计算机工程
Computer Engineering
ISSN 1000-3428,CN 31-1289/TP

《计算机工程》网络首发论文

题目：基于 BP 神经网络的异常轨迹检测方法研究
作者：俞庆英，李倩，陈传明，林文诗
DOI：10.19678/j.issn.1000-3428.0051574
网络首发日期：2018-11-01
引用格式：俞庆英，李倩，陈传明，林文诗．基于 BP 神经网络的异常轨迹检测方法研究[J/OL]．计算机工程. https://doi.org/10.19678/j.issn.1000-3428.0051574

计 算 机 工 程
Computer Engineering

基于 BP 神经网络的异常轨迹检测方法研究

俞庆英，李倩，陈传明，林文诗
(1.安徽师范大学 计算机与信息学院，安徽 芜湖 241002；
2.安徽师范大学 网络与信息安全安徽省重点实验室，安徽 芜湖 241002)

摘 要：针对现有算法不能有效根据轨迹内外部属性进行异常检测问题，提出了一种基于 BP 神经网络的异常轨迹识别方法。首先，对原始轨迹数据进行去噪处理，并上传到百度云的 LBS.云端进行存储；其次，设计并实现了一个基于百度地图的轨迹数据可视化网站，实现轨迹的显示；然后，将所得数据进行归一化处理，计算出轨迹属性值；最后，将轨迹内外部特征属性作为 BP 神经网络算法的输入层，轨迹相似度量值作为输出层，调整隐含层系数，得到训练模型。在 Geolife 项目中的两个用户轨迹数据集上进行了仿真实验，识别用户异常轨迹数据；结果表明，基于所选最优训练方案，在两个数据集上的异常轨迹识别准确率分别达到 92.3%和 100%。因此，该模型可以作为轨迹异常检测工具。

关键词：轨迹数据集；BP 神经网络；百度 LBS.云服务；轨迹属性；训练模型；异常轨迹检测

Research on Abnormal Trajectory Detection Method Based on BP Neural Network

YU Qingying, LI Qian, CHEN Chuanming, LIN Wenshi
(1. School of Computer and Information, Anhui Normal University, Wuhu Anhui 241002, China;
2.Anhui Provincial Key Laboratory of Network and Information Security, Anhui Normal University, Wuhu Anhui 241002, China)

【Abstract】Aiming at the problem that existing algorithms can not effectively detect the anomaly according to the internal and external attributes of trajectories, an abnormal trajectory detection method based on BP neural network is proposed. First, the original trajectory data is denoised and uploaded to Baidu LBS. cloud for storage. Second, a Baidu map-based trajectory data visualization website is designed and implemented to display the trajectory. Then, the resulting data is normalized to calculate the trajectory attribute values. Finally, take the trajectory internal and external feature attributes as the input layer of the BP neural network algorithm, and the trajectory similarity metrics as the output layer, then adjust the hidden layer coefficients to obtain the training model. Simulation experiments are performed on two user trajectory datasets in the Geolife project to detect the user's abnormal trajectory data. The results show that based on the selected optimal training scheme, the accuracy of abnormal trajectory detection on the two datasets reaches 92.3% and 100% respectively. Therefore, this model can be used as a trajectory anomaly detection tool.

【Key words】Trajectory dataset; BP neural network; Baidu LBS. cloud service; Trajectory attributes; Training model; Abnormal trajectory detection
DOI:10.19678/j.issn.1000-3428.0051574

0 概述

随着移动通信技术、无线网络技术、全球定位系统（GPS）、Wi-Fi 和蓝牙室内定位技术的快速发展以及智能设备的普及率越来越高，人们获取移动对象的位置及轨迹数据变得越来越方便[1-2]。同时，轨迹数据存储技术迅猛发展，时空数据库中存储了大量的移动对象位置和轨迹信息。轨迹是由带有时

基金项目：国家自然科学基金资助项目(61702010、61672039)，安徽高校自然科学研究重点项目(KJ2017A327)，芜湖市科技计划项目(2016cxy04).

作者简介：俞庆英(1980-)，女，安徽黄山人，副教授，博士研究生，CCF 会员(会员号: E200047117M)，主要研究方向：空间数据处理、信息安全；李倩(1995-)，女，安徽亳州人，本科生，主要研究方向：数据挖掘；陈传明(1981-)，男，安徽六安人，副教授，博士研究生，主要研究方向：数据挖掘、智能计算；林文诗(1994-)，女，浙江台州人，硕士研究生，主要研究方向：信息安全、空间数据处理。 **E-mail：**ahnuyuq@ahnu.edu.cn

间标记的位置信息所组成的有序位置序列，其中包含丰富的时空信息，分析和挖掘轨迹数据具有重要的意义[3-6]，其中，异常轨迹检测是轨迹模式挖掘中的一个重要研究课题，广泛应用于智能交通和用户行为分析等领域[7-9]。

轨迹数据具有丰富的内外在特征，提取并分析各种属性特征是轨迹数据挖掘的基础。神经网络是一个基于经验知识的有自然倾向的大的并行分布处理器[10]。BP 神经网络无需事先指定输入输出之间的映射关系，只需通过自身的训练，学习某种规则，利用某种算法在给定输入值时得到最接近期望输出值的结果。本文利用 BP 神经网络算法的这种特性，将轨迹自身的基本属性作为 BP 神经网络的输入层，经过隐含层的权值和阈值调整，基于输出层的结果检测出异常轨迹；并提取正常和异常轨迹的时空特性和用户的行为特征，在进一步工作中，可以预测出用户接下来的行为，以便实现用户个性化推荐等应用。

1 相关工作

为了检测出用户的异常轨迹，研究人员已做了大量的工作。刘良旭等[11]使用基于 R-Tree 的高效异常轨迹检测算法，根据轨迹间的距离特征矩阵来计算轨迹之间的距离以确定其是否匹配。鲍苏宁等[12]提出了基于核主成分分析（KPCA）的异常轨迹检测方法，首先采用 KPCA 对轨迹进行空间转换，将非线性空间转换成线性空间；然后，采用一类支持向量机对轨迹特征数据进行无监督学习和预测；最终检测出具有异常行为的轨迹。Zhu 等[13]提出了依赖时间的异常轨迹检测算法 TPRO，将每条轨迹与其相关的常规路线进行比较，找出历史轨迹数据集中所有的异常轨迹。韩旭[14]提出了基于轨迹多特征的在线异常检测方法，首先通过高斯模型（GMM）学习监控场景的轨迹起点位置分布模式，依据该模式建立一个轨迹起点分布模型，然后将移动窗当作基本比较单元，建立一个基于位置距离及方向距离的分类器，最后通过在线多特征异常检测算法从起点分布、空间位置和运动方向三个层次来衡量待测轨迹和正常轨迹模式之间的差异，判断轨迹是否异常；并通过滑动窗口实现对动态递增轨迹数据的在线检测。朱燕等[15]首先提取相同起止点的轨迹集，再基于轨迹间的相似性聚类实现出租车异常轨迹的检测。

毛嘉莉等[16]首先分类总结了现有轨迹异常检测技术的研究成果，然后根据轨迹异常检测方法的不足，提出了一种轨迹大数据异常检测的系统架构，最后，在轨迹异常的演化分析、在线轨迹流的异常检测、轨迹异常检测系统的基准评测、异常检测结果语义分析的数据融合以及轨迹异常检测的可视化技术等方面探讨了今后的研究工作。

以上研究虽取得一定成果，但在检测异常轨迹时没有充分利用轨迹内部各个特征之间的联系；另外，就个人轨迹分类、异常检测而言，目前还没有一套完备的研究体系，缺乏完备的算法来判断用户轨迹是否属于异常。为此，本文提出了一种基于 BP 神经网络的异常轨迹检测方法，主要贡献有两个方面：1）轨迹数据可视化平台设计；2）提取轨迹属性特征向量，通过 BP 神经网络训练得到轨迹异常判断模型，利用此训练好的模型，判定用户轨迹是否异常，得到用户的异常轨迹数据。

2 问题描述及相关定义

2.1 问题描述

目前对轨迹异常检测的研究，主要以轨迹空间特征为主，很少使用轨迹自身的属性作为研究的基础。而通过对非线性轨迹属性数据进行分类，进而检测轨迹数据是否属于异常，是一个切实可行的轨迹分类方法。BP 神经网络具有很好地对非线性数据分类的功能，因此，本文提出一种基于 BP 神经网络的异常轨迹检测方法。

BP 神经网络主要分为两个阶段。第一阶段是信号的前向传播，以一定的学习原则进行学习，主要是利用一些观测样本来训练网络。学习开始，首先设定各节点传递函数的参数、形式以及各边权重的初始值；对每个输入值，BP 神经网络依据边的权重及各节点的函数，计算输出结果。第二阶段是误差的反向传播，在这一阶段，把输出结果和实际结果相比较，计算输出误差，以误差的方向和大小为依据，逆向调整各边的权重，使 BP 神经网络的输出结果逐渐向实际的观测结果靠近。

BP 神经网络是一种单向传播、多层前向网络。除输入、输出节点外，还有一层或多层隐含节点，其上下层节点之间全连接，每层神经元之间无连接。假设输入层有 n 个神经元，可接受输入向量 $\boldsymbol{X}=(x_1,x_2,\cdots,x_n)$，输入值从输入层节点依次经各隐含层节点，然后到达输出节点，产生输出向量 $\boldsymbol{Y}=(y_1,y_2,\cdots)$，每个节点代表单个神经元，其中对应的传递函数为 Sigmoid 型函数。

2.1.1 传递函数设计

传递函数使用非线性函数 *logsig*()，该对数 S 形函数的曲线如图 1 所示。

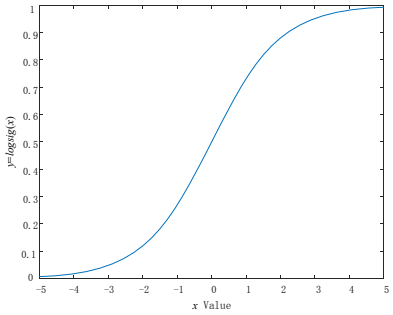


图 1 对数 S 形函数的曲线图

其中 $logsig(x)=\frac{1}{1+e^{-x}}$，产生 0 到 1 之间的输出。本文实验将轨迹分为两个类别，因此采用此函数作为传递函数得到 0 和 1 这两个输出，0 代表异常轨迹，1 代表正常轨迹。

2.1.2 训练函数设计

网络的训练使用 OSS(One Step Secant)算法[17]，其主要函数是 *trainoss*。权值按照 $W=W+f*dX$ 进行修正，其中 dX 为搜索方向，f 是沿着搜索方向的最小化性能函数。一开始的搜索沿着梯度负方向进行，然后在迭代中按照 $dX=gX+Ac*Xstep+Bc*dgX$ 来修改，其中 gX 表示梯度，$Xstep$ 表示前次迭代权值的变化，dgX 代表最近一次迭代梯度的变化，Ac 和 Bc 代表新搜索方向的调整参数。该算法具有较快的收敛速度，且网络性能好，所以本实验训练函数采用 *trainoss*。

2.1.3 网络结构设计

本文中使用多输入、多隐含神经元和单输出的单层 BP 神经网络。其网络结构如图 2 所示。

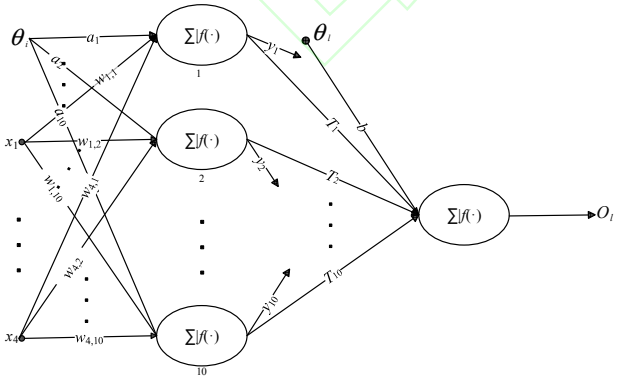


图 2 单层 BP 神经网络结构图

网络学习公式推导的指导思想是，对网络权值($w_{i,j}$,T_j)与阈值(θ)进行修正，使误差函数(E)沿梯度方向下降。其中，x_i 代表 BP 神经网络的输入节点，y_j 代表隐节点，O_l 为输出节点。输入节点与隐节点的网络权值为 $w_{i,j}$，隐节点与输出节点间的网络权值为 T_j。各公式[18]具体如下所示。

本实验中输入节点 x_i（$1\le i\le 4$）的值分别为轨迹的四个属性值，网络权值和阈值使用 MATLAB 中 BP 神经网络工具箱中的初始值。

1) 输入-隐层神经元

输出计算如公式(1)所示：

$$y_j=f\left(\sum_i w_{i,j}x_i-\theta_i\right)=f(net_j) \tag{1}$$

其中 $net_j=\sum_i w_{i,j}x_i-\theta_i$。

隐节点层（输入节点到隐节点）的误差公式如公式(2)所示。

$$\delta_j^{'}=y_j(1-y_j)\sum_l \delta_l T_j \tag{2}$$

隐节点层的权值修正公式如公式(3)所示。

$$w_{i,j}(k+1)=w_{i,j}(k)+\eta^{'}\delta_j^{'}x_i \tag{3}$$

其中，k 为迭代次数，$\eta^{'}\in(0,1)$ 为学习率，其值由算法确定。

隐节点层阈值修正公式如公式(4)所示。

$$\theta_j(k+1)=\theta_j(k)+\eta^{'}\delta_j^{'} \tag{4}$$

2) 隐层-输出层神经元

输出计算如公式(5)所示：

$$O_l=f\left(\sum_j T_j y_j-\theta_l\right)=f(net_l) \tag{5}$$

其中 $net_l=\sum_j T_j y_j-\theta_l$。

设输出节点的期望输出为 t_l，输出层（隐节点到输出节点）的误差计算如公式(6)所示。

$$\delta_l=(t_l-O_l)\cdot O_l\cdot(1-O_l) \tag{6}$$

输出层权修正值计算如公式(7)所示。

$$T_j(k+1)=T_j(k)+\eta\delta_l y_j \tag{7}$$

其中，$\eta\in(0,1)$ 为学习率，本文 η=0.1。

输出层阈值修正如公式(8)所示。

$$\theta_l(k+1)=\theta_l(k)+\eta^{'}\delta_l^{'} \tag{8}$$

其中 $\delta_l^{'}$ 表示 δ_l 的导数，且对于传递函数 $f(x)=\frac{1}{1+e^{-x}}$，存在关系 $f^{'}(x)=f(x)\cdot[1-f(x)]$。

所有样本的总误差 E 如公式(9)所示：

$$E=\sum_{i=1}^{p} e_i<\varepsilon \tag{9}$$

其中一个样本的误差 e_i 如公式(10)所示。

$$e_i=\frac{1}{2}\sum_{l=1}^{n}(t_l-O_l)^2 \tag{10}$$

其中，ε 为总误差，本文设置其值为 10^{-15}，p 为样本数，n 为输出节点数。本实验中 60 条轨迹样本用作训练，输出节点仅有一个。因此 p 的值为 60，n 的值为 1。

2.2 相关定义

轨迹是一个有序位置点构成的数据集，它具有多种属性，本节对与本文实验有关的轨迹及其属性进行定义。

定义 1（轨迹） 轨迹是一个由时空三元组所构成的有序集合，其形式如公式(11)所示：

$$T=\{(t_1,x_1,y_1),(t_2,x_2,y_2),\cdots,(t_n,x_n,y_n)\} \tag{11}$$

其中，时空三元组(t_i, x_i, y_i)为轨迹 T 中的第 i 个位置点，t_i 表示此位置点的采样时刻，(x_i, y_i)表示在 t_i 时刻所处的 GPS 坐标（一般是经度和纬度值）$(1\le i\le n)$。

定义 2（轨迹段） 设有形如式(11)所示的一条轨迹 T，定义其连续两点形成的路径为一个轨迹段，T 中第 i 个轨迹段记为$[(x_i, y_i), (x_{i+1}, y_{i+1})]$，其长度 l_i 的计算方法如公式(12)所示：

$$l_i=\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2} \tag{12}$$

定义 3（轨迹长度） 设有形如式(11)所示的一条轨迹 T，定义其包含的所有轨迹段长度之和为轨迹 T 的长度，计算方法如公式(13)所示：

$$L=\sum_{i=1}^{n-1}l_i \tag{13}$$

定义 4（轨迹角度） 设有形如式(11)所示的一条轨迹 T，定义其连续三点(x_{i-1}, y_{i-1})、(x_i, y_i)、(x_{i+1}, y_{i+1})之间的角度之和为轨迹自身的角度，计算方法如公式(14)所示：

$$A=\sum_{i=2}^{n-1}\cos^{-1}\frac{(x_i-x_{i-1},y_i-y_{i-1})\times(x_{i+1}-x_i,y_{i+1}-y_i)}{|(x_i-x_{i-1},y_i-y_{i-1})|\bullet|(x_{i+1}-x_i,y_{i+1}-y_i)|} \tag{14}$$

定义 5（轨迹所用时间） 设有形如式(11)所示的一条轨迹 T，定义该轨迹从开始时刻 t_s 到结束时刻 t_e 之间的时长为本条轨迹所用时间 Δt，计算方法如公式(15)所示：

$$\Delta t=t_e-t_s \tag{15}$$

定义 6（轨迹平均速度） 设有形如式(11)所示的一条轨迹 T，定义其轨迹长度 L 与其轨迹所用时间 Δt 的商为此轨迹的平均速度 v，计算方法如公式(16)所示：

$$v=L/\Delta t \tag{16}$$

3 基于 BP 神经网络的异常轨迹检测

本文以 BP 神经网络为工具训练得到了轨迹异常检测模型，实现了根据轨迹四个自身属性检测出用户轨迹是否属于异常类的目标。具体研究方案如图 3 所示。

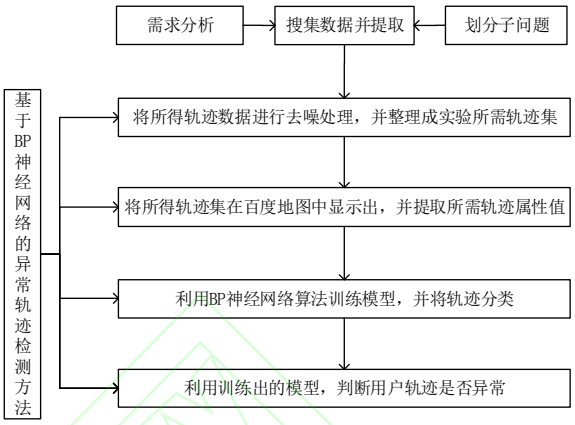


图 3 研究方案示意图

Step 1：轨迹预处理。

对原始轨迹数据集进行去噪处理，根据本文所提轨迹模型，完成轨迹的形式化及预处理工作。

Step 2：轨迹可视化及属性值提取。

利用百度地图的 LBS.云服务，将所获得的轨迹点上传到百度地图 LBS.云服务下的云端服务器存储，然后完成轨迹可视化工作。调用轨迹属性提取算法，计算所需轨迹属性，获得轨迹属性集。

Step 3：基于 BP 神经网络的轨迹聚类模型训练。

将得到的轨迹属性集作为 BP 神经网络的输入，经过训练，调整隐含层系数，直至网络总误差小于给定误差值，得到轨迹聚类模型。

Step 4：基于训练模型，检测用户异常轨迹。

首先调用轨迹属性计算算法，得到待检测用户轨迹的四个属性值，之后将属性值作为训练好的轨迹异常判断模型的输入层，依据模型输出，判断给定用户轨迹所属类别。

3.1 轨迹数据集获取

实验所用轨迹数据源出自微软研究院 GeoLift 项目的 GeoLife GPS Trajectories 数据集。该项目从 2007 年 4 月到 2012 年 8 月收集了 182 个用户的轨迹数据，包含 17621 条轨迹，总距离 120 多万公里，总时间 48000 多小时。每条轨迹含有一系列以时间为序的点，每个轨迹点都有经纬度、海拔等信息。这些轨迹数据不但记录了用户在家和在工作地点的时空轨迹，而且还有大范围的户外活动轨迹，例如购物、远足、旅游、骑自行车[19]。

本文选取 GeoLife GPS Trajectories 中两个用户的轨迹数据集作为实验源数据，记为 *TS*，该数据集包含了用户的轨迹信息及其分类标签。首先对 *TS* 进行噪音去除及坐标转换处理，得到可用轨迹集 *TS'*；接着将用户轨迹显示在百度地图上。

基于百度地图的LBS.云服务技术的轨迹可视化具体步骤如下所示：首先将轨迹集上传到百度地图 LBS.云服务下的云端服务器存储；然后使用 JavaScript API，创建用户窗口类形成类型麻点图形式的用户轨迹数据图层，将其叠加在百度地图上，便可将获得的轨迹数据在百度地图上实时显示出来。

如图 4 所示是某个用户在 2008 年 10 月 25 日 23:48 到 2008 年 10 月 26 日 00:48 的一条轨迹。

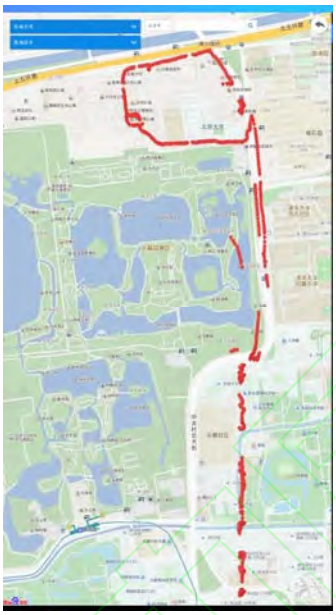


图 4　轨迹显示界面

3.2　轨迹属性值提取

3.2.1　轨迹坐标处理

因轨迹用经纬度表示，且坐标点采样时间间隔为 5 秒，因此两点间的经纬度值差别微小，为了方便计算，且不丢失轨迹自身特征，本文采用直接将轨迹坐标点分别乘以一个适当系数的方法。经实验证明，此方法切实可行。具体操作方法如下。

设有形如式(11)所示的一条轨迹 T，将轨迹集中的 x_i、y_i 分别乘一个系数 λ，此系数值的选取视实际情况而定，以便于计算为原则，本实验中 λ=1000，设处理后得到的轨迹为 $T'=\{(x_1',y_1'),(x_2',y_2'),\cdots,(x_n',y_n')\}$。对轨迹数据集 *TS* 处理后得到的数据集记为 *TS'*。

3.2.2　算法设计

对轨迹集 *TS'*中每条轨迹，将所有位置点之间的距离相加得到轨迹距离属性、将相邻三个位置点形成的角度相加得到轨迹的角度属性、依据百度地图可视化的轨迹中显示的每个点采集的时间，得到轨迹起始时刻 t_s 和轨迹结束时刻 t_e，得到轨迹所用时间 Δt、最后用轨迹长度除以轨迹所用时间得到轨迹平均速度属性。依据定义 3-6 中的公式，轨迹数据集的具体属性值提取算法如下所示：

算法 1： 轨迹属性值提取

输入： 轨迹数据集 $TS'=\{T_1',T_2',\cdots,T_{Tn}'\}$

输出： 长度属性向量 ***Ls***，角度属性向量 ***As***，速度属性向量 ***Vs***，轨迹所用时间向量 ***Dts***

1. Initialize ***Ls***, ***As***, ***Vs***, ***Dts*** ;
2. **for** $i\leftarrow 1$ to Tn **do**
3. $L\leftarrow 0$，$A\leftarrow 0$;
4. **for** $i\leftarrow 2$ to $|T_i'|$-1 **do**
5. $L\leftarrow L+\sqrt{(x_i'-x_{i-1}')^2+(y_i'-y_{i-1}')^2}$;
6. $A\leftarrow A+cos^{-1}\dfrac{(x_i'-x_{i-1}',y_i'-y_{i-1}')\times(x_{i+1}'-x_i',y_{i+1}'-y_i')}{|(x_i'-x_{i-1}',y_i'-y_{i-1}')|\bullet|(x_{i+1}'-x_i',y_{i+1}'-y_i')|}$;
7. **end for**
8. $\Delta t\leftarrow t_e-t_s$;
9. $V\leftarrow L/\Delta t$;
10. ***Ls*** $\leftarrow$ ***Ls*** $\cup L$;
11. ***As*** $\leftarrow$ ***As*** $\cup A$;
12. ***Vs*** $\leftarrow$ ***Vs*** $\cup V$;
13. ***Dts*** $\leftarrow$ ***Dts*** $\cup \Delta t$;
14. **end for**
15. **return** ***Ls***, ***As***, ***Vs***, ***Dts***;

算法 1 针对 *TS'*中的每条轨迹，按照定义 3 至定义 6 分别计算轨迹长度（第 5 行）、轨迹角度（第 6 行）、轨迹所用时间（第 8 行）以及轨迹平均速度（第 9 行）四个轨迹属性值，计算 *TS'*中所有轨迹的长度属性向量 ***Ls***，角度属性向量 ***As***，速度属性向量 ***Vs*** 以及轨迹的所用时间向量 ***Dts***（10-13 行）。

复杂度分析：轨迹数据集 *TS'*中包含 *Tn* 条轨迹，假设每条轨迹最多有 *n* 个位置点，算法 1 的时间复杂度和空间复杂度均为 O(*Tn***n*)。

3.3　利用 BP 神经网络算法训练检测模型

3.3.1　基于 BP 神经网络的异常轨迹检测算法流程

BP 神经网络算法是基于梯度下降的学习算法，学习过程可分为两个阶段，一是信息的正向传递；而是误差的反向传播。流程如图 5 所示。

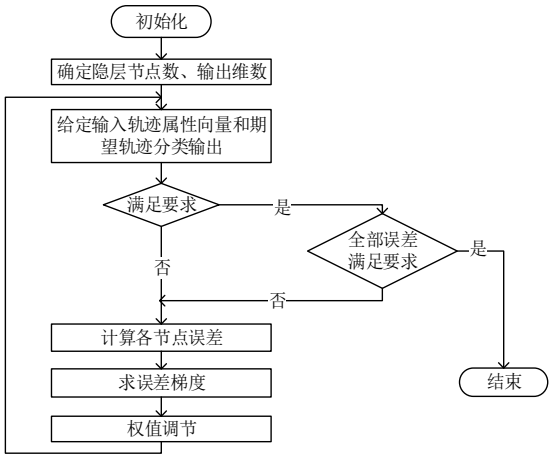


图 5 BP 神经网络算法流程图

（1）确定网络参数。

本实验中，输入层有 4 个神经元，分别输入基于算法 1 提取的四个轨迹属性值；由表 1 可知，并非隐含节点数越多，网络性能越好，当隐含节点增大到 11、12 时误差反而增大，由表 1 也可以看出在隐含节点数为 10 时网络性能最好，因此设定隐含节点数为 10；将用户轨迹分为两类：正常轨迹和异常轨迹，输出节点有两个输出 1 和 0，分别表示这两类轨迹，因此输出层有一个神经元。

表 1 不同节点数网络训练误差对比

节点数	5	6	7	8
网络误差	3.033e-11	1.242e-11	3.324e-11	9.955e-15
节点数	9	10	11	12
网络误差	1.747e-15	3.078e-24	4.730e-11	3.322e-11

（2）初始化网络的权系数和阈值。

输入层-隐含层共有 4*10=40 个权系数，隐含层-输出层有 10*1=10 个权系数，数据量较大，初始系数和阈值采用 Matlab 中 BP 神经网络工具箱的缺省权值和阈值，实验结果表明，此方法具有较好的效果。

（3）确定输入向量和目标输出向量。

输入向量为某用户的所有轨迹属性值，目标输出向量为此用户的某条轨迹分类，即异常和正常轨迹，正常为 1，异常为 0。

（4）计算实际输出值。

按照公式(1)和公式(5)分别计算隐节点和输出节点的输出值。

（5）根据实际输出和目标输出求误差值。

将实际输出和目标输出代入公式(6)，求得输出层误差 δ_l，将 δ_l 代入公式(2)求得隐节点层的误差 δ_j'。

（6）利用求出的误差值修改权系数。

将前一步求得的 δ_l 代入权值修正公式(7)修正权值，根据公式(8)修正阈值；同理按照公式(3)和公式(4)修正隐节点层的权值和阈值。

（7）转到步骤（3），直到模型稳定为止。

当样本误差满足公式(9)和公式(10)时，即当最小均方误差(*mse*)为 10^{-15} 以下即可完成训练，至此 BP 神经网络模型训练完成。

3.3.2 基于 BP 神经网络的异常轨迹检测算法描述

本实验创建一个三层的 BP 神经网络，主要指令为：*net* = *newff* (*PR*, [S_1 S_2 … S_N], {TF_1 TF_2 … TF_N}, *BTF*, *BLF*, *PF*)。其参数意义如下：

PR：输入向量的取值范围。

S_i：第 i $(i=1,2,\cdots,N-1)$ 层神经元的个数，总共 N 层。

TF_i：第 i 层的传递函数，本实验采用 *logsig* 函数。

BTF：BP 神经网络的训练函数，本实验采用 *tranoss* 函数。

BLF：BP 神经网络权值和阈值学习函数，实验采用的是 *learngdm* 函数。

PF：性能函数，本实验使用最小均方误差(*mse*)作为性能函数。

执行结果：创建一个 N 层的 BP 神经网络模型。

基于带分类标签的数据集，首先用现有轨迹对设计好的模型进行训练，得到训练稳定的模型后将未知轨迹放入此模型进行检测，判断该条轨迹是否为异常轨迹。算法描述如下：

算法 2：基于 BP 神经网络的异常轨迹识别

输入：轨迹数据集 *TS'*，长度属性向量 ***Ls***，角度属性向量 ***As***，速度属性向量 ***Vs***，轨迹所用时间向量 ***Dts***

输出：轨迹所属类别 0 或 1

1. $\boldsymbol{P} \leftarrow [\boldsymbol{Ls}; \boldsymbol{As}; \boldsymbol{Ts}; \boldsymbol{Dts}]$;
2. $\boldsymbol{T} \leftarrow$ The vector of labels in *TS'*;
3. $NodeNum \leftarrow 10$;
4. $TypeNum \leftarrow 1$;
5. $Epochs \leftarrow 10000$;
6. $TF_1 \leftarrow$ '*logsig*';
7. $TF_2 \leftarrow$ '*logsig*';
8. *net*←*newff*(*minmax*(***P***),[*NodeNum TypeNum*],{TF_1 TF_2},'*trainoss*', '*learngdm*');
9. *net.trainParam.epochs* ← *Epochs*;

10. *net.trainParam.goal*←1e-15;
11. *net.trainParam.time*←Inf;
12. *train*(*net*, ***P***, ***T***);
13. *x*←*sim*(*net*, ***P_test***);
14. return *x*;

其中 *minmax*(***P***)函数计算实际输入矩阵 ***P*** 的最大最小值；*train*(*net*, ***P***, ***T***)为训练函数，输入的三个参数分别为创建的网络 *net*、实际输入矩阵 ***P*** 和目标输出向量 ***T***。此函数的作用是训练建立的 BP 神经网络；*sim*(*net*, ***P_test***)为仿真函数，其参数为训练完成的 BP 神经网络 *net*、仿真实验的输入矩阵 ***P_test***，此函数返回仿真实验的实际输出。通过判决门限 0.5 区分正常和异常轨迹，Inf 代表系统最长时间。

算法 2 首先根据用户轨迹的四个属性建立一个输入矩阵，基于轨迹所属类别建立一个目标输出向量（第 1~2 行）；然后根据前文的设计设置隐含层节点数、输出维数、训练次数以及各层传递函数，设定好参数后创建 BP 神经网络 *net*（第 3~8 行）；接着设置最大训练次数，确定网络的最小均方误差和训练时间（第 9~11 行）；最后进行训练网络(第 12 行)，待模型稳定时进行网络仿真（第 13 行）。

复杂度分析：本实验采用一个三层 BP 神经网络，每层神经元数量分别为 4、10、1。使用一个样本（4*1，即一组轨迹属性值）进行前馈计算，要进行两次矩阵运算，两次矩阵乘法（实际上是向量和矩阵相乘）分别要进行 4*10 和 10*1 次计算，由于输入层和最终输出层结点数量（4 和 1）是确定的，可以视为常量，中间仅 1 个隐含层，所以对一个样本的前馈计算次数为 4*10+10*1=50，时间复杂度是 O(1)。反向传播的时间复杂度和前馈计算相同，假设本实验总共有 *m* 个训练样本，每个样本只训练一次，那么训练一个神经网络的时间复杂度则是 O(*m*)。

4 实验

实验环境：本文所有算法均采用 Matlab 2016a 实现，软硬件环境为：Intel (R) Core (TM) 2 Duo 3.3GHz CPU，4GB 内存，Windows 10 操作系统。

4.1 数据集选取

如 3.1 节所述，实验数据来自 GeoLife GPS Trajectories 中的两个用户的轨迹数据集。因轨迹集中采用的时间是格林尼治时间，将格林尼治时间 23:00-00:59，9:00-10:59 换算成北京时间是 7:00-8:59 和 17:00-18:59 此时正是上班和下班高峰时间段。因此本文采用的轨迹数据集是某两个用户在 2008 年 9 月 23 日至 2008 年 12 月 23 日间每天的 23:00-00:59，9:00-10:59 时间段里的轨迹，共 180 条。

4.2 实验结果

本实验利用算法 1 提取该轨迹数据集的长度属性向量 ***Ls***、角度属性向量 ***As***、速度属性向量 ***Vs*** 以及轨迹所用时间向量 ***Dts***；利用算法 2 将轨迹数据集和每条轨迹的四个属性值作为输入数据集，经过训练，得到性能稳定的分类模型；基于此模型，将待识别轨迹数据集进行分类。

4.2.1 BP 神经网络训练及仿真结果

本实验将第一个用户 90 条轨迹的属性数据进行分组，基于不同的分组结果，制定以下五种训练方案。方案 1，将其中 30 条轨迹的属性数据用来训练，余下 60 条轨迹的属性数据用来仿真；方案 2，将其中 40 条轨迹的属性数据用来训练，余下 50 条轨迹的属性数据用来仿真；方案 3，将其中 50 条轨迹的属性数据用来训练，余下 40 条轨迹的属性数据用来仿真；方案 4，将其中 60 条轨迹的属性数据用来训练，余下 30 条轨迹的属性数据用来仿真；方案 5，将其中 70 条轨迹的属性数据用来训练，余下 20 条轨迹的属性数据用来仿真。五种方案产生的 *precision*、*recall*、*F* 度量值所构成的柱状图如图 6 所示。

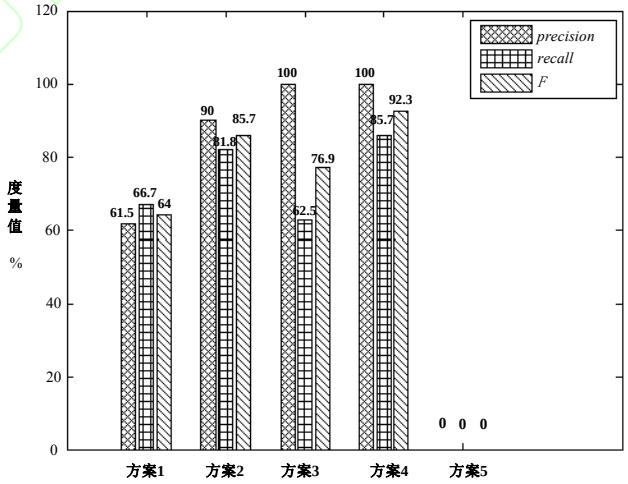


图 6 五种方案度量结果对比

精度（*precision*）可以看作对精确性的度量，即标记为正类的元组（异常轨迹元组）实际为正类所占的百分比，如公式(17)所示：

$$precision = \frac{TP}{TP + FP} \tag{17}$$

其中 *TP* 是指被正确分类的异常轨迹的条数，*FP* 是指被错误标记为异常轨迹的条数。

召回率（*recall*）是对完全性的度量，即正元组（异常轨迹元组）被标记为正类的百分比，如公式(18)所示。

$$recall = \frac{TP}{TP + FN} \tag{18}$$

其中 *FN* 是指被错误标记为正常轨迹的异常轨迹条数。

F 度量是将精度和召回率组合到一个度量中，其表达式如公式(19)所示。

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{19}$$

由图 6 可知，在所有的方案中，方案 4 具有最好的性能，*F* 度量达到 92.3%。因此采用方案 4 对第二个用户的轨迹数据集进行仿真实验，即采用第二个用户的 60 条轨迹用于训练，其余 30 条轨迹用于仿真实验。可以得到本文轨迹检测模型的输出类别和实际类别的对比结果，具体如表 2 所示。

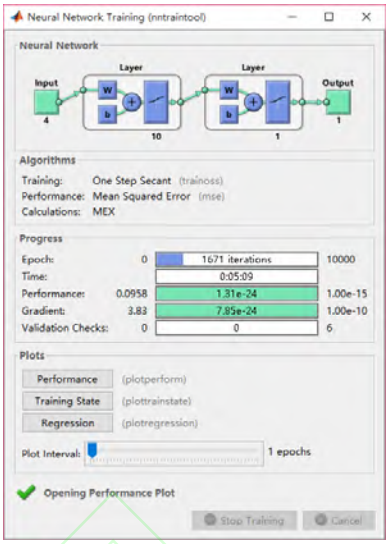
表 2 输出类别与实际类别对比

样本	输出类别	实际类别	样本	输出类别	实际类别
1	1	1	16	1	1
2	1	1	17	1	1
3	0	0	18	0	0
4	1	1	19	0	0
5	1	1	20	1	1
6	0	0	21	1	1
7	1	1	22	1	1
8	1	1	23	1	1
9	1	1	24	1	1
10	1	1	25	1	1
11	0	0	26	1	1
12	1	1	27	1	1
13	0	0	28	1	1
14	1	1	29	1	1
15	1	1	30	1	1

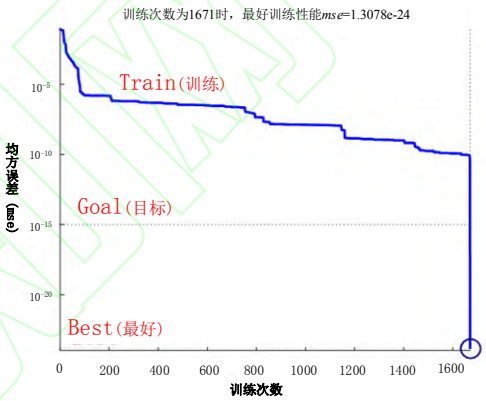
根据表 2 所示的测试结果，本次实验共用 30 个样本，得到模拟结果如实际结果相比较无错误，全部正确分类，即 *precision*、*recall*、*F* 度量值均为 100%。由此可知本训练模型能够很好地检测用户轨迹是否异常。

4.2.2 网络性能分析

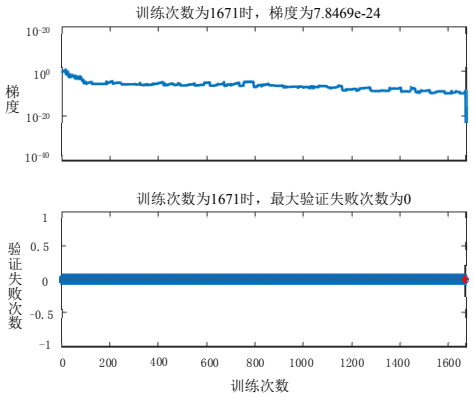
采取方案 4 对第二个用户进行仿真训练的网络性能如图 7 所示。



(a) BP 神经网络综合性能



(b) 最小均方误差(*mse*)



(c) 梯度和验证失败次数变化曲线

图 7 BP 神经网络性能

由图 7(a)的网络综合性能可看出，本实验用的是三层神经网络，第一层是输入层，其输入节点个数为 4；第二层是隐含层，隐节点个数为 10；第三层是输出层，输出节点个数是 1。可以看出本实验采用的 BP 神经网络的训练算法是 One Step Secant (*trainoss*)，BP 神经网络的性能函数使用的是 Mean

Squared Error (*mse*)即均方误差函数，此模型的 *mse* 值为 1.31e-24，Gradient 即梯度为 7.85e-24，Validation Checks 即最大验证失败次数为 0。经过 1671 次迭代，用时 5 分 9 秒，此模型已经稳定，并达到精度要求，可以作为用户轨迹聚类、用户异常轨迹检测的模型。

由图 7(b)可看出，最小均方差(*mse*)在 10^{-11} 时缓慢下降，但是在训练次数增到足够大时，突然下降到 10^{-24}，这是由于本实验采用的训练方法为 *trainoss*，这是一种快速训练方法。因此当网络性能达到一定精度时快速收敛。

由图 7(c)的梯度验证失败次数变化曲线和可以看出，该算法一直保持平稳的梯度下降，验证失败次数一直为 0，在训练次数足够大时梯度呈直线下降。其原因与 *mse* 突然下降的原因相同。

5 结束语

本文提出了一种基于 BP 神经网络的异常轨迹检测方法，主要包括两个阶段。第一阶段，首先收集轨迹数据集，然后对数据进行去除噪音处理，将整理好的数据集上传到百度云的 LBS.云端进行存储，并设计了一个轨迹显示系统，将轨迹数据集显示在百度地图上；第二阶段，首先对轨迹集进行预处理，使用轨迹属性值提取算法，得到轨迹的属性向量（轨迹长度属性、轨迹的角度属性、轨迹所用时间以及轨迹的平均速度），然后将获得的轨迹特征属性作为 BP 神经网络算法的输入层，将轨迹相似度量值作为输出层，训练出稳定的异常轨迹检测模型，利用此模型区分用户轨迹是否属于异常，得到用户的异常轨迹数据。下一步的工作计划是，基于本文的实验结果，提取正常和异常轨迹数据中的时空特征和用户行为特征，从而分析和预测用户行为，并对用户进行如商场消费、旅游路线规划等个性化推荐。

参考文献

[1] Zheng Y. Trajectory Data Mining: An Overview[M]. ACM Transactions on Intelligent Systems and Technology (TIST), 2015, 6(3):29.

[2] Prelipcean A C, Gidofalvi G, Susilo Y O. Measures of transport mode segmentation of trajectories[J]. International Journal of Geographical Information Science, 2016, 30(9):1-22.

[3] Lv M, Chen L, Xu Z, et al. The discovery of personally semantic places based on trajectory data mining[J]. Neurocomputing, 2016, 173: 1142–1153.

[4] Giannotti F, Nanni M, Pedreschi D, et al. Unveiling the complexity of human mobility by querying and mining massive trajectory data[J]. The VLDB Journal, 2011, 20(5):695.

[5] Gupta M, Gao J, Aggarwal C C. Outlier detection for temporal data: A survey[J]. IEEE Transactions on Knowledge and Data Engineering, 2014, 25(1): 1–20.

[6] Hu W, Li X, Tian G, et al. An incremental DPMM-based method for trajectory clustering, modeling, and retrieval[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(5): 1051-1065.

[7] Cai Y, Wang H, Chen X, et al. Trajectory-based anomalous behaviour detection for intelligent traffic surveillance[J]. IET Intelligent Transport Systems, 2015, 9(8): 810–816.

[8] Laxhammar R, Falkman G. Online learning and sequential anomaly detection in trajectories[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2014, 36(6): 1158-1173.

[9] Shen M, Liu D R, Shann S H. Outlier detection from vehicle trajectories to discover roaming events[J]. Information Sciences, 2015, 294: 242-254.

[10] 王嵘冰, 徐红艳, 李波, 等. BP 神经网络隐含层节点数确定方法研究[J]. 计算机技术与发展, 2018,04:1-6.

[11] 刘良旭, 乔少杰, 刘宾, 等. 基于 R-tree 的高效异常轨迹检测算法[J].软件学报, 2009, 20(9): 2426-2435.

[12] 鲍苏宁, 张磊, 杨光. 基于核主成分分析的异常轨迹检测方法[J]. 计算机应用, 2014, 34(07): 2107-2110.

[13] Zhu J, Jiang W, Liu A, et al. Time-dependent popular routes based trajectory outlier detection[C]//Proceedings of the International Conference on Web Information Systems Engineering. Switzerland: Springer International Publishing, 2015: 16–30.

[14] 韩旭. 基于车辆轨迹多特征的聚类分析及异常检测方法的研究[D]. 哈尔滨工程大学, 2014.

[15] 朱燕, 李宏伟, 樊超, 等. 基于聚类的出租车异常轨迹检测[J]. 计算机工程, 2017, 43(2):16-20.

[16] 毛嘉莉, 金澈清, 章志刚, 等. 轨迹大数据异常检测:研究进展及系统框架[J]. 软件学报, 2017, 28(01):17-34.

[17] 蒲春, 孙政顺, 赵世敏. Matlab 神经网络工具箱 BP 算法比较[J]. 计算机仿真, 2006(05):142-144.

[18] 闻新, 周璐, 李翔, 等. MATLAB 神经网络仿真与应用[M]. 北京:科学出版社, 2003.

[19] Zheng Y, Xie X, Ma W Y, GeoLife: A collaborative social networking service among user, location and trajectory[J], Bulletin of the Technical Committee on Data Engineering, 2011, 33(2): 32–39.