A DEEP NEURAL NETWORK STRUCTURE FOR LI-ION BATTERY STATE OF CHARGE ESTIMATION AND HARDWARE IMPLEMENTATION IN ELECTRIC VEHICLES

[1]MUH-TIAN SHIUE, [2]YI-FONG WANG, [3]YANG-CHIEH OU, [4]BING-JUN LIU, [5]PING-HAO LIU

Department of Electrical Engineering, National Central University, Taoyuan City 320,Taiwan(R.O.C)
E-mail: mtshiue@ee.ncu.edu.tw, 109521017@cc.ncu.edu.tw, 995401001@cc.ncu.edu.tw

Abstract - This study proposes a battery model that predicts the Li-ion SOC of electric vehicles using a deep neural network (DNN) structure. Unlike most machine learning-based models, which employ large neural networks with billions of parameters to determine target functions, this study aims to achieve high accuracy rates using smaller DNN models. The study utilizes data collected from the Center for Advanced Life Cycle Engineering (CALCE) and verifies the accuracy of the DNN model using Federal Urban Driving Schedule (FUDS) drive cycle datasets. The study also uses actual electric vehicle battery data from Argonne National Laboratory for testing the DNN model. This approach makes the hardware implementation of deep neural network estimation more efficient. The study develops and evaluates a series of DNN models with varying numbers of neurons and finds that increasing the number of neurons does not reduce the error rate, but it increases costs. Additionally, the study reduces the complexity of the neural network algorithm circuit using battery measurement circuit accuracy analysis[2]. The objectives of this study are to demonstrate that DNN models can accurately estimate the SOC of Li-ion batteries with sufficient battery data, identify the optimal number of neurons for SOC estimation, and implement the testing part of the DNN model in hardware.

Keywords - Battery, Dnn, State Of Charge

I. INTRODUCTION

The Battery Management System (BMS) is a complex system consisting of different types of sensors that ensure the safe and efficient use of battery energy [1]. One of the crucial tasks of the BMS is to estimate the State-of-Charge (SOC) of the battery. However, estimating the SOC of Li-ion batteries is a challenging task as they are electrochemical devices and cannot be directly measured by sensors. However, it is possible to evaluate the SOC using available parameters such as voltage and current. The battery SOCis defined as:

$$\mathrm{SOC} = \mathrm{Q_{available}} \ / \mathrm{Q_{rated}} \ (1)$$

$\mathrm{Q_{available}}$ means the battery remaining charge, and $\mathrm{Q_{rated}}$ means the battery rated capacity. A few years ago, many researchers use Lookup Table, Coulomb Counting and model-based methods to estimate the SOC. However, there are still plenty of shortcomings in Lookup Table, Coulomb Countingand model-based methods. On the other hand, more and more works use data-driven methods to estimate the battery SOC in recent years. Deep neural network (DNN) is the most popular machine learning algorithm among the Data-Driven methods, which mimics the functioning of the human brain to find the correlations between input dataset and output dataset with a logical network structure. DNN uses multiple hidden layers as opposed to traditional neural networks. Conventional model-based algorithms of the battery SOC estimation rely on knowledge about battery. However, the DNN model is supposed to learn the battery behaviors by itself.

A deeplearning process contains two parts, that is, training andinferring. DNN models are trained by a huge set of labeled data without the need of manual extraction. Then the DNN system compares those features of the labeled input data and updates the parameters to memorize the features. This training process will be repeated many times according to the error rate. During the inferring phase, the DNN model finally can make conclusions and estimation the answer of the unexposed data.

II. SIGNIFICANCE

The research on deep learning has gone through a significant period of time, and the current consensus is that deep learning is a black-box model. While we can obtain local minimums through algorithms such as gradient descent, the logic behind the intermediate parameters remains unclear. This paper aims to obtain parameters for a deep learning model, with the goal of improving the accuracy of state of charge (SOC) prediction for batteries. Also, we try to implement the hardware of the testing part on the FPGA board. So, this work uses piecewise linear (PWL) functions to approximate the activation function, then quantize the PWL function for the hardware implementation. Using the quantized function with the specified word length to construct the DNN model and finally obtain the simulation results of SOC prediction.

III. METHOD

A. Deep Neural Network Structure

DNN is a sort of feedforward neural network architecture, containing an input layer, multiple

hidden layers, and an output layer [3]. Every layer is made up of neurons. The output of a single neuron can be expressed as following function:

$$y^l = \sigma \sum_q w^l y^{l-1} + b^l \qquad (2)$$

where y^l is the output of layer, w^l denotes the weights of layer, and b^l denotes the bias of layer. q is the number of neurons in layer. σ is the activation function. The activation function is a nonlinear function. If the function of a neuron doesn't contain a activation function, the whole DNN model will be a linear system. So, the output of a single neuron will be delivered to next layer until the last layer. After lots of computations, the DNN model will provide a predicted numeric value. Fig.1 shows the DNN architecture. This paper uses Gradient Descent algorithm to update the weights and the biases of the DNN model. The formula below sums up the entire Gradient Descent:

$$w_{new} = w_{old} - \eta \times (\partial L w_{old} / \partial w_{old}) \qquad (3)$$

η is the learning rate, which is a hyperparameter that controls how much to alter the model in response to the estimated error for each time the model weights are updated.

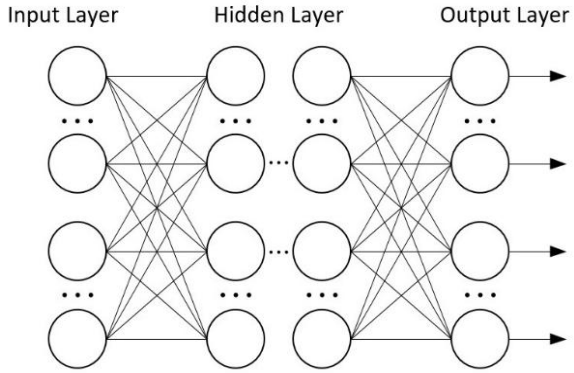


Fig. 1. The DNN architecture

B. Battery Data

As mentioned previously, this paper uses open available data from CALCE. The battery data was collected from applying the different drive cycles onto 18650-20R Li-ion battery. The dataset includes measurements of current, voltage, and state of charge .A drive cycle is a series of data points abouts the speed of a vehicle. A drive cycle is produced to access the performance of vehicles, such as fuel consumption and polluting emissions. The most common drive cycle for DNN model is FUDS[4]. Hyperparameters have a great influence on the behavior of the DNN model. Fig. 2 shows the input data and output data of the FUDS drive cycle. Moving average is used for smoothing the data. The step of moving average is set to 100. This paper researches into the influence of increasing the number of the neurons in hidden layer on the error rate of the DNN model.

Another dataset used in this study is the BMW-i3 battery pack. The training data includes SOC values ranging from 90% to 10%, while the testing data includes SOC values ranging from 56.8% to 9.9%. Fig. 3and Fig. 4shows the current and voltage of the BMW-i3 batterydata.The difference between CALCE data and BMW-i3 data is that the former has a regular pattern in the current, while the latter does not. The current of the CALCE data ranges from -4A to 2A, whereas the current of the BMW-i3 data ranges from approximately -150A to 400A. Due to the current impulse, the step of moving average is set to 1024 for training the DNN model with BMW-i3 data. With this parameter setting, the DNN model can achieve an error of about 5%.

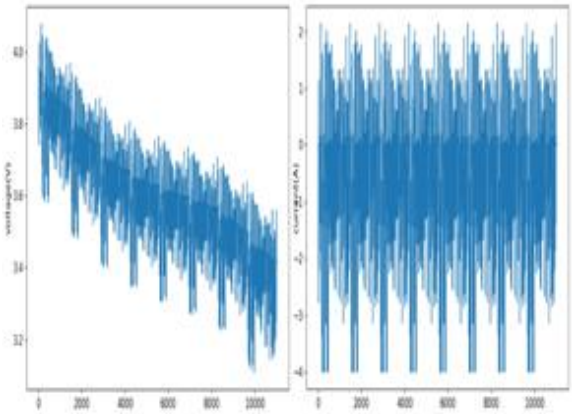


Fig. 2. FUDS drive cycle

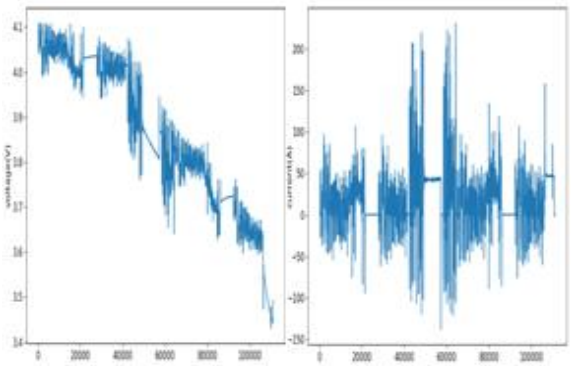


Fig. 3. BMW-i3 battery training data

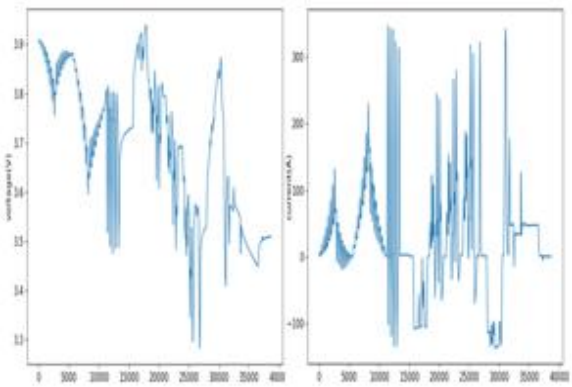


Fig. 4.BMW-i3 battery testing data

C. DNN Hyperparameters

The hyperparameters are the variables which determines the structure of the DNN model and the variables which determines the training process of the DNN model, such as the number of hidden layers, the number of neurons in every hidden layer, and the learning rate.The hyperparameters of the proposed DNN model is shown on Table. 1.

Number of Neuronsin hidden layer	Variable
Number of hidden layer	2
Optimizer	Adam
Learning rate	0.001
Batch size	1
Activation functionof hidden layer	ReLU
Activation functionof output layer	Sigmoid

Table 1. Hyperparameters of the DNN model.

D. Hardware Implementation of Sigmoid Function

The activation function sigmoid is shown below:

$$f(x) = 1/(1 + e^{-x}) \qquad (4)$$

It's obvious that the function is nonlinear. So, using several piecewise linear functions to approximate the sigmoid function for hard implementation is necessary.For hardware implementation, [5] shows an adaptive step-size-based slope and intercept method (AS-SIM) algorithm, which approximates the nonlinear function with a maximum error as input, and [6] explains the hardware structure of the sigmoid function.

IV. RESULTS

A. DNN Model Performance for SOC Testing

Fig. 5 shows the schematic diagram of the proposed DNN model. This paper utilized Python to train and test the DNN model for li-ion battery estimation. The battery data set is divided into training and testing subsets, 80% and 20%, respectively. The number of neurons in hidden layer is a variable, which ranges from 4 to 16. Fig. 6 and Table. 2 shows the simulation results and error rate of the proposed DNN model with 1000 epochs. This work use root mean squared error (RMSE) to evaluate the performance of the DNN model. RMSE is great useful when large errors are particularly undesired. Obviously, the DNN model of 8 neurons in hidden layer has the best performance on the SOC estimation. But the gap between the DNN model of 8 hidden neurons and other DNN models isn't huge. The maximum difference of RMSE between these DNN models is about 2.5%.

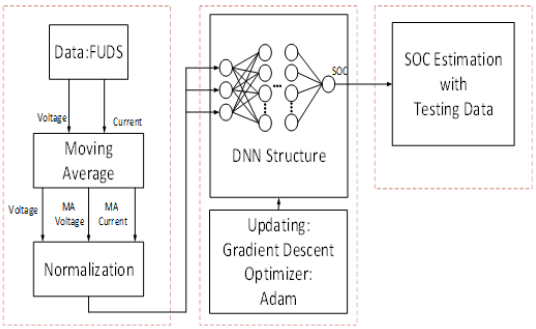


Fig. 5. The schematic diagram of the proposed DNN model

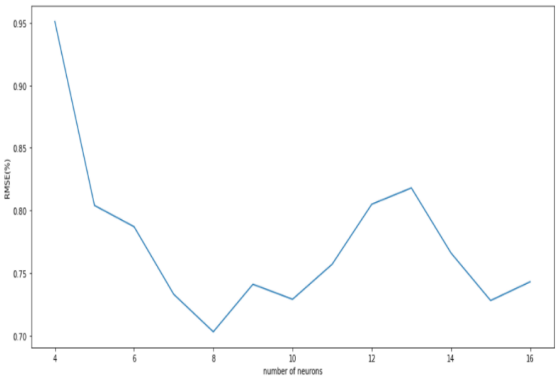


Fig. 6. RMSE for all the DNN model

Neurons	RMSE (%)	Error Range (%)	MAE (%)
4	0.951	[2.627, -4.511]	0.623
5	0.804	[3.472, -5.324]	0.592
6	0.787	[2.762, -3.313]	0.585
7	0.733	[2.427, -3.978]	0.516
8	0.703	[1.902, -3.781]	0.501
9	0.741	[3.101, -3.175]	0.546
10	0.729	[1.899, -3.940]	0.527
11	0.757	[2.508, -3.464]	0.545
12	0.805	[2.137, -3.532]	0.598
13	0.818	[3.154, -3.907]	0.589
14	0.766	[2.613, -3.394]	0.563
15	0.728	[3.489, -3.957]	0.536
16	0.743	[3.941, -3.256]	0.565

Table 2. Simulations and results of DNN models with different number of neurons in the hidden layer

B. Sigmoid Hardware Implementation and Simulations

According to [5], using AS-SIM to obtain the range and corresponding parameters of PWL function. The initial value of the AS-SIM algorithm is: $\varepsilon = 0.0005$, $x_{limit} = 8$.The result shows that it requires 12 PWL functions to approximate the sigmoid function.Fig. 7 showsthe sigmoid hardware architecture [6].

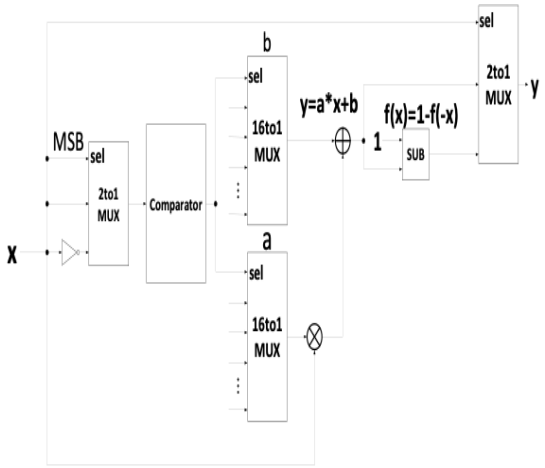


Fig. 7. The sigmoid hardware architecture

Due to the hardware implementation requirements, it is necessary to quantize the float numbers.There are some critical parameters that should be quantized in this hardware structure, including the range, parameter a, parameter b, output of the multiplier, and output of the adder.The final word lengths are as follows: a=(1,8), b=(1,8), range=(3,1), multiplier=(1,12), adder=(1,12). The number before the comma represents the integer, while the number after the comma represents the fraction. All the parameters is unsigned in this sigmoid hardware architecture.Because the sigmoid needs to be broken down into 12 PWLfunctions, the range should be at least 4 bits to achieve this. The simulation results are shown below:

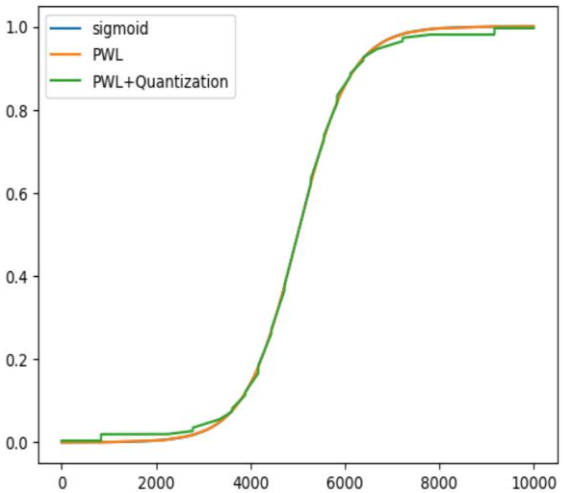


Fig. 8. Comparison of sigmoid, PWL, and PWL+quantization.

Because epsilon is set to 0.05%, curve “PWL” is almost the same as curve “sigmoid”, as shown in Fig. 8.The errors between curve “PWL+Quantization” and “sigmoid” are below 2%.

The overall DNN model testing results are shown below:

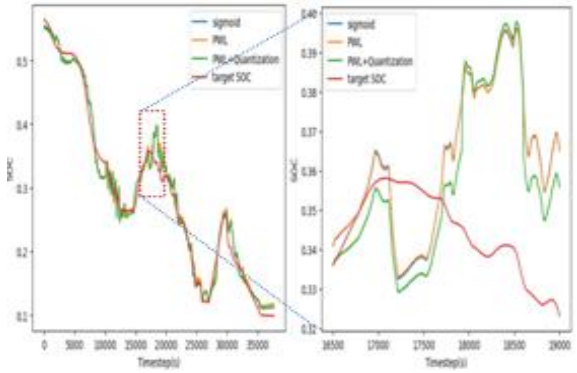


Fig. 9. DNN model testing results. (a)Data from 0 to 38000. (b) Data from 16500 to 19000.

	Error Range (%)	MSE (%)
sigmoid	[-5.782, 2.957]	0.592
PWL	[-5.823, -2.886]	0.623
PWL+ Quantization	[-5.972, -3.023]	0.605

Table 3. The mean square error (MSE) and the range of errors

Using BMW i3 battery pack data to train and test the DNN model. The SOC of the testing data ranges from 0.56% to 0.10%. Fig. 9 (a) shows three predicted curves under different conditions and one actual SOC curve. Fig. 9 (b) focus on the SOC which timestep is from 16500 to 19000,and the difference between the "sigmoid" and "PWL" curves can be observed. Table 3. Shows the mean square error (MSE) and the range of errors obtained from testing the DNN model.

V. CONCLUSIONS

This paper attempts to find a balance between the cost and performance of the DNN model. The proposed DNN model, consisting of 2 hidden layers with 8 neurons per hidden layer, is capable of predicting the SOC of batteries on both the FUDS drive cycle and the BMW-i3 dataset. In order to implement the testing part of the DNN model on the FPGA board, all functions in the DNN model must be quantized. Based on the experimental results, this paper has determined the appropriate word length required for the sigmoid function used in the FPGA implementation for predicting battery SOC.

REFERENCES

[1] RUI XIONG, (Senior Member, IEEE), JIAYI CAO, QUANQING YU, (Student Member, IEEE),HONGWEN HE, (Senior Member, IEEE), AND FENGCHUN SUNT. K. “Critical Review on the Battery State of Charge Estimation Methods for Electric Vehicles” IEEE Trans. Ind. Appl., vol.6, pp. 1832-1843, 2018.Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).

[2] Muh-Tian Shiue, Yang-Chieh Ou, Bing-Jun Liu, Yi-Fong Wang, Ping-Hao Liu” The Battery Measurement Approach of Lithium Battery in Real Driving Data based on Deep Learning Algorithms”, ACEP’S,2022

[3] D. N. T. How, Member, IEEE, M A Hannan, Senior Member, IEEE, M. S. Hossain Lipu, K. S. M. Sahari, Senior Member,

IEEE, P. J. Ker, Member, IEEE and K. M. Muttaqi, Senior Member, IEEE "State-of-Charge Estimation of Li-ion Battery in Electric Vehicles: A Deep Neural Network Approach," Citation information: DOI 10.1109/TIA.2020.3004294, IEEE .Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).

[4] M. A. Hannan, M. S. H. Lipu, A. Hussain, M. H. Saad and A. Ayob, "Neural Network Approach for Estimating State of Charge of Lithium-ion Battery Using Backtracking Search Algorithm", IEEE Access, 2018.

[5] S. R. Chiluveru, S. Chunarkar, M. Tripathy, and B. K. Kaushik, "Efficient hardware implementation of DNN-based speech enhancement algorithm with precise sigmoid activation function," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 68, no. 11, pp. 3461–3465, Nov. 2021.

[6] A. Tisan, S. Oniga, D. Mic and A. Buchman, "Digital implementation of the sigmoid function for FPGA circuits," ACTA TECHNICA NAPOCENSIS Electronics and Telecommunications, vol. 50, no. 2, pp. 15-20, 2009.

★★★