

基于 DCGAN 的人脸生成模型

喻 东 周 丽

四川大学锦城学院 计算机与软件学院 四川 成都 611731

DOI: 10.18686/jsjxt.v1i3.1257

【摘要】在计算机硬件技术取得突破性进展后,深度学习无疑成为学习界的最大的赢家。各种深度神经网络在计算机视觉、语音处理、自然语言处理领域取得突破性进展。本文所讲的 DCGAN 在结合了 CNN 和传统的 GAN 后,在处理无监督学习问题上有了很大的突破。本文使用了 Python 语言在 TensorFlow 深度学习框架上对原始 DCGAN 进行训练,并且使用了现在深度学习中常用的优化网络的方法,最后可以生成与训练样本具有相同特征的人脸图片。

【关键词】卷积神经网络生成对抗网络深度卷积生成对抗网络人脸生成

1 引言

随着深度学习的兴起,GAN 在图像处理、文本生成领域应用广泛,并且日趋成熟。近几年来,DCGAN 在超分辨率、图像修复、图像生成方面大展拳脚,效果显著,由此可以看出 DCGAN 是在计算机视觉领域内处理无监督问题的一个重要深度神经网络。本文将结合一些人脸数据对该网络模型进行训练和优化,最终得到由噪声通过模型生成的图片。

2 浅谈 GAN 与 DCGAN

2.1 GAN

生成式对抗网络,其核心思想就是从大量的训练样本中学习原始数据所对应的特征分布,再根据特征分布模型利用设定的噪声数据来生成符合该特征分布的新样本,所以可以用 GAN 来扩充我们的样本库。GAN 包括两个子网络模型,生成模型和判别模型,两个模型相互竞争,最终可以训练出一个性能强大的生成器^[1]。

2.2 DCGAN

深度卷积生成对抗网络,全称是 Deep Convolution Generative Adversarial Networks,是在 2014 年 Ian Goodfellow 提出的 GAN 的基础上引入了卷积网络结构后,新提出的一种深度网络,以解决 GAN 训练不稳定的问题^[2]。在这之前,CNN 在监督学习的领域取得了很大的成就,比如图片分类、目标检测,但是在无监督学习领域却没有较大的进展。所以 DCGAN 的作者将 CNN 与 GAN 结合起来处理

无监督学习问题,经过试验并取得了很好的成绩。本文将通过实现一个人脸图片生成的模型来复现 DCGAN 网络的整个工作流程,并对传原始的网络加以优化,展现它在处理无监督学习问题上的优势。在关于深度学习的研究中获得的网络优化技巧也将应用到 DCGAN 网络中去,期望从一些方面去优化网络的学习速率和提取特征的能力。

3 模型目标

在现实生活中,我们期望计算机可以根据我们的意思去完成一些事情,比如我们想要一张关于花的图片,那么计算机就可以迅速生成一张花的图片,这就是生成网络的基本目标。本论文将训练一个 DCGAN 模型,让其可以生成一张张人脸图像,这些图像都是在现实中不存在的,是根据训练数据的特征来生成的。

4 数据准备

4.1 数据采集

本文的模型中有一部分采用卷积网络实现,因为卷积网络对于图像数据有很强的特征提取能力。所以本文收集了大量的人脸图片数据来对卷积网络进行训练。这些数据均来自于网络上的人脸数据库。

4.2 数据处理

在收集好数据之后,对数据进行预处理。首先使用 OpenCV 和基于 Harr 特征的级联算法检测图像中的人脸,截取人脸部分保存,将没有人脸的图片

剔除;然后统一修改图片的尺寸大小为 $64 \times 64 \times 3$ 。(3表示图片的通道);最后按目录结构存储图片数据。

4.3 数据读取

本文基于 TensorFlow 框架搭建的网络结构,训练数据必须以 tensor 的数据形式输入到网络中。为了综合训练的速率和特征的学习能力,本文使用 OpenCV 读取所有的图片,按批次输入到网络中。

5 网络结构解读

DCGAN 由两个功能模块构成,生成器模块和判别器模块。生成器的作用来讲是利用输入的噪声 $Z(x)$ 来生成图片 $G(x)$ 。判别器的作用就是对输入的图片 x 和 $G(x)$ 进行识别,最后得出该图片是真实图片还是由生成器生成的假图片。比如本文从训练数据里面抽出一张人脸图片放到判别器,判别器应返回该图片是真实图片的结果。相反,如果将生成器生成的假图片放到判别器中,此时判别器应该返回该图片是假图片的结果。当生成器和判别器训练到一定程度时,判别器输出值在 0.5 左右,此时说明生成器已足够优秀,生成的图片在特征分布上与真实图片保持高度一致,所以判别器无法判定该图片是否是真实的图片。

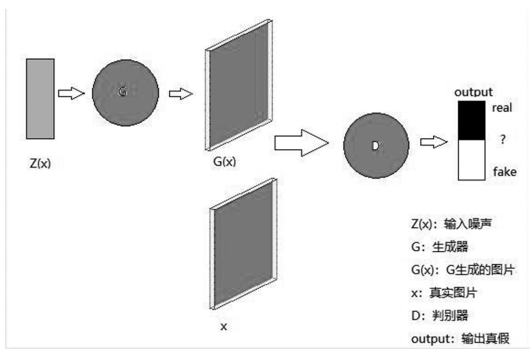


图 1 DCGAN 网络结构图

5.1 生成器

在本文中,生成器的输入是服从均匀分布下的值在 $[-1, 1]$ 的随机噪声 $Z(x)$,假定噪声的维度为 $[None, 100]$ 。在本模型中,采用与 CNN 中卷积方式不同的转置卷积操作,一步一步将噪声转化为一张张特征图,最后得到一张 $64 \times 64 \times 3$ 的彩色图片。为了保证整个网络模型的稳定性,本文采用了输出层、转置卷积(也叫微步卷积)的网络结构来处理噪声到图片的转换,这样的目的是每次卷积后图像尺寸加倍,通道数减半^[3]。

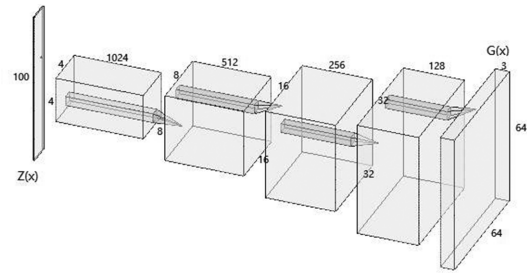


图 2 生成器 G 结构图

5.2 判别器

判别器的作用是对输入的图片进行识别分类,然后输出分类的结果(输入为真实数据时输出为 1,输入为生成器生成的图片时输出为 0)。所以该部分可以使用传统的卷积神经网络如 LeNet、AlexNet、VGGNet 等经典的网络模型^[4]。但是本文对这些网络有一定的修改,一是该部分的所有池化层被带步长的卷积操作所替代,二是除了输出层,其他网络的激活函数都使用 Leaky ReLU 函数。

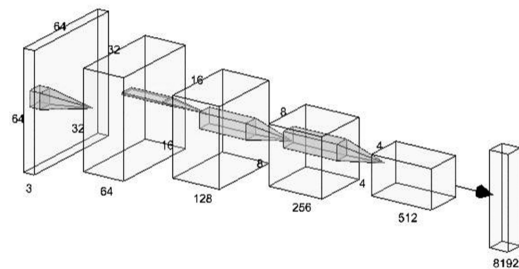


图 3 判别器 D 结构图

6 网络模型优化

在查阅大量资料以及经过多次调参后,本文采用下列所述方法对网络进行优化,以提高模型的稳定性和生成图片的质量。

- a) 所有的判别器的池化层均使用步幅卷积
- b) 在生成器和判别器上均使用批标准化(batch normalization),这个技巧可以解决参数初始化不当引起训练崩溃的问题,但如果将批标准化应用于所有层又会引起模型的不稳定,所以采取的措施为在生成器的输出层和判别器的输入层不使用批标准化。
- c) 移除网络中的全连接层,因为原始 CNN 中大量的参数都早全连接层中,继续使用全连接层虽然可以增加模型的稳定性,但是会大大降低网络的收敛速度,这是非常不好的一件事。
- d) 生成器除输出层外,其余各层外均使用 Relu 函数做非线性映射,只在输出层使用 Tanh 激活函数。

e)在判别器的所有层上使用 LeakyReLU 激活函数。

7 超参数

表 1 超参数表

参数名	参数值
噪声维度	[64,100]
批次大小	64
学习率	0.0 003
迭代次数	100

8 辅助函数

a)读取批量训练数据函数:本文采用小批次数据训练,将数据分为一个个 batch, batch 的大小由我们自己决定,选一个合适的 batch size 可以让网络保持一定的训练速度和稳定性。

b)展示生成图片函数:为了知晓模型优化的进度,本文编写该函数来每隔一定的迭代次数就生成一批噪声数据传到生成器中,去检验此时生成器的图片生成质量,如果每次展示的图片都更加清晰,说明网络正在一步步被优化,这是一件令人兴奋的事情。

c)本文使用交叉熵函数作为损失函数

d)本文采用梯度下降算法训练网络,并使用 AdamOptimizer 优化器

9 图片生成效果



图 4 真实人脸图片

【参考文献】

[1] Goodfellow I J, Pougetabadie J, Mirza M, et al. Generative Adversarial Networks[J]. Advances in Neural Information Processing Systems, 2014, 3:2672-2680

[2] Szegedy C. An Overview of Deep Learning[J]. AITP 2016, 2016.

[3] Liu M Y, Tuzel O. Coupled Generative Adversarial Networks[J]. 2016

[4] Bouvire j. Notes on Convolutional Neural Networks[J]. Neural Nets, 2006.

[5] 焦李成, 赵进, 杨淑媛, 刘芳. 深度学习、优化与识别[M]. 北京:清华大学出版社, 2017.

Epoch 0/100 Discriminator Loss:1.2113 Generator Loss:1.5655

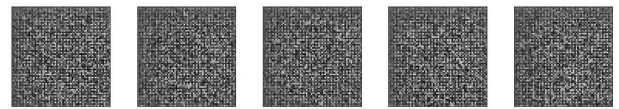


图 5 第 1 次迭代生成人脸图

Epoch 20/100 Discriminator Loss:0.4854 Generator Loss:4.4816



图 6 第 20 次迭代生成人脸图

Epoch 60/100 Discriminator Loss:0.3544 Generator Loss:5.8984



图 7 第 60 次迭代生成人脸图

10 结束语

结合了监督学习和无监督学习的 DCGAN, 打破了传统的深度神经网络的构建思维, 从以前的线性堆叠的方式到卷积与转置卷积结合的双向博弈对抗网络结构, 实现了从样本到特征再到新样本的生成方式, 让我们在神经网络的研究上耳目一新, 大大激发了我们的想象力。当然天下没有免费的午餐, DCGAN 网络也有自己的缺陷, 比如, 我们可以从实验过程生成的损失函数值发现模型迭代的过程不稳定, 损失函数值并不是逐渐减小的, 甚至持续在较大的范围波动, 这也说明了该模型的可解释性差, 模型并不能根据我们设定的梯度下降优化方式去更新模型中的权重, 我们只能在模型的超参数上面下功夫, 以期能得到性能更稳定的模型^[5]。