Assignment #6.2 – Diffusion MNIST

https://github.com/Lemorita95/1FA006/tree/main/6_diffusion-models/diffusion-mnist

Written summary

Also available in https://github.com/Lemorita95/1FA006/blob/main/6_diffusion-models/diffusion-mnist/README.md

approach

1. hyperparameters:

   batch_size = 128

   learning_rate = 4e-4

   num_epochs = 100

   diffusion time_steps = 1000

   diffusion sampling_timesteps = 250

2. Uses Unet and GaussianDifussion from denoising_diffusion_pytorch;
3. Load MNIST dataset as a pytorch DataLoader;
4. Train model with train_validate_model();
5. Generate image samples;

results

1. the training loop for the hyperparameters took around 7.5 minutes per epoch;
2. the training had convergence;
3. most of digits were easy to identify through human perception;
4. generating new data took around 0.5s/sample;

challenges

1. easier to implement training (compared to the simple diffusion) thanks to denoising_diffusion_pytorch;
2. laborous training, however the results were good, slightly better then compared to the GAN implemented;
3. did not implement tensorboard, it would be nice to see the image generation evolving;

Result plots

In this section, the training and validation losses and the images outputted by the trained neural network.

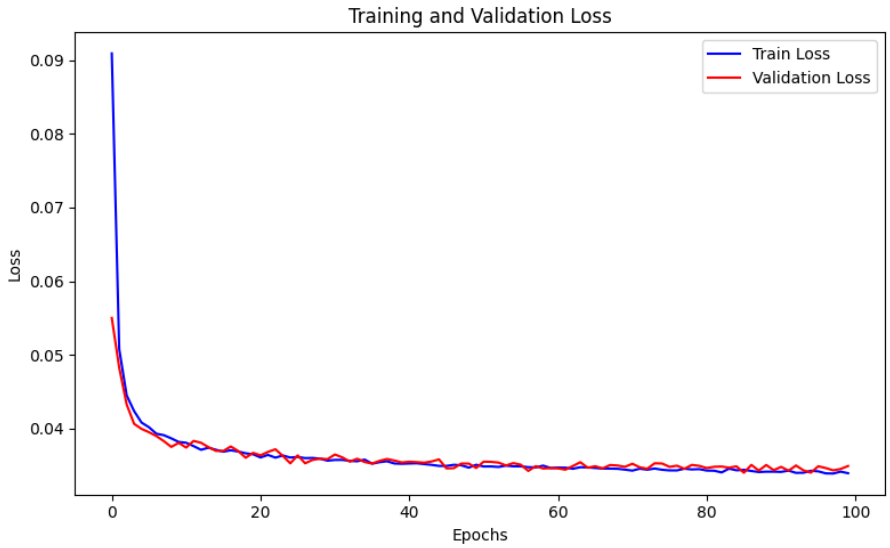

Figure 1: Train and validation losses.

Figure 1 shows a slow but steady loss reduction, however, considering a time of ~7.5 minutes/epoch might not justify the additional 20-25 epochs since after the 75th epoch, not much changed.

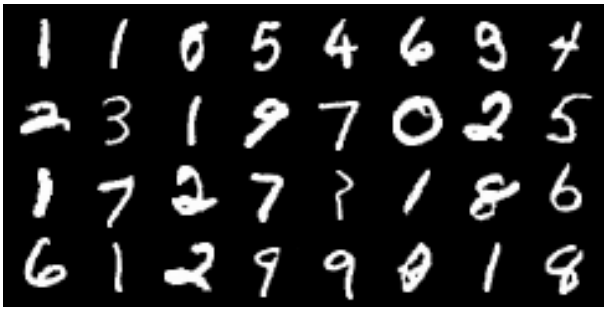

Figure 2: Generated 32 sample sfrom the trained model.

Figure 3: Generated 100 samples from trained model.

At Figure 2 and Figure 3, we see that most of the digits could be easily identified by a human.