IJCTA, 9(10), 2016, pp. 4331-4338

Estimation of State of Charge of Lithiumion Battery Using Artificial Neural Networks

*** Muthumanikandan S **Shanmughasundaram R**

Abstract : Theusage of stored energy has increased rapidly. The main parameter of the battery is its State of Charge (SOC) which indicates how much charge the battery withholds. In this paper, the SOC of a Lithium-ion battery (Li-ion) is estimated using Artificial Neural Networks. The Back Propagation algorithm is used for training the network by controlling some of the training parameters. The complete system is simulated in SIMULINK for online estimation.

Keywords : Neural Networks, Back Propagation algorithm, State of Charge, Lithium ion battery

1. INTRODUCTION

The need for locomotion has increased. The fossil fuels are being depleted and is approaching extinction. Alternate source for powering the transportation is sought. The Electric Vehicle (EV) seems to be less emissive and eco-friendly. The main drawback of an electric vehicle is its inability to travel for longer distance and the batteries are capable of withstanding for a lesser amount of time. SOC is the parameter that denotes the amount of charge that is stored in the battery. So when the SOC is known, it is easy to predict the distance an EV can travellike a fuel meter in the conventional gasoline car.

Lithium-Ion batteries are used in Electric vehiclesbecause of theircharacteristics like higher energy density and power density. These batteries can be charged and discharged many number of timesbecause they do not have any memory effect like other type of batteries. The flat discharge curve of lithium-ion battery makes it difficult to estimate the SOC. In this paper, a single lithium cell of type 18650 is used for estimating the SOC.

The Artificial Neural Networks are a replica to the human brain where the system learns to predict the output by itself. A set of inputs and its corresponding outputs are given to the network. It will learn the relation between the data and will assume a function to the relation. After the learning is complete, the network provides the output for various inputs that are not given while training but comes in the range of the trained inputs. Neural Networks consists of set of neurons connected to each other through connections whose strength depends on the changeable synaptic weights.

In this system, the voltage and charge left in the battery are taken as the inputs for training the Neural Network. The output is the SOC of the battery.

2. RELATED WORK

The SOCof a battery can be estimated by different methods. In [1][2], various methods are described with their pros and cons. The SOC is easily estimated in Coulomb Counting method but the initial SOC is difficult to

* Department of Electrical and Electronics Engineering Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham Amrita University, India muthumanikandan9@gmail.com.

** Department of Electrical and Electronics Engineering Amrita School of Engineering, Coimbatore Amrita Vishwa Vidyapeetham Amrita University,India r_shanmughasundaram@cb.amrita.edu

estimate and it should be recalibrated frequently. In Voltage based method, the open-circuit voltage (OCV) curve is said to be directly proportional to the SOC curve. But OCV should be measured when the battery is at rest for a longer period of time and this method is inapplicable for online estimation. The Kalman filter method produces accurate output from inaccurate data even when the different battery parameters change abruptly. The problem with the Kalman filter is, it needs an accurate battery modeland the mathematical computations are complex. The Neural Network method is self-learning and adaptable. It is accurate in estimating SOC when the training data given to the network is precise. This does not need any initial SOC or a battery model. However, the accuracy of the neural network will change if the training data is wrong and incomplete.

So, this paper deals with the estimation of State of Charge using Neural Networks.

3. METHODOLOGY

Fig.1 shows the whole setup of the system. The Lithium-Ion battery is connected to a load. The voltage across the battery terminals is measured. The current drawn by the load is given to the Coulomb Counter circuit [5]. This circuit calculates the amount of charge left in the battery by multiplying and adding the time with the current drawn by the load. So the net charge left and the terminal voltage are given as inputs to the Neural Network. The output of the network is the SOC. The network is trained with the discharge characteristics of the Li-ion battery. The Feed Forward Neural Network is trained using Back Propagation algorithm (BPA) [3][4].

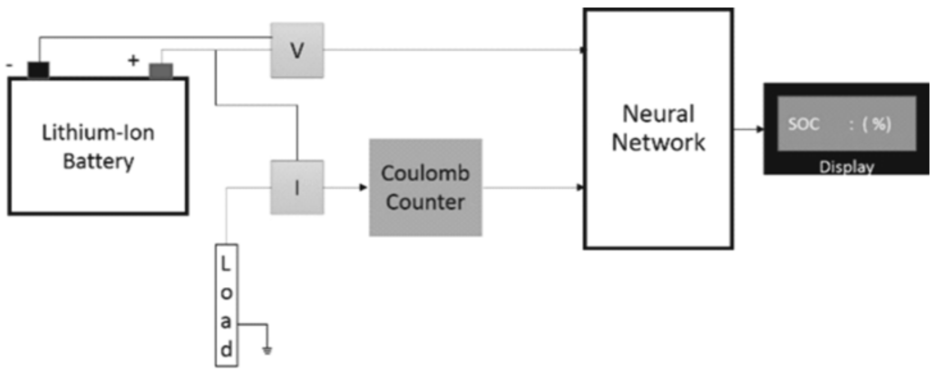

Fig. 1. Block Diagram.

A. Back Propagation algorithm

The Back Propagation Network (BPN) is used because it is a quick learner, robust and avoids over fitting. The network is capable of memorizing the relation between the input-outputs and tends to get generalized. Here, BPN has 3 layers namely input layer, hidden layer and output layer. The network is shown in Fig.2.

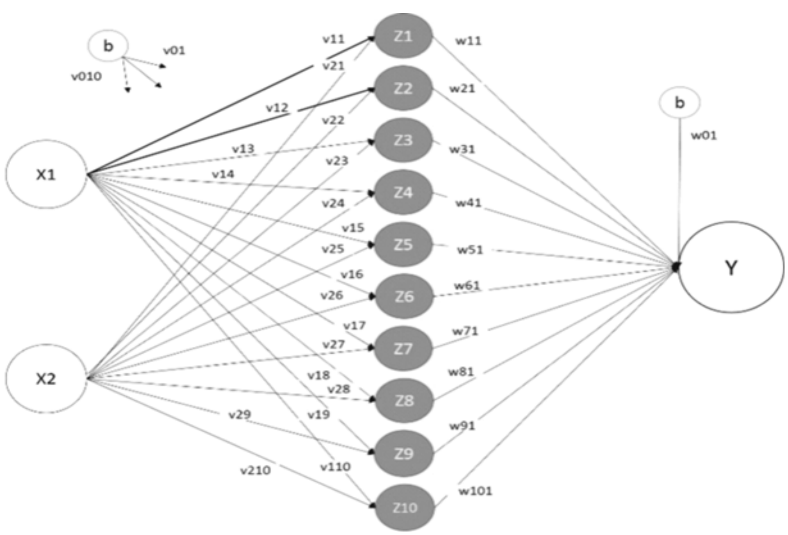

Fig. 2. Neural Network structure.

The training of the BPN is achieved in 3 phases :

1. **Phase I: Feed-Forward phase :** The input layer has two input neurons x_1 and x_2. These inputs are given to the hidden layer and the values of the hidden layer neurons are determined by the inputs, their corresponding weights and the biases. The net input to a hidden neuron j is given by,

$$Z_{inj} = voj + \sum_{i=1}^{2} x_1 v_{ij} \tag{1}$$

where v_{0j} is the bias on j^{th} hidden unit, x_i is the input from i^{th} neuron and v_{ij} is the weight of the connection between the i^{th} input neuron to j^{th} hidden neuron.

The output of each neuron is calculated by applying an activation function to the net inputs in all the layers.In hidden layer, the function used is a sigmoidal activation function. It is given in (2).

$$Zj = \frac{1-e^{-x}}{1+e^{-x}} \tag{2}$$

where x is the net input of the hidden neuron.

The net input to the output layer is given by,

$$y_{ink} = \text{Wok} + \sum_{j=1}^{10} z_j \; w_{jk} \tag{3}$$

wherew_{0k} is the bias on k^{th} output neuron, z_j is the input from j^{th} neuron and w_{jk} is the weight of the connection between the j^{th} hidden neuron to k^{th} output neuron.

The activation function used in this layer is a linear activation function. It is given in (4).

$$Y_k = f(y_{ink}) \tag{4}$$

2. **Phase II: Back-propagation of error :** The final output of the network $Y$is obtained. The error between the actual output and the target is to be calculated and it is called as error correction term (5).

$$\delta_k = (t_k - y_k) f' \; (y_{ink}) \tag{5}$$

wheret_k is the target value and $y_k$is the actual output of the output layer.$f'(y_{ink})$ is the first derivative of the net input to the output layer.

Based on the error correction term, the change in weights and bias are calculated as

$$\Delta w_{jk} = \alpha \delta_{kZj} \tag{6}$$

$$\Delta w_{0k} = \alpha \delta_k \tag{7}$$

The value of δ_k is sent backwards to the hidden layer. α is the learning rate of the network.

Likewise based on δ_k, the error in hidden layer is calculated as

$$\delta_{inj} = \sum_{k=1}^{10} \delta_{kwjk} \tag{8}$$

The error correction term is given by

$$\delta_j = \delta_{inj} \; f'(z_{inj}) \tag{9}$$

Now the change in weight and bias terms is given by

$$\Delta_{vij} = \alpha \delta_{jxi} \tag{10}$$

$$\Delta_{voj} = \alpha \delta_j \tag{11}$$

3. **Phase III: Weight and bias updation :** Each output neuron's weight and bias is updated by

$$w_{jk}\,(\text{new}) = w_{jk}\,(\text{old}) + \Delta_{wjk} \tag{12}$$

$$w_{0k}\,(\text{new}) = w_{0k}\,(\text{old}) + \Delta_{w0k} \tag{13}$$

Each hidden neuron's weight and bias is updated by

$$v_{ij}\,(\text{new}) = v_{ij}(\text{old}) + \Delta_{vij} \tag{14}$$

$$v_{0j}\,(\text{new}) = v_{0j}(\text{old}) + \Delta_{v0j} \tag{12}$$

These three phases are repeated until the target and the output values are equal or the error correction term is minimum.

4. IMPLEMENTATION

The specifications of the Lithium-Ion battery considered are,

Nominal voltage = 3.7V

Rated capacity = 2100mAh

Maximum voltage (full charge) = 4.3V

Minimum voltage (full discharge) = 3.5V

The battery is discharged at 0.5C of standard capacity at a temperature of 25^0 C. The discharge curve is shown in Fig.3.

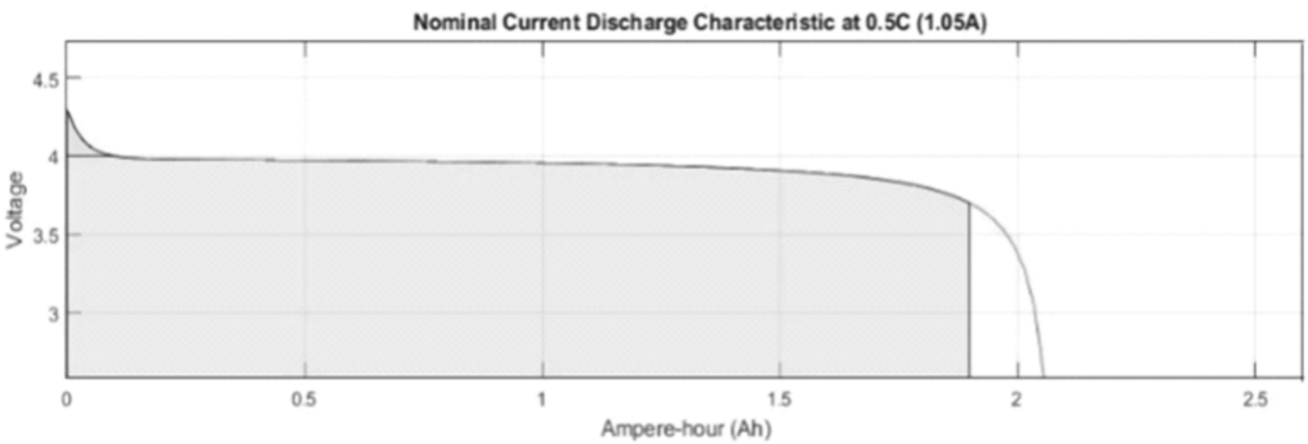

Fig. 3. Discharge curve at 0.5C.

The voltage and charge values are tabulated from the above graph. Sample values are shown in Table I.

Table 1 : Battery discharge values.

Voltage(V)	*Charge left(mAh)*
4.2976	2097.9
3.9807	1888.11
3.9758	1678.32
3.9705	1468.53
3.9633	1258.74

These tabulated voltage and charge values of the battery are given as two inputs x_1 and x_2 for training the Neural Network.

The Feed-Forward Neural Network with Back Propagation algorithm is implemented in MATLAB. The structure of the network is shown in Fig.2. The specifications of the network are given below:

The number of neurons in,

Input layer = 2

Hidden layer = 10

Output layer = 1 and the Learning rate is 0.1.

The Batch mode of training is adopted where all the inputs in the training data are given to the network before the weights are updated.

The BPN is created using command-line scripts. The Neural Network toolbox GUI is used to verify the trained network. Here 101 samples of two inputs are given for training. The data vectors are divided into three subsets namely training set, validation set and test set. These are divided into a ratio of 70:15:15 for facilitating the network to learn and generalize.

The training function used for BPN is Levenberg-Marquardt function. This training function performs better function fitting.

To obtain a generalized BPN, the following parameters are restricted to some constraints. The training process will stop if any of the following exceeds the given values.

$$\begin{aligned} \text{Epoch} &= 1000 \\ \text{Performance Goal} &= 1.00e^{-25} \\ \text{Validation checks} &= 6 \end{aligned}$$

Epoch is the number of times the entire set of training data vectors presented to the network. Iteration is the number of each individual training data presented. Performance Goal function used is the Mean Squared Error (MSE) function. The MSE should be nearer to the above mentioned value.Validation check is used to verify whether the network is trying to overfit. The network will start to train even after reaching the minimum MSE. So it will train for 6 iterationsand then stops.

5. RESULTS AND DISCUSSION

A. Performance

The MSE between the outputs of the training, validation, test data and the target data is plotted. The graph of the validation and test data should be similar. If the test data graph deviates before the validation graph then it is known thatoverfitting has occurred. In Fig.4, the error value is minimum, so the fit is good. The training is stopped at 1000 epoch and that value is taken as the best performance value.

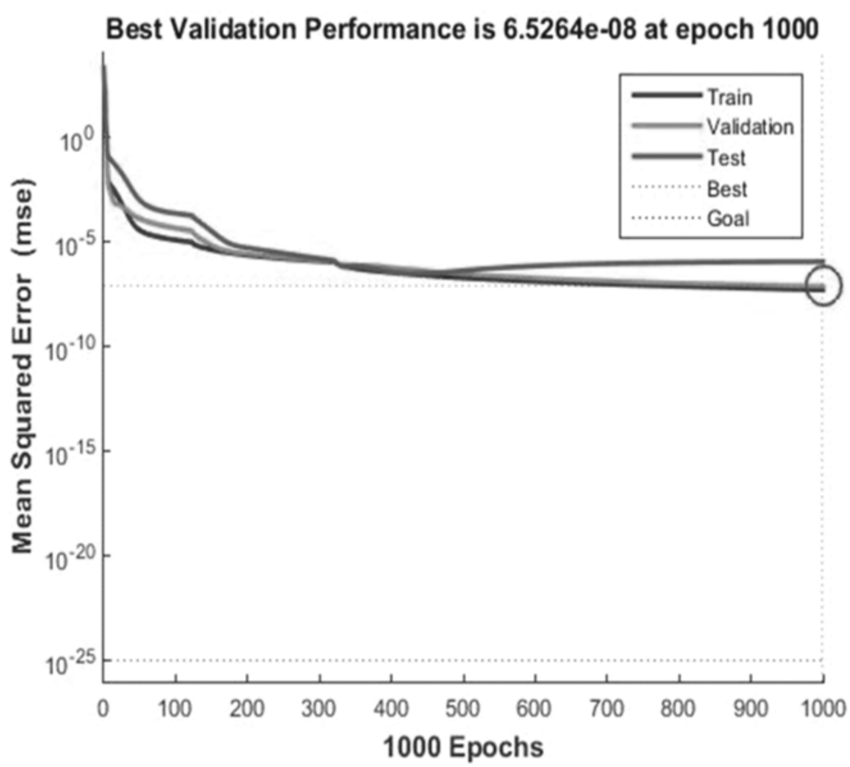

Fig. 4. BPN Performance graph.

B. Regression(R)

The regression plot shows the relationship between the outputs of the network and the corresponding targets for each of the three subsets. The R value is the indication of relationship between output and target. The value should be one or the graph inclination should be 45^0. In Fig.5, the R value is exactly 1. So the network is perfectly trained. R value should always be more than 0.95 for attaining a perfect fit.

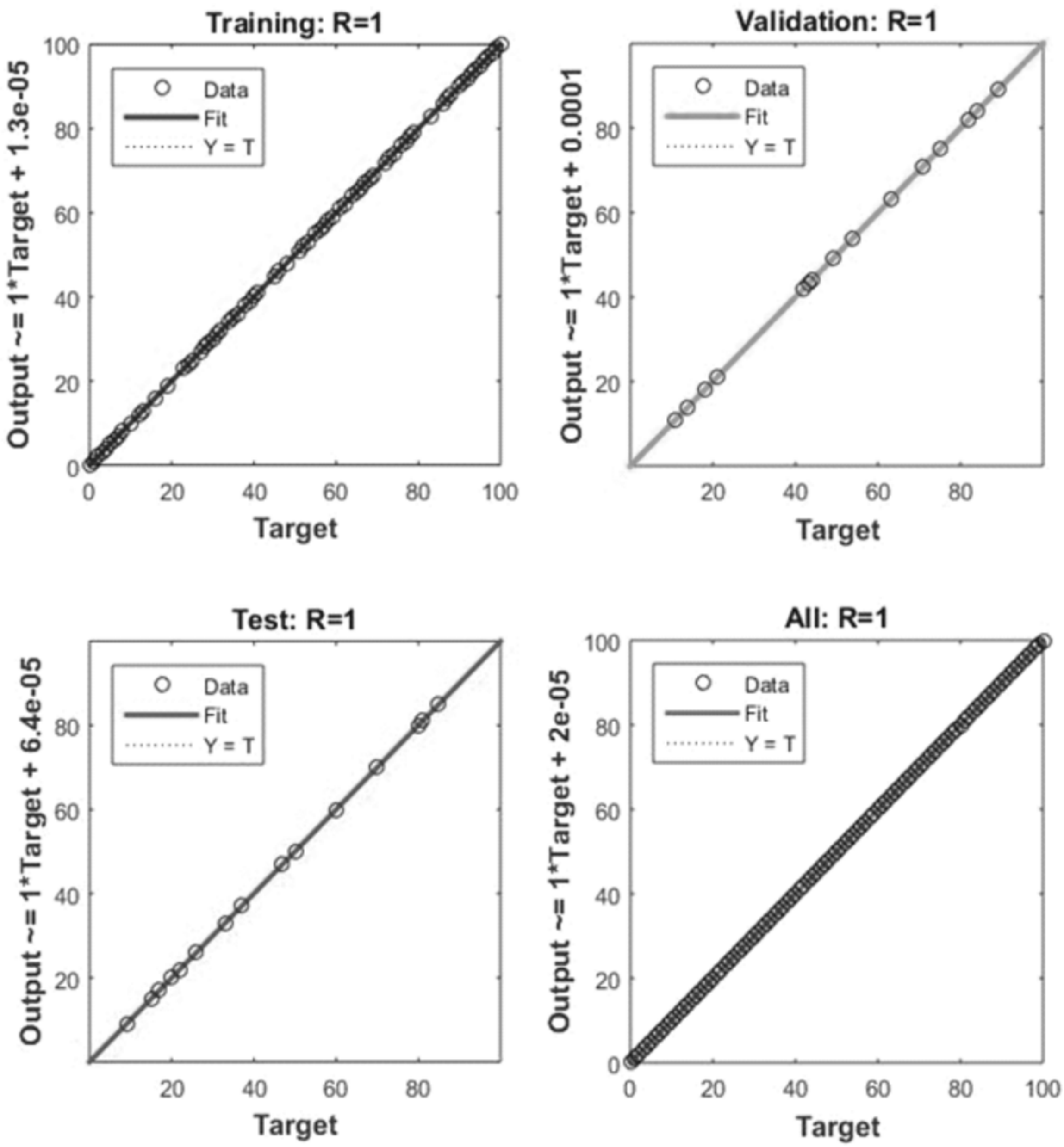

Fig. 5. BPN Regression graph

The trained Neural Network is then exported as a block to the SIMULINK. The type of the in-builtbattery model [6] is changed to Lithium-ion and the battery's parameters like open-circuit voltage, total charge and discharge rate are altered to 3.7V, 2100mAh and 0.5C.

A resistive load is connected to the battery. The battery terminal voltage is measured and is given as input x_1 to the Neural Network. The current drawn by the load is given to the Coulomb Counter circuit for calculating how much charge has been drained from the battery. It is done by multiplying the real time with the current drawn by the load and then accumulating all the multiplied values. This remaining charge value is given as input x_2. Now the trained network estimates the SOCof the battery.The whole setup is shown in Fig.6.

The in-built battery model will show the SOC, voltage and current drawn by the load. This actual SOC value is taken and compared with the estimated SOC of the Neural Network. The error percentage is calculated by,

$$\%\ \text{error} = \frac{|\text{calculated value} - \text{actual value}|}{|\text{actual value}|} \times 100\%$$

The error percentage between the target and the actual output is around 1%. This whole setup of trained Neural Network is tested for online estimation under varying load conditions.

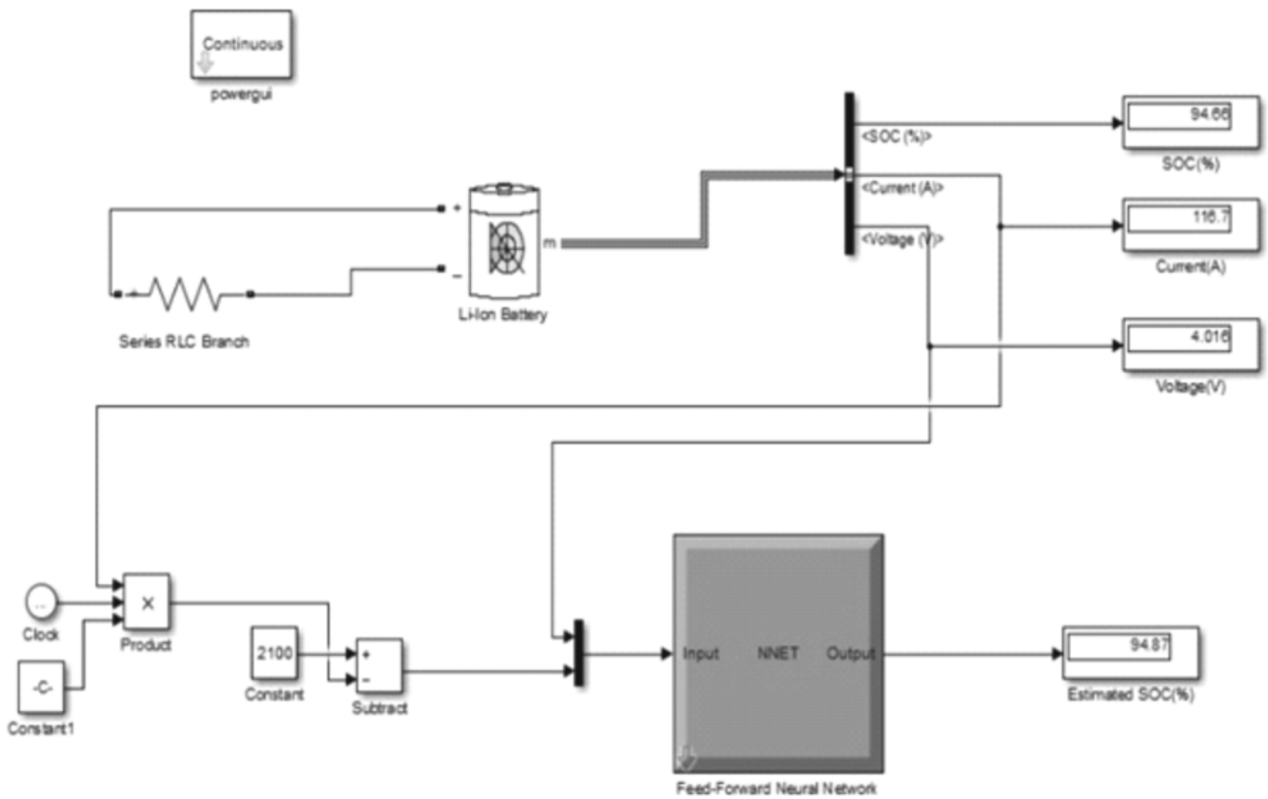

Fig. 6. Simulink model.

6. CONCLUSION

The Back Propagation Network is implemented with minimum error between the target and trained data by controlling different parameters. The SOC is difficult to estimate because of the flat discharge curve of lithium-ion battery. Here the Neural network has estimated precisely. Since the response of the system is quick, this algorithm can be implemented in an Electric vehicle for estimating the SOC to avoid the range anxiety. The temperature dependence of the battery is to be addressed when the Neural Network is implemented in hardware. The whole training and weight update process will take place in the microcontroller.

7. REFERENCES

1. Z. Chen, S. Qiu, M. A. Masrur and Y. L. Murphey, "Battery state of charge estimation based on a combined model of Extended Kalman filter and Neural Networks", International Joint Conference on Neural Networks(IJCNN), 2011, pp. 2156-2163.
2. S. Piller, M. Perrin, A. Jossen, "Methods for state-of-charge determination and their applications", Journal of Power Sources, vol.96, 2001, 113-120.
3. S. Grewal and D. A. Grant, "A novel technique for modelling the state of charge of lithium ion batteries using artificial neural networks", 23rd International conference on Telecommunications energy, INTELEC 2001, UK, pp. 174-179.
4. C. R. Chen, K. H. Huang and H. C. Teng, "The Estimation of the Capacity of Lead-Acid storage battery using Artificial Neural networks", 2006 IEEE International Conference on Systems, Man and Cybernetics, Taipei, 2006, pp. 1575-1579.

5. A. Ausswamaykin and B. Plangklang, “Design of real time management unit for power battery in PV-hybrid power supplies by application of Coulomb counting method”, International Electrical Engineering Congress(iEECON), 2014, pp. 1-4.

6. M. M. Ismail and M. A. M. Hassan, “The state of charge estimation for rechargeable batteries based on artificial neural network techniques”, International conference on Control, Decision and Information Technologies(CoDIT), 2013, pp. 733-739.

7. D. Pavkoviæ, A. Komljenoviæ, M. Hrgetiæ and M. Krznar, “Experimental characterization and development of a SoC/SoH estimator for a LiFePO4 battery cell”, IEEE International Conference on Computer as a Tool (EUROCON), 2015, pp. 1-6.

8. Michael A. Nielsen, “Neural Networks and Deep Learning”, Determination Press, 2015.

9. Shruthi.S, Supriya.P, “Modified State Variable Approach using Kalman filter for Battery Management of Electric Vehicle”, ICPCIT 2016.