doi:10.6041/j.issn.1000-1298.2019.12.026

基于 GAN 网络的菌菇表型数据生成研究

袁培森[1]　吴茂盛[1]　翟肇裕[2]　杨承林[1]　徐焕良[1]

(1.南京农业大学信息科学技术学院,南京 210095;2.马德里理工大学技术工程和电信系统高级学院,马德里 28040)

摘要:生成式对抗网络是基于对抗过程生成数据模型的新框架,它能够生成高质量的图像数据,为解决小样本数据、非均衡数据分析等提供了行之有效的方法。菌菇作为重要的真菌之一,其种类繁多,数据长尾分布、非均衡性等为其表型智能化识别与分类带来了困难。针对蘑菇表型数据,设计了一个高效的蘑菇表型生成式对抗网络 MPGAN。研究了菌菇表型数据生成技术,设计了用于菌菇表型数据生成的生成式对抗网络结构,系统分为模型训练和表型图像生成两个模块。为了提升生成质量,使用 Wasserstein 距离和带有梯度惩罚的损失函数。利用开源数据和私有数据集测试学习率、处理所需的批次数 EPOCH 与 Wasserstein 距离。系统生成的菌菇表型数据为后期菌菇数据分类与识别提供了大数据基础。

关键词:菌菇表型;生成式对抗网络;生成器;判别器;Wasserstein 距离

中图分类号:TP391　　**文献标识码**:A　　**文章编号**:1000-1298(2019)12-0231-09

Mushroom Phenotypic Generation Based on Generative Adversarial Network

YUAN Peisen[1]　WU Maosheng[1]　ZHAI Zhaoyu[2]　YANG Chenglin[1]　XU Huanliang[1]

(1. *College of Information Science and Technology*, *Nanjing Agricultural University*, *Nanjing* 210095, *China*
2. *Superior School of Technical Engineering and Telecommunication Systems*, *Technical University of Madrid*, *Madrid* 28040, *Spain*)

Abstract: Phenotypic data analysis based on image data and machine learning has become one of the important issues in interdisciplinary research. In recent years, the big data and deep learning techniques have provided powerful tools for image analysis and machine vision. Currently, the generative adversarial network is becoming a novel framework for the process estimation generation model. It can generate high-quality image data and provide an effective approach for solving the problem of small sample data and unbalanced data analysis and so on. As one of the important fungi, mushroom has a plenty of varieties and the long tail distribution and non-equilibrium of the data distribution bring great difficulties to its phenotypic intelligent classification and identification. Aiming to design a high-efficiency mushroom phenotype-resistance network MPGAN with mushroom phenotype data. The phenotypic data generation technology of mushroom was studied, and the generated confrontation network structure for mushroom phenotypic data generation was designed. The system was divided into two modules: model training and phenotypic image generation. To improve the quality of the generation, Wasserstein distances and loss functions with gradient penalty were used. Experiments were conducted on two datasets: open source data and private data sets, and results analysis were performed with the learning rate, number of batches required to process EPOCH and Wasserstein distances. The phenotypic data of the mushroom produced with this approach can furnish data basis for the classification of the mushroom data in the later stage, and provide solutions for solving the issues of unbalanced data and long tail distribution of the mushroom classification. The research can provide technical support for the study of high quality mushroom phenotypic data sets.

Key words: mushroom phenotype; generative adversarial network; generator; discriminator; Wasserstein distance

收稿日期:2019-09-18　修回日期:2019-10-19

基金项目:国家自然科学基金项目(61502236、61806097)、中央高校基本科研业务费专项资金项目(KYZ201752)和大学生创新创业训练专项计划项目(S20190025)

作者简介:袁培森(1980—),男,讲师,博士,主要从事智能信息处理、海量数据处理和表型数据分析研究,E-mail:peiseny@njau.edu.cn

通信作者:徐焕良(1963—),男,教授,博士,主要从事农业信息化与大数据技术研究,E-mail:huanliangxu@njau.edu.cn

0 引言

表型(Phenotype)研究核心是获取高质量的性状数据,进而对基因型和环境互作效应(Genotype-by-Environment)进行分析[1-2],表型组学近年来发展迅猛,已成为分子育种和农业应用中的重要技术支撑[3-4]。然而,植物表型数据的获取需搭建实验环境,并需昂贵的数据采集工具,具有周期长、代价高昂等特点[1,5-6]。当前,以大数据为基础的深度学习正在成为表型数据分析的有力工具[7-8],深度学习相关算法的有效性在很大程度上取决于标记样本的数量,因此限制了其在小样本量环境中的应用[9]。数据的非均衡性是生物表型数据具有挑战性的问题[10-13]。

为了提升非均衡数据分析的性能和质量,文献[14-15]提出了数据生成的方法。然而,过采样技术 SMOTE[15]、ADASYN[16]等对于处理经典学习系统中的类不平衡有效,但是此类方法生成的数据不能直接应用于深度学习系统[17]。近年来,生成式对抗网络(Generative adversarial networks, GAN)[18]的出现为计算机视觉应用提供了新的技术和手段,GAN 采用零和博弈与对抗训练的思想生成高质量的样本,具有比传统机器学习算法更强大的特征学习和特征表达能力[19],是一种基于深度学习的学习模型,可以用于海量数据的智能生成,已经广泛用于图像、文本、语音、语言等领域[20-21]。

有学者提出将 GAN 网络技术用于生物学等领域的数据生成问题[9,22-25],结果显示生成数据的质量有显著提高。目前,记录约 8 万种真菌、近 1 500 种野生蘑菇种类的图像数据集,这对种类繁多和分布非均衡的菌类识别和分类具有重要的生态意义[26-28]。

本文提出基于生成对抗网络的菌菇表型数据生成方法(Mushroom phenotypic based on generative adversarial network, MPGAN)。以菌菇表型为研究对象,在特定目标域上训练 GAN 网络,作为 GAN 发生器网络的输入给出潜在模型,以期生成可控制和高质量的蘑菇图像。

1 GAN 网络原理及系统框架

1.1 GAN 网络基本原理

GAN[18]的核心思想来源于博弈论的纳什均衡,它设定双方分别为生成器和判别器,生成器的目的是尽量学习真实的数据分布,而判别器的目的是尽量正确判别输入数据是来自真实数据还是来自生成器。GAN 中的生成器和判别器需要不断优化,各自提高生成能力和判别能力,其学习优化过程就是寻找二者之间的一个纳什均衡[29]。

1.2 GAN 系统框架

GAN 系统一般框架如图 1 所示,系统结构主要包括:生成器(用于生成虚拟图像),它通过接收随机噪声 z,通过这个噪声生成网络 $G(z)$。判别器是负责判断图像真假,输入图像 x,输出对该图像的判别结果 $D(x)$。

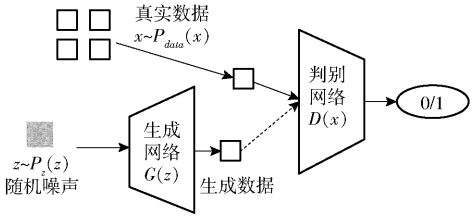

图 1 一般的 GAN 框架

Fig. 1 Framework of GAN

首先,在给定生成器 G 的情况下,最优化判别器 D。采用基于 Sigmoid 的二分类模型的训练方式,判别器 D 的训练是最小化交叉熵的过程,其损失函数表示为

$$O^D(\theta_D,\theta_G) = -\frac{1}{2}E_{x\sim P_{data(x)}}\lg D(x) - \frac{1}{2}E_{z\sim P_z(z)}\lg(1-D(g(z))) \tag{1}$$

式中 x——采样于真实数据分布 $P_{data}(x)$

z——采样于先验分布 $P_z(z)$,例如高斯噪声分布

$E(\cdot)$——计算期望值

式(1)中判别器的训练数据集来源于真实数据集分布 $P_{data}(x)$(标注为 1)和生成器数据分布 $P_g(x)$(标注为 0)。

给定生成器 G,最小化式(1)得到最优解。对于任意的非零实数 m 和 n,且实数值 $y\in[0,1]$,表达式为

$$\Phi = -m\lg y - n\lg(1-y) \tag{2}$$

式(2)在$\frac{m}{m+n}$处得到最小值。因此,给定生成器 G 的情况下,目标函数式(1)最小值为判别器最优解。

$$D_G^*(x) = \frac{P_{data}(x)}{P_{data}(x)+P_g(x)} \tag{3}$$

$D(x)$代表 x 来源于真实数据而非生成数据的概率。当输入数据采样自真实数据 x 时,D 的目标是使得输出概率 $D(x)$趋近于 1,而当输入来自生成数据 $G(z)$时,D 的目标是正确判断数据来源,使得 $D(G(z))$趋近于 0,同时 G 的目标是使得其趋近于 1。生成器 G 损失函数可表示为

$$O^G(\theta_G) = -O^D(\theta_D, \theta_G) \tag{4}$$

其优化问题是一个极值问题，GAN 的目标函数可以描述为

$$\min(G)\max(D)\{f(D,G) = E_{x\sim P_{data}(x)}\lg D(x) + E_{z\sim P_z(z)}\lg(1 - D(G(z)))\} \tag{5}$$

GAN 模型需要训练模型 D 最大化判别数据来源于真实数据或者伪数据分布 $G(z)$ 的准确率，同时，需要训练模型 G 最小化 $\lg(1 - D(G(z)))$。

GAN 学习优化的方法为：先固定生成器 G，优化判别器 D，使得 D 的判别准确率最大化；然后固定判别器 D，优化生成器 G，使得 D 的判别准确率最小化。当且仅当 $P_{data} = P_g$ 时达到全局最优解。

2　MPGAN 系统实现

2.1　MPGAN 系统框架

MPGAN 系统的框架如图 2 所示，蘑菇图像的生成过程为：生成器 $G(z)$ 使用截断到一定范围内的随机正态分布数据作为输入，输入到卷积网络（Convolutional neural network，CNN），最后输出生成图像数据。判别器 $D(x)$ 根据真实图像数据和生成图像数据输出判别结果，并对神经网络的所有参数进行反向更新操作。

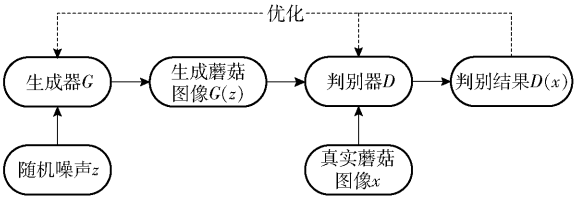

图 2　蘑菇表型数据生成的 MPGAN 框架

Fig. 2　MPGAN framework for mushroom phenotypic data generation

2.1.1　生成器

生成器卷积神经网络结构的作用是通过输入随机数据生成 128 × 128 × 3 的图像，128 表示像素数，3 表示 RGB 的通道数。图 3 是生成器的框架。

生成器采用 8 层的卷积神经网络，首先是 Input 数据输入层，第 2 层是全连接层（Fully connected，FC），然后是连续 5 个反卷积层（Deconvolution，DeConv），其中分为 DC 反卷积层、BN 批归一化层（Batch normalization，BN）和激活函数，批归一化层是对于同一批次数据按照给定的系数进行规范化处理，以防止梯度弥散，最后是 Output 数据输出层。生成器的反卷积层如图 4 所示，各层具体描述如下：

（1）FC 全连接层设计输入为生成 100 个图像的随机数据，经过全连接层的 8 192 个神经元处理以及形状重塑后变为 4 × 4 × 512 大小的数据，再经过批归一化层及 ReLU 激活函数后将结果输出到下一层。

（2）生成器中包括 5 个反卷积层，卷积核的移动步长为 2，卷积核尺寸为 5 × 5，1 ~ 4 层的每一层经过批归一化层及 ReLU 激活函数后将结果输出到下一层，其中：

第 1 层输入数据为 4 × 4 × 512。反卷积层的卷积核数为 256 个，经过反卷积后得到的数据为 8 × 8 × 256。

第 2 层输入数据为 8 × 8 × 256。反卷积层的卷积核数为 128 个，经过反卷积后得到的数据为 16 × 16 × 128。

第 3 层输入数据为 16 × 16 × 128。反卷积层的卷积核数为 64 个，经过反卷积后得到的数据为 32 × 32 × 64。

第 4 层输入数据为 32 × 32 × 64。反卷积层的卷积核数为 32 个，经过反卷积后得到的数据为 64 × 64 × 32。

第 5 层输入数据为 64 × 64 × 32。反卷积层的卷积核数为 3 个。输入数据经过反卷积后得到的数据为 128 × 128 × 3，再经过批归一化层及 tanh 激活函数后将结果输出到下一层。tanh 函数表达式为

$$\tanh(a) = \frac{e^a - e^{-a}}{e^a + e^{-a}} \tag{6}$$

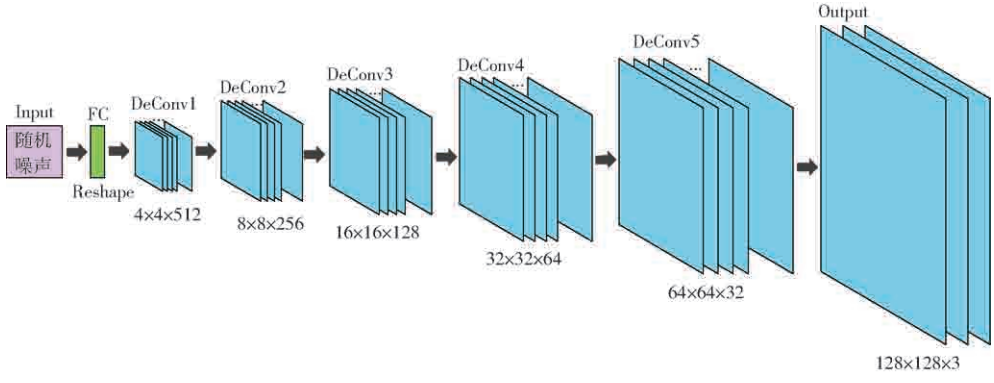

图 3　生成器神经网络框架

Fig. 3　Neural network framework of generator

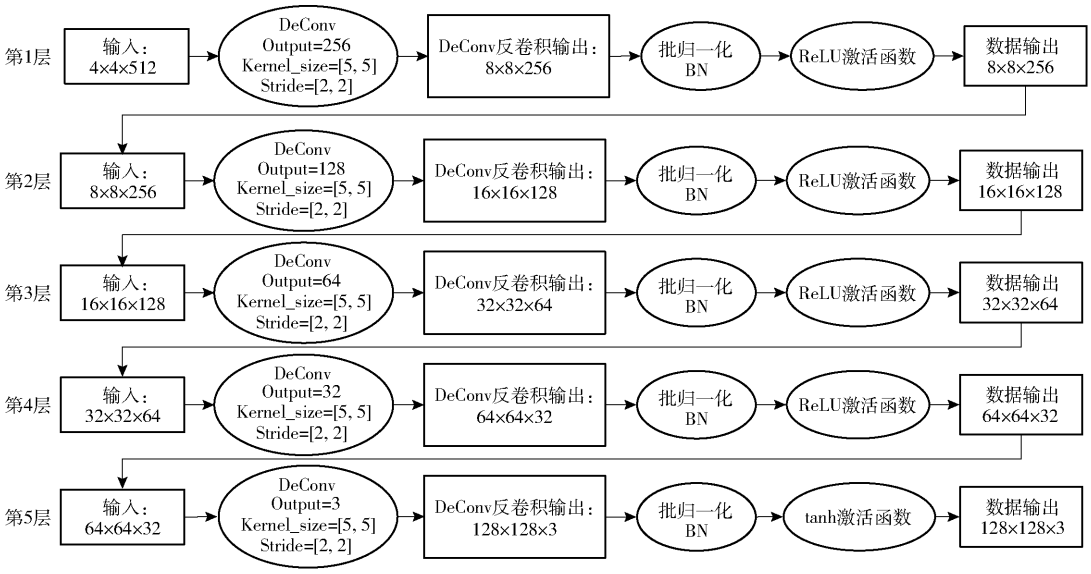

图 4 生成器的反卷积层

Fig. 4 Deconvolution layer of generator

式中 a——参数

不使用传统的 Sigmod 函数进行 Output 输出层，而是直接将上一层输入结果输出。生成器网络参数如表 1 所示。

表 1 生成器网络参数

Tab. 1 Summary of generator network parameters

序号	层	核数	尺寸	输入	输出
0	Input			128×128×3	
1	FC			100	4×4×512
2	DeConv	256	5×5/2	4×4×512	8×8×256
3	DeConv	128	5×5/2	8×8×256	16×16×128
4	DeConv	64	5×5/2	16×16×128	32×32×64
5	DeConv	32	5×5/2	32×32×64	64×64×32
6	DeConv	3	5×5/2	64×64×32	128×128×3
7	Detection				

2.1.2 判别器

判别器的作用是尽量拟合样本之间的 Wasserstein 距离，从而将分类任务转换成回归任务。判别器采用 7 层的卷积神经网络，首先是 Input 数据入层，接着是连续 4 个卷积层（Convolution，Conv），其中分为卷积层、归一化层和激活函数，然后是全连接层 FC，最后是数据输出层 Output。判别器的架构如图 5 所示。

判别器的 Conv 卷积层设计如图 6 所示。判别器共有 4 个卷积层，卷积核的移动步长为 2，卷积核尺寸为 5×5，经过归一化层及 Leaky ReLU 激活函数后将结果输出到下一层。

第 1 层输入数据为 128×128×3。卷积层的卷积核数为 64 个，经过卷积后得到的数据为 64×64×64。

第 2 层输入数据为 64×64×64。卷积层的卷积核数为 128 个，经过卷积后得到的数据为 32×32×128。

第 3 层输入数据为 32×32×128。卷积层的卷积核数为 256 个，经过卷积后得到的数据为 16×16×256。

第 4 层输入数据为 16×16×256。卷积层的卷积核数为 512 个，经过卷积后得到的数据为 8×8×512。

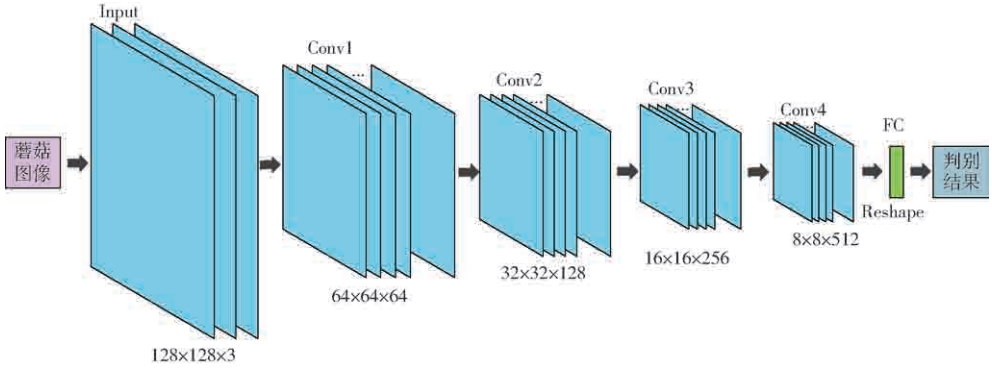

图 5 判别器神经网络框架

Fig. 5 Neural network framework of discriminator

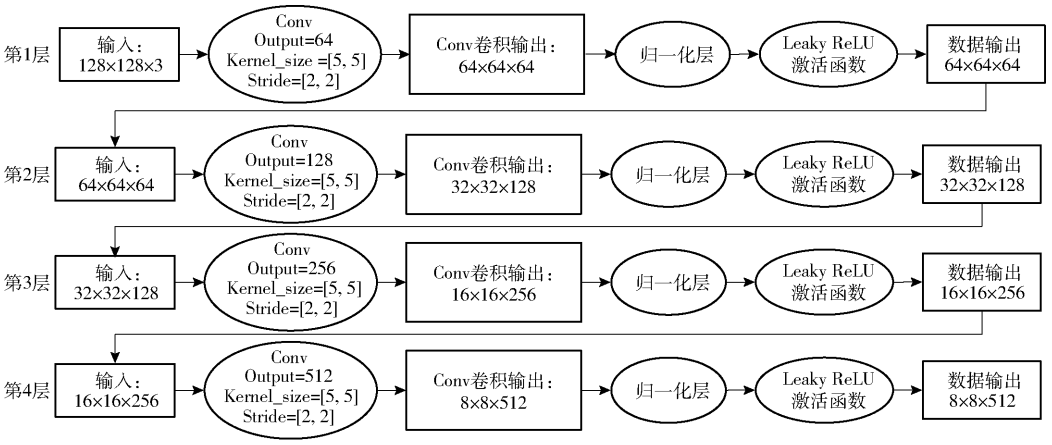

图 6　判别器的卷积层操作

Fig. 6　Convolution layer of discriminator

FC 全连接层设计的输入数据为 8 × 8 × 512，经过全连接层处理以及形状重塑后变为大小为 1 的蘑菇图像，并将结果输出。判别器的网络参数如表 2 所示。

表 2　判别器网络参数

Tab. 2　Summary of discriminator network parameters

序号	层	核数	尺寸	输入	输出
0	Conv	64	5 × 5/2	128 × 128 × 3	64 × 64 × 64
1	Conv	128	5 × 5/2	64 × 64 × 64	32 × 32 × 128
2	Conv	256	5 × 5/2	32 × 32 × 128	16 × 16 × 256
3	Conv	512	5 × 5/2	16 × 16 × 256	8 × 8 × 512
4	FC			8 × 8 × 512	1
5	Detection				

2.2　网络优化设计

2.2.1　Wasserstein 距离

MPGAN 系统采用带有梯度惩罚的 Wasserstein 距离[30]，Wasserstein 距离[9,31-32]又叫推土机(Earth-mover，EM)距离，定义为

$$W(P_r,P_g)=\inf_{\gamma\sim\prod(P_r,P_g)}E_{(r,y)}(\|r-y\|) \quad (7)$$

式中　P_r——真实数据分布

P_g——生成数据分布

r——真实样本

y——生成样本

γ——联合分布

$\prod(P_r,P_g)$——P_r 和 P_g 组合起来的所有可能的联合分布的集合

对于每个可能的联合分布 γ 而言，采样 $(x,y)\sim\gamma$ 得到一个真实样本 x 和一个生成样本 y，并计算这对样本之间的距离 $\|x-y\|$，计算该联合分布 γ 下样本对距离的期望值 $E_{(x,y)\sim\gamma}(\|x-y\|)$。Wasserstein 距离定义为在所有可能的联合分布中能够对这个期望值的下界[31]。

2.2.2　系统损失函数

设定 f_w 代表判别器网络，根据 Lipschitz 连续性条件的要求，该判别器网络含参数 w，并且参数 w 不超过某个范围，根据式(7)定义的 Wasserstein 距离，MPGAN 系统判别器的目的是近似拟合 Wasserstein 距离，因此判别器的损失函数可以表示为

$$L_D=E_{x\sim P_g}(f_w(x))-E_{x\sim P_r}(f_w(x)) \quad (8)$$

MPGAN 系统生成器的目的是近似地最小化 Wasserstein 距离，即最小化式(8)，因此生成器的损失函数可以表示为

$$L_G=E_{x\sim P_r}(f_w(x))-E_{x\sim P_g}(f_w(x)) \quad (9)$$

GULRAJANI 等[30]提出的带有梯度惩罚的 Wasserstein 距离来满足 Lipschitz 连续性。当生成数据分布 P_g 接近真实数据分布 P_r 时，Lipschitz 连续性可表示为

$$\|D(P_g)-D(P_r)\|\leqslant K\|P_g-P_r\| \quad (10)$$

式(10)可转换为

$$\left\|\frac{\partial D(P_c)}{\partial P_c}\right\|\leqslant K \quad (11)$$

式中　P_c——生成数据分布与真实数据分布的差值

K——整数常量

先对真假样本的数据分布进行随机差值采样，即产生一对真假样本 X_r 和 X_g，采样公式为

$$X=\xi X_r+(1-\xi)X_g \quad (12)$$

式中　ξ——[0,1]区间的随机数

使用随机差值采样计算判别器的梯度 $\nabla_x D(x)$，建立与 Lipschitz 常数 K 之间的二范数实现梯度惩罚项，即

$$\Omega=\lambda E_{x\sim P_x}((\|\nabla_x D(x)\|_2-K)^2) \quad (13)$$

式中　λ——调节梯度惩罚项大小的参数

K 为使得 Lipschitz 连续性条件成立的常量，设定 K 为 1，MPGAN 系统的判别器损失函数式(9)和梯度惩罚项式(13)，损失函数可表示为

$$L = E_{x\sim P_r}(f_w(x)) - E_{x\sim P_g}(f_w(x)) + \lambda E_{x\sim P_x}((\| \nabla_x D(x) \|_2 - 1)^2) \quad (14)$$

2.3 MPGAN 系统的训练过程

根据 GAN 网络的框架和优化过程,MPGAN 系统的训练过程如图 7 所示。

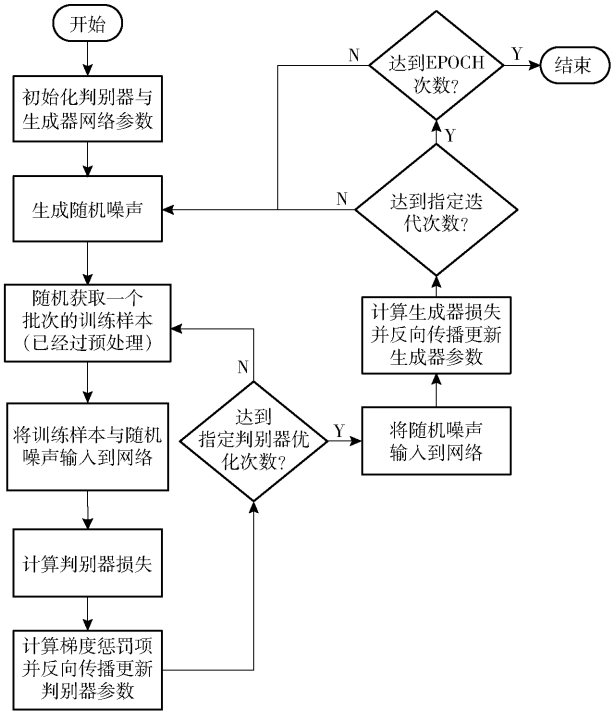

图 7 MPGAN 系统的训练过程

Fig. 7 Training procedure of MPGAN system

图 7 中的训练过程描述如下:

(1)采用方差为 0.02 的截断正态分布初始化网络中的权值参数 W 和卷积核初始化网络的偏置值 b,初始化学习率 η,即每次参数更新幅度。在训练过程中,参数更新向着损失函数梯度下降的方向,表示为

$$W_{n+1} = W_n - \eta\Delta \quad (15)$$

式中 Δ——梯度,即损失函数的导数

(2)采用区间为[-1,1]的均匀分布初始化随机噪声。

(3)采用数据集中随机获取批次大小的训练样本,并在输入队列中进行数据预处理。

(4)将步骤(2)中生成的随机噪声输入到生成器网络,生成虚拟图像数据,将生成的虚拟图像数据输入判别器,得到生成图像判别结果;将步骤(3)中获取的训练样本使用批归一化操作输入判别器,得到真实图像判别结果;计算判别器损失并反向更新判别器参数。

(5)计算梯度惩罚项,为判别器损失施加惩罚,然后使用优化器反向更新判别器参数,使用梯度惩罚项,替换原来的权重截断策略。

(6)判断是否达到指定判别器优化次数,即每优化一次生成器时优化 N 次判别器,若是则进入步骤(7),若否则重新进入步骤(3)。其中 N 由用户设定。

(7)将步骤(2)中生成的随机噪声输入到生成器网络,计算生成器损失并使用优化器反向更新判别器参数。

(8)判断是否达到指定迭代次数,即是否遍历完全部样本,若是则进入步骤(9),否则重新进入步骤(2)。

(9)判断是否达到 EPOCH 次数,EPOCH 为总共训练的轮次,若是则结束,否则重新进入步骤(2)。

3 实验结果与分析

实验平台为 Windows 10 系统,16 GB 内存,256 GB SSD,1 TB HD,Intel QuadCore i7 - 8700,4.2 GHz, Nvidia GTX 1070, 8 GB。算法采用 Tensorflow V1.1 GPU 框架[33]和 Python 3.6 实现。

3.1 数据集

采用两类数据集:开源蘑菇数据集 Fungi[28],选择了其中 375 幅图像;私有数据集,共 138 幅图像。图像预处理方法包括随机翻转、随机亮度变换、随机对比度变换和图像归一化,前面几种预处理方法主要是为了增加样本数量,而图像归一化是为了降低几何变换带来的影响。

图 8 为开源数据集 Fungi 蘑菇示例图像,该数据集环境噪声大且背景复杂,背景中有草地、林地、树叶、木块等多种干扰物。

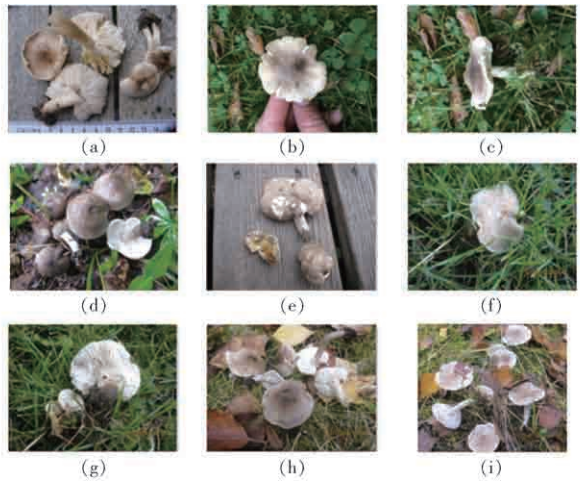

图 8 开源数据集示例

Fig. 8 Examples of public dataset

私有蘑菇数据集采用凤尾菇作为对象,该数据集采用黑色作为背景,背景噪声小,且蘑菇形状不同,适合菌菇表型图像生成。图 9 为私有蘑菇数据集的示例图像。

图 9　私有蘑菇数据集示例

Fig. 9　Examples of private dataset

3.2　参数设置

MPGAN 系统默认使用 Adam 优化器[34]，优化器超参数 $\beta_1 = 0.5$、$\beta_2 = 0.9$、$\varepsilon = 1 \times 10^{-8}$，学习率 η 默认为 0.000 3，判别器优化次数 $N = 5$。

3.2.1　生成器参数设置

由于生成器的输出层直接将前一层的值作为输入，最后激活函数选择 tanh 激活函数，该激活函数可以将输出层的输出约束到区间[-1,1]。

为了保证数据分布的一致性，并防止反向传播权值更新时发生梯度弥散并加速收敛，采用批归一化(Local response normalization)，对同一批次数据按照给定的系数进行规范化处理。其处理步骤如下：

(1)沿通道计算同一批次内所有图像的均值 μ_B，计算式为

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{16}$$

(2)沿通道计算同一批次所有图像的方差 σ_B^2，计算式为

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \tag{17}$$

(3)对图像做归一化处理，计算式为

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \omega}} \tag{18}$$

式中　$\hat{x}_i$——图像归一化处理结果

ω——防止方差为 0 的参数

(4)加入缩放变量 γ 和平移变量 φ，得出结果

$$y_i = \gamma\hat{x}_i + \varphi \equiv BN_{\gamma,\varphi}(x_i) \tag{19}$$

式中　y_i——加入缩放变量 γ 和平移变量 φ 处理结果

3.2.2　判别器参数设置

选择 Leaky ReLU 激活函数作为判别器激活函数，确保梯度更新整个图像。Leaky ReLU 激活函数表达式为

$$y = \begin{cases} x & (x \geqslant 0) \\ \dfrac{x}{\alpha} & (x < 0) \end{cases} \tag{20}$$

式中　α——(1，+∞)区间内的参数

MPGAN 系统生成式对抗网络模型的梯度惩罚策略采用层归一化函数(Layer normalization，LN)。

3.3　Wasserstein 距离与 EPOCH

在学习率 η 为 0.000 3 时，使用开源数据集和私有数据集作为训练数据集，MPGAN 系统的 Wasserstein 距离与 EPOCH 的关系如图 10 所示。

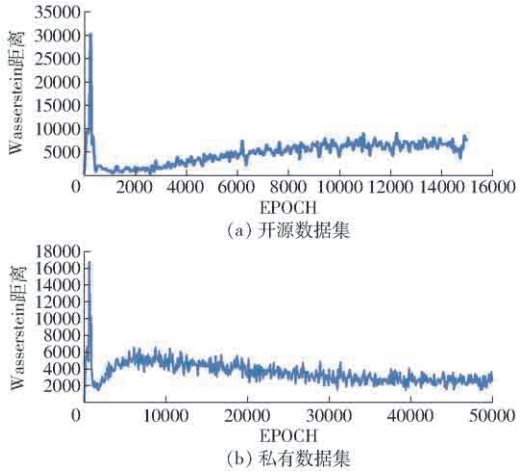

图 10　Wasserstein 距离收敛曲线

Fig. 10　Wasserstein distance convergence curves

由图 10a 可知，在开源数据集，EPOCH 大于 2 000 后逐渐开始学习到真实图像的数据分布，在 EPOCH 达到 10 000 后逐渐趋于稳定，在这个阶段数据集本身噪声较大导致模型的学习能力有所下降，所以模型学习的特征被背景所干扰，并且在曲线尾部的振荡程度明显增大，此时减小学习率 η 可以使模型训练更加稳定。

由图 10b 可知，Wasserstein 距离在 EPOCH 达到 2 000 后不断收敛，在 10 000 左右有小幅振荡，EPOCH 在超过 35 000 之后，振荡幅度减小，模型比较稳定。

由图 10 可知，不同数据集训练的 EPOCH 次数不同，开源数据集的噪声较大，模型不容易收敛，并且相似度衡量指标 Wasserstein 距离在 EPOCH 为 12 000 时开始稳定在一个较高的程度；私有数据集上的噪声较小，当在该数据集，模型收敛更加快速，Wasserstein 距离在 EPOCH 大于 35 000 时开始逐渐收敛稳定。

3.4　学习率与 EPOCH

基于开源数据集的学习率与 EPOCH 关系如图 11 所示。从图 11 可看出，提高学习率 η 时，模型的收敛速度有明显的提升并在 EPOCH 为 1 000 后逐渐稳定，但是随着学习率的提高，收敛的振荡程度

也在加大,因此可以在训练初期使用较大的学习率提高初始收敛速度,然后逐渐减小学习率保证训练过程稳定。由于在私有数据集上的结果类似,因此仅报告了开源数据集上的测试结果。

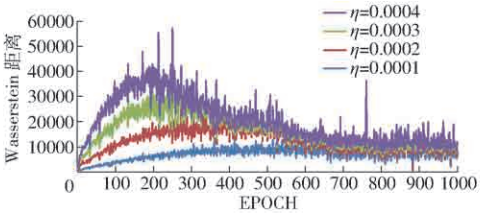

图 11 基于开源数据集的学习率与 EPOCH 关系

Fig. 11 Learning rate and EPOCH relationship based on open source dataset

3.5 蘑菇图像生成

首先,系统测试了数据中的 scalpturatum 口蘑,EPOCH 为 1 000 时,学习率 η 为 0.000 1 ~ 0.000 5 生成图像如图 12 所示。图 12a 为原始图像,从图 12b 可看出,学习率 η 为 0.000 3 时,生成的菌菇图像相对较好。

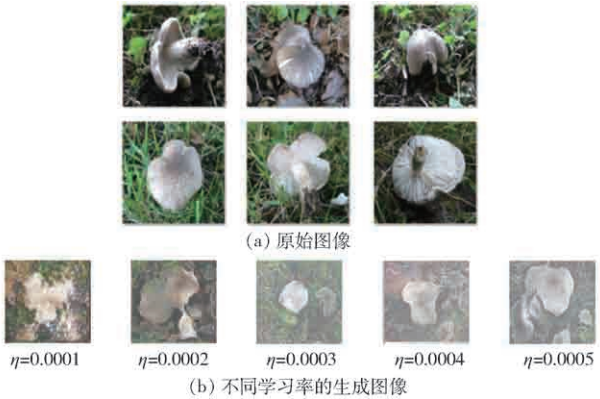

图 12 不同学习率的菌菇图像生成结果对比

Fig. 12 Mushroom image generation results comparison at different learning rates

当学习率 η 为 0.000 3 时,在开源数据集和私有数据集上,测试了系统菌菇图像生成结果,生成图像尺寸设置为 64 像素 × 64 像素,结果分别如图 13 和图 14 所示。图 13 为 EPOCH 为 15 000 时,开源数据集上的生成结果。图 13b 的生成图像能够清晰地显示出原始菌菇的表型特征。

图 14 为 EPOCH 为 50 000 时,私有数据集上的生成结果。图 14b 的生成图像能够清晰地显示出原始菌菇的表型特征。

图 13 基于开源数据集上的蘑菇生成图像

Fig. 13 Illustration of generating Fungi images based on public dataset

图 14 基于私有数据集上的蘑菇生成图像

Fig. 14 Illustration of generating Fungi images based on private dataset

对比图 13b 和图 14b 可以看出,图 14b 质量优于图 13b,表明高质量的菌菇训练数据对图菌菇表型图像的生成有重要影响。

4 结论

(1)研究了菌菇表型数据生成技术,设计了用于菌菇表型数据生成的生成式对抗网络结构。使用 Wasserstein 距离和带有梯度惩罚的损失函数。

(2)利用开源数据和私有数据集进行了测试,结果表明,数据集噪声越小越好,噪声越小则损失越容易收敛,否则背景和主体目标发生混淆时,损失会在一个较大程度上振荡。

(3)测试了学习率 η、EPOCH 与 Wasserstein 距离关系,系统生成的菌菇表型数据可为后期菌菇数据分类与识别提供大数据基础,为解决菌菇分类的数据非均衡、长尾分布等问题提供研究基础。

参 考 文 献

[1] 周济,TARDIEU F, PRIDMORE T,等. 植物表型组学:发展、现状与挑战[J]. 南京农业大学学报,2018, 41(4): 580 - 588.
ZHOU Ji, TARDIEU F, PRIDMORE T, et al. Plant phenomics: history, present status and challenges[J]. Journal of Nanjing Agricultural University, 2018, 41(4): 580 - 588. (in Chinese)

[2] RIBAUT J M, VICENTE M D, DELANNAY X. Molecular breeding in developing countries: challenges and perspectives[J]. Current Opinion in Plant Biology, 2010, 13(2): 213 - 218.

[3] 唐惠燕,倪峰,李小涛,等. 基于 Scopus 的植物表型组学研究进展分析[J]. 南京农业大学学报, 2018, 41(6): 169 - 177.
TANG Huiyan, NI Feng, LI Xiaotao, et al. Analysis of the advance in plant phenomics research based on Scopus tools[J]. Journal of Nanjing Agricultural University, 2018, 41(6): 169 - 177. (in Chinese)

[4] RAHAMAN M, CHEN D, GILLANI Z, et al. Advanced phenotyping and phenotype data analysis for the study of plant growth and development[J]. Frontiers in Plant Science, 2015, 619(6):1 – 15.
[5] HOULE D, GOVINDARAJU D R, OMHOLT S. Phenomics: the next challenge[J]. Nature Reviews Genetics, 2010, 11(12): 855 – 866.
[6] EBERIUS M, LIMA-GUERRA J. High-throughput plant phenotyping-data acquisition, transformation, and analysis[M]// Bioinformatics. New York: Springer, 2009: 259 – 278.
[7] NAMIN S T, ESMAEILZADEH M, NAJAFI M, et al. Deep phenotyping: deep learning for temporal phenotype/genotype classification[J]. Plant Methods, 2018, 14(66): 1 – 14.
[8] SINGH A, GANAPATHYSUBRAMANIAN B, SINGH A K, et al. Machine learning for high-throughput stress phenotyping in plants [J]. Trends in Plant Science, 2016, 21(2): 110 – 124.
[9] LIU Y, ZHOU Y, LIU X, et al. Wasserstein GAN-based small-sample augmentation for new-generation artificial intelligence: a case study of cancer-staging data in biology[J]. Engineering, 2019, 5(1): 156 – 163.
[10] KRAWCZYK B. Learning from imbalanced data: open challenges and future directions[J]. Progress in Artificial Intelligence, 2016, 5(4): 221 – 232.
[11] DEY R, SCHMIDT E M, ABECASIS G R, et al. A fast and accurate algorithm to test for binary phenotypes and its application to PheWAS[J]. The American Journal of Human Genetics, 2017, 101(1): 37 – 49.
[12] FRASCA M, VALENTINI G. COSNet: an R package for label prediction in unbalanced biological networks [J]. Neurocomputing, 2017, 237: 397 – 400.
[13] LADO B, BARRIOS P G, QUINCKE M, et al. Modeling genotype × environment interaction for genomic selection with unbalanced data from a wheat breeding program[J]. Crop Science, 2016, 56(5): 2165 – 2179.
[14] HAIXIANG G, YIJING L, SHANG J, et al. Learning from class-imbalanced data: review of methods and applications[J]. Expert Systems with Applications, 2017, 73: 220 – 239.
[15] CHAWLA N V, BOWYER K W, HALL L O, et al. SMOTE: synthetic minority over-sampling technique[J]. Journal of Artificial Intelligence Research, 2002, 16: 321 – 357.
[16] HE H, BAI Y, GARCIA E A, et al. ADASYN: Adaptive synthetic sampling approach for imbalanced learning[C]// 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence). IEEE, 2008: 1322 – 1328.
[17] MULLICK S S, DATTA S, DAS S. Generative adversarial minority oversampling [J]. arXiv preprint arXiv:1903.09730. 2019.
[18] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[C]// Advances in Neural Information Processing Systems, 2014: 2672 – 2680.
[19] 曹仰杰,贾丽丽,陈永霞,等. 生成式对抗网络及其计算机视觉应用研究综述[J]. 中国图象图形学报,2018, 23(10): 5 – 21.
CAO Yangjie, JIA Lili, CHEN Yongxia, et al. Review of computer vision based on generative adversarial networks[J]. Journal of Image and Graphics, 2018, 23(10): 5 – 21. (in Chinese)
[20] 王坤峰,左旺孟,谭营,等. 生成式对抗网络: 从生成数据到创造智能[J]. 自动化学报, 2018, 44(5): 769 – 774.
WANG Kunfeng, ZUO Wangmeng, TAN Ying, et al. Generative adversarial networks: from generating data to creating intelligence[J]. Acta Automatica Sinica,2018, 44(5): 769 – 774. (in Chinese)
[21] WU X, XU K, HALL P. A survey of image synthesis and editing with generative adversarial networks[J]. Tsinghua Science and Technology, 2017, 22(6): 660 – 674.
[22] GOLDSBOROUGH P, PAWLOWSKI N, CAICEDO J C, et al. CytoGAN: generative modeling of cell images[J]. bioRxiv, 2017: 227645.
[23] ZHENG R, LIU L, ZHANG S, et al. Detection of exudates in fundus photographs with imbalanced learning using conditional generative adversarial network[J]. Biomedical Optics Express, 2018, 9(10): 4863 – 4878.
[24] KADURIN A, NIKOLENKO S, KHRABROV K, et al. druGAN: an advanced generative adversarial auto-encoder model for de novo generation of new molecules with desired molecular properties in silico[J]. Molecular Pharmaceutics, 2017, 14(9): 3098 – 3104.
[25] DOUZAS G, BACAO F. Effective data generation for imbalanced learning using conditional generative adversarial networks [J]. Expert Systems with Applications, 2018, 91: 464 – 471.
[26] LUTZONI F, KAUFF F, COX C J, et al. Assembling the fungal tree of life: progress, classification, and evolution of subcellular traits[J]. American Journal of Botany, 2004, 91(10): 1446 – 1480.
[27] BINDER M, HIBBETT D S. Higher-level phylogenetic relationships of homobasidiomycetes (mushroom-forming fungi) inferred from four rDNA regions[J]. Molecular Phylogenetics and Evolution, 2002, 22(1): 76 – 90.
[28] FRØSLEV T H J L. Danish fungal records database[EB/OL]. https://svampe.databasen.org. 2019.
[29] HEUSEL M, RAMSAUER H, UNTERTHINER T, et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium[C]// Advances in Neural Information Processing Systems, 2017: 6626 – 6637.
[30] GULRAJANI I, AHMED F, ARJOVSKY M, et al. Improved training of Wasserstein gans [C] // Advances in Neural Information Processing Systems, 2017: 5767 – 5777.
[31] ARJOVSKY M, CHINTALA S, BOTTOU L. Wasserstein gan[J]. arXiv preprint arXiv:1701.07875. 2017.
[32] VALLENDER S S. Calculation of the Wasserstein distance between probability distributions on the line[J]. Theory of Probability and Its Applications, 1974, 18(4): 784 – 786.
[33] Google. Tensorflow[EB/OL]. https://www.tensorflow.org. 2019.
[34] KINGMA D P, BA J. Adam: a method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980. 2014.