硕士学位论文

基于 CNN 芯片的抗辐照干扰研究及 SNN 芯片损伤分析

作者姓名：钱欢

指导教师：梁旭文　研究员　谢卓辰　副研究员

中国科学院微小卫星创新研究院

学位类别：工学硕士

学科专业：通信与信息系统

培养单位：中国科学院微小卫星创新研究院

2020 年 6 月

Research on Anti-radiation Interference Based on CNN Chip and SNN Chip Damage Analysis

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Science in Engineering

in Communication and Information System

By

Qian Huan

Supervisor: Professor Liang Xu Wen

Associate Professor Xie Zhuo Chen

Innovation Academy for Microsatellites of Chinese Academy of Sciences

June 2020

中国科学院大学
研究生学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的指导下独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明或致谢。

作者签名：

日　　期：

中国科学院大学
学位论文授权使用声明

本人完全了解并同意遵守中国科学院有关保存和使用学位论文的规定，即中国科学院有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名：　　导师签名：

日　　期：　　日　　期：

摘 要

空间辐照干扰尤其是单粒子翻转（SEU）效应对神经网络芯片的正常稳定运行造成了很大影响，它会导致存储在芯片 SRAM 存储器的权重参数随机发生比特位翻转，进而神经元的权重参数值就会发生变化，这会直接影响神经网络芯片输出的准确度。近年来，随着人工智能技术的发展，原有的微控制单元（MCU）早已无法满足深度学习对海量数据运算和高速运算的要求，人工智能（AI）芯片便应运而生。对于应用在复杂空间的芯片例如卫星上的应用，对芯片的稳定运行则提出了更高的要求。本文在研究和分析针对传统芯片的加固技术上，例如硬件加固发面有三模冗余电路的方法，软件加固方面主要有擦洗方法和纠错编码。虽然这些方法能在一定程度上提升芯片的一个抗干扰能力，但对于神经网络芯片这种强调硬件开销少、恢复时间短与处理速度快等性能的芯片，传统的加固技术并不能起到一个很好的作用并且传统的加固技术也没有充分利用神经网络芯片中神经网络各个神经元相互之间连接的特性。基于以上芯片硬件开销、恢复时间与处理速度的问题，提出采用 dropout 算法构建新的网络框架，以一定概率屏蔽受到 SEU 影响的神经元，并进行了相关实验的仿真验证。本论文主要工作如下：

1、本文对芯片的空间辐照干扰环境做了详细的研究与分析，分析了单粒子翻转效应产生的机理，总结了现有的一些芯片抗辐照干扰加固技术：工艺加固、屏蔽加固和设计加固的原理，并就常用的设计加固方法详细列举几种常见的加固方法如三模冗余、擦洗方法和纠错编码的原理以及不足之处。

2、在手写字体数据集上，训练卷积神经网络(Convolutional Neural Networks,CNN)和脉冲神经网络(Spiking Neuron Networks, SNN)，从准确度上验证不受干扰的神经网络对数据集的识别的准确度。对于训练好的神经网络，进行一个网络参数和模型结构的提取，并对 SNN 网络进行了损伤分析。

3、软件仿真模拟空间辐照干扰效应，先后随机选取 1‰，1%，5%，10%的比例参数进行随机位的注错干扰，这样便可得到一部分出错的参数，并使该错误参数替换原来的参数，并传回网络相应的层，赋值给对应的神经元所表示的权重，结合 dropout 算法和 CNN 网络，测试神经网络芯片在受到辐照干扰的准确度，验证算法的可行性。

关键词：SEU 干扰；SRAM 存储器；神经网络芯片；权重参数

Abstract

Spatial radiation interference, especially the single even upset (SEU) effect, has a great impact on the normal and stable operation of the neural network chip. It will cause the bit parameters of the weight parameters stored in the chip SRAM memory to randomly flip, and then the weight parameters of the neuron The value will change, which will directly affect the accuracy of the neural network chip output. In recent years, with the development of artificial intelligence technology, the original micro control unit (MCU) has long been unable to meet the requirements of deep learning for massive data operations and high-speed operations, and artificial intelligence (AI) chips have emerged. For applications that are applied to chips in complex spaces, such as satellites, higher requirements are placed on the stable operation of the chips. This paper focuses on the research and analysis of the reinforcement technology for traditional chips, such as the method of hardware-reinforced three-mode redundant circuit, and the software reinforcement mainly includes the scrubbing method and error correction coding. Although these methods can improve the anti-jamming capability of the chip to a certain extent, for the neural network chip, a chip that emphasizes performance such as low hardware overhead, short recovery time, and fast processing speed, traditional reinforcement technology cannot play a very important role. Good function and traditional reinforcement technology also failed to make full use of the characteristics of the connection of each neuron of the neural network in the neural network chip. Based on the above problems of chip hardware overhead, recovery time and processing speed, it is proposed to use dropout algorithm to construct a new network framework, shield neurons affected by SEU with a certain probability, and carry out simulation verification of relevant experiments. The main work of this paper is as follows:

1. This article makes a detailed study and analysis of the chip's space irradiation interference environment, analyzes the mechanism of the single event upset effect, and summarizes some existing chip anti-irradiation interference reinforcement technologies: process reinforcement, shield reinforcement and design The principle of reinforcement, and enumerate several common reinforcement methods such as three-mode redundancy, scrubbing methods and error correction coding principles and deficiencies in terms of commonly used design reinforcement methods.

2. On the handwritten font data set, train convolutional neural network CNN and

spiking neural network SNN to verify the accuracy of the undisturbed neural network's recognition of the data set from the accuracy. For the trained neural network, a network parameter and model structure are extracted, and the damage analysis of the SNN network is performed.

3. The software simulates the effect of space irradiation interference, and randomly selects 1 ‰, 1%, 5%, and 10% of the proportional parameters for random bit error injection interference, so that a part of the wrong parameters can be obtained and the wrong parameters can be obtained. Replace the original parameters, and return to the corresponding layer of the network, assign the weight represented by the corresponding neuron, combined with the dropout algorithm and CNN network, test the accuracy of the neural network chip in the interference of irradiation, and verify the feasibility of the algorithm.

Key Words: SEU Interference; SRAM Memory; Neural Network Chip; Weight Parameter

目 录

图 目 录

表 目 录

缩 略 词

BP	Back Propagation	前馈神经网络
CNN	Convolutional Neural Network	卷积神经网络
FC	Fully connected layers	全连接层
FIT	Failures in Time	失效率
GAN	Generative Adversarial Network	生成对抗网络
MBU	Multi bit Upset	多比特错误
MCU	Multi Units Upset	多单元错误
SBU	Single Bit Upset	单比特错误
SEU	Single Event Upset	单粒子翻转
SNN	Spiking Neural Network	脉冲神经网络
SOC	System on Chip	系统级芯片
SRAM	Static Random Access Memory	静态随机存取存储器
STDP	Spike Timing Dependent Plasticity	放电时间依赖可塑性
TMR	Triple Modular Redundancy	三模冗余
VAE	Variational Auto-Encoder	变分自动编码器

第1章 绪论

1.1　论文背景与研究意义

1.1.1　研究背景叙述

随着人工智能技术近些年的快速发展，其在深度学习、自然语言处理与计算机视觉等领域都取得了巨大成就。同时，人工智能也开始从智能化工具逐渐向智能机器进军，这使得原有的微控制单元（MCU）早已无法满足深度学习对海量数据运算和高速运算的要求, 人工智能（AI）芯片便应运而生，搭载神经网络的人工智能芯片这些年也是推陈出新，网络越来越复杂，功能也随之越来越强大。如下表 1.1 所示，其中国外以 IBM 仿人脑的 TrueNorth 芯片[1]和英伟达的 GPU 芯片系列最为著名，国内则有寒武纪公司的 DianNao 芯片[2]和地平线机器人公司的 BPU 芯片等。

表 1.1　不同公司的神经网络芯片比较

Table 1.1　Comparison of neural network chips of different companies

公司	芯片名称	芯片功能
寒武纪	Dian Nao	可以支持较大规模的神经网络架构，加速神经网络的计算
IBM	TrueNorth	最接近人脑的神经网络架构，模拟了人脑的大量神经元以及各种突触结构，再将功耗做低的同时，运算效率却有极大的提高
地平线机器人	BPU	将神经网络算法与芯片进行充分耦合，提升了 AI 计算效率，在自动驾驶方案中使用较多
华为	麒麟系列芯片	基于神经元的硬件处理单元，嵌入式的处理结构，可以用来处理海量的复杂数据类型
英伟达	GPU	主要用于动态图像处理的 3D 加速，尤其在图形以及高性能的并行数据处理上发挥出色
英特尔	Nervana	可以加速各种复杂的神经网络结构并且功耗较低

人工智能芯片技术作为未来人工智能方向的重要方向之一，在卫星应用领域可以大幅提升卫星大数据的快速提取、智能处理与分析等效率, 人工智能芯片技

术在未来将会是卫星大数据领域发展的重要基础之一。而在现实世界里，编写神经网络的代码并不是在真空中运行，其需要一定的介质并且可能在复杂的空间环境中运行。神经网络程序将与其他程序一样也共享着计算资源，其中一些程序有一定攻击性。以 rowhammer 攻击[3-4]为例，在 rowhammer 攻击的期间，一行内存会以极高的速率重复写入数据，在擦除与写入的不断重复过程中由此产生的电磁噪声会导致相邻静态随机存取存储器（SRAM）[5]行中的部分比特位发生随机翻转，例如内部存储的数据如果是 00001111，则可能由于翻转会变成 00001010。如果遭受影响的 SRAM 行存储了神经网络的权值或各层网络输入输出，那么神经网络的整个性能可能会受到很大的负面影响，导致结果准确率不高从而很大概率上输出一个错误的结果。如下图 1.1 所示，一个正常训练好的神经网络对图像的识别大概率是正确的，而由于单粒子翻转效应引起错误的神经网络对图像识别的结果很大概率上是无法预料甚至是错误的的。

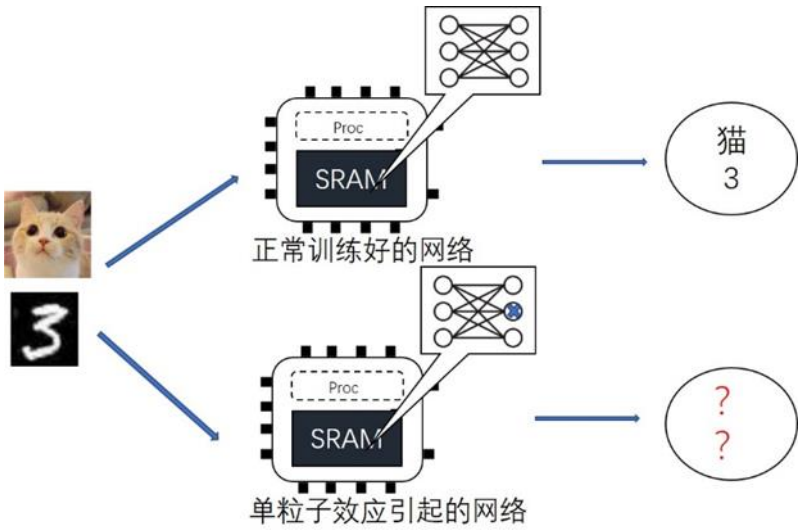


图 1.1　单粒子效应对神经网络判断的影响

Figure 1.1　The influence of single particle effect on the judgment of neural network

芯片器件易出错的另一个计算领域的例子就是高辐射环境，当代比较近的一个例子是日本福岛核电站周围的地区，其工厂附近遭受大量核辐射，辐射对我们人类健康来说是一个极大的危害，同时它对机器人的正常工作来说也是一个极大的威胁，由于空间高能辐射粒子撞击硅芯片并翻转存储和执行单元中的一些比特位，导致芯片正常工作的一些参数出错[6]，进而一些机器人的任务不得已以失败

而告终。高辐射环境还有一个常见的例子是太空，其中最难以克服干扰便是单粒子翻转(SEU)，所以卫星的一些关键性器件需要进行硬件或者软件方面的加固，才能在环绕地球或访问其他行星的轨道上正常工作数年。如美国 UOSAT-2 卫星曾经在太空运行的四年间，由于空间辐照引起的单粒子翻转事件[7] 发生了近万次，我国“风云一号（B）”气象卫星主控芯片也因 SEU 事件而导致姿态控制系统紊乱。

IBM 公司最新研究的 TrueNorth 芯片模拟了人脑神经元的连接方式，由大量的脉冲神经元和突触构成一个复杂的 SNN 网络结构，其内部神经元的激活方式也是像人脑那样与膜值电位的大小有关系，运算的效率在极高提高的同时又将整个芯片的功耗做的很低，这就是模拟类脑结构带来的好处。但是对于芯片受到干扰后，由于其内部大量脉冲神经元前后依靠膜值电位传递数据信号，这就使得脉冲神经元的正常稳定工作变得尤为重要。

1.1.2　研究意义叙述

近二十年来，随着人工智能技术在各个领域的高速发展，各种基于神经网络的芯片也随之层出不穷，在不同领域都发挥着极大的作用。神经网络芯片由于其本身的高度集成化、微小化和复杂化，其对空间环境的要求也愈发变高。像处在高速交换数据的场景和充满辐照干扰的空间中，芯片的正常稳定运行无时无刻都会受到周围环境的影响。对于芯片的传统加固技术由于其在硬件开销、恢复时间与处理速度性能等问题上已经很难满足神经网络芯片运行的需要，如何利用神经网络芯片中神经网络区别于传统芯片的特点，结合现有的芯片发展技术，设计出一种神经网络芯片抗空间辐照干扰研究的算法框架模型就显得尤为重要了。针对神经网络芯片中烧录进去的神经网络模型结构，考虑神经网络之间各个神经元相互之间的连接关系，再分析神经网络各层级之间神经元相连的权重参数以及数据传递路径，建立权重参数与神经元之间的一个算法模型。本文在基于这样的研究背景与要解决的问题之下，利用神经网络中神经元的特性，先是软件模拟芯片在受到空间辐照 SEU 影响下的一个工作状态，而后以一定概率屏蔽那些受到干扰的神经元，基于神经网络芯片，提出新的 dropout 算法框架来提高神经网络芯片的一个抗空间辐照干扰的能力，让结果输出的准确度进一步提升，增强其在复杂环境下稳定运行的一个容错性。SNN 网络由于其独特的模拟人脑神经元的工作

方式，其内部从脉冲神经元、突触以及膜值电位与类脑结构都相似，对其损伤分析的研究为后续芯片中的网络像人脑那样的类脑工作模式学习有着重要意义。

1.2 国内外研究进展

针对神经网络芯片这个大领域而言，增强其抗辐照干扰能力对其加固主要有工艺加固、屏蔽加固和设计加固这几个措施，也即是从硬件和软件两个方面进行针对性地处理。工艺加固一般是指在生产芯片的工艺线上对芯片进行参杂一些特殊物质，用来提高芯片内部电路对空间辐照干扰的容错能力；屏蔽加固就是对生产好的芯片外围电路进行封装一些包敷材料来抵御空间辐照干扰；设计加固相比较前面两种从芯片底层解决的方法要灵活很多，会对不同用途的芯片进行不同的设计，比如三模电路的冗余设计对那些电路硬件开销不大的芯片比较适合，芯片的擦洗方法就对那些不追求处理数据速度的芯片很友好，而软件方面的纠错编码设计对于注重芯片稳定恢复时间短就特别适合。近年来，随着人工智能的大力发展，各种基于神经网络的算法层出不穷，并且很多优秀的算法已经集成到各个神经网络芯片中，在各自的领域发挥着重要的作用。对于一个完整的神经网络芯片，由于前期神经网络已经在相应地大数据中训练得很好，在放置芯片中时，应对新数据也可以能准确判断输出想要的结果，但在复杂空间的辐照干扰下，芯片的输出准确度往往会不尽如意。现在有些抗干扰的算法诸如生成对抗网络 GAN 算法[8-10]，变分自编码器 VAE 算法[11-13]等，它们的运用提高了模型的一个抗干扰能力，不足之处就是主要针对输入数据样本受到干扰的情况下才有一定的容错能力，而对于神经网络本身内部模型一些存储在 SRAM 区的权重参数如果受到干扰出错，则没有一个很好的纠错能力。

针对神经网络芯片抗空间辐照干扰的问题，从上面对国内外近些年的调研与分析可以看出主要的容错方法分别是硬件上的电路加大冗余设计和软件层面的算法设计，下面主要围绕神经网络芯片先是介绍几种传统常见的加固方法，将原理进行了叙述并就优缺点进行了对比。而利用神经网络特点的抗干扰的相关算法，从算法模型的基础理论知识、设计的过程、取得的效果以及算法模型的优缺点进行了说明。

1.2.1　传统的芯片加固方法分析

在传统的芯片加固技术中，可用于对神经网络芯片进行抗辐照干扰加固的主要有：三模冗余(Triple Modular Redundancy, TMR)[14-16]的硬件电路设计、芯片的擦洗校正电路方法以及软件设计的纠错编码的方法。这些加固方法能分别在芯片的稳态恢复时间、硬件资源开销和芯片处理速度方面能取得一定的效果，但随着神经网络芯片的高度集成化，算法的高度复杂化以及处理数据的庞大化，尤其像那些处理语音识别、图像视觉等领域，这些传统方法虽然能在一定程度上起到一个对芯片的加固的作用，但却没有充分利用神经网络芯片中神经元之间的特性进行加固。而且神经网络芯片对电路开销的要求比较小，处理速度要快。

1.2.1.1　电路的冗余方法设计

受制于芯片制造过程中工艺材料的发展瓶颈和工艺制造成本上升的约束，冗余容错方法中最为常见的一般就是芯片内部电路的硬件冗余、后期芯片的软件冗余和信息冗余等设计加固，其中比较典型且应用较为广泛的是硬件的三模冗余容错设计方法[17]，如下图 1.2 所示是一个典型的硬件三模冗余内部电路原理图，其基本原理是在电路中设计三倍冗余的模块，对输入的数据执行相同的操作，而后将各个模块计算后的输出数据再输入到一个多数表决器中，采取三取二的选举判决策略得出最终结果。由于各个模块之间互相独立运行，故两个模块同时因为辐照干扰出现错误的概率极小，从而可以大大提高系统输出的准确度。

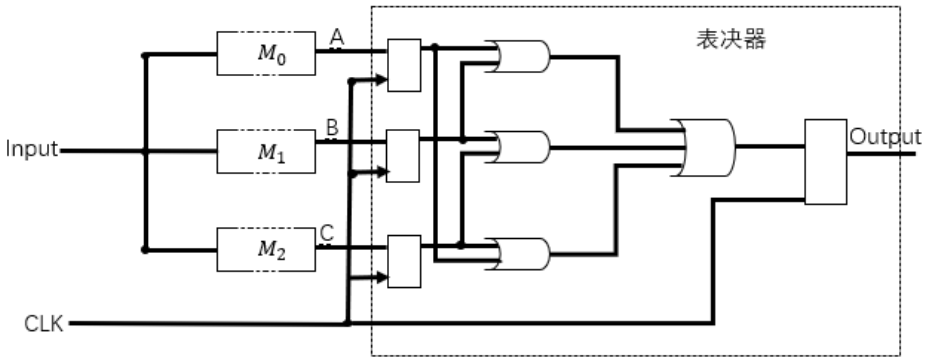


图 1.2　硬件的三模冗余内部电路原理图

Figure 1.2　Hardware schematic of the triple mode redundant internal circuit

1.2.1.2 周期性的擦洗方法设计

芯片在控制单元或者本身烧录程序控制下，通过对存储单元地周期性擦洗可以刷新存储单元中的内容[18]，这样一来能减少存储单元受到辐照干扰暴露的时间，周期性地擦洗可以直接地提高芯片存储单元抗空间辐照干扰的能力。该方法无需额外的硬件电路系统设计，实现起来比较简单。

1.2.1.3 纠错编码的方法设计

为了在一定程度上避免存储单元受到空间辐照干扰的影响，可以采用纠错编码的方法[19]，其基本原理是把输入数据经过编码后的信息码元序列额外添加一些监督码元后放在存储单元中进行存储，在进行处理输出数据读取时，通过比较监督码元与自身的信息码元之间的关联关系是否与初始编码一致时来判断数据在存储与处理读取过程中是否由于辐照干扰发生错误[20]，比较常见的纠错码有海明码、R-S 码、奇偶校验码等。

1.2.1.4 几种传统加固方法对比

硬件的三模冗余电路需要对电路进行三倍同样的硬件设计，这会对芯片宝贵的硬件资源造成过度的开销，同时芯片的体积、重量和能耗都会因此增加，而且在数据的传送处理过程中，进行了过多的数据预处理，这会降低芯片本来的一个处理速度，在极限环境下可能会降低 70%到 80%的一个处理速度。特别是三模电路硬件本身设计比较复杂，如果三模电路自身出现不可预料的故障，会导致整个系统对输入数据的一个错误判决。周期性的擦洗方法设计缺点是擦洗方法本身不能对芯片存储单元内容就是否受到空间辐照干扰进行自我校验，且恢复只是一个暂态的周期，只能起到一定程度的减缓。而纠错编码方法的设计其缺点是发生错误时，需要更新芯片存储器内的单元数据，这会占用芯片的运行开销，有一定的延时并且降低了芯片处理数据的处理速度。

如果希望将上述方法应用到卷积神经网络芯片上，特别是应用在数据量庞大的网络参数以及输入输出等存储数据时，在实际运用时都有一定的困难，具体概括如下：

1） 三模冗余电路需要对电路进行三倍同样的硬件电路设计，考虑到现在广泛应用的卷积神经网络所包含参数以及各类输入输出数据量大，硬件资源开销将非常巨大；

2） 周期性的擦洗方法对数据恢复只是一个暂态的周期，无法适应卷积神经网络的大量高速并行计算；

3） 纠错编码方法有一定的延时，降低了芯片处理数据的工作速度，对于网络层数多、运算量大的卷积神经网络来说，将大大影响网络计算时延等性能指标。

1.2.2　神经网络算法相关的抗干扰方法分析

神经网络芯片在实际运用中其加固过程是一个不可或缺的步骤，此过程就是保护神经网络芯片在应对各种复杂干扰环境下能稳定的运行，对空间辐照干扰造成的影响能起到一定的容错作用。传统的一些针对芯片的加固技术虽然能在一定程度上让芯片保持一个容错的能力，但是随着芯片功能的强大，尤其是搭载复杂神经网络的芯片，其对芯片的稳定运行、处理速度以及响应要求都提出了更加严格的标准。而传统的芯片加固技术只是针对普遍芯片而言，其并没有利用神经网络芯片中神经元以及各层之间的连接关系，没有从神经网络的各种特点入手去进行一个软件算法层面的加固设计，训练好的一个神经网络模型由于其类似黑盒，本身在面对一些干扰时自身就会有一定的容错性，能抵抗一定的干扰。但对于一些复杂的空间辐照干扰，需要进行进一步对神经网络模型的设计和优化。

利用神经网络模型的传递性、冗余性和复杂性，在现在很多抗干扰的应用场合中，各种不同风格的神经网络模型算法被提出来，它们也能很好的起到一定的抗干扰作用。在 Ian Goodfellow[8]等提出生成对抗网络(Generative Adversarial Network ,GAN)研究随机数据的生成问题，其通过对判别器和生成器之间的不断博弈训练，让判别器不断判别生成器的数据准确度，判别器也在这个判别过程中不断训练自己，最后达到一个临近饱和的状态，让生成器生成的数据可以欺骗判别器而通过判别，其对输入数据的要求并不高，即使是受到很大干扰的输入数据也能在经过生成器训练好的的神经网络中输出自己想要的数据，其有一定的准确度。生成对抗网络它本身是一种生成网络模型，它相比较一些其他的神经网络模型其只用到了数据的反向传播，而没有那些复杂的链式传播，链式传播的缺点是对于那些受到干扰的数据再进行层层传播时往往会起到一个放大误差的作用，而生成对抗网络就能很好的避免这个问题；生成对抗网络它也是一种无监督的自我学习训练方式，能产生更加清晰逼真的样本，图片生成领域就是一个典型的应用，

它能对一些受到干扰或者残缺的数据甚至是随机数据也能很好地生成所需要的数据样本。生成对抗网络的缺点也很明显，它很难在自身博弈训练过程中找到生成器和判别器互相欺骗的那个平衡点，有时候可以简单的运用梯度下降法能找到，有时候始终无法达到那个平衡点，这就造成了生成对抗网络的一个难训练的问题，并且在训练过程中生成器总是往发散的方向迁移，而判别器则是往极度收敛的方向发展，这也就是很难在某个平衡点附近对这两个网络形成一个动态平衡，所以在设计生成对抗网络模型时需要考虑数据样本分布特点去精心设计安排博弈训练过程；而且由于生成器和判别器在不断博弈训练，样本的数据集又是有一定的分布特点，这就会造成生成对抗网络在学习过程中自身模式会部分缺失，其生成器开始慢慢退化，导致生成器神经网络总是生成单一的、同样的相似样本数据从而无法继续往后训练学习。

Kingma[13]等人研究出了一种变分自动编码器(Variational Auto-Encoder ,VAE)能使得神经网络模型输出的数据样本分布尽可能地靠近原始输入数据样本的分布，即使输入数据部分受到一定干扰发生错误，只要在不影响原始输入数据一个特征分布上就可以。变分自动编码器网络顾名思义其主要由两大部分组成，第一个主体部分是网络中处理输入数据的编码器(Encoder)，第二个主体部分是网络中处理输出数据的解码器(Decoder)，其中编码器和解码器针对不同数据可以有不同的神经网络模型。输入的数据样本经过神经网络的处理可以降维到一个符合输入数据特征的编码(code)上，而后这个经过编码的数据又会经过另外一个神经网络去进行解码(decode)升维得到能与原来输入数据几乎一致的生成数据，然后网络模型会内部进行对这两个数据的比较，不断地最小化它们之间的误差来训练学习整个自编分网络中编码器和解码器的一系列权重参数。当整个神经网络模型学习训练完成后，就可以利用这个训练好的解码器，传入一个受到干扰的输入数据经过编码的数据，其就可以通过这个解码器生成和原始数据样本类似一致的数据，这个过程就是一种对输入数据的抗干扰作用，有一定的容错能力，不必太在意原始数据是否受到干扰也能得到神经网络模型想要的输出数据。但是由于整个网络中由于没有对抗网络的存在，如果输入数据受到干扰较大以致于能反应输入数据部分特征缺失，往往输出的数据不准确例如输出的图片相比较原始图片可能会显得比较模糊。

通过上文对于神经网络芯片在受到干扰的加固方法分析，传统的加固技术用

于神经网络芯片抗干扰方面主要有硬件上的三模冗余电路设计、芯片自身的周期性擦洗方法设计和纠错编码的软件设计，利用神经网络而设计的抗干扰方法有生成对抗网络算法和变分自动编码器神经网络算法模型等。传统的加固技术几乎是针对各种芯片都适用的，范围比较广，是比较常见的通用方法，而利用神经网络算法的抗干扰技术主要针对神经网络芯片，因为其他类型的芯片并没有搭载神经网络，这就不能利用神经网络自身的网络特性去进行一些抗干扰的设计。随着芯片功能的加强，有些芯片现在已经不局限于在自身使用一种单一的抗干扰方法，而是将多种适合的抗干扰方法进行整合来让芯片达到一个协同作用的目的，比如神经网络芯片的外围逻辑电路部分可以用一些传统方法进行加固，而对于内部神经网络部分的数据流则可以采用一些神经网络算法模型去进行抗干扰容错。

1.3　论文研究内容和结构安排

论文主要围绕空间辐照干扰尤其是单粒子翻转效应会对神经网络芯片的正常稳定运行造成很大影响的问题，分析几种神经网络模型算法对错误干扰前后的输出准确度影响并对 SNN 网络结合其脉冲神经元的特性对其进行了损伤分析，看神经网络在不同注错模式下的一个鲁棒性如何。但是，在单粒子效应中，有诸如单比特错误(Single Bit Upset, SBU) 、多比特错误(Multi bit Upset, MBU) 和多单元错误(Multi Units Upset, MCU) 等多种单粒子翻转模式，需要考虑其对神经网络内部权重参数的一个影响。因为辐照干扰有时影响的不仅是输入数据，还有对内部神经网络的一个权重参数产生影响，往往一个训练好的神经网络在芯片中运作时，由于其存储在 SRAM 区的内部权重参数都是以二进制形式存储的，如果部分 SRAM 区域受到干扰会导致储存在其上的部分二进制发生翻转，也即本来是比特位 1 发生翻转变为 0 或者原来是比特位 0 的发生翻转变为 1。而之前提到的几种神经网络算法只是对输入数据受到干扰有一定的抗干扰和容错能力，对于网络内部参数出错并没有一个很好的抗干扰效果。

论文在用软件仿真研究了在不同比例权重参数出错的情况下神经网络的测试准确度的同时，提出了一种屏蔽受到干扰影响的神经元使其处于失活状态，采用 dropout 算法构建新的网络框架方法，用以提升神经网络相比较出错状态下的一个准确度。该方法主要是按照一定的概率去丢弃一些由于单粒子翻转效应扰动而导致某些权重误差改变的病态神经元，从而提高整个系统输出的准确度。

通过对以上关于神经网络芯片加固方法的研究与分析，论文主要可以从下面这几个部分来做一些对神经网络芯片抗干扰加固的工作：

(1) 分析 dropout 算法的一个基础原理，并结合神经网络芯片在权重参数受到干扰的情况下如何进行屏蔽处理失活的病态神经元；

(2) 基于单粒子翻转效应的一个机理，利用软件仿真去注错神经网络芯片，让其网络的部分参数处于一个随机出错的情况，模拟辐照干扰的效果；

(3) 通过对比神经网络芯片在不同比例参数出错情况下对数据输出的一个准确度，利用 dropout 算法构建的新的网络结构，并比较软件模拟注错错情况下芯片网络输出的准确度，利用神经网络神经元之间连接的特殊性，提高网络算法的抗辐照干扰的能力。

本次论文的章节内容结构安排如下：

第 1 章，主要探讨了论文的研究背景和意义，引出了空间辐照干扰下芯片在运行过程中会出现的哪些问题，围绕这些问题，介绍了近些年国内外针对芯片的各种加固方法，尤其是针对本文所研究的神经网络芯片区别于传统的芯片，进而分析了两种主要的神经网络抗干扰的方法，从算法的原理到其运用做了简单介绍并给出了算法本身的一个优缺点说明。综合传统加固技术和现有的神经网络抗干扰算法，提出本文所要研究的 dropout 算法框架模型。

第 2 章，首先介绍了卷积神经网络的模型结构，从神经元到激活函数再到各层级结构进行了逐一介绍，然后叙述了脉冲神经网络的结构，其中一个重要概念就是脉冲神经元是如何影响脉冲神经网络的工作模式的。

第 3 章，先是介绍了单粒子翻转效应是如何影响神经网络芯片一个正常工作流程的，叙述了此次实验运用的神经网络模型，然后对神经网络 CNN 和 SNN 进行训练学习，得到一个可以稳定输出的神经网络，并将其训练好的参数进行本地化保存，随后对这些参数进行模拟单粒子翻转效应进行一定比例的注错。分析在注错前神经网络对数据集输出的一个准确度如何，注错后的神经网络对数据集的输出准确度又是怎样，并进行它们之间的对比分析，看辐照干扰影响对神经网络的影响差别有多大。并对 SNN 网络进一步进行损伤分析，结合实验结果从脉冲神经元的作用机理给出了解释。

第 4 章，主要叙述了 dropout 算法框架模型运用在注错后的神经网络中，基于前面注错后的 CNN 神经网络对数据集的输出准确度，对比加了 dropout 算法

的神经网络输出准确度，并分析了在不同注错环境下的网络输出情况，验证了嵌入了 dropout 算法的方法对受到干扰的神经网络有一定的修复能力，在神经网络芯片的加固方面能起到一种有效的容错能力。

第 5 章，总结了此次论文的主体研究内容，并给出了论文的创新点，展望了未来神经网络芯片在受到空间辐照干扰下加固容错方面的研究方向。

1.4　本章小结

本章刚开始主要介绍了神经网络芯片发展的一个背景以及神经网络芯片在实际环境运用中可能会遇到的一些干扰的问题，围绕抗干扰的问题提出了本论文所需要对神经网络芯片进行加固的算法模型框架。

随后还介绍了目前针对芯片进行加固方法的研究发展状况，重点从传统的加固方法和基于神经网络芯片算法上的加固方法进行了原理分析，它们能在一定程度上对芯片的抗干扰起到作用，但随着神经网络芯片的高度集成化和性能化以及干扰模式的变化，使得以往的方法并不能适应这些新的变化。针对神经网络芯片中存储在 SRAM 区的权重参数如果受到干扰发生错乱的这个问题，提出了基于 dropout 算法的模型框架去对神经网络芯片进行一个容错来避免受到的干扰。并对 SNN 网络这种模拟人脑的工作方式进行了叙述，为后续其受到干扰后的损伤分析提供了一定依据。

本章的最后对论文的研究内容和每一部分章节结构的安排都进行了相关概括说明。

第2章 相关算法模型理论基础

2.1　卷积神经网络

神经网络是现在社会人工智能行业非常流行的一个研究领域，目前最火热的神经网络就是深度卷积神经网络，它在很多社会行业中都已经取得了巨大成功，例如：计算机视觉、汽车的无人自动驾驶、人脸识别、自然语言处理和安防监控行业等。卷积神经网络它帮助人类解决了很多需要人工处理的工作，而且现在都是大数据时代，它能从这些成千上万的复杂数据中去训练学习特征，最后可以把所需要的结果向同类型但未知的数据去进行映射泛化，往往通过算法模型框架的改进就可以大大提高其工作的效率。

CNN 网络随着研究的深入，现在也产生了基于此理论产生了很多优秀的神经网络模型，比如 LeNet-5、ZFNet、NASNet、DenseNet 等[21-24]。网络的内部层数也在不断增加，对数据的泛化能力也越来越强，人们可以完全不需要去关心数据之间的联系，网络能自己从数据中去归一化那些主要特征然后运算出所需要的结果。但是随着网络模型的复杂化，训练的难度也会逐渐变大，内部网络参数的增多也容易导致过拟合化。所以针对不同的输入数据，设计并训练好一个适合的 CNN 网络是很有必要的，下文会从 CNN 网络的一些运作原理进行进一步分析，为后面在网络中利用权重参数的优化进行抗干扰提供理论参考。

2.1.1　神经元

神经网络有众多功能各异的层级，让这些层级相互连接并构成一个系统进行内部数据传递的是一个个神经元，类似人脑，这些神经元构成了功能繁杂的神经网络。一般刚开始的时候神经元之间的运算只是简单的进行线性加权这种级别的计算，随着层级的推进，为了更好的突显数据的特征化分布，会在某些层级后的神经元加上一些非线性的激活函数，这样就会让神经元之间的计算变成非线性的输出。如下图所示，每两个不同神经元之间的连线表示加权值，也称作为权重参数值(Weights)，例如图中的这些权重参数 $\boldsymbol{w}_1, \boldsymbol{w}_2, \boldsymbol{w}_3, \ldots \boldsymbol{w}_n$。输入的就是一系列经过变化过来的数据值，如图 2.1 中的 $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \ldots \boldsymbol{x}_n$ 等，再接过下一次神经元的加权运算后会变成一个新值，这个过程有时也会经常加上一些偏置参数来使网络

更加容易收敛，此时的新值会作为下一层神经元的输入值，神经网络中的很多数据就是这样在层层之间传递运算的。对于图 2.1 中的神经元输出结果，可以表示为如下公式：

$$y_{out} = f(x_1 * w_1 + x_2 * w_2 + x_3 * w_3 + \cdots + x_n * w_n) \quad \ldots (2.1)$$

对于一组输入参数$(x_1, x_2, x_3, \ldots x_n)$，其权重参数为$(w_1, w_2, w_3, \ldots w_n)$，经过神经元加权变化计算后输出为$y_{out}$。

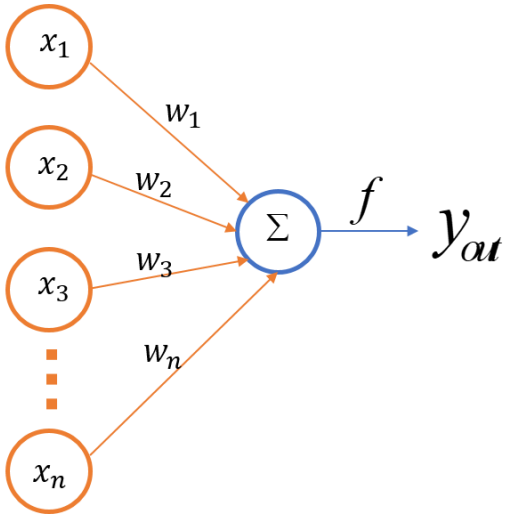


图 2.1　神经元示意图

Figure 2.1　Schematic diagram of neurons

2.1.2　神经网络中的激活函数

(1) 增加激活函数的好处

对于线性加权的函数而言，当输入的数值很大时，那么函数的输出也就是无限趋近于大或者无限趋近于小，那么经过神经网络的内部多层网络计算叠加后，最终的输出的值就会变得没有边界，不符合预期想要的结果，而且很多情形我们希望神经网络输出的是一个概率数值，这样才能很好地区分数据。

如果一个神经网络不使用任何激活函数[25]，那么神经网络内部进行的运算就会一直是一些简单的线性加权变化，对于处理高纬度或者复杂的数据而言就无法进行，例如我们需要对图像分类或者语义分割去进行操作，就需要用到非线性变化。由于计算过程中误差和梯度都会被用来每次去更新权重参数值和偏移值，激活函数也会让神经网络中的反向传播算法变得较为简单。激活函数的

正确使用也会使神经元处于激活状态，能更好的让每个神经元去处理数据，提高对整体数据样本的提取与学习能力。

(2) 常见的激活函数

并非任何函数都可以作为激活函数，在神经网络算法模型的设计中，神经网络的激活函数需要具备其是连续可导的、单调非线性的以及函数输出的范围不饱和性这几个特点。

1) Sigmoid 函数

如下图 2.2 所示是一个 sigmoid 激活函数的示意图，其表达式如下：

$$f(x)=\frac{1}{1+e^{-x}} \qquad \ldots (2.2)$$

可以很明显看出不管输入的数据范围是多大，经过 sigmoid 函数处理后，总能将函数的输出值压缩在 0 到 1 之间，这对一些具有较大范围差的数据集而言就会有一定的收敛作用，会更加方便处理数据。

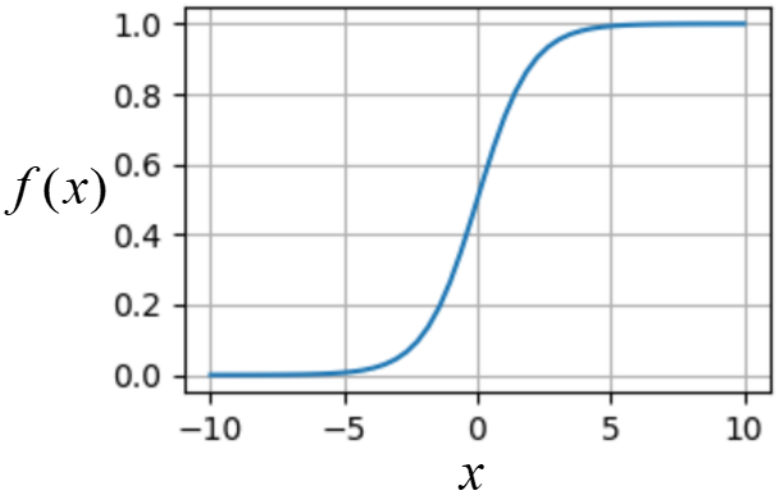


图 2.2　Sigmoid 函数

Figure 2.2　Sigmoid function

2) Tanh 函数

如下图 2.3 所示是一个 tanh 激活函数的示意图，它原理上跟 sigmoid 函数非常相似，实际上就是 sigmoid 函数的一个变化版本，可由下面表达式进行推导得出：

$$f(x) = 2\text{sigmoid(x)} - 1 = 2\frac{1}{1+e^{-x}} - 1 \quad \text{… (2.3)}$$

经计算化简可以得到 tanh 最终的表达式如下所示：

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \quad \text{… (2.4)}$$

它与 sigmoid 函数的最大区别是不用担心输入数据的符号值域问题了，而且 tanh 函数它关于原点中心对称，函数的输出值总能落到-1 到 1 之间，可以让神经网络训练学习收敛速度进一步加快。

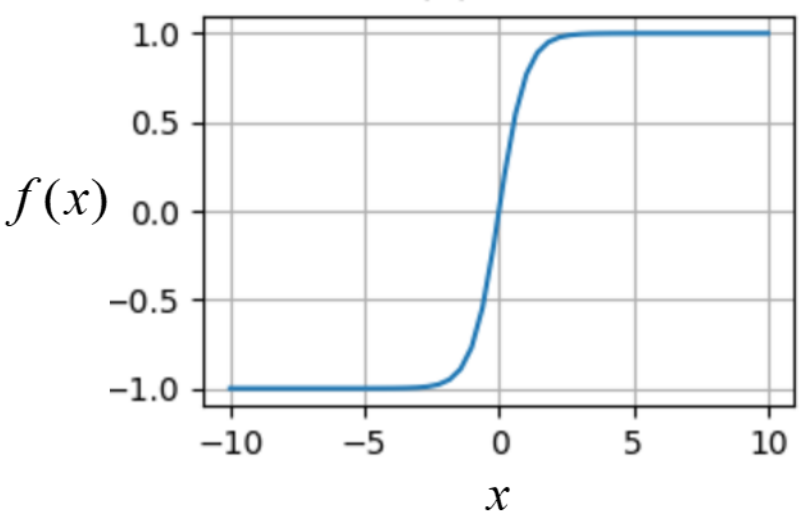


图 2.3　Tanh 函数

Figure 2.3　Tanh function

3)　Relu 函数

Relu 激活函数是这些年来在神经网络设计中比较流行的一个激活函数，如下图 2.4 所示是一个 relu 激活函数的示意图，其函数表达式如下：

$$f(x) = max(0,\ x) \quad \text{…(2.5)}$$

由于有分区划分，所以是一个非线性函数，能很好的反向进行误差的传播计算，并尽可能多的激活神经元，而且从图中可以看出 relu 函数只需要判断数据的输入是否大于 0，可以计算的很快进而神经网络收敛速度也快于前面两个。但是对于输入数据如果 $x < 0$ 的部分，就会直接输出 0，导致部分权重参数无法随着网络的训练学习进行更新，这就会造成部分神经元没有被激活。

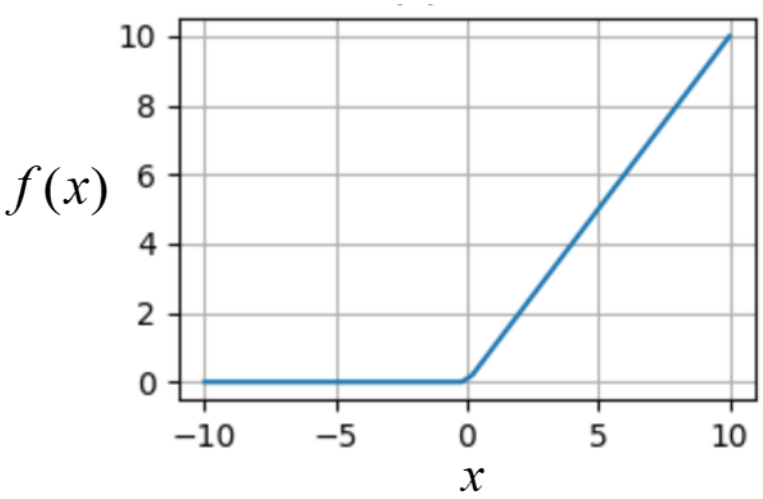


图 2.4　Relu 函数

Figure 2.4　Relu function

4)　Leaky Relu 函数

如下图 2.5 所示是一个 leaky relu[26]激活函数的示意图，它又在 relu 激活函数的基础上进行了进一步改进，其表达式如下：

$$f(x)=\begin{cases} ax, x<0; \\ x, x>=0 \end{cases} \quad \ldots(2.6)$$

其中参数 a 就是斜率，它也可以作为神经网络中一个重要参数来进行学习计算。

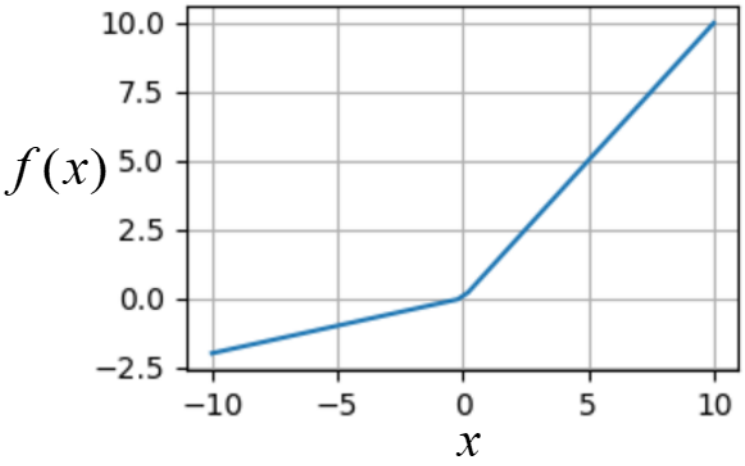


图 2.5　Leaky Relu 函数

Figure 2.5　Leaky Relu function

2.1.3 层级结构

将上文所说的那种单个神经元进行组织结合在一起，并形成了神经网络结构，如下图 2.6 所示就是一个简单的三层神经网络模型结构：

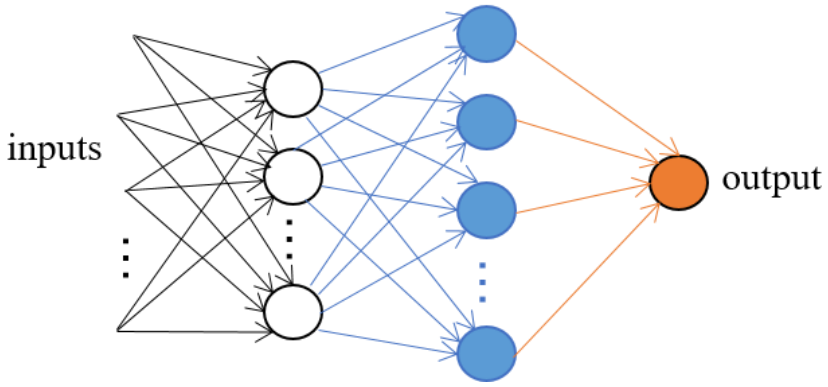


图 2.6 神经网络一般结构

Figure 2.6 General structure of neural network

其中最左边的是网络的输入层，最右边输出的是输出层，而中间的一般是隐藏层，隐藏层一般根据数据集的不同设计出来的网络层次数量也会大相径庭，激活函数一般也会分布在此，隐藏层数量的增多一般对应的激活函数也会增多。同时，对于每一层而言，它可以由单个神经元或者多个神经元在内部连接组成，并且每一层的输出又将作为下一层的输入数据去进行迭代计算，下面对各层进行一个理论说明。

(1) 卷积层

卷积层(Convolutional Layer)是 CNN 网络构成的一个核心层，它会对一系列的输入数据进行特征提取的处理，所以对于神经网络中的大部分计算都是在卷积层完成的，相当于一堆滤波器。对于那些杂乱复杂的大数据，卷积操作可以在计算过程中去重点关注那些数据的特征，例如对图像数据，可以关注某些边缘特征。卷积层几个重要的概念就是感受野(Receptive field)[27]、神经元在网络空间的排列以及权值共享。

其中感受野就是每个神经元与输入数据的某一个局部区域连接的那部分空间大小，实际上它的尺寸大小就是该滤波器的空间大小尺寸，如下图 2.7 所示，在横向的深度方向上，该连接区域空间上的大小总是和输入数据的深度是相等的，图中所示的深度维度上的数值就是 3。

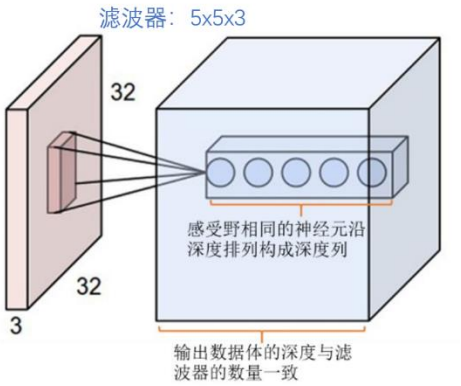


图 2.7　卷积层工作过程

Figure 2.7　Convolutional layer working process

空间排列方式和卷积神经网络中的神经元数量主要由滑动步长、输出深度和边界补零填充等参数控制，其中补零操作的作用是确保输入到卷积层的数据和最终的输出数据具有相同的空间尺寸大小。输入数据尺寸假设为$W_1 \times H_1 \times D_1$，输出数据在空间的尺寸假设为$W_2 \times H_2 \times D_2$，另外神经网络中卷积层中每个神经元的感受野空间尺寸大小为F，滑动步长为S，滤波器数量为K和补零填充的数量为P，则互相间有如下计算关系式：

$$W_2 = \frac{(W_1 - F + 2P)}{S} + 1 \quad \text{… (2.7)}$$

$$H_2 = \frac{(H_1 - F + 2P)}{S} + 1 \quad \text{… (2.8)}$$

$$D_2 = K \quad \text{… (2.9)}$$

一般对于卷积神经网络将其步长设置为 1 时，其补零填充的数值就是：

$$P = \frac{(F - 1)}{2} \quad \text{… (2.10)}$$

而对于卷积核中所含的神经元数量其计算公式如下：

$$\frac{W_1 - F + P}{S} + 1 \quad \text{… (2.11)}$$

有时神经元数量算出来的不是整数，反映到卷积层中的状态就是某些神经元并不能整齐对称地从滤波器窗口中扫描过输入数据，所以这时候就通过补零操作去人为地改变输入数据的空间尺寸大小。

权值共享是针对卷积神经网络中那些卷积核在对数据某一特征提取的情况，这时候权重值集合就可以被卷积核所共享，在神经网络芯片参数存储过程中也能节约一定的地址空间，并且权值共享在一定程度上也会降低神经网络的训练学习过程，而对于处在不同深度的那些神经元其是不会进行共享相同的权值的，只能继续去迭代更新权重值。

(2) 池化层

随着现在数据的复杂和特征维度的上升，在神经网络的训练学习过程中难免会造成庞大的参数矩阵，池化层(Pooling layer)的引入就是用来减少参数矩阵的大小，最后也是为了避免过拟合现象的出现。

目前池化层主要有两种计算方法，平均计算法和最大最大值计算法，其中目前使用最多的数最大值池化方法。如下图 2.8 所示就是一个二维的最大池化计算过程示意图，每次计算时，2X2 大小的池化窗口就会自动地从输入数据的最左上方开始扫描，其扫描的顺序是先从左往右再从上往下，依次扫过的数组矩阵形式如下：

$$\begin{bmatrix} 0 & 1 \\ 3 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix} \rightarrow \begin{bmatrix} 3 & 4 \\ 6 & 7 \end{bmatrix} \rightarrow \begin{bmatrix} 4 & 5 \\ 7 & 8 \end{bmatrix}$$

对每个数组矩阵里面的值取最大值就得到 4，5，7，8 这几个数。

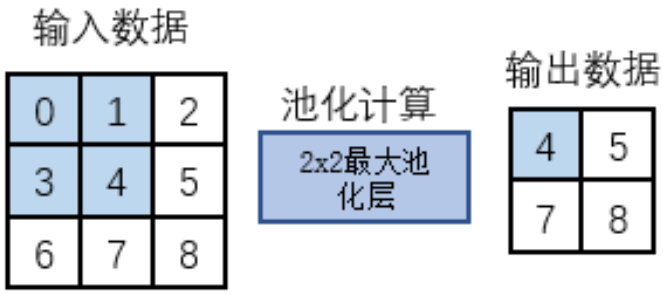


图 2.8 池化层计算过程

Figure 2.8 Pooling layer calculation process

(3) 全连接层

神经网络中的全连接层（Fully connected layers，FC）在卷积神经网络中的作用相当于对前面一层的数据进行了一个分类，起到了一个“分类器”的效果，将输入的样本从前面提取出来的特征空间映射到一个个标签上。

2.2　脉冲神经网络

2.2.1　概念介绍

随着神经网络理论的不断发展和完善，现在脉冲神经网络[30-33]被称为第三代人工神经网络。神经网络的第一代是感知机(Perceptron)，它是一种线性模型结构，一般被用来解决典型的线性二类分类问题，只有两个取值，分别是+1 和-1。感知机的输入数据为处理好的实例数据的特征向量，其输出的是这些实例数据的类别，模型结构比较简单并且很容易设计实现，但它对非线性问题表现得无能无力，只能解决线性的分类问题。神经网络的第二代种类比较多，应用较为广泛的就是 BP 算法(Back Propagation Algorithm)了，它能解决感知机不能解决的非线性问题，并且能取得很好的效果，但是带来的问题就是模型结构变得庞大复杂、参数过多训练难度加大以及容易陷入某个局部的最优解导致过拟合等。

第三代人工神经网络利用了人脑神经元的生物特性，首次将神经元之间数据信号的传递与时间序列结合起来，它将内部神经元的膜电位都设置了兴奋值，随着时间的推移，神经元的信号会随之慢慢衰减，但是在这个传递衰减的过程中，只要膜电位值大于某个神经元所设置兴奋值的阈值，便会将该神经元激活，这样一来，数据信号就可以向更深层的神经网络进行传递，能够获得更丰富更准确的信息，并且会拥有更强的计算处理能力。例如，IBM 公司研制的 TrueNorth 芯片就是基于脉冲神经网络的，功能更加丰富，算力更加强大，但功耗却并没有增加多少。

2.2.2　脉冲神经元

神经网络中一个神经元可以与上一层的多个神经元相连接，脉冲神经网络中的神经元也类似，随着时间序列的推移，它会接收来自多个上层神经元的脉冲序列信号，当该神经元膜电位超过阈值时，这个脉冲神经元[34]就会输出一个新的脉冲信号到下一层的神经元。并且脉冲神经网络由于跟时间有一定的关系[35]，它内部的每个神经元都有可能多次收到数据信号的输入，这样每个神经元就都有被多次激活的可能，而像卷积神经网络那些在每次迭代计算过程中只会利用到一次。所以说脉冲神经网络传递的信息更加丰富，下图 2.9 显示的就是脉冲神经元上信号传递过程以及其上的膜电位变化。

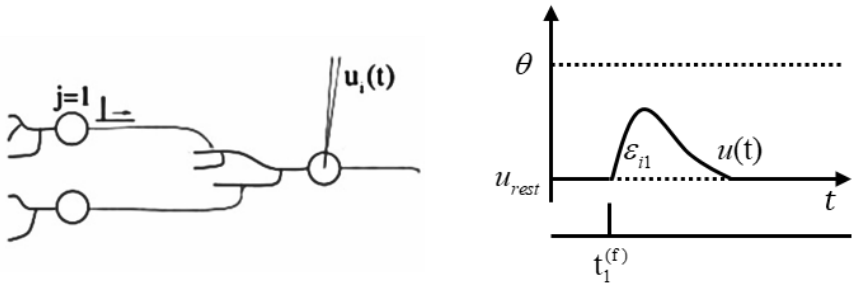


图 2.9　脉冲神经元传递过程及膜电位变化

Figure 2.9　Spiking neuron transmission process and membrane potential changes

图中 $u(t)$ 是与时间序列的函数，表示神经元上膜电位的值大小，u_{rest} 表示睡眠电位，如果有脉冲信号输入，膜电位的值会发生变化，但到最后还是会慢慢的复位到这个睡眠电位上[36],如果神经元接收到的是正值的电位脉冲且超过设定的阈值电位 θ，那么该神经元就会被激活，否则如果是负数电位，那么神经元就会失活不工作。可以看出，在输入脉冲信号还没激励神经元 i 前，有表达式：

$$u_i(t) = u_{rest} \quad \text{… (2.12)}$$

假设在 0 时刻神经元 i 的上一层神经元 j 输出了一个脉冲信号，那么定义神经元 i 与神经元 j 之间的激励后电位为 ε_{ij}，其计算公式为：

$$u_i(t) - u_{rest} =: \varepsilon_{ij}(t) \quad \text{… (2.13)}$$

如果有多个神经元 j =1,2,3…共同作用于神经元 i，则神经元 i 上的膜电位有如下计算式：

$$u_i(t) = \sum_j \sum_f \varepsilon_{ij}\left(t - t_j^{(f)}\right) + u_{\text{rest?}} \quad \text{… (2.14)}$$

随着时间序列的推移，作用在神经元 i 上的脉冲信号激励会越来越多，产生了一个累计效果，如下图 2.10 中所示，当神经元膜电位 $u_i(t)$ 达到并超过该神经元阈值 θ 值时候，神经元就会在内部被激活，并向它后续的神经元发射下一脉冲信号激励，如此迭代传递下去。

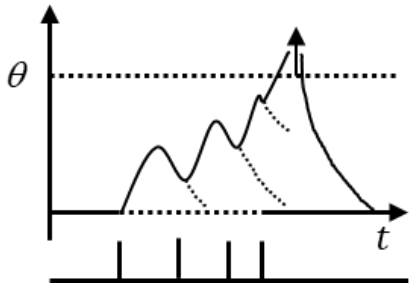


图 2.10　脉冲神经元膜电位超过阈值

Figure 2.10　Spiking neuron membrane potential exceeds threshold

2.3　本章小结

本章节中主要对涉及的一些神经网络算法进行了原理的详细介绍，首先对卷积神经网络的发展历史进行了介绍，说明了卷积神经网络它现在在很多的生活实际领域都非常火，使用非常广泛；接着对卷积神经网络的原理以及用作都进行了分析说明，从构成复杂卷积神经网络的神经元到神经元需要的激活函数都做了阐述，也给出了现在使用比较广泛的激活函数的原理与特点；而后对卷积神经网络中的各个层级做了相应介绍，比如说卷积层是如何发挥作用的，怎样对输入数据去进行卷积计算，怎样利用网络的滤波器数据进行降维操作，池化层是怎样对庞大的参数矩阵进行剪枝精简的，全连接层对输入数据起到的分类器作用等；最后对脉冲神经网络做了理论介绍，重点是对脉冲神经元的介绍，与其他神经网络重要的不同点是脉冲神经网络加入了时间序列这一概念，对脉冲神经元的激活影响因素主要是某一段时间内前前置神经元的膜值电位累加效果。

第3章　软件仿真辐照干扰及参数注错实验过程

3.1　空间辐照环境介绍

现在随着芯片技术的发展，芯片已经越来越智能化、功能化、集成化和复杂化，尤其是搭载着神经网络的芯片。如何保证芯片稳定正常的运行就涉及到对芯片的加固涉及操作。对芯片集成电路构成威胁的第一大要素就是辐射环境了，而且每种辐射环境的情况都不一样，例如像太空中运行的卫星环境，如下图 3.1 所示的卫星，就会受到很多不同类型粒子的干扰，保持维护卫星各个电子器件的稳定健康运行是非常重要的工作。

图 3.1　卫星太空工作示意图

Figure 3.1　Schematic diagram of satellite space work

其实不光只有太空这个环境能产生辐照干扰，有些电子产品中会有各种复杂的新芯片电路，当有大量数据输入输出并且要在瞬时之类处理完成，这就会容易造成一个 rowhammer 攻击，数据的不断重复擦除与写入会增加芯片底层存储器的负担，容易产生电磁噪声，由于数据在底层都是以二进制一位一位储存的，这就会让存储在底层 SRAM 区的一些数据发生二进制翻转，进而导致某些数据出错从而影响整个器件的一个正常工作。另一个重要的辐射环境就是核辐射污染的周围环境，其周围空间会有大量的核辐射，其中某些高能的辐射粒子就会自动去撞击存储器中某些硅材料，这也会导致翻转存储和器件执行单元中某些数据的比特位，进而某些器件由于数据的紊乱而不得没法正常工作。

3.2 单粒子翻转效应

本文在基于神经网路芯片的抗空间辐照干扰研究中，主要考虑单粒子翻转效应模型，它也是一种比较常见的软错误类型。根据研究分析，近年来随着集成电路的发展，基于 SRAM 的芯片在现代 SOC(System on chip)中的使用也是越来越多，其几乎占据了整个芯片尺寸的 90%以上，而这些芯片内部底层的 SRAM 存储器在受到软错误或者相关故障的干扰后，系统的稳定性、可靠性以及可持续性都会受到很大的威胁。

如下图 3.2 所示是芯片底层一个数据在存储区发生比特位翻转的示意图，在高能粒子轰击敏感的存储区时让二进制位的 1 变成了 0，而 0 变成了 1，这就会让存储在这个区域的值发生变化，芯片在下一轮运行中，就会又继续从该存储区域读取这个值进内存重新运行，然而这时这个值却是错误的，芯片内部电路之间的运行是有依赖性的，某些参数值的错误大概率会引起整个芯片电路的链式错误，这样最终就会导致该芯片控制的所有电路发生错误从而让芯片不能正常稳定工作。

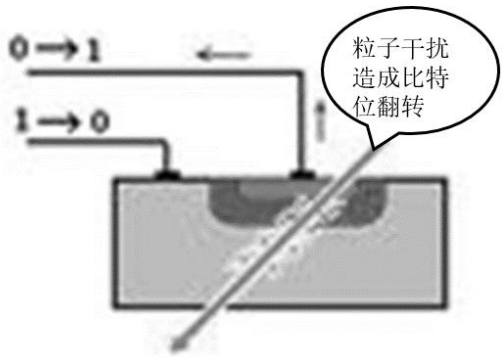


图 3.2 单粒子翻转示意图

Figure 3.2 Single event upset diagram

3.3 辐照影响神经网络芯片的过程

3.3.1 引言

搭载了神经网络的芯片在运行过程中需要处理大量的数据，一般而言，先是对复杂大数据的训练学习，在逐步的训练过程中，神经网络会越来越“聪明”直至最后训练完成，能在测试数据中取得很好的准确度，这时候的神经网络还只是

相当于一套真空中的代码，需要有硬件去搭载。后面的任务就是通过下载平台将神经网络整个模型架构以及一系列参数烧录进芯片，这时候再配合上外围一些复杂的逻辑电路就可以进行正常的工作，如下图 3.3 就是英特尔一款名为 Nervana 的神经网络芯片搭配外围逻辑电路的示意图

图 3.3　英特尔的 Nervana 神经网络芯片

Figure 3.3　Intel's Nervana neural network chip

3.3.2　权重参数的存储

随着神经网络的复杂化、庞大化以及多层化，神经网络的训练参数也越来越多，对于简单的网络少则成千上万，而对于那些复杂的神经网络现在已经有的达到了百万千万级别，可想而知，这些庞大的权重参数也会加大对芯片的硬件资源开销的负担。对于在计算机上训练好的神经网络，其权重参数还不是以二进制形式存储的，而是以一系列的权重参数矩阵组成，如下图 3.4 所示是实验过程中对一个训练好的神经网络进行的部分权重参数截图。

```
tensor([[[[ 0.0127,  0.0487, -0.0666],
          [-0.0592, -0.0727, -0.0318],
          [ 0.0827, -0.0082,  0.0370]],

         [[ 0.0085,  0.0518, -0.0087],
          [-0.0441,  0.0153, -0.0195],
          [-0.0196,  0.0171,  0.0646]],
```

图 3.4　神经网络中的参数矩阵

Figure 3.4　Parameter matrix in neural network

每个参数矩阵都会有它与之对应的维度，这个维度就是在设计神经网络时就根据输入数据的空间尺寸、卷积核、以及步长等参数决定的。由于神经网络每一层的神经元数量以及连接方式都不相同，所以其上相关联的权重参数的矩阵大小也不一样。这些权重参数存储到芯片中时，都是以二进制存储在芯片的 SRAM 区域，例如对于下面图 3.5 中一个存储的 32 位浮点数来说，按照 IEEE754 的对其每一位的规定，可以看出其最高比特位是符号位 S，接着的 8 个比特位是指数位 E，剩下的 23 个比特位是有效位 M，所以图 3.4 中的参数矩阵经过二进制转换就变成下图 3.6 所示的二进制参数矩阵。当因为空间辐照环境的改变造成有单粒子翻转事件发生后，就会作用到芯片的 SRAM 存储区某个部分，会随机将某个权重参数的某一位比特位进行随机翻转，图中所示的是将比特位 1 翻转为 0，这样一来存储在这个区域的权重参数值就放生了改变从而导致变成一个错误的数值，有时单粒子翻转效应作用的并不是某个区域的某一位，可能会由于单粒子轰击的角度不同比如有夹角的轰击，这就会导致某一片区域的比特位都有可能受到单粒子翻转效应从而改变其原来的值。众多权重参数值的改变会影响神经网络的整个处理性能，往往会导致神经网络芯片输出的准确度不高。

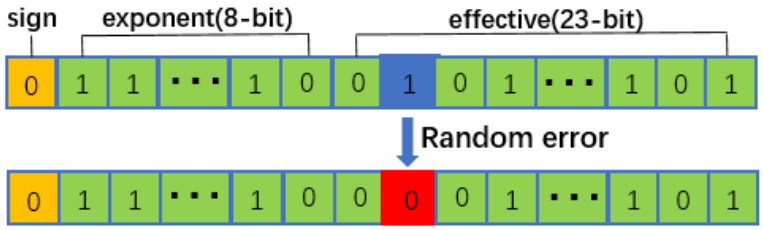


图 3.5 权重参数的二进制存储形式

Figure 3.5 Binary storage form of weight parameters

tensor([[[[0.00000011010000000100111010100 10,0.00001100011101111001101001101 01,1.00010001000011001011001010010 10],
[1.00001111001001111011101100101111,1.00010010100111000111011111001101,1.00001000001001000000101101111 00],
[0.00010101001010111101001111000 01,1.00000010000110010110010100100 101,0.00001001011110001101010011111101]],

[[0.00000010001011010000111001010 11, 0.00001101010000101100001111001 00, 1.00000010001110100010100111000 11],
[1.00001011010010100010001100111 00, 0.00000011111010101011001101001 11, 1.00000100111111011111001110110 10],
[1.00000101000001001000000101101 11, 0.00001000110000010101010011001 10, 0.00010000100010011010000000100 11]],

图 3.6 权重参数矩阵的二进制形式

Figure 3.6 Binary form of weight parameter matrix

3.4　模拟参数注错

3.4.1　网络参数提取

神经网络中的权重参数提取是指在不破坏原有训练好的神经网络的条件下，利用软件程序的嵌入获得网络训练时每次迭代训练的全部结构权重参数，直接让相对应的各层神经元的权重赋值这些参数，保证一致性，这样可以很好的模拟神经网络芯片的工作原理，模拟芯片将训练好的网络和参数预先写入芯片 SRAM 存储区，另外也会对训练好的神经网络进行一个结构保存，在后续测试数据时无需再进行训练，直接处理数据输出结果。

如下图 3.7 所示，就是实验中对神经网络参数提取的一个过程，先是设计好所需要的神经网络模型，这其中就包括神经网络的层级结构、各种激活函数、超参数的选择和调试以及满足误差的迭代次数的控制等。每次迭代训练的学习过程，都将会对神经网络此次的权重参数和网络模型结构进行一次保存，以便在后续的注错环节中使用这些参数来进行下一步实验。这个反映到芯片内部结构地工作就是每次对参数的存储都会在芯片的 SRAM 区去进行不断地重复擦写和擦除，直到最后在训练神经网络时计算出来的误差阈值满足预先设置的误差参数值，这时候神经网络的模型也就训练完成了，但是在每一次的迭代训练过程中都会进行一个对测试数据的准确度记录，以便后面对注错后的神经网络性能的分析。

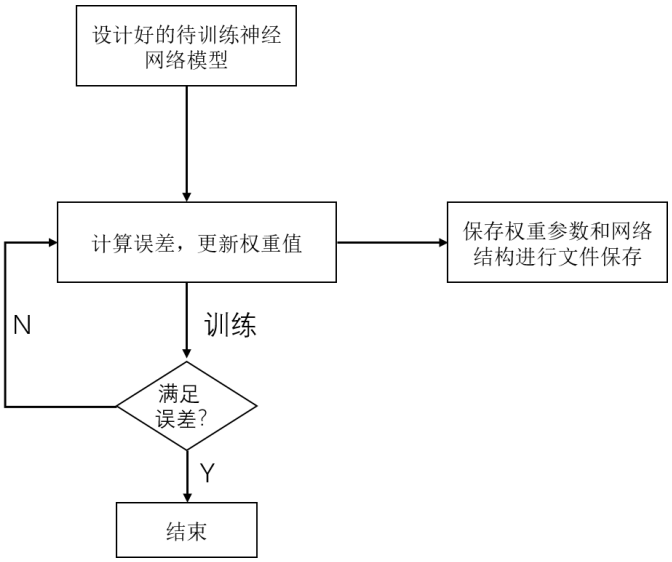


图 3.7　神经网络参数提取过程

Figure 3.7　Neural network parameter extraction process

3.4.2 网络参数的十进制与二进制之间的转化

每次从神经网络中提取出来的参数矩阵，其形式都是方便人们阅读的十进制参数矩阵，而对应到芯片存储区都是经过算法转换后的二进制，只有二进制才可以在芯片内部的 SRAM 区进行存储运行。并且空间辐照的单粒子翻转也是针对存储区的二进制比特位而言，在芯片 SRAM 区，每个二进位就相当于芯片存储区上的各种控制格子单元。如下表 3.1 所示，先是将参数的十进制转换成 32 的二进制数据，然后再利用随机函数 random 在这 32 位随机数据中选取一位进行随机的翻转，这里只演示了对某一位进行翻转，如果需要多个随机位，还可以继续利用随机函数进行翻转模拟转换，最后再将翻转完成后的数据返回回去，这样输出端就得到了一个从十进制经过反转后的二进制。这里芯片就可以直接对二进制数据进行读取处理操作了。

表 3.1 权重参数的十进制转换二进制代码

Table 3.1 Decimal conversion of binary codes for weight parameters

```
import random
def decTobin(x):
    x -= int(x)
    bins = []
    while x:
        x *= 2
        bins.append(1 if x >= 1. else 0)
        x -= int(x)
    len1 = len(bins)
    # 32 位浮点数，不足补零
    for i in range(32 - len1):
        bins.insert(0, 0)
    # 在 32 位数据中随机选取的一位进行翻转
    randomBit = random.randint(0, 32)
    if bins[randomBit] == 0:
        bins[randomBit] = 1
    else:
        bins[randomBit] = 0
  # 最后返回翻转后的二进制数据
    return bins
```

神经网络参数矩阵经过二进制的随机位翻转转换后，由于实验中已经保存了网络模型的参数结构，再测试数据集时，可以不需要再进行训练网络，可以直接拿这个保存的模型结构和对应的参数矩阵去进行测试，所以就需要再将这些二进制参数再转换为十进制参数，如下表 3.2 所示就是权重参数的二进制转换成十进制的相关代码。

表 3.2　权重参数的二进制转换十进制代码

Table 3.2　Binary conversion decimal codes for weight parameters

```
import random
def binTodec(bin):
    binCopy = bin.copy()
    for i in binCopy :
        # 去除前面多余的0位
        if i == 0:
            del bin[0]
        else:
            break
    dec = 0
    for i, x in enumerate(dec):
        dec += 2**(-i - 1) * x
    # 最后返回十进制数据
    return dec
```

3.4.3　网络参数在芯片的映射过程

神经网络的模型复杂，参数众多，在芯片的映射过程大致具体如下图 3.8 所示，该图给出了神经网络映射到芯片存储电路对应的一个示意图，对于初始训练好的神经网络，它们的权重参数都已经确定下来了，每层的每个神经元相互连接之间都会有一个权重参数，虽然我们看到的都是一个个具体的十进制数值，但对于芯片的 SRAM 存储区而言，是一系列转换后的二进制数值，这样它们才能被存储在 SARM 区。待芯片工作后，这些在存储区的权重二进制值会被电路重新加载提取出来并赋值到相应的神经元节点连接处。也正是因为存在这个存储区，由于器件的敏感性，会导致这些存储的二进制数值的某一位或者某几位的二进制比特值发生随机翻转（下图演示的仅仅是一位翻转的情况），这样就会导致一个

错误的权重参数被加载进去电路，对应的神经元就是出错后的神经元，会影响神经网络芯片的整个正常工作流程。

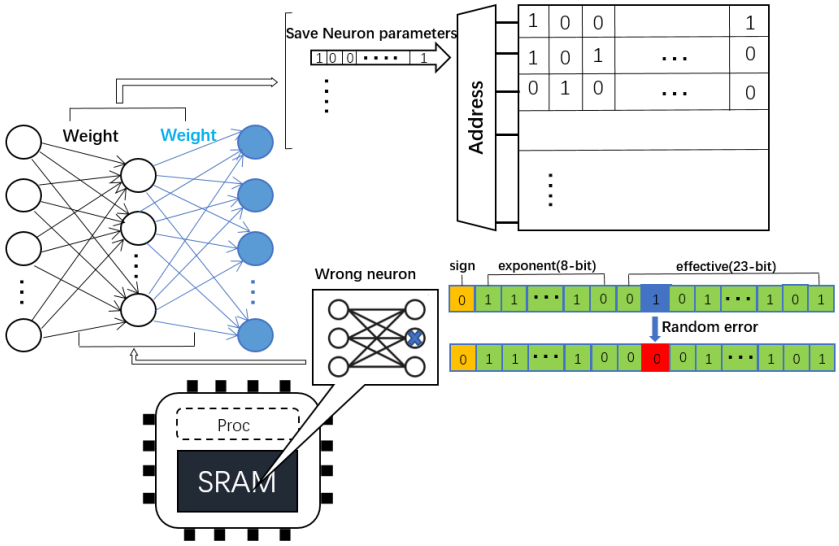


图 3.8　神经网络参数存储电路映射图

Figure 3.8　Neural network parameter storage circuit map

3.5　CNN 注错实验与结果分析

3.5.1　实验说明

(1)　实验的软件仿真环境

本仿真实验硬件环境如下：

操作系统为微软的 windows10 64 位系统，CPU 配置为英特尔的酷睿系列，型号为 i7-8500，8GB 的运行内存，2GB 独立显存，采用 GPU 加速进行处理。

本仿真实验软件环境如下：

软件仿真平台是 Facebook 公司的开源深度学习框架 Pytorch，涉及到的语言是 Python，实验的数据图像绘制主要基于 Matplotlib 库。

(2) 训练与测试数据集

训练网络的数据集是采用来自美国国家标准与技术研究所（National Institute of Standards and Technology，NIST）的 MNIST 手写字体数据集，其训练集是由 60000 个用例所组成，因此其里面也总共包含了对应用例的 60000 个标签，每一个标签的值为 0 到 9 之间的一个数，测试集总共是 10000 个。所以该数据集一直以来被广泛用于测试各种关于图像的神经网络，能在很大程度上说明各种图像处

理方面所遇到的问题。

(3)　CNN 模型结构

CNN 网络类型众多，不失一般性地，本文实验中选取了经典的 LeNet5 网络作为研究对象，该网络是最早获得成功应用的卷积神经网络，包含此类网络的关键构成要素。如下图 3.9 所示，它是一个被广泛应用于图像分类问题的卷积神经网络。对于 LeNet5 网络结构参数，如下表 3.3 所示，给出了具体各层结构的详细描述，经过训练的神经网络权重参数最终都会存储在相关 SRAM 区内，待芯片工作时，便被调用进行运算处理。

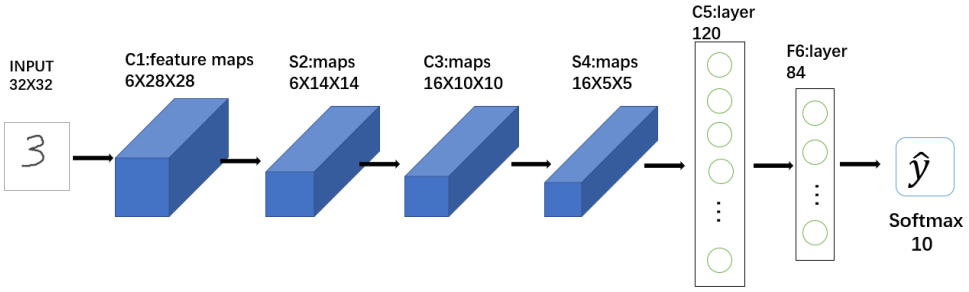


图 3.9　LeNet5 网络模型结构

Figure 3.9　LeNet5 network model structure

表 3.3　LeNet5 各层参数计算

Table3.3　LeNet5 layer parameters

层级		特征图个数	特征图大小	卷积核大小	步长	激活函数
Input	Image	1	32x32	-	-	-
1	C1	6	28x28	5x5	1	Relu
2	S2	6	14x14	2x2	2	Relu
3	C3	16	10x10	5x5	1	Relu
4	S4	16	5x5	2x2	2	Relu
5	C5	120	1x1	5x5	1	Relu
6	FC	-	84	-	-	Relu
Output	FC	-	10	-	-	Softmax

(4)　注错过程

在 CNN 网络训练过程中，对每一次训练好的网络进行一个前文所示的参数

提取过程和网络结构保存步骤，然后再对这些权重参数进行进制转换，最后再按照一定比例选取权重参数进制比特位的翻转变化。

3.5.2 实验仿真结果与分析

在进行软件算法模拟辐照效应中的单粒子翻转扰动时，实验中先后对神经网络中 1‰，1%，5%，10%的比例参数进行随机注错，根据器件的失效率(Failures in Time, FIT)计算公式：

$$1\text{FIT} = \frac{1\text{次失效}}{10\text{^}9\text{小时}} \quad \ldots (3.1)$$

这里实验中所使用的神经网络参数约 60000 个左右，由于是软件程序按照一定比例模拟随机注错，考虑瞬时性，即 1‰，1%，5%，10%的比例参数换算成的失效率结果 FIT 估算分别为 60FIT，600FIT，3000FIT，6000FIT，不同比例参数的选取对于取得的不同结果形成一个对照试验。

如下图 3.10 和 3.11 实验结果所示，在 1‰比例神经网络参数出错的情况下（这种比例已经很大程度模拟单粒子翻转效应），其对测试数据的准确度随着实验迭代次数增加的情况下，最终结果准确度会维持在 0.94 附近左右，而在无单粒子翻转情况下训练好的 LeNet5 网络对 MNIST 数据集的测试结果一般能达到 0.99 左右，所以可以看出单粒子翻转引起参数的随机变化会对结果准确度产生一定的负面影响。

后面实验过程中又继续在 1%，5%，10%的大比例网络参数出错情况下进行测试，发现此时的神经网络结果准确度随着注错比例参数的增加而迅速下降。除此之外，还可以发现，在 1‰比例参数注错的情况下，其迭代过程中的准确度曲线比较平滑，而随着注错比例参数的加大，其相对应的准确度曲线上下偏差震荡的现象表现地非常明显，这是因为此时神经网络中的很多神经元已经被赋值上错误的权重参数，所以导致整个神经网络对测试数据的判断会出现较大的错误，因为对于有几万个参数的神经网络而言，1%比例即为有几百个权重参数同时出错，这对各个神经元之间的关联影响是很大的。

基于上部分软件实验模拟单粒子翻转实验可以看出，芯片中神经网络的权重参数出错对芯片的稳定运行会造成一定的影响。

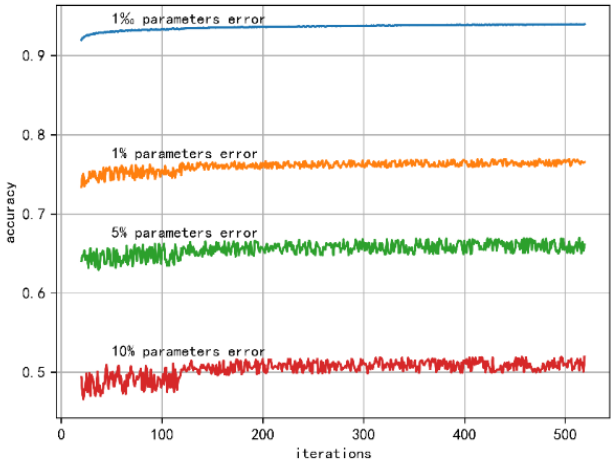


图 3.10　CNN 网络准确度随迭代次数的变化曲线

Figure3.10　Variation curve of accuracy of CNN network with number of iterations

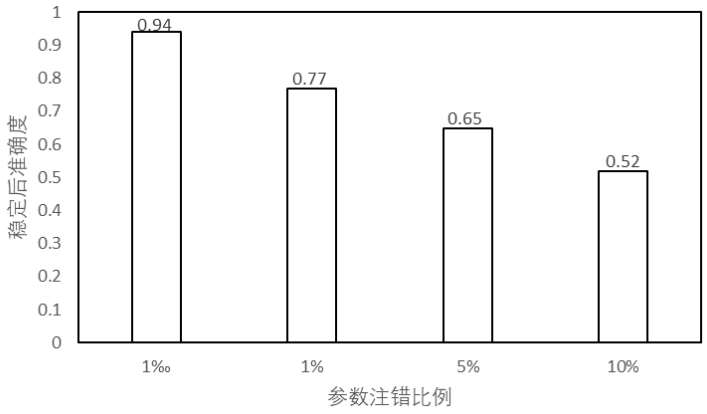


图 3.11　稳定后的准确度对比

Figure 3.11　Comparison of accuracy after stabilization

3.6　SNN 注错实验与损伤分析

3.6.1　实验说明

(1) 实验环境与数据

本节实验所使用的软件平台和硬件平台与上文 CNN 实验一致，并且数据集同样采用了 MNIST 数据集，注错过程依然是按照一定比例去对网络中的权重参数进行随机比特位翻转实验。

(2) SNN 模型结构

在 SNN 网络中，其网络模型结构如下图 3.12 所示，对输入数据加入了一层脉冲编码的转换层[37]，其作用就是将输入数据与时间序列结合起来进行编码操作，假设输入数据 $x \in [\min, \max]$，脉冲编码后对应的时间序列为 $T \in [t_1, t_2]$，则转换层对应的脉冲编码表达式为：

$$T(x) = \frac{t_2 - t_1}{\max - \min} \cdot x + \frac{t_1 \cdot \max - t_2 \cdot \min}{\max - \min} \quad \ldots (3.2)$$

放电时间依赖可塑性(Spike Timing Dependent Plasticity，STDP)层[38-41]的加入，是为了模拟人脑神经元的工作特性，它能根据脉冲神经元在对数据集特征的学习过程中自动调整内部各个神经元之间连接的强度。抑制层的作用也类似人脑，当某个神经元因为接收到某个激励刺激信号而变得兴奋被激活后，它不仅会把该激励信号传递给它连接的后面神经元，同时也会抑制其他对该信号不活跃的神经元的兴奋度，也即是降低其膜值电位的大小。

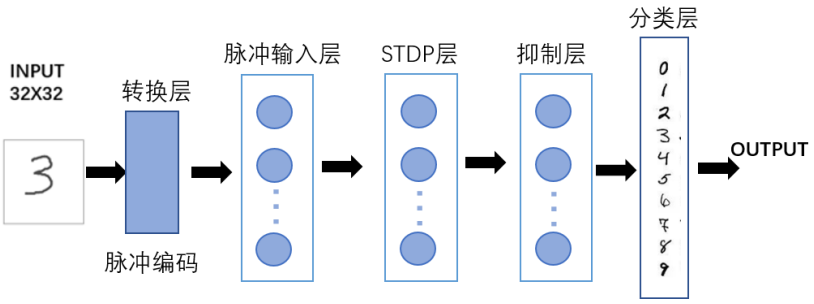


图 3.12　SNN 网络模型结构

Figure 3.12　SNN network model structure

(3)　注错过程

在 SNN 网络训练过程中，也是对每一次训练好的网络权重参数和网络结构进行前文所述的提取保存过程，这里数据预处理不同的就是会对数据进行一个脉冲加时间序列的编码操作，然后再对脉冲神经元上的一定比例权重参数进行比特位翻转操作，最后测试数据也要进行脉冲编码操作最终得出每次的注错结果准确度。

3.6.2　实验结果与损伤分析

将输入数据集进行脉冲序列的处理[42]，并构建上图所示的脉冲神经网络，训练的同时对网络参数权重进行提取并进行注错操作，比例分别为 1‰、1%、5%和 10%，当训练趋于稳定后得到如下图 3.13 和图 3.14 所示的结果，可以看出随着参数注错比例的增加，脉冲神经网络结果输出准确度也在下降，并且曲线的数据

震荡效果比较明显。很明显，SNN 网络相比 CNN 网络在 1‰比例参数出错时其结果准确度只有 0.84 左右，比较低，这是因为脉冲神经网络由于其内部神经元效仿膜值电位的特性，对错误有一个累加的效应并会不断传递给下一层形成一个链式反应，所以准确度变低。

在后面 1%、5%和 10%的比例参数出错情况下，虽然准确度有一定降低，但是随着出错比例增加差距与 CNN 网络结果对比并不明显，因为 SNN 网络有抑制层的加入，它会自动调整其网络内部各个神经元之间连接的强度，类似人脑神经元，会抑制其他神经元的活性，然而这些被抑制的神经元很可能就是那些受到干扰的错误神经元，这就会让错误没有被进一步放大。

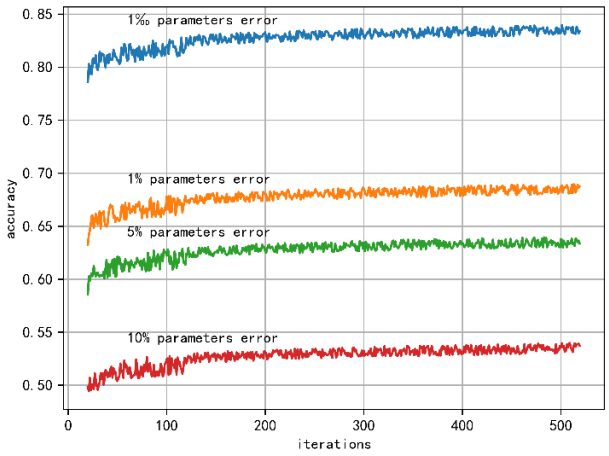


图 3.13　SNN 网络准确度随迭代次数的变化曲线

Figure3.13　Variation curve of SNN network accuracy with number of iterations

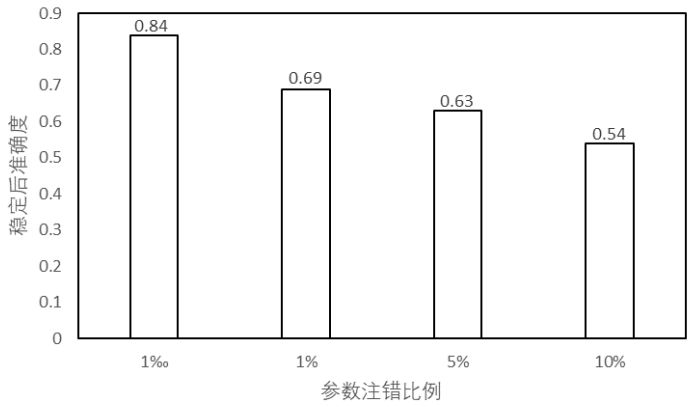


图 3.14　稳定后的准确度对比

Figure 3.14　Comparison of accuracy after stabilization

3.7 本章小结

本章节主要围绕软件仿真辐照干扰以及参数注错实验来说明，先是对空间辐照环境进行了介绍，芯片在这种环境下工作会受到一定的干扰。空间辐照环境中又以单粒子翻转效应最为频繁，它会导致存储在芯片 SRMA 区的一些参数发生比特位的翻转，对于神经网络芯片而言，就是一系列的权重参数了，这些权重参数在芯片底层最终的存储形式都是以二进制一位一位去存储的。接着进行了模拟参数注错的实验说明，这中间会涉及到神经网络参数的提取、参数的十进制与二进制之间的转换以及随机对一些权重参数去进行比特位的翻转等过程。还介绍了网络参数在芯片中的的一个映射电路过程，展示了神经网络的参数是如何在芯片内部进行擦写和清除运作的。

最后选取了 CNN 网络去进行参数的注错分析并对 SNN 网络进行了损伤分析，在注错的过程中先后对神经网络中 1‰，1%，5%，10%的比例参数进行随机注错，得出神经网络芯片在随着权重参数比例上升出错的情况下，神经网络的准确度也会随之下降的结论，除此之外，在 1%，5%，10%的比例参数出错情况下，准确度曲线的震荡现象表现的也比较明显，并且 SNN 在 1‰比例参数出错时相比较 CNN 网络其震荡效果也较为明显。在对上问题的研究可以看出在经过单粒子翻转后干扰的神经网络芯片其准确度会受到一定的影响。

第4章 基于 dropout 算法的抗干扰容错方法

空间辐照干扰尤其是单粒子翻转效应会让存储在神经网络芯片 SRAM 区的一些权重参数发生随机位翻转，进而会直接影响神经网络芯片的输出结果准确度。考虑到这些权重参数是直接作用在神经网络的各层神经元上的，可以从神经元方面去优化神经网络结构，在相比在出错的情况下，能让神经网络芯片的输出准确度得到提高，进而提高神经网络芯片的抗干扰容错的能力。

利用 dropout 算法构建新的网络框架，以一定概率屏蔽因受到辐照干扰产生单粒子翻转而导致某些权重参数出错的病态神经元，让其处于失活状态，这样就可以避免影响后续数据流的传递与输出，从而将神经网络芯片的输出准确度得到一定的提升。

4.1　Dropout 算法原理

一直以来，各种神经网络在面对复杂的大数据训练学习过程中，如何成功训练好神经网络是一个难点。随着神经网络层级的深度化和复杂化，神经网络最终的权重参数也会成百上千的倍数增加，几万几十万甚至几百万都是可以遇见的。这么多参数也就带来了诸多问题，例如难训练、过拟合化、实际运用中存储太占空间以及神经网络受到干扰容易造成参数出错从而影响神经网络的正常输出。怎么样解决这些问题成为了后面神经网络发展领域中的一个重点。

在 2012 年，Geoffery Hinton[43]提出了在神经网络中利用“dropout”算法技术，通过在神经网络的训练过程中去按照一定比例去丢弃一些神经元，让复杂的神经网络模型得以进一步进行剪枝操作，被丢弃的神经元实质还在网络结构中，只是将它们处于不激活的状态，这样它们就不会工作[44-45]，从而依赖于这些神经元上的众多参数也就不需要在神经网络每次训练学习的迭代过程中进行更新操作了，这就大大降低了训练神经网络的复杂度，也节约了时间。在运用中，由于神经网络得以精简，所以后期网络模型和参数值所占用的硬件资源空间也就减少了，提高了芯片的利用率。并且在过拟合问题上[46]，由于神经网络相比较原来的复杂网络剔除了很多不重要的参数，所以整个神经网络的权重参数也因此减少了，这就将参数过多导致的过拟合问题也得到了进一步解决。近年来，在 CNN 网络

模型的训练学习过程中，dropout 已经被广泛使用，并取得了很好的效果。

下面首先从线性网络、非线性的神经网络去介绍算法的使用，为后面在 CNN 芯片中使用该算法创建新的框架模型提供理论基础。

4.1.1 线性网络中使用 dropout

在线性的神经网络中 dropout 的使用是有一定效果的[47]，并且在线性的神经网络中可以根据公式计算得出神经网络的 h 层中神经元 i 的活性，其计算公式如下所示：

$$S_i^h(I) = \sum_{l<h}\sum_j w_{ij}^{hl} S_j^l \quad with \quad S_j^0 = I_j \qquad \ldots (4.1)$$

其中 w 表示对应神经元的权重参数，I 表示神经元前面处理完数据的输入向量。

对于在神经网络单元中使用 dropout，其表达形式如下：

$$S_i^h = \sum_{l<h}\sum_j w_{ij}^{hl} \delta_j^l S_j^l \quad with \quad S_j^0 = I_j \qquad \ldots (4.2)$$

其中 δ_j^l 是伯努利变量，值为 0 或者 1，并且 δ_j^l 是相互独立存在的，没有干扰关系。如果为 0，就表示表达式值为 0，则其所在网络中的神经元出于失活状态，如果为 1，则神经元正常参与神经网络中的各项工作。而对于神经网络的 h 层中神经元 i 的作用，其在 δ_j^l 变量上需要带上神经网络层级之间的关系，其表达式如下：

$$S_i^h = \sum_{l<h}\sum_j \delta_{ij}^{hl} w_{ij}^{hl} S_j^l \quad with \quad S_j^0 = I_j \qquad \ldots (4.3)$$

而对于处理数据后的固定输入向量而言，神经网络中所有神经单元的期望值就可以由被失活和激活状态的神经元计算出来，其期望值如下表达式所示：

$$E\left(S_i^h\right) = \sum_{l<h}\sum_j w_{ij}^{hl} p_j^l E\left(S_j^l\right) \quad for \quad h > 0 \qquad \ldots (4.4)$$

其中 $E\left(S_j^0\right) = I_j$ 则表示神经网络第 0 层也就是输入层的期望值，原始的权重值为 w_{ij}^{hl}，经过 dropout 的伯努利变化后其新的神经元的权重值就变成 $w_{ij}^{hl} p_j^l$。

4.1.2　神经网络中使用 dropout

由前面公式 2.1 可知，如果神经元的权重参数受到空间辐照干扰而让原来存储在芯片 SRAM 区的参数值发生改变，则神经元的输出就会变为：

$$y'_{out} = f(x'_1 * w'_1 + x'_2 * w'_2 + x'_3 * w'_3 + \cdots + x'_n * w'_n) \quad \ldots (4.5)$$

其中 w'_1、w'_2、$w'_3 \cdots w'_n$ 是原始权重值随机出错后的结果，非线性函数 f 一般是神经元的激活函数。

Dropout 是对现在神经网络模型的又一次内部剪枝优化[49-50]，它里面的核心思想是在训练的神经网络中对一部分神经元进行以伯努利概率 p 的屏蔽，也就是使其激活值即权重参数以伯努利概率 p 变为 0，让其停止工作，神经元此时也就失活不参与网络后续的数据传输计算，这样一来就可以减少网络中那些对输入数据检测的特征检测器（隐层节点）间的相互影响作用（检测器的相互作用是一种相互的双向链式关系，它指网络中的某些检测器会依赖其他某些检测器才能发挥自己原来对数据特征检测作用），从而提升神经网络对数据的归纳泛化效果，得出一个很好的准确度。如下面图 4.1(a)所示，对于一个具有 L 个隐藏层的神经网络而言，可以令层级参数为 $l \in \{1,2\ldots,L\}$，其表示神经网络中隐藏层的第 l 层，设 $z^{(l)}$ 为第 l 层神经元的一组输入，$y^{(l)}$ 为神经网络第 l 层的输出。$w^{(l)}$ 和 $b^{(l)}$ 为第 l 层的权重参数和偏差，这样一个标准神经网络的前馈操作可表示为(对于 $l \in \{0,1\ldots,L-1\}$ 和任何隐藏层 i)，其表达式如下：

$$z_i^{(l+1)} = w_i^{(l+1)} y^l + b_i^{(l+1)} \quad \ldots (4.6)$$

$$y_i^{(l+1)} = f(z_i^{(l+1)}) \quad \ldots (4.7)$$

这里的 f 同样是激活函数。

采用 dropout 算法对网络进行操作后，如下图 4.1(b)所示，这里 $r_j^{(l)}$ 的值会根据伯努利分布以概率 p 置为 0 或 1，相应的表示第 l 层的第 j 个权重值会置为 0 或保留原来的值，其过程表达式如下：

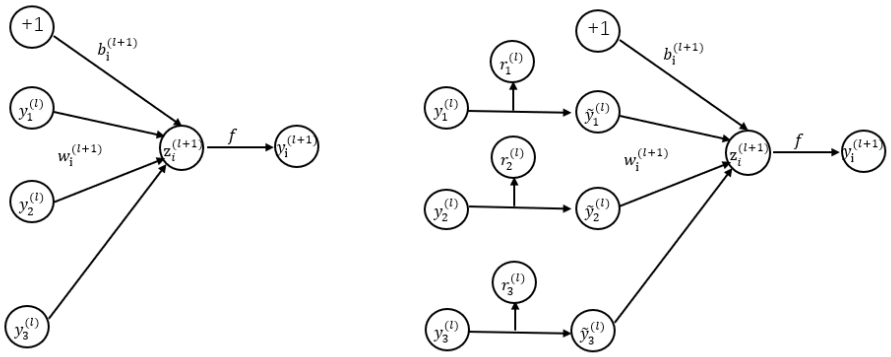
$$r_j^{(l)} \sim Bernoulli(p) \quad \ldots (4.8)$$

$$\boldsymbol{y}^{(l)} = \boldsymbol{r}^{(l)} * \boldsymbol{y}^{(l)} \quad \ldots (4.9)$$

$$\boldsymbol{z}_i^{(l+1)} = \boldsymbol{w}_i^{(l+1)} \boldsymbol{y}^{l} + \boldsymbol{b}_i^{(l+1)} \quad \ldots (4.10)$$

$$\boldsymbol{y}_i^{(l+1)} = f\left(\boldsymbol{z}_i^{(l+1)}\right) \quad \ldots (4.11)$$

其中 $Bernoulli(p)$ 就是按照伯努利的概率 p 去进行赋值操作。

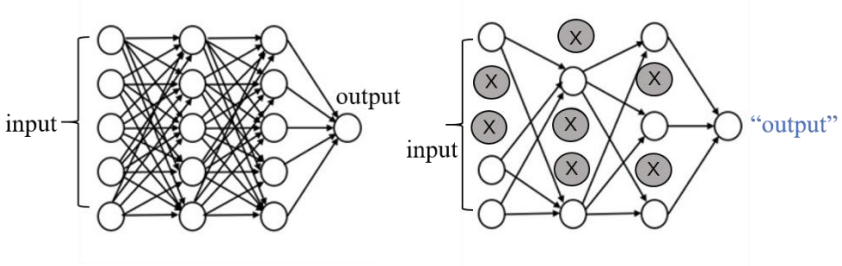


（a）正常标准的神经元　　（b）dropout 后的神经元

图 4.1　神经元参数表示

Figure 4.1　Neuron parameters representation

一个完整的标准的神经网络如下图 4.2(a)所示，正常的训练流程一般是首先把输入数据通过神经网络前向传播，然后再把误差值反向传递以此来决定神经网络如何不断更新参数让网络进行优化学习[51]，而 dropout 的过程是随机删除神经网络中部分隐藏神经元，例如图 4.2(b)所示，未进行连接的神经元即是被删除的神经元，在这一过程中，删除的神经元参数仍然保持不变，只是按照伯努利概率分布将其置为 0，即将部分神经元按照一定概率进行屏蔽关闭。



(a)标准的神经网络　　（b）dropout 后的神经网络

图 4.2　神经网络修改前后对比

Figure 4.2　Comparison of the neural network before and after modification

4.2　算法实验过程

针对神经网络芯片在受到辐照干扰影响后会降低神经网络输出准确度的情况，本章将改进后的 dropout 算法模型框架结合在神经网络后续的测试过程，并在前文对 CNN 网络实验中参数注错实验的基础上，提出了一种基于 dropout 算法的抗干扰容错方法。

基于 dropout 算法的模型实验流程框架图如下图 4.3 所示，从实验流程图可以看出主要有这几个步骤：训练神经网络的同时去提取权重参数、对神经元网络进行不同比例的网络参数注错操作并记录准确度、对部分神经元进行 dropout 操作并记录准确度。

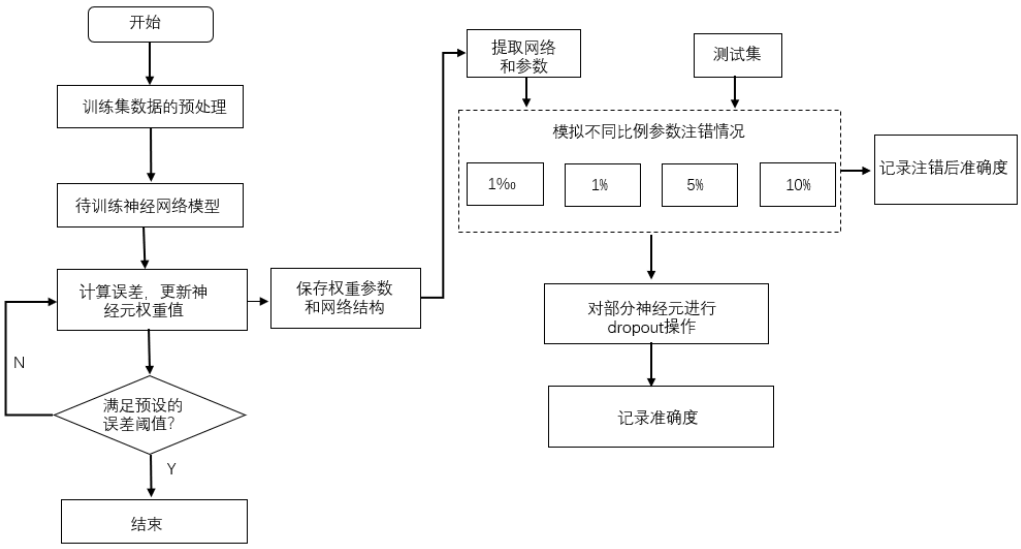


图 4.3　实验流程框架图

Figure 4.3　Diagram of the experiment flow

Dropout 算法中主要的操作就是按照一定概率将神经网络中部分受到干扰的神经元的权重参数 $\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{w}_3,\ldots\boldsymbol{w}_n$ 变为 $\boldsymbol{w}_1^{'}$，$\boldsymbol{w}_2^{'}$，$\boldsymbol{w}_3^{'}\ldots\boldsymbol{w}_n^{'}$，下表 4.1 给出了实验过程中一些伪代码描述以及具体算法过程：

1) 训练未经单粒子翻转引起错误的网络并得到权重参数；
2) 每次迭代结束后的同时将网络的权重参数和网络结构保存并对相关权重参数进行十进制与二进制之间的转换；
3) 对权重参数按照一定比例进行随机注错以达到模拟单粒子翻转效应干扰的效果；

4) 对神经网络的神经元进行 dropout 操作，然后返回原来的网络；

5) 在未满足预设误差阈值下循环进行一定次数的迭代并测试数据集，计算并得到每一次的结果准确度。

表 4.1 神经网络参数注错前后迭代计算伪代码

Table 4.1 Pseudo code of the iterative calculation before and after the neural network parameter error injection

1.	Initialize network and data sets
2.	foreach iteration do
3.	foreach training network and compute loss error do
4.	determine whether the error is consistent
5.	save $\boldsymbol{w}_1$, $\boldsymbol{w}_2$,$\boldsymbol{w}_n$ and convert to binary
6.	different scale parameter errors 1‰，1%，5%，10%
7.	making random errors: $\boldsymbol{w}_1 \rightarrow \boldsymbol{w}'_1$, $\boldsymbol{w}_2 \rightarrow \boldsymbol{w}'_2$,$\boldsymbol{w}_n \rightarrow \boldsymbol{w}'_n$
8.	dropout neural network
9.	end for
10.	record accuracy results
11.	end for

4.3 实验仿真结果与分析

实验环境是在微软的 windows10 64 位系统下，CPU 配置为英特尔的酷睿系列，型号为 i7-8500，8GB 运行内存，2GB 独立显存，采用 GPU 加速进行处理。软件仿真平台基于 Facebook 公司的开源深度学习框架 Pytorch，涉及到的软件程序编写是基于 Python 编程语言，实验的数据图像绘制主要基于 Python 中的 Matplotlib 库。实验的训练数据集和测数据集是 MNIST 手写数据集，神经网络是基于 LeNet-5 卷积神经网络。

针对 1‰比例权重参数在受到干扰出错的情况下，因为 1‰比例的权重参数出错接近现实中受到单粒子翻转引起芯片的失效率，对比了正常训练的网络和注错并进行 dropout 后的网络，得到的实验结果如图 4.4 所示。

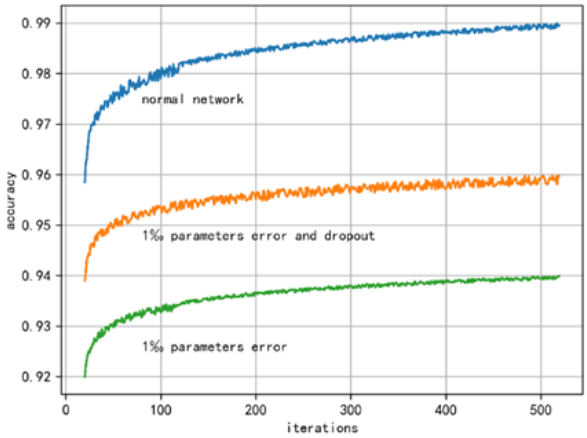


图 4.4　三种情况对比

Figure 4.4　Comparison of the three situations

从上面图 4.4 可以看出，未进行单粒子翻转和进行过 dropout 算法改进的 LeNet5 网络对 MNIST 数据集的测试准确度达到了 0.99 左右，符合预期效果。而在 1‰比例参数出错的情况下，还是跟前图 3.9 一致维持在 0.94 左右，但进行 dropout 算法改进后的神经网络准确度较仅仅注错情况下有明显的提升效果，进行一定次数迭代后，快达到了 0.96 的准确度，提升约 2 个百分点，效果也比较明显，能起到一定的容错作用。

为了进一步验证 dropout 算法在网络中的应用可以提高神经网络的一个容错能力，实验也另外对于 1%，5%和 10%的比例权重参数出错也进行了相关实验，对比之前的注错后的结果，得出的数据如下表 4.2 和图 4.5 所示。

表 4.2　神经元有无 dropout 操作的准确度对比

Table 4.2 Comparison of the accuracy with or without the dropout operation

注错权重参数比例	对神经元无操作的准确度	对神经元进行 dropout 的准确度
1%	0.77	0.80
5%	0.65	0.71
10%	0.52	0.59

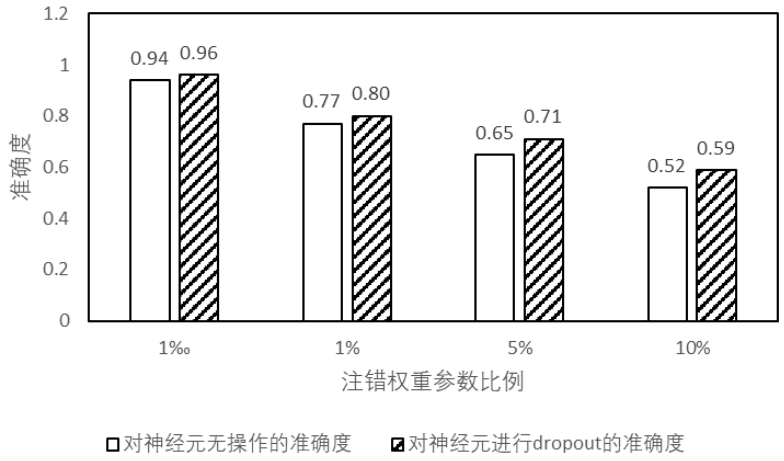


图 4.5 改进前后准确度对比图

Figure 4.5 Comparison chart of accuracy before and after improvement

从以上所得的实验数据可以发现随着权重参数注错比例的不断加大，神经网络准确度下降的比较明显，在利用本文提出的 dropout 算法模型框架对神经网络进行后续的测试过程中，发现神经网络的准确度会有一定的提升，并且在出错比例越大的情况下，准确度提升效果的百分点也会随之增高。综上，在神经网络芯片中，基于 dropout 算法的抗干扰方法起到了良好的容错作用。

4.4 本章小结

针对神经网络芯片在受到空间辐照干扰尤其是单粒子翻转效应时，其网络的权重参数会在芯片 SRAM 区发生二进制比特位翻转，神经网络输出结果准确度下降的问题，提出了一种基于 dropout 算法模型，主要是以一定概率屏蔽因受到辐照干扰产生单粒子翻转而导致某些权重参数出错的病态神经元，让其在运行过程中处于失活状态。本章首先介绍了 dropout 算法的理论知识，说明了其在线性网络和神经网络中使用的理论依据，随后给出了整个实验的设计过程步骤，主要内容就是训练网络的同时保存权重参数和网络结构、对权重参数进行一定比例的注错并记录结果以及对注错后的神经网络再进行 dropout 算法处理并也记录结果，实验结果表明随着对权重参数注错比例的增加，神经网络的结果准确度会随之下降，然后随着 dropout 算法的加入结合，在测试数据上可以看到 dropout 算法的抗干扰设计是一种有效的容错方法，能够提升神经网络的输出准确度。

第5章 总结与展望

5.1　总结

空间辐照干扰尤其是单粒子翻转效应会对神经网络芯片的正常稳定运行造成一定的影响，使其结果输出的准确度降低。本文通过对神经网络赋值权重和单粒子翻转之间找到一定规律，来屏蔽神经网络中某些高风险神经元，避免神经网络芯片故障，达到抗单粒子翻转的目的，进而增加芯片中网络的一个容错性，想法有一定的新意并且在后续相关研究中有重要意义。

此次论文的主要工作成果和创新点汇总如下：

1、实现了在神经网络训练过程中对其神经元上的权重参数进行提取保存，并对这些权重参数进行了随机位的翻转以模拟空间辐照中单粒子翻转的实验，发现随着出错比例参数的增加，神经网络的性能降低，输出结果准确度下降。

2、针对上述问题提出了基于 dropout 算法的模型框架，对那些受到干扰的神经元进行一定概率的屏蔽，让其处于一个失活状态，可以从实验看出，算法对网络输出的结果有一定的提高，增加了神经网络芯片的容错性。

以上两部分的相关实验过程和研究成果经过后续的整理和分析后已在核心期刊上发表论文一篇。

3、也对 SNN 网络的注错后性能进行了损伤分析，由于其增加了脉冲编码、STDP 和抑制层等，脉冲神经元的相关活性对错误的累加效果比较明显，网络在出错后所受到的影响比较大，但随着错误的增多，抑制层对神经元又发挥作用，一定程度上减弱了错误的影响。

5.2　展望

本论文对神经网络芯片在空间辐照抗干扰和容错方面的研究，在实验过程中也有相关的不足之处和后续需要改进的方面，结合现在对神经网络芯片的研究现状和趋势，后续工作可以从下面几个重点着手：

1、空间辐照干扰包括多种形式的干扰，既有软错误方面也有硬错误方面，所以后面工作考虑问题需要更加分类清楚和全面。

2、神经网络芯片本身内部含有很多各种各样功能的电路，除了 SRAM 存储

器部分外，还包括了很多外围的一些数据传输和控制的逻辑电路，可以后续考虑将多种适合的有效加固方法结合起来进行操作。

3、现在神经网络芯片功能强大，搭载的网络层数和规模也进一步增加，尤其是 SNN 网络的脉冲神经元类似人脑神经元，数量非常大，后续工作可以考虑将生物学和下一代神经网络芯片进行结合，对类脑方面的研究有一定意义。

4、芯片工作的环境不光受到辐照干扰影响，还有其他各种因素的影响，所以可以考虑后续有相关实物芯片的搭载试验验证。

参考文献

[1] Akopyan F, Sawada J, Cassidy A, et al. Truenorth: Design and tool flow of a 65 mw 1 million neuron programmable neurosynaptic chip[J]. IEEE transactions on computer-aided design of integrated circuits and systems, 2015, 34(10): 1537-1557.

[2] Chen T, Du Z, Sun N, et al. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning[J]. ACM SIGARCH Computer Architecture News, 2014, 42(1): 269-284.

[3] Xu L, Yu R, Wang L, et al. Memway: in-memorywaylaying acceleration for practical rowhammer attacks against binaries[J]. Tsinghua Science and Technology, 2019, 24(5): 535-545.

[4] Kim Y, Daly R, Kim J, et al. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors[J]. ACM SIGARCH Computer Architecture News, 2014, 42(3): 361-372.

[5] 景乃锋. 面向 SRAM 型 FPGA 软错误的可靠性评估与容错算法研究[D].上海交通大学,2012.

[6] Zhu M, Song N, Pan X. Mitigation and experiment on neutron induced single-event upsets in SRAM-based FPGAs[J]. IEEE Transactions on Nuclear Science, 2013, 60(4): 3063-3073.

[7] 薛玉雄, 曹洲, 杨世宇, 等. IDT6116 单粒子敏感性评估试验技术研究[J]. 原子能科学技术, 2008, 42(1): 22-27.

[8] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Advances in neural information processing systems. 2014: 2672-2680.

[9] Radford A, Metz L, Chintala S. Unsupervised representation learning with deep convolutional generative adversarial networks[J]. arXiv preprint arXiv:1511.06434, 2015.

[10] Karras T, Laine S, Aila T. A style-based generator architecture for generative adversarial networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2019: 4401-4410.

[11] Higgins I, Matthey L, Pal A, et al. beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework[J]. Iclr, 2017, 2(5): 6.

[12] Kusner M J, Paige B, Hernández-Lobato J M. Grammar variational autoencoder[C]//Proceedings of the 34th International Conference on Machine Learning-Volume 70. JMLR. org, 2017: 1945-1954.

[13] Chen X, Kingma D P, Salimans T, et al. Variational lossy autoencoder[J]. arXiv preprint arXiv:1611.02731, 2016.

[14] Wirthlin M J, Keller A M, McCloskey C, et al. SEU mitigation and validation of the LEON3 soft processor using triple modular redundancy for space processing[C]//Proceedings of the

2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2016: 205-214.

[15] Yao R, Chen Q, Li Z, et al. Multi-objective evolutionary design of selective triple modular redundancy systems against SEUs[J]. Chinese Journal of Aeronautics, 2015, 28(3): 804-813.

[16] 王子龙,郑美松,涂吉,王骏也,李立健.SRAM 型 FPGA 的基于可观性度量的选择性三模冗余方法[J].计算机辅助设计与图形学学报,2015,27(11):2184-2191.

[17] 曹靓, 田海燕, 王栋. 一种抗单粒子瞬态辐射效应的自刷新三模冗余触发器[J]. 电子与封装, 2018, 18(9): 36-38.

[18] Saleh A M, Serrano J J, Patel J H. Reliability of scrubbing recovery-techniques for memory systems[J]. IEEE transactions on reliability, 1990, 39(1): 114-122.

[19] 李永进,毛健彪,黄金锋.存储系统中的芯片纠错算法研究与设计[J].计算机工程与科学,2013,35(04):24-28.

[20] 郭向英,赵雷,沈沛.面向单粒子效应的航天嵌入式软件软防护技术研究[J].质量与可靠性,2013(01):54-58.

[21] El-Sawy A, Hazem E L B, Loey M. CNN for handwritten arabic digits recognition based on LeNet-5[C]//International conference on advanced intelligent systems and informatics. Springer, Cham, 2016: 566-575.

[22] Fu L, Feng Y, Majeed Y, et al. Kiwifruit detection in field images using Faster R-CNN with ZFNet[J]. IFAC-PapersOnLine, 2018, 51(17): 45-50.

[23] Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 8697-8710.

[24] Huang G, Liu S, Van der Maaten L, et al. Condensenet: An efficient densenet using learned group convolutions[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018: 2752-2761.

[25] Karlik B, Olgac A V. Performance analysis of various activation functions in generalized MLP architectures of neural networks[J]. International Journal of Artificial Intelligence and Expert Systems, 2011, 1(4): 111-122.

[26] Zhang X, Zou Y, Shi W. Dilated convolution neural network with LeakyReLU for environmental sound classification[C]//2017 22nd International Conference on Digital Signal Processing (DSP). IEEE, 2017: 1-5.

[27] Luo W, Li Y, Urtasun R, et al. Understanding the effective receptive field in deep convolutional neural networks[C]//Advances in neural information processing systems. 2016: 4898-4906.

[28] Dias C, Bueno J, Borges E, et al. Simulating the Behaviour of Choquet-Like (pre) Aggregation Functions for Image Resizing in the Pooling Layer of Deep Learning Networks[C]//International Fuzzy Systems Association World Congress. Springer, Cham, 2019: 224-236.

[29] Zhang C L, Luo J H, Wei X S, et al. In defense of fully connected layers in visual representation transfer[C]//Pacific Rim Conference on Multimedia. Springer, Cham, 2017: 807-817.

[30] 蔺想红,王向文,张宁,马慧芳.脉冲神经网络的监督学习算法研究综述[J].电子学报,2015,43(03):577-586.

[31] 王向文. 基于脉冲序列内积的脉冲神经网络监督学习研究[D].西北师范大学,2015.

[32] Piekniewski F, Richert M, Fisher D, et al. Rate stabilization through plasticity in spiking neuron network: U.S. Patent 9,275,326[P]. 2016-3-1.

[33] Sinyavskiy O, Polonichko V, Izhikevich E, et al. Apparatus and methods for efficient updates in spiking neuron network: U.S. Patent 9,256,823[P]. 2016-2-9.

[34] 徐彦, 熊迎军, 杨静. 脉冲神经元脉冲序列学习方法综述与比较[J]. 计算机应用, 2018: 0-0.

[35] 黄丽鸿,谌先敢,刘海华.模拟初级视皮层脉冲神经元的动作识别系统[J].自动化学报,2012,38(12):1975-1984.

[36] 蔡荣太,吴庆祥,王平.脉冲神经元的信息处理[J].计算机与现代化,2010(11):45-49.

[37] 程龙,刘洋.脉冲神经网络:模型、学习算法与应用[J].控制与决策,2018,33(05):923-937.

[38] Kheradpisheh S R, Ganjtabesh M, Thorpe S J, et al. STDP-based spiking deep convolutional neural networks for object recognition[J]. Neural Networks, 2018, 99: 56-67.

[39] Yousefzadeh A, Masquelier T, Serrano-Gotarredona T, et al. Hardware implementation of convolutional STDP for on-line visual feature learning[C]//2017 IEEE International Symposium on Circuits and Systems (ISCAS). IEEE, 2017: 1-4.

[40] 阮承妹. 基于 STDP 的脉冲神经网络学习算法的研究[D].福建师范大学,2013.

[41] 柯成仁. 基于 STDP 的脉冲神经网络图像识别算法研究[D].西安电子科技大学,2018.

[42] 马建宇, 孟祥, 赵莹. 基于脉冲神经网络的手写体数字识别[J]. 数字技术与应用, 2019 (5): 43.

[43] Srivastava N, Hinton G, Krizhevsky A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. The journal of machine learning research, 2014, 15(1): 1929-1958.

[44] Xiao T, Li H, Ouyang W, et al. Learning deep feature representations with domain guided dropout for person re-identification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1249-1258.

[45] Wager S, Wang S, Liang P S. Dropout training as adaptive regularization[C]//Advances in neural information processing systems. 2013: 351-359.

[46] Gal Y, Ghahramani Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning[C]//international conference on machine learning. 2016: 1050-1059.

[47] Baldi P, Sadowski P J. Understanding dropout[C]//Advances in neural information processing systems. 2013: 2814-2822.

[48] Lucchese C, Nardini F M, Orlando S, et al. X-DART: Blending dropout and pruning for efficient learning to rank[C]//Proceedings of the 40th International ACM SIGIR Conference

on Research and Development in Information Retrieval. 2017: 1077-1080.

[49] Zhou Y, Zhang Y, Wang Y, et al. Accelerate CNN via Recursive Bayesian Pruning[C]//Proceedings of the IEEE International Conference on Computer Vision. 2019: 3306-3315.

[50] Gomez A N, Zhang I, Swersky K, et al. Learning sparse networks using targeted dropout[J]. arXiv preprint arXiv:1905.13678, 2019.

[51] Poernomo A, Kang D K. Biased dropout and crossmap dropout: learning towards effective dropout regularization in convolutional neural network[J]. Neural Networks, 2018, 104: 60-67.

致 谢

时光飞逝，转眼之间，我的三年研究生学习生涯也随着这篇毕业论文的结尾马上要结束了。回首在这三年攻读研究生学位的时光里，我收获了很多重要东西，不光学习科研上的，还有日常生活上的，更重要的是人生道路教诲上的。回想研究生刚入学时的懵懂到现在的慢慢地成长自立，这期间与给予我生活和学习上帮助的指导老师、同学、朋友和支持自己的家人的关系是分不开的，没有他们在背后默默的帮助、关心和支持，也就不会有今天的自己。在研究生学习科研期间，从产生的各种问题疑惑到组会上老师同学之间的各种讨论，课后文献资料的查阅和理解消化，让我在这一步一步的学习中加深了对自己专业知识的了解，老师同学之间的各种集思广益，也大大开阔了我对专业知识观点的认知。

这里首先感谢我的恩师梁旭文导师对我当初的知遇之恩，为我提供了能够进入中科院微小卫星创新研究院学习生活的机会，能够成为他所带的众多学生之一，我由衷的感到非常荣幸。谢谢他在科研方向上的指导，让我对整个科研学习生涯有了明确的规划。也感谢他组织的师生聚会，感受到了梁老师的桃李满天下，在这里，不仅收获了很多已经毕业的师兄师姐传授的工作经验，也收获了在读师兄师姐对于科研细节上的一些建议。这种同门之间的互帮互助，大家庭间的温馨氛围让我收获颇多。再次感谢梁老师，希望他身体健康，工作顺利。

感谢指导老师谢卓辰老师，他无论在我平时的科研任务上，还是学习生活上，都对我帮助很大。从我研一的选课到论文的撰写都给予了莫大的帮助，对我科研路上的每一个细节要求都让我学习到了他身上严谨的治学态度，多少次的身心交谈和教诲让我受益匪浅。组会上的答疑解惑到课后论文逐字逐句的修改，让我对他的感激之情无以言表。此次课题研究和论文撰写的完成，都离不开他的帮助，再次对他表示诚挚的感谢。希望他一生平安幸福，科研顺利。

感谢钱玉璧师兄、陈文豪师兄、侯缋玲师姐、赖训飞师兄的帮助，是他们在关于我开题以及小论文的建议上，对我科研任务的顺利进行以及小论文的最终发表提供了有力帮助。同时也感谢王慧玲、李宗旺、贺晓赫在每次组会上关于课题

的讨论与建议。再次感谢他们，希望他们在今后的道路上前程似锦、生活如意。

然后感谢从研一到研三遇到的不同室友，每一年都因为寝室人员变动，让我三年都有不同的室友，正因为如此，学习生活上也丰富了很多。感谢研一的室友周方明、夏师懿、胡洪铭，在一起上课学习生活上的互帮互助，谢谢你们。感谢研二的室友李鑫师兄，作为一个过来人，他对我学习上还有找工作上都给我提前说了好多建议，让我少走了很多弯路，谢谢他。感谢现在的室友陈少杰同学，在学习生活上和人生态度上的各种疑惑，和他交流给予了我很大帮助，日常带我一起健身房健身，一起去图书馆学习，还有寝室熄灯后的夜谈，都在很大程度上缓解了我科研上的压力，谢谢他。希望他们幸福安康，事业有成。

感谢中科院微小卫星创新研究院和中国科学技术大学提供这么好的资源以及学习条件，让我能够安心科研学习，希望它们未来发展越来越好。

感谢含辛茹苦把我养大的爸爸妈妈，二十多年的养育之恩和对我人生的影响，他们无私奉献的爱还有满怀殷切的希望是我不断前进的动力，我会在未来道路上继续努力。

最后感谢各位评审老师们，感谢你们在百忙之中抽出宝贵时间来评审论文并给出宝贵的意见。

作者简历及攻读学位期间发表的学术论文与研究成果

作者简历：

2013 年 9 月—2017 年 6 月，在安徽大学电气工程与自动化学院获得学士学位；

2017 年 9 月—2018 年 6 月，在中国科学技术大学代培学习一年；

2018 年 8 月—2020 年 6 月，在中国科学院微小卫星创新研究院攻读硕士研究生学位。

已发表（或正式接受）的学术论文：

[1] 钱欢，谢卓辰，梁旭文. 基于 dropout 算法的卷积神经网络单粒子翻转容错方法研究[J].中国科学院大学学报.（已录用）

申请的专利：

王慧玲，钱欢，谢卓辰，梁旭文。一种针对空间应用的神经网络芯片抗辐照系统及方法。申请号：202010098399.7

所获得的奖项：

2019 年华为软件精英挑战赛上合赛区决赛二等奖