CS230

Predicting Cycling Conditions of Lithium-ion Batteries through Air-free Disassembly Imaging

Jerry Liao*
Department of Materials Science and Engineering
Stanford University
liao1226@stanford.edu

Abstract

Without traveling back in time, we are able to know how a battery was charged by feeding the air-free disassembled graphite images into our Convolutional Neural Network. A simple 3-layer neural network was first implemented to check if the idea of predicting cycling conditions through air-free disassembly imaging works. Surprisingly, this simple model ended up revealing exciting scientific results – charging rate dominates the visual appearance of a cycled graphite. We proceeded to build a deep 25-layer CNN that is fed with raw colored images without feature extractions. A 70% accuracy which is close to human level performance (80%) was achieved. Such technique, I believe, will be of great value to both battery industry and academic research in the future.

1 Introduction

Lithium-ion batteries have become the first choice for electronic devices. Among all the commercial batteries, graphite is the dominant anode material, but it suffers from capacity fading upon intense cycling[1]. Recently, researchers at Stanford discovered that the graphite images under an air-free condition might provide clues on how the battery was cycled. Nevertheless, the study of the relationship between visual data and cycling conditions has yet to be accomplished.

To test the hypothesis, we began with a simple 3-layer fully connected neural network model that uses a Mean Absolute Error (MAE) loss function:

$$MAE = \frac{\sum_{i=1}^{n} |y_i - a_i|}{n}$$

Our input is an array containing 12 features extracted from an air-free disassembled graphite image; we use 3 linear activation neurons to output the predicted values of 3 different cycling conditions – charging c-rate (from 3c to 6c), discharging c-rate (from 2c to 8c), and cycle number (from 100 to 1000). The input array looks like this: [average pixel value, STD of pixel values, average of red pixel values, average of green pixel values, average of blue pixel values, STD of red pixel values, STD of green pixel values, STD of blue pixel values, fraction of light pixels that are above 1 STD, fraction of dark pixels that are below 1 STD, fraction of light pixels that are above 3 STD, fraction of dark pixels that are below 3 STD]

Despite the fact that the outputs were erroneous, we found it promising to further study charging rate by feeding raw images to a complex 25-layer deep convolutional neural network that uses a

*https://www.linkedin.com/in/jerry-liao-006a90131/

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

categorical crossentropy loss function:

$$CE = -\sum_{i=1}^{C} v_i log(\frac{e^{a^i}}{\sum_{k=1}^{n} e_k^a})$$

However, only 50 graphite images were collected due to the time-consuming air-free disassembly procedure. To solve the insufficiency quandary, data augmentations were employed – we sectioned every graphite image into 12∼13 equal sub-images – which increased the size of data set by more than 10 times. As a result, we are able to have an input data set of 524 images along with labels converted to one-hot arrays. The output layer has one softmax neuron outputting predicted charging rate from 3c $\sim 6c$.

2 Related work

$Battery\ Work$

X. Fleury et al[2]. also studied graphite degradation mechanism from visual data, yet neither the images were scanned nor colored, which creates nonuniform lighting , different depth, and nonobvious degraded pattern.

$Computer\ Work$

The first simple neural network structure fed with 12 extracted features took inspiration from Kanade [3] who presented an automatic feature extraction method based on ratios of distances. Our CNN architecture was inspired by "Face Recognition: A Convolutional Neural-Network Approach" [4]. Steve et al. presented a high-level block diagram of the system for face recognition that has a Conv2D followed by a pooling sequence. The weakness of this is the shallow structure which was limited by the computational power in 1997. The implementation of our structure also took significant inspiration from "Densely Connected Convolutional Networks" by Gao et al. [5] while tuning hyper-parameters and the github repository of Francois Chollet [6] that showed a useful Conv2D, BN, relu, and pooling architecture for our graphite image recognition.

3 Dataset and Features

$Simple\ Neural\ Network$

A total of 42 graphite images, resolution of 890 x 11000, were obtained. 32 of them are in X$_{train}$; 10 of them are in X$_{test}$. For each graphite image, we extracted 12 features and put them in an array – mean pixel value, std of pixel values, mean R/G/B values, std of R/G/B values, fraction of light/dark pixel values, fraction of very light/very dark pixel values – as described in the introduction section. A feature extractor is implemented to extract the following classes of features from the electrode images:

1. *Mean and standard deviation of pixel values.*
   This is chosen because it is observed that electrodes cycled under harsher conditions tend to be darker in color.
2. *Mean and standard deviation of R/G/B values.*
   This is chosen because electrodes cycled under different conditions appear to show differences in hues.
3. *Fractions of light/dark pixel values (defined to be those exceeding 1 or 3 standard deviations from the mean).*
   This is chosen based on the rationale that electrodes cycled at harsher conditions tend to show greater inhomogeneity in their color distributions.

Each element in the array was normalized by the following formula: $\frac{f_i - mean}{std}$

Figure 1. Scanned image of a whole graphite anode.

Convolutional Neural Network

A total of 50 graphite images, resolution of 890 x 11000, were obtained. Each graphite image was sectioned into of resolution 890 x 914, which augmented the number of total images to 625. Our dataset looks like this: 472 of them are in X_{train} ; 53 of them are in X_{dev} ; 100 of them are in X_{test}. Before inputting into CNN, all sectioned images were normalized by 255 to speed up the training process.

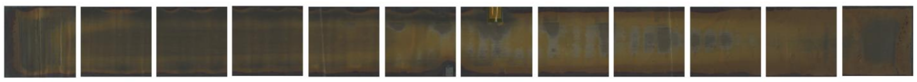

Figure 2. 12 sections of a whole graphite anode.

4 Methods

Simple Neural Network

The array of 12 features serves as input to a three-layer neural network of the following architecture:

Layer	Number of Nodes	Activation
1	64	ReLU
2	64	ReLU
3	3	Linear

Figure 3. Architecture of Simple Neural Network

A simple architecture is chosen due to the relatively few training data and features available. The three nodes of the last layer output the predictions for the discharge and charge C-rate, and the cycle number of the given electrode. Here, cycle number is normalized by a factor of 100 with respect to the C-rates to bring them to the same order of magnitude. The neural network is trained using two loss functions separately, namely that of mean squared error (MSE) and mean absolute error (MAE). The root-mean-square prop (RMSProp) algorithm is used as an alternative to standard stochastic gradient descent, such that the learning rate is divided by an exponentially decaying average of squared gradients to compensate for decreasing gradients and hence speed up convergence.

Convolutional Neural Network

Figure 4 shows the structure of a "battery block" which consists of 7 layers and gives decent predictions.

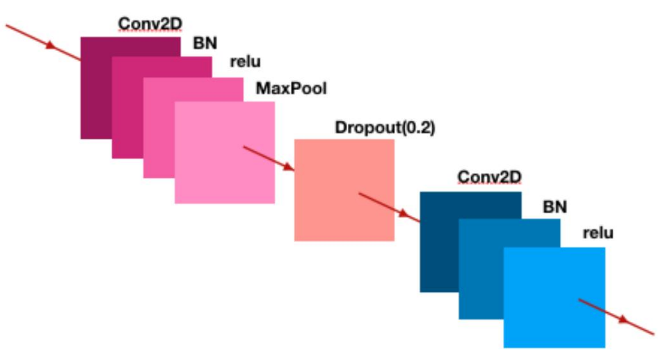

Figure 4. Architecture of a battery block.

Conv2D The filter size is (2,2) with stride (2,2). We have a relatively deep CNN, so small filter size as well as (2,2) stride make our CNN run faster. *BN* Batch normalization is chosen to deal with covariate shift. *relu* We chose relu activation because its linearity speeds up the training process. *MaxPool* We chose MaxPool because the brightest pixel value matters – it indicates critical feature, either super degraded or super healthy. *Dropout(0.2)* Dropout prevents our CNN from over-fitting. We chose 0.2 because it balances the over-fitting and training rate trade off.

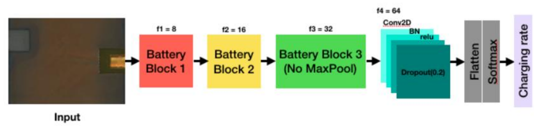

Figure 5. Architecture of our CNN.

A CNN architecture is chosen due to its magnificent power of visual feature detection. The three battery blocks elongate the shape of our input data, from (914, 890, 3) to (1, 1, 64). With the number of filters increases from 8 to 64, the CNN seems to find related features through 17,207 trainable parameters. The final node of the softmax layer outputs the prediction for charging c-rate ranging from 3 to 6. The CNN is trained using categorical crossentropy loss function:

$$CE = -\sum_{i=1}^{C} v_i log(\frac{e^{a^i}}{\sum_{k=1}^{n} e_k^a})$$

which is the typical loss function for a multi-classification task.

5 Experiments/Results/Discussion

Simple Neural Network

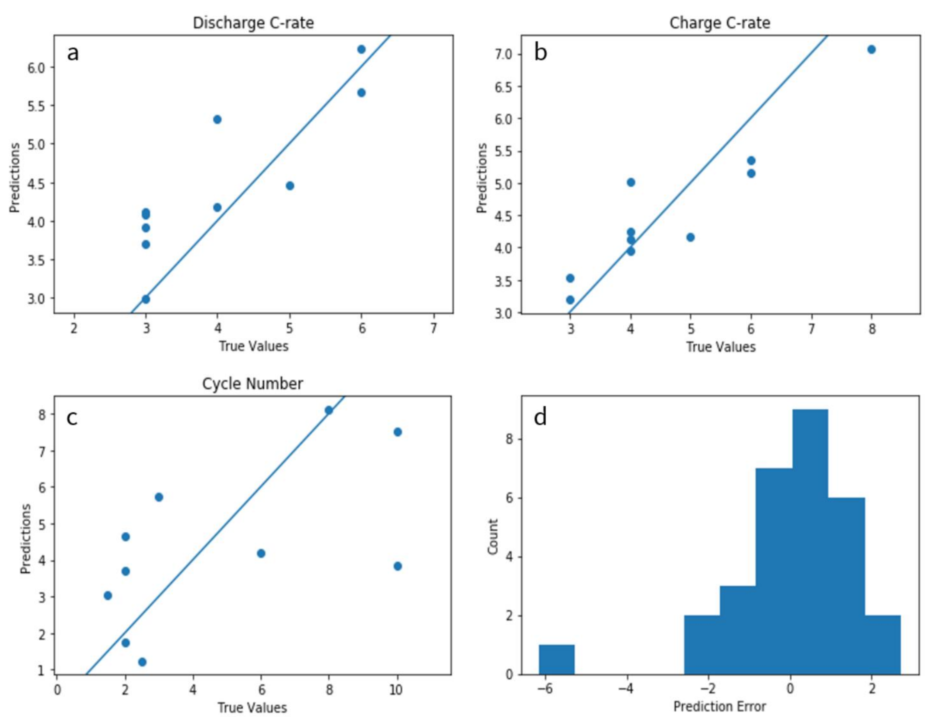

Figure 6. (a-c) Predicted values against true values for discharge C-rate, charge C-rate and cycle number respectively, for the neural network trained using the simple features and the MAE loss function. The solid lines indicate the ideal case in which predictions are equal to true values. (d) Histogram of compiled prediction error over all cycling parameters.

The errors between predicted and true cycling conditions of the evaluation set are found to be 2.55 and 1.08 for the MSE and MAE loss functions respectively. These are of the same order of magnitude as the true values and hence suggest considerable inaccuracy. Nevertheless, it appears that the neural network trained using the MAE loss function performs better on the evaluation data.
Figure 6 shows the errors in the MAE neural network. Notably, the prediction for the charge C-rate appears to be considerably more accurate than the other two conditions. This suggests that there may be a stronger relationship of causation between the charge C-rate and the degradation patterns of the battery. The histogram of compiled prediction errors exhibits a distribution centered and concentrated at zero, but there remain instances of significant inaccuracy in prediction.

$Convolutional\ Neural\ Network$

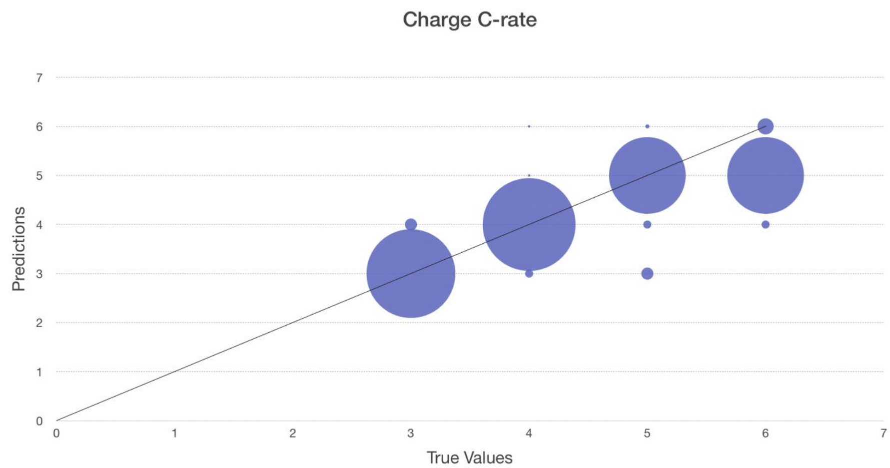

Figure 7. Predicted values against true values for charge C-rate for the CNN trained using the cropped images and categorical crossentropy loss function. The solid lines indicate the ideal case in which predictions are equal to true values. The primary metrics, accuracy, gives 70% acc and the loss in test set is 0.87.

Figure 7 shows the plot of Predicted values against true values for charge C-rate with an acc 70%. As we can see, while most 6c tends to be classified as 5c, the output values have less error compared to the previous NN results.
We chose the filter size and max pooling size to be (2,2) with stride (2,2). We have a relatively deep CNN, so small filter size as well as (2,2) stride make our CNN run faster. We chose 0.2 drop probability for dropout function because it balances the over-fitting and training rate trade off. For mini-batch size, we sampled 10 to 100 and ended up choosing 30 because we were able to see updates and get a decent amount of progress at the same time.
Interestingly, if the training set, validation set and test set contain either inside or outside only, the training accuracy is not as good, which indicates the importance of having large size of data set for CNN.

6 Conclusion/Future Work

A CNN capable of showing charging rate of electrodes through visual images has been constructed. Even though CNN appears to be a more appropriate approach, the simple NN notes that the charge C-rate have a stronger correlation with electrode degradation than the other cycling conditions.

For future work, I would like to explore a saliency map, which will tell us the degree to which each pixel in the image affects the classification score for that image. In other words, we can get scientific intuitions on where is the vulnerable part and how to improve it.

References

[1] Maire, P., Evans, A., Kaiser, H., Scheifele, W., & Novák, P (2008) Colorimetric Determination of Lithium Content in Electrodes of Lithium-Ion Batteries. *Journal of The Electrochemical Society*

[2] X. Fleury, M.H. Noh, S. Genies, P.X. Thivel, C.Lefrou, & Y. Bultel (2018) Fast-charging of lithium iron phosphate battery with ohmic-drop compensation method. *Journal of Energy Storage*

[3] T. Kanade. (1973) Picture processing by computer complex and recongnition of human faces. *Ph.D. dissertation*, Kyoto Univ., Japan

[4] L. Steve, et al., (1997) Face Recognition: A Convolutional Neural-Network Approach. *IEEE transactions on neural networks, vol. 8, no. 1, January*

[5] Huang, G., et al., (2017) Densely Connected Convolutional Networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*

[6] https://github.com/fchollet/deep-learning-models/blob/master/resnet50.py

[7] Library used: Tensorflow, Keras, opencv