Deep Learning

Representative Deep Learning Methods

Xinfeng Zhang（张新峰）
School of Computer Science and Technology
University of Chinese Academy of Sciences
Email: xfzhang@ucas.ac.cn

提纲

- 生成对抗网络
- 胶囊网络
- 注意力机制
- 记忆网络
- 深度强化学习
- 深度森林
- 中英文术语对照

1 生成对抗网络

GAN产生背景

❑ 机器学习方法

– 生成方法，所学到到模型称为生成式模型

- 生成方法通过观测数据学习样本与标签的联合概率分布P(X, Y)，训练好的模型，即生成模型，能够生成符合样本分布的新数据
- 生成式模型在无监督深度学习方面占据主要位置，可以用于在没有目标类标签信息的情况下捕捉观测到或可见数据的高阶相关性

– 判别方法，所学到的模型称为判别式模型

- 判别方法由数据直接学习决策函数f(X)或者条件概率分布P(Y|X)作为预测的模型，即判别模型
- 判别模型经常用在有监督学习方面
- 判别方法关心的是对给定的输入X，应该预测什么样的输出Y

GAN产生背景

❑ 困难和前景

- 有监督学习经常比无监督学习获得更好的模型，但是有监督学习需要大量的标注数据，从长远看无监督学习更有发展前景
- 支持无监督学习的生成式模型遇到两大困难
  - 首先是人们需要大量的先验知识去对真实世界进行建模，而建模的好坏直接影响着我们的生成模型的表现
  - 真实世界的数据往往很复杂，人们要用来拟合模型的计算量往往非常庞大，甚至难以承受

生成对抗网络的提出

- **2014年，生成对抗网络（Generative Adversarial Networks，GAN）由当时还在蒙特利尔读博士的Ian Goodfellow（导师Bengio）提出**
  - 2016年，GAN热潮席卷AI领域顶级会议，从ICLR到NIPS，大量高质量论文被发表和探讨
  - 2017年入选MIT评论35岁以下创新人物

GAN基本原理

- **GAN起源于博弈论中的二人零和博弈（获胜1，失败-1）**
  - 由两个互为敌手的模型组成
    - 生成模型（假币制造者团队）
    - 判别模型（警察团队）
  - 竞争使得两个团队不断改进他们的方法直到无法区分假币与真币

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, Yoshua Bengio. Generative Adversarial Nets. NIPS 2014: 2672-2680

评价

Yann LeCun
(Facebook AI Research)

“我们一直在错过一个关键因素就是**无监督/预测学习**，这是指：机器给真实环境**建模**、**预测**可能的未来、并通过观察和演示来**理解世界是如何运行**的能力”

“GAN为创建无监督学习提供了强有力的算法框架，有望帮助我们为**AI加入常识**，我们认为，沿着这条路走下去，有不小的成功的机会能开发出更智慧的AI”

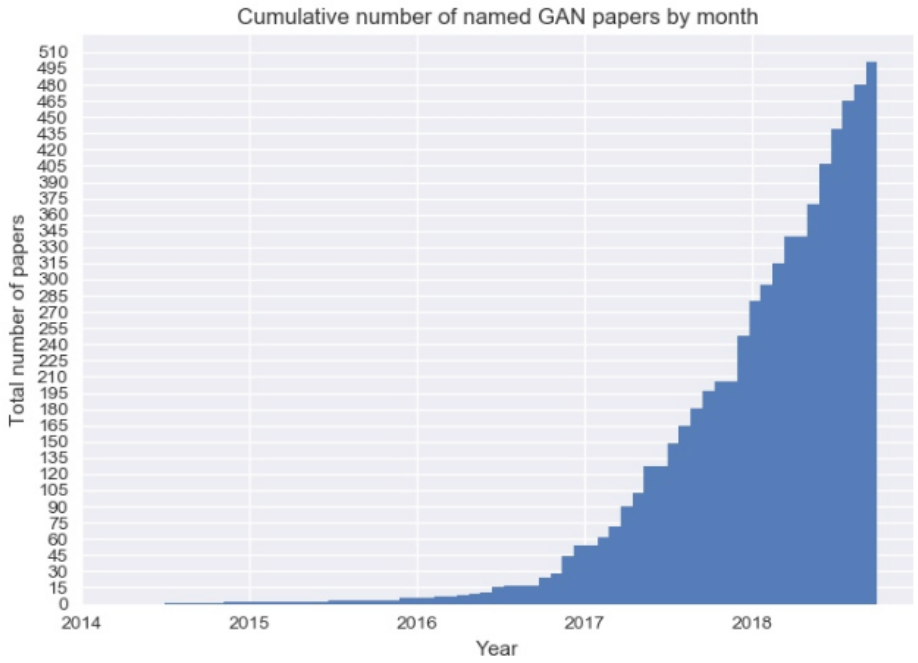
GAN的发展

❑ The GAN Zoo

- https://github.com/hindupuravinash/the-gan-zoo
- ABC-GAN, AC-GAN, …
- BAGAN, BCGAN…
- C-GAN, CA-GAN,…
- …α-GAN, β-GAN,…

Check out Deep Hunt - my weekly AI newsletter for this repo as blogpost and follow me on Twitter.

- 3D-ED-GAN - Shape Inpainting using 3D Generative Adversarial Network and Recurrent Convolutional Networks
- 3D-GAN - Learning a Probabilistic Latent Space of Object Shapes via 3D Generative-Adversarial Modeling (github)
- 3D-IWGAN - Improved Adversarial Systems for 3D Object Generation and Reconstruction (github)
- 3D-PhysNet - 3D-PhysNet: Learning the Intuitive Physics of Non-Rigid Object Deformations
- 3D-RecGAN - 3D Object Reconstruction from a Single Depth View with Adversarial Learning (github)
- ABC-GAN - ABC-GAN: Adaptive Blur and Control for improved training stability of Generative Adversarial Networks (github)
- ABC-GAN - GANs for LIFE: Generative Adversarial Networks for Likelihood Free Inference
- AC-GAN - Conditional Image Synthesis With Auxiliary Classifier GANs
- acGAN - Face Aging With Conditional Generative Adversarial Networks
- ACGAN - Coverless Information Hiding Based on Generative adversarial networks
- acGAN - On-line Adaptative Curriculum Learning for GANs
- ACtuAL - ACtuAL: Actor-Critic Under Adversarial Learning
- AdaGAN - AdaGAN: Boosting Generative Models
- Adaptive GAN - Customizing an Adversarial Example Generator with Class-Conditional GANs
- AdvEntuRe - AdvEntuRe: Adversarial Training for Textual Entailment with Knowledge-Guided Examples
- AdvGAN - Generating adversarial examples with adversarial networks
- AE-GAN - AE-GAN: adversarial eliminating with GAN
- AE-OT - Latent Space Optimal Transport for Generative Models
- AEGAN - Learning Inverse Mapping by Autoencoder based Generative Adversarial Nets
- AF-DCGAN - AF-DCGAN: Amplitude Feature Deep Convolutional GAN for Fingerprint Construction in Indoor Localization System
- AffGAN - Amortised MAP Inference for Image Super-resolution
- AIM - Generating Informative and Diverse Conversational Responses via Adversarial Information Maximization
- AL-CGAN - Learning to Generate Images of Outdoor Scenes from Attributes and Semantic Layouts

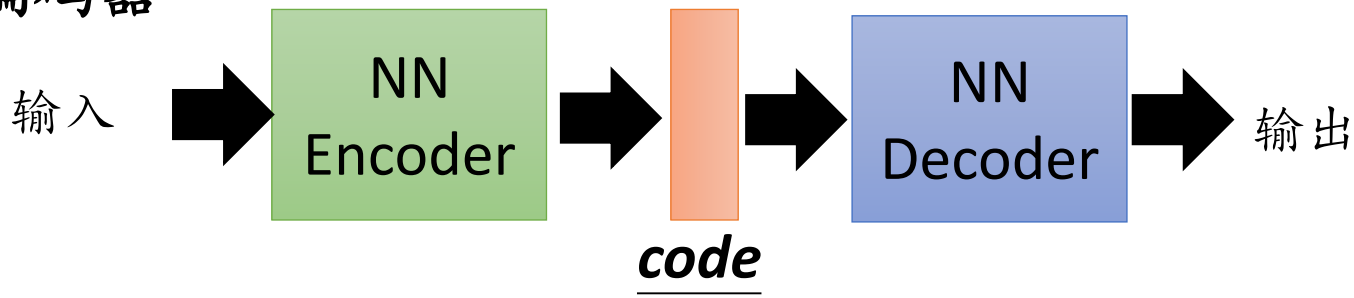


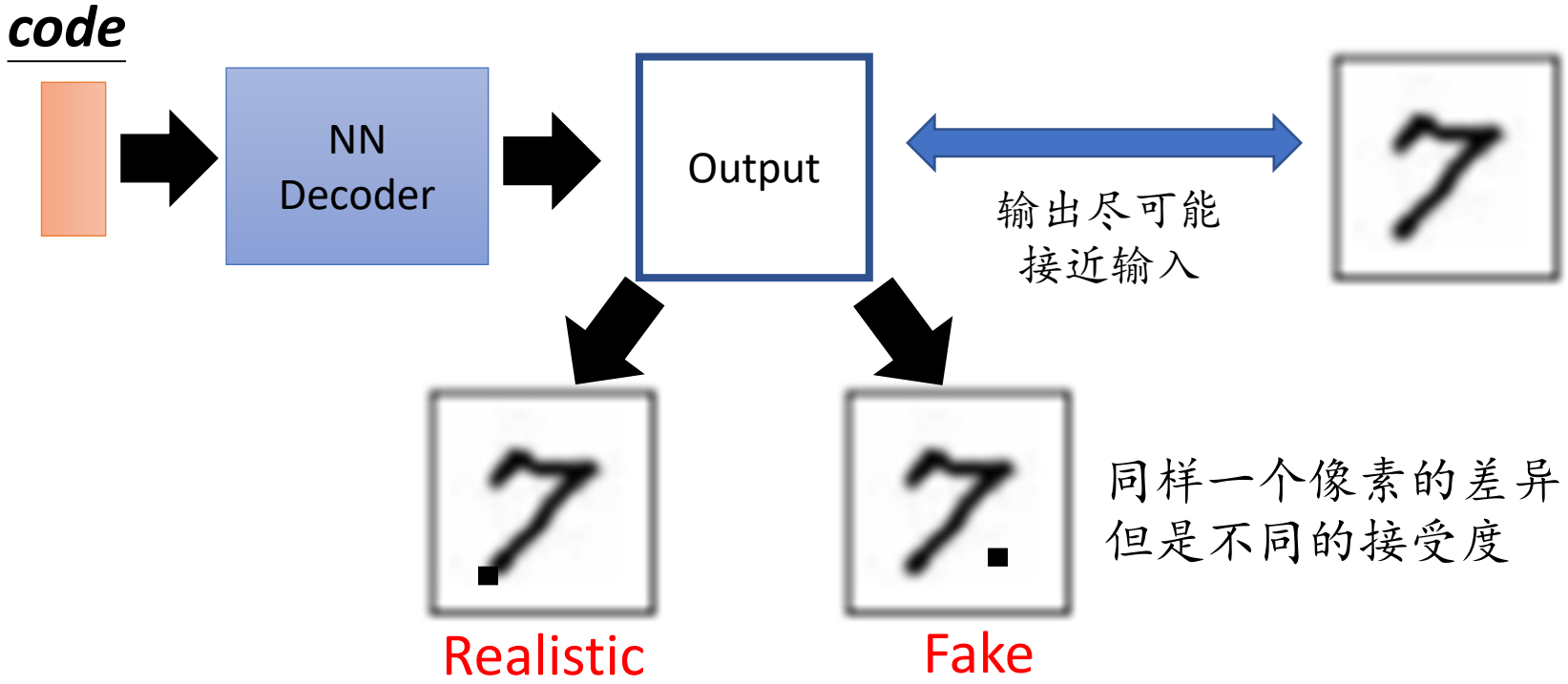
GAN的发展

- **NIPS 2016 Tutorial: Generative Adversarial Networks**
  - Author: Ian Goodfellow
  - Paper: https://arxiv.org/abs/1701.00160
  - Video: https://channel9.msdn.com/Events/Neural-Information-Processing-Systems-Conference/Neural-Information-Processing-Systems-Conference-NIPS-2016/Generative-Adversarial-Networks

生成模型中的问题

❑ 自编码器





生成模型中的问题

❑ 概率模型

- 按照某种概率分布生成数据，得到最能覆盖训练样本的概率分布
- 需要确定样本的概率模型，即显式地定义概率密度函数
  - 例如：深度玻尔兹曼机采用对比散度(Gibbs采样)，样本可能会具有很强的相关性，尤其是在高维的情况下

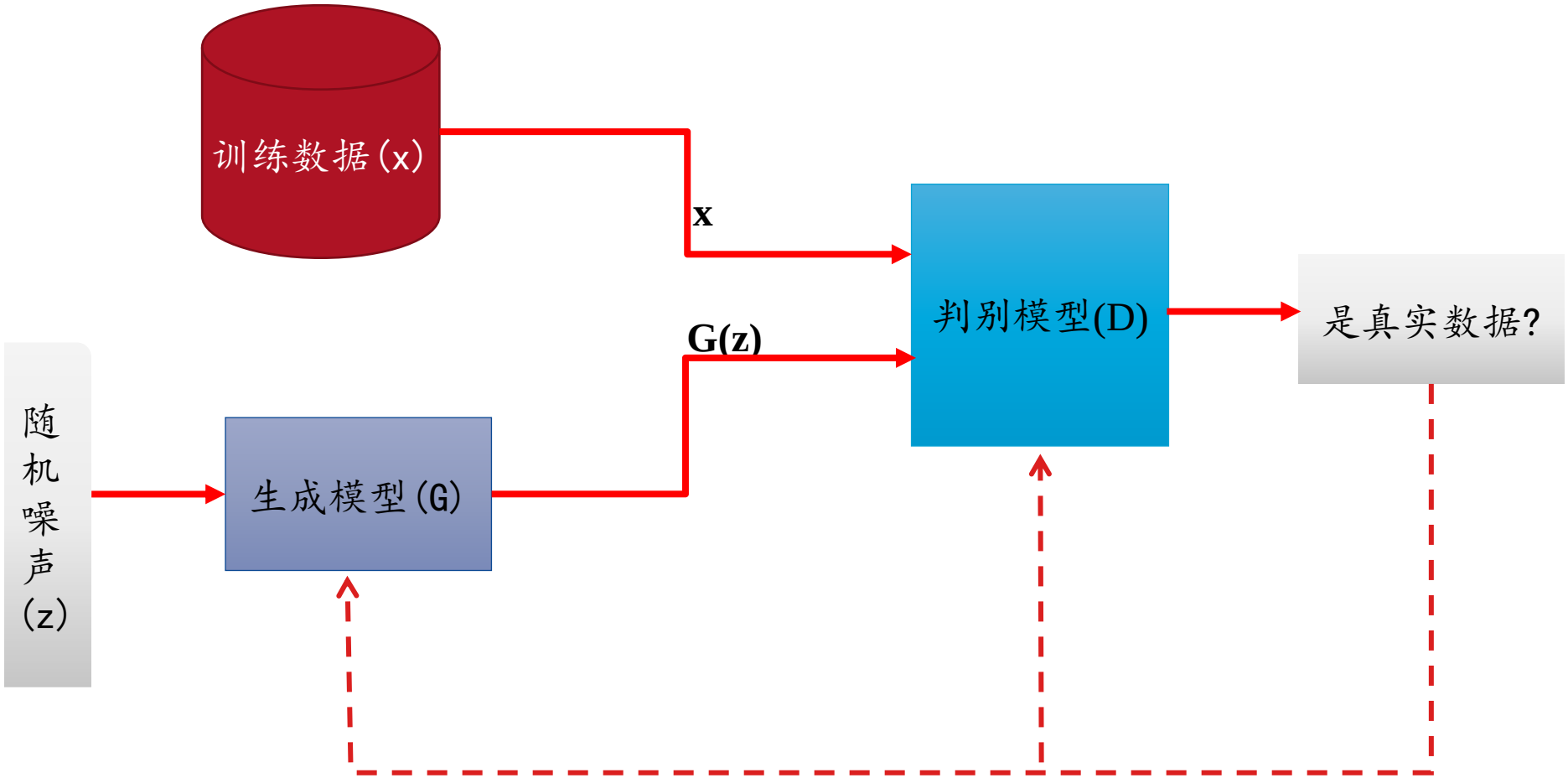
$$\frac{\partial \ln\left(P(\mathbf{v}^i)\right)}{\partial w_{ij}} = P(h_j = 1|\mathbf{v}^i)v_i^i - \sum_{\mathbf{v}} P(\mathbf{v})P(h_j = 1|\mathbf{v})\, v_i$$

$$\frac{\partial \ln\left(P(\mathbf{v}^i)\right)}{\partial a_i} = v_i - \sum_{\mathbf{v}} P(\mathbf{v})\, v_i$$

$$\frac{\partial \ln\left(P(\mathbf{v}^i)\right)}{\partial b_i} = P(h_i = 1|\mathbf{v}^i) + \sum_{\mathbf{v}} P(\mathbf{v})P(h_i = 1|\mathbf{v})$$

GAN模型

- 估计样本概率分布却不需要显示定义概率分布



GAN模型

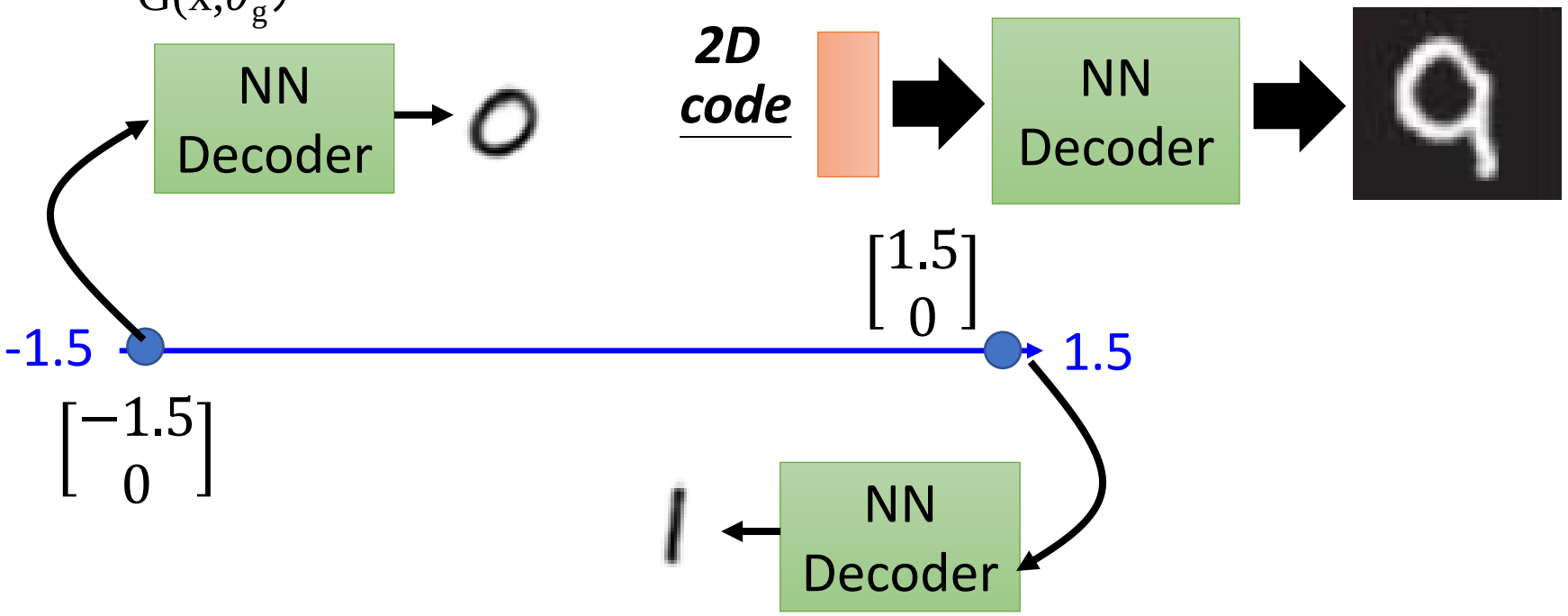
❑ 生成模型

- 捕捉样本数据的分布，用服从某一分布（均匀分布，高斯分布）的噪声z生成一个类似真实训练数据的样本，追求效果是越像真实的数据越好
- $p_{data}(x)$表示真实数据的x分布
- $p_z(z)$表示输入噪声变量z的分布
- p_g 表示在数据x上学习得到的生成样本的分布
- $G(z;\theta_g)$表示生成模型（多层感知器）

GAN模型

❑ 生成模型(示例）

- 可以看做一种映射函数
- 当数据集是图片的时候，那么我们输入的随机噪声其实就是相当于低维的数据，经过生成模型G的映射就变成了一张生成的图片 $G(x,\theta_g)$

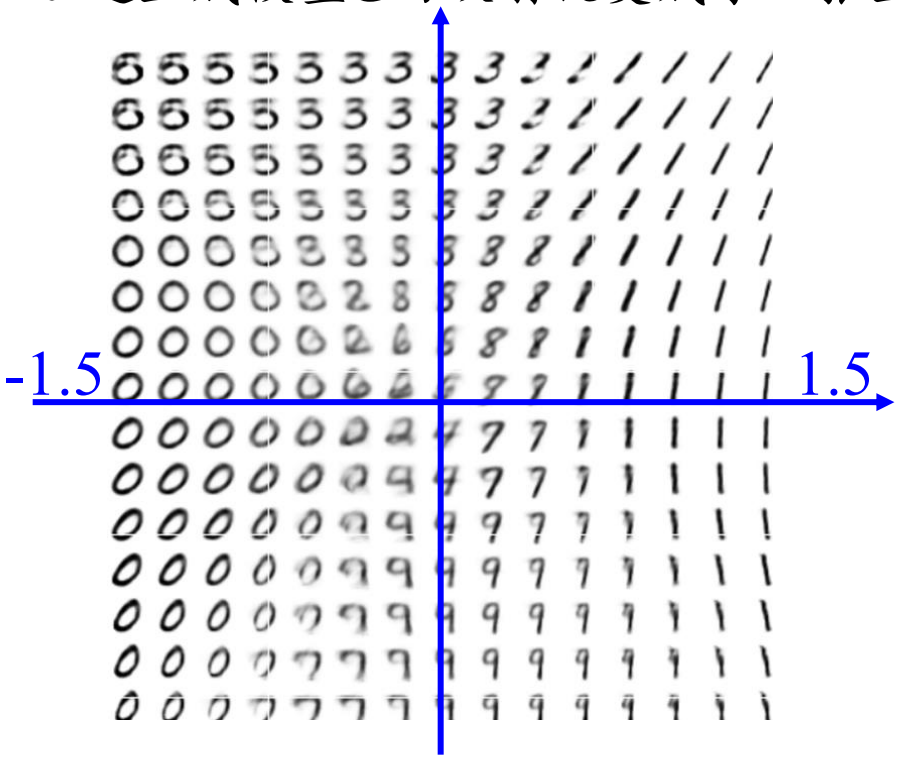


GAN模型

❑ 生成模型(示例）

- 可以看做一种映射函数
- 当数据集是图片的时候，那么我们输入的随机噪声其实就是相当于低维的数据，经过生成模型G的映射就变成了一张生成的图片

$G(x,\theta_g)$



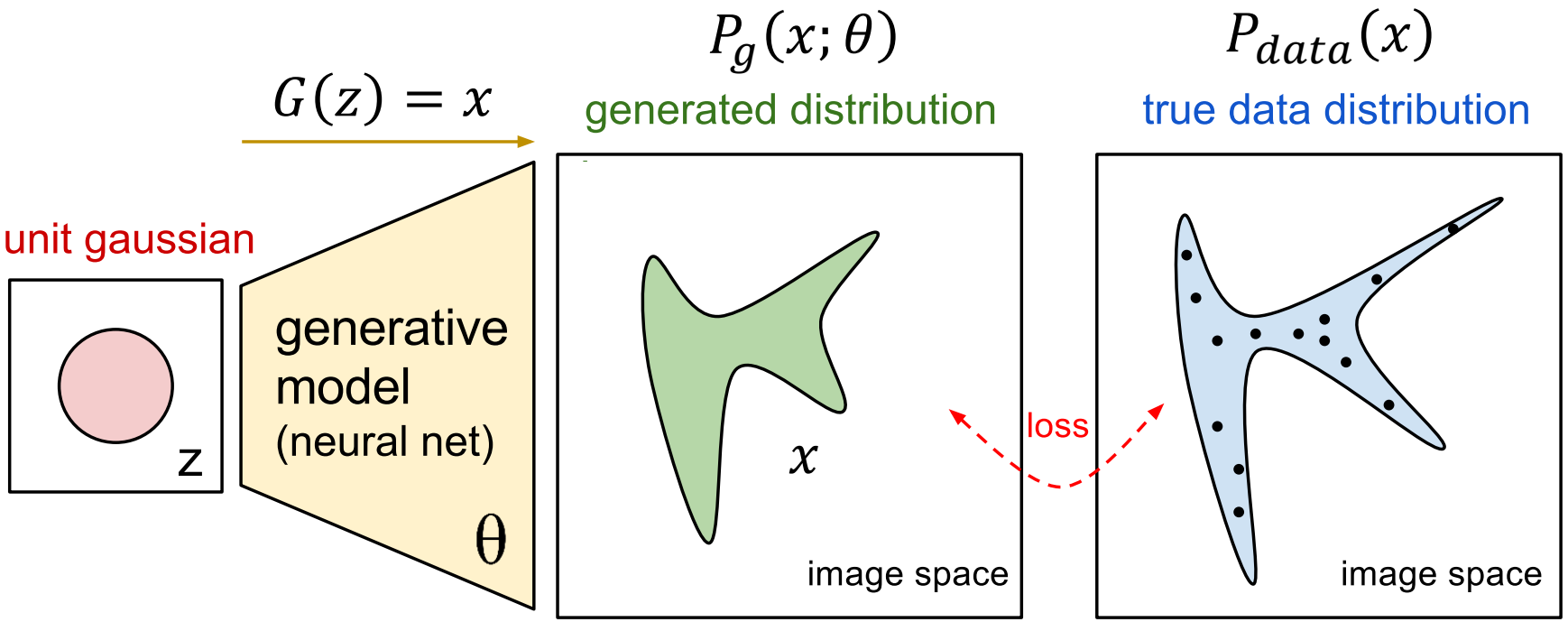
GAN模型

❑ 判别模型

- 一个二分类器，估计一个样本来自训练数据（而非生成数据）的概率，如果样本来自真实的训练数据，输出大概率，否则，输出小概率。
- $D(x;\theta_d)$表示判别模型（多层感知器）
- $D(x)$表示x来自真实数据而非生成数据的概率

GAN模型

❑ 用网络代替概率模型$P_g(x;\theta)$， $P_{data}(x)$



GAN目标函数

❑ GAN目标函数

$$\min_G \max_D V(G,D) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[\log\Big(1 - D\big(G(z)\big)\Big)]$$

– 训练GAN的时候，判别模型希望目标函数最大化，也就是使判别模型判断真实样本为“真”，判断生成样本为“假”的概率最大化，要尽量最大化自己的判别准确率

– 判别模型也可以写成损失函数的形式

$$L(G,D) = -E_{x\sim p_{data}(x)}[logD(x)] - E_{z\sim p_z(z)}[\log\Big(1 - D\big(G(z)\big)\Big)]$$

GAN损失函数

- 与之相反，生成模型希望该目标函数最小化，也就是降低判别模型对数据来源判断正确的概率，要最小化判别模型的判别准确率
- 生成模型训练目标

$$V(G,D) = E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[\log\Big(1 - D\big(G(z)\big)\Big)]$$

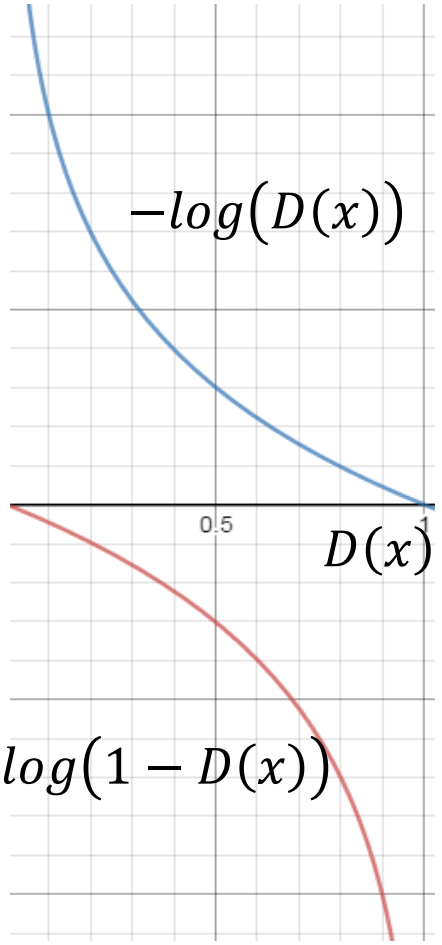
  - 实际中效果并不好，开始时梯度小收敛慢，因此使用如下目标

$$V = E_{z\sim p_z(z)}\Big[-log\Big(D\big(G(z)\big)\Big)\Big]$$

GAN损失函数

$$V = E_{x \sim p_g(x)}\left[-log\left(D(x)\right)\right]$$

$$V = E_{x \sim p_g(x)}\left[log\left(1 - D(x)\right)\right]$$



GAN模型训练

- GAN在训练的过程中固定一方，更新另一方的网络权重
- 交替迭代，在这个过程中，双方都极力优化自己的网络，从而形成竞争对抗，直到双方达到一个动态的平衡（纳什均衡）
- 此时生成模型的数据分布无限接近训练数据的分布（造出了和真实数据一模一样的样本），判别模型再也判别不出来真实数据和生成数据，准确率为 50%

GAN模型训练

❑ 固定G，训练D时，最优的判别器为

$$D_G^* = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

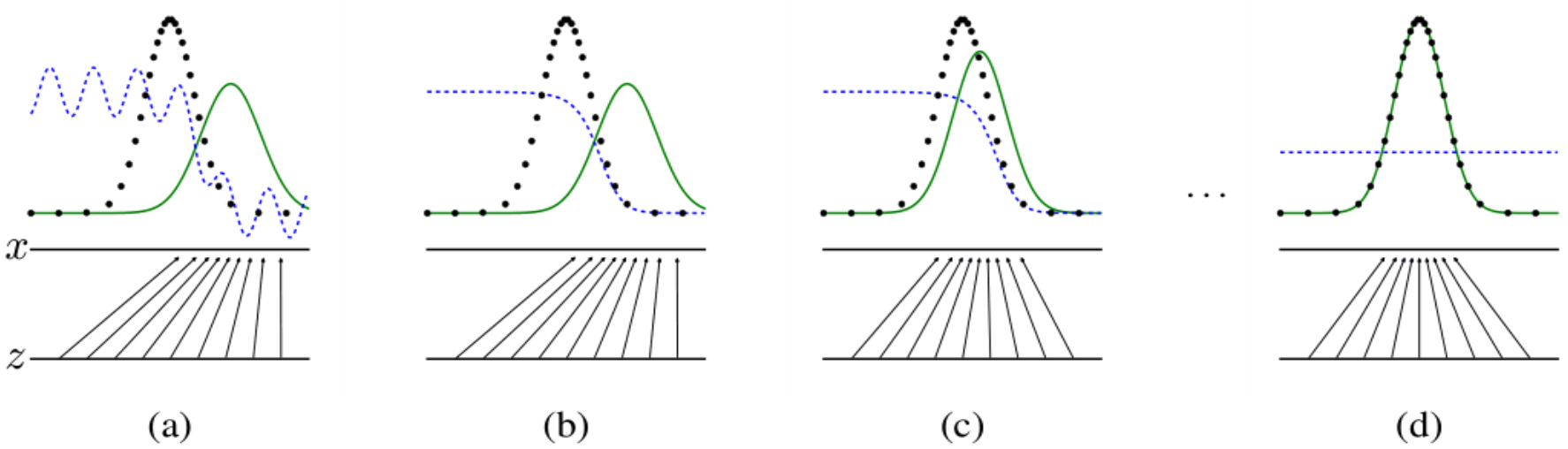
– 证明：

$$\max V(G,D) = \int_{\boldsymbol{x}} p_{data}(\boldsymbol{x}) \log\big(D(x)\big)\, dx + \int_{\boldsymbol{z}} p_{\boldsymbol{z}}(\boldsymbol{z}) \log\Big(1 - D\big(G(\boldsymbol{z})\big)\Big)\, dz$$

$$= \int_{\boldsymbol{x}} \underbrace{p_{data}(\boldsymbol{x}) \log\big(D(x)\big) + p_{\boldsymbol{g}}(\boldsymbol{x}) \log\big(1 - D(\boldsymbol{x})\big)}_{f(\boldsymbol{x}, D)}\, dx$$

$$\frac{\partial f(\boldsymbol{x},D)}{\partial D(\boldsymbol{x})} = \frac{p_{data}(\boldsymbol{x})}{D(\boldsymbol{x})} - \frac{p_g(\boldsymbol{x})}{1 - D(\boldsymbol{x})} = \mathbf{0} \quad \Rightarrow \quad D_G^* = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

GAN模型训练



- 黑色虚线P(x)是真实的数据分布
- 绿线G(z)是通过生成模型产生的数据分布
- 蓝色虚线D(x)代表判别函数
- 较低的水平线是z采样的区域，在这种情况下，上面的水平线是x域的一部分。向上箭头显示映射x=g(z)如何将非均匀分布的P_g强加于转换后的样本上。

1.P_g和P_{data}相似，D是部分精确的分类器

2. D被训练以区分样本和生成数据，并收敛到

$$D_G^* = \frac{p_{data}(\boldsymbol{x})}{p_{data}(\boldsymbol{x}) + p_g(\boldsymbol{x})}$$

3.在更新G之后，D的梯度引导G(z)流向更有可能被归类为数据的区域。

4.产生的绿色分布和真实数据分布已经完全重合。这时，判别函数对所有的数据（无论真实的还是生成的数据），输出都是一样的值， 已经不能正确进行分类。G成功学习到了数据分布，这样就达到了GAN的训练和学习目的。P_g= P_{data} 判别器无法区分这两个分布，此时D(x)=1/2

GAN模型训练

❑ 全局最优的生成函数值-log4

$$C(G) = \max_D V(G,D) = V(G,D^*)$$

$$= E_{x\sim p_{data}}\left[\log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}\right] + E_{x\sim p_g}\left[\log\frac{p_g(x)}{p_{data}(x)+p_g(x)}\right]$$

$$= \int_x p_{data}(\boldsymbol{x})\log\left(\frac{\frac{1}{2}\,p_{data}(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\right)dx + \int_x p_{\boldsymbol{g}}(\boldsymbol{x})\log\left(\frac{\frac{1}{2}\,p_g(x)}{\frac{p_{data}(x)+p_g(x)}{2}}\right)dx$$

$$= -2log2 + KL\left(p_{data}\middle\|\frac{p_{data}(x)+p_g(x)}{2}\right) + KL\left(p_{data}\middle\|\frac{p_{data}(x)+p_g(x)}{2}\right)$$

$$= -log4 + 2JSD\left(p_{data}\|p_g(x)\right)$$

Jensen-Shannon divergence

GAN模型训练

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

Learning D

Repeat k times

for k steps **do**

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Sample minibatch of m examples $\{x^{(1)}, \ldots, x^{(m)}\}$ from data generating distribution $p_{\text{data}}(x)$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \right].$$

$$\theta_d \leftarrow \theta_d + \eta \nabla \tilde{V}(\theta_d)$$

end for

Learning G

Only Once

- Sample minibatch of m noise samples $\{z^{(1)}, \ldots, z^{(m)}\}$ from noise prior $p_g(z)$.
- Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right).$$

$$\theta_g \leftarrow \theta_g - \eta \nabla \tilde{V}(\theta_g)$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

优势和不足

❑ 优势

- 任何一个可微分函数都可以参数化D和G（如深度神经网络）。
- 支持无监督方法实现数据生成，减少了数据标注工作
- 生成模型G的参数更新不是来自于数据样本本身（不是对数据的似然性进行优化），而是来自于判别模型D的一个反传梯度

优势和不足

❑ 不足

- 无需预先建模，数据生成的自由度太大
- 得到的是概率分布，但是没有表达式，可解释性差
- D与G训练无法同步，训练难度大，会产生梯度消失问题

GAN的优化和改进

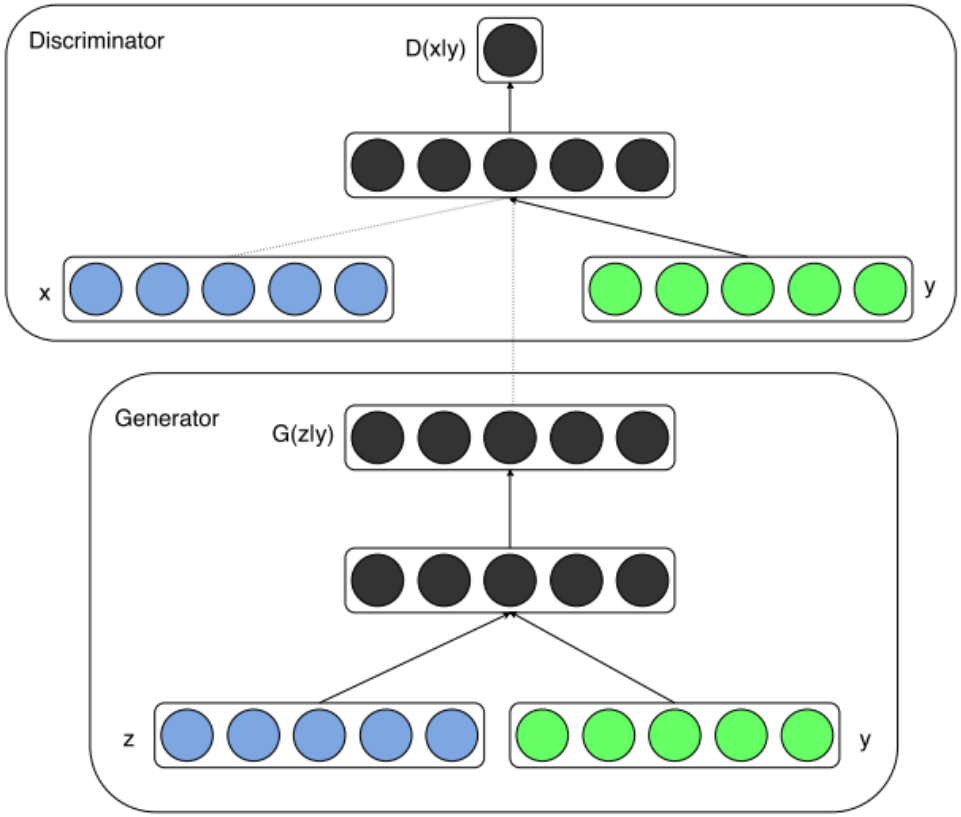
- 限定条件优化
- 迭代式生成优化
- 结构优化

GAN的优化和改进

- 限定条件优化
  - CGAN：Conditional Generative Adversarial Nets
  - Generative Adversarial Text to Image Synthesis
  - InfoGAN： Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets
  - Improved Techniques for Training GANs
  - GP-GAN: Towards Realistic High-Resolution Image Blending

CGAN

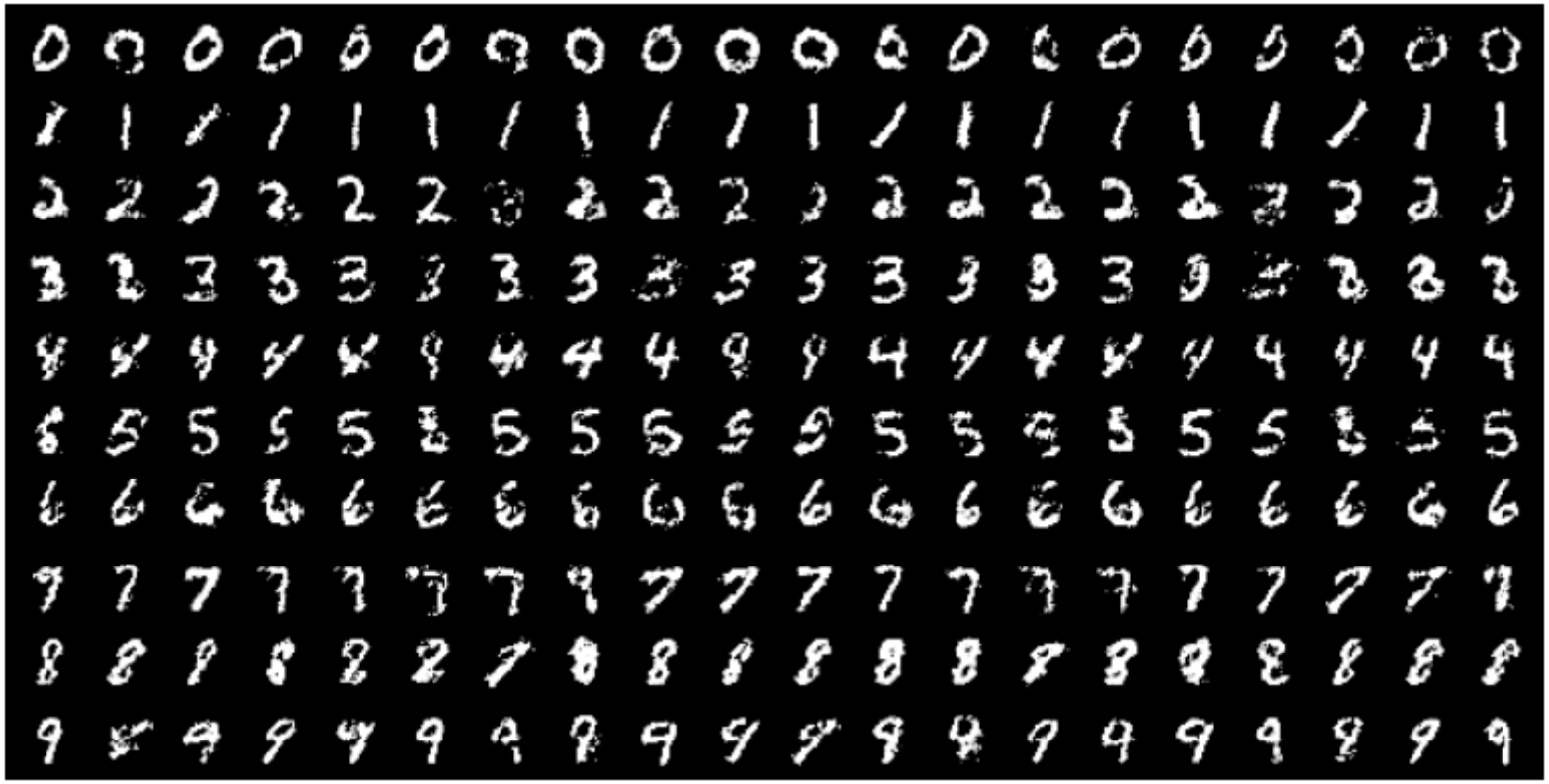
$$\min_G \max_D V(D,G) = E_{\boldsymbol{x}\sim p_{data}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}\left[\log\left(1 - D\left(G(\boldsymbol{z}|\boldsymbol{y})\right)\right)\right]$$



- 在生成模型中，先验输入噪声$p(z)$和条件信息y联合组成了联合隐层表征
- y是一些附加信息，可以是one-hot向量，也可以是图像或者分类标签
- 条件GAN的目标函数是带有条件概率的二人极小极大值博弈(two-player minimax game)

CGAN

❑ CGAN性能

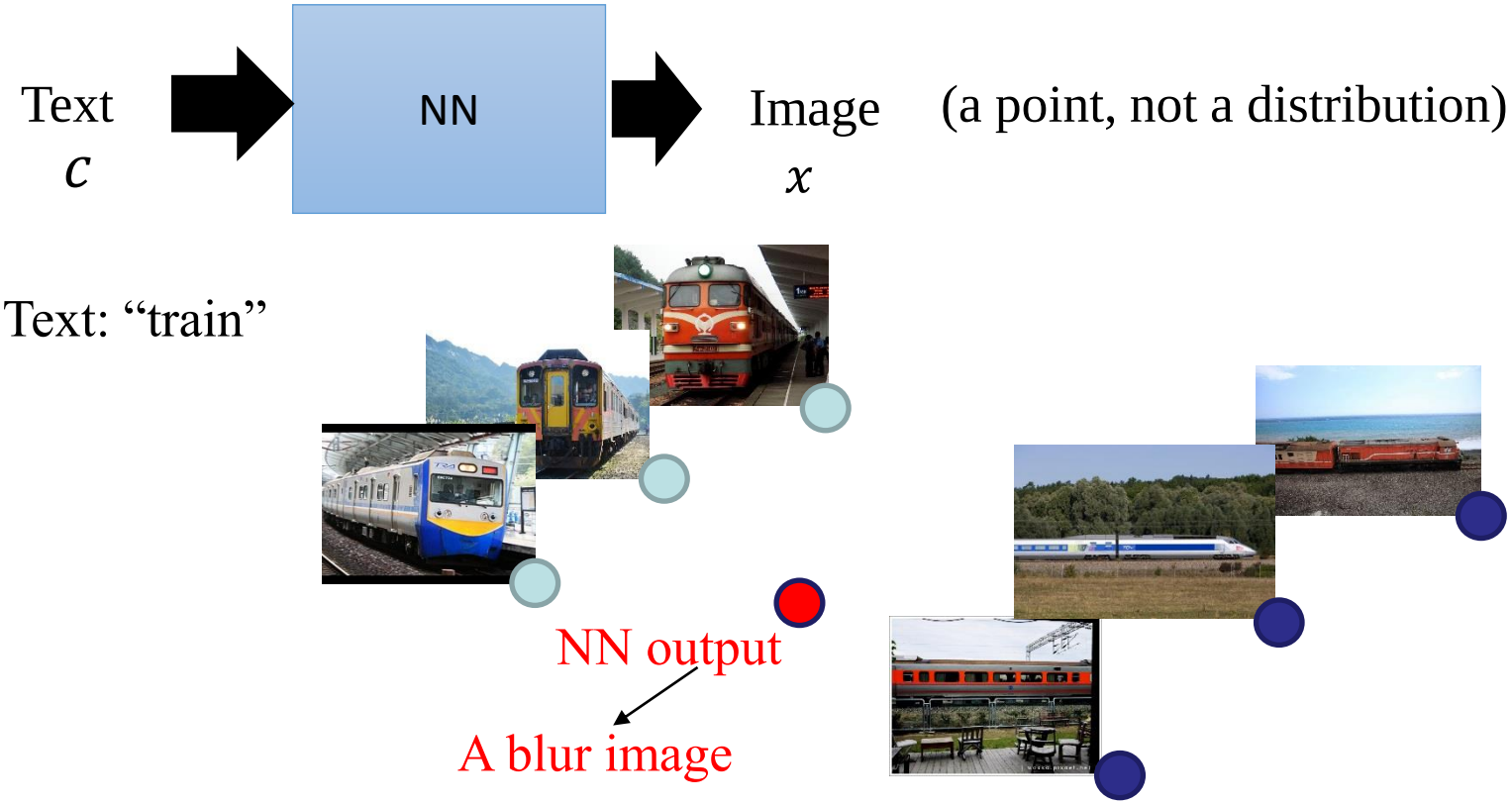


Generated MNIST digits, each row conditioned on one label

Text to Image Synthesis

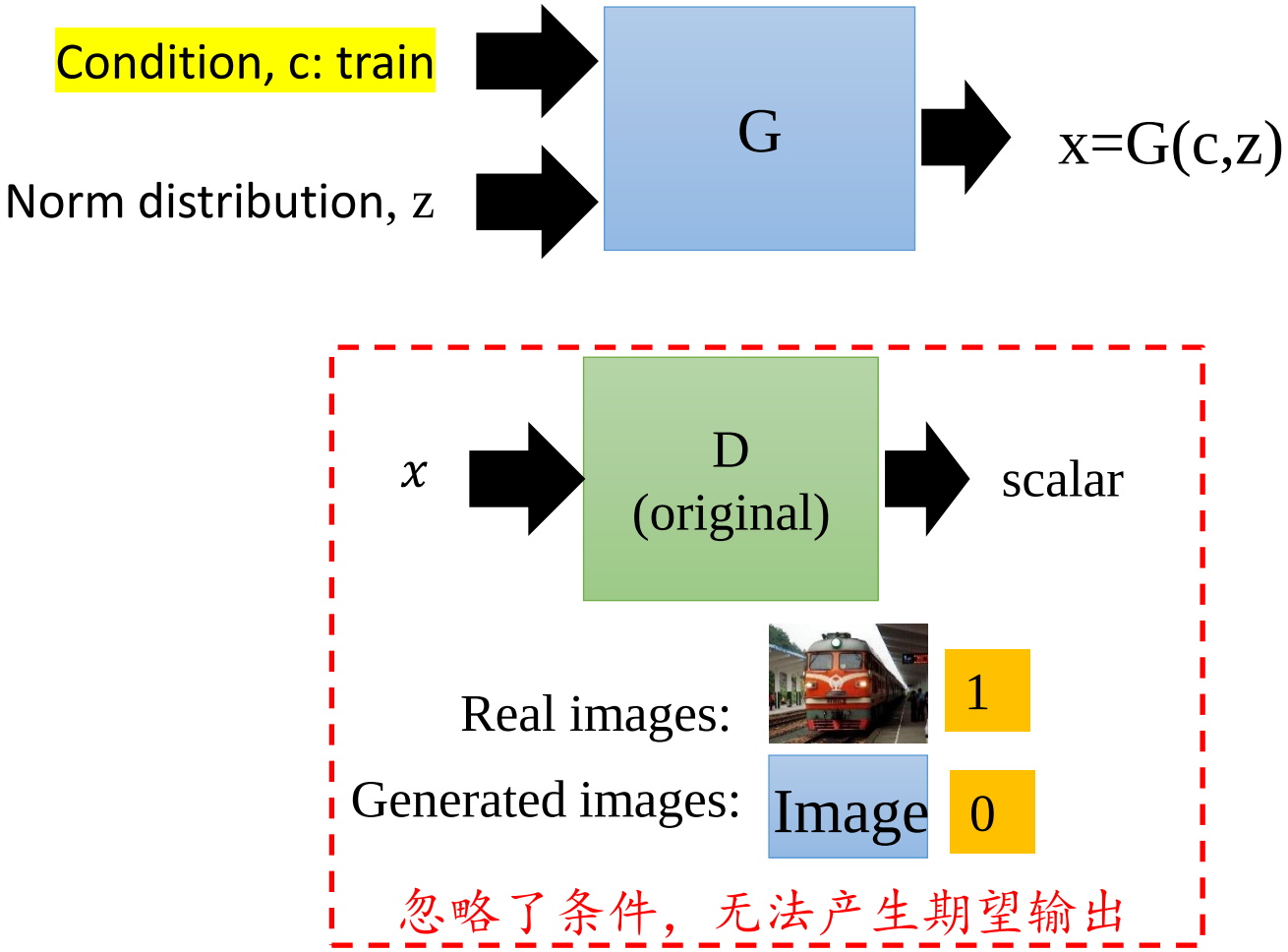
❑ Text to image

– 传统有监督训练方法

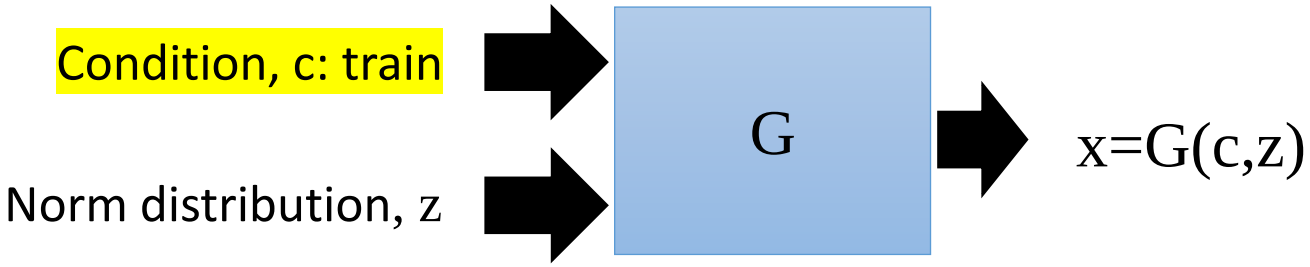


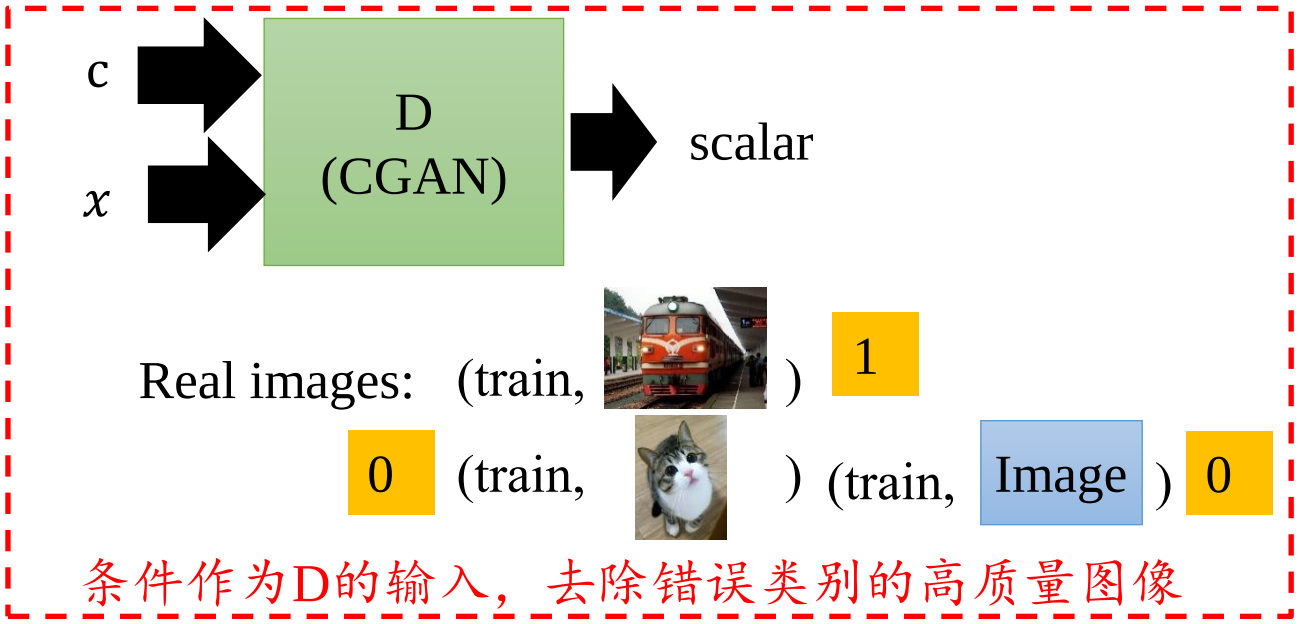
Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee, “Generative Adversarial Text-to-Image Synthesis”, ICML 2016

Text to Image Synthesis



Text to Image Synthesis





Text to Image Synthesis

- In each training iteration
  - Sample m positive examples $\{(c^1,x^1),(c^2,x^2),\ldots,(c^m,x^m)\}$ from database
  - Sample m noise examples $\{z^1,z^2,\ldots,z^m\}$ from a distribution
  - Obtaining generated data $\{\tilde{x}^1,\tilde{x}^2,\ldots,\tilde{x}^m\}$ from $\hat{x}^i = G(c^i, zi)$
  - Sample m objects $\{\hat{x}^1, \hat{x}^2,\ldots, \hat{x}^m\}$ from database
  - Update discriminator parameters to θ_d maximize
    - $V = \frac{1}{m}\sum_{i=1}^{m} \log D\left(c^i, x^i\right)$

      $+\frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(c^i, \tilde{x}^i\right)\right) + \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(c^i, \hat{x}^i\right)\right)$
    - $\theta_d = \theta_d + \lambda\nabla V(\theta_d)$
  - Sample m noise examples $\{z^1,z^2,\ldots,z^m\}$ from a distribution
  - Sample m conditions $\{c^1,c^2,\ldots,c^m\}$ from a database
  - Update generator parameters θ_g to minimize
    - $V = \frac{1}{m}\sum_{i=1}^{m} \log\left(1 - D\left(G\left(c^i, z^i\right)\right)\right),\ \ \theta_g = \theta_g - \lambda\nabla V\left(\theta_g\right)$

InfoGAN

❑ 解决的问题

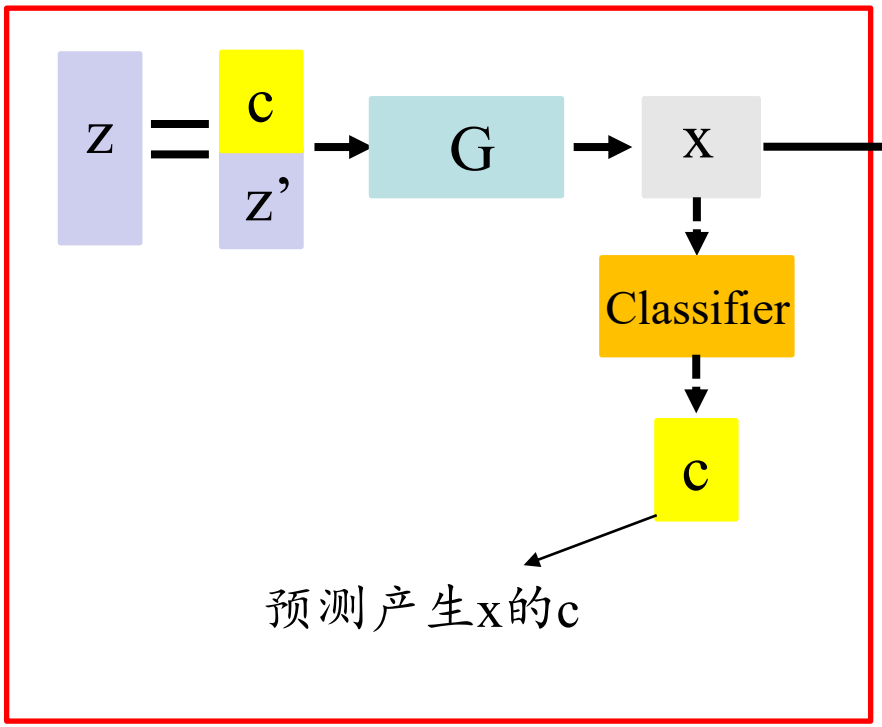
- 输入随机向量，GAN生成模型缺乏解释性
- 例如：修改输入向量某个维度生成图像变化可能没有规律，可能是随机的
- InfoGAN：希望能够让输入向量的不同维度代表不同的特征，具有可解释性

InfoGAN

❑ 基本原理

- 利用z寻找一个可解释的表达，将z进行拆解
  - 不可压缩的噪声z’
  - 可解释的隐变量c, 称为latent code
- 通过约束c与生成数据的关系，使得c里面可以包含某些语义特征
  - 比如MNIST实验的c就可以由一个取值范围为0-9的离散随机变量（用于表示数字）和两个连续的随机变量（分别用于表示倾斜度和粗细度）构成

InfoGAN



- 通过增加正则约束实现

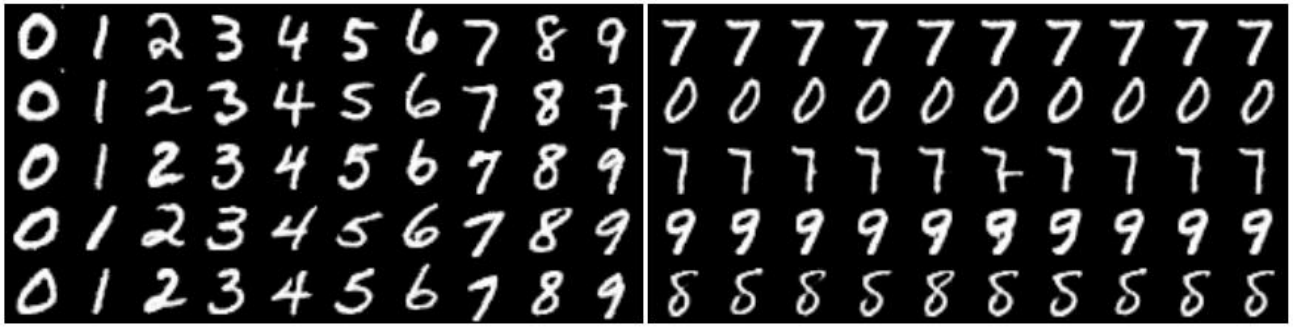
$$\min_G \max_D V_I(D,G) = V(D,G) - \lambda I\big(c; G(z,c)\big)$$

互信息： $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

InfoGAN

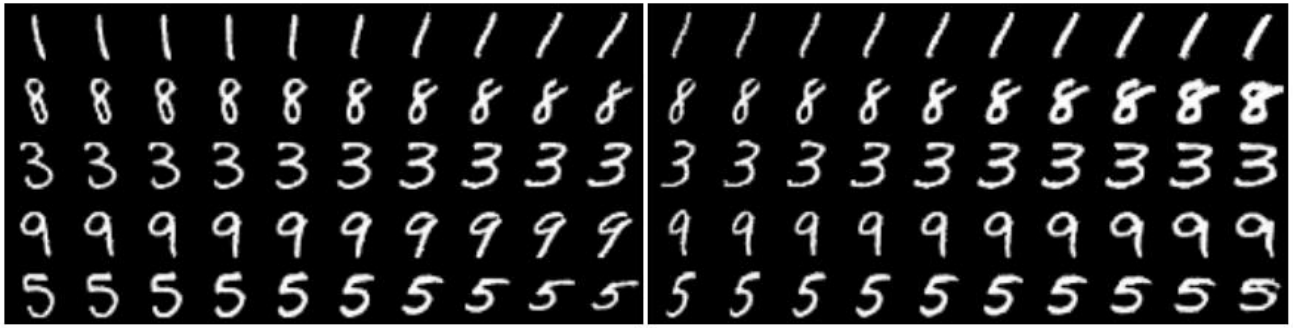
❑ 实验结果

– 从左到右变化c_1, c_2, c_3：c_1控制数字类型, c_2控制向右倾斜, c_3控制字体粗细



(a) Varying c_1 on InfoGAN (Digit type)

(b) Varying c_1 on regular GAN (No clear meaning)



(c) Varying c_2 from -2 to 2 on InfoGAN (Rotation)

(d) Varying c_3 from -2 to 2 on InfoGAN (Width)

InfoGAN

❑ 实验结果

– 控制人脸的生成

(a) Azimuth (pose)　　(b) Elevation

(c) Lighting　　(d) Wide or Narrow

Improved Techniques for Training GANs

❑ 提出了一些训练技巧，提高网络的收敛

- Feature matching：是指把 D 学到特征$f(x)$也“传”给 G，让 G 不仅能知道 D 的输出，还能知道 D 是基于什么输出的
  - 生成器把判别器的中间层输出作为目标，尽量使生成样本的中间输出和真实样本的中间输出相似
  - $\left\| E_{\boldsymbol{x}\sim p_{data}}\left(f(\boldsymbol{x})\right) - E_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}\left(f\left(G(\boldsymbol{z})\right)\right)\right\|_2^2$
- Minibatch discrimination： D 在判断当前传给它的样本是真是假的同时，不要只关注当前的，也要关注其他的样本
- Historical averaging (正则项)： 参数和他过去的时刻有关

$$\left\|\boldsymbol{\theta} - \frac{1}{t}\sum_{i=1}^{t}\boldsymbol{\theta}[i]\right\|^2$$

Improved Techniques for Training GANs

- 提出了一些训练技巧，提高网络的收敛
  - 半监督学习：增加了一个类别K+1表示GAN生成的图像
    - $L = -E_{x,y\sim p_{data}(x,y)}[\log p_{\text{model}}(y|x)] - E_{x\sim p_g(x)}(\log p_{\text{model}}(y = K + 1|x))$

Improved Techniques for Training GANs

❑ 实验结果

– Minibatch discrimination可以提高主观质量

Figure 3: *(Left)* samples generated by model during semi-supervised training. Samples can be clearly distinguished from images coming from MNIST dataset. *(Right)* Samples generated with minibatch discrimination. Samples are completely indistinguishable from dataset images.

Improved Techniques for Training GANs

❑ 实验结果

– 动物皮毛比较相似，但是结构还存在欠缺

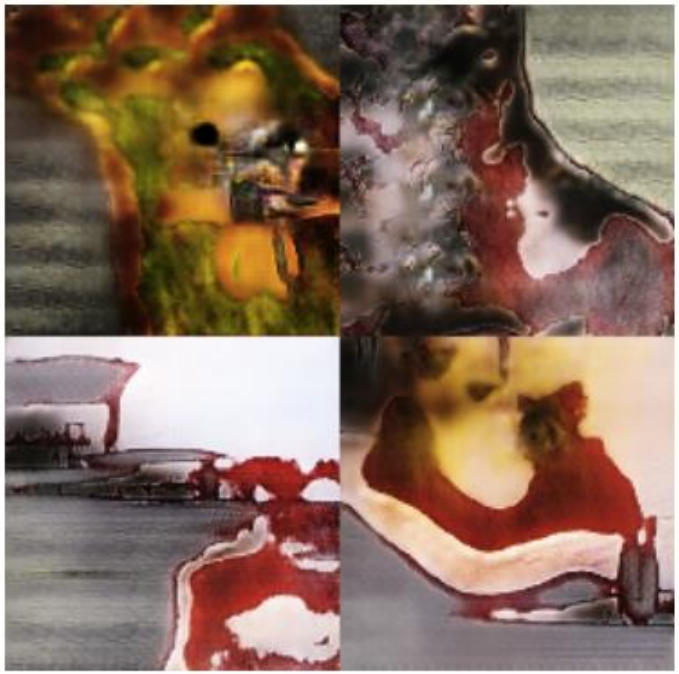


Figure 6: Samples generated from the ImageNet dataset. *(Left)* Samples generated by a DCGAN. *(Right)* Samples generated using the techniques proposed in this work. The new techniques enable GANs to learn recognizable features of animals, such as fur, eyes, and noses, but these features are not correctly combined to form an animal with realistic anatomical structure.

GP-GAN

❑ 解决图像融合问题

- 能在给定简单的复制粘贴合成图像（composited copy-and-paste）的情况下，生成高分辨率、且比较真实的融合图像
- GAN 被用在此领域的第一篇文章

(a) Copy-and-Paste Input (b) MPB [30] (c) MSB [28] (d) Ours

Figure 1: Qualitative illustration of high-resolution image blending. (a) shows the composite copy-and-paste image, where the inserted object is circled out by the red polygon. Our approach (d) produces an image with better quality than those from the alternatives (b) and (c) in terms of illumination, spatial, and color consistencies. Best viewed in color.

GP-GAN

❑ 框架结构

- Blending GAN：作为颜色约束使生成图像更加真实、自然，同时伴随一定的模糊
- 梯度约束进一步提高图像分辨率，使其拥有纹理、边缘等细节

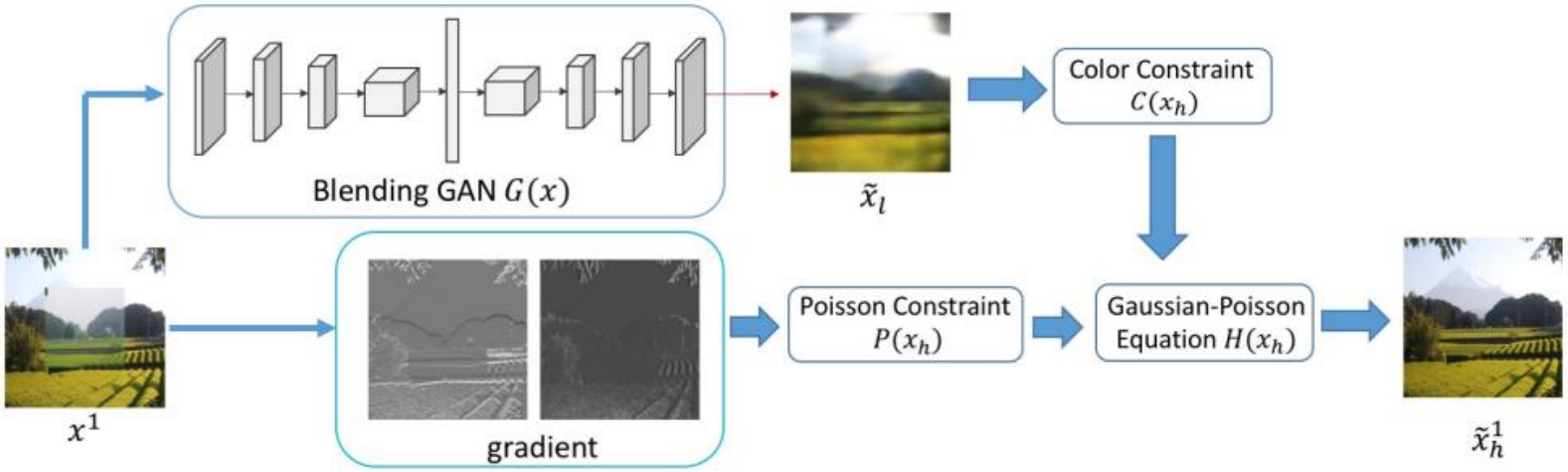


Figure 2: Framework Overview of GP-GAN. Given a composite image x, the low-resolution realistic image $\tilde{x}_l$ is first generated by Blending GAN $G(x)$ with x^1 as the input, where x^1 is the coarsest scale in the Laplacian pyramid of x. Then we optimize the Gaussian-Poison Equation $H(x_h)$ constrained by $C(x_h)$ and $P(x_h)$ with the closed-form solution, resulting in $\tilde{x}_h^1$ that contains rich details. We then upsample $\tilde{x}_h^1$ as the next $\tilde{x}_l$ and optimize the Gaussian-Poisson Equation at a finer scale in the pyramid of x. Best viewed in color.

GP-GAN

❑ Blending GAN结构

$$L(x, x_g) = \lambda L_{l_2}(x, x_g) + (1 - \lambda) L_{adv}(x, x_g)$$

$$L_{l_2}(x, x_g) = \|G(x) - x_g\|_2^2 \qquad \lambda \text{ is } 0.999$$

$$L_{adv}(x, x_g) = \max_D E_{x \in \chi}[D(x_g) - D(G(x))]$$

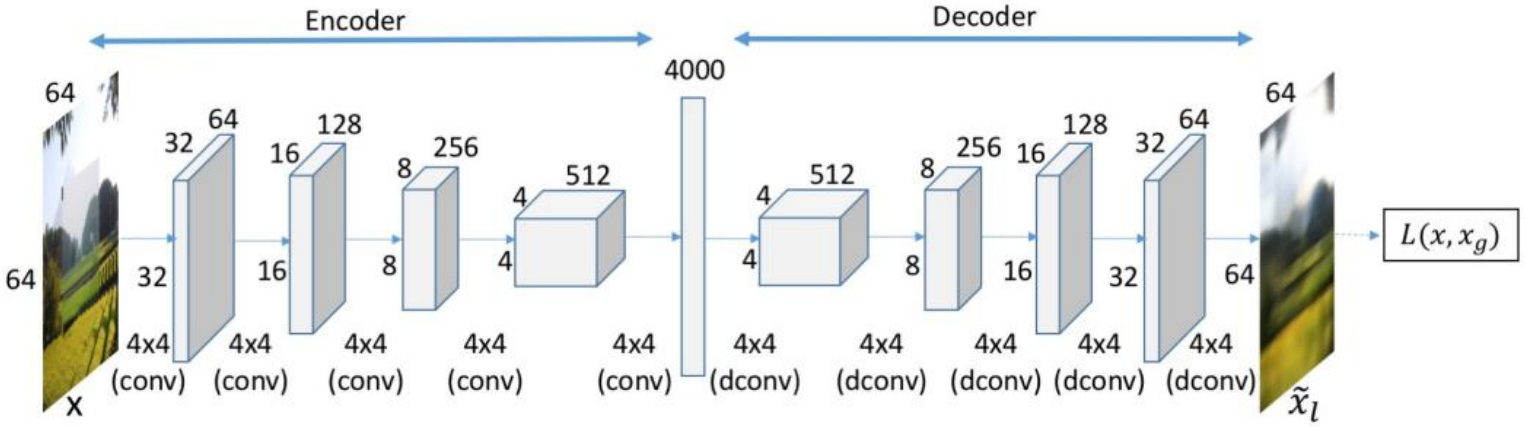


Figure 3: Network architecture of Blending GAN $G(x)$.

GP-GAN

❑ 实验结果

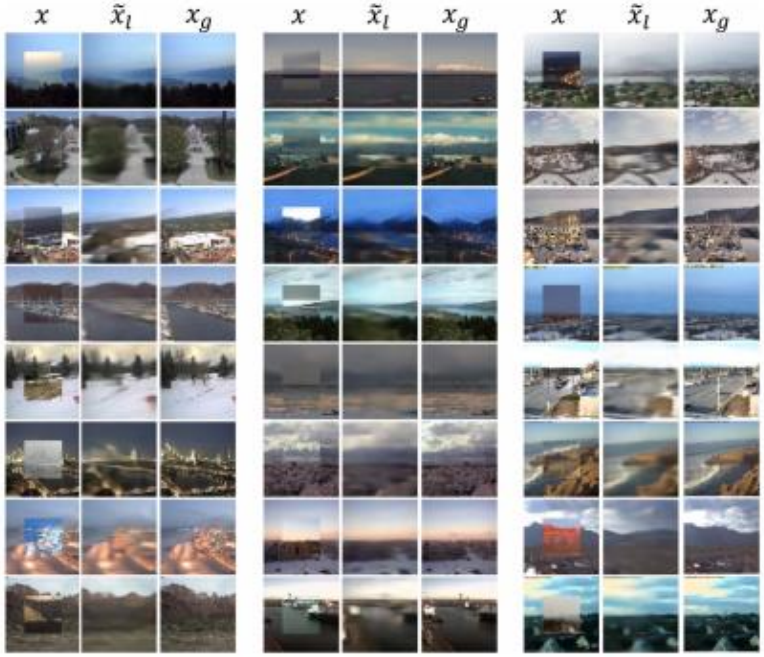


Figure 5: Image blending results generated by $G(x)$. The experiment is conducted on the Transient Attributes Database [18]. x is the copy-and-paste image composited by x_{src} and x_{dst} with a central-squared patch as the mask. $\tilde{x}_l$ is the output of $G(x)$ with size 64×64. x_g is the ground truth image used for training $G(x)$, which is the same as x_{dst}. Best viewed in color.

Figure 10: Results of GP-GAN on real images. The top is the copy-and-paste images and the bottom is the blended images. Best viewed in color.

GAN的优化和改进

- 迭代式生成优化
  - LAPGAN：Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks
  - StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks
  - PPGN: “Plug & Play Generative Networks: Conditional Iterative Generation of Images in Latent Space
- 结构优化
  - DCGAN:Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks
  - Pix2Pix: Image-to-Image Translation with Conditional Adversarial Networks

LAPGAN

❑ 解决的问题

– 高分辨率图像生成

- 在原始 GAN和CGAN中，还只能生成 16*16, 28*28, 32*32 这种低像素小尺寸的图片
- 而LAPGAN首次实现 64*64 的图像生成。采用由coarse-to-fine的渐进生成方式，每一步生成的时候，可以基于上一步的结果，只需要“填充”和“补全”新图片所需要的那些信息

LAPGAN

❑ 生成的结构框图

– 输入噪声z_3，经G_3生成$\tilde{I}_3$，上采样得到l_2, 使用l_2作为条件输入G_2生成差分图像$\tilde{h}_2$，添加到l_2中生成$\tilde{I}_2$

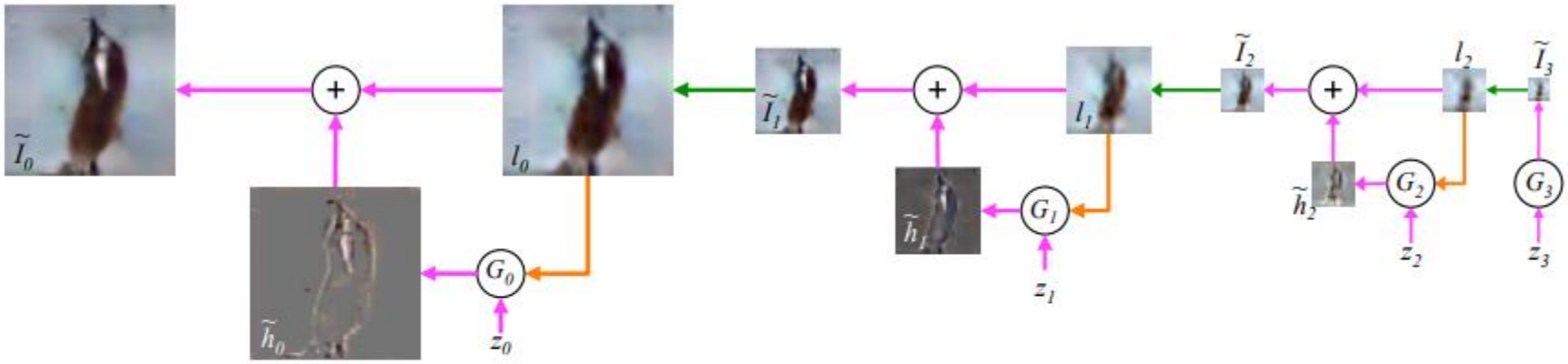
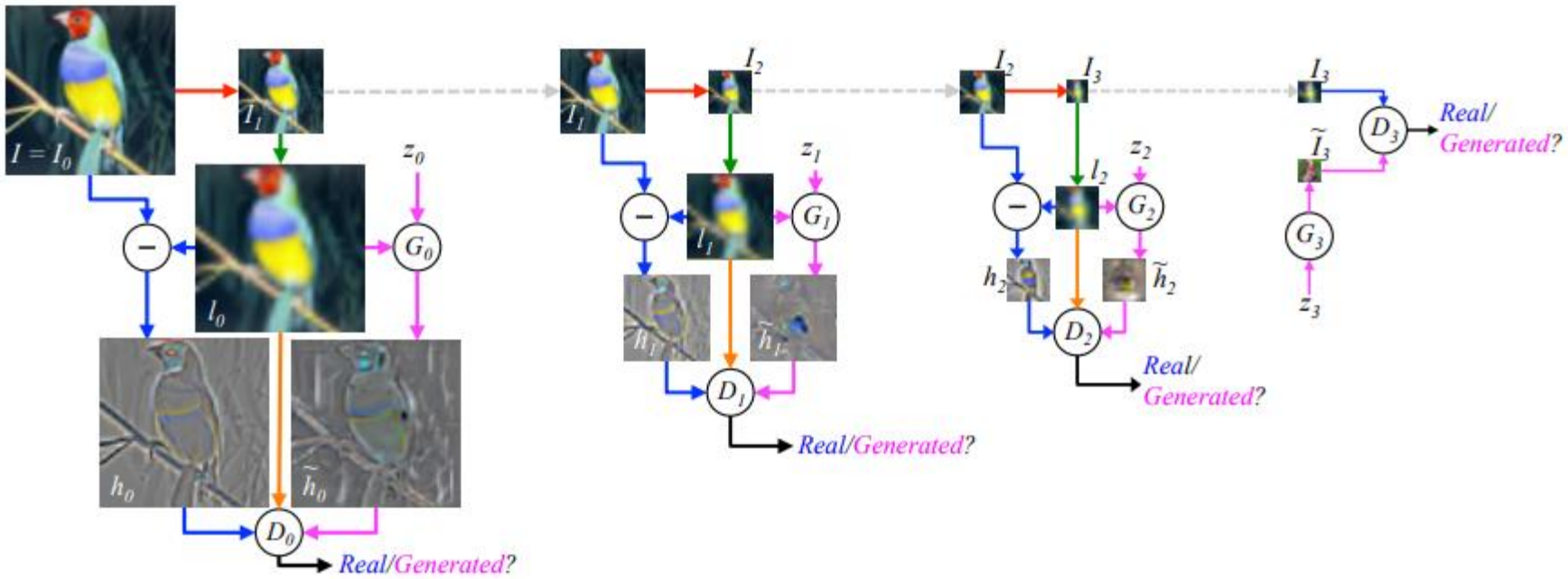


Figure 1: The sampling procedure for our LAPGAN model. We start with a noise sample z_3 (right side) and use a generative model G_3 to generate $\tilde{I}_3$. This is upsampled (green arrow) and then used as the conditioning variable (orange arrow) l_2 for the generative model at the next level, G_2. Together with another noise sample z_2, G_2 generates a difference image $\tilde{h}_2$ which is added to l_2 to create $\tilde{I}_2$. This process repeats across two subsequent levels to yield a final full resolution sample I_0.

LAPGAN

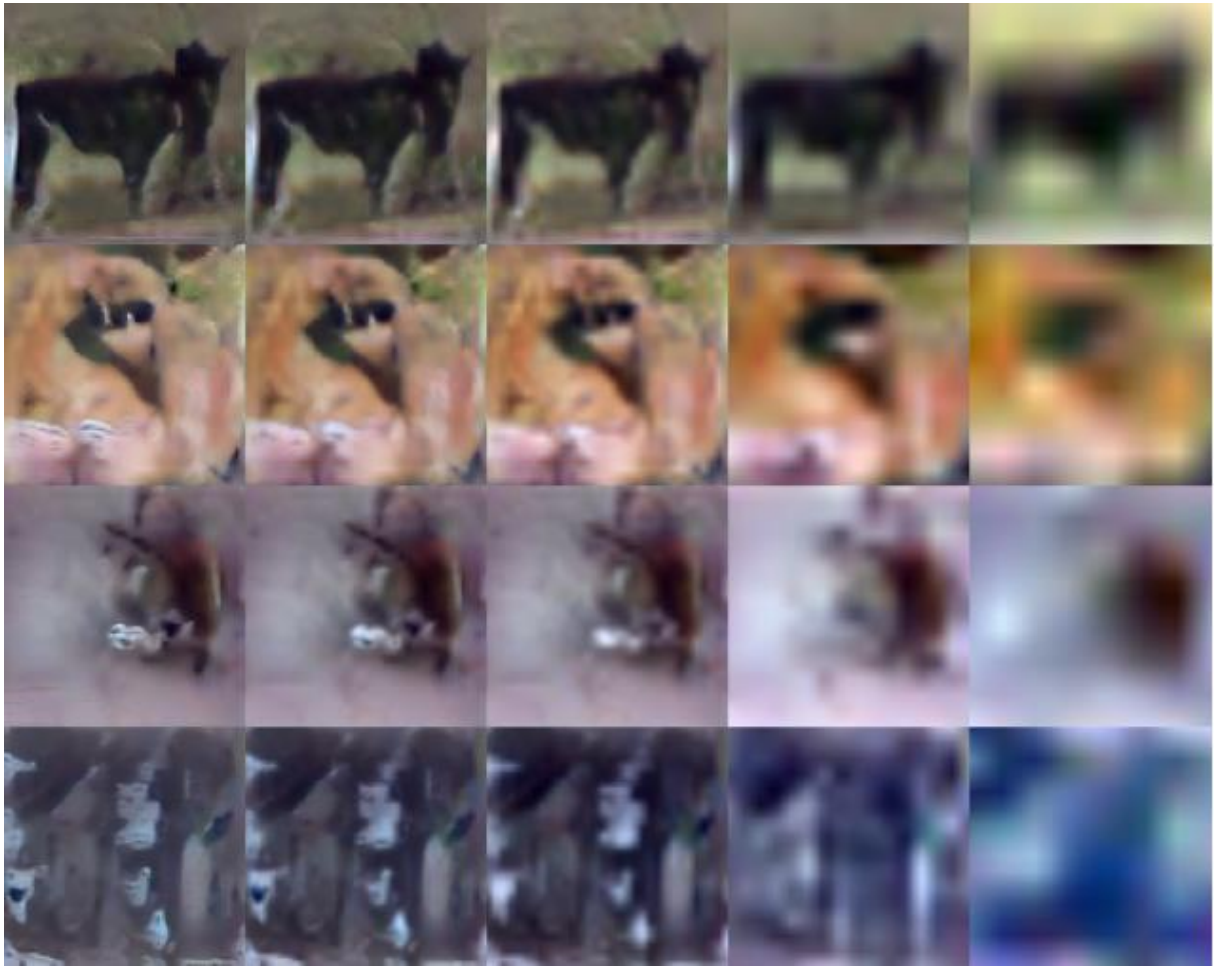
❑ 训练过程

- 输入64×64图像，$I_0=I$, 模糊和2倍因子下采样(红线)，产生I_1
- 2倍上采样I_1(绿线)生成I_0的低通版本l_0，用于生成真实的高频信息和低频信息



LAPGAN

- Coarse-to-fine的视觉效果



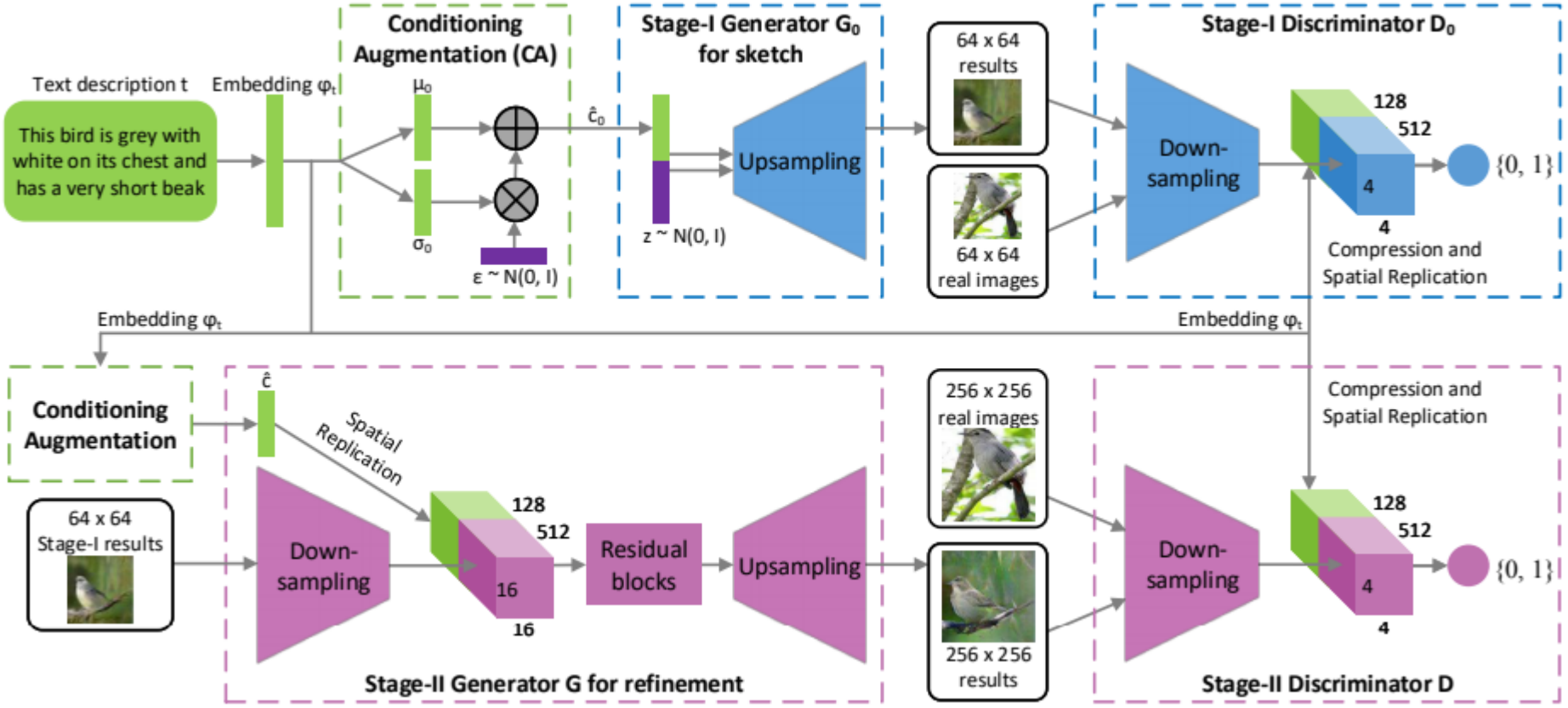
StackGAN

❑ 文本到图像生成

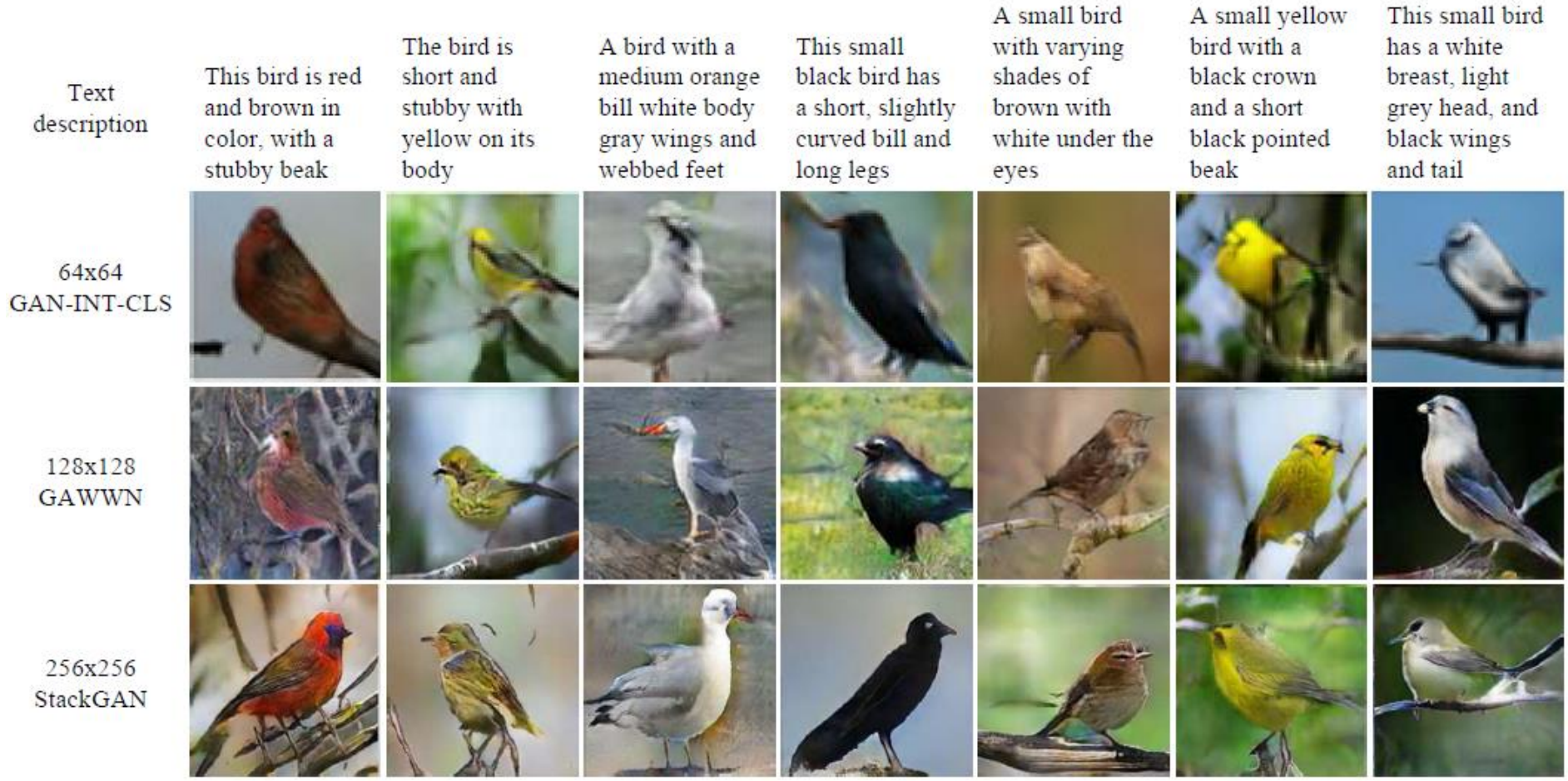
– 解决文本到图像生成分辨率不高的问题，采用与LAPGAN相似的思路，采用coarse-to-fine的思路，构建两个GAN

- 第一个GAN用于根据文本描述生成一张低分辨率的图像
- 第二个GAN将低分辨率图像和文本作为输入，修正之前生成的图像并添加细节纹理，生成高分辨率图像

StackGAN


Text description t
This bird is grey with white on its chest and has a very short beak
Embedding φ_t
Conditioning Augmentation (CA)
μ_0
σ_0
$\varepsilon \sim N(0, I)$
$\hat{c}_0$
Stage-I Generator G_0 for sketch
$z \sim N(0, I)$
Upsampling
64 x 64 results
64 x 64 real images
Stage-I Discriminator D_0
Down-sampling
128
512
4
4
{0, 1}
Compression and Spatial Replication
Embedding φ_t
Embedding φ_t
Conditioning Augmentation
$\hat{c}$
Spatial Replication
64 x 64 Stage-I results
Down-sampling
128
512
16
16
Residual blocks
Upsampling
Stage-II Generator G for refinement
256 x 256 real images
256 x 256 results
Compression and Spatial Replication
Down-sampling
128
512
4
4
{0, 1}
Stage-II Discriminator D

StackGAN


Text description
This bird is red and brown in color, with a stubby beak
The bird is short and stubby with yellow on its body
A bird with a medium orange bill white body gray wings and webbed feet
This small black bird has a short, slightly curved bill and long legs
A small bird with varying shades of brown with white under the eyes
A small yellow bird with a black crown and a short black pointed beak
This small bird has a white breast, light grey head, and black wings and tail
64x64 GAN-INT-CLS
128x128 GAWWN
256x256 StackGAN

DCGAN

❑ Deep convolutional generative adversarial networks

- 将卷积网络和GAN结合的经典论文
- 考虑GAN训练起来非常不稳定，经常会使得生成器产生没有意义的输出，DCGAN为CNN的网络拓扑结构设置了一系列的限制来使得它可以稳定的训练

DCGAN

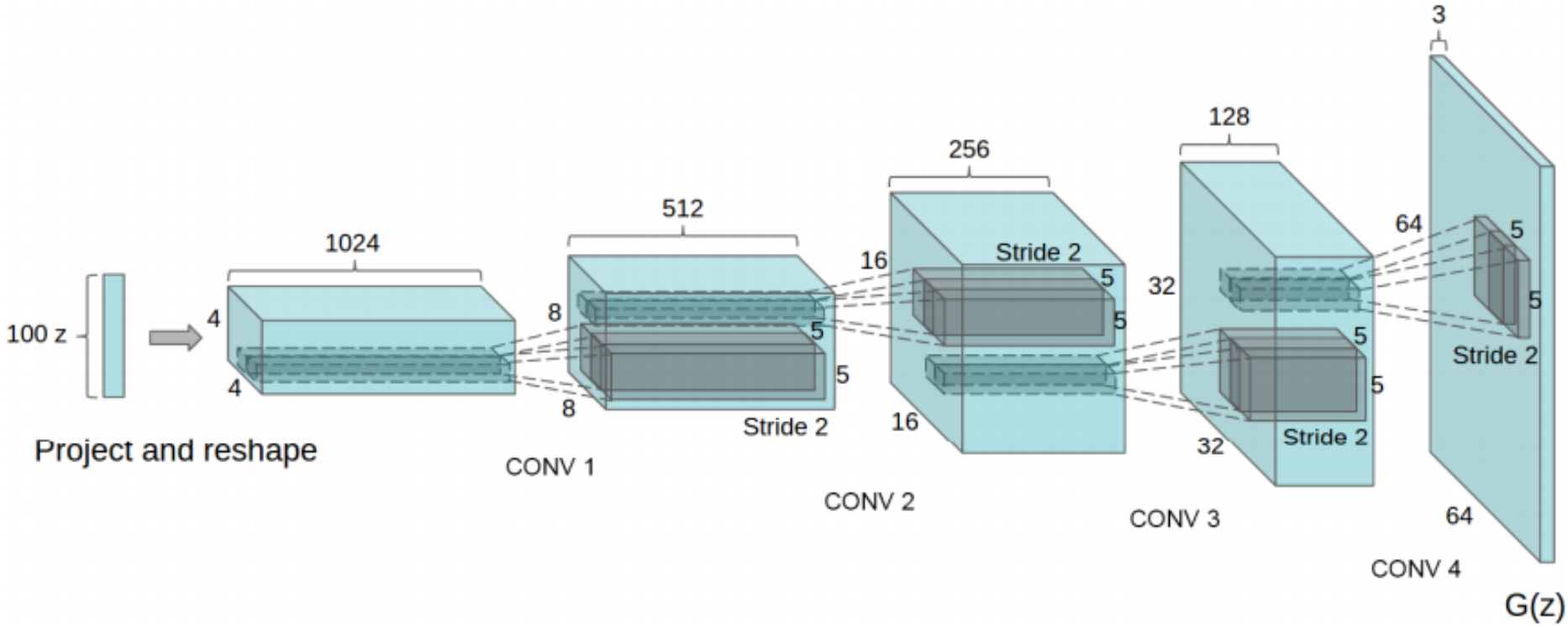
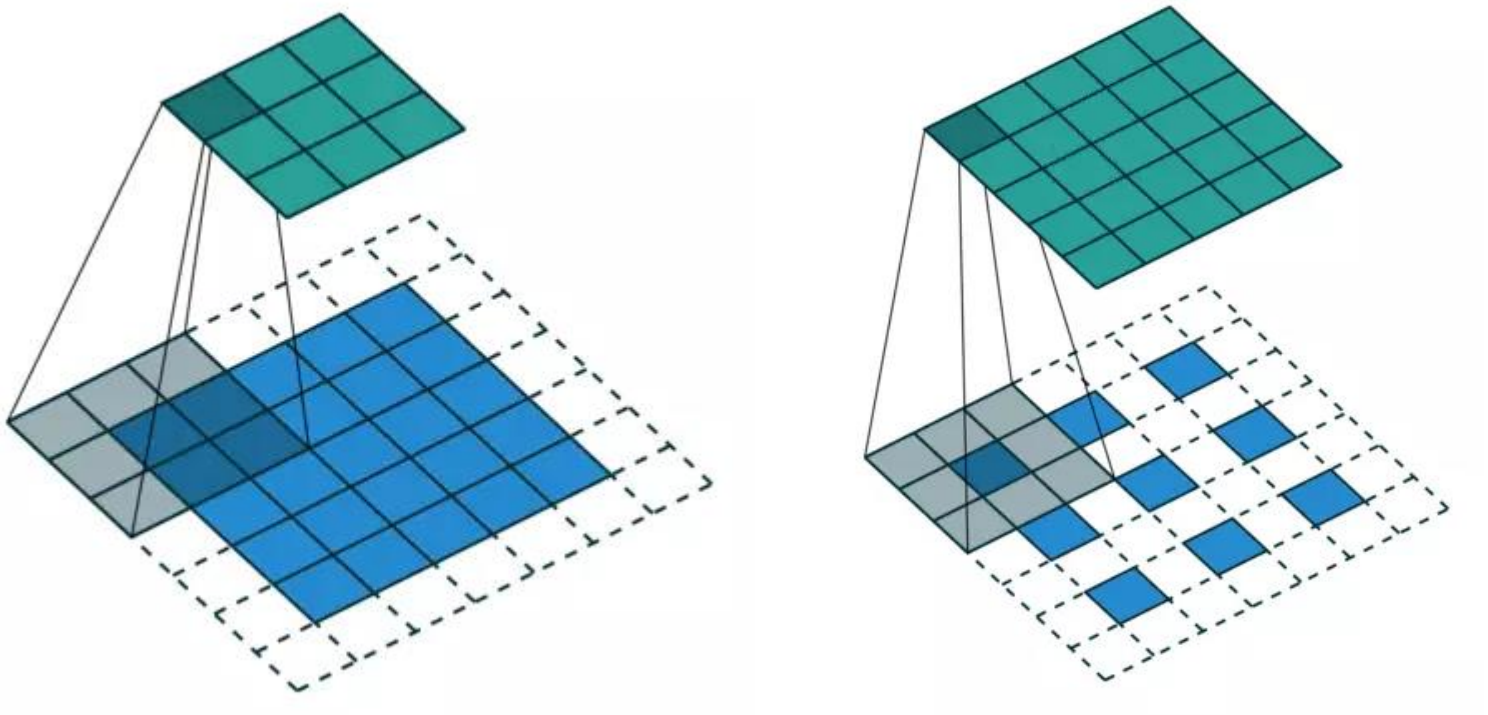


Figure 1: DCGAN generator used for LSUN scene modeling. A 100 dimensional uniform distribution Z is projected to a small spatial extent convolutional representation with many feature maps. A series of four fractionally-strided convolutions (in some recent papers, these are wrongly called deconvolutions) then convert this high level representation into a 64×64 pixel image. Notably, no fully connected or pooling layers are used.

DCGAN

❑ 主要限制

– 所有的pooling层使用strided convolution (判别网络)和fractional-strided convolutions (生成网络)进行替换



DCGAN

❑ 主要限制

- 所有的pooling层使用strided convolution (判别网络)和fractional-strided convolutions (生成网络)进行替换
- 对于更深的架构移除全连接隐藏层
- 在生成网络三维输出层使用Tanh激活函数，其他层使用ReLU激活函数
- 在判别网络的所有层上使用LeakyReLU激活函数

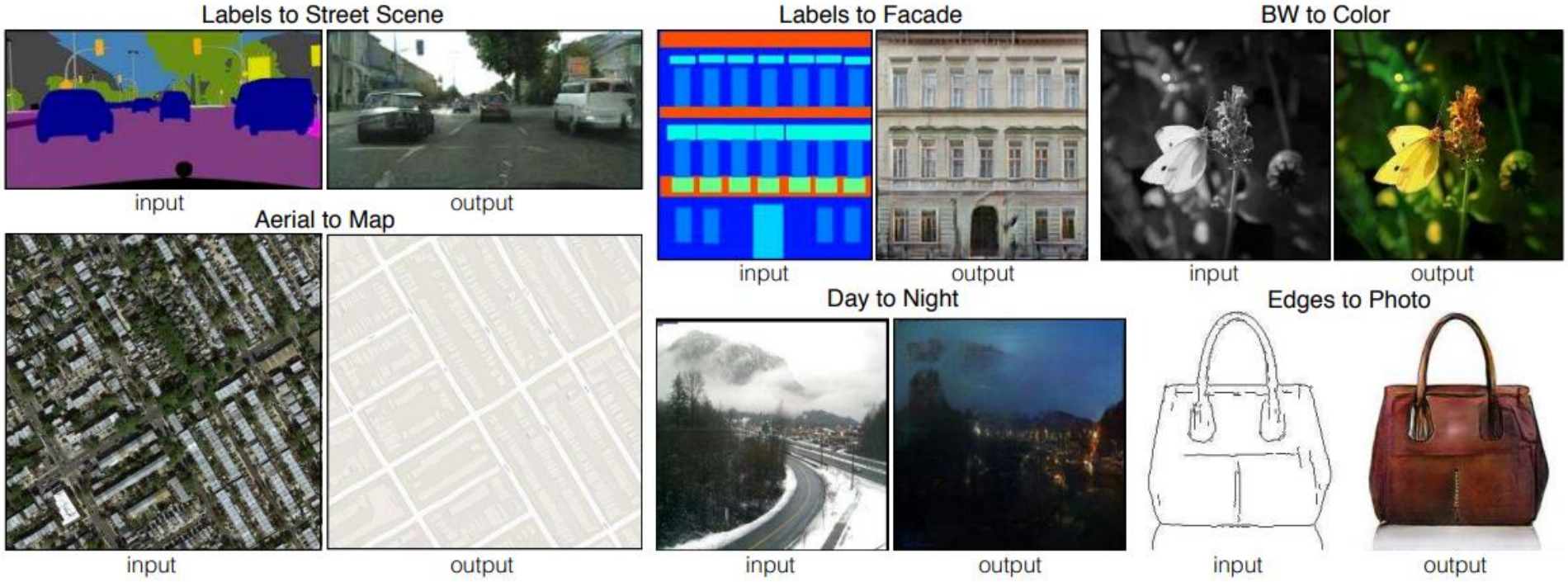
$$y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \frac{x_i}{a_i} & \text{if } x_i < 0, \end{cases}$$

a_i是（1，+∞）区间内的固定参数

DCGAN

GAN的主要应用

❑ 图像转换



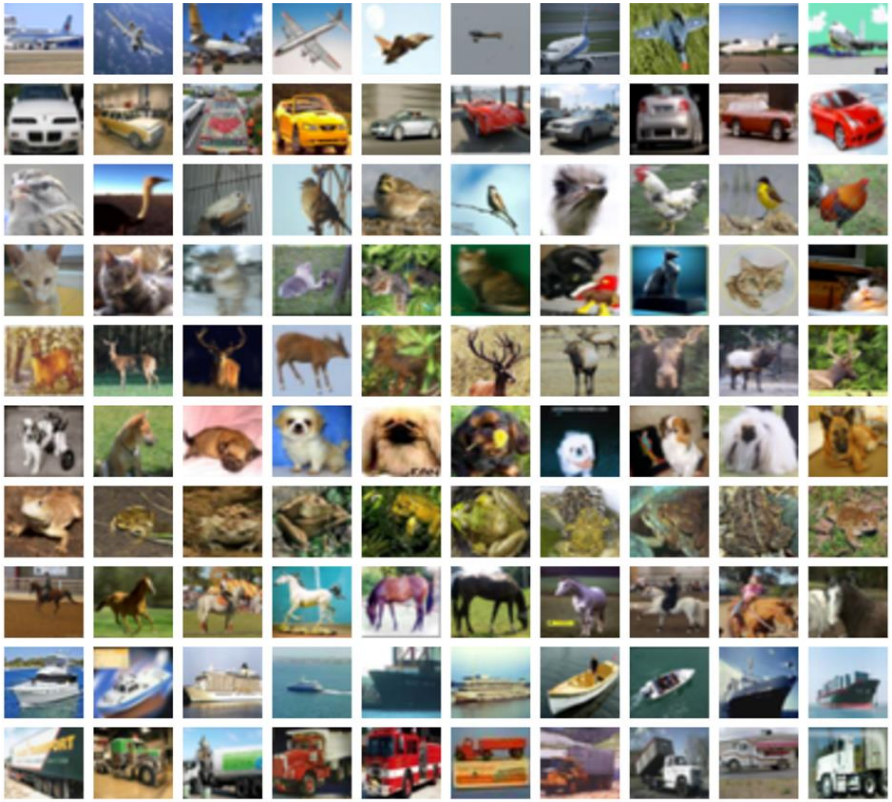
https://affinelayer.com/pixsrv/

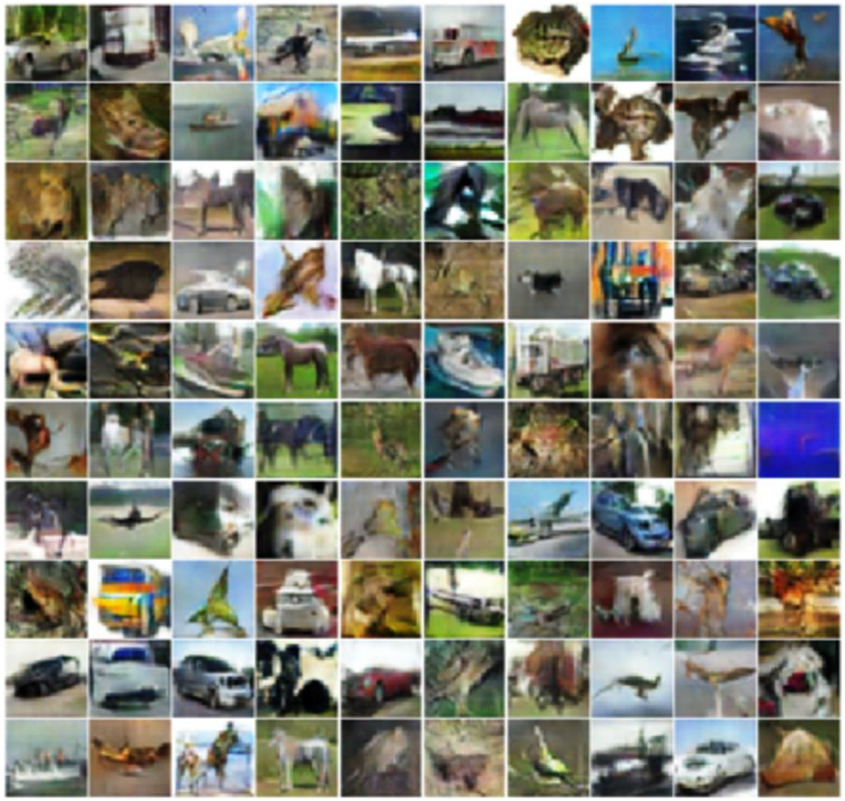
GAN的主要应用

- 图像生成

GAN的主要应用

- 图像生成



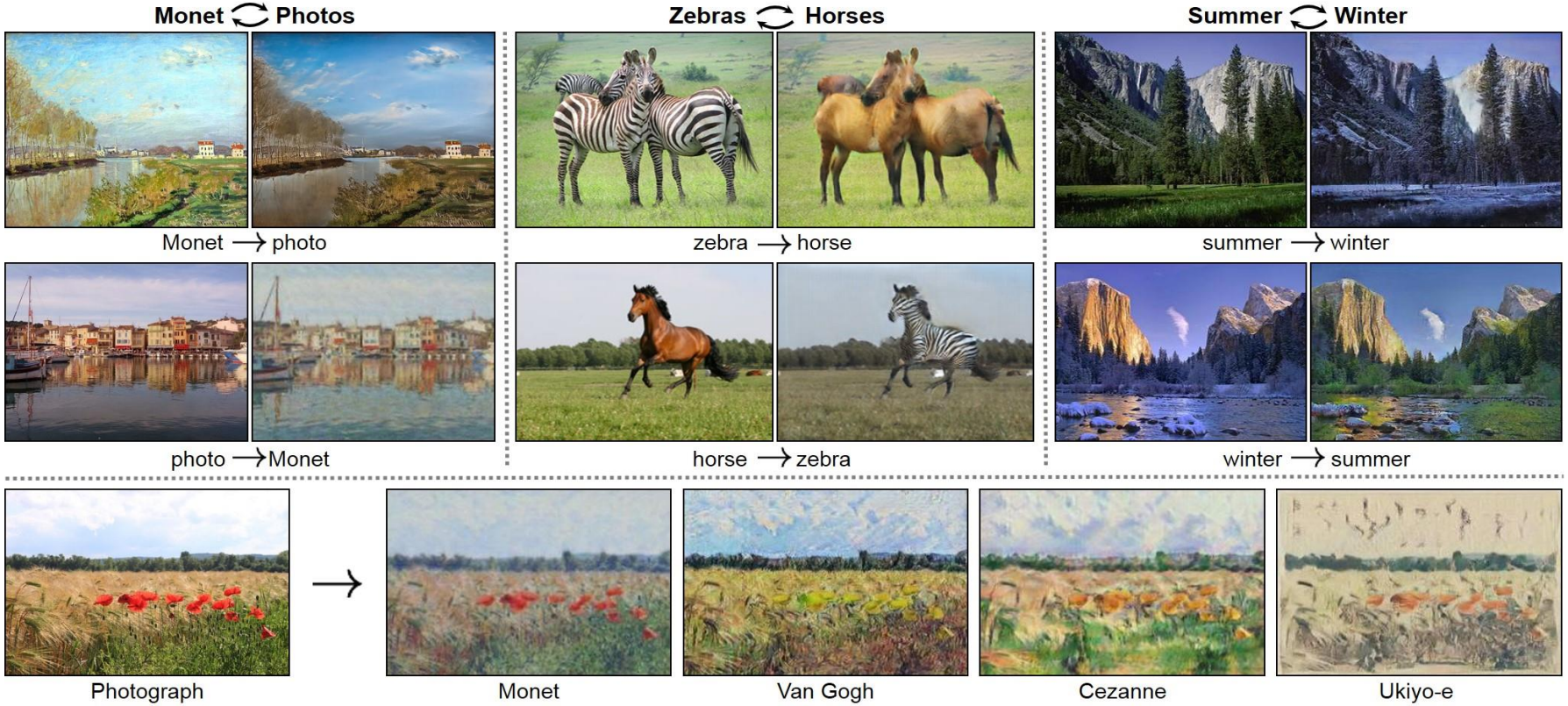


Original CIFAR-10 vs. Generated CIFAR-10 samples

Source: "Improved Techniques for Training GANs" https://arxiv.org/abs/1606.03498

GAN的主要应用

❑ 图像迁移



https://junyanz.github.io/CycleGAN/

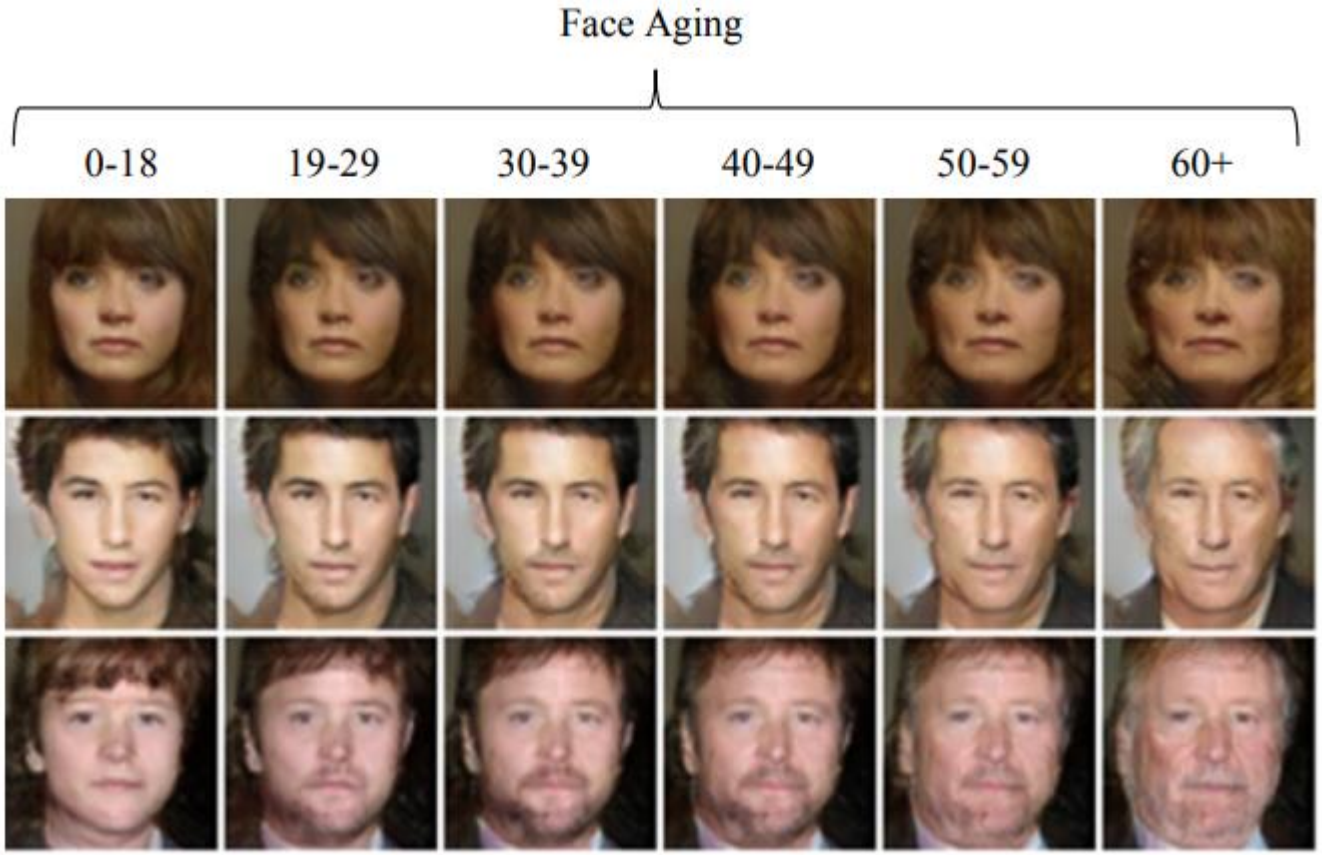
GAN应用

- 图像合成

https://arxiv.org/abs/1704.04086

GAN应用

❑ 图像预测



https://arxiv.org/abs/1702.01983

GAN应用

- 图像修复

https://arxiv.org/abs/1604.07379

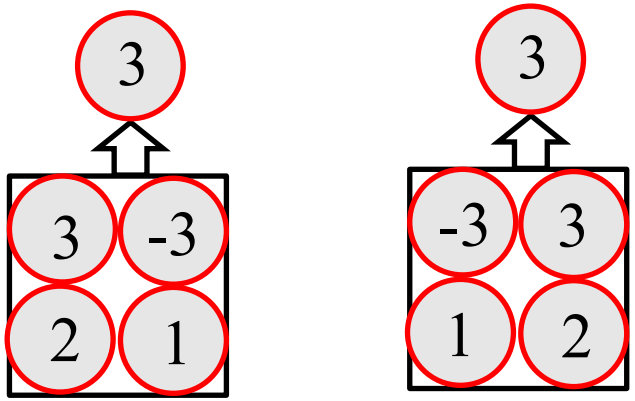
2 胶囊网络

胶囊网络背景简介

❑ CNN现存问题

– 池化操作提供了局部不变性，错误解决了需要解决的等变性问题，从而丢失了位置等信息

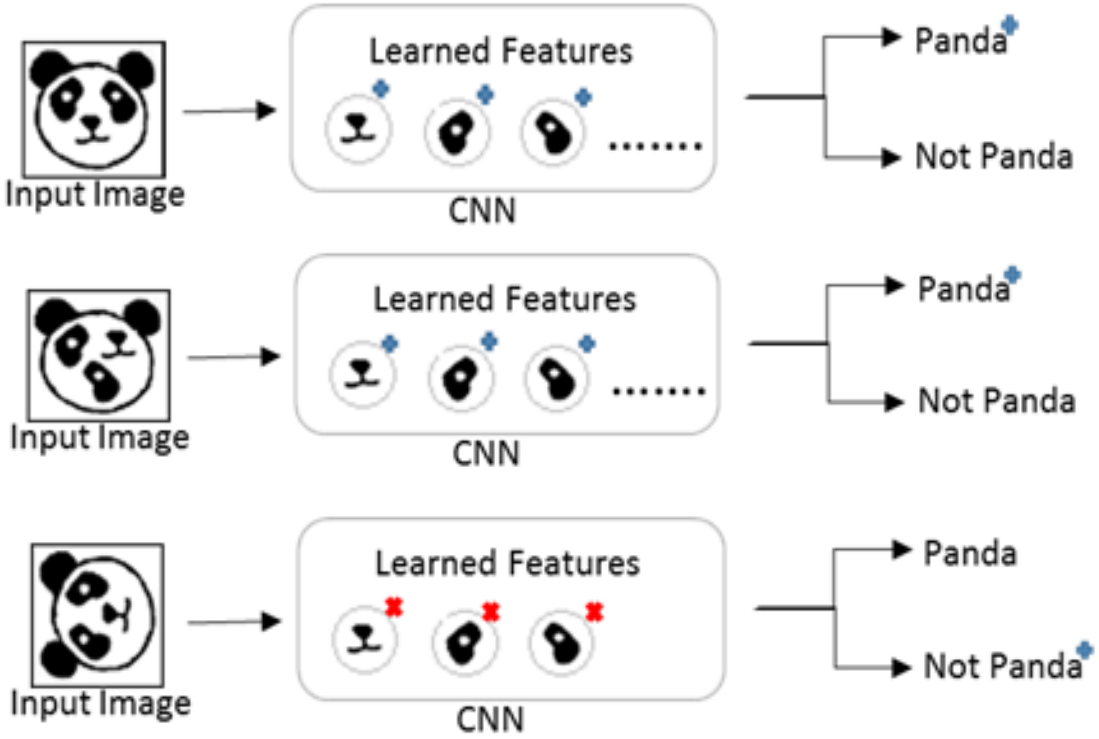
- 平移等变性：对于一个函数，如果对其输入施加的变换也会同样反应在输出上，那么这个函数就对该变换具有等变性
  - 例如：输入0,3,2,0,0产生结果0,1,0,0，那么输入模式0,0,3,2,0 可能会产生0,0,1,0的输出
- 平移不变性：对于一个函数，如果对其输入施加的某种操作丝毫不会影响到输出，那么这个函数就对该变换具有不变性



胶囊网络背景简介

❑ CNN现存问题

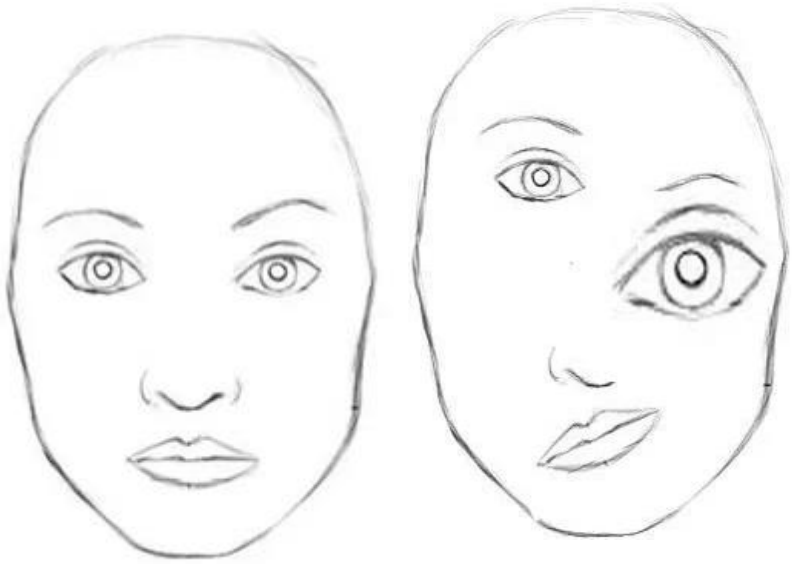
– 池化操作提供了局部不变性，错误解决了需要解决的等变性问题，从而丢失了位置等信息

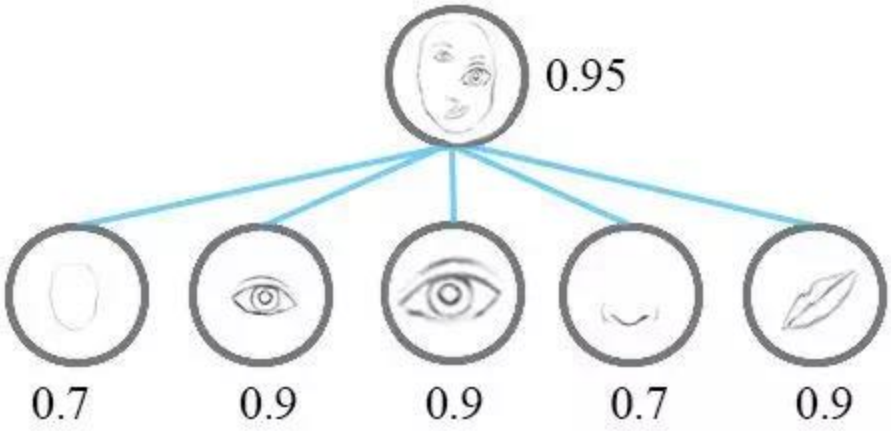


胶囊网络背景简介

❑ CNN现存问题

– 池化操作提供了局部不变性，错误解决了需要解决的等变性问题，从而丢失了位置等信息



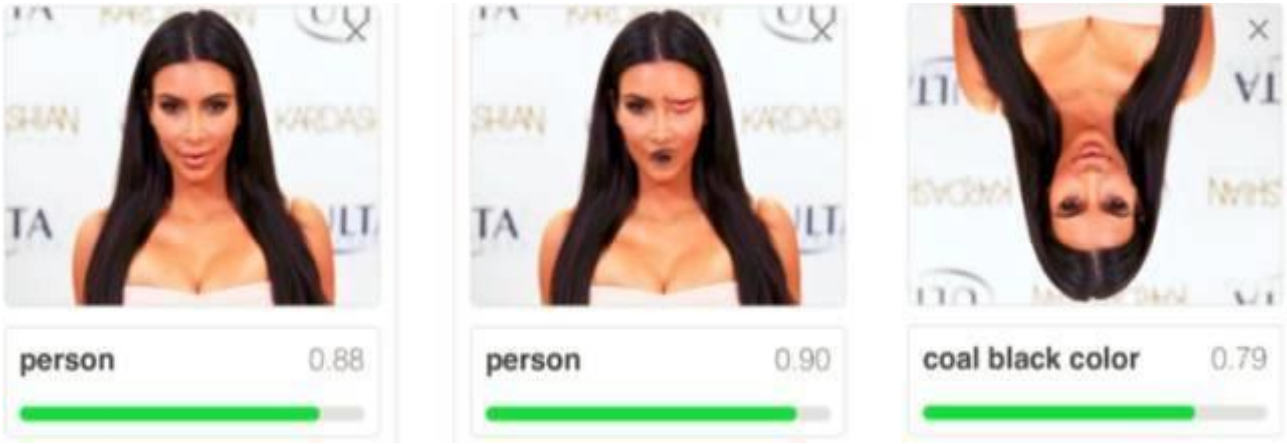


对于CNN而言，两张图片是类似的，因为它们包含相似的部件

胶囊网络背景简介

❑ CNN现存问题

– 池化操作提供了局部不变性，错误解决了需要解决的等变性问题，从而丢失了位置等信息



胶囊网络背景简介

❑ CNN现存问题

– 池化操作提供了局部不变性，错误解决了需要解决的等变性问题，从而丢失了位置等信息

胶囊网络

❑ 胶囊网络的提出

Hinton, Geoffrey E., Alex Krizhevsky, and Sida D. Wang. "Transforming auto-encoders." In *International conference on artificial neural networks*, pp. 44-51. Springer, Berlin, Heidelberg, 2011.

Sabour, Sara, Nicholas Frosst, and Geoffrey E. Hinton. "Dynamic routing between capsules." In Advances in neural information processing systems, pp. 3856-3866. 2017.

胶囊网络背景简介

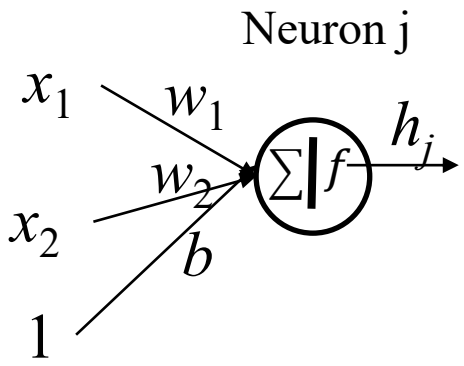
❑ Geoffrey Hinton

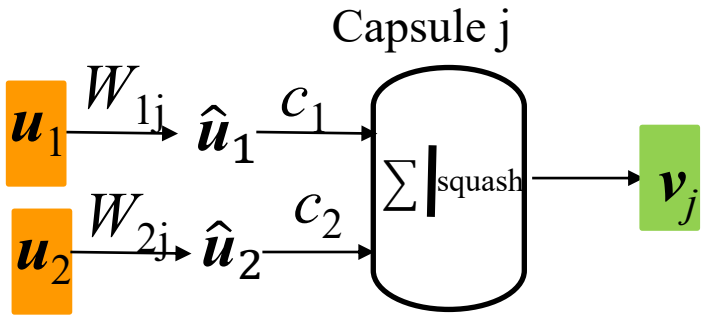
- The pooling operation used in convolutional neural networks is a big mistake and the fact that it works so well is a disaster.
- If the pools do not overlap, pooling loses valuable information about where things are. We need this information to detect precise relationships between the parts of an object. Its true that if the pools overlap enough, the positions of features will be accurately preserved by “coarse coding” (see my paper on “distributed representations” in 1986 for an explanation of this effect). But I no longer believe that coarse coding is the best way to represent the poses of objects relative to the viewer (by pose I mean position, orientation, and scale)

胶囊网络

❑ 胶囊网络的改进

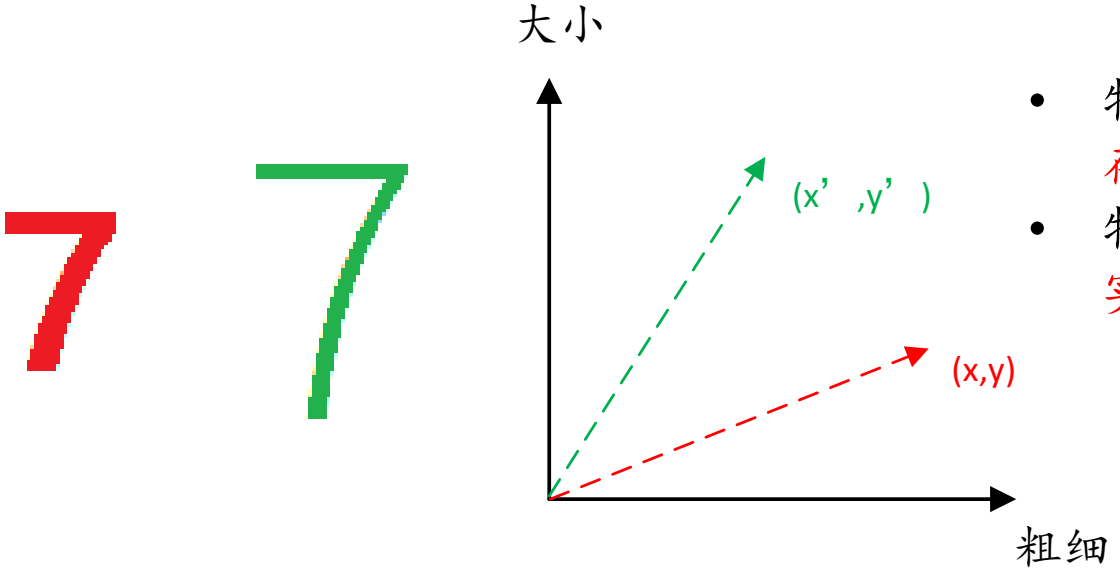
- 使用胶囊作为网络基本单元
- 特征向量表示可视实体，对方位等信息进行编码
- 动态路由算法代替池化操作





特征向量表示可视实体

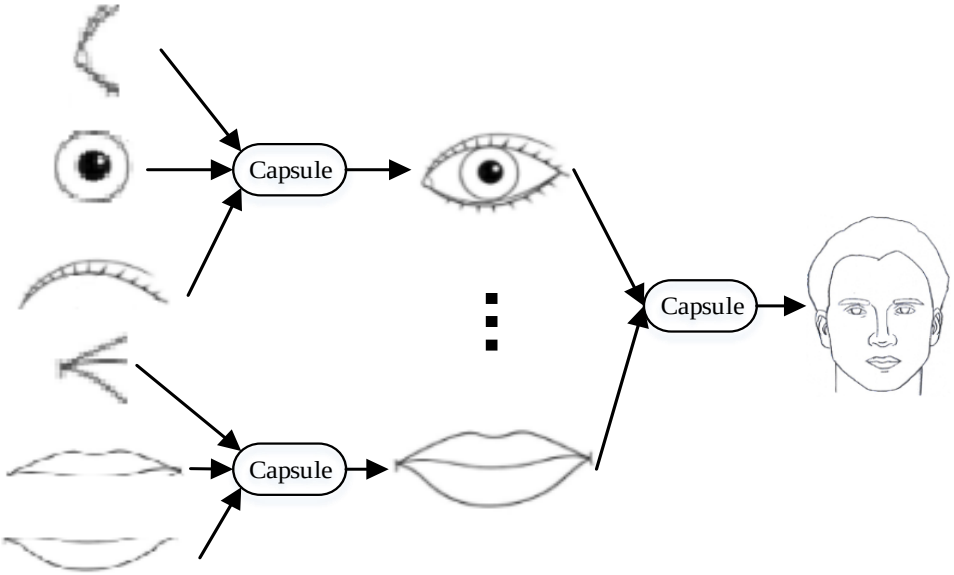
- 实体的存在概率具有局部不变性——当胶囊覆盖的有限视觉域内的实体变换姿态时，它是不变的
- 实体的实例化参数具有等变性——由于实例化参数表示实体的姿态坐标，因此随着实体的姿态变化，实例化参数会相应改变



- 特征向量长度表示实体存在概率；
- 特征向量各维特征表示实体姿态。

胶囊

- 胶囊输出的实例化参数提供了一种由部分到整体的简单识别方法，由单个有限视觉域逐层向上预测，最终形成对目标图像的解析树，达到识别目的。
- 在连接到下一层时，可以做出一个简单选择——实体是否能够由多个激活的具有正确的空间关系的胶囊表示，并且激活更高级别的胶囊



胶囊

- 设胶囊的输入为u_i，使用预测矩阵W_{ij}与u_i相乘，对高层特征向量进行预测，得到预测的特征向量$u_{j|i}$

$$u_{j|i} = W_{ij}u_i$$

- 当前层胶囊对所有预测向量求取加权平均，便得到了向量s_j

$$s_j = \sum_i c_{ij}u_{j|i}$$

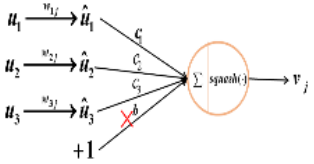
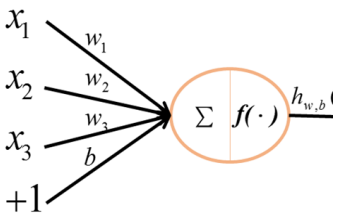
c_{ij}称作耦合系数

胶囊

- 为了使特征向量的长度能够表示实体存在的概率，需要使用非线性压缩(squashing)函数将向量长度限制在0到1之间

$$v_j = \frac{\|s_j\|^2}{1+\|s_j\|^2}\frac{s_j}{\|s_j\|}$$

胶囊与神经元对比

类型	输入	操作				输出	图
		仿射变换	加权	求和	非线性激活		
胶囊	向量 $\mathbf{u_i}$	$\mathbf{u_{j\|i}} = \boldsymbol{W_{ij}} \boldsymbol{u_i}$		$\mathbf{s_j} = \sum_i c_{ij}\, \boldsymbol{u_{j\|i}}$	$\mathbf{v_j} = \dfrac{\|\boldsymbol{s_j}\|^2}{1+\|\boldsymbol{s_j}\|^2} \dfrac{\boldsymbol{s_j}}{\|\boldsymbol{s_j}\|^2}$	向量 $\mathbf{v_j}$	
神经元	标量 $\mathrm{x_i}$	—		$a_j = \sum_i W_{ij} x_i + b$	$\mathrm{h}_j = f(a_j)$	标量 $\mathrm{h_j}$	

胶囊间的动态路由

- 胶囊间的动态路由机制可以确保每一层胶囊输出的特征向量被正确地发送到下一层中对应的胶囊

- 耦合系数c_{ij}是使用Softmax函数来计算的，并且前一层的胶囊i与下一层中的所有胶囊之间的耦合系数总和为1

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

胶囊间的动态路由

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

- 其中，b_{ij}为胶囊i应该耦合到胶囊j的对数先验概率，它取决于两个胶囊的类型和位置，与当前的输入图像无关，其初始值为0，然后通过预测向量$\boldsymbol{u}_{j|i}$和实际输出向量v_j的一致性来修正
- 一个简单的表明一致性关系的函数就是标量积，在计算新的耦合系数之前，要把$a_{ij}=\boldsymbol{v}_j \cdot \boldsymbol{u}_{j|i}$加到初始的$b_{ij}$上

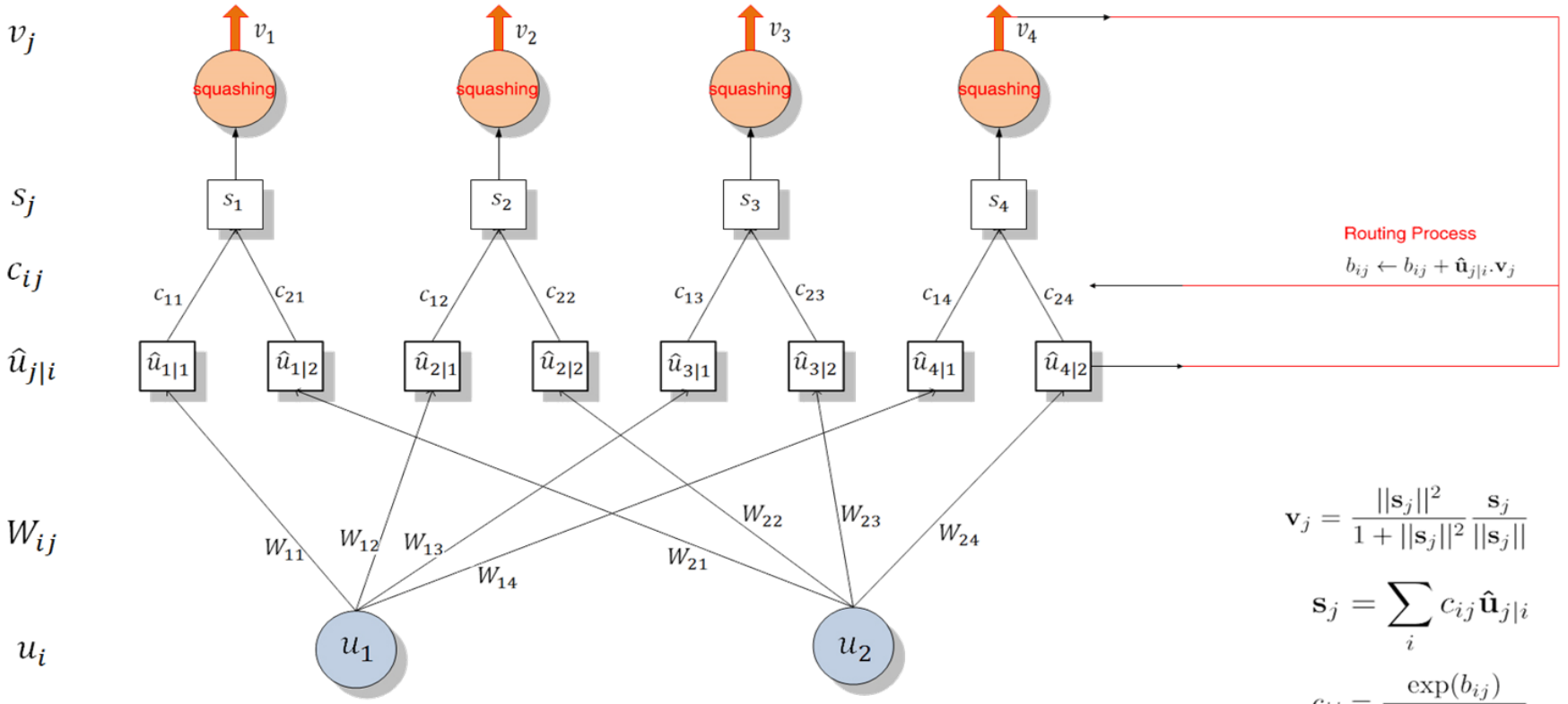
胶囊间的动态路由

- 特别注意的是其整个路由过程不仅仅是存在于胶囊网络的训练过程中，也存在于验证和测试过程，而且需要迭代多次

Procedure 1 Routing algorithm.

1: **procedure** ROUTING($\hat{u}_{j|i}, r, l$)
2: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow 0$.
3: **for** r iterations **do**
4: for all capsule i in layer l: $\mathbf{c}_i \leftarrow \mathtt{softmax}(\mathbf{b}_i)$
5: for all capsule j in layer $(l+1)$: $\mathbf{s}_j \leftarrow \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$
6: for all capsule j in layer $(l+1)$: $\mathbf{v}_j \leftarrow \mathtt{squash}(\mathbf{s}_j)$
7: for all capsule i in layer l and capsule j in layer $(l+1)$: $b_{ij} \leftarrow b_{ij} + \hat{\mathbf{u}}_{j|i}.\mathbf{v}_j$
return $\mathbf{v}_j$

胶囊网络单层结构



$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1+||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

$$\mathbf{s}_j = \sum_i c_{ij} \hat{\mathbf{u}}_{j|i}$$

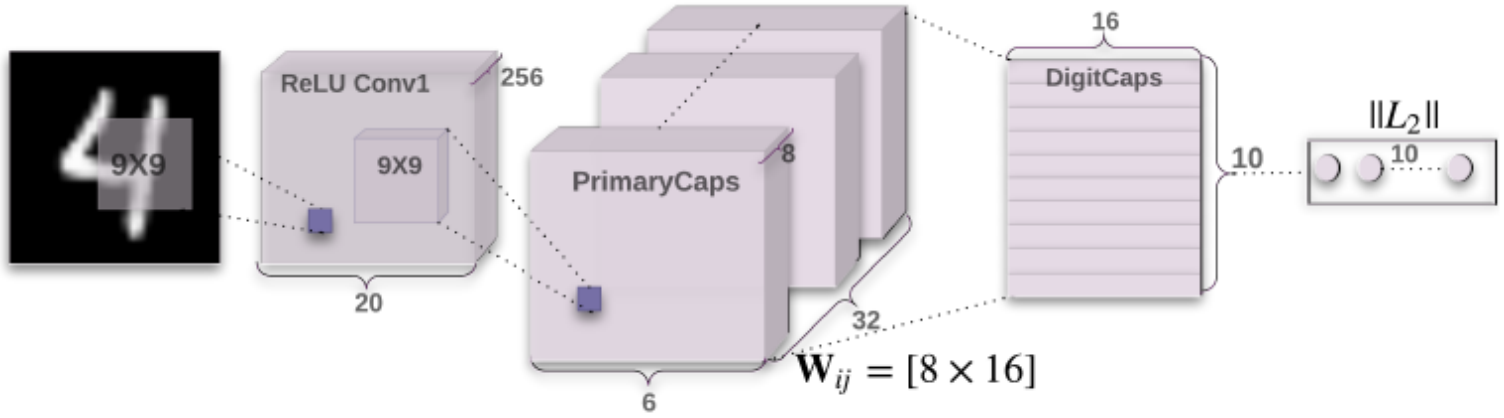
$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}$$

$$\hat{u}_{j|i} = W_{ij} u_i$$

CapsNet

❑ 网络基本结构

- 实现从主胶囊(8D)到数字胶囊(16D)的转换，即：低级特征向量向高级特征向量的转换
- 由于使用特征向量长度来表示对应类别存在概率，所以在最后一层进行分类时，需要将输出的特征向量取L2范数



- 256个步幅为1的9×9的卷积核
- 完成图像信息到低级特征的转换

- 实现低级特征到胶囊多维实体(低级特征向量)的转换
- 具有32个通道的8D胶囊卷积层，每个主胶囊具有8个步幅为2的9×9的卷积核的卷积单元

CapsNet

❑ 边际损失(margin loss)

– 为了实现同时对多个对象的识别，每类目标对象对应的胶囊应分别使用边际损失函数得到类损失L_k，则总边际损失是所有类损失之和

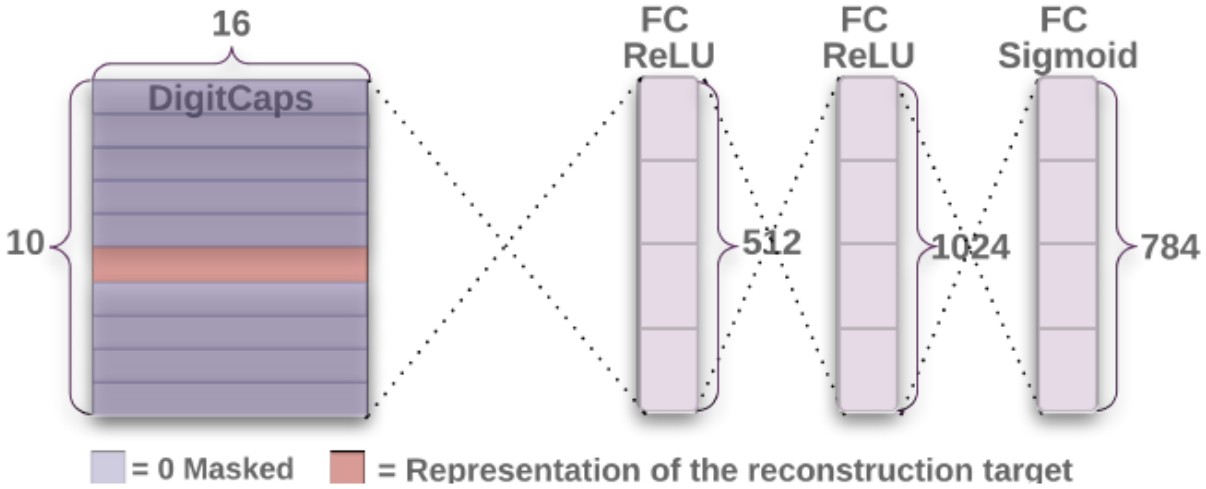
$$L_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k)\max(0, \|v_k\| - m^-)^2$$

- 其中，T_k是表示k类目标对象是否存在，当k类目标对象存在时为1，不存在时为0
- m^+是上界，一般取0.9
- m^-是下界，一般取0.1
- λ为不存在的对象类别的损失的下调权重，避免最开始从不存在分类对应的胶囊输出的特征向量中学习，一般取0.5

CapsNet

❑ 重构正则化

– 鲁棒性强的模型一定具有很强的重构能力。为了使胶囊对输入图像进行编码并输出对应实例化参数，该网络引入了一个重构损失作为正则项。重构损失是逻辑单元的输出和像素强度之间的差的平方和



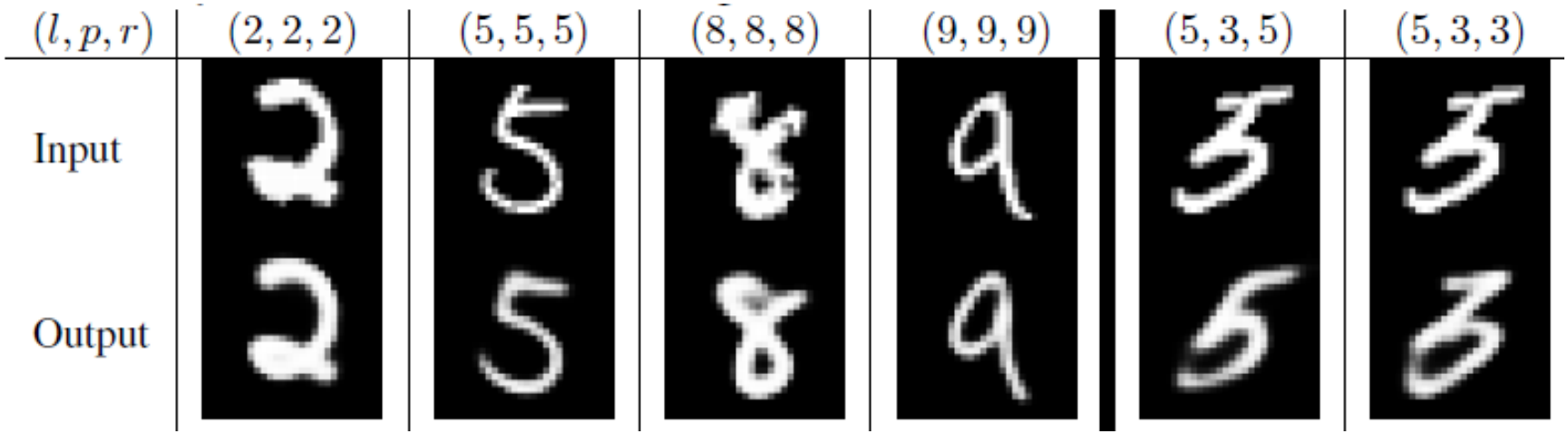
重构网络

- 重构loss的权重因子0.0005

实验结果

❑ 手写数字识别

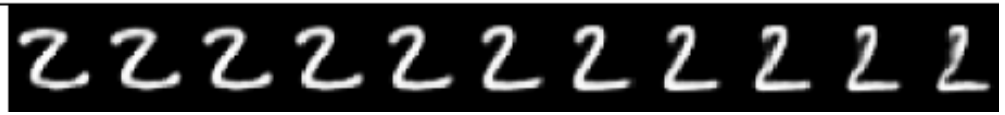
- 使用3次路由迭代
- (l, p, r)代表label的，预测的和重构的
- 最右边是失败的例子



(l, p, r): label, the prediction and reconstruction target

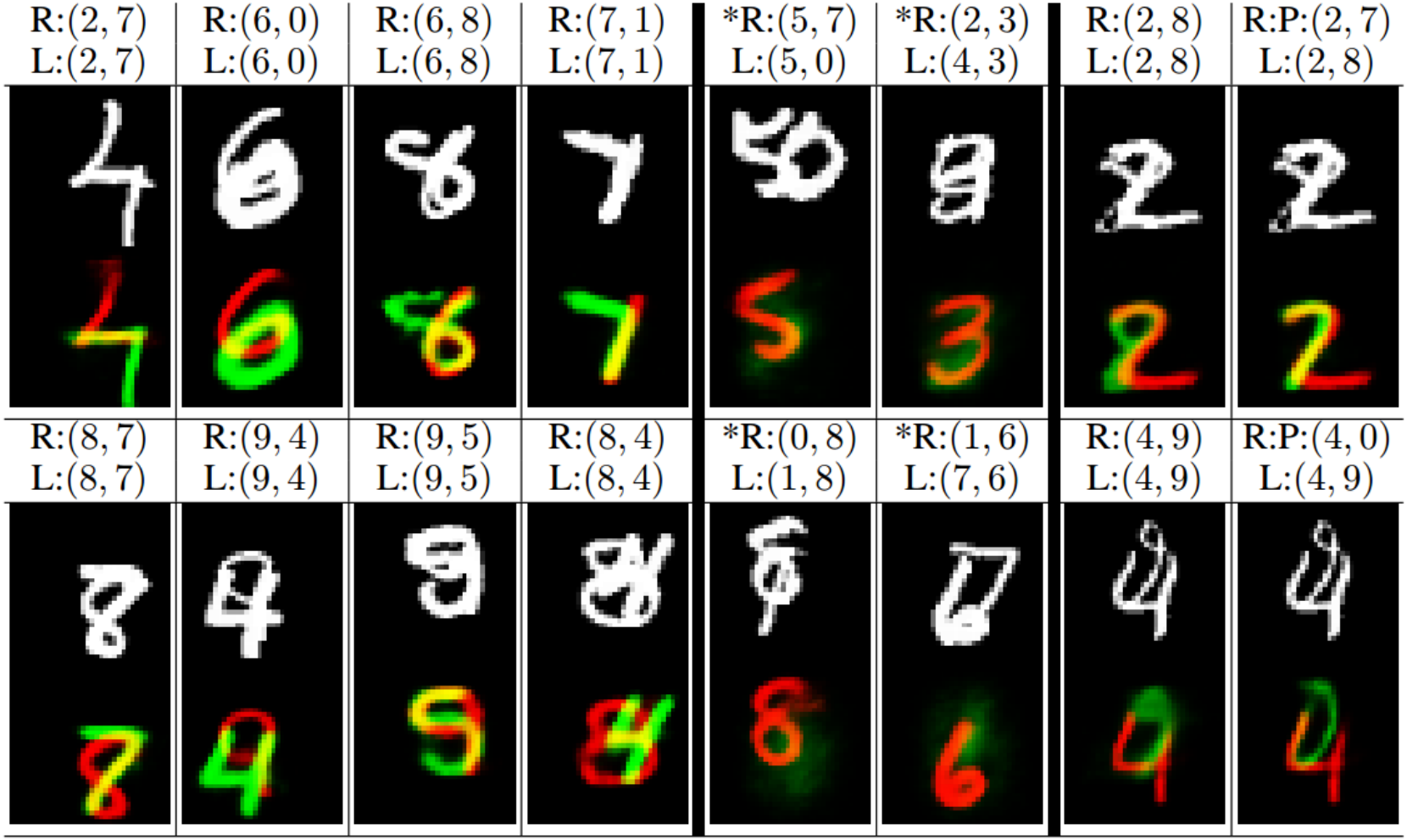
实验结果

- 特征向量的各个维度表示什么
  - 每行展示了16D的DigitCaps中某个维度在区间[-0.25,0.25]间调整0.05

Scale and thickness	
Localized part	
Stroke thickness	
Localized skew	
Width and translation	
Localized part	

实验结果

❑ 重叠手写数字识别



实验结果

❑ 手写数字识别

– test error

Method	Routing	Reconstruction	MNIST (%)	MultiMNIST (%)
Baseline	-	-	0.39	8.1
CapsNet	1	no	$0.34_{\pm 0.032}$	-
CapsNet	1	yes	$0.29_{\pm 0.011}$	7.5
CapsNet	3	no	$0.35_{\pm 0.036}$	-
CapsNet	3	yes	$\mathbf{0.25}_{\pm 0.005}$	**5.2**

谢谢！