구조적 프루닝을 사용한 MobileViT 경량화

설광수, 노시동, 정기석
한양대학교

kwang4010@hanyang.ac.kr, sdroh1027@hanyang.ac.kr, kchung@hanyang.ac.kr

Compressing MobileViT using Structured Pruning

Seol, Kwang-Soo, Roh, Si-Dong, Chung, Ki-Seok
Hanyang University, Seoul, Korea

요 약

MobileViT 은 Transformer 와 convolution 을 결합하여 feature map 의 global representation 을 학습하는 모델이다. MobileViT 은 비슷한 파라미터를 가진 Convolution Neural Network 나 Transformer 보다 정확도가 높은 반면 실행 시간이 느리다는 단점이 있다. 본 논문은 structured pruning 을 사용해 MobileViT 의 파라미터 수를 줄이고 실행 속도를 높이는 방안을 제시한다. 성능 평가를 위해서 CIFAR-10 데이터셋에 대한 이미지 분류 성능을 측정하였으며, 그 실험 결과, 0.06%의 정확도 감소로 파라미터 수를 60% 줄이고, 실행 시간을 약 18% 줄였다.

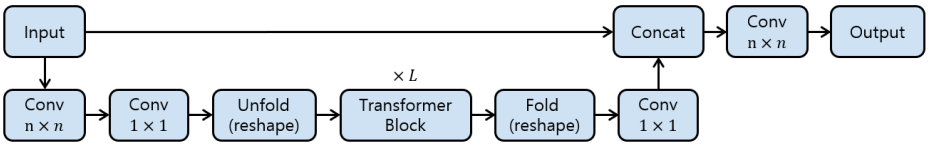

그림 1. MobileViT block 의 구조

Ⅰ. 서 론

Vision Transformer (ViT)는 자연어 처리에 사용되는 transformer 를 이미지 처리를 위해 변형한 모델이다 [1]. MobileViT 은 ViT 계열 모델로, Transformer 와 convolution 을 결합하여 feature map 의 global representation 을 학습하는 모델이다 [2]. MobileViT 은 파라미터의 개수가 비슷한 Convolution Neural Network (CNN) 모델들과 Transformer 기반 모델들보다 더 높은 정확도를 보이나 실행 속도가 느리다는 단점이 있다.

Structured pruning 은 pruning 기법들 중 하나로, 각각의 가중치를 제거하는 것이 아닌, 특정 구조 단위로 제거하는 기법이다. 가중치를 특정 구조 단위로 제거할 경우, 연산량이 줄어들어 실질적인 실행 속도 향상으로 이어진다.

본 논문에서는 structured pruning 을 사용해 MobileViT 을 경량화하여 정확도 감소를 최소화하면서도 모델의 파라미터의 수를 줄이고 실행 속도를 높이는 방안을 제시한다. 성능 평가를 위해서 CIFAR-10 데이터셋에 대한 이미지 분류 성능을 측정하였다.

Ⅱ. 본론

2.1. MobileViT

MobileViT 는 Convolution layer, MV2 block, MobileViT block 으로 구성되어 있다. MV2 block 은 MobileNetV2 의 inverted residual block 을 나타낸다 [3]. Inverted residual block 은 depthwise convolution 과 pointwise convolution 으로 구성된다.

MobileViT block 은 Vision Transformer (ViT)에 convolution 을 추가한 block 으로 그림 1 과 같은 구조를 가지고 있다. MobileViT block 에서 Conv 는 convolution layer 를 의미하고, feature map 의 local representation 을 학습하는 역할을 한다. MobileViT block 의 Transformer block 은 매 MobileViT block 마다 L 번 반복되며, self-attention 연산을 통해 feature map 의 global representation 을 학습하는 역할을 한다. Unfold 는 feature map 을 겹치지 않는 patch 들로 변환하는 연산이다. 반면에, Fold 는 patch 들을 feature map 으로 변환하는 연산이다. MobileViT block 은 Unfold, Transformer, Fold 과정을 통해 Transformer 를 convolution 처럼 사용한다.

2.2. Structured pruning

Structured pruning 은 pruning 기법들 중 하나로, 모델의 가중치를 특정 구조 단위로 제거한다. 가중치를 구조 단위로 제거하게 되면 사용되지 않는 가중치에 대한 연산을 제거하기 더 용이하기 때문에 GPU 등에서 실질적인 가속으로 이어진다. 일반적인 pruning 진행 절차는 다음과 같다. 먼저 pruning 을 진행할 모델을 학습시킨다. 모델의 학습이 완료되면, 학습된 모델에 대해서 pruning 을 진행한다. 그 후, pruning 된 모델을 다시 학습시킨다.

Filter pruning 은 structured pruning 중 하나로, 모델 내의 가중치를 filter 단위로 제거하는 기법이다 [4]. Filter pruning 은 filter 내에 가중치들의 l1-norm 을 기준으로 작은 filter 들을 제거한다. 제거하는 filter 의 비율은 모든 layer 가 같은 비율을 사용한다.

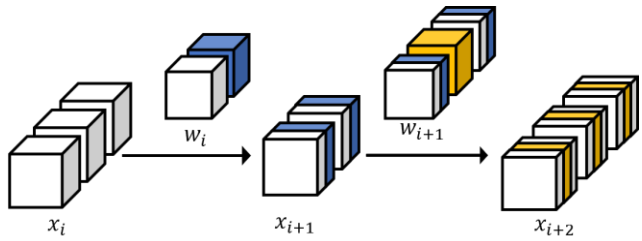

그림 2. Filter pruning 예시

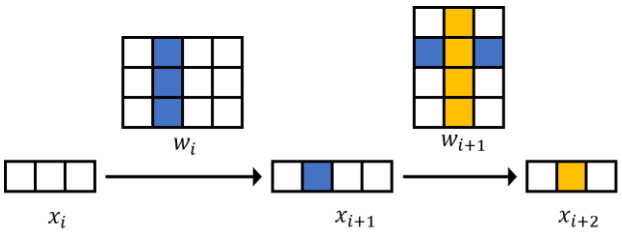

그림 3. FC layer 에서의 structured pruning 예시

2.3. 구현

MobileViT 모델은 논문 [2]를 참고하여 구현하였다. Structured pruning 은 Convolution layer 와 Transformer block 내부의 Fully Connected layer (FC layer)에 적용하였다. Convolution layer 에 사용한 pruning 은 filter pruning 으로 그림 2 와 같은 방식으로 구현하였다. 그림 2 에서 x_i 는 i 번째 feature map 을 가리키고, w_i 는 i 번째 layer 의 가중치를 나타낸다. 파란색과 노란색은 각각 i 번째와 i+1 번째 layer 에서 제거된 가중치들을 나타낸다. 그림 2 와 같이 여러 개의 convolution layer 가 이어져 있을 경우, 이전 layer 에서 제거된 filter 와 곱해져 생성되는 다음 레이어의 입력 channel 들도 제거했다. 이전 layer 에서의 입력 feature map 이 filter 가 제거된 가중치와 convolution 연산을 수행하면, 제거된 filter 의 수만큼 출력 feature map 의 channel 의 수가 줄어든다. Convolution 연산을 진행하기 위해서는 입력 feature map 과 가중치의 channel 의 개수가 동일해야 하므로, 가중치의 filter 뿐만 아니라 channel 도 제거되어야 한다.

FC layer 는 그림 3 과 같이 structured pruning 을 사용해 가중치를 column 단위로 제거하였다. 그림 3 에서 x_i 와 w_i 는 각각 i 번째 입력과 가중치를 나타낸다. 파란색과 주황색은 각각 i 번째와 i+1 번째 layer 에서 제거된 가중치를 나타낸다. 여러 개의 FC layer 가 이어져 있을 경우, 이전 layer 의 가중치에서 제거된 column 에 해당하는 row 도 추가로 제거했다.

2.4. 실험 및 결과 분석

제안하는 방법의 성능 평가를 위해서 structured pruning 을 적용한 MobileViT 의 이미지 분류 성능을 측정하였다. 성능 지표에는 정확도와 latency 를 사용하였다. 실험에 사용한 데이터셋은 CIFAR-10 이고, NVIDIA GeForce RTX 3090 1 개를 사용하여 latency 를 측정하였다. Structured pruning 을 적용할 때 사용한 pruning rate 는 20% ~ 80%이다. Pruning 되기 전 모델과 pruning 을 진행한 후 생성된 모델의 학습은 각각 200 epoch 동안 진행하였다.

실험 결과는 표 1 과 같다. Structured pruning 을 적용하지 않았을 때, 즉, pruning rate 가 0%일 때와 비교해서 pruning rate 가 20%, 40%일 때는 정확도가 각각 0.41%, 0.04% 증가하였다. Pruning rate 가 60%일 때는, 0.09%의 정확도 감소로 17.51%의 latency 가 감소되는 것을 확인하였다. Pruning rate 를 80%까지 높이면 latency 를 20.73% 감소시킬 수 있었으나, 정확도가 1% 이상 감소하므로 감소하는 latency 에 비해 정확도가 크게 떨어지는 것을 확인할 수 있었다.

표 1. Pruning rate 에 따른 MobileViT 의 CIFAR-10 이미지 분류 성능

pruning rate	정확도	파라미터 수	Latency	Latency 감소율
0 %	89.69 %	5.00M	11.44 ms	0.00 %
20 %	90.10 %	4.01M	11.02 ms	6.77 %
40 %	89.73 %	3.03M	11.00 ms	6.94 %
60 %	89.60 %	2.06M	9.75 ms	17.51 %
80 %	88.51 %	1.09M	9.37 ms	20.73 %

Ⅲ. 결론

본 논문에서는 structured pruning 을 사용해 MobileViT 의 파라미터 수를 줄이고 실행 속도를 높이는 방안을 제시하였다. CIFAR-10 데이터셋에 대하여 경량화 한 모델의 이미지 분류 성능을 평가한 결과, pruning rate 가 40% 이하일 때 정확도가 향상되었다. 또한 60%의 가중치를 제거하면 0.09%의 정확도 감소로 latency 를 약 18% 감소시킬 수 있다는 것을 확인하였다.

ACKNOWLEDGMENT

본 논문은 2022 년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2022-0-00153, 빔포밍 경로 분산을 이용한 AI 네트워크관리 기반 인빌딩용 O-RU 개발)

참 고 문 헌

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, N. Houlsby. "An image is worth 16x16 words: Transformers for image recognition at scale". In International Conference on Learning Representations, 2021.

[2] S. Mehta, M. Rastegari. “Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer”. International Conference on Learning Representations, 2021.

[3] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, L. C. Chen. "Mobilenetv2: Inverted residuals and linear bottlenecks". In Proceedings of the IEEE conference on computer vision and pattern recognition, 2018.

[4] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf. “Pruning filters for efficient convnets”. International Conference on Learning Representations, 2016.