|  |  |
| --- | --- |
|  |  |

高能效智能计算实验室

研究生毕业论文

进展报告

毕业论文名称： 面向嵌入式设备神经网络运行时内存优化

报告人： 翟文杰

报告填写时间： 2021/11/15

高能效智能计算实验室

**目录**

[一、 毕业论文基本情况 3](#_Toc55551580)

[1.1 拟完成的主要工作 3](#_Toc55551581)

[1.2 针对开题报告的调整情况 3](#_Toc55551582)

[二、 研究工作进展情况 3](#_Toc55551583)

[2.1 完成的主要工作概述 3](#_Toc55551584)

[2.2 完成的主要工作详述 4](#_Toc55551585)

[2.3 毕业论文撰写情况 18](#_Toc55551586)

[三、 主要问题及解决措施 18](#_Toc55551587)

[四、 下一步研究工作计划 21](#_Toc55551588)

[4.1 未完成部分的工作开展计划 21](#_Toc55551589)

[4.2 期刊、会议论文发表计划 21](#_Toc55551590)

[4.3 毕业论文开展计划 21](#_Toc55551591)

# 毕业论文基本情况

请写明论文需要完成的主要工作，按实际填写。形式可参照开题报告，写清楚总体目标，以及预计完成的2~3个问题。如针对开题报告有调整请在此列出。**此部分简要列出即可。**

## 拟完成的主要工作

1. **分析神经网络计算特点，优化神经网络计算过程，加速网络推理速度的同时减少神经网络运行时的峰值内存和计算期间内存读写量。**
2. **压缩神经网络尺寸体积，以减少神经网络模型在嵌入式设备上所需的存储空间。**

## 针对开题报告的调整情况

**不采用开题报告中所拟定的技术方案，后改用的技术方案在后文中详叙。**

# 研究工作进展情况

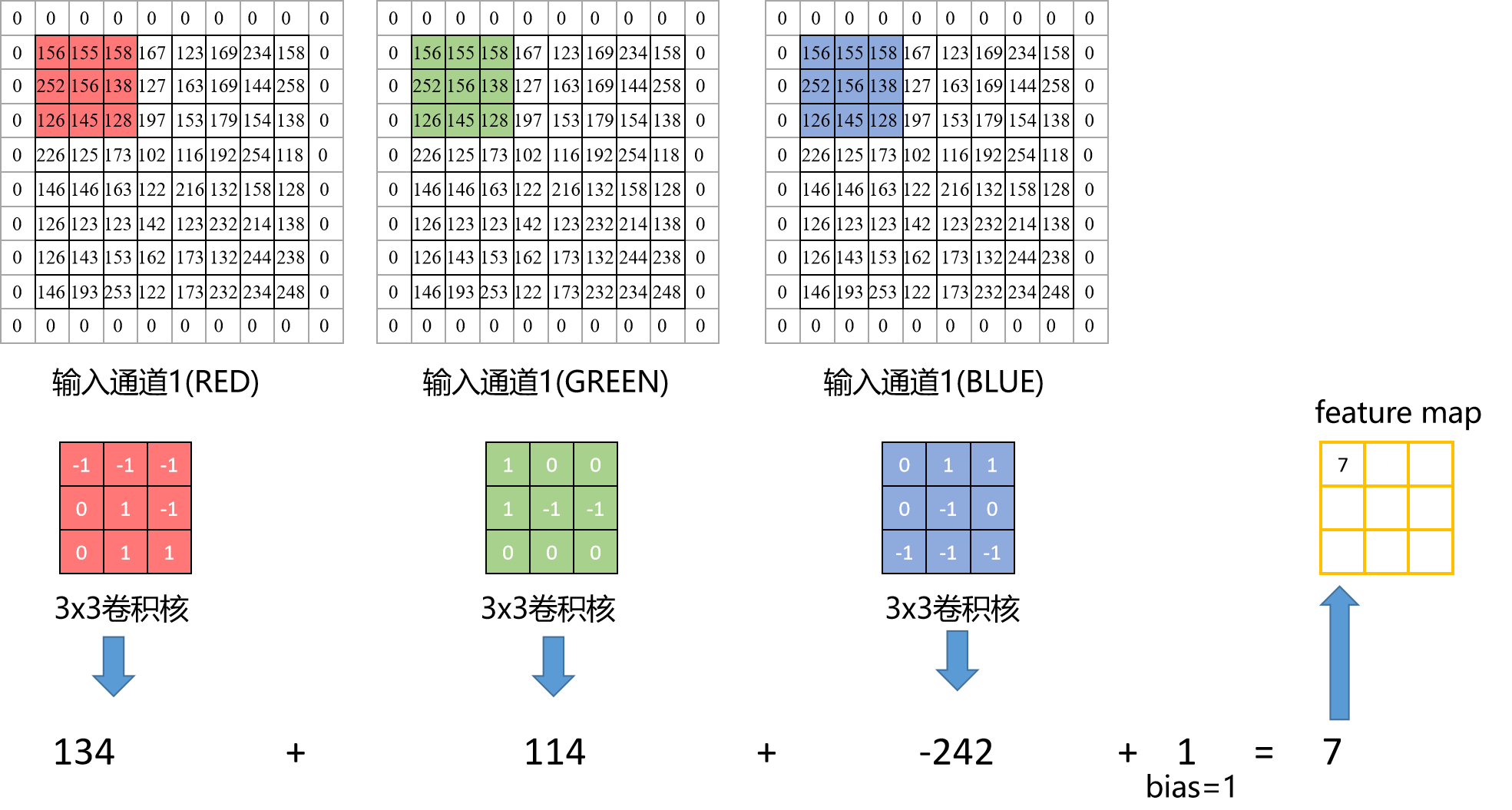
请写明论文已经完成的主要工作，按实际填写。**此部分为详细描述的内容，请针对前述的2~3个问题展开写，包括技术路线、实验数据等，篇幅不限。**

## 2.1 完成的主要工作概述

## 现已完成问题1并在期刊IEEE Transactions on Industrial Informatics上发表文章（DOI: 10.1109/TII.2021.3121783），问题2工作仍在开展中。

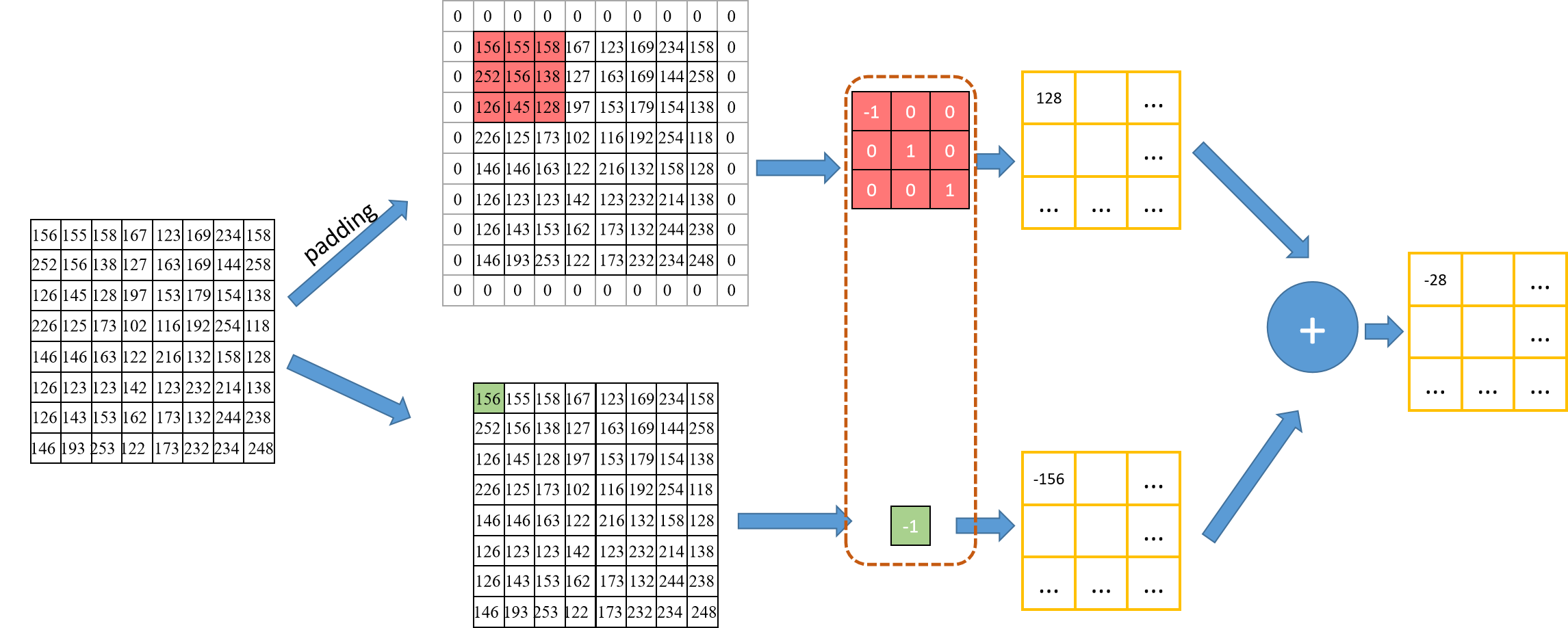
## 2.2 完成的主要工作详述

1. **动机实验和计算分析**
2. **卷积神经网络计算分析**



**图1 卷积核运算过程**

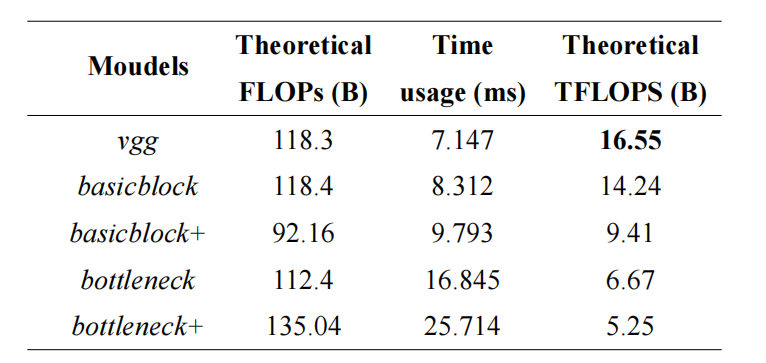
**图1具体的展示了神经网路中卷积核计算像素大小为8x8、RGB三通道图片的具体过程。从图中可以看到，在卷积运算的过程中除了需要存储基本的图片数据外，在运算的过程中还会产生运行时数据feature map。而feature map的大小，取决于输入图片的像素尺寸。一般来说，像素尺寸越大的图片，运行时产生的临时数据越多。**



**图2 分支结构的卷积运算操做**

**图2展示了resnet中shortcut连接处实际发生的卷积运算过程。如图中所展示的那样，在分支处，因分支两边的卷积核不同(一个1x1卷积核，一个3x3卷积核)，因此需要存储两份图片，用于进行不同的卷积运算。除此之外，在运算过程中会产生两份临时数据feature map用于最后的add合并操做。另外，分支结构还会增加推理时延，过多的内存读写会降低卷积神经网络在计算过程中的每秒浮点运算此处FLOPS。**

1. **动机实验**



**表1在 batch size = 32，通道数 C = 256 和输入图片分辨率 resolution = 56 的参数设置下， vgg 和 resnets 模块在 NVIDIA V100 的实际推理时间。测试结果是在预热硬件后，取平均 20 次运行时间所得到的结果。’+’ 表示该模块是有下采样的。**

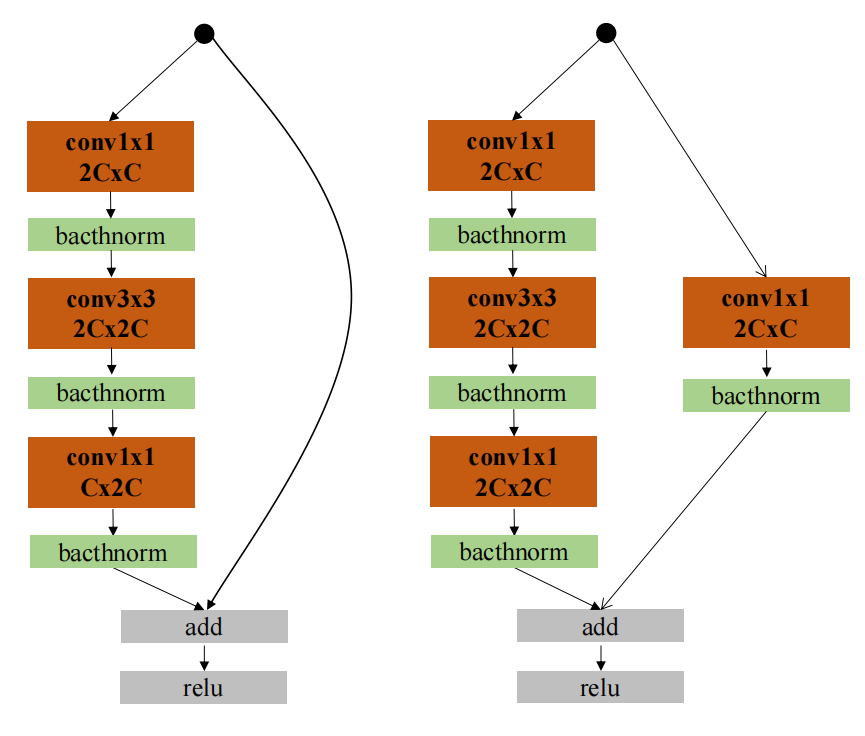
**表1所示 the theoretical FLOPs of vgg, basicblock, basicblock+, bottleneck, 和 bottleneck+ 的理论双精度浮点计算此处几乎是处于同一个水平。然而，线性拓扑结构的vgg 却有最高的理论每秒tera浮点运算(TFLOPS)，这归因于线性拓扑结构网络没有额外的内存开销。此外，在相同的理论双精度浮点数操做次数下，它优于其他多分支体系结构模型，比其他四种模型快了 17.6%—165.3%。因此，理论双精度浮点总操做次数和实际每秒双精计算次数之间的这种明显的差异可归因于两个重要因素：内存访问开销和并行度。在多 分支模型中，内存访问开销占用了很大一部分时间。正如上文中所分析到的那样，因为两个分支的计算过程中需要存储两个独立特征副本来执行独立的计算过程。**

1. **实验结论**

**综上可见，卷积神经网络在运行时的内存峰值占用和内存读写量，主要取决于输入图片像素大小、通道数和模型本身的分支数目。同时，卷积神经网络推理速度也会受到模型本身分支数目的影响。输入图片像素大小和通道数是不可控的，神经网络应具备处理各种类型规格的图片尺寸大小，因此本研究内容主要聚焦于优化模型本身，来减少卷积神经网络模型在运行时的内存占用和推理速度。**

1. **技术方案**
2. **Juggler-resnet可融合的残差结构**

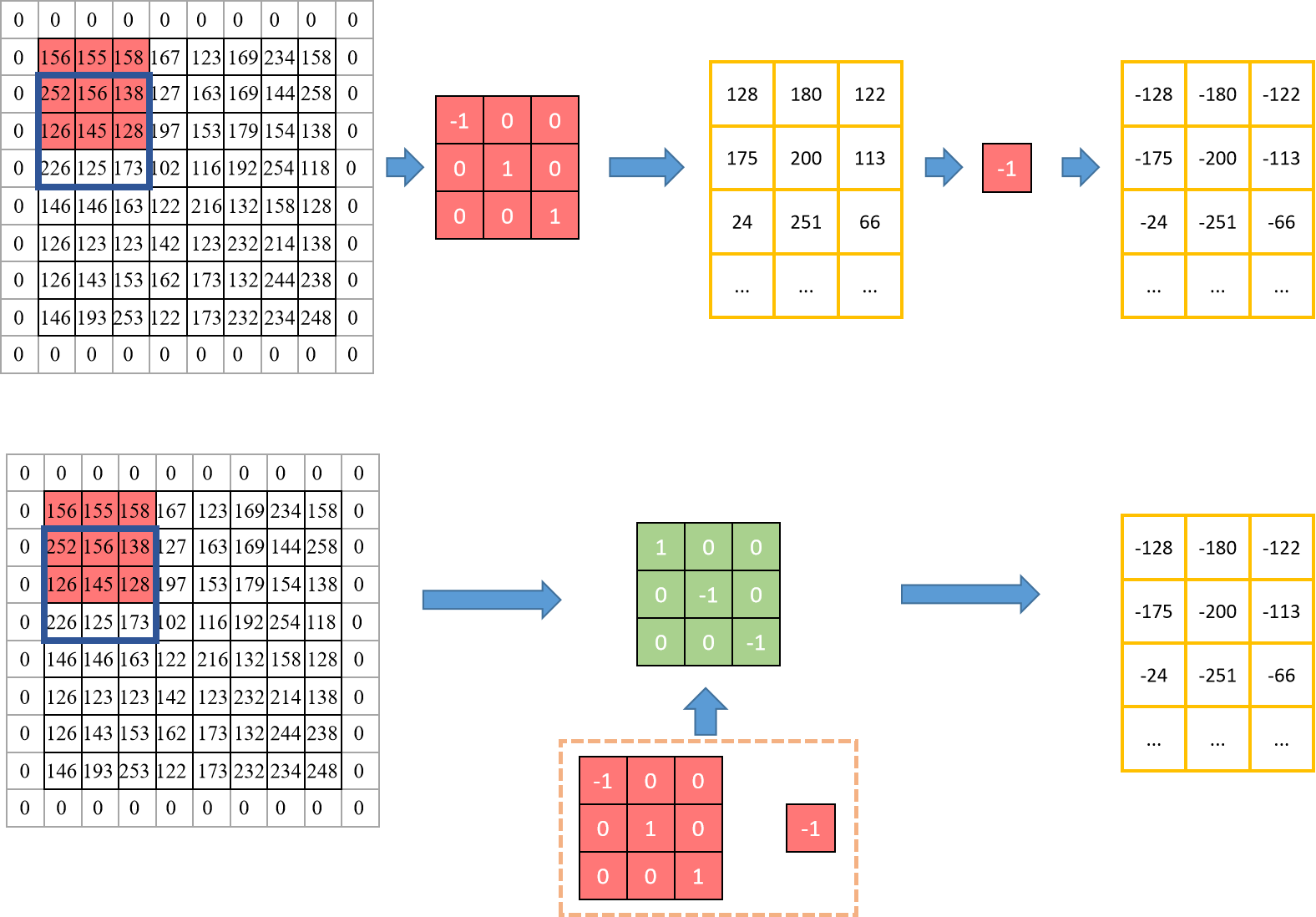
**针对上述的问题，提出了一种可融合的残差结构，在训练时是多分支在部署时可以等价的转换为线性结构，用于加速卷积神经网络的推理速度和减少神经网络运行时所产生的临时数据。**



**图3 Juggler-Resnet可融合的残差结构**

**图3展示了可融合的残差结构。它由8层网络和4种不同的运算符组成，包括convolution、batchnorm、relu和add层。值得注意的是，该结构同样也采用了两个分支的跳层连接，用以减少卷积神经网络在训练时梯度消失的问题。“双分支”的跳层连接在两个互不相邻的图层之间添加具有权重的交叉链接。其中，带有下采样的残差结构在跳层连接上还有一个convolution和batchnorm层。add操作可以放大输入中的轻微干扰，使网络训练收敛更快。同时，还移除convolution和convolution层之间的relu层以保证两个convolution层的等价融合，但保留最后一个relu层以确保结构级别的非线性关系。**

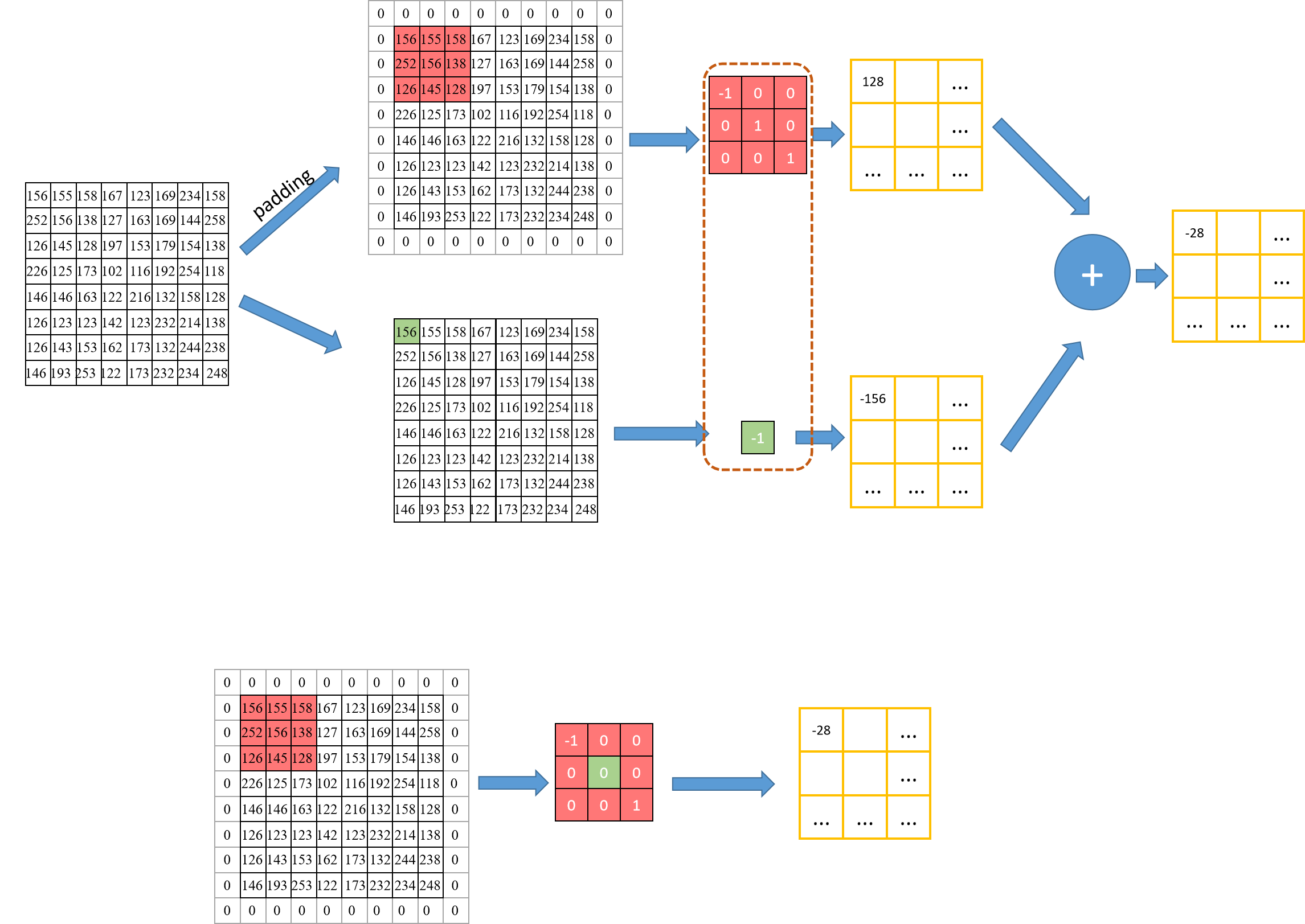
1. **1x1和3x3卷积核的合并方法**



**图4 1x1卷积核和3x3卷积核的合并方式**

**图4具体地展示了在图3结构上所实际进行的计算过程，可以看到去除掉relu层之后，1x1卷积核和3x3卷积核是可以直接合并成一个3x3卷积核，且计算结果不发生变化。实际上便是用1x1卷积核的参数去卷积核3x3卷积核，就可以得到一个等价的3x3卷积核。借此，可以使用这种方式将训练后所得到的参数等价转换为一个3x3卷积核。**

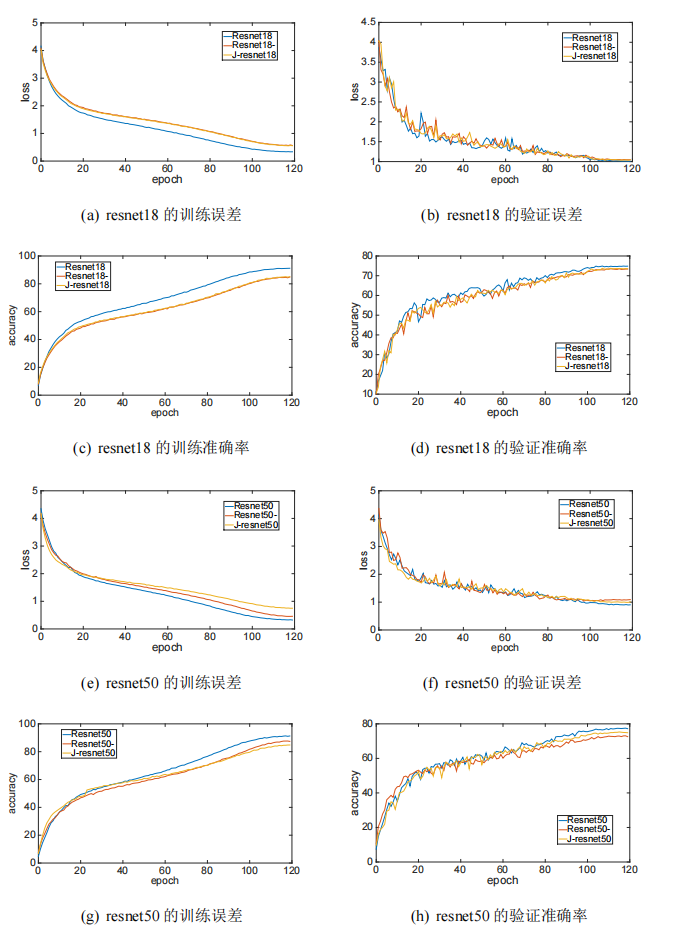
1. **Shortcut连接的合并方法**



**图5 分支运算的合并方式**

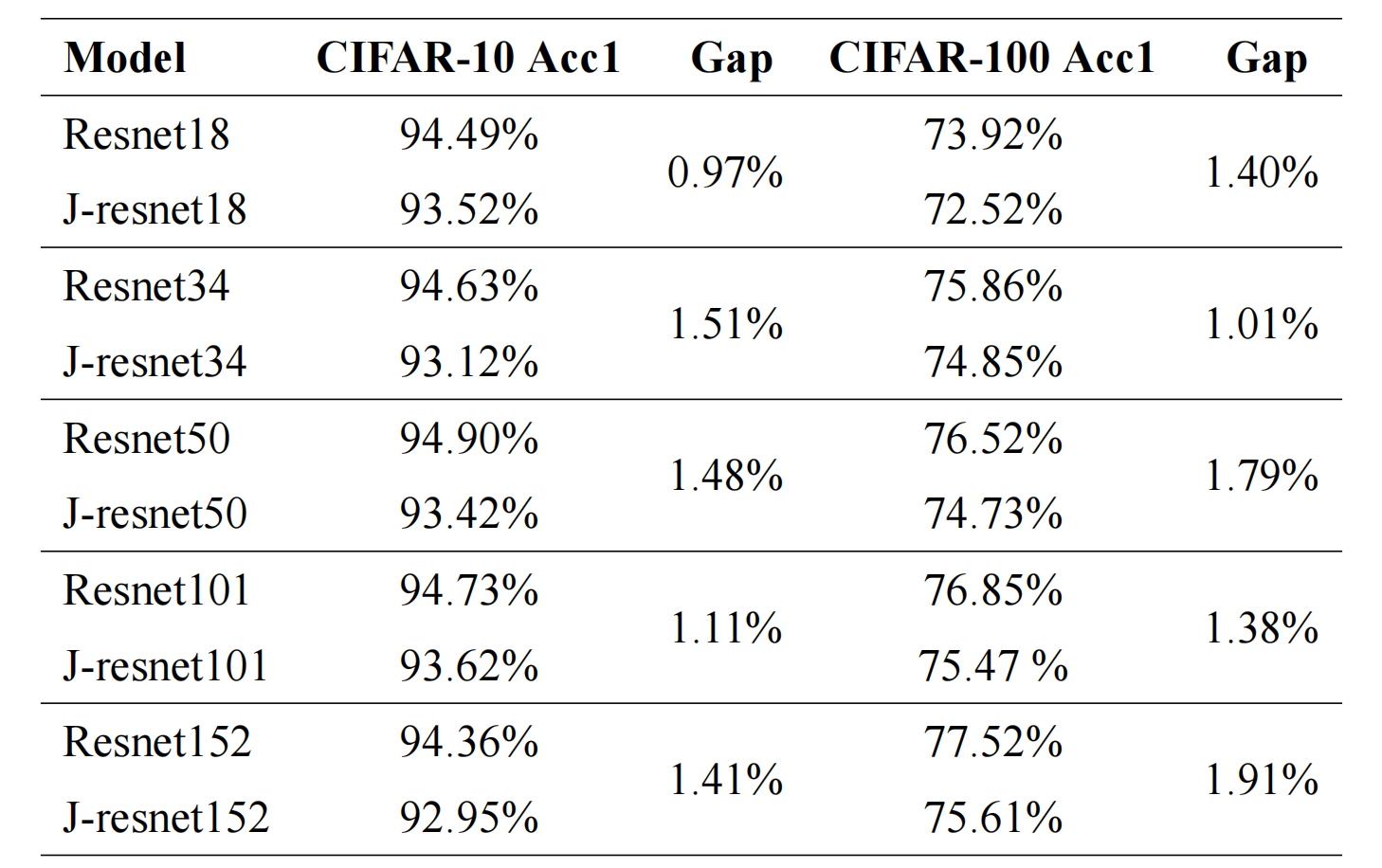
**图5具体地展示了在图3结构上所进行的实际计算过程。可以发现1x1卷积核和3x3卷积核也是可以合并的，并且其计算结果不会改变。实际上只需要将1x1卷积核扩展后，将卷积核进行直接相加，所合并后的一个单独的卷积核，该卷积核的实际计算结果等价于分支结构的计算结果。**

1. **实验结果**
2. **训练结果**



**图6 Resnet和可融合残差结构在CIFAR-100数据集上，以学习率=0.1、批量大小=256、动量系数=0.9的标准SGD，在5个epoch余弦退火的预热后，120个epoch的实验训练结果。J-resneti表示不同深度的可融合残差结构，Resnet\_i-表示两个相邻卷积核之间没有relu层插入的网络。 Train loss和 validate loss分别是在训练集和验证集上的误差。**

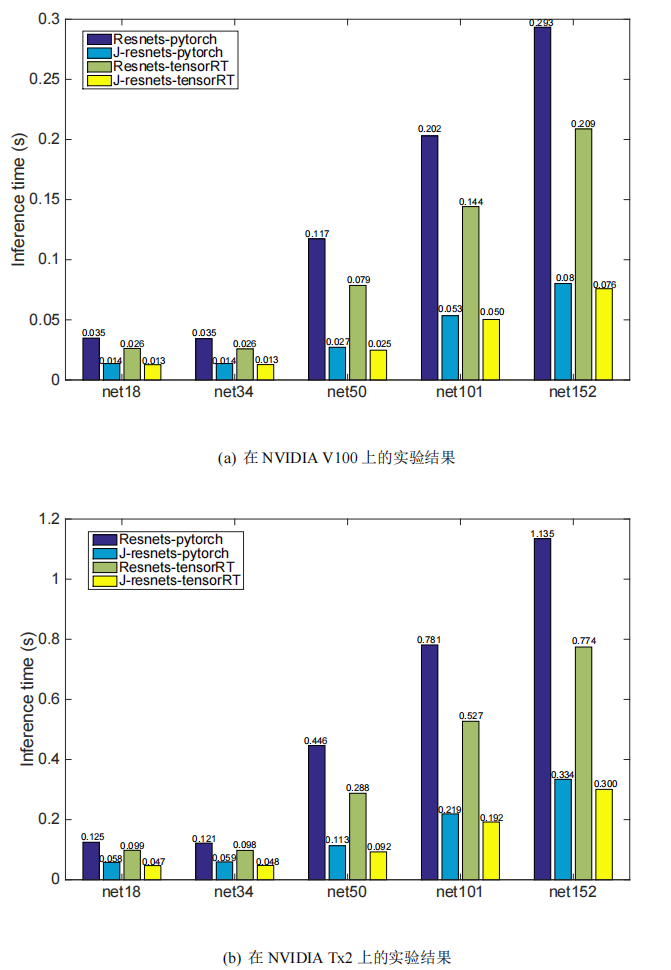
**图6(a),(b),(e),(f)的训练结果表明，可融合的残差结构可以应用于不同深度的ResNet，并且收敛速度几乎相同。图 6(c),(g),(d),(h) 表明可融合的残差结构的图片分类准确度略有下降，比原来的ResNet低约1.4%-1.79%。此外，与没有relu层的ResNet相比，可融合的残差结构的图片分类精度大约高出0.7%-0.9%百分点(即 在图中用'-'标识出来的)。这表明，在3x3卷积核处加宽通道数可以减少因移除relu层而导致的精度降低问题。值得注意的是，可融合的残差结构并没有出现梯度消失的情况，随着网络深度的增加(即18层增加到50层)在验证集上的准确率也是增多的。**



**表2 在CIFAR10和CIFAR100上的训练结果。**

**为了进一步验证Juggler ResNet在不同数据集和不同深度的ResNet上的图片分类准确率性能，我们比较了Juggler-ResNet和ResNet在不同深度范围 (从resnet18到resnet152)上的图片分类准确率。表2显示图片分类精度随深度的增加而降低。Juggler-ResNet和ResNet之间的精度差距在resnet152达到最大值，在CIFAR-10数据集上约为1.41%，在CIFAR-100数据集上约为1.91%。Juggler-ResNet的精度下降主要来自于以下两个原因。1)随着深度的增加，relu层可以通过将部分参数设置为零来减少参数之间的依赖性。2) CIFAR-100数据集的分类类别比CIFAR-10数据集更复杂。然而，Juggler-ResNet在CIFAR-10和CIFAR-100数据集上的平均性能仍然优于其他基于线性拓扑结构的模型，如vgg和alexnet，图片分类准确率分别高出10.9%和39.78%。结果表明，与基于线性拓扑结构的模型相比，Juggler-ResNet保留了多分支结构的特征提取能力，达到了SOTA级别的图片分类精度性能。**

1. **推理加速效果**

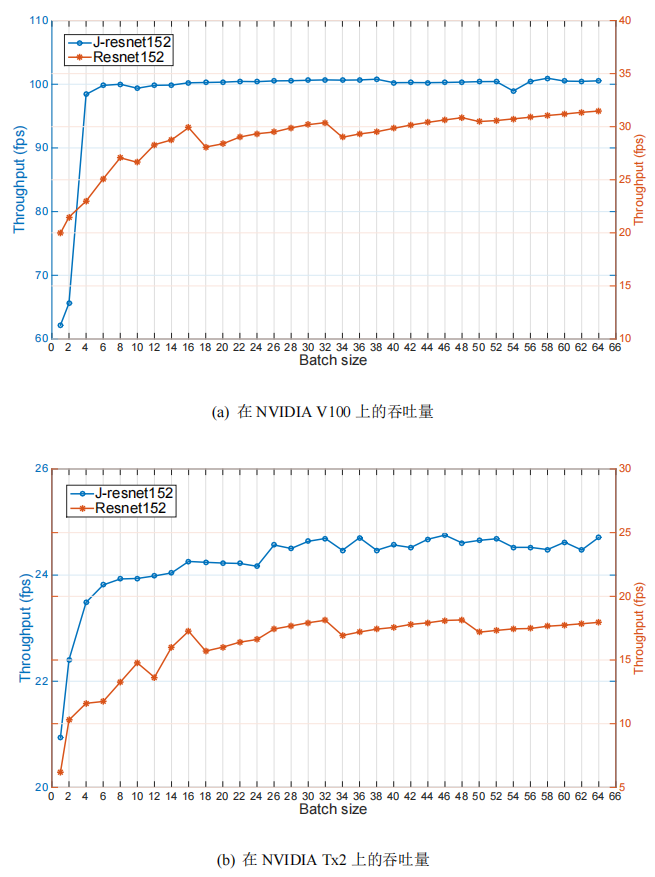


**图7 不同推理计算框架和不同硬件平台下Juggler ResNet和ResNet的推理性能比较。在预热硬件后，使用3通道256分辨率的输入图像，平均运行100次的实验结果。在Tx2上的参数设置batchsize = 1，在NVIDIA V100 GPU上的参数设置batchsize = 8。**

**图7,(b)显示与原始ResNet相比，Juggler-ResNet在NVIDIA Tx2上可实现3.95倍的加速，在NVIDIA V100上可实现4.31倍的加速。 这些结果表明，Juggler ResNet的加速效果不仅适用于硬件资源有限的平台，而且也适用于硬件资源丰富的平台。值得注意的是，加速效果随着网络深度的增加而增加。因为网络越深，内存访问、内核系统调用和上下文切换开销就越高。因此，与原始resnets相比，线性拓扑结构的简单数据流在深度网络中的优势变得更加明显。**

**此外，Juggler ResNet在不同的推理计算框架上都实现了不同程度的加速效果，如图7 (a),(b)所示。 在不同的硬件条件下，它在Pytorh框架上的加速最为显著，比原来的resnet快2.05-4.31倍。即使在主流的推理加速框架TensorRT上，Juggler-ResNet的性能也比原始的resnets高出3.17倍。值得注意的是，对比Juggler ResNet对Pytork和Tensorrt框架的加速效果，Tensorrt仅比Pytork快1.06-1.22倍，这与Pytork的推理速度几乎相同。因为所有推理计算框架在执行之前都会执行基于规则的图转换优化。这一结果表明，由于性能的提高实际上来自于它的线性拓扑结构，所以Juggler-ResNet的优化空间很少。**

1. **推理吞吐量**

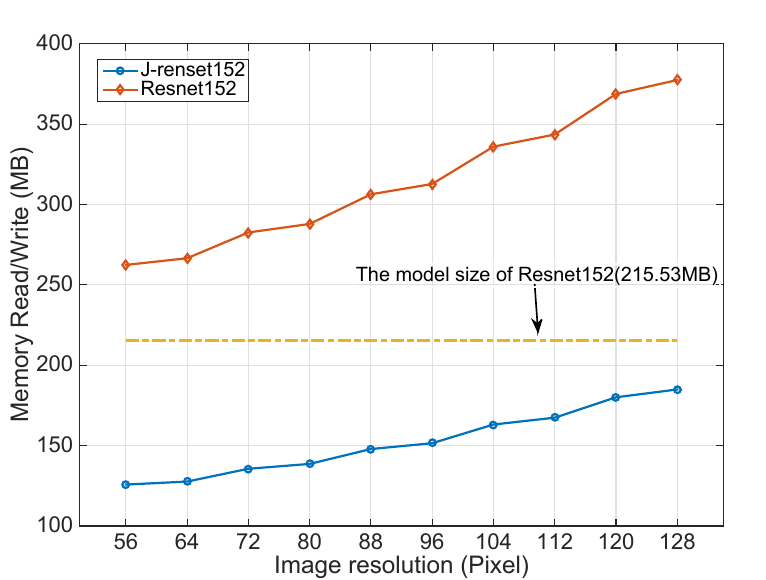


**图8 在不同的批大小设置下，不同硬件平台上的Juggler-ResNet152和ResNet152的吞吐量。在预热硬件后，平均运行100次的实验结果。在Tx2上的输入图像是3通道56分辨率， 在V100上的输入图像是3通道256分辨率**

**鉴于resnet152的参数量最多，这既可以显示优化性能的极限和硬件计算能力的拐点。因此，我们选择它作为比较模型。如图8所示，随着batchsize大小的增加，两个网络的曲线具有几乎相同的上升趋势。然而，Juggler-ResNet152在较大batchsize的大小设置下上的吞吐量明显优于ResNet152。在Tx2上，Juggler-ResNet152的峰值吞吐量是ResNet152的1.38倍。这种差距在V100上更为明显，V100上的Juggler-ResNet152的峰值吞吐量是ResNet152的3.22倍。**

**在图8中，可以观察到Juggler-ResNet152能够在不同的硬件平台上快速达到峰值吞吐量。例如，在V100上，具有线性拓扑结构的Juggler-ResNet152分别在batchsize=10和batchsize=36时达到V100和Tx2硬件平台的峰值吞吐量。然而，多分支resnet152在batchsize=64和batchsize=48时达到峰值吞吐量。此外，我们观察到，当bathsize=2^n+2时，resnet152的吞吐量会显著出现一个下降。例如，在Tx2上，bathsize=18对比bathsize=16的吞吐量减少9.93%，在V100上，bathsize=18对比bathsize=16的吞吐量减少6.51%。这使得图中resnet152的吞吐量曲线呈现出明显的锯齿状上升趋势。然而，在bathsize=2^n+2设置下，Juggler-ResNet152的总体曲线更平滑，吞吐量没有明显下降。因此，当batchsize增加时，Juggler-ResNet152比ResNet152更快地达到最佳吞吐量，并且增长更稳定。**

1. **推理期间内存读写量**



**图9 在不同图片分辨率下， Juggler-ResNet152和ResNet152推理期间的内存读写总量。**

**图9展示了resnet152和Juggler-ResNet152在推理一次期间，总的内存读写量。正如上文所分析道的那样，因Resnet152的多分支跳层连接在计算时，需要存储两份独立的feature map进行各自的运算，所以在整个推理期间，其内存读写总量明显多于单分支部署的Juggler-ResNet152。值得注意的是，Resnet模型本身只有215.53MB，但是其在运行期间所产生的临时数据读写量已经超过它本身大小。从图中展示的到那样，Juggler-ResNet152的内存读写总量正好约是ResNet152的一半左右。该实验表明Juggler-ResNet152的在推理运行时的内存读写总量明显小于ResNet152。除此之外，从图中可以明显的观察到卷积神经网络在推理期间的内存总读写量是随着图片分辨率的增加而增加的。这一现象符合了在上文中对于神经网络计算时特点分析结果。**

## 2.3 毕业论文撰写情况

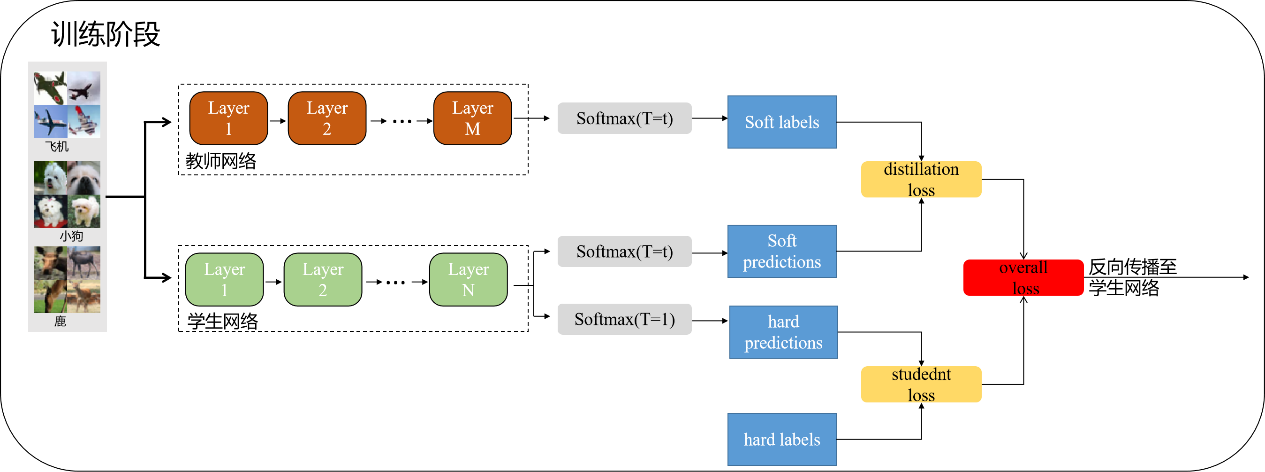
**已经完成了两个核心章节内容。**

# 主要问题及解决措施

此部分主要针对解决的关键问题和技术难点，以及采用的方案。

1. **知识蒸馏**

**针对问题3拟采用知识蒸馏的方法压缩模型尺寸。主要创新点和贡献在于分析了传统知识蒸馏框架的缺点，再此基础之上提出大-小教师网络轮番教学学生网络模式。**



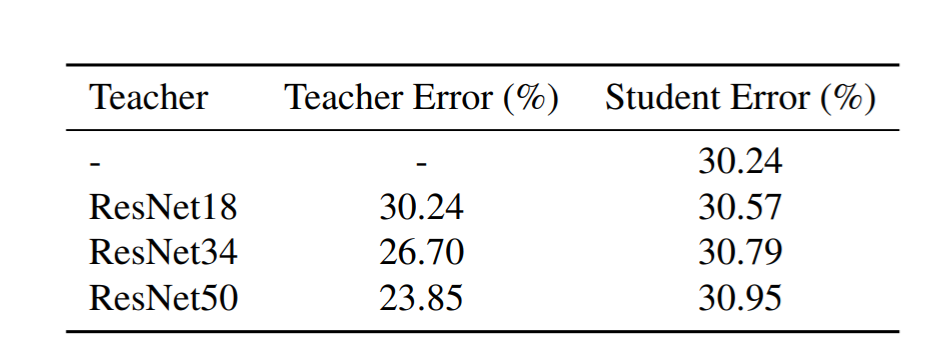
**图7 传统知识蒸馏学习框架**

**图7展示了知识蒸馏压缩神经网络的学习整体框架，其中教师网络是已经提前完全训练的大型网络，学生网络为待部署到边缘智能设备的小网络。首先，我们可以选取预训练完成的大网络作为教师网络，并选取参数量较小的一个网络作为学生网络。然后，将图片数据同时输入到教师网络和学生网络中。图片数据经过教师网络后，并由公式1 softmax(T=t)函数产生软标签。同样地，图片数据经过学生网络后，由softmax(T=t)函数产生软预测，由softmax(T=1)函数产生硬预测。最后，将软预测和软标签输入到蒸馏loss函数中生成第一个loss误差C:\Users\jason\AppData\Local\Temp\1632299213(1).png。将硬预测和硬标签输入到学生loss函数中生成第二个loss误差 C:\Users\jason\AppData\Local\Temp\1632299242(1).png。将蒸馏loss和学生loss使用公式2线性组合在一起，便生成总体loss用于反向传播更新训练参数量较小的学生网络。**

**公式1: 温度softmax (T=t) 函数. 类i的预测值pi 是由交叉熵计算得出. T是蒸馏温度. 当 T=1 时得到标准的softmax. 温度设置的越高，产生越平均的分布。**

**公式2: 总体loss误差α和β是因子.通常来说，α和β是超参。**

1. **动机实验**

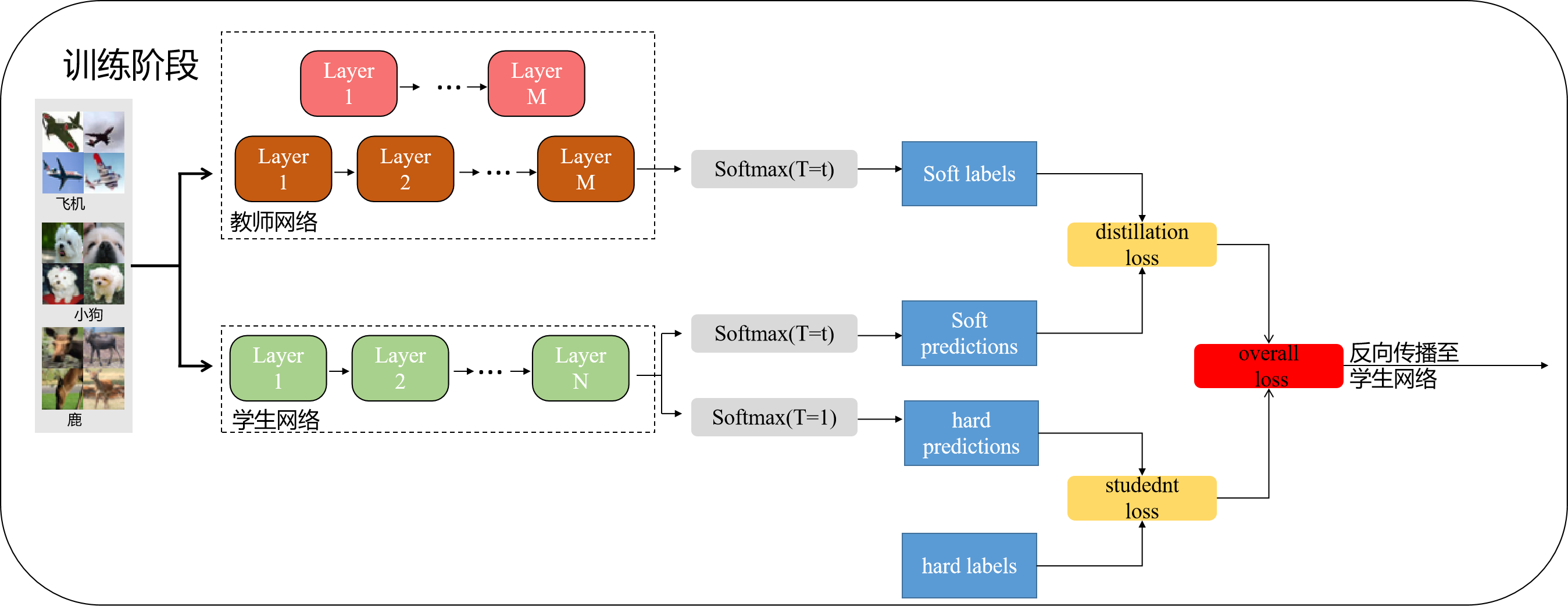


**表3 传统知识蒸馏框架下，教师网络和学生网络的实验误差**

**表3展示了传统知识蒸馏框架下，教师学生网络的训练结果。这表明并不是越大的网络就能教出越好的学生。Resnet18所教出的学生网络的错误率反倒比Resnet50所教出的学生网络更低，即便是Resnet50的网络准确度显著的高于Resnet18。**

1. **技术路线**

**针对该问题，拟采用大-小教师网络的蒸馏方法，联合蒸馏改进后的蒸馏框架如下图8所示。在传统的知识整理与框架中，增加大-小教师网络联合教学学生网络。**



**图8 大-小教师网络知识蒸馏**

1. **技术难点**
2. **大-小网络soft labels的生成策略，即采用何种方式将大-小网络的所产生的不同的soft labels 进行合并。**
3. **学生网络的学习策略，即学生网络在什么阶段采用大网络的知识又在什么阶段采用小网络的知识。**

# 下一步研究工作计划

## 4.1 未完成部分的工作开展计划

1. **针对上文所提到的技术难点，搜索调研相关论文，思考soft labels的生成策略。**
2. **开展更多实验，仔细分析在知识蒸馏中学生网络学习的具体过程，思考为什么会产生动机实验这样的结果。**
3. **根据相应的实验数据和论文调研结果，制定策略并编码验证结果**
4. **完成论文文字内容**

## 4.2 期刊、会议论文发表计划



## 4.3 毕业论文开展计划

**寒假结束前，写完已经完成课题的论文文字内容章节初稿。**

**寒假期间，调研问题3相应论文并思考相应策略。**

**寒假结束后1月，完成初步的实验代码和实验设计。**