Introduction to Computer Vision

GENERATIVE ADVERSARIAL NETs

2014, Ian J.Good et al.

*Presented by **Nguyen Quang Huy***

Faculty of Information Technology, Ton Duc Thang University

Email: 523h0140@student.tdtu.edu.vn

QR Code

Presentation slide

GAN Implementation

Github Resources

Outline

1. Introduction & General Idea
2. Architecture
3. Challenges & improvements
4. Applications
5. Demo / Visualization

1. Introduction & General Idea

Ian Goodfellow (1987) source: Wikipedia

- B.S & M.S at Stanford
- Ph.D at Montreal
- Google Brain
- OpenAI
- Google Research
- Apple
- Google Deepmind

1. Introduction & General Idea

Stage 1 Feature learning

- Image processing
- SVM / KNN, CNN

Stage 2 Detection & Segmentation

- Object detection: 2 - Stage Detector (R-CNN), 1 - Stage Detector (YOLO)
- Image segmentation: Semantic, Instance Segmentation

Stage 3 Tracking & Attention

- Object tracking
- Object Recognition (SORT, DeepSORT)
- Attention Mechanism & VisionTransformer

1. Introduction & General Idea

Why we need Generative Models ?

- Discriminative models P(Class| )

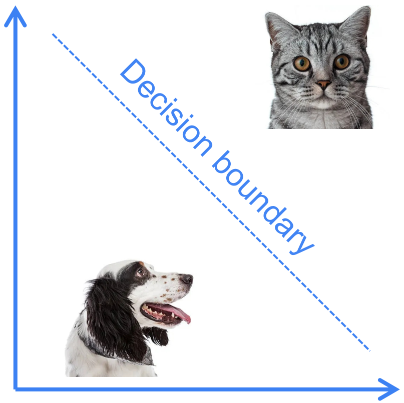
- Generative models P()

1. Introduction & General Idea

Discriminative AI vs Generative AI

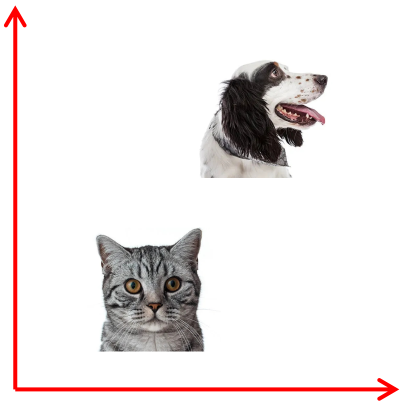
Discriminative

Classify or labeling data points as cat or dog



Generative

Produce a new data points that looks like dog or cat



1. Introduction & General Idea

Probabilistic Graphical Models	2006	Unsupervised Deep Learning Models	2014	Adversarial & Explicit Likelihood Models	2020	Diffusion Models
Gaussian Mixture		Restricted Boltzmann		Var. Autoencoders		**Diffusion models**
Hidden Markov		Deep Belief Networks		**GAN**		Transformer
Bayesian Networks		Autoencoders		PixelRNN & PixelCNN		

2. Architecture

Adversarial Learning

D/G try to maximize/mininize

value of objective function

Expected on noise distribution

$$\boxed{\min_G \max_D V(D, G)} = \boxed{\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})]} + \boxed{\mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]}$$

Expected on real dataset

$D(x) \in [0, 1]$ the probability that the D believes the sample x is real

$G(z)$ generated image from the z noise

$x \sim p_{data}(x)$ random variable x taken according to real data distribution

2. Architecture

Adversarial Learning

Discriminator (D)

$$maxV(D,G)$$

Real: $D(x) \rightarrow 1, x \sim P_{data}(x)$

$$logD(x) \rightarrow 0^-$$

Fake: $D(G(z)) \rightarrow 0, z \sim P_z(z)$

$$log(1 - D(G(z))) \rightarrow 0^-$$

Generator (G)

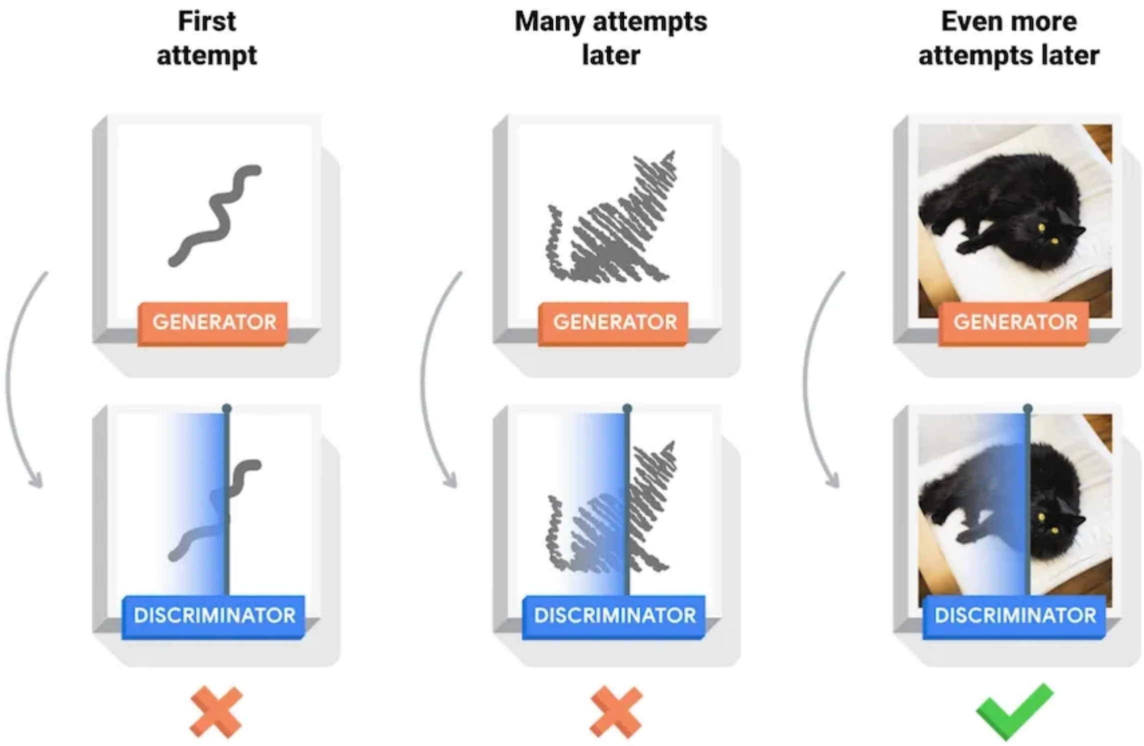
$$minV(D,G)$$

Fake: $D(G(z)) \rightarrow 1, z \sim P_z(z)$

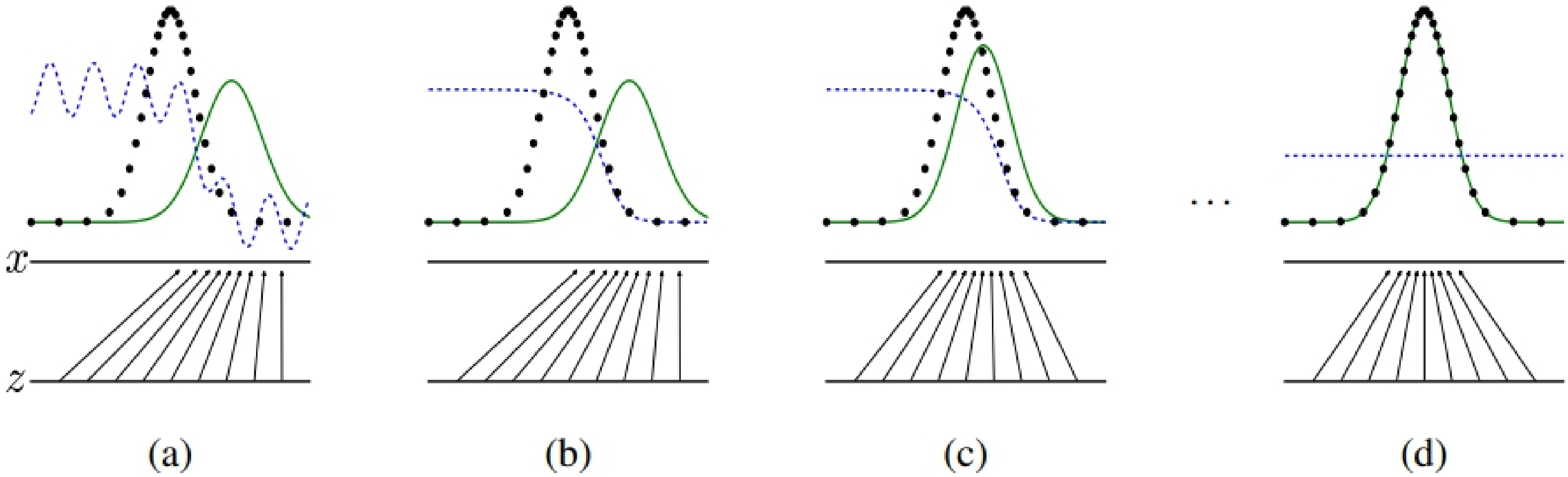
$$log(1 - D(G(z)) \rightarrow -\infty$$

2. Architecture

Adversarial Learning



2. Architecture



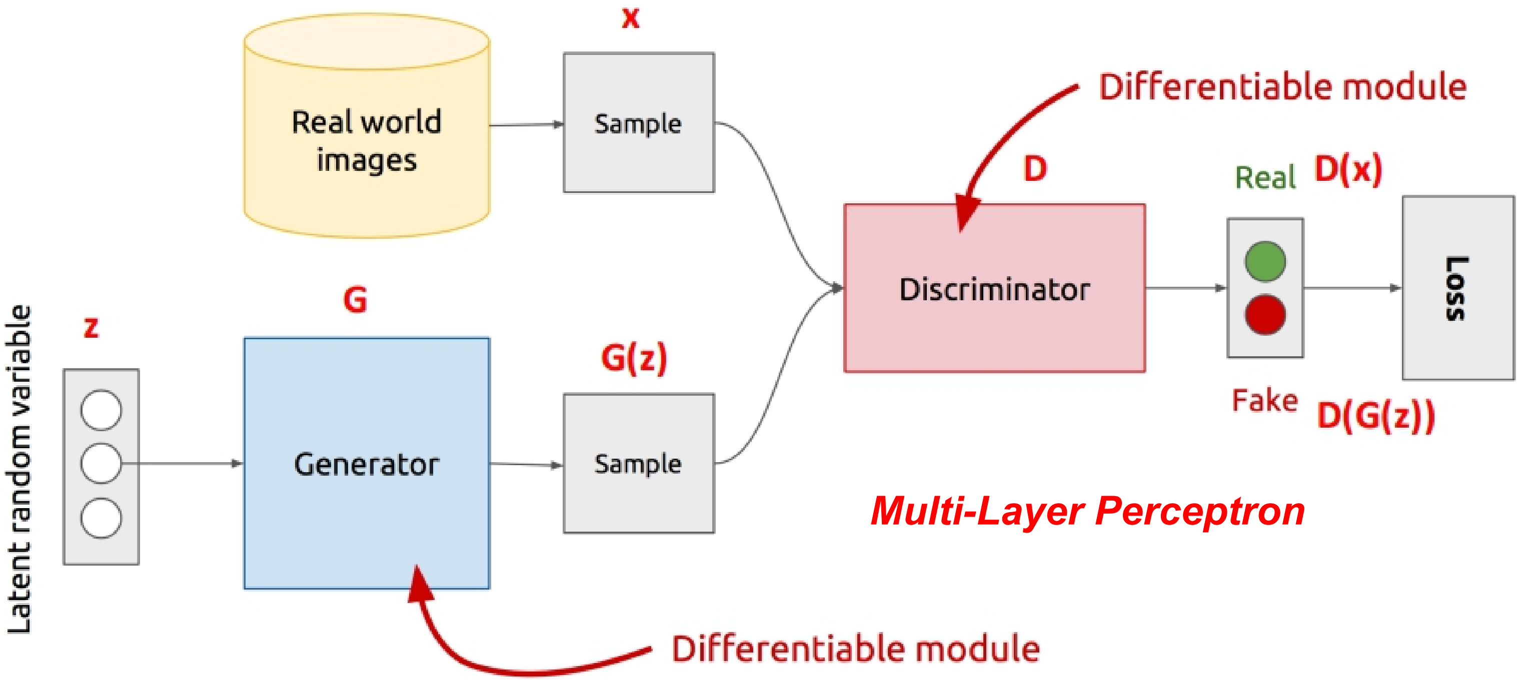
Start training | **After few steps** | **Update G** | **Nash Equilibrium**

discriminative distribution (D, blue, dashed line)

data generating distribution (black, dotted line)

generative distribution (green, solid line)

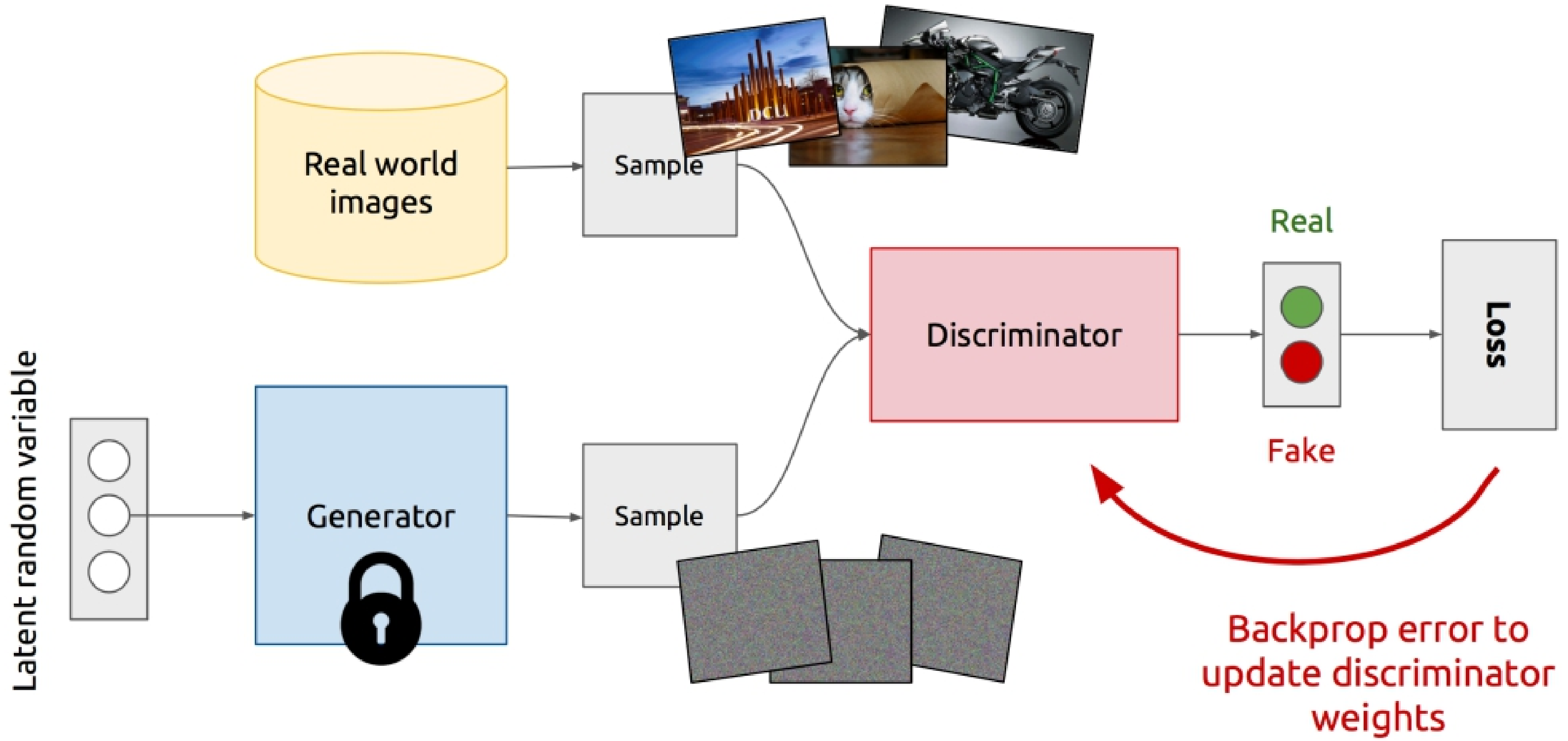
2. Architecture



Multi-Layer Perceptron

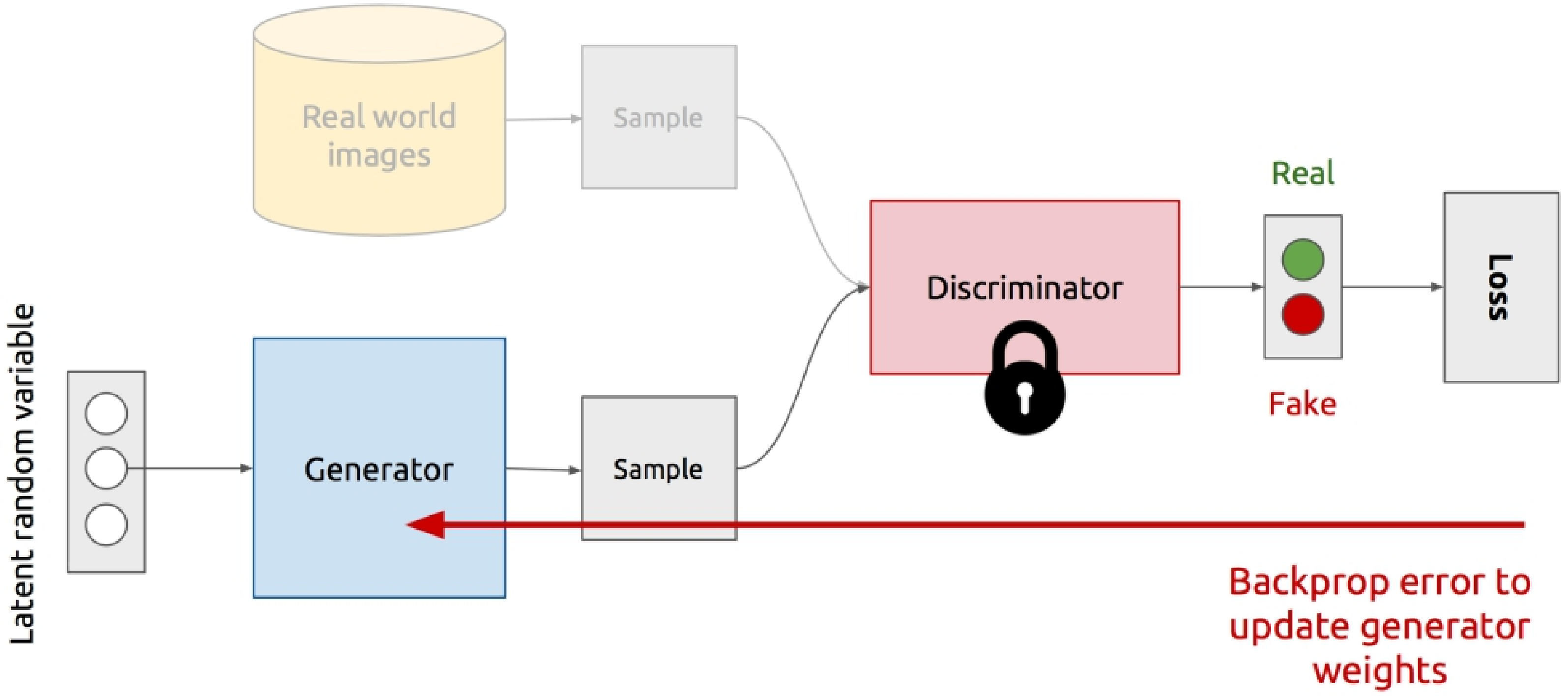
2. Architecture

Discriminator training



2. Architecture

Generator training



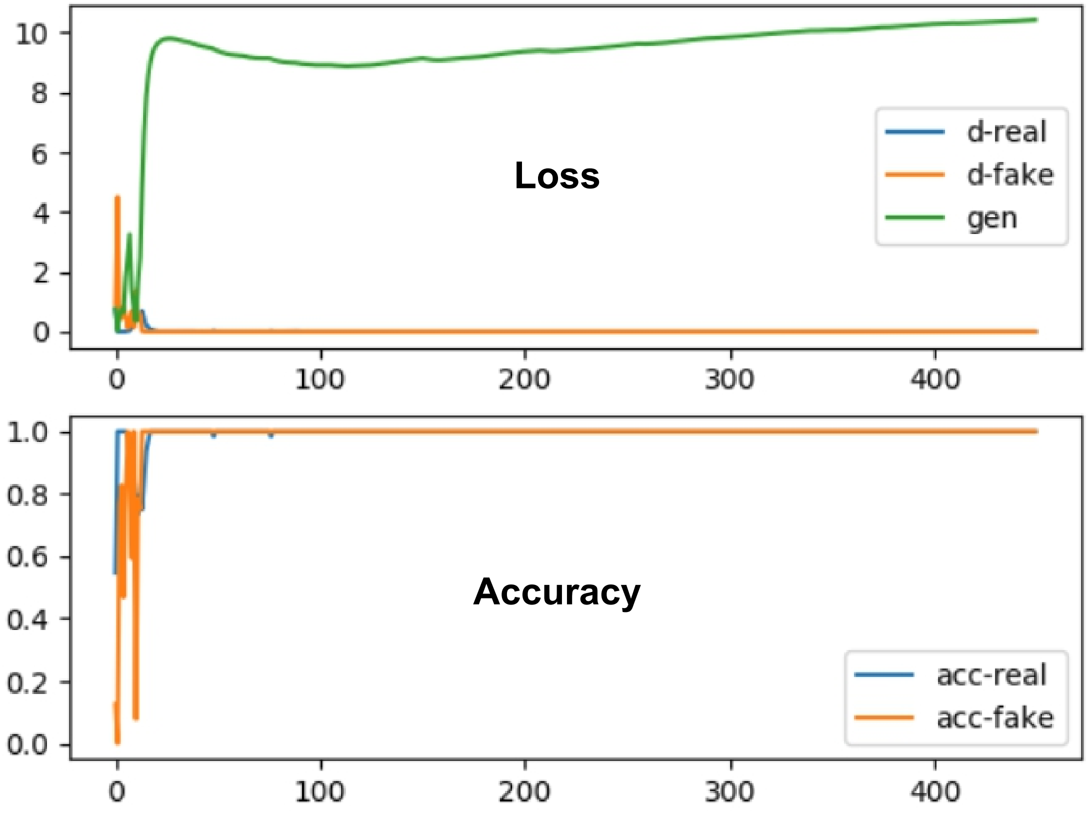
3. Challenges & Improvements

Non-covergence

In GANs architecture:

- The D tries to minimize a cross-entropy while the G tries to maximize it.
- When D confidence is high and starts to reject the samples that are produced by G leads to G's gradient vanishes.

Cause Gradient Vanishing



3. Challenges & Improvements

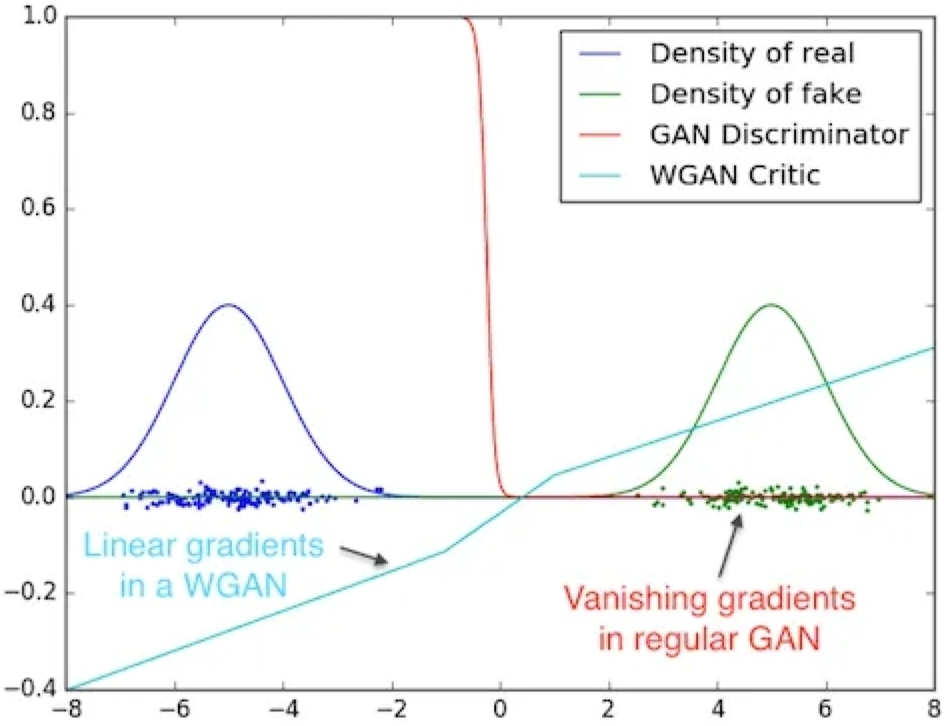
Non-covergence

3. Challenges & Improvements

Non-covergence

WGAN Improvement

- Use the Earth Mover distance.
- Provides meaningful and continuous gradients even when the real and generated distributions do not overlap.
- Removes the need for carefully balanced updates between G and D.
- **Overall**, it provides smoother convergence and a clearer measure of training progress through the critic loss.

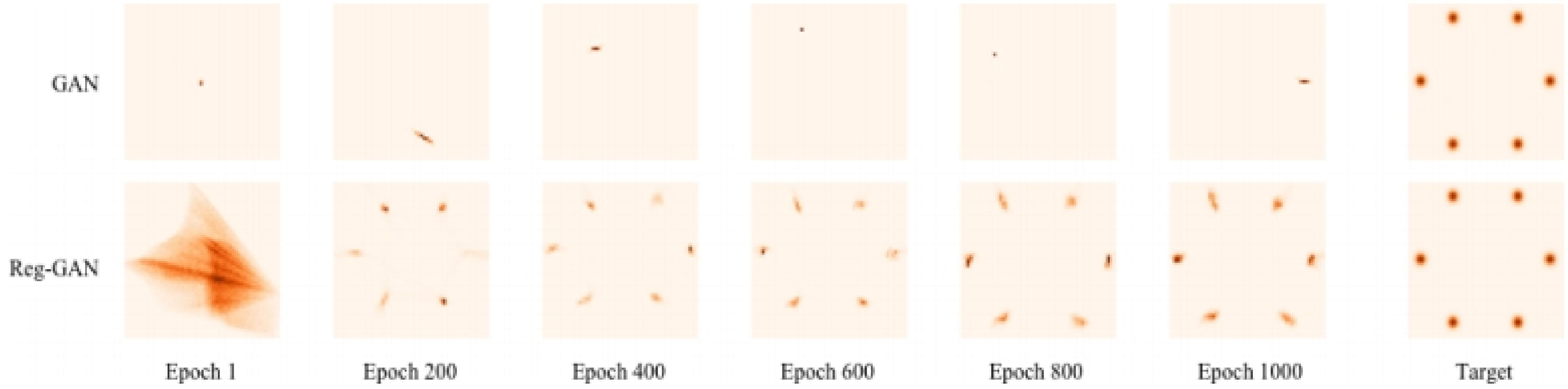


pyimagesearch.com

3. Challenges & Improvements

Mode collapse

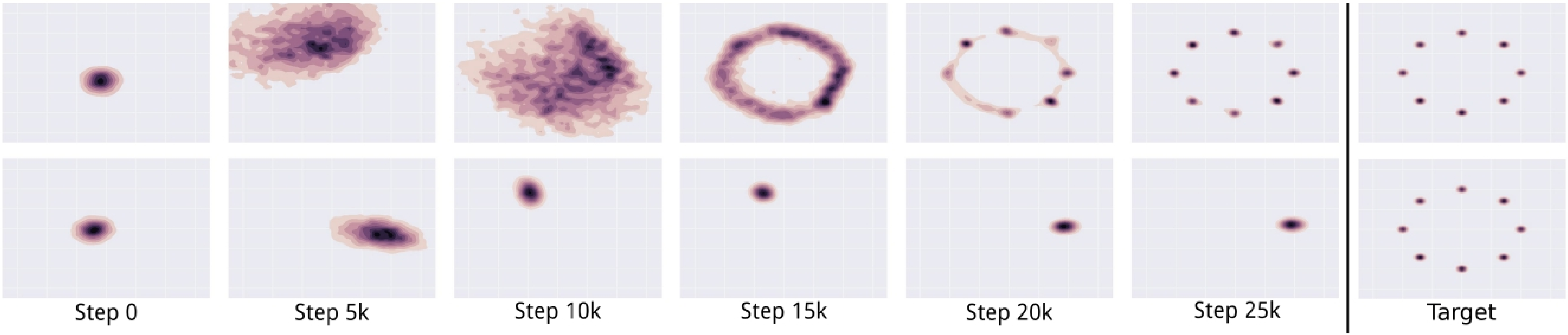
GANs can sometimes suffer from the limitation of generating samples with little representative of the population, which means that, for example, after training a GAN on the MNIST dataset, it may happen that our Generator is unable to generate digits different from digit 0.



3. Challenges & Improvements

Mode collapse Unrolled GAN

- Anticipating how the discriminator would respond to the generator's updates: before updating the generator, they "unroll" several discriminator optimization steps.
- This gives the generator a clearer, more global view of the loss landscape and discourages it from collapsing onto a single mode.



3. Challenges & Improvements

Evaluation Difficulty

- There is no ground truth reference for generated samples: No MSE, RMSE, F1, ...
- Metrics must assess both image quality and diversity, yet common losses do not directly correlate with visual realism.

3. Challenges & Improvements

Evaluation Difficulty

Improvement

- Inception Score: Pros and Cons of GAN Evaluation Measures
- **Fréchet Inception Distance:** Pros and Cons of GAN Evaluation Measures
- Kernel Maximum Mean Discrepancy: An empirical study on evaluation metrics of gans
- 1-Nearest Neighbor: An empirical study on evaluation metrics of gans
- CrossLID: Quality Evaluation of GANs Using Cross Local Intrinsic Dimensionality
- Duality Gap: A Domain Agnostic Measure for Monitoring and Evaluating GANs
- Based on 3rd Models: On the Evaluation of GANs By Discriminative Models

3. Challenges & Improvements

Fréchet Inception Distance

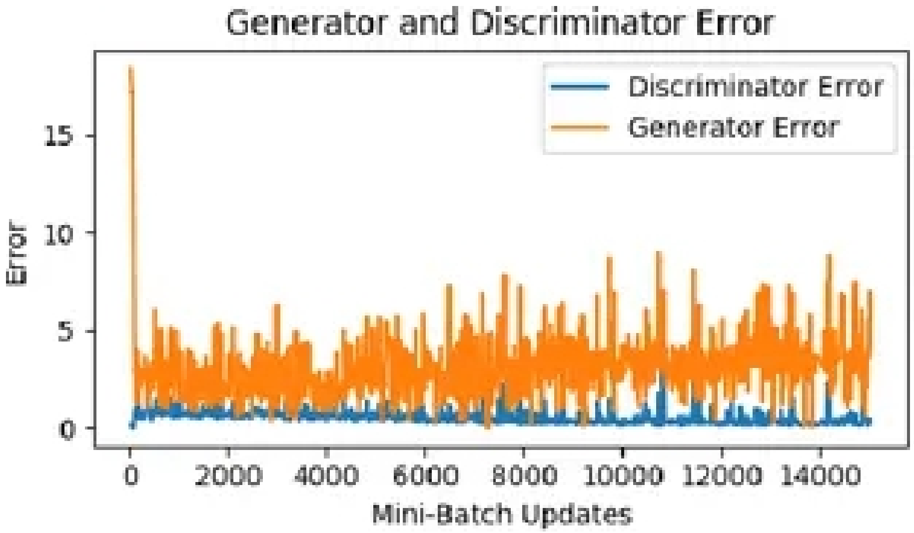
$$FID(P_r, P_g) = ||\mu_r - \mu_g||_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{1/2})$$

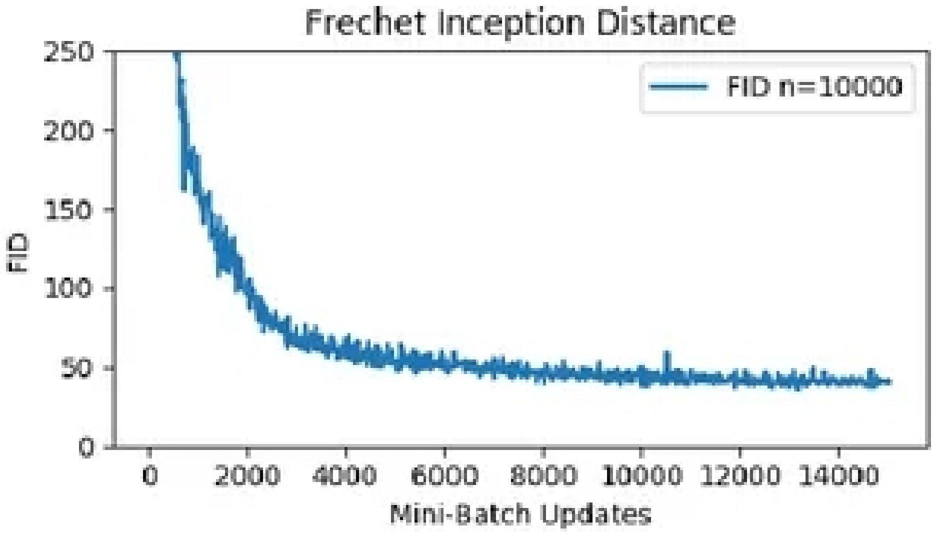
- Where $(\Sigma_r\Sigma_g)^{1/2}$ denotes the matrix square root of the product.

- Compares the distribution of real images and distribution of generated images by a pretrained network (commonly Inception-V3).
- Approximates each distribution by a multivariate Gaussian and computes the Fréchet (Wasserstein-2) distance between the two Gaussians.
- **Lower FID → generated set is closer to the real set in feature statistics (both mean and covariance).**

3. Challenges & Improvements

Fréchet Inception Distance





5. Demo / Visualization

GAN Implementation

Github Resources

References

- PyTorch implementations of GANs
- A list of all named GANs
- Vanilla GANs paper
- WGAN paper
- Fréchet Inception Distance
- A mix of GAN implementations including progressive growing
- GAN-play
- Style GAN