

# 基于 RNN 的故障预测算法及在 GIS 上的应用

张子贤<sup>1</sup>, 李敏<sup>2</sup>, 苗红霞<sup>1</sup>, 孙宁<sup>1</sup>

(1. 河海大学 物联网工程学院, 江苏 常州 213022;

2. 国家电网江苏省电力公司检修分公司, 江苏 常州 211100)

**摘要:** 随着工业化程度的提高, 设备的故障预测的重要性日趋提高; 提出了一种基于循环神经网络 (RNN) 的故障预测算法, 通过数据训练, 充分发掘了 RNN 对时间序列数据的拟合能力; RNN 故障预测模型由数据处理模块和神经网络识别模块组成; 在数据处理模块中, 采用数学函数分配的方法建立了 RNN 模型的训练样本和测试样本; 在神经网络识别模块中, 针对当前故障预测技术中异常点难以确定的问题, 应用了一种逐步逼近的神经网络训练方法; 最后利用气体绝缘开关 (GIS) 故障数据对该算法进行了验证, 结果表明, 该方法可以在故障发生前检测到故障发生趋势, 进而实现故障预测, 并且能在逐步训练中确定异常点的位置。

**关键词:** 故障预测; 循环神经网络; 异常点; 气体绝缘开关

## Fault Prediction Algorithm Based on RNN and Its Application to GIS

Zhang Zixian<sup>1</sup>, Li Min<sup>2</sup>, Miao Hongxia<sup>1</sup>, Sun Ning<sup>1</sup>

(1. College of Internet of Things, Hohai University, Changzhou 213022, China;

2. State Grid Jiangsu Electric Power Company Maintenance Branch, Changzhou 211100, China)

**Abstract:** With the development of industrialization, the importance of equipment fault prediction is increasing day by day. A fault prediction algorithm based on Recurrent Neural Network (RNN) is proposed. The fitting ability of RNN to time series data is fully explored through data training. The fault prediction model of the RNN is composed of a data processing module and a neural network recognition module. In the data processing module, the method of mathematical function assignment is used to build the training and test samples of the RNN model. In the Neural network recognition module, due to the problem that the change point in current fault prediction technology is difficult to determine, a neural network training method based on successive approximation is used. At last, the RNN fault prediction algorithm is verified by Gas insulated switchgear (GIS) fault data. The results show that the trend of fault can be detected by this method before it occurs, the fault prediction can be realized, and the change point can be located in the gradual approximation training.

**Keywords:** fault prediction; recurrent neural network (RNN); change point; GIS

## 0 引言

如今, 为了满足设备运行的高效性与可靠性, 针对设备定期维护和故障发生后才维修等传统方法的不足, 故障预测在工程设备的维修与监测中的重要性逐渐增加。故障预测可以将各传感器测得的信息, 通过一定的理论、算法处理后, 推测出故障发展的趋势, 它是比故障诊断更高级的维修手段, 如果能在故障发生之前识别到故障发生的可能, 从而对设备进行排查, 对零件进行更换, 可以大大减少故障风险, 减少因故障造成的损失, 节约资源, 降低经济成本。

由于故障预测是在故障发生之前进行的, 所以往往具有故障幅值微小、故障特性不明显、容易被噪声掩盖等特点, 相对而言难以被检测出, 因此, 对于故障发生前的信号处理与分析是故障预测技术研究的重点。近年来, 在该方面的研究很是火热, 王姝等提出了一种结合多元统计技

术 (MPCA) 与 AR 序列分析的缓变故障预测方法<sup>[1]</sup>, 该方法通过建立 T2 及 Q 统计量的自回归模型, 预测下一批次统计量值, 将所求值与控制限对比, 进而实现渐变故障的预测, 但是该方法假设故障的趋势为线性的, 并且每个统计量的预测过程、预测结果是独立的, 对于非线性、多耦合的数据较难预测; 东南大学陆宁云等提出一种多层预测型贝叶斯网络结构, 根据工程系统自身固有的网络拓扑结构, 构建了多层贝叶斯网络模型, 利用定性趋势分析法将时间信息融入网络节点中, 使得网络具有处理时序信息的能力, 便于进行故障传播机理分析和故障预测<sup>[2]</sup>, 在该算法中需要构建隶属度函数, 通过一定的先验知识去构造隶属度函数, 这些先验知识的精确与否会直接影响该方法的准确性; 文献 [3] 利用深层神经网络 (DNN) 与 PCA 相结合的方法实现了微小缓变故障早期诊断及寿命预测; 文献 [4] 则提出了一种使用深度卷积神经网络 (DCNN) 预测方法, 然而, 上述两种神经网络结构由于没有对应状态变量的神经元, 因此难以存储之前时间的信息。

近年来, 随着计算机计算能力的提高, 人工神经网络的层数越来越多, 深度学习的研究与应用也日益增加。深

收稿日期: 2020-05-06; 修回日期: 2020-05-26。

作者简介: 张子贤 (1995-), 男, 河南新乡人, 硕士研究生, 主要从事机器学习、多模异构信息融合方向的研究。

度学习的本质是特征提取，即通过组合低层次的特征形成更加抽象的高层表示，以达到获得最佳特征的目的。具体应用可以抽象为将原始数据、信号作为低层次的特征送入神经网络，把人们想要得到的信息作为高层表示的过程。在深度学习的应用上，循环神经网络（RNN）主要用于与时间序列相关的数据，比如自然语言的处理、疾病预测等等，并体现出了良好的拟合能力<sup>[5-8]</sup>。故障预测与语言处理和疾病预测相似，各种设备的传感器信号大多为时间序列，当前时刻的信号值与之前的信号值有不可分割的联系，因此在判断设备状态时仅仅关注当前信号是不够的，还要考虑传感器之前信号的状态，这样才能最大程度的挖掘出时间序列数据所蕴含的信息。

另一方面，对于大多数工程系统而言，从系统安装运行到出现异常的正常阶段，监测得到的数据一般比较平稳，没有明显的趋势出现，而一旦发生故障，所监测的数据则会发生剧烈变化，如发动机、气轮机、旋转机械等的振动幅值信号，都具有这样的特点，其故障剧烈程度近似于指数函数。出现异常的时间点一般被称为异常点，对于异常点的确定是故障预测中较为关键的一环，异常点确定过早，会导致误判发生，而过晚则失去了故障预测的意义。关于异常点的检测，文献 [9] 介绍了贝叶斯、广义贝叶斯、极大极小值等方法，然而这些方法没有自学习能力，并且假定数据的分布符合某种规律，比如高斯分布，而现实情况下的数据往往并不符合这些特殊的分布，因此这些方法在实际工程中难以应用，为了解决该问题，在此，应用了一种逐步逼近的神经网络训练方法。

## 1 基于 RNN 的故障预测算法

### 1.1 整体算法流程

基于 RNN 的故障预测算法整体流程如图 1 所示，该算法主要由数据处理模块与 RNN 网络识别模块构成。数据处理模块用于对传感器输出信号进行处理，而 RNN 网络识别模块用于对经过处理的输入信号进行故障预测的操作。

在数据处理模块中，先对所采集的历史数据进行处理；再使用数学函数赋值的方法来构造待训练 RNN 模型的训练与测试样本。

在 RNN 网络识别模块中，首先确定 RNN 模型结构；然后使用 RNN 模型进行异常点检测，为训练样本数据贴上标签；再使用数据处理模块构建的训练样本对 RNN 模型进行训练，调整模型参数，确定故障预测模型；最后将数据输入 RNN 模型对设备进行故障预测。

### 1.2 数据处理模块

通常，使用循环神经网络进行预测前，对于传感器采集的数据都需要先进行特征提取。若构建特征提取模型进行特征提取，不仅费时费力，而且会延缓识别速度，给识别带来困难，因此，这里采用数学函数赋值的方法来构建输入矩阵，对输入数据进行处理。

首先采集设备多种传感器上较长时间内的各种信号数据，作为历史数据，并对这些历史数据进行采样。接着通

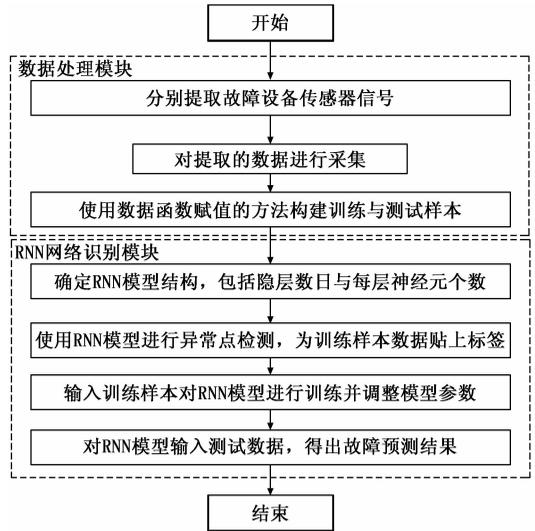


图 1 整体算法流程图

过数学函数赋值的方法构造待训练 RNN 模型的训练样本数据，并使用相同的方法构造测试数据。

数学函数赋值的方法为：

对于装置中的信号，设置  $k$  个传感器采集不同的信号，并将采集到的  $k$  个信号波形函数  $a(1, t)$  到  $a(k, t)$  中的若干采样点提取到一个多维矩阵  $\mathbf{b}$  中，该矩阵的每一列代表一个样本，假设总共有  $n$  个样本。矩阵为：

$$\mathbf{b} = \begin{bmatrix} a(1, x) & \cdots & a(1, x+n) \\ a(1, x+1) & \cdots & a(1, x+n+1) \\ a(1, x+3) & \ddots & a(1, x+n+3) \\ a(2, x) & \cdots & a(2, x+n) \\ a(2, x+1) & \cdots & a(2, x+n+1) \\ \cdots & \ddots & \cdots \\ a(k, x+1) & \cdots & a(k, x+n+1) \\ a(k, x+3) & \cdots & a(k, x+n+3) \end{bmatrix}$$

其中： $\mathbf{b}$  是输入矩阵。函数  $a(1, t)$  到  $a(k, t)$  分别是  $k$  个传感器波形函数， $t$  是历史数据的时间。 $X$  是历史数据的起点。该方法可以根据设备的故障特征确定输入矩阵形式。

### 1.3 RNN 网络识别模块

#### 1.3.1 循环神经网络

由于各种原因，传感器采集到的数据会夹杂噪声信号，而数学方法如贝叶斯和极大极小原理等需要精确的数据才能得出较为可靠的预测结果，因此需要一种不易被噪声信号影响的算法。循环神经网络能够通过学习历史数据来区分噪声信号与故障信号，RNN 可以在迭代的训练过程中，不断调整参数，直到满足精度要求。因此它在解决时间序列问题的实际使用过程中有更高的预测精度。

在传统的前馈神经网络（FNN）中，神经元的信息传递是通过网络层与网络层之间的单向传递完成的，此模式使得 FNN 变得更容易学习，但在一定程度上也削弱了 FNN 模型的能力。在实际情况中，输出不仅和当前输入相

关,也有可能和上一个时刻的输出相关。循环神经网络(RNN)是FNN的一种改进体,是一类具有短期记忆能力的神经网络。RNN形成的环路的网络结构,使得神经元既可以接收到其他神经元的信息,也能接收到本身的信息。所以,相比之下,RNN在处理与时序数据相关的预测问题时有着比FNN更明显的优势。在实际应用中,需要根据故障模型构建合适的RNN网络结构,并需要兼顾快速性、鲁棒性与准确性。

FNN的训练过程通过后向传播算法(BP)来实现,而RNN可以被看作是一个展开的多层FNN,这样就需要依次在时间维上叠加对后向传递的结果。

假设输入RNN的是长度为 $T$ 的序列 $X$ ,并且RNN的输入层、隐含层与输出层的神经元个数分别为 $I$ 、 $H$ 与 $K$ 。

那么RNN前向传播算法公式如(1)~(3)所示:

$$a_h^t = \sum_{i=1}^I \omega_{ih} x_i^t + \sum_{h'=1}^H \omega_{h'h} b_{h'}^{t-1} \quad (1)$$

$$b_h^t = \theta_h(a_h^t) \quad (2)$$

$$a_k^t = \sum_{h=1}^H \omega_{hk} b_h^t \quad (3)$$

其中: $x_i^t$ 是第 $i$ 维输入在时刻 $t$ 的值, $\omega_{ij}$ 表示神经元 $i$ 与 $j$ 的连接权重, $a_j^t$ 和 $b_j^t$ 分别表示神经元 $j$ 在时刻 $t$ 的输入值和激活值, $\theta_h$ 表示神经元 $h$ 使用的激活函数。

对于RNN的时间后向传播算法,首先要定义损失函数对神经元 $j$ 在时刻 $t$ 输入值的偏导数,如式(4)~(6)所示:

$$\delta_j^t = \frac{\partial L}{\partial a_j^t} \quad (4)$$

$$\delta_h^t = \theta'(a_h^t) \left( \sum_{k=1}^K \omega_{hk} \delta_k^t + \sum_{h'=1}^H \omega_{hh'} \delta_{h'}^t + 1_{h'} \right) \quad (5)$$

$$\frac{\partial L}{\partial \omega_{ij}} = \sum_{t=1}^T \frac{\partial L}{\partial a_j^t} \frac{\partial a_j^t}{\partial \omega_{ij}} = \sum_{t=1}^T \delta_j^t b_i^t \quad (6)$$

从RNN的训练过程可以看出,RNN能够对输入数据的序列特性进行“记忆”,因此更适合于解决时间序列问题<sup>[10]</sup>。

### 1.3.2 异常点检测

通常,神经网络训练时需要使用大量的带标签样本,而获取大量有标签样本不仅耗费人力物力,难度也较大。因此,这里使用一种逐步逼近的神经网络训练算法——异常点检测算法来对故障数据样本给定标签。使用该算法能够在短时间内获得样本的标签,能够提高效率。

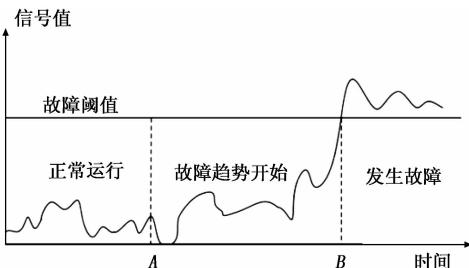


图 2 故障信号变化过程

大部分故障信号变化通常不是突然的,而是渐进的、缓慢的过程,如图2所示,A点称为故障趋势起始点,在此之前,设备状态是健康的,A点表示故障趋势开始的时间,此时故障信号会有微小的变化,但设备仍能正常运行。到B点时,故障信号已经达到故障阈值,并且故障信号的变化开始由平缓趋于剧烈,设备将受到损坏。因此在建立故障模型时,可以使用指数函数来近似表达故障严重程度。基于RNN的故障预测算法针对这一故障变化趋势建立了指数型故障预测模型,并利用异常点检测算法对样本数据进行反复迭代,从而逐次逼近得出故障发生前信号微小变化的时间,进而达到对设备故障进行预测的目的。

实际应用中,在故障信号有微小的变化的时候检测出故障趋势起始点A,接着检测故障真正发生的点B,在 $T=B-A$ 时间段对应的样本贴上 $[0.1-1]$ 的标签。异常点检测流程如图3所示,具体做法为:

1) 确定输入输出向量,建立RNN,确定神经元数目和隐藏层层数。

2) 确定第一次训练的假想异常点 $v_1$ 。

首先根据经验选取异常点时刻 $v$ ,且 $v$ 点在时间上要偏晚,以确保有故障趋势发生, $v$ 点与故障趋势起始点A之间的一段时间称为时间裕量 $M$ ,将数据输入RNN模型进行训练。第一次训练时,时间裕量 $M$ 内的数据不放入RNN模型进行训练。

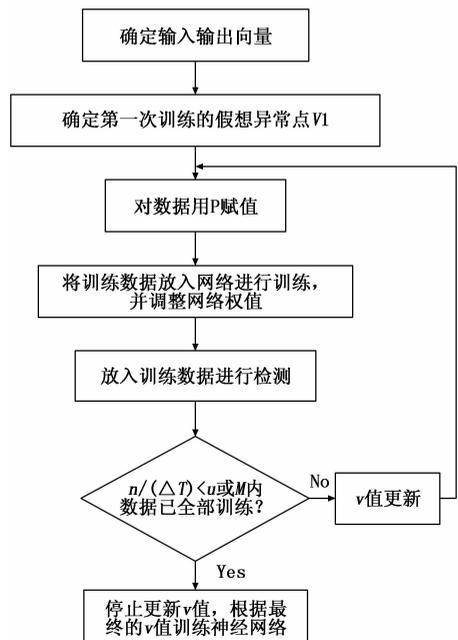


图 3 异常点检测流程图

3) 用函数 $P(t)$ 对数据进行赋值。

从异常点 $v_1$ 到故障发生时刻 $B$ ,按照式(7)利用函数 $P(t)$ 对其进行赋值,故障模型时域为 $T=B-v_1$ 。

$$P(t) = \frac{e^{t-1} - e^{-1}}{1 - e^{-1}} \quad (7)$$

其中:自变量 $t$ 表示出现故障趋势后的时间,从0变化

到  $T$ , 当  $t=T$  时故障发生, 因此  $T$  为从发现故障趋势到故障发生的时间。式 (7) 中的分母起到归一化作用, 分子中的  $e^{-t}$  是为了将起始值设为零。而分子中  $e^{+t-1}$  为指数函数, 该函数的一次导数为递增的, 可以体现故障信号变化趋势的加速度。可见函数  $P(t)$  符合故障信号的变化趋势, 为指数型故障预测模型。

假定选定从  $v_1$  时刻开始, 将异常时刻  $v_1$  的输出值用  $P$  赋值, 并令  $P$  值的变化与实际故障的变化趋势相同, 都是由平缓趋于剧烈的指数函数形式。留出时间裕量  $M$  的原因是真实的异常点  $A$  处于该时间段  $M$  内, 此时并不能给出该段时间内的恰当的  $P$  值, 因为不管  $P$  值超前或者滞后, 都意味着在  $v-A$  这段时间中的数据会得到错误的  $P$  值。

由于猜想的异常点  $v$  是偏晚的, 所以实际的  $P$  值要比公式 (7) 中的结果大一些。按式 (8) 对式 (7) 进行修正:

$$P(t) = \frac{e^{t-v_1} - e^{-1}}{1 - e^{-1}} + \frac{1}{B - v} \quad (8)$$

其中: 自变量  $t$  表示从出现故障趋势开始后的时间,  $B$  表示故障发生的时刻,  $v$  表示最终更新的异常点时刻。

5) 将训练数据放入网络进行训练, 测试并观察输出值, 调整网络权值。

第一次训练完成后, RNN 网络已经具有初步的故障预测能力, 再放入数据, 测试并观察输出值。

RNN 网络此时已经检测出第一次选取的  $v_1$  之前  $\Delta T$  时间段内的故障趋势, 令  $n$  为  $\Delta T$  时间段内检测出故障趋势的数据数目, 在  $\Delta T$  时间段内, 单位时间内有故障趋势的数据数目  $\frac{n}{\Delta T}$  满足  $\frac{n}{\Delta T} > u$  (其中,  $u$  为比例系数且  $u \in (0, 1)$ ) 时, 令  $v_2 = v_1 - \Delta T$ , 此时  $v_2$  已经比  $v_1$  更接近故障趋势起始点  $A$ 。这一步实际含义是当单位时间内有故障趋势的数据数目比例达到  $u$  时, 将异常点提前到当前异常点之前的  $\Delta T$  处。

7) 用重新贴标签的数据对网络进行训练, 重复上述过程, 直到单位时间内有故障趋势的数据数目  $\frac{n}{\Delta T} < u$  或  $M$  时间段内数据全部训练完毕。

8) 停止更新  $v$  值, 此时的  $v$  值已经无限接近于故障趋势起始点  $A$  的时刻, 再根据最终的  $v$  值训练神经网络。可见, 此过程是一个逐次逼近的过程。

检验是否为故障趋势起始点可依据式 (9) 进行, 待定时间点的输出值  $P_d$  若大于设备健康状态时的输出平均值, 则为故障趋势起始点。

$$P_d > s * \frac{\sum_{k=1}^n P_k}{n} \quad (9)$$

其中:  $P_d$  为待定点输出值,  $P_k$  为  $k$  时刻健康状态输出值,  $s$  为比例系数。为防止将健康状态误判为具有故障趋势的点, 通常将  $s$  设定为略大于 1,  $n$  为数据数目。

### 1.3.3 故障预测输出

对于不同的设备, 故障类型的数目不同, 因此预测输

出的矩阵维数也不同。假设用于故障预测的设备有  $k$  种不同的故障类型, 即  $a, b, c, \dots, k$ , 那么最终的预测输出为:

$$y = \begin{bmatrix} a_1 & \cdots & a_i \\ b_1 & \cdots & b_i \\ c_1 & \cdots & c_i \\ \vdots & \ddots & \vdots \\ k_1 & \cdots & k_i \end{bmatrix}$$

其中: 矩阵  $a, b, c$  到  $k$  的行表示预测的当前设备故障类型的概率。  $a, b, c, \dots, k$  的值越接近 1, 故障发生的可能性就越大,  $i$  代表在不同时刻的状态及当时故障预测的结果。

## 2 基于 RNN 的故障预测算法在 GIS 上的应用

本次气体绝缘开关 (GIS) 信号取自某电厂, 取其在 190 天内传感器采集的振动信号与局部放电信号作为历史数据。在数据处理模块中, 使用数学函数赋值方法建立的样本矩阵为:

$$b = \begin{bmatrix} a(1,0) & \cdots & a(1,190) \\ a(1,1) & & a(1,191) \\ a(1,3) & \ddots & a(1,193) \\ a(2,0) & & a(2,190) \\ a(2,1) & \cdots & a(2,191) \end{bmatrix}$$

其中:  $a(1, t)$  表示振动信号,  $a(2, t)$  表示当前局部放电信号。  $a, b, c$  分别代表 3 种不同类型的故障: 载流回路接触不良过热、断路器操作机构故障、盆式绝缘子闪

络, 输出矩阵为:  $y = \begin{bmatrix} a_1 & \cdots & a_i \\ b_1 & \cdots & b_i \\ c_1 & \cdots & c_i \end{bmatrix}$ 。

将振动信号与局部放电信号数据录入 Matlab, 并绘制波形, 如图 4 和图 5 所示。当  $t=120$  天时, 故障未发生, 但开始有微小的信号变化; 当  $t=140$  天时, 故障发生。可以看出, 故障发生后,  $t > 140$  天时, 信号变化较为明显; 当  $120 \text{ 天} < t < 140 \text{ 天}$  时, 特征信号非常微小, 难以识别。

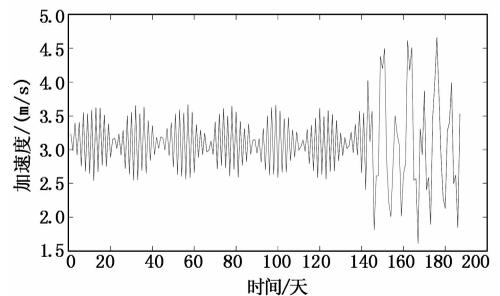


图 4 振动传感器信号

在 RNN 识别模块中, 首先要建立故障预测模型。在最终确定 RNN 网络结构之前, 分别尝试过使用不同层数的 RNN 网络、改变每层的神经元数量等来进行训练。训练结果显示, 网络最终输出的准确性以及响应速度与 RNN 网络的结构关系很大。当每层神经元数量增加到 20 个时, 训练

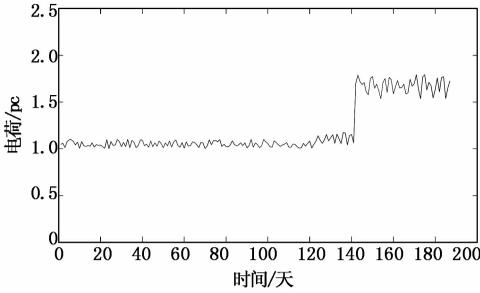


图 5 电流传感器信号

耗时会增加, 然而精度的提高却不甚明显。而当把 RNN 网络层数增加到 8 层时, 网络训练时间上升到 13 s, 训练进行了 27 次重复训练, 训练精度虽有所提高, 但耗时大大增加。因此, 确定 RNN 网络结构时需要综合考虑快速性与准确性。

因为振动与电流信号并不非常复杂, 信号类型不多, 因此选用的 RNN 结构相对简单。选用的 RNN 由一个输入层、四个隐含层和一个输出层构成。

将时间在 40~135 天内的数据作为异常点可能存在的区间, 此区间的的核心数据不放入网络训练。对故障发生前 5 个天的数据进行赋值, 将赋值后的数据放入网络训练后, 重新将所有数据放入网络测试, 训练效果如图 6 所示。

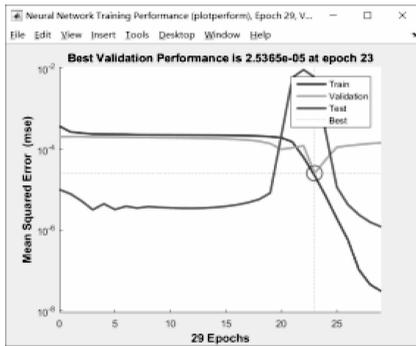


图 6 数据重新标签前训练效果

由图 6 可见, 整个 RNN 网络的训练过程一共耗时 1 s, 训练期间重复了 29 次。在第 23 次训练时, 得到了最好的训练效果。训练后得到的结果如表 1 所示。

表 1 时间裕度  $M1=5$  时的输出

时间 故障	131	132	133	134	135
a	0.049	0.039	0.039	0.047	0.367
b	0	0	0	0	0
c	0	0	0	0	0
时间 故障	136	137	138	139	140
a	0.508	0.744	0.967	0.979	0.976
b	0	0	0	0	0
c	0	0	0	0	0

由表 1 可知, 健康状态的输出均值为 0.029 4, 取比例

系数  $s=1.3$ , 可以看出时间在 131~135 的 5 个数据中, 有 3 个满足式 (9), 于是将时间在 131~140 的数据重新标签, 并用于网络训练。重新将所有数据进行测试, 训练结果如图 7 所示。

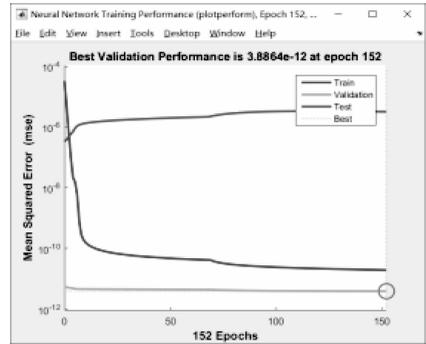


图 7 数据重新标签后训练效果

由图 7 可见, 此次训练的最小均方误差比第一次小, 并且在训练过程中不断收敛, 取得了较好的效果。训练结果如表 2 所示。

表 2 时间裕度  $M2=10$  时的输出

时间 故障	126	127	128	129	130
a	0.053	0.403	0.186	0.062	0.687
b	0	0	0	0	0
c	0	0	0	0	0
时间 故障	131	132	133	134	135
a	0.401	0.318	0.389	0.438	0.534
b	0	0	0	0	0
c	0	0	0	0	0

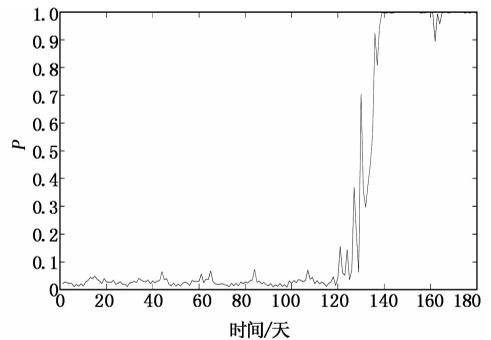


图 8 最终预测结果

此时 126~130 天的输出均满足式 (9)。将数据重新标签后, 再次训练并测试网络, 直到找出真正的异常点时刻  $T=120$ 。最终的预测结果见图 8, 横坐标是时间, 纵坐标是  $P$  值。可以看出, 在  $t < 120$  时, 可以认为  $P$  为 0; 当  $t > 120$  时 (故障趋势开始),  $P$  值开始变大, 直到  $t=140$  (故障发生时),  $P$  值到达 1, 网络能很好地分辨出  $t=120$  前后的健康状态和之后的异常状态。

(下转第 36 页)