用于动漫头像生成的改进 DCGAN

郑美镇 [1)]

[1)](厦门大学　人工智能系, 厦门市　中国 361000)

摘　要　Generative Adversarial Networks (GANs) 是一种通用且优秀的生成模型，其在图像生成任务中得到了广泛的应用。本文记录了我通过动漫脸生成这一任务来学习 GAN 的基础理论、训练技巧、实验设置和评价指标等一系列知识的过程。此外，我还在基线模型（DCGAN）上做了简单的改进，使生成图片的 FID 值得到了一定的改善。

关键词　图像；生成；DCGAN；WGAN-GP；FID

Improved DCGAN for Animation Avatar Generation

Meizhen Zheng[1)]

[1)](Department of Artificial Intelligence, Xiamen University, City Xiamen, China)

Abstract:　Generative Adversarial Networks (GANs) is a general and excellent generative model, which has been widely used in image generation tasks. This paper records the process of learning a series of knowledge of GAN, such as basic theory, training skills, experimental settings and evaluation indicators, through the task of animation face generation. In addition, I also made a simple improvement on the baseline model (DCGAN), so that the FID value of the generated images improve to a certain extent.

Keywords:　Image；Generation；DCGAN；WGAN-GP；FID

1 引言

在图像生成领域，变分自编码器 [1] 和生成对抗网络 [2]（以下简称 GAN）是被研究最多的两类模型。其中 GAN 网络是一种通用的生成式模型，其利用对抗博弈的思想，轮流训练生成网络和判别网络，使二者在相互博弈中分别达到最优。最终生成器能够生成以假乱真的图片，而判别器也具备优秀的分辨真假图片的能力。

在前人工作的基础上，Alec Radford 等人提出了 DCGAN[3]，这是一种将 CNN[4] 引入 GAN 的通用的生成器和判别器架构，遵循其设计原则的网络具备生成高质量图片的能力。

尽管原始的 GAN 具备十分美妙的设计理念，然而它却存在着训练不稳定和模式塌缩等问题。WGAN[5] 将 Wasserstein 距离引入到 GAN 的训练中，解决了上述问题。之后 WGAN-GP[6] 对 WGAN 进行了改进，进一步提高了 GAN 的训练质量和稳定性。

本文沿着以上 GAN 的发展历程，进行了一系列实验来学习和验证 GAN 网络的训练和改进：

- 在一个动漫脸数据集上训练 DCGAN 并以此作为基线模型
- 利用 WGAN-GP 的思想对基线模型进行改进
- 分别测试基线模型和改进模型的 FID 值，验证改进思路的有效性

2 相关工作

2.1 Generative Adversarial Nets

Generative Adversarial Nets（GAN）最早是由 Ian J. Goodfellow 等人于 2014 年 10 月提出的，他的《Generative Adversarial Nets》[2] 可以说是这个领域的开山之作。GAN 受博弈论中的零和博弈启发，将生成问题视作判别器和生成器这两个网络的对抗和博弈：生成器从给定噪声中（一般是指均匀分布或者正态分布）产生合成数据，判别器分辨生成器的的输出和真实数据。前者试图产生更接近真实的数据，相应地，后者试图更完美地分辨真实数据与生成数据。由此，两个网络在对抗中进步，在进步后继续对抗，由生成式网络得的数据也就越来越完美，逼近真实数据，从而可以生成想要得到的数据（图片、序列、视频等）。网络的目标函数可表达如下：

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

整个式子由两项构成。x 表示真实图片，z 表示输入 G 网络的噪声，而 G(z) 表示 G 网络生成的图片。D(x) 表示 D 网络判断真实图片是否真实的概率（因为 x 就是真实的，所以对于 D 来说，这个值越接近 1 越好）。而 D(G(z)) 是 D 网络判断 G 生成的图片的是否真实的概率。

2.2 Deep Convolutional GANs

Alec Radford 和 Luke Metz 在 2016 年提出了 Deep Convolutional GANs（DCGAN）[3]。这是一种有效且通用的生成器和判别器架构，其中包含了一些重要的设计原则：

- 使用指定步长的卷积层代替池化层
- 生成器和判别器中都使用 Batch Normalization[7]
- 移除全连接层
- 生成器除去输出层采用 Tanh 外，全部使用 ReLU[8] 作为激活函数
- 判别器所有层都使用 LeakyReLU[9] 作为激活函数

其网络架构如图 1所示。

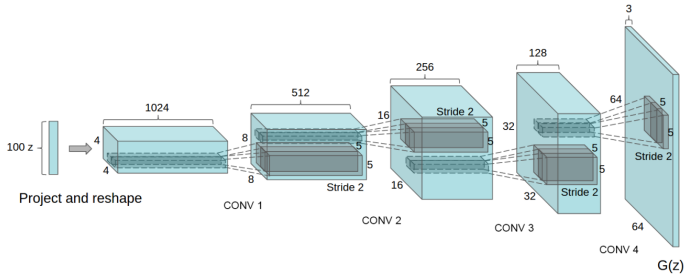

Fig. 1 DCGAN 的架构

2.3 Wasserstein GAN

原始的 GAN 存在着训练困难、生成器和判别器的损失无法指示训练进程、生成样本缺乏多样性等问题。Martin Arjovsky 等人提出了 Wasserstein GAN（WGAN）来解决这些问题。作者先是在这篇文章 [10] 中从数学上论证了在（近似）最优判别器下，最小化生成器的损失等价于最小化 P_r 与 P_g 之间的 JS 散度，而由于 P_r 与 P_g 几乎不可能有不可忽略的重叠，所以无论它们相距多远 JS 散度都是常数 $log2$，最终导致生成器的梯度（近似）为 0，梯度消失。接着作者在第二篇文章 [5] 中提出了一个基于 Wasserstein Distance 的目标函数：

$$W(P_r, P_\theta) = \sup_{\|f\|_L \leq 1} E_{x\sim P_r}[f(x)] - E_{x\sim P_\theta}[f(x)]$$

基于这个函数作者提出了对原始 GAN 的改进：

- 判别器最后一层去掉 sigmoid
- 生成器和判别器的 loss 不取 log
- 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
- 不要用基于动量的优化算法，推荐 RMSProp。

在此之后，这篇文章 [6] 又提出了对于 WGAN 的改进型 WGAN-GP，其目标函数如下：

$$\begin{aligned} L =& E_{\tilde{x}\sim P_g}[D(\tilde{x})] - E_{x\sim P_r}[D(x)] \\ &+ \lambda E_{\hat{x}\sim P_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2] \end{aligned} \tag{1}$$

公式的第二项为梯度惩罚项，其中的 $\hat{x}$ 代表 P_r 与 P_g 的线性插值。作者以此代替 WGAN 中的权重剪裁，避免了 WGAN 中的梯度爆炸和梯度消失问题。并且作者强调判别器网络中不能使用 Batch Normalization。

2.4 Fréchet Inception Distance score

Fréchet Inception Distance score(FID) 是在论文 [11] 中提出的评估生成图像质量的度量标准，专门用于评估生成对抗网络的性能。它的想法是这样的：分别把生成器生成的样本和真实样本送到分类器中（例如 Inception Net-V3 或者其他 CNN 等），抽取分类器的中间层的抽象特征，并假设该抽象特征符合多元高斯分布，估计生成样本高斯分布的均值 μ_g 和方差 Σ_g，以及训练样本 μ_{data} 和方差 Σ_{data}，计算两个高斯分布的弗雷歇距离，此距离值即 FID:

$$d_f = \|\mu_g - \mu_{data}\|_2^2 + tr(\Sigma_{data} + \Sigma_g - 2(\Sigma_{data}\Sigma_g)^{1/2})$$

FID 的数值越小，表示两个高斯分布越接近，GAN 的性能越好。实践中发现，FID 对噪声具有比较好的鲁棒性，能够对生成图像的质量有比较好的评价，其给出的分数与人类的视觉判断比较一致，并且 FID 的计算复杂度并不高。

3 实验

本部分介绍了我基于一个动漫头像数据集进行的一系列用 GAN 生成图片的实验。我先是用 DCGAN 构建了一个基线模型，然后尝试利用 WGAN-GP 的思想改进基线模型，之后对比二者生成的图片的 FID 值来比较它们的性能。

实验代码已放在我的 Github*上。

数据集 实验所使用的数据集是一个动漫少女脸数据集，它包含从网上爬取的总共 71314 张图片，其中每张图片都是 jpg 格式的并且已经标准化成 96×96 的大小。数据集可以在 Google 云盘†下载到。我从数据集中随机抽取 2000 张图片作为验证集，另 2000 张图片作为测试集，剩下的作为训练集。

基线模型 基线模型我参考了这个‡实现。为了与 $tanh$ 函数配合，我首先将图片归一化成 $[-1,1]^{64\times64}$ 的矩阵。生成器的实现参考的是如图 1的结构：先从 100 维的高斯分布中采样出噪声点，然后利用跨步转置卷积进行上采样，最后输出 $[-1,1]^{64\times64}$ 的图片。判别器的实现上利用跨步卷积函数进行下采样，最后一层使用 $Sigmoid$ 函数输出 $[0,1]$[1] 的分数，代表图片是真实的（1）还是虚假的（0）。其中一些网络设置上的技巧参考了§，比如：

- 图片归一化到 $[0,1]$，生成器最后一层使用 $Tanh$
- 从高斯分布上采样噪声数据

*https://github.com/zhengmidon/wdcgan
†https://drive.google.com/uc?id=1IGrTr308mGAaCKotpkkm8wTKlWs9Jq-p
‡https://colab.research.google.com/drive/1JYY_HHtVSSOLixZfLwkxiWTRdPHJCS2t
§https://github.com/soumith/ganhacks

损失函数	BinaryCrossEntropy
批大小	128
学习率	2e-4
噪声维数	100
最大训练轮次	80
优化器	Adam[14]
生成器归一化	BatchNormalization
生成器激活函数	Relu
生成器上采样	ConvTran-Stride
生成器最后激活函数	Tanh
判别器归一化	BatchNormalization
判别器激活函数	LeakyRelu
判别器上采样	Conv-Stride
判别器最后激活函数	Sigmoid

Table 1　基线模型详细配置

损失函数	判别项 + 惩罚项
批大小	128
学习率	2e-4
噪声维数	100
最大训练轮次	80
优化器	RMSProp
生成器归一化	BatchNormalization
生成器激活函数	Relu
生成器上采样	ConvTran-Stride
生成器最后激活函数	Tanh
判别器归一化	InstanceNormalization
判别器激活函数	LeakyRelu
判别器上采样	Conv-Stride
判别器最后激活函数	Sigmoid

Table 2　改进模型详细配置

- 使用 BatchNormalization
- 使用跨步卷积

训练时，给真实图片和生成的图片分别赋予 1 和 0 的标签，然后使用 Binary Cross Entropy 作为损失函数。我在每个 epoch 结束后都在验证集上测试 FID 分数以作为训练早停的依据。FID 的测试利用了网上的开源工具包¶。模型的详细信息已列在表 1上

改进模型　模型改进思路参考了 WGAN-GP 和公式 1。对真实图片和生成图片进行线性插值，把插值后的图片送入判别器然后计算判别器的梯度值，最后计算梯度值相对于 1 的双向二次损失，最后乘以常数项作为惩罚项，将惩罚项加入到判别器的判断损失构成总的损失函数。

网络的结构和基线模型基本相同，除了判别器的归一化函数。论文 [6] 指出，由于梯度惩罚要求对第一张图片分别计算，要提现不同图片的差异，而 BatchNormalization 会消除这种差异，论文作者推荐使用 LayerNormalization[12]，这里我使用的是 InstanceNormalization[13]。而且我发现 Adam 优化器的效果并不好，于是我改用 RMSProp。模型详细配置见表 2

4　实验结果

分别测试基线模型和改进模型的 FID 值，结果列于表 3。发现改进模型获得了不小的性能提升，说明了改进思路的正确性和有效性。使用改进模型生成了一些样例图片，见图 2。

¶https://github.com/mseitzer/pytorch-fid

Fig. 2　改进模型生成图片

模型	FID
基线模型	76.97
改进模型	61.49

Table 3　实验结果

5　结论

本文记录的是我对 GAN 的一些初等认识和入门实验。我先是学习了 GAN 的基本理论和其中对抗博弈的网络训练思想，之后学习了图像 GAN 的一般构建技巧即 DCGAN 的基本结构，认识了 GAN 的一些不足即训练不稳定和模式塌缩的问题，继而学习如何去解决这些问题，即 WGAN 和 WGAN-GP 的思想。实验上，沿着相同的脉络，先用 DCGAN 构建基线模型，然后用 WGAN-GP 的思想对模型进行改进，并测试相应的 FID 值验证改

进获得的性能提升。整体效果不错，收获颇多，但这些都还是入门内容，其中涉及的论文和理论还是多年前的成果，现在的 GAN 已经获得了突飞猛进的发展，所以在未来的工作中我还需要进一步学习更新的理论和模型。

参 考 文 献

[1] Kingma D P, Welling M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.

[2] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in neural information processing systems, 2014, 27.

[3] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

[4] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[5] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein GAN[J]. arXiv preprint arXiv:1701.07875 , 2017.

[6] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[J]. Advances in neural information processing systems, 2017, 30.

[7] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. PMLR, 2015: 448-456.

[8] Glorot X, Bordes A, Bengio Y. Deep sparse rectifier neural networks[C]//Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings, 2011: 315-323.

[9] Maas A L, Hannun A Y, Ng A Y. Rectifier nonlinearities improve neural network acoustic models[C]//Proc. icml. 2013, 30(1): 3.

[10] Arjovsky M, Bottou L. Towards principled methods for training generative adversarial networks[J]. arXiv preprint arXiv:1701.04862, 2017.

[11] Heusel M, Ramsauer H, Unterthiner T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[J]. Advances in neural information processing systems, 2017, 30.

[12] Ba J L, Kiros J R, Hinton G E. Layer normalization[J]. arXiv preprint arXiv:1607.06450, 2016.

[13] Ulyanov D, Vedaldi A, Lempitsky V. Instance normalization: The missing ingredient for fast stylization[J]. arXiv preprint arXiv:1607.08022, 2016.

[14] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

.