Journal of Energy Chemistry 76 (2023) 404–413

Contents lists available at ScienceDirect

Journal of Energy Chemistry

journal homepage: www.elsevier.com/locate/jechem

http://www.journals.elsevier.com/journal-of-energy-chemistry/

Semi-supervised estimation of capacity degradation for lithium ion batteries with electrochemical impedance spectroscopy

Rui Xiong [a], Jinpeng Tian [a,*], Weixiang Shen [b], Jiahuan Lu [a], Fengchun Sun [a]

[a] *Department of Vehicle Engineering, School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China*
[b] *School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Hawthorn, Victoria 3122, Australia*

ARTICLE INFO

Article history:
Received 22 August 2022
Revised 26 September 2022
Accepted 27 September 2022
Available online 5 October 2022

Keywords:
Lithium-ion battery
Capacity degradation
Electrochemical impedance spectroscopy
Deep learning

ABSTRACT

Machine learning-based methods have emerged as a promising solution to accurate battery capacity estimation for battery management systems. However, they are generally developed in a supervised manner which requires a considerable number of input features and corresponding capacities, leading to prohibitive costs and efforts for data collection. In response to this issue, this study proposes a convolutional neural network (CNN) based method to perform end-to-end capacity estimation by taking only raw impedance spectra as input. More importantly, an input reconstruction module is devised to effectively exploit impedance spectra without corresponding capacities in the training process, thereby significantly alleviating the cost of collecting training data. Two large battery degradation datasets encompassing over 4700 impedance spectra are developed to validate the proposed method. The results show that accurate capacity estimation can be achieved when substantial training samples with measured capacities are given. However, the estimation performance of supervised machine learning algorithms sharply deteriorates when fewer samples with measured capacities are available. In this case, the proposed method outperforms supervised benchmarks and can reduce the root mean square error by up to 50.66%. A further validation under different current rates and states of charge confirms the effectiveness of the proposed method. Our method provides a flexible approach to take advantage of unlabelled samples for developing data-driven models and is promising to be generalised to other battery management tasks.

1. Introduction

1.1. Motivations and literature review

Lithium ion batteries as electrochemical energy storage devices have become a key driver to decarbonise the energy sector [1]. Owing to internal side reactions, lithium ion batteries undergo gradual capacity degradation and power fade [2,3]. Therefore, monitoring the state of health (SOH) of the batteries is a critical task for battery management systems [4].

Battery SOH is generally defined as the ratio of the present capacity to the initial capacity. As capacity measurements are rarely applicable in real-world applications, capacity estimation at unknown ageing levels has attracted wide attention [5,6]. In general, capacity estimation methods can be divided into two groups, namely model-based methods and data-driven methods. Model-based methods incorporate battery capacity into voltage models, and then the capacity can be estimated by minimising the errors between simulated and measured voltages [7]. However, the parameterisation of battery models requires cumbersome characterisation tests and modelling errors negatively affect capacity estimation accuracy.

Recently, machine learning-based methods have been explored for capacity estimation. These methods first determine features highly correlated with capacity degradation and then develop the mapping between features and battery capacity. For example, eight features were derived from the relaxation voltage curves obtained after the batteries are fully charged [8]. Three features were selected to evaluate battery capacity estimation using a set of machine learning algorithms. In [9], current pulses were imposed on batteries at a specific state of charge (SOC) and the corresponding voltage response was recorded to compute the sample entropy as a feature. A sparse Bayesian predictive modelling algorithm was trained to establish the mapping between the sample entropy and battery capacity. This method was then refined in [10] by combining multiple pulse-derived features and Gaussian process regres-

* Corresponding author.
E-mail address: tianjinpeng@bit.edu.cn (J. Tian).

https://doi.org/10.1016/j.jechem.2022.09.045
2095-4956/

sion (GPR). Other methods resort to the constant-current charging process to identify features, such as the height [11], position [12] and area [13] of incremental capacity (IC) curves. As the IC transform amplifies the sampling noise, researchers attempted to adopt partial charging data for capacity estimation, as demonstrated by [14–16].

The above methods generally rely on the time-domain battery voltage and current data to conduct battery degradation estimation. On the other hand, electrochemical impedance spectroscopy (EIS) is a versatile tool that can reflect fruitful kinetic information of a battery [17,18]. Compared with the above-discussed methods, EIS is a reliable electrochemical technique that has various advantages. First, EIS adopts a small-amplitude alternating current (AC) to stimulate the tested battery and collect the voltage response. The testing procedure does not require charging/discharging batteries, thereby saving energy and testing efforts [19]. Recent studies have demonstrated the quick acquisition of impedance spectra using easily available hardware [20], paving the way to utilise EIS for battery management purposes. Besides, EIS allows us to probe battery characteristics in a non-invasive approach [21]. For example, EIS is effective to characterise power fade [19], decouple electrochemical processes [22] and diagnose lithium plating [23]. It is also a promising candidate for capacity estimation. Messing et al. [24] fitted impedance spectra to a fractional order model to obtain resistances reflecting solid electrolyte interphase (SEI) and charge transfer for capacity estimation. Carthy et al. [25] found a linear relationship between the real part of the impedance at 1 kHz with capacity fade. Recently, the combination of EIS and machine learning has gained increasing attention. Fu et al. [26] extracted six features from impedance spectra and fed them into an extreme learning machine for capacity estimation. Zhang et al. [27] proposed to directly feed impedance spectra into GPR to estimate battery capacity and predict remaining useful life. Their results show that EIS based estimation methods outperform methods based on discharge profiles. However, Kim et al. [28] revealed that GPR has limited feature learning capability, and a generative adversarial network was proposed to extract ageing-related features from impedance spectra before feeding features into GPR for capacity estimation.

1.2. Gap analysis and article contributions

Despite the advances in the field of battery capacity estimation, existing capacity estimation studies generally rely on supervised training, which requires a considerable number of features and corresponding capacities to train machine learning models. Although ageing-related features are available relatively easily, collecting their corresponding capacities through measurements is a costly task because it requires fully charging and discharging batteries at various ageing levels. In reality, we can easily obtain a considerable number of features at different ageing levels but many of them do not have corresponding capacity measurements. For instance, field data provide a large spectrum of battery operating data, but rarely have corresponding capacity measurements as batteries in service are seldom fully charged or discharged [29]. Accordingly, it is desirable to develop machine learning-based capacity estimation methods that can effectively make use of not only labelled data but also unlabelled data to achieve competitive estimation results.

To this end, we propose a semi-supervised capacity estimation method that can take advantage of labelled and unlabelled data to estimate battery capacity. Considering the potential of EIS in reflecting battery degradation, we take EIS as an example to demonstrate accurate capacity estimation. This study makes the following contributions.

(1) An end-to-end capacity estimation method is proposed to circumvent feature extraction. By resorting to advanced deep convolutional neural networks (CNNs) to integrate feature extraction and capacity estimation, we show that battery capacity can be accurately estimated using raw impedance spectra as input, leading to flexible and general capacity estimation.

(2) A semi-supervised training approach is proposed to take advantage of unlabelled impedance spectra to improve the performance of battery capacity estimation. Thanks to the proposed model architecture and training strategy, the proposed model significantly improves estimation accuracy when the scarce labelled training data are augmented by unlabelled impedance spectra.

(3) Two large battery degradation datasets comprising over 4700 impedance spectra collected from 16 batteries are developed to systematically validate the proposed method under various conditions.

1.3. Article organisation

The rest of the article is organised as follows. ‘‘*Method*” section delineates the proposed method and ‘‘*Battery degradation tests*” section introduces battery degradation tests. Validations and discussions of the proposed method are provided in ‘‘*Results and discussions*” section. ‘‘*Conclusions*” section concludes the present study.

2. Method

2.1. Semi-supervised capacity estimation problem

Data-driven battery capacity estimation seeks to develop a model using the easily available data as input to estimate battery capacity at unknown ageing levels, which can be expressed as.

$$\hat{y} = f(x) \tag{1}$$

where y denotes the actual battery capacity, and $\hat{y}$ denotes the corresponding estimation result. x represents the input data, such as the impedance spectrum adopted in this work. $f()$ denotes the nonlinear mapping which can be described by a machine learning algorithm. In the context of data-driven capacity estimation, it is determined based on a labelled dataset consisting of both input and corresponding capacity measurements.

$$D^{(L)} = \left\{ \left(x_i^{(L)} \quad y_i^{(L)} \right) \right\}_{i=1}^{n^{(L)}} \tag{2}$$

where $x_i^{(\mathrm{L})}$ and $y_i^{(\mathrm{L})}$ are the input and the measured capacity of the ith sample in the labelled dataset, respectively. $n^{(\mathrm{L})}$ denotes the total number of samples. In general, the model training requires a large $D^{(\mathrm{L})}$, where a substantial amount of labelled samples are available. However, although $D^{(\mathrm{L})}$ can be developed offline, capacity measurement requires fully charging and discharging batteries, leading to high costs in terms of energy and effort. In contrast, we can easily acquire a considerable amount of input data x, resulting in an unlabeled dataset.

$$D^{(U)} = \left\{ x_i^{(U)} \right\}_{i=1}^{n^{(U)}} \tag{3}$$

where $D^{(\mathrm{U})}$ denotes the unlabeled dataset which consists of $n^{(\mathrm{U})}$ input samples $x_i^{(\mathrm{U})}$ without corresponding capacity measurements. In this regard, $D^{(\mathrm{U})}$ cannot be exploited by conventional supervised capacity estimation algorithms [30]. Given this, we expect to improve capacity estimation performance by making full use of not only $D^{(L)}$ but also $D^{(U)}$. This is fulfilled by developing a joint-loss CNN (JL-CNN) described in "*Structure of the Joint-loss CNN*" section.

2.2. Structure of the Joint-loss CNN

As shown in Fig. 1, the proposed JL-CNN can carry out an end-to-end capacity estimation by taking raw impedance spectra as input. A key step of capacity estimation is to extract meaningful features from raw data. While conventional machine learning methods rely on manual feature extraction, the proposed method takes advantage of CNN architecture [31] to enable automatic feature extraction. Besides, both labelled and unlabelled impedance spectra can participate in the model training. The JL-CNN is designed with three modules, namely feature extraction, capacity estimation and input reconstruction. The feature extraction module is responsible for automatically encoding the input impedance spectra into meaningful features, which can be either labelled or unlabeled. The extracted features are then fed into the capacity estimation module to generate the capacity estimation results. In this case, only features extracted from the impedance spectra in the $D^{(L)}$ are adopted to train this module. On the other hand, the input reconstruction module is developed to use the extracted features to reconstruct the original input impedance spectra. It is expected to help the feature extraction module to learn meaningful features from impedance spectra. This module does not require its input to be labelled, thereby unlabeled samples can be efficiently exploited. The working principles of the three modules are introduced below.

(1) **Feature extraction module:** As shown in Fig. 1, the feature extraction module begins with three convolutional blocks, each of which is composed of a convolutional layer and a max-pooling layer in order. Two dense layers are concatenated after the convolutional blocks. Instead of manual feature extraction, the convolutional layers utilise a set of kernels to convolve with input data to automatically extract local features. Meaningful representations can be effectively extracted by stacking multiple convolutional blocks. As one of the most important structures of deep neural networks, the convolutional layers have been proven to be successful in various fields [31]. The details of different layers are provided in the Appendix. As the impedance spectra are composed of impedance measured at a series of frequencies, we adopt 1D convolutional layers to extract features and form a feature vector

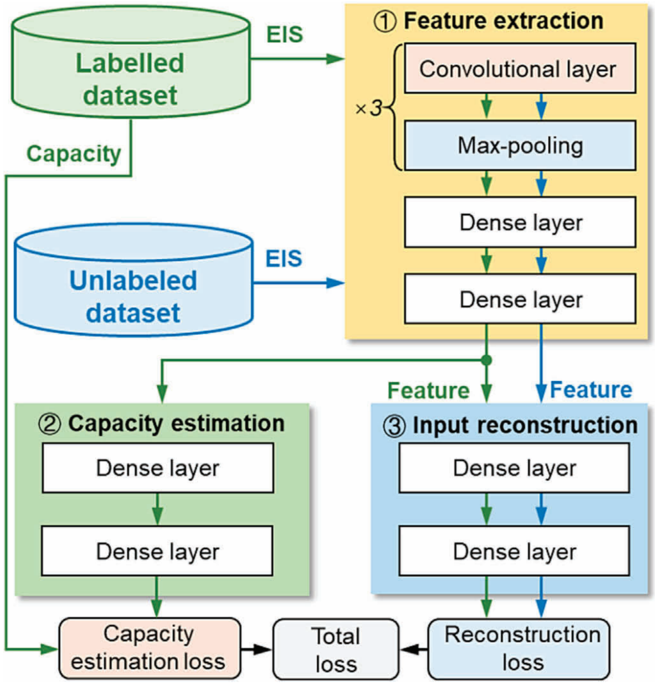

Fig. 1. The structure and training strategy of the proposed joint-loss convolutional neural network for semi-supervised capacity estimation. Arrows of different colours represent the flow of samples from different datasets.

$$h_E = f_{FE}\left(\begin{bmatrix} \boldsymbol{Z}'^{T} & \boldsymbol{Z}''^{T} \end{bmatrix}^{T}\right) \tag{4}$$

where f_{FE} represents the nonlinear transform performed by the feature extraction module. h_E is the $N_E \times 1$ feature vector. $\boldsymbol{Z}\prime$ and $\boldsymbol{Z}\prime\prime$ are the sequences of the real and imaginary impedance, respectively. They are defined as a function of frequency

$$\begin{cases} \boldsymbol{Z}' = \begin{bmatrix} Z'_{f_1} & Z'_{f_2} & \dots & Z'_{f_N} \end{bmatrix} \\ \boldsymbol{Z}'' = \begin{bmatrix} Z''_{f_1} & Z''_{f_2} & \dots & Z''_{f_N} \end{bmatrix} \end{cases} \tag{5}$$

where $Z\prime_{f_i}$ and $Z\prime\prime_{f_i}$ $(1 \le i \le N)$ denote the real and imaginary parts of the impedance measured at the frequency f_i, respectively. A total of N frequencies is covered in the impedance spectra. The rectified linear unit (ReLU) activation is applied to increase the nonlinearity.

(2) **Capacity estimation module:** The extracted feature vector h_E serves as the input of the capacity estimation and input reconstruction modules for different purposes. Specifically, the capacity estimation module is concatenated after the feature extraction module to output the capacity estimation results. This can be written as.

$$\hat{y} = f_{CE}(h_E) \tag{6}$$

where f_{CE} represents the function of the capacity estimation module. It is implemented by stacking two dense layers. It should be noted that the combination of feature extraction and capacity estimation results in typical supervised CNNs, whose performance has been widely demonstrated by existing studies [15,16,32].

(3) **Input reconstruction module:** The input reconstruction module distinguishes the JL-CNN from existing supervised CNNs by taking advantage of unlabeled samples for training. This module also accepts the feature vector h_E as input, aiming to reversely reconstruct the input impedance sequences using two dense layers. The reconstructed real and imaginary parts of an impedance spectrum are computed as.

$$\begin{bmatrix} \hat{\boldsymbol{Z}}\prime & \hat{\boldsymbol{Z}}\prime\prime \end{bmatrix} = f_{IR}(h_E) \tag{7}$$

where f_{IR} represents the function of the input reconstruction module. $\hat{\boldsymbol{Z}}\prime$ and $\hat{\boldsymbol{Z}}''$ represent the reconstruction results of the real and imaginary impedance sequences, respectively.

The input reconstruction module can lead to two benefits. First, as the reconstruction is conducted in an unsupervised manner, both the samples from the labelled and unlabeled datasets are involved. Hence, it can force the feature extraction module to learn meaningful representations from more impedance spectra. Second, the reconstruction also helps alleviate overfitting, as it drives the feature extraction module to preserve meaningful information regarding impedance spectra when learning the capacity estimation task. To preclude the possibility that the feature extractor directly copies the input impedance spectra, we set the dimension of the feature vector h_E smaller than the length of the input impedance sequence, i.e., $N_E \langle N$.

The detailed structure and hyperparameters of the JL-CNN developed in this study are listed in Table 1. Here, Conv (a, b, c) represents a 1D convolutional layer that has a kernel size of a, the number of filters of b and a stride of c. ReLU represents the ReLU activation. Max-pooling (a) represents a max-pooling layer with a window size of a. Dense (a) represents a dense layer with a neurons. Working principles of these kinds of layers can be found in Appendix.

2.3. Model training

The developed JL-CNN is parameterised through data-driven training. First, the real and imaginary parts of impedance spectra are standardised to accelerate training, which is expressed as.

$$\begin{cases} Z'(n) = \frac{1}{\sigma'}\left(Z' - \mu'\right) \\ Z''(n) = \frac{1}{\sigma''}\left(Z'' - \mu''\right) \end{cases} \tag{8}$$

where $Z'^{(n)}$ and $Z''^{(n)}$ represent the standardised real and imaginary parts, respectively. μ and σ are the mean and standard deviation computed from the impedance data in the training dataset, respectively. Afterwards, two optimisation objectives are designed:

(1) To fulfil the capacity estimation task, the feature extraction and capacity estimation modules should be trained to output battery capacity with a given impedance spectrum. Therefore, the developed JL-CNN should minimise the capacity estimation loss on the labelled dataset, which is defined as.

$$J_{\mathrm{CE}} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i^{(\mathrm{L})} - \hat{y}_i^{(\mathrm{L})}\right)^2 \tag{9}$$

where m is a batch size of labelled samples.

(2) The feature extraction and input reconstruction modules are trained by minimising the EIS reconstruction loss on both labelled and unlabeled datasets, which are defined as.

$$\begin{aligned} J_{\mathrm{IR}} = & \tfrac{1}{m}\sum_{i=1}^{m}\left(\left[\widehat{\boldsymbol{Z}}'^{(n,\mathrm{L})} \quad \widehat{\boldsymbol{Z}}''^{(n,\mathrm{L})}\right] - \left[\boldsymbol{Z}'^{(n,\mathrm{L})} \quad \boldsymbol{Z}''^{(n,\mathrm{L})}\right]\right)^2 \\ & + \tfrac{1}{n}\sum_{i=1}^{n}\left(\left[\widehat{\boldsymbol{Z}}'^{(n,\mathrm{U})} \quad \widehat{\boldsymbol{Z}}''^{(n,\mathrm{U})}\right] - \left[\boldsymbol{Z}'^{(n,\mathrm{U})} \quad \boldsymbol{Z}''^{(n,\mathrm{U})}\right]\right)^2 \end{aligned} \tag{10}$$

where n is the batch size of unlabeled samples. Let θ_{FE}, θ_{CE} and θ_{IR} represent the parameter sets of the feature extraction, capacity estimation and input reconstruction modules, the overall training loss can be expressed as the weighted sum of J_{CE} and J_{IR}.

$$J = J_{\mathrm{CE}}(\theta_{\mathrm{FE}}, \theta_{\mathrm{CE}}) + \lambda J_{\mathrm{IR}}(\theta_{\mathrm{FE}}, \theta_{\mathrm{IR}}) \tag{11}$$

where λ is a weight balancing the contributions of capacity estimation and input reconstruction tasks to the overall loss. The model parameterisation problem can be written as.

$$[\theta_{\mathrm{FE}}, \theta_{\mathrm{CE}}, \theta_{\mathrm{IR}}] = \arg\min J \tag{12}$$

Table 1
Structure and hyperparameters of the JL-CNN model.

Module	Layer	Input	Output
Feature extraction	[Conv (3,64,1) ReLU Max-pooling (2)]×3 Dense (128) ReLU Dense (16)	Impedance spectrum $\left[\boldsymbol{Z}'^{\mathrm{T}}, \boldsymbol{Z}''^{\mathrm{T}}\right]^{\mathrm{T}}$	Feature vector h_{E}
Capacity estimation	Dense (16) ReLU Dense (1)	Feature vector h_{E}	Capacity estimate $\widehat{y}$
Input reconstruction	Dense (16) ReLU Dense (102)	Feature vector h_{E}	Reconstructed impedance spectrum $\left[\hat{\boldsymbol{Z}}', \hat{\boldsymbol{Z}}''\right]$

After defining the training loss, we propose the overall model training procedure in Table 2. In general, a grid search algorithm is adopted to determine the weight λ. Then, the other parameters are determined through gradient descent. It is assumed that λ belongs to $D_\lambda = \left[0, 10^{-5}, 10^{-4}, \cdots, 10^{3}\right]$. Given a specific λ, the training loss is computed based on Eq. (11), and the Adam algorithm [33] is adopted to minimise the training loss. In addition, a few labelled training samples in the training dataset are left out as the validation set, which does not participate in the loss computation but is used to monitor capacity estimation performance at each epoch. Different λ values are adopted for training, and the model with the lowest validation loss is selected as the eventual one. In the Adam algorithm, we set the maximum number of training epochs and the batch size to be 1000 and 32, respectively. All CNNs are developed and evaluated based on the PyTorch package (version 1.8.1) in Python 3.8.10. A desktop equipped with an NVIDIA GeForce GTX 1660 SUPER graphical processing unit is used as the training hardware.

3. Battery degradation tests

3.1. Collection of battery impedance spectra over battery life

We develop two battery capacity degradation datasets incorporating cyclic degradation and regular EIS measurements to validate the proposed method. Sixteen 2.4 Ah cylindrical batteries are adopted and their upper and lower voltage limits are 4.2 and 3 V, respectively. The anode and cathode materials are graphite and a mix of lithium cobalt oxide and lithium nickel manganese cobalt oxide, respectively. These batteries are cycled using Gamry Interface 5000P electrochemical workstations at 25 °C in a thermal chamber. The two datasets are named 1-C and 2-C datasets according to the applied current rates and their testing procedures are briefly introduced below.

1-C dataset: Eight batteries are cycled at 1C (2.4 A for the studied batteries) in a constant-current (CC) mode. As shown in Fig. 2(a), a complete cycle includes 6 steps, i.e., (I) CC charge to the upper voltage limit, (II) 15-minute rest, (III) EIS measurement at the fully-charged state, (IV) CC discharge to the lower voltage limit, (V) 15-min rest and (VI) EIS measurement at the fully-discharged state. The EIS tests in steps (III) and (VI) are performed in a galvanostatic mode over the frequency range of 0.1 Hz to 10 kHz with 10 points per decade. Each impedance spectrum comprises impedance sequences at 51 frequency points, covering most electrochemical processes of interest to battery degradation studies [32]. A current amplitude of 1/24C is chosen to ensure a stable battery state. This group of batteries have been cycled 200 times. Some cycles are removed owing to the incomplete charging process caused by the power outage and eventually 3136 impedance spectra are obtained.

2-C dataset: The other eight batteries are tested using the same testing procedures but a 2C current rate is adopted to explore battery capacity degradation under fast charging conditions. In this case, a total of 1584 impedance spectra are collected over 100 degradation cycles.

In total, the two datasets incorporate 4720 impedance spectra at different degradation levels, and their capacity degradation trajectories are visualised in Fig. 2(b), indicating that the batteries experience a severe capacity drop. The capacities of batteries in 1-C and 2-C datasets degrade to 0.86 and 0.25 Ah, respectively,

Table 2
Training procedures of the joint-loss convolutional neural network.

```
Input: Labelled training dataset $D^{(L)}$ and unlabelled training dataset $D^{(U)}$
Standardise all impedance spectra according to Eq. (8).
For $\lambda$ in $D_\lambda$:
    Randomly divide $D^{(L)}$ into training set $D^{(L,t)}$ and validation set $D^{(L,v)}$.
    Initialise the parameter sets $[\theta_{FE}, \theta_{CE}, \theta_{IR}]$.
    Initialise the validation loss $J_v$=+∞.
    while epoch ≤ threshold:
        Divide $D^{(L,t)}$ and $D^{(U)}$ into $N_b$ batches, respectively.
        while $j \le N_b$:
            Estimate capacity $\hat{y}$ using the feature extraction and capacity estimation modules (Eqs. (4) and (6)).
            Reconstruct the input impedance sequence using feature extraction and input reconstruction modules (Eqs. (4) and (7)).
            Compute the training loss of the $j$th batch (Eq. (9) to Eq. (11)).
            Update $[\theta_{FE}, \theta_{CE}, \theta_{IR}]$ using the Adam algorithm.
            $j$=$j$+1
        Compute $J_v$ of this epoch on $D^{(L,v)}$ (Eq. (9)).
        if $J_v$ decreases:
            Update $J_v$
            Save $[\theta_{FE}, \theta_{CE}, \theta_{IR}]$ and its corresponding $J_v$.
        Save $[\theta_{FE}, \theta_{CE}, \theta_{IR}]$ and its corresponding $J_v$.
        epoch=epoch+1
Output: $[\theta_{FE}, \theta_{CE}, \theta_{IR}]$ with the lowest $J_v$.
```

such that battery degradation levels can cover most applications. Experimental results show that the capacity fade is accompanied by an increase in impedance spectra, as demonstrated in Fig. 2(c and d). Interestingly, further comparison between the impedance spectra in Fig. 2(c and d) reveals different impedance increase patterns. Although impedance spectra in both datasets move to the right owing to the increase in ohmic resistance, the 2-C ones show an evident increase in the middle-frequency arcs, which can be ascribed to sluggish charge transfer reactions and SEI growth [34]. Such inconsistent degradation patterns imply that manually extracted features to represent degradation may vary for batteries that undergo different degradation tests, thereby further posing challenges to the following capacity estimation. In contrast, the proposed method adopts an end-to-end deep learning framework that allows automatic feature extraction and capacity estimation, and can better adapt to different degradation conditions.

3.2. Division of training and testing samples

Given a battery degradation dataset (1-C or 2-C datasets), we randomly select 20% of impedance spectra and their corresponding capacities to form a test dataset to examine the proposed method. The remaining impedance spectra and capacities are used as the training samples, and they are further divided into the labelled training dataset, unlabelled training dataset and validation set. First, 10% of all training samples are randomly picked up as the validation set, and then a label rate (LR) is defined to simulate the conditions where different numbers of labelled samples are available.

$$\mathrm{LR} = \frac{n^{(\mathrm{L})}}{n^{(\mathrm{A})}} \times 100\% \tag{13}$$

where $n^{(\mathrm{A})}$ denotes the number of all training samples, and $n^{(\mathrm{L})}$ denotes the number of samples in the labelled training dataset. Therefore, an LR of 90% means all training samples are labelled. In this case, the JL-CNN degrades to a supervised CNN. On the other hand, a smaller LR indicates that both labelled and unlabelled datasets exist. This poses challenges to supervised CNNs as fewer training samples are available, and we expect the JL-CNN can exploit both types of samples to improve battery capacity estimation performance.

4. Results and discussion

4.1. Capacity estimation results

Without loss of generality, this section first adopts fully-charged impedance spectra collected in the 1-C dataset for

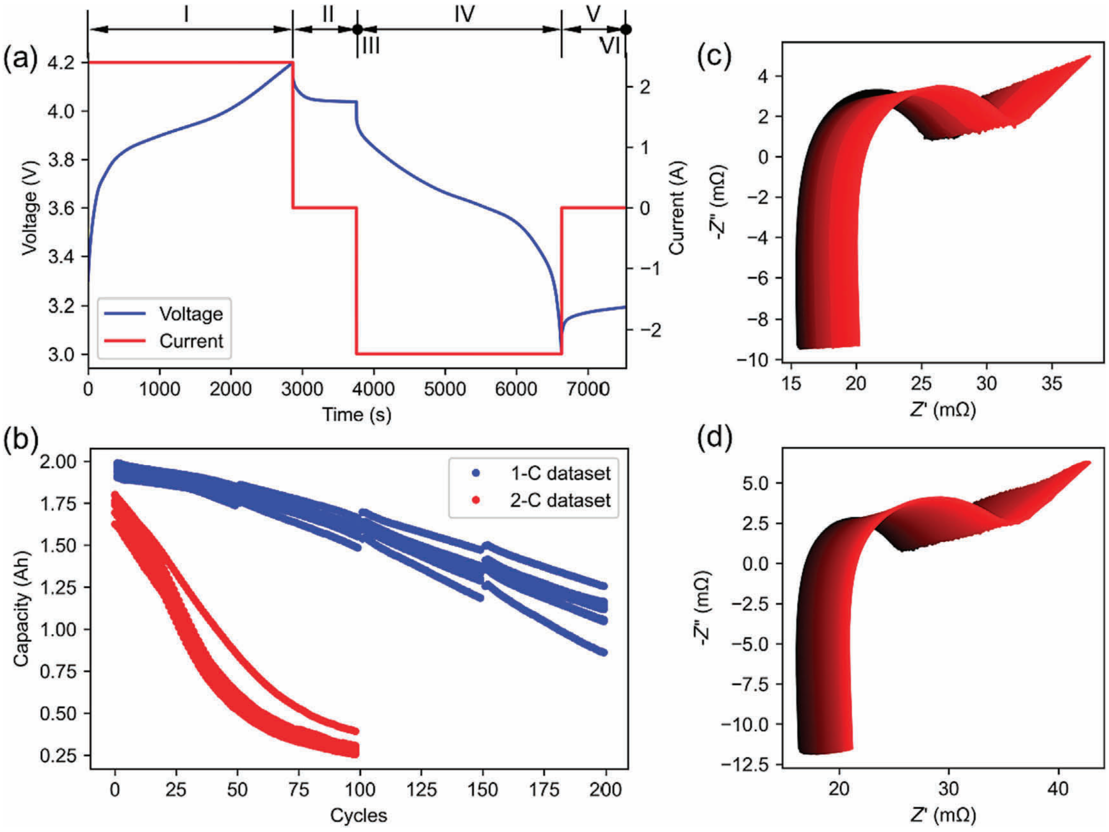

Fig. 2. (a) An illustration of one cycle and the six testing steps. (b) Capacity degradation trajectories of the 1-C and 2-C datasets. Examples showing the evolution of fully-charged impedance spectra in (c) 1-C and (d) 2-C datasets during battery degradation. In (c) and (d), the colour changes from black to red as the battery degrades.

validation. Note that in this study, we intentionally use unprocessed impedance spectra as the input for capacity estimation and let the deep learning models automatically extract features and conduct regression. As an alternative, prior knowledge may be utilised to pre-process the impedance spectra. For example, impedance data with positive imaginary parts can be removed as they do not provide information regarding battery characteristics [35]. We examine two capacity estimation cases with ample and scarce training samples, respectively, to intuitively demonstrate the impact of unlabelled samples on capacity estimation:

Case 1.

Samples in the training dataset are all labelled (LR = 90%), which is an ideal case to develop capacity estimation models. A CNN is trained in a supervised manner. In this case, the CNN only performs capacity estimation using feature extraction and capacity estimation modules.

Case 2.

A low LR of 10% is chosen to simulate the condition where only scarce labelled samples are available. In this case, a supervised CNN is developed based on labelled training samples while the proposed JL-CNN is developed to take advantage of both labelled and unlabelled samples.

In each case, the CNN adopts the same hyperparameter settings except that the JL-CNN has an additional input reconstruction module. A root mean square error (RMSE) is computed to quantify the overall estimation errors. It is defined as.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n^{(\mathrm{T})}}\sum_{i=1}^{n^{(\mathrm{T})}}\left(y_i - \hat{y}_i\right)^2} \tag{14}$$

where $n^{(\mathrm{T})}$ is the total number of samples in the test dataset.

Fig. 3(a) shows the capacity estimation results in case 1. In this figure, the estimation result is compared with its ground truth, thereby the capacity estimation results are not continuous. It can be observed that the CNN can efficiently estimate battery capacity over the range of 0.86 to 1.98 Ah with the RMSE of 7.03 mA h. Such high fidelity confirms the end-to-end capacity estimation ability of CNN-based methods. However, deteriorated performance is observed when it comes to case 2. Owing to the lack of labelled training samples, the CNN experiences increased capacity estimation error with the RMSE surging to 36.45 mA h, as demonstrated in Fig. 3(b and c). By making use of unlabelled training samples, the JL-CNN witnesses an improved performance, as substantiated by the narrower error distribution with fewer outliers. The RMSE reduces by 26.06% to 26.95 mA h.

As the JL-CNN and supervised CNN shares identical structures of the feature extraction and capacity estimation modules, the auxiliary input reconstruction module plays a critical role. To in-depth check the contributions of the input reconstruction module, an EIS reconstruction RMSE is defined to evaluate the EIS reconstruction performance, and it is computed as.

$$\mathrm{RMSE}_Z = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left[\left(Z'_{f_i} - \hat{Z}'_{f_i}\right)^2 + \left(Z''_{f_i} - Z''_{f_i}\right)^2\right]} \tag{15}$$

The results for case 2 are plotted in Fig. 4(a), where both the input impedance spectra and reconstruction results are standard-

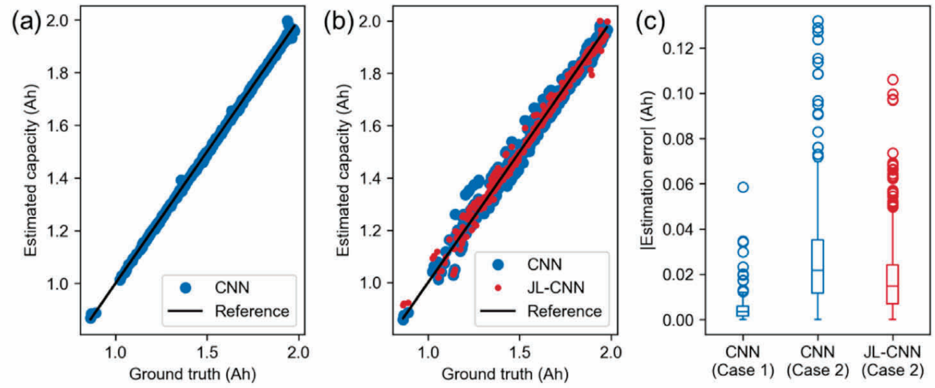

Fig. 3. Capacity estimation results using fully-charged impedance spectra of the 1-C dataset. (a) Estimation results in case 1. (b) Estimation results in case 2. (c) Absolute capacity estimation errors of cases 1 and 2.

ised according to Eq. (8) and the RMSEs are less than 0.047. In addition, two examples with the highest and lowest capacities are plotted in Fig. 4(b), where high-fidelity reconstructions can be intuitively observed. As mentioned early, the extracted features are 16 × 1 vectors, which have a much smaller size than the input impedance spectra (51 × 2 matrices). This design forces the feature extraction module to extract meaningful features that can concisely represent the input impedance spectra while excluding unnecessary information. The reconstruction results on the test dataset imply that the input reconstruction module takes effect, namely the input reconstruction module can not only guide the feature extraction module to learn capacity-relevant features from impedance spectra but also prevent it from overfitting to few labelled training samples.

4.2. Capacity estimation with different numbers of labelled samples

We further examine the performance of the proposed method in the presence of different labelled and unlabelled training samples. This is implemented by varying the LR from 10% to 90%. The distribution of capacity estimation errors of supervised CNN and JL-CNN are shown in Fig. 5 as a function of LR. In general, both methods show high accuracy when a large number of labelled training samples are available, e.g., the capacity estimation errors can maintain lower than 60 mA h for the label rate larger than 60%. An increasing RMSE trend is observed as LR decreases. On the other hand, the comparison between the results of CNN and JL-CNN reveals that the JL-CNN has generally narrower error distributions and fewer outliers.

Table 3 compares the performance of CNN, JL-CNN and a GPR-based method by computing their estimation RMSEs. The GPR-based capacity estimation method is developed in a supervised manner as a general benchmark. It was first proposed in [27] for capacity estimation by receiving raw impedance spectrum as input and was demonstrated to outperform the methods based on time-domain signals [36]. The comparison among the three methods reveals that the GPR has the highest error. As a traditional machine learning algorithm, it does not have intrinsic feature extraction ability as discussed in [28]. In contrast, CNNs stack advanced convolutional layers, max-pooling layers and dense layers to form deep architecture for feature extraction and regression, thereby showing superior performance. The RMSEs of the JL-CNN are within the range of [5.71, 26.95] mA h while those of the CNN are within [7.03,36.45] mA h. To quantify the improvement of JL-CNN over the supervised CNN, a relative error reduction (RER) [37] is defined as.

$$\mathrm{RER}=\left(1-\frac{\mathrm{RMSE}_{\mathrm{JL-CNN}}}{\mathrm{RMSE}_{\mathrm{CNN}}}\right)\times 100\% \tag{16}$$

The corresponding results are listed in Table 3. It can be seen that the JL-CNN reduces RMSE by over 14.00% in the case of differ-

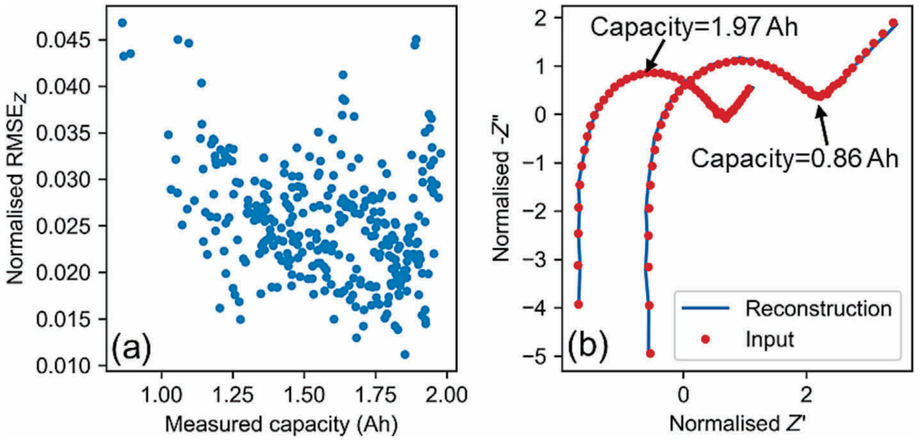

Fig. 4. (a) RMSEs of EIS reconstruction (b) EIS reconstruction examples with maximum and minimum measured capacities.

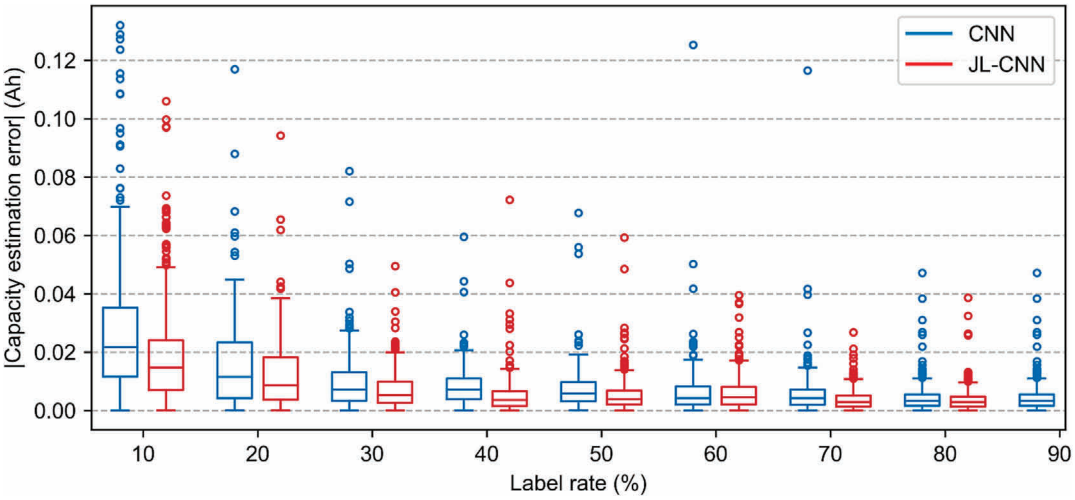

Fig. 5. Capacity estimation performance of the CNN and JL-CNN with different label ratios.

Table 3
Estimation RMSEs for the maximum and remaining capacities with different loss weights.

Label rate (%)	10	20	30	40	50	60	70	80	90
RMSE of GPR (mA h)	123.77	49.17	33.97	24.88	24.41	23.77	23.42	9.50	8.91
RMSE of CNN (mA h)	36.45	21.26	13.92	10.56	10.10	11.02	9.85	6.64	7.03
RMSE of JL-CNN (mA h)	26.95	16.64	9.98	8.24	8.09	8.41	5.22	5.71	–
RER (%)	26.05	21.70	28.26	21.93	19.91	23.73	46.98	14.00	–

ent label rates after adopting unlabelled training samples for model development.

4.3. Capacity estimation under different conditions

As a data-driven approach, the proposed method can be generalised to other battery degradation conditions. Particularly, high current rates can give rise to Li plating on the anode, resulting in different battery degradation patterns [38]. The 2-C dataset is adopted to explore the influence of current rates on capacity estimation. Besides, as EIS is reported to be sensitive to SOC variation, we investigate capacity estimation with fully-charged and fully-discharged impedance spectra, respectively.

The estimation results of case 2 are reported in Fig. 6 to illustrate the performance of JL-CNN with few training samples. In general, the results with different SOCs and current rates are consistent with the results reported above and the estimation errors can be reduced when the unlabelled samples are effectively employed for training. We further report the estimation RMSEs as a function of LR in Tables 4–6. Accurate estimation results can be obtained when sufficient labelled samples are given, and the RMSE is less than 18.82 mA h at the LR of 90% in various cases. The RMSEs increase as LR decreases because fewer labelled samples are available for training. In all cases, the JL-CNN can significantly improve the performance of the supervised CNN and the RER is in the range of [4.11%, 50.66%]. The results imply that the CNN is an effective measure to realise end-to-end capacity estimation using only raw impedance spectra as input and it can adapt to different battery operating conditions without manual feature extraction. Furthermore, the proposed JL-CNN can effectively utilise unlabelled impedance spectra to improve the estimation performance, which can relax the requirement of a large labelled training dataset.

4.4. Outlook and discussion

The validation results in this section demonstrate that the proposed CNN-based methods enable accurate battery capacity estimation under various conditions. By incorporating a reconstruction module, the proposed JL-CNN can further improve capacity estimation performance by exploiting unlabelled impedance spectra for training. As a result, the cost and efforts to collect battery capacity data can be significantly alleviated. For instance, in a lab-based scenario, researchers can adopt batteries that undergo various tests for a short-term EIS measurement. On the other hand, in industrial applications, EIS measurement has been reported to be feasible with various hardware [20]. Consequently, a considerable number of EIS samples can be gathered by resorting to cloud battery management systems [39].

The proposed method can be further improved in the future. First, advanced impedance analysis methods can be incorporated with the proposed deep learning based method. In particular, Lu et al. [21] systematically analysed the impedance spectra from the perspective of time scales through the distribution of relaxation times (DRT). Their results reveal that the DRT is promising to be used to augment the input of DNNs to reduce the effort to find meaningful features and help inform the DNNs with prior knowledge regarding battery degradation. In addition, thanks to the flexibility of deep learning models, our method can be generalised to other battery management tasks where unlabelled training data are easily available, such as life prediction [36] and SOC estimation [40]. Furthermore, as CNNs highlight the capability of feature extraction, it is an interesting research opportunity to investigate the correlation between extracted features and kinetic information hidden in impedance spectra [41], especially in the presence of different degradation patterns. Besides, the proposed method can be extended to real-world applications where low-quality samples are employed. This motivates further research on

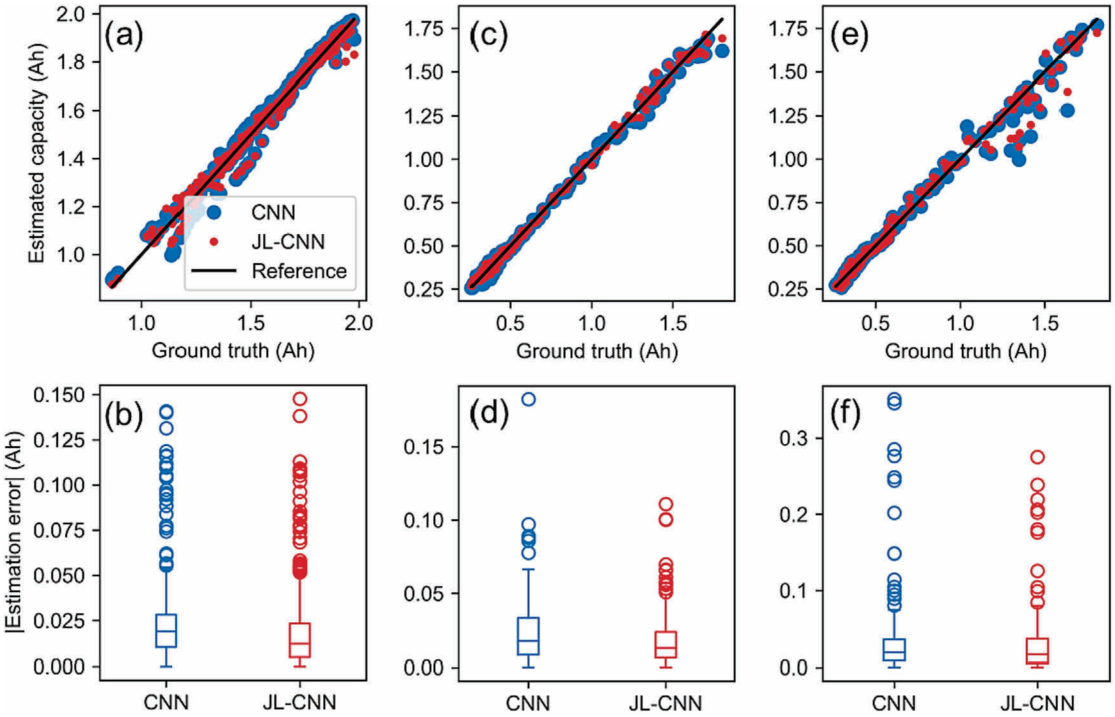

Fig. 6. Capacity estimation results using different inputs. (a, b) Capacity estimation results using fully-discharged impedance spectra of the 1-C dataset. (c, d) Capacity estimation results using fully-charged impedance spectra of the 2-C dataset. (e, f) Capacity estimation results using fully-discharged impedance spectra of the 2-C dataset.

Table 4
Capacity estimation errors using fully-discharged EIS of the 1-C dataset.

Label rate (%)	10	20	30	40	50	60	70	80	90
RMSE of CNN (mA h)	34.53	17.36	10.65	9.81	8.56	7.17	6.61	6.65	6.72
RMSE of JL-CNN (mA h)	30.95	15.81	10.21	6.80	6.20	6.17	5.91	5.03	–
RER (%)	10.35	8.89	4.11	30.73	27.60	14.01	10.60	24.32	–

Table 5
Capacity estimation errors using fully-charged EIS of the 2-C dataset.

Label rate (%)	10	20	30	40	50	60	70	80	90
RMSE of CNN (mA h)	33.57	17.35	15.74	14.16	13.32	14.00	13.78	12.07	11.91
RMSE of JL-CNN (mA h)	26.69	15.09	14.28	13.25	8.39	8.03	9.58	9.57	–
RER (%)	20.51	13.05	9.23	6.37	37.00	42.62	30.43	20.70	–

Table 6
Capacity estimation errors using fully-discharged EIS of the 2-C dataset.

Label rate (%)	10	20	30	40	50	60	70	80	90
RMSE of CNN (mA h)	69.43	48.57	39.33	24.97	17.77	20.53	15.57	16.85	18.82
RMSE of JL-CNN (mA h)	55.74	34.06	19.41	18.39	16.23	16.03	14.64	14.78	–
RER (%)	19.72	29.86	50.66	26.34	8.68	21.93	5.95	12.30	–

robust training strategies to accommodate data of different qualities, such as the Bayesian approach [42] to generate estimation confidence intervals.

5. Conclusions

Battery capacity estimation is an indispensable task for battery management systems. Although plenty of machine learning-based methods have been adopted to address this issue, they generally require a large number of samples that consist of both features and corresponding capacities for training.

In this study, we propose a convolutional neural network (CNN) based method for capacity estimation. Raw impedance spectra can be directly fed into the CNN for feature extraction and then capacity estimation, resulting in a flexible end-to-end estimation framework. Besides, an input reconstruction module is designed to augment the supervised CNN to a joint-loss CNN (JL-CNN) that can effectively employ impedance spectra without corresponding capacities to facilitate feature extraction.

Two large battery degradation datasets encompassing over 4700 impedance spectra collected from 16 batteries are developed for method validation. We show that the supervised CNNs can accurately estimate battery capacities using raw impedance spectra and the estimation RMSEs are within 7.03 mA h. However, deteriorated performance is observed when fewer labelled training samples are available and the errors can surge up to 36.45 mA h. Thanks to the proposed input reconstruction module, the JL-CNN can significantly reduce the RMSEs. Validations using impedance spectra collected from batteries working under different conditions further substantiate the effectiveness of the proposed method.

Our study offers an efficient way to utilise easily available unlabelled data to improve capacity estimation performance. It is also promising to be generalised to other battery management tasks where data-driven methods play a critical role.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Key R&D Program of China (2021YFB2402002), the National Natural Science Foundation of China (51922006 and 51877009), the China Postdoctoral Science Foundation (BX2021035 and 2022M710379), and the Beijing Natural Science Foundation (Grant No. L223013).

Appendix A. Layers in convolutional neural networks

1D convolutional layer: Given an input sequence $x_{1:L} = [x_1, x_2, \cdots, x_L]$, the 1D convolutional layer utilises n filters to convolve with the input sequence respectively, which can be expressed as.

$$x^{(k)} = \omega^{(k)} * x_{1:L} + \beta^{(k)} \tag{17}$$

where $\omega^{(k)}$ and $\beta^{(k)}$ denote the weight matrix and bias of the kth filter, respectively. The symbol * represents convolution. The output of the 1D convolutional layer is obtained by stacking the results from individual kernels.

$$y^{(C)} = \left[x^{(1),\mathrm{T}} x^{(2),\mathrm{T}} \ldots x^{(n),\mathrm{T}}\right]^{\mathrm{T}} \tag{18}$$

Max-pooling layer: The max-pooling layer is designed to condense its input by down-sampling through outputting the maximum in a given window. This process can be expressed as.

$$y_i^{(\mathrm{P})} = \max_{(i-1)l \le j \le il} x_j \tag{19}$$

where x denotes the input and l denotes the pooling size.

Dense layer: The dense layer is also known as the fully-connected layer. In this layer, a set of neurons are used to yield the dot between weights and layer input. Given an input sequence $x_{1:L} = [x_1, x_2, \cdots, x_L]$, the output is calculated as.

$$y_i^{(D)} = \sum_j \omega_{j,i} x_j + \beta_{j,i} \tag{20}$$

where $y_i^{(\mathrm{D})}$ represents the output of the ith unit in the dense layer. $\omega_{j,k}$ and $\beta_{j,k}$ are the weight and bias between the kth neuron and jth element in the input sequence.

ReLU activation: The ReLU activation function is designed to enhance model nonlinearity by outputting its positive argument. Given its input x, its output is.

$$y^{(\mathrm{ReLU})} = \max(0, x) \tag{21}$$

References

[1] J. Deng, C. Bae, A. Denlinger, T. Miller, Joule 4 (2020) 511–515.
[2] M.R. Palacín, A. de Guibert, Science 351 (2016) 1253292.
[3] H. You, J. Zhu, X. Wang, B. Jiang, H. Sun, X. Liu, X. Wei, G. Han, S. Ding, H. Yu, W. Li, D.U. Sauer, H. Dai, J. Energy Chem. 72 (2022) 333–341.
[4] K. Luo, X. Chen, H. Zheng, Z. Shi, J. Energy Chem. 74 (2022) 159–173.
[5] J. Tian, R. Xiong, W. Shen, J. Lu, F. Sun, Energy Storage Mater. 51 (2022) 372–381.
[6] Z. Tong, J. Miao, J. Mao, Z. Wang, Y. Lu, Energy Storage Mater. 50 (2022) 533–542.
[7] J. Tian, R. Xiong, W. Shen, eTransportation 2 (2019) 100028.
[8] J. Zhu, Y. Wang, Y. Huang, R. Bhushan Gopaluni, Y. Cao, M. Heere, M.J. Mühlbauer, L. Mereacre, H. Dai, X. Liu, A. Senyshyn, X. Wei, M. Knapp, H. Ehrenberg, Nat. Commun. 13 (2022) 2261.
[9] X. Hu, J. Jiang, D. Cao, B. Egardt, IEEE Trans. Ind. Electron. 63 (2016) 2645–2656.
[10] A. Ran, M. Cheng, S. Chen, Z. Liang, Z. Zhou, G. Zhou, F. Kang, X. Zhang, B. Li, G. Wei, ENERGY Environ. Mater. (2022), https://doi.org/10.1002/eem2.12386.
[11] C. Weng, X. Feng, J. Sun, H. Peng, Appl. Energy 180 (2016) 360–368.
[12] Y. Li, M. Abdel-Monem, R. Gopalakrishnan, M. Berecibar, E. Nanini-Maury, N. Omar, P. van den Bossche, J. Van Mierlo, J. Power Sources 373 (2018) 40–53.
[13] X. Tang, C. Zou, K. Yao, G. Chen, B. Liu, Z. He, F. Gao, J. Power Sources 396 (2018) 453–458.
[14] R.R. Richardson, C.R. Birkl, M.A. Osborne, D.A. Howey, IEEE Trans. Ind. Informatics 15 (2019) 127–138.
[15] J. Tian, R. Xiong, W. Shen, F. Sun, Energy Storage Mater. 37 (2021) 283–295.
[16] J. Tian, R. Xiong, W. Shen, J. Lu, X.-G. Yang, Joule 5 (2021) 1521–1534.
[17] M. Gaberšček, Nat. Commun. 12 (2021) 6513.
[18] S.S. Zhang, J. Energy Chem. 41 (2020) 135–141.
[19] D. Andre, M. Meiler, K. Steiner, C. Wimmer, T. Soczka-Guth, D.U. Sauer, J. Power Sources 196 (2011) 5334–5341.
[20] X. Wang, X. Wei, J. Zhu, H. Dai, Y. Zheng, X. Xu, Q. Chen, eTransportation 7 (2021) 100093.
[21] Y. Lu, C.Z. Zhao, J.Q. Huang, Q. Zhang, Joule 6 (2022) 1172–1198.
[22] D. Capkova, V. Knap, A.S. Fedorkova, D.I. Stroe, J. Energy Chem. 72 (2022) 318–325.
[23] X. Chen, L. Li, M. Liu, T. Huang, A. Yu, J. Power Sources 496 (2021).
[24] M. Messing, T. Shoa, S. Habibi, J. Energy Storage 43 (2021).
[25] K. Mc, Carthy, H. Gullapalli, T. Kennedy, Appl. Energy 307 (2022).
[26] Y. Fu, J. Xu, M. Shi, X. Mei, IEEE Trans. Ind. Electron. 69 (2022) 7019–7028.
[27] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, A.A. Lee, Nat. Commun. 11 (2020) 6–11.
[28] S. Kim, Y.Y. Choi, J.-I. Choi, Appl. Energy 308 (2022).
[29] V. Sulzer, P. Mohtat, A. Aitio, S. Lee, Y.T. Yeh, F. Steinbacher, M.U. Khan, J.W. Lee, J.B. Siegel, A.G. Stefanopoulou, D.A. Howey, Joule 5 (2021) 1934–1955.
[30] X. Shu, S. Shen, J. Shen, Y. Zhang, G. Li, Z. Chen, Y. Liu, iScience (2021) 103265.
[31] Y. LeCun, Y. Bengio, G. Hinton, Nature 521 (2015) 436–444.
[32] Y. Duan, J. Tian, J. Lu, C. Wang, W. Shen, R. Xiong, Energy Storage Mater. 41 (2021) 24–31.
[33] D.P. Kingma, J. Ba, in: the 3rd International Conference for Learning Representations, arxiv, San Diego, 2015, pp. 1-15
[34] X. Zhou, J. Huang, J. Energy Storage 31 (2020).
[35] D. Andre, M. Meiler, K. Steiner, H. Walz, T. Soczka-Guth, D.U. Sauer, J. Power Sources 196 (2011) 5349–5356.
[36] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Nat. Energy 4 (2019) 383–391.
[37] H. Tu, S. Moura, Y. Wang, H. Fang, ArXiv Prepr. ArXiv2112.12979 (2021).
[38] X. Lin, K. Khosravinia, X. Hu, J. Li, W. Lu, Prog. Energy Combust. Sci. 87 (2021).
[39] W. Li, M. Rentemeister, J. Badeda, D. Jöst, D. Schulte, D.U. Sauer, J. Energy Storage 30 (2020).
[40] J. Tian, R. Xiong, J. Lu, C. Chen, W. Shen, Energy Storage Mater. 50 (2022) 718–729.
[41] J. Liu, F. Ciucci, J. Electrochem. Soc. 167 (2020).
[42] Y. Gal, Z. Ghahramani, in: M.F. Balcan, K.Q. Weinberger (Eds.), Proceedings of the 33rd International Conference on Machine Learning, Association for Computing Machinery, New York, 2016, pp. 1651–1660.