**电 子 科 技 大 学**

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

**学士学位论文**

**BACHELOR THESIS**



**论文题目基于机器学习的行波管放大器计算程序的评估与参数优**

**化**

**学 院 电子科学与工程学院**

**专 业 物理电子**

**学 号 2019021407008**

**作者姓名 许千龙**

**指导教师 殷海荣　　副教授**

摘 要

为了适应日益增长的宽带信号和非线性系统的工程应用，用于分析瞬态电磁散射问题的时域积分方程方法研究日趋活跃。本文以时域积分方程时间步进算法及其快速算法为研究课题，重点研究了时间步进算法的数值实现技术、后时稳定性问题以及两层平面波算法加速计算等，主要研究内容分为四部分。

……

**关键词：**时域电磁散射，时域积分方程，时间步进算法，后时不稳定性，时域平面波算法

ABSTRACT

With the widespread engineering applications ranging from broadband signals and non-linear systems, time-domain integral equations (TDIE) methods for analyzing transient electromagnetic scattering problems are becoming widely used nowadays. TDIE-based marching-on-in-time (MOT) scheme and its fast algorithm are researched in this dissertation, including the numerical techniques of MOT scheme, late-time stability of MOT scheme, and two-level PWTD-enhanced MOT scheme. The contents are divided into four parts shown as follows.

……

**Keywords:** time-domain electromagnetic scattering, time-domain integral equation (TDIE), marching-on in-time (MOT) scheme, late·-time instability, plane wave time-domain (PWTD) algorithm

目 录

[第一章 绪 论 1](#_Toc466640616)

[1.1 研究工作的背景与意义 1](#_Toc466640617)

[1.2 国内外研究历史与现状 1](#_Toc466640618)

[1.3 本文的主要贡献与创新 1](#_Toc466640619)

[1.4 本论文的结构安排 1](#_Toc466640620)

[第二章 软件功能介绍 2](#_Toc466640621)

[2.1 时域积分方程的类型 2](#_Toc466640622)

[2.2空间基函数与时间基函数 2](#_Toc466640623)

[2.2.1 空间基函数 2](#_Toc466640624)

[2.2.2 时间基函数 3](#_Toc466640625)

[2.3 入射波 3](#_Toc466640626)

[2.4 本章小结 3](#_Toc466640627)

[第三章 程序评估与参数优化 4](#_Toc466640628)

[3.1 时域积分方程时间步进算法的阻抗元素精确计算 4](#_Toc466640629)

[3.2 时域积分方程时间步进算法阻抗矩阵的存储 4](#_Toc466640630)

[3.2.1 时域积分方程时间步进算法产生的阻抗矩阵的特征 4](#_Toc466640631)

[3.2.2 数值算例与分析 4](#_Toc466640632)

[3.3 时域积分方程时间步进算法矩阵方程的求解 5](#_Toc466640633)

[3.4 本章小结 5](#_Toc466640634)

[第四章 全文总结与展望 6](#_Toc466640635)

[4.1 全文总结 6](#_Toc466640636)

[4.2 后续工作展望 6](#_Toc466640637)

[致 谢 7](#_Toc466640638)

[参考文献 8](#_Toc466640639)

[外文资料原文 9](#_Toc466640640)

[外文资料译文 10](#_Toc466640641)

第一章 绪论

1.1研究工作的背景与意义

行波管（Traveling Wave Tube，TWT）是一种电子管，由美国GE公司于1940年代末发明。行波管作为一种微波放大器，广泛应用于雷达、卫星通信、广播电视、导航定位等领域。从20世纪上半叶开始，由于无线电通信的快速发展，需要越来越高的频率和更大量级的功率，而传统的电子管并不能满足要求。行波管的诞生，实现了在微波领域的大功率、高效率放大，填补了电子管和半导体器件之间功率输出的真空，扩大了微波技术的应用领域。

行波管是一种以“慢波系统”作为放大器的一种特殊电子管。慢波系统将电上电磁波引导到行波管内，使这个系统能够将输入的高频波信号在整个行波管中传播，维持了暴增长的相互作用过程，实现了电磁波能量的转化。慢波系统的典型构成方法有螺旋线、螺旋槽及螺旋延长鳍等结构，它们通过电磁波与慢波的相互作用，实现了信号的放大。在行波管技术的发展过程中，关键技术不断创新，如高温电子器件技术、灰色铵盐涂层技术、材料和微纳加工技术等。这些技术的应用使得行波管的动态范围、频带宽度、工作电压水平、线路匹配水平均得到了提高，同时使得行波管得到了非常广泛的应用。

作为传输微波的放大器，行波管的技术应用已经非常广泛，如雷达、卫星通信、无线电侦察、卫星导航、天文观测、脑科学研究等。行波管具有机械结构牢固、波形失真小、寿命长、抗辐照能力强等特点，具有广阔的应用前景。行波管放大器在卫星通信中得到了大规模的应用，在国防和社会方面发挥着重要的作用。随着信息技术的不断迅速发展，微波技术的发展也变得尤为重要。作为微波放大器、微波发生器等部件，行波管在信息技术领域中具有不可替代的地位。行波管在卫星通信、高速数据传输、脑科学研究等领域中的应用，将会产生一系列的经济、社会和科学的效益，为人类事业的发展做出积极的贡献。继续深入行波管研究，不仅能让行波管技术更加完善，还可以催生出更多的行波管新应用，带动相关产业快速发展。

随着计算机和互联网发展的迅速，人工智能技术日益成熟，深度学习成为其中最为重要的分支之一。深度学习是人工智能的一个重要子领域，它是一种基于多层神经网络的机器学习方法，可以在不断学习中根据数据自动调整模型参数，进而实现模型的优化和预测分析。深度学习能够利用大量数据进行训练，并从中自动发现隐含的规律和特征，从而提高了数据处理的效率和准确性，广泛应用于自然语言处理、语音识别、图像识别、机器翻译等方面。在以往的机器学习领域中，数据和特征提取通常需要人工手动处理，而深度学习的出现，打破了人工智能在数据处理及其相关技术上的局限性，通过人工智能不断学习更新模型，模型能够自从判断特征值，完成自我调整，从而使得机器在处理数据时，不再需要人工手动干预特征提取。深度学习基于人工神经网络，其核心是层次化的学习，最终能够使得机器具有自我学习和自我适应的能力，达到人工智能的目的。  
 深度学习跨学科的交叉组合了计算机科学、数学、统计学等学科，致力于研究解决机器智能和智能算法的问题。深度学习在人工智能领域中应用广泛，其核心原理在于训练深度神经网络模型，自动发掘和学习特征，可解决多维数据的识别、自然语言处理、语音识别、人脸识别、图像识别、机器翻译以及自动控制系统等问题。深度学习不仅可以应用于科学研究，也可以用于商业和工业的生产和管理，甚至利用深度学习来预测股市走势以及完成自动化投资等。  
 深度学习的意义在于通过高效的数据学习方法，让机器能够应对人们处理数据的几乎所有需求，根据实时数据快速学习适应，成为人类的智能工具。深度学习技术可以在医疗诊断、智能汽车驾驶、智能物流、智能家居等的领域发挥作用，提高效率和准确性，降低成本和风险。深度学习不仅可以通过模拟真实情况来达到人类认知/决策的效果，还可以与人类进行智能交互，满足用户个性化需求，并为人们提供更好的生活体验。总之，深度学习是现代人工智能技术的重要组成部分，能够极大地拓展机器处理数据的能力，解决实际问题，提高社会经济效益，同时也将有助于推动机器智能化的发展，使人工智能得到进一步的普及和应用。

1.2 国内外研究历史与现状

1.2.1 行波管的国内外研究历史与现状

20世纪初期，美国工程师张衡(Henry Huang，中译名淦隴)和如花舞厅所有者和爱好者维廉·帕克(Willian C. P. Holzman)，合作发明了行波管。1921年，他们申请了相关专利并搬到了欧洲，带着他们的发明向欧洲学者展示。同年，欧洲著名物理学家麦克斯·普朗克(Max Planck)参加了一次公开演讲，称行波管是“让他目瞪口呆的东西”。 20世纪50年代，随着航空航天工业在冷战和太空竞赛中兴起，行波管逐渐得到推广。美国国家航空航天局(NASA)和其他国际航空航天机构开始使用行波管。60年代，美国Raytheon公司(雷神公司)为美国政府制造了第一颗卫星，使用了电子学技术，行波管是其中的核心部件。1937年Philips成功开发出一种基于行波管原理的射频放大器，开发了欧洲工业微波技术的先河，这也被认为是行波管在微波电子学中应用的开端。进入21世纪，行波管逐渐被一些新型的微波放大器部件所取代。但在高功率、高频带宽、低噪声等方面，行波管仍然具有优势。现在，行波管主要应用在科研、军事、航天、通信等领域。

20世纪50年代，中国开始着手发展微波技术。1956年开始，中国科学院电子学研究所组建微波技术研究组，开始开展微波技术研究。1960年，中国开展了第一次微波技术探月，使用的就是行波管。随着国家经济的快速发展，电子科技也得到了迅猛发展。在20世纪80年代之前，国内行波管还处于模仿和跟踪国外技术的阶段。20世纪80年代之后，国内电子科技逐渐走向自主创新。1989年，中国推出了第一款国产行波管-E188CC，标志着中国航天领域在微波技术上的实质性突破。21世纪以来，随着国内微波通信技术的发展，各大企业和研究机构更加注重行波管领域的创新和研究成果的转化。2016年，中国科学院微波技术研究所研制成功了具有全球领先水平的低噪声行波管，首次实现了我国低噪声行波管的量产。作为一种重要的微波电子学元器件，行波管在国内外都取得了不小的进展。目前，国内外行波管科研领域的专家学者密切合作，保持着各自的研究进展，致力于推动行波管领域的发展。

目前用于分析行波管注波互作用的理论体系，主要包括行波管小信号理论和行波管大信号理论。行波管小信号理论，又称为线性理论，最先由皮尔斯提出，该理论基于等效电路法，分析了三个正向波和一个反向波，但是小信号理论忽略了相位变化对注波互作用的影响，无法准确分析出行波管注波互作用中的非线性变换过程[2]。

行波管小信号理论，又称为线性理论，最先由皮尔斯提出，该理论基于等效电路法，分析了三个正向波和一个反向波，但是小信号理论忽略了相位变化对注波互作用的影响，无法准确分析出行波管注波互作用中的非线性变换过程。

行波管的大信号理论在上世纪五十年代提出，大信号理论可以更好地描述注波互作用的非线性特征，因此，大信号理论也可以称为非线性理论。在舍弃小信号的假设后，需要考虑空间电荷之间的作用力，田炳耕、诺埃等人都对空间电荷场进行了推导求解。二者理论的相似点为，田炳耕与诺埃利用等效传输线路方法推导激发方程，将慢波结构等效成由电阻、电容、电感等组成的高频电路。而在求解空间电荷场时，田炳耕将电子注等效为刚性电荷圆盘，通过格林函数法对空间电荷场进行求解。当粒子运动复杂性增加后，空间电荷场的计算就更加困难，可以通过粒子模拟的方法进行计算。粒子模拟的原理是，在电磁场环境中，给出带电粒子的初始位置以及初始速度后，可以通过求解麦克斯韦方程组得到电磁场分布情况；带电粒子由于受到电磁场的作用力而发生变化，因此需要通过求解运动方程以求得粒子变化后的位置。不断重复以上的计算过程，最终就得到了空间各点处电磁场与带电粒子随时间变化的情况。粒子模拟是目前通过数值模拟方法研究带电粒子与场的相互作用的一种常见工具，如CST粒子工作室、KARAT、MAGIC软件等，均结合了PIC方法对注波互作用过程进行模拟，这种方法能够具体反映整个系统的发展变化过程，精确度较高，然而花费时间较长。还有一种分析方法是建立参数化非线性注波互作用代码，如Christine模型，Freund等人开发的三维频域代码Genesis等，这些模型使用场叠加的方法，将射频场表示为慢波系统中存在的不同频率、不同模式的线性叠加形式。Christine模型分为一维Christine频域代码与三维Christine频域代码，两种模型对空间电荷场的计算方法不同。而两种三维频域代码中均结合了粒子模拟的方法，计算出的空间电荷场更加精确。这种使用计算机程序进行模拟计算的方法具有计算时间短，精度高的优点。

1.2.2 深度学习的国内外研究历史与现状

深度学习的历史可以追溯到20世纪80年代，神经网络就是当时研究的核心。但由于计算资源、数据集和算法等方面的限制，神经网络在90年代中期之前并没有得到广泛应用。随着计算机性能不断提升、数据集不断壮大，深度学习也逐渐迎来重大转机。2006年，深度学习的先驱杰弗里·辛顿（Geoffrey Hinton）等科学家提出了深度置信网络（Deep Belief Network），并应用于手写数字的图像识别中，取得了优异的结果。之后，基于深度学习的卷积神经网络（CNN）逐渐被引入到图像识别领域中，并逐渐成为主流技术。2012年，谷歌的Jeff Dean等人成功地训练了一个拥有16000个CPU核和1200个硬盘的深度学习网络，并用于识别YouTube上的视频，取得了重大突破。同年，基于深度学习的AlexNet在ImageNet图像识别竞赛中成功击败了传统的机器学习算法，标志着深度学习在图像识别领域的崛起。继续研究和发展深度学习算法的科学家们，不断提出更为复杂和高效的深度学习模型。例如2014，Ian Goodfellow等人提出了生成对抗网络（GAN），开创了新型神经网络的研究方向。这些研究者的努力，推动了深度学习技术在语音识别、自然语言处理、推荐系统等各个领域中的广泛应用。

我国在深度学习领域的研究和应用也得到了迅速的发展。2006年，百度AI实验室李飞飞团队提出了一种用于语音识别的深度学习模型，称之为“深度网络解码器”。该模型得到了突破性的进展，并于2010年应用于汉语语音识别。随后，国内的科研机构和企业纷纷投入到深度学习领域的研究和应用，例如华为、阿里巴巴、腾讯等。多家企业开发了自己的深度学习框架，例如阿里巴巴的Aliyun、百度的PaddlePaddle和腾讯的Tencent AI等。在各个领域如图像识别、智能语音和自然语言处理等方面，国内的深度学习技术已经逐渐赶上了国外的领先水平。

深度学习的应用领域较为广泛，已经覆盖了图像识别、自然语言处理、语音识别、控制系统等多个领域。在各个领域，国外的深度学习技术已经取得了很大的成功。例如，在图像识别领域，深度学习技术在检测和识别各种物体，人脸识别，场景语义理解等方面都取得了重大进展。Google开发的图像处理引擎TensorFlow，被运用在大规模计算机视觉、语音识别、自然语言处理、机器翻译等多个领域，成为了深度学习领域的重要技术。在图像识别方面，国内已经取得了很大的成功。华为自研的神经网络框架MindSpore在图像识别方面具有出色表现，阿里巴巴的深度学习平台PAI也得到了广泛应用。在自然语言处理方面，腾讯AI的智能聊天机器人“小薇”已经成为了一大亮点。另外，国内企业也在深度学习领域积极合作。例如，百度和前沿智能技术公司YITU合作，研制出了应用于智能安防领域的人脸识别技术，世界领先；阿里巴巴与华为合作，共同研发AI芯片Atlas，展示了国内在AI基础设施方面的新突破。另外，深度学习技术在自然语言处理方面也取得了很大的成功。包括机器翻译、文本分类、语义分析、信息抽取、对话系统等。例如，Facebook开发的机器翻译平台神经机器翻译（NMT）已经取得了令人瞩目的翻译效果。在控制系统方面，深度学习技术已广泛应用于自主驾驶汽车、机器人等机器人方向，从而实现智能的决策和行动能力。

在通信领域，毫米波通信与大规模多输入多输出技术是未来无线通信的主要研究方向，将两者融合应用能够大幅提升系统容量、频谱资源利用率和传输速度，目前有许多研究人员结合深度学习的方法进行探究。

基于上述行波管注波互作用理论，近年来国内外的学者们继续研究发展，并在多信号输入或考虑相对论效应的条件下，开发了各种用于行波管分析的理论与代码。国内的毫米波太赫兹源技术及应用团队已实现将深度学习应用于行波管非线性模型，并且开发了基于机器学习的行波管放大器计算软件。

1.3 本论文的主要贡献与创新

……

1.4 本论文的结构安排

本文的章节结构安排如下：

第一章 绪论。介绍了本论文研究的背景与意义，对行波管大信号模型及深度学习的发展进行了概述，说明了本论文的主要贡献和创新点。

第二章 基于深度学习的行波管大信号模型。本章利用一维行波管大信号理论编写的C++程序，分别收集了6-9GHz单段螺旋线行波管和208-212GHz单段耦合腔行波管的数据，并对其进行预处理，包括核函数、特征选择、归一化、数据储存方式等，使输入输出参量之间存在一定的线性关系。其次搭建深度学习模型并训练，根据训练结果调整超参数，得到最优网络模型。最后利用训练好的模型预测新数据，并与C++程序的计算数据对比验证。

第三章 二级训练算法。本章提出了针对实验数据的二级训练算法，即根据实验数据计算出修正电压和耦合阻抗的修正系数，训练得到用于计算修正系数的深度学习模型，利用修正系数对相应参量进行修正处理，再输入到C++程序中计算得到行波管输出功率，减小预测值与实验数据之间的误差。

第四章 界面程序开发。开发了一款具有存储数据、深度学习、计算行波管输出功率等功能的应用程序，本章内容主要介绍了该程序的功能规划，框架设计和功能展示。

第五章 全文总结与展望。

第2章 6-9GHz螺旋线行波管的功能实现

* 1. 6-9Ghz 螺旋形行波管介绍

## 2.1.1数据集

训练所用数据集存放在服务器路径/share/twtlargesignal/lnk/kernel2/6-9/data中。

管长固定为0.2m，工作电流范围为0.01A-1A，输入功率范围为1mW-1W，其余输入参量的取值范围如表1-1所示，耦合阻抗的范围以CST仿真软件计算所得值为中心±5倍，上限为120Ω，工作电压范围以根据式（2-49）计算所得理论工作电压为中心±200V，共收集到48万条数据。

表1-1 6GHz-9GHz数据集的部分参量取值范围

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **频率（GHz）** | **相速（m/s）** | **耦合阻抗范围（Ω）** | **理论工作电压（V）** | **工作电压范围（V）** |
| 6 | 3.2724e7 | 7.70—120.00 | 3034 | 2834—3234 |
| 6 | 3.4130e7 | 4.17—104.25 | 3310 | 3110—3510 |
| 6 | 3.4653e7 | 8.52—120.00 | 3412 | 3212—3612 |
| 7 | 3.2595e7 | 5.35—120.00 | 3019 | 2819—3219 |
| 7 | 3.4109e7 | 2.56—64.01 | 3306 | 3106—3506 |
| 7 | 3.4626e7 | 6.74—120.00 | 3407 | 3207—3407 |
| 8 | 3.2487e7 | 3.47—86.74 | 2999 | 2799—3199 |
| 8 | 3.4109e7 | 1.48—36.91 | 3306 | 3106—3506 |
| 8 | 3.5772e7 | 1.68—42.11 | 3636 | 3436—3836 |
| 9 | 3.2400e7 | 2.15—53.83 | 2983 | 2783—3183 |
| 9 | 3.4572e7 | 3.77—94.36 | 3396 | 3196—3596 |
| 9 | 3.5559e7 | 4.43—120.00 | 3593 | 3393—3793 |

## 2.1.2 核函数变换

如果只考虑行波管工作电流、电压与耦合阻抗，我们可以得到如下关系式：

(1-1)

上式表示在输出功率饱和点处的输入输出关系，且

(1-2)

考虑相速与工作电压不同步情况，对2-1式进行修正：

(1-3)

在小信号激励阶段，行波管的输入输出功率之间存在如下关系：

(1-4)

其中表示管长，是电子波数，近似等于线路上的慢波波数。增益参量C的表达式如下：

(1-5)

**1.修正系数D**

根据式1-1与1-3，我们发现 , 修正系数D可以通过, 得到。而与工作电压的关系如公式1-2所示。

将1-3的关系写为如下形式：

(1-6)

如果将 看做不随 和 变化的常量，那么第三项也可以看做是一个常数系数，即

(1-7)

我们将依照上式对进行计算，得到与D成正比关系的一组数据dt，并取这一组数据的三次方根，记为d，每一个d都对应一组与，共有samples个d，samples表示读取的样本数。

对应程序如图1-1所示，在数据处理时，我们令，让等于一个较大的数，是为了避免因为精度较低而出现计算结果为0的情况。

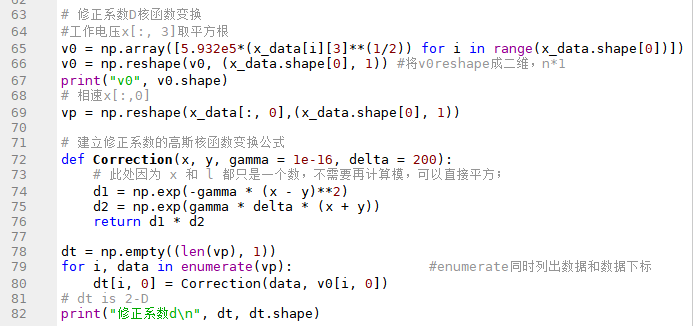


图1-1 处理修正系数D的程序截图

**2.工作电压，工作电流，耦合阻抗，输入功率的处理**

从式（1-1）中可以知道，在考虑饱和功率点附近的输出功率时，这三个输入参量都与输出功率的三次方有一定关系，因此我们先计算这三个变量的三次方根，再加入修正因子d，对电压，电流，耦合阻抗与修正系数的三次方根（ev,ei,kc,d）做阶数为4的多项式扩展，其中不仅包括公式1-1中对应的这一项，还考虑了可能与输出功率线性相关性较高的其他可能的项，即

(1-8)

而对于小信号激励段的行波管，工作电压，工作电流，耦合阻抗与输出功率的关系更接近公式（1-4）、（1-5）所示的情况，此时可以把输入功率考虑进来，首先对、修正系数d的三次方根这四个系数做指数变换，再加入输入功率Pin对这五个参数做多项式扩展，即

(1-9)

这里的表示引入的常数系数。对应程序如图1-2所示。

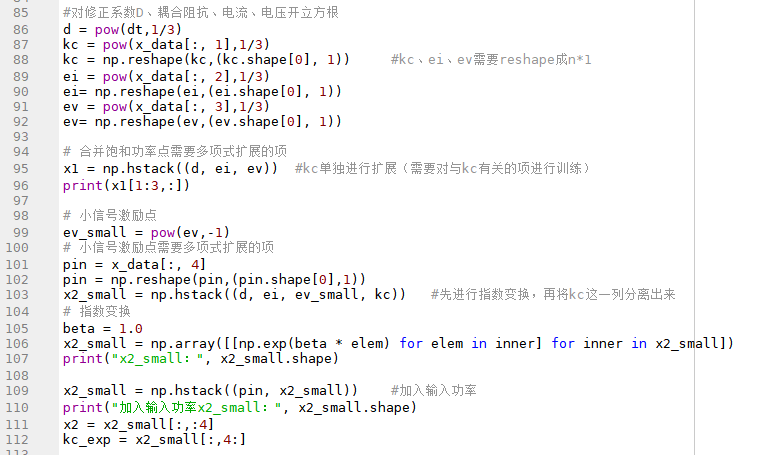


图1-2 工作电压，工作电流，耦合阻抗，输入功率处理的程序截图

由于需要对kc有关项进行训练，考虑将kc单独进行多项式扩展，具体实现思路是，首先将除kc项以外的参数进行多项式扩展，让kc的幂分别乘以扩展后的参数。程序如图1-3所示，函数polyExtension实现kc项的扩展，多项式扩展后要通过np.unique()函数删除重复列，并将列索引值保存至txt文件,便于今后根据现有的处理规则对数据进行预处理。

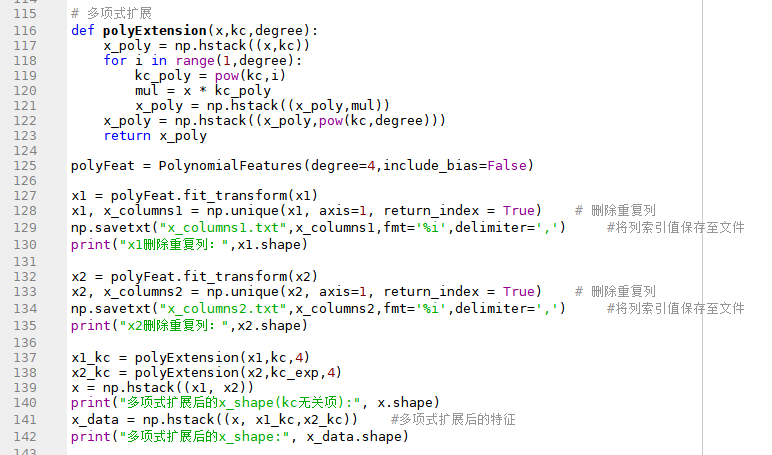


图1-3 多项式扩展的程序截图

x1删除重复列后有34项特征，x2删除重复列后有69项特征；多项式扩展后的x\_shape(kc无关项)有103项，多项式扩展后的x\_shape有519项。

## 2.1.3数据集标准化

对多项式扩展后的数据进行标准化处理，同样地，需要将标准化的规则（平均值和方差）保存至txt文件中。然后将数据集划分为训练集、验证集、测试集。代码如图1-4所示。

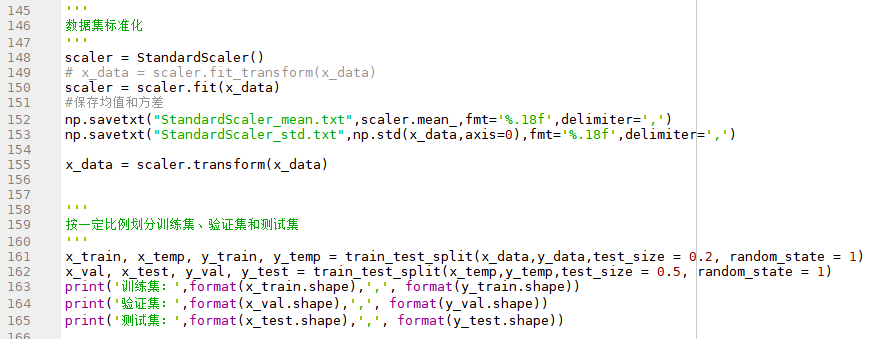


图1-4 数据集标准化和划分训练集的程序截图

## 1特征选择

对于相关性强的两个或多个特征来说，它们会将冗余信息传递给模型，因此减少特征具有重要的现实意义，不仅减少过拟合、提高模型泛化能力，而且还可以使模型获得更好的解释性，增强对特征和特征值之间的理解，加快模型的训练速度，一般还会获得更好的性能。因此要对多项式扩展后的特征进行特征选择。

首先计算各个特征之间的斯皮尔曼相关系数，删除相关性大于0.99的特征，并将删除项的列索引值保存至txt文件。代码如图1-5所示，经过特征选择后，共有75项特征参量，其中kc无关项有15项，kc有关项有60项。



图1-5 特征选择的程序截图

## TFRecord格式

在每次开始训练前，需要将数据集加载进内存，并进行特征变量扩展和特征选择的处理（如1.1节内容所述），随着数据量增多，数据预处理的时间也大大增加，以现在30万组数据为例，全部处理完需要2个多小时，不仅极大地影响了训练效率，并且会占用很大的内存，导致训练中断。

考虑tensorflow中的tf.data模块，它包括了一套灵活的数据集构建API，可以快速高效地构建数据输入的流水线，适用于数据量巨大的场景。

1.2节中涉及到的代码均在dataPreProcessing.py文件中。主要包括加载原始数据集、特征多项式扩展、标准化及数据集划分、特征选择、处理后的数据保存为TFRecord格式、读取TFRecord文件、并处理为可用于训练的tf.data.Dataset对象，这几个部分。

1. TFRecord 数据集存储格式

从硬盘里的txt或csv文件读取，IO操作比较耗时，TFRecord是tensorflow中的数据集存储格式，是一种能将data和label一起存储的二进制文件，能更好地利用内存，在tensorflow的graph中更快地复制、移动、读取。

TFRecord可以理解为一系列序列化的tf.train.Example元素组成的列表文件，而每一个tf.train.Example又由若干个tf.train.Feature的字典组成。

为了将数据集整理为TFRecord格式，主要步骤如下：

1. 读取数据元素到内存。
2. 先建立Feature字典，将该元素转换为tf.train.Example对象。
3. 将tf.train.Example对象序列化为字符串，并通过一个预先定义的tf.io.TFRecordWriter写入TFRecord文件。

将上述步骤以代码实现，定义一个函数save\_tfrecords()，输入参量为特征变量x、标签值y、TFRecord文件保存路径。代码如下：

def save\_tfrecords(data, label, desfile):

with tf.io.TFRecordWriter(desfile) as writer:

for i in range(len(data)):

features = tf.train.Features(

feature = {

"data":

tf.train.Feature(bytes\_list = tf.train.BytesList(value = [data[i].astype(np.float32).tobytes()])),

"label":

tf.train.Feature(bytes\_list = tf.train.BytesList(value = [label[i].astype(np.float32).tobytes()]))

}

)

example = tf.train.Example(features = features)

serialized = example.SerializeToString()

writer.write(serialized)

writer.close()

save\_tfrecords(x\_train, y\_train, "./data\_train.tfrecords")

save\_tfrecords(x\_val, y\_val, "./data\_val.tfrecords")

值得注意的是tf.train.Feature支持三种数据格式：

①tf.train.BytesList:字符串或原始byte文件（如图片），通过bytes\_list参数传入一个由字符串数组初始化的tf.train.BytesList对象。

②tf.train.FloatList：浮点数，通过float\_list参数传入一个由浮点数数组初始化的tf.train.FloatList对象。

③tf.train.Int64List：整数，通过int64\_list参数传入一个由整数数组初始化的tf.train.Int64List对象。

而读取TFRecord文件则可按照以下步骤：

1. 通过tf.data.TFRecordDataset读入原始的TFRecord文件，获得一个tf.data.Dataset对象。
2. 通过Dataset.map方法，对该数据集对象中的每个序列化的tf.train.Example字符串执行tf.io.parse\_single\_example函数，从而实现反序列化。

读取TFRecord文件的代码如下，定义两个函数\_parse\_function()和load\_tfrecords()。函数\_parse\_function()实现了第二步的功能，即反序列化，其中tf.io.decode\_raw()函数里的第二个参数，也就是解析后的类型,之前的数据在转成字符串之前是什么类型，这里的参数就要填成对应的类型，否则会报错。

def \_parse\_function(example\_proto):

features = {"data": tf.io.FixedLenFeature((), tf.string),

"label": tf.io.FixedLenFeature((), tf.string)}

parsed\_features = tf.io.parse\_single\_example(example\_proto, features)

label = tf.io.decode\_raw(parsed\_features['label'], tf.float32)

data = tf.io.decode\_raw(parsed\_features['data'], tf.float32)

return data, label

def load\_tfrecords(srcfile):

dataset = tf.data.TFRecordDataset(srcfile) # load tfrecord file

dataset = dataset.map(\_parse\_function) # parse data into tensor

for feature, label in dataset.take(2):

print(feature)

print(label)

return dataset

train\_dataset = load\_tfrecords("./data\_train.tfrecords")

val\_dataset = load\_tfrecords("./data\_val.tfrecords")

1. tf.data.Dataset

tf.data的核心是tf.data.Dataset 类，提供了对数据集的高层封装。tf.data.Dataset由一系列的可迭代访问的元素组成，每个元素包含一个或多个张量。一般构建tf.data.Dataset的方法是使用 from\_tensor\_slices() 、from\_generator(）方法，在此不作赘述。

Dataset.batch(batch\_size)方法可以将数据集分成批次，即对每batch\_size个元素，使用tf.stack()在第0维合并，成为一个元素。

tf.data的数据集对象提供了Dataset.prefetch()预加载方法，我们可以让数据集对象Dataset在训练时预取出若干个元素，使得在GPU训练的同时CPU可以准备数据，从而提升训练流程的效率。Dataset.prefetch()中的参数buffer\_size既可以手动设置，也可设置为tf.data.experimental.AUTOTUNE，即由tensorflow自动选择合适的数值。通过图1-1可以看出两种训练流程的区别。

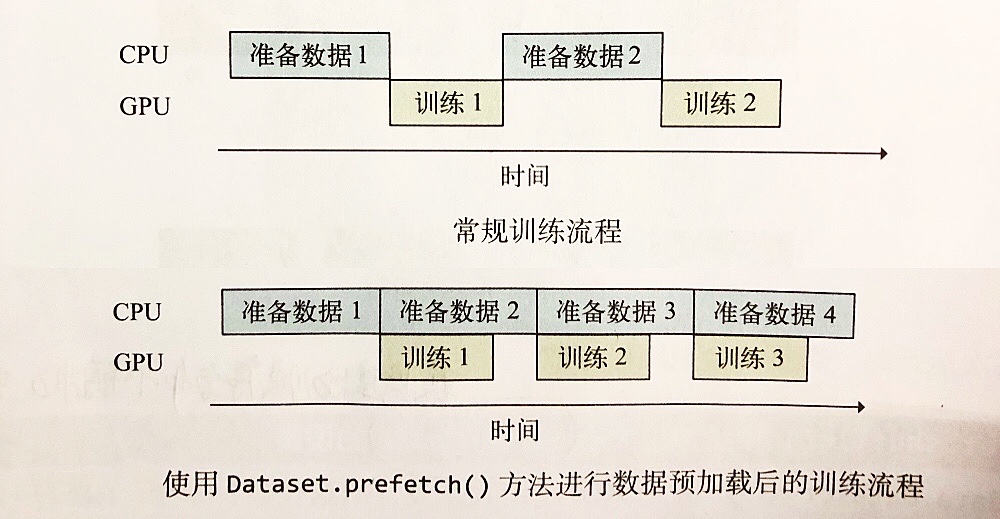


图1-1 两种训练流程对比

Keras支持使用tf.data.Dataset直接作为输入。当调用tf.keras.Model的fit()和evaluate()方法时，可以将参数中的输入数据x指定为一个元素格式为 (输入数据, 标签数据) 的Dataset，并忽略掉参数中的标签数据y。比如：model.fit(dataset,epochs=num\_epochs),由于已经通过Dataset.batch()方法划分了数据集批次，这里无需再提供batch的大小.

通过1.2.3节中读取TFRecord文件，已经得到了一个数据集对象dataset，只需通过Dataset.batch()和Dataset.prefetch()方法处理一下即可用于训练，代码如下：

train\_dataset = load\_tfrecords("./data\_train.tfrecords")

train\_dataset = train\_dataset.batch(200)

train\_dataset = train\_dataset.prefetch(tf.data.experimental.AUTOTUNE)

for feature, label in train\_dataset.take(1):

print(feature)

print(label)

val\_dataset = load\_tfrecords("./data\_val.tfrecords")

val\_dataset = val\_dataset.batch(200)

经过这样的处理后程序加载TFRecord文件的速度不到1秒，相比之前数据处理的时间大大提升。

1.4 时域方法特有的展开函数

……

2.2.2.2 频域方法特有的展开函数

……

致 谢

本论文的工作是在我的导师XX老师悉心指导下完成的，……

……

参考文献

1. W. C. Chew, J. M. Jin, E. Michielssen, et al. Fast and efficient algorithms in computational electromagnetics[M]. Boston: Artech House, 2000
2. 盛新庆.计算电磁学要论[M].北京:科学出版社, 2004
3. 王秉中.计算电磁学[M].北京:科学出版社, 2001
4. 吕英华.计算电磁学的数值方法[M].北京:清华大学出版社, 2006
5. 王长清.现代计算电磁学基础[M].北京:北京大学出版社, 2005
6. 潘小敏.计算电磁学中的并行技术及其应用[D].北京:中国科学院电子学研究所, 2006
7. 中华人民共和国国家技术监督局.GB3100-3102.中华人民共和国国家标准--量与单位[S]. 北京:中国标准出版社, 1994年11月1日
8. W. C. Gibson. The method of moments in electromagnetics[M]. New York: Chapman and Hall/CRC, 2008
9. 胡俊.复杂目标矢量电磁散射的高效算法——快速多极子方法及其应用[D].成都:电子科技大学, 2000
10. H. C. Martin, G. F. Carey. Introduction to finite element analysis: theory and application [M]. New York: McGraw Hill, 1973
11. 金建铭 (著), 王建国 (译).电磁场有限元方法[M].西安:西安电子科技大学出版社, 1998
12. M. Clerc. Discrete particle swarm optimization: a fuzzy combinatorial box[EB/OL]. http://clere.maurice.free.fr/pso/Fuzzy\_Discrere\_PSO/Fuzzy\_DPSO.htm, July 16, 2010
13. S. P. Walker, C. Y. Leung. Parallel computation of integral equation methods for three-dimensional transient wave propagation[J]. Communications in Numerical Methods in Engineering, 1997, 11(6): 515-524
14. 肖珍新.一种新型排渣阀调节降温装置[P].中国,实用新型专利,ZL201120085830.0, 2012年4月25日
15. X. F. Liu, B. Z. Wang, W. Shao. A marching-on-in-order scheme for exact attenuation constant extraction of lossy transmission lines[C]. China-Japan Joint Microwave Conference Proceedings, Chengdu, 2006, 527-529

外文资料原文



外文资料译文

基于多载波索引键控的正交多路复用系统的误码率上界

二．基于多载波索引键控的正交频分多路复用系统模型

我们考虑一个端到端的M-QAM，Nc子载波的基于多载波索引键控的正交频分多路复用系统有n个簇，每个簇有N个子载波（Nc=nN）。M-QAM的符号流经过串并转换之后每n个符号组成一个相量，是和传统正交频分多路复用一样是用来调制子载波的，但是不同的是只有这n个活跃子载波进行了调制。……

……