GAN的无监督条件生成（二）：CoGAN与UNIT

【参考文献】

[1] Coupled Generative Adversarial Networks 2017

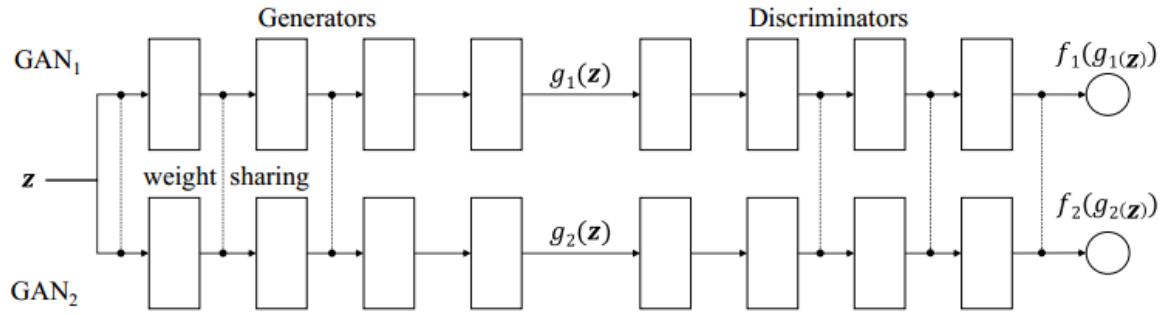
[2] Unsupervised Image-to-Image Translation Networks 2017

GAN的无监督条件生成一般是指图像翻译的任务，通常做法有两类，一类是直接进行转换，另一类是将不同的domain投影到同一个空间。本篇笔记介绍后一类方法的两个经典工作：CoGAN和UNIT。

1. CoGAN

生成模型，比如GAN中的生成器，一般以随机采样的隐变量作为输入，生成图像作为输出，在这其中，浅层的layer负责从隐变量中解码出抽象的语义信息，而深层layer则负责进一步解码出更具体的图像细节信息。而判别模型则正好相反，比如GAN中的判别器，浅层的layer负责抽取低级别的信息，而高层layer负责抽取更高级的抽象语义信息。

CoGAN（Couple GAN）的基本假设是，**不同domain的一对图像，应该共享同样的高级语义信息**。因此，CoGAN使用了一对GAN，每个GAN负责生成一个domain，对于生成器的浅层layer和判别器的高层layer采用参数共享的策略，保证不同domain对高级语义信息处理的一致性。而domain-specific的信息，则由生成器的高层layer和判别器的浅层layer负责处理。



因此，现在每次只要采样一个隐变量z，送入到两支GAN中，就能产生相互关联的一对图像。整个CoGAN可以看作是两组网络的互相对抗，一组是两支GAN的生成器，它们互相合作，另一组是两支GAN的判别器，它们同样有着相互合作的关系。

作者的实验表明 [1]，生成器中共享的层数越多，那么生成的一对图像关联程度就越高，但是判别器中共享层数的多少并没有太大影响，尽管如此，共享判别器的参数依然可以减少参数的数量。

Unsupervised Domain Adaptation

CoGAN可以用于无监督域适应。假如现在domain D_1是有label的source domain，domain D_2是没有label的target domain。那么CoGAN实现UDA的做法是在判别器上再加一层softmax layer用于分类，并且使用D_1的图片和label进行训练。同时，CoGAN还要进行原本的生成一对图像的任务。

由于判别器中高层layer都是共享的，所以D_1和D_2的分类器的唯一区别就是浅层layer不同。D_1的判别器需要处理D_1的生成图片和真实图片的鉴别，以及D_1的图片分类两个任务，而D_2的判别器只需要处理D_2的生成图片和真实图片的鉴别。

整个网络训练完毕后，将共享的高层layer与D_2的浅层layer组合，就能得到D_2上的分类器。这背后的假设依然是两个domain共享同样的高级语义信息，而低级别的domain信息则由domain-specific的低层layer分别处理。

Cross-Domain Image Transformation

对于domain D_1的一张给定图像x_1，CoGAN可以生成它在domain D_2的对应图像x_2。具体做法是，先根据训练好的生成器g_1，找到x_1的对应的隐变量z，求解如下优化问题：

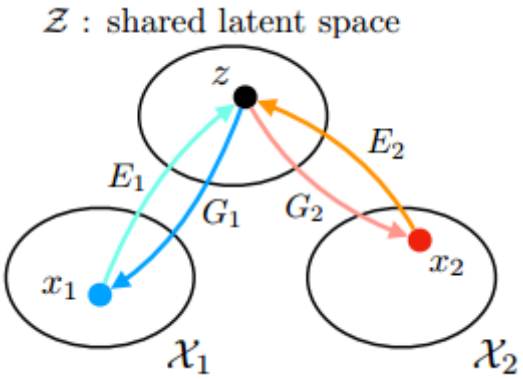
$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \mathcal{L}\left(g_1(\mathbf{z}), \mathbf{x}_1\right)$$

文章中使用了L-BFGS算法来求解这个优化问题。

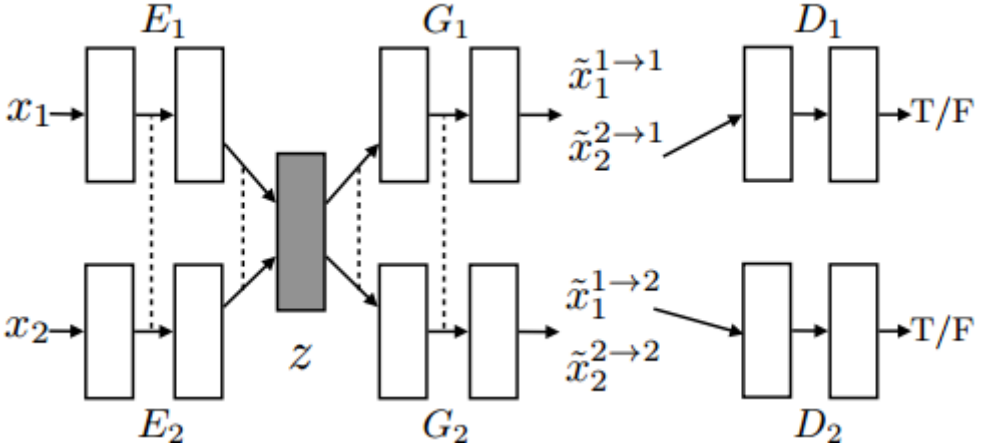
找到隐变量z后，再将其送入D_2对应的生成器g_2，就能得到转换后的图像x_2。

2. UNIT

UNIT（UNsupervised Image-to-image Translation）可以看作是CoGAN的一个改进工作，增强了跨域图像转换的能力。**UNIT的基本假设是，两个domain中的一对图像共享一个相同的隐空间**，因此可以先把其中一个domain的图像转换到共同隐空间，再从共同隐空间映射到另一个domain对应的图像，这种思想与CycleGAN的直接转换不同，中间多了一个隐空间作为桥梁。



UNIT采用了共享参数的两支VAE-GAN结构，并且假定两支VAE的部分拥有相同的隐空间。



与CoGAN类似，UNIT中对编码器E_1和E_2的高层layer，生成器G_1和G_2的底层layer都进行了参数共享，来保证语义信息编码和解码的一致性。

UNIT中不同部分扮演的角色如下：

Networks	$\{E_1, G_1\}$	$\{E_1, G_2\}$	$\{G_1, D_1\}$	$\{E_1, G_1, D_1\}$	$\{G_1, G_2, D_1, D_2\}$
Roles	VAE for $\mathcal{X}_1$	Image Translator $\mathcal{X}_1 \to \mathcal{X}_2$	GAN for $\mathcal{X}_1$	VAE-GAN [14]	CoGAN [17]

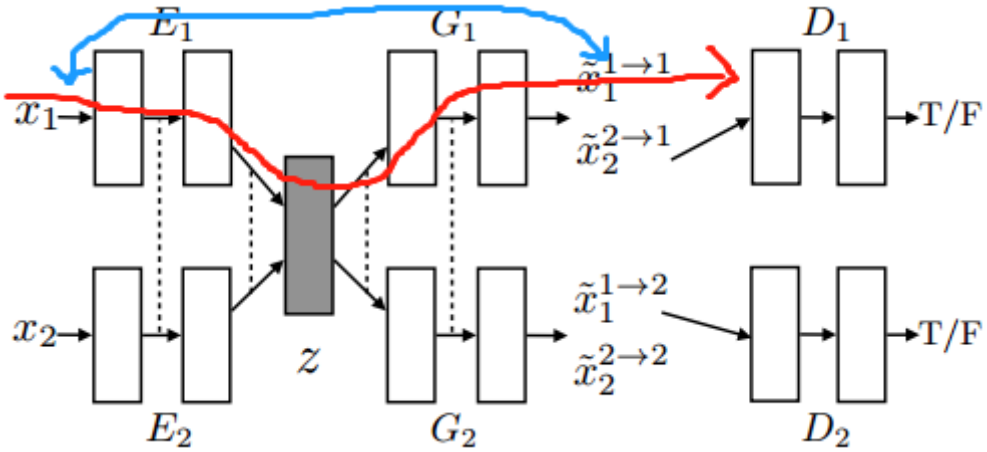
图像转换的流程如下：假设给定图像来自domain $\mathcal{X}_1$，用x_1表示，要生成另一个domain $\mathcal{X}_2$中对应的图像x_2，首先用对应domain $\mathcal{X}_1$的编码器E_1对x_1进行编码，再用domain $\mathcal{X}_2$ 对应的生成器（解码器）G_2解码即可。

UNIT的**训练过程**比较复杂，涉及到三种loss：

1）VAE loss

要求输入图片x_1或x_2经过VAE后要能重建回原来图像。不同domain产生的后验
$q_1\left(z_1|x_1\right) \equiv \mathcal{N}\left(z_1|E_{\mu,1}\left(x_1\right), I\right)$和$q_2\left(z_2|x_2\right) \equiv \mathcal{N}\left(z_2|E_{\mu,2}\left(x_2\right), I\right)$要向同一个先验
$p_\eta(z) = \mathcal{N}(z|0, I)$逼近。注意这里的后验方差固定为1，也就是说没有编码器中没有方差拟合的网络。
同时，解码器用Laplacian分布建模，对应的重建loss是L1的形式。

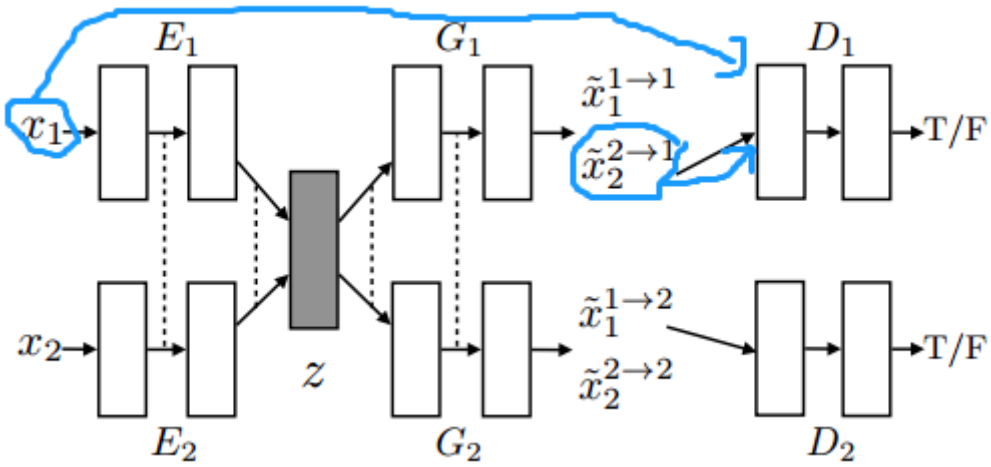
$$\mathcal{L}_{\mathrm{VAE}_1}\left(E_1, G_1\right) = \lambda_1 \mathrm{KL}(q_1\left(z_1|x_1\right)\|p_\eta(z)) - \lambda_2 \mathbb{E}_{z_1\sim q_1(z_1|x_1)}\left[\log p_{G_1}\left(x_1|z_1\right)\right]$$
$$\mathcal{L}_{\mathrm{VAE}_2}\left(E_2, G_2\right) = \lambda_1 \mathrm{KL}(q_2\left(z_2|x_2\right)\|p_\eta(z)) - \lambda_2 \mathbb{E}_{z_2\sim q_2(z_2|x_2)}\left[\log p_{G_2}\left(x_2|z_2\right)\right]$$



2）GAN loss

负责鉴别转换后的图像和真实图像。注意这里只针对转换后的图像（如$\tilde{x}_2^2 \to 1$）作判别，因为重建图像（如$\tilde{x}_1^1 \to 1$）已经有reconstruction loss作为约束了。

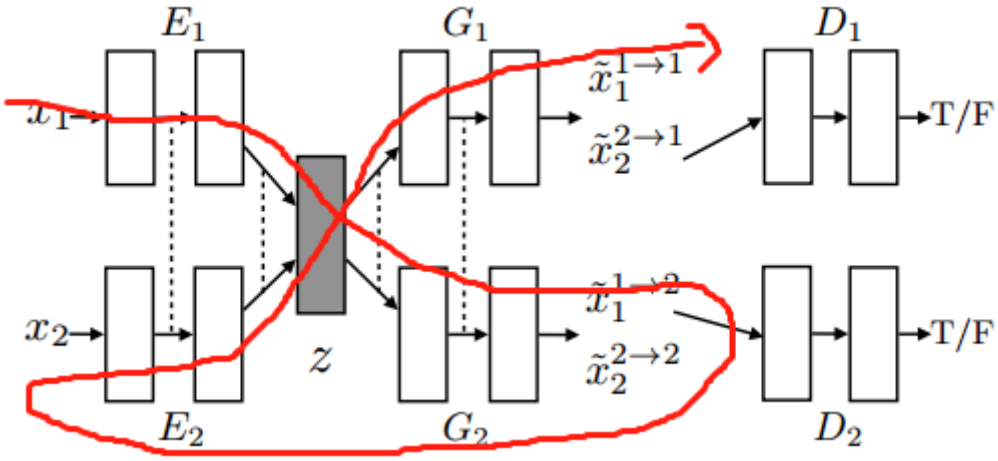
$$\mathcal{L}_{\mathrm{GAN}_1}\left(E_1, G_1, D_1\right) = \lambda_0 \mathbb{E}_{x_1\sim P_{x_1}}\left[\log D_1\left(x_1\right)\right] + \lambda_0 \mathbb{E}_{z_2\sim q_2(z_2|x_2)}\left[\log(1 - D_1\left(G_1\left(z_2\right)\right))\right]$$
$$\mathcal{L}_{\mathrm{GAN}_2}\left(E_2, G_2, D_2\right) = \lambda_0 \mathbb{E}_{x_2\sim P_{x_2}}\left[\log D_2\left(x_2\right)\right] + \lambda_0 \mathbb{E}_{z_1\sim q_1(z_1|x_1)}\left[\log(1 - D_2\left(G_2\left(z_1\right)\right))\right]$$



3）Cycle-consistency loss

为了进一步保证图像转换的质量，还加了Cycle-consistency的约束，即要求转换后的图像再转换回来，要能重建原本的图像。

$$\mathcal{L}_{\mathrm{CC}_1}\left(E_1, G_1, E_2, G_2\right) = \lambda_3 \mathrm{KL}\left(q_1\left(z_1|x_1\right)\|p_\eta(z)\right) + \lambda_3 \mathrm{KL}\left(q_2\left(z_2|x_1^{1\to2}\right)\right)\|p_\eta(z)) - \lambda_4 \mathbb{E}_{z_2\sim q_2(z_2|x_1^{1\to2})}\left[\log p_{G_1}\left(x_1|z_2\right)\right]$$
$$\mathcal{L}_{\mathrm{CC}_2}\left(E_2, G_2, E_1, G_1\right) = \lambda_3 \mathrm{KL}\left(q_2\left(z_2|x_2\right)\|p_\eta(z)\right) + \lambda_3 \mathrm{KL}\left(q_1\left(z_1|x_2^{2\to1}\right)\right)\|p_\eta(z)) - \lambda_4 \mathbb{E}_{z_1\sim q_1(z_1|x_2^{2\to1})}\left[\log p_{G_2}\left(x_2|z_1\right)\right]$$



所以，**总的loss表示为**：

$$\max_{E_1,E_2,G_1,G_2} \max_{D_1,D_2} \mathcal{L}_{\text{VAE}_1}(E_1,G_1) + \mathcal{L}_{\text{GAN}_1}(E_1,G_1,D_1) + \mathcal{L}_{\text{CC}_1}(E_1,G_1,E_2,G_2)$$
$$\mathcal{L}_{\text{VAE}_2}(E_2,G_2) + \mathcal{L}_{\text{GAN}_2}(E_2,G_2,D_2) + \mathcal{L}_{\text{CC}_2}(E_2,G_2,E_1,G_1)$$

训练D_1和D_2的时候采用梯度上升，$E_1, E_2, G_1,$ 和 G_2是固定的；训练$E_1, E_2, G_1,$ 和 G_2的时候采用梯度下降，D_1和D_2是固定的。

Domain Adaptation

UNIT同样适用于UDA，与CoGAN的做法类似，共享判别器D_1和D_2高层的参数，同时再加一个softmax layer用于分类。此时网络需要处理两个任务，一是两个domain间的图像翻译，二是对source domain的图像分类。

此外，为了保证D_1和D_2抽取的高级语义特征的一致性，还对D_1和D_2的最高层抽取的特征进行了L1约束。所以，为了作UDA，UNIT又引入了两个新的loss，加上之前的图像翻译任务，一共有8个loss。

UNIT的做法比较复杂，但是在UDA任务上，效果并没有比CoGAN提升多少。

Method	SA [4]	DANN [5]	DTN [26]	CoGAN	UNIT (proposed)
SVHN→ MNIST	0.5932	0.7385	0.8488	-	**0.9053**
MNIST→ USPS	-	-	-	0.9565	**0.9597**
USPS→ MNIST	-	-	-	0.9315	**0.9358**