(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 108665058 B
(45)授权公告日 2021.01.05

(21)申请号 201810321286.1

(22)申请日 2018.04.11

(65)同一申请的已公布的文献号
申请公布号 CN 108665058 A

(43)申请公布日 2018.10.16

(73)专利权人 徐州工程学院
地址 221111 江苏省徐州市泉山区南三环路18号徐州工程学院大学科技园(徐州市2.5产业园)

(72)发明人 姜代红 刘其开 黄轲

(74)专利代理机构 北京淮海知识产权代理事务所(普通合伙) 32205
代理人 刘振祥

(51)Int.Cl.
G06N 3/04(2006.01)
G06K 9/00(2006.01)

(56)对比文件
CN 107590532 A,2018.01.16
CN 107577651 A,2018.01.12
CN 107392312 A,2017.11.24
CN 107220600 A,2017.09.29
CN 106355191 A,2017.01.25
Ishaan Gulrajani等.Improved Training of Wasserstein GANs.《arXiv》.2017,第1-20页.
Ian J. Goodfellow等.Generative Adversarial Nets.《arXiv》.2017,第1-10页.
Xi Chen等.InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets.《arXiv》.2016,第1-14页.

审查员 丁蓬莉

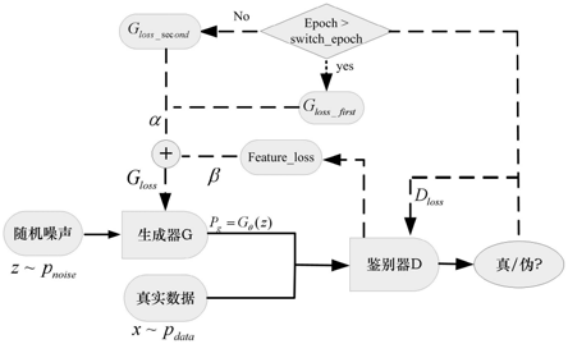
权利要求书1页 说明书12页 附图5页

(54)发明名称

一种基于分段损失的生成对抗网络方法

(57)摘要

一种基于分段损失的生成对抗网络方法,步骤如下:1、参数初始化:设批大小m=100,超参数k=1,用Xavier方法进行参数初始化,确定最大迭代次数和损失切换迭代次数参数T,令迭代次数epoch=0;2、训练判别器参数:令i=1,i为循环变量;3、训练生成器参数:epoch=epoch+1,判断epoch是否大于最大迭代次数,如小于最大迭代次数,则重复步骤2和3,如满足,则训练结束。该方法能实现生成器在不同的训练阶段采用不同形式的损失函数,一定程度上弥补了单一损失形式下GAN理论的不足,使网络训练更加稳定;通过引入真实样本与生成样本之间特征级损失,使判别器提取的特征更加鲁棒。



CN 108665058 B

1.一种基于分段损失的生成对抗网络方法,包括以下步骤:

步骤1:参数初始化:批大小m=100,即每一次参数更新时所需的样本数;设超参数k=1,即训练判别器k次才训练生成器1次;对数损失和特征损失权重分别为α=β=0.5;用Xavier方法进行参数初始化;根据数据集确定最大迭代次数和损失切换迭代次数参数T;令迭代次数epoch=0;

步骤2:训练判别器参数:令i=1,变量i是一个循环变量;

(1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$,抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$,抽取m个来自真实样本分布的带标签的样本$\{(x_1^{(1)},y^{(1)}),(x_1^{(2)},y^{(2)})\cdots\cdots(x_1^{(m)},y^{(m)})\}$;所述真实样本为手写字体数据集图像;

(2)计算判别器的无监督损失C_{unsup}:

$$C_{unsup}=-\frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)})+\log(1-D(G(z^{(i)})))];$$

(3)计算判别器的监督损失C_{sup}:

$$C_{sup}=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}*\log(\exp(x_l^{(i)})/z(x_l^{(i)}))+(1-y^{(i)})*\log(1-\exp(x_l^{(i)})/z(x_l^{(i)}))];$$

(4)通过Adam优化算法更新判别器的参数:$\nabla\theta_d(C_{unsup}+C_{sup})$;

(5)判断循环变量是否等于参数k,若小于k则重复步骤2,直至满足条件为止;若等于k,则转至下一步;

步骤3:训练生成器参数:

(1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$,抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$;

(2)计算生成器的特征级损失$V_{feature}(x,z)$:

$$V_{feature}(x,z)=\left\|E_{x\sim p_{data}}D(x)-E_{z\sim p_{noise}}D(G(z))\right\|_2^2;$$

(3)计算时间参数w(t):$w(t)=\exp[-10*(1-t)^2]$,t等于当前epoch与转换切换次数参数T的比值;

(4)计算生成器的对数损失$V_{log}(z)$:

$$V_{\log}(z)=w(t)E_{z\sim p_{noise}}[\log(1-D(G(z)))]+(1-w(t))E_{z\sim p_{noise}}[\log(-D(G(z)))];$$

(5)通过Adam优化算法更新生成器的参数:

$$\nabla\theta_g\frac{1}{m}\sum_{i=1}^{m}\alpha V_{\log}(z^{(i)})+\beta V_{feature}(x^{(i)},z^{(i)});$$

步骤4:epoch=epoch+1;判断epoch是否大于最大迭代次数,如小于最大迭代次数,则重复步骤2和步骤3,如满足,则训练结束。

一种基于分段损失的生成对抗网络方法

技术领域

[0001] 本发明属于深度学习神经网络技术领域，具体是一种基于分段损失的生成对抗网络方法。

背景技术

[0002] 生成对抗网络(Generative Adversarial Network，简称GAN)是由Goodfellow在2014年提出的无监督深度学习框架，借鉴“博弈论”的思想，构造了两个玩家：生成器(generator)和判别器(discriminator)，前者通过输入参数为(0,1)的均匀噪声或高斯随机噪声来生成图像，后者对输入的图像进行判别，判断输入是来自数据集的图像还是由生成器生成的图像。判别器将判断的结果反馈给生成器，使其朝着真实数据的分布进行优化。

[0003] 近几年，生成对抗网络在图像生成和半监督学习上应用广泛。但理论上依然存在不足，原始GAN模型难以把握生成器与判别器的同步更新，导致模型训练不稳定以及模式崩溃的现象，从而导致判别器提取的特征鲁棒性较差。

发明内容

[0004] 针对上述现有技术存在的问题，本发明提供一种基于分段损失的生成对抗网络方法，该方法能避免常规生成对抗网络在单一形式损失下出现的训练不稳定以及模式崩溃现象，从而解决判别器提取的特征较差的问题；该方法能实现生成器在不同的训练时期采用不同形式的损失函数，通过使生成器引入真实样本与生成样本之间特征级损失，使网络训练更加稳定，判别器提取的特征更加鲁棒。

[0005] 为了实现上述目的，算法主要分为以下几个步骤：

[0006] 步骤1：参数初始化：批大小m＝100，即每一次参数更新时所需的样本数；设超参数k＝1，即训练判别器k次才训练生成器1次；对数损失和特征损失权重分别为α＝β＝0.5；用Xavier方法进行参数初始化；根据数据集确定最大迭代次数和损失切换迭代次数参数T；令迭代次数epoch＝0；

[0007] 步骤2：训练判别器参数：令i＝1，变量i是一个循环变量；

[0008] (1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$，抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$，抽取m个来自真实样本分布的带标签的样本$\{(x_l^{(1)},y^{(1)}),(x_l^{(2)},y^{(2)})\cdots\cdots(x_l^{(m)},y^{(m)})\}$；

[0009] (2)计算判别器的无监督损失C_{unsup}：

[0010] $$C_{unsup}=-\frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)})+\log(1-D(G(z^{(i)})))];$$

[0011] (3)计算判别器的监督损失C_{sup}：

[0012] $$C_{sup}=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}*\log(\exp(x_l^{(i)})/z(x_l^{(i)}))+(1-y^{(i)})*\log(1-\exp(x_l^{(i)})/z(x_l^{(i)}))];$$

[0013] (4)通过Adam优化算法更新判别器的参数：$\nabla\theta_{\mathrm{d}}(C_{un\mathrm{sup}}+C_{\mathrm{sup}})$；

[0014] (5)判断循环变量是否等于参数k，若小于k则重复步骤2，直至满足条件为止；若等于k，则转至下一步；

[0015] 步骤3：训练生成器参数：

[0016] (1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$，抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$；

[0017] (2)计算生成器的特征级损失$V_{feature}(x,z)$：

[0018] $$V_{feature}(x,z)=\left\|E_{x\sim p_{data}}D(x)-E_{z\sim p_{\mathrm{noise}}}D(G(z))\right\|_2^2;$$

[0019] (3)计算时间参数w(t)：w(t)＝exp[-10*(1-t)^2]，t等于当前epoch与转换切换次数参数T的比值；

[0020] (4)计算生成器的对数损失$V_{\log}(z)$：

[0021] $$V_{\log}(z)=w(t)E_{z\sim p_{\mathrm{noise}}}[\log(1-D(G(z)))]+(1\text{-}w(t))E_{z\sim p_{noise}}[\log(-D(G(z)))];$$

[0022] (5)通过Adam优化算法更新生成器的参数：

[0023] $$\nabla\theta_g\frac{1}{m}\sum_{i=1}^{m}\alpha V_{\log}(z^{(i)})+\beta V_{feature}(x^{(i)},z^{(i)});$$

[0024] 步骤4：epoch＝epoch+1；判断epoch是否大于最大迭代次数，如小于最大迭代次数，则重复步骤2和步骤3，如满足，则训练结束。

[0025] 本发明针对生成对抗网络模型训练不稳定以及模式崩溃的问题，提出了基于分段损失的生成对抗网络。通过引入时间参数来改变生成器与判别器的训练过程，使衡量生成分布与真实分布之间差异的JS散度能够更好发挥良性作用；对生成器引入真实样本与生成样本之间的特征级损失，使训练过程更加稳定，一定程度上能够改善模型的模式崩溃现象，同时判别器提取到的特征更加鲁棒。模型在半监督图像分类上有较好的表现，其分类精度与其他算法相比有一定的优势。

附图说明

[0026] 图1是PL-GAN的计算流程；

[0027] 图2是GAN半监督分类的示意流程；

[0028] 图3是输入的真实样本的示意图；

[0029] 图4是feature-wise GAN下的生成样本；

[0030] 图5是是regular GAN的生成样本；

[0031] 图6是PL-GAN生成的样本；

[0032] 图7是PL-GAN与regular GAN以及feature-wise GAN在训练过程中的损失变化趋势对比图；

[0033] 图8是PL-GAN在mnist数据集测试对比结果；

[0034] 图9是PL-GAN在cifar10数据集测试对比结果。

具体实施方式

[0035] 机器学习算法以训练样本有无标签,可以分为带标签的监督学习和不带标签的无监督学习。由于监督学习的标签数据获取的成本很高,无监督学习算法表现不足,因此半监督学习(semi-supervised learning,SSL)成为研究学者重要的一个研究方向。SSL利用海量的无标签样本和少量标签样本能够学习具有鲁棒性的特征,在图像分类方面有着较好的表现。Lee等提出了一种对无标签数据的伪标签来帮助模型训练的高效方法。Rasmus等人提出了基于自动编码器的阶梯网络,编码器用于监督学习,解码器的每一层与编码器一一对应,形成阶梯,用于无监督学习训练。

[0036] 近年来,深度生成模型(Deep generative models,DGMs)与生成对抗网络(GAN)在半监督学习上有着良好的表现。Springenberg等提出的Cat-GAN在判别器中引进适当的目标函数来权衡输入样本与对应预测类别的互信息,通过最大化生成数据类别的条件交叉熵来训练分类器。Odena和Salimans等将判别器的二分类的概率输出扩展到N+1类别概率输出,真实样本对应的N个类加上生成样本类别。Li等提出了triple GAN,通过引入额外的分类器,改善了GAN在半监督学习上的生成器和判别器在训练时无法同时达到最优的问题且生成器能够学习到样本的语义特征。文献中提到可以利用无标签数据对GAN的判别器进行预训练,用少量有标签数据对判别器进行微调,再用于分类任务。

[0037] GAN在训练过程中会出现不稳定以及模式崩溃的问题,Arjovsky等从理论分析了其原因,当生成样本分布与真实样本分布之间的支撑集没有重叠或可忽略的重叠部分为0时,生成器的损失梯度近似常数。常规GAN采用衡量两者分布差异的JS散度受限于分布之间支撑集没有重叠。于是作者提出了Wesserstein-GAN(W-GAN),通过满足一定约束条件下神经网络逼近函数来近似度量两者分布的距离。Che等提出了模态正则化的GAN,通过设计一系列的相似度量方法对判别器进行约束,提供更加稳定的梯度来训练生成器。Metz等针对模式崩溃的问题,提出了基于梯度损失的Unrolled GAN,通过计算二阶梯度来指示生成器一阶梯度变化的方向。

[0038] 针对上述问题,本申请提出一种基于分段损失的生成对抗网络(Piecewise-Loss Generative Adversarial Networks,PL-GAN),通过引入时间参数,使生成器在不同的训练阶段采用不同的损失函数,JS散度就可以发挥良性作用。为了给生成器提供足够的梯度,本申请额外引入了生成样本和真实样本在特征空间的均方差损失,从而PL-GAN不仅有效改善了生成器梯度消失的问题,而且在半监督学习方面有着不俗的表现。

[0039] GAN可以理解为两个玩家相互博弈的二人游戏,即生成器(Generator,G)和判别器(Discriminator,D)之间的“零和游戏”。G的输入是噪声变量z,目的是拟合真实样本的数据分布,输出生成样本G(z)。D来判断输入的样本是来自真实样本还是生成样本,输出输入的样本来自真实样本的概率D(x)。因此判别器D的优化目标对输入为真实样本的概率尽可能为1,输入为生成样本的概率尽可能为0。生成器G的优化目的是最大化判别器D给出错误判断的概率,最终的优化目的是寻找两者之间的一个纳什均衡。其优化过程是一个极大极小的一个对抗过程,其目标函数为V(D,G):

[0040]
$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}\left[\log D(x)\right] + E_{z\sim p_{noise}(z)}[\log(1-D(G(z)))] \quad (1)$$

[0041] 其中，$p_{data}(x)$表示的是真实样本的分布，$p_{noise}(z)$表示的是输入噪声分布。

[0042] GAN的优化目标是通过双玩家游戏策略隐式地训练一个逼近真实样本的生成器$p_g(z)=G_0(z)$，常规GAN判别器D的目的是尽可能判别出真实样本与生成样本的真伪，其损失函数为：

$$D_{loss}=-E_{x\sim p_{data}}[\log D(x)]-E_{z\sim p_{noise}}[\log(1-D(G(z)))] \quad (2)$$

[0043] 生成器G的损失函数有以下两种形式：

$$G_{loss\_first}=E_{z\sim p_{noise}}[\log(1-D(G(z)))]=E_{x\sim p_g}\log(1-D(x)) \quad (3)$$

[0044] $$G_{loss\_second}=E_{z\sim p_{noise}}[-\log D(G(z))] \quad (4)$$

[0045] 当$p_{data}(x)=p_g(x)$时，最优判别器表示为：$D^*(x)=\frac{P_{data}(x)}{P_{data}(x)+P_g(x)}=\frac{1}{2}$ (5)

[0046] 最优判别器下，式(5)代入式(1)得生成器第一种损失函数形式为：

[0047] $$E_{x\sim p_g}\log(1-D^*(x))+E_{x\sim p_{data}}\log(D^*(x)) \quad (6)$$

[0048] 式(6)代入$D^*(x)$后，引入衡量相似度的两个指标KL散度和JS散度。

[0049] $$KL(P_1\|P_2)=E_{x\sim P_1}\log\frac{P_1}{P_2}$$

[0050] $$JS(P_1\|P_2)=\frac{1}{2}KL(P_1\|\frac{P_1+P_2}{2})+\frac{1}{2}KL(P_2\|\frac{P_1+P_2}{2})$$

[0051] 则最优判别器下G第一种损失函数形式最终为：

[0052] $$G_{loss\_first|D^*}=2JS(P_{data}\|P_g)-2\log 2 \quad (7)$$

[0053] 即在最优判别器下，最小化生成器的损失等价为最小化生成样本与真实样本之间的JS散度。但Arjovsky等证明了在第一种损失函数形式下，JS散度衡量分布差异的前提是两者的分布要有所重叠或有不可忽略的重叠，否则JS散度将会是一个常数。但网络初始化后的生成样本分布很难与真实样本分布有不可忽略的重叠。

[0054] 同样，由式(6)和式(7)可得G的第二种形式的损失函数如下：

[0055] $$\begin{aligned}&G_{loss\_second}\\&=KL(P_g\|P_{data})-E_{x\sim p_g}\log(1-D^*(x))\\&=KL(P_g\|P_{data})-2JS(P_{data}\|P_g)+2\log 2+E_{x\sim p_{data}}\log(D^*(x))\end{aligned} \quad (8)$$

[0056] 由于后两项不依赖于G，最终最小化式(4)等价于最小化

[0057] $$G_{loss\_second|D^*}=KL(P_g\|P_{data})-2JS(P_{data}\|P_g) \quad (9)$$

[0058] 该目标形式一方面要求最小化生成分布与真实分布的KL散度，另一方面又要求最大化两者的JS散度，优化目标相互矛盾。且$KL(p_g||p_{data})$不是一个对称的度量，$KL(p_g||p_{data})$不等于$KL(p_{data}||p_g)$，当p_{data}与p_g的取值相对改变时，KL散度也会变化，这就迫使生成

器生成大量重复且置信度较高的样本，导致了模式崩溃。

[0059] WGAN作者针对生成器第一种损失函数存在的缺点，提出了对生成样本和真实样本加噪声的方法，使得原本的两个低维流形弥散到整个高维的空间，迫使它们产生不可忽略的重叠，而一旦存在重叠，JS散度就能真正发挥作用，梯度消失的问题便得到了解决，随着训练的进行，再进行噪声退火，JS散度照样能发挥作用，继续产生有意义的梯度把两个低维流形拉近，直到完全重合。

[0060] 本文借鉴了噪声退火的思想，通过引入时间参数$w(t)=\exp[-10*(1-t)^2]$，来控制GAN在不同的训练阶段采用不同形式的损失。训练的前期以第二种损失函数方式为主，随着训练的进行，真实样本和生成样本就能够有所重叠，训练进行到某一阶段，再切换到以第一种损失方式为主，此时JS散度就可以发挥良性作用，从而避免生成器梯度消失和模式崩溃。同时为了给生成器提供足够的梯度，本文引入生成样本和真实样本之间特征级的均方差损失。最终生成器的损失函数如下：

[0061]
$$\begin{aligned}\min_G V(G) = \alpha\{&w(t)E_{z\sim p_{noise}}[\log(1-D(G(z)))] \\ &+(1\text{-}w(t))E_{z\sim p_{noise}}\log(-D(G(z)))\} \\ &+\beta\{\left\|E_{x\sim p_{data}}D_f(x)-E_{z\sim p_{noise}}D_f(G(z))\right\|_2^2\}\end{aligned} \quad (10)$$

[0062] 其中$D_f(*)$表示判别器特征层的输出。

[0063] 有关PL-GAN的计算流程如图1所示，考虑到GAN的监督损失，假设标准的分类器输出是N维向量$logits=\{l_1,l_2,\cdots,l_N\}$，N为样本的类别数。则用softmax计算输出的概率为：

[0064]
$$p_{model}(y=j\mid x)=exp(l_j)\Big/\sum_{n=1}^{N}\exp(l_n) \quad (11)$$

[0065] 将生成样本所属的类别定义为第N+1类，则来自生成样本的概率可表示为$P_{model}(y=N+1\mid x)$，对应常规GAN的1-D(x)。假设判别器的训练样本一半来自真实样本，另一半来自生成样本，则D的损失函数可表示为：

[0066]
$$\begin{aligned}C(D)&=-E_{x,y\sim p_{data(x,y)}}[\log P_{model}(y\mid x)]-E_{x\sim p_g}[\log P_{model}(y=N+1\mid x)] \\ &=C_{sup}+C_{unsup\_adv}\end{aligned} \quad (12)$$

[0067] 其中，真实样本由带标签样本和不带标签样本组成。由带标签样本参与的监督损失为：

[0068]
$$C_{sup}=-E_{x,y\sim p_{data(x,y)}}[\log P_{model}(y\mid x,y<N+1)] \quad (13)$$

[0069] 由不带标签的真实样本和生成样本参与的无监督对抗损失为：

[0070]
$$\begin{aligned}C_{unsup\_adv}=&-E_{x\sim p_{data}(x)}[\log(1-P_{model}(y=N+1\mid x))] \\ &-E_{x\sim p_g}[\log P_{model}(y=N+1\mid x)]\end{aligned} \quad (14)$$

[0071] 令$D(x)=1-P_{model}(y=N+1\mid x)$，则

[0072]
$$C_{unsup\_adv}=-E_{x\sim p_{data}(x)}[\log(1-D(x))]-E_{z\sim p_{noise}}[\log(1-D(G(z)))] \quad (15)$$

[0073] 如何计算监督损失和无监督对抗损失成为问题的关键。从最终的优化目标的角度分析，存在一个未知的映射函数$f(x)$，使$\forall j<N+1$，$p(y=j,x)=f(x)\cdot\exp[l_j(x)]$，且$p_G$

(x)＝f(x)·exp[l_{N+1}(x)]成立。由于判别器输出维度为N+1的概率向量是过参数化的，假设$\forall x, l_{N+1}(x)=0$，则不会改变判别器softmax概率值。此时，GAN监督损失变为标准的分类器的监督损失，输出为$D(x)=\frac{Z(x)}{Z(x)+1}$，其中，$Z(x)=\sum_{n=1}^{N}\exp[l_n(x)]$。有关GAN半监督图像分类示意流程如图2所示，其中，标签样本为D贡献监督损失，无标签样本为D贡献无监督损失。

[0074] 本发明提供了一种基于分段损失的生成对抗网络方法，包括以下步骤：

[0075] 步骤1：参数初始化：批大小m＝100，即每一次参数更新时所需的样本数；设超参数k＝1，即训练判别器k次才训练生成器1次；对数损失和特征损失权重分别为α＝β＝0.5；用Xavier方法进行参数初始化；根据数据集确定最大迭代次数和损失切换迭代次数参数T；令迭代次数epoch＝0；

[0076] 步骤2：训练判别器参数：令i＝1，变量i是一个循环变量；

[0077] (1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$，抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$，抽取m个来自真实样本分布的带标签的样本$\{(x_1^{(1)},y^{(1)}),(x_1^{(2)},y^{(2)})\cdots\cdots(x_1^{(m)},y^{(m)})\}$；

[0078] (2)计算判别器的无监督损失C_{unsup}：

[0079] $$C_{unsup}=-\frac{1}{m}\sum_{i=1}^{m}[\log D(x^{(i)})+\log(1-D(G(z^{(i)})))];$$

[0080] (3)计算判别器的监督损失C_{sup}：

[0081] $$C_{sup}=-\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}*\log(\exp(x_l^{(i)})/z(x_l^{(i)}))+(1-y^{(i)})*\log(1-\exp(x_l^{(i)})/z(x_l^{(i)}))];$$

[0082] (4)通过Adam优化算法更新判别器的参数：$\nabla\theta_d(C_{unsup}+C_{sup})$；

[0083] (5)判断循环变量是否等于参数k，若小于k则重复步骤2，直至满足条件为止；若等于k，则转至下一步；

[0084] 步骤3：训练生成器参数：

[0085] (1)抽取m个来自噪声分布$P_{noise}(z)$的随机样本$\{z^{(1)},z^{(2)}\cdots\cdots z^{(m)}\}$，抽取m个来自真实样本分布的无标签样本$\{x^{(1)},x^{(2)}\cdots\cdots x^{(m)}\}$；

[0086] (2)计算生成器的特征级损失$V_{feature}(x,z)$：

[0087] $$V_{feature}(x,z)=\left\|E_{x\sim p_{data}}D(x)-E_{z\sim p_{noise}}D(G(z))\right\|_2^2;$$

[0088] (3)计算时间参数w(t)：w(t)＝exp[-10*(1-t)2]，t等于当前epoch与转换切换次数参数T的比值；

[0089] (4)计算生成器的对数损失$V_{log}(z)$：

[0090] $$V_{log}(z)=w(t)E_{z\sim p_{noise}}[\log(1-D(G(z)))]+(1\text{-}w(t))E_{z\sim p_{noise}}[\log(-D(G(z)))];$$

[0091] (5)通过Adam优化算法更新生成器的参数：

[0092] $$\nabla\theta_g\frac{1}{m}\sum_{i=1}^{m}\alpha V_{log}(z^{(i)})+\beta V_{feature}(x^{(i)},z^{(i)});$$

[0093] 步骤4：epoch＝epoch+1；判断epoch是否大于最大迭代次数，如小于最大迭代次

数,则重复步骤2和步骤3,如满足,则训练结束。

[0094] 基于分段损失的生成对抗网络半监督分类算法步骤如表1所示。设超参数k＝1,即训练判别器k次才训练生成器1次;对数损失和特征级均方差损失权重分别为:α＝β＝0.5;用Xavier方法进行参数初始化。

[0095] 表1 PL-GAN的参数设置及算法步骤

Require：批大小（Batchsize）m=100，即每一次参数更新时所需的样本数；

iterations：遍历一次训练集所需 m 的个数，即总的样本数除以 m 的值；epoch：一个 epoch 等于遍历训练集一次。

[0096] Require：时间参数 $w(t)=\exp[-10*(1-t)^2]$，t 等于当前 epoch 与损失切换 epoch 的比值。

Require:为了方便抽样，确定训练集标签样本个数为 n，并将扩展到无标签样本的大小，如 n=500，无标签样本个数为 5000，则标签样本复制扩展 10 倍。

for number of training iterations **do**

for k steps **do**

●抽样 m 个噪声样本$\{z^{(1)}, z^{(2)} \ldots\ldots z^{(m)}\}$

●抽样 m 个无标签样本$\{x^{(1)}, x^{(2)} \ldots\ldots x^{(m)}\}$

●抽样 m 个有标签样本$\{(x_l^{(1)}, y^{(1)}), (x_l^{(2)}, y^{(2)}) \cdots\cdots (x_l^{(m)}, y^{(m)})\}$

●计算判别器的无监督损失：

$$C_{unsup} = \frac{1}{m}\sum_{i=1}^{m}[-\log D(x^{(i)}) - \log(1 - D(G(z^{(i)})))]$$

●计算判别器的有监督损失：

$$C_{\sup} = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)} * \log(\exp(x_l^{(i)}) / z(x_l^{(i)})) + (1 - y^{(i)}) * \log(1 - \exp(x_l^{(i)}) / z(x_l^{(i)}))]$$

●通过 SGD 更新判别器的参数：

$$\nabla\theta_d[C_{unsup} + C_{\sup}]$$

[0097]

end for

●抽样 m 个噪声样本$\{z^{(1)}, z^{(2)} \ldots\ldots z^{(m)}\}$

●抽样 m 个无标签样本$\{x^{(1)}, x^{(2)} \ldots\ldots x^{(m)}\}$

●计算生成器的特征级均方差损失

$$V_{feature}(x,z) = \| E_{x\sim p_{data}(x)} D_f(x) - E_{z\sim p_{noise}(z)} D_f(G(z)) \|_2^2$$

●计算生成器的对数损失

$$V_{\log}(z) = w(t)E_{z\sim p_{noise}} \log(1 - D(G(z))) + (1\text{-}w(t))E_{z\sim p_{noise}} \log(-D(G(z)))$$

●通过 SGD 更新生成器的参数：

$$\nabla\theta_g \frac{1}{m}\sum_{i=1}^{m}\alpha V_{\log}(z^{(i)}) + \beta V_{feature}(x^{(i)}, z^{(i)})$$

end for

对于梯度的优化算法，本文使用的是 Adam 优化学习算法

[0098] 实验结果与分析

[0099] 1实验数据集

[0100] 为了验证本文方法的表现，采用两个数据集MNIST、CIFAR-10进行相关的实验。MNIST是深度学习领域常见的手写字体数据集，共十类(数字0～9)，每类包含单通道的6000个训练样本和1000个测试样本。CIFAR-10包含着6万张包括10类的三通道样本，每类有5000个训练样本和1000个测试样本。

[0101] 2网络参数分析

[0102] 基于theano深度学习框架上进行实验对比，在单块GPU型号为GTX980上运行，对于生成器的损失函数来说，损失切换迭代次数参数switch epoch以及学习率衰减策略非常重要，直接影响着生成图像的质量以及稳定性，从而影响判别器的好坏。关于每个数据集中对应的switch epoch与learning rate参数的设置如下表2所示。

[0103] 表2参数设置

[0104]

数据集	Mnist	Cifar10
总迭代次数	300	600
损失切换迭代次数	60	100
初始学习率	0.003	0.0003
衰减因子	min(2.0-epoch/250，1.0)	min(2.0-epoch/450，1.0)

[0105] 3实验对比分析

[0106] 1)MNIST数据集

[0107] 本实验的模型框架主要是由多层感知机组成，实验的评价标准一方面是生成样本的质量，另一方面是半监督分类的精度。对比的算法主要是常规GAN(regular GAN)，其生成器的损失为对数损失$E_{z\sim p_{noise}}\log(-D(G(z)))$；特征级损失GAN(feature-wise GAN)，其生成器的损失仅仅是特征级均方差损失$\| E_{x\sim p_{data}(x)}D_f(x)-E_{z\sim p_{noise}(z)}D_f(G(z))\|_2^2$；PL-GAN在常规GAN的基础上改变了对数损失的形式且引入了特征级损失，然后经过加权，作为PL-GAN的生成器损失。

[0108] 如图3至6所示，其中图3表示的是真实样本的输入，图4表示的是feature-wise GAN下的生成样本，图5表示的是regularGAN的生成样本，图6表示的是PL-GAN生成的样本；与feature-wise GAN相比，PL-GAN生成的样本质量较好，与regular GAN相比，PL-GAN生成的样本虽然质量上稍差，但多样性明显好于前者。即改善了常规的GAN模式崩溃的问题。

[0109] 如图7所示是PL-GAN与regular GAN以及feature-wise GAN在训练过程中的损失变化趋势对比图。其中对生成器的损失而言，PL-GAN相比feature-wise GAN损失下降得更加稳定；与regular GAN相比，PL-GAN损失的变化呈现逐渐下降的趋势，而不是趋近于一个常数。对判别器的损失而言，PL-GAN的变化趋势相比regular GAN较好，与feature-wise GAN相当。在保证模型的结构框架相同的情况下，当标签样本为100时，比较半监督分类性

能。分类对比结果如图8所示，PL-GAN相比regular GAN分类错误率较低，相比feature-wise GAN，分类性能虽基本接近，但收敛性更好。

[0110] 表3 MNIST测试错误率对比(labels＝100)

[0111]

方法	M1+M2[4]	VAT[19]	Ladder[3]	ADGM[20]	CatGAN[7]	Improved-GAN[9]	Triple-GAN	PL-GAN
错误率(%)	3.33(±0.14)	2.33	1.06(±0.37)	0.96(±0.02)	1.39(±0.28)	0.93(±0.07)	0.91(±0.58)	**0.90**(±0.06)

[0112] 同样设标签样本为100，与传统的半监督学习算法，如基于深度生成模型算法的ADGM、M1+M2、基于虚拟对抗网络VAT、CatGAN等方法比较，PL-GAN有较好的表现。比较结果如表3所示。

[0113] 为了加速模型的训练速度，在判别器结构中，加入了weightNormalization正则化，有关PL-GAN模型的参数配置如表4所示。

[0114] 表4 MNIST数据集网络参数配置

[0115]

Discriminator D	Generator G
Input：28×28 gray image one-hot labels $y \in R^{10}$	Input Noise $\in R^{100}$
Denselayer 1000 UnitslReLU, guassian niose,Weight norm	Denselayer 500 Units,
Denselayer 500 Units，lReLU, guassian niose, Weight norm	Softplus, batch norm

[0116]

Denselayer 250 Units，lReLU, guassian niose, Weight norm	Denselayer 500 Units
Denselayer 250 Units，lReLU, guassian niose, Weight norm	Softplus, batch norm
Denselayer 10 Units，lReLU, guassian niose, Weight norm	Denselayer 784 Units, sigmoid

[0117] 2)c i far 10数据集

[0118] 本实验的模型框架主要是以DCGAN的框架为基准，训练之前采用ZCA白化对数据集进行预处理，初始基准学习率设为0.0003，为了加速模型的训练和防止模型的过拟合，模型中加入weight normalization和dropout策略。关于网络模型参数的配置如表5所示。

[0119] 表5 CIFAR10网络参数配置

[0120]

Discriminator D	Generator G
Input： 32×32 Colored image one-hot labelsclass $y \in R^{10}$	Input Noise $\in R^{100}$
Dropout=0.2 3×3 conv, 96, lReLU,weight norm 3×3 conv, 96, lReLU,weightnorm 3×3 conv, 96, lReLU,weight norm	MLP 8192 units ReLU, batch norm Reshape 512×4×4 5×5 deconv, 256, stride 2 ReLU, batch norm
Dropout=0.2 3×3 conv, 192,lReLU,weight norm 3×3 conv, 192,lReLU,weight norm	5×5 deconv. 128. stride 2 ReLU, batch norm

[0121]

3×3 conv, 192,lReLU,weight norm	
Dropout=0.2 3×3 conv, 192,lReLU,weight norm NIN, 192,lReLU,weight norm NIN, 192,lReLU,weight norm Global pool Denselayer 10 Unitswith weight norm	5×5 deconv. 3. stride 2, tanh,weight norm

[0122] 在保证模型的结构框架相同的情况下，当标签样本为4000时，半监督分类对比结果如图9所示，PL-GAN相比regular GAN分类错误率较低，相比feature-wise GAN，分类性能虽基本接近，但收敛性更好。

[0123] 表6不同模型生成样本的IS值

[0124]

方法	真实样本	Regular GAN	feature-wise GAN	PL-GAN
score±std.	11.24±0.12	6.16±0.46	5.54±0.26	6.87±0.86

[0125] 与其他传统的半监督分类算法相比，设同样设带标签的训练样本为4000，对比结果如表7所示，PL-GAN有较好的表现，优于大部分传统算法。

[0126] 表7 CIFAR10测试错误率对比结果(labels＝4000)

[0127]

方法	VAT[19]	Ladder[3]	CatGAN[7]	Improved-GAN[9]	Triple GAN[10]	ALI[24]	PL-GAN
错误率(%)	24.65	20.40 (±0.47)	19.58 (±0.58)	18.63 (±2.32)	**16.99**(±0.36)	18.3	17.30(±0.56)

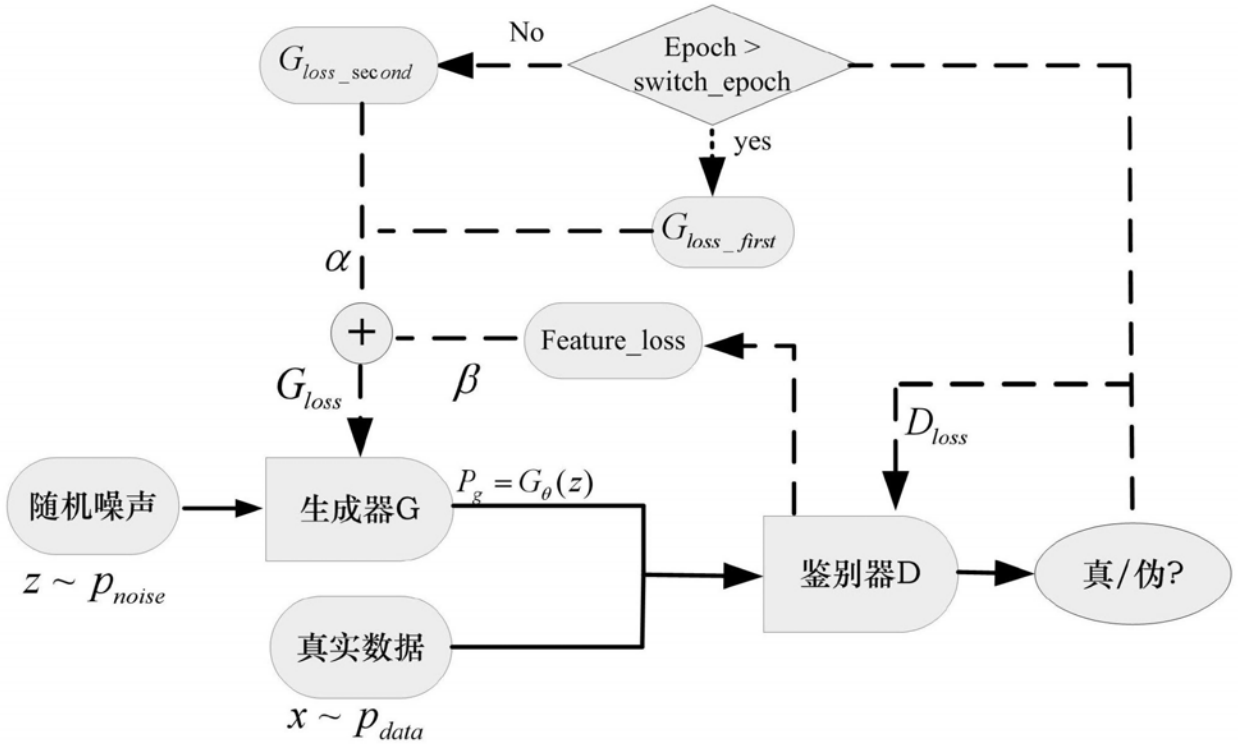

No
Epoch > switch_epoch
yes
G_{loss_second}
G_{loss_first}
α
β
Feature_loss
G_{loss}
D_{loss}
随机噪声
$z \sim p_{noise}$
生成器G
$P_g = G_\theta(z)$
真实数据
$x \sim p_{data}$
鉴别器D
真/伪?

图1

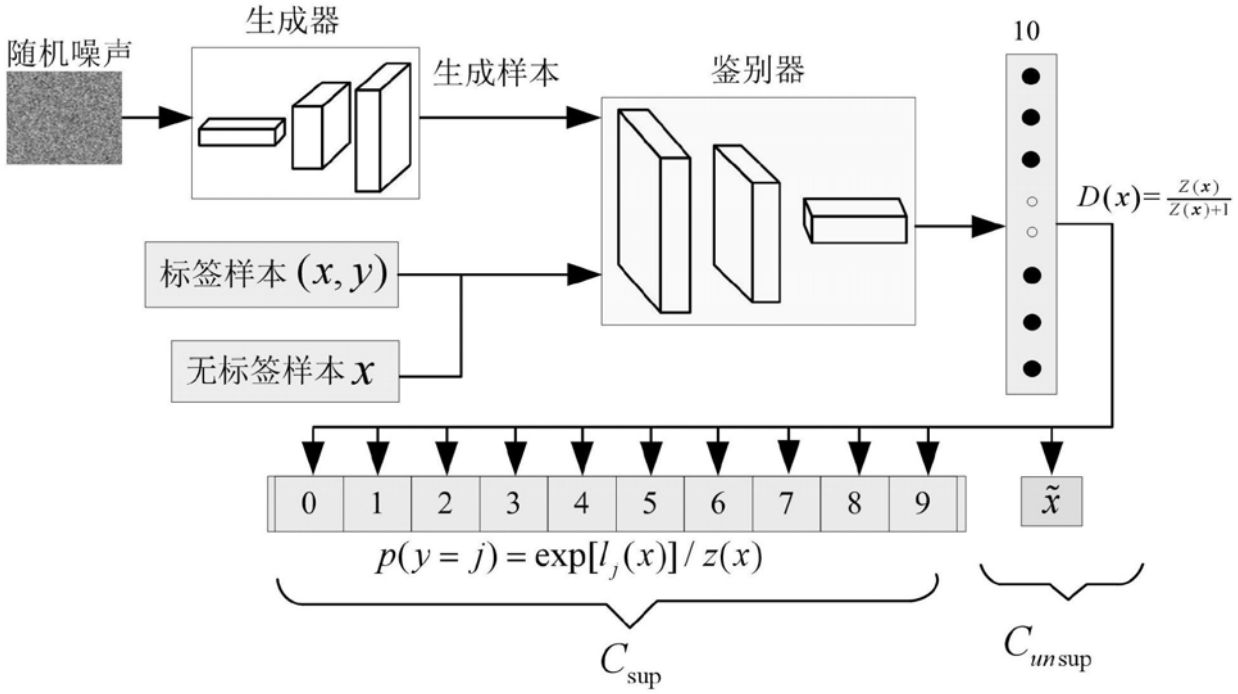

随机噪声
生成器
生成样本
鉴别器
10
$D(x)=\frac{Z(x)}{Z(x)+1}$
标签样本 (x, y)
无标签样本 x
0 1 2 3 4 5 6 7 8 9
$\tilde{x}$
$p(y=j)=\exp[l_j(x)]/z(x)$
C_{sup}
C_{unsup}

图2

图3

图4

图5

图6

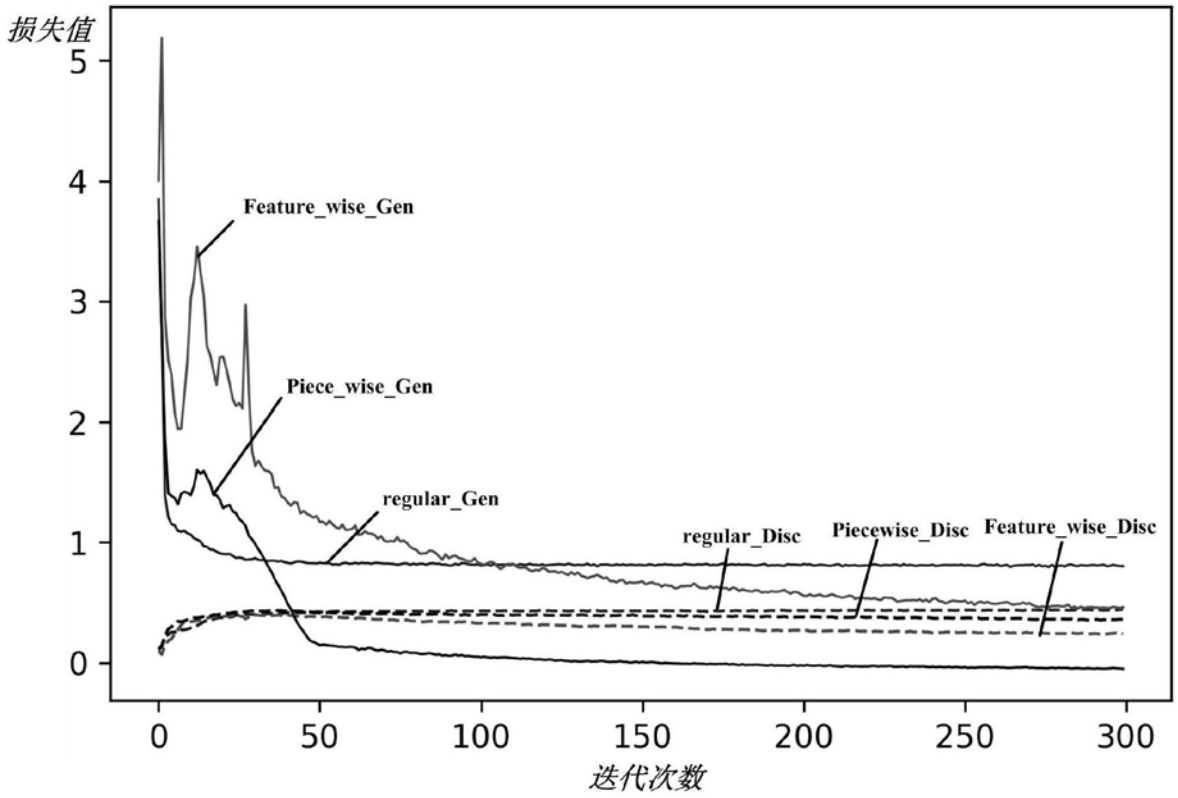

损失值
5
4
3
2
1
0
Feature_wise_Gen
Piece_wise_Gen
regular_Gen
regular_Disc
Piecewise_Disc
Feature_wise_Disc
0
50
100
150
200
250
300
迭代次数

图7

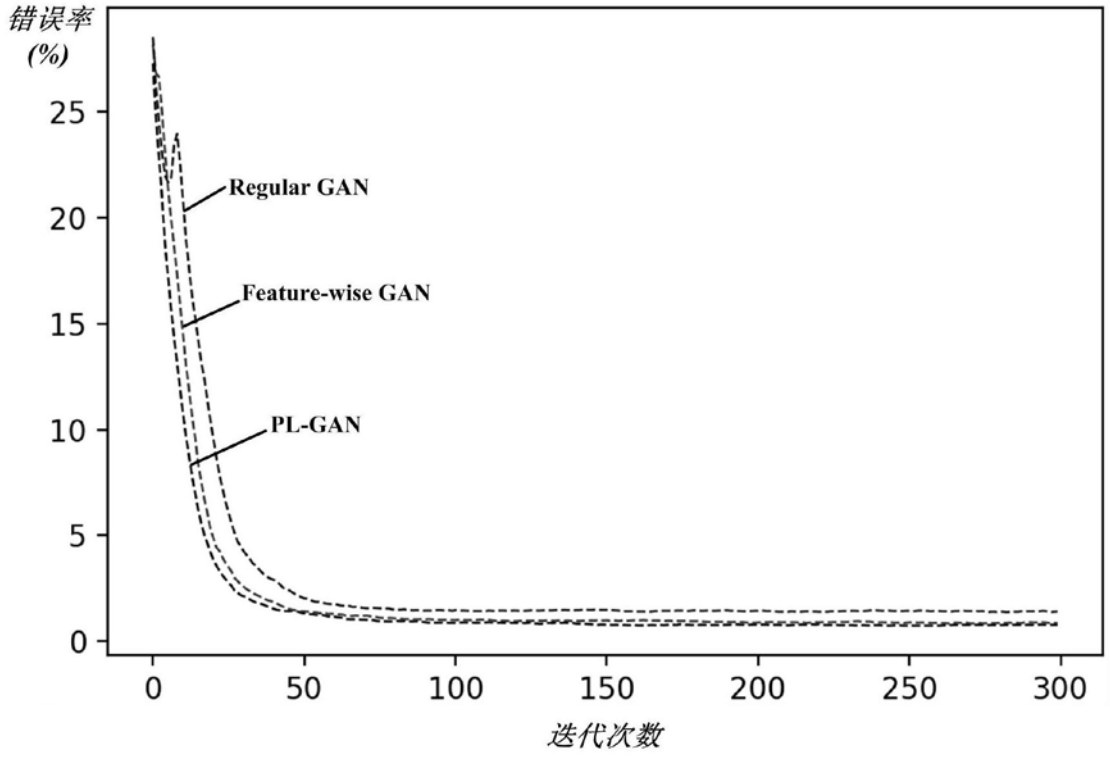

错误率
(%)
25
20
15
10
5
0
Regular GAN
Feature-wise GAN
PL-GAN
0
50
100
150
200
250
300
迭代次数

图8

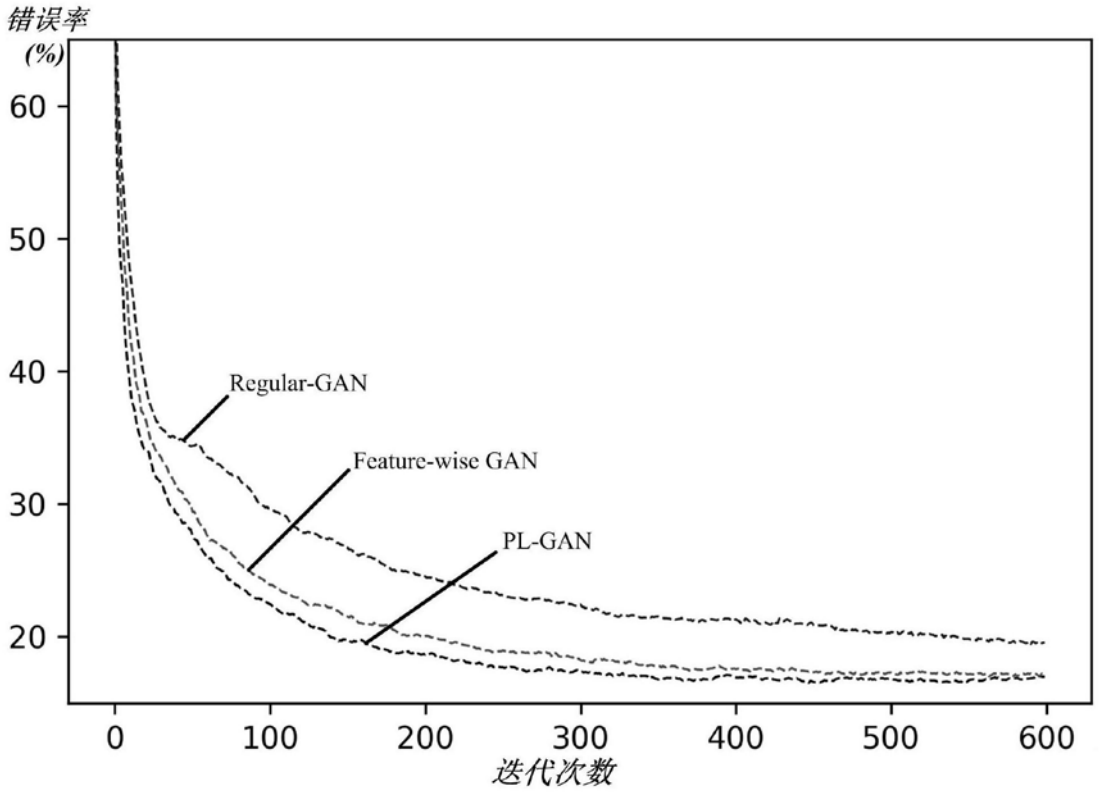

错误率
(%)
60
50
40
30
20
Regular-GAN
Feature-wise GAN
PL-GAN
0
100
200
300
400
500
600
迭代次数

图9