

基于 ArcReLU 函数的神经网络激活函数优化研究

许赞杰 徐菲菲

(上海电力学院计算机科学与技术学院, 上海, 200090)

摘要: 近年来深度学习发展迅猛。由于深度学习的概念源于神经网络, 而激活函数更是神经网络模型在学习理解非线性函数时不可或缺的部分, 因此本文对常用的激活函数进行了研究比较。针对常用的激活函数在反向传播神经网络中具有收敛速度较慢、存在局部极小或梯度消失的问题, 将 Sigmoid 系和 ReLU 系激活函数进行了对比, 分别讨论了其性能, 详细分析了几类常用激活函数的优点及不足, 并通过研究 Arctan 函数在神经网络中应用的可能性, 结合 ReLU 函数, 提出了一种新型的激活函数 ArcReLU。实验证明, 该函数既能显著加快反向传播神经网络的训练速度, 又能有效降低训练误差并避免梯度消失的问题。

关键词: 神经网络; 激活函数; 反正切函数; ArcReLU

中图分类号: TP18 **文献标志码:** A

Optimization of Activation Function in Neural Network Based on ArcReLU Function

Xu Yunjie, Xu Feifei

(College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, 200090, China)

Abstract: Deep learning has developed rapidly in recent years. The concept of deep learning originates from the neural networks. And the activation function is an indispensable part of the neural network model in learning to understand non-linear functions. Therefore, the common activation functions are studied and compared, aiming at the problems of slow convergence speed, local minimum or gradient disappearance of the commonly used activation functions in back propagation neural networks. In this paper, the Sigmoid and ReLU activation functions are compared, their performances are discussed respectively, and the advantages and disadvantages of several common activation functions are analyzed in detail. Finally, a new activation function, ArcReLU, is proposed by studying the possibility of applying Arctan functions in neural networks and combining with ReLU functions. Experiments show that the function can not only significantly accelerate the training speed of BP neural network, but also effectively reduce the training error and avoid the problem of gradient disappearance.

Key words: neural network; activation function; inverse tangent function; ArcReLU

引言

深度学习的概念源于人工神经网络^[1]的研究,而激活函数更是人工神经网络模型在理解和学习非线性函数时不可或缺的部分。若不使用激活函数,神经网络每一层的输出都是上一层输入的线性函数,无论神经网络具有多少层,输出皆为输入的线性组合,该类情况就是最基本的感知机。因此需要使用激活函数为神经元引入非线性因素,使神经网络可以任意逼近任何非线性函数,这样才能让神经网络应用到众多的非线性模型中。本文则基于误差反向传播神经网络,对常用激活函数进行研究对比,而后对其不足之处进行改进,以提高其最终的收敛速度和计算精度。在激活函数中最为常见的为 Sigmoid 系函数和 ReLU 系函数。

Sigmoid 函数^[2]在 Sigmoid 系函数中最具代表性,其具有软饱和性^[3],即该函数在定义域内处处可导,但当输入值过大或过小时,其斜率趋近于 0,同时其导数也趋近于 0,这将导致向底层传递时的梯度变得非常小。由于其在 BP 神经网络向下传导的梯度内包含了一个自身关于输入的导数因子,一旦输入落入饱和区之中,该因子将会接近于 0,致使向底层传递的梯度变得极小,此时,神经网络的参数很难得到有效的训练,即会出现梯度消失^[4]现象。这一现象使得 BP 网络一直难以得到有效的训练。同时,由于基本没有信号通过神经元传至权重再到输入值,这时梯度在模型更新中将难以起到作用。这也导致了无法对参数进行微调,随即影响到最终结果的精确值。这些也是阻碍神经网络进一步发展的重要原因。

Tanh 函数^[5]作为 Sigmoid 函数的一个变体,同样存在软饱和性的问题,但该函数以 0 点为中心,缓解了 Sigmoid 均值偏移的问题,同时提高了收敛速度。

经过学者们的研究,目前较为流行的神经网络的激活函数为修正线性单元(ReLU)^[6]。它首先被用于限制玻尔兹曼机器,然后成功应用于神经网络。ReLU 的导数在正轴部分恒为 1,保持梯度不衰减,从而有效缓解了梯度消失的问题。该函数在反向传播过程中能够将梯度更好地传递给后层网络,同时计算速度更快。但其负轴部分会使其输入值落入硬饱和区,致使出现神经元死亡的情况,然而这一现象可以减少参数间的相互依存关系,继而缓解了过拟合问题的发生。另一方面,该函数也存在均值偏移的问题,即输出均值恒大于 0。

指数线性单元(Exponential linear unit, ELU)^[7]是 ReLU 的修正类激活函数。当输入值为负数时,ELU 输出一个负值,这使得单元激活均值可以趋近于 0,同时只需要更低的计算复杂度。ELU 在输入取较小值时具有软饱和的特性,提升了对噪声的鲁棒性。

本文通过对经典的 Sigmoid 系和 ReLU 系激活函数的研究与分析,提出与 Sigmoid 系激活函数同为 S 型图像的反正切函数(Arctan),分析其在 BP 神经网络中应用的优点和不足之处。最终提出构想,通过结合 ReLU 函数和 Arctan 函数,构造出一种新型的激活函数 ArcReLU。实验结果可以说明,一方面, ArcReLU 函数相较于 Sigmoid 系和 ReLU 系函数具有更快的收敛速度并能有效地降低训练误差,同时还能有效缓解梯度消失的问题,解决 ReLU 函数具有的硬饱和性,进一步由于其负轴部分的导数趋于 0 的速度更慢,相较于 Sigmoid 系函数更为缓和,这一点使其负轴部分的饱和区间更为广泛,学习效率也会得到提高。另一方面,其导数的计算相较于另外两系的激活函数也将更为昂贵。

1 背景知识

1.1 ReLU 函数

ReLU 函数有效地解决了 Sigmoid 系函数在神经网络中梯度消失的问题,但从函数图(图 1)中不难

看出,该函数依旧存在均值偏移的问题。其定义如下

$$f(x) = \max(0, x) \tag{1}$$

从函数图像及表达式中可以看出,当 $x \geq 0$ 时,其导数值恒为 1,因此,ReLU 函数在 $x \geq 0$ 时能够保持梯度不衰减,可以有效缓解梯度消失的问题。ReLU 函数在反向传播过程中能够将梯度更好地传递给后层网络,同时计算速度较快。当 $x < 0$ 时该函数具有硬饱和性^[2]。如果此时有输入值落入该区域,则该神经元的梯度将永远为 0,其对应权重也将无法更新,即出现神经元死亡的情况,致使计算结果不收敛。由于一部分的神经元输出为 0,减少了参数间的相互依存关系,这也有效缓解了过拟合问题的发生。而 ReLU 函数在 $x < 0$ 时输出为 0,使得整体输出均值大于 0,即存在均值偏移问题^[6],这也在一定程度上造成了神经网络的稀疏特性。

1.2 Arctan 函数

在图像上与 Sigmoid 系函数相类似的 Arctan 函数,输出范围在 $(-\frac{\pi}{2}, \frac{\pi}{2})$,其定义为

$$f(x) = \arctan x \tag{2}$$

Arctan 函数图像如图 2 所示。从表达式和图像中可以看出,Arctan 函数具有软饱和性,即会使 BP 神经网络出现梯度消失的现象。相较于其他 Sigmoid 系函数,Arctan 函数更为平缓,这使其比其他双曲线更为清晰,也意味着该函数没有 Sigmoid 和 Tanh 函数那么敏感,处于饱和度的区间范围比这两个函数更广。同时,其导数趋于 0 的速度更慢,这意味着学习效率更高,也能更好地缓解梯度消失的问题。然而,其导数的计算将比 Tanh 函数更加昂贵。

2 基于 ReLU 函数的变体 ArcReLU 函数

基于对上述经典激活函数的研究及分析,结合 ReLU 函数以及 Arctan 函数的优点,为缓解 ReLU 函数神经元死亡的问题,降低 Arctan 函数的计算消费,结合两者构造出一种新的 ArcReLU 激活函数。将 ReLU 函数输入值小于 0 的部分替换为 $\frac{2}{\pi} \arctan x$,在输入值大于 0 的部分使用 ReLU 函数。在使用 Arctan 函数时,为了限制其输出范围,将算子乘以 $\frac{2}{\pi}$,当网络进入一些比较大的输入值时也能保持稳定。其定义如下

$$f(x) = \begin{cases} x & x > 0 \\ \frac{2}{\pi} \arctan x & x \leq 0 \end{cases} \tag{3}$$

ArcReLU 函数图像如图 3 所示。由图像上可以初步推断,该函数在其定义域范围内可导且单调递

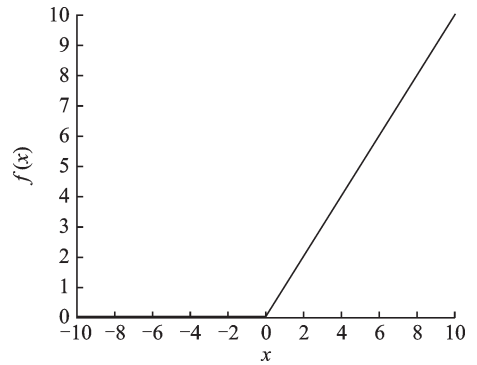


图 1 ReLU 函数图像

Fig.1 Graph of function ReLU

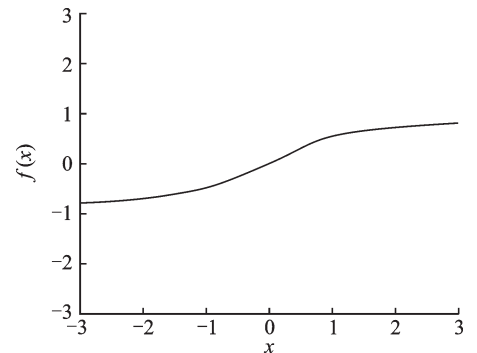


图 2 Arctan 函数图像

Fig.2 Graph of function Arctan

表 2 鸢尾花数据集

Tab. 2 Iris data set

萼片长度	萼片宽度	花瓣长度	花瓣宽度	种类
------	------	------	------	----

表 3 汽车评估数据集

Tab. 3 Car evaluation data set

售价	保养价格	车门数量	乘坐人数	汽车后备箱	安全性	评估结果
----	------	------	------	-------	-----	------

表 4 美国人口普查收入数据集

Tab. 4 Adult data set

年龄	职业类型	序号	受教育程度	受教育时间	婚姻状况	职业	社会角色	种族	性别	资本收益	资本支出	每周工作时间	国籍	收入
----	------	----	-------	-------	------	----	------	----	----	------	------	--------	----	----

表 5 阿维拉数据集

Tab. 5 Avila data set

柱间距离	顶边	底边	开发	行数	模量比	行间间距	权重	峰数	模数比/行间间距	类别
------	----	----	----	----	-----	------	----	----	----------	----

此缓解局部最小点的出现并提高收敛速度。

鉴于不同的评价指标往往具有不同的量纲和量纲单位,将会影响到数据分析的结果。为了消除指标之间的量纲影响,在开始实验前对数据进行了 z-score 标准化的预处理,使指标的特征保持在相同范围内,以解决数据指标之间的可比性。为了直观地比较各激活函数之间的训练时间和误差率的差异,以下通过图表的形式将实验结果进行展示。

3.2 皮马印第安人糖尿病数据集实验结果

表 6 为 3 种函数在第 1 组数据集中的计算时间,从小到大的排序为 ReLU < ELU < ArcReLU,由此可以得出在本次实验中 ArcReLU 的计算相较于另外两个函数稍为昂贵。图 4 显示了各激活函数在本次实验中的收敛速度。从图 4 可以看出实验过程中各函数的收敛速度从小到大排序为 ReLU < ELU < ArcReLU,由此进一步验证前期理论证明的 ArcReLU 收敛速度高于另外两种函数。

图 5 是 3 种激活函数的受试者工作特性曲线 ROC 比较图,从图中较难看出这 3 种激活函数的优劣,因此需要通过求得 ROC 曲线下的面积值 AUC 进行比对。图中横坐标 FPR 表示将负例错分为正例的概率,纵坐标 TPR 表示将正例分对的概率。各函数 AUC 值及分类精度均值如表 7,8 所示。从表 7,8 可以看出 ArcReLU 函数的 AUC 面积以及分类精度均大于 ReLU 以及 ELU,由此可以得出,在当前数据集中 ArcReLU 的分类效果优于另外两种函数。

表 6 第 1 组数据集中各激活函数计算时间

Tab. 6 Calculating time of each activation function in the first data set

函数	ReLU	ELU	ArcReLU
1	0:05:26.44	0:05:41.80	0:05:53.30
2	0:05:25.32	0:05:41.38	0:05:55.25
3	0:05:28.54	0:05:45.03	0:05:53.48
4	0:05:29.10	0:05:44.41	0:05:52.83
5	0:05:45.17	0:06:08.54	0:06:21.32
6	0:05:43.21	0:05:54.99	0:06:05.17
7	0:06:03.29	0:06:19.58	0:06:13.36
8	0:05:48.01	0:05:56.59	0:06:04.10
9	0:05:55.68	0:06:08.18	0:06:15.23
10	0:05:46.40	0:06:01.08	0:06:14.12
平均	0:05:41.12	0:05:56.16	0:06:04.83

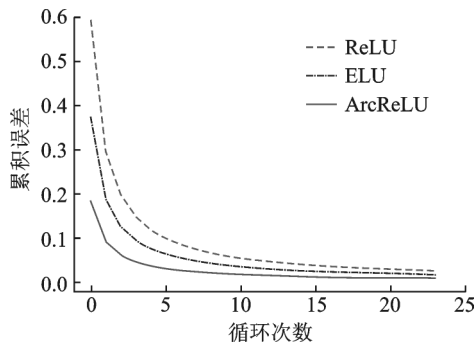


图4 第1组数据集中各激活函数收敛速度比较图
Fig.4 Convergence rate comparison of each activation function in the first data set

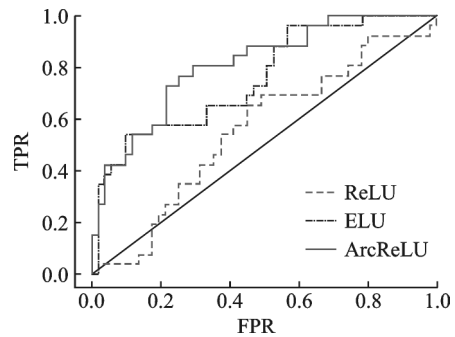


图5 第1组数据集中各激活函数ROC比较图
Fig.5 ROC comparison of each activation function in the first data set

表7 第1组数据集中各激活函数AUC

Tab. 7 AUC comparison of each activation function in the first data set

函数	ReLU	ELU	ArcReLU
1	0.470 2	0.623 6	0.614 4
2	0.425 7	0.752 9	0.763 9
3	0.555 1	0.756 4	0.810 0
4	0.462 6	0.596 9	0.620 8
5	0.518 2	0.727 4	0.725 2
6	0.489 6	0.748 5	0.790 9
7	0.457 0	0.650 5	0.634 8
8	0.499 2	0.802 7	0.783 4
9	0.672 7	0.631 9	0.618 1
10	0.415 6	0.743 7	0.700 7
平均	0.496 6	0.703 5	0.706 2

表8 第1组数据集中各激活函数分类精度均值

Tab. 8 Mean classification accuracy of each activation function in the first data set

函数	ReLU	ELU	ArcReLU
训练精度均值	65.63	65.93	69.24
测试精度均值	65.74	65.88	68.35

3.3 鸢尾花数据集分类实验结果

表9为3种函数在第2组数据集中的计算时间,从小到大的排序为 $\text{ReLU} < \text{ELU} < \text{ArcReLU}$,由此可以得出在本次实验中ArcReLU的计算相较于另外两个函数较为昂贵。图6显示了各激活函数在本次实验中的收敛速度。从图6可以看出实验过程中各函数的收敛速度从小到大排序为 $\text{ReLU} < \text{ELU} < \text{ArcReLU}$,由此可以得出ArcReLU的收敛速度高于另外两种函数。

图7是3种激活函数的ROC比较图,从图中较难看出这3种激活函数的优劣,因此通过计算ROC曲线下的面积值AUC进行比对,各函数AUC及分类精度均值如表10,11所示。从表10,11可以看出

表 9 第 2 组数据集中各激活函数计算时间

Tab. 9 Calculating time of each activation function in the second data set

	ReLU	ELU	ArcReLU
1	0:00:20.51	0:00:21.91	0:00:23.08
2	0:00:20.72	0:00:21.80	0:00:22.58
3	0:00:21.38	0:00:22.02	0:00:22.81
4	0:00:20.99	0:00:22.72	0:00:23.44
5	0:00:21.53	0:00:21.71	0:00:23.28
6	0:00:20.99	0:00:21.63	0:00:23.53
7	0:00:20.70	0:00:21.78	0:00:22.60
8	0:00:21.06	0:00:21.66	0:00:22.55
9	0:00:20.78	0:00:21.56	0:00:22.57
10	0:00:20.57	0:00:21.68	0:00:22.75
平均	0:00:20.92	0:00:21.85	0:00:22.92

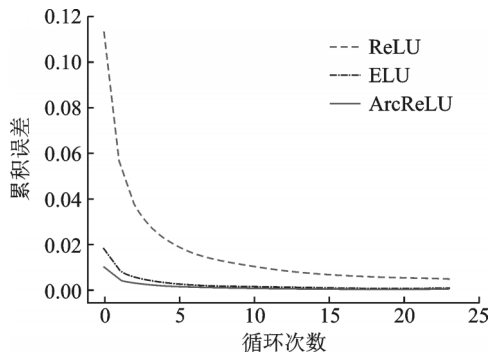


图 6 第 2 组数据集中各激活函数收敛速度比较图
Fig.6 Convergence rate comparison of each activation function in the second data set

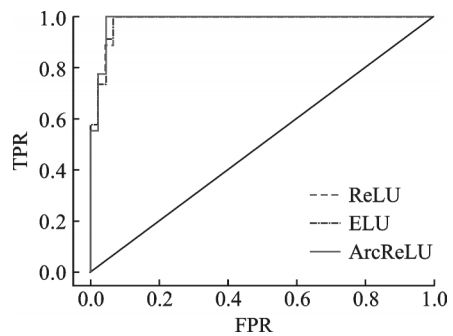


图 7 第 2 组数据集中各激活函数 ROC 比较图
Fig.7 ROC comparison of each activation function in the second data set

ArcReLU 函数的 AUC 面积以及分类精度均大于 ReLU 以及 ELU,由此可以得出,在当前数据集中 ArcReLU 的分类效果优于另外两种函数。

3.4 汽车评估实验结果

表 12 为 3 种函数在第 3 组数据集中的计算时间,从小到大的排序为 $ReLU < ELU < ArcReLU$,由此可以得出在本次实验中 ArcReLU 的计算相较于另外两个函数较为昂贵。图 8 显示了各激活函数在本次实验中的收敛速度。从图 8 可以看出实验过程中各函数的收敛速度从小到大排序为 $ReLU < ELU < ArcReLU$,由此可以得出 ArcReLU 的收敛速度高于另外两种函数。

图 9 是 3 种激活函数的 ROC 比较图,从图中较难看出这 3 种激活函数的优劣,因此通过计算 ROC 曲线下的面积值 AUC 进行比对得出结论。各函数 AUC 及分类精度均值如表 13, 14 所示。从表 13, 14 可以看出 ArcReLU 函数的 AUC 面积以及分类精度均大于 ReLU 以及 ELU,由此可以得出,在当前数据集中 ArcReLU 的分类效果优于另外两种函数。

表 10 第 2 组数据集中各激活函数 AUC

Tab. 10 AUC comparison of each activation function in the second data set

函数	ReLU	ELU	ArcReLU
1	0.983 2	0.976 7	0.984 2
2	0.983 2	0.982 7	0.984 2
3	0.981 2	0.970 7	0.979 7
4	0.983 7	0.980 1	0.984 7
5	0.981 2	0.980 7	0.981 2
6	0.477 8	0.977 6	0.986 2
7	0.980 2	0.979 7	0.979 7
8	0.983 7	0.973 2	0.982 7
9	0.990 6	0.980 1	0.989 6
10	0.982 7	0.979 7	0.981 2
平均	0.932 8	0.978 1	0.983 3

表 11 第 2 组数据集中各激活函数分类精度均值

Tab. 11 Mean classification accuracy of each activation function in the second data set

%

函数	ReLU	ELU	ArcReLU
训练精度均值	58.56	92.33	92.56
测试精度均值	59.00	93.00	93.10

表 12 第 3 组数据集中各激活函数计算时间

Tab. 12 Calculating time of each activation function in the third data set

函数	ReLU	ELU	ArcReLU
1	0:09:17.69	0:09:47.49	0:10:12.74
2	0:09:07.61	0:09:34.86	0:10:00.52
3	0:08:40.51	0:09:17.14	0:09:53.02
4	0:09:04.49	0:09:12.60	0:09:36.44
5	0:08:47.47	0:09:15.80	0:09:37.68
6	0:08:49.29	0:09:16.88	0:09:40.56
7	0:09:43.18	0:10:15.65	0:10:04.45
8	0:09:29.01	0:10:12.80	0:10:45.25
9	0:09:17.57	0:09:50.08	0:10:26.01
10	0:09:19.51	0:10:05.43	0:10:24.07
平均	0:09:09.63	0:09:40.87	0:10:04.07

3.5 美国人口普查收入实验结果

表 15 为 3 种函数在第 4 组数据集中的计算时间,从小到大的排序为 $\text{ReLU} < \text{ArcReLU} < \text{ELU}$,由此可以得出在本次实验中 ELU 的计算相较于另外两个函数较为昂贵。

这一结果与前 3 组的结果有所不同。由于每组实验都是在相同的运行环境下计算的,从以上实验结果看,ELU 适合数据量较小的计算,ArcReLU 适合数据量大的计算。为了确认随着数据量的增减是

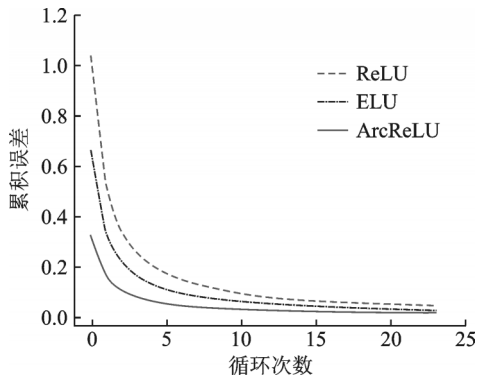


图8 第3组数据集中各激活函数收敛速度比较图
Fig.8 Convergence rate comparison of each activation function in the third data set

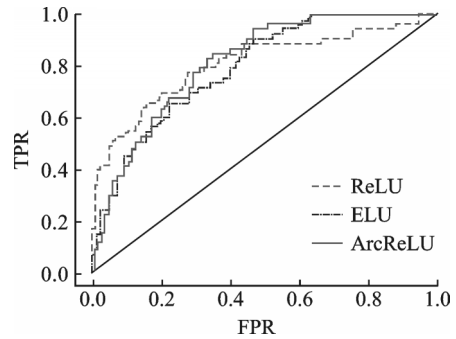


图9 第3组数据集中各激活函数ROC比较图
Fig.9 ROC comparison of each activation function in the third data set

表 13 第3组数据集中各激活函数 AUC

Tab. 13 AUC comparison of each activation function in the third data set

函数	ReLU	ELU	ArcReLU
1	0.472 7	0.767 0	0.808 6
2	0.545 1	0.807 3	0.832 5
3	0.441 5	0.768 4	0.816 5
4	0.514 9	0.777 4	0.799 3
5	0.808 3	0.793 7	0.813 1
6	0.830 4	0.732 8	0.788 6
7	0.456 7	0.713 5	0.738 3
8	0.509 2	0.775 4	0.778 2
9	0.519 4	0.815 6	0.821 3
10	0.480 2	0.768 3	0.766 6
平均	0.557 9	0.771 9	0.796 3

表 14 第3组数据集中各激活函数分类精度均值

Tab. 14 Mean classification accuracy of each activation function in the third data set

函数	ReLU	ELU	ArcReLU
训练精度均值	73.65	75.69	76.79
测试精度均值	73.78	75.58	77.46

否对 ReLU 函数的计算时间有影响,在 3.6 节中分别计算了 5 组实验中, ArcReLU 相较于 ELU 的计算时间增量和 ArcReLU 相较于 ReLU 的计算时间增量,结果如表 16 所示。图 10 显示了各激活函数在本次实验中的收敛速度。从图 10 可以看出实验过程中各函数的收敛速度从小到大排序为 ReLU < ELU < ArcReLU, 由此可以得出 ArcReLU 的收敛速度高于另外两种函数。

图 11 是 3 种激活函数的 ROC 比较图,从图中较难看出这 3 种激活函数的优劣,因此通过计算 ROC 曲线下的面积值 AUC 进行比对得出结论。各函数 AUC 和分类精度均值如表 17, 18 所示。从表 17, 18 可以看出 ArcReLU 函数的 AUC 面积以及分类精度均大于 ReLU 以及 ELU, 由此可以得出,在当前数

表 15 第 4 组数据集中各激活函数计算时间

Tab. 15 Calculating time of each activation function in the fourth data set

函数	ReLU	ELU	ArcReLU
1	0:33:33.83	0:37:04.69	0:36:36.00
2	0:33:54.20	0:37:06.58	0:38:25.26
3	0:34:00.34	0:37:04.62	0:36:03.46
4	0:34:09.22	0:37:07.00	0:36:22.22
5	0:33:54.67	0:37:15.03	0:36:27.91
6	0:34:45.88	0:37:50.40	0:36:46.70
7	0:35:37.62	0:38:01.66	0:37:17.77
8	0:33:49.48	0:37:01.78	0:37:51.61
9	0:36:26.11	0:39:55.57	0:37:56.82
10	0:36:16.60	0:40:31.25	0:39:14.33
平均	0:34:44.79	0:41:42.02	0:37:24.31

表 16 各激活函数间计算时间增量的比较

Tab. 16 Comparisons of calculating time increments among activation functions

实验	数据集名	数据集大小/KB	ArcReLU 相较于 ELU 的计算时间增量/%	ArcReLU 相较于 ReLU 的计算时间增量/%
1	鸢尾花的分类	1.74	4.9	9.56
2	皮马印第安人糖尿病的数据统计	23.4	2.43	6.95
3	汽车评估	25.3	3.99	9.9
4	美国人口普查收入	90.5	-10.3	7.65
5	阿维拉	1 140	1.12	1.28

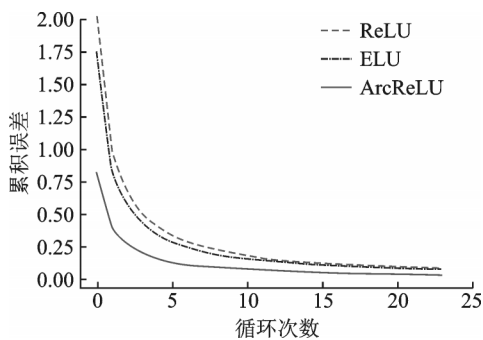


图 10 第 4 组数据集中各激活函数收敛速度比较
Fig.10 Convergence rate comparison of each activation function in the fourth data set

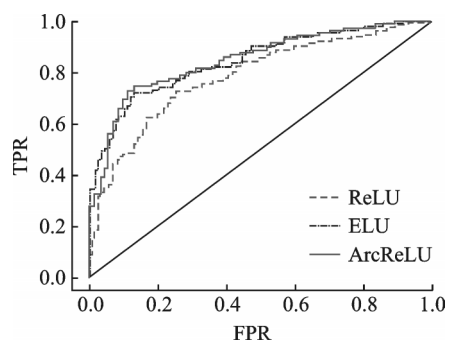


图 11 第 4 组数据集中各激活函数 ROC 比较
Fig.11 ROC comparison of each activation function in the fourth data set

据集中 ArcReLU 的分类效果优于另外两种函数。

3.6 阿维拉实验结果

表 19 为 3 种函数在第 5 组数据集中的计算耗时,从小到大的排序为 $\text{ReLU} < \text{ELU} < \text{ArcReLU}$,由此可以得出在本次实验中 ArcReLU 的计算相较于另外两个函数较为昂贵。由于 5 组实验数据中第 4 组的

表 17 第 4 组数据集中各激活函数 AUC

Tab. 17 AUC comparison of each activation function in the fourth data set

函数	ReLU	ELU	ArcReLU
1	0.500 6	0.836 0	0.836 4
2	0.778 9	0.798 1	0.802 1
3	0.782 5	0.803 7	0.807 5
4	0.786 0	0.846 2	0.851 8
5	0.761 9	0.837 5	0.841 5
6	0.747 8	0.841 2	0.841 3
7	0.777 5	0.752 3	0.783 5
8	0.536 4	0.803 6	0.808 4
9	0.763 2	0.793 6	0.785 9
10	0.548 9	0.770 2	0.759 0
平均	0.698 4	0.808 2	0.811 7

表 18 第 4 组数据集中各激活函数分类精度均值

Tab. 18 Mean classification accuracy of each activation function in the fourth data set

%

函数	ReLU	ELU	ArcReLU
训练精度均值	60.19	77.18	77.41
测试精度均值	60.12	76.74	77.23

表 19 第 5 组数据集中各激活函数计算时间

Tab. 19 Calculating time of each activation function in the fifth data set

函数	ReLU	ELU	ArcReLU
1	2:24:43.66	2:25:54.31	2:29:51.45
2	2:28:03.93	2:29:07.71	2:33:03.55
3	2:49:25.60	2:34:03.75	2:34:47.67
4	2:39:48.43	2:39:25.50	2:40:57.36
5	2:37:04.15	2:42:47.76	2:44:04.48
6	2:40:25.60	2:43:18.96	2:46:54.79
7	2:51:43.59	2:52:32.24	2:56:51.12
8	2:23:15.66	2:27:16.91	2:30:43.26
9	2:47:21.39	2:48:32.57	2:50:29.54
10	2:33:36.81	2:34:53.21	2:27:42.96
平均	2:37:32.54	2:37:47.24	2:39:33.45

实验结果较为不同,此处分别计算了 5 组实验中,ArcReLU 相较于 ELU 的计算时间增量和 ArcReLU 相较于 ReLU 的计算时间增量,见表 16。

基于表 6,9,12,15 和表 19 中的各激活函数计算时间,可以得出结论,随着数据集的增大,ArcReLU 所需的计算时间也会增加,同时从实验结果可以看出三者之间的时间增量在逐步缩小。在后续的研究工作中,将添加更多不同的数据集,对相同的数据集反复进行计算,排除单次实验的特殊性,从而进行进一步的研究。

图 12 显示了各激活函数在本次实验中的收敛速度。从图 12 可以看出实验过程中各函数的收敛速度从小到大排序为 $ReLU < ELU < ArcReLU$, 由此可以得出 ArcReLU 的收敛速度高于另外两种函数。

图 13 是 3 种激活函数的 ROC 比较图, 从图中较难看出这 3 种激活函数的优劣, 因此通过计算 ROC 曲线下的面积值 AUC, 进行比对得出结论。各函数 AUC 和分类精度均值如表 20, 21 所示。从表 20, 21 可以看出 ArcReLU 函数的 AUC 面积以及分类精度均大于 ReLU 以及 ELU, 由此可以得出, 在当前数据集中 ArcReLU 的分类效果优于另外两种函数。

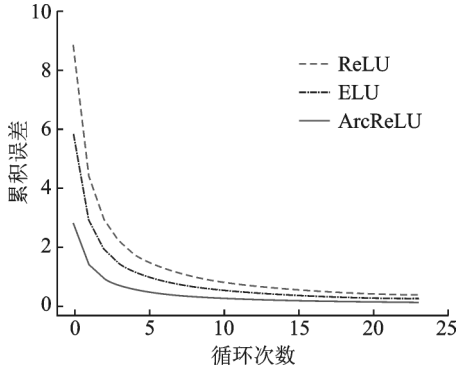


图 12 第 5 组数据集中各激活函数收敛速度比较
Fig.12 Convergence rate comparison of each activation function in the fifth data set

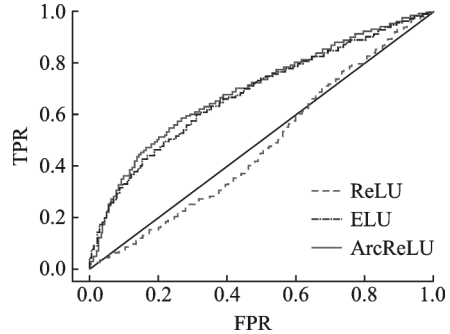


图 13 第 5 组数据集中各激活函数 ROC 比较
Fig.13 ROC comparison of each activation function in the fifth data set

表 20 第 5 组数据集中各激活函数 AUC

Tab. 20 AUC comparison of each activation function in the fifth data set

函数	ReLU	ELU	ArcReLU
1	0.514 0	0.520 5	0.690 7
2	0.492 1	0.655 8	0.667 1
3	0.521 0	0.581 6	0.633 3
4	0.769 8	0.710 9	0.751 1
5	0.545 9	0.614 7	0.652 8
6	0.521 1	0.523 3	0.580 2
7	0.619 2	0.635 3	0.573 1
8	0.673 7	0.699 7	0.703 3
9	0.538 8	0.541 5	0.595 6
10	0.678 7	0.558 1	0.673 5
平均	0.587 4	0.604 1	0.652 1

表 21 第 5 组数据集中各激活函数分类精度均值

Tab. 21 Mean classification accuracy of each activation function in the fifth data set

函数	ReLU	ELU	ArcReLU
训练精度均值	68.66	69.94	70.69
测试精度均值	68.05	68.37	70.59

%

从上述5组实验结果可以看出,虽然ArcReLU的计算时间略多于ReLU和ELU两个函数,但5次实验结果均说明其收敛速度、分类精度以及AUC面积高于其他两种激活函数。同时也验证了Arctan函数由于导数趋于0的速度更为缓慢,因此当输入值落入负轴时,ArcReLU函数的收敛速度高于另外两个激活函数,即该函数可以有效地减少训练误差。

4 结束语

本文通过分析研究经典的激活函数,结合Arctan函数的性质,构造出一种新的激活函数ArcReLU。随后,通过5组不同的数据集,分别将该函数与两种较为常用的ReLU系激活函数进行对比实验。从结果可以看出,ArcReLU函数相较于另外两个函数,初始的累积误差小,具有较快的收敛速度。伴随着迭代次数的增加,ArcReLU函数将更快趋于平稳,进一步说明它具有较好的收敛性并能有效地降低训练误差。同时,结合ReLU系函数的特性,ArcReLU函数能够有效缓解梯度消失的问题。负轴部分为Arctan函数,可以缓解ReLU函数的硬饱和性,进一步由于Arctan函数更为缓和,使得负轴的饱和区间范围相较于Sigmoid系函数更为广泛。在实验中也能够看出,ArcReLU的AUC值比另外两个函数大,由此可见,ArcReLU的泛化性能优于另外两种函数。

另外,由于ArcReLU函数负轴部分为Arctan函数,三角函数在计算机中的运算复杂度相当于乘除法,而另外两种激活函数的运算复杂度仅相当于加法运算。因此理论上ArcReLU函数在计算消耗方面会略大于另外两个激活函数,所需的计算时间也就略长。通过5组实验结果可以得出,随着数据量的增加,ArcReLU的计算消耗也略为昂贵。下一步研究工作将对于ArcReLU的鲁棒性进行探讨,从而确认其是否适用于无监督学习、多种分类结果的监督学习或深度学习。另外,还会对其计算时间与数据集大小的关联性进行研究,通过添加不同的数据集,反复进行实验,从而确认其是否有所关联。

参考文献:

- [1] Haykin S. A comprehensive foundation[J]. Neural Networks, 1994(2): 71-80.
- [2] Gomar S, Mirhassani M, Ahmadi M. Precise digital implementations of hyperbolic tanh and sigmoid function[C]//Conference on Signals, Systems & Computers. [S.l.]: IEEE, 2016: 1586-1589.
- [3] Gulcehre C, Moczulski M, Denil M, et al. Noisy activation functions [C]//Proceedings of the 33rd International Conference on Machine Learning. New York: ICML, 2016: 1603. 00391.
- [4] Hochreiter S. The vanishing gradient problem during learning recurrent neural nets and problem solutions[J]. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 1998, 6(2): 107-116.
- [5] Djork-arné C, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs)[J]. Computer Science, 2015: 1511.07289v5.
- [6] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]//Proceedings of the 27th International Conference on Machine Learning. Haifa, Isreal: ICML, 2010: 807-814.
- [7] Djork-Arné C, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by exponential linear units (ELUs)[C]//International Conference of Learning Representation 2016. Puerto Rico: ICLR 2016, 2016: 1-14.
- [8] Wu H. Global stability analysis of a general class of discontinuous neural networks with linear growth activation functions[J]. Information Sciences, 2009, 179(19): 3432-3441.

作者简介:



许赞杰(1995-),男,硕士研究生,研究方向:机器学习、人工智能,E-mail:18918171789@189.cn。



徐菲菲(1983-),女,博士,副教授,研究方向:粗糙集、模糊粗糙集、粒计算、数据挖掘、人工智能,E-mail:xufeifei-1983@hotmail.com。