

Inception, 批量归一化和残差网络(ResNet)

越深越好？一个简单而深刻的问题

- 背景 (2012-2014): 从AlexNet到VGG，学术界和工业界发现，增加神经网络的深度通常能带来性能的提升。
 - VGGNet通过堆叠大量简单的3x3卷积，证明了“深度”本身是提取高级语义特征的关键。
- 浮现的瓶颈：
 - 计算效率问题：更深、更大的卷积核意味着巨大的参数量和计算成本。
 - 训练稳定性问题：网络越深，梯度消失/爆炸问题越严重，训练变得极其困难，收敛缓慢。
 - 网络退化问题 (Degradation): 当网络达到一定深度后，继续增加层数，训练集准确率反而会下降。这并非过拟合，而是网络本身难以优化。

概要

➤ Inception

- 卷积的不均匀混合（不同深度）
- 批量归一正则化

➤ ResNet

- 泰勒展开式
- 残差网络（ResNext） 分解卷积

➤ Zoo

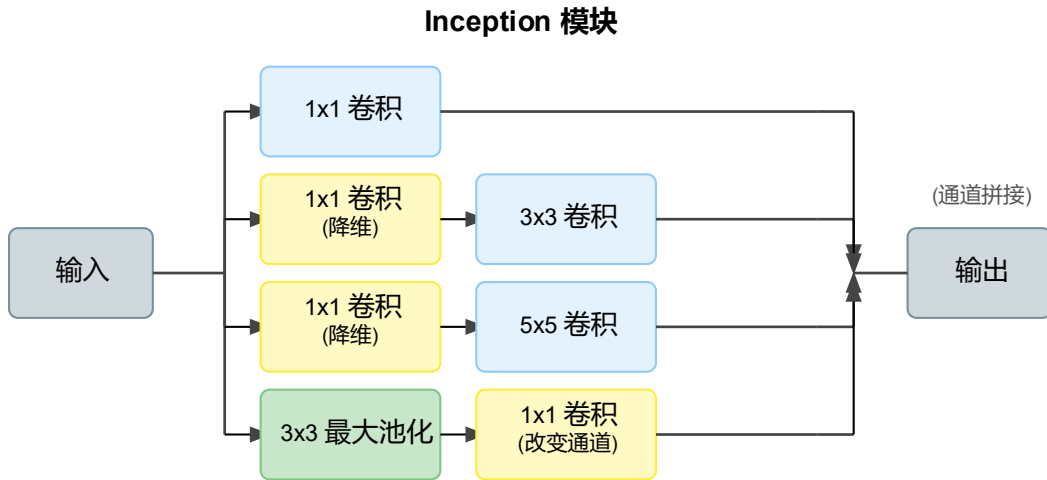
稠密连接网络（DenseNet）, ShuffleNet, 可分解卷积层, ...

Inception - 从“宽度”和“效率”入手

- 哪种尺寸的卷积核是最好的？1x1, 3x3, 还是 5x5?
 - 大卷积核感受野大，适合捕捉全局特征
 - 小卷积核计算量小，适合捕捉局部细节
- Inception的核心思想： 并行处理，特征融合
 - 在一个网络块内，同时使用多种不同尺寸的卷积核（和池化）进行特征提取
 - 将所有路径提取到的特征在通道维度上拼接 (Concatenate) 起来
 - 这样，网络就可以自适应地学习，在当前阶段应该更多地依赖哪种尺度的特征
- 一种“横向扩展”网络的思路，在增加网络表达能力的同时，力求高效

深入Inception: 1x1卷积的妙用

➤ Inception V1 模块结构图



选最合适的卷积 ...

1x1

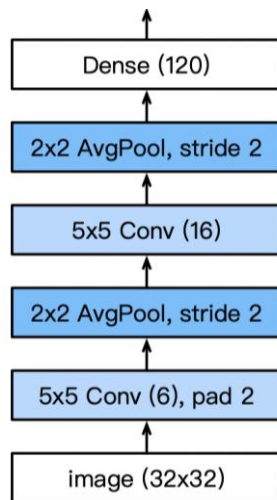
3x3

5x5

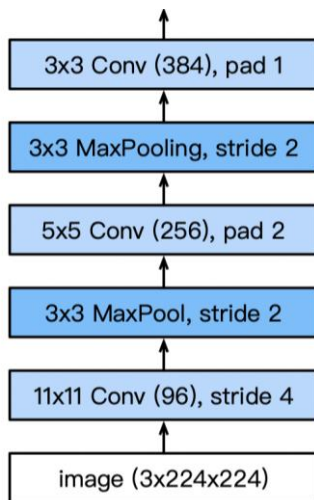
最大汇聚

许多 1x1

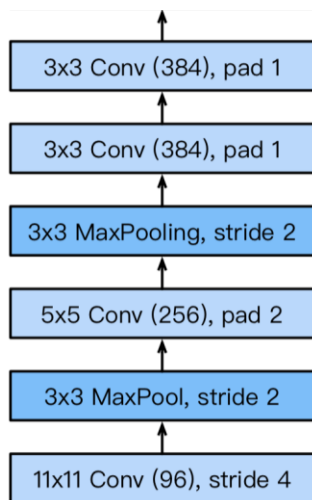
LeNet



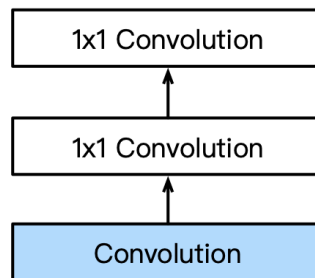
AlexNet



VGG

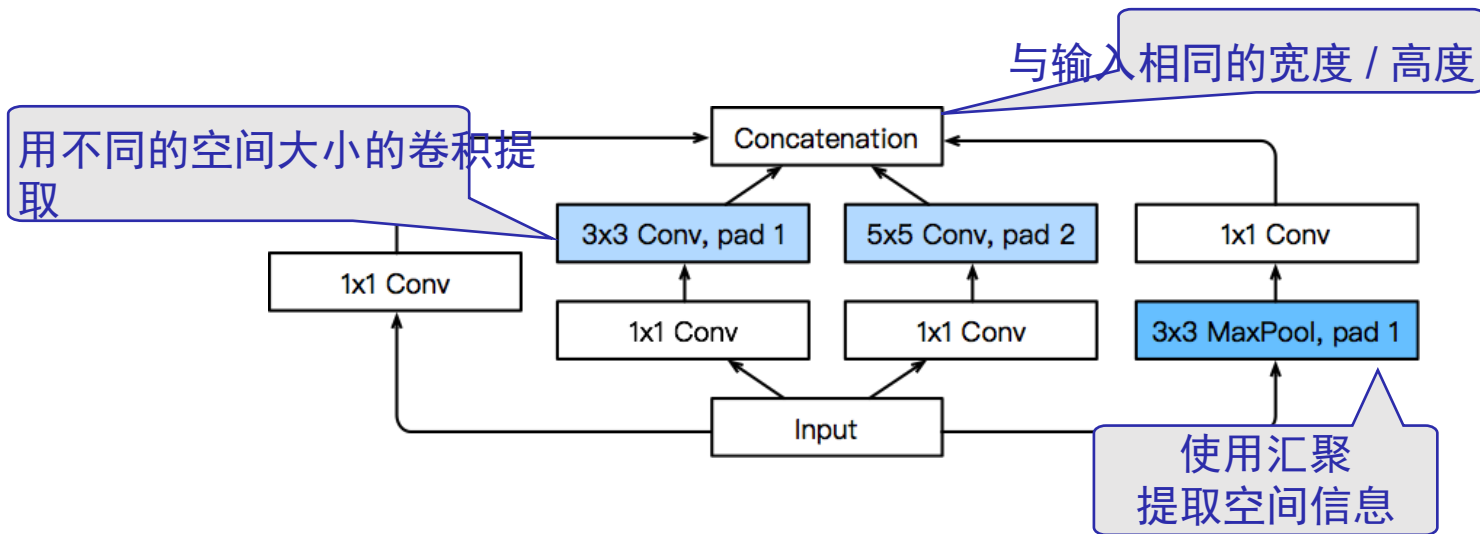


NiN



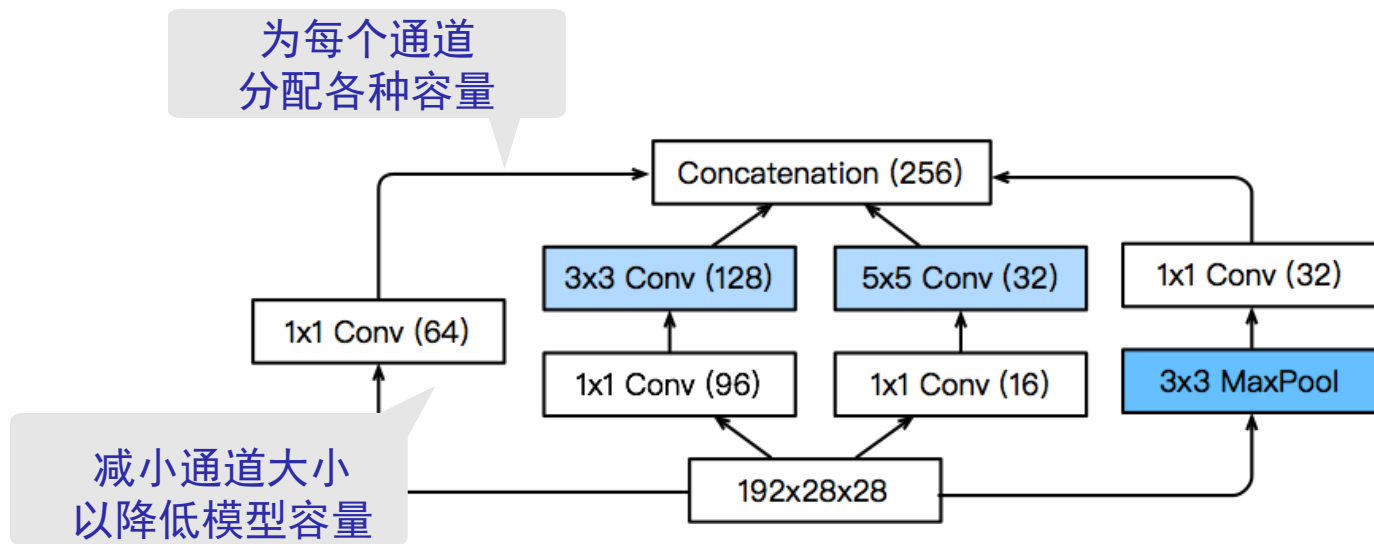
Inception块

➤ 4个路径从不同方面提取信息，然后拼接作为输出通道



Inception块

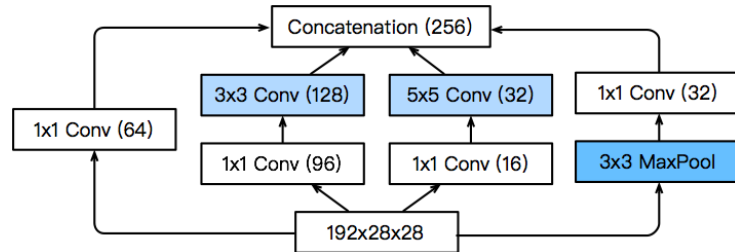
➤（第一个初始块）指定的通道大小



Inception块

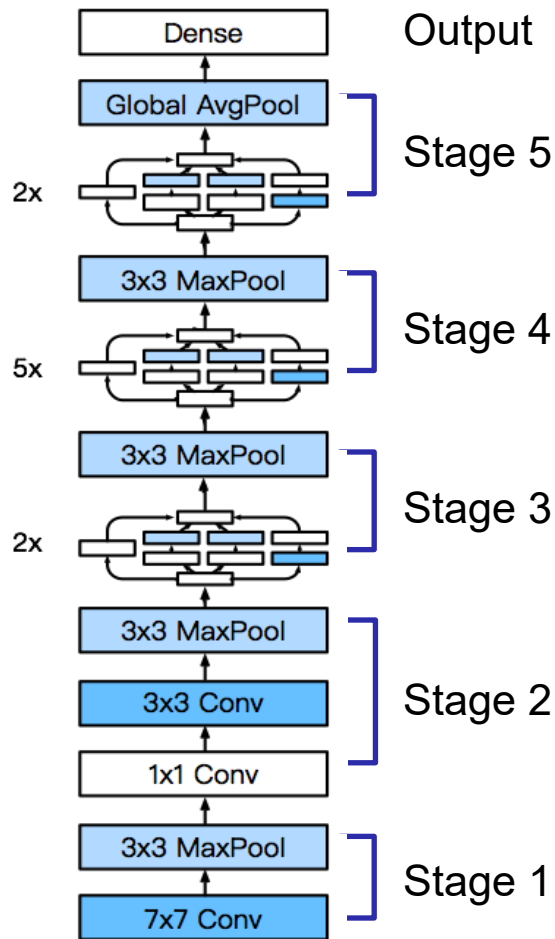
- 与单个3x3或5x5卷积层相比，初始块具有更少的参数和更低的计算复杂度
 - 不同功能混合（多样的功能类）
 - 卷积核计算高效（良好的泛化）

	# 参数	浮点运算 FLOPS
Inception	0.16 M	128 M
3x3 卷积	0.44 M	346 M
5x5 卷积	1.22 M	963 M



GoogLeNet

- 5 个阶段
- 9 个 Inception 块



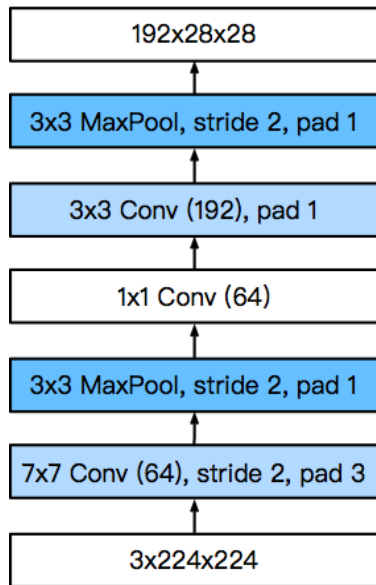
阶段1&2

➤ 由于更多层：

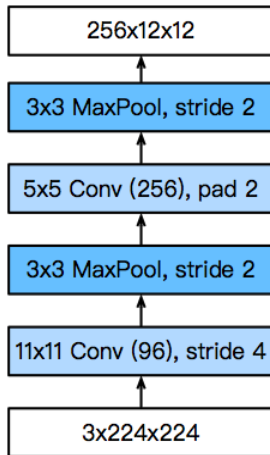
➤ 更小的内核

➤ 更小的输出通道

GoogLeNet



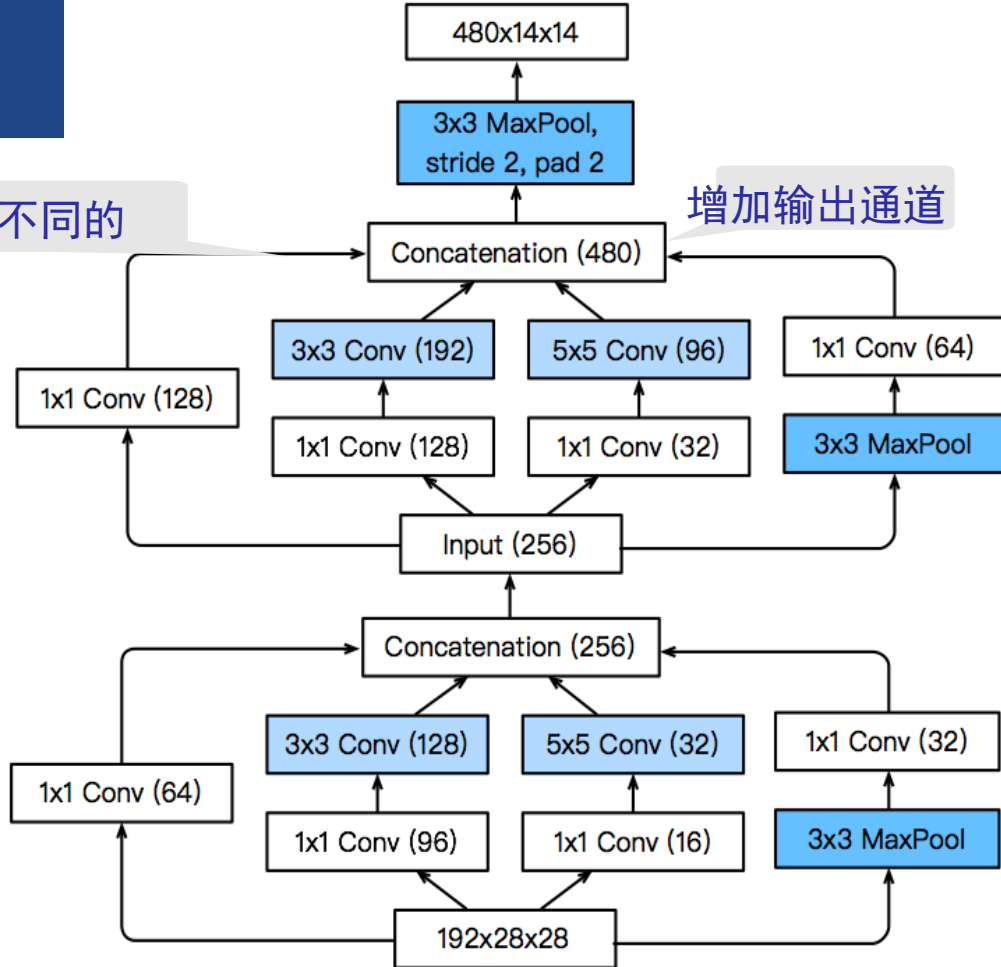
AlexNet



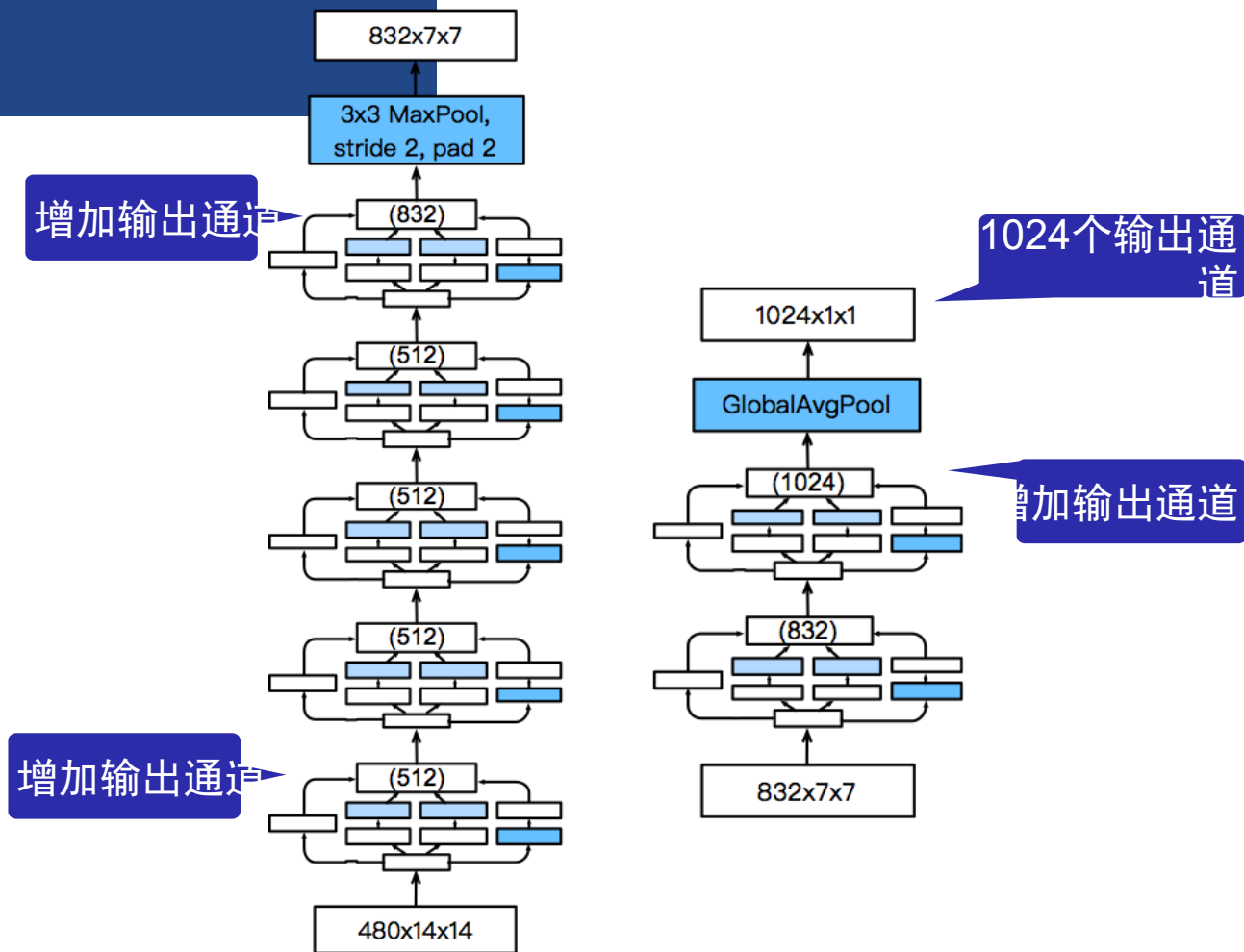
阶段3

通道分配是不同的

增加输出通道



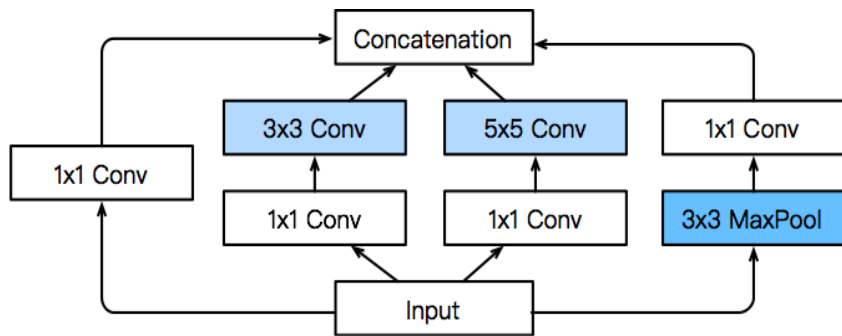
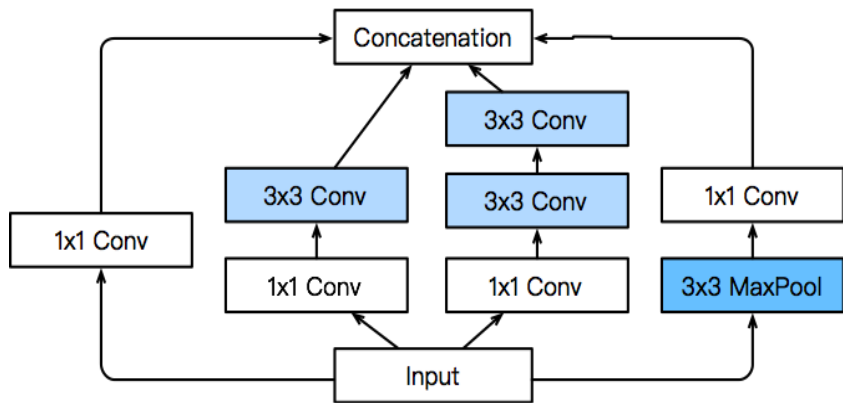
阶段4&5



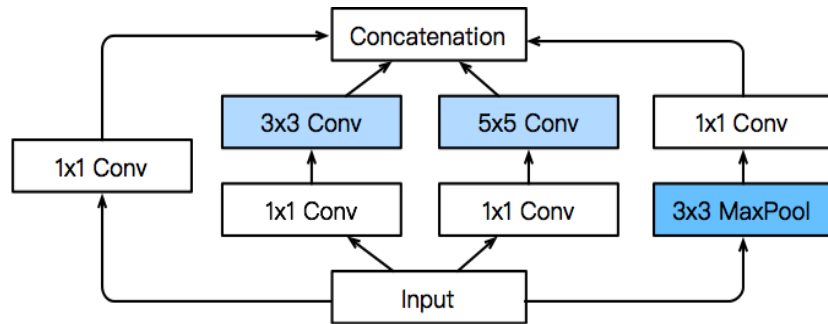
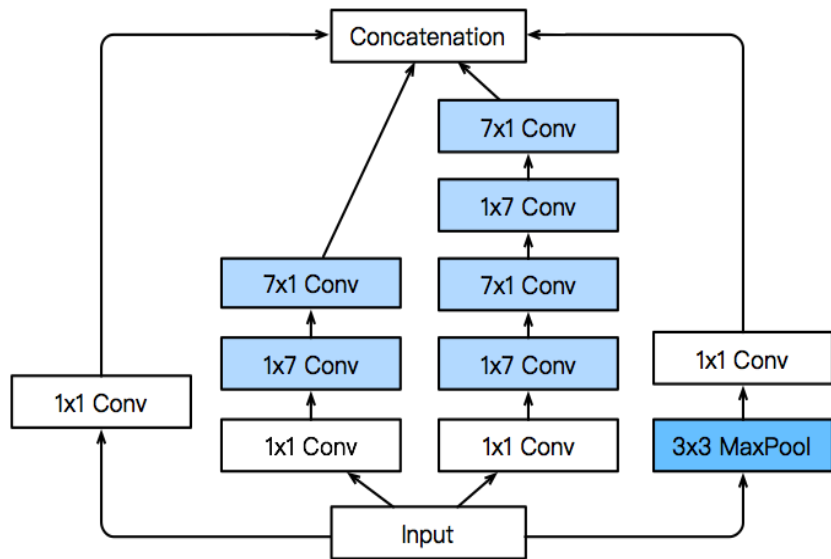
许多种类的Inception网络

- Inception-BN (v2) - 添加批量归一化
- Inception-V3 - 修改了初始块
 - 用多个 3x3 卷积替换 5x5
 - 用 1x7 和 7x1 卷积替换 5x5
 - 用 1x3 和 3x1 卷积替换 3x3
 - 通常用更深的堆
- Inception-V4 - 添加残差块连接

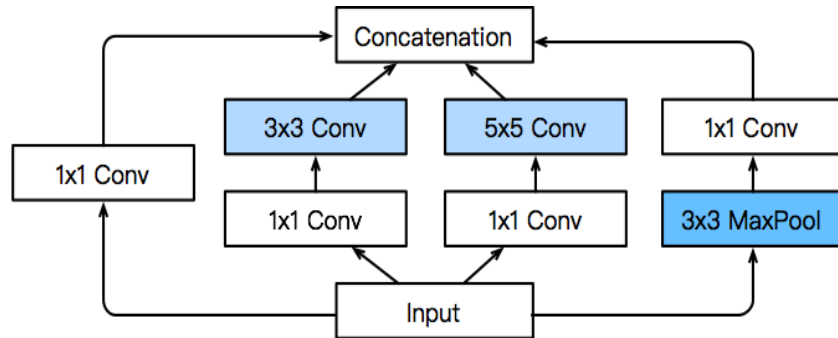
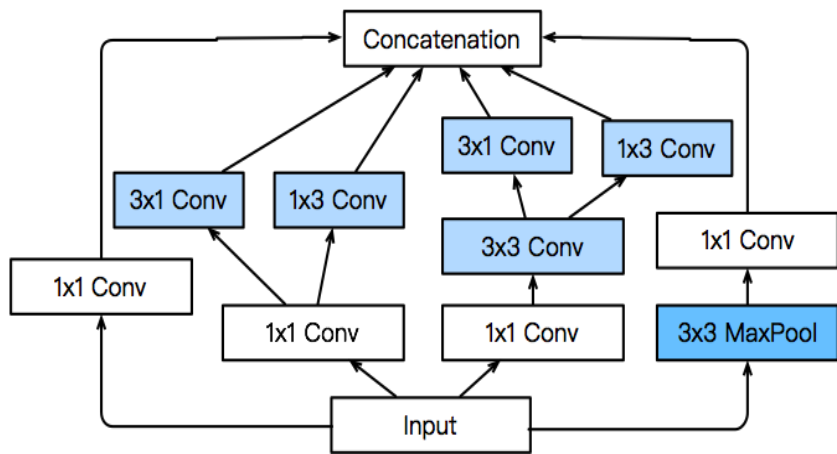
Inception V3块 - 阶段3



Inception V3块 - 阶段4



Inception V3 块 - 阶段 5



Batch Normalization



批量归一化 (BN): 深度学习的“稳定器”与“加速器”

➤ 为什么深度神经网络 (DNN) 难以训练?

- 随着网络层数加深, 训练变得极其缓慢且不稳定
- 模型性能对学习率、参数初始化等超参数极为敏感

➤ 一个直观的类比: 建造摩天大楼

- 训练深层网络, 就像一层层地盖楼
- 如果下面几层在施工时 (参数更新), 地基和结构还在不停地晃动 (数据分布变化), 那么上面楼层的施工将异常困难, 甚至可能导致整个建筑的崩塌

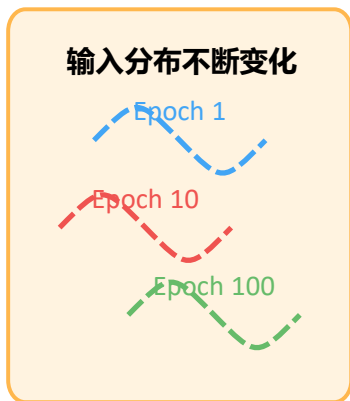
困境：不稳定的内部环境 (内部协变量偏移)

➤ 内部协变量偏移 (Internal Covariate Shift, ICS)

- 在训练过程中，由于前序网络层的参数不断更新，导致后序网络层接收到的输入数据分布持续发生变化
- 后续层被迫不断适应新的数据分布，就像在追逐一个移动的靶心，这极大地拖慢了整个网络的收敛速度

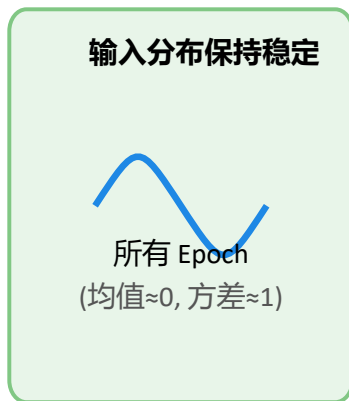
网络层 L+1 的学习困境

传统网络



后果：学习效率低下

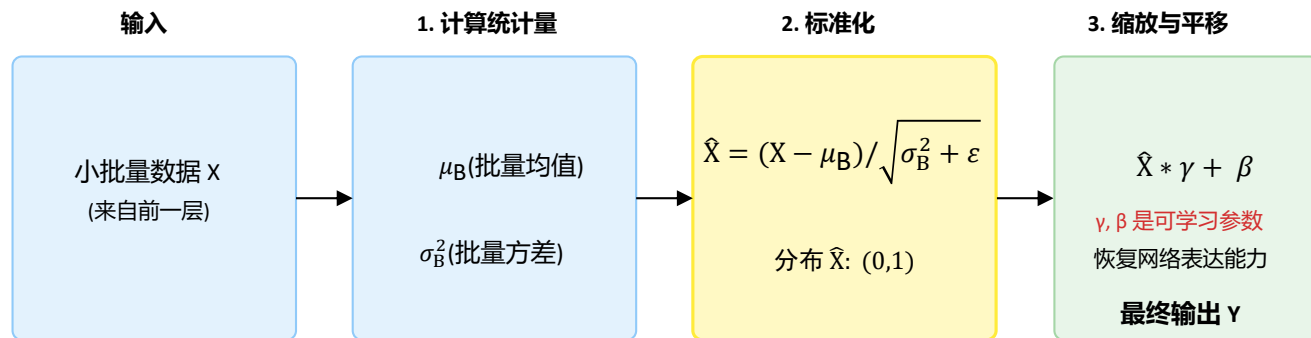
引入批量归一化



结果：学习环境稳定

BN的核心机制：标准化 + 自适应调整

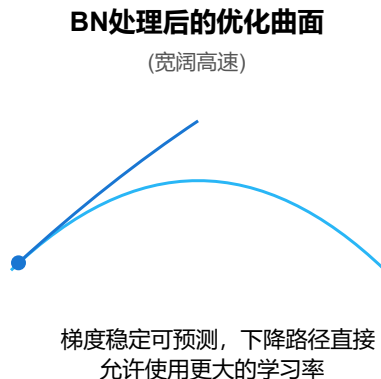
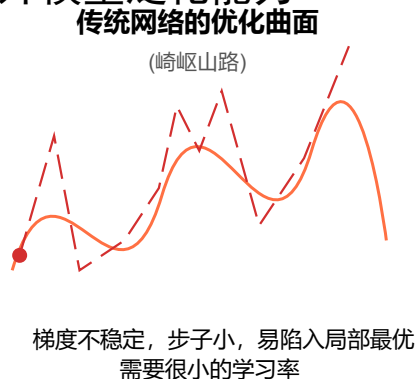
- BN层在训练时对每个小批量（Mini-Batch）数据执行四步操作
- 为何需要第3步（缩放与平移）？
 - 如果只做标准化，会强制将每层输入都限制在激活函数（如Sigmoid）的线性区域，从而扼杀网络的非线性表达能力
 - 引入可学习的参数 γ 和 β ，相当于给了网络一个“撤销”标准化的权利。网络可以自主学习将数据调整到最适合下一层学习的分布，既保证了稳定性，又不失灵活性



BN更大的作用：平滑优化曲面

➤ 后续研究发现，BN的成功并不仅仅因为解决了ICS。其更本质的作用是极大地平滑了损失函数的优化曲面

- 允许更高的学习率：优化过程不再那么“颠簸”，可以用更大的步长（学习率）快速前进，极大加速了模型收敛
- 降低对初始化的敏感度：无论从哪个初始点出发，都更容易找到通往最优解的平坦大道
- 自带正则化效果：每个样本的输出都受到同批次其他样本的影响，这种轻微的噪声起到了正则化作用，有助于提升模型泛化能力



关键细节：训练 vs. 推理 & 批量大小的影响

➤ BN在训练和推理（预测）阶段的行为模式不同，这是实践中必须理解的关键点。

阶段	训练 (Training)	推理 (Inference)
目标	学习网络参数，稳定训练过程	对新样本进行确定性预测
统计量来源	当前小批量的均值 μ_B 和方差 σ_B^2	全局统计量 (在训练中通过移动平均估算出的 <code>running_mean</code> 和 <code>running_var</code>)
行为	动态计算，每次迭代都不同	固定不变，对任何输入都使用同一套统计量
参数更新	反向传播更新 γ, β ；同时更新全局统计量	所有参数均固定

➤ 批量大小 (Batch Size) 的重要性

➤ BN的假设：小批量的统计量是对全局数据统计量的良好近似

➤ 大批量 (如 64, 128, 256): 统计量稳定，BN效果好

➤ 小批量 (如 2, 4, 8): 统计量噪声大，波动剧烈，反而会损害模型性能

➤ 场景：受限于GPU显存的大模型、高分辨率图像任务

➤ 替代方案：在这种情况下，应考虑使用不依赖批量的归一化方法，如层归一化 (Layer Normalization) 或 组归一化 (Group Normalization)

总结：批量归一化的革命性贡献

- 批量归一化是深度学习发展史上的一个里程碑，使得训练真正“深”的网络成为可能
 - 加速训练 (Accelerates Training)
 - 允许使用更高的学习率，收敛速度提升数倍甚至数十倍
 - 稳定网络 (Stabilizes Network):
 - 极大缓解了梯度消失/爆炸问题，降低了对参数初始化的敏感度，让深度网络训练变得简单
 - 提升泛化 (Improves Generalization)
 - 其固有的噪声起到了正则化作用，在很多场景下可以减少甚至替代Dropout
 - 成为标配 (Became a Standard)
 - 自提出以来，迅速成为现代卷积神经网络（如ResNet及其变体）中不可或缺的标准组件
- BN的核心思想
 - 在网络内部动态地维持数据分布的稳定性，为后续的各种归一化技术铺平了道路

残差网络 (ResNet)

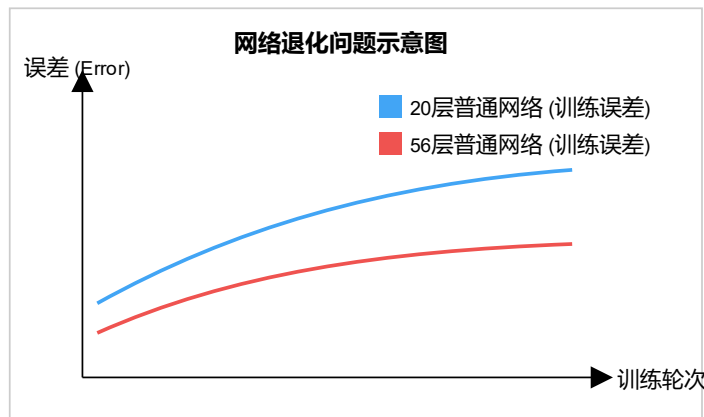
网络越深，效果一定越好吗？

➤理论预期

- 更深的网络 -> 更多参数 -> 更强的函数拟合能力 -> 理论上性能更优
- 一个深层网络至少可以模拟一个浅层网络（通过恒等映射）

➤残酷现实——网络退化 (Degradation)

- 实验表明，当“普通”网络深度超过一定阈值（如20层），继续加深会导致训练误差和测试误差双双上升



问题不在于过拟合，而在于优化。深层网络极难训练

根本困境：深度与性能的悖论

- 在深度学习的早期探索中，一个普遍的信念是：更深的网络，因其拥有更多的参数和更复杂的非线性变换，理应具备更强的函数拟合能力，从而更好的性能
- 实践揭示了一个令人困惑的反直觉现象——网络退化 (Degradation Problem)
- 一个常见的误解，将网络退化问题归因于网络的表示能力 (Representation)，认为更深的网络反而无法表达那些由浅层网络已经完美实现的功能
 - 这种解释在逻辑上是站不住脚。一个56层网络，其函数空间应包含一个20层网络函数空间
 - 让前20层学习与浅层网络完全相同的参数，而新增的36层学习恒等映射 (Identity Mapping)。理想情况下，深层网络的性能至少应该等同于浅层网络！
 - 也就是说，如果一个 N 层网络已达最优，那么一个 $N+k$ 层的网络，至少应该能通过让新增的 k 层学习 恒等映射 (Identity Mapping, 即 $H(x) = x$) 来达到同样的效果

为何深层网络难以优化？

➤ 因此，问题的根源不在于表示，而在于优化！

➤ 让一堆非线性层（如 Conv -> BN -> ReLU）去精确拟合一个 $H(x) = x$ 是一个非凸优化难题

➤ 权重需要被精细地调整到一个非常特殊的值组合

➤ 优化器（如SGD）很难在巨大的参数空间中找到这个精确解

➤ 尽管更优的解存在于深层网络的解空间中，但我们找不到它

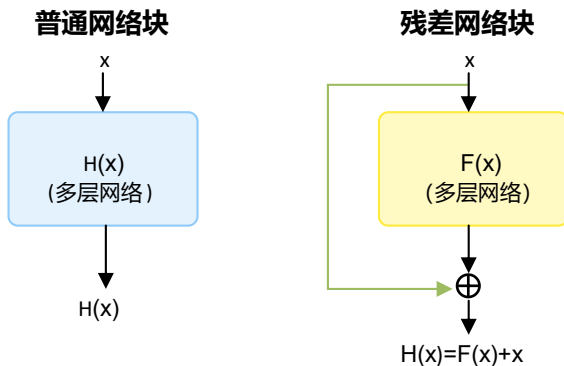
➤ 实验

➤ 任务A：训练一个网络去拟合 $H(x) = x$ 。（困难）

➤ 任务B：训练一个网络去拟合 $F(x) = 0$ 。（简单，只需将权重推向零）

➤ 我们能否重新设计网络结构，将“学习恒等映射”这个难题，转化为“学习零映射”这个简单问题？

- 将网络层的学习任务，从学习一个完整的输出，转变为学习对输入的修正
- ResNet: $H(x) = x + F(x)$
 - 信息主体 (Shortcut Connection, x):
 - 引入“跳跃连接”，将输入 x 直接与处理后的结果相加。该路径构成了信息流的主干道
 - 它默认传递了上一层输出的全部信息。这相当于网络有了一个“默认选项”：保持现状
 - 增量修正 (Residual Block, $F(x)$):
 - 残差块的学习目标被重新定义为学习残差 $F(x)$ 。最终的输出由 $H(x) = x + F(x)$ 构成
 - “在保留主体信息 x 的基础上，我需要进行什么样的增量调整，才能使结果更好？”



两种核心的残差块实现

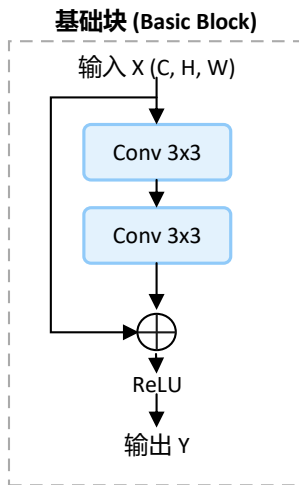
➤ 基础残差块 (Basic Block) - 用于ResNet-18/34

➤ Conv(3x3) -> BN -> ReLU -> Conv(3x3) -> BN

➤ 跳跃连接:

➤ 情况A (维度匹配): 直接相加 (Identity Shortcut)。

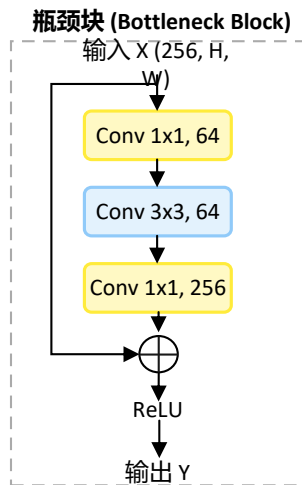
➤ 情况B (维度不匹配): 使用带步幅的1x1卷积调整维度 (Projection Shortcut)



➤ 瓶颈残差块 (Bottleneck Block) - 用于ResNet-50/101/152

➤ Conv(1x1, 降维) -> Conv(3x3) -> Conv(1x1, 升维)

➤ 目的: 在保持感受野的同时, 大幅减少参数量和计算量, 使更深的网络成为可能

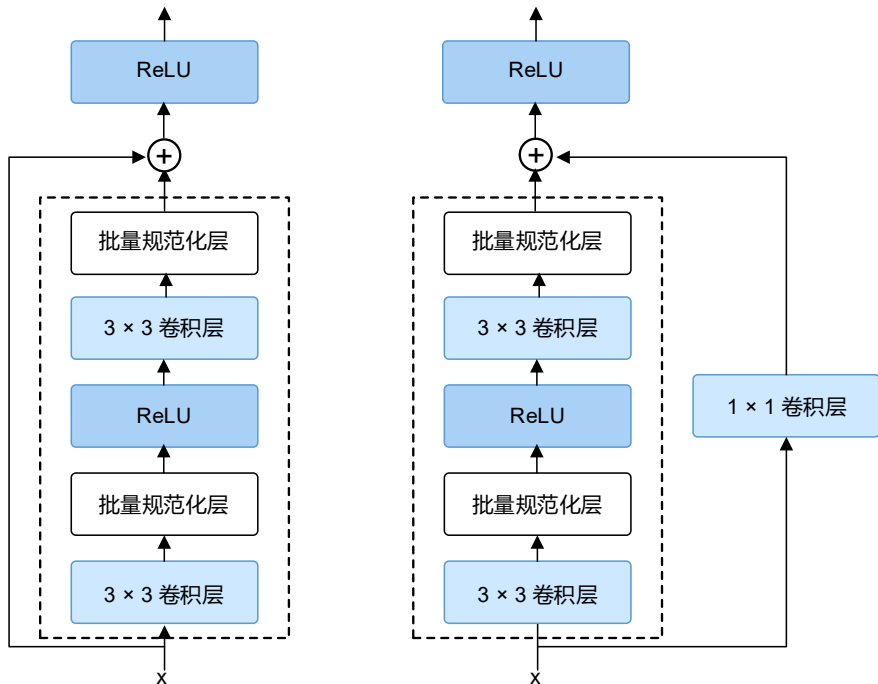


从残差块到完整的ResNet

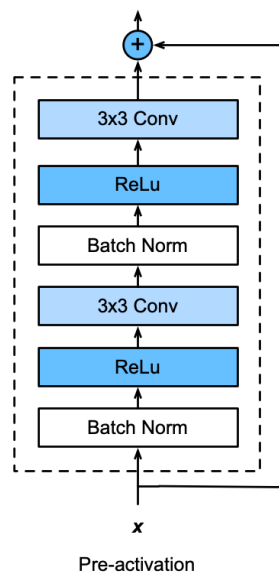
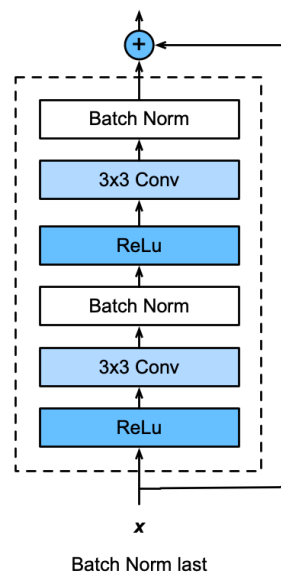
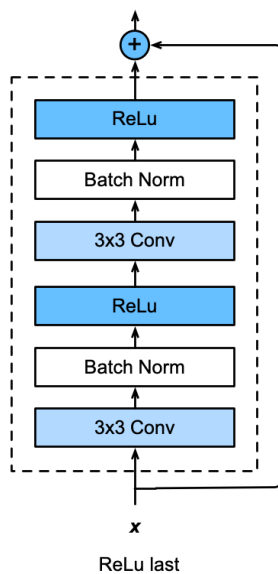
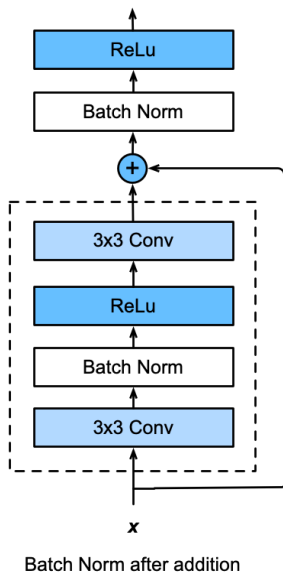
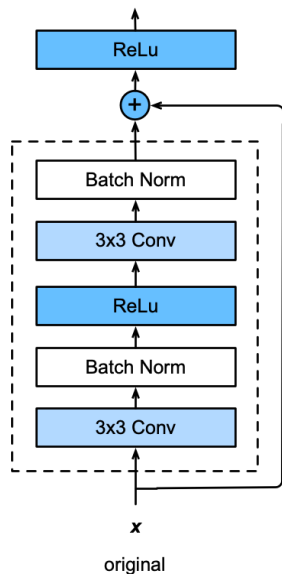
- ResNet架构由几个阶段 (Stage) 组成，每个阶段包含多个残差块
 - (页面中央为ResNet-34的宏观架构图)
- 输入图像 (224x224x3)
 - -> Conv1 (7x7, 64, stride=2) -> MaxPool (3x3, stride=2)
 - > Stage 2 (conv2_x): 3个残差块, 64通道, 尺寸56x56
 - > Stage 3 (conv3_x): 4个残差块, 128通道, 尺寸28x28 (首个块stride=2)
 - > Stage 4 (conv4_x): 6个残差块, 256通道, 尺寸14x14 (首个块stride=2)
 - > Stage 5 (conv5_x): 3个残差块, 512通道, 尺寸7x7 (首个块stride=2)
 - > Global Avg Pool -> FC (1000) -> Softmax
- 关键设计模式：
 - 尺寸减半，通道加倍：当空间尺寸 (H, W) 通过步幅为2的卷积减半时，特征图的通道数 (C) 加倍。这维持了每层大致的信息容量
 - 统一步伐：在同一阶段内的所有残差块，特征图尺寸保持不变

残差块

- 如果想改变通道数，引入一个额外的 1×1 卷积层将输入变换成需要的形状后再做相加运算



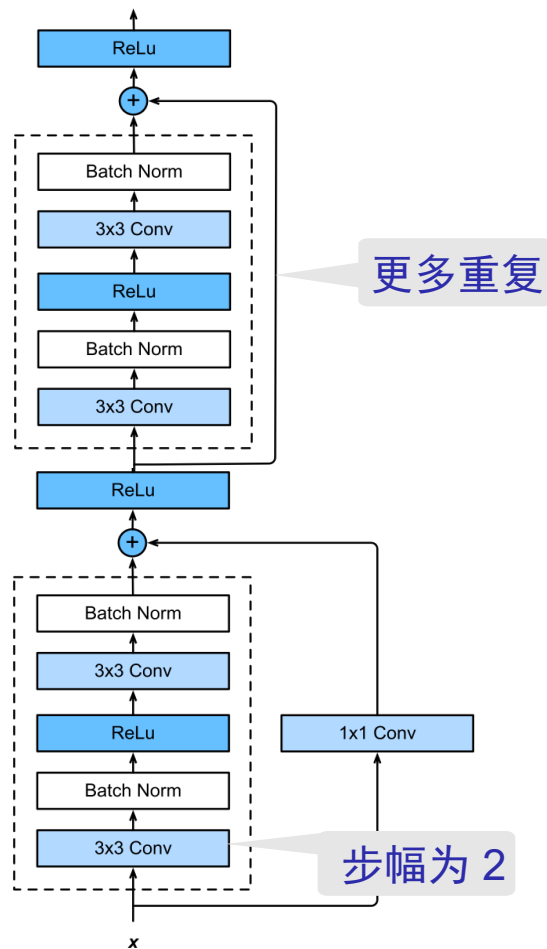
不同的残差块



尝试每一个排列

残差块

- 每个模块进行降采样（步幅 = 2）
- 强制实现每个模块中一些非常见，非线性的函数（通过1x1卷积）
- 堆积在块中



残差网络(ResNet)

➤ 相同块结构

➤ 与VGG或GoogleNet使用块结构

➤ 残差块连接可增加表现力

➤ 汇聚 / 步幅 - 减少维度

➤ 批量归一化 - 容量控制

➤大规模训练.....

➤ Res-18 (18层: 卷积的层数)

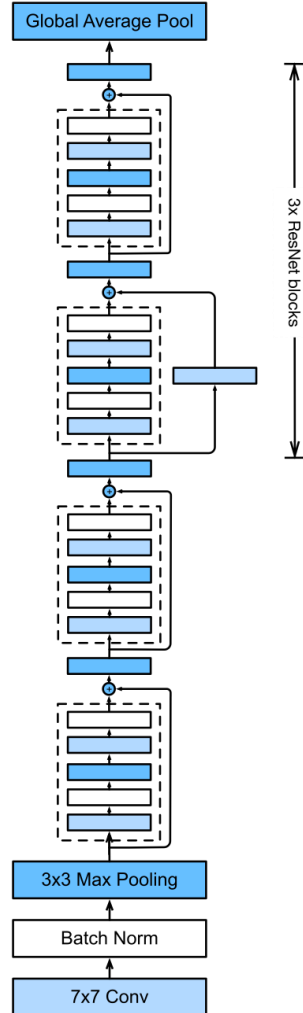
➤ 1卷积 (7*7): 1, 降采样4倍

➤ 4残差块: 4*4

➤ 第一个: 无1*1卷积, 降采样1倍

➤ 后3个: 先1*1卷积, 再普通, 降采样2倍

➤ 1全连接: 1



ResNet为何如此有效？

➤ 优化曲面平滑化 (Smoother Optimization Landscape)

- 残差连接通过提供“直连通道”，极大地简化了损失函数的结构
- 类比：将崎岖、充满局部极小值的山路，变成了平坦、宽阔的高速公路。优化器可以采用更大的学习率，更快、更稳定地收敛到优质解

➤ 无障碍的梯度流 (Unobstructed Gradient Flow)

- 在反向传播中，梯度可以直接通过跳跃连接从深层传到浅层

$$\frac{\partial L}{\partial x_l} = \frac{\partial L}{\partial x_{l+1}} \frac{\partial x_{l+1}}{\partial x_l} = \frac{\partial L}{\partial x_{l+1}} \left(1 + \frac{\partial F(x_l)}{\partial x_l} \right)$$

- +1 项确保了梯度至少可以无衰减地回传，有效缓解了梯度消失问题，使得数百甚至上千层的网络训练成为可能

➤ 隐式集成学习 (Implicit Ensemble)

- 从另一个角度看，ResNet可以被视为许多不同深度网络的隐式集成。数据可以通过不同的路径（经过或绕过某些残差块）在网络中传播

ResNet——深度学习的里程碑

➤核心贡献：

- 解决了网络退化问题，证明了深度是提升性能的关键，前提是网络结构能够被有效优化
- 使训练数百乃至上千层的网络成为现实，将深度学习带入了“超深”时代
- 残差块成为标准组件，其“跳跃连接”思想被后来的SOTA架构（如DenseNet, U-Net, Transformer）广泛借鉴和发扬

➤结论

- ResNet不仅是一个特定的网络架构，它提供了一种强大的设计范式——通过构建信息和梯度的“高速公路”，来释放深度网络的巨大潜力。它从根本上改变了我们对网络设计和优化的思考方式

更多模型

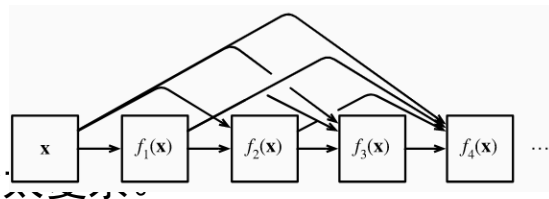
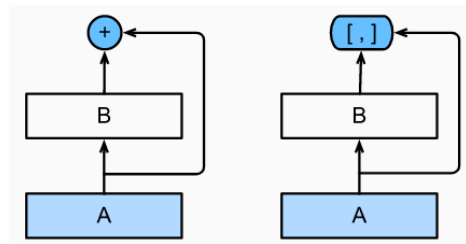
稠密连接网络 (DenseNet)

- DenseNet (Huang et al., 2016)
- ResNet结合x和f(x)
- DenseNet使用更高阶'泰勒系列'扩展

$$\begin{aligned}x_{i+1} &= [x_i, f_i(x_i)] \\x_1 &= x \\x_2 &= [x, f_1(x)] \\x_2 &= [x, f_1(x), f_2([x, f_1(x)])]\end{aligned}$$

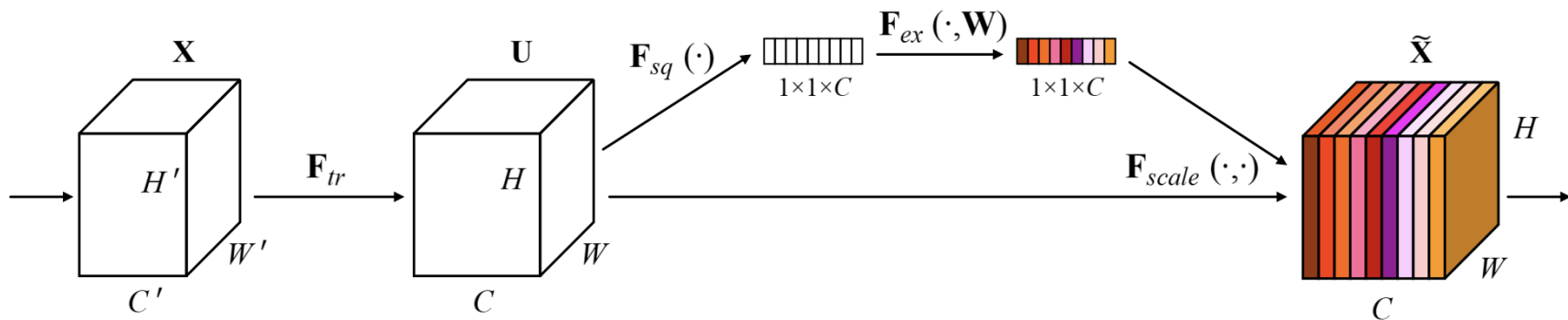
- 稠密网络主要由2部分构成

- 稠密块 (dense block)
- 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



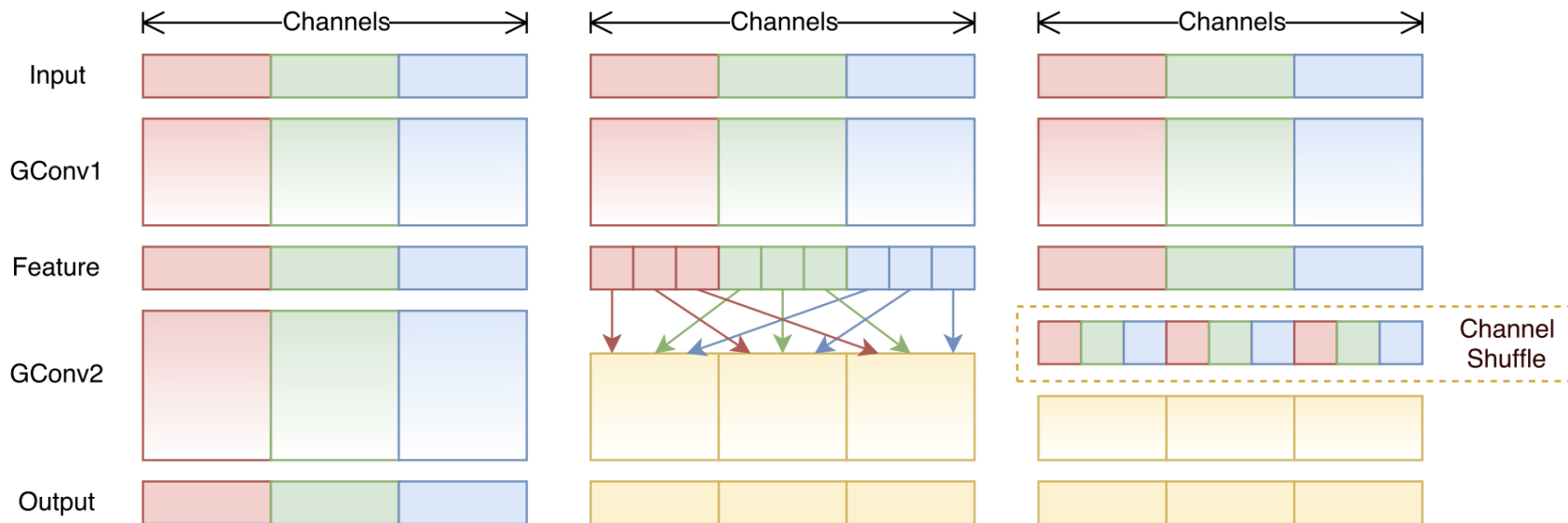
Squeeze-Excite Net

- Squeeze-Excite Net (Hu et al., 2017)
- 学习每个通道的全局加权函数
- 允许在图像的不同位置的像素之间快速传输信息



ShuffleNet (Zhang et al., 2018)

- ResNeXt 将卷积层分成不同通道
- ShuffleNet 通过分组混合不同通道（对移动设备非常有效）



总结

➤ Inception

- 卷积的不均匀混合（不同深度）
- 批量归一正则化

➤ ResNet

- 泰勒扩展式
- 残差网络（ResNext）分解卷积

➤ Zoo

稠密连接网络（DenseNet）, ShuffleNet, 可分解卷积层, ...