### 工艺与装备

# 自动化包装生产线电机无传感器驱动故障诊断

吴强<sup>1</sup>,张伟<sup>1</sup>,岳秀清<sup>2</sup>

(1.上海理工大学 光电信息与计算机工程学院,上海 200093;2.中国北方电子设备研究所,北京 100083)

摘要:目的 为了解决自动化包装生产线针对电机驱动故障诊断复杂化和精度低的问题,提高复杂生产 环境下电机运行的稳定和人员的安全,提出一种基于 XGBoost 特征重构和神经网络预测电机驱动故障 的精准预测方法。方法 首先通过 XGBoost 算法运用一部分训练数据构建特征树,随后将剩余训练数据 输入 XGBoost 算法得到重构的特征,然后再运用 One-hot 编码,将重构特征映射到欧式空间,进一步放 大特征的差异,最后输入经过参数调整的神经网络模型中完成故障预测。结果 相较于未经 XGBoost 特 征构建的神经网络模型,文中提出的结构在数据测试集随机分割的验证集和测试集上均取得了接近 100%的分类精度,验证了模型的有效性和稳定性。结论 较好地实现了针对自动化包装生产线电机驱动 故障的无传感器高精度诊断,有利于提高复杂生产环境下的电机稳定性和人员安全性。 关键词:故障诊断;电机驱动;XGBoost;特征构建;神经网络 中图分类号:TP391 文献标识码:A 文章编号:1001-3563(2021)11-0182-09

**DOI:** 10.19554/j.cnki.1001-3563.2021.11.027

#### Fault Diagnosis of Sensor Less Motor Drive in Automatic Packaging Production Line

WU Qiang<sup>1</sup>, ZHANG Wei<sup>1</sup>, YUE Xiu-qing<sup>2</sup>

(1.School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China; 2.China North Institute of Electronic Equipment, Beijing 100083, China)

**ABSTRACT:** In order to solve the problem of complex and low accuracy of motor drive fault diagnosis in automatic packaging production line, and to improve the stability of motor operation and personnel safety in complex production environment, a precise prediction method of motor drive fault based on XGBoost feature reconstruction and neural network prediction is proposed. The method first uses a part of the training data to construct a feature tree through the XGBoost algorithm, and then inputs the remaining training data into the XGBoost algorithm to obtain the reconstructed features. Consequently, using the One-hot encoding to map the reconstructed features to the Euclidean space to further amplify the difference in features. Finally, the obtained features are input into the neural network model with parameter adjustment to complete the fault prediction. Compared with the neural network model constructed without XGBoost features, the structure proposed in this paper achieves nearly 100% classification accuracy on the verification set and the test set of the data test set randomly divided, which verifies the effectiveness and stability of the model. The sensorless high-precision diagnosis of the motor drive fault in automatic packaging production line is realized, which is beneficial to improve motor stability and personnel safety in complex production environment.

KEY WORDS: fault diagnosis; motor drive; XGBoost; feature construction; neural network

- 基金项目: 国家自然科学基金(11502145)
- 作者简介:吴强(1995—),男,上海理工大学硕士生,主攻深度学习、数据挖掘。

通信作者:张伟(1981-),男,博士,上海理工大学副教授,主要研究方向为最优化与最优控制。

收稿日期: 2020-10-21

基于过程自动化的包装生产线在计算机技术发 展的不断推动下变得越来越复杂,作为包装机械动力 源核心电机驱动产生微小的故障就可能导致整条生 产线的崩溃。因为需要面对的生产任务与主题很广, 因此解决方案常常被分开处理,从而增加了解决此类 问题的复杂性。在大型自动化生产工程中,电机驱动 故障的尽早诊断,对于整体上及时有效诊断系统故障 是一个十分关键的问题。及时的电机驱动故障诊断与 隔离对于系统的经济性和人员的安全性都是十分必 要的。

随着机器学习和深度学习的发展,越来越多的故障诊断场合融合使用机器学习和深度学习工具,大大地提高了检测速度与精度。范勇等<sup>[11</sup>运用改进经验模态分解将自动机驱动信号分解为若干不同的固有模式分量,然后运用一系列特征提取算法提取主要信息分量,最后输入 SMO 神经网络完成故障诊断。高佳豪等<sup>[2]</sup>使用自适应噪声消除技术将齿轮箱振动信号分离为周期性信号分量,从而抑制强干扰成分,融合高效的特征提取神经网络 1D-CNN,实现了在强干扰状态下的齿轮箱故障检测模型。朱浩等<sup>[3]</sup>运用基于注意力机制的 Inception-CNN 实现了对滚动轴承模型的故障分类,得益于注意力机制对不同特征的权重赋值和 CNN 对特征的高度敏感,使得故障分类精度得到大大提升。

包含以上案例的绝大多数故障诊断中,基于机器 学习算法的故障诊断对有限样本下特征的要求极高, 特征的数量和质量决定了机器学习的上限<sup>[4]</sup>。对于复 杂的自动化系统故障分类也依赖于大量的特征数据, 特征的筛选构建也依赖于专业的工程背景知识,特定 生产环境下其泛化能力也就十分有限。如果能有一种 无需依赖专业的背景知识就能对特征进行有效筛选 整合并实现高精度的故障检测,则在实际工程应用中 具有广阔的应用背景。

针对以上问题,文中提出一种基于 XGBoost 特 征提取和神经网络分类的无传感器电机故障诊断方 法,研究对象是一台某食品包装生产线的同步电动机 和轴承、齿轮箱等附属部件,其代表一个自动化包装 生产线的典型部件。驱动器的损坏会引发一连串问 题,增加故障的成本和风险<sup>[5]</sup>,而此类应用的状态监 测通常需要额外的传感器。文中直接使用电机的相电 流来表征整个装置的健康状态,除了电机控制的内置 电流传感器外,测量不涉及任何具体生产场景下的传 感器,增加该方法的应用范围。

### 1 XGBoost 算法基本理论

XGBoost(Extreme Gradient Boosting)是由陈天 奇等<sup>[6]</sup>主导开发的高效率、高速的 Boosting 算法,通

过不断地进行特征分裂生成新树来进行残差拟合多 个弱分类器,最后将弱分类器累加成一个强学习器。 相较于传统 GBDT (Gradient Boosting Decision Tree) 算法仅仅提取目标函数的一阶导数信息,XGBoost 对目标函数进行泰勒展开,引入其二阶导数信息,提 升了算法的精度与速度。同时,XGBoost 算法通过在 损失函数上添加正则项来控制模型的复杂度来降低 应用模型的过拟合。XGBoost 在回归任务上的推导细 节如下所述<sup>[7]</sup>。

对于一个训练集 D,包含 n 个样本,其中每个样本含 d 个特征;  $x_i$ 为第 i 个样本。弱分类器是 CART 回归树 (Classification and Regression Tree),其假设树为二叉树,不断地将特征进行分裂。设当前树基于第 j 个特征进行分裂,对于分裂阈值 s,小于 s 样本分到左子树,大于 s 的样本分到右子树;

$$R_{1}(j,s) = \{x \mid x^{(j)} \leq s\}$$

$$R_{2}(j,s) = \{x \mid x^{(j)} > s\}$$
(1)

CART 在该特征维度对样本空间进行划分,计 算在当前分裂策略下的目标函数值,典型的目标函 数为:

$$\sum_{x_i \in R_m} \left( y_i - f(x_i) \right)^2 \tag{2}$$

对于最优切分特征 j 和切分点 s 的求解转化到式 (3)目标函数,在 j, s 固定时, c 为该划分区域下 预测值的平均值:

$$\min_{j,s} \left[ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right]$$
(3)

遍历所有特征值和所有切分点,就能找到最优的 *j*和 *s*,从而得到1棵回归树。

XGBoost 不断地通过特征分裂添加树,每添加1 棵树本质上就是生成1个函数去拟合上一次预测的 残差。通过训练 K 棵树,每当输入1个样本,对应 的,每棵树最终都会落到自身的某个叶子节点,每 颗叶子节点对应1个分数,将所有分数相加即为最 终预测值:

$$\hat{y}_i = \sum_{k=1}^k f_k(x_i)$$
 (4)

XGBoost 的目标函数为:

$$O = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k)$$
(5)

其中:

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2$$
(6)

式中: Ω为正则项; T为叶子节点数量; ω为叶 子节点的分数。常数 y 控制叶子节点的数量, λ 控制 叶子分数值不至于过大。

前面提到,新树要拟合前一次的残差,当生成 t

棵树后,预测值可以写成:

$$\hat{y}_{i}^{(t)} = \hat{y}_{i}^{(t-1)} + f_{t}(x_{i})$$
 (7)  
同时,目标函数可以改写成:

$$O^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}^{(t-1)} + f_t(X_i)) + \Omega(f_t)$$
(8)

将目标函数转化成与 f<sub>t</sub> 有关的目标函数最小化问题。XGBoost 利用其泰勒展开去近似目标函数的值,因此,目标函数可以近似为:

$$O \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}_i^{(t-1)}) + g_i f_t(X_i) + \frac{1}{2} h_i f_t^2(X_i)] + \Omega(f_t)$$
(9)

因为前 t-1 棵树的预测分数与 y 的残差对目标函数的优化不影响,可以舍去,因此目标函数简化为:

$$\tilde{O}^{(t)} = \sum_{i=1}^{n} [g_i f_i(X_i) + \frac{1}{2} h_i f_i^2(X_i)] + \Omega(f_i) \quad (10)$$
  

$$\vec{x} (9) \ \pi \vec{x} (10) \ \theta \ g_i, h_i \ \beta \ B \ \delta :$$

$$g_{i} = \frac{\partial l(y_{i}, y^{(t-1)})}{\partial y^{(t-1)}}, h_{i} = \frac{\partial l(y_{i}, y^{(t-1)})}{\partial^{2} y^{(t-1)}}$$
(11)

对于每棵树,输入样本最终会落到某个叶子节点上,因此可以将统一叶子节点的样本重组起来:

$$O^{(t)} \simeq \sum_{i=1}^{n} [g_i f_i(X_i) + \frac{1}{2} h_i f_i^2(X_i)] + \Omega(f_i) =$$

$$\sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma T + \lambda \frac{1}{2} \sum_{j=1}^{T} \omega_j^2 = (12)$$

$$\sum_{j=1}^{T} [(\sum_{i \in I_j} g_i) \omega_j + \frac{1}{2} \left( \sum_{i \in I_j} h_i + \lambda \right) \omega_j^2] + \gamma T$$

通过改写,将目标函数改写成关于叶子节点分数 ω的一元二次函数,解得:

$$\omega_j^* = -\frac{G_j}{H_j + \lambda}$$

$$O = -\frac{1}{2} \sum_{j=1}^T \frac{G_j^2}{H_j + \lambda} + \gamma T$$
(13)

式中:  $\omega_j^*$ 为每个叶子节点的最优预测分数; *O* 为最小损失。XGBoost 对分裂结果打分,若分裂后损失值减小,就证明分裂是正确的,分裂前后增益为:

$$G = \frac{1}{2} \left[ \frac{G_{\rm L}^2}{H_{\rm L} + \lambda} + \frac{G_{\rm R}^2}{H_{\rm R} + \lambda} - \frac{(G_{\rm L} + G_{\rm R})}{H_{\rm L} + H_{\rm R} + \lambda} \right] - \gamma \quad (14)$$

式中: 
$$\frac{G_{L}^{2}}{H_{L}+\lambda}$$
为左子树分数;  $\frac{G_{R}^{2}}{H_{R}+\lambda}$ 为右子树

的分数;  $\frac{(G_{\rm L}+G_{\rm R})}{H_{\rm L}+H_{\rm R}+\lambda}$ 为不分割可以取得分数,

XGBoost 在构建弱分类器 CART 时,从根节点起遍历 所有特征,选出其中增益最高的分裂点,依次分裂到 最大深度停止,随后依据残差构建下一颗树,直到生 成所有的树。

# 2 XGBoost 特征提取

在大多数实践中,通常需要从原始数据中观察 出数据潜在结构,依靠专业背景去对数据特征之间 做相关运算,实现对数据的分割与结合。这样一来, 首先需要分析者拥有相关专业背景知识,其次还需 要分析者拥有大量的实战经验去形成构建直觉。在 拥有前两大前提下,对构建的新特征还需要验证 其有效性进行进一步的筛选,这也需要极大的工 作量。

前面提到 XGBoost 在数据输入后通过对特征分 割选择最终会落到每棵树的某个叶子节点,这实质上 是一种特征选择的过程。XGBoost 每棵树分割的特征 和分割点采用贪心选择,即穷举所有可能,这事实上 就是替代了人工特征选择。文中选用每颗树最终的叶 子节点替代原始特征,自动实现了特征构建,通过 One-Hot 编码和控制树的数目提升数据维度,输入神 经网络进行分类工作。具体操作见图 1。



图 1 XGBoost 特征构建树 Fig.1 XGBoost feature construction tree

假设有 K 棵树,每棵树叶子节点数最大分裂数 为b,则经转化后一条输入特征在每棵树落到的叶子 节点数都是介于 0,b之间的数,形成其度为 K 的行向 量。如此循环将所有原始特征输入 XGBoost 构建树, 完成原始特征重构,重构后的特征为:

$$\boldsymbol{M} = \begin{pmatrix} a_{11} & \dots & a_{1K} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nK} \end{pmatrix}$$
(15)

对于深度学习分类和回归算法来说,特征之间的 距离或相似度计算十分重要,经过特征重构后的特征 表征的是每棵树的节点值,每条输入特征之间的差异 也仅仅是叶子节点位置的差异,其对于故障分类的隐 藏差异是不显著的,为了更合理地表征不同输入特征 之间的差异以及更好地挖掘其中的隐式信息,还需 要对特征矩阵进行进一步的哑变量编码——One-hot 编码<sup>[8]</sup>。

One-hot 编码是将整数型特征变量转化为二进制向量表示的方法,其主要构成方法是将重构后的特征 值正整数  $a_{nK}$ 视为一个索引值,除索引值位数置1外, 其他位均置0。One-hot 索引的向量长度则由 XGBoost 每棵树能分裂的最大叶子结点序列值 b 决定,假 定 b=3,  $a_{nK}=2$ , One-hot 编码则将  $a_{nK}$ 转化成索引为2, 长度为3的稀疏向量(0,1,0)。构建好的稀疏矩阵见式 (16)。

$$\widetilde{\boldsymbol{M}} = \begin{pmatrix} 0 & 1 & 0 & 0 \dots & 1 & 0 \\ 1 & 0 & 0 & 0 \dots & 0 & 0 \\ \vdots & & & & \\ 0 & 0 & 1 & 0 \dots & 0 & 0 \end{pmatrix}$$
(16)

经过 One-hot 编码,将原先的位置离散值拓展到 了欧式空间,其每维的特征也可以视为连续的特征变 化,更好地发掘了经 XGBoost 重构后的特征隐式信 息,进一步扩充了数据的特征,能更好地体现神经网 络的计算优势。此外,经过 One-hot 处理后,虽然将 原有特征转化为了一个稀疏矩阵有可能使得部分神 经元不能得到训练,事实上在 $\widetilde{M}$ 中,不存在全为 0 的列向量,稀疏化的矩阵的 0 值在某方面起到了类似 于 Drop-out 的降低过拟合的作用,后续章节的实验验 证也证明了这一点。

## 3 神经网络

#### 3.1 神经网络基本结构

文中采用基本深度神经网络 DNN(Deep Neural Network)作为后层分类器,其整体结构由神经元构成的输入层、隐藏层和输出层组成<sup>[9]</sup>。神经元作是神经 网络的基本处理单元,又称感知器。利用神经元构成 相应的输入的特征向量通过隐含层变换达到输出层, 在输出层得到分类结果为:

$$f(x) = s(wx + \theta) \tag{17}$$

神经网络的预测的输入层、隐含层和输出层节点数的选取取决于研究对象及其复杂程度。输入层节点数为经前序处理的数据维数,网络隐含层节点数的确定通常按照式(18)来进行试算<sup>[10]</sup>。

$$n_1 = \sqrt{n + m} + a \, \, \, \, \, \pi n_1 = \log_2 n \tag{18}$$

式中: *m* 为输出神经元数; *n* 为输入神经元数量; *a* 为1到10之间的常数。

文中最终任务是得到1个分类值,神经网络分类 任务需要将分类值向量化,最常用的是 One-hot 对标 签样本来进行分类编码,而输出层的结果为1个介于 0和1的浮点数,数值最大的神经元即为神经网络认 为的分类结果。浮点数由上层隐藏层经 Softmax<sup>[11]</sup>计 算得到, Softmax 函数表达式为:

$$S_i = \frac{e^i}{\sum_j e^j} \tag{19}$$

神经网络基本结构见图 2。



图 2 神经网络示意 Fig.2 Schematic diagram of neural network

#### 3.2 神经网络学习过程

多个层相互链接组成网络结构,通过各神经元经激活函数(Activate function)随机权重计算将输入值 映射为输出值,通过损失函数(Loss function)将预 测值与真实样本进行比较得到损失值,以此反向传播 修改各层权重。最后通过优化器逐步缩小损失值<sup>[12]</sup>。 具体流程见图 3。



图 3 神经网络学习流程 Fig.3 Neural network learning process

文中作为多分类问题,损失函数一般采用交叉熵 (Cross-entropy),见式(20)。

$$C = -\frac{1}{n} \sum_{x} [y \ln \sigma(z) + (1 - y) \ln(1 - \sigma(z))]$$
(20)  
式中:  $\sigma(z)$ 为上层输出值。

# 4 实例分析

#### 4.1 数据简介

文中研究数据采集于某食品包装生产线的同步 电动机和轴承、齿轮箱等附属部件,此类应用的状态 监测通常需要额外的传感器。文中直接使用电机的相 电流来表征整个装置的健康状态,除了电机控制的内 置电流传感器外,测量不涉及任何传感器。数据为内 置电流传感器采集两相电流数据经过经验模式分解, 通过统计学方法计算基本均值、峰度、偏差等统计指 标,采集后按照文献[13]方法处理后得到 48 个特征, 共计 58 509 条数据,数据共有 11 种故障类型,每条 特征数据对应一种故障类型。

将原始数据保留一部分用作 XGBoost 特征树的 构建,将剩余的数据自行分为训练集和测试集两部 分,其中在训练集中神经网络的每次迭代中也需要随 机抽取一定数量数据作为每次迭代的验证集,实时监 测训练效果。

训练结果采用准确率和宏 F1 值(Macro-F1)衡 量其有效性,准确率计算式为:

$$\gamma(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} \Omega_i$$
(21)

$$\Omega_{i} = \begin{cases}
1 \quad y_{i} = \widehat{y_{i}} \\
0 \quad y_{i} \neq \widehat{y_{i}}
\end{cases}$$
(22)

Macro-F1 由  $F_1$  值求全局平均得到,对于一个二 分类问题  $F_1$  值计算式为:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R} \tag{23}$$

$$P = \frac{T_P}{T_P + F_P}, R = \frac{T_P}{T_P + F_N}$$
(24)

 $T_{\rm P}$ ,  $F_{\rm P}$ ,  $F_{\rm N}$ 由表1确定。

ストレジの古文 Tab.1 Parameter meaning				
预测结果	真实结果			
	真实	错误		
真实	$T_{ m P}$	$F_{\mathrm{P}}$		
错误	$F_{ m N}$	$T_{ m N}$		

主1 会粉会ツ

从而由式(23)可以得到 *n* 种不同类别的 Macro-F1 值为:

$$M_{F_{i}} = \frac{1}{n} \sum_{i=1}^{n} F_{i}$$
(25)

4.2 实验流程与数据分割

实验流程见图 4。



图 4 整体实验流程 Fig.4 Overall experimental process

文中随机抽取 15%(8777条)的数据作为最终 的测试集。好的特征构建结果更有利于充分发掘特征 之间的隐式信息,因此保留的 85%数据中,使用 75% (37 299条)数据用来训练 XGBoost 特征树;25%(12 433条)数据经 XGBoost 特征转换和 One-hot 编码后 进行神经网络训练,在这批数据中,每次神经网络迭 代随机抽取其中的 15%数据作为实时的验证监测,具 体数据分割数量及占总数比例见表 2。

表 2 数据分割比例 Tab.2 Data segmentation ratio

	8	
名称	数量	比例(占总数据)/%
测试集	8777	15
训练集	49 732	85
XGB 训练	37 299	64
DNN 训练集	12 433	21

#### 4.3 模型的参数调整

为了使故障预测模型尽量准确,在 XGBoost 特 征树的训练中,考虑到数据的量级和复杂程度将 XGBoost 树数量设置为 50,其他参数为默认参数, 最终得到一个分类正确率为 96.7%的 XGBoost 模型, 基于默认参数的参数微调可能会对 XGBoost 分类结 果有一定影响,但其对特征的构建没有太大影响,且 后续神经网络结果影响甚微。 神经网络的各参数对其能否对特征进行有效拟 合至关重要。一般来讲,每层神经元的数量通过 3.1 节经验公式试算确定,细微的个数差异对于结果的影 响不大<sup>[14]</sup>。通俗来讲,只要神经元的数量足以塞进绝 大多数上层信息,神经网络就能够进行训练。通过 对测试集数据的预测可以直观地评价模型的好坏。 另外,神经网络训练集和每次迭代随机抽取的验证 集的迭代精度变化也可作为该组参数的好坏评价依 据<sup>[15]</sup>。文中研究优化器、学习率、激活函数参数的 选取和调整。

1)优化器。优化器(Optimizer)<sup>[16]</sup>是神经网络 误差反向传播对神经元权值更新的关键,目前常见的 优化器都是梯度下降(Gradient Descent)以及基于他 的变体构成的,如:随机梯度下降(Stochastic Gradient Descent,GD), Momentum, Adagrad, Adadelta等, 基本梯度下降公式为:

$$\theta_{t+1} = \theta_t - \eta \frac{\partial L}{\partial \theta_t} \tag{26}$$

式中: θ 为神经元权值; η 为学习率。各优化器 集中在对于误差偏导数求解的各种变体上, 一般是引 入误差的一阶动量(梯度相关函数)和二阶相关动量 (梯度平方相关函数)通过相关组合来优化, 优化的 一般形式为:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{V}}$$
(27)

式中: *m* 为一阶动量; *V* 为二阶动量。这里研究 常用的 3 种优化器 SGD, RMSProp 和 Adam 对故障 预测精度的影响。使用经 XGBoost 特征构建的输入 数据按 4.2 节划分数据集,通过每次迭代验证集的 精度观察优化器的有效性,迭代 20 次,比较结果见 图 5。



如图 5 所示, 3 种优化器在正确率上都能得到提升, 但是 SGD 上升速度非常缓慢, 另外两者最终都能达到理想的收敛结果, 但对比 RMSProp, Adam 明

显波动更小,收敛更快。综合而言,使用 Adam 作为 文中的优化器。

2)学习率。选定 Adam 作为优化器,下面就有 必要对 Adam 的内置参数做一定调整,其中最主要的 就是上文提到的学习率为 η,学习率大损失函数更容 易达到最低和跳出局部最低点,但也容易跳过最低 点;学习率过小则容易引发落入损失函数的局部最低 和学习过慢<sup>[17]</sup>。文中在实验时对学习率进行了微调, 选取 Learning-rate 为 0.01, 0.001, 0.005, 0.0005, 4 组量级数据的对比结果见图 6。



如图 6 所示,除 *l*<sub>r</sub>=0.01 外几种学习率对应下的 准确率都收敛到了理想的值,综合而言 *l*<sub>r</sub>=0.005 准确 率波动最小,因此选择 0.005 为文中优化器 Adam 的 学习率。

3)激活函数。激活函数为神经元的输出函数, 神经网络希望通过激活函数在不失真的条件下将输 入信息往非线性空间进行映射,这样一来,随着多层 神经元的交叉组合就能尽可能模拟出各种非线性关 联。最常见的激活函数有 sigmoid 函数和 Relu 函数等 2 种,其中 sigmoid 函数对于输入要求较高,输入必 须为 0~1 之间的浮点数,对于该实验而言,经过 XGBoost编码后为0,1稀疏矩阵,显然不满足 sigmoid 的输入条件<sup>[18]</sup>,因此文中选用 Relu 作为激活函数, 具体对比见图 7。

如图 7 所示,采用 sigmoid 为激活函数模型最终 拟合到一个极低的分类精度,而采用 Relu 函数取得 了较为理想的拟合结果。

#### 4.4 实验对比

为了验证文中提出的 XGBoost 特征提取算法对 于故障特征的有效性,使用同样的训练集、测试集数 据和神经网络结构,分别将原始特征和经过 XGBoost 特征重构后的特征输入神经网络,在每次迭代随机抽 取等额的验证集数据,从训练集和验证集 2 个指标观 察训练情况,结果见图 8。







如图 8 所示, 未经 XGBoost 特征构建的网络对 于驱动故障识别正确率确实在提升,但是每次迭代随 机抽取的验证集数据在模型上的表现有较大的波动, 且相比较而言, 未经 XGBoost 特征重构后的数据神 经网络需要更多的迭代次数才能收敛,且收敛结果也 远远比不上经过 XGBoost 特征重构的。如图 9 所示, 经过 XGBoost 特征重构后的特征充分挖掘了原始特 征中的隐式信息,神经网络仅使用约3次迭代驱动故 障监测的精度就收敛到了 100%附近,远远优于未经 XGBst 特征重构的 150 次迭代和 90%的识别精度。

将测试集数据分别输入2个模型中,比较驱动故 障的 11 种故障类型的预测准确率,比较结果见表 3。

如表3所示,在测试集上的表现进一步证实了对 训练集和验证集走势的判断,经过 XGBoost 特征构 建后,神经网络稳定出色地完成了全部 11 种故障的 预测,总体正确率接近 100%。未经 XGBoost 特征构 建的神经网络,虽然在测试集和验证集上最终正确率 接近 90%, 由于模型的不稳定性, 最终模型在测试集 上的正确率仅 81.6%。综上所述, XGBoost 特征构建



图 9 XGBoost 特征构建神经网络构建 Fig.9 Neural network training process with xgboost feature construction

表 3 各故障类型预测结果 Tab.3 Prediction results of various fault types

故障 类型	数量	XGB 特征重构神经网 络正确率/%	神经网络正 确率/%
1	741	100	81
2	793	100	58.6
3	784	100	97.7
4	812	100	99.9
5	820	99.88	85.9
6	744	100	86.8
7	809	100	100
8	811	100	79.8
9	855	99.77	93.3
10	816	99.63	96.3
11	792	100	99.8
总计	8777	99.993	81.577

充分发掘了原特征之间的隐式信息,使得神经网络结 构在该实验驱动故障预测上相较于未经特征构建的 相同神经网络结构表现提升十分明显。

为验证文中方法与常规故障预测机器学习算法 的差异,在相同训练和测试样本下,采用 Logistic 回 归、朴素贝叶斯(Naive Bayes)、K 近邻(KNN)、支持 向量机(SVM)、随机森林(Random forest)以及 XGBoost 分类做相同的建模训练。这几种算法的预测 正确度以及 Macro-F1 值对比见表 4。

如表 4 所示, 文中分类算法相对于常见的分类算 法在电机驱动故障分类上取得了最优的结果,分类精 度远远高于其他算法。Random forest 和 XGBoost 算 法取得了超过 90%正确率的分类精度和接近 0.9 的 Macro-F1 值,这也说明了树类算法对复杂隐式特征

表 4 各分类算法预测精度对比 Tab.4 Comparison of prediction accuracy of various classification algorithm

	8	
算法	正确率/%	Macro-F1 值
XGB+DNN	99.99	0.99
Logistic	75.38	0.70
Naive Bayes	76.52	0.71
KNN	80.27	0.75
SVM	76.74	0.72
Random forest	90.63	0.87
XGBoost	96.68	0.96

的高解构能力,且 XGBoost 取得了仅次于文中方法 的正确率和 Macro-F1 值,同样也验证了文中方法的 有效性。

# 5 结语

文中提出了一种基于 XGBoost 特征构建方法的 神经网络模型对自动化包装生产线电机驱动故障的 诊断方法, XGBoost 算法充分挖掘了原有特征的隐式 信息,并通过 One-hot 二次编码将重构的特征映射到 欧式空间,进一步放大了特征之间的差异,也扩充了 特征数量,从而更有利于发挥神经网络的计算优势。 最终在无传感器电机驱动 11 类故障诊断中取得了接 近 100%的正确率结果,验证了算法模型的有效性及 稳定性。由于该实验除了采集相电流外没有多余的传 感器,因此对于未来生产自动化场景下特征质量要 求较高或特征数量不足的自动化包装、印刷生产等 复杂故障诊断的场景,该算法模型拥有广阔的通用 应用价值。

#### 参考文献:

[1] 范勇, 王鹏. 基于改进 EMD 与 SOM 神经网络的自动机驱动机构故障诊断研究[J]. 国外电子测量技术, 2020, 39(8): 5—10.

FAN Yong, WANG Peng. Research on Fault Diagnosis of Automata Driving Mechanism Based on Improved EMD and SOM Neural Network[J]. Foreign Electronic Measurement Technology, 2020, 39(8): 5–10.

[2] 高佳豪, 郭瑜. 基于 SANC 和一维卷积神经网络的 齿轮箱轴承故障诊断[J]. 振动与冲击, 2020, 39(19): 204—209.

GAO Jia-hao, GUO Yu. Gearbox Bearing Fault Diagnosis Based on SANC and One-Dimensional Convolutional Neural Network[J]. Journal of Vibration and Shock, 2020, 39(19): 204-209.

 [3] 朱浩, 宁芊. 基于注意力机制-Inception-CNN 模型的滚动轴承故障分类[J]. 振动与冲击, 2020, 39(19): 84—93.

ZHU Hao, NING Qian. Rolling Bearing Fault Classification Based on Attention Mechanism-Inception-CNN Model[J]. Journal of Vibration and Shock, 2020, 39(19): 84—93.

[4] 李晖照, 王雪, 郭莹. 基于 KL 变换和 KL 散度的电
 网数据特征提取与分类[J]. 电测与仪表, 2019, 56(6):
 87—92.

LI Hui-zhao, WANG Xue, GUO Ying. Feature Extraction and Classification of Power Grid Data Based on KL Transform and KL Divergence[J]. Electrical Measurement and Instrumentation, 2019, 56(6): 87—92.

- [5] 郑怀亮, 王日新, 杨远涛, 等. 数据驱动故障诊断方法泛化性能的经验性分析[J]. 机械工程学报, 2020, 56(9): 116—131.
  ZHENG Huai-liang, WANG Ri-xin, YANG Yuan-tao.
  Empirical Analysis of Generalization Performance of Data Driven Fault Diagnosis Methods[J]. Acta Mechanical Engineering, 2020, 56(9): 116—131.
- [6] CHEN T, GUESTRIN C. XGBoost: A Scalable Tree Boosting System[J]. ACM, 2016(1): 1–10.
- [7] ZHENG Rui-qing, LI Min, CHEN Xiang. BiXGBoost: a Scalable, Flexible Boosting-Based Method for Reconstructing Gene Regulatory Networks[J]. Bioinformatics, 2019, 35(11): 1893—1900.
- [8] YING Xiong, CHEN Shuai, QING Hao-ming. Distributed Representation and One-Hot Representation Fusion with Gated Network for Clinical Semantic Textual Similarity[J]. BMC Medical Informatics and Decision Making, 2020, 20(S1): 1212—1213.
- [9] 李丹, 翟震. 改进的 RBF 神经网络 PID 算法在电磁 振机中的应用[J]. 包装工程, 2019, 40(7): 192—196.
  LI Dan, ZHAI Zhen. Application of Improved RBF Neural Network PID Algorithm in Electromagnetic Vibrator[J]. Packaging Engineering, 2019, 40(7): 192—196.
- [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, 28(4): 31—35.
  WANG Rong-bing, XU Hong-yan, LI Bo, et al. Research on Determination Method of Hidden Layer Node Number of BP Neural Network[J]. Computer Technology and Development, 2018, 28(4): 31—35.
- [11] SHI Wei-wei, GONG Yi-hong, TAO Xiao-yu. Fine-Grained Image Classification Using Modified DCNNs Trained by Cascaded Softmax and Generalized Large-Margin Losses[J]. IEEE Transactions on Neural

Networks and Learning Systems, 2019, 30: 683-694.

- [12] 刘浩,赵丁选,张祝新,等. 基于 BP 神经网络的高速开关阀多级电压控制策略[J]. 农业机械学报, 2019, 50(4): 427—433.
  LIU Hao, ZHAO Ding-xuan, ZHANG Zhu-xin. Multilevel Voltage Control Strategy of High Speed on-off Valve Based on BP Neural Network[J]. Acta Agricultrual Machinery, 2019, 50(4): 427—433.
- [13] BAYER C, BATOR M, MNKS U. Sensorless Drive Diagnosis Using Automated Feature Extraction, Significance Ranking and Reduction[C]// Emerging Technologies & Factory Automation (ETFA 2013), Cagliari, 2013: 1—6.
- [14] 李翱翔,陈健. BP 神经网络参数改进方法综述[J].
   电子科技, 2007(2): 79—82.

LI Ao-xiang, CHEN Jian. Summary of BP Neural Network Parameter Improvement Methods[J]. Electronic Science and Technology, 2007(2): 79—82.

- [15] 林顺富, 顾乡, 汤继开, 等. 基于稀疏自动编码器神 经网络的负荷曲线分类方法[J]. 电网技术, 2020(9): 3508—3515.
  LIN Shun-fu, GU Xiang, TANG Ji-ka, et al. Load Curve Classification Method Based on Sparse Automatic Encoder Neural Network[J]. Power Grid Technology, 2020(9): 3508—3515.
- [16] MALKA N, HALGAMUGE, ESHAN D. Best Optimizer Selection for Predicting Bushfire Occurrences Using Deep Learning[J]. Natural Hazards, 2020(2): 1—16.
- [17] YU X H, CHEN G A, CHENG S X. Dynamic Learning Rate Optimization of the Backpropagation Algorithm[J]. Neural Networks, IEEE Transactions on, 1995, 6(3): 669—770.
- [18] YIN X Y, GOUDRIAAN J. A Flexible Sigmoid Function of Determinate Growth[J]. Annals of Botany, 2003, 91(6): 753-753.