上海交通大學

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS

论文题目：生成式对抗网络的超参数优化研究

学生姓名：金颖

学生学号：5140369003

专　　业：信息安全

指导教师：张月国、龙明盛

学院(系)：电子信息与电气工程学院

生成式对抗网络的超参数优化研究

摘要

生成式对抗网络（GAN）是人工智能学界一个热门的研究方向，被广泛应用于计算机视觉等领域。在训练 GAN 时，不同的超参数设置会严重影响训练结果，目前缺乏 GAN 的自动化超参数优化工具，主要依赖手工调节，效率很低。因此，本文旨在研究一套 GAN 超参数优化方法，支持 GAN 自动调参及超参数粗调等。

传统机器学习算法的超参数优化一般采用验证准确率辅以损失函数作为搜索准则，这种准则对于 GAN 不适用。本文创新提出将 GAN 评价指标作为搜索准则，利用算法进行超参数搜索，并遴选五种主流数据集开展多项实验。提出评价指标 JS-IS 和 WD，通过实验综合 IS、FID 横向比较了这四种指标，JS-IS 稳定性高于 IS，WD 的灵敏度最高、对生成图片的清晰度和模式坍塌程度描述能力最强。实验揭示四种指标难以反映有监督 GAN 生成图片的标签准确性，本文进一步提出标签正确率计算方法。此外，由于评价指标在清晰度和多样性上各有侧重，提出了指标加权算法并通过实验证明其有效性。为了提升优化效率，提出 exploit & explore 算法，实验表明其所选的超参数分布情况、有效搜索次数和准确率都高于传统搜索算法：网格搜索和随机搜索。

整合上述成果，实现了一套可视化的超参数自动优化系统，支持无监督、有监督 GAN，显著提高了 GAN 超参数调节的效率。

关键词：生成式对抗网络，超参数优化，生成式对抗网络评价指标，模式坍塌

HYPERPARAMETER OPTIMIZATION OF GENERATIVE ADVERSARIAL NETWORKS

ABSTRACT

Generative Adversarial Network(GAN), a popular research area in artificial intelligence, is applied into many fields such as computer vision. When training GANs, different hyperparameters often has dramatic influence on training results. Now lack of automatic hyperparameter optimization tools makes researchers tune hyperparameters manually, which is low in efficiency. Therefore, in this paper, we conduct research on automatic hyperparameter optimization of GANs and implement the system.

Hyperparameter optimization tools of traditional machine learning methods often take validation accuracy, with the help of loss function, as search criteria, but it is not suitable for GANs. Creatively, we take evaluation criteria of GANs as search criteria. In addition, we introduce two novel evaluation criteria, JS-IS and WD. Through comprehensive experiments with IS and FID, we compare the performance of four criteria, finding that JS-IS is more stable than IS, and WD is the best criterior among the four, both in evaluating image sharpness and mode collapse phenomenon. Experiment results show that the four criteria all fail in evaluating label accuracy in supervised learning and then we propose a method to calculate label accuracy. In addition, considering four criteria favor different things, we introduce criterior tradeoff and prove its effectiveness. In order to enhance efficiency, we propose a novel search method: exploit & explore algorithm. Trough experiments we find that exploit & explore algorithm outperforms existing search methods, gird search and random search, in hyperparameter distribution, validity and accuracy.

Based on the results above, we implement an automatic and visualized hyper-parameter optimization system called Auto-GAN for both unsupervised and supervised GANs. Comparing with traditional manual tuning, this system can obviously improve the efficiency of GANs' hyperparameter optimization.

Key words: Generative Adversarial Network, hyperparameter optimization, evaluation criterior for Generative Adversarial Network, mode collapse

目 录

第一章 绪论

1.1 研究背景与意义

生成式对抗网络[1]（Generative Adversarial Network，以下简称 GAN）是一种深度学习模型，在 2014 年由 Ian GoodFellow 提出，它是通过其中的生成模型和判别模型的相互博弈来产生输出的。其中，生成模型 G 用噪声数据来生成图片，判别模型 D 判别图片是来自真实数据集还是 G 生成的。G 设法让 D 无法分辨是生成的图片还是真实图片，而 D 则要设法让自己有能力将两种图片区分开来。双方在博弈中不断优化自己，最终可以达到一个最优状态。

GAN 这种深度学习模型的提出在研究界引起了广泛关注，是人工智能学界一个热门的研究方向，被广泛地应用到计算机视觉（如图像翻译、图像补全）和自然语言处理等领域，研究人员也将从无监督学习拓展到了有监督学习和半监督学习领域。GAN 能够生成与训练集数据类似的数据，可以用于数据增强、图片风格迁移等。图 1-1 是 GAN 的一个应用示例。

图 1-1 GAN 应用示例图

在图 1-1 中，四个有监督 GAN 的训练集分别为莫奈、梵高等四位画家的画作集，在训练完成之后，向四个 GAN 分别输入最左边的真实风景照作为有监督 GAN 的输入，四个 GAN 会分别输出对应画家风格的画作。可见 GAN 在生成数据方面有着很强的能力。

然而，在 GAN 的训练中，由于 G 的输入只需要是一组噪声，G 和 D 的训练本身就比较自由，加之实际应用中往往用较不稳定的深度神经网络来作为 G 和 D，导致 G 和 D 的博弈过程存在了很大的不确定性，甚至在一些时候，多次训练同一个 GAN 得到的结果也会有很大不同，GAN 的提出者 Ian GoodFellow 就指出，如果要证明一种 GAN 的性能，必须在同样的环境下将模型重复运行至少 3 次，这体现出 GAN 的训练是很不稳定的。

GAN 的变种包括 WGAN[2]、BEGAN[3]、LSGAN[4]，在对各种 GAN 的复现实验中可以发现，它们的训练效果受超参数的影响都很大[5]。在同样的训练集和资源条件下，同一个 GAN，不同的超参数可以得到截然不同的训练结果。

在一些超参数下，GAN 的训练结果良好，且多次训练同一个 GAN 得到的结果大致相同，体现出了比较好的稳定性；而在另一些超参数下，GAN 的训练结果质量直线下降或者完全失败。经验表明，GAN 对超参数的苛刻程度要远远高于传统的机器学习算法，也要略高于传统的深度神经网络。在实际应用中，往往需要研究人员人工地对 GAN 进行超参数优

化，以在一个待选的超参数空间中寻找到可用的超参数，由于 GAN 对超参数很是苛刻，这一过程常常需要耗费大量的时间和精力。因此，研究出一套快速有效的超参数优化方法，形成一套自动化的超参数优化系统，对于提升 GAN 的训练效率和应用效果都是大有裨益的。

1.2 国内外研究现状

如今，深度学习在计算机视觉领域具有很重要的地位。而在超参数优化方面，研究人员也提出了一些超参数优化的方法。GAN 的自身架构和评价指标在 GAN 的研究中都是十分关键的。因此下面从深度学习的发展、超参数优化、GAN 的研究现状与 GAN 的现有评价指标四个方面分别介绍国内外的研究现状。

1.2.1 深度学习的发展

自 AlexNet[6]在图片分类竞赛 ImageNet[7]中获得冠军以来，深度卷积神经网络在机器学习界大放异彩。AlexNet 的结构如图 1-2 所示。

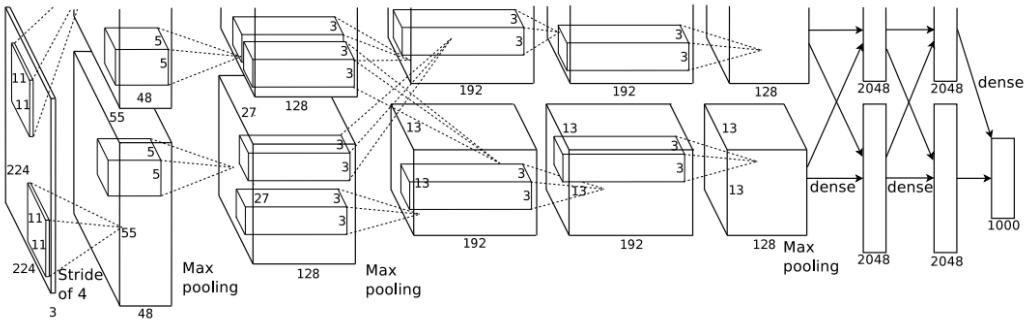


图 1-2 AlexNet 结构

从图 1-2 可以看出，AlexNet 共有八层，前五层为卷积层，后三层为全连接层，当网络被用于图片分类时，它的学习过程可以看成是一个提取特征并且根据特征将图片转换成概率值，从而对图片进行分类的过程。

随着深度网络的不断发展，后续又出现了 VGG[8]，GoogLeNet[9]等深度网络。近期，ResNet[10]在 ImageNet 上甚至展现出了比人类更高的分类准确率，体现出了深度网络极其优秀的抽取特征进行分类的能力。

介于深度网络具有很好的特征抽取能力，它被广泛地应用到了计算机视觉、自然语言处理等领域，在许多领域较传统机器学习方法具有很大的优势。

在 GAN 的原型中，生成模型和判别模型为多层感知机。而在 GAN 提出之后，研究人员将深度卷积网络应用到 GAN 中[11]，取得了很好的效果。

1.2.2 超参数优化

如今，研究界希望机器学习系统能够实现一定程度上的自动化[12]，而超参数的自动选择

和优化是机器学习系统自动化中十分重要的一步。在自动超参数调节方面，研究界目前广泛采用的是网格式搜索和随机搜索的方法，并且配合一些提前终止测试的方法，如HyperBand[13]，根据训练过程中的效果，将一些明显不合适的超参数组合提前终止训练，从而提升训练的效率。

1.2.3 GAN 的研究现状

为了克服 GAN 的不稳定性等弱点，国内外研究人员提出了各种 GAN 的变形，如WGAN[2]、BEGAN[3]、LSGAN[4]等。从损失函数、训练方法、训练步骤等方面对原始的 GAN 进行改进，试图降低 GAN 的不稳定性，达到更好的训练效果。

然而谷歌公司的研究[5]指出，不同的 GAN 在相同的资源条件下，将超参数调节到最优并且训练直至稳定后表现出的训练效果相似，并不存在明显的优劣之分。可见，要提高 GAN 的训练效率，研究人员不能仅仅致力于寻找一种更加优秀的 GAN。解决它的超参数优化问题，高效、准确地寻找到表现较优的超参数在 GAN 的应用方面有着重要的意义。

1.2.4 GAN 的现有评价指标

GAN 的评价指标即用来评价 GAN 训练效果的量化指标。在计算机视觉任务中，GAN 的训练效果一般指的就是 GAN 生成图片的质量，包括了图片的清晰度，模式坍塌程度，标签准确率。

目前研究界提出了 Inception Score 和 Frechet Inception Distance[5]这两种指标来评价 GAN 生成的图片质量，这两项指标也是目前使用比较广泛的评价指标。

1.3 本文主要研究内容

本文主要研究一套 GAN 的自动超参数选择方法，并将它实现成为一套超参数优化系统 Auto-GAN，将一套可靠的超参数评价指标作为搜索准则，对 GAN 进行自动化的超参数搜索和优化。研究内容主要包括 GAN 的评价指标、超参数搜索方法和搜索准则对超参数搜索结果的影响三方面。本文的研究路线图如图 1-3 所示。

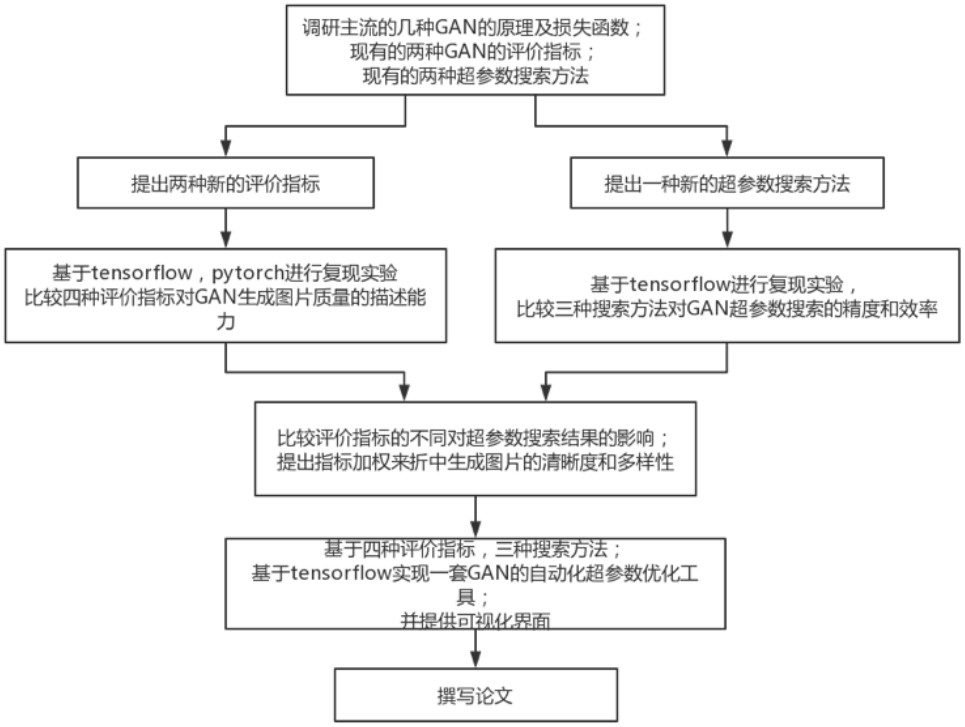


图 1-3 研究路线图

如图 1-3 所示，本文首先对几种主流 GAN 的原理和损失函数，现有的 GAN 的评价指标和超参数搜索方法进行了调研，进而提出了两种新的评价指标和一种新的搜索方法，分别比较了新提出的指标和方法与现有指标和方法之间的优劣。

之后结合两方面，比较了评价指标的不同对超参数搜索结果的影响，提出指标加权来折中生成图片的清晰度和多样性。

最后，基于四种评价指标和三种搜索方法，本文实现了一套 GAN 的自动化超参数搜索优化系统 Auto-GAN，可以用于超参数自动搜索，并且提供了可视化界面。这一系统提供的自动化超参数搜索功能可以大大提升 GAN 超参数优化的效率。

1.3.1 GAN 的评价指标

本文研究现有的这两种 GAN 的评价指标在无监督 GAN 和有监督 GAN 上的表现，并且基于 JS 散度和 Wasserstein 距离提出两种新的 GAN 的评价指标，横向比较了四种不同评价指标的效果，在数字数据集和真实物体数据集上研究了四种评价指标在衡量生成图片清晰度，无监督 GAN 模式坍塌情况和有监督 GAN 标签准确性方面的表现。

针对四种评价指标在评价有监督 GAN 标签准确性时不可靠的问题，本文基于分类网络，给出了标签准确率的计算方法。

1.3.2 超参数搜索方法

在超参数搜索方法方面，本文提出了基于遗传算法思想的超参数搜索算法 exploit &

explore 算法，也实现了现有的网格搜索和随机搜索方法，观察了三种搜索算法在进行 GAN 超参数选择时的效率和效果，横向比较了三种搜索方法在所选超参数分布、有效搜索次数、搜索准确率方面的优劣。最后，对比了不同的评价指标对超参数选择的影响，提出了指标加权对生成图片的清晰度和多样性进行折中。

1.3.3 超参数优化系统实现

本文创新性地采用文中研究的四种 GAN 的评价指标作为超参数搜索的准则，结合本文中研究的三种搜索算法，实现了一套超参数优化系统 Auto-GAN。系统支持了目前典型的几种无监督和有监督的 GAN，并且提供了模型、训练过程和结果的可视化。可以供技术人员进行 GAN 调参，也可以供研究人员进行 GAN 的超参数粗调。相比纯人工调节，本系统能够显著提高调节的效率。

1.4 本文组织结构

本文分绪论、生成式对抗网络及超参数优化方法相关研究、生成式对抗网络评价指标研究、超参数优化方法研究、系统设计与实现、总结与展望六个部分。

第一章绪论。主要介绍了生成式对抗网络的概念，不稳定性和超参数选择对生成式对抗网络训练效果具有的很大影响；指出人工调节超参数效率低下；介绍了深度学习的发展，深度卷积网络强大的特征抽取能力以及它与 GAN 的结合方式；介绍了自动超参数优化的思想和目前常用的超参数搜索方法；介绍了 GAN 的研究现状和评价指标。最后，概括了本文的主要研究内容和组织结构。

第二章生成式对抗网络及超参数优化方法相关研究。首先介绍了 GAN 的相关概念，包括无监督 GAN，有监督 GAN，半监督 GAN，指出了 GAN 容易出现的问题：模式坍塌。之后，介绍了现有的两种 GAN 的评价指标 Inception Score 和 Frechet Inception Distance，从概率统计和线性代数两种角度对 GAN 的生成图片质量进行了描述。最后，介绍了超参数优化目标和现有的两种广泛应用的超参数搜索方法：网格搜索和随机搜索，从理论上比较了它们的优劣之处。

第三章基于评价指标的 GAN 超参数优化方法。首先指出一套合格的生成式对抗网络评价指标不仅仅需要衡量生成图片的质量，还需要衡量无监督 GAN 的模式坍塌程度和有监督 GAN 的生成图片标签准确性。之后，提出了基于 JS 散度的和 Wasserstein 距离的 GAN 评价指标 JS-IS 和 WD，给出了新评价指标的定义式；基于遗传算法思想的 exploit & explore 算法，介绍了算法的原理思想和设计。

第四章 GAN 超参数优化方法实现和综合比较实验。首先，介绍了实验环境，所用的数据集和工具包。

GAN 评价指标实验基于 Tensorflow 提供的 tfgan 工具包计算了传统的 IS、FID 与本文提出的 JS-IS、WD，在 MNIST、CIFAR 等数据集上对无监督 GAN 和有监督 GAN 分别进行了实验，给出了原有的两种指标和新提出的两种指标在训练过程中的变化情况，并对结果做出分析，证明了这四种评价指标在评价 GAN 生成图片质量方面的有效性。发现 JS-IS 的稳

定性优于 IS，但灵敏度不如 IS。又设计了实验，发现四种评价指标中 FID 和 WD 在衡量无监督 GAN 的模式坍塌程度方面比较可靠，其它两种不太可靠。无论是评价清晰度还是模式坍塌程度，WD 的灵敏度在四种评价指标中都是最高的。

而在有监督 GAN 标签准确性方面，四种指标都不可靠，本文提出将分类模型得到的分类准确率作为有监督 GAN 的标签准确率，通过实验证明这种算法在评价标签准确率方面是可靠的。

超参数优化方法实验首先展示了现有的两种超参数优化方法：网格搜索和随机搜索的实现和实验结果，分析了现有的两种方法在 GAN 的超参数优化中展现出的各自的优缺点。之后给出了 exploit & explore 算法的实现以及实验结果。

超参数优化方法比较实验在相同时间开销下，比较了这三种超参数优化方法所选的超参数的分布情况，有效搜索次数和搜索准确率，对实验结果进行了分析和总结，指出 exploit & explore 算法在以上三个方面较传统方法都有较大提升。

同时本文给出了不同的评价指标对实验结果的影响，比较了四种评价指标在评价生成图片质量方面的效果，进而用评价指标加权的方法来对清晰度和多样性进行折中。

第五章系统设计与实现提出了一套 GAN 的超参数优化系统 Auto-GAN，描述了系统的整体架构，展示了系统的使用和可视化的用户界面。

第六章总结与展望对整篇论文的工作内容和创新点，以及研究结论进行了总结。并且对未来的工作做了展望。

1.5 本章小结

本章介绍了本文的研究背景和意义、国内外研究现状、本文的主要研究内容以及全文的组织结构。

第二章 生成式对抗网络及超参数优化方法相关研究

2.1 GAN 相关概念

2.1.1 无监督 GAN

（1）无监督 GAN 基本概念

生成式对抗网络由生成模型 G 和判别模型 D 两部分组成。在 Ian GoodFellow 提出的 GAN 中，G 和 D 为两个多层感知机。在 GAN 中，训练集为一个无标签的数据集，G 的输入为噪声，输出为生成的图片；而 D 的输入为图片，输出为二分类的结果，即 0 表示图片是由 G 生成的，1 表示图片来自真实数据。在 GAN 的训练过程中，G 和 D 通过不断的博弈来优化自己。具体公式如下：

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{2-1}$$

从式 2-1 中可以看出，在 GAN 的训练中，判别模型 D 需要让自己对来自真实数据集的数据 x 的输出 D(x)尽量接近 1，对生成模型 G 所生成的数据 G(z)的输出 D(G(z))尽量接近 0，即让式 2-1 越大，D 的判别能力就越强。而对于生成器 G 而言，它需要让 D 认为它所生成的数据是真实数据，所以 G 需要 D 对它生成的数据的输出尽量大，即让式 2-1 越小，D 就越无法判别 G 生成的数据和真实数据，说明 G 生成的数据与真实数据越相似，G 的生成能力越强。

在实际的训练中，由于一开始 G 的生成能力很弱，D 可以十分轻易地将 G 生成的数据和真实数据区分开来，因此 D(G(z))会十分接近 0。此时 log(1-D(G(z)))趋于饱和，会出现梯度消失，不好训练 G。所以通常在训练中用最大化 log(D(G(z)))的方法来训练 G，这种做法可以达到和式 2-1 一样的效果，并且可以避免 G 在训练前期出现的梯度消失问题，从而达到更好的训练效果。

图 2-1 对 GAN 的训练过程做了形象化的描述。图 2-1 中 z 表示随机噪声，x 表示 G 所生成的数据 G(z)，细虚线表示判别模型 D 的输出分布，圆点线代表真实数据的分布，细实线代表生成数据的分布。可以看出，在训练早期(a)图中，判别模型 D 的输出并不平稳，说明它还尚未能很好地区别真实数据和生成数据。在优化了判别模型 D 后，输出结果变为(b)图所示，这时判别模型 D 的判别能力明显增强了。接下来是在固定判别模型的情况下，优化生成模型 G，可以看出经过优化，真实数据和生成数据分布之间的距离被拉近了，说明 G 正在优化自己，让 D 无法区别真实图片和自己生成的图片。这样的博弈反复进行，最后可以达到(d)图中所示的状态，即经过不断的优化，G 生成的数据分布已经和真实的数据分布重合，D 也再也没有能力将 G 生成的数据和真实数据区分开来，此时 D 的输出为 1/2。

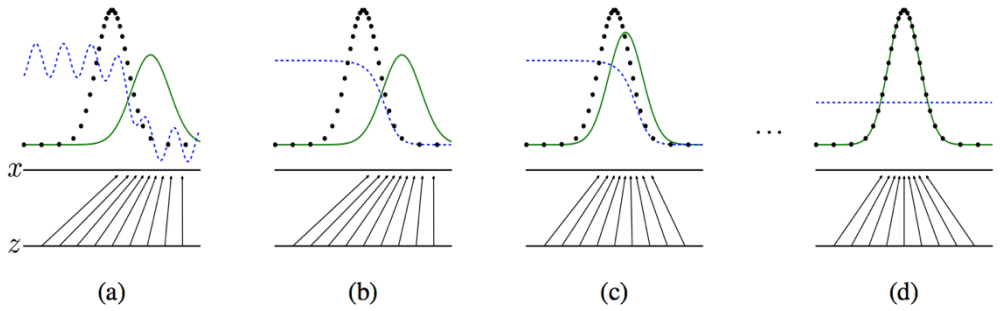


图 2-1 GAN 训练过程[1]

（2）无监督 GAN 改进：DCGAN

在 Ian GoodFellow 的论文中，GAN 的生成模型 G 和判别模型 D 都是多层感知机。在 GAN 提出之后，研究人员将它应用到了计算机视觉的领域，与深度学习相结合，进而提出了 DCGAN[11]。

如 1.1.2 中所提到的，深度学习技术，如卷积神经网络(Convolutional Neural Network)展现出了优秀的特征抽取能力，在图片分类方面有着优异的表现。DCGAN 就将 CNN 与 GAN 进行了结合，用一个 CNN 形成二分类器作为判别模型 D。在生成模型方面，DCGAN 的生成模型示例如图 2-2 所示。

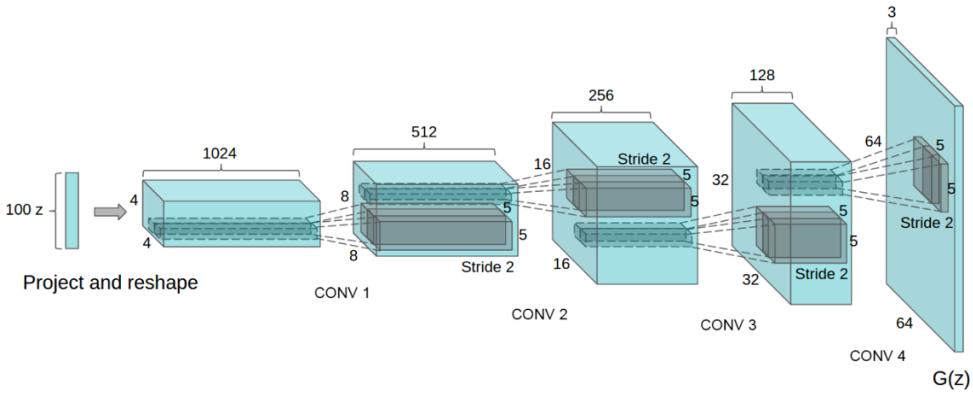


图 2-2 DCGAN 中的生成模型示例[11]

在图 2-2 中，G 的输入为一个 100 维的噪声 z，模型由四层转置卷积层[14]组成。转置卷积操作某种程度上可以看成是卷积操作的逆运算，即转置卷积的前向传播过程相当于卷积操作的反向传播过程。图 2-2 中的 G 通过四层转置卷积层，将一个 100 维的噪声转换为了一个 64*64 的像素矩阵，这与真实图片集的数据大小相同。

在超参数调节得当之后，DCGAN 在各个数据集上的表现都要好于传统的多层感知机 GAN，可谓是无监督 GAN 和深度学习比较成功的一次结合。

在本文之后的实验中，GAN 所用的生成模型 G 和判别模型 D 也均为深度卷积网络。

（3）模式坍塌

模式坍塌(mode collapse)指的是生成模型 G 所生成的数据种类不全，即不能覆盖真实数据的所有种类，这是 GAN 当中一个非常需要关注的问题。在 GAN 提出之后，研究人员发现了模式坍塌问题，并且提出了各种新的 GAN，改进了 GAN 的损失函数和训练方法，如 WGAN[2]，LSGAN[4]，WGAN GP[15]等，或者采用多个 GAN 分别训练，如 AdaGAN[16]，试图解决这一问题，可见模式坍塌问题已经引起了研究界的广泛关注，解决这一问题对提高 GAN 的训练效果具有着十分重要的意义。

因此，在 GAN 的生成图片质量评价方面，研究界也一致将模式坍塌程度作为一个重要的考虑因素。如谷歌公司的工作[5]中就明确指出一个好的 GAN 的评价指标必须能够评价出生成数据的模式坍塌情况。

本文第三章的实验也对模式坍塌予以关注，在对四种 GAN 的评价指标进行了横向比较时，其中一项比较内容即为评价指标对模式坍塌程度的描述能力。

（4）WGAN 与 WGAN-GP

为了改善模式坍塌问题并且提升 GAN 的生成图片质量，研究人员提出了许多 GAN 的改进版本，其中比较典型和常用的是 WGAN 和 WGAN 的改进形式 WGAN-GP[17]。

在标准的 GAN 中，损失函数利用交叉熵来描述生成数据分布和真实数据分布之间的距离，而在 WGAN[2]中，研究人员提出用 Wasserstein 距离来描述这两个分布之间的距离。

$$W(\mathbb{P}_r,\mathbb{P}_g)=\inf_{\gamma\in\Pi(\mathbb{P}_r,\mathbb{P}_g)}\mathbb{E}_{(x,y)\sim\gamma}\left[\|x-y\|\right] \tag{2-2}$$

如式 2-2 所示，π 表示两个分布所能够组合出的所有联合分布的集合，对于每一个联合分布，可以采样 x, y，计算出两者之间距离的期望，而 Wasserstein 距离就是将一个分布拉近到另一个分布所需要的最小的开销。WGAN 采用 Wasserstein 距离，而非 KL 散度或者 JS 散度等作为损失函数，是因为 Wasserstein 距离较后两者具有更加好的平滑特性。在 WGAN 的论文[2]中研究人员举例：如果考虑二维空间中的两个分布，它们分别是两条距离为 θ 的线段上的均匀分布，那么非常容易计算得到：

$$\begin{aligned}&W(\mathbb{P}_0,\mathbb{P}_\theta)=|\theta|,\\&JS(\mathbb{P}_0,\mathbb{P}_\theta)=\begin{cases}\log 2 & \text{if } \theta\neq 0\ ,\\ 0 & \text{if } \theta=0\ ,\end{cases}\\&KL(\mathbb{P}_\theta\|\mathbb{P}_0)=KL(\mathbb{P}_0\|\mathbb{P}_\theta)=\begin{cases}+\infty & \text{if } \theta\neq 0\ ,\\ 0 & \text{if } \theta=0\ ,\end{cases}\end{aligned} \tag{2-3}$$

从式 2-3 可以看出，JS 散度和 KL 散度在 θ =0 处都出现了突变的情况，KL 散度更是达到了无穷大，这是不能接受的。而 Wasserstein 距离体现出的平滑性比两者都要优秀。

由于式2-2在数学上难以求解，可以通过数学变化Kantorovich and Rubinstein对偶性将它转化成式2-4。

$$W(\mathbb{P}_r,\mathbb{P}_\theta)=\sup_{\|f\|_L\leq 1}\mathbb{E}_{x\sim\mathbb{P}_r}[f(x)]-\mathbb{E}_{x\sim\mathbb{P}_\theta}[f(x)] \tag{2-4}$$

式 2-4 中 f 满足利普希茨条件：即对于 f 定义域上的任意两个点有：

$$|f(x_1)-f(x_2)|\leq K|x_1-x_2| \tag{2-5}$$

若 f 满足式 2-5，则称 f 满足利普希茨条件，且 f 的利普希茨常数为 K。

在式 2-4 中，W 代表的就是对于所有的利普希茨常数小于 1 的映射 f，两个分布的最大距离，用这个距离来近似 2-2 中数学上难以求解的 Wasserstein 距离。在深度学习中，f 可以是一个神经网络。

因此，WGAN 较传统的 GAN，主要做了几点变化：

（1）用近似的 Wasserstein 距离代替了原先的交叉熵作为损失函数，在数学上表现为不取 log，此外 D 的任务从二分类任务变为拟合 Wasserstein 距离，为回归问题，因此在 D 的网络结构中，去掉最后一层用于二分类的 sigmoid 层。

（2）由于要让 f 满足利普希茨条件，对神经网络的权重进行 clip 操作，将其限制在某个范围内。

（3）在实际试验中，发现基于动量的优化算法，如 Adam 优化器对 WGAN 的训练效果有时不好，WGAN 使用 RMSProp 或者 SGD 来训练。

WGAN 在提出之后引起了学术界的很大关注，被认为是 GAN 的改进版本中较为成功和典型的实例，在清晰度和减轻模式坍塌方面都有着不错的表现。

而 WGAN-GP 是 WGAN 的一个改进版本，它的主要工作在于在 WGAN 的损失函数上加上了一个梯度惩罚项，如式 2-6。

$$L=\underset{\tilde{\boldsymbol{x}}\sim\mathbb{P}_g}{\mathbb{E}}\left[D(\tilde{\boldsymbol{x}})\right]-\underset{\boldsymbol{x}\sim\mathbb{P}_r}{\mathbb{E}}\left[D(\boldsymbol{x})\right]+\lambda\underset{\hat{\boldsymbol{x}}\sim\mathbb{P}_{\hat{\boldsymbol{x}}}}{\mathbb{E}}\left[\left(\|\nabla_{\hat{\boldsymbol{x}}}D(\hat{\boldsymbol{x}})\|_2-1\right)^2\right] \tag{2-6}$$

由于在损失函数中添加了梯度惩罚项，在 WGAN-GP 的训练过程中，无需再对权重进行 clip 操作。

目前 WGAN-GP 在生成图片方面的效果较其他模型还是较为优秀的，也是如今比较常用的一种 GAN。

在本文第三章的实验中，考虑到训练的稳定性和收敛速度，一般都采用 WGAN 和它的改进形式作为实验所用的无监督 GAN。

(5) BEGAN

BEGAN 是本文的另一个研究对象。之前介绍的几种 GAN 的核心思想都在于拉近生成数据分布和真实数据分布之间的距离，而 BEGAN 的思路与它们不同。

在 BEGAN 中，判别模型 D 是一个自编码器，用式 2-7 表示图像本身和经过自编码器编码之后的图像之间的相似程度。

$$\mathcal{L}(v)=|v-D(v)|^{\eta}\text{ where }\begin{cases}D:\mathbb{R}^{N_x}\mapsto\mathbb{R}^{N_x}\\ \eta\in\{1,2\}\\ v\in\mathbb{R}^{N_x}\end{cases} \tag{2-7}$$

如式 2-7 所示，L 越小，说明原图像和自编码器输出的图像越相似。

由 Jensen 不等式，可以得出 1 阶 Wasserstein 距离的下界，如式 2-8。

$$\inf\ \mathbb{E}[|x_1-x_2|]\geqslant\inf\ |\mathbb{E}[x_1-x_2]|=|m_1-m_2| \tag{2-8}$$

由此，BEGAN 的损失函数可以定义如式 2-9。

$$\begin{cases}\mathcal{L}_D=\mathcal{L}(x;\theta_D)-\mathcal{L}(G(z_D;\theta_G);\theta_D)\\ \mathcal{L}_G=-\mathcal{L}_D\end{cases} \tag{2-9}$$

如式 2-9 所示，BEGAN 的判别模型 D 的训练目标就是让 D 中自编码器对真实数据的误差 L 尽可能趋近于 0，而让对生成数据的误差 L 尽量大，而生成模型 G 的目标则是与 D 进行博弈。

本文第四章将 BEGAN 作为一个重要的研究对象，很关键的一点在于 BEGAN 提出了图像的清晰度和多样性之间存在着一对矛盾，因此引入一个参数来进行控制，如式 2-10。

$$\gamma = \frac{\mathbb{E}\left[\mathcal{L}(G(z))\right]}{\mathbb{E}\left[\mathcal{L}(x)\right]} \tag{2-10}$$

式 2-10 中的 γ 控制的是当整个 BEGAN 达到平衡时，生成数据的 L 和真实数据的 L 之间可以满足一个比值关系，研究人员指出 γ 较小时，生成图片较为清晰，但是多样性差，容易模式坍塌，而在 γ 较大时，生成图片的多样性得到提升，但清晰度不如前。在引入 γ 参数之后，对 BEGAN 的损失函数进行了调整，得到最终的损失函数如式 2-11。

$$\begin{cases} \mathcal{L}_D = \mathcal{L}(x) - k_t.\mathcal{L}(G(z_D)) \\ \mathcal{L}_G = \mathcal{L}(G(z_G)) \\ k_{t+1} = k_t + \lambda_k(\gamma\mathcal{L}(x) - \mathcal{L}(G(z_G))) \end{cases} \tag{2-11}$$

本文第四章中研究了评价指标对清晰度和多样性的侧重性，而 BEGAN 既可以生成清晰度较高但多样性较差的图片，又可以生成清晰度欠佳但富有多样性的图片，是一个十分理想的研究对象。

2.1.2 有监督 GAN

（1）CGAN

由于 GAN 的训练呈现出自由，不太可控的特点，研究人员考虑在 GAN 中添加一些约束条件，提出了 Conditional GAN[18]，简称 CGAN，它的结构如图 2-3 所示。

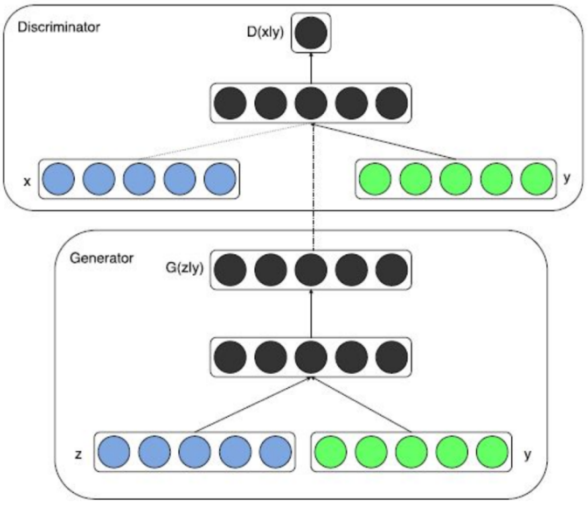


图 2-3 CGAN 结构图[18]

约束条件 y 和噪声 z 一起送进生成模型 G，和数据 x 一起送进判别模型 D，约束条件 y 可以是任何补充的信息，用于引导数据的生成。当约束条件 y 是类标时，CGAN 就是一个以类标为监督信息的有监督 GAN。

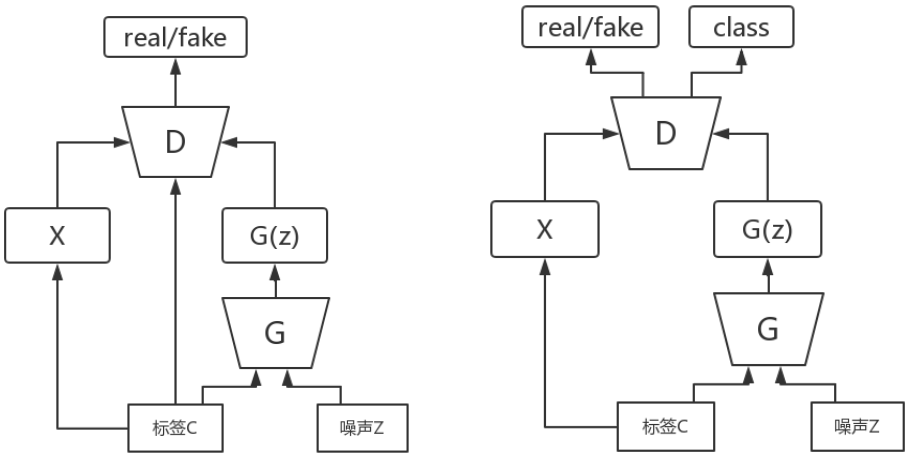
式 2-12 为 CGAN 的具体公式，可以看出 CGAN 的公式即为在原始的 GAN 上加上了约束条件 y。

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \tag{2-12}$$

由于 CGAN 的类标和噪声只是做了一个简单的矩阵连接(concat)运算，在实际实验中，作者发现即使十分仔细地调节过超参数，CGAN 的训练还是存在着很不稳定的现象，不是一个理想的有监督 GAN。

（2）ACGAN

在有监督 GAN 中，2016 年提出的 ACGAN[19]在生成图片的质量方面是比较出色的。图 2-4 为 CGAN 和 ACGAN 的结构示意与对比。



(a) CGAN　　(b) ACGAN

图 2-4 CGAN 与 ACGAN 结构示意[19]

如图 2-4 所示，ACGAN 引入了辅助分类器来提高有监督 GAN 的训练效果。判别模型 D 为多层的卷积神经网络，传统的 CGAN 中 D 为一个二分类器，而 ACGAN 中将 D 的倒数第二层提取的特征分别接入二分类层和 n 分类层，n 为类别总数。实验显示这样做能够较好地提高有监督 GAN 生成的图片清晰度和质量。

由于 ACGAN 在训练的稳定性和图片的生成质量方面都要优于传统的 CGAN，本文第三章的实验中选取了 ACGAN 作为有监督 GAN 的研究对象。本文第五章实现的系统同时支持 CGAN 和 ACGAN。

2.1.3 半监督 GAN

对于部分数据有标签的数据集，研究人员也提出了半监督 GAN，如 SGAN[15]，Triple GAN[20]， Triangle GAN[21]等，由于半监督 GAN 不是本文的研究重点，在此不做过多赘述。

2.2 GAN 评价指标

2.2.1 Inception Score

研究人员提出 Inception Score[22](以下简称 IS)来对 GAN 生成的图片质量进行量化评估，作为一种 GAN 的评价指标。

$$\mathrm{IS}(G) = \exp(\mathbb{E}_{x\sim G}[d_{KL}(p(y \mid x), p(y)]) \tag{2-13}$$

如式 2-13 所示，x＝G(z)为生成器 G 所生成的数据，y 为 x 经过分类器之后的输出，此处的分类器是一个以 ImageNet[7]作为预训练数据集训练的 Inception Net[23]。IS 认为，对于生成的图片，它经过分类器后的输出 p(y|x)的熵要尽量低，而总体的图片应该尽量平均分布于多个类，避免模式坍塌，所以 p(y)的熵要尽量大。因此，IS 用 p(y|x)和 p(y)的 KL 散度来作为两个概率分布之间的距离描述，以此来评价 GAN 的生成图片质量。

研究人员们认为这种指标可以从一定程度上描述 GAN 生成的图片数据的质量，评价结果与人工评价相符合。

IS 公认的缺点主要有无法检测类内的模式坍塌，即如果生成模型对每一类，只生成一张特定的图片，IS 检测不到这种情况，依旧显示该模型为一个很好的生成模型。

本文之后会对 IS 的科学性进行实验分析。

2.2.2 Frechet Inception Distance

Frechet Inception Distance[24]（以下简称 FID）是研究人员在 2017 年新提出的一种 GAN 的评价指标。FID 的思想是：将产生的数据嵌入到 Inception Net 的某一特定层的特征空间中，将嵌入层的输出看成是一个连续多变量的高斯分布，这样，生成数据和真实数据的均值和方差就都是可计算的。进而给出 FID 的定义式如式 2-14。

$$\mathrm{FID}(x, g) = ||\mu_x - \mu_g||_2^2 + \mathrm{Tr}(\Sigma_x + \Sigma_g - 2(\Sigma_x\Sigma_g)^{\frac{1}{2}}) \tag{2-14}$$

如式 2-14 所示，μ_x，μ_g，Σ_x，Σ_g 分别为真实数据和生成数据的均值和方差。研究人员认为 FID 与人工检测的结果十分相符，比 IS 更抗噪声，并且也可以检测出 IS 无法检测的类内模式坍塌。图 2-5 为研究人员给出的基于数据集 CelebA 的 FID 实验结果图，其中 CelebA 是香港中文大学汤晓鸥教授团队发布的大型人脸识别数据集。

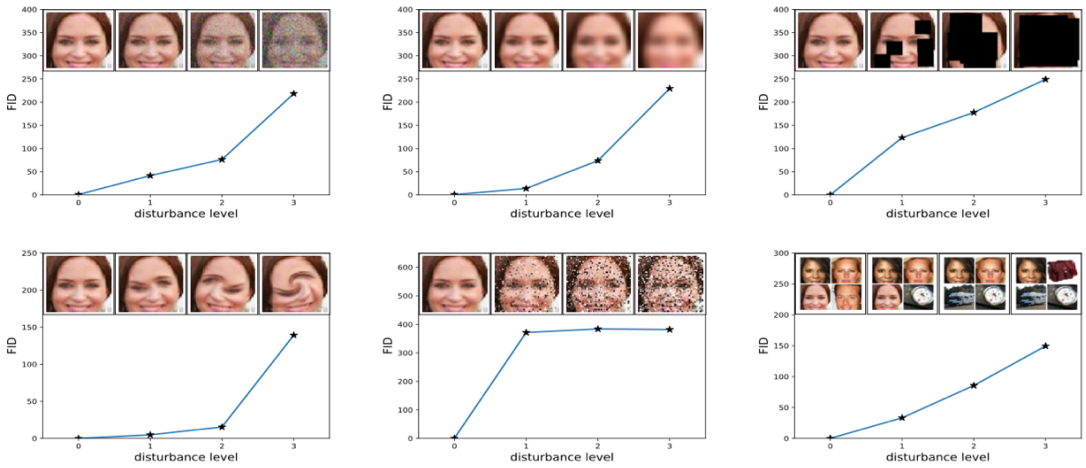


图 2-5 FID 实验结果图[24]

如图 2-5 所示，从左至右，从上至下，分别展示了 FID 对高斯噪声，模糊，黑色矩形遮挡，旋转扭曲，离散噪声，混淆数据的变化。可以看出，FID 在这几种影响下都出现了明显上升，说明 FID 对各种干扰有检测的效果，FID 越高，图片的质量越差，且 FID 的结果和人眼的观感相符。

谷歌公司的研究[5]指出在评价 GAN 时，相比 IS，FID 能够比较客观地检测出生成数据的模式坍塌，具有着重要的意义。GAN 的提出者 Ian GoodFellow 也指出：介于模式坍塌是 GAN 当中比较常见，但也较为严重的问题，因此研究人员在提出新的 GAN 时，必须给出相应的 FID 值作为对模式坍塌情况的评估才能让人信服，可见研究界对 FID 还是颇为认可，将它作为了现在比较常用和公认的 GAN 的评价指标。对于 FID 的科学性，本文之后也会设计实验进行分析。

2.3 超参数优化方法

2.3.1 超参数优化目标

超参数优化的最终目标如式 2-15 所示。

$$\lambda^{(*)} = \underset{\lambda \in \Lambda}{\operatorname{argmin}} \, \mathbb{E}_{x \sim \mathcal{G}_x}[\mathcal{L}\left(x; \mathcal{A}_\lambda(X^{(\text{train})})\right)] \tag{2-15}$$

从式 2-15 可以看出，超参数优化的目标就在于找到某个超参数空间中的一组超参数，使得模型的损失函数期望降到最低，即达到最好的训练效果。

在传统的机器学习中，通常通过损失函数来衡量模型的表现，但 GAN 的训练过程为一个博弈的过程，最终的稳定状态为一种那什均衡状态，用损失函数来评价显然是不合理的。因此，本文采用 GAN 的评价指标来作为搜索准则，超参数优化的目标就是在一个给定的离散的超参数空间中，自动、高效地找到使评价指标达到最优的超参数组合，将它作为 GAN 的超参数。

下面介绍两种常用的超参数搜索方法：网格搜索和随机搜索。

2.3.2 网格搜索

网格搜索[25]是一种较为古老的超参数搜索方法。即在超参数空间中对每一维的超参数进行遍历，从中找出训练效果最好的超参数组合。如果说参数的取值是连续的，则考虑在连续取值上进行等间隔采样。

这种方法的主要优点是：

（1）执行简单；

（2）可以在计算机集群上进行分布式并行的实现来降低时间的开销；

（3）在低维空间(如 1 维、2 维空间)中具有很高的可靠性。

但网格搜索的缺点也十分明显：当超参数维数增加时，超参数的组合会指数性增加，此时该种方法会消耗大量的时间或者计算资源。

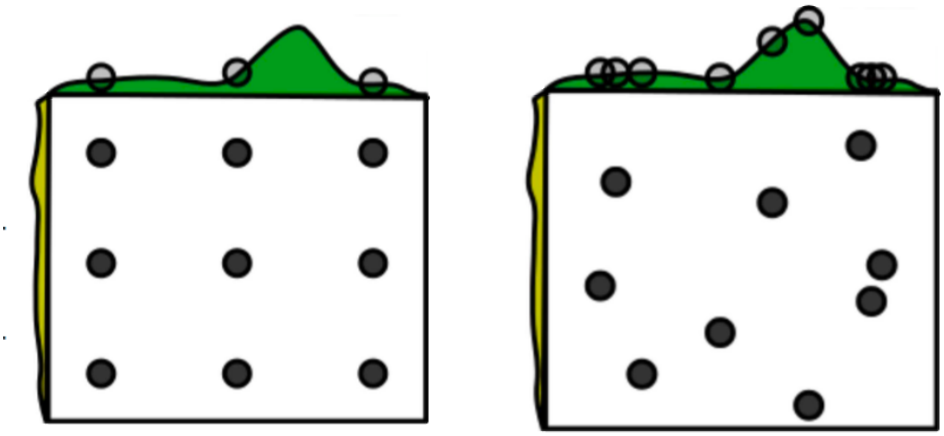
在很长一段时间内，网格搜索结合手工调优是研究人员普遍采用的超参数调优的方法，比如 Hinton 在 2010 年还提出了用这种方法来优化玻尔兹曼机[26]，一些机器学习的工具包，比如 libsvm[27]和 scikits.learn 也采用了这种方法。

可见，网格搜索作为一种提出较早的超参数搜索方法，具有着良好的并行性和可靠性，也已经在科研和工程实践中都得以应用。但当超参数组合较多时，这种方法将带来大量的资源开销。

本文在第四章中会对网格搜索进行复现实验：应用网格搜索进行 GAN 的超参数搜索。

2.3.3 随机搜索

随机搜索[28]是一种在 2012 年提出的超参数优化方法，它的思想是每次在超参数空间当中随机去一个点作为算法的超参数进行训练，在经过若干次随机采样之后取其中最优的一组超参数，图 2-6 展示了随机搜索的原理并且将它与网格搜索进行了比较。



(a) 网格搜索　　(b)随机搜索

图 2-6 网格搜索与随机搜索原理图[28]

从图 2-6 可以看出网格搜索和随机搜索的原理对比。图中，上方函数为待优化的目标式子，矩形表示一个二维的超参数空间，为表述方便不妨称横向的为第一维，纵向的为第二维。图中用两种方法分别进行了 9 次取样来求上方的函数的最大值。在网格搜索中，第一维和第二维的参数分别只能有 3 种取值，这种做法很容易错过函数的最大值。而在随机搜索中，第一维和第二维的参数都有 9 种取法，更加容易找到最大值。

研究人员指出随机搜索比网格搜索在同样的资源开销下可以找到更加多样的点，从而更有可能找到最值。而且介于随机搜索可以指定取样的次数，在资源有限的情况下，随机搜索比网格搜索更加适宜于进行超参数调优，此外，它也可以在保证一定的训练质量的前提下，节省超参数搜索的时间。随机搜索也是目前研究人员较为青睐的一种超参数搜索的方法。

在本文的研究中，GAN 所选取的超参数均采用离散分布，因此，随机搜索并不能够覆盖比网格搜索更多的点。第四章中进行的随机搜索实验主要研究：在一定程度上保证 GAN 的生成图像质量的情况下，随机搜索在效率方面能否较网格搜索有较大的提升。

2.4 深度学习框架

深度学习在各类人工智能任务中展现出了超凡的魅力，随之市面上涌现出了大量的深度学习框架供研究界和工业界使用，比较常用的框架包括 Tensorflow，Pytorch，Keras，Caffe 等。本文第四章的实验主要基于 Tensorflow 和 Pytorch 进行开展，第五章的 Auto-GAN 工具在 Tensorflow 平台上开发。下面简单介绍这两种框架。

2.4.1 Tensorflow 框架

Tensorflow 框架是目前应用最为广泛的一种深度学习框架。最早由 Google Brain 团队开发，于 2015 年 11 月开源。在开源之后，Tensorflow 也成为了 Github 上人气最高的深度学习框架。

Tensorflow 的计算流为静态图，支持 python，C++等语言，可以在 GPU 和 CPU 上进行工作，具有完善的文档，并且提供了 Tensorboard 进行可视化，是如今工程开发中首选的深度学习框架。

2.4.2 Pytorch 框架

Pytorch 是本文中用到的另一个深度学习框架，它对 python 语言提供了良好支持，是一种动态图框架。

自 2017 年 1 月开源之后，Pytorch 成为了研究界的宠儿，由于 Pytorch 具有十分友好的调试功能，使用起来也很好上手，许多研究人员都乐于用 Pytorch 来实现算法，进行研究工作。

2.5 本章小结

首先介绍了 GAN 的相关概念，无监督 GAN，有监督 GAN，半监督 GAN 各自的原理，并对这三类 GAN 进行了举例，详细介绍了几种常用的无监督 GAN 和有监督 GAN，说明了在第三章的实验中将采用哪些种类的 GAN 作为研究对象，以及选择这些 GAN 的出发点和原因。

之后介绍了现有的两种 GAN 的评价指标 IS 和 FID，阐述了它们各自的原理。介绍了现有的两种超参数优化方法：网格搜索和随机搜索，分析了二者的优缺点和现有应用，指出本文中研究的超参数优化问题为离散空间中的优化问题。

第三章 基于评价指标的 GAN 超参数优化方法

3.1 研究框架

本文的研究主要分为评价指标与比较实验、搜索算法与比较实验、评价指标的不同对搜索结果的影响以及 GAN 超参数优化系统与可视化界面，研究框架如图 3-1 所示。

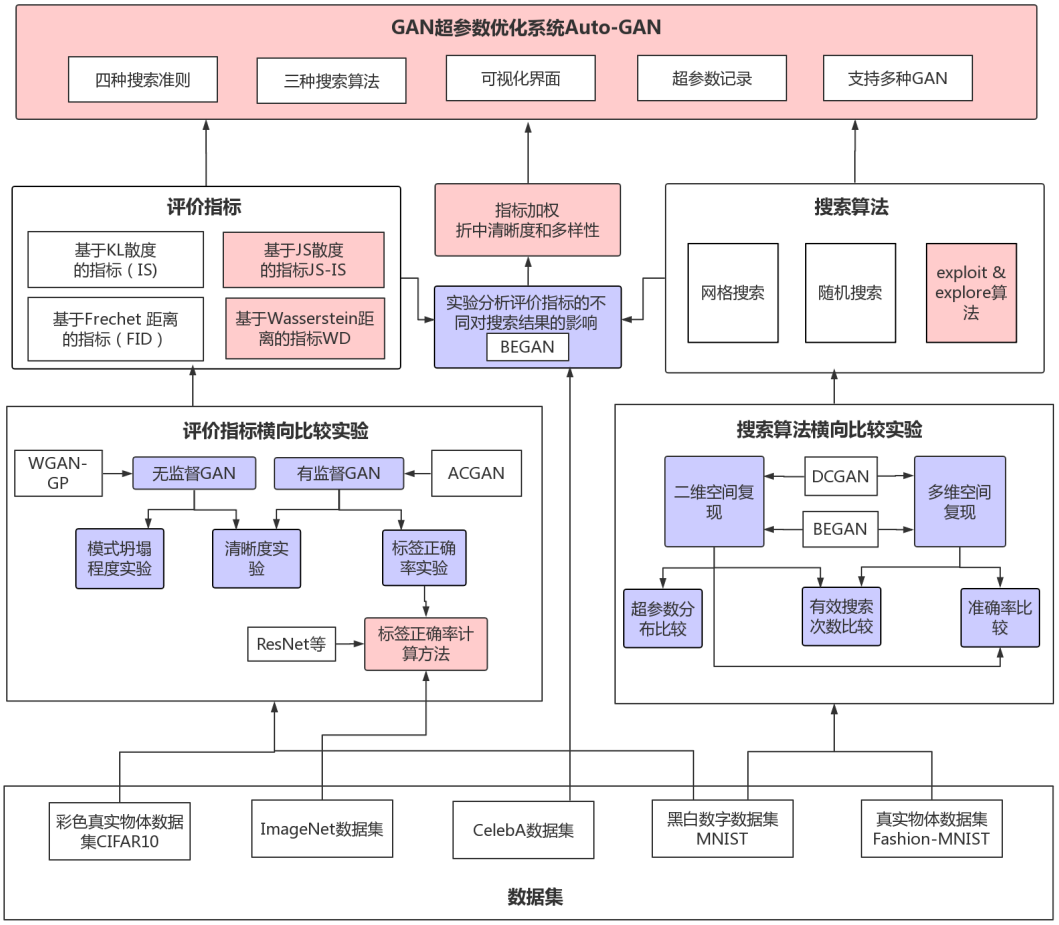


图 3-1 研究框架图

本文的第三章研究评价指标和搜索算法。在评价指标方面，本文分析了 GAN 的评价指标所需要描述的内容，基于 JS 散度和 Wasserstein 距离分别提出了两种新的评价指标 JS-IS、WD。在搜索算法方面，本文基于遗传算法的思想，提出了一种新的超参数搜索算法 exploit

& explore 算法，描述了算法的思想和设计。

本文的第四章进行了综合比较实验。在评价指标的横向比较实验中，将现有的评价指标 IS、FID 和本文提出的两种新的指标 JS-IS、WD 共 4 种指标进行了横向比较，选取了彩色真实物体数据集 CIFAR10 和黑白数字数据集 MNIST 两种比较有代表性的数据集，在无监督 GAN 和有监督 GAN 上分别进行了实验。考虑到需要 GAN 生成质量较好的图片，无监督 GAN 实验选用 WGAN-GP，有监督 GAN 实验选用 ACGAN。实验的内容包括了清晰度，无监督 GAN 模式坍塌程度和有监督 GAN 的标签准确性。最后，利用 ImageNet 数据集和 ResNet 等分类网络给出了一套可靠的标签正确率计算方法。

在搜索算法的横向比较实验中，在数字数据集 MNIST 和真实物体数据集 Fashion-MNIST 上，以 DCGAN 和 BEGAN 为实验对象进行实验，对传统的网格搜索、随机搜索以及本文提出的 exploit & explore 算法分别在二维空间和多维空间中进行了实现和比较，比较的内容包括了有效搜索次数和准确率，此外在二维空间中还比较了所选超参数的分布情况。

第四章第五小节研究了评价指标的不同对搜索结果的影响，在人脸数据集 CelebA 上对 BEGAN 进行了实验，总结不同评价指标对生成图片清晰度和多样性的侧重，给出指标加权的方法在清晰度和多样性之间进行折中。

最后，本文以四种 GAN 的评价指标作为超参数的搜索准则，利用三种搜索方法，实现了一套 GAN 超参数优化系统 Auto-GAN，在第五章中给出了系统的详细描述和使用展示。此外，第五章中也展示了系统的可视化界面和超参数记录。这样的一套自动化的超参数优化系统支持多种 GAN，是目前市面上所没有的。运用它进行超参数调节较传统的纯手工调优大大提高了超参数优化的效率。

3.2 GAN 评价指标

3.2.1 无监督 GAN 评价指标

（1）生成图片的清晰度

GAN 在提出之后被广泛应用到了图像翻译 [29]、残缺图片补全[30]、风格迁移[31]等领域，这些应用都对 GAN 所生成的图片的清晰度提出了要求。生成图片的清晰度必定是评价一个生成模型好坏的重要指标，同样，一项可靠的 GAN 的评价指标对生成图片质量的描述也必须和人眼对图片的感知相符。

之前，研究人员就已经指出，在深度学习中一些量化的数据可能会与人眼观察出的实际效果存在很大偏差。在本文的实验中，作者对实验中记录的评价指标对应的 GAN 生成的图片都做了展示，通过人眼可以直接对生成图片的清晰度做出判断，以体现评价指标与生成图片清晰度之间的关系。

（2）模式坍塌的程度

如 2.1.1 中所提到的，无监督 GAN 中很大的一个问题就是模式坍塌的问题，这个问题引起了研究界的广泛关注，需要进行改善和解决。权威机构如谷歌公司，研究人员包括 GAN 的提出者 Ian GoodFellow 也都认为在评价 GAN 时必须考虑模式坍塌的程度，如果模式坍塌严重，那么这个 GAN 就不能称为一个合格的模型。可见，GAN 的评价指标也必须能够有效地反映出模型所生成图片的模式坍塌情况。

从 IS 的定义式就可以看出，它无法感知到类内的模式坍塌，这是 IS 一直被诟病的一个重大的缺点。

本文第四章设计了实验，使实验对象 GAN 出现不同程度的模式坍塌，观察评价指标的变化以比较不同评价指标对模式坍塌的描述能力。

3.2.2 有监督 GAN 评价指标

（1）生成图片的清晰度

无论是无监督 GAN 还是有监督 GAN，其生成图片的清晰度都是衡量一个模型好坏的重要指标，因此，有监督 GAN 的评价指标也必须能够描述生成图片的清晰度，评价结果要符合人眼观感。

和无监督 GAN 的实验相似地，作者也通过实验观察了有监督 GAN 评价指标与生成图片清晰度之间的对应关系。

（2）生成图片标签的正确率

在有监督 GAN 中，由于生成图片的标签是指定的，有了这项约束，无需再考虑模式坍塌的问题。这里，我们有另外一个需要考虑的问题：生成模型生成的图片的标签正确率，即如果生成模型再给定标签 A 的情况下，如果生成图片是 B 类的，那么即便它生成的图片清晰度很高，这也是一个不合格的模型。可见，GAN 的评价指标要能够反应有监督 GAN 生成图片的标签正确率。

目前，还未有研究人员检验过现有的评价指标对有监督 GAN 生成图片标签的正确性的描述能力。本文第四章也设计了实验，分析评价指标和生成图片标签正确性之间有无对应关系。

3.3 JS-IS 评价指标

3.3.1 KL 散度和 JS 散度

3.3.1.1 KL 散度

Kullback-Leibler 散度，简称 KL 散度[32]，又称相对熵，是对两个概率分布差别的一种度量。相对熵的概念在信息论中解释如下：对于两个概率分布 P 和 Q 而言，两者的 KL 散度表示用 Q 的编码来编码 P 的样本所需要的额外的位元数。在机器学习中，KL 散度也常常用来描述两个分布之间的距离。KL 散度的在离散空间中的具体定义式如式 3-1。

$$D_{\mathrm{KL}}(P\|Q) = -\sum_{i} P(i)\ln\frac{Q(i)}{P(i)} \tag{3-1}$$

从式 3-1 可以得出 KL 散度的如下性质：

（1）非负性

KL 散度当且仅当 P=Q 时为 0。这种性质和它表示的两个分布之间的距离的意义是十分

相符的。

$$D_{\mathrm{KL}}(P\|Q) \geq 0, \tag{3-2}$$

（2）非对称性

KL 散度的另一特点是它不具有对称性，即式 3-3。

$$D_{\mathrm{KL}}(P\|Q) \neq D_{\mathrm{KL}}(Q\|P) \tag{3-3}$$

如 2.2.1 中所提到的，IS 指标在描述 p(y|x)和 p(y)这两个概率分布的距离时，采用的就是 KL 散度。KL 散度的非对称性和无界性导致 IS 指标在一些极端情况下会由于 KL 散度过大导致溢出等问题。

3.3.1.2 JS 散度

Jensen-Shannon 散度，简称 JS 散度，它是 KL 散度的一种变形，它的定义式如式 3-4 所示。

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2}D(P \parallel M) + \frac{1}{2}D(Q \parallel M) \quad \text{其中} \quad M = \frac{1}{2}(P+Q) \tag{3-4}$$

在式 3-4 中，D 表示两个分布之间的 KL 散度，M 为两个分布的平均值。

相比KL散度，JS散度与KL散度原理相似，但具有对称性和相对平滑的特征，是KL散度的一种改进。JS的主要特点有以下两点：

（1）对称性

JS具有KL所没有的对称性。

（2）有界性

对于两个分布P, Q而言，它们之间的JS散度是有界的。如式3-5所示。

$$0 \leq \mathrm{JSD}(P \parallel Q) \leq 1 \tag{3-5}$$

3.3.2 基于 JS 散度的 IS 指标 JS-IS

如3.2.1中提到的，JS散度相比KL散度具有更好的数学性质，本文基于IS指标的思路，基于JS散度提出一种IS指标的变形JS-IS，定义式如式3-6。

$$JS\text{-}IS(G) = exp(E_{x\sim G}[2d_{JS}(p(y|x),p(y))]) \tag{3-6}$$

如式3-6所示，本文提出的IS-JS把p(y|x)与p(y)间的JS散度作为了p(y|x)和p(y)间的距离度量，此处乘以2是为了免去一次乘法运算，由于单调性，并不影响式子的意义。

在理论上，JS-IS相比IS具有以下优势。

（1）稳定性

JS散度具有对称性和比KL散度更好的平滑性，这使基于JS散度的评价指标JS-IS比基于KL散度的IS也会有更好的稳定性。

（2）有界性

如3.2.1所言，对于任意两个分布，它们之间的JS散度都是有界的。因此，任何情况下，JS-IS都会被限制在一个范围内，不会出现溢出或者无法计算的情况。

3.4 WD 评价指标

3.4.1 Wasserstein 距离

Wasserstein距离又称推土机距离[33]，形象来说，推土机距离指的是将一个分布转移到另一个所需要的最小的开销。p阶Wasserstein距离的定义式如式3-7。

$$W_p(\mu,\nu) := \left(\inf_{\gamma\in\Gamma(\mu,\nu)}\int_{M\times M} d(x,y)^p \,\mathrm{d}\gamma(x,y)\right)^{1/p} \tag{3-7}$$

对于两个高斯分布而言，2阶Wasserstein距离可以用来描述分布之间的距离。2阶Wasserstein距离的具体定义如式3-7。

$$W_2(\mu_1,\mu_2)^2 = \|m_1 - m_2\|_2^2 + \operatorname{trace}\left(C_1 + C_2 - 2\left(C_2^{1/2}C_1C_2^{1/2}\right)^{1/2}\right) \tag{3-8}$$

如式3-8所示，μ_1和μ_2分别服从$N(m_1,C_1)$和$N(m_2,C_2)$的高斯分布，Wasserstein距离描述了两个分布之间的距离。

3.4.2 基于 Wasserstein 距离的评价指标 WD

Wasserstein距离一直都以其良好的数学性质和鲁棒性收到研究界的青睐。受FID的启发，本文提出将生成的数据和真实数据嵌入到与FID中相同的特定的一层Inception Net中，将提取的特征看成两个多变量高斯分布，用3.3.1中的2阶Wasserstein距离的思想来描述两个分布之间的距离，称为WD。定义式如式3-9。

$$\mathrm{WD}(x,g) = ||\mu_x-\mu_g||_2^2 + \mathrm{Tr}(\Sigma_x+\Sigma_g-2(\Sigma_x^{1/2}\Sigma_g\Sigma_x^{1/2})) \tag{3-9}$$

如式3-9所示，μ_x，μ_g，Σ_x，Σ_g分别为真实数据和生成数据的均值和方差，WD用2阶Wasserstein距离描述了真实数据分布和生成数据分布之间的距离，作为一种新提出的GAN的评价指标。

本文提出WD，是希望能够借2阶Wasserstein距离良好的数学特性对真实数据分布和生成数据分布进行描述。

3.5 exploit & explore 算法

3.5.1 算法思想介绍

exploit & explore 算法是本文提出的一种新的优化 GAN 超参数的方法。它的思想类似于遗传算法，但略有不同。遗传算法主要是用较好的解之间的交配变异等来寻找较优解的，类似的，exploit & explore 也分为两步：

（1）exploit 过程

exploit & explore 算法在训练过程中与随机搜索一样，也是随机地在超参数空间中选取一点作为一次训练的超参数，通过一个评价指标来衡量当前 GAN 的训练效果好坏，记录目前为止达到的最好的训练效果和对应的超参数组合。在 exploit 过程中，如果当前尝试的超参数组合训练效果优于过去的最好效果，那么就 exploit 当前的超参数组合为最优的超参数组合，并且记录下最优的训练效果。Exploit 过程和随机搜索在原理上是完全相同的。

（2）explore 过程

exploit & explore 算法的另一部分为 explore 过程，这一过程类似于遗传算法中的变异过程。即如果找到了更优的超参数组合，在进行完 exploit 过程之后，将超参数在各个维度上进行一个微小的扰动（比如乘上一个数，或者加上一个数等等），将扰动后的超参数添加到待选的超参数列表中，即相当于将优胜劣汰之后得到的较优的基因进行微小的变异，这些基因有很大可能也是比较好的超参数，用它们来扩充超参数空间，explore 过程让超参数空间中比较优秀的超参数组合所占的比例提升，并且很有可能 explore 到新的更加好的点，这样无疑是很有利于提升超参数空间的质量的。

3.5.2 算法设计

exploit & explore 算法同样限定了训练模型的次数，在规定了次数，基本上限定了时间和资源的开销的情况下来进行超参数的优化。首先和随机搜索相同，随机选取超参数空间中的一个点进行训练，计算评价指标来评价模型的训练结果，如果模型的训练结果优于之前所有的模型，则记录下模型的训练效果和对应的超参数。

然后进行 explore 操作，对于现在得到的新的，也是迄今最优的超参数 best_param 做一个小的扰动 disturb，这个扰动可以是对学习率乘上一个类似于 1.1，0.9 等的数字，或者对 D 的学习次数加上 1 或者减去 1 等等，类似于遗传算法中的变异，所以扰动一般是比较微小的。之后将扰动后得到的 added_param 作为一组新的超参数，训练一个新的模型，如果新的模型效果比现在的更好，则继续更新模型的最佳训练效果和所对应的超参数组合。此外，将扰动过的超参数添加到超参数空间的各个维度上，这样，这些新加入的参数与其他原有的超参数的组合，就有可能在下一次随机选取时被选中。这样，超参数空间这个基因库中，优秀的基因的比例就会逐渐上升，选取到比较好的超参数的概率也就相应增大了。

算法的流程图如图 3-2。

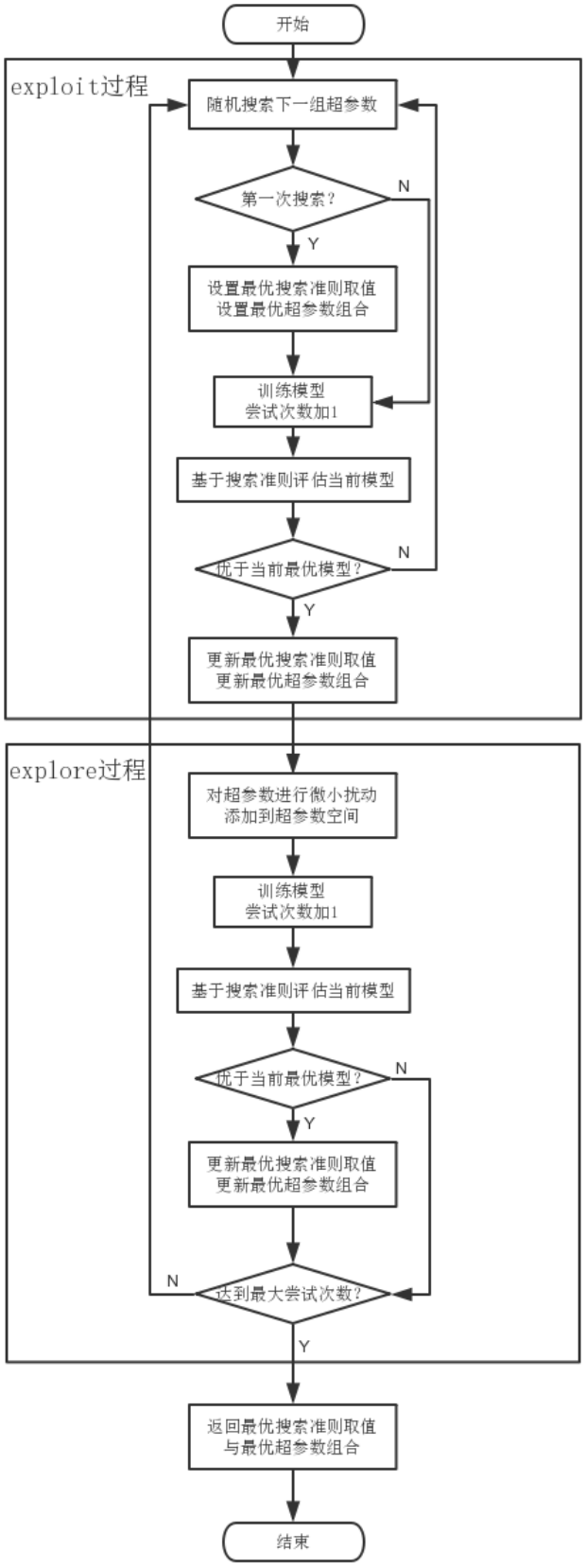


图 3-2 exploit & explore 算法流程图

3.6 本章小结

首先，指出一项可靠，成熟的 GAN 的评价指标必须能够同时反映生成图片的清晰程度，无监督学习中模式坍塌的程度和有监督学习中的标签准确性。

之后，基于 JS 散度和 Wasserstein 距离提出了两种新的评价指标 JS-IS 和 WD，给出了评价指标的定义式。

最后，基于遗传算法的思想，提出了一套新的超参数搜索算法 exploit & explore 算法，介绍了算法的设计思想，给出了流程图。

第四章 GAN 超参数优化方法实现和综合比较实验

4.1 实验环境、数据集与工具包

本文中所有实验的实验环境和所涉及的数据集如下。

4.1.1 实验环境

本文中的实验均在linux服务器上完成，深度学习框架方面，在进行实验时，考虑到Pytorch友好的调试功能为科研实验带来了很大的便利，使用了Tensorflow和Pytorch两种框架。而在工具开发时，采用了如今工业界最为常用的框架Tensorflow进行工具开发，服务器配置如下：

操作系统：ubuntu 16.04
内存：512G
显卡：Titan X
Python版本：2.7
Tensorflow版本：1.4.1
Pytorch版本：0.2.0

4.1.2 数据集

（1）MNIST数据集

MNIST数据集是一个在机器学习实验中使用极其频繁的数据，它是一个分辨率为28*28的黑白手写数字图片数据集，包括了60000张训练图片和10000张测试图片。

MNIST数据集在机器学习实验中可以算是一个必测的数据集。来自机器学习、机器视觉、人工智能、深度学习领域的研究员们把MNIST数据集作为衡量算法的基准之一。

考虑到MNIST数据集为0-9的数字，可以简单地用人眼分为10类，将它作为无监督、有监督GAN实验中一个基础的黑白数字数据集，用于衡量生成图片质量，模式坍塌和标签准确性。

（2）Fashion-MNIST数据集

Fashion-MNIST是一个MNIST数据集的变形，展示了10类，共70000张不同商品的正面图。与MNIST相同，它的分辨率为28*28，包括了60000张训练图片和10000张测试图片。

介于MNIST数据集为黑白数字数据集，相对比较简单，Fashion-MNIST数据集和MNIST数据集像素规格完全相同，同样具有10类，可以通过人眼进行分类，但具有更加复杂的轮廓特征。作者在实验中将Fashion-MNIST作为MNIST数据集的一个补充。

（3）CIFAR10数据集

CIFAR10 数据集由 Geoffrey Hinton 等人提出，是一个彩色的数据集，包括了 10 类，共 60000 张 RGB 图片，其中 50000 张训练图片，10000 张测试图片，类别包括飞机，汽车，鸟，猫等，分辨率为 32*32*3。与 MNIST 不同，这是一个更加普适的数据集，描述了现实世界中的物体，也是如今机器学习实验中非常常用的一个数据集。之后 CIFAR10 也扩展到了 CIFAR100，具有 100 类。

本文暂时只使用 CIFAR10 作为实验数据集，由于 CIFAR10 是一个彩色真实物体图片数据集，同样可以通过人眼或者预训练过的分类网络进行分类，在实验中将它作为实验所用的真实物体图片数据集，和 MNIST 一起从数字和真实物体两个方面对无监督、有监督 GAN 分别进行实验。

（4）ImageNet 数据集

ImageNet[7]是另一个在机器学习界大名鼎鼎的数据集，它由著名的计算机科学家，斯坦福大学的李飞飞教授发起，也是现在世界上进行计算机视觉实验时使用极其频繁的数据集。现在可以通过网络下载到包含 1000 类图片的 ImageNet 数据集，其中的图片均为真实图片，可以通过 opencv 等库函数进行裁剪大小，读取数据。

作者考虑 ImageNet 大而全的特点，将它作为分类网络的预训练数据集。经过 ImageNet 海量数据的训练的预训练网络具有很好的分类能力，作者将它作为分类网络，用于测试生成图片数据集，得到它的标签准确性。

（5）CelebA数据集

CelebA 是由香港中文大学汤晓鸥教授实验室公布的大型人脸识别数据集。该数据集包含有 200K 张人脸图片，人脸属性有 40 多种，主要用于人脸属性的识别。

CelebA 数据集中的图片结构较为规整，清晰度较高，人脸细节也很丰富，可以很好地比较出生成图片的清晰度和多样性。考虑到人脸数据集的性别，肤色等特征不方便分成多类，本文将 CelebA 数据集作为 BEGAN 的训练数据集。

4.1.3 tfgan 工具包

tfgan是谷歌公司开源的一个轻量级的工具包，对GAN的一些评价指标，损失函数等都进行了封装，让用户可以更加方便地训练和评估GAN。本次实验中的一些代码也是利用或者修改tfgan工具包来实现的。tfgan工具包的主要代码结构如图3-1所示。

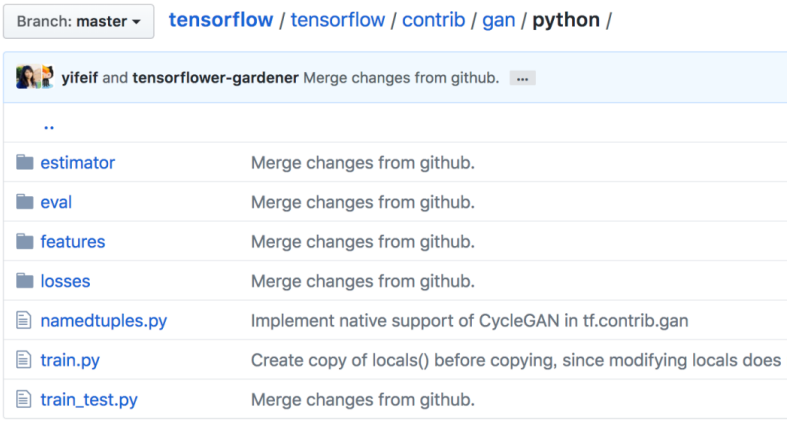


图4-1 tfgan代码结构

图4-1展示了tfgan工具包的主要部分，包括estimator、eval（评价指标）、losses（损失函

数）等。本次实验中，主要利用和改写eval中的方法来实现GAN的评价。

在 eval 文件夹下的 classifier_metrics_impl.py 文件中实现了 IS、FID 等评价指标。在本次实验中，又补充修改了一部分的源码，额外实现了 JS-IS、WD 两种指标。

4.2 GAN 评价指标实验

目前，研究界普遍采用 IS 和 FID 等评价指标来进行模型之间的优劣比较，但对不同评价指标之间的横向比较依旧较少。

本部分的实验目的在于对四种评价指标进行横向比较，分别在数字数据集和真实图片数据集上对无监督 GAN 和有监督 GAN 进行实验，观察四种指标与生成图片质量之间的对应关系。

4.2.1 GAN 评价指标计算

实验对 tfgan 的代码进行了改写，计算了四种 GAN 的评价指标。下面给出部分核心代码，展示四种评价指标的实现过程。IS 和 JS-IS 的核心代码思路如下：将图片数据嵌入到 Inception Net 中之后，会得到一个向量，即代码中的 logits，让其经过 softmax 运算得到概率矩阵 p，即 p(y|x)，在第 0 维上取平均值，得到 q，即 p(y)。之后，分别计算 p 和 q 之间的 KL 散度和 JS 散度，进而计算 IS 和 JS-IS 的值。

FID 和 WD 的核心代码思路如下：函数的输入是将真实数据和生成数据嵌入到 Inception Net 的一层之后抽取到的特征，FID 和 WD 将它们看成是多变量的高斯分布。在进行了数据检查之后，先计算了两个分布的均值和方差，之后按照 FID 和 WD 的公式计算了 FID 和 WD 的值进行返回。

4.2.2 无监督 GAN 实验

（1）数字数据集实验

无监督GAN的数字数据集实验在MNIST数据集上用WGAN-GP进行，输入的噪声维度为62，batch_size为64，优化器为RMSProp，其中学习率为0.00005，β_1为0.5，β_2为0.9。

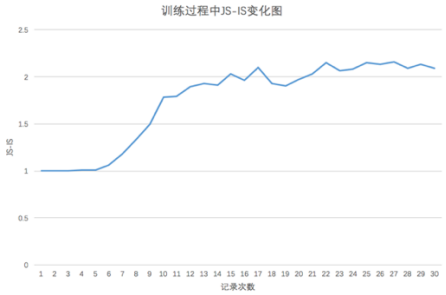
实验中，每20个迭代进行一次数据记录，计算出四种评价指标的值。在进行了1000次迭代之后，生成的图像质量已经趋于稳定。

四种评价指标在训练过程中的值的变化情况如表4-1，图4-2所示，对应的生成图片质量如图4-3所示。

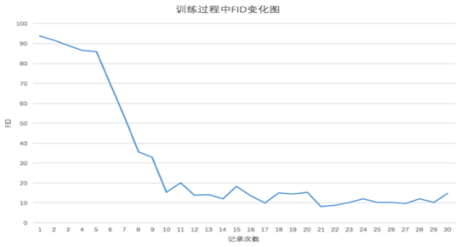
表4-1 无监督GAN评价指标随训练变化(MNIST)

记录次数	IS	JS-IS	FID	WD
5	1.014	1.007	85.950	88.850
10	3.404	1.783	15.353	-647.041
15	4.431	2.029	18.453	-868.966
20	4.268	1.967	15.176	-791.619
25	5.106	2.153	10.142	-1274.514
30	4.976	2.093	14.674	-1016.718

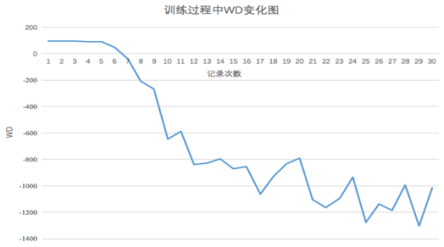
（a）IS 变化图



（b）JS-IS 变化图

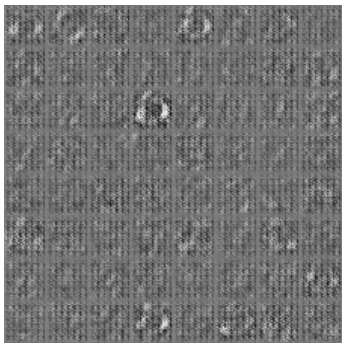


（c）FID 变化图



（d）WD 变化图

图 4-2 无监督 GAN 训练过程中四种评价指标变化图(MNIST)



（a）第 5 次记录

（b）第 10 次记录

（c）第 15 次记录

（d）第 20 次记录

（e）第 25 次记录

（f）第 30 次记录

图 4-3 无监督 GAN 训练中生成图像(MNIST)

从表 4-1，图 4-2 可以看出，随着训练的进行，IS 和 JS-IS 上升，FID 和 WD 下降，IS 的取值范围比 JS-IS 宽，WID 的取值要比 FID 宽的多。从图 4-3 可以看出，随着迭代的进行，生成图片从（a）中的噪声渐渐变清晰，最后趋于平稳。可以得到这样的正相关关系：图片越清晰，IS 和 JS-IS 越大，FID 和 WD 越小。

如 2.2.2 中所提到的，FID 的原理主要是将生成数据和真实数据嵌入到一层神经网络中进行特征提取，将提取的特征看做多变量的高斯分布，从而，实际上各种数学距离都可以用来描述两个高斯分布之间的距离，WD 就是运用了 2 阶 Wasserstein 距离对两个高斯分布距离进行描述的。从图 4-2 可以看出，WD 对图片质量变化的灵敏度比 FID 要高，但 FID 的稳定性也要比 WD 好一些。

综上，在评价无监督 GAN 在黑白数字数据集上的训练效果方面，IS、JS-IS、FID 和 WD 都有一定的意义。WD 的灵敏度比其他三种指标要好。

（2）真实物体图片数据集实验

无监督GAN的真实物体图片数据集实验在CIFAR10数据集上用WGAN-GP进行。输入的噪声维度为110，batch_size为100，优化器为RMSProp，其中学习率为0.00005，β_1为0.5，β_2为0.9。

实验中，每 10 个 epoch 进行一次数据记录，计算出四种评价指标的值。结果如表 4-2，图 4-4。

图 4-5 展示了第 1、2、4、7 次记录时生成的图片。

从表 4-2 和图 4-4 可以看出，真实图片数据集 CIFAR10 上，无监督 GAN 的训练过程中四种评价指标总体上的变化方向与数字数据集上的相同，均为 IS 和 JS-IS 上升，FID 和 WD 下降。

表4-2 无监督GAN评价指标随训练变化(CIFAR10)

记录次数	IS	JS-IS	FID	WD
1	2.47	1.527	235.821	-738.881
2	3.692	1.747	86.44	-1760.94
3	3.466	1.708	67.433	-1581.11
4	3.3	1.677	70.527	-1539.548
5	3.424	1.696	70.471	-1600.93
6	3.443	1.715	73.212	-1509.43
7	4.187	1.86	53.353	-1726.253
8	4.241	1.872	53.325	-1896.638
9	4.236	1.860	48.734	-1999.646

（a）IS 变化图

（b）JS-IS 变化图

（c）FID 变化图　　（d）WD 变化图

图 4-4 无监督 GAN 训练过程中四种评价指标变化图(CIFAR10)

(a)第 1 次记录　(b)第 2 次记录　(c)第 3 次记录　(d)第 9 次记录

图 4-5 无监督 GAN 训练中生成图像(CIFAR10)

从图 4-4 可以看出，JS-IS 的稳定性要高于 IS，而 IS 的灵敏度比 JS-IS 要高。从表 4-2 可以看出，到了训练后期，FID 的变化基本上已经到了 5 左右，而 WD 的变化还可以达到 100 左右，可见 WD 的灵敏度要高于 FID。

(3) 模式坍塌实验

模式坍塌是无监督 GAN 中一个很重要的问题，引起了研究界的广泛关注，一项可靠的 GAN 的评价指标必须具有判别模式坍塌的能力。谷歌公司曾经进行过实验证明 FID 对模式坍塌具有描述能力，具体实验如下[5]：在一个包括 10 类数据的数据集中，分别取十个子集，其中包括了 1 类、2 类、……和全部 10 类数据，计算这 10 个子集的 FID，结果发现在 MNIST、FASHION-MNIST、CIFAR10 数据集上，在只包含 1 类的数据集上，FID 的值很高，随着包含类别的增加，FID 显著降低，到了包含 10 类的数据集上，FID 基本上为 0。这个实验证明了 FID 对类别的缺失是有感知能力的。

但考虑到这个实验是在原数据集上取的图片，并非生成的图片，本文希望能够验证四种评价指标对生成图片的模式坍塌的感知能力，采样的图片应该为生成图片。因此设计了如下实验，在 MNIST 数据集上，调节各种超参数和网络结构，让生成的图片分别包含全部 10 类数字、9 类数字、5 类数字和 2 类数字，将对应的生成模型保存下来。再对这些生成模型输入同一组固定噪声，计算四种评价指标的值，结果如下：

表4-3 无监督GAN模式坍塌实验（MNIST）

模式情况	IS	JS-IS	FID	WD
2 类	1.969	1.336	84.62	-709.33
3 类	1.949	1.330	93.801	-618.290
9 类	1.996	1.347	43.78	-982.727
10 类	2.142	1.401	12.758	-1203.053

按照之前的实验，图片质量越好，IS 和 JS-IS 应该越大，FID 和 WD 应该越小。从表 4-

3 展示的实验结果来看，数据的第一行和第二行出现了相反的情况，但其他数据均表现良好，并且在不出现模式坍塌的情况下，IS 和 JS-IS 都是最大的，而 FID 和 WD 都是最小的。总体上，四种评价指标是可以反应模式坍塌的程度的。

另外，从表 4-3 可以看出，FID 和 WD 对模式坍塌的灵敏度是要优于 IS 和 JS-IS 的，在数值上，WD 对模式坍塌的灵敏度也要高于 FID，如果对生成结果的多样性要求较高，可以侧重于考虑 WD。关于评价指标的侧重点，第四章中会进一步探讨。

综上所述，本小节选取了 DCGAN，在数字数据集和真实图片数据集上分别进行了实验，检验四种评价指标与图片的清晰度和模式坍塌情况的关系。四种评价指标在评价无监督 GAN 的图片清晰度质量和模式坍塌方面有一定的意义，可以作为超参数搜索时的准则来对超参数进行粗调，得到可用的超参数。

IS 散度由于其欠佳的数学性质遭到诟病，而 JS-IS 利用了 JS 散度良好的数学性质：对称性和有界性，稳定性要好于 IS，避免了溢出等情况。但相应的不足是 JS-IS 的取值范围较窄，灵敏度要略逊于 IS。

结合本小节的实验，在无监督 GAN 上，WD 无论是评价生成图片清晰度还是模式坍塌程度，都体现了优于其他三种指标的灵敏度。

4.2.3 有监督 GAN 实验

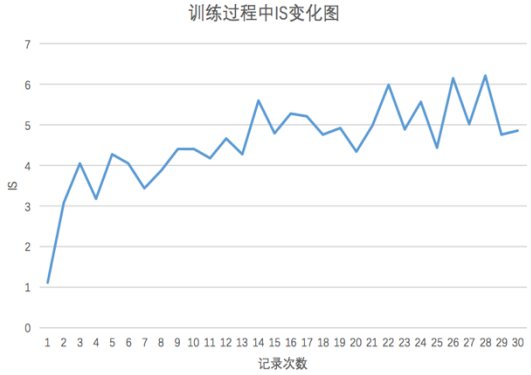
（1）数字数据集

有监督 GAN 实验在 MNIST 数据集上，用 ACGAN 进行，输入的噪声维度为 62，batch_size 为 64，优化器为 Adam，其中学习率为 0.0002，β 为 0.5，即网络结构与无监督 GAN 相似，但在此基础上加入约束条件：类标 y。

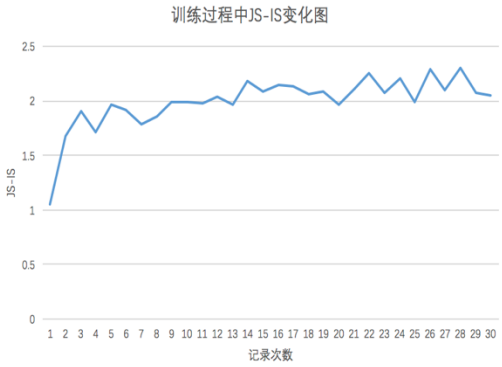
同样地，每20个迭代进行一次数据记录，计算出四种评价指标的值，直到生成的图像趋于稳定。

表4-4 有监督GAN评价指标随训练变化（MNIST）

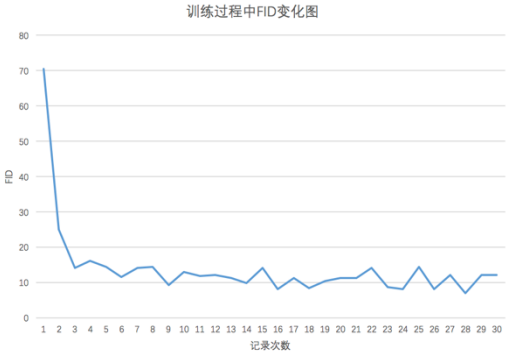
记录次数	IS	JS-IS	FID	WD
1	1.118	1.057	70.539	50.684
4	3.178	1.717	16.017	-581.114
8	3.883	1.860	14.518	-768.528
12	4.674	2.031	12.06	-946.49
16	5.271	2.151	8.057	-1106.978
30	4.355	1.967	11.421	-864.155



（a）IS 变化图



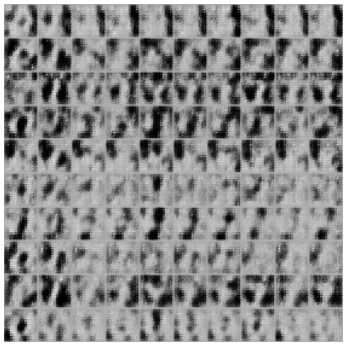
（b）JS-IS 变化图



（c）FID 变化图

（d）WD 变化图

图 4-6 有监督 GAN 训练过程中四种评价指标变化图（MNIST）



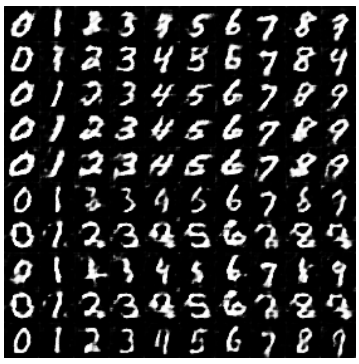
（a）第 1 次记录

（b）第 2 次记录

（c）第 3 次记录

（d）第 15 次记录

（e）第 20 次记录



（f）第 26 次记录

图 4-7 有监督 GAN 训练中生成图像（MNIST）

如表 4-4 和图 4-6 所示，随着训练的进行，IS、JS-IS、FID、WD 的变化与无监督学习中的类似，均为前两者逐渐变大，后两者逐渐变小，最后趋于稳定。同样的，在敏感性方面，JS-IS 要略逊于 IS，而 WD 则要比其他三种指标都灵敏地多。

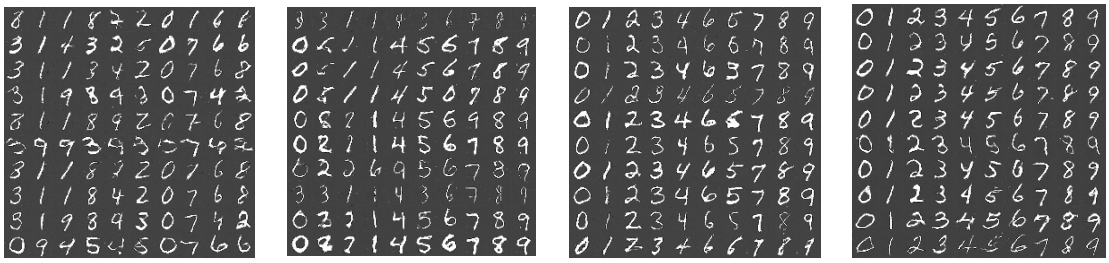
图 4-7 展示了几个节点的图片，可以看到，图片随着训练的进行，从（a）中的噪声逐渐变清晰，到了大约第 15 次记录之后，图片的质量约有波动，总体上，这之后图片的质量和评价指标一样，逐渐趋于稳定。可见，这四种评价指标对评价 ACGAN 生成图片的清晰度都是可用的。

另外，有监督 GAN 中还有一个重要的考虑点：是否能够根据给定的标签生成正确标签的图片。本实验中给定的标签是 0，1，…, 9, 在图 4-7 中，生成的图片标签都是正确的。现在，本实验将修改部分网络结构，但不修改输入输出张量的维度，让生成图片的标签发生错乱，由此观察四种评价指标的变化情况。

在实验中，由于各个修改过的网络是分开训练，为了更好地控制变量，本实验中采取的方法与 3.4.3.1 中模式坍塌实验中的方法相同：分别训练各个修改过的模型，将训练完成后的生成模型保存下来。然后对每一个生成模型输入同一组噪声和标签，在它们生成的图片中进行采样，计算四种评价指标的值，实验结果如表 4-5、图 4-8 所示。

表4-5 有监督GAN评价指标（MNIST）

生成图片	IS	JS-IS	FID	WD
图 4-8 (a)	2.006	1.354	14.809	-1296.831
图 4-8 (b)	2.099	1.377	13.502	-1373.8701
图 4-8 (c)	1.849	1.303	15.081	-1467.5269
图 4-8 (d)	2.168	1.393	11.972	-1384.842



(a) (b) (c) (d)

图 4-8 有监督 GAN 图片示例

在本实验中，正确的图片应该是从左至右依次为 0-9，即图 4-8 中（d）是较好的，(c)在 5 和 6 两列略有错误，(b)的前 5 列有错，而(a)可以算是非常糟糕。

而从表 4-5 来看，首先，四种评价指标的变化方向还是与之前相同：即 IS，JS-IS 上升时，FID 和 WD 下降。但结合图片的标签准确性来看，可以发现，这四种评价指标都有问题。比如：(c)的图片标签准确性要远好于(a)和(b),但(c)的 FID 却比(a)和(b)要高， IS 和 JS-IS 却略低于(b)。(d)的图片标签准确率好于(c)，但(d)的 WD 要高于(c)。

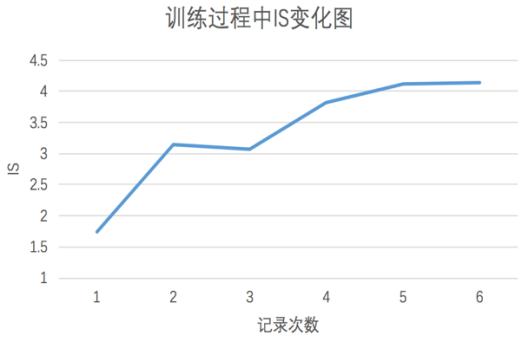
本实验为在数字数据集上进行的一个简单实验，但已经暴露出了这四种评价指标在评价标签准确性方面的不足。结合之前的研究，IS、JS-IS、FID、WD 这四种评价指标在描述生成图片的清晰度方面有一定的意义，但在有监督学习中，它们对生成图片标签准确性的描述是不可靠的，需要别的方法加以衡量。

（2）真实物体图片数据集

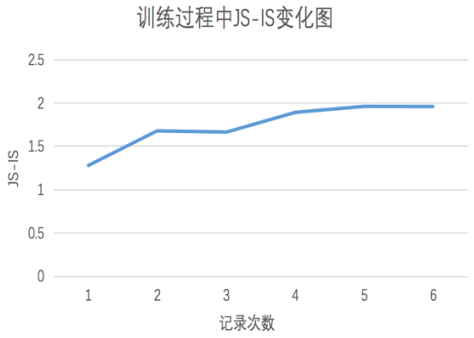
有监督GAN的真实物体图片数据集实验在CIFAR10数据集上，用ACGAN进行。输入的噪声维度为110，batch_size为100，优化器为Adam，其中学习率为0.0002，β_1为0.5，β_2为0.999。每5个epoch进行1次记录，实验结果如表4-6。

表4-6 有监督GAN评价指标随训练变化（CIFAR）

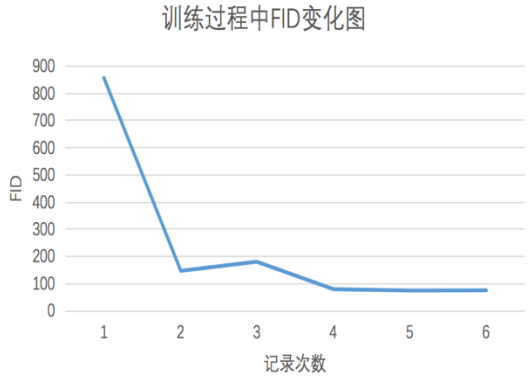
记录次数	IS	JS-IS	FID	WD
1	1.734	1.285	855.105	270.269
2	3.151	1.679	144.836	-1337.653
3	3.075	1.666	178.925	-1378.235
4	3.818	1.899	77.119	-1174.848
5	4.119	1.968	76.188	-1371.218
6	4.152	1.961	73.114	-1335.429



（a）IS 变化图



（b）JS-IS 变化图



（c）FID 变化图

（d）WD 变化图

图 4-9 有监督 GAN 训练过程中四种评价指标变化图（CIFAR10）

(a)第 1 次记录　(b)第 2 次记录　(c)第 4 次记录　(d)第 6 次记录

图 4-10 有监督 GAN 训练中生成图像(CIFAR10)

从表 4-6，图 4-9 可以看出，CIFAR10 数据集上有监督 GAN 训练过程中四种评价指标的变化情况与 MNIST 数据集上的大致相同，总体情况尚可。从图 4-10 也可以看出四种评价指标总体上和图片质量有着大致的对应的关系。

因此，四种评价指标在评价有监督 GAN 在真实数据集上的表现时都有一定的意义，结合真实数据集上无监督 GAN 的表现，如果要选择一种评价指标，推荐 WD。

同样的，在真实图片数据集上，也要进行检查生成图片标签准确性的实验。这部分实验采用的方法和数字数据集上的略有不同：在评估生成图片的标签准确性时，数字数据集可以简单地通过人眼来区分，而在真实图片数据集上，人眼出现判断失误的概率较高，加之 CIFAR 数据集本身也是比较模糊的，直接人工判定标签并不是很准确。因此，本实验在判别标签时使用了一个在 CIFAR 数据集上进行过预训练的神经网络 ResNet20，ResNet 在分类准确率方面是非常优秀的，本实验中采用 ResNet 进行分类以达到准确评价生成图片标签准确性的目的。实验结果如表 4-7。

表4-7 有监督GAN评价指标与分类准确率关系（CIFAR10）

准确率	IS	JS-IS	FID	WD
49.69	2.819	1.615	171.498	-1351.479
50.43	3.08	1.713	62.97	-1041.43
34.06	3.038	1.629	74.476	-1258.674

从表 4-7 可以看出，第一行和第二行体现出在准确率相近时，FID 和 WD 可以相差许多，而第二行和第三行中，准确率下降时，FID 和 WD 出现了下降的情况。与 MNIST 数据集上的实验结果一样，对于有监督 GAN，在 CIFAR 数据集上，四种评价指标对生成图像的标签准确率也都没有明显的描述能力。

4.2.4 有监督 GAN 评价指标修正

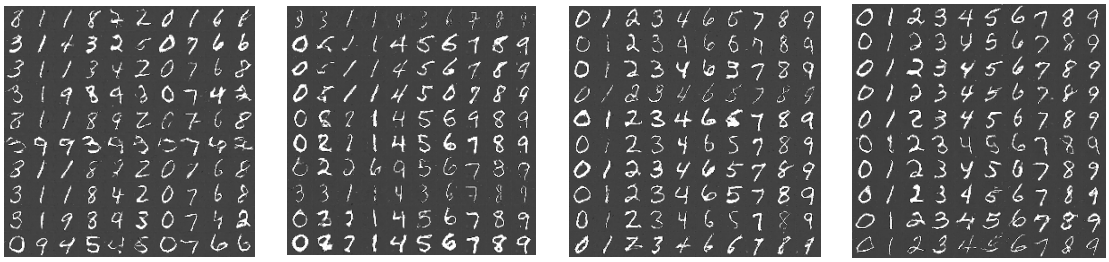
如 4.2.3 中所证明的，已有的四种 GAN 评价指标 IS、JS-IS、FID 和 WD 对有监督 GAN 生成的标签准确率都没有明显的描述能力。本文提出需要对有监督 GAN 的评价指标进行修正：将标签准确率作为评价指标之一。

在标签准确率的计算上，本文提出以下方法：对于比较简单的数据集，可以用真实数据集训练一个深度卷积分类网络，用训练得到的模型作为分类网络，生成图片作为测试集进行测试，测试的准确率即为标签准确率。

由于 CIFAR 数据集比较模糊，通过人眼不容易准确判断图片所属类别。这里采用 MNIST 数据集进行实验。实验的场景同前。实验结果如表 4-8。

表4-8 有监督GAN标签准确率

生成图片	测得准确率
图 4-11(a)	40.81
图 4-11(b)	58.68
图 4-11(c)	83.29
图 4-11(d)	92.34



(a) (b) (c) (d)

图 4-11 标签准确率实验图片

图 4-11 与图 4-8 相同，现在将它用作分类网络的测试集进行标签准确率的计算。从图 4-11 中可以看出，图片对应标签准确率(a) < (b) < (c) < (d)_。从表 4-8 可以看出，分类网络测得的准确率很好地描述了生成图片的标签准确率。

对于比较复杂的数据集，用真实数据集从头开始训练分类器会耗费比较多的时间。作者建议可以利用在数据集 ImageNet 上进行过预训练的 Resnet 网络，这样的预训练模型可以非常方便地在网络上下载到。ImageNet 庞大的数据基础让 ResNet 在预训练阶段就具有很好的分类能力；再用真实数据集对网络进行二次训练（fine-tune），让模型能够适应当前数据集的分类任务，得到二次训练后的模型作为分类模型对生成图片进行测试，这样可以在保证准确率的同时，节省资源和时间的开销。

4.2.5 评价指标实验结论

（1）清晰度

在黑白数字数据集和彩色真实物体数据集两类数据集上进行了实验，横向比较四种评价指标评价生成图片清晰度的能力。实验结果表明，IS、JS-IS、FID 和 WD 这四种评价指标对评价 GAN 生成图片的清晰度都有一定的意义：IS 和 JS-IS 越大，FID 和 WD 越小，对应的生成图片越清晰。其中 JS-IS 的稳定性高于 IS，可以应对较为极端的情况，但灵敏度不如 IS。WD 的灵敏度高于其他三种指标。

（2）模式坍塌程度

通过设计实验构造不同程度的模式坍塌，比较四种评价指标的变化情况，得出 FID 和 WD 评价无监督 GAN 模式坍塌的表现要好于 IS 和 JS-IS，应该使用 FID 和 WD 来衡量模式坍塌程度，相比较而言，WD 在评价模式坍塌程度时的灵敏度要好于 FID。

（3）标签准确性

通过设计实验，使有监督 GAN 出现不同程度的标签准确性错误，比较四种评价指标，得出四种评价指标对标签准确性都没有描述能力。

进而，本文提出将生成图片数据集作为测试集，用分类网络来对测试集进行分类，将得到的分类准确率作为生成图片的标签准确率，证明了这种评价方法的可靠性。

4.3 超参数优化方法实验

本节对传统的搜索算法：网格搜索和随机搜索以及本文新提出的 exploit & explore 算法进行了实现，将三种超参数优化方法在所选的超参数分布、有效搜索次数和准确率方面分别

进行比较，最后研究了评价指标对超参数优化方法的影响。

考虑到 GAN 超参数优化的特点，大部分超参数都可以取一个离散的列表，本章中涉及到的超参数优化均只考虑离散情况，即已知每个超参数各自可以取的值，在它们构成的超参数空间中搜索最优的超参数组合。

由于深度学习的训练时间较长，第三章中已经证明各个评价指标在不同数据集上的有效性。本章中主要研究的各个超参数搜索方法在搜索 GAN 的超参数时的表现，所以实验多在 MNIST 和 Fashion-MNIST 这两个黑白数据集上开展，既覆盖了数字数据集和真实物体数据集，又减少了资源开销。

4.3.1 网格搜索实现与结果

4.3.1.1 算法实现

假设有 n 个超参数需要优化，它们各自可能的取值构成了列表 $list_1$, $list_2$,…, $list_n$，由一个评价指标 criterior 来衡量在某一个超参数组合下 GAN 生成数据的质量。网格搜索通过遍历所有的超参数组合，来找到其中能够使 criterior 达到最优的超参数组合，返回此时 criterior 的取值和对应的最优的超参数组合。其中如果评价指标 criterior 为 IS 或 JS-IS，则 criterior 越大越好，如果评价指标 criterior 为 FID 或 WD，则 criterior 越小越好。

网格搜索的具体伪代码实现如下：

网格搜索算法

输入：超参数列表 $list_1$, $list_2$,…, $list_n$,

评价指标：criterior，记 criterior($param_1$, $param_2$,…, $param_n$)表示在超参数为 $param_1$，$param_2$，…, $param_n$ 时评价指标的值

输出：最优的超参数组合

此时 criterior 的取值

```
param_len_k = len(list_k)   k=1, 2, …, n
best_index = null
for i = 0, 1, …, param_len_1 – 1:
    for j = 0, 1, …, param_len_2 – 1:
        ……
        for m = 0, 1,…, param_len_n -1:
            current_criterior = criterior(list_1[i], list_2[j],…,list_n[m])
                if i,j,…,m = 0:
                    best_criterior = current_criterior
                    best_index =i,j,…,m
                else:
                if criterior is IS or JS-IS:
                    if current_criterior >best_criterior:
```

```
                    best_index =i,j,…,m
                    best_criterior = current_criterior
            else:
                if current_criterior<best_criterior:
                    best_index = i,j,…,m
                    best_criterior = current_criterior
return best_criterior, best_index
```

算法的流程图如图 4-12 所示。

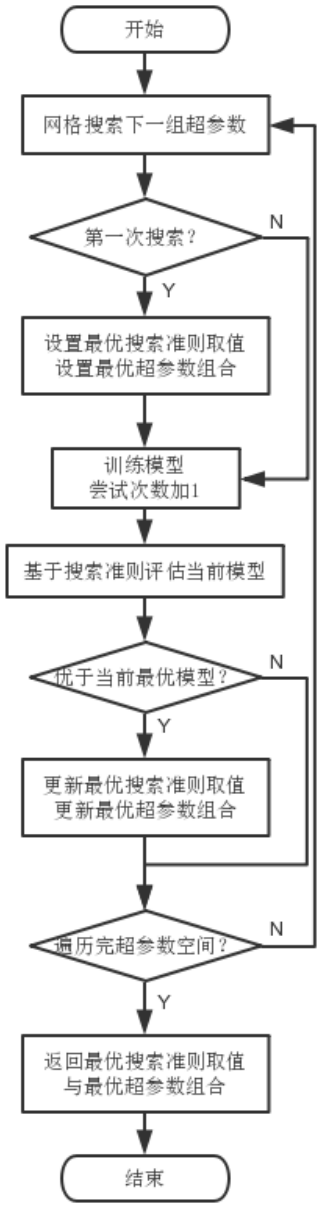


图 4-12 网格搜索算法流程图

本实验中，先选取 MNIST 和 Fashion-MNIST 作为数据集，考虑到本实验是为了研究超参数搜索的性能，选取的生成式对抗网络需要满足的条件是：在不同的超参数下训练效果有比较大的不同，在超参数设置得当的情况下能够展现出比较好的训练效果。因此，作者选取了 DCGAN 作为实验所用的生成式对抗网络，优化器为 Adam，输入和输出图片规格均为 28 * 28 * 1，DCGAN 的训练过程一般是在一个 batch 上，先让 D 优化几次，再让 G 优化一次，具体的伪代码如下：

DCGAN 训练过程:epoch：在整个数据集上遍历次数
batch_size：batch 的大小
d_iter：在一个 batch 上 D 学习的次数

```
Begin:
    for i = 1,2 …, epoch:
        for batch in len(dataset)/batch_size:
            for j=1,2,…,d_iter:
                compute D Loss
                update D network
            compute G Loss
            update G network
```

4.3.1.2 结果展示与分析

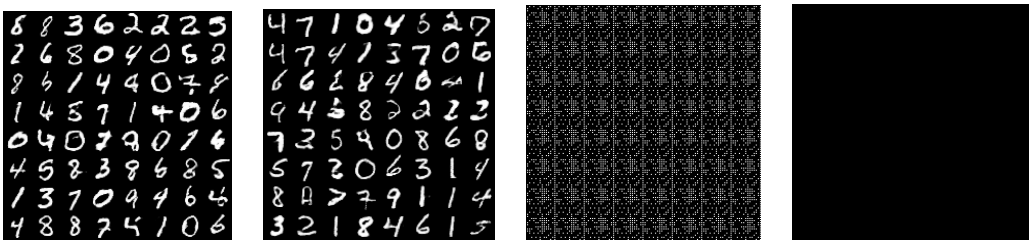
本实验中，将优化器的学习率和 D 的学习次数 d_iter 作为两个待优化的超参数，在二维超参数空间中寻找最优的超参数组合。先猜测可能的取值列表分别为[0.000002, 0.00002, 0.0002, 0.002, 0.02, 0.2], [1, 4, 7, 10, 13]，dataset 的长度为 60000，batch_size 取 64（最后一个 batch 为 32），epoch 取 3，共进行了 2814 次迭代，以四种评价指标分别进行了网格搜索的实验，得到的结果如表 4-9。

表 4-9 网格搜索实验结果（DCGAN MNIST）

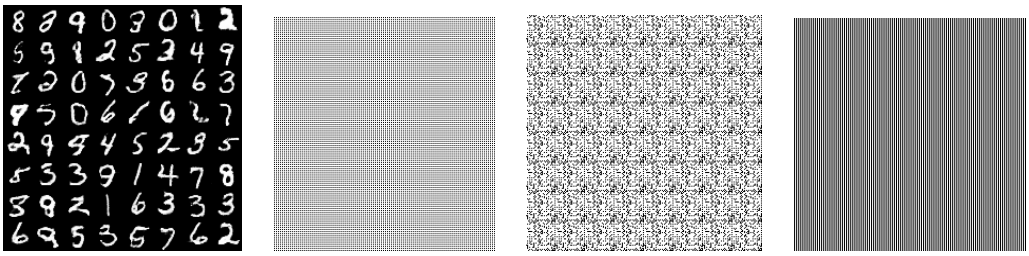
评价指标	所选学习率	所选 D 学习次数	搜索用时(s)
IS	0.0002	13	23095.68
JS-IS	0.0002	1	21772.83
FID	0.0002	7	19291.72
WD	0.0002	4	22989.65

从表 4-9 可以看出，采用四种评价指标进行网格搜索的时间开销相差不大，相比较而言 FID 较快，四种评价指标一致选择了 0.0002 作为学习率，在 D 学习次数上则略有不同，接下来分析网格搜索的优化效果。

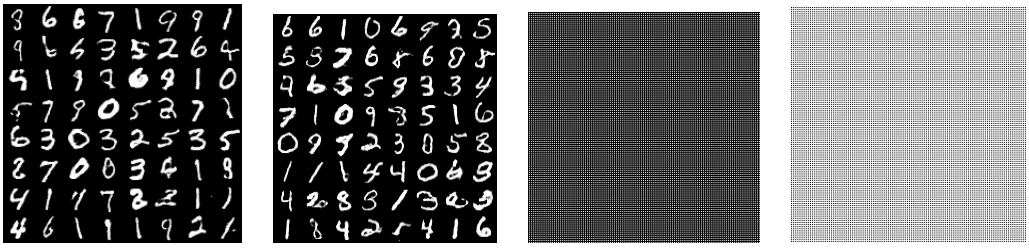
表 4-9 中展示的所选取的最优超参数对应的生成图片以及其他超参数生成的图片如图 4-1 所示，每一行最左边为最优的超参数生成的图片，介于四种评价指标均选择了 0.0002 作为最优的学习率，推断学习率是本任务中比较重要的超参数，故每一行的后三列展示在 D 学习次数相同的情况下，学习率分别取 0.002、0.02、0.2 时的生成图片。



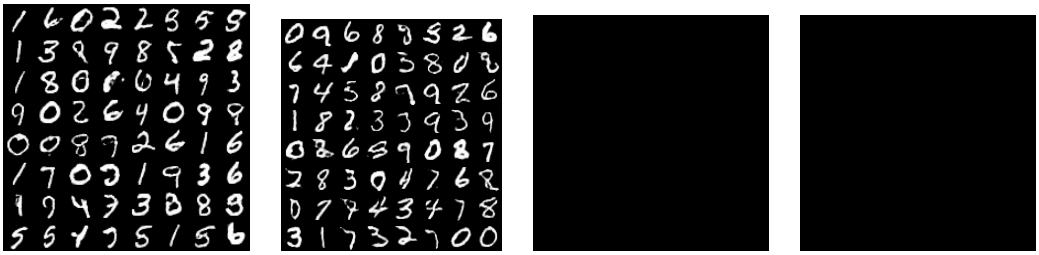
(a) 以 IS 为指标的网格搜索生成图片



(b) 以 JS-IS 为指标的网格搜索生成图片



(c) 以 FID 为指标的网格搜索生成图片



(d) 以 WD 为指标的网格搜索生成图片

图 4-13 网格搜索生成图片（DCGAN，MNIST）

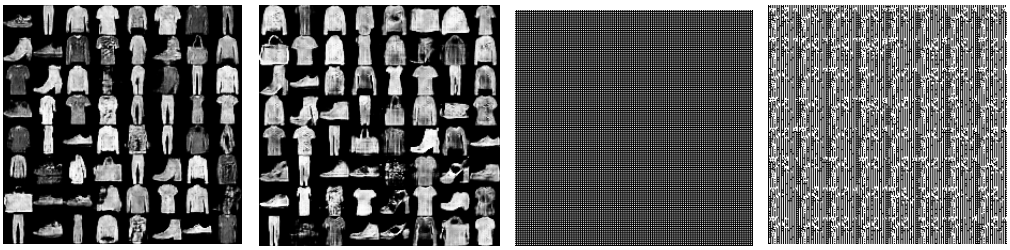
从图 4-13 可以看出，每一行最左边的图片，即该种评价指标下选取的最优超参数组合产生的图片质量都是合格的，没有出现右边三列中有时会出现的一团噪声或者一片黑色的情况，此外，最左边图片的质量可以说与右边图片相比，都是伯仲之间或者优于右边的。而分析第一列图片，即四种指标下选取的不同的超参数组合产生的图片，发现很难分辨图片的孰优孰劣，可见四种指标选出的超参数都是比较合理的，当然这体现出 MNIST 数据集相对较为简单，DCGAN 比较容易生成质量较好的图片。关于四种指标在选取超参数方面的侧重将在 4.5 中详细讨论。

接下来，同样的，在 Fashion-MNIST 数据集上进行了网格搜索实验，网络的结构和超参数等与 MNIST 实验中的相同，实验结果如表 4-10。

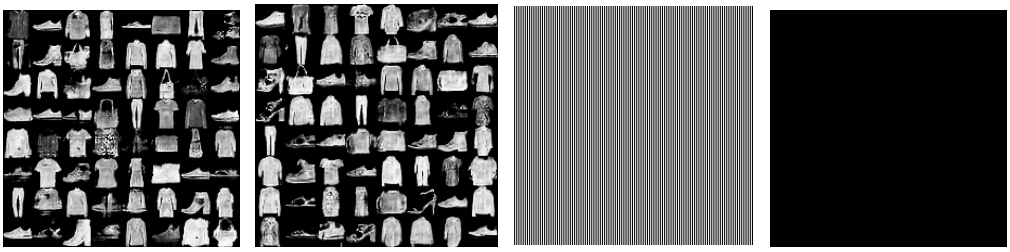
表 4-10 网格搜索实验结果（DCGAN Fashion-MNIST）

评价指标	所选学习率	所选 D 学习次数	搜索用时(s)
IS	0.002	1	15554.59
JS-IS	0.002	4	14744.61
FID	0.0002	10	13758.67
WD	0.002	13	14513.20

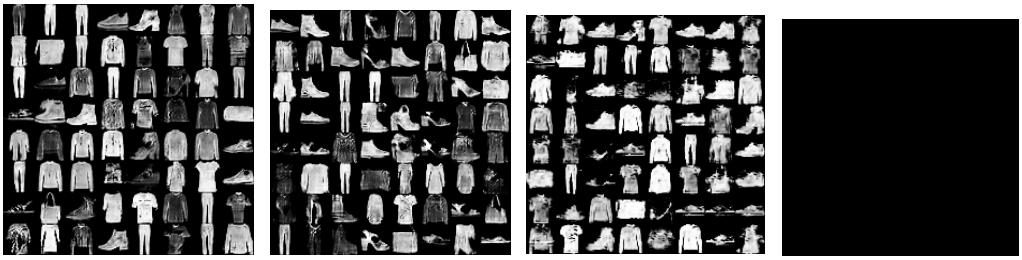
在这次实验中，四种指标选择的学习次数均不相同，并且 FID 选择的学习率与其他三种指标不同，但学习率仍然是两个超参数中比较重要的一个。图 4-13 展示了各个超参数组合对应的生成图片，从上至下分别为 D 学习 1，4，10，13 次，从左至右学习率分别为 0.0002，0.002，0.02，0.2。



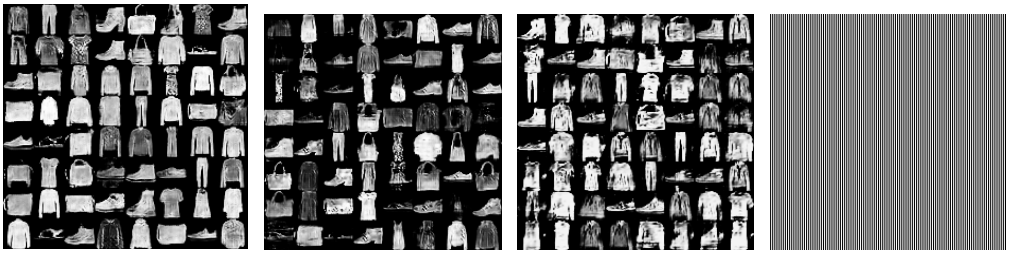
(a) 以 IS 为指标的网格搜索生成图片



(b) 以 JS-IS 为指标的网格搜索生成图片



(c) 以 FID 为指标的网格搜索生成图片



(d) 以 WD 为指标的网格搜索生成图片

图 4-13 网格搜索生成图片（DCGAN，MNIST）

如图 4-13 所示，FID 选择的超参数对应的是左起第二张图片，其他的 3 种指标选择的超参数对应的都是最左边的图片。首先，可以看出，四种指标选择的超参数对应的生成图片的质量都是合格的，避开了一片黑暗或者一团噪声的情况。在图片质量方面，也可以看出网格搜索得到的超参数对应的图片质量还是要优于其他一些参数的，比如第三行第二列为基于 FID 选择的最优超参数对应的图片，它的质量明显好于第三列的图片，第四行第一列是基于 WD 选择的最优超参数对应的图片，它的质量也是要好于第二列和第三列的。可见，网格搜索得到的图片质量还是比较好的。

综上所述，以第三章中的四种 GAN 评价指标作为搜索准则，用网格搜索穷举离散超参

数空间中的所有超参数来寻找最优的超参数，在效果上是不错的，如果有足够的资源，可以进行并行分布式计算，那么所用的时间会大大减少。但在资源不足，如本文中只用一块 GPU 显卡的情况下，网格搜索的效率较低，花费的时间较长。

4.3.2 随机搜索实现与结果

4.3.2.1 算法实现

随机搜索可以分为在连续、离散的超参数空间中进行随机搜索两种。在本文中，由于已知各个超参数的待选超参数列表，故研究离散超参数空间中的随机搜索。

随机搜索的算法实现也较为简单，即先限定尝试次数，每次随机选取离散超参数空间中的一个点作为超参数，训练网络后根据评价指标评估网络的优劣，最后返回最优的超参数组合，具体的伪代码如下。

随机搜索：

输入：超参数列表：$list_1, list_2,\ldots, list_n$,

评价指标 criterior：记 $criterior(param_1, param_2,\ldots, param_n)$为在超参数为 $param_1$，$param_2$，…, $param_n$ 时评价指标的值

try_times：指定的总的尝试次数

输出：最优的超参数组合

此时 criterior 的取值

```
param_len_k = len(list_k)k=1, 2, …, n
best_index = null
for try_time = 0, 1,…, try_times:
i_k = rand(0,param_len_k)
current_criterior = criterior(list_1[i_1], list_2[i_2],…,list_n[i_n])
        if i_k = 0:
                best_criterior = current_criterior
                best_index = i_k
        else:
                if criterior is IS or JS-IS:
                        if current_criterior >best_criterior:
                                best_index = i_k
                                best_criterior = current_criterior
                else:
                        if current_criterior <best_criterior:
                                best_index = i_k
                                best_criterior = current_criterior
    return best_criterior, best_index
```

算法的流程图如图 4-14 所示。

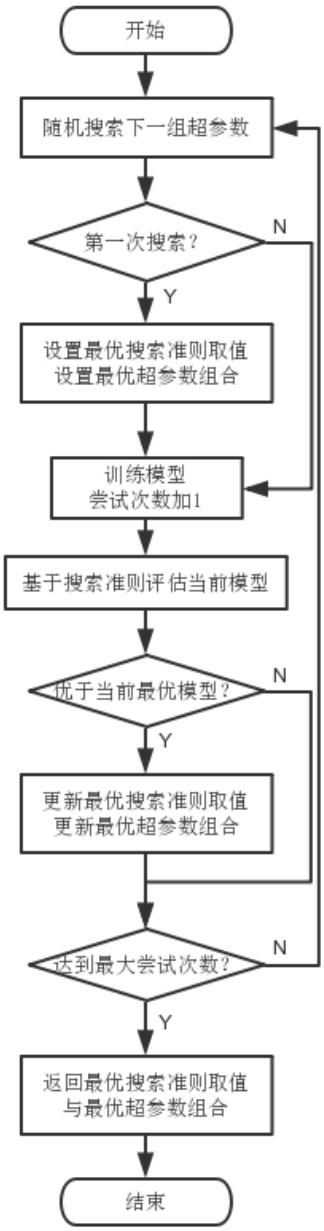


图 4-14 随机搜索算法流程图

本实验中，所选取的网络结构和其他超参数均与 4.3.1 中相同，并且同样取学习率和 D 的学习次数 d_iter 作为两个待优化的超参数，在二维超参数空间中寻找最优的超参数组合。先猜测可能的取值列表分别为[0.000002, 0.00002, 0.0002, 0.002, 0.02, 0.2], [1, 4, 7, 10, 13]。

4.3.2.2 结果展示与分析

本实验中的目的是比较随机搜索和网格搜索，因此选定了相对而言计算速度较快的 FID 为搜索准则，设定搜索次数为 10 次，即在 30 种超参数组合中随机地选取 10 种进行训

练，返回这 10 种超参数组合中最优的一种作为超参数优化的结果。旨在研究随机搜索在限定了搜索次数的情况下能够达到的训练效果，并与网格搜索进行效果和效率的比较。

表 4-11 随机搜索实验结果

尝试次数	本次尝试超参数	最优超参数	最优 FID
2	0.002,10	0.002,10	8.360
4	0.00002,1	0.002,10	8.360
6	0.2,1	0.002,10	8.360
8	0.02,10	0.002,10	8.360
9	0.0002,7	0.0002, 7	7.193
10	0.00002,13	0.0002, 7	7.193

图 4-15 分别展示了 FID 为 8.360 和 7.193 时的生成结果。

(a) FID = 8.360　　(b) FID = 7.193

图 4-15 随机搜索生成图片

随机搜索的实验用时 6113.03 秒，约为网格搜索的 1/3，这和搜索的次数是成比例的。从表 4-11 可以看出，随机搜索在尝试了 2 次之后，就将找到了一次尚可的超参数组合，并且在尝试了 9 次的时候，找到了网格搜索中找到的最优的超参数。从图 4-15 可以看出，尽管 FID 有将近 1.2 的差距，但两张图片的质量都在可接受的范围内，可见随机搜索早期选出的参数也很有可能是可用的。另外，值得注意的是，本次实验中，同样选取 0.0002，7 的超参数组合对 GAN 进行训练，得到的 FID 值与之前网格搜索实验中的不同，可见由于 GAN 的不稳定性，相同的超参数下训练的网络的评价指标值也不稳定，因此，实际操作中也可考虑多次训练同一个网络，对评价指标取平均值的方法来缓解这种不稳定性所带来的影响。

从随机搜索的实验结果来看，虽然超参数空间中最优的超参数组合只有一个，但仍有其他的一些点可以达到不错的训练效果，这些点也是可用的，随机搜索能够比较快的找到这些点，并且由于随机性，随机搜索也可能用比较少的搜索次数就率先搜索到了最优的点。可见，随机搜索相比网格搜索，在可能牺牲部分生成图片质量的情况下，大大减少了需要的时间和资源开销，并且也保留了网格搜索的并行性，可以在分布式环境中进行使用。在实际实验中，只要尝试次数设置得当，随机搜索基本都能得到可用的超参数，也可用于超参数的粗调，即根据随机搜索的结果确定最优的超参数大概所处的范围，然后再在此基础上进行手工细调，节约了资源，也提升了超参数优化的效率。

4.3.3 exploit & explore 算法实现与结果

4.3.3.1 算法实现

与网格搜索和随机搜索一样，exploit & explore 算法的应用场景依旧是在已知各个超参数各自可能的取值列表后，在离散的超参数空间中对超参数进行搜索。具体的伪代码如下。

exploit & explore 算法：

输入：超参数列表：$list_1, list_2,\ldots, list_n$,

评价指标 criterior：记 $criterior(param_1, param_2,\ldots, param_n)$为在超参数为 $param_1$，$param_2$，…, $param_n$ 时评价指标的值

try_times：指定的总的尝试次数

输出：最优的超参数组合

此时 criterior 的取值

```
param_len_k = len(list_k)k=1, 2, …, n
best_param = null
try_time = 0
while try_time < try_times:
        i_k = rand(0,param_len_k)
        current_criterior = criterior(list_1[i_1], list_2[i_2],…,list_n[i_k])
        try_time ++
        if i_k = 0:
            best_criterior = current_criterior
            best_param = i_k
        else:
        if criterior is IS or JS-IS:
                if current_criterior >best_criterior:
                    // exploit
                     best_param = i_k
                     best_criterior = current_criterior
                    // explore
                      added_param_k = disturb(best_param)
                     current_criterior = criterior(added_param_k)
                     try_time++
                     if current_criterior is better than best_criterior:
                           update best_criterior and best_param
                      list_k.append(added_param_k)
                     param_len_k = param_len_k + len(added_param_k)
           else:
                if current_criterior <best_criterior:
                    // exploit
                     best_param = i_k
                     best_criterior = current_criterior
```

```
                    // explore
                      added_param_k = disturb(best_param)
                      current_criterior = criterior(added_param_k)
                      try_time++
                      if current_criterior is better than best_criterior:
                            update best_criterior and best_param
                      list_k.append(added_param_k)
                      param_len_k = param_len_k + len(added_param_k)
    return best_criterior, best_index
```

4.3.3.2 结果展示与分析

本实验中，所选取的网络结构和其他超参数均与 4.3.1，4.3.2 中相同，两个待优化的超参数同样为学习率和 D 的学习次数 d_iter，超参数可能的取值列表分别为[0.00002, 0.0002, 0.002, 0.02, 0.2], [1, 4, 7, 10, 13]。

在本次实验中，对学习率的扰动为乘以 0.8 和 1.2，对学习次数的扰动是随机加一和减一，这样，得到了 2 个扰动过的学习率和 2 个扰动过的学习次数，在它们当中各自随机取一个学习率和学习次数作为新的超参数对网络进行训练。

本次实验一样进行 10 次搜索，用 exploit & explore 算法搜索不同的 10 组超参数，训练模型，试图找出最优的超参数。

并且在其他条件都相同的情况下，对同一个网络在同一个超参数空间中用随机搜索和网格搜索各做一次超参数选择。

表 4-12 exploit & explore 实验结果

尝试次数	尝试的学习率	尝试的 d_iter	最优 FID
1	0.00002	7	96.902
2	0.002	10	7.193
3	0.0024	1	7.193
4	0.02	9	7.193
5	0.0016	11	7.193
6	0.0002	7	7.193
7	0.00128	11	7.193
8	0.2	4	7.193
9	0.0016	10	7.193
10	0.002	9	6.860

如表 4-13 所示，随机搜索得到的最优的超参数组合是[0.002,1], 对应的 FID 值为 7.192，此外，网格搜索得到的[0.0002, 7],对应的 FID 值为 7.145。

在时间开销方面，随机搜索用时 1755.64s，exploit & explore 算法用时 2035.53s，网格搜索用时 4676.79s，可以看出，在本实验中，随机搜索和 exploit & explore 算法的时间开销相当接近，而网格搜索要要明显高于前两者，这与理论是相符的。

表 4-13 随机搜索实验结果

尝试次数	尝试的学习率	尝试的 d_iter	最优 FID
1	0.002	10	8.360
2	0.002	1	7.192
3	0.002	1	7.192
4	0.2	13	7.192
5	0.2	4	7.192
6	0.0002	1	7.192
7	0.00002	10	7.192
8	0.02	4	7.192
9	0.02	4	7.192
10	0.002	1	7.192

在超参数优化的效果方面，介于三种算法得到的最优的超参数组合对应的 FID 都较低，三种搜索算法都可以搜索到可用的超参数。另外，从 exploit & explore 算法选择的超参数可以看出，它会向超参数空间中添加新的可能的超参数，表 4-12 中 0.0024、0.0016 等都是在 explore 过程中新找到的超参数。最后 exploit & explore 算法搜索到的超参数组合也有一维是来自于 explore 过程中添加的超参数，在这次实验中，exploit & explore 算法搜索到的最优超参数的 FID 也是低于了网格搜索所得到的 FID，可见本实验中 exploit & explore 算法在有限的时间内的搜索结果还是十分成功的。

4.4 超参数优化方法比较实验

4.4.1 选择的超参数的分布比较

本小节研究搜索算法选择的超参数的分布情况，由于网格搜索选择的超参数分布是固定的，本小节主要研究随机搜索和 exploit & explore 算法选择的超参数的分布情况。

本文选择了二维空间和多维空间分别进行实验，其中多维空间选择了三维空间进行实验，保证了多维空间的性质，降低了实验的开销。在实验方法方面，本实验选择了一些开源的 GAN 模型，其中的超参数基本已经由前人调至最优。在得到了最优的超参数后，本实验有对其他的一些超参数进行了尝试，得到了训练结果尚可时超参数所在的范围，发现这些点往往聚集在一块或者几块区域内。在调整过坐标系的单位后，二维空间中这些超参数组成的区域接近一个或几个椭圆，而在三维空间中则接近一个或几个球。

因此，本文设计了如下的实验，在二维和三维的超参数空间中分别模拟随机搜索和 exploit & explore 算法选择超参数。先选择了一个已经优化到最优的 BEGAN 模型，它的最优的超参数组合为[D 学习次数，学习率，γ] = [4, 0.0001, 0.5],可以对学习率取 10 为底的对数，再将 γ 乘 10 进行坐标调整，在经过坐标调整之后，该模型表现尚可的超参数集合恰好可以看成一个二维平面中的圆或者三维空间中的球，比较容易模拟和观察超参数的选择分布，所以选择该模型作为实验模型。具体的待选超参数，最优超参数范围见表 4-14。

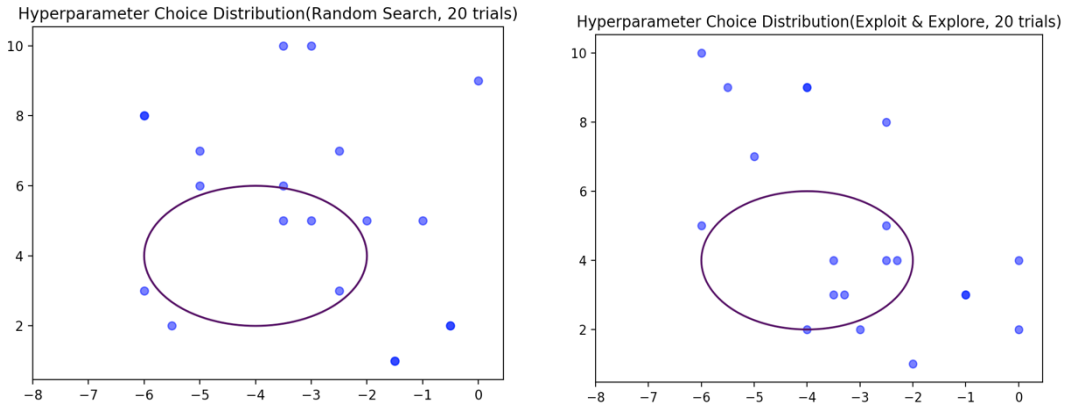
表 4-14 BEGAN 超参数

超参数	待选超参数范围	最优超参数	超参数空间坐标
D 学习次数	[1,10]	4	[1, 2, …,10]
学习率	$[1*10^{-6},1*10^{-0}]$	$1 * 10^{-4}$	[-6, -5.5, …,0]
γ	[0.1,1]	0.5	[1, 2,…,10]

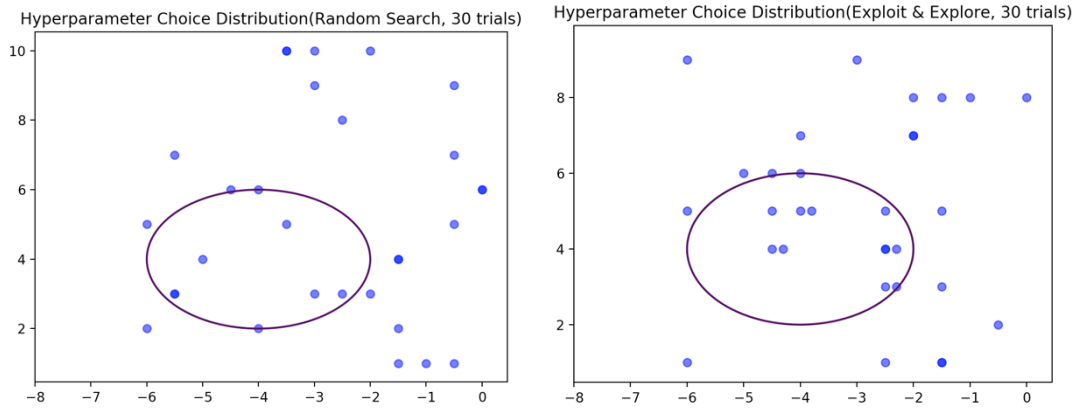
现在分别在二维和三维的空间中模拟随机搜索的 exploit & explore 算法选择的超参数分布。本实验中，将 D 学习次数每隔 1 取一点，学习率对 10 取对数后每隔 0.5 取一点，γ 乘以 10 后每隔 1 取一点，构成待选的的超参数空间。

如表 4-14 的最后一列，经过数学变换之后，取 D 学习次数和学习率构成待选的二维超参数平面，所有三个超参数构成待选的三维超参数空间。下面实验模拟随机搜索和 exploit & explore 算法。

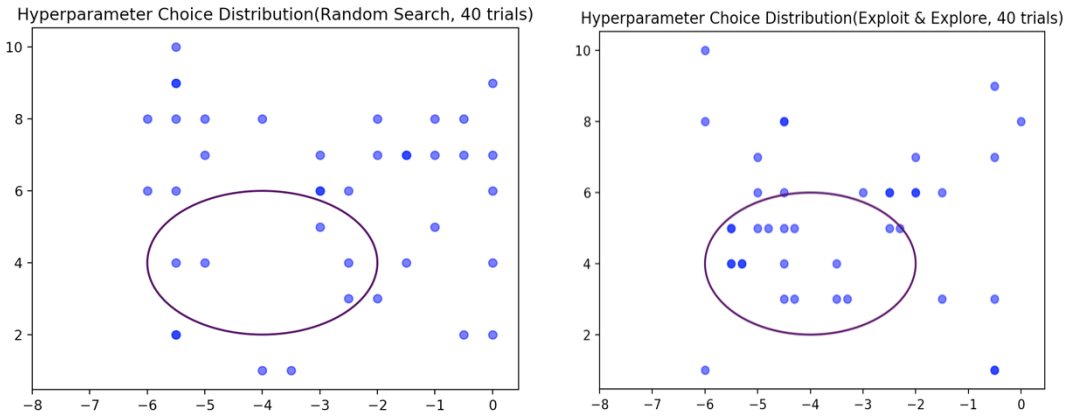
本小节在二维空间中进行实验，方便观察超参数的分布情况。本实验中超参数组合的总数为 130 种，下面分别展示在限定搜索次数为 20、30、40 的情况下，随机搜索和 exploit & explore 算法选择的超参数的分布情况，实验结果如图 4-16。



(a) 搜索 20 次



(b) 搜索 30 次



(c) 搜索 40 次

图 4-16 二维空间超参数搜索分布

如图 4-16 所示，在二维的超参数平面中，圆内为可用的超参数。注意横轴的待选超参数是从-6 开始取值的，图中让横轴包括了-8 是为了让椭圆能够接近图片中心，方便观察。可以看出，相比而言，随机搜索的搜索结果较为均匀地分布在整个超参数平面中，而 exploit & explore 算法更加倾向于聚集在圆圈内。

图 4-16 中显示三次搜索中，exploit & explore 算法选择的超参数中位于圆内，即为可用超参数的次数都要多于随机搜索，在量化指标上证明了 exploit & explore 算法选择的超参数更加容易聚集到可用超参数范围内。

4.4.2 有效搜索次数比较

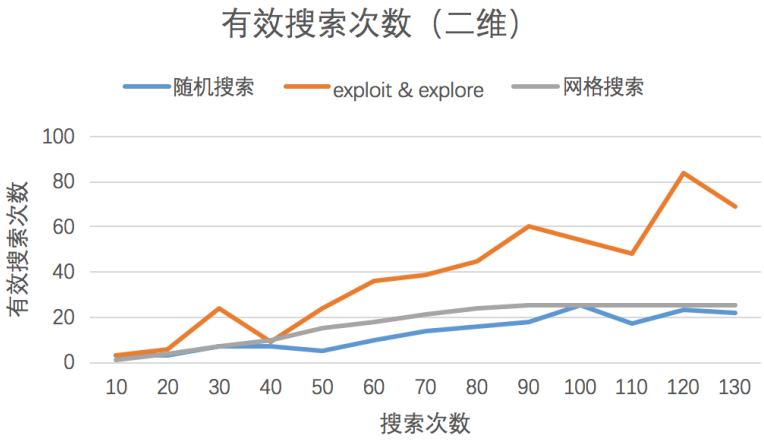
本小节主要比较网格搜索，随机搜索和 exploit & explore 算法在尝试次数相同的情况下的有效搜索次数。分别在二维和三维超参数空间中模拟三种搜索算法，本实验中二维空间为 D 学习次数和学习率，为 10 * 13 ；三维空间为 D 学习次数，学习率和 γ，为 10 * 13 * 10 。分别取不同的搜索次数为进行实验，结果如表 4-15、表 4-16、图 4-17。

表 4-15 搜索算法比较(二维空间)

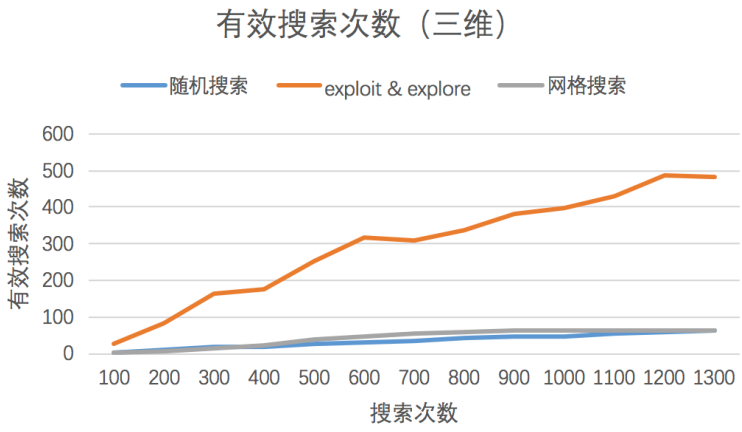
搜索次数	随机搜索有效次数	exploit & explore 有效次数	网格搜索有效次数
10	3	3	1
20	3	6	4
30	7	24	7
40	7	9	10
50	5	24	15
60	10	36	18
70	14	39	21
80	16	45	24
90	18	60	25
100	25	54	25
110	17	48	25
120	23	84	25
130	22	69	25

表 4-16　搜索算法比较（三维空间)

搜索次数	随机搜索有效次数	exploit & explore 有效次数	网格搜索有效次数
10	4	28	1
20	10	84	6
30	17	164	15
40	18	176	24
50	25	252	37
60	32	316	46
70	35	308	55
80	43	336	60
90	46	380	61
100	45	396	61
110	53	428	61
120	50	488	61
130	61	484	61



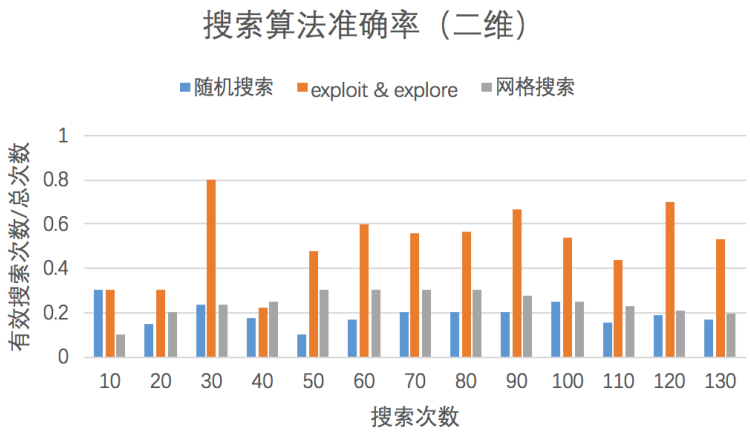
（a）二维超参数空间有效搜索次数图



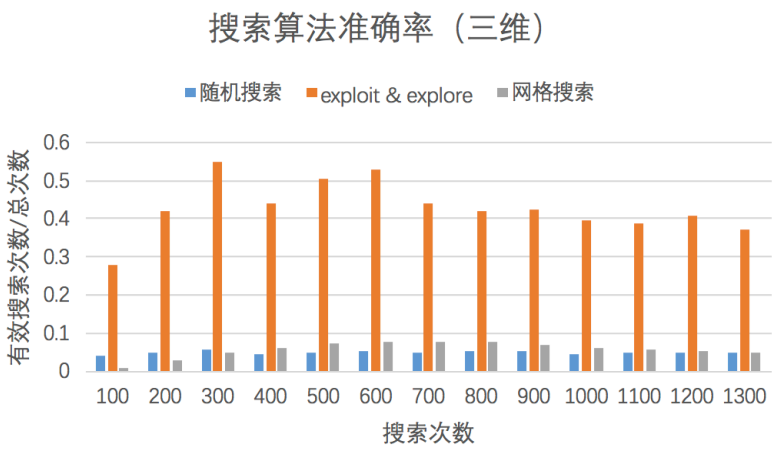
（b）三维超参数空间有效搜索次数图

图 4-17 有效搜索次数图

从图 4-17 可以看出，exploit & explore 算法的有效搜索次数远远高于网格搜索和随机搜索。这在理论上也是很好解释的：在离散的超参数空间中，可用的超参数组合的数量是有限的，所以网格搜索和随机搜索的有效搜索次数有封顶值，即为超参数空间中所有已有的可用的超参数。但 exploit & explore 算法在搜索的过程中，不断向超参数空间中添加可能可用的超参数，并且根据它类似于遗传算法的思想，它所添加的超参数都是已经搜索到的可用的超参数的变异体，这些超参数可以提高超参数空间中可用超参数的比例。下面计算有效次数占搜索总次数的比例，结果如图 4-18 所示。



（a）二维超参数空间搜索算法准确率图



（b）三维超参数空间搜索算法准确率图

图 4-18 搜索算法准确率图

从图 4-18 可以看出，exploit & explore 算法的准确率也要远远高于网格搜索和随机搜索。

4.4.3 结果总结与分析

本小节首先说明一个超参数空间中可用的超参数往往聚集在一块区域中，设计一个实验，取一个 BEGAN，通过经验和人工调节找到其可用超参数的分布，对其各个维度超参

数做坐标变化，使可用的超参数聚集的区域近似一个二维空间的圆和三维空间的球。这里的圆和球均是为了方便进行实验和展示实验结果，实际上，只要可用的超参数聚集在一块区域中或者甚至几块区域内，本小节的实验就具有普适意义。

本小节模拟了网格搜索、随机搜索，exploit & explore 算法的超参数搜索过程，分别在二维和三维超参数空间中进行了实验，计算了在限定超参数搜索次数的情况下，三种搜索算法的有效搜索次数和搜索准确率，发现 exploit & explore 算法要远远高于网格搜索和随机搜索，这主要归功于 exploit & explore 算法在搜索超参数的过程中将一些优质的超参数添加到超参数空间中去，不断地优化了超参数空间。

综上所述，exploit & explore 算法对提高超参数搜索的效率和准确率有着显著的意义。

4.5 评价指标对超参数优化的影响

本小节主要研究评价指标的不同对同一网络的超参数选择的影响。鉴于 GAN 的评价指标所需要描述的是生成图片的清晰度和模式坍塌的程度即多样性，而这两者本身又是一对矛盾，需要在两者之间进行权衡，本小节将重点研究四种评价指标对这两点的侧重和具体的超参数优化效果。

4.5.1 评价指标对优化结果影响的比较实验

在本次实验中，鉴于 BEGAN 在生成复杂图片时的表现较好，并且提供了一个超参数可以权衡 GAN 的生成图片清晰度和多样性，本实验选取了 BEGAN 作为实验所用的网络。

本实验的数据集为 CelebA，这是目前研究界十分常用的一种人脸数据集，本实验中也是以 CelebA 为训练集，让 BEGAN 来生成人脸图像。

由于本实验研究的是评价指标对超参数优化的影响，所以实验选择了网格搜索作为搜索算法，这样可以确保每次基于不同评价指标进行超参数搜索时，待选的超参数都是相同的，不会引入 explore 过程中得到的新的不确定的超参数，另外，由于网格搜索遍历了整个超参数空间，搜索最终得到的超参数一定是给定的超参数空间中最优的。

由于数据集较大也较为复杂，本实验又旨在研究四种评价指标对清晰度和多样性的侧重，所以本实验只调节一个超参数：即 BEGAN 中独有的权衡清晰度和多样性的超参数 γ 。

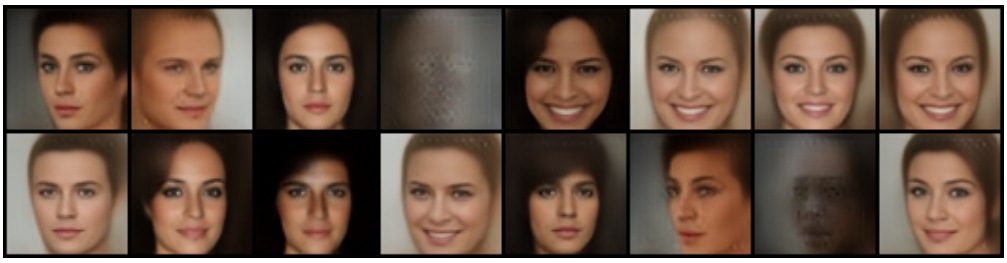
本实验中，输入数据尺寸为 64 * 64 * 3，学习率为 0.00008，batch_size 取 16，优化器为 Adam（$\beta_1 = 0.5$，$\beta_2 = 0.999$），$\lambda_k = 0.001$，待选的超参数 γ 的列表为[0.3,, 0.5, 0.7]。本实验基于 IS、JS-IS、FID 和 WD 进行超参数搜索，以找到各个评价指标下最优的 γ 。

4.5.2 结果展示与分析

下面展示 3 次实验的结果如表 4-17。

表 4-17 BEGAN 实验结果（迭代 60000 次）

评价指标	选择的 γ	对应图片
IS	0.5	图 4-19(b)
JS-IS	0.5	图 4-19(b)
FID	0.5	图 4-19(b)
WD	0.7	图 4-19(c)



(a) γ =0.3 生成图片

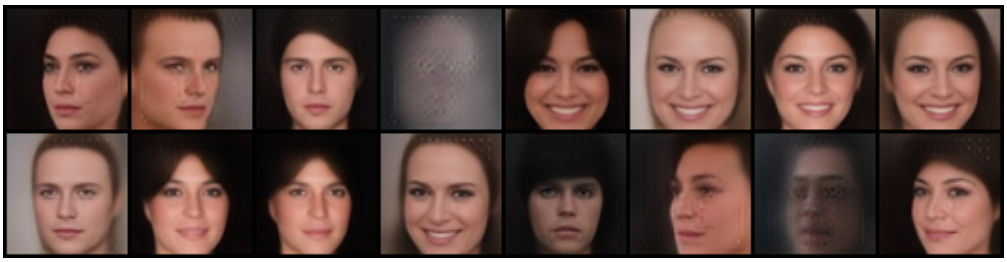
(b) γ =0.5 生成图片

(c) γ =0.7 生成图片

图 4-19 生成图片(迭代 60000 次)

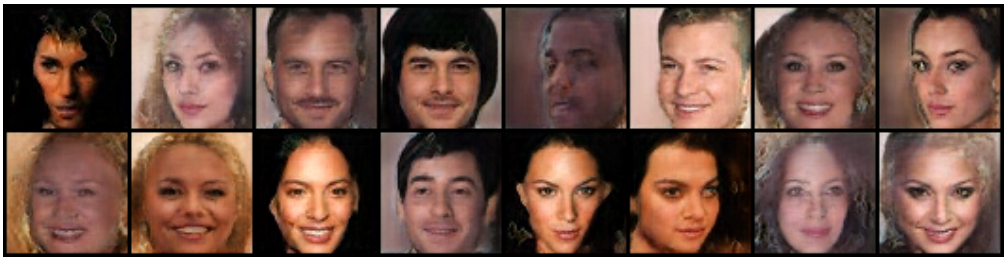
表 4-18 BEGAN 实验结果（迭代 80000 次）

评价指标	选择的 γ	对应图片
IS	0.3	图 4-20(a)
JS-IS	0.3	图 4-20(a)
FID	0.7	图 4-20(c)
WD	0.7	图 4-20(c)



(a) γ =0.3 生成图片

(b) γ =0.5 生成图片

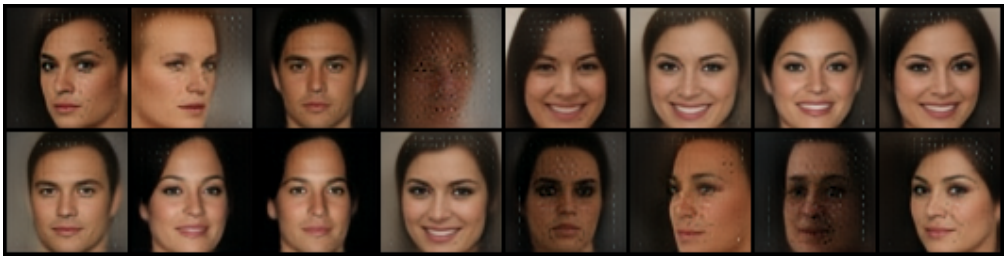


(c) γ =0.7 生成图片

图 4-20 生成图片(迭代 80000 次)

表 4-19 BEGAN 实验结果（迭代 100000 次）

评价指标	选择的 γ	对应图片
IS	0.3	图 4-21(a)
JS-IS	0.3	图 4-21(a)
FID	0.7	图 4-21(c)
WD	0.7	图 4-21(c)



(a) γ =0.3 生成图片

(b) γ =0.5 生成图片

(c) γ =0.7 生成图片

图 4-21 生成图片(迭代 100000 次)

从表 4-17、表 4-18、表 4-19 可以看出，IS 和 JS-IS 选择的超参数都是相同的，而 FID 和 WD 选择的超参数也较为接近相同的次数也居多，这在理论上也是比较好解释的，IS 和 JS-IS 基于的都是 p(y|x)和 p(y)之间的距离，并且 KL 散度和 JS 散度本就原理相似，导致了 IS 和 JS-IS 的超参数搜索结果也相同。而 FID 和 WD 也都是将图片的特征看成是多变量的高斯分布，从而用一些数学距离来描述两者之间的区别，思路是相似的，因此超参数搜索的结果较为相近也就不难解释了。

从图 4-19、图 4-20、图 4-21 可以看出，首先本次实验中 BEGAN 生成的图片都是有效的，人眼观感尚可，但质量上还是有些区别，总体上说 γ 为 0.3 和 0.5 时的图片清晰度比 γ 为 0.7 时要高，但 γ 为 0.3 时也出现了个别图片偏暗的情况。另外，通过图 4-21 就可以看出不同的评价指标对清晰度和多样性的侧重。IS 和 JS-IS 选择的是 γ =0.3，这时生成的图片清晰度较高，细节较为锐利，但多样性不足，如本次试验中生成的基本上都是长相相似的女性和个别清秀的男性，整体五官风格相同。而 FID 和 WD 选择了 γ =0.7，这时生成的图片相比之前多样性就要上升许多，有男有女，并且同性之间脸型，五官等特征也不相同，但缺点就在于清晰度不如 γ =0.3 时的生成图片，有些细节也有些失真。

可以看出，在图片质量都尚可的情况下，IS 和 JS-IS 更加侧重于生成图片的清晰度，而 FID 和 WD 则更加侧重生成图片的多样性。

4.5.3 评价指标加权

鉴于之前的实验结论：IS 和 JS-IS 侧重清晰度，而 FID 和 WD 侧重多样性，本文提出一种评价指标加权的思想，即定义一种评价指标，将这两类评价指标整合起来。以整合 IS 和 FID 为例：

$$Cri = IS（or\ JS\text{-}IS） + \lambda * FID(or\ WD) \tag{4-1}$$

如式 4-1 所示，本文提出的评价指标 Cri 将 IS 和 FID 通过加权求和整合到了一起，如果本次 GAN 的训练倾向清晰度，则将 λ 调小一些，如果倾向多样性，则将 λ 调大一些。其中 IS 也可以换成 JS-IS；FID 可以换成 WD。

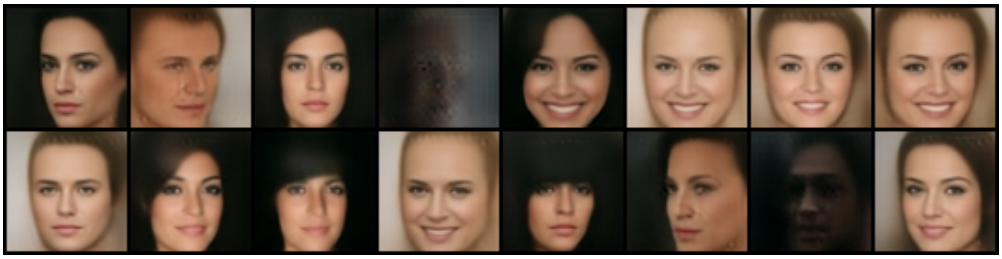
式 4-1 的核心思想就是在生成图片的清晰度和多样性之间通过一个 λ 进行折中，使超参数搜索的结果可以按照需求有所侧重，也可以保持清晰度和多样性之间的平衡，让评价指标更加灵活。

下面基于这一项评价标准，在同一个 BEGAN 上进行 γ 参数的搜索，进行了 70000 次迭代之后得到的结果如表 4-20。对应的图片生成结果如图 4-22。

表 4-20 BEGAN 指标加权实验结果（迭代 70000 次）

λ	选择的 γ	对应图片
0.001	0.3	图 4-22(a)
0.01	0.5	图 4-22(b)
1	0.7	图 4-22(c)

从表 4-20 可以看出，当 λ 很小时，超参数搜索的结果和只考虑 IS，JS-IS 时的相同，当 λ 很大时，超参数搜索的结果和只考虑 FID、WD 时的相同，而当 λ 取它们中间的一些值时，超参数搜索的结果综合考虑了清晰度和多样性。图 4-22 展示了实验对应的图片生成结果。如之前提到的，λ 取值很小时，选择的超参数 γ =0.3 对应的生成图片的清晰度比 γ =0.7 的要高，图片更加锐利，λ 取值较大时，选择的超参数 γ =0.7 对应的生成图片多样性要高于 γ =0.3 的图片，而当 γ 取值中等时，选择的超参数 γ =0.5 对应的生成图片在清晰度和多样性之间会有一些折中和均衡。



(a) γ =0.3 生成图片

(b) γ =0.5 生成图片

(c) γ =0.7 生成图片

图 4-22 生成图片(迭代 70000 次)

从图 4-22 可以看出，γ =0.5 时的图片基本上均衡了清晰度和多样性，在三张图片中属于质量比较好的。可见指标加权的思想在 λ 设置得当的情况下，对超参数优化带来了一定的效果。

当然，介于不同的评价指标的取值范围各不相同，在调节 λ 的值的时候仍需要一定的技巧，可以考虑利用正则化等操作来减少取值范围不同对指标加权带来的困难，这也是未来可以继续开展工作的一个方向。

4.6 本章小结

本章实现了 GAN 超参数优化方法，并进行了综合性的比较实验，实验包括 GAN 评价指标实验、搜索算法比较实验和评价指标对搜索结果影响的实验。

首先介绍了实验环境，所选择的数据集和实验所用的 tfgan 工具包。

在 GAN 评价指标实验中，基于 tfgan 工具包对四种评价指标进行了计算。在数字数据集和真实图片数据集上分别进行了实验，得出了新提出的两种评价指标和原有的两种指标 IS 和 FID 在训练过程中的变化情况，比较了对图片生成质量、模式坍塌程度和标签正确率的描述能力。这些评价指标在评价 GAN 的生成图片清晰度方面有着一定的意义，并且给出了 IS 和 JS-IS 越大，FID 和 WD 越小，生成的图片质量就相对越高的结论。比较实验结果，得到 JS-IS 的稳定性高于 IS，但灵敏性略逊于 IS。在评价清晰度和模式坍塌情况时，WD 的灵敏度都要优于其他三种指标。

最后，通过实验，证明了用这四种评价指标评价有监督 GAN 生成图片的标签准确性都是不可靠的。进而提出用真实数据集训练过的深度卷积网络作为训练集，生成图片作为测试集，将测得的分类准确率作为有监督 GAN 标签准确性的评价指标，通过实验证明了这种方法对有监督 GAN 标签准确性具有很好的描述能力。

在搜索算法比较实验中，本文实现了网格搜索和随机搜索这两种已有的超参数搜索方法和一种新的超参数搜索方法：exploit & explore 算法，进行了实验比较了三种算法的优缺点，证明了 exploit & explore 算法选择的超参数分布，有效搜索次数和准确率都要优于传统的两种方法。

接着研究了评价指标的不同对超参数搜索结果的影响，进行得出了各个评价指标对生成图片清晰度和多样性的侧重，最后，基于实验结论，提出了指标加权对生成图片的清晰度和多样性做了折中。

提出了这样一套体系：创新性地以几种 GAN 的评价指标作为超参数搜索的准则，利用已有的网格搜索和随机搜索，以及本文新提出的 exploit & explore 算法对 GAN 进行超

参数的搜索。第五章将基于此思想，实现一套自动化的 GAN 的超参数优化系统。

第五章 Auto-GAN 系统设计与实现

目前，市面上已经有一些针对传统机器学习算法的自动化超参数优化工具，比如 auto scikit-learn 这些工具往往将损失函数等作为搜索准则，支持传统的网格搜索和随机搜索，结合贝叶斯优化和元学习等技术进行优化，但这些自动化超参数优化工具尚未支持 GAN。2018 年初，谷歌公司提供了 AutoML 的云服务，让用户只需要提供数据，便可自动生成模型架构和对应的超参数，但这项服务中的模型架构等都不能由用户指定，和技术人员的需求显然是不符的。加之这项服务也是收费的，使得 GAN 的超参数调优目前仍依赖于技术人员手工进行调节，效率较低。

本文基于第三章和第四章中对 GAN 的评价指标和超参数搜索算法的研究，实现了一个对于 GAN 的自动超参数调优的系统 Auto-GAN。创新性地将研究过的四种评价指标作为超参数搜索的搜索准则，实现了网格搜索、随机搜索和本文中提出的 exploit & explore 算法，形成了一套完整的系统，用户可以通过该系统对 GAN 进行自动化的超参数调节：即在指定了待选的超参数空间之后，无需其他人工干预，系统会自动地进行超参数搜索，返回搜索到的最优的超参数，用户也可以通过可视化界面查看到模型的结构图和训练的过程。

下面介绍 Auto-GAN 系统的设计和具体实现。

5.1 Auto-GAN 系统设计

Auto-GAN 系统共分模型、搜索算法与搜索准则、结果返回与可视化三部分。

模型方面，系统支持的无监督 GAN 模型包括 DCGAN、WGAN、WGAN-GP 和 BEGAN，有监督 GAN 包括 CGAN 和 ACGAN。

搜索算法与准则方面，系统实现了离散超参数空间中的网格搜索，随机搜索和 exploit & explore 算法，其中随机搜索和 exploit & explore 算法每一步选择的超参数会记录到对应的 txt 文件中。基于改写过的 tfgan 代码，实现了 IS、JS-IS、FID、WD 这四种评价指标作为搜索准则。

结果返回和与可视化方面：本系统返回所选出的最优的超参数组合和此时对应的评价指标的值。另外，用户可以在浏览器中查看到网络的结构和训练过程。

系统的总体架构设计如图 5-1。

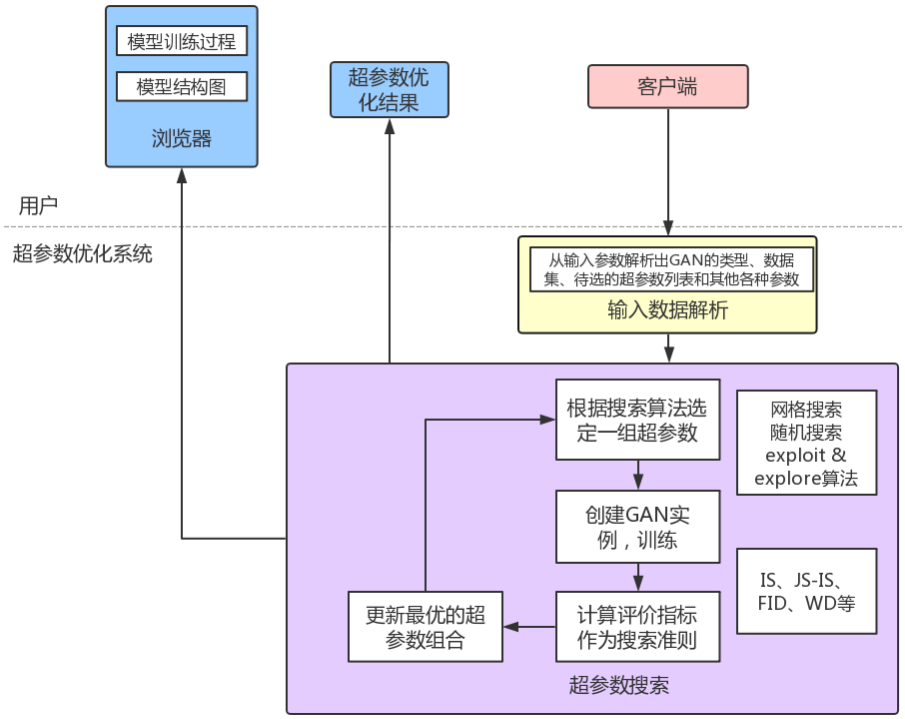


图 5-1 Auto-GAN 系统架构图

如图 5-1 所示，本系统的运行流程主要分为以下几步：

（1）用户通过命令行输入 GAN 的类型、数据集、搜索所基于的评价指标、搜索的算法、待选的超参数列表等数据，本系统解析这些数据。

（2）基于某种搜索算法选定一组超参数，用之前解析出的数据和这组超参数创建一个 GAN 的实例，对它进行训练。

（3）在一个 GAN 实例训练完成之后，计算评价指标作为搜索准则，这里的评价指标可以是本文中提到的 IS、JS-IS 等，也可以是它们加权的和或者是用户另外定义的其他准则。

（4）根据计算出的评价指标，对最优的超参数组合进行更新，然后在按照搜索算法，选择一组新的超参数，创建新的 GAN 实例，如此循环直至搜索结束。

（5）搜索结束后，返回搜索的结果。用户可以通过浏览器查看最优模型的结构和训练过程。

5.2 Auto-GAN 系统实现

5.2.1 系统环境

本系统运行在 GPU 服务器上，为 linux 操作系统，在深度学习框架的选择上，现如今主要有 Caffe、Tensorflow、Pytorch 等框架，本文中展示的实验均由 Tensorflow 和 Pytorch 进行实现，一般来说，Tensorflow 更加适合应用于大型工程，而 Pytorch 具有比 Tensorflow 更友好的调试功能，比较适合用于研究工作。

5.2.2 系统具体实现

在实现系统时，考虑到两种框架的性质以及 Tensorflow 中提供的 tfgan 工具包可以大大提升开发效率，也考虑到如今 Tensorflow 的普及度仍然要高于 Pytorch 等其他深度学习框架，在工业界的应用也更为广泛。基于以上原因作者选择了 Tensorflow 作为开发框架。

模型方面，每个模型通过一个类来实现，每个类内部存放学习率，batch_size 等成员变量，提供 build_model、train、load、save 等方法来进行模型的建立，训练，加载和保存的工作。

搜索算法方面，每个搜索算法通过一个函数实现，输入每一维上待选的超参数列表和选择的搜索准则，返回搜索到的最优的超参数组合和此时对应的搜索准则的取值。

搜索准则方面，每个搜索准则由一个函数实现，本文中修改了 tfgan 的源码，通过调用其中的接口完成了方法，输入生成图片和真实图片，输出对应的搜索准则的取值，也可以定义其他的搜索准则。

参数配置方面，表 5-1 展示了系统所需要传输的参数。其中包括了 GAN 的类型，系统现在已经支持 DCGAN、BEGAN、WGAN、CGAN、ACGAN 等主流的无监督和有监督 GAN；数据集的名称，比如 mnist、celebA 等；训练轮数 batch 大小等；允许用户指定模型、生成图片、训练日志的保存路径，选择所需要的评价指标和超参数搜索方法

表 5-1 参数列表

参数名称	参数类型	描述
gan_type	str	GAN 的类型
dataset	str	数据集名称
epoch	int	训练轮数
batch_size	int	batch 的大小
z_dim	int	噪声维度
checkpoint_dir	str	模型保存路径
result_dir	str	生成图片保存路径
log_dir	str	日志保存路径
criterior	str	评价指标
search_method	str	搜索算法

可视化方面，基于 Tensorflow 提供的 tensorboard，可以通过浏览器查看模型图和训练的过程与结果。

5.3 Auto-GAN 系统展示

本文中涉及到的许多实验都可以通过本系统进行实现，在本小节中，将简单展示系统的使用。在下面的例子中，将展示用该系统调节传统 GAN 中学习率和 D 学习次数这两个参数的过程。

5.3.1 搜索准则展示

首先，用户通过命令行输入必要的参数。系统解析获得的参数，进行自动化的超参数优化搜索，最终，系统输出搜索到的最优的超参数组合和此时对应的评价指标的值。图 5-2 为用户分别基于四种评价指标作为搜索准则，用网格式搜索进行超参数优化的输出结果。

```
[*] Training finished!
[*] Testing finished!
criterior = 6.276811
best learning rate = 0.002000
best discriminator iteration time = 10.000000
total time = 6696.79
```

（a）IS 指标

```
[*] Training finished!
[*] Testing finished!
criterior = 2.308072
best learning rate = 0.002000
best discriminator iteration time = 10.000000
total time = 6654.16
```

（b）JS-IS 指标

```
[*] Training finished!
[*] Testing finished!
criterior = 7.647928
best learning rate = 0.002000
best discriminator iteration time = 4.000000
total time = 6169.25
```

(c) FID 指标

```
[*] Training finished!
[*] Testing finished!
criterior = -1369.336792
best learning rate = 0.002000
best discriminator iteration time = 10.000000
total time = 6294.49
```

（d）WD 指标

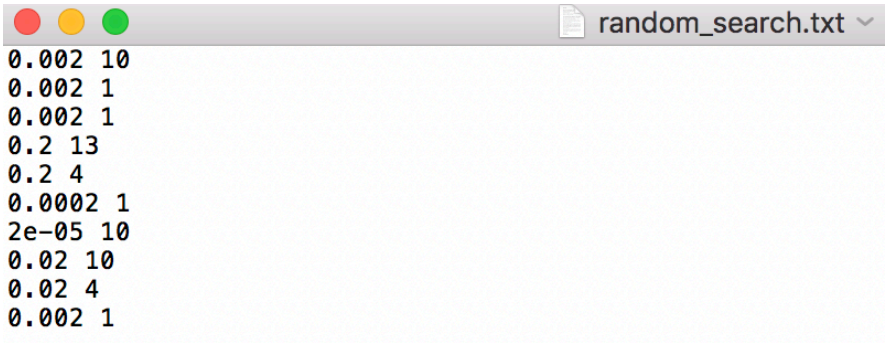
图 5-2 基于四种搜索准则的超参数搜索结果

系统对每一次 GAN 的训练结果都进行了保存，生成的文件如图 5-3 所示。

图 5-3 系统保存的训练结果

5.3.2 搜索算法展示

Auto-GAN 系统对随机搜索和 exploit & explore 算法每次选择的超参数都进行了记录，如图 5-4 所示。



（a）随机搜索所选参数

exploit_explore.txt

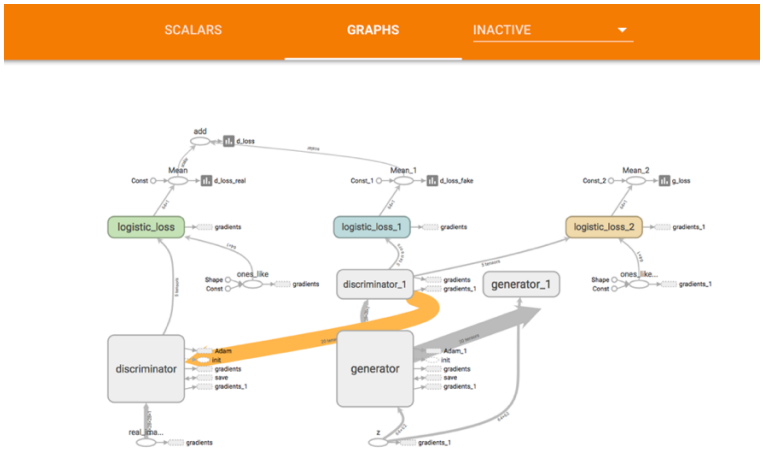
```
2e-05 7
0.002 10
0.0024 1
0.02 9
0.0016 11
0.0002 7
0.00128 11
0.2 4
0.0016 10
0.002 9
```

（b）exploit & explore 算法所选参数

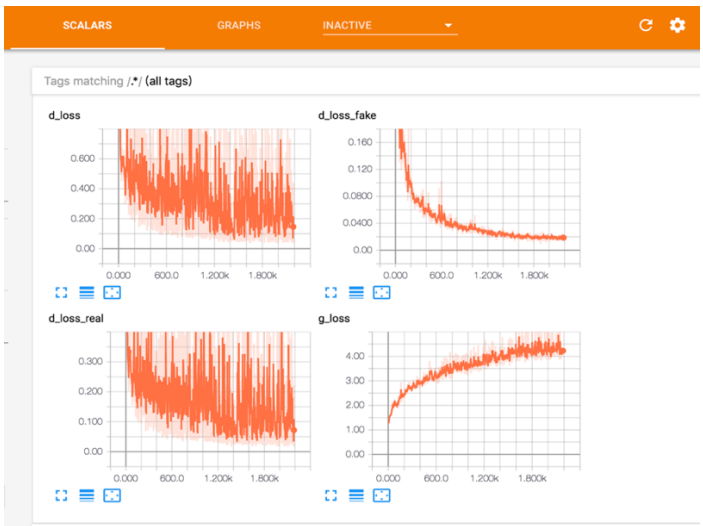
图 5-3 所选参数记录

5.3.3 可视化界面展示

用户也可以通过浏览器查看模型的结构和训练的结果。如图 5-4 所示，图中为一个 GAN 的模型图和它在训练过程中，损失函数各个部分的变化曲线。



(a)查看模型结构



(b) 查看训练结果

图 5-4 可视化界面

本系统可以让机器学习工程师和研究员进行一些 GAN 的自动超参数选择工作，充分利用计算机的运算资源，搜索过程中不再需要任何的人工干预，搜索结果、模型信息和训练过程由系统保存，可供用户在搜索完成后随时查看和分析。

在实际使用中，将自动调优作为粗调步骤，在缩小了超参数的选择空间之后再进行细调，将自动调优和手工调优进行结合可以达到更好的优化效果，相比纯手工调优大大减少了工程师或者研究人员的工作量，提高了效率。

5.4 本章小结

目前市场上尚未有合适的支持 GAN 的超参数调优工具，导致 GAN 的超参数调优依赖人工调节，效率较低。

整合之前的研究成果，利用传统的网格搜索、随机搜索和本文提出的 exploit & explore 算法三种超参数优化方法，创新性地将已有的 IS、FID 和本文提出的 JS-IS、WD 四种评价指标作为搜索准则，对各种常见的无监督 GAN，有监督 GAN 提供了一套自动的超参数优化系统 Auto-GAN。

Auto-GAN 提供了用户接口和可视化界面，自动搜索过程无需任何人工干预，训练模型图和训练过程均可通过可视化界面查看。使用本系统或者采用本系统和手工调优相结合的方法进行 GAN 的超参数优化比纯手工调优效率大大提高，为研究和工程工作的开展都可以带来很大的便利。

本章主要介绍了 Auto-GAN 系统的设计思路和具体实验，展示了工具的使用。

第六章 总结与展望

6.1 总结

本文的主要工作如下：

（1）在传统的机器学习算法自动调优工具中，往往采用损失函数作为搜索准则寻找最优的超参数组合，由于 GAN 的训练是一个博弈过程，这种方法显然是不适用的。本文创新性地提出用 GAN 的评价指标作为搜索准则，更加准确地评价了 GAN 在不同超参数下的训练表现。

（2）本文调研了现有的 GAN 的评价指标 IS 和 FID，从它们的数学原理出发分析了它们存在的不足，提出了两种新的评价指标 JS-IS 和 WD。通过在黑白数字数据集和彩色真实物体数据集上分别进行实验，验证了四种评价指标对衡量无监督和有监督 GAN 生成图片的清晰度有一定的意义，反映了 GAN 生成图片的质量，可以作为调节 GAN 的超参数时所用的评价准则。并且指出 JS-IS 的稳定性优于 IS，但灵敏性不如 IS。设计实验证明了 FID 和 WD 在评价无监督 GAN 模式坍塌情况方面要优于 IS 和 JS-IS。通过比较，总结出无论是评价清晰度还是模式坍塌程度，WD 的灵敏性都要优于其他三种指标。

（3）本文通过在黑白数字数据集和彩色真实物体数据集上进行实验，证明现有的 IS、FID 和本文提出的 JS-IS、WD 都不能衡量有监督 GAN 中生成图片的标签准确率。之后，本文将标签准确率作为有监督 GAN 的一项新的评价指标，给出了标签准确率的计算方法。

（4）本文调研了现有的两种超参数搜索方法：网格搜索和随机搜索，在数字数据集和真实物体数据集上通过实验比较了两者的效果和效率。发现网格搜索资源开销大，可以保证搜索到待选超参数空间中最优的点；随机搜索可以大量减少时间开销，并且大部分时候也可以搜索到较优的超参数组合，是一种性价比比较高的方法，但稳定性上肯定有所欠缺。

本文新提出了 exploit & explore 算法，这种算法的思想类似于遗传算法，在资源开销方面与随机搜索相近，但它的 explore 过程可以向超参数空间添加新的点，提高了超参数空间的质量。本文将 exploit & explore 算法和传统的网格搜索、随机搜索在选择分布、有效搜索次数、准确率三个方面分别进行了比较，得出 exploit & explore 算法选择的超参数的分布更加聚集在最优点的周围，有效搜索次数和准确率都要高于网格搜索和随机搜索，证明了 exploit & explore 算法是一种更加高效，准确的超参数搜索方法。

（5）本文的最重要、核心的贡献是提出了这样一套思路：构造一套 GAN 的超参数优化系统，基于某个 GAN 的评价指标作为超参数搜索的准则，在给定的超参数空间中通过某种搜索方法，搜索出最优的超参数组合。此外，本文也比较了四种评价指标在选择超参数方面的侧重和不同，推断 IS 和 JS-IS 略侧重清晰度，而 FID 和 WD 更侧重多样性，进而提出了指标加权的思想在两者之间进行平衡。当然，在评价指标对超参数选择的影响上，本文所做的实验还是十分有限的，还需要大规模的实验才能得到可靠的结论，但指标加权的思想依旧对超参数选择有意义。

（6）目前市面上尚未有合适的针对 GAN 的超参数自动优化工具，人工调节的效率又较低。针对这一问题，本文实现了一套 GAN 的自动超参数优化系统，创新性地将 IS、JS-IS、FID、WD 这四种 GAN 的评价指标作为超参数的搜索准则，支持了网格搜索、随机搜索和 exploit & explore 算法，对 GAN 的超参数进行自动化的调优。

第五章展示了本文实现的系统：系统支持现在比较常用的几种无监督和有监督 GAN，并且在模型方面具有灵活的扩展性，超参数优化过程不需要任何的人工干预，并且提供了可视化的界面用于查看模型和训练结果。

这样的一套系统可以应用到实际的工程和研究中，为研究人员和工程师提供 GAN 的自动超参数优化工具。用户可以通过该系统得到一组可用的超参数，也可以利用系统进行自动调优得到的超参数将超参数的待选范围大大缩小，结合人工调节，得到更优的超参数，相比纯手工操作大大提升了超参数调节的效率和准确性。

6.2 展望

GAN 的评价指标和自动化机器学习都是非常值得深入研究的方向，由于时间和作者的研究水平有限，目前还存在着一些问题有待未来进一步研究：

（1）在 GAN 的评价指标，也是本文中的超参数搜索准则方面，IS 和 FID 提供的是两种不同的思路，本文中将它们用 JS 散度和 Wasserstein 距离重新定义，得到了 JS-IS 和 WD，并且进行了一些实验研究了几种评价指标的侧重点。但现在研究界对 GAN 的评价指标还没有一套明确的共识，本文中也发现了各种评价指标尚不能衡量标签准确性，本文中对标签准确性的问题，采取了将分类准确率作为评价指标的方法来解决，但作者认为这种方法还是略显生硬。最理想的评价指标应该能够同时反映清晰度、模式坍塌程度和标签准确性。目前，研究界也认为 GAN 领域的一个突出的问题是缺乏公认的定量评价指标[34]，希望未来能够研究得到更加成熟、完备、准确的 GAN 的评价指标。

（2）在超参数搜索优化方面，网格搜索和随机搜索在工业界往往与人工调节结合起来进行超参数优化，是两种非常常用的搜索算法。本文借鉴遗传算法的思想提出了新的一套 exploit & explore 算法。如今，自动化机器学习的概念也引起了研究界的广泛关注，learning to learn 的思想打开了一个极其值得研究的领域。在传统机器学习方面已经有研究人员利用贝叶斯优化等算法来进行超参数优化，在深度学习算法中也有研究人员提出用 LSTM[35]等方法来进行自动的超参数优化[36]。用一个神经网络来对另一个神经网络进行超参数优化是一个很有创新性，值得研究的课题。众所周知，调参是机器学习中比较繁琐但也极其重要的一环，GAN 的训练过程比较自由，对超参数的要求更加苛刻。作者也希望后续能够研究得到适合 GAN 的更加高效、巧妙、自动化的超参数搜索方法。当然，介于调参无论如何都是一件费时费力的事情，也希望未来能够研究提出对超参数容忍度较高，较为稳定的模型。

（3）本文所实现的系统是在单 GPU 环境中实现的，如今分布式计算可以大大提升计算效率，Tensorflow 可以支持多 GPU 运算，此外也有方法可以让本文中使用的 Tensorflow 在 Spark 环境中运行，将系统部署到分布式的集群中去。理论上本文中的网格搜索和随机搜索完全是可以分布式实现的，而 exploit & explore 算法由于当前选择的参数要依赖之前搜索的结果，无法进行完全的分布式，但可以借用 batch 的思想，一次同时尝试几种超参数组合，从而实现部分分布式的系统，在时间开销和资源开销之间做出折中，让整个系统的效率更

高，也有更好的实用价值。

（4）本文未能在 ImageNet 数据集上对 ACGAN 进行复现。2018 年最近的 ICLR 的工作 SNGAN[37]和标签投影[38]的结合可以解决这个问题，生成的图片质量堪称惊艳。但由于复现工作依赖 chainer 框架，不能包括到本文的工作中去，而将这项工作移植到 Tensorflow 上存在着一些困难，希望之后能够将更多优秀的工作集成到这套系统中去，让系统更加丰富。

参考文献

[1] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]. Advances in neural information processing systems. 2014: 2672-2680.
[2] Arjovsky M, Chintala S, Bottou L. Wasserstein gan[J]. arXiv preprint arXiv:1701.07875, 2017.
[3] Berthelot D, Schumm T, Metz L. Began: Boundary equilibrium generative adversarial networks[J]. arXiv preprint arXiv:1703.10717, 2017.
[4] Mao X, Li Q, Xie H, et al. Least squares generative adversarial networks[C].2017 IEEE International Conference on Computer Vision (ICCV). IEEE, 2017: 2813-2821.
[5] Lucic M, Kurach K, Michalski M, et al. Are GANs Created Equal? A Large-Scale Study[J]. arXiv preprint arXiv:1711.10337, 2017.
[6] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in neural information processing systems. 2012: 1097-1105.
[7] Deng J, Dong W, Socher R, et al. Imagenet: A large-scale hierarchical image database[C]. Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 2009: 248-255.
[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.
[9] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. Cvpr, 2015.
[10] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[11] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.
[12] Feurer M, Klein A, Eggensperger K, et al. Efficient and robust automated machine learning[C]. Advances in Neural Information Processing Systems. 2015: 2962-2970.
[13] Li L, Jamieson K, DeSalvo G, et al. Hyperband: A novel bandit-based approach to hyperparameter optimization[J]. arXiv preprint arXiv:1603.06560, 2016.
[14] Zeiler M D, Krishnan D, Taylor G W, et al. Deconvolutional networks[C]. Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 2528-2535.
[15] Odena A. Semi-supervised learning with generative adversarial networks[J]. arXiv preprint arXiv:1606.01583, 2016.
[16] Tolstikhin I O, Gelly S, Bousquet O, et al. Adagan: Boosting generative models[C]. Advances in Neural Information Processing Systems. 2017: 5430-5439.
[17] Gulrajani I, Ahmed F, Arjovsky M, et al. Improved training of wasserstein gans[C]. Advances in Neural Information Processing Systems. 2017: 5769-5779.
[18] Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.
[19] Odena A, Olah C, Shlens J. Conditional image synthesis with auxiliary classifier gans[J]. arXiv

preprint arXiv:1610.09585, 2016.
[20] Chongxuan L I, Xu T, Zhu J, et al. Triple generative adversarial nets[C]. Advances in Neural Information Processing Systems. 2017: 4091-4101.
[21] Gan Z, Chen L, Wang W, et al. Triangle generative adversarial networks[C]. Advances in Neural Information Processing Systems. 2017: 5253-5262.
[22] Salimans T, Goodfellow I, Zaremba W, et al. Improved techniques for training gans[C]. Advances in Neural Information Processing Systems. 2016: 2234-2242.
[23] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2818-2826.
[24] 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: 6629-6640.
[25] Nelder J A, Mead R. A simplex method for function minimization[J]. The computer journal, 1965, 7(4): 308-313.
[26] Hinton G E. A practical guide to training restricted Boltzmann machines[M]. Neural networks: Tricks of the trade. Springer, Berlin, Heidelberg, 2012: 599-619.
[27] Chang C C, Lin C J. LIBSVM: a library for support vector machines[J]. ACM transactions on intelligent systems and technology (TIST), 2011, 2(3): 27.
[28] Bergstra J, Bengio Y. Random search for hyper-parameter optimization[J]. Journal of Machine Learning Research, 2012, 13(Feb): 281-305.
[29] Isola P, Zhu J Y, Zhou T, et al. Image-to-image translation with conditional adversarial networks[J]. arXiv preprint, 2017.
[30] Pathak D, Krahenbuhl P, Donahue J, et al. Context encoders: Feature learning by inpainting[C]. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 2536-2544.
[31] Zhu J Y, Park T, Isola P, et al. Unpaired image-to-image translation using cycle-consistent adversarial networks[J]. arXiv preprint arXiv:1703.10593, 2017.
[32] Kullback S, Leibler R A. On information and sufficiency[J]. The annals of mathematical statistics, 1951, 22(1): 79-86.
[33] Olkin I, Pukelsheim F. The distance between two random vectors with given dispersion matrices[J]. Linear Algebra and its Applications, 1982, 48: 257-263.
[34] 王万良，李卓蓉. 生成式对抗网络研究进展[J]. 通信学报，2018，39(2)：135-148.
[35] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[36] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks[J]. arXiv preprint arXiv:1703.03400, 2017.
[37] Miyato T, Kataoka T, Koyama M, et al. Spectral normalization for generative adversarial networks[J]. arXiv preprint arXiv:1802.05957, 2018.
[38] Miyato T, Koyama M. cGANs with projection discriminator[J]. arXiv preprint arXiv:1802.05637, 2018.

谢辞

经过一段时间的努力，我的本科毕业设计已经基本完成。从一开始对生成式对抗网络和各种深度学习框架一无所知，到之后阅读各种文献，学习框架，进行实验，这个过程使我成长了许多。我清楚地明白这篇论文中的工作仍十分稚嫩，但我会将它看成我在深度学习和计算机视觉方面研究的一个起点，在之后的日子里不断继续探索。

在毕设的过程中我从国内外研究人员的工作中学到了许多许多，他们扎实的研究工作让我万分敬佩。感谢他们用自己的研究成果推动了一个领域的进步，也让无数与我一样处于入门阶段的学生汲取到了许多养分。在谢辞的开始，请允许我先向他们致敬！

本篇毕业设计标志着我大学四年的结束，感谢我的导师：上海交通大学的张月国老师和清华大学的龙明盛老师。张老师在整个毕设的过程中，一直和我进行交流，在我撰写论文的过程中他对我的论文提出了十分细致的修改建议，指导我进行了好几次大改，让我受益颇多。龙老师对我进行了细致的指导，为我的实验提供了 GPU 等计算资源。他厚实的科研功底和严谨忘我的科学态度让我自叹弗如，也让我以更高的标准去要求我自己，向着成为像老师这样的人不断努力。在此，请让我对两位老师表达由衷的感谢，感谢你们的指导，辛苦你们了！

回顾大学四年，成成败败，悲悲喜喜，如今都成了成长路上独一无二的风景。在我的毕设致谢中，我想向上海交通大学表示我由衷的感恩之情，能够成为交大人是我莫大的幸运和骄傲，感谢母校！感谢所有交大的老师们传道授业解惑之恩，祝你们桃李天下。感谢我的室友沈倩颖，吴雨桐，和现在已经远在日本的马诗慧，感谢你们这几年的陪伴，包容和帮助。感谢所有信安的同学，感谢你们的支持与鼓励，和你们度过大学四年真的是一件极其快乐的事情。感谢我的学长学姐们，倾囊相授地给我分享了许多经验，在我想要放弃的时候给了我信心，让我勇敢地继续走下去。愿大家都能有一个灿烂的前程，愿大家无论在中国，在世界的哪个地方都能获得成功和幸福，更愿大家春暖花开时，饮水思源处，岁岁常相见。

此外，我最要感谢的是我的家人，感谢 22 年来父母对我的关爱，对我的各种决定都表示了支持，让我能够去追求自己的梦想。在即将披上学士服之际，真的想对他们说：辛苦你们了！谢谢你们！

如今，本篇论文代表着我本科学习生活的结束，也将标志我下一段研究生涯的开始。本科四年的荣誉也好，遗憾也罢都已成过去，也十分感谢那个坚持下来了的自己，希望在未来的研究生涯中能做出更好的成果。

最后，再对所有帮助过我，陪伴过我，给我过爱和鼓励的人们说声谢谢。我是一个十分幸运的人，一路上总能遇到很多很好的人，给了我莫大的帮助和支持，这一切，我视如瑰宝，无以为报。本科毕业是我人生中一个很有意义的时刻，谨以此文，表达我最真挚的感谢之情。

HYPERPARAMETER OPTIMIZATION OF GENERATIVE ADVERSARIAL NETWORKS

Generative Adversarial Network (GAN) is invented by Ian GoodFellow in NIPS 2014. It simultaneously trains two models: generative model G which transforms noise to generated data and discriminative model D which tells generative data from real data. During the training procedure, G tries to fool D and D strives to distinguish fake from real, which is a two-player minimax game.

Rapid development of deep learning model makes it possible to extract features effectively. Combination of GANs and Convolutional Neural Networks(CNNs) makes GANs useful in Computer Vision(CV) and Natural Language Processing(NLP) tasks. Having strong ability to generate image data, GANs is widely applied into image-to-image translation, image style transformation and many other fields, but suffers from unstability. Due to structure and training procedure as a minimax game, GANs' training is much more complex and unstable than traditional machine learning models. Futhermore, different hyperparameters can contributed to totally different training results.

To address the problem of GANs' unstability, a large amount of GANs, such as WGAN, LSGAN and BEGAN are introduced. These GANs change loss functions, training methods and model architectures in order to make GANs more stable to train. But researchers from Google point out that after conducting large-scaled study, they find none of these GANs is definitely superior to the original one.

Comparably, demand for finding the optimal hyperparameter becomes more urgent in GANs' application. However, nowadays most researchers can only tune hyperparameters manually, which is very time-consuming. This paper conduct research on automatic hyperpameter optimization of GANs and implement a hyperparameter optimization system: Auto-GAN, which can obviously improve the efficiency of tuning hyperparameters of GANs.

The main contributions are as follows:

1. We introduce a novel hyperparameter search criterior for GANs. Different from the loss function in traditional machine learning algorithms, GANs' evaluation criteria are viewed as search criteria in our system to compare GANs' performance.

2. We introduce two new evaluation criteria for GANs, JS-IS and WD and compare them with the existing criterior: IS and FID.

3. We introduce one new search algorithm called exploit & explore algorithm, which can obviously improve search efficiency.

4. We conduct research on the role different criteria play in GANs' hyperparameter optimization and introduce criterior tradeoff to keep the balance between sharpness and diversity of generated images.

5. We implement a hyperparameter optimization system: Auto-GAN on TensorFlow. Based on four criteria and three search methods, Auto-GAN can tune hyperparameters for a variety of unsupervised and supervised GANs.

A hyperparameter optimization system of GANs consists of search criterior and method. In traditional machine learning algorithm, loss function is often taken as search criterior. However, loss function can't be the search criterior of GANs because GANs' training is a minimax game and the optimal solution shows Nash Equilibrium, which is much more complicated. This paper takes evaluation criteria of GANs as search criteria instead in order to enhance the quality of generated images.

The existing evaluation criteria of GANs are Inception Score(IS) and Frechet Inception Distance(FID). The definition of IS and FID can be found in 2.2. Inspired by Jensen Shannon Divergence, we introduce two new criteria: JS-IS and WD. Traditional search methods include Grid Search and Random Search. Grid Search means trying all the hyperparameters one by one to find the best one, while Random Search chooses hyperparameters randomly to enhance efficiency. Inspired by Genetic Algorithm, we introduce JS-IS, WD and exploit & explore algorithm. Trough experiments we compare the performance of evaluation criteria and search algorithm we introduced with traditional ones.

A reliable evaluation criterior of GANs should be able to evaluate sharpness of generated images, mode collapse phenomenon and label accuracy. Up till now, researchers have not reach consensus on GANs' evaluation criterior, and the performance of different criteria of GANs also has not be compared. In this paper we revise some code in tfgan tool to compute criteria and train unsupervised and supervised GANs, respectively on digit dataset MNIST and real object dataset CIFAR10 to compare the four criteria. Generated images and corresponding criteria values are shown to compare relation between image quality and criteria value. Trough large-scaled experiments, we find that all of them are helpful to evaluate image sharpness. Benefiting from mathematic properties of Jensen Shannon Divergence, JS-IS can remain useful under some extreme circumstances when IS becomes invalid, showing stronger stability than IS. FID and WD are more effective than IS and JS-IS in evaluating mode collapse phenomenon. WD performs best among four, both in evaluating image sharpness and mode collapse phenomenon. Experiment results are shown in 4.2.

In addition, all four criteria fail to evaluate label accuracy. In this paper, we use a classification model trained on real data to test the generated images and take classification accuracy as label accuracy, which is effective in evaluating label accuracy. When real dataset is very large, training classification model may take lots of time. We recommend to finetune a pretrained model, such as ResNet pretrained on ImageNet dataset. These pretrained models can be accessed on open-source websites easily and fine-tuning is much faster.

When it come to search method, we implement the two commoly used traditional search methods, Grid Search and Random Search. Grid Search can cover the whole hyperparameter space but is time-consuming, while Random Search can't find the optimal solution in hyperparameter space and is sometimes unstable.

Exploit & explore algorithm contains two procedures. In exploit procedure, similar to Random Search, it selects one hyperparameter randomly, trains the model, evaluates performance by evaluation criteria and keeps record of the best hyperparameter and corresponding criterior value.

In explore procedure, if the current hyperparameter performs better than all the previous ones, a small “disturbance” will be added to this hyperparameter. After disturbance, a new hyperparameter is added to hyperparameter space. Through two procedures, the quality of hyperparameter space can be improved when searching hyperparameters, which can improve the accuracy and efficiency of hyperparameter optimization. Detailed description of algorithm is shown in 3.5.

We apply exploit & explore algorithm and the two traditional methods to tune hyperparameters of GANs, both on digit data MNIST and real object dataset Fashion-MNIST. Exploit & explore algorithm is much more efficient than Grid Search and accurate than Random Search.

Useful hyperparameters of one specified GAN on one dataset often gather in one or several fields in hyperparameter space. Therefore, if the chosen hyperparameter lies in these field, this trial can be viewed as valid. Both in two dimensional and three dimensional hyperparameter space, exploit & explore algorithm outperforms other two methods obviously in hyperparameter distribution, validity and accuracy. 4.4 shows experiment results.

The role different search criteria play in hyperparameter optimization is studied by training BEGAN on CelebA dataset. IS and JS-IS prefer sharpness, while FID and WD favor diversity of generated images. We introduce criteria tradeoff to keep the balance between sharpness and diversity. Through changing tradeoff coefficient, sharp or diverse images that meet our needs will be generated.

Now some automatic machine learning tools, for instance, auto scikit-learn and AutoML, can be used in traditional machine learning and some deep learning models. But all these tools are not suitable for GANs. We implement an automatic hyperparameter optimization system Auto-GAN for GANs. Considering TensorFlow is the most popular framework in industry, we develop Auto-GAN on TensorFlow.

Auto-GAN resolves the parameters including GAN types, evaluation criterior and search methods. Then it searches one hyperparameter and trains the corresponding GAN model. If the current model outperforms all the previous ones, Auto-GAN will record its performance and corresponding hyperparameter. Based on four criteria and three search method, Auto-GAN can tune hyperparameter automatically and efficiently. All the generated images and search results are saved orderly in system and users can view model graph and training details through browsers.

With Auto-GAN, researchers can tune hyperparameters of GANs automatically, saving time in research. They can also combine Auto-GAN with manul search, which is far more efficient than pure manual search. So Auto-GAN is quite useful to researchers work on GANs. Futhermore, Auto-GANs is also easy to extend that researchers can define their own model or introduce other evaluation criteria.

In conclusion, this paper introduces two novel evaluation criteria for GANs: JS-IS and WD and one novel search method: exploit & explore algorithm. JS-IS is more stable than IS and WD performs best among four criteria. In addition, we suggest taking classification accuracy as label accuracy when evaluating supervised GAN. Exploit & explore algorithm outperforms traditional Grid Search and Random Search in hyperparameter distribution, validity and accuracy. Finally, this paper implements an automatic hyperparameter optimization system: Auto-GAN. Able to improve efficiency of hyperparameter optimization, Auto-GAN is very helpful to researchers working on GANs.

At the end of my Bachelor's Thesis, please allow me to express my sincere gratitude to Shanghai Jiao Tong University: my beloved Alma Mater.