

# 基于 ResNet 的图像处理网络实现

申广辉 机械 13 2021010407

## 一、ResNet 的介绍

ResNet (Residual Network) 是一种深度残差网络，由微软亚洲研究院的何凯明等人于 2015 年提出。ResNet 的出现解决了深度神经网络中出现的梯度消失和梯度爆炸等问题，使得可以训练更深层的神经网络模型。传统的深度神经网络存在梯度消失和梯度爆炸问题，这是由于在反向传播过程中，随着网络层数的增加，梯度逐渐变得非常小，导致训练过程中参数更新缓慢，甚至无法收敛。ResNet 通过引入了残差连接 (residual connection) 来解决这一问题。

在 ResNet 中，每个基本的网络单元 (或称为残差块) 不再直接学习原始输入与目标映射之间的映射关系，而是学习输入的残差 (即输入与目标之间的差异)。这样一来，即使网络深度增加，模型仍然可以通过学习恒等映射 (identity mapping) 来实现，从而避免了梯度消失的问题。

ResNet 的核心思想就是通过残差学习，将输入和输出之间的关系建模为残差函数。如果一个恒等映射可以使网络性能不降低，那么这个残差块就可以学习到一个更优的映射，进一步提升网络性能。此外，残差连接还有利于梯度的反向传播，使得可以训练非常深层的网络模型。

ResNet 的网络结构通常由多个残差块组成，每个残差块包含若干个卷积层和残差连接。在网络的尾部通常会添加全局平均池化层和全连接层，用于实现分类等任务。

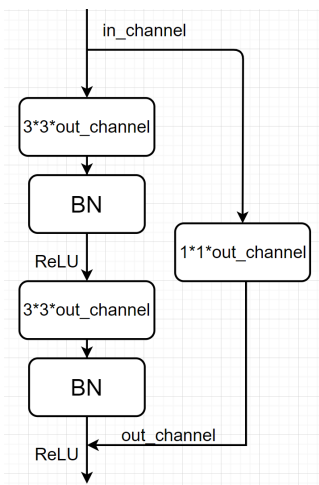
## 二、任务介绍

采用 cifar-10 数据集对 60000 张 10 类图片进行分类，图像尺寸为 32\*32 的 RGB 图像。类别包括: 'plane', 'car', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', 'truck'。

## 三、ResNet 网络搭建

ResBlock 模块：

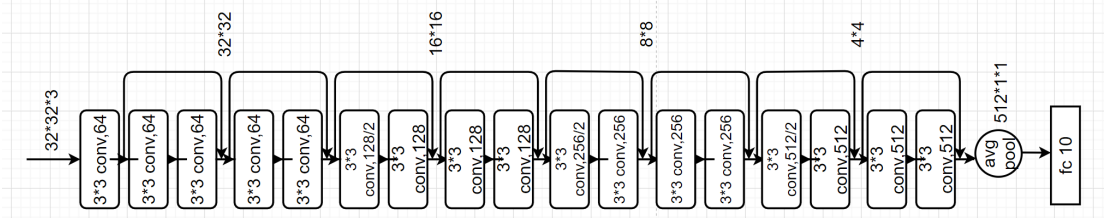
残差模块采用两个 3\*3 卷积层实现，每层卷积完成后进行 BatchNorm 操作进行归一化，归一化的目的是为了使得特征平滑，改善地形图，便于优化器求解，随后通过 ReLU 函数进行激活，提高网络的非线性表达能力。直连结构通过一个 1\*1 卷积层实现，保证在输入通道和输出通道数不同时进行调整，保证最后结果可加。



图表 1 ResBlock

ResNet 搭建:

输入图像大小为 32\*32, RGB 三通道, 通过一个卷积层增加其通道到 64 个, 然后通过上述残差模块进行网络搭建, 每经过两个残差模块图像的通道数翻倍, 图像的尺寸减半, 目的是将图像的特征分摊到不同的通道, 慢慢的减小图像尺寸, 一方面可以减小参数量, 另一方面可以避免最后在做平均值池化时的信息损失。经过 8 个残差模块后图像的通道数为 512 个 (特征有 512 个), 图像尺寸仅为 4\*4, 平均值池化后得到 512 维向量, 经过一个线性层后得到 10 类图片的概率。



图表 2 ResNet

其它细节:

1.损失函数的选择: 对于分类问题一般直接选择交叉熵损失函数即可, 交叉熵损失函数的含义是最大似然估计函数的负对数函数。最大似然估计是指已知统计结果, 当前概率模型发生与统计结果一致的结果的概率。因此似然函数越大, 说明网络概率模型越接近实际模型, 也就是交叉熵损失函数越小, 模型越准确。

2.优化器的选择: 最终选择带冲量的 SGD 优化器 (也尝试过 Adam, 但 Adam 衰减很快, 很快达到收敛, 但最终的收敛值不如精调之后的 SGD), SGD 优化器是随机梯度优化器, 它是指每次对一个 Batch 的样本结果求梯度, 更新权重的值。更新公式满足

$$\theta_k = \theta_{k-1} - \eta \nabla J_{k-1}$$

其中 $\eta$ 为超参数学习率。带冲量的 SGD 是指其在更新时, 考虑了历史的更新值。如果上一时刻还在下降, 那更新值将会有继续下降的趋势, 这样将有利于优化器逃离局部极值, 因此更新公式变为如下形式:

$$\theta_k = \theta_{k-1} - \eta \Delta_k$$

$$\Delta_k = \beta \Delta_{k-1} + (1 - \beta) \nabla J_{k-1}$$

其中 $\beta$ 为超参数,代表上一步的积累对当前更新值的影响, 一般取 0.9 或者 0.99, 这说明当前的更新值将会很大程度上以来之前的结果。

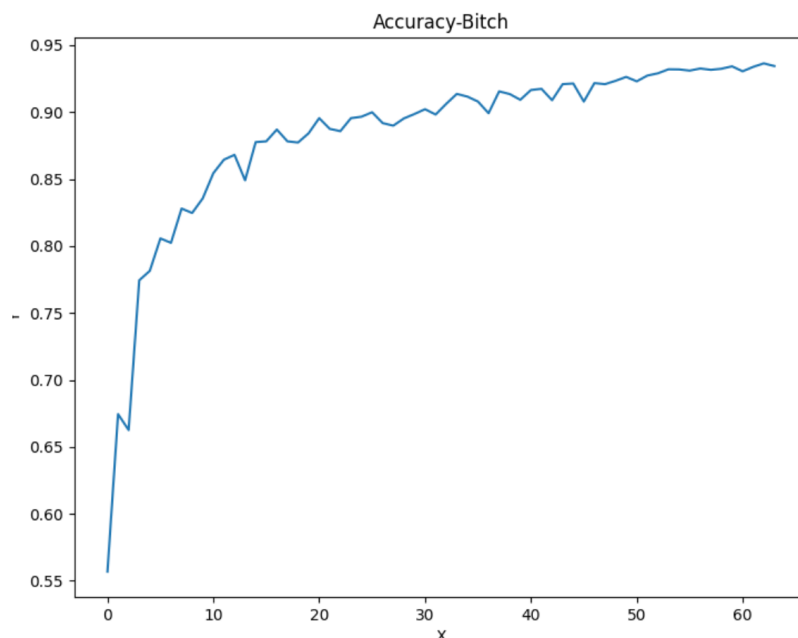
3.自适应学习率: 学习率是深度学习中一个必须要调整的超参数, 一般会将其设置为 0.001 或更小, 以便于网络收敛, 但明显 0.001 是个极小的值, 并不利于优化器找到最优值, 而会停在局部极值, 因此, 自适应学习率应运而生。正如其名称所示, 自适应学习率是自适应的, 会随着训练的进行而调整, 调整测率也是百花齐放。在我的训练过程中, 最终发现余弦退火函数的效果是最好的, 余弦退火函数是指其学习率按照余弦函数变化, 这样相当于在系统训练到一定阶段后进行“重启”, 重新检索极值, 可以使得优化器逃离局部极值, 参数调整得到可以得到比较好的结果。

4.数据增广: 数据增广不仅可以增加数据量, 更重要的是其能增强模型的泛化能力。在图像识别的任务中, 神经网络通过识别图像的关键特征进行图像的分类, 但是对于神经网络来说, 识别主体在图像中的位置和姿态也是重要特征。同样是一只猫, 其出现在左上角还是右下角, 它的头朝左还是头朝右都是猫的特征, 但这显然不是我们想要的, 我们想要的是它能认识猫, 而与其出现位置无关, 因此我们需要通过数据增广的办法, 将图像进行随机旋转、

剪切和对称, 让其统统进行学习, 在最后的测试中, 会得到更好的效果。

#### 四、模型调试

这块儿还是省略吧.....总之经过了百八十代(有记录的是 30 代)的炼丹过程, 综合运用上述细节和反复的参数调试, 最终得到了一个比较好的结果, 在 50000 张训练集上的识别准确率为 99%, 在 10000 张测试集上的准确率为 93%, 随着学习其准确率的曲线非常的优美。



图表 3 准确率随训练的曲线

#### 五、模型的优缺点

##### 优点:

1. 参数量小, 模型训练时间短: 在没有使用任何 dropout 技巧的情况下, 整个模型的大小为 40Mb, 训练环境是 CPU:i9 13900HX, GPU:RTX4060, 操作系统: Ubuntu20.04.06, 使用了 GPU 加速, 训练一代的时间为 40 分钟。

2. 梯度性质好, 模型收敛快: 可以从上述曲线看到, 模型的准确率基本呈现稳步上升趋势, 这得益于残差直连良好的梯度性质和 BatchNorm 归一化操作。

3. 可推广性强: 残差网络的直连模块使得梯度的传播不再受到层数限制, 残差网络使用很深的网络进行训练, 依旧会有很好的梯度性质。这使得残差网络可以轻松应对更多类别, 更大数据量的训练任务。何凯明同学设计了 ResNet152, 有 152 层, ResNet-152 在 2015 年的比赛中获得了第一名, 达到了当时最佳的性能水平。

##### 缺点:

ResNet 其实是一种非常成功的深度神经网络结构, 但它也有一些缺点:

1. 内存消耗大: 由于残差连接的存在, ResNet 模型相对于普通的深度神经网络来说具有更多的参数, 这会导致模型占用更多的内存。

2. 过拟合风险: 当处理较小的数据集时, ResNet 可能会面临过拟合的风险, 因为模型的复杂性可能会导致在训练数据上表现得很好, 但在未见过的数据上表现不佳。

3. 不适用于所有任务: 尽管 ResNet 在图像分类等任务上表现出色, 但并不是所有的任务都适合使用 ResNet。对于一些特定的任务, 其他的网络结构可能会更有效。

#### 六、结果展示

##### 1. 准确率:

```
Accuracy of the network on the test images: 93 %
10000
```

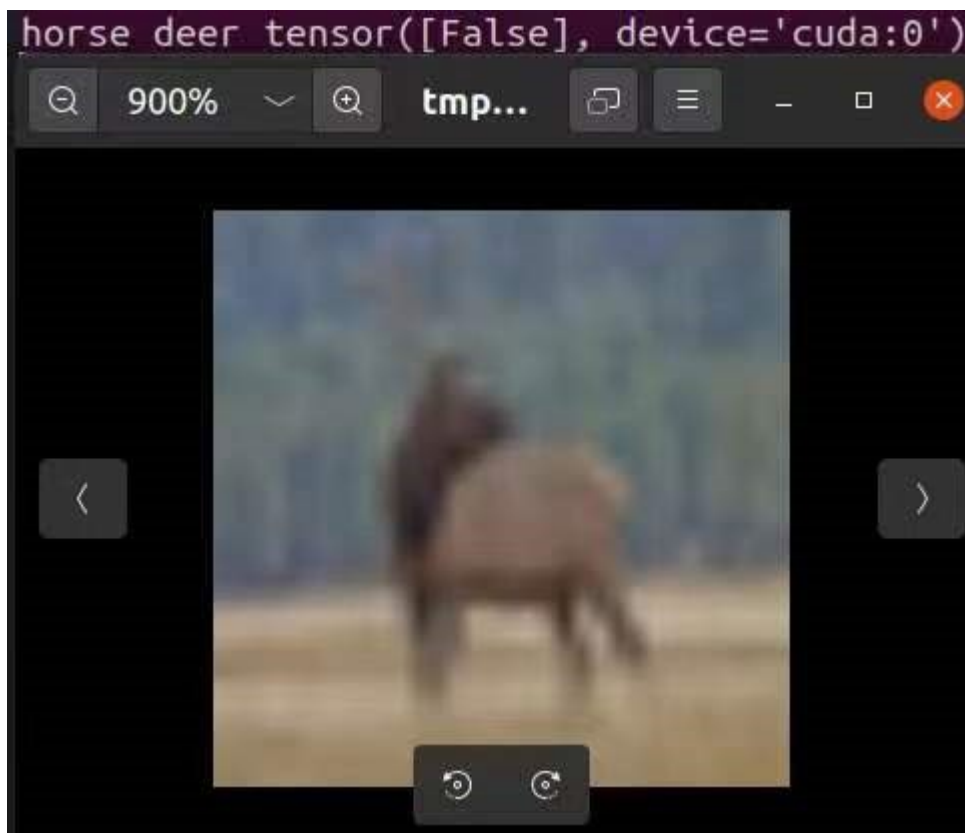
图表 4 测试集 10000 张图片的准确率

```
Accuracy of the network on the train images: 99 %
50000
```

图表 5 训练集 50000 张图片的准确率

## 2.结果可视化:

为了方便分析模型的结果, 对其测试结果进行了可视化和日志打印。日志的打印格式为[预测结果][标签结果][是否预测成功],并且对预测出错的图片进行了可视化输出。结果如下图。



图表 6 错误图片输出和日志打印

可以看到, 这本来是一头鹿 (大概是吧), 但模型预测为了马 (指鹿为马), 因此预测结果出错。

## 3.项目结构:

-ResNet	
--cifar	CIFAR-10 数据集
-- cifar-10-batches-py	
-- cifar-10-python.tar.gz	
--doc	测试结果记录
--test.md	记录说明文档
--*.jpg	
--model	已训练完成的模型

--model_weights.pth	模型参数文件
--ModelTest.py	模型测试程序
--ResBlock.py	残差网络实现
--TrainAndTest.py	网络训练和测试

注：本项目运行需要安装一些依赖包括但不限于 Pytorch、torchvision、cuda、matplotlib