

基于演化 LSTM 神经网络的用户终端睡眠预测模型^①



李晓敏, 秦晓卫

(中国科学技术大学 中国科学院无线光电通信重点实验室, 合肥 230026)
通讯作者: 秦晓卫, E-mail: qinxw@ustc.edu.cn

摘要: 用户处于睡眠状态时手机后台自主挂起不必要的系统或应用进程可以有效降低能耗, 因此在不损害用户使用体验的前提下准确判断用户是否处于睡眠状态具有重要意义. 围绕该问题设计了覆盖率和唤醒率作为新的衡量指标, 提出一种基于 LSTM 神经网络的睡眠预测模型, 结合 LSTM 神经网络能够较好处理时序特征数据的特点和演化算法能够优化不可导目标函数的特性, 将 LSTM 神经网络的参数作为差分演化算法的优化参数, 覆盖率和唤醒率的综合目标作为选择函数, 同时在每次迭代中重新评估选择函数使其适应小批量训练法. 实验结果表明, 采用演化算法训练 LSTM 神经网络得到的预测结果相较于传统分类模型能在低唤醒率时达到更好的覆盖率, 平均提升约 5%.

关键词: 演化算法; LSTM 网络; 用户体验; 睡眠预测; 智能手机

引用格式: 李晓敏, 秦晓卫. 基于演化 LSTM 神经网络的用户终端睡眠预测模型. 计算机系统应用, 2020, 29(11): 196-203. <http://www.c-s-a.org.cn/1003-3254/7660.html>

Evolutionary LSTM Neural Network as Sleep Prediction Model on Smartphone

LI Xiao-Min, QIN Xiao-Wei

(CAS Key Laboratory of Wireless-Optical Communications, University of Science and Technology of China, Hefei 230026, China)

Abstract: Suspending unnecessary system or application processes in the background of the mobile phone while the user is sleeping can effectively reduce energy consumption, so it is of great significance to accurately determine whether the user is sleeping without compromising the user experience. Based on this problem, the coverage rate and wake rate are designed as new metrics. A sleep prediction model based on LSTM neural network is proposed, the LSTM neural network can handle time-series feature data and the evolution algorithm can optimize non derivable optimization targets. The parameters of the LSTM neural network are used as the optimization parameters of the differential evolution algorithm, and the comprehensive target of coverage and wake-up rates are used as the selection function. The selection function is re-evaluated in each iteration to use the mini-batch training. The experimental results show that compared with the traditional classification model, the prediction results obtained by training the LSTM neural network with evolutionary algorithm can achieve better coverage at low wake-up rate, with an average improvement of about 5%.

Key words: evolutionary algorithm; LSTM network; user experience; sleep detection; smartphone

1 引言

近年来, 智能手机的性能越来越强大, 能够承载的

功能也越来越丰富. 但与此同时, 智能手机的功耗也不断增加. 电池技术的发展比较缓慢, 并不遵循摩尔定律,

^① 基金项目: 国家重点研发计划 (2018YFA0701603)

Foundation item: National Key Research and Development Program of China (2018YFA0701603)

收稿时间: 2020-03-22; 修改时间: 2020-04-21; 采用时间: 2020-05-10; csa 在线出版时间: 2020-10-29

导致手机电池的供给能力增长缓慢,无法满足手机能耗的需求,制约手机的续航能力.因此,在电池技术不能取得快速突破的背景下,研究如何有效的降低手机功耗具有非常重要的意义.

用户在夜间休息的时候,手机实际上处于不使用状态,但手机后台仍有很多不必要的进程和模块在运行,如后台应用和GPS定位模块等.这些后台非必要进程或模块占用大量的传输能耗和计算能耗.如果手机终端能够准确识别用户是否处于睡眠状态,就可以在用户处于睡眠状态时,终止后台不必要的进程或模块,进入低功耗模式,提升手机终端续航能力.

手机上丰富的传感器数据以及人机交互信息可以反应用户的行为和状态,因此我们可以利用这些信息推断用户是否处于睡眠状态.现有睡眠预测方法常用于用户健康监测领域,Roomkham等^[1]介绍了使用消费级设备如智能手机和可穿戴设备来预测睡眠的优点和缺点以及常用的预测方法,其基本思想是基于机器学习分类模型,如SVM^[2]、贝叶斯网络^[3]、C4.5决策树^[4]和LSTM神经网络^[5]等,利用手机上传感器等信息来识别用户是否处于睡眠状态.Min等^[6]开发了一款Toss 'N' Turn应用程序用来预测用户睡眠时长和睡眠质量,该应用采集7种传感器数据,收集27个人一个月的手机传感器数据和睡眠日志,采用C4.5决策树和贝叶斯网络作为模型,分别基于群体数据和个体数据训练获得群体模型和个体模型.该应用在群体模型上睡眠准确率和睡眠质量准确率分别达到了93.06%和81.48%,在个体模型上睡眠准确率和睡眠质量预测准确率分别达到了94.52%和83.97%.Chen等^[7]介绍了一种基于best effort sleep的方法,采用不干扰用户正常作息的方法,通过对比手机使用时长,充电时间的长短等来预测用户的睡眠时长以及睡眠质量,预测的粒度为小时级别.Bai等^[8]通过设计基于匹兹堡睡眠质量指数(Pittsburgh Sleep Quality Index, PSQI)^[9]的问卷来获取用户实际睡眠质量,并提取日常活动、生活环境和社交活动等特征,在此基础上提出了一种基于因子图的睡眠预测方法,预测睡眠质量准确率达到了78%.Chen等^[10]提出了使用双向LSTM神经网络预测睡眠状态.与上述研究不同的是,本文的研究出发点是节能,当判断用户处于睡眠状态后,手机将进入低功耗状态,需要同时考虑用户使用体验和手机省电效率,极大化避免误杀操作.直接采用上述的分类模型会带来较多的误杀操作,不能保证用户使用体验.

预测用户是否处于睡眠状态是一个二分类问题,

但与普通的二分类问题不同的是,用户不会频繁地在睡眠状态和非睡眠状态之间切换,用户通常一个晚上只入睡一次,睡眠状态和非睡眠状态在时间上都具有一定的连续性.所以在衡量模型性能时,不仅要考虑准确率,预测结果也不能频繁跳变,预测结果频繁的跳变会带来较多的误杀,这严重影响用户的使用体验.为此,本文定义覆盖率和唤醒率为模型衡量指标,覆盖率关注用户处于睡眠时手机进入低功耗模式的时长比例,覆盖率高意味着高省电效率;唤醒率关注预测结果状态跳变的次数,唤醒率低意味着好的用户体验.

传统的机器学习分类模型如SVM,决策树等模型损失函数固定,模型不易修改,可通过调整参数如正负样本权重的方法达到调节覆盖率和唤醒率的效果,但实验结果显示此类方法在唤醒率较低时,覆盖率表现较差.神经网络也可以作为分类模型,其中LSTM神经网络可以较好地处理时序数据,神经网络具有损失函数定义灵活,模型结构容易调整的特点,通常取可导的损失函数,如均方误差或交叉熵等.在本文所述的问题中,唤醒率很难转换为可导的损失函数,不能使用梯度下降法进行训练,直接对唤醒率优化较为困难.神经网络也可采用调整正负样本比重的方法间接的达到调节覆盖率和唤醒率的效果,但是同样在唤醒率较低时,覆盖率表现较差.演化算法^[11]是一种具有高鲁棒性和广泛适用性的全局优化算法,具有自组织、自适应和自学习的特点,能够不受问题性质的限制,可以用来解决不可导的优化问题.为此,本文提出了采用演化算法训练LSTM神经网络的方式,直接优化覆盖率和唤醒率的综合目标.实验结果表明,采用演化算法训练LSTM神经网络的方法相比于传统模型能够在唤醒率较低时达到更好的覆盖率效果.

2 数据集构建

2.1 数据采集

睡眠预测的前提和基础在于睡眠数据的获取,本文采用智能手机获取睡眠数据,主要包括传感器数据和人机交互信息.与睡眠相关的传感器信息如环境光信息(通过环境光传感器),运动信息(通过加速度传感器和重力传感器)等.当手机放在口袋中或屏幕朝下放置在桌面时,环境光信息将不可靠.因此,还收集了屏幕接近传感器信息.环境嘈杂状态也可提供睡眠相关信息,可通过麦克风传感器获得,但出于隐私保护的考

虑,不收集此类传感器信息.人机交互信息(如屏幕开关,电池充电状态)也可提供睡眠方面信息,例如,屏幕亮表明用户大概率没有睡着,人们在睡觉前可能有为手机充电的习惯,两者都提供睡眠相关的信息.采集的原始数据如表1所示.

表1 采集数据列表

传感器数据	数据类型	作用
加速度传感器	浮点型	用于获取运动状态
重力传感器	浮点型	用于获取运动状态
环境光传感器	浮点型	获取环境光
屏幕接近传感器	浮点型	判断手机是否处于口袋等
屏幕开关状态	01型	屏幕是否唤醒
电池状态	01型	用户睡前可能有充电习惯

我们开发了数据采集APP,并与华为公司合作采集Beta用户数据,采集了629个用户18天的睡眠数据.将用户按照7:3的比例划分为A组和B组,A组用户产生的数据作为训练集,B组用户产生的数据作为测试集.传感器数据每5秒采集一条,屏幕状态和充电状态每变化一次采集一条.

2.2 数据标注

模型训练需要首先给数据打标签,一个方案是采用问卷调查的方式,但是这种方式要求使用者能够准确的回忆起入睡的细节,这通常是困难的^[8].从节能角度,我们实际不关心用户是否真实处于睡眠模式,故按照一定的规则根据屏幕变化状态将用户一天中较长时间没有使用手机的时间段视为睡眠.睡眠起止时间的确定主要考虑以下几点:

(1) 用户在睡眠过程中可能会有少量、短时间的唤醒手机,如半夜醒来上厕所.在设计规则时应将此种情况忽略,在一定时间内手机唤醒时长较短且唤醒次数较少时应该仍然视为睡眠,只有大于一定时长和次数的唤醒才认为是清醒状态.

(2) 用户进入睡眠开始时,应该有一段持续时间比较长的息屏状态.但Beta用户可能拥有多台手机,其中一台备用机可能长时间处于息屏状态,对于这种情况,应该过滤.

(3) 有部分用户白天用手机比较少,但一般白天的唤醒次数和唤醒时间比晚上多.故需要在以上的处理之外再加上睡眠期间手机最大唤醒次数的限制,以排除白天轻量的手机使用被误判为睡眠的情况.

基于以上因素考虑,本文提出了以下几条规则:

规则1. 取一个长度为30分钟的时间窗口,当窗口内亮屏累计小于等于3分钟,且亮屏次数小于等于

3次,将亮屏数据忽略.

规则2. 当息屏持续时间大于4小时且小于16小时,视为睡眠.

规则3. 对视为睡眠的时间段内的数据,如果唤醒次数超过8次,则不视为睡眠.

2.3 特征处理

采集的原始数据并不都具备明确的意义,模型直接难以学习,因此需要对原始数据做进一步的处理.将时间序列采集数据划分为一系列不重叠的10分钟窗口作为数据分析和特征提取的时间单位,选用10分钟窗口作为基本分析单位是因为10分钟为一个平均睡眠潜伏期(从清醒到入睡的过渡时期)^[6],且更细粒度的省电效率也无必要.静止持续时间可通过加速度传感器和重力传感器获得,加速度传感器值减去重力传感器值为用户的真实加速度值,利用每个窗口内的加速度方差判断用户的运动状态.用户在睡眠状态时通常处于静止状态,取用户在睡眠时间段的所有窗口的加速度方差值,计算均值 $mean$ 和标准差 std ,取门限 $T=mean+2std$,则当某个窗口的方差在门限之下时视为静止状态,否则为运动状态,得到运动状态后可以计算出静止持续时间.根据息屏状态可直接计算出用户的息屏持续时间.最后特征包括:当前时刻 t ,静止持续时间 s_d ,息屏持续时间 sc_d ,环境光的强度 l ,用户充电状态 c ,接近传感器数据 d .

部分特征还存在数值分散的问题,如在非灯光直射时环境光强度值 l 分布从零到数百,但在灯光直射时强度值可达到数万,这使得模型容易受到极值影响,难以学习,且鲁棒性较差.而数值在不同区域的意义不同,采用归一化处理的效果较差.本文采用最短描述长度原则(Minimum Description Length Principle, MDLP),利用信息增益最大化的方式寻找连续变量的最佳分点,输入一个维度的数据,寻找当前的最优切分点,使得切分后的信息增益保持最大值.当切分点确定后,将连续变量一分为二,分为两部分数据集,在这两部分数据集中用同样的方法循环切分.并在MDLP的基础上引入随机化查找的方法,先得到的 m 个切分点,然后多次随机选取 $n(m > n)$ 个切分点计算信息增益,最终选择信息增益最大的一组切分点作为实际使用的切分点.这样可以使每一维度数据与标签的互信息量达到最大,在保留有效信息的同时,将数据切分成段,用整形量化数值表示真实值所属区间.

3 研究方法

3.1 模型判断流程

模型的输入为处理之后的特征数据, 输出为是否处于睡眠状态, 这是一个二分类问题. 与普通的二分类问题不同的是, 用户的睡眠状态和非睡眠状态都具有一定的持续性, 不会频繁发生跳变. 如果在 t 时刻判断用户处于睡眠状态, 手机后台终止不必要的进程或模块, 则在之后的时间, 除非用户主动唤醒手机, 否则一直预测为睡眠状态, 因为在用户未唤醒手机之前, 即使重新预测为非睡眠状态, 手机后台进程已经被终止. 模型判断流程如图 1 所示. 当模型预测为非睡眠状态时, 下一时刻继续判断, 当预测为睡眠状态时, 则在用户主动唤醒手机之前, 一直预测为睡眠状态. 当用户处于睡眠状态后, 当前时刻 t , 用户静止持续时间 s_d 和用户息屏持续时间 sc_d 值不断增加, 环境光 l 值、充电状态 c 和接近传感器数据 d 也保持稳定, 所以睡眠开始之后很远的的数据意义不大, 对模型判断影响较大的为睡眠前和睡眠开始时刻之后附近的数据.

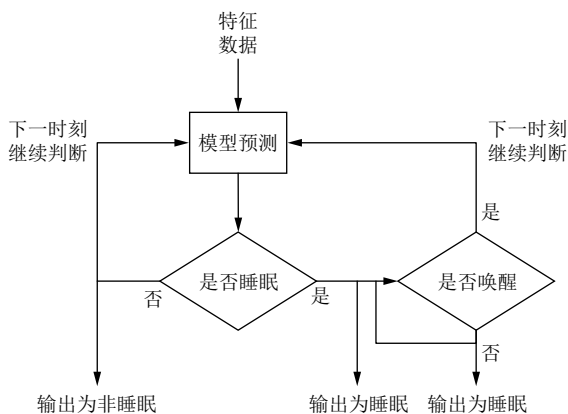


图 1 模型判断流程图

3.2 综合优化目标

在衡量模型性能时, 一方面, 希望在用户睡眠时手机尽可能长时间的处于低功耗模式, 达到更好的省电效率; 另一方面, 如果错误的判断用户进入睡眠状态, 则手机会终止相应的后台进程或模块, 当用户下次拿起手机时发现后台应用或模块被终止, 这会造成糟糕的用户使用体验, 所以需要同时考虑手机省电效率和用户使用体验. 为此, 本文采用覆盖率, 唤醒率作为模型的衡量指标, 公式为:

$$C = \frac{TP}{TP + FN} \quad (1)$$

$$W = \frac{C_F}{C_T + C_F} \quad (2)$$

模型输入为特征数据, 输出为睡眠状态, 用 0 表示非睡眠状态, 1 表示睡眠状态, TP 表示预测为 1 且实际为 1 的样本数, FN 表示预测为 0 但实际为 1 的样本数, C_F 表示错误进入睡眠的次数, 定义为用户实际入睡时刻之前预测状态从 0 跳变为 1 且在实际睡眠开始之前手机被唤醒的次数, C_T 表示正确进入睡眠的次数, 定义为用户实际入睡时刻之后预测状态从 0 跳变为 1 或在用户实际入睡时刻之前预测状态从 0 跳变为 1 但在实际睡眠开始之前手机一直未被唤醒的次数, 后者也为正确进入睡眠的原因是一直没有被唤醒, 后台终止进程的动作是正确的. 值得注意的是, 覆盖率的定义等价于召回率, 覆盖率越高越好, 高覆盖率意味着良好的省电效率, 但唤醒率定义并不等于虚警率, 虚警率表示负样本被唤醒率关注的是状态跳变的次数, 低唤醒率意味着低误杀率和好的用户体验. 在实际预测时, 当模型判断用户进入睡眠后, 除非用户主动唤醒, 否则一直处于睡眠状态, 即 $C_T=1$. 图 2 为一个夜晚的实际和预测序列举例.

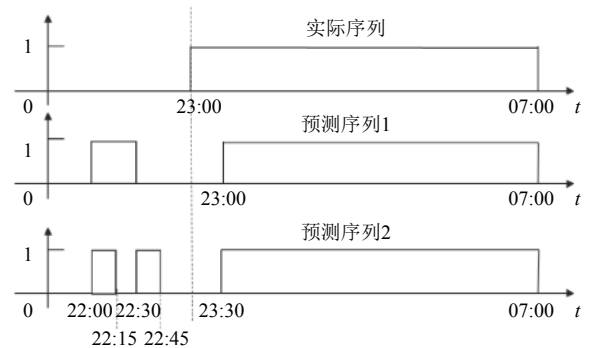


图 2 实际和预测序列

由图 2 可以说明, 用户晚上 23 点入睡, 睡眠持续时长 8 小时. 预测序列 1 判断用户在 22:00~22:30 和 23:30~7:00 处于睡眠状态. 预测序列 2 判断用户在 22:00~22:15、22:30~22:45 和 23:30~7:00 处于睡眠状态. 预测序列 1 和预测序列 2 的覆盖率相等, $C_1=C_2=0.9375$, 意味着手机处于低功耗状态时长占用户总睡眠时长的 93.75%. 预测序列 1 错误进入睡眠一次, 则 $W=1/2$, 预测序列 2 错误进入睡眠两次, 则 $W=2/3$, 错误预测的原因是用户暂时放下手机被系统误判为睡眠, 进而终止后台进程或模块, 当用户重新唤醒手机后, 发现后台进程或模块被终止, 这严重降低了用户的使用体验. 预测

序列 1 和预测序列 2 的虚警率一样, 均为 0.0625, 但实际上序列 2 造成了系统两次误杀后台进程或模块, 虚警率表示错误判为睡眠状态样本所占的比例, 而唤醒率关注误判次数, 预测结果频繁跳变对虚警率的结果影响可能不大, 但对唤醒率的影响会很大. 所以本文选用唤醒率作为模型的一个衡量指标. 在训练模型时, 要同时优化覆盖率和唤醒率, 在提高省电效率的同时保证用户的使用体验.

计算唤醒率需要计算状态跳变的次数, 而状态的跳变需要在一个时间窗口内发生, 只根据某个时刻的预测结果无法判断在此时刻睡眠状态是否发生变化. 考虑到用户在一个夜晚只会进入一次睡眠, 所以窗口长度取为一天, 可以保证在此窗口内用户实际发生了睡眠状态的跳变. 故在训练时以一天的数据为一个批次输入模型. 另外值得注意的是, 唤醒率的计算很难转换为可导的损失函数.

模型的优化目标为最大化覆盖率 C 和最小化唤醒率 W , 这是一个多目标优化问题^[12], 线性加权是多目标优化广泛使用的一种模型, SAW (Simple Additive Weighting) 是其中经典的一种线性加权求和方法, 它忽略了不同目标函数的不同单位和范围, 给不同的目标函数制定相应的权重, 将所有的目标函数进行线性加权, 用一个综合函数来代表总的优化目标^[13,14], 从而将多目标优化问题转化为单目标优化问题. 对于第 i 个目标函数 $f_i(x)$, 用 w_i 表示它的权重, 则多目标优化模型可转换为:

$$\max \sum_{i=1}^k w_i f_i(x), \text{ subject to } x \in S \quad (3)$$

其中, $k \geq 2$ 是优化目标函数个数, S 是决策变量 x 的可行域.

SAW 模型主要包括两个步骤, 第一为缩放, 将各个目标函数从原始值缩放; 第二为指定权重, 根据不同目标的优先级制定权重. 覆盖率 C 和唤醒率 W 的取值范围都为 $[0, 1]$, 无需缩放. 则多目标优化目标可表示为 $\max w_1(1-W) + w_2C$. 进一步的, 令 $\lambda = w_1/w_2$, 则综合优化目标为:

$$\max \lambda(1 - W) + C \quad (4)$$

式中, λ 可用来调整唤醒率和覆盖率的权重, λ 越大表示对唤醒率要求越严格.

3.3 LSTM 神经网络模型

模型目标为根据时序的特征数据持续判断用户是否处于睡眠状态, 睡眠状态通常与之前时刻用户的状态有关, 因此这是一个时序二分类问题. 循环神经网络

(Recurrent Neural Networks, RNNs)^[15] 是一类专门处理序列数据的神经网络模型. 循环神经网络的神经元输入不仅包括当前时刻的特征输入, 还包括上一时刻的输出. 这让循环神经网络可以将之前的信息和现在的任务联系起来, 拥有学习过去信息的能力. 但标准的循环神经网络难以学习到相隔比较远的信息, 即存在长期依赖问题. LSTM 神经网络是一种特别的循环神经网络, 被设计用来解决长期依赖问题. LSTM 神经网络通过门结构来控制信息的传递, 分别是输入门, 遗忘门和输出门. LSTM 每一个时间步的计算公式如下所示:

$$\begin{cases} i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \\ o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \\ h_t = o_t \odot \tanh(C_t) \end{cases} \quad (5)$$

其中, i_t 、 f_t 、 o_t 、 C_t 分别表示 t 时刻输入门的输出值、遗忘门的输出值、输出门的输出值、细胞记忆状态, h_t 为 t 时刻的隐藏状态, σ 是 Sigmoid 非线性激活函数, \odot 表示向量元素逐个相乘.

预测是否处于睡眠是一个二分类问题, 本文采取的神经网络结构如图 3 所示, 在 LSTM 模块之上再加一层全连接层, 最后输出维度为 1, 值为 0 或 1, 0 表示非睡眠, 1 表示睡眠.

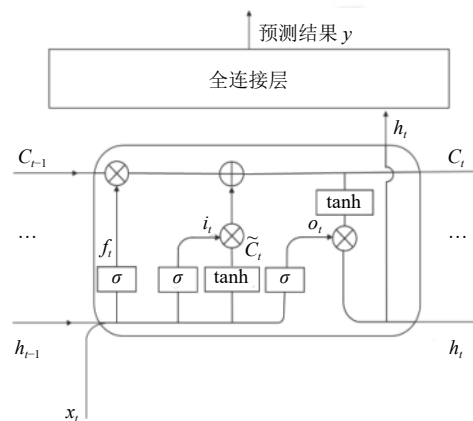


图 3 神经网络结构图

3.4 演化 LSTM 神经网络

LSTM 神经网络通常用梯度下降法训练, 要求损失函数可导, 通常取均方误差或交叉熵等. 本文中, 唤醒率关注的是睡眠状态的跳变, 很难转换为可导的损失函数, 无法直接用梯度下降法训练, 而演化算法可以用来解决不可导的优化问题. 演化算法也被称为进化算法 (Evolution Algorithm, EA), 演化算法的产生灵感

借鉴了大自然中生物的进化操作,自然界生物种群每一代都会有一定概率的变异出现,不同个体之间以一定概率进行交叉,产生下一代.环境会对变异和交叉产生的新个体进行选择,适应的被保留,不适应的被淘汰.演化算法在复杂的非线性最优问题、复杂的组合规划和整数规划问题等领域有较多的应用. Montana 等^[16]提出了用演化算法代替梯度下降法,对全连接神经网络可导优化目标进行优化.

本文选择用演化算法训练 LSTM 神经网络, LSTM 神经网络输入为特征数据,输出为是否处于睡眠状态,分别用 0 和 1 表示睡眠和非睡眠,优化目标为最大化 $\lambda(1-W)+C$, 其中 λ 越大表示对唤醒率的要求越严格. 当用户处于睡眠状态后,特征数据稳定,距离睡眠开始很远的的数据意义不大,对模型判断影响较大的为睡眠前和睡眠开始时刻之后附近的数据. 用演化算法训练时,随着迭代的不断进行,模型预测进入睡眠的时刻将不断逼近实际进入睡眠的时刻. 差分演化算法 (Differential Evolution, DE)^[17] 在连续空间内收敛速度较快,确定性高,本文采用差分演化算法对 LSTM 神经网络参数进行优化. 训练时,将一天的特征数据作为一个批次输入 LSTM 神经网络,得到唤醒率和覆盖率,计算综合优化目标,用差分演化算法对 LSTM 神经网络参数进行优化. 整体流程如图 4 所示,其具体步骤如下:

(1) 初始化

初始化种群,设定种群大小,将 LSTM 神经网络的全部参数扁平化为一个一维向量,则一个个体为一个一维向量. 对于每个个体 X_i , 采用高斯分布初始化:

$$\{X_i(0)|x_j^L \leq x_{i,j}(0) \leq x_j^U; i = 1, 2, \dots, N_p; j = 1, 2, \dots, D\} \quad (6)$$

其中, $X_i(0)$ 表示第 0 代第 i 个个体. $x_{i,j}(0)$ 表示第 0 代种群第 i 个个体第 j 维的数据, x_j^L 和 x_j^U 分别表示第 j 维的下界和上界,每个种群有 N_p 个个体,每个个体有 D 个维度.

(2) 变异

采用差分策略实现个体变异,随机选取种群中的两个不同的个体,将二者向量差缩放后与待变异个体进行向量合成,得到变异个体 $V_i(g+1)$.

$$V_i(g+1) = X_{r_1}(g) + F(X_{r_2}(g) - X_{r_3}(g)) \quad (7)$$

其中, r_1, r_2 和 r_3 是 3 个随机数,区间为 $[1, N_p]$, F 被称为缩放因子,为一个确定的常数, g 表示第 g 代种群.

(3) 交叉

交叉操作可以增加种群多样性,具体操作如式 (8)

所示,其中 $CR \in [0, 1]$ 为交叉概率, U 为交叉后个体.

$$U_{i,j}(g+1) = \begin{cases} V_{i,j}(g+1), & \text{if } \text{rand}(0, 1) \leq CR \\ x_{i,j}(g), & \text{otherwise} \end{cases} \quad (8)$$

(4) 选择

模拟自然界环境选择,将优化目标作为选择函数,采用贪婪选择策略,选择优化函数值较优的个体作为新的子代个体. 如式 (9) 所示,其中 f 是优化目标函数.

$$X_i(g+1) = \begin{cases} U_i(g+1), & \text{if } f(U_i(g+1)) \leq f(X_i(g)) \\ X_i(g), & \text{else} \end{cases} \quad (9)$$

完成上述 4 个步骤为一轮迭代,每轮迭代对种群进行更新,当达到一定代数则训练停止. 选择此时种群中最优的个体作为 LSTM 神经网络的最终参数.

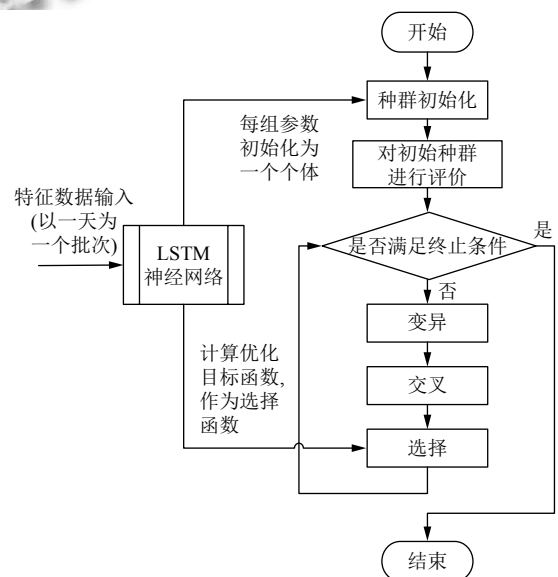


图 4 模型整体流程框图

当数据量较大时,若直接使用演化算法进行训练,每轮迭代时间过长,参数更新缓慢,训练速度很慢. 本文借鉴 mini-batch 训练方法,采用小批量训练. 小批量训练需要对演化算法进行改动,因为在评估上一代解和下一代解时,训练数据已经发生改变,则选择函数也发生改变,所以采用小批量训练时,在每次迭代结束后需要重新评估上一代解.

4 实验结果分析

4.1 与传统神经网络对比

演化神经网络和传统神经网络的对比结果如图 5 所示,分别选择 LSTM 神经网络和全连接神经网络 (Multi-Layer Perceptron, MLP). LSTM_EA 和 MLP_EA

分别表示用演化算法训练 LSTM 神经网络和全连接神经网络, LSTM_BP 和 MLP_BP 分别表示用梯度下降法训练 LSTM 神经网络和全连接神经网络. 演化神经网络以 $\lambda(I-W)+C$ 为学习目标, λ 用来调节覆盖率和唤醒率的权重, λ 越大表示对唤醒率的要求越严格. 图 5 中从左到右, λ 从 1.0 降低到 0.1, 间隔为 0.1, 演化种群大小为 50, $F=0.5$, $CR=0.9$. 传统神经网络以加权交叉熵为损失函数 (Tensorflow 中 `weighted_cross_entropy_with_logits` 函数), 通过调整正负样本权重达到间接的调整覆盖率和唤醒率的效果, 图中从左到右, 正样本与负样本权重比从 1:10 依次下降至 1:1, 梯度下降法选用动量法, 学习率为 0.01, 动量为 0.9. 全连接网络尝试不同隐藏层数, 因为输入维度简单, 选择一层隐藏层就有很好的效果, 隐藏节点为 8, LSTM 单元的输出维度为 4.

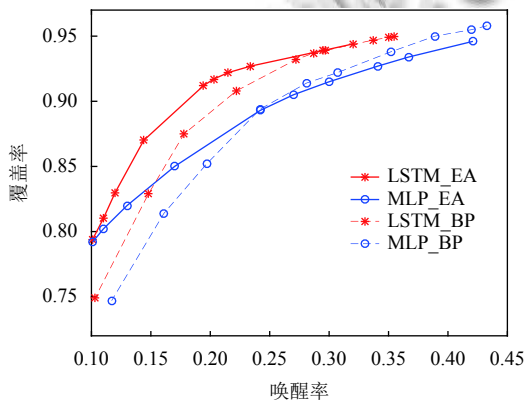


图 5 演化神经网络与传统神经网络对比

分析每条曲线的走势, 采用演化算法直接对覆盖率和唤醒率进行优化训练时, 随着 λ 从 1.0 降低到 0.1, 目标函数中唤醒率的权重不断下降, 曲线走势为从左下角到右上角, 即唤醒率不断上升, 覆盖率不断上升. 采用梯度下降法, 以加权交叉熵为优化目标, 通过调整正负样本权重间接的调整唤醒率和覆盖率, 随着负样本权重不断下降, 曲线走势为从左下角到右上角. 可以看到两类训练方法的曲线走势是相同的. 横向对比多条曲线, 采用同种训练方法时 LSTM 网络结果曲线在全连接神经网络结果曲线的上方, 这与 LSTM 网络能更好的处理时序数据的预期符合; 采用同种网络结构, 唤醒率较高时两种方法的结果曲线非常接近, 但在唤醒率较低时, 采用演化算法的结果要明显优于采用梯度下降法的结果, 这是因为采用演化算法直接以唤醒率为优化目标. 因此当对唤醒率要求较为严格时, 采用

演化算法训练得到的结果要更优. 而在实际的使用中, 高覆盖率意味着高省电效率, 低唤醒率保证用户的使用体验, 用户对后台误杀更为敏感, 所以我们更关注唤醒率较低的部分. 可以看到在唤醒率为 0.1 左右时, 采用演化算法训练神经网络比采用梯度下降法训练神经网络在覆盖率上有约 5% 的提升. 故在整体表现上, 演化 LSTM 神经网络的效果最优.

4.2 与传统机器学习模型对比

传统的机器学习模型也可以通过调整正负样本权重的方法间接的达到调整唤醒率和覆盖率的效果, 选择 SVM, 随机森林, 决策树 3 种模型和演化神经网络模型的结果对比如图 6 所示, SVM, Random Forest, Decision Tree 分别表示 SVM, 随机森林和决策树的结果曲线. 同传统神经网络方法一样, 改变正负样本权重比从 1:10 到 1:1, 覆盖率和唤醒率都变大. 在唤醒率较高时传统机器学习方法的结果与采用演化算法的结果差距不大, 其中 LSTM 网络效果最优, 其次为 SVM. 但在唤醒率较低时, 采用传统机器学习方法要明显劣于演化神经网络的方法. 即当对唤醒率的要求严格时, 采用演化算法直接优化的方法要更优. 这说明对传统神经网络和传统机器学习模型采用调整正负样本权重间接调整地唤醒率和覆盖率的方法都难以在低唤醒率时覆盖率达到较好的结果, 即难以在保证用户使用体验前提下达到更好的省电效率.

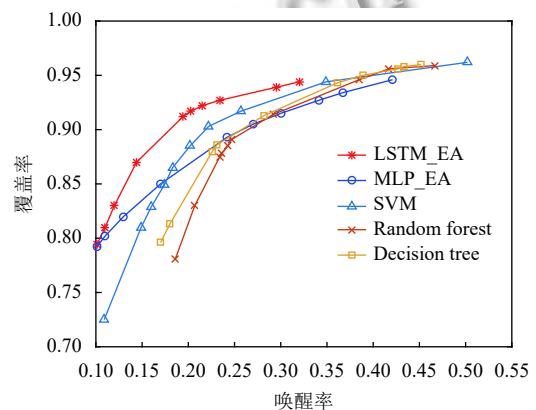


图 6 演化神经网络与传统机器学习模型对比

4.3 能耗分析

模型在服务器端训练, 演化算法种群大小为 50, 每轮迭代计算量主要为神经网络前向计算量, 而梯度下降法还包括反向求梯度运算, 反向计算量约为前向计算的量的两倍, 所以与采用梯度下降法相比, 演化算法计算时间复杂度约为梯度下降法的 16 倍, 但两者训练

耗时都不大,分别是秒级和分钟级,训练完成后部署至手机终端,我们主要关心在手机端的能耗.手机端能耗主要包括数据采集和存储能耗以及预测能耗,经过实际测验,数据采集和存储能耗极低,根据式(5),在预测时 LSTM 神经网络计算量为 $6 \times 4 \times 4 \times 8 + 3 \times 4 + 4 \times 1 = 400$ 次浮点乘运算和 $5 \times 4 + 1 = 21$ 次浮点加运算,此外还有 3 个 Sigmoid 运算和两个 tanh 运算,经过实际测试,在手机端模型预测部分耗时为 1 ms 以内,所以可实时预测用户睡眠状态.考虑到用户从清醒到入睡的过渡时间为分钟级,模型每 10 分钟判断一次用户是否进入睡眠状态,故模型在手机终端部署后计算能耗很低.

5 结论与展望

本文提出了使用演化算法训练 LSTM 网络的方法来判断用户是否处于睡眠状态,采用覆盖率和唤醒率作为模型的衡量指标,高覆盖率可保证手机省电效率,低唤醒率可保证用户的使用体验.采用演化 LSTM 神经网络模型处理时序特征数据,对覆盖率和唤醒率的综合优化目标进行优化,并借鉴梯度下降法中的 mini-batch 训练方法,采用小批量训练法.实验结果表明,采用演化算法训练 LSTM 神经网络的整体结果最优,在唤醒率较低时能够达到更好的覆盖率,相较于其他方法在低唤醒率时覆盖率提升约 5%.在未来的工作中,将会考虑不同用户使用习惯之间的差异,以及用户使用习惯的变化对模型的影响.

参考文献

- 1 Roomkham S, Lovell D, Cheung J, *et al.* Promises and challenges in the use of consumer-grade devices for sleep monitoring. *IEEE Reviews in Biomedical Engineering*, 2018, 11: 53–67. [doi: [10.1109/RBME.2018.2811735](https://doi.org/10.1109/RBME.2018.2811735)]
- 2 Melgani F, Bruzzone L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 2004, 42(8): 1778–1790. [doi: [10.1109/TGRS.2004.831865](https://doi.org/10.1109/TGRS.2004.831865)]
- 3 Auld T, Moore AW, Gull SF. Bayesian neural networks for internet traffic classification. *IEEE Transactions on Neural Networks*, 2007, 18(1): 223–239. [doi: [10.1109/TNN.2006.883010](https://doi.org/10.1109/TNN.2006.883010)]
- 4 Williams N, Zander S, Armitage G. A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification. *ACM SIGCOMM Computer Communication Review*, 2006, 36(5): 5–16. [doi: [10.1145/1163593.1163596](https://doi.org/10.1145/1163593.1163596)]
- 5 Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*, 1997, 9(8): 1735–1780. [doi: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735)]
- 6 Min JK, Doryab A, Wiese J, *et al.* Toss ‘N’ turn: Smartphone as sleep and sleep quality detector. *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*. Toronto, ON, Canada. 2014. 477–486.
- 7 Chen ZY, Lin M, Chen FL, *et al.* Unobtrusive sleep monitoring using smartphones. *Proceedings of the 2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*. Venice, Italy. 2013. 145–152.
- 8 Bai Y, Xu B, Ma YC, *et al.* Will you have a good sleep tonight? Sleep quality prediction with mobile phone. *Proceedings of the 7th International Conference on Body Area Networks*. Oslo, Norway. 2012. 124–130
- 9 Buysse DJ, Reynolds III CF, Monk TH, *et al.* The Pittsburgh sleep quality index: A new instrument for psychiatric practice and research. *Psychiatry Research*, 1989, 28(2): 193–213. [doi: [10.1016/0165-1781\(89\)90047-4](https://doi.org/10.1016/0165-1781(89)90047-4)]
- 10 Chen WX, Sano A, Martinez DL, *et al.* Multimodal ambulatory sleep detection. *Proceedings of 2017 IEEE EMBS International Conference on Biomedical & Health Informatics*. Orlando, FL, USA. 2017. 465–468.
- 11 Eiben AE, Smith JE. *Introduction to Evolutionary Computing*. Berlin: Springer, 2003.
- 12 林闯, 陈莹, 黄霁威, 等. 服务计算中服务质量的多目标优化模型与求解研究. *计算机学报*, 2015, 38(10): 1907–1923. [doi: [10.11897/SP.J.1016.2015.01907](https://doi.org/10.11897/SP.J.1016.2015.01907)]
- 13 Alrifai M, Skoutas D, Risse T. Selecting skyline services for QoS-based web service composition. *Proceedings of the 19th International Conference on World Wide Web*. Raleigh, NC, USA. 2010. 11–20.
- 14 Qi LY, Tang Y, Dou WC, *et al.* Combining local optimization and enumeration for QoS-aware web service composition. *Proceedings of 2010 IEEE International Conference on Web Services*. Miami, FL, USA. 2010. 34–41.
- 15 Graves A. *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin: Springer, 2012. 15–35.
- 16 Montana DJ, Davis L. Training feedforward neural networks using genetic algorithms. *Proceedings of the 11th International Joint Conference on Artificial Intelligence*. Detroit, MI, USA. 1989. 762–767.
- 17 Storn R, Price K. Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 1997, 11(4): 341–359. [doi: [10.1023/A:1008202821328](https://doi.org/10.1023/A:1008202821328)]