Project B: Vision Language Models

ECE1512 Digital Image Processing and Applications
Siyu Zhang
1010403653
Date due: December 10, 2024
Date handed in: December 10, 2024

Part 3 Summarization

Key points for VILA [5]

The mainstream multimodal large models typically follow two key stages: pre-training and fine-tuning. Pre-training serves as the initial training phase, providing foundational knowledge and enable models to handle diverse tasks. Fine-tuning is the second stage, and it helps the model focus on specific areas or tasks, adapting its broad capabilities to achieve high performance in targeted applications. Building on this framework, different models use their unique ways to solve specific tasks. For example, large language models (LLMs), as foundational models, excel in pre-training stages and perform exceptionally well on natural-language tasks. Vision Language Models (VLMs) are built on LLMs by adjusting the pre-training and fine-tuning stages to focus on visual elements. In-Context Learning (ICL) is defined as ability of model to perform tasks when presented with a few examples without additional fine-tuning, and this is the capability that VLMs aim to expand. Visual Language model (VILA) is a special type of VLM that improves learning from examples and reasoning across different types of input through careful pre-training. By using a large amount of mixed image-text data, VILA successfully connects visual and text information, and shows advanced abilities like understanding multiple images and visual chain-of-thought (CoT) reasoning, even though it mainly trains on single-image tasks. With fine-tuning, it further helps VILA preserve the core of LLMs. VILA demonstrates significant improvements over state-of-the-art VLMs on visual language tasks and it also shows that it can effectively combine the broad capabilities of language models with strong visual understanding.

Technical contributions

VILA is proposed after practical experiments on VLM models. The three main experiments applied to VLMs. First one is examining the effect of freezing LLMs during pre-training, second one is using interleaved pre-training data, and re-blending text-only instruction data with image-text data during instruction fine-tuning. Based on these experiments, the results indicate that freezing LLMs during pre-training is not optimal. Instead, fine-tuning the entire LLM during pre-training is crucial for developing strong in-context learning capabilities while also maintaining zero-shot performance. Additionally, the experiments show the importance of interleaved pre-training data. This data helps the model maintain its text abilities while getting better at visual-language tasks because it can connect visual and text information better. It enhances the model abilities of learning from examples. The third experiment solves two problems together, first it prevents the declined text processing during pre-training and also improves performance on visual-language tasks. By training on both visual and textual data simultaneously, the model develops strong capabilities in processing both types of information effectively. These findings and experimental results led to the development of VILA, which consistently outperforms existing state-of-the-art models across various benchmarks while maintaining practical abilities.

The diagram for the first experiment is shown in Fig.1.[1] It shows how does the AI system combines vision and language processing. It starts by processing a visual input, like a cat image. Then pass two main parts: a Vision Transformer (ViT) and a Projector. These components turn the image into visual tokens (pink boxes), which work together with text tokens (blue boxes) to help the Large Language Model (LLM) generate accurate text descriptions, like "a cat." On the right side,

it shows that the system is built in three steps. First is setting up the Projector, training the LLM to improve its understanding, and fine-tuning the model to make sure the text output is clear and accurate. Symbols like fire and snowflakes in the diagram show differences in computational effort or model size. Overall, this system connects visual input with language generation and it improves AI understanding and description the world.

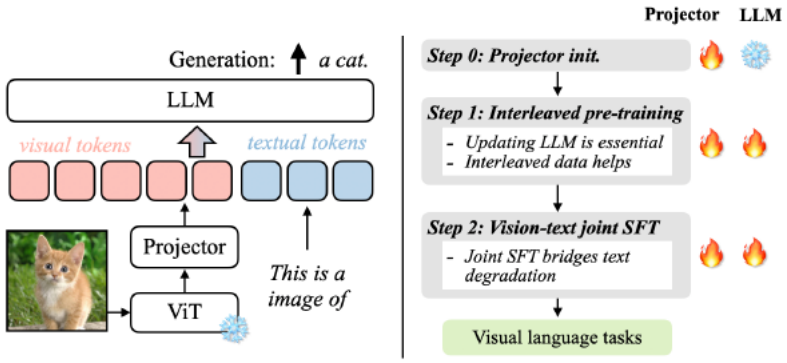

Fig.1 First important experiment in VILA[1]

Areas for improvement

The paper mentions several key areas where visual language models (VLMs) can be improved. The first area concerns the training data. Currently, the model uses two datasets, Multimodal C4 (MMC4) and COYO, but using more varied and larger datasets could help the model learn better and handle more real-world situations. Adding data in different languages or from specific fields could also make the model more useful. The second area focuses on token compression. The current method does not use token compression, which means it takes a lot of computing power to process visual information. Using better compression methods could save resources while keeping or even improving accuracy. The third area involves making training more efficient, like using sample packing. Methods such as combining multiple training samples can reduce training time and use fewer resources, making it easier to develop models. Also, using much larger dataset or trying different combinations of efficiency methods could help these models perform even better.

Part 4 Vision Language Models (VLMs)

Introduction

Token compression is a widely recognized method for improving model efficiency. In this project, token pruning is implemented as a key technique to achieve this goal. Token pruning evaluates the importance of tokens (small pieces of information) derived from images and retains only the most significant ones. The optimized model includes a scoring system that ranks tokens based on their importance. By keeping the highest-ranked tokens and discarding the rest, the model reduces the amount of information it processes, leading to faster and more efficient computations.

The dataset used in this project is Flickr8k, which consists of a collection of images paired with their corresponding captions. There are two models used for comparison. The first one is a baseline vision-language model and the second one is an optimized model enhanced with token pruning. The

results demonstrate that the optimized model significantly reduces both training and testing times compared to the baseline while achieving higher accuracy. To compare model complexities, Floating Point Operations Per Second (FLOPS) are calculated, further validating the efficiency improvements introduced by token pruning.

This approach demonstrates a significant improvement in computational efficiency through token pruning, achieving a notable reduction in both training time (from 714.00s to 54.70s) and FLOPS (from 83494912 to 78807104). However, this comes with a considerable trade-off in model performance, as evidenced by the increase in test loss from 0.0007 to 0.5050. While the method shows promise as a viable approach for enhancing vision-language models, particularly in resource-constrained environments, it requires further evaluation across a wider range of datasets and tasks to validate its adaptability and reliability in various real-world scenarios.

Dataset

The dataset used in this project is the Flickr8k dataset [3], which has 8,000 images, each paired with five captions describing the image. The dataset is split into 6,000 images for training and 2,000 for testing. The image captions give helpful text details about each image, which helps models better connect and understand both the images and their matching text descriptions. Images are resized to 112x112 pixels to prevent memory overwhelming. Captions are converted into sequences of words for processing. The sample of images is shown in Fig.2 in results.

Method

The dataset is quite large and requires a long time for processing, since that in the set-up part, a random sampling mechanism is added and is set to be 0.4, enable a faster experiment. For this project, the baseline model processes images by dividing them into small patches, turning these patches into tokens, and embedding them into a feature space. These tokens are passed through a feature projection layer and pooled to create a single feature vector representing the entire image. The baseline model uses a linear classification layer with normalization to transform the pooled features into predictions. It focuses solely on processing images and does not incorporate any textual information. This approach provides a straightforward pipeline for image feature extraction and serves as a reference point for comparison.

The optimized model extends this design by introducing a multimodal architecture that processes both images and captions. Similar to the baseline, it divides images into patches and embeds them as tokens. However, the optimized model implements a scoring mechanism that evaluates the importance of each token, retaining only the most important ones, thus reducing the computational load. At the same time, captions are processed through an embedding layer that converts text into a feature space similar to the image tokens. The model aligns these image and text features using a bilinear alignment layer, which measures their compatibility. The final output is an alignment score, indicating the semantic relationship between images and captions.

Both models are trained using the Adam optimizer and the number of epochs is set to be 3 and the batch size is 16, which is for overcome memory overwhelming. For the baseline model, its pooled outputs are used as features, while the optimized model compares its alignment scores to a cosine

similarity-based target. The models are evaluated based on training time, testing time, floating-point operations (FLOPs), and a metric called lossy-ness. This metric quantifies the trade-off between accuracy and efficiency improvements achieved by the optimized model. The general equation for lossy-ness is provided in Eq.6.

$$Lossyness = 1 - \frac{Optimized\ Accuracy}{Baseline\ Accuracy} \quad \text{Eq.6}$$

This approach demonstrates how the optimized model with token pruning and multimodal alignment can improve efficiency and performance. The structures of two models are show in Fig.3.

Besides the token pruning, sparse attention is also can be used to limit computation to a subset of tokens. Since it computed interactions between tokens, focusing only on pairs that are likely to contribute meaningful information.

Discussion

The detailed result is shown in Fig.4. The evaluation highlights significant differences between the baseline and optimized models regarding training efficiency and performance. The baseline model exhibits a training time of 714.00s with a FLOPS value of 83494912. In contrast, the optimized model achieves a reduced training time of 54.70s and a slightly lower FLOPS count of 78807104, showing that the optimized model significantly reduces computational time while slightly reducing model complexity. However, the training loss for the baseline model is much smaller compared to the optimized model, where the final loss in the optimized model is around 0.5275 compared to the baseline's final loss of 0.0011. The test loss shows a similar pattern, with the baseline achieving 0.0007 versus the optimized model's 0.5050. The one of the reasons of the not good accuracy is also due to the random sample ratio at the beginning. This indicates a clear trade-off between computational efficiency and model performance due to the optimization techniques applied.

Based on the experimental results, several improvements could help enhance both computational efficiency and model performance. The significant difference in loss values between the baseline, 0.0007 and optimized model, 0.5050 suggests that exploring alternative loss functions, including custom ones that better balance performance and efficiency, could be beneficial. The optimization currently achieves only a modest reduction in FLOPS (from 83494912 to 78807104) while significantly impacting accuracy, indicating that different pruning strategies, quantization techniques, or hybrid approaches might yield better results. For future work, the number of epochs should be increased to allow the optimized model more time to converge, and the hyperparameters in the training loop should be tuned more extensively. Also, if possible, it would be better to use the whole dataset without splitting to maximize training data. Besides token pruning, sparse attention should also be tested since it focuses on parts that have strong relationships, making it particularly effective for handling both images and text data.

Conclusion

This project investigated token pruning's effectiveness in vision-language models using the Flickr8k dataset, comparing a baseline model with an optimized version. The results showed significant

improvements in computational efficiency, with the optimized model reducing training time from 714.00s to 54.70s and FLOPS from 83494912 to 78807104. However, this came with a notable trade-off in performance, as the test loss increased from 0.0007 to 0.5050. While token pruning successfully reduced computational demands, the impact on model performance suggests that further refinements are needed. Future work should focus on exploring alternative loss functions, implementing more sophisticated pruning strategies, and investigating hybrid approaches that better preserve model accuracy while maintaining efficiency gains. These findings can help to develop vision-language models more and highlight the importance of balancing computational efficiency with model performance.

Result

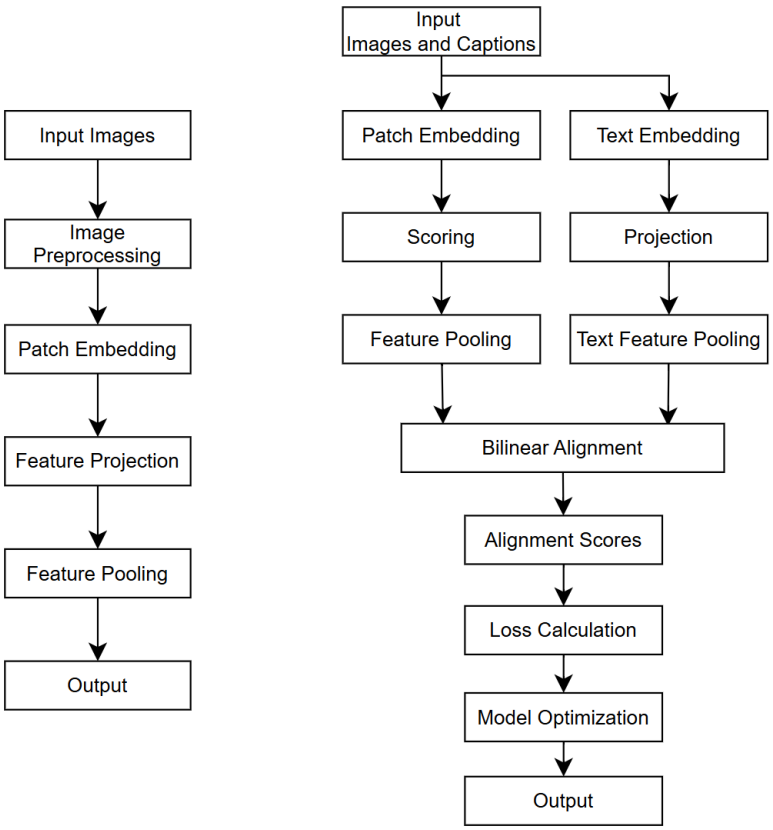

Fig.2 Flow charts of the baseline model and efficiency improved model

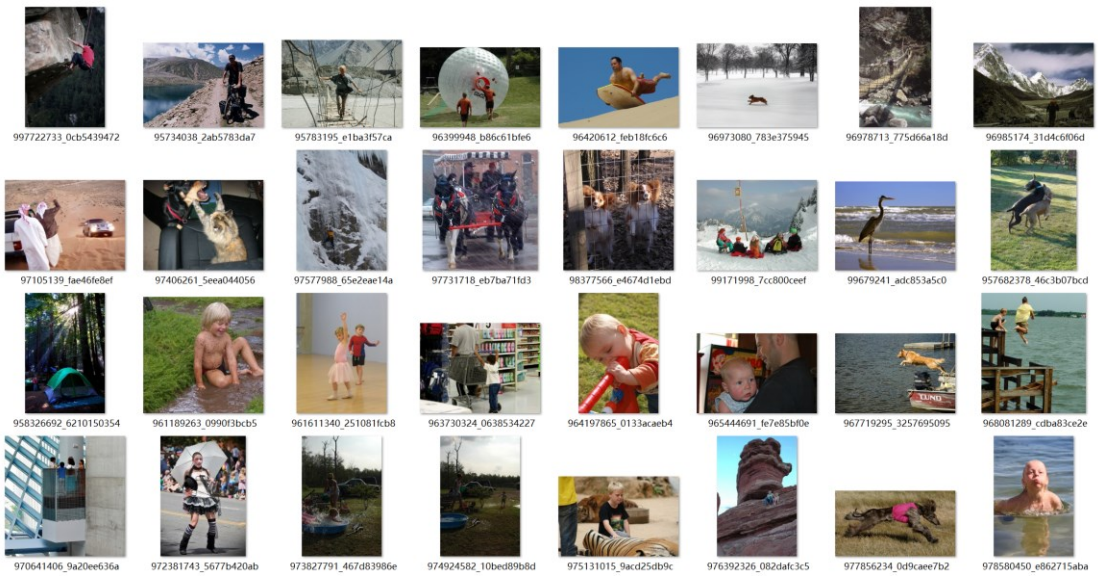

Fig.3 Example Images from the Flickr8k Dataset with Associated Identifier

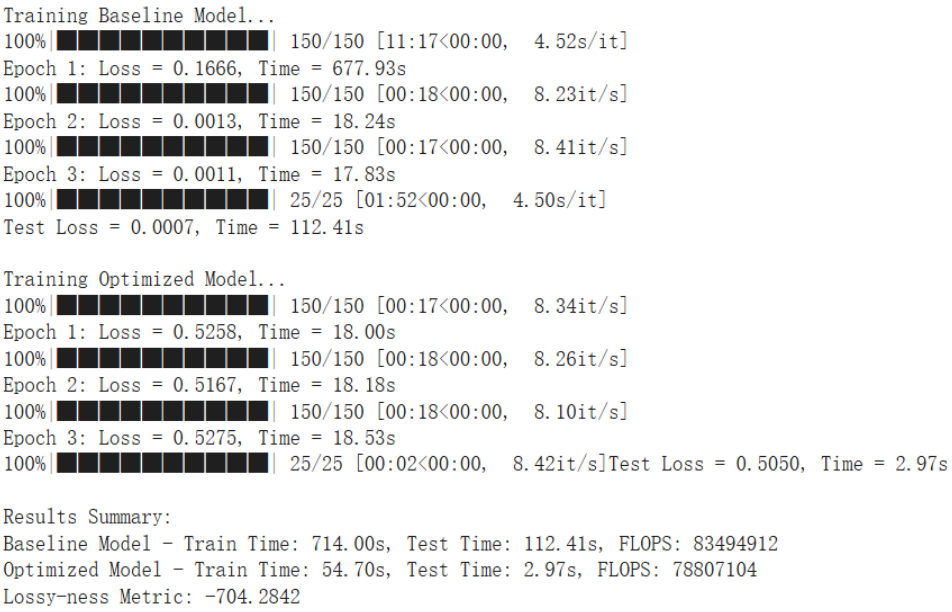

Fig.4 Results of baseline model and optimized model

Codes

https://github.com/Alicesyz/ECE1512_2024_ProjectB_SSMs_VLMs_SiyuZhang

Appendix

[1] Ji Lin, Hongxu Yin, Wei Ping, Yao Lu, Pavlo Molchanov, Andrew Tao, Huizi Mao, Jan Kautz, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models, 2023.

[2] Tang, H., Gao, B., Yang, W., Xie, H., & Li, S. (2023). Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

[3] Wang, Y., Huang, Z., Xia, F., & Lin, D. (2022). TokenCut: Segmenting Objects in Images and Videos with Self-supervised Transformer and Token Labeling. *arXiv preprint arXiv:2204.08721.*

[4] Hodosh, M., Young, P., & Hockenmaier, J. (2013). Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research, 47*, 853–899. https://doi.org/10.1613/jair.3994

[5] Author Unknown. (2024). 如何通过 Pandas 实现数据筛选与分析？CSDN 博客. CSDN. Retrieved December 10, 2024, from https://blog.csdn.net/m0_59164304/article/details/143239049

[6] ECE1512_Project_B_Details

[7] ECE1512_AssignmentB