Generative Adversarial Networks

Challenges and Variations

Nima Rafiee
Institute of Mathematical Modeling for Biological Systems
Heinrich Hein University

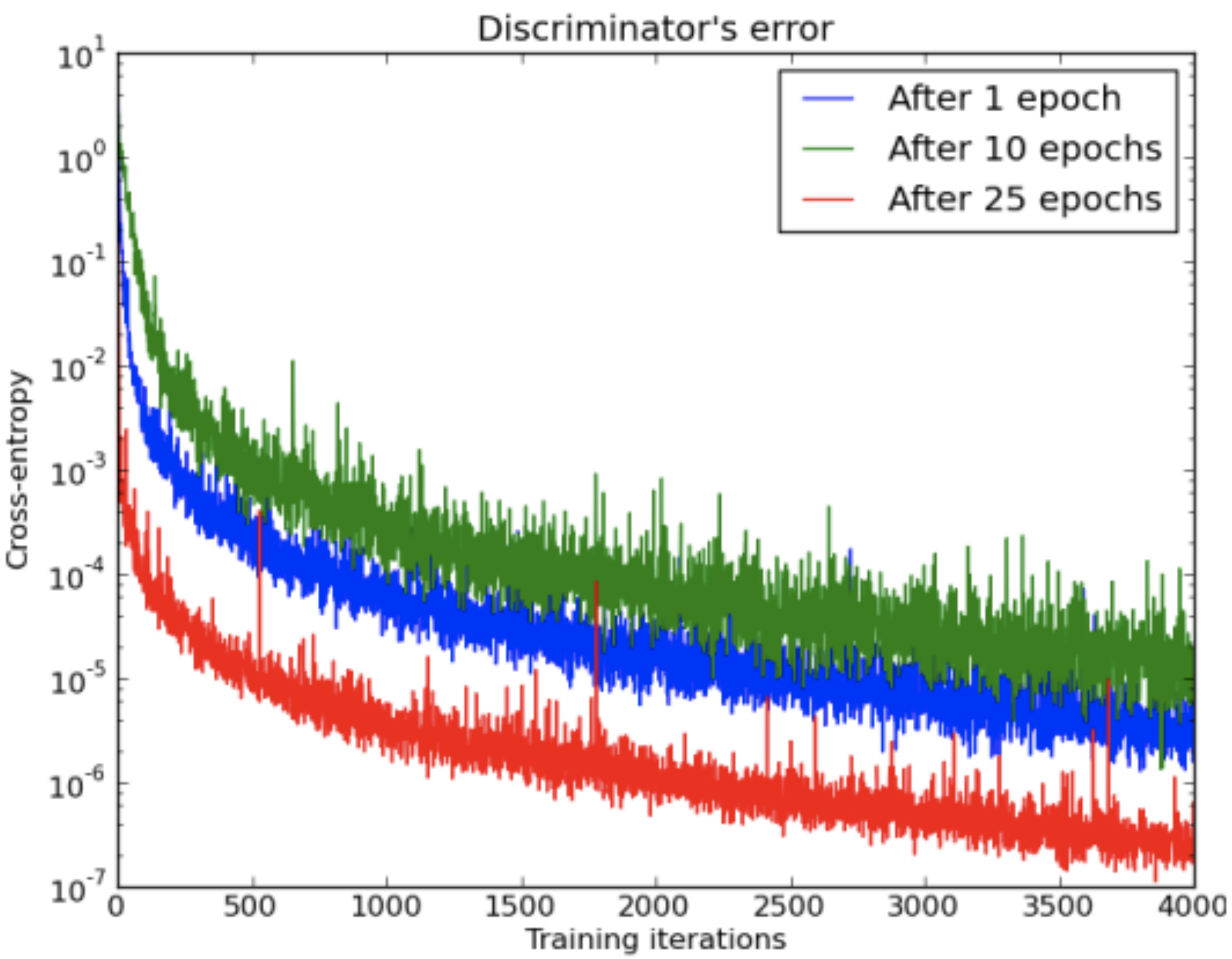
Outlines

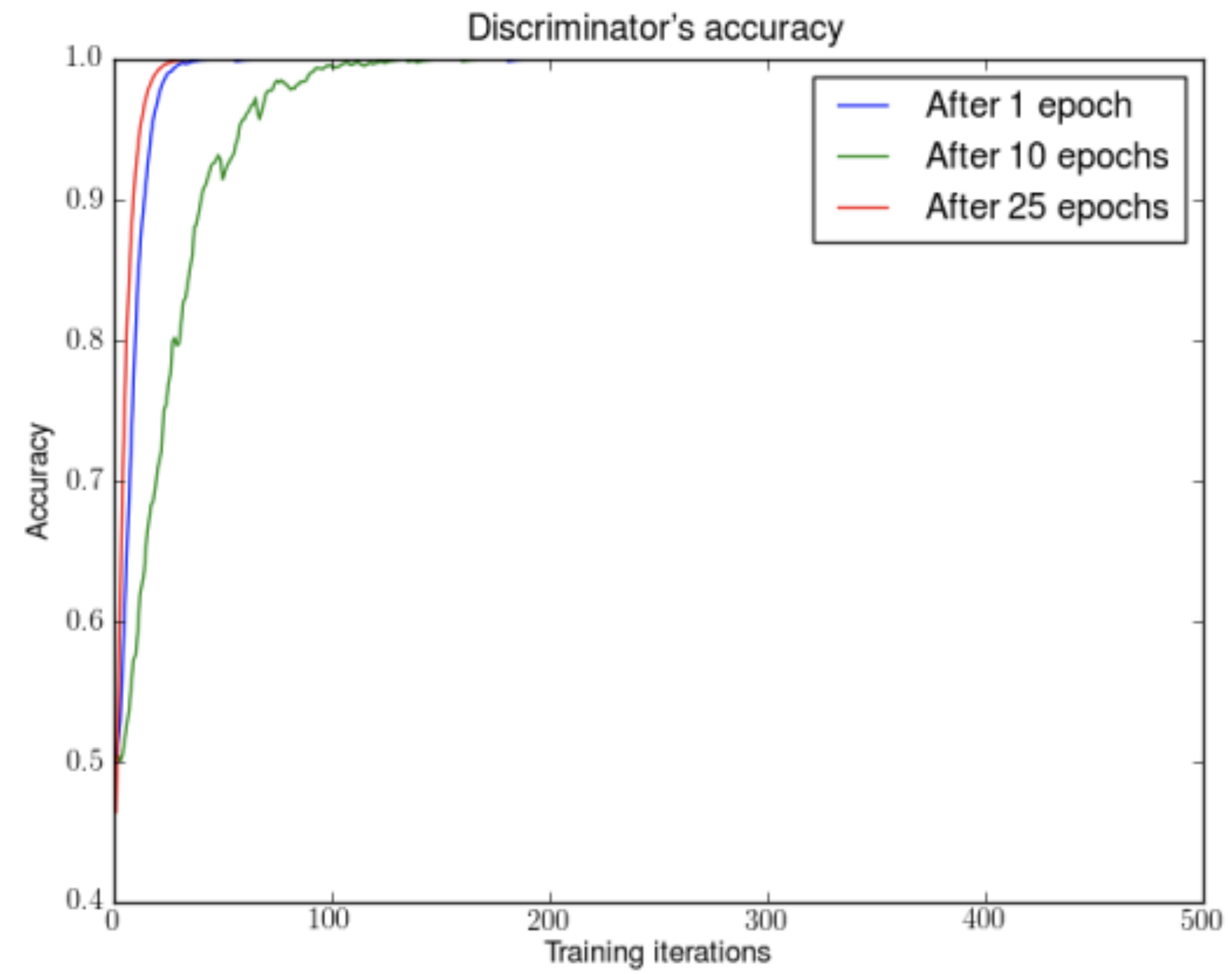
- Challenges
- Tips and Tricks for Training GAN’s
- Variations

Challenges

- Vanishing gradient
- Unstability
- Mode collapse

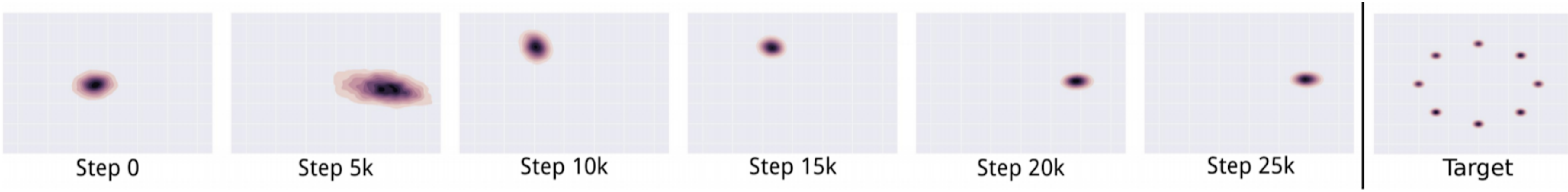
Vanishing Gradients

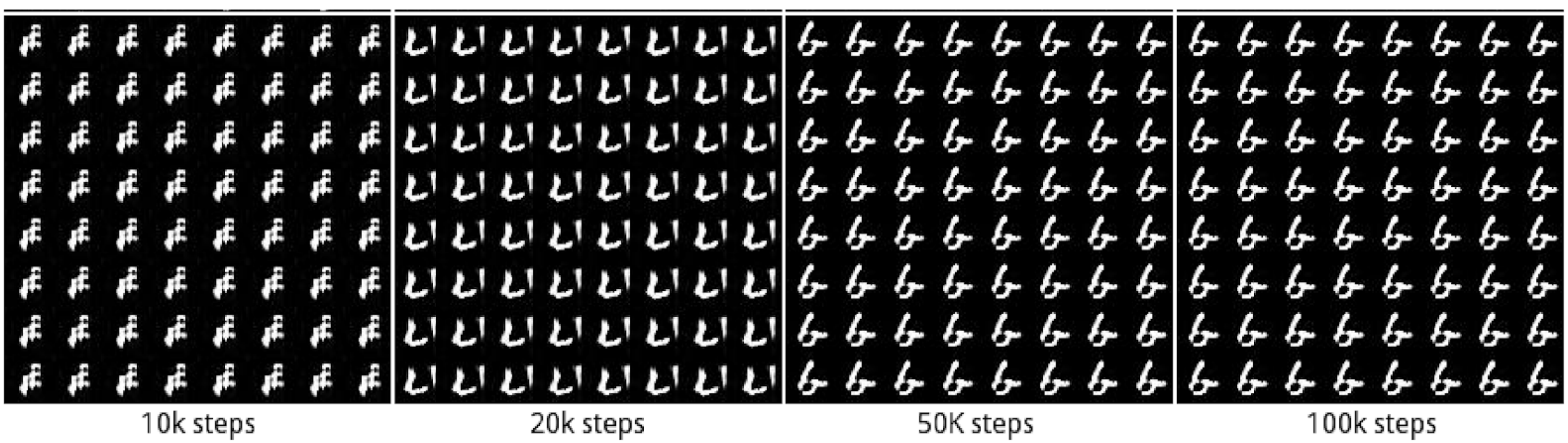




[TOWARDS PRINCIPLED METHODS FOR TRAINING GENERATIVE ADVERSARIAL NETWORKS, Arjovsky, 2017]

Mode Collapse



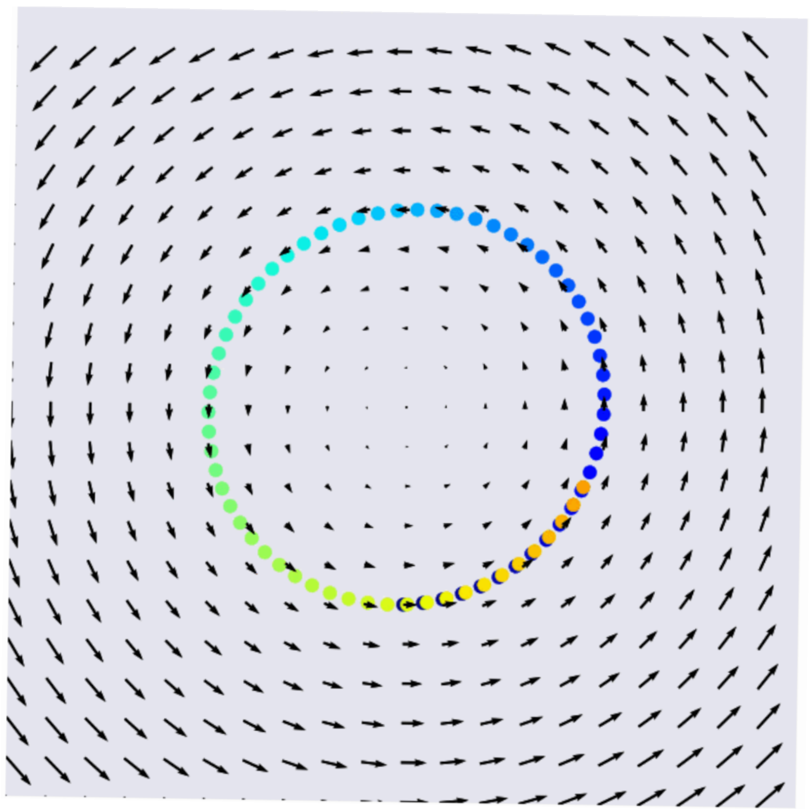


Unstability

- GAN IS a Two Player Zero-Sum Game
- Zero-sum results in a non-conservative vector field

$$D(\theta_1,\theta_2) = -G(\theta_1,\theta_2)$$

$$v(x) = \begin{pmatrix} \frac{\partial}{\partial\theta_1} D(\theta_1,\theta_2) \\ \frac{\partial}{\partial\theta_2} - D(\theta_1,\theta_2) \end{pmatrix}$$



-

Training GAN's Tips and Tricks

- Train with labels
  - Class conditional GAN's
- One-sided label smoothing
  - Decrease the confidence (not accuracy) of discriminator
  - sigmoid_cross_entropy_with_logits(D(x), 0.9) + sigmoid_cross_entropy_with_logits(D(G(z)), 0.9)
- Virtual batch normalization
  - Combinationn of batch and reference batch normalisation
- G > D or G < D or G=D ?
  - GANs work by estimating the ratio of the data density and model density
  - fine for the discriminator to overpower the generator.
  - increasing the generator size and problem of mode collapse

Training GAN's Tips and Tricks

- Minibatch features (discrimination)
  - Addresses mode collapse problem
  - Allow the discriminator to compare an example to a minibatch of generated samples and a minibatch of real samples
- Feature matching
  - Adresses non-convergence by a new objective function for generator

$$\| E_{x \sim pdata} f(x) - E_{z \sim p_z(z)} f(G(z)) \|_2^2$$

  - Train the generator to match the expected value of features on an intermediate layer of discriminator
  - Prevents generator from overfilling on current discriminator

Variations

- WGAN (Wasserstein GAN)
  - Smoothed Gradients
- SeqGAN
  - Sequential Data
- CGAN
  - Conditioning on Class Labels
- InfoGAN
- CycleGAN

Try to make things simple but not simpler

–Albert Einstein

REFERENCES

[1] Ian GoodFellow. NIPS 2016 Tutorial: Generative Adversarial Networks, 2016

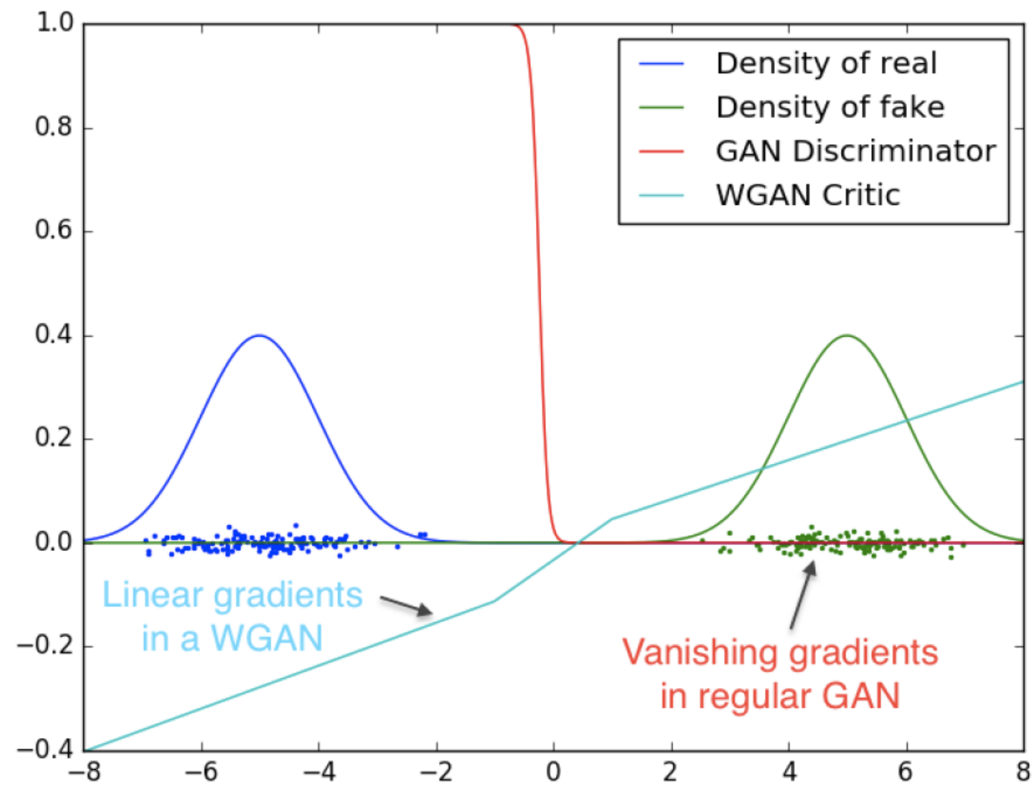
[2] David Marr, Vision. A Computational Investigation into the Human Representation and Processing of Visual Information,1982

[3] Lars Mescheder, Sebastian Nowozin. The Numerics of GANs

WGAN (Wasserstein GAN)

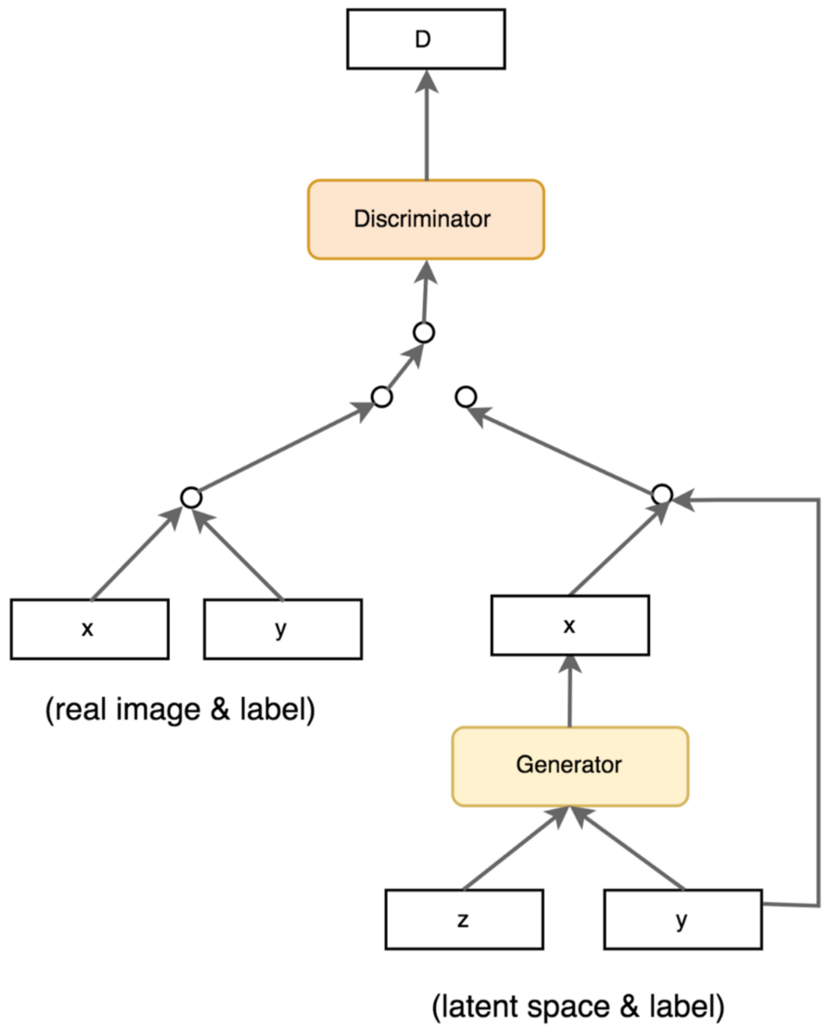
	Discriminator/Critic	Generator
GAN	$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$	$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} -\log\left(D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$
WGAN	$\nabla_w \frac{1}{m} \sum_{i=1}^{m} \left[f\left(x^{(i)}\right) - f\left(G\left(z^{(i)}\right)\right)\right]$	$\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -f\left(G\left(z^{(i)}\right)\right)$

More smooth gradient, so model always learns



CGAN

- Provides labels in both G



InfoGAN

$$\min_G \max_D V_{infoGAN}(D, G) = V_{GAN}(D, G) - \lambda I(c; G(z, c))$$

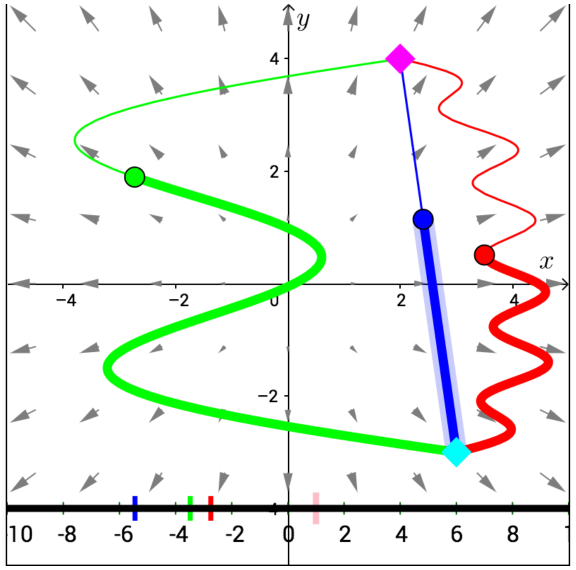
with

$$V_{GAN}(D, G) \equiv \mathbb{E}_{x \sim p_{data}} log D(x) + \mathbb{E}_{z \sim p_z(z)} log(1 - D(G(z, c)))$$

The discriminator should predict the class also

Conservative vs Non-conservative Vector Fields

- Non-conservative vector fields are independent of the path



http://mathinsight.org/

- Conservative vector fields are path dependent
  - Often used for gradient of a scalar function

GAN IS a Two Player Zero-Sum Game

- Discriminator and Generator are trying to maximise their own payoff

- Gradient ascend to find Nash equilibrium

- $D(\theta_1, \theta_2)$
  $G(\theta_1, \theta_2)$

$$x_{t+1} = x_t + hv(x)$$

$$v(x) = \begin{pmatrix} \frac{\partial}{\partial \theta_1} D(\theta_1, \theta_2) \\ \frac{\partial}{\partial \theta_2} G(\theta_1, \theta_2) \end{pmatrix}$$