Task-1

By

Shashank Gupta

📘 From Language to Light: A LoRA Bootcamp for Text-to-Vision Learning

---

☑ Phase 1: LoRA Basics – Text Fine-tuning

🎯 Objective

Implement and evaluate **Low-Rank Adaptation (LoRA)** on a text classification task, and compare its performance with traditional full fine-tuning.

📝 Task

- Fine-tuned `distilbert-base-uncased` for **IMDb sentiment classification**
- Compared full vs LoRA tuning in terms of accuracy, efficiency, and model size

---

📌 What is LoRA?

LoRA (Low-Rank Adaptation) is a parameter-efficient fine-tuning (PEFT) method. It introduces small trainable adapter layers and keeps the rest of the model frozen.

- Trains only a fraction of total parameters
- Reduces memory and compute cost
- Speeds up training
- Adapter weights are lightweight and shareable

📊 Results Summary

Metric	Full Fine-Tuning	LoRA Fine-Tuning	Key Insight
Accuracy	87.5%	87.0%	Comparable performance
Training Time	~22.6 minutes (1359s)	~19 minutes (1143s)	~15% faster
Trainable Parameters	~67M (100%)	~740K (1.09%)	Huge reduction
Model Size	~256 MB	~3 MB (adapter only)	Tiny adapter

🧪 How to Run

```
pip install transformers==4.40.0 datasets==3.6.0 peft==0.10.0 accelerate torch wandb
numpy
gurukul-lora-env\Scripts\activate
python LoRA_Text/train_distilbert_full_vs_lora.py
```

📂 Outputs:

- `./results/`
- `./logs/`
- `./LoRA_Text/`
- `./wandb/`

📷 [Comparison Chart of Accuracy & Training Time] :

☑ Phase 2: LoRA for Vision – Image Fine-tuning with Stable Diffusion

🎯 Objective

Use LoRA to fine-tune **Stable Diffusion v1.4** and generate stylized Gurukul-themed images.

📝 Task

- Trained LoRA adapter on **10 themed images**
- Used adapter to generate images from prompts
- Focused on reducing parameter update size and memory use

⚙️ Technical Insights

Metric	Value	Insight
Trainable Parameters	297,984 (~0.035%)	Only adapters updated
Total Parameters	~859M	Core model untouched
Training Steps	10	Very low resource need
Adapter Size	~3 MB	Lightweight, reusable

📸 Sample Prompts

- “AI robot teaching students in a village school”
- “A traditional Gurukul reimagined with VR and AR”
- “Students learning under a hologram tree”

📷 [Generated Gurukul images]

1.A child meditating in a virtual garden

2. AI robot teaching students in a village school

3. Students learning under a hologram tree

🍼 How to Run

```
pip install diffusers transformers peft accelerate xformers
gurukul-lora-env\Scripts\activate
python LoRA_StableDiffusion/train_lora_sd.py
python LoRA_StableDiffusion/inference_generate.py
```

📂 Outputs:

- `./LoRA_StableDiffusion/lora_output/`
- `./LoRA_StableDiffusion/outputs/`
- `./training_data/gurukul_hologram_style/`

☑ Phase 3: Visual Sequencing – Image-to-Video Integration

🎯 Objective

Convert sequential prompts into images and merge them into animated video clips.

📝 Key Features

- Paragraph prompts are split and converted to images
- Scenes saved in separate folders: `frames_scene_X`
- Final videos saved in `video_outputs/`

🛠️ Tools Used

- Hugging Face Diffusers + Stable Diffusion
- Prompt Engineering
- Python automation (`os`, `pathlib`)
- **MoviePy** for frame stitching

📷 **[Insert Image Placeholder – Side-by-side frames from one story scene]**

📷 **[Insert Image Placeholder – Screenshot of final video output folder]**

📁 Project Structure Overview

Folder	Description
`LoRA_Text/`	Text classification with LoRA
`LoRA_StableDiffusion/`	SD LoRA training and inference scripts
`VideoMaker/`	Code for video creation from image sequences
`lora_output/`	Saved adapter models
`outputs/`	Generated images
`training_data/`	Custom Gurukul-themed dataset used for Phase 2

💡 Challenges Faced

- ⚠️ **Limited GPU VRAM**
- ⚠️ **xFormers / triton unavailable** during training
- ⚠️ Maintaining **object consistency across frames** is still tough

---

🔮 Future Scope

- Integrate `ControlNet` or `img2img` for smoother temporal transitions
- Use `Streamlit` or `Gradio` to build a Web UI
- Add scene tracking and frame-by-frame evolution
- Explore `AnimateDiff` for dynamic video generation

☑ Installation Steps

```
pip install -r requirements.txt
gurukul-lora-env\Scripts\activate
```
