**附录：浙江大学硕士研究生读书报告模板**



硕 士 研 究 生 读 书 报 告



题目 DCGAN原理分析

作者姓名 王汉臣

作者学号 21851201

指导教师 姚诚伟

学科专业 软件工程

所在学院 软件学院

提交日期 二○ 19 年 1 月

Analysis of DCGAN

A Dissertation Submitted to

Zhejiang University

in partial fulfillment of the requirements for

the degree of

Master of Engineering

Major Subject: Software Engineering

Advisor: YaoChengwei

By

WangHanchen

Zhejiang University, P.R. China

2019

摘要

本文主要的研究对象是DCGAN,文章依次从DCGAN的提出背景、相关技术、体系结构和实现细节等方面进行了研究。着重对DCGAN的构建方法进行了研究，对生成器和鉴别器的具体组成结构进行了分析，并对DCGAN的关键技术环节，如转置卷积操作进行了详细的推导。此外，还对DCGAN这一网络架构的应用场景、性能和提出意义进行了分析，通过在多个图像数据集上的测试，发现将其用于特征提取时，具有很好的效果，而且学习到的特征还具有算数性质，DCGAN网络框架的提出不仅通过提供的一系列网络构造规则，在一定程度上缓解了GAN训练不稳定的问题，还将卷积神经网络成功应用在了非监督学习领域，极大推动了深度学习的发展。

**关键词**：DCGAN，生成对抗网络，卷积神经网络，深度学习

Abstract

The main research object of this paper is DCGAN. This paper studies the background, related technology, architecture and implementation details of DCGAN in turn. The construction method of DCGAN is emphatically studied. The concrete structure of generator and discriminator is analyzed. The key technology of DCGAN, such as transposition convolution operation, is deduced in detail. In addition, the application scenario, performance and significance of DCGAN network architecture are analyzed. Through testing on several image data sets, it is found that DCGAN network framework has good effect in feature extraction, and the learned features also have arithmetic properties. The proposed DCGAN network framework not only alleviates the problem by providing a series of network construction rules, but also alleviates the problem to a certain extent. The problem of unstable GAN training is solved, and the convolutional neural network is successfully applied to unsupervised learning, which greatly promotes the development of deep learning.

**Keywords**: DCGAN, GAN, CNN, deep learning

1. 引言

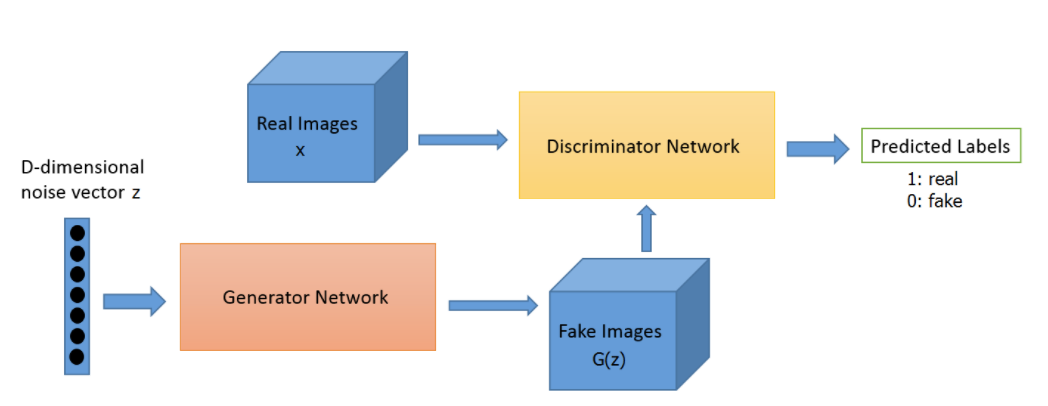
## 背景

DCGAN是将卷积神经网络（CNN）和生成对抗网络相结合的网络结构，DCGAN使得CNN能够在无监督学习领域有了更大的作为。另一方面，在人们将GAN用于特征提取的时候，往往希望能够从大量未标注的数据中学习到有用的特征表达，然后将这些特征应用于监督学习领域，但是GAN的一大问题就是训练不稳定，而DCGAN这种网络结构，使得卷积GAN在训练过程中可以保持稳定[1]。

## DCGAN相关

### GAN

GAN 的核心思想源于博弈论的纳什均衡。即设定参与对抗的双方分别为一个生成器(Generator)和一个判别器(Discriminator), 生成器捕捉真实数据样本的潜在分布, 并生成新的数据样本; 判别器是一个二分类器, 判别输入是真实数据还是生成的样本。为了取得对抗胜利, 生成器和鉴别器需要不断优化, 各自提高自己的生成能力和判别能力, 这个学习优化过程就是寻找二者之间的一个纳什均衡。GAN的计算流程与结构如下图所示：

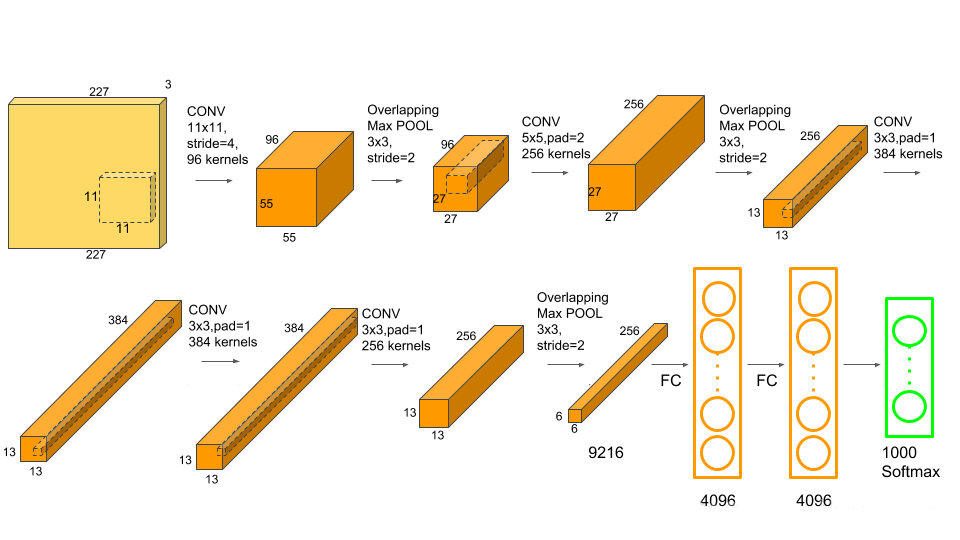


**图 1-1 GAN结构图**

DCGAN出现之前，一般的GAN的输入为随机噪声，生成器和鉴别器的网络结构使用的都是普通的全连接层。GAN在无监督学习领域中的特征提取有很大的潜力[2]，但是GAN的问题在于训练的不稳定性，经常会产生无意义的输出。

### CNN

卷积神经网络一般被用于图像数据的特征提取，CNN的一般结构是卷积层、池化层、激活层的组合，在CNN的末端是全连接层。以CNN经典模型之一——AlexNet为例，就是通过组合卷积层、最大池化层、全连接层再加上softmax实现对图片数据的多分类。其具体结构如下图所示：



**图 1-2 AlexNet计算示意图**

CNN的两大特性：稀疏连接、权值共享使得CNN在处理图像数据时具有较高的效率，因此在图像分类、目标检测领域已经有了较为广泛和成熟的应用。但是也可以发现，CNN的这些成就几乎都是在监督学习领域，对于无标签数据集的分析处理，CNN并没有做出多少贡献。

1. DCGAN原理

## DCGAN简介

[DCGAN](https://arxiv.org/abs/1511.06434)，全称是Deep Convolution Generative Adversarial Networks，即深度卷积生成对抗网络，是由作者Ian J.Goodfellow于2014年提出的一种将GAN和卷积神经网络相结合的一种深度学习网络模型，旨在解决GAN训练不稳定的问题。

## DCGAN体系结构

DCGAN，即 GAN+CNN，但是DCGAN的生成器和鉴别器使用的卷积神经网络相比传统的CNN有一定的调整，核心的工作是对现有的CNN架构做了如下三个方面的修改:

（1）全卷积网络，DCGAN的生成器和鉴别器都采用了这种方法，主要就是使用了strides不为1的卷积操作替代池化操作，从而可以让网络自己学习下采样过程；

（2） 取消全连接层。使用全局平均池化替代全连接层，因为全局平均池化的方式可以提高模型的稳定性，缺点是会降低收敛速度；

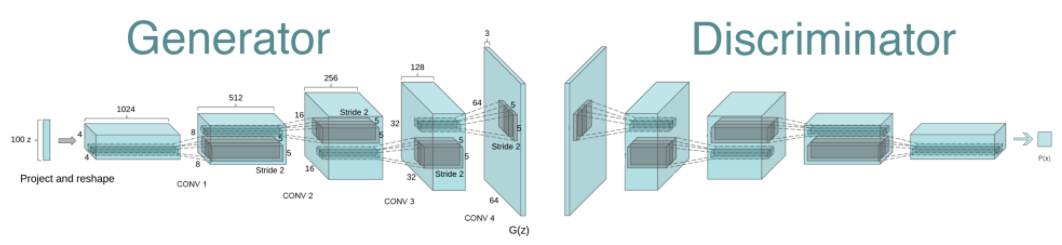
1. 批标准化，即Batch Normalization，对生成器的输出层和鉴别器的输入层使用BN。BN可以加速网络的学习和收敛速度，通过BN将每一层的输入参数变换成均值为0、单位标准差的数据，这样可以减小训练难度，可以让梯度传播到更深的层中，也可以防止生成器将所有的样本变成一个单点；

（4）使用Relu和Leaky Relu激活函数。生成器的输出层使用tanh 激活函数，其余层使用relu 激活函数；鉴别器使用leaky relu激活函数。

## DCGAN实现细节

### 2.3.1 网络结构

结构如下图：



**图2-1 DGAN网络结构图**

DCGAN网络模型的具体组成：

（1）生成器网络：100 z->fc layer->reshape ->deconv+batchNorm+RELU(4) ->tanh 64x64

（2）生成器网络（版本1）：conv+batchNorm+leakyRELU (4) ->reshape -> fc layer 1-> sigmoid

生成器网络使用4层反卷积，而鉴别器网络使用了4层卷积。基本上生成器网络和鉴别器的结构正好是反过来的。鉴别器网络最终的输出是使用sigmoid输出一个0到1之间的单值作为概率，实际上也可以使用softmax，区别在于前者是真的概率值，后者是相对概率，但二者本质上都是一样的。

### 2.3.2 超参数

训练的一些细节如下:

- 图片预处理。没有对图片进行预处理,只将生成器的输出变换到[-1,1]；

- 随机小批量梯度下降法，batch size = 128；

- 参数初始化。所有的参数都采用0均值,标准差为0.02的初始化方式。

- 激活函数。leaky relu，α的取值为0.2；

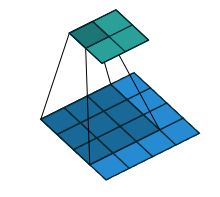
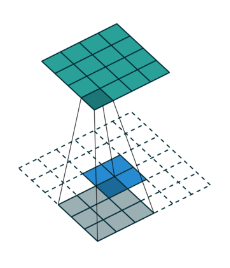
- 优化器。我们使用Adam optimizer,并且对参数做了一些微调,我们实验发现默认的学习率从0.001调整为0.0002。Adam中的 β1=0.9太高了，会使得训练过程震荡，不稳定，我们将其调整为0.5发现可以使训练过程更加稳定[3]。

### 2.3.3 卷积与转置卷积

DCGAN生成器是将随机噪声序列生成图片，采用的主要手段是转置卷积，也称为反卷积。转置卷积是一种上采样的方式，其运算过程和卷积操作恰好是相反的[4]。

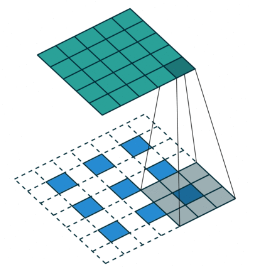
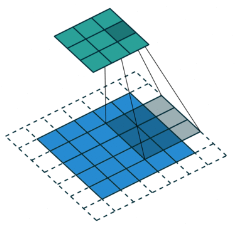
卷积和转置卷积对比如下：

（1）strides=1，kenel\_size=3，no padding

**(a)Convolution (b)Transpose of conv**

**图2-2 卷积操作示意图**

1. strides=2，kenel\_size=3，no padding  
     
   

**(a)Convolution (b)Transpose of conv**

**图2-3 转置卷积操作示意图**

在计算过程中，要将卷积操作、反卷积操作转化为矩阵乘积的形式，例如，将卷积操作转换为矩阵乘积的形式：

一个输入样本A的维度为3\*3，将其reshape为1\*9；

卷积层B的维度为9\*4；  
令A 与B做矩阵的乘法，得到结果 C，C是一个1\*4的向量，再将C reshape为2\*2的矩阵，此时就使用B对样本A进行了卷积，输入数据的维度也由3\*3变成了2\*2，和原卷积操作得到的结果相同。  
 相反的，对于转置卷积操作：  
输入样本A的维度为2\*2，将其维度调整为1\*4； B转置的维度为4\*9；  
 令A与B做矩阵的乘法，得到结果C，C的维度为1\*9，再将其维度调整为3\*3，这样就通过B的转置将维度为2\*2的输入数据 "反卷积"成了3\*3的输出数据。

### 2.3.4 卷积与转置卷积推导

**（1）核对矩阵维度**

首先，将反卷积运算转换成矩阵乘积的形式，卷积层（滤波器）转化为Toeplitz matrix的形式，输入矩阵reshape为列向量。  
 例如：

**卷积操作**

Input = [3，3]，reshape之后，得到A = [1，9]，B = [9，4]，那么A\*B = C = [1，4]，Reshape C = [2，2]，所以，通过B 卷积，我们从shape=[3，3]变成了shape=[2，2]  
 **转置卷积操作**  
 input = [2，2]，reshape之后得到A = [1，4] ，B.T = [4，9]，那么A\*B=C=[1，9]，reshape为[3，3]，此时就通过B的转置"反卷积"将shape=[2，2]的矩阵变换成了shape=[3，3]的矩阵。  
 总结一下就是对于卷积操作，输入feature map A = [3，3]经过了卷积滤波B = [2，2] 输出为 [2，2] ，padding = 0，stride = 1；对于转置卷积，则是输入feature map A = [2，2]，经过了反卷积滤波B = [2，2]，输出为[3，3]，padding = 0，stride = 1。

**（2）反卷积操作具体推导**

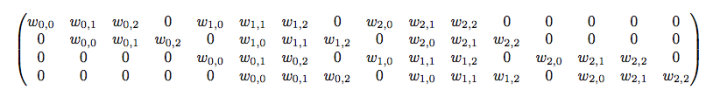
当stride=1时候的转置卷积：no padding对应的是正常卷积操作的时no padding，对应的转置卷积实际上是full padding。而具体padding的值一边拿等于（kernel\_size - stride），在此处就是padding = kernel\_size - stride = 3 - 1 = 2，所以padding添加为2。同样对于下面stride=2的时候，padding = 3 - 2 = 1。

当stride>1的时候，需要在原输入中插入0像素值，如图2-3(b)所显示的，当stride=2的时候，填充padding实际上进行了两个步骤：其一是根据步长stride来填充，即在原输入矩阵中间插入（stride-1）= 2 - 1 = 1个像素值为0的值；其二再根据padding来填充，padding = kernel\_size - stride = 3 - 2 = 1，所以需要在输入外围填充1个像素值为0的值。

**（3）Toeplitz matrix**

假设的输入input = [4，4]，reshape之后是[1，16]，B(滤波器) = [16，4](Toeplitz matrix)，那么A \* B = C = [1,4]。Reshape C=[2，2]所以，通过B 卷积，输入矩阵从shape=[4，4]变成了shape=[2，2]。

B滤波器中的权重值也就是9个值，对应着3×3的卷积核。B滤波器的具体内容如下图所示：



**图2-3 B滤波器对应的[托普利兹](http://www.baidu.com/link?url=RKjJYRQ0re9lgTwgN6PIEucskBel-P9jIXTi8stbOIoMFwP8vFgxbc6JRmdc34ST9O6N419k3eYKp0q6ZSEyh_ZBR-SwvgPh_a3EVy1QaQ4oWLo5bLEwk3KeKZf-TLqPjnpzvphNUF4ZMEizboVOka" \t "https://www.baidu.com/_blank)矩阵**

其实这里卷积核B的参数仍然只有9个，加上多个稀疏值0，来构成一个Toeplitz matrix和输入进行矩阵乘积操作。B的维度之所以会是16×4，是因为输入是[4，4]，有16个值，所以第一维是16，然后因为kernel\_size为3，4×4大小的输入需要计算四次，所以第二维是4。通过验证，一个4×4的输入矩阵和一个3×3的卷积核进行卷积操作与[1，16] \* [16，4]矩阵相乘的计算的结果确实是一样的。

## DCGAN的应用

### 2.4.1 cifar-10

将DCGAN用于特征提取，并对cifar-10数据进行分类。通常评估DCGAN特征学习性能方法是将它作为特征提取器(feature extrator)应用于监督学习，然后再利用学习到的特征构建线性模型对图片数据进行分类，通过线性模型的变现来衡量特征提取效果的好坏。

具体而言，对于cifar-10数据集，k-means模型在非监督特征的提取中有着较好的表现，所以通过来比较一下DCGAN和k-means在同一问题上的性能。为了评估DCGAN在cifar-10上的表现，我们使用imagenet-1k训练DCGAN，然后使用鉴别器的所有卷积特征,对每一层使用最大池化，可以得到一个4x4的网格空间。然后再将再将这些网格空间展开，连接成一个28672维的向量，然后再输入到L2正则化的svm中进行分类。由于DCGAN每一个卷积层得到的最大特征图数量为512个，低于k-means，但是由于DCGAN有多个层，所以总的特征图要多于k-means。最终的分类准确率为82.8%，高于所有基于k-means的方法[5]。由于DCGAN学习得到的特征图并不是通过训练cifar-10数据集得到的，但在cifar-10上却有着较好的变现，充分说明了通过DCGAN得到的图像特征图具有良好的通用性和泛化能力。

### 2.4.2 SVHN

在 StreetView House Numbers dataset(SVHN)数据集中，在带标签数据不足的情况下使用DCGAN的鉴别器进行监督学习。采用和处理cifar-10数据时一样的特征提取方法，还是基于这些特征训练一个L2正则化的SVM，最终测试集的错误率为22.48%。通过进一步的实验分析，发现CNN并非是导致错误分类的主要原因，因为在使用纯粹CNN架构时，经过64次的随机超参数搜索，最终得到的错误率是28.87%，要远高于之前的22.48%[6]。

### 2.4.3 在人脸样本中的向量算术性质

2013年那篇著名的(google word2vec)评估单词学习到的表达文章中，揭示了单词在表达空间中满足简单的算术操作，这表明其具有良好的线性性质。一个经典的例子是vector(”King”) - vector(”Man”) + vector(”Woman”)得到的结果，与之最相近的vector是Queen[7]。通过对一系列典型随机序列进行实验，结果发现确实有类似的性质。实验还表明脸部姿态在Z空间中也满足类似的线性性质。

3.总结

DCGAN是一个更加稳定的GAN模型,而且实验表明这种对抗网络可以学习到无论是对监督学习还是生成模型都都非常好的特征表达。但是同时也注意到模型仍然有一些不稳定的地方，比如随着训练时间的增加，有些filters就会崩溃发生震荡。未来仍需要做一些工作来改善这种不稳定性。但是仍然可以将DCGAN应用到其他领域，比如视频的帧预测，音频领域语音合成的特征预训练等。当然，目前已经在DCGAN的基础上发展出了性能更优的网络模型，值得进一步去学习了解。

参考文献

[1]Francois Chollet.[Deep](https://book.douban.com/search/Ethem Alpaydin) Learning with Python[M].Greenwich，Manning Publications，2017-10-31.

[2]周志华.机器学习[M].北京，清华大学出版社，2016-1-1.

[3]Lan Goodfellow,Yoshua Bengio,Aaron Courville.[Deep](https://book.douban.com/search/Ethem Alpaydin) Learning[M].Massachusetts，The MIT Press，2016-11-11.

[4][邓力](https://baike.baidu.com/item/%E9%82%93%E5%8A%9B" \t "https://baike.baidu.com/item/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%EF%BC%9A%E6%96%B9%E6%B3%95%E5%8F%8A%E5%BA%94%E7%94%A8/_blank),[俞栋](https://baike.baidu.com/item/%E4%BF%9E%E6%A0%8B" \t "https://baike.baidu.com/item/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0%EF%BC%9A%E6%96%B9%E6%B3%95%E5%8F%8A%E5%BA%94%E7%94%A8/_blank) .深度学习：方法及应用[M].北京，机械工业出版社，2016-3-1.

[5]Jan Erik Solem.Python计算机视觉编程[M].北京，人民邮电出版社，2014-08.

[6][Shai Shalev Shwartz](https://book.douban.com/search/Shai Shalev  Shwartz),[Shai Ben David](https://book.douban.com/search/Shai Ben David).深入理解机器学习：从原理到算法[M].北京，机械工业出版社，2016-7.

[7]Willi Richert,Luis Pedro Coelho.机器学习系统设计[M].北京，人民邮电出版社，2014-07.