Theory and Application of Generative Adversarial Networks

Ming-Yu Liu, Julie Bernauer, Jan Kautz
NVIDIA

Tutorial schedule

- 1:30 - 2:50 Part 1(Generative modeling, GAN training)
- 2:50 - 3:20 Demo
- 3:30 - 4:00 Break
- 4:00 - 5:30 Part 2 (Video and image translation)

Before we start

- Due to the exponential growth, we will miss several important GAN works.
- The GAN Zoo https://deephunt.in/the-gan-zoo-79597dc8c347

Image from Deep Hunt, blog by Avinash Hindupur

Outlines

1. Introduction
2. GAN objective
3. GAN training
4. Joint image distribution and video distribution
5. Computer vision applications

1. Introduction

1. Introduction

1. Generative modeling
2. Gaussian mixture models
3. Manifold assumption
4. Variational autoencoders
5. Autoregressive models
6. Generative adversarial networks
7. Taxonomy of generative models

Generative modeling

- Density estimation

- Sample generation

Images & slide credit Goodfellow 2016

Example: Gaussian mixture models

Step 1: observe a set of samples

Step 2: assume a GMM model

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

Step 3: perform maximum likelihood learning

$$\max_\theta \sum_{x^{(j)} \in \text{Dataset}} \log p(\theta|x^{(j)})$$

Example: Gaussian mixture models

$$p(x|\theta) = \sum_i \pi_i \mathcal{N}(x|\mu_i, \Sigma_i)$$

1. [Density Estimation]

2. [Sample Generation]

Good for low-dimensional data

Manifold assumption

- Data lie approximately on a manifold of much lower dimension than the input space.

Images credit Ward, A. D. et al 2007

Manifold examples

Frey Face dataset:

MNIST

Slide credit, Courville 2017

Generative modeling via autoencoders and GMM

- Two stage process
  1. Apply an autoencoder to embed images in a low dimensional space.

$$\min_{E,D} \sum_{x^{(j)}} ||x^{(j)} - D(E(x^{(j)}))||^2$$

  2. Fit a GMM to the embeddings $E(x^{(j)})$
- Drawbacks
  - Not end-to-end
  - $E(x^{(j)})$ can still be high-dimensional
  - Tend to memorize samples.

Vincent et al. JMLR 2011

Variational autoencoders

- Kingma and Welling, “Auto-encoding variational Bayes," ICLR 2014
- Rezende, Mohamed, and Wierstra, “Stochastic back-propagation and variational inference in deep latent Gaussian models,” ICML 2014

- Derive from a variational framework
  - Constrain the encoder to output a conditional Gaussian distribution

$$E(x^{(j)}) \sim q_\theta(z|x^{(j)}) = \mathcal{N}(z|\mu(x^{(j)}), \Sigma(x^{(j)}))$$

  - The decoder reconstructs inputs from samples from the conditional distribution

$$D(z^{(j)}) \sim p_\theta(x^{(j)}|z^{(j)} \sim q_\theta(z|x^{(j)}))$$

Maximizing a variational lower bound

Ideally, we would like to maximize

$$\max_{\theta} L(\theta|\text{Dataset}) \equiv \sum_{x^{(j)}} \log p(\theta|x^{(j)})$$

But we maximize a lower bound for tractability

$$\max_{\theta} L_V(\theta|\text{Dataset}) \equiv$$

$$\sum_{x^{(j)}} -\text{KL}(q_\theta(z|x^{(j)})||\mathcal{N}(z|0,I)) +$$

$$E_{z\sim(q_\theta(z|x^{(j)})}[\log p_\theta(x^{(j)}|z)]$$

$$\leq L(\theta|\text{Dataset})$$

Reparameterization trick

- Make sampling a differentiable operation
- The variational distribution $q_\theta(z|x) = \mathcal{N}(z|\mu(x), \Sigma(x))$
- Sampling $z = \mu(x) + \Sigma(x)\epsilon, \text{where } \epsilon \sim \mathcal{N}(0, I)$

Image credit, Courville 2017

Samples from variational autoencoders

LFW dataset

ImageNet dataset

Slide credit, Courville 2017

Why are generated samples blurry?

- Pixel values are modeled as a conditional Gaussian, which leads to Euclidean loss minimization.

$$\log p_\theta(x^{(j)}|z) = -||x^{(j)} - D(z)||^2 + Const$$

- Regress to the mean problem and rendering blurry image.

- Difficult to hand-craft a good perceptual loss function.

Pitfall of Euclidean distance for image modeling

- Blue curve plots the Euclidean distance between a reference image and its horizontal translation.
- Red curve is the Euclidean distance between and

Autoregressive generative models

- Choose an ordering of the dimensions in x.
- Define the conditionals in the product rule expression of p(x).

$$p(\mathbf{x}) = \prod_{k=1}^{D} p(x_k | \mathbf{x}_{<k})$$

- Properties
  - Pros: p(x) is tractable, so easy to train, easy to sample (though slower)
  - Cons: doesn't have a natural latent representation

Slide credit, Courville 2017

PixelCNNs

- Oord, Aaron van den, et al. 2016
- Use masked convolution to enforce the autoregressive relationship.
- Minimizing cross entropy loss instead of Euclidean loss.

$$p(x_i \mid \mathbf{x}_{<i})$$

Slide credit, Courville 2017

PixelCNNs

$$p(x_i \mid \mathbf{x}_{<i}) = p(x_{i,R} \mid \mathbf{x}_{<i})p(x_{i,G} \mid x_{i,R}, \mathbf{x}_{<i})p(x_{i,B} \mid x_{i,R}, x_{i,G}, \mathbf{x}_{<i})$$

autoregressive over color channels

8-bits pixel values (multinoulli distribution)

Slide credit, Courville 2017

Example results from PixelCNNs

Slide credit, Courville 2017

Parallel multiscale autoregressive density estimation

- Reed et al. ICML 2017
- Can we speed generation time of PixelCNNs?
  - Yes, via multiscale generation

Slide credit, Courville 2017

Parallel multiscale autoregressive density estimation

- Also seems to help provide better global structure.

Slide credit, Courville 2017

Generative adversarial networks

- Goodfellow et al. NIPS 2014
- Forget about designing a perceptual loss. Let's train a discriminator to differential real and fake image.

Deep convolutional generative adversarial networks (DCGANs)

By Redford et al. ICLR 2016

Results from DCGANs

Z space interpolation

Vector space arithmetic

(Radford et al, 2015)

Slide credit, Goodfellow 2016

GANs vs VAEs

Taxonomy of generative models

Slide credit Goodfellow 2016

Quick recaps

- Generative modeling
- Manifold assumption
- VAEs, PixelCNNs, GANs
- Taxonomy

Outlines

1. Introduction
2. **GAN objective**
3. GAN training
4. Joint image distribution and video distribution
5. Computer vision applications

2. GAN objective

2. GAN objective

1. GAN objective
2. Discriminator strategy
3. GAN theory
4. Effect of limited discriminator capacity

GAN objectives

Solving a minimax problem

$$\min_G \max_D \quad E_{x \sim p_X}[\log D(x)] + E_{z \sim p_Z}[\log(1 - D(G(z))]$$

p_X : Data distribution,
usually represented by samples.

$p_{G(Z)}$: Model distribution, where
Z is usually modeled as uniform or Gaussian.

Alternating gradient updates

- Step 1: Fix G and perform a gradient step to

$$\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z))]$$

- Step 2: Fix D and perform a gradient step to (in theory)

$$\min_G E_{z\sim p_Z}[\log(1 - D(G(z))]$$

(in practice)

$$\max_G E_{z\sim p_Z}[\log D(G(z))]$$

Discriminator strategy

- Optimal (non-parametric) discriminator

$$D(x) = \frac{p_X(x)}{p_X(x) + p_{G(Z)}(x)}$$

Estimating this ratio using supervised learning is the key approximation mechanism used by GANs

Slide credit Goodfellow 2016

Learning a Gaussian mixture model using GANs

- Red points: true samples; samples from data distribution
- Blue points: fake samples samples from model distribution
- Z-space: a Gaussian distribution of *N(0,I)*
- The model first learns 1 mode, then 2 modes, and finally 3 modes.

GAN theory

- Optimal (non-parametric) discriminator

$$D(x) = \frac{p_X(x)}{p_X(x) + p_{G(Z)}(x)}$$

- Under an ideal discriminator, the generator minimizes the Jensen-Shannon divergence between p_X and $p_{G(Z)}$. This also requires that *D* and *G* have sufficient capacity and a sufficiently large dataset.

$$JS(p_X|p_{G(Z)}) = \quad KL(p_X||\frac{p_X + p_{G(Z)}}{2}) + KL(p_{G(Z)}||\frac{p_X + p_{G(Z)}}{2})$$

$$KL(p_X|p_{G(Z)}) = E_{p_X}[\log \frac{p_X(x)}{p_{G(Z)}(x)}]$$

In practice

- Dataset sizes are limited.
- Finite network capacity
- (Sanjeev Arora et al. ICML 2017)

For a discriminator with capacity *n*, the generator only need to remembers $C\frac{n}{\epsilon^2}\log n$ samples to be epsilon away from the optimal objective.

Consequences:

1. The generator does not need to generalize.
2. Larger training datasets have limited utility.

Capacity meaning # of trainable parameters.

Effect of limited discriminator capacity

CIFAR10 dataset

Image credit Arora and Zhang 2017

Quick Recaps

- Discriminator strategy
- GAN theory
- Effect of limited discriminator capacity

$$D(x) = \frac{p_X(x)}{p_X(x) + p_{G(Z)}(x)}$$

Outlines

1. Introduction
2. GAN objective
3. **GAN training**
4. Joint image distribution and video distribution
5. Computer vision applications

3. GAN Training

3. GAN Training

1. Non-convergence in GANs
2. Mode collapse
3. Lake of global structure
4. Tricks
5. New objective functions
6. Surrogate objective functions
7. Network architectures

Non-convergence in GANs

- GAN training is theoretically guaranteed to converge if we can modify the density functions directly, but
  - Instead, we modify *G* (sample generation function)
  - We represent *G* and *D* as highly non-convex parametric functions.
- “Oscillation”: can train for a very long time, generating very many different categories of samples, without clearly generating better samples.
- Mode collapse: most severe form of non-convergence

Slide credit Goodfellow 2016

Mode collapse

$$\min_G \max_D V(G, D) \neq \max_D \min_G V(G, D)$$

- D in inner loop: convergence to correct distribution
- G in inner loop: place all mass on most likely point

Figure credit, Metz et al, 2016
Slide credit Goodfellow 2016

Mode collapse

- Same data dimension.
- Performance correlated with number of modes.
- Performance correlated with manifold dimensions.

Mode collapse

- Same number of modes.
- Performance correlated with manifold dimensions.
- Performance non-correlated with data dimensions.

Problem with global structure

Slide credit Goodfellow 2016

Problem with counting

(Goodfellow 2016)

Slide credit Goodfellow 2016

Improve GAN training

Tricks
- Label smoothing
- Historical batches
- ...

New objectives
- EBGAN
- LSGAN
- WGAN
- BEGAN
- fGAN
- ...

Surrogate or auxiliary objective
- UnrolledGAN
- WGAN-GP
- DRAGAN
- ...

Network architecture
- LAPGAN
- ...

Improve GAN training

Tricks
- Label smoothing
- Historical batches
- ...

New objectives
- EBGAN
- LSGAN
- WGAN
- BEGAN
- fGAN
- ...

Surrogate or auxiliary objective
- UnrolledGAN
- WGAN-GP
- DRAGAN
- ...

Network architecture
- LAPGAN
- ...

Tricks - one sided label smoothing

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, "Improved techniques for training GANs", NIPS 2016

$$E_{x \sim p_X}[0.9 \log D(x)] + E_{z \sim p_Z}[\log(1 - D(G(z)))]$$

- Does not reduce classification accuracy, only confidence.
- Prevents discriminator from giving very large gradient signal to generator.
- Prevents extrapolating to encourage extreme samples.
- Two sided label smoothing https://github.com/soumith/ganhacks

Tricks - historical generator batches

A. Srivastava, T. Ofister, O. Tuzel, J. Susskind, W. Wang, R. Webb, “Learning from Simulated and Unsupervised Images through adversarial training”. CVPR’17

Help stabilize the discriminator training in the early stage.

Other tricks

- https://github.com/soumith/ganhacks
  - Use labels
  - Normalize the inputs to [-1,1]
  - Use TanH
  - Use BatchNorm
  - Avoid sparse gradients: ReLU, MaxPool
  - …
- Salimans et al. NIPS 2016
  - Use Virtual BatchNorm
  - MiniBatch Discrimination
  - …

Improve GAN training

Tricks
- Label smoothing
- Historical batches
- ...

New objectives
- EBGAN
- LSGAN
- WGAN
- BEGAN
- fGAN
- ...

Surrogate or auxiliary objective
- UnrolledGAN
- WGAN-GP
- DRAGAN
- ...

Network architecture
- LAPGAN
- ...

EBGAN

Junbo Zhao, Michael Mathieu, Yann LeCun, “Energy-based generative adversarial networks,” ICLR 2017

Energy function can be modeled by an auto-encoder.

EBGAN

Junbo Zhao, Michael Mathieu, Yann LeCun, “Energy-based generative adversarial networks,” ICLR 2017

$$En(x) = ||Decoder(Encoder(x)) - x||$$

Discriminator objective

$$\min_{En} E_{x \sim p_X}[En(x)] + E_{z \sim p_Z} E[[m - En(G(z))]^+]$$

Generator objective

$$\min_{G} E_{z \sim p_Z} E[En(G(z))]$$

EBGAN

Junbo Zhao, Michael Mathieu, Yann LeCun, “Energy-based generative adversarial networks,” ICLR 2017

LSGAN

X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, “Least squares generative adversarial networks” 2016

Still use a classifier but replace cross-entropy loss with Euclidean loss.

Discriminator

GAN $$\min_D E_{x\sim p_X}[-\log D(x)] + E_{z\sim p_Z}[-\log(1 - D(G(z)))]$$

LSGAN $$\min_D E_{x\sim p_X}[(D(x) - 1)^2] + E_{z\sim p_Z}[D(G(z))^2]$$

Generator

GAN $$\min_G E_{z\sim p_Z}[-\log D(G(z))]$$

LSGAN $$\min_G E_{z\sim p_Z}[(D(G(z)) - 1)^2]$$

LSGAN

X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, “Least squares generative adversarial networks” 2016

GAN: minimize Jensen-Shannon divergence between p_X and $p_{G(Z)}$

$$JS(p_X||p_{G(Z)}) = KL(p_X||\frac{p_X + p_{G(Z)}}{2}) + KL(p_{G(Z)}||\frac{p_X + p_{G(Z)}}{2})$$

LSGAN: minimize chi-square distance between $p_X + p_{G(Z)}$ and $2p_{G(Z)}$

$$\chi^2(\frac{p_X + p_{G(Z)}}{2}||p_{G(Z)}) \quad = \int_X \frac{2p_{G(Z)}(x) - (p_{G(Z)}(x) + p_X(x))}{p_{G(Z)}(x) + p_X(x)} dx$$

LSGAN

X. Mao, Q. Li, H. Xie, R. Lau, Z. Wang, “Least squares generative adversarial networks” 2016

Wasserstein GAN

M. Arjovsky, S. Chintala, L. Bottou “Wasserstein GAN” 2016

Replace classifier with a critic function

Discriminator

GAN $$\max_D E_{x\sim p_X}[\log D(x)] + E_{z\sim p_Z}[\log(1 - D(G(z))]$$

WGAN $$\max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(z))]$$

Generator

GAN $$\max_G E_{z\sim p_Z}[\log D(G(z))]$$

WGAN $$\max_G E_{z\sim p_Z}[D(G(z))]$$

Wasserstein GAN

M. Arjovsky, S. Chintala, L. Bottou “Wasserstein GAN” 2016

GAN: minimize Jensen-Shannon divergence between p_X and $p_{G(Z)}$

$$JS(p_X||p_{G(Z)}) = KL(p_X||\frac{p_X + p_{G(Z)}}{2}) + KL(p_{G(Z)}||\frac{p_X + p_{G(Z)}}{2})$$

WGAN: minimize earth mover distance between p_X and $p_{G(Z)}$

$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x - y||]$$

Wasserstein GAN

M. Arjovsky, S. Chintala, L. Bottou “Wasserstein GAN” 2016

GAN vs WGAN

Slide credit, Courville 2017

GAN vs WGAN

$$JS(p_X||p_{G(Z)}) = KL(p_X||\frac{p_X + p_{G(Z)}}{2}) + KL(p_{G(Z)}||\frac{p_X + p_{G(Z)}}{2})$$

Jesen-Shannon divergence in this example

$$JS(p_X||p_{G(Z)}) = \begin{cases} \log 2 & \text{if } \theta \neq 0, \\ 0 & \text{if } \theta = 0, \end{cases}$$

Example from Arjovsky et al. 2017

Slide credit, Courville 2017

GAN vs WGAN

$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x-y||]$$

Example from Arjovsky et al. 2017

Earth Mover distance in this example

$$EM(p_X, p_{G(Z)}) = |\theta|$$

θ

Slide credit, Courville 2017

GAN vs WGAN

Example from Arjovsky et al. 2017

- If we can directly change the density shape parameter, the Earth Mover distance is smoother.
- But we do not directly change the density shape parameter, we change the generation function.

Boundary Equilibrium GAN (BEGAN)

David Berthelot, Thomas Schumm, Luke Metz, “Boundary equilibrium generative adversarial networks” 2017

- Use EBGAN autoencoder-based discriminator.
- Not based on minimizing distance between p_X and $p_{G(Z)}$
- Based on minimizing distance between $p_{En(X)}$ and $p_{En(G(Z))}$
- The distance is a lower bound on the Wasserstein 1 (Earth Mover) distance.

Boundary Equilibrium GAN (BEGAN)

David Berthelot, Thomas Schumm, Luke Metz, “Boundary equilibrium generative adversarial networks” 2017

WGAN

$$EM(p_X, p_{G(Z)}) = \inf_{\gamma \in \prod(p_X, p_{G(Z)})} E_{(x,y)\sim\gamma}[||x - y||]$$

BEGAN

$$EM(p_{En(X)}, p_{En(G(Z))}) = \inf_{\gamma \in \prod(p_{En(X)}, p_{En(G(Z))})} E_{(x_1,x_2)\sim\gamma}[|x_1 - x_2|]$$

But low bound

$$\inf E_{(x_1,x_2)\sim\gamma}[|x_1 - x_2|] \geq \inf |E_{(x_1,x_2)\sim\gamma}[x_1 - x_2]| = \left|mean(x_1) - mean(x_2)\right|$$

Final objective function

$$\max_G \min_D \left| E_{x\sim p_X}[En(x)] - E_{z\sim p_Z}[En(G(z))] \right|$$

Boundary Equilibrium GAN (BEGAN)

David Berthelot, Thomas Schumm, Luke Metz, “Boundary equilibrium generative adversarial networks” 2017

$$\text{Discriminator objective:} \quad \min_{En} En(x) - k_t En(G(z))$$

$$\text{Generator objective:} \quad \min_{G} En(G(z))$$

$$\text{Equilibrium objective:} \quad k_{t+1} = k_t + \lambda_k(\gamma En(x) - En(G(z)))$$

$$\text{Hyperparamete:} \quad \gamma = \frac{En(G(z))}{En(x)}$$

value between 0 and 1

Boundary Equilibrium GAN (BEGAN)

David Berthelot, Thomas Schumm, Luke Metz, “Boundary equilibrium generative adversarial networks” 2017

(a) EBGAN (64x64)

(b) Our results (128x128)

Improve GAN training

Tricks
- Label smoothing
- Historical batches
- ...

New objectives
- EBGAN
- LSGAN
- WGAN
- BEGAN
- fGAN
- ...

Surrogate or auxiliary objective
- UnrolledGAN
- WGAN-GP
- DRAGAN
- ...

Network architecture
- LAPGAN
- ...

UnrolledGAN

L. Metz, B. Poole, D. Pfau, J. Sohl-Dickstein, “Unrolled generative adversarial networks” ICLR 2017

- Considering several future updates of the discriminator as updating the generator.
- Update the discriminator in the same way.
- Avoid that the generator generates samples from few modes.

UnrolledGAN

Surrogate objective $$\max_G E_{z\sim p_Z}[\log D_K(G(z))]$$

- D_K if a function of D and G
- Considering several future updates of the discriminator as updating the generator.

WGAN-GP

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Domoulin, A. Courville “Improved Training of Wasserstein GANs” 2017

$$\min_G \max_D E_{x\sim p_X}[D(x)] - E_{z\sim p_Z}[D(G(Z))] + \lambda E_{y\sim p_Y}[(||\nabla_y D(y)||_2 - 1)^2]$$

$$y = ux + (1-u)G(z)$$

- *y:* imaginary samples

Optimal critic has unit gradient norm almost everywhere

DCGAN	LSGAN	WGAN (clipping)	WGAN-GP (ours)
Baseline (G: DCGAN, D: DCGAN)			

DRAGAN

N. Kodali, J. Abernethy, J. Hays, Z. Kira “How to train your DRAGAN” 2017

$$\min_G \max_D E_{x\sim p_X}[\log D(x)] - E_{z\sim p_Z}[\log(1 - D(G(Z)))] + \lambda E_{y\sim p_Y}[(||\nabla_y D(y)||_2 - 1)^2]$$

$$y = \alpha x + (1-\alpha)(x+\delta)$$ *y:* imaginary samples around true sample

Improve GAN training

Tricks
- Label smoothing
- Historical batches
- ...

New objectives
- EBGAN
- LSGAN
- WGAN
- BEGAN
- fGAN
- ...

Surrogate or auxiliary objective
- UnrolledGAN
- WGAN-GP
- DRAGAN
- ...

Network architecture
- LAPGAN
- ...

LAPGAN

E. Denton, S. Chintala, A. Szlam, R. Fergus “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks” NIPS 2015

Testing

LAPGAN

E. Denton, S. Chintala, A. Szlam, R. Fergus “Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks” NIPS 2015

Training of each resolution is performed independent of the others.

Evaluation

- Popular metrics for GAN evaluation.
  - Inception loss
    - Train an accurate classifier
    - Train a conditional image generation model.
    - Check how accurate the classifier can recognize the the generated images.
  - Human evaluation
- Generative models are difficult to evaluate. (This et al. ICLR 2016)
- You can hack each evaluation metric.
- Still an open problem.

Tutorial slides will be made available in https://github.com/mingyuliutw

Outlines

1. Introduction
2. GAN objective
3. GAN training
4. **Joint image distribution and video distribution**
5. Computer vision applications

4. Joint image distribution and video distribution

4. Joint image distribution and video distribution

1. Joint image distribution learning
2. Video distribution learning

Joint distribution of multi-domain images

- $p(X_1, X_2, \ldots, X_N)$: where X_i are images of the scene in different modalities.
- Ex. $p(X_{color}, X_{thermal}, X_{depth})$:
- In this presentation, “modality” = “domain”

Joint distribution of multi-domain images

- Define domain by attribute.
- Multi-domain images are views of an object with different attributes.
- $p(X_1, X_2, \ldots, X_N)$: where X_i are images of the object with different attributes.

Non-smiling Smiling Non-beard Beard Young Senior Font#1 Font#2

images Hand-drawings summer winter

Extending GAN for joint image distribution learning

What if we do not have paired images from different domains for learning?

CoGAN: Coupled Generative Adversarial Networks

Ming-Yu Liu, Oncel Tuzel “Coupled Generative Adversarial Networks” NIPS 2016

Learning a joint distribution of images using samples from marginal distributions

Video GANs

C. Vondrick, H. Pirsiavash, A. Torralba "Generating videos with scene dynamics" NIPS 2016

MoCoGANs

S. Tulyakov, M. Liu, X. Yang, J. Kautz “MoCoGAN: Decomposing Motion and Content for Video Generation” 2017

MoCoGANs

S. Tulyakov, M. Liu, X. Yang, J. Kautz “MoCoGAN: Decomposing Motion and Content for Video Generation” 2017

Sampled content

$$\mathbf{Z}_\mathrm{C} = [\mathbf{z}_\mathrm{C}, \mathbf{z}_\mathrm{C}, ..., \mathbf{z}_\mathrm{C}]$$

Motion trajectory

$$\mathbf{Z}_\mathrm{M} = [\mathbf{z}_\mathrm{M}^{(1)}, \mathbf{z}_\mathrm{M}^{(2)}, ..., \mathbf{z}_\mathrm{M}^{(K)}]$$

MoCoGANs

S. Tulyakov, M. Liu, X. Yang, J. Kautz “MoCoGAN: Decomposing Motion and Content for Video Generation” 2017

MoCoGANs

S. Tulyakov, M. Liu, X. Yang, J. Kautz “MoCoGAN: Decomposing Motion and Content for Video Generation” 2017

Training:

$$\min_{D_\mathrm{I}} \mathbb{E}_{\mathbf{v}\sim p_\mathrm{V}}[-\log D_\mathrm{I}(S_1(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}\sim p_{\tilde{\mathrm{V}}}}[-\log(1 - D_\mathrm{I}(S_1(\tilde{\mathbf{v}})))]$$

$$\min_{D_\mathrm{V}} \mathbb{E}_{\mathbf{v}\sim p_\mathrm{V}}[-\log D_\mathrm{V}(S_\mathrm{T}(\mathbf{v}))] + \mathbb{E}_{\tilde{\mathbf{v}}\sim p_{\tilde{\mathrm{V}}}}[-\log(1 - D_\mathrm{V}(S_\mathrm{T}(\tilde{\mathbf{v}})))]$$

$$\max_{G_\mathrm{I}, R_\mathrm{M}} \mathbb{E}_{\tilde{\mathbf{v}}\sim p_{\tilde{\mathrm{V}}}}[-\log(1 - D_\mathrm{I}(S_1(\tilde{\mathbf{v}})))] + \mathbb{E}_{\tilde{\mathbf{v}}\sim p_{\tilde{\mathrm{V}}}}[-\log(1 - D_\mathrm{V}(S_\mathrm{T}(\tilde{\mathbf{v}})))]$$

MoCoGANs

S. Tulyakov, M. Liu, X. Yang, J. Kautz “MoCoGAN: Decomposing Motion and Content for Video Generation” 2017

VGAN vs. MoCoGAN

	VGAN	**MoCoGAN**
Video generation	Directly generating 3D volume	Video frames are generated sequentially.
Generator	Motion dynamic video and static background	Image
Discriminator	One discriminator - 3D CNN for clips	Two discriminators - 3D CNN for clips - 2D CNN for frames
Variable length	No	Yes
Motion and Content Separation	No	Yes
User preference on generated face video quality	5%	95%

Outlines

1. Introduction
2. GAN objective
3. GAN Training
4. Video generation models
5. **Computer vision applications**

5. Computer vision applications

Why are GANs useful for computer vision?

Hand-crafted features → Deep Networks

Hand-crafted objective function → Generative Adversarial Networks

Image-to-image Translation

- Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two different image domains (day-time image and night-time image domains).
- Let $x_1 \in \mathcal{X}_1$.
- Image-to-image translation concerns the problem of translating x_1 to a *corresponding* image $x_2 \in \mathcal{X}_2$
- Correspondence can mean different things in different contexts.

Examples and use cases

Low-res to high-res

Blurry to sharp

Thermal to color

Synthetic to real

LDR to HDR

Noisy to clean

Image to painting

Day to night

Summer to winter

- Bad weather to good weather
- Greyscale to color
- …

Prior Works

- Image-to-image translation has been studied for decades in computer vision.
- Different approaches have been exploited including:
  - Filtering-based approaches
  - Optimization-based approaches
  - Dictionary learning-based approaches
  - Deep learning-based approaches
  - GAN-based approaches.

Generative Adversarial Networks (GANs)

Goodfellow et al. NIPS'14

- Training is done via applying alternating stochastic gradient updates to the following two learning problems.

$$\max_G E_{z \sim p_{\mathcal{N}}}[\log D(G(z))]$$

$$\max_D E_{x \sim p_{\mathcal{X}}}[\log D(X)] + E_{z \sim p_{\mathcal{N}}}[\log(1 - D(G(z)))]$$

- Effect: minimizing JSD between $p_{G(z), z \sim \mathcal{N}}$ and $p_{\mathcal{X}}$

Conditional GANs

- Training is done via applying alternating stochastic gradient updates to the following two learning problems.

$$\max_F E_{x_1 \sim p_{\mathcal{X}_1}} [\log D(F(x_1))]$$

$$\max_D E_{x_2 \sim p_{\mathcal{X}_2}} [\log D(x_2)] + E_{x_1 \sim p_{\mathcal{X}_1}} [\log(1 - D(F(x_1)))]$$

- Effect: minimizing JSD between $p_{F(x_1), x_1 \sim \mathcal{X}_1}$ and $p_{\mathcal{X}_2}$

Conditional GAN for Image-to-image Translation

- Conditional GAN alone is <u>insufficient</u> for image translation.
- No guarantee that the conditionally generated image is related to the source image in a desired way.
- In the supervised setting,
  - $\text{Dataset} = \{(x_1^{(1)}, x_2^{(1)}), (x_1^{(2)}, x_2^{(2)}), ..., (x_1^{(N)}, x_2^{(N)})\}$
  - This can be easily fixed.

Supervised Image-to-image Translation

- Supervisedly relating $x = F(x_1^{(i)})$ to $x_2^{(i)}$
- Ledig et al, CVPR'17: Adding content loss

$$||x - x_2^{(i)}||_2 + ||\mathrm{VGG}(x) - \mathrm{VGG}(x_2^{(i)})||_2$$

SRGAN

C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi “Photo-realistic image superresolution using a generative adversarial networks “, CVPR 2017

Supervised Image-to-image Translation

- Supervisedly relating $x = F(x_1^{(i)})$ to $x_2^{(i)}$
- Isola et al, CVPR'17: Learning a joint distribution.

$$\max_F E_{p_{\mathcal{X}_1}} [\log(D(x_1, F(x_1)))]$$

Pix2Pix

P. Isola, J. Zhu, T. Zhou, A. Efros “Image-to-image translation with conditional generative networks“, CVPR 2017

Unsupervised Image-to-image Translation

- Corresponding images could be expensive to get.
- In the unsupervised setting
  - $\text{Dataset}_1 = \{x_1^{(n_1)}, x_1^{(n_2)}, ..., x_1^{(n_N)}\}$
  - $\text{Dataset}_2 = \{x_2^{(m_1)}, x_2^{(m_2)}, ..., x_2^{(m_M)}\}$
- With no correspondences, we need additional constraints/assumptions for learning image-to-image translation.

SimGAN

- Srivastava et al. CVPR'17: adding cross-domain content loss.

$$\max_F E_{p_{\mathcal{X}_1}} [\log D(F(x_1)) - \lambda ||F(x_1) - x_1||_1]$$

Cycle Constraint

- Learning two-way translation.
- DiscoGAN by Kim et al. ICML'17 (arXiv 1703.05192)
- CycleGAN by Zhu et al. arXiv 1703.10593

$$\max_{F_{12},F_{21}} E_{p_{\mathcal{X}_1}}[\log(D_2(F_{12}(x_1)) - \lambda||F_{21}(F_{12}(x_1)) - x_1||_p^p] +$$
$$E_{p_{\mathcal{X}_2}}[\log(D_1(F_{21}(x_2)) - \lambda||F_{12}(F_{21}(x_2)) - x_2||_p^p]$$

CycleGAN unsupervised image to image translation results

Shared Latent Space Constraint

- CoGAN by Liu et al. NIPS'16
- Assume corresponding images in different domains share the same representation in a hidden space.
- $G_1 = G_{L1} \circ G_H$ and $G_2 = G_{L2} \circ G_H$
- $$\max_{G_H, G_{L1}, G_{L2}} E_{p_{\mathcal{N}}}[\log D_1(G_{L1}(G_H(z)) + \log D_2(G_{L2}(G_H(z))]$$

Shared Latent Space Constraint

- Under sufficient capacity and descending to local minimum assumptions as in Goodfellow et al., CoGAN learns a joint distribution of multi-domain images.
- Image translation

$$G_2(\text{argmin}_z ||G_1(z) - x_1||_p^p)$$

The CoVAE-GAN Framework

Ming-Yu Liu, Thomas Breuel, Jan Kautz "Unsupervised Image-to-image translation networks", arXiv 2017

- Augmenting CoGAN with VAEs for mapping images to the shared latent space.
- Applying weight-sharing constraints to encoders E_1 and E_2
- Image reconstruction streams and image translation streams

The CoVAE-GAN Framework

$$
\begin{aligned}
&E_{p_{\mathcal{X}_1}}[\log D_1(G_1(E_1(x_1))) + \log D_2(G_2(E_1(x_1)))- \\
&\lambda_1 ||G_1(E_1(x_1)) - x_1||_p^p - \lambda_2 KL(E_1(x_1)||p_{\mathcal{N}})]+ \\
&E_{p_{\mathcal{X}_2}}[\log D_1(G_1(E_2(x_2))) + \log D_2(G_2(E_2(x_2)))- \\
&\lambda_1 ||G_2(E_2(x_2)) - x_2||_p^p - \lambda_2 KL(E_2(x_2)||p_{\mathcal{N}})]
\end{aligned}
$$

Intuition

In the ideal case (supervised setting)

We can enforce that corresponding images are mapping to the same point in the embedding space.

$$E_1(x_1^{(i)}) \equiv E_2(x_2^{(i)})$$

Intuition

In the unsupervised setting

Although we can use a GAN discriminator to make sure the images coming out from G_2 resembling real images, there is no guarantee that the output image from G_1 and G_2 form a corresponding pair.

Intuition

In the unsupervised setting

However, if we apply shared latent space assumption via weight-sharing, then the generated images from G_1 and G_2 will resemble a pair of corresponding images.

In the unsupervised setting

Similarly, we can enforce weight-sharing to E1 and E2 to encourage a pair of corresponding images to have the same embedding.

Toy Examples

Domain 1 Domain 2

Image Translation Errors

Subnetworks included	$\mathcal{X}_1 \rightarrow \mathcal{X}_2$	$\mathcal{X}_2 \rightarrow \mathcal{X}_1$
$E_1, E_2, G_1, G_2, D_1, D_2$ (UNIT)	**59.3**	**47.0**
E_1, E_2, G_1, G_2 (Coupled VAEs)	66.3	59.0
E_1, G_1, G_2, D_1, D_2	292.0	-
E_2, G_1, G_2, D_1, D_2	-	297.8
E_1, E_2, G_1, D_1	-	186.6
E_1, E_2, G_2, D_2	176.5	-

Image Translation Results

Results on Unsupervised Thermal-Image-to-RGB-Image Translation. Left: input thermal image. Right: Output color image.

Results on Unsupervised RGB-Image-to-Thermal-Image Translation. Left: input color image. Right: Output thermal image.

Results on Unsupervised Day-Image-to-Night-Image Translation. Left: input day image. Right: Output night image.

Results on Unsupervised Night-Image-to-Day-Image Translation. Left: input night image. Right: Output day image.

Image Translation Results

Results on Unsupervised Sunny-Image-to-Rainy-Image Translation. Left: input sunny image. Right: Output rainy image.

Results on Unsupervised Rainy-Image-to-Sunny-Image Translation. Left: input rainy image. Right: Output sunny image.

Back View Front View Left View Right View

Foggy image to clear sky image

Per-frame video translation results

Attribute-based Face Image Translation

Image translation results

Input **+Blondhair**

Input **+Eyeglasses**

Input **-Goatee**

Input **-Smiling**

Image translation results

Input **+Blondhair**

Input **+Eyeglasses**

Input **+Goatee**

Input **+Smiling**

Conclusions

- Family of generative models
- Theory of GANs
- Various training techniques
- Joint image distribution and video distribution
- Image translation applications

THANK YOU