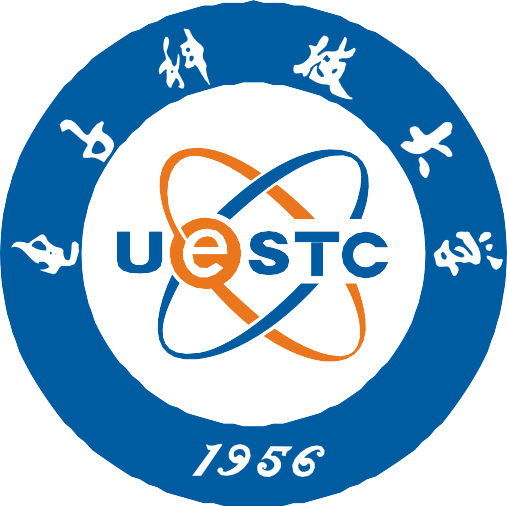
电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

专业学位硕士学位论文

**MASTER THESIS FOR PROFESSIONAL DEGREE**



论文题目 深度学习在探地雷达目标识别中的应用研究

学科专业 电子与通信工程

学 号 201421040223

作者姓名 廖彬彬

指导老师 赵青 教授

分类号 密级

UDC 注 1

学 位 论 文

深度学习在探地雷达目标识别中的应用研究

（题名和副题名）

廖彬彬

（作者姓名）

指导老师 赵青 教授

（姓名、职称、单位名称）

申请学位级别 硕士 学科专业 电子与通信工程提交论文日期 论文答辩日期

学位授予单位和日期 电子科技大学答辩委员会主席

评阅人

注 1：注明《国际十进分类法 UDC》的类号。

摘要

# 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

**ABSTRACT**

With the widespread engineering applications ranging from broadband signals and non-linear systems, time-domain integral equations (TDIE) methods for analyzing tran- sient 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](#_bookmark0) 1

* 1. [研究工作的背景与意义](#_bookmark1) 1
  2. [国内外研究历史与现状](#_bookmark2) 2
     1. [探地雷达发展历史](#_bookmark3) 2
     2. [深度学习研究历史](#_bookmark4) 2
     3. [机器学习以及深度学习在探地雷达中的应用](#_bookmark5) 3
  3. [本文的主要贡献与创新](#_bookmark6) 4
  4. [本论文的结构安排](#_bookmark7) 4

[第二章 深度学习基础](#_bookmark8) 5

* 1. [感知器](#_bookmark9) 5
  2. [激活函数](#_bookmark13) 7
  3. [神经网络结构](#_bookmark16) 9
  4. [反向传播算法](#_bookmark18) 11
  5. [卷积神经网络](#_bookmark30) 15
  6. [本章小结](#_bookmark36) 19

[第三章 地下探测 FDTD 仿真技术研究](#_bookmark37) 20

* 1. [仿真方法与计算平台](#_bookmark39) 21
     1. [FDTD 方法基本原理](#_bookmark40) 22
     2. [Gprmax 仿真平台](#_bookmark44) 24
  2. [拟真土壤模型](#_bookmark45) 25
     1. [拟真土壤模型原理](#_bookmark46) 25
     2. [建模实例](#_bookmark52) 27
  3. [天线模型仿真](#_bookmark55) 29
     1. [FDTD 天线建模方法](#_bookmark56) 29
     2. [仿真实例](#_bookmark58) 31
  4. [B 扫数据的批量生成](#_bookmark63) 33
  5. [本章小结](#_bookmark64) 33

[第四章 时域积分方程数值方法研究](#_bookmark66) 35

* 1. [时域积分方程时间步进算法的阻抗元素精确计算](#_bookmark67) 35
  2. [时域积分方程时间步进算法阻抗矩阵的存储](#_bookmark68) 35
     1. [时域积分方程时间步进算法产生的阻抗矩阵的特征](#_bookmark69) 35
     2. [数值算例与分析](#_bookmark70) 35
  3. [时域积分方程时间步进算法矩阵方程的求解](#_bookmark74) 36
  4. [本章小结](#_bookmark75) 36

[第五章 全文总结与展望](#_bookmark76) 37

* 1. [全文总结](#_bookmark77) 37
  2. [后续工作展望](#_bookmark78) 37

[第六章 template](#_bookmark79) 38

[6.1 s1](#_bookmark80) 38

[6.1.1 sub1](#_bookmark81) 38

[致 谢](#_bookmark82) 39

# 第一章 绪 论

## 研究工作的背景与意义

探地雷达（GPR）使用电磁场探测有损介电材料，以检测介质内内的结构和材料特性的变化。迄今为止，大多数应用都采用天然地下介质，但也会出现在人造复合材料如混凝土，沥青和其他建筑材料中的广泛应用。在这种有损介电材料中，电磁场在被吸收之前会渗透到某个深度。对于 GPR，电磁场作为基本上非色散的波传播。发射的信号穿过介质，被阻抗的变化散射或反射，产生类似于发射信号形状的回波，通过雷达波组成的 B 扫图像便可以观察到地下结构或者目标物的特征。GPR 平台通常工作在 1 MHz 至 1000 MHz 的频率范围内。在低于 1M 的频率下，电磁波具有较大的散射特性，这种工作频率下一般称为电磁感性探测法。在较高频率下，信号在地层中的衰减较大，使得穿透极为有限。在过去的三十年来 [3]，探地雷达广泛应用于地质探测、管道勘查、遗迹探查、道路和隧道勘探、扫雷和爆炸物探测等领域。

目前，探地雷达主要采用瞬态脉冲体制。瞬态脉冲雷达具有较宽的工作带宽， 而且探地雷达所面对的是具有非均匀性、强衰减性及色散效应的地下有耗介质, 电磁波传播环境复杂多变。因此探地雷达接收回波中会有较多的杂波干扰，给后期的数据解释工作带来困难 [4]。已有的探地雷达数据处理方法主要集中在杂波抑制与图像重建两个方面，常见方法有噪声抑制、时变增益、背景去除、滤波、反卷积、 基尔霍夫偏移、Stolt 偏移等等 [5]。这些方法的使用，需要探地雷达研究人员根据不同的工作环境特点与目标特性，选择合适的方法与合适的算法参数以获得理想的处理效果，对研究人员的要求较高。在某些复杂的工作环境下，这些传统方法甚至可能完全达不到预期效果。因此，开发一种对算法使用人员要求较低，能在不同的使用环境下自动提取探地雷达回波信号中关键信息并由此对目标位置、深度、大小、材质等关键参数进行识别的探地雷达处理算法是极其有必要的。

近年来，随着硬件计算能力的增强和研究的不断深入，机器学习领域已发展至深度学习阶段 [6] 。深度学习网络模型已经开始广泛应用于普通的图像的识别和处理。在图像识别中采用深度学习能大大提高其准确性，并且耗时短，从而大大提高了计算效率 [7]。探地雷达的 B 扫数据是由实数组成的二维数组，和数字图像具有相似性。特征提取是雷达数据处理中的一个重要研究内容，而深度学习可以更为有效地对特征进行分层提取。深度学习可以对输入信号进行分层特征提取， 能更好地使用特征表达原始输入，并且这些特征集合代表了原始数据中不同层次

的抽象概念及意义。因此，将深度学习相关方法应用到探地雷达目标识别具有广 阔的应用前景。

## 国内外研究历史与现状

## 探地雷达发展历史

## 深度学习研究历史

深度学习作为机器学习的一个分支，采用算法处理数据并模仿思维过程，或 开发抽象。深度学习使用多层算法来处理数据，理解人类语音并在视觉上识别对 象。信息通过每一层传递，前一层的输出为下一层提供输入。网络中的第一层称 为输入层，而最后一层称为输出层。两者之间的所有层都称为隐藏层。每层通常 是包含一种激活函数的简单统一算法。特征提取是深度学习的另一个方面。特征提取使用算法自动构建数据的有意义的“特征”，以用于训练，学习和理解。

深度学习的历史可以追溯到 1943 年，当时 Walter Pitts 和Warren McCulloch 创建了一个基于人类大脑神经网络的计算机模型。他们使用算法和数学的组合，他 们称之为“阈值逻辑”来模仿思维过程。从那时起，深度学习已经稳步发展，其 发展经历了两次重大突破。

开发深度学习算法的最早努力来自 Alexey Grigoryevich Ivakhnenko 和 Valentin Grigor’evich Lapa。1965 年，他们使用具有多项式激活函数的模型，然后进行统计分析。每一层都会将最佳统计选择的特征转发到下一层。

Kunihiko Fukushima 使用了第一个“卷积神经网络”。Fukushima 设计了具有多个汇集和卷积层的神经网络。1979 年，他开发了一种名为 Neocognitron 的人工神经网络，该网络采用分层的多层设计。这种设计允许计算机“学习”识别视觉 模式。这些网络类似于当代的网络结构，经过多层重复激活的强化策略训练，随 着时间的推移网络性能逐渐增强。此外，Fukushima 的设计允许通过增加某些连接的权值来手动调整重要功能。

反向传播算法，及其在深度学习模型的训练的在应用，在 1970 年开始得到长足发展。1985 年，Rumelhart，Williams 和 Hinton 证明在神经网络中的反向传播可以学习到数据的分布特征。1989 年，Yann LeCun 在贝尔实验室提供了第一次反向传播的实际演示。他将卷积神经网络与反向传播结合到读“手写”数字上。该系 统最终用于读取手写支票的数量。

深度学习的下一个重要里程碑发生在 1999 年，当时计算机开始变得更快处理数据并开发了 GPU（图形处理单元）。使用 GPU 处理图片的处理速度更快，在 10 年的时间内，计算速度提高了 1000 倍。在此期间，神经网络开始与支持向量机竞

争。虽然神经网络与支持向量机相比可能较慢，但神经网络使用相同的数据提供 了更好的结果。随着更多训练数据的添加，神经网络还具有继续改进的优点。

大约在 2000 年，研究者发现“梯度消失”问题。即在较低层中形成的特征没有被上层学习，因为没有学习信号到达这些层。这是基于梯度的学习算法所存在的问题。问题的根源被证明是某些激活函数引起的。许多激活函数限制了它们的输入，从而减小了它们的输出范围。这产生了在极小范围内映射的大范围的输入。在这些输入范围内，大的变化将减少到输出的微小变化，导致梯度消失。用于解决该问题的两种解决方案是逐层预训练和长短期记忆结构的开发。

2009 年，斯坦福大学教授 Fei-Fei Li 发起了 ImageNet，组建了一个包含超过

1400 万张标记图像的免费数据库。互联网充满了未标记的图像。需要标记的图像来“训练”神经网络。李教授说：“我们的愿景是大数据会改变机器学习的方式。数据驱动学习。”

到 2011 年，GPU 的速度显着提高，从而可以使训练卷积神经网络无需逐层预训练。随着计算速度的提高，深度学习在效率和速度方面具有显着优势。其中一 个例子是 AlexNet，这是一个大型卷积神经网络，其架构在 2011 年和 2012 年赢得了多项国际比赛。

## 机器学习以及深度学习在探地雷达中的应用

机器学习在探地雷达信号处理中的早期应用主要基于较简单的浅层神经网络 或支持向量机模型。2001 年，王群等人将前向神经网络应用于探地雷达地雷探测识别，实现对干扰物信号与地雷信号的分离 [15]。2004 年，刘敦文等人运用人工神经网络理论和方法, 建立了用于隧道衬砌厚度探地雷达探测信号解释的 BP 神经网络模型, 对某公路隧道衬砌检测厚度进行了分析应用, 提高了探地雷达信号解释精度和工作效率 [16]。2005 年，电子科技大学胡进峰等人，将多目标识别支撑矢量机与探地雷达目标识别相结合, 得到了基于一对一支撑矢量机的探地雷达多目标识别方法 [17]。

近期，深度学习在探地雷达信号处理方面的应用开始见于文献。2014 年，美国 Besaw 等人提出使用深度信念网络（DBN）对探地雷达数据进行处理 [18]。此文献中，DBN 首先以无监督学习算法预训练，以此获得输入数据的压缩化表示并作为特征识别器。然后，DBN 又被标签数据监督学习，以此获得预测模型。此

DBN 模型成功达到百分之 91 的预测正确率和百分之 1.4 的虚警率。同年，西班牙 Núñez-Nieto 等人将 2.3GHz 和 1GHz 的 MALA 商业探地雷达所采集的数据使用深度感知机进行监督训练，并与传统线性回归方法做比较，取得良好的效果 [19]。

2017 年，Wei Wang 等人利用深度自编码器识别墙后人体目标，并用降噪编码器进一步提高特征表示的效率 [20]。

从上述研究进展可看出，目前机器学习特别是深度学习已在探地雷达数据处 理领域得到初步尝试。但是，绝大部分已有研究将重点放在目标有无的识别上，还 没有对目标的几何特征介质特征做进一步的探索。因此，此论文开展的研究具有 较强的现实意义。

## 本文的主要贡献与创新

本论文以探地雷达地下目标的特征识别为主线，主要创新点与贡献如下：

* + 1. 将土壤的介质参数模型和分形理论结合起来，实现拟真土壤模型的算法生

成。

* + 1. 构思出一系列建模步骤，实现将实际天线模型融入 FDTD 网格仿真过程。
    2. 基于卷积神经网络实现对地下目标的特征识别。

## 本论文的结构安排

本文的章节结构安排如下：

第二章研究深度领域最核心的基础概念和理论。梳理了机器学习相关概念包 括感知器和激活函数的数学意义和提出背景；在介绍前馈神经网络的基础上着重 阐述反向传播算法的原理和流程；最后本章引入卷积神经网络这种网络结构，为 本文后面章节的研究提供理论基础。

第三章研究探地雷达地下目标识别的仿真技术。通过拟真土壤模型的生成算 法和真实天线模型的构建，并结合地下目标模型的批量生成技术得到大量与实际 数据相似的雷达 B 扫图像，为下一章深度学习算法的性能验证提供数据基础。

# 第二章 深度学习基础

人工智能这个学科底下的一个子学科叫做机器学习。机器学习主要是通过概 率统计等方法让机器从大量已有数据中找到规律进而使用这些习得的联系在新数 据上解决问题。这里解决的问题主要有分类和回归问题两类，其中分类问题的结 果是离散的，主要是给出输入数据应该属于预先定义的哪个类别里，而回归问题 的结果是连续，它的输出是关于数据特性的某个数值。机器学习在某些传统算法 束手无策的问题上，比如手写识别，语音辨认等领域，取得令人可喜的效果。人 工神经网络是最常用的实现学习的一类算法，其理论基础是利用神经元的数学模 型来近似模仿生物体大脑处理信息的过程。人工神经网络本质上一种具有自学习 能力的非线性建模工具。

近年来使人工神经网络再度成为研究热点的一类算法叫做深度学习。其名称 里的“深”表示相比于上世纪的人工神经网络，深度学习的网络层数大，结构复 杂。深度学习现在几乎已经成为大型神经网络的代名词。

本章主要介绍深度学习的理论基础，包括感知器、激活函数、反向传播算法、卷积神经网络等相关概念与理论。

## 感知器

感知器（perceptron）是构成神经网络的基础，感知器的概念由 Frank Rosenblatt 在 1950 年代到 1960 年代基于 Warren McCulloch 和 Walter Pitts 的关于人类神经活动的模拟的研究提出并发展而来。如今最常用的感知器被称作 sigmoid 感知器，但是它的基础仍然是最初提出的感知器模型，下面对其原理进行介绍。

𝑥1

𝑥2

𝑥3

output

图 2-1 感知器模型

如图//所示，感知器接收若干个二进制输入 *x*1*, x*2*, x*3*, ...* 并输出一个二进制结

果。Rosenblatt 提出了一种计算感知机输出值的简易准则，他引入“权重（weight）”的概念，用实数 *w*1*, w*2*, ...* 来表示每个输入的重要性。最后神经元的输出，0 或者

∑

1，是由加权和 *j wjxj* 是否小于某个门限值决定的。和权重一样，门限值也是神经元的一个实数参数。以上准则可由下面的公式[2-1](#_bookmark10)表示：

*output* =  0 if ∑*j wjxj* ≤ 门限

 1 if ∑*j wjxj >* 门限

(2-1)

以上就是感知器的简单数学模型。可以把感知器看成是一种对输入特征进行 加权并输出决策的设备。可以结合实际举一个简单的例子。假如某人要决定周末 是否外出游玩，那么影响他决策的因素就可能是以下三个方面：

* + 1. 天气是否晴好？
    2. 是否有同伴陪同？
    3. 游玩地点是否在地铁站附近？

我们可以用变量 *x*1*, x*2*, x*3 来表征这三个二进制影响因素，比如 *x*1 = 1 表示天气晴好，*x*2 = 1 表示有同伴陪同，*x*3 = 1 表示目标地点在地铁站附近。

以上三个影响因素对最后的决定的影响程度是不一样的，这就需要对每个影 响因素设置一个权重。假如天气是最关键的决定因素，在天气坏的情况下仍然出 门的可能性很小，那么就可将 *x*1 对应的权重设置为 *w*1 = 6，其余两个影响因素的权重都设置为 *w*2 = 1*, w*3 = 1。现在假如我们把门限值设置为 3，那么在天气坏的情况下，无论其他两个因素是什么样的，最后的加权和都不会超过 3，从而做出周末不外出游玩这个决定。

通过设置不同的权重和门限，可得到不同的决策模型。比如将门限值降低可 增加外出游玩的可能性，将是否有同伴陪同的权重提高可增加同伴对外出游玩与 否的影响程度。

显然以上的感知器模型并不能完全体现人类决策的复杂性而只是一个简单的 示例。通过堆叠多层感知器，可得到做出更加微妙决策的感知器网络。

如图[2-2](#_bookmark11)，第一层感知器一共做出三项简单决策，这三项决策又接着输入到第二层的两个感知器得出两个更进一步的决策，最后这两个决策又输入到最后一个 感知器得到最后的决策。这种级联方式可以让后面的感知器得出比前面的感知器 更抽象的结果，从而解决更加复杂的决策问题。

𝑥1



𝑥2

𝑥3

output

图 2-2 多层感知器模型

通过引入“偏置（bias）”*b*，公式[2-1](#_bookmark10)也可以改写为公式[2-2](#_bookmark12):

*output* =  0 if *w* · *x* + *b* ≤ 0

 1 if *w* · *x* + *b >* 0

(2-2)

偏置的概念可理解为输出 1 的容易程度。从生物学的角度也可说成神经元达到激发态的容易程度。偏置的引入可简化后面的公式表示。

## 激活函数

本节所介绍的激活函数的引入是为了使人工神经网络中感知器的权重和偏置的自动化训练成为可能。如果我们想通过调整感知器参数从而使网络的整体表现满足某种特定需求，那么对感知器参数的微小改动必须也对应输出的微小改动， 即稍稍改变权重和偏置不会导致输出的剧烈变化。前面的感知器模型只能输出 0

和 1 两种结果，显然不能满足输出缓慢变化的特征。为了得到满足这种特征的感知器，需要对上节感知器模型做出修改，在输出时应用激活函数 *f*：

1

*σ*(*z*) ≡

1 + *e*−*z*

(2-3)

即，对于输入 *x*1*, x*2*, ...*，权重 *w*1*, w*2*, w*3*, ...* 和权重 *b*，sigmoid 神经元的输出是：

1

1 + exp (− ∑*j wjxj* − *b*)

(2-4)

从图[2-3](#_bookmark14)可以看出，本质上 sigmoid 函数是阶跃函数的平滑版本，这也说明了为什么 sigmoid 激活函数的引入可以解决输出剧烈变化的问题。

*o* 函数的重要性在于其平滑性，而与其具体的形状并没有太大的关系，*σ* 的平

1.0 1.0

0.8 0.8

0.6 0.6

sigrnoid(x)

step(x)

0.4 0.4

0.2 0.2

0.0

10 5 0 5 10

x

0.0

10 5 0 5 10

x

(a) (b)

图 2-3 Sigmoid 与阶跃函数对比图

滑度意味着权重中的微小变化 Δ*wj* 和偏置中的 Δ*wj* 将在神经元的输出中产生小的变化 Δ*output*。事实上，微积分告诉我们 Δ*output* 很接近：

output

∑ *∂* +

Δoutput ≈ Δ*wj*

*j*

*∂wj*

*∂* output

*∂b*

Δ*b* (2-5)

*o* 函数并不是唯一的激活函数形式，常用的激活函数还有ReLU 激活函数。此函数是一个分段线性函数：

函数图像是：

ReLU(*x*) =  *x* if *x >* 0

 0 if *x* ≤ 0

(2-6)

从公式[2-6](#_bookmark15)和图[2-4](#_bookmark17)可以观察到 ReLU 函数把所有的负值都变为 0，而正值线性变化，这种操作叫做单侧抑制。尤其体现在深度神经网络模型 (如 CNN) 中，当模型增加 N 层之后，理论上 ReLU 神经元的激活率将降低 2 的 N 次方倍。相对于

sigmoid 函数，ReLU 能更好地实现网络模型的稀疏性，同时也符合近年来神经科学对神经元工作稀疏性的研究【引用】，同时 ReLu 在正区间梯度总是为定值的特性也有利于避免梯度消失问题。

10

8

6

ReLU(x)

4

2

0

10 5 0 5 10

x

图 2-4 ReLU 激活函数图像

## 神经网络结构

单个感知器或者说人工神经元只是生物神经细胞的近似理想化实现，功能更 加简单。要想模仿人类的神经系统的推理运算功能，需要多个人工神经的协调配 合实现高阶的抽象能力并完成复杂的功能。通过特定的连接或信息传递方式进行 配合的神经元可以看作是一个网络，我们称之为人工神经网络。

图[2-5](#_bookmark19)给出了人工神经网络的一般结构。人工神经网络一般由多层感知器级联而成。第一层感知器，也就是图中的最左边那层称为输入层，它的作用是将输入数据传递到网络中。最后一层感知器被称为输出层，也就是图中的最右那层，用来得到网络的计算结果。而输入层和输出层之间的各层统称为隐含层（hidden layer），隐含的意思是这些神经元处于神经网络内部，不与外部直接进行信息交换。上图中的隐含层层数为 2，不过隐含层的层数也可以是任意正整数。历史上这类简单的多层神经网络也被成为多层感知机（multilayer perceptrons）。

目前所提到的神经网络结构都是不包含反馈结构的，即所有网络层的输入 都来自于上一层的输出，这种网络结构被称为前馈神经网络（feedforward neural

networks）。另外一种神经网络类型包含循环结构，即存在某些层的输出又作为前面网络层输入的情况，这种网络结构统称为循环神经网络（recurrent neural networks）。循环神经网络在时序信号的处理，如语音识别领域，取得了良好的效果。

前馈神经网络是应用最广泛的神经网络模型，性能良好，也比较易于实现与 调参，本文后面章节在处理探地雷达信号目标识别问题时使用的卷积神经网络也 属于前馈神经网络的一种。

输入层 隐含层

···

隐含层 输出层

𝑥1

𝑥2

𝑥3

·

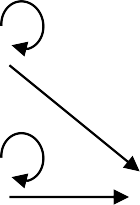
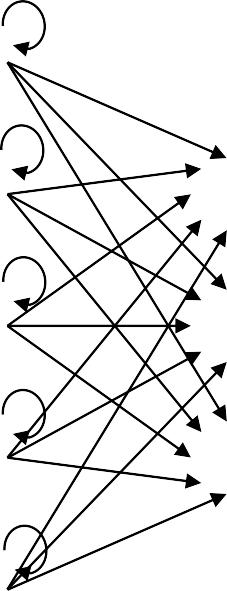
·

·

𝑥𝑛

图 2-5 神经网络一般结构

输入层 隐含层



𝑥1

𝑥2

𝑥3

·

·

·

𝑥𝑛

···

隐含层 输出层

图 2-6 循环神经网络

## 反向传播算法

在将神经网络应用到分类和回归任务时，神经网络的预测性能由每层各神经 元的权重和偏置组成的参数矩阵决定。神经网络的训练过程即用特定的优化方法 调整参数矩阵，使神经网络能特定的输入产生我们期望的输出。

对于一个前馈神经网络，我们使用下面的方法标记各参量。用 *L*：表示神经网络的层数；用 *m*(*l*) 表示第 *l* 层神经元的个数；用 *fl*() 表示 *l* 层的激活函数； *W*(*l*) ∈ R*m*(*l*)×*ml−*1 用来表示 *l* − 1 层到 *l* 层的权重矩阵；**b**(*l*) ∈ R*ml* 表示偏置矩阵； **z**(*l*) ∈ R*ml* 表示 *l* 层神经元的净输入；**a**(*l*) ∈ R*ml* 表示 *l* 层神经元的输出。

则我们可以用以下公式描述信息在前馈神经网络中的传播过程：

**z**(*l*) = *W*(*l*) · **a**(*l*−1) + **b**(*l*) (2-7)

**a**(*l*) = *fl* (**z**(*l*)) (2-8)

[2-7](#_bookmark20) 与 [2-8](#_bookmark21) 也可以合并为：

**z**(*l*) = *W*(*l*) · *fl*−1 (**z**(*l*−1)) + **b**(*l*) (2-9)

或者：

**a**(*l*) = *fl* (*W*(*l*) · **a**(*l*−1) + **b**(*l*)) (2-10)

由此可以得到数据在更层传播的路径 [2-11](#_bookmark22)，将向量 **x** 作为第 1 层的输入经过逐次传播后可得到将整个网络看成一个整体的复合函数 *φ*(**x**; *W,* **b**) ，其中 *L* 表示总层数。

**x** = **a**(0) → **z**(1) → **a**(1) → **z**(2) → · · · → **a**(*L*−1) → **z**(*L*) → **a**(*L*) = *ϕ*(**x**; *W,* **b**)) (2-11)

1989 年，Cybenko 等人的研究证明【引用】，上述前馈神经网络所表征的

*φ*(**x**; *W,* **b**) 函数在隐含神经元足够多的情况下，可以无限你和任意的连续非线性函数。这说明神经网络具有通用近似能力，这也是用神经网络来解决机器学习问题 的理论基础。

对于分类问题，还需要将 *φ* 函数作为某种分类器的输入，从而得到每种类别的概率。

ˆ*y* = *g*(*ϕ*(**x**)*, θ*) (2-12)

其中，ˆ*y* 为每类的概率输出，*θ* 为分类器的参数，*g* 为分类器。

对于多分类问题，常用的分类器为 Softmax 分类器。对于具有 *N* 个分类的多分类问题，式[2-13](#_bookmark23)给出了 softmax 函数的定义。softmax 函数可以将一组 *φ* 给出的输出转化为一组总和为 1 的概率，并且不影响转化前的大小相对关系。通过分类器的归一化之后的概率输出便可得出最可能的预测，因此分类器基本上位于网络 的最后一层。

*s*

*e i*

softmax (*s* ) =

∑*i N*

*j*=1

*esj* (*i* = 1*,* · · · *, N*) (2-13)

上面说明了数据在前馈神经网络中正向传播的过程。为了使神经网络得到可 靠的预测结果，需要对网络的参数，包括各层的权值和偏置，进行训练。训练一 般采用梯度下降算法，梯度下降时需要一个评估模型预测性能的损失函数。损失 函数一般由模型预测结构和真实结构的交叉熵计算得出。交叉熵由式[2-14](#_bookmark24)定义。

L(**y***,* **y**ˆ) = −**y**T log **y**ˆ

其中 **y** 与 **y**ˆ 分别表示实际标签与预测结果。 在交叉熵的基础上还可以定义风险函数：

R(*W,* **b**) = 1 ∑ L (**y**(*n*)*,* **y**ˆ(*n*)) + 1 *λ*∥*W*∥2

*N*

*N*

*n*=1

2

*F*

(2-14)

(2-15)

= 1 ∑ L (**y**(*n*)*,* **y**ˆ(*n*)) + 1 *λ*∥*W*∥2

*N*

*N*

*n*=1

2

*F*

(2-16)

其中 **W** 和 **b** 分别表示网络中的权重和偏置；其中 ∥*W*∥2 为了防止过拟合现象而引入的正则化项；*λ* 是为正数的参数。这里的正则化项一般由 Frobenius 范数（式[2-17](#_bookmark25)） 给出：

*F*

(2-17)

*L m*(*l*) (*l*−1) (

*W*

∥*W*∥2 = ∑ ∑ ∑

*l ij*

*F*

*l*=1 *i*=1

*j*=1

( ))2

对模型的训练过程即调整模型参数使得风险函数最小的过程。按照梯度下降

原理，每次迭代过程中第 *l* 层的参数更新公式为：

(*l*) (*l*) *∂*R(*W,* **b**)

*W* ← *W* − *α*

*∂W*(*l*)

= (*l*)

 1 ∑*N*

 *∂*L (**y**(*n*)*,* **y**ˆ(*n*))  +

1. 

*W* − *α*  *N*

*n*=1 

*∂W*(*l*)

 *λW* 

(2-18)

(*l*) (*l*) *∂*R(*W,* **b**)

**b** ← **b** − *α*

*∂***b**(*l*)

= (*l*)

 1 ∑*N*

*∂*L (**y**(*n*)*,* **y**ˆ(*n*)) 

### b

式中的 *α* 代表学习率。

− *α*  *N*

*n*=1

*∂***b**(*l*) 

使用梯度下降对神经网络参数进行优化时需要计算损失函数 L(**y***,* **y**ˆ) 关于每个参数的导数。现在假设需要计算损失函数相对于 **W**(*l*) 和 **b**(*l*) 的偏导数。由于 **W**(*l*) 和 **b**(*l*) 都是矩阵，如果要计算微分的话十分繁琐，所以可以先计算矩阵中某个元素的偏导数。根据求导法则，可得

*∂*L(**y***,* **y**ˆ)

*∂W*

*ij*

 *∂***z**(*l*) T *∂*L(**y***,* **y**ˆ)

(*l*)

*ij*

=  *∂W*(*l*) 

*∂***z**(*l*) (2-19)

*∂*L(**y***,* **y**ˆ)

( *∂***z**(*l*) )T *∂*L(**y***,* **y**ˆ)

*∂***b**(*l*)

现在来计算 *∂***z**(*l*) 。**z**(*l*) 和 *∂W*(*l*) 存在函数关系 **z**(*l*) = *W*(*l*)**a**(*l*−1) + **b**(*l*)，所以

*∂W*

*∂***b**(*l*) =

*∂***z**(*l*) (2-20)

(*l*)

*ij*

*∂***z**(*l*)

*∂W*

(*l*)

*ij*

*ij*

= *∂* (*W*(*l*)**a**(*l*−1) + **b**(*l*))

*∂* (*l*)*W*

*ij*

( )*−∂ W* **a** +**b**





(*l*) (*l* 1) (*l*)

1:

(*l*)

*ij*

*∂W*





.





 ( ) 

=



 =  *al*−1







*i*:



(*l*)

 0 

 . 

*j*

*∂ W*(*l*)**a**(*l−*1)+**b**(*l*)

(2-21)

*∂Wij*





.



( )*−∂* +







*W*(*l*) **a**(*l* 1) **b**(*l*)

 *l*



*m* :

*∂W*

(*l*)

*ij*





 . 

0

≜ I*i* (*a*(*l*−1))

*j*

其中 *W*(*l*) 代表 *W*(*l*) 的第 *i* 行。I*i* (*a*(*l*−1)) 代表第 *i* 行的值为 *a*(*l*−1) 的列向量。

*i*:

*j*

*j*

下面来计算偏导数 *∂***z**(*l*) 。带入函数关系 **z**(*l*) = *W*(*l*)**a**(*l*−1) + **b**(*l*) 可得

*∂* (*l*)**b**

*∂***z**(*l*)

*∂***b**(*l*)

= *∂*(*W*(*l*)**a**(*l*−1) + **b**(*l*)) =

*∂***b**(*l*)

**I***m*(*l*) (2-22)

其中 **I***m*(*l*) 为 *m*(*l*) 阶单位矩阵。

最后我们来计算 *∂*L(**y***,*ˆ**y**)。这个偏导项通常被称为误差项，用来表征某层神经元

(*l*)**z**

*∂*

对网络最后总误差的影响，我们把第 *l* 层的误差项记作 *δ*(*l*)。根据 **z**(*l*+1) = *W*(*l*+1)**a**(*l*) + **b**(*l*+1)，对 **a** 求导可得

(*l*+1)

( )T*∂***z** *W* (2-23)

= (*l*+1)

*∂***a**(*l*)

用按位计算函数 *fl*(·) 表示 **a**(*l*) 可得

( )

根据求导法则，可得

*∂***a**(*l*)

*∂***z**(*l*)

= *∂fl* **z**(*l*)

*∂***z**(*l*)

= diag (*f*′*l* (**z**(*l*)))

(2-24)

(*l*) *∂*L(**y***,* **y**ˆ)

*δ* ≜

*∂***z**(*l*)

= *∂***a**(*l*) *∂***z**(*l*+1)

*∂*L(**y***,* **y**ˆ)

⊙ 是向量点积运算。

*∂***z**(*l*) · *∂***a**(*l*+1) · *∂***z**(*l* + 1)

= diag (*f*′ (**z**(*l*))) · (*W*(*l*+1))T · *δ*(*l*+1)

*l*

= *f*′ (**z**(*l*)) ⊙ ((*W*(*l*+1))T *δ*(*l*+1))

*l*

(2-25)

从式[2-25](#_bookmark27)可以看出，后一层（*l* + 1 层）的误差项决定了前一层（*l* 层）的误差项，这种误差沿着网络向上一层传播的现象被称为反向传播。式[2-25](#_bookmark27)的含义是后一 层某个神经元的误差项是后一层与这个神经元连接相连的神经元的误差项的权重 和再乘上该神经元激活函数的梯度。

算得这三个偏导之后，公式[2-19](#_bookmark26)可写成

*∂*L(**y***,* **y**ˆ) = I (*a*(*l*−1))T *δ*(*l*) = *δ*(*l*)*a*(*l*−1)

*∂W*

(*l*)

*ij*

*i*

*j*

*i*

*j*

(2-26)

因此，损失函数关于第 *l* 层权重的梯度是：

*∂*L(**y***,* **y**ˆ) = *δ*(*l*) (**a**(*l*−1))T (2-27)

*∂W*(*l*)

损失函数关于第 *l* 层偏置的梯度为：

*∂*L(**y***,* **y**ˆ) = *δ*(*l*) (2-28)

*∂***b**(*l*)

因此，基于梯度下降和反向传播的神经网络训练算法可以归纳为：

* 1. 按从输入层到输出层的方向计算每一次的输入值 **z**(**l**) 和激活值 **a**(**l**)；
  2. 按从输出层到输入层的方向，即传播方向的反方向计算误差项 *δ*(*l*)；
  3. 按照式[2-27](#_bookmark28)和式[2-28](#_bookmark29)计算每层梯度，并根据梯度下降算法修改参数。 上述过程也可用伪代码来描述：

**Data:** 训练集D = {(**x**(*n*)*, y*(*n*))}*N* , ，验证集V，学习率 *α*，正则化系数 *λ*，层数 *L*，

*n*=1

第 *l* 层神经元数量 *m*(*l*)

随机初始化 *W,* **b while** 错误率在验证集 V 上持续下降 **do**

随机打乱训练集中的样本; **for** *n* = 1*...N* **do**

选取训练样本 (*x*(*n*)*, y*(*n*));

从前往后计算输入值 **z**(**l**) 和激活值 **a**(**l**)； 计算误差项 *δ*(*l*);

//计算每层导数

( )−= *δ* **a** ;

∀*l,*

∀*l,*

*∂*L(**y**(*n*)*,*ˆ**y**(*n*)) (*l*) (*l* 1) T

*∂W*(*l*)

*∂*L(**y**(*n*)*,*ˆ**y**(*n*)) (*l*)

= *δ* ;

*∂***b**(*l*)

//更新参数

*W*(*l*) ← *W*(*l*) − *α* (*δ*(*l*) (**a**(*l*−1))T + *λW*(*l*));

**b**(*l*) ← **b**(*l*) − *αδ*(*l*);

### end end

**Result:** 权重和偏置 *W,* **b**

算法 2-1 如何训练前馈神经网络

## 卷积神经网络

上节所述传统的前馈神经网络一般采用全连接结构，即每层所有神经元的输 出都与下层的每一个神经元相连，也就是每一个连接都尤其对应的权重参数。随 着网络层数的增加，网络模型中参数的数量也将急剧增加。这样将会导致两个问 题，一是参数过多时导致模型训练时计算量过大训练效率低，二是过多的参数很 容易导致过拟合现象即网络模型不能正确提取数据中的关键特征而过分强调无关

细节。

在图像处理领域，以上所述问题尤其严重，因为相比于一维信号，图片数据需要用二维矩阵来表示，数据量大。特别是在图片为彩色的情况下，对于红、绿、蓝三个通道，都要对应一个二维矩阵，这又使数据成倍增加。另外，对于图像处理来说，有很多图像上的特征在局部都是不变的，如材质的纹理，人脸的五官等， 如果使用全连接网络则很难提取出这些局部的特征。

卷积神经网络就是针对以上两个问题提出的一种改进型神经网络结构，可以很好地应用于图像和视频的各种图像分类、物体识别、图像分割等问题。如前面章节所述，探地雷达的回波信号是发射天线产生的电磁波脉冲在地层中的反射传到接收天线并通过接收机与采集板回传到 PC 端信号采集软件的一维实数信号，也就是所谓的 A 扫。随着收发天线的移动，不断采集到的二维信号可组合形成 B 扫数据。B 扫数据在形式上可看成是一种图像数据，另外，雷达 B 扫图像中目标反射所形成的类似于双曲线图像也有明显的局部化特征。虽然反射特征在整体 B 扫中的位置是不确定的，但是在每个反射特征的周围一块区域却存在着相似的特性。考虑到以上特点，探地雷达数据符合卷积神经网络的应用领域和数据特点。本文后面章节将使用卷积神经网络对探地雷达目标识别问题进行处理，并与多层感知机的效果作对比。

卷积神经网络是受生物学上感受野的机制而提出。感受野（receptive field）主要是指听觉、视觉等神经系统中一些神经元的特性，即神经元只接受其所支配的 刺激区域内的信号。在视觉神经系统中，视觉皮层中的神经细胞的输出依赖于视 网膜上的光感受器。视网膜上的光感受器受刺激兴奋时，将神经冲动信号传到视 觉皮层，但不是所有视觉皮层中的神经元都会接受这些信号。一个神经元的感受 野是指视网膜上的特定区域，只有这个区域内的刺激才能够激活该神经元。

应用到机器学习领域。卷积神经网络主要用到了三个基本思想：局部感受野、权值共享和子采样。下面分别对这三种思想作阐述。

【局部感受野】

前面所述全连接前馈神经网络中，输入数据是一维的列向量。在卷积神经网 络中输入是以多维的形式组织的.

和前面的网络类似，输入层数据也要连接到隐含层的神经元上，不过在卷积 神经网络中，我们不会将每个输入都连接到每一个隐含层神经元上。反之我们只 会连接一小部分局部的输入数据。更精确地说，第一个隐含层的每一个神经元都 会连接到输入神经元的一小块区域，比如说一个 5x5 尺寸的正方形区域（共 25 个输入数据）。这种连接方式可由图[2-7](#_bookmark31)来说明（图中没有画出所有的输入神经元到

隐含层神经元的连接）。









隐含神经元

感受野

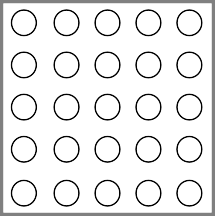








图 2-7 卷积层连接方式

上述正方形区域被称为隐含神经元的局部感受野。局部感受野可看成作用在 输入数据上一个窗。每一个从局部感受野到对应隐含神经元的连接都包含一个权 重，另外这个对应的隐含神经元本身也会有一个偏置参量。

将代表局部感受野的正方形窗在输入数据上平移，同时平移对应的隐含神经 元，我们即可将所有的局部感受野与隐含神经元连接起来。

【权值共享】

上面提到局部感受野到隐含神经元的每一个连接都包含一个权重，此权重可以由一个矩阵 **w** 来表示，这个矩阵的尺寸与局部感受野的尺寸相同。但是在这里有一个值得注意的地方，即对于卷积神经网络从输入层到隐含层的每一个权重矩阵都是共享的。权重共享的目的是解决局部特征的学习问题，因为权重共享之后， 在二维数据某一位置学习到的特征也可应用到二维数据的其他位置。应用权重共享之后，对于位置为 *j*,*k* 的隐含层神经元，其输出是：

*f* (*b* + ∑

*k*

∑ *wl,maj*+*l,k*+*m*)

(2-29)

*l*=0 *m*=0

*k*

其中，*f* 是激活函数，*b* 是共享的偏置，*wl,m* 是共享的权重矩阵，*aj*+*l,k*+*m* 是对应的感受野输入，*k* 是感受野尺寸。同时这里还要指出，式[2-29](#_bookmark32)中的求和部分可能看成是一个二维卷积的过程，这也就是卷积神经网络名称的由来。

从输入层到隐含层的映射被称为特征映射，即此映射包含了从输入数据中学 习到的一般特征。在卷积神经网络术语中，共享的权重矩阵 **w** 和偏置 *b* 又被称为

“核”或者“滤波器”。一般情况下，单个特征映射无法满足抽象出数据中足够多 二维特征的要求。所以在实际应用中，输入层可以映射到多个并列的隐含层，其 中每一个隐含层都对应一个独立的特征映射及滤波器。

【子采样】

上面包含卷积运算的隐含层叫做卷积层。除了卷积层，卷积神经网络中还包含汇聚层（pooling layer）。汇聚层的作用是对卷积层输出的数据做简化，也就是子采样，它通常紧接着卷积层使用。汇聚层将隐含层的输出结果矩阵划分成一批重 叠或者不重叠的区域，然后对每个区域应用汇聚函数得到一个单个的值，这样做 实质上就是把隐含层的输出结果尺寸减小，即子采样的过程。常用的汇聚函数有 最大汇聚和平均汇聚两种，分别可以表示为 [2-30](#_bookmark33) 与 [2-31](#_bookmark34)。

*d m,n*

*Y*

= max *xi* (2-30)

*d m,n*

*i*∈*R*

*d m,n*

*Y*

= 1

1*Rd* 1

∑ *xi* (2-31)

*m,n*

1 *m,n*1 *i*∈*Rd*

使用汇聚层之后，可以在大大减少神经元的数量的同时使神经网络对于一些微小 细节的改变保持不敏感性，避免过拟合的发生。

如图[2-8](#_bookmark35)所示，卷积层、汇聚层和全连接层一起构成一个典型的卷积神经网络。一个卷积块为连续 M 个卷积层和 b 个汇聚层（M 通常设置为 2 ∼ 5，b 为 0 或 1）。一个卷积网络中可以堆叠 *N* 个连续的卷积块，然后在接着 K 个全连接层（*N* 的取值区间比较大，比如 1 ∼ 100 或者更大；*K* 一般为 0 2）。

输入 softmax

全连接层

×M

×b

汇聚层

激活函数

卷积

×K

×N

图 2-8 卷积神经网络结构示意图

按照图[2-8](#_bookmark35)给出的连接方式，结合实际问题的特点，不断增加网络的层数，达

到足够提取出数据中的抽象特征进行高准确度预测的效果，这样的网络被称为深 度卷积网络。由于深度卷积网络可以利用数据的二维结构，而且所需要训练的参 数相对来说更少，因此它是目前深度学习领域中应用较为广泛的网络结构。

## 本章小结

二十世纪五十年代提出的感知器是用来模拟生物体神经元活动的模型。感知器接收二进制输入并输出一个二进制结果。感知器是组成人工神经网络的基础模块，通过多个感知器的级联可用来表征复杂的决策过程。在原始感知器模型的基础上引入的激活函数是感知器中参数的自动化训练成为可能，因为它使感知器的输出由离散的量变为连续的量，从而让感知器内部参数的微小变化只会导致输出的微小改动。多个带有激活函数的感知器的互联可用来处理需要高阶抽象能力的问题，这种互联的网络被称作人工神经网络。人工神经网络一般多层感知器级联而成。所有网络的输入都直接来自于上一层输出的神经网络被称为前馈神经网络， 若网络中存在循环结构则被称为循环神经网络。神经网络的误差函数可在交叉熵的基础上定义，误差函数用来在采用梯度下降算法优化网络参数时评估模型预测性能。反向传播算法使计算误差函数对于各层权值和偏置的梯度称为可能，这是梯度下降法求解最优化问题时的关键步骤。卷积网络是近年来火热的解决图像等二维数据识别问题的有效网络结构，其结构主要特点是局部感受野、子采样和权值共享。卷积神经网络特别适合解决雷达 B 扫图像的相关识别问题。

# 第三章 地下探测 FDTD 仿真技术研究

本章的主要是内容是探地雷达应用到地下目标探测时的仿真技术。在将深度学习相关技术，特别是深度卷积神经网络技术应用到探地雷达地下目标识别时需要提供大量的训练与测试数据用于神经网络的性能调优、结构调整、参数选择等等过程。由于电磁仿真所得的数据具有目标位置明确模型参数已知的特点，在进行神经网络建模时就有了明确的参考依据，可以验证网络的预测正确性和性能。 另外，数值仿真也有数据获取简单，能用批量得到不同目标参数的 B 扫数据的特点。因此，在将深度学习技术应用到实际实验数据之前，先在大量仿真数据上进行初始研究是有重要意义的，这将为后面的工作提供良好的准备。

目前在探地雷达数值仿真领域，由于其原理简单、容易并行加速的特点，最常用的且最成熟的方法为时域有限差分（FDTD）法（引用）。常见的探地雷达数值仿真计算为了模型简便，通常采用简单几何模型、均匀介质、点源等近似方法， 在普通应用时通常也能带来令人足够满意的效果。但是，由于本文进行探地雷达数值仿真的目的是为深度学习目标识别技术的研究提供训练数据，需要仿真所得的数据尽量接近实际数据的特点，这样才能对网络的实际性能做可信的评判，如果仿真模型过于理想化则生成的 B 扫数据会具有明显的目标特征而在非目标位置呈现干净的图像从而与实际实验中复杂的土壤和仪器环境所带来的诸多干扰和虚假目标不相吻合。具体来说，笔者在应用这些常规方法建模并进行数值仿真时发现以下几个问题：

* + 1. 简单几何模型不能模拟土壤表面复杂的起伏特征。如图[3-1](#_bookmark38)所示，真实土壤表面呈现凹凸不平的起伏特征，这将会在雷达回波中引入相当程度的抖动等干扰， 而常规仿真方法中使用的标准长方体模型的边界是完美的平面，不能代表实际土壤的形态。所以需要研究土壤表面起伏的数据特征并构想相关的模型生成算法。
    2. 均匀介质模型不能模拟土壤中介质的随机分布特征。观察图[3-1](#_bookmark38)中的土壤剖面可看到不均匀的土壤颗粒分布特征，土壤中介质的不均匀分布必将导致雷达波传播过程中的路径和散射特征的改变，从来对雷达 B 扫图像引起干扰。常规方法中所使用的均匀介质，除了在不同介质的相交面这样的介质参数突变的位置，对雷达波不会产生任何扭曲和路径改变，产生的数据过于理想化。因此需要研究土壤介质的构成模型和分布特点并提供相应的建模算法以模拟真实土壤的介质形态。
    3. 点源模型不能模拟近场状态时天线的传输特征。常规仿真方法通常将探地 雷达天线近似为单个网格的点源模型，但是由于本文所研究的目标识别问题中目



图 3-1 某地土壤切面图，来源 [https://www.nrcs.usda.gov](http://www.nrcs.usda.gov/)

标和天线的距离较近，且该距离接近天线的工作频率波长，因此此时天线的传输 特征对雷达 B 扫图像将产生可见的影响，不符合尽量获取拟真数据的目标，因此继续使用常规方法中的点源模型将不是理想的选择。

本章主要内容安排为：1. 介绍 FDTD 仿真的基本原理和开源数值仿真平台

Gprmax；2. 针对模拟土壤表面复杂起伏特征和介质随机分布特征的问题，基于土壤介质参数半经验模型和 FFT 分形数据生成算法形成真实土壤模型构建算法；3. 针对模拟天线传输特征的问题，研究将真实天线几何模型内置到 FDTD 网格的方法。4. 综合前面几节的结果，批量得到拟真雷达 B 扫图像。

## 仿真方法与计算平台

常用的电磁仿真方法有矩量法、有限元法和 FDTD 法。其中有限元法的理论基础是变分和插值，适用于求解各类微分方程问题，特别是物理场的求解问题； 矩量法的求解目标是代数方程组，这些方程组由待求解积分或微分方程转化而来；

FDTD 法以差分原理为基础，将麦克斯韦方程中的微分算子通过差分近似。FDTD 法具有容易掌握，直观简洁的特点，其计算迭代方程可由麦克斯韦方程直接导出 而且不需要进行其他复杂的推导过程，另外由于其将空间划分为均匀的立方体网 格，所以也特别适合复杂模型的程序化生成，特别是本章要解决的土壤与天线建 模问题。

## FDTD 方法基本原理

(𝑖 + 1, 𝑗, 𝑘)

𝐻𝑥

(𝑖 + 1, 𝑗 + 1, 𝑘 + 1)

𝐸𝑧



𝐸𝑥

𝐻𝑧

𝐸𝑦

𝐻𝑦

x

z



y

(𝑖, 𝑗, 𝑘) (𝑖, 𝑗 + 1, 𝑘)

图 3-2 FDTD 网格示意图

FDTD 法的基本思想是将计算空间离散为一个一个的矩形网格（图[3-2](#_bookmark41)），然后再将磁场与电场矢量分布到这些网格上，其中磁场位于每个立方体各个面的重心处且垂直于该表面，电常位于各条边中心且与其平行。将 **E***x*、**E***y*、**E***z*、**H***x*、**H***y*、

**H***z* 六个分量以 Maxwell 方程组联系起来，即可得到：

*ϵ∂Ex* = *∂Hz* − *∂Hy* − *σE*

*∂t*

*∂y*

*∂z*

(3-1)

*x*

*ϵ∂Ey* = *∂Hx* − *∂Hz* − *σE*

*∂t*

*∂z*

*∂x*

(3-2)

*y*

*ϵ∂Ez* = *∂Hy* − *∂Hx* − *σE*

*∂t*

*∂x*

*∂y*

(3-3)

*z*

*μ∂Hx* = *∂Ey* − *∂Ez*

*∂t*

*∂z*

*∂y*

(3-4)

*μ∂Hy* = *∂Ez* − *∂Ex*

*∂t*

*∂x*

*∂z*

(3-5)

*μ∂Hz* = *∂Ex* − *∂Ey*

*∂t*

*∂y*

*∂x*

(3-6)

以上 6 个分量均可由对于三个空间方向以及时间方向的偏导数，也就是对于

*x, y, z, t* 的偏导数都可以以下中心差分公式近似表示：

*∂f*(*ξ*) = *∂f*(*ξ* + Δ*ξ/*2) − *∂f*(*ξ* − Δ*ξ/*2)

*∂ξ*

*∂ξ*

*∂ξ*

(3-7)

代入式[3-1](#_bookmark42)-[3-6](#_bookmark43)即可得到 Maxwell 方程的离散形式：

*En*+1 (*i* + 1 *, j, k*) = ( 2*ϵ*−*σ*Δ*t* ) *En* (*i* + 1 *, j, k*) + ( 2Δ*t* ) {

*x*

2

2*ϵ*+*σ*Δ*t*

*x*

2

2*ϵ*+*σ*Δ*t*

2

2 *,*

2 *,*

2

2 *,*

2 *,*

1 [*Hn*+ 1 (*i* + 1

*j* + 1 *k*) − *Hn*+ 1 (*i* + 1

*j* − 1 *k*)] −

(3-8)

1 [*Hn*+ 1 (*i* + 1

Δ*y*

*z*

*z*

Δ*z*

*y*

2

2 *,*

*,*

2

*y*

2

2 *,*

*j k* + 1 ) − *Hn*+ 1 (*i* + 1

*j k* − 1 )]

*En*+1 (*i, j* + 1 *, k*) = ( 2*ϵ*−*σ*Δ*t* ) *En* (*i, j* + 1 *, k*) + ( 2Δ*t* ) {

*,*

2

*y*

2

2*ϵ*+*σ*Δ*t*

*y*

2

2*ϵ*+*σ*Δ*t*

2

*,*

2 *,*

2

*x*

2

2 *,*

*,*

2

1 [*Hn*+ 1 (*i*

*j* + 1

*k* + 1 ) − *Hn*+ 1 (*i* + 1

*j k* − 1 )] −

(3-9)

1 [*Hn*+ 1 (*i* + 1

Δ*z*

*x*

Δ*x*

*z*

2

2 *,*

*j* + 1

*k*) − *Hn*+ 1 (*i* − 1

*j* + 1

*k*)]

*En*+1 (*i, j, k* + 1 ) = ( 2*ϵ*−*σ*Δ*t* ) *En* (*i, j* + 1 *, k*) + ( 2Δ*t* ) {

2 *,*

*z*

2

2 *,*

2 *,*

*z*

2

2*ϵ*+*σ*Δ*t*

*z*

2

2*ϵ*+*σ*Δ*t*

2

2 *,*

*,*

2

*y*

2

2 *,*

*,*

2

1 [*Hn*+ 1 (*i* + 1

*j k* + 1 ) − *Hn*+ 1 (*i* − 1

*j k* + 1 )] −

(3-10)

1 [*Hn*+ 1 (*i*

Δ*x*

*y*

Δ*y*

*x*

2

*,*

2 *,*

2

*x*

2

*,*

*j* + 1

*k* + 1 ) − *Hn*+ 1 (*i*

*j* − 1

*k* + 1 )]

*Hn*+ 1 (*i*

*x*

2

*,*

*j* + 1

*k* + 1 ) = ( 2*μ*−*σ*Δ*t* ) *Hn*− 1 (*i*

*j* + 1

*k* + 1 ) −

( 2Δ*t* ) { 1 [*En* (*i, j* + 1*, k* + 1 ) − *En* (*i, j, k* + 1 )] −

2 *,*

2

2 *,*

2

2*μ*+*σ*Δ*t*

*x*

2

*,*

2 *,*

2

2*μ*+*σ*Δ*t*

Δ*y*

*z*

2

*z*

2

(3-11)

1 [*En* (*i, j* + 1 *, k* + 1) − *En* (*i, j* + 1 *, k*)]

Δ*z*

*y*

2

*y*

2

*Hn*+ 1 (*i* + 1

*y*

2

2 *,*

*j k* + 1 ) = ( 2*μ*−*σ*Δ*t* ) *Hn*− 1 (*i* + 1

*j k* + 1 ) −

( 2Δ*t* ) { 1 [*En* (*i* + 1 *, j, k* + 1) − *En* (*i* + 1 *, j, k*)] −

*,*

2

2*μ*+*σ*Δ*t*

*y*

2

2 *,*

*,*

2

2*μ*+*σ*Δ*t*

Δ*z*

*x*

2

*z*

2

(3-12)

1 [*En* (*i* + 1*, j, k* + 1 ) − *En* (*i, j, k* + 1 )]

Δ*x*

*z*

2

*z*

2

*Hn*+ 1 (*i* + 1

*z*

2

2 *,*

*j* + 1

*k*) = ( 2*μ*−*σ*Δ*t* ) *Hn*− 1 (*i* + 1

*j* + 1

*k*) −

( 2Δ*t* ) { 1 [*En* (*i* + 1*, j* + 1 *, k*) − *En* (*i, j* + 1 *, k*)] −

2 *,*

2*μ*+*σ*Δ*t*

*z*

2

2 *,*

2 *,*

2*μ*+*σ*Δ*t*

Δ*x*

*y*

2

*z*

2

(3-13)

1 [*En* (*i* + 1 *, j* + 1*, k*) − *En* (*i* + 1 *, j, k*)]

Δ*y*

*x*

2

*x*

2

通过以上各式，在 *t*0 时刻已知当前空间所有网格电场值的情况下，便可推得*t*0 + Δ*t/*2 时刻所有网格的磁场值，再次步进Δ*t/*2 时刻即又能得到 *t*0 + Δ*t* 时刻的电场值。如此循环往复即可进行电磁波在自由空间与介质中随时间传播的数值模拟 结果。以上是 FDTD 方法的时间步进基本过程，关于其数值稳定性、误差分析和边界条件等细节在此便不再赘述。

## Gprmax 仿真平台

通过上节的原理很容易自行编写可用的 FDTD 数值计算程序。笔者曾编写过基于 C++ 与 Matlab 的三维 FDTD 仿真软件，两种程序均存在一些难以克服的缺点。Matlab 版由于 Matlab 强大的工程计算函数库和友好的界面很容易进行快速建模与结果可视化，但是其计算缓慢，即使使用内置的并行接口其速度也难以达到 本文的计算量需求。C 语言版可生成高效原生代码，并且可借助并行计算程序包

OpenMP 在实验室数值计算服务器上利用多核 CPU 的优势得到更快的计算速度。但是相对于 Matlab 版，C 语言版却缺乏足够的灵活性，在复杂形体建模方面需要书写大量的底层逻辑甚至需要用到专业的计算几何知识，此外 C 语言在数据的可视化方面也缺乏足够的工具，往往还是需要借助其他工具进行数据处理。考虑到这些因素，本文采用英国爱丁堡大学团队开发的开源电磁仿真程序包 Gprmax 作为 FDTD 计算引擎，并利用其开源的优势进行大量的二次开发以满足本文中建模仿真的需要。

结合本文所研究问题的实际特点，Gprmax 与 C、Matlab 以及其他常见商业软件相比以下显著优势：

计算速度快。最新版 Gprmax 采用 python 语言开发，python 作为一门脚本语言本身运行效率并不高，但是却可因为其胶水语言的特性克服这一缺点。其主 要加速途径主要有三方面：1. 可以对 Gprmax 源码进行研究可发现内部其使用了Cython 程序包。Cython 程序包可将普通的 Python 语句编译到C 语言，从而解决解释性语言运行速度慢的问题。2. FDTD 时间步进方程在计算时每个网格场值只与上一时刻的场值有关，每个网格的计算是互相独立，因此可将计算空间划分为多 个区域，并将每个区域分配给单独的 CPU 核心计算。另外，在同时仿真多个模型时，还可以利用 MPI 调度集群内的多个CPU 同时计算并汇总计算结果。3. 最新版本的 Gprmax 开始支持基于英伟达CUDA 平台的 GPU 加速，实测利用显卡的多核心矩阵运算能力可将计算时间缩短到之前的四十分之一左右。

极强的可扩展性。Gprmax 的扩展能力使得用户可以很方便的定义属于自己的模型库和脚本化控制仿真的运行，特别适合需要建立复杂模型与大量仿真的情形。

在阅读其源码后也可以对内部计算流程进行修改和补充，比如改变仿真结果的保 存格式。Gprmax 的扩展能力一方面来自于其开源的特点，即所有代码可供用户阅读与修改，另一方面由于 python 语言在运行时不需要编译，用户所作的修改和自定义库可以和原有 Gprmax 程序无缝融合而不需要传统软件工程的构建流程，方便了使用者的自定义流程。

本章后面小节的建模与仿真充分利用了这两个优势。首先土壤天线建模和补 充均以 python 用户模块的形式增强 Gprmax 原有的功能。其次仿真时均利用实验室高性能显卡实现了显著加速。

## 拟真土壤模型

本节所介绍的拟真土壤模型由两方面的理论组成，其一是土壤混合物的电磁 参数模型，其二是地下介质分布和地形所蕴含的分形特征。

## 拟真土壤模型原理

Peplinski 等人提出一种模拟真实土壤电磁特性的半经验模型。在此模型中，土壤由三种成分构成，分别是：颗粒直径在 0.05mm 到 2.0mm 之间的沙土，颗粒直径在 0.002mm 到 0.05mm 之间的粉土，以及颗粒直径在 0.002 以下的粘土。这三种成分的配比不同可表示不同的土壤类型，进而也影响到其电磁特性。同时，土 壤中含水量也对其电磁特性起重大影响。综合这些因素，土壤的复介电常数 *εm* 由式[3-14](#_bookmark47)给出。

*εm* = *ε*′*m* − *jε*′*m*′

[ *ρ*

*ε*′*m* = 1*.*15

1 +

*b* (*εα*) + *mβ′ εα*

]1*/α*

− *mv*

− 0*.*68

(3-14)

*ρs s*

*ε*′′ = [*mβ′′ ε*′′*α*]1*/α*

*m*

*v*

*fw*

*v fw*

上式中，*ε*′*m* 与 *ε*′*m* 分别是复介电常数 *εm* 的实部与虚部；*ρb* 与 *ρs* 分别是土壤的总密度和其中沙土成分的密度（单位：*g/cm*3）;*β*′ 与 *β*′′ 分别是与土壤构成相关的常数，其表达式为[3-15](#_bookmark48)，其中 *S* 与 *C* 分别是沙土与粘土的构成比例（0 *< S <* 1*,* 0 *< C <* 1）。

*β*′ = 1*.*2748 − 0*.*519*S* − 0*.*152*C β*′′ = 1*.*33797 − 0*.*603*S* − 0*.*166*C*

(3-15)

式[3-14](#_bookmark47)中 *ε*′*fw* 与 *ε*′*fw*′

分别是土壤中自由水的实部与虚部，其表达式由式[3-16](#_bookmark49)给

出。

*ε*′*fw* = *εw*∞

+ *εw*0 − *εw*∞

1 + (2*πfτw*)2

(3-16)

*ε*′′

= 2 *πfτw* (*εw*0 − *εw*∞) + *σ*eff

(*ρs* − *ρb*)

*fw* 1 + (2*πfτw*)2

2*πε*0*f*

*ρsmv*

其中，*εw*∞ 是 *ε*′*fw* 在高频时的极限；*τw* 为自由水的弛豫时间常数；*εw*0 为水的静态相对介电常数，其值为 80.1；*σ*eff 为有效电导率，由经验公式[3-17](#_bookmark50)给出。

*σ*eff = 0*.*0467 + 0*.*2204*ρb* − 0*.*4111*S* + 0*.*6614*C* (3-17)

以上便是拟真土壤模型的基本原理，下面讨论如何将此模型应用到探地雷达

FDTD 仿真中。按照式[3-14](#_bookmark47)，在给定土壤构成比例 *S* 与 *C* 后，该土壤类型的电磁参量主要由含水量决定。实际土壤中含水量的分布不是均匀的，因此可定义含水量 的最小值与最大值分别为 *mvmin* 与 *mvmax*。将此含水量范围等距离取 *n* 个点并代入式[3-14](#_bookmark47)便可得到一系列复介电常数 *εi*(*i* = 1*,* 2*,* 3*...n*)

得到组成土壤的介质列表之后，接下来了的问题便是考虑这些介质应该如何分布在 FDTD 三维网格中。文献（[https://www.cambridge.org/core/books/fractals-](http://www.cambridge.org/core/books/fractals-) and-chaos-in-geology-and-geophysics/FA8339855DBBF054CCF18D8E5DDFFAC9）指出地下空间内孔隙度含水量等特性的分布并不是完全随机的而是呈随机分形特征。因此可以利用分形生成算法安排前面得到的一系列介质在 FDTD 网格中的分布从而达到对真实土壤较好的模拟效果。

分形在数学中是一种抽象的物体，用于描述自然界中存在的事物。人工分形 通常在放大后能展现出相似的形状。分形也被称为扩展对称或展开对称。如果在 每次放大后，形状的重复是完全相同的，这被称为自相似。分形在不同的缩放级 别上可以是近似相似的。分形也包有图像的细节重复自身的意味。一种效率较高 的分形数据生成算法是以离散傅里叶变换为基础的。其算法理论基础是具有分形 特征的数据的离散傅里叶变换（DFT）上各频率的系数 *mf* 与频率 *f* 近似呈幂律关系，即：

*C*

=

*mf −*(2*D−*7)

(3-18)

*f* 2

其中 *D* 为分形维度，其值在 1 到 3 之间。

因此，可以用以下步骤生成生成复介电常数列表在土壤所在三维空间内 FDTD

网格的分形分布：

1. 对于尺寸为 *x, y, z* 的 FDTD 网格区域，生成相同大小的随机三维数据 **A**。
2. 对 **A** 应用三维快速傅里叶变换算法，并将结果进行平移，使零频率处于三维数组的中心。
3. 设 **A** 的中心点的坐标为 (*ic, jc, kc*)，对于每一个数据点 **A**(*i, j, k*) ，计算其与

√

中心点的模，即 *L* = (*i* − *ic*)2 + (*j* − *jc*)2 + (*k* − *jc*)2。

1. 应用公式 **A**(*i, j, k*) = **A**(*i, j, k*) 1 。

*−*(2*D−*7)

*L* 2

1. 对 **A** 进行逆傅里叶变换，并将结果归一化并量化为 1 到 *n* 的整数（对应土

壤介质的个数），并在对应位置为 FDTD 网格的赋予相应的介质参数。

要获得高度拟真的土壤模型，除了要考虑介质参数分布方面的特征，还要考 虑到地形的特征。和土壤介质的分布一样，土壤表面起伏的形态也符合分形的特征（图[3-1](#_bookmark38)）。

由于同样满足分形特征，土壤起伏表面也可由上述类似的方式生成，比较重 要的变化是上面三维的数据变为二维的数据，空间上的三维傅里叶变换变成平面 上的二维傅里叶变换，但是其算法本质不变。其过程可描述为（定义土壤起伏程 度为 *k*，分形维数为 *D*：

1. 对于尺寸为 *x, y* 的 FDTD 网格区域（该区域为需要生成土壤起伏表面的土壤区域顶部所围成的二维区域），生成相同大小的随机二维数据 **A**。
2. 对 **A** 应用二维快速傅里叶变换算法，并将结果进行平移，使零频率处于二维数组的中心。
3. 设 **A** 的中心点的坐标为 (*ic, jc*)，对于每一个数据点 **A**(*i, j*) ，计算其与中心

√

点的模，即 *L* = (*i* − *ic*)2 + (*j* − *jc*)2。

1. 应用公式

**A**(*i, j*) = **A**(*i, j*) 1

(2*D* 7)

*− −*

(3-19)

*L* 2

1. 对 **A** 进行逆傅里叶变换，将所得结果归一化到[−*k, k*] 范围内，此时 **A** 即代表土壤区域顶部对应的起伏程度，将此起伏程度数据应用于建模过程便可得到具 有起伏表面的土壤模型。

## 建模实例

为了方便展示计算过程，这里对土壤起伏表面的生成过程做建模演示。图[3-3](#_bookmark53)展 示了上述土壤分形生成算法中各步骤的计算结果，其中第一步中 *x* 和 *y* 的值皆为 400，分形维数 *D* = 1*.*5。图[3-3](#_bookmark53)(a) 为原始随机矩阵，在经过傅里叶变换后变为图[3-3](#_bookmark53)(b)，频域数据经过公式[3-19](#_bookmark51)处理后能量集中到低频区域，这一点可在图[3-3](#_bookmark53)(c) 看出来，其中能量由图中心的低频部分像周围的高频区域逐渐下降（图中的颜色

以对数尺度标出）。最后，如图[3-3](#_bookmark53)d 所示，恢复到时域的数据呈现类似于地形起伏的分形特征。

0

50

100

150

200

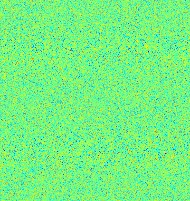
y

250

300

350

400



0 50 100 150 200 250 300 350 400

X

0

50

100

150

200

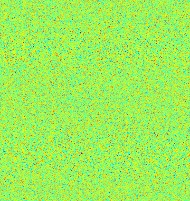
y

250

300

350

400



0 50 100 150 200 250 300 350 400

X

(a) (b)

0

50

100

150

200

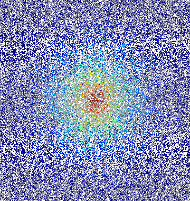
y

250

300

350

400



0 50 100 150 200 250 300 350 400

X

0

50

100

150

200

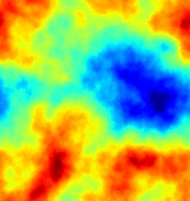
y

250

300

350

400



0 50 100 150 200 250 300 350 400

X

(c) (d)

图 3-3 土壤起伏表面建模实例二维图。(a) 随机二维数据；(b) 随机数据的二维傅里叶变换；(c) 应用公式[3-19](#_bookmark51)后的结果；(d) 最后生成的土壤起伏数据

直观起见，图[3-4](#_bookmark54)给出了[3-3](#_bookmark53)(d) 中数据的三维曲面图。设该曲面所代表的数据为 **A***fractal*，且其最大值为 *Amax* 最小值为 *Amin*。在将该结果应用到 FDTD 建模之前需要对其值的范围进行调整，若指定模型中该土壤表面高度值的范围为 *hmin* 到 *hmax*， 则可根据下面的公式得出在 FDTD 模型中的实际表面高度。

−

**A***fractal*

= **A** *hmax* − *hmin* + *h*

*fractal Amax* − *Amin*

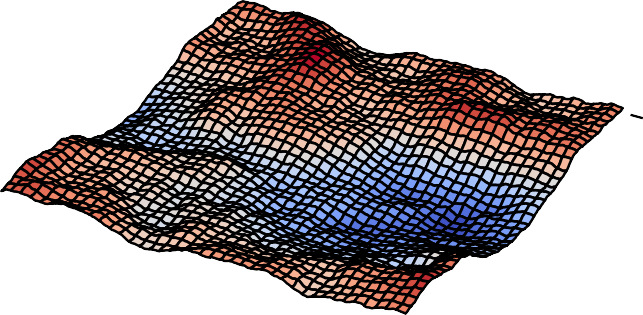
*min*

( *hmax* − *hmin* )*h Amax* − *Amin*

*min*

(3-20)

0.10



0.05

z

0.00

0.05

0.10

0.0

0.2

0.4

X

0.6

0.8

1.0

0.0

0.2

0.4

0.6

y

0.8

1.0

图 3-4 土壤起伏表面建模实例三维图

按照前述类似的过程也可以建立前述土壤介质分布模型，将此模型与起伏表面模型结合起来便得到完整的拟真土壤模型（图[3-5](#_bookmark57)）。

## 天线模型仿真

探地雷达主要采用偶极子天线、蝶形天线、维瓦尔第天线等几种常用的天线 类型。本节讨论 FDTD 网格中天线建模的方法，并给出建模实例。

## FDTD 天线建模方法

将天线模型置入 FDTD 过程中首先要考虑的问题是如何将实际的天线几何模型转化成离散的网格模型。对于结构较为简单的天线来说，如偶极子天线和蝶形 天线，天线的形状可由基本的几何形体构成，而这些基本的几何形体（三角形、矩 形、长方体、球、圆柱等）在 Gprmax 中都有对应的建模指令，因此处理起来较为方便，只需要选择合适的网格尺寸将天线的结构用基本几何体描述出来即可。而

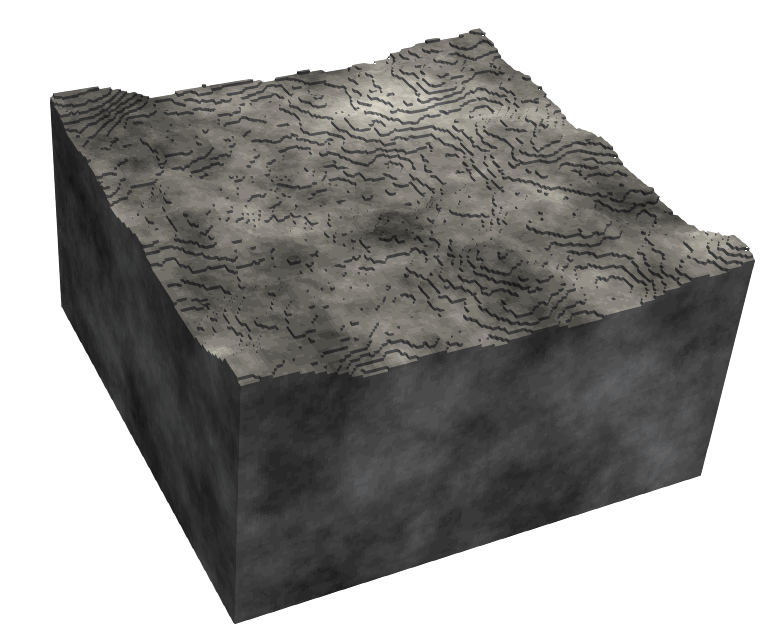


图 3-5 拟真土壤模型建模示例

对于包含复杂弧度的天线来说，直接用几何体无法拼接出想要的形状。在实践中 笔者为了解决这一问题提出了一套较为实用的解决方案。即先采用图像处理的方 法先对天线设计稿进行预处理，标记出天线上金属板的范围。然后，根据 FDTD 网格尺寸的需要，将处理好的天线金属板图片进行离散化。最后再次通过图像处 理程序将图片中已标记的像素点的位置信息提取出来，然后在 FDTD 网格模型中使用边长正好等于网格尺寸的矩形板结构依次对应图片中的像素点位置信息。这 样便能将复杂的天线形状完美地移植到 FDTD 网格中。这一系列步骤将在后面的例子中演示。

其次，还需要考虑仿真过程中信号源加载的问题。在探地雷达实际工作过程 中激励信号是由超宽带脉冲信号源产生并通过同轴电缆加载到收发天线的对应端 口上。而在 FDTD 仿真中，无法做到对信号源以及同轴电缆进行仿真。笔者通过数值实验发现，Gprmax 中带有内阻的点源可以很好的模拟同轴线端口并将激励信号通过天线金属板辐射出去。这里还有两方面的问题需要注意。第一点是点源的 极化方向需要与实际端口引出的连接到金属板的两个触电的方向一致，这样才能 保证信号的传输方式和实际一样。第二点，因为点源只占一个网格，因此为了让 点源的电场同时加载到正负极板上，需要保证正负极板在加载位置只间隔一个网 格，这样才不会出现点源与极板之间间隔有自由空间的情况。

## 仿真实例

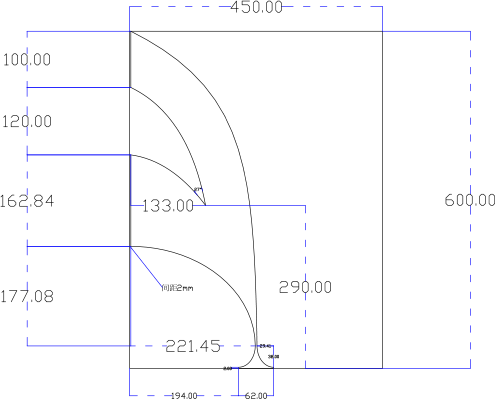
本节以笔者所在团队所设计的维瓦尔第天线为例阐释复杂形状天线的建模过 程。图[3-6](#_bookmark59)分别给出了此天线的实物图与设计稿。此天线尺寸为 600mm x 450mm x

1mm，两片金属板分别位于基板的正反两面，基板由相对介电常数为 3 的绝缘材料构成，馈电位置位于实物图设计稿底部中间位置。

图[3-7](#_bookmark60)为天线设计稿经过图像处理并离散化后的结果，其中离散化的精度为

10mm，这是兼顾了仿真精度与仿真速度的选择。此图片为像素值只有 0 或 1 两种可能的二值化图形，因此很容易将其中的白色区域的坐标数据提取出来并将其导 入 Gprmax 的 FDTD 网格中。

图[3-8](#_bookmark61)是该天线金属部分在 FDTD 网格中的可视化图像。基板正反两面的金属板分别以不同颜色标出，两者所在的平面相隔一个 FDTD 网格以便加载信号源。由于所用金属的电阻率很小，天线金属部分的介质设置为 PEC（完美电导体）。基板部分按照实际所用材料的参数将其相对介电常数设置为 3。点源位于馈电位置在 FDTD 网格的对应位置，设置其内阻为 50Ω。



* + - 1. (b)

图 3-6 维瓦尔第天线。(a) 实物图；(b) 设计稿

为了检验在 FDTD 网格中建模的天线的辐射特性在多大程度上符合实际天线的参数，在电源出加载一中心频率为 500MHz 的高斯脉冲，运行仿真并记录电源处的场值，由此可计算出此天线的 *S*11 参数。图[3-9](#_bookmark62)给出了天线 *S*11 参数的仿真与实测结果对比，其中实测结果由矢量网络分析仪在微波暗室中测得。可以看出从

0.6GHz 以下，仿真曲线与实测曲线基本吻合，而这也包括了团队探地雷达的实际工作频率范围。0.6GHz 以上与实测曲线出现的较大偏离的原因是为了提高计算效率 FDTD 网格的尺寸设置为 10mm，从而使波长短的高频部分出现了较大的误差。



图 3-7 维瓦尔第天线几何模型离散化结果（网格尺寸 10mm）

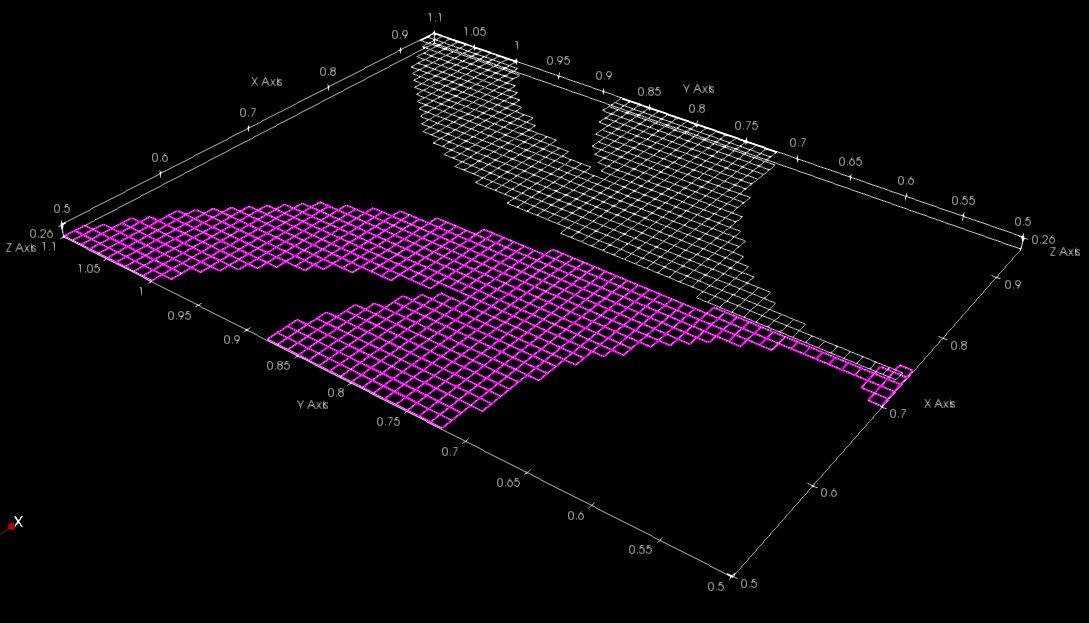
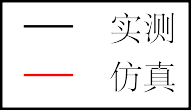


图 3-8 维瓦尔第天线 FDTD 网格模型

10



5

0

5

10

S11(db)

15

20

25

30

35

0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0

(GHz)

图 3-9 维瓦尔第天线 *S*11 曲线比较

图[3-10](#_bookmark65)给出了信号源能量在向外传播过程中某时刻的电场值分布图。可以看 出天线有明显的方向性，其辐射能量集中在天线的正前方，这也与天线的设计符 合。

## B 扫数据的批量生成

## 本章小结

本章首先研究了时域积分方程时间步进算法的阻抗元素精确计算技术，分别 采用 DUFFY 变换法与卷积积分精度计算法计算时域阻抗元素，通过算例验证了计算方法的高精度。

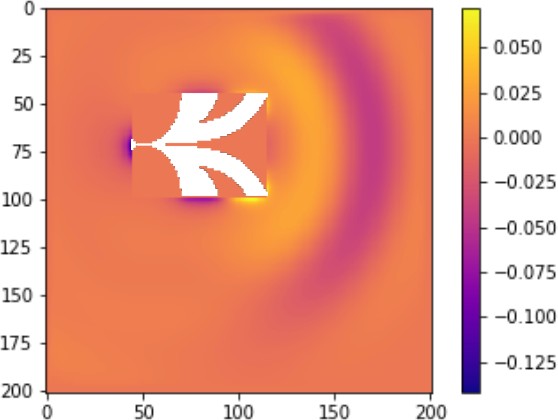


图 3-10 维瓦尔第天线辐射电场值图

# 第四章 时域积分方程数值方法研究

## 时域积分方程时间步进算法的阻抗元素精确计算

时域积分方程时间步进算法的阻抗元素直接影响算法的后时稳定性，因此阻 抗元素的计算是算法的关键之一，采用精度高效的方法计算时域阻抗元素是时域 积分方程时间步进算法研究的重点之一。

## 时域积分方程时间步进算法阻抗矩阵的存储

时域阻抗元素的存储技术也是时间步进算法并行化的关键技术之一，采用合 适的阻抗元素存储方式可以很大的提高并行时间步进算法的计算效率。

## 时域积分方程时间步进算法产生的阻抗矩阵的特征

由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组 合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程 时间步进算法的阻抗矩阵特征相同。

## 数值算例与分析

如表[4-1](#_bookmark71)所示给出了时间步长分别取 0.4ns、0.5ns、0.6ns 时的三种存储方式的存储量大小。

表 4-1 计算 2*m* × 2*m* 理想导体平板时域感应电流采用的三种存储方式的存储量比较。

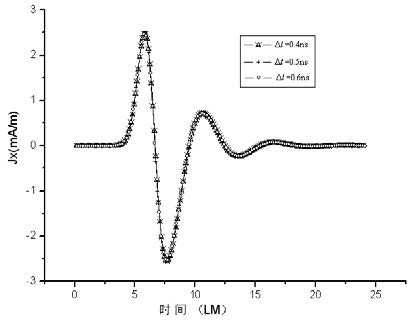
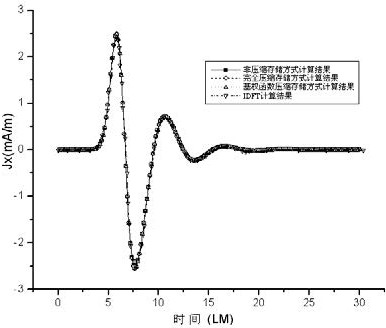
|  |  |  |  |
| --- | --- | --- | --- |
| 时间步长 | 非压缩存储方式 | 完全压缩存储方式 | 基权函数压缩存储方式 |
| 0.4ns | 5.59 MB | 6.78 MB | 6.78 MB |
| 0.5ns | 10.17 MB | 5.58 MB | 5.58 MB |
| 0.6ns | 8.38MB | 4.98 MB | 4.98 MB |

如图[4-1(a)](#_bookmark72)所示给出了时间步长选取为 0.5ns 时采用三种不同存储方式计算的平板中心处 *x* 方向的感应电流值与 IDFT 方法计算结果的比较，……。如图[4-1(b)](#_bookmark73) 所示给出了存储方式为基权函数压缩存储方式，时间步长分别取 0.4ns、0.5ns、

* 1. ns 时平板中心处 *x* 方向的感应电流计算结果，从图中可以看出不同时间步长的计算结果基本相同。

由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组 合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程

时间步进算法的阻抗矩阵特征相同。



* + 1. (b)

图 4-1 2*m* × 2*m* 的理想导体平板中心处感应电流 *x* 分量随时间的变化关系

由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组 合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程 时间步进算法的阻抗矩阵特征相同。

## 时域积分方程时间步进算法矩阵方程的求解

定理 4.1 如果时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组合。

证明: 由于时域混合场积分方程是时域电场积分方程与时域磁场积分方程的线性组合，因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积 分方程时间步进算法的阻抗矩阵特征相同。 ■ 推论 4.2 时域积分方程方法的研究近几年发展迅速，在本文研究工作的基础

上，仍有以下方向值得进一步研究。

引理 4.3 因此时域混合场积分方程时间步进算法的阻抗矩阵特征与时域电场积分方程时间步进算法的阻抗矩阵特征相同。

## 本章小结

本章首先研究了时域积分方程时间步进算法的阻抗元素精确计算技术，分别 采用 DUFFY 变换法与卷积积分精度计算法计算时域阻抗元素，通过算例验证了计算方法的高精度。

# 第五章 全文总结与展望

## 全文总结

本文以时域积分方程方法为研究背景，主要对求解时域积分方程的时间步进 算法以及两层平面波快速算法进行了研究。

## 后续工作展望

时域积分方程方法的研究近几年发展迅速，在本文研究工作的基础上，仍有 以下方向值得进一步研究：

# 第六章 template

This is the template of the chapter in split file.

## 6.1 s1

This is a section.

## 6.1.1 sub1

This is a subsection.

# 致 谢

在攻读博士学位期间，首先衷心感谢我的导师 XXX 教授