生成模型

Sisyphes

2020 年 9 月 23 日

目录

第一节　GAN 的基本认识

基本理论，结构，问题，trick，评估方式。

1 Generative Adversarial Networks(2014.7)

2 Wasserstein gan(2017.2)

3 Spectral Norm Regularization for Improving the Generalizability of Deep Learning(2017.5)

4 Spectral Normalization for Generative Adversarial Networks(2018.2)

5 The relativistic discriminator: a key element missing from standard GAN(2018.9)

6 GANs for Medical Image Analysis(2018.9)

7 Auto-Encoding Variational Bayes(2014.5)

8 GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium(2017.6)

9 A Second-order Equilibrium in Nonconvex-Nonconcave Min-max Optimization: Existence and Algorithm(2020.6)

10 ...

1.1　什么是生成模型

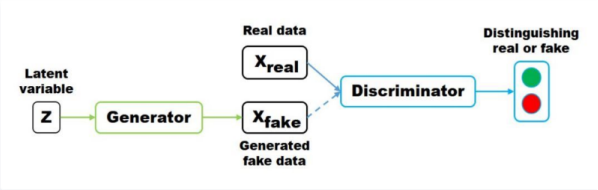

图 1: Gan Pipeline

$$\min_G \max_D V(G,D) = \min_G \max_D \mathbb{E}_{x\sim p_{\text{data}}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1-D(G(z))]] \tag{1}$$

1.2 问题及一些解决方法

在WGAN及前作中，作者仔细分析了 GAN 的问题，中文可参考令人判案叫绝的 WGAN。这里简要说明一下：

公式 (1) 中，当判别器 D 最优时 $D^*(x) = \frac{P_r(x)}{P_r(x)+P_g(x)}$，将其带入 (1) 式，因 $JS(P||Q) = \frac{1}{2}(KL(P||\frac{P+Q}{2}) + KL(Q||\frac{P+Q}{2}))$ 得到生成器损失为:$2JS(p_r||p_g) - 2log2$。直观上说，生成样本是从低微随机变量映射到高维图像的，其生成的分布与真实的高维图像分布的交集很小，这会导致 JS 散度为常数 $log2$ ，其梯度为 0，因此生成器得不到有效优化。换一种说法就是，" 判别器越好，生成器梯度消失越严重"。WGAN 给出的解决办法就是将 JS(KL) 替换为连续度量Wasserstein metric ，也即

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

其中 $\Pi(P_r, P_g)$ 表不 P_r, P_g 的联合分布 $\gamma(x, y)$ 集。但是上式在实际情况中难以操作，至少联合分布采样就无法完成，然而根据 Kantorovich-Rubinstein duality (证明细节可参考Monge-Kantorovich Transportation Problem) 此距离可改写成如下形式:

$$W(P_r, P_g) = \sup_{\|f\|_L \leq K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

其中 $||\ ||_L \leq K$ 表示满足 K-Lipschitz 条件的函数。

于是得到 WGAN 的优化目标：

$$\min_G \max_{D_f} V(D_f, G) = \min_{P_g} \sup_{\|f\|_L \leq K} \mathbb{E}_{x\sim P_r}[f(x)] - \mathbb{E}_{x\sim P_g}[f(x)]$$

理论上是解决了原始 GAN 的生成器难训练问题，因为此时，即使生成样本和原始样本分布没有重叠，距离仍然是光滑的，光滑即可导，可导即可优化。而上面的第二点，对判别器的 Lipschitz 限制，在实际操作中有三种方法。第一简单粗暴的 weight clip，将迭代后的权重 clip 到一个固定的范围，比如 (-0.01, 0.01)，实验表明，此种方式会倾向导致判别器的权重集中在 clip 所给定的两端。第二，利普希茨条件是对函数变化幅度的限制，因此可以给梯度加一个惩罚项 (正则化)，于是有了 gradient penalty 技巧，此时判别器的损失函数变为

$$L(D) = -\mathbb{E}_{x\sim P_r}[D(x)] + \mathbb{E}_{x\sim P_g}[D(x)] + \lambda \mathbb{E}_{x\sim \mathcal{P}_{\hat{x}}}\left[\|\nabla_x D(x)\|_p - 1\right]^2$$

其中 $\hat{x} = \epsilon x_r + (1-\epsilon)x_g, \epsilon \sim Uniforl[0,1]$。第三，一二对判别器的利普希茨限制的求解较为粗糙，文章 [4] 公式 6,7,8 表明，对要限制每一层的权重的谱范数为 1，即可使得最终的判别器满足利普希茨约束。范数和模型的泛化能力有很大关系，这点在基础这一节中会详细展开。

spectral-normal

经过一些数学推导之后，WGAN 的工程改进为如下几点：

1 判别器最后一层去掉 sigmoid

2 生成器和判别器的 loss 不取 log

3 每次更新判别器的参数之后把它们的绝对值截断到不超过一个固定常数 c

4 弃用基于动量的优化算法（包括 momentum 和 Adam），推荐 RMSProp，SGD

5 给判别器梯度加正则项 (3 的又一改进)

6 对判别器的每一层权重做谱归一化映射 (3,5 的改进)

除了 WGAN，GAN 本身的问题分析以及改进点都有很文章，比如:

a RGAN[5]，认为在优化判别器的同时应该降低其对真实样本的概率值，并给出了新的相对判别器

b 和 VAE 的结合用，生成网络损失退化到某些特征层的平方差损失和，用来提升图片的生成质量

c 把标签 1 替换为 0.8 1.0 之间的随机数，减缓判别器预测出来的 confidence 倾向于更高值的问题

d PatchGAN：于对图像的每一个小 Patch 进行判别，以生成更加锐利清晰的边缘

e PgGAN：在尺度上由低到高逐级生成，能极大提高生成图像的分辨率

补一个改进论文图表：

Table 1: An overview of GANs discussed in Section 2 and 3.

Subject	Topic	Reference
Object functions	f-divergence	GAN [36], f-GAN [89], LSGAN [76]
	IPM	WGAN [5], WGAN-GP [42], FISHER GAN [84], McGAN [85], MMDGAN [68]
Architecture	DCGAN	DCGAN [100]
	Hierarchy	StackedGAN [49], GoGAN [54], Progressive GAN [56]
	Auto encoder	BEGAN [10], EBGAN [143], MAGAN [128]
Issues	Theoretical analysis	Towards principled methods for training GANs [4]
		Generalization and equilibrium in GAN [6]
	Mode collapse	MRGAN [13], DRAGAN [61], MAD-GAN [33], Unrolled GAN [79]
Latent space	Decomposition	CGAN [80], ACGAN [90], InfoGAN [15], ss-InfoGAN [116]
	Encoder	ALI [26], BiGAN [24], Adversarial Generator-Encoder Networks [123]
	VAE	VAEGAN [64], α-GAN [102]

图 2: Gan Improve

就个人而言，我比较关心下面四个问题：

- 如何提高生成图像的质量？见2.2
- 如何控制生成图像的构成元素？见2.3
- 有什么有趣的应用，如何搭建有趣的应用？见2.1
- 生成网络的可能理论是什么？见五

1.3　评价指标

如何评价模型生成的准确性和非单一性？答案是可以辅助一个分类网络作为基准，用概率分布来刻画。

对于给定的数据集，训练一个分类模型，比如 Inception network，训练一个生成模型，将生成的样本用分类网络判断，直观上来说，如果分类模型非常好，则

准确性：同一类别的输出概率分布应该趋向于脉冲分布。

非单一性：所有类别的输出概率趋于均匀分布。

以上两点，Inception score 有如下刻画：

$$IS\left(P_g\right)=e^{E_{x\sim P_g}[KL(p_M(y|x)||p_M(y))]} \tag{2}$$

其中 M 表示分类模型。

因此，若 GAN 训练良好，则 $p_M(y\mid x)$ 趋近于脉冲分布，$p_M(y)$ 趋近于均匀分布。二者 KL 散度会很大，Inception Score 会比较高。实际实验表明，Inception Score 和人的主观判别趋向一致。

可阅读Inception score。

IS 的改进：Fréchet Inception Distance (FID)

FID 距离计算真实样本，生成样本在特征空间之间的距离。首先利用分类网络 (Inception) 来提取特征，然后各自计算在特征空间的均值和协方差 (详细分析略)。最后用如下公式计算：

$$FID\left(\mathbb{P}_r,\mathbb{P}_g\right)=||\mu_r-\mu_g||+\operatorname{Tr}\left(C_r+C_g-2\left(C_rC_g\right)^{1/2}\right) \tag{3}$$

μ, C, 分别代表特征的均值和协方差。

可阅读FID。

如何评价生成模图像关于生成向量的渐变性？直观上说，给定两个端点图像的特征向量 A,B，若隐空间较平滑，则在 A 到 B 中间选两点 $t,t+\epsilon$，其生成图像的差异值会比较小，遍历所有情况，数学语言描述即为：

$$\begin{aligned} l_{\mathcal{W}}=\mathbb{E}\left[\tfrac{1}{\epsilon^2}d\left(g\left(\operatorname{lerp}\left(f\left(\mathbf{z}_1\right),f\left(\mathbf{z}_2\right);t\right)\right)\right.\right. \\ \left.\left.g\left(\operatorname{lerp}\left(f\left(\mathbf{z}_1\right),f\left(\mathbf{z}_2\right);t+\epsilon\right)\right)\right)\right] \end{aligned}$$

其中 lerp 为在隐空间 $\mathcal{W}$ 中的线性插值，$t\in\{0,1\},\epsilon$ 是微变量，表示从 $A\to B$ 的步长，g 生成模型，d 欧式距离，f 是原始向量 z 的映射函数。该

评价指标名为 Perceptual Path Length (PPL)，准确的说是刻画了隐空间的感知平滑性。。

如何刻画隐空间的线性可分性？和评价丰富性类似，借助于分类模型。首先训练一个给定数据集的属性分类模型，其次在隐空间中生成大量样本，比如 200000，然后用分类模型对其分类，筛选出分类分数较好的图像，同时包含对应的特征向量，用 SVM 对特征向量进行分类，每个属性训一个 SVM，然后计算 SVM 预测的标签和分类模型的标签的条件熵 $H(Y|X)$，X 为 SVM 预测的标签，Y 为分类模型预测的标签，最后关于属性集合求和，取指数。

第二节 Cycle, Style, SeFa

2.1 CycleGAN

18 年的CycleGAN的前身是 pix2pix，因成对数据集的制作成本太高，于是 Jun-Yan Zhu 等人将图像翻译从成对图像扩展到两个图像域的变换, 完成了图域 X 到图域 Y 的一一映射 (数据集数量相等时)。文中作者抓住了 GAN 的生成核心，对抗损失。围绕这个建模，做实验，加一些改进，就做到了很漂亮的结果。

首先数学符号化，给定两个数据集，X, Y 希望找到映射 $G : X \to Y$，使得从 G 映射的元素其分布同 Y 一致，对应的 $F : Y \to X$，F 映射的分布同 X 一致。

如何达到这个目的？回想 GAN 从某一分布生成真实样本的关键在于对抗损失，从给定分布采样送入生成器生成的图像和真实图像经过判别器和生成器的对抗，达到目的。这里也如此，给 G 一个对抗，判别器为 D_Y，给 F 一个对抗，判别器为 D_X。沿用原始 GAN 的对抗损失，可写出 G 的对抗损失：

$$\begin{aligned}\mathcal{L}_{\mathrm{GAN}}\left(G, D_Y, X, Y\right) = & \mathbb{E}_{y \sim p_{\mathrm{data}}(y)}\left[\log D_Y(y)\right] \\ & + \mathbb{E}_{x \sim p_{\mathrm{data}}(x)}\left[\log\left(1 - D_Y(G(x))\right)\right],\end{aligned}$$

F 同理，优化形式即为：$\min_F \max_{D_X} \mathcal{L}_{\mathrm{GAN}}\left(F, D_X, Y, X\right)$。

因 X 到 Y 有 $|Y|^{|X|}$ 多个映射满足以上情况，会导致结果的随机化，需要增加一些限制，问题是如何保证对固定的 x_i，映射到 y_i？若 $x \to G(x) \to$

$F(G(x)) \approx x$ 同时 $y \to F(y) \to G(F(y)) \approx y$ 则容易看出满足条件。于是得到 cycle consistency loss

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) = &\mathbb{E}_{x \sim p_{\text{data}}(x)}\left[\|F(G(x)) - x\|_1\right] \\ &+ \mathbb{E}_{y \sim p_{\text{data}}(y)}\left[\|G(F(y)) - y\|_1\right]\end{aligned}$$

最终的损失函数为：

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{GAN}(G, D_Y, X, Y) \quad + \mathcal{L}_{GAN}(F, D_X, Y, X) + \lambda \mathcal{L}_{cyc}(G, F)$$

优化目标为：

$$G^*, F^* = \arg\min_{G,F} \max_{D_x, D_Y} \mathcal{L}\left(G, F, D_X, D_Y\right).$$

剩下的代码实现细节，一些改动，各种实验等，跳到讲代码环节。

2.2 StyleGAN

18 年的stylegan结合了多方技术，从生成器着手，实现了 1024×1024 高清图生成。

回忆AdaIN：

$$\text{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

x 表示归一化的内容图，y 表示风格统计量，平均值和方差。在 stylegan 中，y 表示风格隐空间 $\mathcal{W}$ 中向量的风格参数。

1 以 Progressive Growing Network 为基础，提出 Network Architecture of StyleGAN

2 2.1 生成向量从随机 z 替换为固定常量 $4 \times 4 \times 512$ 大小的 tensor

2.2 随机向量 z 被 8 个叠加的全连接层 (含非线性激活) 映射到风格空间 $\mathcal{W}$

2.3 不同层级加入了噪声输入

2.4 上采样线性插值改为双线性插值

3 AdaIN：风格参数 $y_{s,i}, y_{b,i}$ 替换了经典风格转换网络的均值和方差。

4 Mixing Regularization：风格向量输入两个 $w_1 = f(z_1), w_2 = f(z_2)$，$w_1$ 生成 4×4，w_2 生成 8×8，以此混合两种图像风格。

设输入风格向量 w_1, w_2，w_2 在不同分辨率上和 w_1 混合，stylegan 团队将 w_2 生成图分为三个区间，低分辨率 $(4^2 - 8^2)$，中等分辨率 $(16^2 - 32^2)$，高分辨率 $(64^2 - 1024^2)$，他们发现低分辨率控制 (向 w_2 属性靠齐) 了脸型，肤色，性别，而高分辨率则影响背景，发色等。

stylegan 存在 water droplet-like artifacts 问题，2020 年同样的团队，据此对 stylegan 做了深入分析，提出了升级版 stylegan2。解决了伪影问题，同时提高了各方面的性能。

1 观察发现生成图有类水滴伪影，去掉 AdaIN 则没有，然后仔细分析，提出了用估计统计量对权重进行归一化替换了 AdaIN 以实际数据的统计量来归一化的方式

  1.1 AdaIN 可分解成两部分，输入的归一化 (Norm)，归一化后的线性变换 (Mod),实验发现去掉均值不影响效果,于是将 Norm+Mod+Conv 改为 Mod(std)+Conv+Norm(std)，并将噪声改到这之后。

  1.2 改过之后的模块可以被写成 Mod+Demod+Conv 形式,取名 Weight demodulation，见原始论文 figure2。

  1.3 实际操作会采用 grouped convolution，因为每个 bactch 的权重都会不同，将 N 个 samples 一个 group 变为一个 sample N 个 groups

2 同观察发现随着面部角度的变化，牙齿排列没发生改变，然后仔细分析，发现 progressive growing 的问题，提出了 a hierarchical Generator with skip connection(类似 MSG-GAN)

  2.1 以 MSG-GAN 设计为基础，在 skips 和 residual 两种连接方式上组合实验，选择了最优结构

  2.2 输入由 RGB 变为 tRGB 和 fRGB

3 将 Perceptual Path Length 以正则化的形式融合到生成模型中，从而从优化上提高了隐空间的 perceptual 光滑性，同时也提高了生成图像的质量。

4 Lazy Regularization：分离主函数和regularization terms，每 16 个 mini-batches 才计算一次正则项，能减少内存，运算量，加速训练速度。

感知正则化表达式：

$$\mathbb{E}_{\mathbf{w},\mathbf{y}\sim\mathcal{N}(0,\mathbf{I})}\left(\left\|\mathbf{J}_{\mathbf{w}}^{T}\mathbf{y}\right\|_{2}-a\right)^{2}$$
$$\mathbf{J}_{\mathbf{w}}=\partial g(\mathbf{w})/\partial\mathbf{w}$$

其中常量 a 在学习过程中进行移动平均变化。

新的生成模型表明，提高生成器复杂度，不需要 progressive growing 形式的生成模型，也能生成高清图。而渐进增加各自独立的生成器 (progressive growing)，会使得生成器倾向生成固定的高频特征。另一方面，新的生成器也具有从地分辨率到高分辨率逐渐生成的特点。

stylegan2 参考阅读

细节展开和代码实现，视频传达。

题外话：训练 StyleGAN2 是奢侈的，原作者使用 8 GPUs (V100), 在 FFHQ 数据集上花费了 9 天，LSUN CAR 数据集上花费了 13 天。不过利用其核心做一些改造，还是可以单 GPU 训练出一些有趣的模型。

2.2.1 StyleGAN 的改造

这一小节参考了seeprettyface，GAN Face portrait。

StyleGAN(1,2) 能生成高清图像，如何利用其基本组件，做一些有趣的事呢？比如给人戴眼镜，换性别，生成某种特定类型的图像，给人设计衣着搭配等。

可以利用的元素有强大的生成器，风格迁移 AdaIN，以及带监督的 InfoGAN 等思想。要实现对固定属性改变，利用监督信息是个思路，虽然 StyleGAN 表明不同尺度 (级) 控制了不同属性，但各属性并不能较好的线性分离，可以做，但需要调节的地方会比较多。

InfoGAN 的思想是将类别信息插入编码向量中，对生成图像做一个多分类器，从而实现了控制编码的类别信息，达到控制生成图的样式。实际操作可以将判别器和分类器共享一部分网络，达到减参效果。但是这样也有弊端，得对数据进行人为划分，且划分的数据类往往不是细粒度分类，差异相对较大 (不大效果一般不好)。

如何实现对现实人脸的各种基本属性的渐变控制？比如表情，眼镜，角度，年龄，脸型等。

渐变控制也即连续控制，因此人脸的表示向量是有必要实现的，有了表示向量，可以求解出表示向量和属性变化的方向向量，于是就沿着这个方向变化表示向量，就能生成渐变属性图了。

表示向量编码采用 VAE 思想，写一个与 StyleGAN 的生成器对应的编码器即可。而人脸对应的基本特征信息，可以利用现成的 API 获取这些标签 (这里思路来自 seeprettyface)。在同类标签中，以其中位数为基准，低于该阈值的设为 **0**，高于该阈值的标签设为 **1**，然后求解一个关于标签中位数的分割面，以此为方向向量，可以实现效果：

$$\hat{e} = \vec{w} = \text{argmax}_w \, P(\vec{w} \cdot x + \vec{b} = y), y \in \{0, 1\}.$$

其中 x 表示给定数据的表示向量，在 StyleGAN 中，其维度为 $(18, 512)$。

2.3 SeFa

选SeFa这篇文章，主要因为它简单，对入门者比较友好。

文章关心的是隐空间的非监督解耦，有监督解耦需要数据的监督信息 (比如 infogan)，一方面，训出的模型依赖于数据和人给定的标签信息，这往往导致形成的表示向量不唯一，另一方面，不能有效的迁移到未知属性上。

考察到当前生成网络从随机向量到隐向量都是采用一个全连接映射而得，若假设隐向量变动幅度大，生成的图变动幅度也大。那么隐空间的坐标方向向量，就可以充当控制生成图的属性角色。

符号化：

令隐空间 $\mathcal{Z} \subseteq \mathbb{R}^d$，生成图空间为 $\mathcal{I} \subseteq \mathbb{R}^{H \times W \times C}$，全连接到生成图可表示为：$\mathbf{I} = G'(F_C(\mathbf{z})) \triangleq G'(\mathbf{y})$。对于固定的生成网络 G，因假设 $\mathbf{z}$ 的大改变 $(\mathbf{z}' = \mathbf{z} + \alpha\mathbf{n})$，对 $\mathbf{I}$ 影响较大，且 $\mathbf{y} = FC(\mathbf{z}) = \mathbf{A}\mathbf{z} + \mathbf{b}$，于是求解影响 $\Delta\mathbf{y} = FC(\mathbf{z}') - FC(\mathbf{z}) = (\mathbf{A}(\mathbf{z} + \alpha\mathbf{n}) + \mathbf{b}) - (\mathbf{A}\mathbf{z} + \mathbf{b}) = \alpha\mathbf{A}\mathbf{n}$ 变化较大的 k 个方向向量 $\mathbf{n}$，等价于求解：

$$\mathbf{N}^* = \underset{\left\{\mathbf{N} \in \mathbb{R}^{d \times k} : \mathbf{n}_i^T \mathbf{n}_i = 1 \forall i = 1, \cdots, k\right\}}{\text{argmax}} \sum_{i=1}^{k} \|\mathbf{A}\mathbf{n}_i\|_2^2$$

这里 $\mathbf{n_i}$ 各不相同。

将方向向量限制单位模长上，由拉格朗日法，

$$\begin{aligned}\mathbf{N}^* &= \arg\max_{\mathbf{N}\in\mathbb{R}^{d\times k}} \sum_{i=1}^{k} \|\mathbf{A}\mathbf{n}_i\|_2^2 - \sum_{i=1}^{k} \lambda_i \left(\mathbf{n}_i^T \mathbf{n}_i - 1\right) \\ &= \arg\max_{\mathbf{N}\in\mathbb{R}^{d\times k}} \sum_{i=1}^{k} \left(\mathbf{n}_i^T \mathbf{A}^T \mathbf{A}\mathbf{n}_i - \lambda_i \mathbf{n}_i^T \mathbf{n}_i + \lambda_i\right)\end{aligned}$$

可求得，由全连接矩阵 $\mathbf{A}$ 组成的半正定矩阵 $\mathbf{A}^T\mathbf{A}$ 的特征向量控制了生成图的属性。

第三节　GAN 的应用

Table 2: Categorization of GANs applied for various topics.

Domain	Topic	Reference
Image	Image translation	Pix2pix [52], PAN [127], CycleGAN [145], DiscoGAN [57]
	Super resolution	SRGAN [65]
	Object detection	SeGAN [28], Perceptual GAN for small object detection [69]
	Object transfiguration	GeneGAN [144], GP-GAN [132]
	Joint image generation	Coupled GAN [74]
	Video generation	VGAN [125], Pose-GAN [126], MoCoGAN [122]
	Text to image	Stack GAN [49], TAC-GAN [18]
	Change facial attributes	SD-GAN [23], SL-GAN [138], DR-GAN [121], AGEGAN [3]
Sequential data	Music generation	C-RNN-GAN [83], SeqGAN [141], ORGAN [41]
	Text generation	RankGAN [73]
	Speech conversion	VAW-GAN [48]
Others	Semi-supervised learning	SSL-GAN [104], CatGAN [115], Triple-GAN [67]
	Domain adaptation	DANN [2], CyCADA [47]
		Unsupervised pixel-level domain adaptation [12]
	Continual learning	Deep generative replay [110]
	Medical image segmentation	DI2IN [136], SCAN [16], SegAN [134]
	Steganography	Steganography GAN [124], Secure steganography GAN [109]

图 3: Gan Application

3.1　例子

Paired two domain data：能实现图像的分割，原图线稿化，上色等。

pix2pix：参考项目 vid2vid。

Unpaired two domain data：降低了数据制作成本，可实现图像的风格转化功能。

CycleGAN：参考项目 photo2cartoon。

Super resolution：提升图像的分辨率，老片高清还原等。

SRGAN：参考项目Anime4K。

colorize images：上色。参考项目NoGAN，style2paints。

Fake face：换脸。参考项目：faceswap。

Augment：生成数据，减少样本少的问题，同时也有增分类，检测等的性能功能。参考项目：小麦检测。

第四节　与 RL 的联系

4.1　演员与评论家

actor-critic(可以看我写的 RL 入门读本RL-foocker)

第五节　一些理论尝试

理论尝试，需要从庞杂的现象中，抽丝剥茧，提出问题，简化，符号化，给出定义，推导引理，定理。不同人背景不同，因此会出现不同的理论，但，最终那些有效的理论都将汇合在某些地方。

5.1　min-max optimization

min-max Optimization 暂略。

5.2　最优传输与几何

本小节内容主要来自 Geometric Understanding of Deep Learning，A Geometric Understanding of Deep Learning，Geometric Understanding of Deep Learning-beamer。

这里我斗胆说一下感受，前两篇文章，看的过程中，感觉有些符号上有小问题，包括编解码的理解会错位，个别单词错误，个别符号的前后含义模糊等，但这些不影响其论文的价值部分。具体过程中会给出很多定义，做一些简化，但，内容和 DL 的自洽度还是很高的。现在还处于比较单薄的阶段，还有大量的工作需要丰富，最后一篇是最近 (2020.9) 的演讲，内容上确实又丰富了一些。

要完全掌握还是有难度的 (几何部分)，所以我也简化到体会构建过程，掌握一些基本定理，省去一些具体的几何内容。

5.2.1　流形假设

定义 5.1 (流形) 一个流形 *(manifold)* 是一个拓扑空间 S，被一族开集所覆盖 $S \subset \bigcup_\alpha U_\alpha$，对于每个开集 U_α 存在一个同胚映射 $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$，$\varphi_\alpha, \mathbb{R}^n$ 分别称为坐标映射，参数域。$(U_\alpha, \varphi_\alpha)\varphi_\alpha$ 构成一个局部坐标卡，所有局部坐标卡构成流形的图册 *(atlas)*。$\mathcal{A} := \{(U_\alpha, \varphi_\alpha)\}$。在交集 $U_\alpha \cap U_\beta$ 上，每个点可以有多个局部坐标，在局部坐标间存在变换 $\varphi_{\alpha\beta} = \varphi_\beta \circ \varphi_\alpha^{-1}$。

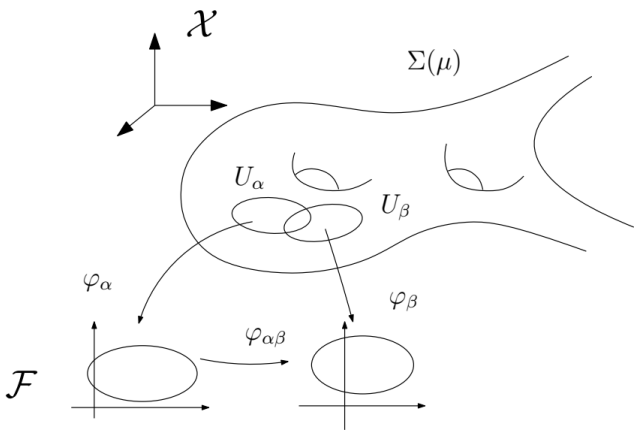

图 4: A manifold structure in the data.

Hypothesis 5.1 (流形假设) 自然界中的高维数据是嵌入在高维空间中的低维流形。

可参考manifold-hypothesis，NN-Manifolds-Topology，同时这个假设也有 Fields 奖获得者 Charles Fefferman 给出一些理论性验证Testing the Manifold Hypothesis。

回到 DL，该假设直观且有大量例证：自编码，生成模型，backbone 的特征向量，metric-learning 的表示向量等都可以给出具体例子。最简单的比如 MNIST 的 t-SNE，UMap 可视化，将 28*28 维度的数据映射到 2 维空间。可以说，MNIST 实际上是嵌入在 28*28 维空间中的 2 维流形。我们熟知的人脸也是嵌入在大约 100 维的流形中，在 ArcFace 中，可以从 100+ 维选择 2 维，将其可视化到一个 3 维球体的面上。

数学上的具体例子：嵌在三维欧氏空间中的单位球面是最为简单的二维流形，其局部参数表示为 $\gamma : (\theta, \varphi) \mapsto (x, y, z)$

$$\begin{cases} x = \cos\varphi\cos\theta \\ y = \cos\varphi\sin\theta \\ z = \sin\varphi \end{cases}$$

这里球面是流形，三维欧氏空间是背景空间，$\gamma : (\theta, \varphi)$ 是局部坐标。参数化映射可以写成

$$\begin{cases} \theta = \tan^{-1}\frac{y}{x} \\ \varphi = \sin^{-1} z \end{cases}$$

更复杂的例子：

(a) Input manifold　(b) latent representation　(c) reconstructed manifold

图 5: Auto-encoder pipeline.

采样自编码网络，将三维弥勒佛映射到二维平面，然后重构回三维曲面。弥勒佛是嵌入在我们生活的世界中的三维流形，但可以用二维来表示。代码参考reconstruct pointcloud ae。

利用图4的框架，容易写出 VAE 模型的交换图：

$$\begin{array}{ccc} \{(\mathcal{X}, \mathbf{x}), \mu, \Sigma\} & \xrightarrow{\varphi} & \{(\mathcal{F}, \mathbf{z}), D\} \\ & \searrow^{\varphi\circ\psi} & \downarrow{\scriptstyle \psi} \\ & & \{(\mathcal{X}, \tilde{\mathbf{x}}), \tilde{\Sigma}\} \end{array}$$

其中 φ 是编码映射 (网络)，ϕ 是解码映射，μ 是背景空间 $\mathcal{X}$ 上的概率密度函数，Σ 是 μ 的支撑集，$\mathcal{F}$ 是隐空间。于是自编码模型可表示为

$$\varphi, \psi = \operatorname{argmin}_{\varphi,\psi} \int_{\mathcal{X}} \mathcal{L}(\mathbf{x}, \psi \circ \varphi(\mathbf{x})) d\mu(\mathbf{x})$$

其中 $\mathcal{L}(\cdot,\cdot)$ 是损失函数。转化实际离散样本，即

$$\min_{\theta} \mathcal{L}(\theta) = \min_{\theta} \frac{1}{k} \sum_{i=1}^{k} \left\| \mathbf{x}^{(i)} - \psi_\theta \circ \varphi_\theta \left(\mathbf{x}^{(i)} \right) \right\|^2$$

这里将 VAE 的编解码简化为 ReLU DNN 模型 (原始的 VAE 就是如此)，但在工程应用上，模型会增加很多其他元素，BN，结构设计等来满足更复杂的样本空间，对这些的理论分析，还处于发展中。

5.2.2　网络的学习能力

定义 5.2 (分片线性映射) 映射 $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ 是分片线性映射，如果 $\mathbb{R}^n$ 能写成有限多面体的并，且 φ 在每个多面体上都仿射线性。用 $\mathbb{R}^n$ 上的最大连通子集且 φ 在其上仿射线性，来表示映射 φ 的片数，记为 $\mathcal{N}(\varphi)$，称其为 φ 的分片线性复杂度。

定理 5.2 (万有逼近定理) 任意的连续函数 $f : \mathbb{R}^n \to \mathbb{R}$，都可以被深度神经网络以任意精度逼近。

证明 5.3 分片线性函数（*piecewise linear function*）在希尔伯特空间 $L^2(\mathbb{R}^n)$ 中稠密，因此能够以任意精度逼近任何可积的连续函数。任意分片线性函数 $f : \mathbb{R}^n \to \mathbb{R}$ 可以分解为分片线性凸函数之和、之差，

$$f(\mathbf{x}) = \sum_{i=1}^{k} s_i g_i(\mathbf{x})$$

这里系数 $s_i = \pm 1$　$g_i(\mathbf{x})$ 是分片线性凸函数，

$$g_i(\mathbf{x}) = \max \{ \langle \mathbf{w}_j, \mathbf{x} \rangle + b_j \}$$

$\max(x, y)$ 可以用两层神经网络来实现，隐层的激活函数为 $ReLU$，由此给定任意分片线性函数 $f : \mathbb{R}^n \to \mathbb{R}$，我们都可以构造一个 $ReLU$ 神经网络来加以实现。由此，对于给定的连续函数和误差界，我们都可以构造一个神经网络来逼近函数，其误差小于误差界。

若采用流形观点，Lei Na 等人思考了这样一些问题：给定一个流形，给定一个深度神经网络，这个网络能否学习这个流形，即能否实现参数化映射，构造参数表示？在论文G中作者首先从简单的螺旋线出发，认识到：

“ReLU 深度神经网络的每个神经元代表一个超平面，将输入空间一分为二；众多超平面将输入空间剖分，然后将每个胞腔线性映射到输出空间，由此得到编码、解码映射的分片线性逼近。进一步，我们可以得到如下关键的观察：流形（螺旋线）被输入空间上的胞腔分解分割成很多片，每片流形所在的胞腔被线性映射到参数域上（一段直线），这个线性映射限制在流形上是拓扑同胚”。引自这里。

由此，他们思考了如下问题，定义相应概念，给出了具体答案 (虽然目前相对粗糙)。

1. 如何从几何上刻画一个深度神经网络的学习能力？是否可以定义一个指标来明确表示神经网络学习能力的上限？
2. 如何从几何上刻画一个流形被学习的难度？是否可以定义一个指标来明确表示这一难度？
3. 对于任意一个深度神经网络，如何构造一个它无法学习的流形？

首先给出基本网络定义：

定义 5.3 (ReLU DNN) 给定序列正整数 $\{w_0, w_1, w_2, \ldots, w_k, w_{k+1}\}$，仿射变换 $T_i : \mathbb{R}^{w_{i-1}} \to \mathbb{R}^{w_i}, i = 1, \ldots k$ 和线性变换 $T_{k+1} : \mathbb{R}^{w_k} \to \mathbb{R}^{w_{k+1}}$，则映射 $\varphi_\theta : T_{k+1} : \mathbb{R}^{w_0} \to \mathbb{R}^{w_{k+1}}$

$$\varphi_\theta = T_{k+1} \circ \sigma \circ T_k \circ \cdots \circ T_2 \circ \sigma \circ T_1$$

表示一个 $k+1$ 层的 *ReLU DNN*。其中 w_0, w_{k+1} 表示网络的输入输出维度，$k+1$ 表示网络的深度，$\max\{w_1, \ldots, w_k\}$ 表示网络的宽度，$\sum_{i=1}^{k} w_i$ 表示网络的大小。

根据螺旋线的考察，引发出如下概念：

定义 5.4 (激活路径) 给定 $\mathrm{x} \in \mathcal{X}$,x 的激活路径为在神经网络中所有被激活的神经元,表示为 $\rho(\mathbf{x})$。于是激活路径定义了一个集值函数 $\rho : \mathcal{X} \to 2^{\mathcal{S}}$。其中 $\mathcal{S}$ 表示网络的所有神经元。

定义 5.5 (胞腔分解) 对给定的 *ReLU DNN* 编码映射 φ_θ，输入数据 $\mathrm{x}_1, \mathrm{x}_2 \in \mathcal{X}$ 关于其激活路径的等价类给出 $\mathcal{X}$ 的一个分解，称为胞腔分解。即

$$\mathcal{D}(\varphi_\theta) : \mathcal{X} = \bigcup_\alpha U_\alpha$$

$\mathrm{x}_1, \mathrm{x}_2 \in \mathcal{U}_\alpha$ 当且仅当 $\mathrm{x}_1 \sim \mathrm{x}_2$。

在此基础上，给出了流形的分片线性复杂度定义 (定义较多，略)，从定义出发，得到一些基本性质，并给出了上面问题的答案。比如第 3 问：

定理 5.4 对任意的 *ReLU DNN* $N(w_0, w_1, \ldots, w_k, w_{k+1})$，都存在一个嵌入在 $\mathbb{R}^{w_0}$ 的流形 Σ，使得 Σ 不能被 N 编码。

证明 5.5 论文中作者利用 *Peano* 曲线给出了构造性证明。

上面激活路径推出的胞腔分解，从我个人经验来看，很符合 DL。

总结：“深度神经网络将输入空间分解的最多胞腔个数定义为网络的分片线性复杂度，代表了网络学习能力的上限；流形需要被分解，每一片可以被背景空间的线性映射所参数化，这种分解所需的最少片数定义为流形的分片线性复杂度。一个网络能够学习一个流形的必要条件是：流形的复杂度低于网络的复杂度。对于任意一个网络，我们都可以构造一个流形，使得此网络无法学习”。

5.2.3　概率变换

实际 DL 不止是隐空间的表示，还需要拟合不同类别的概率分布，放到流形上来说，就是需要控制隐空间的概率分布。实际工业中分类，识别，生成模型等采用了交叉熵，度量损失等来学习概率分布。后者虽然在整个实验上有一些直观认知来指导实验 (感受野，HeatMap，语义特征，Context 等)，但逃离不了其黑箱的特点。在仔细考察生成模型后，Lei Na 等人提出一个半黑半白的模型AE-OMT。

Hypothesis 5.6 (聚类分布假设) 自然界中一类数据的不同子类对应着流形上的不同概率分布，这些分布之间的距离大到能够将这些子类区分。

见Cluster hypothesis。

定义 5.6 (保测度映射) 给定欧氏空间中的两个区域和定义其上的概率测度 (X,μ) 和 (Y,ν)，总测度相等 $\mu(X)=\nu(Y)$。假设 $T:X\to Y$ 是一个区域间的映射，如果对于任意的可测集合 $B\subset Y$，都有

$$\int_{T^{-1}(B)} d\mu = \int_B d\nu$$

则称此映射保持测度，记成 $T_*\mu=\nu$。

对任意 $x\in X, y\in Y$，它们的距离记为 $c(x,y)$，保测度映射 T 的传输代价定义为：

$$C(T) := \int_X c(x,T(x))d\mu(x)$$

1781 法国数学家 Monge 提出最优传输问题：求使得代价最小的保测度映射。这个映射被称为是最优传输映射，最优传输映射的传输代价被称为是两个概率测度之间的 Wasserstein 距离，记为 $W_c(\mu,\nu)$。这就回到了1.2。在 1.2 节，我们轻描淡写的梳理了 Wasserstein 距离的转化逻辑，事实上该系列工作获得了 1975 年的诺贝尔经济奖，所以我们还是回到一些简要的逻辑梳理的流程上来吧。

Kantarovich 将最优传输问题转化成 Kantarovich 问题，Wasserstein 距离等于

$$\min_{\rho}\left\{\int_{X\times Y} c(x,y)d\rho(x,y) \mid \rho_{x*}=\mu, \rho_{y*}=\nu\right\}$$

Kantarovich 问题等价于其对偶形式，Wasserstein 距离等于

$$\max_{\varphi}\left\{\int_X \varphi(x)d\mu(x) + \int_Y \varphi^c(y)d\nu(y)\right\}$$

这里 $\varphi^c(y)$ 是 c-变换

$$\varphi^c(y) = \inf_{x\in X}\{c(x,y)-\varphi(x)\}$$

φ 被称为 Kantarovich 势能函数。

如果距离函数 $c(x,y)$ 为 $|x-y|$ 即 L^1 距离，则 $\varphi^c=-\varphi$，并且 φ 是 1-Lipsitz 函数，也就是 1.2 小节处理的情况。但，若采用 L^2 距离，就有了一系列联系：Brenier 定理，蒙日-安培方程，凸几何闵可夫斯基理论等 (这块内容较多，略)。在论文 AG中，作者证明了最优传输的 Brenier 理论和凸几何的 Alexandroff 理论是等价的，并给出了半透明的 AE-OMT 模型。

现在我们来结合图6，给出此小节的书写逻辑，和半黑半白的含义。

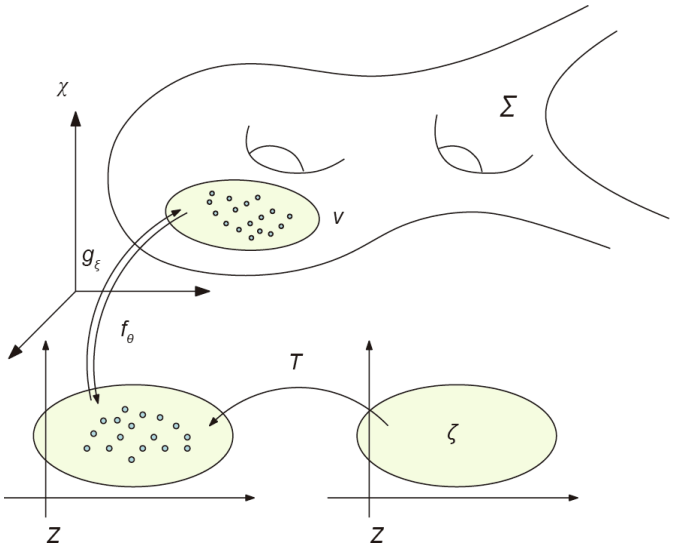

图 6: 半透明深度网络模型.

从前面小节我们知道，实际应用中，生成模型的隐空间一般选为高斯分布或均匀分布 (采样方便)，然后将此样本用生成网络映射到实际图像。那么回到流形观点，我们需要控制隐空间的“推前”分布为给定的高斯或均匀分布，这里推前是指，对给定的流形和隐空间，流形到隐空间的映射需要保概率测度。比如图5 中的弥勒佛，将其映射到圆盘，在圆盘上均匀采样，再映射回曲面，就不在保持均匀。若要使其均匀，需要在圆盘 (隐空间) 上构造一个自同胚映射，而这一块就是最优传输理论可以解决的问题。

在图6中的 T 就是最优传输映射，它可以具体的计算出来 (基于变分法的构造，较为复杂)，是为白，而左边 g_ξ, f_θ 是一个简单的自编码网络，是为黑。这里没有对抗，原因在于原始 GAN 在判别器最优时，生成器的优化，这整个过程可以用最优传输一步解决。

个人感受：整个网络变得简单，但最优传输映射求解效率较低，另外，隐空间的映射，与此类似的操作，在 StyleGAN 中有用到。

5.2.4　基本定理

要构建 DL 的理论，是一个长足的过程，不仅需要普通研究员的经验知识，更需要理论知识的洞见，前者不仅需要丰富的工程经验，也需要对一些细节的理论分析，后者不仅需要深厚的理论知识，还需要来自实际的工程经

验。从现实来看，这是一个合作的过程。这一小节就是在这种认识下，给出一些“洞见性”定理。

引理 5.7 (Urysohn's Lemma) *X 是一个正规拓扑空间，当且仅当只要 A 和 B 是 X 的不交闭子集，就存在一个从 X 到单位区间 $[0,1]$ 的连续函数：$f : X \to [0,1]$，使得对于所有 A 内的 a，都有 $f(a) = 0$，而对于所有 B 内的 b，都有 $f(b) = 1$。*

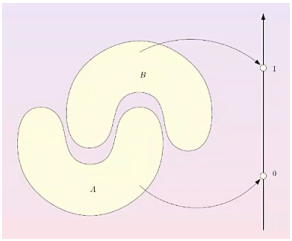

图 7: Urysohn lemma example

编码映射将原始变量映射成可分开的闭集 (特征空间的分布)，模式识别函数完成对应的标签映射 (特征向量到类别映射)。

定理 5.8 (General Position Theorem) *任何嵌入到 $\mathbb{R}^n$ 中的 m 维流形，若没有扭结，则 $n \geq 2m + 2$。*

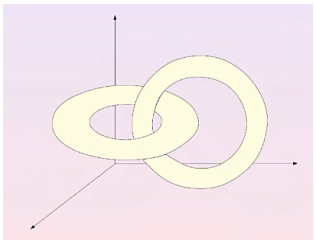

图 8: 低维扭结.

这表明，将有扭结 (缠绕，类类混合) 的复杂数据升到高维，然后再投影，可使其分开，这在实际的 DL 经验中是惯用手法。

定理 5.9 (Whiteny Embedding) *任何光滑的实 m 维流形 (Hausdorff 且第二可数) 能被光滑的嵌入到实 $2m$ 维欧式空间 ($\mathbb{R}^{2m}$)。*

升维的保证。

定理 5.10 (Kolmgorov-Arnold Respresentation) *f 是一个 n 元连续函数，则 f 可由有限个单变量连续函数和加法二元运算表示*

$$f(\mathbf{x}) = f(x_1, \ldots, x_n) = \sum_{q=0}^{2n} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

多元函数可以用单元函数逼近，Kolmgorov-Arnold Respresentation。单个神经元可能就是一些单元函数。

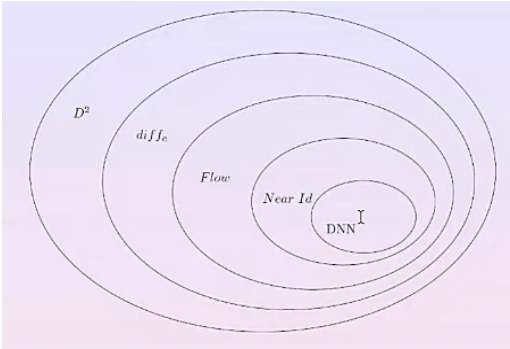

图 9: Universal-Approximation.

构造一个嵌套的映射空间序列, $\mathscr{F}_{k+1}$ 比 $\mathscr{F}_k$ 简单，

$$\mathscr{F}_0 \supset \mathscr{F}_1 \supset \mathscr{F}_2 \cdots \supset \mathscr{F}_n$$

每个映射 $f \in \mathscr{F}_k$ 可以通过映射的有限组成来近似 $g_1, g_2, \ldots, g_r \in \mathscr{F}_{k+1}$ $f = g_1 \circ g_2 \circ g_3 \cdots g_r$。最终 $\mathscr{F}_n$ 可由深度神经网络计算。

从外层到内层的函数空间越来越简单，直到最简单的 DNN。Flow 流式生成模型则采用了类似流程。