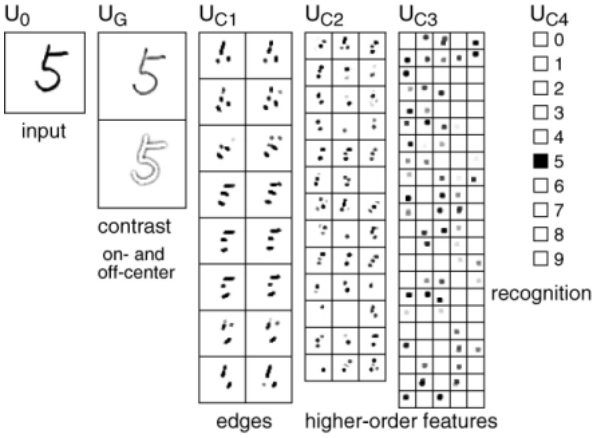
卷积神经网络

卷积神经网络（Convolutional Neural Networks, CNN）是一类包含卷积计算且具有深度结构的前馈神经网络（Feedforward Neural Networks），是深度学习（deep learning）的代表算法之一[1-2]。卷积神经网络具有表征学习（representation learning）能力，能够按其阶层结构对输入信息进行平移不变分类（shift-invariant classification），因此也被称为“平移不变人工神经网络（Shift-Invariant Artificial Neural Networks, SIANN）”[3]。
对卷积神经网络的研究始于二十世纪80至90年代，时间延迟网络和LeNet-5是最早出现的卷积神经网络[4]；在二十一世纪后，随着深度学习理论的提出和数值计算设备的改进，卷积神经网络得到了快速发展，并被应用于计算机视觉、自然语言处理等领域[2]。
卷积神经网络仿造生物的视知觉（visual perception）机制构建，可以进行监督学习和非监督学习，其隐含层内的卷积核参数共享和层间连接的稀疏性使得卷积神经网络能够以较小的计算量对格点化（grid-like topology）特征，例如像素和音频进行学习、有稳定的效果且对数据没有额外的特征工程（feature engineering）要求[1-2]。

历史

编辑 语音



neocognitron的构筑与特征可视化[6]
对卷积神经网络的研究可追溯至日本学者福岛邦彦（Kunihiko Fukushima）提出的neocognitron模型。在其1979[7-8]和1980年[9]发表的论文中，福岛仿造生物的视觉皮层（visual cortex）设计了以“neocognitron”命名的神经网络。
neocognitron是一个具有深度结构的神经网络，并且是最早被提出的深度学习算法之一[10]，其隐含层由S层（Simple-layer）和C层（Complex-layer）交替构成。其中S层单元在感受野（receptive field）内对图像特征进行提取，C层单元接收和响应不同感受野返回的相同特征[9]。neocognitron的S层-C层组合能够进行特征提取和筛选，部分实现了卷积神经网络中卷积层（convolution layer）和池化层（pooling layer）的功能，被认为是启发了卷积神经网络的开创性研究[11]。
第一个卷积神经网络是1987年由Alexander Waibel等提出的时间延迟网络（Time Delay Neural Network, TDNN）[12]。TDNN是一个应用于语音识别问题的卷积神经网络，使用FFT预处理的语音信号作为输入，其隐含层由2个一维卷积核组成，以提取频率域上的平移不变特征[13]。由于在TDNN出现之前，人工智能领域在反向传播算法（Back-Propagation, BP）的研究中取得了突破性进展[14]，因此TDNN得以使用BP框架内进行学习。在原作者的比较试验中，TDNN的表现超过了同等条件下的隐马尔可夫模型（Hidden Markov Model, HMM），而后者是二十世纪80年代语音识别的主流算法[13]。
1988年，Wei Zhang提出了第一个二维卷积神经网络：平移不变人工神经网络（SIANN），并将其应用于检测医学影像[3]。独立于Zhang (1988)，Yann LeCun在1989年同样构建了应用于计算机视觉问题的卷积神经网络，即LeNet的最初版本[5]。LeNet包含两个卷积层，2个全连接层，共计6万个学习参数，规模远超TDNN和SIANN，且在结构上与现代的卷积神经网络十分接近[11]。LeCun (1989)[5]对权重进行随机初始化后使用了随机梯度下降（Stochastic Gradient Descent, SGD）进行学习，这一策略被其后的深度学习研究所保留。此外，LeCun (1989)在论述其网络结构时首次使用了“卷积”一词[5]，“卷积神经网络”也因此得名。
LeCun (1989)[5]的工作在1993年由贝尔实验室（AT&T Bell Laboratories）完成代码开发并被部署于NCR（National Cash Register Coporation）的支票读取系统[11]。但总体而言，由于数值计算能力有限、学习样本不足，加上同一时期以支持向量机（Support Vector Machine, SVM）为代表的核学习（kernel learning）方法的兴起，这一时期为各类图像处理问题设计的卷积神经网络停留在了研究阶段，应用端的推广较少[2]。
在LeNet的基础上，1998年Yann LeCun及其合作者构建了更加完备的卷积神经网络LeNet-5并在手写数字的识别问题中取得成功[15]。LeNet-5沿用了LeCun (1989)的学习策略并在原有设计中加入了池化层对输入特征进行筛选[15]。LeNet-5及其后产生的变体定义了现代卷积神经网络的基本结构，其构筑中交替出现的卷积层-池化层被认为能够提取输入图像的平移不变特征[16]。LeNet-5的成功使卷积神经网络的应用得到关注，微软在2003年使用卷积神经网络开发了光学字符读取（Optical Character Recognition, OCR）系统[17]。其它基于卷积神经网络的应用研究也得到展开，包括人像识别[18]、手势识别[19]等。
在2006年深度学习理论被提出后[20]，卷积神经网络的表征学习能力得到了关注，并随着数值计算设备的更新得到发展[2]。自2012年的AlexNet[21]开始，得到GPU计算集群支持的复杂卷积神经网络多次成为ImageNet大规模视觉识别竞赛（ImageNet Large Scale Visual Recognition Challenge, ILSVRC）[22]的优胜算法，包括2013年的ZFNet[23]、2014年的VGGNet、GoogLeNet[24]和2015年的ResNet[25]。

结构

编辑 语音

输入层

卷积神经网络的输入层可以处理多维数据，常见地，一维卷积神经网络的输入层接收一维或二维数组，其中一维数组通常为时间或频谱采样；二维数组可能包含多个通道；二维卷积神经网络的输入层接收二维或三维数组；三维卷积神经网络的输入层接收四维数组[16]。由于卷积神经网络在计算机视觉领域应用较广，因此许多研究在介绍其结构时预先假设了三维输入数据，即平面上的二维像素点和RGB通道。
与其它神经网络算法类似，由于使用梯度下降算法进行学习，卷积神经网络的输入特征需要进行标准化处理。具体地，在将学习数据输入卷积神经网络前，需在通道或时间/频率维对输入数据进行归一化，若输入数据为像素，也可将分布于
$[0, 255]$
的原始像素值归一化至
$[0, 1]$
区间[16]。输入特征的标准化有利于提升卷积神经网络的学习效率和表现[16]。

隐含层

卷积神经网络的隐含层包含卷积层、池化层和全连接层3类常见构筑，在一些更为现代的算法中可能有Inception模块、残差块（residual block）等复杂构筑。在常见构筑中，卷积层和池化层为卷积神经网络特有。卷积层中的卷积核包含权重系数，而池化层不包含权重系数，因此在文献中，池化层可能不被认为是独立的层。以LeNet-5为例，3类常见构筑在隐含层中的顺序通常为：输入-卷积层-池化层-全连接层-输出。

卷积层（convolutional layer）

1. 卷积核（convolutional kernel）

卷积层的功能是对输入数据进行特征提取，其内部包含多个卷积核，组成卷积核的每个元素都对应一个权重系数和一个偏差量（bias vector），类似于一个前馈神经网络的神经元（neuron）。卷积层内每个神经元都与前一层中位置接近的区域的多个神经元相连，区域的大小取决于卷积核的大小，在文献中被称为“感受野（receptive field）”，其含义可类比视觉皮层细胞的感受野 [2] 。卷积核在工作时，会有规律地扫过输入特征，在感受野内对输入特征做矩阵元素乘法求和并叠加偏差量 [1] ：

一维和二维卷积运算示例 [26]

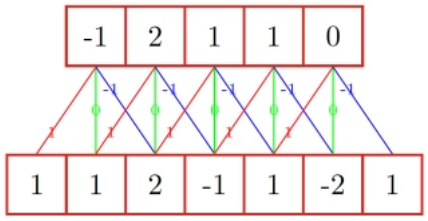


图 5.1 一维卷积示例

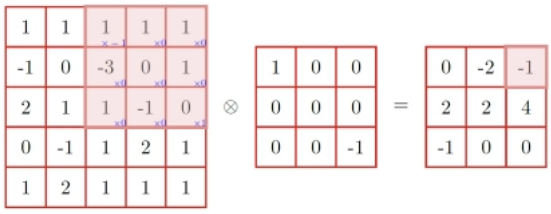


图 5.2 二维卷积示例

$$\boldsymbol{Z}^{l+1}(i,j)=\left[\boldsymbol{Z}^{l}\otimes\boldsymbol{w}^{l+1}\right](i,j)+\boldsymbol{b}=\sum_{k=1}^{K_l}\sum_{x=1}^{f}\sum_{y=1}^{f}\left[\boldsymbol{Z}_k^{l}(s_0 i+x,s_0 j+y)\,\boldsymbol{w}_k^{l+1}(x,y)\right]+\boldsymbol{b}$$

式中的求和部分等价于求解一次交叉相关（cross-correlation）。

$\boldsymbol{b}$ 为偏差量，

$\boldsymbol{Z}^{l}$ 和 $\boldsymbol{Z}^{l+1}$ 表示第 $l+1$ 层的卷积输入和输出，也被称为特征图（feature map），

L_{l+1} 为 $\boldsymbol{Z}_{l+1}$ 的尺寸，这里假设特征图长宽相同。

$\boldsymbol{Z}(i,j)$ 对应特征图的像素，

K 为特征图的通道数，

f、s_0 和 p 是卷积层参数，对应卷积核大小、卷积步长（stride）和填充（padding）层数 [1] 。

$$(i,j)\in\{0,1,\dots L_{l+1}\}\quad L_{l+1}=\frac{L_l+2p-f}{s_0}+1$$

上式以二维卷积核作为例子，一维或三维卷积核的工作方式与之类似。理论上卷积核也可以先翻转180度，再求解交叉相关，其结果等价于满足交换律的线性卷积（linear convolution），但这样做在增加求解步骤的同时并不能为求解参数取得便利，因此线性卷积核使用交叉相关代替了卷积 [4] [16] 。

特殊地，当卷积核是大小 $f=1$，步长 $s_0=1$ 且不包含填充的单位卷积核时，卷积层内的交叉相关计算等价于矩阵乘法，并由此在卷积层间构建了全连接网络 [2] ：

$$\boldsymbol{Z}^{l+1}=\sum_{k=1}^{K_l}\sum_{i=1}^{L}\sum_{j=1}^{L}\left(\boldsymbol{Z}_{i,j,k}^{l}\boldsymbol{w}_k^{l+1}\right)+\boldsymbol{b}=\boldsymbol{w}_{l+1}^{T}\boldsymbol{Z}_{l+1}+\boldsymbol{b},\quad L^{l+1}=L$$

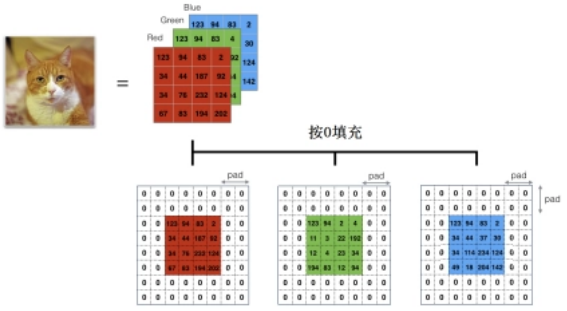
由单位卷积核组成的卷积层也被称为网中网（Network-In-Network, NIN）或多层感知器卷积层（multilayer perceptron convolution layer, mlpconv）[27] 。单位卷积核可以在保持特征图尺寸的同时减少图的通道数从而降低卷积层的计算量。完全由单位卷积核构建的卷积神经网络是一个包含参数共享的多层感知器（Muti-Layer Perceptron, MLP） [27] 。

在线性卷积的基础上，一些卷积神经网络使用了更为复杂的卷积，包括平铺卷积（tiled convolution）、反卷积（deconvolution）和扩张卷积（dilated convolution） [2] 。平铺卷积的卷积核只扫过特征图的一部份，剩余部分由同层的其它卷积核处理，因此卷积层间的参数仅被部分共享，有利于神经

网络捕捉输入图像的旋转不变（shift-invariant）特征 [28] 。反卷积或转置卷积（transposed convolution）将单个的输入激励与多个输出激励相连接，对输入图像进行放大。由反卷积和向上池化层（up-pooling layer）构成的卷积神经网络在图像语义分割（semantic segmentation）领域有应用 [29] ，也被用于构建卷积自编码器（Convolutional AutoEncoder, CAE） [30] 。扩张卷积在线性卷积的基础上引入扩张率以提高卷积核的感受野，从而获得特征图的更多信息 [31] ，在面向序列数据使用时有利于捕捉学习目标的长距离依赖（long-range dependency）。使用扩张卷积的卷积神经网络主要被用于自然语言处理（Natrual Language Processing, NLP）领域，例如机器翻译 [31] 、语音识别 [32] 等。

2. 卷积层参数

卷积核中RGB图像的按0填充 [16]



卷积层参数包括卷积核大小、步长和填充，三者共同决定了卷积层输出特征图的尺寸，是卷积神经网络的超参数 [1] 。其中卷积核大小可以指定为小于输入图像尺寸的任意值，卷积核越大，可提取的输入特征越复杂 [1] 。

卷积步长定义了卷积核相邻两次扫过特征图时位置的距离，卷积步长为1时，卷积核会逐个扫过特征图的元素，步长为n时会在下一次扫描跳过n-1个像素 [33] 。

由卷积核的交叉相关计算可知，随着卷积层的堆叠，特征图的尺寸会逐步减小，例如16×16的输入图像在经过单位步长、无填充的5×5的卷积核后，会输出12×12的特征图。为此，填充是在特征图通过卷积核之前人为增大其尺寸以抵消计算中尺寸收缩影响的方法。常见的填充方法为按0填充和重复边界值填充（replication padding）。填充依据其层数和目的可分为四类 [33] ：

- 有效填充（valid padding）：即完全不使用填充，卷积核只允许访问特征图中包含完整感受野的位置。输出的所有像素都是输入中相同数量像素的函数。使用有效填充的卷积被称为“窄卷积（narrow convolution）”，窄卷积输出的特征图尺寸为(L-f)/s+1。
- 相同填充/半填充（same/half padding）：只进行足够的填充来保持输出和输入的特征图尺寸相同。相同填充下特征图的尺寸不会缩减但输入像素中靠近边界的部分相比于中间部分对于特征图的影响更小，即存在边界像素的欠表达。使用相同填充的卷积被称为“等长卷积（equal-width convolution）”。
- 全填充（full padding）：进行足够多的填充使得每个像素在每个方向上被访问的次数相同。步长为1时，全填充输出的特征图尺寸为L+f-1，大于输入值。使用全填充的卷积被称为“宽卷积（wide convolution）”
- 任意填充（arbitrary padding）：介于有效填充和全填充之间，人为设定的填充，较少使用。

带入先前的例子，若16×16的输入图像在经过单位步长的5×5的卷积核之前先进行相同填充，则会在水平和垂直方向填充两层，即两侧各增加2个像素（$p = 2$）变为20×20大小的图像，通过卷积核后，输出的特征图尺寸为16×16，保持了原本的尺寸。

3. 激励函数（activation function）

卷积层中包含激励函数以协助表达复杂特征，其表示形式如下 [1] ：

$$\boldsymbol{A}_{i,j,k}^{l} = f\left(\boldsymbol{Z}_{i,j,k}^{l}\right)$$

类似于其它深度学习算法，卷积神经网络通常使用线性整流函数（Rectified Linear Unit, ReLU），其它类似ReLU的变体包括有斜率的ReLU（Leaky ReLU, LReLU）、参数化的ReLU（Parametric ReLU, PReLU）、随机化的ReLU（Randomized ReLU, RReLU）、指数线性单元（Exponential Linear Unit, ELU）等 [2] 。在ReLU出现以前，Sigmoid函数和双曲正切函数（hyperbolic tangent）也有被使用 [15] 。

激励函数操作通常在卷积核之后，一些使用预激活（preactivation）技术的算法将激励函数置于卷积核之前 [34] 。在一些早期的卷积神经网络研究，例如LeNet-5中，激励函数在池化层之后 [5] 。

池化层（pooling layer）

在卷积层进行特征提取后，输出的特征图会被传递至池化层进行特征选择和信息过滤。池化层包含预设定的池化函数，其功能是将特征图中单个点的结果替换为其相邻区域的特征图统计量。池化层选取池化区域与卷积核扫描特征图步骤相同，由池化大小、步长和填充控制 [1] 。

1. L_p池化（Lp pooling）

L_p池化是一类受视觉皮层内阶层结构启发而建立的池化模型 [35] ，其一般表示形式为 [36] ：

$$\boldsymbol{A}_{k}^{l}(i,j) = \left[\sum_{x=1}^{f}\sum_{y=1}^{f}\boldsymbol{A}_{k}^{l}(s_0 i + x, s_0 j + y)^{p}\right]^{\frac{1}{p}}$$

式中步长s_0、像素(i,j)的含义与卷积层相同，p是预指定参数。当$p = 1$时，L_p池化在池化区域内取均值，被称为均值池化（average pooling）；当$p \to \infty$时，L_p池化在区域内取极大值，被称为极大池化（max pooling）。均值池化和极大池化是在卷积神经网络的设计中被长期使用的池化方法，二者以损失特征图的部分信息或尺寸为代价保留图像的背景和纹理信息 [36] 。此外$p = 2$时的L_2池化在一些工作中也有使用 [37] 。

2. 随机/混合池化

混合池化（mixed pooling）和随机池化（stochastic pooling）是L_p池化概念的延伸。随机池化会在其池化区域内按特定的概率分布随机选取一值，以确保部分非极大的激励信号能够进入下一个构筑 [38] 。混合池化可以表示为均值池化和极大池化的线性组合 [39] ：

$$A_k^l = \lambda L_1\left(A_k^l\right) + L_\infty\left(A_k^l\right), \quad \lambda \in [0,1]$$

有研究表明，相比于均值和极大池化，混合池化和随机池化具有正则化的功能，有利于避免卷积神经网络出现过拟合 [2] 。

3. 谱池化（spectral pooling）

谱池化是基于FFT的池化方法，可以和FFT卷积一起被用于构建基于FFT的卷积神经网络 [40] 。在给定特征图尺寸
$\mathbb{R}_{m\times m}$
，和池化层输出尺寸时
$\mathbb{R}_{n\times n}$
，谱池化对特征图的每个通道分别进行DFT变换，并从频谱中心截取n×n大小的序列进行DFT逆变换得到池化结果 [40] 。谱池化有滤波功能，可以在保存输入特征的低频变化信息的同时，调整特征图的大小 [40] 。基于成熟的FFT算法，谱池化能够以很小的计算量完成。

Inception模块（Inception module）

Inception模块：原始版本（a），GoogLeNet使用的版本（b-d） [2]

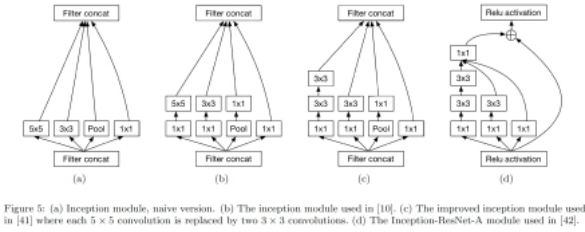


Figure 5: (a) Inception module, naive version. (b) The inception module used in [10]. (c) The improved inception module used in [41] where each 5×5 convolution is replaced by two 3×3 convolutions. (d) The Inception-ResNet-A module used in [42].

Inception模块是对多个卷积层和池化层进行堆叠所得的隐含层构筑。具体而言，一个Inception模块会同时包含多个不同类型的卷积和池化操作，并使用相同填充使上述操作得到相同尺寸的特征图，随后在数组中将这些特征图的通道进行叠加并通过激励函数 [41] 。由于上述做法在一个构筑中引入了多个卷积核，因此为简化计算，Inception模块通常设计了瓶颈层，首先使用单位卷积核，即NIN结构减少特征图的通道数，再进行其它卷积操作 [41] 。Inception模块最早被应用于GoogLeNet并在ImageNet数据集中取得了成功 [41] ，并启发了（或推广得到了）基于深度可分卷积（depthwise separable convolution）搭建的一系列轻量级卷积神经网络，包括Xception和MobileNet [42] 。

全连接层（fully-connected layer）

卷积神经网络中的全连接层等价于传统前馈神经网络中的隐含层。全连接层位于卷积神经网络隐含层的最后部分，并只向其它全连接层传递信号。特征图在全连接层中会失去空间拓扑结构，被展开为向量并通过激励函数 [1] 。

按表征学习观点，卷积神经网络中的卷积层和池化层能够对输入数据进行特征提取，全连接层的作用则是对提取的特征进行非线性组合以得到输出，即全连接层本身不被期望具有特征提取能力，而是试图利用现有的高阶特征完成学习目标。
在一些卷积神经网络中，全连接层的功能可由全局均值池化（global average pooling）取代 [41] ，全局均值池化会将特征图每个通道的所有值取平均，即若有7×7×256的特征图，全局均值池化将返回一个256的向量，其中每个元素都是7×7，步长为7，无填充的均值池化 [41] 。

输出层

卷积神经网络中输出层的上游通常是全连接层，因此其结构和工作原理与传统前馈神经网络中的输出层相同。对于图像分类问题，输出层使用逻辑函数或归一化指数函数（softmax function）输出分类标签 [16] 。在物体识别（object detection）问题中，输出层可设计为输出物体的中心坐标、大小和分类 [16] 。在图像语义分割中，输出层直接输出每个像素的分类结果 [16] 。

理论

编辑 语音

学习范式

监督学习（supervised learning）

参见：反向传播算法

卷积神经网络在监督学习中使用BP框架进行学习，其计算流程在LeCun (1989) 中就已经确定 [5] ，是最早在BP框架进行学习的深度算法之一。卷积神经网络中的BP分为三部分，即全连接层与卷积核的反向传播和池化层的反向通路（backward pass） [1] [43] 。全连接层的BP计算与传统的前馈神经网络相同，卷积层的反向传播是一个与前向传播类似的交叉相关计算：

$$\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{A}}\right)_{i,j}^{l}=\sum_{k=1}^{K_l}\sum_{x-1}^{f}\sum_{y-1}^{f}\left[\boldsymbol{w}_{k}^{l+1}\left(x,y\right)\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{A}}\right)_{s_0 i+x,s_0 j+y,k}^{l+1}\right]f'\left(\boldsymbol{A}_{i,j}^{l}\right)$$

$$\boldsymbol{w}^{l}=\boldsymbol{w}^{l-1}-\alpha\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{w}}\right)_{k}=\boldsymbol{w}^{l+1}-\alpha\left[\boldsymbol{A}^{l+1}\left(\frac{\partial \boldsymbol{E}}{\partial \boldsymbol{A}}\right)_{k}^{l+1}\right]$$

式中
$\boldsymbol{E}$
为代价函数（cost function）计算的误差、
f'
为激励函数的导数、
α
是学习速率（learning rate），若卷积核的前向传播使用卷积计算，则反向传播也对卷积核翻转以进行卷积运算。卷积神经网络的误差函数可以有多种选择，常见的包括Softmax损失函数（softmax loss）、铰链损失函数（hinge loss）、三重损失函数（triplet loss）等 [2] 。
池化层在反向传播中没有参数更新，因此只需要根据池化方法将误差分配到特征图的合适位置即可，对极大池化，所有误差会被赋予到极大值所在位置；对均值池化，误差会平均分配到整个池化区域 [43] 。

卷积神经网络通常使用BP框架内的随机梯度下降（Stochastic Gradient Descent, SGD） [44] 和其变体，例如Adam算法（Adaptive moment estimation） [45] 。SGD在每次迭代中随机选择样本计算梯度，在学习样本充足的情形下有利于信息筛选，在迭代初期能快速收敛，且计算复杂度更小 [44] 。

非监督学习（unsupervised learning）

卷积神经网络最初是面向监督学习问题设计的，但其也发展出了非监督学习范式 [30] [46] ，包括卷积自编码器（Convolutional AutoEncoders, CAE） [47] 、卷积受限玻尔兹曼机（Convolutional Restricted Boltzmann Machines, CRBM）/卷积深度置信网络（Convolutional Deep Belief Networks, CDBN） [48] 和深度卷积生成对抗网络（Deep Convolutional Generative Adversarial Networks, DCGAN） [49] 。这些算法也可以视为在非监督学习算法的原始版本中引入卷积神经网络构筑的混合算法。

CAE的构建逻辑与传统AE类似，首先使用卷积层和池化层建立常规的卷积神经网络作为编码器，随后使用反卷积和向上池化（up-pooling）作为解码器，以样本编码前后的误差进行学习，并输出编码器的编码结果实现对样本的维度消减（dimentionality reduction）和聚类（clustering）。在图像识别问题，例如MNIST中，CAE与其编码器同样结构的卷积神经网络在大样本时表现相当，但在小样本问题中具有更好的识别效果 [47] 。
CRBM是以卷积层作为隐含层的受限玻尔兹曼机（Boltzmann Machines, RBM），在传统RBMs的基础上将隐含层分为多个“组（group）”，每个组包含一个卷积核，卷积核参数由该组对应的所有二元节点共享 [48] 。CDBN是以CRBM作为构筑进行堆叠得到的阶层式生成模型，为了在结构中提取高阶特征，CDBN加入了概率极大池化层（ probabilistic max-pooling layer），和其对应的能量函数。CRBMs和CDBMs使用逐层贪婪算法（greedy layer-wise training）进行学习 [50] ，并可以使用稀疏正则化（sparsity regularization）技术。在Caltech-101数据的物体识别问题中，一个24-100的两层CDBN识别准确率持平或超过了很多包含高度特化特征的分类和聚类算法 [48] 。
生成对抗网络（ Generative Adversarial Networks, GAN）可被用于卷积神经网络的非监督学习，DCGAN从一组概率分布，即潜空间（latent space）中随机采样，并将信号输入一组完全由转置卷积核组成的生成器；生成器生成图像后输入以卷积神经网络构成的判别模型，判别模型判断生成图像是否是真实的学习样本。当生成模型能够使判别模型无法判断其生成图像与学习样本的区别时学习结束。研究表明DCGANs能够在图像处理问题中提取输入图像的高阶层表征，在CIFAR-10数据的试验中，对DCGAN判别模型的特征进行处理后做为其它算法的输入，能以很高的准确率对图像进行分类 [49] 。

优化

正则化（regularization）

参见：正则化
在神经网络算法的各类正则化方法都可以用于卷积神经网络以防止过度拟合，常见的正则化方法包括L_p正则化（L_p-norm regularization）、随机失活（spatial dropout）和随机连接失活（drop connect）。
L_p正则化在定义损失函数时加入隐含层参数以约束神经网络的复杂度：

式中
$L(X,Y,w)$
为损失函数，包含弗罗贝尼乌斯范数（Frobenius norm）的求和项被称为正则化项，其中
λ
是正则化参数，用以确定正则化项的约束力。可证明，当
$p \geq 1$
时，正则化项是凸函数（convex function） [51] ；特别地，当
$p = 2$
时，L_2正则化又被成为Tikhonov正则化（Tikhonov regularization） [52] 。
$p \leq 1$
时的L_p正则化有利于卷积核权重的稀疏化，但此时的正则化向不是凸函数 [2] 。

$$L(X,Y,w) = L(X,Y,w) + \lambda \sum \|\boldsymbol{w}\|^p$$

卷积神经网络中的空间随机失活（spatial dropout）是前馈神经网络中随机失活理论的推广。在全连接网络的学习中，随机失活会随机将神经元的输出归零，而空间随机失活在迭代中会随机选取特征图的通道使其归零 [53] 。进一步地，随机连接失活直接作用于卷积核，在迭代中使卷积核的部分权重归零 [54] 。研究表明空间随机失活和随机连接失活提升了卷积神经网络的泛化能力，在学习样本不足时有利于提升学习表现 [53] [54] 。

分批归一化（Batch Normalization, BN）

数据的标准化是神经网络输入管道中预处理的常见步骤，但在深度网络中，随着输入数据在隐含层内的逐级传递，其均值和标准差会发生改变，产生协变漂移（covariate shift）现象 [55] 。协变漂移被认为是深度网络发生梯度消失（vanishing gradient）的原因之一 [55] 。BN以引入额外学习参数为代价部分解决了此类问题，其策略是在隐含层中首先将特征标准化，然后使用两个线性参数将标准化的特征放大作为新的输入，神经网络会在学习过程中更新其BN参数 [55] 。卷积神经网络中的BN参数与卷积核参数具有相同的性质，即特征图中同一个通道的像素共享一组BN参数 [55] 。此外使用BN时卷积层不需要偏差项，其功能由BN参数代替。
包含跳跃连接的残差块 [56]
跳跃连接（skip connection）
跳跃连接或短路连接（shortcut connection）来源于循环神经网络（Recurrent Neural Network, RNN）中的跳跃连接和各类门控算法，是被用于缓解深度结构中梯度消失问题的技术 [56] 。卷积神经网络中的跳跃连接可以跨越任意数量的隐含层 [56] ，这里以相邻隐含层间的跳跃进行说明：

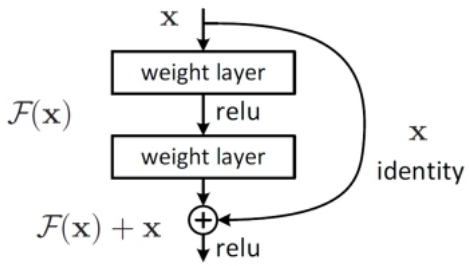


Figure 2. Residual learning: a building block.

式中
u
是特征图的转换系数，当
$\boldsymbol{Z}^l$
和
$\boldsymbol{Z}^{l-1}$
的尺寸不同时，转换系数将尺寸更小的特征图，通常是
$\boldsymbol{Z}^{l-1}$
转换为
$\boldsymbol{Z}^l$
的尺寸，确保矩阵元素运算成立 [56] 。当
$\boldsymbol{Z}^l$
的输出值小而
$\boldsymbol{Z}^{l-1}$
的输出值大时，卷积层
l
的输出近似于等值函数，对该层的特征传递没有负面影响，因此设定了
l
层的学习基线，使该层在迭代中至少不会退化。在BP框架内，部分误差在反向传播时可以跳过
l
层直接作用于
$l-1$
层，补偿了其在深度结构中逐级传播造成的梯度损失，因此有利于深度结构的误差传播。包含跳跃连接的多个卷积层的组合被称为残差块（residual block），是一些卷积神经网络算法，例如ResNet的构筑单元 [56] 。

$$\boldsymbol{A}^l = f\left(\boldsymbol{Z}^l + u\boldsymbol{Z}^{l-1}\right)$$

加速

通用加速技术

卷积神经网络可以使用和其它深度学习算法类似的加速技术以提升运行效率，包括量化（quantization）、迁移学习（transfer learning）等[2]。量化即在计算中使用低数值精度以提升计算速度，该技术在一些深度算法中有得到尝试。对于卷积神经网络，一个极端的例子是XNOR-Net，即仅由异或门（XNOR）搭建的卷积神经网络[57]。迁移学习一般性的策略是将非标签数据迁移至标签数据以提升神经网络的表现，卷积神经网络中迁移学习通常为使用在标签数据下完成学习的卷积核权重初始化新的卷积神经网络，对非标签数据进行迁移，或应用于其它标签数据以缩短学习过程[58]。

FFT卷积

卷积神经网络的卷积和池化计算都可以通过FFT转换至频率域内进行，此时卷积核权重与BP算法中梯度的FFT能够被重复利用，逆FFT也只需在输出结果时使用，降低了计算复杂度[40]。此外，作为应用较广的科学和工程数值计算方法，一些数值计算工具包含了GPU设备的FFT，能提供进一步加速[40]。FFT卷积在处理小尺寸的卷积核时可使用Winograd算法降低内存开销[59]。

权重稀疏化

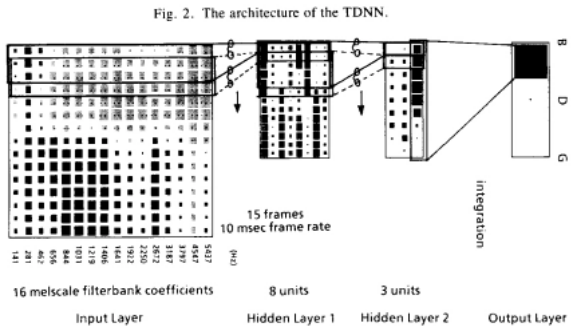
在卷积神经网络中对权重进行稀疏化，能够减少卷积核的冗余，降低计算复杂度，使用该技术的构筑被称为稀疏卷积神经网络（Sparse Convolutional Neural Networks）[60]。在对ImageNet数据的学习中，一个以90%比率稀疏化的卷积神经网络的运行速度是同结构传统卷积神经网络的2至10倍，而输出的分类精度仅损失了2%[60]。

构筑与算法

一维构筑

时间延迟网络（Time Delay Neural Network, TDNN）

TDNN构筑示意图[12]



TDNN是一类应用于语音识别问题的一维卷积神经网络，也是历史上最早被提出的卷积神经网络算法之一。这里以TDNN的原始版本Waibel et al. (1987)[12]为例进行介绍。

TDNN的学习目标为对FFT变换的3个语音音节/b,d,g/进行分类，其隐含层完全由单位步长，无填充的卷积层组成[12]。在文献中，TDNN的卷积核尺寸使用“延迟（delay）”表述，由尺寸为3的一维卷积核构成的隐含层被定义为“时间延迟为2的隐含层”，即感受野包含无延迟输入和2个延迟输入[12]。在此基础上，TDNN有两个卷积层，时间延迟分别为2和4，神经网络中每个输入信号与8个隐含层神经元相连[12]。TDNN没有全连接层，而是将尾端卷积层的输出直接相加通过激励函数得到分类结果。按原作，输入TDNN的预处理数据为15个10毫秒采样的样本（frame），每个样本包含16个通道参数（filterbank coefficients），此时TDNN的结构如下[12]：

1. (3)×16×8的卷积层（步长为1，无填充，Sigmoid函数）
2. (5)×8×3的卷积层（步长为1，无填充，Sigmoid函数）
3. 对9×3的特征图求和输出

列表中数字的含义为：(卷积核尺寸)×卷积核通道（与输入数据通道数相同）×卷积核个数。TDNN的输出层和两个卷积层均使用Sigmoid函数作为激励函数。除上述原始版本外，TDNN的后续研究中出现了应用于字符识别[61]和物体识别[62]的算法，其工作方式是将空间在通道维度展开并使用时间上的一维卷积核，即时间延迟进行学习。

WaveNet

WaveNet构筑示意图[63]

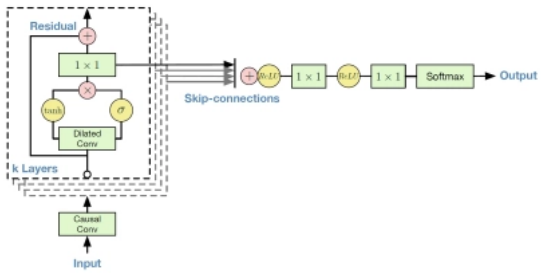


Figure 4: Overview of the residual block and the entire architecture.

WaveNet是被用于语音建模的一维卷积神经网络，其特点是采用扩张卷积和跳跃连接提升了神经网络对长距离依赖的学习能力。WaveNet面向序列数据设计，其结构和常见的卷积神经网络有较大差异，这里按Van Den Oord et al. (2016)[63]做简单介绍：

WaveNet以经过量化和独热编码（one-hot encoding）的音频作为输入特征，具体为一个包含采样和通道的二维数组[63]。输入特征在WaveNet中首先进入线性卷积核，得到的特征图会通过多个扩张卷积块（dilated stack），每个扩张卷积块包含一个过滤器（filter）和一个门（gate），两者都是步长为1，相同填充的线性卷积核，但前者使用双曲正切函数作为激励函数，后者使用Sigmoid函数[64]。特征图从过滤器和门输出后会做矩阵元素乘法并通过由NIN构建的瓶颈层，所得结果的一部分会由跳跃连接直接输出，另一部分与进入该扩张卷积块前的特征图进行线性组合进入下一个构筑[64]。WaveNet的末端部分将跳跃连接和扩张卷积块的所有输出相加并通过两个ReLU-NIN结构，最后由归一化指数函数输出结果并使用交叉熵作为损失函数进行监督学习[63][64]。WaveNet是一个生成模型（generative model），其输出为每个序列元素相对于其之前所有元素的条件概率，与输入序列具有相同的维度[63]：

$$p(\boldsymbol{x}) = \prod_{t=1}^{T} p(x_t | x_1, x_2, \ldots, x_{t-1}), \quad \boldsymbol{x} = \{x_1, x_2, \ldots, x_T\}$$

WaveNet被证实能够生成接近真实的英文、中文和德文语音[63][65]。在经过算法和运行效率的改进后，自2017年11月起，WaveNet开始为谷歌的商业应用“谷歌助手（Google Assistant）”提供语音合成[66]。

二维构筑

LeNet-5

LeNet-5的构筑与特征可视化 [2]

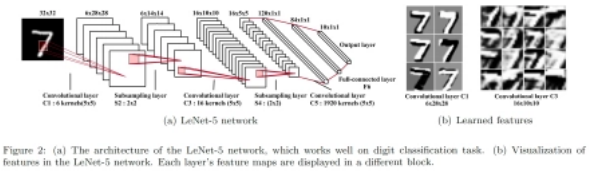


Figure 2: (a) The architecture of the LeNet-5 network, which works well on digit classification task. (b) Visualization of features in the LeNet-5 network. Each layer's feature maps are displayed in a different block.

LeNet-5是一个应用于图像分类问题的卷积神经网络，其学习目标是从一系列由32×32×1灰度图像表示的手写数字中识别和区分0-9。LeNet-5的隐含层由2个卷积层、2个池化层构筑和2个全连接层组成，按如下方式构建：

1. (3×3)×1×6的卷积层（步长为1，无填充），2×2均值池化（步长为2，无填充），tanh激励函数
2. (5×5)×6×16的卷积层（步长为1，无填充），2×2均值池化（步长为2，无填充），tanh激励函数
3. 2个全连接层，神经元数量为120和84

从现代深度学习的观点来看，LeNet-5规模很小，但考虑LeCun et al. (1998) [15] 的数值计算条件，LeNet-5在该时期仍具有相当的复杂度 [16] 。LeNet-5使用双曲正切函数作为激励函数，使用均方差（Mean Squared Error, MSE）作为误差函数并对卷积操作进行了修改以减少计算开销，这些设置在随后的卷积神经网络算法中已被更优化的方法取代 [16] 。在现代机器学习库的范式下，LeNet-5是一个易于实现的算法，这里提供一个使用TensorFlow和Keras的计算例子：

```
# 导入模块
import numpy as np
import tensorflow as tf
from tensorflow import keras
import matplotlib.pyplot as plt
# 读取MNIST数据
mnist = keras.datasets.mnist
(x_train, y_train),(x_test, y_test) = mnist.load_data()
#    重构数据至4维（样本，像素X，像素Y，通道）
x_train = x_train.reshape(x_train.shape + ( 1 ,))
x_test = x_test.reshape(x_test.shape + ( 1 ,))
x_train, x_test = x_train / 255.0 , x_test / 255.0
#     数据标签
label_train = keras.utils.to_categorical(y_train, 10 )
label_test = keras.utils.to_categorical(y_test, 10 )
# LeNet-5构筑
model = keras.Sequential([
keras.layers.Conv2D( 6 , kernel_size = ( 3 , 3 ), strides = ( 1 , 1 ), activation = 'tanh' , padding
keras.layers.AveragePooling2D(pool_size = ( 2 , 2 ), strides = ( 2 , 2 ), padding = 'valid' ),
keras.layers.Conv2D( 16 , kernel_size = ( 5 , 5 ), strides = ( 1 , 1 ), activation = 'tanh' , padding
keras.layers.AveragePooling2D(pool_size = ( 2 , 2 ), strides = ( 2 , 2 ), padding = 'valid' ),
keras.layers.Flatten(),
keras.layers.Dense( 120 , activation = 'tanh' ),
keras.layers.Dense( 84 , activation = 'tanh' ),
keras.layers.Dense( 10 , activation = 'softmax' ),
  ])
# 使用SGD编译模型
model. compile (loss = keras.losses.categorical_crossentropy, optimizer = 'SGD' )
# 学习30个纪元（可依据CPU计算力调整），使用20%数据交叉验证
records = model.fit(x_train, label_train, epochs = 20 , validation_split = 0.2 )
# 预测
y_pred = np.argmax(model.predict(x_test), axis = 1 )
print ( "prediction accuracy: {}" . format ( sum (y_pred = = y_test) / len (y_test)))
# 绘制结果
plt.plot(records.history[ 'loss' ],label = 'training set loss' )
plt.plot(records.history[ 'val_loss' ],label = 'validation set loss' )
plt.ylabel( 'categorical cross-entropy' ); plt.xlabel( 'epoch' )
plt.legend()
```

该例子使用MNIST数据代替LeCun et al. (1998) [15] 的原始数据，使用交叉熵（categorical cross-entropy）作为损失函数。

ILSVRC中的优胜算法

ILSVRC [22] 为各类应用于计算机视觉的人工智能算法提供了比较的平台，其中有多个卷积神经网络算法在图像分类和物体识别任务中获得优胜，包括AlexNet、ZFNet、VGGNet、GoogLeNet和ResNet，这些算法在ImageNet数据中展现了良好的学习性能，也是卷积神经网络发展中具有代表意义的算法。

对AlexNet、ZFNet的编程实现与LeNet-5类似，对VGGNet、GoogLeNet和ResNet的编程实现较为繁琐，一些机器学习库提供了完整的封装模型和预学习的权重，这里提供一些使用TensorFlow和Keras的例子：

```
from tensorflow import keras
from keras import applications
# load model and pre-trained weights
model_vgg16 = applications.VGG16(weights = 'imagenet', include_top = False)
model_vgg19 = applications.VGG19(weights = 'imagenet', include_top = False)
model_resnet = applications.ResNet50(weights = 'imagenet', include_top = False)
model_googlenet = applications.InceptionV3(weights = 'imagenet', include_top = False)
```

1. AlexNet

AlexNet构筑示意图 [67]

参见：AlexNet

AlexNet是2012年ILSVRC图像分类和物体识别算法的优胜者，也是LetNet-5之后受到人工智能领域关注的现代卷积神经网络算法 [16] [26] 。AlexNet的隐含层由5个卷积层、3个池化层和3个全连接层组成，按如下方式构建 [68] ：

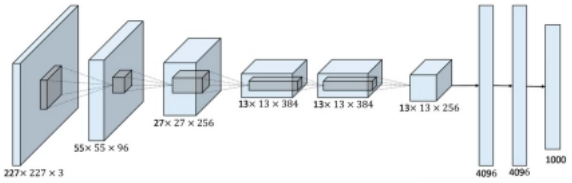


Figure 1. The AlexNet architecture.

1. (11×11)×3×96的卷积层（步长为4，无填充，ReLU），3×3极大池化（步长为2、无填充），LRN
2. (5×5)×96×256的卷积层（步长为1，相同填充，ReLU），3×3极大池化（步长为2、无填充），LRN
3. (3×3)×256×384的卷积层（步长为1，相同填充，ReLU）
4. (3×3)×384×384的卷积层（步长为1，相同填充，ReLU）
5. (3×3)×384×256的卷积层（步长为1，相同填充，ReLU），3×3极大池化（步长为2、无填充）
6. 3个全连接层，神经元数量为4096、4096和1000

AlexNet在卷积层中选择ReLU作为激励函数，使用了随机失活，和数据增强（data data augmentation）技术 [68] ，这些策略在其后的卷积神经网络中被保留和使用 [26] 。AlexNet也是首个基于GPU进行学习的卷积神经网络，Krizhevsky (2012) 将AlexNet按结构分为两部分，分别在两块GPU设备上运行。此外AlexNet的1-2部分使用了局部响应归一化（local response normalization, LRN），在2014年后出现的卷积神经网络中，LRN已由分批归一化取代 [16] [56] 。

2. ZFNet

ZFNet是2013年ILSVRC图像分类算法的优胜者，其结构与AlexNet相近，仅将第一个卷积层的卷积核大小调整为7×7、步长减半 [38] ：

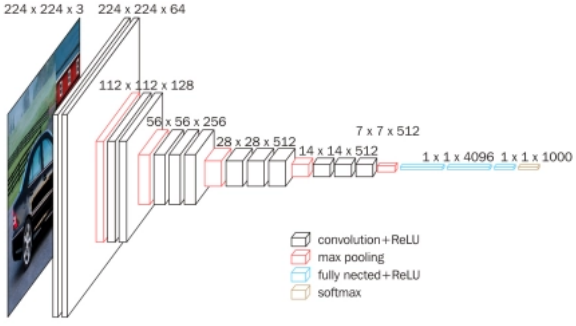
1. (7×7)×3×96的卷积层（步长为2，无填充，ReLU），3×3极大池化（步长为2、无填充），LRN
2. (5×5)×96×256的卷积层（步长为1，相同填充，ReLU），3×3极大池化（步长为2、无填充），LRN
3. (3×3)×256×384的卷积层（步长为1，相同填充，ReLU）
4. (3×3)×384×384的卷积层（步长为1，相同填充，ReLU）
5. (3×3)×384×256的卷积层（步长为1，相同填充，ReLU），3×3极大池化（步长为2、无填充）
6. 3个全连接层，神经元数量为4096、4096和1000

ZFNet对卷积神经网络的贡献不在其构筑本身，而在于其原作者通过反卷积考察了ZFNet内部的特征提取细节，解释了卷积神经网络的特征传递规律，即由简单的边缘、夹角过渡至更为复杂的全局特征 [38] 。

3. VGGNet

VGGNet构筑示意图 [16]

VGGNet是牛津大学视觉几何团队（Visual Geometry Group, VGG）开发的一组卷积神经网络算法，包括VGG-11、VGG-11-LRN、VGG-13、VGG-16和VGG-19 [69] [70] 。其中VGG-16是2014年ILSVRC物体识别算法的优胜者，其规模是AlexNet的2倍以上并拥有规律的结构，这里以VGG-16为例介绍其构筑。VGG-16的隐含层由13个卷积层、3个全连接层和5个池化层组成，按如下方式构建：



1. (3×3)×3×64的卷积层（步长为1，相同填充，ReLU），(3×3)×64×64的卷积层（步长为1，相同填充，ReLU），2×2极大池化（步长为2、无填充）
2. (3×3)×64×128的卷积层（步长为1，相同填充，ReLU），(3×3)×128×128的卷积层（步长为1，相同填充，ReLU），2×2极大池化（步长为2、无填充）
3. (3×3)×128×256的卷积层（步长为1，相同填充，ReLU），(3×3)×256×256的卷积层（步长为1，相同填充，ReLU），(3×3)×256×256的卷积层（步长为1，相同填充，ReLU），2×2极大池化（步长为2、无填充）

4. (3×3)×256×512的卷积层（步长为1，相同填充，ReLU），(3×3)×512×512的卷积层（步长为1，相同填充，ReLU），(3×3)×512×512的卷积层（步长为1，相同填充，ReLU），2×2极大池化（步长为2、无填充）

5. (3×3)×512×512的卷积层（步长为1，相同填充，ReLU），(3×3)×512×512的卷积层（步长为1，相同填充，ReLU），(3×3)×512×512的卷积层（步长为1，相同填充，ReLU），2×2极大池化（步长为2、无填充）

6. 3个全连接层，神经元数量为4096、4096和1000

VGGNet构筑中仅使用3×3的卷积核并保持卷积层中输出特征图尺寸不变，通道数加倍，池化层中输出的特征图尺寸减半，简化了神经网络的拓扑结构并取得了良好效果 [16] [69] 。

4. GoogLeNet

Inception v1构筑示意图 [41]

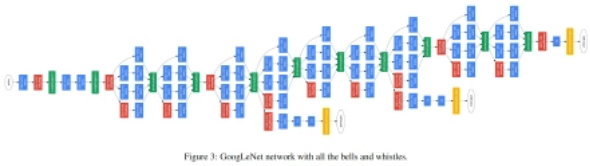


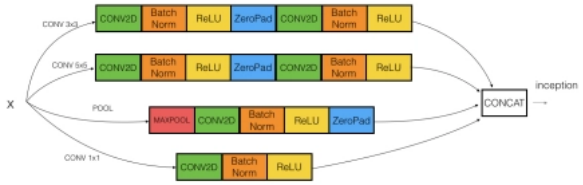
Figure 3: GoogLeNet network with all the bells and whistles.

参见：GoogLeNet

GoogLeNet是2014年ILSVRC图像分类算法的优胜者，是首个以Inception模块进行堆叠形成的大规模卷积神经网络。GoogLeNet共有四个版本：Inception v1、Inception v2、Inception v3、Inception v4 [71] ，这里以Inception v1为例介绍。首先，Inception v1的Inception模块被分为四部分 [41] ：

- N1个(1×1)×C的卷积核
- B3个(1×1)×C的卷积核（BN，ReLU），N3个(3×3)×96的卷积核（步长为1，相同填充，BN，ReLU）
- B5个(1×1)×C的卷积核（BN，ReLU），N5个(5×5)×16的卷积核（步长为1，相同填充，BN，ReLU）
- 3×3的极大池化（步长为1，相同填充），Np个(1×1)×C的卷积核（BN，ReLU）

Inception v1中的Iception模块 [16]



在此基础上，对3通道的RGB图像输入，Inception v1按如下方式构建 [41] ：

1. (7×7)×3×64的卷积层（步长为2，无填充，BN，ReLU），3×3的极大池化（步长为2，相同填充），LRN
2. (3×3)×64×192的卷积层（步长为1，相同填充，BN，ReLU），LRN，3×3极大池化（步长为2，相同填充）
3. Inception模块（N1=64，B3=96，N3=128，B5=16，N5=32，Np=32）
4. Inception模块（N1=128，B3=128，N3=192，B5=32，N5=96，Np=64）
5. 3×3极大池化（步长为2，相同填充）
6. Inception模块（N1=192，B3=96，N3=208，B5=16，N5=48，Np=64）
7. 旁枝：5×5均值池化（步长为3，无填充）
8. Inception模块（N1=160，B3=112，N3=224，B5=24，N5=64，Np=64）
9. Inception模块（N1=128，B3=128，N3=256，B5=24，N5=64，Np=64）
10. Inception模块（N1=112，B3=144，N3=288，B5=32，N5=64，Np=64）
11. 旁枝：5×5均值池化（步长为3，无填充）
12. Inception模块（N1=256，B3=160，N3=320，B5=32，N5=128，Np=128）
13. Inception模块（N1=384，B3=192，N3=384，B5=48，N5=128，Np=128）
14. 全局均值池化，1个全连接层，神经元数量为1000，权重40%随机失活

GoogLeNet中的Inception模块启发了一些更为现代的算法，例如2017年提出的Xception [42] 。Inception v1的另一特色是其隐含层中的两个旁枝输出，旁枝和主干的所有输出会通过指数归一化函数得到结果，对神经网络起正则化的作用 [41] 。

5. 残差神经网络（Residual Network, ResNet）

ResNet（上），同规模的普通CNN（中）和VGG-19（下）构筑的比较 [56]

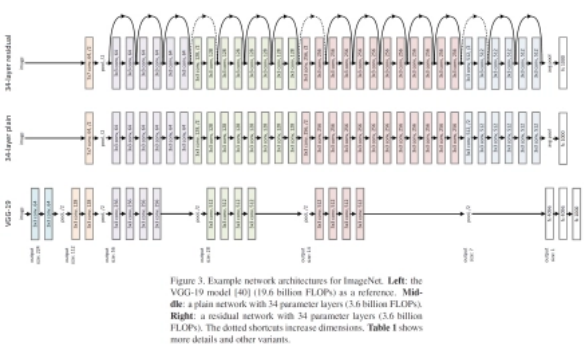


Figure 3. Example network architectures for ImageNet. **Left**: the VGG-19 model [40] (19.6 billion FLOPs) as a reference. **Middle**: a plain network with 34 parameter layers (3.6 billion FLOPs). **Right**: a residual network with 34 parameter layers (3.6 billion FLOPs). The dotted shortcuts increase dimensions. **Table 1** shows more details and other variants.

参见：残差网络

ResNet来自微软的人工智能团队Microsoft Research，是2015年ILSVRC图像分类和物体识别算法的优胜者，其表现超过了GoogLeNet的第三代版本Inception v3 [25] 。ResNet是使用残差块建立的大规模卷积神经网络，其规模是AlexNet的20倍、VGG-16的8倍，在ResNet的原始版本中，其残差块由2个卷积层、1个跳跃连接、BN和激励函数组成，ResNet的隐含层共包含16个残差块，按如下方式构建 [56] ：

1. (7×7)×3×64的卷积层（步长为2，无填充，ReLU，BN），3×3的极大池化（步长为2，相同填充）
2. 3个残差块：3×3×64×64卷积层（步长为1，无填充，ReLU，BN），3×3×64×64卷积层（步长为1，无填充）

3. 1个残差块：3×3×64×128（步长为2，无填充，ReLU，BN），3×3×128×128（步长为1，无填充，ReLU，BN）
4. 3个残差块：3×3×128×128（步长为1，无填充，ReLU，BN），3×3×128×128（步长为1，无填充，ReLU，BN）
5. 1个残差块：3×3×128×256（步长为2，无填充，ReLU，BN），3×3×256×256（步长为1，无填充，ReLU，BN）
6. 5个残差块：3×3×256×256（步长为1，无填充，ReLU，BN），3×3×256×256（步长为1，无填充，ReLU，BN）
7. 1个残差块：3×3×256×512（步长为2，无填充，ReLU，BN），3×3×512×512（步长为1，无填充，ReLU，BN）
8. 2个残差块：3×3×512×512（步长为1，无填充，ReLU，BN），3×3×512×512（步长为1，无填充，ReLU，BN）
9. 全局均值池化，1个全连接层，神经元数量为1000

ResNet受到关注的原因是其在隐含层中通过跳跃连接构建的残差块。残差块的堆叠缓解了深度神经网络普遍出现的梯度消失（gradient vanishing）问题，被其后的诸多算法使用，包括GoogLeNet中的Inception v4 [71] 。

在ResNet的基础上诸多研究尝试了改进算法，包括预激活ResNet（preactivation ResNet）、宽ResNet（wide ResNet）、随机深度ResNets（Stochastic Depth ResNets, SDR）和RiR（ResNet in ResNet）等 [2] 。预激活ResNet将激励函数和BN计算置于卷积核之前以提升学习表现和更快的学习速度 [34] ；宽ResNet使用更多通道的卷积核以提升原ResNet的宽度，并尝试在学习中引入随机失活等正则化技术 [72] ；SDR在学习中随机使卷积层失活并用等值函数取代以达到正则化的效果 [73] ；RiR使用包含跳跃连接和传统卷积层的并行结构建立广义残差块，对ResNet进行了推广 [74] 。上述改进算法都报告了比传统ResNet更好的学习表现，但尚未在使用基准数据的大规模比较，例如ILSVRC中得到验证。

全卷积构筑

部分计算机视觉问题，例如图像语义分割（semantic segmentation）和超分辨率图像生成（super resolution imaging）要求输入与输出均为格点数据且输入端的特征图大小可变。全卷积构筑为解决上述问题而设计的神经网络算法。

SRCNN（Super Resolution CNN）

SRCNN构筑示意图 [75]

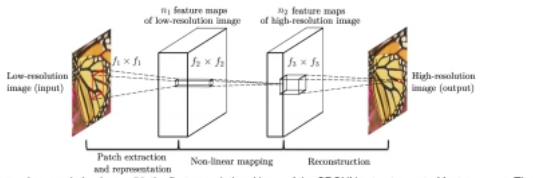
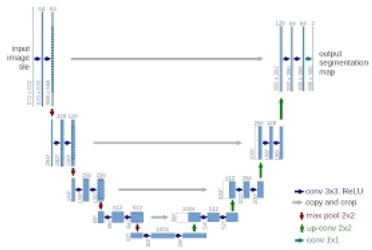


Fig. 2. Given a low-resolution image $\mathbf{Y}$, the first convolutional layer of the SRCNN extracts a set of feature maps. The second layer maps these feature maps nonlinearly to high-resolution patch representations. The last layer combines the predictions within a spatial neighbourhood to produce the final high-resolution image $F(\mathbf{Y})$.

SRCNN是最早被提出的全卷积构筑之一，被应用于超分辨率图像生成。其构筑分为3部分：特征提取端、非线性映射和特征重构，其中特征提取端将低分辨率输入按插值算法升采样至目标分辨率并使用9x9的卷积核提取特征；非线性映射是一个瓶颈层，进行低分辨率特征和高分辨率特征的线性变换。特征重构端是一个转置卷积，将高分辨率特征重构是目标分辨率并输出结果 [75] 。

UNet

UNet构筑示意图 [76]



UNet是一个包含4层降采样、4层升采样和类似跳跃连接结构的全卷积网络，其特点是卷积层在降采样和升采样部分完全对称，且降采样端的特征图可以跳过深层采样，被拼接至对应的升采样端 [76] 。UNet在其提出之初主要被用于医学影像的语义分割 [76] ，并在之后的应用研究中被扩展至3维视频数据的语义分割 [77] 和超分辨率图像生成 [78] 。UNet是一个泛用性较好的全卷积网络，也衍生出了一些面向特定问题的改进版本，例如在降采样端引入残差块构筑的HDense-UNet [79] 、包含深监督设计和模型剪枝的UNet++等 [80] 。

性质

编辑 语音

连接性

卷积神经网络中卷积层间的连接被称为稀疏连接（sparse connection），即相比于前馈神经网络中的全连接，卷积层中的神经元仅与其相邻层的部分，而非全部神经元相连。具体地，卷积神经网络第l层特征图中的任意一个像素（神经元）都仅是l-1层中卷积核所定义的感受野内的像素的线性组合 [1] 。卷积神经网络的稀疏连接具有正则化的效果，提高了网络结构的稳定性和泛化能力，避免过度拟合，同时，稀疏连接减少了权重参数的总量，有利于神经网络的快速学习，和在计算时减少内存开销 [1] 。
卷积神经网络中特征图同一通道内的所有像素共享一组卷积核权重系数，该性质被称为权重共享（weight sharing）。权重共享将卷积神经网络和其它包含局部连接结构的神经网络相区分，后者虽然使用了稀疏连接，但不同连接的权重是不同的 [1] 。权重共享和稀疏连接一样，减少了卷积神经网络的参数总量，并具有正则化的效果 [1] 。

在全连接网络视角下，卷积神经网络的稀疏连接和权重共享可以被视为两个无限强的先验（pirior），即一个隐含层神经元在其感受野之外的所有权重系数恒为0（但感受野可以在空间移动）；且在一个通道内，所有神经元的权重系数相同 [1] 。

表征学习

基于反卷积和向上池化的卷积神经网络特征重构 [81]
作为深度学习的代表算法，卷积神经网络具有表征学习能力，即能够从输入信息中提取高阶特征。具体地，卷积神经网络中的卷积层和池化层能够响应输入特征的平移不变性，即能够识别位于空间不同位置的相近特征。能够提取平移不变特征是卷积神经网络在计算机视觉问题中得到应用的原因之一。

平移不变特征在卷积神经网络内部的传递具有一般性的规律。在图像处理问题中，卷积神经网络前部的特征图通常会提取图像中有代表性的高频和低频特征；随后经过池化的特征图会显示出输入图像的边缘特征（aliasing artifacts）；当信号进入更深的隐含层后，其更一般、更完整的特征会被提取[81]。反卷积和反池化（un-pooling）可以对卷积神经网络的隐含层特征进行可视化[81]。一个成功的卷积神经网络中，传递至全连接层的特征图会包含与学习目标相同的特征，例如图像分类中各个类别的完整图像[81]。

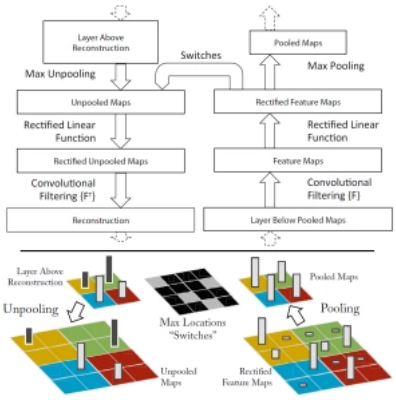


Fig. 1. Top: A deconvnet layer (left) attached to a convnet layer (right). The deconvnet will reconstruct an approximate version of the convnet features from the layer beneath. Bottom: An illustration of the unpooling operation in the deconvnet, using *switches* which record the location of the local max in each pooling region (colored zones) during pooling in the convnet. The black/white bars are negative/positive activations within the feature map.

生物学相似性

卷积神经网络中基于感受野设定的稀疏连接有明确对应的神经科学过程——视觉神经系统中视觉皮层（visual cortex）对视觉空间（visual space）的组织[82] [4]。视觉皮层细胞从视网膜上的光感受器接收信号，但单个视觉皮层细胞不会接收光感受器的所有信号，而是只接受其所支配的刺激区域，即感受野内的信号。只有感受野内的刺激才能够激活该神经元。多个视觉皮层细胞通过系统地将感受野叠加完整接收视网膜传递的信号并建立视觉空间[26] [82]。事实上机器学习的“感受野”一词即来自其对应的生物学研究[9]。卷积神经网络中的权重共享的性质在生物学中没有明确证据，但在对与大脑学习密切相关的目标传播（target-propagation, TP）和反馈调整（feedback alignment, FA） 机制的研究中，权重共享提升了学习效果[83]。

应用

编辑 语音

计算机视觉

图像识别（image classification）

参见：图像识别

基于卷积神经网络的鸟类识别[84]

卷积神经网络长期以来是图像识别领域的核心算法之一，并在学习数据充足时有稳定的表现[85]。对于一般的大规模图像分类问题，卷积神经网络可用于构建阶层分类器（hierarchical classifier）[86]，也可以在精细分类识别（fine-grained recognition）中用于提取图像的判别特征以供其它分类器进行学习[87]。对于后者，特征提取可以人为地将图像的不同部分分别输入卷积神经网络[84]，也可以由卷积神经网络通过非监督学习自行提取[88]。

CNN-RNN的字符识别和序列标注[89]

对于字符检测（text detection）和字符识别（text recognition）/光学字符读取，卷积神经网络被用于判断输入的图像是否包含字符，并从中剪取有效的字符片断[90-91]。其中使用多个归一化指数函数直接分类的卷积神经网络被用于谷歌街景图像的门牌号识别[92]、包含条件随机场（Conditional Random Fields, CRF）图模型的卷积神经网络可以识别图像中的单词[93]，卷积神经网络与循环神经网络（Recurrent Neural Network, RNN）相结合可以分别从图像中提取字符特征和进行序列标注（sequence labelling）[89]。

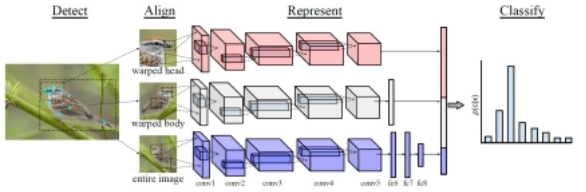


Figure 1: Pipeline Overview: Given a test image, we use groups of detected keypoints to compute multiple warped image regions that are aligned with prototypical models. Each region is fed through a deep convolutional network, and features are extracted from multiple layers. Features are concatenated and fed to a classifier.

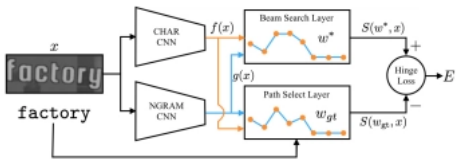


Figure 3: The architecture for training the joint model, comprising of the character sequence model (CHAR) and and the N-gram encoding model (NGRAM) with structured output loss. The Path Select Layer generates the score $S(w_{gt}, x)$ by summing the inputs of the groundtruth word. The Beam Search Layer uses beam search to try to select the path with the largest score $S(w^*, x)$ from the inputs. The hinge loss implements a ranking loss, constraining the highest scoring path to be the groundtruth path, and can be back-propagated through the entire network to jointly learn all the parameters.

物体识别（object recognition）

卷积神经网络可以通过三类方法进行物体识别：滑动窗口（sliding window）、选择性搜索（selective search）和YOLO（You Only Look Once）[2]。滑动窗口出现最早，并被用于手势识别等问题[19]，但由于计算量大，已经被后两者淘汰[2]。选择性搜索对应区域卷积神经网络（Region-based CNN），该算法首先通过一般性步骤判断一个窗口是否可能有目标物体，并进一步将其输入复杂的识别器中[94]。YOLO算法将物体识别定义为对图像中分割框内各目标出现概率的回归问题，并对所有分割框使用同一个卷积神经网络输出各个目标的概率，中心坐标和框的尺寸[95]。基于卷积神经网络的物体识别已被应用于自动驾驶[96]和交通实时监测系统[97]。

此外，卷积神经网在图像语义分割（semantic segmentation）[29] [98]、场景分类（scene labeling）[99-100]和图像显著度检测（Visual Saliency Detection）[101]等问题中也有应用，其表现被证实超过了很多使用特征工程的分类系统。

行为认知（action recognition）

在针对图像的行为认知研究中，卷积神经网络提取的图像特征被应用于行为分类（action classification）[102-103]。在视频的行为认知问题中，卷积神经网络可以保持其二维结构并通过堆叠连续时间片段的特征进行学习[104]、建立沿时间轴变化的3D卷积神经网络[105]、或者逐帧提取特征并输入循环神经网络[106]，三者在特定问题下都可以表现出良好的效果。

姿态估计（pose estimation）

姿态估计在图像中将人的姿态用坐标的形式输出，最早在姿态估计中使用的卷积神经网络是DeepPose，DeepPose的结构类似于AlexNet，以完整的图片作为输出，按监督学习的方式训练并输出坐标点 [107] 。此外也有关于局部姿态估计的卷积神经网络应用研究 [108] 。对于视频数据，有研究使用滑动窗口的卷积神经网络进行逐帧的姿态估计 [109] 。

神经风格转换：内容（左下）、风格（左上）、输出（右） [110]

神经风格迁移（neural style transfer）

神经风格迁移是卷积神经网络的一项特殊应用，其功能是在给定的两份图像的基础上创作第三份图像，并使其内容和风格与给定的图像尽可能地接近 [111] 。

神经风格迁移在本质上不是一个机器学习问题，而是对卷积神经网络表征学习能力的应用。具体地，神经风格迁移在预学习的卷积神经网络中提取高阶层表征，通过表征定义内容（content loss）和风格（style loss）损失，并在第三份图像（通常初始化为白噪声）中对内容和风格的线性组合进行逐格点优化以输出结果 [111] 。

神经风格迁移除进行艺术创作外，也被用于照片的后处理 [112] 和超分辨率图像生成 [113] 。

自然语言处理

总体而言，由于受到窗口或卷积核尺寸的限制，无法很好地学习自然语言数据的长距离依赖和结构化语法特征，卷积神经网络在自然语言处理（Natural Language Processing, NLP）中的应用要少于循环神经网络，且在很多问题中会在循环神经网络的构架上进行设计，但也有一些卷积神经网络算法在多个NLP主题中取得成功 [2] 。

在语音处理（speech processing）领域，卷积神经网络的表现被证实优于隐马尔可夫模型（Hidden Markov Model, HMM）、高斯混合模型（Gaussian Mixture Model, GMM ）和其它一些深度算法 [114-115] 。有研究使用卷积神经网络和HMM的混合模型进行语音处理，模型使用了小的卷积核并将替池化层用全连接层代替以提升其学习能力 [116] 。卷积神经网络也可用于语音合成（speech synthesis）和语言建模（language modeling），例如WaveNet使用卷积神经网络构建的生成模型输出语音的条件概率，并采样合成语音 [63] 。卷积神经网络与长短记忆模型（Long Short Term Memory model, LSTM）相结合可以很好地对输入句子进行补全 [117] 。其它有关的工作包括genCNN、ByteNet等 [31] [118] 。

其它

物理学

使用CNN提取喷流图特征 [119]

卷积神经网络在包含大数据问题的物理学研究中有得到关注。在高能物理学中，卷积神经网络被用于粒子对撞机（particle colliders）输出的喷流图（jet image）的分析和特征学习，有关研究包括夸克（quark）/胶子（gluon）分类 [120] 、W玻色子（W boson）识别 [119] 和中微子相互作用（neutrino interaction）研究 [121] 等。卷积神经网络在天体物理学中也有应用，有研究使用卷积神经网络对天文望远镜图像进行星系形态学（galaxy morphology）分析 [122] 和提取星系模型（galactic model）参数 [123] 。利用迁移学习技术，预训练的卷积神经网络可以对LIGO（Laser Interferometer Gravitational-wave Observatory）数据中的噪声（glitch）进行检测，为数据的预处理提供帮助 [124] 。

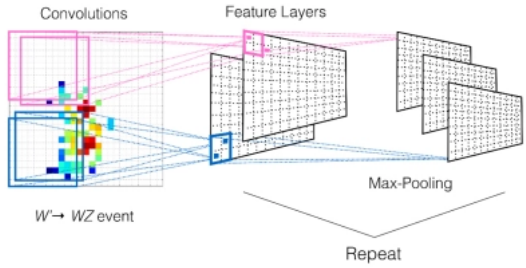


Figure 5. The convolution neural network concept as applied to jet-images.

遥感科学

卷积神经网络在遥感科学，尤其是卫星遥感中有得到应用，并被认为在解析遥感图像的几何、纹理和空间分布特征时，有计算效率和分类准确度方面的优势 [125] 。依据遥感图像的来源和目的，卷积神经网络被用于下垫面使用和类型改变（land use/land cover change） 研究 [67] [126] 以及物理量，例如海冰覆盖率（sea-ice concentration）的遥感反演 [127] 。此外卷积神经网络被用于遥感图像的物体识别 [128] 和图像语义分割 [129] ，后两者是直接的计算机视觉问题，这里不再赘述。

大气科学

在大气科学中，卷积神经网络被用于数值模式格点输出的后处理问题，包括统计降尺度（Statistical Downscaling, SD）、预报校准、极端天气检测等。

基于UNet的统计降尺度（降水）与BCSD和台站观测的比较。 [130]

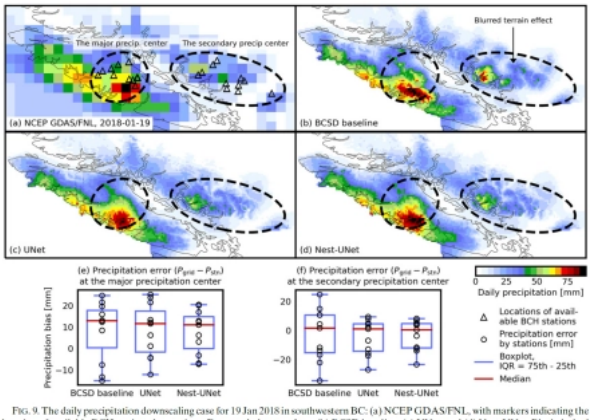


FIG. 9. The daily precipitation downscaling case for 19 Jan 2018 in southwestern BC: (a) NCEP GDAS/FNL, with markers indicating the location of available BCH station observations. Downscaled output from (b) BCSD baseline, (c) UNet, and (d) Nest-UNet. Black dashed circles highlight the two precipitation centers that are measured by the BCH stations. (e),(f) Boxplots of precipitation error ($P_{grid} - P_{stn}$) for the two precipitation centers, as defined in (a). The arrow in (b) points to the blurred terrain effect in the output of BCSD baseline.

在统计降尺度方面，以SRCNN、UNet为代表的全卷积网络可以将插值到高分辨率的（低分辨率）原始气象数据和高分辨率的数字高程模型（Degital Elevation Model, DEM）作为输入，并输出高分辨率的气象数据，其准确率超过了传统的空间分解误差订正（Bias Corrected Spatial Disaggregation, BCSD）方法 [130-132] 。

在极端天气检测方面，仿AlexNet结构的卷积神经网络在监督学习和半监督学习中被证实能以很高的准确度识别气候模式输出和再分析数据（reanalysis data）中的热带气旋（tropical cyclones）、大气层河流（atmospheric rivers）和锋面（weather fronts）现象 [133-134] 。

包含卷积神经网络的编程模块

现代主流的机器学习库和界面，包括TensorFlow、Keras、Thenao、Microsoft-CNTK等都可以运行卷积神经网络算法。此外一些商用数值计算软件，例如MATLAB也有卷积神经网络的构建工具可用 [135] 。

词条图册 更多图册

参考资料

- 1. Goodfellow, I., Bengio, Y., Courville, A. . Deep learning (Vol. 1) . Cambridge：MIT press，2016：326-366

- 2. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, L., Wang, G. and Cai, J., 2015. Recent advances in convolutional neural networks. arXiv preprint arXiv:1512.07108.
- 3. Zhang, W., 1988. Shift-invariant pattern recognition neural network and its optical architecture. In Proceedings of annual conference of the Japan Society of Applied Physics.
- 4. LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.
- 5. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), pp.541-551.
- 7. 福島邦彦, 1979. コグニトロンのパターン分离能力の向上. 电子情报通信学会论文志 A, 62(10), 650-657.
- 8. 福島邦彦, 1979. 位置ずれに影响されないパターン认识机构の神経回路モデルーネオコグニトロンー. 电子情报通信学会论文志 A, 62(10), 658-665.
- 9. Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological Cybernetics, 36(4), 193-202.
- 10. Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

展开全部

学术论文
内容来自

查看全部 ›