Review on Generative Adversarial Networks

Presenter: Zhe Wang

https://qdata.github.io/deep2Read

Zhe Wang

201909

Content

1. Model
   - GAN, f-GAN
   - WGAN, WGAN-GP, SN-GAN
   - GANs, VAEs and GMMNs, Statistical Analysis and Information Theory
   - A unified model

2. Application and architectures
   - Generative models
   - Other applications: I to I translation, domain adaptation, adversarial samples, inverse problems

Vanilla GAN analysis[1]

Task: A image dataset, whose distribution is represented by P_r. Find a function G, s.t. $G(N(0,1)) = P_r$, we note $f(N(0,1))$ as P_g.

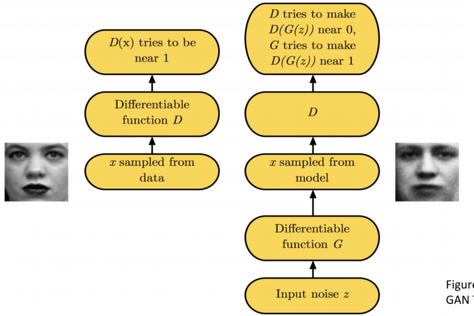


Figure from NeurIPS 2016 GAN Tutorial (Goodfellow)

Two player minimax problem (zero-sum, saddle point):

- player D distinguishes P_r from p_g,
- player G fools discriminator D.

[1] Generative Adversarial Nets

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_r}[\log D(x)] + \mathbb{E}_{x \sim P_g}[\log(1 - D(x))] \tag{1}$$

- D maximize the log-likelihood of a binary classification
- G minimize the log probability of being classified as 'fake' by D

To see clearly, fix G, find the optimal D, take the derivative over D:

$$D^* = \frac{p_r(x)}{p_r(x) + p_g(x)}, \tag{2}$$

take into loss function, we get:

$$\min_G 2JSD(P_r || P_g) - 2log2 \tag{3}$$

Minimizing the loss function is equivalent to minimize the JS divergence betweent P_r and P_g.

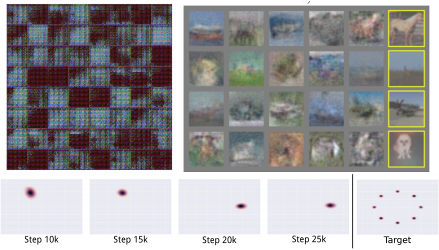
Pros and Cons

Pros:

- fast sample method (Compared with MCMC)
- no inference (Compared with graphic models)
- visually satisfaction

Cons:

- Training unstable[2]
- Mode collapse[3]
- Just do sample memorization[4]



[2] Improved techniques for training GANS

This JS divergence is a special case of f-divergence family, which is defined as:

$$D_f(P_r||P_g) = \int_x p_g(x) f(\frac{p_r(x)}{p_g(x)}) dx \tag{4}$$

where $f : \mathbb{R}^+ \rightarrow \mathbb{R}$ is convex, lower semi-continuous with $f(1) = 0$, also for the same reason $f^{**}(u) = f(u)$, and:

$$f(u) = \sup_{t \in dom_{f*}} \{tu - f^*(t)\}. \tag{5}$$

Take it into the definition, we get a lower bound for f-divergence

$$D_f(P_r||P_g) \geqslant \sup_{T \in \mathcal{T}} (\mathbb{E}_{x \sim P_r}[T(x)] - \mathbb{E}_{x \sim P_g}[f^*(T(x))] \tag{6}$$

[5] f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.

Using the variational method w.r.t $T(x)$, find the optimal value for $T(x)$:

$$T^*(x) = f'(\frac{p_r(x)}{p_g(x)}), \tag{7}$$

The lower bound get tight if $T(x) = T^*(X)$.
With this lower bound, we can do reparameterization for f divergence and get the loss function:

$$\min_{p_g}(P_r||P_g) = \min_{\theta_g} \max_{w}(\mathbb{E}_{x\sim P_r}[T_w(x)] - \mathbb{E}_{x\sim P_g}[f^*(T_w(x))]) \tag{8}$$

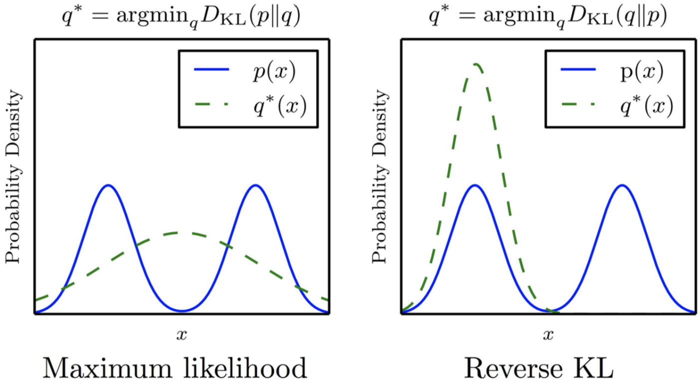
Name	$D_f(P\|Q)$	Generator $f(u)$	$T^*(x)$
Kullback-Leibler	$\int p(x)\log\frac{p(x)}{q(x)}\,\mathrm{d}x$	$u\log u$	$1+\log\frac{p(x)}{q(x)}$
Reverse KL	$\int q(x)\log\frac{q(x)}{p(x)}\,\mathrm{d}x$	$-\log u$	$-\frac{q(x)}{p(x)}$
Pearson χ^2	$\int\frac{(q(x)-p(x))^2}{p(x)}\,\mathrm{d}x$	$(u-1)^2$	$2(\frac{p(x)}{q(x)}-1)$
Squared Hellinger	$\int\left(\sqrt{p(x)}-\sqrt{q(x)}\right)^2\,\mathrm{d}x$	$(\sqrt{u}-1)^2$	$(\sqrt{\frac{p(x)}{q(x)}}-1)\cdot\sqrt{\frac{q(x)}{p(x)}}$
Jensen-Shannon	$\frac{1}{2}\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x$	$-(u+1)\log\frac{1+u}{2}+u\log u$	$\log\frac{2p(x)}{p(x)+q(x)}$
GAN	$\int p(x)\log\frac{2p(x)}{p(x)+q(x)}+q(x)\log\frac{2q(x)}{p(x)+q(x)}\,\mathrm{d}x-\log(4)$	$u\log u-(u+1)\log(u+1)$	$\log\frac{p(x)}{p(x)+q(x)}$

For each non-trivial f, there is an important paper come out[6].

[6] Least square GAN

Shared limitations

For divergence-based distance, there is a trade off between model covering and perceptual satisfaction:

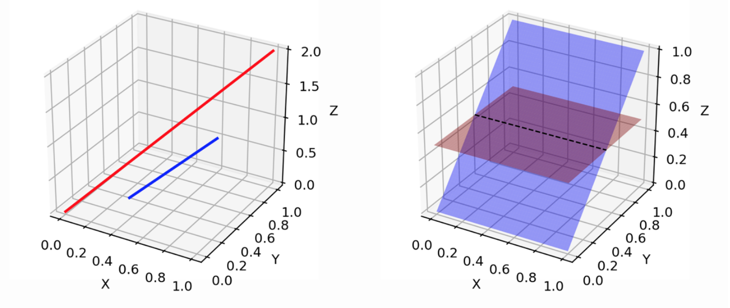


- picking one mode generate good looking images
- captures more modes generate blur images

One possible reason: It is not proper to use JSD to measure the distance of two distributions. [7]

Why?(Three lemma)

- Because P_r and P_g are two lower-dimensional sub-manifolds.
- The probability for P_r and P_g "not perfect align" is 1
- If P_r, P_g don't perfect align, then there is always a perfect discriminator D(Takes 1 on P_r, 0 on P_g).



[7] Towards Principle Methods for Training GAN

Theorem

If P_r and P_g are two lower-dimensional submanifolds, and they don't perfect align(with probability 1), JSD between P_r and P_g is a constant $log2$, regardless of their real distance.

Which means, as the convergence of the discriminator to the optimal, it can't provide any guidance to the optimization of G.

$$\boxed{\text{Under a mild condition, JSD}(P_r||P_g) \text{ is not continuous w.r.t } P_g}$$

Target:

- $P_{g\theta}$ is continuous w.r.t θ
- $d(P_r||P_g)$ is continuous w.r.t P_{g_θ}

Wasserstein GAN[8]

Wasserstein distance(Earth mover distance):

$$W(P_r, P_g) = \inf_{\gamma \in \Pi(P_r, P_g)} \mathbb{E}_{(x,y)\sim\gamma}[||x - y||] \tag{9}$$

Property:

- If g_θ is continuous w.r.t. θ, then $W(P_r, P_{g_\theta})$ is continuous w.r.t θ
- If g_θ is local Lipschitz, $W(P_r, P_{g_\theta})$ is continuous and differentiable a.e.

[8] Wasserstein GAN

Comparison:

- TV Distance: $\delta(P_r||P_g) = \sup_{A \in \sum} |p_r(A) - p_g(A)|$
- KL Divergence: $KL(P_r||P_g) = \int \log(\frac{p_r(x)}{p_g(x)}) p_r(x) d\mu(x)$
- JSD: $JSD(P_r||P_g) = KL(P_r||\frac{1}{2}(P_r + P_g)) + KL(P_g||\frac{1}{2}(P_r + P_g))$

Which tells:

- $\delta(P_r||P_g) \to 0 \iff JSD(P_r||P_g) \to 0$ Norm induced by JSD and TV are equivalent
- $KL(P_g||P_r) \to 0 \Longrightarrow JSD(P_r||P_g) \to 0 \Longrightarrow W(P_r||P_g) \to 0$
- KL gives strongest topology, then comes JSD, W distance gives the weakest topology.

Why?

Because W convergence correspond to convergence in distribution.

Implementation

Kantorovich Rubinstein duality:

$$W(P_r||P_{g_\theta}) = \sup_{||f||_L \leqslant 1} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_{g_\theta}}[f(x)] \tag{10}$$

Now, we can apply paramterize the distance:

$$\min_\theta \max_w \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{z\sim P(z)}[f_w(g_\theta(z))] \tag{11}$$

with the constraint $Lip(f) \leqslant 1$.

How to let the nn satisfies the constraint:

- In WGAN, they use weight clipping, this operation will greatly reduce function space
- In WGAN-GP[9], a better method is proposed.

$$\max_{w} \mathbb{E}_{x\sim P_r}[f_w(x)] - \mathbb{E}_{\tilde{x}\sim P_g}[f_w(\tilde{x})] + \lambda \mathbb{E}_{\hat{x}\sim P_{\hat{x}}}[(||\nabla_{\hat{x}} f_w(\hat{x})||^2 - 1)^2], \quad (12)$$

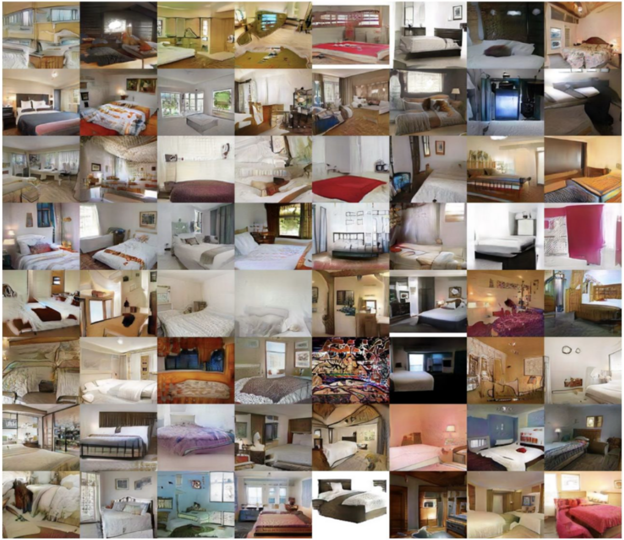
where $\hat{x} = \beta x + (1-\beta)\tilde{x}$.

- robust versus architectures
- generate high quality images
- more cute-edge architectures used resnet, widely used (2000+ citation)

[9] Improved Training of Wasserstein GANs

Robust versus architecture

High quality images:



Spectral Normalization GAN[10]

For now, GANs are able to generate high quality images of small size, next target: Imagenet.
How to revise this Lipschitz constraint.
Consider the discriminator of the form:

$$f(x,\theta) = W^{L+1} a_L(W^L(a_{L-1}(...a_1(W1(x))...))) \tag{13}$$

Three lemma used:

- For a linear function $Y = WX$, the Lipschitz constant M for function is exactly the spectral norm of the matrix W.
- $||h1 \cdot h2||_{Lip} \leqslant ||h1||_{Lip}||h2||_{Lip}$.
- For ReLU, Lipschitz constant is 1.

[10] spectral normalization for generative adversarial networks

pseudo code: normal WGAN, but each linear layer in discriminator is followed by a spectral normalization $W = W/\sigma(W)$.
To avoid heavy computation, they replace SVD with power method, also prove the back-propogation for power method.

Miyato et al 2017

First time able to train on full Imagenet. Simpler than WGAN-GP.

So far, we finish the GANs model section, next section is about comparison with VAE[11] and GMMN.[12][13]

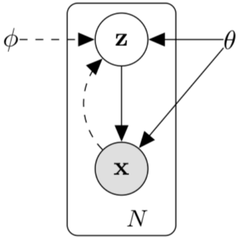
[11] Auto-Encoding Variational Bayes

[12] Training generative neural networks via Maximum Mean Discrepancy optimization

[13] Generative Moment Matching Networks

Comparison of popular generative models

VAE: graphical models, data generation process can be summarized in figure:



Because the intractable of posterior of $P(Z|X)$, so they use an auxiliary normal distribution $Q(Z|X)$ to approximate $P(Z|X)$.

Loss function:

$$\log p_\theta(X) \geqslant E_{z\sim q(z|x)} \log p_\theta(x|z) - D_{KL}(q(z|x)||p_\theta(z)) \quad (14)$$
$$= E_{z\sim q(z|x)} \log p_\theta(z,x) + H(q(z|x)) \quad (15)$$

- The first term is the joint log-likelihood of the complete data under the approximate posterior
- The second term is the entropy of the approximated posterior.If the $q(z|x)$ is taken as a normal distribution, the maximization of the entropy encourage the variance to be bigger, rather than collapse to a single point.

VAEs Pro:

- clear mechanism behind
- no mode collapse
- stable to train

VAEs Cons:

- Generate blurry images (Lots of work claim this is the universal problem for all MLE method)

Moment Matching Networks

GMMN (generative moment matching networks), it contains only one branch, no need of the discriminator or encoder.
Based on What?
If P and Q are same distributions, then all orders moment of the P and Q should be same under any kind of transformation f

$$P = Q \iff \forall f, E_{x \sim p(x)} f(x) = E_{x \sim q(x)} f(x) \tag{16}$$

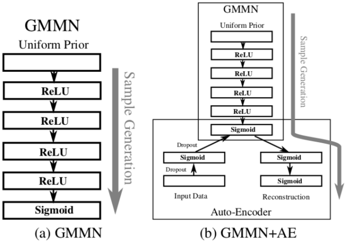
Thus, the measurement of the distance can be formed as

$$L_{MMD}^2 = || \frac{1}{N} \sum_{i=1}^{N} \Phi(x_i) - \frac{1}{M} \sum_{j=1}^{M} \Phi(y_i) ||^2 \tag{17}$$

However, instead of parameterizing the function Φ, $< \Phi(x_i), \Phi(y_j) >$ is replaced with $K(x_i, y_j)$.

Loss function

$$\min_{\theta} \left|\left| \frac{1}{N}\sum_{i=1}^{N}\Phi(x_i) - \frac{1}{M}\sum_{j=1}^{M}\Phi(G_\theta(z_i)) \right|\right|^2 \tag{18}$$



One can also use a learned kernel in which case, loss function becomes:

$$\min_{\theta}\max_{w} \left|\left| \frac{1}{N}\sum_{i=1}^{N}\Phi_w(x_i) - \frac{1}{M}\sum_{j=1}^{M}\Phi_w(G_\theta(z_i)) \right|\right|^2 \tag{19}$$

Understand GANs from Information Theory[14][15]

$Z \sim Ber(\pi)$, P_r real distribution and P_g generated distribution, there is a random variable X satisfies:

$$P(X|Z=0) = P_g, P(X|Z=1) = P_r \tag{20}$$

Target: Seeing lots of samples from both P_r and P_g, you won't be able to infer π.
Method: minimize the mutual information, which is defined as:

$$I(X,Z) = KL(p(x,z)||p(x)p(z)) \tag{21}$$

$\mathrm{I(X,\ Z)} = 0 \iff X,\ Z$ are independent $\iff P_r = P_g$

[14] InfoGAN:Interpretable Representation Learning by Information Maximising Generative Adversarial

[15] How (not) to train your generative model: scheduled sampling, likelihood, adversarial?

$$
\begin{aligned}
I(X,Z) &= H(Z) + \mathbb{E}_X \mathbb{E}_{Z|X} \log q(z|x) + \mathbb{E}_X KL[p(z|x)||q(y|x)] \\
&= \max_q H(Z) + \mathbb{E}_X \mathbb{E}_{Z|X} \log q(z|x)
\end{aligned} \tag{22}
$$

$$
\begin{aligned}
I(X,Z) &\geqslant H(Z) + \max_\Psi \mathbb{E}_{X,Z} \log q(z|x;\Psi) \\
&= H(Z) + \max_\Psi \pi \mathbb{E}_{P_r} \log q(1|x;\Psi) + (1-\pi)\mathbb{E}_{P_g} \log q(0|x;\Psi) \\
\min I(X,Z) &\Longrightarrow \min_{g_\theta} \max_\Psi \pi \mathbb{E}_{P_r} \log q(1|x;\Psi) + (1-\pi)\mathbb{E}_{P_g}(1 - \log q(1|x;\Psi))
\end{aligned} \tag{23}
$$

If $\pi = 1/2$, this minimization of mutual information gives loss function of vanilla GAN.

A unfied model[16]

It is always an elegant thing to give an unify model for various generative models:

Integral Probability Metrics

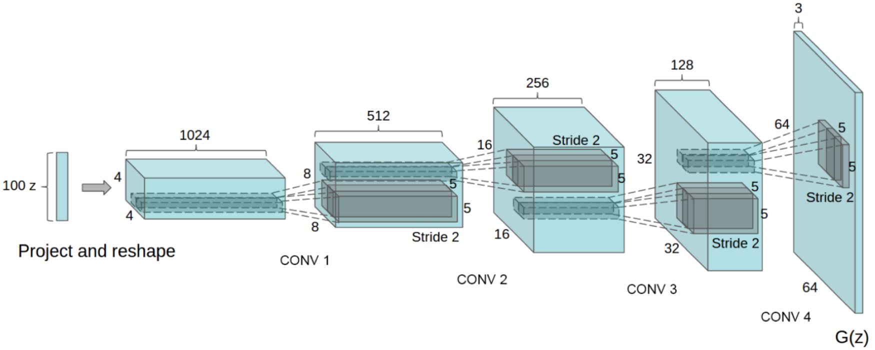
$$\gamma_F(P_r, P_g) := \sup_{f \in F} \left| \int_M f dP_r - \int_M f dP_g \right| \tag{24}$$

- Wasserstein distance: $F = \{f : ||f||_L \leqslant 1\}$
- TV distance or Kolmogorov distance: $F = \{f : ||f||_\infty \leqslant 1\}$
- MMD:$F = \{f : ||f||_H \leqslant 1\}$

[16] Non-parametric Estimation of Integral Probability Metrics

Content

DCGAN[17]



17 Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks

[from lecture slides of UCB]

Supervised Learning CNNs not directly usable

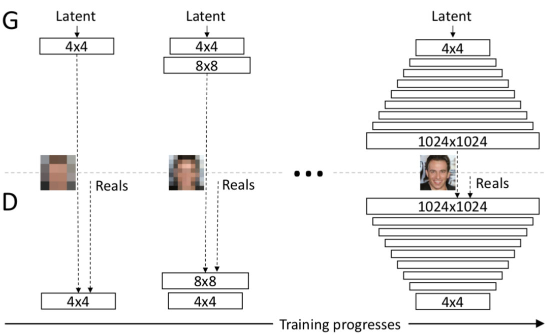
- Remove max-pooling and mean-pooling
- Upsample using transposed convolutions in the generator
- Downsample with strided convolutions and average pooling
- Non-Linearity: ReLU for generator, Leaky-ReLU (0.2) for discriminator
- Output Non-Linearity: tanh for Generator, sigmoid for discriminator
- Batch Normalization used to prevent mode collapse
- Batch Normalization is not applied at the output of G and input of D

Optimization details

- Adam: small LR - 2e-4; small momentum: 0.5, batch-size: 128

First visually accepted results:

Progressive GAN[18]



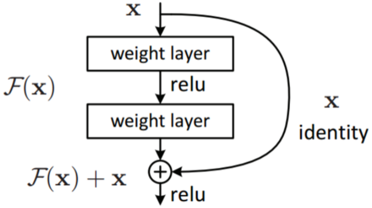
- WGAN-GP framework + Engineering work
- For G: nearest neighbor filtering, for D: avg-pooling
- Progressive adding resolution for G and D
- Batch normalization is important
- We adding new layer for G and D, previous layers are trainable.

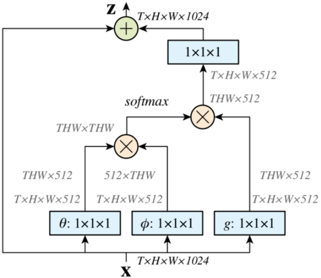
[18] Progressive Growing of GANs for Improved Quality, Stability, and Variation

Really exciting results images of size 1024×1024

It is widely accepted that a conditional version of GAN help the generating tasks, for example: conditional GAN[19], AC-GAN[20], BigGAN[21].
For BigGAN:

- Residual block are used
- Non-local block are used
- Constrains the Lipschitz constant via an implicit regularizer (Compared with SN-GAN): $||W^T W - I||_F^2$ in the loss.

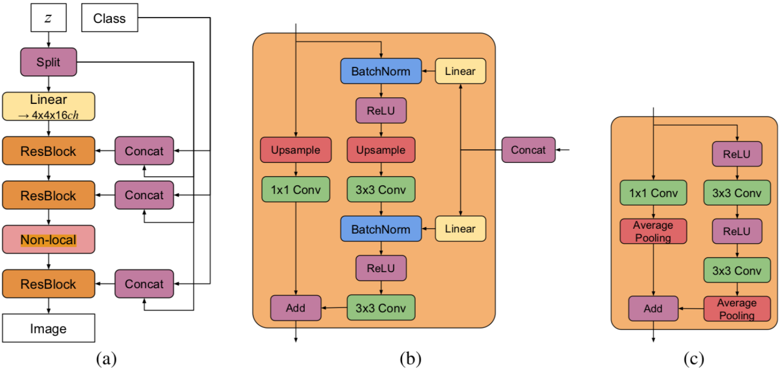




[19] Conditional Generative Adversarial Nets

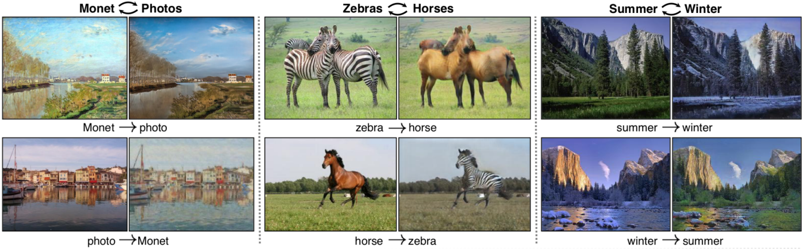
[20] Conditional Image Synthesis With Auxiliary Classifier GANs

Architecture of BigGAN



Other applications

Cycle GAN[22]: 4000+ citation, widely use in image to image translation, combining with U-net is a very powerful tool in medical image processing: Cross-modality image synthesis.



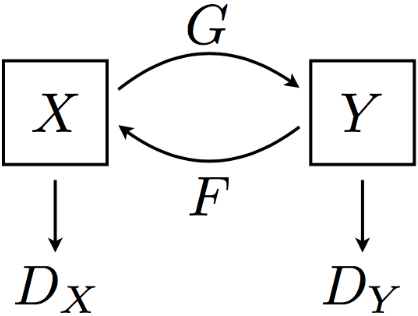
Unsupervised framework: no need of image pairs during training.

[22] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

apple → orange

orange → apple

Vanilla GANs can transform the style but can't keep the content



Aiming at:

$$F(G(x)) = x, \ \forall x \in X \quad G(F(y)) = y, \ \forall y \in Y \tag{25}$$

The loss function for Generator

$$L(G, F, D_X, D_Y) = L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_x, Y, X) + \lambda L_{cyc}(G, F)$$

in which

$$L_{GAN}(G, D_Y, X, Y) = \mathrm{E}_{y \sim P_Y}[\log D_Y(y)] + \mathrm{E}_{x \sim P_x}[\log(1 - D_Y(G(x))]$$
$$L_{cyc}(G, F) = \mathrm{E}_{x \sim P_x}[||F(G(x)) - X||_1] + \mathrm{E}_{y \sim P_y}[||G(F(y)) - y||_1]$$

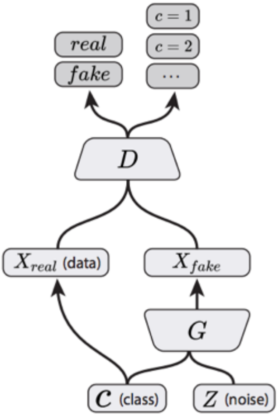
$$G^*, F^* = \arg\min_{F,G} \max_{D_x, D_y} L(G, F, D_x, D_y)$$

Adversarial samples[23]: Perturbation-based adversarial examples: mis-classified images that lie on the neighbor of a correctly-classified images.
Unrestricted adversarial examples: images which are classified differently from oracle.

Different adversarial examples:

- formulation: WGAN-GP
- architecture: AC-GAN[24]

[23] Constructing Unrestricted Adversarial Examples with Generative Models
[24] Conditional Image Synthesis With Auxiliary Classifier GANs



AC-GAN
(Present Work)

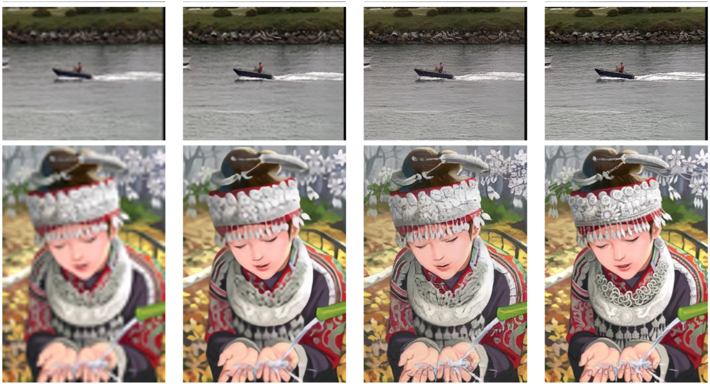
Loss function for WGAN-GP

$$\max_{w} \mathbb{E}_{x \sim P_r}[f_w(x)] - \mathbb{E}_{\tilde{x} \sim P_g}[f_w(\tilde{x})] + \lambda \mathbb{E}_{\hat{x} \sim P_{\hat{x}}}[(||\nabla_{\hat{x}} f_w(\hat{x})||^2 - 1)^2], \quad (26)$$

Loss for adversarial attack:

$$\begin{aligned} l_2 &= \log c(y_{source} | g(z, y_{source})) \\ l_1 &= \log f(y_{target} | g(z, y_{source})) \\ l_o &= \frac{1}{m} \sum_{i=1}^{m} \max(|z_i - z_i^0| - \epsilon, 0) \end{aligned} \quad (27)$$

image super-resolution[25] (3000+ citation)

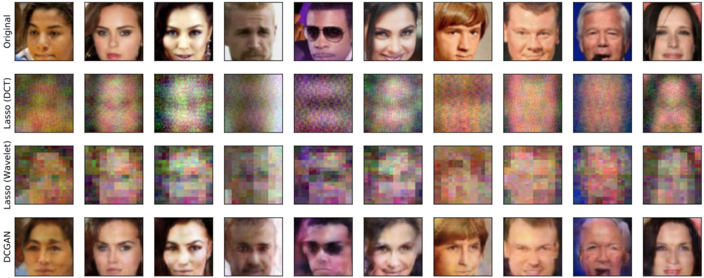


Reconstruct 4 pixels from 1 pixel.

[25] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

Compressive Sensing[26]

Compressed Sensing using Generative Models: faster convergence rate + better results.



Reconstruction from 500 measurements (of $n = 12288$ dimensional vector)

[26] Compressed Sensing using Generative Models