硕士学位论文

神经网络加速器在太空辐照下的损伤分析与控制

作者姓名：王慧玲

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：工程硕士

学科专业：电子与通信工程

培养单位：中国科学院微小卫星创新研究院

2020 年 6 月

Damage analysis and control of neural network accelerator under space irradiation

A thesis submitted to

University of Chinese Academy of Sciences

in partial fulfillment of the requirement

for the degree of

Master of Engineering

in Electronics and Communication Engineering

By

Wang Hui Ling

Supervisor : Professor Liang Xu Wen

Associate Professor Xie Zhuo Chen

Institute of Microsatellite Innovation, 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

神经网络技术在图像识别，数据挖掘、计算机视觉等领域得到了广泛的应用，随着要处理的数据越来越多，网络结构越来越复杂，从而需要消耗大量的计算资源。为了保证神经网络推断过程的实效性，谷歌、ARM 和寒武纪等相继推出神经网络加速器，这类系统对工作环境的要求非常苛刻。尤其是当考虑将将这类加速器应用到卫星系统中时，空间辐照对于加速器存储单元的干扰，尤其是单粒子翻转效应，会使得存储单元的参数出错，该错误映射到神经网络中会造成神经网络最后的输出结果出现偏差。本文从神经网络加速器的结构特点和神经网络算法的特性分析，结合神经网络算法自身的容错能力来展开研究，以达到提高神经网络容错能力的目的。主要研究内容如下：

1、分析神经网络加速器的数据流加速方式以及神经网络推断过程的主要特点，权值参数出错随着数据流的扩散会对最后的输出结果造成影响。结合太空辐照对加速器存储单元 SRAM 造成的单粒子翻转的影响，建立单粒子翻转概率模型，通过软件模拟方式来对网络参数进行注错。

2、从激活函数的非线性分析不同激活函数的错误隐蔽能力，通过注错实验分析验证双边抑制效果的函数错误隐蔽能力更强，进一步提出在目标函数中添加惩罚项来寻找 CNN 最优模型与容错模型之间的平衡。考虑到 CNN 中卷积层权值共享的特点，其参数出错对输出结果准确率的影响更大，本论文提出在卷积层后加入归一化层，并通过注错实验验证其提高网络的容错能力的可行性。并通过对 RNN 进行注错实验分析，验证归一化算法依然可以提高网络的容错能力。

关键词：神经网络加速器 ，神经网络算法 ，容错

Abstract

Neural network technology is widely used in the fields of image recognition, data mining, computer vision, etc. More and more data to be processed is converted, and the network structure is more complex, which requires the consumption of a lot of computing resources. In order to ensure the effectiveness of the inference process of neural networks, Google, ARM and Cambrian have successively launched neural network accelerators. Such systems have very strict requirements on the working environment. The interference of the accelerator's storage unit, especially the single-particle staggering effect, will cause the storage unit's parameters to be wrong. This error mapping into the neural network will cause the final output of the neural network to be biased. The characteristic analysis of the algorithm, combined with the fault tolerance of the neural network algorithm itself, expands the research to achieve the purpose of improving the fault tolerance of the neural network. The main research contents are as follows:

1. Analyzing the dataflow acceleration method of the neural network accelerator and the main characteristics of the neural network inference process. The error of the weight parameter will affect the final output result as the data stream spreads. Combined with the effect of space irradiation on the single particle flip caused by the accelerator storage unit SRAM, a single event upset probability model was established, and the error injection of network parameters were annotated through software simulation.

2. From the non-linear analysis of the activation function, the error concealment ability of different activation functions is analyzed, and the error concealment ability of the function that verifies the bilateral suppression effect through the analysis of error injection is stronger. It is further proposed to add a penalty term to the objective function to find the balance between the CNN optimal model and the fault-tolerant model Considering the characteristics of weight sharing of convolutional layers in CNN, the

error of the parameters has a greater impact on the accuracy of the output results. This paper proposes to add a normalization layer after the convolution layer, and to verify its feasibility to improve the fault tolerance of the network through error injection experiments And through error analysis of RNN, it is verified that the normalization algorithm can still improve the fault tolerance of the network.

Key words: Neural network accelerator, neural network algorithm,fault tolerance

目 录

图 目 录

表 目 录

缩 略 词

DNN	Deep Neural Networks	深度神经网络
k-NN	k-Nearest Neighbor	K 近邻
SVM	Super Vector Machine	支持向量机
SEL	Single Event Latchup	单粒子闩锁
SEGR	Single Event Gate Rupture	单粒子栅穿
SEB	Singles Event Burnour	单粒子烧毁
SET	Single Event Transient	单粒子瞬态
SEU	Single Event Upset	单粒子翻转
ASIC	Application Specific Integrated Circuit	专用集成电路
CNN	Convolutional Neural Network	卷积神经网络
MLP	Multi-Layer Perception	多层感知机
FPGA	Field Programmable Gate Array	现场可编程阵列
RNN	Recursive(Recurrent) Neural Network	结构(时间)型递归神经网络
LSTM	Long Short Term Memory	长短时记忆网络

第 1 章　绪论

神经网络在图像识别[1]、语音识别[2]以及信号检测[3]等领域的应用越来越广泛，并且在相关领域表现出了很大的优势。但是随着神经网络的发展，其网络结构和计算都越来越复杂，对相关运算处理器的要求越来越高，因此为了获得更好地处理性能，越来越多的研究者开始提出专用的神经网络处理器来对网络运算进行加速。通常情况下这些神经网络专用加速器都只进行神经网络的推理阶段，而关于模型的训练是通过其他通用处理器进行离线训练得到的，然后在理想条件下将其网络结构和参数等按照相关逻辑部署到专用处理器中。考虑到当将这类专用加速器应用到卫星系统，作为星载设备对卫星数据进行处理时，其会受到太空辐照的影响，系统的工作环境并非理想状态，从而使得实际的计算结果与预期的结果之间有一定误差，原本训练好的最优网络模型在太空的实际工作环境中可能会发生准确率下降甚至整个加速器工作失效的情况。因此，本文在训练过程中，考虑到太空辐照对加速器的存储单元造成的单粒子翻转效应的影响，得到网络容错模型，提高网络的识别准确率。

1.1　研究背景与意义

早在上个世纪四十年代神经网络的概念就被提出，六十年代深度神经网络提出，但并未得到广泛的应用。随着大数据和高性能计算系统的出现，为神经网络的发展及研究提供了很好地条件，神经网络的优越性逐渐得到体现。不同于早期人为设定的一些规则与方法，神经网络在于对大量数据进行统计学习后获得输入数据的有效表示，具有从原始的数据中提取高级特征的能力。虽然神经网络在许多的人工智能任务上都表现出了很好地处理性能，但针对不同的应用和处理神经网络有不同的计算需求。在 DNN 模型训练的过程中，大量的权值更新迭代通常需要一个大的数据集和大量的计算资源，而且训练时间通常也需要消耗数小时到数天，因此模型训练的过程通常是在云端进行或者是通过其他通用处理器处理。而在实际的应用过程中，为了减少通信的成本，提高计算效率，通常需要在传感器附近进行网络的推断处理，也就是需要本地处理，减少云计算的依赖，降低安全

风险。因此专家学者开始了神经网络加速器的研究，当将这类加速器应用到卫星系统中时，考虑到太空辐照环境对硬件设备的影响，会导致相关的网络模型的精度下降或者失效。

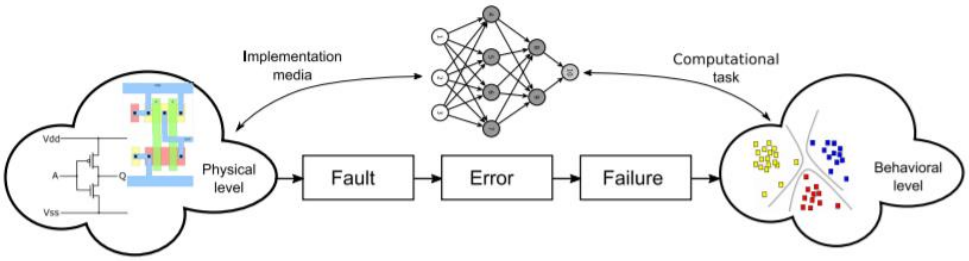


图 1.1　故障与错误之间的关系以及其从神经网络模型实现层面到行为应用层面的传播[4]

Fig1.1 Cause effect relationship between fault, error and failure, and its propagation from the physical-implementation level to the behavioral application level of a neural network model

如图 1.1 所示，首先就神经网络模型结构和其物理实现层面的关系，由于在空间环境中，由于粒子的辐射与冲击造成的电子设备的故障通过电路的传输，并且映射到神经网络模型中，从而影响网络最后输出结果的准确率。

1.1.1　深度学习与神经网络

上世纪 90 年代初，LeCun[5、6]等人关于手写数字识别的 LeNet 网络结构的提出开启了深度学习的新研究。2012 年，Krizhesky 在 ImageNet 大规模视觉识别竞赛中利用深度学习方法取得了很好的比赛结果，掀起了深度学习发展的新浪潮。因此大规模的训练数据集，新的深度学习技术，高性能的计算硬件都给深度学习的发展提供了条件，越来越多的场景中可以看到深度学习的的身影，其在电子侦察，电子支援侦察和雷达威胁警告系统以及遥感数据处理中也有所应用，并且已经得了很好的效果[7、8]。

神经网络是深度学习领域中应用最为广泛的网络框架，例如在图像领域的卷积神经网络和在语音文本识别领域的循环神经网络。神经网络的应用主要分为两个阶段：训练阶段和推理阶段。训练阶段是通过学习不断的更新自身的网络参数从而达到目标优化的一个过程，推理阶段则是将已经学习得到最优的网络参数及

网络模型保存下来用于实际的应用中。近些年来，随着相关技术的发展，伴随着网络的识别结果测准确率越来越高，网络的层数也在逐渐增加，网络结构越来越复杂。

1.1.2 神经网络加速器

伴随着神经网络深度越来越深，网络参数的量级也在逐渐增加，无论是模型训练还是模型推断的过程都需要大量的计算，例如 Alexnet、ResNet[9]等的参数量已经达到百万级，推理一次需要进行的乘法和加法的运算量则超过千万级。对于类似于 ImageNet 数据集[10]，如果采用 CPU 进行推算，需要大量的时间，作为实际应用，其时间消耗在大多数情况下是不满足要求的。因此研究者开始提出各类基于 FPGA、ASIC 的加速器以提高相关的性能要求。Chen 等人在 2013 年提出了国际上首个深度学习处理器——“Diannao”[11],其面积是通用 CPU 的 1/10，性能却是通用 CPU 的 100 倍；2014 年提出国际首个多核深度学习处理器，相较于主流 GPU，21 倍性能比，300 倍功耗比[12]；2015 年提出国际首个通用机器学习处理器[13]，除了可以实现人工神经网络还可以支持 k-NN、SVM 等主流机器学习方法，在性能相当的条件下面积功耗只有 GPU 的 1/100，同年还提出一款专注于手机等终端的智能识别处理器——“Shidiannao”[14]。但是由于神经网络在训练过程中的反向传播算法较为复杂，在这类专用处理器上不易实现，且训练过程需要消耗大量的时间，因此在实际应用中更多的还是采用离线训练好的模型，而训练过程则是在其他处理器上进行。因此大多数情况下的加速器主要是对网络的推理过程进行加速，部署到加速器上的参数数据都是先一步离线训练得到的。

我们知道星地之间的数据传输过程是一个大数据、高通量的过程，首先由卫星将原始数据传送给数据收集、遥测和跟踪站，然后通过地面通信系统再将数据传送到地面的计算机系统和控制中心。如果将神经网络加速器应用到卫星系统对原始数据先进行一个预处理，例如受到观测环境的影响，大部分情况下会有云层的遮挡，这类受到云覆盖影响的数据不仅影响图像的质量，对于后期的研究也并不能起到很大的作用，从而可以在进行星、地数据传输之前就将这类数据舍弃，进而节省大量资源，达到一个高效率的传输。但将神经网络加速器作为星载设备应用时，需要考虑太空辐照的影响。

1.1.3 太空辐照带来的损伤问题分析

在空间环境中，分布着来自宇宙深处的大量粒子，例如质子、电子、重离子等，这些粒子造成的辐射和冲击会对电子设备造成诸多的影响[15]。但就卫星系统的实际情况，一般对于器件影响较大的为总剂量效应和单粒子效应。

总剂量效应：高能粒子入射到电子设备时，将部分能量或者全部能力转移给吸收体，带电粒子损失的能量也就是吸收体所吸收的能量，即为总剂量效应。总剂量效应会导致星载电子设备性能损失或失效。在典型的任务周期和环境下，卫星系统所受到的总剂量效应通常可以不作考虑[16]。

单粒子效应：是指在太空辐照环境中，高能粒子入射到半导体电路中在粒子入射路径上会发生电离效应，电路中的节点可能会受到损坏[17]。虽然不同粒子对于不同工艺材料的电路的作用机理各不一样，但其最终的表现形式是大致相同的。根据辐射机理和辐射导致的结果不同，一般将单粒子效应分成硬错误和软错误两种类型。硬错误是指电路受到辐射后，通过重新写入或者断电都无法使受到损伤的电路状态恢复到正常，即粒子撞击对电路造成了永久性的伤害，主要包括单粒子闩锁（SEL）、单粒子栅穿(SEGR)以及单粒子烧毁（SEB）等。软错误则是指可以通过重写或者是复位的方式进行修复，不会对半导体电路造成永久性的损伤，主要包括单粒子瞬态（SET）和单粒子翻转（SEU）等[18,19,20]。空间环境中的高能粒子的能量很大，电子设备作为星载设备应用时如果仅仅依靠硬件进行屏蔽加固并不能完全避免这类错误的发生，尤其是随着电路集成化越来越高，电子器件的尺寸越来越小，电路中软错误的发生频率亦越来越高。汪波等人[21]指出专用集成芯片（ASIC）在轨翻转率和其总位数与使用率成正比，对于同一款 ASIC 芯片，使用的位数越多，翻转的概率越大。

随着基础半导体技术变得越来越不可靠，计算设备的某些组件发生故障的可能性也随之增加，尤其是在复杂的空间环境中，如果想将神经网络算法应用到卫星系统中，更是需要考虑相关网络的容错能力。

1.1.4 研究意义

为了减少这类干扰的影响，随着技术的发展，器件工艺对于辐照效应有了很大的改进，例如 SOI FinFET 器件[22, 23]，其体硅（SOI）结构的埋氧化层使器件

间完全隔离，消除了单粒子闩锁效应。但是该结构相对较厚的衬底和埋氧层依然无法阻止质子的穿透，从底部入射的质子同样可以在耗尽区产生能量沉积从而产生单粒子翻转错误。另外随着电路集成度越来越高，采用的器件越来越倾向于纳米级，它的直径可能小于重离子入射径迹的直径，导致的单粒子效应通常会对邻近的几个器件同时造成影响，进而发生多位翻转。

软件算法上可以利用冗余信息来进行容错，较为经典的有基于硬件的三模冗余的方法设计，在电路中设计三个相同的模块，对相同的输入数据进行相同的操作，再对三个模块的输出数据进行表决，取两个及两个以上相同的数据作为最后的输出，原因是因为模块之间的独立性导致了两个模块同时发生错误的概率非常低，进而可以得到一个较为准确的结果。对于集成度较高的星载神经网络芯片来说，这不仅是对芯片硬件资源以及卫星系统空间体积的一个消耗，三模冗余方法加上了一个多数表决器操作，对于神经网络这类模型结构，本身的已经是一种分布式并行计算模式，如果再加上三模冗余的设计，这对于神经网络加速器而言，具有一定的挑战性。

周期性擦洗方法不需要附加的电路系统设计，但是对于神经网络处理器的计算效率以及在推断过程中的数据更新速度而言，周期性擦洗对于神经网络处理器的控制单元具有一定的效果，但是针对存储网络参数的部分而言，如果发生了单粒子翻转错误，并不能自行自我校验达到纠错容错的目的。

传统芯片针对单粒子翻转所采取的硬件措施例如看门狗技术，配置擦洗加三模冗余等方法并不适用于集成度较高的神经网络加速器，因此从神经网络加速器自身所实现的神经网络算法角度分析这类问题非常有必要。

1.2 国内外研究现状

深度学习近年来的迅速发展，为各类问题的处理都提供了一个新的思路，同时为了得到更好地性能，越来越多的学者提出采用 ASIC 来对 CNN 进行加速，并且期待将这类处理器应用到卫星系统中，但需要考虑到空间辐照环境的影响，尽可能提高结果准确率的精度。

1.2.1 深度学习以及神经网络的研究现状

神经网络的研究在其实在某种意义上来说是一种仿生物学的研究，基本原理在于模拟人脑神经网络的信号传输处理过程。LeCun 等人针对手写数字数据集的提出了著名的 LeNet-5 网络结构，为后续神经网络的发展奠定了一定的基础。但是随着大数据时代的到来，这类网络渐渐不足以满足要求。于是伴随着 AlexNet、Vgg[24]、GoogleNet[25]、ResNet 等越来越深层结构的网络模型出现，网络在数据不断的非线性叠加中最终得到的结果也越来越精确，因此其应用也越来越广泛。譬如：在电子侦察系统中，利用卷积神经网络（CNN）提取雷达辐射源信号的有效特征细节，并学习特征编码过程，可以实现雷达辐射源的准确识别[6]；在遥感图像处理中，采用神经网络提取融合光谱特征和空间特征，进而提高图像分类精度[7]。

针对一些真实应用的场景，对于大而复杂的网络模型，会面临内存不足的问题，同时还需要尽量要求要有较快的响应速度，因此这些层数多、结构复杂的网络很多时候很难达到相关要求。因此针对这种情况，研究者提出了一系列解决方案：韩松等人提出了一种基于低值连接的删除策略，该剪枝方法包括 3 个阶段，即训练连接，删除连接，重训练权值[26]。2016 年 Inadola F N 等人提出了一种轻量化的网络 sequezze net，采用 1*1 的小卷积核替换 3*3 的卷积核并对网络结构进行模块化操作，在大量减少模型参数的情况下依然保证精度不损失[27]。Courbariaux M.等人提出的二值化网络，将权值和隐含层的激活值二值化为 1 或者-1，减小模型参数占用的存储空间的同时，利用位操作代替乘法加法运算，提高运算效率的同时，依然可以取得和浮点型网络相当的精度[28]。

1.2.2 神经网络加速器的研究现状

人工智能应用场景的特定化以及相关产权的保护，越来越多的企业以及研究者加大对于人工智能芯片的研究力度。对于用于端侧的处理器，主要是用于推理，往往无法承担巨大的运算量，因此一般芯片架构以 FPGA[29]、ASIC[10]为主。当前神经网络加速器的发展方向主要是以硬件加速为主要目的，由通用向专用过渡，例如以谷歌为代表的张量处理器 TPU，配合 TensorFlow 框架使用；麻省理工学院（MIT）的 Eyeriss 以及 IBM 的 Truenorth（真北）芯片基于打破了传统的冯诺依曼结构，模仿了人脑的结果，使用运算单元作为神经元，存储

单元当做神经突触，将所有的处理单元采用全新的方式互联。

国内关于人工智能芯片的研究，最早由中科院的寒武纪系列芯片，在 2014-2016 年期间提出了多个基于 ASIC 的专为深度学习设计的 IP，后来有占比很大部分关于神经网络加速器的研究都是在寒武纪“电脑”系列芯片的基础上进行展开的，例如华为发布的端侧芯片麒麟 980；百度发布的面向人工智能技术的昆仑芯片等。

1.2.3 抗辐照干扰研究现状

小卫星具有轻量级、体积小、研发周期短但功能密度大等特点，为了满足这些要求，星载设备一般的集成度都很高，而且其工作在复杂的空间环境，会受到太空辐照的影响，因此对于其抗辐照能力、可靠性都具有一定的要求。

常规的关于电子设备的抗辐照能力、容错能力的提高通常是通过增加其在空间或者时间上的冗余来实现的。文献[30]提出的龙芯 X 微处理器从不同方面进行抗辐照考虑，综合了电路设计、物理实现、系统结构以及软件应用四个橙子来进行抗辐照的加固。文献[31]基于 SOI 工艺的 SRAM，提出了一种脉冲屏蔽元结构，通过在标准的 SRAM 的六管单元中加入延迟结构，从而增大电路对于单粒子效应的响应时间，达到对粒子冲击造成的脉冲电流的屏蔽作用。

随着神经网络以及神经网络加速器的迅猛发展，近年来研究辐照干扰对于神经网络加速器以及神经网络鲁棒性的影响也逐渐称为热点问题。

Cesar Torres-Huitzil 为了提高神经网络的容错能力，从神经网络的设计和硬件实施两个方面给出了参考性建议[32]。Minjae Lee[33]分析了定点前馈深度神经网络的容错能力，考虑了由互连以及处理单元引起的电路错误,提出了在训练过程中随机断开权重以提高错误恢复能力。A.Assoum[34]分析了人工神经网络在空间环境中抗单粒子翻转的鲁棒性。Austin P.Arechiga[35]测试了 VGG16,ResNet50 和 InceptionV3 三种不同的网络结构在推断阶段参数出错时的网络鲁棒性。Sangheon Kwon[36]测试了 LeNet 结构对网络权值注入高斯噪声的鲁棒性，发现卷积层具有彼此不同的误差容限。Austin P.Arechiga[37]分析了单粒子翻转错误造成的权值出错对纯卷积网络和多层感知器的影响。

1.3 论文的主要工作

从前文的分析可以得到，当把神经网络加速器作为星载设备应用到卫星系统中时，由于空间环境的影响会产生辐照效应，因此并不能直接应用到卫星系统中，需要对其进行抗辐照加固。传统的抗辐照技术虽然已经成熟，但是对于基于 ASIC 实现神经网络推断的专用芯片来说，这类技术并不能很好地适用。神经网络模型一般分为两个阶段训练和推断，但当它部署到卫星系统中时，一般只进行网络推断过程，不存在针对已经发生的相关错误对在重新训练学习以达到提高网络准确率的效果。

因此，本文从神经网络加速器的架构和数据流分析，考虑太空辐照对加速器的存储单元产生的单粒子翻转错误，并将此错误映射到神经网络的权值参数上，分析其对神经网络预测精度的影响。

从卷积神经网络自身的网络结构和算法特点进行研究，首先就实现神经网络非线性关系的激活函数出发，分析不同的类型的激活函数的错误隐蔽能力，实验结果表明具有双边抑制效果的激活函数容错能力比单边抑制函数要强。

另外，分析神经网络模型结构，可以得到的结论是神经网络的权值参数是用来提取相关输入数据的特征的，权值的绝对值越大，所提取的特征的占比就越大。因此，可以考虑在网络训练过程中在目标函数中添加一个惩罚量作为参数出错的容错学习模型，尽量减小数据的绝对值大小，每个参数都参与其中用于综合提取数据特征，从而减少单个参数出错对输出结果的影响，提高识别精度。

另外在实验过程中发现当错误出现在不同的网络层对于最终的识别结果的影响也不同，卷积层相较于全连接层更敏感。为了提高神经网络的容错性，在网络结构中添加归一化层，减少有网络参数出错导致下一层网络输入数据的位移影响，实验结果表明针对不同的激活函数，归一化层都能起到很好地容错作用，且依然满足双边抑制函数错误抑制能力比单边抑制函数要好的结论。

论文结构安排如下：

第 1 章主要论述了论文的背景与意义，当神经网络处理器作为星载芯片应用到卫星系统中时会受到太空辐照环境的影响，尤其是单粒子翻转错误。然后介绍了国内外目前神经网络以及神经网络处理器的研究现状，进一步介绍了目

前关于抗辐照技术的研究现状，从神经网络处理器自身的结构特点以及其实现的神经网络算法出发，引出利用神经网络自身的容错特点提高其抗单子里翻转的容错能力的重要性以及相对于传统抗辐照技术的优势。

第 2 章主要论述了神经网络算法的理论基础，介绍了相关神经网络模型结构、设计原理以及优化算法。其次介绍了神经网络处理器的架构，数据重用以及数据流，进一步分析了太空辐照对电子设备造成的故障类型，为后文的分析奠定理论基础。

第 3 章，从神经网络加速器的数据流分析，利用卷积神经网络的权值共享特性进行加速计算，得到分析权值参数出错随着数据流扩散对于最后结果影响的必要性。考虑太空辐照会造成的 SRAM 存储单元发生单粒子错误，以单个网络参数建立单粒子翻转概率模型，服务于论文实验中单个参数的比特翻转注错。接下来分析了不同类型的函数作为激活函数的错误隐蔽能力，从模型设计到注错实验分析对比，论证了双边抑制函数作为激活函数错误隐蔽能力最优，此时的神经网络模型的容错性能最优。进一步在网络训练过程中针对权值参数添加 L2 正则化惩罚项，分析论证了其可以提高神经网络的容错能力。

第 4 章，考虑到用于容错的 L2 正则化系数的选取的局限性，在网络模型中引入归一化层，从模型设计到实验参数调整以及与传统的神经网络模型、考虑了 L2 正则化的网络模型进行对比分析，验证了归一化算法可以提高网络的容错能力。进一步提出联合 L2 正则化和归一化来提高神经网络的容错能力。且分析了循环神经网络的容错特性，以及这类算法是否适用于提高 RNN 的容错能力。

第 5 章，总结了本文的研究内容，展望了神经网络容错未来的的研究方向。

本文的创新点在于：

（1） 从激活函数角度分析，得到双边抑制函数作为激活函数可以提高网络的容错能力。

（2） 从 L2 正则化和归一化角度提高神经网络的容错能力，不同于其他神经网络容错方法涉及到再学习的过程。

1.4 本章小结

本章叙述介绍了神经网络加速器作为星载设备应用的背景，结合神经网络的模型结构特点与算法特性，针对太空辐照环境造成的单粒子翻转错误，提出了提高神经网络容错能力的任务和目标。

进一步介绍了国内外关于神经网络、神经网络加速器以及抗辐照技术的研究现状，主要基于神经网络算法的应用场景以及神经网络加速器的应用发展进行分析，肯定了神经网络加速器未来作为星载设备应用到卫星系统中进行数据处理的必然性，进一步肯定了传统的抗辐照技术。但是对于专用的神经网络加速器其涉及到大量的数据计算，传统的抗辐照技术并不能满足神经网络处理器的要求。因此基于神经网络自身的结构特性与算法特性，从算法角度提高神经网络的容错能力的必要性。最后，本章就论文研究内容和结构进行了叙述，同时还对论文研究的创新点进行了说明。

第 2 章　神经网络以及神经网络加速器计算背景分析

2.1 人工神经网络介绍

人工神经网络是一种模仿人脑神经网络的非线性模型，可以用来对输入输出之间的未知关系进行建模，反复学习达到一个较好的预测结果。随着大数据时代的到来，人工神经网络在图像识别、图像分类、语音识别等领域取得了很好地效果。随着神经网络的迅速 ，对于硬件的运算能力也是一个很大的挑战。为了解决这一问题，采用可以模仿人脑神经网络结构的神经网络加速器（一种专用集成芯片），在进行图像识别等智能处理过程中相较于传统芯片很大的提高了工作效率。下面主要阐述深度神经网络的相关原理。

2.1.1 深度神经网络

在生物学上，神经元的神经冲动一般分为兴奋和抑制，一般情况下都是处于抑制的状态，但当处于抑制状态的神经元受到刺激，导致神经递质超出某个范围值后，就会处于兴奋状态，进一步向后面的神经元传递相关信息。模仿生物学的神经元概念，在深度学习中同样提出了神经元的概念，作为神经网络的基本单元。其基本结构如图 2.1 所示：

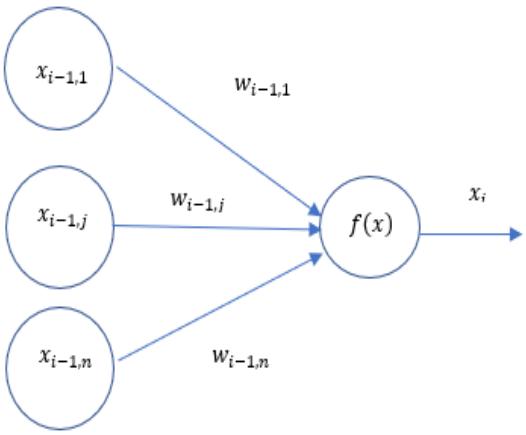


图 2.1　单个神经元

Fig2.1 a single neural unit

$$x_i = f\left(\sum_{j=1}^{n} w_{i-1,j} x_{i-1,j} + b_{i-1}\right) \quad (2.1)$$

其中，x_i表示第i个神经元的输出，$x_{i-1,j}$表示输入，n表示神经元i有n个输入，$w_{i-1,j}$表示输入x_{i-1}与神经元i连接的权值。网络模型的训练就是一个根据实际结果与理想结果之间的误差损失不断对网络权值进行更新的过程，等满足一定的条件训练结束。b_{i-1}表示神经元的偏置。f表示激活函数。

激活函数可以使得将模型从线性变成非线性，使得模型能够处理非线性的问题，提高神经网络模型的对高维非线性数据的处理能力，扩大模型的适用范围。

常用的激活函数有以下几种：

1)双曲正切函数（Tanh 函数）：

$$Tanh = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.2}$$

Tanh 函数将输出数据的数值控制在[-1,1]范围以内，通过控制数值的幅度，保证数据的幅度在网络中不会发生较大的变化，可以理解成双侧抑制。

2）修正线性单元[38]（Relu 函数）

$$Relu(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{2.3}$$

不同于 tanh 函数，relu 函数是一个分段线性函数。 Relu 函数扩大了数据激活状态的范围，模型训练时，使得重要的神经元处于激活状态，大大提高了整个神经网络的数据特征提取能力，同时将一些不必要的神经元的输出置为 0，减少了网络的计算量，提高了模型训练的效率。可以理解成单边抑制函数。

但 relu 函数也有自身的一些缺点，在模型训练的过程中当有较大的梯度经过 relu 函数之后，经过参数更新之后可能会进入到 relu 函数单边抑制的状态，该神经元对于所有的输入都保持相同输出。因此在 relu 函数的基础上，又有了很多新的激励函数被提出，例如泄漏修正线性单元（Leaky Relu）等。

3）高斯误差线性的单元[39]（GELU）

GELU 的非线性是通过随机的应用恒等或 0 来映射一个神经网络的输入的随机正则化的预期转换，通过将 dropout、 zoneout 和 Relu 三者的属性相结合来诱导激活函数。Relu 和 dropout 都可以生成一个神经元的输出，Relu 是确切的用 0 或者 1 乘以输入，负值乘 0，正值乘 1；dropout 是随机的用 0 乘以输入，zoneout 是随机的用 1 乘以输入。GELU 通过使用 0 或者 1 乘以输入合并这

三个函数的能力，但 0-1 的值的选择是依赖于输入值随机选取的，具体的实现是神经元的输入乘以一个伯努利分布 n~Bernoulli（∅(x)），其中$\emptyset(x) = P(X \leq x), X \sim N(0,1)$为高斯函数的累积分布函数。随机的非线性是根据输入$x$随机优化$\emptyset(x) \times I(x) + \big(1 - \emptyset(x)\big) \times 0x = x\emptyset(x)$，因此 GELU 被定义为：

$$GELU(x) = xP(X \leq x) = x\emptyset(x) \tag{2.4}$$

一般将 GELU 近似为$0.5x(1 + tanh\left[\sqrt{2/\Pi}\,(x + 0.044715x^3)\right])$。

神经网络就是由无数个神经元，假设一个深度神经网络有 L 层，z_i^l表示第 l（0≤ l ≤ L）层第i个神经元的输出激活值，其中 layer0 表示输入层，layerL 表示输出层，针对数据集$D = \{(x_1\ , y_1), \ldots, (x_n\ , y_n)\}$，其中 x 是输入，y 是预期输出，对输入层 layer0：

$$z^0 = x \tag{2.5}$$

隐含层 layerl（$1 \leq l \leq L - 1$）：

$$z_i^l = f(x_i^l) \tag{2.6}$$

其中$f(x)$是激活函数，x_i^l是第 l 层第 i 个隐含单元的预激活值，可表示为：

$$x_i^l = \sum_{i=1}^{n_{l-1}} w_{ji}^l z_i^{l-1} + b_j^l \tag{2.7}$$

n_{l-1}表示第 l-1 层的神经元个数，w_{ji}^l表示连接第 l-1 层的第i神经元和第l层第j个神经元的权值，b_j^l为偏置。

输出层 layerL：输出单元的个数等同于目标输出的维度，通常采用 softmax 激活函数。

$$z_i^L = \frac{e^{-x_i^L}}{\sum_i e^{-x_i^L}} \tag{2.8}$$

2.1.2 卷积神经网络（CNN）

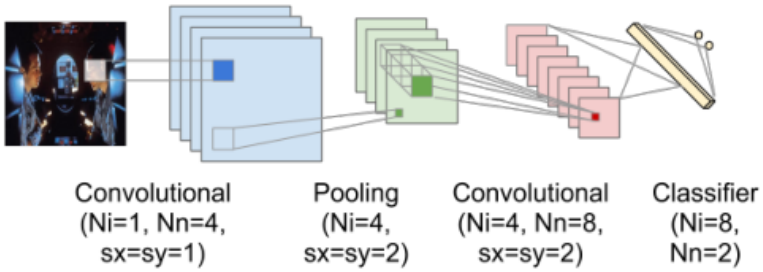


图 2.2 卷积神经网络结构图[11]

Fig2.2 a CNN structure

当要处理的是图像问题时，如果采用一般的神经网络，各个网络层之间以全连接的方式连接，针对于图片中每个像素点都要作为网络的输入来说，将会是很大的计算量，因此针对这类图像问题，人们提出了卷积神经网络，引入卷积核的概念进行关键特征的提取，提高运算效率。

传统的卷积神经网路主要包含两个模块，一是特征提取，二是特征分类。特征提取一般是由卷积层和池化层进行操作，而特征分类层通过将前面所提取到的信息分类整合，输出结果，一般是由全连接层进行操作。

卷积神经网络一般由以下几个网络层构成：

(1) 卷积层(Convolution layer)：卷积层是卷积神经网络中最重要的一层，其主要是对图片进行特征提取，具有权值共享和局部连接的特性。一般 CNN 包含多个卷积层，每个卷积层提取的特征不同，每层对于输入数据产生的一个更高层次的抽象被称为特征图，保留了输入数据重要且唯一的信息。卷积层的主要构成是卷积核。卷积核一般是个多维矩阵，有限的特征提取范围，同时针对卷积层的神经元与上一层的局部区域连接，卷积核的大小决定了局部区域的大小，另外局部区域的大小又被定义为感受野，表示与输入特征图接触的范围多少。感受野的值越大其对应的信息越趋向于全局，值越小则表示其提取的特征更倾向于局部和细节。卷积核除了对输入特征图进行特征提取还可以起到对高维数据降维的作用。

将通过卷积核提取得到的特征图按照一定的顺序进行组合就得到了卷积层的输出，这个输出由输入特征图的大小，卷积核的维度大小，滑动步长和填充

值决定。对于同一个输入特征图，不同的卷积核学习提取到的信息不同因此得到的输出特征图也不一样。一个卷积层的具体操作如图 2.3 所示：

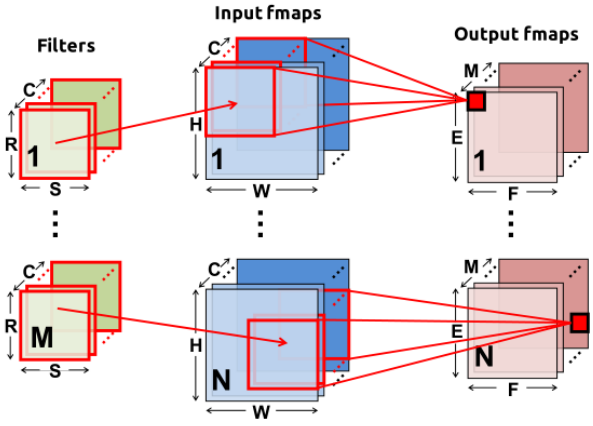


图 2.3 CNN 中的高维卷积[40]

Fig2.3 High-dimensional convolution in CNN

输入图像为 C 张$H \times W$的特征图，卷积核有 M 组，其中每组有 C 个$R \times S$卷积核，在图像问题处理中，一般 R=S。计算过程中，每一组的 C 个卷积核都分别要和 C 张输入特征图进行卷积操作，最后将结果求和得到一张输出特征图。整个过程的伪代码为：

表 2.1 卷积层操作伪代码

Table2.1 Pseudo Code for Convolution Layer

```
For n=1:N
    For m=1:M
        For e=1:E
            For f=1:F
                For h=1:H
                    For   w=1:W
                        For c=1:C
                            OUT[n][m][e][f]+=Input[n][h+se][w+sf][c]*
                                                kenerl[n][h][w][e][f]
```

实际的卷积操作中，卷积核相当于一个滑动窗口，通过步长移动来提取数据特征，但是卷积核的大小和输入特征图之间的维度很多时候不是恰好被整除的关系，因此需要对图像进行填充操作，同时也保证了下一个输入数据的有效维度，不过不进行填充数据降维之后不足以满足后续网络层进行下一步计算。

对于一个已知输入维度 W，卷积核大小为 r，步长为 s, 填充个数为 p, 输出维度为 F，则满足等式（2.9）：

$$F = \frac{W-r+p}{s} + 1 \tag{2.9}$$

(2)池化层（Pooling Layer）:池化层在卷积神经网络中另一个重要的存在，又被称为下采样层，通常有最大值池化和平均池化两种操作，依然具有非线性特性。如图 2.4 所示，展现的最大值池化的一个过程，获取采样窗口内的最大值，同理平均池化就是获得采样窗口的平均值。可以看到每个采样子区域没有重叠，即采样窗口滑动的步长等于采样窗口的维度，从而每个采样区域不会有重叠的情况发生。

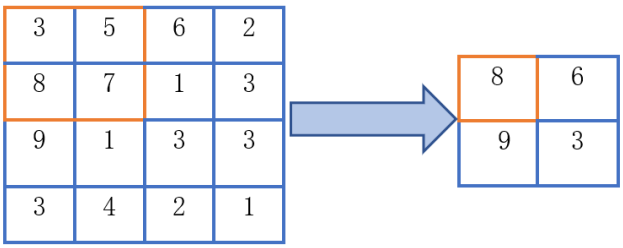


图 2.4　池化操作示意图

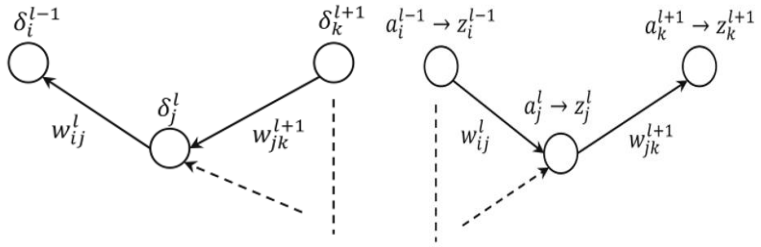
Fig2.4　Pooling operation

池化层主要用来采样降维，减少下一层输入特征图的大小，减少参与计算的参数量，同时还可以防止过拟合。

(3)全连接层：CNN 中全连接层一般在网络的最后用于特征分类。全连接层中的每个神经元与上一层的每个神经元都进行连接，用来整合卷积层和池化层提取到的特征信息，通常全连接层的最后一层会跟一个 softmax 函数用来计算每一个类的概率，从而达到分类的效果。

2.1.3 卷积神经网络的训练方法

神经网络的训练分为两个过程前向传播和反向传播，如图 2.5 所示：



（a）前向传播　　（b）反向传播

图 2.5 神经网络训练过程

Fig2.5　Neural network training process

较为经典的训练方法为 BP（误差反向传播）算法。一个神经网络模型可以被视作一个非线性函数，函数的输入是x，输出是y, w, b为参数，通过学习优化参数，神经网络从给定的输入中产生期望的输出y'，然后通过一个误差函数$J(W,b)$来实现目标优化，即将误差反向从输出层传递给输出层的过程中，利用梯度下降算法调整网络权值。常见的误差损失函数有平方差损失函数：

$$J(W,b;x,y)=\frac{1}{2}||y-y'||^2 \tag{2.10}$$

和交叉熵损失函数：

$$J(W,b;x,y)=-\sum_j y_j\, ln\left(y'_j\right)+(1-y_j)ln(1-y'_j) \tag{2.11}$$

下面以等式（2.10）作为损失函数来分析神经网络参数的训练更新：

逆向传播梯度一般定义为 δ_i^l，

对于输出层的权值W和偏置量b有：

$$W_{ij}^l=W_{ij}^l-\alpha\frac{\partial}{\partial W_{ij}^l}J(W,b) \tag{2.12}$$

$$b_i^l=b_i^l-\alpha\frac{\partial}{\partial b_i^l}J(W,b) \tag{2.13}$$

对于中间的隐含层，有

$$\delta_i^l=f\left(a_i^l\right)\sum_k w_{ik}\ \ \delta_k^{l+1} \tag{2.14}$$

综上，神经网络的反向传播算法主要步骤如下：

a)如图 2.5（a）所示通过前向传播计算得到所有神经元的激活值

b)输出层，计算偏导数的值

c)中间的隐含层，利用反向传播，计算偏导数的值

d)最后利用所得的偏导数的值进行参数更新

2.1.4 循环神经网络

在语音识别、文本识别等领域广泛采用的神经网络结构为循环神经网络（RNN），主要有两种形式：一种是结构型递归神经网络，一种是时间型递归神经网络。这些要处理的数据本身具有时序上的相关性，因此 RNN 实质上是将数据按照时序多次输入，每次的计算结果在传递给下一个单元的同时也将继续保留作为自己的输入。主要思想是在当前已有信息的基础上，关联过去时刻已经存在的信息，从而判断未来信息发展的趋势。相较于深度神经网络其在输出在时间上具有了“记忆”的功能，形成了一种循环的数据流，同时相较于卷积神经网路，可以在时间属性上挖掘数据之间的相关性，对于具有时间序列的高位非线性数据，采用循环神经网络结构处理相关问题通常可以达到很好地性能。

2.1.4.1 RNN

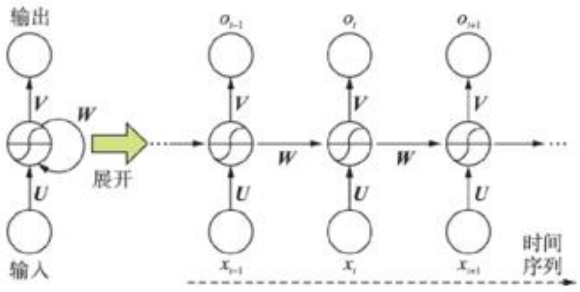


图 2.6　循环神经网络结构图

Fig2.6　Structure of R NN

如图 2.6 所示，一种经典的循环神经网络模型结构，将其展开之后可以看到随着时间序列每个时间步都有相同的结构。假设输入层、隐含层和输出层的神经元个数分别为m、n和q，模型中主要参数有U、V、W，则U、V、W分别表示输入层和隐含层，前后时间步隐含层和输出层与隐含层之间的权重关系矩阵。其中U的大小为（$m \times n$），W 的大小为（$m \times m$），V 的大小为（$m \times q$）。

2.1.4.2 LSTM

长短时记忆网络（LSTM）是时间递归神经网络的一种变形。和传统的循环

神经网络相比，LSTM 对隐含层做了更加复杂的设计。如图 2.7 所示，表示的是一个 LSTM 逻辑单元，有四个门的设计，分别是遗忘门、输入门、候选门和输出门。遗忘门控制着前面记忆中信息丢失了多少；输入门决定了t这个时间步的输入信息，即有多少信息是可以被用来做下一步处理的；候选门用来计算当前信息与前面的记忆信息的和；输出门则用来控制有多少信息是可以被用于下一个阶段。LSTM 通过这 4 个“门”来对每个时间步输入的信息做修改处理，从而达到将有效的保存长时间的记忆信息。在实际应用中，一般是将多个 LSTM 逻辑单元连接到一起形成一个 LSTM 网络模型。

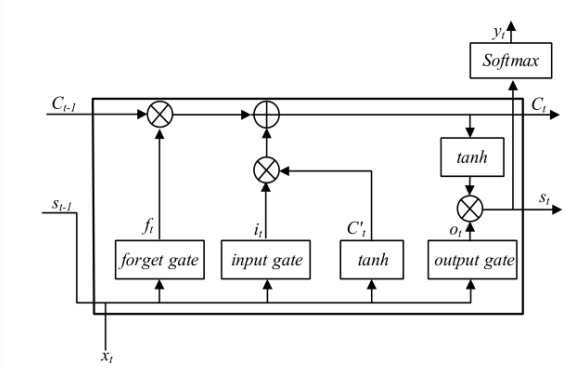


图 2.7 LSTM 隐藏层的逻辑设计结构

Fig2.7 Logic design structure of LSTM hidden layer

LSTM 网络模型的训练过程具体分为前向传播和后向传播两个部分：

1） 前向传播过程：

假设输入的时间序列为$X = （x_1, \ldots, x_t）$，隐含层序列为$S = （s_1, s_2, .., s_t）$,输出序列$Y = （y_1, y_2, \ldots, y_t）$，则计算过程如下：

$$f_t = sigmoid\left(W_f^T \times s_{t-1} + U_f^T \times x_t + b_f\right) \tag{2.15}$$

$$i_t = sigmoid\left(W_f^T \times s_{t-1} + U_f^T \times x_t + b_i\right) \tag{2.16}$$

$$C_t' = tanh\left(W_f^T \times s_{t-1} + U_f^T \times x_t + b_c\right) \tag{2.17}$$

$$C_t = f_t \times C_{t-1} + i_t \times C_t' \tag{2.18}$$

$$o_t = sigmoid(W_o^T \times s_{t-1} + U_o^T \times x_t + b_o) \tag{2.19}$$

$$s_t = o_t \times \tanh(C_t) \tag{2.20}$$

$$y_t = softmax(V^T \times s_t + c) \tag{2.21}$$

其中，f_t，i_t，C_t，s_t，y_t分别表示更新的遗忘门，输入门、候选门和输出门

以及t时间步的预测输出，W、U 表示权重矩阵，b表示偏置向量。

2）反向传播过程：

LSTM 结构中采用时间反向传播算法来进行网络训练，通过叠加每个 time step 的参数梯度来更新网络参数，目标函数采用交叉熵损失函数，计算每个 time step 预测输出层的交叉熵，在第t时间步时，损失函数的计算如下式：

$$J(y_t, y_t') = -\sum_j y_j \, ln\left(y'_j\right) + (1 - y_j) ln(1 - y'_j) \tag{2.22}$$

其中y_t表示实际输出，y_t'表示期望输出，j表示输出层的第j个神经元

迭代过程中，参数更新表达式如下：

$$\Gamma = \Gamma + \alpha \sum_t \frac{\partial L(y_t, y_t')}{\partial \Gamma}, \quad \Gamma = W, U, V, b, c \tag{2.23}$$

其中，α是学习率，时间步$t = 0,1,2,\ldots,T$.

2.2 神经网络加速器

随着要处理的数据量增多，神经网络模型结构越来越复杂，传统的处理器不足以满足计算要求，出现了许多主要用于网络推断的专用神经网络加速器，这些专用处理器的性能和功耗相较于传统的处理器提升了很多。

2.2.1 神经网络加速器概述

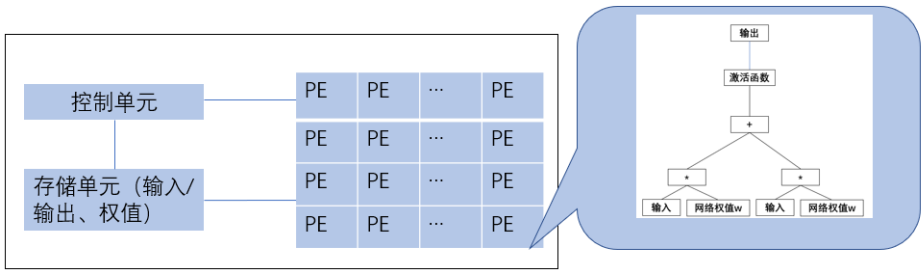


图 2.8 神经网络芯片架构

Fig 2.8 Architecture of Neural Network Chip

神经网络算法相较于传统算法，网络模型无论是在训练过程中还是网络推断过程中都需要进行大量的计算。当这类算法应用到实际问题处理中时，算法本身的复杂性和超大计算量，传统计算芯片并不能满足要求，于是实现神经网络算法的专用芯片被提出，也被称为神经网络加速器。通常在实际应用中为了追求实效性，神经网络加速器更应该集中于实现网络推断过程的部署与实现。

图 2.8 就是一个典型的神经网络加速器的架构，其主要组成部分包含控制单元、计算单元和存储单元。其中存储单元主要存储的参数有输入、输出和权值。神经网络模型在神经网络加速器上的部署通常是按照网络层的顺序一层一层的映射到加速器上的，不同层的输入输出以及权值参数都不相同，进行的乘法加法运算顺序也不相同，因此需要控制单元按照不同的逻辑顺序尽量调度。通常情况下在数据处理过程中，神经网络加速器是按照模型的网络层顺序来执行计算的，由于每个网络层输入数据的维度大小和用来进行该网络层特征提取的权值参数的维度大小不尽相同，而片上存储器的大小往往有限，因此有时候不能将一个网络层所有的数据都读取下来，需要将数据分成很多个块，进行分块计算。在分块计算时，为了减少片上存储器和片外存储器之间的数据通信次数，尽量减少数据读取上的时间延迟，从而涉及到了数据重用[40]。在数据重用过程中，对于中间数值往往存入一个缓冲区，等待该网络层所有的数据都计算完毕再将其传输到存储单元的 output buffer。因为数据重用的提出，在神经网络处理器的计算结构上会有不同的设计考量，一般分为时间架构和空间架构。主要区别在于计算架构中的处理单元(PE)之间能否相互进行数据通信。针对神经网络算法，更多的采用空间架构，因此空间架构的 PE 除了从数据缓冲区获取数据之外，其自身还具有本地存储器和控制逻辑，各个 PE 之间实现数据的通信。

寒武纪的“shidiannao”为例[14]，为了支持有效的数据重用，PE 网格上的 PE 之间进行数据传播，权值参数固定在 PE 单元里，输入数据和输出数据在 PE 网格之间流动，每个 PE 可以将本地存储的输入神经元发送到其左邻居和下邻居，通过在每个 PE 中的两个 FIFO（水平和垂直：FIFO-H 和 FIFO-V）来临时存储它接收的输入值来实现数据重用。对于每个处理单元的本地存储结构依然会有发生单粒子翻转的可能，其错误会随着处理单元之间的相互通信而扩散。

2.2.2 神经网络加速器的数据流加速方式

在神经网络加速器中，在提高计算效率和节约成本之间，通常会有个平衡，通常情况下实现乘法加法运算需要从内存读入输入数据，权值参数和部分和，同时要向内存写入更新后的部分和，因此结合内存访问和计算单元的并行结构很重要。

由表 2.1 卷积层的伪代码可以看出，神经网络的计算过程实际上就是一个多

个 for 循环嵌套的过程，而对神经网络进行加速其实也就是改变这个 for 循环的嵌套的顺序，达到数据复用率最高，减少片上存储器对于片外存储器的访问次数。

以卷积神经网络为例，它的卷积层和全连接层的基本组成部分都是乘法和加法操作，很容易实现并行操作，基于空间架构使用数据流处理，在 PE 之间形成一个处理链，实现数据的通信，更多的集中于如何在内存层次提高内存单元的数据重用，从而降低能耗和提高计算效率。因此数据流决定将什么数据读入到内存的具体位置以及何时处理这些数据，由于神经网络的处理过程不具有随机性，因此在设计神经网络加速器针对特定的神经网络结构可以拥有一个固定的数据流，来适应网络的形状、大小以及加速器的内存容量。

卷积神经网络的一个特点就是权值共享，在对于同一个输入特征图进行卷积操作时，不同的卷积窗口所用的卷积核是相同的，由于滑窗步长的关系，不同卷积窗口之间针对输入数据会有重叠，可以对输入数据进行复用，这样在进行下一个窗口数据读取的时候，只需要读取步长大小的新数据根据输入特征图进行拼接即可。

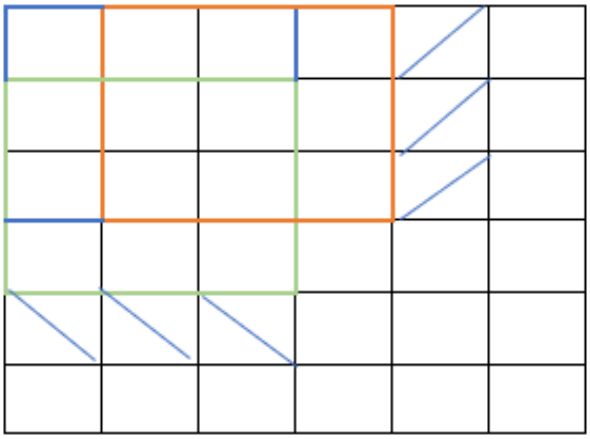


图 2. 9 权值共享并行计算示意图

Fig2.9 parameter sharing parallel

如图 2.9 所示，输入特征图的大小为 6*6，用来进行并行计算的卷积核大小为 3*3，步长为 1，卷积核在滑动窗口进行计算时，无论是横向移动还是纵向移动，输入数据都有 2*3 大小的重合部分，在权值共享的基础上，为了更好的利用已经 download 到片上内存的输入数据，卷积窗口可以同时右移和下移。

对于同一个特征图和同一个卷积核，在同一个网络层之间设计数据流，因

此在执行计算时，都是需要从芯片的外部 DRAM 读取相关的输入数据，再将对应的输出数据写入，随着网络模型结构越来越复杂，不同网络层之间的数据移动量会越来越大。文献[41]提出了一种跨卷积层的数据流，通过调整输入数据部署到加速器中的顺序来融合多个卷积层，根据相邻卷积层之间的映射关系来缓存和重用相关数据。如图 2.10 所示:

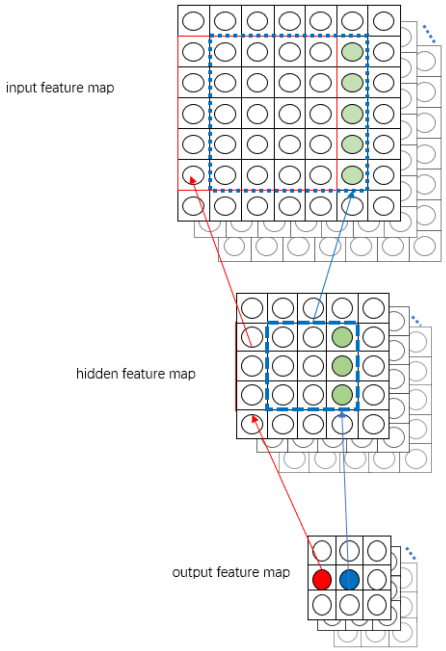


图 2. 10 融合两个卷积层的并行计算示意图

Fig2.10 fuse two layer parallel

从输出特征图出发，由输入特征图反向推理出其所对应的上一网络层的数据以及感受野，形成一个倒金字塔结构，最终确定金字塔底层输入的大小。一旦红色边框内的数据加载到了加速器的内存单元中就可以计算出整个倒金字塔的中间值，同时缓存这些中间值，从图 2.10 可以看到每个倒金字塔内部的数据流是相同的，相邻金字塔之间有部分中间值是可以用来计算两者之间的输出结果的。当达到金字塔的顶端时，只需要保存最后一层的输出像素点的值，图 2.10 中红点所示。当倒金字塔结构随着步长进行移动时，在金字塔的底层输入数据有重叠的部分，输入数据依然可以进行复用，只需要从片外加载一个步长大小的数据即可，设定一定的逻辑关系可以使得基础的金字塔结构右移和下移。

以 LeNet-5 网络为例，融合它的三个卷积层，考虑到要处理的图片的大小

为 28*28，通过数据填充将其扩充到 32*32，第三个卷积层的输出大小为 5*5，因此将 32*32 的输入分成 25 个块，步长设为 4，每一块的大小为 16*16，分块计算，最后根据每块的相对位置拼接数据得到结果。

2.3 太空辐照效应造成的故障损伤分析

空间环境复杂，神经网络加速器作为星载设备应用时，空间粒子造成的辐射和冲击会对其电路产生诸多的影响,其中最主要的影响为单粒子效应。单粒子效应使得电路发生故障，例如系统中的组合电路的某个节点受到粒子冲击会有电流电压的变化，这些都是这类故障发生的物理表现。

错误则是系统故障的一种表现，即与预期输出的偏差，其中元素的逻辑状态不同于其预期值[42]。将错误传播到系统级别会导致系统故障，但是，系统不一定会表现出来错误或故障，因为可能不会被捕捉到。可以按时间特征将故障分类如下：

（1）永久故障是连续的，并且随着时间的推移稳定；这主要是不可逆的物理损坏的结果。

（2）瞬态故障可能仅持续很短一段时间，通过复位或者是刷新可以恢复到正常状态，但是其依然可以对系统的预期结果造成影响。

进一步将这些故障抽象成数字错误，也就是其表现出来的数字（0、1）缺陷。

（1）永久故障：使得电路中某个数据或者是控制逻辑一直保持高电平位（stuck_falult_1）或者是低电平位（stuck_fault_0），因此又被称为固定故障模型。

（2）随机位翻转：数据或存储元素具有一些不正确但随机的值，即其比特位存储的数值受到外界环境的影响发生比特翻转，0 翻转成 1,1 翻转成 0。

固定故障模型仍然非常流行，因为已经证明晶体管和互连结构上的许多缺陷可以以合理的精度在逻辑级别建模为永久性故障[43]。但是这类故障模型并不能捕捉到随机位翻转的错误，也就是故障所引起的不确定状态。因此需要随机翻转模型来对于外部环境带来的存储单元的造成的类似于单粒子翻转错误的瞬态故障进行建模，只对数据发生损坏而不是对电路本身进行损坏。从概念上来

说，在空间环境中，由于粒子的撞击导致该存储单元保存了错误的逻辑值。

本文主要考虑这类随机错误带来的损伤影响。

2.4 神经网络容错特性

无论是适应硬件系统自身存在的缺陷，还是减少空间辐照环境造成的软错误的影响，都是目前关于神经网络加速器需要考虑的一个问题。因此需要有一种新的模式来利用神经网络加速器的优势，处理制造缺陷或者是瞬态故障，甚至可以将故障作为系统设计的一个固有部分，为神经网络的容错和可靠性提供一个新的解决思路。根据生物神经学的研究，人脑是可以忍受一些突触或者是神经元错误，甚至可以将这些错误当做计算学习的一个资源[44]，即神经网络其自身具有一定的容错特性。于是越来越多的研究开始利用神经网络自身的容错性能来改善环境或者是硬件系统对于加速器造成的影响。Atif Hashmi 等人的研究工作表明神经网络可以容忍 GPU 中的 stuck-at-fault 这类永久性错误[45]，Olivier Temam 提出了一种自定义的人工神经网络去适应晶体管水平的故障缺陷[46]。Xia 针对可以用来计算神经网络运算中的矩阵乘法运算的 RRAM 设备其本身具有的工艺上的缺陷，采用神经网络在线训练方法，通过神经网络反复去学习这些物理器件上的缺陷，从而达到容错的效果[47]。神经网络能够容忍这些故障的原因在于神经网络可以被重新训练。通过网络的前向传播将这些硬件损伤错误映射到神经网络模型中，在网络的再训练过程中学习这些错误分布，从而提高网络的容错能力。

这些算法都主要是针对 stuck-at-fault 永久性故障去提高网络的容错能力，因为这些错误抑制存在，可以通过网络传播的过程学习到，所以都涉及到的再训练。但是神经网络加速器作为星载设备进行应用时，首先并不存在再训练的条件，另外对于空间环境导致的单粒子翻转这类暂态的软错误不易捕捉，通过数据刷新这些存储单元的状态就可以恢复正常，虽然其对硬件不存在永久损害，但暂态的错误依然会映射到神经网络模型中，造成网络最后输出结果的精度损失。当神经网络加速器在执行网络推断过程时，其实就是对输入数据进行一个识别或者是分类。神经网络加速器的存储单元受到辐照影响发生单粒子效应时，造成网络参数出错是会影响到最后输出结果的准确率的，但是由于无法捕捉到这类错误，又无法确

定输入图像的类别，因此在错误无法捕捉，且不存在再学习的过程中，如何提高神经网络的容错能力的研究很有必要。

2.5 本章小结

本章首先叙述了神经网络加速器所涉及到相关神经网络的模型结构与算法，其次介绍了神经网络处理器的结构以及其计算单元的空间架构和数据重用模式以及数据流向。分析了太空辐照对于电子设备的故障影响，主要可以分为永久性故障和暂态故障，为后文主要分析存储单元的神经网络权值参数受到单粒子翻转效应的干扰，进一步进行网络容错性能分析提供了研究基础和指导作用。

第 3 章　卷积神经网络容错性分析

当芯片的存储单元发生单粒子翻转时，映射到神经网络的权值参数中，可以认为发生了权值扰动。神经网络的容错性能大致可以分为在线容错和通过再学习容错。在线容错指的是当出现错误的时候，不需要重新学习或采取任何纠正行动的能力，再学习容错指的是通过将错误元素执行的任务重新分配给好的元素来从错误中恢复能力。仅考虑 inference 阶段，不存在再学习过程，因此在线容错能力是非常有吸引力的，具有这种特性的网络即使出现了部分故障也会正常工作，只要不超过其容错能力，不需要诊断、重新学习或者是重新分配，可以避免故障检测和定位，因此通过错误模型对网络进行注错分析，并在网络用于推断之前，通过网络的容错学习尽量提高网络的容错能力对星载神经网络很有必要。

3.1　单粒子翻转概率模型

神经网络加速器主要包含存储单元、计算单元和控制单元。当有单粒子翻转效应发生时，本文主要考虑存储单元出错，芯片的存储单元中主要存储的数据是输入、输出和权值参数。

神经网络加速器应用卫星系统中做网络推断应用时，由于用于推断的输入数据的未知性，在研究太空辐照造成的存储单元出错时，分析网络权值参数出错更具有意义。另外考虑到片上的内存较小而网络参数较多，由 2.2 节对于神经网加速器中数据流的加速分析中，在数据流处理过程中会涉及到数据重用，且更多的并行计算都是基于 CNN 中卷积层权值共享的特点进行设计，用于复用的权值参数出错随着数据流的扩散会对最后的输出结果造成一定的影响。由于时间的累计效应用于复用的数据相较于其他数据更容易发生单粒子翻转错误，进一步可以肯定分析网络权值权值参数出错的意义。

当神经网络的权值参数出错时，可以肯定对神经网络最终的推断结果会有影响，在一定程度上准确度会下降。在对网络进行注错分析时，考虑到空间粒子对 SRAM 的辐射和冲击由于电荷共享的影响，会有多位翻转的情况发生。以单个参数为例，它有几个比特位发生翻转的概率并不相同，因此可以抽象出单粒子翻转

概率模型，更好地模拟参数出错。

芯片的存储单元 **SRAM** 的某个比特位单粒子翻转错误发生时，受到电荷共享的影响，其周围的比特位也会发生翻转，从而会出现连续的几个比特位都会发生翻转的现象。由泊松分布得到启发，当有多位发生翻转时，用复合泊松模型[48、49]。在复合泊松分布中，每个泊松事件都与一个随机变量 m 和一个分布函数 G(m)有关，给定 G(m)的分布之后，n 个独立同分布随机变量的和的分布函数可以表示为

$$F(m/n) = [G(m)]^{n*} \tag{3.1}$$

其中$[G(m)]^{n*}$是$G(m)$的 n 次卷积，n 是带有参数 λ 的泊松变量，λ 表示事件发生的平均概率，在本文表示平均翻转错误。对所有 n 的可能值求和得到：

$$F(m) = \sum_{n=0}^{\infty} \frac{\lambda^n e^{-\lambda}}{n!} ([G(m)]^{n^*}) \tag{3.2}$$

因此，$F(m)$是一个具有复合分布 G（m）的复合泊松分布，其中$G(m)$满足几何分布，其概率公式为：

$$p(m) = (1-r)r^{m-1} \qquad m=1, 2, 3, \cdots \tag{3.3}$$

其中 r 表示在前一个错误发生时下一个错误发生的概率。等式（3.2）代入等式（3-3）可以得到一个复合分布模型，泊松-几何分布：

$$p(m) = \sum_{n=1}^{m} \frac{\lambda^m e^{-\lambda}}{n!} \binom{m-1}{n-1} r^{m-n} (1-r)^n \tag{3.4}$$

其中$\binom{m-1}{n-1}$表示已知有 1 个错误发生的情况下的组合分布。

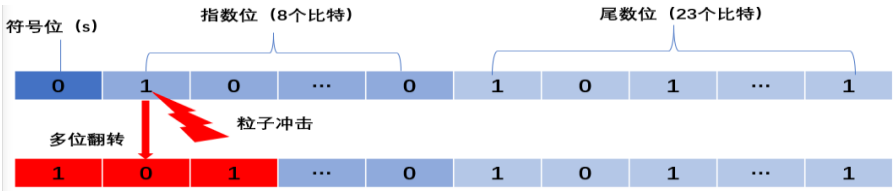


图 3.1 32 位浮点数

Fig 3.1 32-bit floating point

当将数值从十进制转换成二进制时，往往不会转的刚好相等，对于绝对值大于 1 的数不存在损失，但是对于绝对值小于 1 的数则会存在损失因此在将数据映射到硬件存储时，需要注意精度问题。

本文实验中参数为 32-bit 浮点数，如图 3.1 所示，首位 s 为符号位，中间 8 位(e)为指数位，剩下的 23 位(m)为尾数位，即一个浮点数可以表示为：

$$(-1)^{s} \times m \times 2^{(e-127)} \tag{3.5}$$

当发生单粒子翻转时，如果翻转是在尾数位，分析等式（3.1），可以看到对参数值影响不是很大。但如果是在符号位，则会使数据由正数变成负数，负数变成正数；同样如果翻转发生在指数位，可能会导致数据翻倍或缩小，最后的结果会导致该参数提取的某个特征被放大或是弱化。考虑到粒子会造成多比特翻转，因此某个参数在有某一个位发生比特翻转的情况下，这个参数的其他位有较大可能会发生翻转。

以 32 位浮点数为例，在已知有错误发生的情况下，某个参数内最少有一个比特发生翻转，最多有 32 个比特发生翻转。在已知错误发生的情况下，有m个比特发生翻转的概率：

$$p(m|m_1, m_2, \ldots, m_{32}) = \frac{p(m)}{\sum_{i=1}^{32} p(m_i)} \tag{3.6}$$

假设 λ=0.01，考虑到粒子会造成多比特翻转，因此某个参数在有某一个位发生比特翻转的情况下，这个参数的其他位有较大可能会发生翻转，取 r=0.57, 实验中采用的的参数为 32 位浮点数，在已知有错误发生的情况下，某个参数内最少有一个比特发生翻转，最多有 32 个比特发生翻转，得到概率图 3.2：

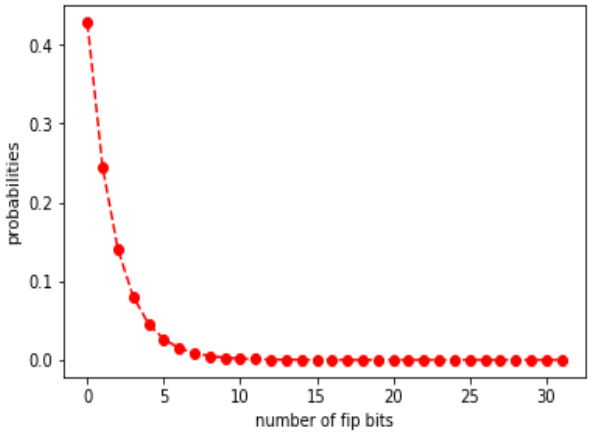


图 3.2　单个参数比特翻转概率

Fig3.2 the upset probability of single parameter

3.2 激活函数角度分析网络容错性

当考虑存储单元发生单粒子翻转，而对于计算单元中乘法器、加法器等是不易受辐照影响发生单粒子翻转的，其中激活函数的处理是放在计算单元中的，又激活函数$f(x)$的具有非线性特性，因此从激活函数角度研究其对错误的隐蔽能力很有必要。

3.2.1 激活函数容错能力分析

分析神经网络加速器的架构，主要包含控制单元，存储单元和计算单元。存储单元主要村存储的数据有输入数据，输出数据以及网络参数。计算单元主要实现乘法加法以及激活函数等功能。神经网络的每个节点接受输入值，并将输入值传递给下一层，在网络隐含层和输出层的输入输出之间有个函数关系，即为激活函数$f(x)$。以单个输出神经元为例：

$$x_i = \mathrm{f}\left(\sum_{j=1}^{n} w_{i-1,j}x_{i-1,j} + b_{i-1}\right) = f(W^T X + b) \tag{3.7}$$

粒子在对芯片存储单元的辐射和冲击造成的单粒子翻转作用映射到网络参数是随机的，任一网络层的权值参数都有可能发生错误。假设权值参数出现单粒子翻转错误对输出的影响等价为 $\hat{\theta} = \Delta W^T X$ ，则出错后的输出为：

$$\tilde{x_i} = \mathrm{f}\left(\sum_{j=1}^{n} w_{i-1,j}x_{i-1,j} + b_{i-1} + \hat{\theta}\right) = f(W^T X + b + \hat{\theta}) \tag{3.8}$$

令$\theta = \sum_{j=1}^{n} w_{i-1,j}x_{i-1,j} + b_{i-1}$ ，误差$error(\theta, \hat{\theta}) = \tilde{x_i} - x_i = f\left(\hat{\theta} + \theta\right) - f(\theta)$。

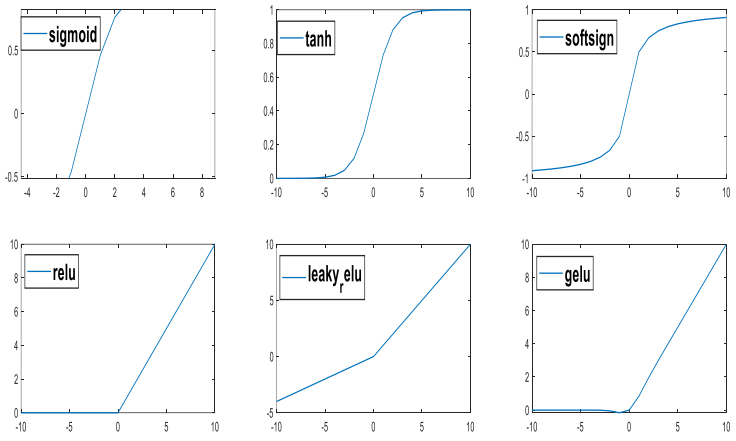


图 3.3 几种典型的激活函数

Fig3.3 Several typical activation functions

目前神经网络中采用的激活函数主要有具有双边抑制效果的 sigmoid、

softsign 和 tanh 函数等，具有单边抑制效果的 relu、leaky_relu 函数以及将 relu、zoneout 和 dropout 三者属性相结合的 GELU 函数。从图 3.3 不同激活函数的函数图像中可以看出双边抑制函数将函数值限制在一定的范围以内，保障数据的幅度在网络中不会有太大的变化，当网络权值出现错误对输出造成影响时，依然能够很好地隐蔽错误，拟合数据，较为准确地预测输出。对于单边抑制函数而言，虽然其使得神经网络的神经元具有了稀疏激活性，但是当出现的错误造成的误差影响较大时，相较于双边抑制函数，其错误隐蔽能力较弱。GELU 函数则是通过随机的将输入值乘以 1 或者 0 诱导激活函数的非线性，当刚好将出现错误的神经元置为 0 时，可以减少其出错对结果的影响，虽有一定的隐蔽错误能力，但过于随机性，如果置为 0 的神经元都未出错，反而放大了错误的影响效果。

以具有双边抑制效果的 tanh 函数和单边抑制效果的 relu 函数为例，图 3.4 和图 3.5 表示的就是，当函数的的输入出现局部错误时，函数输出的偏移情况。由于 relu 函数可以放大正数的有效激活，仅具有单边抑制的效果，相对于 tanh 函数，输入出现错误其错误隐蔽能力较弱。

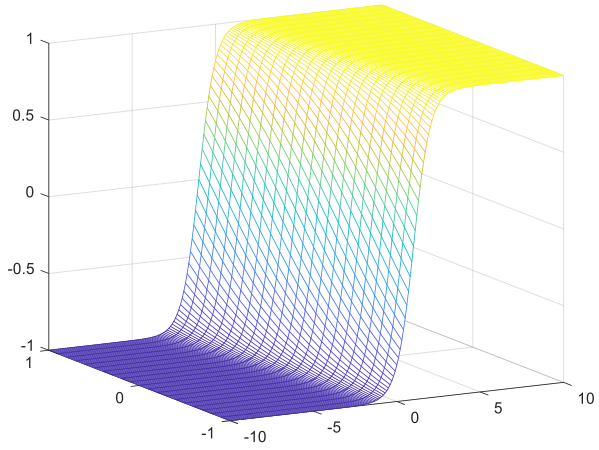


图 3.4　tanh 函数局部错误

Fig3.4 the local error of tanh

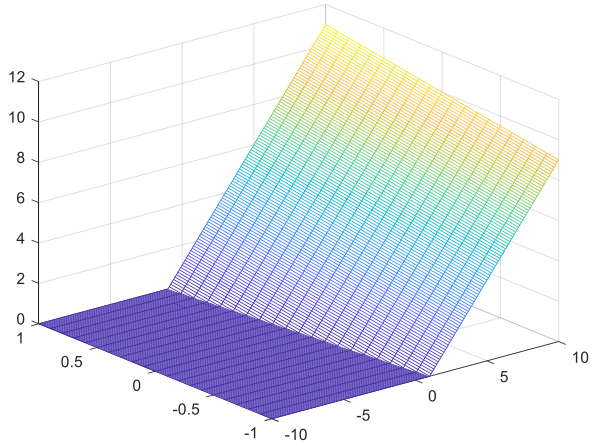


图 3.5　relu 函数局部错误

Fig 3.5 the local error of relu

3.2.2 实验过程与实验结果验证

1）数据集与模型结构选取

本文采用 LeNet 网络替代各卷积神经网络进行实验研究与分析，主要有以下两个方面的原因：一是为了更直接的展现单粒子翻转错误造成的网络参数出错对神经网络识别准确率的影响，同时也为了更加直接的展现各类激活函数的错误隐蔽能力，二是因为 LeNet 网络是最早成功应用于图像识别度网络，是其他经典深度神经网络的基础，且它们的主要工作原理相同，都是对二维数据的卷积、池化并在输出端采用全连接网络。

LeNet-5 网络结以及具体的网络层参数如表 3.1 所示:

表 3.1 LeNet-5 网络架构

Table3.1 Architecture of LeNet-5

网络层	卷积核	步长	图像填充（padding）	是否采用激活函数	可训练权值参数个数
Conv1	5*5	1	2	是	156
S1	2*2	2	0	否	0
Conv2	5*5	1	2	是	1516
S2	2*2	2	0	否	0
Conv3	5*5	-	0	否	48120
Fc1	-	-	0	否	10164
Output	-	-	0	Softmax	924

实验中采用的数据集是来自美国国家标准与技术研究所的 Mnist 手写数据集，训练集 60,000 张不同人手写的数字图像组成，测试集的大小则为 10,000 张。采用该数据集的原因有两部分，一是因为 LeNet-5 网络的提出就是为了解决手写数字的识别问题；二是因为 Mnist 手写数据集已经高度成熟，本文主要考虑神经网络权值参数出错对于整个网络推断阶段识别准确率的影响，因此数据集本身的噪声亦或是其他因素不做任何考虑，因此 Mnist 手写数据集提供了很好的输入数据的基础。

2) 实验过程

整体实验流程如图 3.6 所示，首先根据相关数据集的训练集来训练神经网络，考虑到不同的激活函数的错误掩蔽能力不同，在训练过程中采用不同的激活函数使网络训练结果达到最优，保存网络参数用于网络推断。结合网络参数错误概率模型对网络参数注错，根据输出结果分析激活函数函数的错误隐蔽能力。

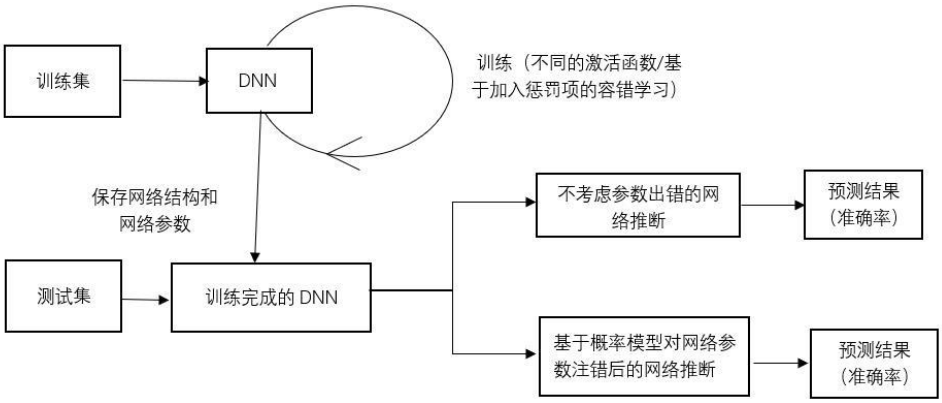


图 3.6　实验整体流程图

Fig 3.6 Overall flow chart of the experiments

采用 LeNet-5 网络迭代训练达到最优准确率98~99%，训练过程中参数优化算法为 SGD(随机梯度下降)，采用的损失函数为交叉熵损失函数。

LeNet-5 网络大约由 60000 个参数，参数用单精度浮点数表示。因此，如果粒子翻转发生是在尾数中，对于参数数值的影响非常小，如果粒子翻转发生在指数中，则影响非常大。影响的参数数量是根据网络中参数的总数确定的(如表 1 所示)。

3） 实验结果分析

分别采用不同的函数作为激活函数应用到 LeNet-5 网络模型中，由于单粒子翻转是小概率事件，影响的错误参数不可能会有很多，因此基于 0.01%和 0.1%占比对权值参数进行注错实验 100 次统计实验结果求平均准确率得到图 3.7，其中横轴表示 LeNet-5 网络中应用的不同激活函数，纵轴表示平均准确率。

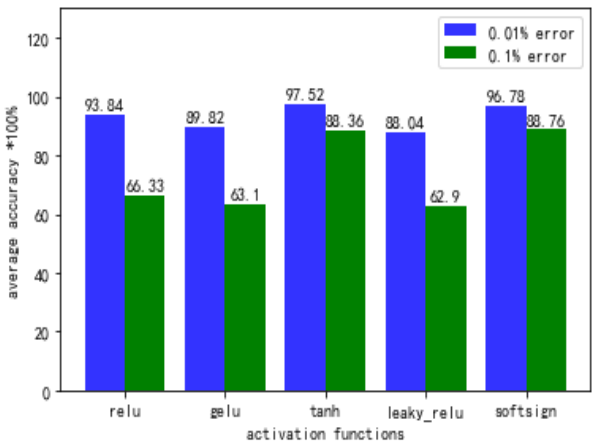


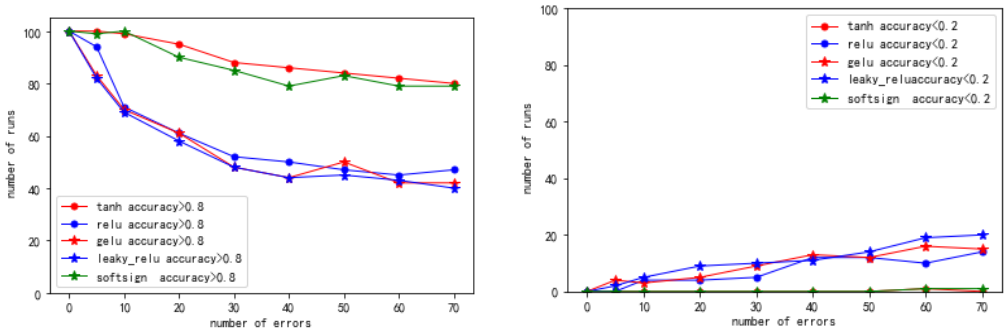
图 3.7　基于不同激活函数注错实验平均准确率统计

Fig3.7 Statistics of the average accuracy of error injection　experiments based on different activation functions

由图 3.7 可以看到当错误参数占比为 0.01%时，分析多次实验所得到的平均准确率结果，两者之间相差不是很大，但双边抑制函数作为激活函数的错误隐蔽能力要优于单边抑制激函数和 GELU 函数；但是当错误参数的占比为 0.1%时，双边抑制函数的隐蔽能力要明显优于其他函数，在网络中应用双边抑制函数多次实验所得的平均准确率可以高于其他函数 20%。

为了更好地分析权值参数出错下 LeNet-5 网络模型应用不同的函数作为激活函数的网络性能，实验中细化出错参数的占比，假设其错误参数的错误数量级

在10^{-4}~10^{-3}，基于10^{-4}~10^{-3}的错误数量级（也就是 5~70 个网络参数出错），随机选取参数进行注错，单个参数的注错依然基于 3.1 节提到的单粒子翻转概率模型进行翻转比特个数的采样。由于参数出错的随机性，不同的参数出错最后对于神经网络准确率的影响并不相同，因此本文进行 100 次注错实验，统计分析得到图 3.8 的实验结果，其中横轴表示参数的错误个数，纵轴表示准确率统计次数，即随着错误参数的增加，统计准确率高于 80%和低于 20%的比例变化。



(a)准确率大于 80%次数统计　　**（b）准确率低于 20%次数统计**

图 3.8　LeNet-5 不同的激活函数权值误差下的性能分析

Fig3.8 The performance of the Lenet model with different activate function under various weight errors.

从图 3.8 可以看到随着错误参数的增加，大于 80%准确率的占比在下降，低于 20%准确率的占比在增加，但是对于不同的激活函数，LeNet-5 网络表现出来的容错性能也不一样。由上文激活函数的数学特性分析可得，权值出错映射到激活函数的函数图像上其实就是该点的值在坐标轴上的位置平移，从而导致最终的输出结果出现偏差，但因为激活函数本身具有一定的错误隐蔽能力，可以减少这种影响。图 3.8(a)表示的是，随着错误参数占比的增加，网络结构中采用不同的激活函数其识别准确率大于 80%的次数统计，可以看到具有双边抑制效果的函数作为激活函数其大于 80%的准确率占比要明显高于其他激活函数。图 3.8(b)中则表示的为随着错误参数占比的增加低于 20%的准确率占比结果，可以看到采用具有双边抑制效果的激活函数的 LeNet-5 网络准确率低于 20%的次数几乎为 0，明显优于网络模型中采用其他的函数作为激活函数。而同时具有双边抑制效果的激活函数 tanh 和 softsign 函数，tanh 函数的错误隐蔽能力略优于 softsign 函数。

3.3 从正则化角度提高神经网络的容错能力

分析神经网络其训练的目的是为了获得最优的网络参数从而达到更接近理想输出的实际输出。假设神经网络的输入向量是x,输出向量是y,学习获得的权值连接矩阵为W,则一个神经网络可以表示为$y = x * W$。若理想输出是y'，神经网络不断学习调优的过程即可以表示为找到一个最优的W。但是当考虑到空间辐照环境引起的单粒子翻转软错误，进一步导致网络权值参数出错时，为了解决这类问题，有两种方法：一是结构稳定化，通过增加网络层的复杂性，达到即使有部分网络参数出现误差，依然可以有较好的准确率。但神经网络处理器用于网络推断时，其中有一个目的就是为了提高运算效率，往往实际应用的都是被压缩后的网络模型；二则是正则化。

本节主要从正则化分析，为了减少权值扰动对输出结果的影响，采用正则化操作，而正则化一般是增加一个惩罚量[50]来修正误差函数。

3.3.1 算法描述

针对神经网络，传统的梯度下降的算法的目的，是为了不断学习调优获得一个最优的网络参数以达到精确分类。对于神经网络参数而言，其进行训练的方式表示为：

$$\min ||y' - x \cdot W|| \tag{3.9}$$

其中y'表示期望输出。网络优化的目标就是为了最小化实际输出与期望输出之间的差距。

当神经网络加速器存储单元的权值参数受到单粒子翻转的影响发生错误时，此时参数将由原来的参数W变成W',同样需要满足

$$\min ||y' - x \cdot W'|| \tag{3.10}$$

分析神经网络特性，其网络参数的作用是用来进行特征提取和特征分类。当参数出现错误时，需要尽可能的减少出错参数对于整个网络系统的特征提取的影响。因此在网络训练过程中，针对目标函数添加一个惩罚约束，即正则化，减少网络参数出错带来的影响，即考虑尽量减少各个参数之间的关联性。具体算法流程如下所示：

算法： 基于权值参数 w 带正则项的容错学习算法

Require: learning rate α, regularization constant γ

epochs N, number of layers L

weight parameters W

loss:$\boldsymbol{J}$, total_loss: $\boldsymbol{J}+R$

1. random initialize parameter W
2. for n=1 to N do
3. for l=1 to L do
4. save the parameter w^l
5. end for
6. for l=L to 1 do
7. calculate derivate $\frac{\partial J}{\partial w^l}$, $\frac{\partial R}{\partial w^l}$
8. update parameters $W \leftarrow W - \alpha(\frac{\partial J}{\partial W} + \gamma \frac{\partial R}{\partial W})$
9. end for
10. end for

针对权值参数的惩罚项 R，一般有 L1 正则化

$$total\_loss = -\sum_j y_j ln y_j' + \gamma \sum |w| \quad (3.11)$$

和 L2 正则化

$$total\_loss = -\sum_j y_j ln y_j' + \gamma \sum |w|^2 \quad (3.12)$$

当考虑了 L1 正则化的数据取值更小，且零值参数占据的比例非常大，即摒弃了一些无效特征，不仅达到一个网络稀疏化的效果另外在保证一定的准确率的条件下可以提高网络推断的效率，但若考虑网络参数出现受到太空辐照的影响权值参数出现错误，尤其是 0 值参数发生翻转，这会影响到输入数据特征的有效提取。因此为了提高神经网络模型的容错能力，考虑在目标函数这种针对权值参数添加 L2 惩罚项。

3.3.2 实验过程与实验结果分析

基于等式(3.12)的目标优化函数，采用 tanh 函数作为激活函数，不断调整

正则化系数进行迭代训练，达到最优准确率，基于最优准确率保存最优参数用于网络推断。提取 LeNet-5 网络 conv1 的权值参数，其权值参数的一个分布分别如图 3.9、3.10 所示：

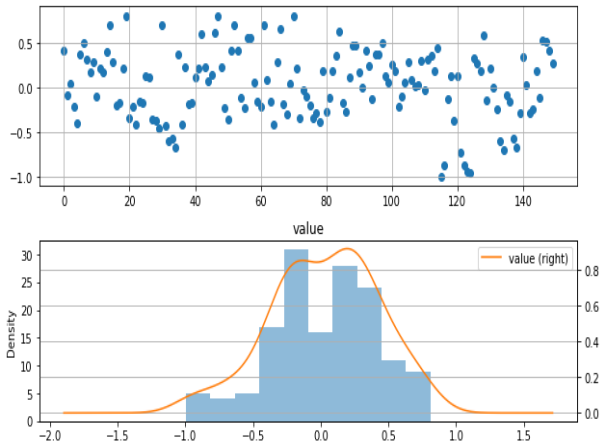


图 3.9　未考虑正则化的卷积层参数分布

Fig 3.9 Convolutional layer parameter distribution without considering regularization

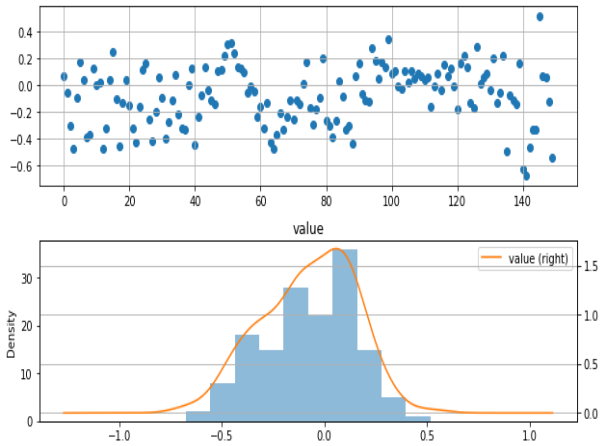


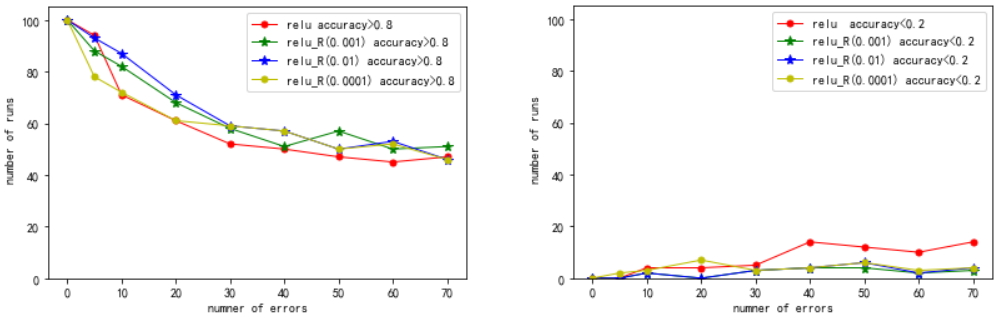
图 3.10　考虑 L2 正则化的卷积层参数分布

Fig3.10 Convolutional layer parameter distribution with considering L2 regularization

以卷积层参数为例,未考虑正则化，其数值在（-1,0.75）之间；考虑了 L2 正则化的数据，其数值范围在（-0.75，0.5）之间，以单个参数为例，考虑了正则化的之后的网络参数数值较小，当其出现错误时对整体的参数提取特征的影响较小，从而减少由于网络层的数据叠加造成对最终结果的准确度的影响，可以起到网络容错的能力。

以单边抑制函数 relu 为例，基于等式（3.12）也就是在目标函数中添加

L2 正则化对 LeNet-5 进行网络训练，在网络迭代训练达到最优准确率保存参数用于网络推断，基于不同的错误参数量级对网络权值参数注错，每个错误量级均实验一百次统计实验结果得到图 3.11。通过添加惩罚项来提高神经网络的容错能力，由于不同的正则化系数对于网络权值参数的最终表现出来的作用不同，为了更好地提高神经网模型针对权值参数发生单粒子翻转错误的容错能力，在实验过程中，基于网格化搜索，通过对权值参数注错分析不同的系数值下的网络性能，从而确定用于网络容错的正则化系数值。



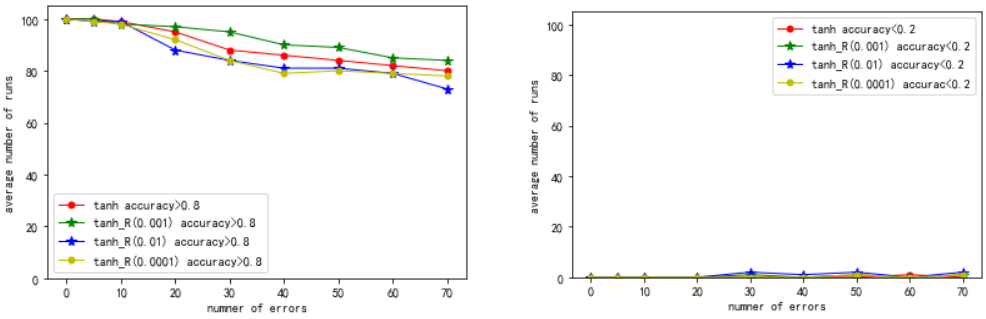
（a）准确率大于 80%次数统计　　（b）准确率低于 20%次数统计

图 3.11 LeNet-5（relu）是否考虑 L2 正则化权值错误特性分析

Fig 3.11 the performance of LeNet-5(relu) weight error with considering L2 regularization

从图 3.11 看到随着错误参数的增加，大于 80%准确率的占比在下降，低于 20%准确率的占比在增加，但是对于不同的正则化系数，LeNet-5 网络表现出来的容错性能不一样。图 3.11(a)表示的是，随着错误参数占比的增加，实验一百次大于 80%的准确率的次数，可以看到针对权值参数添加了 L2 惩罚的 LeNet-5 网络的容错性能优于未采用 L2 正则化的网络，图 3.11(b)中则表示的为随着错误参数占比的增加低于 20%的准确率占比结果，考虑了正则化的 LeNet-5 网络在权值参数出错后，实验 100 次低于 20%的准确率的次数要低于未考虑正则化的网络。综合分析可以得到针对 LeNet-5 网络模型中采用 relu 函数作为激活函数，当正则化系数 γ =0.01 时，LeNet-5 网络表现出来的容错性能最优。

同样以 tanh 函数作为激活函数，LeNet-5 网络模型基于等式（3.12）迭代训练达到最优准确率，基于最优准确率保存最优参数用于网络推断。根据上文的错误概率模型对参数进行注错，进行相同实验得到结果如图 3.12 所示。



（a）准确率大于 80%次数统计　　（b）准确率低于 20%次数统计

图 3.12 LeNet-5（tanh）是否考虑 L2 正则化权值错误特性分析

Fig3.12 the performance of LeNet-5（tanh）weight error with considering L2 regularization

综合图 3.12(a)和(b)可以看到随着错误参数的增加，但是对于不同的正则化系数，基于 tanh 函数作为激活函数的 LeNet-5 模型表现出来的容错性能亦不一样。但是通过正则化系数的不断调优，当正则化系数 γ =0.001 时，LeNet-5 对于权值参数出错表现出来的网络性能最优。

同样是针对 LeNet-5 网络模型，在模型训练过程中针对目标函数添加 L2 惩罚项提高网络的容错能力，为了分析激活函数是采用双边抑制的 tanh 函数还是单边抑制的 relu 函数，基于两者的最优正则化系数，统计进行注错实验 100 次中大于 80%的准确率的次数和低于 20%的准确率次数得到图 3.13，可以看到在 LeNet-5 网络模型中采用 tanh 函数作为激活函数的网络容错性能更好。

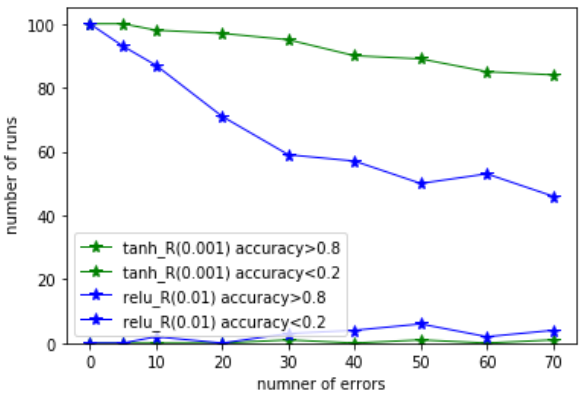


图 3.13　引入 L2 正则化, 不同激活函数的权值错误特性分析

Fig3.13 Considering L2 regularization, analysis of weight error characteristics of different activation functions

3.4 本章小结

分析神经网络权加速器中数据流加速方式，主要特点是基于权值共享来设计并行计算，当权值参数出错时，随着数据流的扩散对最后的输出造成一定的影响。本章主要基于卷积神经网络模型，当其作为星载神经网络应用时受到太空辐照的影响网络的权值参数会发生单粒子翻转错误从而影响整个网络的结果输出，造成输出准确率下降。首先考虑太空辐照对于芯片存储单元 SRAM 的影响建立单粒子翻转概率模型，在实验过程中对单个网络参数依据概率模型采样参数的比特翻转个数进行 0、1 翻转。进一步从激活函数出发，以激活函数的错误隐蔽能力作为选择目标，基于 LeNet-5 网络模型，引入不同的激活函数训练模型达到最优，在网络推断过程中基于不同的错误参数占比随机选取网络权值参数进行注错实验，分析实验结果并统计多次实验所得的平均准确率，可以得到具有双边抑制效果的非线性函数作为激活函数比具有单边抑制效果的非线性函数错误隐蔽能力更强，从而对应的卷积神经网络的容错能力更强。

考虑神经网络参数的作用是用来提取输入数据的特征，且网络参数具有较大的稀疏性，对网络的权值参数加以 L2 惩罚，以减小各个参数之间的关联性，达到网络容错的目的。通过分别对 relu 函数和 tanh 函数作为 LeNet-5 网络的激活函数进行注错实验分析，得到如若考虑网络权值参数出错，tanh 函数作为激活函数的网络容错能力更优。

对网络权值参数采用 L2 正则化来提高网络的容错能力，正则化系数相当于在网络训练过程中考虑了单粒子翻转对于网络权值参数的影响，需要考虑的问题是正则化系数的选择，通过注错实验验证来获取容错性能最优的正则化系数，虽采用单粒子翻转概率模型进行参数注错并且多次实验来对结果进行统计，但单粒子翻转依然是一个随机事件， 在最优正则化系数的选择上依然具有一定的局限性。

第 4 章 基于归一化提高网络容错能力

神经网络用于数据处理时，数据需要经过多个网络层的叠加。当网络层的权值参数出现错误会导致下一层的输入数据发生变化，进一步经过网络层的叠加，对高层网络层的输入分布会造成很大的影响，从而影响最终输出结果的准确率。在网络训练过程中在目标函数中添加针对权值参数的 L2 正则化的惩罚项可以提高网络的容错能力，但由于最优正则化参数选取的局限性，且实验过程中发现卷积层的容错能力依然低于全连接层。因此为了避免因为神经网络卷积层参数出错造成的影响，考虑对数据进行归一化处理来提高网络推断的准确率。

以神经网络的单个神经元为例：神经元接受一组输入向量的 $x=(x_1,x_2,\ldots,x_d)$, 通过某种运算后，得到一个输出$y=f(wx+b)$。本文考虑神经网络加速器存储单元受到太空辐照的影响有比特位发生单粒子翻转错误，又 2.2.2 节中对于神经网络加速器中数据流的分析，数据复用率最高的依然是权值参数，因此仅考虑其影响网络的权值参数，即网络权值参数出错后会导致输出数据的分布与未出错时的数据分布有一定的差别，对数据进行归一化操作可以减少出错带来的数据分布差别。

4.1 基于批归一化算法提高 CNN 的容错能力

4.1.1 算法描述

批归一化（Batch Normalization）是针对单个神经元进行，在神经网络训练过程中就一个 batch size 的数据来计算该神经元的均值和方差。BN 独立的规范化每一个输入维度x，规范化的参数是一个 batch size 的一阶统计量和二阶统计量。在整个神经网络模型中引入 BN 层，对网络层之间的数据进行归一化处理以此来抑制网络推断阶段卷积层网络参数出错对最终识别结果的准确率的影响。

当对神经元数据进行了规范化处理，将其限制在一个取值范围内，这样可以减少某一网络层参数出错造成的所提取的数据特征发生偏移带来的影响。但还需要考虑的问题时该措施可能会弱化未进行规范化时的数据特征，为了减少

规范化带来的数据特征变化，在模型训练过程中采用变换重构技术，引入可学习的参数 γ、β

$$y^{(k)} = \gamma^{(k)}\hat{x}^{(k)} + \beta^{(k)} \tag{4.1}$$

每一个神经元x^k都会有一对这样的参数γ、β

这样其实当：

$$\gamma^{(k)} = \sqrt{Var[x^k]} \tag{4.2}$$

$$\beta^{(k)} = E[x^k] \tag{4.3}$$

是可以恢复出原始的某一层所学到的特征分布。

BN 层的前向传播公式为：

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \tag{4.4}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2 \tag{4.5}$$

$$\widehat{x_i} \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \tag{4.6}$$

$$y_i \leftarrow \gamma\widehat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{4.7}$$

上面的公式中m指的是 batch size

当模型训练达到最优保存网络参数用于网络推断时，BN 层计算的网络参数亦固定不变，计算公式如下：

$$E[x] \leftarrow E_B[\mu_B] \tag{4.8}$$

$$Var[x] \leftarrow \frac{m}{m-1}E_B[\sigma_B^2] \tag{4.9}$$

即在均值由计算所有 batch 的u的均值得到，标准差所有 batch 得到的σ_B的无偏估计。因此在网络推断过程中 BN 层的计算公式为：

$$y = \frac{\gamma}{\sqrt{Var[x]+\varepsilon}} \cdot x + (\beta - \frac{\gamma E[x]}{\sqrt{Var[x]+\varepsilon}}) \tag{4.10}$$

4.1.2 实验过程与实验结果分析

当单粒子翻转造成的参数错误集中在同一网络层时，由于每一层提取的特征和实现的的功能不同对最终的结果影响也不同。以 tanh、relu 和 gelu 函数为例，分别针对这三种函数作为激活函数进行网络权值参数注错分析得到表 4.1，可以看出全连接层相较于卷积层，它的网络容错能力更强。

表 4.1 针对不同网络层错误参数为 30 的准确率分析

Table 4.1 Accuracy analysis for 30 error parameters of different network layer

LeNet5	准确率高于 80%占比			准确率低于 20%占比		
网络层	relu	tanh	GELU	relu	tanh	GELU
Conv1	0	50%	0	50%	0	95%
Conv2	3%	75%	45%	0	0	25%
Conv3	95%	100%	100%	0	0	0
Fc1	80%	100%	100%	0	0	0
Fc2	35%	75%	75%	5%	5%	0

以单边抑制函数 relu 和双边抑制函数 tanh 为例，在卷积层后加入 BN 层，网络迭代训练达到最优准确率保存参数用于网络推断，基于不同的错误参数量级对网络权值参数注错，每个错误量级均实验一百次统计实验结果。由于 BN 层所学习到的参数与 batch size 的大小有关，实验过程中通过对 batch size 设置不同的值进行对比分析，最终确定 batch size 大小为 64 时，其容错效果最佳。

在 LeNet-5 网络模型中采用 relu 函数作为激活函数，当网络权值参数出错时，其并不能表现出很好地容错能力。由表 4.1 的数据分析得到，卷积层参数出错对于最后的输出结果的影响明显高于全连接层的参数出错，当在卷积层后加上 BN 层后，网络模型经过训练迭代依然可以达到 99%的准确率，进一步对网络参数进行注错实验得到图 4.1，可以发现，实验 100 次无论是大于 80%的准确率次数还是低于 20%的准确率次数，相较于未在卷积层后引入 BN 层的 LeNet-5 网络都有了明显的改善，验证了在 LeNet-5 网络模型中引入 BN 层可以提高网络容错能力。

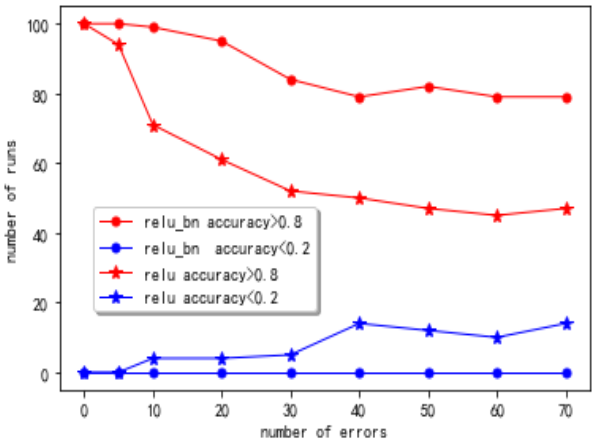


图 4.1　LeNet-5（relu）是否考虑 BN 权值错误特性分析

Fig4.1 LeNet-5 (relu) whether consider BN weight error characteristics analysis

在 LeNet 网络模型中采用 tanh 函数作为激活函数,由 3.2 节分析得到，其相较于 relu 函数作为激活函数，错误隐蔽能力要优于 relu 函数。但是同样不能避免的是卷积层参数出错对于最后的输出结果的影响高于全连接层的参数出错，当在卷积层后加上 BN 层后，网络模型经过训练迭代依然可以达到 98%-99%的准确率，进一步对网络参数进行注错统计实验结果得到图 4.2，可以发现，实验 100 次低于 20%的准确率的次数为 0，而高于 80%的准确率的占比均大于 80 次，对于双边抑制的 tanh 函数作为激活函数，BN 层的引入依旧可以提高网络的容错能力。

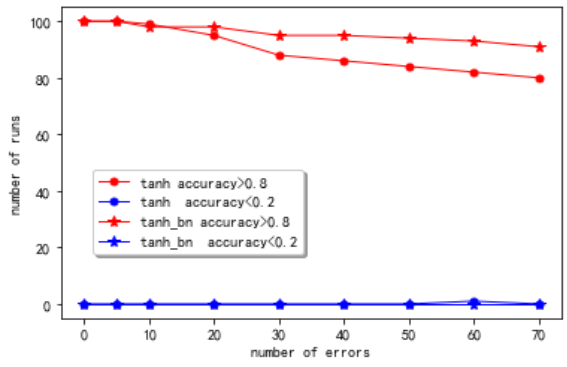


图 4.2　LeNet-5（tanh）是否考虑 BN 权值错误特性分析

Fig 4.2 LeNet-5 (tanh) whether consider BN weight error characteristics analysis

比较 relu 函数和 tanh 函数作为激活函数，在卷积层后引入 BN 层，LeNet-5 网络的容错性能得到图 4.3,统计分析上述实验结果得到表 4.2。对于 relu 函

数，引入 BN 层后，同样对网络权值参数进行注错分析，其容错能力明显提高，尤其是当错误参数的占比增大时，平均准确率相较于未引入 BN 层提高了 20%；对于 LeNet-5 网络模型中采用 tanh 函数本身就具有很好地错误隐蔽能力，但加入 BN 层之后，容错效果同样得到 3%-5%的提升。但在引入 BN 层后，比较激活函数分别是 tanh 函数和 relu 函数的实验结果，表 4.2 可以看到，虽然在错误参数占比较小时，两者之间的差别不是很大，但是当错误参数的占比增加时，不仅大于 80%的准确率的占比 tanh 函数要高于 relu 函数，多次实验得到的平均准确率亦优于 relu 函数作为激活函数。因此依然可以得到 LeNet-5 网络模型中采用具有双边抑制效果的函数作为激活函数网络容错能力更优。

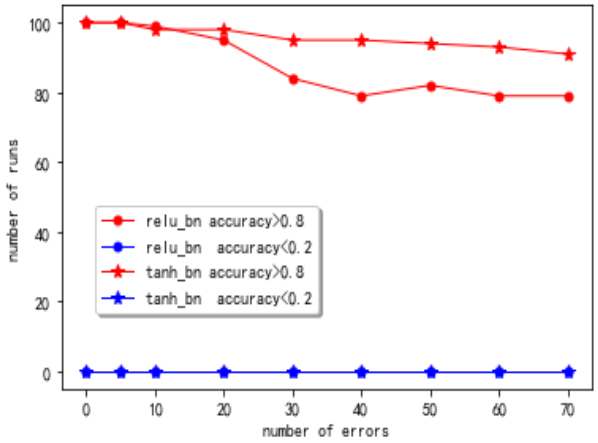


图 4.3 考虑 BN, 不同激活函数的权值错误特性分析

Fig 4.3 Considering BN, analysis of weight error characteristics of different activation functions

表 4.2　平均准确率统计结果

Table 4.2 the average accuracy

参数出错个数	(tanh)卷积后不加入 BN 层实验结果平均准确率	(tanh)卷积层后加入 BN 层后实验结果平均准确率	(relu)卷积后不加入 BN 层实验结果平均准确率	(relu)卷积层后加入 BN 层后实验结果平均准确率
10	96.83%	96.89%	90.21%	97.01%
20	94.96%	96.61%	85.64%	94.80%
30	92.31%	95.68%	76.16%	91.97%
40	89.47%	93.59%	68.05%	87.20%
50	88.60%	91.48%	64.57%	89.94%
60	87.43%	91.53%	63.86%	89.87%
70	86.97%	91.00%	63.20%	89.37%

4.2 归一化联合 L2 正则化提高 CNN 容错能力

由空间环境造成的单粒子翻转会导致神经网络的权值参数出错，并且会通过网络层之间的计算叠加而对网络中最后的输出结果造成影响，更有可能会导致神经网络的准确率精度下降。由 3.3 节正则化算法可以提高 CNN 的容错能力，4.1 节在 CNN 网络中的卷积层后添加 BN 层亦可以提高网络的容错能力，因此考虑到将两种算法联合到一起，批归一化联合 L2 正则化方法提高 CNN 的容错能力。依然不涉及到重新训练。

4.2.1 算法描述

采用批归一化联合 L2 正则化方法来达到神经网络容错能力的算法流程主要包含三个部分：神经网络模型结构搭建、模型训练和模型测试。

1）　模型结构

依然选取经典卷积神经网络 LeNet-5 作为实验模型，考虑到卷积层参数出错对最后识别准确率的影响要大于全连接层，很有可能会出现识别准确率低于 20%的情况，因此在卷积层后考虑添加批归一化(BN)层来提高 CNN 的容错能力，

新的容错 LeNet-5 模型结构如图 4.4 所示。

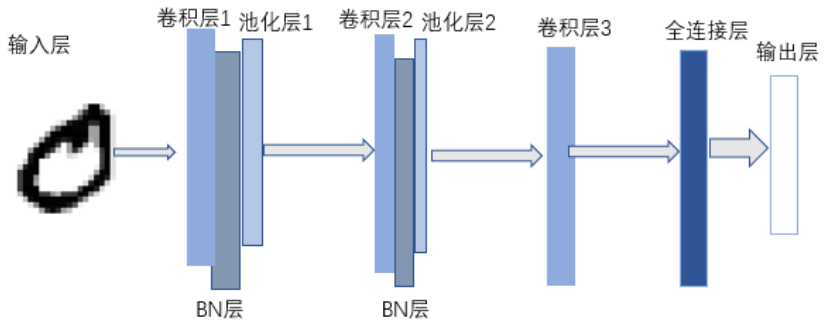


图 4.4　添加了 BN 层的 LeNet-5 网络结构

Fig 4.4　the structure of LeNet-5 with BN layer

2）　模型训练

数据集依然选择 Mnist 手写数据集，在网络模型中引入 BN 层后，不同于非正则化目标函数仅依赖于神经元输出的权值，具有 L2 惩罚项的神经网络的目标函数则是一个非正则目标与一个正则项的组合。采用 BN 和 L2 正则化算法的神经网络目标函数为：

$$J(w) = \sum_{i=1}^{N} l_i(y(X;w,\gamma,\beta) + \gamma||w||^2 \qquad (4.11)$$

其中l_i表示的是第i个输出单元的损失，单个输出单元的损失依然采用交叉熵计算。

3）　模型推断

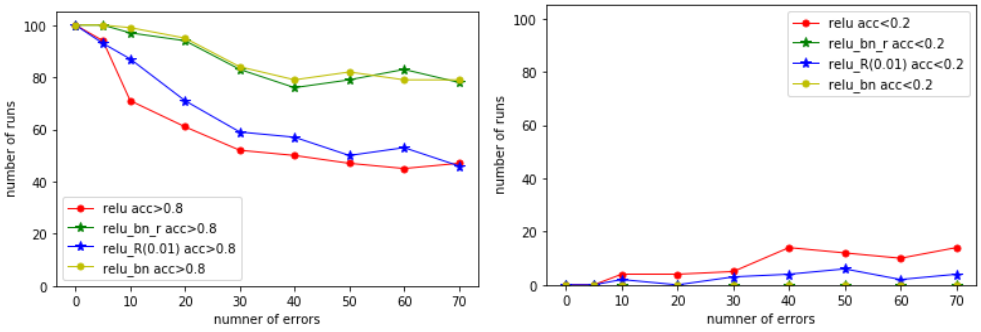
依据 mnist 手写数据集的测试集，对 2）中训练得到的最优网络参数，输入到网络模型中进行推断，并通过对网络的权值参数进行注错来分析模型的容错能力。

4.2.2 实验过程与实验结果分析

1） 实验过程

依然采用 LeNet-5 网络结构进行实验，基于目标函数（4.1）进行网络训练达到最优，保存最优模型下的网络参数用于网络推断过程。为了检验 BN 联合 L2 正则化算法对于 CNN 权值参数出错有一个相对更好地容错能力，同样进行注错实验分析。基于 10-4~10-3 量级的错误参数占比，对网络的权值参数注错，单个参数的注错依旧是根据 3.1 节提到的复合泊松模型进行比特翻转，多次实

验统计实验结果得到图 4.5 和图 4.6。



（a）准确率大于 80%次数统计　　（b）准确率小于 20%次数统计

图 4.5　不同算法的 LeNet-5（relu）权值出错准确率大于 80%的次数统计

Fig 4.5 Comparison of error performance of LeNet-5 (relu) weight parameters of different algorithms

从图 4.5 可以看出，当 relu 函数作为激活函数的 LeNet-5 网络模型进行，基于 L2 正则化算法、卷积层后添加 BN 层以及基于 BN 层联合 L2 正则化，当网络权值参数出错，网络所表现出来的性能并不相同。归一化算法优于在目标函数中添加 L2 惩罚项，采用了归一化算法的 LeNet-5 网络在（10^{-4}~10^{-3}）量级范围的参数出错之后不会出现准确率低于 20%的情况，网络的稳定性提高。但是由图 4.5 不能很好的判别出归一化算法和基于归一化联合 L2 正则化算法的性能好坏，进一步统计不同算法下 100 次实验结果的平均准确率得到表 4.3.

表 4.3 不同算法 LeNet-5(relu)实验 100 次平均准确率

Table 4.3 The average accuracy of 100 experiments of different algorithms LeNet-5 (relu)

参数出错个数	(relu)卷积后不加入 BN 层实验结果平均准确率	(relu)卷积层后加入 BN 层后实验结果平均准确率	(relu)考虑 L2 正则化实验结果平均准确率	(relu)卷积层后加入 BN 层联合 L2 正则化实验结果平均准确率
10	90.21%	97.01%	93.80%	97.45%
20	85.64%	94.80%	88.05%	96.53%
30	76.16%	91.97%	80.53%	93.96%
40	68.05%	87.20%	74.01%	89.56%
50	64.57%	89.94%	67.00%	87.23%
60	63.86%	89.87%	69.36%	89.74%
70	63.20%	89.37%	67.60%	89.63%

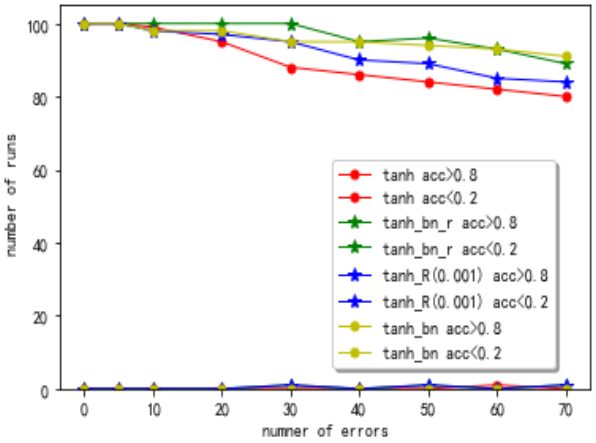


图 4.6 不同算法的 LeNet-5（tanh）权值参数出错性能比较

Fig 4.6 Comparison of error performance of LeNet-5 (tanh) weight parameters of different algorithms

从图 4.6 可以看出，tanh 函数作为激活函数的 LeNet-5 网络模型，基于 L2 正则化算法、卷积层后添加 BN 层以及基于 BN 层联合 L2 正则化，当网络权值参数出错，网络所表现出来的性能也是不一样的。实验一百次统计大于 80%的准确率的次数和低于 20%的准确率次数，采用 tanh 函数作为激活函数其准确率

低于 20%的次数几乎为 0，而高于 80%的准确率次数统计不同算法之间的差距并不是很大，但还是可以看出归一化算法优于在目标函数中添加 L2 惩罚项。为了进一步精确不同算法在参数出错情况下的的准确率值，统计不同算法下 100 次实验结果的平均准确率得到表 4.4。

表 4.4　不同算法 LeNet-5(tanh)实验 100 次平均准确率

Table 4.4 The average accuracy of 100 experiments of different algorithms LeNet-5 (tanh)

参数出错个数	(tanh)卷积后不加入 BN 层实验结果平均准确率	(tanh)卷积层后加入 BN 层后实验结果平均准确率	(tanh)考虑 L2 正则化实验结果平均准确率	(tanh)卷积层后加入 BN 层联合 L2 正则化实验结果平均准确率
10	96.83%	96.89%	97.23%	97.48%
20	94.96%	96.61%	96.30%	97.13%
30	92.31%	95.68%	94.22%	96.07%
40	89.47%	93.59%	91.29%	95.35%
50	88.60%	91.48%	89.83%	94.13%
60	87.43%	91.53%	89.50%	94.62%
70	86.97%	91.00%	89.41%	94.11%

从表 4.3、表 4.4 可以看到在 LeNet-5 网络中，无论是单边抑制的 relu 函数还是双边抑制的 tanh 函数作为激活函数，相较于单一的在网络模型中添加 BN 层和模型训练针对目标函数添加一个 L2 正则化的惩罚项，联合 BN 层和 L2 正则化的方法网络容错能力更优。同单一的在网络模型中添加 BN 层比较，联合 L2 正则化依旧可以使网络权重变得更小，减少各个参数之间的关联性，虽然没有使得网络更加稀疏减少网络计算量，但是在网络容错方面有很好的效果。相较于单一的针对目标函数添加一个惩罚项，在网络模型中添加 BN 层的优越性更为明显，当网络参数出错时，尤其是在卷积神经网络的卷积层出现错误，会使得卷积层提取到的数据特征发生偏移，如果对其进行归一化操作，可以减少这种偏移在网络叠加过程中造成的影响，不仅可以提高网络的鲁棒性，网络的

容错能力也能够得到提升。

在 LeNet-5 网络模型中，在网络模型中添加了 BN 层和模型训练针对目标函数添加了一个 L2 正则化的惩罚项后，分析网络模型中分别采用 tanh 函数和 relu 函数作为激活函数，tanh 函数的错误隐蔽能力是否依然优于 relu 函数。

针对两种不同的 LeNet-5 模型进行注错实验分析，同样统计 100 次实验结果中大于 80%的准确率次数和低于 20%的准确率次数得到图 4.7。由图 4.7 可以看到低于 20%的准确率次数都为 0，两者差别不是很大，但大于 80%的准确率的次数统计里，tanh 函数依然是优于 relu 函数的。分析表 4-3、表 4-4 的第四列的平均准确率数值，不同量级的参数出错，tanh 函数作为激活函数的 LeNet-5 网络模型的准确率精度依然优于 relu 函数作为激活函数。因此在对于权值参数出现错误扰动后网络表现出来的性能进行分析后，神经网络模型作为星载神经网络应用到卫星系统中，权值参数出现单粒子翻转错误扰动时，双边抑制效果的函数作为激活函数，网络模型表现出来的错误隐蔽能力更好。

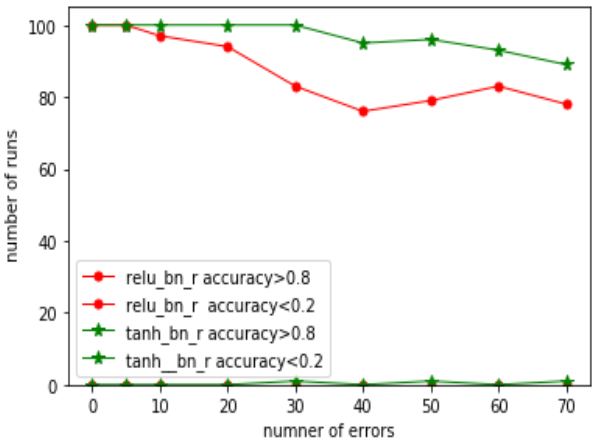


图 4.7　不同激活函数的 LeNet-5 权值参数出错分析

Fig4.7 Error Analysis of LeNet-5 Weight Parameters of Different Activation Functions

4.3 循环神经网络

前面所有的分析都是基于卷积神经网络进行讨论，本节主要分析循环神经网络的容错特性，分析循环神经网络模型的结构特点，它的权值在网络推断过程中会被反复使用，在硬件系统中同理，参数在被 download 到片上内存中数据复用率会很高，单粒子翻转造成的参数出错会在时序上不断的扩散，进而过结果产生影响。进一步分析归一化和 L2 正则化算法对于提高循环神经网络的

容错能力是否依然适用。

4.3.1 算法流程

对于循环神经网络，分析采用归一化方法是否可以提高该网络的容错能力的算法流程主要包含三个部分：数据预处理、模型训练和模型测试。

1）数据预处理

本实验依旧采用 mnist 数据集，并对其进行归一化处理，处理方式为：

$$x' = \frac{x - x_{min}}{x - x_{max}} \tag{4.12}$$

其中x_{max}和x_{min}分别表示 mnist 手写图像中像素值的最大值和最小值，x'为归一化以后的数据，然后将数据按照时序建模。

2）模型训练

采用 LSTM 网络的循环神经网络结构，考虑到 LSTM 网络层的结构和输入数据的特点，确定输入数据的维度为 28，隐含层节点数为 64，确定好相关参数之后，基于时间反向传播算法来更新网络结构的参数，得到一个最优模型。

当将归一化方法应用到 LSTM 结构的 RNN 网络中，不同于卷积神经网络的 batch normalization,这里采用的是 layer normalization ,即针对一个层的所有神经元求均值和标准差。对于循环神经网络即在某一个 time step 上分别计算归一化统计量。

对于时间步为 t 的节点，其输入a^t是 t-1 时间步的隐层状态h^{t-1}和 t 时刻的输入数据x_t，则有：

$$a^t = Wh^{t-1} + Ux^t \tag{4.13}$$

求归一化有：

$$u^t = \frac{1}{H}\sum_i^H a_i^t \tag{4.14}$$

$$\sigma^t = \sqrt{\frac{1}{H}\sum_i^H (a_i^t - u^t)^2} \tag{4.15}$$

$$h^t = f\left(\frac{g(a^t - u^t)}{\sqrt{(\sigma^t)^2}} + b\right) \tag{4.16}$$

其中g、b是需要学习的增益和偏置参数，f则表示为激活函数。

3）模型测试

依据 mnist 手写数据集的测试集，对 2）中训练得到的最优网络参数，输

入到网络模型中进行推断。

4.3.2 实验过程与实验结果分析

首先分析 LSTM 结构的循环神经网络的网络容错能力，计算实验所用的循环神经网络的参数共有约 25000 个，基于 0.1%，1%的量级对网络权值参数注错，实验一百次统计实验结果的准确率得到表 4.5。

表 4.5 LSTM 网络不同算法的容错能力分析

Table 4.5 Analysis of fault tolerance of different algorithms in LSTM network

	LSTM	LSTM+L2	LSTM+LN
0% error	96.95%	85.35%	97.85%
0.1% error	96.80%	85.31%	97.74%
1% error	96.03%	83.66%	96.92%

由表 4.5 可以看到基于 LSTM 结构的循环神经网络，首先由于其激活函数采用的是具有双边抑制效果的 sigmoid 函数和 tanh 函数，本身的错误隐蔽能力就很好。而且相较于卷积神经网络中尤其是卷积层的稀疏连接，RNN 可以展开成一个隐含层共享参数的 MLP，它的层数是由输入数据的时间片的数量决定。神经元之间的连接方式是全连接，所以当有很大占比的参数出现错误时，网络自身有一个很好的容错能力。但因为 RNN 网络自身的结构特性，当在目标函数中添加 L2 正则化来提高网络的容错能力时，反而放大了噪声扰动带来的影响，训练得到的一个最优网络模型反而性能变差。但是对每个时间步的数据进行归一化处理之后不仅可以提高网络的测试准确率，在网络参数出现错误时，进行注错实验分析得到的平均准确率亦优于未采用 LN 的网络结构。

所以当要处理的问题涉及到循环神经网络时，依旧可以考虑归一化提高 RNN 的稳定性和容错能力。

4.4 本章小结

本章首先就卷积神经网络在分析不同的激活函数的容错能力时，发现卷积层参数出错比全连接层参数出错对于卷积层神经网络最后输出结果的准确率的

影响要大，基于 LeNet-5 网络模型进行实验分析，在卷积层后添加 BN 层，对卷积层的输出数据进行一个归一化操作，无论是单边抑制的 relu 函数还是双边抑制的 tanh 函数作为激活函数，都可以提高网络的容错能力，且 tanh 函数亦优于 relu 函数。

联系到在网络训练的目标函数中针对网络权值参数添加一个 L2 正则化的惩罚项同样可以提高神经网络的容错能力，因此考虑将两种算法进行联合，经过实验验证，基于归一化联合 L2 正则化算法可以减少权值参数出错对于神经网络最后输出准确率的性能影响，提高神经网络的容错能力。

进一步分析这些算法对与循环神经网络结构是否适用，基于 LSTM 结构的循环神经网络，分析不同算法下的网络容错能力，得到归一化算法依然可以提高 RNN 网络模型的鲁棒性和容错能力。

第 5 章　结论与展望

5.1 结论

本文考虑到神经网络处理器在轨应用会受到太空辐照的影响，首先基于太空辐照对芯片存储单元造成的单粒子翻转错误影响，假设网络参数出错，基于参数的比特位建立空间上的错误概率模型：复合泊松分布；接着考虑到网络层的输入和输出之间存在函数关系-激活函数，从激活函数的数学特性分析，不同特性的激活函数其容错能力不同，基于理论和实验分析验证得到具有双边抑制效应的激活函数错误隐蔽能力更好。

另外，为了更好的提高网络容错能力，基于权值噪声容错模型，在网络训练过程中针对网络参数的 L2 正则化添加一个惩罚项以寻找最优模型和容错模型之间的平衡。当数据用神经网络来处理时，数据经过多个网络层的叠加，当网络参数出现单粒子翻转错误时会导致该网络层的输出数据发生变化，通过层层叠加从而对高层网络的输出有较大的影响，从而导致最终输出结果的准确率存在一定的误差。尤其是卷积层的参数出错时对于整个网络最后的输出结果的影响很大，因此考虑在卷积层后添加一个 BN 层来提高网络的容错能力。进一步将两优化算法结合到一起亦可以提高 CNN 的容错能力。针对卷积神经网络，上述算法的基本思想都是尽可能减少参数出错导致的出错后的数据和出错前的数据的分布差别，达到抑制错误的效果。

进一步分析了循环神经网络的容错性能，由于其不同于卷积神经网络的结构特点，实验得到正则化算法对于 RNN 模型没有很好地效果，但是归一化算法依然可以提高 RNN 的容错能力。

5.2 展望

通过本文对于神经网络容错性能的研究，认为后续的工作可以围绕以下几个方面继续深入研究：

1、本文在分析卷积神经网络的容错特性时，仅针对 LeNet-5 网络进行了分析，因为其结构经典，是后续复杂卷积神经网络研究的基础。因此，后续的工作

可以添加更加复杂神经网络例如 Vgg16,Vgg19,Alexnet 等进行验证分析。

2、考虑到提高神经网络加速器的运算速度和减少网络推理时间，一般部署到神经网络加速器上的模型结构都是在复杂的神经网络模型结构上进一步进行模型压缩得到的。因此下一步工作可以在不同的压缩算法在保证网络测试效果的基础上，就如何提高压缩后的网络模型的容错能力做进一步研究。

3、本文从神经网络加速器的数据流加速方式分析，尤其是利用权重值共享特点设计的并行计算结构，随着权值参数的复用在时间的累计效应上其受到太空辐照的影响发生单粒子翻转的可能性更大，参数错误进一步随着数据流扩散会对神经网络最后的输出结果造成影响。在实际的应用中，将神经网络部署到神经网络加速器时，为了提高神经网络的加速器的处理速度，减少加速器片上存储单元和片外存储单元之间的数据通信次数，会依据权值共享、数据重用设计网络的数据流，但 pytorch 等相关深度学习框架中的数据并不能直接部署到神经网络加速器中，需要依照数据流重新排列部署。而单粒子翻转又是一个随机过程，可以发生在神经网络加速器的任一过程中。因此后续的工作可以结合神经网络加速器内部的数据流，在进行注错分析实验时，虽然创造单粒子翻转发生的条件代价太大，只能对关键特性进行局部验证，且验证次数有限，但依然可以采用软件模拟的方式来将错误注入到加速器的片上存储单元和计算单元的本地存储中，更加贴近神经网络加速器行为应用层面的表现。

参考文献

[1] A.Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.

[2] A. Graves, A. r. Mohamed, and G. Hinton, “Speech recognition with deep recurrent neural networks,”[C]// 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2013, pp. 6645–6649.

[3] S. C. Hauser, W. C. Headley, and A. J. Michaels, “Signal detection effects on deep neural networks utilizing raw IQ for modulation classification,”[C]//MILCOM 2017 - 2017 IEEE Military Communications Conference (MILCOM), Oct. 2017, pp. 121–127.

[4] CESAR TORRES-HUITZIL, BERNARD GIRAU. Fault and Error Tolerance in Neural Networks: A Review.[J].IEEE Access(Volume:5)August 2017 pp.17322-17341 10.1109/ACCESS.2017.2742698

[5] Le Cun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[6] Le Cun Y., Boser B.E., Denker J.S., Henderson D., Howard R.E., Hubbard W.E., Jackel, L.D. Handwritten digit recognition with a back-propagation network[C]//Advances in neural information processing systems. 1990: 396-404.

[7] 王利伟,朱晓丹, 王建, 刘宇辰.人工智能在电子侦察中的应用分析[J].航天电子对抗，2018 年第 2 期

[8] 曾锐,陈锻生.结合双深度学习特征的高光谱遥感图像分类[J].小型微型计算机系统,39(2):396-400.

[9] He Kaiming,Zhang Xinangyu, et al.Deep residual learning for image recognition[C]//Proc of CVPR 2016.Piscataway,NJ:IEEE,2016:770-778

[10] Deng, J. Dong, W., Socher, R., Li, L.-J.,Li, K., Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Conference on Computer Vision and Pattern Recognition (CVPR) (2009).IEEE, 248–255.

[11] Chen, T., Du, Z., Sun, N., Wang, J., Wu, C.,Chen, Y., Temam, O. Diannao: A small-footprint high-throughput acceleratornfor ubiquitous machine-learning.In International Conference on

Architectural Support for Programming Languages and Operating Systems (ASPLOS), (March 2014). ACM 49(4):269–284.

[12] Chen, Y., Luo, T., Liu, S., Zhang, S.,He, L., Wang, J., Li, L., Chen, T., Xu, Z.,Sun, N., Temam, O. Dadiannao: A machine-learning supercomputer. In ACM/IEEE International Symposium on Microarchitecture (MICRO)(December 2014). IEEE ComputerSociety, 609–622

[13] Daofu Liu, Tianshi Chen, Shaoli Liu, Jinhong Zhou, Shengyuan Zhou, Olivier Temam, Xiaobing Feng, Xuehai Zhou, and Yunji Chen, "PuDianNao: A Polyvalent Machine Learning Accelerator", in Proceedings of the 20th ACM International Conference on 2015

[14] Du, Z., Fasthuber, R., Chen, T., Ienne, P.,Li, L., Luo, T., Feng, X., Chen, Y.,Temam, O. Shidiannao: Shifting visionprocessing closer to the sensor.In Proceedings of the 42nd ACM/IEEE International Symposium onComputer Architecture (ISCA'15)(2015). ACM, 92–104

[15] Sophie Duzellier. "Radiation effects on electronic devices in space. Aerospace Science and Technology Volume 9, Issue 1, January 2005, Pages 93-99".

[16] 袁建国.星载交换设备可靠性关键技术研究[D].国防科技大学，2018

[17] 陈毅华. 场效应晶体管单粒子效应的机制与加固方法研究[D].湘潭大学，2016

[18] WALLNERJ.GORDERM. Single event upset protection circuit and method: US9292378［P］2016.03.22

[19] CARONP，INGUIMBERTC，ARTOLAL，et al. Physical mechanisms inducing electron single-event upset[J].IEEE Transactions on Nuclear Science,2018,65 (8) : 1759－1767.

[20] JIN Y，HUAN Y X，CHU H M，et al. TMR group coding method for optimized SEU and MBU tolerant memory design[C]//Proceedings of International Symposium on Circuits and Systems. Florence，Italy,2018: 1－5

[21] 王波，王佳等. 星载 ASIC 芯片单粒子效应检测及在轨翻转率预估[J].半导体检测与设备, DOI: 10.13290 /j.cnki.bdtjs.2019.09.013

[22] 刘永杰. SOI FinFET 器件与组合逻辑电路单粒子效应研究[D].西安：西安电子科技大学，2015

[23] 吕挺. FinFET SRAM 辐射效应仿真建模研究[D].西安电子科技大学,2015

[24] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[25] Wang T, Wu D J, Coates A, Andrew Y. Ng. End-to-end text recognition with convolutional neural networks[C]. Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012). IEEE, 2012: 33043308

[26] Han S, Mao H, Dally W J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding[J]. arXiv preprint arXiv:1510.00149, 2015.

[27] Inadola F N,Han S,Moskewicz M W,et al.SqueezeNet:AlexNet-level accuracy with 50x fewer parameters and < 0.5MB model size[J] .arXiv:1602.07360,2016CARONP，INGUIMBERTC，ARTOLAL， et al. Physical mechanisms inducing electron single-event upset[J].IEEE Transactions on Nuclear Science,2018,65 (8) : 1759－1767.

[28] Courbariaux M., Hubara I., Soudry D., El-Yaniv R., & Bengio Y. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1[J]. arXiv preprint arXiv:1602.02830, 2016.

[29] DiCecco R., Lacey G., Vasiljevic J., Chow P., Taylor G., Areibi S. Caffeinated FPGAs: FPGA framework for convolutional neural networks[C]. 2016 International Conference on Field-Programmable Technology (FPT). IEEE, 2016: 265-268.

[30] 杨旭, 范煜川, 范宝峡. 龙芯 X 微处理器抗辐照加固设计[J]中国科学: 信息科学 2015 年 第 45 卷 第 4 期: 501–512.

[31] 郝宁,罗家俊 et al. SOI 工艺抗辐照 SRAM 型 FPGA 设计与实现[J]宇航学报 2018 年 9 月 第 39 卷 第 9 期

[32] Cesar Torres-Huitzil, Bernard Girau. Fault tolerance in neural networks: neural design and hardware implementation.

[33] Minjae Lee, Kyuyeon Hwang, and Wonyong Sung, “Fault tolerance analysis of digital feed-forward deep neural network,” [C]//2014 IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP)

[34] A. Assoum, N. E. Radi, R. Velazco, F. Elie, and R. Ecoffet, “Robustness against SEU of an artificial neural network space application,” IEEE Transactions on Nuclear Science, vol. 43, no. 3, pp. 973–978, Jun. 1996.

[35] Austin P. Arechiga and Alan J. Michaels. The Robustness of Modern Deep Learning Architectures against Single Event Upset Errors[C]// 2018 IEEE High Performance extreme Computing Conference (HPEC).DOI: 10.1109/HPEC.2018.8547532 .

[36] S. Kwon, K. Lee, Y. Kim, K. Kim, C. Lee, and W. W. Ro, “Measuring error-tolerance in SRAM architecture on hardware accelerated neural network,” in 2016 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia), Oct. 2016, pp. 1–4.

[37] A. P. Arechiga and A. J. Michaels, The effect of weight errors on neural networks[C]// 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Jan. 2018, pp. 190–196.

[38] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines[C]//In International Conference on Machine Learning, 2010.

[39] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units[J]. CoRR, abs/1606.08415.

[40] Yu-Hsin Chen , Joel Emer and Vivienne Sze .Eyeriss: A Spatial Architecture for Energy-Efficient Dataflow for Convolutional Neural Networks [C]// 2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture

[41] G. Buja and R. Menis, “Dependability and functional safety: Applications in industrial electronics systems,” IEEE Industrial Electronics Magazine, vol. 6, no. 3, pp. 4–12, Sept 2012.

[42] M. Alwani, H. Chen, M. Ferdman, and P. Milder, “Fused-layer CNN accelerators,” in MICRO, 2016.

[43] G. Buja and R. Menis, “Dependability and functional safety: Applications in industrial electronics systems,” IEEE Industrial Electronics Magazine, vol. 6, no. 3, pp. 4–12, Sept 2012.

[44] W. Maass, “Noise as a resource for computation and learning in networks of spiking neurons,” Proceedings of the IEEE, vol. 102, no. 5, pp. 860–880, May 2014.

[45] Hashmi A., Berry H., Temam O., Lipasti M. Automatic abstraction and fault tolerance in cortical microachitectures[C]//ACM SIGARCH computer architecture news. ACM, 2011, 39(3): 1-10

[46] Temam O. A defect-tolerant accelerator for emerging high-performance applications[C]// ACM SIGARCH Computer Architecture News. IEEE Computer Society, 2012, 40(3): 356-367.

[47] XIA L, LIU M, NING X, et al. Fault-tolerant training with on-line fault detection for RRAM-based neural computing systems[C]// Proceedings of the 54th Annual Design Automation Conference 2017. ACM. [S.l.]: [s.n.], 2017: 33.

[48] Barraza, B. Cernuschi-Frias, and F. Cernuschi, “A probabilistic model for grouped events analysis,” in Proc. IEEE Int. Conf. Systems, Man, Cybernetics, Oct. 1995, vol. 4, pp. 3386–3390.

[49] Sanghyeon Baeg, ShiJie Wen, and Richard Wong. SRAM Interleaving Distance Selection With a Soft Error Failure Model[J].IEEE TRANSACTIONS ON NUCLEAR SCIENCE, VOL. 56, NO. 4, AUGUST 2009

[50] Bishop C.M., Training with noise is equivalent to Tikhnov regularization[J].Neural Computation, Vol.7, 108-116, 1995

致 谢

三年的时间眨眼间就过去了，转眼就来到了毕业，回首这三年的读研时光，有过迷惘彷徨，但更多收获的是成长与感恩。借此机会，我要向所有关心支持过我的老师、朋友致以最真挚的感谢。

首先，要感谢平日里悉心指导我的导师们。首先要感谢我的导师梁旭文研究员，科研上，梁老师渊博的学识，严谨的态度给我产生了深远的影响。同时还要感谢谢卓辰副研究员，在研究方向的确定以及研究思路的拓展等方面，谢老师都给我了很大的帮助，实验的优化和论文的修改与完善，谢老师事无巨细样样躬亲。不过日常里，我们都亲切的喊他师兄，但在心里，他永远是老师般值得我去尊敬。感谢两位老师的教导和关心，永远感激，铭记在心。

其次，我要感谢我的同学们。感谢研一在中科大的那一年，很开心可以和本科同学段晨、高长生、陈志还有蒋涛继续共处一年的时间，这一年理论课的学习，感谢他们给予了我很大帮助。尤其要感谢孙宇豪和孔鑫玮两位小伙伴，从 2019 年的 9 月 5 号认识的起，除去假期的时间，几乎每天都和他们待在一起。三年的形影不离，谢谢他们给我带来的快乐和积极的人生态度。

最后，我要感谢我的家人，在硕士阶段一直默默关心着我，伴我成长，教会我良善与坚韧。

这三年的硕士经历让我拥有了很多，感触最多的是要坚信、要自信、要学会与自己和解！

作者简历以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，在安徽大学电子信息工程学院获得学士学位。

2017 年 9 月-2020 年 6 月，在中科院微小卫星创新研究院攻读硕士研究生学位。

申请或已获得的专利：

[1]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（已受理）