Instructor: Laugelli
STS 1500

Provisional Patent Application for a Machine Learning Battery Optimization Software

Preamble

We, Charlotte Browder, Hunter Brown, Lily Burns, Sidhardh Burre, Keith Butler, and Josiah Calhoun, engineering students at the University of Virginia in Charlottesville, VA, have invented a method for maximizing the life and health of a smartphone battery in an iterative, unobtrusive mode by monitoring usage patterns.

Definition

This machine learning battery optimization software will record user charging habits, analyze the usage pattern through machine learning, generate a battery charging plan, and iterate upon itself to have the least effect on user usage patterns while maximizing the battery's longevity. The charging plan will ensure that the battery is charged optimally to minimize battery degradation.

Advantages over Prior Art

Batteries are currently consumed irresponsibly, resulting in immense amounts of physical and electronic waste. Poor charging habits such as using high watt chargers, leaving batteries to charge over extended periods of time, and running high-impact programs can negatively affect battery life by requiring greater amounts of power to reach the same charge as well as decreasing battery lifespan. These effects not only inconvenience the user, but also contribute to higher levels of chemical pollution due to increased disposal of batteries.

In the past, there have been battery optimization softwares that have attempted to maximize battery longevity and performance. In 2014, Joshua P. De Cesare and Gaurav Kapoor from Apple Inc. developed a software that limits the amount of charge passed into an iPhone through analysis of daily usage patterns (U.S. Patent No. 20140082384A1, 2014). Apple's charge-limiting optimization stops the iPhone's charge level at 80% and then allows it to fully recharge immediately prior to the user waking up. The problem with this software is that it only occurs overnight and does not monitor user-specific charging habits throughout the day. This misses out on potential optimizations that could be made.

In 2018, Microsoft Technology Licensing LLC developed a software that reads a user's manually-inputted schedule and optimizes the charging of the device based off that schedule (U.S. Patent No. 10061366B2, 2018). By analyzing the user's schedule, the software would determine a charging schedule for the device. The efficacy of this design is limited because a user's schedule is highly variable and cannot be encapsulated by written agendas alone. Furthermore, the requirement of manual input greatly inconveniences the user.

Our method of battery optimization is preferable over previous battery optimization softwares because it completes its function without user input and is designed for minimal interference with the user's lifestyle. Instead of manual input, our design uses machine learning to evaluate specific charging habits and then automatically implement battery-optimizing results to maximize the battery life of any given device. In addition to requiring minimal input, these optimizations occur at all times throughout the day.

Description of Form

The user interface of the battery optimization software consists of three main pages: the home page, the settings, and the statistics page.

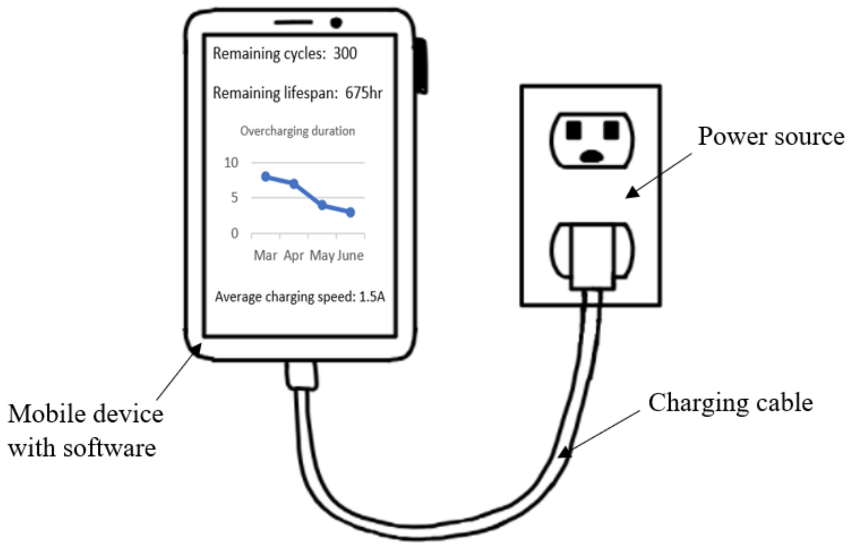

Figure 1: Overview of Charging Software with Device

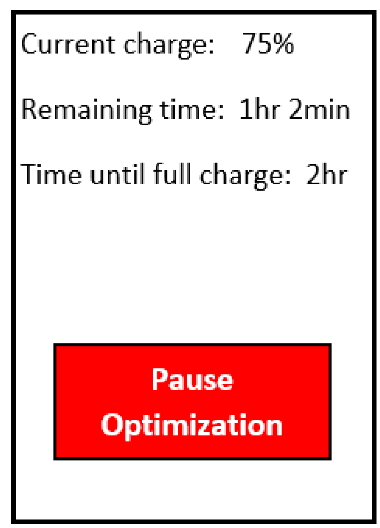

Figure 2: Example Home Page Interface

1. The *home page* is the software's central interface (see Figure 2). It is viewed directly upon opening the software application. The home page provides generalized information that allows the user to determine the predicted lifespan of their current battery cycle, as well as provide input to change the current charging plan. For example, the home page displays the expected time of use remaining with the phone's current charge, as well as the expected time to fully charge the user's phone. Additionally, the home page offers the ability for the user to override the current charging plan and charge their battery as quickly as possible.
2. The *settings* of the application allow for users to customize the functions of the software to their personal preferences (see Figure 3). Within the settings, there is an option to toggle the software's data collection on and off. The settings page also includes options to

reset the schedule data, input manual data, or name a specific usage pattern.

3. The *statistics page* is where the machine learning statistics that the software uses to make decisions are displayed (see Figure 4). The statistics page contains information gathered about the user and some of the performance benefits that the software achieves. This includes the predicted amount of charge cycles remaining on the phone's battery and the remaining duration of the battery's lifespan based on the user's past charging habits. Other statistics include, but are not limited to, improval in battery life since use of the software, applications with high energy usage that would affect the life of the battery, and average speed of charging.

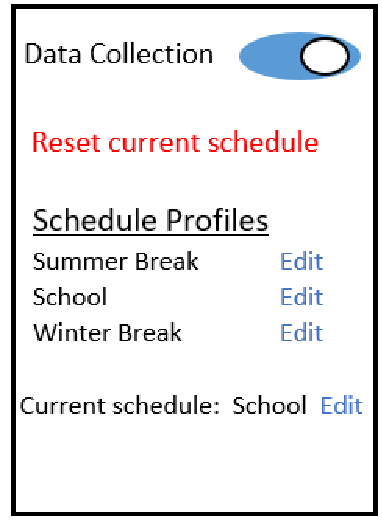

Figure 3: Example Settings Interface

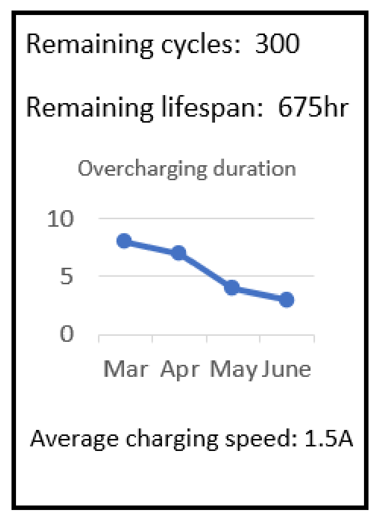

Figure 4: Example Statistics Page Interface

Description of Process

The battery optimization software's process consists of four main steps: The software measures the user's charging habits in a warm-up period, generates an initial battery charging plan, implements the battery charging plan in the user's device, and then updates the user's charging habits to make refinements to their battery charging plan.

The warm-up period begins immediately after the user downloads the software. During the warm-up period, the software measures several statistics related to the user's charging habits.

These statistics include number of charging sessions, average speed of charging, number of occurrences of extremely low charge, and average duration of overcharging. These statistics are tracked throughout the duration of the warm-up period, and are continuously updated in a nested class data structure, shown in Figure 5.

```
public class UserProfile {
    Map<Integer, ChargeStats> statsByHour;
    int totalChargeSessions;

    public void updateChargeSpeed(int hour, double percentCharged, double chargeDuration) {}

    public void updateChargeSessions(int hour, int numSessions) {}

    public void updateLowChargeCount(int hour, int numLowCharges) {}

    public void updateOverchargeDuration(int hour, double overchargeDuration) {}

    public void measureHabits() {/**Implementation varies based on OS*/}
}

class ChargeStats{
    double avgBatteryLevel;
    double totalChargeDuration;
    double totalPercentCharged;
    double avgChargeSpeed;
    int numChargeSessions;
    int lowChargeCount;
    double avgOverchargeDuration;
}
```

Figure 5: Class Data Structure for Storing User Habits

After a sufficient amount of data has been collected during the warm-up period, the software generates an initial charging plan. The initial charging plan consists of a recommended charging speed for each hour of the day based on the user's charging habits. If the user does not charge frequently during a given hour, the software assumes the user's schedule is busy and increases the recommended charging speed prior to that hour. Conversely, if the user does charge frequently during a given hour, the software decreases the recommended charging speed during that time to preserve battery health. The software also adjusts the recommended charging speed based on the user's previous incidence of low battery charge and overcharging during each given time frame.

Once the software has generated a charging plan, it then implements the charging plan. The implementation begins when the user plugs in their device to charge. Upon initiation of the charging session, the software immediately calculates the minimum charge value that will be necessary to sustain the user until the next predicted charging session in their schedule. Based on this value and previously measured data, the software makes a final adjustment to the recommended charging speed. The software then directs the user's device to charge at the recommended speed, taking into account hardware limitations. A basic outline of the code used to implement and update the charging plan is shown in Figure 6.

```
class ChargingPlan{
    Map<Integer, Double> chargingSpeedByHour;

    // Method for updating recommended charging speed based on user's stats
    public void adjustChargingSpeed(int hour, UserProfile user) {}

    // Method for determining current battery level of phone
    public double getCurrentCharge() {}

    // Method for calculating minimum charge needed to last till next session
    public double minChargeNeeded() {}

}
```

Figure 6: Outline of Code for Charging Plan

After implementing the charging plan, the software continues to update the user's charging statistics that were initially measured during the warm-up period. These updated statistics are used to generate a new, refined charging plan for the user's next charging session. Ultimately, as the software completes more charging cycles, it is expected that the charging plan becomes more effective and more tailored to the user's individual needs. A summary figure of this cyclical process is shown in Figure 7.

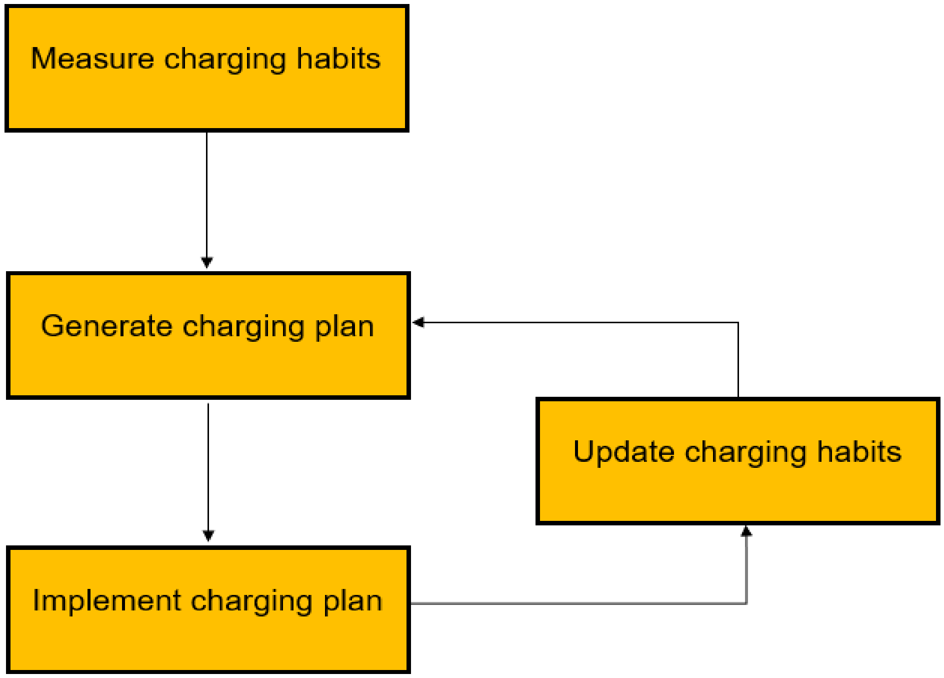

Figure 7: Flow Chart of Battery Optimization Algorithm

Alternatives

The inventors of the machine learning battery optimization software have alternative methods of producing their invention as described below:

- The software can be used on other batteries beyond smartphones such as, but not limited to, tablets, laptops, or smart watches.
- The software can be expanded to include other factors that affect battery life when determining optimal charging. This could include taking in information such as the screen brightness, if background apps are running, location services, and many other factors. It would use this information to further determine the best charging plan.
- The software can take in more user input, utilizing this as well as machine learning so that the user has more control to change the settings as they wish. For example, the user

could set their own charging schedule, inputting when they want their phone to be fully charged by.

- The software can be expanded to control other aspects of technological devices that affect charging. For example, to charge the device quicker it could lower the brightness or turn on airplane mode.
- The software could be replaced by a complete hardware-based implementation that circumvents the need for any software or code.

On our honor as students, we have neither given nor received aid on this assignment.

Keith Butler, Charlotte Browder, Josiah Calhoun, Lily Burns, Sidhardh Burre, Hunter Brown

References

Badam, A., Chandra, R., Priyantha, N.A., Dutra, J.A., Meinershagen, J.L., Hodges, S.E., & Moscibroda, T. (2015). *U.S. Patent No. 0061366.* Washington, DC: U.S. Patent and Trademark Office.

De Cesare, J.P., & Kapoor G. (2014). *U.S. Patent No. 0082384*. Washington, DC: U.S. Patent and Trademark Office.

Dear ladies and gentleman of the Jefferson Trust “Designing for a Sustainable World” selection committee,

We hope this letter finds you in good health. Through the class STS 1500, we, Charlotte Browder, Hunter Brown, Lily Burns, Sidhardh Burre, Keith Butler, and Josiah Calhoun, have designed a software that monitors a user's smartphone charging habits to improve battery life in a manner that is unobtrusive to the user. We request the award of the Jefferson Committee’s development grant due to the significant progress our software makes towards the United Nations’ Sustainable Development Goals (UN SDGs) regarding responsible resource consumption as well as its ability to be easily implemented among members of the University of Virginia with a mobile device.

Out of all the UN SDGs, our design most directly addresses Sustainable Development Goal 12 (United Nations Division for Sustainable Development Goals, n.d.a). According to the UN SDG website, Goal 12 is to “ensure sustainable consumption and production patterns” (United Nations Division for Sustainable Development Goals, n.d.b). Our design specifically addresses targets 12.1, 12.2, and 12.4 which focus on responsible resource utilization throughout the lifecycle of the resource. By implementing charging patterns that minimize the occurrence of overcharging and extreme discharging, our software will help batteries last longer and hold charge better. This, in turn, will reduce chemical and electronic waste resulting from battery disposal, encourage greater reuse of existing batteries, and improve consumer experience. In addition to Goal 12, our design also addresses UN Goals 7 and 11. Our design increases the affordability of energy by improving reusability of batteries, relating to Goal 7. Additionally, it helps society become more sustainable as it enables grid operators to plan for energy consumption and avoid over or under production, addressing Goal 11.

Specifically at UVA, this innovation will provide great effect. Throughout the University, there are nearly 43,000 faculty and students combined, indicating that there are on the order of 100,000 mobile devices that rely on the grid for power (University of Virginia, n.d.). Many of these mobile devices utilize batteries that are sourced unsustainably (Katwala, 2018). Furthermore, users of these devices often follow inefficient charging regimens that worsen the lifespan of the battery, contributing to increased rates of disposal (Digi-key, 2016). Because our software takes the form of a free mobile application, it can be easily downloaded by all mobile device owners at the university. Furthermore, since the device uses machine learning to optimize charging in the background, it can be utilized to improve battery life with very little user input and overhead. By improving the life of batteries across the UVA campus, the demand and turnover rate of batteries will be lessened, addressing the points of unsustainable sourcing and increased disposal. Moreover, the University can avoid expending large amounts of electricity to charge batteries that leak electricity, improving the efficiency and the effectiveness of the grid system.

Our design would seamlessly integrate into members of the UVA community’s lives, and it would lead the University towards a more sustainable future in line with the UN SDGs. We would like to sincerely thank the members of the Jefferson Trust “Designing for a Sustainable World” selection committee for taking the time to consider our design and for the opportunity to contribute toward the noble cause of propelling UVA to attaining the UNSDGs.

Sincerely,

Charlotte Browder, Hunter Brown, Lily Burns, Sidhardh Burre, Keith Butler, and Josiah Calhoun

References

Digi-Key. (2016 September, 1). *A Designer's Guide to Lithium (Li-ion) Battery Charging.* Digi-key. https://www.digikey.com/en/articles/a-designer-guide-fast-lithium-ion-battery-charging

Katwala, A. (2018 May, 8). *The Spiralling Environmental Cost of our Lithium Battery Addiction*. Wired. https://www.wired.co.uk/article/lithium-batteries-environment-impact

United Nations Division for Sustainable Development Goals. (n.d.b). *Goal 12.* United Nations. https://sdgs.un.org/goals/goal12

United Nations Division for Sustainable Development Goals. (n.d.a). *The 17 Goals.* United Nations. https://sdgs.un.org/goals

University of Virginia. (n.d.). *Facts and Figures.* University of Virginia. https://www.virginia.edu/facts