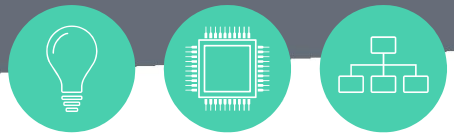
深度学习之 PyTorch 实战

卷积神经网络 Part2



主讲老师: 土豆老师

目录

现代卷积神经网络

- 我们已经介绍了卷积神经网络的基本原理，本节我们将带你了解现代的卷积神经网络结构，许多现代卷积神经网络的研究都是建立在这一章的基础上的。
- 在本节中的每一个模型都曾一度占据主导地位，其中许多模型都是 ImageNet 竞赛的优胜者。ImageNet 竞赛自2010年以来，一直是计算机视觉中监督学习进展的指向标。
  - AlexNet。第一个在大规模视觉竞赛中击败传统计算机视觉模型的大型神经网络；
  - 使用重复块的网络 (VGG)。它利用许多重复的神经网络块；
  - 网络中的网络 (NiN)。它重复使用由卷积层和 1×1 卷积层（用来代替全连接层）来构建深层网络；
  - 含并行连结的网络 (GoogLeNet)。它使用并行连结的网络，通过不同窗口大小的卷积层和最大池化层来并行抽取信息；
  - 残差网络 (ResNet)。它通过残差块构建跨层的数据通道，是计算机视觉中最流行的体系结构；
  - 稠密连接网络 (DenseNet)。它的计算成本很高，但给我们带来了更好的效果。

深度卷积神经网络(AlexNet)

- 在 LeNet 提出后，卷积神经网络在计算机视觉和机器学习领域中很有名气。但卷积神经网络并没有主导这些领域。这是因为虽然 LeNet 在小数据集上取得了很好的效果，但是在更大、更真实的数据集上训练卷积神经网络的性能和可行性还有待研究。
- 事实上，在上世纪90年代初到2012年之间的大部分时间里，神经网络往往被其他机器学习方法超越，如支持向量机 (support vector machines)。
- 在计算机视觉中，直接将神经网络与其他机器学习方法进行比较也许不公平。这是因为，卷积神经网络的输入是由原始像素值或是经过简单预处理（例如居中、缩放）的像素值组成的。但在使用传统机器学习方法时，从业者永远不会将原始像素作为输入。在传统机器学习方法中，计算机视觉流水线是由经过人的手工精心设计的特征流水线组成的。对于这些传统方法，大部分的进展都来自于对特征有了更聪明的想法，并且学习到的算法往往归于事后的解释。

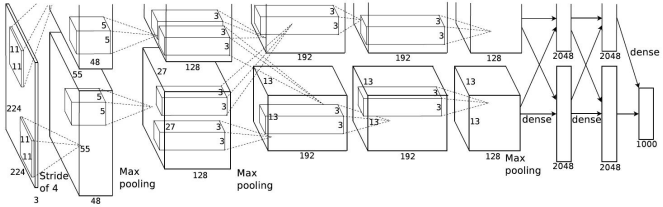
深度卷积神经网络(AlexNet)

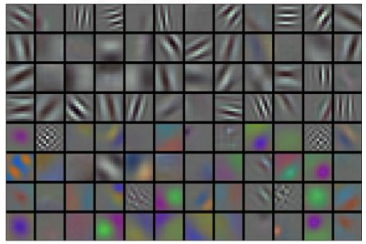
- 与训练端到端 (从像素到分类结果) 系统不同，经典机器学习的流水线看起来更像下面这样：
  1. 获取一个有趣的数据集。在早期，收集这些数据集需要昂贵的传感器 (在当时最先进的图像也就 100 万像素)。
  2. 根据光学、几何学、其他知识以及偶然的发现，手工对特征数据集进行预处理。
  3. 通过标准的特征提取算法，如SIFT (尺度不变特征变换) [Lowe, 2004] 、SURF (加速鲁棒特征) [Bay et al., 2006] 或其他手动调整的流水线来输入数据。
  4. 将提取的特征放到最喜欢的分类器中 (例如线性模型或其它核方法)，以训练分类器。
- 在机器学习计算机视觉研究里，推动领域进步的是数据特征，而不是学习算法。
- 计算机视觉研究人员相信，从对最终模型精度的影响来说，更大或更干净的数据集、或是稍微改进的特征提取，比任何学习算法带来的进步要大得多。

- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 91–110.
- Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: speeded up robust features. *European conference on computer vision* (pp. 404–417).

深度卷积神经网络(AlexNet)

- 包括 Yann LeCun、Geoff Hinton、Yoshua Bengio、Andrew Ng、Shun ichi Amari 和 Juergen Schmidhuber，想法则与众不同：他们认为特征本身应该被学习。此外，他们还认为，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，最底层可能检测边缘、颜色和纹理。
- 事实上，Alex Krizhevsky、Ilya Sutskever 和 Geoff Hinton 提出了一种新的卷积神经网络变体 AlexNet。在2012年ImageNet挑战赛中取得了轰动一时的成绩。AlexNet 以 Alex Krizhevsky 的名字命名，他是论文 [Krizhevsky et al., 2012] 的第一作者。
- 有趣的是，在网络的最底层，模型学习到了一些类似于传统滤波器的特征抽取器。

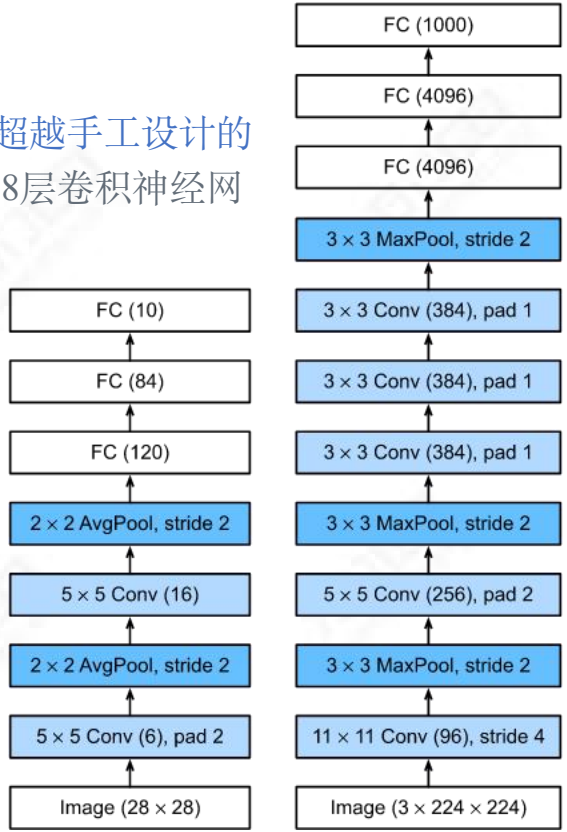




- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* (pp. 1097–1105).

深度卷积神经网络(AlexNet)

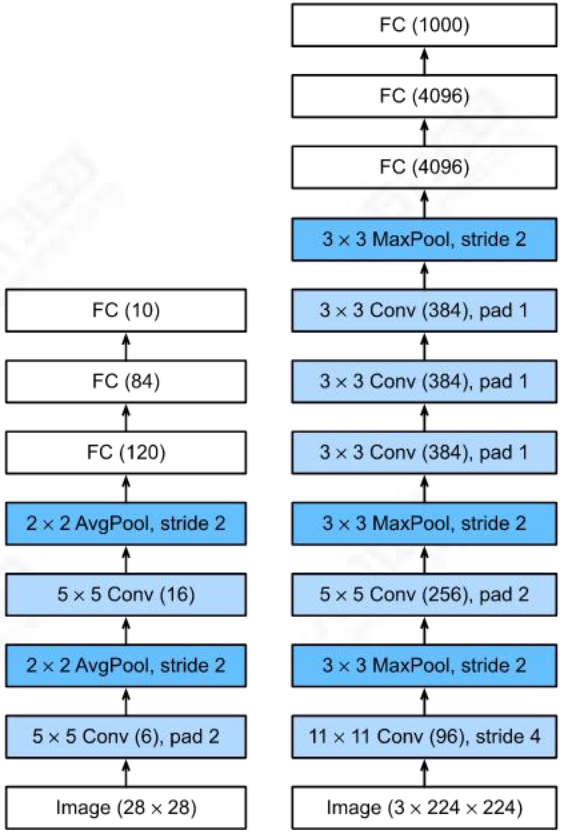
- 2012年，AlexNet 横空出世。它首次证明了学习到的特征可以超越手工设计的特征。它一举了打破计算机视觉研究的现状。 AlexNet 使用了8层卷积神经网络，并以很大的优势赢得了2012年 ImageNet 图像识别挑战赛。
- AlexNet 和 LeNet 的架构非常相似，如右图所示。 注意，这里我们提供了一个稍微精简版本的 AlexNet，去除了当年需要两个小型 GPU 同时运算的设计特点。
- AlexNet 和 LeNet 的设计理念非常相似，但也存在显著差异。首先，AlexNet 比相对较小的 LeNet5 要深得多。 AlexNet 由八层组成：五个卷积层、两个全连接隐藏层和一个全连接输出层。 其次，AlexNet 使用 ReLU 而不是 sigmoid 作为其激活函数。
- 下面，让我们深入研究 AlexNet 的细节。



LeNet (左)， AlexNet (右)

深度卷积神经网络(AlexNet)

- 在 AlexNet 的第一层，卷积窗口的形状是 11×11 。由于大多数 ImageNet 中图像的宽和高比 MNIST 图像的多10倍以上，因此，需要一个更大的卷积窗口来捕获目标。 第二层中的卷积窗形状被缩减为 5×5 ，然后是 3×3 。 此外，在第一层、第二层和第五层之后，加入窗口形状为 3×3 、步幅为 2 的最大池化层。 此外，AlexNet 的卷积通道是 LeNet 的10倍。
- 在最后一个卷积层后有两个全连接层，分别有4096个输出。这两个巨大的全连接层拥有将近 1GB 的模型参数。 由于早期 GPU 显存有限，原版的 AlexNet 采用了双数据流设计，使得每个 GPU 只负责存储和计算模型的一半参数。 幸运的是，现在GPU显存相对充裕，所以我们现在很少需要跨 GPU 分解模型 (因此，我们的AlexNet模型在这方面与原始论文稍有不同)。

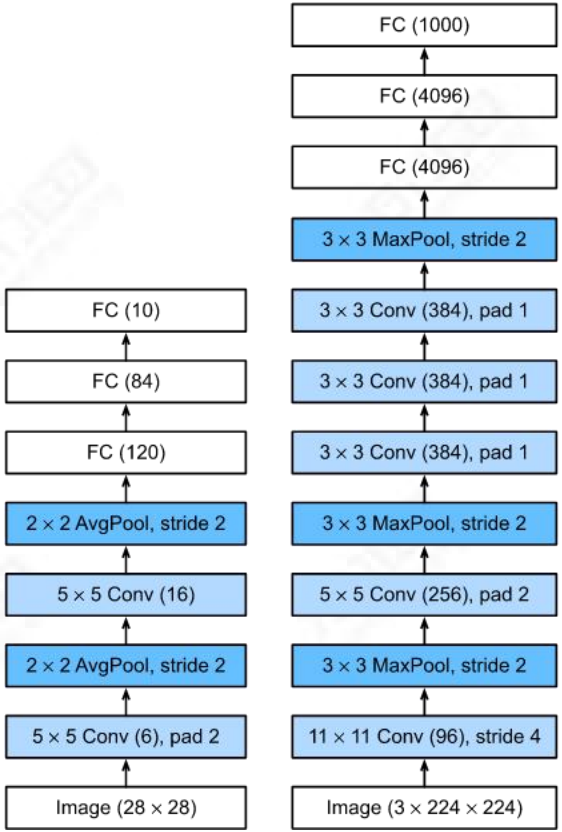


LeNet (左)， AlexNet (右)

深度卷积神经网络(AlexNet)

- AlexNet 将 sigmoid 激活函数改为更简单的 ReLU 激活函数。
- 一方面，ReLU 激活函数的计算更简单，它不需要如 sigmoid 激活函数那般复杂的求幂运算。
- 另一方面，当使用不同的参数初始化方法时，ReLU 激活函数使训练模型更加容易。 当 sigmoid 激活函数的输出非常接近于 0 或 1 时，这些区域的梯度几乎为 0，因此反向传播无法继续更新一些模型参数。 相反，ReLU 激活函数在正区间的梯度总是1。
- 因此，如果模型参数没有正确初始化，sigmoid 函数可能在正区间内得到几乎为 0 的梯度，从而使模型无法得到有效的训练。

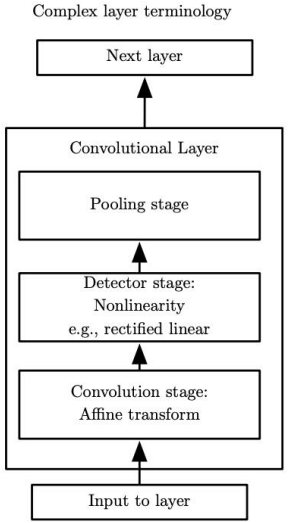
“ Talk is cheap. Show me the code. ”



LeNet (左)， AlexNet (右)

使用块的网络(VGG)

- 虽然 AlexNet 证明深层神经网络卓有成效，但它没有提供一个通用的模板来指导后续的研究人员设计新的网络。
- 在下面的几个小节中，我们将介绍一些常用于设计深层神经网络的启发式概念。
- 经典卷积神经网络的基本组成部分是下面的这个序列：
  1. 带填充以保持分辨率的卷积层；
  2. 非线性激活函数，如 ReLU；
  3. 池化层，如最大池化层。

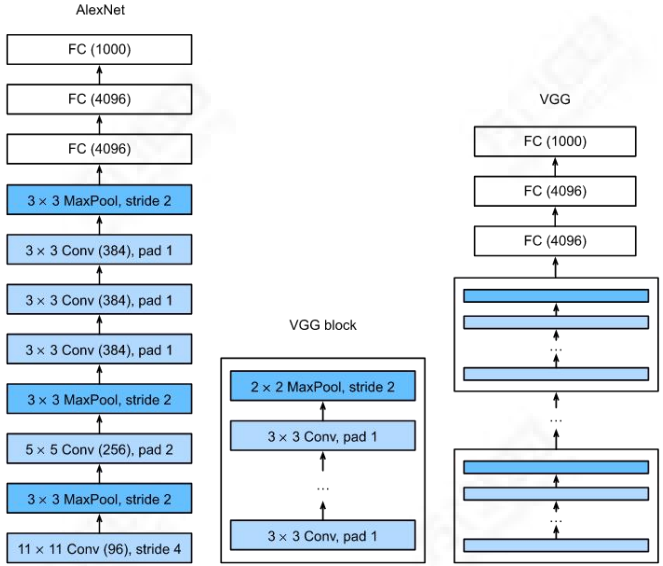


- Bengio Y, Goodfellow I, Courville A. Deep learning[M]. Massachusetts, USA:: *MIT press*, 2017.

使用块的网络(VGG)

- 一个 VGG 块由一系列卷积层组成，后面再加上用于空间下采样的最大池化层。在最初的 VGG 论文 [Simonyan & Zisserman, 2014] 中，作者使用了带有 3×3 卷积核、填充为 1 (保持高度和宽度) 的卷积层，和带有 2×2 池化窗口、步幅为 2 (每个块后的分辨率减半) 的最大池化层。
- 与 AlexNet、LeNet 一样，VGG 网络可以分为两部分：第一部分主要由卷积层和池化层组成，第二部分由全连接层组成。
- 原始 VGG 网络有 5 个卷积块，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。由于该网络使用 8 个卷积层和 3 个全连接层，因此它通常被称为 VGG-11。

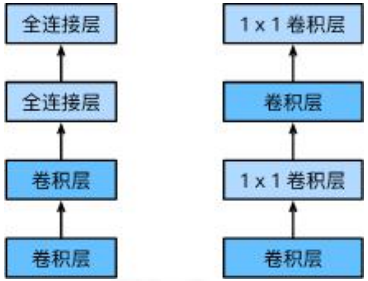
“ Talk is cheap. Show me the code. ”



- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556.*

网络中的网络（NiN）

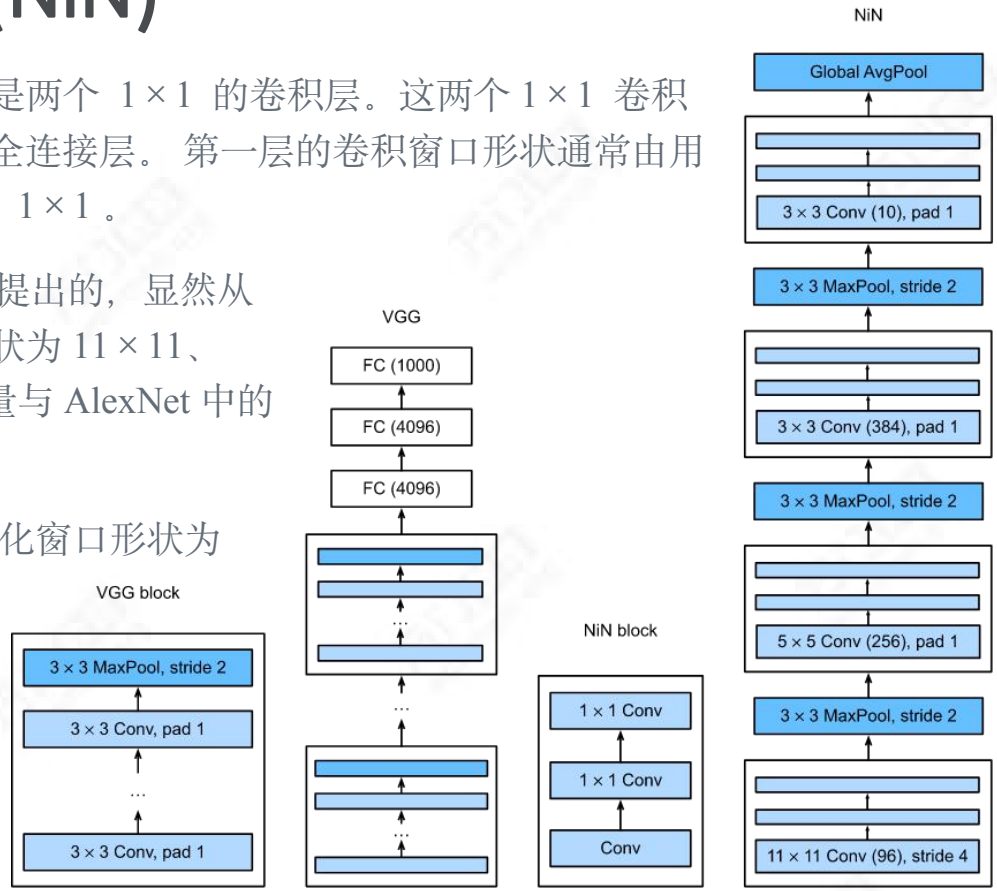
- AlexNet 和 VGG 对 LeNet 的改进主要在于如何扩大和加深这两个模块。即串联多个由卷积层和“全连接”层构成的小网络来构建一个深层网络。
- 网络中的网络 (NiN) 提供了一个非常简单的解决方案：在每个像素的通道上分别使用多层感知机 [Lin et al., 2013]。
- 回想一下，卷积层的输入和输出由四维张量组成，张量的每个轴分别对应样本、通道、高度和宽度。 另外，全连接层的输入和输出通常是分别对应于样本和特征的二维张量。
- NiN 的想法是在每个像素位置（针对每个高度和宽度）应用一个全连接层。如果我们将权重连接到每个空间位置，我们可以将其视为 1×1 卷积层 (如上一讲中所述)，或作为在每个像素位置上独立作用的全连接层。 从另一个角度看，即将空间维度中的每个像素视为单个样本，将通道维度视为不同特征 (feature)。



- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

网络中的网络（NiN）

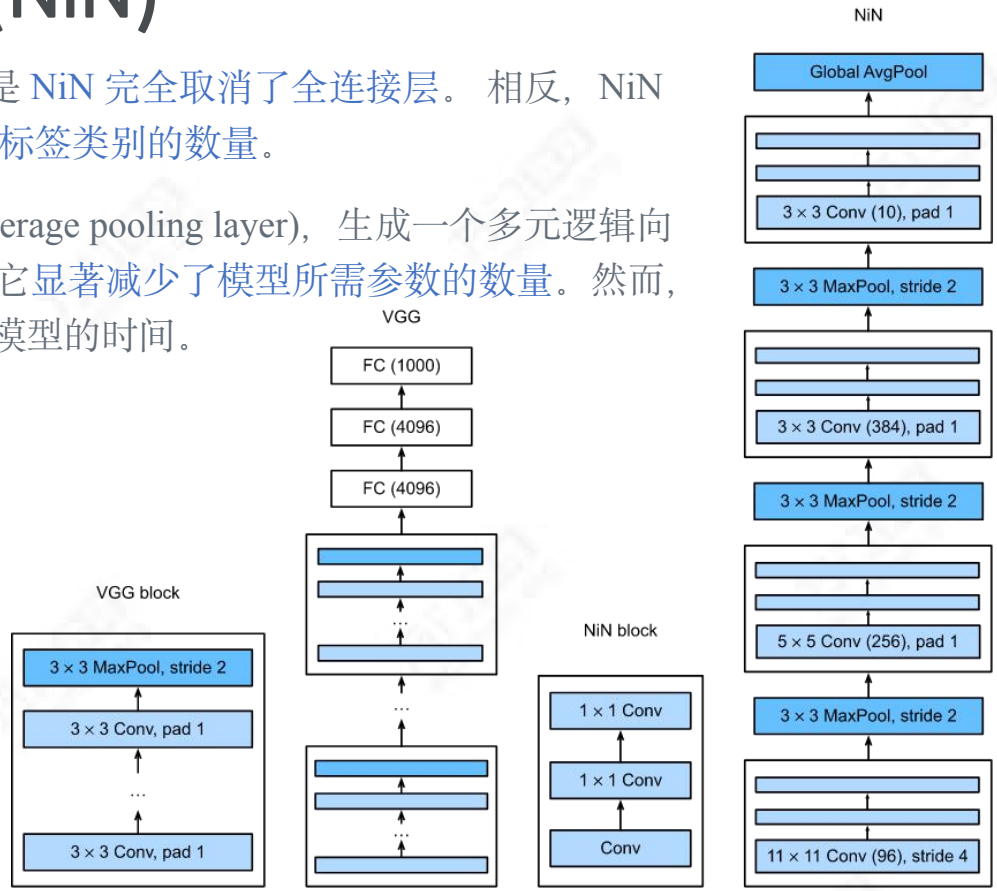
- NiN 块以一个普通卷积层开始，后面是两个 1×1 的卷积层。这两个 1×1 卷积层充当带有 ReLU 激活函数的逐像素全连接层。 第一层的卷积窗口形状通常由用户设置。 随后的卷积窗口形状固定为 1×1 。
- 最初的 NiN 网络是在 AlexNet 后不久提出的，显然从中得到了一些启示。NiN 使用窗口形状为 11×11、5×5 和 3×3 的卷积层，输出通道数量与 AlexNet 中的相同。
- 每个 NiN 块后有一个最大池化层，池化窗口形状为 3×3，步幅为 2。
- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.



网络中的网络（NiN）

- NiN 和 AlexNet 之间的一个显著区别是 NiN 完全取消了全连接层。 相反，NiN 使用一个 NiN 块，其输出通道数等于标签类别的数量。
- 最后放一个全局平均池化层 (global average pooling layer)，生成一个多元逻辑向量 (logits)。NiN 设计的一个优点是，它显著减少了模型所需参数的数量。然而，在实践中，这种设计有时会增加训练模型的时间。

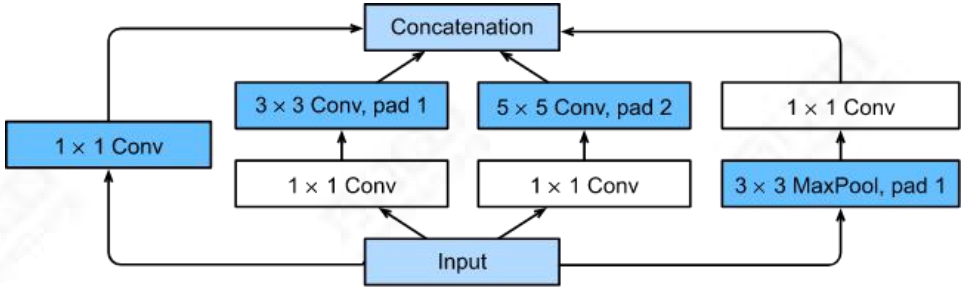
“ Talk is cheap. Show me the code. ”



- Lin, M., Chen, Q., & Yan, S. (2013). Network in network. *arXiv preprint arXiv:1312.4400*.

含并行连结的网络(GoogLeNet)

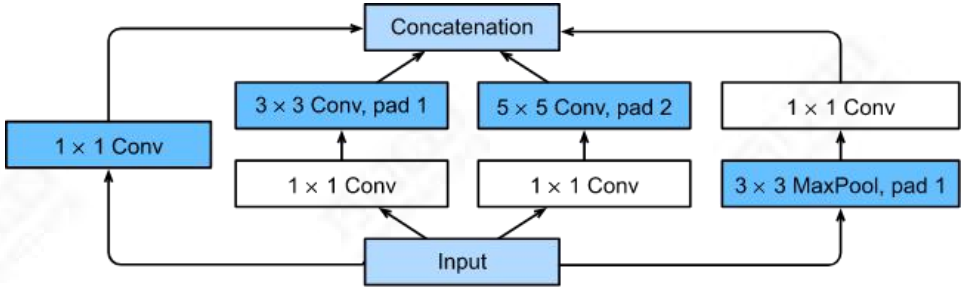
- 在2014年的ImageNet图像识别挑战赛中，一个名叫 GoogLeNet [Szegedy et al., 2015] 的网络结构大放异彩。 GoogLeNet 吸收了 NiN 中串联网络的思想，并在此基础上做了改进。 这篇论文的一个重点是解决了什么样大小的卷积核最合适的问题。 毕竟，以前流行的网络使用小到 1×1 ，大到 11×11 的卷积核。 此文的一个观点是，有时使用不同大小的卷积核组合是有利的。
- 在本节中，我们将介绍一个稍微简化的 GoogLeNet 版本：我们省略了一些为稳定训练而添加的特殊特性，但是现在有了更好的训练算法，这些特性不是必要的。



- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

含并行连结的网络 (GoogLeNet)

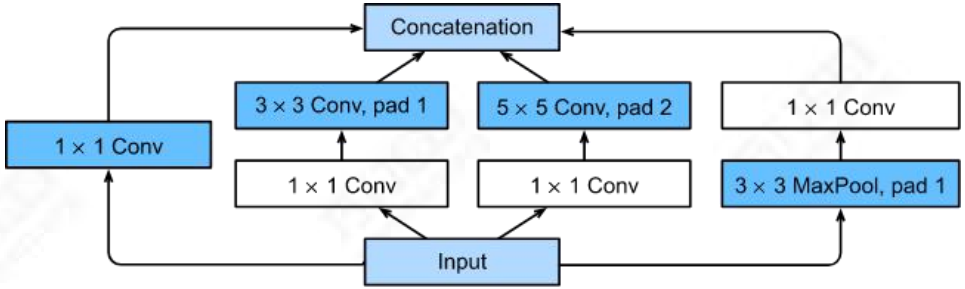
- 在 GoogLeNet 中，基本的卷积块被称为 Inception 块 (Inception block) 。这很可能得名于电影《盗梦空间》(Inception)，因为电影中的一句话“我们需要走得更深”(“We need to go deeper”)。
- 如下图所示，Inception 块由四条并行路径组成。 前三条路径使用窗口大小为 1×1 、3×3 和 5×5 的卷积层，从不同空间大小中提取信息。 中间的两条路径在输入上执行 1×1 卷积，以减少通道数，从而降低模型的复杂性。 第四条路径使用 3×3 最大池化层，然后使用 1×1 卷积层来改变通道数。 这四条路径都使用合适的填充来使输入与输出的高和宽一致，最后我们将每条线路的输出在通道维度上连结，并构成 Inception 块的输出。



- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

含并行连结的网络 (GoogLeNet)

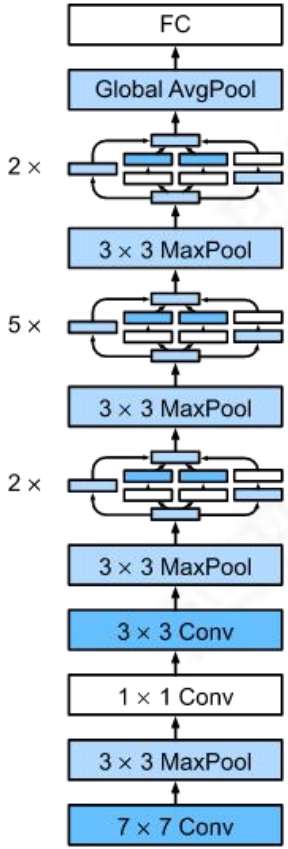
- 在 Inception 块中，通常调整的超参数是每层输出通道的数量。
- 那么为什么 GoogLeNet 这个网络如此有效呢？
  - 首先我们考虑一下滤波器（filter）的组合，它们可以用各种滤波器尺寸探索图像，这意味着不同大小的滤波器可以有效地识别不同范围的图像细节。
  - 同时，我们可以为不同的滤波器分配不同数量的参数。



- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

含并行连结的网络 (GoogLeNet)

- 如右图所示，GoogLeNet 一共使用 9 个 Inception 块和全局平均池化层的堆叠来生成其估计值。
- Inception 块之间的最大池化层可降低维度。
- 第一个模块类似于 AlexNet 和 LeNet，Inception块的栈从VGG继承，全局平均池化层避免了在最后使用全连接层。



“ Talk is cheap. Show me the code. ”

- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1–9).

批量归一化

- 训练深层神经网络是十分困难的，特别是在较短的时间内使他们收敛更加棘手。
- 我们将介绍批量归一化 (batch normalization) [Ioffe & Szegedy, 2015]，这是一种流行且有效的技术，可持续加速深层网络的收敛速度。再结合在后面要介绍的残差块，批量归一化使得研究人员能够训练 100 层以上的网络。
- 为什么需要批量归一化层呢?
    1. 数据预处理的标准化方式通常会对最终结果产生巨大影响。
    2. 对于典型的多层感知机或卷积神经网络，训练中间层中的变量可能具有更广的变化范围
    3. 更深层的网络很复杂，容易过拟合。这意味着正则化变得更加重要。
- 批量归一化应用于单个可选层 (也可以应用到所有层)，其原理如下：
  在每次训练迭代中，我们首先归一化输入，即通过减去其均值并除以其标准差，其中两者均基于当前小批量处理。接下来，我们应用比例系数和比例偏移。正是由于这个基于批量统计的标准化，才有了批量归一化的名称。。

批量归一化

- 先考虑如何对全连接层做批量归一化。
- 通常，我们将批量归一化层置于全连接层中的仿射变换和激活函数之间。设全连接层的输入为 $\boldsymbol{u}$，权重参数和偏差参数分别为 $\boldsymbol{W}$ 和 $\boldsymbol{b}$，激活函数为 ϕ 。设批量归一化的运算符为 BN。那么，使用批量归一化的全连接层的输出为 $\phi(\mathrm{BN}(\boldsymbol{x}))$，其中批量归一化输入 $\boldsymbol{x}$ 由仿射变换 $\boldsymbol{x} = \boldsymbol{W}\boldsymbol{u} + \boldsymbol{b}$ 得到。
- 考虑一个由 m 个样本组成的小批量，仿射变换的输出为一个新的小批量 $\mathcal{B} = \left\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\right\}$ 。它们正是批量归一化层的输入。对于小批量 $\mathcal{B}$ 中任意样本 $\boldsymbol{x}^{(i)} \in \mathbb{R}^d, 1 \leq i \leq m$, 批量归一化层的输出同样是 d 维向量

$$\boldsymbol{y}^{(i)} = \mathrm{BN}\left(\boldsymbol{x}^{(i)}\right)$$

- 并由以下几步求得。

- Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

批量归一化

- 首先，对小批量 $\mathcal{B}$ 求均值和方差：

$$\boldsymbol{\mu}_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}^{(i)}$$
$$\boldsymbol{\sigma}_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{\mathcal{B}}\right)^2,$$

  其中的平方计算是按元素求平方。

- 接下来，使用按元素开方和按元素除法对 $\boldsymbol{x}^{(i)}$ 标准化：

$$\hat{\boldsymbol{x}}^{(i)} \leftarrow \frac{\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_{\mathcal{B}}}{\sqrt{\boldsymbol{\sigma}_{\mathcal{B}}^2 + \epsilon}},$$

  这里 $\epsilon > 0$ 是一个很小的常数，保证分母大于 0。

- 在上面标准化的基础上，批量归一化层引入了两个可以学习的模型参数，拉伸 (scale)参数 $\boldsymbol{\gamma}$ 和偏移 (shift) 参数 $\boldsymbol{\beta}$。这两个参数和 $\boldsymbol{x}^{(i)}$ 形状相同，皆为 d 维向量。它们与 $\boldsymbol{x}^{(i)}$ 分别做按元素乘法 (符号 $\odot$) 和加法计算：

$$\boldsymbol{y}^{(i)} \leftarrow \boldsymbol{\gamma} \odot \hat{\boldsymbol{x}}^{(i)} + \boldsymbol{\beta}$$

- 至此，我们得到了 $\boldsymbol{x}^{(i)}$ 的批量归一化的输出 $\boldsymbol{y}^{(i)}$。值得注意的是，可学习的拉伸和偏移参数保留了不对 $\hat{\boldsymbol{x}}^{(i)}$ 做批量归一化的可能：此时只需学出 $\gamma = \sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}$ 和 $\boldsymbol{\beta} = \boldsymbol{\mu}_{\mathcal{B}}$。我们可以对此这样理解：如果批量归一化无益，理论上，学出的模型可以不使用批量归一化。

批量归一化

- 对卷积层来说，批量归一化发生在卷积计算之后、应用激活函数之前。如果卷积计算输出多个通道，我们需要对这些通道的输出分别做批量归一化，且每个通道都拥有独立的拉伸和偏移参数，并均为标量。设小批量中有 m 个样本。在单个通道上，假设卷积计算输出的高和宽分别为 p 和 q。我们需要对该通道中 $m \times p \times q$ 个元素同时做批量归一化。对这些元素做标准化计算时，我们使用相同的均值和方差，即该通道中 $m \times p \times q$ 个元素的均值和方差。

“ Talk is cheap. Show me the code. ”

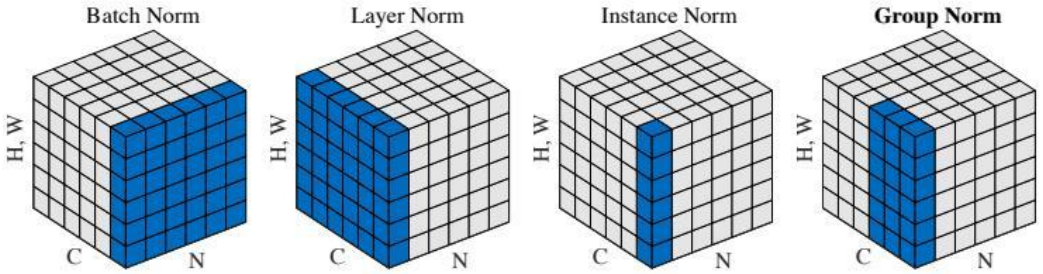


Figure 2. **Normalization methods**. Each subplot shows a feature map tensor, with N as the batch axis, C as the channel axis, and (H, W) as the spatial axes. The pixels in blue are normalized by the same mean and variance, computed by aggregating the values of these pixels.

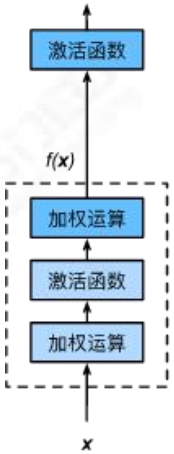
残差网络(ResNet)

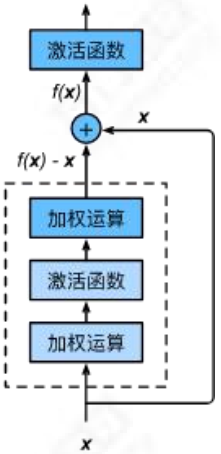
- 思考一个问题：对神经网络模型添加新的层，充分训练后的模型是否只可能更有效地降低训练误差?
- 理论上，原模型解的空间只是新模型解的空间的子空间。也就是说，如果我们能将新添加的层训练成恒等映射$f(x) = x$，新模型和原模型将同样有效。由于新模型可能得出更优的解来拟合训练数据集，因此添加层似乎更容易降低训练误差。然而在实践中，添加过多的层后训练误差往往不降反升。即使利用批量归一化带来的数值稳定性使训练深层模型更加容易，该问题仍然存在。
- 针对这一问题，何恺明等人提出了残差网络 (ResNet)。它在 2015 年的 ImageNet 图像识别挑战赛夺魁，并深刻影响了后来的深度神经网络的设计。

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770–778).*

残差网络(ResNet)

- 让我们聚焦于神经网络局部。如右图所示，设输入为 $\boldsymbol{x}$ 。假设我们希望学出的理想映射为 $f(\boldsymbol{x})$，从而作为右图中的上方激活函数的输入。左图虚线框中的部分需要直接拟合出该映射 $f(\boldsymbol{x})$，而右图虚线框中的部分则需要拟合出有关恒等映射的残差映射 $f(x)-x$ 。
- 残差映射在实际中往往更容易优化。以恒等映射作为我们希望学出的理想映射 $f(\boldsymbol{x})$。我们只需将右图中的右图虚线框内上方的加权运算 (如仿射) 的权重和偏差参数学成 0，那么 $f(\boldsymbol{x})$ 即为恒等映射。实际中，当理想映射 $f(\boldsymbol{x})$ 极接近于恒等映射时，残差映射也易于捕捉恒等映射的细微波动。右图中的右侧也就是 ResNet 的基础块，即残差块 (residual block)。在残差块中，输入可通过跨层的数据线路更快地向前传播。

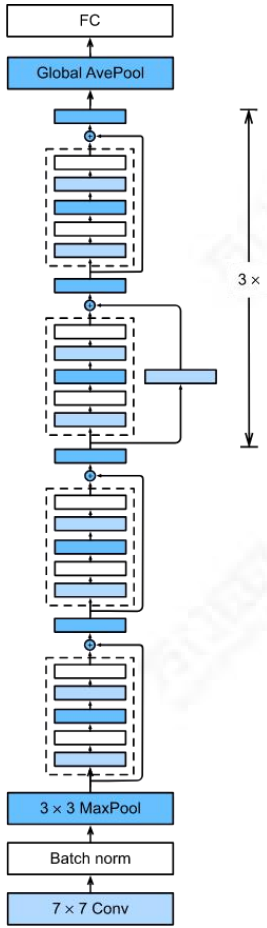




残差网络(ResNet)

- ResNet 沿用了 VGG 全 3 × 3 卷积层的设计。残差块里首先有 2 个有相同输出通道数的 3 × 3 卷积层。每个卷积层后接一个批量归一化层和 ReLU 激活函数。然后我们将输入跳过这两个卷积运算后直接加在最后的 ReLU 激活函数前。这样的设计要求两个卷积层的输出与输入形状一样，从而可以相加。如果想改变通道数，就需要引入一个额外的 1 × 1 卷积层来将输入变换成需要的形状后再做相加运算。

“ Talk is cheap. Show me the code. ”



稠密连接网络(DenseNet)

- ResNet极大地改变了如何参数化深层网络中函数的观点。 稠密连接网络 (DenseNet) [Huang et al., 2017] 在某种程度上是 ResNet 的逻辑扩展。让我们先从数学上了解一下。
- 回想一下任意函数的泰勒展开式 (Taylor expansion)，它把这个函数分解成越来越高阶的项。在 x 接近 0 时，

$$f(x) = f(0) + f'(0)x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \dots$$

- 同样，ResNet 将函数展开为

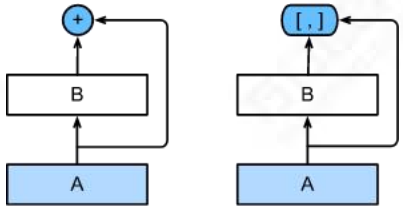
$$f(\mathbf{x}) = \mathbf{x} + g(\mathbf{x})$$

- 也就是说，ResNet 将 f 分解为两部分：一个简单的线性项和一个更复杂的非线性项。那么再向前拓展一步，如果我们想将 f 拓展成超过两部分的信息呢？ 一种方案便是 DenseNet。

- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700–4708).*

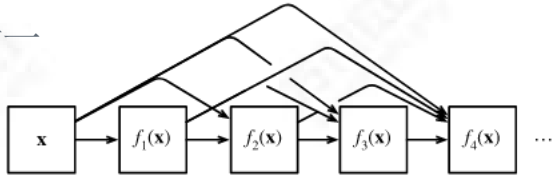
稠密连接网络(DenseNet)

- 如右图所示，ResNet 和 DenseNet 的关键区别在于，DenseNet 输出是连接（用图中的 [,] 表示）而不是如 ResNet 的简单相加。因此，在应用越来越复杂的函数序列后，我们执行从 $\mathbf{x}$ 到其展开式的映射：



$$\mathbf{x} \to [\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})]), f_3([\mathbf{x}, f_1(\mathbf{x}), f_2([\mathbf{x}, f_1(\mathbf{x})])]), \ldots]$$

- 最后，将这些展开式结合到多层感知机中，再次减少特征的数量。实现起来非常简单：我们不需要添加术语，而是将它们连接起来。DenseNet 这个名字由变量之间的“稠密连接”而得来，最后一层与之前的所有层紧密相连。稠密连接如右图所示。



- 稠密网络主要由 2 部分构成： 稠密块 (dense block) 和过渡层 (transition layer)。 前者定义如何连接输入和输出，而后者则控制通道数量，使其不会太复杂。

“Talk is cheap. Show me the code.”

小结

- 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。
- VGG-11 使用可复用的卷积块构造网络。
- NiN 使用由一个卷积层和多个 1×1 卷积层组成的块。该块可以在卷积神经网络中使用，以允许更多的每像素非线性。
- GoogLeNet 将多个设计精细的 Inception 块与其他层（卷积层、全连接层）串联起来。其中 Inception 块的通道数分配之比是在 ImageNet 数据集上通过大量的实验得来的。
- 在模型训练过程中，批量归一化利用小批量的均值和标准差，不断调整神经网络的中间输出，使整个神经网络各层的中间输出值更加稳定。
- 利用残差块（residual blocks）可以训练出一个有效的深层神经网络：输入可以通过层间的残余连接更快地向前传播。
- 在跨层连接上，不同于 ResNet 中将输入与输出相加，稠密连接网络（DenseNet）在通道维上连结输入与输出。

谢谢观看

更多好课，请关注万门大学APP