



南開大學
Nankai University

南 开 大 学

计 算 机 学 院

深度学习及应用实验报告

深度学习及应用期末报告

林坤、崔江浩、段辰睿、徐宇昂、朱莞尔

指导教师：侯淇彬

2023 年 6 月

目录

一、 概述 1

二、 工作背景及相关工作 1

 (一) Res2net 1

 (二) GCNet 2

 (三) SKNet 2

三、 基础方法改进及实验 4

 (一) Res2net 改进：利用隐式语义增强来提高分类精度 4

 1. 改进实验 5

 2. 实验结果 5

 3. 实验结论 6

 (二) GCNet 改进 6

 1. 结合 SENet 与 GCNet 结构 7

 2. 针对 GC-SENet 进一步优化 8

 3. 结合以上两点改进 9

 4. 实验总结 9

 (三) SKNet 改进 10

 1. 原始问题 10

 2. 增加 Dropout 层 10

 3. 在 SKNet 与 VAN（Visual Attention Networks）深度学习模型相结合。 10

 4. 针对两个改进的对比消融实验 11

四、 成员贡献 12

一、 概述

本实验工作首先以 Res2net、GCNet、SKNet 为实验基础, 对 Cifar100 数据集进行图像分类实验并将结果作为实验 baseline。之后我们对每个网络进行改进方法的尝试, 包括将 Res2net 进行结构融合, GCNet 与 SENet 结合, 以及 SKNet 与 VAN 模型结合等工作, 并以此进行消融实验, 对出现的问题进一步提出创新点优化网络性能。报告按照三个相关模型介绍、基础方法改进、消融实验结果及分析、总结与成员贡献的顺序。实验模型项目见 <https://github.com/klh11111/Deeplearning-24fall>

二、 工作背景及相关工作

这里将简要介绍以下作为 baseline 复现的三篇论文提出的网络模型, 分别为 Res2net、GCNet 和 SKNet。

(一) Res2net

在图像分类、目标检测、分割等计算机视觉任务中, 网络提取多尺度特征的能力是至关重要的。当前很多 Backbone 都是在 layer-wise 这个层面做改进, 以提高网络提取多尺度特征的能力。Res2Net 提出在一个更细的粒度 (a more granular level) 中改进, 即改进 Residual Block 内部, 在几乎不改变计算量的前提下, 提高了网络提取多尺度特征的能力。

ResNet 中的 Bottleneck block 如下图中图 a 所示, Res2Net 的如图 b 所示。Res2Net 重建了普通的 ResNet 架构的 bottleneck block, 用 “4scale-(3x3)” 残差分层架构取代了标准的 1-3-1 的 CNN 布局。因此, 这个更改构建了 “Res2Net”。这样, 中间的主卷积从单分支变为多分支。

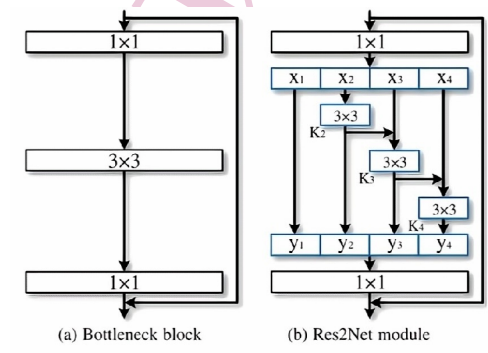


图 1: 残差块结构 (来源于 Res2net: A new multi-scale backbone architecture)

Res2Net 模块不仅可以无缝替换原 ResNet 中的残差结构, 还可以与多种结构融合, 形成更强的 Backbone。比如可以将 Res2Net 模块与 ResNeXt、SE 结合: 参照 ResNeXt, 将每组中的卷积改为分组卷积; 在残差连接前加入 SE block。形成的网络结构如下图所示:

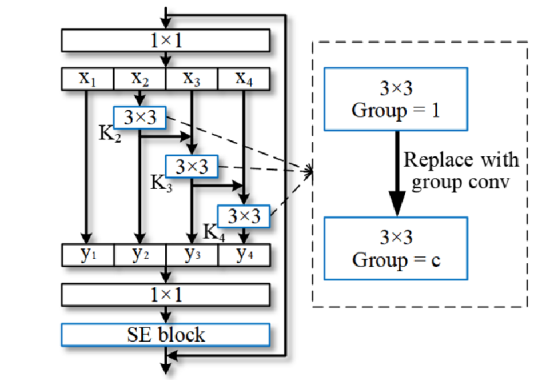


图 2: 结合分组卷积和 SE 块的残差结构 (来源于 Res2net: A new multi-scale backbone architecture)

(二) GCNet

GCNet (Global Context Net) 是基于 NLNet (Non-Local Net) 的改进版。GCNet 相比于 NLNet 的优点主要在于节省了许多不必要的计算并且减少了许多参数。

首先介绍一下 NLNet。该网络提供了一种捕获远程依赖关系的方法, NLNet 通过将特定于查询的全局上下文聚合, 得到每个查询的位置。具体来讲, NLNet 对于每一个像素点都要计算一次该像素点与其他像素点的相关性, 再根据相关性将其他像素点的信息按照权重加到该像素点上。权重越大, 说明对于该像素点来说, 对应的区域越重要。

然后介绍 SENet。该网络考虑了特征通道之间的关系, 对特征通道之间的相互依赖关系进行了建模。具体来讲, 就是通过学习的方式来获取每个特征通道的重要程度, 然后根据这个重要程度去提升有用的特征并抑制对当前任务用处不大的特征。

然后介绍 GCNet。GCNet 的作者通过实验发现 NLNet 中每一个像素点与其他像素点的相关性几乎是相同的, 也就是每个像素点的 attention map 几乎是一样的。这说明其实不需要针对每一个像素点都计算一个 attention map, 可以计算出一个全局的 attention map, 然后对于所有位置共享这个全局的 attention map。这样的优点在于减少了许多计算量。然后 GCNet 的作者为了捕获各通道间的依赖关系又参考了 SENet 的结构。作者将简化后的 NLNet 中的 1×1 卷积层改为了 SENet 中的 bottleneck transform 结构, 然后就获得了 GCNet。这样既可以获取通道间的依赖关系, 又可以通过 bottleneck transform 结构减少参数量和计算量。综上, GCNet 充分结合了 Non-local 全局上下文建模能力强和 SENet 节省计算量的特点。

(三) SKNet

在传统的卷积神经网络 (CNN) 中, 每一层的人工神经元的感受野都被设计为固定的大小。这一设计模式是基于人工模拟对视觉神经元的理解。然而, 实际上, 在自然界中, 生物视觉皮层神经元的感受野大小并不是固定的, 而是会根据接受的刺激进行调节。这样的现象在标准的 CNN 设计中并未得到充分的考虑, 因此可能会在某些情况下导致特征捕捉的不准确。

为了解决这个问题, 作者在这篇文章中提出了一种全新的动态选择机制, 这种机制允许每个神经元根据输入信息的多个尺度自适应调整其感受野的大小。这一机制的实现主要依赖于作者设计的一种新型的构造块, 也就是选择内核 (Selective Kernel, 简称 SK) 单元。

在 SK 单元中, 作者构建了多个具有不同内核大小的分支, 这些分支使用 softmax attention 机制进行融合。融合的过程是由分支中的信息进行引导的, 也就是说, 每个分支被赋予了不同的权重, 这些权重由信息的重要性决定。因此, 对不同分支的关注程度不同, 导致了融合层神经元

的有效接受野大小也有所不同。在实现过程中，作者将多个 SK 单元堆叠到一起，形成了一种深度神经网络结构，这种结构被称为选择性内核网络（Selective Kernel Networks，简称 SKNet）。

SKNet 的主要特性在于，它的神经元能够根据输入自适应地捕获不同尺度的目标对象。这种特性验证了作者设计的神经元具有根据输入自适应调整感受野大小的能力，这一点在传统的 CNN 中是无法实现的。总的来说，SKNet 为深度学习模型提供了一个全新的视角，有助于更准确地捕捉和处理视觉信息。

Selective Kernel Networks 结构如图3所示：

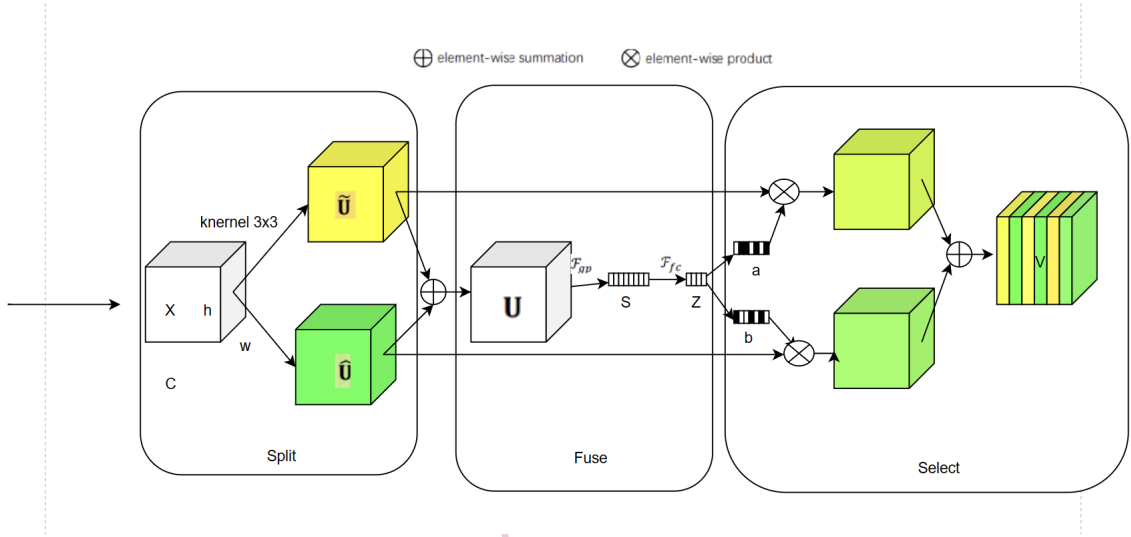


图 3: SKnet 结构示意图

1.Split

对输入的特征图 $\mathbf{X} \in \mathbb{R}^{H' \times W' \times C'}$ ，分别进行两次卷积变换 $\tilde{\mathcal{F}} \hat{\mathcal{F}}$ 。

(1) $\tilde{\mathcal{F}}: \mathbf{X} \rightarrow \tilde{\mathbf{U}} \in \mathbb{R}^{H \times W \times C}$: $\tilde{\mathcal{F}}$ 过程中卷积核大小为 3×3 ，特征图 \mathbf{X} 经过卷积变换为 $\tilde{\mathbf{U}}$ 。

(2) $\hat{\mathcal{F}}: \mathbf{X} \rightarrow \hat{\mathbf{U}} \in \mathbb{R}^{H \times W \times C}$: $\hat{\mathcal{F}}$ 过程中卷积核大小为 5×5 ，特征图 \mathbf{X} 经过卷积变换为 $\hat{\mathbf{U}}$ 。

特征图 \mathbf{X} 经过 Split 操作，输出两个新的特征图元、 $\hat{\mathbf{U}}$ 。

2.Fuse

目的：使神经元能够根据刺激内容自适应地调整其 RF 大小，因此需要使用门来控制来自多个分支的信息流，这些分支携带不同规模的信息进入下一层的神经元。

方法：使用门整合来自所有分支的信息，也就是将来自多个分支的特征进行融合。

步骤：

(1) $\mathbf{U} = \tilde{\mathbf{U}} + \hat{\mathbf{U}}$: 特征图 $\tilde{\mathbf{U}}$ 、 $\hat{\mathbf{U}}$ 相加，得到新的特征图 \mathbf{U} ， \mathbf{U} 中融合了多个感受野的信息。

(2) \mathcal{F}_{gp} : \mathbf{U} 通过全局平均池化生成 $\mathbf{s} \in \mathbb{R}^C$ 来嵌入全局信息， $\mathbf{s} \in \mathbb{R}^{C \times 1}$ 是一个有 C 个元素的列向量。对应的算子公式如下：

$$s_c = \mathcal{F}_{gp}(\mathbf{U}_c) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W \mathbf{U}_c(i, j)$$

通过将 \mathbf{U} 的第 C 个 feature map 缩小空间尺寸 $H \times W$ 来计算得到 \mathbf{s} 的第 C 个元素。

(3) \mathcal{F}_{fc} : 通过一个简单的全连接 (fc) 层，将向量 \mathbf{s} 压缩为特征向量 $\mathbf{z} \in \mathbb{R}^{d \times 1}$ 。对应的算子公式如下：

$$\mathbf{z} = \mathcal{F}_{fc}(\mathbf{s}) = \delta(\mathcal{B}(\mathbf{W}\mathbf{s}))$$

δ 是 ReLU 函数, \mathcal{B} 表示批量归一化, $\mathbf{W} \in \mathbb{R}^{d \times C}$ 为权重矩阵。很明显这里, 向量 \mathbf{s} 先是通过了一个全连接层将 c 个通道变成 d 个, 减少参数量, 再经过批量归一化函数, 最后通过 ReLU 函数得到特征向量 \mathbf{z} 。

(4) 论文中还研究了 d 对模型效率的影响, 其算子公式如下:

$$d = \max(C/r, L),$$

这里 C/r , 可以看出 SENet 论文的痕迹, 目的与 SENet 论文中的一致, 因为一个全连接层无法同时应用 relu 和 sigmoid 两个非线性函数, 但是两者又缺一不可。为了减少参数, 所以设置了 r 比率。

3.Select

目的: 在紧凑的特征描述符 \mathbf{z} 的引导下, 利用跨通道的软注意来自适应地选择不同的信息空间尺度。

方法: softmax 运算符应用于通道数, 得到各分支上特征图的软注意力向量。这里的特征图示例为 $\tilde{\mathbf{U}}, \hat{\mathbf{U}}$, 因此得到软注意力向量为 $\tilde{\mathbf{U}}, \mathbf{U}$ 的软注意力。

步骤:

(1) 分别两次对特征向量 \mathbf{z} 使用 softmax 函数得到软注意力向量 \mathbf{a}, \mathbf{b} , 这时向量中的每一个数值对应一个 channel 的分数, 代表其 channel 的重要程度, 同时将 \mathbf{a}, \mathbf{b} 再次变回了 c 个维度, 这里又可以看出 SENet 论文的痕迹。算子公式如下:

$$a_c = \frac{e^{\mathbf{A}_c \mathbf{z}}}{e^{\mathbf{A}_c \mathbf{z}} + e^{\mathbf{B}_c \mathbf{z}}}, b_c = \frac{e^{\mathbf{B}_c \mathbf{z}}}{e^{\mathbf{A}_c \mathbf{z}} + e^{\mathbf{B}_c \mathbf{z}}}$$

其中 $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{C \times d}$, \mathbf{a}, \mathbf{b} 分别表示 $\tilde{\mathbf{U}}$ 和 $\hat{\mathbf{U}}$ 的软注意力向量。请注意, $\mathbf{A}_c \in \mathbb{R}^{1 \times d}$ 是 \mathbf{A} 的第 c 行, a_c 是 \mathbf{a} 的第 c 个元素, \mathbf{B}_c 和 b_c 也是如此。

(2) 各特征图与对应的注意力权重相乘, 再相加, 最终得到特征图 \mathbf{V} 。

$$\mathbf{V}_c = a_c \cdot \tilde{\mathbf{U}}_c + b_c \cdot \hat{\mathbf{U}}_c, \quad a_c + b_c = 1$$

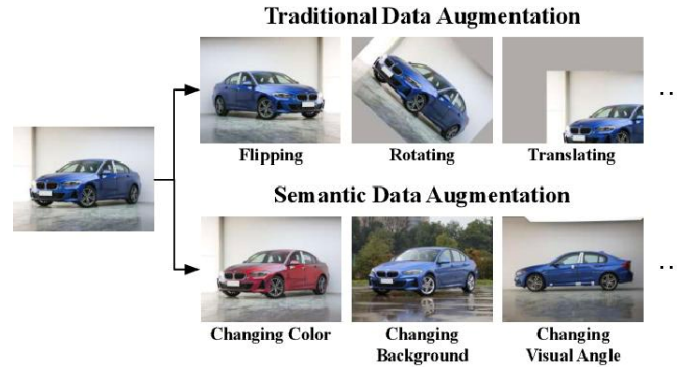
其中 $\mathbf{V} = [\mathbf{V}_1, \mathbf{V}_2, \dots, \mathbf{V}_C]$, $\mathbf{V}_c \in \mathbb{R}^{H \times W}$ 。这个特征图 \mathbf{V} 是通过各种内核上的注意力权重获得的, 融合了多个感受野的信息, 具有了多尺度的信息。

ResNext-50、SENet-ResNext-50 与 SKNet-ResNext-50 三种网络结构进行对比。SENet 通过在完成 $(1 \times 1$ 卷积 $+ 3 \times 3$ 卷积 $+ 1 \times 1$ 卷积) 的卷积操作后直接加入全连接层, 学习通道间的依赖关系, 然后将所学习的通道权重应用回原向量。SKNet 的设计是将 ResNext 中的 3×3 卷积部分替换为两个或多个不同大小的卷积核的卷积操作, 同时加上学习通道权重的全连接层。然后, 输出向量继续进行 1×1 卷积操作。从参数数量角度来看, 因为模块嵌入位置的差异, SKNet 的参数数量与 SENet 相当, 或者略少。相对的, 计算量也有少许增加, 尽管带来的精度提升远低于增加的计算量成本。

三、基础方法改进及实验

(一) Res2net 改进: 利用隐式语义增强来提高分类精度

我们的工作则提出了一种隐式语义数据扩增算法: ISDA, 意在实现对样本进行更为「高级」的、「语义」层面的变换, 例如改变物体的背景、颜色、视角等。



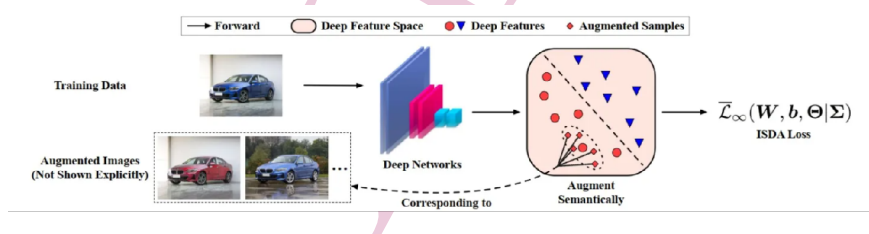
ISDA 具有几个重要的特点:

- 与传统数据扩增方法高度互补;
- 巧妙地利用深度神经网络长于学习线性化表征的性质, 在特征空间完成扩增过程, 无需训练任何辅助生成模型 (如 GAN 等);
- 直接优化无穷扩增样本期望损失的一个上界, 最终形式仅为一个全新的损失函数;
- 可以广泛应用于全监督、半监督图像识别、语义分割等视觉任务, 在较大规模的数据集上效果比较明显;

1. 改进实验

在 CIFAR100 数据集上进行分类识别任务。

具体实验设置: 为了保证实验的公平性, 模型的超参数设置保持一致, 如下:



- dataset: cifar100
- epochs: 300
- lr: 0.1
- milestones: 150,225
- gamma: 0.1
- momentum: 0.9
- weight_decay: 0.0005
- nesterov: False

2. 实验结果

骨干网络为 Res2NeXt29_6c×24w×4scale 的实验结果:

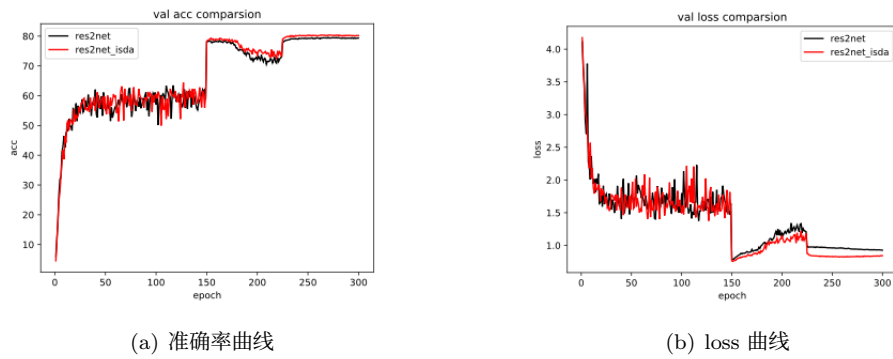


图 4: 实验结果

骨干网络为 Res2NeXt29_8c×25w×4scale 的实验结果:

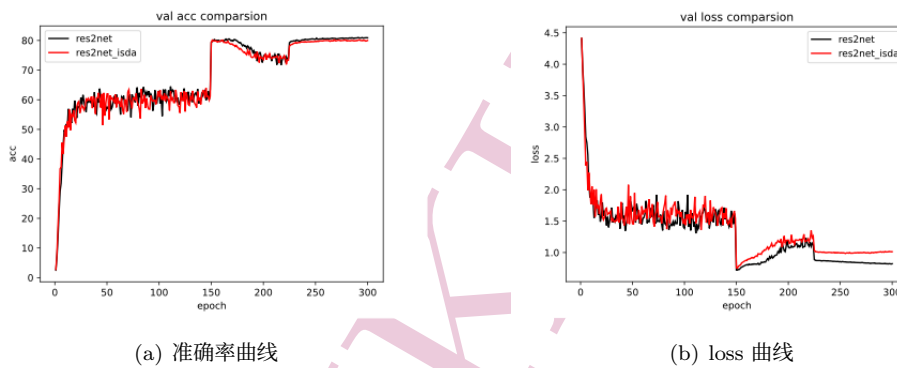


图 5: 实验结果

3. 实验结论

我们的改进方法在测试集的模型精度和损失上面均有明显的提升效果, 这说明了隐式语义增强技术能够在增加较小模型参数的前提下在一定程度上在深度特征空间对类别特征进行数据增强, 从而改善模型性能。

(二) GCNet 改进

我们选择个人实现的 miniResNet(ResNet-12) 和 ResNet-18 作为基底网络结构 (所采用的残差块均为两层 3×3 卷积的浅层 ResNet 块), 在此基础上实现三种基础方法的复现。而对于改进方法, 我们选择对实现了 GCNet 的精简 ResNet 作为优化对象, 其结构如下表:

Output	GC-ResNet12	GC-ResNet18
32×32	3×3, 8, stride 1	3×3, 64, stride 1
	BN, 8	BN, 64
	GCBlock, 8, r=0.25	3×3, 64
		3×3, 64
	3×3, 8	GCBlock, 64, r=0.25
	3×3, 8	
	3×3, 8	3×3, 64
	3×3, 16, stride 2	3×3, 64
	GCBlock, 16, r=0.25	3×3, 64
		3×3, 128, stride 2
	3×3, 16	GCBlock, 128, r=0.25
	3×3, 16	
	3×3, 16	3×3, 128
	3×3, 32, stride 2	3×3, 128
	GCBlock, 32, r=0.25	3×3, 128
		3×3, 256, stride 2
	3×3, 32	GCBlock, 256, r=0.25
	3×3, 32	
	(GC-ResNet12) 4×4	4×4 global average pool, 512-100 fc
		3×3, 256
		3×3, 256
		3×3, 512, stride 2
		GCBlock, 512, r=0.25
		3×3, 512
(GC-ResNet18) 4×4		3×3, 512
		4×4 global average pool, 512-100 fc

图 6: miniResNet-12 和 ResNet-18 结构（来源于 GCNet: Non-local Networks Meet Squeeze-Excitation Networks and Beyond）

1. 结合 SENet 与 GCNet 结构

在阅读 GCNet 的论文时不仅了解了 GCNet, 还学习到了 NLNet 和 SENet 的相关知识。这三种网络结构都可以捕获远程依赖关系, 但是原理和利用的维度不同。NLNet 利用的是空间维度的依赖关系, SENet 利用的是特征通道之间的关系, 而 GCNet 同时利用了空间维度和特征通道的依赖关系。三种网络结构的使用方法也不同, GCNet 和 NLNet 可以插入在两个 ResNet block 之间（通常是插在 ResNet 的每个 stage 的最后一个 block 之前）, 而 SENet 则可以嵌入到 ResNet block 中形成 SE-ResNet block。

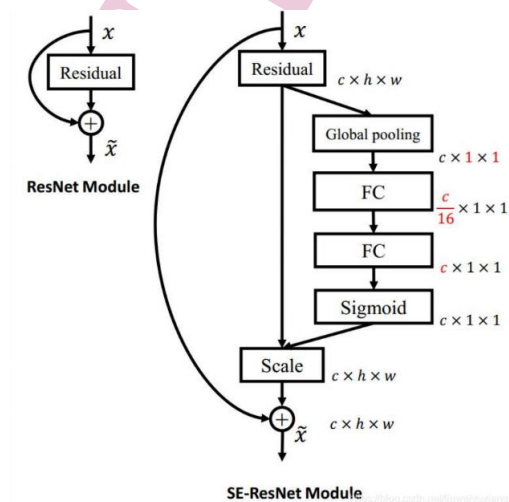


图 7: SE-ResNet block 示意图(来源于 <https://blog.csdn.net/liuweiyuxiang/article/details/84075343>)

由于这三种网络有着以上的差异和联系, 我们提出了一种改进方法: 使用 SE-ResNet block 替换基础的 ResNet block。这种改进方法的理论依据在于可以更好的建立起远程的依赖关系, 并且由于 SENet 具有轻便、计算量小的优点, 不会明显增大整个神经网络的参数量和计算量。而且由于 GCNet 的使用方法是添加在 ResNet block 之间, 而 SE-ResNet block 的使用方法是替换基础的 ResNet block, 这使得两者可以更加有效的结合在一起。

实验结果

Method	ResNet12(top1)	12-Model Size(M)	ResNet18(top1)	18-Model Size(M)
GCNet	43.0%	0.095M	54.9%	11.31M
SENet	43.6%	0.095M	54.75%	11.4M
GC-SENet	43.85%	0.096M	54.2%	11.58M

表 1: GCNet 与 SENet 结合 top1 实验结果 (Epoch = 10)

如表所示, 可以看到在 ResNet12 上将 SENet 与 GCNet 结合之后的效果要好于单独使用一种网络结构的效果。虽然参数量有所增长, 但是增长的比例很小。而在 ResNet18 上的效果反而更差, 我们认为这可能是出现了过拟合现象, 这部分将在下一节中详细讨论。

2. 针对 GC-SENet 进一步优化

在上面 SENet 和 GCNet 结合的尝试性实验中, 训练时出现了过早过拟合的现象; 从本次实验结果也可以看到, 相较于网络结构更简单的 ResNet12, ResNet18 在结合了 SENet 后对 cifar100 验证集的准确率不升反降。我们推测是简单结合 SENet 和 GCNet 使得网络结构复杂, 导致容易出现过拟合现象。在这一节中针对这一问题, 提出了两点改进:

将 GC Block 插入不同 SENet 的 stage 前

我们抛弃了简单地将 SENet 网络和 GCNet 网络结构结合的策略, 考虑到网络存在过拟合现象并且可能有网络过于复杂的问题, 我们不在每一个 SE Block 前都加入 GC Block, 而是选择性在不同 SENet stage 前添加 GC Block 且仅添加一个。

为了探究此种方法对网络性能的改进效果, 并探究 GC Block 在不同 SENet stage 中的不同作用, 我们做了以下消融实验: 以两种网络为基底 (ResNet12 和 ResNet18) 在不同 stage 前加入 GC Block (12 有 3 个 stage, 18 有 4 个 stage)。实验结果及各模型参数量如下表所示:

Method	ResNet12(top1)	12-Model Size(M)	ResNet18(top1)	18-Model Size(M)
GCNet	43.0%	0.095M	54.9%	11.31M
SENet	43.6%	0.095M	54.75%	11.4M
GC-SENet	43.85%	0.096M	54.2%	11.58M
GC-SENet(Stage1)	43.3%	0.0944M	54.72%	11.32M
GC-SENet(Stage2)	43.5%	0.0945M	-	-
GC-SENet(Stage3)	44.83%	0.095M	-	-
GC-SENet(Stage4)	-	-	55.8%	11.30M

表 2: GC Block 插入不同 stage 中

从实验结果来看, 无论是 ResNet12 还是 18, 将 GC Block 插入 SENet 的最后一个 stage 中 (ResNet12-stage3 和 ResNet18-stage4) 可以得到最好的预测效果, 并且预测准确率均要好于 GC-SENet 的简单结合网络。

在 GC Block 和 SE Block 间加入适当 dropout 层

除了上面提出的方法, 针对前文提到的过拟合问题, 我们尝试在 GC Block 和 SE Block 间加入 dropout 层改善过拟合现象, 提升网络性能。我们使用 drop out 率为 0.15% 的 nn.Dropout(0.15) 添加在 GC 和 SE Block 块间, 实验结果如下表所示:

从表中实验结果可以看到, 在 GC 和 SE Block 块间增加 dropout 层后, 在 ResNet12 和 ResNet18 上均带来更好的准确率效果, 且均优于 GCNet 和 SENet。

Method	ResNet12(top1)	12-Model Size(M)	ResNet18(top1)	18-Model Size(M)
GCNet	43.0%	0.095M	54.9%	11.31M
SENet	43.6%	0.095M	54.75%	11.4M
GC-SENet	43.85%	0.096M	54.2%	11.58M
GC-SENet(dropout)	44.1%	0.096M	55.1%	11.58M

表 3: 在 GC 和 SE Block 块间增加 dropout 层

3. 结合以上两点改进

通过结合以上两点改进方法，在 SENet 最后的 stage 前插入 GC Block，并在块间增加 dropout 层，实验结果如下表所示：

Method	ResNet12(top1)	12-Model Size(M)	ResNet18(top1)	18-Model Size(M)
GCNet	43.0%	0.095M	54.9%	11.31M
SENet	43.6%	0.095M	54.75%	11.4M
GC-SENet	43.85%	0.096M	54.2%	11.58M
GC-SENet(last stage+dropout)	46.14%	0.096M	55.6%	11.50M

表 4: 结合两点改进方法与 baseline 结果对比

4. 实验总结

最开始尝试将 GCNet 与 CANet 结合，但通过实验结果来看效果并不理想，于是我们选择了利用特征通道之间的关系的 SENet 再次尝试与 GCNet 结合，这是因为 GCNet 同时利用了空间维度和特征通道的依赖关系，相较于 CANet 来说 SENet 更适合。

在尝试将 GCNet 与 SENet 结合之后，我们发现网络模型在训练过程中存在过拟合现象，实验结果在 ResNet12 上略有提升而在 ResNet18 上的准确率不升反降，于是我们对与 SENet 结合后的 GCNet 做进一步改进。

考虑到在网络更加复杂的 ResNet18 上的效果更差以及存在过拟合的现象，我们提出了两点针对 GC-SENet 的改进方法，在 SENet 的最后 stage 前插入 GC Block。如下图 4 所示，可以看到最后 stage 前的 GCBlock 对网络性能的提升作用是最大的。除此之外另一个改进则在 SE 和 GC Block 增加 dropout 层缓解过拟合现象，同样对网络性能带来提升。

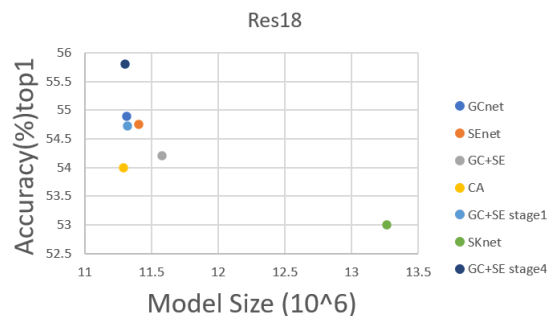


图 8: 改进网络与复现的三种 baseline 比较图

所以最终我们选择将 GCNet 和 SENet 结合，并同时使用提出的两点改进方法，得到预测性能最优的 GC-SENet(last stage+dropout) 网络。在 miniResNet12 上的分类准确率达到 46.14%，

在 ResNet18 上的准确率达到 55.6%。比改进前的基础网络 SENet 和 GC-SENet 准确率高，并且相较于作为 baseline 的 SKNet、CANet、GCNet 三种复现方法，准确率均要高出 2 3%。

(三) SKNet 改进

1. 原始问题

复现上述 SKNet 模型，并在 CIFAR-100 数据集上的准确率曲线和损失函数曲线：

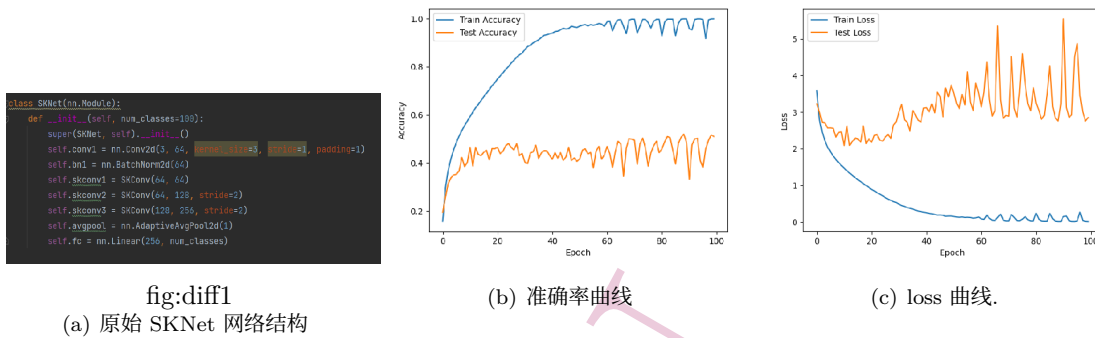


图 9: 原始 SKNet 的准确率和 loss 曲线

很明显，原始 SKNet 在 CIFAR-100 数据集上产生了严重的过拟合现象，因此为解决过拟合现象，我们可以入 Dropout 层。

2. 增加 Dropout 层

增加 Dropout 层，随机忽略部分神经元的输出，可以提高模型的鲁棒性。改进后的 SKNet 在 CIFAR-100 数据集上的准确率曲线和损失函数曲线：

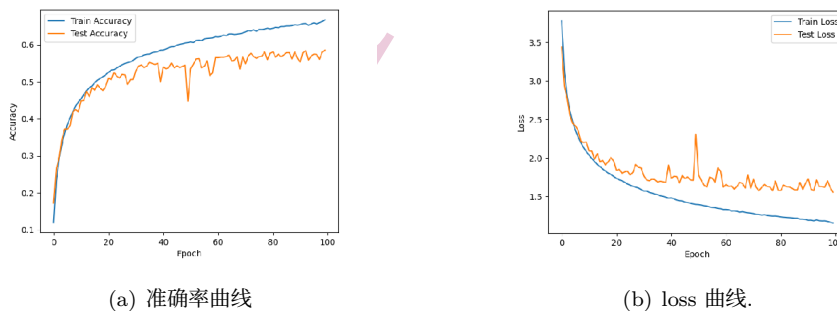


图 10: 改进 1 的准确率和 loss 曲线

可以发现，消除过拟合现象后，准确率达到 70%。

3. 在 SKNet 与 VAN (Visual Attention Networks) 深度学习模型相结合。

SKNet 的动态选择机制虽然能够对不同尺度的输入信息进行自适应的调整，但其本质上还是一种局部调整策略，即在每个 SK 单元中，神经元的感受野调整都是独立进行的。这样就可能忽略了图像中各部分之间的全局关系，例如一些结构性特征和上下文信息。此外，SKNet 的计算

复杂度较高，多个不同大小的内核需要独立计算，然后通过 softmax attention 进行融合，这使得 SKNet 在一些需要实时反馈的场景下可能无法满足需求。

结合视觉注意力网络 (VAN) 可以有助于解决这些问题。视觉注意力机制可以模拟人类视觉的特性，根据场景的复杂度和任务的需要，有选择性地聚焦于图像的一部分，忽略其他部分，这使得网络能够专注于关键信息的处理。这样不仅能有效减少模型的计算复杂度，同时还能通过全局的上下文信息，增强网络对于复杂场景的理解和处理能力。

因此，将 SKNet 与 VAN 相结合，可以充分利用 SKNet 的自适应感受野调整能力和 VAN 的视觉注意力机制，对局部特征和全局上下文信息进行有效的整合，从而提升模型的表现。具体来说，SKNet 可以在局部范围内对特征进行精细的调整，而 VAN 则可以帮助模型关注更具决策价值的区域，从而在保持 SKNet 优良性能的同时，降低模型的计算复杂度，增强模型的解决问题的全局视野。

改进 2-1：并行结合——SKConvAttention 模块：

在这种结构设计中，我们在 SKConv 模块生成特征映射之后，立即接入了视觉注意力机制 (Visual Attention Block)。这种设计策略意味着每个 SKConv 模块的输出都会经过视觉注意力机制的筛选和调整，每一个特征都会根据其重要性得到适当的权重分配。

此种并行结构的好处在于，它可以确保每一层的输出都经过了视觉注意力的精细调整，使得不同的特征根据其对最终任务的贡献大小得到不同程度的重视。换句话说，这种策略让网络在整个前向传播过程中，始终将每个特征的重要性纳入考虑，让模型更加聚焦于对结果影响较大的关键信息。因此，这种 SKConv 和视觉注意力机制的并行结合方式，能够提升模型对复杂数据的解析能力，有助于提高模型的性能。

改进 2-2：序列结合——CombinedNet 模型：

在这种设计框架中，我们首先利用一系列的 SKConv 模块对输入信息进行特征提取，然后在所有的 SKConv 模块完成特征提取后，再整体地应用视觉注意力机制 (Visual Attention Block)。换言之，视觉注意力机制的应用是在所有的 SKConv 模块之后的整体级别进行的。

这样的设计策略的优点在于，它能够在全局的视角下一次性地考虑所有特征的重要性。相对于在每个 SKConv 模块之后都分别进行注意力调整，这种方法可以对所有特征进行整体的权重分配和调整，更好地捕捉和利用特征之间的依赖和关联性。

在这种结构中，SKConv 模块负责从各个尺度上捕获丰富的特征，而视觉注意力机制则对这些特征进行全局的重要性评估和调整，使得模型的决策更加注重全局上下文信息，同时也减少了过度关注局部特征的可能性。因此，CombinedNet 模型能够有效整合 SKConv 模块和视觉注意力机制的优点，进一步提升网络对复杂视觉任务的处理能力。

4. 针对两个改进的对比消融实验

下述是原始 Selective Kernel Networks，改进后的 Selective Kernel Networks 和 SKNet 与 GAN 相结合的内容四种网络的准确率曲线：

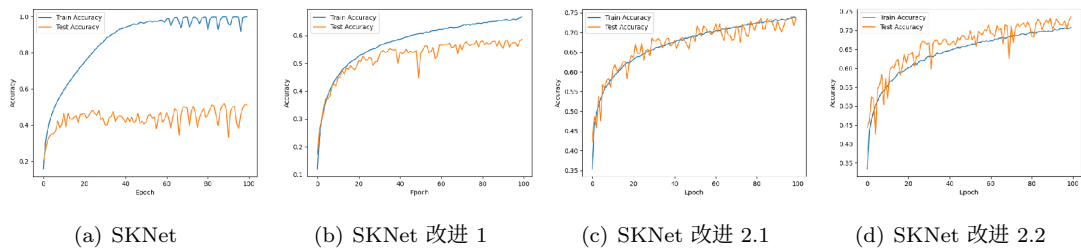


图 11: 对比试验

通过观察, 我们发现在 SKConv 模块生成特征后立即应用视觉注意力机制 (Visual Attention Block) 的模型在准确率上要稍微超过先使用一系列 SKConv 模块生成特征, 然后再整体应用注意力机制的模型, 且这两种方法的表现都显著优于单独使用 SKNet 模型。这种现象表明, SKConv 模块和视觉注意力机制的更深度结合, 可能会为模型性能的提升带来更大的潜力。

换言之, 立即对 SKConv 模块的输出应用视觉注意力机制, 能更加精细地调整每一层特征的重要性, 这种细粒度的特征调整对模型的性能提升有显著的影响。然而, 将视觉注意力机制整体应用于一系列 SKConv 模块生成的特征也具有一定的效果, 尽管在准确率上略逊一筹。相比单独使用 SKNet 模型, 这两种结合策略都展现出了更高的性能。因此, 我们有理由相信, 通过深度地融合 SKConv 模块和视觉注意力机制, 可能会实现更大幅度的模型性能提升。

四、 成员贡献

在概述中给出的项目链接中包含五个文件夹分别代表我们五名组员的工作文件夹。

各组员贡献如下:

- 1911433-林坤: 复现 Res2net, 提出隐式语义数据扩增算法来提高分类精度, 部分文档撰写
- 2011915-崔江浩: 复现 SKNet, 在此基础上提出 SKNet 与 VAN 模型结合, 进行两种改进方法并进行相应实验, 部分文档撰写
- 2013154-段辰睿: 复现 GCNet, 提出将 GCNet 与 SENet 结合的 idea、进行将 GCNet 与 SENet 结合的实验, 部分文档撰写
- 2011742-徐宇昂: 提出进一步改进 GC-SENet 的两种方法、GC-SENet 进一步改进的实验, 部分文档撰写
- 2013289-朱莞尔: 复现 GCNet 和 SENet 结合的实验, 进行 Res2net 改进实验, 部分文档的撰写, PPT 制作