

DOI: 10.3969/j.issn.1674-1579.2020.03.004

多斜率自适应卷积神经网络激活函数*

刘 海^{1,2}, 刘波³, 胡 瑜^{1,2*}

摘 要: 针对卷积神经网络中激活函数无法有效为处于不同激活程度的像素点提供特定梯度响应的问题,设计了一种由多个分段线性函数组成的自适应激活函数.首先依据像素激活值的取值范围,自适应地生成多个独立的激活域,各个激活域的并集包含激活图中全体像素点的激活值;随后在每个激活域中学习一个特定的线性函数,为处于该激活域中的像素点提供特定的梯度响应;最后以NIN网络和ResNet18网络为例,在CIFAR-10和CIFAR-100数据集上,验证所提激活函数的性能.实验结果表明,与现有激活函数相比,本文提出的激活函数,由于能够更好地为处于不同激活程度的像素点提供适当的梯度响应,使分类准确率在NIN网络上分别达到87.96%和69.01%,在ResNet18网络上分别达到88.56%和73.54%.

关键词: 激活函数,卷积神经网络,图像分类

中图分类号: TP18

文献标志码: A

文章编号: 1674-1579(2020)03-0028-08

Multi-Slope Adaptive Convolutional Neural Network Activation Function

LIU Hai^{1,2}, LIU Bo³, HU Yu^{1,2}

Abstract: Aiming at the problem that the activation function in Convolutional Neural Networks can't effectively provide a specific gradient response for pixels at different activation levels, an adaptive activation function is designed, which is composed of multiple piecewise linear functions. Firstly, according to the ranges of pixel activation value, multiple independent activation domains are adaptively generated, and the union of each activation domain contains the activation values of all pixels in the activation map; then a specific linear function is learned in each activation domain in order to provide a specific gradient response for the pixels in the activation domain; finally taking the NIN and ResNet18 as examples, on the CIFAR-10 and CIFAR-100 dataset, the performance of the proposed activation function is verified. Experimental results show that, compared with the existing activation function, the activation function proposed in this paper can provide a suitable gradient response for pixels with different activation levels, so that the classification accuracies reach 87.96% and 69.01% on the NIN respectively. The accuracies reach 88.56% and 73.54% on the ResNet18, respectively.

Keywords: activation function, convolutional neural network, image classification

收稿日期:2020-04-28;录用日期:2020-06-10.

Manuscript received April. 28, 2020; accepted June 10, 2020.

国家重点研发计划科技创新2030—“新一代人工智能”重大资助项目(2018AAA0102701).

National Key R&D Program of China (2018AAA0102701).

1. 中国科学院计算技术研究所智能计算机研究中心, 北京 100190;
2. 中国科学院大学, 北京 100049; 3. 北京控制工程研究所, 北京 100190.

1. Research Center for Intelligent Computing Systems, Institute of Computing Technology, CAS, Beijing 100090, China; 2. University of Chinese Academy of Sciences, Beijing 100049, China; 3. Beijing Institute of Control Engineering, Beijing 100090, China.

0 引 言

近年来卷积神经网络技术在诸多领域取得了突破性成果,例如计算机视觉和医学图像处理领域,卷积神经网络的物体分类和物体检测能力,甚至超越了人类水平.卷积神经网络主要由卷积层、池化层和激活层三部分构成,其中激活函数处于激活层中,是实现网络非线性表达能力的关键组件.激活函数作用于卷积层产生的特征图,得到相应的激活图,激活图中包含大量的像素点,不同的像素点处于不同的

激活状态.因此设计适当的激活函数,为处于不同激活状态下的像素点提供适当的梯度响应,是提高卷积神经网络性能的重要方法.

按照是否能够通过学习训练提高激活函数性能进行划分,现有激活函数可分为:自适应激活函数和非自适应激活函数.自适应激活函数包含一定数量的可学习参数,能够从输入数据中学习出最恰当的参数,而非自适应激活函数只对卷积层产生的特征图进行激活变换,所有变换过程均是固定不变的.

激活函数的发展起始于非自适应激活函数.最初的 Sigmoid 激活函数^[1]被广泛应用于各种卷积神经网络中.但随着网络深度的增加,Sigmoid 激活函数容易造成梯度消失和梯度爆炸. ReLU 激活函数^[2]虽然能够缓解 Sigmoid 激活函数面临的梯度消失和梯度爆炸问题,提升了网络的收敛速度,然而 ReLU 激活函数在负值域中梯度为零,导致在网络中存在一定数量的失效神经单元,为此 Leaky ReLU (LReLU)^[3]、RReLU^[4]、ELU^[5]和 Swish^[6]等激活函数被相继提出,以增加负值域中的梯度响应.不同于上述的非自适应激活函数,文献[7]提出了梯度截断策略,一定程度上为不同激活程度的像素点提供了具有差异性的梯度响应.

相比于非自适应激活函数,自适应激活函数由于能够从输入数据中学习部分可变参数,因而对数据具有更好的适应性.具有代表性的自适应激活函数包括 PReLU^[8]和 GReLU^[9].其中 GReLU 为不同激活程度的像素点采取了差异性的梯度响应.

本文方法也是直接针对不同激活程度像素点的梯度响应这一问题,提出了多斜率自适应激活函数 (Multi-Slope Adaptive Activation Function, MReLU).与现有激活函数相比,本文所提 MReLU 具有以下两大优势.第一:MReLU 通过增加可学习参数,为处于不同激活程度的像素点提供自适应梯度响应函数.第二:多斜率自适应激活函数包含有限个跳跃间断点,为激活函数提供了更高的灵活性,使该激活函数能更好为处于不同激活程度的像素点提供恰当的梯度响应.

本文的结构如下:第一部分详细讨论激活图中像素点激活程度的分布情况,第二部分详细解释多斜率自适应激活函数并与已有激活函数进行分析对比,第三部分通过实验对比本文提出的激活函数与现有激活函数的在图像分类任务上的性能,并讨论不同配置对于该激活函数的影响,最后第四部分给

出本文主要结论和进一步的研究方向.

1 激活分布

卷积神经网络主要由卷积层、池化层和激活层三部分构成,其中激活层是卷积神经网络实现非线性特征提取功能的主要组件.通过激活函数,激活层对卷积层的输出特征图进行非线性变换,得到对应的激活图.以 Network in Network (NIN)^[10]网络位于第三层的卷积层所采用的 ReLU 激活函数为例,当 NIN 网络对 CIFAR-10^[11]数据集进行分类时,第三层卷积层所输出的特征图经由 ReLU 激活函数后得到对应的激活图,如图 1 所示.图 1 中的左图是第三层卷积层所输出的特征图,右图是通过 ReLU 激活函数得到的激活图.各个小图右侧的图例标识出了图中颜色和数值的对应关系.

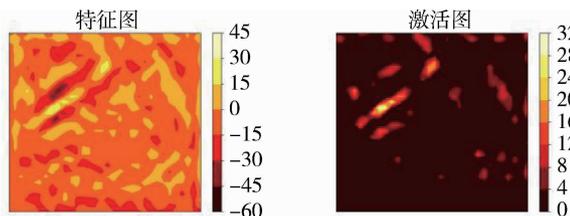


图 1 NIN 网络第三个卷积层输出的特征图及 ReLU 函数作用于该特征图所得的激活图

Fig. 1 The feature map output by the third convolutional layer of the NIN network and the activation map obtained by the ReLU function acting on the feature map

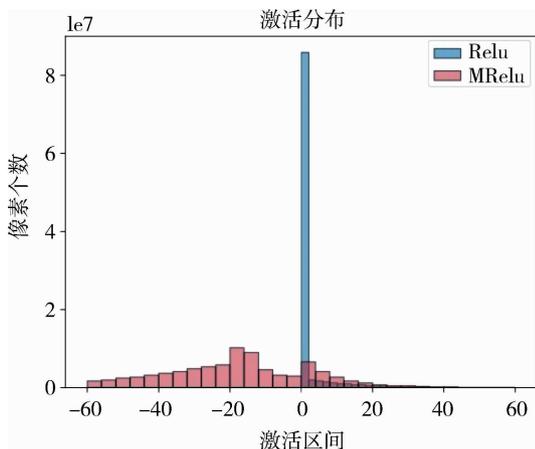


图 2 NIN 网络中第三个卷积层特征图分别利用 ReLU 和 MReLU 函数得到对应的激活图,激活图中激活值的分布统计

Fig. 2 The third convolutional layer feature map in the NIN network uses ReLU and MReLU functions to obtain corresponding activation maps. The distribution statistics of activation values in activation maps.

由图 1 可见,在左侧特征图中,像素的取值范围为 $[-60, 45]$,而在右侧激活图中,像素的取值范围为 $[0, 32]$,这意味着不同像素点处于强弱不同的激活状态.然而从 ReLU 函数的表达式可见(详见表 1 第 2 行第 2 列),ReLU 函数对所有正激活状态的像素点提供了相同的梯度响应.这一特性使得网络无法对处于不同激活程度的像素点提供具有差异的梯度响应,即不能够让处于弱激活程度的像素点具有较多机会获得更高层次的激活,如图 2 所示.

图 2 中对比了将 ReLU 和本文提出的 MReLU 激活函数用于 NIN 网络时,在 CIFAR-10 数据集上, NIN 网络第三层激活图中像素点的激活值分布情况.图 2 中的数据来自于对 1024 张输入图片(每张输入图片在该激活层中有 96 个通道对应 96 张激活图)像素点激活值分布的统计.图中横坐标表示对应的激活值区间,纵坐标表示对应区间上的像素点个数.其中蓝色柱状图表示的是 ReLU 激活函数的激活分布,红色柱状图表示的是 MReLU 激活函数的激活分布.

图 3 展示了 ReLU 族(ReLU, PReLU, LReLU, RReLU), ELU, SELU^[12], Swish 激活函数及其导函数,图中蓝色线条表示激活函数,橙色线条表示该激活函数的导函数.由图可见,上述激活函数的导函数均接近常函数,说明上述激活函数无法有效地为处于不同激活程度的像素点提供有效的差异梯度响应.虽然有研究结合了神经网络自动搜索技术^[13]为网络选择适当的激活函数,然而其函数空间为已有激活函数的集合,无法得出能够为不同激活程度像素点提供差异梯度响应的新型激活函数.

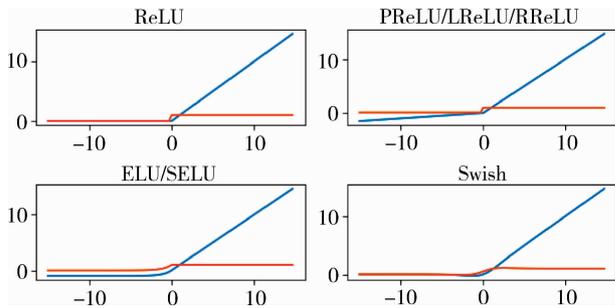


图 3 常见激活函数及其导函数图像

Fig. 3 Various activation functions and their derivatives

本文针对这一问题提出了多斜率自适应激活函数 MReLU,该激活函数能够从数据中学习出适当的梯度响应,为不同激活程度的像素点提供更加适合的梯度响应.

2 多斜率自适应激活函数

网络激活图中不同像素点处于不同的激活状态,若对不同的像素点采用同等的梯度响应,就意味着这些像素点得到同等的增强程度(激活值位于正区间的像素点)或削弱程度(激活值位于负区间的像素点).然而像素点的增强程度和削弱程度,如果被人为地统一固定,则很难有效提升网络性能,因此我们设计了多斜率自适应激活函数.

多斜率自适应激活函数 MReLU 的定义如下:

$$f(x) = \begin{cases} k_1 x & \text{if } x \in A_1 \\ \dots \\ f_i x & \text{if } x \in A_i \\ \dots \\ k_n x & \text{if } x \in A_n \end{cases}$$

其中, k_1, \dots, k_n 为可学习的斜率参数, A_1, \dots, A_n 为 n 个互不相交的实数区间,其并集为整个实数域.任意像素点输入到该激活函数中,依据激活值的大小,唯一地映射到某一区间中,并获得对应的梯度响应. MReLU 在各个区间中使用线性函数作为梯度响应函数,简化了激活函数的计算复杂度,同时简化了激活函数的梯度计算,其导函数定义如下:

$$\frac{\partial f(x)}{\partial x} = \begin{cases} k_1 & \text{if } x \in A_1 \\ \dots \\ k_i & \text{if } x \in A_i \\ \dots \\ k_n & \text{if } x \in A_n \end{cases}$$

其中, k_1, \dots, k_n 为可学习的斜率参数, A_1, \dots, A_n 为 n 个互不相交的实数区间,其并集为整个实数域.相比与现有激活函数, MReLU 激活函数虽然增加了少量的可学习参数,但其反向传播过程简单,依然是一个轻量的激活函数.

图 4 给出了 MReLU 函数的一个具体示例.图中的函数示例指定了 $((-\infty, -7), (-7, -3), (-3, -1), (-1, 0), (0, 1), (1, 3), (3, 7), (7, +\infty))$ 八个不同的区间,每个区间上对应的斜率为 $(0.1, 0.25, 0.5, 1.0, 1.0, 0.5, 0.25, 0.1)$.图中的虚线表示了 MReLU 函数中包含的跳跃间断点.

图 4 的示例展示了 MReLU 函数的两大特性——第一:该函数包含有限的跳跃间断点,为每个区间上的函数选择提供了更大的灵活性,同时也能自适应地为不同区间上的像素点提供对应的梯度响

应. 第二:该函数在不同区间上具有非单调性,该特性使得像素点在激活前后的有序性将会被一定程度的破坏,改变了常见激活函数单调递增的特性,为低激活像素点提供了更多的机会获得更高层次的激活,同时对高激活像素点具有一定的抑制作用,这一特性有助于网络将更多的注意力集中到需要被提升的像素点之上.

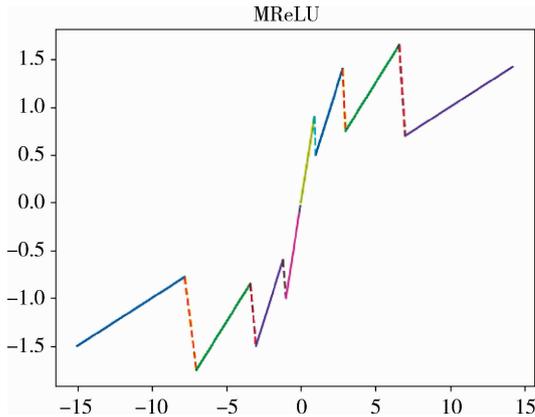


图 4 MReLU 函数示例,分段区间为 $((-\infty, -7), (-7, -3), (-3, -1), (-1, 0), (0, 1), (1, 3), (3, 7), (7, +\infty))$, 各区间斜率为 $(0.1, 0.25, 0.5, 1.0, 1.0, 0.5, 0.25, 0.1)$
 Fig. 4 Example of MReLU function, and the segment interval is specified as $((-\infty, -7), (-7, -3), (-3, -1), (-1, 0), (0, 1), (1, 3), (3, 7), (7, +\infty))$, and the corresponding slope of each interval is $(0.1, 0.25, 0.5, 1.0, 1.0, 0.5, 0.25, 0.1)$

依据激活函数中是否包含可学习参数,可将激活函数分为两大类型:自适应激活函数和非自适应激活函数. 本文所提的 MReLU 属于自适应激活函数. 表 1 对比了非自适应激活函数 ReLU, Leaky ReLU/RRReLU, ELU, SELU, Swish 和 BReLU, 以及自适应激活函数 PReLU、GReLU 和本文所提 MReLU 的差异和特性. 其中 ReLU 只有正值域中的梯度,因此容易导致网络中存在失效神经单元. Leaky ReLU 和 RRReLU 通过引入一个非学习数值,在 ReLU 的基础上增加了一个负值域上的线性梯度响应. ELU/SELU 则通过引入一个非线性函数来提供负值域中的梯度响应. Swish 改进了 Sigmoid 函数,缓解了右侧区间上的梯度饱和. 非自适应激活函数通常灵活性较低,无法充分利用数据中的信息,也无法为处于不同激活程度的像素点提供特定的梯度响应.

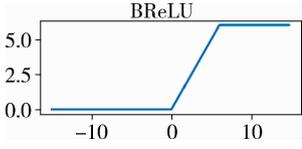
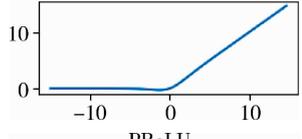
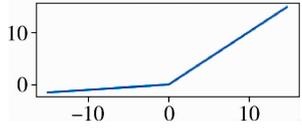
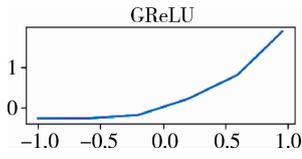
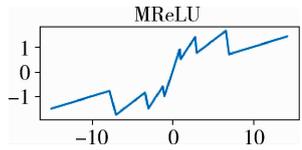
不同于其他的激活函数, BReLU 和 GReLU 这两种激活函数均一定程度上考虑了为不同激活程度的像素点提供不同的梯度响应. 相比于 BReLU 和 GReLU, 本文提出激活函数具有两大优势——第一:引入跳跃间断点,让每个区间中激活函数的选择具有更大的灵活性;第二:改变像素点激活前后的有序性,提高了低激活像素点获得更高激活的机会.

表 1 MReLU 与常见激活函数的比较

Tab. 1 Comparisons between MReLU and common activation functions

函数名称	表达式	图像	类型	特性
ReLU	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases}$		非自适应	负值域中梯度为零
LReLU/RRReLU	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ ax & \text{if } x < 0 \end{cases}$		非自适应	负值域中增加了一个额外的线性梯度响应
ELU	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ b(e^x - 1) & \text{if } x < 0 \end{cases}$		非自适应	负值域中使用一个非线性函数用于梯度响应
SELU	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ r(b(e^x - 1)) & \text{if } x < 0 \end{cases}$		非自适应	ELU 的改进版本,增加了一个固定参数提升了 ELU 的性能

续表

函数名称	表达式	图像	类型	特性
BReLU	$f(x) = \begin{cases} n & \text{if } x > n \\ x & \text{if } 0 \leq x \leq n \\ 0 & \text{if } x < 0 \end{cases}$		非自适应	通过引入梯度截断,对高激活程度像素点进行梯度截断
Swish	$f(x) = x \frac{1}{1 + e^{-x}}$		非自适应	Sigmoid 改进型,收敛速度快
PReLU	$f(x) = \begin{cases} x & \text{if } x \geq 0 \\ k & \text{if } x < 0 \end{cases}$		自适应	ReLU 的改进版本,在负值域中引入了一个可学习参数
GReLU	$f(x) = \begin{cases} l_1 + \sum_{i=1}^{n-1} k_i(l_{i+1} - l_1) + k_n(x - l_n), & \text{if } x \in [l_n, +\infty) \\ \vdots \\ l_2 + k_1(x - l_1), & \text{if } x \in [l_1, l_2] \\ x, & \text{if } x \in [l_{-1}, l_1] \\ l_{-1} + k_{-1}(x - l_{-1}), & \text{if } x \in [l_{-2}, l_{-1}] \\ \vdots \\ l_{-1} + \sum_{i=1}^{n-1} k_{-1}(l_{-(i+1)} - l_{-1})(k_{-n}x - l_{-n}), & \text{if } x \in (-\infty, l_{-n}) \end{cases}$		自适应	通过引入多斜率线性区间为不同激活程度的像素点提供不同梯度响应
MReLU	$f(x) = \begin{cases} k_1 x & \text{if } x \in A_1 \\ \dots \\ k_i x & \text{if } x \in A_i \\ \dots \\ k_n x & \text{if } x \in A_n \end{cases}$		自适应	通过引入多个独立区间,为不同激活程度的像素点提供不同梯度;跳跃间断点为函数提供更高的灵活度

3 实验与分析

3.1 实验配置

本文将在图片分类任务上,验证 MReLU 的性能,并与现有的激活函数进行性能对比. 图片分类任务是计算机视觉领域的基础任务. CIFAR-10 和 CIFAR-100^[11]是被广泛使用的图片分类数据集,因此本文也在 CIFAR-10 和 CIFAR-100 数据集上开展实验. 为体现 MReLU 激活函数的普适性,实验中采用 NIN 和 ResNet 两种常见的网络结构进行验证.

3.1.1 CIFAR-10 和 CIFAR-100 数据集

CIFAR-10 和 CIFAR-100^[11]数据集是两个常用的分类数据集. 其中 CIFAR-10 数据集包含了 10 个独立类别的物品用于分类任务,每个类别包含 5000 张训练图片,1000 张测试图片,每张图片的大小均是 32x32 像素. CIFAR-100 数据集包含了 20 个大

类,100 个小类,每个小类包含 500 张训练图片,100 张测试图片,每张图片的大小均是 32x32 像素. 在实验过程中,为了避免引入其他因素的影响,我们直接使用原始图片进行训练,没有使用任何数据增强和数据预处理技术.

3.1.2 NIN 和 ResNet 网络结构

NIN 网络是一个只包含九个卷积层的浅层网络,表 2 给出了 NIN 网络的结构参数. 表中第一列表示输入特征图的大小;第二列为对应的操作,第一个值表示卷积核或者 pooling 核的大小,第二个值表示当前层的输出通道数,第三个值表示步长.

ResNet^[14]是在分类网络中被使用最多的网络结构,目前在 CIFAR-10 和 CIFAR-100 数据排行榜上的最优分类方法采用了 ResNet152 网络结构,由于该排行榜上的结果均使用了迁移学习进行增强,为了避免引入其他因素对实验的影响,同时不失一般性,本文使用了 ResNet18 作为基础分类网络,用于对比不同激活函数的性能.

3.2 实验结果

将 MReLU 与当前常见的九种激活函数:ReLU、Leaky ReLU、PReLU、RReLU、ELU、SELU、Swish、BReLU 和 GReLU 进行了实验对比.表 3 所示为各激活函数在 NIN 网络上的实验结果.表 3 中包含两行分别表示各个激活函数在 CIFAR-10 和 CIFAR-100 数据集上的准确率,其中最优结果用黑色粗体标记.由表 3 可见,在 NIN 网络结构上,MReLU 激活函数在 CIFAR-10 和 CIFAR-100 两个数据集上分别均取得了最优和次优的结果,其准确率分别达到了 87.96% 和 69.01%.

表 2 NIN 网络结构

Tab.2 The NIN network structure

Input Size	NIN
32x32	5x5,192
32x32	1x1,160
32x32	1x1,96
32x32	3x3, max pooling, / 2
16x16	dropout,0.5
16x16	5x5,192
16x16	1x1,192
16x16	1x1,192
16x16	3x3, avg pooling, / 2
8x8	dropout,0.5
8x8	3x3,192
8x8	1x1,192
8x8	1x1,10
8x8	8x8, avg pooling / 1
10 or 100	softmax

表 4 所示为各激活函数在 ResNet18 网络上的实验结果.由表 4 可见,在 ResNet18 网络上,MReLU 激活函数在 CIFAR-10 和 CIFAR-100 两个数据集上均取得了最优的结果,其准确率分别达到了 88.56% 和 73.54%.

通过对比表 3 和表 4 中的数据,得到以下三个观察——第一:CIFAR-100 数据集上 MReLU 在 NIN 网络中的准确率为 69.01%,低于 GReLU 的 70.41%,而在其他实验配置(NIN 网络中的 CIFAR-10,ResNet18 网络中的 CIFAR-10 和 CIFAR-100)中 MReLU 的分类准确率均高于 GReLU.这一现象说明,MReLU 比 GReLU 具有明显的性能优势,这一优势主要来自于 MReLU 更加灵活的函数可选择性.第二:ResNet18 网络结构中 MReLU 在 CIFAR-10 数据集上准确率为 88.56% 比次优的 Swish 高出了 0.08%,而在 CIFAR-100 数据集上 MReLU 的准确率为 73.54% 仅比次优的 SELU 高出了 0.02%.这

一现象说明在同等网络结构下,随着数据集复杂度的提升,将减弱激活函数差异对网络性能的影响.第三:CIFAR-10 数据集上 MReLU 在 NIN 网络中的准确率为 87.96% 比次优的 RReLU 高出了 0.29%,而在 ResNet18 网络中 MReLU 的准确率为 88.56% 仅比次优的 Swish 高出了 0.08%.这一现象说明在同一数据集上,随着网络结构复杂度的提升,激活函数的差异对网络性能的影响减弱.

3.3 性能分析

MReLU 激活函数中主要包含两个部分参数:激活区间的划分和各个区间上的梯度.其中,各区间上的梯度是由网络从数据中逐步学习,而区间的划分则相对复杂.

从图 2 中可以看出,MReLU 激活函数得到的激活图中像素激活值的分布关于零点具有一定程度的对称性,因此本文设计了三种关于零点对称的区间划分方案.同时为了更好地对比 MReLU 与 ReLU 族(ReLU, RReLU, PReLU, LReLU)函数的性能差异,设计了三种仅在负值域中进行分段梯度响应的区间划分方案.

这六种不同的区间划分方案依次为——1: $((-\infty, -15), (-15, -7), (-7, -3), (-3, 0), (0, 3), (3, 7), (7, 15), (15, +\infty))$, 2: $((-\infty, -7), (-7, -3), (-3, -1), (-1, 0), (0, 1), (1, 3), (3, 7), (7, +\infty))$, 3: $((-\infty, -3), (-3, -2), (-2, -1), (-1, 0), (0, 1), (1, 2), (2, 3), (3, +\infty))$, 4: $((-\infty, -15), (-15, -7), (-7, -3), (-3, 0), (0, +\infty))$, 5: $((-\infty, -7), (-7, -3), (-3, -1), (-1, 0), (0, +\infty))$, 6: $((-\infty, -3), (-3, -2), (-2, -1), (-1, 0), (0, +\infty))$.

图 5 展示了 ResNet18 网络在 CIFAR-10 数据集上的训练进程.图中的横坐标表示训练的轮数,纵坐标表示 CIFAR-10 验证集上的准确率.从图 5 可见,在不同的区间划分方案下,各 MReLU 激活函数的收敛速度和最终的性能有较大差异,其中,第 2 个区间划分方案取得了最优性能.

除了区间的划分,梯度的初始值也对激活函数的性能有着一定的影响.为了讨论不同的梯度初始值对激活函数性能的影响,本文指定了四个不同的梯度初始值:0.5,1.0,1.5,2.0.图 6 展示 ResNet18 网络在 CIFAR-10 数据集上的训练进程.从图 6 中可见,当梯度初始值为 2.0 时,ResNet18 网络取得了最优性能,此结果与已知的各种基于 ReLU 的激活函

数均不同. ReLU 的变种函数,在负值域上通常初始化为一个远小于 1.0 的初始值. 这一点说明为不同

激活程度的像素点提供不同的梯度响应,其效果明显不同于对所有像素点提供同等的梯度响应.

表 3 在 NIN 网络上的分类准确率

Tab. 3 Classification accuracies with the NIN network

NIN (%)	ReLU	LReLU	PRReLU	RReLU	ELU	SELU	Swish	BReLU	GRReLU	MReLU
CIFAR-10	86.27	86.34	86.59	87.67	86.76	85.95	85.71	83.87	86.65	87.96
CIFAR-100	64.95	65.50	66.70	68.62	66.07	67.34	66.25	59.37	70.41	69.01

表 4 在 ResNet18 网络上的分类准确率

Tab. 4 Classification accuracies with the ResNet18 network

ResNet18 (%)	ReLU	LReLU	PRReLU	RReLU	ELU	SELU	Swish	BReLU	GRReLU	MReLU
CIFAR-10	88.19	87.86	88.32	88.22	87.35	88.03	88.48	88.12	88.42	88.56
CIFAR-100	72.22	72.71	73.11	72.63	72.75	73.52	72.01	72.31	72.58	73.54

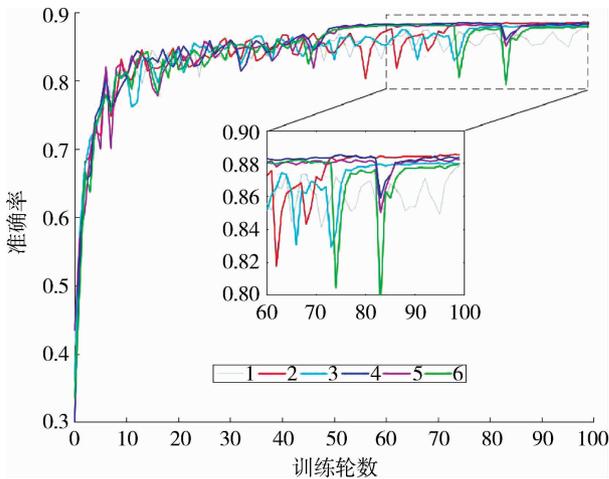


图 5 ResNet18 网络中 MReLU 各区间划分方案性能对比 (CIFAR-10 数据集)

Fig. 5 Performance comparison of different intervals of MReLU on the CIFAR-10 dataset in the ResNet18 network

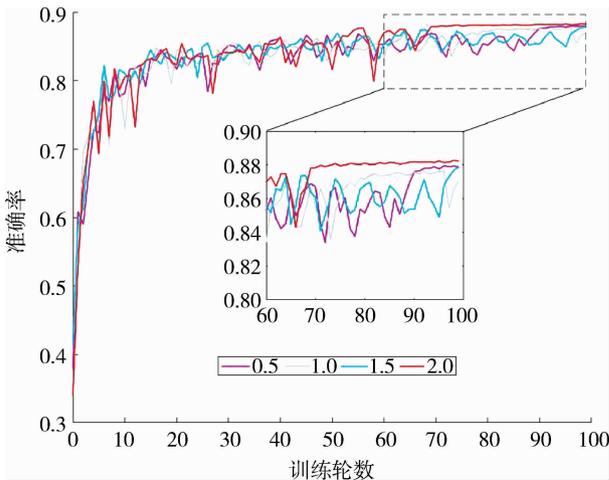


图 6 ResNet18 网络中 MReLU 不同初始值的性能对比 (CIFAR-10 数据集)

Fig. 6 Performance comparison of different initial values of MReLU on the CIFAR-10 dataset in the ResNet18 network

4 结 论

本文针对神经网络中不同像素点存在激活程度不同这一现象,提出了多斜率自适应激活函数. 实验结果表明针对处于不同激活程度的像素点采用不同程度的梯度响应,有助于进一步提升网络的性能. 同时包含有限个跳跃间断点的函数形式为激活函数带来了更大的灵活性和学习空间. 后续我们将在更多的网络结构、数据集和目标检测、语义分割等更多的视觉任务上,验证 MReLU 激活函数的有效性.

参 考 文 献

[1] HAN J, MORAGA C. The influence of the sigmoid function parameters on the speed of backpropagation learning [C] // International Workshop on Artificial Neural Networks. Springer, 1995.

[2] VINOD N, GEOFFREY E H. Rectified linear units improve restricted boltzmann machines vinod nair [C] // Proceedings of the 27th International Conference on Machine Learning. Haifa, 2010.

[3] MAAS A L, HANNUN A Y, NG A Y. Rectifier nonlinearities improve neural network acoustic models. [C] International Conference on Machine Learning, Atlanta, GA, USA 2013.

[4] XU B, WANG N Y, CHEN T Q, et al. Empirical evaluation of rectified activations in convolutional network [EB/OL]. [2015-05-05]. <https://arxiv.org/abs/1505.00853>.

[5] CLEVERT D, UNTERTHINER T, HOCHREITER S. Fast and accurate deep network learning by exponential linear units (ELUs) [C] // International Conference on

- Learning Representations. San Juan, 2016.
- [6] RAMACHANDRAN P, ZOPH B, QUOC V L. Searching for Activation Functions [EB/OL]. [2017-10-16]. <https://arxiv.org/abs/1710.05941>.
- [7] KRIZHEVSKY A. Convolutional deep belief networks on CIFAR-10 [EB/OL]. [2010]. <http://www.cs.utoronto.ca/~kriz/conv-cifar10-aug2010.pdf>.
- [8] HE K M, ZHANG X Y, REN S Q, et al. Delving deep into rectifiers: surpassing human-level performance on ImageNet classification [EB/OL]. [2015-02-06]. <https://arxiv.org/abs/1502.01852>.
- [9] CHEN Z, HO P H. Deep global-connected net with the generalized multi-piecewise ReLU activation in deep learning [EB/OL]. [2018-06-19]. <https://arxiv.org/abs/1807.03116>.
- [10] LIN M, CHEN Q, YAN S. Network in network [EB/OL]. [2013-12-16]. <https://arxiv.org/abs/1312.4400>.
- [11] KRIZHEVSKY A, HINTON G. Learning multiple layers of features from tiny images. Computer Science Department [J]. University of Toronto Technician Rep, 2009, 1(4):7.
- [12] KLAMBAUER G, UNTERTHINER T, MAYR A, et al. Self-normalizing neural networks [EB/OL]. [2017-09-07]. <http://arxiv.org/abs/1706.02515>.
- [13] ANDREW H, MARK S, GRACE C, et al. Searching for MobileNetV3 [C] // IEEE International Conference on Computer Vision. Seoul, 2019.
- [14] HE K M, ZHANG X Y, REN S Q, et al. Deep residual learning for image recognition [C] // In Proceeding of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. New York:IEEE,2016.

作者简介:刘海(1994—),男,硕士研究生,研究方向为计算机视觉、深度学习;刘波(1977—),男,研究员,研究方向为容错计算技术与星载计算机体系结构;胡瑜(1975—),女,研究员,研究方向为自动驾驶感知与决策、深度学习算法加速。

(上接第10页)

- [16] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient Estimation of Word Representations in Vector Space [C] // IEEE International Conference on Computer Vision. New York:IEEE, 2015.
- [17] ALON U, ZILBERSTEIN M, OMER L, et al. Code2vec: learning distributed representations of code [J]. Proceedings of the ACM on Programming Languages, 2019,1:29.
- [18] KIM H, JIANG Y, KANNAN S, et al. Deepcode: feedback codes via deep learning [C] // Advances in Neural Information Processing Systems 31. Montréal, 2018.
- [19] LILJESON M. Software defect prediction using machine learning on test and source code metrics [J]. International Journal of Science and Research, 2014,3(12): 1394-1397.
- [20] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets [J]. Neural Computation, 2006,18(7): 1527-1554.
- [21] GUPTA A, SUNDARESAN N. Intelligent code reviews using deep learning [C] // The 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. London: ACM, 2018.
- [22] VASWANI A, SHAZEER N, PARMAR N, et al. Attention Is All You Need [C] // The 31th Advances in Neural Information Processing Systems, Long Beach, 2017.
- [23] SCARSELLI F, GORI M, TSOI A C, et al. The graph neural network model [J]. IEEE Transactions on Neural Networks, 2009,20(1): 61-80.
- [24] ZHANG Z, CUI P, ZHU W. Deep learning on graphs: a survey [EB/OL]. [2020-3-13]. <https://arxiv.org/abs/1812.04202v3>
- [25] ALLAMANIS M, BROCKSCHMIDT M, KHADEMI M. Learning to represent programs with graphs [C] // The 6th International Conference on Learning Representations. Vancouver, 2018.
- [26] GU X, ZHANG H, KIM S. Deep code search [C] // International Conference on Software Engineering, 2018, 933-944.

作者简介:李杨阳(1995—),男,硕士研究生,研究方向为计算机工程。万波(1976—),男,副教授,研究方向为嵌入式系统,人机交互。梁森(1995—),男,硕士研究生,研究方向为系统结构。胡涛(1997—),男,硕士研究生,研究方向为系统结构。