南开大学

Nankai University

南　　开　　大　　学

网 络 空 间 安 全 学 院

深度学习实验报告

Cifar100 分类

姓名：曹维伦

年级：2020 级

2023 年 6 月 25 日

摘要

深度学习大作业报告，Github 链接：https://github.com/Lunn88/Cifar-100

关键字：Deep Learning

目录

一、 基础方法

(一) 实验细节

数据增强使用 RandomHorizontalFlip、RandomCrop(reflect)、RandAugment(n 取 2，m 取 10)，Batch size 为 64 or 256。

优化器使用 Adam, lr scheduler 使用 One Cycle lR(0 to 0.01)，没有使用 warm up(效果差)。

(二) Selective Kernel Networks

在一般的卷积神经网络当中，每层人工神经元的感受野通常被设计为相同大小。但是生物却会根据不同场景的刺激自动调整感受野，因此选择性内核 (selective Kernel) 被提出，通过分割、融合、选择三个操作实验 SK Conv。

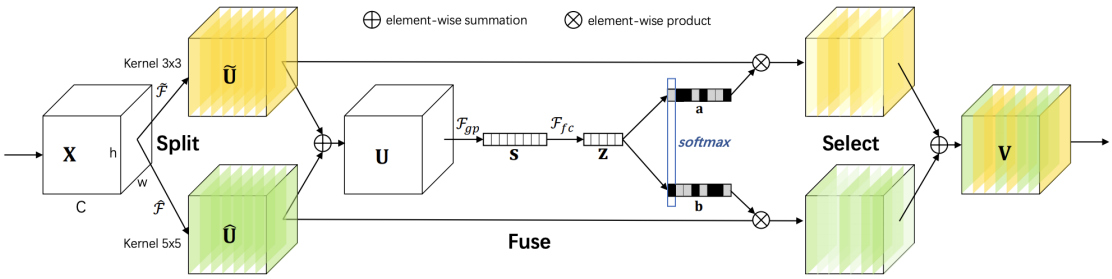

Figure 1. Selective Kernel Convolution.

图 1: SK Conv 结构图 [5]

首先是分割 (Split)，使用 3x3, 5x5 等不同大小的卷积核进行卷积，为神经网络提供不一样的感受野，进而增加准确率。其中 5x5 的卷积使用的是 3x3 的扩张卷积，为的是提高效率并减少参数量

其次是融合 (Fuse)，将前向得到的几个不同分支卷机出的特征图相加进行信息融合，接着使用全局平均池化来嵌入全局信息，以生成通道相关的注意力信息，再通过一个全连接层压缩信息的同时降低维数，同样是为了提高效率，最后过一个 Softmax 得到注意力权重。

最后是选择 (Select)，将前向得到的注意力权重与分割中得到的特征图相乘，最后将这几个分支相加得到输出。

实验数据如下

- SKNet26
- 4.81M Parameters
- Loss on train set: 0.36
- Loss on test set: 1.28
- Top1 acc:69.65
- Top5 acc:91.06

SKNet 相比于其他 ResNet 增加的参数量和计算成本都不高，却带来相当不错的性能提升，是个非常优秀的结构，此外，SK Conv 也非常适合做为插件嵌入其他网络。

(三) Coordinate Attention for Efficient Mobile Network Design

在无论是 SENet，SKNet 中都已经证明通道注意力对网络的性能占有举足轻重地地位，但却通常忽略了位置信息，这对基于生成空间选择的注意力十分重要。

在 Coordinate Attention 这篇论文中提出了一种称之为" 坐标注意力" 的机制，通过分解两个一维特征编码过程，沿着两个平面方向 X, Y 聚合特征，这样一来不仅可以沿着一个空间方向捕获远程依赖关系，同时可以严令一个空间方向保留精确的位置信息，然后将得到的特征图单独编码成一对方向感知和位置敏感的注意力图，加强了网络对平面上的感知。

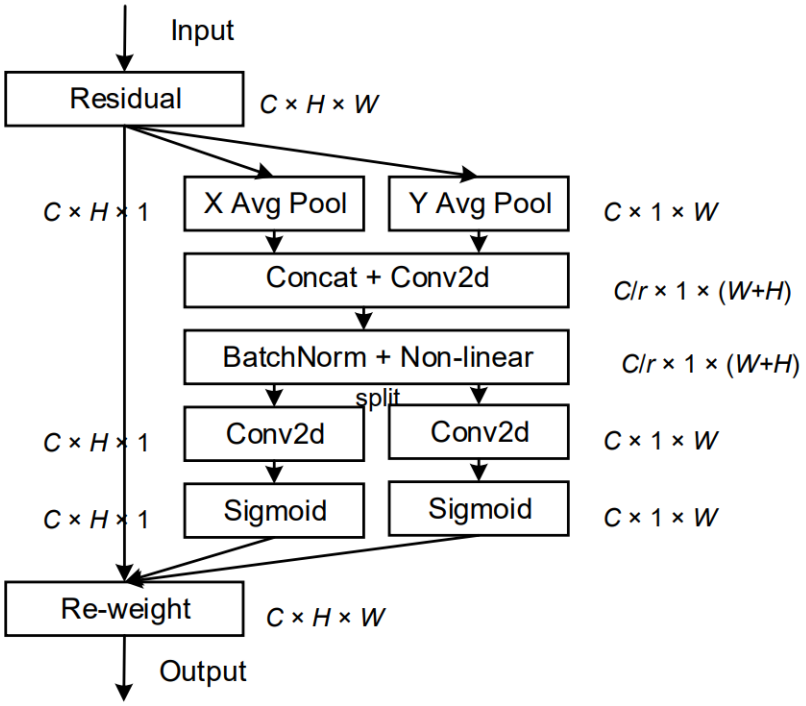

图 2: CA Conv 结构图 [3]

实现方式为使用 (H, 1), (1, W) 的 Kernel 对输入进行 Pooling，得到基于平面方向的特征图，将他们相连接后过一层卷积再将它们分开，分别再过一个卷积后使用 Sigmoid 激活函数乘回输入得到涵盖空间位置信息的特征图。

实验数据如下

- Coordinate Attention
- 9.7M Parameters
- Loss on train set: 0.91
- Loss on test set: 1.22
- Top1 acc:66.66
- Top5 acc:90.25

CA Conv 同样非常灵活且轻量，可以简单的插入经典网络中。

(四) Rotate to attend: Convolutional triplet attention module

近年的研究中都专注于建立通道或空间位置之间的相互依赖关系，也就是注意力机制，本文中作者提出了轻量但十分有效的注意力机制，通过使用三个分支的结构捕捉交叉维度交互来计算注意权重的新方法，藉由旋转操作和残差变换建立不同维度间的依赖关系，以几乎没有增长的计算开销对通道和空间信息进行编码，简单高效。

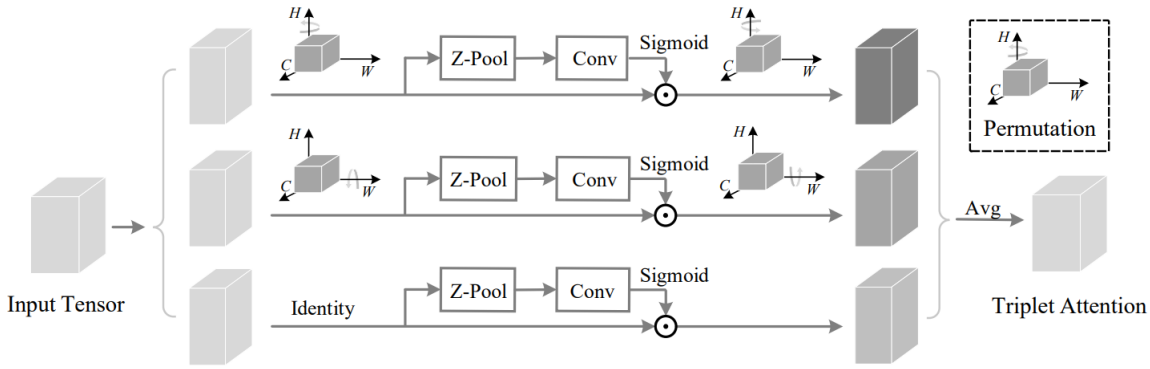

图 3: Triple attention 结构图 [1]

Triplet Attention 由三个独立的分支组成，藉由旋转操作构建 (C, W), (C, H), (H, W) 三个维度之间的跨维交互，最后将三个输出使用平均聚合。

其中的 Z-pool 将通道维度缩减到 2 维，将该维度上的平均汇集特征 (AvgPool) 与最大汇集特征 (MaxPool) 连接起来，使得该层能够保留原有信息的同时缩小其维度，使得计算量更轻。

实验数据如下

- Triplet Attention
- 11.23M Parameters
- Loss on train set: 1.30
- Loss on test set: 1.54
- Top1 acc:59.6
- Top5 acc:85.5

(五) Visual Attention Networks

自注意力机制 (Self-Attention) 本是为了 NLP 而设计，但如今已被广泛地使用在 CV 领域中，然而与一维的文本不一样，要在图像上使用该机制有三个问题，首先是图像为二维结构，使用一维序列的处里方式显然不合理，再来是二次的复杂度对高分辨率图像代价太昂贵，最后是该方法只捕捉到了空间适应性而忽略了通道适应性。

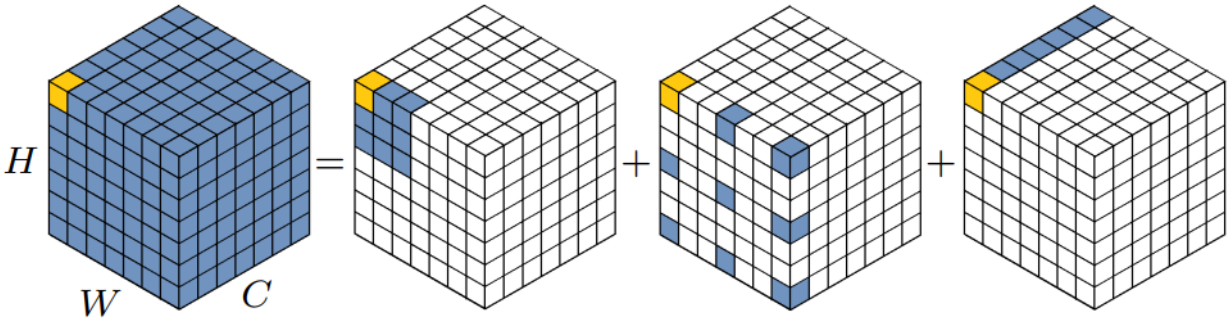

图 4: LKA 结构图 [2]

本文提出的大核注意力机制 (LKA) 可以很好地解决上述问题，LKA 将卷积分解为三部分：深度卷积 (捕捉空间局部信息)、深度扩张卷积 (捕捉空间全局信息)、和点卷积 (捕捉通道注意力信息)，将一个 KxK 卷积分解为一个 k/d x k/d 的深度卷积、一个 (2d−1)x(2d−1) 深度膨胀卷积（扩张率为 d）和一个 1x1 卷积。通过上述分解，可以捕捉到计算成本和参数很小的远程关系。在获得远程关系后，可以生成注意力图。

实验数据如下

- VAN
- 11.06M Parameters
- Loss on train set: 1.09
- Loss on test set: 1.39
- Top1 acc:63.02
- Top5 acc:88.46

LKA 结合了 CNN 和自注意力的优点，包括局部结构信息、长距离依赖性和适应性，并且避免了自注意力的缺点，对通道适应性也考虑进去。

(六) Res2net: A new multi-scale backbone architecture

以往认为视觉认物要想获得更好的效果就是堆叠卷积层数，希望网络能学习到多尺度的特征，但在 Res2Net 中程老师提出的并非 layer 的堆叠组合，而是在一个 block 中通过更细粒度的卷积产生多个感受野的组合

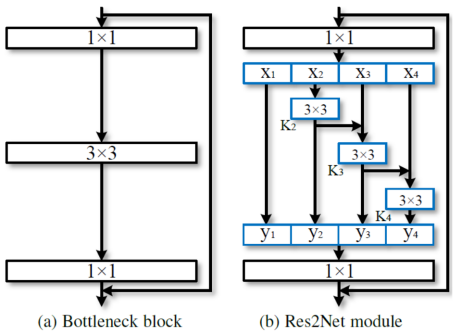

图 5: Res2Net Bottleneck [4]

将输入分为四祖，第一组直接往下传，第二组过一层 3x3 卷积提取特征后往下传，下一组的输入再加上前一组的输入过一层 3x3 卷积重新提取更细粒度的特征，最后将四组都 Concat 起来。

虽然在同深度下 Res2Net50/101 的表现都比 ResNet50/101 好，但文章中对该结构为何能 work 并没有做太多解释，但从可视化的结果来看，我个人的想法是更丰富的感受野带来更强的辨识一整个整体事物的能力。

实验数据如下

- Res2Net
- 14.03M Parameters
- Loss on train set: 1.1
- Loss on test set: 1.55
- Top1 acc:60
- Top5 acc:85.36

二、 改进、创新

(一) Mix with Selective Kernel, Coordinate attention, Triple attention

基础方法中 Selective Kernel 的准确率最高，因此以该方法为基础做改进，首先观察到的是在 SKNet 中仅关注通道间的注意力，对空间位置 (H*W) 的特征利用并不充分。

将 Coordinate convolution 作为插件嵌入或能加强位置信息的捕捉，此外，若将捕捉完空间位置信息的特征图再使用 Triple attention 让不管是通道间的注意力信息，或是空间位置的注意力信息更好的交融混合，还能些许提升准确率。

这部份改进的消融实验探讨的是嵌入方法的先后顺序，ResNet 的 Basic Block 中有两层卷积，再加上基础方法中的卷积共四种，在不同的先后顺序下准确率也各有不同。

以下为实验结果

Method	Parameters (M)	Top1 acc	Top5 acc
Coordinate attention	9.7	66.66%	90.25%
Coordinate attention+Triple attention	9.7	67%	90.5%
SKNet	4.81	69.65%	91.06%
SKNet(Conv1->SK Conv->CA Conv->Conv2)	6.65	70.02%	91.86%
SKNet(CA Conv->Conv1->SK Conv->Conv2)	6.65	70.12%	91.72

表 1: 实验数据

(二) Change backbone-DenseNet

除了引入 CA Convolution 外，我还想到另一种加强空间上特征的利用方法，那就是将 backbone 改成 DenseNet，借此弥补 SKNet 欠缺对空间位置信息利用的不足。

实现方法为在 DenseNet 的基本块 Bottleneck 中加入 SK Conv，但这就会引申出一些问题：加在哪？加多少？是否保留分组卷积？是先 Concatenate 再做 SK Conv 还是先做 SK Conv 再做 Convatenate？等等，加的太过紧密或疏散都不好、加的太少效果可能不够好、加的太多又会大幅增加参数量进而影响到训练时间、又分太多组会不会使信息交融不够使得拟合能力差，这些都是要考虑并取得一个较好平衡的点。

Bottleneck forward

```
def forward(self, x):
    out = self.relu(self.bn1(self.conv1(x)))
    if self.do_sk == True:
        out = self.sk_conv(out)
    out = self.relu(self.bn2(self.conv2(out)))
    out = torch.cat([x, out], 1)
    return out
```

藉由多次实验结果的对比后发现平均地 (一层加一层不加 or 两层加两层不加) 加入 SK Conv，并且当 i % (blocks_config // 5) == 0 时做 SK Conv 效果较好，而原本的 SKNet 中为了减少参数量和防止过拟合，使用 32 组的分组卷积，但是在 DenseNet 中效果不是很好，经过调整后并实验发现 4 组的结果最好。

make dense block

```
def make_dense_block(self, block, in_channel, blocks_config):
    dense_block = nn.Sequential()
    for i in range(blocks_config):
        if i % (blocks_config // 5) == 0:
            dense_block.add_module('bottle_neck_layer_{}'.format(i), block(
                in_channel, self.growth_rate, do_sk=True),)
        else:
            dense_block.add_module('bottle_neck_layer_{}'.format(i), block(
                in_channel, self.growth_rate, do_sk=False),)
        in_channel += self.growth_rate
    return dense_block
```

以下为实验结果

Method	Parameters (M)	Top1 acc	Top5 acc
DenseNet121	2.8	74.36%	94.3%
DenseNet+SK Conv(//2, Conv2->Cat->SK Conv)	5.9	74.68%	94.15%
DenseNet+SK Conv(//2, Conv1->SK Conv->Conv2)	4.13	75.41%	94.25%
DenseNet+SK Conv(//4, Conv1->SK Conv->Conv2)	4.78	76.94%	94.72%
DenseNet+SK Conv(//5, Conv1->SK Conv->Conv2)	5.18	77.27%	94.76%
DenseNet+SK Conv(//6, Conv1->SK Conv->Conv2)	5.31	76.74%	94.68%
DenseNet+CA Conv(Conv1->CA Conv->Conv2)	2.92	75.54%	94.11%
DenseNet+Triple attention(Conv1->Triple attention->Conv2)	2.81	74.87%	94.04%

表 2: 实验数据

最后效果最好的模型为 DenseNet+SK Conv(//5, Conv1->SK Conv->Conv2)，训练集上的 Loss 为 0.207，测试集上的 Loss 为 0.96，Loss, Arr 曲线如下：

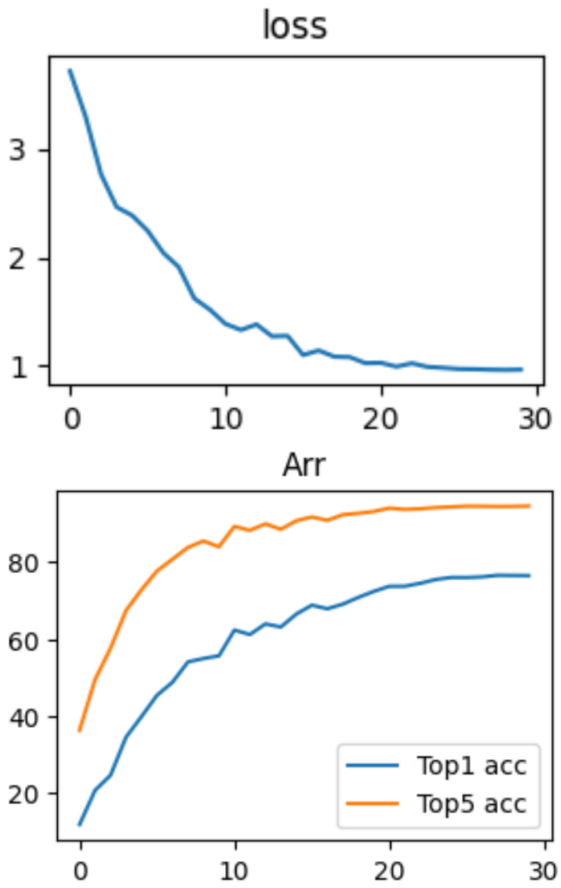

图 6: Loss&Arr

三、 总结与未来工作设想

构想创新 Idea 时有想过只使用 3x3(5x5) 的卷积来提取特征是否过于简单？我有尝试将 CA Conv 加在提取特征这步，或是使用类似 DenseNet 的原理多次提取在 Concatenate 在一起，但是不仅参数量大增，最后效果也不怎么样，便没有再多做尝试。

另外提取出的特征相加后得到的 U，只是简单的做一个 Global Average Pooling 来得到注意力信息也有点简单，可以尝试在这步多多利用空间位置上的信息，融合进通道注意力中。

最后也可以更换其他比 ResNet 更好的 backbone，如 DenseNet, Res2Net 等等。

四、 小组分工

1. 曹维伦 (组长)：实现基础方法与实验、构想创新方法与实验、制作 PPT 并展示、撰写本篇报告
2. 徐熙航：帮助制作 Res2Net, VAN 的 PPT 内容 (2-3 页)
3. 秦一航：无
4. 魏大景：无

我 (曹维伦) 有请组员们在我的 github 仓库上建分支 push，但是没人建。做作业过程中无人提出任何想法。

我完成所有实验和展示后请组员们写报告中介绍基础方法的部份就好，我负责写改进创新的部份，他们说好没问题，但是直到 6/25 下午 3:30 无人动笔，我只好自己完成。

整个作业基本上由我独自完成，烦请老师斟酌给过。

参考文献

[1] Ajay Uppili Arasanipalai Qibin Hou Diganta Misra, Trikay Nalamada. Rotate to attend: Convolutional triplet attention module. 2021.

[2] Zheng-Ning Liu-Ming-Ming Cheng-Shi-Min Hu Meng-Hao Guo, Cheng-Ze Lu. Visual attention network. 2022.

[3] Jiashi Feng Qibin Hou, Daquan Zhou. Coordinate attention for efficient mobile network design. 2021.

[4] Kai Zhao Xin-Yu Zhang Ming-Hsuan Yang Philip Torr Shang-Hua Gao, Ming-Ming Cheng. Res2net: A new multi-scale backbone architecture. 2019.

[5] Xiaolin Hu Jian Yang Do Xiang Li, Wenhai Wang. Selective kernel networks. 2019.