基于 GAN 的天气场景图像生成算法的研究与实现

1951976　李林飞

日期：2021 年 4 月 22 日

摘　要

机器学习（Machine Learning, ML）是对计算机算法的研究，该算法可通过经验和数据使用来自动提高。虽然它只是被认为是人工智能的一个领域，但其本身包含的范围也是非常广泛，比如监督学习、无监督学习、强化学习。目前主流的算法包括回归算法、支持向量机算法、决策树算法、神经网络、深度学习、降维算法等。鉴于其内容的庞大性与复杂性，本文只对其一个分支上的实例——生成对抗网络 (GAN) 进行研究。通过神经网络和深度学习实现的 GAN 是一种半监督和无监督学习模型，它可以在不需要大量标注数据的情况下学习深度表征，并采用对抗博弈的思想产生“以假乱真”的输出信息。本文通过采集校园中的天气图像作为数据集，实现了天气图像的生成与风格迁移。

关键词： GAN, 梯度下降, 模式崩溃, 图像风格迁移, 机器学习 (Machine Learning)

1　引言

机器学习 (Machine Learning) 涉及的内容纷繁复杂，我对目前主流机器学习算法进行调研，汇总如下 (见图 1)。

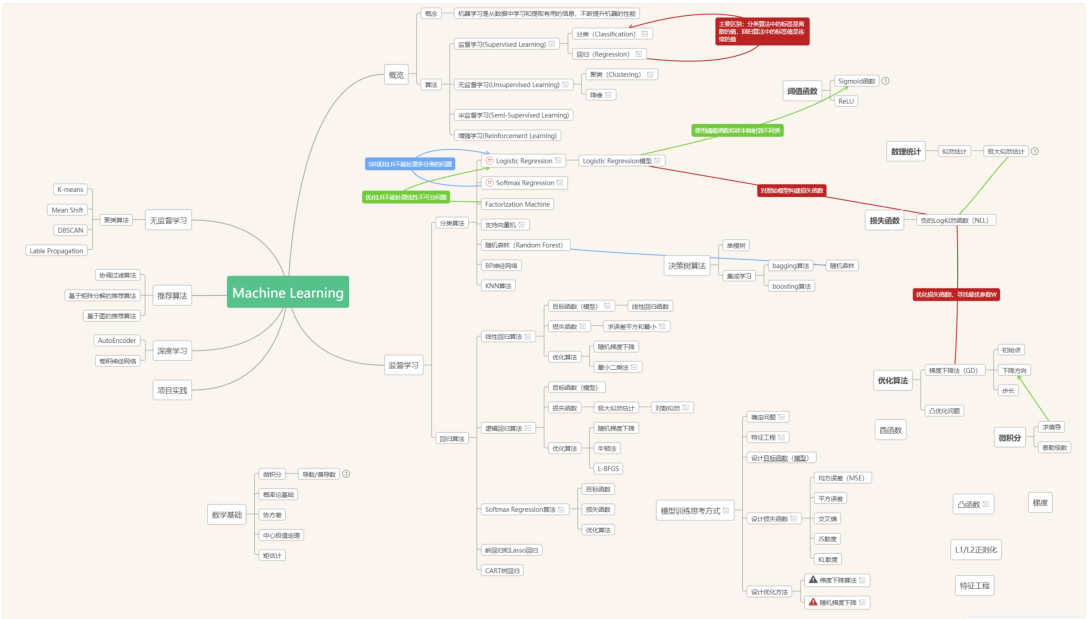

图 1: ML 算法汇总

而本文研究的主题只是机器学习分支深度学习和神经网络结合的一个模型——生成对抗式网络 (GAN,Generative Adversarial Networks)。它是一种半监督和无监督学习模型，无监督意味着

它能生成不存在于真实世界的数据，所有的生成内容都是算法自己学习产生的。不过通过加入少量标签，就是半监督学习模型，也可以产生我们想要的数据。关系图见图 2.

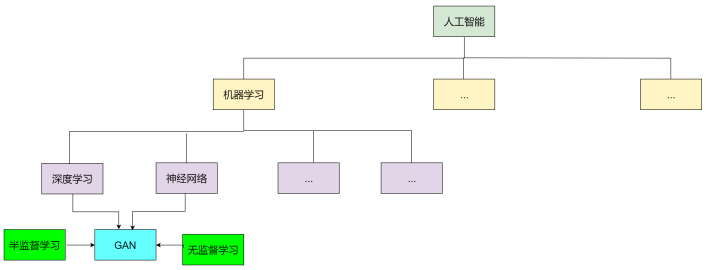

图 2: GAN 关系图

生成对抗网络 (GAN) 自从 2014 年被 Good Fellow 提出后发展迅速Goodfellow et al. (2014)，被 Yann LeCun 称为 “adversarial training is the coolest thing since sliced bread”。它主要应用于图像生成、风格迁移、语音迁移等方面。我对二次元图像生成并不感兴趣，但可以借鉴相关实现的方法。故将 GAN 与天气图像结合，可以在很多领域中发挥着重要的作用。比如，如果 GAN 生成的场景图像足够真实，完全可以模拟无人驾驶的路况场景，从而在实验阶段就可以完成无人驾驶汽车的上路测试工作，大大降低了测试的成本和风险。基于此，本文通过人工采集校园中的天气图像作为数据集，以及网上的数据集 summer2winter_yosemite，通过朴素 GAN 实现了天气场景图像生成和风格迁移，探讨了深度学习过程中存在的普遍问题，即梯度消失和模式崩溃。整个实验过程就是一个从零到一的学习过程，参考了目前已有的一些二次元图像生成的实例。

2 GAN

2.1 GAN 的基本原理

生成对抗网络（GAN）是通过生成器（Generator）和判别器（Discriminator）两者之间彼此对抗学习实现的。在模型图中（图 3），D 代表判别器，用来判别生成的图像是否是真实的，G 代表生成器，通过学习输入的真实图像数据 x 的概率分布来生成相应的图像，这个过程中需要为生成器提供随机噪声 z 指示它如何去刻画输入图像的分布，比如生成符合正态分布的噪声。X' 就是合成的图像。

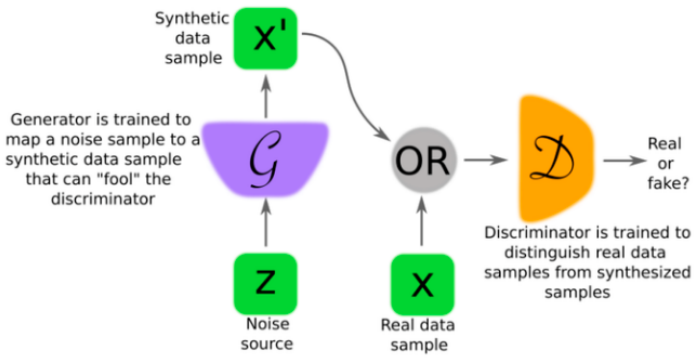

图 3: GAN 的模型图

从中我们可以看到，GAN 的核心原理就是生成器与判别器之间相互对抗、相互博弈。生成器的目的是生成更加真实的图像直到可以“骗过”判别器；而判别器的目的就是尽可能的掌握真实图像的特征，提高区分真假图像的能力，在这两者之间的不断迭代学习过程中，我们就可以得到“以假乱真”的图像。

从数学的角度，判别器与生成器之间的博弈对可以用一个数学公式表示Goodfellow et al. (2014)，如下：

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

这个公式并不复杂，只要明确各个符号的含义就能很好的理解。其中 D 为判别器，G 为生成器。$E_{x\sim P_{data}(x)}$ 表示期望 x 是从 P_{data} 分布中获取；x 表示真实数据，P_{data} 表示真实数据的分布。这项的含义是：判别器判别出真实数据的概率，判别器的目的就是最大化这一项，即对于服从 P_{data} 分布的 x，判别器可以准确得出 $D(x) \approx 1$。$E_{z\sim P_z(z)}[log(1-D(G(z)))]$ 表示期望 z 是从 $P_z(z)$ 分布中获取；z 表示生成数据；$P(z)$ 表示生成数据的分布。这一项的含义要分别从生成器和判别器的角度进行分析：对于判别器 D 而言，如果向其输入的是生成数据，即 $D(G(z))$，判别器的目标就是最小化 $D(G(z))$，即判别器希望 $D(G(z)) \approx 0$, 或者说使 $log(1-D(G(z)))$ 最大化。但对于生成器而言，其目标与判别器相反，它希望 $D(G(z)) \approx 1$, 使 $log(1-D(G(z)))$ 最小化。而 $min_G max_D$ 的含义就是从判别器 D 的角度最大化 $V(D,G)$, 再从生成器 G 的角度最小化 $V(D,G)$。

进一步推导，我们可以得到生成器 G 的目标函数为

$$G^* = argmin_G max_D V(G,D)$$

判别器 D 的目标函数为

$$D^* = argmax_D V(D,G)$$

2.2 GAN 训练流程

首先，让判别器“学习”一些好的图片，从而知道好的图片的标准是什么。然后开始训练生成器，随便生成一组噪声给生成器，生成器会根据噪声生成一张图片。此时判别器会判断这张图片与真实图片的相似度，如果相似度较低，则需要生成器继续不断地学习，直到判别器无法判断是生成图片还是真实图片。而真实图片与生成图片的差距就是损失，其实质是两种图片在高维空间中概率分布的不同之处，即两个概率分布的 JS 散度（Jensen-Shannon divergnece）。

GAN 的大致训练流程如下：

（1）初始化生成器和判别器，这些参数可以随机生成；

（2）在每一轮的训练中，执行如下步骤（3）、（4）；

（3）固定生成器的参数，训练判别器的参数

a. 因为生成器的参数被固定，此时判别器的参数没有收敛，则生成器通过的未收敛参数生成的图片与真实图片之间的损失会较大。

b. 从准备好的图片数据库中选择一组真实图片数据。

c. 通过上面的操作，此时有了两组数据：一组是生成器生成的图片数据；另一组是真实图片数据。通过这两组数据训练判别器，让其对真实图片赋予高分，给生成图片赋

予低分。

（4）固定判别器的参数，训练生成器的参数

a. 随机生成一组噪声喂给生成器，让生成器生成一张图片。

b. 将生成的图片传入判别器中，判别器会给该图片一个分数，生成器的目标是这个分数更高，生成出判别器可以赋予高分的图片。

2.3 Tensorflow 实现 GAN

在实现过程中，采用的数据集是在校园里人工采集的图片。在朴素 GAN 的网络架构中，生成器的网络结构比较简单，只是一个具有单隐藏层的神经网络，即输入层 → 隐藏层 → 输出层。对于激活函数，目前常见的是 sigmoid 函数、Tanh 函数、ReLU 函数、Leakey ReLU 函数等。在生成器中，我选择了 Leakey ReLU 函数作为隐藏层的激活函数，而输出层则使用 Tahn 函数。判别器与生成器的构造差别不大，只是在输出层使用了 sigmoid 函数作为激活函数。激活函数具体表达式为：

sigmoid 函数：

$$y = \frac{1}{1 + e^{-x}}$$

该函数可以将输入值压缩到 $0 \sim 1$。输入较大的正数时，输出为 1；输入较大的负数时，输出为 0。因此，比较适合作为判别器输出层的激活函数。

Tanh 函数：

$$y = \frac{e^{x} - e{-x}}{e^{x} + e^{-x}}$$

可以看出，Tanh 函数的输出范围是 $-1 \sim 1$, 均值为 0，但计算会比较复杂，不过实验中效果还不错。

Leakey ReLU 函数：

$$f(x_i) = \begin{cases} x_i, & x_i \geq 0 \\ a_i x_i, & x_i < 0. \end{cases}$$

该函数计算比较简单，也避免 ReLU 函数让负半轴恒为 0 的情况。

对于损失函数，本实验采用的是交叉熵损失函数。通俗地讲，交叉熵描述的是一种不确定性，而朴素 GAN 本身就是一种无监督模型，因此损失函数使用交叉熵损失定义比较适合。其在二分类情况下的公式如下：

$$L(Y, f(X)) = -\frac{1}{n}\sum_{i=1}^{n}[y_i(log(f(x_i))) + (1 - y_i)log(1 - f(x_i))]$$

而最小化损失使用的是 AdamOptimizer 方法，其内部是基于梯度下降算法实现的 Adam 算法，可以动态地调整每个参数的学习速率。

使用 tensorflow 实现的训练结果如下：

训练 500 轮结果：

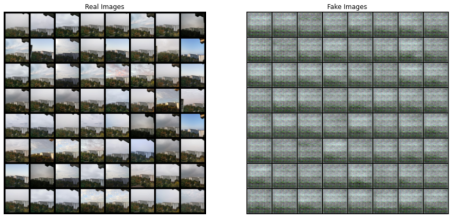

图 4: 500 轮图像结果

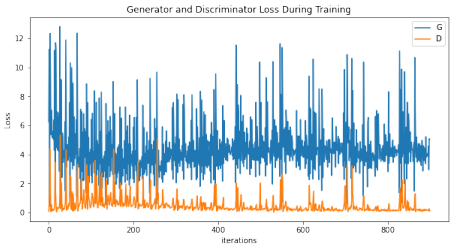

图 5: 500 轮损失变化

训练 2000 轮结果：

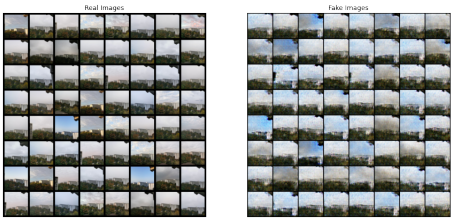

图 6: 2000 轮图像结果

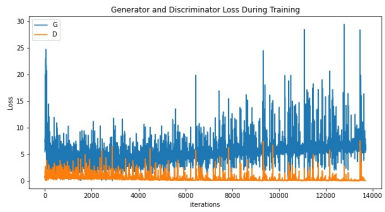

图 7: 2000 轮损失变化

由于在校园中拍摄的图片规范性比较差，且重复率较高。因此我又使用了 summer2winter_yosemite 数据集进行训练，其结果如下：

图 8: summer2winter_yosemite 数据集训练结果

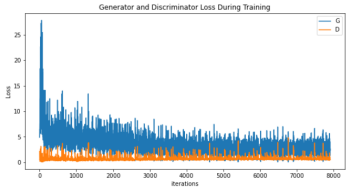

图 9: summer2winter_yosemite 训练时损失变化

通过对比可以明显看出，训练次数和数据集对训练结果有着明显的影响。在使用 summer2winter_yosemite 数据集后，其结果的清晰度和真实度也大大提高了。另外，对比生成器和判别器中的损失变化：随着训练次数的增加，损失值会慢慢趋于稳定，而一个好的数据集也能使损失降得更快、更低。这是一种很常见的技术，通过修改数据集的特征来增强模型在某方面的能力。除此之外，在图 7 中，为什么生成器的损失不降反升，极不稳定？表面上可能与数据集有关，根本上是 GAN 本身存在梯度下降和模式崩溃的问题。接下来我将从数学上推导这两个明显的问题。

3 浅探 GAN 面临的问题

3.1 梯度消失

由 GAN 的目标公式

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

我们可以得到判别器的最优解为

$$D_G^* = \frac{P_{data}(x)}{P_{data}(x) + P_G(x)}$$

将其带入生成器 G 中，则生成器 G 的最优解为

$$max_D V(D,G) = KL[P_{data}(x) \,||\, \frac{P_{data}(x) + P_z(z)}{2}] + KL[P_z(z) \,||\, \frac{P_{data}(x) + P_z(z)}{2}] - 2log2$$

这里引入了 KL 散度。当生成器 G 最优时，就有 $P_{data}(x) = P_z(z)$，即生成数据分布与真实数据分布相同，此时生成器可以生成以假乱真的数据。引入 JS 散度来简化上述公式Arjovsky et al. (2017)：

$$C(G) = max_D V(D,G) = 2JS[P_{data}(x) \,||\, P_z(z)] - 2log2$$

从公式中可以得出：当两个分布之间没有重叠时,JS 散度恒为 $2log2$，此时生成器的梯度 $C(G)$ 也恒为 0，则无论怎样训练生成器都无法降低其损失，这就是传统 GAN 所面临的梯度消失问题。其本质原因是在最优的情况下，JS 散度无法继续表示真实图像概率分布与生成图像概率分布之间的距离。

3.2 模式崩溃

由上面的推导可知，当判别器 G 最优时，生成器的最优解为

$$E_{x\sim P_{data}(x)}[logD^*(x)] + E_{x\sim P_z(z)}[log(1 - D^*(x))] = 2JS[P_{data}(x) \,||\, P_z(z)] - 2log2$$

引入 KL 散度，并作进一步推导可得：

$$E_{x\sim P_z(z)}[-logD^*(x)] = KL(P_z(z) \,||\, P_{data}(x)) - 2JS(P_{data}(x)||P_z(z)) + 2log2 + E_{x\sim P_{data}(x)}[logD^*(x)]$$

这个公式中右边后两项与生成器 G 无关，故生成器 G 的目标函数为Choi et al. (2017)

$$L_G = KL(P_z(z) \,||\, P_{data}(x)) - 2JS(P_{data}(x) \,||\, P_z(z))$$

从目标函数中可分析出两个问题：

（1）训练生成器的目的是最小化 L_G，即最小化生成数据分布 $P_z(z)$ 与真实数据分布 $P_{data}(x)$ 的 KL 散度，同时又要最大化生成数据分布 $P_z(z)$ 与真实数据分布 $P_{data}(x)$ 的 JS 散度。直观上就是既要增大两分布之间的距离，又要减小两分布之间的距离，这显然是不合理的。由此将导致梯度不稳定，使 GAN 的训练极易震荡。

（2）KL 散度是不对称的，即 $KL(P_z(z) \,||\, P_{data}(x)) \neq KL(P_{data}(x) \,||\, (P_z(z))$。由此会引发两种错误：其一是当 KL 接近 0，生成器的惩罚很小，对应着生成器缺乏多样性的现象；其二是 KL 接近正无穷大时，生成器的惩罚很大，对应着生成器缺乏准确性。这将导致模式崩溃现象，即 GAN 产生了大量类似的图像，其原因就是数据集缺少多样性。

4 结论

这是一个有趣且有用的实验。通过这个实验，我既感受到了人工智能技术所带来的魔幻，也认清了其目前的局限性，但还是为人类的创造力感到敬畏。GAN 并不是真正意义上的从零到一的生成图像，而是通过统计获得图像的概率分布，然后模仿拼凑形成新的图像，这种能力很大程度上取决于数据的广度与深度。

参考文献

ARJOVSKY M, CHINTALA S, BOTTOU L, 2017. Wasserstein gan[J].

CHOI Y, CHOI M, KIM M, et al., 2017. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation[J]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 8789-8797.

DAI J, QI H, XIONG Y, et al., 2017. Deformable convolutional networks[J].

GOODFELLOW I J, POUGET-ABADIE J, MIRZA M, et al., 2014. Generative adversarial networks[J].

LIU S, SUN Y, ZHU D, et al., 2018. Face aging with contextual generative adversarial nets[J].

RADFORD A, METZ L, CHINTALA S, 2015. Unsupervised representation learning with deep convolutional generative adversarial networks[J].

SALIMANS T, GOODFELLOW I, ZAREMBA W, et al., 2016. Improved techniques for training gans[J].

WU S, KAN M, HE Z, et al., 2017. Funnel-structured cascade for multi-view face detection with alignment-awareness[J/OL]. Neurocomputing, 221:138-145. DOI: 10.1016/j.neucom.2016.09.072.

ZHU J Y, PARK T, ISOLA P, et al., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks[J].