深度学习入门

基于Python的理论与实现

Stephen CUI

March 14, 2023

Chapter 1

深度学习

深度学习是加深了层的深度神经网络。基于之前介绍的网络，只需通过叠加层，就可以创建深度网络。

1.1 加深网络

1.1.1 向更深的网络出发

这里我们来创建一个如 Figure 1.1 所示的网络结构的CNN。

1.1.2 进一步提高识别精度

在一个标题为“What is the class of this image ?”的网站上，以排行榜的形式刊登了目前为止通过论文等渠道发表的针对各种数据集的方法的识别精度。 排行榜中前几名的方法，可以发现进一步提高识别精度的技术和线索。比如，集成学习、学习率衰减、**Data Augmentation**（数据扩充）等都有助于提高识别精度。尤其是Data Augmentation，虽然方法很简单，但在提高识别精度上效果显著。

Data Augmentation基于算法“人为地”扩充输入图像（训练图像）。具体地说，如 Figure 1.3 所示，对于输入图像，通过施加旋转、垂直或水平方向上的移动等微小变化，增加图像的数量。这在数据集的图像数量有限时尤其有效。

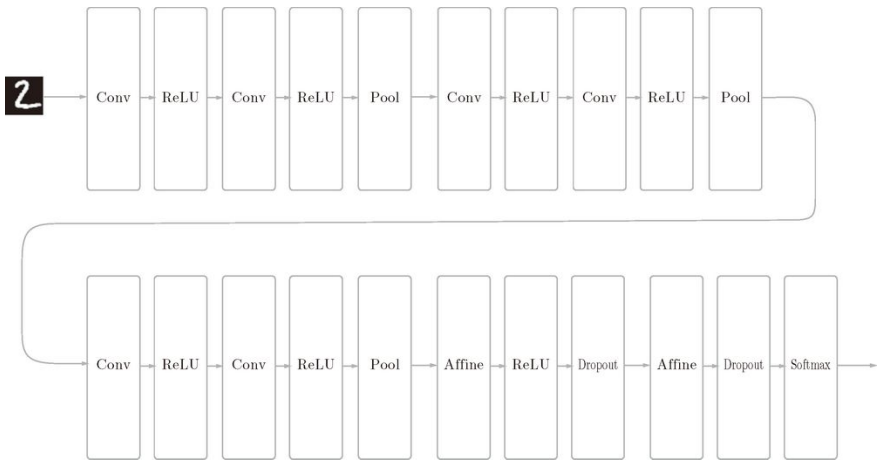

Figure 1.1: Deep CNN for Handwritten Digit Recognition

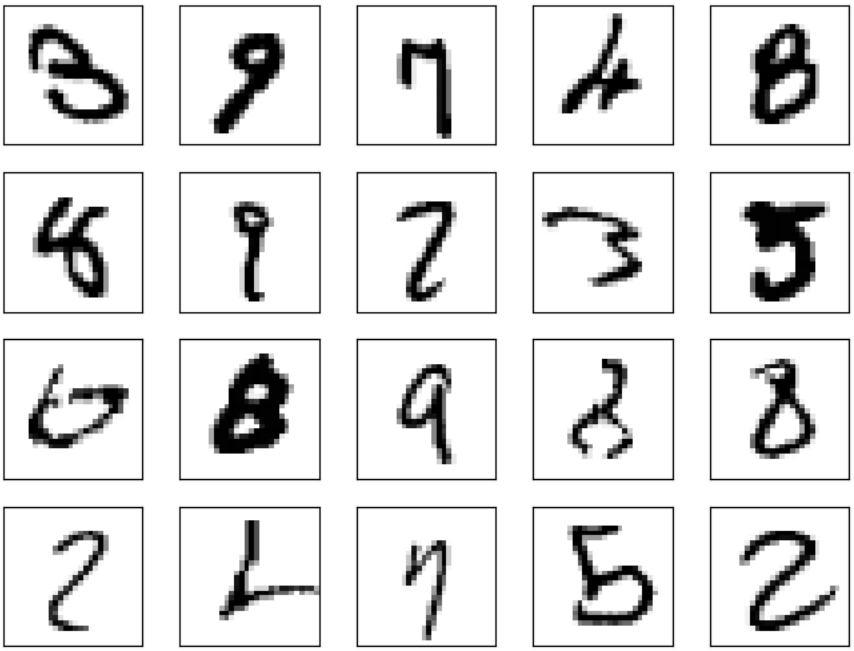

Figure 1.2: Examples of misidentified images

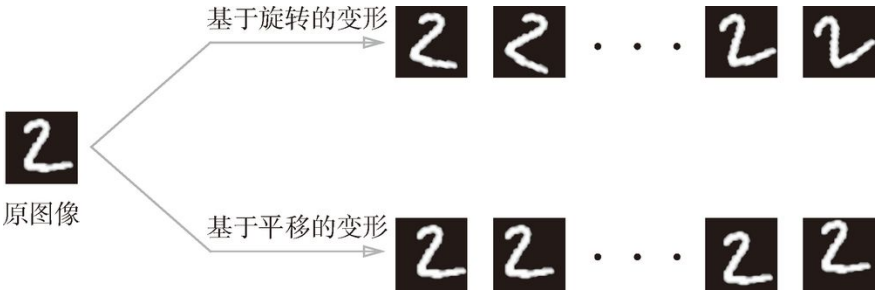

Figure 1.3: Data Augmentation example

除了如 Figure 1.3 所示的变形之外，Data Augmentation还可以通过其他各种方法扩充图像，比如裁剪图像的“crop处理”、将图像左右翻转的“fl ip处理”[1]等。对于一般的图像，施加亮度等外观上的变化、放大缩小等尺度上的变化也是有效的。不管怎样，通过Data Augmentation巧妙地增加训练图像，就可以提高深度学习的识别精度。虽然这个看上去只是一个简单的技巧，不过经常会有很好的效果。

1.1.3 加深层的动机

关于加深层的重要性，现状是理论研究还不够透彻。尽管目前相关理论还比较贫乏，但是有几点可以从过往的研究和实验中得以解释。

首先，从以ILSVRC为代表的大规模图像识别的比赛结果中可以看出加深层的重要性（详细内容请参考下一节）。这种比赛的结果显示，最近前几名的方法多是基于深度学习的，并且有逐渐加深网络的层的趋势。也就是说，可以看到层越深，识别性能也越高。下面我们说一下加深层的好处。其中一个好处就是可以减少网络的参数数量。说得详细一点，就是与没有加深层的网络相比，加深了层的网络可以用更少的参数达到同等水平（或者更强）的表现力。

[1] flip处理只在不需要考虑图像对称性的情况下有效。

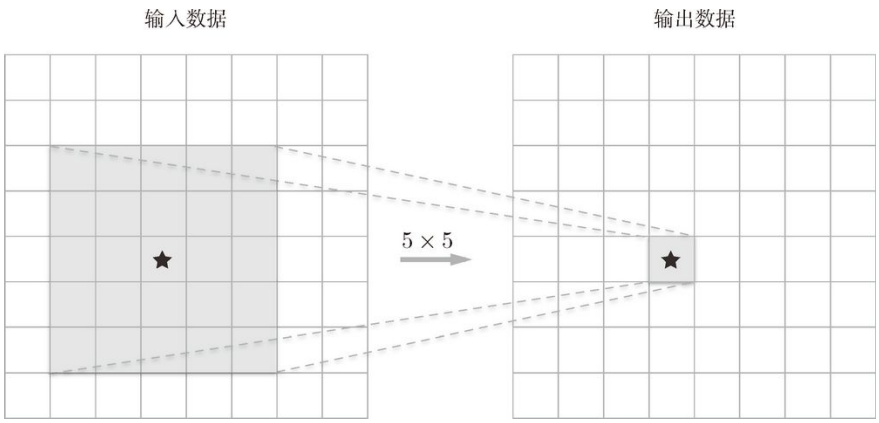

Figure 1.4: 5-5 convolution example

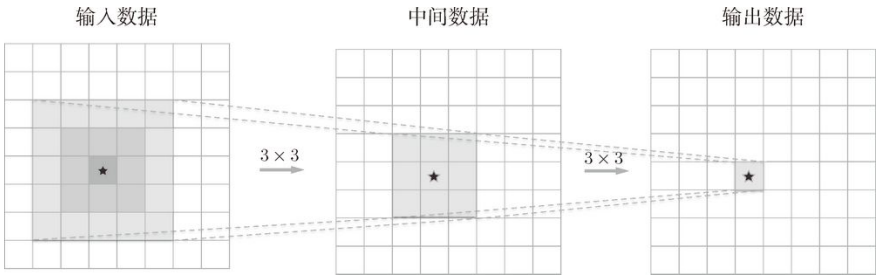

Figure 1.5: Example of a convolutional layer repeated twice 3-3

显然，在 Figure 1.4 的例子中，每个输出节点都是从输入数据的某个 5×5的区域算出来的。接下来我们思考一下 Figure 1.5 中重复两次5×5的卷积运算的情形。此时，每个输出节点将由中间数据的某个3×3的区域计算出来。

一次5×5的卷积运算的区域可以由两次3×3的卷积运算抵充。并且，相对于前者的参数数量25（5×5），后者一共是18（$2 \times 3 \times 3$），通过叠加卷积层，参数数量减少了。而且，这个参数数量之差会随着层的加深而变大。比如，重复三次3×3的卷积运算时，参数的数量总共是27。而为了用一次卷积运算“观察”与之相同的区域，需要一个7×7的滤波器，此时的参数数量是49。

> 叠加小型滤波器来加深网络的好处是可以减少参数的数量，扩大感受野（receptive field，给神经元施加变化的某个局部空间区域）。并且，通过叠加层，将ReLU等激活函数夹在卷积层的中间，进一步提高了网络的表现力。这是因为向网络添加了基于激活函数的“非线性” 表现力，通过非线性函数的叠加，可以表现更加复杂的东西。

加深层的另一个好处就是使学习更加高效。与没有加深层的网络相比，通过加深层，可以减少学习数据，从而高效地进行学习。

1.2 深度学习的小历史

1.2.1 VGG

VGG 是由卷积层和池化层构成的基础的 CNN。不过，如图 8-9所示，它的特点在于将有权重的层（卷积层或者全连接层）叠加至16层（或者19层），具备了深度（根据层的深度，有时也称为“VGG16”

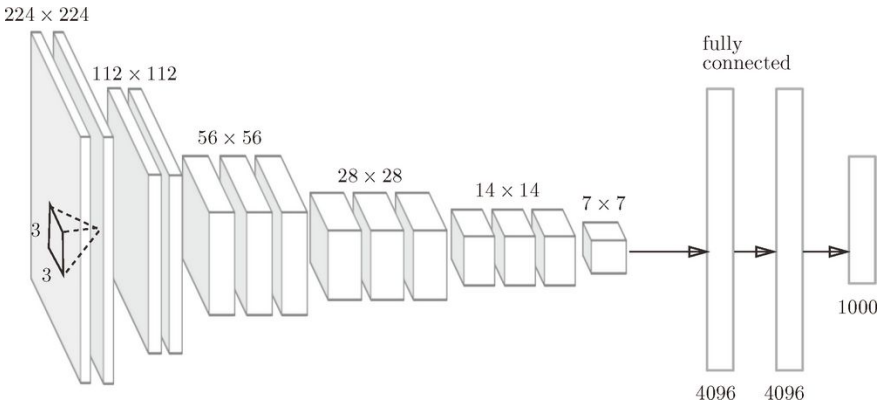

Figure 1.6: VGG

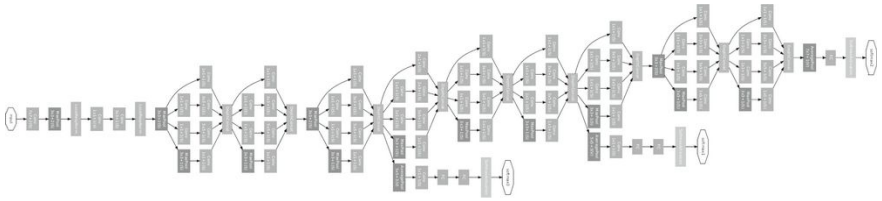

Figure 1.7: GoogLeNet

或“VGG19”）。

VGG中需要注意的地方是，基于 3×3 的小型滤波器的卷积层的运算是连续进行的。如 Figure 1.6 所示，重复进行“卷积层重叠2次到4次，再通过池化层将大小减半”的处理，最后经由全连接层输出结果。

1.2.2 GoogLeNet

只看 Figure 1.7 的话，这似乎是一个看上去非常复杂的网络结构，但实际上它基本上和之前介绍的CNN结构相同。不过，GoogLeNet的特征是，网络不仅在纵向上有深度，在横向上也有深度（广度）。

GoogLeNet在横向上有“宽度”，这称为“Inception结构”。

Figure 1.8，Inception结构使用了多个大小不同的滤波器（和池化），最后再合并它们的结果。GoogLeNet的特征就是将这个Inception结构用作一个构件（构成元素）。此外，在GoogLeNet中，很多地方都使用了大小为 1×1 的滤波器的卷积层。这个1 × 1的卷积运算通过在通道方向上减小大小，有助于减少参数和实现高速化处理。

1.2.3 ResNet

ResNet是微软团队开发的网络。它的特征在于具有比以前的网络更深的结构。

我们已经知道加深层对于提升性能很重要。但是，在深度学习中，过度加深层的话，很多情况下学习将不能顺利进行，导致最终性能不佳。ResNet中，为了解决这类问题，导入了“快捷结构”（也称为“捷径”或“小路”）。导入这个快捷结构后，就可以随着层的加深而不断提高性能了（当然，层的加深也是有限度的）。

如 Figure 1.9 所示，快捷结构横跨（跳过）了输入数据的卷积层，将输入 x合计到输出。 Figure 1.9 中，在连续2层的卷积层中，将输入x跳着连接至2层后的输出。这里的重点是，通过快捷结构，原

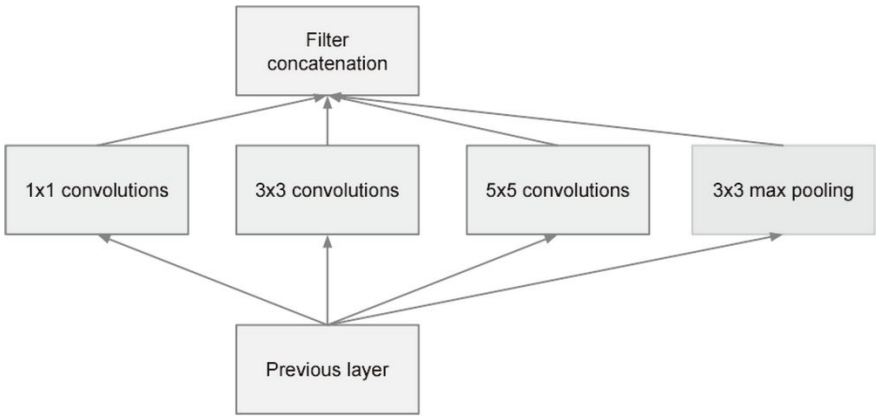

Figure 1.8: Inception structure of GoogLeNet

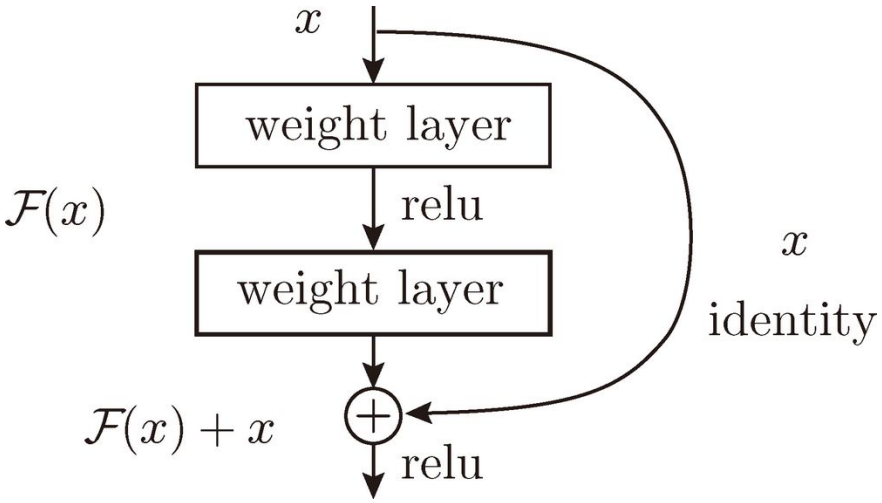

Figure 1.9: Components of ResNet

来的2层卷积层的输出$\mathcal{F}(x)$变成了$\mathcal{F}(x)+x$。通过引入这种快捷结构，即使加深层，也能高效地学习。这是因为，通过快捷结构，反向传播时信号可以无衰减地传递。

因为快捷结构只是原封不动地传递输入数据，所以反向传播时会将来自上游的梯度原封不动地传向下游。这里的重点是不对来自上游的梯度进行任何处理，将其原封不动地传向下游。因此，基于快捷结构，不用担心梯度会变小（或变大），能够向前一层传递“有意义的梯度”。通过这个快捷结构，之前因为加深层而导致的梯度变小的梯度消失问题就有望得到缓解。

实践中经常会灵活应用使用ImageNet这个巨大的数据集学习到的权重数据，这称为**迁移学习**，将学习完的权重（的一部分）复制到其他神经网络，进行再学习（fine tuning）。比如，准备一个和VGG相同结构的网络，把学习完的权重作为初始值，以新数据集为对象，进行再学习。迁移学习在手头数据集较少时非常有效。

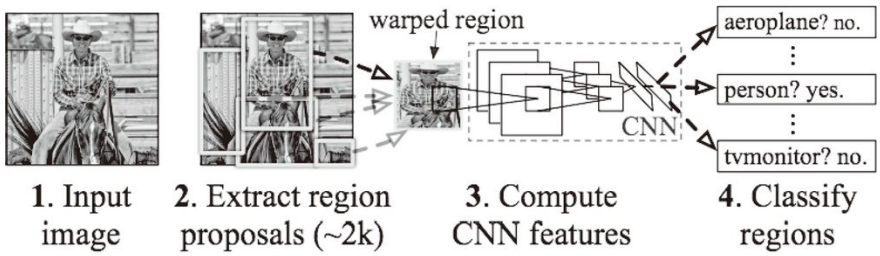

Figure 1.10: Processing flow of R-CNN

1.3 深度学习的高速化

1.3.1 需要努力解决的问题

1.3.2 基于GPU的高速化

GPU计算，是指基于GPU进行通用的数值计算的操作。

深度学习中需要进行大量的乘积累加运算（或者大型矩阵的乘积运算）。这种大量的并行运算正是GPU所擅长的（反过来说，CPU比较擅长连续的、复杂的计算）。

1.3.3 分布式学习

1.3.4 运算精度的位数缩减

1.4 深度学习的应用案例

1.4.1 物体检测

物体检测是从图像中确定物体的位置，并进行分类的问题。

对于这样的物体检测问题，人们提出了多个基于CNN的方法。这些方法展示了非常优异的性能，并且证明了在物体检测的问题上，深度学习是非常有效的。

在使用CNN进行物体检测的方法中，有一个叫作R-CNN的有名的方法。Figure 1.10 显示了R-CNN的处理流。

1.4.2 图像分割

1.4.3 图像标题生成

1.5 深度学习的未来

1.5.1 图像风格变换

1.5.2 图像的生成

1.5.3 自动驾驶

1.5.4 Deep Q-Network(强化学习)