

分类号_____

学校代码 10487

学号 M201877220

密级_____

华中科技大学

硕士学位论文

面向时序数据分类的 GAN 数据 增强

学位申请人： 常永林

学 科 专 业： 计算机技术

指 导 教 师： 石柯 教授

答 辩 日 期： 2020 年 6 月 4 日

**A Thesis Submitted in Partial Fulfillment of the Requirements for
the Degree of Master of Engineering**

**GAN based Data Augmentation for Time
Series Data Classification**

Candidate : YongLin Chang

Major : Computer Technology

Supervisor : Prof. Ke Shi

Huazhong University of Science & Technology

Wuhan 430074, P.R.China

May, 2020

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到，本声明的法律结果由本人承担。

学位论文作者签名：常永林

日期：2020 年 6 月 4 日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本论文属于 保密 ☐ ，在 _____ 年解密后适用本授权书。
不保密 ☒ 。

（请在以上方框内打“√”）

学位论文作者签名：常永林

日期：2020 年 6 月 4 日

指导教师签名：石所

日期：2020 年 6 月 5 日

摘要

在时间序列数据领域，深度学习技术已被用于完成分类任务，相比于传统的机器学习方法其分类效果有较大的提升。但当训练数据规模较小时，模型容易过拟合导致性能严重下降，因此就需要通过数据增强的方式扩大数据的规模，来减小过拟合发生的可能性。现有的数据增强技术能够扩充时间序列数据，但并不能保证扩充后的数据同原样本保持一致分布，而生成对抗网络能够基于原始数据分布生成新的相同分布的数据。现有的生成对抗网络不是针对时间序列数据设计的，因此需要因地制宜地对网络进行改进，以期达到好的效果。

TS-GAN 模型针对时间序列数据的特点和分类任务的要求，改进了原始 GAN 模型的网络架构和损失函数。通过对 JS 损失函数导致模型不稳定的分析，TS-GAN 选用 Wasserstein 距离来衡量生成分布和真实分布的距离，该衡量距离需要满足 lipschitz 连续性条件，为限制梯度参数爆炸，在损失函数中加入惩罚机制。时间序列是一维数据，鉴于卷积层生成特征较为稳定，G 网络舍弃全链接层，使用一维反卷积网络生成数据，以此来增强 G 网络的特征生成能力。D 网络中引入注意力机制，提高该网络对时间序列的相似性判定能力，进而督促 G 网络生成更好的分布。除此之外，为防止在数据增强时也出现不收敛的情况，TS-GAN 模型中的 D 网络需要使用实例标准化，G 网络和 D 网络更新次数也需要调整。

实验使用 GAN 和 TS-GAN 分别扩充来自 UCR 时序数据集的训练集，基于增强后的数据集训练 FCN 分类模型，再使用各数据集中的测试集验证分类效果。实验证明经 TS-GAN 扩充的数据集，其分类精确度和 F1 分数分别提高 4.19% 和 5.01%，而经 GAN 网络扩充的数据集无法提高性能，反而导致相应的指标相比下降 1.9% 和 1.94%。这说明 TS-GAN 能克服 GAN 扩充时间序列上的局限性，能够提高 UCR 数据集分类效果，说明 TS-GAN 模型具有较好的数据增强能力。

关键词： 生成对抗式网络 数据增强 深度学习 时间序列

Abstract

In the field of time series data, deep learning technology has been used to complete the task of classification. Compared with the traditional machine learning method, its classification effect is greatly improved. However, when the size of training data is small, the model is easy to over fit, resulting in serious performance degradation. Therefore, it is necessary to expand the size of data through data enhancement to reduce the possibility of over fitting. The existing data enhancement technology can expand the time series data, but it can't guarantee that the expanded data and the original sample keep the same distribution, and the generation countermeasure network can generate new data with the same distribution based on the original data distribution. The existing generate adversarial network is not designed for time series data, so it needs to be improved according to local conditions in order to achieve good results.

TS-GAN model improves the network architecture and loss function of the original GAN model according to the characteristics of time series data and the requirements of classification tasks. Through the analysis of JS loss function leading to the instability of the model, TS-GAN chooses Wasserstein distance to measure the distance between the generated distribution and the real distribution, which needs to meet the Lipschitz continuity condition. In order to limit the gradient parameter explosion, a penalty mechanism is added to the loss function. Time series is one-dimensional data. In view of the relatively stable generation characteristics of convolution layer, G network abandons the full link layer and uses one-dimensional deconvolution network to generate data, so as to enhance the generation ability of G network characteristics. The attention mechanism is introduced into the D-Network to improve its ability to judge the similarity of time series, and then to urge the G-Network to generate a better distribution. In addition, in order to prevent non convergence during data enhancement, the D-Network in TS-GAN model needs to use instance standardization, and the update times of G-Network and D-Network also need to be adjusted.

In the experiment, GAN and TS-GAN are used to expand the training set from UCR time series data set respectively, and FCN classification model is trained based on the enhanced data set, then the test sets of each data set are used to verify the classification effect. Experiments show that the classification accuracy and F1 score of the data set expanded by TS-GAN are increased by 4.19% and 5.01% respectively, while the performance of the data set expanded by GAN network can not be improved, but the corresponding indexes are decreased by 1.9% and 1.94% respectively. This shows that TS-GAN can overcome the limitation of the extended time series of GAN and improve the classification effect of UCR data set significantly, which shows that TS-GAN model has better data enhancement ability.

Key words: Generate adversarial network data enhancement deep learning time series

目录

摘 要.....	I
Abstract.....	II
1 绪论	
1.1 研究背景.....	1
1.2 研究现状及研究趋势.....	2
1.3 本文的主要工作.....	3
1.4 全文的结构安排.....	4
2. 数据增强	
2.1 数据增强定义.....	6
2.2 数据增强常见方法.....	7
2.3 本章小结.....	9
3 TS-GAN 模型架构	
3.1 TS-GAN 整体结构.....	10
3.2 卷积与反卷积.....	13
3.3 全连接层.....	16
3.4 标准化操作.....	17
3.5 激活函数.....	18
3.6 本章小结.....	19
4 关键技术	
4.1 损失函数设计.....	20

4.2 注意力机制.....	24
4.3 本章小结.....	27
5 实现与测试	
5.1 平台和评价方法.....	28
5.2 数据来源.....	29
5.3 GAN 网络以及分类网络.....	29
5.4 数据增强方法.....	31
5.5 生成数据对比.....	32
5.6 结果评价.....	37
5.7 小结.....	37
6 总结	
6.1 主要工作总结.....	38
6.2 未来工作展望.....	39
参考文献	40
致谢	44

1 绪论

1.1 研究背景

时间序列是指在相同指标条件下,根据数据产生时间的先后顺序排列而成的数列,广泛存在于金融、医药、电力等诸多领域中^[1]。随着互联网技术的发展,其数量规模及复杂程度逐渐增大,此时再通过机器学习技术对时间序列分类效果往往会不理想,因此需要使用能够捕捉复杂特征的深度学习技术。

深度学习是一种基于数据进行表征学习的方法,其作为机器学习领域中最有发展前景的分支,深度学习模型相比机器学习算法更需要大量数据的支持^[2]。目前大量的工作和研究证明,模型的学习能力越好越需要大量的数据进行训练,同样,深度学习算法对时间序列的分类识别也依赖时间序列数据的数量和质量,越多数量越高质量的数据使模型学习时间序列的特征越多越丰富,进而使网络模型分类效果越好^[3]。由于其过分依赖大量的数据,当数据集较小时十分影响深度学习算法模型的性能,容易导致模型陷入过拟合和崩溃状态。在实际项目过程中,数据规模较小往往不足以完成任务,如果要保证完美的完成任务,就需要大量采集真实的数据,但这些数据的采集、标注以及分类会耗费很多时间和人力成本,因此只有大的科研机构和公司才具备拥有大数据的能力。在实际个人研究中,为了尽可能解决数据数量和质量不足和缺陷,充分利用已有的数据进行数据增强显得较为实用。例如在图像领域具有划时代的 AlexNet 模型,其成功背后不仅在于使用了一系列新算法的网络结构,更重点的是其高达 1400 万张的训练集,这是一个包括了 2.2 万个类别的 ImageNet 数据^[4],通过对图像在数量上进行扩充,起到了数据增强的效果。继 AlexNet 之后,接下来出现的一些深层卷积神经网络,如 VGG16^[5]、ResNet^[6],在使用数量庞大、质量较高的图片作为训练集的同时,也会通过一些物理变换来扩大数据集。由此可见,数据集的大小和质量对模型分类识别的提升是多么重要^[7],当数据集规模较难增大时,通过数据增强技术扩充数据集是提升模型识别网络性能的关键技术。将目前应用在分类图像的深度学习的方法稍加改进,如卷积神经网络(CNN)就能很好的对时间序列进行分类^[8],性能好于传统的机器学习方法,但由于数据集大小的限制,并不能展现出深度学

习模型的最大性能，因此有必要使用数据增强技术去扩充数据集，以此来提升分类模型性能。

数据增强对训练神经网络如此重要的原因在于当数据集大小与网络深度不匹配时易造成过拟合现象^[9]。网络的深度一般是指参与计算的各种网络结构层，包括卷积层、激活层和池化层等。越多越复杂地使用这些网络层就意味着网络越深，虽然深层的网络具有更好的性能潜力，但其过拟合现象也会更加严重。

过拟合是指神经网络高度拟合训练数据的分布情况，但在对测试集进行测试时准确率很低，使模型缺乏泛化能力^[10]。虽然有防止过拟合的方法被应用，如正则化、Dropout 等方法，但在数据集较差的时候，这些方法并不能解决模型的过拟合问题，因此缓解过拟合现象的有效方法就是对数据集进行扩充和增加数据质量。

图像领域当中，相应的数据增强方法取得了不错的效果，但是用在时间序列方面效果却不理想，这可能是因为图片在数据增强之后不会改变图片类别，一只猫的照片无论怎样变换也不会变成一只狗，但对时间序列来说人们无法轻易控制这种特殊变换，即使如此仍有人通过移动窗口和窗口压缩等物理方法对数据集进行扩充，但这些方法增强效果有限，并且不具有通用性。

综上所述，在时间序列领域中，在不改变网络模型的情况下，为了提高分类精度减轻过拟合问题，使用合适的数据增强方法来弥补数据不足的问题显得十分必要。

1.2 研究现状及研究趋势

基于深度学习的时间序列分类领域，常见的时间序列数据增强方法基于物理变换和机器学习方法，通过人工设计，针对不同的数据集使用不同的增强策略来获得最佳结果。例如在基于变换的方法中，并不是每一种方法都能有效提高，有时候某些方法结合起来效果更好。

一般来说，时间序列数据增强的方法分为基于变换、基于平均和基于统计等方法。基于变换的方法是在原始时间序列上进行切片，对切片进行物理变换，使用相应小序列训练分类模型，测试时也同样使用小序列进行，通过多数投票法确定最终预测结果。基于平均的方法是对数据集中全部或部分时间序列数据按权重加权平均合成新的数据，一般采用动态时间弯曲（DTW）距离度量下的加权平

均合成新的数据,这有助于更好地维护时间序列的结构特征,其进一步又可以细分为全平均(AA)法、平均选择(AS)法、距离平均选择(ASD)法。基于统计的方法包括马尔可夫链蒙特卡罗拟合统计算法,其他增强方法也是基于机器学习通过数学计算得到扩充数据。在实际应用过程中上述方法扩充效果对数据具有针对性较强适应性较弱的特点,操作繁琐且增强效果往往不理想。

生成式对抗网络 GAN (Generative Adversarial Networks) 在 2014 年由 Goodfellow 提出后^[11],因其开创性的思想被誉为近年来最具想象力的深度学习网络模型,其变体 CGAN^[12]、DCGAN、WGAN^[13]、WGAN-GP 等模型都能通过训练进而生成相同分布的数据。生成对抗网络(GAN)经常被用来生成具有高逼真度的图像,这些图像在原有数据集基础上通过对抗训练提取高维特征组而成,和原始数据集中的图像并不完全相同,实验证明这些生成图像可以作为额外的训练数据来支持其他过程。目前 GAN 模型能够产生类似于训练数据的所需要的高质量样本可以用于数据增强,在以往的工作中,人们提出了不同的 GAN 模型来增强数据。Eyad Elyan 等人通过使用 MFC-GAN 生成多类伪造数据来提高算法分类性能。Giovanni Mariani 等人经过研究,发现 BAGAN 在图像数据扩充方面相对机器学习具有良好的效果^[14]。可以看出 GAN 网络及其衍生网络能够通过原始数据进行数据增强进而提高分类效果^[15]。但目前已有的 GAN 中并没有针对时间序列进行生成的,并且其自身还存在训练不稳定以及模型崩溃的问题,在面对时间序列这种对生成质量高度敏感的数据,更不能满足数据集的要求,其数据增强效果需要验证。即便如此,数据增强方面依然在 GAN 的衍生体上看到了可能性,例如 DCGAN 模型在 GAN 的基础上改进,使用新的模型缓和了 GAN 模型训练不稳定的问题,其优势在于生成数据质量较高,内容多样且稳定。而 WGAN-GP 使用新的损失函数根本上解决了 GAN 模型崩溃和训练不稳定等问题。因此结合各 GAN 衍生体的巨大潜力,设计一种能够针对时间序列进行高质量生成的模型,最终将其应用于数据增强。

1.3 本文的主要工作

针对深度学习中时间序列分类和识别问题,为了最大限度发挥模型的分类效果,需要大量真实数据给网络学习特征,但往往会因数据集规模较小,捕捉到的特征不够明显,导致分类效果较差。本文通过使用 GAN 网络生成时间序列数据

来达到数据增强的目的,缓解数据集规模较小造成的过拟合问题。GAN 的一系列衍生模型相对于 GAN 有较大改进,例如有人在 DCGAN 中加入动态标签等方法能够有效扩充图像数据集^[16]。由于时间序列相对图片其扩充数据质量要求较高,当前出现的 GAN 及其衍生模型在时间序列扩充方面效果一般,不过由于这些模型有着各自的优势和缺陷,因此可以通过对 GAN 模型原理缺陷的深入研究,对该模型进行改进和优化,最终设计出适合时间序列数据集进行数据增强的模型架构。

在吸收各 GAN 衍生模型优点的基础上,本文提出适合时间序列数据增强的 TS-GAN 模型。DCGAN 通过引入卷积神经网络极大的提升了 GAN 训练的稳定性以及生成结果质量^[17]。WGAN-GP 通过 Wasserstein 距离来衡量生成分布和真实分布的距离,并且为满足 lipschitz 连续性条件,重新设计损失函数,加入了限制梯度变化的惩罚,从而在根本上解决了训练不稳定的缺点^[18]。因此结合这些模型特点针对时间序列的数学特性设计出 TS-GAN 模型,来达到扩充数据集的目的。

本文的主要贡献如下:

- 为了提高训练的稳定性,使用 Wasserstein 距离衡量真实数据与生成数据的距离,并将梯度惩罚一同纳入损失函数来满足 lipschitz 连续性条件。
- 为了提高模型生成多样性和准确性,G 网络使用一维反卷积神经网络进行特征生成。
- 为提高 D 网络的特征判定能力,引入注意力机制的思想,结合一维卷积神经网络进行设计。

1.4 全文的结构安排

本文首先给出深度学习中数据增强的定义,说明了数据增强在时间序列生成当中起到的作用,随后将数据增强方法分为机器学习法和深度学习法,并详细介绍其原理,最终引出 GAN 网络模型。接着介绍 GAN 的思想原理,说明其本身在生成数据过程中的缺陷和原因,为了更好的利用 GAN 模型生成高质量的时间序列数据,对网络结构进行修改,提出改造之后的模型 TS-GAN。随后详细介绍 TS-GAN 的网络结构及其原理,并就其所涉及的原理做详细介绍。接着说明 TS-GAN 相对于 GAN 的改进点以及相应的优势,说明其生成高质量数据的可行

性。接着介绍仿真平台、数据来源、所需的分类网络模型以及参数，最后进行实验得出结论。

本文的总体结构安排如下

第一章主要说明研究背景及意义，并分析该方法设计来源以及可行性。

第二章阐述数据增强的定义以及目前所流行的一些技术，为最终引出 TS-GAN 做铺垫。

第三章阐述 TS-GAN 所使用的网络结构及其原理，说明其合理性。

第四章说明 TS-GAN 针对已有 GAN 模型的改进及其涉及的方法。

第五章实验仿真说明了 TS-GAN 在数据增强方面较好的性能, 相对 GAN 网络数据生成的质量提升, 可以有效实现数据增强, 减弱分类模型的过拟合, 提高分类效果。

第六章总结 TS-GAN 数据增强存在的不足以及未来工作的展望

2. 数据增强

不同的数据类型会有不同的增强方式,但绝大部分方法的理念是相通的。本章首先给出数据增强的定义,然后对数据增强方法分类展示,从机器学习和深度学习方法角度介绍现有的增强方法,并介绍几种在时间序列方面效果比较好的例子,最终引出 GAN 网络作为本文研究的重点,说明该网络在数据增强领域的开创性,为下一步引出 TS-GAN 做铺垫。

2.1 数据增强定义

一般来说,对数据进行处理之后能够更好符合现有要求需要的技术都可以称为数据增强技术。在机器学习中,数据增强指对有限的训练集通过某种变换得到更多数据的过程^[19]。在深度学习中,由于数据集是由一种或多种相同属性的训练数据组成,所有又被称为数据集增强。总体来说,数据增强能够利用原有数据,让其产生相当于更多数据价值的一种方法。数据增强方法多种多样,根据样本大小可分为单样本和多样本的数据增强,从增强方式上可以分为生成新的数据和学习增强策略。

单样本数据增强指在数据增强时,围绕单个数据样本进行的操作,例如图片的旋转、切割等操作;多样本数据增强方法指的是利用同一类别下的多个样本,通过人为设计算法来产生新的样本,主要是用一些机器学习的方法,如 SMOTE^[20];数据增强中生成新的数据方法,可以通过模型学习数据的分布,随机生成与训练数据集分布一致的数据,代表方法 GAN;学习增强策略是通过模型任意组合已知的数据增强方法,根据结果反馈找出最佳策略的方法,代表方法 AutoAugment^[21]。

不同的数据类型在数据增强时所使用的方法有所不同。比如文本数据,其数据信息具有离散特性,针对该数据类型,数据增强思路一般是对文本数据中的关键词序进行调整或替换词义等方法;由于时间序列是一维矩阵,数据构成比较简单,可以直接运用一些常见的机器学习方法扩充时间序列数据集的规模,但数据质量不容易保证,常常会导致数据分类精度下降,因此需要在扩充数据规模的同时保证扩充的数据具有较高的质量。

数据增强的本质就是通过扩充数据集来减弱模型的过拟合现象,增强其泛化能力,相比于 dropout^[22]、权重衰减等显式正则化的方法,数据增强既没有改变

网络模型架构，也不增加模型计算复杂度，是一种充分挖掘现有数据特征的隐式方法，在实际应用中更有意义。

2.2 数据增强常见方法

通过对原始数据进行数据增强，能够获得更多数据，但这些数据必须确保正确性和有用性，图片数据经过物理变换不太会增加无关数据，但时间序列数据较为相近，随意变换往往会产生其他类别的数据，因此基于时间序列的数据增强操作一定要把数据的准确性放在首位。

2.2.1 机器学习法

在图像领域，常见的数据增强方法主要是一些简单物理变换，例如缩放变换、裁剪^[23]等。时间序列也有相似的方式进行转换，基于变换的方式就能做到。例如 WW（window wrapping 窗口弯曲）随机在原始数据上选择定长的切片，通过对切片延时间轴按照一定倍数进行伸长或者压缩，然后对经过变换的时间序列再次按定长进行切片，在切片层次上进行训练及预测。如图 2.1 所示，不同的变换方法得到不同的变换结果。

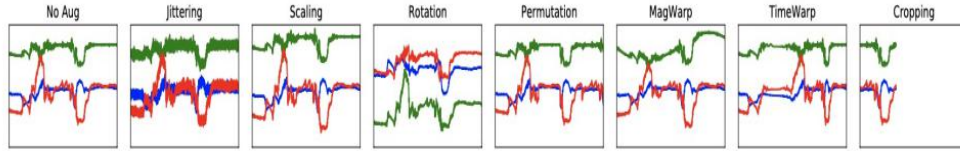


图 2.1 不同变换结果

除了物理变换的方法，还存在一种基于平均的合成数据方法。该类方法在动态时间弯曲(DTW)距离度量下，采用加权平均方法合成新的时间序列数据。公式如 2.1 所示，给定样本空间 S ，带权时间序列数据集：

$$D = \{(T_1, w_1), (T_2, w_2), \dots, (T_N, w_N)\}, \quad (2.1)$$

DTW 下的加权平均时间序列定义为：

$$\operatorname{argmin} \bar{T} \in S \sum_{i=1}^N w_i \cdot \text{DTW}^2(\bar{T}, T_i) \quad (2.2)$$

其中 T_n 为原始时间序列， \bar{T} 为生成的数据，其中 w_n 为权重。权重的计算方式是从数据集中随机选取一个时间序列作为初始平均向量并赋予 α 的权重，然后

随机选取 j 个该时间序列的近邻分别赋予 β 权重, 其余赋予 $(1-\alpha-\beta)/(N-j-1)$ 的权重。

其他机器学习方法中, 能够对时间序列数据进行增强的最具代表性的算法有 mixup 方法和 SamplePairing 方法。

mixup 是采用线性插值得到新样本数据, 随机选取两个数据, 取一介于 0 到 1 的 λ 值, 进行插值计算通过单点叠加得到一个新的样本数据^[24]。该方法因操作简单成为数据增强的常用的方法, 并且其线性插值思想也经常被其他增强方法所借鉴。

SamplePairing 方法原理比较简单, 首先从训练集中随机选取两个数据, 并将两个数据以平均值的形式叠加合成一个新样本, 这两个数据都不需要是同一类别。这在思想上很反常态, 但由于其对称性, 性能提升往往较为可观, 但却解释性不强, 无法深层次理解其原因^[25]。

mixup 和 SamplePairing 在思想上很相似, 通过将各点数据分布视为连续数据来拟合真实样本, 因此导致其所增加的样本点始终处于该类数据围成的区域内, 并不能在给定范围外进行插值。而物理变换等方法产生的数据不确定性比较大, 比较容易生成错误数据。而与其对应的深度学习方法, 不仅能够跳出给定的范围, 还可以通过学习样本的分布状况生成同分布更为真实的样本, 这给数据增强提供了另外一种思路。

2.2.2 深度学习方法

深度学习数据增强法包括两类: GAN 和 AutoAugment, 其中通过 GAN 网络及其变体在多种数据类型上往往会取得比较好的效果。

近些年来生成对抗神经网络被认为是一种能够人工生成逼真数据的工具, 其基本思想是在对抗模式下训练一个生成网络, 这是目前最具创新性的无监督学习方法。GAN 能作为数据增强的关键点是其能够用来拟合数据分布, 就是通过 GAN 产生和这个数据分布相似的一些数据, 正是因为拟合的思想使其比以上所有的方法更具想象力, 但它本身比较难以训练, 难以收敛并且模型很容易崩溃, 甚至会导致生成器欺骗判别器生成错误数据。其优点也很明显, 扩充后的数据集具有相同的分布以及足够多的变量, 且不包含连续重复的数据, 因此在使用该方法时, 尽量避免模式崩溃产生错误数据。虽然 GAN 网络有着严重缺陷, 但其对

抗思想在数据增强方面具有极强的吸引力，现阶段 GAN 及其衍生模型已经融入多种数据当中，只需要根据数据特点修改一些结构，加一些损失函数就能提高任务上的性能。

AutoAugment 的基本思路是在原有增强方法基础上，对数据本身进行多种变换方法，可以是单一变换也可以是多种策略同时使用，然后根据数据增强结果，通过负反馈来学习数据增强的组合策略，这省去了通过人工选择策略的不确定性，大大提高了数据增强的效果。但总体而言还是基于现有的数据增强方法进行组合，只是将人工调参的组合变为学习到的组合，经实验证明在图像增强方面其具有较好的增强效果，但是它远没有 GAN 网络有想象力和灵活。由于 GAN 网络的在生成数据质量的局限性，特别是面对时间序列数据这种数据量本身就少的情况下，效果更不容易提高，很少有人专注于在时间序列上使用 GAN 进行数据增强。因此本文借鉴 GAN 及其衍生模型设计的精妙之处，针对性改造其缺陷，重新设计一种适合扩充时间序列数据的网络结构。

2.3 本章小结

本章首先阐述了数据增强的定义，将数据增强技术分为两大类别，分别从机器学习和深度学习的类别对现有的增强技术进行介绍，在总结其他数据增强方式的局限性的同时引出 GAN 网络模型，在介绍其优点的同时也说明其局限性，为下一章引出 TS-GAN 网络做铺垫。

3 TS-GAN 模型架构

GAN 网络在数据增强时的局限性是不可避免的，在训练过程中，模型往往很容易崩坏，生成错误或者特征不明显的的数据。针对 GAN 模型的不足，设计出适合时间序列数据增强的 TS-GAN，本章就对其基本原理和模型架构进行详细介绍。

3.1 TS-GAN 整体结构

TS-GAN 是在 GAN 的基础上改进而来的，因此在数据增强方面其运行方式和传统 GAN 完全相同。接下来将介绍传统的 GAN 网络训练过程。

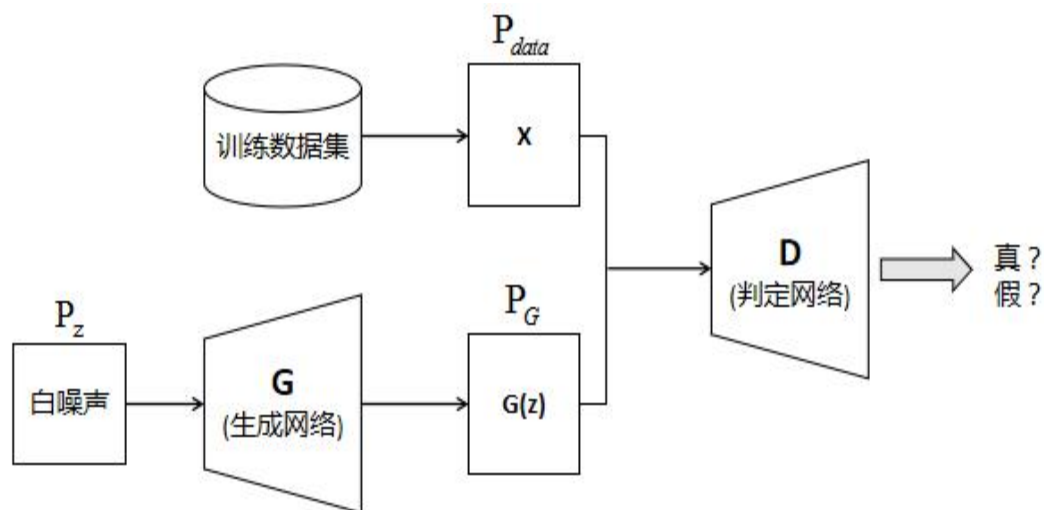


图 3.1 GAN 网络运行过程

如图 3.1 所示，生成对抗网络由生成网络 G 和判定网络 D 两部分组成，生成网络通过输入噪声伪造具有样本分布的数据，判定网络用于区分真实数据和伪造数据。当进行训练时，首先固定 G 网路，并将其伪造的数据和真实数据交给判定网络区分打分，真实数据尽量打 1 分，伪造数据尽量打 0 分，将判定的分数和实际分数生成交叉熵作为损失函数反向传播更新 D 网络，然后固定 D 网络，这次只将 G 网络生成的数据交给判定网络打分，将 D 网络打的分数与真实数据打分的交叉熵当作损失函数，用于更新 G 网络。重复训练多次直至判定网络区分不出伪造数据和真实数据，理想情况下使 $D(G(z)) = D(x) = 0.5$ ，此时得到的 G 生成网络用来扩充数据集。训练过程是 G 和 D 的动态博弈，其中时间序列数据为 P_{data} ， P_z 是输入 G 网络的噪声， $G(z)$ 表示生成器经由噪声得到的生成数据。

由于交叉熵本身具有局限性，其只能描述两个具有重叠分布的距离，当 G 初始化生成的数据与真实分布并没有重叠，交叉熵值为常数，这就使得反向传播更替权重变得没有意义，这是 GAN 生成数据不稳定的主要原因。而且由于网络结构简单，对模型捕捉能力较差，使得 G 很容易生成单一数据甚至是错误数据来欺骗判定网络。综上所述，需要对损失函数和网络结构进行修改来设计出适合时间序列数据扩充的生成对抗网络模型。

3.1.2 G 网络结构

根据前人经验，从时间序列分类模型可以看出，带有卷积层的神经网络更适合处理时间序列分类问题。为了增强模型生成能力，GAN 网络使用的全连接层需要被替换成扩充能力更强的一维卷积神经网络架构，使其生成数据分布更加稳定多样。其网络结构如图 3.2 所示：

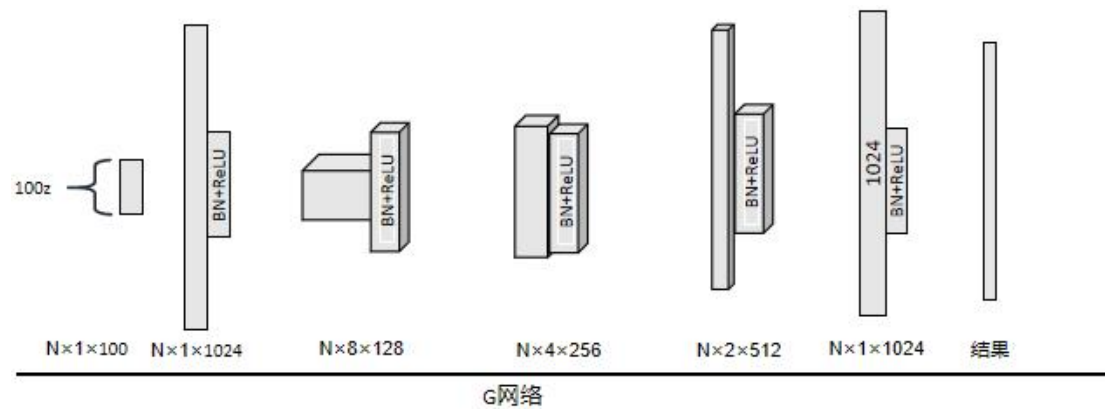


图 3.2 G 网络架构

G 网络输入一个 $[N, 1, 100]$ 向量的随机噪声，首先经过全连接映射为 $[N, 1, 1024]$ 维，经过批标准化和激活函数再改变大小为 $[N, 8, 128]$ ，再经三次反卷积操作，每次使通道数减少一倍，通道维度增加一倍，使生成的特征向量为 $[N, 1, 1024]$ ，由于生成数据长度并不固定，将数据降维，并添加一层全连接层用以调整生成数据的长度。

3.1.2 D 网络结构

D 网络对数据判断的准确与否直接关系到 G 网络生成数据的准确性, 因此为了增强 D 网络的判定能力, 本文通过引入目前时间序列分类模型中效果较好的注意力机制的网络架构^[26], 来提高特征捕捉能力, D 网络具体结构如图 3.3 所示。

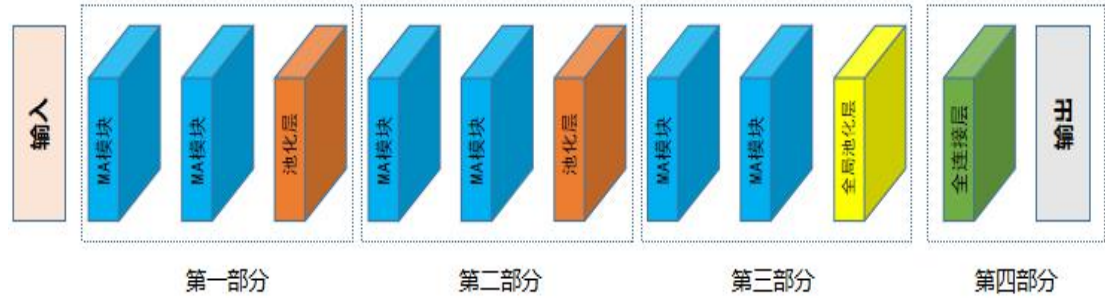


图 3.3 D 网络架构

D 网络结构中共有四部分, 对于前两个部分, 每个部分由三层组成如下, 前两层是 MA 模块, 是整个网络的核心模块。其作用是为了找出由多尺度块生成的每个特征映射重要部分。该部分最后一层为最大池化层, 目的是选择识别能力较强的特征, 减少下一部分输入数据参数数量, 进而延缓过拟合现象并提高模型性能。第三部分除池化层以外, 结构与前面部分相同, 使用了全局平均池化层, 通过计算单通道数据的平均值, 将特征映射与类别进行对应。除此之外, 也能减少模型中的参数数量, 避免过拟合。最后一部分是输出部分, 通过全连接层得到最终的判定结果, 因其更适合综合总体的特征得出判别值而被使用。

D 网络中其池化层是求指定步长范围内的最大值, 通过限制每一步池化尺寸的大小和步长来缩小数据特征。本文中采用池化尺寸为 3, 步长为 2, 填充为 1。因此其输出结果特征向量维度减少一半, 其主要作用是特征映射不改变的前提下, 尽可能减少下一阶段输入数据的规模和整体计算量, 缓解过拟合现象, 进而提高模型泛化能力。

当输入数据向量特征时, 让 N 表示输入的特征向量的数量, L 表示特征向量的长度。其中第一部分的第一个 MA 模块将通道 1 变为 3×64 , 其余维度数值不变, 经过第二个 MA 模块, 数据向量的通道数不变, 最后接池化层将数据维度减半输出结果。数据向量经过第一部分后, 由 $[N, 1, L]$ 变为 $[N, 3 \times 64, L/2]$, 经第二部分时 $[N, 3 \times 64, L/2]$ 变为 $[N, 3 \times 64 \times 2, L/4]$, 当通过第三部分前两层 MA 模块之后, 数据维度由 $[N, 3 \times 64 \times 2, L/4]$ 变为 $[N, 3 \times 64 \times 4, L/4]$, 然后再经全局池化层。

全局池化层是自适应的平均池化函数，取单通道的向量平均值作为该通道的特征，输入向量为 $[N, 3 \times 64 \times 4, L/4]$ ，输出向量为 $[N, 3 \times 64 \times 4, 1]$ ，降维之后接全连接层整合输出维度为 $[N, 1]$ 的向量，作为判定器的判定结果。

3.2 卷积与反卷积

卷积操作^[27]是 TS-GAN 网络中基本的计算单元，本文中 D 网络的每一层是由多个卷积操作进行的，因此为了详细了解 TS-GAN 架构需要熟悉卷积运算的过程。

卷积核中的权重与特征向量相应位置映射逐一相乘并求和，将该值作为单次卷积操作的结果，在进行卷积之前需提前设定好卷积核的大小、步长以及填充方式。由于本文中涉及到的卷积为一维卷积，即卷积核大小为 $1 \times \text{kernel_size}$ 的向量形式，卷积步长即每次卷积之后卷积核延时间轴移动的距离。

$\times W1$ 1.8	$\times W2$ 1.8	$\times W3$ 1.8	1.8	1.7	1.7	1.7	1.7	1.6	1.6	1.5	1.5
1.8	$\times W1$ 1.8	$\times W2$ 1.8	$\times W3$ 1.8	1.7	1.7	1.7	1.7	1.6	1.6	1.5	1.5
• • • • •											
1.8	1.8	1.8	1.8	1.7	1.7	1.7	1.7	1.6	$\times W1$ 1.6	$\times W2$ 1.5	$\times W3$ 1.5

图 3.4 卷积计算过程

如图 3.4 所示，时间序列模型的输入特征向量为 $[1.8, 1.8, 1.8, 1.8, 1.7, 1.7, 1.7, 1.7, 1.6, 1.6, 1.5, 1.5]$ ，设卷积核大小为 1×3 ，卷积步长为 1，其中卷积核的参数为 $[W1, W2, W3]$ 。从特征向量头部开始卷积，第一次卷积得到的结果为 $1.8 \times W1 + 1.8 \times W2 + 1.8 \times W3$ ，卷积核整体移动一个间隔，下一步的结果为 $1.8 \times W1 + 1.8 \times W2 + 1.8 \times W3$ ，依次往下计算，直到最后一个卷积计算完毕。

值得注意的是，当卷积步长不为 1 时，卷积核很有可能移动不到特征向量边缘，而边缘数据往往能提供比较重要的特征，采集不到边缘特征信息会导致该卷积操作丢失特征。为使得卷积核能够完整地提取特征，需要在特征向量外填充一些零，使得卷积核能到达映射特征向量边缘，提取到边缘特征。针对一维数据的

填充是在数据的左右两边对称补指定的零,而填充的具体数值需要根据输入卷积核大小和卷积步长进行调整,使得卷积核刚好达到数据边缘。

卷积之后特征数据的大小会发生变化,设输入的时间序列数据向量大小为 W ,卷积核的大小为 F ,卷积步长为 S ,填充层数为 P ,经过一轮卷积之后的特征大小为:

$$W' = \frac{W - F + 2P}{S} + 1 \quad (3.1)$$

为了方便模型运行,本文让卷积输入尺寸等于卷积输出尺寸,其实现方法即当卷积核中心与特征向量的边缘重合时开始卷积,这使得卷积之后特征向量的大小保持不变。该模型将卷积核的大小 F 设置为 3, 5, 11 等奇数,将填充的大小 $P = (F-1) / 2$,刚好特征向量的边缘部分为卷积核的中间,,这种方式被称为 same 卷积模式。

×W1	×W2	×W3									×W1	×W2	×W3
0	1.8	1.8	1.8	1.8	1.7	1.7	1.7	1.7	1.6	1.6	1.5	1.5	0

图 3.5 same 卷积过程

在单轮卷积操作结束之后,需要在卷积向量所有数据都加上共享偏置项 b ,并经激活函数激活,得到该轮卷积的结果,公式表示为:

$$Cov = \text{conv1d}(W,X,"same") + b \quad (3.2)$$

其中 conv1d 表示为一维卷积,其中已包含激活操作, same 表示为输入尺寸和输出尺寸一样, W 表示卷积核权重, X 表示为特征向量,其均为训练所得,在训练过程中通过反向传播算法不停地调整,以期更好地拟合数据的映射关系。

以上描述的卷积方式仅能检测一种局部特征,但为了获取准确的分类特征需要使用多个卷积核进行卷积操作,这就涉及到卷积中的通道概念,通道可以根据需求定义。如图 3.6 所示,输出通道有 3 个,分别是由 3 个不同的卷积核和特征映射经卷积操作得到。训练完成后,网络可以通过相对应的通道检测出 3 种不同的特征,特征向量中的每个特征都可被检测到。

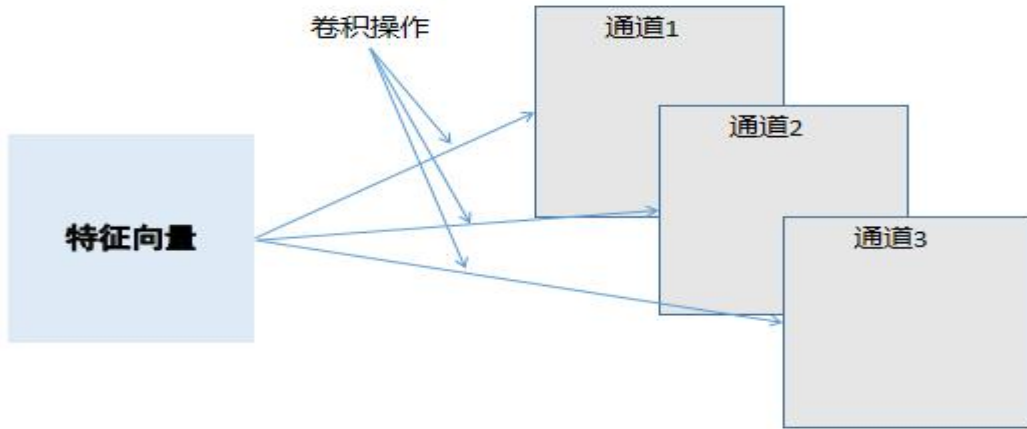


图 3.6 卷积通道

反卷积^[28]操作在 TS-GAN 的 G 网络中较为常见，因此需要介绍一下反卷积运算的过程。

卷积操作本质上是一种特征抽取，数据向量压缩的过程，反卷积则正好相反，是一个数据向量扩充并填补特征的过程。首先根据步长和额外扩充对原向量进行补零操作，再将卷积核中的参数与补零后的数据向量逐位相乘，取累加和作为一次卷积操作的结果。

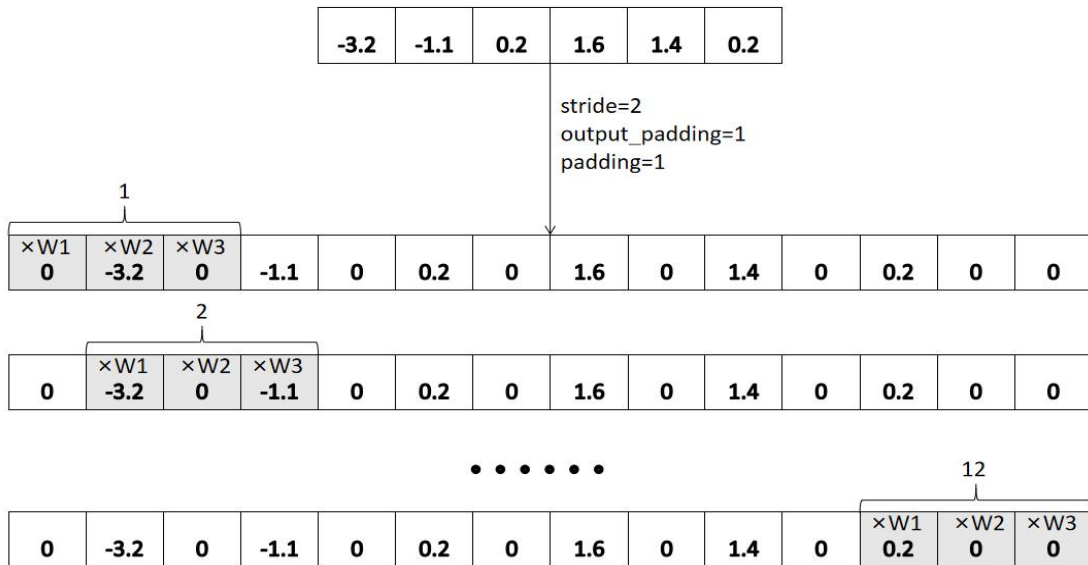


图 3.7 反卷积计算过程

本节仍然以实例为例，随机初始特征向 $[-3.2, -1.1, 0.2, 1.6, 1.4, 0.2]$ ，卷积核为 1×3 ，步长为 2，训练卷积核参数为 $[w_1, w_2, w_3]$ 。步长为 2，因此需要在元素后面补 1 个零（补零数一般为 $\text{stride}-1$ ），padding 为 1，需要在两端补零，由于 output_padding 为 1，则需要在扩充向量的一端补零。因此扩充之后的向量就变为 $[0, -3.2, 0, -1.1, 0, 0.2, 0, 1.6, 0, 1.4, 0, 0.2, 0, 0]$ ，因此第一次卷积结果为 $0 \times w_1 - 3.2 \times w_2 +$

$w3 \times 0$ 。下一次卷积结果为 $-3.2 \times w1 + 0 \times w2 - 1.1 \times w3$, 依次类推。整个过程如图 3.7 所示。

反卷积的目的是为了增大特征数据的向量, 仍然假设初始向量为 W , 步长为 S , 卷积核大小为 K , 扩充为 P , 输出扩充为 OP 。则经过一轮反卷积之后的特征大小为:

$$W' = (W - 1) \times S - 2 \times P + K + OP \quad (3.3)$$

在本文中所有的反卷积使用的卷积核 K 大小为 1×3 , 步长 S 为 2, 扩充 P 为 1, 输出扩充 OP 为 1, 将其带入公式得 $W' = 2 \times W$, 因此每经过反卷积一次将使该通道向量长度增加一倍。

3.3 全连接层

卷积层取得是局部特征, 并没有直接给出特性映射的类别结果, 因此需要全连接层把局部特征重新进行映射组装成完整的类别特征。本文所使用的网络也多次使用到了全连接层, 因此需要熟悉全连接层的运算方式。

全连接层的每个结点都与上一层所有结点相连, 其主要作用是运用该层线性变换的能力, 把前面提取到的特征综合起来, 一般放在多个卷积层后, 用于整合卷积层或池化层对该类别区分性的局部信息的特征映射, 其网络运算过程如图 3.8 所示。

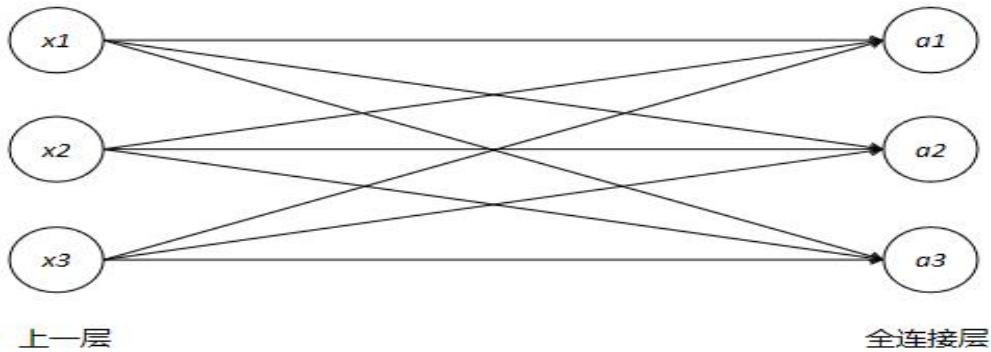


图 3.8 全链接层

其中, $x1, x2, x3$ 为全连接层的输入, $a1, a2, a3$ 为输出, W 分别是各个连线的权重。

$$a1 = W_{11} * x_1 + W_{12} * x_2 + W_{13} * x_3 + b_1 \quad (3.4)$$

$$a2 = W_{21} * x_1 + W_{22} * x_2 + W_{23} * x_3 + b_2 \quad (3.5)$$

$$a_3 = W_{31} * x_1 + W_{32} * x_2 + W_{33} * x_3 + b_3 \quad (3.6)$$

此外全连接层还用来捕捉特征向量特征，因此可以叠加多层作为分类模型，比如 MLP 模型，在时间序列特征不复杂的条件下具有较好的分类效果。

3.4 标准化操作

卷积层有助于模型能够获得更好的特征捕捉能力，但是在训练过程中，任意神经网络中的任意神经元输出 x_{na} （线性变化后未使用激活函数）的分布一直发生变动，此时若整体分布比较靠近激活函数的上下限时（以 sigmoid 函数为例，当输出值 x_{na} 超过 5 或者小于 -5，此时经过激活将分别达到 sigmoid 的上限和下限），并且在进行反向传播时，层次靠后的神经网络会出现梯度消失，参数更替缓慢就会导致深层网络越来越难以收敛。因此为了加速收敛，需要在神经元线性变换输出值 x_{na} 在通过激活函数之前，预先采用标准化操作，而本实验使用了两种标准化操作，G 网络使用了批标准化，D 网络使用了实例标准化。

批标准化的思想是将指定输出通道的数据向量的分布，强制拉回均值为 0，标准差为 1 的标准正态分布，避免了经激活函数激活后数据的上下限接近饱和区，使批标准化后的数据分布处于激活函数的输入敏感区，从而避免了梯度消失问题^[29]。

本文采用 mini-batch 训练方法，一次训练中只随机的包含 batch 个样本（由于训练过程引入了随机性，能够提升模型的训练速度）。对于隐藏层的任意神经元来说，批处理化操作 x_{na} 将进行如下变换：

$$\hat{x}_{na} = \frac{x_n - E[x_{na}]}{\sqrt{Var[x_{na}]}} \quad (3.7)$$

$E[x_{na}]$ 和 $Var[x_{na}]$ 分别表示单次训练的多个样本的均值和方差。此时的特征映射向量已经变换为标准正态分布，这样就可以避免进入激活函数的上下限，并一定程度上缓解梯度消失，加快模型收敛。但这会降低整个网络的对数据的表达能力，为了减少其影响，需要添加 γ 及 β 两个调节参数，其表达式如公式 3.8 所示，其中 γ 及 β 两个都是通过模型训练得出的。

$$x_{nm} = \gamma \hat{x}_{na} + \beta \quad (3.8)$$

由于本文使用的梯度惩罚是对每一 batch 中的每一个样本都做了梯度惩罚，引入批归一化会引起同一批样本之间的依赖关系，这会导致模型不收敛，因此 D 网络中不能使用批标准化，选择实例归一化。实例归一化指对一个批次中单个数据进行归一化，任意神经网络中的任意神经元输出 x_{na} 的变换公式和公式 3.7 相同，但此时的 $E[x_{na}]$ 和 $\text{Var}[x_{na}]$ 分别表示指单个样本的均值和方差。

3.5 激活函数

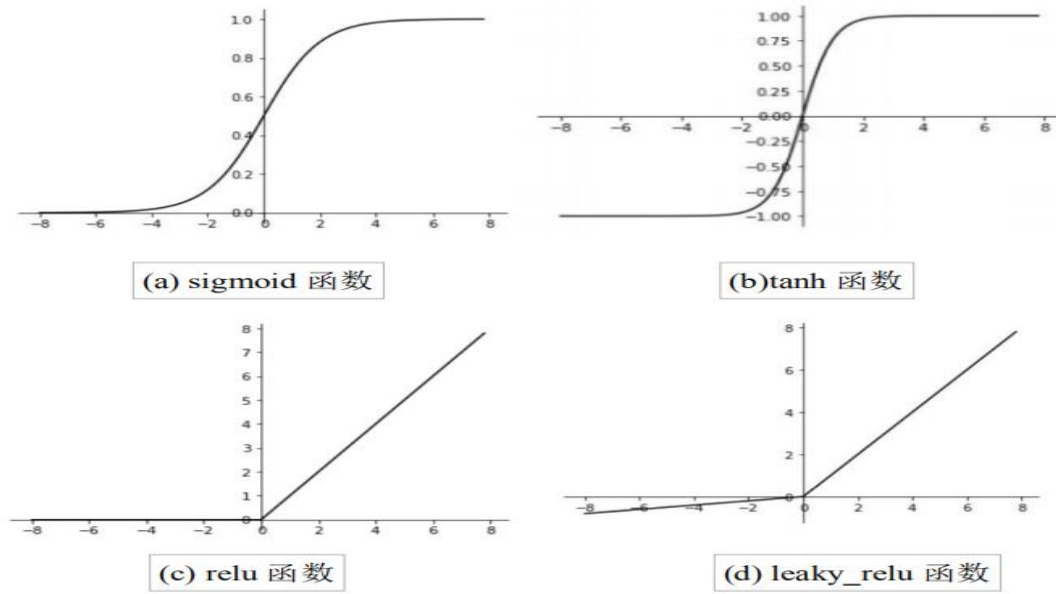


图 3.9 激活函数

在输出特征映射之前，经标准化处理后还需激活函数激活操作，通过非线性变换来增强深度神经网络对数据向量的表示能力。模型当中无论是卷积层还是全连接层都需要激活函数激活来提高其非线性表达能力，否则模型只能建立线性关系，极大的浪费深度网络的表达能力。

在实际操作中，激活函数有 Sigmoid 函数^[30]、Tanh 函数、Relu 函数^[31]和 LeakyRelu 函数等，其公式表示如下，其中 z 是为未经激活的线性输出。

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (3.9)$$

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (3.10)$$

$$\text{relu}(z) = \max(0, z) \quad (3.11)$$

$$f(z) = \begin{cases} z & z > 0 \\ \alpha z & z \leq 0 \end{cases} \quad (3.12)$$

不同的模型有不同的激活函数选择，以上提及的激活函数在本文都有使用。例如 Sigmoid 用在 FCN 当中，放在最后一层激活值作为预测输出，Tanh 用在 GAN 中的 G 网路平衡正负区间的数值分布，其 D 网络为了对负区间产生梯度使用了 LeakyRelu 激活函数，Relu 作为一般的激活函数被大量使用在 TS-GAN 中。

3.6 本章小结

本章详细介绍了 TS-GAN 模型的网络结构，生成网络基于改进的 DCGAN 鉴于其较高的稳定性，因此抛弃掉简单的全连接层，选用一维反卷积神经网络。判定层网络采用注意力机制，因其具有较好的特征捕捉能力而被采用。由于这些都是针对时间序列这类数据进行改进的，具有针对性和特殊性，因此本文将此模型称为 TS-GAN（时间序列生成网络）。

4 关键技术

GAN 及其衍生模型在图像生成上非常成功,并逐步向其他领域进军以期获得同样较好的效果,但由于网络设计的局限性导致其没有表现出较好的网络泛性,都需要较大的改动才能有好的效果,所以需要改进网络结构或损失函数等。本章介绍 TS-GAN 的改进及其涉及的关键技术。

4.1 损失函数设计

在 GAN 中使用交叉熵作为损失函数。 JS 散度作为损失函数,用来衡量随机噪声生成数据的概率分布与真实数据的概率分布的信息损耗^[32]。公式如下所示:

$$JS(P \parallel Q) = \frac{1}{2} KL(P \parallel \frac{P+Q}{2}) + \frac{1}{2} KL(Q \parallel \frac{P+Q}{2}) \quad (4.1)$$

$$KL(P \parallel Q) = \sum p(x) \log \frac{p(x)}{q(x)} \quad (4.2)$$

将公式 4.2 带入公式 4.1 并展开

$$JS(P \parallel Q) = \frac{1}{2} \sum p(x) \log(\frac{p(x)}{p(x)+q(x)}) + \frac{1}{2} \sum q(x) \log(\frac{q(x)}{p(x)+q(x)}) + \log 2 \quad (4.3)$$

当 $p(x)$ 和 $q(x)$ 不重叠时,即 $\frac{q(x)}{p(x)+q(x)} = 1$, 此时 $JS(P \parallel Q) = \log 2$, 因此当

初始化生成数据的概率分布没有和真实数据重合时,损失函数值没有意义,便会
导致梯度消失,这是导致 GAN 模型不稳定的原因之一。由于模型设计的不合理,
常常导致模型每次训练得到的结果不一样。并且当训练次数达到一定数值时,其
训练效果越来越差劲,反映在损失函数上,表现为震荡不收敛。如图 4.1 所示,
当训练次数超过 40000 轮,其损失值有震荡不收敛现象。

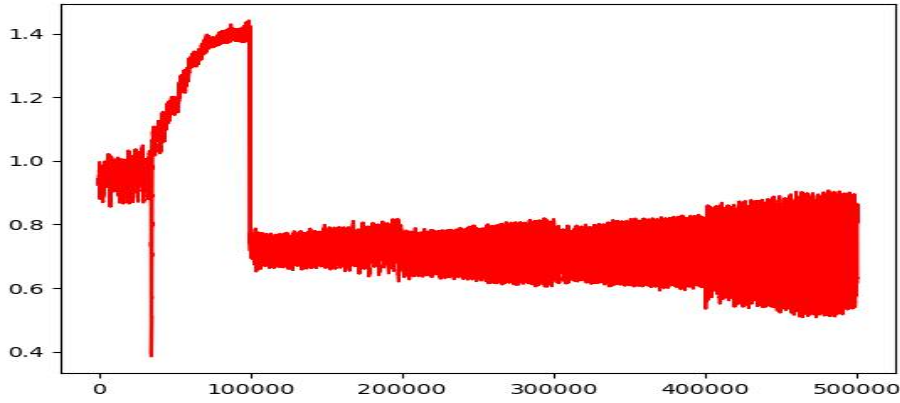


图 4.1 损失函数值

鉴于 GAN 网络损失函数的不合理性，参考 WGAN-GP 模型，拥有较快的收敛速度，几乎不需要调参，得益于其优秀的损失函数就能生成高质量的样本。因此本模型选择使用 Wasserstein 距离衡量生成分布和真实分布的距离^[33]。其公式表达如下：

$$W[p, q] = \max_{D, \|D\|_L \leq 1} E_{x \sim p(x)}[D(x)] - E_{x \sim q(z)}[f(G(z))] \quad (4.4)$$

其中 $\|D\|_L \leq 1$ 称为 Lipschitz 约束^[34]，为维持约束需要将梯度惩罚加入损失函数。在实现上，在训练时，取每一批次中的 D 网络权重变化之和，取其平方值加入损失当中，这样就可以将梯度惩罚在 1 附近浮动。整体的损失函数如下：

$$L = E_{z \sim P_z}[D(G(z))] - E_{x \sim P_{data}}[D(x)] + \lambda E_{x \sim \chi}[\|\nabla D(x_{inter})\|_p - 1]^2 \quad (4.5)$$

其中 x_{inter} 采用随机插值的方式：

$$\begin{aligned} x_{inter} &= \varepsilon x_{real} + (1 - \varepsilon) x_{fake} \\ \varepsilon &\sim U[0, 1], \quad x_{real} \sim P_r, \quad x_{fake} \sim P_g \end{aligned} \quad (4.6)$$

P_r 为真实数据， P_g 为生成数据。这种方案在很多情况下工作的比较好，但是当真实样本的类别数比较多时却比较差，这是因为多类别随机插值的空间很有可能不在真实样本与生成样本之间，导致权重更新混乱判别器失灵，因此本文使用单类别训练来避免这种情况发生。

判定器试图确定样本是否真实，继续调整 G 和 D，直到 D 不能区分真实数据和生成训练的数据，得到 $P_G = P_{data}$ 的最优性。G 试图欺骗 D，而 D 尽力去区分真实样本和伪造样本。

固定 D 网络，更新 G 网络时，其目标为：

$$\min_G E_{z \sim P_z}[D(G(z))] \quad (4.7)$$

固定 G 网络，更新 D 网络时，其目标为：

$$\max_D E_{x \sim P_{data}}[D(x)] - E_{z \sim P_z}[D(G(z))] - \lambda E_{x \sim \mathcal{X}}[\|\nabla D(x_{inter})\|_p - 1]^2 \quad (4.8)$$

在参数更新之前，模型首先是经过前向传播，无论是更新 D 网络还是更新 G 网络都需要通过 D 网络得到一个 $N \times 1$ 的向量数值，在计算损失函数之后，将损失函数值作为误差值反向传播计算梯度大小，更新权重 W 和偏置项 b。

$$L_1 = \frac{1}{N} \sum_{k=0}^N y_k^{(i)} \quad (4.9)$$

其中 $y_k^{(i)}$ 作为当输入 $h_i^{(L-1)}$ 时，第 k 个神经元的输出， $y_i = f(\sum_{j=1}^n w_{ij} h_j^{(L-1)} + b)$ ，

w_{ij} 即为前一层第 i 个神经元与当前层第 j 个神经元之间的连接权重，f 为激活函数， x_i 为该神经元的输入。

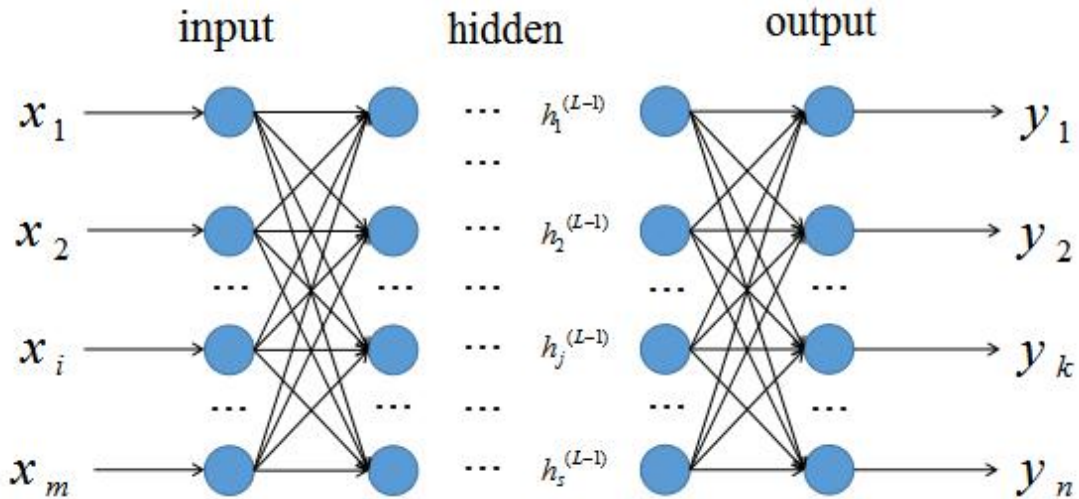


图 4.2 参数更新

$h_i^{(l)}$ ：第 l 层第 i 个神经元的输出，并且 $h_i^{(l)} = f(ne_i^{(l)})$

由于梯度爆炸的原因，需要加一个限制梯度爆炸的损失函数。

$$L_2 = \lambda(\|\nabla_{\hat{x}_i} D(\hat{x}_i)\|_2 - 1)^2 \quad (4.10)$$

其中 $\nabla_{\hat{x}_i} D(\hat{x}_i)$ 是 D 网络在输入 \hat{x}_i 时的梯度, 对于单个样本 x_k , 本节以更新 G 网络当中的 $w_{kj}^{(L)}$ 权重为例, 以输出层(即第 L 层)第 k 个神经元与第 L-1 层第 j 个神经元权值, 损失函数 L_1 对 w_{kj} 求偏导可得:

$$\frac{\partial L_1(j)}{\partial w_{kj}^{(L)}} = \frac{\partial L_1(j)}{\partial y_k(j)} \frac{\partial y_k(j)}{\partial net_k^{(L)}} \frac{\partial net_k^{(L)}}{\partial w_{kj}^{(L)}} \quad (4.11)$$

$$\frac{\partial L_1(j)}{\partial y_k(j)} = \frac{1}{N} \quad (4.12)$$

$$\frac{\partial y_k(j)}{\partial net_k^{(L)}} = f'(x) \big|_{x=net_k^{(L)}} \quad (4.13)$$

$$\frac{\partial net_k^{(L)}}{\partial w_{kj}^{(L)}} = h_j^{(L-1)} \quad (4.14)$$

结合公式(4.11), (4.12), (4.13), (4.14)可得:

$$\frac{\partial L_1(j)}{\partial w_{kj}^{(L)}} = \frac{1}{N} f'(x) \big|_{x=net_k^{(L)}} h_j^{(L-1)} \quad (4.15)$$

同理可得:

$$\frac{\partial L_1(j)}{\partial b_k^{(L)}} = \frac{1}{N} f'(x) \big|_{x=net_k^{(L)}} \quad (4.16)$$

对隐含层 L-1 层:

$$\frac{\partial L_1}{\partial w_{ji}^{(L-1)}} = \sum_{k=1}^n \frac{\partial L_1}{\partial y_k(j)} \frac{\partial y_k(j)}{\partial net_k^{(L)}} \frac{\partial net_k^{(L)}}{\partial w_{ji}^{(L-1)}} \quad (4.17)$$

而

$$net_k^{(L)} = \sum_{p=1}^{S_{L-1}} w_{kp}^L h_p^{(L-1)} + b_k^{(L-1)} \quad (4.18)$$

因此

$$\frac{\partial net_k^{(L)}}{\partial w_{ji}^{(L-1)}} = w_{kj}^L f'(x) \big|_{x=net_k^{(L)}} h_i^{(L-2)} \quad (4.19)$$

结合(4.16), (4.17), (4.18), (4.19)可得

$$\frac{\partial L_1}{\partial w_{ji}^{(L-1)}} = \sum_{k=1}^n \frac{1}{N} f(x)' \big|_{x=ne t_k^{(L)}} w_{kj}^{(L)} f(x)' \big|_{x=ne t_j^{(L-1)}} \quad (4.20)$$

$$\text{另 } \delta_k^{(L)} = \frac{1}{N} f(x)' \big|_{x=ne t_k^{(L)}} \text{ 和 } \delta_j^{(L-1)} = \sum_{k=1}^n \delta_k^{(L)} w_{kj}^{(L)} f(x)' \big|_{x=ne t_j^{(L-1)}} \quad (4.21)$$

$$\frac{\partial L_1}{\partial w_{ji}^{(L-1)}} = \delta_j^{(L-1)} h_i^{(L-2)}, \quad \frac{\partial L_1}{\partial b_i^{(L-1)}} = \delta_j^{(L-1)} \quad (4.22)$$

依次类推第 l 层的权重和偏导为:

$$\begin{aligned} \frac{\partial L_1}{\partial w_{ji}^{(l)}} &= \delta_j^{(l)} h_i^{(l-1)} \\ \frac{\partial L_1}{\partial b_i^{(l)}} &= \delta_j^{(l)} \end{aligned} \quad (4.23)$$

其中,

$$\delta_j^{(l)} = \sum_{k=1}^{s_{h1}} w_{kj}^{(l+1)} \delta_k^{(l+1)} f(x)' \big|_{x=ne t_j^{(l)}} \quad (4.24)$$

在进行反向传播时可以通过分步更新权值, 第 l 层的权重和偏导见公式 4.23。

另 $\Delta w_1^{(l)}$ 和 $\Delta b_1^{(l)}$ 代表第 l 层的真实数据的权重梯度和偏置梯度, 将 $\Delta w_2^{(l)}$ 和 $\Delta b_2^{(l)}$ 代表第 l 层的生成数据的权重梯度和偏置梯度。因此 G 网络梯度改变为

$$w^{(l)} \leftarrow w^{(l)} - \frac{\alpha}{N} \Delta w_2^{(l)} + \frac{\alpha}{N} \Delta w_1^{(l)} \quad b^{(l)} \leftarrow b^{(l)} - \frac{\alpha}{N} \Delta b_2^{(l)} + \frac{\alpha}{N} \Delta b_1^{(l)} \quad (4.25)$$

同样 D 网络的权重的更新只需将 L_2 损失添加到损失函数当中, 权重更新和上述方法相同。

4.2 注意力机制

生成式对抗网络的本质是 D 网络和 G 网络的博弈, 其中 G 的生成能力依靠 D 网络的判定能力, D 网络对数据真假的区分能力越强, G 网络随机生成的数据分布与真实数据的分布越接近。但考虑到模型崩溃和梯度消失的问题, D 网络必须泛化能力较强且层数不能很深, 因此就需要在每一层上尽可能多的提取特征。在时间序列分类时引入 CNN 网络极大地提高了分类器的性能, 其中注意力机制网络架构性能更好并且满足以上要求, 因此选择引入注意力机制的一维卷积模型作为判定网络。

注意力机制是分配可用处理资源，使之偏向于输入数据的最重要的特征。其广泛应用于自然语言处理和计算机视觉中，主要是通过关注重要特征和抑制不必要的特征来增强向量特征代表性。

多注意力机制分类是在 CNN 技术的基础上提出的一种新的分类模型，在图像方面的应用，主要依赖于具有区分度的局部定位和基于局部的精细特征学习，将两者联系起来，让局部定位和局部的精细化特征学习能相互强化^[35]。模型提出了两个损失函数，一个是让不同通道中峰值相近的通道聚在一块的损失函数，另一个是一般分类网络模型中的交叉熵。该网络架构由通道分组层和卷积层组成，在进行训练时首先固定卷积层参数，使用通道损失函数来优化通道分组层寻找部件，然后固定通道分组层，通过分类损失函数去优化卷积层和 Softmax 层来学习细粒度特征，通过交替训练，直到两个损失不在发生变化。鉴于在图像方面的良好分类的效果，将其思想应用于时间序列分类上，根据其数据特点设计的网络架构也取得了良好的效果。

注意模型有很多方式，其中较为典型的有空间变换网络（STN）、SE 网络（Squeeze-and-Excitation Networks）和卷积块注意模块（CBAM）。STN 通过对每个输入样本进行适当的转换来提供空间变换能力，变换操作可以包括缩放、旋转等^[36]，通过在整个特征映射上执行，使得网络不仅可以选取数据当中最相关的区域，还可以将这些区域转换为规范的、期望的数据类型，以简化接下来卷积层的识别。因此，STN 被视为空间注意力。SE 块侧重信道注意，能够让整个网络在全局信息基础上选择性的关注重要的特征，并抑制不太重要的特征映射，起到特征重新校准作用^[37]。SE 块增强了早期层中特征向量映射的低水平表示，并在后面各层的响应中，在面对不同输入时，捕捉特征变得越来越专一化。由于数据中每个特征映射的权重分别在各自层自动学习，就可以通过整个网络积累起来，增强特征鉴别能力。CBAM 依次通过信道和空间注意模块，注重沿通道和空间轴有意义的特征^[38]。由于时间序列数据没有空间轴，因此需要在信道维度上进行注意机制，以提高分类精度。

本文所介绍的 TS-GAN 模型中，其 D 网络通过叠加 MA 模块来实现多尺度注意机制，根据特征映射的重要性自动学习来增强有用的特征映射，并抑制不太有用特征映射的策略。

MA 模块分为多尺度块和注意力块，如图 4.3 所示

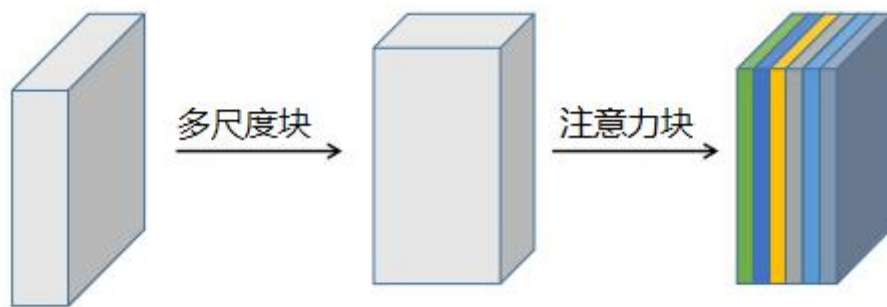


图 4.3 MA 模块架构

MA 模块的目标是提高网络的判定能力，为实现这一目标，需要使用多尺度时间序列的短、中和长期依赖关系，通过使用多尺度卷积来获得多尺度时间信息，增加特征映射，这是多尺度块的作用。然后注意力块选择性地关注重要特征，通过重新计算每个特征映射的权重来忽略不重要的信息。

其中多尺度块结构如图 4.4 所示

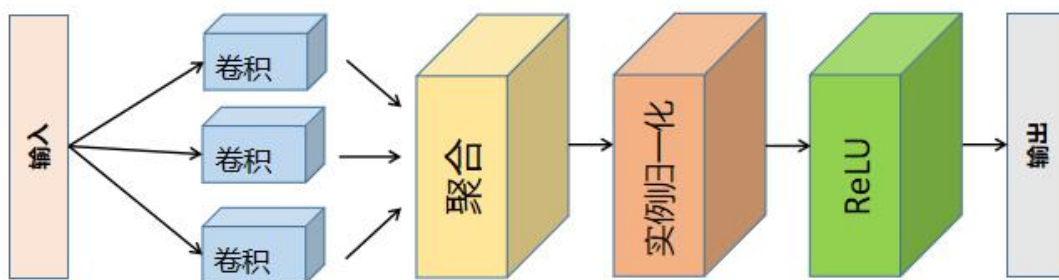


图 4.4 多尺度模块架构

其中卷积阶段具体做法是使用三种不同的卷积方式，步长为 1，卷积核长度分别为 3，7，11，填充为 1，3，5，同时通过设置通道的参数来规范特征向量。其目的是为了获取特征向量在不同视野下的特征表现，多尺度的获取特征，这将有利于增强模型的判定能力。聚合阶段是将三个不同的特征向量叠加在一起，即使得通道数增加三倍，可以捕捉获取更多的特征。

注意力块结构如图 4.5 所示

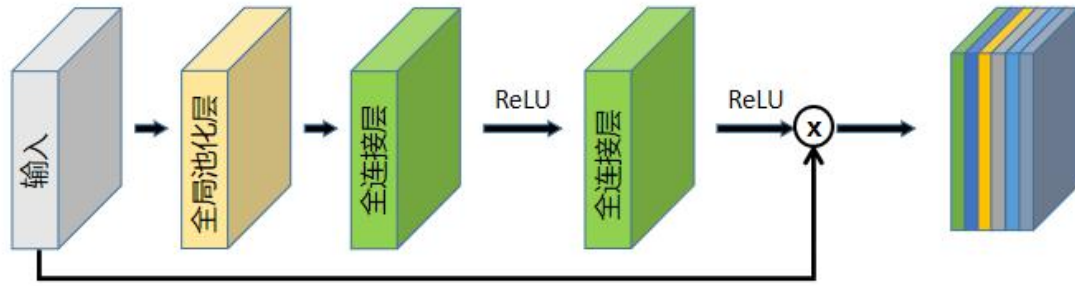


图 4.5 注意力模块架构

第一层是全局池化层，计算出所有通道的平均值，将全局时间序列信息压缩到单一信道内，生成该信道内的代表数据，其他两层是全连接层，经过第一层减少了特征映射通道，经过第二层时将其再次放大至原始大小，能够重新校准信道中的目标数据。最后将原输入数据与所得的矩阵通道信息相乘，输出结果通道特征明显的被加强，通道特征较弱的被抑制。本模型采用 16 作为特征通道的缩小倍数，每一次经过注意力块其数据维度并不变化。

多尺度注意模块由多尺度块和注意块组成，多尺度块是由通道维度上的不同尺度卷积操作构建而成，它产生不同大小的接收场来捕获不同尺度的信息。注意模块通过压缩和计算，引入信道权重来关注重要的特征映射。由于省去预训练，其在特征向量或数据预处理中没有繁重的加工。事实也证明，引入多注意力机制的分类器在时间序列分类上要优于其他 CNN 模型，因此需要选择将其引入增加模型特征判定能力，更好的监督生成网络。

4.3 本章小结

本章首先对 GAN 网络训练过程不稳定进行了公式推导，说明了其损失函数不稳定的原因，其次通过实验当中损失函数曲线的波动程度来说明其生成数据的不稳定性。接着针对其不稳定性针对性地提出改进方法，依次详细介绍损失函数机器参数、引入注意力机制构架以及其他方面的优化，通过这些方面改进设计出适合时间序列生成的 TS-GAN 网络。

5 实现与测试

5.1 平台和评价方法

在硬件环境方面，本实验室使用了 Intel(R) core I5-4590X CPU(3.30GHz), 8GB RAM 和 Window10 操作系统。实验模拟使用 Pytorch 框架，Python 作为编程语言。

在本文中，由于测试数据集数据类别大部分存在类别不平衡现象，因此使用混淆矩阵来度量结果，用于克服数据本身的缺陷。混淆矩阵通过统计单个类别数值来表示精度评价，是一种标准评价格式，该矩阵具有 n 行 n 列，分别统计分类模型分类正确和分类错误的观测值个数，把结果放在一个表里展示出来，用来描述已知类别的测试集数据在分类模型上的性能。

混淆矩阵中使用的基本术语如下：

- 真阳性 (TP)：实际为阳性，预测为阳性。
- 假阴性 (FN)：观察为阳性，但预判为阴性。
- 真阴性 (TN)：观察结果为阴性，并预先指定为阴性。
- 假阳性 (FP)：观察结果为阴性，但预判为阳性。

本章通过测量准确性(Accuracy)、精确性(Precision)、召回率(Recall)和 F1 分数(F1- Score)来评估测试性能。具体计算方法如方程式所示：

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.1)$$

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (5.3)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5.4)$$

准确度是衡量一个模型性能的最直观的评价指标。但当数据集规模不平衡时，需要补充一些其他指标。由于精确率和召回率通常是此消彼长，很难兼得，

因此综合考虑就需要精度的调和平均值。当数据不平衡时，F1 分数可以准确地评价模型的性能。

5.2 数据来源

UCR 是时间序列分类中常用的数据集，数据来源丰富且多样，涉及各方面类型的数据，因此本实验使用 UCR 文件中的一部分数据集来测试本文提出的方法，由于大多数研究人员都是采用该数据集评估时间序列分类模型的性能，并且大部分数据集存在训练集数据较少的问题，这有可能导致模型在学习的时候过拟合严重，影响分类效果。因此本章将使用 TS-GAN 模型对这些数据集进行数据增强，并通过原始数据以及 GAN 网络扩充之后的数据作为对照，来说明 TS-GAN 能够扩充高质量同分布的数据。

本实验在 UCR 中选择 Beef、CBF、ECGFiveDays、Gun_Point、DiatomSizeR、FaceFour 这些长度相对较短、分类精度有待提升的数据集，分别对其数据集中的训练集进行数据增强。

5.3 GAN 网络以及分类网络

5.3.1 GAN 模型

前人研究表明，GAN 网络对数据的增强能力要高于机器学习方法，例如 JooHwa Lee 等人使用 GAN 扩充入侵系统的不平衡数据，通过对少数类的扩充，使总体的分类性能得到提高，其增强效果好于 SMOTE^[39]。为了突出 TS-GAN 在时间序列数据方面具有良好扩充效果，就需要一个传统的生成对抗网络模型作为对照。因此本文根据其原理设计出适合时间序列的生成对抗网络，并在同样的条件下扩充时间序列以验证 TS-GAN 相对较好的效果。

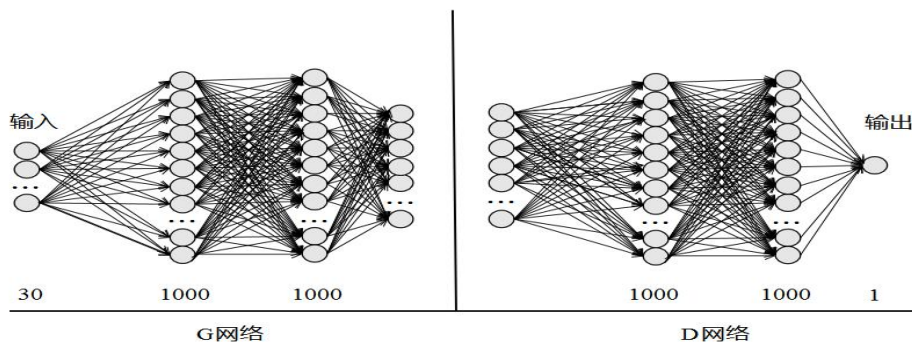


图 5.1 GAN 网络架构

由于时间序列是一维数据，一般使用最朴素的 GAN 网络就能模拟出特征较

明显的数。如图 5.1 所示，本章所使用的 G 网络由三层全连接层组成，每层神经元为 1000 个，前两层使用 Relu 激活函数，最后一层使用 Tanh 激活函数生成具有负区间的数据。D 网络也同样适用相同的全连接层，但为了照顾负区间的数据并使其也具有梯度学习，前两层使用 LeakyReLU 作为激活函数，最后一层接 Sigmoid 层输出 0 到 1 之间的判定值，使用传统的交叉熵作为损失函数，梯度下降算法使用 Adam 进行参数的更替。

虽然此网络结构简单，但模型捕捉和生成能力效果还不错，在 MINST 数据集上具有良好的生成效果。引入此网络的目的是为了验证 TS-GAN 相对于原始的 GAN 是否具有较好的特征捕捉能力以及更好的抗模型崩溃和抗过拟合的能力。

5.3.2 FCN 网络

时间序列的分类目前所用的方法分为传统方法和基于特征的方法，由于传统方法需要手工设计，操作复杂且分类效果又差，因此本文采用基于特征且需要大量数据才能更好工作的方法，这些方法中包括 MLP^[40]、FCN^[41]、ResNet^[42] 模型，MLP 没有捕捉数据局部特征的能力，对扩充数据质量识别度不高，固不采用。ResNet 分类效果虽然稍微优于 FCN，但由于其模型复杂且训练难度较大，综合考虑之后选择 FCN 作为本章用于比较数据分类效果的模型。

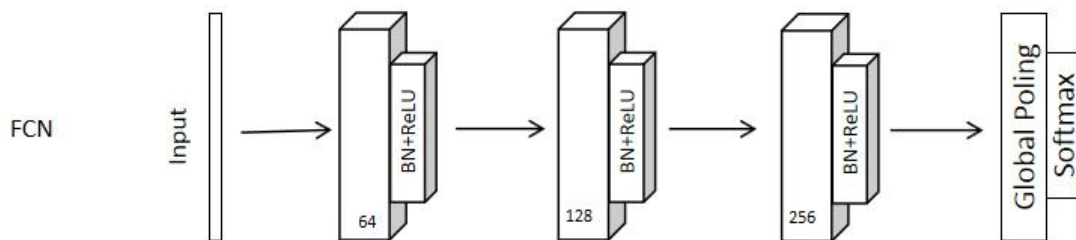


图 5.2 全卷积神经网络架构

FCN 模型架构如图 5.2 所示。FCN 网络又叫全卷积神经网络，该模型共使用三层卷积层，第一层卷积使用 64 个 1×3 的卷积核，步长为 1，第二层卷积使用 128 个 1×7 的卷积核，步长为 3，第三层使用 256 个 1×11 的卷积核，步长设置为 5。每个卷积块都需批归一化，经由 ReLU 函数激活。最后经全局池化层连接 Softmax 层进行分类。本文设输出数据为 $N \times 1 \times L$ ，经过三层卷积矩阵大小依次变为 $N \times 64 \times L$ ， $N \times 128 \times L$ ， $N \times 256 \times L$ ，经全局池化变为 $N \times 256$ ，后接 Softmax 层

输出分类结果,通过使用交叉熵作为损失函数反向传播来更新权重。由于每一层的输入输出数据长度都为原数据向量大小,并不需要反卷积层扩至原始大小,固被舍去。

FCN 具有语义级别的数据特征分割能力,因此可以用于检测扩充数据集在特征信息方面是否丢失细节^[43]。同时 FCN 相比于机器学习等方法,具有不限制输入格式、使用方便和分类效果好等优点,相对于其他深度学习方法,具有结构简单、调参少、运行速度快等优点。FCN 虽然结构简单,但却能够学习和存储大量输入输出的映射关系,并具有一定的泛化能力,当输入样本带有较大误差甚至个别错误时对网络的影响较小。卷积操作具有局部感知以及多核和参数共享能力,在分类性能上要强于机器学习等的方法,但相对而言需要大量样本才能有好的效果,因此当使用数据增强扩充大量数据时,其分类性能较容易提高。

综上所述,本实验选用 FCN 时间序列分类模型来验证生成对抗网络数据增强之后的数据集,观察生成数据在卷积操作之后是否还具有明显特征,进而提高模型分类精度。依靠模型分类效果,便能判断 TS-GAN 模型相比 GAN 模型在数据集扩充的提升效果,以此说明其较优异的性能。

5.4 数据增强方法

所提出的基于 TS-GAN 和 GAN 的数据增强是在规范化后的数据集中,挑选因数据集中训练集规模较小而导致分类效果较差的数据集,使用 TS-GAN 和 GAN 分别对训练集进行数据增强,通过对数据增强之后的训练集进行学习,建立分类模型,并使用测试数据集验证模型的分类效果。本方法一般是通过模型扩充数据集中的训练集来补充学习模型的特征学习能力进而提高分类能力,具体方法如图 5.3 所示。

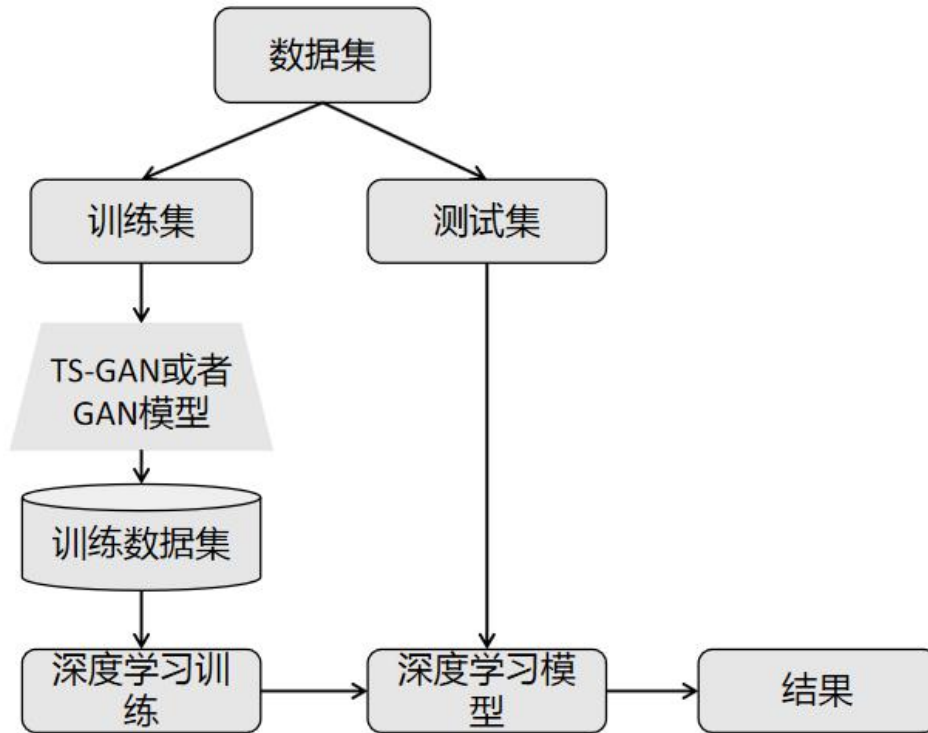


图 5.3 数据增强操作

本实验使用 Pytorch 库进行模型搭建，通过人工设置模型参数，包括下降算法、学习率、卷积核大小、卷积核数目、批处理大小等。

TS-GAN 采用 min-batch 训练方法进行模型的训练，每次使用一个 batch 的样本参与训练，受到处理速度和文件大小的影响，批处理一般设置为小于最小类别的 2 的倍数的值。由于基于使用动量的优化算法不适合本模型，会导致模型收敛困难，因此使用 RMSprop 优化器，其学习率都设置为 $lr=0.00003$ ，为了加快收敛速度，每更新两次 D 网络的参数之后，在选择更新一次 G 网络。

GAN 也同样采用 min-batch 训练方法进行模型的训练，使用 Adam 优化器，其学习率都设置为 $lr=0.000001$ ，同样为了加快收敛速度，每更新 5 次 D 网络的参数选择更新一次 G 网络。

本文所涉及的数据扩充都是基于单一分类，即分别对训练集中单类别进行学习和生成，然后将生成数据加入到原数据集当中，使数据整体规模扩充一倍。

5.5 生成数据对比

为测试算法性能，本文的 TS-GAN 模型与 GAN 模型进行对比，分别观察生成数据的图像以及各扩充之后数据集的分类效果，以此来证明本模型数据增强的有效性。

5.5.1 生成效果对比仿真

为了保证每个模型的训练能够达到最优效果，本文的 TS-GAN 模型和 GAN 模型分别对其进行 5000 轮和 10000 轮的训练，每个类别生成 100 个数据，并通过图像化的方式展现出来，绿色的是生成数据，红色的是原始数据，左边为 TS-GAN 生成，右边为 GAN 生成。

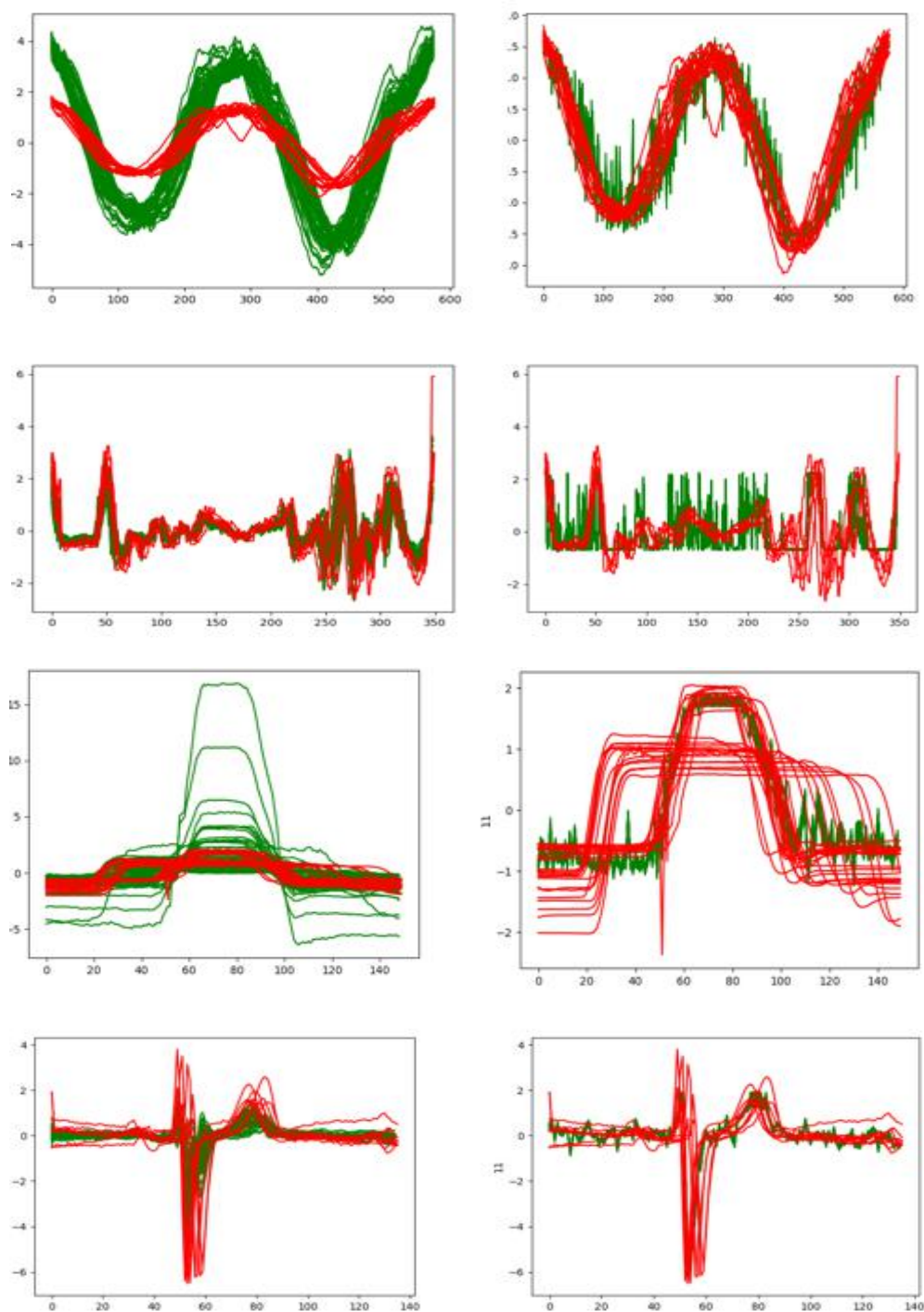


图 5.4 仿真图对比

仿真结果如图 5.4 所示，对比图片发现 GAN 网络在扩充特征较为简单的数据时，其数据范围始终围绕着原始数据，并且生成特征单一且粗糙，很多时候只是简单的模拟原始数据，并没有真正的获取特征，甚至在生成细节方面单个点具有较大误差。当面对复杂数据时，其模拟能力不够，扩充的数据与原始数据差异较大。除此之外 GAN 模型生成数据的细节不够平滑且有较大随机性，整体特征生成样本较为单一，出现了过拟合现象。而 TS-GAN 生成的数据能够跳出原始图像的范围，并且特征依然明显，细节随机性比较低，数据具有多样性。即使在面对复杂特征时，模拟能力足以生成具有该复杂特征的数据，生成数据也并不单一，没有表现出过拟合现象。

5.5.2 GAN 模型与原数据分类效果对比

由于精度是分类正确的数据与总数据的比值，所以对所有类别都会产生相同的性能，但因测试集数据不平衡，有必要对准确率和 F1 分数进行检测，作为每个类的性能评价指标。实验分为训练集数据增强前后的分类算法性能评估，其中原始数据集是未经数据增强的数据集，GAN 扩充是经 GAN 模型数据增强后的数据集，在相同条件下经 FCN 训练并通过测试集测试分类效果。

数据集	类别	准确率		精确率		召回率		F1分数	
		原始数据集	GAN扩充	原始数据集	GAN扩充	原始数据集	GAN扩充	原始数据集	GAN扩充
Beef	1			100	100	83.33	83.33	90.9	90.9
	2			85.71	100	100	100	92.3	100
	3	86.66	80	83.33	75	83.33	50	83.33	60
	4			71.42	54.54	83.33	100	76.92	70.58
	5			100	100	83.33	66.66	90.9	80
CBF	1	98.33	98.33	99.65	99.3	96	95.66	97.79	97.45
	2			99.32	99.33	99.32	100	99.32	99.66
	3			96.16	96.46	99.66	99.33	97.88	97.87
ECGFiveDays	1	85.36	85.13	86.29	96.01	83.87	73.13	85.07	83.02
	2			84.49	78.5	86.83	96.99	85.64	86.77
Gun_Point	1	98.66	94	100	91.35	97.36	97.36	98.66	94.26
	2			97.36	97.1	100	90.54	98.66	93.7
FaceFour	1	87.5	88.63	93.33	93.33	100	100	96.55	96.55
	2			80.64	86.2	96.15	96.15	87.71	90.9
	3			95.23	100	76.92	80.76	85.1	89.36
	4			85.71	78.26	81.81	81.81	83.72	80
DiatomSizeR	1			100	100	72.72	81.81	84.21	90
	2			96.84	91.08	100	100	98.39	95.33
	3	96.73	95.75	93	97.75	98.93	92.55	95.87	95.08
	4			100	97.75	100	100	100	98.86
平均		92.21	90.31	92.42	91.6	91.14	89.3	91.45	89.51

图 5.5 原始数据与 GAN 扩充对比

如图 5.5 所示，经 GAN 扩充的数据集无论是准确率还是 F1 分数，其精度分别下降了 1.9%和 1.94%。分类精度下降说明模型可能学习了不正确或者不平衡的数据，但无论是何种原因都证明 GAN 网络在时间序列生成方面效果较差，这可能是因为时间序列数据是一维的，点与点之间相关性较强，任何一个点出现问题都会引起整个数据的剧烈变化，而 GAN 并不能使相邻点有联系，使点的生成

具有极高的不确定性。进而导致生成效果较差。从结果看 GAN 并不适合时间序列的数据增强操作。

5.5.3 TS-GAN 模型与原数据分类效果对比

本节依然采用混淆矩阵统计各分类的数据，以说明经 TS-GAN 数据增强的数据集能够提高分类模型的分类效果。图 5.6 为原数据集与增强数据集的效果对比，TS-GAN 扩充为本模型数据增强后的数据集

数据集	类别	准确率		精确率		召回率		F1分数	
		原始数据集	TS-GAN扩充	原始数据集	TS-GAN扩充	原始数据集	TS-GAN扩充	原始数据集	TS-GAN扩充
Beef	1			100	100	83.33	83.33	90.9	90.9
	2			85.71	100	100	100	92.3	100
	3	86.66	96.66	83.33	100	83.33	100	83.33	100
	4			71.42	85.71	83.33	100	76.92	92.3
	5			100	100	83.33	100	90.9	100
CBF	1			99.65	99.65	96	96.33	97.79	97.96
	2	98.33	98.66	99.32	100	99.32	100	99.32	100
	3			96.16	96.47	99.66	99.66	97.88	98.04
ECGFiveDays	1	85.36	91.05	86.29	95.34	83.87	86.21	85.07	90.55
	2			84.49	87.55	86.83	95.84	85.64	91.51
Gun_Point	1	98.66	100	100	100	97.36	100	98.66	100
	2			97.36	100	100	100	98.66	100
FaceFour	1			93.33	100	100	100	96.55	100
	2	87.5	94.31	80.64	86.66	96.15	100	87.71	92.85
	3			95.23	100	76.92	88.46	85.1	93.87
	4			85.71	95.23	81.81	90.9	83.72	93.02
DiatomSizeR	1			100	100	72.72	87.87	84.21	93.54
	2			96.84	97.87	100	100	98.39	98.92
	3	96.73	97.71	93	95.78	98.93	96.8	95.87	96.29
	4			100	98.86	100	100	100	99.42
平均		92.21	96.4	92.42	96.96	91.14	96.27	91.45	96.46

图 5.6 原始数据与 TS-GAN 扩充对比

图中 TS-GAN 数据平均分类准确率达 96.4%高于原始数据的 92.21%，并且其 F1 分数也同样高于原始数据，其准确率和 F1 分数分别提高 4.19%和 5.01%，这说明了 TS-GAN 数据增强的有效性。从对比结果上看，本文采用的 TS-GAN 模型能够提高数据的分类性能，这使得时间序列数据在分类效果上，不只是通过应用更好的模型和增大数据量来提高分类精度，还可以在原有数据基础上生成相同分布的数据，解决分类模型因数据量不足而引起的过拟合的问题，加速了模型的收敛。

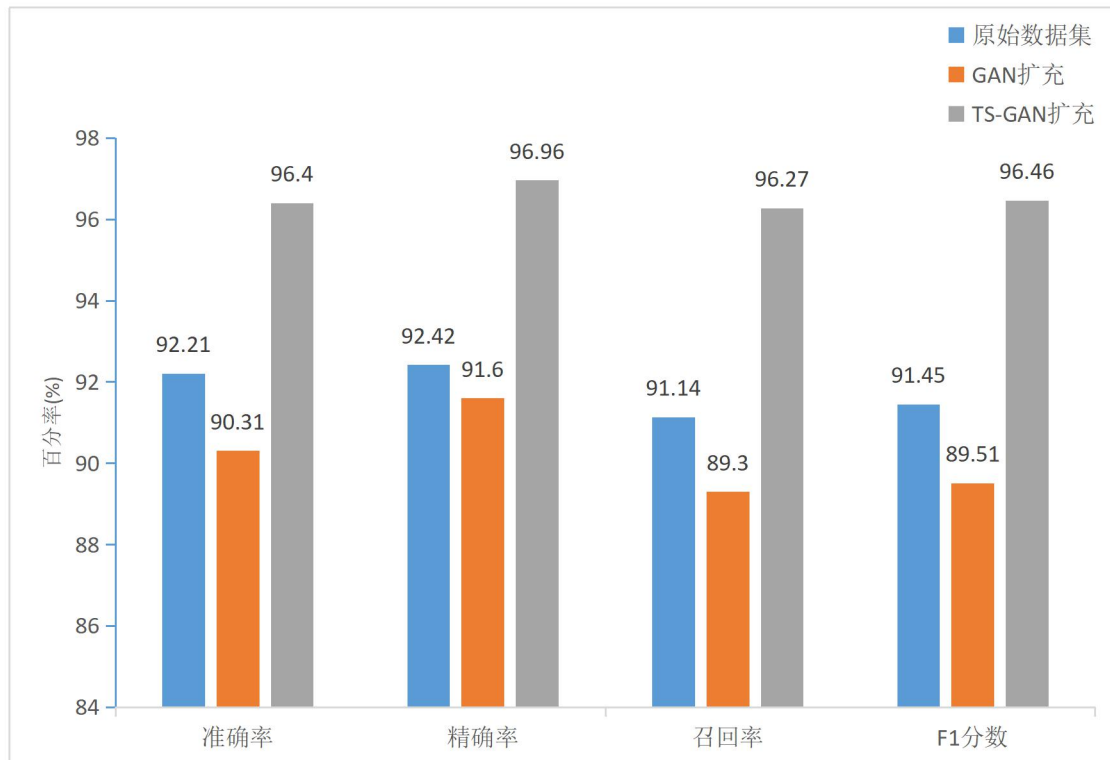


图 5.7 分类效果对比

如图 5.7 所示，本实验通过比较原始数据与基于 TS-GAN 和 GAN 模型数据增强后的数据的分类性能。发现 GAN 模型扩充的数据集分类精度低于原始数据数据集，分析原因是 GAN 网络本身就存在模型易崩溃，训练不稳定，特征的生成和判定能力较差等问题，且在一维的时间序列这种对单点较为敏感的数据，扩充数据的问题更加突出，往往会出现特征不全或者特征单一的数据，使扩充之后的数据集存在错误数据或者类别不平衡等问题，在分类算法学习的过程中，重要的特征不被学习或者较少被学习，同时特征较少的数据被大量学习就会使模型出现过拟合，这些问题的出现就导致了分类效果不如原始数据。而 TS-GAN 针对性对 GAN 出现的问题一一解决，针对时间序列数据的特征设计判定网络，加强了模型对该类数据的判定能力，并且舍弃生成能力较弱的全连接网络，使用一维反卷层来加强模型的特征生成能力，选用新的损失函数来衡量生成数据与真实数据的距离。这些改进极大地改变了模型捕捉和数据生成能力，实验结果表明，经 TS-GAN 增强后的数据集分类精度总是高于原始数据，这证明了时间序列数据可以通过数据增强来提高分类精度，即使数据相对较少也可通过 TS-GAN 准确捕捉其特征进行扩充。

5.6 结果评价

本文分别使用多种类别的数据，在相同条件下分别使用 GAN 和 TS-GAN 进行训练，单独使用各自的生成网络生成数据，并以二维图像方式将其描绘出来，在外观上对生成的数据进行评价。通过对比图像发现，GAN 网络的图像模型崩溃比较严重，生成数据较为单一，并不能脱离原数据各部分数值的范畴。TS-GAN 生成的数据具有连贯性，生成数据较为多样，并且能够脱离原来数据的范畴，具有明显的特征。将扩充之后的数据加入训练集分别使用 FCN 在同等条件下使用测试集进行分类，经 TS-GAN 扩充的数据分类精度高于原始数据，GAN 网络扩充后的数据集却低于原数据集，这说明 TS-GAN 相较于 GAN 有更好的数据增强能力，能够准确学习原始数据的特征，通过扩充数据强化某些特征在模型中出现的频率，最终提高模型分类效果。

5.7 小结

本章首先从硬件和软件方面介绍测试环境和所用的运行平台，再介绍实验数据集来源和对数据的评价标准。除此之外介绍了对照 GAN 网络以及 FCN 分类模型的组成，在数据增强方法中详细介绍了模型的搭建与训练，包括网络层的配置、各参数的设置等，最终分别使用 TS-GAN 与 GAN 扩充原始数据集，并经 FCN 模型就分类效果进行比较，结果表明经 GAN 扩充数据集，无论是准确度还是 F1 分数上都低于原始数据集，这说明 GAN 不适用对时间序列进行数据增强。而 TS-GAN 模型却恰好相反，通过学习原始数据的特征扩充原始数据集，能够提高该数据集的分类效果。这使得针对时间序列数据特点而设计的 TS-GAN 达到了数据增强的目的，克服了 GAN 网络在面对时间序列数据时其本身存在的缺陷。

6 总结

通常来说,传统的机器学习方法(例如 KNN^[44])在时间序列数据分类上虽然能取得比较好的效果,但往往在大数据集上弱于深度学习方法。而深度网络优秀的表现必须依赖较大的训练数据规模,否则精度很难超过传统机器学习方法。在这种情况下,进行数据增强、收集更多的数据等都是提高深度网络模型精度的方法,这其中数据增强被用来解决数据量不足的问题。鉴于 GAN 网络的局限性,本文提出基于注意力机制的判定网络,使用一维反卷积神经网络设计 G 网络,并引入更好的损失函数,使用 Wasserstein 距离并加入梯度惩罚。通过较强的 D 网络来捕捉准确的特征, G 网络生成能力也随之加强,最后通过损失函数使生成的数据与原始数据分布尽可能一致,因此本文所介绍的 TS-GAN 在时间序列扩充方面有比 GAN 较好的表现。

6.1 主要工作总结

为了充分利用现有时间序列数据,发挥其所有潜能去提高分类模型的分类精度,本文在 GAN 的基础上针对性的设计了 TS-GAN,并在不增加额外数据的情况下,只是将原始数据进行数据增强就取得了较好的分类效果。本文做出的主要工作内容如下:

(1) 提出了基于注意力机制设计的 D 网络模型,增强 TS-GAN 对数据特征的捕捉以及判定能力,引入多注意力机制使得每一层网络上尽可能多的保留特征,相比于传统的卷积神经网络,其传递给下一层的特征更加丰富,同时更多的卷积核也能捕捉到更多的特征来增加分类的效果,当 D 网络变得越好, G 网络模拟数据的难度也越来越大,以至于传统的全连接层并不能够满足要求。因此本文设计反卷积网络增加 G 网络模拟数据的能力。为了更好的衡量生成数据与真实数据的距离,该模型使用了 Wasserstein 距离,通过惩罚梯度限制梯度爆炸,这使得网络能够更好的对抗生成。实验证明,相比于原始数据,使用 TS-GAN 增强之后的数据,其 FCN 模型分类效果得到提高。

(2) 为了验证 TS-GAN 是否对时间序列的数据增强具有针对性,本实验以 GAN 网络作为对照,分别扩充同样的数据以相同的数量,并与原数据一块对比其在 FCN 模型分类效果。通过实验证明, GAN 网络扩充的数据分类效果相对

于原始数据分类效果下降,远低于 TS-GAN 扩充的数据,这说明 GAN 网络在时间序列数据扩充方面能力较差,并不能准确扩充数据特征,而具有相同工作模式下的 TS-GAN 却表现优异适合时间序列的数据增强操作,这说明 TS-GAN 对时间序列的数据增强具有针对性。

6.2 未来工作展望

TS-GAN 对抗网络模型通过沿用 GAN 生成对抗的思想,使用以注意力机制为核心的架构作为判定网络,针对时间序列的特征设计生成网络,并使用当前最好的损失函数,本文以一种新颖的方式将其结合起来,与其他生成对抗网络相比,进一步提高了数据增强之后模型分类的精度,但该模型仍有多种不足,经过分析有以下需要改进之处:

(1) 本实验中使用的 UCR 文件中,由于每个数据集的训练集数据较少,导致单个类别数据也较少,极端情况下只有一个,平均也就十几个,但根据在图片领域的经验,其训练集中都是几千几万张图片,才能取得比较稳定的效果,因此为了验证其扩充的稳定性,需要寻找大量数据进行数据扩充,观察其是否会因为数据量较大而导致模型崩坏。

(2) 本次使用的 D 网络通过引入注意力机制设计架构,由于其对神经网络层数有限制,即网络模型较浅的层次梯度消失,因此并不能单纯叠加层数来增强判定网络的特征捕捉能力,因此接下来尝试既能增加深度又不失去梯度的网络层,因此使用 LSTM 以及 RCNN 等网络层去设计 D 判定网络,其性能有望超过本文所介绍的网络。

(3) 由于 TS-GAN 扩充数据时,是学习单个数据特征然后再扩充,当类别较少时这样做不会引起很大问题,但如果类别变为几十甚至几百,这就会导致训练时间长,磁盘空间占用比较大等问题,这些问题使得扩充数据变得极其麻烦,因此需要找出即能准确扩充相应数据又能在扩充时耗费较少资源的一种新型的生成对抗网络。例如可以增加额外信息 y ,将 y 作为类别信息加入到原始数据中,一同输入判别模型和生成模型,通过对所有类别一同进行训练,共用同一模型中的同一权值,当需要生成指定类别数据时,将类别信息 y 和随机噪声输入生成网络即得到该类别扩充的数据,通过这种方式可以有效减少生成对抗网络模型所需要的资源。

参考文献

- [1] John Paparrizos and Luis Gravano. Fast and Accurate Time-Series Clustering. ACM Transactions on Database Systems, 42(2): 1-49, June 2017.
- [2] Lecun Y, Bengio Y, Hinton G. Deep learning[J]. Nature, 2015, 521(7553): 436-444.
- [3] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. Science, 2006, 313(5786): 504-507
- [4] Krizhevsky A, Sutskever I, Hinton G. ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2), 1097-1105, DOI: 10.1145/3065386
- [5] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. arXiv preprint arXiv:1409.1556, 2014
- [6] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. 2015: 770-778.
- [7] Chen Sun, Abhinav Shrivastava, Saurabh Singh, Abhinav Gupta. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era[J]. arXiv preprint arXiv:1707.02968
- [8] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
- [9] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, vol. 15, no. 1, pp. 1929-1958, 2014.
- [10] Bartlett Peter L, Long Philip M, Lugosi Gábor, Tsigler Alexander. Benign overfitting in linear regression.[J]. Proceedings of the National Academy of Sciences of the United States of America, 2020.
- [11] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets[C]. Advances in Neural Information Processing Systems, pp. 2672-2680, 2014
- [12] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep

- convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.
- [13] Arjovsky M, Chintala S, Bottou L. Wasserstein gan[J]. arXiv preprint arXiv:1701.07875, 2017.
- [14] Giovanni Mariani, Florian Scheidegger, Roxana Istrate, Costas Bekas, et al. BAPAN: Data Augmentation with Balancing GAN. 2018.
- [15] Ali-Gombe A, Eyad E. MFC-GAN: Class-imbalanced Dataset Classification using Multiple Fake Class Generative Adversarial Network[J]. Neurocomputing, 2019, 361.
- [16] 王海文. 基于生成式对抗网络的数据增强方法研究[D]. 南京邮电大学, 2019.
- [17] Mirza M, Osindero S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.
- [18] Jin Q, Lin R, Yang F. E-WACGAN: Enhanced Generative Model of Signaling Data Based on WGAN-GP and ACGAN[J]. IEEE Systems Journal, 2019, PP(99):1-12.
- [19] Tanner M A, Wong W H. The calculation of posterior distributions by data augmentation[J]. Journal of the American statistical Association, 1987, 82(398): 528-540.
- [20] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, & W. Philip Kegelmeyer. (2002). Smote: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16(1), 321-357.
- [21] Cubuk E D, Zoph B, Mane D, et al. AutoAugment: Learning Augmentation Policies from Data[J]. 2018.
- [22] Gal Y, Ghahramani Z. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning[J]. 2015.
- [23] 高彦平. 图像增强方法的研究与实现[D]. 山东科技大学, 2005.
- [24] Zhang H, Cisse M, Dauphin Y N, et al. mixup: Beyond Empirical Risk Minimization[J]. 2017.
- [25] Inoue H. Data Augmentation by Pairing Samples for Images Classification[J]. 2018.

- [26] Zheng H , Fu J , Mei T , et al. Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition[C]// IEEE International Conference on Computer Vision. IEEE, 2017.
- [27] Inoue H . Data Augmentation by Pairing Samples for Images Classification[J]. 2018.
- [28] Krizhevsky A , Sutskever I , Hinton G . ImageNet Classification with Deep Convolutional Neural Networks[J]. Advances in neural information processing systems, 2012, 25(2).
- [29] Sergey Ioffe, Christian Szegedy. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on International Conference on Machine Learning -Volume 37 (ICML'15). Lille: ACM, 2015. 448-456.
- [30] 圣飞, 阴亚芳, 秦晨蕊,等. 基于对数运算的 Sigmoid 函数的研究与实现[J]. 微电子学与计算机, 2018, v.35;No.405(02):17-20.
- [31] He K , Zhang X , Ren S , et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification[J]. 2015.
- [32] Barz B , Rodner E , Garcia Y G , et al. Detecting Regions of Maximal Divergence for Spatio-Temporal Anomaly Detection[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(5):1088-1101.
- [33] Zhao Y , Takaki S , Luong H T , et al. Wasserstein GAN and Waveform Loss-based Acoustic Model Training for Multi-speaker Text-to-Speech Synthesis Systems Using a WaveNet Vocoder[J]. IEEE Access, 2018:1-1.
- [34] 徐柳静, 彭定涛, 王鑫. 一类非 Lipschitz 约束优化的最优性条件%Optimal Conditions for a Class of Non-Lipschitz Constrained Optimizations[J]. 贵州大学学报:自然科学版, 2017, 034(002):10-13,17.
- [35] Zheng H , Fu J , Mei T , et al. Learning Multi-attention Convolutional Neural Network for Fine-Grained Image Recognition[C]// IEEE International Conference on Computer Vision. IEEE, 2017.
- [36] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial Transformer Networks. In 2015 Annual Conference on Neural Information Processing Systems, pages 2017- 2025, Montreal, Canada, December 2015. MIT

Press.

- [37]Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Enhua Wu. Squeeze-and-Excitation Networks. *ArXiv* Homepage, <https://arxiv.org/abs/1709.01507>.
- [38]Sanghyun Woo, Jongchan Park, JoonYoung Lee, and In So Kweon. CBAM: Convolutional Block Attention Module. In Proceedings of the 15th European Conference on Computer Vision, Part VII, pages 3-19, Munich, Germany, September 2018. Springer Press.
- [39]Lee J H , Park K H . GAN-based imbalanced data intrusion detection system[J]. Personal and Ubiquitous Computing, 2019(9).
- [40]Kok Keong Teo, Lipo Wang, Zhiping Lin. Wavelet Packet Multi-layer Perceptron for Chaotic Time Series Prediction: Effects of Weight Initialization[M]// Computational Science - ICCS 2001. Springer Berlin Heidelberg, 2001.
- [41]Berger L , Hyde E , Cardoso M J , et al. An Adaptive Sampling Scheme to Efficiently Train Fully Convolutional Networks for Semantic Segmentation[J]. 2017.
- [42]Wang Z , Yan W , Oates T . Time series classification from scratch with deep neural networks: A strong baseline[C]// 2017 International Joint Conference on Neural Networks (IJCNN). IEEE, 2017.
- [43]Zhiguang Wang, Weizhong Yan, and Tim Oates. Time series classification from scratch with deep neural networks: A strong baseline. In 2017 International Joint Conference on Neural Networks, pages 1578-1585, Anchorage, USA, May 2017. IEEE Press.
- [44]Hsu H H , Yang A C , Lu M D . KNN-DTW Based Missing Value Imputation for Microarray Time Series Data[J]. Journal of Computers, 2011, 6(3):418-425.

致谢

随着毕业论文的撰写完成，仅仅两年的研究生生活即将结束。在华中科技大学的两年生活时间中，有着很多的开心和艰难的岁月。非常感谢这所学校所提高的优异平台和环境让我们不断充实自己、磨练自己，并最终在步入社会时拥有扎实的能力和对技术的热情。

由于本科不是相关专业，学习相对吃力一些，但比较幸运的是，导师石柯给予充分的时间去熟悉和学习相关内容，在确定研究方向上面，和他的每一次谈话都深受启发。渊博的学识使其看问题一针见血，对学术的严谨态度和无私认真的工作态度让我们感到钦佩，这对我们未来的人生、工作产生积极长远的影响。

此外还要感谢陈炜博士的指导和引领，每当我遇到学习困难时，他总能在思路上给予我启发，使我更快地调整自己的想法。生活上非常感谢家人对我的支持和信任，让我在避免学习之外的打扰。

最后感谢在座所有的评审老师，感谢您在百忙之中抽出时间评阅我的论文，提供宝贵的意见！