中国科学技术大学

工程硕士研究生学位论文

开题报告

论文题目：面向嵌入式设备神经网络运行时内存优化

学 生 姓 名：翟文杰

学 校 导 师：朱宗卫

企 业 导 师：

工 程 领 域：软件工程

领 域 代 码：085212

研 究 方 向：嵌入式系统设计

所 在 院 系：软件学院

实 习 单 位：

中国科学技术大学研究生院

填表日期：　　年　月　日

说 明

1. 工程硕士学位论文的开题报告是保证论文质量的一个重要环节，为了加强对工程硕士研究生培养的过程管理，规范其学位论文的开题报告，特制此表。
2. 工程硕士学位论文开题报告，应该在工程硕士学位授予点或培养单位组织的学术报告会上报告，听取意见，论证后再填写此表。
3. 此表一式两份经导师和培养单位负责人签字后，交培养单位研究生教学管理办公室存档。
4. 工程硕士研究生在申请学位论文答辩时，必须提交该学位论文开题报告。

一、 简况

<table>
<tr><td rowspan="5">研究生简况</td><td>学　号</td><td colspan="2">SA19225464</td><td>姓 名</td><td>翟文杰</td><td>姓名拼音</td><td>Zhai WenJie</td></tr>
<tr><td>性　别</td><td>男</td><td>身份证号</td><td colspan="2">510106199612055112</td><td>出生年月</td><td>1996.12.05</td></tr>
<tr><td>工程领域</td><td colspan="3">软件工程</td><td>研究方向</td><td colspan="2"></td></tr>
<tr><td>入学时间</td><td>2019 年 9 月</td><td>录取方式</td><td>统考</td><td>培养方式</td><td colspan="2">脱产　不脱产 √</td></tr>
<tr><td>本科毕业时间</td><td>2019 年 6 月</td><td>本科毕业学校</td><td>西南石油大学</td><td>本科专业</td><td colspan="2">软件工程</td></tr>
<tr><td rowspan="6">论文类型与性质</td><td rowspan="2">名称</td><td>中文</td><td colspan="5">面向嵌入式计算设备的卷积神经网络运行时内存优化</td></tr>
<tr><td>英文</td><td colspan="5">Memory optimization of deep neural networks on edge computing devices based on task characteristics</td></tr>
<tr><td>领域</td><td colspan="6">智能嵌入式系统设计</td></tr>
<tr><td>类别</td><td colspan="6">1．技术攻关研究 √　　2．工程项目策划
3．工程设计或技术改造　　4．新工艺、新材料、新设备、新产品的研制与开发</td></tr>
<tr><td>形式</td><td colspan="6">1．工程设计
2．研究论文 √</td></tr>
<tr><td>性质</td><td colspan="6">1．基础研究
2．应用基础研究 √</td></tr>
<tr><td rowspan="4">论文内容和意义</td><td>摘要</td><td colspan="6">近年来，由于神经网络在视觉识别和图像分类方面取得的巨大成功，极大地推动了物联网加速向智能化方向发展。然而，现有的深度神经网络模型计算量大且内存密集，阻碍了其在内存和计算资源受限的物联网嵌入式设备上的部署。因此，许多研究聚都焦于如何压缩深度网络模型尺寸和加速其推理效率，而不显著降低模型性能和精度。在过去五年中，这方面取得了巨大进展，如参数剪枝、量化、低秩分解、转移/压缩卷积滤波器和知识蒸馏等。这些研究和技术拓宽了 AI 的应用场景，深度网络模型得以直接部署在边缘计算嵌入式设备上直接进行运算，而无须将数据交付至云计算中心运算，显著的降低了计算延时，提高了许多实时应用如自动驾驶、人脸识别、织物瑕疵检测等的服务质量。但模型压缩会带来模型精度的下降，当精度降低到不可接受时，再进一步压缩剪枝就变得不可能。因此本研究从体系结构的角度出发，分析 DNN 网络推理时计算和访存模式，提出一种在不影响模型精度之上减少 DNN 网络运行时内存的方法。本研究首先探讨了相同类型的任务的共同特征，并利用这些粗粒度上的特征构建关键局部子网络，用于热网络预取。其次，探讨了一种模型权值的增量加载策略，用于模型推理时网络模型的动态加载。</td></tr>
<tr><td rowspan="3">主题词</td><td colspan="6">主题词数量不多于三个，主题词之间空一格（英文用“/ ”分隔）</td></tr>
<tr><td>中文</td><td colspan="5">嵌入式平台 内存管理 深度学习推理</td></tr>
<tr><td>英文</td><td colspan="5">Embedded devices/Memory Management/Deep learning reasoning</td></tr>
</table>

二、选题依据

1. 本课题研究意义及同类研究工作国内外研究现状及分析（引用的参考文献需在文中标注）。

在上个世纪末，Yann LeCun 等人已经使用神经网络成功识别了邮件上的手写邮编。深度学习卷积神经网络的概念是由 Geoffrey Hinton 等人首次提出。在 2012 年，Krizhevsky 等人采用深度学习算法，以超过第二名以传统人工设计特征方法准确率 10%的巨大领先取得了 ImageNet 图像分类比赛冠军。此后的计算机视觉比赛大奖已经被各种深度学习模型所包揽。但这些模型依赖于具有数百甚至数十亿参数的深度网络，传统 CPU 无法训练如此庞大的网络，只有具有高计算能力的 GPU 才能让网络得以相对快速训练。如 Krizhevsky 等人在比赛中使用的模型使用了 1 个包含 5 个卷积层和 3 个完全连接层的 6000 万参数的网络。通常情况下，即使使用当时性能顶级的 GPU NVIDIA K40 来训练整个模型仍需要花费两到三天时间。对于使用全连接的大规模网络，其参数规模甚至可以达到数十亿量级。为了解决全连接层参数规模的问题，许多学者转而考虑增加卷积层，使全连接参数降低。随之带来的负面影响便是大大增长了计算时间与能耗。此时的深度学习任务尚且只能够通过大规模集群的 GPU 运算完成。同时随着物联网加速发展，也让许多学者开始思考如何将深度学习算法部署在嵌入式设备上，进一步推动物联网向智能化发展。

但将深度学习算法部署在计算和存储资源都十分有限的嵌入式设备上是一件十分挑战性的任务。为了满足深度学习的计算需求，许多学者提出了云计算的概念。即将数据从网络边缘的数据源(从智能手机到物联网(IoT)传感器)传输到云服务平台，让云服务平台进行计算后再将结果返回到边缘设备端。但同时这种将数据从源迁移到云的解决方案带来了几个问题：

a) **延迟**：实时推理对于许多应用程序都是至关重要的。例如，自动驾驶汽车上的摄像机帧需要快速处理以检测和避免障碍物，或者基于语音的辅助应用程序需要快速解析和理解用户的查询并返回响应。但是，将数据发送到云中进行推理或训练可能会导致额外的排队和网络的传播延迟，无法满足这些有着端到端、低延迟需求的交互式应用场景。例如，实验表明，将摄像机帧 offload 到 Amazon Web 服务服务器并执行计算机视觉任务需要 200 ms 以上的端到端时间。

b) **隐私性**: 向云发送数据可能导致数据的泄露问题。例如，最近在纽约市的智能城市环境中部署的摄像头和其他传感器引起了隐私监管机构的严重关注

这些致命的缺点让许多学者开始思考新的解决方案，即如何将深度学习网络模型直接部署在边缘嵌入式设备端直接进行计算。如 Krizhevsky 在 2014 年的文章中，提出了两点观察结论：卷积层占据了大约 90-95%的计算时间和参数规模，有较大的值；全连接层占据了大约 5-10%的计算时间，95%的参数规模，并且值较小。这为后来的研究卷积神经网络的压缩与加速提供了统计依据。一个典型的例子是具有 50 个卷积层的 ResNet-50 需要超过 95MB 的存储器以及 38 亿次浮点运算。在丢弃了一些冗余的权重后，网络仍照常工作，但节省了超过 75％的参数和 50％的计算时间。此后，许多研究者在降低内存需求和计算开销方面做了大量的工作，极大地提高深度神经网络（DNNs）的推理效率。现有的方法包括网络剪枝[3]、[4] 、轻量级体系结构设计[5]、[6]和自适应推理[7]。其中，网络剪枝和自适应推理因其有效性而日益受到关注。这些研究和技术

极大地推进了人工智能应用在边缘嵌入式设备上的落地实现，推动加速物联网智能化的发展。多层节点的大型网络存储和计算成本的减少，让许多实时应用成为现实，如在线学习、自动驾驶、智能工厂、智能电梯以及智能高速公路等。通过直接在嵌入式设备中部署深度学习模型，并分析运算第一数据，从而实现广泛的人工智能应用。具体地来说 CNNs 模型的存储需求涉及两部分[1][2] a)模型数据：CNNs 模型训练完成后，需要在嵌入式设备上部署模型结构和权重，并加载模型权重。b)运行时数据：在模型推理过程中，需要额外的存储空间来存储网络生成的特征映射和算子并行计算的中间结果。

虽然已经有很多剪枝技术来减少深度学习模型的大小，但是在 DRAM 受限的嵌入式设备中部署一个剪枝后的深度神经网络仍是有待改进的地方。通常来说，剪枝模型会减少模型的尺寸大小，但一方面也会减少模型的精度。例如， 裁剪掉 1.96%大小的 AlexNet 模型在 imageNet 数据集上的精确度损失了 0.97%。当精度降低到变得不可接受时，再进一步修剪就变得不可能。最近，有人提出了一种交换技术来在有限的 GPU 全局内存上操作大型模型[8]。然而，这种方法依赖于大量的 CPU-DRAM，这在 DRAM 资源受限的边缘嵌入式设备上是不适用的。就如何压缩模型并使其能够部署在嵌入式设备上，国内外已有广泛的研究，下面将从多个角度来分析讨论国内外研究现状。

首先，分析嵌入式设备上深度网络模型的推理和部署上的特点，因为有限的计算和内存的资源使得深度神经网络在嵌入式设备上的在部署、计算推理上较之于其他平台上的部署有很大的不同和区别。

1.1 嵌入式设备上的推理

在任务种类上，与传统的图像分类和识别系统相比，大多数边缘计算设备通常工作在相对简单的场景中，例如工厂自动化管道、高速公路、办公室和电梯内部[26]。在实际的工业和民用环境中，嵌入式设备只需要正确识别少数几类对象。例如，在工业自动化生产过程中，嵌入式计算设备上的神经网络通常在质量控制过程中被训练识别少量对象。在纺织行业，网络只训练织物疵点图像[27]，以提高产品质量，取代人工检测方法。在民用安全领域，嵌入式计算设备通常使用神经网络来识别一类目标。以前的研究使用专门的硬件来优化嵌入式设备的推理过程[28]。提高具有大型 DRAM 的高端平台的能效和性能。然而，这些优化需要特殊的硬件（FPGA），这限制了模型只能在某些硬件平台上部署。然而，上述工作仍然可以忍受 DRAM 不足的瓶颈，只适用于深度学习功能有限的简单任务。在部署上，常常使用深度学习框架。这些常用的深度学习框架可以分为两类：命令式编程和声明式编程。命令式编程的运行方式在执行过程中进行计算，便于模型结构和算子的研究和优化。虽然声明式编程是基于图形执行的，但它的优点是易于部署，并且在推 阶段具有更高的性能。因此目前几乎所有嵌入式设备上的部署模型都使用声明式编程。如 Tensorflow-Lite[24]、LibTorch[25]、NCNN[11]等。除此之外，为了加速资源约束下的模型推理，声明式编程也更适合于嵌入式计算设备上的部署。这是因为在模型推理运行之前，可以通过静态分析获得模型内存分布和数据流等详细信息，从而加速推理。

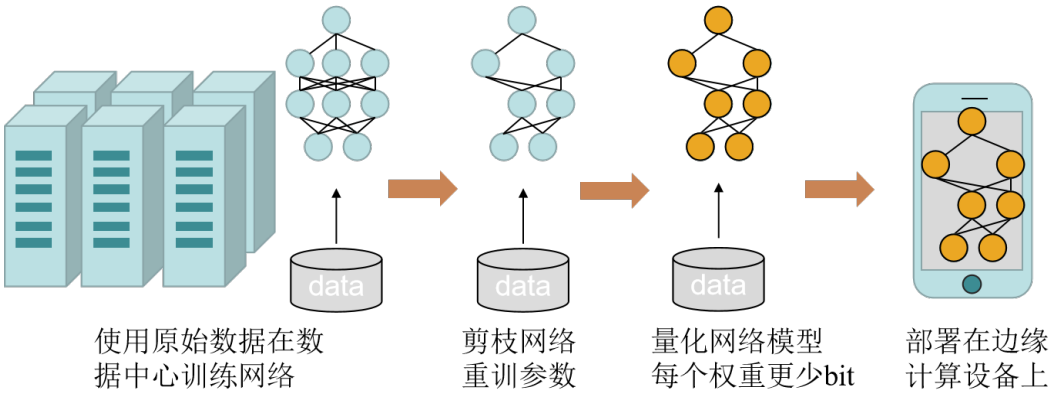


图 1 在边缘嵌入式计算设备上部署神经网络的流程

其次，在对嵌入式设备上部署神经网络和其推理过程的特点的分析之后，许多学者和研究员基于对嵌入式设备上特点的观察，从深度神经网络模型本身的角度出发，提出了许多压缩模型的方法以适配嵌入式平台下深度神经网络模型推理和部署特点。

1.2 卷积神经网络压缩

最近几年将深度神经网络部署在资源受限制的嵌入式设备上收到的了广泛的关注。一部分的研究主要通过压缩预训练的网络使得神经网络能够部署在资源受县的嵌入式设备上。其中包括 i)网络修剪，删除不重要的参数。如 Song Han, Jeff Pool 等人在[3]中提出参数剪枝的方式，通过判断参数是否大于某个阈值，来确定哪些是需要保留的，将 vgg-16 压缩了 13 倍。或是删除不重要的通道，如 Yihui He, Xiangyu Zhang 等人在[2]采用选择保留权值最大的通道，裁剪网络。ii) 减少参数比特宽度的网络量化。如 Song Han, Huizi Mao 等人在[13]中提出剪枝网络模型，量化权重增加共享权重，通过霍夫曼编码减少权值位数，从而减少网络模型大小。但是这些方法的起点都是一个训练效果良好的网络，无法处理训练阶段。另一部分研究主要通过人工设计[5,6,7,8]。如 Yihui He, Xiangyu Zhang 等人设计了一个两个超参数来建立非常小的、低延迟的模型 Mobilenets[7]，以便轻松匹配移动和嵌入式视觉应用的设计要求。Forrest N Iandola, Song Han 实现了一个高效的网络模型 Squeezenet[8]，在和 Alexnet 同等精度的情况下，参数少了 50 倍，模型压缩了 510 倍。诸如此类的许多专家特制的 Shufflflenet、CondenseNet[5,6]都在资源受限的嵌入式设备上取得了不错的效果。这些轻量级的网络保持了较高的识别准确性，并能够成功部署在嵌入式设备上。然而，设计出高性能的轻量级神经网络需要大量的专业知识与反复试验，成本极高，限制了神经网络在很多问题上的应用。因此最近兴起的 NAS 神经网络结构搜索引起了广泛的注意。从[16,17]开始，制定算法根据样本集自动设计出高性能的网络结构，在某些任务上甚至可以媲美人类专家的水准。但 NAS 的性能在很大程度上取决于搜索空间的质量[18]。传统上，许多学者在 NAS 搜索空间设计中遵循手动设计启发式方法。例如，广泛使用的移动设置(mobile-setting)搜索空间[20]源于 MobileNetV2[19]：它们都使用 224 输入分辨率和类似的基本信道数配置，并搜索内核大小、块深度和扩展率。然而，对于内存有限的嵌入式计算设备，缺乏标准的模型设计，搜索空间的设计也是如此。一种可能的方法是手动调整每个嵌入式计算设备的搜索空间。但是，通过试验和错误进行手动调整是一种劳动密集型的工作。

最后，除了优化模型尺寸以适配嵌入式平台下推理和计算的特点的研究之外，最近也有学者从体系结构特点提出了交换扩展 DRAM 空间的方法从逻辑上扩大 GPU 设备的 DRAM 空间大小，但却很少关注于适应嵌入式设备的内存约束。

1.3 交换扩展 DRAM 空间

传统的交换技术通过将冷数据交换到外部存储器（如 NAND 闪存设备）并按需获取所需数据，从而在逻辑上扩展了 DRAM 空间[21]，[22]。然而，在当前交换技术下简单地传输运行时数据可能会在缓存未命中发生时产生 I/O 惩罚，因为 DRAM 中命中的数据必须从闪存中读取，并且在推理过程中会产生不可接受的 I/O 时间开销。最近的一项研究提出通过交换图形处理单元中的主机 CPU 内存来扩展 GPU 内存[23]。对于深度学习训练，Capuchin[10]被提出通过在 GPU 和 CPU 空间之间的张量逐出(eviction)/预取来减少内存占用。然而，这些研究仅限于存储空间之间的数据交换，很少关注在推理过程中如何适应边缘计算嵌入式设备的内存约束。由于 DRAM 未命中造成的 I/O 代价巨大，目前很少有人致力于集成 DRAM 和 flash 存储之间的交换解决方案。

综上所述，通过以上三个角度的观察和分析，嵌入式设备上因其有限的硬件资源产生对深度网络模型特有的需求。这些需求主要可以被归结为 a) 低主存(DRAM)占用 b)低浮点计算。基于这些特点，许多学者从不同角度提出了不同的解决方案，一是从模型本身角度出发，降低模型本身尺寸大小和计算力需求。二是从程序运行角度提出交换扩展 DRAM 空间。本课题研究的目的是在不影响模型精度的基础之上，减少图像分类推理过程中的主存空间需求。本课题首先从深度神经网络模型权重加载和张量生成行为和图片分类的任务特征两个角度进行了深入分析。1) 模型角度：在资源受限的设备（如嵌入式设备）下，深度神经模型的计算推理过程可以序列化[1]，以避免多个算子同时进行并行计算[9]-[11]，这使得序列化加载算子从而减少主存占用成为可能。 2)任务特征：相似图像在卷积核通道的激活值上表现相似。例如，CIFAR-10[12]上的层次聚类结果可以分为两大类。可以根据通道激活对不同类别的图像进行聚类，因此可以通过层次聚类来检测输入图像的模糊类别。这使得能够根据任务的不同特征，预取热子网络，换出冷子网络于外存上，减少模型运行时主存占用。

２．主要参考文献（列出作者、论文名称、期刊名称、出版年月）。

[1]Z. Liu, J. Li, Z. Shen, G. Huang, S. Yan, and C. Zhang,“Learning effificient convolutional networks through network slimming,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2736–2744.

[2] L. Ren, X. Cheng, X. Wang, J. Cui, and L. Zhang, “Multi scale dense gate recurrent unit networks for bearing remaining useful life prediction,” Future Generation Computer Systems,

vol. 94, pp. 601–609, 2019.

[3]Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for effificient neural network. In Advances in neural information processing systems, pages 1135–1143, 2015.

[4] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. 2017.

[5] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflflenet: An extremely effificient convolutional neural network for mobile devices. CoRR, abs/1707.01083, 2017.

[6] Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Condensenet: An effificient densenet using learned group convolutions. CoRR, abs/1711.09224, 2017.

[7] Howard, A. G. , Zhu, M. , Chen, B. , Kalenichenko, D. , Wang, W. , & Weyand, T. , et al. (2017). Mobilenets: efficient convolutional neural networks for mobile vision applications.

[8] Iandola, F. N. , Han, S. , Moskewicz, M. W. , Ashraf, K. , Dally, W. J. , & Keutzer, K. . (2016). Squeezenet: alexnet-level accuracy with 50x fewer parameters and <0.5mb model size.

[9] Andreas Veit and Serge Belongie. Convolutional networks with adaptive inference graphs. In Proceedings of the European Conference on Computer Vision (ECCV), pages 3–18, 2018.

[10]Xuan Peng, Xuanhua Shi, Hulin Dai, Hai Jin, Weiliang Ma, Qian Xiong, Fan Yang, and Xuehai Qian. Capuchin: Tensor-based gpu memory management for deep learning. In ASPLOS, 2020.

[11] Tencent, “Accessed: 2019-09-01,” https://github.com/Tencent/ ncnn, 2017.

[12] J. Cui, L. Ren, X. Wang, and L. Zhang, “Pairwise comparison learning based bearing health quantitative modeling and its application in service life prediction,” Future Generation Computer Systems, vol. 97, pp. 578–586, 2019.

[13] J. Zhou, X. S. Hu, Y. Ma, J. Sun, T. Wei, and S. Hu, “Improving availability of multicore real-time systems suffering both permanent and transient faults,” IEEE Transactions on Computers, vol. 68, no. 12, pp. 1785–1801, 2019.

[14]A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Handbook of Systemic Autoimmune Diseases, 1(4), 2009.

[15]Song Han, Huizi Mao, and William J Dally. Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding. In ICLR, 2016

[16] Cai, H. , Chen, T. , Zhang, W. , Yu, Y. , & Wang, J. . (2017). Efficient architecture search by network transformation.

[17] Wu, B. , Keutzer, K. , Dai, X. , Zhang, P. , & Jia, Y. . (2019). FBNet: Hardware-Aware Efficient ConvNet Design via Differentiable Neural Architecture Search. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

[18] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. arXiv preprint arXiv:2003.13678, 2020.

[19] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc VLe. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In CVPR, 2019

[20] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. arXiv preprint arXiv:1605.07678, 2016.

[21]Duo Liu, Kan Zhong, Xiao Zhu, Yang Li, Lingbo Long, and Zili Shao. Non-volatile memory based page swapping for building high performance mobile devices. IEEE Transactions on Computers, 2017.

[22] Niel Lebeck, Arvind Krishnamurthy, Henry M. Levy, and Irene Zhang. End the senseless killing: Improving memory management for mobile operating systems. In USENIX ATC, 2020.

[23] Chen Li, Rachata Ausavarungnirun, Christopher J. Rossbach, Youtao Zhang, and Jun Yang. A framework for memory oversubscription management in graphics processing units. In ASPLOS, 2019

[24] J. Lee, N. Chirkov, E. Ignasheva, Y. Pisarchyk, M. Shieh, F. Riccardi, R. Sarokin, A. Kulik, and M. Grundmann, "On device neural net inference with mobile gpus," arXiv preprint arXiv:1907.01989, 2019.

[25] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., "Pytorch: An imperative style, high-performance deep learning library," in Advances in neural information processing systems, 2019, pp. 8026–8037.

[26]Xin He and Delu Zeng. Real-time pedestrian warning system on highway using deep learning methods. In 2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2017

[27] Xianghui Liu, Zhoufeng Liu, Chunlei Li, Bicao Li, and Baorui Wang. Fabric defect detection based on faster r-cnn. In International Conference on Graphic Image Processing, 2018

[28] Tianshi Chen, Zidong Du, Ninghui Sun, Jia Wang, Chengyong Wu, Yunji Chen, and Olivier Temam. Diannao: A small-footprint high-throughput accelerator for ubiquitous machine-learning. ACM SIGARCH Computer Architecture News, 42(1):269–284, 2014

三、课题内容及具体方案

1．课题的研究目标

本课题主要研究目标是提出一种与现有主流模型压缩方法正交的减少模型运行时内存的方法，该方法是其它现有模型压缩方法的补充。重点关注于神经网络模型运行时内存的使用情况，分析神经网络模型运行时特征，并基于进一步减少网络模型运行时内存占用。

2． 课题的研究内容（列出具体的研究内容，而不是学习某些课题相关的知识）

本课题主要研究以下几个方面

1）神经网络模型的运行时特征分析；

2）基于通道选择局部子网络构建和热网络动态预取策略；

3）面向深度神经网络运行时动态增量加载模型权重的方法；

4）面向深度神经网络框架层和 OS 内存分配器的优化；

3． 拟解决的关键问题

1） 基于任务特征局部子网络构建；

2） 基于局部子网络感知的热网络动态预取策略；

3） 面向深度网络模型推理时的内存访问特点构建增量加载模型权重策略；

4． 拟采用的研究方案和技术路线等

本节按照第 2 节**课题研究内容**分层展开叙述，并在最后总结出本研究的整体方案策略。

1） 神经网络模型的运行时特征分析

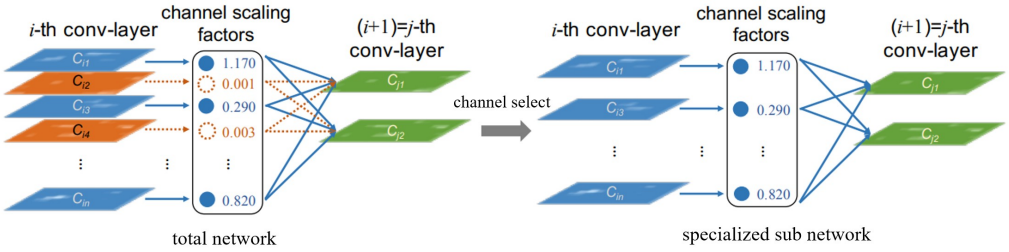
在分析神经网络模型运行时的特点时，拟定主要从以下几点进行分析。

a) 数据精度的需求; 对数据精度的需求又可以细分为训练时精度需求和推理时精度需求，因本研究着重在于神经网络在嵌入式设备上的推理，故拟定分析嵌入式设备上神经网络模型推理时对精度的需求。

b）计算热点；对于计算热点的分析，也能够细分为推理时的计算热点和训练上的计算热点。主要分析神经网络模型在嵌入式设备上推理时的计算热点。

c) 访存模式；在访存模式上，在网络前向推理(推理)和反向传播(训练)时,访存模式基本一致，本研究更注重于框架层上模型运行时张量内存的释放和回收。

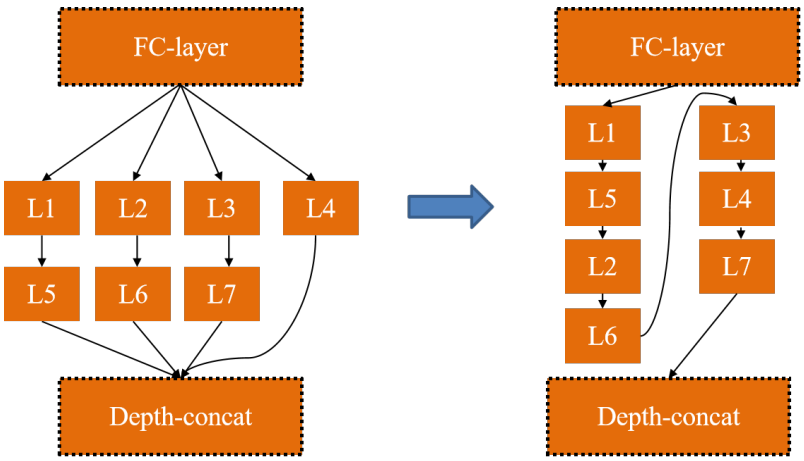
2） 基于通道选择的局部子网络构建



图二 比例因子裁剪形成专用子网络[1]

在构建局部子网络时拟定使用比例因子与卷积层中的每个通道相关联。在训练的过程中，对这些缩放因子进行稀疏正则化，以自动识别不重要的通道。比例因子值较小的通道（橙色）将被修剪。修剪后就能够获得裁剪后的一个专用模型，然后对其进行微调以达到与正常训练网络相当的准确性。这样一来就能通过输入划分好的数据集，进行训练，自动的按照输入数据集的内容而进行自动的裁剪形成一组裁剪过后的专用子网络。裁剪后子网络尺寸大小之和应与未裁剪子网络大小相当。

3） 面向深度神经网络运行时动态增量加载模型权重的方法

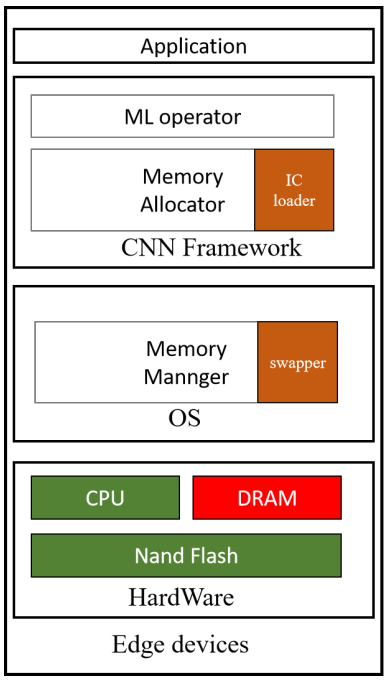


图三 并行计算结构序列化

在模型训练中，最终得到模型的网络结构和权重两部分。随后的推理是将图像数据输入到模型中。网络结构决定了计算的方式，在网络结构中通常会有如图三所示的并行计算结构，权重是对图像数据的矩阵运算。对于 Alexnet 模型，由于其网络具有多个完全连通的层，其权值占内存空间的 93.63%。可以看出，模型的大量的权重过大，这使得一次性将所有权重读入到内存有限的嵌入式设备上是不可接受的。基于此对模型的观察，如果权重的读取在其使用前完成，就能够正确的完成整个推理过程，同时能够显著的减少模型运行时主存占用开销。增量加载器首先需要保证的是每层的权重应读取操做应在每层运行之前完成，确保在执

行一层推理时，其所需要的权重参数都是存在内存当中的。同时交换器会负责将那些不需要用到的参数立即换出到磁盘外存上。在设计实现权重增量加载器有两个挑战：a) 如何保证权重的读取时间开销不会影响网络的推理执行 b) 如何保证每层网络在使用前都能将其所用到的参数加载至内存中。

4） 面向深度神经网络框架层和 OS 内存分配器的优化



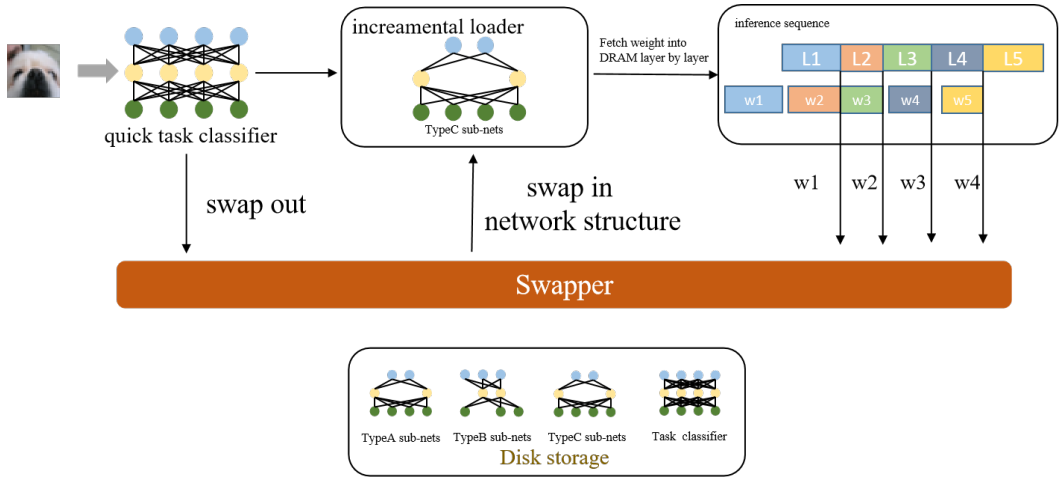
图四 swapper 和 IC loader 在整体架构上的位置

在 OS 层面上，快速分类器也是将交换技术引入到嵌入式系统推理的关键点。拟定的快速分类器能够达到将图片快速分类的效果，将图片划分为一个大致的类别，而不是直接识别出其具体属于哪一类，在这个基础之上，就可以将后续的网络裁剪为一个专用的网络，这样能够进一步减少主存的占用。比如一个数据集中存在 10 类不同的物体，快速识别器首先将这 10 类物体划分为 2 类，那么就可以基于此单独训练两类不同的专用网络，A 网络负责识别 5 类，B 网络负责识别另外 5 类。然后对 A 网络和 B 网络进行裁剪，减少其大小。但至于快速分类器如何设计，以及采用何种方式实现，是一个需要考虑的问题。交换技术在保证 DNN 模型完整性的同时，能够扩大与外部 flash 存储器的存储空间。然而，推理过程通常会激活 DNN 模型的动态子网执行，使得精确选择交换数据变得困难。综上所述，在 OS 层面将交换技术集成到基于 DNN 的嵌入式系统中有三个挑战：a）如何准确地识别和划分不同任务的子网工作；b）如何有效地获取对推理任务性能至关重要的数据 c)如何训练构建快速分类器以及快速分类器。

在深度学习框架层面上，对于模型中张量的创建释放，现今主流框架 tensoflow，pytorch 等都有一套自己的内存管理器负责。对框架层上内存管理器修改是将增量权重加载策略引入

的关键点。拟定对修改框架层内存管理器，从而实现权重增量加载的策略。

5） “热网络预取+模型权值动态加载”整体策略



图五 整体架构

综上所述，将所有模块整合形成拟定的整体架构如图四所示，梳理一张图片在深度神经网络的推理流程。一张图片首先被一个快速分类器划分为一个大致类别，该快速分类器并不常驻内存，而是用完立即被换出到磁盘上。后面的增量加速器会根据快速分类器得到的结果，将对应的专用网络的图结构加载到内存中，网络的图结构占内存空间并不大，模型主要的体积是其浮点数表示的参数权重。随后增量加载器会根据网络的图结构，动态增量将权重加载至内存中。没有用到的参数将会被换出到外存上，以减少内存占用。

6） 方案总结与实施

针对上述讨论中所遇到的挑战，逐个调研落地方案。a) 如何准确地识别和划分不同任务的子网工作。b) 如何训练构建快速分类器以及快速分类器的可行性。这部分内容主要设计灵感来自于机器学习的中的决策分类树，但是数据集却是图像数据。国内外有不少学者提出了使用决策树对图像数据进行分类的方法。同时也有相关文献使用神经网络使用相同的思想做该项工作如在 2019 年 ICLR 上 Wieland Brendel and Matthias Bethge 提出了近似 CNN 的网络 BAG net。在调研这两个问题的答案时，拟定采用传统机器学习方法+深度神经网络的调研步骤去研究论证该方法的可行性。

在设计实现权重增量加载器时所遇到的两个挑战 a) 如何保证权重的读取时间开销不会影响网络的推理执行 b) 如何保证每层网络在使用前都能将其所用到的参数加载至内存中。CNN 框架由一组计算层组成和一层内存管理分配层组成，每个计算层可以产生许多中间数据，通常为了加速内存的分配和释放，tensor 使用内存池进行管理，这也带来了碎片。同时 CNN 框架层也运行在 OS 之上，OS 也会有自己的内存管理器。因此调研这部分内容时，拟定采用研究 OS+CNN 框架层中的内存管理器，通过修改这两个内存管理器完成增量加载模型的实现。

5. 项目的特色与创新之处

近年来随着卷积神经网络的高速发展，已出现越来越多的 AI 应用。但将卷积神经网络部署在主存 DRAM 受限的嵌入式计算设备上是一个非常具有挑战性的问题。最主要原因是因为 CNN 网络体积过于庞大，为了解决这个问题许多学者提出了很多卷积神经网络压缩的方法，如知识蒸馏、轻量卷积神经网络设计、低秩近似、参数量化、结构化剪枝、自动化神经结构搜索等，这些方法都取得了不错的表现，使得将卷积神经网络部署在内存受限的嵌入式计算设备上成为可能。本课题的创新之处主要在于

1） 从体系结构的角度出发，结合卷积神经网络在运行时的访存特点和计算特点，提出用交换和增量加载的方式减少卷积神经网络运行时内存的内存占用

2） 该项目所使用的方法与现今主流的模型压缩方法正交，是它们的进一步补充。能够结合任意模型压缩的方法，并在它们的基础之上，进一步减少运行时空间内存占用。

6、本人主要工作描述

1）调研模型压缩的相关文献

2）调研嵌入式计算设备部署 CNN 的文献

3）总结归纳调研有用信息

4）提出交换增量加载权重技术减少运行时内存

5）设计实验，检验方法的有效性

6）对相关模块编写代码程序

7）编写测试程序，验证实验结果

四、工作进度的大致安排

应包括文献调研，工程设计，项目开发和调试，实验数据的分析处理，撰写论文等。

时间	工作	预期成果
2020.12-2021.1	调研相关文献，验证问题的研究意义	开题报告和相关技术设想
2021.2-2021 .4	深入调研相关文献，并研究相关的技术方案，调研高效神经网络的技术实现方法	搭建初步框架，能够完成基本的测试程序
2021.5-2021 .7	深入调研相同类型的任务的共同特征，并利用这些粗粒度上的特征构建关键局部子网络的可行性和实现方案	完成局部子网络的构建
2021.7-2021 .8	深入调研模型权值的增量加载策略，用于模型推理时网络模型的动态加载的可行性和实现方案。	完成模型权值的增量加载策略，并实现程序
2021.9-2021.11	深度调研从空间维度对数据布局进行重组，优化嵌入式计算设备运行时的内存空间的可行性和实现方案	完成数据布局的重组，并实现程序
2021.11-2021.12	整合所有部分的内容	整合所有内容，完成集成测试
2022.1-2022 .6	撰写毕业论文	论文初稿和论文定稿

五、预期成果

应包括软硬件产品、文档、模型、专利、论文等

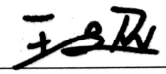
1） 提出完整的嵌入式设备高效 DNN 网络的实现落地方案；

2） 实现测试程序的编写、开发与部署；

3） 完成毕业论文；

4） 发表相关学术论文；

六、审核意见

导师意见 同意开题 导师签名：  年 月 日
培养单位负责人意见 同意开题 培养单位负责人签名：  年 月 日

七、评审表

中国科学技术大学研究生学位论文开题报告评审表

研究生姓名	翟文杰	学　　号	SA19225464	所在院系	软件学院
学科、专业	软件工程	研究方向	嵌入式系统设计	指导教师	朱宗卫
拟撰写的学位论文题目	面向嵌入式设备深度神经网络运行时内存优化				
支持论文研究的科研项目	基于寒武纪国产智能处理器的异构集群分布式训练任务调度方法研究				
学位论文是否保密	1．不保密（√ ） 2．保密　（　）　密级：绝密（　）、机密（　）、秘密（　）			导师签字	

开题报告评审组成员名单

姓名	职称	工作单位	签名
周学海	教授	中国科学技术大学计算机学院	
韩恺	教授	中国科学技术大学计算机学院	
汪炀	副教授	中国科学技术大学计算机学院	
杨威	副教授	中国科学技术大学计算机学院	

指导教师意见：

同意开题

指导教师签字：　　年　月　日

评审小组意见：（是否通过开题论证，是否需要修改等）

通过开题，建议进一步修改研究内容、关键问题、研究方案和技术路线并细化。

评审小组组长签字：　　年　月　日

八、院审历史


开题报告提交
关闭
提交
开题报告已通过院审
审核开题报告
审核历史
审核人 审核时间 审核步骤 审核意见
院审导师 2021-3-16 院审通过 同意
朱宗卫 2021-2-18 导师审核通过 同意
翟文杰 2021-1-13 学生提交开题报告
翟文杰 2021-1-13 学生收回开题报告
翟文杰 2021-1-13 学生提交开题报告