

LeNet, AlexNet, VGG和NiN

机器学习

VGG

概要

- LeNet （第一个卷积神经网络）
- AlexNet
 - 升级版的 LeNet
 - ReLu 激活， 丢弃法， 平移不变性
- VGG
 - 升华版的 AlexNet
 - 重复的 VGG 块
- NiN
 - 1x1 卷积 + 全局池化

深度学习的早期演进

➤ 时代背景：深度学习的黎明

➤ 从“特征工程”到“特征学习”的范式转移。算法、数据、算力三位一体的革命

➤ LeNet-5 (1998): 开山之作 (The Genesis)

➤ 如何实现端到端的图像模式识别？核心思想：局部感受野、权值共享、下采样

➤ AlexNet (2012): 王者归来 (The Breakthrough)

➤ 如何在“大数据、大模型”时代有效训练更深的CNN？规模压倒一切，辅以ReLU、Dropout和GPU

➤ VGG (2014): 深度与优雅 (The Architect)

➤ 如何系统性构建更深的网络？深度的收益是什么？用小的、重复的构建块（Block）来探索网络深度

➤ NiN (2014): 网络中的网络 (The Innovator)

➤ 全连接层的参数和计算瓶颈，以及卷积层表达力不足。用微型网络（1x1卷积）增强局部建模能力，用全局平均池化取代全连接层。

➤ 演进脉络与思想总结

➤ 一条从“手工调参”到“结构化设计”的道路

机器学习范式的变革

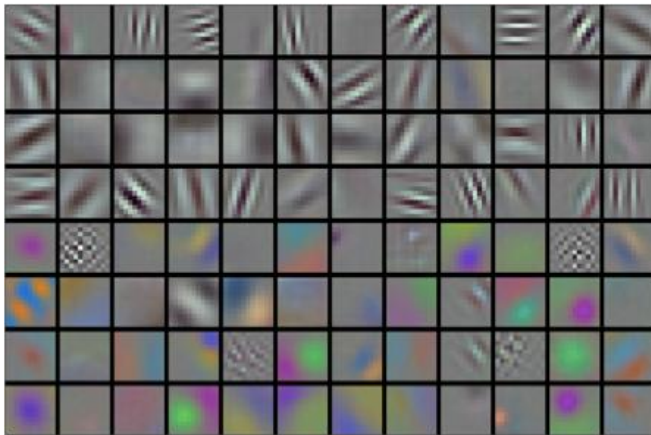
深度学习的黎明

- 在21世纪初，计算机视觉的主流范式是什么？其根本瓶颈在哪里？
- 旧范式：经典机器学习流水线 (- 2012)
 - 特征工程：依赖专家知识，手工设计特征提取器（如SIFT）。耗时泛化能力弱
 - 特征编码：将提取的特征编码为向量（如视觉词袋 BoVW）
 - 分类器训练：将特征向量送入传统分类器（如SVM, 线性模型）
- 模型的性能上限被手工设计的特征所“锚定”。特征的好坏，直接决定了任务的成败
- 新范式：深度学习 (2012 -)
 - 特征学习。让网络从数据中端到端地自动学习从低级（边缘）到高级（部件）的层次化特征
 - 特征学习成功的三个支柱
 - 算法 (Algorithm): 卷积神经网络（以LeNet为原型）的复兴与改进
 - 数据 (Data): 大规模、高质量的标注数据集（如 ImageNet）的出现，为训练复杂模型提供了燃料
 - 算力 (Compute): GPU 的并行计算能力，将训练大型网络的时间加速，使研究迭代成为可能

机器学习经典范式

特征本身应该被学习

- 在合理地复杂性前提下，特征应该由多个共同学习的神经网络层组成，每个层都有可学习的参数
- 在机器视觉中，最底层【开始层】可能检测到边缘、颜色和纹理



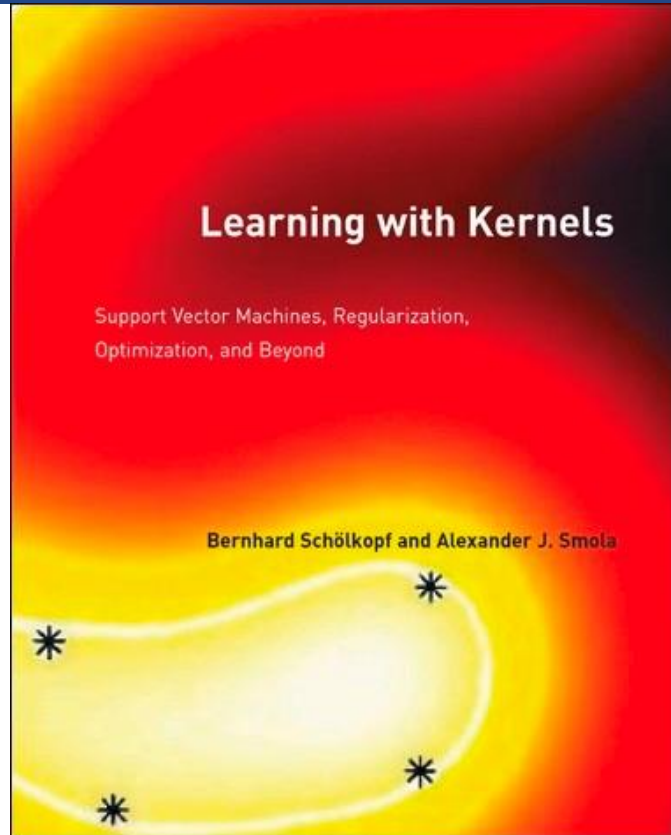
经典机器学习的流水线

- 获取一个有趣的数据集
- 根据光学、几何学、其他知识以及偶然的发现，手工对特征数据集进行预处理
- 通过标准的特征提取算法，如SIFT（尺度不变特征变换）和SURF（加速鲁棒特征）或其他手动调整的流水线来输入数据
- 将提取的特征送入最喜欢的分类器中（例如线性模型或其它核方法），以训练分类器

机器学习

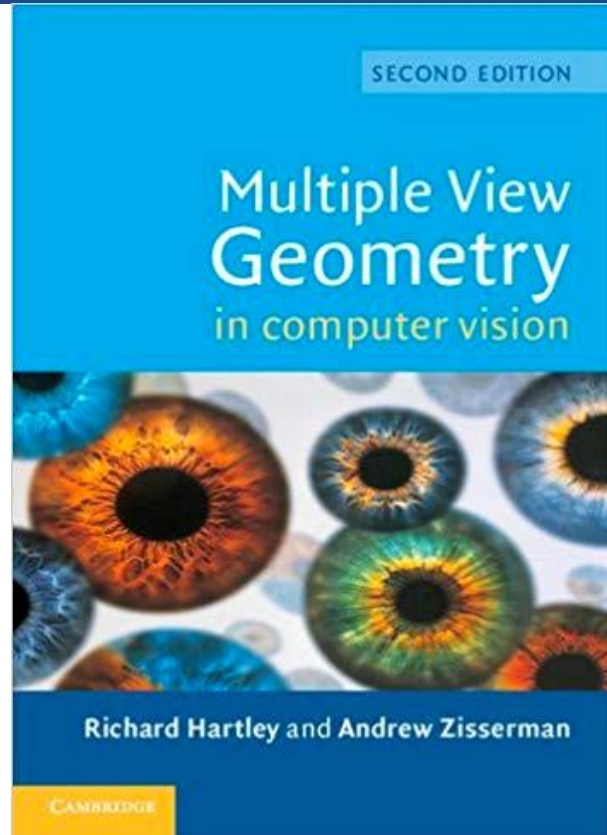
- 提取特征
- 选择内核以获得相似性
- 凸优化问题
- 许多完美的定理

In the 1990s, a new type of learning algorithm was developed, based on results from statistical learning theory: the Support Vector Machine (SVM). This gave rise to a new class of theoretically elegant learning machines that use a central concept of SVMs – kernels – for a number of



几何学

- 提取特征
- 不同角度描述几何（例如多个相机）
- （非）凸优化问题
- 许多完美的定理.....
- 当假设得到满足时，在理论上非常有效



特征工程

- 特征工程很重要
- 特征描述，例如 SIFT（尺度不变特征变换），SURF（更高效的完成特征的提取和描述）
- 视觉词袋（聚类）
- 应用SVM



变革：数据、算法、算力

ImageNet 数据集 (2010)



图像	自然物体的彩色图像	手写数字的灰色图像
尺寸	469 x 387	28 x 28
# 样本数	1200万	6万
# 类别数	1,000	10

硬件

	1970	1980	1990	2000	2010	2020
	10x	10x	100x	100x		
Data (samples)	10^2 (e.g. iris)	深度网络	10^4 OCR	深度网络	10^{10} advertising	10^{12} social nets
RAM	1kB	100kB	10MB	100MB	1GB	100GB
	10x	10x	100x	100x	10,000x	
CPU	100kF (8080)	1MF (80186)	10MF (80486)	1GF (Intel Core)	100GF NVIDIA	>1PF (8xP3 Volta)

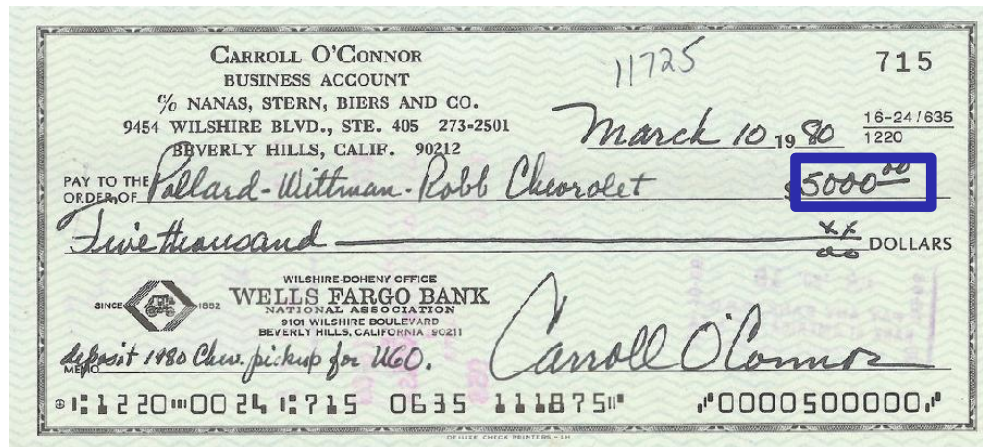
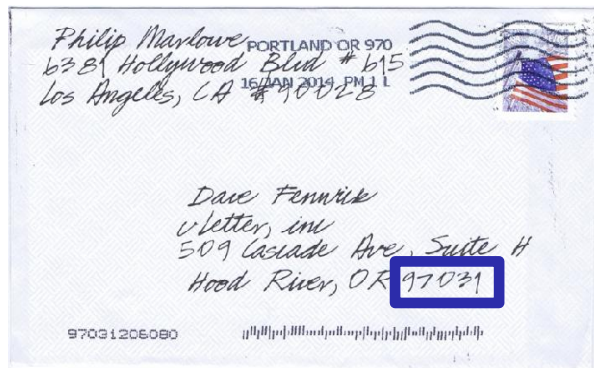
缺少的成分：硬件

- 相比于CPU，GPU由多个小处理单元组成，通常被分成更大的组
 - 虽然每个GPU核心相对较弱，但庞大的核心数量使GPU比CPU快几个数量级
 - CPU的浮点性能到目前为止还没有超过1 TFlops
 - 首先，功耗往往会随时钟频率呈二次方增长
 - 其次，GPU内核要简单得多，这使得它们更节能
- 深度学习中的许多操作需要相对较高的内存带宽，而GPU拥有10倍于CPU的带宽
- 2012年的重大突破，当Alex Krizhevsky和Ilya Sutskever实现了可以在GPU硬件上运行的深度卷积神经网络时，一个重大突破出现了。他们意识到卷积神经网络中的计算瓶颈：卷积和矩阵乘法，都是可以在硬件上并行化的操作

深度学习

LeNet-5

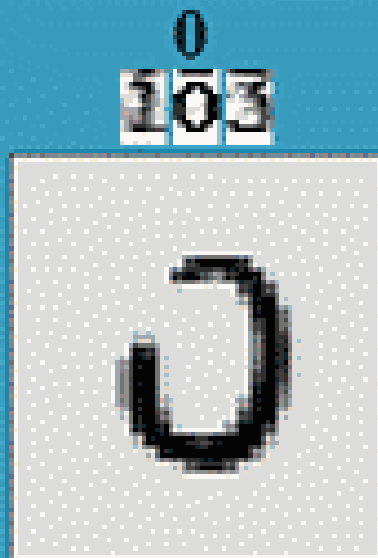
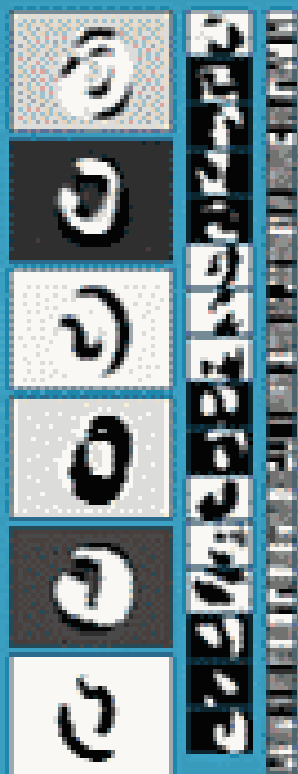
LeNet-5 (1998) - 开山之作（手写的数字识别）



MNIST

- 居中和缩放
- 50,000 个训练数据
- 10,000 个测试数据
- 图像大小28*28
- 10 类





Y. LeCun, L. Bottou, Y.
Bengio, P. Haffner, 1998
Gradient-based learning
applied to document
recognition

LeNet-5

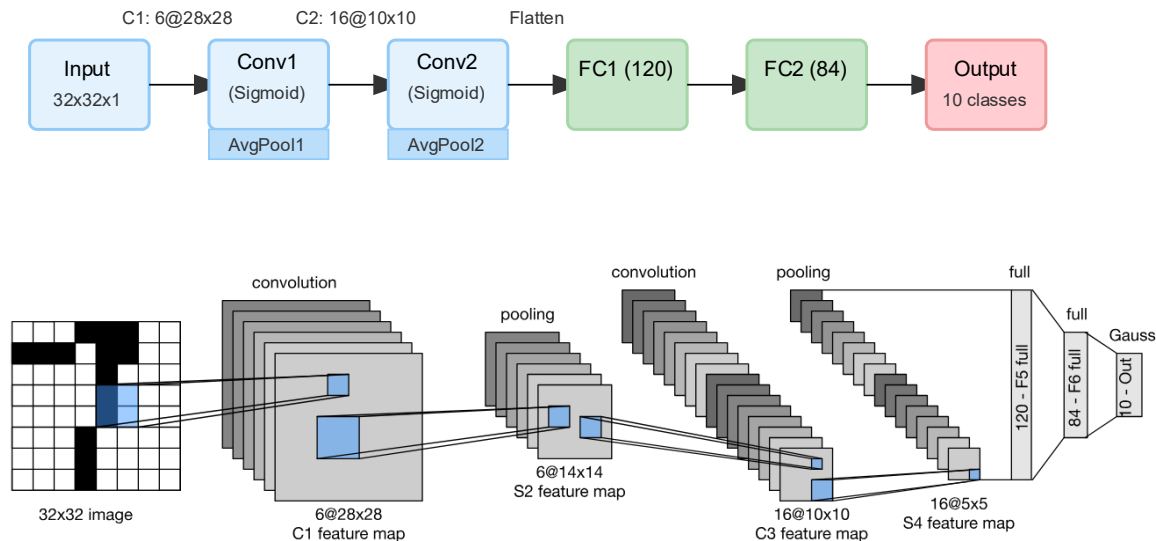
- 如何构建一个能自动从原始像素学习，并识别手写数字的端到端系统？
 - LeNet的设计模拟了生物的视觉皮层。神经元只对视野中的一小块区域（局部感受野）敏感，并通过层级结构，将简单的视觉基元组合成更复杂的模式
- 核心思想与工作机制：
 - 卷积层 (Convolutional Layer):
 - 局部感受野 (Local Receptive Fields): 每个神经元只连接输入一小块区域，用于捕捉局部特征
 - 权值共享 (Weight Sharing): 一个卷积核（滤波器）在整个图像上滑动，用同一组参数检测同一种特征，极大地减少了模型参数量，并带来了 平移不变性
 - 池化层 (Pooling/Subsampling Layer):
 - 作用：降低特征图的空间分辨率，减少计算量，同时增强模型的平移、缩放、旋转不变性
 - LeNet使用：平均池化 (Average Pooling)，对邻域内的特征取平均值
 - 全连接层 (Fully Connected Layer):
 - 在提取了层次化特征后，将这些特征“展平”并送入传统的多层感知机（MLP）进行分类

LeNet 架构

➤ 关键假设与局限

- 激活函数：使用Sigmoid/Tanh，在网络较深时易导致梯度消失，训练困难
- 池化方式：平均池化会模糊特征，可能丢失重要信息
- 规模：专为小尺寸、灰度图像设计，难以直接扩展到ImageNet这样复杂、高分辨率的彩色图像数据集

LeNet-5 架构流程



新一代神经网络

AlexNet

AlexNet (2012)

- 在ImageNet大规模图像分类挑战赛上，证明深度卷积网络优于当时所有的经典方法。
目标是精度
 - AlexNet在2012年赢得ImageNet 竞赛
- 推出更深更大的LeNet，直接导致计算机视觉的范式转变
 - AlexNet是CNN发展史上的一个分水岭。它不是理论上的全新发明，而是通过卓越的工程实践和对现有技术的巧妙整合，雄辩地证明了深度学习在复杂视觉任务上的压倒性优势，开启了深度学习的黄金时代

AlexNet (2012)

➤ ReLU激活函数 (Rectified Linear Unit)

- 为什么是突破？解决梯度消失，计算高效，稀疏性

➤ Dropout (丢弃法)

- 在训练过程中，以一定概率随机“丢弃”（即输出置为0）一部分神经元的输出
- 为什么有效？它是一种强大的正则化技术。强迫网络学习冗余的表示

➤ 重叠最大池化 (Overlapping Max Pooling)

- 池化窗口的步长（stride）小于窗口大小（kernel size），使得相邻窗口之间有重叠
- 为什么有效？减少了信息丢失，提高了特征的丰富度，并具有更好的泛化性能

➤ 数据增强 (Data Augmentation)

- 对训练图像进行随机裁剪、水平翻转、颜色抖动等变换
- 为什么有效？人为地扩充了数据集，让模型看到更多样样本，提升模型的鲁棒性和泛化能力

➤ GPU并行训练

- 利用两块GPU并行训练。工程上的创举，为后续更大模型的发展铺平了道路

AlexNet

➤ AlexNet由八层组成

➤五个卷积层，两个全连接隐藏层和一个全连接输出层

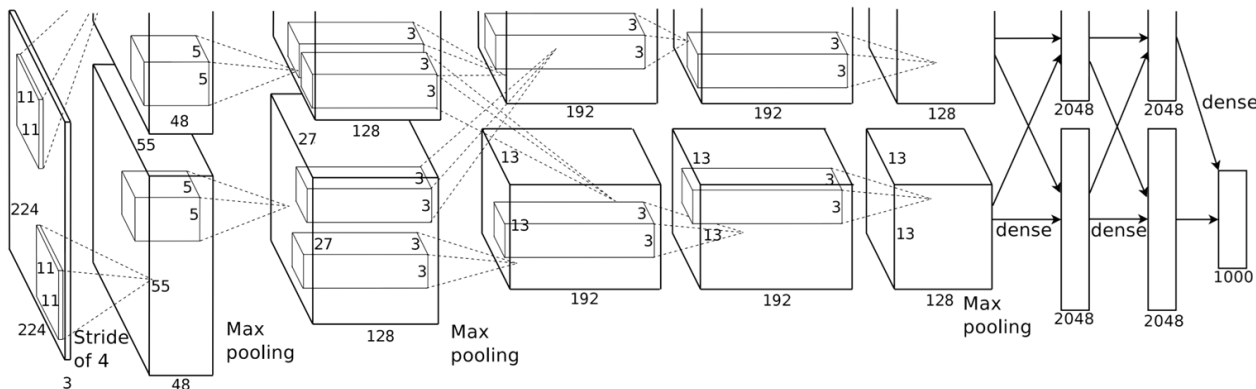
➤卷积通道数目是LeNet的10倍

➤最后两全连接层将近1GB的模型参数

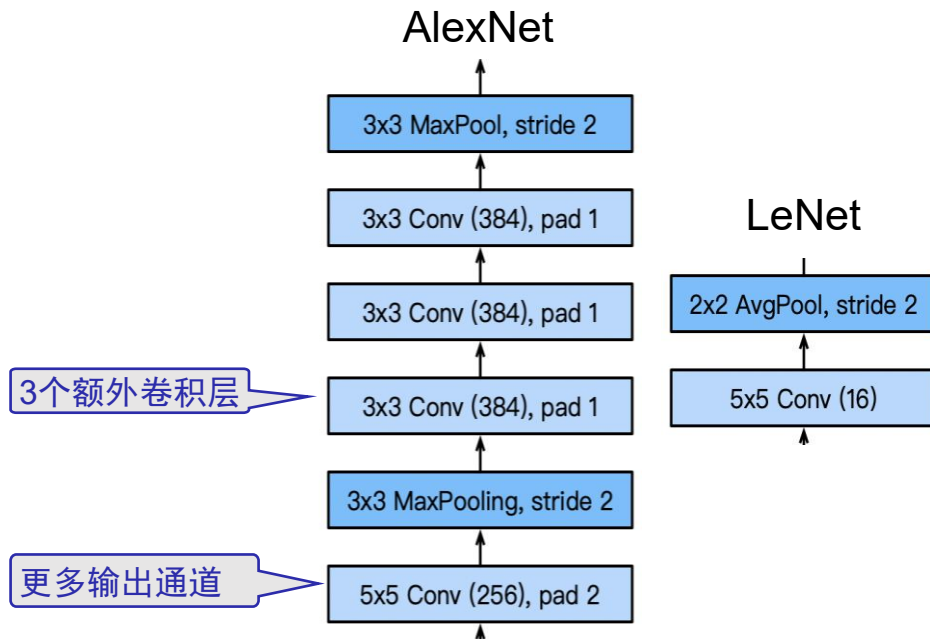
➤早期GPU显存有限，原版的AlexNet采用了双数据流设计

➤AlexNet使用ReLU

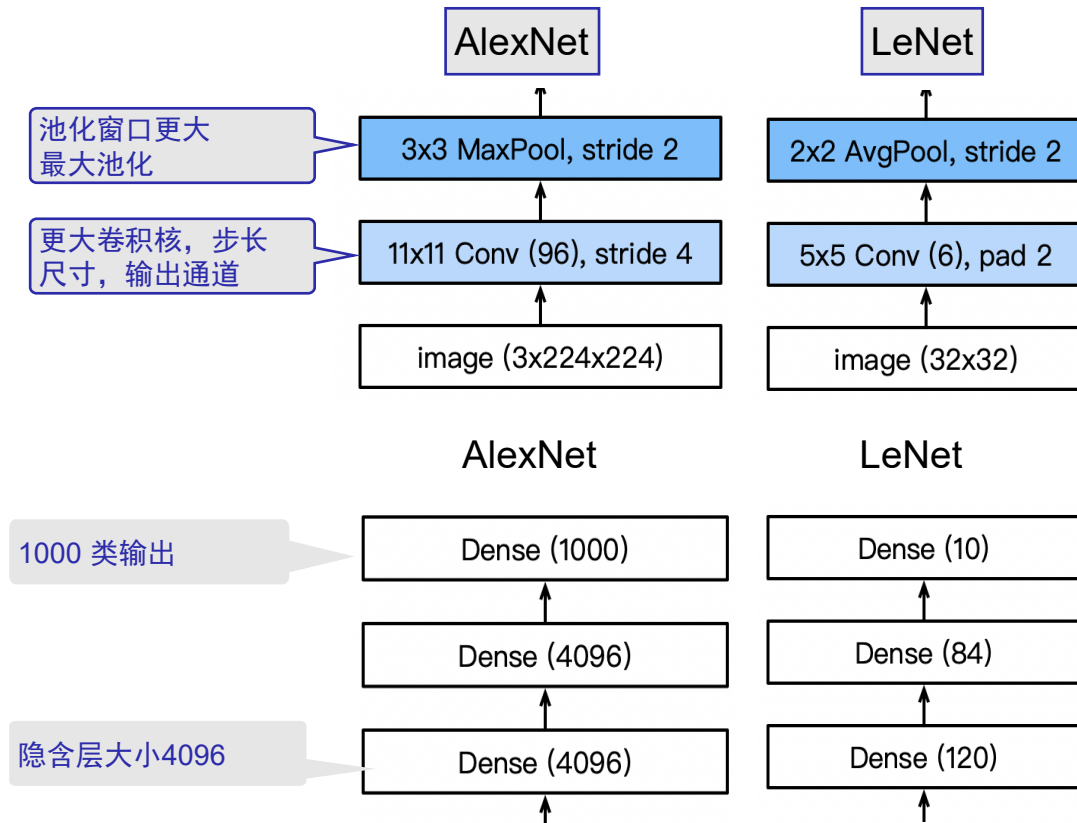
➤AlexNet在训练时增加了大量的图像增强数据，如翻转、裁切和变色



AlexNet VS LeNet

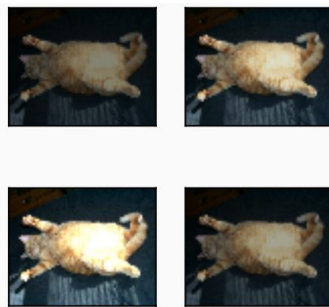
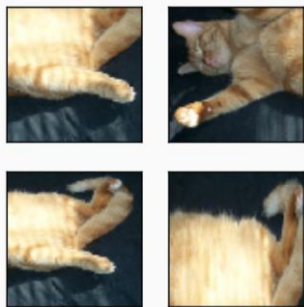
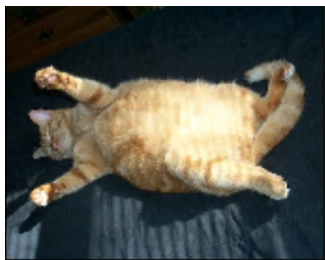


AlexNet 架构



更多技巧

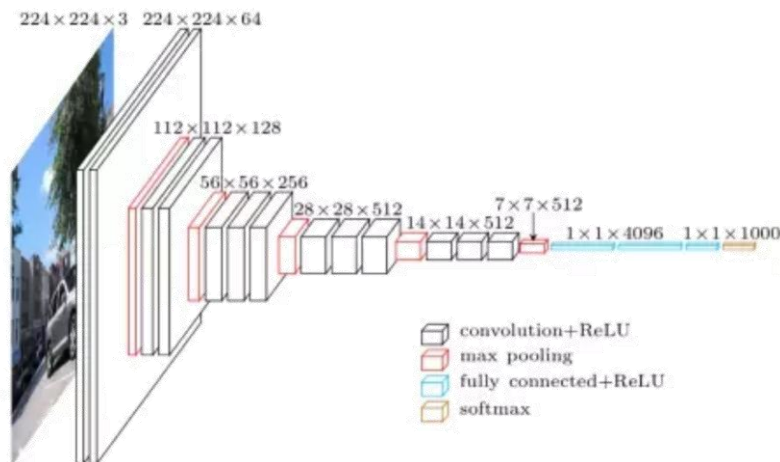
- 将激活函数从 sigmoid 更改为 ReLu（不再梯度消失）
- 在两个隐含层之后应用丢弃法（更好的稳定性 / 正则化）
- 数据增强



VGG

VGG (2014) - 深度与优雅

- 我们能否找到一种 系统性、模块化 的方法来构建更深的网络，并量化深度的影响？
- 神经网络架构的设计逐渐变得更加抽象
 - 从单个神经元的角度思考问题，发展到整个层，又转向块，重复层的模式
 - 由一系列卷积层组成，后面再加上用于空间下采样的最大汇聚层
 - 首先出现在VGG中。通过使用循环和子程序，可以很容易在深度学习框架的代码中实现这些重复的架构



➤ VGG块 (VGG Block):

- 构成： 由一连串连续的3x3卷积层 组成，最后跟一个 2x2最大池化层 用于下采样
- 设计哲学： 整个网络由若干个这样的VGG块堆叠而成，结构非常规整

➤ 用小卷积核堆叠代替大卷积核

- 两个连续3x3卷积层，感受野等效一个5x5卷积层；三个连续3x3卷积层，等效一个7x7卷积层
- 为什么更好？
 - 更多非线性： 堆叠小卷积核意味可以插入更多ReLU激活函数，增加网络的非线性表达能力
 - 更少参数： 通道数 C ，5x5卷积层参数量 $25C^2$ 。两3x3卷积层 $18C^2$ 。参数更少，模型更紧凑

➤ 在算法谱系中的位置

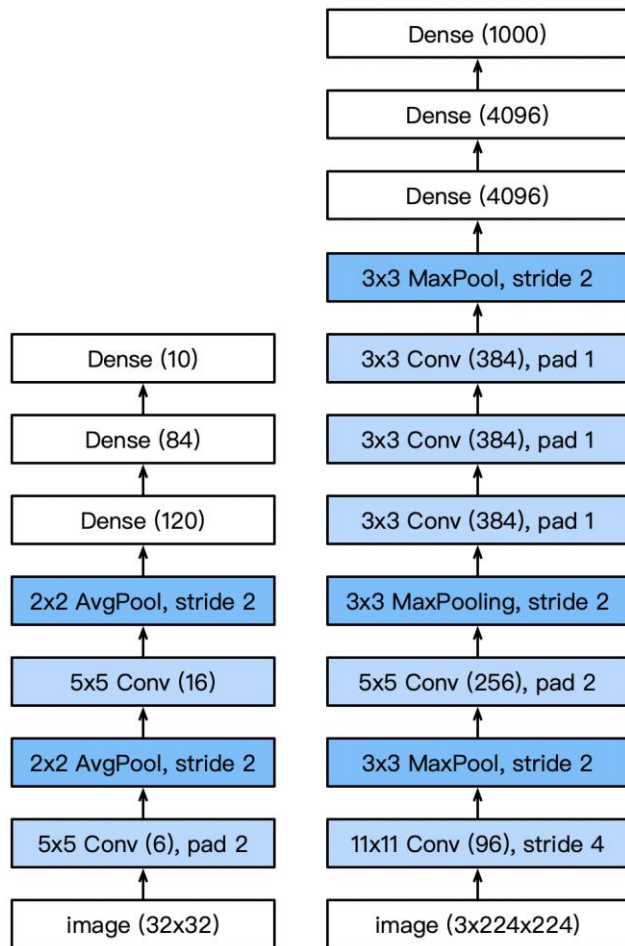
- VGG是AlexNet思想的继承和升华。它证明了只要简单地增加网络深度，就能显著提升性能，为后来的研究（如ResNet）指明了方向
- VGG参数量巨大（尤其全连接层），计算成本高昂，被称为 “最后一层的诅咒”

➤ AlexNet 比 LeNet 更深入更大，以获得更强性能

➤ 怎么更大更深？

➤ 选项

- 更多稠密层(开销太大)
- 更多的卷积层
- 分组成块



VGG块

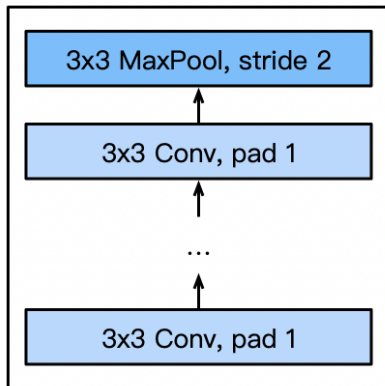
➤ 更深还是更宽？

- 5x5 卷积
- 3x3 卷积（更多）
- 更深和更窄更好

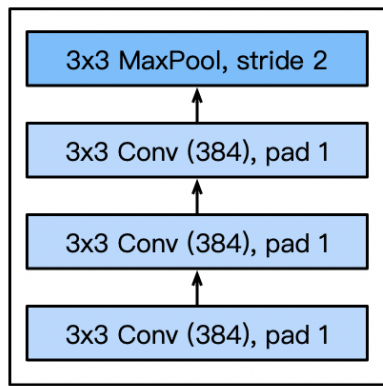
➤ VGG块

- 3x3卷积（填充=1）
 - （n层，m个通道）
- 3x3最大池化层
 - （步幅=2）

VGG 块

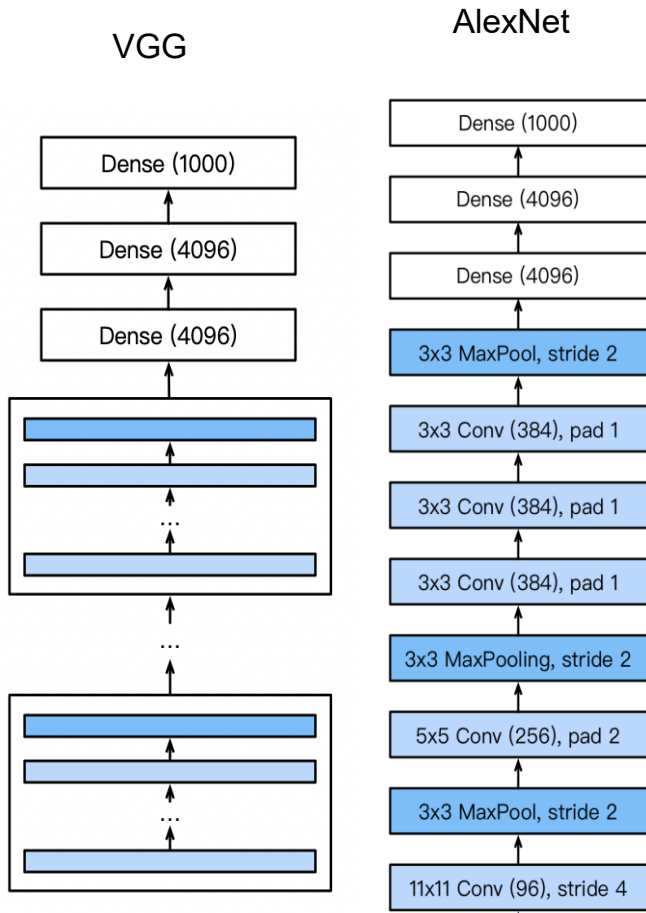


AlexNet 一部分



VGG 结构

- 多个VGG块后加稠密层
- 不同数目的重复VGG 块，可获得不同的架构
 - 例如VGG-16， VGG-19



NiN

NiN (2014) - 深入理解

➤在VGG将“堆叠卷积层”的范式推向极致后，两个根本性问题变得日益突出：

➤卷积层的表达能力有限

- 标准的卷积核本质上是广义线性模型，对局部感受野进行线性加权求和，然后通过一个固定的非线性激活函数
- 我们期望网络能在局部区域内提取高度抽象的特征，简单的线性滤波器可能不足以胜任

➤全连接层的诅咒

- VGG-16中超过85%的参数（约1.1亿）集中最后三个全连接层，训练缓慢，易过拟合
- 全连接层之前的特征空间结构被Flatten操作“压平”成一维长向量
- 全连接层的权重与特定的空间位置绑定，使得模型对输入的空间平移和形变较为敏感

➤NiN的设计目标

- 增强局部建模能力：用更强大的非线性函数逼近器取代传统的线性卷积核
- 彻底取代全连接层：设计一种新的、更符合卷积网络精神的分类输出层

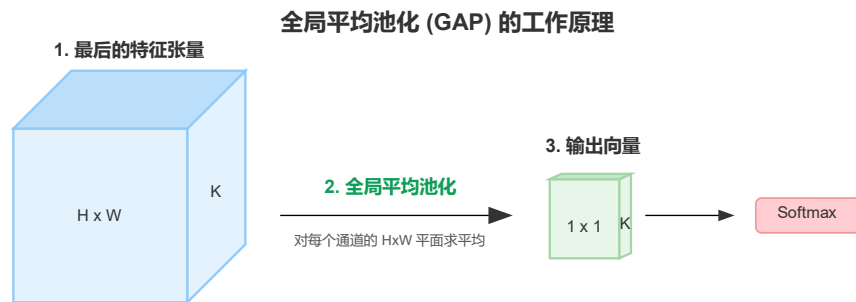
核心思想：mlpconv层 (微型网络) 增强局部抽象

- 在每个局部感受野上放一个迷你的“神经网络”！
- 用 1x1 卷积实现跨通道的MLP
 - mlpconv层（或NiN块），由一个标准卷积层和其后跟随的多个 1x1卷积层构成。1x1卷积：
 - 1x1的卷积核，其输入通道为 C_{in} ，输出通道为 C_{out}
 - 在每个像素点上，将 C_{in} 个通道的值进行线性加权组合，生成 C_{out} 个新的通道值
 - 完全等价于一个作用于通道维度的全连接层。这个全连接层在整个特征图的所有空间位置上是共享的
 - 一个 $\text{Conv}(k,k) \rightarrow \text{ReLU} \rightarrow \text{Conv}(1,1) \rightarrow \text{ReLU}$ 的序列，等价于在每个 $k \times k$ 的感受野上，先进行一次线性特征提取，然后将结果送入一个微型的多层感知机（MLP）进行复杂的非线性特征变换
- 网络中的网络：用一个微型MLP网络，增强主干卷积网络中每个局部计算的抽象能力



核心思想：全局平均池化取代全连接层

- 在网络最后卷积层（通常是一个mlpconv层）后，取代Flatten和全连接层：
 - 假设最后一层卷积输出了 K 个特征图（feature maps），每个图的尺寸为 $H \times W$
 - 对于第 k 个特征图，计算其所有 $H \times W$ 个像素值的平均值，得到一个标量
 - 对所有 K 个特征图执行此操作，最终得到一个 K 维的向量
 - 这个 K 维向量可以直接送入 Softmax 层进行分类（如果类别数就是 K ）



➤全局平均池化 (GAP) 为何是革命性的？

➤极致的参数效率：GAP本身没有任何需要学习的参数

➤彻底消除了全连接层带来的巨大参数负担，使得模型更轻量，训练更快，并从根本上抑制了过拟合

➤增强了可解释性与鲁棒性

➤强制的特征-类别对应：GAP在特征图和最终类别之间建立了直接的对应关系。为了让某个类别的得分高，网络必须学会在对应的特征图上产生强烈的、全局性的激活。这使得最后一个卷积层的特征图在概念上类似于类别置信度图

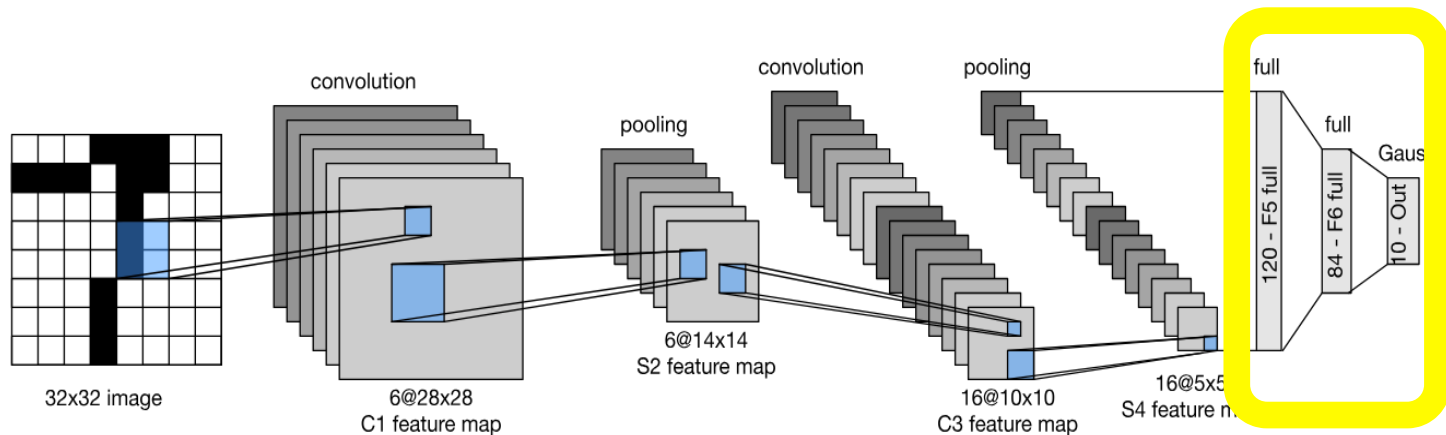
➤空间不变性：由于是对整个特征图求平均，模型对目标在图像中的小范围平移不那么敏感，增强了模型的空间鲁棒性

➤ NiN 的历史地位与深远影响

- NiN本身在ImageNet竞赛上的表现并未超越当时的顶尖模型，但它的思想却极具前瞻性，并被后续几乎所有成功的架构所吸收
 - 1x1 卷积成为现代CNN架构的标准组件。它被广泛用于
 - 降维/升维：在不改变空间分辨率的情况下，灵活地调整通道数，如GoogLeNet的Inception模块
 - 构建瓶颈结构 (Bottleneck)：在ResNet中，使用 $1 \times 1 \rightarrow 3 \times 3 \rightarrow 1 \times 1$ 的结构，先降维减少计算量，再升维恢复通道，极大提升了深层网络的效率
 - 跨通道信息融合：作为一种轻量级的特征融合方式
 - 全局平均池化 (GAP) 几乎完全取代了全连接层，成为现代CNN分类模型的标准结尾
 - 从GoogLeNet, ResNet到DenseNet, MobileNet, 再到Vision Transformer, 其思想无处不在
- NiN是一位思想家，而非仅仅一个高性能模型
 - 没有满足于堆叠更多的层，深刻反思了卷积网络的基本组件，并提出了两个影响深远的结构性创新，为CNN架构设计开辟了全新的道路

最后一层的诅咒

- 卷积层需要相对较少的参数 $c_i \times c_o \times k^2$
- 最后一层(稠密层)对于n个类的需要参数: $c \times m_w \times m_h \times n$
 - LeNet $16 \times 5 \times 5 \times 120 = 48k$
 - AlexNet $256 \times 5 \times 5 \times 4096 = 26M$
 - VGG $512 \times 7 \times 7 \times 4096 = 102M$



NiN 块

➤ 卷积层

➤ 超参数：卷积核大小，步幅和填充

➤ 接下来是两个 1x1 卷积层

➤ 步幅1

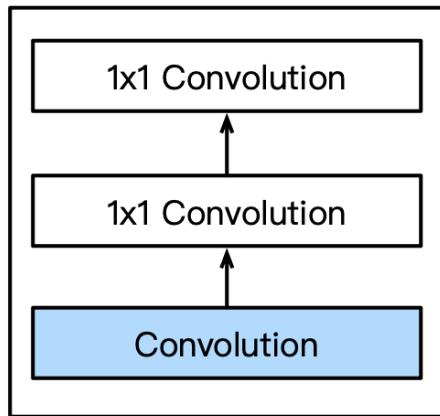
➤ 无填充

➤ 与第一层输出通道相同

➤ 充当稠密层

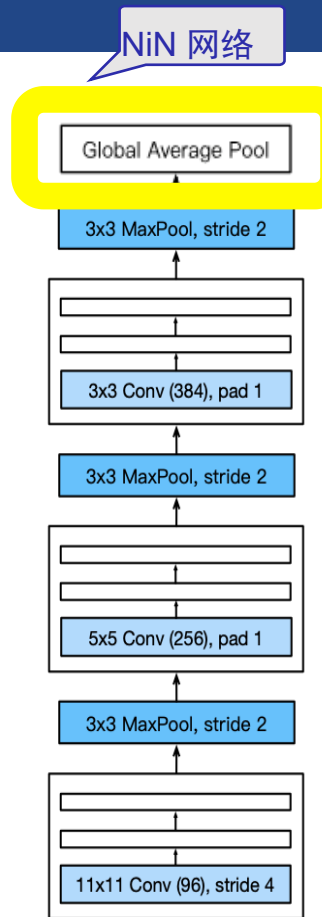
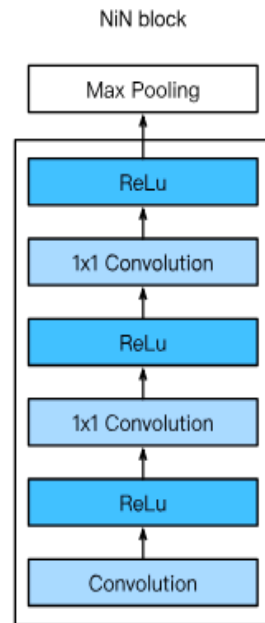
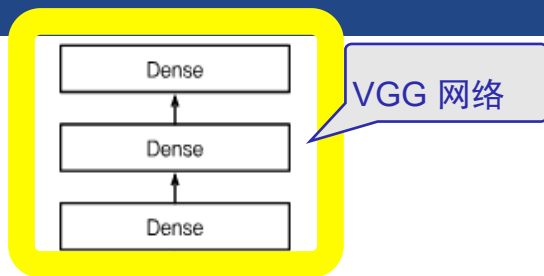
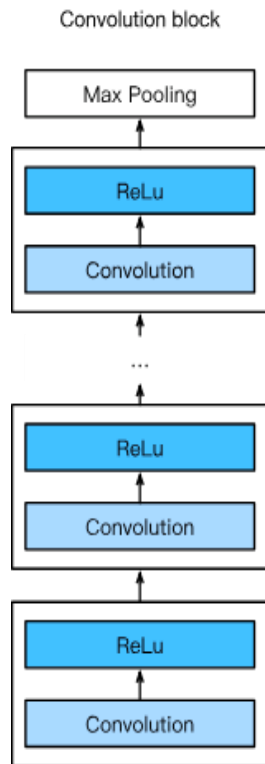
➤ NiN的想法

➤ 在每个像素位置（针对每个高度和宽度）应用一个全连接层。如果将权重连接到每个空间位置，可将其视为 1×1 卷积层



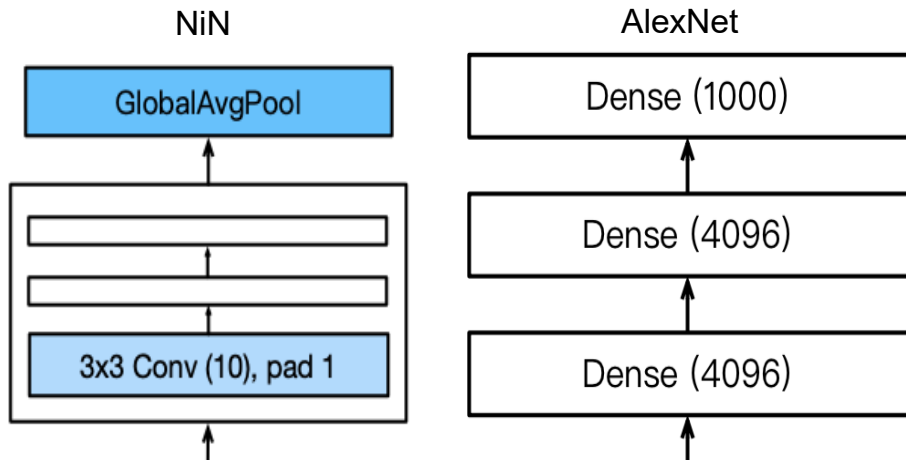
NiN网络

- 在每个像素的通道上
- 分别使用多层感知机



NiN最后一层

- 用NiN块替换了AlexNet的稠密层
- 输出：全局平均池化层



总结

总结

- LeNet 提出了CNN的基本蓝图。
- AlexNet 证明了这张蓝图在大数据和强算力下的巨大潜力
 - 升级版的 LeNet
 - ReLu 激活， 丢弃法， 平移不变性
- VGG 教会我们如何用一种优雅、系统的方式构建更深的网络
 - 升华版的 AlexNet
 - 重复的 VGG 块
- NiN 则对CNN的基本组件进行了深刻反思和创新，为更高效、更强大的现代网络架构（如GoogLeNet, ResNet）铺平了道路
 - 1x1卷积 + 使用全局池化层替代稠密层