Fast Battery Capacity Estimation Using Convolutional Neural Networks

Yihuan Li[a], Kang Li[a,*], Xuan Liu[a], Li Zhang[b]

[a]*School of Electronic and Electrical Engineering, University of Leeds, LS2 9JT, UK*
[b]*School of Mechatronics and Automation, Shanghai University, Shanghai 200072, China*

Abstract

Lithium-ion batteries have been widely used in electric vehicles, smart grids and many other applications as energy storage devices, for which the aging assessment is crucial to guarantee their safe and reliable operation. The battery capacity is a popular indicator for assessing the battery aging, however, its accurate estimation is challenging due to a range of time-varying situation-dependent internal and external factors. Traditional simplified models and machine learning tools are difficult to capture these characteristics. As a class of deep neural networks, the Convolutional Neural Network (CNN) is powerful to capture hidden information from a huge amount of input data, making it an ideal tool for battery capacity estimation. This paper proposes a CNN based battery capacity estimation method, which can accurately estimate the battery capacity using limited available measurements, without resorting to other offline information. Further, the proposed method only requires partial charging segment of voltage, current and temperature curves, making it possible to achieve fast online health monitoring. The partial charging curves have a fixed length of 225 consecutive points and a flexible starting point, thereby short-term charging data of the battery charged from any initial state-of-charge can be used to produce accurate capacity estimation. To employ CNN for capacity estimation using partial charging curves is however not trivial, this paper presents a comprehensive approach covering time series-to-image transformation, data segmentation, and CNN configuration. The CNN based method is applied to two battery degradation datasets and achieves root mean square errors (RMSEs) of less than 0.0279 Ah (2.54%) and 0.0217 Ah (2.93%) respectively, outperforming

*Corresponding author
Email addresses: elyli2@leeds.ac.uk (Yihuan Li), K.Li1@leeds.ac.uk (Kang Li), elxl@leeds.ac.uk (Xuan Liu), zl_qee@163.com (Li Zhang)

Preprint submitted to Elsevier

existing machine learning methods.

Keywords: Battery capacity estimation, Convolutional neural networks, Lithium-ion batteries, Machine learning

1. Introduction

Due to continual falling costs, and features of high energy density, low self-discharge rate and long lifespan relative to other battery types, Lithium-ion batteries have been widely used as energy storage devices for electric vehicles (EVs), electric power grid, portable electronic devices, and many other applications (Li et al., 2019a; Zhang et al., 2016). However, undesirable side reactions and processes inside the batteries while in use will continuously degrade their performance, leading to capacity loss and increase of internal resistance (Couto et al., 2019). Therefore battery capacity and internal resistance are two important indicators for assessing battery ageing and performance degradation known as battery state of health (SOH). For example, SOH can be defined as $SOH_t = Q_t/Q_0$, where Q_0 is the rated capacity of a battery, and Q_t is the battery's maximum available capacity at current cycle t (Liu et al., 2019a).

Accurate capacity estimation provides insights into the SOH, thus plays a critical role in the battery management system, ensuring safe and reliable battery operation, preventing incipient failures and catastrophic hazards, and prolonging the battery service life (Liu et al., 2019b). However, the battery capacity can not be measured in real time, and a variety of estimation/prediction methods have been developed (Tang et al., 2019). These methods can generally be classified into three categories: model-based, differential analysis-based, and machine learning-based.

Model-based methods use battery electrochemical, electrical, or other empirical models to depict the battery dynamics, and estimate the battery capacity with a combination of observers or adaptive filtering algorithms (Garg et al., 2018; Ouyang et al., 2016). A comprehensive review of battery modelling methods including electrochemical models, reduced-order models, equivalent circuit models, empirical models and black-box models has been presented by Zhang et al. (2014). Liu et al. (2020a) have systematically evaluated the performance of three modelling techniques (i.e. electrochemical model, semi-empirical model and Gaussian process regression-based model) for calendar ageing prediction in terms of accuracy, generalization ability and uncertainty management. Zheng et al. (2016) propose to estimate the battery capacity by using proportional

integral observers based on an accurate electrochemical model, which can capture the spatiotemporal dynamics of batteries based upon the electrochemical principles. The equivalent circuit model with online identified parameters is used by Yu et al. (2019), and based on this model, an adaptive H infinite filter is applied to estimate the battery capacity. An empirical model that can reflect the battery dynamic capacity fading is proposed to predict the capacity degradation (Xu et al., 2016). However, the accuracy of the model-based capacity estimation methods is dependent on the quality of the estimation model. Unfortunately, it is difficult to build precise battery models due to the complex electrochemical reactions inside the battery under different operation conditions. Given the sheer complexity of the ageing mechanisms, simple lumped parameter models will lead to inaccurate estimation of the battery capacity.

The differential analysis-based methods correlate the features extracted from the differentiated curves of some electrical, thermal or mechanical parameters with battery capacity fade. For example, incremental capacity (IC) analysis and differential voltage (DV) analysis have been frequently used (Xiong et al., 2018). IC is calculated by differentiating the capacity change corresponding to its terminal voltage (dQ/dV) through charging or discharging the battery under a small and constant current rate. The DV curves (dV/dQ) is defined as the inverse of IC. The voltage plateaus can be easily identified from the IC/DV curves (peaks/valleys) after the differential operation. The features extracted from the curves such as IC peak position, peak shape, corresponding peak voltage/SOC, and peak area, are analyzed to estimate the battery capacity. For example, Li et al. (2018) have extracted five different features from the IC curves, the first two are peaks and the last two are valleys, the rest is the shoulder of the IC curves. The capacity is estimated by analyzing the position, value and associated area changes of these features. As described in (Weng et al., 2016), the IC peak values are tracked to estimate the capacity for single cells as well as battery packs. In (Zheng et al., 2018), three corresponding SOC positions are extracted from the SOC based IC and DV curves for battery capacity estimation. While Tang et al. (2018) use a regional voltage, which is calculated by the terminal voltage corresponding to the IC peak, for fast capacity estimation. However, the IC/DV analysis is sensitive to measurement noise and subject to operation temperature, further, it requires very low current rate, therefore their applications are severely constrained (Li et al., 2019b).

With the unprecedented progress of machine learning (ML) techniques and the documentation

of a large volume of battery test data worldwide, ML techniques have shown a greater potential in benefiting the battery capacity estimation. These methods are model-free, and do not need prior knowledge on the complex working principles of the battery. Various ML techniques have been applied to estimate the battery capacity fade, such as Neural Networks (NNs) (Zhang et al., 2019; Dai et al., 2018; You et al., 2016), Recurrent Neural Network (RNN) (Eddahech et al., 2012; Chaoui and Ibe-Ekeocha, 2017), Support Vector Machine (SVM) (Liu et al., 2018), Support Vector Regression (SVR) (Weng et al., 2013), and Relevance Vector Machine (RVM) (Hu et al., 2015; Guo et al., 2019), just to name a few. In (You et al., 2016), a NN with various optimization strategies is used for capacity estimation, by combining with the k-means clustering algorithm, achieving a RMSE of less than 2.44%. The inputs fed into this NN are the features manually extracted from the raw data. The RNN is used to predict the battery performance degradation in (Eddahech et al., 2012), the mean square errors for capacity and resistance prediction are 0.462 and 0.296 respectively. In (Liu et al., 2018), the nonlinear relationship between the extracted battery degradation features and battery capacity is established using the least square SVM method. The mean error for the capacity estimation is less than 5%. In (Weng et al., 2013), a linear programming based SVR is proposed to correlate the IC peaks with the faded battery capacity, the model developed using one cell data is able to estimate the capacity fade of other cells with absolute error less than 1%. In (Guo et al., 2019), a RVM based on particle swarm optimization is used to predict the battery capacity by modelling the relationship between the health feature and capacity, a relative error of less than 5% and 10% is achieved for single and multiple battery experiments respectively. In (Eleftheroglou et al., 2019), three ML-based methods are used for battery health prediction, and the uncertainty associated with each point prediction is quantified. Liu et al. (2020b) have proposed a hybrid method for battery capacity and remaining useful life prediction, where the long short term memory model is used to capture the long-term capacity degradation dynamics and the Gaussian process regression model is used for the uncertainty quantification caused by the capacity regeneration phenomena. Further, the Convolutional Neural Network (CNN) is applied to estimate the battery capacity using the measured voltage, current and the calculated cumulative capacity as inputs, of which the overall RMSEs are less than 2% on the NASA dataset (Shen et al., 2019).

However, the aforementioned ML-based estimation methods require either a non-trivial health

features extraction process or an extra cumulative capacity calculation process, rather than directly use the measurements (e.g. current, terminal voltage, surface temperature). In summary, the battery capacity estimation is still a challenging topic due to a range of time-varying situation-dependent internal and external factors. Traditional simplified models and machine learning tools are difficult to capture these characteristics. As a class of deep neural networks, CNN is powerful to capture hidden information from a huge amount of input data, making it an ideal tool for battery capacity estimation. In order to make full use of the information embedded in the direct measurements, while eliminating the necessity to manually extract features as well as fully charge a battery from a pre-defined state-of-charge, this paper proposes a CNN based battery capacity estimation method using partial charging segment with flexible starting point. The paper has the following four contributions.

- Firstly, the CNN based method will eliminate the need for priori knowledge and accurate battery physical model, making the method intelligent and adaptive for real-time capacity estimation.

- Secondly, the proposed method can deal with raw signals directly, mapping the measurements such as the terminal voltage, current, and surface temperature to the battery capacity, instead of relying on the pre-extracted health features. The representative features will be automatically learnt from the raw data.

- Thirdly, the paper introduces a novel data segmentation and time series-to-image transformation method which makes it feasible to use CNN for battery capacity estimation. Further, the proposed method only requires flexible partial charging segment of voltage, current and surface temperature curves, allowing fast and accurate capacity estimation, a key issue in real-time battery management.

- Finally, the proposed CNN-based method can self-learn its parameters and weights by using optimization algorithms like Adam. Once the parameters are properly learned offline, the model can be directly applied for fast online estimation.

The remainder of this paper is organized as follows. In Section 2, a brief introduction of the CNN is presented. Section 3 details the proposed CNN-based battery capacity estimation method,

including the signal-to-image transformation method and the proposed CNN architecture. Section 4 validates the proposed method on two battery degradation data-sets and the experimental results are presented and analyzed. Finally, Section 5 concludes the paper.

2. Convolutional Neural Network

2.1. Overview

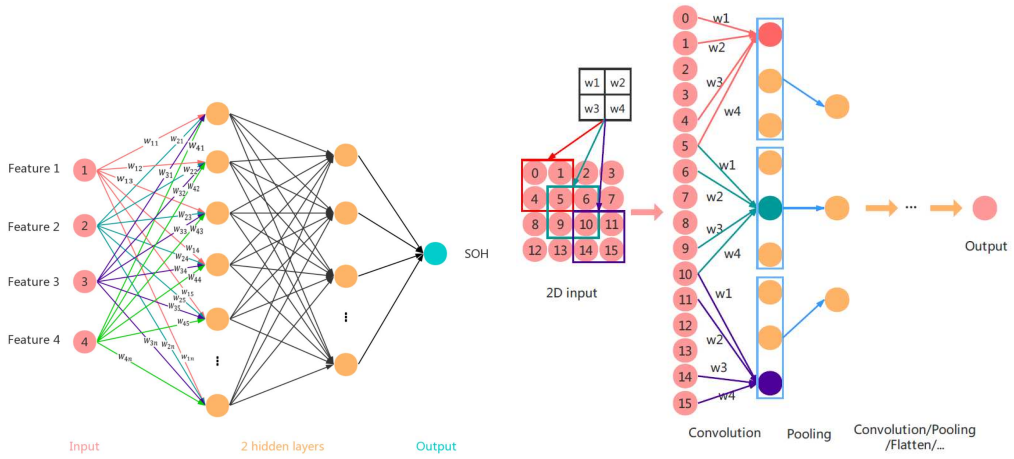
The Convolutional Neural Network is probably one of the most popular Neural Networks (NNs) in recent years. Compared with traditional deep neural networks (DNNs) with the same number of layers, the number of parameters (weights) of a CNN that are required to maintain the accuracy is significantly reduced, due to the sparse connectivity, shared weights, and pooling architectures. The sparse connectivity is achieved by making the size of filter smaller than the input, and enforcing a local connectivity pattern among neurons of adjacent layers. This architecture can reduce the overfitting risk, because the number of parameters are dramatically reduced. Shared weights refers to using the same weights for more than one activation function in a model, that is, each filter is used across the whole visual field. The architecture of shared weights has endowed the CNN with a property called equivariance, meaning that the output will change in the same way as the input changes (Liu et al., 2017). Then, the use of pooling architecture replaces the outputs of the convolutional layer with summary statistic, and this subsampling operation makes the output insensitive to small translation of the input.

CNNs are effective tools for extracting features from a high-dimensional data, and have been widely used in a range of fields, such as image processing, text classification, and speech recognition. These high-dimensional signals usually have high spatial or temporal correlations in adjacent variables, which can be effectively extracted through the convolution operations. Due to the fact that time series data is ubiquitous and is constantly generated in many engineering processes and in our daily life, there are imperative needs to develop efficient techniques to extract useful information from time series data. Considering the merits of CNNs in terms of automatic feature extraction and low overfitting risk, their applications in dealing with large amount of time series signals have also been investigated. For example, some reports have confirmed the potential of CNNs in extracting the representative features from time series data. In (Yang et al., 2015), a

CNN is used for solving a human activity recognition problem where the inputs of the network are multichannel time series signals collected from inertial sensors, and the outputs are related human activities. In this application, the filters in the CNN move along the temporal dimension for each sensor (each sensor corresponds to a row in the 2-dimensional input). In (Cui et al., 2016), a multi-scale CNN is used for time series data classification problems. The CNN architecture has multiple branches in its first layer that can extract features of different frequency and time scales. Further, CNNs have also been used for time series forecasting and estimation, and fault diagnosis.

2.2. CNN Architecture

A 3-layer fully connected feedforward neural network and a simple convolutional neural network are compared in Figure 1. To illustrate the differences in neuron connection between conventional neural networks and CNNs, Figure 1(b) reformulates the 2-dimensional input into a column, it is obvious that each output node in a convolutional layer is connected to a small subset of the inputs. This sparse connectivity is different from the fully connected NNs, and this



(a) 3-layer Neural Network (b) Simple CNN

Figure 1: (a) A fully connected 3-layer feedforward neural network. (b) A convolutional neural network, with convolutional layer as the first layer and pooling layer as the second layer. Here, the filter size is 2× 2 with stride (1,1), and the pooling size is 1× 3 with stride (1,1).

sparsity is achieved by replacing the matrix multiplication in NNs with convolutions (Borovykh et al., 2017). The filter (also called weight matrix) slides over the input space and generates a

set of output nodes, and each output node is calculated by convolving the input with the filter. The number of involved inputs for one output node is dependent on the filter size. All the output nodes produced by the same filter form a feature map, which is a matrix, while the number of feature maps is decided by the number of filters. In other words, all the nodes in one output feature map share the same weights. For the rth feature map in layer l, the node $\mathbf{C}_{r,l}^{a,t}$ at ath row and tth column can be calculated by

$$\mathbf{C}_{r,l}^{a,t} = f(\sum_{v} \sum_{i=0}^{m_l-1} \sum_{j=0}^{n_l-1} \boldsymbol{x}_{v,l}^{a\cdot s_l+i,t\cdot d_l+j} \boldsymbol{\omega}_{v,r,l}^{i,j} + \boldsymbol{b}_{r,l}) \tag{1}$$

where $\boldsymbol{x}_{v,l} \in \mathbf{R}^{p\times q\times v}$ denotes the input of the lth layer, of which the number of channels (also called depth) is v, and each channel has a size of $p \times q$. The inputs can be the initial input signal or the output of the preceding layer. For the output of the preceding layer, v refers to the number of feature maps in the $(l-1)$th layer. $\boldsymbol{\omega}_{v,r,l} \in \mathbf{R}^{m_l\times n_l}$ refers to the vth channel of the rth filter in layer l, with size $m_l \times n_l$ and stride set to (s_l, d_l). $\boldsymbol{b}_{r,l}$ is the bias for the rth feature map. $f(.)$ is the activation function that endows the network with the ability to learn complex nonlinear relationships in the data. The activation function used in this paper is Rectified Linear Unit (ReLU), which is given by

$$f(x) = \begin{cases} 0 & for \quad x < 0 \\ x & for \quad x \geq 0 \end{cases} \tag{2}$$

The pooling layer is a down-sampling process which reduces the size of the feature maps extracted in a convolution layer as well as the number of parameters introduced to the following layers by either max pooling strategy

$$\mathbf{P}_{r,l+1}^{a,t} = \underset{0\leq i\leq m_{(l+1)}-1, 0\leq j\leq n_{(l+1)}-1}{max} \{\mathbf{C}_{r,l}^{a\cdot s_{(l+1)}+i,t\cdot d_{(l+1)}+j}\} \tag{3}$$

or average pooling strategy

$$\mathbf{P}_{r,l+1}^{a,t} = \frac{\sum_{i=0}^{m_{(l+1)}-1} \sum_{j=0}^{n_{(l+1)}-1} (\mathbf{C}_{r,l}^{a\cdot s_{(l+1)}+i,,t\cdot d_{(l+1)}+j})}{m_{(l+1)} \times n_{(l+1)}} \tag{4}$$

where $(m_{(l+1)}, n_{(l+1)})$ is the size of the pooling region, while $(s_{(l+1)}, d_{(l+1)})$ is the strides of the pooling filter in layer $l+1$.

In the example shown in Figure1(b), only one filter is used, the filter size is 2×2 with stride being set to $(1, 1)$, which means four neurons of the input generate one output node, the pooling size is 1×3 with stride being set to $(1, 1)$, thus a 3-by-3 matrix becomes a 3-by-1 matrix after the pooling stage. Followed by adding selective convolution/pooling/flatten/fully connected layer, the final output will be obtained.

$$\mathbf{O} = f(\sum_{j=1}^{z} \boldsymbol{x}(j)\boldsymbol{\omega}(j) + \boldsymbol{b}) \tag{5}$$

where $\boldsymbol{x}$ denotes the input of the output layer, $\boldsymbol{\omega}$ and $\boldsymbol{b}$ are the weights and bias that connect the $\boldsymbol{x}$ and final output, respectively. In this paper, the output $\mathbf{O}$ is the estimated battery capacity. To evaluate the accuracy of the estimation results, the estimated capacity is compared with the reference value $Q(i)$ using the following measures: mean-square error (MSE) E_{mse}, root mean-square error (RMSE) E_{rmse} and mean absolute error (MAE) E_{mae}, which are defined as

$$e(i) = Q(i) - \mathbf{O}(\mathbf{i}) \tag{6}$$

$$E_{mse} = \frac{1}{N}\sum_{i=1}^{N}(e(i))^2 \tag{7}$$

$$E_{rmse} = \sqrt{E_{mse}} \tag{8}$$

$$E_{mae} = \frac{1}{N}\sum_{i=1}^{N}|e(i)| \tag{9}$$

where N is the sample size. Here one sample is referred to one image input to the CNN.

All the weights and bias are tunable parameters (θ) which are updated by minimizing the loss function $J(\theta)$ through an optimization algorithm. For prediction problems, it is common to use MSE loss function (Reed and MarksII, 1999), that is $J(\theta) = E_{mse}$. To update the parameters, the Adam algorithm (Kingma and Ba, 2014) is used in this work, which has been suggested as the default optimization method for deep learning applications (Ruder, 2016).

$$g_t = \nabla_\theta J_t(\theta_{t-1}) \tag{10}$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{11}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{12}$$

$$\tilde{m_t} = \frac{m_t}{1-\beta_1^t} \tag{13}$$

$$\tilde{v_t} = \frac{v_t}{1-\beta_2^t} \tag{14}$$

$$\theta_t = \theta_{t-1} - \frac{\alpha \cdot \tilde{m_t}}{\sqrt{\tilde{v_t}} + \epsilon} \tag{15}$$

where g_t is set to be the gradient of the loss function $J(\theta)$ at tth training iteration. m_t and v_t are the estimated first moment (the mean) and second moment (the uncentered variance) of the gradient respectively, and $\tilde{m_t}$ and $\tilde{v_t}$ are their bias-corrected values. β_1 and β_2 are exponential decay rates, while β_1^t and β_2^t are β_1 and β_2 to the power t. θ_t is the updated parameters.

3. Methodology

In this section, the proposed CNN-based battery capacity estimation method is described in detail. First, the method to transform the measured time series signals consisting of battery current, terminal voltage and cell temperature to a 3D image representation is introduced. Then the CNN is designed based on the classical LeNet-5 configuration (LeCun et al., 1998).

3.1. Time Series Signal Transformation

For other popular capacity estimation methods, the measurement data are not directly used for capacity estimation, and some features need to be extracted from the data first. For these methods, the estimation performance is dependent on both the number of extracted features and the way they are combined (Cai et al., 2019). However, it is not easy to effectively and efficiently extract features form the raw data. To make full use of the large volume of historic measurements, the correlations among different measured variables at different sampling periods have to be investigated. This is however not a trivial task to handle manually. CNNs however can overcome this difficulty, but to apply CNN for capacity estimation, a transformation stage is first required, which is elaborated below.

As illustrated in Figure 2 which shows one complete charging and discharging cycle, n continuous data points are extracted from each measured variable (e.g. current, voltage, cell temperature) in

one charging and discharging cycle, and this operation is executed for M times. That is, M data chunks are generated for each variable in a cycle. Then the n points in each chunk are converted into a $\sqrt{n} \times \sqrt{n}$ matrix, and the matrix represents a 2-dimensional image. The three variables together form 3-dimensional images ($\sqrt{n} \times \sqrt{n} \times 3$) as the input samples to the CNN, with each variable being associated to one channel of the image. In this signal-to-image transformation, the number of data chunks M for each variable in a cycle, which contains L data points in total, is determined by the segmentation length n and the overlap size c between two adjacent data chunks.

$$M = floor(\frac{L-n}{n-c}) + 1 \tag{16}$$

The function floor(.) gives the greatest integer less than or equal to the input parameter. The samples generated from the same cycle correspond to the same capacity value. Since each sample intercepted from the full charging and discharging cycle corresponds to a part of the charging/discharging process, based on the model trained with such samples, it is possible to estimate the capacity of a battery only using a part of the charging/discharging data. Besides, the part of the charging/discharging curve intercepted from the whole cycle may start at any point, meaning that the trained model can estimate the capacity of the battery charged/discharged from any unknown initial SOC.

As shown in Figure 2, these measurements have different scale, which may slow the training process and degrade the estimation accuracy. Thus, data normalization is applied to process the signals before feed them into the network. In this work, the min-max normalization strategy is adopted, which retains the original distribution of data and all transformed data fall into the range of [-1,1], reflecting both the charging and discharging phases. The normalized value z of the set of measurement $\mathbf{x}^k = \{x_i^k, x_2^k, ..., x_n^k\}$ is calculated by

$$z_i^k = \frac{x_i^k - min(\mathbf{x})}{max(\mathbf{x}) - min(\mathbf{x})} \times 2 - 1, \quad i \in 1, ..., n \tag{17}$$

where $\mathbf{x}^k$ refers to the measurements of the kth data chunk, $\mathbf{x}$ denotes a collection of all data points used in the training, n is the number of data points per chunk.

After the data normalization and time series to image transformation step, the final input of

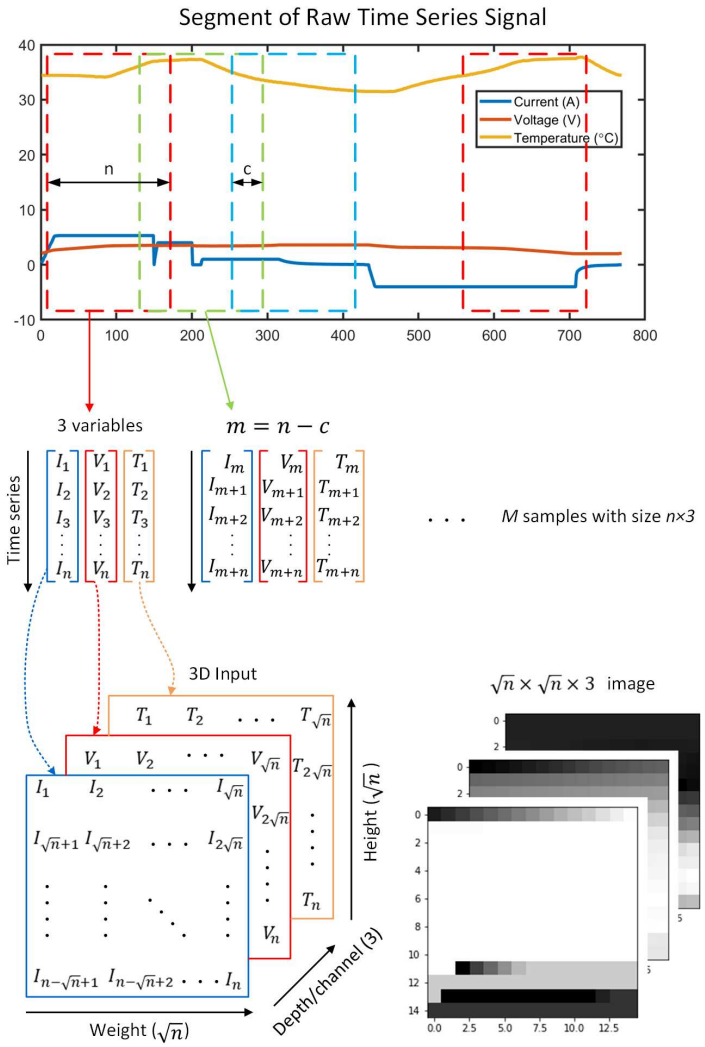


Figure 2: Transformation method : convert the time series measurements to 3-D images

the CNN is illustrated in Figure 3. This data transformation method is simple to use because no predefined parameters are required, and it is an enabling block to apply the CNNs for time series signals.

3.2. Model Construction

With the transformed 3D data, the CNN can then be trained to estimate the battery capacity. Considering that the size of the input sample is relatively small ($15 \times 15 \times 3$), a rather simpler CNN structure is adopted. In this study, the architecture of the proposed model is designed

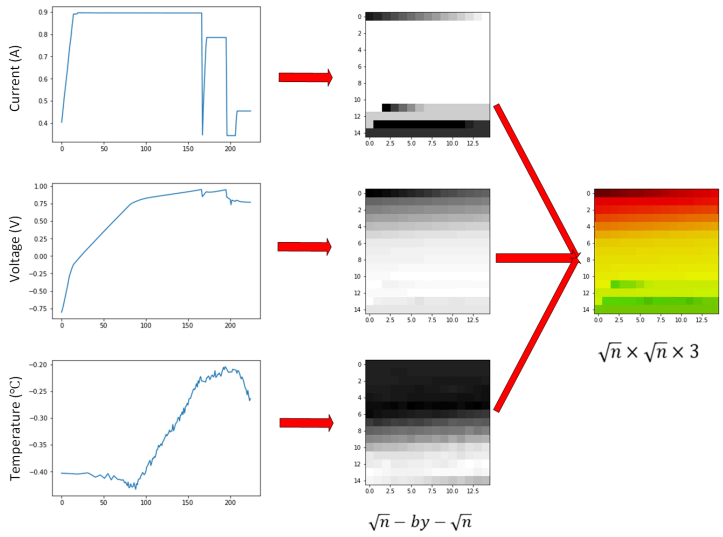


Figure 3: The input of the proposed CNN

based on LeNet-5 (LeCun et al., 1998), a classical and effective CNN structure. The proposed CNN architecture consists of two alternating convolutional and pooling layers, followed by two convolutional layers, and finally a flatten layer and two fully connected layers are utilized. The zero-padding method (Li et al., 2015) is used in the last two convolutional layers to control the size of the feature maps. This architecture is graphically illustrated in Figure 4, and the output shape and number of parameters for each layer are summarized in Table 1. The input image has

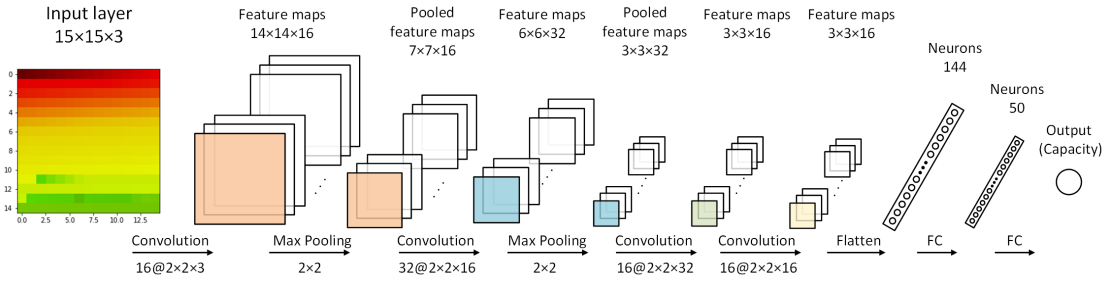


Figure 4: Proposed CNN architecture with $15 \times 15 \times 3$ inputs for battery capacity estimation

three dimensions: weight, height and depth, where the weight and height are determined by the n data points in each data chunk, and the depth is determined by the number of variables. Take $n = 225$ for example, the weight and height of the input both equal to 15, and the depth is 3 (the 3 variables are current, voltage and cell temperature). The input is fed into the CNN model, it is

Table 1: Layer configurations and parameters of CNN models

Layer	CNN Models	Stride	Output shape	Parameters	Activation
L1	Conv(16@$2 \times 2 \times 3$)	(1,1)	$(14 \times 14 \times 16)$	208	ReLU
L2	Maxpool(2×2)	(2,2)	$(7 \times 7 \times 16)$	0	ReLU
L3	Conv(32@$2 \times 2 \times 16$)	(1,1)	$(6 \times 6 \times 32)$	2080	ReLU
L4	Maxpool(2×2)	(2,2)	$(3 \times 3 \times 32)$	0	ReLU
L5	Conv(16@$2 \times 2 \times 32$)	(1,1)	$(3 \times 3 \times 16)$	2064	ReLU
L6	Conv(16@$2 \times 2 \times 16$)	(1,1)	$(3 \times 3 \times 16)$	1040	ReLU
L7	Flatten	-	144	0	ReLU
L8	FC	-	50	7250	ReLU
L9	FC	-	1	51	ReLU

first convolved with the filters. The filter design, denoted as $M@w \times h \times d$ means that there are M filters with the size of $w \times h \times d$ for a particular convolutional layer, d is determined by the depth of this layer's input. The M feature maps generated by the convolutional layer are then passed through the pooling layer with 2×2 pooling region and output smaller feature maps. After the last convolutional layer, the flatten layer transforms the $i \times j \times k$ features into a vector with $i \times j \times k$ neurons. Finally, the vector is fed into fully connected (FC) layers to calculate the final output, which is the capacity value.

4. Experiment and Analysis

In this section, the proposed CNN-based capacity estimation method is applied to two battery experimental datasets. The first is sourced from 124 commercial lithium-ion batteries cycled to failure under fast-charging conditions (Severson et al., 2019), and the other is the Oxford Battery Degradation Dataset (Birkl, 2017). During the training process, the number of the maximum training epochs is set to 80 and the mini-batch size is set to 128 samples. Early stopping method with patience set to 4 is used to avoid overfitting problem. Further, the learning rate is set to 0.001.

Table 2: Summary of the policies for charging the cells from 0% to 80% SOC, and estimation errors on test batteries for each trial in Case 1

	Battery	Charging policies	Cycles	RMSE (Ah)	MaxE (Ah)	MAE (Ah)
Group 1 (test dataset for Trial 1)	1	'5C(67%)-4C'	1008	0.0068	0.0415	0.0046
	2	'5.3C(54%)-4C'	1062	0.0092	0.0533	0.0061
	3	'5.6C(19%)-4.6C'	1266	0.0147	0.0835	0.0097
	4	'5.6C(36%)-4.3C'	1114	0.0107	0.0617	0.0074
Group 2 (test dataset for Trial 2)	5	'5.6C(19%)-4.6C'	1047	0.0093	0.0484	0.0061
	6	'5.6C(36%)-4.3C'	827	0.0202	0.0798	0.0158
	7	'3.7C(31%)-5.9C'	666	0.0279	0.0896	0.0206
	8	'4.8C(80%)-4.8C'	1835	0.0107	0.0601	0.0072
Group 3 (test dataset for Trial 3)	9	'5C(67%)-4C'	827	0.0061	0.0330	0.0043
	10	'5.3C(54%)-4C'	1038	0.0055	0.0404	0.0034
	11	'4.8C(80%)-4.8C'	1077	0.0086	0.0424	0.0060
	12	'5.6C(19%)-4.6C'	816	0.0062	0.0388	0.0038
Group 4 (test dataset for Trial 4)	13	'5.6C(36%)-4.3C'	931	0.0089	0.0444	0.0060
	14	'5.6C(19%)-4.6C'	815	0.0079	0.0476	0.0053
	15	'5.6C(36%)-4.3C'	857	0.0081	0.0450	0.0053
	16	'5.9C(15%)-4.6C'	875	0.0083	0.0503	0.0048

4.1. Case 1: 124 Commercial Cells

In this public available dataset, the 124 lithium iron phosphate (LFP)/graphite cells are manufactured by A123 System (APR18650M1A), with a nominal capacity of 1.1 Ah and a nominal voltage of 3.3 V. All the cells in this dataset are charged at a constant temperature of 30°C with the fast-charging policy, namely "C1(Q1)-C2". In this charging scheme, the cell is first charged at a constant current (CC) C1, and when the SOC reaches Q1, the CC switches to C2. This CC step ends at 80% SOC, after which the cells are charged at 1C until the battery voltage reaches its upper cutoff potential 3.6 V. Then a constant voltage (CV) mode continues until the charge current falls to 22 mA. All the cells are discharged under a CC-CV protocol, discharging at CC of 4C until the cell voltage falls to 2.0 V with a current cutoff of 22 mA.

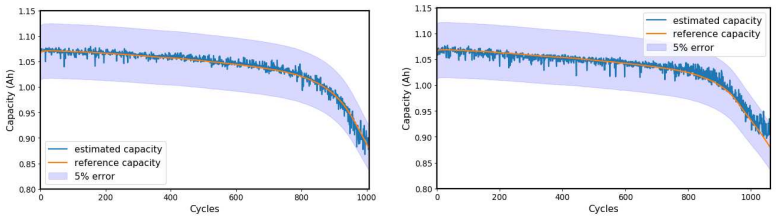
In this work, data of the first 16 batteries in dataset 'batch3' are used. These 16 batteries are divided into 4 groups, each group contains 4 different batteries. The detailed policies applied to

charging these 16 cells from 0% to 80% SOC are summarized in Table 2, and the test cells in each trial are given in details. Each trial, samples generated from 3 of the 4 groups are first shuffled and randomly split into a training set and a validation set with the ratio of 7:3, which are then used to train the CNN model. The remaining group is finally used for testing the performance of the trained CNN model.

The size of one sample inputted to the CNN is $15 \times 15 \times 3$, and the training samples are first shuffled before they are fed into the network to train the model. Once trained, the model is used to estimate the capacity on the test group, some of which may have slightly different charging policies from the training dataset. For a test cell, 225 consecutive data points are selected for each variable (i.e. current, voltage and temperature) in each cycle, and the selected data chunks from the three variables are then transformed to a 3D image as an input to the network. It should be noted that the selected data chunks for the three variables are from the same time segment, though the starting point can be randomly chosen. The training and testing procedures are repeated 100 times, and the estimation results with the lowest RMSE on the test dataset among the 100 runs in each trial are illustrated in Figure 5 - 8 and listed in Table 2. In figures 5 - 8, the estimated capacities are compared with reference values for all the test cells, where the blue region represents 5% error boundaries of the actual capacities. In Table 2, RMSE, max error (MaxE) and MAE of the estimation results are all listed. It is shown that the capacity degradation trend is well traced, and the RMSE, MaxE and MAE of the capacity estimations for the test batteries are less than 0.0279 Ah, 0.0896 Ah and 0.0206 Ah(2.54%, 8.15% and 1.87% of the rated capacity), respectively. It is noted that the estimation errors of battery 7 are slightly bigger than other battery cells. This is due to the fact that the charging policy of battery 7 is not included in the training dataset. However, as shown in Table 2, the estimation performance reveals that satisfactory results can still be achievable when the proposed method is applied to estimate the capacity of a battery whose charging policy is different from the training dataset.

4.2. Case 2: Oxford Dataset

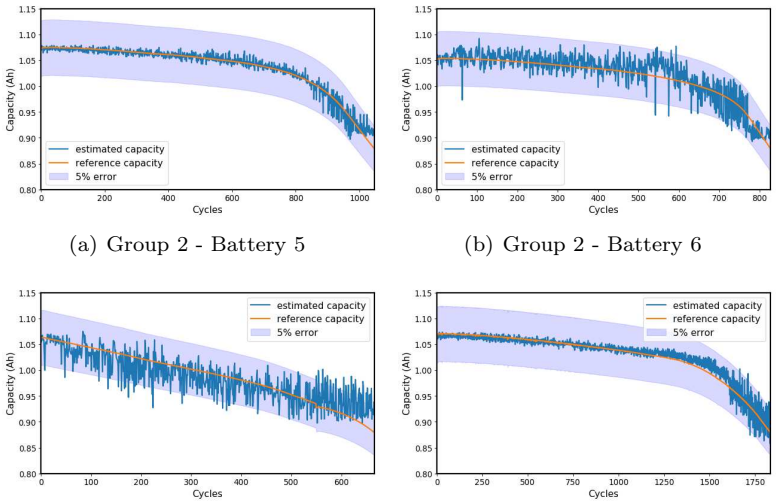
In this dataset, aging experiments are applied to eight commercial Kokam pouch cells, with a nominal capacity of 0.74 Ah. The dynamic driving profile used to degrade these cells is the Artemix urban drive cycle, and a characterization cycle is carried out every 100 dynamic cycles.



(a) Group 1 - Battery 1 (b) Group 1 - Battery 2

(c) Group 1 - Battery 3 (d) Group 1 - Battery 4

Figure 5: Capacity estimation results on Group 1 (Group 2, 3, 4 for training, and Group 1 for testing)

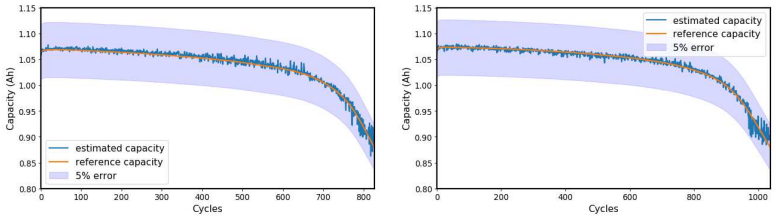


(a) Group 2 - Battery 5 (b) Group 2 - Battery 6

(c) Group 2 - Battery 7 (d) Group 2 - Battery 8

Figure 6: Capacity estimation results on Group 2 (Group 1, 3, 4 for training, and Group 2 for testing)

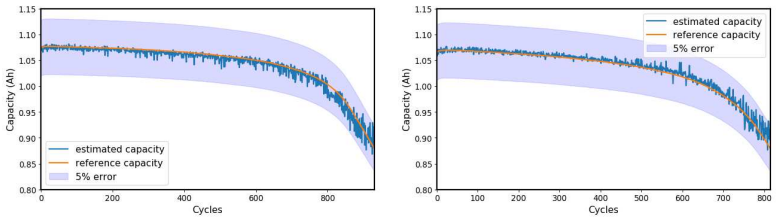
The data collected from the characterization cycles, which charge and discharge the cells under a CC profile (1C) and the thermal chamber is set at a constant temperature of 40°C, are used for capacity estimation (Birkl, 2017). Each time data of 7 cells are used to train the model, of which the generated samples are shuffled and split into training and validation sets with the ratio of 7:3, while data from the remaining cell is used for testing. The capacity estimation procedure in



(a) Group 3 - Battery 9 (b) Group 3 - Battery 10

(c) Group 3 - Battery 11 (d) Group 3 - Battery 12

Figure 7: Capacity estimation results on Group 3 (Group 1, 2, 4 for training, and Group 3 for testing)

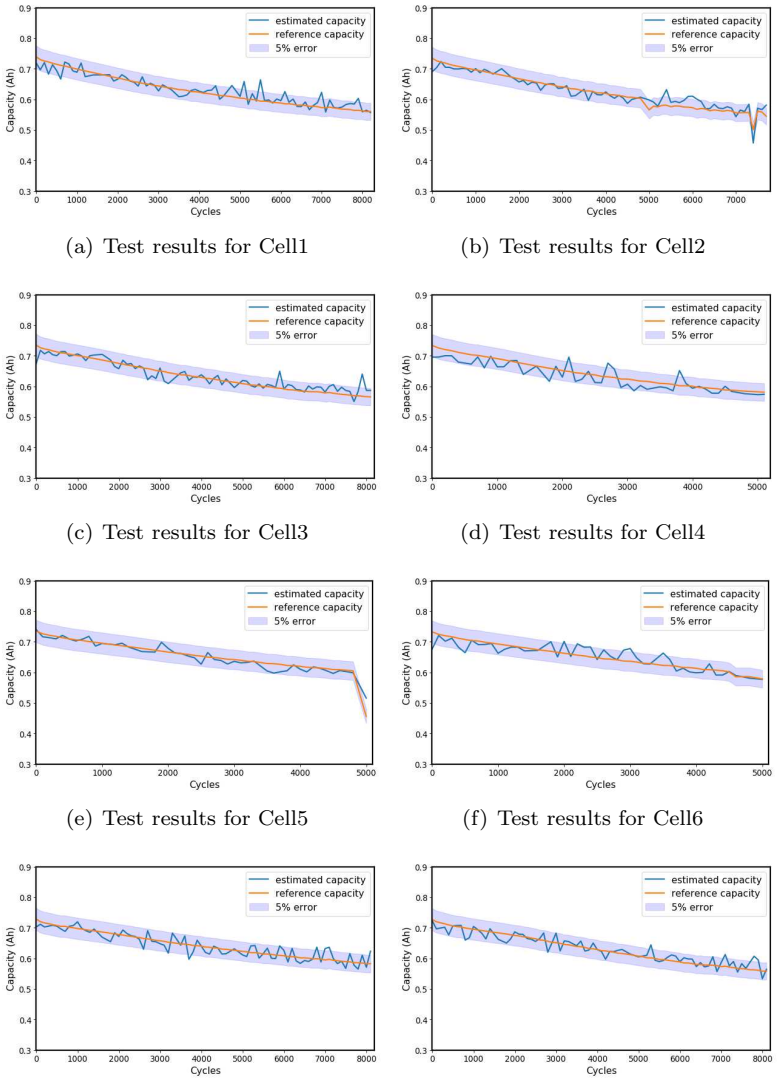


(a) Group 4 - Battery 13 (b) Group 4 - Battery 14

(c) Group 4 - Battery 15 (d) Group 4 - Battery 16

Figure 8: Capacity estimation results on Group 4 (Group 1, 2, 3 for training, and Group 4 for testing)

this case is the same as in case 1. The whole training and testing procedures are executed 100 times, the best estimation results on the testing dataset out of the 100 runs are shown in Figure 9, and the related RMSE, MaxE and MAE are summarized in Table 3. The RMSE is less than 0.0217 Ah, which is 2.93% of the rated capacity.



(a) Test results for Cell1

(b) Test results for Cell2

(c) Test results for Cell3

(d) Test results for Cell4

(e) Test results for Cell5

(f) Test results for Cell6

(g) Test results for Cell7

(h) Test results for Cell8

Figure 9: Capacity estimation results of Cell1 to Cell8

4.3. Analysis

To investigate the performance of the CNN model with different number of convolutional layers in both cases, the identical training and testing datasets are used for all tests. The training and testing procedures are executed 100 times for each CNN configuration, and the average RMSE, MaxE and MAE of 100 runs are summarized in Table 4. Further, Figure 10 shows RMSE bar charts for different CNN models. It is revealed that the CNN model with 2 convolutional

Table 3: Estimation errors in test batteries for Case 2

	RMSE (Ah)	MaxE (Ah)	MAE (Ah)
Cell1	0.0196	0.0698	0.0140
Cell2	0.0175	0.0511	0.0129
Cell3	0.0185	0.0727	0.0131
Cell4	0.0196	0.0505	0.0158
Cell5	0.0148	0.0598	0.0101
Cell6	0.0217	0.0568	0.0163
Cell7	0.0188	0.0478	0.0151
Cell8	0.0178	0.0446	0.0137

layers can achieve satisfactory results in Case 1, while 4 convolutional layers are required in Case 2. This is because Case 2 has less samples for training, therefore requires deeper architecture than Case 1 to extract more detailed information from limited training samples. Comparing the results of networks with 4, 5 and 6 convolutional layers, and considering the total number of parameters involved in each configuration (as shown in Table 4), the CNN with 4 convolutional layers is the best trade-off, which can achieve satisfactory estimation results with relatively fewer parameters.

Table 4: Comparison of estimation results with different number of convolutional layers

Number of Conv Layers		2	3	4	5	6
Case 1	RMSE (Ah)	0.0104	0.0107	0.0101	0.0105	0.0105
	MaxE (Ah)	0.0803	0.0873	0.0765	0.0847	0.0861
	MAE (Ah)	0.0070	0.0070	0.0065	0.0069	0.0071
Case 2	RMSE (Ah)	0.0385	0.0337	0.0210	0.0200	0.0198
	MaxE (Ah)	0.1077	0.1042	0.0651	0.0675	0.0645
	MAE (Ah)	0.0281	0.0237	0.0152	0.0143	0.0142
Total Parameters		16789	11653	12693	13733	14773

In this paper, the length of consecutive data points cut from the charging and discharging curves is

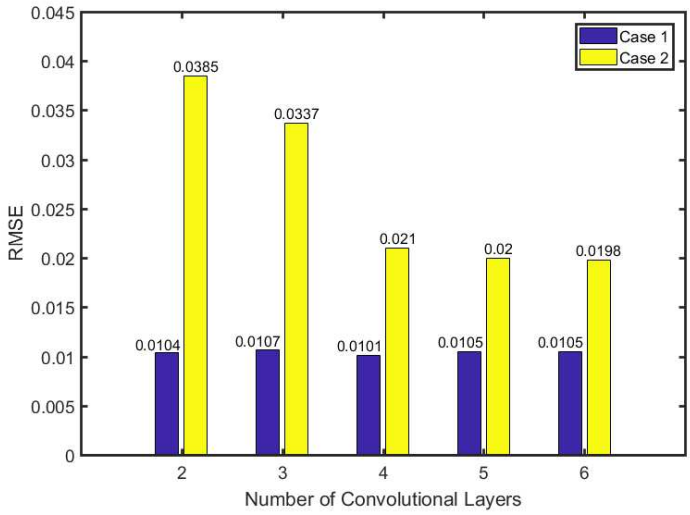


Figure 10: Estimation RMSE (Ah) on test datasets versus the number of convolutional layers

chosen to be 225 for each variable, and the three data chunks for current, voltage and temperature are fused to generate a $15 \times 15 \times 3$ input sample (image) to the CNN, referring to a partial charging/discharging segment with unknown initial and final SOC. This is again the best trade-off between the number of generated samples and the information embedded in each sample. To illustrate this, experiments are conducted with different lengths of segmentation, where the training and testing procedures are repeated 100 times on the identical training and testing datasets in Case 1. Figure 11 shows the average estimation error with respect to different length of segmentation. It is clear that 225 continuous data points provide relatively small estimation errors, while too small or too large segmentation both produce unsatisfactory estimation results. This is because shorter segments contain less useful information to describe the features, while longer segments will generate too few samples to train a proper model. In addition, the larger the segment size, the more parameters are involved, leading to longer training time.

Further, Table 5 compares the capacity estimation results of CNN, ANN and DNN (with different number of hidden layers and each layer has 40 neurons) using average RMSE, MaxE and MAE of 100 runs in Case 1, and the error ratio against the rated capacity are given in parenthesis. It is obvious that the CNN model has achieved the best results while involving much less parameters.

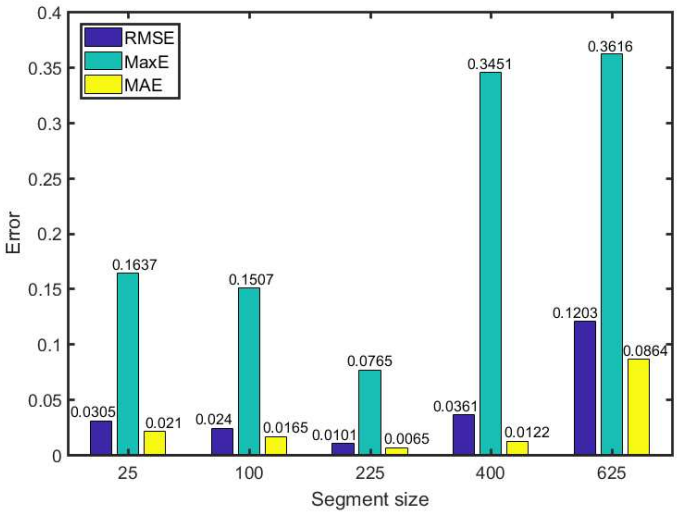


Figure 11: Estimation error on test datasets versus the length of segment

Table 5: Comparison of estimation results under CNN, ANN and DNN (() normalized by rated capacity)

	RMSE	MaxE	MAE	Parameters
CNN	0.0101 Ah (0.95%)	0.0765 Ah (7.54%)	0.0065 Ah (0.63%)	12693
ANN	0.0189 Ah (1.72%)	0.1568 Ah (14.25%)	0.0129 Ah (1.17%)	27081
2- layer DNN	0.0142 Ah (1.29%)	0.1930 Ah (17.55%)	0.0091 Ah (0.83%)	28721
3-layer DNN	0.0135 Ah (1.23%)	0.1433 Ah (13.03%)	0.0090 Ah (0.82%)	30361
4-layer DNN	0.0147 Ah (1.34%)	0.1476 Ah (13.42%)	0.0092 Ah (0.84%)	32001
5-layer DNN	0.0145 Ah (1.32%)	0.1544 Ah (14.04%)	0.0094 Ah (0.85%)	33641

In summary, the normalized RMSEs are less than 2.54% and 2.93% respectively on the two datasets, outperforming other machine-learning based estimation methods.

5. Conclusions

This paper has proposed a novel CNN-based battery capacity estimation method only using partial charging segment of the direct measurements (e.g. current, voltage, and cell surface temperature). Compared to other ML-based methods, the proposed method is easy to implement, and can achieve fast online capacity estimation without extra health features extraction or cumulative charge calculation processes, while only raw data of a partial charging process is required. The CNN has demonstrated the capability of handling a massive amount of data to learn representative features, and the feature extraction and capacity estimation are automatically executed in one framework. To apply CNN for capacity estimation using measurable variables, a transformation method is developed to convert the time series to image representations that are acceptable by CNNs, and the converted 3-D images embed the spatially and temporally correlated information among these variables. The data segmentation method performed priori to the transformation stage not only increases the sample numbers, but also makes it possible to achieve fast online capacity estimation only using partial charging segment of direct measurements with flexible starting point. The proposed method is evaluated on two battery degradation datasets, the estimation results confirm that the proposed CNN-based method can achieve satisfactory results and can be used for fast online capacity estimation once the model is properly trained offline. The CNN model developed in this paper has a large number of parameters to tune, and to reduce the size and number of tunable parameters in the CNN model will be our future work.

6. Acknowledgments

YH. LI would like to thank the China Scholarship Council for sponsoring her research. The work is partially funded by EPSRC under grant EP/R030243/1.

References

Birkl, C., 2017. Oxford battery degradation dataset 1. University of Oxford .

Borovykh, A., Bohte, S., Oosterlee, C.W., 2017. Conditional time series forecasting with convolutional neural networks. arXiv preprint arXiv:1703.04691 .

Cai, L., Meng, J., Stroe, D.I., Luo, G., Teodorescu, R., 2019. An evolutionary framework for lithium-ion battery state of health estimation. Journal of Power Sources 412, 615–622.

Chaoui, H., Ibe-Ekeocha, C.C., 2017. State of charge and state of health estimation for lithium batteries using recurrent neural networks. IEEE Transactions on vehicular technology 66, 8773–8783.

Couto, L.D., Schorsch, J., Job, N., Léonard, A., Kinnaert, M., 2019. State of health estimation for lithium ion batteries based on an equivalent-hydraulic model: An iron phosphate application. Journal of Energy Storage 21, 259–271.

Cui, Z., Chen, W., Chen, Y., 2016. Multi-scale convolutional neural networks for time series classification. arXiv preprint arXiv:1603.06995 .

Dai, H., Zhao, G., Lin, M., Wu, J., Zheng, G., 2018. A novel estimation method for the state of health of lithium-ion battery using prior knowledge-based neural network and markov chain. IEEE Transactions on Industrial Electronics 66, 7706–7716.

Eddahech, A., Briat, O., Bertrand, N., Deletage, J.Y., Vinassa, J.M., 2012. Behavior and state-of-health monitoring of li-ion batteries using impedance spectroscopy and recurrent neural networks. International Journal of Electrical Power & Energy Systems 42, 487–494.

Eleftheroglou, N., Mansouri, S.S., Loutas, T., Karvelis, P., Georgoulas, G., Nikolakopoulos, G., Zarouchas, D., 2019. Intelligent data-driven prognostic methodologies for the real-time remaining useful life until the end-of-discharge estimation of the lithium-polymer batteries of unmanned aerial vehicles with uncertainty quantification. Applied Energy 254, 113677.

Garg, A., Peng, X., Le, M.L.P., Pareek, K., Chin, C., 2018. Design and analysis of capacity models for lithium-ion battery. Measurement 120, 114–120.

Guo, P., Cheng, Z., Yang, L., 2019. A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction. Journal of Power Sources 412, 442–450.

Hu, C., Jain, G., Schmidt, C., Strief, C., Sullivan, M., 2015. Online estimation of lithium-ion battery capacity using sparse bayesian learning. Journal of Power Sources 289, 105–113.

Kingma, D.P., Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 .

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.

Li, F.F., Karpathy, A., Johnson, J., 2015. Cs231n: Convolutional neural networks for visual recognition. University Lecture .

Li, X., Yuan, C., Li, X., Wang, Z., 2019a. State of health estimation for li-ion battery using incremental capacity analysis and gaussian process regression. Energy , 116467.

Li, Y., Abdel-Monem, M., Gopalakrishnan, R., Berecibar, M., Nanini-Maury, E., Omar, N., van den Bossche, P., Van Mierlo, J., 2018. A quick on-line state of health estimation method for li-ion battery with incremental capacity curves processed by gaussian filter. Journal of Power Sources 373, 40–53.

Li, Y., Liu, K., Foley, A.M., Zülke, A., Berecibar, M., Nanini-Maury, E., Van Mierlo, J., Hoster, H.E., 2019b. Data-driven health estimation and lifetime prediction of lithium-ion batteries: A review. Renewable and Sustainable Energy Reviews 113, 109254.

Liu, D., Song, Y., Li, L., Liao, H., Peng, Y., 2018. On-line life cycle health assessment for lithium-ion battery in electric vehicles. Journal of cleaner production 199, 1050–1065.

Liu, K., Ashwin, T., Hu, X., Lucu, M., Widanage, W.D., 2020a. An evaluation study of different modelling techniques for calendar ageing prediction of lithium-ion batteries. Renewable and Sustainable Energy Reviews 131, 110017.

Liu, K., Li, K., Peng, Q., Zhang, C., 2019a. A brief review on key technologies in the battery management system of electric vehicles. Frontiers of mechanical engineering 14, 47–64.

Liu, K., Li, Y., Hu, X., Zhang, C., 2019b. Gaussian process regression with automatic relevance determination kernel for calendar aging prediction of lithium-ion batteries. IEEE Transactions on Industrial Informatics PP, 1–1. doi:10.1109/TII.2019.2941747.

Liu, K., Shang, Y., Ouyang, Q., Widanage, W., 2020b. A data-driven approach with uncertainty quantification for predicting future capacities and remaining useful life of lithium-ion battery. IEEE Transactions on Industrial Electronics .

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. Neurocomputing 234, 11–26.

Ouyang, M., Feng, X., Han, X., Lu, L., Li, Z., He, X., 2016. A dynamic capacity degradation model and its applications considering varying load for a large format li-ion battery. Applied Energy 165, 48–59.

Reed, R., MarksII, R.J., 1999. Neural smithing: supervised learning in feedforward artificial neural networks. Mit Press.

Ruder, S., 2016. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 .

Severson, K.A., Attia, P.M., Jin, N., Perkins, N., Jiang, B., Yang, Z., Chen, M.H., Aykol, M., Herring, P.K., Fraggedakis, D., et al., 2019. Data-driven prediction of battery cycle life before capacity degradation. Nature Energy 4, 383.

Shen, S., Sadoughi, M., Chen, X., Hong, M., Hu, C., 2019. A deep learning method for online capacity estimation of lithium-ion batteries. Journal of Energy Storage 25, 100817.

Tang, X., Liu, K., Wang, X., Liu, B., Gao, F., Widanage, W.D., 2019. Real-time aging trajectory prediction using a base model-oriented gradient-correction particle filter for lithium-ion batteries. Journal of Power Sources 440, 227118.

Tang, X., Zou, C., Yao, K., Chen, G., Liu, B., He, Z., Gao, F., 2018. A fast estimation algorithm for lithium-ion battery state of health. Journal of Power Sources 396, 453–458.

Weng, C., Cui, Y., Sun, J., Peng, H., 2013. On-board state of health monitoring of lithium-ion batteries using incremental capacity analysis with support vector regression. Journal of Power Sources 235, 36–44.

Weng, C., Feng, X., Sun, J., Peng, H., 2016. State-of-health monitoring of lithium-ion battery modules and packs via incremental capacity peak tracking. Applied Energy 180, 360–368.

Xiong, R., Li, L., Tian, J., 2018. Towards a smarter battery management system: A critical review on battery state of health monitoring methods. Journal of Power Sources 405, 18–29.

Xu, B., Oudalov, A., Ulbig, A., Andersson, G., Kirschen, D.S., 2016. Modeling of lithium-ion battery degradation for cell life assessment. IEEE Transactions on Smart Grid 9, 1131–1140.

Yang, J., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S., 2015. Deep convolutional neural networks on multichannel time series for human activity recognition, in: Twenty-Fourth International Joint Conference on Artificial Intelligence.

You, G.w., Park, S., Oh, D., 2016. Real-time state-of-health estimation for electric vehicle batteries: A data-driven approach. Applied energy 176, 92–103.

Yu, Q., Xiong, R., Yang, R., Pecht, M.G., 2019. Online capacity estimation for lithium-ion batteries through joint estimation method. Applied Energy 255, 113817.

Zhang, C., Li, K., Deng, J., Song, S., 2016. Improved realtime state-of-charge estimation of $LiFePO_4$ battery based on a novel thermoelectric model. IEEE Transactions on Industrial Electronics 64, 654–663.

Zhang, C., Li, K., Mcloone, S., Yang, Z., 2014. Battery modelling methods for electric vehicles-a review, in: 2014 European Control Conference (ECC), IEEE. pp. 2673–2678.

Zhang, S., Zhai, B., Guo, X., Wang, K., Peng, N., Zhang, X., 2019. Synchronous estimation of state of health and remaining useful lifetime for lithium-ion battery using the incremental capacity and artificial neural networks. Journal of Energy Storage 26, 100951.

Zheng, L., Zhang, L., Zhu, J., Wang, G., Jiang, J., 2016. Co-estimation of state-of-charge, capacity and resistance for lithium-ion batteries based on a high-fidelity electrochemical model. Applied Energy 180, 424–434.

Zheng, L., Zhu, J., Lu, D.D.C., Wang, G., He, T., 2018. Incremental capacity analysis and differential voltage analysis based state of charge and capacity estimation for lithium-ion batteries. Energy 150, 759–769.