

第5章

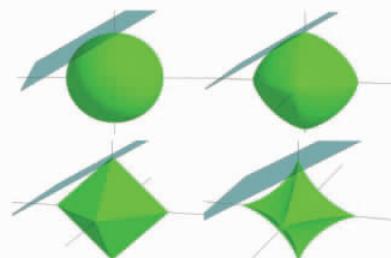


稀疏深度神经网络

CHAPTER 5

深度神经网络中的稀疏表达

- 数据中的稀疏性——稀疏表示假设
- 权值连接矩阵的稀疏性——稀疏技巧
- 激活函数的稀疏性——内蕴稀疏性
- 隐层输出的稀疏性——稀疏正则
- 稀疏模式识别——稀疏分类器设计
- 稀疏初始化策略——稀疏编码、自编码网络





5.1 稀疏性的生物机理

深度学习与稀疏认知学习、计算与识别之间的关系深刻而且本质,从机器学习中的特征工程(人工特征提取与特征筛选)到深度学习中的特征学习(通过线性与非线性操作的不断复合获取数据的高层统计或语义特性),无论是以显性还是隐性的嵌入方式,稀疏性都在模型中扮演着重要的角色。下面简要从生物视觉机理和数学物理角度来描述稀疏性。

备注:稀疏认知学习、计算与识别的范畴包括表示理论(即基于稀疏表示的压缩感知和稀疏编码),数学计算(最优匹配追踪算法)和模式识别(稀疏表示分类器 SRC 及稀疏分类器设计 SparseMax)等。

5.1.1 生物视觉机理

视觉感知机理的研究表明,视觉系统可以看成一种合理而且高效的图像处理系统,从视网膜到大脑皮层存在一系列具有不同生物学功能的神经细胞,例如随着层级信息不断的“加深”,不同视觉皮层上的神经细胞对特定形状的视觉图案有最佳的响应和偏好的刺激,简言之,层级越高感受野越大,即信息处理从局部到更大的区域,类似尺度特性。层级较低时,感受野所处理的区域越小,稀疏性越强(特指层级间的连接特性),层级较高时,感受野所处理的区域越大,稀疏性越弱。另外,Barlow 推论出在稀疏性和自然环境的统计特性之间必然存在某种联系,随后诸多基于生物视觉和计算的模型被提出来,都成功地例证了生物视觉针对自然环境所反馈出的物理统计特性蕴含着稀疏性。当层级较低时,其简单细胞对应着严格的方向和带通特性,而复杂细胞在保持简单细胞特性的基础上进一步具有局部变换(如平移)不变性,简言之,简单细胞处理信息具有稀疏(即局部连接)特性,而复杂细胞具有聚类(连接计算共享)特性。神经科学研究成果表明,稀疏编码是视觉系统中图像表示的主要方式,初级视觉皮层(V1 区)中的神经元对视觉信息的反应具有稀疏性,V4 区的神经元通过稀疏编码的方式实现视觉信息的表示。从表 5.1 中可知,随着对计算机视觉研究的深入,人类对自身视觉感知系统的理解也在不断加深。借鉴生物视觉机理的研究成果,模拟建立相应的视觉计算模型,将成为一个极具挑战性和吸引力的研究方向。下面给出生物(人类)视觉与计算机视觉的对比表(表 5.1)。

表 5.1 生物(人类)视觉与计算机视觉对比

对比项	人类视觉	计算机视觉
适应性	适应性强,可在复杂及变化的环境中识别目标	适应性差,容易受复杂背景及环境变化的影响
智能	具有高级智能,可运用逻辑分析及推理能力识别变化的目标,并能总结规律	虽然可利用人工智能及神经网络技术,但智能很差,不能很好地识别变化的目标



续表

对比项	人类视觉	计算机视觉
彩色识别能力	对色彩的分辨能力强,但容易受人的心 理影响,不能量化	受硬件条件的制约,目前一般的图像采集系 统对色彩的分辨能力较差,但具有可量化的优点
灰度分辨能力	差,一般只能分辨 64 个灰度级	强,目前一般使用 256 灰度级,采集系统可具有 10bit、12bit、16bit 等灰度级
空间分辨能力	分辨率较差,不能观看微小的目标	目前有 $4K \times 4K$ 的面阵摄像机和 $8K$ 的线阵摄 像机,通过备置各种光学镜头,可以观测小到微 米大到天体的目标
速度	0.1 秒的视觉暂留使人眼无法看清较 快速运动的目标	快门时间可达到 10 微秒左右,高速相机帧率可 达到 1000 以上,处理器的速度越来越快
感光范围	400~750nm 范围的可见光	从紫外到红外的较宽光谱范围,另外有 X 光等 特殊摄像机
环境要求	对环境温度、湿度的适应性差,另外有 许多场合对人有损害	对环境适应性强,另外可加防护装置
观测精度	精度低,无法量化	精度高,可到微米级,易量化
其他	主观性,受心理影响,易疲劳	客观性,可连续工作

另外,关于生物视觉与计算机视觉之间核心的模块对应关系见图 5.1,值得注意的是:理解并分析大脑是如何在算法层面上工作的尝试是鲜活且发展良好的,这项尝试被称为“计算神经科学”,并且是独立于深度学习的一个领域。研究人员两个领域间反复研究是很常见的,深度学习主要关注如何构建智能的计算机系统,以用来解决需要智能才能解决的任务,而计算神经科学领域主要是关注构建大脑如何工作的更精确的模型。

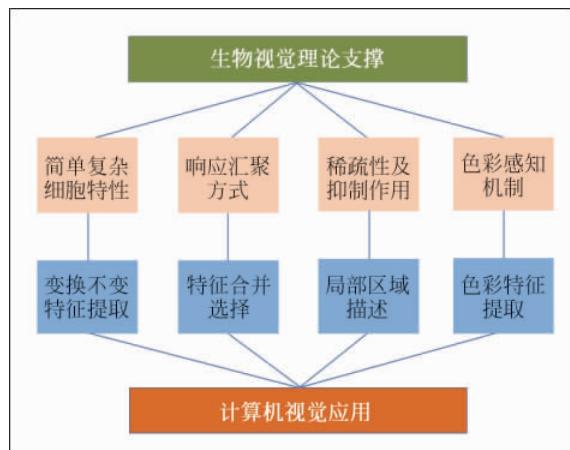


图 5.1 生物视觉与计算机视觉核心模块对应



5.1.2 稀疏性响应与数学物理描述

目前,构建高性能的计算模型,并不是模型越复杂越好,特别针对变量维数很高,样本量不是很大的情形下,构建一个合理的,相对简单的稀疏模型往往具有更高的性能,更为重要的是还具有生物可解释性。从数学角度来看,依据模型的低复杂性结构(如向量的稀疏性,矩阵的低秩性等),如何高效地从病态的线性逆问题中唯一且稳健地恢复出特定的信息。值得指出的是:常见的稀疏性是指向量中绝大多数元素的值为零或者接近于零;而广义的稀疏性是指通过特定变换后目标的稀疏性。可以看出,当前为了使得模型具备学习能力、高容量的表达能力、快速推断能力以及多任务信息共享能力;借鉴生物视觉的认知机理已成为一种必然趋势。众所周知,1996年Olshausen和Field在Nature杂志上发表的一篇重要论文指出,自然图像经过稀疏编码后得到的基函数类似于初级视觉皮层V1区上简单细胞感受野的反应特性(即空间域的局部性、时域和频域的方向性和选择性)。需要指出的是稀疏编码与稀疏表示是不同的,例如关于系数的稀疏性约束,前者采用光滑可导的函数,而后者采用伪范数或L1范数;另外稀疏编码不要求基原子个数一定要大于数据的维数。本节更为详细的论述与解释请参考第1章的稀疏表示,另外稀疏编码部分可参考相关论文,这里不再赘述。

5.2 稀疏深度网络模型及基本性质

在深度神经网络引入显式稀疏性之前,关于稀疏模型的研究就已经成为机器学习中的热点,特别是针对线性稀疏模型的研究,如压缩感知、双稀疏模型、结构化稀疏模型(如群稀疏)、S-HMAX模型、SRC模型等。当然,除了显式稀疏性(如稀疏正则化理论等)外,还有隐式稀疏性的研究,它通常内蕴在非线性激活函数和损失函数(如交互熵,非L2范数下的能量损失)的构建过程中。众所周知,自从2006年至今深度神经网络的一个重要体现或要求便是训练数据量的规模要大(衡量标准可利用模型的参数个数与训练数据量的个数来比较),由于以往训练数据集规模很小,加上计算性能很慢(硬件加速设备导致),同时权值矩阵的初始化方式较为笨拙(容易出现梯度弥散现象),以及使用了某种错误的非线性模型,导致深度神经网络在过去的表现并不好。经过十余年的积累,目前深度神经网络可简略地认为是大规模训练数据集,并行计算和规模化、灵巧的算法三者的结合。深度神经网络中引入稀疏正则或蕴含稀疏性可以认为是病态模型良态化的过程,如稀疏正则的核心是解决过拟合问题,稀疏权值连接(DropOut策略)的本质是通过约减参数量间接增加训练数据,以及非线性激活函数中所隐含的稀疏性是为了增加“扭曲”程度,即不同类别的(线性不可分)输入随着层级的增加,隐层特征所对应的线性可分性逐渐增强。下面简要地分析深度神经网络在各阶段所出现的稀疏性及其优势。



备注：S-HMAX 为稀疏层次识别模型，SRC 为稀疏表示分类器，结构化稀疏模型，基于稀疏正则的设计有群稀疏、图稀疏、随机场稀疏等。

5.2.1 数据的稀疏性

数据的稀疏性包含三点：一是数据中所包含某种拓扑特性或目标相对数据本身呈现出非零元素较少的情形；二是数据在某种（线性或非线性）自适应或非自适应变换下对应的表示系数具有非零元素较少的状况；三是随着数据集规模的增加，呈现出某种统计或物理特性的数据占整个数据集的少数，例如分辨率特别好的样本或分辨率特别差的样本在整个数据集中呈较少的状态。目前，常用的稀疏性描述是基于第二点假设，并且作为一种有效的（稀疏性）正则约束，在优化目标函数关于解存在多样性的问题中给出合理的解释与逼近。而基于第一点，通常可作为一种有效的处理方式（如二值化处理，或者零化无关区域），例如输入到深度神经网络中的一幅图像，有效的目标占图像的比例较少，便可以将图像中除去目标的部分置为零；值得注意的是：利用视觉机制中的显著性检测方法。另外针对第三点，其核心问题是利用稀疏编码筛选出这些重要样本（或剔除少数样本）。从框架（Frame Analysis）分析角度，认为比较好的冗余框架应该是紧框架，进而对输入描述便可以得到较好的紧表示系数，也就是说框架上界和框架下界尽可能相等。但是通常获取到的字典，也就是框架，不是紧的，能否利用大量无类标样本将框架的上界与下界估计出来，然后利用输入信号的逼近表示的二范数比上表示系数的二范数，看这个比值是否在框架上界与下界的中间，来判断该样本对字典（框架或系统）的表示是否是 well-defined 的，进而实现对样本的有效筛选。

备注：本小节讲的框架，是数学分析中的一支理论，继傅里叶分析、时频分析和小波分析之后，框架分析被提出，它指带有冗余特性“基”的表示理论。

5.2.2 稀疏正则

众所周知，正则化的目的在于减少学习算法的泛化误差（亦称测试误差）以期提高测试识别率。目前，有许多正则化策略，常用的方式是对参数进行约束或限制，以及基于某种特定类型的先验知识进行约束与惩罚设计，注意这些惩罚和约束通过将模型求解参数良态化的过程来实现泛化性能的提升。基于如下的优化目标函数：

$$\min_{\theta} J(\theta) = \frac{1}{N} \sum_{n=1}^N \text{loss}(\mathbf{x}^{(n)}, \mathbf{y}^{(n)}, \theta) + \lambda \cdot R(\theta) \quad (5.1)$$

其中的 $R(\theta)$ 为参数范数惩罚，例如常用的有 L_2 范数下的吉洪诺夫正则（Tikhonov Regularization），但它并没有蕴含稀疏特性。而使用 L_1 范数则通常可以诱导出稀疏特性，即

$$R(\theta) = \|\mathbf{W}\|_1 = \sum_i |W_i| \quad (5.2)$$



注意参数 θ 包括权值连接 \mathbf{W} 与偏置 b , 而正则约束往往只针对权值连接。除了在权值连接上引入稀疏正则外, 还可以在某个隐层输出层引入稀疏性, 例如对于如下的目标函数:

$$\min_{\vartheta} J(\vartheta) = \| \mathbf{x} - \mathbf{D} \cdot \vartheta \|_2^2 + \lambda \cdot \| \vartheta \|_1 \quad (5.3)$$

注意这里的 \mathbf{D} 为字典, 数学中称其为框架, 即有冗余的“基”; x 为输入, ϑ 为输出, 其 L_1 范数的定义与式(5.2)对应。值得指出的是反卷积神经网络中的卷积稀疏编码可以认为是一种带有共享机制下的权值稀疏性约束策略。

备注: 除了上述具有稀疏特性的 L_1 范数外, 还可以引入群稀疏的策略, 以及伪范数 $L_{1/2}$ 等, 这里不再赘述。

5.2.3 稀疏连接

众所周知, 卷积神经网络的特性包括局部连接, 权值共享和变换不变等特性且都蕴含着稀疏性, 首先针对局部连接, 相比较全连接策略, 它更符合外侧膝状体到初级视觉皮层上的稀疏响应特性; 其次权值共享, 进一步约束相似隐单元具有同样的激活特性, 使得局部连接后的权值具有结构特性, 实际应用中可进一步约减参数个数, 间接增加数据量; 最后, 变换不变性是由池化方式诱导获取, 也可认为是一种有效的“删减”参数的方式, 即带有稀疏性的零化操作。下面介绍一种经典的自适应权值删减技巧 DropOut, 即指在模型训练时随机让网络某些隐含层节点的权重不工作, 不工作的那些节点可以暂时认为不是网络结构的一部分, 但是它的权重需保留下来(注意只是暂时不更新), 因为下次样本输入时它可能又得工作了, 见图 5.2。

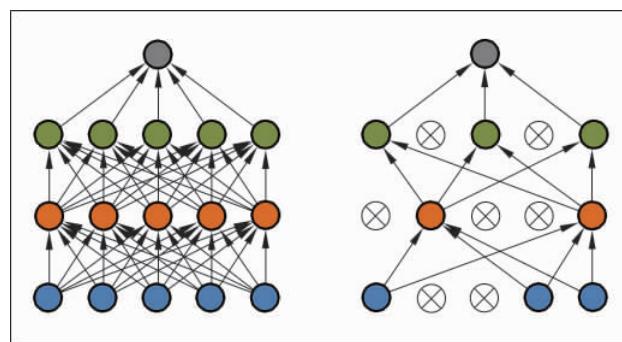


图 5.2 Dropout 网络连接

在图 5.2 的基础上, 对第 l 层到第 $l+1$ 层上的第 i 个隐单元, 在训练阶段, Dropout 具体的工作原理如图 5.3 所示。

其中图 5.3 中左边的网络结构为正常的连接, 右边的为带有 Dropout 策略的连接, 其数学物理解释如下。

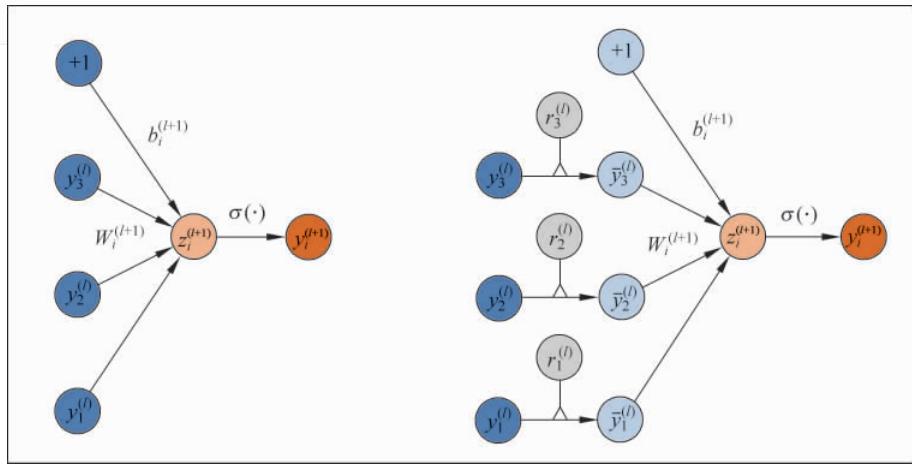


图 5.3 Dropout 的工作原理

1. 正常的连接

$$\begin{cases} z_i^{(l+1)} = \mathbf{W}_i^{(l+1)} \cdot \mathbf{y}^{(l)} + b_i^{(l+1)} = \sum_{j=1}^3 \mathbf{W}_{i,j}^{(l+1)} \cdot y_j^{(l)} + b_i^{(l+1)} \\ y_i^{(l+1)} = \sigma(z_i^{(l+1)}) \end{cases} \quad (5.4)$$

注意其中权值连接为 $\mathbf{W}_i^{(l+1)} \in \mathbb{R}^3$, 另外 $b_i^{(l+1)} \in \mathbb{R}$ 为偏置, $\sigma(\cdot)$ 为激活函数。

2. 带有 Dropout 策略的连接

$$\begin{cases} r_j^{(l)} \sim \text{Bernoulli}(p) \\ \tilde{\mathbf{y}}^{(l)} = \mathbf{r}^{(l)} \odot \mathbf{y}^{(l)} \in \mathbb{R}^3 \\ z_i^{(l+1)} = \mathbf{W}_i^{(l+1)} \cdot \tilde{\mathbf{y}}^{(l)} + b_i^{(l+1)} = \sum_{j=1}^3 \mathbf{W}_{i,j}^{(l+1)} \cdot \tilde{y}_j^{(l)} + b_i^{(l+1)} \\ y_i^{(l+1)} = \sigma(z_i^{(l+1)}) \end{cases} \quad (5.5)$$

其中符号 \odot 为对应元素相乘, 另外, 伯努利(Bernoulli)分布是一种离散分布, 有两种可能的结果, 其中 1 表示成功, 0 表示失败, 注意符号 p 表示概率值, 即 $r_j^{(l)}$ ($j=1, 2, 3$) 是以概率 p 成功响应的。对比式(5.5)和式(5.4)可知, 从输入 $\mathbf{y}^{(l)}$ 到 $\tilde{\mathbf{y}}^{(l)}$, 导致第 l 层上部分节点不响应, 注意由于每个节点是独立同分布下的响应或不响应, 所以处理完后响应节点的个数为:

$$\tilde{n}_l = n_l \cdot p \quad (5.6)$$

其中 p 为相应概率, 即 Dropout 率; n_l 为隐层节点的个数, \tilde{n}_l 为随机概率处理完后的第 l 层上的响应节点的个数。应用中, 经过交叉验证, 隐含节点 Dropout 率等于 0.5 的时候效果最好, 主要原因是此时 Dropout 随机生成的网络结构最多。

另一种稀疏连接可以通过约减参数的方式来实现, 通常有两个思路: 一是直接将较小



的权值连接置为零(但有风险,因为随着层级的上升,较小的权值将会使得输入累积较为大的输出);二是通过矩阵分解来实现。下面简要介绍基于矩阵分解的参数约减,假设输入 $x \in \mathbb{R}^m$ 与输出 $y \in \mathbb{R}^n$ 之间的关系为:

$$\begin{cases} y = \sigma(W \cdot x + b) \\ W \in \mathbb{R}^{n \times m} \end{cases} \quad (5.7)$$

其中 W 为权值连接, $b \in \mathbb{R}$ 为偏置。进一步,对于权值连接 W 通过奇异值矩阵分解得到:

$$W = U \cdot \Sigma \cdot V^T \quad (5.8)$$

这里假设 $\text{rank}(W) = r$,则有 $U \in \mathbb{R}^{n \times r}$, $\Sigma \in \mathbb{R}^{r \times r}$ 和 $V \in \mathbb{R}^{m \times r}$;通过组合策略得到权值连接的表示为:

$$\begin{cases} W = W_2 \cdot W_1 \\ W_2 \in \mathbb{R}^{n \times r} \\ W_1 \in \mathbb{R}^{r \times m} \end{cases} \quad (5.9)$$

注意当 $W_1 = U \cdot \Sigma$ 时,则 $W_2 = V^T$;抑或当 $W_1 = U$ 时,则 $W_2 = \Sigma \cdot V^T$;模型相对应的式(5.7)则变为:

$$\begin{cases} z = W_1 \cdot x \\ y = \sigma(W_2 \cdot z + b) \end{cases} \quad (5.10)$$

需要注意的是:网络模型相对应式(5.7)中的权值连接 W 和式(5.10)中的权值连接(W_1 , W_2),其参数量由 $n \cdot m$ 变为 $r \cdot (n+m)$ 。注意该规则有效的前提是权值连接 W 是低秩矩阵,即 $\text{rank}(W) < \min(n, m)$ 。

备注:由于在实际大多数情形下,权值矩阵 W 是满秩的,因此通常取 Σ 的较大的 k 个奇异值并将其余奇异值置零,来实现对 W 的逼近。

5.2.4 稀疏分类器设计

常见的稀疏分类器设计是基于表示学习的,如稀疏表示分类器,其核心步骤包括:首先,字典构造:

$$D = [D_1, D_2, \dots, D_K] \quad (5.11)$$

其中 K 为类别个数, $D_k (k=1, 2, \dots, K)$ 为第 k 类样本或数据集构造(直接将样本堆叠形成)或学习(通过 K-SVD 算法)的字典;其次,对于样本 x 进行如下的表示学习:

$$\begin{aligned} & \min_{\alpha} \frac{1}{2} \cdot \|x - D \cdot \alpha\|_2^2 + \lambda \cdot \|\alpha\|_1 \\ &= \frac{1}{2} \cdot \|x - \sum_{k=1}^K D_k \cdot \alpha_k\|_2^2 + \lambda \cdot \|\alpha\|_1 \end{aligned} \quad (5.12)$$

注意这里的表示系数:

$$\alpha = [\alpha_1, \alpha_2, \dots, \alpha_K]^T \quad (5.13)$$

其中基于假设,若样本 x 属于第 k 类,则表示系数主要集中在 α_k ,而其他表示系数 $\alpha_j (j \neq k)$ 期



望为零,需要注意的是,这里的 α_k 是向量,而不是标量。最后类标的判定通过如下的公式实现:

$$\text{label}(\mathbf{x}) = \arg \min_{1 \leq k \leq K} \{ \| \mathbf{x} - \mathbf{D}_k \cdot \alpha_k \|^2_2 \} \quad (5.14)$$

另一种稀疏分类器的设计则是基于改进的 Softmax 分类器来实现的,其动机是改进 Softmax 输出处处不为零以期获得输出大多数为零,并记此为 Sparsemax 分类器,具体的数学物理描述如下。

1. Softmax 分类器

$$\begin{cases} \mathbf{y} = \text{Softmax}(\mathbf{x}, \theta) = [y_1, y_2, \dots, y_K]^T \\ y_k = P(\text{label}(\mathbf{x}) = k \mid \mathbf{x}, \theta_k) = \frac{1}{Z} \cdot e^{(\mathbf{x} \cdot \theta_k)} \end{cases} \quad (5.15)$$

其中 K 为类别个数,参数 $\theta = [\theta_1, \theta_2, \dots, \theta_K]$, Z 为归一化因子,即

$$Z = \sum_{j=1}^K e^{(\mathbf{x} \cdot \theta_j)} \quad (5.16)$$

待优化的参数为 θ 。

2. Sparsemax 分类器

$$\begin{cases} \mathbf{y} = \text{Sparsemax}(\mathbf{x}, \vartheta) = \arg \min_{\mathbf{p} \in \Delta^{k-1}} \| \mathbf{p} - (\mathbf{W} \cdot \mathbf{x} + \mathbf{b}) \|^2_2 \\ \vartheta = (\mathbf{W}, \mathbf{b}) \end{cases} \quad (5.17)$$

其中这里的“ Δ^{k-1} ”为单形,即满足:

$$\Delta^{k-1} = \left\{ \mathbf{p} \in \mathbb{R}^K : \sum_{i=1}^K p(i) = 1, p(i) \geq 0 \right\} \quad (5.18)$$

如何优化求解 p ? 对于 Softmax 分类器而言,已知参数 θ 和输入 \mathbf{x} ,则可以通过式(5.15)求出输出 \mathbf{y} 。而对于 Sparsemax 分类器,若知参数 ϑ 和输入 \mathbf{x} ,如何对式(5.17)进行优化呢?先简记符号:

$$\mathbf{z} = \mathbf{W} \cdot \mathbf{x} + \mathbf{b} \quad (5.19)$$

则优化目标变为:

$$\arg \min_{\mathbf{p} \in \Delta^{k-1}} \| \mathbf{p} - \mathbf{z} \|^2 \quad (5.20)$$

给定 \mathbf{z} ,关于 $\mathbf{p} \in \Delta^{k-1}$ 有如下形式的解:

$$p_k = [z_k - \tau(\mathbf{z})]_+ \quad (5.21)$$

这里的 p_k 为将输入 \mathbf{x} 分到第 k 类的概率,其中 $k=1, 2, \dots, K$ 和 $[t]_+ = \max(0, t)$ 。

另外 $\tau: \mathbb{R}^K \rightarrow \mathbb{R}$ 满足以下式子:

$$\sum_k [z_k - \tau(\mathbf{z})]_+ = 1 \quad (5.22)$$



进一步,记 z 中的坐标排序为:

$$z_1 \geq z_2 \geq \dots \geq z_K \quad (5.23)$$

并定义:

$$k(z) = \max \left\{ k \in [1, 2, \dots, K] : 1 + k \cdot z_k \geq \sum_{j \leq k} z_j \right\} \quad (5.24)$$

则 $\tau(\cdot)$ 可以被如下表示:

$$\tau(z) = \frac{\left(\sum_{j \leq k(z)} z_j \right) - 1}{k(z)} \quad (5.25)$$

5.2.5 深度学习中关于稀疏的技巧与策略

众所周知,深度学习是一类借鉴生物的多层神经网络处理模式所发展起来的智能处理技术,稀疏性可以大幅度削减深度神经网络中权值连接数量,因此被广泛采用。目前,对于深度卷积神经网络,便可以认为是深度前馈(全连接)神经网络的稀疏化;另外,稀疏深度网络模型的设计包括以下三条准则。

(1) 第一条准则,层级间模块化,逐层堆栈。

依据自编码网络进行逐层初始化,例如常用的有稀疏自编码器,其中关于隐结点输出的稀疏正则性约束包括KL散度和L1范数或伪范数,对应的稀疏深度网络模型称为(稀疏)深度堆栈网络;另外还有稀疏受限玻尔兹曼机所对应的稀疏深度置信网络和卷积稀疏编码所对应的(稀疏)反卷积神经网络等。

(2) 第二条准则,逐阶段模块化。

与层级间模块化不同,针对特定的任务,例如分类,利用生成式对抗网络(包括两个子网络,即生成模型和判别模型)在无监督学习方式下获取非合作状态下的零和博奕解,提取其判别网络中的特征学习部分(去掉后面的真伪二值分类器设计),结合分类器设计(如Softmax分类器),再利用监督学习的方式进行整个网络(由提取的特征学习部分和分类器设计部分组合而成)的精调。其中稀疏化可以内蕴在特征学习部分和分类器中。

(3) 第三条准则,多通路网络设计。

多分辨特性可以认为是输入在不同尺度或不同频带上的响应,相比较单尺度上对输入的(稠密性)表征,多分辨特性通过多通路或多通道来散化对输入的表征,使其在每一个尺度或频带上呈现稀疏性。另外,根据深度神经网络的设计准则,塔式、对称和多通路可以削弱“深度”对输入与输出之间的非线性刻画,即极深神经网络(例如深度残差网络、深度分形网络等)可由多通路、带有融合特性的深度神经网络来逼近。

在以上三条准则的基础上,常使用的稀疏性策略包括Dropout(目的是通过随机化权值连接实现参数的有效约减,间接提升训练数据量,以实现网络泛化性能的提升,有助于防止



过拟合现象),DropConnect,DropNeuron 等;另外,网络中激活函数的有效选择将有助于通过内蕴稀疏性来提升网络的泛化性能和计算开销,以及缓解反向传播时所带来的“梯度弥散”的现象,常用的激活函数见图 5.4。

目前,深度网络设计中最为常用的激活函数是修正线性单元 ReLU 及其改进版 RePLU,Maxout(本质上 ReLU 是 Maxout 的一种特例,操作见图 5.5)等。值得注意的是:深度学习的基础是数据,由于数据本身存在着差异性,对深度网络模型的影响也不一样;能否通过对数据的“分级处理”,如常见的基于无监督方式的数据聚类,通过划定与聚类中心的亲疏来实现样本的分级处理,如“优良中差”子数据集;进一步,对每级样本分别来学习深度神经网络,以期探索数据的差异性对深度卷积神经网络的影响。换言之,数据的分级处理体现着输入与输出之间映射的差异性,犹如大脑的多分辨特性,对信息结构完整或分辨率高的输入识别精度高,相反,对结构缺失或分辨率较低的输入识别精度低;若将这种多分辨特性与深度卷积神经网络相结合,形成多分辨深度卷积神经网络,实现对样本的筛选并改善基于差异性数据集学习到的深度卷积神经网络的性能。

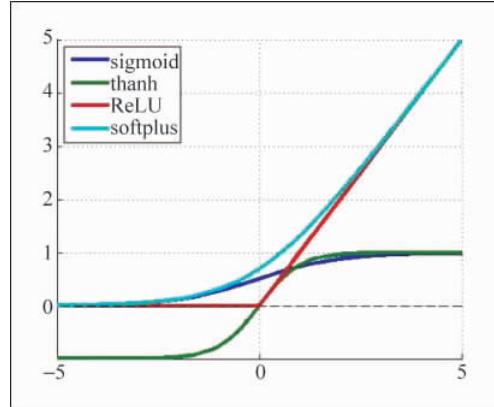


图 5.4 常用的激活函数

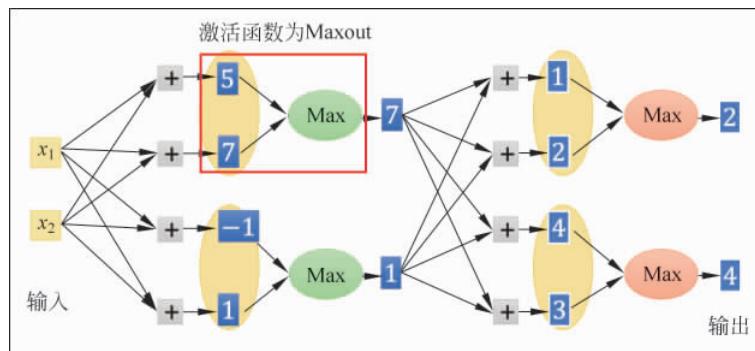


图 5.5 激活函数使用 Maxout

备注:关于激活函数 ReLU 中蕴含着的稀疏性请参考链接 <http://www.cnblogs.com/yymn/p/5616709.html>。

注意:Maxout 与 ReLU 的唯一区别是,前者是对若干个“隐隐层”单元的值执行最大化操作,而后者是对隐层上每一个单元执行与 0 相比较的最大化操作。



5.3 网络模型的性能分析

5.3.1 稀疏性对深度学习的影响

通常,原始数据中缠绕着高度密集的特征,稠密分布内蕴稀疏表达往往比局部少数点携载的特征成倍地有效,当然,在网络的设计过程中,过分地强调稀疏性处理,会减少模型的有效容量,即特征屏蔽太多,导致模型无法学到有效的特征;研究发现,理想的稀疏性比率保持在 70%~85%(量化的指标说明请参考备注中的参考文献),超过 85% 的深度网络模型的网络容量就成了问题,导致泛化性能锐减错误率极高。总之,模型稀疏化有诸多优点,但是过度的(显式)稀疏性通常也会导致模型的稳定性变差,从而泛化性能降低。

5.3.2 对比试验及结果分析

本节简要地给出稀疏深度堆栈网络的几组实验说明及结果分析。首先,网络的结构见图 5.6,网络优化分为预训练和精调两个阶段,超参数中关于激活函数和特征学习后分类器的设计作为对比点。

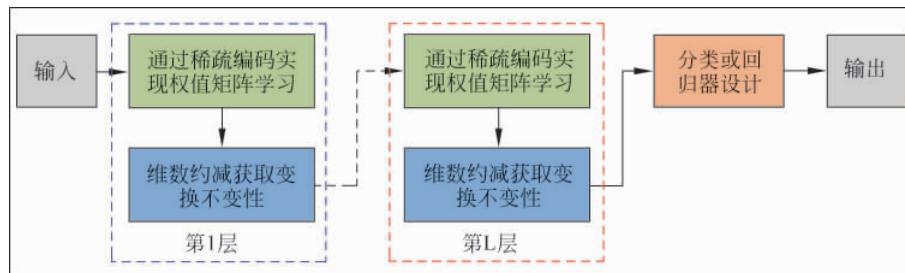


图 5.6 稀疏深度堆栈网络

1. 激活函数所蕴含着的稀疏性(分类器固定为 Softmax)(表 5.2)

表 5.2 不同激活函数下稀疏深度堆栈网络的测试误差

激活函数	Mnist	Cifar10	Mstar	ImageNet
Sigmoid	3.28%	48.12%	12.18%	58.65%
Tanh	3.16%	51.04%	11.97%	56.92%
Softplus	2.75%	43.76%	10.34%	54.21%
ReLU	2.93%	46.23%	10.93%	55.63%
Maxout	2.64%	44.74%	9.73%	52.17%



注意网络的隐层个数设计为3层,预训练阶段,权值初始化的方式采用稀疏编码策略,即将学到的字典进行转置得到相应的滤波器,通过最大池化的方式进行维数约减。

2. 稀疏分类器设计 VS 分类器设计(激活函数使用 ReLU)(表 5.3)

表 5.3 不同分类器下稀疏深度堆栈网络的测试误差

分 类 器	Mnist	Cifar10	Mstar	ImageNet
SVM	2.68%	43.47%	9.10%	54.96%
SMM	2.57%	42.71%	8.92%	54.05%
Softmax	2.93%	46.23%	10.93%	55.63%
Sparsemax	2.52%	44.38%	7.52%	55.15%

注意这里的 SMM 为支撑矩阵机,详尽参考第 1 章机器学习小结中支撑向量机第二种改进的方案。

结果分析,从实验中可以看出 ReLU 激活函数隐含对特征“非负稀疏性”的要求;而(最大)池化操作隐含对特征“强稀疏性”(特征选择)的要求;以及参数层偏置隐含对特征“稀疏度”的调节。另外,深度神经网络是关于自动学习要建模的数据的潜在(隐含)分布的多层次(复杂)表达的算法。换言之,深度学习算法自动地提取分类需要的高层抽象特征,而适当地引入显式或隐式稀疏规则将有助于克服过拟合现象的发生,同时提升网络的泛化性能。

参考文献

- [5.1] Hu X, Zhang J, Li J, et al. Sparsity-regularized HMAX for visual recognition[J]. Plos One, 2014, 9(1): e81813.
- [5.2] Wright J, Yang A Y, Ganesh A, et al. Robust Face Recognition via Sparse Representation[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2009, 31(2): 210-227.
- [5.3] Martins A F T, Astudillo R F. From Softmax to Sparsemax: A Sparse Model of Attention and Multi-Label Classification[J]. 2016.
- [5.4] Glorot X, Bordes A, Bengio Y. Deep Sparse Rectifier Neural Networks[J]. Journal of Machine Learning Research, 2011, 15.
- [5.5] Toth L. Phone recognition with deep sparse rectifier neural networks[C]. IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013: 6985-6989.
- [5.6] Pironkov G, Dupont S, Dutoit T. Investigating sparse deep neural networks for speech recognition [C]. Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE, 2015: 124-129.
- [5.7] Scardapane S, Comminiello D, Hussain A, et al. Group sparse regularization for deep neural networks[J]. Neurocomputing, 2016.
- [5.8] Zheng H, Chen M, Liu W, et al. Improving deep neural networks by using sparse dropout strategy [C]. IEEE China Summit & International Conference on Signal and Information Processing. IEEE,



2014: 21-26.

- [5.9] Han S, Pool J, Narang S, et al. DSD: Regularizing Deep Neural Networks with Dense-Sparse-Dense Training Flow[J]. 2016.
- [5.10] Graham B. Spatially-sparse convolutional neural networks[J]. Computer Science, 2014, 34(6): 864-867.
- [5.11] Liu B, Wang M, Foroosh H, et al. Sparse Convolutional Neural Networks[C]. Computer Vision and Pattern Recognition. IEEE, 2015: 806-814.
- [5.12] Gripon V, Berrou C. Sparse neural networks with large learning diversity[J]. IEEE Transactions on Neural Networks, 2011, 22(7): 1087-1096.
- [5.13] Balavoine A, Romberg J, Rozell C J. Convergence and Rate Analysis of Neural Networks for Sparse Approximation[J]. IEEE Transactions on Neural Networks & Learning Systems, 2011, 23(9): 1377-1389.
- [5.14] Pernice V, Rotter S. Reconstruction of sparse connectivity in neural networks from spike train covariances [J]. Journal of Statistical Mechanics Theory & Experiment, 2013, 2013 (2013): P03008.