

深度稀疏修正神经网络在股票预测中的应用

邓烜堃^{1,2}, 万良^{1,2}, 马彦勤¹

(1. 贵州大学 计算机科学与技术学院, 贵州 贵阳 550025;

2. 贵州大学 计算机软件与理论研究所, 贵州 贵阳 550025)

摘要:传统的股票预测基于统计学的方法,虽然在股票的趋势预测中有效,但是价格预测的准确度不够令人满意。因此,文中利用神经网络研究股票价格的预测问题。神经网络具有很强的非线性拟合能力,适用于股票等非线性系统建模。文中采集了股票交易的历史数据作为数据集,对数据使用了归一化和主成分分析方法进行预处理,降低了数据维度,有利于简化模型和降低计算复杂度。针对模型的构建,设计了一种深度稀疏修正神经网络模型(deep sparse rectifier neural networks, DSRNN)。DSRNN具有多层网络结构,基于带动量项的BP学习算法训练参数,利用了激活函数ReLU(rectified linear units)和提出的权值初始化方法。并将其与其他三种基于传统方法建立的模型进行了比较,结果表明DSRNN在健壮性、精确度方面都有更好的表现。

关键词:深度神经网络;股票预测;主成分分析;激活函数;权值初始化

中图分类号:TP183

文献标识码:A

文章编号:1673-629X(2018)09-0199-06

doi:10.3969/j.issn.1673-629X.2018.09.041

Application of Deep Sparse Modified Neural Network in Stock Forecasting

DENG Xuan-kun^{1,2}, WAN Liang^{1,2}, MA Yan-qin¹

(1. School of Computer Science & Technology, Guizhou University, Guiyang 550025, China;

2. Institute of Computer Software and Theory, Guizhou University, Guiyang 550025, China)

Abstract: The traditional stock forecasting is based on statistics, which is effective in forecasting stock trends but not satisfactory in the accuracy of price forecasting. Therefore, we apply the neural network for study on the forecast of stock prices. Neural network has a strong nonlinear fitting, suitable for non-linear system modeling like stock. In this paper, we collect the historical data of stock trading as the data sets, using the normalization and principal component analysis to preprocess them, which are advantageous to model simplification and computational complexity reduction. For the model construction, we design a deep sparse correction neural network model (DSRNN) which has a multi-layer network structure, training parameters based on the BP learning algorithm with momentum, using the ReLU (rectified function units) and weight initialization method proposed. Comparing with the other three models based on traditional methods, it shows that DSRNN performs better in terms of robustness and accuracy.

Key words: deep neural network; stock forecasting; principal component analysis; activation function; weight initialization

0 引言

随着经济的发展,中国股市机制日趋完善和健全,吸引了越来越多的投资者投身到股票交易市场当中。股票价格的预测一直是投资者关注的重要问题,从股票产生之初,人们就开始研究股票的价格走势,希望从股票价格的历史走势中分析出特定的规律,从而指导投资者进行买卖操作。

股票价格预测的准确与否直接影响到投资者的盈

利或者亏损,专家和学者针对股票预测问题提出了不同的方案和解决思路。文献[1]提出了一种基于ARMA-GARCH的股票预测模型,对传统的时间序列模型ARMA进行了改进,优化了预测效果。文献[2]提出了一种基于FOA算法优化的SVR模型,相比传统模型,该模型进一步减小了误差。文献[3]提出一种将自适应遗传算法和粗糙集相结合的预测模型,通过实验证明了该模型具有较好的精度。

收稿日期:2017-10-20

修回日期:2018-03-01

网络出版时间:2018-05-16

基金项目:贵州省科学基金(黔科合J字[2011]2328号,黔科合LH字[2014]7634号)

作者简介:邓烜堃(1991-),男,研究生,CCF会员(76234G),研究方向为机器学习;万良,博士,教授,研究方向为形式化方法、信息安全。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20180515.1655.052.html>

股票预测是一项极具挑战性的任务。由于股票系统是非线性的复杂动态系统,其价格走势具有很强的波动性,并且影响价格的因素众多,线性模型并不能很好地解决股票预测问题。人工神经网络(ANN)具有良好的非线性特点,适用于研究股票预测问题。

人工神经网络的研究起源于感知器(perceptron)^[4],感知器通过模拟神经细胞的工作原理使计算机有了学习—推理的能力。但是,单层感知器甚至无法解决异或问题,促使了 BP(error back propagation)学习算法^[5]的产生。使用 BP 学习算法可以训练多层感知器,解决更加复杂的问题,如手写体识别^[6]。但是,BP 学习算法依旧无法训练深层神经网络,直到深度信念网络(deep belief nets)^[7]的出现才破解了这一难题。使用受限布尔兹曼机(restricted Boltzmann machines)对神经网络进行预训练^[8],使神经网络可以向更深的层次进展。

随着深度学习的发展,神经网络的改变不仅在于层数的增加,解决问题的能力也大大提高,并在图像识别^[9]、语音识别^[10]、文本处理^[11]等领域的应用日益增多。因此,文中使用深度神经网络建立股票模型,并对其性能进行实验验证。

1 BP 神经网络

1.1 BP 神经网络模型

BP 神经网络是应用较广泛的多层感知器模型,神经网络结构通常包含三个部分:输入层、隐藏层和输出层。BP 神经网络的学习过程由两部分组成:信号向前传播和误差反向传播。误差反向传播的学习过程使得 BP 神经网络具有较强的非线性映射能力、泛化能力和容错能力。

1.2 BP 学习算法

BP 学习算法是一种基于 Delta 学习规则的算法,其原理是利用梯度下降的方法,将每次训练产生的误差向前传导,依次更新每层的权值和偏置值,反复迭代,最终达到收敛条件后迭代结束。

定义 X 为输入向量, Y 为隐藏层输出向量,输出层输出向量为 O , D 为期望输出向量, $f(x)$ 为激活函数, η 为学习率。

首先,定义系统的损失函数,见式 1:

$$E = \frac{1}{2} \sum_{k=1}^l (d_k - o_k) \quad (1)$$

其中, k 表示第 k 个神经元。

对于隐藏层和输出层,使用梯度下降法调整权值,其中权值改变量 Δw 的第一个下标为前一层的第 j 个神经元,第二个下标为后一层的第 k 个神经元, m 为前一层神经元的数量,对于输出层,权值的变化量为:

$$\Delta w_{jk} = \eta (d_k - o_k) f'(\text{net}_k) y_j \quad (2)$$

$$\text{net}_k = \sum_{j=0}^m w_{jk} y_j \quad (3)$$

对于隐藏层,权值的变化量为:

$$\Delta v_{ij} = \eta \left(\sum_{k=1}^l \frac{\partial E}{\partial \text{net}_k} w_{jk} \right) f'(\text{net}_j) x_i \quad (4)$$

$$\text{net}_j = \sum_{i=0}^m v_{ij} x_i \quad (5)$$

从式 4 可以看出,隐藏层权值调整的过程中引入了输出层的权值,输出层误差的调整会传导到隐藏层,这就是 BP 算法的核心思想。

传统的 BP 学习算法在更新权值时只考虑第 t 轮迭代的更新,并没有考虑第 t 轮以前权值的改变,这使得神经网络在训练过程中可能会发生振荡^[12]。文中使用改进 BP 学习算法,在权值更新的过程中引入了动量项。添加了动量项后提高了模型的训练速度,且在梯度下降的过程中防止陷入局部极小值,提高了模型的准确度。

增加了动量项的权值调整公式如下所示:

$$\Delta W(t) = \eta \delta X + a \Delta W(t-1) \quad (6)$$

其中, W 为权值矩阵; a 为动量系数, a 的取值一般在 $(0,1)$ 之间。

1.3 激活函数

激活函数(activation function)又称转移函数,是神经网络重要的组成要素。激活函数通常具有以下几方面的性质:非线性、连续性、单调性。神经网络模型本身不具备求解非线性问题的能力,之所以能处理非线性问题,很大程度上归功于激活函数。由于激活函数的上述特性,神经元可以将输入映射为非线性的输出。随着神经元个数的增加和网络结构的复杂化,提升了神经网络模型的非线性表达能力。

2 深度稀疏修正神经网络

文中设计了一种深度稀疏修正神经网络模型(deep sparse rectifier neural networks,DSRNN),该模型基于 BP 学习算法,并采用了深度学习中的技术。与传统 BP 神经网络相比主要有以下不同:

(1)使用新型激活函数 ReLU(rectified linear units)。传统的 BP 神经网络通常使用 logistic、tanh 等非线性激活函数。文中使用的 ReLU 是一种分段线性激活函数。

(2)设计了深层的网络结构。传统的 BP 神经网络由于其算法的局限,通常只有一个隐藏层,而 DSRNN 有多个隐藏层,是一种深度神经网络。

(3)使用了新的权值初始化方法。传统的权值初始化方法不利于误差的反向传播,针对梯度消失问题,文中提出了一种新的权值初始化方法。

2.1 稀疏性与修正线性函数

神经科学家通过研究生物大脑发现,大脑同一时间被激活的神经元数目约在1%~4%^[13],这种特性被称作稀疏性。Bengio^[14]针对稀疏性的优势进行了探讨,在面临特征种类多且特征之间关系复杂的数据时,特征之间可能会相互耦合,输入数据的微小改变都可能会产生不同的输出结果。但是,利用神经元的稀疏性可以解决这个问题,通过将部分不重要的神经元置为关闭状态,仅保持部分的神经元激活,神经网络可以从众多特征中解析出少量关键特征,简化数据之间复杂的耦合关系,从而过滤噪音,增强模型的鲁棒性。

获得稀疏性的关键在于只激活部分的神经元,其余的神经元置为关闭。为了达到这个目的,文中使用了ReLU作为激活函数。ReLU由Glorot等^[15]提出,随着神经网络的发展,广泛应用于图像识别领域,并在卷积神经网络(convolutional neural networks, CNN)中有不错的效果^[16]。

ReLU的出现颠覆了对传统激活函数的认识,传统的激活函数在设计上通常满足连续性、对称性、可微性,如S型函数(sigmoid function)logistic和tanh,其定义为式7和式8。

$$f(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

$$f(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (8)$$

相比较早期的阶跃型激活函数,S型函数可以将数值压缩到一个很小的区间内,其可微性也可以使S型函数直接应用于BP神经网络,提高了神经网络求解复杂问题的能力。

随后,Charles Dugas等^[17]提出了softplus激活函数,定义为式9:

$$f(x) = \lg(1 + e^x) \quad (9)$$

softplus继承了传统激活函数的对称性和可微性,但是放弃了对称性,使得softplus出现了一端激活的状态,因此具有一定的生物特性。同时,softplus还是一种不饱和非线性函数,因此不会出现S型函数中的梯度饱和现象。

ReLU既不对称,在0处也不可微,其本质上是一种阈值函数(threshold function),定义为式10。

$$f(x) = \max(0, x) \quad (10)$$

以0为阈值,大于0的原值输出,小于0的使用0作为输出,使得神经元可以很容易地获得稀疏性。相比S型函数,ReLU的线性部分也有利于梯度传导,因此更加适用于深度神经网络。

2.2 模型设计

股票数据数据复杂,波动性强,浅层的神经网络难

以把握数据中蕴藏的规律,因此文中使用深度神经网络设计股票预测模型。DSRNN一共有7层,包含1个输入层、5个隐藏层、1个输出层。输入层的神经元个数与输入数据的维度相对应。通常隐藏层的层数越多,模型的拟合能力就越强,但是,层数的增加也意味着模型越难以训练。综合数据集的规模以及反复试验后的最优结果,将DSRNN设置为5个隐藏层。隐藏层神经元的数量可以不固定,但是经过实验验证,每一层的神经元数量最好大于输入层神经元数,这样可以使输入层的每一个特征与其他特征组合更加充分,学习到数据中更多的细节。输出层有1个神经元,用于输出计算结果。模型所有层之间使用全连接的方式,拓扑结构如图1所示。

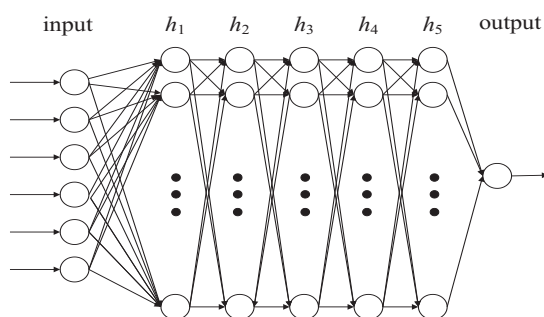


图1 DSRNN 拓扑结构

DSRNN基于BP算法,因此DSRNN的计算过程也包含向前传播和向后传播两个部分。在向前传播过程中,隐藏层的输出 y 如式11:

$$y_j = f\left(\sum_{i=0}^m w_{ji}x_{ji} + b_{ji}\right) \quad (11)$$

其中, b_{ji} 为神经元 j 和 i 之间的偏置值。

在反向传播的过程中,使用梯度下降算法,逐层计算误差并更新权值,权值调整公式为:

$$w_{jil} = w_{jil} - \eta \frac{\partial E}{\partial w_{jil}} \quad (12)$$

其中, w 的第三个下标 l 表示第 l 层。

将式12逐层展开,可以得到每一层的权值调整公式。输出层到第六层的权值调整为:

$$w_{j6k} = w_{j6k} - \eta(d_k - o_k)y_{j5} \quad (13)$$

第六层到第五层的权值调整为:

$$w_{j5s} = w_{j5s} - \eta(d_k - o_k)w_{j6k}y_{j4} \quad (14)$$

第五层到第四层的权值调整为:

$$w_{j4t} = w_{j4t} - \eta(d_k - o_k)w_{j5s}y_{j3} \quad (15)$$

第四层到第三层的权值调整为:

$$w_{j3r} = w_{j3r} - \eta(d_k - o_k)w_{j4t}y_{j2} \quad (16)$$

第三层到第二层的权值调整为:

$$w_{j2u} = w_{j2u} - \eta(d_k - o_k)w_{j3r}y_{j1} \quad (17)$$

第二层到输入层的权值调整为:

$$w_{ji1} = w_{ji1} - \eta(d_k - o_k)w_{j2u}y_{ji} \quad (18)$$

在权值调整过程中,若隐藏层中输出的 $y \leq 0$, 根据 ReLU 的性质,权值不会发生调整。

2.3 权值初始化方法

权值初始化的合适与否对深度神经网络模型十分重要。深度神经网络结构复杂,使用 BP 算法反向传播误差时,误差会随着传播层数的增加而逐渐减小。通过分析式 12~18 可以发现,由于 w 是一个小于 0 的数,连续的相乘会使结果越来越接近 0,导致靠前的神经元权值改变量极小,误差不容易收敛,这种现象叫做梯度消失 (gradient vanish)^[18]。但是,合适的初始化方法可以加快误差的收敛速度,并且可以减小模型陷入局部极小值的概率。

传统的权值初始化方法如式 19 和式 20。式 19 是一种常用的初始化方法,权值的取值服从 -0.5 到 0.5 的均匀分布;式 19 称为标准权值初始化方法 (standard weight initialization),权值初始化的范围根据当前层的结构而定,其中 n 为当前层节点的数量,取值服从均匀分布。

$$w \sim U[-0.5, 0.5] \quad (19)$$

$$w \sim U\left[-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right] \quad (20)$$

针对 DSRNN 权值的初始化,文中提出了一种新的权值初始化方法。为了使信息有效地在神经网络中传播,每一层输出的方差应尽可能相等。首先,任意一层输出的方差可以表示如下:

$$\text{Var}(w_i x_i) = E[w_i]^2 \text{Var}(x_i) + E[x_i]^2 \text{Var}(w_i) + \text{Var}(w_i) \text{Var}(x_i) \quad (21)$$

由于 w 和 x 的期望均为 0,式 21 可进一步化简为:

$$\text{Var}(w_i x_i) = \text{Var}(w_i) \text{Var}(x_i) \quad (22)$$

w 和 x 独立同分布,其中 n 为第 i 层节点的数量,则有:

$$\text{Var}(y) = n_i \text{Var}(w_i) \text{Var}(x_i) \quad (23)$$

假设每一层输出的方差均相等,对于单独一层来说,等价于输入和输出的方差相等,式 23 可以表示为:

$$\text{Var}(w_i) = \frac{1}{n_i} \quad (24)$$

在多层神经网络中,为了保证每一层输出的方差都相等,应满足:

$$\forall i, n_i \text{Var}[w_i] = n_{i+1} \text{Var}[w_{i+1}] \quad (25)$$

联立式 24 和式 25,可得到 w 的方差为:

$$\forall i, \text{Var}[w_i] = \frac{n_i + n_{i+1}}{2n_i n_{i+1}} \quad (26)$$

根据均匀分布的性质,已知其方差,可以求得其分布区间, w 的分布区间为:

$$w \sim U\left[\frac{\sqrt{3(n_i + n_{i+1})}}{\sqrt{n_i n_{i+1}}}, \frac{\sqrt{3(n_i + n_{i+1})}}{2\sqrt{n_i n_{i+1}}}\right] \quad (27)$$

式 27 即为文中提出的权值初始化方法,该方法初始化时考虑了第 i 层和第 $i+1$ 层神经元个数。经实验验证,该方法在多层神经网络中有利于误差反向传播。

2.4 模型输入输出

模型的输入为股票交易中的历史数据。但是,原始数据维度高,且存在多重共线性以及噪声等问题,不利于模型训练。因此,需要对数据进行预处理,使处理后的数据方便神经网络计算。文中数据预处理包含归一化和特征降维两个部分。

1. 归一化。

首先,先将输入的数据样本进行归一化处理。由于原始数据量纲不同,这会导致数量级较大的数据对最终的结果产生较大的影响,而数量级较小数据对结果几乎没有产生影响。这对最终的实验结果造成了极大的干扰,因此需要将数据的数量级进行统一。文中使用线性函数归一化 (min-max scaling) 方法:

$$x' = \frac{x - \min}{\max - \min} \quad (28)$$

通过归一化操作,将数据压缩到 (0,1) 之间,并防止使用 logistic 和 tanh 激活函数的神经元出现梯度饱和的现象^[19]。

2. 特征降维。

数据归一化后,使用 PCA (principal component analysis) 法进行特征降维。特征降维的主要目的是从众多特征中提取出主要特征,通过投影变换,组成新的低维特征空间^[20]。特征降维对于神经网络有两点好处:去除了数据中的噪音,防止过度拟合;简化了神经网络结构,减少了神经元个数,降低了神经网络的训练难度。

特征降维的过程主要分为:

(1) 对归一化后的数据集 M 计算其协方差矩阵,得到一个 $n \times n$ 的矩阵 X ,其中 M 为 $n \times m$ 的矩阵, m 为数据集 M 的特征维度。

(2) 计算矩阵 X 的特征值和特征向量。

(3) 使用步骤 2 求出的特征值计算方差贡献度。为了减少信息的损失,将方差贡献度的阈值设置在 95%,即最终将累计方差贡献度在 95% 以上且排在前 k 个的特征值选出,记录其对应的特征向量。

(4) 使用步骤 3 选出的特征向量组成新的矩阵 Y ,矩阵 M 在 Y 上做投影运算,得到一个 $n \times k$ 的新矩阵 Z , Z 即为降维后的数据。

模型输出为股票的预测价格,但是输出值在 -1 到 1 之间,不方便与真实值进行比较,需要反归一化处理,反归一化方法如下:

$$x = x'(\max - \min) + \min \quad (29)$$

其中, \max 、 \min 为求解式 28 时计算出的最大和最

小值。

3 实 验

3.1 实验数据

使用贵州茅台(600519)2013 年 1 月 4 日至 2017 年 5 月 4 日的交易数据作为数据集。该数据集包含 28 个维度的数据,基本囊括了技术分析中的常用指标。分别是开盘价,收盘价,最高价,成交量,最低价,换手率,MA5,MA10,MA20,VOLUME (VOLUME, MA5, MA10),MACD(DIF,DEA,MACD),RSI(RSI1,RSI2,RSI3),KDJ(K,D,J),PSY(PSY,PSYMA),VR(VR,MAVR),CCI,ROC(ROC,MAROC)。

数据集使用 2.4 节提到的方法进行归一化和降维处理,降维后得到只有 6 个维度的数据集。这 6 个维度的数据实际方差贡献率为 95.2%,包含了数据中绝大部分的特征。处理后的数据集划分为两部分:训练集和测试集。训练集包含 831 条数据,测试集包含 220 条数据,分别用于模型训练和模型测试。

3.2 实验结果与分析

文中使用第一天的数据拟合第二天的股票价格,从准确度和鲁棒性两个方面将 DSRNN 与传统的 BP 神经网络进行了对比,并比较了 DSRNN 使用文中提出的初始化方法和传统初始化方法后误差收敛的差异。实验的详细结果如下:

1. 模型准确度。

为了比较模型的准确性,使用了带动量项的神经网络模型,同时也对比了不带动量项的神经网络模型,动量系数为 0.7。表 1 中的数据是使用相同权值初始化方式,不同的激活函数计算得出。其中训练误差由测试集数据计算误差平方和 (SSE) 得出,预测误差由输出值和目标值计算平均绝对误差 (MAE) 得到,数据是 10 次测量取平均值的结果。

表 1 不同模型预测结果对比

激活函数	网络结构	学习率	误差	不带动量项	带动量项
ReLU	6-10-10-10-10-10-1	0.000 3	预测误差	24.62	9.45
			训练误差	1.97	0.30
tanh	6-12-1	0.001	预测误差	69.67	28.58
			训练误差	1.56	0.12
softplus	6-5-1	0.005	预测误差	92.13	25.99
			训练误差	2.37	1.24
logistic	6-5-1	0.01	预测误差	115.42	18.54
			训练误差	6.17	1.02

分析表 1 数据可以看出,使用动量项优化过的模型比没经过优化的模型在训练精度和预测精度方面均有了显著提高,说明使用动量项优化可行且有效。总

体来看,DSRNN 在两种条件下比其他三种模型都具有更好的表现。由于 DSRNN 是一种深度神经网络,在训练时可以学习到数据中更多的特征,从而可以更好地把握数据中隐含的规律,因此具有较高的精度。

2. 模型健壮性。

DSRNN 拥有较多的神经元,但是,过多的神经元通常也会导致过度拟合的问题。为了比较不同神经网络模型在不同隐层节点数目下抵抗过度拟合的能力,使用 5~20 个隐藏层神经元分别进行实验。数据为 10 次测量取平均值所得,结果如表 2 所示。

表 2 不同隐层节点数目预测结果对比

模型	误差	5	10	15	20
DSRNN	预测误差	15.25	10.52	9.64	9.90
	训练误差	0.33	0.29	0.29	0.27
tanh-BP	预测误差	28.94	30.41	27.93	28.99
	训练误差	0.23	0.17	0.13	0.12
softplus-BP	预测误差	27.79	85.94	123.6	131.17
	训练误差	1.38	0.65	0.53	0.52
logistic-BP	预测误差	17.49	27.50	34.04	38.98
	训练误差	2.48	1.09	0.81	0.62

分析表 2 数据可知,随着隐藏层节点数目的增加,基于 tanh、softplus 和 logistic 的模型均出现了不同程度的过度拟合。即数据在训练集上表现优异,训练误差逐渐减小,但是在测试集上误差却不断增加。这是由于隐藏层神经元过多导致的,过多的隐藏层神经元会学习到数据中更多的细节,但是过度地强调细节会使模型的泛化能力减弱,因此在测试集中表现较差。

与其他几种模型不同,DSRNN 并没有表现出过度拟合的现象,随着隐藏层神经元的增加,训练误差和预测误差均保持在比较稳定的状态。由于 DSRNN 的稀疏性会自动地关闭其中不重要的神经元,激活的神经元始终保持在相对稳定的数量,避免了过度拟合的发生。因此,DSRNN 具有更好的鲁棒性。

3. 权值初始化方式。

DSRNN 相比传统的浅层神经网络更不容易训练,因此不好的权值初始化方式会影响 DSRNN 的性能。为了证明文中提出的初始化方式对 DSRNN 性能的提高,进行了以下实验。为表述简便起见,将式 19 表示为 A,式 20 表示为 B,式 27 表示为 C,分别对权值进行初始化,记录实验误差。对 10 次实验结果取平均,如表 3 所示。

表 3 不同初始化方式预测结果对比

误差	A	B	C
平均预测误差	10.19	11.04	9.79
平均训练误差	0.35	0.31	0.34

由表 3 可知,模型收敛后,三种初始化方式最终的训练误差和预测误差相差无几,具有相似的精度,因此需要比较三种方式在训练过程中的差异。文中采集了三种初始化方式在前 600 轮训练中的误差变动数据,并绘制成图 2。

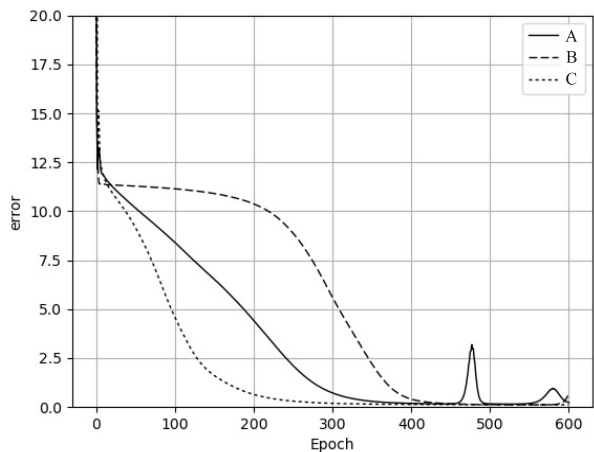


图 2 训练过程中的误差变动

可以看出,经过 600 次训练,三种方案最终收敛到了相同精度。比较训练的过程可以发现,A 进行了约 300 多轮训练收敛,B 进行了约 400 多轮训练收敛,C 则进行了 200 多轮训练就收敛,C 的误差收敛速度明显快于 A 和 B,说明文中提出的初始化方法要优于其余两种初始化方法,在 DSRNN 中有利于误差的反向传播,大大提升了模型的训练效率。

4 结束语

由于股票系统具有非线性特点,传统的预测方法往往精度不高,对此设计了一种深度神经网络 DSRNN 用于股票预测。针对深度神经网络训练过程中梯度消失的问题,DSRNN 使用了 ReLU 激活函数并提出一种新的权值初始化方法。实验结果表明,设计的股票预测模型 DSRNN 相比传统的 BP 神经网络模型,具有更好的准确度和健壮性,抗过度拟合能力强。而权值初始化方法可以提高误差在 DSRNN 中的传播效率,使其有更快的训练速度。模型的平均相对误差 (ARE) 在 3.5% 左右,而股票的涨跌幅限制在 10% 以内,因此对实际的投资具有一定的指导意义。

参考文献:

- [1] 杨琦,曹显兵. 基于 ARMA-GARCH 模型的股票价格分析与预测[J]. 数学的实践与认识,2016,46(6):80-86.
- [2] 王卫红,卓鹏宇. 基于 PCA-FOA-SVR 的股票价格预测研究[J]. 浙江工业大学学报,2016,44(4):399-404.
- [3] 张炜,范年柏,汪文佳. 基于自适应遗传算法的股票预测模型研究[J]. 计算机工程与应用,2015,51(4):254-259.

万方数据

- [4] ROSENBLATT F. The perceptron: a perceiving and recognizing automaton[R]. NY: Cornell Aeronautical Laboratory, 1957.
- [5] RUMELHART D E, HINTON G E, WILLIAMS R J. Learning representations by back-propagating errors[J]. Nature, 1986, 323(6088):533-536.
- [6] LECUN Y, BOSER B, DENKER J S, et al. Backpropagation applied to handwritten zip code recognition[J]. Neural Computation, 1989, 1(4):541-551.
- [7] HINTON G E, OSINDERO S, TEH Y W. A fast learning algorithm for deep belief nets[J]. Neural Computation, 2006, 18(7):1527-1554.
- [8] NAIR V, HINTON G E. Rectified linear units improve restricted Boltzmann machines[C]//Proceedings of the 27th international conference on machine learning. Haifa, Israel: Omnipress, 2010:807-814.
- [9] 孙晓,潘汀,任福继. 基于 ROI-KNN 卷积神经网络的面部表情识别[J]. 自动化学报,2016,42(6):883-891.
- [10] 张晴晴,刘勇,潘接林,等. 基于卷积神经网络的连续语音识别[J]. 工程科学学报,2015,37(9):1212-1217.
- [11] 黄磊,杜昌顺. 基于递归神经网络的文本分类研究[J]. 北京化工大学学报:自然科学版,2017,44(1):98-104.
- [12] 韩力群. 人工神经网络理论、设计及应用[M]. 北京:化学工业出版社,2007.
- [13] LENNIE P. The cost of cortical computation[J]. Current Biology, 2003, 13(6):493-497.
- [14] BENGIO Y. Learning deep architectures for AI[J]. Foundations and Trends in Machine Learning, 2009, 2(1):1-127.
- [15] GLOROT X, BORDES A, BENGIO Y. Deep sparse rectifier neural networks[C]//International conference on artificial intelligence and statistics. Fort Lauderdale, FL, USA: [s. n.], 2012:315-323.
- [16] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks[C]//International conference on neural information processing systems. Lake Tahoe, Nevada: Curran Associates Inc., 2012:1097-1105.
- [17] DUGAS C, BENGIO Y, BELISLE F, et al. Incorporating second order functional knowledge into learning algorithms[J]. Advances in Neural Information Processing Systems, 2001, 8(13):472-478.
- [18] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks[C]//International conference on artificial intelligence and statistics. [s. l.]: [s. n.], 2010:249-256.
- [19] 胡上尉,刘琼荪,孙海雷. BP 网络模型中输出神经元过早饱和的机理研究[J]. 计算机工程,2007,33(3):202-204.
- [20] ABDI H, WILLIAMS L J. Principal component analysis[J]. Wiley Interdisciplinary Reviews Computational Statistics, 2010, 2(4):433-459.