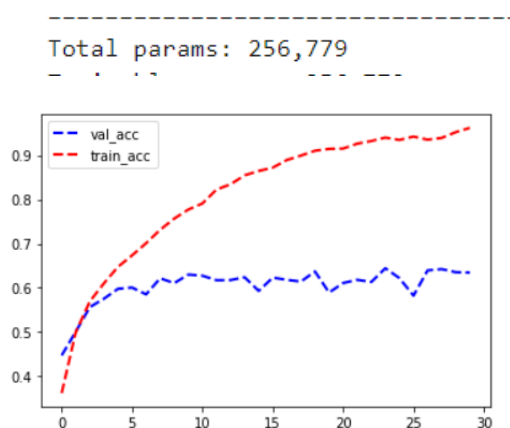


学号：3119001616 班级：数据科学与大数据技术 2 班 姓名：邹润豪

1、请对 Knowledge Distillation/Design Architecture 两种方法中实作并详述你的方法，将同一个大 model 压缩至接近相同的参数量（将模型参数总量截图放在该问题回答下），记录其 accuracy 并分析。（占分 40%）

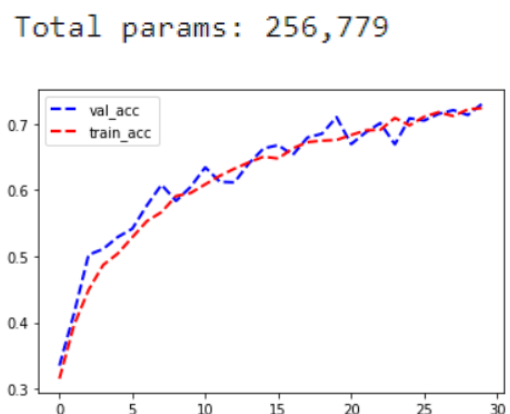
答：

Design Architecture :



该模型收敛速度快，验证集准确率在 0.61 左右

Knowledge Distillation :

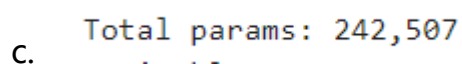
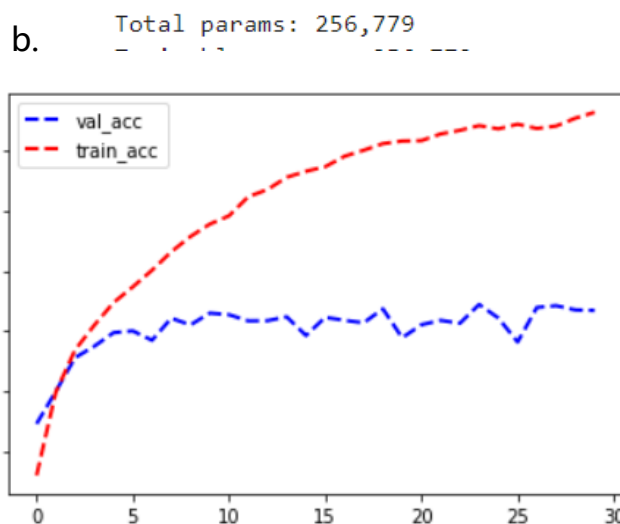
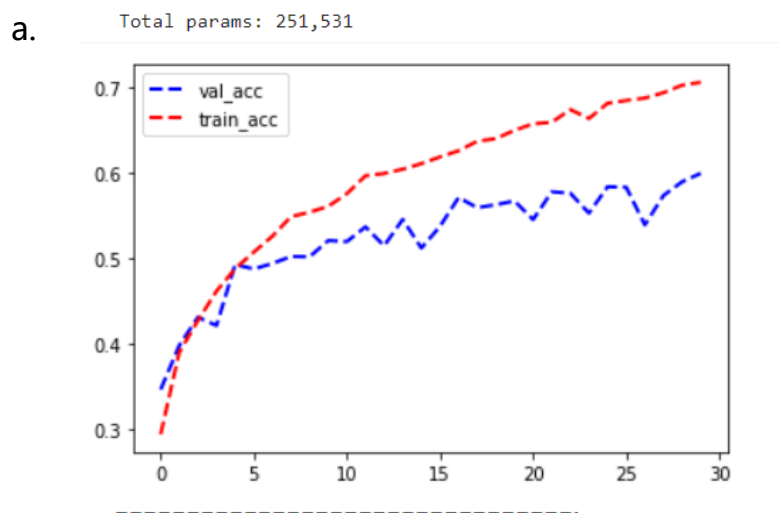


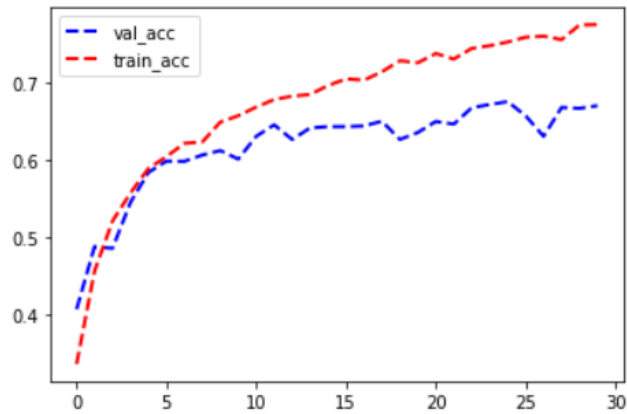
验证集和训练集准确率都比较相似，收敛速度较慢，受 teacher 的影响较大

2、请尝试比较以下 validation accuracy,以下三种方式均以画图的方式在报告中体现: (占分 40%)

- 原始 CNN model (用一般的 Convolution Layer) 的 accuracy(10%)
- 将 CNN model 的 Convolution Layer 换成总参数量接近的 Depthwise & Pointwise 后的 accuracy(15%)
- 将 CNN model 的 Convolution Layer 换成总参数量接近的 Group Convolution Layer (Group 数量自订, 但不要设为 1 或 in_filters)(15%)

答:





3、实验过程的记录: (占分 20%)

(1) 任何实验过程 , 遇到的问题记录及解决 , 思考等(5%);

A. 提升模型:

将激活函数改为 Hardswish

优化器改为 Adam

B . 传统 CNN 在参数下降后是否能媲美相似参数 Depthwise & Pointwise 的 CNN :

Depthwise & Pointwise 的 CNN 表现更好 , 说明 Depthwise & Pointwise 能够在参数较低的情况下提取比传统 CNN 更好的 feature map

C . 在移植效果不理想后提升模型:将最后两层的输出通道增大为 18×16 , 将验证集和训练集一并放入训练

(2) 请说明你实际使用的模型架构(5%);

```

class StudentNet(nn.Module):
    def __init__(self, base=16, width_mult=1):
        super(StudentNet, self).__init__()
        multiplier = [1, 2, 4, 8, 16, 16, 18, 18]
        bandwidth = [base * m for m in multiplier]
        for i in range(3, 7):
            bandwidth[i] = int(bandwidth[i] * width_mult)
        self.cnn = nn.Sequential(
            nn.Sequential(
                nn.Conv2d(3, bandwidth[0], 3, 1, 1),
                nn.BatchNorm2d(bandwidth[0]),
                nn.Hardswish(),
                nn.MaxPool2d(2, 2, 0),
            ),
            nn.Sequential(
                nn.Conv2d(bandwidth[0], bandwidth[0], 3, 1, 1, groups=bandwidth[0]),
                nn.BatchNorm2d(bandwidth[0]),
                nn.Hardswish(),
                nn.Conv2d(bandwidth[0], bandwidth[1], 1),
                nn.MaxPool2d(2, 2, 0),
            ),

```

```

            nn.Sequential(
                nn.Conv2d(bandwidth[1], bandwidth[1], 3, 1, 1, groups=bandwidth[1]),
                nn.BatchNorm2d(bandwidth[1]),
                nn.Hardswish(),
                nn.Conv2d(bandwidth[1], bandwidth[2], 1),
                nn.MaxPool2d(2, 2, 0),
            ),

            nn.Sequential(
                nn.Conv2d(bandwidth[2], bandwidth[2], 3, 1, 1, groups=bandwidth[2]),
                nn.BatchNorm2d(bandwidth[2]),
                nn.Hardswish(),
                nn.Conv2d(bandwidth[2], bandwidth[3], 1),
                nn.MaxPool2d(2, 2, 0),
            ),

            nn.Sequential(
                nn.Conv2d(bandwidth[3], bandwidth[3], 3, 1, 1, groups=bandwidth[3]),
                nn.BatchNorm2d(bandwidth[3]),
                nn.Hardswish(),
                nn.Conv2d(bandwidth[3], bandwidth[4], 1),
            ),

            nn.Sequential(
                nn.Conv2d(bandwidth[4], bandwidth[4], 3, 1, 1, groups=bandwidth[4]),
                nn.BatchNorm2d(bandwidth[4]),
                nn.Hardswish(),
                nn.Conv2d(bandwidth[4], bandwidth[5], 1),
            ),

```

```

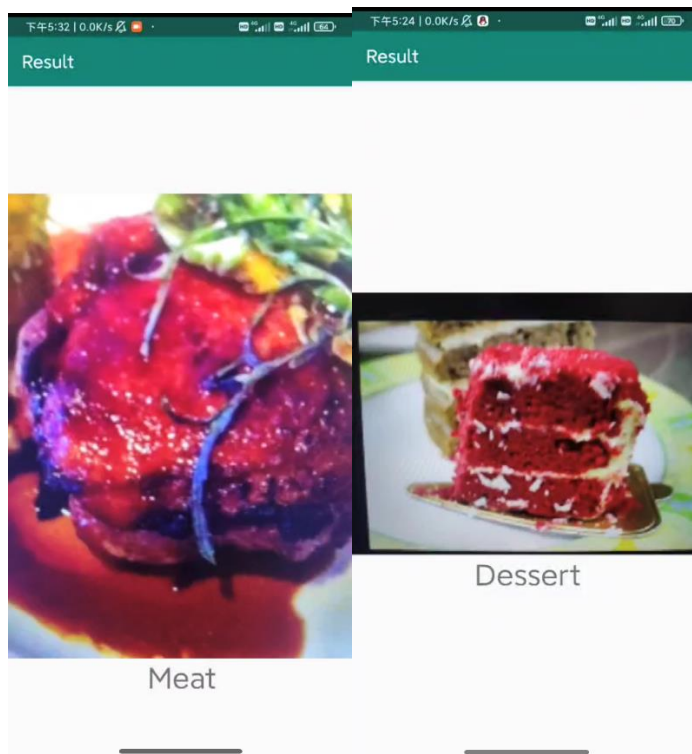
.....nn.Sequential(
.....nn.Conv2d(bandwidth[6], bandwidth[6], 3, 1, 1, groups=bandwidth[6]),
.....nn.BatchNorm2d(bandwidth[6]),
.....nn.Hardswish(),
.....nn.Conv2d(bandwidth[6], bandwidth[7], 1),
.....),
.....nn.AdaptiveAvgPool2d((1, 1)),
.....)
self.fc = nn.Sequential(
.....nn.Linear(bandwidth[7], 11),
.....)

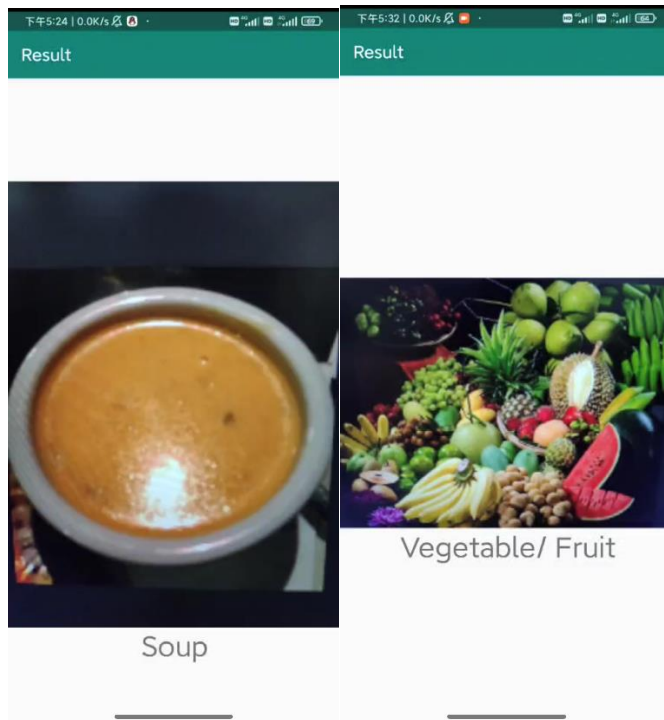
def forward(self, x):
out = self.cnn(x)
out = out.view(out.size()[0], -1)
return self.fc(out)

```

(3) 记录手机端模型移植的过程(5%);

- a. 将训练好的模型转换为 TorchScript 格式
- b. 下载配置 Android Studio 并打开参考项目
- c. 将 TorchScript 格式的模型放入指定目录
- d. 创建相关文件及安卓模拟器
- e. 运行程序



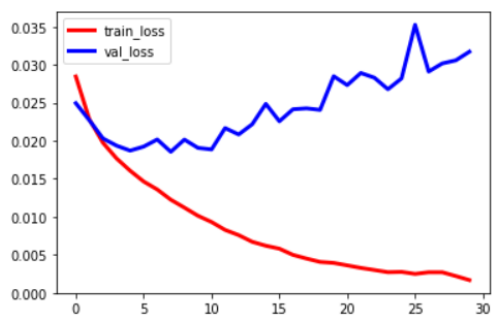


f. 能够正常运行后将安装包安装到手机上

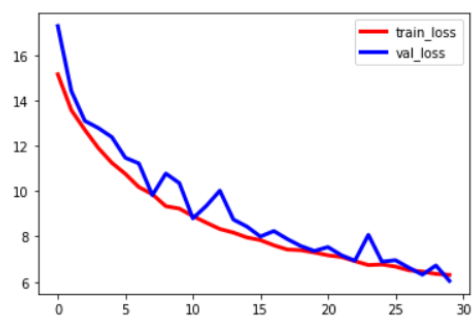
(4) train/valid loss 曲线等训练过程(5%) ;

1、

Design Architecture :

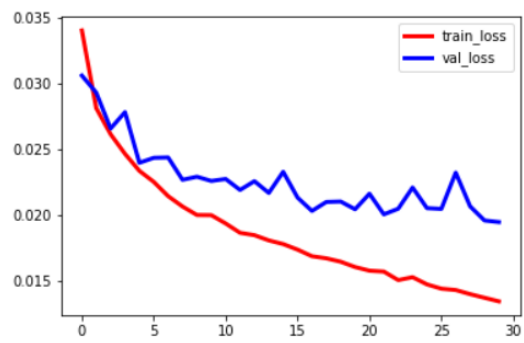


Knowledge Distillation :



2、

a.



b.



c.

