

一种基于 TensorFlow 的 DCGAN 模型实现

樊雷

(连云港师范高等专科学校, 江苏 连云港 222006)

摘要: 卷积神经网络 CNN 目前在有监督学习领域有着优秀的表现,但在无监督学习领域研究进展缓慢。该文将 CNN 引入到 GAN 中的生成式模型中做无监督训练,利用 CNN 强大的特征提取能力来提高生成式模型的学习效果,采用 TensorFlow 和 Python 代码实现了 DCGAN 中的 D 模型和 G 模型,并在 MNIST 部分数据集下验证了模型生成数字图像效果。实验结果表明采用 DCGAN 可以有效获取图像表征用于分类并具备生成较高分辨率的图像能力。

关键词: DCGAN; 深度学习; TensorFlow; GAN

中图分类号: TP391 文献标识码: A 文章编号: 1009-3044(2017)36-0219-03

DOI: 10.14004/j.cnki.ckt.2017.4017

Implementation of DCGAN Model Based on TensorFlow

FAN Lei

(Lianyungang Normal College, Lianyungang 222006, China)

Abstract: Convolutional neural network (CNN) has excellent performance in supervised learning field, but the progress of research is slow in unsupervised learning. This paper introduced CNN into the generative model of GAN in unsupervised training, to improve the ability to extract generative model learning effect by the powerful character of CNN, using TensorFlow and Python code to achieve the D model and G model within DCGAN, and validates the model to generate digital image effect by the dataset of MNIST. Experimental results show that DCGAN can effectively obtain image representation for classification, and has the ability to generate high resolution of images.

Key words: DCGAN; Deep learning; TensorFlow; GAN

1 TensorFlow 概述

TensorFlow 是 Google 公司在 2015 年 11 月开源的第二代人工智能学习系统,具备了高层次的机器学习计算能力,其特点是灵活性强、速度快、扩展性好,支持异种设备分布式计算,特别是对 GPU 分布式计算具备良好的支持,能够在不同异种平台上实现分散节点自动运行各种目前流行的深度学习模型,可以构建从单个 CPU 或 GPU 到成百上千处理单元组成的分布式系统来加速深度学习模型的训练速度。目前 TensorFlow 已经开

源支持了卷积神经网络 CNN、递归神经网络 RNN 和 LSTM 神经网络等经典深度学习神经网络模型。TensorFlow 提供了 C、C++、Python 编程语言接口,特别是 Python 库更加易学易用,同时提供了大量的内置函数来简化深度学习计算图的构建。图 1 为 TensorFlow 分层架构^[1]。

2 生成式对抗网络 GAN

Christian Szegedy 等人在 ICLR 2014 发表的论文^[2]中,提出了对抗样本(Adversarial examples)的概念,即在输入样本数据集中设计细微的干扰,以合成噪声构造的图像样本数据作为输入,致使多种深度学习模型以高置信度给出了错误的输出结果。实验表明包括卷积神经网络(CNN)在内的各种深度学习模型和其他机器学习模型对于对抗样本都表现出了极低的鲁棒性。生成式对抗网络(GAN)的设计灵感来源于博弈论中的零和博弈,由 Goodfellow 在论文^[3]中创造性地提出。GAN 通过研究深度学习本身对识别对抗样本的缺陷,利用“欺骗”和“反欺骗”的博弈,实现了模型内部的监督学习。GAN 中的博弈模型分别是生成模型 G 和判别模型 D,其模型中使用的 $G(z)$ 和 $D(x)$ 通常为非线性映射函数,输入样本一般来源于多层感知机、卷积神经网络或其他人工智能模型。

GAN 模型的优化过程是一个“二元极小极大”博弈问题:训

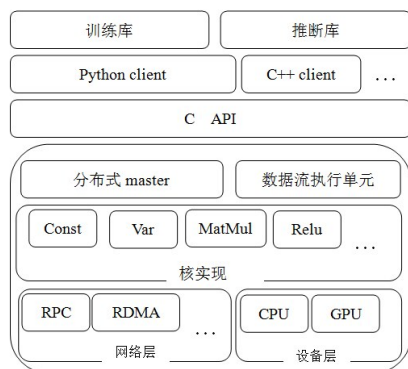


图 1 TensorFlow 分层架构^[1]

收稿日期: 2017-11-09

作者简介: 樊雷(1981—),男,江苏连云港人,连云港师范高等专科学校讲师,研究方向为 Deep Learning。

本栏目责任编辑: 梁书

计算机工程应用技术

训练D期望能最大概率地区分开自然训练样本和G生成样本的标签(最大化 $\log D(x)$), 训练G最小化 $\log(1 - D(G(z)))$ 。实际训练过程中固定一方, 更新另一方参数, 交替迭代判别和生成过程, 使得对方的错误最大化。G模型隐式定义了一个概率分布 P_g , 期望 P_g 收敛到样本真实分布 P_{data} 。Goodfellow在论文^[3]中证明了当且仅当 $P_g = P_{data}$ 时存在最优解, 即达到纳什均衡。^[3]

$E_{x \sim P_{data}}$ 为来自真实数据的期望, $E_{z \sim noise}$ 为来自生成网络的期望。整个网络的价值函数定义为^[3]:

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

3 DCGAN原理和实现方法

DCGAN全称是Deep Convolution GAN, 即采用深度卷积网络对G进行建模。G的输入是一个100维随机噪声向量, 服从均匀分布, 区间为 $[-1, 1]$ 。CNN使用了一个全连接层和4个卷积层, 通过卷积算法将高层表征转换为 $64 * 64 * 3$ 的RGB图片。第一个卷积层采用全连接 $100 \rightarrow 1024$, 并将1024的一维向量重塑成1024个通道的 $4 * 4$ 的特征图。G的后面三层采用反卷积层, 其通道数取半并将图像尺寸增加一倍。

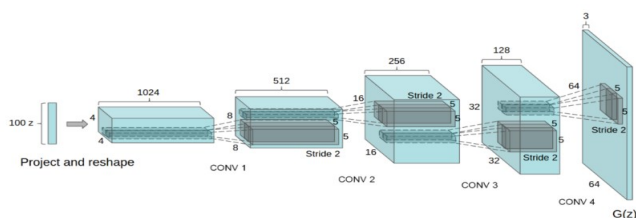


图2 $G(z)$ 结合CNN卷积原理^[4]

判别模型D是一个没有pooling的全卷积网络。D的输入维度和G的输出保持一致, 训练样本的结果输出是一个长度为1的向量, 数字的范围从0到1, 表示输入数据属于训练数据而非生成样本的概率。

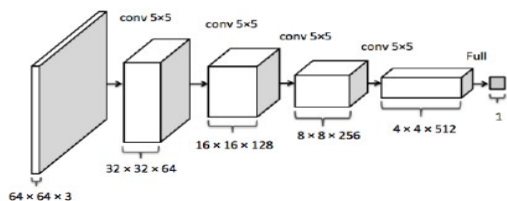


图3 $D(x)$ 转置卷积原理^[5]

以下采用Tensorflow的Python代码实现G和D的部分代码如表1所示。

代码中的linear(线性函数), conv2d_transpose(转置卷积), conv2d(卷积函数)和lrelu(激活函数)均为自定义函数, 篇幅有限, 未能全部列出。

4 DCGAN 图片生成效果

以下利用DCGAN来生成MNIST数字手写体灰度图像, 由于实验条件限制, 硬件采用Intel I7 4770处理器和8G内存对Tensorflow进行了CPU运算, 分别迭代100、1000、2000轮, 3次Epoch后, 选取的部分MNIST数字图像生成效果如图4所示。

图5采用一张花卉图片样本, 对比DCGAN和Autoencoder无监督的学习算法效果。

通过观察可以发现DCGAN能够更为有效、快速地实现生

表1 G和D模型的Python部分代码

G 模型	D 模型
<pre>def gen(self, z): self.z_ = self.conv0_weight, self. conv0_b = linear(z, self.g_d*8*4*4, 'g_conv0_lin', with_weight=True) #全连接层 conv0 self.conv0 = tf.reshape(self.z_, [-1, 4, 4, self.g_d * 8]) conv0 = tf.nn.relu(self.g_bn0(self. conv0)) #反卷积层 conv1 self.conv1, self.conv1_weight, self. conv1_b = conv2d_transpose(conv0, [self.batch_size, 8, 8, self.g_d*4], name= 'g_conv1', with_weight=True) conv1 = tf.nn.relu(self.g_bn1(self. conv1)) #反卷积层 conv2 conv2, self.conv2_weight, self. conv2_b = conv2d_transpose(conv1, [self.batch_size, 16, 16, self.g_d*2], name='g_conv2', with_weight=True) conv2 = tf.nn.relu(self.g_bn2(conv2)) #反卷积层 conv3 conv3, self.conv3_weight, self. conv3_b = conv2d_transpose(conv2, [self.batch_size, 32, 32, self.g_d*1], name='g_conv3', with_weight=True) conv3 = tf.nn.relu(self.g_bn3(conv3)) #反卷积层 conv4 conv4, self.conv4_weight, self. conv4_b = conv2d_transpose(conv3, [self.batch_size, 64, 64, 3], name= 'g_conv4', with_weight=True) return tf.nn.tanh(conv4) #采用 tanh 函数返回生成结果</pre>	<pre>def dsc(self, image): #卷积层 conv0 conv0 = lrelu(conv2d(image, self.d_d, name= 'd_conv0_conv')) #卷积层 conv1 conv1 = lrelu(self.d_bn1 (conv2d(conv0, self.d_d*2, name='d_conv1_conv'))) #卷积层 conv2 conv2 = lrelu(self.d_bn2 (conv2d(conv1, self.d_d*4, name='d_conv2_conv'))) #卷积层 conv3 conv3 = lrelu(self.d_bn3 (conv2d(conv2, self.d_d*8, name='d_conv3_conv'))) #卷积层 conv4 conv4 = linear(tf.reshape (conv3, [-1, 8192]), 1, 'd_conv3_lin') return tf.nn.sigmoid(conv4), conv4 #采用 sigmoid 函数返回判别 结果</pre>

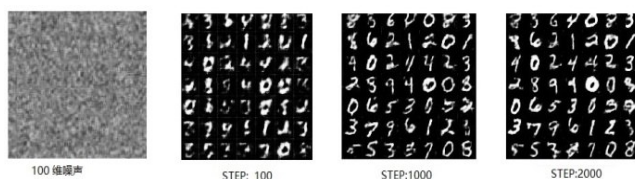


图4 MNIST 数字图像生成效果

成高质量图片。实验还表明通过增大Epoch代数, 当Epoch达到一定次数后, 对生成的图像分辨率提高意义不大。

5 结束语

DCGAN结合了CNN和GAN模型各自的优点, 可以从物体或场景图像数据集中学习到多维深层次表征, 工程实践中训练的结果可以有效的用于生成图像的多维向量表征, 在人工智能图像生成方面具有较高的应用价值。从目前国外学者们的实验效果来看, 在人脸和场景生成等方向上已经取得了突破性的进展。TensorFlow下实现DCGAN可以有效地将学术领域和工

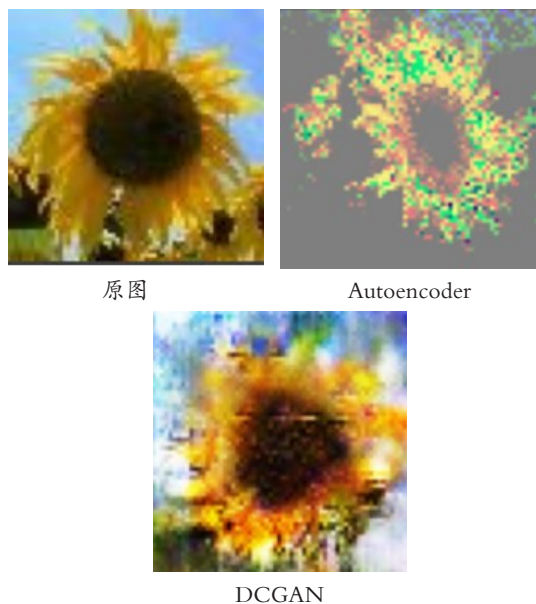


图5 花卉图像算法生成对比效果

程领域结合起来,通过TensorFlow的优秀计算特性、扩展特性可以有效地加速人工智能深度学习各种模型的训练速度,快速

验证模型算法的有效性和鲁棒性,具有广阔的实验、科研以及工程应用价值。

参考文献:

- [1] Abadi M, Barham P. TensorFlow: A system for large-scale machine learning [J]. arXiv preprint arXiv: 1605.08695v2, 2016(5):9-10.
- [2] Szegedy C, Zaremba W, Sutskever I, et al. Intriguing properties of neural networks[J]. arXiv preprint arXiv: 1312.6199v4, 2014(2):1-2.
- [3] Goodfellow Ian, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. arXiv: 1406.2661v1, 2014(6):4-5
- [4] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv: 1511.06434, 2015: 4
- [5] Yeh R, Chen C, Lim T Y, et al. Semantic Image Inpainting with Perceptual and Contextual Losses[J]. arXiv preprint arXiv: arXiv: 1607.07539 v1, 2016: 5.
- [6] Im D J, Kim C D, Jiang H, et al. Generating images with recurrent adversarial networks[J]. arXiv preprint arXiv: 1602.05110, 2016.

(上接第199页)

4 结束语

本文利用加速度计采集的数据,通过对波峰、波谷、步时设置阈值,有效检测行人行走过程中的有效步数,减少行人随意性行走、无规则抖动带来的非正常步带来的影响。通过提出一种自适应步长估计方法,将行人行走过程的步长看作动态,精确估计出行人每一步的步长。方向估计通过对陀螺仪、磁力计计算的方向角进行加权融合,有效降低单一传感器带来的误差,从而提高了定位精度。

参考文献:

- [1] 席瑞,李玉军,侯孟书. 室内定位方法综述[J]. 计算机科学, 2016, 43(4):1-6.
- [2] 周瑞,罗磊,李志强,等. 一种基于智能手机传感器的行人室内定位算法[J]. 计算机工程, 2016, 42(11):22-26.
- [3] 胡安冬,王坚,高井祥. 一种基于地图匹配辅助行人航位推算的室内定位方法[J]. 测绘科学技术学报, 2014(5):529-532.

- [4] 黄承恺. 多传感器信息融合算法在室内定位中的应用[D]. 北京邮电大学, 2015.
- [5] 洪皓,武刚,陈飞翔,等. 基于智能手机传感器的室内行人三维定位算法[J]. 测绘科学, 2016, 41(7):47-52.
- [6] Hsu H H, Peng W J, Shih T K, et al. Smartphone Indoor Localization with Accelerometer and Gyroscope[C]// International Conference on Network-Based Information Systems. IEEE, 2015:465-469.
- [7] Li F, Zhao C, Ding G, et al. A reliable and accurate indoor localization method using phone inertial sensors[C]// ACM Conference on Ubiquitous Computing. ACM, 2012:421-430.
- [8] 李由. 基于MEMS惯性传感器、WiFi、磁场特征的移动智能终端室内行人导航算法[D]. 武汉大学, 2015.
- [9] Qian J, Ma J, Ying R, et al. An improved indoor localization method using smartphone inertial sensors[C]// International Conference on Indoor Positioning and Indoor Navigation. IEEE, 2014:1-7.