Improving GAN on MNIST

Team: mayuna

Marc KASPAR
Nan AN
Yuyan ZHAO

Content

f-GAN

f-GAN: a generalized approach to extending GANs to any f-divergence.

f-divergence:

$$D_f(\mathcal{P} \parallel \mathcal{Q}) = \int_x q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

where f is a convex function

Training Objective Function:

(GAN) $$F(\theta, \omega) = \mathbb{E}_{x \sim P}\left[\log D_\omega(x)\right] + \mathbb{E}_{x \sim Q_\theta}\left[\log(1 - D_\omega(x))\right]$$

(f-GAN) $$F(\theta, \omega) = \mathbb{E}_{x \sim P}\left[g_f(V_\omega(x))\right] + \mathbb{E}_{x \sim Q_\theta}\left[-f^*(g_f(V_\omega(x)))\right]$$

f-GAN

From Vanilla GAN to f-GAN:

- The final layer activation function of the discriminator: sigmoid(v) to g_f
- Loss function: f∗(t) for computing D_fake_loss, G_loss
- Accuracy metrics: f’(1) for defining the decision boundary in the

Name	**Output activation** g_f	**Conjugate** $f^*(t)$	**Threshold** $f'(1)$
Kullback-Leibler (KL)	v	$\exp(t-1)$	1
Reverse KL	$-\exp(-v)$	$-1-\log(-t)$	-1
Jensen-Shannon	$\log(2)-\log(1+\exp(-v))$	$-\log(2-\exp(t))$	0

Table 1: Activation functions, Conjugates, Thresholds for various f-divergences.

f-GAN

- Performance can vary significantly depending on the divergence used. Best performance FID 45.53 with JS divergence. Convergence difficulty with KL divergence

- Require gradient clipping or learning rate adjustments. Adding complexity to the training process.

- Find out the most effective divergence for application is time-consuming and resource-intensive.

Wasserstein GAN

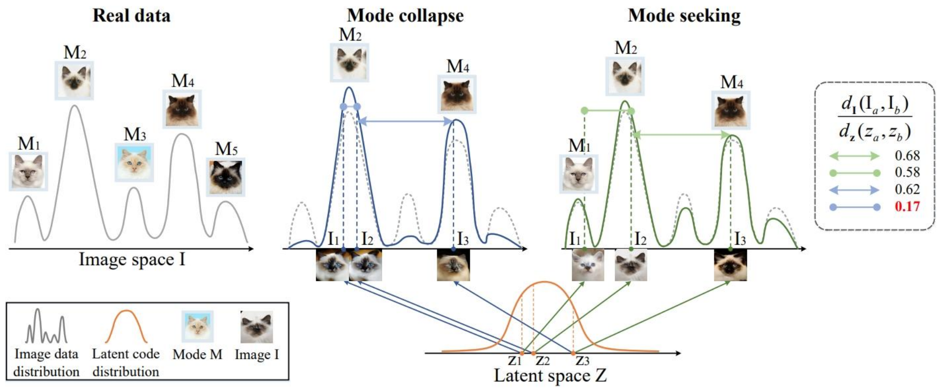
Uses Wasserstein distance:

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} E_{(x,y)\sim\gamma}[||x - y||]$$

FID: 87.54

Mode collapse

Illustration of motivation



In mode collapse data distribution

- When distance between two latent vectors Z1 and Z2 is decreasing, the mapped images distance I1 and I2 will become shorter in a **disproportionate** rate

- This lead the generator produce nearly identical images, even when the inputs are different.

Mode-Seeking GAN

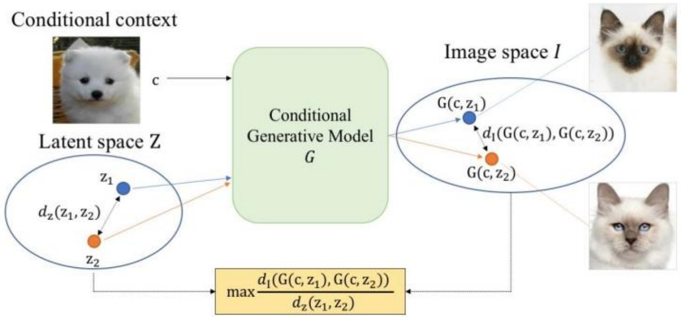
Loss Function

Regularization Term:

$$L_{ms} = \max_G \left(\frac{d_I(G(c, z_1), G(c, z_2))}{d_Z(z_1, z_2)} \right)$$

Loss Fuction:

$$L_{new} = L_{ori} + \lambda_{ms} L_{ms}$$



- $d*(\cdot)$: Distance metric.
- $G(c, z1)$: Image generated from z 1 by G.
- λ: Weight for mode-seeking regularization

- encourages the generator to explore the image space

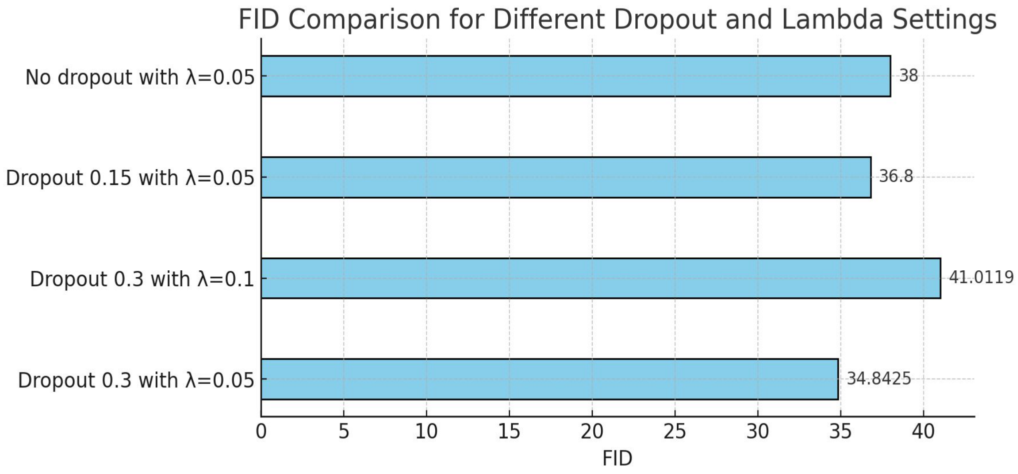
Hyperparameter Selection

Dropout and λ

Dropout in the discriminator: 0.3

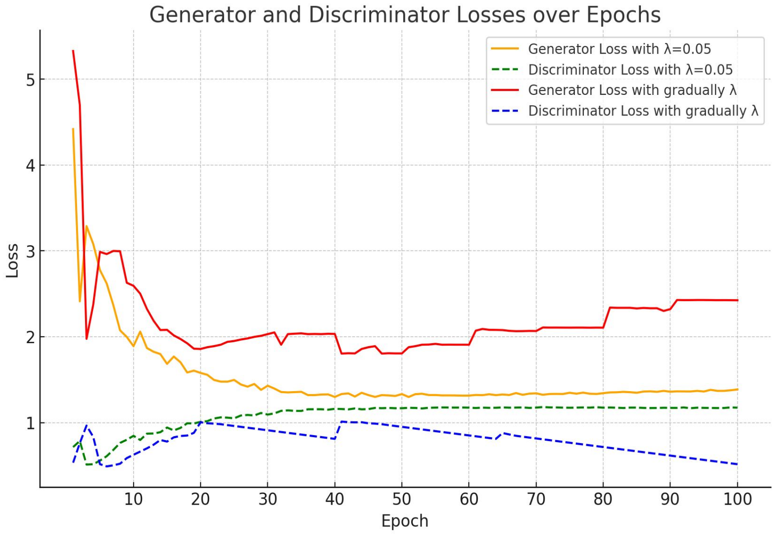
Mode-Seeking Loss Parameter λ: 0.05

FID: 34.8425



Hyperparameter Selection

Modify λ Gradually to get better precision and recall



Initially set Lambda to 0.05 and increased it by 0.01 every 10 epochs.

- FID: 39.06
- Weakness of the generator
- Discriminator is strong compare to generator

Optimize Generate Code

Use Rejection sampling and latent space Interpolation

Rejection Sampling

selecting only image that are above a confidence threshold (0.6).

Latent Space Interpolation

Introduces smooth transitions between generated samples by interpolating between two latent vectors.

Final result: FID 27.72, Precision 0.49, Recall 0.16

Thank you!

Marc KASPAR
Nan AN
Yuyan ZHAO