Training DreamBooth LoRA with Stable Diffusion XL on Thumbs Up Images

Paresh Natarajan

I. Design of the Training Approach

The training approach utilized the DreamBooth technique, which involves fine-tuning a pre-trained Stable Diffusion model on a small set of subject-specific images. This approach leverages the Low-Rank Adaptation (LoRA) technique, which allows for efficient fine-tuning by introducing sparse trainable parameters. The training process involved the following steps:

1. Data Preparation: A dataset of Trump thumbs up images were taken and stored in a specified directory.

2. Model Selection: The `stabilityai/stable-diffusion-xl-base-1.0` model was chosen as the base model for fine-tuning, along with the `stabilityai/sdxl-vae` model for the variational autoencoder.

3. Hyperparameter Configuration: Various hyperparameters were set, including learning rate, batch size, gradient accumulation steps, and maximum training steps.

4.Training Setup: The training environment was set up using the Hugging Face Diffusers library, along with necessary dependencies such as TensorRT, bitsandbytes, xformers, and Weights & Biases (wandb) for logging and monitoring.

5. Training Process: The training was initiated using the `accelerate launch train_dreambooth_lora_sdxl.py` command, which fine-tuned the base model using the provided dataset and specified hyperparameters.

6. Evaluation: During training, the model was evaluated on a validation prompt at specified checkpoints to monitor its performance.

II. Results of the Evaluation

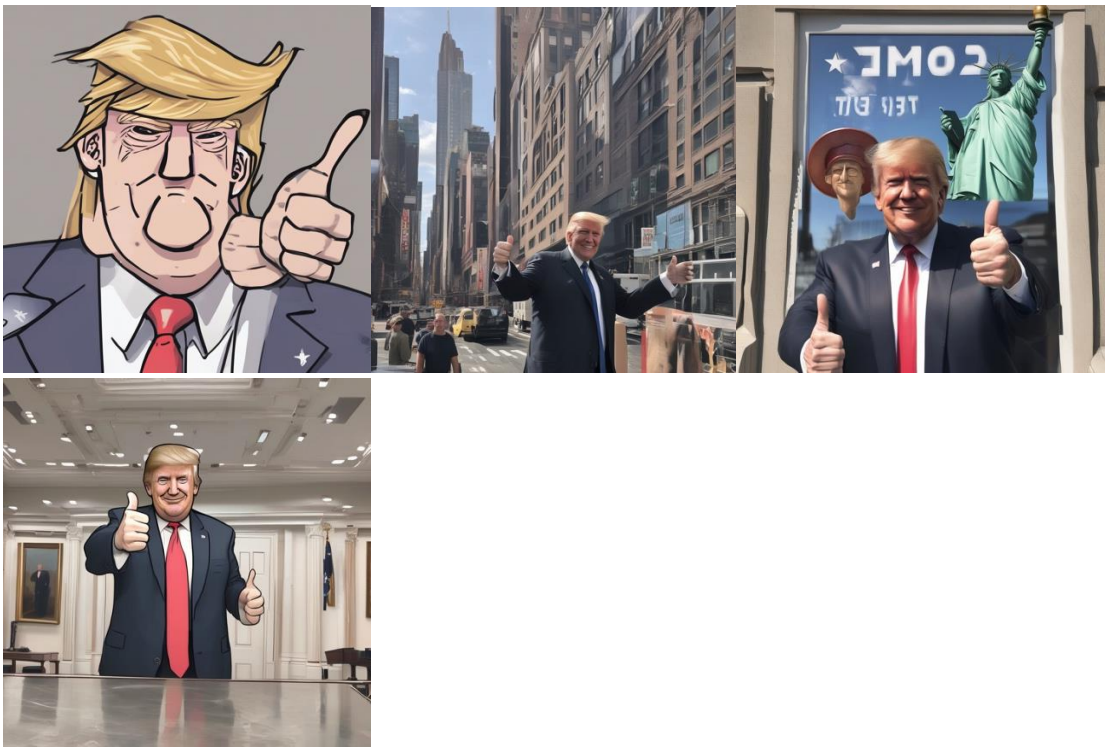
Two training runs were performed with different hyperparameter configurations:

1. Base Run:
   - Learning Rate: 1e-5
   - Max Train Steps: 100
   - Checkpoint: 20
   - Prior Loss Weight: 0.5
   - Num Class Images: 5
   - Validation Epochs: 10

2. Improved Run:
   - Learning Rate: 2e-4
   - Max Train Steps: 500
   - Checkpoint: 100
   - Prior Loss Weight: 0.8
   - Num Class Images: 10
   - Validation Epochs: 15

The improved run with higher learning rate, more training steps, and a larger number of class images resulted in better performance, as evaluated by the generated images below.

Output – Base:



Output – Improved:

III. Logging and Deployment

1. Logging and Monitoring: All training data and metrics were logged using Weights & Biases (Wandb) for monitoring and visualization purposes.

2. Inference Infrastructure: A serverless inference environment was created within the Hugging Face platform, leveraging their infrastructure for serving the fine-tuned model.

3. API Server:

An API server was created to expose the fine-tuned model as a service. The server includes an image generation endpoint (/generate_image) that accepts POST requests with prompts for generating Trump thumbs up images. Users can send a prompt such as "a high-quality photo of Trump showing thumbs up" to the /generate_image endpoint to generate the corresponding image. The server implements API key authentication, requiring users to provide an authorized API key (API_KEY) to access the /generate_image endpoint. Unauthorized requests to the endpoint are rejected.

To track API usage, the server includes an /api_key_usage endpoint that allows retrieving the usage count for each API key. Additionally, the server integrates with Postman, enabling easy testing and monitoring of the image generation process. The /generate_image endpoint can be triggered with specified trigger keywords and styles to generate images. The server maintains a count of successful image generations per API key, facilitating usage monitoring and cost optimization.

IV. Future Considerations

1. DevOps Strategies:

Adopting DevOps practices and implementing a CI/CD pipeline can streamline the deployment process for the Stable Diffusion XL model fine-tuned with the DreamBooth LoRA technique. This would involve automating the model training and validation processes, leveraging containerization technologies for consistent deployments across environments, enabling

automated deployment and rollback mechanisms, and integrating with Weights & Biases (Wandb) for seamless integration with the CI/CD pipeline.

2. Monitoring and Logging:

Integrating advanced monitoring and logging tools is crucial for effective management and optimization of the deployed Stable Diffusion XL model. This includes leveraging tools like Prometheus and Grafana for performance monitoring and visualization, implementing centralized logging and log analysis for troubleshooting and optimization, correlating model performance with training metrics logged in Wandb, setting up alerting and anomaly detection mechanisms, and tracking usage patterns and API key usage for cost optimization and capacity planning.