基于神经网络的集装箱船港口作业时间预测模型

韩宗垒 徐 斌 陈 佳
(大连海事大学辽宁省物流航运管理系统工程重点实验室 辽宁 大连 116026)

摘 要 集装箱船港口作业时间是制作泊位计划的一个重要依据,而集装箱船港口作业时间获取的主要来源是预测。传统预测方法是用装卸集装箱量除以岸桥装卸效率,预测精度较低,且受多种因素的影响,具有复杂的非线性特点。而神经网络在解决复杂的非线性问题方面具有很强的建模能力,所以选取神经网络建立集装箱船港口作业时间预测模型。通过真实数据对预测模型进行训练学习,用测试数据集对模型进行验证,并且与传统预测方法相对比,结果表明了该预测模型在某集装箱港口预测应用的有效性。

关键词 水路运输 集装箱船港口作业时间 神经网络 预测 模型

中图分类号 TP391 **文献标志码** A DOI:10.3969/j.issn.1000-386x.2021.02.014

CONTAINER SHIP PORT OPERATION TIME PREDICTION MODEL BASED ON NEURAL NETWORK

Han Zonglei Xu Bin Chen Jia
(*Liaoning Provincial Key Laboratory of Logistics and Shipping Management System Engineering, Dalian Maritime University, Dalian 116026, Liaoning, China*)

Abstract Container ship port operation time is an important basis for the production of berth plans, and the main source of container ship port operation time is prediction. The traditional prediction method is the loading and unloading container volume divided by the shore bridge loading and unloading efficiency. The prediction accuracy of this method is low, and it is affected by many factors and has complex nonlinear characteristics. The neural network has strong modeling ability in solving complex nonlinear problems. Therefore, the neural network was selected to establish a container ship port operation time prediction model. The prediction model was trained by real data, and the model was validated by the test data set. Compared with the traditional prediction method, the effectiveness of the prediction model in a container port prediction application is demonstrated.

Keywords Waterway transportation Container ship port operation time Neural networks Prediction Model

0 引 言

泊位是港口的重要资源,合理的泊位计划能有效提高港口生产效率,减少船舶在港时间,从而提高港口的竞争力。Imai 等[1]采用连续区位空间的方法对连续型泊位计划进行研究,建立了船舶等待时间和作业时间最小的数学模型并采用拉格朗日松弛系数算法进行求解。张煜等[2]考虑在泊位计划中岸桥的分配影响集装箱船的装卸作业效率,依据规则建立确定船舶集装箱装卸作业时间和分配岸边起重机的算法。秦进等[3]提出基于时间窗约束的离散型泊位计划模型,建立了模拟退火算法模型进行优化求解。韩晓龙等[4]在港口船舶的目标函数中加入了船舶作业时间,对船舶的服务时间和作业时间提出了时间窗约束,建立了以最小化卸船完工时间为优化目标的混合整数规划模型和约束规划模型。曾庆成等[5]用船舶装卸集装箱量除以岸桥装卸效率来预测集装箱船港口作业时间。

综上所述,大量的研究表明集装箱船港口作业时间对于制作科学高效的泊位计划是非常重要的,研究中用的是传统装卸集装箱量除以岸桥装卸效率的预测方法,预测精度较低,因此寻找一种科学的方法来预测

收稿日期:2019-07-05。**韩宗垒**,硕士生,主研领域:信息管理与信息系统。**徐斌**,副教授。**陈佳**,教授。

集装箱船港口作业时间是非常有意义的。本文基于神经网络针对集装箱船港口作业时间预测的问题建立模型,通过与传统预测方法的对比验证了模型的可行性。

1　问题描述

在集装箱船实际到达港口之前,港口需要根据船期表制作泊位计划确定集装箱船靠泊的时间和位置。集装箱船港口作业时间(船舶开始装卸第一个集装箱到完成装卸最后一个集装箱的这段时间)是制作泊位计划的主要依据,而集装箱船港口作业时间的主要获取方法是预测,所以预测出精确的集装箱船港口作业时间可以提高泊位计划的效率。

传统的集装箱船港口作业时间预测方法是船舶待装卸的集装箱量除以岸桥的装卸效率。这种预测方法不灵活并且预测精度较低,集装箱船港口作业时间受多种因素的影响[6],如船舶类型、岸桥数、装卸集装箱量、天气等,并且存在复杂的非线性关系。考虑到集装箱船港口作业时间与影响因素之间复杂的非线性关系,本文选取了比较适用的 BP 神经网络建立集装箱船港口作业时间预测模型。模型的目标是预测出更加精确的集装箱船港口作业时间,从而保证制作的泊位计划更加科学高效。

2　模型建立

BP(Back Propagation)神经网络是一种多层前馈神经网络,由输入层、隐含层、输出层组成,可以以任意的精确度逼近任意一个连续的函数,所以经常被用于非线性建模、函数逼近和模式分类等方面。BP 神经网络的主要特点:信号是前向传播的,误差是反向传播的。

2.1　网络的层数

1989 年 Robert Hecht-Nielson 证明了对于任何一个闭区间内的函数,都可以用有一个隐含层的 BP 神经网络来逼近,所以一个三层(含一个隐含层)的 BP 神经网络可以完成任意的 n 维向 m 维的映射。在多数的实际应用中一般都取三层的 BP 神经网络来解决问题,所以本模型选取了三层的神经网络。

2.2　输入和输出层神经元个数

BP 神经网络的输入和输出层的神经元个数完全根据使用者的要求进行设计。影响船舶港口作业时间的因素有很多,通过分析确定了船舶类型、岸桥数量、卸 20 尺箱量、装 20 尺箱量、卸 40 尺箱量、装 40 尺箱量、卸特种箱量、装特种箱量和天气作为模型的输入。本模型根据输入样本的维度,将输入层设置为 9 个神经元。集装箱船港口作业时间预测模型最后输出的是集装箱船港口作业时间,所以输出层神经元的个数为 1。

2.3　隐含层神经元个数

隐含层对于整个神经网络的精度起着至关重要的作用,隐含层神经元个数的选取与输入、输出层神经元个数都有直接的关系。神经网络精度的提高可以通过增加隐含层神经元的个数来实现,隐含层神经元的个数较少时,神经网络就不能较好地学习,导致模型不能较好地拟合数据,训练精度和预测精度都不高,出现欠拟合的情况;隐含层神经元个数较多时,模型结构可能过于复杂,导致过度的拟合训练数据,而对测试数据的预测能力较差,出现过拟合的情况,所以选取准确的隐含层神经元个数是很重要的。隐含层神经元个数选取的方法很多,结合开发的神经网络生成器,本模型采用式(1)选取隐含层神经元个数[7]。

$$p = \sqrt{m+n} + a \tag{1}$$

式中:p 为隐含层神经元个数;n 为输入层神经元个数;m 为输出层神经元个数;a 为 1 ~ 10 之间的常数。经过试算最终确定隐含层神经元个数为 10。

2.4　激活函数

BP 神经网络的激活函数有多种。其中 Sigmoid 函数对于网络不同输入,将其输出范围控制在(0,1)之间,公式如下:

$$f(x) = \frac{1}{1+e^{-x}} \tag{2}$$

Tanh(双曲正切)激活函数对于不同范围的输入,将其输出值范围控制在(-1,1)之间,公式如下:

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{3}$$

线性激活函数 Purelin 的输入与输出值可取任意值,公式如下:

$$f(x) = x \tag{4}$$

本模型选取 Sigmoid 作为隐含层的激活函数,Purelin 作为输出层的激活函数。

2.5　LM-BP 算法

标准 BP 算法是根据梯度下降法来调整权值的:

$$\Delta w = -\eta g \tag{5}$$

式中:Δw 为权值阈值更新量;η 为学习速率;g 为梯度。

权值沿着与误差相反的方向移动,使得误差函数减小,缺点是神经网络收敛较慢,且学习速率不容易被确定。LM(Levenberg Marquardt)算法是一种利用标准的数值优化技术的快速算法,是梯度下降法和高斯-牛顿法的结合,既有高斯-牛顿法的局部收敛性,又有梯度下降法的全局特性,具有收敛速度快、鲁棒性好的特点。下面对 LM 算法做简要阐述:

设网络的误差函数为:

$$E(\boldsymbol{w}) = \frac{1}{2}\sum_{i=1}^{p} e_i^2(\boldsymbol{w}) \tag{6}$$

式中:$\boldsymbol{w}$ 是权值和阈值组成的向量;p 是样本数;$e_i^2(\boldsymbol{w})$ 是误差的平方。

用 $\boldsymbol{w}^k$ 表示在第 k 次迭代的权值和阈值,迭代完成后的权值和阈值组成的向量为 $\boldsymbol{w}^{k+1}$,$\Delta\boldsymbol{w}$ 是权值和阈值的改变量,则有:

$$\boldsymbol{w}^{k+1} = \boldsymbol{w}^k + \Delta\boldsymbol{w} \tag{7}$$

牛顿法是通过最小二乘法求解误差函数 $E(\boldsymbol{w})$:

$$\Delta\boldsymbol{w} = -[\nabla E^2(\boldsymbol{w})]^{-1}\nabla E(\boldsymbol{w}) \tag{8}$$

式中:$\nabla E^2(\boldsymbol{w})$ 是误差 $E(\boldsymbol{w})$ 的 Hessian 矩阵,$\nabla \boldsymbol{E}(\boldsymbol{w})$ 表示梯度,对 Hessian 矩阵进行近似计算,可以表明:

$$\nabla E(w) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{w})e(\boldsymbol{w}) \tag{9}$$

$$\nabla E^2(w) = \boldsymbol{J}^{\mathrm{T}}(\boldsymbol{w})e(\boldsymbol{w}) + \boldsymbol{S}(\boldsymbol{w}) \tag{10}$$

式中:$\boldsymbol{J}(\boldsymbol{w})$ 是 $e(\boldsymbol{w})$ 的 Jacobian 矩阵;$\boldsymbol{S}(\boldsymbol{w})$ 是误差矩阵。

$$\boldsymbol{J}(\boldsymbol{w}) = \begin{bmatrix} \dfrac{\partial e_1(\boldsymbol{w})}{\partial w_1} & \dfrac{\partial e_1(\boldsymbol{w})}{\partial w_2} & \cdots & \dfrac{\partial e_1(\boldsymbol{w})}{\partial w_n} \\ \dfrac{\partial e_2(\boldsymbol{w})}{\partial w_1} & \dfrac{\partial e_2(\boldsymbol{w})}{\partial w_2} & \cdots & \dfrac{\partial e_2(\boldsymbol{w})}{\partial w_n} \\ \vdots & \vdots & & \vdots \\ \dfrac{\partial e_N(\boldsymbol{w})}{\partial w_1} & \dfrac{\partial e_N(\boldsymbol{w})}{\partial w_2} & \cdots & \dfrac{\partial e_N(\boldsymbol{w})}{\partial w_n} \end{bmatrix}$$

在靠近极值点时 $\boldsymbol{S}(\boldsymbol{w}) \approx 0$,牛顿法可以修正为高斯-牛顿法,经过改进得到修正权值阈值的公式:

$$\Delta\boldsymbol{w} = -[\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{w})\boldsymbol{J}(\boldsymbol{w})]^{-1}\boldsymbol{J}(\boldsymbol{w})e(\boldsymbol{w}) \tag{11}$$

LM 算法将高斯-牛顿法经过改进得到修正权值阈值的公式:

$$\Delta\boldsymbol{w} = [\boldsymbol{J}^{\mathrm{T}}(\boldsymbol{w})\boldsymbol{J}(\boldsymbol{w}) + \mu\boldsymbol{I}]^{-1}\boldsymbol{J}(\boldsymbol{w})e(\boldsymbol{w}) \tag{12}$$

式中:$\boldsymbol{I}$ 为单位矩阵;μ 为大于 0 的常数。

系数 μ 的值很小时,LM 算法就近似等于高斯-牛顿法,当 μ 的值很大时,就近似等于梯度下降法。每迭代成功一次 μ 就会除以比例系数 $\beta(\beta > 1)$,这样在接近目标误差的时候就基本与高斯-牛顿法相等,计算速度快,精确度也高,否则 μ 乘比例系数 β,LM 算法利用近似二阶导数信息,比梯度下降法快得多。在实际应用中 μ 是一个试探性的参数,对于一个给定值,如果求得 $\Delta\boldsymbol{w}$ 能使 $E(\boldsymbol{w})$ 降低,则 μ 降低,反之 μ 增加。

2.6 相关参数

(1) 权值和阈值。选取处于(-1,1)之间的随机数作为权值和阈值的初始值。

(2) 学习速率。神经网络权值每次的变化量都取决于学习速率的大小,如果学习速率选取较大,系统可能因此而动荡不稳定;学习速率选取较小则收敛速度慢,训练时间长,网络误差值与误差最小值更趋近的目标无法保障。实际应用中常选取较小学习速率给系统提供稳定性保障,所以学习速率的选取区间是[0.01, 0.9]。本模型选取的学习速率为 0.01。

(3) 其他。最大训练次数为 1 000 次,训练要求精度为 0.000 1,极小值认定次数为 50,μ 的初始值为 0.000 01,比例系数 β 为 10。

3 算例分析

某港口近年集装箱船港口作业数据中包含船舶类型、分配岸桥数、装卸集装箱量、天气、计划作业时间、实际作业时间等。集装箱船的第一代和第六代相当少,这里只考虑第二至第五代的船型。因为原始数据中船舶类型和天气都是文字形式,所以需要对船舶类型和天气情况进行编码数字化。船舶类型和天气情况编码后的结果如表 1 和表 2 所示。

表 1 编码后的船舶类型

船类型	船长/m	船宽/m	吃水深度/m	载箱量/TEU	编码
二代	175 ~ 225	25 ~ 30	9.5 ~ 10.5	1 000 ~ 2 000	1
三代	240 ~ 275	30 ~ 32	10.5 ~ 12.0	2 000 ~ 3 000	2
四代	275 ~ 295	32 ~ 35	11.5 ~ 12.5	3 000 ~ 4 500	3
五代	280 ~ 300	32.2 ~ 39.4	11.5 ~ 13.5	4 500 ~ 6 000	4

表 2　编码后的天气

风速	天气	编码
1 ~ 6 级	晴天、多云、阴天	1
	小雨	2
	中雨	3
	小雪	4
	中雪	5
6 ~ 8 级	晴天、多云、阴天	6
	小雨	7
	中雨	8
	小雪	9
	中雪	10

随机选取 1 000 条集装箱船港口作业信息数据作为模型的训练集和测试集数据,由于数据量较大,这里只展示部分编码后的数据,如表 3 所示。

表 3　编码后的船舶信息

船舶类型	岸桥数	卸 20 尺箱量	装 20 尺箱量	卸 40 尺箱量	装 40 尺箱量	卸特种箱量	装特种箱量	天气	作业时间/h
2	3	241	298	88	122	45	33	9	9
3	2	238	175	95	99	0	115	4	12
4	2	229	74	129	61	47	73	2	10
4	2	133	118	98	142	85	47	2	10
2	2	87	185	219	189	93	28	9	13.5
3	3	273	126	229	101	26	79	6	9.5
3	3	173	159	90	78	112	140	4	8.3

1) 传统的集装箱船装卸集装箱量除以岸桥装卸效率的预测方法可以根据式(13)计算:

$$h = \frac{Y}{v \times n} \tag{13}$$

式中:h 为集装箱船港口作业时间;Y 为装卸集装箱总量;v 为单个岸桥装卸效率;n 为分配岸桥数。

港口单个岸桥装卸效率为 35 箱/h,用传统预测方法计算集装箱船港口作业时间结果如表 4 所示。

表 4　传统方法预测作业时间

装卸集装箱量	分配岸桥数	预计作业时间/h
827	3	7.9
722	2	10.3
613	2	8.8
623	2	8.9
801	2	11.4
834	3	7.9
752	3	7.2

2) 用神经网络模型预测集装箱船港口作业时间:为了消除各参数由于单位等的影响,并且样本不一定包含极大和极小值,所以对数据用式(14)做规范化处理,使规范化后的数据范围为 [-1,1]。

$$y = \frac{(y_{\max} - y_{\min}) \times (x - x_{\min})}{x_{\max} - x_{\min}} + y_{\min} \tag{14}$$

式中:y 为规范化后的数值;$y_{\max} = 1$;$y_{\min} = -1$;$x_{\max}$ 为每一属性中的最大值;$x_{\min}$ 为每一属性中的最小值;x 为需要规范化的数据值。

模型训练学习过程如下:选取样本中的 3 条数据如表 5 所示,规范化后的样本数据如表 6 所示,模型的输入为 $x_1, x_2, \cdots, x_9$,输出为 y。

表 5　样本数据

船舶类型	岸桥数	卸 20 尺箱量	装 20 尺箱量	卸 40 尺箱量	装 40 尺箱量	卸特种箱量	装特种箱量	天气	作业时间/h
3	2	62	137	26	92	146	118	3	10
2	2	176	101	85	93	99	128	7	12
4	3	114	101	114	107	25	142	2	6.5

表 6　规范化后的样本数据

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y
0	-1	-1	1	-1	-1	1	-1	-0.6	0.27
-1	-1	1	-1	0.34	-0.87	0.22	-0.17	1	1
1	1	-0.09	-1	1	1	-1	1	-1	-1

(1) 设定好网络的期望误差值 $\varepsilon = 0.000\,1$,系数 $\beta = 10$,$\mu = 0.1$,学习速率 $\eta = 0.01$,以及权值和阈值的向量:

$$\boldsymbol{w}^k = \begin{bmatrix} 0.84,-0.86,0.38,0.51,-0.18,-0.60,0.67, \\ 0.86,0.56,0.75,0.87,-0.86,0.47,-0.21, \\ 0.34,-0.46,-0.45,-0.75,-0.83,0.99,0.78, \\ -0.90,0.60,0.35,0.28,0.70,0.85,0.82, \\ -0.03,0.87,-0.79,0.59,-0.67,0.39,-0.50, \\ 0.37,0.82,-0.03,0.87,-0.79,0.59,-0.67, \\ 0.39,-0.50,0.37,0.25,0.59,0.36,0.69, \\ 0.61,-0.75,0.77,0.85,-0.29,-0.86,-0.76, \\ 0.54,0.69,-0.05,-0.03,0.06,0.36,0.85, \\ -0.63,-0.24,0.69,0.56,-0.12,0.29,-0.10, \\ 0.80,0.64,0.03,0.05,-0.20,-0.41,0.49, \\ -0.34,-0.85,0.59,0.32,0.72,0.67,0.31, \\ 0.97,0.43,0.16,0.75,0.23,0.72,0.87, \\ 0.86,-0.61,-0.87,0.50,-0.51,0.47,-0.03, \\ 0.30,0.68,0.68,0.72,-0.93,0.20,-0.56, \\ 0.66,-0.17,0.82,-0.64,0.58 \end{bmatrix}$$

（2）计算神经网络的输出 $\hat{y} = 0.437\ 7$ 。

（3）计算 *Jacobian* 矩阵：

$$\boldsymbol{J}(\boldsymbol{w}) = \begin{bmatrix} -0.005\ 28,0.005\ 40,-0.002\ 39,-0.003\ 20, \\ 0.001\ 13,0.003\ 77,0.003\ 77,-0.004\ 21, \\ -0.005\ 40,-0.003\ 52,-0.006\ 62,-0.007\ 68, \\ 0.007\ 59,-0.004\ 15,0.001\ 85,-0.003\ 00, \\ -0.003\ 00,0.004\ 06,0.003\ 97,0.006\ 62 \\ -0.002\ 06,0.002\ 46,0.001\ 94,-0.002\ 23, \\ 0.001\ 49,0.000\ 87,0.000\ 87,0.000\ 69, \\ 0.001\ 74,0.002\ 11,-0.001\ 19,0.000\ 04, \\ -0.001\ 26,0.001\ 15,-0.000\ 86,0.000\ 97, \\ 0.000\ 97,-0.000\ 57,0.000\ 73,-0.000\ 54, \\ 0.000\ 47,0.001\ 12,0.000\ 68,0.001\ 31, \\ 0.001\ 16,-0.001\ 42,-0.001\ 42,0.001\ 46, \\ 0.001\ 61,-0.000\ 55,0.003\ 33,0.002\ 94, \\ -0.002\ 09,-0.002\ 67,0.000\ 19,0.000\ 12, \\ 0.000\ 12,-0.000\ 23,-0.001\ 39,-0.003\ 29 \\ 0.000\ 98,0.000\ 37,-0.001\ 07,-0.000\ 87, \\ 0.000\ 19,-0.000\ 45,-0.000\ 45,0.000\ 16, \\ -0.001\ 24,-0.000\ 99,-0.000\ 07,-0.000\ 11, \\ 0.000\ 45,0.000\ 92,-0.001\ 10,0.000\ 77, \\ 0.000\ 77,0.001\ 91,-0.001\ 33,-0.000\ 72 \\ 0.001\ 04,0.000\ 97,0.000\ 45,0.001\ 40, \\ 0.000\ 62,0.000\ 23,0.000\ 23,0.001\ 08, \\ 0.000\ 33,0.001\ 04,-0.000\ 85,-0.000\ 84, \\ 0.000\ 60,0.000\ 85,-0.000\ 49,0.000\ 50, \\ 0.000\ 50,-0.000\ 46,0.000\ 03,-0.000\ 29, \\ -0.027\ 19,-0.035\ 43,0.029\ 66,-0.005\ 77, \\ 0.032\ 14,-0.017\ 72,-0.014\ 83,-0.034\ 61, \\ 0.023\ 07,-0.016\ 89,0.023\ 89 \end{bmatrix}$$

（4）通过式(12)、式(6)计算出 $\Delta\boldsymbol{w}$：

$$\Delta\boldsymbol{w} = \begin{bmatrix} 0,0.000\ 06,0.000\ 06,-0.000\ 06,0.000\ 06,0.000\ 06, \\ -0.000\ 06,0.000\ 06,0.000\ 04,0,0.000\ 09,0.000\ 09, \\ -0.000\ 09,0.000\ 09,0.000\ 09,-0.000\ 09,0.000\ 09, \\ 0.000\ 05,0,-0.000\ 02,-0.000\ 02,0.000\ 02,-0.000\ 02, \\ -0.000\ 02,0.000\ 02,-0.000\ 02,-0.000\ 01,0.000\ 01, \\ 0.000\ 01,-0.000\ 01,0.000\ 01,0.000\ 01,-0.000\ 01, \\ 0.000\ 01,0.000\ 01,0,-0.000\ 02,-0.000\ 02,0.000\ 02, \\ -0.000\ 02,-0.000\ 02,0.000\ 02,-0.000\ 02,-0.000\ 01 \\ 0,0.000\ 04,0.000\ 04,-0.000\ 04,0.000\ 04,0.000\ 04, \\ -0.000\ 04,0.000\ 04,0.000\ 02,0,0.000\ 02,0.000\ 02, \\ -0.000\ 02,0.000\ 02,0.000\ 02,-0.000\ 02,0.000\ 02, \\ 0.000\ 01,0,0.000\ 02,0.000\ 02,-0.000\ 02,0.000\ 02, \\ 0.000\ 02,-0.000\ 02,0.000\ 02,0.000\ 01,0,-0.000\ 01, \\ -0.000\ 01,0.000\ 01,-0.000\ 01,-0.000\ 01,0.000\ 01, \\ -0.000\ 01,-0.000\ 01,0,0.000\ 01,0.000\ 01,-0.000\ 01, \\ 0.000\ 01,0.000\ 01,-0.000\ 01,0.000\ 01,0.000\ 01, \\ -0.000\ 15,-0.000\ 20,-0.000\ 04,-0.000\ 20, \\ -0.000\ 03,-0.000\ 13,-0.000\ 05,-0.000\ 03, \\ -0.000\ 03,-0.000\ 03,-0.000\ 06,-0.000\ 09, \\ 0.000\ 02,-0.000\ 01,0.000\ 02,-0.000\ 04,-0.000\ 02, \\ -0.000\ 02,0.000\ 01,-0.000\ 01,0.000\ 41 \end{bmatrix}$$

$$E(\boldsymbol{w}^k) = 0.014\ 06$$

（5）如果 $E(\boldsymbol{w}^k) < \varepsilon$ ，转到式(10)，否则，用 $\boldsymbol{w}^{k+1}$ 为权值和阈值计算误差 $E(\boldsymbol{w}^{k+1}) = 0.011\ 37$ 。

$$\boldsymbol{w}^{k+1} = \begin{bmatrix} 0.840\ 00,-0.859\ 94,0.380\ 06,0.509\ 94,-0.179\ 94, \\ -0.599\ 94,0.669\ 94,0.860\ 06,0.560\ 04,0.750\ 00, \\ 0.870\ 09,-0.859\ 91,0.469\ 91,-0.209\ 91,0.340\ 09, \\ -0.460\ 09,-0.449\ 91,-0.749\ 95,-0.830\ 00,0.989\ 98, \\ 0.779\ 98,-0.899\ 98,0.599\ 98,0.349\ 98,0.280\ 02, \\ 0.699\ 98,0.849\ 99,0.820\ 00,-0.029\ 99,0.870\ 01, \\ -0.790\ 01,0.590\ 01,-0.669\ 99,0.389\ 99,-0.499\ 99, \\ 0.370\ 01,0.250\ 00,0.589\ 98,0.359\ 98,0.690\ 02, \\ 0.609\ 98,-0.750\ 02,0.770\ 02,0.849\ 98,-0.290\ 01, \\ -0.860\ 00,-0.759\ 96,0.540\ 04,0.689\ 96,-0.049\ 96, \\ -0.029\ 96,0.059\ 96,0.360\ 04,0.850\ 02,-0.630\ 00, \\ -0.239\ 98,0.690\ 02,0.559\ 98,-0.119\ 98,0.290\ 02, \\ -0.100\ 02,0.800\ 02,0.640\ 01,0.030\ 00,0.050\ 02, \\ -0.199\ 98,-0.410\ 02,0.490\ 02,-0.339\ 98,-0.850\ 02, \\ 0.590\ 02,0.320\ 01,0.720\ 00,0.669\ 99,0.309\ 99, \\ 0.970\ 01,0.429\ 99,0.159\ 99,0.750\ 01,0.229\ 99, \\ 0.719\ 99,0.870\ 00,0.860\ 01,-0.609\ 99,-0.870\ 01, \\ 0.500\ 01,-0.509\ 99,0.469\ 99,-0.029\ 99,0.300\ 01, \\ 0.659\ 85,0.859\ 80,-0.720\ 04,0.139\ 80,-0.780\ 03, \\ 0.429\ 87,0.359\ 95,0.839\ 97,-0.560\ 03,0.409\ 97, \\ 0.679\ 94,0.679\ 91,0.720\ 02,-0.930\ 01,0.200\ 02, \\ -0.560\ 04,0.659\ 98,-0.170\ 02,0.820\ 01,-0.640\ 01, \\ 0.579\ 59 \end{bmatrix}$$

(6) 如果 $E(\boldsymbol{w}^{k+1}) < E(\boldsymbol{w}^k)$,令 $k = k+1$,$\mu = \mu/\beta$,回到(5),否则本次迭代不更新权值和阈值,令 $\boldsymbol{w}^{k+1} = \boldsymbol{w}^k$,$\mu = \mu\beta$,回到(6)。

(7) 结束。

选取数据的 70% 作为模型的训练数据,数据的 30% 作为模型的验证数据,运用 MATLAB 平台进行集装箱船港口作业时间预测模型实验[8]。通过神经网络模型的拟合效果,神经网络模型和传统方法预测值的绝对百分比误差(Absolute Percentage Error, APE)、平均绝对百分比误差(Mean Absolute Percentage Error, MAPE)的对比,评价模型的优越性。

$$APE = \frac{|\hat{y} - y|}{y} \times 100\% \tag{15}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{y} - y|}{y} \times 100\% \tag{16}$$

经过多次训练学习取得最好一次结果。从图 1 可以看出,随着神经网络迭代次数的增加,网络的误差逐渐变小并且趋于稳定。从神经网络收敛效果看,当网络训练的次数达到 87 次后网络收敛,网络的均方误差为 0.000 27,基本达到误差的设置要求,模型训练效果优秀。

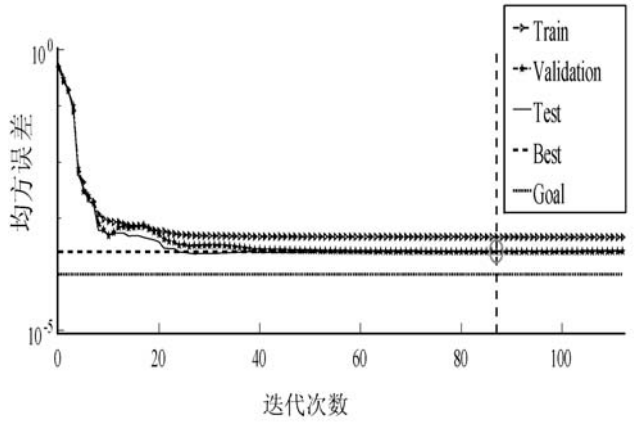

图 1　神经网络迭代图

图 2 横坐标为模型输出的期望目标值,纵坐标为模型对数据拟合实际输出的目标值,当模型实际输出的值满足在期望值上下 0.001($output \approx 1 \times Target \pm 0.001$) 误差的范围内,则说明该数据可以被模型解释。$R$ 为被模型解释的数据量占总数据量的比例,*Fit* 实线表示网络对数据训练学习将非线性关系转化成的线性关系,$Y = T$ 虚线代表模型拟合数据的期望线性关系。可以看出,整个网络的数据拟合度达到 99.92%,通过网络对数据的拟合效果可以验证该网络模型的可行性。

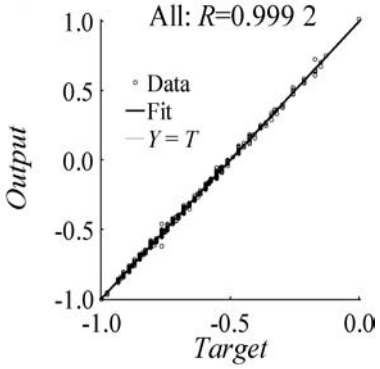

图 2　神经网络迭代图

根据式(13)计算测试集的集装箱船港口作业时间,根据式(15)和式(16)分别计算 APE 和 MAPE。

通过神经网络预测模型输出测试集的集装箱船港口作业时间预测值,根据式(15)和式(16)分别计算 APE 和 MAPE。

两种预测方法对比如下:

从图 3 中可以看出,神经网络预测集装箱船港口作业时间的 APE 均明显低于传统方法预测集装箱船港口作业时间的 APE。从表 7 中可以看出,传统预测方法的最大 APE 是 10.3%,MAPE 是 4.5%;神经网络模型的最大 APE 是 7.1%,MAPE 是 1.6%,这说明了用神经网络预测集装箱船港口作业时间比传统方法更精确。

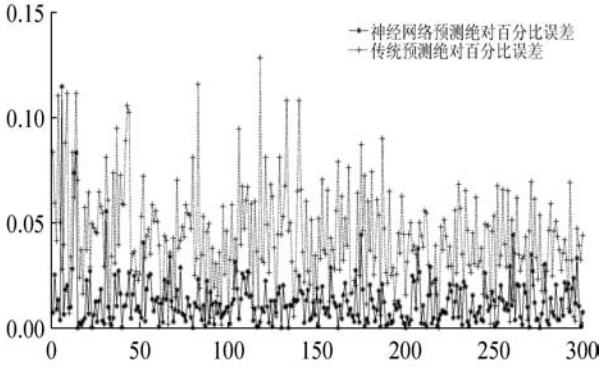

图 3　神经网络和传统方法预测值的 APE

表 7　测试集最大 APE 和 MAPE　　%

预测方法	最大 APE	MAPE
传统方法	10.3	4.5
神经网络	7.1	1.6

4　结　语

本文基于传统预测集装箱船港口作业时间方法的预测精度低、不灵活等问题,分析了集装箱船港口作业

时间受多种因素的影响、存在非线性的特点，结合神经网络非线性拟合能力强的特点，构建神经网络集装箱船港口作业时间模型。通过与传统预测方法预测效果的对比，展现了本文模型的优越性，提高了预测的精度，为制作科学高效的泊位计划奠定了基础。

参 考 文 献

[1] Imai A, Nishimura E, Papadimitriou S. The dynamic berth allocation problem for a container port[J]. Transportation Research Part B: Methodological, 2001, 35(4): 401 - 417.

[2] 张煜，王少梅. 基于遗传算法的泊位连续化动态调度研究[J]. 系统仿真学报，2007，19(10)：2161 - 2164.

[3] 秦进，缪立新，陈长彬，等. 时间窗限制下港口泊位优化分配问题模型[J]. 航海工程，2010，39(2)：142 - 145.

[4] 韩晓龙，赵书杰. 时间窗限制下泊位分配问题的约束规划模型[J]. 辽宁工程技术大学学报(自然科学版)，2014，33(7)：983 - 987.

[5] 曾庆成，赵孝峰，胡祥培，等. 集装箱码头泊位计划的鲁棒优化模型[J]. 运筹与管理，2015，24(2)：71 - 77.

[6] 陈洋. 船舶在港停泊时间的影响因素分析[J]. 珠江水运，2009(11)：70 - 71.

[7] 尚钢，钟珞，陈立耀. 神经网络结构与训练参数选取[J]. 武汉工业大学学报，1997(2)：108 - 110.

[8] 桂现才. BP 神经网络在 MATLAB 上的实现与应用[J]. 湛江师范学院学报，2004，25(3)：79 - 83.

[9] 丁硕，巫庆辉. 基于改进 BP 神经网络的函数逼近性能对比研究[J]. 计算机与现代化，2012(11)：10 - 13，17.

[10] 丁硕，常晓恒，巫庆辉，等. 数值优化改进的 BP 神经网络逼近性能对比研究[J]. 山东科学，2014，27(1)：68 - 72，91.

(上接第 32 页)

4 结 语

本文提出基于改进引力搜索机制的数据聚类算法。定义了引力搜索进化聚类解编码方式，并设计了基于汉明距离的引力搜索粒子距离度量方法，可以有效衡量数据对象在各维度属性上的不同。在粒子速度更新上，引入加速因子到粒子速度更新中，利用最优粒子位置代表的聚类解来加速局部开发过程，加速粒子向最优粒子移动，有效均衡局部开发与全局搜索间的平衡。实验结果证明了算法在降低聚类失误率上的优势。

参 考 文 献

[1] Aggarwal C C, Reddy C K. Data clustering: Algorithms and applications[M]. Chapman & Hall/CRC, 2013.

[2] Noack A, Poll R, Fischer W J, et al. QRS pattern recognition using a simple clustering approach for continuous data[C]// 2013 IEEE XXXIII International Scientific Conference Electronics and Nanotechnology (ELNANO), 2013.

[3] Gan G J. Application of data clustering and machine learning in variable annuity valuation[J]. Insurance Mathematics and Economics, 2013, 53(3): 795 - 801.

[4] Bruse J L, Zuluaga M A, Khushnood A, et al. Detecting clinically meaningful shape clusters in medical image data: Metrics analysis for hierarchical clustering applied to healthy and pathological aortic arches[J]. IEEE Transactions on Biomedical Engineering, 2017, 64(10): 2373 - 2383.

[5] Bo Y. The data clustering based dynamic risk identification of biological immune system: mechanism, method and simulation [J]. Cluster Computing, 2019, 22: 6253 - 6266.

[6] Kanungo T, Mount D M, Netanyahu N S, et al. An efficient k-means clustering algorithm: analysis and implementation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7): 881 - 892.

[7] 张强，李淼. 基于遗传算法和遗传模糊聚类的混合聚类算法[J]. 计算机工程与应用，2007，43(3)：164 - 165，197.

[8] Abdel-Kader R. Genetically improved PSO algorithm for efficient data clustering[C]//Second International Conference on Machine Learning & Computing, 2010.

[9] Saatchi S, Hung C. Hybridization of the ant colony optimization with the k-means algorithm for clustering[C]//Scandinavian Conference on Image Analysis, 2005.

[10] Karaboga D, Ozturk C. A novel clustering approach: Artificial Bee Colony (ABC) algorithm[J]. Applied Soft Computing, 2011, 11(1): 652 - 657.

[11] Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm[M]. Information Sciences: an International Journal, 2009, 179(13): 2232 - 2248.

[12] Dowlatshahi M B, Nezamabadi-Pour H. GGSA: A grouping gravitational search algorithm for data clustering[J]. Engineering Applications of Artificial Intelligence, 2014, 36: 114 - 121.

[13] Yin M, Hu Y, Yang F, et al. A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering[J]. Expert Systems with Applications An International Journal, 2011, 38(8): 9319 - 9324.

[14] 王彩霞. 基于改进引力搜索的混合 K-调和均值聚类算法研究[J]. 计算机应用研究，2016(1)：118 - 121.

[15] Hatamlou A, Abdullah S, Nezamabadi-Pour H. A combined approach for clustering based on K-means and gravitational search algorithms[J]. Swarm and Evolutionary Computation, 2012, 6: 47 - 52.

[16] Han X H, Quan L, Xiong X Y, et al. A novel data clustering algorithm based on modified gravitational search algorithm [J]. Engineering Applications of Artificial Intelligence, 2017, 61: 1 - 7.