레터논문 (Letter Paper)
방송공학회논문지 제29권 제5호, 2024년 9월 (JBE Vol.29, No.5, September 2024)
https://doi.org/10.5909/JBE.2024.29.5.752
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

Transformer의 개별 가지치기를 이용한 효율적인 이미지 캡셔닝 기법

권 오 설[a)‡]

Efficient Image Captioning Using Individual Pruning on Transformer

Oh-Seol Kwon[a)‡]

요 약

본 논문에서는 이미지 캡셔닝에서 개별 가지치기 기법을 통해 효율적인 트랜스포머 네트워크를 제안한다. 일반적으로 이미지 캡션 모델은 사전 학습된 CNN 인코더, 트랜스포머 인코더 및 디코더의 세 가지로 구성된다. 본 연구에서는 캡션 모델의 각 구성 요소를 개별적으로 최적화하도록 설계한 가지치기(Pruning) 기술을 통해, 전체 구조가 기존 캡셔닝 모델과 다르더라도 인코더 또는 디코더 네트워크와 같은 유사한 구성 요소를 공유하는 모델에 대한 적용성을 넓혔다. 또한 디코더에서 캡셔닝을 위한 손실함수를 적용함으로써 성능을 향상시켰다. 본 모델을 영문 및 한글 버전에 적용한 결과 기존 대비 우수한 성능을 확인하였다.

Abstract

In this letter, we propose an efficient transformer network using individual pruning techniques in image captioning. Typically, an image caption model consists of three things: a pre-trained CNN encoder, a transformer encoder, and a decoder. In this study, a proposed pruning technique was designed to optimize each component of a caption model individually and shared similar components, such as encoder or decoder networks, even if the overall structure is different from conventional captioning models. Additionally, proposed method was applied a loss function for captioning in the decoder. As a result of applying this model to the English and Korean versions, superior performance was confirmed compared to the existing model.

Keyword : Super-resolution, Deep Learning, Deep Residual Block

a) 창원대학교 로봇제어계측공학과(Dept. of Robot Control and Instrumentation Engineering, Changwon National University)
‡ Corresponding Author : 권오설(Oh-Seol Kwon)
E-mail: osk1@changwon.ac.kr
Tel: +82-55-13-3669
ORCID: http://orcid.org/0000-0002-1077-9615
※본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT 연구센터사업의 연구결과(IITP-2024-RS-2024-00438409) 및 2024년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다.(2021RIS-003)
· Manuscript August 16, 2024; Revised September 4, 2024; Accepted September 6, 2024.

Ⅰ. 서 론

이미지 캡션을 위한 심층 신경망(DNN)에 대한 연구는 모델의 성능을 크게 향상시켰다. 특히 MS-COCO[1] 데이터 세트의 최첨단 모델에 대한 CIDEr[2] 점수는 66점에서 130점 이상으로 상승했다. 그러나 이러한 발전은 일반적으로 모델 크기의 상당한 증가를 기반으로 이루어낸 결과이다. 대표적인 사례로 디코더 크기가 1,200만 개에서 5,500만 개

의 매개변수로 증가한 것을 확인할 수 있다. 최신 이미지 캡셔닝 모델은 일반적으로 사전 훈련된 CNN 인코더, 트랜스포머 인코더 및 트랜스포머 디코더의 세 가지 주요 구성 요소로 구성된다. 이때, 우수한 성능과 모델 크기의 증가를 방지하기 위해 네트워크에서 필수적이지 않은 가중치를 제거하는 다양한 가지치기(Pruning) 기법에 대한 연구가 진행되고 있다. 이러한 가지치기 기법들은 배포 과정에서의 속도 향상, 스토리지 요구 사항 감소, 에너지 소비 감소 등에서 장점이 있다. 최근에는 이미지 캡셔닝 모델에 적용하기 위한 end-to-end 가지치기 기법[3][4]이 제안되었다. 이 연구들은 기존의 기법들이 재사용되는 가중치의 존재로 인해 순방향 신경망(feedforward network)을 위해 설계된 변형 가지치기 방법을 적용하기 어렵고, 또한 이미지 캡셔닝의 다중 모드 작업의 내재적 복잡성에 원인을 두고 있다. 이를 극복하기 위해서 Tan et al.[3]은 end-to-end 방식을 사용해 파라미터 민감도를 기반으로 훈련 단계에서 지속적이고 점진적인 희소화를 구현하는 SMP(Super-Mask Pruning) 기술을 제안하였다. Dai et al.[4]는 기울기(Gradient) 기반의 성장(Growth)과 가지치기(Pruning)를 이용하여 네트워크를 최적화하는 방법을 제안하였다. 이를 통해 기존 방법의 문제인 중복성과 수행 시간의 연장에 대한 문제를 해결하고자 하였다. 그러나 이러한 방식[3][4]들은 서로 유사하지만 다른 아키텍처를 가진 모델들을 쉽게 일반화할 수 없는 단점이 있다. 예를 들어, 이미지 캡셔닝 모델에는 일반적으로 ViT(Vision Transformer) 인코더 및 LM(Language Model) 디코더가 포함된다. 따라서 각 구성 요소, 즉 사전 훈련된 인코더, 트랜스포머 인코더 네트워크 및 트랜스포머 디코더 네트워크를 개별적으로 처리하는 새로운 가지치기 방법이 필요하다.

Ⅱ. 제안한 개별 가지치기에 기반한 이미지 캡셔닝 기법

본 연구에서는 이미지 캡셔닝 모델용 프레임워크에서 백본 역할을 하는 사전 학습된 ResNet, 트랜스포머 인코더, 트랜스포머 디코더의 세 가지 주요 구성 요소로 이루어진다. 본 연구의 핵심인 개별 가지치기(Individual Pruning)를 위해서 ResNet, 트랜스포머 인코더 및 트랜스포머 디코더와 같이 각 네트워크에 대해서 개별적으로 가지치기를 진행하였다.

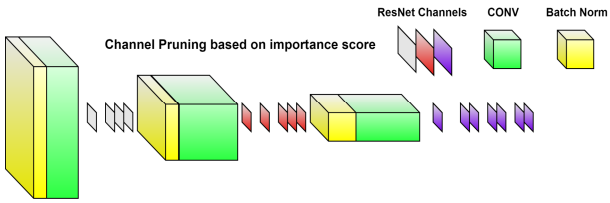

그림 1. 제안한 가지치기 기법을 적용한 ResNet 모델
Fig. 1. Proposed pruning ResNet model

먼저 ResNet 단계에서 기존의 가지치기 방법은 다음과 같다. 사전 훈련된 네트워크에서 희소성 유도 항을 사용하여 중복 채널을 제거한 후 미세 조정을 거치게 된다. 이때, 사용되는 그룹 올가미 기법(group lasso technique)은 계산이 까다롭고 수렴하기 어려우며 모델 구조의 단순화로 인해 성능이 저하되는 경우가 많다. Kethan et al.[5]은 네트워크의 모든 계층에 적용할 수 있는 채널 정리 기법을 도입하여 서로 다른 계층에서 다양한 수의 채널을 제거할 수 있도록 하였다. 이 기법은 컨벌루션-배치 정규화 및 ReLU 활성화를 통합하는 표준 ResNet-101 구조로 설계하였다. B가 현재 미니 배치를 나타낸다고 가정하면, 표준 BN 레이어는 수식 (1)에 표현된 것처럼 {1,2,…, n}에 대해 i번째 피처 맵 각각에 대한 어파인(Affine) 변환을 수행한다.

$$\hat{z}_i = \frac{z_i^{(in)} - \mu_{B_i}}{\sqrt{\sigma_{B_i}^2 + \epsilon}} \;;\; z_i^{(out)} = \gamma_i \hat{z}_i + \beta_i \tag{1}$$

위 수식에서, $\hat{z}_i$는 정규화된 i번째 피처 맵, $z_i^{(out)}$는 i번째 출력 피처 맵, μ_β는 배치 B에 대한 i번째 피처 맵의 평균, γ_i는 i번째 출력 채널 $z_i^{(out)}$의 표준 편차, β_i는 i번째 출력 채널의 평균을 나타낸다. γ_i^2항은 i번째 출력 채널의 분산을 조절하고 ϵ는 임의의 작은 양의 실수이다. 활성화 효과를 무시하면, l번째 컨볼루션 레이어의 i번째 입력 채널은 $\gamma_{l-1,i}^2$의 분산을 갖는다. 전체 Resnet 네트워크에서 $W \equiv \{W_l\}_{\{1,2,\ldots,L\}}$은 모든 합성곱 매개변수 집합을 나타내고 $\gamma = \{\gamma_{l,i}, \beta_{l,i}\}_{l,i}$은 배치 정규화 레이어의 매개변수

를 나타낸다. 따라서 l번째 합성곱 계층에서 j번째 출력의 분산에 대한 i번째 입력의 기여도는 다음 수식 (2)와 같다.

$$\gamma_i^2 = \gamma_{\ell-1,i}^2 \left\| W_{\ell,i,j} \right\|_2^2 \tag{2}$$

이때, 각 입력 채널이 출력 채널의 분산에 미치는 영향(I_c)은 모든 출력을 동시에 고려하여 수식 (3)과 같이 설정된다.

$$I_c = \gamma_{\ell-1,i}^2 \sum_{j=1}^{n_\ell} \left\| W_{\ell,i,j} \right\|_2^2 \tag{3}$$

합계는 간단히 1로 합산되는 스칼라 값으로 변형할 수 있다. 결과적으로 최종 전역 중요도 점수는 $\gamma_{(L-1,i)}$로 표시되며, 이는 i번째 입력 채널이 l번째 합성곱 계층의 분산에 기여하는 정도를 정량화한다.

다음으로 인코더 네트워크(Encoder network) 단계는 인코더와 디코더의 상관성 및 인코더 레이어 수에 관한 Ko et al.[6]의 결과를 이용하였다. 그 연구에 따르면 인코더 네트워크의 희소성(sparsity)이 인코더-디코더 LM(Language Model)의 출력 품질에 상당한 영향을 미치는 반면 인코더 레이어의 수는 추론 시간에 큰 영향을 미치지 않음을 제안하였다. 이미지 캡션 모델에서 사용된 인코더-디코더 네트워크가 기존 LM과 동일한 아키텍처를 반영한다는 점을 반영하여 너비 가지치기(Width pruning)을 적용하였다. 다만, Ko et al.[6]는 대상과 현재 희소성 간에 등가 제약 조건을 적용하여 ℓ_0 정규화한 반면에 본 연구에서는 모든 인코더 가중치에 걸쳐 일반 가중치 정규화를 적용했다. 또한, ℓ_2 정규화를 사용하여 모델 전체에서 적절한 그래디언트 흐름을 유지하고 λ_1을 0.01로 설정한다. 따라서 인코더의 손실 기여도는 다음과 같이 표현된다.

$$L_{enc} = \lambda_1 \sum_{\{i,j\}} \left\| W^{2\,enc}_{i,j} \right\| \tag{4}$$

마지막 단계인 디코더 네트워크(Decoder network)에서는 디코더 레이어의 수가 추론 시간과 모델 크기에 직접적으로 비례한다[6]는 것을 기반으로 깊이 가지치기(Depth pruning)를 적용하였다. 지정된 수의 선택된 레이어 ds에 대해, Ls는 선택된 레이어의 인덱스를 나타내고 디코더 하위 네트워크는 수식 (5)에 설명된 대로 균일 샘플링(Uniform sampling) 통해 생성된다.

$$L_s = \left\{ \left\lfloor \frac{L-1}{d_s-1} \right\rfloor \cdot \ell + 1 \mid \ell \in \{0, \ldots, d_s-1\} \right\} \tag{5}$$

Ko et al.[6]은 디코더 서브네트워크 $\left(H^l_{dec,s}\right)$의 숨겨진 상태를 정렬하기 위해 은닉 상태 증류(hidden state distillation)를 활용하였다. 평균 제곱 오차(MSE), $H_{dec,l}\left[\frac{L-1}{d_s-1}, l+1\right]$은 원래 디코더 네트워크에서 선택된 상태이고, 은닉 상태 증류 손실$\left(L_h^{dec}\right)$은 수식 (6)과 같다. 이 방정식은 Ko et al.[6]의 디코더 네트워크의 손실 기여도를 나타낸다.

$$L_h^{dec} = \sum_{\ell \in \{1,2,\ldots,d_s\}} \mathrm{MSE}\left(H^{\ell}_{dec,s}, H_{dec,\ell}\left\lfloor \frac{L-1}{d_s-1} \cdot \ell + 1 \right\rfloor \right) \tag{6}$$

본 논문에서는 이미지 캡셔닝 응용을 위해 수식 (7)과 같이 최종 레이어를 제외하고, 디코더 서브 네트워크의 모든 은닉 상태를 원래 디코더 마지막 레이어의 출력 이미지의 실제 캡션으로 정렬하였다.

$$L_h^{dec} = \sum_{\ell=1}^{d_s-1} \mathrm{MSE}\left(H^{\ell}_{dec,s}, H_{dec,\ell}\left\lfloor \frac{L-1}{d_s-1} \cdot \ell + 1 \right\rfloor \right) \tag{7}$$

여기서 CC는 올바른 캡션을 나타내고 CE는 수식 (8)의 교차 엔트로피 손실을 나타낸다.

$$L_{dec}^{total} = L_h^{dec} + CE\left(CC, H_{dec,d_s}\right) \tag{8}$$

제안한 방식은 출력이 원래 캡션과 밀접하게 반영되도록 하는 목적으로 구성하였다. 따라서 가지치기된 디코더와 가지치기되지 않은 디코더의 마지막 레이어 출력을 정렬하는 대신, 가지치기된 디코더의 최종 출력을 이미지의 원본 캡션과 매칭시켰다.

Ⅲ. 실험 및 결과

제안된 방법의 실험 방법 및 결과를 기존 방법과 비교를 통해서 제시하였다. 모든 실험은 이미지당 5개의 캡션이 포함된 Flickr8k 데이터 세트를 통해 수행되었다. 먼저 데이터 세트에서 5번 이상 나타나는 단어로 어휘를 만들고 Spacy tokenizer를 사용해 이를 토큰화하였다. 학습률은 모든 실험에 대해 “Karpathy의 학습률”이라고 불리는 0.0003값으로 설정하였다. 어텐션 헤드(Attention head)의 수는 셀프 어텐션(self-attention)과 교차 어텐션(cross-attention) 레이어 모두 8개로 고정하였고, 배치 크기는 32개로 유지하였다. 모든 모델은 약 20-30 에포치(epoch) 동안 학습하였다. 이러한 하이퍼파라미터는 기존 연구와 이미지 캡션 모델의 오픈 소스 구현에서 설정하였다. 실험을 위해 본 논문에서는 이미지 캡션 작업에 대해 잘 정립된 두 가지 지표인 ROUGE-L 및 CIDEr 점수를 사용하여 모델을 평가하였다. 표 1에서 실험 결과 모델 크기의 감소에도 불구하고 성능 저하가 ROUGE-L 점수에 유의미하지 않으나 CIDEr 점수에 대해서는 눈에 띄는 감소가 됨을 확인하였다. 또한 표 2에서 보듯이 네트워크의 복잡도 또한 레이어의 수에 따라 감소하게 된다.

표 1. 가지치기 전후의 텍스트 성능 결과 비교
Table 1. Comparative analysis of performance scores between pruned and unpruned models

Name of the Model	ROUGE-1	ROUGE-L	CIDEr
Original network	0.3740	0.3478	0.7980
Ko et al. method[6]	0.3104	0.2880	0.4320
Proposed method	0.3110	0.2894	0.4377

표 2. 가지치기 전후의 복잡도 비교
Table 2. Comparison of model size of pruned and unpruned models in terms of memory (MB)

Name of the Model	Unpruned	1D	2D	3D
Original network(4D)	346.5	-	-	-
Proposed method	-	218.2	240.0	256.4

Ⅳ. 결 론

본 논문은 개별 가지치기 기법을 통해 효율적인 이미지 캡셔닝을 위한 트랜스포머 네트워크를 제안하였다. 제안한 방법은 캡션 모델의 각 부분에 대해 설계된 특정 가지치기 기술을 통해 인코더 또는 디코더 네트워크와 같은 유사한 구성 요소를 가진 모델에 대한 일반화 가능성을 도출하였다. 또한 추론 성능 향상을 위해 디코더 가지치기에 대한 새로운 접근 방법을 제안하였다. 실험 결과 제안한 방법은 기존 연구에 적용할 수 있을 뿐만 아니라 이미지 캡션 분야에서 모델의 효율성과 성능의 우수성을 확인하였다.

참 고 문 헌 (References)

[1] T. Y. Lin, M. Marie, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in 13th European Conference: Computer Vision, Zurich, Switzerland, pp. 745-755, 2014.
doi: https://doi.org/10.48550/arXiv.1405.0312

[2] R. Vedantam, C. L. Zitnick, and D. Parikh, “CIDEr: Consensus-based image description evaluation,” in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015.
doi: https://doi.org/10.1109/CVPR.2015.7299087

[3] J. H. Tan, C. Chan, and J. J. Chuah, “End-to-End Super mask Pruning: Learning to Prune Image Captioning Models,” Pattern Recognition, vol. 122, no. 1, pp. 1-12, 2022.
doi: https://doi.org/10.1016/j.patcog.2021.108366

[4] X. Dai, H. Yin, “Grow and Prune Compact, Fast, and Accurate LSTMs,” IEEE Transactions on Computers, vol. 69, no. 3, pp. 441-452, 2020.
doi: https://doi.org/10.1109/TC.2019.2954495

[5] A. Khetan, and Z. Karnin, “PruneNet: Channel Pruning via Global Importance,” 2020, arXiv:2005.11282 [cs. LG].
doi: https://doi.org/10.48550/arXiv.2005.11282

[6] J. Ko, S. Park, Y. Kim, S. Ahn, D. Chang, E. Ahn, and S. Yun, “NASH: A Simple Unified Framework of Structured Pruning for Accelerating Encoder-Decoder Language Models,” in Findings of the Association for Computational Linguistics: EMNLP 2023, Sentosa Gateway, Singapore, 2023.
doi: https://doi.org/10.18653/v1/2023.findings-emnlp.404

[7] M. Tanti, A. Gatt, and K. P. Camilleri, “Where to put the image in an image caption generator,” Natural Language Engineering, vol. 24, no. 3, pp. 467-489, May 2018.
doi: https://doi.org/10.1017/S1351324918000098