Optimization of Image Captioning Networks Using Targeted Component Pruning Method

Jishu Sen Gupta[a], Yogendra Rao Musunuri[b], Ih-Man Seo[b], and Oh-Seol Kwon[b]
[a] Department of Mathematical Sciences, Indian Institute of Technology (BHU), Varanasi, Uttar Pradesh, 221005, India.
[b]Department of Control and Instrumentation Engineering, Changwon National University, Changwon, Gyeongnam 644-731, Korea.

Abstract

Advancements in deep learning models have significantly enhanced the image captioning performance over the past decade. However, these improvements have resulted in increased model complexity and higher computational costs. Contemporary captioning models typically consist of three components: a pre-trained CNN encoder, a transformer encoder, and a decoder. Although research has extensively explored network pruning for captioning models, it has not specifically addressed the pruning of these three individual components. As a result, existing methods lack the generalizability required for models that deviate from the traditional configuration of image captioning systems. In this study, we introduce a pruning technique designed to optimize each component of the captioning model individually, thus broadening its applicability to models that share similar components, such as encoders or decoder networks, even if their overall architectures differ from conventional captioning models. Additionally, we implemented a novel modification during the pruning of the decoder, which significantly improved the inference performance of the image-captioning model.

Introduction

In recent years, the field of image captioning has witnessed significant advancements, primarily driven by the development of deep learning models. These enhancements have improved performance metrics considerably; however, they have also led to increased model complexity and elevated computational demands. Over the past decade, research focused on deep neural networks (DNNs) for image captioning has significantly enhanced model performance. Notably, the CIDEr [1] scores for state-of-the-art models on the MS-COCO dataset have risen from 66 to over 130 points. However, these advancements have typically resulted in substantial increases in model size, exemplified by the growth in decoder size from 12 million to 55 million parameters.

Contemporary image-captioning models typically consist of three primary components: a pre-trained convolutional neural network (CNN) encoder, a transformer encoder, and a transformer decoder. To mitigate the increase in model size, various pruning techniques have been developed to remove non-essential weights from the network. These pruning methods offer multiple benefits, including enhanced speed, reduced storage requirements, and lower energy consumption, especially during deployment.

Despite these developments, current research on network pruning for these models has not thoroughly addressed the distinct components, resulting in methods that are not universally applicable to models with varying architectures. This oversight restricts the generalizability of pruning techniques, particularly for models that incorporate similar components but differ in their overall structure of the models.

In this paper, we introduce a novel approach that employs distinct pruning techniques for each component of an image-captioning model. Our objective is to establish a generalized pruning strategy applicable to various models featuring encoder and decoder networks. This approach advances prior research by presenting a unique method for pruning the decoder, thereby enhancing inference performance. The study aims to offer a more adaptable and efficient solution for optimizing image-captioning models, which could benefit a wider array of deep learning applications.

Related Work

Recent studies [2] [3] have conducted end-to-end pruning of image-captioning models. Tan et al. [2] developed a super-mask pruning (SMP) technique that implements continuous and gradual sparsification during the training phase, based on parameter sensitivity in an end-to-end fashion. In [2], the authors note the scarcity of previous work on pruning image captioning models due to two primary challenges: first, the presence of weights that are shared and reused across time steps, which complicates the application of variational pruning methods designed for feed-forward networks; second, the inherent complexity of the multi-modal task of image captioning, requiring any proposed method to perform effectively across both image and language domains.

Furthermore, methods [2] and [3] address the pruning of image captioning models within an end -to -end framework. However, their approaches are not easily generalizable to models with similar yet distinct architectures. For example, image -captioning models often include vision transformer (ViT) encoders and language model (LM) decoders. Therefore, we propose a new method for pruning that treats each component—namely, the pre-trained ResNet encoder, the transformer encoder network, and the transformer decoder network—separately, as depicted in Figure 1. This method ensures that the pruning processes for each component are independent of one another.

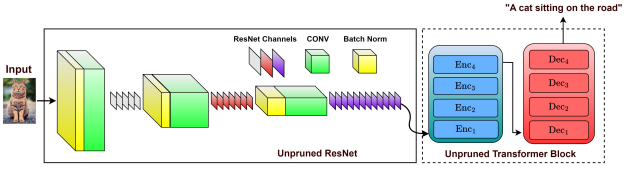

Figure 1. *Pipeline of the image captioning model.*

Method

To implement the pruning techniques, we utilized a framework for the image-captioning model as depicted in Figure 1, which comprises three main components: a pre-trained ResNet serving as the backbone, a transformer encoder, and a transformer decoder. We propose pruning each of these components separately, as follows: ResNet pruning, transformer encoder network pruning, and transformer decoder network pruning.

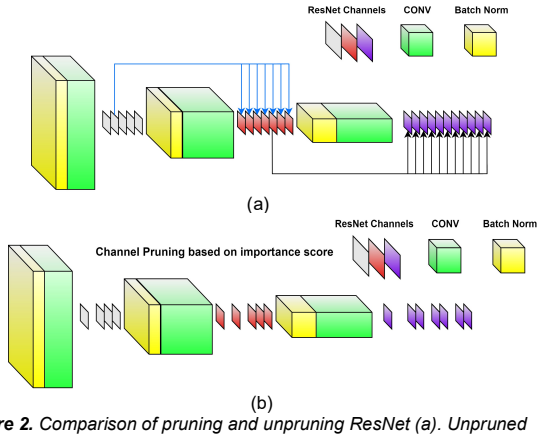

Figure 2. *Comparison of pruning and unpruning ResNet (a). Unpruned ResNet, and (b). Pruned ResNet model.*

ResNet Pruning

Traditional pruning methods, which involve removing redundant channels through the use of a sparsity-inducing term in a pretrained network followed by fine-tuning, face several challenges. The group lasso technique employed in these methods is computationally demanding, challenging to converge, and often leads to diminished performance due to the simplified model architecture. Kethan et al. [4] introduced a channel pruning method applicable across all layers of a network, allowing for a varying number of channels to be pruned across different layers. This method is designed for a standard ResNet-101 architecture incorporating convolution-batch normalization, and ReLU activation. Let B denote the current mini-batch, a standard BN layer performs the following affine transformation for each of the i-th feature map $z_i = \mathbb{R}^{r \times r}$, for $i \in \{1, 2, \ldots, n\}$ as shown in (1).

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

In this context, $\hat{z}_i$ represents the normalized i-th feature map, $z_i^{(out)}$ denotes the i-th output feature map, μ_{B_i} represents the mean of the i-th feature map over the batch B, γ_i is the standard deviation of the i-th output channel $z_i^{(out)}$ and β_i denotes the mean of the i-th output channel $z_i^{(out)}$. The term γ_i^2 controls the variance of the i-th output channel $z_i^{(out)}$ and ϵ is a small positive number. Neglecting the effect of activations, the i-th input channel of l-th convolution layer is has a variance of $\gamma_{\ell-1,i}^2$. For the entire ResNet, $W \equiv \{W_l\}_{\{1,2,\ldots,L\}}$ denotes the set of all convolution parameters, and $\gamma = \{\gamma_{l,i}, \beta_{l,i}\}_{l,i}$ represents the parameters of the batch normalization layers. Thus, the contribution of the i-th input to the variance of the j-th output in the l-th convolution layer is described by Eq. (2):

$$\gamma_{\ell-1,i}^2 \left\| W_{\ell,i,j} \right\|_2^2 \qquad (2)$$

Figure 2(a) illustrates the influence of each input channel on the variance of the output channels. When all outputs are considered simultaneously, the importance criteria are established in Eq. (3).

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

The summation can simply be modified with a scalar that it sums as one: $\sum_{j=1}^{n_\ell} \left\| W_{\ell,i,j} \right\|_2^2 = 1$. Consequently, the final global importance score is denoted by $\gamma_{(L-1,i)}$, which quantifies the extent to which i-th input channel contributes to the variance of the l-th convolution layer. To achieve the desired pruning ratio (η) over T iterations, the following steps are undertaken:

- Train the ResNet backbone on a large dataset, applying appropriate regularization on the batch normalization variance γ_i^2.
- Rank the channels according to their global importance score γ_i.
- Prune η/T channels based on to their importance score and fine tune the pruned model on a downstream target dataset.
- Repeat the process starting from step 2 for T iterations to attain the desired level of sparsity.

The ResNet backbone is pruned independently. During the training of the image captioning model, the pruned ResNet is loaded while its weights are frozen. Figure 2(b) displays the final pruned ResNet configuration.

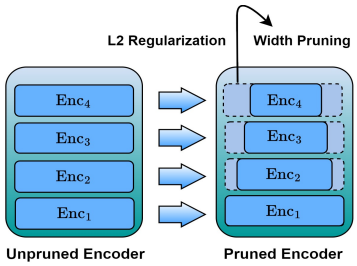

Figure 3. *Pruning process applied to the Encoder.*

Encoder Network Pruning

Ko et al. [5] demonstrated that the sparsity of the encoder network significantly influences the output quality of encoder-decoder LMs, where as the number of encoder layers does not significantly affect inference time. Given that, the encoder-decoder network utilized in our image-captioning model mirrors the same architecture of traditional LMs, similar trends are anticipated. Width pruning is applied to the encoder network, as depicted in Figure 3. Unlike Ref. [5], which employed ℓ_0 regularization by enforcing an equality constraint between the target and current sparsity, we do not specify a target sparsity and instead apply regular weight regularization across all encoder weights. Additionally, ℓ_2 regularization is employed to maintain appropriate gradient flows throughout the model, with λ_1 set at 0.01. Thus, the loss contribution from the encoder is expressed by Eq. (4).

$$\lambda_1 \sum_{\{i,j\}} \left\| W^2{}_{i,j}^{enc} \right\| \qquad (4)$$

Decoder Network Pruning

Ko et al. [5] observed that the number of decoder layers was directly proportional to both the inference time and the model size. Accordingly, depth pruning was applied to the decoder network, as depicted in Figure 4. For a specified number of selected layers d_s, L_s represents the index of the selected layer, and a decoder subnetwork is generated through uniform sampling, as described in Eq. (5).

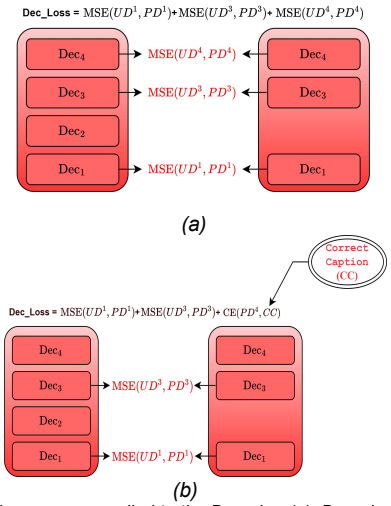

Figure 4. *Pruning process applied to the Decoder. (a). Decoder pruning [5], and (b). Proposed novel change in decoder pruning.*

$$L_s = \left\{ \left\lfloor \frac{L-1}{d_s-1} \right\rfloor . \ell + 1 \mid \ell \in \{0, \ldots, d_s - 1\} \right\} \quad (5)$$

Ko et al. [5] utilized hidden state distillation to align the hidden states of the decoder subnetwork ($H^{\ell}_{dec,s}$) with those of the original decoder network. The mean square error (MSE), $H_{dec,\ell} \left\lfloor \frac{L-1}{d_s-1} . \ell + 1 \right\rfloor$ is the selected state from the original decoder network, and the hidden state distillation loss (L_h^{dec}) is illustrated in Eq. (6). This equation represents the loss contribution from the decoder network, referred to as the pruned_pipeline [5].

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

We adopted a slightly different approach. We matched all hidden states of the decoder subnetwork to those of the original decoder network, except for the final laye, as stated in Eq. (7).

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

The output of the last layer is aligned with the true caption of the image, where CC denotes the correct captions, and CE represents the cross-entropy loss as expressed in Eq. (8). When employing the loss described in Eq. (8) as the loss contribution from the decoder network, this is referred to as the proposed pruned-novel change in the decoder (ours).

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

The rationale for this approach is to ensure that the output closely mirrors the original caption. Therefore, rather than aligning the last layer outputs of the pruned and unpruned decoders, we aligned the final output of the pruned decoder with the true caption of the image. Additionally, the proposed decoder pruning method demonstrated superior performance compared to the original decoder pruning technique. The final loss optimized during training is outlined in Eq. (9):

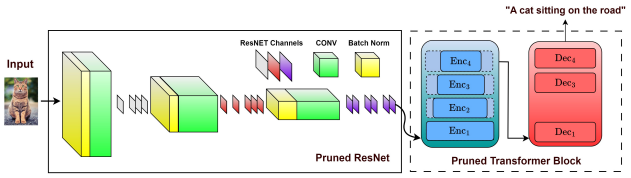

Figure 5. *Pipeline of the pruned image captioning model.*

$$L^{total} = \lambda_1 \sum_{\{i,j\}} \left\| W^{2\,enc}_{i,j} \right\| + \lambda_2 L_h^{dec} + CE\left(CC, H_{dec,d_s}\right) \quad (9)$$

where L^{total} represents the total loss optimized for the pruned network during training, and λ_2 represents the decoder loss contribution coefficient set at 0.01. The final pruning model for the image-captioning model is displayed in Figure 5. This model integrates individual network components, including ResNet pruning, encoder pruning, and decoder pruning. Experiments have demonstrated that this proposed method enhances performance.

Table 1. Comparative analysis of performance scores between pruned and unpruned models.

Name of the Model	ROUGE-1	ROUGE-L	CIDEr
Unpruned Model	0.3740	0.3478	0.7980
Pruned_pipeline [5]	0.3104	0.2880	0.4320
Proposed pruned novel change in decoder (ours)	0.3110	0.2894	0.4377

Experiments and Results

All experiments were conducted using the Flickr8k dataset, which contains five captions per image. Initially, a vocabulary was created from words that appeared at least five times across the dataset and were tokenized using the Spacy tokenizer. The learning rate was set at 0.0003, referred to as "Karpathy's learning rate," for all experiments. The number of attention heads in both the self-attention and cross-attention layers was fixed at 8, while the batch size was maintained at 32. All models underwent training for approximately 20-30 epochs.

These hyperparameters draw inspiration from previous studies and open-source implementations of image-captioning models. We primarily evaluated our model using two well-established metrics for image-captioning tasks: the ROUGE-L and CIDEr scores. Preliminary results indicate that despite a reduction in the model size, the decrease in performance is not significant for the ROUGE-L score, whereas a noticeable decline is observed for the CIDEr score. Additionally, our proposed decoder pruning method outperforms the results achieved by relying solely on hidden state distillation for the decoder network. These findings are documented in Table 1. A comparison of the sizes of the pruned and unpruned components is presented in Table 2. The target sparsity for the ResNet encoder and decoder network is established prior to determining the sparsity of the encoder network, which varies depending on the training process and the data input. Qualitative results are illustrated in Figure 6.

Table 2. Size comparisons of pruned and unpruned models.

Component	Size Ratio (Pruned/Unpruned)
ResNet	0.5
Decoder	0.5
Encoder	0.5-0.7

Figure 5. *The quantitative results of the image captioning model, (a). Original Model, (b). Pruned_pipeline [5], and (c). Pruned_novel change in decoder (ours).*

Conclusion

In conclusion, the advancement of deep learning models has significantly enhanced image captioning performance, though this improvement often comes at the cost of increased model complexity and computational demands. Contemporary image-captioning models typically comprise a pre-trained CNN encoder, a transformer encoder, and a transformer decoder. Previous efforts in network pruning of these models have not addressed these components individually, thereby limiting their applicability to diverse model architectures. This study introduces specific pruning techniques designed for each part of the captioning model, thereby enhancing their generalizability to models with similar components, such as encoder or decoder networks. Additionally, this study proposes a novel approach to decoder pruning, which shows considerable promise in enhancing inference performance. This methodology not only builds upon existing research but also advances the frontiers of model efficiency and adaptability in the field of image captioning.

Acknowledgments

This work was supported in part by the Regional Innovation Strategy (RIS) through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) under Grant 2021RIS-003; in part by the Development Project of Industry Academic-Research Platform Collaboration Technology in 2022 under Grant S3311946; and in part by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program under Grant RS-2024-00438409 supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP).

References

[1] 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.

[2] 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.

[3] X. Dai, H. Yin, "Grow and Prune Compact, Fast, and Accurate LSTMs," IEEE Transactions on Computers, vol. 69, no. 3, pp. 441-452, 2020.

[4] A. Khetan, and Z. Karnin, "PruneNet: Channel Pruning via Global Importance," 2020, arXiv:2005.11282 [cs. LG].

[5] 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.

Author Biography

Jishu Sen Gupta is currently pursuing an Integrated Dual Degree (B-Tech + M-Tech) in Mathematics and Computing from Indian Institute of Technology (IIT BHU) Varanasi. His primary research interests in computer vision, multi modal research, and diffusion modelling.

Yogendra Rao Musunuri received his B. Tech degree in electronics and communication engineering from the affiliated college (DPREC) of JNTU, Hyderabad, India in 2012, and M.E degree in image processing and computer vision from the BUFS, Busan, South Korea in 2017. Currently, he is a doctoral student, pursuing a major in the Control and Instrumentation Engineering, from Changwon National University, South Korea. His work has focused on computer vision, NLP, and multi modal architectures.

Oh-seol Kwon received his B.S. and M.S. degrees in Electrical Engineering & Computer Science from Kyungpook National University, Republic of Korea in 2002 and 2004, respectively and Ph. D. degree in Electronics from the same university in 2008. From 2010 to 2011, he was a senior researcher with the Visual Display Division, Samsung Electronics, Korea. He joined Changwon National University in 2011, and is currently a Professor. He focused on signal processing, network pruning, language, and multi modal architectures.