

密级： 保密期限：

北京邮电大学

硕士学位论文



题目： 基于权值共享的深度卷积网络的
电力系统暂态稳定预测

学 号：

姓 名：

专 业： 信息与通信工程

导 师：

学 院： 信息与通信工程学院

年 月 日

基于权值共享的深度卷积网络的电力系统暂态稳定预测

摘 要

随着可再生能源的高度渗透，电力负荷的进一步增长，以及电力电子化的广泛应用，电力系统的结构和运行方式变得越来越复杂，其安全稳定运行正在面临严峻的挑战。近年来，国内外发生了多起大面积停电事故，造成了严重的经济和社会影响，也暴露了现有的电网稳定分析系统在运行机制与时效性上的诸多不足。这些系统在预测的精确性和实时性等方面还存在巨大的改善空间。

受深度卷积神经网络在特征学习上的启发，本文提出了一种利用深度卷积网络进行电力系统暂态稳定评估的方法。本方法可以自动从 PMU 采集的电压和相角原始数据中获取不同抽象层次的特征信息，因此不需要进行对数据的前期特征处理。具体来说，我们提出了一种有效的处理多维多通道时间序列的深度卷积网络模型，每个通道包含一个二维的时间序列，表示电力网络中单个母线的复电压时序信息（由母线电压和相角数据经过简单坐标变换构成）。各个通道的数据通过权值共享的深度卷积网络，得到其隐含的抽象信息。然后，模型将这些信息输入一个多层感知器（MLP），来完成稳定预测。我们利用基于模型梯度的方法来确定网络参数，然后在基于 IEEE39 系统的仿真场景中进行了性能验证。结果表明我们的模型在速度和精度两方面都有良好的效果。

此外，为了利用现实电网中存在的大量无标签真实数据，本文提出了一种针对此模型的无监督预训练方法，以提高模型精度。本文也对学习到的特征进行了可视化分析，来提供对学习到的特征的直观表示。

关键词 卷积神经网络 电力系统 暂态稳定 深度学习

MULTI-CHANNEL DEEP CONVOLUTIONAL NETWORKS WITH SHARED WEIGHT LAYERS FOR POWER SYSTEM TRANSIENT STABILITY ASSESSMENT

ABSTRACT

With the gradual popularization of renewable energy supply, the further increase of power load and the extensive application of power electronic equipment, the network structure and operating mechanism of power system has become more and more complex, and its safe and stable operation is facing severe challenges. In recent years, there have been many large-scale blackouts at home and abroad, resulting in serious economic and social impact, but also exposed the various deficiencies of the existing grid stability analysis system in the operating mechanism and timeliness. There is still a huge room for improvement.

In this paper, an architecture of deep convolutional network for power system transient stability assessment is proposed, which is inspired by deep convolutional neural network used in the feature learning studies. The method can automatically obtain the characteristic information of different abstraction levels from the voltage and phase angle raw data collected by the PMU, and therefore does not need to deal with the complex manual feature extraction process. In particular, we propose an efficient deep convolution network model for multidimensional multichannel time series. Each channel receives a two-dimensional time series, which represents the complex voltage timing information of a single bus in the power network (from the bus voltage and the phase angle data are transformed by simple coordinates). The data from each channel are then processed by the weight sharing deep convolutional networks, and the hidden feature information is obtained. The model then

inputs this information into a multi-layer perceptron (MLP) to perform a stability prediction. We use the model-based gradient method to determine the network parameters, and then based on IEEE39 system simulation scenarios for performance verification. The results indicate that our model has a good effect in both predicting speed and precision.

In addition, in order to take advantage of the large amount of untagged real data existing in the real power grid system, this paper proposes an unsupervised pre-training method to improve the accuracy of the model. This paper also provides a visual representation of the learned features by visualizing the convolution kernels.

KEY WORDS convolutional neural network, power system, transient stability, deep learning

目 录

第一章 绪论.....	1
1.1 课题背景与意义.....	1
1.2 国内外研究现状.....	2
1.2.1 电力系统暂态稳定研究现状.....	2
1.2.2 机器学习方法的研究现状.....	5
1.2.3 基于机器学习的电力暂态稳定预测方法.....	7
1.3 本文的主要工作.....	8
1.4 论文结构安排.....	9
第二章 电力系统的暂态稳定性理论分析.....	11
2.1 电力系统的暂态过程概述.....	11
2.2 暂态稳定的物理过程分析.....	11
第三章 人工神经网络理论.....	15
3.1 人工神经网络.....	15
3.1.1 多层感知器网络.....	15
3.1.2 卷积神经网络.....	18
3.1.3 现代人工神经网络.....	20
3.1.4 深度神经网络的预训练.....	23
3.2 硬件加速的人工神经网络计算.....	24
3.2.1 Cuda.....	25
3.2.2 Tensorflow.....	26
3.2.3 Keras.....	28
3.3 小结.....	28
第四章 基于深度学习的暂态稳定预测方案.....	31
4.1 定义与符号.....	32
4.2 问题描述.....	32
4.3 模型架构.....	33
4.4 坐标转换层.....	34
4.5 卷积网络.....	35
4.5.1 滤波层.....	35
4.5.2 激活层.....	35
4.5.3 最大池化层.....	36

4.6 多层感知器.....	36
4.7 前向传播.....	36
4.8 模型训练.....	37
4.8.1 通道间权值共享机制.....	37
4.8.2 反向传播过程.....	38
4.8.3 更新权值.....	39
4.9 决策边界与评价指标.....	40
4.10 神经网络预训练.....	41
4.11 工程实现.....	41
4.12 小结.....	42
第五章 实验设计与结果验证.....	43
5.1 实验数据.....	43
5.2 模型设置.....	44
5.3 结果分析.....	44
5.4 卷积网络可视化.....	48
5.5 小结.....	50
第六章 总结与展望.....	51
参考文献.....	53
攻读学位期间发表的学术论文目录.....	57

第一章 绪论

1.1 课题背景与意义

自 1882 年 T.A.托马斯·阿尔瓦·爱迪生在纽约珍珠街主持建造人类历史上第一座中心电站式的供电系统^[1]以来,电力系统的安全稳定运行就一直是电力领域的核心问题。1990 年以降,三相交流供电研制成功,并成为了电力系统的主流输电方式,也将电力系统的稳定问题变得更为复杂。近年来,随着国民经济的快速发展和可再生能源的逐渐渗透,电力系统已经形成了大系统、大电网、超高压、长距离、重负荷、大区联网、交直流联合输电^[2]等复杂特性,电力系统呈现出更为复杂的结和行为特性,为电力系统的安全稳定运行提出了新的挑战。

近年来,国内外电力系统由于稳定破坏,发生了多起大面积停电事故,对国民经济造成了极大损害,社会和人民的生活也受到了很大的影响。2012 年 7 月 30 日和 31 日,印度连续两次发生大停电事故,损失负荷达到 4800 万 kW,停电波及 23 个邦的 6.7 亿人口,造成约数百亿美元的损失^[3]。

因此,提高电力系统的调度运行水平非常重要。目前,美国能源部提出了《Grid 2030》,计划到 2030 年,实现一个完全自动化的电力传输网络,从而可以监视和控制每个用户和电网节点,保证从电厂到终端用户整个输配电网络中所有节点之间信息和电能的双向流动^[4]。德国于 2008 年在智能电网的基础上选择了 6 个试点地区进行为期 4 年的 E-Energy 技术创新促进计划,成为实践能源互联网最早的国家^[5]。中国于 2010 年确立了建立电网调度技术支持系统的基本思路。2016 年,中国国家电网公司提出了依托特高压交直流和智能电网技术的全球能源互联网概念,为推动世界能源安全、清洁、高效、可持续发展提供了全新的解决方案^[6]。

提升电力系统调度运行水平的关键在于在线安全稳定分析。它考虑了电网多变的运行方式,提高了电网仿真分析的时效性。但随着应用的深入和调度自动化水平的不断提高,如何有效利用不断积累的稳定运行和计算结果数据,成为了特约在线功能提升的瓶颈问题。

当前投入使用的电网在线安全评估与防御系统以建模仿真为核心,受模型参数和数值近似计算的制约,其性能往往无法满足大电网在线安全防控的需求。现有的在线评估系统在实际应用中,往往将电网中某一时刻的潮流信息作为初始状态,并针对各种形式的预想故障集进行建模与仿真分析,或者针对某一确定时段的动态信息进行研究,对电网运行数据中存在着的时空关联数据缺乏有效的挖掘利用,

造成了巨大的广域量测资源的浪费。当前大停电的主要原因之一在于在电网情况缓慢恶化阶段,缺乏有效的稳定态势量化评估与预测信息,因此也难以给出基于此的关键防控措施,从而错失了力挽狂澜的最佳时机^[7]。

另一方面,随着电网与互联网,通信网等的深度融合,和智能化水平的不断提高,反映电网运行及发展进程的细粒度能源数据的收集已经成为可能,这为数据驱动的实时有效处理方法提供了信息基础,有助于新见解的揭示,为大电网系统的进一步发展带来新的机遇。

在现阶段开展面向在线稳定计算需求的大数据综合分析处理技术基础性研究,从众多的在线数据中自动发现电网安全运行的特征与规则,进而提升大电网调度运行在线安全分析能力,增强电网抵御外部因素对安全稳定运行影响的能力,具有重大的现实意义。

本文的研究点定位于使用机器学习方法解决电网的暂态稳定问题。本文针对电力系统的现行调度需求,提出了基于权值共享的深度卷积神经网络的电力系统暂态稳定预测方法。直接将 PMU 采集到的电压和相角数据以时空矩阵的形式作为输入,利用卷积网络进行特征提取与稳定判别。

1.2 国内外研究现状

本文的研究点是基于机器学习方法的电网暂态问题研究,具有很强的交叉学科特性。因此,对于国内外研究现状的分析也将从这两方面同时展开,来寻找双方的交叉切入点。从结构安排上,一方面,本文将首先介绍电力系统稳定问题的明确定义与分类法,引出暂态稳定问题的详细内容与描述,并讨论几种经典的暂态稳定分析方法。另一方面,本文也会对机器学习的发展作一个简要的概括,分析各种方法在电力系统暂态稳定问题上的应用可能性。最后,本文将结合两者的具体情况,给出一种可行的解决方案。

1.2.1 电力系统暂态稳定研究现状

国内外对于电力系统暂态稳定问题的研究大约从 19 世纪 80 年代开始,在近几年达到最热,至今共有数千篇相关研究文献。随着研究的不断深入,出现了越来越多的相关研究点,主要包括电力系统暂态稳定的评估与预测,电力系统的紧急控制,暂态能量函数等。电力系统暂态稳定的跨学科研究也发展迅猛,已经深入到计算机科学与技术,应用数学,机械工程等诸多领域,并衍生出一些交叉主题学科。在我国的电力研究领域,中国电力科学研究院,清华大学,华北电力大学,浙江大学,武汉大学,四川大学等都贡献了自己的突出力量。

电力系统稳定分类

随着时代的进步和对电力系统稳定问题研究的逐步深入,电力系统稳定的定义和分类也在不断变化。此外,国内外的标准也有所不同。我国的行业标准主要包括《电力系统安全稳定导则》、《电力系统安全稳定计算技术规范》(DL/T 1234-2013、《国家电网公司电力系统安全稳定计算规定》^[8]等。国外的定义和分类标准主要由 IEEE/CIGRE 等制定。孙华东等人在《电力系统的稳定与分类叙述》中,对国内外不同的定义和分类依据做出了详细的介绍和对比描述。

从本质上来说,电力系统稳定(power system stability)是指电力系统在受到扰动后保持稳定运行的能力。根据电力系统失稳的物理特性、扰动的大小以及设备、过程和时间框架等,可以将电力系统稳定分为功角稳定、电压稳定和频率稳定三大类,其中,每类还包含若干子类。图 1-1 给出了电力系统稳定性的一种分类方法。本文主要研究电力系统的暂态功角稳定问题,它是电力系统稳定问题研究的焦点之一。以下给出相关问题的详细分类^[9]。

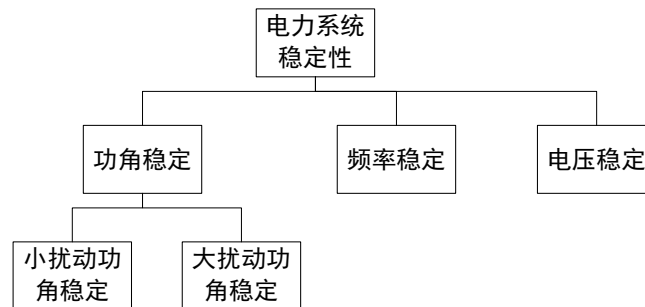


图 1-1 电力系统稳定性的分类

暂态功角稳定是功角稳定的一类特殊问题。其中,功角稳定(Rotor Angle Stability)是指电力系统中的同步发电机受到扰动冲击之后,维持稳定同步运行的能力。功角失稳意味着某些发电机转子之间的相对角度增大,失去同步运行能力。它会在电网中产生了能量的振荡。严重的功角失稳甚至会导致系统崩溃^[10]。

从扰动的大小来看,功角稳定可以分为大扰动功角稳定和小扰动功角稳定两类问题。暂态功角稳定问题从属于大扰动问题,它主要研究电力系统在经历短路、元件(线路、发电机、负荷等)切除、发电机励磁失效、冲击性负荷等极端情况和严重故障时,电力系统保持同步运行的能力。不同于小扰动问题,在系统受到大扰动冲击时,系统的稳定状态不仅与其初始运行状态有关,还与扰动的实际严重程度有着莫大的关系。

经典暂态稳定分析方法

经典的电力系统暂态稳定分析方法主要包括时域仿真法、直接法和混合法等

[11]。

时域仿真法

时域仿真法(time simulation)是目前发展最为成熟,应用最为广泛的暂态稳定分析方法,又称逐步积分法(step by step)。它被广泛应用于电力系统研究、设计,运营等各个方面。时域法依据电力系统元件的物理性质和拓扑关系对整个系统进行数学建模。在其上建立描述各部分行为的微分——代数方程组。它充分保留了系统的非线性特征和动态响应特性,将稳态潮流计算的结果作为系统初始值,通过联立并求解微分方程的形式,计算系统状态量随时间的变化曲线^[12]。

时域仿真法的优点在于具有很强的模型适应能力,可以综合考虑到各种励磁系统、调速系统、稳定控制措施和继电保护装置等对电力系统暂态稳定性的影响。此外,它还具有发展成熟的数值算法,可以方便地在计算机上实现。其缺点在于其过度依赖系统模型和参数的准确性,在研究是大规模电力系统时,维数较高、计算时间和空间复杂度大。时域仿真法的稳定判别主要依靠人工经验进行,无法给出系统的暂态功角稳定裕度和灵敏度信息。多用于暂态功角稳定问题离线分析,在实时分析领域中的应用较少。

2000 年, Liu C W 等人提出了一种改进的时域仿真方法,扩大了系统初值的选择范围,使其不仅仅局限于潮流计算的结果^[13]。2007 年,李琰等人提出了一种对仿真系统进行降阶等值的方法,通过这个方法可以减少系统规模,进而提高运行效率^[14,15]。

直接法

直接法(Direct Method)主要利用系统能量及其转化等角度分析暂态稳定问题。相对于时域仿真法,它不通过仿真系统的运行轨迹来观察暂态功角稳定问题,因此不要求解系统微分方程,计算速度有了很大提升。同时,直接法还可以提供系统的稳定裕度等信息。按照分析方法,直接法大体上可以分为暂态能量函数法(Transient Energy Function Method)和扩展等面积准则(Extended Equal Area Criterion, EEAC)两大类^[16]。

暂态能量函数法的主要方法包括相关不稳定均衡点法(Relevant Unstable Equilibrium Point, RUEP)、潜在势能边界法(Potential Energy Boundary Surface, PEBS)和基于稳定域边界的主导不稳定均衡点法(Boundary of Stability Region Based Controlling Equilibrium Point, BCU)等。

扩展等面积法根据电力系统的物理特性进行系统简化。它首先把整个系统机组分为临界机组和非临界机组,然后再对其进行聚合和保稳变换,使得整个系统等值为一个单机无穷大系统(One-machine Infinite Bus system),最后应用等面积准

则判断该系统的稳定性。扩展等面积法的关键在于临界发电机组的确定，这主要借助时域仿真法来完成。通过反复试探得到临界清除时间最小的候选机组。这与基于稳定理论的上一种直接法有本质的区别。相对于直接法而言，其主要优势在于可以进行灵敏度分析。

近年来，学术圈在临界机组的判定上有所突破。Ota Y 等人提出了一种利用 PMU 数据进行临界机器判定的方法^[17]。此外，在扩展等面积法的基础上，滕林等人基于广域量测信息提出的新的判定法则，在实时分析和控制上取得了不错的效果^[18]。

混合法

混合法（Hybrid Method）是指一类将时域仿真和暂态能量函数法相结合的分析方法。它兼具时域法和直接法两者的一些优点，既能适应复杂电力系统元件，也能给出系统的稳定裕度。它首先通过时域法仿真得到系统的变量曲线，然后再据此进行系统能量的计算，从而得出系统的暂态稳定性^[19]。

1.2.2 机器学习方法的研究现状

机器学习发展简史

本文的研究可以落入机器学习的范畴，机器学习是人工智能的一个核心研究领域。机器学习的一个经典定义是“计算机利用经验改善系统自身性能的行为”，由 Mitchell T M 在 1997 年完成的《Machine Learning》一书中给出^[20]。机器学习意图模仿人类获取知识（学习）的非平凡过程，具体体现为机器从数据中获取知识，并对这些知识进行创新和重构，进而实现自身问题处理能力提升的过程。

近年来，机器学习已经成为了一门新的交叉学科，并成为了众多高校的独立课程。通过融合心理学、生物学、神经生理学以及数学、自动化、计算机科学与技术等众多学科的先进理论，机器学习已经构建了充实的理论基础。机器学习与人工智能的各种基本问题的统一观点也正逐步形成。各类机器学习方法的应用范围正在不断扩大，一部分应用已经成为商品，正在渗透进入人们的日常生活。2016 年，由谷歌子公司 DeepMind 开发的基于机器学习算法的 AlphaGo 围棋 AI 以 4:1 的悬殊比分战胜了世界围棋冠军李世石，成为机器学习发展的里程碑事件。

机器学习是人工智能中较为年轻的分支。其早期研究出现于 20 世纪 40 年代。1943 年，Warren 等人提出了神经网络层次结构模型，成为了人工神经网络和深度学习的基础计算模型^[21]。1950 年，有“人工智能之父”之称的天才科学家图灵提出了著名的“图灵测试”方案，使人工智能成为了计算科学领域一个重要的研究课题。

在 1955 年到 1965 年期间,机器学习相关的研究曾非常活跃。当时的主要研究集中在“无知知识”学习上,研究目标是各类自组织与自适应系统。这段期间最有代表性的成果是塞缪尔(Samuel)发明的下棋程序。值得遗憾的是,这一阶段机器学习的发展并不尽如人意。

也因为这个原因,机器学习的研究在之后的十年中进行了一个冷静时期。本阶段的研究目标转向模拟人类学习过程的概念学习阶段,尝试采用逻辑结构或图结构的形式对机器内部进行概念描述。这一时期的代表性成果是由 Winston 的结构学习系统,以及由 Hayes Roth 等人提出的归纳学习系统。

从 70 年代中叶开始,机器学习的研究进入了一个复兴时期。人们通过将学习单个概念扩展到学习多个概念,并开始探索不同的学习策略和方法。这些方案取得了很大的成功,有力促进了相关领域研究的发展。

1980 年,福岛邦彦提出了新认知机,成功通过多层神经网络学习进行手写字母的图像识别^[22]。1989 年,燕乐存(Yann LeCun)等人将 1974 年提出的标准反向传播算法应用于深度神经网络,来进行手写邮政编码识别^[23]。1991 年,赛普·霍克赖特(Sepp Hochreiter)提出了梯度消失问题,解释了深度神经网络训练效率低下的原因^[24]。

1991 年,翁巨扬(Juyang Weng)发表了生长网(Cresceptron),一种用于进行一般自然杂乱图像中自然物体识别的深度学习网络。其中首次提到了后来很多实验中广泛采用的一个方法:最大池化(Max-pooling)用于处理大物体的变体问题。生长网不仅可以从杂乱自然场景中识别给定物体,还可以通过反向网络分析的方法确定物体在背景图像中的位置^[25]。

90 年代也有了多种浅层机器学习模型先后问世,如逻辑回归,支持向量机等,这些模型以凸优化问题作为数学基础,理论相对简单,训练较为容易,因此得到了广泛的应用。基于统计规律的浅层学习方法相对之前的基于规则的方法在很多方面有明显优势,因此得到了不少成功的商业应用。但是相对神经网络算法,由于计算单元有限,这些算法对复杂函数的表示能力不强,对复杂问题的求解主要依赖人工的特征提取工作。

2007 年前后,Geoffrey Hinton 和 Ruslan Salakhutdinov 等人提出了一种在前馈人工神经网络中进行有效预训练的算法。这种算法将网络中的每一层视为无监督的受限波尔兹曼机(Restricted Boltzmann Machine),对其进行逐层贪婪预训练,组合构成深度网络后,再通过有监督的反向传播算法进行调优。这种算法为训练深度的人工神经网络提供了一种可行方案^[26]。

如今,深度学习已经成为了机器学习新兴的发展方向。高性能图形处理器以及专用 FPGA 和 DSP 芯片的出现极大地提高了数值和矩阵运算的速度,显著缩

短了模型的训练时间。深度学习的新模型，新技术正在不断涌现，如深度卷积网络、深度信念网络、递归神经网络等。这些技术在计算机视觉、语音识别、自然语言处理、音频识别与生物信息学等领域获得了深入应用并获取了极好的效果。

国内对于机器学习的研究从 1970 年左右开始兴起。到近年来达到最热，40 多年间，一共积累了超过 6000 余篇相关论文与文献。随着相关研究的不断深入，出现了越来越多的相关研究点，形成了巨大的研究网络。主要的相关研究点包括支持向量机、决策树、人工神经网络、模式识别、最优化算法、特征分析、专家系统等。

机器学习算法分类

从学习方式上来看，机器学习算法可以分为有监督学习和无监督学习两大类：

有监督学习算法包括一个目标变量（因变量）和用来预测目标变量的特征变量（自变量）。学习算法通过这些变量搭建一个模型，根据已知的预测变量值计算对应的目标变量值。学习算法的核心在于通过给模型提供对应的目标变量和特征变量的集合（训练集）来重复训练模型，使其在达到理想的输出精度。有监督学习的算法包括：回归模型，决策树，随机森林，K 邻近算法，逻辑回归等。

与监督学习相对应的是无监督学习，此时数据没有类别信息和目标值。无监督学习的终极目标是找出数据存在着的一类隐含规律。具体来说，在无监督学习中，将数据集合分成由类似的对象组成的多个类的过程被称为聚类；将寻找描述数据统计值的过程称之为密度估计。此外，无监督学习还可以减少数据特征的维度，以便我们可以使用二维或三维图形更加直观地展示数据信息。无监督式学习的算法有：关联规则挖掘算法，K-means、DBSCAN 等聚类算法，PCA 降维算法等。

1.2.3 基于机器学习的电力暂态稳定预测方法

近年来，随着基于相量测量装置（Phasor Measurement Unit, PMU）的广域量测系统（Wide-area Measurement, WAMS）的逐步成熟，在电网调度中心实时获取全国同步的细粒度量测数据成为可能，这极大地促进了暂态稳定分析方法和手段的变革，为通过数据进行知识学习的机器学习方法在电力系统稳定预测中的应用提供了契机。

然而，现有的基于在线数据的稳定分析手段相对不足。现有的在线稳定分析是针对确定时间断面的稳定问题展开的，定点定性的成分要多于定量分析，且计算结果在时序和内在联系上相对孤立，缺乏对时变的电网规律的自动发现。在处理手段上，长期以来都是依靠具有一定技术和经验的人员通过大量的计算、分析、协调、找出薄弱点来制定相应的防范措施。在系统规模不大时，这种依赖领域专

家处理信息的模式会取得准确、迅速的结果。但是，在现有的系统规模下，由于缺乏关于电网整体特征的描述，无法根据现有的运行状态预知系统的安全稳定性并做出快速决策。

基于机器学习的电力暂态稳定预测方法的核心方案在于另辟蹊径，从电力系统的运行轨迹信息而非电网理论中提取电网稳定运行的数学边界，从而实现对于未知场景的稳定预测。

一般来说，基于机器学习的方法都要通过构建分类器模型，从离线数据中学习电力系统的稳定运行规律，之后对系统未来是否稳定进行预测。1999 年，刘玉田等人提出了一种新的基于模糊聚类的径向基神经网络，来对电力系统的稳定性进行预测^[27]。2007 年，吴琼等人提出了一种基于最小二乘支持向量机的暂态稳定预测算法，根据分块矩阵求逆的思路对标准算法做出了改进，提高了计算速度^[28]。2013 年，王亚俊等人提出了一种基于响应轨迹和核心向量机的电力系统暂态稳定评估方法，与之前方法相比，所提方法具有更低的时间和空间复杂度，并具有更高的评估精度^[29]。

相对于传统的暂态稳定分析方法，这些机器学习算法在精度和运算复杂度上有一定的改进，为基于响应的电网实时稳定分析和控制方案提供新的思路。然而，这些方法在应用中存在人工提取特征较为困难，需要的数据类型多，预处理复杂等问题，也无法更好地结合利用 PMU 数据本身的时空特性。还有很大的进一步改进的空间。

1.3 本文的主要工作

受深度卷积神经网络在特征学习上的启发，本文提出了一种利用深度卷积神经网络进行电力系统暂态稳定评估的方法。本方法可以自动从 PMU 采集的电压和相角原始数据中获取不同抽象层次的特征信息，因此不需要进行对数据的前期特征处理。具体来说，我们提出了一种有效的处理多维多通道时间序列的深度卷积神经网络模型，每个通道包含一个二维的时间序列，表示电力网络中单个母线的复电压时序信息（由母线电压和相角数据经过简单坐标变换构成）。不同通道的数据都会通过一个共享的深度卷积网络，从而得到其隐含的抽象信息。然后，模型将各个通道经过卷积网络之后的信息输入一个多层感知器（MLP），以此完成故障的判别。我们利用基于梯度下降的方法来确定网络参数，然后在 IEEE39 系统的仿真数据中进行了性能验证。结果表明在同样的样本中，我们的方法在分类精确度上超越了同类算法。

在此基础上，本文尝试应用了最新的激活函数和池化策略，并与传统方法进

行了对比，来分析其在收敛速度的区别。

此外，为了利用现实电网中存在的大量无标签真实数据，本文提出了一种针对此模型的无监督预训练方法，以提高模型精度。本文也对学习到的特征进行了可视化分析，来提供对学习到的特征的直观表示。

1.4 论文结构安排

本论文提出并详细设计了一种基于权值共享的深度卷积网络的电力系统暂态稳定预测模型与系统，并对其进行了实验验证。论文的研究内容和章节安排如下：

第一章对本文的研究目的和意义进行了分析探讨，并简要总结了在相关问题上的国内外研究内容，并描述了本文的主要思路和工作方法。最后，介绍了论文的结构安排。

第二章从课题的需求出发，详细介绍了本文涉及的各项技术的基本原理和实现方案。包括数据来源，人工神经网络与深度学习、硬件加速的人工神经网络等。

第三章是本文工作和内容的核心，从模型架构，理论推导，实现原理，工程方案等诸多角度，详细描述了基于权值共享的深度卷积网络的核心算法实现，包括各层网络定义，前向预测过程，反向传播算法，模型评价指标，以及模型预训练方案等。

第四章对上述模型进行了细致的实验验证工作。本章首先给出了关于实验数据来源，模型参数设置等具体细节的设计方案，然后根据方案对模型进行了实验验证，并对结果进行了分析与评价，证明了模型的可行性，体现了模型的优良性能。最后，该部分给出了卷积网络提供的网络特征的可视化展示。

第五章对整个论文进行总结与展望。该部分总结了论文的主要研究成果和意义，并尝试对下一步的研究方向做出展望。

第二章 电力系统的暂态稳定性理论分析

2.1 电力系统的暂态过程概述

上文中已经提到,电力系统的暂态稳定问题是指电力系统在某个运行情况下突然受到大的冲击之后,能否经过暂态过程重新达到稳定运行状态的问题。本部分将详细分析电网暂态稳定问题的作用机理与表现形式,为之后的分析提供理论基础。

电力系统的暂态稳定问题分为稳定和失稳两种情况,如果系统在受到大的冲击后仍能达到稳态运行,则说明系统在这种运行情况下是暂态稳定的。反之,如果在这种情况下,系统无法再建立稳态运行状态,各发电机组之间一直有相对运动,相对角不断变化,导致整个系统的功率、电流和电压都在不断震荡,整个系统无法继续运行,那么就称系统在这种情况下不能保持暂态稳定^[30]。

电力系统受到大扰动后到逐步趋向稳定运行或者失去同步需要经过一段时间。这段时间的长短也与系统本身的状况和扰动大小有关。从1秒左右到几秒甚至几分钟都有可能。在受到扰动后,不同时间内系统各部分的反应并不一致。一般来说,在分析大扰动后的暂态过程中往往将整个时间段分为以下三个部分^[31]。

起始阶段,指故障后约1s的时间段。在这期间,系统中的保护和自动装置开始进行一系列的动作,如切除故障线路,重新合闸,切除发电机等在这个时间段中,发电机的调节系统还来不及起到明显的作用。

中间阶段,在起始阶段后,大约持续5s左右的时间段。在此期间,发电机组的调节系统发挥了重要的作用。

后期阶段,指中间阶段后的时间。这里动力设备(如锅炉)等中的过程也将影响到电力系统的暂态稳定。此外,由于电网上频率和电压的下降,自动装置可能会切除部分负荷。

一般来说,电力系统的暂态稳定分析集中在前两个阶段。

2.2 暂态稳定的物理过程分析

以下将给出电力系统暂态稳定的一些数学机理分析,在分析中,经常采用一系列有效的假设^[32,33]:

由于发电机级惯性较大,在所研究的短暂时间内各机组的电角速度相对于同步角速度变化不大。因此,在分析系统的暂态稳定性时,假定网络中频率依然是

50HZ

忽略突然发生故障后网络中的非周期分量电流。一方面,这些电流衰减较快。另一方面,非周期电流产生的磁场在空间中不发生改变,它和发电机转子的绕组产生的磁场相互作用将产生以同步频率交变的平均值接近 0 的制动转矩,因此对发电机的机电暂态过程影响很小,可以忽略不计。

当故障不对称时,发电机定子回路中将流过电力负序电流,这些电流产生的磁场和转子绕组电流的磁场间形成的转矩主要以两位同步频率进行交变,其平均值也接近于 0,因此对电力系统的暂态过程也没有明显影响,可以略去不计。

根据以上假定,在进行电力系统的暂态过程分析时,可以只考虑频率为 50Hz 的电流和电压分量,这样,就依然可以用代数的形式来对电力系统进行描述。

功率特性分析

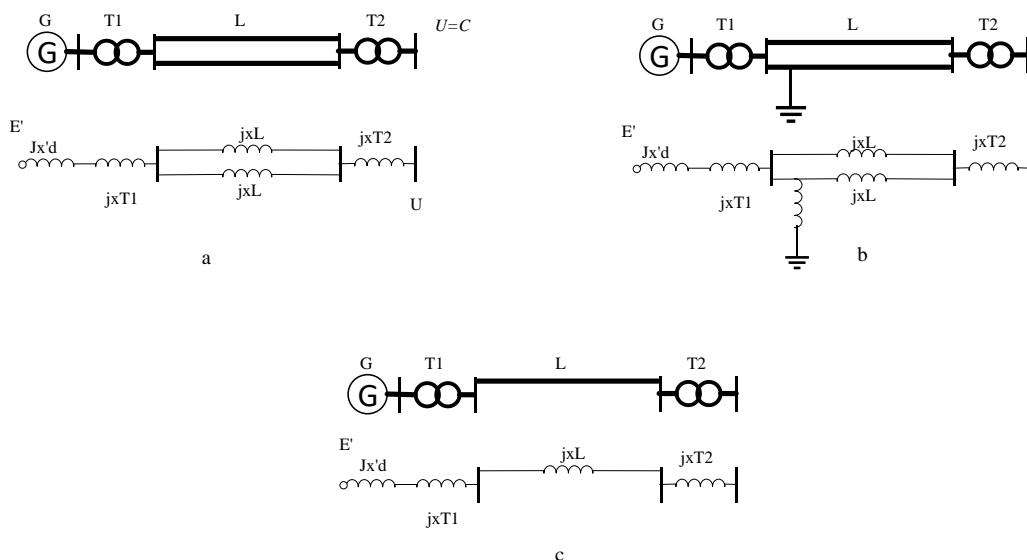


图 2-1 简单电力系统的暂态过程

图 2-1 中 a 表示一个简单电力系统,在其正常运行时,发电机经过变压器和双向线路向无限大系统送电。如果 E' 表示发电机等值电动势,那么 E' 与无限大系统间的电抗可以表示为如下公式的形式:

$$x_1 = x'_d + x_{T1} + \frac{x_L}{2} + x_{T2} \quad (2-1)$$

此时,发电机的电磁功率为:

$$P_1 = \frac{E'U}{x_1} \sin \delta \quad (2-2)$$

如果突然在一个输电线路的始端发电不对称短路,如图 2-1 中 b 所示,则电路可以等效为 b 图下半部分的形式,即在正序网络的故障点上附加一个电抗 jx_Δ 。附加电抗的大小由不对称故障和故障点的相关情况计算而来。这时,发电机电动

势和无限大系统之间的联系电抗可以将 b 图的星形网络转化为三角网络而得：

$$x_2 = (x'_d + x_{T1}) + \left(\frac{x_L}{2} + x_{T2}\right) + \frac{(x'_0 + x_{T1})\left(\frac{x_L}{2} + x_{T2}\right)}{x_\Delta} \quad (2-3)$$

从公式中可以看出，这个电抗会大于正常运行时的电抗。如果故障是三相短路，那么 x_Δ 为 0， x_2 为无限大。这表示三相短路切断了发电机和系统间的联系。

在故障情况下，发电机的输出功率为

$$P_2 = \frac{E'U}{x_2} \sin\delta \quad (2-4)$$

在三相短路发生时，可以看出，发电机的输出功率为 0。

在短路故障发生后，线路继电保护装置会迅速断开独步线路两端的断路器，如图 c 所示。此时，发电机的电动势和无限大系统间的联系电抗为：

$$x_3 = x'_d + x_{T1} + x_L + x_{T2} \quad (2-5)$$

发电机输出功率为：

$$P_3 = \frac{E'U}{x_3} \sin\delta \quad (2-6)$$

扰动前后的运行情况

正常运行方式。如果正常运行时发电机向无限大系统输送的功率为 P_0 ，则原动机输出的机械功率 p_T 等于 P_0 。此时功角为 δ_0 。

故障阶段。发生短路后功率特性立即降为 P_2 ，但由于转子的惯性，转子的角度不会立即发生变化，其相对于无限大的系统母线的角度 δ_0 仍保持不变。因此发电机的输出功率显著减小，而原动机的机械功率 P_T 不变，故产生大量的过剩功率。故障越严重， P_2 功率曲线幅值越低（三相短路时为零），过剩功率越大。在过剩转矩的作用下发电机转子将加速，其相对速度和相对角度 δ 逐渐增大。如果故障一直存在，那么发电机将不断加速，最终与无限大系统失去同步。

故障及时切除的情况。在故障切除后，发电机的功率特性变为 P_3 。发电机的输出功率比原动机的机械功率大，使转子的转动受到制动，转子速度逐渐减慢。但由于此时的速度依然大于同步转速，所以相对角度仍会进一步增大。在转子的速度逐渐减至和同步转速一致时，机械功率和电磁功率仍不平衡，转子仍会继续减速，但相对角度开始减小。最终当达到机械功率和电磁功率的平衡点时，转子又开始加速，相对角度继续减小。当转子速度再次达到同步转速时，相对角度恢复到 δ_0 ，但这时机械功率仍不平衡，因此，系统会一直这样振荡下去。当然，在实际中，在振荡过程中伴随有能量损耗，因而振荡会逐渐衰减，发电机最后会停留在一个新的运行点。

故障切除过晚的情况。如果故障切除过晚，那么此时故障线路切除前转子加

速已经比较严重。当故障线路切除后，电磁功率和机械功率的差距不足以将转子速度降至同步转速附近，相对角度会呈现出一直增大的趋势，发电机和无限大系统最终将失去同步。

以上定性的简要介绍了电力系统中暂态稳定问题的基本物理过程。在现实中，由于受到多种因素的影响，电力系统的暂态过程远比上述描述中的复杂，但遵循的基本原理是一致的。

第三章 人工神经网络理论

3.1 人工神经网络

我们使用深度卷积神经网络来处理上述问题。深度卷积网络是深度神经网络的一种。神经网络由感知器（神经元）按照特定的结构组合而成，采用梯度下降的算法进行模型训练。卷积网络是一种特殊的神经网络。下面进行简要说明其理论基础。

3.1.1 多层感知器网络

人工神经元模型

人工神经元模型是人工神经网络的基本组成部分，它可以模拟神经元信号的传递原理。神经元（Neuron），或者说神经细胞（Nerve Cell），是神经系统结构与功能的基本单位。它的基本构造包括树突、轴突、髓鞘、细胞核等。一个神经元有大量的树突和少数轴突，其树突和其它神经元的轴突通过突触相连接。突触有化学物质（如多巴胺、乙酰胆碱等）进行信息的传递。神经元在树突受到特定的刺激后，会在轴突端产生放电信号。这些放电信息又会通过突触被神经网络中其它神经元的树突接收，引起网络的连锁反应^[34]。

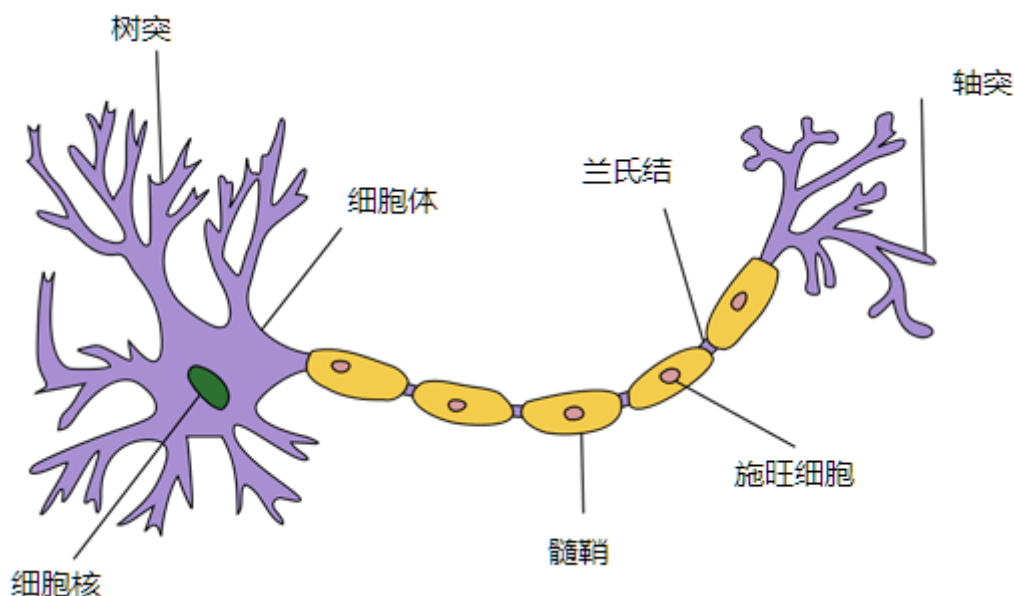


图 2-2 神经元 引自维基百科

人工神经元模型具有类似的特性。它首先将输入分量 x_i 分别与自己对应权值 w_i 相乘，来模拟神经元突触的传递，然后将相乘之后的值求和，并与偏置相加，

输入到激活函数获取到神经元的输出。公式表示为

$$y = \sigma(\mathbf{w}^T \mathbf{x} + b) \quad (2-7)$$

其中， σ 表示激活函数，其特殊的函数特点最初用于模拟神经元的兴奋和抑制两种状态，如果函数的输出值接近 0，就表示这个神经元被抑制，如果函数的输出值接近 1，则说明函数处理兴奋(激活)状态。经典的激活函数有两种，sigmoid 和 tanh，其公式如下：

$$\text{sigmoid}(x) = \frac{1}{1+e^{-x}} \quad (2-8)$$

$$\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2-9)$$

函数绘制如下：

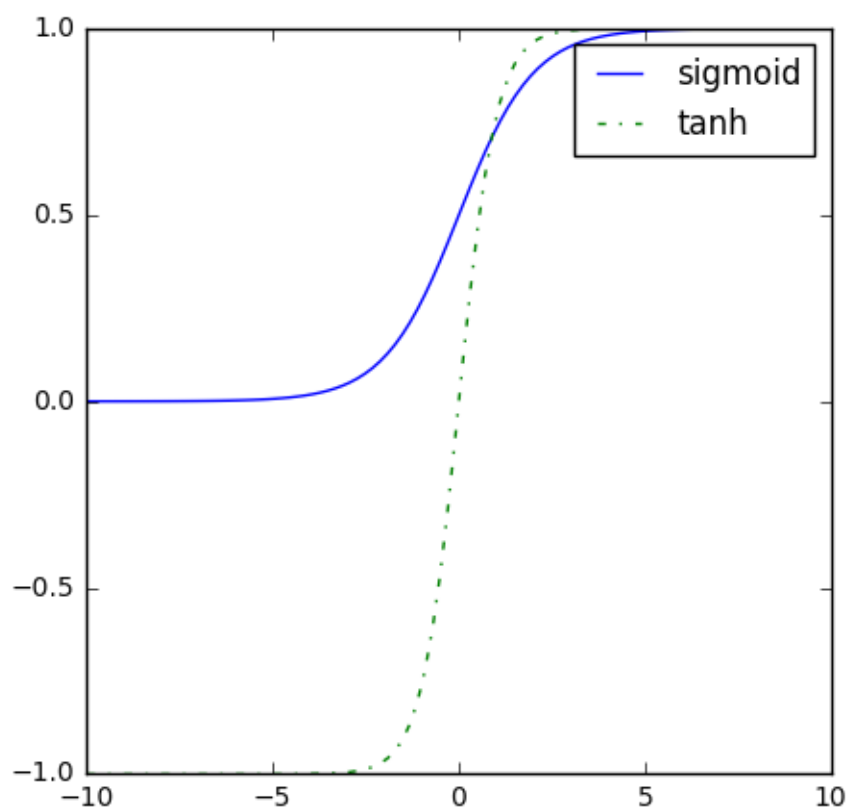


图 2-3 Sigmoid 和 tanh 函数的对比

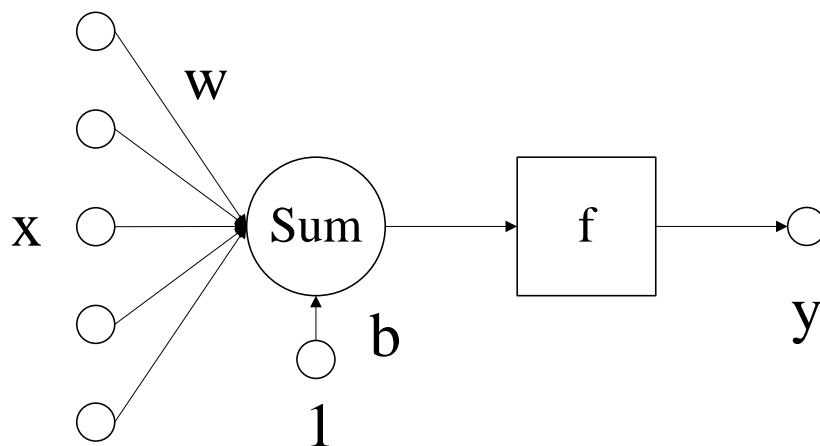


图 2-4 人工神经元模型

单个神经元就可以解决一些简单的模式分类问题。如果一个问题线性可分的，那么就可以在其输入空间 x 上找出一个超平面将两者分割为左右两部分。神经元的权值向量 W 就表示这个超平面的参数。

多层感知器

将多个感知器“神经元”依据某种规则按层次顺序连接起来，前一层的神经元输出作为下一层所有神经元的输入，为构成了多层感知器。它是一种简单的人工神经网络。通过对线性可分问题的特定组合，多层感知器可以用来解决复杂的非线性可分的问题。

输入层接受外界信息，并且与中间的隐藏层相连，将信息传递个隐藏层的每个神经元。隐藏层主要对输入的信息按要求进行变换，是网络的内部处理层，可以是一层或者多层。输出层是网络与外界的连接层，将最后一层隐藏层传递来的信息经过处理后输出给外界。每层的前向传播公式如下：

$$a^l = \sigma(W^{l-1}a^{l-1} + b^l) \quad (2-10)$$

其中， L 表示网络层数， $l \in [1, L]$ ， a^l 表示 l 层的激活值。 W^l 是权值矩阵， W_{ij}^l 表示第 l 层第 j 个神经元和第 $l+1$ 层第 i 个神经元之间的连接权值。 b^l 表示第 l 层的偏置。

多层感知器的训练

反向传播算法是一种训练人工神经网络的算法^[35]。网络首先进行前向计算，得出输出值。当实际的输出与预期的存在差距时，令误差从最末尾的输出层开始，逐层进行误差传播，将误差作为网络的梯度，使用梯度下降的算法来更新各层的权值。模型不断地进行反向传播和前向传播，直至网络的实际输出与预期之间的

误差平方值收敛，或达到一定的学习次数时，则停止学习。

下面是简要算法描述：

输入： 样本集合 D , 学习率 α

输出： 网络的分类结果 y_{out}

算法步骤：

网络初始化：

构建具有 n_{in} 个输入， L 个隐层，每层 n_{hidden} 个隐藏单元， n_{out} 个输出单元的神经网络。使用随机数初始化网络中所有的权值。

前向传播：

对于大小为 m 的训练样本集合 D 中的每个样本 (x, y_0) ，把输入向量输入 x 网络，计算网络中每个神经元的输出。对于网络中第 l 层的神经元 i ，有：

$$z_i^l = \sum_j w_{ji}^l x_j^{l-1} + b \quad (2-11)$$

$$x_i^l = \sigma(z_i^l) \quad (2-12)$$

反向传播：

计算 m 个样本的总误差

$$E = \frac{1}{m} \sum_{i=1}^m Cost(h_W(x), y) \quad (2-13)$$

$\forall k \in \text{output}$, 求 k 的误差项 δ_k

$$\delta_k = \frac{\partial E}{\partial x} \cdot \sigma'(z_k) \quad (2-14)$$

计算隐层的误差项 δ_i^l

$$\delta_i^l = \sum_k w_{ik}^{l+1} \delta_k^{l+1} \sigma'(z_i^l) \quad (2-15)$$

更新网络的权值 w_{ij}

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \quad (2-16)$$

其中

$$\Delta w_{ij} = \alpha \delta_j x_{ij} \quad (2-17)$$

3.1.2 卷积神经网络

概念

卷积神经网络(Convolutional Neural Networks, CNN)是一种特殊的神经网络模型，它的神经元具有局部连接性并且这些神经元之间的权值是共享的。这使得模型对平移、比例缩放、倾斜和其他形式的变形具有高度的不变性。1989年 Y.le

Cun 首次提出了一个简单的 4 层卷积神经网络 LeNet-5，它在 Mnist 数据集上获得了前所未有的识别率。随后各种网络层数更多学习能力更强的卷积神经网络模型在文本识别、语言识别、人脸识别、物体检测等领域得到了广泛应用。典型的卷积神经网络包括卷积层与池化层两点网络结构。

卷积层通过类似连续扫视的方式来处理输入数据，这相当于数学上的卷积操作。在典型的图像处理领域，考虑一个大小为 5×5 的图像，和一个 3×3 的卷积核。这里的卷积核共有 9 个参数，可以表示为矩阵的形式 $W = [w_{ij}]_{3 \times 3}$ 。这种情况下，卷积核实际上有 9 个神经元，他们的输出又组成一个 3×3 的矩阵，称为特征映射。第一个神经元连接到图像的的第一个 3×3 的局部，第二个神经元则连接到第二个局部。

卷积层的操作可以表示为如下公式的形式：

$$h_{ij} = \sigma((W * x)_{ij} + b) \quad (2-18)$$

其中， $*$ 表示卷积操作。

$$(W * x)_{ij} = \sum_u \sum_v W_{uv} x_{(i-u)(j-v)} \quad (2-19)$$

池化层进行池化操作。池化(下采样)的目的是减小特征映射的维度。常用的池化方式有最大池化，均值池化等。具体做法是将特征映射分为重叠或不重叠的多个大小相等部分，然后根据池化方式的不同，对每一部分分为进行池化操作。对于最大池化而言，就是对每一部分求解其最大值。

卷积网络的训练

卷积神经网络的参数估计依旧使用反向传播的方法，不过需要针对卷积神经网络的特点进行一些修改。

卷积层的反向传播过程有两个任务，一是更新权值，另一是反传残差。

对于多层感知器而言，其每个神经元的梯度公式为

$$\frac{\partial E}{\partial w_{ji}} = \delta_j o_i \quad (2-20)$$

其中， E 为目标函数， w 为权值， δ 为残差， o 为神经元的输出。

对于卷积层而言，因为局部感受野和权值共享的关系，相当于使用滑动窗口的方式对卷积核的权值进行更新。这依然可以被视为卷积操作。即

$$\frac{\partial E}{\partial w} = \text{rot180}(O * \text{rot180}(\Delta)) \quad (2-21a)$$

$$\frac{\partial E}{\partial b} = \sum_{uv} \delta_{uv} \quad (2-21b)$$

其中 rot180 是将一个矩阵反转 180° ， Δ 为残差矩阵。偏置的梯度为残差矩

阵所有元素的和。

残差的传播同样是一个卷积过程，公式描述如下：

$$\Delta' = (\Delta * \text{rot180}(W)) \circ \sigma(O) \quad (2-22)$$

3.1.3 现代人工神经网络

梯度下降算法的改进

在训练人工神经网络时，我们通常采用反向传播算法求梯度，结合梯度下降法来对模型进行训练。假设我们的目标函数为 $f(w)$ ，梯度下降就是采用迭代的策略，从初始点开始，每次沿着目标函数在当前点的负梯度方向前进一定的步长。即：

$$w_{t+1} = w_t - \eta_t \nabla f(w_t) \quad (2-23)$$

在机器学习中，目标函数往往表现为样本误差之和的形式。即：

$$f(w) = \sum_{i=1}^M f(w, x_i) \quad (2-24)$$

M 表示样本数量。算法在更新每一次参数时都使用所有的样本进行计算损失函数及偏导。这样，在样本数据很多时，训练过程会非常缓慢。此外，梯度下降算法在某些情况下收敛速度较慢。

随机梯度下降算法

随机梯度下降算法（SGD）是梯度下降的 Batch 版本^[36]。对于训练数据集，我们首先将其分成 n 个 batch，每个 batch 包含 m 个样本。我们每次更新都利用一个 batch 的数据，而非整个训练集。

随机梯度下降用梯度的无偏估计 $g(w)$ 来代替梯度，即

$$w_{t+1} = w_t - \eta_t g(w_t) \quad (2-25)$$

$$E[g(w_t)|w_t] = \nabla f(w) \quad (2-26)$$

此时， $f(w)$ 依然期望收敛到最小值。

当训练数据太多时，利用整个数据集更新往往时间上不现实。batch 的方法可以减少机器的压力，并且可以更快地收敛。

当训练集有很多冗余时（类似的样本出现多次），batch 方法收敛更快。以一个极端情况为例，若训练集前一半和后一半梯度相同。那么如果前一半作为一个 batch，后一半作为另一个 batch，那么在一次遍历训练集时，batch 的方法向最优解前进两个 step，而整体的方法只前进一个 step。

Adam 算法

Adam 算法的中文名字是自适应矩估计（Adaptive Moment Estimation）。矩是

一个概率中的概念。如果一个随机变量 X 服从某个分布， X 的一阶矩 $E(X)$ 就是样本平均值， X 的二阶矩 $E(X^2)$ 表示样本平方的平均值。

Adam 算法根据损失函数对每个参数的梯度的—阶矩估计和二阶矩估计动态调整针对于每个参数的学习速率。**Adam** 也是基于梯度下降的方法，但是每次迭代参数的学习步长都有一个确定的范围，不会因为很大的梯度导致很大的学习步长，参数的值比较稳定。同时，它有利于模型跳出局部最优点，降低模型收敛到局部最优的风险^[37]。

其过程可以表示为如下公式：

$$m = \beta_1 m + (1 - \beta_1) \Delta x \quad (2-27)$$

$$v = \beta_2 v + (1 - \beta_2) \Delta x^2 \quad (2-28)$$

$$x = x - \frac{\alpha m}{\sqrt{v} + \epsilon} \quad (2-29)$$

其中， α 表示学习率， $\beta_1, \beta_2, \epsilon$ 为超参数。一般情况下， $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e - 6$ 。

激活函数的改进

在传统神经网络中，Sigmoid 系（Logistic-Sigmoid、Tanh-Sigmoid）激活函数被视为神经网络的核心所在。从数学上来看，非线性的 Sigmoid 函数对中央区的信号增益较大，对两侧区的信号增益小，在信号的特征空间映射上，有很好的效果。

然而，Sigmoid 系函数还受梯度消失问题的影响。必须通过网络逐层预训练等方式来克服这一问题。在反向传播的过程中，误差从输出层反向传播计算梯度。这需要在各层乘当前层的输入神经元值与激活函数的一阶导数。

$$Grad = Error \cdot \frac{dSigmoid(x)}{dx} \cdot x \quad (2-30)$$

这会导致两个问题。

导数缩放：

$$\frac{dSigmoid(x)}{dx} \in (0,1) \quad (2-31)$$

饱和值缩放： $x \in (0,1)$

从而，经过每一层时，Error 都是成倍的衰减，一旦进行递推式的多层的反向传播，梯度就会不停的衰减，消失，使得网络学习变慢。

此外，Sigmoid 系函数还存在稀疏激活性不足的问题。2001 年，Attwell 等人基于大脑能量消耗的观察学习上，推测神经元编码工作方式具有稀疏性和分布性^[38]。2003 年 Lennie 等人估测大脑同时被激活的神经元只有 1~4%，进一步表

明神经元工作的稀疏性^[39]。从信号方面来看，即神经元同时只对输入信号的少部分选择性响应，大量信号被刻意的屏蔽了，这样可以提高学习的精度，更好更快地提取稀疏特征。从这个角度来看，在使用传统的 Sigmoid 系函数时，神经网络中同时近乎有一半的神经元被激活，这并不符合神经科学的研究。

ReLU 函数较好地解决了上述问题，它可以表示为如下公式的形式^[40]：

$$f(x) = \max(0, x) \quad (2-32)$$

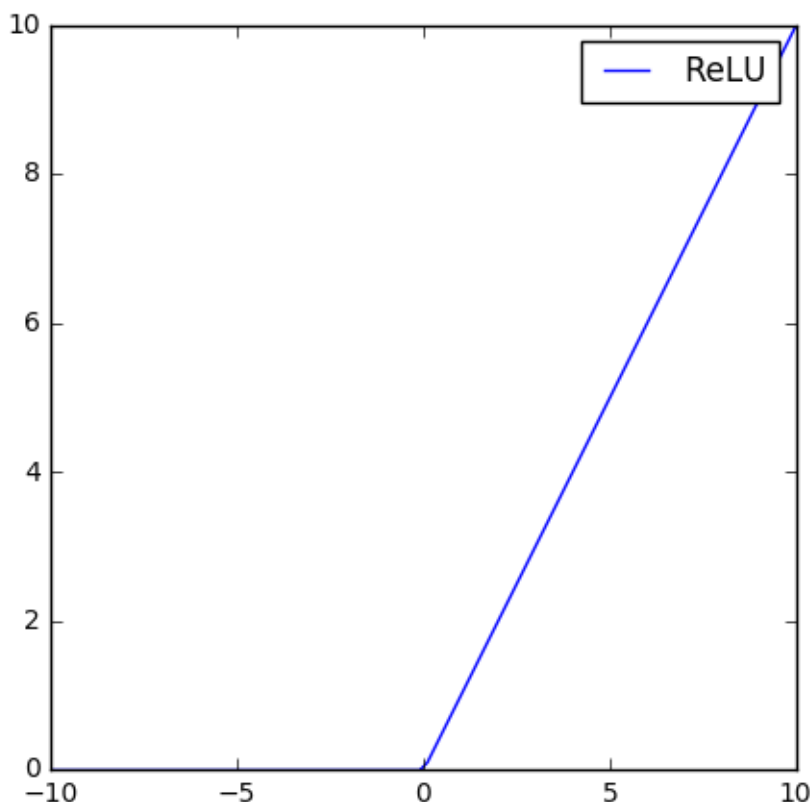


图 2-5 ReLU 激活函数

ReLU 函数可以绘制为图 2-5 的形式。相对于传统的 Sigmoid 系激活函数，它的抑制范围更广，满足了网络的稀疏激活性；同时，其导数为 1 或 0，不会随着网络的加深而消失。使用 ReLU 作为非线性激活函数使得深度网络即使不经过预训练过程也可以较快地收敛，在处理大而复杂的问题时，相比于 sigmoid 函数等更有效率。

此外，神经生物学证明，其单边的激活方式更具有更加可靠的生物合理性。且只需要比较和乘加运算，运算速度更快。

深度学习

深度学习（英语：deep learning）是机器学习拉出的分支，它试图使用包含复杂结构或由多重非线性变换构成的多个处理层对数据进行高层抽象的算法。

传统的浅层学习模型的学习表达能力有限，有处理复杂的分类问题时，往往需要人工进行很多的特征提取工作，而且其泛化能力较差。相对而言，深度神经网络中增加的隐含层能更好的完成对特征的学习，为模型提供了更高的抽象层次，因而提高了模型的整体性能^[41]。

因此，深度学习可以看作是机器学习中特征学习方法。观测值（例如一幅图像）可以使用多种方式来表示，如每个像素强度值的矢量，或者更抽象地表示成一系列边、特定形状的区域等。而使用某些特定的表示方法更容易从实例中学习任务（例如，人脸识别或面部表情识别^[42]）。深度学习的主要优势在于使用非监督式或半监督式的特征学习和分层特征提取的高效算法来尝试替代人工特征提取过程^[43]。

深度神经网络实际上是由多层人工神经元组构成，总的来说，深度神经网络的每一层都是由前一层的所有节点求和后通过分类器或回归得到不同的节点所构成的，深度神经网络的参数学习则是通过误差逐层反向传播来修改相对应的权重完成的。对于深度神经网络，也可以分解为自底向上的输入沿网络前向传播阶段和自顶向下的误差沿网络反向传播阶段两个阶段。

自深度学习出现以来，它已经在很多领域成为各种领先系统的一部分，尤其是在计算机视觉和语音识别中。在通用的用于检验的数据集，例如语音识别中的 TIMIT 和图像识别中的 ImageNet, Cifar10 上的实验证明，深度学习能够可观地提高识别的精度。

3.1.4 深度神经网络的预训练

深度神经网络的一大挑战是其损失函数的非凸性。因此需要在训练过程中避免收敛到较差的局部最优点。之前一些工作证明了，贪婪逐层预训练在解决这个问题上有很好的效果。本部分将首先介绍全连接网络的预训练方案，然后再给出卷积网络的预训练过程。

自编码器

全连接网络的预训练主要通过自编码器^[44]实现。自编码器是一个三层的人工神经网络，与传统的 MLP 非常相似。它是一种无监督的学习算法，使用反向传播算法来使目标值等于输入值。自编码器的目标是学习原始输入特征的降维表示。这些降维后的特征可以被用来整合入更大的神经网络当中。

自编码器的工作方式可以用如下公式表示。令 x 为网络输入， y 为网络输出， h 为隐含层激活。那么，有：

$$h = \phi(Wx + b) \quad (2-33)$$

$$y = \phi(W'h + b') \quad (2-34)$$

其中,有约束:

$$W' = W^T \quad (2-35)$$

其损失函数为:

$$(\theta) = ||x - y||^2 \quad (2-36)$$

一种全连接网络的预训练方法就是逐层训练自编码器并使其相互叠加。具体来说,就是将利用上一层网络的输出来训练自编码器,然后把训练好的自编码器的隐含层加入网络中,作为新的最后一层。

卷积自编码器

普通的自编码器只能应用于全连接网络,对于卷积网络而言并不适用。卷积网络的自编码应该被视为卷积运算和其逆运算的组合,然后再使用反向传播的方式进行训练,达到输入值近似等于输出值的目的^[45]。

用公式表示,有:

$$h_j = \phi(z_j) = \phi(\sum_i \text{pad}(x_i) * k_{ij} + b_j) \quad (2-37)$$

其中, x_i 表示第 i 个输入。 h_j 表示第 j 个卷积核的输出。 pad 函数的作用是使 x_i 两端添加 0 值,使其大小可以被卷积核大小整除。

之后,再利用一个与上一步卷积核相关的函数,使得 h 得以反向转换回 x :

$$y_i = \phi(\sum_j h_j * \text{reverse}(k_{ij}) + c_i) \quad (2-38)$$

其中 reverse 函数用于反转卷积核。 c_i 是偏置项。

卷积网络的自编码器和传统自编码器的损失函数是一致的,即:

$$E(\theta) = \sum_i ||x_i - y_i||^2 \quad (2-39)$$

3.2 硬件加速的人工神经网络计算

人工神经网络的计算可以使用商用的图形显示卡上的图形处理单元 (GPU) 来进行。相对于使用 CPU 的计算方式, GPU 在性能,功耗和成本上有明显优势。

图形处理器的概念由英伟达公司在 1999 年 8 月发布 GeForce 256 绘图处理芯片时率先提出。图形处理器使显卡减少了对中央处理器的依赖,并分担了部分原来属于 CPU 的工作。随着显卡的发展, GPU 越来越强大,在浮点计算上凭借流处理单元的支持,远远超越了同等级的 CPU。现有的 GPU 往往具有成百上千个流处理单元,通过在流处理单元间高度并行计算的方式,可以达到几百到几

千 GFlops 的运算能力。此外，显示芯片通常具有更大的内存带宽。

神经网络的核心算法在数学上属于矩阵相乘或傅立叶变换等线性代数问题。而这些问题的计算过程中具有大量上下文无关的循环过程，可以较为容易地展开成可以被细粒度并行执行的形式。

现有的人工神经网络实现框架包括 Caffe、CNTK、MXNet、Neon、TensorFlow、Theano 和 Torch 等。这些实现框架有其各自的优劣和适用场景。本文借助 Tensorflow 及其封装实现 Keras 完成深度神经网络的工程实现工作。Tensorflow 是谷歌研发的第二代人工智能学习系统，它将复杂的计算抽象成数据流图的形式，并通过领域特定语言（DSL Domain Specific Language）来对其进行声明式地构建和控制。其主要优势在于较好的灵活性和可延展性。Keras 是一个高阶的神经网络框架，使用 Python 开发，可以将 Tensorflow 作为其运行内核。Keras 主要用于快速的实验与开发工作，这对于研究性质的工作非常重要。

Tensorflow 借助现有的高度优化过的数值运算库完成底层的运算操作。比方说，它使用高度优化过的基础线性代数库如 Blas，cuBLAS 等完成大矩阵的相乘等操作，使用 cuda-convnet 和 cuDNN 等完成深度卷积网络中卷积计算的实现。这些线性代数库大都可以借助 Nvidia 公司的 Cuda（Compute Unified Device Architecture，统一计算设备架构）编程模型利用 GPU 加速运行。

本小节将主要从 Cuda、Tensorflow 与 Keras 四个方面简要介绍硬件加速的人工神经网络计算方法。

3.2.1 Cuda

Cuda 编程模型

Cuda 是指 Compute Unified Device Architecture（统一计算设备架构）。由 Nvidia 公司提出，可以在 Nvidia 生产的 GPU 中进行大规模的并行计算。CUDA 程序由 C 语言的扩展版本编写，通过设计资源分配和数值计算流程将运算工作分配到数千个线程上，并在成百上千个核心上并行运行。借助 Cuda 技术，开发人员能够利用 GPU 解决非常复杂的并行计算问题。

在 CUDA 编程模型中，GPU 被视为 CPU 的协处理器，CPU 主要负责协调控制，GPU 主要负责计算密集型任务。这种编程模型将程序分为 Host 和 Device 两部分。Host 程序在 CPU 上运行，主要负责调用和控制 GPU 上的程序，也用于读写数据，管理内存，调用外部函数等。Device 就是指在 GPU 上执行的部分。GPU 有其独立的计算核心和专属内存（显存），Host 需要将运算数据复制到显存中，之后才能调用 Device 程序对其加以处理。

在 CUDA 架构中，GPU 执行的最小单位是线程（thread），数个线程可以组

成一个线程块 (block)，执行相同程序的 block 还可以组成线程网络 (Grid)。在 Device 中执行的程序被称作内核 (Kernel)。通常有成百上千的线程执行同一个内核。GPU 中的线程块，是可以协调工作的一批线程的组合。这些线程之间通过显存共享数据，并同步地进行执行来协调内存的访问。线程之间可以实现细粒度的并行，块之间可以实现粗粒度的并行^[46]。

在内核启动时，会指定线程风格和块的排列方式，每个线程和块按照排列顺序有其独立的编号。通过这个过程，不同的数据可以被分配到不同的线程上进行处理。

图形显示卡的硬件

图形显示卡的硬件由许多个流多处理器 (Streaming Multiprocessors, SMs) 和全局内存 (global Memory) 构成。每个 SM 中包含数个标量处理单元 (Scalar Processors, SPs)、共享内存 (Shared Memory) 等单元。Cuda 程序中的一个线程块会分配到一个 SM 上进行执行，线程块中的每个线程都在这个 SM 中的 SP 上运行，因此这些线程可以共享相同的内存区段，也可以执行同步 (Synchronized) 的指令。

图形显示卡中的内存有很多种，全局内存由各个线程块共用，也可以被 host 程序所访问。每个 SM 上的共享内存由线程块内部共用。还有让每个线程进行存取数据的寄存器 (Register)。寄存器的速度最快。通过特殊的存取方式，共享内存存在特殊的存取方式下，可以达到和寄存器类似速度。这主要是通过并行化执行的方式来实现的，当第一个线程需要等待内存读取结果时，则开始执行第二个线程，并以此类推。全局内存的访问速度最慢，只有共享内存的百分之一。

CUDA 科学计算

为了方便科学计算，CUDA 将常用的数学函数，如 sin, exp 等包装进了编译环境中。也有内建的数据函数可以使用，这些函数直接对应 GPU 上的指令集，具有非常高的运算效率。CUDA 还提供了用于处理向量计算的基础线性代数库，数字信号处理，FFT 等。

CUDA 进行科学计算的缺点在于其精度有限，它的双精度运算能力只有单精度运算能力的几分之一。然后，对于人工神经网络而言，其特性决定了并不需要很高的精度，单精度的浮点数字就足以满足大部分情况的需求。

3.2.2 Tensorflow

Tensorflow 是一个用于表示机器算法的接口与实现，该系统非常灵活，可以用来表示各式各样的算法，并在各种各样的异构系统上进行运行。Tensorflow 主

要用于深度神经网络模型的训练和运行，并在多个科研和生产领域得以应用，包括语音识别，计算机视觉，机器人，信息检索，自然语言处理，地理信息提取和药物发现等。本部分将主要介绍 Tensorflow 的主要编程模型和实现^[47]。

Tensorflow 编程模型

Tensorflow 的计算可以被描述为有向图的形式。这些有向图由一系列运算节点组成，用于表示数据流的计算过程。有向图中也包括一些用于保持和更新持久状态的扩展，以及分支与循环控制结构的支持。客户端使用支持的前端语言（C++ 或 Python）来构建程序流图。

在 Tensorflow 图中，一个节点表示一种操作，它可以包含若干个输入和输出边。在边上流动的数据被称为张量（Tensor），它是由相同类型的数值构成的任意维向量。有向图中也可以有一些特殊的边，用于依赖控制：其终止节点的计算任务必需在起始节点的计算全部完成后才能进行。Tensorflow 的模型支持可变的状态，因此，依赖控制边可以用来指定事件的严格先后顺序。依赖控制边也可以用来控制同时执行的节点数量，并以此来限制峰值内存占用。

Tensorflow 程序通过“会话”（Session）来与 Tensorflow 核心进行交互。Session 接口使用 Extend 方法来为现有的运算图添加边和节点，并通过 Run 接口来实现计算的运行。

在大部分情况下，Tensorflow 图都会被执行多次，而大部分张量的生命周期都是一次执行。Tensorflow 引入了变量（Variable）的概念来保持数据。变量是一种特殊的操作，它可以返回一个持续存在的可变张量来保存数据。变量可以被用来赋值或更新。在 Tensorflow 中，变量的典型应用是在机器学习模型中保存参数，这些参数会在 Run 过程中被动态更新。

Tensorflow 的实现细节

Tensorflow 系统的主要组成部分是客户端（Client），服务端（Master）与一个或多个工人进程（Worker Process）。客户端使用 Session 接口与服务端通信。服务端指定工人进程的工作，这些工人进程可以访问一个或多个计算设备（如 GPU）。Tensorflow 同时支持本地与分布式的实现。在本地模式中，客户端，服务器和工人进程同处于一台机器，在分布式实现中，这些模块可以同时位于不同的硬件机器上。本文使用本地单设备的方式使用 Tensorflow。

在 Tensorflow 中，张量使用有类型的多维向量实现。这些类型包括有符号和无符号整数，IEEE 单精度和双精度浮点数，复数，字符串等。在不同的设备中，这些多维向量可能被保存为不同的形式。张量使用引用计数的方式进行垃圾回收工作。

在本地单设备的执行过程中，只存在一个工人进程和一个设备。计算流图中的节点按照依赖顺序进行执行。具体来说，Tensorflow 会记录每个节点未完成的引用数量。当这一数值降至 0 时，这个节点就会被加入就绪队列等待执行。GPU 会随机从就绪队列中选择节点进行执行。一旦一个节点已经被执行完成，所有依赖这个节点的其他节点的未完成引用数量都会减 1。

3.2.3 Keras

本文中提到的模型使用基于 Python 的深度学习工具 Keras 实现^[48]。Keras 完成了对人工神经网络常用模型计算过程的简单封装，可以在 Tensorflow 或 Theano 上运行。在本文中，我们使用 Tensorflow 作为 Keras 的张量运算库。Keras 的主要优势在于支持简单快速的原型开发工作。

Keras 采用了模块化的实现方案，将一个人工神经网络模型简化为一个由众多可配置的计算模块相互连接而成的序列或图的形式。这些计算模块可以是单层的神经网络，也可以损失函数、优化方法、初始化策略、激活函数或正则化策略等。通过按照一定的规则组合这些模块，可以轻松构建出想要的模型。

Keras 有两种编程模型，序列式（Sequential）模型，或函数式（Functional）模型。序列式模型比较简单，用来表示一系列模块的线性组合。函数式模型相对复杂，可以用来表示非常复杂的模型，如多输入多输出模型、带环模型或权值共享的模型。本文使用函数式模型来完成工程实现工作。

在函数式模型中，所有的模块都可以被表示为可配置的函数的形式。模块本身的参数由构造函数指定。模块本身的输入参数为预想的输入张量，返回值为输出张量。下面的代码表示一个由三层 MLP 构成的简单神经网络，它前两层每层有 64 个节点，激活函数是 ReLu。最后一层有 10 个节点，激活函数是 softmax。

```
# this returns a tensor

inputs = Input(shape=(784,))

x = Dense(64, activation='relu')(inputs)
x = Dense(64, activation='relu')(x)
predictions = Dense(10, activation='softmax')(x)
```

3.3 小结

本章首先简要介绍了之后使用的定义与符号，并给出了论文解决问题的正式

描述。本论文使用人工神经网络的方法来解决电力网络暂态稳定预测问题，本章结合人工神经网络的研究内容与现状，简要性地给出了人工神经网络与深度学习的理论基础，以及硬件加速的人工神经网络实现方案，为下文的工作作出了铺垫。

第四章 基于深度学习的暂态稳定预测方案

在本部分中，我们将介绍一种进行在线暂态稳定预测的深度卷积网络模型。本模型包含坐标变换层、卷积网络和多层感知器三个部分。

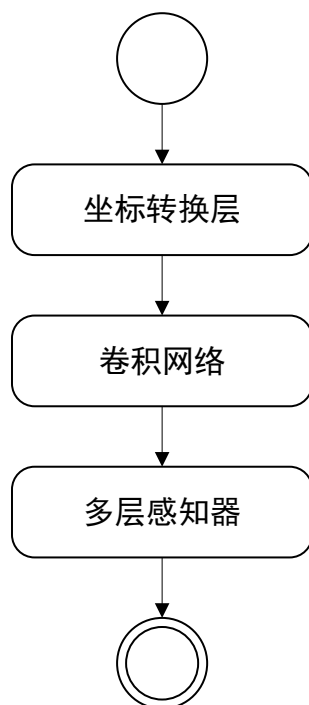


图 3-1 模型结构

坐标变换层的主要作用是将输入的电压和相角数据转换为易于被卷积网络处理的形式。之后的卷积网络用于学习时间序列的层次化特征描述。最后，这种特征描述被引入到一个多层感知器中，来学习揭示这些特征与未来系统稳定性的关系。

模型可以通过反向传播的方式进行训练。在训练过程中，时间序列和其对应的标注分组进入模型，对于每一组数据，系统基于当前的模型参数，从时间序列中得出预测值。然后计算预测值和真实标注之间的误差，并将其反向传播来更新模型参数。最终不断迭代至收敛。

值得注意的是，在电力网络中，不同的母线节点都服从同样的物理定律，那么我们可以假设从其上采集到的 PMU 电压和相角数据具有相似的数学特征。因此，我们可以使用共享权值的多个卷积网络来处理各个节点的 PMU 数据，这样可以提高模型的泛化程度，加强面对特征微小改变时的鲁棒性，减少所需参数数量，降低训练时间。

本部分剩下的篇幅将首先用于介绍系统架构和各层网络模型。然后描述基于梯度下降的模型训练算法。最后,本文给出了一种使用无监督数据的预训练方法。

4.1 定义与符号

以下给出几个下文使用的几个定义与符号^[49]。

定义一：单维时间序列：单维时间序列是指由同一数据源的单个数值按其发生的时间先后顺序排列而成的数列,这些数值通常是时域上等间隔连续采样的结果。一个单维时间序列可以表示为 $T = \{t_1, t_2, \dots, t_n\}$ 。

定义二：多维时间序列：多维时间序列是由多个共享时间戳的单维时间序列的构成的集合。一个多维时间序列 M 可以表示为一个 $N \times T$ 的矩阵,其中 n 表示其中单维时间序列的个数, t 表示时间长度。令 m_i 和 m_j 表示 M 的第 i 行和第 j 列,那么 m_i 表示第 i 个单维时间序列,而 m_j 表示在 j 时刻各个单维时间序列的值构成的向量。

定义三：子序列：子序列是从一个时间序列 T 中连续提取出来的一段数值点(或向量)构成的序列。可以表示为 $S = \{t_i, t_{i+1}, \dots, t_{i+k-1}\}$ 。

定义四：强标注时间序列：如果一个时间序列可以划分为一些系列紧密衔接的子序列,而且其中每个子序列都具有来自领域专家的良好标注,来表示这一段子序列的性质或类别,那么这个时间序列就是强标注的。举例来说,心脏病专家会对心电图按照心率划分子序列,然后针对每个子序列按照其波形给予“正常”或“心律不齐”的标注。

定义五：弱标注时间序列：如果一个长且未分段的时间序列只有一个关联的全局标签,那么这就是一个弱标注的时间序列。比方说记录 PM2.5 的数值在一天中的变化情况的时间序列可以关联一个当天的天气状况,那么这就是一个弱标注时间序列。

相对于强标注时间序列,弱标注时间序列更为容易取得。在电力系统中,可以利用系统仿真的方法获取大量的弱标注时间序列。因此,本文的主要研究对象是弱标注时间序列。

4.2 问题描述

在电网系统中,在各个母线上部署的 PMU 单元可以实时地进行运行数据的采集工作。我们这里只考虑采集的母线电压和相角信息,这可以构成二个多维时间序列 U 、 Θ , 分别表示电压和相角。如前文所述, U 和 Θ 是一个 $N \times T$ 的矩阵,其中每一行分别表示某个母线电压或相角信号构成的单维时间序列。在下方

数据预处理部分会提到，我们有必要将 U 和 Θ 转化为实电压和虚电压时间序列 U^R 和 U^I 的形式。

电力系统的暂态稳定分为稳定和失稳两种。根据先验知识，我们可以给时间序列 U 和 A 赋予一个标签 $y \in \{0,1\}$ ，来表示其对应的系统在 $[0,T]$ 时间段内的稳定性，0 表示系统稳定，1 表示系统失稳。那么 U^R 和 U^I 就可以构成弱标注时间序列。

我们的主要任务是截取 $U^R, U^I \in R^{N \times T}$ 中的一段 $U^{R'}, U^{I'} \in R^{N \times K}, K < T$ ，来预测 $[0,T]$ 时间段内系统的稳定性。这可以看作是针对于多维弱标注时间序列的分类问题。

在本方法中，除去预处理阶段，我们不需要关于电力系统的任何先验知识，也不关注电力网络的拓扑结构。唯一需要的信息是 U^R, U^I 的时间序列。然后构造训练集 $D_{train} = \{(U^R, U^I), y\}$ 进行模型训练，来预测测试集 $D_{test} = \{(U^R, U^I)\}$ 中的稳定情况。

4.3 模型架构

如前文所述，我们提出的模型可以分为三个部分：进行数据预处理的坐标变换层、自动进行特征提取的多层卷积网络和用于稳定预测的多层感知器

一般来说由卷积神经网络构成的分类器通常包括两个部分：特征提取层和全连接的多层感知器（MLP）。特征提取层的任务是从原始数据中进行自动地特征提取，多层感知器基于特征数据来进行分类操作。特征提取层由数个相同的部分堆叠而成，每个部分都包括连续的二层网络：卷积层和池化层。

在 CNN 的传统应用领域图像数据挖掘中，典型的输入对象是 2 维的 RGB 图像，而非时间序列。因此，我们对传统的 CNN 进行了一些改进，将其用于多维时间序列的特征提取当中。具体的做法是将经过坐标变换层后的时间序列按照 PMU 节点分组，构成 N 个多维时间序列，然后分别利用权值共享的卷积网络对这些时间序列进行独立的特征分析。最后再组合特征通过一个传统的 MLP 来进行稳定预测。

图 3-2 描述了在以 IEEE39 系统作为稳定评估对象时的模型结构。在可能的预训练之后，数据首先通过坐标变换层，然后被分为 39 组，分别进入 39 个对应的权值共享的 2 阶卷积神经网络中，在其中连续通过组合的卷积层和池化层来学习不同层次的特征，接着将各个卷积网络的输出展开为 1 维向量，合并后通过多层的密集网络来完成分类。

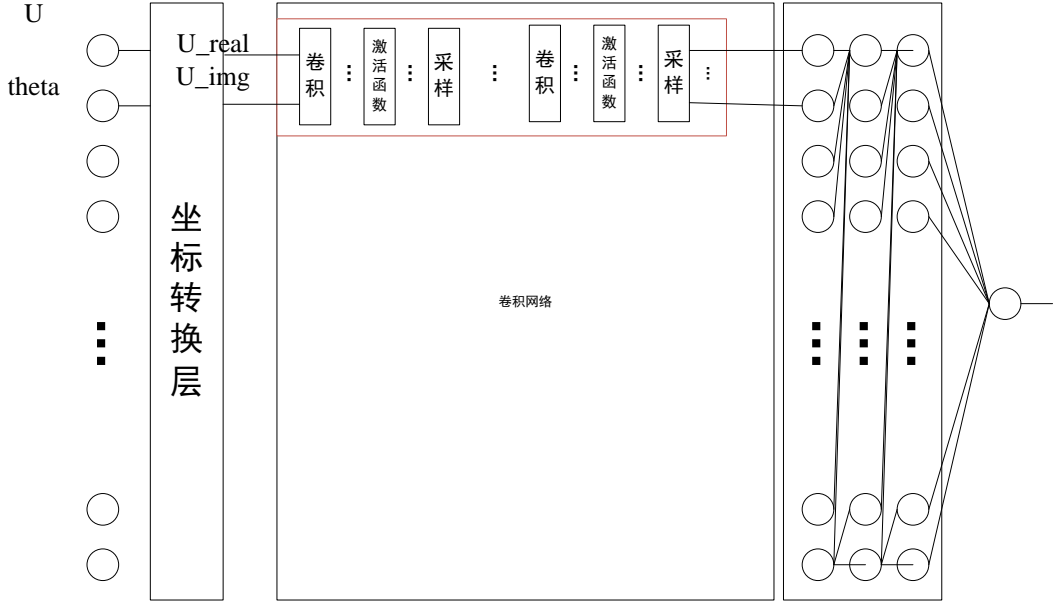


图 3-2 以 IEEE39 系统作为稳定评估对象时的模型结构

4.4 坐标转换层

模型的输入样本是 PMU 采集到的母线的交流电峰值电压 u 和相角 θ 数据，其中，某个节点的相角 θ 表示该节点电压的相位相对于某个标准节点的相位的差。 u 和 θ 具有不同的量纲和取值范围，这会对模型的决策边界造成影响，进而导致模型精度的降低。此外，在 u 和 θ 构成的极坐标空间中，一个绕原点旋转的电压向量会在经过(1,0)向量时使 θ 的取值产生从 0° 到 360° 或从 360° 到 0° 的跳变，这会使卷积网络学习到错误的特征，增加误判率。

本文提出了对 PMU 电压相角数据进行处理的概念，来解决上述问题。交流电压 $u(t) = u_0 \cos(\omega t + \theta)$ 可以用复数表示，复数的模等于交流电压的峰值 u_0 ，其幅角等于交流电压的相位 $\omega t + \theta$ 。于是，这个复数可以写为：

$$u^c = u_0 \exp[j(\omega t + \theta)] = u_0 e^{j\omega t} e^{j\theta} = u_0 e^{j\omega t} (\cos \theta + j \sin \theta) \quad (3-1)$$

可以看到，在某个时刻 t ，母线的复电压可以由 $u \cos \theta$ 和 $u \sin \theta$ 两个变量完全确定。令

$$u^r = u \cos \theta \quad (3-2)$$

$$u^i = u \sin \theta \quad (3-3)$$

那么，有：

$$u^c = e^{j\omega t} (u^r + j u^i) \quad (3-4)$$

在坐标转换层中，我们将 u 和 θ 转换为实电压 u^r 和复电压 u^i 的形式，这可以认为是极坐标和直角坐标的转换。

本层可以看作是针对 PMU 数据的一种特殊的预处理方法。常规的方法是对

数据进行标准化（Standardization）或正则化（Normalization）。标准化首先假设样本中的每个特征服从正态分布，然后将特征转化为标准正态分布的形式。正则化则是使每个样本的 L2 范数为 1。可以看出，这两种方法都不适用于以时间序列作为输入参数的 PMU 数据的预处理工作。而本层的方法产生 u^r 和 u^i 构成的时间序列 U^R 和 U^I 具有明确的物理特征和一致的量纲，且保留了 U 和 Θ 的全部信息和各个节点时间序列之间的相对关系。因此是一种更好的预处理解决方案。

4.5 卷积网络

卷积网络由若干连续的滤波层（Filter Layer）、激活层（Activation Layer）和池化层（Pooling Layer）堆叠而成。每一层的输入和输出称作特征映射（Feature Map）。

4.5.1 滤波层

滤波层的输入的特征映射可以看作是一个 N 维时间序列，表示为 $x^l \in R^{N \times T}$ ， l 表示所在层数。每个滤波层中包含 K 个大小相同的卷积核 W_k^l 。与对图像的卷积不同，时间序列的 N 个维度之间并不一定存在连续的空间关系，因此，我们设置卷积核的大小为 $N \times m$ ， m 为卷积核在时域上的大小。其意义在于同时对时间序列的 N 个维度进行卷积操作。

具体来说，每个卷积核的操作可以表示为如下公式的形式。

$$h_{k_t}^l = (W_k^l * x_k^l)_t + b_k^l \quad (3-5)$$

* 表示卷积操作，定义如下：

$$(W * x)_t = \sum_{i=0}^N \sum_{j=0}^m W_{ij} x_{(N-i)(t-j)} \quad (3-6)$$

卷积核的作用是提取时间序列中的显著模式，这种模式可能是时间序列的峰值，时域变化、零交点或周期波动。为了提取出时序数据中的细节信息，需要使用较小的卷积核。在这里，我们令 $m=5$ 。

考虑到在电网系统中，网络各节点之间存在拓扑连接关系，而非在单维欧式空间内的邻域关系。我们的卷积网络只针对单个节点的时间序列数据进行特征提取。对于第一个滤波层而言，输入数据是 2 维的复电压时间序列， $N=2$ 。之后的每个滤波层中， N 为上一个滤波层之后的卷积核的数量。

4.5.2 激活层

激活层将非线性算子引入神经网络中，使网络具有拟合任意形状的能力。两种常见的激活函数是 $\text{sigmoid}(x) = \frac{1}{1+e^{-x}}$ 和 $\text{tanh}(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ ，它们在神经网络领

域有着广泛的应用。近年来，一种新的激活函数 ReLu 被提了出来，并在一些领域得到了初步应用。ReLu 可以定义为 $f(x) = \max(0, x)$ 的形式。其优点在于计算简单，可以加快训练速度，并提高模型的泛化程度。此外，此激活函数也可以避免梯度消失的问题。

4.5.3 最大池化层

池化层的目的是对特征映射进行子采样。它可以降低输入时间序列的分辨率，并使得在卷积网络在处理与先前学习到的特征具有微小差别的样本时更为鲁棒。最大池化层首先将输入的特征映射分割成重叠或不重叠的若干等长子序列，然后针对每一段子序列，计算其特征映射的最大值。设我们每隔 P 个时间点进行一次采样，那么最大池化可以正式表示成如下公式的形式：

$$h_t^k = \max\{h_{(t-1)P+1}^k, h_{(t-1)P+2}^k, \dots, h_{tP}^k\} \quad (3-6)$$

最大池化层有以下两方面的优势。首先，通过获取最大值的形式，它可以只提取特征映射的最显著模式，从而剔除微小局部变化和相位平移的影响，可以生成更为鲁棒的特征描述。其次，它也可以降低特征数，降低后续处理的难度。其结构如图 3-3 所示：

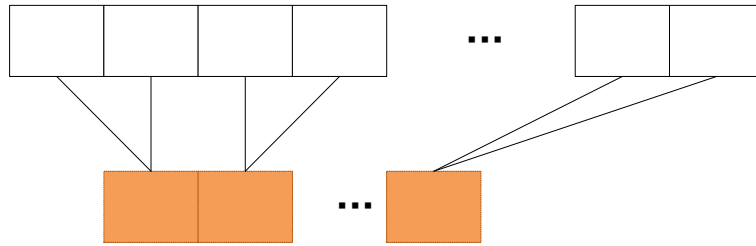


图 3-3 最大池化层

4.6 多层感知器

模型的最后部分是级联的多层感知器。它联合各个节点通过卷积网络后的特征映射输出作为网络的输入。其输出是模型的预测结果。感知器的最后一层是网络输出，只有一个节点，激活函数是 sigmoid，其它层的激活函数是 ReLu。

4.7 前向传播

输入的电压和相角数据首先通过坐标转换层，然后分组进入卷积网络进行特征提取，最后将得到的特征映射组合输入多层感知器获取预测结果。具体算法描述如下：

输入：电网中各个节点的电压和相角构成的时间序列 U 和 Θ ，大小为 $N \times T$
坐标转换层：对于时间序列的每个值

$$u_{ij}^r = u_{ij} \cos \theta_{ij} \quad (3-7)$$

$$u_{ij}^i = u_{ij} \sin \theta_{ij} \quad (3-8)$$

卷积网络：对于 $i = 1:N$

$$x^1 = \{u_{ij}^r, u_{ij}^i\} \quad (3-9)$$

$$x_k^2 = \max_pooling[ReLU(W_k^2 * x^1 + b_k^2)] \quad (3-10)$$

...

$$x_k^l = \max_pooling[ReLU(W_k^l * x^{l-1} + b_k^l)] \quad (3-11)$$

暂存特征

$$f_i = x_k^l \quad (3-12)$$

多层感知器：

$$x^{1'} = \{f_1, f_2, \dots, f_N\} \quad (3-13)$$

$$x^{l'} = ReLU(W^{l'} x^{l'-1} + b^{l'}) \quad (3-14)$$

$$y = sigmoid(W^{l'+1} x^{l'} + b^{l'+1}) \quad (3-15)$$

4.8 模型训练

4.8.1 通道间权值共享机制

深度神经网络在训练过程中，存在容易训练难度大，训练参数多，容易过拟合等问题。而深度卷积网络作为一种特征提取器，也有其自身的局限性。它特征提取的效果会受到训练数据的影响，对训练数据中未出现的特征提取效果较差。

对于我们的多通道模型而言，各个通道的输入具有一致或类似的物理特性。因此，我们可以认为这些输入具有相似的行为模式（**pattern**）。如果每个通道的卷积网络都只能从对应通道上有限的训练数据中进行学习，整个系统就会产生巨大的数据“浪费”。同时，也会导致模型参数数量的激增，进一步加大了训练难度。

从数学上来说，假设对于一个物理过程，按照某种方式采样构成的时间序列中，所有可能表现出的行为模式属于集合 P 。对于我们的多通道模型而言，每个通道上的输入可以认为是对同一物理过程中同种属性在不同部位方式采样得到

的时间序列。对于通道 i 而言, 其时间序列可能出现的行为模式属于集合 P_i , $P_i \subset P$ 。对于由有限个采样构成的训练集合而言, 通道 i 对应的时间序列可能出现的行为模式属于集合 P'_i , $P'_i \subset P_i$ 。如果每个通道的卷积网络只能从对应输入中进行学习, 那么其可能学习到的特征集合的上界为 P'_i 。这样的卷积网络在面对测试集中可能存在的包含模式 $p \in P_i - P'_i$ 的时间序列时表现欠佳。

为了解决这些问题, 本文提出了通道间的权值共享机制。不同通道间共享参数权值, 这可以通过特殊的反向传播过程实现。

在反向传播时, 误差通过神经网络从输出端向输入端进行传递, 在此过程中更新经过的连接的权值。因此, 权值的共享可以通过特殊的反向传播方式来实现: 在反向传播过程中, 卷积网络更新的权值为各个通道内的卷积网络单独更新的权值的和。通俗来说, 相当于在一次训练过程中, 卷积网络的权值更新了 n 次, n 为通道数。

通过这种方式, 每个通道的卷积网络可能学习到的模式集合为的上界 $P'_0 \cup P'_1 \cup \dots \cup P'_n \geq P'_i$, 可以降低训练集中未识别模式出现的概率。

4.8.2 反向传播过程

神经网络通过反向传播的方式进行训练。在通过前向传播获取到输出 y 后, 可以根据损失函数计算出预测的误差。然后, 可以通过沿连接逐层反向传播误差的方式来得出每一层的误差。将误差作为网络的梯度, 使用梯度下降的方式来更新每一层的权值。

每个训练样本的损失函数为:

$$Cost(h_w(x), y) = -y \log(h_w(x)) - (1 - y) \log(1 - h_w(x)) \quad (3-16)$$

m 个训练样本的损失函数为:

$$J(W) = \frac{1}{m} \sum_{i=1}^m Cost(h_w(x), y) \quad (3-17)$$

损失函数计算的误差首先反向传播进入 MLP 中, 由链式法则可知, 网络中某个连接的权值 w_{ij}^l 对整体误差的影响可以用如下公式表示:

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial w_{ij}^l} \quad (3-18)$$

其中, o 表示网络的输出, z 表示神经的输入的加权和。如果用 a_i^l 表示每个神经元的激活, 用 $f(x)$ 表示激活函数, 那么有:

$$z_j^l = \sum_i w_{ij}^{l-1} a_i^{l-1} \quad (3-19)$$

所以

$$\frac{\partial E}{\partial w_{ij}^l} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial z_j^{l+1}} \cdot a_i^l = \delta_j^{l+1} \cdot a_i^l \quad (3-20)$$

$$\delta_i^l = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial z_i^l} = \sum_j \frac{\partial E}{\partial z_j^{l+1}} \cdot \frac{\partial z_j^{l+1}}{\partial a_i^l} \cdot \frac{\partial a_i^l}{\partial z_i^l} = (\sum_j \delta_j^{l+1} w_{ij}^l) f'(z_i^l) \quad (3-21)$$

其中， δ_i^l 表示神经元的残差，由之前的残差沿路径逐步反向传播求导得到。

卷积网络的每一部分由池化层和卷积层组成。

对于池化层而言，反向传播是一个上采样过程。

$$\frac{\partial E}{\partial x_i^{l-1}} = up \left(\frac{\partial E}{\partial x_i^l} \right) \quad (3-22)$$

对于滤波层而言

$$\delta_i^l = \frac{\partial E}{\partial z_i^l} = \frac{\partial E}{\partial x_i^l} \cdot \frac{\partial x_i^l}{\partial z_i^l} = f'(z_i^l) \circ up \left(\frac{\partial E}{\partial x_i^{l+1}} \right) \quad (3-23)$$

其中， \circ 表示矩阵各元素分别相乘。

卷积层的偏置的微分可以由以下公式给出：

$$\frac{\partial E}{\partial b_i} = \sum_u (\delta_i^l)_u \quad (3-24)$$

每个卷积核的微分是所有与这个卷积核相关的运算产生的微分的和。我们使用卷积来表示这个操作：

$$\frac{\partial E}{\partial k_{ij}^l} = \frac{\partial E}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial k_{ij}^l} = \delta_j^l * reverse(x_i^{l-1}) \quad (3-25)$$

$$\frac{\partial E}{\partial x_i^{l-1}} = \sum_j \frac{\partial E}{\partial z_j^l} \cdot \frac{\partial z_j^l}{\partial x_i^{l-1}} = \sum_j pad(\delta_j^l) * reverse(k_{ij}^l) \quad (3-26)$$

通道间的权值共享机制也是通过特殊的反向传播机制实现。由于 MLP 输入向量是各个通道的卷积网络输出向量的组合。所以对于通道 c 的最后一层，有：

$$\frac{\partial E}{\partial (x_i^l)_c} = \frac{\partial E}{\partial x_{c \cdot n + i}^{l+1}} \quad (3-27)$$

其中， n 表示每个通道特征映射展开为一维向量后的大小。

卷积网络的参数的微分是各个通道中参数的微分的和。

$$\frac{\partial E}{\partial w} = \sum_c \frac{\partial E}{\partial w_c} \quad (3-28)$$

4.8.3 更新权值

在获取每个参数的梯度之后，我们通过 Adam 算法来更新权值。相对传统的梯度下降算法，它更快地收敛，并降低收敛到局部最优的风险。其过程可以如下公式表示：

$$m = \beta_1 m + (1 - \beta_1) \Delta x \quad (3-29)$$

$$v = \beta_2 v + (1 - \beta_2) \Delta x^2 \quad (3-30)$$

$$x = x - \frac{\alpha m}{\sqrt{v} + \epsilon} \quad (3-31)$$

其中, α 表示学习率, $\beta_1, \beta_2, \epsilon$ 为超参数。 $\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1e - 6$ 。

4.9 决策边界与评价指标

在应用于电网系统时, 模型的最终输出可以认为是将来电网失稳的概率。因此, 还需要一个超参数 p , 来作为分割稳定与失稳的决策边界。 p 的取值会对最终结果产生影响。增大 p 值会减小将稳定情况判断为不稳定 (误动) 的概率, 但会增大将不稳定的判断为稳定的概率 (拒动)。

本文主要使用精确度、召回率、操作特征曲线 (ROC) 与其下方面积来评估分类器的性能。

对于每个场景, 根据其是否失稳可以分为两类, 正例 (Positive) 与反例 (Negative)。场景数据经过模型预测的结果是否失稳也可以分为两类。

表 3-1 真值表

	场景失稳 (Positive)	场景失稳 (Negative)
预测失稳 (Positive)	True Positives (TP)	False Positives (FP)
预测稳定 (Negative)	False Negatives (FN)	True Negatives (TN)

那么, 精确度 (precision) 可以定义为如下公式的形式, 表示预测失稳的场景中, 真正失稳的比例。

$$P = \frac{TP}{TP + FP} \quad (3-32)$$

召回率 (recall) 可以定义为如下公式, 表示在所有失稳场景中, 成功预测的比例。

$$R = \frac{TP}{TP + FN} \quad (3-33)$$

精确度与召回率与分类器判决边界超参数 p 的选取有很大的关系。对于同一预测模型而言, 精确度越高, 召回率就越低。连续选取不同的 p 值, 可以得出精确度召回率曲线 (PR 曲线)。

此外, 也可以使用操作特征曲线 (ROC) 与其下方面积 (AUC) 来评估分类器的性能。ROC 关注命中率 (True Positive Rate TPR) 和误报率 (False Positive Rate FPR) 两个指标。

$$TPR = \frac{TP}{TP + FN} \quad (3-34)$$

$$FPR = \frac{FP}{FP+TN} \quad (3-35)$$

命中率代表能将正例（失稳）成功预测的概率，误报率代表将负例（稳定）错误预测为正例（失稳）的概率。ROC 曲线是以 FPR 为横轴，TPR 为纵轴的曲线，其下方面积越大，说明分类器效果越好^[50,51,52]。

本文通过单个 Epoch 训练耗时、收敛所需的 Epoch 数目和预测耗时来评价模型的时间复杂度。

4.10 神经网络预训练

本文中神经网络的预训练主要是指用于特征提取的卷积神经网络的预训练，其训练方法如如小节 3.1.4 中所示。值得注意的是，由于采用了权值共享的机制，训练卷积网络样本各个场景中的每个 PMU 节点产生的电压相角时间序列，而非单个场景的所有 PMU 节点的数据。

4.11 工程实现

本文的工程实现主要是使用 Python 语言，在开源人工神经网络工具包 Keras 上进行二次开发。Keras 使用 Tensorflow 作为后端，使用 Nvidia 图形显示卡加速网络的运行。

模型在普通商用台式机平台上进行训练和运行。其 CPU 为 i5-6300，内存为 8GB DDR4，模型训练和测试的主要计算均在商用图形显示卡 NVidia GeForce GTX950 上进行，基于开源工具包 Keras 和 Tensorflow 实现。GTX950 具有 768 个 CUDA 核心，核心频率为 1024MHZ，搭载有位宽为 128bit 的 2GB GDDR5 显存。

在软件方面，模型运行在 Ubuntu 16.04 系统上，运行的 Python 版本是 3.5。系统上安装有 Cuda 驱动。在 Tensorflow 中设置运算在 GPU 上运行，采用单精度浮点运算。在 Keras 中设置内核为 Tensorflow。

模型中的坐标转换层实现较为简单，核心源代码如下^[53]：

```
for i in range(T):
    for j in range(N):
        V_real[i,j] = V[i,j] * math.cos(Theta[i,j] /
            180 * math.pi)
        V_img[i,j+39] = V[i,j] * math.sin(Theta[i,39 +
            j] / 180 * math.pi)
```

模型的其他层网络采用 Keras 实现，包括 N 个通道，每个通道有两层卷积和池化层，以及 3 层全连接的 MLP 层。值得注意的是，本文通过一次定义，多次运行的方式，实现了各个通道之间的权值共享过程。其具体实现过程在前文中已详细描述，在此不再赘述。

本文最后提出了电网时间序列卷积核的可视化方法，将卷积核表示为多维时间序列的形式，其核心代码如下所示：

```
c1 = model.get_layer('convolution2d_1')
w,b = c1.get_weights()

n_kernels = w.shape[-1]

f,axs = plt.subplots(n_kernels//4,4,sharex=True,
                    sharey=True, figsize=(8,2 * n_kernels // 4))
k = 0
for i in range(n_kernels//4):
    for j in range(4):
        ax = axs[i,j]
        ax.plot(w[:,0,0,k], 'o-')
        ax.plot(w[:,1,0,k], '^--')
        k += 1
f.subplots_adjust(wspace=0,hspace=0)
```

4.12 小结

本章提出了针对课题研究内容的基于深度神经网络的具体解决方案，包括模型架构，每层的定义与参数，前向与反向传播算法，模型评价指标，预训练方法等。最后，给出了关于模型具体实现的一些细节。

第五章 实验设计与结果验证

5.1 实验数据

首先，我们基于新英格兰 10 机 39 节点系统（IEEE39）仿真生成了一批训练样本。仿真计算使用 PSAT 在 OCTAVE 环境中进行。IEEE39 系统代表美国新英格兰镇的一个真实电力网络。该标准系统拥有 10 台发电机，39 条母线和 46 条交流线，基准功率为 100MVA，基准电压为 345kV。

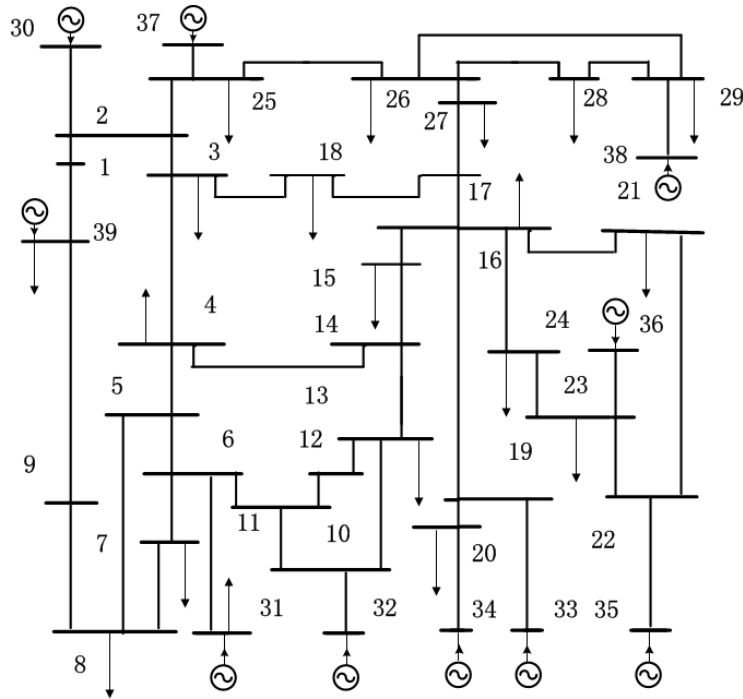


图 4-1 IEEE39 节点系统 引自维基百科

通过仿真，我们模拟了在 0.1s 时，在某条线路的某侧发生三相短路故障，在一段时间（取 0.2~0.8s 内某值）后切断故障的场景。仿真时长为 4s，步长为 0.005s。

通过仿真，我们得到了每个场景中电网各个节点的电压 V 和相角数据 A ，以及发电机的功角数据 G ，来模拟在所有母线上加装 PMU 采集数据的情况。其中 V 和 A 作为多维时间序列用于训练和测试模型性能。 G 用于判断场景的暂态稳定性，来给这些场景添加标注。

电力系统的暂态稳定的判决依据是发电机内的相对功角差的最大值 Δg_{\max} 。如果在某一时刻 $\Delta g_{\max} > 360^\circ$ ，那么就认为系统已经失稳。电力系统中的暂态稳定具有一定的不可逆性，如果系统已经失稳，那么是无法自行回复到稳定状态的。因此，对于 $V, A, G \in R^{N \times T}$ 而言，可以利用 g_T 的值，给时间序列 V 和 A 赋予一个

标签 $y \in \{0,1\}$, 来表示其对应的系统在 $[0, T]$ 时间段内的稳定性, 0 表示系统稳定, 1 表示系统失稳。那么 V, A 就可以构成弱标注时间序列。

在去除电压失稳样本后, 共得到了 2962 组样本, 每组为 72×800 的矩阵, 表示电压和相角的时间序列, 共 2.26GB 数据。其中, 稳定样本 2121 组, 失稳样本 841 组。我们按照 60%, 40% 的比例将其划分为训练集和测试集, 大小分别为 1777 和 1185。

5.2 模型设置

IEEE39 节点系统中共有 39 个发电或负荷节点。因此, 在我们的模型中共需要 39 个通道来接收输入数据。如前文所述, 这些数据在通过坐标转换层后进入卷积神经网络, 之后再组合特征映射进入 MLP 完成预测。

其中, 每个通道是一个两层卷积网络, 第一层包含 8 个卷积核, 第二层包含 4 个卷积层, 卷积步长为 10, 激活函数为 ReLu。每层卷积层后接一个池化层, 步长为 2。各个通道间共享网络权值。

最后的 MLP 由 4 层全连接层构成, 前三层每层包含 64 个节点, 激活函数为 ReLu, 最后一层只有一个节点, 激活函数为 Sigmoid。模型的具体结构如图 4-2 所示:

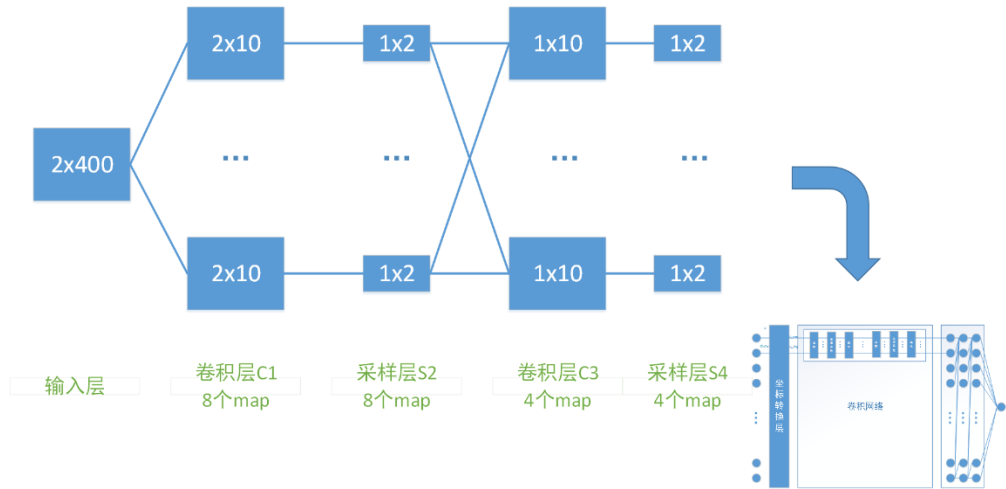


图 4-2 模型设置

模型采用 Early stopping 技术训练若干个 epoch, 每个 batch 的大小为 50。因为训练样本中正负样本的数量并不相同, 训练时会根据样本中正负样本的比例为每个样本进行加权。对于某一类样本 y , 加权公式为:

$$w_y = \frac{n_{samples}}{n_{classes} \times n_y} \quad (4-1)$$

5.3 结果分析

作为一个预测系统, 其核心在于足够的精确度和运算速度。本文的结果分析

主要从这两个方面展开。模型截取 $U^R, U^I \in R^{N \times T}$ 中的一段 $U^{R'}, U^{I'} \in R^{N \times K}, K < T$, 来预测 $[0, T]$ 时间段内系统的稳定性。所以 K 的选取非常重要, 不同的 K 值对模型的精度和时间复杂度都有影响。具体来说, K 值越大, 模型所获取的信息量就越大, 但是运算的时间复杂度也会提升, 此外也会缩短系统在预测失稳后的反应时间。另一方面, 一个小的 K 值意味着模型的精度有所下降, 但是其更大的时间裕度和较低的时间复杂度可以为系统赢得更多宝贵的反应时间。

图 4-3 表示了在选择不同 K 值 (单位为秒) 时, 模型的 ROC 的 AUC 值。AUC 表示模型 ROC 曲线的下方面积, 取值范围是 0 到 1, AUC 越大, 说明模型效果越好。从图中可以看出, 我们的模型具有非常好的预测效果, AUC 普遍在 0.99 以上。而且随着 K 值的增大, 模型的预测性能也有进一步的提升。

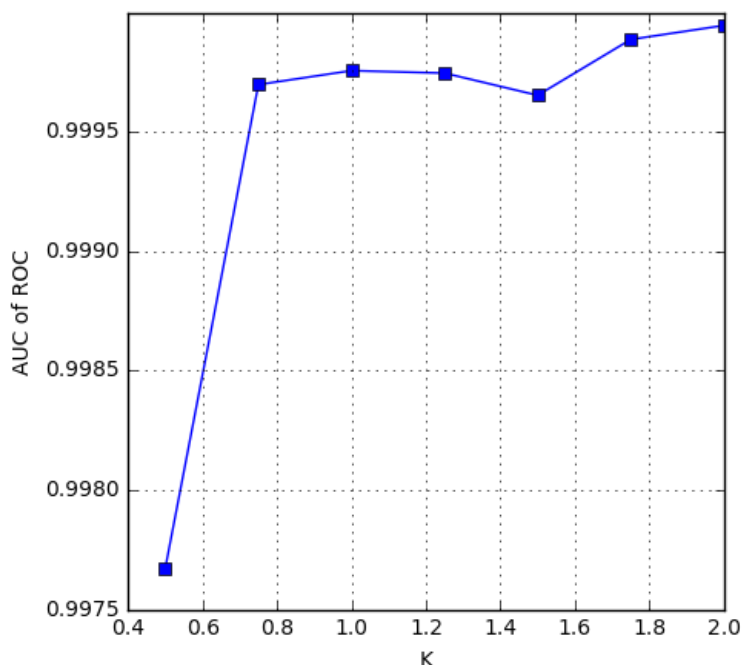


图 4-3 不同 k 值下 ROC 曲线的 AUC 值

图 4-4 为选取不同 K 值 (单位秒) 时模型的精确度和召回率曲线。从图中也可以看出模型的优良效果, 而且其性能随着 K 值的增大有着进一步的提升。对于最好的模型而言, 当其召回率为 0.95 左右时, 精确度仍可以达到 1。精确度为 0.99 左右时, 召回率为 1。对于 1185 个训练样本而言, 这意味着在仅有 59 个未预测样本 (拒动) 的情况下, 模型预测的成功率为 100%。在仅有 12 个预测失误 (误动) 的情况下, 模型预测出了所有的失稳事件。而此时距离故障开始只过了 1.9s。即使只使用 0.5s 的样本数据进行预测, 模型依然有 0.9 以上的成功率。

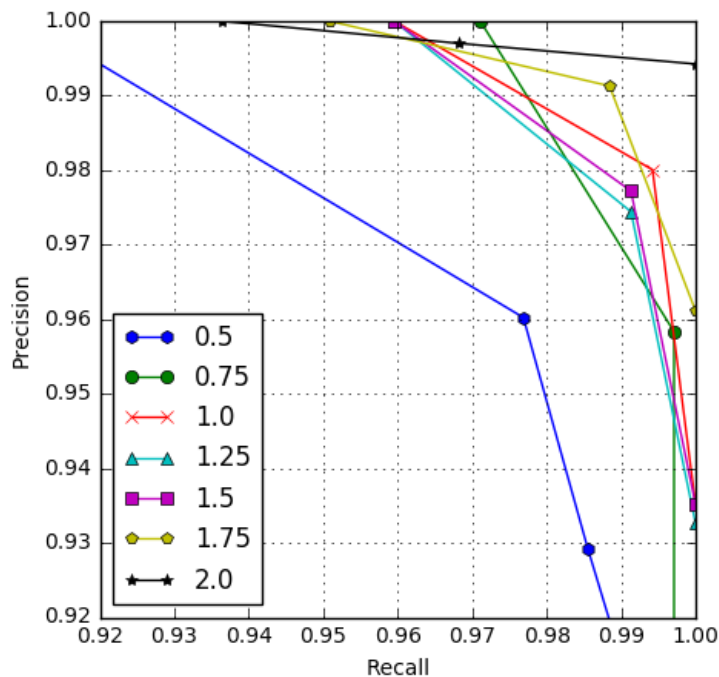


图 4-4 精确度召回率曲线

图 4-5 为模型的 ROC 曲线。和图 4-4 类似，它也体现了模型在选取不同决策边界时预测的性能。其下方面积可以体现模型的综合性能。从图中可以得出与上述分析一致的结论。

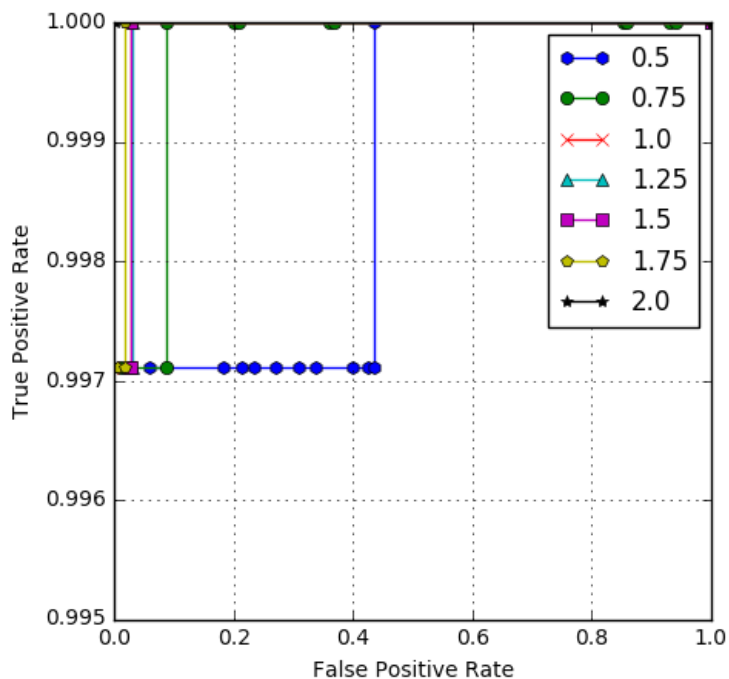


图 4-5 ROC 曲线

模型在预测速度上也有不错的效果。图 4-6 为不同 K 的取值下模型连续预

测 1183 个样本所需时间。可以看出,虽然 K 的取值对模型的预测时间有所影响,但在最耗资源的情况下,模型对单个样本的预测时间也只需 3.5ms。在故障发生后,模型的响应时间近似等于 K 的取值。

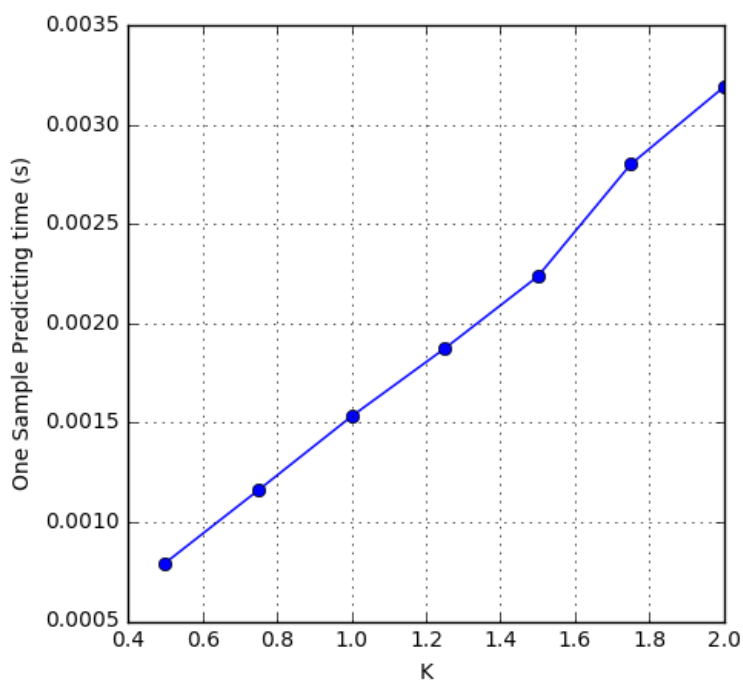


图 4-6 预测时间

受人工神经网络复杂性的影响,模型的训练需要较长时间,且与 K 的取值呈现线性关系。图 4-7 为模型在选取不同 K 值时,单个 Epoch 的训练时间(单位为秒)。一个 Epoch 指在随机梯度下降中,对样本的一次遍历过程。

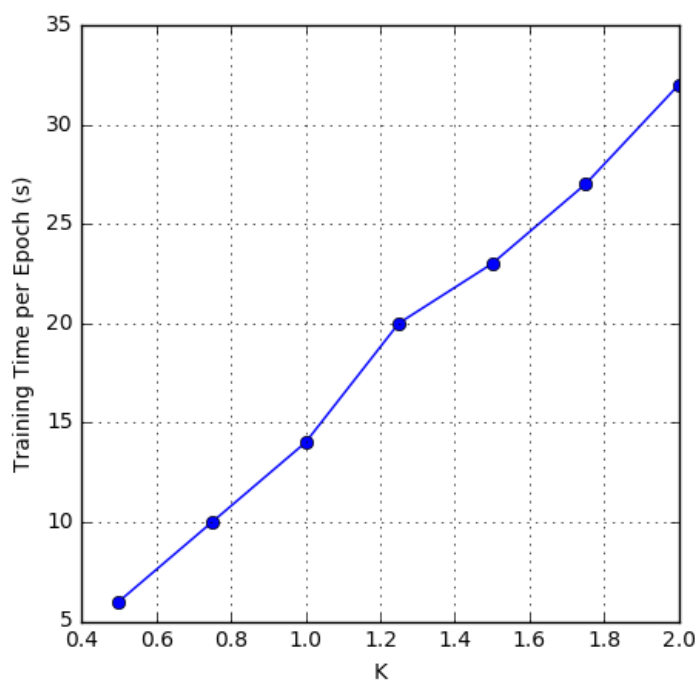


图 4-7 模型训练时间

图 4-8 与图 4-9 为模型在 $K=1.0s$ 时，在训练过程中，损失函数和精准度的变化情况。可以看出，模型拟合状况良好，且可以较快地完成收敛。

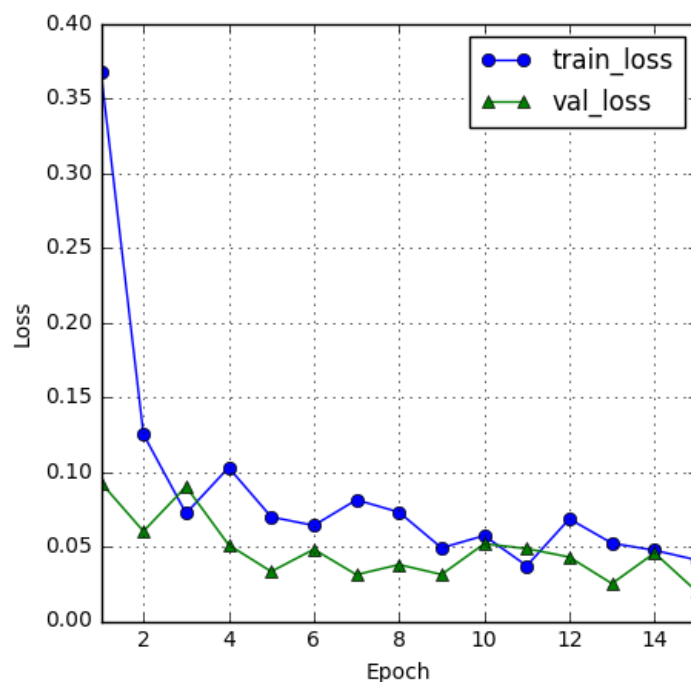


图 4-8 损失函数收敛情况

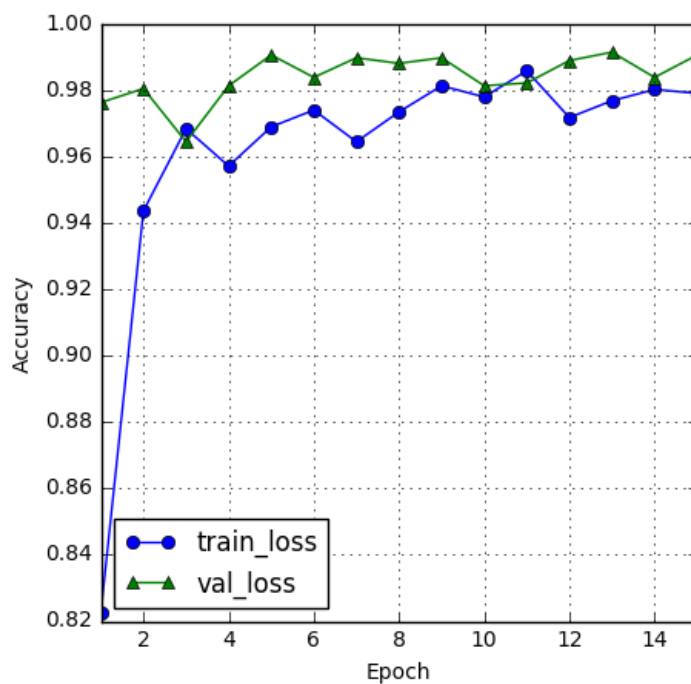


图 4-9 精确率收敛情况

5.4 卷积网络可视化

在我们的卷积网络中，训练后的卷积网络的卷积核的权值包含输入时间序列

的模式信息。这些信息本身也可以视为一个时间序列。在我们的网络中，第一层卷积网络包含 8 个卷积核。第二层网络包含 $8 * 4 = 32$ 个卷积核。在第一层中，每个卷积核的内容是一个二维定长时间序列，第二层的卷积核是一维时间序列。

图 4-10 和图 4-11 分别是第一层和第二层网络的卷积核权值的直观可视化展示。从图中可以看出，卷积网络的确自动完成了时间序列模式的提取工作。其中，第一层的卷积主要提取的是节点电压实部和虚部的局部变化模式，包括波峰波谷，上升下降，同向异向等具体变化模式。而第二层卷积主要是更长期的变化规律，如平稳变化，周期波动，断崖式变动等。这些卷积核包含的时间序列信息，也可以为领域专家提供一些启发。

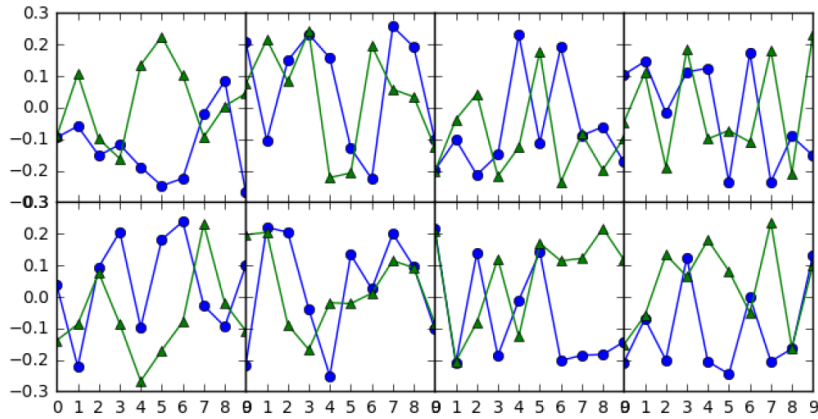


图 4-10 第一层卷积网络的可视化

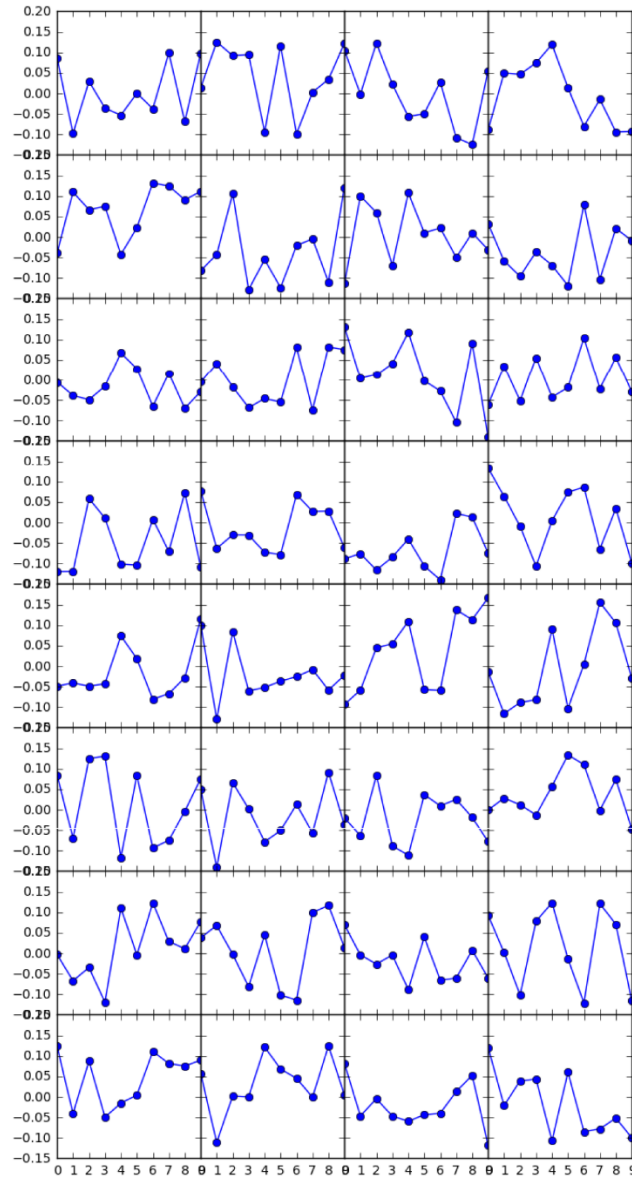


图 4-12 第二层卷积网络的可视化

5.5 小结

本章详细分析了模型的具体性能情况。首先给出了实验的设计方案，然后根据方案调整模型设置，最后对结果进行了分析。可以看出，我们提出的模型在预测精度和预测速度都有不错的表现。对卷积网络的可视化分析表明，我们的模型的在特征提取上确实有不错的效果。

第六章 总结与展望

本文提出了一种基于深度神经网络的电网暂态稳定预测模型，本模型最大的优势在于免去了繁琐复杂的人工特征提取工作，直接使用电网的电压和相角原始数据作为输入，从数据中判断电网系统的稳定边界，在不依赖领域专家的先验知识的情况下达到了优秀的判别精度和反应速度。本文是深度学习在电力系统核心问题上的探索应用，具有一定的学术价值。

在论文中，首先详细介绍了当年电力系统暂态稳定分析的研究现状和机器学习理论的发展历程，再进一步研究了电力系统暂态稳定的物理机理和过程，以及人工神经网络技术与深度学习的相关理论。在这基础上，第三章提出了本文的核心模型，并给出了明确的数学定义和推导。最后，我们通过设计并实施实验，验证了模型的良好性能。

由于时间和精力有限，本文提出的方法还有很大的改进空间。首先，本文的方法与电网系统的实时在线工程化应用依然有一定的距离，在实际应用中还有诸多问题有待解决，如在线连续检测方法，预测告警机制，模型随时间的概念飘移等。此外，由于经济条件有限，受实验样本规模和机器性能的影响，文中的实验模型在网络深度和规模上还远不是最优。在之后的工作中，倘若可以基于文中模型，结合残差网络等先进结构构建大型或超大型神经网络，预期可以在解决大规模电网的稳定问题上发挥一定的作用。最后，使用机器学习解决电网稳定问题，不仅需要高效快速的检测方法，还需要在控制理论上有所创新，如果可以将本模型与增强学习（Reinforcement Learning）理论^[54]相结合，或许可以研发出有效的电网暂态稳定自适应监测与控制系统。

参考文献

- [1] Edison T A. Transmission of power: U.S. Patent 493,858[P]. 1893-3-21.
- [2] 王春成. 交直流并联输电系统大扰动短期电压稳定实用动态安全域[D]. 天津大学, 2007.
- [3] 曾鸣, 李娜, 董军,等. 基于大安全观的电网运行管理关键技术--关于印度大停电的思考[J]. 电力系统自动化, 2012, 36(16):9-13.
- [4] Transmission O O E. Grid 2030: A national vision for electricity's second 100 years[J]. 2010.
- [5] DKE/VDE, "German standardization roadmap E-energy/ Smart Grid", Available: <http://www.vde.de>, 2010.DKE, The German Standardization Roadmap E-Energy/Smart Grid.
- [6] 王继业, 刘道伟, 马世英,等. 信息驱动的全球能源互联网全景安全防御系统[J]. 电力信息与通信技术, 2016(3):13-19.
- [7] 刘道伟, 张东霞, 孙华东,等. 时空大数据环境下的大电网稳定态势量化评估与自适应防控体系构建[J]. 中国电机工程学报, 2015, 35(2):268-276.
- [8] 周兴德. 基于PSASP7.1的单机-无穷大电力系统暂态稳定仿真[J]. 通讯世界, 2014(7):51-53.
- [9] 叶圣永. 基于机器学习的电力系统暂态稳定评估研究[D]. 西南交通大学, 2010.
- [10] 顾卓远. 基于响应的电力系统暂态稳定控制技术研究[D]. 中国电力科学研究院, 2014.
- [11] 李彬. 考虑联络线暂态稳定性约束的最优潮流计算[D]. 天津大学, 2005.
- [12] 吴为, 汤涌, 孙华东,等. 基于广域量测信息的电力系统暂态稳定研究综述[J]. 电网技术, 2012, 36(9):81-87.
- [13] Liu C W, Thorp J S. New methods for computing power system dynamic response for real-time transient stability prediction [J]. IEEE Trans on Circuits and Systems, 2000, 47(6): 324-337.
- [14] 李琰, 周孝信, 周京阳. 基于新增虚拟节点的系统受扰轨迹预测[J]. 电力系统自动化, 2007, 31(12): 19-22.
- [15] 李琰, 周孝信, 周京阳. 基于引入虚拟负荷的发电机暂态稳定预测[J]. 电工技术学报, 2008, 23(3): 103-107.
- [16] 万秋兰, 单渊达. 对应用直接法分析电力系统暂态稳定性的再认识[J]. 电力

- 系统自动化, 1998(9):13-15.
- [17] Ota Y, Ukai H, Nakamura K, et al. Evaluation of stability and electric power quality in power system by using phasor measurements[C] Proceedings of International Conference on Power System Technology. Australia: IEEE, 2000: 1135-1140.
- [18] 滕林, 刘万顺, 賁志皓, 等. 电力系统暂态稳定实时紧急控制的研究[J]. 中国电机工程学报, 2003, 23(1): 64-69.
- [19] Maria G A, Tang C, Kim J. Hybrid transient stability analysis[J]. IEEE Transactions on Power Systems, 1990, 5(2):384-393.
- [20] 陈康, 向勇, 喻超. 大数据时代机器学习的新趋势[J]. 电信科学, 2012, 28(12):88-95.
- [21] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity The bulletin of mathematical biophysics, 1943, 5 (4):115 — 133.
- [22] 福岛邦彦. 视觉生理与仿生学[M]. 科学出版社, 1980.
- [23] LeCun Y, Boser B, Denker J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural computation, 1989, 1(4): 541-551.
- [24] Hochreiter S. Untersuchungen zu dynamischen neuronalen Netzen[J]. Diploma, Technische Universität München, 1991: 91.
- [25] Weng J, Ahuja N, Huang T S. Cresceptron: a self-organizing neural network which grows adaptively[C]//Neural Networks, 1992. IJCNN., International Joint Conference on. IEEE, 1992, 1: 576-581.
- [26] Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]//Proceedings of the 24th international conference on Machine learning. ACM, 2007: 791-798.
- [27] 刘玉田, 林飞. 基于相量测量技术和模糊径向基网络的暂态稳定性预测[J]. 中国电机工程学报, 2000, 20(2):19-23.
- [28] 吴琼, 杨以涵, 刘文颖. 基于最小二乘支持向量机的电力系统暂态稳定在线预测[J]. 中国电机工程学报, 2007, 27(25):38-43.
- [29] 王亚俊, 王波, 唐飞, 等. 基于响应轨迹和核心向量机的电力系统在线暂态稳定评估[J]. 中国电机工程学报, 2014(19):3178-3186.
- [30] 孙景强. 电力系统暂态稳定约束下的预防控制新算法研究[D]. 天津大学, 2005.
- [31] 李光琦编. 电力系统暂态分析[M]. 中国电力出版社, 2007.

- [32] 刘金波. 舰船电力系统暂态稳定性分析[D]. 哈尔滨工程大学, 2002.
- [33] 梁振光. 发电机功角的实时计算方法[J]. 电力系统保护与控制, 2004, 32(1):12-15.
- [34] 吴相钰, 陈阅增. 陈阅增普通生物学:普通生物学[M]. 高等教育出版社, 2005.
- [35] Møller M F. A scaled conjugate gradient algorithm for fast supervised learning[J]. Neural networks, 1993, 6(4): 525-533.
- [36] Bottou L. Stochastic gradient learning in neural networks[J]. Proceedings of Neuro-Nimes, 1991, 91(8).
- [37] Kingma D, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
- [38] Attwell D, Laughlin S B. An energy budget for signaling in the grey matter of the brain[J]. Journal of Cerebral Blood Flow & Metabolism, 2001, 21(10): 1133-1145.
- [39] Lennie P. The cost of cortical computation[J]. Current biology, 2003, 13(6): 493-497.
- [40] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[C]. Aistats. 2011, 15(106): 275.
- [41] Bengio Y, Courville A, Vincent P. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2013, 35 (8): 1798–1828. arXiv:1206.5538
- [42] Glauner P. Deep Convolutional Neural Networks for Smile Recognition (MSc Thesis). Imperial College London, Department of Computing. 2015. arXiv:1508.06535
- [43] Song H A, Lee S Y. Hierarchical representation using NMF[C]//International Conference on Neural Information Processing. Springer Berlin Heidelberg, 2013: 466-473.
- [44] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786): 504-507.
- [45] Masci J, Meier U, Cireşan D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction[C]//International Conference on Artificial Neural Networks. Springer Berlin Heidelberg, 2011: 52-59.
- [46] 张丹. 面向异构体系结构的任务流化技术研究[D]. 中国人民解放军信息工程大学, 解放军信息工程大学, 2012.

- [47] Abadi M, Agarwal A, Barham P, et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015[J]. Software available from tensorflow. org, 2015, 1.
- [48] Chollet F. Keras (2015)[J]. URL <http://keras.io>.
- [49] Zheng Y, Liu Q, Chen E, et al. Exploiting multi-channels deep convolutional neural networks for multivariate time series classification[J]. Frontiers of Computer Science, 2016, 10(1): 96-112.
- [50] Davis J, Goadrich M. The relationship between Precision-Recall and ROC curves[C]//Proceedings of the 23rd international conference on Machine learning. ACM, 2006: 233-240.
- [51] Powers D M. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation[J]. 2011.Goutte C, Gaussier E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation[C]//European Conference on Information Retrieval. Springer Berlin Heidelberg, 2005: 345-359.
- [52] Goutte C, Gaussier E. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation[C]//European Conference on Information Retrieval. Springer Berlin Heidelberg, 2005: 345-359.
- [53] Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: Machine learning in Python[J]. Journal of Machine Learning Research, 2011, 12(Oct): 2825-2830.
- [54] Sutton R S, Barto A G. Reinforcement learning: An introduction[M]. Cambridge: MIT press, 1998.

攻读学位期间发表的学术论文目录

- [1] 王臻, 李晓宇, 李书芳. 应用于电网电压稳定分析的的海量 WAMS 数据实时分析系统 [EB/OL]. 北京: 中国科技论文在线 [2017-01-25].
<http://www.paper.edu.cn/releasepaper/content/201701-294>.
- [2] 李书芳, 王臻. 一种数据处理的方法、装置及系统[P]. 中国: 2016104709904, 2016-11-02