A Method for Battery Capacity Prediction in the Grading Stage Based on Deep Learning Neural Networks

Authors: Sun Yu, Sun Yu

Date: 2021-05-29T00:00:00+00:00

Abstract

This paper proposes a deep learning-based solution for predicting the capacity of lithium-ion batteries during the grading process. This solution extracts physical observation records from partial steps of the formation and grading procedures as features, trains a deep neural network (Deep Neural Network, DNN), and achieves accurate prediction of battery capacity. Experimental results demonstrate that the Mean Absolute Percentage Error (MAPE) between the battery capacity predicted by this model and the true values is only 0.78%. Integrating this model with the production line can significantly reduce production time and energy consumption, thereby lowering battery production costs.

Full Text

A Deep Neural Network-Based Method for Predicting Lithium-Ion Battery Capacity in the Grading Process

Sun Yu[1]*
[1]Guangzhou Lingbo Technology Co., Ltd.
*Corresponding author (Email: suny226@mail2.sysu.edu.cn)

Abstract

This paper proposes a deep learning solution for predicting lithium-ion battery capacity during the grading process. The method extracts physical observation records from partial steps in the formation and grading procedures as features, and trains a Deep Neural Network (DNN) to achieve accurate battery capacity prediction. Testing demonstrates that the model achieves a Mean Absolute Percentage Error (MAPE) of only 0.78% compared to actual measured values.

Integrating this model into production lines can significantly reduce production time and energy consumption, thereby lowering battery manufacturing costs.

Keywords: Lithium-ion battery capacity prediction; deep learning; neural network; grading

In lithium-ion battery manufacturing, capacity data from the battery grading process serves as a critical quality indicator for individual cells before shipment. Manufacturers classify batteries into different capacity tiers and group cells from the same tier into packs to ensure better stability and lifespan.

The conventional method for obtaining lithium-ion battery capacity data involves fully charging the battery using charge-discharge equipment, then performing multiple stepped-current discharge cycles until complete discharge. The total capacity from each discharge step is summed to determine the final shipping capacity. This approach suffers from extreme time and energy consumption—for a standard 3200mAh 18650 battery, the entire charge-discharge process takes 4–5 hours, with each stage consuming substantial electrical energy.

Predicting battery capacity during production through technical means can satisfy engineering requirements for battery grouping while simultaneously reducing production time and energy usage. Traditional capacity prediction methods typically assume linear voltage changes, combining voltage differences at two time points within a step to estimate the target voltage timing, thereby approximating total discharge time and capacity (Figure 1 [Figure 1: see original paper]). However, this approach requires actual battery discharge and suffers from poor prediction accuracy.

In recent years, deep learning technology has advanced considerably, with various deep learning models increasingly applied to data prediction research. Deep learning models are neural networks with multiple nonlinear mapping layers that can abstract input information across hidden layers to extract features and uncover deeper latent patterns. These models have demonstrated excellent performance in applications including object detection, Natural Language Processing (NLP), mathematical proof improvement, and molecular model prediction.

In this work, we extract features from partial data of individual lithium-ion batteries under charging conditions and train a deep neural network to predict total discharge capacity, achieving excellent prediction results.

1 Deep Neural Networks

Neural networks are extensions of the perceptron, and Deep Neural Networks (DNN) can be understood as neural networks with many hidden layers. DNNs are sometimes referred to as Multi-Layer Perceptrons (MLP) [**?**].

[Figure 2: see original paper] DNN Neural Network Standard Model

Based on layer position, DNN internal layers can be divided into three categories: input layer, hidden layers, and output layer. Typically, the first layer is the input layer, the last layer is the output layer, and all intermediate layers are hidden layers. Layers are fully connected, meaning any neuron in layer i is connected to every neuron in layer i+1. Although DNNs appear complex, at the local level they operate like perceptrons: a linear relationship plus an activation function $\sigma(z)$.

Using the classic DNN model in Figure 2 as an example, when a matrix sequence $(x_1, x_2, x_3 \cdots x_n)$ is input through the input layer, its primary features are obtained through hidden layer 1 as feature sequence $z = \Sigma w_i x_i + b$, then secondary feature sequence $W = (w_{x1}, w_{x2}, w_{x3} \cdots w_{x_n})$ is obtained through hidden layer 2, and so forth, ultimately producing the output matrix $H = \begin{pmatrix} w_{x1} \\ w_{x2} \\ w_{x3} \\ \vdots \\ w_{x_n} \end{pmatrix}$.

In practice, we typically use backpropagation to control the model fitting process. With a fixed number of neurons in the input and output layers plus some hidden layers containing neurons, we must find appropriate linear coefficient matrices W and bias vectors b for all hidden and output layers, such that the computed results from input-hidden-output layer processing approximate the true measured values as closely as possible for all training samples.

To achieve this, we first select an appropriate loss function to measure the difference between computed outputs and corresponding true measured values from training samples. We then apply a suitable optimizer to minimize this loss function, continuously updating the linear coefficient matrices W and bias vectors b in reverse until achieving the desired effect.

2.1 Experimental Data

This study uses data from 18650 batteries produced by a lithium-ion battery manufacturer. The battery specifications are: rated voltage 3.6V, rated capacity 3.2Ah, rated internal resistance 28mΩ, maximum charging voltage 4.2V, discharge cutoff voltage 2.5V, and charge-discharge temperature 26$±$5°C. The dataset comprises 100,000 individual battery records, with the complete charge-discharge profile for a single battery shown in Figure 3 [Figure 3: see original paper].

[Figure 3: see original paper] Battery Charge-Discharge Overall Data Chart

The data is split into approximately 80% (80,000 batteries) for training, from which 20% (16,000 batteries) is further extracted as validation data. The test dataset uses the remaining approximately 20,000 batteries that did not participate in training or validation, maintaining identical data structure to the experimental data.

2.2 Feature Matrix

Data is stored in CSV format, with raw data containing 11 numerical features: sampling time, cycle, step, working status, working time, voltage, current, capacity, energy, temperature, and battery anomaly marker (see Table 1).

```
2021-02-28 14:18:43,0,0,3,0.0,3604.5,0,0.0,0.0,23.4,0,
2021-02-28 14:19:24,0,0,3,0.4,3604.6,1.5,0.0,0.0,23.3,0,
2021-02-28 14:19:27,0,0,3,0.4,3604.6,-3.5,0.0,0.0,23.2,0,
2021-02-28 14:19:31,0,0,3,0.5,3604.5,-1.1,0.0,0.0,23.2,0,
2021-02-28 14:19:34,0,0,3,0.5,3604.5,0.2,0.0,0.0,23.2,0,
2021-02-28 14:19:38,0,0,3,0.6,3604.6,0.1,0.0,0.0,23.1,0,
2021-02-28 14:19:41,0,0,3,0.7,3604.5,-0.1,0.0,0.0,23.1,0,
2021-02-28 14:19:45,0,0,3,0.7,3604.6,-0.3,0.0,0.0,23.1,0,
2021-02-28 14:19:48,0,0,3,0.8,3604.6,0.0,0.0,0.0,23.1,0,
2021-02-28 14:19:52,0,0,3,0.8,3604.5,-0.1,0.0,0.0,23.1,0,
2021-02-28 14:19:56,0,0,3,0.9,3604.5,-0.2,0.0,0.0,23.1,0,
2021-02-28 14:19:59,0,0,3,1.0,3604.6,-0.2,0.0,0.0,23.1,1,Time elapsed
```

Table 1: Raw Data Example

We identify current voltage, current, and accumulated capacity as suitable features for neural network input, organizing them chronologically to form the final input feature matrix.

2.3 Data Preprocessing

Approximately 0.5% of battery data in the raw dataset (Table 2) exhibits anomalies during charge-discharge (including equipment, battery, or contact issues), specifically identified by anomaly marker values >10. This invalid data interferes with normal model training and prediction, so samples exhibiting this condition are removed during feature extraction.

```
2020-11-04 06:14:40,0,0,3,0.0,879.4,0,0.0,0.0,29.7,0,
2020-11-04 06:15:05,0,0,3,0.0,889.4,0.9,0.0,0.0,29.6,3020,OCV anomaly
2020-11-04 06:15:20,0,0,3,0.4,889.3,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:23,0,0,3,0.4,889.2,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:27,0,0,3,0.5,889.1,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:30,0,0,3,0.5,888.9,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:34,0,0,3,0.6,888.8,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:37,0,0,3,0.7,888.7,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:41,0,0,3,0.7,888.5,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:45,0,0,3,0.8,888.4,0.9,0.0,0.0,29.5,3020,OCV anomaly
2020-11-04 06:15:49,0,0,3,0.8,888.5,0.8,0.0,0.0,29.5,3020,OCV anomaly
```

Table 2: Anomalous Data Example

In both training and test datasets, the shipping capacity is obtained by sequentially accumulating the final capacity from all discharge steps, which serves as

the dependent variable during model training. Prior to training, the data undergoes additional filtering: (1) Z-scores are calculated for the dependent variable using the SciPy module, retaining only samples with absolute z-score < 3 [**?**]; (2) Features are filtered based on missing value proportion. As shown in Figure 4 [Figure 4: see original paper], the total area under the curve bounded by current and time (area under the curve in the XoY plane) represents the battery' s shipping capacity.

[Figure 4: see original paper] Definition of Shipping Capacity

2.4 Network Training

We construct and train the network using the TensorFlow Keras API (v2.4.1) [**?**]. The architecture consists of an input layer, one hidden layer with 2048 neurons, and an output layer, using Gaussian Error Linear Unit (GELU) as the activation function [**?**]. The model is compiled with Mean Absolute Percentage Error (MAPE) as the loss function and NAdam as the optimizer [**?**].

Training converges after 4739 epochs (Figure 5 [Figure 5: see original paper]), achieving a validation MAPE of approximately 0.8251%.

[Figure 5: see original paper] Model Training Process

3 Prediction Performance

To further demonstrate the model' s performance in production environments, this section tests approximately 20,000 batteries with data structure consistent with the original dataset. The test data undergoes identical preprocessing as described in Section 2.3. By comparing predicted values with actual measurements, the Absolute Percentage Error (APE) is calculated for each battery.

[Figure 6: see original paper] Battery Capacity Value Distribution
[Figure 7: see original paper] Prediction Error Distribution in [0, 1] Range

In Figure 6, the red line represents predicted shipping capacity for each battery, while the blue line shows actual shipping capacity. The maximum observed error is 4.6%, minimum error is 0.0032%, and average error is 0.83%. Figure 7 shows the APE distribution in the [0, 1] interval, with values concentrated near the y-axis, indicating excellent model performance on the vast majority of samples.

4 Conclusion

To address lithium-ion battery shipping capacity prediction, this paper extracts feature values from partial charge-discharge steps and establishes a DNN model. Through training, validation, and testing, the model effectively predicts final shipping capacity with an average error of only 0.83%, demonstrating high estimation accuracy.

In actual production, the model can be fine-tuned according to the physico-chemical properties of specific battery batches to maintain similar prediction performance. This approach exhibits strong generalization capability and is suitable for deployment in industrial environments. Integrating this method with production line MES systems, SCADA systems, or host computer systems can substantially reduce production time and energy consumption in practical applications. This provides enterprises with feasible technical support, actively responds to national energy conservation and emission reduction initiatives, and holds profound application value.

References

[?] GU J, WANG Z, KUEN J, et al. Exploring Strategies for Training Deep Neural Networks [J]. 2014.

[?] OLIPHANT T E. SciPy: Open source scientific tools for Python [J]. 2014.

[?] TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems [J]. 2016.

[?] HENDRYCKS D, GIMPEL K. Gaussian Error Linear Units (GELUs) [J]. 2016.

[?] SINGARIMBUN R N, NABABAN E B, SITOMPUL O S. Adaptive Moment Estimation To Minimize Square Error In Backpropagation Algorithm; proceedings of the 2019 International Conference of Computer Science and Information Technology (ICoSNIKOM), F, 2019 [C].

Note: Figure translations are in progress. See original paper for figures.