

Machine Learning Engineer Nanodegree

Capstone Proposal

Gergo Kiss

January 25th, 2020

Domain Background

I will apply an alternative solution for battery state-of-charge estimation by using data driven nonlinear models to compare their predictions with a modelless approach as [coulomb counting](#). The battery state-of-charge (later SOC) is essential in battery management systems (later BMS). I develop software in a renewable energy source project where we measure batteries' voltage, temperature, SOC and some other stats. We are using a printed circuit board (later PCB) to measure the battery and control it by turning it on or off or in other words engaging it or bypassing it with the integrated electronics on the circuit board. The batteries can be connected through the PCBs thus the connected system can be used for EV car charging, peak load shaving on grids or simply as an energy buffer.

The batteries are individually controlled and measured which allows the user to set different DC voltage levels without DC/DC converters. Therefore, our technology is different than the usual BMS where the batteries only connected in series via power connection and only one measuring unit reads the overall voltage of the connected batteries. There are studies like "[Battery state of charge estimation using a load-classifying neural network](#)" and "[State-of-charge prediction of batteries and battery-supercapacitor hybrids using artificial neural networks](#)" which are probably detailing the effectiveness of neural networks in the field but I don't have access to these publications and I want to test our dataset with a tech specific solution.

Problem Statement

To measure a battery SOC, you must fully discharge it until it reaches its lower voltage limit. At this point of time the battery has 0% SOC. The charging process starts, and a current measurement unit tells the PCB how much current runs through the battery. The PCB uses the current value and duty time of the battery to estimate the SOC by calculating the capacity which is gained or lost inside the battery. This modelless method is coulomb counting. Since no measurement can be perfect, this method suffers from long-term drift and lack of a reference point. Therefore, the SOC must be re-calibrated on a regular basis, such as by resetting the SOC to 0% when a charger determines that the battery is fully discharged. I would like to test how a data driven non-linear model such as neural networks, decision trees, random forest or other models which can be applied for regression problem.

Datasets and Inputs

I will use two datasets. Both datasets are generated by our PCB measurements and by a laboratory power supply (during charging) or an electric load (during discharging). The first dataset includes 5 full charging cycles where only 1 battery is used. It is fully charged until it reaches the higher voltage cut-off limit and then it is fully discharged until it reaches the lower voltage cut-off limit (5 times).

The second dataset includes data about 5 partial charging cycles where the battery is part of a battery module which contains 24 batteries in total. In this scenario, the batteries are connected to maintain a constant voltage based on a topology selection algorithm. The topology of the engaged batteries is always changing during charge/discharge. Therefore, each battery spends time in turned off or turned on mode during the partial charging procedure. I will use the measurement data of only 1 cell to keep the dataset simplified for this project.

The first dataset called 'battery_data_1.csv' which includes approx. 165000 measurements. The second dataset called 'battery_data_2.csv' which includes approx. 107000 measurements.

The inputs will be the voltage, status, temperature, current, mode and time. The output is the SOC.

INPUT						OUTPUT
U _b	B _E	T	I _m	mode	time	SOC
Battery Voltage [V]	Battery Status Bypass = 0 Engage = 1	Battery Temperature [°C]	Current Measurement [A]	Charge = 1 Discharge = -1 Off = 0	Time between 2 meas. [s]	State-of-Charge 100 Ah = 100%

Table 1 – Description of the dataset

Battery limits:

Lower Cut-off Voltage	Higher Cut- off Voltage	Nominal Voltage	Charge current MAX	Discharge current MAX	Standard current charge/discharge
2.5 V	3.7 V	3.2 V	200 A	300 A	33 A

Table 2 – Battery Charging Specification

Both figures show the SOC output calculated by coulomb counting and the battery voltage U_b. As you can see on Figure 1 the SOC goes higher then 100%. It can happen because of two main reasons. Firstly, if the battery is new then its capacity can be higher than the nominal capacity which is 100 Ah here. Secondly the battery operates in a range of current levels during charge/discharge. Therefore, the energy loss during the charging procedure can vary and effect the battery capacity.

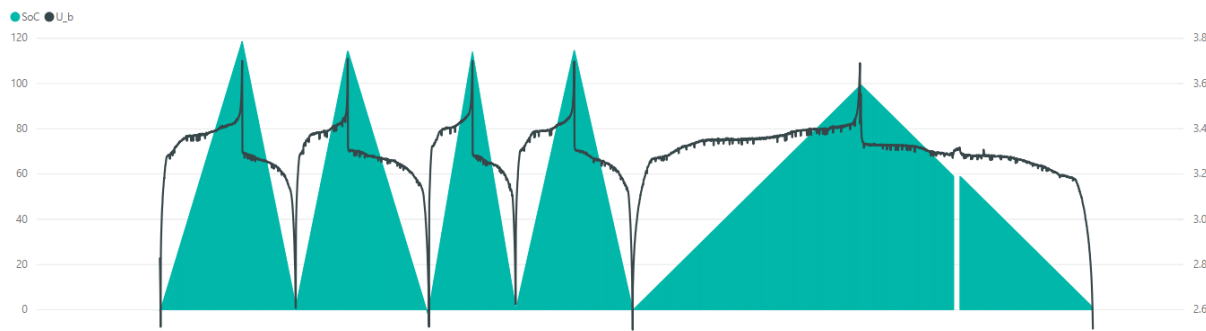


Figure 2 – [SOC, U_b] / runtime plot from 1st dataset, left vertical axis SOC, right vertical axis U_b

Figure 3 represents a different charging profile where the battery has rest time, so the 'time' input parameter either can be rest time if the status is 0 or duty time if the status is 1. In the first case the SOC should stay the same while in the second case SOC should increase or decrease based on the mode value.

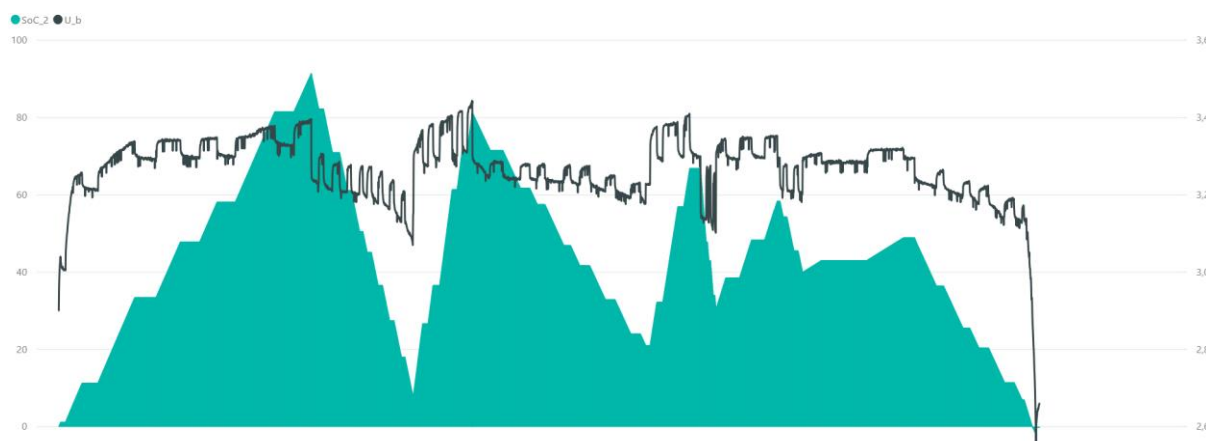


Figure 4 – [SOC, U_b] / runtime plot from 2nd dataset, left vertical axis SOC, right vertical axis U_b

I will experiment with these two datasets by possibly merging them or using one for training, the other one for testing and vice versa or just separately checking two different solutions for the two datasets. I would like to see which combination has better scores overall.

Solution Statement

The solution for this project is proposed to be a non-linear data driven model which results a SOC prediction algorithm with high accuracy rate and low error rate based on the given battery data. The algorithm will use the stated input values in Table 3 to predict the output value SOC. The solution will be quantifiable (the solution is expressed as the sum of discrete time measurements of current multiplied with delta duty time of the battery between two measurements), measurable (the solution can be measured by the capacity of the battery), and replicable (the solution can be reproduced and occurs more than once).

Benchmark Model

The values of the 'SOC' column in the datasets are the results of my benchmark model called coulomb counting. The nominal capacity is 100 Ah. Simplified coulomb counting looks like the following.

if mode == 1:

SOC += delta duty time [s] * measured current [A] / (3600 * nominal capacity)

if mode == 0:

SOC -= delta duty time [s] * measured current [A] / (3600 * nominal capacity)

Evaluation Metrics

For simplicity the benchmark model has 100% performance. Therefore, the solution model will be compared to the benchmark model SOC values. Metric is chosen to be the Error function in Equation 1 and the related accuracy in Equation 2. Since we are interested in how close the predicted value of the solution model to the actual value given by the benchmark model is.

$$Error = \frac{|SOC_{benchmark\ model} - SOC_{predicted}|}{SOC_{benchmark\ model}}$$

Equation 3 – Error function

$$Accuracy = 1 - Error$$

Equation 4 – Accuracy of a prediction

Project Design

1. Translate raw measurement data from 'battery_data_1.xls' and 'battery_data_2.xls' to '.csv' file format excluding all unused parameters. Also calculate the time difference between measurements to replace raw timestamps. If prep. is done, import the datasets into python.
2. Try out and implement few of the following options for the solution model:
 - a. neural networks,
 - b. decision trees,
 - c. random forest
 - d. or other models which can be applied for this regression problem.
3. Experiment with the chosen models by changing hyperparameters.
4. Train the selected model(s).
 - a. Try different episode numbers, batch sizes and other training parameters.
5. Evaluate the model performance by using the chosen metrics.
6. If necessary, select to best performing model and try to fine-tune it by making small adjustments.
7. Derive the conclusion.
8. Discuss future development potential.
9. Overall try to reach more than 90% accuracy if that seems to be not possible then discuss what can be missing from the dataset and include into future development.