A LSTM-RNN method for the lithuim-ion battery remaining useful life prediction

YongZhi Zhang[1], Rui Xiong[1,*] , HongWen He[1], Zhiru Liu[2]

[1]Collaborative Innovation Center of Electric Vehicles in Beijing and National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, Beijing Institute of Technology, Beijing, 100081, China; [2]Power Battery System Research Institute, Shenzhen klclear technology co., LTD, Shenzhen, 518052, China.

* Corresponding author, rxiong@bit.edu.cn (R. Xiong)

***Abstract*—Prognostics and health management (PHM) can ensure that a battery system is working safely and reliably. Remaining useful life (RUL) prediction, as one main approach of PHM, provides early warning of failures that can be used to determine the necessary maintenance and replacement of batteries in advance. The existing RUL prediction techniques for lithium-ion batteries are inefficient to learn the long-term dependencies of aging characteristics with the degradation evolution. This paper investigates deep-learning-enabled battery RUL prediction. The long short-term memory (LSTM) recurrent neural network (RNN) is employed to learn the capacity degradation trajectories of lithium-ion batteries. The LSTM RNN is adaptively optimized using the resilient mean square back-propagation method. The developed LSTM RNN is able to capture the underlying long-term dependencies among the degraded capacities such that an explicitly capacity-oriented RUL predictor is constructed. Experimental data from one lithium-ion battery cell is deployed for model construction and verification. This is the first known application of deep learning theory to battery RUL predictions.**

Keywords—lithium-ion battery, deep learning, remaining useful life, long short-term memory

I. INTRODUCTION

Lithium-ion batteries have been widely applied in consumer electronics, including cell phones, laptops, electric vehicles, and military and aerospace electronics [1, 2]. However, their performance degrades over repeated charge/discharge cycles. In practice, failure is generally considered to occur if the battery's capacity decreases below the threshold value [3]. Because the battery's capacity and power both tend to drop sharply, its performance is unreliable at this point, and it should be replaced. Battery failure could result in degraded capability, unavailable operation, downtime, and even catastrophic occurrence.

Prognostics and health management (PHM), which is an enabling discipline, consists of methods and technologies to evaluate system reliability under real cycle life conditions to diagnose incipient faults and prognosis probable failure [4]. PHM of lithium-ion batteries enables users to make maintenance decisions in advance to avoid unexpected failure. Remaining useful life (RUL) prediction is a key approach of PHM [5]. RUL prediction methods for batteries can be divided into two main categories: model-based methods and data-driven methods.

Model-based methods are dependent on mathematics of the battery's degradation evolution, and mainly use advanced filtering methods including Kalman filter (KF), particle filter (PF) algorithms to prognose battery's RUL. He et al. [6] applied the Dempster–Shafer theory (DST) and PF method to predict the RUL of batteries. Model parameters were initialized using DST, and then updated using PF, and the RUL was predicted based on the available monitored data online. Because of the lack of knowledge of lithium-ion battery's degradation evolution, the mathematical model is generally constructed by fitting the degraded capacities of lithium-ion batteries, but this step could result in over-fitting. To solve the over-fitting, Xing et al. [7] developed an ensemble model to characterize the capacity degradation and predict the RUL of batteries. In addition, the Akaike information criterion (AIC) was introduced to penalize model complexity and thus avoid overfitting. Instead of the fitted model, Lyu et al. [8] introduced an electrochemical model-based PF framework to describe the capacity degradation of a battery and thus predict battery RUL. In the PF framework, model parameters that reflected battery degradation were seen as state variables.

Unlike model-based methods, data-driven methods do not need an explicit mathematical model to describe the degradation evolution of batteries and are only dependent on historical degradation data. Nuhic et al. [9] applied Support vector machine (SVM) to embed diagnosis and prognostics of system health with an aim to estimate the lithium-ion battery's state of health (SOH) and RUL. Wang et al. [10] introduced relevance vector machine (RVM) to derive the relevance vectors that can represent the capacity degradation trajectories. These relevance vectors were then utilized to fit a capacity degradation model to extrapolate the RUL. Besides the kernel techniques, neural network (NN) techniques were also employed to predict the lithium-ion battery's RUL. Much earlier, Liu et al. [11] developed an adaptive recurrent neural network (RNN), which can forecast system dynamic state and thus predict the lithium-ion battery's RUL. RNN is able to memorize the clues from the past with the feedback of previous state as inputs into the hidden layers. Based on the lithium-ion battery degradation data from NASA, Liu et al. [11] validated that the adaptive RNN showed better learning capability than the classical training algorithms, including RVM and PF methods. Since then, there have been no

This work was supported in part by the National Natural Science Foundation of China (Grant No. 51507012), Beijing Nova Program (Grant No. Z171100001117063), the National High Technology Research and Development Program of China (2015BAG01B01), the Joint Funds of the National Natural Science Foundation of China (Grant No. U1564206) and Key Project of Science & Technology Innovation Committee in Shenzhen, China(Grant No. 20150098).

978-1-5386-0370-3/17/$31.00 ©2017 IEEE

studies on employing NN techniques for RUL prediction of batteries except for one paper published by Rezvani et al. [12]. This lack of literature is surprising because the learning ability of NN has improved significantly with the recent development of deep learning techniques. Although simple RNN (SimRNN) is capable of connecting previous information to the present prediction, long-term dependency is the "Achilles heel" of SimRNN. The deep learning NN–long short-term memory (LSTM) RNN is specifically developed to solve the long-term dependency problem. It is able to remember information for long periods of time via the introduced gates. Guo et al. [13] presented LSTM RNN to predict streaming time series data against anomalies and change points. Qu et al. [14] developed a wind power prediction model by using the LSTM RNN. Simulation results showed that LSTM RNN had higher prediction accuracy than SVM. The capacity degradation data, which can cover thousands of charge/discharge cycles and represents the degradation evolution of batteries, can be regarded as long-term time series data. Therefore, LSTM RNN is used to learn the long-term dependency of the degradation data of capacities and predict the lithium-ion battery's RUL. The resilient mean square back-propagation (RMSprop) method is applied to train the network.

The rest of this paper is organized as follows: Section 2 describes the battery testing and degradation data. Section 3 describes the LSTM-RNN-oriented RUL prediction method. Section 4 discusses the experimental results of RUL predictions, and Section 5 presents conclusions.

II. Battery Testing and Degradation Data

To investigate the aging characteristics of lithium-ion batteries, a high-energy 18650 lithium-ion battery labeled NCR18650PF, which was manufactured by Panasonic, was applied. The nominal capacity of the cell is 2.7 Ah, and the nominal voltage is 3.6 V, with the lower and upper cut-off voltages being, respectively, 2.5 V and 4.2 V. The material on the positive electrode is Li(Ni, Co, Al)O2. The cell specifications are concluded in Tab. I.

TABLE I. CELL SPECIFICATION OF THE NCR18650PF BATTERY

Nominal capacity	2.7 Ah
Material of positive electrode	Li(Ni, Co, Al)O_2
Maximum continuous discharge current	10 A
Allowed voltage range	2.5–4.2 V
Operation temperature	
Storage	–20–50 ℃
Charge	10–45 □
Discharge	–20–60 ℃

The cell was cycled in the SOC range of 0–100% at 25 ℃ with 1C rate, where cycling device (Arbin, 30A 0–10 V) was applied. Temperature was monitored using a temperature sensor on the cell surface. Temperature was controlled with a climate chamber. At the end of each cycle, a 0.5C current was loaded on the cell to discharge all charges. The aging test was ended at 600 cycles, when the capacity retention is 79%. The trajectory of the lithium-ion battery's capacity degradation is presented in Fig. 1.

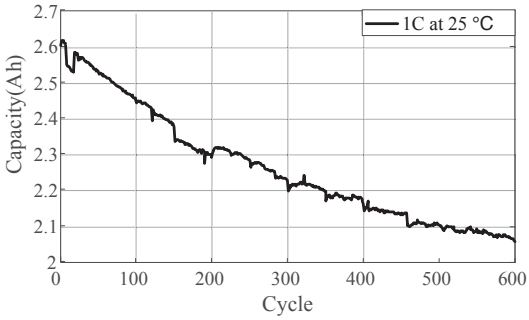

Fig. 1 Capacity degradation trajectories of battery at different temperatures and currents.

III. LSTM-RNN-Oriented RUL Prediction

The developed LSTM RNN predictor includes two main components: the long short-term memory cell, and the network parameter optimization using the RMSprop method.

A. LSTM RNN architecture

The network architecture of the LSTM RNN predictor is schematically presented in Fig. 2. All elements of the LSTM RNN are enumerated and described below. Note that the subscript *c* of each vector represents the cell, and the superscript (t) represents the moment *t*. $x^{(t)}$ is the input layer at present moment *t*, and $h^{(t-1)}$ is the hidden layer at the previous moment $t-1$.

- Internal state: The key to the LSTM cell is the internal state s_c. It is at the heart of each cell that has been linearly activated. The internal state s_c has a self-connected recurrent edge with fixed unit weight. The internal state can be regarded as a carrier to which information has been added or from which has been removed. This flow of information can be carefully regulated by structures called "gates." Gates are a distinctive feature of the LSTM approach and are able to optionally decide what information should pass though. A gate is a sigmoidal unit that is activated from the current input layer $x^{(t)}$ as well as from the hidden layer at the previous time step.
- Forget gate: The first step in the LSTM RNN is to decide what information is going to be discarded. This decision is made by a sigmoid layer called the "forget gate," f_c. It takes inputs of $x^{(t)}$ and $h^{(t-1)}$ and outputs a number between 0 and 1 for each state value in the internal state s_c. A 1 represents retaining the state value completely whereas a 0 represents discarding the value completely. The forget gate is calculated as follows:

$$f^{(t)} = \sigma\left(W^{fX} x^{(t)} + W^{fh} h^{(t-1)} + b_f\right) \quad (1)$$

- Input gate and input node: The next step is to decide what new information is going to be stored in the internal state. This step has two parts. First, a sigmoid layer called the "input gate," i_c, decides which values to update. Then, a tanh layer called the "input node," g_c, creates a vector of the new candidate state $\hat{s}_c^{(t)}$, which could be added to the

state. The equations to calculate the two outputs are

$$i^{(t)} = \sigma\left(W^{iX}x^{(t)} + W^{ih}h^{(t-1)} + b_i\right) \quad (2)$$

$$g^{(t)} = \tanh\left(W^{gX}x^{(t)} + W^{gh}h^{(t-1)} + b_g\right) \quad (3)$$

Combining Eqs. (1)—(3) to update the previous internal state $s_c^{(t-1)}$ into the current state $s_c^{(t)}$:

$$s^{(t)} = g^{(t)} * i^{(t)} + s^{(t-1)} * f^{(t)} \quad (4)$$

- Output gate: Finally, there is a sigmoid layer called the "output gate," o_c, that determines what information to output. After putting the internal state s_c through a tanh layer (to push the values to be between –1 and 1), it is multiplied by the sigmoid gate's output to calculate the remaining state value. This can be implemented as

$$o^{(t)} = \sigma\left(W^{oX}x^{(t)} + W^{oh}h^{(t-1)} + b_o\right) \quad (5)$$

$$h^{(t)} = \tanh\left(s^{(t)}\right) * o^{(t)} \quad (6)$$

where W and b values are the layer weights and biases, respectively. Therefore, if LSTM RNN detects a key feature from an input at an early stage, it easily carries this feature information over a long period of time, thus learning the hidden long-term dependencies [15].

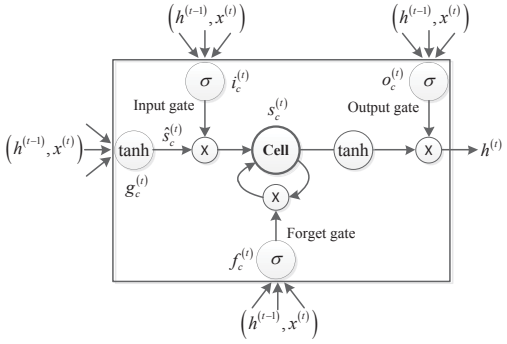

Fig. 2 Network architecture of the LSTM RNN predictor.

B. LSTM RNN training

The RMSprop method was developed by Tieleman and Hinton [16], who were inspired by the resilient back-propagation (Rprop) algorithm [17]. Rprop was proposed to solve inherent problems of the back-propagation algorithms, such as the low rate of convergence and strong oscillations for an acceptable solution. Although Rprop achieves is successful for full batch learning, it fails for mini-batch learning because the gradient is divided by a different number for each mini-batch. To solve this problem, RMSprop keeps a moving average of the squared gradient for each weight and uses the moving average to divide the gradient. This can force the dividend number to be very similar for adjacent mini-batches. An update of the network parameters θ (weights and biases) for RMSprop is as follows:

$$E\left[g^2\right]_t = \gamma E\left[g^2\right]_{t-1} + (1-\gamma)g_t^2 \quad (7)$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left[g^2\right]_t + \varepsilon}} g_t \quad (8)$$

where g_t denotes the gradient of the objective function with regard to the parameter θ at time step t; γ is a coefficient that determines the average level of the squared gradient; η denotes the learning rate; and ε is a smoothing term that avoids division by zero. Tieleman and Hinton [17] suggested γ to be set to 0.9, while a good default value for η was 0.001. Note that these two default values are both used in this paper.

IV. Results and Discussion

The LSTM RNN was constructed with one input layer that fed into an LSTM layer with 50 neurons. This layer in turn fed into another LSTM layer with 50 neurons, which then fed into a fully connected normal layer of 1 neuron with a linear activation function. For accurate comparisons, the SimRNN had the same structure as the LSTM RNN except that the LSTM layers were all replaced with the simple RNN layers. The dropout probability of the hidden layers of both LSTM RNN and SimRNN was 0.2. This meant for each neuron in the hidden layer, there was a 20% probability it would be neglected at each epoch of training.

As the battery capacity can be estimated online based on the measured current and voltage, the capacity is considered to be available and thus used to indicate the battery health state [5]. Because of manufacturing variances and application differences of lithium-ion batteries, the degradation trajectories of capacities can be quite different. It is thus difficult in practice to extract accurate features from the capacity degradation trajectory of one or several other cells for the particular one. To reduce the training difficulty, LSTM RNN is trained based on the capacity data of previous cycles and then used to predict the multi-step ahead capacity data in the following cycles. When the predicted capacity is lower than the threshold value, a failure is reported. For the simulation, two training strategies are conducted on the network, respectively. One strategy of LSTM RNN training covers 50% of the whole degradation data, and the other one covers 70% of the whole degradation data.

The lithium-ion battery failed at 506 cycles. The RUL PDF is generated by Monte Carlo method. Fig. 3 shows that the PDF of RUL prediction based on LSTM RNN is narrower when more data is used for training, indicating an increase of the confidence level of the prediction. Tab. II shows that, when the RUL is predicted at 253 cycles, the mean error of RUL prediction based on LSTM RNN is –3 cycles, which is a little delayed failure prediction. While the delayed failure cycles predicted by SVM are up to –21 cycles. Although the STD of the RUL prediction based on SVM is smaller, the 95% confidence bound, which is 511–546 cycles, covers a range after the failure. The delayed prediction of failure, which fails at an early failure report, is thus unexpected in practice. When more data is used for training, the RUL prediction by both LSTM RNN and SVM change into an early prediction of failure. LSTM RNN predicts a more accurate RUL, and the mean error is 15 cycles, while for SVM, the mean

error is 30 cycles. The STD of the RUL prediction based on SVM is only 2 cycles lower than that based on LSTM RNN. Also, the 95% confidence bound of the predicted RUL based on SVM, which is 458–496 cycles, fails to cover the real failure cycle.

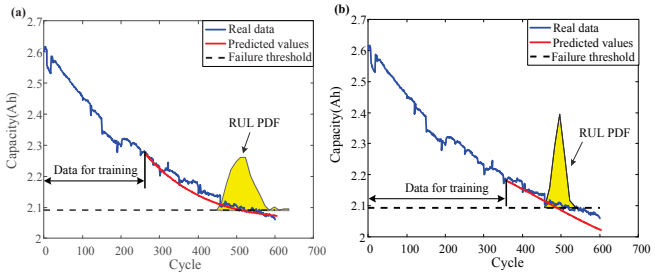

Fig. 3 LSTM RNN-based prediction results of RUL for cell 1: (a) prediction results at 253 cycles; (b) prediction results at 354 cycles.

TABLE II. PREDICTION RESULTS OF RUL FOR CELL 1

Method	Starting cycle	Mean error	STD	95% confidence bound
LSTM RNN	253	–3	28	[470, 564]
	354	15	11	[473, 511]
SVM	253	–21	9	[511, 546]
	354	30	9	[458, 496]

V. CONCLUSIONS

In this paper, an LSTM recurrent neural network has been used to construct a data-driven battery RUL predictor. The RMSprop technique, which is specifically designed for mini-batch training and able to quickly find an acceptable solution, is used to train the constructed neural network.

Testing data for one cell have been utilized to assess the prediction performance of LSTM RNN. The result shows that the predictor is accurate with a RUL prediction error of -6 cycles. The introduction of LSTM RNN to predict lithium-ion battery RUL not only helps to report an accurate early failure warning for the maintenance or replacement purposes of batteries, but also develops a new domain for the application of deep learning. The future work could focus on involving more degradation data of lithium-ion batteries working under more complicated conditions to further exploit the data mining ability of LSTM RNN in practice.

REFERENCES

[1] Y. Z. Zhang, R. Xiong, H. W He, and W. X. Shen, "Lithium-ion battery pack state of charge and state of energy estimation algorithms using a hardware-in-the-loop validation," IEEE Transactions on Power Electronics, vol. 32, no. 6, pp. 4421–4431, Jun. 2017.

[2] F. Sun, R. Xiong, and H. He, "A systematic state-of-charge estimation framework for multi-cell battery pack in electric vehicles using bias correction technique," Applied Energy, vol. 162, pp. 1399-1409, Jan. 2016.

[3] J. L. Zhang and J. Lee, "A review on prognostics and health monitoring of Li-ion battery," Journal of Power Sources, vol. 196, no. 15, pp. 6007–6014, Aug. 2011.

[4] H. M. Elattar, H. K. Elminir, and A. M. Riad, "Prognostics: A literature review," Complex & Intelligent Systems, vol. 2, no. 2, pp. 125–154, Jun. 2016.

[5] W. Waag, C. Fleischer, and D. U. Sauer, "Critical review of the methods for monitoring of lithium-ion batteries in electric and hybrid vehicles," Journal of Power Sources, vol. 258, pp. 321–339, Jul. 2014.

[6] W. He, N. Williard, M. Osterman, and M. Pecht, "Prognostics of lithium-ion batteries based on Dempster–Shafer theory and the Bayesian Monte Carlo method," Journal of Power Sources, vol. 196, no. 23, pp. 10314–10321, Dec. 2011.

[7] Y. Xing, E. W. Ma, K. L. Tsui, and M. Pecht, "An ensemble model for predicting the remaining useful performance of lithium-ion batteries," Microelectronics Reliability, vol. 53, no. 6, pp. 811–820, Jun. 2013.

[8] C. Lyu, Q. Lai, T. Ge, H. Yu, L. Wang, and N. Ma, "A lead-acid battery's remaining useful life prediction by using electrochemical model in the Particle Filtering framework," Energy, vol. 120, pp. 975–984, Feb. 2017.

[9] A. Nuhic, T. Terzimehic, T. Soczka-Guth, M. Buchholz, and K. Dietmayer, "Health diagnosis and remaining useful life prognostics of lithium-ion batteries using data-driven methods," Journal of Power Sources, vol. 239, pp.680–688, Oct. 2013.

[10] D. Wang, Q. Miao, and M. Pecht, "Prognostics of lithium-ion batteries based on relevance vectors and a conditional three-parameter capacity degradation model," Journal of Power Sources, vol. 239, pp. 253–264, Oct. 2013.

[11] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang, "An adaptive recurrent neural network for remaining useful life prediction of lithium-ion batteries," in Proc. of Annual Conference of the Prognostics Health Management Society, 2010, pp. 1–9.

[12] M. Rezvani, M. AbuAli, S. Lee, J. Lee, and J. Ni, "A comparative analysis of techniques for electric vehicle battery prognostics and health management (PHM)," SAE Technical Paper 2011-01-2247, 2011, doi: 10.4271/2011-01-2247.

[13] T. Guo, Z. Xu, X. Yao, H. Chen, K. Aberer, and K. Funaya, "Robust online time series prediction with recurrent neural networks," In Proc. of Data Science and Advanced Analytics (DSAA), 2016 IEEE International Conference, pp. 816–825, Oct. 2016.

[14] X. Y. Qu, X. N. Kang, C. Zhang, S. Jiang, and X. D. Ma, "Short-term prediction of wind power based on deep long short-term memory," In Proc. of Power and Energy Engineering Conference (APPEEC), 2016 IEEE PES Asia-Pacific, pp. 1148–1152, Oct. 2016.

[15] Q. Lyu and J. Zhu, "Revisit long short-term memory: An optimization perspective," In Proc. of Advances in Neural Information Processing Systems Workshop on Deep Learning and Representation Learning, 2014.

[16] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," Coursera: Neural Networks for Machine Learning, 2012.

[17] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The RPROP algorithm," In Proc. of IEEE International Conference, Neural Networks, pp. 586–591, 1993.