DreamBooth Stable Diffusion with Low-Rank Adaptations

Monil Lodha

November 6, 2024

1 Introduction

This report documents the process of incorporating Low Rank Adapttion(LoRA) into the Stable Diffusion model used in Dreambooth.DreamBooth allows for the training of unique, personalized visual concepts, making it possible to generate images that closely resemble or incorporate new subjects

2 Pretrained Model Used

I have used the same Stable Diffusion v1-4 model available freely on the web as CompVis/stable-diffusion-v1-4.

3 Dataset

The dataset I have used is one of the subjects Dog from the hugging face diffusers Dataset Dogs. The dataset used for fine-tuning consists of images downloaded from the URLs provided in the code. The images represent a small dog, which is the new concept being introduced to the model. A total of 5 images are used.

4 Implementation Details

The implementation is based on the Hugging Face Diffusers library. The Stable Diffusion v1-4 model is used as the base model.Extending the code from the original project:

- Install peft library along with the other ones.
- Setting up the `DreamBoothDataset` and `PromptDataset` classes.
- Generating class images (if prior preservation is enabled).
- Loading the Stable Diffusion model.
- Setting up training arguments.
- In the training function, we made the following changes:-
  - Defined proper LoRA configurations by checking on what layers(mainly conv2D and linear) of unet and text encoder can it be applied
  - Added lora adapters to both the models

  - Defined functions to unwrap a peft model to be able to actually get the lora layers and then defined 2 functions to save and load the lora layers as and when required.
  - At every checkpoint and at the end of training we save just the LoRA layer weights not the whole model's parameters resulting in very efficient memory and space management.
- Running the training process using `accelerate.notebook_launcher`.
- Install CLIP and import a CLIPTextModel to calculate CLIP score for the prompts.
- Setting up the pipeline for inference :-
  - First load the basic pretrained model.
  - Load the LoRA weight layers to the pretrained and output the final model ready for inference
- Running the Stable Diffusion pipeline to generate images using various prompts alongside calculating their CLIPScore.

Key hyperparameters used in the training process include a learning rate of 5e-06, a maximum of 100 training steps, a batch size of 2, and gradient accumulation steps of 2.

5 Testing and Results

The results of the fine-tuning process are presented in the form of images generated by the model.The images are generated based on a list of prompts with our identifier being [sks] dog, including:

- a sks dog in the jungle
- a sks dog in the snow
- a sks dog on the beach
- a sks dog on a cobblestone street
- a sks dog on top of pink fabric
- a sks dog on top of a wooden floor

All the outputs are available in the Outputs directory. The final ipynb submitted contains the run for an alarm clock. To show that context is indeed being learned, we tried these 2 prompts :-

Prompt1: A sks dog


in the snow


on a cobblestone street


on a wooden floor

Prompt2: A dog


in the snow


on a cobblestone street


on a wooden floor

6 Conclusion and Future Works

LoRA integration significantly reduces training costs while maintaining high-quality output, making DreamBooth more efficient and practical. Possible Future Works are :-

- **Fine-Tuning for Broader Applications**: Extend DreamBooth's LoRA-based fine-tuning for applications like video generation, style transfer, and domain adaptation, where efficiency and memory are critical.
- **Evaluate Generalization on Diverse Datasets**: Test the LoRA-enhanced model across a wide range of datasets to analyze its adaptability to various subjects, scenes, and styles.
- **Extending LoRA to Multi-Concept Learning**: Investigate how LoRA can be adapted for multi-concept fine-tuning, where the model learns multiple subjects or styles simultaneously without losing fidelity.

References

[1] Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, L., & Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models. *arXiv preprint arXiv:2106.09685*. Retrieved from `https://arxiv.org/abs/2106.09685`

[2] Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., & Freeman, W. T. (2022). DreamBooth: Fine Tuning Text-to-Image Diffusion Models for Subject-Driven Generation. *arXiv preprint arXiv:2208.12242*. Retrieved from `https://arxiv.org/abs/2208.12242`