FINAL-REPORT: Stable diffusion customizing

SongSeungWu

1 What is Stable Diffusion Customizing?

Stable Diffusion Customizing refers to the process of adapting a pre-trained Stable Diffusion model to specific tasks or datasets.
Steps

- Data Preparation: Preparing a custom dataset that the model will be trained on. (ex: a dataset comprising images of a specific style and their corresponding captions can be used.)
- Model Configuration: Loading and setting up the core components of Stable Diffusion, including the Variational Autoencoder, UNet, and Noise Scheduler.
- Using LoRA: Applying LoRA to specific layers of the model to enable efficient learning of additional parameters.
- Training and Fine-Tuning: Adjusting the model using the custom dataset. Techniques such as data preprocessing, setting training parameters, and Gradient Accumulation are utilized during this step.
- Result Analysis and Application: Utilizing the fine-tuned model to generate images in specific styles, conduct experiments, or integrate it into other applications.

Stable Diffusion Customizing focuses on tailoring large pre-trained models to meet specific needs, maximizing performance, and producing outputs suitable for targeted use cases.

2 Introduction

The goal of this lab is to learn how to customize the Stable Diffusion model to generate images in a desired style. Stable Diffusion is a pre-trained large-scale model known for its high performance in image generation. By leveraging the Low-Rank Adaptation technique, we efficiently train specific parameters of the model.
Steps

- Collecting and Define Datasets
- Configuring the model
- Training Process
- Inference and Results

3 What is LoRA?

LoRA is a technique designed to efficiently train specific parameters of large pre-trained models for new tasks or domains without retraining the entire model.

- Efficient Parameter Training: LoRA decomposes the original weight matrix of the model into low-rank matrices, significantly reducing the number of trainable parameters.
- Preservation of Pre-trained Weights: Instead of altering the original model weights, LoRA adds separate adaptation layers that handle the training.
- Scalability and Flexibility: It is widely used across domains such as image generation, natural language processing, and speech recognition.

LoRA was applied to the Attention Layer of the UNet in the Stable Diffusion model to customize it for the "8bit illustration" style.

4 Collecting and Define Datasets

To tailor the Stable Diffusion model for a specific style, a custom dataset was prepared.

- Image Collection: Bing Image Downloader was used to crawl 100 images with the keyword "8bit illustration." The collected images were saved in the gen_ai_custom_dataset/train/ directory.
- Image Preprocessing: All images were converted to RGB format using PIL and resized to match the model's required resolution.
- Caption Generation: Captions for each image were generated using the BLIP model. These captions were used as training data along with the images.
- Metadata Storage: The generated captions were saved as a metadata.csv file using pandas DataFrame. The file includes the image names and their corresponding captions.

5 Configuring the model

Model Configuration
The components of Stable Diffusion were configured and extended using the LoRA technique.

- VAE: Used to represent latent space stably.
- UNet: The core architecture responsible for image generation. LoRA layers were added to the Attention Layer for efficient training.
- Noise Scheduler: Controls the magnitude of noise at each timestep in the diffusion process.

6 Training Process

- Preprocessing Training Data: Data augmentation techniques such as resizing, center cropping, and random horizontal flipping were applied.
- LoRA Configuration: LoRA was applied to the Attention Layer of UNet, and only the added parameters were set to be trainable.
- Parallel Training with Accelerate: The Accelerate library was used to train the model in parallel.
- Loss Function and Learning Rate Scheduling: MSE was used as the loss function, and cosine decay was applied for learning rate scheduling.
- Checkpointing: Checkpoints were saved every 5,000 steps to allow for resumption in case of interruptions.

7 Inference and Results

- The trained LoRA weights were loaded into the Stable Diffusion pipeline for image generation.
- Example Prompt: "8bit illustration of city view"
- The NSFW checker was disabled to prevent interruptions during image generation.
- Result Images: A total of 3 images were generated and visualized using make_image_grid.

- Some generated images showed distortions, which can be improved through enhanced data diversity and further parameter tuning.

8 Conclusion

The Stable Diffusion model was successfully customized to generate images in a specific style. LoRA effectively enhanced parameter learning efficiency, and the importance of custom datasets was demonstrated.