**学习进度**

本周学习任务：

结合网络资源、ChatGPT及相关论文，学习李沐老师《动手学深度学习（PyTorch版）》，关于**卷积神经网络**部分的内容。

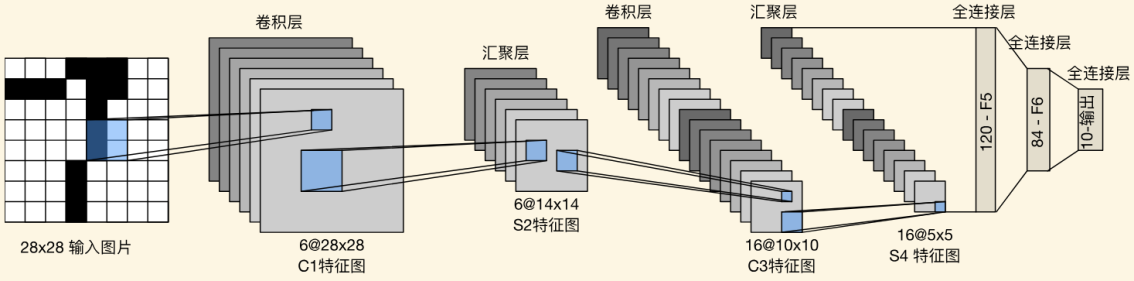
下周学习任务：

学习李沐老师《动手学深度学习（PyTorch版）》，关于循环神经网络部分的内容。

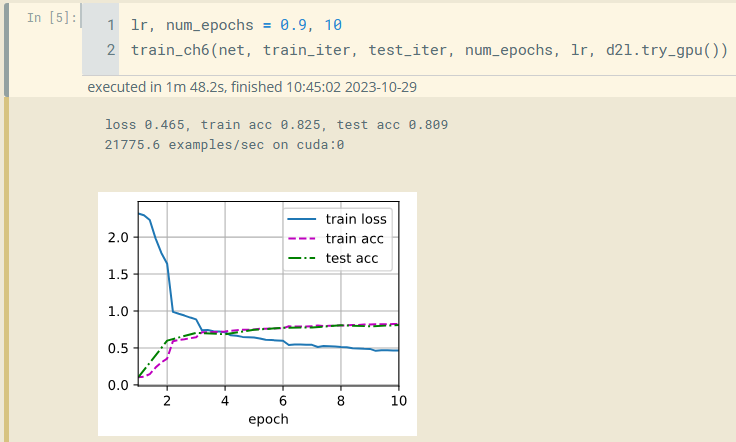
**一、LeNet**

1. 应知必会

LeNet作为卷积神经网络的早期经典模型架构，最初设计用于处理手写字符识别任务，适用于小尺寸图像的处理。然而，由于模型架构较简单且网络深度较浅，不适用于处理大规模数据集及复杂任务，实际效果尚不及同期的多层感知机模型。



1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P245】



1. 相关思考

（1）卷积层代替全连接层的好处

局部感知、减少可学习参数（基于卷积核的参数共享）及平移不变性。

（2）旋转不变性

常规CNN架构缺乏处理图像旋转不变性问题的能力，解决方法：数据增强（缺啥学啥，即：随机旋转图像）；迁移学习（使用具有旋转不变性的大规模数据集投喂而来的预训练模型）；设计旋转不变的特征、网络架构及损失函数。

**（3）特征图与多输出通道**

某卷积层若使用了M个卷积核，则该层输出的特征图数及通道数亦为M，即：每个卷积核在卷积计算后，都能生成一个相应的抽象特征图。由此，随着网络的加深，特征图维度将趋于：愈小的空间尺度，愈大的通道深度（不同通道对应的特征图可以看作不同类型的抽象特征）。值得注意的是，不同通道的特征图并不是独立学习生成的，而是共同优化而来。

**（4）如何理解特征图中的通道？**

代码实践中的特征图为四维：图像批次数、高度、宽度及通道数。对于一张普通的图片而言，通道数一般为3，分别表示RGB值，用于表达对应像素点的颜色信息。但是如果把Alpha Go理解为图像分类任务，则某时刻的棋盘对局状况则象征特征图，此时的特征图通道还是用来表示颜色的吗？显而易见的是，不同通道表征的信息，可以人为指定。以Alpha Go为例，可以根据专家经验，设定不同通道上对应位置的值表示：对应棋盘位置上是否有棋子、是否为黑子、左边是否有白子及右边是否有白子等信息。

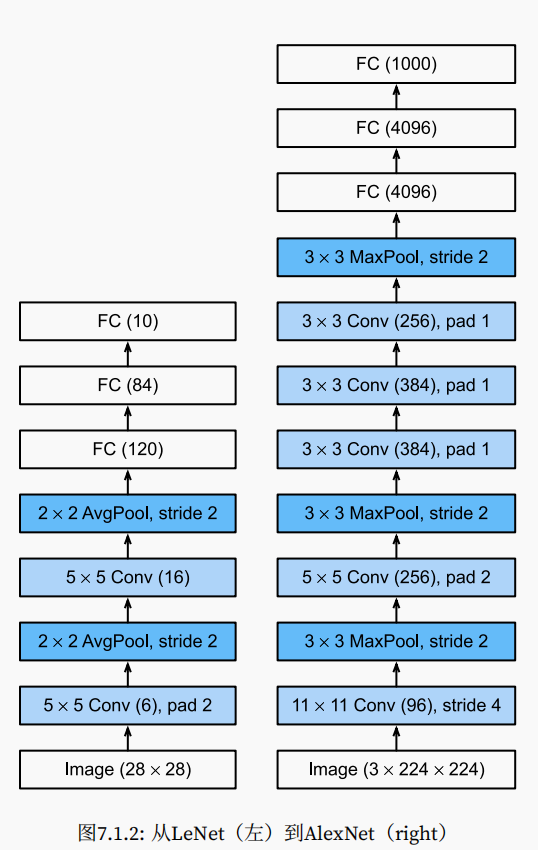
（5）填充与步幅

Padding能够防止图像边缘像素信息的丢失，亦能维持卷积后特征图的空间维数。Stride通过定距跳步的方式缩减采样次数，亦能降低卷积后特征图的空间维数。

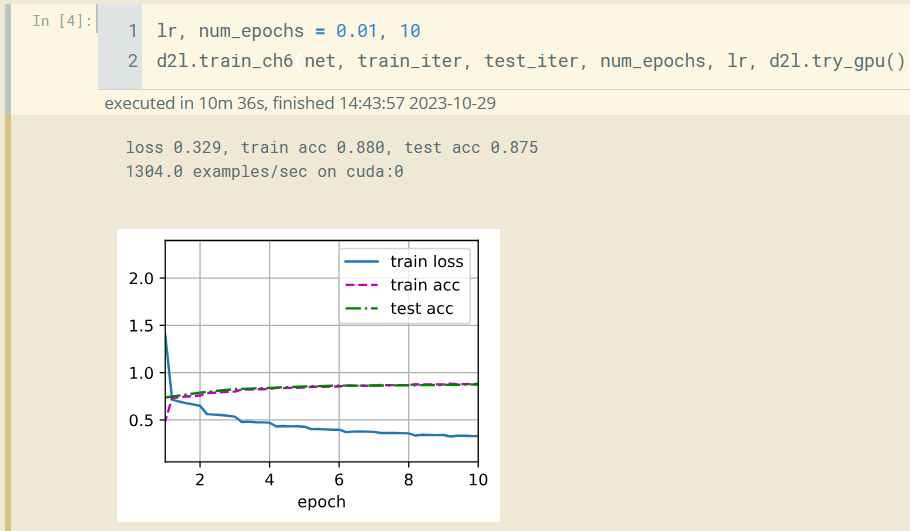
**二、AlexNet**

1. 应知必会

为提升处理复杂视觉任务的能力，AlexNet在LeNet的基础上，采用了更为复杂的深层网络架构，它是从浅层架构走向深层架构的早期尝试。相较于LeNet，AlexNet的创新点是：ReLU+MaxPool+Dropout、分组训练（基于多个GPU）、局部响应归一化。



1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P254】



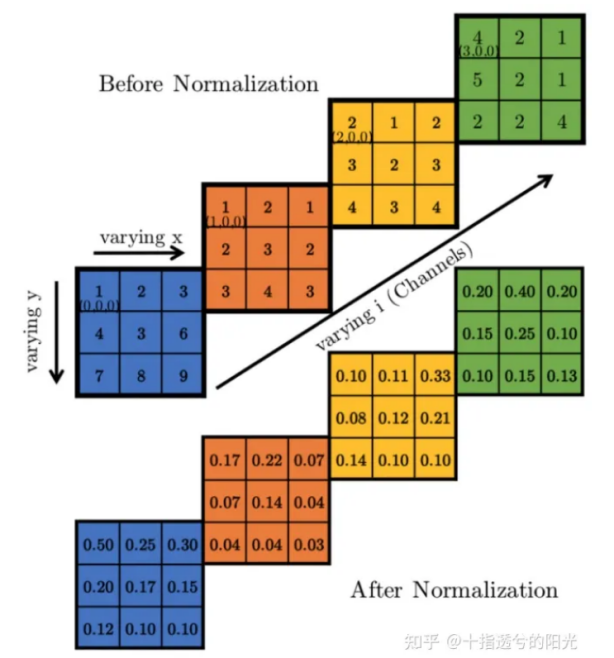
1. 相关思考

**（1）视觉数据的逐级表征**

浅层卷积通常用于提取图像中的具体局部特征：物体边缘、颜色及纹理；深层卷积则在浅层输出的基础上，进一步提取更大的物体特征；而网络的更深层则在更为广阔的感知域内，提取抽象特征，用于更准确地辅助完成目标预测任务。

1. 如何理解创新点：局部响应归一化（Local Response Normalization，LRN）

LRN依赖相应邻域范围内的所有特征值，对指定特征值进行平方归一化操作，从而通过增强局部对比度的方式，达到了提高模型泛化能力的目的。AlexNet模型采用的Inter-Channel LRN如下图所示，邻域定义在通道Channel维度上。当设置邻域范围超参n=2时，若需归一化橙色特征图上左上角(1, 0, 0)位置的像素值，则需要前一通道(0, 0, 0)位置上的像素值1、本通道对应位置的像素值1、后一通道(2, 0, 0)位置上的像素值2，根据LRN公式计算出本通道对应位置的归一化结果值0.17。

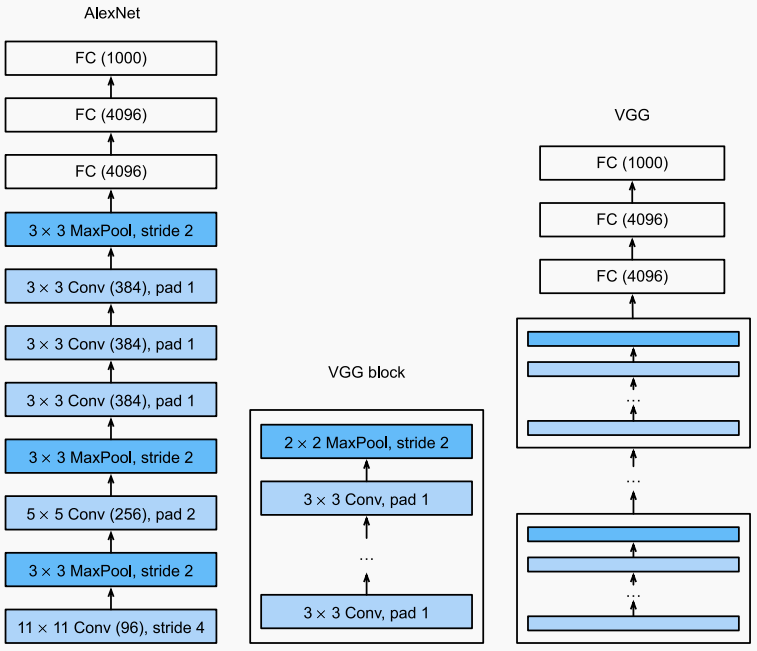


值得注意的是，LRN在后续网络架构中并不常用，被其它正则化方法或激活函数代替。

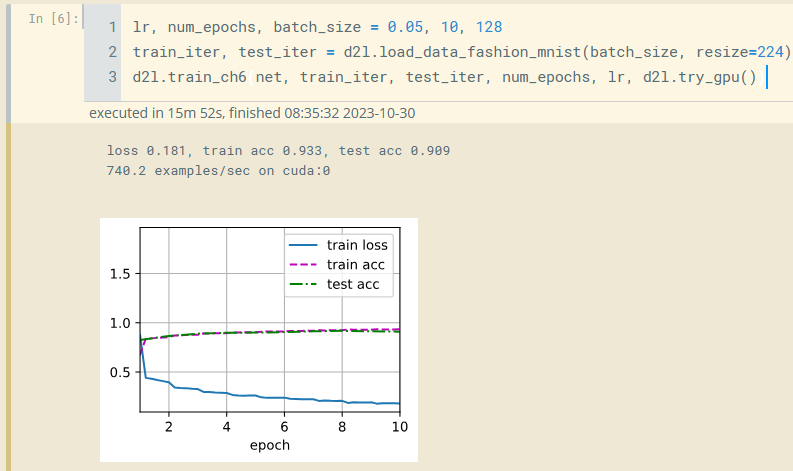
**三、VGG**

1. 应知必会

VGG引入模块化思想，通过在模型构建过程中灵活重复使用自定义卷积块的方式，保证了深层网络架构的简单有效性。由于较大规模的模型尺寸，VGG获得了更为优秀的复杂任务处理能力，代价是内存和计算资源的要求更高。相较于AlexNet的大尺寸卷积核，VGG Block连续采用多个更小的卷积核代替大卷积核，在同等感知域内以更小代价学习到了更复杂的抽象特征信息表征，从而走向了更深！



1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P257】

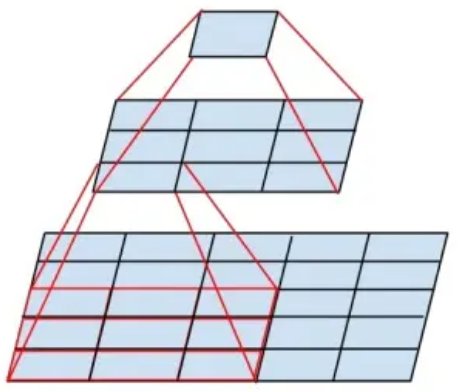


1. 相关思考
2. **为什么很多预训练方法采用VGG模型架构？**

相较于前述模型，VGG模型能够高效利用有限参数空间，从而更好地学习数据特征。相较于后续模型架构，VGG又是性能与计算效率的优秀平衡结果。此外VGG模型具有广泛的研究支持，更易获得处理特定任务的优良预训练模型，是网络改造的首选基础网络。

1. VGG Block中连续堆叠的小卷积核，为什么能够代替更大的卷积核？

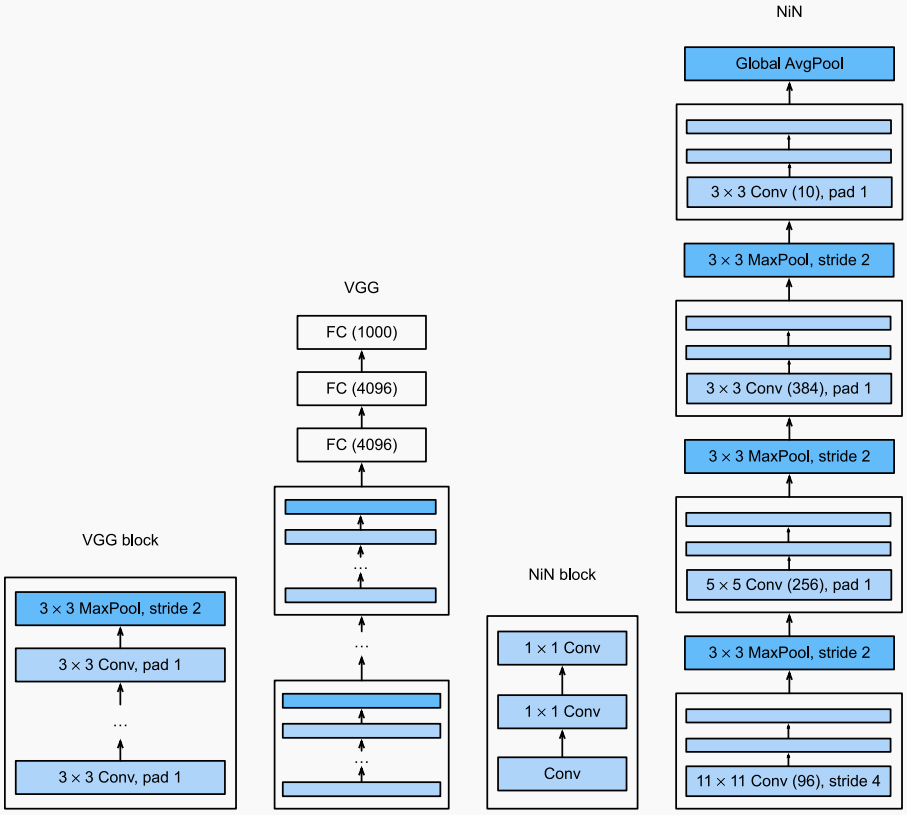
如下图所示，可以通过连续堆叠两个3\*3卷积核的方式，代替一个5\*5卷积核。好处是：减少参数量（2\*3\*3<5\*5）、增加网络深度（较深的网络能抽取更抽象的特征，以提高性能）、增加非线性（更多的小尺寸卷积核将引入更多的非线性激活函数，有助于复杂特征的学习）。



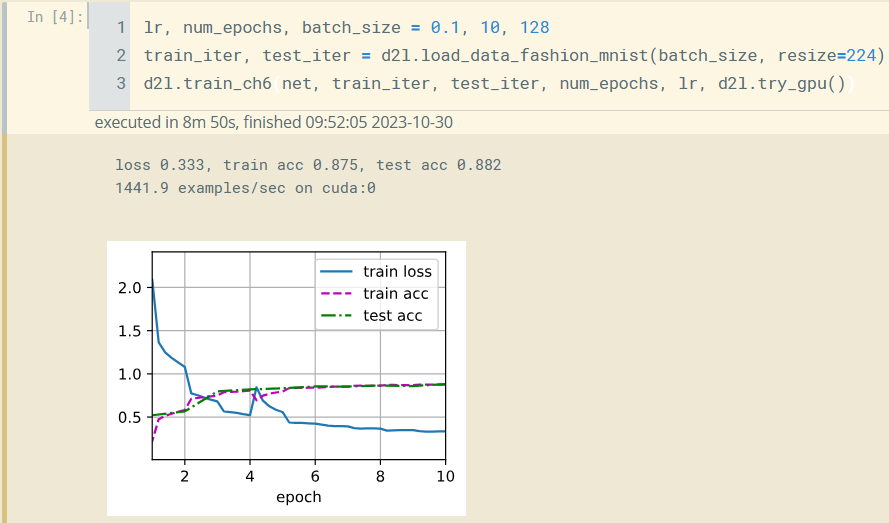
**四、NiN**

1. 应知必会

NiN模型通过在每个像素通道上分别使用“多层感知机”的方式，在网络浅层中引入了“全连接层”，从而获得了更好的映射表达能力，并保证了图像空间结构的表征。NiN同样应用了模块化思想，与VGG不同的是：NiN Block固定了两个1\*1卷积层（等价于逐像素全连接层）、最后的三个全连接层被替换为全局平均池化层。由此，NiN优势在于：更强的网络表达能力、更少的可学习参数。缺陷在于：引入多层感知机，增加了网络的计算复杂性，亦需要更多的数据支持；更加依赖调参质量。

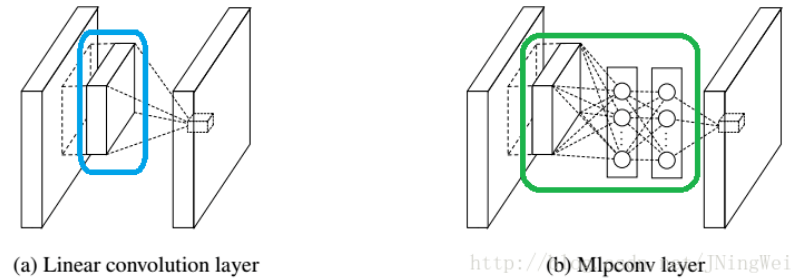


1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P262】



1. 相关思考
2. **如何理解NiN Block?**

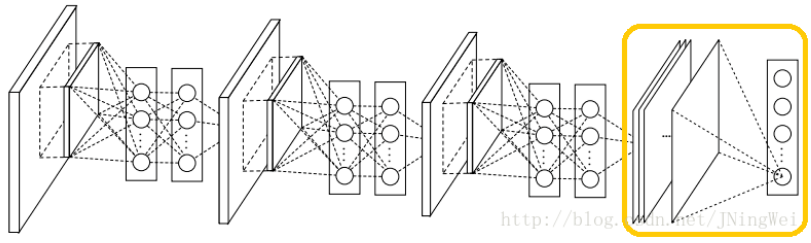
自定义的NiN Block固定了两个1\*1卷积层，在使用NiN Block时：首先需要进行人为控制的卷积操作，以获取不同位置上的局部信息；然后将特征图中的每个像素看作一个样本，像素的不同通道看作样本特征；最后通过1\*1卷积层对每个像素样本进行加权求和处理，相当于进行了全连接操作。



将线性卷积块（卷积+非线性激活+池化）替换为多层感知机块（卷积+固定的1\*1卷积），不仅赋予了网络层之间更强的映射表达能力，还极大地减少了参数数量（1\*1卷积能有效控制特征图的通道维数）。该设计后来为ResNet和Inception等网络模型所借鉴。

（2）如何理解最后的全局平均池化（Global AvePool）？

在最后一次引用NiN Block之后，不再接续作为惯例的全连接层，也不接续NiN网络之前使用的MaxPool，而是引入全局平均池化层Global AvePool。全局池化层的实现方式是： nn.AdaptiveAvgPool2d，能够自适应输入特征图的大小，平均池化输出特定空间维度的预测结果。若是处理M分类任务，应为nn.AdaptiveAvgPool2d((1, M))，以便输出空间维度为1且通道数为M的预测结果（不同分量表示为判定不同类别的得分）。

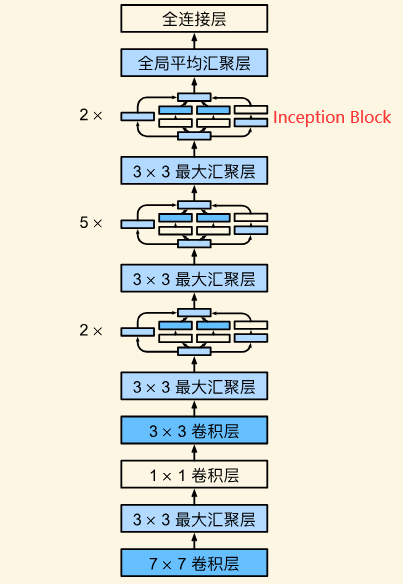


**五、GoogLeNet**

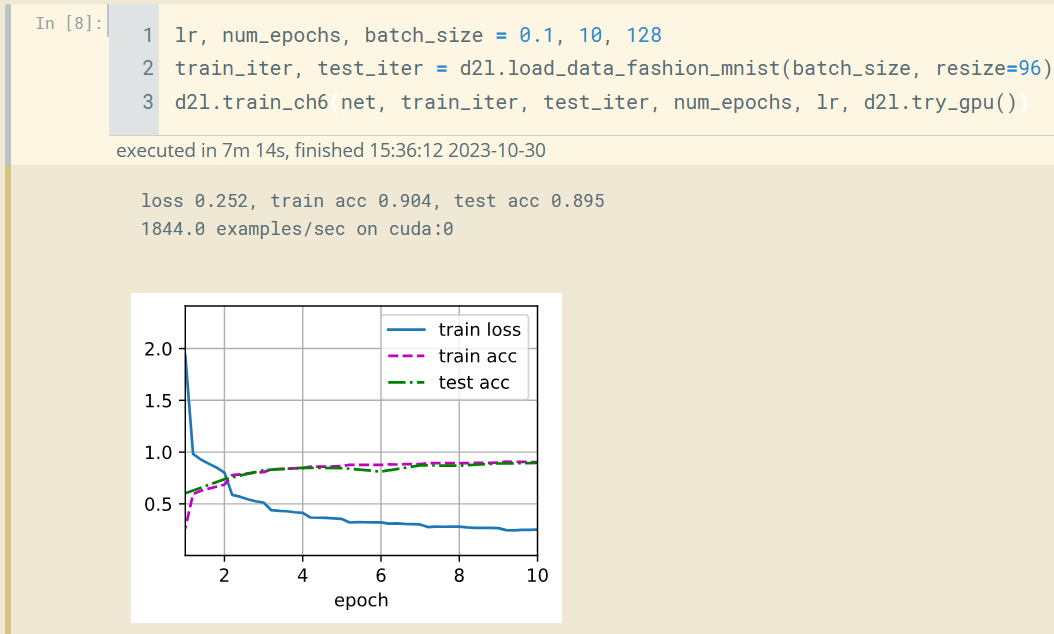
1. 应知必会

由于不同大小的卷积核，可以有效识别不同范围的图像细节。GoogLeNet基于串、并联及模块化思想，灵活使用自定义Inception Block，通过联结多种不同卷积路径的方式，并行获取不同空间尺度上的抽象特征信息，解决了什么样大小的卷积核才是最合适的问题。相较于前述模型架构，GoogLeNet依赖如下手段，极大提高了模型性能：通过Inception Block的并行卷积路径，增加了网络宽度及多尺度适应性；利用全局平均池化层部分替代全连接层，极大地减少了参数量；引入辅助分类器，缓解了前几层的梯度消失问题；通过Inception Block、卷积层、池化层及全连接层的堆叠，增加了模型深度。

缺点是：网络架构相对复杂，较难调整且特征表示缺乏可解释性。

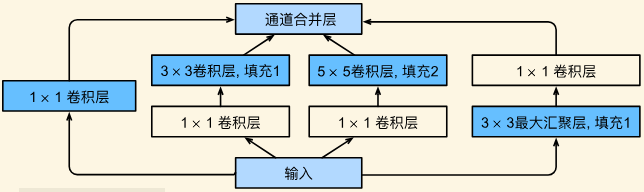


1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P265】



1. 相关思考
2. 如何理解Inception Block？

传统的Inception Block由四条并行路径构成，每条路径都存在一个1\*1卷积层，能够削减通道数。对于第二、三条路径而言，输入数据通过1\*1卷积层后，还需分别进行不同方式的卷积计算，以获得不同空间尺度上的特征信息。对于第四条路径而言，在1\*1卷积层之前又引入了一个最大池化层，能够削减特征图的空间维度，提取不同位置上的局部突出信息。值得注意的是：虽然，四条并行路径的卷积方式不同。但是，都需要分别设计合适的填充方式，保证输出特征图空间维度上的一致性，从而降低通道合并层的复杂程度。由此，基于自定义Inception Block构建而成的GoogLeNet模型，能够以更有效的方式捕获多尺度的特征信息，帮助学习更具泛化能力的特征表示。



1. **使⽤GoogLeNet的最小图像⼤小是多少？**

使用GoogLeNet模型处理的最小图像尺寸，应不小于224\*224像素。若图像尺寸过小，Inception Block在提取不同尺度上的特征信息时，由于分辨率不足，将导致信息丢失问题。如若必须使用GoogLeNet模型处理小尺寸图像，应考虑将图像进行放大或者填充。

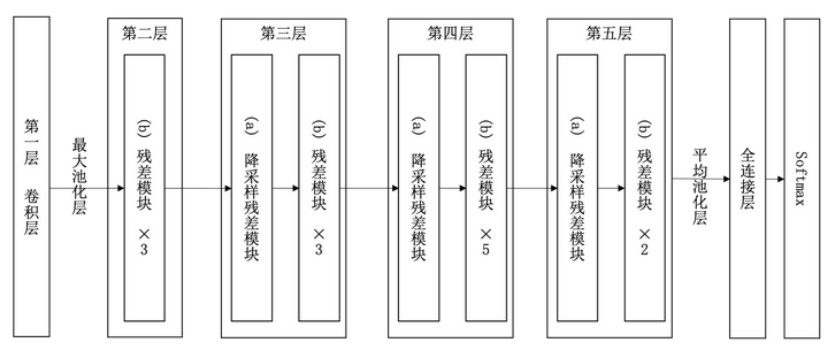
1. **比较AlexNet、VGG、NiN和GoogLeNet模型的参数数量**

AlexNet作为经典的早期深度卷积神经网络架构，约包含60百万左右的参数，大部分参数来自于全连接层。VGG引入模块化思想，简便地构建了一个更深的网络架构，约包含138百万左右的参数。NiN则是一个轻量级网络架构，通过使用全局平均池化及1\*1卷积层极大地减少了参数数量，通常约包含50万个参数。而GoogLeNet在NiN的基础上，通过自定义Inception Block、1\*1卷积及全局平均池化等方式，参数缩减到了约40万左右的规模。后两者参数规模显著缩减的关键因素在于：贡献大量参数的罪魁祸首“全连接层”被替代。

**六、ResNet**

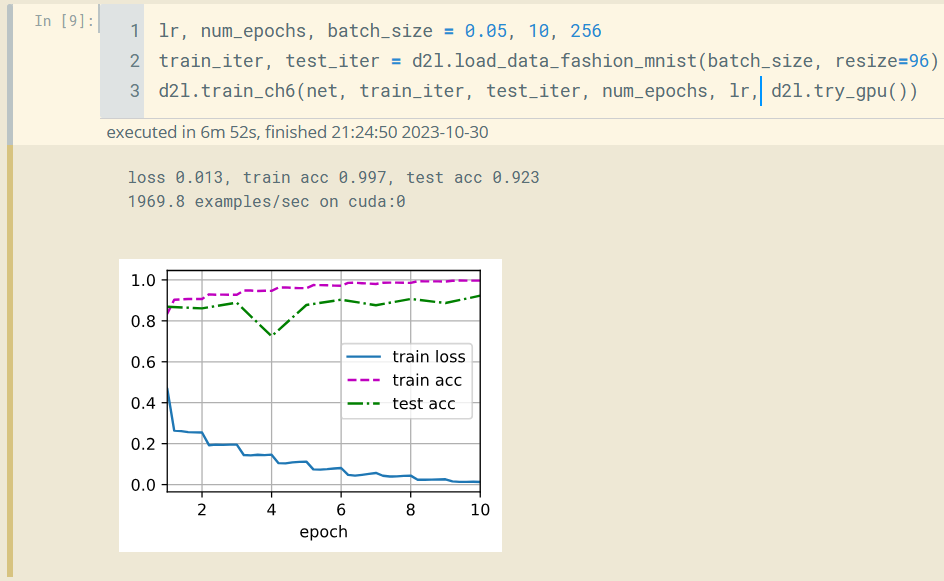
1. 应知必会

为解决深层网络架构的退化问题，ResNet模型通过自定义Residual Block引入线性转换分支的方式，企图学习线性转换和非线性激活之间的平衡状态，从而达到能够有效使用更深层网络架构以获取更优性能的目的。具体而言，自定义Residual Block通过跳跃连接的方式，尝试学习原始输入与最终输出之间的残差（差异），并依赖可学习权重控制原始输入信息与残差信息的传递保留程度，从而避免梯度消失问题。



总而言之，ResNet模型在保证参数效率的前提下，依赖引入跳跃连接的自定义Residual Block，通过有效扩展网络深度的方式，获取了更好的泛化性能。缺点是：更深的网络架构，具有更大的计算复杂度；更复杂的自定义卷积块，导致调参工作的复杂性。

1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P279】



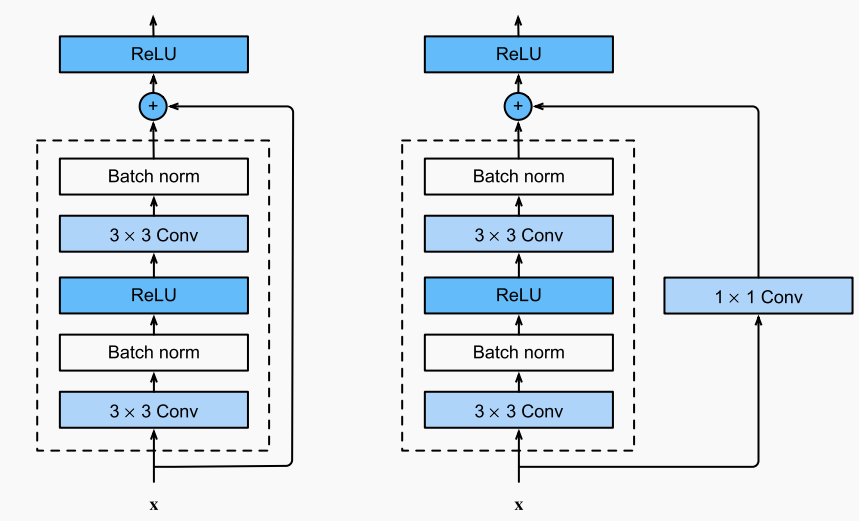
1. 相关思考
2. **如何理解退化现象（Degradation）？**

关于“越深的网络，准确率越高”的直觉似乎存在问题！随着网络深度的逐渐增加，准确率到达峰值之后，将发生退化现象。研究表明：虽然引入非线性激活，能够极大地提高数据分类能力。但是随着网络深度不断地增大，更多的非线性激活使数据映射到愈加离散的空间中，从而难以实现线性转换。也可以理解为：网络愈深，梯度愈难收敛。由此，通过构建更深网络架构来获得更优泛化性能的方式，存在退化现象的制约。

1. 如何理解残差块（Residual Block）？

自VGG以来的后续卷积神经网络，都引入了模块化思想，通过自定义卷积块的方式，将模型架构走向更宽更深层，从而获得更好的泛化性能。显而易见的是，ResNet模型架构的主要创新点都凸显在Residual Block之中。

如下图所示，ResNet模型架构中存在残差模块（图左）及降采样残差模块（图右），后者在加权合并卷积前后特征图之前，对原始输入特征图进行了1\*1卷积处理，用于调整原始输入特征图的分辨率及通道数。此外，每个Residual Block都包含两个分支，即：恒等映射分支及残差学习分支，前者用于传递原始输入，后者用于学习输入、出之间的残差信息。

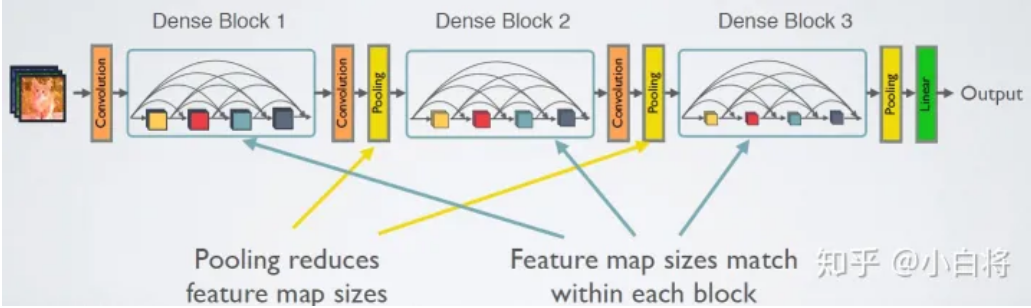


在ResNet模型架构中，通常将降采样残差模块及残差模块组合使用，用于适应不同尺度的特征信息。此外，相较于VGG及GoogLeNet等模型架构，自定义Residual Block中的另一个显著区别是：在每个卷积层后增加了批量归一化层，主要用于调整传输数据的分布，从而加快模型收敛的速度。

**七、DenseNet**

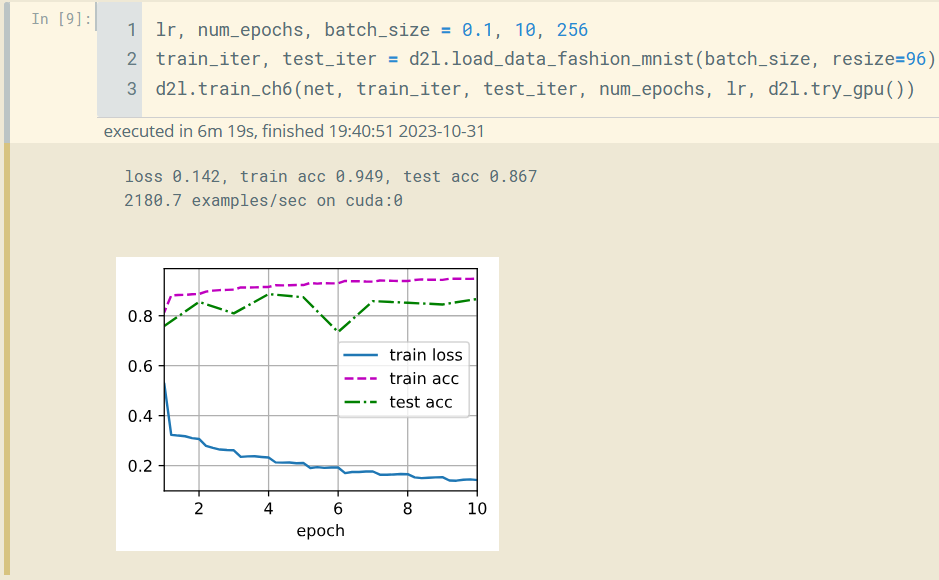
1. 应知必会

主要由稠密块Dense Block及过渡层Transition Layer构成的稠密连接网络（DenseNet），本质上是ResNet的逻辑扩展。依赖Dense Block内部的连结结构，实现了块内各层之间的密集连接，使得各层提取的特征图，可以更加自由地在块内的网络层之间传递，从而提高了特征的利用效率，一定程度上缓解了深层网络必然面临的梯度消失问题。



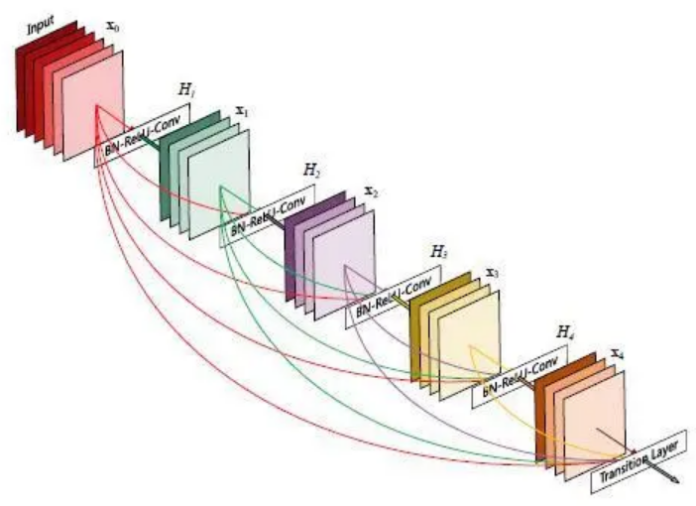
相较于ResNet，DenseNet模型由于特征重用及参数共享特性，能够更有效地利用有限参数空间，从而走向了更深层，提高了模型的泛化性能。缺点是：特征重用可能会导致部分冗余特征的传递、内存占用量大（每一层都需要接收前面所有层的特征图作为输出）。

1. 实验结果【代码省略，详见：李沐《动手学深度学习（PyTorch版）》P286】

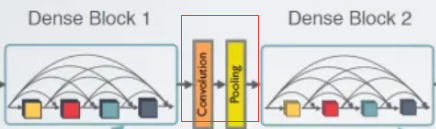


1. 相关思考
2. 如何理解稠密块与过渡层？

**稠密块Dense Block**的结构如下图所示，其中每一层的输入都来自于前面所有层的输出。若网络架构存在M层，传统CNN模型有M个连接，DenseNet模型则有个连接。与Residual Block将残差与原始输入信息进行加权求和的操作方式不同；Dense Block中的每一层都接收前面所有层的输出，然后通过连结（concatenate）组合这些特征图。连结结构能在保持参数量级不至于增长太快的前提下，将网络结构扩展到更深度。此外，每一层的卷积方式源自Residual Block卷积，卷积逻辑是：Batch Normalization + ReLU + 3\*3 Conv。



稠密块Dense Block之间，存在一个由瓶颈层（Bottleneck layer，1\*1卷积层）及池化层构成的**过渡层Transition Layer**，主要用于控制特征图的维数，从而保证网络的计算效率，减少过拟合的风险。此外，由于稠密块内部的密集连接需要保证各层输出特征图维度的一致性，直接在内部插入池化层是不可行的。由此，在稠密块之间引入包含池化层的过渡层，是保证网络训练效率的必然选择。



1. 为什么过渡层使用平均池化层而不是最⼤池化层？

使用平均池化层是为了更好地：抽取全局信息、平滑特征图（缓解噪声干扰）、降低参数数量、减小过拟合风险，从而提高DenseNet模型的泛化性能。

1. **如何理解DenseNet模型的参数⽐ResNet小？**

相较于同等深度的ResNet模型，DenseNet模型利用基于密集连接的共享特征提取、1\*1 卷积及平均池化层等方式，极大减少了等深网络的参数数量，从而提高了训练效益。

**八、ViT**

针对于计算机视觉任务，存在CNN同注意力的结合，存在引入注意力替代CNN组件，亦存在基于注意力机制来处理图像数据的深度学习模型，比如：ViT模型。

ViT模型的核心创新点是：在计算机视觉领域，可以摆脱对CNN的依赖，直接使用Transformer Encoder展开图像处理工作。尤其是使用依赖海量数据预训练而成的ViT模型，迁移处理中小型图像集的识别任务时，效果不逊色于最先进的卷积神经网络。

1. ViT模型的网络架构

基于Transformer Encoder架构，处理图像数据的过程，如下图所示。

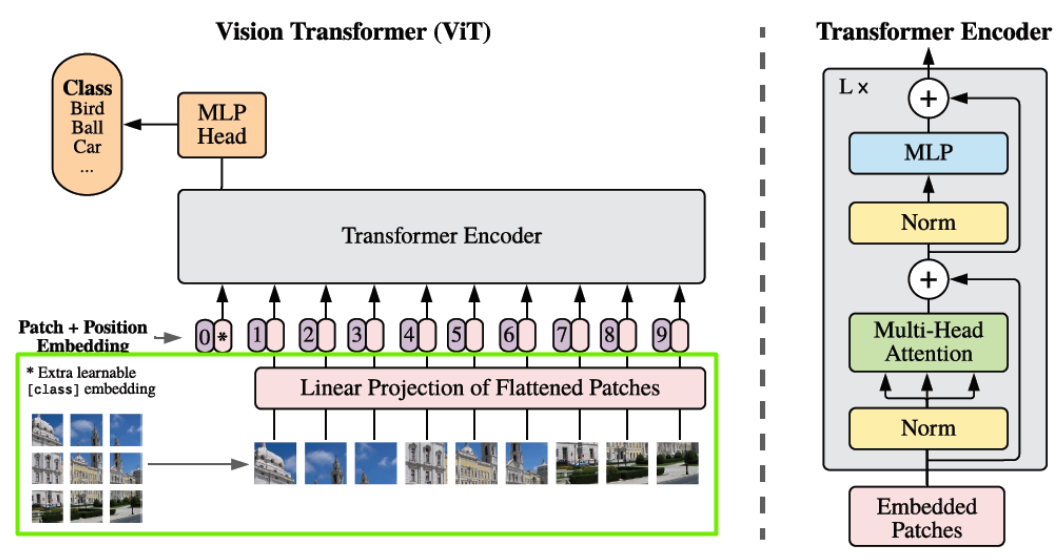
✸ 首先，将图片分割成固定大小的9个补丁；

✸ 其次，将所有补丁输入到Embedding层进行线性嵌入，从而获得对应的Token向量；

✸ 然后，在序列首部添加一个额外的可学习Class Token向量（图中的\*），且所有的Token向量都需引入对应的位置编码（图中的0-9），用于图像分块位置信息的表征；

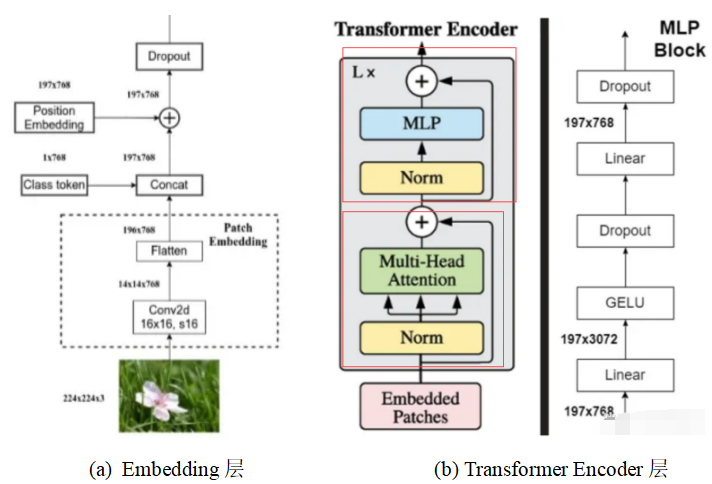
✸ 此外，将由所有Token向量构成的Patches序列，输入到重复使用的Transformer Encoder层中，用于处理和提取图像的抽象特征；

✸ 最后，将经过Encoder层处理后的Class Token分量，提取并投喂给MLP Head，从而获得最终的预测结果。



1. Embedding层（Linear Projection of Flattened Patches）

如下图左所示，在具体的代码实践中：首先通过一个卷积层（Stride=16,conv\_size=16, kernel\_num=768），将一个224\*224\*3维的图片进行卷积及展平处理，获得了一个196\*768维的序列；然后，在序列首部添加一个等长的token向量；最后，序列中的每个分量都引入对应的位置编码，从而获得最终的197\*768维序列。



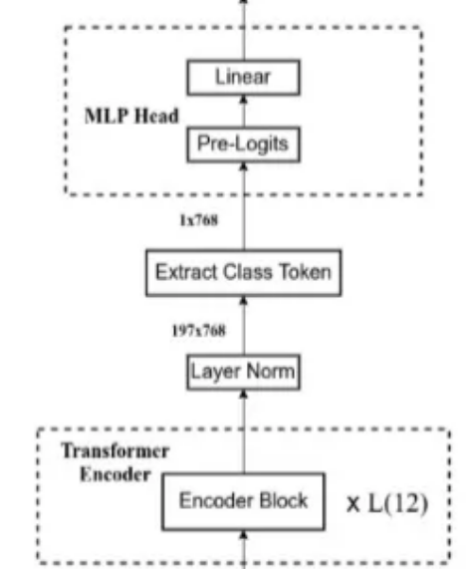
1. Transformer Encoder层

Transformer Encoder层网络架构，如上图右所示。首先：将Embedding层输出的Patches序列进行归一化处理，投喂至多头注意力模块中学习残差信息并施行Dropout操作，依赖跳跃连接将残差信息同原始序列按权加和。然后：将前层输出进行归一化处理，投喂至MLP Block中学习残差信息并施行Dropout操作，依赖跳跃连接将残差信息同原始序列按权加和，从而获得对应的Encoder输出。最后：将上次Encoder的输出序列作为输入，重复执行L次前述步骤，从而获得最终的Encoder输出。

MLP Block的网络结构，如上图右所示：全连接层Linear（节点个数扩大四倍）、GELU激活函数、Dropout层、全连接层Linear（结点个数还原）、Dropout层。MLP Block作用是，对每个位置上的特征进行非线性变换映射，以帮助模型更好地学习图像的抽象特征信息。

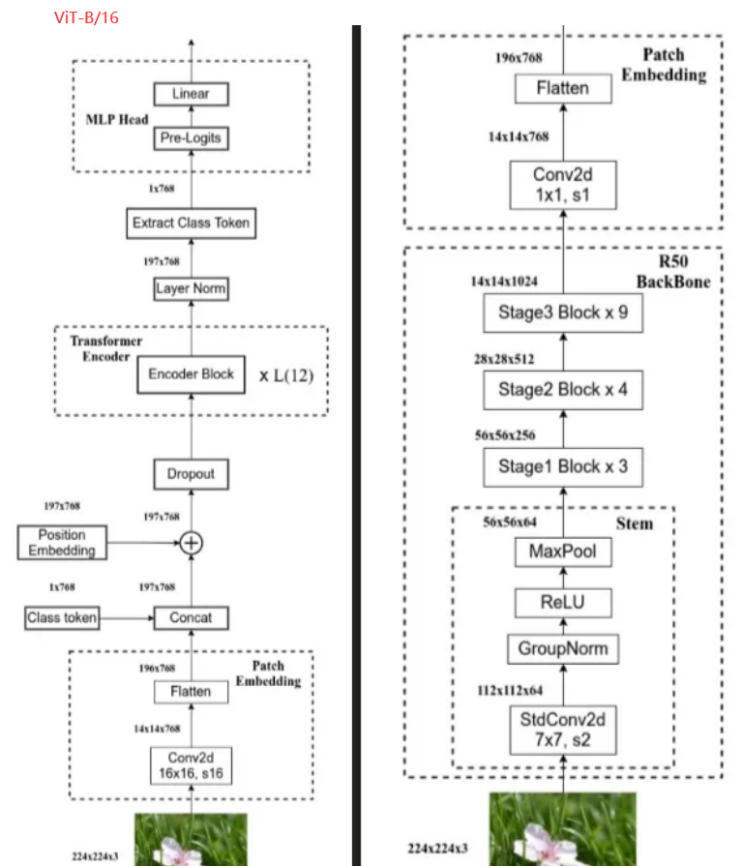
1. MLP Head层

将重复L次TransFormer Encoder处理的Patches序列，进行归一化操作后，切片提取序列中的Class token分量，紧接着通过MLP Head层得到最终的输出。其中，MLP Head层由Pre-Logits层（全连接层+tanh激活函数）及全连接层构成。在MLP Head的Pre-Logits中，首先将Class token的抽象特征，映射到更高维度的特征空间中；然后依赖激活函数进行非线性转换，从而更好地捕捉特征之间的复杂关系，以提高模型的表达能力。在MLP Head的Linear中，将Pre-Logits层输出的高级抽象特征映射为模型输出的格式，比如类别概率分布。



1. 基于ResNet+ViT的Hybrid混合模型

基于ResNet+ViT的Hybrid混合模型，使用基于CNN架构的ResNet作为骨干网络Backbone，用于提取原始输入图像的抽象特征表示，然后通过前述的ViT架构，获得最终的预测结果。R50 Backbone架构，如下图右所示。首先使用StdConv2d卷积层、GroupNorm归一化层、ReLU激活层及MaxPool层，处理原始输入图像；然后依次通过由多个残差块灵活构成的Stage1~3，继而抽取出更高效的抽象特征表示；最后继续使用前述介绍的ViT-B/16网络（下图左所示），处理表征图像特征信息的输入序列即可。



**总 结**

在计算机视觉领域，深度学习算法的发展经历了多个里程碑式的模型，每个模型都有其独特的技术路线、创新点及优势作用域。以下是前述模型的简要总结：

**LeNet**首次引入了卷积神经网络的概念，采用卷积层与池化层的组合，实现了手写字符识别任务。LeNet适用于小尺寸图像处理，在处理大规模数据集及复杂任务时，效果较差。**AlexNet**作为从浅层走向深层架构的早期尝试，采用了ReLU+MaxPool+Dropout、分组训练及局部响应归一化等创新性手段，企图利用更深的网络架构，提升处理复杂视觉任务的能力。**VGG**引入模块化思想，通过自定义VGG Block将大尺寸卷积核替换为连续的若干个小尺寸卷积核，提高了有限参数空间的利用效率，从而走向了更深层。VGG在处理大规模数据集上的图像分类任务时，表现出色。**NiN**则通过自定义NiN Block引用1\*1卷积层的方式，反惯例的在网络浅层中使用了“多层感知机”，从而获得了更好的映射表达能力。此外，NiN为了走向更深层，创新性地使用全局平均池化层代替了末尾的全连接层，极大减少了等深度网络的参数数量。NiN的优势作用域：图像分类及对象检测，对小目标检测有一定优势。**GoogLeNet**通过自定义Inception Block，并行获取不同空间尺度上的抽象特征信息，增加了网络宽度及多尺度适应性，解决了什么样大小的卷积核才是最合适的问题。GoogLeNet的优势作用域：图像分类和对象检测，特别在计算资源有限的情况下表现出色。**ResNet**通过自定义引入残差连接的Residual Block，缓解了深层神经网络的梯度消失问题，从而走向了更深层。ResNet的优势作用域：图像分类、目标检测及图像分割等任务，特别擅长深层网络训练。**DenseNet**通过自定义引入密集连接的Dense Block，具备特征重用性质，从而进一步地缓解了梯度消失问题。DenseNet的优势作用域：图像分类、目标检测，通常具有更高的参数效率。**ViT**摆脱对CNN的依赖，直接使用Transformer Encoder架构展开图像处理工作。尤其是使用依赖海量数据预训练而成的ViT模型，迁移处理中小型图像集的识别任务时，效果不逊色于最先进的卷积神经网络。ViT的优势作用域：图像分类、对象检测及图像分割，特别适用于序列数据（如：自然语言处理和计算机视觉任务的跨模态应用）。

在计算机视觉领域，除了前述稍微提及的：CNN路线、注意力机制路线及迁移学习路线，还存在如下技术路线：目标检测路线（Faster R-CNN、YOLO、SSD）；图像分割路线（U-Net、SegNet、DeepLab）；GAN路线(DCGAN、CycleGAN、StyleGAN)及强化学习路线（DQN、A3C、DDPG）。前述技术路线及典型模型架构，代表了深度学习在计算机视觉领域的多个关键应用方向，每个路线都适用于不同类型的问题和任务。