State of Charge Estimation for Li-ion Battery

Koneru Saketh
Shaik Asif Ahmad
Vishwas Chepuri

Problem Statement

State Of Charge estimation is a fundamental challenge for battery use. The SOC of a battery, which is used to describe its remaining capacity, is a very important parameter for a control strategy. As the SOC is an important parameter, which reflects the battery performance, so accurate estimation of the SOC can not only protect battery, prevent over discharge, and improve the battery life but also allow the application to make rational control strategies to save energy.

What is State of Charge (SoC) ?

- It is defined as the ratio of the remaining charge in the battery and the maximum charge that can be delivered by the battery

$$\mathrm{SoC}\ (t) = Q(t) / Q\square$$

- There are various methods of estimating SoC and they are classified according to their methodology such as Direct Measurement, Book-keeping estimation, Adaptive systems, Hybrid methods.

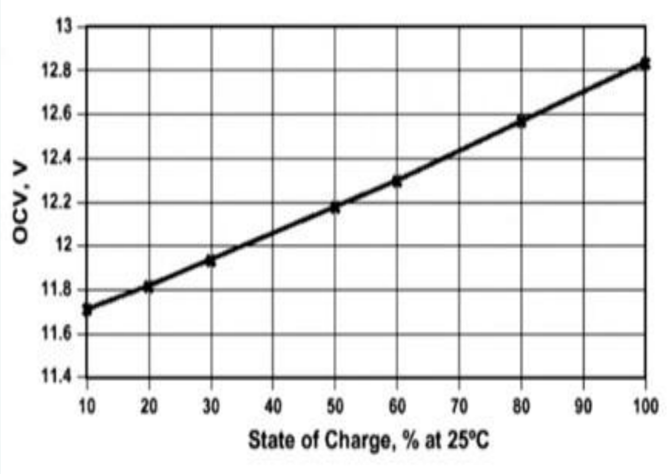
Voltage vs SoC Method

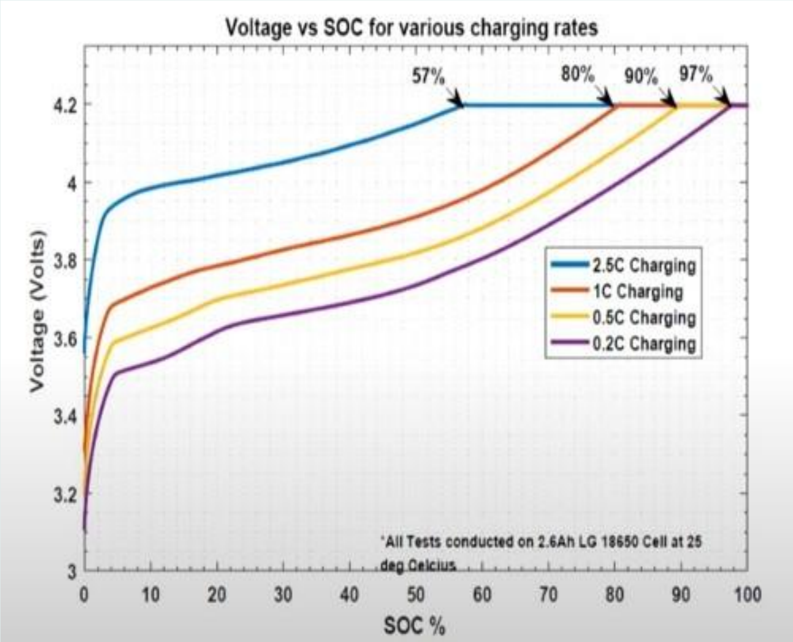
SOC estimation through Open Circuit Voltage vs SoC Graph

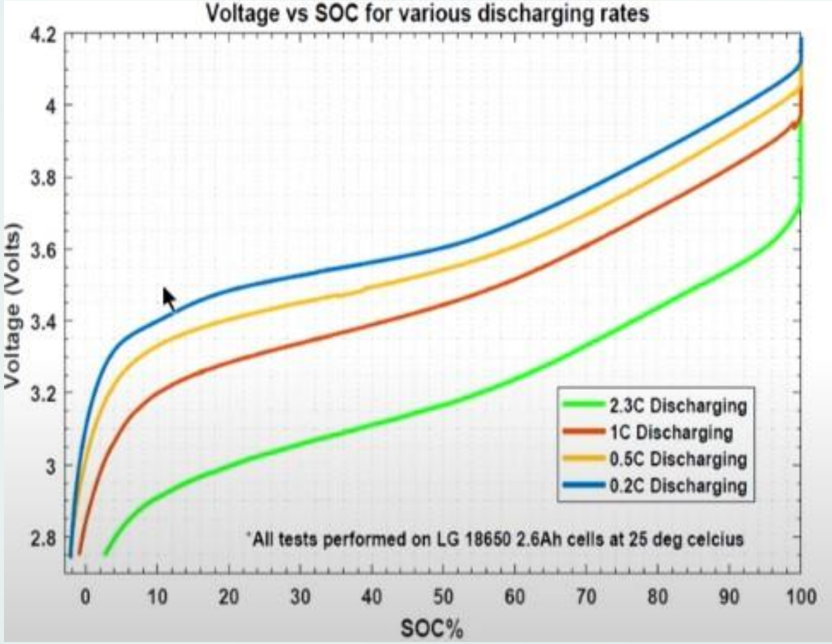
The value of the voltage obtained when measuring battery voltage in open circuit conditions used a reference for estimating SoC because Open circuit battery voltage is directly proportional to its SoC in Lead-Acid battery.

$$V_{OC}(t) = a_1 * SoC(t) + a_0$$

But in case of Li-ion battery, the proportionality does not hold. The voltage depends on many factors such as whether the battery is charging or discharging and the rate at what we are charging (0.2C, 0.5C, 1C, 2.5C), temperature etc.





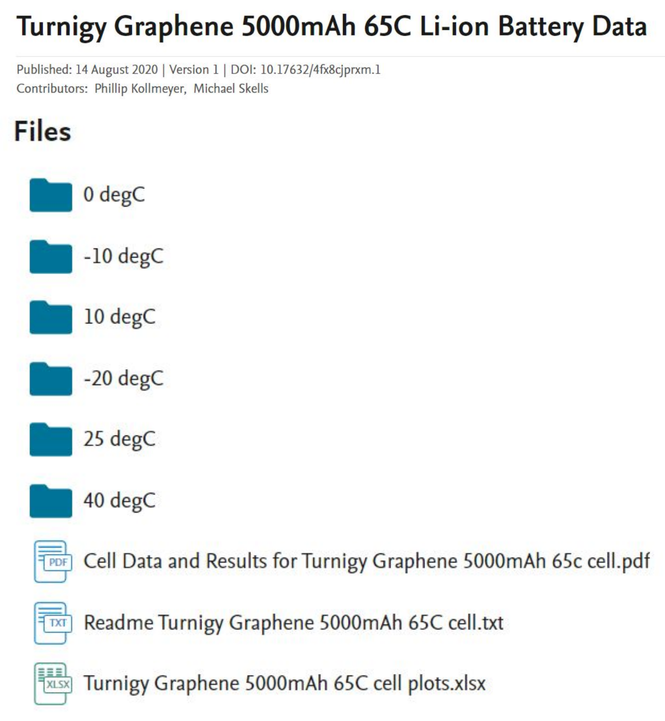


Due to its non linearity and dependence on lot many factors, It is very problematic to use Voltage vs SoC method for estimating the state of charge in case of Li-ion Batteries.

The methods that we worked on in this session are:

- Using Extended Kalman Filter
- Using Feedforward Neural Network

DATA SOURCE USED



Turnigy Graphene 5000mAh 65C Li-ion Battery Data

Published: 14 August 2020 | Version 1 | DOI: 10.17632/4fx8cjprxm.1
Contributors: Phillip Kollmeyer, Michael Skells

Files

- 0 degC
- -10 degC
- 10 degC
- -20 degC
- 25 degC
- 40 degC
- Cell Data and Results for Turnigy Graphene 5000mAh 65c cell.pdf
- Readme Turnigy Graphene 5000mAh 65C cell.txt
- Turnigy Graphene 5000mAh 65C cell plots.xlsx

- A new 5Ah Turnigy cell (Turnigy Graphene 5000mAh 65C cell) was tested in an 8 cu.ft. thermal chamber with a 75 amp, 5 volt Digatron Firing Circuits Universal Battery Tester channel with a voltage and current accuracy of 0.1% of full scale.
- 4-Drive Cycles were run at each of the 6 mentioned ambient temperatures.
- Data pertaining to each Drive Cycle consisted of Measured Temperature, Measured SOC values, Measured Terminal Voltage, Measured Current and Battery Parameters in the process of discharge.

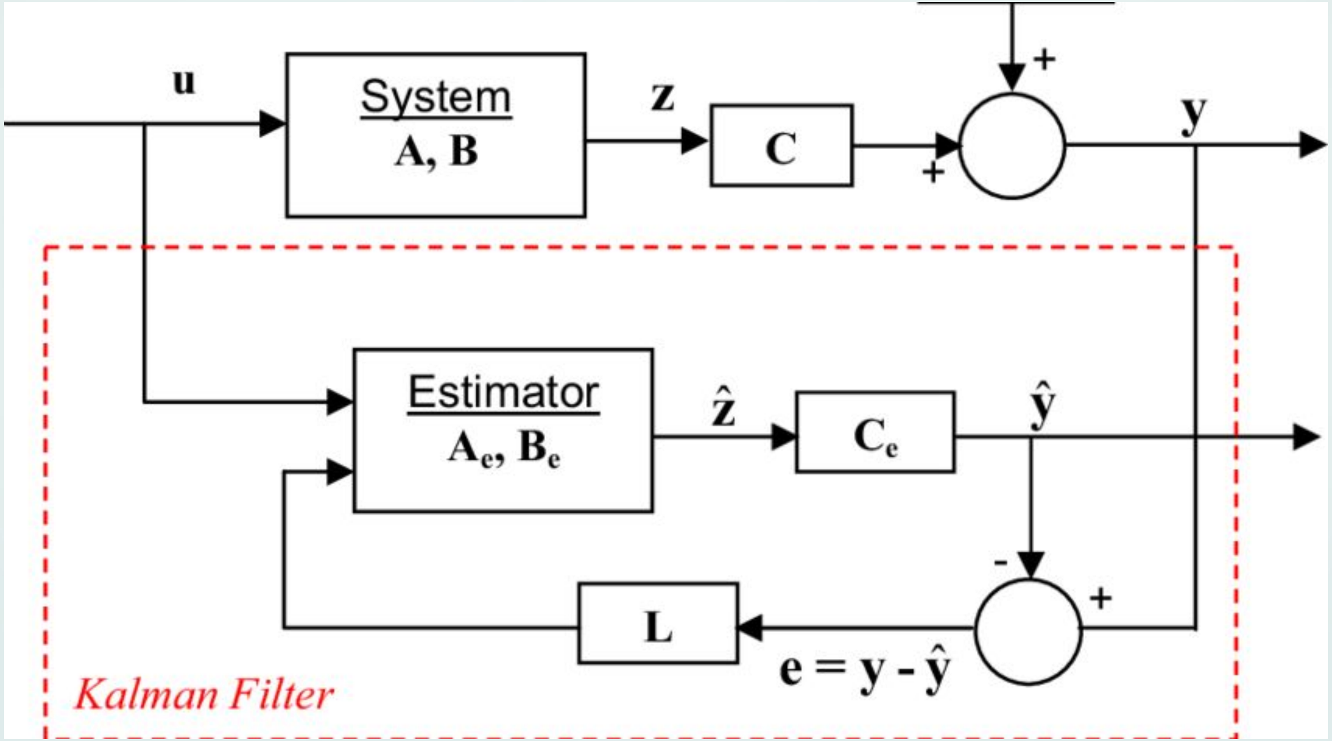
Extended Kalman Filter

SOC estimation through EKF

Kalman Filter

- Kalman Filtering is an algorithm that uses a series of measurements observed over time, including statistical noises and other inaccuracies, and produces estimates of unknown variables that tend to be more accurate than those based on a single measurement alone, by estimating a joint probability distribution over the variables for each timeframe.
- Essentially, by converging the error of estimated output w.r.t real output to zero, we converge the value of the estimated state variable to the actual value. This is just like a feedback control system.
- With the help of a controller, we can dictate the speed at which the error decreases.
- Assuming a Gaussian distribution of the probability density function of varied covariances for the estimated values of variables, we can use all of them to get a much more nuanced and accurate value of desired variables.

Block Diagram of Kalman Filter



Extended Kalman Filter (EKF)

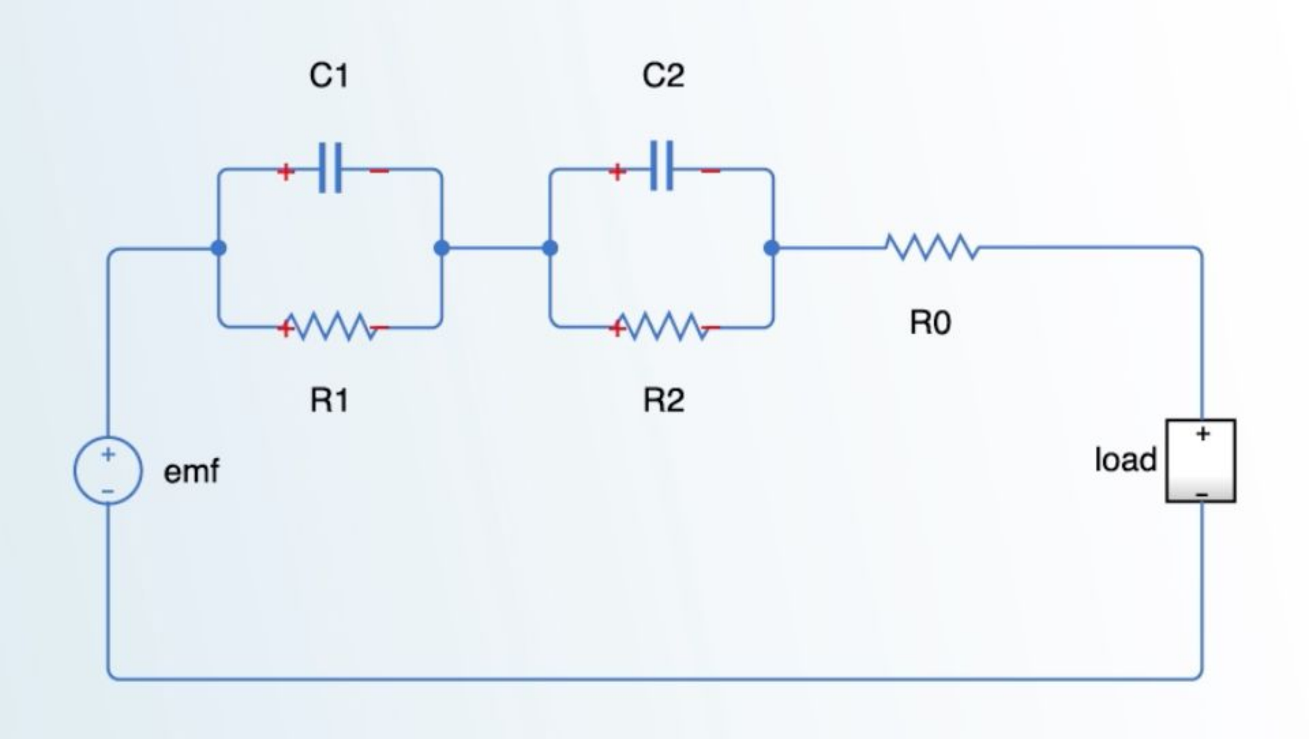
- An Extended Kalman Filter (EKF) is used when a system is non-linear but can be linearized about an estimate of the current mean and covariance.
- It is a two-step prediction-correction algorithm where first the variables are assigned a priori values through prediction and then are updated using corrections giving us the a posteriori values which will now be used in the next iteration.
- EKF has three variables P, Q and R which are:

  P = Covariance of a priori predicted state estimate

  Q = Covariance of Gaussian distribution of process noise

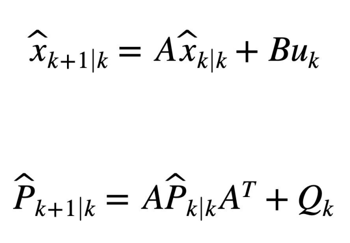
  R = Covariance of Gaussian distribution of measurement/output noise

Equivalent Circuit Model of Battery



Equations

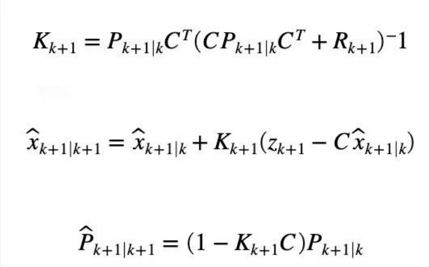
- Equations for prediction step(a priori values of state variable and error covariance):



$$\hat{x}_{k+1|k} = A\hat{x}_{k|k} + Bu_k$$

$$\hat{P}_{k+1|k} = A\hat{P}_{k|k}A^T + Q_k$$

- Equations for computing Kalman gain and updating the previous measurements:

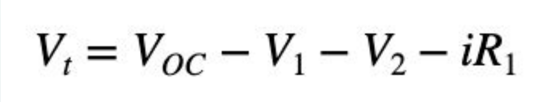


$$K_{k+1} = P_{k+1|k}C^T(CP_{k+1|k}C^T + R_{k+1})^{-1}$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K_{k+1}(z_{k+1} - C\hat{x}_{k+1|k})$$

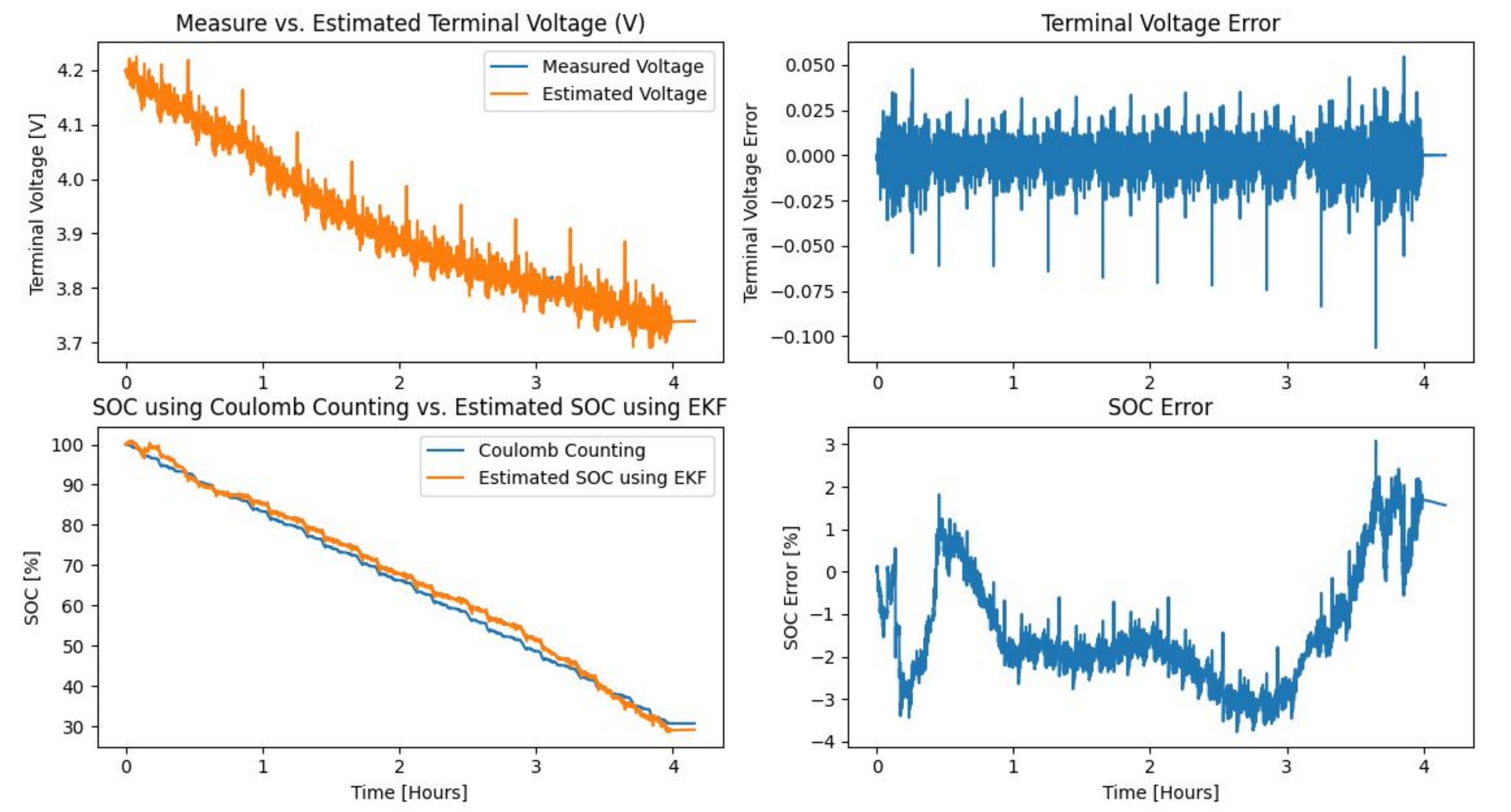
$$\hat{P}_{k+1|k+1} = (1 - K_{k+1}C)P_{k+1|k}$$

- Equation for Terminal Voltage :



$$V_t = V_{OC} - V_1 - V_2 - iR_1$$

EKF Predictions

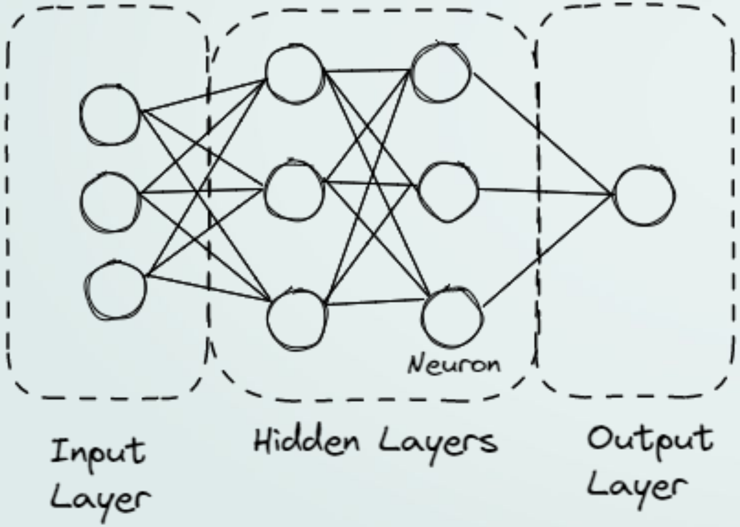


Feedforward Neural Networks

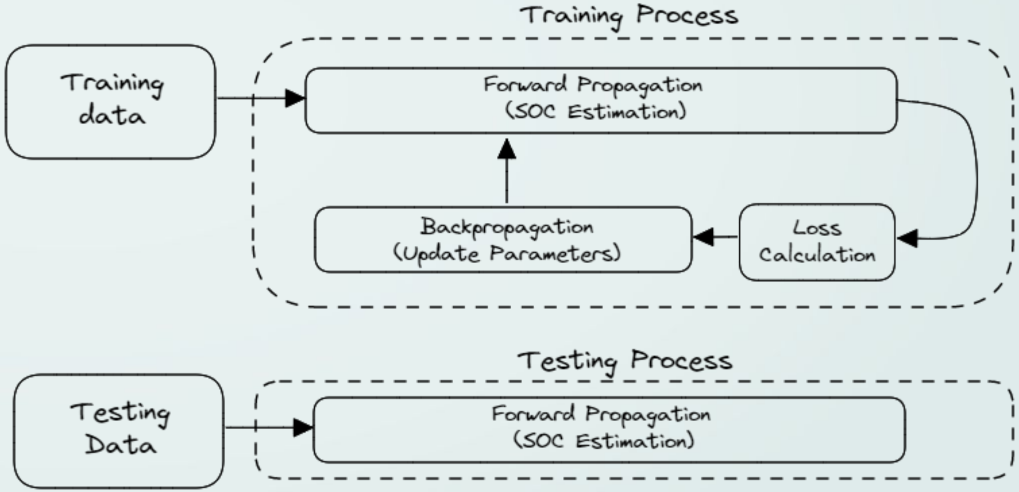
SOC estimation using FNNs

Overview of Training & Testing process

Simplified concept FNN structure for SOC Estimation



Neural Network training and testing/validation process overview


Training Process
Training data
Forward Propagation (SOC Estimation)
Backpropagation (Update Parameters)
Loss Calculation
Testing Process
Testing Data
Forward Propagation (SOC Estimation)

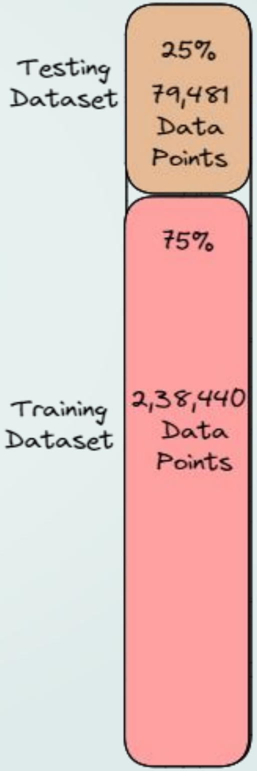
Data preparation

- We used following six measurements as inputs to the network.

  1) Measured Voltage
  2) Measured Current
  3) Measured Temperature
  4) SWA of Voltage
  5) SWA of Current
  6) SWA of Temperature

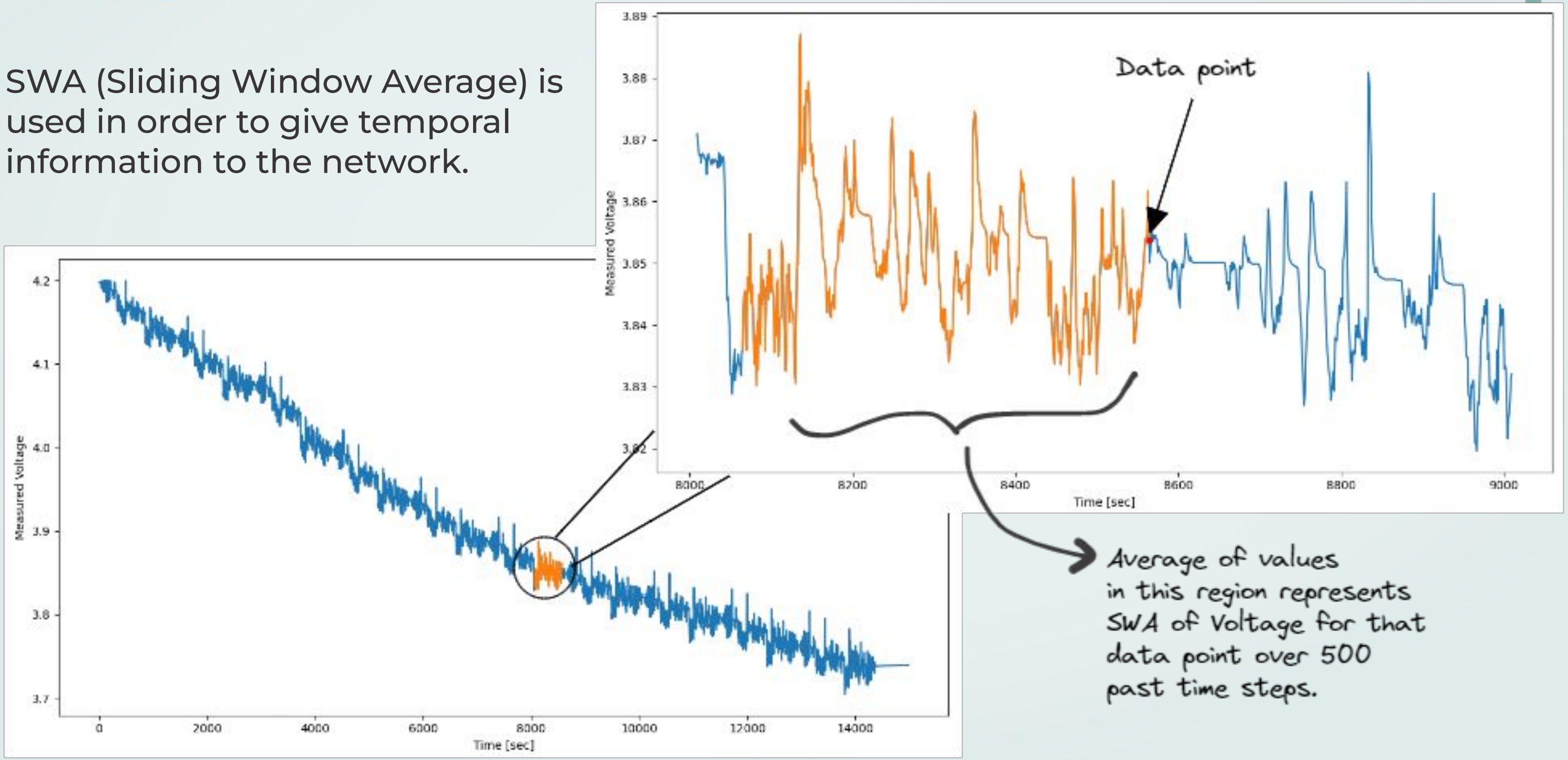
  Here SWA means sliding window average which was taken with a window size of 500sec.

- We normalize the SOC values before training so that values are between 0 and 1.



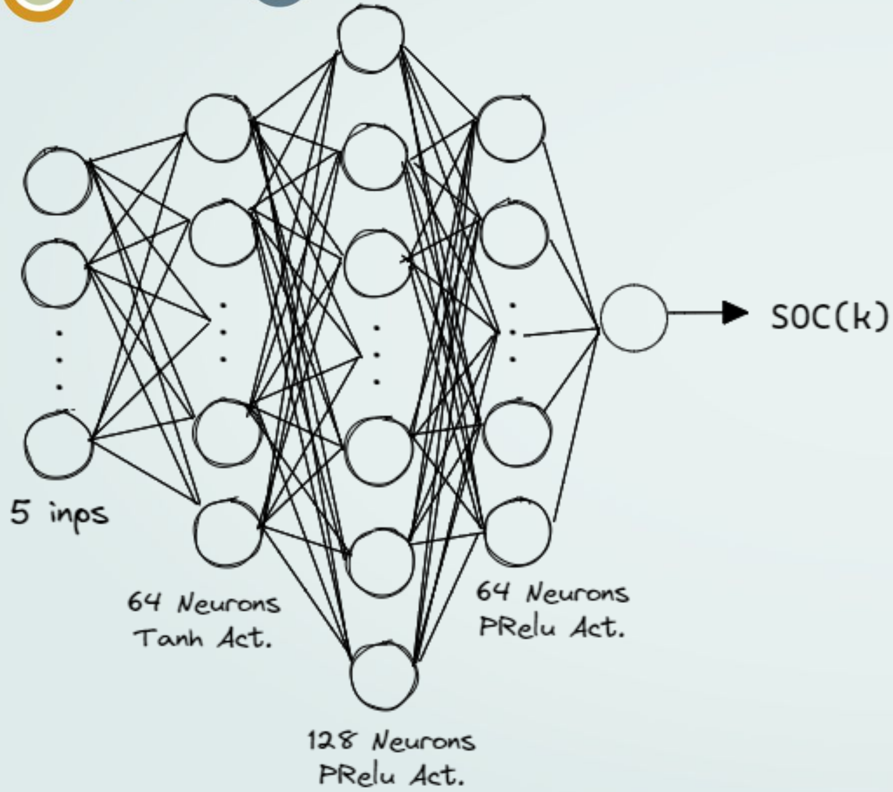
Data preparation

SWA (Sliding Window Average) is used in order to give temporal information to the network.



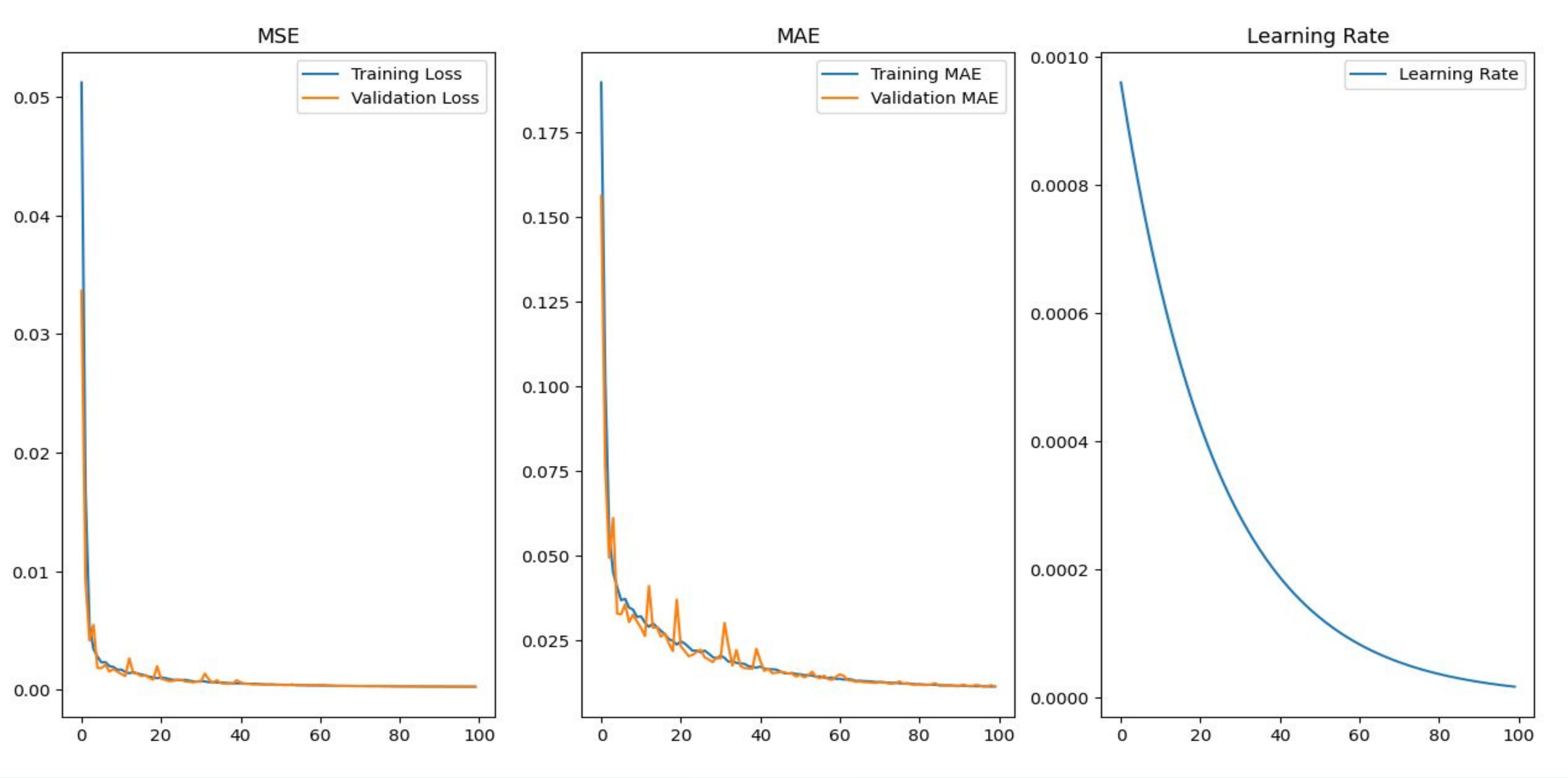
Average of values
in this region represents
SWA of Voltage for that
data point over 500
past time steps.

Network and training parameters

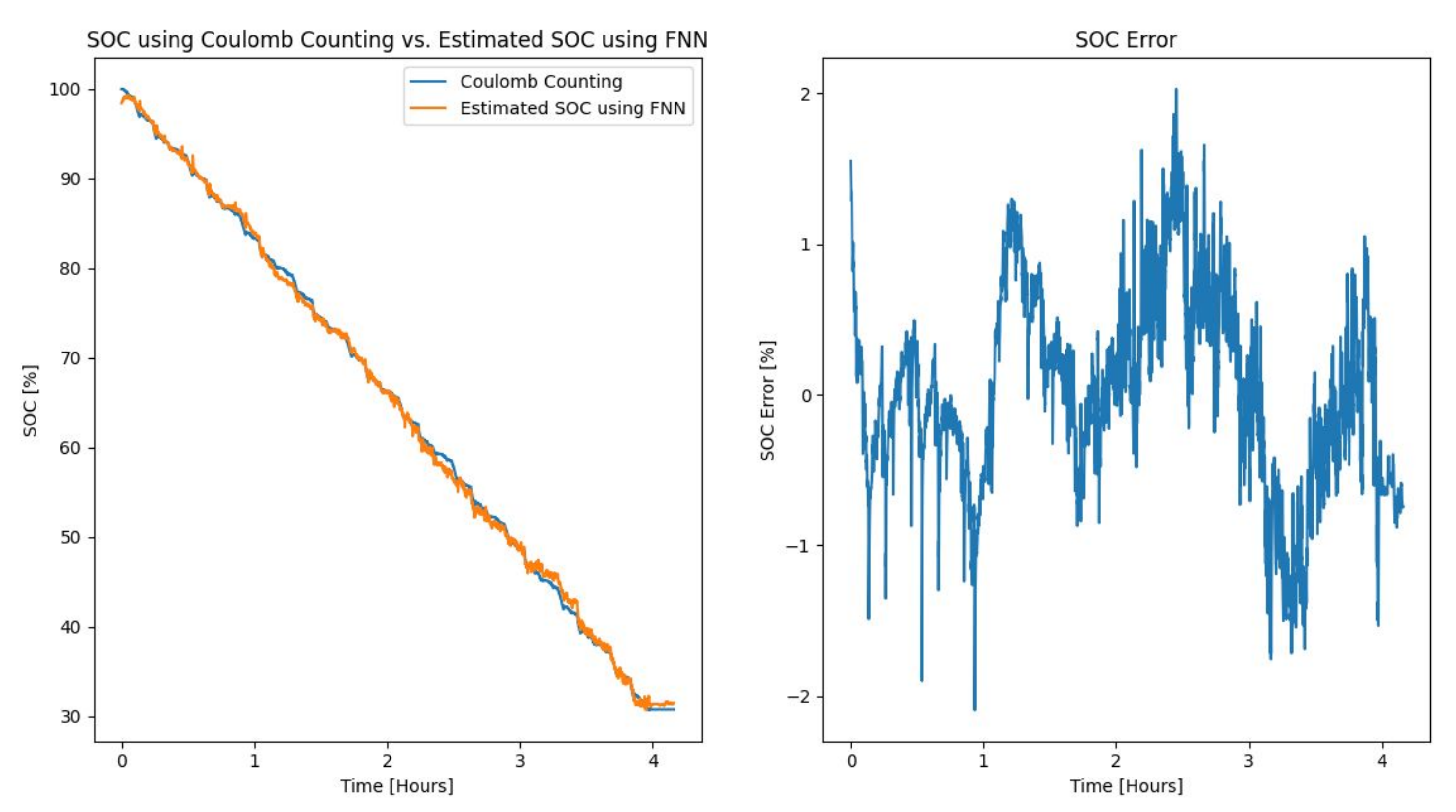


Number of hidden layers	3
Number of Neurons in each layer	[64, 128, 64]
Number of epochs	100
Initial learning rate	0.001
Learning rate drop factor	0.96
Optimizer	Adam
Batch size	1024
Loss	MSE

Training results



Predictions



THANK YOU

References

- How to Estimate Battery State of Charge using Deep Learning
- Overview of batteries State of Charge estimation methods
- https://data.mendeley.com/datasets/4fx8cjprxm/
- State of Charge Estimation Using Extended Kalman Filters for Battery Management System
- State of Charge Estimation based on Kalman Filter