1 实例探讨

讲到的经典 CNN 模型包括：

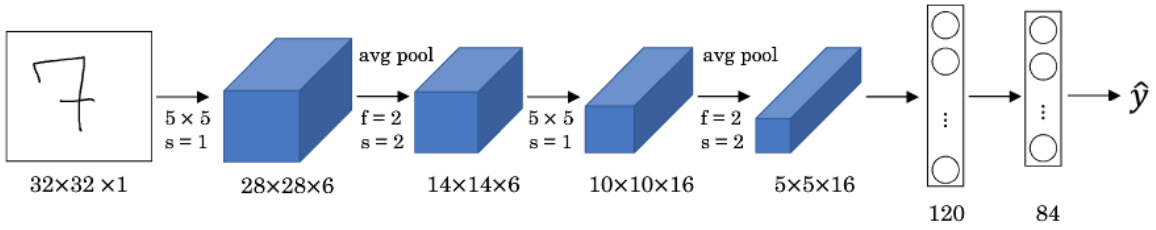
- LeNet-5
- AlexNet
- VGG

此外还有 ResNet（Residual Network，残差网络），以及 Inception Neural Network。

1.1 经典的卷积网络

1.1.1 LeNet-5

LeNet-5 是 LeCun 等人 1998 年在论文 Gradient-based learning applied to document recognition (https://ieeexplore.ieee.org/document/726791) 中提出的卷积网络。其结构如下图：

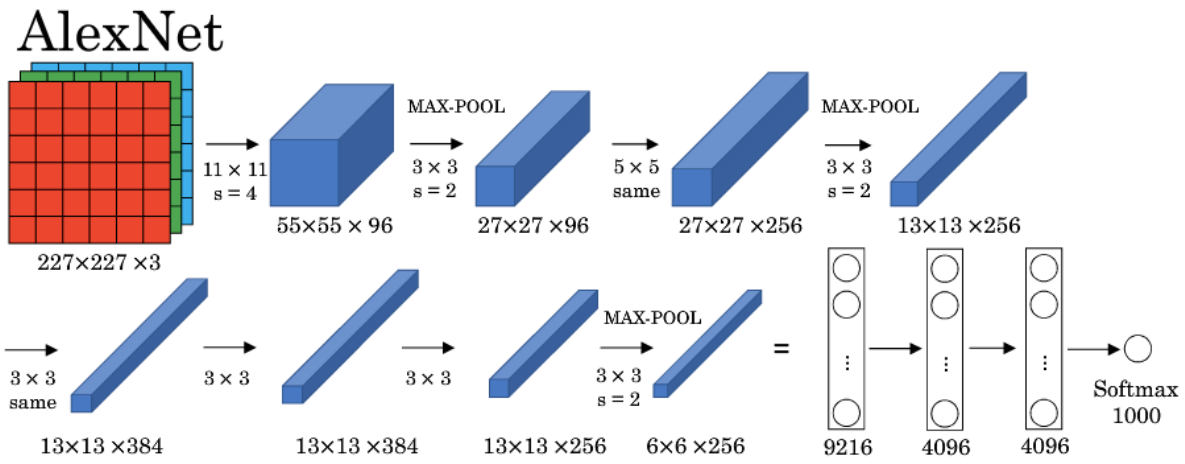


特点：

- LeNet-5 针对灰度图像而训练，因此输入图片的通道数为 1。
- 该模型总共包含了约 6 万个参数，远少于标准神经网络所需。
- 典型的 LeNet-5 结构包含卷积层（CONV layer），池化层（POOL layer）和全连接层（FC layer），排列顺序一般为 CONV layer->POOL layer->CONV layer->POOL layer->FC layer->FC layer->OUTPUT layer。一个或多个卷积层后面跟着一个池化层的模式至今仍十分常用。
- 当 LeNet-5 模型被提出时，其池化层使用的是平均池化，而且各层激活函数一般选用 Sigmoid 和 tanh。现在，我们可以根据需要，做出改进，使用最大池化并选用 ReLU 作为激活函数。
- 随着深度的增加，n_H、n_W 的值在不断减小，n_c 却在不断增加。其中的 Conv-Pool-Conv-Pool-FC-FC-Output 是现在用到的卷积网络中的一种很常见的结构。

1.1.2 AlexNet

AlexNet 是 Krizhevsky 等人 2012 年在论文 ImageNet classification with deep convolutional neural networks (http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf) 中提出的卷积网络。其结构如下图：

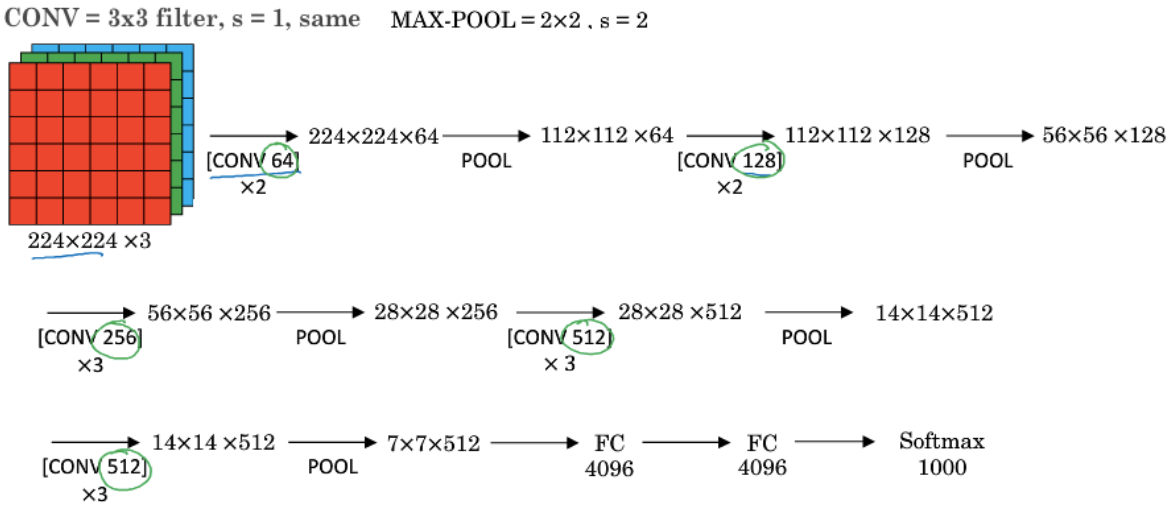


特点：

- AlexNet 模型与 LeNet-5 模型类似，但是更复杂，包含约 6000 万个参数。另外，AlexNet 模型使用了 ReLU 函数。
- 当用于训练图像和数据集时，AlexNet 能够处理非常相似的基本构造模块，这些模块往往包含大量的隐藏单元或数据。

1.1.3 VGG

VGG-16 是 Simonyan 和 Zisserman 2015 年在论文 Very deep convolutional networks for large-scale image recognition (https://arxiv.org/pdf/1409.1556.pdf) 中提出的卷积网络。其结构如下图：



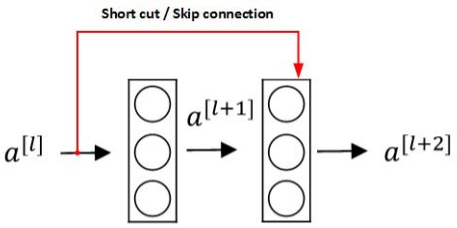
特点：

- VGG 又称 VGG-16 网络，“16”指网络中包含 16 个卷积层和全连接层。
- 超参数较少，只需要专注于构建卷积层。
- 结构不复杂且规整，在每一组卷积层进行滤波器翻倍操作。
- VGG 需要训练的特征数量巨大，包含多达约 1.38 亿个参数。

1.2 残差网络

当一个神经网络某个深度时，将会出现梯度消失（Vanishing Gradient）和梯度爆炸（Exploding Gradient）等问题。而 ResNets 能很好得解决这些问题。

ResNets 全称为**残差网络（Residual Networks）**，它是微软研究院 2015 年在论文 Deep Residual Learning for Image Recognition (https://arxiv.org/pdf/1512.03385.pdf) 中提出的卷积网络。



上图的结构被称为**残差块（Residual block）**。

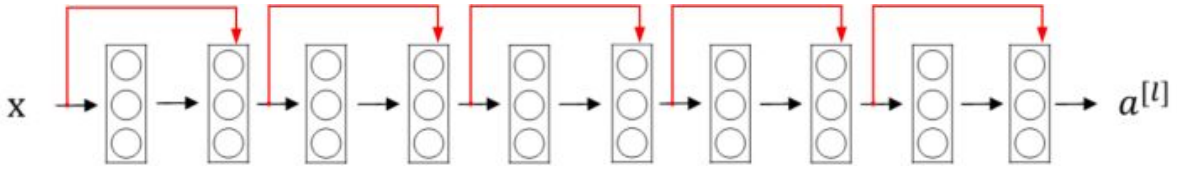
一般的前向传播的过程，也称为 “主要路径（main path）”为 $a^{[l]}$-linear-ReLU-linear-ReLU-$a^{[l+2]}$，计算过程如下：

$$\begin{aligned} z^{[l+1]} &= W^{[l+1]}a^{[l]} + b^{[l+1]} \\ a^{[l+1]} &= g(z^{[l+1]}) \\ z^{[l+2]} &= W^{[l+2]}a^{[l+1]} + b^{[l+2]} \\ a^{[l+2]} &= g(z^{[l+2]}) \end{aligned} \tag{1}$$

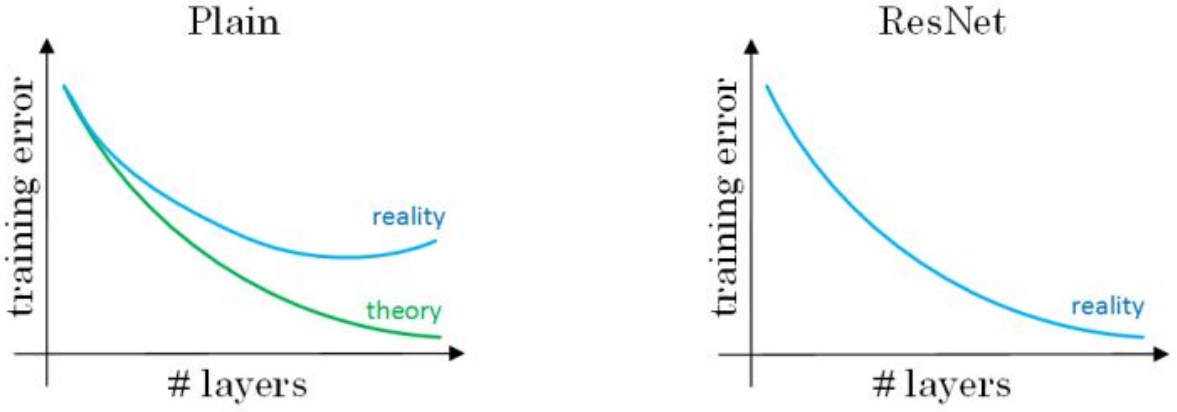
在残差网络中，通过捷径（Short cut，或者称跳远连接，Skip connections）可以将 $a^{[l]}$ 添加到第二个 ReLU 过程中，直接建立 $a^{[l]}$ 与 $a^{[l+2]}$ 之间的隔层联系。也就是最后一个方程变为：

$$a^{[l+2]} = g(z^{[l+2]} + W_s a^{[l]}) \tag{2}$$

构建一个残差网络就是将许多残差块堆积在一起，形成一个深度网络。



在理论上，随着网络深度的增加，性能应该越来越好。但实际上，对于一个普通网络，随着神经网络层数增加，训练错误会先减少，然后开始增多。但残差网络的训练效果显示，即使网络再深，其在训练集上的表现也会越来越好。

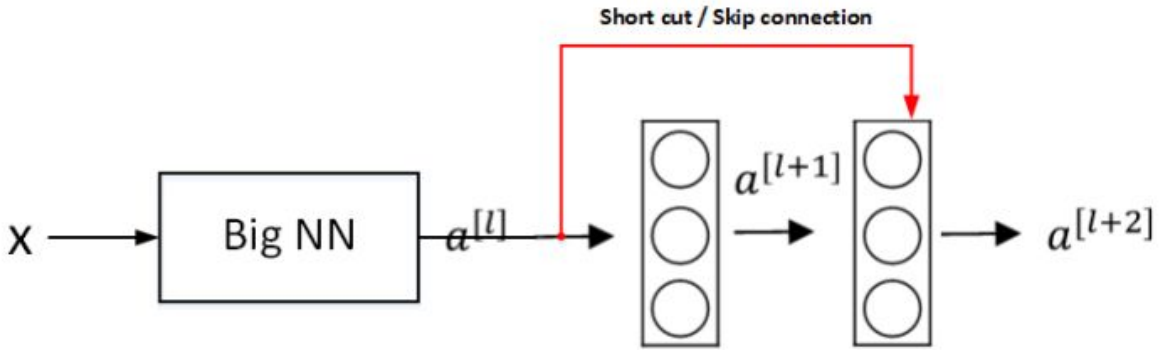


残差网络有助于解决梯度消失和梯度爆炸问题，使得在训练更深的网络的同时，又能保证良好的性能。

残差网络解释

对于一个神经网络中存在的一些**恒等函数（Identity Function）**，残差网络在不影响这个神经网络的整体性能下，使得对这些恒等函数的学习更加容易，而且很多时候还能提高整体的学习效率。

假设有一个大型神经网络，其输入为 X，输出为 $a^{[l]}$。给这个神经网络额外增加两层，输出为 $a^{[l+2]}$。将这两层看作一个具有跳远连接的残差块。为了方便说明，假设整个网络中都选用 ReLU 作为激活函数，因此输出的所有激活值都大于等于 0。



则有：

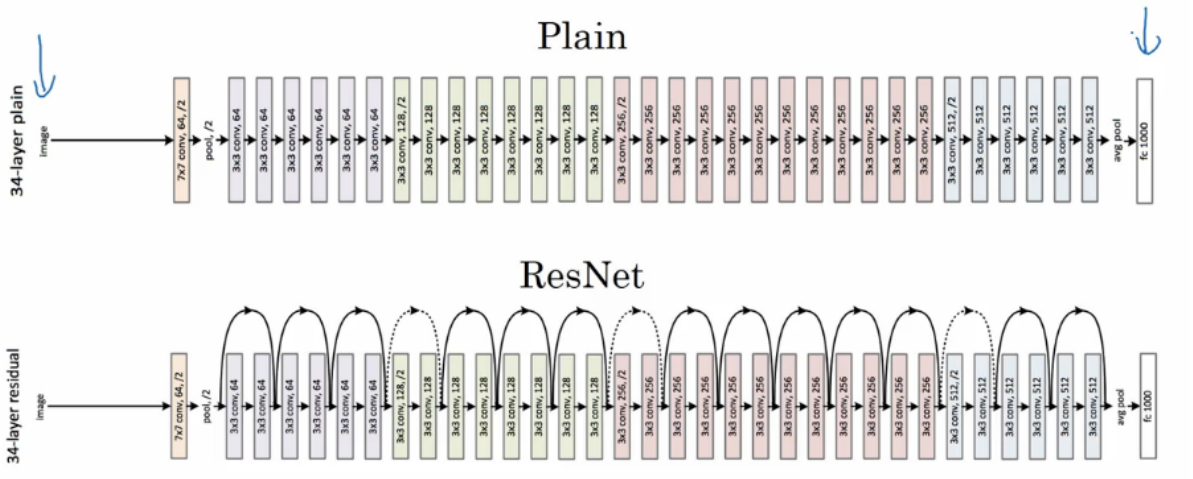
$$\begin{aligned} a^{[l+2]} &= g(z^{[l+2]} + a^{[l]}) \\ &= g(W^{[l+2]}a^{[l+1]} + b^{[l+2]} + a^{[l]}) \end{aligned} \tag{3}$$

当发生梯度消失时，$W^{[l+2]} \approx 0$，$b^{[l+2]} \approx 0$，则有：

$$a^{[l+2]} = g(a^{[l]}) = ReLU(a^{[l]}) = a^{[l]} \tag{4}$$

因此，这两层额外的残差块不会降低网络性能。而如果没有发生梯度消失时，训练得到的非线性关系会使得表现效果进一步提高。

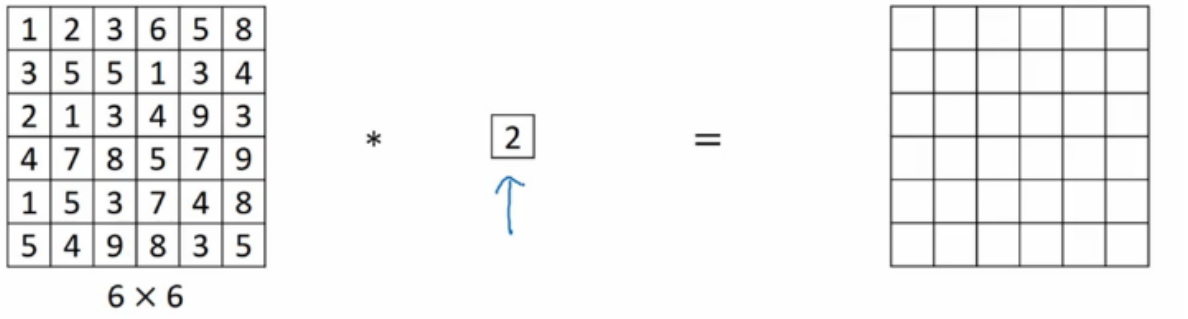
注意，如果 $a^{[l]}$ 与 $a^{[l+2]}$ 的维度不同，需要引入矩阵 W_s 与 $a^{[l]}$ 相乘，使得二者的维度相匹配。参数矩阵 W_s 既可以通过模型训练得到，也可以作为固定值，仅使 $a^{[l]}$ 截断或者补零。



上图是论文提供的 CNN 中 ResNet 的一个典型结构。卷积层通常使用 Same 卷积以保持维度相同，而不同类型层之间的连接（例如卷积层和池化层），如果维度不同，则需要引入矩阵 W_s。

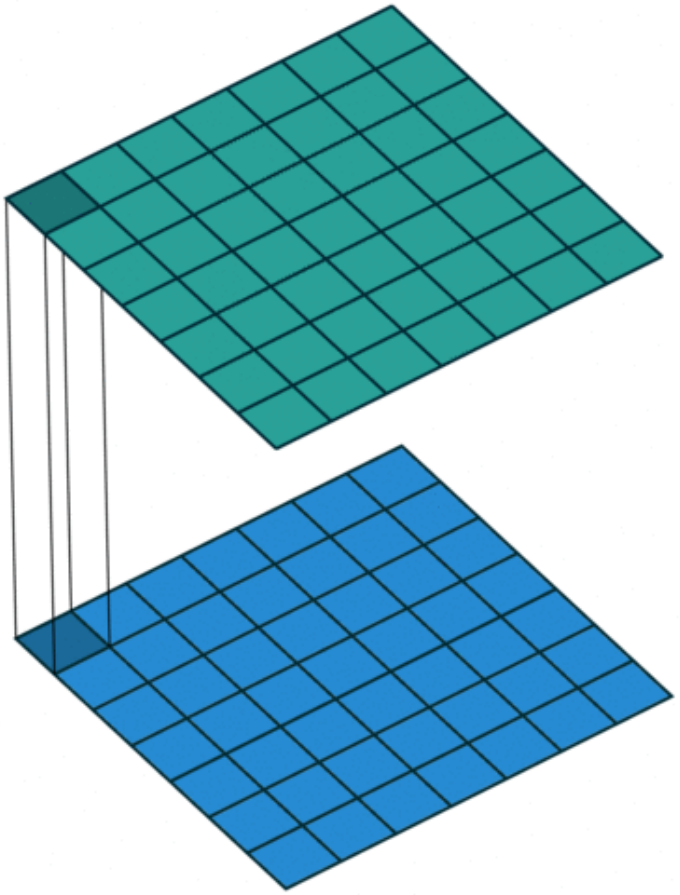
1.3 1×1 卷积

1x1 卷积（1x1 convolution，或称为 Network in Network）指滤波器的尺寸为 1。当通道数为 1 时，1x1 卷积意味着卷积操作等同于乘积操作。



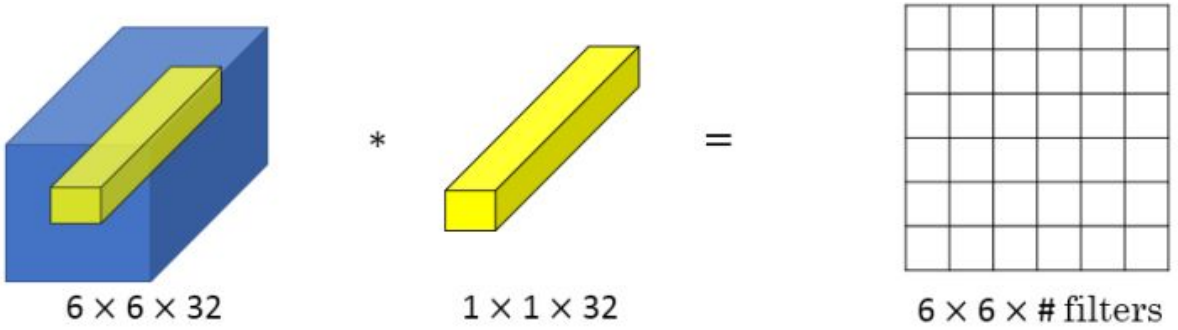
而当通道数更多时，1x1 卷积的作用实际上类似全连接层的神经网络结构，从而降低（或升高，取决于滤波器组数）数据的维度。

2013 年新加坡国立大学的林敏等人在论文 Network In NetWork (https://arxiv.org/pdf/1312.4400.pdf) 中提出了1×1 卷积核及 NIN 网络。



使用 1×1 卷积核进行卷积的过程如上图，它就是在卷积过程中采用大小为 1×1 的滤波器。如果神经网络的当前一层和下一层都只有一个信道，也就是 $n_C = 1$，那么采用 1×1 卷积核起不到什么作用的。但是当它们分别为有 m 和 n 个信道时，采用 1×1 卷积核就可以起到跨信道聚合的作用，从而降低（或升高）数据的维度，可以达到减少参数的目的。换句话说，1×1 的卷积核操作实现的其实就是一个特征数据中的多个 Feature Map 的线性组合，所以这个方法就可以用来改变特征数据的信道数。

池化能压缩数据的高度（n_H）及宽度（n_W），而 1×1 卷积能压缩数据的通道数（n_C）。在如下图所示的例子中，用 32 个大小为 1×1×192 的滤波器进行卷积，就能使原先数据包含的 192 个通道压缩为 32 个。

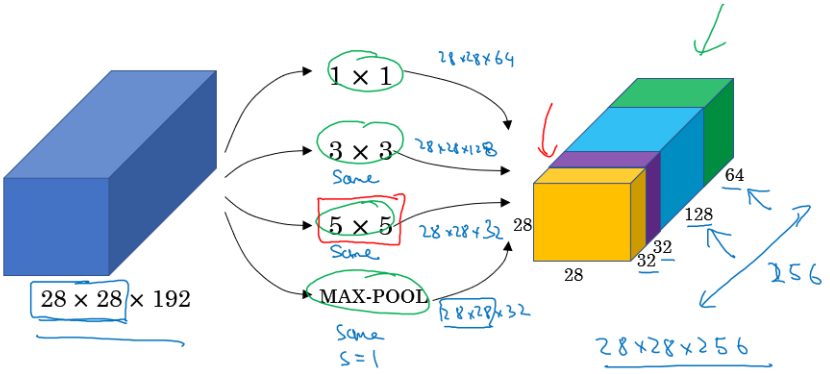


1.4 Inception 网络

最早的 Inception 结构的 V1 版本是由 Google 的 Szegedy 2014 年在论文 Going deeper with convolutions (https://arxiv.org/pdf/1409.4842.pdf) 中提出的，它是 ILSVRC 2014 中取得最好成绩的GoogLeNet 中采用的的核心结构。通过不断改进，现在已经衍生有了 V4 版本。

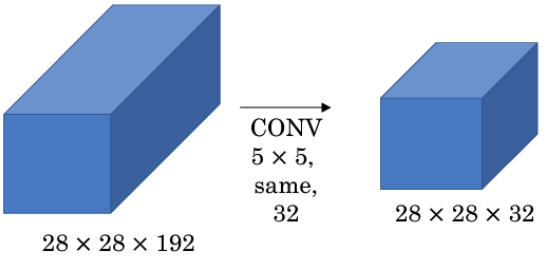
在之前的卷积网络中，我们只能选择单一尺寸和类型的滤波器。而 Inception 网络的作用即是代替人工来确定卷积层中的滤波器尺寸与类型，或者确定是否需要创建卷积层或池化层。

早期的 V1 版本的结构借鉴了NIN 的设计思路，对网络中的传统卷积层进行了修改，其结构大致如下面的例子中所示：

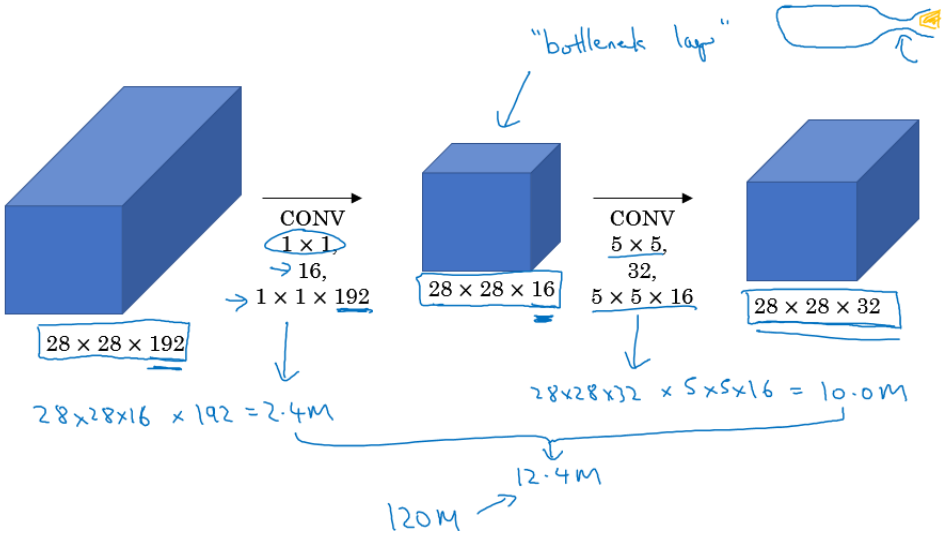


计算成本问题

通常在设计一个卷积网络的结构时，需要考虑卷积过程、池化过程的滤波器的大小，甚至是要不要使用 1×1 卷积核。在 Inception 结构中，考虑到多个不同大小的卷积核（滤波器）能够增强网络的适应力，于是分别使用三个大小分别为 1×1、3×3、5×5 的卷积核进行 same 卷积，同时再加入了一个 same 最大池化过程。最后将它们各自得到的结果放在一起，得到了图中一个大小为 28×28×256 的结果。然而，这种结构中包含的参数数量庞大，对计算资源有着极大的依赖，上面的例子中光是与大小为 5×5 的滤波器进行卷积的过程就会产生 1 亿多个参数！



为了解决计算量大的问题，可以引入 1x1 卷积**降维**来减少其计算量。



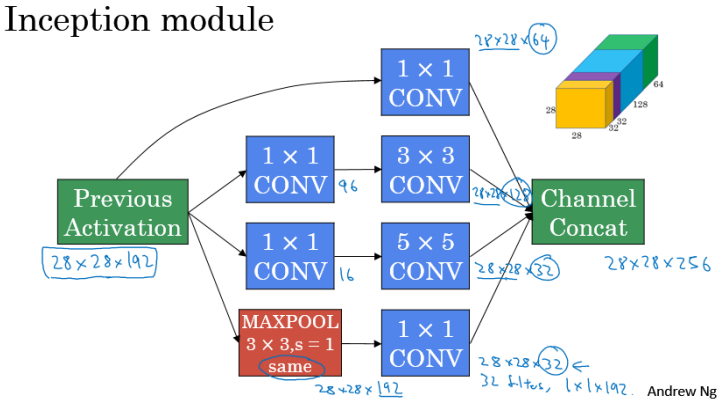
对于同一个例子，我们使用 1x1 卷积把输入数据从 192 个通道减少到 16 个通道，然后对这个较小层运行 5x5 卷积，得到最终输出。这个 1x1 的卷积层通常被称作**瓶颈层（Bottleneck layer）**。

改进后的计算量为 28x28x192x16 + 28x28x32x5x5x15 = 1.24 千万，减少了约 90%。

只要合理构建瓶颈层，就可以既显著缩小计算规模，又不会降低网络性能。

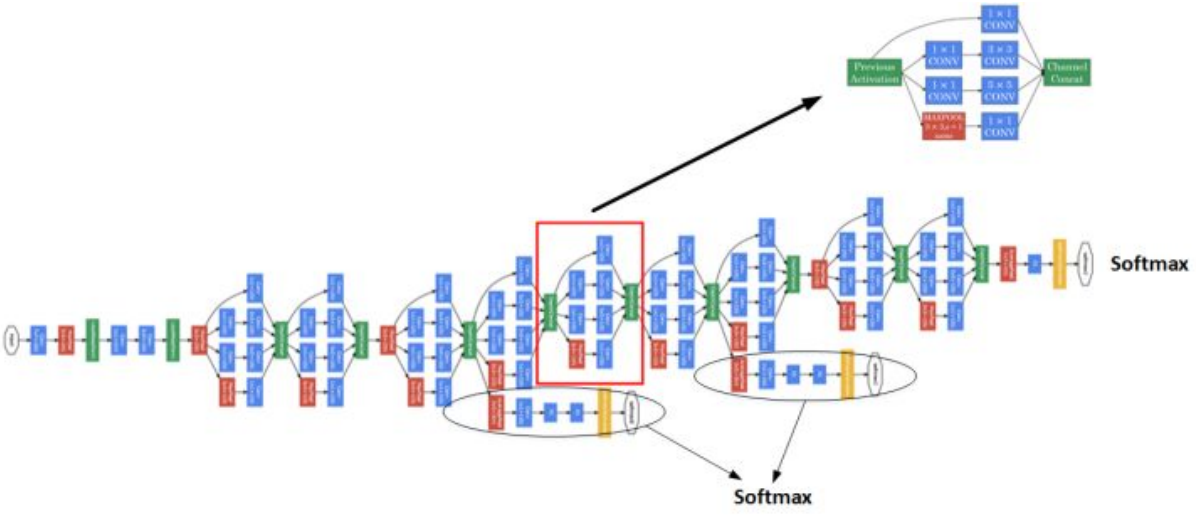
完整的 Inception 网络

在论文中提出的整个Inception模型结构如下：



上图是引入 1x1 卷积后的 Inception 模块。值得注意的是，为了将所有的输出组合起来，红色的池化层使用 Same 类型的填充（padding）来池化使得输出的宽高不变，通道数也不变。

多个 Inception 模块组成一个完整的 Inception 网络（被称为 GoogLeNet，以向 LeNet 致敬），如下图所示：



注意黑色椭圆圈出的隐藏层，这些分支都是 Softmax 的输出层，可以用来参与特征的计算及结果预测，起到调整并防止发生过拟合的效果。

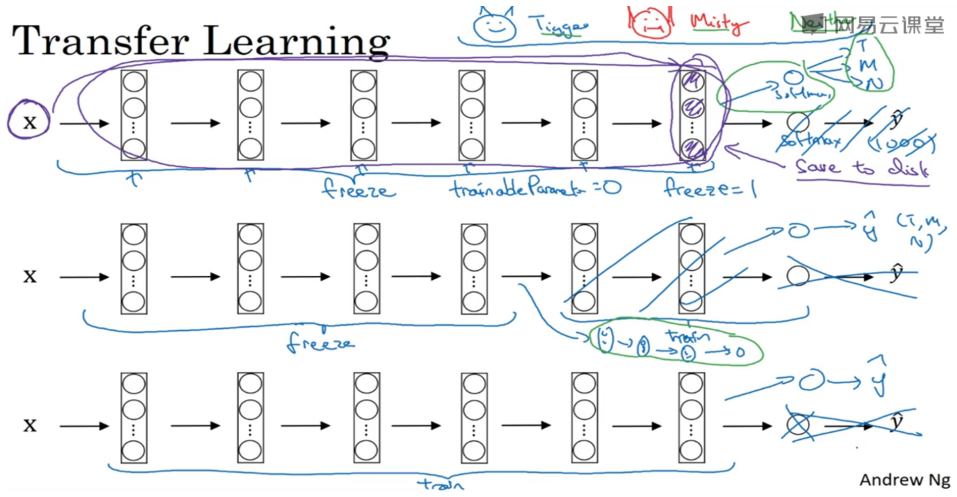
经过研究者们的不断发展，Inception 模型的 V2、V3、V4 以及引入残差网络的版本被提出，这些变体都基于 Inception V1 版本的基础思想上。顺便一提，Inception 模型的名字来自电影《盗梦空间》。

1.5 使用开源的实现方案

很多神经网络复杂细致，并充斥着参数调节的细节问题，因而很难仅通过阅读论文来重现他人的成果。想要搭建一个同样的神经网络，查看开源的实现方案会快很多，可以从参考Github等网站上其他人建立过的相关模型，必要时可以直接拿来根据拥有的数据量大小进行前面介绍过的迁移学习，从而减轻一些工作负担。

1.6 迁移学习

在“搭建机器学习项目”课程中，迁移学习已经被提到过。计算机视觉是一个经常用到迁移学习的领域。在搭建计算机视觉的应用时，相比于从头训练权重，下载别人已经训练好的网络结构的权重，用其做预训练，然后转换到自己感兴趣的任务上，有助于加速开发。



对于已训练好的卷积神经网络，可以将所有层都看作是冻结的，只需要训练与你的 Softmax 层有关的参数即可。大多数深度学习框架都允许用户指定是否训练特定层的权重。

而冻结的层由于不需要改变和训练，可以看作一个固定函数。可以将这个固定函数存入硬盘，以便后续使用，而不必每次再使用训练集进行训练了。

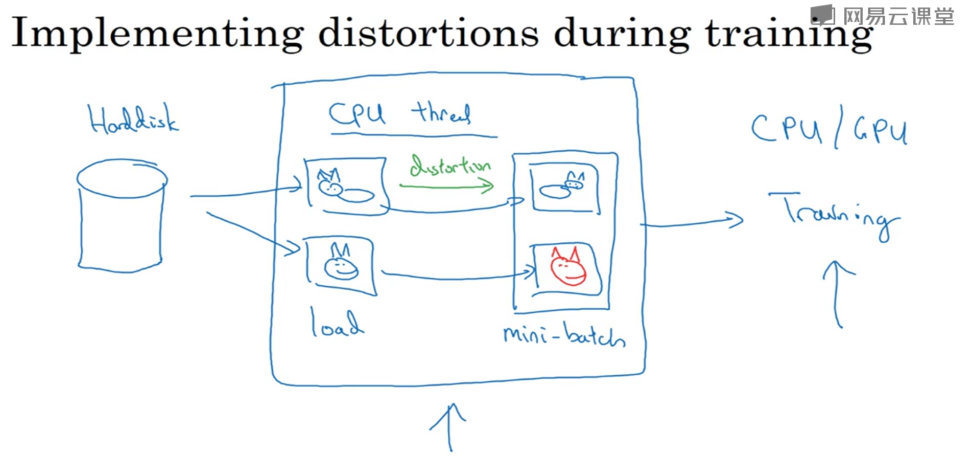
上述的做法适用于你只有一个较小的数据集。如果你有一个更大的数据集，应该冻结更少的层，然后训练后面的层。越多的数据意味着冻结越少的层，训练更多的层。如果有一个极大的数据集，你可以将开源的网络和它的权重整个当作初始化（代替随机初始化），然后训练整个网络。

1.7 数据扩增

计算机视觉领域的应用都需要大量的数据。当数据不够时，**数据扩增（Data Augmentation）**就有帮助。常用的数据扩增包括镜像翻转、随机裁剪、色彩转换。

其中，色彩转换是对图片的 RGB 通道数值进行随意增加或者减少，改变图片色调。另外，PCA 颜色增强指更有针对性地对图片的 RGB 通道进行主成分分析（Principles Components Analysis，PCA），对主要的通道颜色进行增加或减少，可以采用高斯扰动做法来增加有效的样本数量。具体的 PCA 颜色增强做法可以查阅 AlexNet 的相关论文或者开源代码。

在构建大型神经网络的时候，数据扩增和模型训练可以由两个或多个不同的线程并行来实现。



1.8 计算机视觉现状

通常，学习算法有两种知识来源：

- 被标记的数据

- 手工工程

手工工程（Hand-engineering，又称 hacks）指精心设计的特性、网络体系结构或是系统的其他组件。手工工程是一项非常重要也比较困难的工作。在数据量不多的情况下，手工工程是获得良好表现的最佳方式。正因为数据量不能满足需要，历史上计算机视觉领域更多地依赖于手工工程。近几年数据量急剧增加，因此手工工程量大幅减少。

另外，在模型研究或者竞赛方面，有一些方法能够有助于提升神经网络模型的性能：

- 集成（Ensembling）：独立地训练几个神经网络，并平均输出它们的输出
- Multi-crop at test time：将数据扩增应用到测试集，对结果进行平均

但是由于这些方法计算和内存成本较大，一般不适用于构建实际的生产项目。