论文框架：

1. Introduction

最近几年，CNN已经成为了计算机视觉任务的主要技术手段，在图像分类、目标检测、深度估计、语义分割等方向都大放异彩。尤其是自从AlexNet一举夺得ILSVRC 2012 ImageNet图像分类竞赛的冠军后，深度神经网络的热潮便席卷了整个计算机视觉领域。自此深度模型火速替代了传统人工设计（hand-crafted）特征和分类器。各种深度模型不仅提供了一种端到端的处理方法，还大幅度地刷新了各个任务的精度，更甚者已经超越了人类知识的准确度。目前为止，在图像分类模型的准确率已经达到95%（top5）语音识别率95%，与人类知识持平。

然而，深度模型优异的表现背后是强大的计算需求：在不断刷新任务精度极限的同时，其深度和尺寸也在成倍增长。各种新平台正在成为深度学习应用目标，包括移动平台，自治系统**和智能设备。所以随之而来的问题之一是：如此巨大的模型只能在有限的平台下使用，根本无法移植到移动端和嵌入式芯片当中，而且较高的带宽占用也让很多用户望而生畏；另一方面，大尺寸的模型也对设备功耗和运行速度带来了巨大的挑战。所以深度模型目前面临着在终端部署和低延迟需求场景下难以应用的问题。**

**因此，模型压缩是让深度模型真正应用在移动端不可缺少的一步，近年来，该领域实现了极大的发展。可以将主流的压缩方法分为三个类别：参数修剪和共享、低秩分解、知识提纯等。基于参数修剪（Parameter Pruning）和共享的方法可以分为两部分：权值量化和权值修剪，前者关注减小模型大小，后者则关注于探索模型参数中冗余的部分，并尝试去除冗余和不重要的参数。Gong[6]和Wu[7]对参数值使用K均值标量量化。Vanhoucke[7]展示了8bit参数**量化可以在准确率损失极小的同时实现大幅加速。基于模型裁剪的方法很多，其思路源头都是来自于Oracle pruning 的方法，即挑选出模型中不重要的参数，将其剔除而不会对模型的效果造成太大的影响。基于低秩分解（Low-rank factorization）[12, 13]技术的方法使用矩阵分解以估计深层网络中最具信息量的参数，然后通过将一个大矩阵分解为多个小矩阵进行计算。知识提纯（Knowledge Distillation, KD）[14, 15]则学习了一个精炼模型，即训练一个更加紧凑的神经网络以再现大型网络的输出结果。KD的基本思想是通过软Softmax 学习[16]教师输出的类别分布而将大型教师模型（Teacher Model）的知识提纯为较小的模型。

为了更好的将模型部署到移动端或者嵌入式系统上使用，我们的工作主要分为两步：

（1）对目前最经典使用广泛的深度学习模型在NVIDIA TX2嵌入式系统上进行性能的测试和分析，包括推理时间，模型大小，各操作数的时间花费，从而分析这些模型在嵌入式系统下的具体性能表现，这也是深度学习模型在移动平台的移植的第一步。

（2）研究基于深度学习的模型压缩技术，实现对几种经典网络的模型压缩，对比不同压缩方法下模型的负载特征，主要包括模型压缩前后的推理时间变化，模型大小变化以及准确率的的的损失等，为不同模型在移动端的使用更具可能。

1. Background
2. Workloads

对mobilenet，vgg，inception，Resnet介绍，仿照Fathom里的那部分。

Mobilenet:

Vgg:

Inception:

Resnet:

（2）Motivation:

从AlexNet、VGGNet到InceptionNet、ResNets，ImageNet分类比赛的冠军已经将模型由8层提高到了100+层，深度模型优异的表现背后是强大的计算需求：在不断刷新任务精度极限的同时，其深度和尺寸也在成倍增长。目前，最先进的深度学习模型通常具有数百万个需要存储的参数，而设备上的内存是有限的。此外，甚至一个模型推断调用O（109）存储器访问和算术运算并不罕见，所有这些都会消耗功率并耗散热量，从而可能消耗有限的电池容量和/或测试设备的热限制。

虽然这些模型通常部署在数据中心后端，但保留用户隐私并减少用户感知的查询时间需要将这些深度神经网络提供的智能服务迁移到边缘计算设备。然而将大型精确的深度学习模型部署到资源受限的计算环境（如手机，智能相机等）以进行设备推断时会带来一些关键挑战。所以要想让产品落地，模型压缩是必不可少的一环。

为了将深度学习模型部署到移动/嵌入式设备上，不仅需要对模型的性能在平台环境下做具体的分析，更重要的工作应该致力于减少模型的内存占用，缩短推断时间，减少耗电。因此，一种很自然的解决方案就是在保证分类准确率不显著下降的前提下对深层卷积神经网络进行压缩和加速。

综上所述，深度学习模型在移动端的部署所面临的问题已经亟待解决，近两年来，相关研究工作也在逐步开展，一方面需要分析模型本身的性能，一方面模型的压缩也是非常必要的研究点。模型压缩主要是围绕减小模型尺寸大小和推理时间进行实验的。通过量化共享权值可以对模型的尺寸起到很好的压缩作用，通过修剪权重可以进一步加速模型的推理时间，但是这些方法主要还是停留在实验结论上并没有真正在移动端或者嵌入式系统上部署模型并实践效果，所以真正在嵌入式平台上部署模型后进行性能测试以及压缩实验对于进一步检验模型在移动端或者嵌入式系统上部署可行性是必要的一步，也是本文的工作重点。

It is expected that the size of the DNNs (i.e., number

of weights) and the number of MACs will be larger for

the more difficult task than the simpler task and thus require

more energy

* 1. RETRAINING PRUNED NETWORKS TO REGAIN ACCURACY

1. Overview

Choose meaningful models

影响时间的主要因素 是operation么？

解释CPU为什么没有必要

不同模型在哪个阶段的什么操作数 用的cost高

1. 实验

（1）Inception系列，Resnet系列，Vgg以及Mobilenet等模型在Imagenet下的推理时间测试

（2）Inception系列，Resnet系列，Vgg以及Mobilenet等模型在Imagenet下的不同标准下(top1,3,5)的准确率测试

（3）时间花费的的百分比（我还测了embedding,word2vec,memnet这些自然语言处理模型的可以放在一起对比下）（Fathom里的图）

（4）模型量化

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 模型名称 | 量化前大小 | 量化后大小 | 准确率变换 | 推理时间变化 |
| mobilenet | 17.1M | 4.3M | -23% | -31.25% |
| Inception\_v1 | 26.1M | 6.8M | -3.2% | +25.3% |
| Vgg\_16 | 540.4M | 135.1M | -2.8% | +40% |
| Vgg\_19 | 561.2M | 140.3M | -3.1% | +45% |
| Resnet\_50 | 100.2M | 25.3M | -2.3% | 51% |
| Inception\_v2 | 43.9M | 11.3M | -2% | +41% |
| Resnet\_101 | 174.5M | 44.1M | -2.8% | +158% |
| Resnet\_152 | 235.9M | 58.73M | -3.3% | +180% |
| Inception\_v3 | 92.4M | 24.2M | -2.6% | +280% |
| Inception\_v4 | 167.2M | 42.5M | -1.8% | +198% |

由于mobilenet量化压缩后准确率损失很大，所以我们将mobilenet通过减少宽度，将模型变为更加紧密的小模型从而达到压缩的目的。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 宽度乘数 | 大小 | Top1 | Top2 | Top3 | 推理时间 |
| V1.0 | 17.1M | 0.707 | 0.855 | 0.896 | 18ms |
| V0.75 | 10.5M | 0.618 | 0.838 | 0.880 | 14.7ms |
| V0.5 | 5.4M | 0.641 | 0.805 | 0.854 | 11.3ms |
| V0.25 | 2M | 0.505 | 0.689 | 0.532 | 8.85ms |

（5）剪裁

前后对比然后把inception的再单独列出来分析

（6）量化+剪裁合并分析

5. 相关工作

6. 总结