

基于事件检测与 CNN 模型的非侵入式负荷识别方法及实现

凌家源, 彭勇刚

(浙江大学电气工程学院, 浙江 杭州 310027)

摘要: 非侵入式负荷识别是非侵入式负荷监测的一个重要任务。考虑到实际部署的成本问题, 负荷识别的算法常常需要在计算能力与内存受限的设备上运行, 因此本文提出了一种基于事件检测和卷积神经网络模型的负荷识别方法并在基于 STM32 微处理器的嵌入式系统上实现。首先引入带边距的滑动窗口, 选取合适的特征值使用霍特林 T^2 检验进行事件检测, 对检测到的事件使用卷积神经网络模型进行分类从而实现负荷识别, 对训练好的神经网络模型进行压缩后部署到嵌入式设备上, 在 Blued 数据集上对压缩后的算法进行整体性能测试, 取得了较好的结果。

关键词: 非侵入式负荷识别; 事件检测; 卷积神经网络

DOI: 10.12067/ATEEE2010034

文章编号: 1003-3076(2021)03-0046-09

中图分类号: TM721

1 引言

非侵入式负荷监测 (Non-Intrusive Load Monitoring, NILM) 由 Hart 在 20 世纪提出^[1], 负荷识别是非侵入式负荷监测的任务之一, 其目标是从单一的测量点获取总负荷信息, 通过计算的方法对发生动作的负荷进行识别。负荷识别的信息对于用户, 政策制定者以及电网都能带来效益^[2]。

基于事件检测的非侵入式负荷识别可以分为 5 个步骤, 包括数据测量, 数据处理, 事件检测, 特征提取与负荷识别^[3], 当前的主要研究在于事件检测, 特征提取和负荷识别三部分。事件检测常见的算法有专家启发, 匹配滤波和概率模型三类^[4], 在 NILM 问题中, 常用的是概率模型检测算法。文献[5]提出了基于贝叶斯信息准则 (Bayesian Information Criterion, BIC) 的事件检测方法, 通过计算时间窗口内两种假设的最大似然比, 并加入模型复杂度惩罚因子进行假设检验, 获得了相较于累计和 (Cumulative Sum, CUSUM) 算法更高的准确率和计算效率。文献[6]中引入了多维信号, 对贝叶斯信息准则, 累计和和广义似然比检验 (Generalized Likelihood Ratio Test, GLRT) 三种算法进行理论推导和输入二维信

号进行检验, 认为合理选取多维信号有助于提高算法精度, 且不同算法适用于不同的参数条件, 这种方法对超参数的选择要求较高。文献[7]中, 同样引入了多维变量, 并提出了一种特征选取算法 (Feature Selection Algorithm for Detection Purposes, FSADP) 选取合适的特征变量, 使用 BIC, 霍特林 T^2 检验 (Hotelling T^2 test), CUSUM 和有效残差 (Effective residual) 四种方法进行事件检测, 在选取合适的特征及阈值后, Hotelling T^2 检验结果最佳。

深度学习目前是图像识别, 自然语言处理, 机器翻译等领域的主流方法。在负荷识别领域, 得益于公开数据集的增多与完善, 基于深度学习的方法取得良好的表现。文献[8]中首次提出在 NILM 中使用深度神经网络, 提出了长短期记忆网络 (Long Short Term Memory, LSTM), 去噪自动编码器 (Denoise AutoEncoder, DAE) 及一种获得起止时间和平均功率的“矩形”网络模型, 相较于传统的组合优化、隐马尔可夫模型方法性能获得显著的提升。文献[9]提出了一种循环神经网络模型 (Recurrent Neural Network, RNN), 在检测到事件后提取稳态区段的负荷特征作为模型输入进行辨识。文献[10]提出一种基于双向长短期记忆网络 (Bilateral Long-

收稿日期: 2020-10-25

基金项目: 国家重点研发计划项目 (2020YFB0906002)、国家自然科学基金项目 (51877188)、浙江省重点研发计划项目 (2019C01150)

作者简介: 凌家源 (1996-), 男, 广东籍, 硕士研究生, 研究方向为非侵入式负荷监测;

彭勇刚 (1978-), 男, 湖北籍, 教授, 博士, 研究方向为分布式发电、智能电网等 (通讯作者)。

term and Short-Term Memory network, Bi-LSTM), 同时使用负荷投入和切出的有功功率、无功功率和电流谐波作为输入变量, 对多状态和小功率电器也取得了较好的效果, 但是对于事件检测算法的要求较高。上述的算法对于不同的设备需要单独训练不同的模型, 复杂度较高且模型占用空间太大。文献[11]提出了一种通用的负荷识别模型, 模型采用卷积神经网络(Convolutional Neural Networks, CNN)结构, 超参数不依赖于负荷类别, 可以获得0和1的时间序列表示单一负荷的开关状态, 这种模型较为复杂, 但是提出和验证了通用模型的可行性。

综上所述, 非侵入式负荷识别的问题可以分解为事件检测与负荷识别两部分。在事件检测算法中, 多变量的概率模型检测方法取得了更好的效果, 特征变量的选择可以基于文献[7]提出的 FSADP 方法。深度学习表现出良好的性能为负荷识别提供了方法, 多种负荷使用统一的模型也具有可行性。考虑到实际场景中的成本问题, 硬件性能往往受限, 当前的研究中往往使用多个模型, 且模型较为复杂, 不利于实际部署。考虑到上述的问题, 在当前的研究基础上, 本文提出了一种基于事件检测的卷积神经网络模型进行负荷识别。使用带边距的滑动窗口(Window with Margins)^[12]与 Hotelling T² 算法结合进行事件检测, 设计了一个统一的且复杂度较低的 CNN 模型, 把检测到的事件输入 CNN 模型进行识别。将事件检测算法和训练好的 CNN 模型压缩后部署到基于 STM32F407 微处理器的嵌入式系统中并进行性能测试, 验证了算法的有效性和具有实际部署的可行性。算法的训练和测试基于公开数据集 Blued (Building-Level fully-labeled Dataset for Electricity Disaggregation)^[13]。

2 算法模型

本文提出的算法包含事件检测算法和负荷识别算法两部分, 具体的流程为: 事件检测算法通过滑动窗口扫描特征量时间序列并进行运算, 找到事件发生的时间段, 把该时间段内的特征量作为负荷识别算法的输入, 负荷识别算法采用 CNN 模型, 对输入的事件进行识别, 获得发生事件的电器种类。本节对这两部分算法的模型进行了介绍, 并介绍了评估算法性能的一些指标。

2.1 数据集介绍与使用

Blued 数据集是一个带有事件发生标签的高频

数据集。数据集采集了美国一户居民住宅一周的用电数据, 并且对每个设备进行单独监测, 设备状态发生变化时, 会记录该设备的设备号和事件发生的时间, 保存在事件列表文件中^[13]。数据集主要包含了两部分内容, 第一部分是高频采集的电压及电流数据, 采样频率为 12 kHz。第二部分是标记了电器种类, 事件发生时间以及电器所在相位的标签数据。数据集中包含了 35 种不同的电器标签, 共记录了 2 116 个事件, 其中 A 相 740 个, B 相 1 376 个。由于电器标签中包含了一些未知电器, 不明确的标签及相同的电器, 筛选并把相同的电器合并后得表 1 中的 15 种电器标签。

表 1 电器标签列表
Tab.1 List of appliances

序号	电器名称	相
1	台灯	B
2	车库门	B
3	厨房辅助刀	A
4	冰箱	A
5	客厅影视系统	B
6	电脑	B
7	笔记本电脑	B
8	DVD/蓝光播放器	B
9	空气压缩机	A
10	液晶显示器 A	B
11	电视	B
12	打印机	B
13	显示器 B	B
14	电吹风	A
15	灯	A+B

对选取的电器, 根据事件标签中的时间戳, 在连续的电流数据中找到所有事件发生的时间点。对 A 相电器, 以长度为 4 800 个样本点的窗口采集这些时间点附近的电流数据, 使用独热码(one-hot code)记录对应电器的真实标签, 形成一个数据集 A。独热码在多分类问题中常常使用, 即对于 N 分类问题, 有 N 个独热码, 每个码长度为 N 比特, 每个独热码只有一个比特位为 1, 其余为 0, 对应多分类问题中的每一个类别。对于 B 相电器则使用长度为 18 000 的窗口, 使用同样的方法形成数据集 B。把数据集内的样本随机打乱顺序后, 前 70% 作为训练集, 后 30% 作为测试集。这两个数据集分别应用于对应模型的训练和测试。

2.2 事件检测算法

本节主要介绍带边距的滑动窗口及事件检测算

法。假定时间序列 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_j, \dots, \mathbf{x}_n\}$, 其中 \mathbf{x}_j 是所选负荷特征在时间点 j 的值, 对于多维特征, $\mathbf{x}_j = \{x_{j,1}, x_{j,2}, \dots, x_{j,k}, \dots, x_{j,p}\}$, $x_{j,k}$ 是特征 k 在时间 j 的值, p 为特征数量。带边距的滑动窗口有更高的事件检测率^[12], 如图 1 所示, 带边距的滑动窗口分为三部分, 假设窗口长度为 n_w , 对窗口内的操作是分别对前 n_a 个数据及后 n_a 个数据进行计算, 中间 n_b 个数据则不使用, 其中 $n_w = 2n_a + n_b$ 。

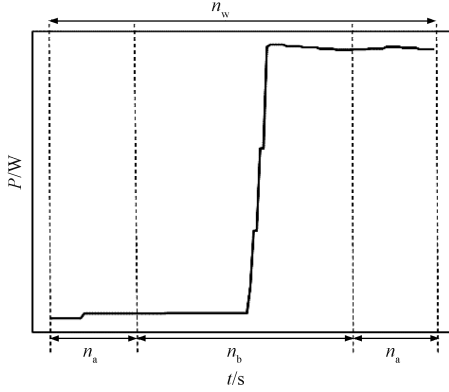


图1 带边距的滑动窗口

Fig.1 Window with margins

概率模型检测的方法可以描述为, 假设上述的时间序列 \mathbf{X} 的元素 x_j 服从高斯分布 $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, 其中 $\boldsymbol{\mu}$ 是 $p \times 1$ 维均值向量, $\boldsymbol{\Sigma}$ 是 $p \times p$ 维协方差矩阵。对滑动窗中的序列, 考虑两个假设:

$$H_0: \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_a}, \mathbf{x}_{n_a+n_b+1}, \dots, \mathbf{x}_{n_w}\} \sim N(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

$$H_1: \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n_a}\} \sim N(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$$

$$\{\mathbf{x}_{n_a+n_b+1}, \dots, \mathbf{x}_{n_w}\} \sim N(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$$

假设 H_0 表示在滑动窗口中未发生变点, 所有样本服从均值向量为 $\boldsymbol{\mu}_0$, 协方差矩阵为 $\boldsymbol{\Sigma}_0$ 的多维高斯分布。假设 H_1 表示在滑动窗口中有变点发生, 在 \mathbf{x}_1 到 \mathbf{x}_{n_a} 间的样本服从均值向量为 $\boldsymbol{\mu}_1$, 协方差矩阵为 $\boldsymbol{\Sigma}_1$ 的多维高斯分布, 而在 $\mathbf{x}_{n_a+n_b}$ 至 \mathbf{x}_{n_w} 间的样本服从均值向量为 $\boldsymbol{\mu}_2$, 协方差矩阵为 $\boldsymbol{\Sigma}_2$ 的多维高斯分布。其中, $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$ 或 $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ 。判断接受哪个假设, 可以通过计算决策值 g_n 与阈值 h 比较, 如果 $g_n > h$ 则接受假设 H_1 , 反之接受假设 H_0 。根据文献[7], Hotelling T^2 检验的决策函数可以表达为:

$$g_n = \frac{1}{4} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_3^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \quad (1)$$

决策值 g_n 与选定的阈值 h 比较, 从而选择接受假设 H_0 或 H_1 , 假设检验可以表示为:

$$g_n = \frac{1}{4} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_3^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \begin{matrix} H_1 \\ > h \\ H_0 \end{matrix} \quad (2)$$

式(2)可以转化为:

$$(\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_3^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \begin{matrix} H_1 \\ > 4h \\ H_0 \end{matrix} \quad (3)$$

由于 h 是经验选择的常数阈值, 因此可以把 $4h$ 仍然记作阈值 h , 令 $g'_n = 4g_n$, 则可以描述为 $g'_n > h$ 则接受假设 H_1 , 若 $g'_n \leq h$ 则接受假设 H_0 , 因此假设检验可以同等地描述为:

$$g'_n = (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_3^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \begin{matrix} H_1 \\ > h \\ H_0 \end{matrix} \quad (4)$$

此外, 常用事件检测算法还有 BIC 和 CUSUM 算法, 根据文献[7], BIC 算法的决策函数和假设检验可以描述为式(5), CUSUM 法的决策函数和假设检验可以表达为式(6):

$$g_{n_bic} = \frac{\det(\hat{\boldsymbol{\Sigma}}_0)^2}{\det(\hat{\boldsymbol{\Sigma}}_1)^2 \det(\hat{\boldsymbol{\Sigma}}_2)^2} \begin{matrix} H_1 \\ > h \\ H_0 \end{matrix} \quad h > 1 \quad (5)$$

$$g_{n_cusum} = (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^T \hat{\boldsymbol{\Sigma}}_1^{-1} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1) \begin{matrix} H_1 \\ > h \\ H_0 \end{matrix} \quad (6)$$

其中均值向量和协方差矩阵的表达式为:

$$\hat{\boldsymbol{\mu}}_1 = \frac{1}{n_a} \sum_{j=1}^{n_a} \mathbf{x}_j \quad (7)$$

$$\hat{\boldsymbol{\mu}}_2 = \frac{1}{n_{aj=n_a+n_b+1}} \sum_{j=n_a+n_b+1}^{n_w} \mathbf{x}_j \quad (8)$$

$$\hat{\boldsymbol{\Sigma}}_1 = \frac{1}{n_a} \sum_{j=1}^{n_a} (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_1) (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_1)^T \quad (9)$$

$$\hat{\boldsymbol{\Sigma}}_2 = \frac{1}{n_{aj=n_a+n_b+1}} \sum_{j=n_a+n_b+1}^{n_w} (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_2) (\mathbf{x}_j - \hat{\boldsymbol{\mu}}_2)^T \quad (10)$$

$$\hat{\boldsymbol{\Sigma}}_3 = \frac{1}{2} (\hat{\boldsymbol{\Sigma}}_1 + \hat{\boldsymbol{\Sigma}}_2) \quad (11)$$

$$\hat{\boldsymbol{\Sigma}}_0 = \frac{1}{2} (\hat{\boldsymbol{\Sigma}}_1 + \hat{\boldsymbol{\Sigma}}_2) + \frac{1}{4} (\hat{\boldsymbol{\mu}}_2 - \hat{\boldsymbol{\mu}}_1)^2 \quad (12)$$

2.3 卷积神经网络简介

卷积神经网络在图像识别领域有广泛的应用。事实证明,在图象识别中,诸如不同方向的边缘、拐角、斑点等低级特征有重要的作用^[8]。卷积神经网络使用多个具有局部感受野的卷积核,在整个输入图像上滑动这些卷积核以自动获得有效的低级特征。

负荷识别任务有类似的特点,应用于负荷识别的 CNN 模型使用一维卷积, CNN 模型如图 2 所示,使用一维卷积核滑动通过输入的一维序列进行卷积计算,可以得到一组特征(feature maps),卷积后一般进行最大池化(maxpooling),池化是一个下采样过程,对卷积得到的特征进行最大值池化可以去除冗余的特征,从而减少后面的计算量。对于提取得到的多个特征,可以使用 flatten 操作展开成单通道的一维序列再连接到全连接层,经过全连接后,输出层使用 softmax 函数进行多分类,从而实现负荷识别。

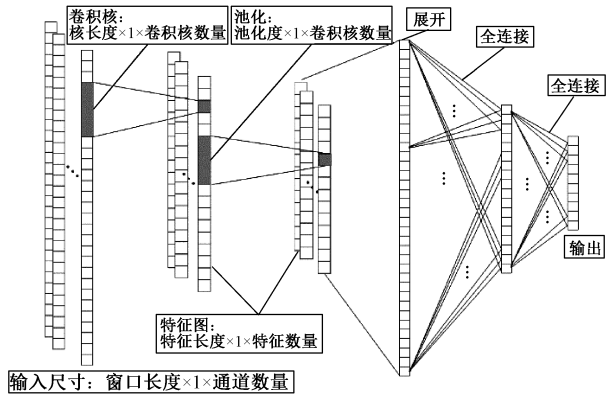


图2 卷积神经网络模型

Fig.2 CNN model

2.4 性能指标

为了评估事件检测算法,神经网络模型以及两部分合并后的整体算法性能,选用了一些在此类问题中常用的性能指标^[14]。

对于多分类问题,可以通过混淆矩阵评估。混淆矩阵包含了每一类被正确分类的数量及被分为其他类的数量,可以直观地观察到分类算法的性能。

精确率(Precision Rate, PR),表示所有检测到的事件中正确分类的比例;召回率(Recall),表示所有事件中被检测到的比例。这两个指标计算公式如下:

$$PR = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

式中,TP 为设备被正确划分到相应的开启状态的数量;FP 为设备被分类为开启状态而实际为关闭状态的数量;FN 为被分类为关闭而实际为开启的数量;TN 为被分类为关闭且实际是关闭的数量。

精确率和召回率两个参数常常因为实际原因会发生冲突而难以评估,F1 分数(F1 score)是这两个指标的调和平均值,兼顾了两者,计算公式为:

$$F1 = \frac{2PR \times Recall}{PR + Recall} \quad (15)$$

3 实验与结果

3.1 特征量选取

对于事件检测,根据文献[7],实验结果表明 A 相选取有功功率基波 P_1 , B 相选取有功功率 P ,基波功率 P_1 ,无功功率谐波分量总和 Q_H 及无功功率三次谐波分量 Q_3 作为事件检测的特征量得到最好效果,实际上 B 相仅使用 P_1 得到的效果与最佳效果十分接近,考虑到最终部署设备的计算能力与内存的限制,统一使用 P_1 作为事件检测的特征变量。根据最新的 IEEE 1459—2010 标准,非正弦条件下的单相有功功率计算公式为^[15]:

$$P = P_1 + P_H \quad (16)$$

式中, P_1 是有功功率基波; P_H 为有功功率谐波总和。 P_1 计算公式为:

$$P_1 = \frac{1}{kT} \int_{\tau}^{\tau+kT} v_1 i_1 dt = V_1 I_1 \cos \theta_1 \quad (17)$$

电压与电流的傅里叶系数表达式为^[16]:

$$v_{ak} = \frac{2}{M} \sum_{m=i}^{i+M-1} v[m] \cos(2\pi m k f_0 / f_s) \quad (18)$$

$$v_{bk} = \frac{2}{M} \sum_{m=i}^{i+M-1} v[m] \sin(2\pi m k f_0 / f_s) \quad (19)$$

$$i_{ak} = \frac{2}{M} \sum_{m=i}^{i+M-1} i[m] \cos(2\pi m k f_0 / f_s) \quad (20)$$

$$i_{bk} = \frac{2}{M} \sum_{m=i}^{i+M-1} i[m] \sin(2\pi m k f_0 / f_s) \quad (21)$$

式中, f_0 为基波频率; f_s 为采样频率,根据 Blued 数据集, $f_0 = 60 \text{ Hz}$, $f_s = 12 \text{ kHz}$, $M = f_0 / f_s$; k 为谐波次数,由式(11)~式(15),令 $k=1$,有功功率基波计算公式可以转化为:

$$P_1 = \frac{1}{2} (v_{a1} i_{a1} + v_{b1} i_{b1}) \quad (22)$$

在设备发生事件时,高频电流数据包含了丰富

的设备特征信息,对于功率相近的电器有更好的辨识度。直接使用电流作为负荷识别的特征量无需进一步数据处理,对于计算能力和内存受限的嵌入式设备更为友好。因此,在上述的事件检测算法执行后,得到事件发生的时间段,可以直接使用时间段内的高频电流数据作为卷积神经网络的输入进行负荷识别,实验表明取得了较好的效果。

3.2 事件检测

对 Blued 数据集的原始数据使用式 (18) ~ 式 (22) 计算后,可以分别获得 A 相和 B 相的有功功率的基波 P_{1A}, P_{1B} 。由于 A、B 相负载差别较大,A 相负荷种类较少且结构简单,B 相负荷种类繁多复杂,且有电脑、电视、液晶显示器等具有大量电力电子器件的设备,图 3 展示了 B 相的一段有功功率信号,这些设备产生了大量的高频谐波,导致一些小功率电器事件容易被丢失,或由于噪音过大而产生误检。因此需要使用不同的参数进行事件检测才能分别获得较好的结果,对于负荷复杂的场景,可以适当增大滑动窗口的长度从而获得更多的状态变化信息。因为检测得到的事件会作为神经网络模型的输入进行分类,对于误检测的事件可以分类为负类,即没有发生事件,因此查全率对于整体性能有更大的影响。为了提高查全率,滑动窗口的滑动步长设置为 1,滑动窗口经过整个时间序列并计算每个窗口的决策值 g_n 。通过尝试,适当调高阈值可以提高精确率 PR,但是可能导致召回率降低,为了整体性能,事件检测算法调低了阈值以提高召回率,牺牲了部分精确率。

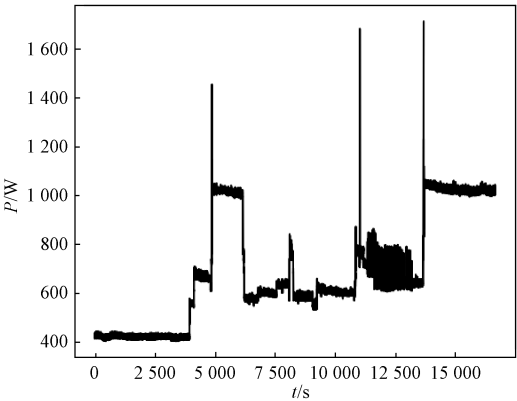


图 3 B 相有功功率信号
Fig.3 Power signal on phase B

在事件发生附近的功率曲线与 g_n 曲线图 4 所示,可见在事件发生处 g_n 曲线产生了大于阈值 h 的峰值,不考虑同时发生的事件,则此处可以视作发生了一个事件。

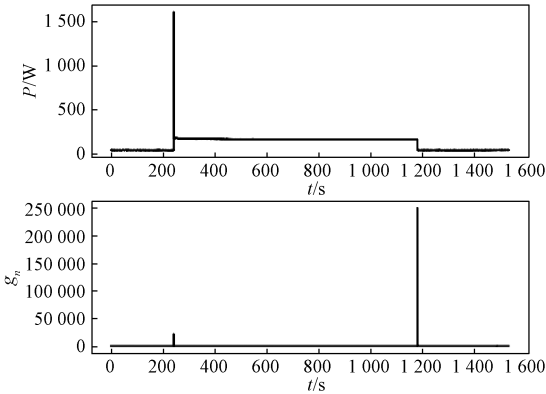


图 4 发生事件时的功率曲线与 g_n 曲线
Fig.4 Power curve and g_n value at events

为 Hotelling T^2 , BIC, CUSUM 三种算法选取合适的参数值,具体数值如表 2 所示,在 Blued 数据集上运行三种算法,可以得到表 3 所示的结果。由实验结果可以观察到,在 A 相上,由于负荷组成较为简单,三种算法均取得较好的效果,其中 CUSUM 算法表现最佳,略优于 Hotelling T^2 算法,但差距不大。在 B 相上,负荷较多且复杂,BIC 算法的准确率与召回率均有明显下降,不符合实际的性能需求。CUSUM 算法的召回率略优于 Hotelling T^2 算法,但是准确率比 Hotelling T^2 差较多。较高的准确率可以减少负荷识别算法的运行次数,这对于计算能力有限的嵌入式设备尤为重要,有利于满足实时性能的需求。可见 Hotelling T^2 算法在实际场景下更具有优势。

表 2 事件检测算法的参数

Tab.2 Parameters of event detection algorithm						
算法	Hotelling T^2		BIC		CUSUM	
相	A	B	A	B	A	B
n_w	24	90	24	90	24	90
n_a	8	8	8	6	8	8
n_b	8	74	8	78	8	74
h	800	100	1 000	80	10 000	100

表 3 事件检测算法结果

Tab.3 Event detection performance						
算法	Hotelling T^2		BIC		CUSUM	
相	A	B	A	B	A	B
PR	0.923	0.636	0.888	0.102	0.956	0.464
Recall	0.985	0.954	0.991	0.684	0.992	0.961
F1	0.953	0.763	0.936	0.178	0.973	0.626

3.3 负荷识别卷积神经网络模型

本节主要介绍了在事件检测后用于负荷识别的

卷积神经网络的结构,以及基于 Blued 数据集对模型进行训练和测试的结果。

3.3.1 卷积神经网络模型结构

由事件检测算法得到事件发生时间点,使用窗口截取该时间点附近的高频电流数据作为模型的输入进行负荷识别。图 5 展示了本文提出的卷积神经网络结构。模型由 8 层网络组成,损失函数使用交叉熵损失函数,对于 A,B 两相的模型结构相同,但其中的超参数不同,具体结构为:

(1)输入层,用于接受输入数据,确定输入数据的尺寸。用于 A 相的模型输入尺寸为 $4\ 800 \times 1$,B 相的尺寸为 $18\ 000 \times 1$ 。

(2)一维卷积层,两个模型参数相同,卷积核数量 $\text{filters} = 2$,卷积核尺寸 $\text{kernel_size} = 20 \times 1$,步长 $\text{strides} = 1$,通过填充保证每个通道的输出与输入尺寸一致,即 $\text{padding} = \text{same}$,激活函数为 ReLU 函数。

(3)池化层,对 1 维信号进行最大值池化,A 相模型种下采样因子 $\text{pool_size} = 32$,步长 $\text{strides} = 16$,B 相模型种下采样因子 $\text{pool_size} = 32$,步长 $\text{strides} = 20$ 。

(4)Flatten 层,用于将输入“压平”,把池化层输出的多通道数据展开为一维数据。

(5)Dropout 层,为防止过拟合在每次训练时随机断开一定比例(p)数据的连接,A 相模型中 $p = 0.05$,B 相中 $p = 0$ 。

(6)全连接层,A 相模型节点数量 $N = 128$,B 相模型 $N = 128$,激活函数为 ReLU 函数。

(7)全连接层,A 相模型节点数量 $N = 32$,B 相模型 $N = 64$,激活函数为 ReLU 函数。

(8)输出层,A 相模型节点数量 $N = 5$,B 相模型 $N = 11$,激活函数为 softmax 函数。

ReLU 即线性整流单元(Rectified Linear Unit, ReLU),是神经网络模型中常用的激活函数,能抵抗梯度消失问题,且运算速度相较于 sigmoid, tanh 等函数更快,其函数表达式为:

$$f(x) = \max(0, x) \quad (23)$$

softmax 函数常用于多分类模型的输出层,它将多个神经元的输出映射到 $(0, 1)$ 区间,表示属于该分类的概率,所有输出的概率和为 1,输出中概率最大的作为本次分类的结果,将该位置的比特置为 1,其余位置置为 0 得到分类结果的独热码,再与实际标签的独热码进行对比分析。softmax 函数表达式为:

$$y_i = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}}, \sum_{i=1}^n y_i = 1 \quad (24)$$

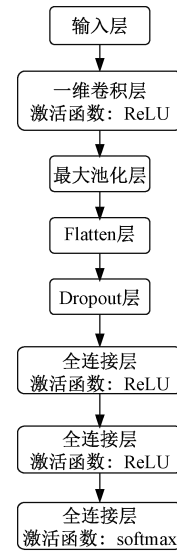


图 5 CNN 模型结构

Fig.5 CNN model structure

3.3.2 负荷识别结果

本文使用 Python 及 Keras 实现了上述的神经网络模型,模型在两块 NVidia GTX1080ti 具有 11 GB 显存的 GPU 上进行训练,训练时间约为 10 min,训练后的 A 相模型大小为 986 KB,B 相模型大小为 2 840 KB。

A 相负荷识别结果的混淆矩阵如图 6 所示,横轴表示模型预测的电器种类,纵轴表示该样本的实际标签,图中的数值已经进行归一化处理,表示每种类型的电器被分类为不同标签的比例。表 4 展示了 A 相各类电器的分类性能。

表 4 A 相电器分类性能指标

Tab.4 Identification performance of phase A

	PR	Recall	F1
厨房辅助刀	1.00	0.99	0.99
冰箱	1.00	1.00	1.00
空气压缩机	1.00	1.00	1.00
电吹风	1.00	0.95	0.98
灯	1.00	0.99	0.99

B 相上的电器种类多且复杂,存在特征相近的电器,负荷识别结果的混淆矩阵如图 7 所示,表 5 展示了 B 相各类电器的分类性能指标。

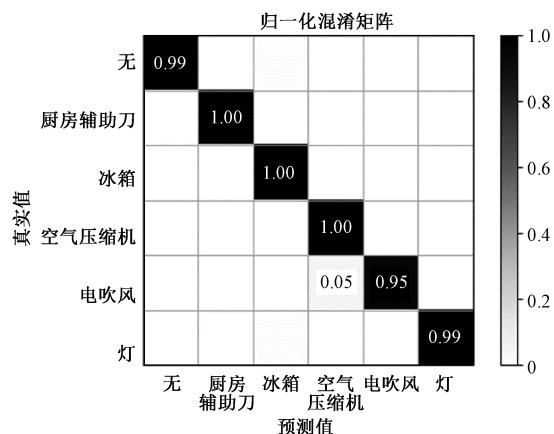


图6 A相负荷识别混淆矩阵

Fig.6 Confusion matrix of phase A

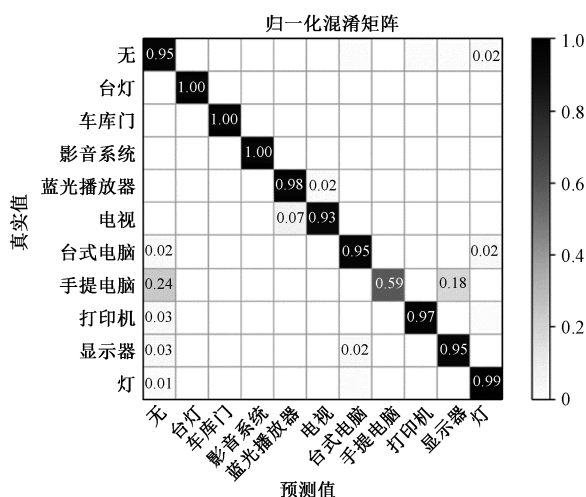


图7 B相负荷识别混淆矩阵

Fig.7 Confusion matrix of phase B

表5 B相电器分类性能指标

Tab.5 Identification performance of phase B

	PR	Recall	F1
台灯	0.94	1.00	0.97
车库门	1.00	1.00	1.00
影音系统	0.85	1.00	0.92
蓝光播放器	0.87	0.98	0.92
电视	0.96	0.93	0.95
台式电脑	0.91	0.95	0.93
手提电脑	0.77	0.59	0.67
打印机	0.98	0.97	0.97
显示器	0.90	0.95	0.92
灯	0.92	0.99	0.95

3.3.3 压缩后的模型性能

要把神经网络模型部署到基于 STM32F407 微处理器的嵌入式系统上,需要把通过 Python 和

Keras 实现的模型转化成 C 语言代码并压缩。意法半导体公司为 STM32 系列微处理器提供了图形化的初始代码生成器软件 STM32CubeMX,它的扩展功能 X-CUBE-AI 可以把主流的深度神经网络模型压缩并转换为高效的 C 语言代码。

选用的 STM32F407VET6 微处理器具有 512 KB 的 Flash 和 192 KB 的 RAM,时钟频率最高可达 168 MHz。选择压缩率为 4,把两个训练好的模型压缩并转换后,A 相模型占用了 84.29 KB 的 Flash 及 2.9 KB 的 RAM,B 相模型占用了 267.73 KB 的 Flash 及 7.7 KB 的 RAM。在测试集中随机挑选了 2 000 个 A 相样本及 1 000 个 B 相样本,微处理器时钟频率为 168 MHz,在嵌入式设备上运行神经网络模型,A 相模型平均每运行一次耗时 58.295 ms,分类的性能指标如表 6 所示。B 相模型平均每运行一次耗时 207.898 ms,分类的性能指标如表 7 所示。由测试结果可得,经过压缩和转换后的模型性能基本没有下降。不考虑同时发生的事件的前提下,模型的运行时间可以满足性能需求。

表6 在嵌入式设备上运行的 A 相电器分类性能

Tab.6 Identification performance of phase A running on embedded devices

	PR	Recall	F1
厨房辅助刀	1.00	1.00	1.00
冰箱	1.00	1.00	1.00
空气压缩机	0.93	1.00	0.96
电吹风	1.00	1.00	1.00
灯	1.00	0.99	0.99

表7 在嵌入式设备上运行的 B 相电器分类性能

Tab.7 Identification performance of phase B running on embedded devices

	PR	Recall	F1
台灯	0.98	1.00	0.99
车库门	1.00	1.00	1.00
影音系统	0.85	1.00	0.92
蓝光播放器	0.90	0.94	0.92
电视	0.97	0.90	0.94
台式电脑	0.88	0.92	0.90
手提电脑	0.71	0.68	0.70
打印机	0.99	0.96	0.97
显示器	0.90	0.91	0.90
灯	0.98	0.95	0.96

3.4 整体算法性能评估

把事件检测算法与压缩后的神经网络模型合

并,先进行事件检测,找到事件发生的时间段后,把该段时间的电流数据作为神经网络的输入进行负荷识别。把整体算法下载到嵌入式设备上,在上位机 PC 上准备好测试数据后,通过串口把电压电流数据流传输到嵌入式设备中,从而模拟实时采集的数据流。嵌入式设备接收数据并存储,同时对存储的数据运行整体算法,记录结果并上传到上位机 PC 中,上位机编程语言使用 Python。使用 Blued 数据集中的约 11 h 的数据作为本次评估的测试数据,其中 A 相上发生 75 个事件,B 相上发生 80 个事件,结果如表 8 所示。算法在 A 相上的表现较好,F1 分数达到了 0.9,而 B 相上由于设备的复杂性原因,F1 分数为 0.85。在硬件能力受限的实际情况下,算法达到了较为理想的准确率,可见本文提出的方法有效且具有实际部署的可行性。

表 8 整体算法的性能

Tab.8 Overall algorithm performance

	PR	Recall	F1
A 相	0.90	0.91	0.90
B 相	0.86	0.85	0.85

4 结论

考虑到实际场景中硬件受限的问题,本文提出了一种运行于嵌入式设备上的非侵入式负荷识别方法。使用了带边距的滑动窗口与 Hotelling T^2 算法进行事件检测,应用 CNN 模型进行负荷识别,利用 CNN 可以自动提取特征的特点,使用特征丰富的高频电流数据作为输入,取得了较好的识别效果。为了在嵌入式设备上实现,对神经网络模型压缩和转换,并在板上进行了性能测试,识别性能基本没有下降,运行时间也满足实际需求。把事件检测算法与压缩后的模型合并后得到一个整体算法,串口模拟实际数据流发送到嵌入式设备进行整体算法的性能测试,获得了较好的识别性能,验证了所提出的算法的有效性与可行性。

参考文献 (References):

- [1] Hart G. Nonintrusive appliance load monitoring [J]. Proceedings of the IEEE, 1992, 80 (12): 1870-1891.
- [2] Froehlich J, Larson E, Gupta S, et al. Disaggregated end-use energy sensing for the smart grid [J]. IEEE Pervasive Computing, 2010, 10 (1): 28-39.
- [3] 程祥, 李林芝, 吴浩, 等 (Cheng Xiang, Li Linzhi, Wu Hao, et al.). 非侵入式负荷监测与分解研究综述 (A survey of the research on non-intrusive load monitoring and disaggregation) [J]. 电网技术 (Power System Technology), 2016, 40 (10): 3108-3117.
- [4] Anderson K, Berges M, Ocneanu A, et al. Event detection for non intrusive load monitoring [A]. IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society [C]. Montreal, Canada, 2012. 3312-3317.
- [5] 肖江, Auger F, 荆朝霞, 等 (Xiao Jiang, Auger F, Jing Zhaoxia, et al.). 基于贝叶斯信息准则的非侵入式负荷事件检测算法 (Non-intrusive load event detection algorithm based on Bayesian information criterion) [J]. 电力系统保护与控制 (Power System Protection and Control), 2018, 46 (22): 8-14.
- [6] 张露, Auger F, 荆朝霞, 等 (Zhang Lu, Auger F, Jing Zhaoxia, et al.). 基于非侵入式的事件检测方法统计评估 (Statistical assessment of abrupt change detection for NILM) [J]. 电测与仪表 (Electrical Measurement & Instrumentation), 2020, 57 (1): 106-112, 120.
- [7] Houidi S, Auger F, Ben A S H, et al. Multivariate event detection methods for non-intrusive load monitoring in smart homes and residential buildings [J]. Energy and Buildings, 2020, 208 (Feb.): 109624. 1-109624. 14.
- [8] Kelly J, Knottenbelt W. Neural NILM: Deep neural networks applied to energy disaggregation [A]. 2nd ACM International Conference on Embedded System For Energy-Efficient Built Environments [C]. Seoul, South Korea, 2015.
- [9] 刘恒勇, 史帅彬, 徐旭辉, 等 (Liu Hengyong, Shi Shuaibin, Xu Xuhui, et al.). 一种关联 RNN 模型的非侵入式负荷辨识方法 (A non-intrusive load identification method based on RNN model) [J]. 电力系统保护与控制 (Power System Protection and Control), 2019, 47 (13): 162-170.
- [10] 何恒靖, 王灏, 肖勇, 等 (He Hengjing, Wang Hao, Xiao Yong, et al.). 基于 Bi-LSTM 算法的非侵入式负荷监测模型 (Non-intrusive load monitoring model based on Bi-ISTM algorithm) [J]. 南方电网技术 (Southern Power System Technology), 2019, 13 (2): 20-26.
- [11] Barsim K S, Wiewel F, Yang B. On the feasibility of generic deep disaggregation for single-load extraction [A]. The 4rd International Workshop on Non-intrusive Load Monitoring [C]. Austin, America, 2018.
- [12] Azzini H, Torquato R, Silva L C P. Event detection methods for nonintrusive load monitoring [A]. 2014 IEEE Power & Energy Society General Meeting [C]. Washington, America, 2014. 1-5.
- [13] Anderson K, Ocneanu A, Benitez D, et al. BLUED: A fully labeled public dataset for event-based non-intrusive

- load monitoring research [A]. Proceedings of the 2nd KDD Workshop on Data Mining Applications in Sustainability (SustKDD) [C]. Beijing, China, 2012. 1-5.
- [14] Batra N, Kelly J, Parson O, et al. NILMTK: An open source toolkit for non-intrusive load monitoring [A]. 5th International Conference on Future Energy Systems (ACM e-Energy) [C]. Cambridge, UK, 2014. 265-276.
- [15] IEEE standard 1459-2000. IEEE standard definition for the measurement of electric power quantities under sinusoidal, nonsinusoidal, balanced and unbalanced conditions [S].
- [16] Houidi S, Auger F, Fourer D, et al. Relevant feature selection for home appliances recognition [A]. Electrimacs 2017 [C]. Toulouse, France, 2017.

Implementation of a non-intrusive load identification method based on event detection and CNN model

LING Jia-yuan, PENG Yong-gang

(College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China)

Abstract: Load identification is one of the important tasks of non-intrusive load monitoring (NILM). Considering the cost of deployment, load identification algorithms need to run on devices with limited computing power and memory. This paper proposes an identification method based on event detection and Convolutional Neural Network (CNN). The method is implemented on an embedded device based on STM32 MCU. Sliding Window with Margins method and Hotelling T^2 test are used for event detection. Then the detected events are identified by the CNN models. The trained CNN models have to be compressed and then deployed to the embedded device. The performance of the overall algorithm is tested on the Blued dataset and satisfied results are obtained.

Key words: non-intrusive load identification; event detection; convolutional neural network