文章编号:1671-4598(2021)08-0047-04 DOI:10.16526/j.cnki.11—4762/tp.2021.08.010 中图分类号:TM912 文献标识码:A

基于布谷鸟搜索优化神经网络的锂电池荷电状态预测

陆佳伟，余世刚，魏新尧，王雪砚，朱 雅
（常州大学 机械工程学院，江苏 常州 213164）

摘要：锂电池荷电状态（SOC）的预测是电动汽车锂电池管理系统中最为关键的技术之一；为实现对SOC的高精度的预测，提出了一种基于布谷鸟搜索算法（CS）优化的误差反向传播（BP）神经网络的锂电池SOC预测方法，该方法的核心难点之一，在于优化BP神经网络的初始权值和阈值，从而可以改善易陷入局部最优的情况，减小算法对初始值的依赖；Matlab仿真结果表明，CS—BP神经网络算法的均方根误差值比BP算法的均方根误差值平均降低了0.010 6，CS—BP算法具有更高的预测精度和极强的泛化性能。

关键词：锂电池；荷电状态；布谷鸟搜索算法；BP神经网络

Lithium Battery Charge Status Prediction Based on Cuckoo Search Optimization Neural Networks

LU Jiawei, SHE Shigang, WEI Xinyao, WANG Xueyan, ZHU Ya
(School of Mechanical Engineering, Changzhou University, Changzhou 213164, China)

Abstract: The prediction of lithium battery charge status (SOC) is one of the most critical technologies in lithium battery management system of electric vehicles. In order to realize the high-precision prediction of SOC, a lithium battery SOC prediction method based on Cuckoo search algorithm (CS) optimization is proposed, the core of which is to optimize the initial weight and threshold of BP neural network, so as to overcome the disadvantages of local optimality and reduce the algorithm's dependence on the initial value. The MATLAB simulation results show that the average square root error value of CS-BP neural network algorithm is 0.010 6 lower than that of BP algorithm, and the CS-BP algorithm has better prediction accuracy and generalization performance.

Keywords: lithium battery; state of charge; cuckoo search algorithm; BP neural network

0 引言

为了减少大气污染，加强生态文明建设，国家大力扶持和推广新能源汽车。目前新能源汽车的性能、价格、使用寿命、安全性制约着动力电池系统的进一步发展。锂离子电池凭借工作电压稳定，体积小但储能大，无记忆效应，自放电低且安全环保等特点，已经成为新能源汽车动力源的新宠[1]。在实际使用中，要想把电池系统的动力性能充分发挥出来，有效延长电池的使用寿命，提高电动汽车的使用体验，那就对电池的荷电状态（SOC，state of charge）的估算精度提出了一定的要求。SOC的预测在电池管理系统（BMS，battery management system）中有着举足轻重的地位，精确的预测锂电池SOC可以有效避免电池的过充、过放，尤其是在寒冷的冬天，锂电池的续航能力会明显下降，这就需要精准预测锂电池剩余电量，方便使用者提前做好行程规划，合理使用剩余电量，同时保证整个电池系统的安全性，更好保护驾驶人员和乘车人员的生命安全。

SOC是用来描述电池在使用过程中，可以充入和放出容量的重要参数[2]，可以通过计算电池的剩余容量与电池总容量的比值获得，计算公式如式（1）所示：

$$SOC = \frac{Q_r}{Q_i} \times 100\% \tag{1}$$

式中，Q_r为电池剩余电量；Q_i为电池基准容量。影响电池SOC的因素不计其数，比如电池内阻、温度、充放电倍率等，所以在日常实际操作预测中，具有一定的挑战难度。目前研究者们惯用的一些SOC估算方法，比如安时积分法、开路电压法、卡尔曼滤波法、人工神经网络法[3]。徐尖峰等采用安时积分法对电池SOC进行估算，安时积分法相对简单，但是对电流的积分会伴随一定误差，且不断累积，而且需要知道SOC初始值[4]。李争等采用的是开路电压法，虽然简单易行，精度较高，但是电池组需要静置一段时间，无法用于实时估算[5]。文献[6]利用卡尔曼滤波法，需要

收稿日期:2021-01-18； 修回日期:2021-03-01。
基金项目:江苏省研究生科研与实践创新计划项目(KYCX21—2789;SJCX21—1201);常州大学机械与轨道交通学院SIETP基金(SIETP—2102)。
作者简介:陆佳伟(1995-),男,江苏启东人,硕士研究生,主要从事电动汽车锂电池管理系统方向的研究。
引用格式:陆佳伟,余世刚,魏新尧,等.基于布谷鸟搜索优化神经网络的锂电池荷电状态预测[J].计算机测量与控制,2021,29(8):47-50,88.

知道 SOC 信号模拟方程及测量方程，但也存在一些问题无法解决，比如单体的离散等。人工神经网络法快速易行，能有效克服以上几种方法的不足，有强大的自学能力，具有较强的非线性逼近能力，非常适合锂电池 SOC 的估算。赵钢等将放电电压、放电电流、电池表面温度及当前放电总量作为输入量，SOC 作为输出量，建立网络结构，在恒电流工况实验下得到了良好的预测结果[7]。

误差反向传播（BP，back propagation）神经网络凭借其通用性好的特点，而得到广泛的应用，但也存在着一些缺点，比如网络的初始权值和阈值，是随机产生的，会大大影响网络的输出特性，这样带来的结果就是易入局部最优，从而也会影响收敛速度，还有一些难点急需研究解决，如隐含层的设计，目前所用的都是根据前人的经验，加上设计者的反复试验，来确定隐含层节点数目。

为了降低初始权值和阈值对整个 BP 神经网络的影响，提高算法估算的精度，加快收敛的速度，本文提出一种基于布谷鸟搜索算法（CS，cuckoo search）优化 BP 神经网络的锂电池 SOC 预测方法。其中 CS 是通过模拟布谷鸟的寄生育雏的方式，寻找全局最佳的智能优化算法，它与神经网络结合在一起，可以改善陷入局部最优点的情况，还伴有收敛速度明显加快的效果。为验证 CS－BP 算法的真实预测效果，将它与 BP 算法作比较，结果表明该算法具有更好的预测能力。

1 BP 神经网络

BP 神经网络是一种简单的多层前馈神经网络，其模型是按照误差逆向传播算法训练的。算法由两个部分组成，一个是输入信号的前向传递，在这一过程中，输入信号先传递给隐含层，再传递给输出层，逐层处理，前面一层的输出结果，作为后面一层的输入，最后输出层得到的数据再与计算得到的期望值比较，若不满足误差要求，则转入第二个部分误差的反向传播，这一阶段，再根据预测误差的大小调整网络的阈值和权值，从而可以得到想要拟合函数的模型[8]。

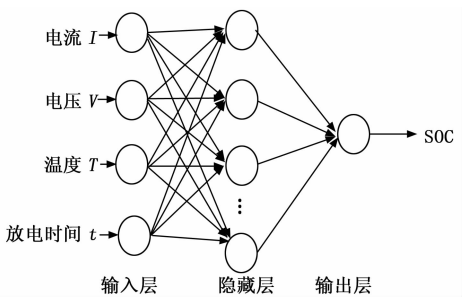

图 1 BP 神经网络的 SOC 估计模型

锂电池的电压 U、电流 I、温度 T、放电时间 t 是影响电池 SOC 的重要参数，作为神经网络输入层的输入矢量，输入矢量为 $[X_1, X_2, X_3, X_4]$，输入层的节点数为 $n=4$，输出矢量为 $[Y]$，节点数 $m=1$，隐含层的节点数的设计，利用经验公式：

$$l = \sqrt{n+m} + \alpha \tag{2}$$

其中：α 为 [1, 10] 范围内的任一常数，经过多次仿真试验，取 $l=12$ 时，网络的误差精度更高，收敛速度也更快。$x_i(i=1,2,\cdots,4)$ 表示其输入亦即该层的输出；隐含层由 12 个神经元组成，$z_k(k=1,2,\cdots,12)$ 代表的是隐含层的输出，$\theta_k(k=1,2,\cdots,12)$ 表示的是隐含层的阈值；这里输出层只有 1 个神经元，用 y 表示输出结果，即锂电池 SOC 的预测值，用 φ 表示输出层的阈值；用 $v_{ik}(i=1,2,\cdots,4;k=1,2,\cdots,12)$ 表示连接输入层与隐含层之间的权值；用 $w_k(k=1,2,\cdots,12)$ 代表连接隐含层与输出层之间的权值。隐含层采用 logsig 函数 $f(x)=1/(1+e^{-x})$ 为激活函数，输出层采用 Purelin 函数 $g(y)=y$ 为激活函数，性能函数采用均方误差性能函数（MSE，mean squared error）。算法步骤如下：

1）初始化。假设初始迭代次数 $t=0$，BP 神经网络初始权值和阈值为 $[-1,1]$ 内的随机数，即 $v_{ik}(t)\in[-1,1]$，$w_k(t)\in[-1,1]$，$\theta_k(t)\in[-1,1]$，$\varphi(t)\in[-1,1]$；

2）信号正向传播。输入一个样本 (X_m, Y_m)，其中 $m\in\{1,2,\cdots,M\}$，M 为样本数，$X=[I,V,T,t]^T\in R^{4\times m}$，$Y$ 表示 SOC 的真实值；

3）计算隐含层的输入与输出：

$$S_k = \sum_{i=1}^{4} v_{ki} \cdot x_i \tag{3}$$

$$z_k = f(S_k - \theta_k) \tag{4}$$

4）计算输出层的输入与输出：

$$S = \sum_{k=1}^{12} w_k \cdot z_k \tag{5}$$

$$y = f(S - \varphi) \tag{6}$$

5）误差逆向传播。计算输出层各节点误差：

$$Err = y(1-y)(Y-y) \tag{7}$$

6）计算隐层各节点误差：

$$Err_k = z_k(1-z_k)w_k Err \tag{8}$$

7）权重更新：

$$w_k(t+1) = w_k(t) + \alpha z_k Err \tag{9}$$

$$v_{ik}(t+1) = v_{ik}(t) + \alpha x_i Err_k \tag{10}$$

其中：α 为学习速率；

8）阈值更新：

$$\varphi(t+1) = \varphi(t) + \eta Err \tag{11}$$

$$\theta_k(t+1) = \theta_k(t) + \eta Err_k \tag{12}$$

式中，η 为学习率；

9）随机抽取样本进行学习，直到取完所有学习样本为止，进行下一步，否则重新返回步骤 2）开始执行；

10）计算每一个样本在输出层的误差函数 E_m：

$$E_m = (Y_m - y_m)^2 \tag{13}$$

对所有的样本的均方误差进行累计求和，就可得到基于 BP 神经网络模型的 SOC 估计的目标函数：

$$E = \frac{1}{M}\sum_{k=1}^{M} E_k = \frac{1}{M}\sum_{k=1}^{M}(Y_k - y_k)^2 \tag{14}$$

将既定的误差与 E 比较，如果 E 小于误差精度要求或者学习次数达到规定次数，则算法结束；否则更新学习次数 $t \leftarrow t+1$，并重新回到步骤 2）继续执行。

BP 网络易出现过度拟合，过度拟合是指将一组测试集的某些特性视为所有可能的测试集都具有的通用特性，从

而降低了泛化的有效性。在机器学习中，这种现象被称为“过度拟合”。一旦出现过度拟合情况，BP 神经网络的训练误差会继续减少，但测试误差反而增加。解决 BP 神经网络的过度拟合主要有两种思路，一种是“早停”：将数据分为训练、验证集。训练集改变梯度，更新各层权重和阈值，而验证集负责误差估算。在训练集误差减少，验证集误差增加时停止训练，并返回验证集误差最小时的权重和阈值。另一种思路是“正则化”。主要思路是通过增加一个描述复杂网络部分的函数来优化误差目标函数。

2　CS－BP 神经网络

2.1　布谷鸟算法

布谷鸟搜索（CS，cuckoo search）最早是在 2009 年，由剑桥大学的 X. S. Yang 等提出开发的一种新型自然启发式算法[9]，该算法思想是基于布谷鸟产卵繁殖方式和 Levy 飞行[10]。其中 Levy 飞行是一种随机游走，每一步方向完全任意，经常性的步长较小的短距离行走与偶尔性的步长较大的长距离行走相互交替，且步长是一个稳定的重尾分布。将 Levy 飞行注入到智能搜索算法中，在搜索开始后，长距离行走用于探索整个大范围，也就是全局大范围搜索占优势，不会陷入局部最优；在搜索后期，短距离行走用于小范围内仔细搜索，也就是局部搜索占优势，收敛于全局最优解，步长距离越小，得到的精度就会越高，从而算法会收敛于全局最大值。Levy 飞行中大量的短距离行走，有利于布谷鸟算法局部寻优，可以提高算法的精度，偶然性的长距离行走又可以将搜索的范围扩大，有助于算法全局寻优。Levy 飞行每步游走都由两个因素控制：一是游走方向，一般选取一个服从均匀分布的数；二是步长，步长服从 Levy 分布。为了简化和模拟布谷鸟的育雏行为，提出了 CS 算法遵循 3 条规则[9]：

1）一只布谷鸟一次只下一个蛋，并随机地放在某个鸟巢中。

2）只有能孵化出蛋的最佳鸟巢才会被保留，供下一代孵化。

3）有数量固定的、可用的宿主鸟巢，鸟蛋以一定的概率被宿主鸟发现，如果被发现，鸟蛋就会被抛弃。

在这 3 条规则的基础上，把布谷鸟寄住的鸟巢看成一个解，由此衍生出的布谷鸟算法的主要步骤为：

1）初始参数，随机生成 n 个鸟巢位置，计算各个鸟巢的适应度函数值，记录当前最优解；

2）利用 Lévy 飞行更新巢穴的位置，更新规则如下：

$$x_i(t+1) = x_i(t) + \alpha \otimes Levy(\beta) \tag{15}$$

式中，$x_i(t)$ 和 $x_i(t+1)$ 分别为第 t 次和 $t+1$ 次迭代第 i 个巢穴的位置；$\alpha=\alpha_0$（x_i（t）$-x_{\text{best}}$）是步长信息，用于控制搜索范围，$\alpha_0 = 0.01$ 是常数，x_{best}表示当前最优解；$\otimes$表示点对点乘法；Levy（β）是随机搜索路径，服从 Lévy 概率分布：

$$Levy \sim u = t^{-1-\beta}, 0 < \beta \leqslant 2; \tag{16}$$

3）将更新后的鸟巢位置与之前的鸟巢位置一一对比，保留测试结果较优的解；

4）根据鸟巢主人发现外来物种的概率，对被发现鸟巢的位置进行更新，得到新的位置代替被发现的：

$$x_i(t+1) = x_i(t) + r \otimes Heaviside(P_a - \varepsilon) \otimes (x_k(t) - x_j(t)) \tag{17}$$

式中，r 和 ε 为 [0,1] 区间内正态分布的随机数，$Heaviside(u)$ 表示阶跃函数，P_a 为发现概率，$x_k(t)$ 和 $x_j(t)$ 为第 t 次迭代时的两个随机解；

5）测试并比较新一代解的适应度值，对于较优的个体，保留传递到下一代；

6）如果到了最大迭代次数，则停止，否则反复迭代步骤 2）至步骤 5）。

2.2　CS－BP 神经网络

针对 BP 神经网络依赖初始权值阈值、易陷局部最小值等问题，提出利用 CS 优化 BP 神经网络，主要是先利用 CS 算法进行全局寻优，优化初始权值和阈值，将最优解再赋给 BP 神经网络，这样可有效避免陷入局部最优问题，同时提高了神经网络学习效率，优化可以分为三个部分：确定 BP 网络结构、CS 优化以及 BP 网络预测。第一部分根据数据样本初步确定 BP 网络结构，从而确定权值和阈值总数；第二部分采用 CS 优化寻找具备最优适应度值的个体，找到后对其进行反编码，从而可以得到最优的权值和阈值，可以构造 CS－BP 神经网络；第三部分就是对样本数据进行训练并输出预测结果。

总算法流程如图 2 所示。

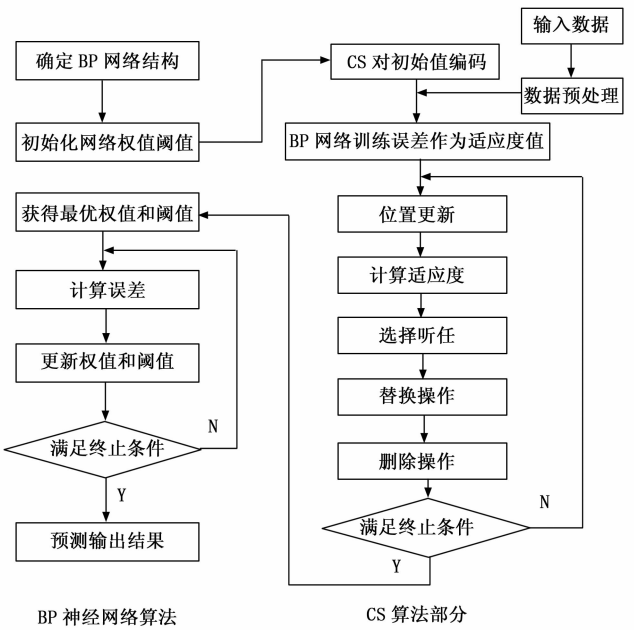

图 2　CS－BP 神经网络流程图

CS 算法优化 BP 神经网络算法步骤：

1）初始化，种群规模 n、发现概率 P_a 及最大迭代次数 $N_{\max}$ 等参数；

2）依据 BP 神经网络阈值和权值的特点，随机生成 n 个鸟巢的初始位置：$q_i^{(0)} = [x_1^{(0)}, x_2^{(0)}, \cdots, x_n^{(0)}]^T$，BP 神经网络训练得到的误差作为其适应度值，计算鸟巢初始位置的适应度值；

3）比较筛选并保留最佳鸟巢位置 $x_d^{(0)}$，根据前面所提更新规则式（15）更新鸟巢位置，将新的鸟巢位置与旧的

鸟巢位置比较，如果新的位置优于上一代，则更新位置，否则保留上一代位置；

4）测试改变后的鸟巢位置，从而可以获得一组较优的鸟巢位置 $e_k=[x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)}]^T$；

5）假如 $r<P_a$，保留此鸟巢位置不变，否则对被发现鸟巢位置进行随机更新，由此得到新的鸟巢位置，测试比较替代 e_k 中较差的鸟巢位置，可获得新的较佳鸟巢位置 $q_k=[x_1^{(k)},x_2^{(k)},\cdots,x_n^{(k)}]^T$；

6）计算出 q_k 中最优的鸟巢位置 $x_d^{(k)}$，以及此解的适应度值，同时检查是否满足终止条件，如果已经达到最大迭代次数，则停止搜索，并输出最佳位置 $x_d^{(k)}$，否则，返回继续执行步骤 3），直到满足算法终止条件。

3 实验仿真和结果分析

3.1 实验数据预处理

本实验用到的锂电池数据集，取自于美国国家航空航天局官网，公开的锂电池数据储存库，从数据库中选择 6、7 号电池数据集进行实验。本文中将 7 号电池充放电循环数据作为训练集，将 6 号电池充放电循环数据作为测试集。首先把两种电池放在同一环境内静置 30 min，然后在恒流模式下进行充电，一直充，电压达到 4.2 V，转入恒压充电模式，直到电池完全充满，充电结束；再静置 30 min 后，在 2 A 恒流模式下放电开始，直到 6 号电池的电压降至2.5 V，7 号电池的电压降至 2.2 V，结束放电。对 6 号、7 号电池都开展 168 次充放电循环，选取放电数据在 Matlab 环境下建立模型，得到如图 3 所示，两种锂电池的容量变化曲线图。观察可知，随着锂电池充放电循环次数的增加，电池内部活性材料发生损耗，锂电池的实际容量逐渐衰减。这里选取的是均方根误差（RMSE，root mean square error）和绝对误差作为衡量预测模型好坏的标准，计算公式如式（18）：

$$RMSE=\sqrt{\frac{1}{M}\sum_{i=k}^{M}(Y_k-y_k)^2} \tag{18}$$

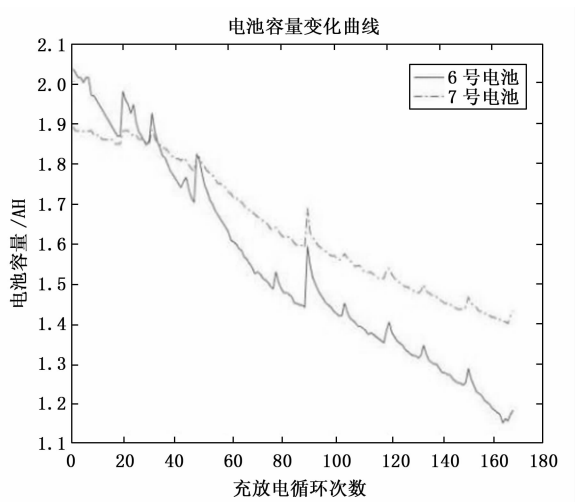

图 3 两种锂电池的容量变化曲线图

对原始数据进行无量纲化处理，通常使用归一化的方法将每个向量映射到 [0，1] 的区间内，这里通过式（19）做归一化处理：

$$x^*=\frac{x-min}{max-min} \tag{19}$$

式中，max 和 min 分别对应样本数据的最大值和最小值。

3.2 仿真分析

用 Matlab 建立 BP、CS－BP 锂电池 SOC 预测模型。电池的电压、电流、环境温度、放电时间作为输入参数，SOC 值作为输出，训练目标为 10^{-4}，隐含层的激活函数是 logsig，输出层的激活函数是 purelin，训练函数为 trainlm，学习率为 $\eta=0.01$。其中 CS 算法中发现概率 $P_a=0.25$，步长 $\alpha=0.05$。CS 算法的本质是一种寻求最优解的算法，对 BP 算法进行优化，建立了 CS－BP 预测模型。为了验证该算法的泛化能力，用建立好的两种算法对 6 号和 7 号电池数据集进行训练和仿真。对于这两种预测模型，图 4 给出两种算法的均方根误差曲线图，随机选取第 168 次的 CS－BP 算法的 $RMSE$ 值比 BP 算法的 $RMSE$ 值降低了 0.015 2，CS－BP 算法的 $RMSE$ 值比 BP 算法的 $RMSE$ 值平均降低了 0.010 6。

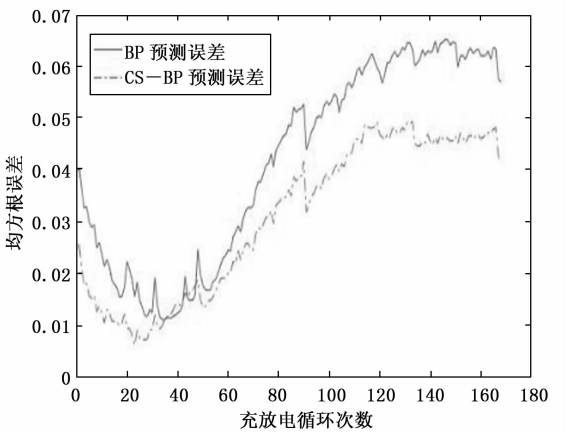

图 4 均方根误差曲线图

仿真后得到如图 5 所示的 CS－BP 算法的 SOC 预测值和真实值。从图 5 可以看出，CS－BP 算法的预测值比 BP 算法要更接近真实值，并且如图 6 所示，单次放电下电池 SOC 随时间变化预测误差曲线，CS－BP 算法的最大绝对误差比 BP 算法的最大绝对误差降低了 0.024 1，表明了 CS－BP 算法和 BP 算法相比，对锂电池 SOC 的估算更加精确。

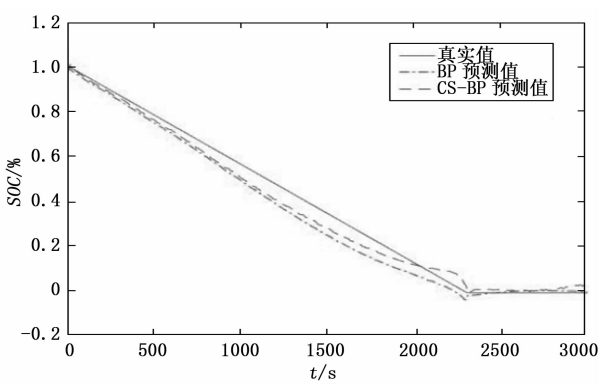

图 5 SOC 随时间变化曲线图

（下转第 88 页）