A visual question answering deep learning model

Ahmad Hosseini, Data Science, ZHAW School of engineering

Abstract

This report investigates the efficacy of combining a Visual Transformer (VIT) architecture, incorporating convolution, with a Text Transformer (BERT) framework to construct a multimodal Visual Question Answering (VQA) model. The VIT model demonstrates exceptional capabilities in processing visual information by leveraging self-attention mechanisms, while BERT excels in natural language understanding tasks. By fusing these two architectures, This work aims to harness the complementary strengths of both approaches, enabling the model to effectively comprehend and respond to questions related to visual content. Through experiments and evaluations on VQAv2 dataset, this work analyzes the performance of the proposed multimodal model.

1 Introduction

A multimodal model was developed to handle the VQAv2 dataset, which involves answering questions based on pairs of images and questions. The objective was to create a model capable of processing both images and text as inputs. The VQAv2 dataset used in this study was obtained from Hugging Face, specifically from the following link: https://huggingface.co/datasets/HuggingFaceM4/VQAv2.

The annotated answers in the VQAv2 dataset are originally in free-form natural language. However, it is a common practice to convert this task into a classification problem with 3,129 answer classes. Accordingly, the ViLT-B/32 model was fine-tuned on the train and validation sets of the VQAv2 dataset.

By utilizing this multimodal model and fine-tuning it on the relevant dataset subsets, the goal was to enable the model to process visual information and effectively provide answers to questions formulated in natural language. This approach aimed to address the VQAv2 task by leveraging the capabilities of the ViLT-B/32 model and adapting it to the specific requirements of the dataset.

2 Model

The proposed multimodal model, as illustrated in Figure 1, incorporates both BERT and VIT embeddings for visual question answering. The model architecture involves concatenating the embeddings generated by BERT and VIT. More specifically final encoding vector of the [CLS] token of BERT and VIT was used as the representation of the text and image respectively. These concatenated embeddings are then passed through a fusion layer, which aims to reduce the dimensionality while preserving the relevant information. The fusion layer produces a vector of length 512.

On top of the fused embeddings, a linear layer with a size of 3129 is added. This linear layer serves as a classifier for the 3129 answer labels in the VQAv2 dataset. By applying this classifier, the model can predict the most suitable answer given the visual and textual inputs.

Overall, the proposed model leverages the power of both BERT and VIT embeddings, combining their strengths in processing textual and visual information. The fusion layer and the subsequent linear layer enable the model to generate predictions for the wide range of answer classes in the VQAv2 task.

3 Related Work

The paper Kim et al. (2021) presents a novel approach to Vision-and-Language Pre-training (VLP) models. The authors address the limitations of existing VLP models that heavily rely on image feature extraction processes, such as object detection, and convolutional architectures like ResNet. They argue that these processes are computationally expensive and have limited expressive power.

The authors propose a minimal VLP model called Vision-and-Language Transformer (ViLT) that simplifies the processing of visual inputs to a convolution-free manner, similar to how textual in-

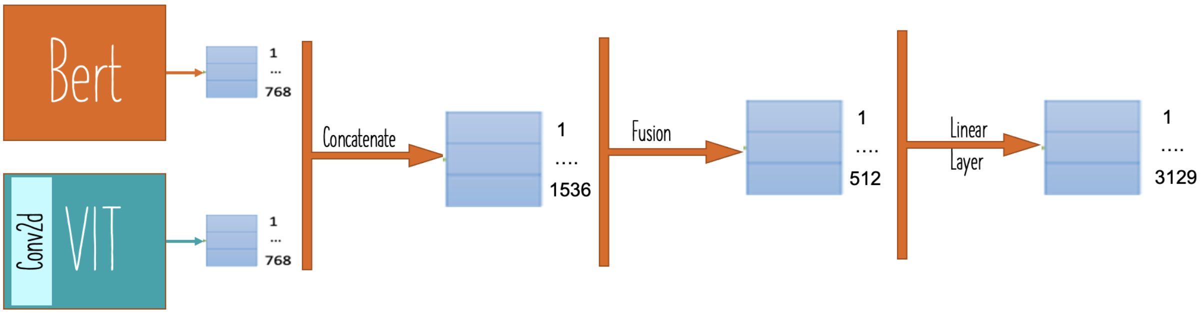

Figure 1: The proposed Multimodal Model

puts are processed. By removing convolutional neural networks from the VLP pipeline, ViLT achieves significantly faster processing times while maintaining competitive or better performance on downstream tasks. In their paper, the authors fine-tune the ViLT-B/32 model on the VQAv2 dataset. The VQAv2 task is described in the Introduction section of. The authors fine-tune the ViLT-B/32 model on the train and validation sets of the VQAv2 dataset. They reserve 1,000 validation images and their related questions for internal validation.

The authors report the test-dev score results from their submission to the evaluation server. They achieve a 71.26 ± 0.06 result on the VQAv2 evaluation metric on the test-dev set. The VQA score is calculated by comparing the inferred answer to 10 ground-truth answers, and more details about the evaluation can be found at the following link: https://visualqa.org/evaluation.html.

4 Experiment

The proposed model in Figure 1 was trained on the VQAv2 dataset using specific data splits as shown in Figure 2. For training, only the train and validation subsets of the dataset available from HuggingFace were utilized. The training process was conducted using the PyTorch Lightning module with a single GPU. The model underwent training for three epochs, employing an Adam optimizer with a learning rate of 5e-5.

During the training, the model's performance was monitored, and the best accuracy achieved throughout a single epoch was saved. The evaluation of the model's accuracy was based on the commonly used accuracy metric provided by the Scikit-learn library, rather than the accuracy metric defined by the evaluation server.

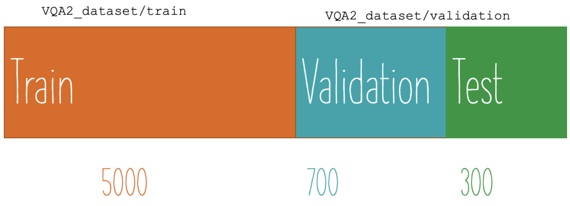

Figure 2: The train validation test split

5 Results

The obtained results on the test split, as depicted in Figure 2, showed an accuracy of 0.23. Throughout the training process, an interesting pattern was observed. Initially, the training loss increased, but subsequently, it started to decrease steadily. On the other hand, the validation accuracy exhibited a decreasing trend.

These observations suggest that the model initially struggled to capture the underlying patterns in the training data, resulting in an increase in the training loss. However, as the training progressed, the model gradually improved its performance and learned to better fit the training data, leading to a decrease in the training loss. Despite this improvement, the model's generalization ability seemed to be limited, as reflected by the decreasing validation accuracy.

These findings indicate that further investigation and potential adjustments to the model architecture or training process may be necessary to enhance its performance and address the issue of decreasing validation accuracy.

6 Conclusion

Examining the results of previous papers that have conducted experiments on the VQAv2 dataset (referenced at https://paperswithcode.com/sota/visual-question-answering-on-vqa-v2-test-dev), it is evident that some papers have achieved impressive accuracies, reaching up to 84.3 percent on the test-dev set evaluated on visualqa.org. It is important to note that the accuracy metric employed on the evaluation server differs slightly from the one used in my experiment, making a direct quantitative comparison challenging. However, this metric can be utilized to provide a qualitative indication of the gap between my model and the state-of-the-art models.

In future work, it would be valuable to employ the evaluation metric implemented by the evaluation server to report the results accurately. Additionally, considering the potential differences in evaluation metrics, directly utilizing the evaluation server for evaluating the model's performance could provide more reliable and directly comparable results.

6.1 Outlook

Based on the observations of increasing training loss and decreasing validation accuracy, several investigation and potential adjustments to the model architecture or training process could be considered:

- Model Architecture Modifications: Explore modifications to the model architecture to enhance its performance. This could involve experimenting with different fusion techniques, introducing additional layers or modules, or adjusting the dimensionality of the embeddings to improve the model's representation learning capabilities.

- Regularization Techniques: Apply regularization techniques to prevent overfitting and improve generalization. Techniques such as dropout, weight decay, or batch normalization could be incorporated into the model to reduce the gap between training and validation performance.

- Learning Rate Adjustment: Experiment with different learning rates during training. Gradually increasing or decreasing the learning rate, or implementing learning rate schedules (e.g., cosine annealing, learning rate decay), might help the model converge more effectively and improve generalization.

- Hyperparameter Tuning: Perform a systematic hyperparameter search to identify optimal values for various hyperparameters such as batch size, optimizer parameters, or the number of layers in the model. This could involve using techniques like grid search, random search, or Bayesian optimization to find the best combination of hyperparameters for improved performance.

References

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision.

A Appendix

The first two hyperlinks lead to the polls we conducted:

- Link to the jupyter notebook of the code in public github repository.