CNN 流行框架总结

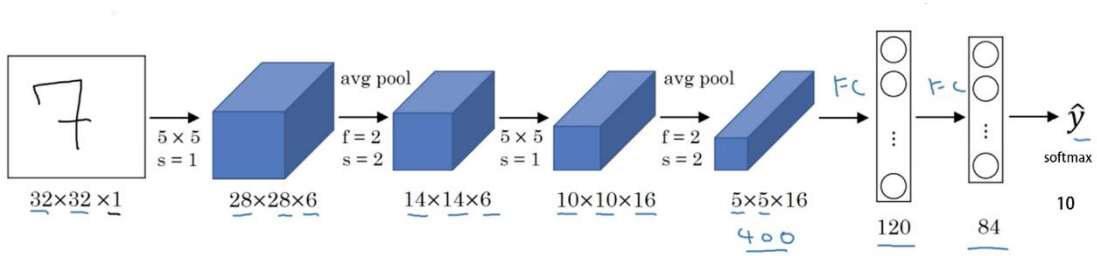
郑俊杰　　18052137

LeNet——

LeNet-5 是 Yann LeCun 等人在多次研究后提出的最终卷积神经网络结构，一般 LeNet 即指代 LeNet-5。

手写字体识别模型 LeNet5 诞生于 1994 年，是最早的卷积神经网络之一。LeNet5 通过巧妙的设计，利用卷积、参数共享、池化等操作提取特征，避免了大量的计算成本，最后再使用全连接神经网络进行分类识别，这个网络也是最近大量神经网络架构的起点。

LeNet-5 共有 7 层，分别为：两组卷积池化层、和三层全连接层，每层都包含可训练参数；



输入层

图片大小为 32×32×1，其中 1 表示为黑白图像，只有一个 channel。

卷积层

卷积核大小 5×5，卷积核深度（个数）为 6，padding 为 0， 卷积步长 s=1，输出矩阵大小为 28×28×6，其中 6 表示 卷积核的个数。

池化层

卷积核大小 2×2（即 f=2），步长 s=2，no padding，输出矩阵大小为 14×14×6。

卷积层

卷积核大小 5×5，卷积核个数为 16，padding 为 0， 卷积步长 s=1，输出矩阵大小为 10×10×16，其中 16 表示 卷积核的个数。

池化层

卷积核大小 2×2（即 f=2），步长 s=2，no padding，输出矩阵大小为 5×5×16。注意，在该层结束，需要将 5×5×16 的矩阵 flatten 成一个 400 维的向量。

全连接层（Fully Connected layer，FC）

neuron 数量为 120。

全连接层（Fully Connected layer，FC）

neuron 数量为 84。

全连接层，输出层

现在版本的 LeNet-5 输出层一般会采用 softmax 激活函数，在 LeNet-5 提出的论文中使用的激活函数不是 softmax，但其现在不常用。该层神经元数量为 10，代表 0～9 十

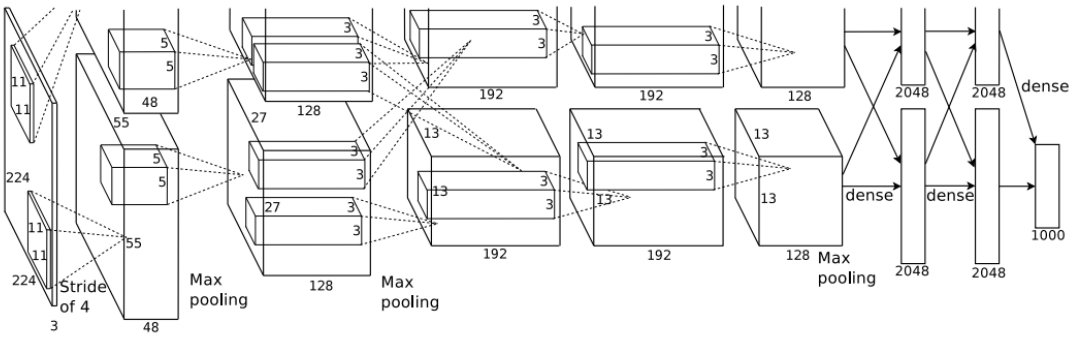
个数字类别。

LeNet 最开始应用在手写数字识别中，应用于识别美国邮政服务提供的手写邮政编码数字。

LeNet 模型在当时的表现超过了其他所有模型，泛化能力强，能够得出原始图像的有效表征，这使得它能够直接从原始像素中，经过极少的预处理，识别视觉上面的规律。
但是，LeNet 的设计较为简单，因此其处理复杂数据的能力有限；此外，在近年来的研究中许多学者已经发现全连接层的计算代价过大，而使用全部由卷积层组成的神经网络。

AlexNet——

AlexNet 由 Alex Krizhevsky 于 2012 年提出，夺得 2012 年 ILSVRC 比赛的冠军，top5 预测的错误率为 16.4%，远超第一名。AlexNet 采用 8 层的神经网络，5 个卷积层和 3 个全连接层(3 个卷积层后面加了最大池化层)，包含 6 亿 3000 万个链接，6000 万个 参数和 65 万个神经元。



上图中的输入是 224×224，不过经过计算(224-11)/4=54.75 并不是论文中的 55×55，而使用 227×227 作为输入，则(227-11)/4=55
（卷积池化作为同一层）

输入层

图像大小为 227×227×3，其中 3 表示输入图像的 channel 数（R，G，B）为 3。

卷积层

卷积核大小 11×11，卷积核个数 96，卷积步长 s=4。（卷积核大小只列出了宽和高，filter 矩阵的 channel 数和输入图片的 channel 数一样，在这里没有列出）

池化层

max pooling，卷积核大小 3×3，步长 s=2。

卷积层

卷积核大小 5×5，卷积核个数 256，步长 s=1，padding 使用 same convolution，即使得卷积层输出图像和输入图像在宽和高上保持不变。

池化层

max pooling，卷积核大小 3×3，步长 s=2。

卷积层

卷积核大小 3×3，卷积核个数 384，步长 s=1，padding 使用 same convolution。

卷积层

卷积核大小 3×3，卷积核个数 384，步长 s=1，padding 使用 same convolution。

卷积层

卷积核大小 3×3，卷积核个数 256，步长 s=1，padding 使用 same convolution。

池化层

max pooling，卷积核大小 3×3，步长 s=2；池化操作结束后，将大小为 6×6×256 的输出矩阵 flatten 成一个 9216 维的向量。

全连接层

neuron 数量为 4096。

全连接层

neuron 数量为 4096。

全连接层，输出层

使用了 ReLU 激活函数，neuron 数量为 1000，代表 1000 个类别。

AlexNet 有如下的创新点：

（1）ReLU 作为激活函数：ReLU 为非饱和函数，论文中验证其效果在较深的网络超过了 sigmoid，成功解决了 sigmoid 在网络较深时的梯度弥散问题。

（2）Dropout 避免模型过拟合：在训练时使用 Dropout 随机忽略一部分神经元，以避免模型过拟合。在 AlexNet 的最后几个全连接层中使用了 Dropout。

（3）重叠的最大池化：之前的 CNN 中普遍使用平均池化，而 AlexNet 全部使用最大池化，避免平均池化的模糊化效果。并且，池化的步长小于核尺寸，这样使得池化层的输出之间会有重叠和覆盖，提升了特征的丰富性。

（4）提出 LRN 层：提出 LRN 层，对局部神经元的活动创建竞争机制，使得响应较大的值变得相对更大，并抑制其他反馈较小的神经元，增强了模型的泛化能力。

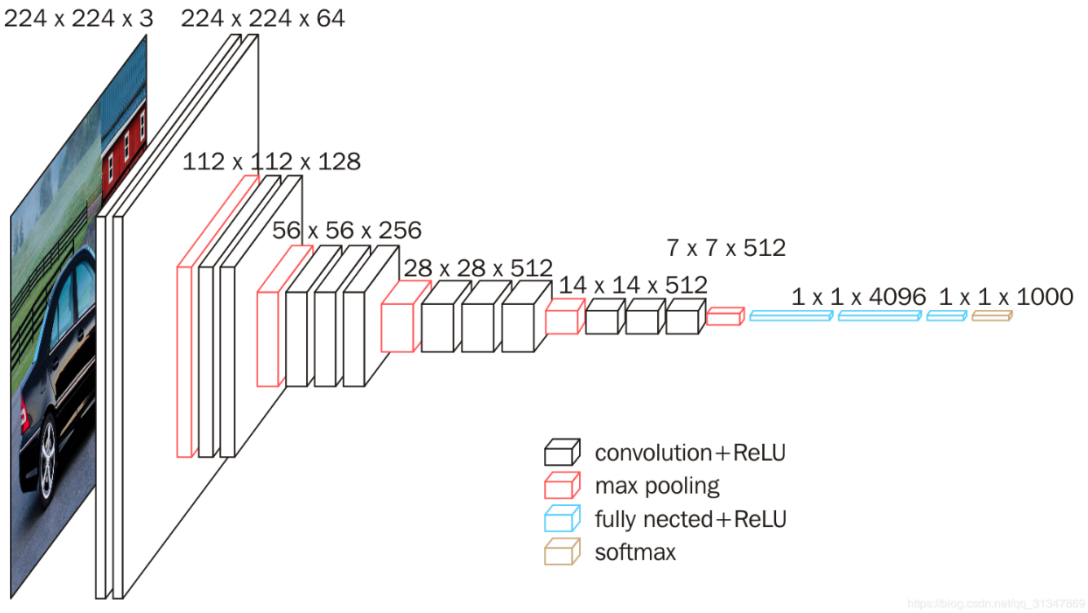
（5）GPU 加速

（6）数据增强：随机从 256*256 的原始图像中截取 224*224 大小的区域（以及水平翻转的镜像），相当于增强了（256-224）*（256-224）*2=2048 倍的数据量。使用了数据增强后，减轻过拟合，提升泛化能力。避免因为原始数据量的大小使得参数众多的 CNN 陷入过拟合中。

VGG——

VGG 是 Oxford Visual Geometry Group 的简称。该小组隶属于 1985 年成立的 Robotics Research Group，该 Group 研究范围包括了机器学习到移动机器人。该团队斩获 2014 年 ImageNet 挑战赛分类第二（第一是 GoogLeNet），定位任务第一。

VGG-16 中的 16 表示整个网络中有 trainable 参数的层数为 16 层。（trainable 参数指的是可以通过 back-propagation 更新的参数）



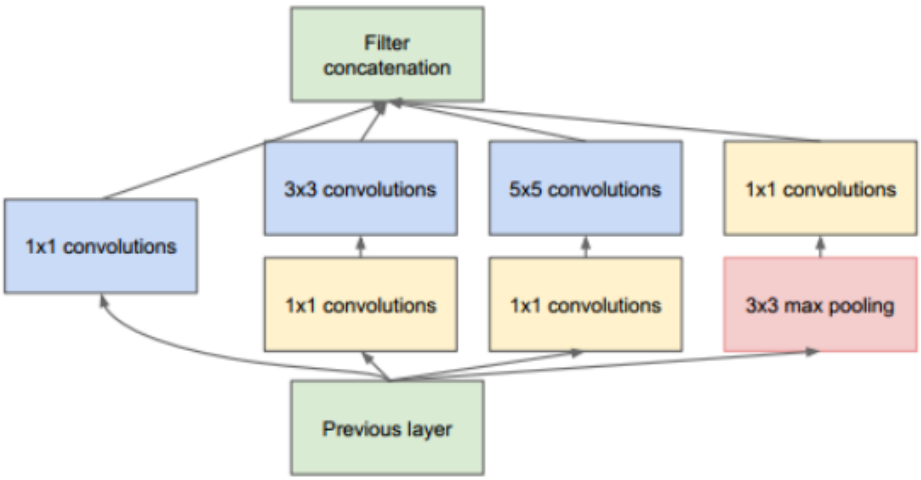
输入图像大小为 224×224×3。

VGG Net 使用的全部都是 3x3 的小卷积核和 2x2 的池化核，由 5 个卷积段和最后的全连接构成，每个卷积段由 2-3 个卷积层组成。每段内的卷积核数量都一样，越靠后的段的卷积核数量越多：64-128-256-512-512，结尾都会连接一个最大池化层。在最后由三段全连接层加上 softmax 输出结果。

VGG 的卷积层全部由 3×3 和 1×1 构成。两层 3x3 的串联卷积结果相当于一个 5x5 的卷积，即最后一个像素会与周围 5x5 个像素产生关联，可以说感受野大小为 5x5，而 3 层 3x3 的卷积核的串联结果则相当于 1 个 7x7 的卷积层。除此之外，3 个串联的 3x3 卷积层的参数数量要比一个 7x7 卷积层的参数数量小得多，即 3*3*3*Cˆ2/7*7*Cˆ2= 55%，更少的参数意味着减少过拟合，而且更重要的是 3 个 3x3 卷积层拥有比 1 个 7x7 的卷积层更少的非线性变换（前者拥有 3 次而后者只有一次），使得 CNN 对特征的学习能力更强。而之所以卷积层中卷积核的个数是由小变大的，那是因为低维度的特征较为简单，并且开始时候特征图的尺寸较大，这样做可以节省一部分内存。随着网络的加深，一方面特征图在经过池化后不断缩小尺寸，另一方面卷积核感受野不断增大，学习到了更加复杂的特征，因此卷积核个数需要加大。

GoogLeNet——

GoogLeNet 是 2014 年 Christian Szegedy 提出的一种全新的深度学习结构，在这之前的 AlexNet、VGG 等结构都是通过增大网络的深度（层数）来获得更好的训练效果，但层数的增加会带来很多副作用，比如过拟合、梯度消失、梯度爆炸等。inception 的提出则从另一种角度来提升训练结果：能更高效的利用计算资源，在相同的计算量下能提取到更多的特征，从而提升训练结果。



inception 模块从神经网络的宽度着手，在增加网络深度和宽度的同时又减少了参数。他对输入做了 4 个分支，每个分支用大小不同的 filter(1×1，3×3，5×5)进行卷积或池化，同时，又加入了 1×1 的 filter 卷积，对数据进行降维，极大的减少了计算量和参数数量。

输入

原始输入图像为 224x224x3，且都进行了零均值化的预处理操作（图像每个像素减去均值）。

卷积层

使用 7x7 的卷积核（滑动步长 2，padding 为 3），64 通道，输出为 112x112x64，卷积后进行 ReLU 操作

经过 3x3 的 max pooling（步长为 2），输出为((112 - 3+1)/2)+1=56，即 56x56x64，再进行 ReLU 操作

卷积层

使用 3x3 的卷积核（滑动步长为 1，padding 为 1），192 通道，输出为 56x56x192，卷积后进行 ReLU 操作

经过 3x3 的 max pooling（步长为 2），输出为((56 - 3+1)/2)+1=28，即 28x28x192，再进行 ReLU 操作

Inception 3a 层

分为四个分支，采用不同尺度的卷积核来进行处理

（1）64 个 1x1 的卷积核，然后 RuLU，输出 28x28x64

（2）96 个 1x1 的卷积核，作为 3x3 卷积核之前的降维，变成 28x28x96，然后进行 ReLU 计算，再进行 128 个 3x3 的卷积（padding 为 1），输出 28x28x128

（3）16 个 1x1 的卷积核，作为 5x5 卷积核之前的降维，变成 28x28x16，进行 ReLU 计算后，再进行 32 个 5x5 的卷积（padding 为 2），输出 28x28x32

（4）pool 层，使用 3x3 的核（padding 为 1），输出 28x28x192，然后进行 32 个 1x1 的卷积，输出 28x28x32。

将四个结果进行连接，对这四部分输出结果的第三维并联，即 64+128+32+32=256，最终输出 28x28x256

…

…

GoogLeNet 深度为 22，数据先是进入两个卷积段，对输入的数据进行第一步处理，之后进入 9 个 inception 之中进行计算。

GoogLeNet 中第 4 和第 7 个 inception 之后又连接了一个辅助 softmax 分类器来处理梯度下降的问题。在训练模型过程中这一结构可以增强反向传播的信号，对其起到正面作用。在实际预测中，这两个分类器则被闲置，不起作用。