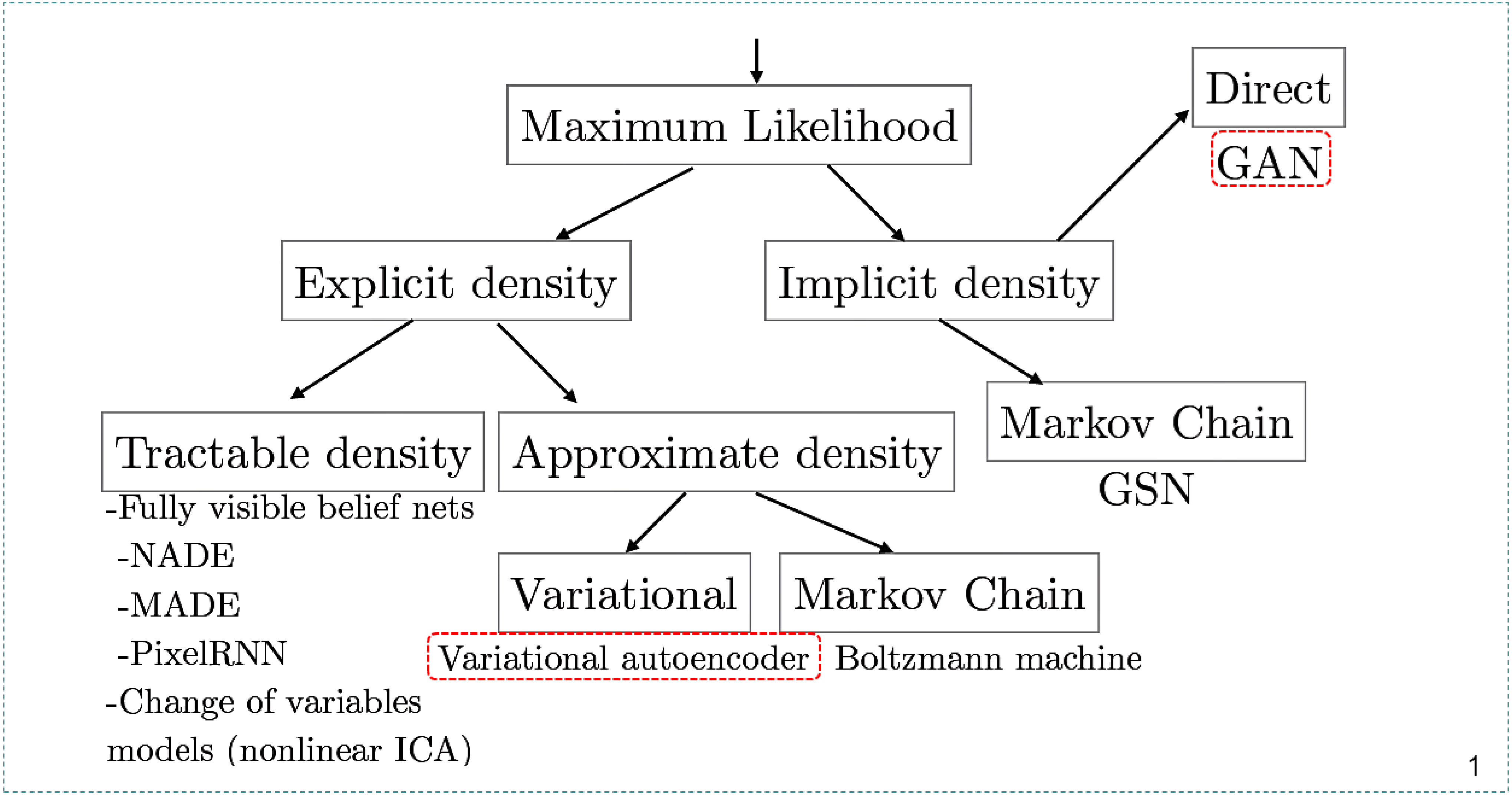
ClusterGAN：生成对抗网络中的潜在空间聚类

作者：凯鲁嘎吉 - 博客园 http://www.cnblogs.com/kailugaji/

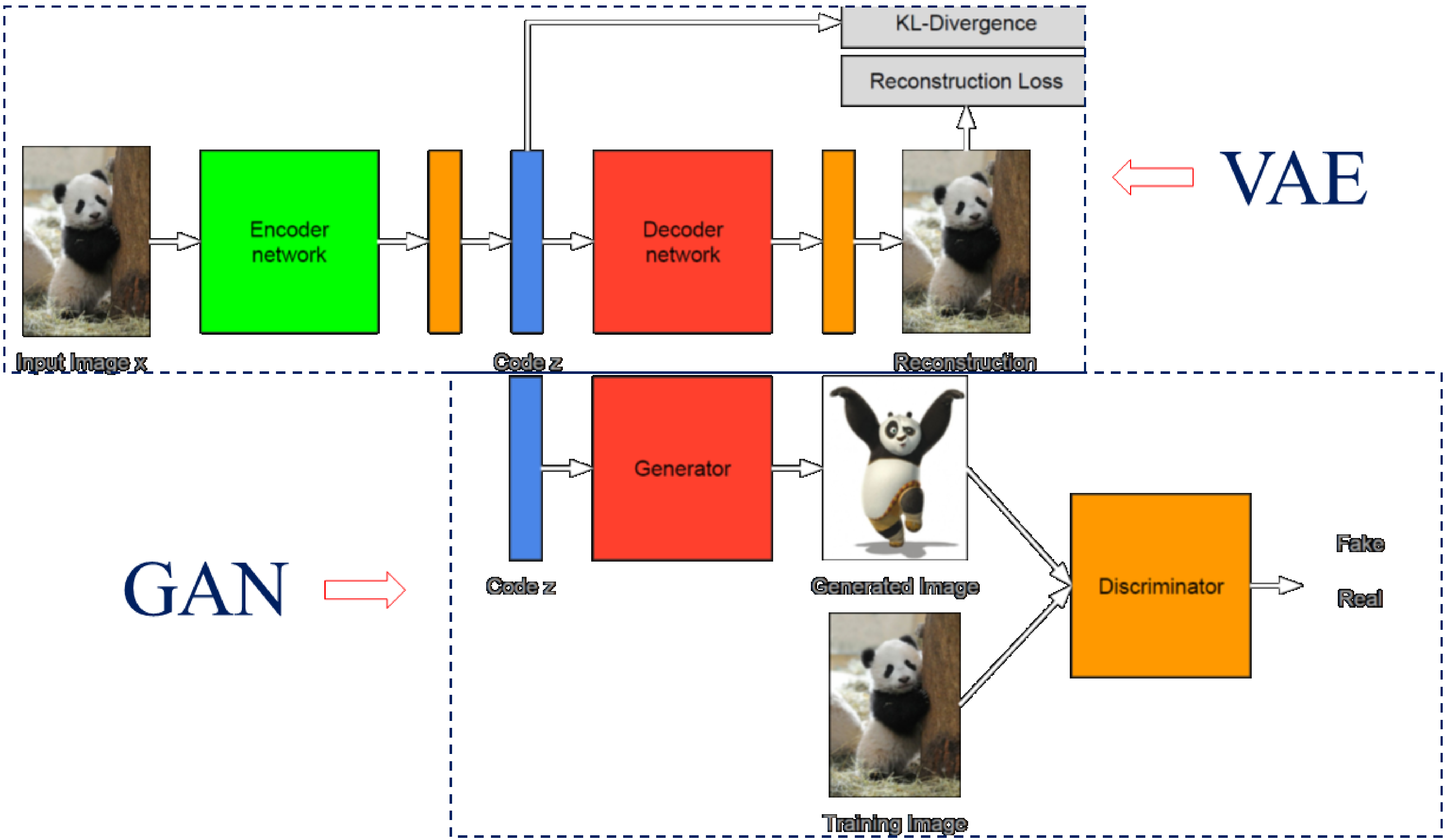
生成对抗网络(GANs)在许多无监督学习任务中取得了显著的成功，毫无疑问，聚类是一个重要的无监督学习问题。虽然可以利用GANs中的潜在空间反向投影进行聚类，但[1]证明了在GAN的潜在空间中并没有保留聚类结构。[1]提出了一种新的基于GANs的聚类机制—ClusterGAN。通过从one-hot离散编码变量和连续编码变量的混合变量中采样潜在变量，再结合通过特定聚类损失训练的反向映射网络(将数据投影到潜在空间)，可以实现潜在空间聚类。

这篇博客首先给出了生成模型的主要分支架构，比较了变分自编码器(VAE)与生成对抗网络(GANs)的区别与优缺点，并简要介绍了GAN的基本原理与训练过程，最后介绍ClusterGAN。比较了GAN与ClusterGAN之间的区别与联系，以及ClusterGAN的主要贡献与思路。

1. 生成模型主要分支


Maximum Likelihood
Explicit density
Implicit density
Direct
GAN
Markov Chain
GSN
Tractable density
Approximate density
-Fully visible belief nets
-NADE
-MADE
-PixelRNN
-Change of variables models (nonlinear ICA)
Variational
Markov Chain
Variational autoencoder
Boltzmann machine
1

2. VAE vs GAN


KL-Divergence
Reconstruction Loss
Encoder network
Decoder network
VAE
Input Image x
Code z
Reconstruction
Generator
Code z
Generated Image
Discriminator
Fake
Real
GAN
Training Image

- Variational Autoencoders (VAE)

  Optimize variational lower bound on likelihood. Useful latent representation, inference queries. But current sample quality not the best.

  Probabilistic spin to traditional autoencoders => allows generating data.

  Defines an intractable density => derive and optimize a (variational) lower bound.

- Pros:
  - Principled approach to generative models
  - Allows inference of q(z|x), can be useful feature representation for other tasks
- Cons:
  - Maximizes lower bound of likelihood: okay, but not as good evaluation as PixelRNN/PixelCNN
  - Samples blurrier and lower quality compared to state-of-the-art (GANs)
- Active areas of research:
  - More flexible approximations, e.g. richer approximate posterior instead of diagonal Gaussian
  - Incorporating structure in latent variables

- Generative Adversarial Nets (GAN)

  Game-theoretic approach, best samples!

  But can be tricky and unstable to train, no inference queries.

  Don't work with an explicit density function.

  Take game-theoretic approach: learn to generate from training distribution through 2-player game.

➢ Pros:
  - Beautiful, state-of-the-art samples!

➢ Cons:
  - Trickier / more unstable to train
  - Can't solve inference queries such as p(x), p(z|x)

➢ Active areas of research:
  - Better loss functions, more stable training (Wasserstein GAN, LSGAN, many others)
  - Conditional GANs, GANs for all kinds of applications

3. GAN

- PixelCNNs define tractable density function, optimize likelihood of training data:

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

- VAEs define intractable density function with latent z:

$$p_\theta(x) = \int p_\theta(z) p_\theta(x \mid z) dz$$

- Cannot optimize directly, derive and optimize lower bound on likelihood instead.

- What if we give up on explicitly modeling density, and just want ability to sample?
- GANs: don't work with any explicit density function!
- Instead, take game-theoretic approach: learn to generate from training distribution through 2-player game

- Minimax objective function

$$\min_{\theta} \max_{\phi} \mathbf{E}_{x\sim p_x^r}\left[\log D(x;\phi)\right] + \mathbf{E}_{z\sim p_z}\left[\log(1-D\left(G(z;\theta);\phi\right))\right]$$

➢ Gradient ascent on discriminator

$$\max_{\phi} \mathbf{E}_{x\sim p_x^r}\left[\log D(x;\phi)\right] + \mathbf{E}_{z\sim p_z}\left[\log(1-D\left(G(z;\theta);\phi\right))\right]$$

➢ Gradient descent on generator

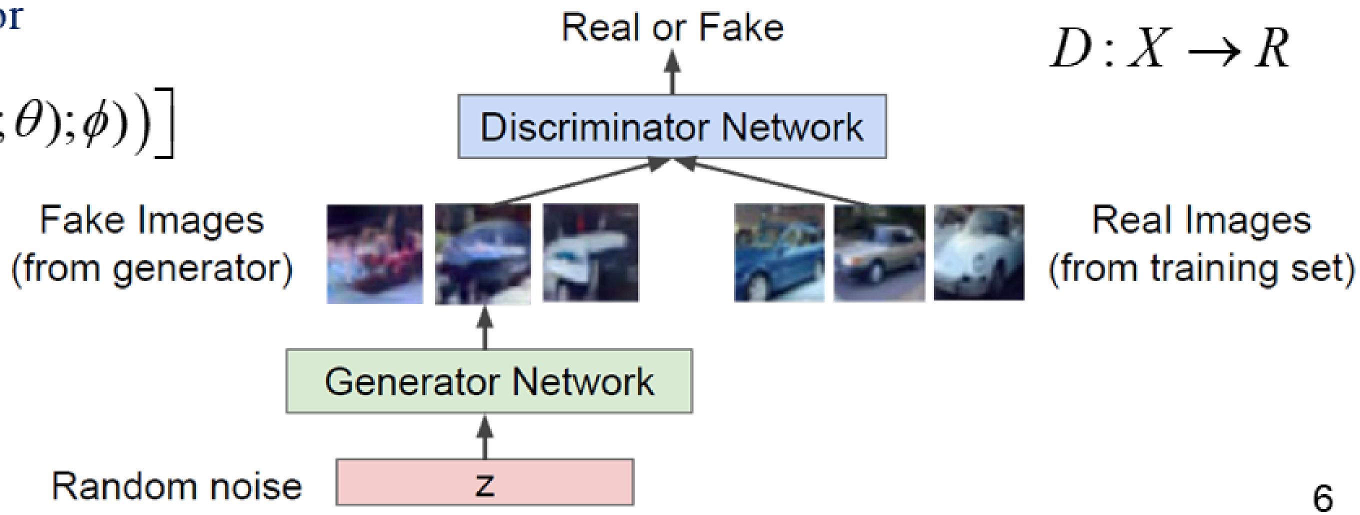
$$\min_{\theta} \mathbf{E}_{z\sim p_z}\left[\log(1-D\left(G(z;\theta);\phi\right))\right]$$

θ是生成器G的

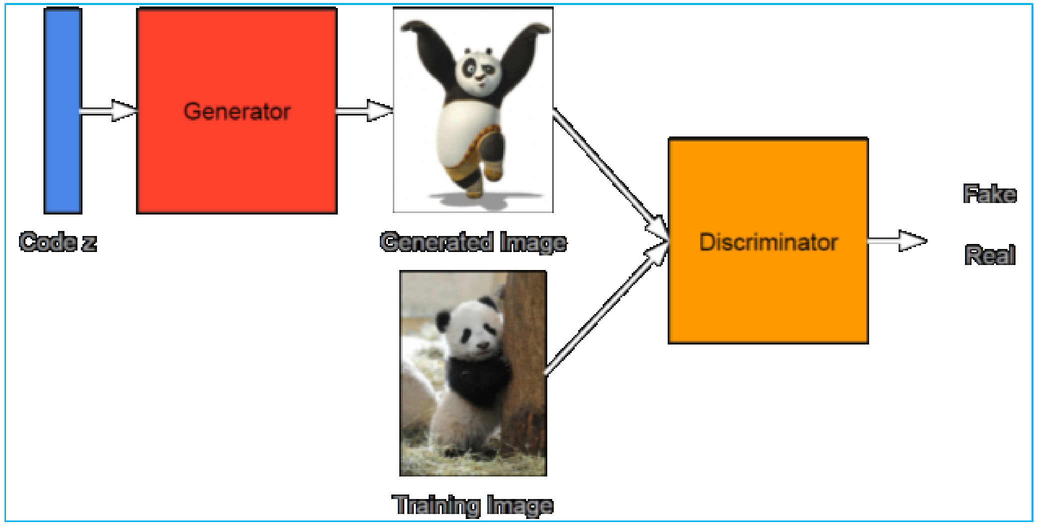
ϕ是判别器D的

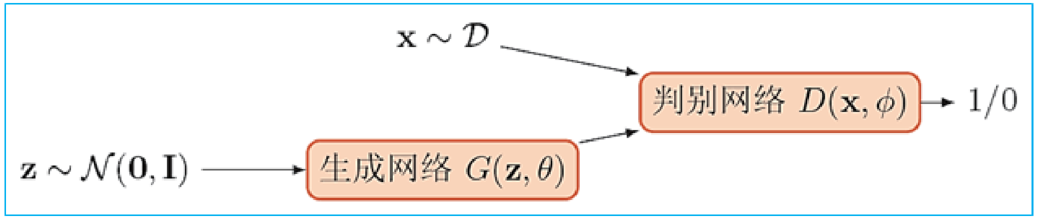
$$G : Z \to X$$

$$D : X \to R$$



- GAN训练算法





输入: 训练集 $\mathcal{D}$，对抗训练迭代次数 T，每次判别网络的训练迭代次数 K，小批量样本数量 M

随机初始化 θ, ϕ;

for $t \leftarrow 1$ **to** T **do**

// 训练判别网络 $D(\mathbf{x}; \phi)$

for $k \leftarrow 1$ **to** K **do**

// 采集小批量训练样本

从训练集 $\mathcal{D}$ 中采集 M 个样本 $\{\mathbf{x}^{(m)}\}, 1 \le m \le M$;

从分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采集 M 个样本 $\{\mathbf{z}^{(m)}\}, 1 \le m \le M$;

使用随机梯度上升更新 ϕ，梯度为

$$\frac{\partial}{\partial \phi}\left[\frac{1}{M}\sum_{m=1}^{M}\left(\log D(\mathbf{x}^{(m)}; \phi) + \log\left(1 - D\left(G(\mathbf{z}^{(m)}; \theta); \phi\right)\right)\right)\right];$$

end

// 训练生成网络 $G(\mathbf{z}; \theta)$

从分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采集 M 个样本 $\{\mathbf{z}^{(m)}\}, 1 \le m \le M$;

使用随机梯度上升更新 θ，梯度为

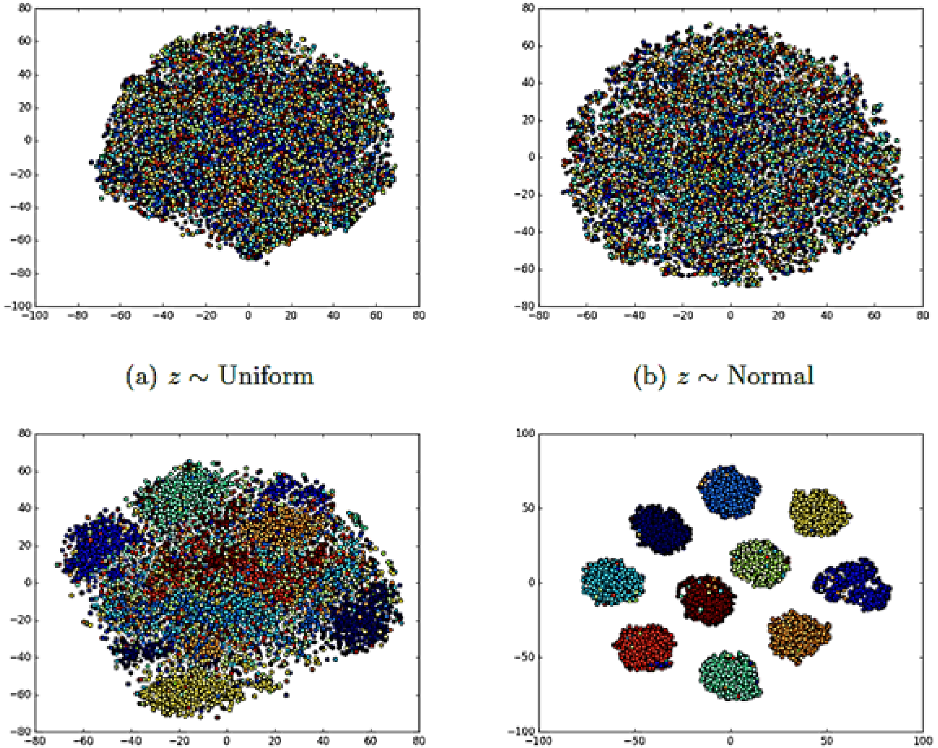
$$\frac{\partial}{\partial \theta}\left[\frac{1}{M}\sum_{m=1}^{M} D\left(G(\mathbf{z}^{(m)}; \theta), \phi\right)\right];$$

end

输出: 生成网络 $G(\mathbf{z}; \theta)$

4. ClusterGAN

- 动机：Vanilla (adj. 普通的，基本的) GAN 在潜在空间不能很好地聚类



(a) $z \sim$ Uniform

(b) $z \sim$ Normal

(c) $z \sim$ Gaussian Mix

(d) $z \sim (z_n, z_c)$

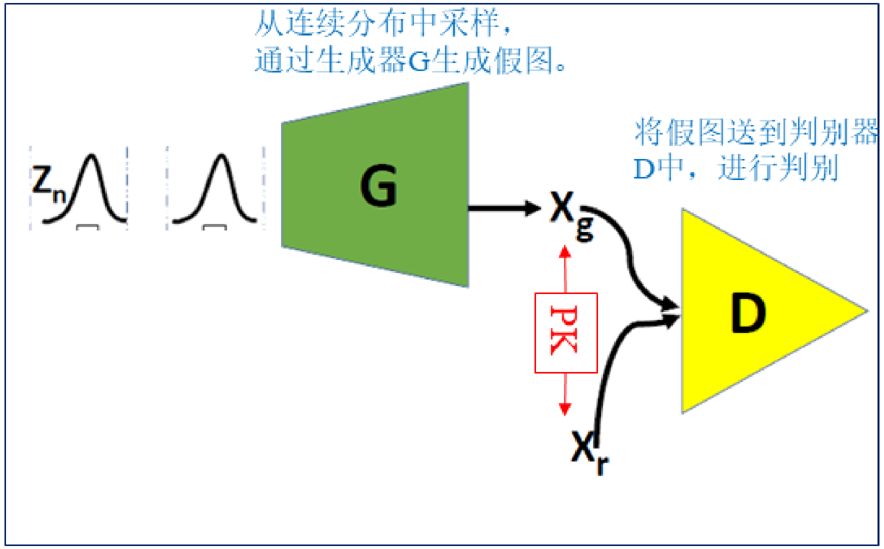
Figure 3: TSNE visualization of latent vectors for GANs trained with different priors on MNIST.

使用GAN进行聚类的一种可能方法是将数据反向传播到潜在空间(使用反向传播解码)并对潜在空间进行聚类。但是，这种方法通常会导致非常糟糕的结果(图3是在MNIST上的聚类结果)。关键原因是，如果反向传播成功，那么反向投影的数据分布应该与潜在空间分布相似，潜在空间分布通常被选择为高斯分布或均匀分布，我们不能期望在该空间中聚类。因此，即使潜在空间可能包含数据的全部信息，潜在空间的距离几何并不反映固有的聚类。在[11]中，作者从一个高斯混合先验采样，即使在有限的数据区间也获得了不同的样本。然而，即使是高斯混合的GANs也无法进行聚类，如图3(c)所示。正如DeLiGAN的作者所观察到的，高斯分量趋向于“拥挤”并且变得多余。利用分类变量提升空间可以有效地解决这一问题。但是潜在空间的连续性传统上被认为是实现良好插值目标的先决条件。换句话说，插值似乎与聚类目标存在冲突。本文证明ClusterGAN是如何同时获得良好的插值和聚类效果的。

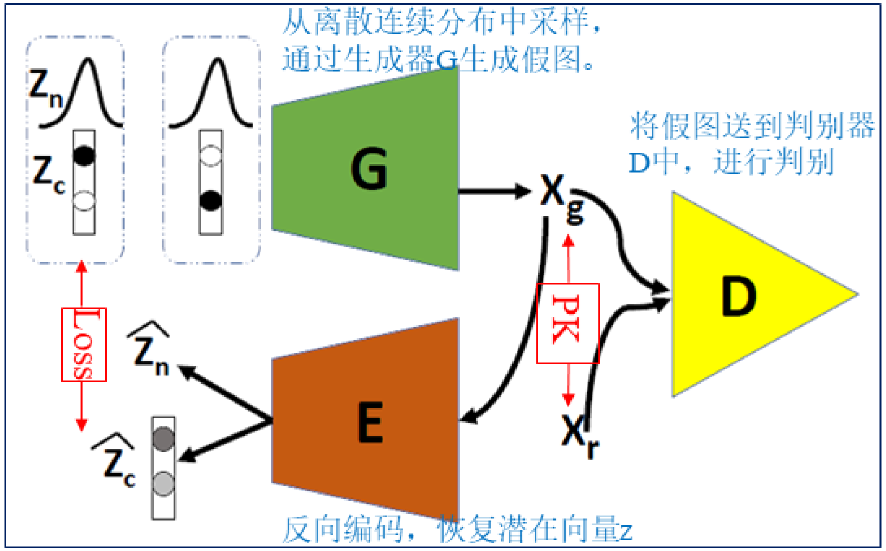
本文主要贡献：

1. 利用一个混合的离散和连续潜在变量(one-hot编码和正态随机变量)，以创建一个非光滑几何潜空间，如图3(d)所示。目的是既保证良好的类间插值(正态随机变量)，又能实现更好地聚类效果(one-hot编码)。
2. 由于问题的非凸性，提出一种新的适应离散连续混合的反向传播算法，以及一个显式反映射网络来获得给定数据点的潜在变量。
3. 本文提出将GAN与带有特定聚类损失函数的反映射网络进行联合训练，以便投影空间中的距离几何反映变量的距离几何。

- GAN vs ClusterGAN



VS



GAN

ClusterGAN

$$\min_{\theta} \max_{\phi} \mathbf{E}_{x\sim p_x^r}\left[\log D(x;\phi)\right] + \mathbf{E}_{z\sim p_z}\left[\log(1-D\left(G(z;\theta);\phi\right))\right]$$

$G: Z \rightarrow X$　　θ是生成器G的

$D: X \rightarrow R$　　ϕ是判别器D的

$E: X \rightarrow Z$　　φ是编码器E的

$$\min_{\theta,\varphi} \max_{\phi} \mathbf{E}_{x\sim p_x^r}\left[\log D(x;\phi)\right] + \mathbf{E}_{z\sim p_z}\left[\log(1-D\left(G(z;\theta);\phi\right))\right]$$

$$\boxed{+\beta_n \mathbf{E}_{z\sim p_z}\left[\left\|z_n - E(G(z_n;\theta);\varphi)\right\|_2^2\right] + \beta_c \mathbf{E}_{z\sim p_z}\left[H(z_c, E(G(z_c;\theta));\varphi)\right]}$$

- ## Modified Backpropagation Based Decoding

$$L(G(z),x)=\|G(z)-x\|_{\boxed{1}}+\lambda\|z_n\|_2$$

目标函数采用一范数，是非凸优化问题，因此之前的反向传播算法不适用。

改进的BP：将采样得到的样本转换为潜在向量(即作为生成器G 的输入)。

以前的工作已探索解决z中的优化问题以恢复潜在向量。但即使反向传播是无损的并且准确地恢复潜在向量，这种方法用传统的潜在先验也不足以进行聚类。为了使情况更糟，传统的BP优化问题在z中定义为非凸的并且可以基于初始化在Z空间中获得不同的嵌入。本文，目标函数恢复隐变量这一项采用一范数。而对于正则化项，由于我们从正态分布中采样，因此采用二范数惩罚正态随机变量。

离散连续混合采样

$z=(z_n,z_c)$

连续：$z_n\sim N(0,\sigma^2\mathbf{I}_{d_n})$

离散：$z_c=e_k,k\sim U\{1,\ldots,K\}$

Input: Real sampler x, Generator function $\mathcal{G}$, Number of Clusters K, Regularization parameter λ, Adam iterations τ
Output: Latent embedding z^*
for $k\in\{1,2,\ldots K\}$ **do**
  Sample $z_n^0\sim\mathcal{N}(0,\sigma^2 I_{d_n})$
  Initialization $z_k^0\leftarrow(z_n^0,e_k)$ (e_k is k^{th} elementary unit vector in K dimensions)
  for $t\in\{1,2,\ldots\tau\}$ **do**
    Obtain the gradient of loss function
    $g\leftarrow\nabla_{z_n}\left(\|\mathcal{G}(z_k^{t-1})-x\|_1+\lambda\|z_n^{t-1}\|_2\right)$
    Update z_n^t using g with Adam iteration to minimize loss.
    Clipping of z_n^t, i.e., $z_n^t\leftarrow\mathcal{P}_{[-0.6,0.6]}(z_n^t)$
    $z_k^t\leftarrow(z_n^t,e_k)$
  end
  Update z^* if z_k^τ has lowest loss obtained so far.
end
return z^*

- Minimax objective function (引入特定的聚类损失项，E用于强制精确恢复潜在向量)

$$\min_{\theta,\varphi}\max_{\phi}\ \mathbf{E}_{x\sim p_x^r}\left[\log D(x;\phi)\right]+\mathbf{E}_{z\sim p_z}\left[\log\left(1-D\left(G(z;\theta);\phi\right)\right)\right]$$
$$\boxed{+\beta_n\mathbf{E}_{z\sim p_z}\left[\left\|z_n-E(G(z_n;\theta);\varphi)\right\|_2^2\right]+\beta_c\mathbf{E}_{z\sim p_z}\left[H(z_c,E(G(z_c;\theta));\varphi)\right]}$$

其中，βn 和 βc 的相对大小使得能够灵活地选择以改变保留潜在编码的离散和连续部分的重要性。E(G(z))相当于聚类中心。

θ是生成器G的

ϕ是判别器D的

φ是编码器E的

$G: Z\to X$

$D: X\to R$

$E: X\to Z$

$H(.,.)$：交叉熵

离散连续混合采样

$z=(z_n,z_c)$

连续：$z_n\sim N(0,\sigma^2\mathbf{I}_{d_n})$

离散：$z_c=e_k, k\sim U\{1,\ldots,K\}$

➢ 参数更新算法

Input: Functions $\mathcal{G}$, $\mathcal{D}$ and $\mathcal{E}$, Regularization parameters β_n, β_c, learning rate η, parameters Θ_G^t, Θ_E^t

Output: $\Theta_G^{(t+1)}, \Theta_E^{(t+1)}$

Sample $z^{(i)}{}_{i=1}^{m}$ from $\mathbb{P}^z, z=(z_n,z_c)$

$g_{\Theta_G}\leftarrow$

$$\nabla_{\Theta_G}\left(-\sum_{i=1}^{m}q(\mathcal{D}(\mathcal{G}(z^{(i)}))+\beta_n\sum_{i=1}^{m}\|z_n^{(i)}-\mathcal{E}(\mathcal{G}(z_n^{(i)}))\|_2^2+\beta_c\sum_{i=1}^{m}\mathcal{H}(z_c^{(i)},\mathcal{E}(\mathcal{G}(z_c^{(i)})))\right)$$

$$g_{\Theta_E}\leftarrow\nabla_{\Theta_E}\left(\beta_n\sum_{i=1}^{m}\|z_n^{(i)}-\mathcal{E}(\mathcal{G}(z_n^{(i)}))\|_2^2+\beta_c\sum_{i=1}^{m}\mathcal{H}(z_c^{(i)},\mathcal{E}(\mathcal{G}(z_c^{(i)})))\right)$$

Update Θ_G using $(g_{\Theta_G},\Theta_G^t)$ with Adam ; similarly for Θ_E.

return Θ_G, Θ_E

11

5. 参考文献

[1] Mukherjee S , Asnani H , Lin E , et al. ClusterGAN: Latent Space Clustering in Generative Adversarial Networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33:4610-4617.

[2] ClusterGAN代码 --GitHub

[3] Lecture 13: Generative Models --Fei-Fei Li

[4] Deep Generative Models --Shenlong Wang

[5] 变分推断与变分自编码器 - 凯鲁嘎吉 - 博客园

[6] 邱锡鹏, 神经网络与深度学习[M]. 2020.