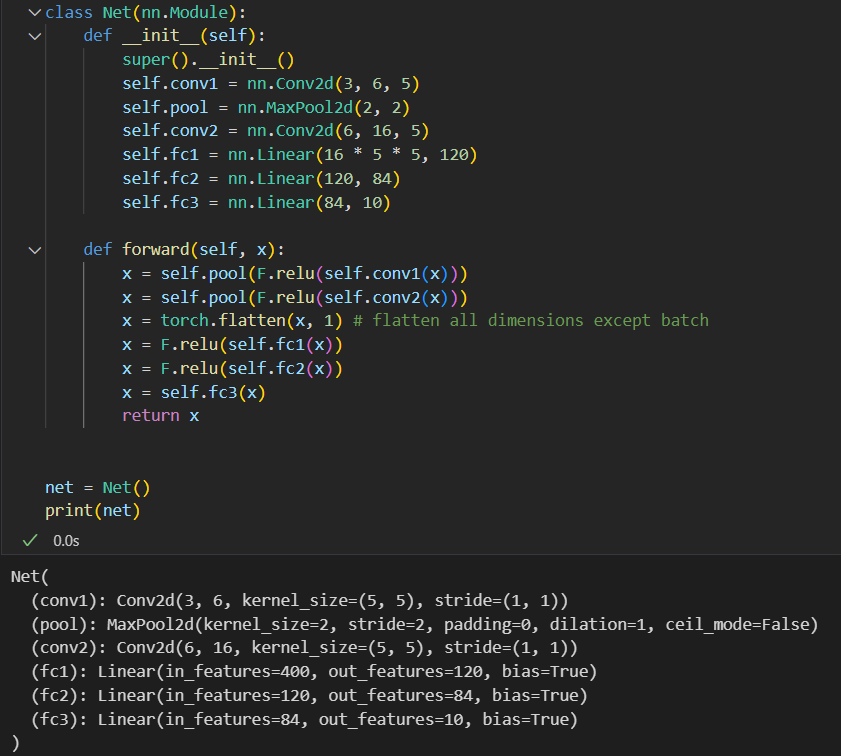
## 卷积神经网络实验报告

姓名：管昀玫 学号：2013750

### 实验要求

* 掌握卷积的基本原理
* 学会使用PyTorch搭建简单的CNN实现Cifar10数据集分类
* 学会使用PyTorch搭建简单的ResNet实现Cifar10数据集分类
* 学会使用PyTorch搭建简单的DenseNet实现Cifar10数据集分类
* 学会使用PyTorch搭建简单的SE-ResNet实现Cifar10数据集分类

### CNN



上图为老师给的CNN版本。注意，老师给的版本的forward函数有误，上图已进行了修正。该CNN结构解读如下：

1. 输入层是一个3通道的图像（RGB图像），因此输入的形状为(3, height, width)。
2. 第一个卷积层(conv1)有6个输出通道，使用5x5的卷积核进行卷积操作，卷积操作的步长为1。
3. 接下来是一个最大池化层(pool)，使用2x2的池化窗口进行池化操作，步长为2。这个池化层的作用是降低特征图的尺寸，提取主要特征并减少模型中的参数数量。
4. 第二个卷积层(conv2)有16个输出通道，同样使用5x5的卷积核进行卷积操作，步长为1。
5. 再次使用一个最大池化层(pool)，进行相同的2x2的池化操作，步长为2。
6. 紧接着是三个全连接层(fc1、fc2和fc3)，分别将特征图的展平后输入到这些全连接层中。fc1有120个输出神经元，fc2有84个输出神经元，fc3有10个输出神经元，对应于10个分类类别。

注意，老师给予的ipynb中有一句代码为：

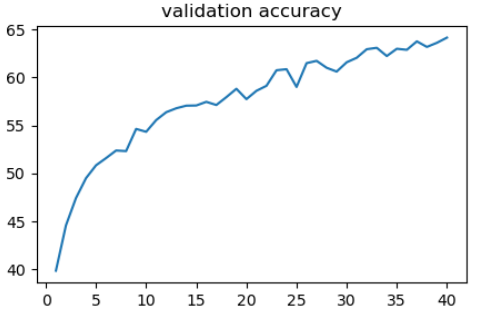
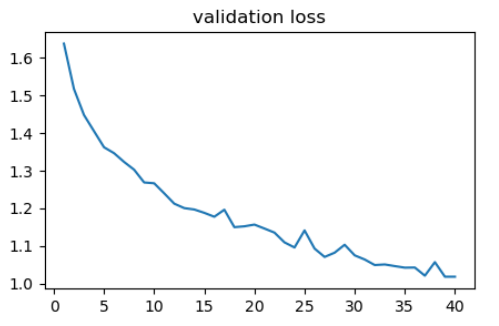
dataiter = iter(trainloader)

images, labels = dataiter.next()

由于pytorch版本不一致，在我本地应修改为：

images, labels = next(dataiter)

将该网络训练40轮，得到以下结果：



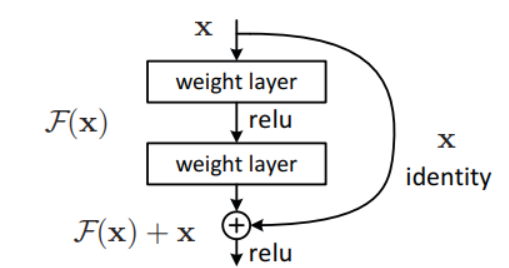
网络的准确率大约为64%左右。

### ResNet

随着网络深度的增加，精度变得饱和，然后迅速退化，这是因为网络可能收敛到局部最优，而非全局最优。resnet的提出主要是为了解决训练网络深度加深但准确率并没有提高，甚至不如浅层网络的问题。

为了解决第二个问题，论文作者提出了残差结构。在 ResNet 论文中，有两种映射方式：恒等映射（identity mapping）和残差映射（residual mapping）。恒等映射指的是通过右侧标有 "x" 的曲线来表示，而残差映射则指的是表示为 $F(x)$ 部分。最终的输出是 $F(x)$ 加上输入 $x$，即 $F(x) + x$。

ResNet 的优点在于，如果一个网络已经达到了最优状态，继续加深网络的层数可能会导致其他网络出现退化问题（错误率上升）。然而，对于 ResNet 网络来说，当网络已经接近最优状态时，残差映射中的 F(x) 部分会逐渐趋近于零，只剩下恒等映射部分。这意味着理论上网络将一直处于最优状态，网络的性能不会随着深度的增加而降低。这种残差映射的设计使得 ResNet 能够更好地训练非常深的神经网络，并获得更好的性能。



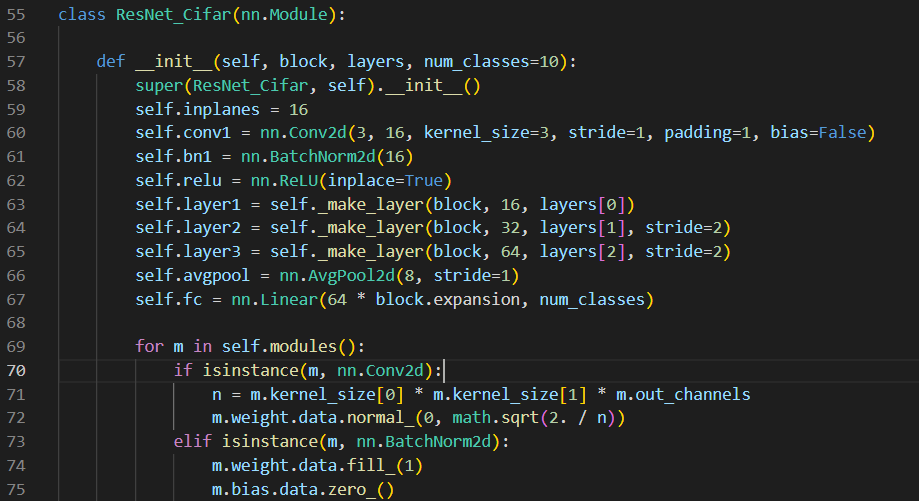
该残差块图源He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.

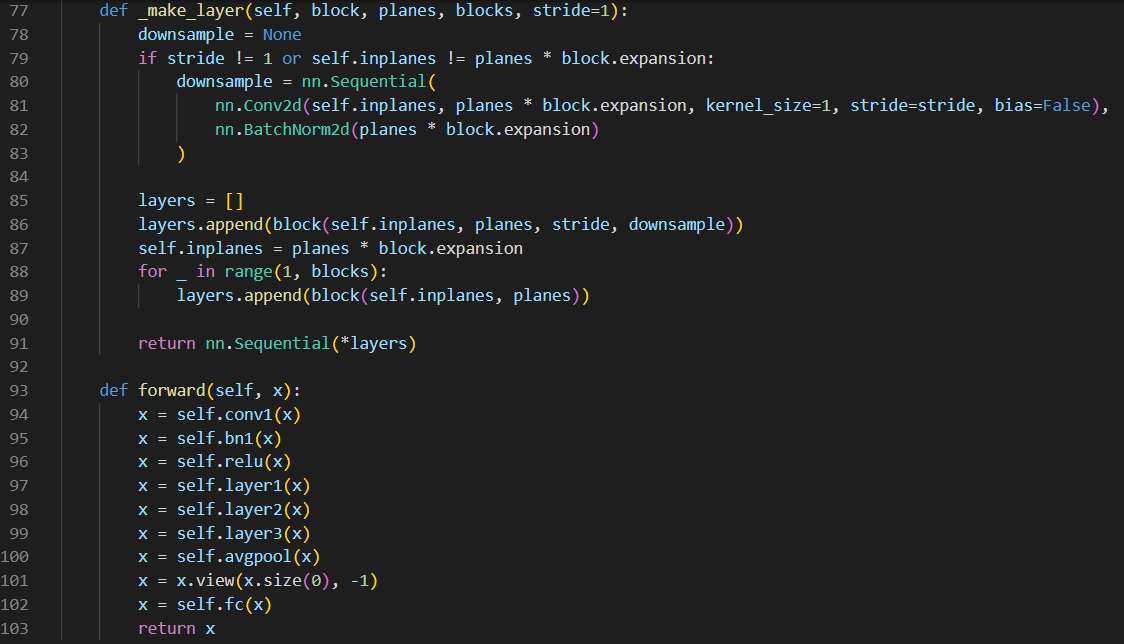
在论文中有两种残差块：

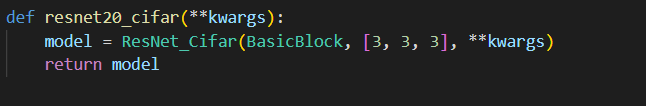
1. BasicBlock: 应用在较浅的网络中，如：Resnet 18和Resnet 34
2. BottleNeck: 应用在较深的网络中，如：Resnet 50 Resnet 101

我实现的resnet代码如下：





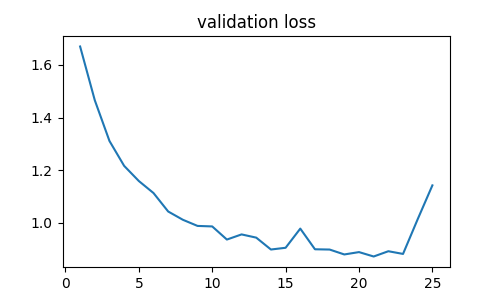
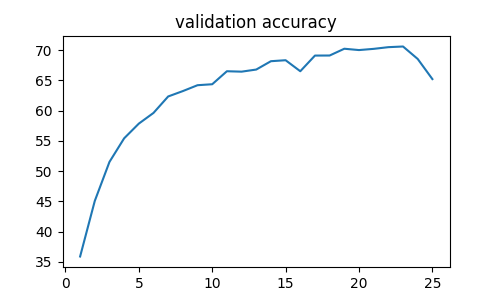




输入的数据x首先经过第一个卷积层、批归一化和ReLU激活函数，然后依次经过三个层（layer1、layer2、layer3），每个层包含3个残差块。接下来，通过平均池化层self.avgpool对特征进行空间降维，然后将其展平成一维向量，最后通过全连接层self.fc进行分类预测。

具体网络结构可见附件ResNet.txt文件。

将该网络训练50轮，得到以下结果：

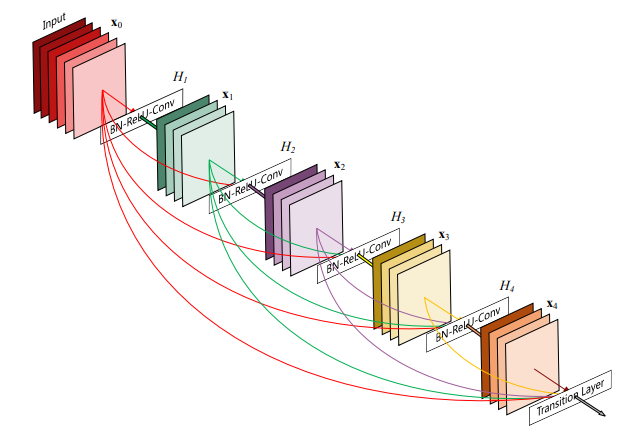


可以看见，在23轮左右，模型达到最佳，之后便开始过拟合，最高的准确率能达到70%左右，即使是20轮的准确率也超过了CNN，模型表现良好。

### DenseNet

与传统的卷积神经网络（CNN）不同，DenseNet通过密集连接的方式使得每一层都可以直接访问前面所有层的特征图。这种紧密连接的设计使得DenseNet在参数利用、特征重用和梯度传播等方面具有显著的优势。

DenseNet的核心思想是密集连接块（Dense Block）。在Dense Block中，每个层的输入来自前面所有层的拼接，这样每个层都可以直接接收到前面层的特征图作为输入。通过这种方式，网络中的信息传递更加充分，每个层都可以直接访问到前面层的特征，从而提高了特征的复用和网络的表达能力。



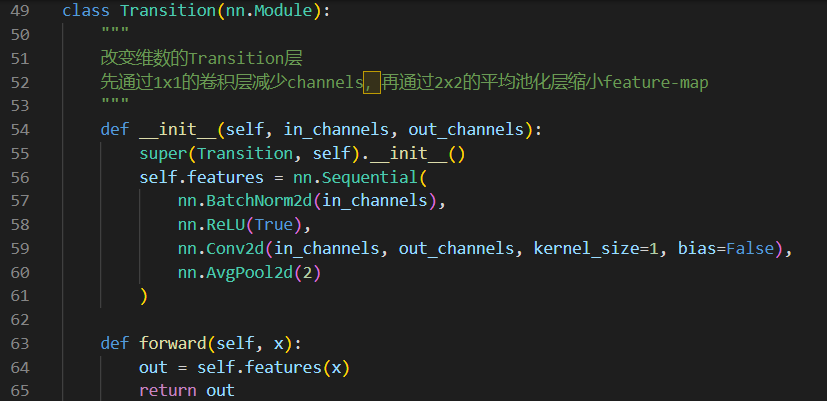
该图源自论文Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 4700-4708.

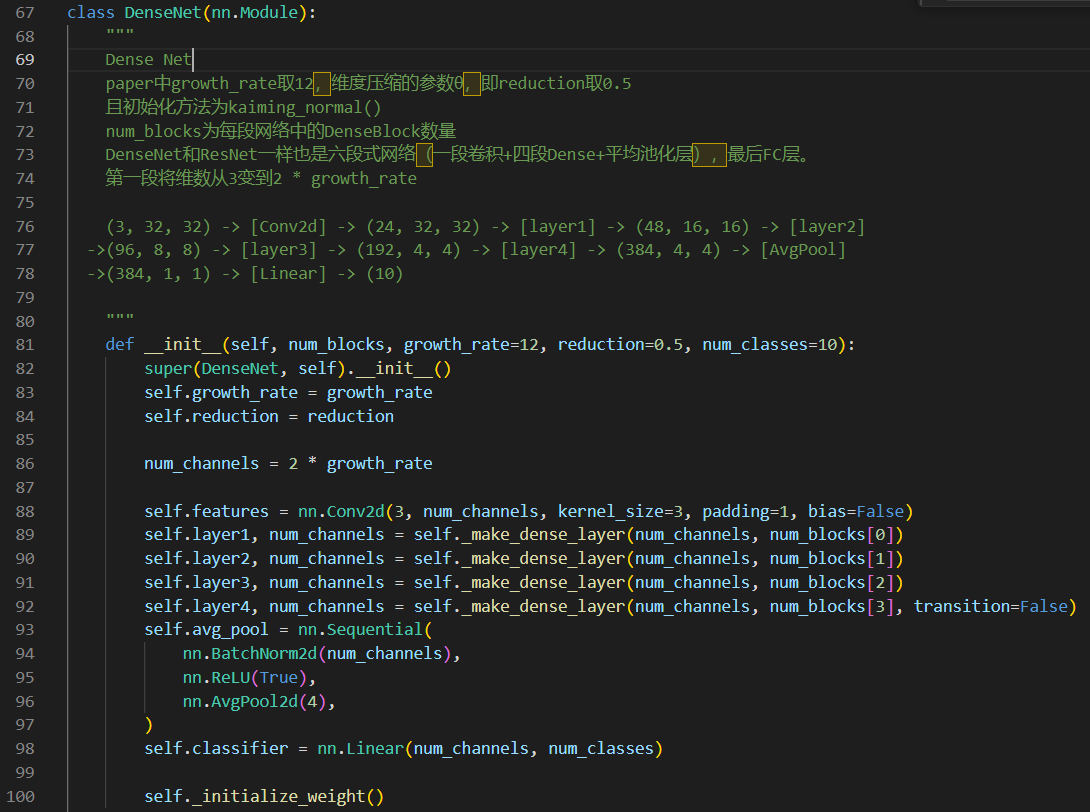
DenseNet有以下突出的特点：

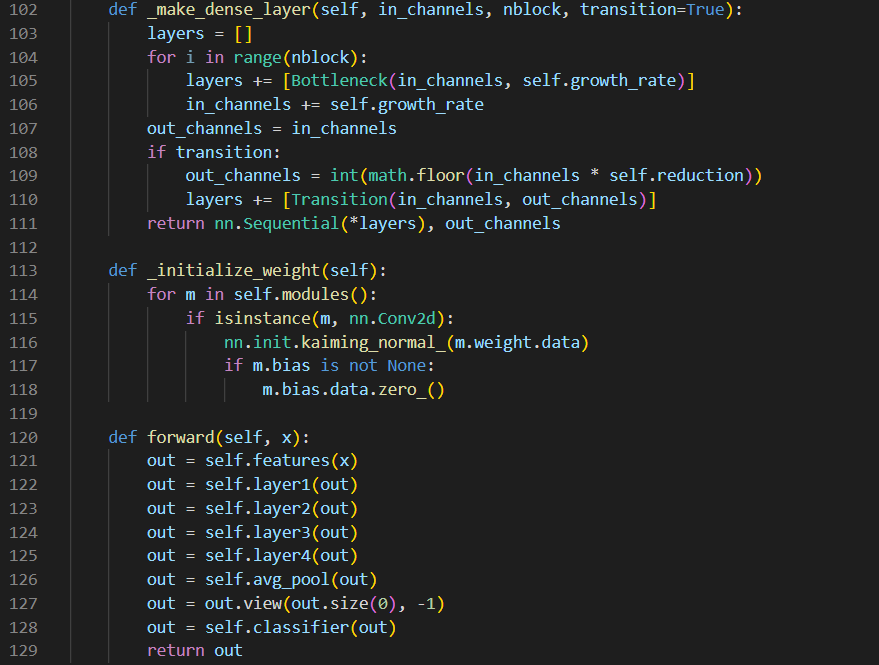
1. **密集连接块（Dense Block）**：Dense Block由多个卷积层组成，每个卷积层的输入是前面所有层的特征图的拼接。这种连接方式增加了网络的深度，同时使得每一层都能够接收到前面层的信息，有利于特征的传递和重用。
2. **过渡层（Transition Layer）**：在Dense Block之间，可以使用过渡层来减小特征图的维度。过渡层通常由一个卷积层和一个池化层组成，它可以减少特征图的尺寸和通道数，从而控制模型的复杂度并加速计算。
3. **稠密连接（Dense Connection）**：通过密集连接，每个层都可以直接接收到前面层的特征图，这种设计有助于特征的传播和信息的融合。相比于传统的网络结构，DenseNet减少了信息的丢失，并提高了特征的利用效率。
4. **全局特征复用**：由于每个层都可以直接访问前面层的特征，DenseNet具有较强的全局特征复用能力。这使得网络可以更好地捕捉到不同层次的特征，从而提升了模型的性能和泛化能力。

我实现的DenseNet如下所示：









具体网络结构可见附件DenseNet.txt文件。

该网络模仿DenseNet论文，growth\_rate取12，维度压缩的参数θ，即reduction取0.5

，且初始化方法为kaiming\_normal()。num\_blocks为每段网络中的DenseBlock数量。DenseNet和ResNet一样也是六段式网络（一段卷积+四段Dense+平均池化层），最后FC层。第一段将维数从3变到2 \* growth\_rate。

维度变化如下：

(3, 32, 32) -> [Conv2d] -> (24, 32, 32) -> [layer1] -> (48, 16, 16) -> [layer2]

->(96, 8, 8) -> [layer3] -> (192, 4, 4) -> [layer4] -> (384, 4, 4) -> [AvgPool]

->(384, 1, 1) -> [Linear] -> (10)

报告内容：

* 老师提供的原始版本CNN网络结构（可用print(net)打印，复制文字或截图皆可）、在Cifar10验证集上的训练loss曲线、准确度曲线图
* 个人实现的ResNet网络结构在上述验证集上的训练loss曲线、准确度曲线图
* 个人实现的DenseNet网络结构在上述验证集上的训练loss曲线、准确度曲线图
* 个人实现的带有SE模块（Squeeze-and-Excitation Networks）的ResNet网络结构在上述验证集上的训练loss曲线、准确度曲线图
* 解释没有跳跃连接的卷积网络、ResNet、DenseNet、SE-ResNet在训练过程中有什么不同（重点部分）
* 格式不限

作业提交：

* 期末前将报告和代码（可将jupyter notebook里代码复制到一个xxx.py文件中）打包（学号+姓名.zip），提交方式另行通知
* 实验报告内容应工整