 뛰어난 정확도에도 불구하고, ViT 는 거대한 모델 크기로 인한 추론 시간의 지연이라는 문제점이 있다. 이를 해결하기 위해 이미지 내의 불필요한 토큰들을 가지치기해서 처리해야 할 연산량을 줄이는 token pruning 기법을 ViT 에 적용하는 연구들이 제안되어 왔지만, 토큰을 제거함에 따라 정보 손실이 일어나기 때문에 정확도가 떨어지고, 모델의 성능을 높이기 위해 fine-tuning 을 해야 했다.

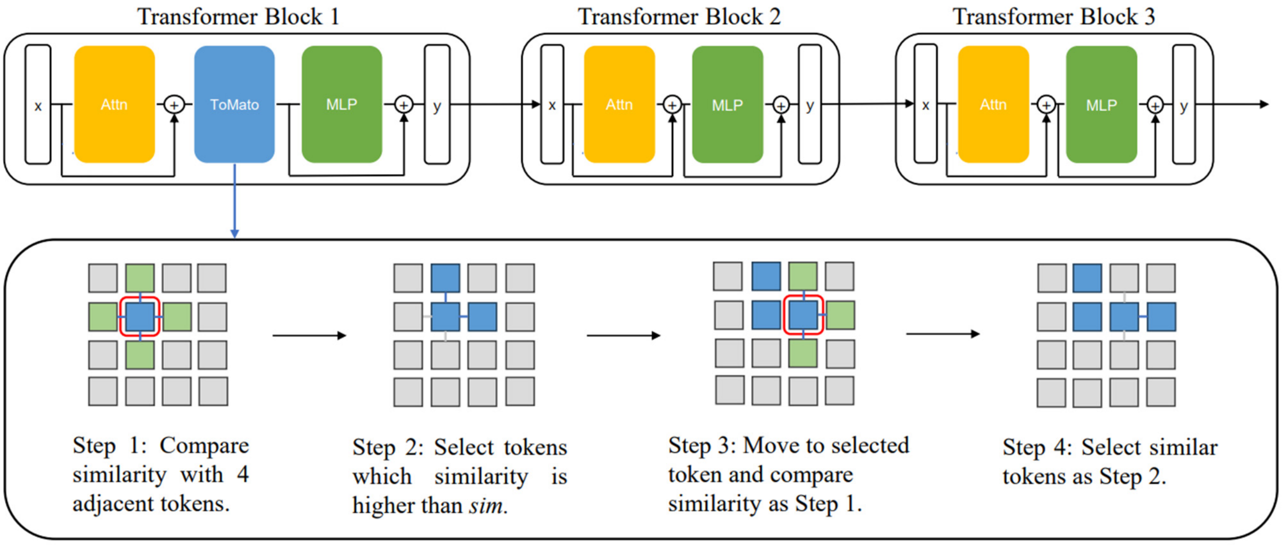
Token pruning 기법의 문제점을 보완하기 위해 ToMe 에서는 token merging 을 제안한다[3]. Token merging 은 유사한 토큰들을 점진적으로 결합하여 총 토큰의 수를 줄이고 정보 손실을 방지한다. 이 접근 방법은 별도의 사전학습을 요구하지 않고 추론 속도를 가속화하지만, 여러 개의 transformer block 에 걸친 점진적 merging 이 요구된다.

본 연구에서는 ToMato(Token Merging at Once)라는 프레임워크를 제안한다. 이는 가장 초기의 transformer block 에서 한 번의 token merging 을 수행하여 이전 연구들과 비교했을 때 시간적 이점을 얻을 수 있다. 뿐만 아니라, 가까운 픽셀들끼리 의미적 유사성을 가질 확률이 높은 이미지 데이터의 특징을 활용해 인접한 토큰들끼리 유사도를 비교하여 추가적인 모델 학습 없이도 기존의 DeiT[4] 와 유사한 성능을 얻을 수 있다.

ToMato 를 DeiT-base 모델에 적용하여 ImageNet-Mini 데이터셋으로 실험한 결과 추론 시간은 약 22.19%, Top-1 정확도는 약 0.82%


Transformer Block 1
Transformer Block 2
Transformer Block 3
x
Attn
ToMe
MLP
y

(a) ToMe


Transformer Block 1
Transformer Block 2
Transformer Block 3
x
Attn
ToMato
MLP
y
Step 1: Compare similarity with 4 adjacent tokens.
Step 2: Select tokens which similarity is higher than *sim*.
Step 3: Move to selected token and compare similarity as Step 1.
Step 4: Select similar tokens as Step 2.

(b) Token Merging at Once

그림 1. (a) **ToMe**는 각 transformer block에 대해 attention과 MLP 사이에서 토큰 병합을 진행한다. (b) **ToMato**는 첫 번째 transformer block에서만 토큰 병합을 진행한다. 본 연구의 병합 알고리즘은 인접한 4개의 토큰과 유사도를 비교하는 과정을 재귀적으로 반복한다.

감소하였다. 또한, ToMe 와 비교했을 때에는 추론 시간은 약 6.06%, Top-1 정확도는 0.59% 감소하였다.

II. 본론

2.1 ToMato (Token Merging at Once)

본 연구에서는 딥러닝 모델 추론 전 한 번에 token merging 을 진행하는 ToMato 라는 프레임워크를 개발하였다. 이는 여러 유사한 토큰들을 하나로 한 번에 merging 하는 기법이다. 그림 1 (a)에서 볼 수 있듯이 매 transformer block 마다 token merging 을 진행하는 ToMe 와 달리, ToMato 는 (b)에서처럼 첫번째 block에서 한 번만 token merging을 진행하여 merging 에 필요한 전체적인 소요 시간을 줄였다. 사용자가 similarity threshold (sim)를 직접 설정하여 토큰이 병합되는 정도를 조절할 수 있으며, 이 값이 작을수록 더 많은 토큰들이 병합된다.

Bipartite soft matching 을 사용하는 ToMe와 달리, 본 연구에서는 한 토큰을 기준으로 인접한 4 개의 토큰들과의 유사도만을 비교한다. 이 방법은 이미지 상에서 인접한 패치들끼리 비슷한 배경 또는 특징을 가질 확률이 높다는 점을 활용한 것이다. 그림 2 를 통해 균일한 색상을 가진 인접한 배경이 한 번에 병합되는 것을 확인할 수 있다. 이를 바탕으로 토큰끼리의 유사도 비교 횟수를 크게 줄이기 위해 상하좌우로 인접한 토큰들끼리만 비교하도록 하였다.

토큰들 간의 유사도를 측정할 때는 attention score 에 softmax 함수를 적용한 attention distribution 을 사용한다. sim 을 여러 값으로 변경하며 실험을 진행하였고, sim 값이 작아질수록 더 많은 양의 토큰들이 병합되는 것을 확인하였다.

```
Input: similarity threshold sim, attention score similarity attn
for i=0 to numTokens -1 do
  if i not in visited then
    visit_recursive()
    merge tokens
  end if
end for
```

알고리즘 1. ToMato 는 주어진 이미지에 존재하는 모든 토큰들에 대해 인접한 토큰을 방문한 뒤에 유사한 토큰을 병합하는 과정을 반복한다.

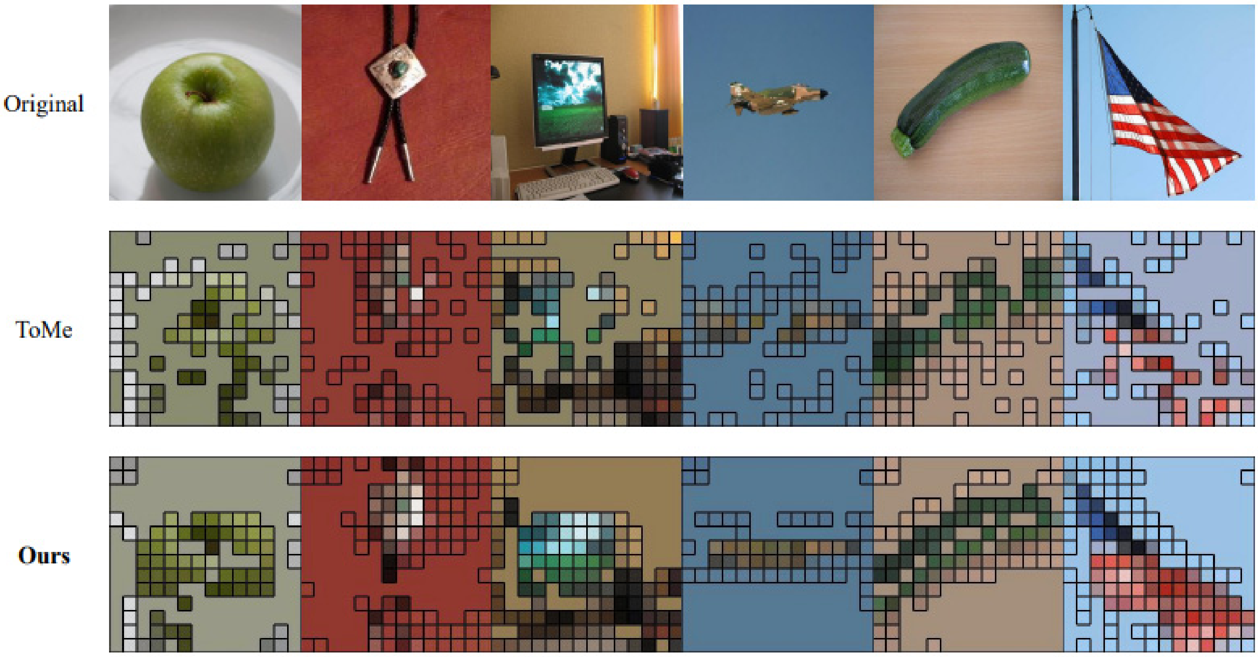


그림 2. ToMato를 사용하여 ImageNet-mini 이미지 데이터들의 패치들이 병합된 모습. 인접한 토큰들과의 유사도 비교를 통해 ToMato는 ToMe보다 의미적으로 유사한 패치들을 효과적으로 병합할 수 있다.

```
function: visit_recursive
Input: similarity threshold sim, attention score similarity attn, token index idx, visited tokens visited
if idx not in visited then
  visited.add(idx)
  if each 4 adjacent tokens(idx±1,idx±14) attn >= sim then
    visit_recursive()
  end if
end if
```

알고리즘 2. 알고리즘 1 에서 사용된 visit_recursive 함수는, 주어진 하나의 토큰에 대해 상하좌우로 인접한 4 개의 토큰과의 유사도를 비교하는 과정을 재귀적으로 반복한다.

구체적인 알고리즘은 알고리즘 1과 알고리즘 2에 설명되어 있다. 첫번째 decoder에서 multi-head attention이 진행되고 난 직후에 token merging이 시작된다. 첫번째 토큰부터 시작하여, 인접한 4개의 토큰들에 대하여 병합을 진행할 토큰을 판단하고, 병합을 진행할 토큰들을 기준으로 다시 인접한 토큰 4개에 대해 병합을 진행하는 과정을 똑같이 반복한다. 비슷한 토큰들을 재귀적으로 찾아낸 뒤, 더 이상 인접한 토큰들 중 유사한 토큰을 찾을 수 없을 때, 찾아낸 모든 비슷한 토큰들을 하나로 병합한다. 위 과정을 첫번째 토큰에서 시작하여 마지막 토큰까지 진행한다.

III. 구현

3.1 실험 환경

본 연구의 실험은 NVIDIA Jetson Nano Developer Kit 에서 진행되었다. 데이터셋은 ImageNet-Mini 를 사용했고 비교군으로 DeiT-base 와 ToMe 모델을 사용했다.

ToMato 모델은 별도의 학습을 거치지 않았으며, 모델들의 성능은 정확도와 추론 시간을 기준으로 하여 비교하였다. 추론 시간의 측정을 위해 batch 크기를 1 로 설정하였고, 실제 시간은 NVIDIA 의 Nsight Systems 를 이용하여 측정하였으며, 총 10 번의 추론을 수행하여 나온 평균 시간을 최종 결과로 보고하였다.

3.2 실험 결과

ToMato 를 DeiT-base 모델과 ToMe 를 이용하여 성능과 정확도를 비교한 결과는 표 1 에서 확인할 수 있다. Sim 값을 10^{-7} 로 설정한 ToMato 모델에서 총 토큰들의 수가 최종적으로 약 8.16%에서 20.92%로 감소했다. ToMe 모델은 DeiT-base 위에서 실험한 모델을 사용하였고, 각 transformer block 에서 병합되는 토큰들의 수를 나타내는 r 값을 2 로

설정했을 때 최종적으로 약 15%의 토큰들이 병합되었다.

ToMato 는 다른 두 모델과 비교했을 때 각각 22.19%, 6.06%의 추론 시간 감소를 보인다. Top-1 정확도는 DeiT-base 와 ToMe 에 비해 0.82%, 0.59% 감소하였으며 Top-5 정확도는 0.40%, 0.31% 감소하였다. 이를 통해 ToMato 가 성능의 저하를 줄이면서 DeiT 모델의 추론 속도를 개선시킨다는 결과를 확인할 수 있다.

표 1. ToMato의 정확도와 추론 시간을 측정하여 DeiT-base, ToMe 모델과 비교한 결과

Model	Top-1 acc (%)	Top-5 acc (%)	Latency (s)
DeiT-B	80.96	95.40	13.2132
ToMe-B	80.73	95.31	10.9445
OURS-B	**80.14**	**95.00**	**10.2804**

그리고 ToMato 모델에서 sim 값을 여러 가지로 달리 하여 나타나는 정확도와 추론 시간의 상관관계를 확인하였다. 그림 3에서 r 값에 따른 ToMe와 sim 값에 따른 ToMato의 Top-1 정확도와 추론 시간을 확인할 수 있다. ToMe에서는 추론 시간에 따른 정확도의 변화폭이 큰 반면, ToMato는 추론 시간이 줄어도 정확도의 변화가 크지 않은 특성을 보이고 있다.

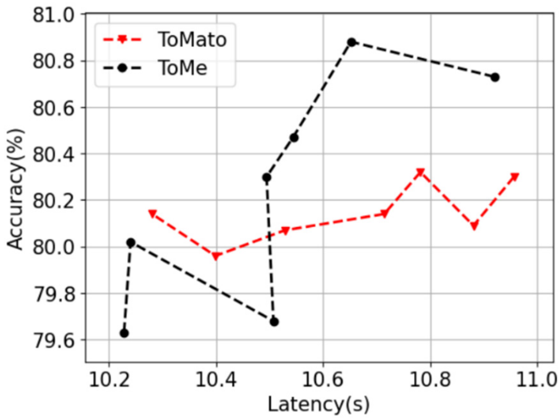


그림 3. ToMe, ToMato 의 정확도와 추론 시간의 상관관계를 나타낸 그래프

IV. 결론 및 향후 연구 방향

본 논문에서는 token merging 을 이용하여 ViT 를 가속화하는 프레임워크인 ToMato(Token Merging at Once)를 제안한다. 처음의 transformer block 에서 하나의 토큰을 기준으로 인접한 토큰들과의 유사도를 비교하며 연쇄적으로 토큰 병합을 진행하는 모델을 생성하였다. 실험 결과, 제안한 모델이 추가적인 모델 학습 없이도 높은 정확도를 유지하면서 전반적인 추론 속도를 개선하였음을 확인하였다.

그러나 현재 sim 값을 수동적으로 설정하고 있어 이에 대한 기준을 결정하는 것이 어렵다는 한계점이 있다. 따라서 향후 sim 값을 자동으로 조정하고, 각 토큰을 병합하는 과정을 병렬적으로 처리할 수 있도록 하는 연구를 추가적으로 진행할 예정이다.

사사

이 연구는 2021 학년도 이화여자대학교 교내 연구비 지원에 의한 연구임.

참고문헌

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *"Attention is all you need"*, NIPS, California, USA, Jul. 2017.

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. *"An image is worth 16x16 words: Transformers for image recognition at scale"*, ICLR, Vienna, Austria, May. 2021.

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. *"Token merging: Your vit but faster"*, ICLR, Kigali, Rwanda, May. 2023.

[4] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. *"Training data-efficient image transformers & distillation through attention"*, ICML, Jul. 2021.