生成对抗网络基本原理

讲师：加号

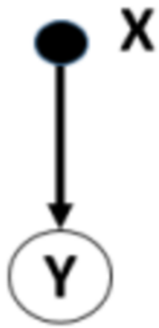
@翻滚吧_加号

本篇课件会用到GAN之父Ian Goodfellow本人2016年NIPS上的slides
(Goodfellow, 2016)

Generative Adversarial Networks (GANs)

Ian Goodfellow, OpenAI Research Scientist
NIPS 2016 tutorial
Barcelona, 2016-12-4

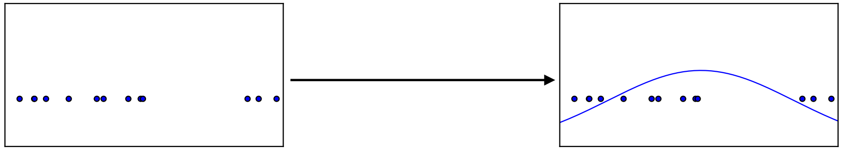
什么是 生成模型与判别模型



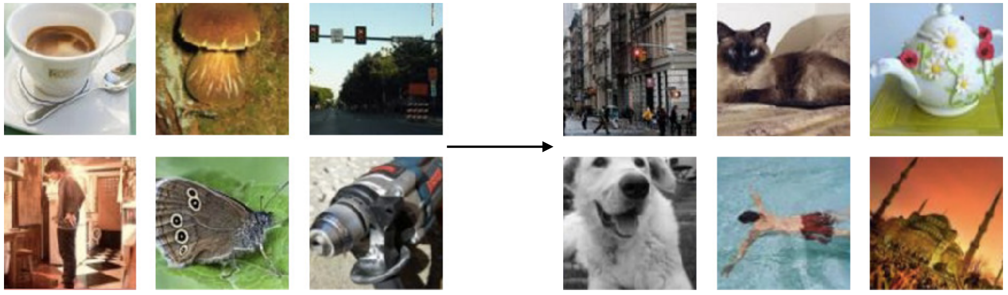
生成模型

Generative Modeling

- Density estimation



- Sample generation



Training examples　　Model samples

(Goodfellow 2016)

为什么要学习生成模型?

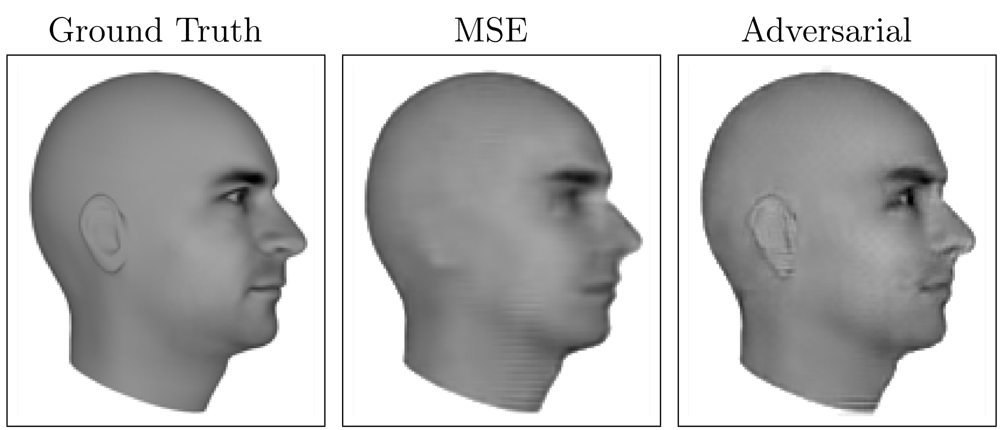
Why study generative models?

- Excellent test of our ability to use high-dimensional, complicated probability distributions
- Simulate possible futures for planning or simulated RL
- Missing data
    - Semi-supervised learning
- Multi-modal outputs
- Realistic generation tasks

(Goodfellow 2016)

为什么要学习生成模型?

Next Video Frame Prediction

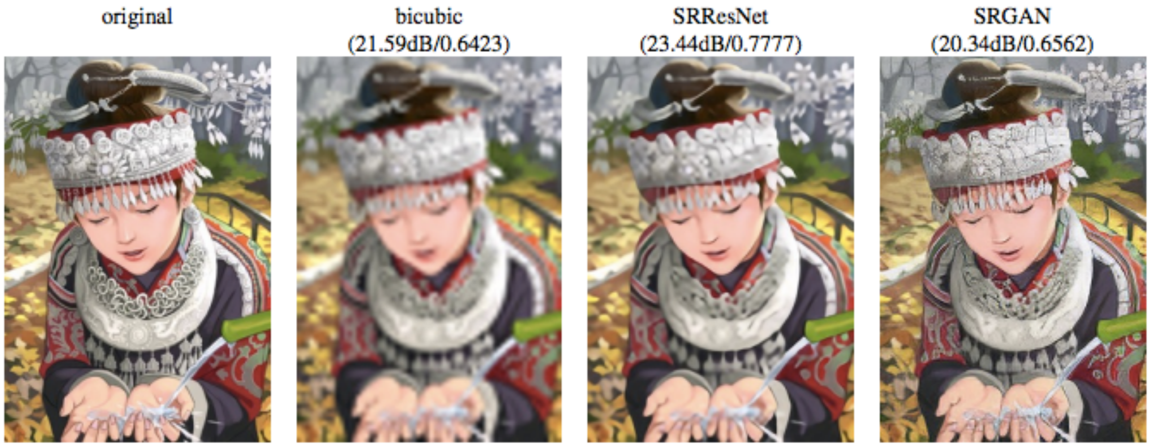


(Lotter et al 2016)

(Goodfellow 2016)

为什么要学习生成模型?

Single Image Super-Resolution

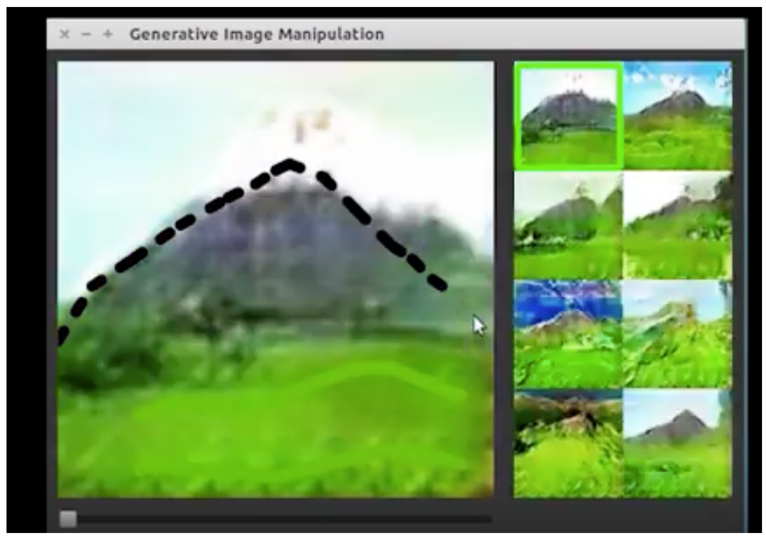


(Ledig et al 2016)

(Goodfellow 2016)

为什么要学习生成模型?

iGAN



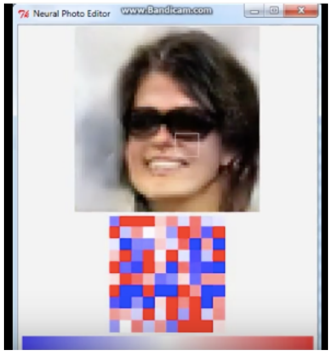
youtube

(Zhu et al 2016)

(Goodfellow 2016)

为什么要学习生成模型?

Introspective Adversarial Networks



youtube

(Brock et al 2016)

(Goodfellow 2016)

为什么要学习生成模型?

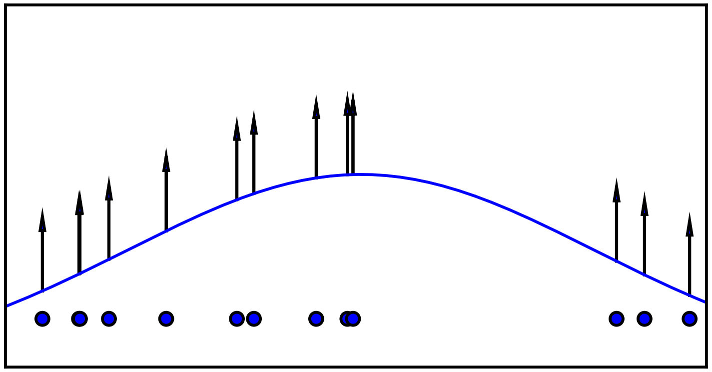
Image to Image Translation

(Isola et al 2016)

(Goodfellow 2016)

生成模型原理

Maximum Likelihood

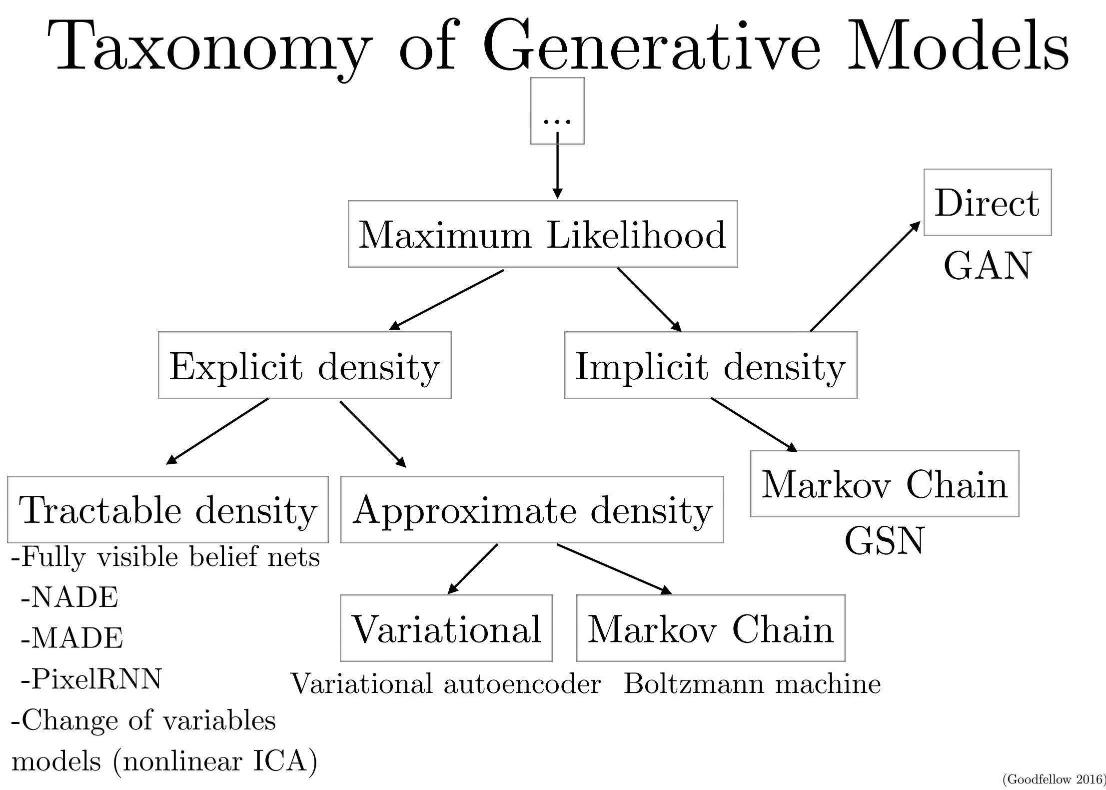


$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{x\sim p_{\text{data}}} \log p_{\text{model}}(\boldsymbol{x} \mid \boldsymbol{\theta})$$

(Goodfellow 2016)

最大似然估计

生成模型大家族


Taxonomy of Generative Models
...
Maximum Likelihood
Explicit density
Implicit density
Direct
GAN
Tractable density
-Fully visible belief nets
-NADE
-MADE
-PixelRNN
-Change of variables
models (nonlinear ICA)
Approximate density
Variational
Variational autoencoder
Markov Chain
Boltzmann machine
Markov Chain
GSN
(Goodfellow 2016)

生成模型大家族

Fully Visible Belief Nets

- Explicit formula based on chain rule: (Frey et al, 1996)

$$p_{\text{model}}(\boldsymbol{x}) = p_{\text{model}}(x_1) \prod_{i=2}^{n} p_{\text{model}}(x_i \mid x_1, \ldots, x_{i-1})$$

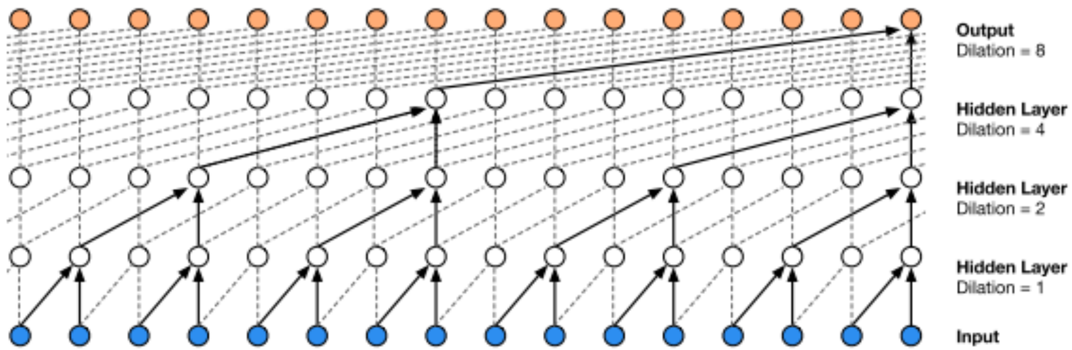
- Disadvantages:
  - O(n) sample generation cost
  - Generation not controlled by a latent code

PixelCNN elephants
(van den Ord et al 2016)

(Goodfellow 2016)

生成模型大家族

WaveNet



Amazing quality
Sample generation slow

Two minutes to synthesize
one second of audio

(Goodfellow 2016)

生成模型大家族

Change of Variables

$$y = g(x) \Rightarrow p_x(\boldsymbol{x}) = p_y(g(\boldsymbol{x})) \left| \det \left(\frac{\partial g(\boldsymbol{x})}{\partial \boldsymbol{x}} \right) \right|$$

e.g. Nonlinear ICA (Hyvärinen 1999)

64x64 ImageNet Samples
Real NVP (Dinh et al 2016)

Disadvantages:

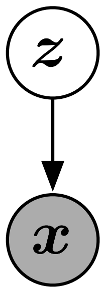
- Transformation must be invertible
- Latent dimension must match visible dimension

(Goodfellow 2016)

生成模型大家族

Variational Autoencoder

(Kingma and Welling 2013, Rezende et al 2014)



$$\log p(\boldsymbol{x}) \geq \log p(\boldsymbol{x}) - D_{\mathrm{KL}}\left(q(\boldsymbol{z}) \| p(\boldsymbol{z} \mid \boldsymbol{x})\right)$$
$$= \mathbb{E}_{\boldsymbol{z} \sim q} \log p(\boldsymbol{x}, \boldsymbol{z}) + H(q)$$

CIFAR-10 samples
(Kingma et al 2016)

Disadvantages:
-Not asymptotically consistent unless q is perfect
-Samples tend to have lower quality

(Goodfellow 2016)

生成模型大家族

Boltzmann Machines

$$p(\boldsymbol{x}) = \frac{1}{Z} \exp\left(-E(\boldsymbol{x}, \boldsymbol{z})\right)$$

$$Z = \sum_{\boldsymbol{x}} \sum_{\boldsymbol{z}} \exp\left(-E(\boldsymbol{x}, \boldsymbol{z})\right)$$

- Partition function is intractable
- May be estimated with Markov chain methods
- Generating samples requires Markov chains too

(Goodfellow 2016)

生成对抗模型

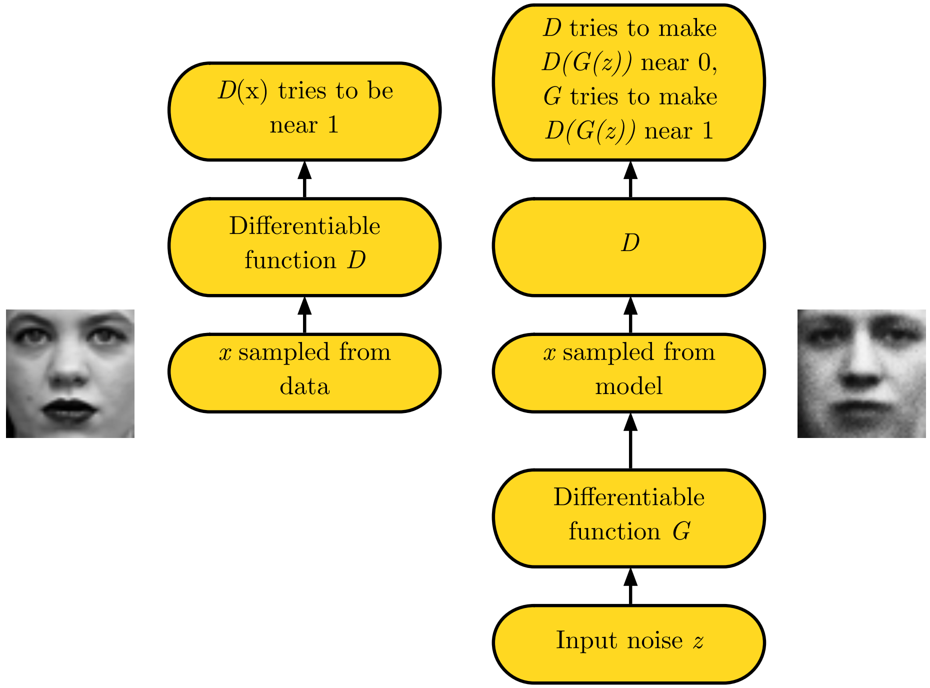
GANs

- Use a latent code
- Asymptotically consistent (unlike variational methods)
- No Markov chains needed
- Often regarded as producing the best samples
    - No good way to quantify this

(Goodfellow 2016)

生成对抗模型

Adversarial Nets Framework



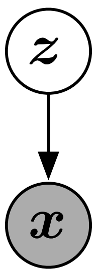
(Goodfellow 2016)

生成对抗模型

生成对抗模型

Generator Network

$$\boldsymbol{x} = G(\boldsymbol{z}; \boldsymbol{\theta}^{(G)})$$



-Must be differentiable
- No invertibility requirement
- Trainable for any size of z
- Some guarantees require z to have higher dimension than x
- Can make x conditionally Gaussian given z but need not do so

(Goodfellow 2016)

生成对抗模型

Training Procedure

- Use SGD-like algorithm of choice (Adam) on two minibatches simultaneously:
  - A minibatch of training examples
  - A minibatch of generated samples
- Optional: run k steps of one player for every step of the other player.

(Goodfellow 2016)

生成对抗模型

Minimax Game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -J^{(D)}$$

-Equilibrium is a saddle point of the discriminator loss
-Resembles Jensen-Shannon divergence
-Generator minimizes the log-probability of the discriminator being correct

(Goodfellow 2016)

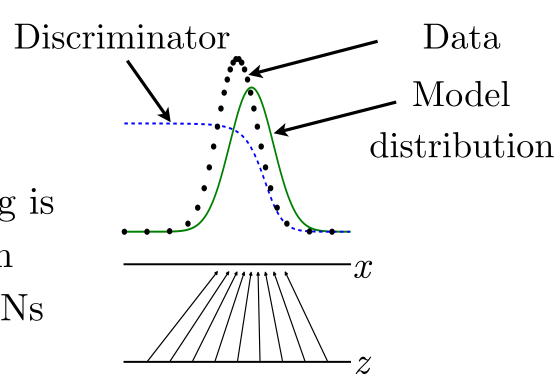
生成对抗模型

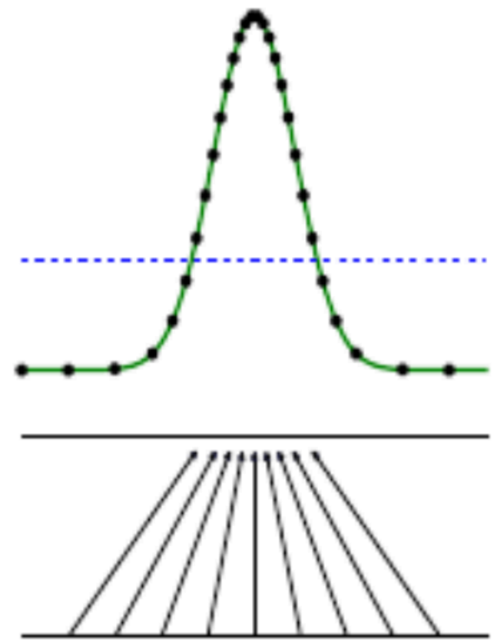
Discriminator Strategy

Optimal $D(\boldsymbol{x})$ for any $p_{\text{data}}(\boldsymbol{x})$ and $p_{\text{model}}(\boldsymbol{x})$ is always

$$D(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_{\text{model}}(x)}$$

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





(Goodfellow 2016)

生成对抗模型

Non-Saturating Game

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log D\left(G(\boldsymbol{z})\right)$$

-Equilibrium no longer describable with a single loss

-Generator maximizes the log-probability of the discriminator being mistaken

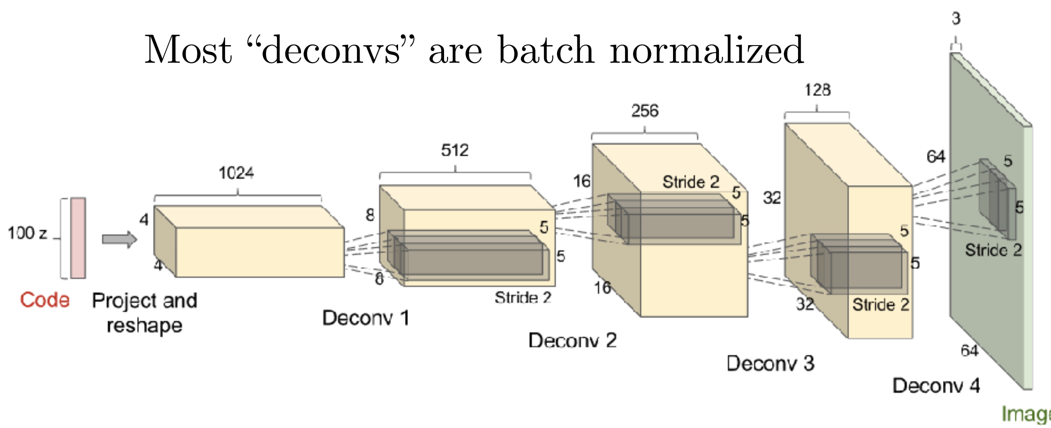
-Heuristically motivated; generator can still learn even when discriminator successfully rejects all generator samples

(Goodfellow 2016)

DCGAN

DCGAN Architecture

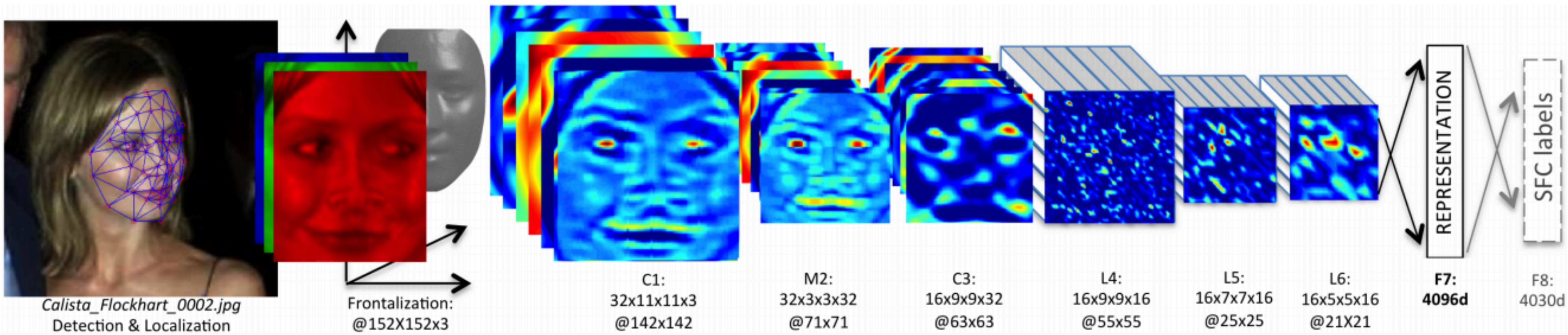
Most "deconvs" are batch normalized



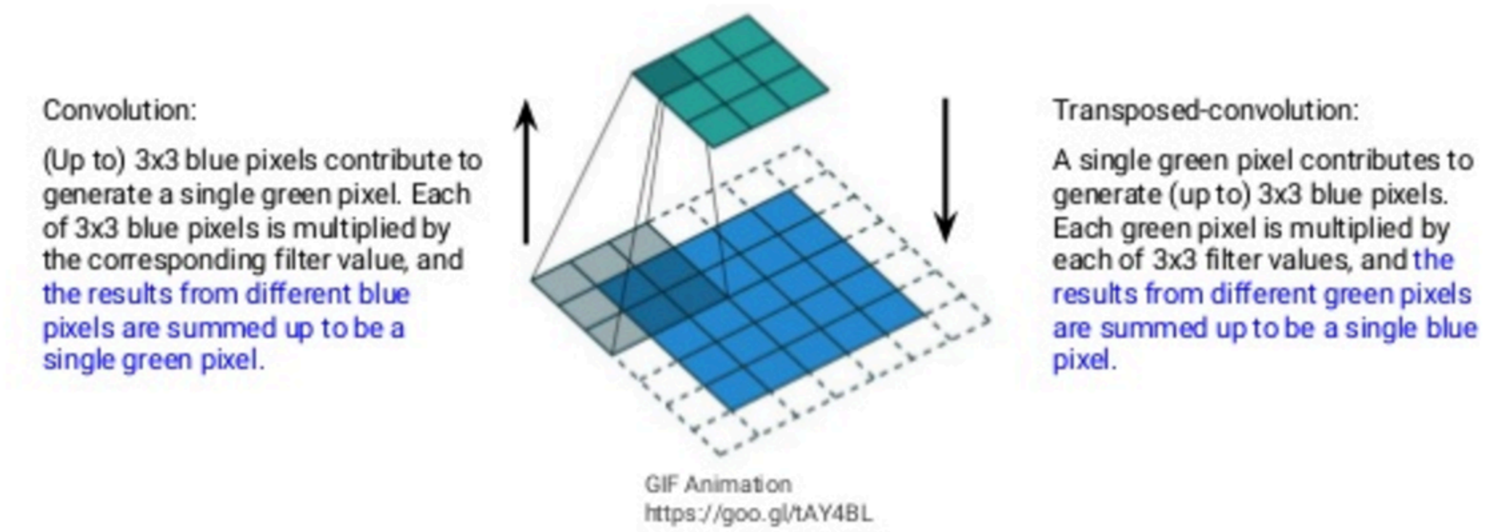
(Radford et al 2015)

(Goodfellow 2016)

DCGAN


Calista_Flockhart_0002.jpg
Detection & Localization
Frontalization:
@152X152x3
C1:
32x11x11x3
@142x142
M2:
32x3x3x32
@71x71
C3:
16x9x9x32
@63x63
L4:
16x9x9x16
@55x55
L5:
16x7x7x16
@25x25
L6:
16x5x5x16
@21X21
REPRESENTATION
F7:
4096d
SFC labels
F8:
4030d

DCGAN


Convolution:
(Up to) 3x3 blue pixels contribute to generate a single green pixel. Each of 3x3 blue pixels is multiplied by the corresponding filter value, and the results from different blue pixels are summed up to be a single green pixel.
Transposed-convolution:
A single green pixel contributes to generate (up to) 3x3 blue pixels. Each green pixel is multiplied by each of 3x3 filter values, and the results from different green pixels are summed up to be a single blue pixel.
GIF Animation
https://goo.gl/tAY4BL

DCGAN

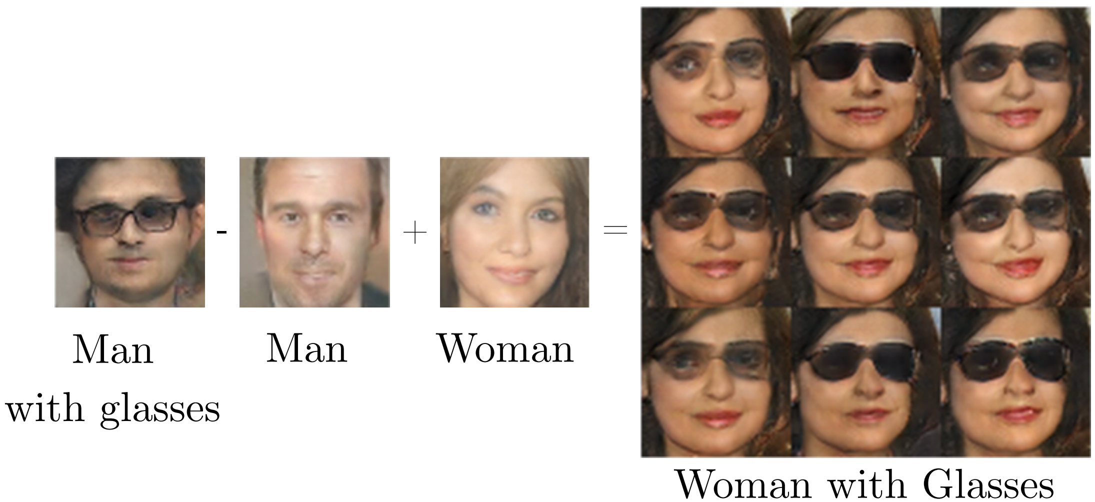
DCGANs for LSUN Bedrooms

(Radford et al 2015)

(Goodfellow 2016)

DCGAN

Vector Space Arithmetic



(Radford et al, 2015)

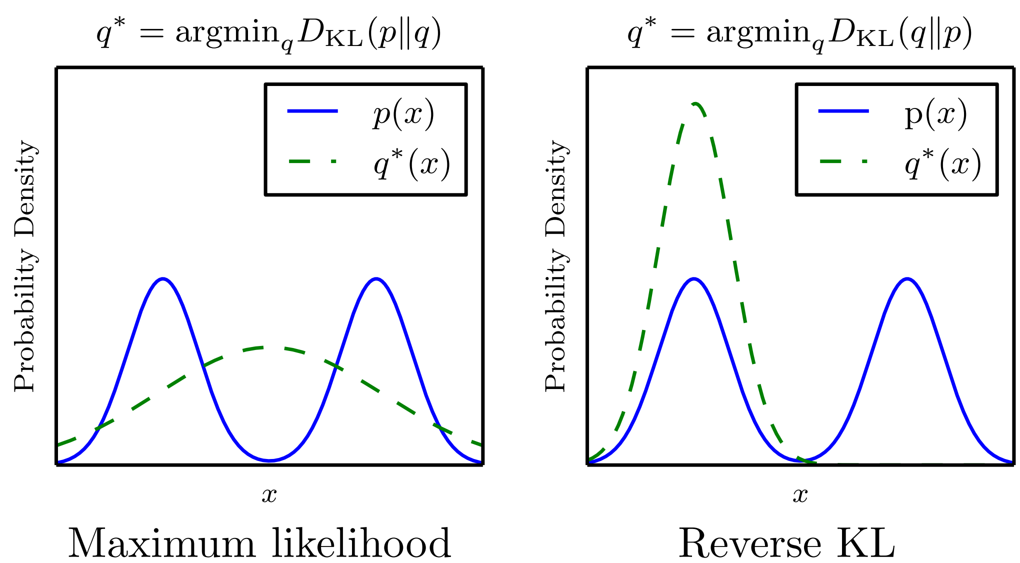
(Goodfellow 2016)

KL Divergence

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} \, \mathrm{d}x,$$

KL Divergence

Is the divergence important?



(Goodfellow et al 2016)

(Goodfellow 2016)

KL Divergence

用最大似然法写G的目标函数

Modifying GANs to do Maximum Likelihood

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$

$$J^{(G)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \exp\left(\sigma^{-1}\left(D\left(G(\boldsymbol{z})\right)\right)\right)$$

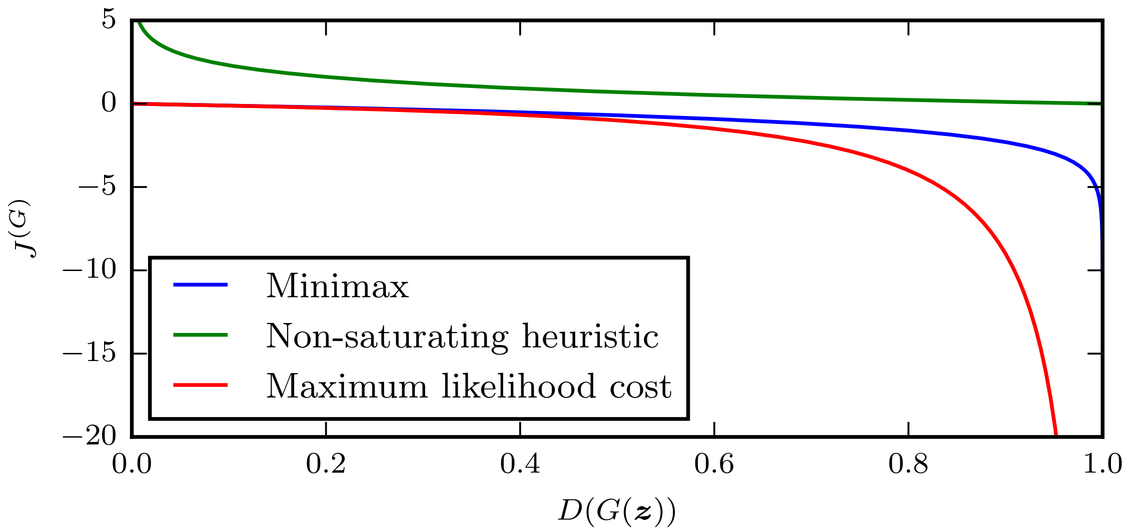
When discriminator is optimal, the generator gradient matches that of maximum likelihood

("On Distinguishability Criteria for Estimating Generative Models", Goodfellow 2014, pg 5)

(Goodfellow 2016)

以上，

Comparison of Generator Losses



(Goodfellow 2014)

(Goodfellow 2016)

代码走一波

【见随堂iPython Notebook】

Thank you!