

逆传输类神经网络中非对称数据优化算法研究^①

魏传佳

(泉州轻工职业学院智能工学院 福建泉州 362200)

摘要 [目的] 本算法在不影响算法复杂度的情况下, 提高了对非对称数据运算的精确性与有效性。[方法] 提出一种修改权重的逆传输类神经网络算法, 通过修改自学习效率, 对占有较少类的数据分配不同权重来解决非对称平衡问题。[结果] 仿真结果表明, 与其他四种分类算法对比, 该算法切实解决数据分类的问题。[局限] 处理海量数据时, 时间冗余度会比较大, 如何处理此问题, 将在以后的研究中改进。[结论] 解决了逆传输类神经网络处理非对称数据时效率低下的问题, 效果显著。

关键词 神经网络; 非对称数据; 逆传输; 算法有效性

The Optimization Algorithm Research for Asymmetrical Data of Reverse Transmission Neural Network

Wei Chuanjia

(School of Intelligent Engineering, Quanzhou Technology College, Quanzhou, Fujian, 362200, China)

Abstract [Objective] This algorithm improves the accuracy and effectiveness of operations on asymmetric data without affecting the complexity of the algorithm. [Methods] An inverse transport neural network algorithm with modified weights is proposed, which solves the asymmetric balance problem by modifying the self-learning efficiency and assigning different weights to the data with fewer classes. [Results] The simulation results show that, compared with the other four classification algorithms, this algorithm can effectively solve the problem of data classification. [Limitation] When dealing with massive data, the time redundancy will be relatively large. How to deal with this problem will be improved in future research. [Conclusions] The problem of low efficiency in processing asymmetric data with reverse transmission neural network is solved, and the effect is remarkable.

Keywords Neural Network; Asymmetric Data; Reverse Transmission; The Effectiveness of the Algorithm

^①本文系 2020 年福建省教育厅科研项目“基于模拟退火算法的无线网络优化算法研究”(项目编号: JAT201502)的研究成果之一。

1 引言

随着数据数量的不断增长,机器学习算法在处理海量数据时,其运算速度会逐渐下降,非对称数据分类是否有达到平衡状态是一个研究热点,所以产生了修改训练集^[1]的算法去处理非对称问题。以往的研究中主要专注于二重数据类别的分类,即只研究两种分类的结果,如:支付设计中的成功与否等。本文主要解决算法在处理非对称数据时,分类效果不佳的问题,利用修改逆传输类神经网络的分类权重方法,在处理非对称数据情况下,仍能保持较高的精准度与较低的时间冗余度。

2 非对称数据分类

本章节中,首先介绍非对称数据的问题,其次介绍目前处理非对称数据的几种方法,最后介绍评定非对称数据分类精准度的方法。

2.1 非对称数据问题描述

非对称数据的问题主要体现在单一数集中,每一个类别的实例与其他样本的数量存在着显著的差异。计算处理有关非对称问题的时候,主要是针对二元次的数据分类。二元次分类指的是如果其中一个实例的数量相对来说是较多的,那么我们把它称为多数类或负类;如果一个实例的数量是较少的,那么我们把它称为少数类或正类。这种分类真实存在于如学生的休学可能性、生物信息、破产评估、疾病发现等领域。而这些应用中重要的信息都属于少数类,如果无法辨识出这些少数类结果,成本代价会非常高昂。出现这种问题的主要原因是针对目标的机器学习算法是通过全局的搜索概念,并不会考虑每一个实例的数量。这会造成对少数类的忽略,因少数类反映出的特征较少,会导致错判率的上升。

以往,评判一个数据集通常使用不平衡率

(IR)去判断一个数据集的不平衡程度。首先,判断数据集中哪个属于多数类,哪个属于少数类,然后用多数类去除以少数类,得到不平衡率(IR),因此,就能够去判断此数据集的不平衡的程度。当进行分类时,不仅不平衡数会影响分类的精准度,数据固有的特征也会影响到分类的精准度,如样本的大小、离群值、边界样本、缺失值等。

2.2 处理非对称数据的方法

处理非对称数据的方法很多,技术特点通常分为三个层次。第一是数据层级的方法,主要修改原始训练集,得到一个近似于对称的数据集,这个数据集可被使用在标准的机器学习方法中;第二是修改演算法,主要是修改现有的演算法的内部操作,使其可以处理非对称数据;第三是成本敏感性分析,给予少数类实例较高的误判成本,对于其他的类别给予较低的误判成本。

数据层级计算方法主要有三种:过采样方法(Oversampling)、欠采样方法(Under Sampling)和杂交方法(Hybrid)。这三种方法中,相对简单的有随机过采样法,它随机地从原始数据集中产生少数类(即正类)数据,直到少数类与多数类数据达到平衡。另一个是随机欠采样法,它会随机删除多数类,直到多数类与少数类相趋近。SMOTE算法与随机过采样法和随机欠采样法相比是一种较为复杂的方法,属于过采样方法的一种,通过合成少数类数据,使得原数据集趋于平衡。

成本敏感性分析是充分考虑错误分类的成本,假设数据资料中有 M 种类别,错误分类成本可以用一个 $M \times M$ 的矩阵来表示,如表1所示,设 $\cos(i, j)$ 为实际类别是 j 的资料被分为 i 的错误分类成本。在表1中, $\cos(T, F) = 200$, $\cos(F, T) = 800$,当且仅当只有两

个类别时, $\text{cost}(i, j)$ 也表示为将实际类别标记为 j 的错误分类成本。

表1 成本敏感性分析矩阵举例

类别	T	F
T	0	2000
F	8000	0

对于错误成本的分类设计方法, 通常不去改变传统的方法, 而是对训练数据进行预处理或者对结果进行结果处理, 用传统算法去考虑错误分类的结果, 这样就不需要去修改原有的分类方法, 因此, 可以直接套用在许多算法里。常用的方法有修改权重法、门限法等。

2.3 评定非对称数据分类精准度

$$\frac{TP + TN}{TP + TN + FP + FN} = \text{ACCURACY (精准度)} \quad (1)$$

此种方法对少数类的分类结果不是很满意, 但对分类的结果的评价却很高。进行分类精准度判断的时候, 采用区分少数类与多数类的计算方法, 本文定义区分度公式(GM)如下:

$$GM = \sqrt{\text{sensitivity} \cdot \text{specificity}} \quad (2)$$

公式(2)中的灵敏度(sensitivity)计算公式为:

$$\text{sensitivity} = \frac{TP}{TP + FN} \quad (3)$$

公式(2)中的特异性(specificity)计算公式为:

$$\text{specificity} = \frac{TN}{FP + TN} \quad (4)$$

此种操作方法可最大化两个类别的精准度, 并且具有很好的平衡性。

对于分类的精准度, 需要通过一个混合矩阵去说明, 混合矩阵是通过每个类别的错误分类或者正确分类的区分而组成, 如表2所示。

表2 成本敏感性分析矩阵举例

	正向预测	反向预测
正确分类	正确 (T) 正向 (P)	错误 (F) 反向 (N)
错误分类	错误 (F) 正向 (P)	正确 (T) 反向 (N)

当进行分类精准度判断的时候, 将其分为两种计算方式: 一种是不区分少数类和多数类的计算; 一种是区分少数类和多数类的计算。精准度公式 (ACCURACY) 如下:

3 逆传输类神经网络

3.1 类神经网络

人脑中有超过1000亿个神经细胞, 每个神经细胞中, 含有神经元^[2]、细胞核、轴突、树突、突触等。细胞核为处理器, 轴突为信号传输介质, 树突为信号线, 突触为神经元之间的连接点。图1为神经元; 图2为人工类神经元运算模型; 图3为类神经网络模型。

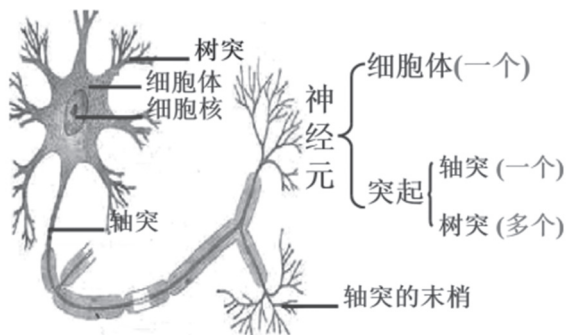


图1 神经元结构

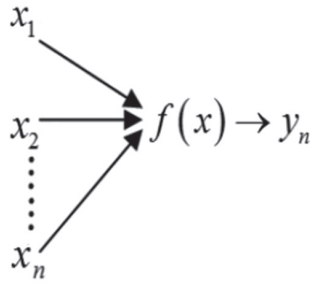


图2 人工类神经元运算模型

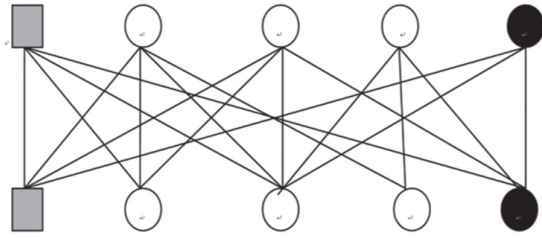


图3 类神经网络模型

类神经网络是仿神经网络去处理复杂网络问题，信息来源从人工神经元与外在环境中获得，根据不同的网络拓扑结构和不同的机器自学习算法去训练类神经网络，最终获得目标值。

3.2 逆传输类神经网络

逆传输类神经网络^[3-5]是一种检测型自学习算法，利用多层次架构模式去构造模型。本文定义公式如下：

i 为输入层的第 i 个节点， $i=1, 2, \dots, p$ 。

j 为隐藏层的第 j 个节点， $j=1, 2, \dots, h$ 。

k 为输出层的第 k 个节点， $k=1, 2, \dots, q$ 。

l 为第 l 个训练元素， $l=1, 2, \dots, n$ 。

w_{kj} 为隐藏层与输出层之间节点的权重。

z_j^l 为第 l 个训练元素在隐藏层中节点 j 的输出值。

y_k^l 为第 l 个训练元素在输出层中节点 k 的输出值。

f 为神经元动态化函数。

d_k^l 为第 l 个训练元素在输出层中节点 k 的目标值。

w_{j0} 为隐藏层的门限值。

w_{k0} 为输出层的门限值。

w_{ji} 为输入层与输出层之间节点的权重。

η 为自学习效率。

x_i^l 为第 l 个训练元素中节点 i 的输入值。

δ_k 为输出层的误差。

3.3 逆传输类神经网络原始算法

①初始化各项参数；

②初始化 w_{ji} 与 w_{kj} ，选定动态化函数；

③选取训练集 $x_i^l = (x_1^l, x_2^l, \dots, x_p^l)$ ，目标集 $d_i^l = (d_1^l, d_2^l, \dots, d_p^l)$ ；

④计算隐藏层输出值 z_j^l 、输出层的输出值 y_k^l ；

⑤求出误差函数 E ；

⑥求出输出层的差距量 δ_k 、隐藏层的差距量 δ_j ；

$$\delta_k = (d_k - y_k) y_k (1 - y_k) \quad (5)$$

$$\delta_j = \sum_k (\delta_k w_{kj}) z_j (1 - z_j) \quad (6)$$

⑦求出输出层与隐藏层之间的权重修正值 Δw_{kj}^l 、输入层与隐藏层之间的权重修正值 Δw_{ji}^l ；

$$\Delta w_{kj}^l = \eta \delta_k z_j \quad (7)$$

$$\Delta w_{ji}^l = -\frac{\partial E}{\partial w_{ji}} = \eta \delta_j x_i \quad (8)$$

⑧权重更新；

$$w_{kj}^{l+1} = w_{kj}^l + \Delta w_{kj}^l \quad (9)$$

$$w_{ji}^{l+1} = w_{ji}^l + \Delta w_{ji}^l \quad (10)$$

⑨返回步骤③，循环计算，令 $l=1+l$ ，直到训练完所有元素；

⑩循环计算，直到循环次数达到最大为止。

4 改进优化逆传输类神经网络算法

针对非对称数据分类问题^[6-8]，本文提出改进权重的逆传输类神经网络算法 (MWRTNN)，此算法通过修改自学习效率，对占有较少类的数据分配高权重^[9]来解决非对称平衡问题。算法步骤如下：

①网络拓扑模型建立, 初始化各层数据以及最大自学习周期, 令 $l=1$;

②权重初始化 w_{ji} 与 w_{kj} , 选定节点输出活化函数;

③选取训练集, 令 $x^l_i = (x^l_1, x^l_2, \dots, x^l_p)$, $d^l_i = (d^l_1, d^l_2, \dots, d^l_p)$;

④求出隐藏层各个节点的输出 z^l_j 、输出层各个节点的输出 y^l_k ;

⑤求出误差函数 E ;

⑥求出输出层的差距量 δ_k 、隐藏层的差距量 δ_j ;

$$\delta_k = (d_k - y_k)y_k(1 - y_k) \quad (11)$$

$$\delta_j = \sum_k (\delta_k w_{kj})z_j(1 - z_j) \quad (12)$$

⑦在训练数据时, 对于连接点的权重修改问题, 先去判断训练集属于正分类还是负分类。如果属于正分类, 就使用正分类权重^[12]; 如果属于负分类, 就使用负分类权重。求出输出层与隐藏层之间的连接点修正权重, 正分类修正权重为 Δw_{kj}^+ , 负分类修正权重为 Δw_{kj}^- ; 求出隐藏层与输入层之间的连接点^[13]修正权重, 正分类修正权重为 Δw_{ji}^+ , 负分类修正权重为 Δw_{ji}^- ;

$$\Delta w_{kj}^+ = \eta IR \delta_k z_j \quad (13)$$

$$\Delta w_{kj}^- = \eta \delta_k z_j \quad (14)$$

$$\Delta w_{ji}^+ = \eta IR \delta_j x_i \quad (15)$$

$$\Delta w_{ji}^- = \eta \delta_j x_i \quad (16)$$

⑧连接点权重更新, 求出输出层与隐藏层之间的连接点权重更新, 正分类修正权重更新为 w_{kj}^{l+1} , 如式(17)所示, 负分类修正权重更新为 w_{kj}^{l+1} , 如式(18)所示; 求出隐藏层与输入层之间的连接点权重更新, 正分类修正权重更新为 w_{ji}^{l+1} , 如式(19)所示, 负分类修正权重更新为 w_{ji}^{l+1} , 如式(20)所示;

$$w_{kj}^{l+1} = w_{kj}^l + \Delta w_{kj}^+ \quad (17)$$

$$w_{kj}^{l+1} = w_{kj}^l + \Delta w_{kj}^- \quad (18)$$

$$w_{ji}^{l+1} = w_{ji}^l + \Delta w_{ji}^+ \quad (19)$$

$$w_{ji}^{l+1} = w_{ji}^l + \Delta w_{ji}^- \quad (20)$$

⑨令 $l=l+1$, 返回到步骤③, 直至训练完成;

⑩重复执行步骤②至步骤⑨, 直到匹配最大周期。

5 实验结果比较

为了分析算法执行效果, 本文从UCI库中, 选取了三种不同的数据集: KDD CUP99、RLCP、POKE HAND DATA SET。这三种数据集属于多类别分类集^[9], 将其分为多个数据集, 用来解决类间的分类问题。三个数据集信息如表3所示, 其中包含实例数量(*Ex.)、各类别所占百分比 [%Class (Maj; Min)]、不平衡比率(Ubr)。为扩展实验, 本文使用5-Fold Cross Validation进行验证。

表3 非对称数据表

Data sets	*Ex	%Class (Maj; Min)	Ubr
Ku U2L	972832	(99.994%;0.006%)	18707.2
Ku R2L	973906	(99.883%;0.117%)	863.925
Ku PRB	1013882	(95.945%;4.055%)	23.666
Pr 0-2	562531	(91.32%;8.68%)	10.52
Pr 0-3	535335	(95.958%;4.042%)	23.74
Pr 0-4	517681	(99.231%;0.769%)	129.12
Pr 1-2	481924	(89.867%;10.133%)	8.86
Pr 1-3	454730	(95.241%;4.759%)	20.01
RP	5749131	(99.635%;0.365%)	273.66

为证明RTNN对非对称数据的分类效果的提升，本文以五种算法做比较，结果如表4与表5所示。

表4 RTNN与MWRTNN比较

	RTNN		MWRTNN	
	Train	Test	Train	Test
Pr 0-2	0.282	0.281	0.537	0.535
Pr 0-3	0.222	0.224	0.503	0.502
Pr 0-4	0.289	0.292	0.487	0.487
Pr 1-2	0.046	0.044	0.489	0.493
Pr 1-3	0.033	0.031	0.512	0.514
Ku 2r	0.597	0.599	0.761	0.762
Ku R2L	0.442	0.442	0.557	0.563
Ku PRB	0.000	0.000	0.754	0.741
RP	0.000	0.000	0.613	0.613

由表4可知，RTNN网络在处理非对称数据时，效率不高，而本文提出的MWRTNN算法，大大改善了分类上的效率，对比其他算法，分类效果有了较大的改善，如表5所示。

表5 MWRTNN与其他算法比较

	MWRTNN		RTNN-ROS		RTNN-RUS		RTNN-SMOTE	
	Train	Test	Train	Test	Train	Test	Train	Test
Pr 0-2	0.536	0.535	0.321	0.321	0.421	0.424	0.517	0.517
Pr 0-3	0.503	0.502	0.483	0.483	0.489	0.492	0.495	0.494
Pr 0-4	0.487	0.487	0.568	0.568	0.677	0.676	0.518	0.518
Pr 1-2	0.489	0.493	0.453	0.452	0.479	0.483	0.264	0.264
Pr 1-3	0.512	0.514	0.506	0.506	0.503	0.504	0.434	0.434
Ku 2r	0.761	0.762	0.600	0.600	0.631	0.628	0.600	0.599
Ku R2L	0.557	0.563	0.593	0.593	0.583	0.583	0.605	0.605
Ku PRB	0.754	0.741	0.605	0.606	0.573	0.556	0.648	0.648
RP	0.613	0.613	0.410	0.410	0.501	0.502	0.457	0.456

6 结语

对于逆传输类神经网络处理非对称数据时效率低下的问题，本文提出了一种基于权重修改的逆传输类神经网络算法，该算法切实解

决了数据分类的问题，对比其他算法，效果显著。但该算法在处理海量数据时，时间冗余度会比较大，如何处理此问题，将在以后的研究中继续改进。

参考文献

- [1] LABOVITZ C, JEKEL-JOHNSON S, MCPHERSON D, et al. Internet inter-Domain traffic [C]. [S. l.]: ACM SIGCOMM Conference, 2019.
- [2] 豆育升, 崔晟圆, 唐红, 等. 云数据中心高能效的虚拟机放置算法. 小型微型计算机系统 [J], 2014, 35 (11): 2543—2547.
- [3] FALKENAUER E, DELCHAMBRE A. A Genetic Algorithm for Bin Packing and Line Balancing [C]. [S. l.]: Proceedings of IEEE International Conference on Robotics and Automation, 2018.
- [4] GAO Y, GUAN H, QI Z, et al. A Multi-Objective Ant Colony System Algorithm for Virtual Machine Placement in Cloud Computing [J]. Journal of Computer and System Sciences, 2019, 79 (8): 1230—1242.
- [5] NISHANT K, SHARMA P, KRISHNA V, et al. Load Balancing of Nodes in Cloud Using Ant Colony Optimization [C]. [S. l.]: International Conference on Computer Modeling and Simulation, 2017: 3—8.
- [6] EHLEAarts, PJM Van Laarhoven, A General Approach to Combinatorial Optimization Problems [J]. Philips J Res, 2017, 40 (4): 193—226.
- [7] GANDLII A, HARCLIO-BALTER M, DAS R, et al. Optimal Power Allocation in Server Farms [C]. New York: Measurement and Modeling of Computer Systems, 2019: 157—168.
- [8] CHEN G, HE W, LIU J, et al. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services [C]. [S. l.]: Proceedings of Symposium on Networked Systems Design and Implementation, 2018, 8: 337—350.
- [9] TOE, SATORI. Touching Performance Evaluation of a Green Scheduling Algorithm for Energy Savings in Cloud Computing [C]. [S. l.]: 2019 IEEE International Symposium on Parallel, 2019: 30—50.