

---

# 《人工神经网络》大作业终期报告

---

陶天骅

2017010255

计算机系

tth17@mails.tsinghua.edu.cn

杨雅儒

2017011071

计算机系

yangyr17@mails.tsinghua.edu.cn

## Abstract

本课题尝试构建一个神经网络，用于自动生成尽可能真实的风景图片。我们分别对AutoEncoder、GAN、DCGAN、WGAN、StackGAN等诸多方案进行了尝试和对比，尽可能提高训练的稳定性，并最终在采用DCGAN的网络下得到了较好的结果。

## 1 引言

我们一开始希望构建一个神经网络以及一些简单的界面，可以根据用户提供的一些特征的比例，自动生成一张尽可能真实且清晰的风景图片。首先在利用已有知识的情况下，我们尝试了使用autoEncoder，但是即便是在经过多次调整尝试之后，效果仍然很不理想。在经过文献查阅之后，我们决定尝试使用生成对抗网络（Generative Adversarial Networks, GAN）[1]来解决问题，并且为了提高清晰度使用了stackGAN [4, 8, 9]对已生成的低清晰度图片进行再次对抗生成，以得到更高清的图片。接下来我们还参考了Radford 等与DCGAN相关的工作[5]，向网络中添加了卷积层，并进行了若干优化。

尽管使用DCGAN，加上一些训练技巧并且小心调参的情况下，已经能够得到不错的结果，但训练的稳定性问题一直没有得到解决。一方面，我们尝试了设计一个自适应的算法来自动调整训练过程，在尽可能小的影响训练质量的情况下，矫正训练中出现的过度不均衡的情况；另一方面，我们参考了Arjovsky 等与Wasserstein GAN (WGAN) 相关的工作[11, 12]，并且尝试使用其中的Earth-Mover(EM) 距离，结合之前实现的DCGAN 来解决问题。尽管这样有效提高了训练的稳定性，但由于WGAN 在对每一层的网络参数调整上过于简单地使用weight clipping，导致我们尝试用更深层的网络生成更高清的图片时出现了梯度消失的问题。

最终，由于WGAN 实际产生图片的效果同DCGAN 差不多，而DCGAN 在使用一定训练技巧的情况下已经能够比较稳定地产生较为优质且高清的图片，我们还是决定使用DCGAN 作为最终的网络方案，并且在各个数据集上进行了测试。另外对于我们一开始的目标——用户的比例选择以及界面等，在GAN 的训练难度本身就很高的情况下已经没有足够的时间完成，但我们仍参阅了一些相关的文献[2, 6]，并大致有了一些解决方案。总之，虽然一开始的目标没有完成，但是我们确实已经向它迈进了一大步，并获得了足够的收获和成果。

## 2 相关工作

自从2014 年Ian Goodfellow 等[1] 设计出了起，各种各样的GAN 变种便开始出现，研究者们在不断尝试提高GAN 质量的同时，也尝试着将GAN 应用于各种各样的场合。在应用方面，最常见的一种应用便是生成图像，除了本文中比较直接的生成图像方法以外，还有加入一些条件或风格的生成图像[2, 6] 的CGAN 和StyleGAN 等，也可以用CGAN 来做有监督的图像到图像的生成，而[13, 14] 中的CycleGAN 和DualGAN 还能做到无监督的图像到图像生成，这些可以应用到风格转换等。另外还有例如[4] 中做的文本到图像的生成等。

而在质量提高方面，又可粗略地分为两部分，一是提高训练的稳定性，二是提高生成图像的质量。在稳定性提高上本文主要参考了[11, 12]，在[11] 中Arjovsky 等尝试了研究GAN

训练不稳定的本质原因，即当Discriminator训练过优时将导致Generator的训练出现梯度消失。之后又在[12]中提出了WGAN，使用EM距离替代原本的JS散度，使得在Discriminator训练得更好的情况下，Generator训练的效果也越好。不过正如引言中所述的，WGAN过于简单地使用weight clipping导致出现一些问题，于是Gulrajani等又提出了改进之后的WGAN-GP [15]，使用梯度惩罚来代替简单的weight clipping，实现了更好的效果。在提高生成图像质量上，本文主要考虑到的是图片的清晰度，而StackGAN [4, 8, 9]的相关工作便利用分层的思想，逐步提高生成的图片尺寸，以达到生成高清图片的目的。当然还有一些更加强大的GAN模型，但由于受算力和时间等限制，本课题便没有对其深入研究。

### 3 方法

#### 3.1 训练目标

##### 3.1.1 DCGAN

对于我们使用的GAN、DCGAN以及StackGAN，用D表示判别器（discriminator），G表示生成器（generator），训练目标如下：

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

于是考虑分别构建D、G两部分网络，并且进行轮流训练，在训练D和G时都分别需要固定另一方的网络。

##### 3.1.2 WGAN

根据[11]的推导，式(1)在最优判别器 $D^*$ 下，实际上有：

$$\begin{aligned} & \mathbb{E}_{x \sim p_{data}(x)} [\log D^*(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D^*(G(z)))] \\ &= \mathbb{E}_{x \sim P_r} [\log D^*(x)] + \mathbb{E}_{x \sim P_g} [\log(1 - D^*(x))] \\ &= 2JS(p_{data} || p_{gen}) - 2\log 2 \end{aligned} \quad (2)$$

而Earth-Mover (EM) 距离定义如下：

$$W(P_r, P_g) = \inf_{\gamma \sim \Pi(P_r, P_g)} \mathbb{E}_{(x, y) \sim \gamma} [| | x - y | |] \quad (3)$$

使用EM距离而不是JS散度的优点即，即便两个分布没有重叠，EM距离仍然能够反映它们的远近关系，而JS散度并不能做到这一点。再利用[12]的思路和数学推导，我们便可以得到WGAN具体的网络形式。

#### 3.2 DCGAN 模型

整体分为两部分，Generator部分和Discriminator部分，参数和尺寸在图3.2中标注。

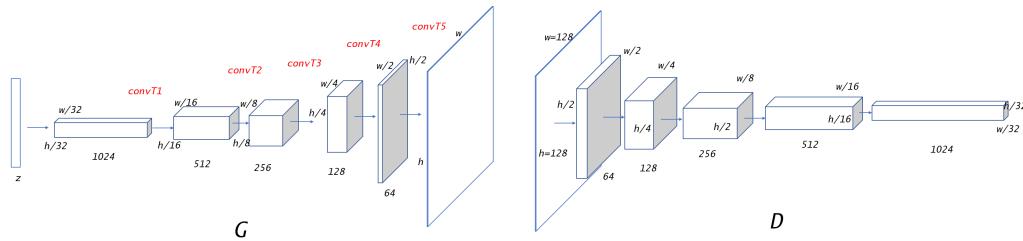


Figure 1: The model of generator in DCGAN. Figure 2: The model of discriminator in DCGAN.

在Generator 中，除了最后一层以tanh 作为激活函数以外，其它层均以relu 作为激活函数；在Discriminator 中，除了最后一层以sigmoid 作为激活函数以外，其它层均以leakyRelu 作为激活函数。最终的loss 采用交叉熵误差函数。

另外在训练过程进行之前，首先需要对数据图片进行值域调整，将其线性调整为[-1, 1] 范围内的float 类型即可。

### 3.3 WGAN 模型

在DCGAN 模型的基础上，为每一层的参数增加了clipping，并且删去了Discriminator 最后的激活函数。另外在每一个conv/deconv层之后加入了BatchNormalization，并在Discriminator 的每一个卷积层激活之后加入了Dropout 层。模型图可见3.3。

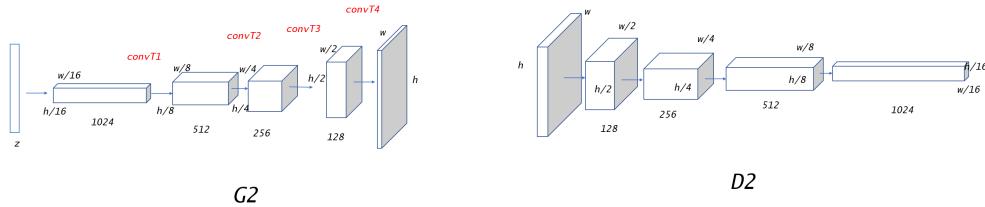


Figure 3: The model of generator in WGAN. Figure 4: The model of discriminator in WGAN.

并且使用如下loss，将其称为Wasserstein Loss：

$$loss = \frac{1}{L} \sum_{i=1}^L y_{label} * y_{pred} \quad (4)$$

其中  $y_{label}$  表示实际是否为真实图片， $y_{pred}$  表示Discriminator 的输出结果。

## 4 实验

由于最终我们的WGAN 并没有在尺寸更大的图片中取得很好的结果，在本课题中仅作为一个稳定性优化方向给出，更具有理论意义。所以为了减少冗余，在实验部分就省去了WGAN 相关的实验信息，而仅以在大尺寸图片中表现仍旧较好的DCGAN 作为最终网络，给出具体实验信息。

### 4.1 数据集

我们使用了六组数据集，详细见下表格，并且对于每一组数据集进行了resize，分为64\*64, 128\*128, 256\*256，分别进行训练。数据大多由我们自行爬取和整理，可以在如下链接中下载：cloud.tsinghua.edu.cn/d/c00d7f1e66914948937e/。

| Name     | landscape | mountains | birds | lake  | dog   | cat   |
|----------|-----------|-----------|-------|-------|-------|-------|
| Pictures | 61132     | 14090     | 11788 | 13840 | 12462 | 12467 |

Table 1: The name and number of pictures of the datasets.

图片4.1 展示出了部分数据集的一些例子。

### 4.2 参数设置

参数设置在4.2 中给出，其中output height 和output width 中的64/128/256 表示任选其一，会直接导致生成图片的尺寸不同。

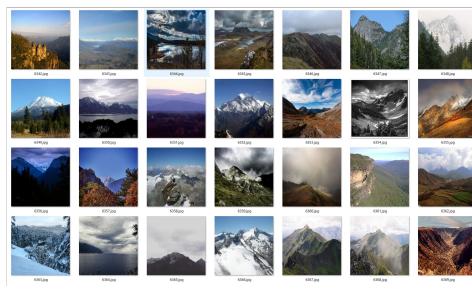


Figure 5: Some pictures in mountains datasets



Figure 6: Some pictures in landscape datasets

| Name          | Type    | Value      | Description                      |
|---------------|---------|------------|----------------------------------|
| input height  | Integer | 256        | The height of images in datasets |
| input width   | Integer | 256        | The width of images in datasets  |
| output height | Integer | 64/128/256 | The height of images generated   |
| output width  | Integer | 64/128/256 | The width of images generated    |
| beta          | Float   | 0.5        | Momentum                         |
| z_dim         | Integer | 100        | The number of noise's dimensions |
| batch size    | Integer | 64         | The size of each batch           |
| learning rate | float   | 0.0002     | Learning rate                    |

Table 2: The details of parameters.

### 4.3 baseline模型

我们以文献[1] 中的GAN 作为baseline。

在[1] 中，作者用GAN实现了一些简单的图片生成工作，有基于全连接的和基于卷积操作的。

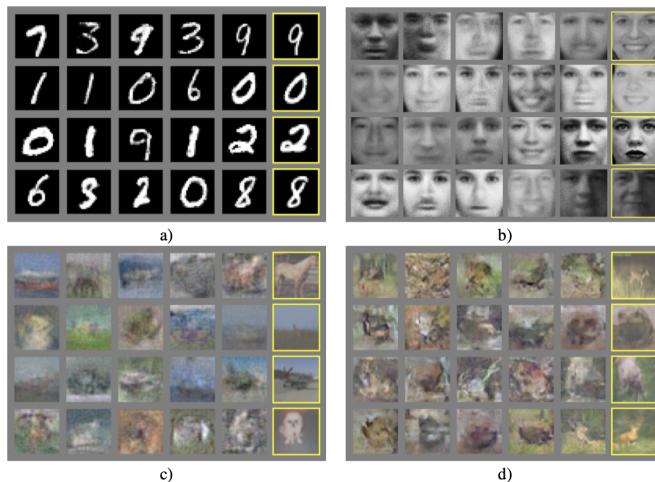


Figure 7: Figure in [1]

我们希望生成较高清一些的图， 分辨率达到128x128或者256x256， 同时所使用的计算资源尽量少。

#### 4.4 实验结果与分析

我们在mountians数据集上做了比较完整的实验，生成了较为清晰的64/128/256分辨率的山的图片。

但是在landscape数据集上训练的结果则没有前者好，这可能是因为数据集的内容比较杂、数量大，还混入了一些干扰图片。



Figure 8: mountains, 64x64, 4convT, 105400 iterations



Figure 9: mountains, 128x128, 5convT, 17200 iterations

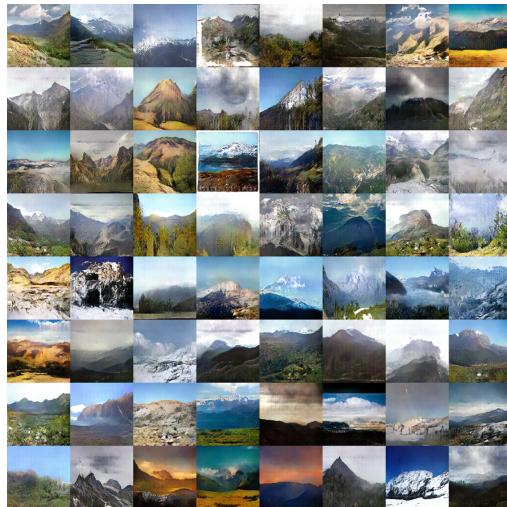


Figure 10: mountains, 256x256, 5convT, 28800 iterations



Figure 11: landscape, 64x64, 5convT, 38000 iterations

我们的模型是通用的，不只是可以生成风景图片，也可以在任意的数据集上训练。我们尝试了一些其他数据集，但训练时间都比较短，且只使用了有4个convT的模型。

我们在训练的模型上测试了Inception Score，结果如4.4所示。测试都是使用了5xConvT的模型，一些没有测的用/表示。

mountians数据集上是几个中最好的，模型的Inception Score与原数据集上的比较接近。

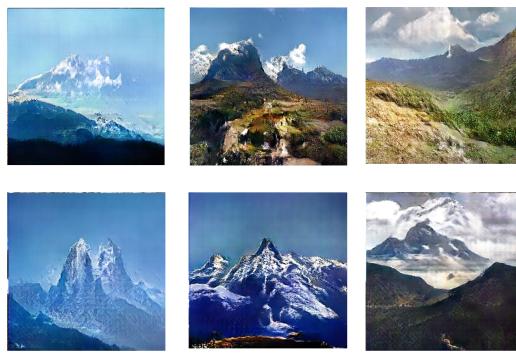


Figure 12: some details



Figure 13: some details



Figure 14: birds, 64x64, 4convT



Figure 15: dog, 64x64, 4convT



Figure 16: cat, 64x64, 4convT



Figure 17: lake, 64x64, 5convT

|           | datasets     | 5xConvT-64x64 | 5xConvT-128x128 | 5xConvT-256x256 |
|-----------|--------------|---------------|-----------------|-----------------|
| landscape | 5.07 ± 0.51  | 2.99 ± 0.04   | 3.02 ± 0.06     | /               |
| mountains | 2.37 ± 0.17  | 2.24 ± 0.01   | 2.19 ± 0.01     | 1.99 ± 0.02     |
| lake      | 2.977 ± 0.25 | 2.04 ± 0.02   | /               | /               |
| dog       | 12.58 ± 0.19 | 4.54 ± 0.10   | /               | /               |
| cat       | 3.62 ± 0.09  | 2.69 ± 0.04   | /               | /               |

Table 3: Inception Score.

## 5 结论

在本次课题中，经过对不同模型在生成图片上的研究，以及对参数的不断调整和试验，我们现在较为朴素的DCGAN上加上一些训练技巧便已经可以达到较好的水准。

例如在mountians数据集上，生成128x128的图片Inception Score可以达到2.24，与原数据集的2.37比较接近。

在分工上，陶天骅同学在整体上对本次课题进行了方向规划和指导，收集了山、鸟、猫、狗、森林、湖等数据集并进行图片的resize 处理，并对AutoEncoder、朴素GAN、DCGAN、StackGAN 等分别进行了尝试，最终选定基于DCGAN的模型，搭建了各有5个卷积和转置卷积层的模型，并加入了Inception Score 对最终网络进行评价。杨雅儒同学收集了Kaggle 上的风景数据集，并独立进行了对DCGAN 的搭建和调参，设计了一个自适应算法，并尝试对其改进来稳定DCGAN 的训练，在参阅文献发现不稳定的本质原因之后，进行了一定的推导验证并尝试更换训练目标以使用WGAN 来进行训练。文档方面，陶天骅同学撰写了中期报告和展示PPT的大部分内容，杨雅儒同学撰写了结题报告的大部分内容。

## 参考文献

- [1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza , et al. Generative Adversarial Networks[J]. 2014.
- [2] Karras, Tero, Laine, Samuli, Aila, Timo. A Style-Based Generator Architecture for Generative Adversarial Networks[J]. 2019
- [3] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In Proc. NIPS, pages 6626–6637, 2017.
- [4] Zhang H , Xu T , Li H , et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks[J]. 2016.
- [5] A. Radford, L. Metz, S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks[J]. 2016.
- [6] M. Mirza, S. Osindero. Conditional Generative Adversarial Nets[J]. 2014.
- [7] Heusel M , Ramsauer H , Unterthiner T , et al. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium[J]. 2017.
- [8] Zhang H , Xu T , Li H , et al. StackGAN: Text to Photo-realistic Image Synthesis with Stacked Generative Adversarial Networks[J]. 2016.
- [9] Han Z , Tao X , Hongsheng L , et al. StackGAN++: Realistic Image Synthesis with Stacked Generative Adversarial Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018:1-1.
- [10] Diederik P Kingma, Max Welling. Auto-Encoding Variational Bayes[J]. 2014
- [11] Arjovsky M , Bottou, Léon. Towards Principled Methods for Training Generative Adversarial Networks[J]. Stat, 2017.
- [12] Martin Arjovsky, Soumith Chintala, Léon Bottou. Wasserstein GAN[J]. 2017.
- [13] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks[J]. 2018.
- [14] Yi Z, Zhang H, Tan P , et al. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation[J]. 2018

[15] I. Gulrajani, F. Ahmed, M. Arjovsky , et at. Improved Training of Wasserstein GANs[J]. 2017