Dive into GAN

Intuition

Revolution

由于天敌的捕食，枯叶蝶尝试着让自己变成树叶的样子，同时天敌对枯叶蝶的识别能力不断增强，枯叶蝶才不断变得像真正的树叶，这其实体现的就是枯叶蝶和天敌之间的对抗（Adversarial）

我们设想以下最终可能达到的状态，一种可能是天敌的识别能力变得特别高，以致于没有一只枯叶蝶能逃过它们的火眼金睛；还有一种可能是枯叶蝶的伪装能力不断提高，最终达到能够以假乱真的地步

这可能对应着对抗网络的两个方向

1. 增强分类器的准确度
2. 用于样本生成

Concept of Divergence

正如论文作者说的那样，The promise of deep learning is to discover rich, hierarchical models that represent probability distributions over the kinds of data

深度学习的目标是为了找到数据的某种概率分布，基于这一思想，就有了divergence这一概念，这被用来刻画两种分布之间的相似性，生成任务也就是要找到与数据的总体分布相似的分布，也就是要最小化divergence

Generator&Discriminator

枯叶蝶视为Generator，其任务就是map formulaic distribution(Gussian Distribution) to distribution over the given data

天敌视为Discriminator，其任务是尽可能地准确识别藏在真样本中的假样本

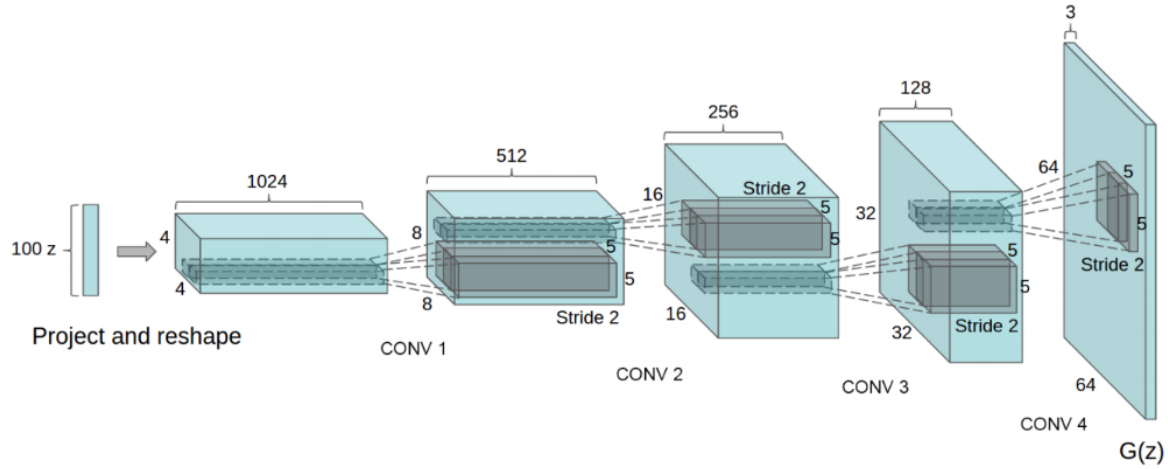
Discriminating Model

可以使用CNN等分类器构建Discriminator，网络结构比较简单

Generating Model

之前讲到DL的目标是找到两个distribution 之间的map，Generator 可以是从Gussian Noise到图像的map，这似乎听起来非常不可思议，我们竟然能从Noise当中提取出有用的特征，最后生成图像，但是已知两个distribution 的samples，神经网络确实可以找到这两者之间的联系

从特征恢复到图像，这是一个上采样的过程，可以通过转置卷积（Transpose Conv）实现



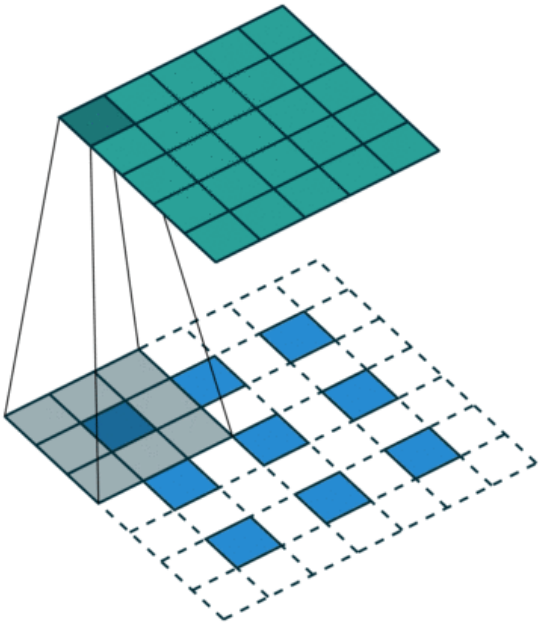
图像生成网络（基于DCGAN）

卷积公式：

$$n_{output} = \lfloor \frac{n_{input} - kernel\_size + 2 * padding}{stride} + 1 \rfloor$$

反卷积公式：

$$n_{output} = (n_{input} - 1) * stride - 2 * padding + output\_padding + kernel\_size$$



转置卷积过程

输入为3x3的feature map，输出为5x5的feature map，这里stride=2，表示对输入进行插值，间隔为1，这里对输入的padding=1，对输出的padding=0

有一个问题是这里进行插值处理的目的是什么？采用不同的插值方法对效果会不会有影响？

How to trade off Generator&Discriminator

Optimization Objectives

我们最终的目的是要让Generator产生尽可能逼真的图片，基于前文的Intution，得到与所给数据尽可能相似的分布，也就是最小化Divergence，因此G的优化目标如下

$$G^* = arg\min_G Div(P_G, P_{data})$$

其中P_G, P_{data}分别为生成图像和真实图像的分布，Div为Divergence的度量

一种直觉是Divergence越大，那么G(z)和输入图像x就越难区分，换言之，Discriminator的分类效果越差，那么什么可以用于衡量D的分类效果呢？分类任务中，最终训练得到结果的似然函数越小分类效果越差，似然函数作为Div

但是在生成任务中单独对Discriminator进行反复的训练是没有必要的，因为在前期的训练中，真样本和假样本是非常容易区分的。不过我们改为在经历一个batch之后（视为D暂时优化结束）优化G

D的优化目标为

$$D^* = arg\max_D E_{x\sim p_{data}(x)}[logD(x)] + E_{z\sim p_z(z)}[log(1-D(G(z)))]$$

简写为

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

代入得到G的优化目标

$$G^* = arg\,\min_G \max_D E_{z\sim p_z(z)}[log(1-D(G(z)))]$$

但是，这实现起来有个问题，在训练早期$E_{z\sim p_z(z)}[log(1-D(G(z)))]$基本为0，这也就导致了G无法训练，论文作者提出了不如最大化$E_{x\sim p_{data}(x)}[logD(G(z))]$因此得到最终的优化目标为

"

Balance Point

根据论文中的推导

$$C(G) = max\,V(G,D) = E_{x\sim p_{data}(x)}[log\frac{p_{data}(x)}{p_{data}(x)+p_g(x)}] + E_{x\sim p_g(x)}[log\frac{p_g(x)}{p_{data}(x)+p_g(x)}]$$

当$p_{data}(x)=p_g(x)$，$C(G)=-log4$，此时$D(x)=\frac{1}{2}$

Questions

训练到平衡点时，Discriminator是不是失去分类功能了，按照公式推导来说好像是这样的

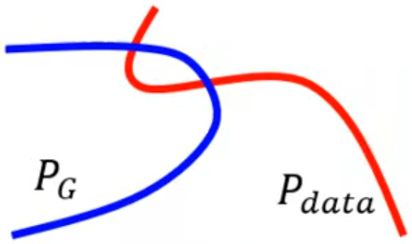
Challenges

事实上，p_{data}, p_g很难达到完全相同，一是因为维度太高，而是因为本来我们用的就是抽样的方法，因此很难完整描述整体的分布

- 1. The nature of data

  Both P_{data} and P_G are low-dim manifold in high-dim space.

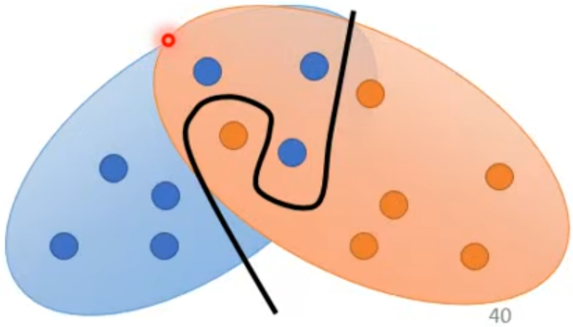
  The overlap can be ignored.



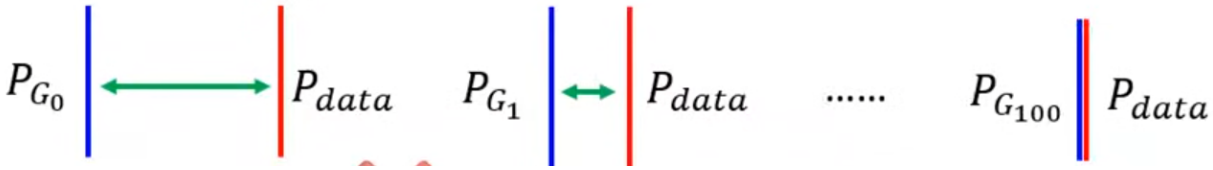
- 2. Sampling

  Even though P_{data} and P_G have overlap.

  If you do not have enough sampling ……



这种情况下，Dicriminator的准确度是100%，当我们选择用极大的似然函数描述Divergence，当两个分布完全不重合的时候，$C(G)$为常数，Generator无法训练



实际上我们可以采用其他的Divergence（例如Wasserstein distance），在上面的情况下Generator仍然可以训练。

Future Work

WGAN

有没有更高效的上采样方法？

Coding Issues

1. torchvision 用的是RGB，因此在transform中需要进行转化，展示图片的时候需要转回来
2. glob.glob() 函数获取目录下的所有指定类型的文件

3. plt.imshow() 显示彩色图像时对数据格式有要求：像素值在[0，1]范围的图片数据格式要求是float，[0，255]范围内的图片数据格式要求是uint8

Problems While Training

发现一开始采用常用的归一化方法，发现效果奇差，Discriminator的loss始终为0，将数据归一化到[-1,1]后，效果得到了明显的提升

推测可能与激活函数有关，正负在信息上的差异是巨大的

Task: Generate Figures

epoch 1

epoch 4

epoch 14

epoch 20

result

Conclusion

本文介绍了生成对抗网络（GAN）的概念、原理和实现。GAN 的目的是生成尽可能逼真的图片，通过让生成器（Generator）和判别器（Discriminator）相互对抗的方式训练，使生成器生成的图片越来越逼真。GAN 的核心在于如何平衡生成器和判别器的训练，以达到最优的训练效果。文章介绍了如何实现生成器和判别器的网络结构，并详细讲解了如何优化 GAN。此外，文章还介绍了一些训练 GAN 时可能遇到的问题，以及如何解决这些问题。最后，文章给出了一些生成器生成的图片，以展示 GAN 的效果。