

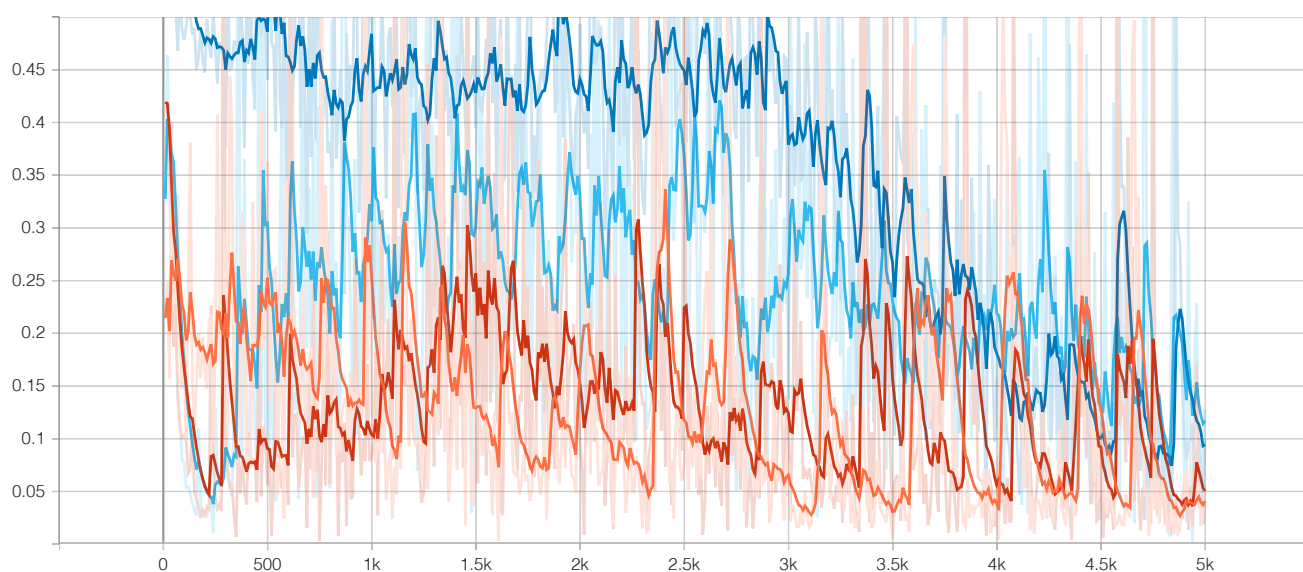
ANN HW4 Image Generation with GAN

1. 四组实验中, `batch_size=64` , `num_training_steps=5000` .

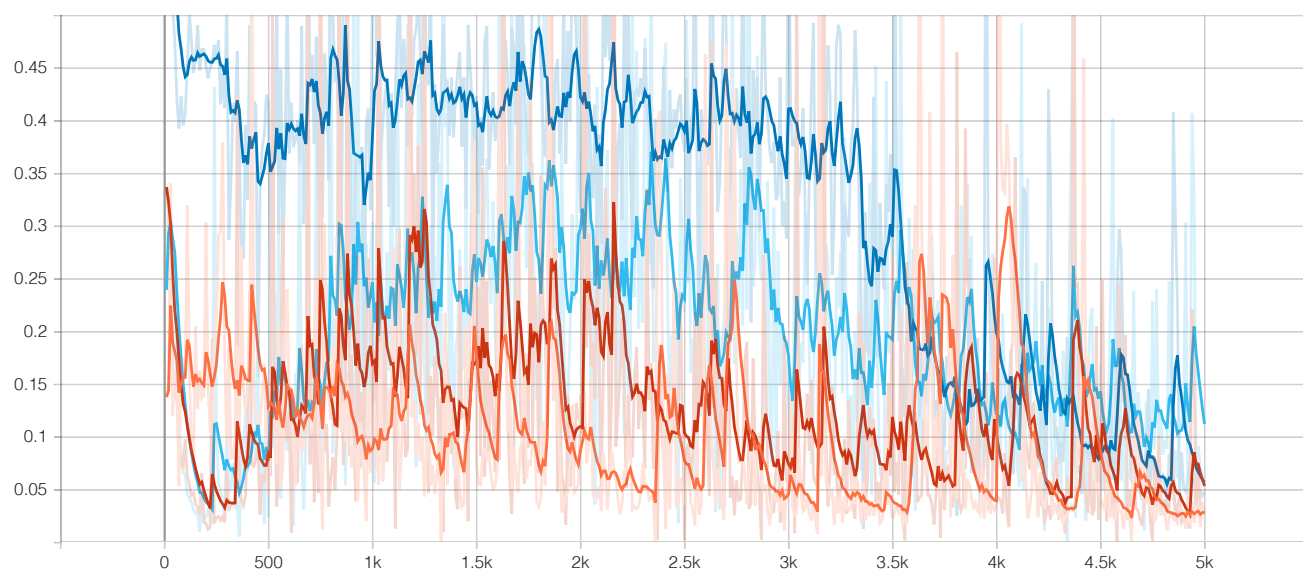
`latent_dim` , `hidden_dim` 的设置以及与相应曲线的对应关系如图所示。

- latent_dim-100_hidden_dim-100
- latent_dim-16_hidden_dim-100
- latent_dim-100_hidden_dim-16
- latent_dim-16_hidden_dim-16

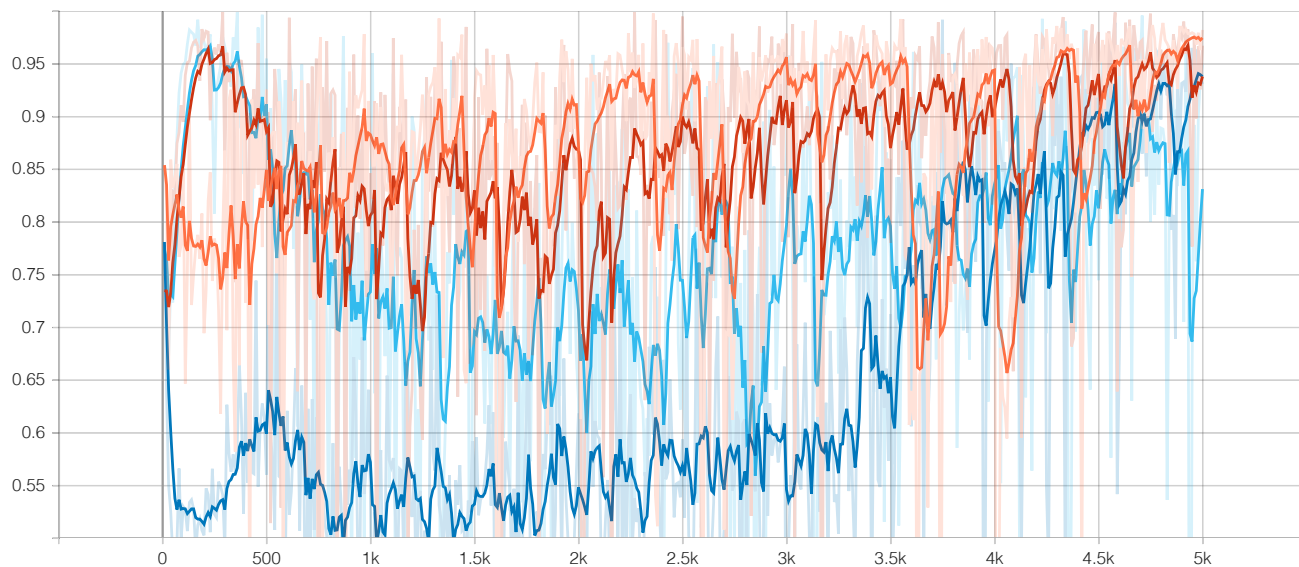
D_G_z1



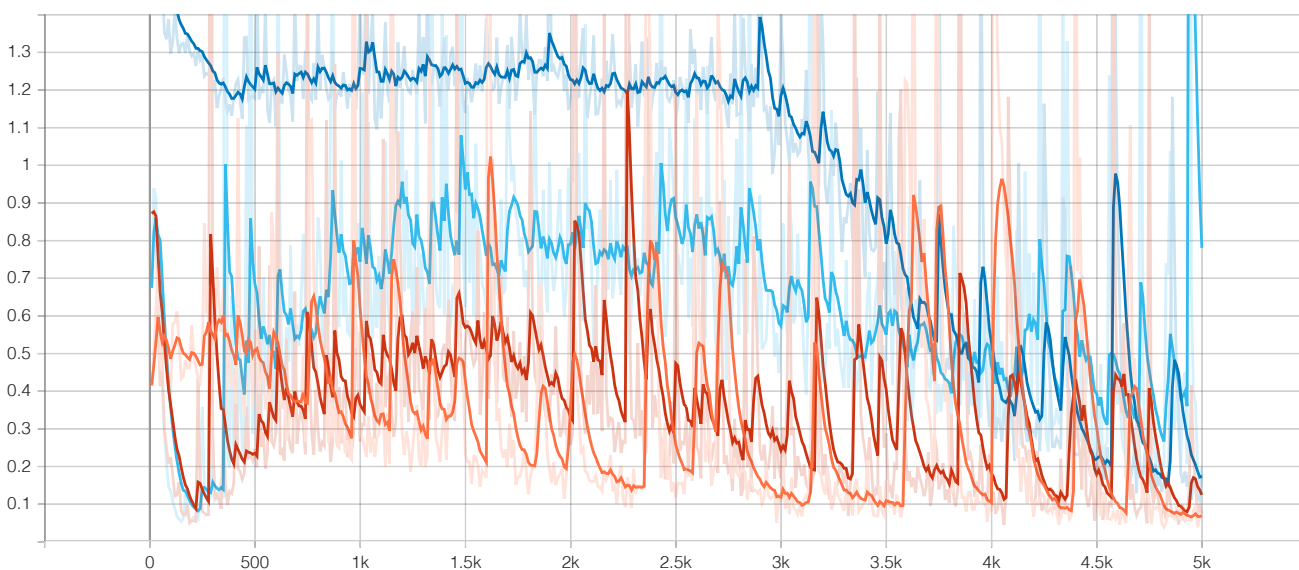
D_G_z2



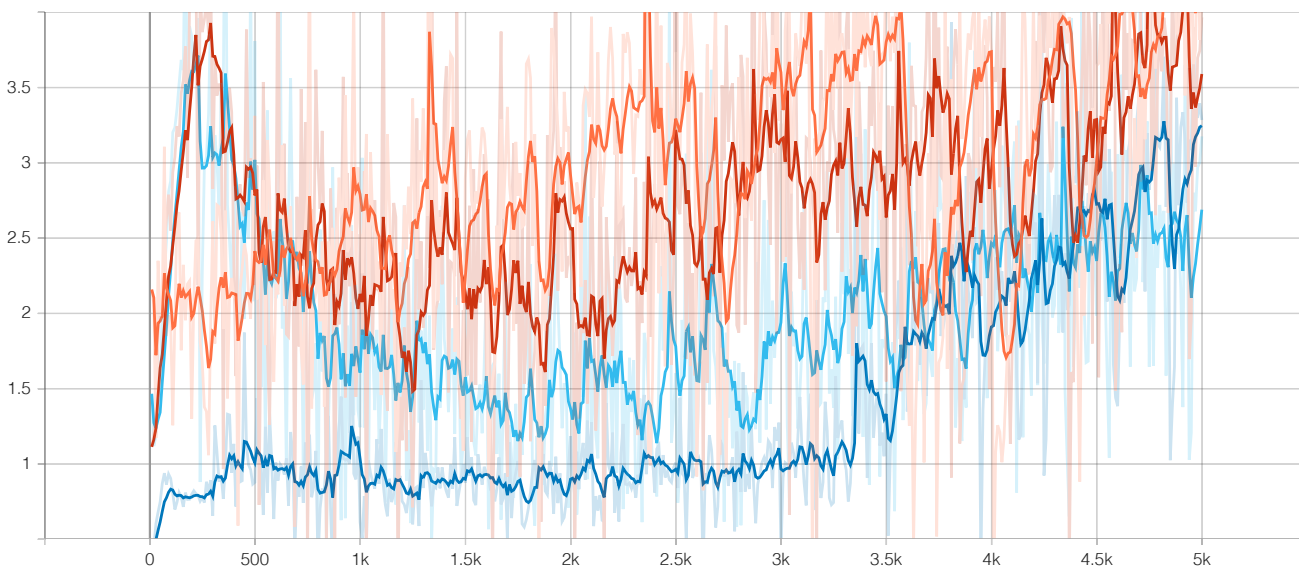
D_x



Discriminator_loss



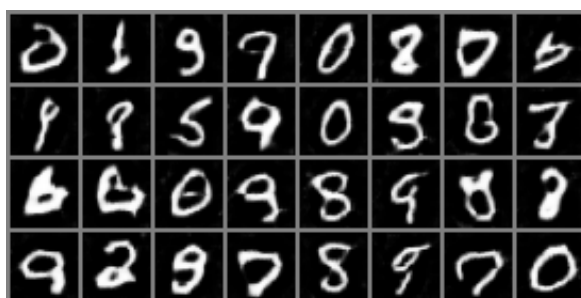
Generator_loss



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

(Latent_dim, hidden_dim)	(100,100)	(16,100)	(100,16)	(16,16)
FID Score	45.34	76.25	80.89	87.1

Best model: (Latent_dim, hidden_dim)=(100, 100).

- 由上述四组实验可以看到，无论是提高latent_dim，还是提高hidden_dim,都能够降低FID Score。个人认为，原因是latent_dim和hidden_dim都与生成图片过程中的通道数相关，通道数越多，模型更有可能学习到 real images的分布（至少不会劣于通道数较少的情况，因为可以让新增的通道不起作用，始终为0）。
- 4个GAN均未收敛到纳什均衡：Discriminator “强于” Generator。

- 当训练进行到接近5000steps时， D_x 接近1， D_{G_z1} & D_{G_z2} 接近0，Discriminator能以很高的准

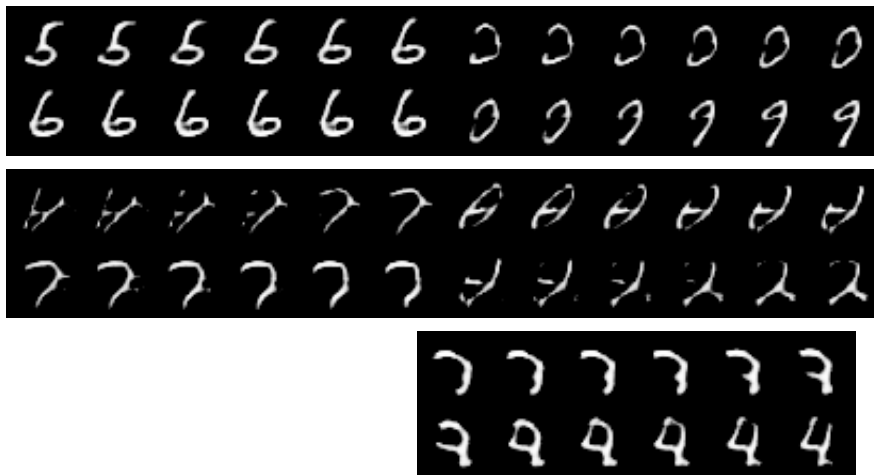
准确率分辨real/fake images；若达到均衡，`D_x`, `D_G_z1`, `D_G_z2` 应当均接近0.5；

2. `Discriminator_loss` 较初始值有所降低，而 `Generator_loss` 较初始值有所升高，也可以看出二者没有达到均衡，Generator生成的图片不能欺骗Discriminator。

5. 对latent representation做插值法

使用命令 `python main.py --interpolation` 即可得到结果。

1. 实验结果



2. 实验分析

每张图像中，左上角（第1张）图像由 z_1 生成，右下角（第12张）图像由 z_2 生成；中间的10张图像由插值法 $z_i = z_1 + \frac{i}{11}(z_2 - z_1)$ 生成。可以看到，插值法得到的latent representation生成的图像介于上下界生成的图像之间，且越靠近上（下）界的interpolation latent representation，生成的图像越接近上（下）界生成的图像。

6. Mode_collapse

使用命令 `python main.py --mode_collapse` 即可得到结果。

1. 实验结果

基于Best model生成，参数设置为

`batch_size=64`, `num_training_steps=5000`, `latent_dim=hidden_dim=100`。



	0	1	2	3	4	5	6	7	8	9
频数	11	4	0	3	0	1	9	12	7	3

MNIST数据集的分布比较均匀，训练集中每个数字出现的频数都在6000上下；而生成结果中0、6、7、8的频率明显高于其他数字，存在mode_collapse。

7. GAN with an MLP-based generator and discriminator

使用命令 `python main.py --do_train --use_mlp --latent_dim 100` 即可得到结果。

1. 实验结果



FID Score: 140.168

2. 结果分析

相比于CNN-based GAN，MLP-based GAN生成的结果更加模糊，边界不够清晰。个人认为原因在于CNN-based GAN的卷积核可以捕捉到边界信息，而MLP-based GAN接触到的是被展开成一维向量的图像，丢失了一部分边界信息，因此生成的结果不具有清晰的边界。