Software Engineering and Applications 软件工程与应用, 2020, 9(5), 360-372
Published Online October 2020 in Hans. http://www.hanspub.org/journal/sea
https://doi.org/10.12677/sea.2020.95041

Hans汉斯

卷积神经网络池化方法综述

袁铭阳[1]，周长胜[1,2]，黄宏博[1,2*]，胡志颖[1]，李　颖[1]

[1]北京信息科技大学计算机学院，北京
[2]北京信息科技大学计算智能研究所，北京
Email: 1413241523@qq.com, *hhb@bistu.edu.cn

收稿日期：2020年9月23日；录用日期：2020年10月6日；发布日期：2020年10月13日

摘　要

池化层是卷积神经网络的重要组成部分，池化层通过池化计算对经过卷积层后的特征图进行降维。随着卷积神经网络的发展，产生了许多新的池化方法代替传统的池化方法，在多类任务中取得了突破性进展。本文针对基于卷积神经网络的池化方法进行综述，对池化方法进行了分类，详细阐述了各种新的池化方法相较于传统池化方法的改进之处，介绍了池化方法的具体计算方法，并且对各种池化方法的效果进行了对比，最后给出了池化方法在主流数据集上的性能指标。

关键词

卷积神经网络，池化方法，池化层

Survey on Convolutional Neural Network Pooling Methods

Mingyang Yuan[1], Changsheng Zhou[1,2], Hongbo Huang[1,2*], Zhiying Hu[1], Ying Li[1]

[1]Computer School, Beijing Information Science & Technology University, Beijing
[2]Institute of Computing Intelligence, Beijing Information Science & Technology University, Beijing
Email: 1413241523@qq.com, *hhb@bistu.edu.cn

Received: Sep. 23rd, 2020; accepted: Oct. 6th, 2020; published: Oct. 13th, 2020

Abstract

The pooling layer is an important part of convolution neural network. The pooling layer reduces the dimension of the feature map after convolution layer through pool calculation. With the de-

*通讯作者。

文章引用: 袁铭阳, 周长胜, 黄宏博, 胡志颖, 李颖. 卷积神经网络池化方法综述[J]. 软件工程与应用, 2020, 9(5): 360-372. DOI: 10.12677/sea.2020.95041

velopment of convolutional neural network, many new pooling methods have been produced to replace the traditional pooling methods, and a breakthrough has been made in many kinds of tasks. This paper summarizes the pooling methods based on convolution neural network, classifies the pooling methods, describes the improvements of various new pooling methods compared with the traditional pooling methods, introduces the specific calculation methods of pooling methods, and compares the effects of various pooling methods, and finally gives the performance indicators of pooling methods on the mainstream datasets.

Keywords

Convolutional Neural Network, Pooling Method, Pooling Layer

1. 引言

传统神经网络的结构主要包括输入层、隐藏层和输出层，各层之间通过密集连接进行通信。深度卷积神经网络则主要通过卷积来实现层与层之间的信息传递，并引入池化层来进行数据维度的约减，非线性映射主要由激活函数实现。输入层的原始数据经过卷积层和激活层后会得到特征图，这些特征图仍包含了大量的冗余信息，往往通过池化层进行信息的再次提取。池化层的目的就是对这些大量的特征信息进行过滤，去除其中的冗余信息并筛选出最具代表性的特征信息，因此可以把池化层当作是一个滤波器。池化层的作用包括减少网络中参数的数量、压缩数据以及减少网络的过拟合。池化层里面主要包含了两个参数，分别是步长和池化核大小。池化核以滑动窗口的方式对输入的特征图进行处理，经过不同的池化函数的计算，得到相应的关键特征，其中每个池化层中的池化函数是固定的，一般不需要再引入其他参数。池化函数是池化层的核心，池化函数的不同也就对应着不同的池化方法。一个较好的池化方法通常能够在删除大量的无关信息的同时并且尽可能多的保留关键信息，进而在很大程度上提升整个卷积神经网络的性能。

池化方法中最常见的传统方法是最大池化和平均池化。最大池化只保留池化框中的最大值，因而最大池化可以有效提取出特征图中最具代表性的信息。平均池化则计算出池化框中所有值的均值，因而可以平均获取特征图中的所有信息，进而不致丢失过多关键信息。这两种方法由于计算简单且效果较好因而被广泛利用在了各种结构的卷积神经网络中，但这两种方法的缺点也是不可忽视的。最大池化由于完全删除了最大值以外的其他值，这往往导致保留了特征图中的前景信息而忽略了所有的背景信息；而平均池化由于取得了所有值之和的均值，虽然对特征图中的背景信息有所保留，但是无法将特征图中的前景信息和背景信息有效地区分开。随着卷积神经网络的不断优化，国内外有许多新的池化方法相继被提出。其中大部分是通过改变传统池化方法的计算方式，也有少部分将最大池化和平均池化以不同形式结合，这些池化方法在目标检测、图像识别等领域均有较好的运用。本文把这些新的方法分为池化核大小固定的池化方法和池化核大小不固定的池化方法。池化核大小固定的池化方法分为序相关和序无关的两类。序无关的池化方法又分为确定性池化方法和随机性池化方法。新的池化方法运用在不同的数据集上均较传统方法在精度、错误率和鲁棒性等关键指标中获得较大提升。本文将按时间顺序对所有新的池化方法进行阐述，并重点介绍各种新的池化方法中极具创新性和突破性的要点。

2. 池化核大小固定的池化方法

池化核大小固定的池化方法包含序相关和序无关的两类，这两类池化方法的区别在于池化结果是否与池化核中的各元素的序有关。与序无关的池化方法指的是池化核中各元素大小的排序不影响最后的池化结果，反之，与序相关的池化方法则需要注意池化核中各元素大小的排序。例如最大池化就属于序相关的池化方法，需要计算池化核内的各元素中的最大值作为输出。

2.1. 序无关的池化方法

序无关的这类池化方法又分为确定性池化方法和随机性池化方法。两种池化方法的区别在于池化层的计算方式是否会随着网络结构、池化核内元素等各种因素的不同而改变，会发生改变的则属于随机性池化，反之则属于确定性池化。例如平均池化就属于确定性池化，会保持计算池化核内所有元素的平均值这一计算方式，不会发生改变。

2.1.1. 确定性池化

(1) 谱池化

谱池化(Spectral Pooling) [1] [2]是一种基于快速傅里叶变换(FFT)的池化方法。傅里叶变换可以把输入的信号从时间域转换到频率域。由于其效率较高和潜在的成本降低，离散傅里叶变换一直被深度学习界认为是快速卷积的自然方法，实验也证明使用离散傅里叶变换计算卷积比直接在空间域计算卷积要快得多。在卷积神经网络中，由于输入的图片是二维信号，因此需要二维离散傅里叶变换对图片进行转换，并构建一个基于 FFT 的卷积神经网络。谱池化首先把输入信号投影到频率基极，然后截断其中的部分频率来进行降维。具体来说，假设输入的是一张 $m \times n$ 的特征图而在池化后需要得到 $a \times a$ 的尺度(其中 m、n 均大于 a)，谱池化先通过傅里叶变换将 $m \times n$ 的特征图转换为频率，再将频率中心的 $a \times a$ 部分截取出来，然后将截取的部分通过傅里叶变换的逆运算得到 $a \times a$ 尺度的特征图。由于 m、n 和 a 的大小都可以是任意值，因此谱池化同样可以对任意大小的图片进行处理且能输出任意维度的图片。谱池化的第一个优点在于通过精确的调整输入的分辨率用于匹配期望的输出维度来降低信息容量，在同种输出维度下相对于传统池化方法可以显著增加保留的信息量。第二个优点在于谱池化允许指定任意的输出维度，因此不易受其他池化方法所表现出的输出维度急剧下降对性能带来的负面影响。从 CIFAR-10 和 CIFAR-100 数据集的实验结果来看，谱池化的验证集错误率相较于随机池化降低了近一半，这能充分体现出谱池化的有效性。不过需要注意的是，由时间域转换到频率域在计算上非常密集，因此最好严格地保持在频率范围内。

(2) 双线性池化

细粒度识别长期以来都极具挑战性，因为这些类别之间的视觉差异很小。例如对不同的鸟的种类进行区分，有时需要捕获到鸟喙的特征才能识别其种类，同时，这些微小的差别也容易受遮挡、光照等自然环境因素影响。双线性池化(Bilinear Pooling) [3] [4] [5] [6]是专门用来处理细粒度识别问题的一种池化方法，其基本思想是通过把两个网络提取的同一位置的两个不同特征结合起来用于细粒度识别。双线性卷积神经网络模型示意图如图 1 所示。

从图中可以看出，输入图像通过两个卷积神经网络，A 和 B，它们的输出在每个位置进行双线性组合。将 A 和 B 两个结果汇集在一起，得到双线性向量，然后通过分类层获得预测结果。其具体操作是，网络 A 可以对图片中各部分的坐标信息进行建模，而网络 B 则对图片中各部分的事物信息进行建模，再把同一位置通过网络 A 和 B 提取的特征向量，将网络 A 提取的特征向量转置再与网络 B 提取的特征向量相乘获得一个矩阵，之后将所有位置的矩阵相加起来组成新矩阵，接着将新矩阵拉伸成为一个向量，

最后再通过 softmax 获得融合特征用于细粒度识别。对于细粒度识别任务，传统做法有两种，第一是手动定位每个关键位置，并对这些关键位置建模提取特征，第二种是将细粒度识别当成纹理识别进行训练。然而第一种方法需要人工干预，第二种方法所需的网络较深且识别率过低，尤其是当识别对象较小时。双线性池化可以很好的解决这些缺点，既不需要手动标记，也能够用较浅的网络来达到目的，而且这种双线性的形式可以大幅简化网络梯度运算。在 CUB-200-2011 数据集中的实验结果显示，双线性模型的两个网络选择 MNet 和 DNet 时能在该数据集中取得最高精度。

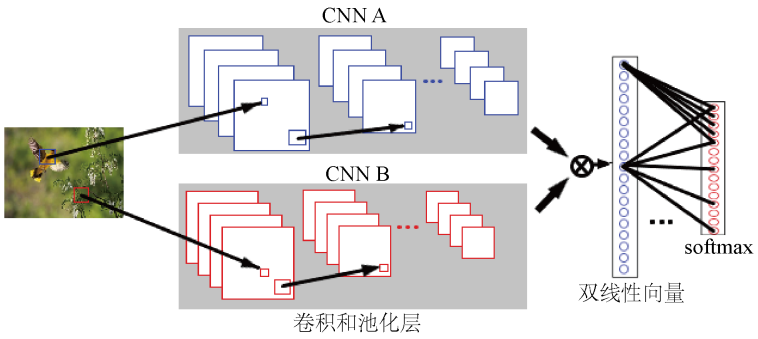

Figure 1. Bilinear CNN network model
图 1. 双线性 CNN 网络模型

(3) 协方差池化

协方差池化(Covariance Pooling) [7] [8] [9]更多被用于图像表情识别和视频表情识别中。由于表情识别任务需要捕捉的是面部关键点的形变程度而不是面部关键点是否存在，因此传统的池化方法例如最大池化平均池化这类传统池化不适用于表情识别任务，而协方差池化则更加适合提取形变特征。假设 $f_1, f_2, \cdots, f_n$ 为多个特征，协方差池化的具体计算公式为：

$$C = \frac{1}{n-1}\sum_{i=1}^{n}\left(\left(f_i - \overline{f}\right)\left(f_i - \overline{f}\right)^{\mathrm{T}}\right) \tag{1}$$

从公式中可以看出，C 为输出的协方差矩阵，其中 $\overline{f}$ 为所有特征值之和的平均数，n 为特征的数量。由于表情识别任务的网络结构是在协方差池化后还需要输入进入正定矩阵网络(SPDNet)，也就意味着协方差矩阵 C 还需要转化为正定矩阵，转化为正定矩阵 C^{+}的计算公式如下：

$$C^{+} = C + \lambda \mathrm{trace}(C) I \tag{2}$$

其中 trace(C)表示矩阵 C 的迹，λ 为正则化参数，而 I 为单位矩阵。协方差池化的使用是灵活的，首先计算出的协方差矩阵，然后可以根据各种网络结构的需求来调整矩阵，例如对矩阵进行正则化处理。协方差池化属于二阶池化方法，对比大多数一阶池化方法可以更多地捕获特征图的信息。从 RAF 数据集的结果表明，在类似于处理表情识别这类任务中，使用协方差池化这种二阶池化方法可以获得相较一阶池化方法更高的准确率。但不可忽视的是，二阶池化方法的计算量也明显要高于传统池化方法。

2.1.2. 随机性池化

(1) Lp 池化

Lp 池化(Lp Pooling) [10] [11]是一个以复杂细胞为模型的受生物学启发的池化方法，其计算方式用公式可以表示为：

$$o = \left(\frac{1}{N}\sum_{i=1}^{N} v_i^{p}\right)^{1/p} \tag{3}$$

其中 o 表示池化函数的输出，N 表示池化框的大小，v 代表这池化核内的各个元素的值，而 p 则是一个变量。当 p 取不同值的时候，池化函数也随之改变。当 $p = 1$ 时，Lp 池化对所有池化区域内的值取均值，相当于是传统的平均池化。而当 $p \to \infty$时，Lp 池化对所有池化区域取最大值，则退化为最大池化。随着 p 值从 1 开始不断增大，Lp 池化成功的实现了从平均池化逐渐转变为最大池化，是这两种池化推广的方法。大量实验表明，大多数问题的最优池化方法既不是平均池化也不是最大池化，而是介于两者之间的某种类型。其中，当 $p = 2$ 时，也就是 L2 池化，在大多数图像分类问题上可以取得较好的效果，而当 $p = 12$ 时，在 SVHN 数据集中识别图像中的数字的时候，能取得验证集中最低的错误率，其错误率相较于最大池化和平均池化低了将近一倍。由于 Lp 池化可以根据不同实验目的的需要而改变 p 值以达到最佳效果，因而 Lp 池化可以适用于大多数的卷积神经网络。

(2) 随机池化

随机池化(Stochastic Pooling) [12] [13]是 Lp 池化概念的一种延伸形式，都能将最大池化和平均池化联系起来。而与其他池化方法不同的是，随机池化用一个随机过程代替了传统的确定池化操作，根据池化区域的活动给出的多项式分布，在每个池化区域随机选择一个值激活。更具体地，首先通过规范化区域内的激活来计算每个区域的概率 p，具体的公式如下：

$$p_i = \frac{a_i}{\sum a_k}, k \in R_j \tag{4}$$

其中 R_j 是池化域，不难看出每个池化域中的元素值越大，其概率 p 也越大，之后按照概率值的大小随机选择池化域一个值作为最终值。随机池化和最大池化的不同之处在于，最大池化将百分百保留池化区域里面的最大值，而随机池化则对池化区域中的最大值赋予最大概率被选中，但也不会完全忽略掉池化区域的其他值。这种池化方式简洁直观，具有较少的计算开销且不需要引入额外的超参数，因此可以方便的使用随机池化替换其他卷积神经网络中的传统池化方法。由于随机池化具有随机性，在很大程度上保留了最大池化的优势并改善了过于武断的不足，相当于引入了一种正则化方法，可以很好的避免卷积神经网络过拟合。因此，随机池化很大程度上保留了最大池化的优势并改善了最大池化的缺点，从 CIFAR-10、CIFAR-100、MNIST 和 SVHN 等数据集的实验结果中可以看出随机池化的错误率要远低于最大池化。然而随机池化也存在一定的限制，当池化域中存在部分元素值为负数时，按公式计算其概率也是负数，这是不合理的，因而使用随机池化时要确保池化层之前的激活层中激活函数算出的值不会是负数，如果激活函数使用的是 Relu 等具有非负输出的函数则可以规避该问题。

(3) 混合池化

混合池化(Mixed Pooling) [14]通过把最大池化和平均池化进行融合，提出了一种新的映射方式。混合池化用随机采样代替确定性的池化操作，随机使用传统的最大池化和平均池化方法。混合池化的具体公式如下：

$$y_{kij} = \lambda \max\left(x_{kqp}\right) + (1-\lambda)\frac{1}{\left|R_{ij}\right|} \cdot \sum x_{kqp}\left(p,q\right) \in R_{ij} \tag{5}$$

其中，λ 是 0 或 1 的随机值，表示选择使用最大池化还是平均池化。虽然最大池化和平均池化在部分数据集上能取得好的效果，但是当遇到一个新问题时，仍然对选用哪种方法更好缺乏指导。而混合池化就是最大池化和随机池化的线性组合，该方法以随机的方式改变了池化调节方案，在一定程度上解决了最大池化和平均池化所遇到的问题。这种混合池化可以很好的和其他形式的正则化方法例如数据增强、Dropout 和权值分解等相结合来提升模型的整体性能。且这种混合池化卷积神经网络的在反向传播过程中也能根据参数 λ 的值来进行调整，不用担心因为混合了两种池化方法而导致无法反向传播。在 CIFAR-10、

CIFAR-100 和 SVHN 数据集上的实验结果表明，混合池化在解决分割问题和提高分类精度方面优于传统的最大池化和平均池化。此外，混合池化还提供了一种新的思路，可以尝试将传统的池化方法的进行线性融合，如多种池化方法进行线性组合等，在特定的模型或数据集上可能会使性能进一步提高。

(4) 通用池化

大多数池化方法的计算方式是固定的，由于不同的 CNN 要实现的目的各不相同，所使用的数据集也是可变的，因此固定的计算方式不可能在多个不同的 CNN 上取得最好的效果，而通用池化则可以根据给定的问题和数据集生成任何池化函数。通用池化(Universal Pooling) [15]的公式如下。

$$o_{0,0} = \text{universal\_pooling}\left(f_{0,0}, f_{0,1}, f_{1,0}, f_{1,1}\right) = \pi_{0,0} f_{0,0} + \pi_{0,1} f_{0,1} + \pi_{1,0} f_{1,0} + \pi_{1,1} f_{1,1} \triangleq \pi \otimes f \tag{6}$$

其中 f 表示池化块中的元素，π 表示池化权重。通用池化为每个通道选择池化权重 π，池化权重 π 是在不同信道之间分别训练的，并将它们与其他特征提取部分一起训练，其具体操作是确定特征映射的每个池块中每个元素的贡献，相当于确定池权重 π，并将$\otimes$运算符应用于池化权重和特征映射中。例如平均池化，相当于每个元素的池化权重都是 0.25，而通用池化，只要满足池化块中所有元素的权重 π 的和是 1，就能进行正确的运算。由于池化权重 π 是从给定的数据集动态学习到的，因此替换网络中原有的池化方法将会提高性能，最后在 CIFAR-10 数据集中通过对比试验表明通用池化确实能取得更好的效果。

2.2. 序有关的池化方法

2.2.1. 基于序的池化

传统的池化方法是对经过激活层激活后的值进行操作的，不同的值可能会对池化计算方法产生影响。如随机池化，若激活后的值为负数，则无法计算选中该值的概率。但在一个池化区域中，尽管值有多种取值可能，但其排序一般是相对稳定的，因此基于序的池化方法(Rank-based Pooling) [16]可以尽量避免取值变化的影响。基于序的池化方法分为三种，基于序的平均池化(RAP)、基于序的加权池化(RWP)和基于序的随机池化(RSP)，这三种池化方法的示意图如图 2 所示。

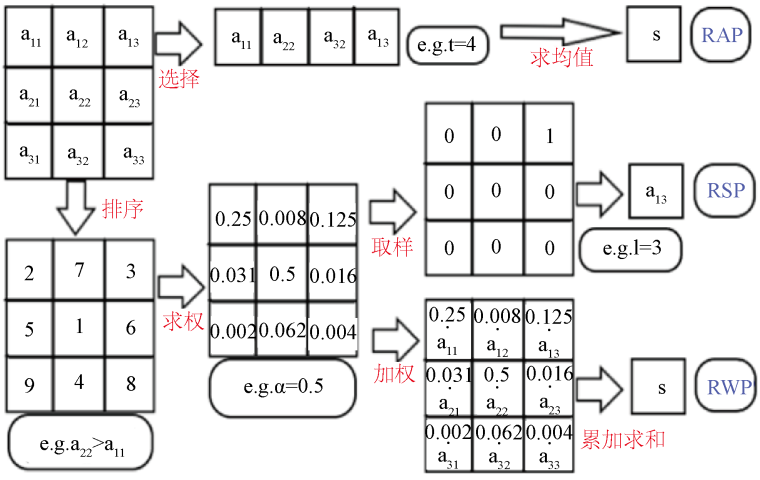

Figure 2. Order based pooling method
图 2. 基于序的池化方法

首先根据池化核内值的大小进行逆序排序：以 a 表示排序的激活，$a:T\to\{a_{\max},\cdots,a_{\min}\}$。然后，根据它们在池化框中的位置将位置坐标分配给 $a(i)$。设 r 为 $r:T\to\{1,\cdots,n\}$，其中 $r(i)$是激活 i 的序，n 表示池化区域的大小。

$$a(i) > a(j) \Rightarrow r(i) < r(j)$$

如上图所示，假定 a_{22} 为最大值，则赋予最小的序 1，a_{31} 最小则赋予最大的序 9，而若存在激活的值相等的时候，则添加如下约束。

$$a(i) = a(j) \wedge i < j \Rightarrow r(i) < r(j)$$

对于 RAP，需要额外设置一个参数 t，t 是小于 n 的正整数。RAP 的计算方式是选取池化框内最大的 t 个值，再计算出这 t 个值之和的平均数 s 作为输出。当 t 为 1 时，则退化为最大池化，而 t 为 n 时，则是平均池化，RAP 通过调整 t 来实现从最大池化到平均池化的过渡。RAP 相当于对每个值都赋予相同的 $1/n$ 权重，然而更合理的假设是每个值重要性不同，RWP 和 RSP 可以很容易地通过将较大的权重分配给较高的激活值来解决这个问题。对于 RWP 和 RSP，需要先计算出池化框中各个值的权重 p_r，具体公式为：

$$P_r = \alpha(1-\alpha)^{r-1} \quad r = 1, \cdots, n \tag{7}$$

其中 α 是一个超参数，且 $0 < \alpha < 1$，这样设计权重公式是为了保证$\sum P_r = 1$。RSP 是根据概率 p 随机选择一个值作为输出，而 RWP 则是通过加权求和的方式计算输出 s，具体公式为：

$$s = \sum p_i a_i \tag{8}$$

RWP 和 RSP 可以根据数据集的不同来调整 α 的大小，对于大多数数据集，α 设置为 0.5 左右时可以达到最好效果。在 CIFAR-10、CIFAR-100 和 MNIST 数据集上的实验结果表明，基于序的池化方法在多个数据集中的表现要明显优于传统池化方法。

2.2.2. 最大池化的改进池化方法

最大池化是最常用的池化方法之一，可以提取特征图中关键的特征，但是代价是完全忽略了其他特征，因此衍生出了很多基于最大池化的改进池化方法来弥补最大池化的不足。K-Max Pooling 和 Chunk-Max Pooling 大多用在自然语言处理中，其中 K-Max Pooling 用来提取卷积层后一系列特征值中前 k 个最大值，并且保留这些特征值的先后顺序，其中当 $k = 1$ 时则退化为最大池化。K-Max Pooling 可以根据需求调整 k 值，能比最大池化保留更多信息。Chunk-Max Pooling 则是先把卷积层后的所有特征向量进行分段，之后从每个分段中取最大值作为输出，最后同样能保留多个特征值。划分段落的方法既可以事先设置，也可以根据特征值动态划分。

广义最大池化(Generalized Max Pooling) [17] [18]可以视为最大池化的泛化，可以根据全局描述符相似度来平衡频繁描述符和稀有描述符的影响。在很多细粒度识别的任务中，例如对鸟的种类进行分类，由于大部分鸟图中都有大量的树叶构成频繁描述符，而鸟类之间的区别的关键之处，例如鸟喙，在图片中占比过小而构成稀有描述符，这些无用的频繁描述符将对关键的稀有描述符造成较大影响。广义最大值池化是通过求解最优化问题来获得权重的过程，而权重则根据每个元素和其他元素的相似度来获得，相似度较低的稀有描述符则会有较大的权重，最终把每个元素进行加权求和作为输出。

分数最大池化(Fractional Max Pooling) [19]是另一种特殊形式的最大池化。最大池化的池化框大小是 $n \times n$，其中 n 一般为 2，而分数最大池化则允许 n 为非整数值，这样可以避免 n 为整数时由于池化核不相交而产生性能受到限制的问题。根据输出的特征图的尺寸大小 $a \times a$，将输入的尺寸为 $b \times b$ 的特征图平分成 $a \times a$ 块，并对每个块做最大池化的计算。在具体计算中，由于 $1 < (b/a) < 2$，则固定间隔为 1 或 2，且由于输入和输出特征图的尺寸固定，因此 1 和 2 的数量也固定，接着使用随机或伪随机的方式产生两个由 1 和 2 组成的序列，可以得知序列中一共有 a 项，这 a 项 1 和 2 的累加和为 b。由于序列组成的随机性，分数最大池化减少了对各种数据集的过拟合。

3. 池化核大小不固定的池化方法

3.1. 重叠池化

传统的池化方法池化区域相互间不会重叠，与此不同，重叠池化(Overlapping Pooling) [20]则使用重叠的模式来选择池化区域。更准确地说，池化层可以被认为是由间隔 s 个像素的池单元组成的网格，每个单元汇总以池化单元的位置为中心的 $z \times z$ 大小的邻域。如果设置 $s = z$，将获得卷积神经网络中常用的传统池化，而设置成 $s < z$ 时，则会获得重叠池化。大多数池化都会把步长 s 设置为 2，而池化框 z 也同样设置为 2，这样可以使得在卷积神经网络正向或反向传播时更容易计算。但是当特征图的分辨率较大时，大多数池化方法不能快速降维，若是通过提高网络层数以达到降维目的又会导致参数过多，增加过拟合的风险。毕竟网络层数并不一定是越多越好，过多的网络层会使特征图维度骤降，而重叠池化相较传统的池化方法在同样数量的网络层中可以达到保留更多特征图维度的效果。在 ILSVRC 数据集的实验表明，保持步长 s 为 2 不变，而将池化框设置为 3 时，与之前相比 top 1 和 top 5 的错误率分别下降了 0.3% 和 0.4%，且从网络训练过程可以看出，重叠池化可以更好的避免过拟合。重叠池化虽然可以更快降维，但是在设置步长和池化框大小时也需要慎重，若设置不合理，则可能因为降维过快导致网络精确率大幅降低。

3.2. 空间金字塔池化

对于大多数卷积神经网络而言，都有着一个限制，那就是输入的图像需要是固定尺寸(224 × 224)。为很好地解决这一限制。空间金字塔池化(Spatial Pyramid Pooling) [21] [22]提出一种多尺度池化方法，这种池化方法可以处理图片中不同尺度的信息，按照三个不同的尺度对一张输入图片或特征进行划分，具体的示意图如图 3 所示。

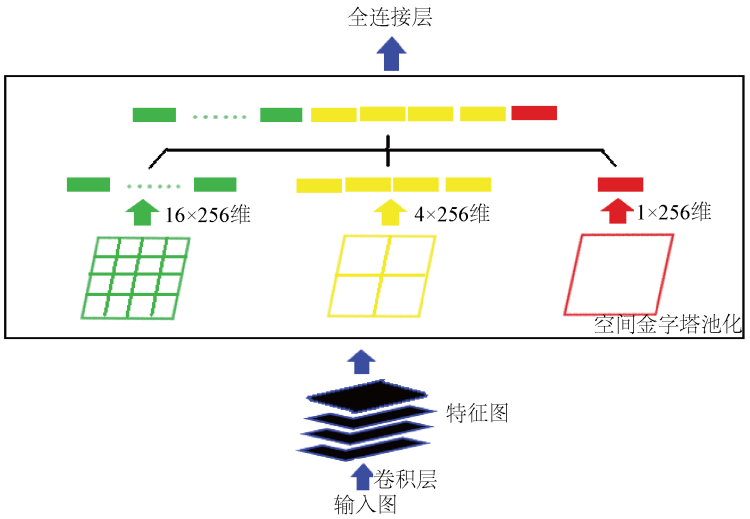

Figure 3. Spatial pyramid pooling
图 3. 空间金字塔池化

从示意图中可以看出，把一个图片分别经过三个不同的池化窗口最终得到了 1 × 1，2 × 2，4 × 4 一共 21 个池化结果，将这 21 个特征连接起来再输入到下一层。通常空间金字塔池化的池化窗口的步长都会设置为$\lfloor a/n \rfloor$，而池化框大小设置为$\lceil a/n \rceil$，其中 a 为特征图大小，而 n 指一个池化需要得到 $n \times n$ 个池

化结果，上图的例子中 n 分别为 1、2 和 4。不难看出，无论特征图大小 a 如何改变，只要固定了 n 的值，最终都会得到固定数量的特征，进而可以正常的传递给全连接层。空间金字塔池化使用起来非常的灵活，既可以用于图像分类也可以用于目标检测，根据网络的不同需求来调整 n 值。大多数网络中 n 都会保留 1 和 2 这两个值，而最后一个值 n 大多情况取 3 或者 4，但也可以取更大的正整数。空间金字塔池化由于可以处理不同尺度的图像，因此使用非常灵活，并可以有效防止过拟合。从 Pascal VOC 数据集的实验结果来看，相比于固定大小输入的图像，多尺度图像的输入可以使网络更好的收敛。但空间金字塔池化也有少数情况不能计算，例如 a 为 14，n 为 4 的时候，此时池化框为 5，步长为 4，这样会导致小部分特征图未被池化计算，因此设置 n 的时候也要考虑到实际情况的限制。

3.3. 全局最大/平均池化

传统的卷积神经网络都会包含全连接层，而全连接层过多的参数会严重影响到网络训练的速度，同时也容易导致过拟合。为了解决全连接层导致的一系列问题，NIN 提出了全局池化。全局池化[23]分为全局平均池化(Global Average Pooling)和全局最大池化(Global Max Pooling)。全局平均池化和全局最大池化的计算方法和传统的平均池化和最大池化类似，区别在于池化框大小设置成和整个特征图的尺寸相同。这样就把每个 $w \times h \times c$ 的特征图转化为 $1 \times 1 \times c$ 即 c 个通道的输出，作为最终提取的分类或回归特征。全局池化对于输入图像的尺寸大小没有要求，因此使用起来更加灵活。其中，全局最大池化由于提取整个特征图中的最大值，因此更易受噪声的影响，且从消融实验的结果也显示其错误率要略高于全局平均池化，因而大多模型选择全局平均池化。全局平均池化的优点在于通过增强特征映射和类别之间的对应关系，使得其更适合卷积结构。另一个优点是在全局平均池化中没有需要优化的参数，因而相对于全连接来说更有利于避免过拟合。此外，全局平均池化将空间信息进行归并，从而对输入的空间平移更具鲁棒性。从 CIFAR-10、CIFAR-100、MNIST 和 SVHN 等数据集的实验结果来看，全局平均池化代替了全连接层之后，测试集的错误率有了明显下降。即使全连接层加上了 dropout，错误率依旧高于全局平均池化。自从全局平均池化方法提出后，深度卷积神经网络模型基本都用其替代了全连接层。

3.4. 多尺度无序池化

传统的卷积神经网络(CNN)对整张输入图进行卷积和激活后都会获取许多特征，而全局卷积神经网络激活后获得的特征会缺少几何不变性，几何不变性包括旋转不变性、平移不变性等，导致了 CNN 对图片的旋转、缩小和平移相当敏感，这限制了它们对可变场景的分类和匹配的鲁棒性。而多尺度无序池化(Multi-Scale Orderless Pooling) [24]的提出，就是为了在不降低卷积神经网络的分辨能力的前提下去提高几何不变性。多尺度无序池化的示意图如图 4 所示：

从图中可以看出，多尺度无序池化共有三个 level，每个 level 用于提取不同尺度的特征。对于第一个 level，需要对整个图像缩放至 256 × 256 像素，首先把经过卷积层的特征图中的每个像素值都减去像素值的平均值，对全局进行 Relu 激活后，最后取全连接层的 4096 维的特征作为 level 1 特征。对于第二个 level，将池化框设为 128 × 128，以 32 像素作为步长进行计算，维度也是 4096。为了提高计算效率，把每次通过滑动窗口池化计算提取的卷积神经网络特征聚集起来后，通过主成分分析法(PCA)将维度从 4096 降低至 500。接着采用 k 均值算法，将 k 设置为 100，并且使用局部聚合向量(VLAD)对多个 500 维的特征进行编码，生成 50000 维的特征。接着再利用 PCA 把维度恢复成 4096，最终这 4096 维的特征就是 level 2 的特征。Level 3 的计算过程和 level 2 一样，区别在于池化框设为 64 × 64，最后把三个 level 的特征拼接起来作为输出特征。在 SUN397 和 ILSVRC 数据集中的实验结果证明，多尺度无序池化提取出的一般特征均可以用于图像分类这等有监督任务或实例检索等无监督识别任务，其表现明显优于传统池化方法。

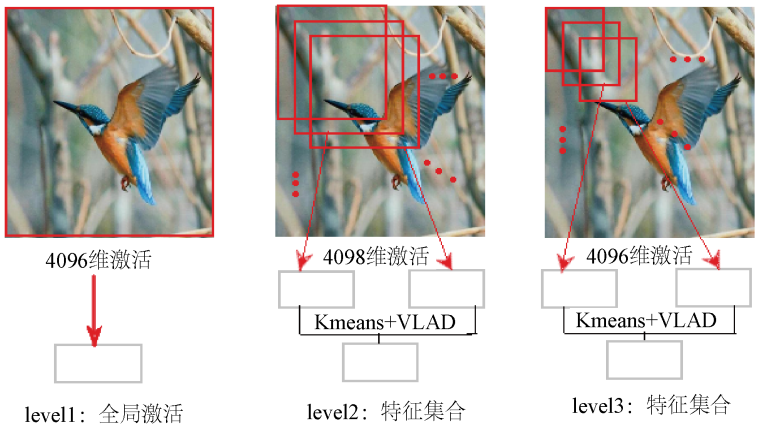

Figure 4. Multi-scale disordered pooling
图 4. 多尺度无序池化

3.5. 感兴趣区域池化

感兴趣区域池化(Region of Interest Pooling) [25]首先被用在了 Fast RCNN [26]中，输入的图片经过卷积层后获得特征图，特征图再通过 RPN 算法获得多个目标的候选框，其中候选框中的区域就是感兴趣区域。网络结构图如图 5 所示：

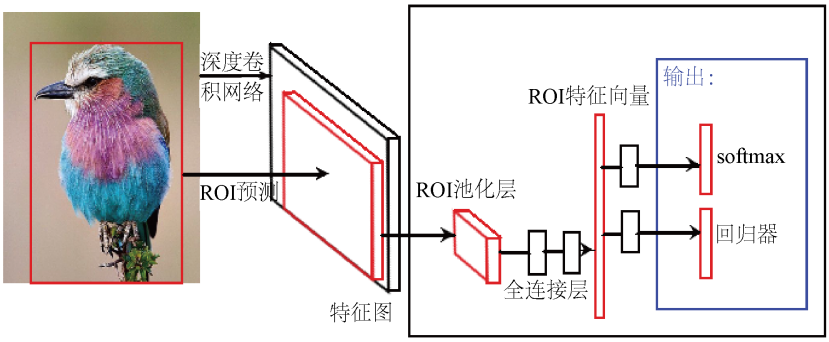

Figure 5. ROI pooling network structure
图 5. ROI 池化网络结构图

上图中只有一个候选框，大多时候会有多个候选框，候选框中的往往是图片中的关键目标。感兴趣区域池化将任何大小的有效感兴趣区域内的特征图转换为固定尺寸为 $h \times w$ 的小特征图，其中 h 和 w 是独立于任何特定感兴趣区域池化层的超参数。这也意味着经过感兴趣区域池化后输出的特征图尺度是固定的，而输入的特征图的尺寸却是可变的。假设输入的感兴趣区域内的一个特征图的尺寸大小为 $H \times W$，为了保证能获得固定尺寸 $h \times w$ 的输出，则需要把该感兴趣区域划分成多个尺寸为 $(H/h) \times (W/w)$ 的子网格，感兴趣区域池化的计算方法和最大池化一样，因此每个子网格都保留该网格中的最大值。若 H/h 或 W/w 不为整数，则采用量化操作，去除小数点后面的数字。该池化方法的好处在于可以根据需求任意指定输出尺寸的大小而不用对原图进行缩放，且只需要对原图中感兴趣的部分进行计算，从而很大程度上减少了计算量。但由于会采用量化操作，将不可避免的导致在计算过程中候选框的位置会出现偏差。从 VOC 系列的数据集实验结果上看，添加了感兴趣区域池化的 Fast RCNN 相较之前的 RCNN，测试和训练的速度都大幅提升，且在多个数据集上，目标检测的识别率也有所上升。

4. 实验对比分析

为了进行有效的对比分析，这些主要评估数据集所使用的网络都由三个卷积层、5 × 5 滤波器和 64 个特征映射组成，三个池化层分别在三个卷积层之后。三个池化层中的每一层使用 3 × 3 池化框且步长设为 2。最后，使用一个具有 softmax 输出的完全连接层来组成网络模型。将此网络模型应用于两个不同的数据集：CIFAR-10 [27]、CIFAR-100 [27]。

4.1. 主要评估数据集

现在卷积神经网络训练所使用的数据集有很多，多个数据集的区别主要包括数据的种类和数量。第一部分介绍的各种池化方法对应的数据集主要包括 SVHN、CIFAR-10、CIFAR-100、MNIST 和 Pascal VOC 等，由于每个池化方法都存在于不同的 CNN 中，而各种 CNN 所要达到的目的各不相同，相应的使用的数据集也都各不一样，这导致很多池化方法的性能无法相互比较，因此只介绍几个用的较多的数据集来比较各种池化方法的效果。本部分将对 CIFAR-10 和 CIFAR-100 两个数据集进行介绍，并给出部分模型在这些数据集上的性能。

CIFAR-10 数据集共有六万张 32 × 32 的彩色图像，这些彩色图像分为 10 个类，每个类包含六千张图像。这六万张彩色图像又分为 6 个批次，每个批次一万张，其中五个批次是是训练图像共计五万张，另外一个批次的一万张图像是测试图像。需要注意的是，每个批次中每类图像的数量都是不一定相同的。

CIFAR-100 数据集和 CIFAR-10 数据集类似，同样包含六万张 32 × 32 的彩色图像并平分给六个批次，五个批次是训练图像。与 CIFAR-10 不同之处在于共分为 100 个类，每个类包含六百张图像，且这 100 个类被平分成 20 个超类，因此每张图像都包含一个粗糙标签(所属的超类)和一个精细标签(所属的类)。

4.2. 数据集上不同池化方法的性能比较

在上述网络模型中，将每种池化方法都应用于三个池化层中进行性能对比。对于多个不同的池化方法，将分别在 CIFAR-10 和 CIFAR-100 这两个数据集上的性能表现展现出来，如表 1 所示。

Table 1. Test error of pooling method on dataset
表 1. 池化方法在数据集上的 Test error

池化方法	CIFAR-10 Test Error %	CIFAR-100 Test Error %
最大池化	19.40	50.90
平均池化	19.24	47.77
随机池化	15.13	42.51
谱池化	**8.60**	**31.60**
RAP	17.97	45.66
RWP	18.91	46.69
RSP	13.84	43.91
全局池化	10.41	35.68

从表 1 的实验结果可以看出，各种新的池化方法相较于最大池化和平均池化在性能方面均有提升，其中谱池化则获得了 CIFAR-10 和 CIFAR-100 这两个数据集中的最佳性能。

5. 总结

随着卷积神经网络的不断发展，卷积神经网络在目标检测，语义分割等多个任务上也都取得了很大

程度的改进，这些改进很多都归因于卷积神经网络中池化层中的池化方法的不断优化。每年都有新的池化方法被提出，相较于传统的池化方法，新的池化方法运用在卷积神经网络中可以起到在训练和测试速度上提高，在准确率上提高，减少网络中的计算量和减少网络过拟合等作用。随着数据集的不断扩展，网络设备的不断优化，会有更多新的池化方法出现。本文重点把各种新的池化方法与传统池化方法进行对比，解释了新的池化方法产生改进的原因，并介绍了不同池化方法的具体计算步骤以及需要注意的地方，另外也介绍了数据集以及这些新方法在不同数据集上的性能。

基金项目

北京市教委科技计划一般项目(KM201811232024)；北京信息科技大学促进高校内涵发展“信息+”项目－多源光谱生物特征活体识别平台建设；北京信息科技大学高教研究重点项目(2019GJZD01)。

参考文献

[1] Rippel, O., Snoek, J. and Adams, R.P. (2015) Spectral Representations for Convolutional Neural Networks. *Advances in Neural Information Processing Systems* 28 (*NIPS* 2015), 2449-2457.

[2] Zhang, H. and Ma, J. (2018) Hartley Spectral Pooling for Deep Learning. arXiv Preprint arXiv: 1810.04028.

[3] Lin, T.Y., Roychowdhury, A. and Maji, S. (2015) Bilinear CNN Models for Fine-Grained Visual Recognition. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 1449-1457. https://doi.org/10.1109/ICCV.2015.170

[4] Li, X., Yang, C., Chen, S., *et al*. (2019) Semantic Bilinear Pooling for Fine-Grained Recognition. arXiv:1904.01893.

[5] Kek, X.Y., Chin, C.S. and Li, Y. (2019) Acoustic Scene Classification Using Bilinear Pooling on Time-Liked and Frequency-Liked Convolution Neural Network. 2019 *IEEE Symposium Series on Computational Intelligence* (*SSCI*), Xiamen, 6-9 December 2019, 3189-3194. https://doi.org/10.1109/SSCI44817.2019.9003150

[6] Gao, Y., Beijbom, O., Zhang, N. and Darrell, T. (2016) Compact Bilinear Pooling. 2016 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, 27-30 June 2016, 317-326. https://doi.org/10.1109/CVPR.2016.41

[7] Li, P., Xie, J., Wang, Q., *et al*. (2018) Towards Faster Training of Global Covariance Pooling Networks by Iterative Matrix Square Root Normalization. 2018 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, 18-23 June 2018, 947-955. https://doi.org/10.1109/CVPR.2018.00105

[8] Acharya, D., Huang, Z., Panipaudel, D., *et al*. (2018) Covariance Pooling for Facial Expression Recognition. 2018 *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (*CVPRW*), Salt Lake City, 18-22 June 2018, 480-4807. https://doi.org/10.1109/CVPRW.2018.00077

[9] Ionescu, C., Vantzos, O. and Sminchisescu, C. (2015) Matrix Backpropagation for Deep Networks with Structured Layers. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 2965-2973. https://doi.org/10.1109/ICCV.2015.339

[10] Sermanet, P., Chintala, S. and LeCun, Y. (2012) Convolutional Neural Networks Applied to House Numbers Digit Classification. *Proceedings of the* 21*st International Conference on Pattern Recognition* (*ICPR*2012), 3288-3291.

[11] Hyvärinen, A. and Köster, U. (2007) Complex Cell Pooling and the Statistics of Natural Images. *Network*: *Computation in Neural Systems*, **18**, 81-100. https://doi.org/10.1080/09548980701418942

[12] Zeiler, M.D. and Fergus, R. (2013) Stochastic Pooling for Regularization of Deep Convolutional Neural Networks arXiv Preprint arXiv: 1301.3557.

[13] Zhai, S., Wu, H., Kumar, A., *et al*. (2017) S3pool: Pooling with Stochastic Spatial Sampling. 2017 *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), Honolulu, 21-26 July 2017, 4003-4011. https://doi.org/10.1109/CVPR.2017.426

[14] Yu, D., Wang, H., Chen, P., *et al*. (2014) Mixed Pooling for Convolutional Neural Networks. In: Miao, D., Pedrycz, W., Ślęzak, D., Peters, G., Hu, Q. and Wang, R., Eds., *International Conference on Rough Sets and Knowledge Technology*, Springer, Cham, 364-375. https://doi.org/10.1007/978-3-319-11740-9_34

[15] Hyun, J., Seong, H. and Kim, E. (2019) Universal Pooling—A New Pooling Method for Convolutional Neural Networks. arXiv:1907.11440.

[16] Shi, Z.L., Ye, Y.D. and Wu, Y.P. (2016) Rank-Based Pooling for Deep Convolutional Neural Networks. *Neural Networks*, **83**, 21-31.

[17] Murray, N. and Perronnin, F. (2014) Generalized Max Pooling. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, 23-28 June 2014, 2473-2480.

[18] Christlein, V., Spranger, L., Seuret, M., *et al*. (2019) Deep Generalized Max Pooling. 2019 *International Conference on Document Analysis and Recognition* (*ICDAR*), Sydney, 20-25 September 2019, 1090-1096.

[19] Graham, B. (2014) Fractional Max-Pooling. arXiv:1412.6071.

[20] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) Imagenet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*, 1097-1105.

[21] He, K., Zhang, X., Ren, S., *et al*. (2015) Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37**, 1904-1916. https://doi.org/10.1109/TPAMI.2015.2389824

[22] Asgari, R., Waldstein, S., Schlanitz, F., *et al*. (2019) U-Net with Spatial Pyramid Pooling for Drusen Segmentation in Optical Coherence Tomography. In: Fu, H., Garvin, M., MacGillivray, T., Xu, Y. and Zheng, Y., Eds., *International Workshop on Ophthalmic Medical Image Analysis*, Springer, Cham, 77-85. https://doi.org/10.1007/978-3-030-32956-3_10

[23] Lin, M., Chen, Q. and Yan, S. (2013) Network in Network. arXiv Preprint arXiv:1312.4400.

[24] Gong, Y., Wang, L., Guo, R., *et al*. (2014) Multi-Scale Orderless Pooling of Deep Convolutional Activation Features. In: Fleet, D., Pajdla, T., Schiele, B. and Tuytelaars, T., Eds., *Computer Vision—ECCV* 2014. *ECCV* 2014. *Lecture Notes in Computer Science*, Springer, Cham, 392-407. https://doi.org/10.1007/978-3-319-10584-0_26

[25] Sun, Y.X., Sun, C., Wang, D., *et al*. (2019) ROI Pooled Correlation Filters for Visual Tracking. 2019 *IEEE*/*CVF Conference on Computer Vision and Pattern Recognition* (*CVPR*), Long Beach, 15-20 June 2019, 5776-5784. https://doi.org/10.1109/CVPR.2019.00593

[26] Girshick, R. (2015) Fast R-CNN. 2015 *IEEE International Conference on Computer Vision* (*ICCV*), Santiago, 7-13 December 2015, 1440-1448.https://doi.org/10.1109/ICCV.2015.169

[27] Krizhevsky, A. and Hinton, G. (2009) Learning Multiple Layers of Features from Tiny Images. 7.