HW4 Image Generation with GAN

计11 张凯文 2020080094

Project environment

```
Python 3.8
PyTorch = 1.1
torchvision
tensorboard
scipy = 1.3
Macbook Pro Apple M1 chip
```

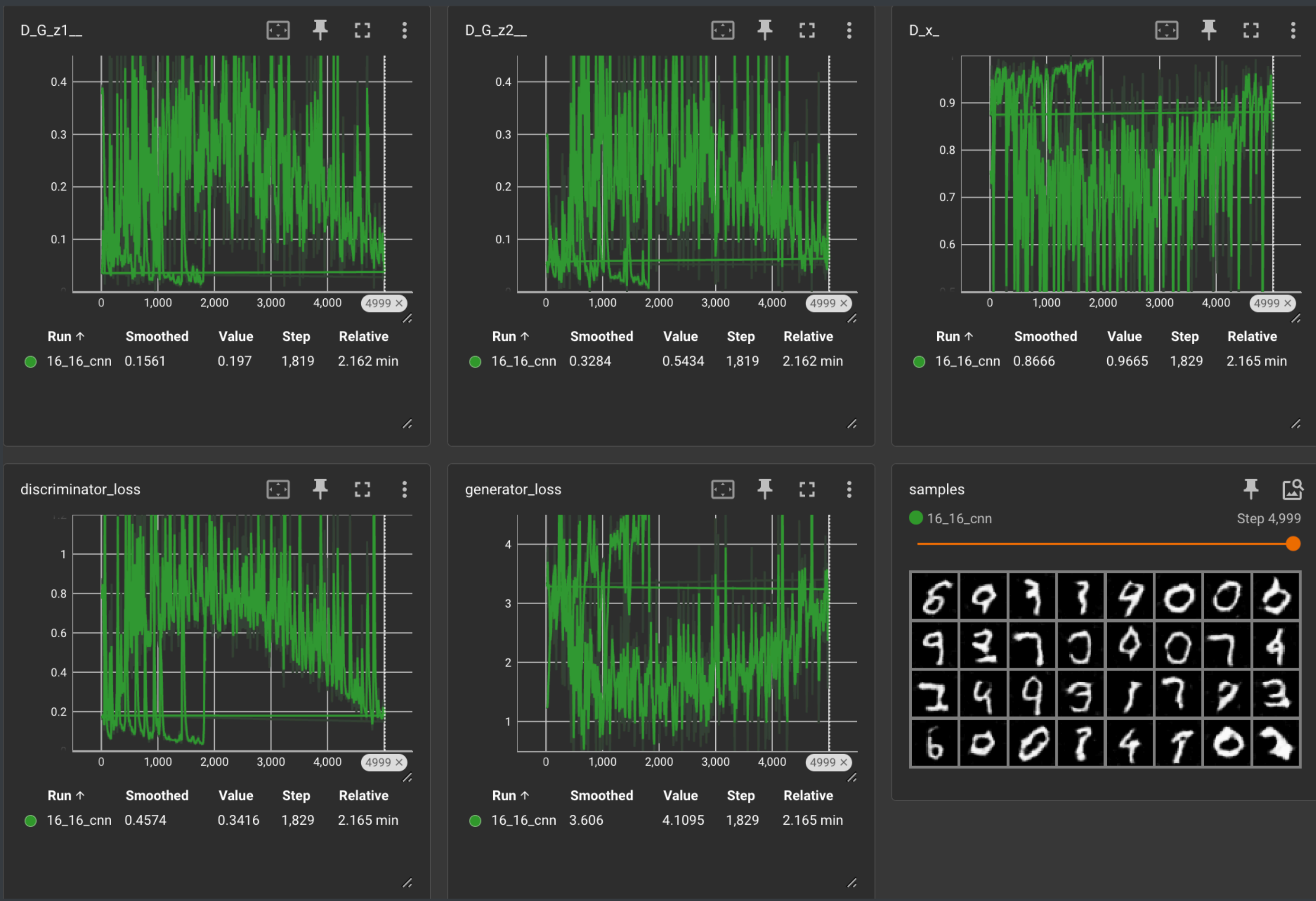
1. GAN with default parameters

basic parameters were:

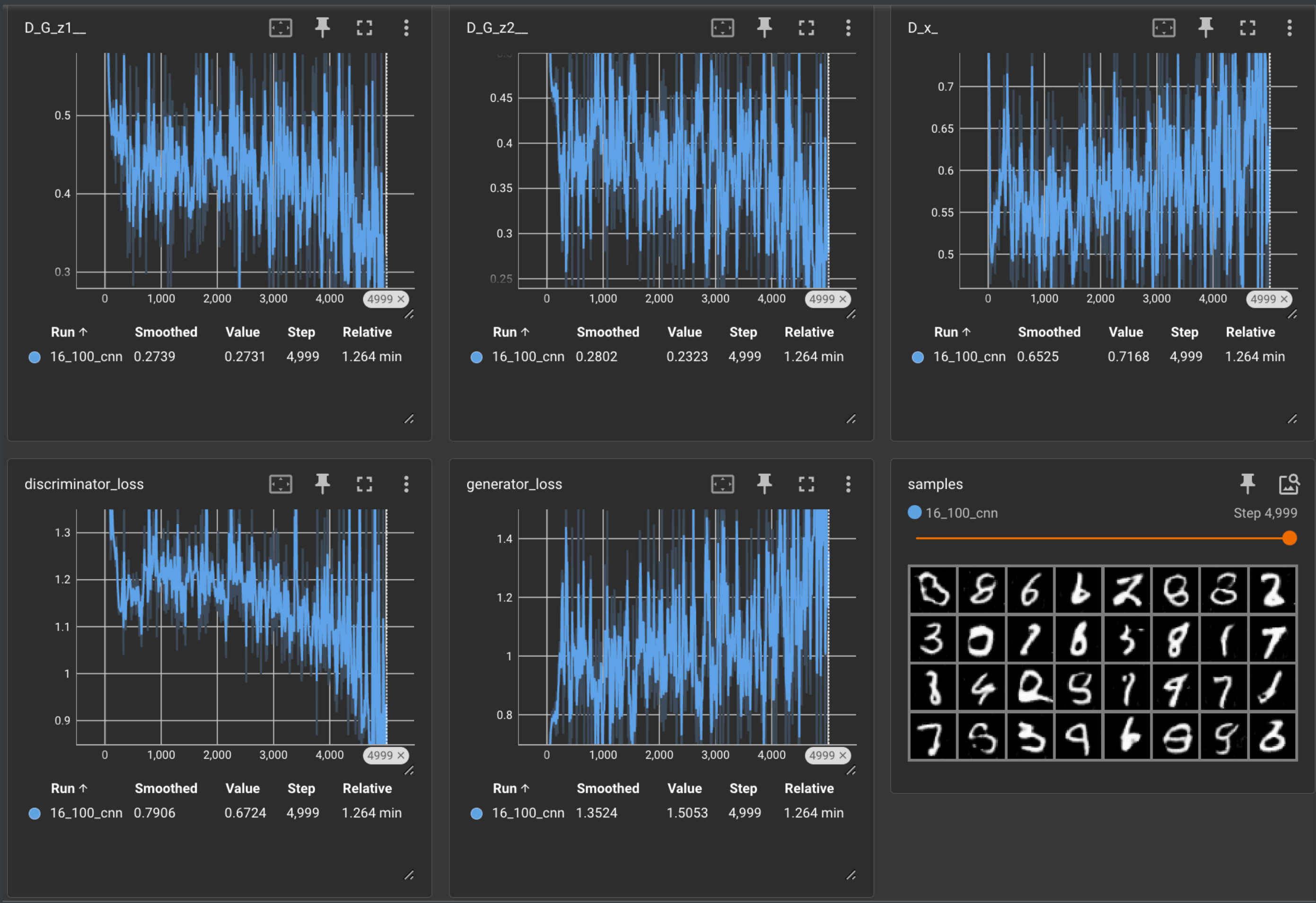
```
batch_size = 64
num_train_steps = 5000
```

based on the basic parameters, changes to `latent_dim` and `hidden_dim` were made:

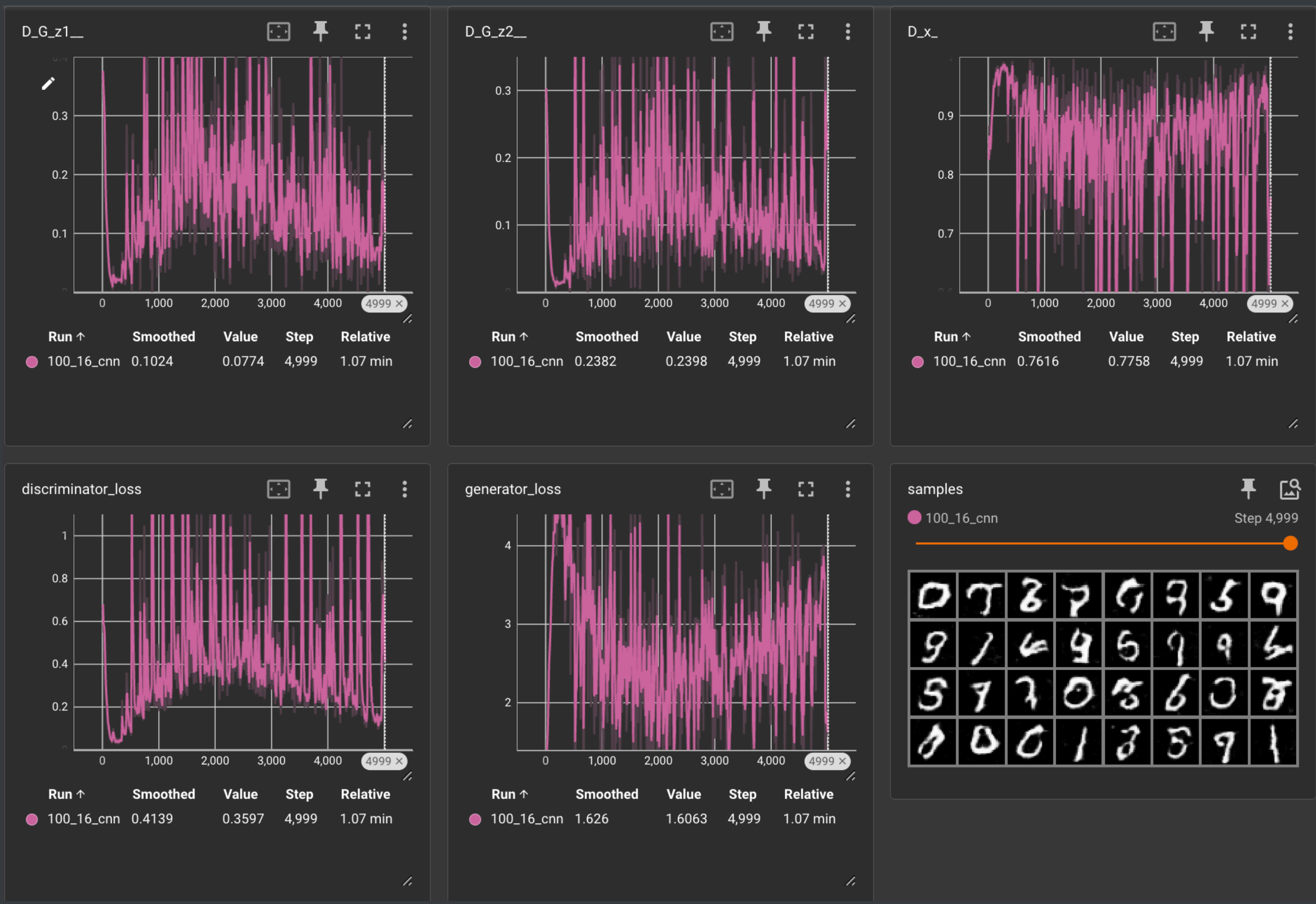
`latent_dim` = 16, `hidden_dim` = 16:



`latent_dim` = 16, `hidden_dim` = 100:



`latent_dim` = 100, `hidden_dim` = 16:



`latent_dim` = 100, `hidden_dim` = 100:

2. FID Score

based on the four variations of `latent_dim` and `hidden_dim` above, the FID scores for each fluctuated depending on the value of the seed used, therefore the `seed` value was randomly selected between 2022 - 2025, the training was then run for each model **5** times and the best FID score taken is shown below:

```
latent_dim = 16, hidden_dim = 16       FID = 83.6592
latent_dim = 100, hidden_dim = 16      FID = 79.2028
latent_dim = 16, hidden_dim = 100      FID = 43.3676
latent_dim = 100, hidden_dim = 100     FID = 41.9363
```

3. The Impact of `latent_dim` vs `hidden_dim`

`hidden_dim`:

- **Model Complexity:**

  The hidden dimension directly influences the capacity and complexity of the neural networks in a GAN. A larger `hidden_dim` provides the model with more parameters, enabling it to learn more detailed features of the data distribution. However, an excessively large `hidden_dim` does not guarantee improved performance; it could lead to overfitting and make the training process more challenging.

- **Balance in Training:**

  With a `latent_dim` of 100, increasing `hidden_dim` from 16 to 100 significantly enhanced the GAN's ability to model complex data distributions ($79.2028 \to 41.9363$), as evidenced by the improvement in FID scores. But the balance between the generator and discriminator is also a factor that should be taken into account, as an imbalance can lead to the overpowering of one network over the other, potentially causing the training to diverge. Therefore Increasing `hidden_dim` can improve the generator's performance up to a point but must be handled carefully to maintain a competitive training environment.

`latent_dim`:

- **Variability and Quality:**

  The FID scores indicate that a larger `latent_dim` can lead to improved generative quality, when holding the value of `hidden_dim` constant. This is because a larger `latent space` provides a better source of variability for the generator to draw upon.

  When `hidden_dim` was 16, increasing `latent_dim` from 16 to 100 resulted in a better FID score, demonstrating that `latent_dim` contributes to the diversity of the generated images. However, the improvement was more pronounced when `hidden_dim` was also increased to 100, highlighting the importance of having both sufficient latent space and model capacity.

- **Quality of Generation:**

  A higher `latent_dim` generally leads to a broader range of features that the generator can potentially learn, resulting in more diverse and high-quality image generation. However, the most substantial improvement in FID scores was observed when both `latent_dim` and `hidden_dim` were increased, indicating that these parameters work synergistically. A GAN with `latent_dim` and `hidden_dim` both set to 100 achieved the lowest FID score, suggesting that increasing both dimensions can lead to a better representation of complex data and, consequently, higher-quality image generation.

Both `latent_dim` and `hidden_dim` are both important to the GAN's learning capability and performance. While `hidden_dim` strongly affects the model's capacity to learn and generate detailed features, `latent_dim` provides the necessary variability in the input space. The FID scores reflect that a balanced increase in both dimensions tends to yield the best results, enhancing the GAN's ability to produce diverse and realistic images.

4. Nash Equilibrium

The curves for our model **does not** seem to be converging towards the Nash Equilibrium.

This is mainly because:

- For the Generator and Discriminator, the optimal solution of the loss function does not occur at the same point. They do not reach a Nash equilibrium (where `latent_dim = 100`) at the same time; instead, the Discriminator reaches the optimal solution of the loss function first, having effectively learned to distinguish between real and generated images. Afterward, the Generator gradually improves its generation strategy, leading to a continuous cycle of competition.
- In the early stages of training, the Discriminator learns quickly and outperforms the Generator, which motivates the Generator to improve. As the training progresses, the Generator starts to catch up, creating more realistic images, and the Discriminator's performance begins to plateau, leading to the Generator eventually closing the gap.

5. Interpolation

To apply linear interpolation in the `latent space` , args `--do_interpolated` was added and used and produced the images below:

We can see from the interpolated images above that GAN's Generator can effectively navigate the `latent space` to create intermediate images that gradually transition from one set of features to another. The quality of these images are mainly assessed by their smooth transformation, demonstrating the Generator's ability to interpolate without abrupt changes or loss of realism.

Analyzing a sequence of interpolated images reveals the model's proficiency in producing varied and realistic results without succumbing to mode collapse. This evaluation of interpolated images is crucial when trying to understand the Generator's performance in generating diverse, high-quality data.