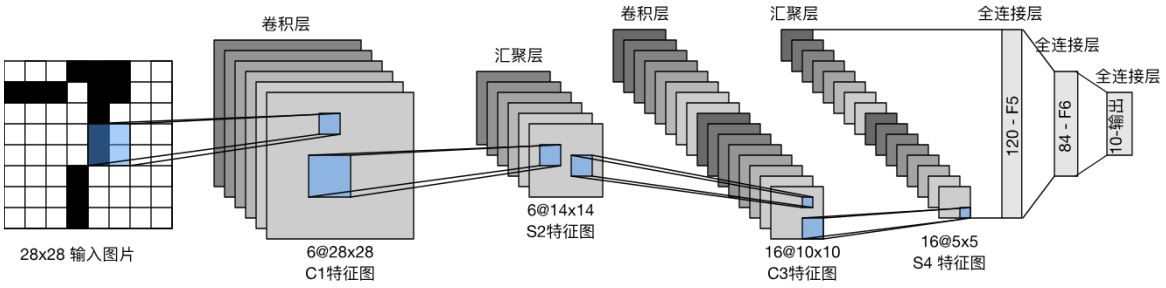
笔记：卷积神经网络

第六章 卷积神经网络

经典卷积神经网络 LeNet



1.卷积层代替全连接层：1.可以在图像中保留空间结构；2.模型更简洁，所需参数更少

2.LeCun 1989：识别图像中的手写数字，第一篇通过反向传播成功训练卷积神经网络论文

3.LeNet（LeNet-5）由两个部分组成：

卷积编码器：由两个卷积层组成;

全连接层密集块：由三个全连接层组成

每个卷积块中的基本单元是一个卷积层、一个sigmoid激活函数和平均汇聚层

第七章 现代卷积神经网络

7.1 深度卷积神经网络 AlexNet

1.更深更大的LeNet

2.主要改进：dropout（模型控制，正则）、relu（梯度更大）、maxpooling（输出大，梯度更大）、数据增强（模拟改变）

3.past：image-人工特征提取（重点）-SVM（标准机器学习模型：后两部份独立

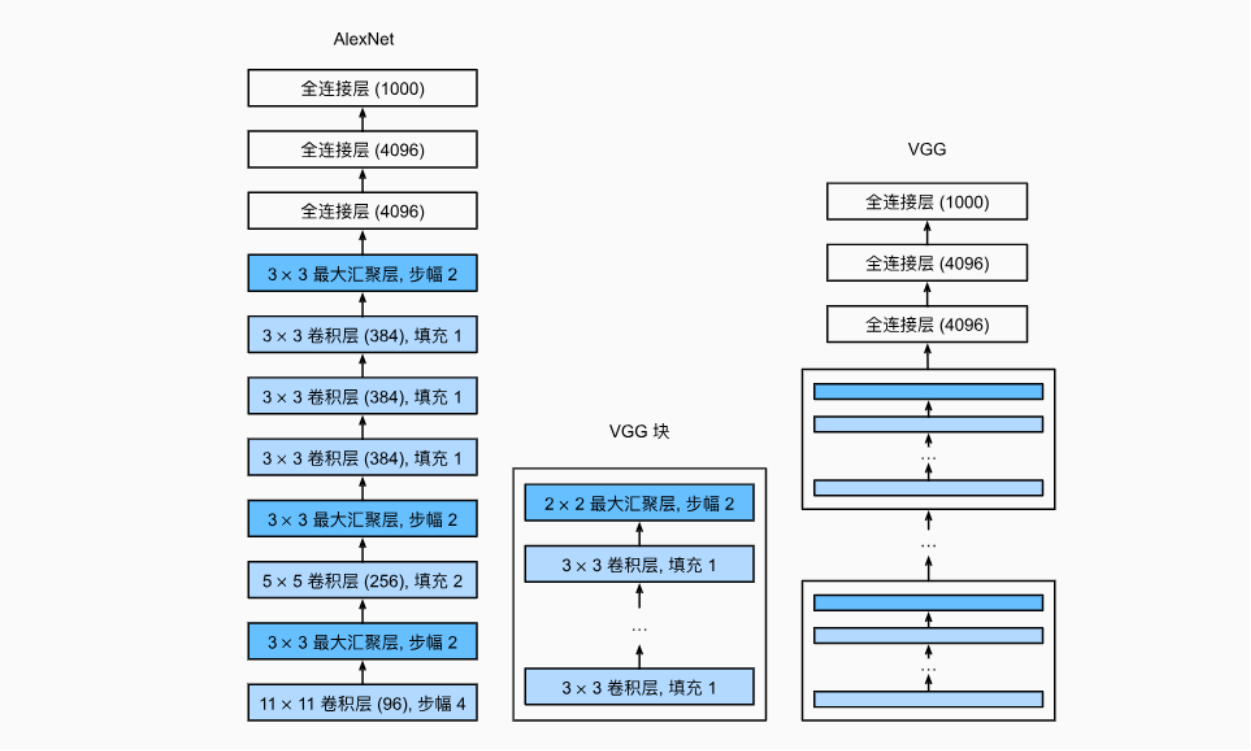
now：image-通过CNN学习特征-Softmax回归：后两部份一起，一起训练

优势：1.CNN相对简单，不需要太多cv知识；2.一起训练更加高效，端到端（原始输入直接到结果（分类等））

7.2 使用块的网络 VGG

1.更大更深的AlexNet（重复的VGG块）

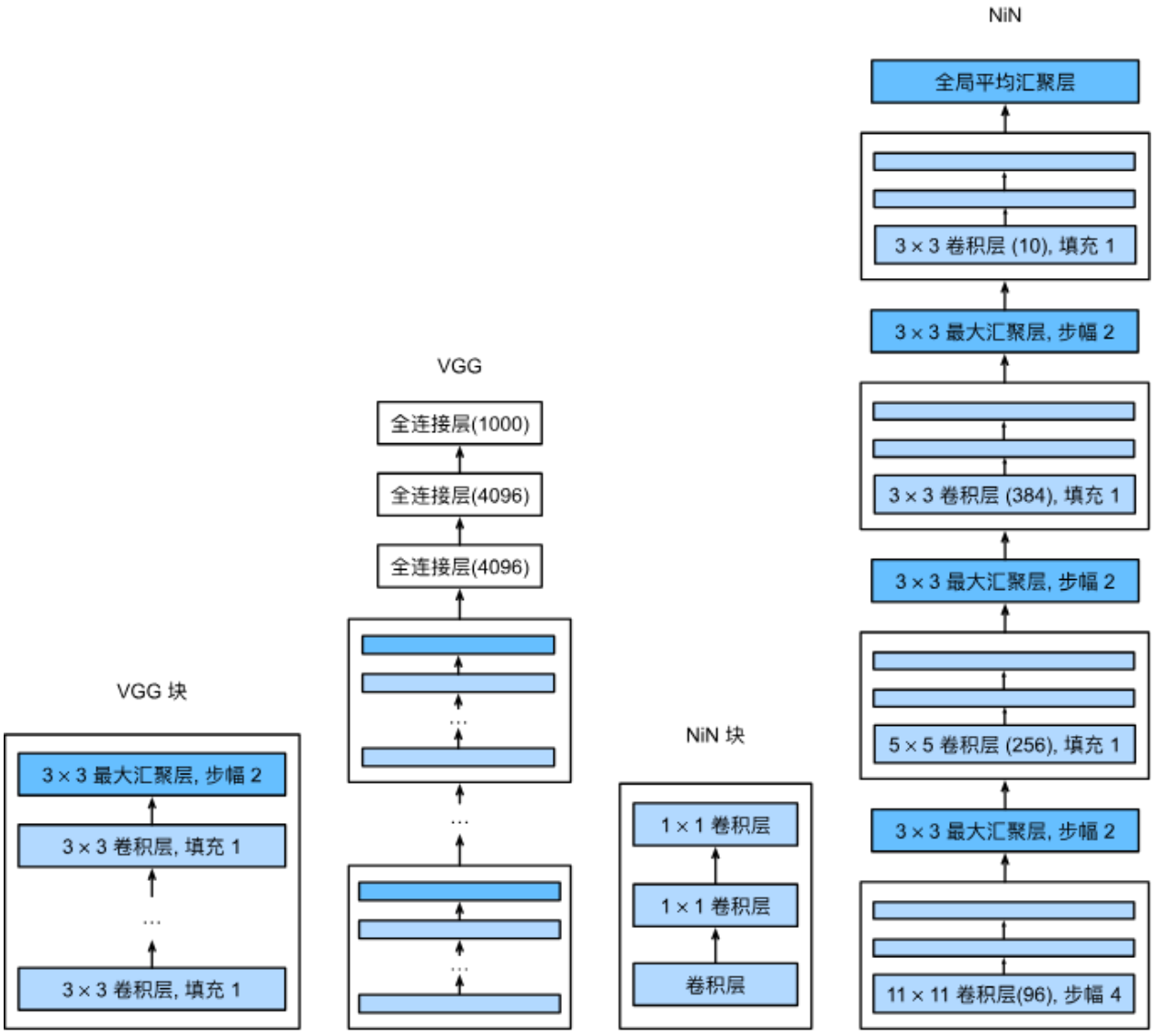
2.深且窄的卷积比浅且宽的更有效



7.3 网络中的网络 NiN

1.前述网络卷积层后第一个全连接层的参数太多了，内存+过拟合

2.用1x1kernal，stride=1，padding=0起到全连接层作用。在每个像素位置应用一个全连接层



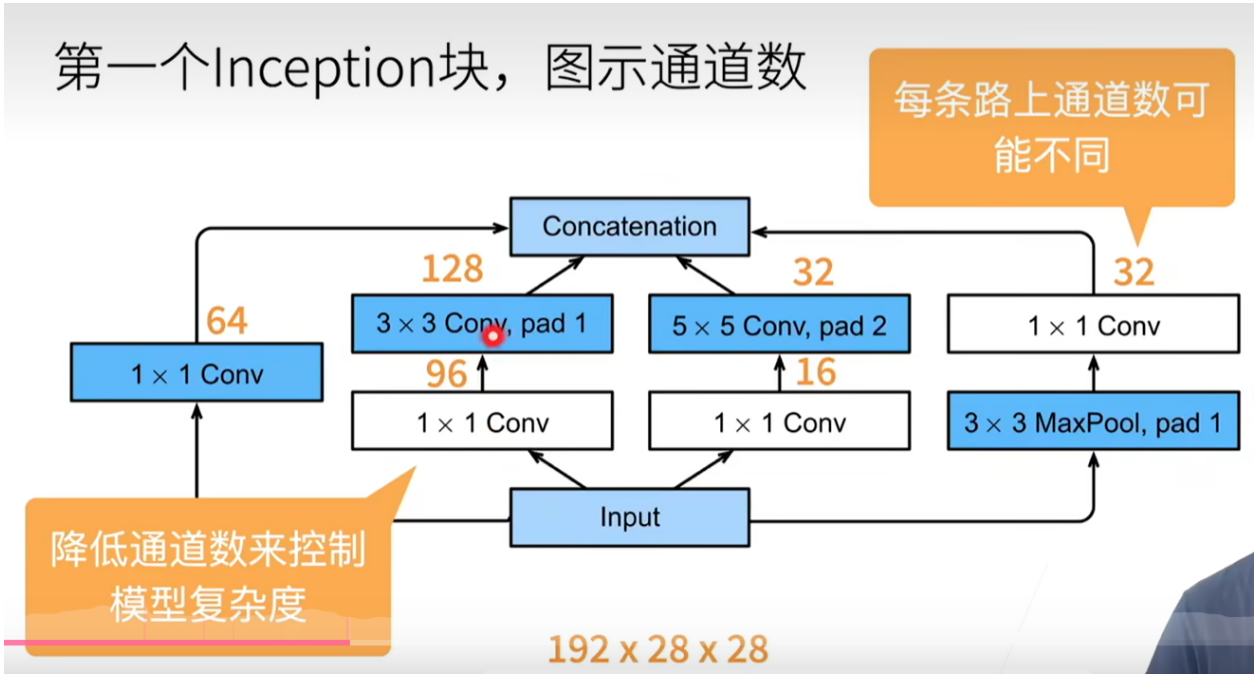
3.全局平均池化层：输入通道是类别数，每一个通道算出平均值，再用softmax作为概率（用卷积层的通道来输出类别预测）

4.卷积层输入输出-四维张量：样本、通道、高度、宽度

全连接层输入输出-二维张量：样本、特征

——>空间维度中的每个像素视为单个样本，通道维度视为不同特征

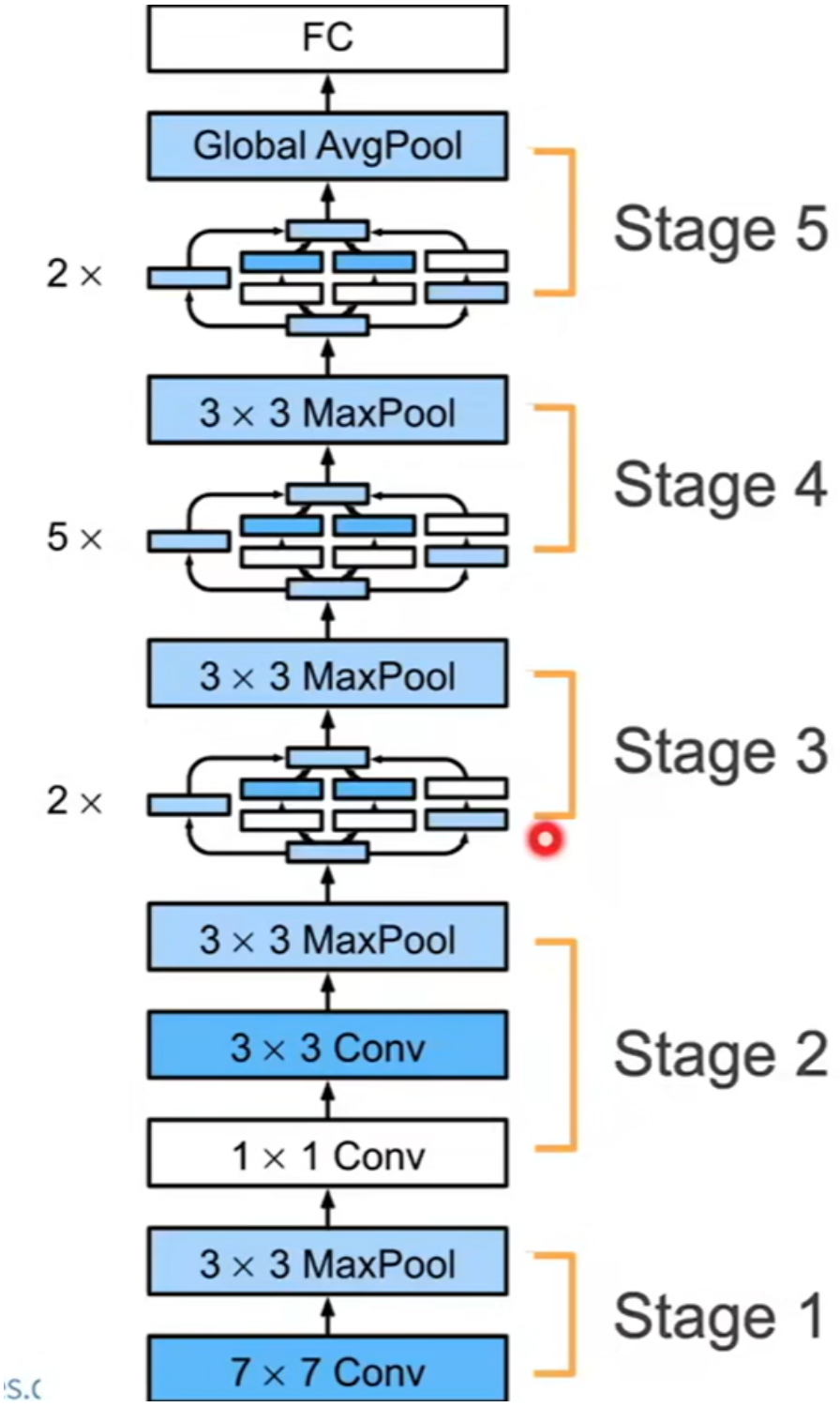
7.4 含并行连接的网络 GoogLeNet



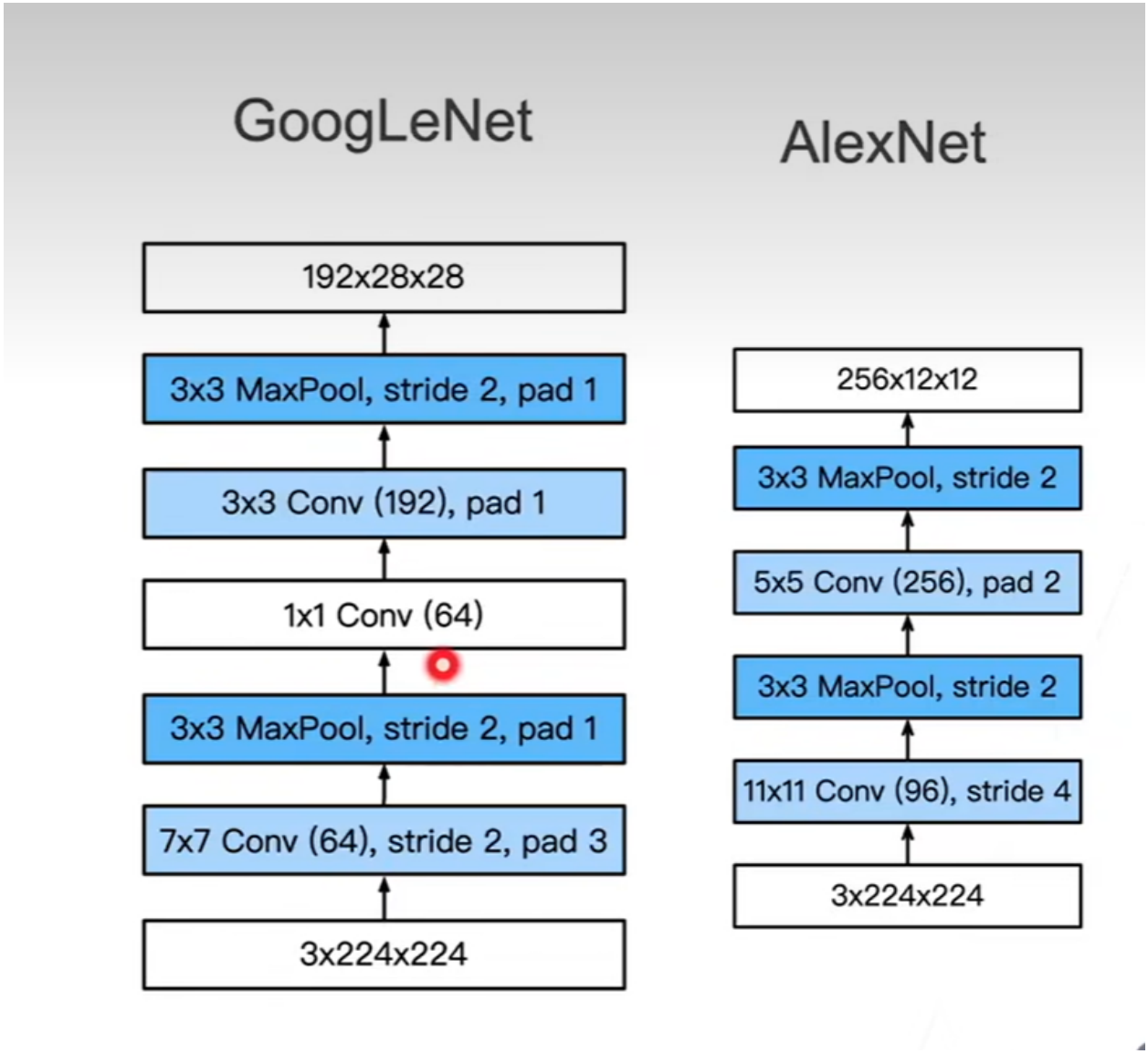
1.模型复杂度-参数个数-输入通道x输出通道xkernal大小

2.白色1x1：改变通道数 蓝色：抽取空间信息

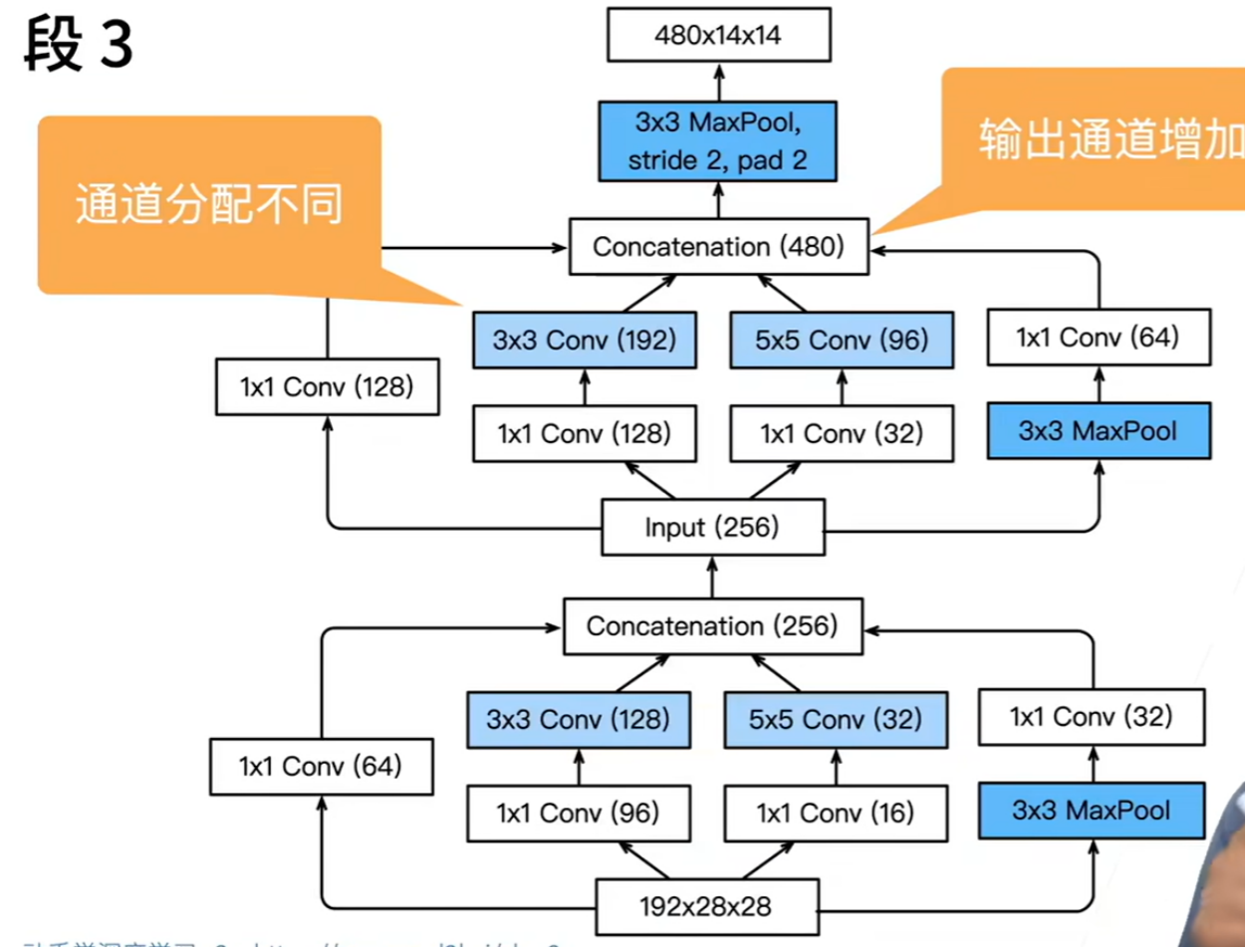
3.inception块比单3x3、5x5卷积层有更少的参数个数和计算复杂度



4.每个stage代表高宽减半，inception不改变高宽，只改变通道数



5.段1、2：比AlexNet小的kernal，保留更大的高宽，方便后续更深的网络，同时增加通道数



6.inception块用4条有不同超参数的卷积层和池化层的路来抽取不一样的信息，解决了多大卷积核是合适的这个问题，不同滤波器可以识别不同范围的图像细节

7.注意：

- 卷积层：通常用来提取特征（识别相邻元素间作用力），并且通常不希望改变图像的大小（老的也有就让他变小的），所以通常会取padding=（k-1）/2 （这个padding值是一边加多少）。同时也可以通过kernal的数量来改变通道数（主要是增加）
- 1x1卷积层：通常用于调整网络层的通道数量和控制模型复杂度，可增加可减少
- 汇聚层：通常用于降低卷积层对位置的敏感性和对空间降采样表示的敏感性，通常会取stride=2来使得高宽减半(默认情况下，深度学习框架中的步幅与汇聚窗口的大小相，所以直接MaxPoo2d(2))（在inception中也可以起到提取信息的作用）
- flatten层：用于展平（直接列成像素），用于后面的全连接层
- 线性层（全连接）：用于把那些值的数量最后变成分类种类的数量

7.5 批量规范化

1.问题：损失在最后计算，在反向传播中，后面的层梯度较大，因此后面的层训练较快，底部的层训练较慢。而底部往往训练底层的东西（线段之类），导致底部层一变化所有都得变，最后的那些层需要重新学习多次，导致收敛变慢--->学习底部层时能否避免改变顶部层

也即加速深层网络的收敛

- 1.数据预处理（标准化）可以将参数的量级进行统一
- 2.控制中间层变量的变化，控制变量分布的偏移（不同层如果参数可变范围不同，学习率也应该对应不同，所以在适用一个学习率的情况下，就需要控制）
- 3.更深的网络很复杂，容易过拟合，需要正则化

  - Regularization，中文翻译过来可以称为正则化，或者是规范化。可以说是一个限制通过这种规则去规范他们再接下来的循环迭代中，不要自我膨胀

  - 欠拟合原因：1.训练样本数量少2.模型复杂度过低3.参数还未收敛就停止循环

  - 欠拟合的解决办法：1.增加样本数量2.增加模型参数，提高模型复杂度3.增加循环次数4.查看是否是学习率过高导致模型无法收敛

  - 过拟合原因：1.数据噪声太大2.特征太多3.模型太复杂

  - 过拟合的解决办法：1.清洗数据2.减少模型参数，降低模型复杂度3.增加惩罚因子（正则化），保留所有的特征，但是减少参数的大小（magnitude）。

  2.批量归一化：

  要使得分布在不同层之间尽量不变化->固定小批量里面的均值和方差

  $\mu_B = \frac{1}{|B|}\sum_{i\in B} x_i$

  $\sigma_B^2 = \frac{1}{|B|}\sum_{i\in B}(x_i-\mu_B)^2 + \epsilon$

  $\Rightarrow x_{i+1} = \gamma\frac{x_i-\mu_B}{\sigma_B} + \beta$

  其中γ和β是可以学习的参数（比例系数和比例偏移or拉伸参数和偏移参数），因为如果变成均值为0，方差为1的分布不那么合适的话可以学习新的均值（后者）和方差（前者）-->基于批量统计的标准化-->批量规范化

  注：$+\epsilon$以保证分母永远不会等于零

  3.全连接层：作用在特征维，仿射变化和激活函数之间

- 线性模型：输入包含d个特征：$\hat{y} = w_1x_1+\ldots+w_dx_d+b$
- 将所有特征放在一个向量$\mathbf{x}\in\mathbb{R}^d$里,将所有权重放在向量$\mathbf{w}\in\mathbb{R}^d$里：$\hat{y} = \mathbf{w}^T\mathbf{x}+b$
- 这里的向量x是单个数据的样本特征，所以可以用矩阵$\mathbf{X}\in\mathbb{R}^{n\times d}$引用n个样本，每一行是一个样本，每一列是一种特征：$\hat{y} = \mathbf{X}\mathbf{w}+b$

- 也即对每一个特征。计算标量的均值和方差，然后进行变换

  4.卷积层：作用在通道维，卷积层和激活函数之间

- 对于卷积层的输入，假设批量大小为1，输入一个图片，每一个像素就是一个样本，所以样本的数量就是高乘以宽，而多个通道就是每个样本的不同特征，因此通道维可以看作是全连接层的特征维

  5.最初的想法是控制分布，后来发现可能是将$\hat{\mu}_B$作为随机偏移，$\hat{\sigma}_B$作为随机缩放，也即可以理解为在每个小批量加入噪音来控制模型复杂度，因此没必要和dropout混合使用

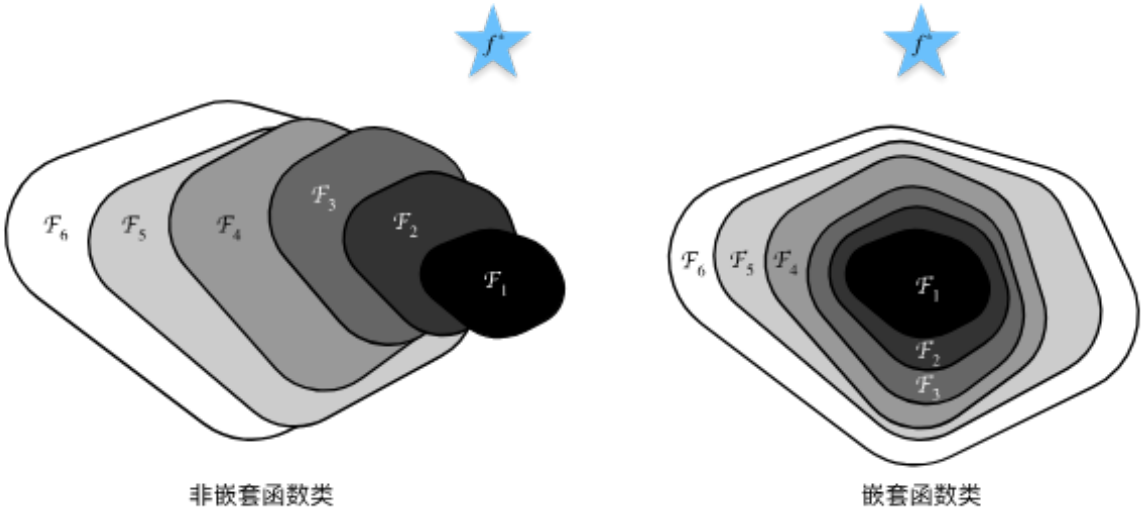
  6.批量归一化可以加速收敛速度（学习率可以调的更大），但一般不改变模型精度

  7.代码部分：

```
#在定义batch_norm函数中
#moving_是全局变量，momentum用于更新
# 更新移动平均的均值和方差
        moving_mean = momentum * moving_mean + (1.0 - momentum) * mean
        moving_var = momentum * moving_var + (1.0 - momentum) * var
```

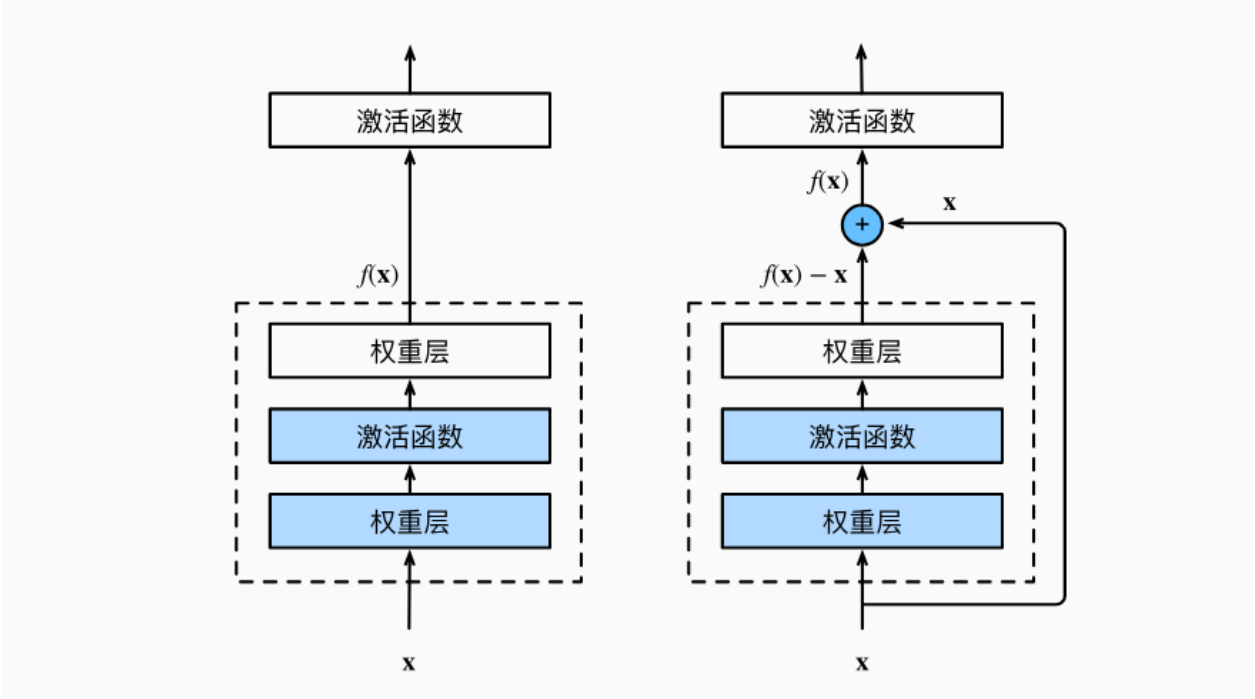
BN在训练模式：每个小批量的均值和方差；在预测模式：整个数据集的均值和方差

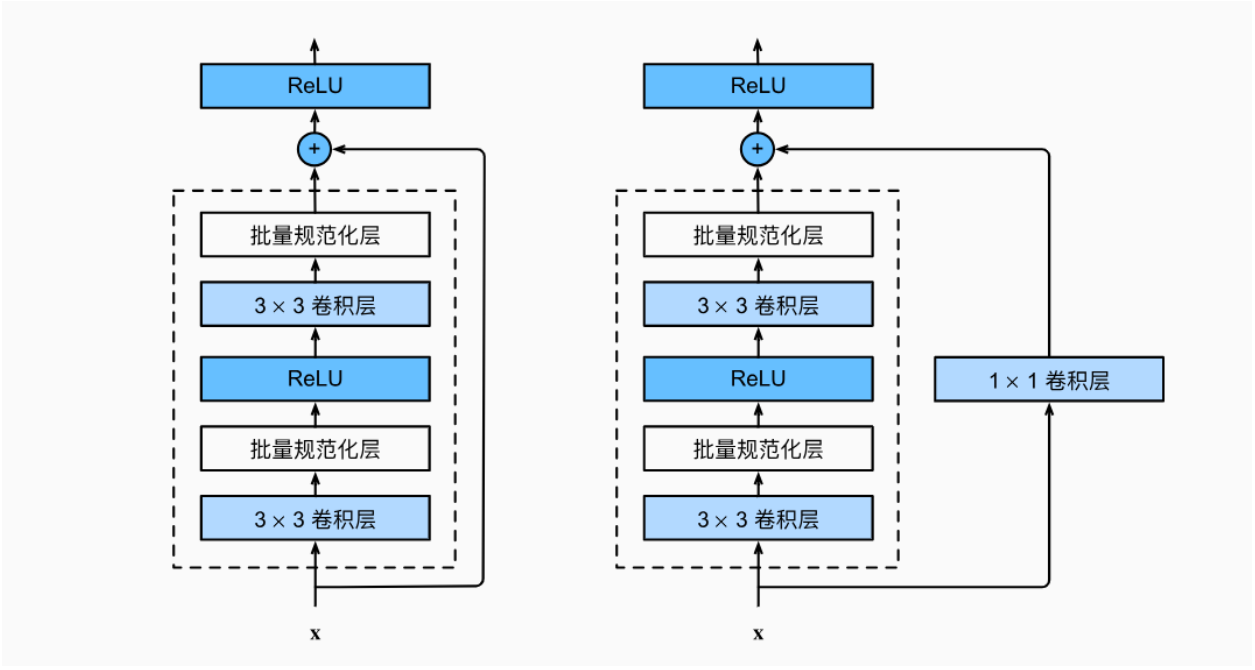
7.6 残差网络 ResNet

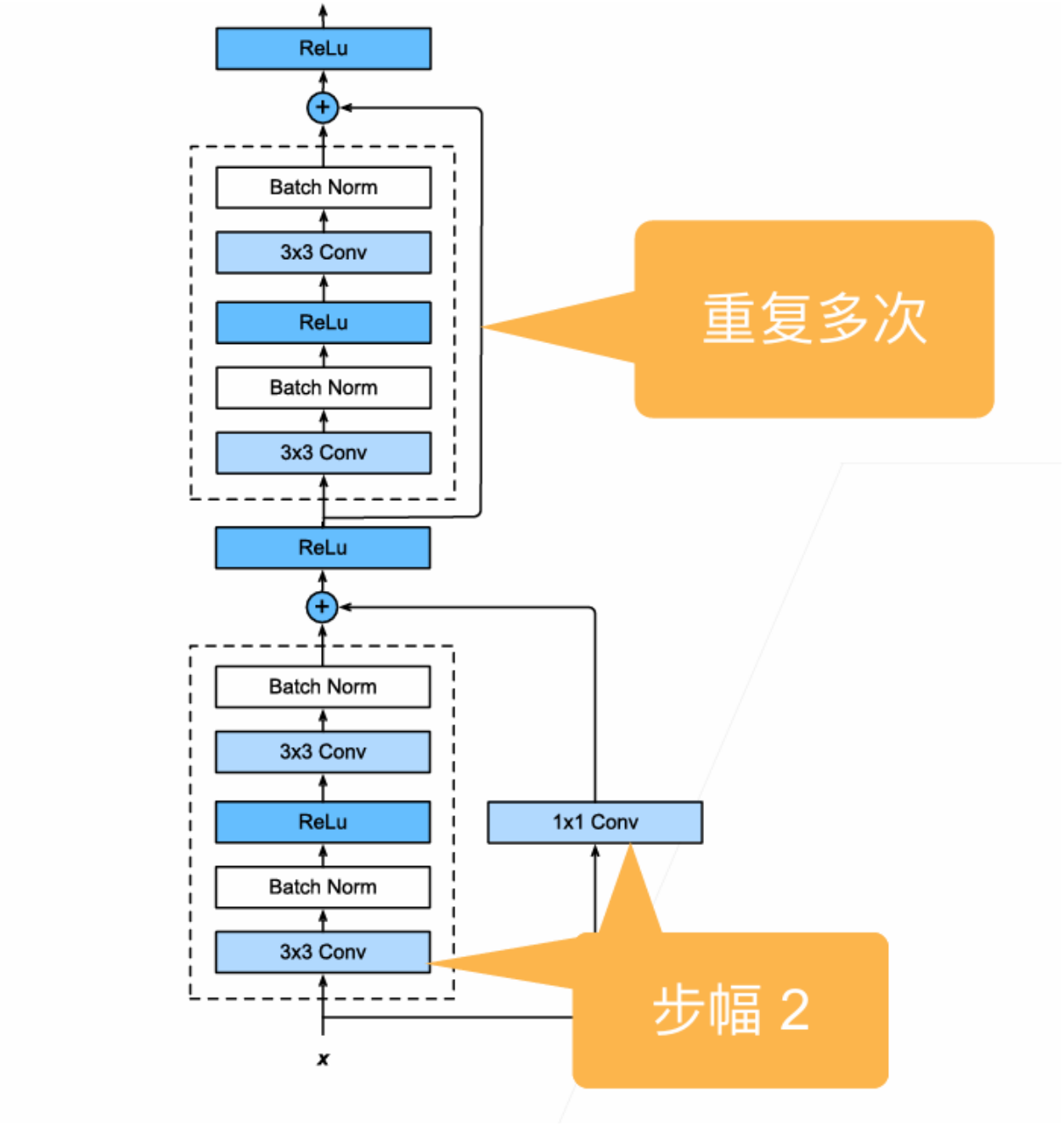


1.后面的层覆盖范围包含之前的层覆盖范围，防止走偏离目标越来越远->使得很深的网络更加容易训练

也即残差快（residual block)核心：每一个附加层都应该更容易地包含原始函数作为其元素之一


激活函数
$f(\mathbf{x})$
权重层
激活函数
权重层
$\mathbf{x}$
激活函数
$f(\mathbf{x})$
$\mathbf{x}$
$f(\mathbf{x}) - \mathbf{x}$
权重层
激活函数
权重层
$\mathbf{x}$


ReLU
批量规范化层
3 × 3 卷积层
ReLU
批量规范化层
3 × 3 卷积层
x
ReLU
批量规范化层
3 × 3 卷积层
ReLU
批量规范化层
3 × 3 卷积层
1 × 1 卷积层
x



2.通常在第一块步幅为2来使高宽减半，同时用1x1kernal来增加通道数，后接多个高宽不变的ResNet块

3.总体架构类似VGG、GoogleNet，替换成ResNet块

4.学习嵌套函数是训练神经网络的理想情况。在深层神经网络中，学习另一层作为恒等映射较容易（尽管这是一个极端情况）

5.残差映射可以更容易地学习同一函数，例如将权重层中的参数近似为零