

doi: 10.12052/gdutxb.170093

# 卷积网络深度学习算法与实例

陈旭, 张军, 陈文伟, 李硕豪

(国防科技大学 信息系统与管理学院, 湖南 长沙 410073)

**摘要:** 卷积神经网络(CNN)和反向传播神经网络结合组成的深度学习模型,是目前应用比较广泛的深度学习模型,但是具体算法有多种.本文针对以CNN为主体的深度学习算法的实现技术,包括卷积核设计、池化的作用、激活函数的选取和训练过程等问题进行详细讨论,并结合实例进行说明,便于对卷积网络深度学习算法的掌握,使其能得到更广泛的应用.最后总结并展望卷积神经网络未来的研究方向.

**关键词:** 卷积神经网络; 反向传播; 深度学习; 卷积层; 池化层

中图分类号: TP391

文献标志码: A

文章编号: 1007-7162(2017)06-0020-07

## Convolutional Neural Network Algorithm and Case

Chen Xu, Zhang Jun, Chen Wen-wei, Li Shuo-hao

(College of Information System and Management, National University of Defense Technology, Changsha 410073, China)

**Abstract:** Convolutional neural network (CNN) is a deep learning model with strong expression and classification ability, and is currently widely used, but there are a variety of specific algorithms. In this paper, the realization of deep learning algorithm based on convolutional neural network CNN, including the function of convolution kernel, the role of pooling, the selection of activation function and the training process are discussed. And one example is explained, which facilitates the mastery of CNN. Finally, the future research direction of convolution neural network is summarized and forecasted.

**Key words:** convolutional neural network; back propagation; deep learning; convolutional layer; pooling layer

2006年以卷积神经网络(convolutional neural network, CNN)为主体的深度学习模型的提出在图像识别任务上取得了重大突破.深度学习成了大家关注的焦点.典型的卷积神经网络是Krizhevsky et al.提出的AlexNet<sup>[1]</sup>、ZFNet<sup>[2]</sup>、VGGNet<sup>[3]</sup>、GoogleNet<sup>[4]</sup>和ResNet<sup>[5]</sup>等.

为了提高其特征的表征能力,网络深度越来越深,网络的参数也随着深度的增加越来越多,从而导致网络的优化越来越难,更容易出现过拟合,一些防止网络过拟合的方法也就逐渐出现.

本文针对以卷积神经网络CNN为主体的深度学习算法的实现技术:卷积核的含义、卷积核的大小、池化的作用、激活函数的选取、学习率的选取和初始值的确定等问题进行讨论,这是实现深度学习算法的核心问题.

## 1 卷积神经网络结构和训练过程

### 1.1 网络结构

卷积神经网络的结构和传统的神经网络非常相似:都是由神经元构成,神经元包括可以学习的网络权重和偏置(阈值);每个神经元对输入进行运算后输出(前向传播);输出结果和样本输出进行对比后获得误差(计算残差);将残差逐层往后传递,用来更新神经元的权重和偏置(反向传播);循环往复,直到残差收敛且满足精度需求.一般情况下, CNN的结构形式是:数据层(data layer)→卷积层(convolutional layer)→池化层(pooling layer)→... (重复卷积、池化层)→全连接层(fully connected layer)→输出结果层.其中最重要的就是卷积层和池化层.

收稿日期: 2017-04-14

基金项目: 国家自然科学基金资助项目(6117019)

作者简介: 陈旭(1992-),男,博士研究生,主要研究方向为深度学习和自然场景文字检测.

通信作者: 陈文伟(1940-),男,教授,博士生导师,主要方向为决策支持系统、数据仓库和数据挖掘等. E-mail: chenww93@126.com

### 1.1.1 卷积层

CNN的主要思想来自局部连接和权值共享. 如图1所示, 不同于感知机每个神经元与前一层的所有神经元进行全连接, CNN的每个神经元只与一个局部区域的神经元相连. 局部连接可以显著减少参数的数量, 同时每个局部区域神经元可以提取初级的视觉特征, 如方向线段、端点和角点等. 假设每个神经元连接的神经元(权重)是相同的, 可以理解为每个神经元只关注图像的一个特性. 图1(c)中相同颜色线条代表相同权重. CNN主要借助卷积层来完成以上任务.

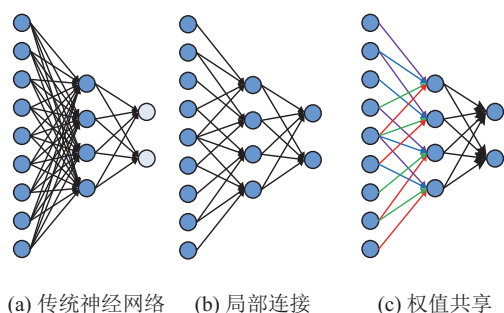


图1 卷积神经网络参数减少过程

Fig.1 The process of reducing parameters in CNN

卷积层是通过卷积核计算后的网络层. 卷积核即为神经网络的权重, 卷积层的几何意义相当于神经网络的超平面. 通过卷积层的运算, 把原图像转换到超平面坐标系下. 超平面能把同类的图像尽量地集中起来, 多层卷积层的运算能达到同类图像的合并. 卷积核的大小应该和图像特征的大小相关.

每个卷积核在空间上比较小, 但是深度和输入数据的通道数一致. 在前向传播的过程中, 每个卷积核都在输入数据的宽度和高度上滑动并进行卷积计算, 生成一个2维的特征图(feature map), 激活图给出了卷积核在每个空间位置处的反应. 通过网络训练, 最后会让卷积核“看到”某些类型的视觉特征时被激活, 这个特征可能是颜色斑点、某种形状、纹理等. 将这些卷积核结果激活图进行叠加, 就生成了输出结果.

卷积层的算法过程为:

如图2所示, 假设有输入图像 $A_{3 \times 3}$ 和卷积核 $B_{2 \times 2}$ , 那么 $B$ 卷积 $A$ 的结果就是让 $B$ 在矩阵 $A$ 上滑动, 换言之, 就是 $B$ 与 $A$ 的所有 $2 \times 2$ 连续子矩阵做“对应元素积之和”运算, 所以, 此时的结果 $C$ 应该为:

假设 $A$ 的大小为 $h_a \times w_a$ ,  $B$ 的大小为 $h_b \times w_b$  (其中 $h_a \geq h_b, w_a \geq w_b$ ), 则 $C$ 的大小为 $(h_a - h_b + 1) \times (w_a - w_b + 1)$ , 矩阵 $B$ 称为卷积核(滤波器), 矩阵 $C$ 称为特征图(feature map). 每个卷积核都会将图像生成为另一幅

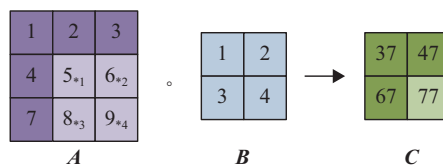


图2 卷积层操作说明

Fig.2 Description of the operation of the convolutional layer

图像. 上述运算称为窄卷积, 若矩阵 $A$ 预先上下各添加 $h_b - 1$ 行零向量, 左右各添加 $w_b - 1$ 列零向量, 再与 $B$ 卷积, 则称为宽卷积. 宽卷积的作用主要是对于图像边缘可以有效计算, 当卷积核大小相对输入图像的尺寸较大时, 宽卷积更加必要. 除此之外借助宽卷积可以让输入大小和输出大小相同, 降低计算复杂度, 这在很多网络中都得到了广泛应用. 窄卷积和宽卷积分别用 $\text{conv2}(A, B, \text{'valid'})$ 和 $\text{conv2}(A, B, \text{'full'})$ 表示.

### 1.1.2 池化层

通常在卷积层之后会进行池化层的操作, 借助池化层来对结果进行压缩, 降低数据的空间尺寸, 减少网络中参数的数量, 使得计算资源耗费变少, 有效控制过拟合. 池化也称为下采样, 用 $S = \text{down}(C)$ 表示. 常见的池化操作主要有最大池化、平均池化等, 其中应用最广泛的是最大池化. 最大池化是对输入数据体的每一个深度切片(单个卷积核生成的二维结果)取最大化的操作. 最常见的池化操作, 如图3所示, 从每个 $2 \times 2$ 区域内取最大值, 过滤掉其中75%的信息.

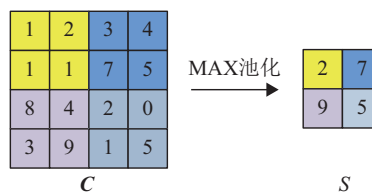


图3 池化层操作说明

Fig.3 Description of the operation of the pooling layer

最大池化对局部区域取其极大值, 有一定概率在尺度变化后对应的局部区域取到的极大值不变, 这样就可以保证特征图不变. 在图像识别等任务中, 重要的不是显著特征的绝对位置而是相对的位置, 所以为了避免把过多的位置信息编码进去, 卷积和池化的操作都可以对局部的纹理进行模糊化, 这样也就使图像有了一定的形状不变性.

## 1.2 卷积神经网络的训练过程

CNN的训练过程就是借助反向传播算法更新神经元权值的过程. CNN在本质上是一种输入到输出的映射, 通过学习大量的有输入与输出的样本, 用卷

积网络进行训练,网络就可以拟合输入和输出之间的映射能力。

CNN的训练是监督学习(有样本),其样本集是由输入和输出数据构成的训练对。训练过程主要分为两个阶段:前向传播阶段、后向传播(网络权值更新)阶段。前向传播阶段完成数据输入到结果的输出,这个过程也是网络在完成训练后进行测试时执行的过程;后向传播是根据输出误差和权重修改公式,进行误差的反向传递,借此更新网络权重。网络训练过程如下:

(1) 选定训练集,从样本集中随机地寻求 $N$ 个样本作为训练样本。

(2) 对各权值、阈值、精度控制参数和学习率进行初始化。

(3) 从训练样本集中取一个样本的输入值到网络,计算网络各层输出结果向量。

(4) 将网络的计算输出与样本的实际输出相比较,计算误差。利用误差修正连接的权值。

(5) 对于中间卷积层的误差,是由上层的误差反向传播过来,再用此层误差来修正卷积核的值(网络权值),即按反向权值的修正公式进行修改。

(6) 依次反向修正各层网络权值和偏置(阈值)。

(7) 对所有样本都训练一遍(从输入到输出计算误差,用误差反向修改网络权值)。

(8) 经历大量的迭代后,判断计算输出与样本输出的差是否满足精度要求,如果不满足,则返回(3),继续迭代,如果满足就进入下一步。

(9) 训练结束,保存网络的权值和偏置。这时可以认为各个权值已经达到稳定,网络训练已经完成。

以2个卷积层和2个池化层的卷积神经网络为例,其结构如图4所示,接下来对于卷积的反向传播过程进行详细解释。设 $C$ 为当前层输出, $f(*)$ 为激活函数, $b$ 为偏置, $a$ 为卷积层或池化层的图元素。

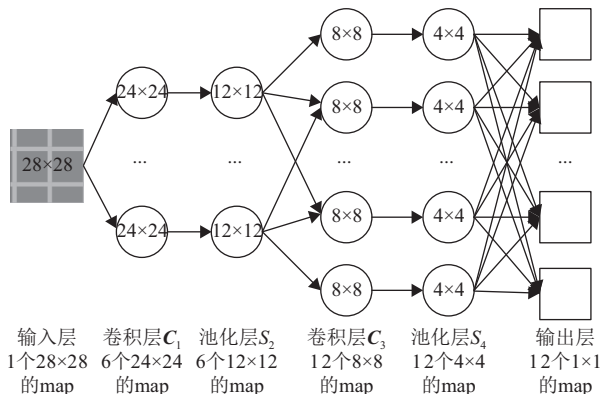


图4 一个简单的卷积神经网络

Fig.4 A simple example of CNN

### 1.2.1 前向网络计算

(1) 卷积层的网络计算。

$C_1$ 层:输入是 $28 \times 28$ 的数据矩阵 $A$ ,与 $F_1$ 个大小为 $5 \times 5$ 的窄卷积核 $K_i^1 (i = 1, 2, \dots, F_1)$ 进行卷积,产生 $F_1$ 个 $24 \times 24$ 大小的特征图(feature maps)。

$C_3$ 层:再次进行卷积, $S_2$ 中的所有 $F_1$ 个池化图经过 $F_3$ 个 $5 \times 5$ 的卷积核 $K_{ij}^3 (j = 1, 2, \dots, F_1)$ ,生成 $F_3$ 个特征图。

以 $C_1$ 层为例,计算公式为

$$\begin{cases} C_i^1 = \text{conv2}(A, K_i^1, \text{'valid'}) + b_i^1; \\ u_i^1 = C_i^1; \\ a_i^1 = f(u_i^1). \end{cases} \quad (1)$$

其中, $C_i$ 为当前层的卷积结果, $a_i$ 为下一层的输入(卷积结果经过激活函数后的结果)。

(2) 池化层的网络计算。

$S_2$ 层:卷积后进行池化操作,池化窗口大小为 $2 \times 2$ ,每一个 $24 \times 24$ 的特征图生成一个 $12 \times 12$ 的池化图,共生成 $F_1$ 个池化图。

$S_4$ 层:再次进行池化操作,池化窗口大小为 $2 \times 2$ ,每一个 $8 \times 8$ 的特征图池化成一个 $4 \times 4$ 的池化图,共生成 $F_3$ 个池化图。

以 $S_2$ 层为例,前向传播过程计算如下

$$\begin{cases} S_i^2 = \beta_i \text{down}(a_i^1) + b_i; \\ u_i^2 = S_i^2; \\ a_i^2 = f(u_i^2). \end{cases} \quad (2)$$

(3) 全连接层的计算。

将 $a_i^4 (i = 1, 2, \dots, F_3)$ 按顺序展开成一维向量,并有序连接作为全连接层的输入,按照反向传播神经网络BP算法公式计算输出。

$$\begin{cases} z_i^5 = w \times a^4 + b^5; \\ u_i^5 = z_i^5; \\ a_i^5 = f(u_i^5). \end{cases} \quad (3)$$

### 1.2.2 反向误差传播

此处我们是要找到最佳的权重( $w$ )和偏置( $b$ )使得损失函数的值最小,因此损失函数 $J$ 是关于 $w$ 和 $b$ 的函数,其中 $w$ 和 $b$ 是网络中所有权重和偏差的集合。因此,假设网络的损失函数为

$$\begin{cases} J(w, b) = \frac{1}{m} \sum_{i=1}^m J(w, b; x^{(i)}, y^{(i)}); \\ J(w, b; x^{(i)}, y^{(i)}) = \frac{1}{2} (y^{(i)} - h_{w, b}(x^{(i)}))^2. \end{cases} \quad (4)$$

$x^{(i)}, y^{(i)}$ 分别为输入和样本输出。 $h(x)$ 是网络对 $x$ 的计算结果。



卷积神经网络的反向传播过程采用了BP神经网络的网络计算公式,区别在于卷积神经网络只连接了一部分神经元;而BP神经网络中相邻两层的神经元与另一层的所有神经元全部相连。

#### (1) 全连接层的权值修正。

卷积神经网络全连接层的反向求导与BP神经网络的反向求导一致。

$$\delta^{(5)} = \frac{\partial J}{\partial z^5} = \frac{\partial}{\partial z}(y - h(x))^2 = \frac{\partial}{\partial z^5}(y - f(z^5))^2 = (y - f(z^5))f'(z^5). \quad (5)$$

其中 $\delta$ 为残差,等于网络计算的输出与样本输出的差值与激活函数的导数的乘积。

在全连接神经网络中,残差在BP神经网络中用于修正网络权值,并将输出层残差反向传播到下层。

#### (2) 池化层的处理。

池化层是在卷积层中应用求值函数(求最大值或平均值)得到的,这样就不存在权值修正问题。由于它缩小了卷积层的矩阵维度,为此需要放大它的矩阵维度,进行上采样up运算,也称克罗内克积。若采样窗口为 $2 \times 2$ ,up为

$$\text{up}\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right) = \begin{bmatrix} 1 & 1 & 2 & 2 \\ 1 & 1 & 2 & 2 \\ 3 & 3 & 4 & 4 \\ 3 & 3 & 4 & 4 \end{bmatrix}. \quad (6)$$

在图4中,全连接层 $S_4$ 是池化层,它的误差由输出层误差反向传播而来,按BP网络公式计算。在池化层的维度扩大以后,原节点误差也随之扩散到相同值节点。

$S_4$ 池化层的残差 $\delta$ 反传给卷积层 $C_3$ 的公式是:扩展维度后 $S_4$ 池化层的残差 $\delta$ 乘以卷积层 $C_3$ 的输出,得到卷积层 $C_3$ 的残差 $\delta$ 。

以图5为例进行说明。

0.1	0.1	0.2	0.2	×	1	1	1	1	=	0.1	0.1	0.2	0.2
0.1	0.1	0.2	0.2		0	0	1	1		0	0	0.2	0.2
0.4	0.4	0.5	0.5		0	1	1	0		0	0.4	0.5	0
0.4	0.4	0.5	0.5		0	1	1	0		0	0.4	0.5	0

扩展后的采样层  
l+1的残差

卷积层l的输出

卷积层l+1的残差

图5 卷积层的残差计算

Fig.5 Weight decay of convolutional layers

对于 $S_2$ 池化层 $l$ ,下一层 $l+1(C_3)$ 为卷积层,池化层 $l$ 的各个神经元的 $\delta$ 只和 $l+1$ 层的部分相关神经元有关,池化层 $l$ 到卷积层 $l+1$ 的过程中做了窄卷积运算,使得矩阵维度减小,因此, $\delta_i^{(l+1)}$ 需要与相应的卷积核进行宽卷积运算,以扩展矩阵维度。

#### (3) 卷积层的修正。

对于卷积层而言,它是通过中间卷积核的网络权值运算得来。权值的修正是利用卷积层的误差,按反向神经网络权值修正公式来修改。卷积的激励函数一般用ReLU函数。

权值和偏置(阈值)修正公式为

$$\begin{cases} w^{(l)} = w^{(l)} - \alpha \frac{\partial J}{\partial w^{(l)}}; \\ b^{(l)} = b^{(l)} - \alpha \frac{\partial J}{\partial b^{(l)}}. \end{cases} \quad (7)$$

其中 $\alpha$ 是学习率系数。

### 1.3 算法讨论

深度学习的模型在应用的过程中,最重要的也是最难的便是参数的设计,为此,针对卷积神经网络的算法及参数进行讨论。

#### (1) 样本数据的获取和预处理。

训练神经网络首先需要获取高质量的输入和输出样本数据集。需要对数据进行标准化处理,一般使数据均值为0,每个维度上的方差为1。当输入数据在不同维度上差异较大时尤其需要归一化操作。这样可以避免小数被忽略;同时消除不同数据维度之间的差异,加快训练算法的收敛速度。

#### (2) Mini-batches。

训练过程中同时进行128个样本的训练是有效的。Mini-batches方法是同时对多个(如128个)训练样本的输入进行前向传播,在最后计算误差时,将多个误差求均值后作为本次训练的误差进行后向传播。设置过大的Mini-batches会导致效果不好,而且容易过拟合。

#### (3) 学习率的设定。

学习速率设置太小,会收敛过慢;学习速率太大,会造成误差函数振荡。学习率设定一个初始学习率,然后依次减小。这样既可以保证收敛速度,也可以提高训练效果。

初始学习率设定后(通常为0.1),训练过程中验证集正确率不再变化时,将学习率除以2(或5),继续训练,最终结束时学习率将非常小。这样的策略也可以有效避免过拟合。

#### (4) 权重的意义和初始化。

常用的方法是高斯随机初始化:将权重初始化为均值0,方差为0.01的随机数。偏置通常初始化为0,权重不同后,偏置可以相同。

### 1.4 实例分析

下面我们以VGGNet<sup>[3]</sup>为例,做进一步的说明。

VGGNet由牛津大学的视觉几何组提出,是ILSVRC-2014中定位任务第一名和分类任务第二名,其后,在大量的其他应用中,很多网络结构都是在其基础上的改进,例如目标检测网络(fast、faster)R-CNN<sup>[6-8]</sup>,文字检测网络CTPN<sup>[9]</sup>等。

VGGNet有6种不同结构,其中最常用的就是VGG16,如表1所示,该网络包含13个卷积层,5个最大池化层和3个全连接层,其中每个卷积层后面都有一个激活函数层。这里采用的激活函数是ReLU函数,池化层采用最大池化,这也是当前最常用的设置。通常以池化操作为分界线,将结构分为6个阶段,前5个阶段为特征提取阶段,最后一个阶段为分类阶段。

表1 VGG16网络结构说明  
Tab.1 Structure of VGG16

输入	输入(224×224RGB图像)
卷积层	size:3, pad:1, stride:1, num:64
卷积层	size:3, pad:1, stride:1, num:64
池化层	MAX, size:2×2, stride:2
卷积层	size:3, pad:1, stride:1, num:128
卷积层	size:3, pad:1, stride:1, num:128
池化层	MAX, size:2×2, stride:2
卷积层	size:3, pad:1, stride:1, num:256
卷积层	size:3, pad:1, stride:1, num:256
卷积层	size:3, pad:1, stride:1, num:256
池化层	MAX, size:2×2, stride:2
卷积层	size:3, pad:1, stride:1, num:512
卷积层	size:3, pad:1, stride:1, num:512
卷积层	size:3, pad:1, stride:1, num:512
池化层	MAX, size:2×2, stride:2
卷积层	size:3, pad:1, stride:1, num:512
卷积层	size:3, pad:1, stride:1, num:512
卷积层	size:3, pad:1, stride:1, num:512
池化层	MAX, size:2×2, stride:2
全连接层	num:4096
全连接层	num:4096
全连接层	num:1000
Soft-max层	获得分类结果

该网络设计的主要目的是完成图像分类任务,网络的输入是包含多个类别物体(比如猫、狗、船等)的大量不同姿态的图像,给定的输出结果是物体类别向量,每个类别物体用一个神经元的(0,1)表示。使用的数据是ImageNet数据集。

首先,图像在输入前需要进行一定预处理:每个像素减去RGB的均值,并将图像大小缩放到固定大小(224×224)。

随后,图像数据输入到网络后,借助一系列的卷积和池化操作进行特征的提取。网络基本都是采用3×3的卷积核。填充为1,既保证计算前后大小不变,又充分考虑图像边缘信息。

网络中间层结果如图6所示,从可视化结果可以看出,图像经过卷积操作能够对图像中的特征进行自动汇集与提取,随着卷积进行,特征越来越趋于抽象。前两阶段还可以看出浅层的边缘信息,随着网络的深入,特征趋于抽象,包含的更多是隐藏于图像背后的语义特征。

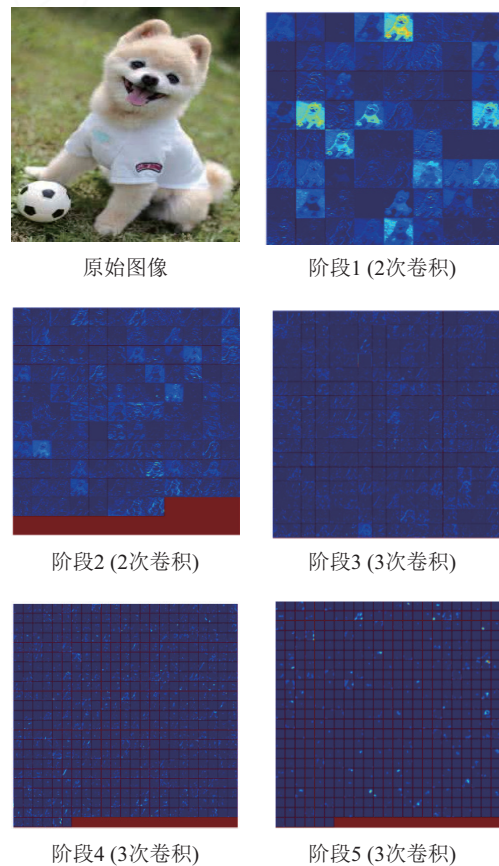


图6 网络中间层结果可视化

Fig.6 The visualized results of the middle layers

特征提取结束后,进一步完成分类任务。卷积操作是与上一层节点进行局部连接,关注的是局部信息;而图像分类需要综合整个图像的全局信息,就需要借助全连接层。全连接层与前一层所有节点相连,可以对全局特征进行融合,采用多个全连接层可以充分融合图像特征,将学到的特征映射到样本标记空间。这里采用的是传统的全连接层,参数个数占总体近90%。在后续的研究应用中,人们倾向于采用1×1卷积替代该全连接层,在一些模型中全连接层更是直接被去掉。

最后借助soft-max层完成分类,确定出图像所属

类别. 其中, soft-max分类器是逻辑回归二分类器泛化到多分类的情形, 可以对多种互斥类别进行分类.

训练时设置的Mini-batches大小为256, 初始学习率为0.01, 当验证集结果不再上升时, 学习率除以10. 权重初始化采用的方法是均值为0, 方差为0.01的高斯分布. 需要注意的是, 由于训练耗时等问题, 在应用该模型解决其他问题时, 一般用该训练好的模型进行初始化, 在此基础上做进一步训练, 这样可以大大提高执行效率.

深度学习模型相对于传统方法的最大特点就在于其强大的特征提取能力. 随着网络层次越来越高, 表达能力越来越强, 层次性的结构使其能够自底向上地逐步得到更高层次的抽象特征, 综合各种底层因素之间的复杂关系, 具有很好的不变性, 处理更复杂的任务. 经过大量实验证明, 该模型很好地发挥了深度学习的这一特征, 具备强大的特征提取能力. 随后的大量研究都采用该结构的特征提取结构, 然后针对不同任务, 对结果进行识别、检测等操作.

## 2 卷积神经网络的进展

在卷积神经网络取得一系列成功的同时, 一些问题也逐渐出现, 相应的也推动着卷积神经网络的进一步发展完善.

### 2.1 池化层的讨论

特征提取的误差主要来自两个方面: (1) 邻域大小受限造成的估计值方差增大; (2) 卷积层参数误差造成估计均值的偏移. 平均池化能一定程度上减小第一种误差, 保留图像更多的背景信息, 而最大池化能减小第二种误差, 更多地保留纹理信息. 为了更有效地处理这些误差, 研究者们提出了许多新的池化方法. 例如, 随机池化<sup>[10]</sup>对于邻域内像素点, 按照数值大小赋予概率, 按照概率进行降采样, 平均意义上与平均池化相似, 局部意义上与最大池化相近, 一定程度上对以上两种误差都能有效抑制.

### 2.2 激活函数的讨论

常见的激活函数包括sigmoid、tanh、ReLU等. 近年来, ReLU因其线性的特性且计算简单, 在众多应用场景都取得了很好的效果. 但ReLU在训练的过程中对于取值小于0的神经元进行舍弃, 所以此激活函数也相对“脆弱”, 有可能导致神经元“坏死”.

Leaky ReLU: 就是用来解决ReLU坏死的问题的. 如图7(b)和ReLU不同, 当 $x < 0$ 时, 它不再是0, 而是一个较小斜率(如0.01)的函数.

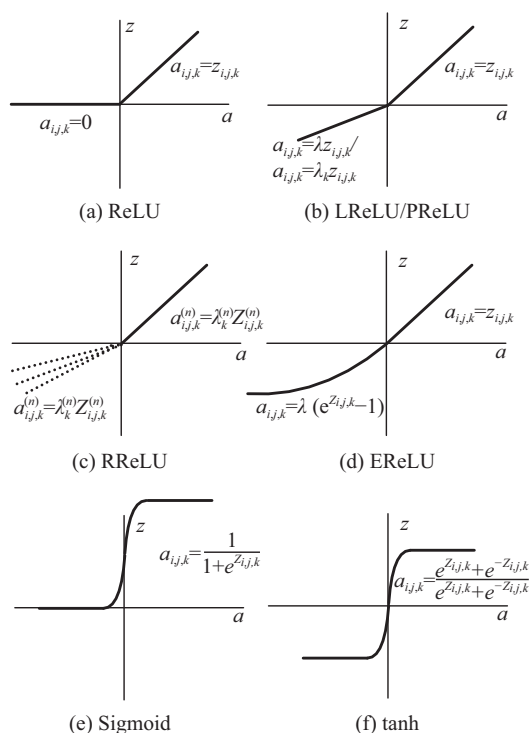


图7 几种激活函数

Fig.7 Several activation functions

ELU<sup>[11]</sup>: 融合了sigmoid和ReLU, 具有左侧软饱和性, 不仅保持了ReLU快速训练的优点, 而且使得对于噪声承受能力更强, 如图7(d)所示.

### 2.3 优化过程的讨论

Dropout<sup>[12]</sup>: 在网络训练中, 很容易出现过拟合的现象, 即在训练集上表现良好, 但在测试集上效果一般. 一般解决过拟合问题主要借助多个模型进行组合训练, 但是这样会让训练非常耗时. Dropout在CNN训练过程中, 对于神经网络单元, 按照一定的概率将其暂时从网络中丢弃, 可以看作多个模型进行组合训练, 但训练参数不变, 训练时间问题也可以得到解决.

快捷连接(Shortcut connections)<sup>[13]</sup>: 在网络的训练过程中, 加入数据层的信息, 综合数据层和当前层结果可以有效提高训练效率, 尤其是对于层数较深的网络效果尤其明显. 每层的输入不仅是上一层的输出, 还包括数据层的信息. 借助数据层的先验信息可以有效提高训练效率, 尤其是对于层数较深的网络效果尤其明显.

权重初始化方法. 权重初始化就是网络中所有神经元权值(包括阈值)赋予一个初始值. 传统的方法有高斯法等, 最近的方法有如文献[14]提出的Xavier方法和文献[15]提出的MSRA方法, 在训练的收敛速度和避免梯度消失等方面都有一定提升.



### 3 深度学习中算法的讨论

卷积神经网络CNN在深度学习中取得了显著的效果,在图像分类、目标检测、人脸识别等方面得到广泛应用.在著名的图像分类比赛(ILSVRC)中,采用的数据集是ImageNet数据集的子集(训练集126万张图像,验证集5万张图像,测试集10万张图像). (1) 2012年Alex Krizhevsky等首次将CNN应用于大规模图像分类,并取得了16.4%的错误率,比第2名(传统图像分类方法,即借助SIFT算法和支持向量机SVM算法)的低了大约10%. CNN成为图像分类领域的主流方法. (2) Zeiler等提出的反卷积方法改进AlexNet将错误率降到11.7%. (3) 微软亚洲研究院团队的基于空间金字塔池化的SPPNet<sup>[16]</sup>获得的错误率是8.1%.

### 4 总结与展望

CNN在许多领域取得了巨大的成功,仍然有一些问题需要进一步研究: (1) 深度学习面临的最大的问题是运行原理不完善,算法中采用了一些试探性计算; (2) 参数的设定一直是CNN应用过程中的一个主要的难点,当前主要借助经验设置; (3) 网络层数的增加,使参数越来越多,对硬件的要求越来越高,计算耗时也越来越久; (4) CNN的训练需要对大量样本数据的运算,因此,无监督学习CNN的研究也值得深入研究.

#### 参考文献:

- [1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//International Conference on Neural Information Processing Systems.[S.l.]: [s.n.], 2012: 1097-1105.
- [2] ZEILER M D, FERGUS R. Visualizing and understanding convolutional networks[C]//[s.n.]. European conference on computer vision.[S.l.]: Springer, Cham, 2014: 818-833.
- [3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv, 2014: 1409. 1556.
- [4] SZEGEDY C, LIU W, JIA Y, *et al.* Going deeper with convolutions[C]//IEEE. Proceedings of the IEEE conference on computer vision and pattern recognition.[S.l.]: [s.n.], 2015: 1-9.
- [5] HE K, ZHANG X, REN S, *et al.* Deep residual learning for image recognition[C]//IEEE. Proceedings of the IEEE conference on computer vision and pattern recognition.[S.l.]: [s.n.], 2016: 770-778.
- [6] GIRSHICK R, DONAHUE J, DARRELL T, *et al.* Rich feature hierarchies for accurate object detection and semantic segmentation[C]//IEEE. Proceedings of the IEEE conference on computer vision and pattern recognition.[S.l.]: [s.n.], 2014: 580-587.
- [7] GIRSHICK R. Fast r-cnn[C]//IEEE.Proceedings of the IEEE international conference on computer vision.[S.l.]: [s.n.], 2015: 1440-1448.
- [8] REN S, HE K, GIRSHICK R, *et al.* Faster R-CNN: Towards real-time object detection with region proposal networks [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.
- [9] TIAN Z, HUANG W, HE T, *et al.* Detecting text in natural image with connectionist text proposal network[C]//European Conference on Computer Vision.[S.l.]: Springer International Publishing, 2016: 56-72.
- [10] ZEILER M D, FERGUS R. Stochastic pooling for regularization of deep convolutional neural networks[J]. arXiv preprint arXiv, 2013: 1301. 3557.
- [11] CLEVERT D, UNTERTHINER T, HOCHREITER S, *et al.* Fast and accurate deep network learning by exponential linear units (elus)[J]. arXiv preprint arXiv, 2015: 1511. 07289.
- [12] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, *et al.* Dropout: a simple way to prevent neural networks from overfitting [J]. Journal of Machine Learning Research, 2014, 15(1): 1929-1958.
- [13] SRIVASTAVA R K, GREFF K, SCHMIDHUBER J, *et al.* Training very deep networks[C]//Advances in neural information processing systems.[S.l.]: [s.n.], 2015: 2377-2385.
- [14] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feed forward neural networks [J]. Journal of Machine Learning Research, 2010, 9: 249-256.
- [15] HE K, ZHANG X, REN S, *et al.* Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]//IEEE. Proceedings of the IEEE international conference on computer vision.[S.l.]: [s.n.] 2015: 1026-1034.
- [16] HE K, ZHANG X, REN S, *et al.* Spatial Pyramid Pooling in deep convolutional networks for visual recognition [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2014, 37(9): 1904-1916.