SDXL DreamBooth 전이 학습

이미지로 변환하는 Stable Diffusion 모델을 파인 튜닝 하기

출처

Playing with SDXL Dreambooth (full workflow)

https://www.reddit.com/r/StableDiffusion/comments/14ylbqx/playing_with_sdxl_dreambooth_full_workflow/

Hi!

I'm playing with SDXL 0.9 dreambooth parameters to find how to get good results with few steps.

Here I attempted 1000 steps with a cosine 5e-5 learning rate and 12 pics. It took ~45 min and a bit more than 16GB vram on a 3090 (less vram might be possible with a batch size of 1 and gradient_accumulation_step=2)

It's still not perfect, as you can see the reference on last pic, it make my jaw more squared (not that I dislike it)

You only need the base model, not the refiner

Full workflow:

- Install torch 2 (use less vram)
- Download kohya-ss training repo and install deps

```
git clone https://github.com/kohya-ss/sd-scripts
cd sd-scripts
git checkout sdxl
pip install -r requirements.txt
pip install git+https://github.com/huggingface/diffusers
pip install xformers invisible-watermark
```

- Define your dataset in a file config_dataset.toml, class_token is "<learned_token><space><class>" with class being "man", "woman", "dog", "car" ...

```
[[datasets]]
keep_tokens = 2

  [[datasets.subsets]]
  image_dir = '/workspace/data/me'
  class_tokens = 'shs man'
  num_repeats = 10

[general]
resolution = 1024
```

I was not able to find how keep_tokens works or if it's even usefull, please let me know if you have a clue

- Define your training hyperparameters in a file config.toml

```
[model_arguments]
v2 = false
v_parameterization = false
pretrained_model_name_or_path = "./models/sd_xl_base_0.9.safetensors"

[optimizer_arguments]
optimizer_type = "adafactor"
optimizer_args = [ "scale_parameter=False", "relative_step=False", "warmup_init=False" ]

# use_8bit_adam = true
lr_scheduler = "cosine_with_restarts"
lr_warmup_steps = 100
learning_rate = 5e-5 #SDXL original learning rate: 4e-7
max_grad_norm = 0.0
train_text_encoder = true

[dataset_arguments]
debug_dataset = false
```

```
[training_arguments]
output_dir = "ComfyUI/models/checkpoints/"
output_name = "1000_lr1e-5_cosine"
save_precision = "fp32"
save_n_epoch_ratio = 1
save_state = false
train_batch_size = 2
mem_eff_attn = false
max_train_steps = 1000
max_data_loader_n_workers = 1
persistent_data_loader_workers = true
gradient_checkpointing = true
# gradient_accumulation_steps = 1
mixed_precision = "bf16"
logging_dir = "logs"
log_prefix = "last"
sample_prompts = "prompts.txt"
log_with = "wandb"
xformers = true
cache_latents = true
full_bf16 = true

[sample_prompt_arguments]
sample_every_n_steps = 10000 # cant make it work
sample_sampler = "euler a"

[saving_arguments]
save_model_as = "safetensors"
```

- Run the training command

```
accelerate launch sdxl_train.py \
    --config_file=config.toml\
    --dataset_config=config_dataset.toml \
```

- Sanity check: try your model with some short prompts

```
python sd-scripts/sdxl_minimal_inference.py --ckpt_path ComfyUI/models/checkpoints/1000_lr1e-5_cosine.safetensors --interactive --output_path=./out/
```

I felt that it was diffucult with long prompts to get good likeness (might be my model not trained enough)

"Picture of shs man", "shs man, close up" should get you a pic close to the training

"shs man oil painting" should get you a stylised version

- Load the model in comfy/auto to try longer prompts. You can now increase the token weight to get better likeness

```
beautiful shs man oil painting, masterpiece, trending on artstation, best quality, starry night's style
50 steps
dpm 2M karras
```

Let me know if you see any improvements