Machine Learning Degree Capstone Proposal

I. Abstract

Rechargeable batteries are a very important component of several daily-use gadgets and EV automobiles too. Demand for Lithium batteries has grown tremendously in the last decade and is expected to continue growing in the foreseeable future. As a result, electronics suppliers and OEMs invest hugely on Battery Management Systems (BMS). Central to the BMS is the function of estimating the State of Charge (SOC) of a given battery. In my present role with my employer, I am working on a hardware solution for SOC estimation, and so to come up with a solution that is entirely different from a (expensive) dedicated hardware-based solution, a solution that leverages Machine Learning is of interest to my employer and me.

II. Problem Statement

It is imperative for the BMS to accurately estimate the battery capacity at any given point of time, under different operating conditions. Battery capacity, also known as State of Charge, is a measure of total energy stored in a battery. The SoC is related to the nominal rating of a battery. Factors such as amps extracted from (or supplied to) the battery and operating temperature create non-linearities in estimating the instantaneous SoC.

III. Dataset

I will generate the data relevant to one given battery using the above-mentioned hardware-based solution. Simulations will be run to gather Voltage, Current, Temperature and (the target) SoC measurements.

I expect to generate about 1000 datapoints using the benchmark (simulations).

IV. Solution Statement

The design of an ML regression model will be the solution that estimates the SoC value. I could choose from sklearn's regressors such as ensemble AdaBoost, but I prefer to try solving this by designing a Keras/TF CNN from scratch.

For the capstone, I will address one phase of the BMS/SoC problem – namely, making accurate prediction of the battery's "*first SoC*" – when a device is powered-on via a battery for the first time. The continuous estimation of SoC during normal operation of a device is critically dependent on accurate estimation of the first SoC. That is because, in the event the first SoC is inaccurate, continuous SoC estimation cannot converge to the correct SoC.

V. Benchmark model

The Benchmark model is one that estimates SoC using a hardware solution - one that is being used in a multitude of real world mobile devices (Samsung, LG, Xiaomi, et all) that are out in the market.

VI. Evaluation Metric

I will optimize the mean squared error loss function using one of few different optimizers. This metric (MSE loss) is a good indicator of performance of the CNN, given the regression problem statement.

VII. Project Design

The following workflow will, in my opinion, lead to a satisfactory culmination-

- Use Keras (TF backend)
- Either a fully connected MLP architecture or CNN
- With Dense arch-
    - I will experiment with a deep vs. wide network
- With CNN arch-
    - I will experiment with 2x2 or 3x3 filters, padding and strides.
    - Overfitting can be addressed by L2 Regularization or Normalization
- Since the data varies in different units, I will standardize.
- Split data into train, test and validation sets.
- Sklearn will be used – since it provides nice cross validation tools to evaluate the model.