面向金融数据的神经网络时间序列预测模型*

张栗粽，王谨平，刘贵松，罗光春，卢国明
（电子科技大学 计算机科学与工程学院，成都 611731）

摘　要：针对 Elman 神经网络模型，通过引入时间权重与随机性因素，提出了改进的 Elman 神经网络模型，提高了现有 Elman 神经网络针对时序数据预测的精度。提出了基于时序数据的特征学习框架，可评估多个特征参数对结果的联合影响。在此基础上，提出了一个互联网金融风险预测模型，实验结果表明，所提出的模型在金融时序预测中具有更好的准确度。

关键词：时间序列；Elman 神经网络；特征选择；特征提取；Clamping 神经网络
中图分类号：TP183　　**文献标志码**：A　　**文章编号**：1001-3695(2018)09-2632-06
doi:10.3969/j.issn.1001-3695.2018.09.017

Neural network time series prediction model for financial data

Zhang Lizong, Wang Jinping, Liu Guisong, Luo Guangchun, Lu Guoming
(*School of Computer Science & Engineering, University of Electronic Science & Technology of China, Chengdu* 611731, *China*)

Abstract: This paper studied the Elman neural network model used in time series data predictions. It proposed an enhanced Elman model with the introduction of time weight and random factor for the improvement of prediction accuracy. In addition, it also proposed a feature selection framework as a part of the enhanced model for time series data training, the framework was able to evaluate the joint effect of multiple features. On this basis, it gave an Internet financial risk prediction model. The result indicates that the model has better accuracies in the predictions of financial time series data.

Key words: time series; Elman neural network; feature selection; feature extraction; Clamping neural network

随着互联网金融的发展，越来越多的互联网金融公司成立，如蚂蚁金服、美团金融、宜人贷等。这些互联网金融公司中如融 360 提供互联网贷款时，根据用户的历史还款时间序列，来对用户信贷进行预测，从而避免贷款给不良用户造成的损失。所以对互联网金融公司而言，根据公司历史的金融时序数据，通过建立互联网金融风险预测模型，挖掘并预测未来对公司有利信息，从而将风险降低到最小，已成为互联网金融公司的重要目标。本文通过提取时序数据的特征并通过时序数据特征选择模型来完成时序数据的特征学习，将选取的特征作为时间序列预测模型的输入，运用时间序列预测模型来完成时间序列分析预测，并将此模型应用在互联网金融公司中，为互联网金融公司在未来预测上提供相应参考信息，从而将风险降低到最小。

时间序列是一组数字序列，每个数据按时间顺序排序。通过对这组数据使用数理统计方法完成时间序列分析，以探索包含在数据中的所有信息，从而达到预测未来事物发展的目的[1]。时间序列分析应用广泛，如风能预测[2]、交通流预测[3]、水位预测[4]，特别是在金融领域上的预测更为广泛[5]。基于神经网络的时间序列分析更适合非线性、数据并行化的预测，所以更多的学者使用神经网络来进行时间序列分析[6]。

1　现有时间序列预测算法

1.1　基于传统的时间序列预测算法

时序数据中传统的预测模型算法是根据已采集的时序数据，然后通过参数估计与曲线拟合来建立预测模型的方法与理论。如基于非线性最小二乘法中主要的模型以 ARMA 模型为主。ARMA 主要有三种基本形式：AR(auto-regressive，自回归模型)、MA(moving-average，滑动平均模型)和 ARMA(auto-regressive moving-average，自回归滑动平均模型)。

1.2　基于人工神经网络的时间序列预测算法

人工神经网络具有良好的自学能力、自适应能力、泛化能力以及很好的非线性映射能力，根据神经元中的非线性函数如 Sigmoid 函数，将这些神经元组织起来就能够重建任意的非线性的时间序列。因此，神经网络很适合处理具有很强的随机性和非线性的时间序列。

人工神经网络在时间序列预测应用广泛，特别是在金融领域。例如 Zhan 等人[7]使用 BP 神经网络应用于股票的时序预测，然后与文中的另外三种预测算法进行对比，BP 神经网络模型对股票时序预测拟合度更好，预测得更精确；Falat 等人[8]将径向基神经网络应用在交换汇率上的预测，取得了对交换汇率更精确的预测；Li 等人[9]使用 Elman 神经网络来预测股票综合指数预测中，预测值与实际值之间的绝对平均误差与最小平方误差两个指标都较小，能更佳地拟合股票综合指数序列的实际值。神经网络在时间序列预测上主要分为以下几类：

a)基于 BP 预测模型。BP 神经网络是使用误差反向传播算法作为训练算法的多层前馈网络，于 1986 年由 Rumelhart 和 McCelland 提出，是前馈神经网络最常使用的一种神经网络模型。将 BP 神经网络模型应用在时序预测中，具有计算简单、容错性好的优点。

b)基于 RBF 预测模型。RBF 神经网络是基于径向基函

收稿日期：2017-04-18；**修回日期**：2017-05-31　　**基金项目**：四川省科技厅应用基础资助项目(2017JY0007,2017JY0037,2018JY0073)；国际合作项目(2018HH0075)；省院省校合作项目(2017JZ0031)；海外留学回国人员科研启动费基金资助项目

作者简介：张栗粽(1981-)，男，黑龙江人，副教授，博士，主要研究方向为大数据、知识工程、人工智能(l.zhang@uestc.edu.cn)；王谨平(1990-)，男，四川人，硕士，主要研究方向为大数据；刘贵松(1973-)，男，山东人，教授，博士，主要研究方向为云计算、大数据；罗光春(1974-)，男，四川人，教授，博士，主要研究方向为机器学习、大数据；卢国明(1976-)，男，四川人，副教授，硕士，主要研究方向为云计算、大数据.

数,在前馈网络中属于较优的网络,理论上它能够逼近任何一个非线性函数,这在时间序列预测中能尽可能地逼近原始时序数据,对原始时序数据的学习相当于在多维时序空间中寻找训练的时序数据最佳拟合平面。

c)基于 Elman 预测模型。Elman 神经网络预测模型是属于前向反馈神经网络的一类。不同于 BP 与 RBF 神经网络,Elman 神经网络的输入到输出具有反馈功能,即多了一层连接层神经元。通过连接层的神经元能够存储隐含层的输出状态,使得网络具有映射动态特征的功能,从而能使 Elman 神经网络能更好地预测时间序列并具有适应时变特征的性能。

2　现有的 Elman 预测模型

2.1　模型结构

Elman 神经网络由输入层、隐含层、连接层和输出层组成。数据经过输入层并与连接层一起作为隐含层的输入[10]。Elman 神经网络一般为一个输入层、一个隐含层、一个连接层与一个输出层。所以 Elman 神经网络模型中的隐含层数和连接层数可设置为 1。Elman 模型如图 1 所示。

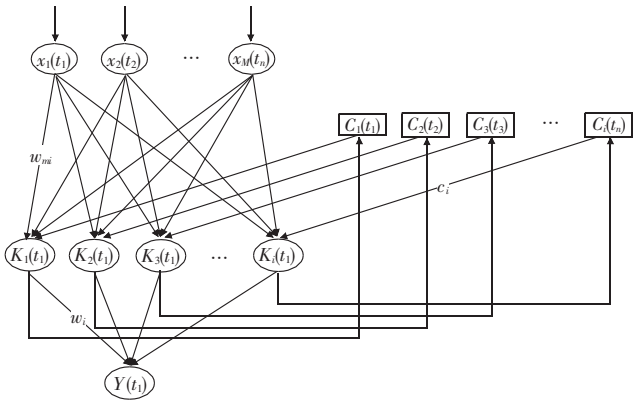

图 1　Elman 模型

2.2　各层节点数

a)输入层节点数。输入层节点数可根据时序训练数据的维数来确定。应用 Elman 神经网络模型进行时序数据预测时,应根据实际应用的场景来确定各层的节点数。所以时序训练数据的表示方式会影响输入层神经元个数。例如,时间序列中使用前三个时序数据 t_1、t_2、t_3 来预测第四个时序数据 t_4 时,则输入层神经元的个数为 3;而如果使用前四个时序数据 t_1、t_2、t_3、t_4 来预测第四个时序数据 t_5 时,则输入层神经元的个数为 4。

b)隐含层节点数。Elman 神经网络隐含层数一般设置为 1。这一层节点数对 Elman 预测影响较大。节点数越多则可以更好地对时序数据进行预测但同时也会使训练时间过长。隐含层中的节点数可由以下方式进行确定:

(a)$\sum_{i=0}^{n} C_M^i > k$。其中 M 为 Elman 神经网络隐含层节点的个数,k 为时序数据训练样本的数目,n 为训练样本的输入维数。当 $i > M$ 时设置 $C_M^i = 0$。

(b)$M = \sqrt{n+m} + a$。其中 m 是时序数据预测输出维数,一般设置为 1;n 为时序数据训练样本的维数;a 可设置为[0,10]的任意整数值。

(c)$M = \log_2 n$,n 为时序数据预测输出的向量维数,同(b)一般设置为 1。

c)输出层节点数。在时间序列分析中,一般是以 t_1,t_2,t_3,…,t_n 来预测 t_{n+1},所以输出层的神经元个数设置为 1 即可。

2.3　现有 Elman 神经网络预测算法的缺陷

a)时间序列中的每个数据对当前预测值的贡献是不同的,距离当前预测值越近的时刻,其产生的贡献也就越多。因此,每个历史数据产生的作用应与其对应的发生时间点作为依据,即不同时刻的历史数据有对应的权重。可以在 Elman 训练算法时,根据时序数据与当前训练数据的远近程度赋予相应的权重。

b)在进行时序数据预测中,只使用原始时序数据作为网络的输入特征来进行预测,而未对原始时序特征进行处理,如对原始时序数据进行特征提取与特征选择。在特征选择中 Clamping 网络只考虑到了单一特征参数对结果的影响,但没有考虑到某些自身对结果影响很小,而与其他特征组合时却对结果有较大的影响;另外 Clamping 网络未对冗余特征进行处理。

3　改进的时间序列预测模型

3.1　理论依据

1)改进 Clamping 网络特征选择理论依据

通过时序数据的特征提取后,会得到很多的特征集,典型的方法是通过 Clamping 网络确认这些特征集里面哪一些是真正重要的特征参数[12]。Clamping 网络主要用于特征选择上,将时序数据通过特征提取后得到特征备用集合,然后将这些备用特征依次通过 Clamping 网络,根据网络的输出误差可以得到一个排序的特征集合,即对当前系统越重要的特征,其排名的次序就越靠前。但是,Clamping 网络只考虑到了单一特征参数对结果的影响,但没有考虑到某些自身对结果影响很小,而与其他特征组合时却对结果有较大的影响;另外 Clamping 网络未对冗余特征进行处理。

2)改进 Elman 模型的理论基础

时间序列中的每个数据对当前预测值的贡献是不同的,距离当前预测值越近的时刻,其产生的贡献也就越多。因此,每个历史数据产生的作用应与其对应的发生时间点作为依据,即不同时刻的历史数据有对应的权重。另外,可以给 Elman 神经网络模型在处理原有的数据序列时,添加一部分的随机性[11],即作为数据预测中的不确定随机因素。因此,本文将引入一个时间相关强度函数 $\Phi(t_n)$,其定义如下:

$$\Phi(t_n) = \frac{1}{\alpha} e_{t_n}^{\int_{t_n}^{t_a} \rho(t)\,dt + \int_{t_n}^{t_a} \theta(t)\,dG(t)}$$

其中:α 是大于 0 的时间强度系数;t_n 是数据集中最新的值;t_a 是数据集中的随机值;$\rho(t)$ 是漂移函数;$\theta(t)$ 是波动率函数;$G(t)$ 是一个随机过程。

a)$\rho(t)$ 的定义如下,其中 c 为样本的个数。

$$\rho(t) = \frac{1}{(c+t)^2}$$

b)$\theta(t)$ 的定义式如下,其中 $\bar{x}$ 是样本的均值。

$$\theta(t) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}$$

c)$G(t)$ 是一个随机过程且 $G(t)$ 满足下列条件:

(a)$G(0) = 0$;

(b)$E[G(t)] = 0$;

(c)具有平衡独立增量;

(d)$t > 0$,$G(t) \sim N(0, \sigma^2 t)$ $(\sigma > 0)$。

随机过程 $G(t)$ 定义如下:设 $(\Omega, \mathcal{F}, P)$ 是一个概率空间,T 是一个参数集 $(T \in R)$,$X(t,w)$,$t \in T$,$w \in \Omega$ 是 $T \times \Omega$ 上的二元函数。如果对于每一个 $t \in T$,$X(t,w)$,$w \in \Omega$ 是 $(\Omega, \mathcal{F}, P)$ 上的随机变量,则称随机变量族 $\{X(t,w), t \in T\}$ 为定义在 $(\Omega, \mathcal{F}, P)$ 上的随机过程,可记为 $\{X(t), t \in T\}$,其中 t 为参数,T 为参数集。

3.2　改进的 Clamping 网络特征选择算法

Clamping 网络只考虑到了单一特征参数对结果的影响,但

没有考虑到某些自身对结果影响很小,而与其他特征组合时却对结果有较大的影响;另外 Clamping 网络未对冗余特征进行处理。因此本节提出一种改进的 Clamping 网络(即 DS-Clamping)。首先通过 Clamping 网络将特征对结果的影响进行排序(ranking),再通过排序顺序将特征依次增加到测试特征集中,测验组合对结果准确度的影响。每增加一种特征参数,就使用此特征集对分类神经网络进行训练与测试,如果精确度提升超过一定阈值,那么就将其保留,如果精确度下降超过一定阈值,那么就将其剔除。如果上升下降均未超过阈值,那么将其放回待选列表的末尾。这样,通过阈值的设定,可以控制特征集的大小。而最终获得的特征集,即为所需要的特征组合。测试的神经网络可以使用多层感知器神经网络,需要通过多次实验来确定隐含层节点数及训练次数。具体的步骤如下所示。

算法 1　时序数据特征选择算法

定义:

x_i:特征集中的第 i 个特征,其中 $1 \leqslant i \leqslant n$。

x_{ip}:第 p 个用例中的第 i 个特征,其中 $1 \leqslant p \leqslant P$。

$g(x)$:网络的总体误差。

$g(x|_{x_i = \mathrm{mean}(x_i)})$:当 $x_i = \mathrm{mean}(x_i)$ 时的网络误差。

算法步骤:

1)使用所有特征参数训练网络;

2)计算网络的 $g(x)$;

3)对于使用的每一个特征执行步骤 3.1)~3.5);

3.1)设置网络的输入为所有特征参数;

3.2)设置当前特征的值为均值;

3.3)测试整个网络;

3.4)计算此时的 $g(x|_{x_i = \mathrm{mean}(x_i)})$

3.5)计算此特征对网络表现的影响 $\xi(x_i)$ 如下所示:

$$\xi(x_i) = 1 - \frac{g(x|_{x_i = \mathrm{mean}(x_i)})}{g(x)}$$

3.6)重复执行步骤 3),直至所有特征参数计算完成;

4)根据 $\xi(x_i)$,按照升序对所有特征参数进行排序;

5)创建一个空集合;

6)对于排序中的每一特征执行步骤 6.1)~6.4);

6.1)将该特征加入集合;

6.2)使用当前集合对网络进行训练;

6.3)测试当前网络,并计算精确度;

6.4)如果精确度提升超过一定阈值,那么就将其保留,如果精确度下降超过一定阈值,那么就将其剔除。如果上升下降均未超过阈值,那么将其放回待选列表的末尾;

6.5)如果排在末尾的特征在第二次测试时,并没有使网络的精确度超过一个阈值,将其剔除;

6.6)重复步骤 6)直至特征排序列表为空。

7)集合中的特征即为所需特征。

通过步骤 1)~3)得到 Clamping 网络的特征排序,然后通过步骤 4)~7)从特征排序中挑选出最终的特征集合,挑选的过程中判断每个将要加入特征集合的特征是否对网络的误差影响超过阈值,如果有超过阈值则添加进特征集合,如果没有超过则放到待挑选特征的末尾。第二次再次遇到这个特征时,如果对网络的误差影响还是没超过指定的阈值,则将该特征从特征排序列表中剔除,该过程直到特征排序列表为空时停止。

3.3　改进的 Elman 神经网络训练算法

改进的 Elman 神经网络训练算法包含网络信号的正向传播,根据误差反向传播来训练各层权值,为了方便本节推导改进的 Elman 训练算法,现作如下定义:

M:时序训练数据的特征个数,即 Elman 输入层神经元节点数。

N:Elman 隐含层神经元节点数。

Y:时序数据预测的输出向量维数,即输出层神经元节点个数,本模型中输出神经元个数为 1。

x_m:训练中的第 m 个特征,即 Elman 输入层中的第 m 个神经元。

k_i:Elman 隐含层第 i 个神经元。

y:Elman 输出层神经元。

w_{mn}:Elman 中从输入层的第 m 个节点与隐含层中第 i 个节点的权值。

w_i:Elman 隐含层中第 i 个节点与输出节点的权值。

c_n:Elman 隐含层到连接层的权值。

$f(\cdot)$:隐含层传递函数(如 sigmoid 函数)。

$g(\cdot)$:输出层传递函数(如线性函数 purline 函数)

$u(t_n)$:t_n 时刻网络层中的输入,如 $u_I^i(t_n)$ 表示第 I 层的第 i 个神经元的输入。

$v(t_n)$:t_n 时刻网络层中的输出,如 $v_I^i(t_n)$ 表示第 I 层的第 i 个神经元的输出。

通过最速下降法调整各层权值,以单隐含层 Elman 神经网络推导 Elman 预测模型的训练算法。网络模型输入 M 维的训练样本,并在输出层输出维数为 1 的预测值。第 n 个样本的时刻用 t_n 表示,用 $u(t_n)$ 和 $v(t_n)$ 分别代表 Elman 网络各层在 t_n 时刻的输入值和输出值,如 $u_K^1(t_n)$ 表示 K 层,即 Elman 隐含层第 1 个神经元节点在 t_n 时刻所接收到的网络输入。$u_Y(t_n)$ 表示 Y 层的输入,即输出层在 t_n 时刻的输入。$v_Y(t_n)$ 表示 Y 层的输出,即输出层在 t_n 时刻的输出。$u_C^1(t_n)$ 表示 Elman 的连接层,即连接层第 1 个神经元在 t_n 时刻的输出。网络的实际输出如下:

$$Y(t_n) = [v_Y(t_n)]$$

网络的期望输出为 $d(t_n)$。

第 t_n 时刻迭代的误差信号定义如下:

$$e_Y(t_n) = d(t_n) - Y(t_n)$$

$$e(t_n) = \frac{1}{2}(e_Y(t_n))^2$$

下面对改进的 Elman 神经网络训练算法的具体步骤进行详细说明。首先创建 GT-Elman 神经网络模型,根据时序数据的维数确定输入层神经元个数;输出层的节点数根据时序数据输出值的维数确定,本文中所讨论的输出均是时序数据的预测输出值,因为输出神经元的个数为 1;根据输入层神经元个数、输出层神经元个数和输入样本个数确定节点数;连接层节点数同隐含层节点数相同。设置好 GT-Elman 神经网络各层的神经元个数后,创建网络拓扑,然后初始化输入层到隐含层的权重、隐含层到输出层权重、隐含层到连接层的权重。这些权重的初始设置可以直接使用[0,1]中的随机值或是采用遗传算法来优化 GT-Elman 初始权重。最后设置网络训练算法的学习率 η,最小误差 ξ 与迭代次数 p。改进的 Elman 神经网络训练算法主要分为隐含层到输出层、输入层到隐含层、连接层到隐含层三部分,总训练算法流程如图 2 所示。

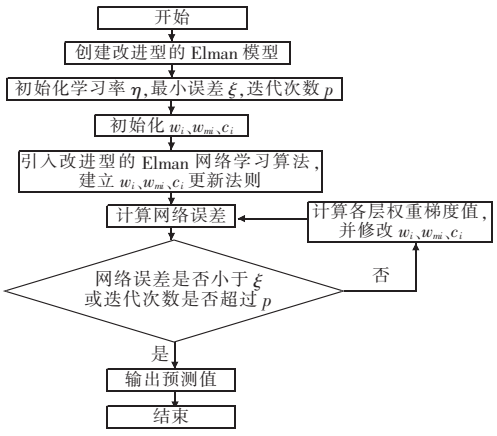

图 2　训练算法流程

首先进行隐含层到输出层的训练。隐含层到输出层之间的权值可以通过输出层的误差直接求导来进行求解,在求解的过程中,在误差计算中加入改进的函数,具体训练算法如算法 2 所示。

算法 2　隐含层到输出层的训练算法

数据定义:

t_n:时序数据中的第 n 个时刻;

$w_i(t_n)$:t_n 时刻隐含层到输出层的权值;

$e_Y(t_n)$:t_n 时刻输出层的预测值与实际值之差;

η:训练过程中的学习率。

训练过程:

t_n 时刻输出层的误差如下:

$$e(t_n)=\frac{1}{2}\Phi(t_n)(e_Y(t_n))^2$$

计算误差对 w_i 的梯度,$\frac{\partial e(t_n)}{\partial w_i(t_n)}$,再沿着该方向反向进行调整,如下:

$$\Delta w_i(t_n)=-\eta\frac{\partial e(t_n)}{\partial w_i(t_n)}$$

$$w_i(t_{n+1})=\Delta w_i(t_n)+w_i(t_n)$$

根据微分的链式规则,$\frac{\partial e(t_n)}{\partial w_i(t_n)}$的计算公式如下:

$$\frac{\partial e(t_n)}{\partial w_i(t_n)}=\frac{\partial e(t_n)}{\partial e_Y(t_n)}\times\frac{\partial e_Y(t_n)}{\partial v_Y(t_n)}\times\frac{\partial v_Y(t_n)}{\partial u_Y(t_n)}\times\frac{\partial u_Y(t_n)}{\partial w_i(t_n)}$$

可得隐含层到输出层的更新式如下:

$$w_i(t_{n+1})=w_i(t_n)-\eta\frac{1}{\alpha}\mathrm{e}^{\int_{t_n}^{t_a}\rho(t)\mathrm{d}t+\int_{t_n}^{t_a}\theta(t)\mathrm{d}G(t)}e_Y(t_n)g'(u_Y(t_n))v_I^i(t_n)$$

输入层到隐含层训练算法相比于隐含层到输出层的训练算法较为复杂,主要是因为输出层的误差不能直接对输入层隐含层之间的权值进行求导,所以需要通过隐含层中的局部梯度进行计算,相应地在误差计算过程中添加改进的函数,具体的算法流程如算法 3 所示。

算法 3　输入层到隐含层训练算法

数据定义:

$w_{mi}(t_n)$:t_n 时刻输入层到输出层的权值。

训练过程:

计算误差对 w_{mi}的梯度,$\frac{\partial e(t_n)}{\partial w_i(t_n)}$,并将该梯度乘以负 1 即相反的方向调整,公式如下:

$$\Delta w_{mi}(t_n)=-\eta\frac{\partial e(t_n)}{\partial w_{mi}(t_n)}$$

$$w_{mi}(t_n)=\Delta w_{mi}(t_n)+w_{mi}(t_n)$$

根据微分的链式规则,$\frac{\partial e(t_n)}{\partial w_{mi}(t_n)}$的计算公式如下:

$$\Delta w_{mi}(t_n)=-\eta\frac{\partial e(t_n)}{\partial v_I^i(t_n)}\times\frac{\partial v_I^i(t_n)}{\partial u_I^i(t_n)}\times\frac{\partial u_I^i(t_n)}{\partial w_{mi}(t_n)}$$

将上式代入可得隐含层到输出层的更新公式如下:

$$w_{mi}(t_{n+1})=w_{mi}(t_n)-\eta\frac{1}{\alpha}\mathrm{e}^{\int_{t_n}^{t_a}\rho(t)\mathrm{d}t+\int_{t_n}^{t_a}\theta(t)\mathrm{d}G(t)}\times e_Y(t_n)g'(u_Y(t_n))w_i f'(u_I^i(t_n))v_M^m(t_n)$$

连接层到隐含层的训练方法类似于输入层到隐含层的训练方法,可得到隐含层的权值更新公式如下:

$$c_i(t_{n+1})=c_i(t_n)-\eta\frac{1}{\alpha}\mathrm{e}^{\int_{t_n}^{t_a}\rho(t)\mathrm{d}t+\int_{t_n}^{t_a}\theta(t)\mathrm{d}G(t)}e_Y(t_n)\times g'(u_Y(t_n))w_i f'(u_I^i(t_n))v_I^i(t_n)$$

4　实验结果

实验采用配置为 Intel Core i5-4440 CPU 处理器,处理器主频为 3.1 GHz,8 GB 内存,1 TB 硬盘,千兆网卡,操作系统为 Ubuntu14.04 64 位。本实验通过对 ARMA、BP 神经网络、基础的 Elman 神经网络与改进的 Elman 神经网络对余额宝总体申购量的预测准确性进行了对比。

4.1　实验数据

本实验的数据来自于互联网金融公司蚂蚁金服旗下的余额宝用户日常的交易流水时序信息,通过根据窗口的大小对流水数据进行预处理后可得到不同窗口下的数据信息表:窗口为 w 的原始数据、窗口为 w 的原始数据快速傅里叶变换、窗口为 w 的原始数据离散小波变换、窗口为 w 的原始数据离散余弦变换等。

4.2　实验方案

a)特征选择模型实验方案。本实验首先通过对原始时序数据经过特征提取后得到的特征集合作为 Clamping 网络的输入,从集合中挑选出 Clamping 网络认为能代表时序数据的特征,然后使用这些特征来对时序数据进行预测,求得预测值并与实际值进行比较,计算网络的误差指标;然后使用改进的 DS-Clamping 网络对同样的特征集合进行挑选,筛选得到的特征集合用来求得预测值并计算与实际值的误差指标。通过对比两个模型挑选的特征对网络预测的误差指标来评判特征选择模型的优劣,实验方案如图 3 所示。

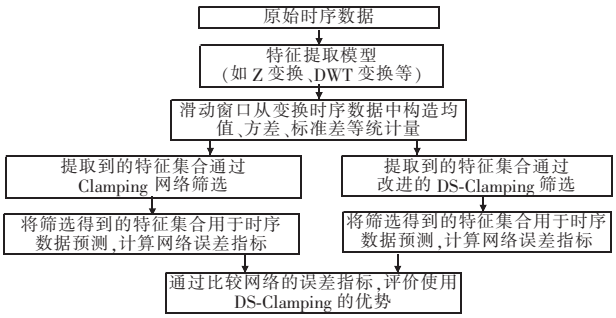

图3　特征选择模型实验方案

b)互联网金融风险预测模型实验方案。该实验对原始时序数据先进行特征学习得到有利于时序数据预测的特征集合,然后使用这些特征集合作为 GT-Elman 神经网络的输入来预测时序数据,得到预测值并与实际值进行比较,计算得到误差指标。将得到的误差指标与未对原始时序进行特征提取而是直接使用原始时序数据特征作为 GT-Elman 的输入得到的误差进行比对,通过比对这两个模型的误差指标来说明对时序数据使用特征学习模型提取时序数据特征的必要性,实验方案流程如图 4 所示。

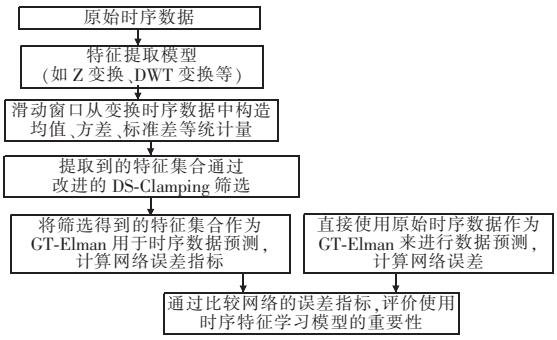

图4　互联网金融风险预测模型实验方案

4.3　实验结果分析

1)特征选择模型实验分析

从时序数据中选取余额宝 20130705 ~ 20140522 这个时间段内用户申购时序数据、银行间拆借利率时间序列、余额宝每天收益率时间序列、festival 时间序列作为训练集 TrainSet,而 20140523 ~ 20140831 这个时间段内的相应时间序列数据作为测试集 TestSet。将这些原始时序数据特征经过特征提取模型得到 312 个时序数据特征,然后将这 312 个时序特征通过 Clamping 网络分别计算各个特征的 impact ratio 值,即 $\xi(x_i)$,各特征对应的 impact ratio 分布值如图 5 所示。

图5 impact ratio 分布值

图 5 中,横坐标是特征序号(即 312 个特征),纵坐标为特征 $\xi(x_i)$值。将这些特征按照 impact ratio 从小到大的顺序进行排序,即按照特征对系统预测的重要程度进行排序。因为实验是根据每个特征对系统误差的影响进行排序,所以计算得到 $g(x|_{x_i=}\text{mean}(x_i))$值比 $g(x)$值越大,则表示该特征 x_i 对网络的预测贡献越大,即在没有使用这个特征进行预测时网络的误差变大了。

根据排序后的特征贡献,可知对网络预测值起重要作用的前十个特征分别是特征标号为 286、271、175、91、297、178、284、295、278、173。这些特征分别是从 festival 时间序列(特征 286、271、297、284、295、278),银行间拆借时间序列(特征 175、178、173),收益率时间序列(特征 91)中提取出来。而其中 festival 时间序列提取的特征对网络预测的贡献最大,前十个特征中占了 6 个,而且大多在 impact ratio 中排前列,这主要是因为在节假日时间里,余额宝申购量相对于日常有较大的提升。另外,相对于 festival 时间序列对网络贡献第二突出的是银行间拆借利润时间序列,从这可以看出银行间的拆借利率对余额宝用户的申购量具有较大的作用。相比较而言,余额宝的收益率时间序列对余额宝用户的申购量的作用相对前两者要弱一些。根据 Clamping 网络,在表 1 ~3 中,根据 impact ratio 排序中选取 impact ratio 小于 0 的特征,即该特征能降低网络整体的误差。通过 Clamping 网络挑选出来的这 153 个特征作为输入来计算网络的误差;然后根据 DS-Clamping 网络,依次将 impact ratio 排序的特征加入到特征的集合中,然后使用该特征集合作为网络的输入并计算相应的误差,添加过程中的阈值设置为 0.05。通过 DS-Clamping 挑选到的特征(相对 Clamping 挑选的特征)如表 1 所示。

表 1 特征挑选集合表

Clamping 特征:{286,271,175,91,297,178,284,295,278,173,…,45,56,81,147,203,206,257,258}
DS-Clamping 剔除的特征:{134,71,236,49,137,162,8,101,84,204,253,99,92,237,260,310,152,56,81,147,203,206,257,258}
DS-Clamping 新加入的特征:{11,55,7,274,23,112,192,31,61,239}
DS-Clamping 特征:{286,271,175,91,297,178,284,295,278,173,…,45,112,165,249,192,31,61,186,239}

分别根据 Clamping 网络筛选的特征集与 DS-Clamping 网络筛选的特征集来计算网络的误差,两种特征集所得的误差指标如表 2 所示。

表 2 预测模型指标

误差	Clamping	DS-Clamping	误差下降百分比/%
MAE(10^7)	5.281 4	5.101 8	3.4
MAPE	22.592 5	21.848 7	3.29
MSE(10^{15})	4.285 2	4.048 3	5.53
RMSE(10^7)	6.546 2	6.362 7	2.9

根据表 2 可得到误差柱状图如图 6 所示。

通过计算 DS-Clamping 网络相对于 Clamping 网络在 MAE、MAPE、MSE、RMSE 指标方面分别下降了 3.4%、3.29%、5.53%、2.9%。通过这个实验可知,相比 Clamping 网络,DS-Clamping 网络能从备选特征集合中提取到对网络预测更有用的特征。

2)互联网金融风险预测模型实验

通过将特征学习模型与 GT-Elman 预测模型相整合得到互联网金融风险预测模型(FLGT-Elman 模型),即将时序特征学习模型学习到的特征集合作为 GT-Elman 网络的输入特征,根据 TrainSet 提取特征并进行训练,然后使用 FLGT-Elman 来预测 TestSet 得到预测的时序数据。将第三章中使用 GT-Elman 得到的预测时序数据,FLGT-Elman 预测到的数据与实际数据进行拟合可得到图 7。

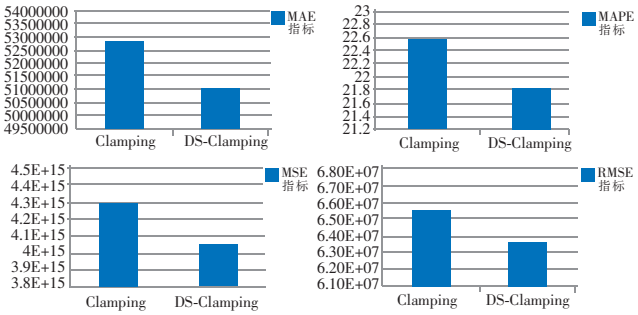

图6 误差柱状图

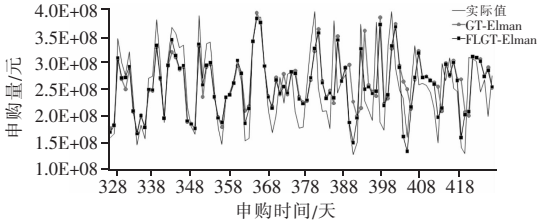

图7 FLGT-Elman 预测拟合图

从图 7 可以看出,GT-Elman 与 FLGT-Elman 模型预测的数据走势总体与实际数据的走势相同,但相比于 GT-Elman 模型,FLGT-Elman 模型能更多地学习到实际数据的一些转折尖峰点时刻的数据。图 8 是截取图 7 中横坐标[390,427]中的数据并将其放大,相比于未对时序数据进行特征学习的 GT-Elman 神经网络预测模型,FLGT-Elman 对于尖锋数据能更好地预测和拟合,如图 8 中 FLGT-Elman 拟合曲线上所画的圆圈点所示。实验证明了时序数据特征学习模型能够更多地提取到原始时序数据的特征,能提高时序数据预测的精度。但从图 8 中也可看到,虽然 FLGT-Elman 预测精度有所提高,但实际时序数据中还是有一部分的尖峰转折数据没能很好地预测到。未能预测到的主要原因是:a)金融时序数据是高维非线性非平稳时序数据;b)实验中使用的原始时序数据序列有限,本实验中只使用了余额宝原始申购量时序数据、余额宝收益率时序数据与银行拆借利率时序数据以及节假日时序数据,如果能提供采集更多与申购量相关的时序数据,则能更好地拟合实际值。

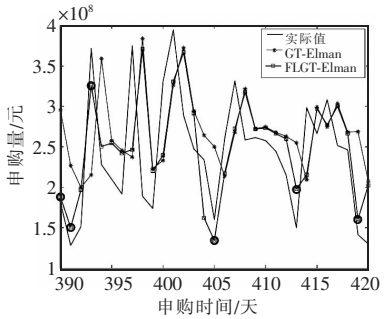

图8 GT-Elman 模型与 FLGT-Elman 模型对尖锋数据拟合

通过计算网络的误差指标,并与直接使用原始特征的 GT-Elman 网络所得的误差进行比较,如表 3 所示。

表 3 GT-Elman 与 FLGT-Elman 预测模型指标

误差	GT-Elman	FLGT-Elman	误差下降百分比/%
MAE(10^7)	4.748 8	4.522 3	4.77
MAPE	20.824 9	19.736 6	5.23
MSE(10^{15})	3.766 9	3.281 4	12.89
RMSE(10^7)	6.137 5	5.728 4	6.67

根据表 3 得到的误差柱状图如图 9 所示。

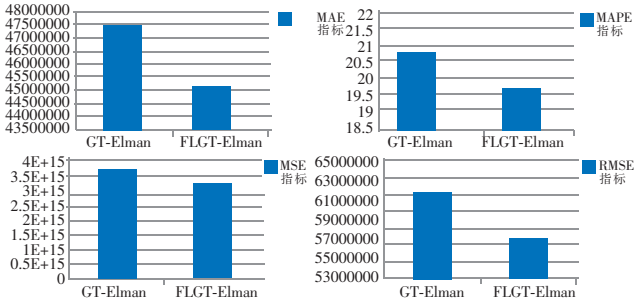

图9　GT-Elman 与 FLGT-Elman 预测指标

4.4　实验结论

本实验一方面首先通过对原始时序数据进行特征提取操作,得到 312 维的特征集合。然后将该特征集合分别通过 Clamping 网络与 DS-Clamping 进行筛选,将两个特征选择模型筛选到的特征用于时间序列预测,分别得到对应的误差指标,通过计算 DS-Clamping 网络相对于 Clamping 网络在 MAE、MAPE、MSE、RMSE 指标方面分别下降了 3.4%、3.29%、5.53%、2.9%。即改进的 DS-Clamping 相对于 Clamping 网络更能从特征集合中挑选出有利于时间序列预测的特征,从而能更好地拟合实际值。

另一方面本实验还通过直接使用原始时序数据特征作为 GT-Elman 模型的输入来预测时序数据与将原始时序数据通过特征学习模型得到的特征作为 GT-Elman 模型的输入来预测时序数据,通过对比两个模型的预测值与实际值之间的误差指标,得到互联网金融风险预测模型(特征学习模型 + GT-Elman 神经网络预测模型)相对于未使用特征学习模型的 GT-Elman 神经网络在 MAE、MAPE、MSE、RMSE 指标方面分别下降了 4.77%、5.23%、12.89%、6.67%,即本章提出的互联网金融风险预测模型在时序数据预测性能上更优。

5　结束语

本文通过对时间序列进行特征学习得到的特征通过神经网络来进行预测。通过改进的 Elman 神经网络,提出了 GT-Elman 神经网络预测模型应用于金融时序数据的预测;较于基础的 Elman 神经网络,GT-Elman 神经网络模型在时序预测中具有更好的准确度。针对 Clamping 网络只考虑到了单一特征参数对结果的影响,改进了 Clamping 网络,提出了 DS-Clamping 应用于时序数据特征选择;通过使用改进型的Clamping网络,能使时序数据预测的精度更高。虽然在通过改进的 Elman 与改进的 Clamping 网络能使预测准确度提高,但还是有一些尖峰数据没能很好地预测到,这主要的原因是只使用了原始的时序数据特征,而未用到用户的特征如用户的性别、城市、消费等一些用户特征,后续研究将考虑用户特征从而提高预测准确率。

参考文献:

[1] Brockwell P J, Davis R A. Introduction to time series and forecasting [M]. Berlin: Springer, 2009.

[2] Yan Juan, Zhao Xiaodong, Li Kang. On temporal resolution selection in time series wind power forecasting[C]//Proc of the 11th International Conference on Control. Piscataway, NJ: IEEE Press, 2016: 1-6.

[3] Jiang Shan, Wang Shuofeng, Li Zhiheng, *et al*. Fluctuation similarity modeling for traffic flow time series: a clustering approach[C]//Proc of IEEE International Conference on Intelligent Transportation Systems. Piscataway, NJ: IEEE Press, 2015: 848-853.

[4] Nguyen T T, Huu Q N, Li M J. Forecasting time series water levels on Mekong river using machine learning models[C]//Proc of the 7th International Conference on Knowledge and Systems Engineering. Piscataway, NJ: IEEE Press, 2015: 292-297.

[5] Liu Yingying, Thulasiraman P, Thulasiram R K. Parallelizing active memory ants with MapReduce for clustering financial time series data [C]//Proc of IEEE International Conferences on Big Data and Cloud Computing. Piscataway, NJ: IEEE Press, 2016: 137-144.

[6] 聂淑媛. 时间序列分析的早期发展[D]. 西安: 西北大学, 2012.

[7] Zhan Shu, Li Weihao, Zhuang Xuan. A novel data mining algorithm based on BP neural network and its applications on stock price prediction[C]//Proc of International Conference on Materials Engineering, Manufacturing Technology and Control. 2016: 1688-1693.

[8] Falat L, Marcek D, Durisova M. Intelligent soft computing on forex: exchange rates forecasting with hybrid radial basis neural network[J]. The Scientific World Journal, 2016, 2016(4): 3460293.

[9] Li Ming, Wang Limin, Liu Yang, *et al*. An improved OIF Elman neural network model with direction profit factor and its applications[C]// Proc of the 2nd International Conference on Machine Vision. 2009: 208-211.

[10] Elman J L. Finding structure in time[J]. Cognitive Science, 1990, 14(2): 179-211.

[11] Niu Hongli, Wang Jun. Financial time series prediction by a random data-time effective RBF neural network[J]. Soft Computing, 2014, 18(3): 497-508.

[12] Wang W, Jones P, Partridge D. Assessing the impact of input features in a feedforward neural network[J]. Neural Computing & Applications, 2000, 9(2): 101-112.

(上接第 2631 页)

[3] 徐鹏飞,苗启广. 基于函数复杂度的自适应模拟退火和禁忌搜索新算法[J]. 电子学报, 2012, 40(6): 1218-1222.

[4] 曹秀爽,姚明林,李兵. 具有捕食策略的混合随机优化算法及其多极值函数优化[J]. 计算机应用, 2014, 34(S2): 162-165.

[5] 张顶学,关治洪,刘新芝. 基于捕食搜索策略的遗传算法研究[J]. 计算机应用研究, 2008, 25(4): 1006-1012.

[6] 刘毅,熊盛武. TSP 问题的禁忌模拟退火求解[J]. 计算机工程与应用, 2009, 45(31): 43-45.

[7] 焦巍,刘光斌,张艳红. 求解约束优化的模拟退火 PSO 算法[J]. 系统工程与电子技术, 2010, 32(7): 1532-1536.

[8] Garcia-Martinez C, Lozano M, Rodriguez-Diaz F J. A simulated annealing method based on specialized evolutionary algorithm[J]. Applied Soft Computing, 2012, 12(2): 573-588.

[9] 付文渊,凌朝东. 布朗运动模拟退火算法[J]. 计算机学报, 2014, 37(6): 1301-1308.

[10] Zhi Jianzhuang, Yu Guibo, Deng Shijie, *et al*. Modeling and simulation about TSP based on simulated annealing algorithm[J]. Applied Mechanics and Materials, 2013, 384(8): 1109-1112.

[11] Rodriguez F J, Garcia M C, Lozano M. Hybrid meta-heuristics based on evolutionary algorithm and simulated annealing: taxonomy comparison and synergy test[J]. IEEE Trans on Evolutionary Computation, 2012, 16(6): 787-800.

[12] 姚明海,王娜,赵连朋. 改进的模拟退火和遗传算法求解 TSP 问题[J]. 计算机工程与应用, 2013, 49(14): 60-65.

[13] Liu Minghua, Shi Yong, Yan Jiashu, *et al*. Lattice Boltzmann simulation of flow and heat transfer in random porous media constructed by simulated annealing algorithm[J]. Applied Thermal Engineering, 2016, 115(3): 1348-1356.

[14] 贺兴时,丁文静,杨新社. 基于模拟退火高斯扰动的蝙蝠优化算法[J]. 计算机应用研究, 2014, 31(2): 392-397.

[15] 刘爱军,杨育,李斐. 混沌模拟退火粒子群优化算法研究及应用[J]. 浙江大学学报: 工学版, 2013, 47(10): 1722-1730.

[16] 孙士平,吴建军. 直接搜索模拟退火算法的自适应改进[J]. 计算机工程与应用, 2015, 51(23): 31-37.

[17] Hedar J. Test functions for unconstrained global optimization[EB/OL]. http://www-optima. amp. i. kyoto-u. ac. jp/member/student/hedar_files/Test GO_files/Page364. htm.