在 Fashion-MNIST 数据集上的性能表现。

2.2 现代卷积神经网络

2.2.1 AlextNet —— 飞跃

类似于 LeNet 这样的浅层小规模神经网络在较小的数据集上取得了与 SVM 相媲美的结果，但是在大型数据集上仍然无法取得优势。原因在于传统模型通常使用了人工精心设计的特征描述子，如，LBP、SIFT、SURF、HOG，但是神经网络模型则处理的是原始的图像像素信息。

因此，势必要赋予神经网络模型更多的使命，比如，进行特征表示学习。这样一来，就需要增加模型容量，进而对算力提出更高的要求，对据量也应该具有一定的规模。

21世纪，已经具备了 ImageNet 这样大规模的图像数据集(2009CVPR)，采用众包方式进行了图片类别标注。另外，算力方面，Nvida 公司把原本用来图形加速的 GPU 做成了面向通用并行计算的 GPGPU (general-purpose GPUs)，并有了配套的编程运算平台库 CUDA。使得卷积神经网络中大量存在的卷积运算和矩阵乘法运算能够借用通用 GPU 快速运算。

2012 年，AlexNet 在 ImageNet 图像分类竞赛上拔得头筹，所取得的图片分类精度远超第二名，首次表明了特征表示学习可以超越人工特征描述子。AlexNet 和 LeNet 的架构非常相似，其结构如下：

- 8 个卷积层：使用 ReLU 激活函数和最大汇聚层。ReLU 具有计算简单，梯度稳定的特点。
- 采用 dropout 控制全连接层的模型复杂度。
- 训练时增加了大量的图像增强数据，如翻转、裁切和变色。增强了模型的健壮性，有效减少过拟合。

从历史发展的角度来总结一下 AlexNet (摘抄自d2l.ai)：

- AlexNet的架构与LeNet相似，但使用了更多的卷积层和更多的参数来拟合大规模的ImageNet数据集。
- 今天，AlexNet已经被更有效的架构所超越，但它是从浅层网络到深层网络的关键一步。

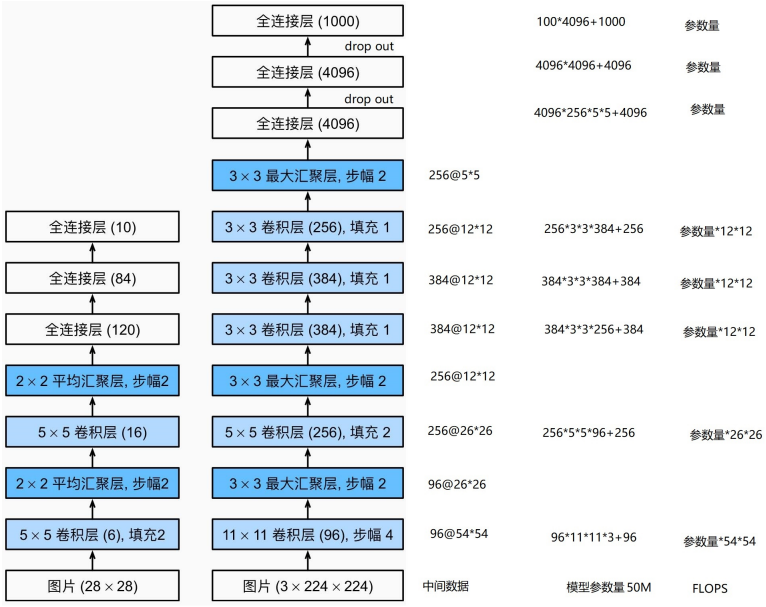

图 2.4: AlexNet (右) 与 LetNet (左) 的结构对比图。

- 尽管AlexNet的代码只比LeNet多出几行，但学术界花了很多年才接受深度学习这一概念，并应用其出色的实验结果。这也是由于缺乏有效的计算工具。
- Dropout、ReLU和预处理是提升计算机视觉任务性能的其他关键步骤。

作业：实验代码测试

可参照 d2i.ai，在 AI stuido 上实现 paddlepaddle 版本，并测试 AlexNet 在 Fashion-MNIST 数据集上的性能表现。

2.2.2 VGG —— 神经网络块 (block)

可以看出，AlexNet 中的卷积层中间穿插了激活函数与汇聚处理，成组的网络层呈现出典型的重复出现现象。由此，启发了神经网络模型的设计从层延伸到一组神经网络层的设计，即，块 (block)。这一做法是封装集成的思想产物，比如，芯片设计中从的晶体管、逻辑元件、逻辑块的逐级

延伸。

2014 年，有牛津大学著名视觉几何研究组 (visualgeometry group,VGG) 提出的 VGG 模型中使用了 VGG 块的概念，具体定义了 5 个卷积块 (由卷积层数量和输出通道数量的差异来定义)：其中前两个块各有一个卷积层，后三个块各包含两个卷积层。第一个模块有64个输出通道，每个后续模块将输出通道数量翻倍，直到该数字达到512。类似于 AlexNet 紧随 5 个卷积块追加了 3 个全连接层，因此，总共包含8个卷积层和3个全连接层，也称为 VGG-11。

基于卷积块的灵活设计，可以尝试不同的方案，VGG 实验表明：深层且窄的卷积 (即 3×3) 比较浅层且宽的卷积更有效。

作业：实验代码测试

可参照 d2i.ai ，在 AI stuido 上实现 paddlepaddle 版本，并测试 VGG 在 Fashion-MNIST 数据集上的性能表现。

2.2.3 NiN —— 干掉全连接层

到目前为止逐级介绍的代表性神经网络呈现出典型的架构特点：底层使用多个卷积块，近任务层使用多个全连接层。其中，全连接层带来的最大问题是模型参数量巨大，AlexNet 中全连接层的参数数目相对整个模型的参数量占比高达 93%，因此，这种模型结构迫切需要破局。首先，分析全连接层的作用是什么：

- 汇总多个空间位置及通道上的信息。
  - LeNet 第一个全连接层中的每个神经元汇总了 16 个通道上空间尺寸为 5*5 的信息，并紧随两个全连接层逐级提高汇总的程度(120-84-10)，最终匹配 10 个类后验概率。
  - AlexNet 第一个全连接层中的每个神经元汇总了 256 个通道上的空间尺寸为 5*5 的信息，并紧随两个全连接层逐级提高汇总程度(4096-4096-1000)，最终匹配 1000 个类后验概率。VGG 也采用了相同的汇总方法。

如何破局呢？

- 考虑去除空间维度上的信息融合，因为带步长的卷积操作以及汇聚操作已经进行了空间信息融合，同时可以在最后施加一个显示的汇聚操作来达到终极目标任务层面上的融合，比如汇聚出类后验概率。

- 考虑在早期融入通道层面上的信息汇总，比如，在卷积层之后进行通道层面的信息汇总，这种做法等同于一个 1*1 的卷积层。

以上就是 NiN 模型的核心想法，同时沿用了 VGG 的块结构思想，典型的 NiN 块结构如图 2.5 所示。

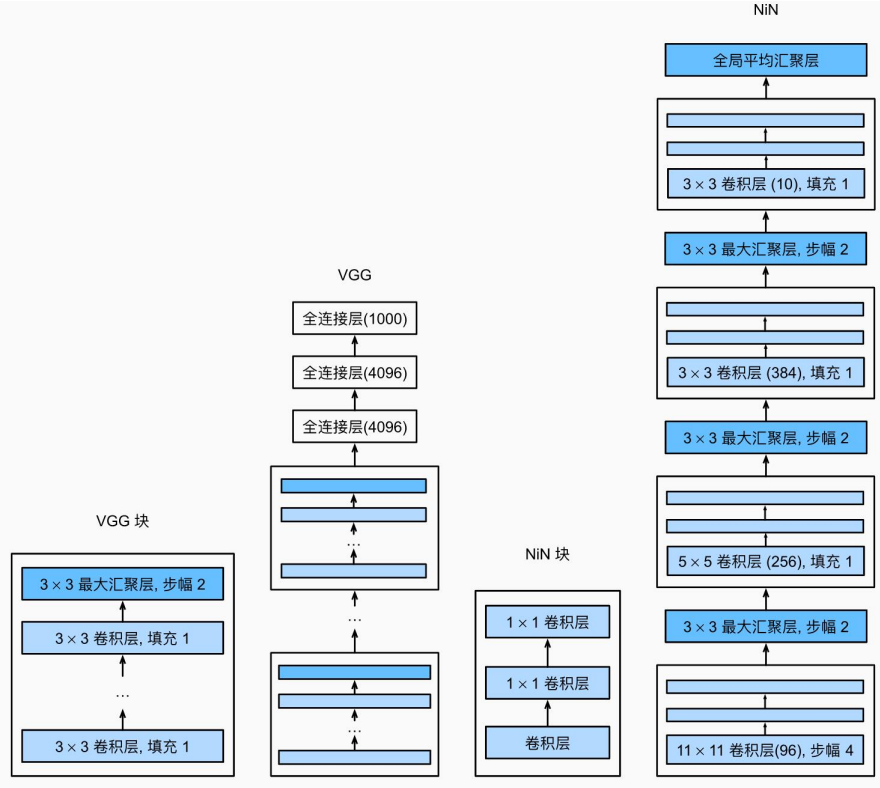

图 2.5: NiN 块结构与 VGG 块结构对比示意图

作业：实验代码测试

可参照 d2i.ai ，在 AI stuido 上实现 paddlepaddle 版本，并测试 NiN 在 Fashion-MNIST 数据集上的性能表现。

2.2.4 GoogleNet —— 卷积核尺寸对性能的影响

不同尺寸的卷积核侧重于学习不同空间粒度的模式，因此，最简单的应对多粒度模式学习的方式就是融合多尺度卷积核信息，为了能够进行简单的级联融合，追求不同尺寸卷积核最后的输出空间尺寸一致，这样就可以在通道层面上进行融合。

2014年的ImageNet竞赛上，发布的 GoogleNet 中正式提出了多尺度卷积核融合的块，称为 Inception 块 (可能关联自《盗梦空间》(inception)中筑梦时所使用的一句话“We need to go deeper”)。一个典型的 Inception 块的结构如图 2.6 所示：

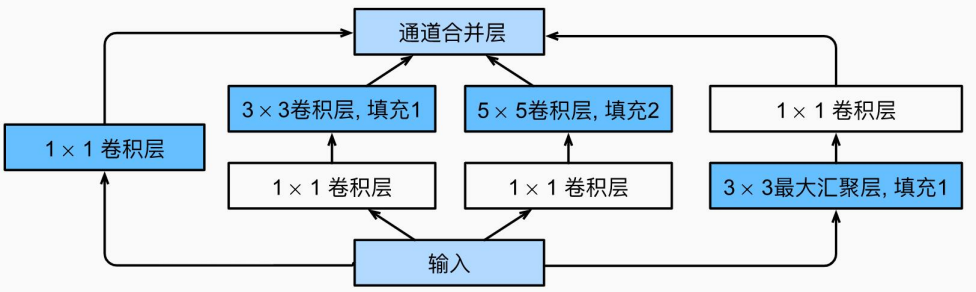

图 2.6: GoogleNet 中使用的 Inception 块的结构示意图。

简单介绍一下 inception block：由 4 条并行路径构成，涵盖了卷积层的尺寸为：1*1，3*3，5*5，其中第一条路径专注于通道信息融合的1*1卷积核，路径2和3专注于通道融合之后再分别进行3*3和5*5的卷积核，第 4 条路径则探索先进行3*3尺寸上的空间汇聚再进行通道融合 (1*1卷积核)。

作业：实验代码测试

可参照 d2i.ai，在 AI stuido 上实现 paddlepaddle 版本，并测试 GoogleNet 在 Fashion-MNIST 数据集上的性能表现。

2.2.5 批归一化 (batch normalization)

致力于解决深度神经网络模型训练时收敛困难的问题，尝试从数据预处理的角度来设计方案。考虑到不同特征的量级上可能存在差异，因此，

常用的数据预处理方式为均值中心化以及标准差单位化。在深度神经网络模型的训练时也考虑这种数据特征取值的归一化处理方案，结合深度神经网络训练时通常采用一批一批数据的处理方式，把这种归一化手段用到批数据上，因此称为批归一化。具体做法为：

$$\mathrm{BN}(\mathbf{x}) = \boldsymbol{\gamma} \odot \frac{\mathbf{x} - \hat{\boldsymbol{\mu}}_{\mathcal{B}}}{\hat{\boldsymbol{\sigma}}_{\mathcal{B}}} + \boldsymbol{\beta}$$

其中，均值采用批数据上的样本均值，标准差采用批数据上的样本标准差，归一化之后又进行了尺度变换和偏移操作，目的是让因训练所需而加入的BN能够有可能还原最初的输入（即当$\gamma = \sigma_B$和$\beta = \mu_B$ 时），从而保证了整个网络的容量 (capacity)。

批归一化作为深度神经网络模型中的一层，通常放置在线性运算之后，激活函数之前。对于全连接层而言，是一个很自然的过程，全连接层的每个输出可以看做是一个单独的变量，在批数据上统计每个变量的样本均值和样本标准差，直接归一化就行了。

对于卷积层而言，如何定义变量，难道还是把卷积层的所有输出都看做单独的变量吗，比如，与卷积核对应的输出特征(称为“通道”)的空间尺寸为 $p \times q$，难道看做是 pq 个变量吗？答案是否定的。这就要彻底理解卷积核的作用了，每一个卷积核可以看做是某种特定模式 (pattern) 的发现器，自然地，同一通道 (channel) 上的所有值都应该看做是当前pattern所对应变量的不同激活值，即，该变量的多个采样值，因此，应该在通道上进行归一化。假设一批数据共 m 个，则一个通道上对应 mpq 个输出，统一看做是一个变量的样本点，进行归一化处理，每个通道上独立的进行这样的归一化操作。

2.2.6 残差网络模型 (ResNet)

缘起：随着网络模型设计的逐层加深，模型学习出现退化现象 (degradation)：模型训练性能与测试性能都下降 (原因？？)。如何保障模型性能也逐层提升呢？

疑惑：假如浅层模型已经达到了一个性能巅峰，那么只要后续堆加的模块学习一个恒等映射，也会保障性能不降。

猜测：普通的深度神经网络模型不能够有效地学习到阶段性的恒等映射。

应对策略：从网络模型设计上为学习恒等映射提供前提条件，即，直接在堆叠的模块上增加一个直连通道，或者称为跳层连接，从而减缓新堆叠的模块的学习负担。

以上是理解残差网络的一种思路。几点发散的思考：

- 残差模块只是从模型的局部层面上提供了一种有利于新堆叠的模块学习出恒等映射的模型架构，事实上，这些模块的学习结果大概率地不是参数为 0 ，如何解释新堆叠的模块的功能？
- 优化过程是从模型全局的角度的展开的，局部残差设计对于优化层面上的影响是什么？
- 普通的深度神经网络模型为什么会出现退化现象？
  - 基于梯度下降的优化方法的难题：梯度弥散/爆炸问题，造成梯度难以有效地在各层之间传播。

嵌套类函数为模型性能提升提供了“保底扩增”的搜索空间基础，其示意图如下：

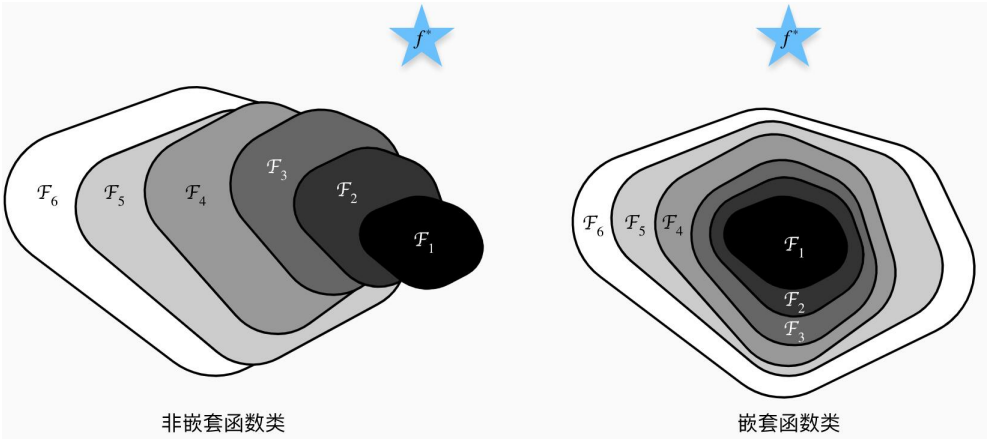

图 2.7: 嵌套类函数与非嵌套类函数的对比示意图。

如何构造嵌套类函数呢？设计一个恒等映射学习结构。残差模块正是遵循了这一思路，其结构如图 2.7 所示。嵌套类函数保证了模型假设空间的保底扩增，即，在一个较小的空间上的取得的最优函数 f^*，一定被包含在扩增后的模型假设空间中。

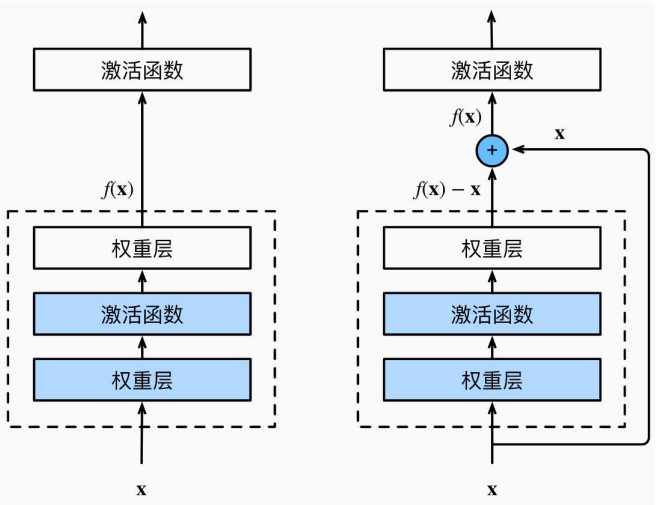

图 2.8: 残差模块与普通模块结构示意图。

恒等映射结构对梯度传播的影响：这种模型结构为残差模块层赋予了较低的学习期望，使得能够把梯度传递到更远的浅层网络中，同时从全局的角度适当的发力，获得保底扩增的效果。

梯度传递的视角

允许当前层输出端的梯度直接传递到当前层输入端。

作业：实验代码测试

可参照 d2i.ai，在 AI stuido 上实现 paddlepaddle 版本，并测试 ResNet 在 Fashion-MNIST 数据集上的性能表现。

2.2.7 稠密连接网络 (DenseNet)

相比于残差网络模块中把当前层的输出与跳层连接直接相加的方式来融合特征，还可以考虑在通道层面上进行特征拼接，即叠置跳层特征图与当前模块输出的特征图。为了后续网络的参数数量，通常先对拼接后的特征进行融合，降低特征的通道数，即添加一层过渡层 (transition layer)，比如 1*1 的卷积层。这种跳层连接与残差网络模块的对比示意图如图 2.9 所示：

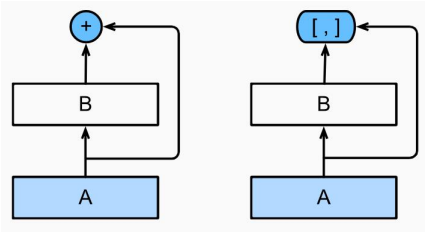

图 2.9: 特征拼接式的跳层连接与残差特征加性融合式的跳层连接的对比示意图。

另一方面拓展跳层连接到后续所有层，由此形成了一种“稠密”连接，即，当前层与前序所有层的输出特征都进行连接。稠密连接网络的结构示意图如图 2.10 所示。

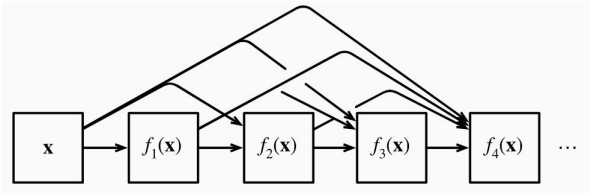

图 2.10: 稠密连接网络结构示意图。

梯度传递的视角

允许中间任意层直接从损失函数终端接受梯度回传。

作业：实验代码测试

可参照 d2i.ai，在 AI stuido 上实现 paddlepaddle 版本，并测试 DenseNet 在 Fashion-MNIST 数据集上的性能表现。

2.2.8 U-Net 中的跳层连接

在典型的编码器与解码器模型中，编码器与解码器通常是对称设计的。编码器负责逐层进行编码抽象级的学习，解码器则是依据编码器的最终编码逐级解码信息。考虑到编码器与解码器结构上的对称性，但是信息传递上却并没有进行对称传递，即，解码总是依据编码器的最终编码来逐级解

码，因此，可以考虑在进行当前级的信息解码时不仅依赖于上一层解码信息，还依赖于编码器中对称层的编码信息。由此，便形成了 U-Net 模型，在医学图像分割上取得显著效果。其结构如图 2.11 所示。具体实施时，是在模块层面上进行跳层连接的。

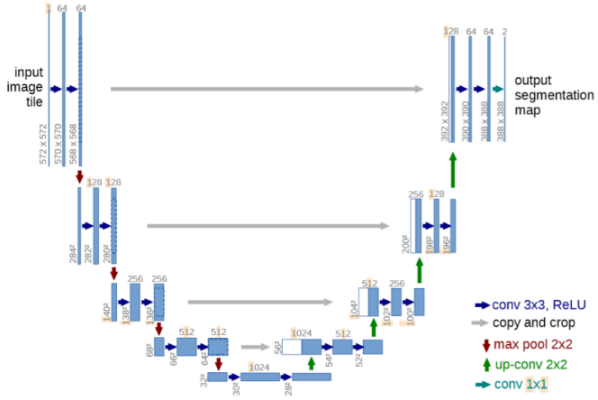

图 2.11: U-Net网络结构示意图。

2.3 计算机视觉领域中的深度网络模型解决方案

2.3.1 DenseBox

采用全卷积网络，输出空间尺寸为输入空间尺寸的1/4，输出层的通道数为 5，分别代表输出特征图上当前空间位置 (x_i, y_i) 的视觉信息所支撑的感兴趣目标的信息，包括：类别置信度，以及当前位置 (x_i, y_i) 与真实框 $[x_t, y_t, x_b, y_b]$ 的相对坐标(此处真实框的坐标也是在输出特征图上)。

感受野 在用来解决对象检测问题的神经网络模型中，随着网络层数的堆叠，通常伴随着空间下采样，后一层特征图空间中的一个点对应原图像空间中多大的区域，称为当前特征度的感受野。

检测模型通常最后一层设定为 5 个channel，即，类别置信度、当前位置与真实框的相对位置。可以看做是基于当前特征图中的一个点对应的感受野信息来预测感受野区域内/外的信息。