REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH
وزارة التعليم العالي والبحث العلمي

HIGHER SCHOOL IN APPLIED SCIENCES
–T L E M C E N–
المدرسة العليا في العلوم التطبيقية
- تلمسان -

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière: Électrotechnique
Spécialité: Énergie et environnement

Présenté par : Hadj Abdelkader GUERMOUDI

Thème

Comparative Study of SOC Estimation Methods Based on Artificial Intelligence for Lithium Batteries

Soutenu publiquement, le 26/06/2023 , devant le jury composé de:

M. A. TAHOUR	Professeur	ESSA. Tlemcen	Président
M. A. F. KERBOUA	MCA	ESSA. Tlemcen	Directeur de mémoire
M. F. BOUKLI HACEN	Professeur	ESSA. Tlemcen	Co- Directeur de mémoire
M. A. CHELLAL	Doctorant	IPB. Portugal	Co- Directeur de mémoire
M. A. K. CHEMIDI	MCA	ESSA. Tlemcen	Examinateur 01
M. S. BELAROUCI	MCB	ESSA. Tlemcen	Examinateur 02

Année universitaire : 2022 /2023

بسم الله الرحمن الرحيم

Étude comparative de l'estimation de l'état de charge des accumulateurs au lithium basée sur l'IA

École Supèrieure en Sciences Appliquées de Tlemcen

Ingénieur en Electrotechnique - Énergie et Environnement

M. GUERMOUDI Hadj Abdelkader

2022-2023

To my dear parents, for their love, their sacrifices and their support in the most difficult moments, which are at the origin of our success, may GOD keep and protect them.

To my dear brothers, for their constant encouragement and all the help they give me on a daily basis.

To all the people who, from near or far, took part in this work.

I dedicate this humble work to them

Acknowledgements

First of all, I would like to thank Almighty and Merciful Allah, who has given me the strength and patience to accomplish this modest work.

The first person I would like to thank most sincerely is my supervisor ***Pr. A. KERBOUA***, Senior Lecturer at ESSA Tlemcen, for all his efforts, his consistent help and the trust he places in me on a daily basis, without which this work would not have been possible. He was always there to give me good advice.

I would also like to express my gratitude to Professor ***Pr. F. BOUKLI HACEN*** for his invaluable contribution to this work. His innovative ideas and dedication have greatly enriched my project. Working with him was a real pleasure, and I'm grateful for his close collaboration and support throughout the process. His presence and commitment were an undeniable source of motivation.

I would also like to express my gratitude to ***Pr. A. CHELAL*** and ***Pr. L. MERAD*** for their invaluable contribution to this work. Their in-depth knowledge and expertise were of great added value to my project. Their support and sound advice have greatly contributed to its success. I am grateful to have had the opportunity to collaborate with such competent and committed professionals.

I would also like to thank ***Pr. A. Tahour***, Professor at ESSA Tlemcen, for agreeing to chair the jury.

My sincere thanks also go to the members of the jury ***Pr. A.K. Chemidi***, Senior Lecturer at ESSA Tlemcen and ***Pr. S. Belarouci***, Assistant Lecturer at ESSA Tlemcen for their interest in my study and for agreeing to examine my work and enrich it with their proposals.

Résumé

La technologie des batteries joue un rôle clé dans le développement des voitures électriques et des robots mobiles. La gestion efficace des batteries, en particulier l'estimation précise de l'état de charge (SOC), est d'une importance capitale tant pour la recherche théorique que pour les applications pratiques. Cependant, l'estimation de l'état de charge des batteries est une tâche difficile en raison de leurs caractéristiques non linéaires et variables dans le temps.

Dans cette étude, nous nous penchons sur la nature complexe des piles au lithium et explorons le potentiel des réseaux neuronaux pour la prédiction de l'état des piles. Plus précisément, nous comparons trois architectures de réseaux neuronaux différentes en utilisant l'ensemble de données LG HG2, qui fournit des informations précieuses sur le comportement des cellules au lithium.

En analysant la théorie des réseaux neuronaux et en menant des expériences complètes, nous visons à améliorer la précision de l'estimation du SOC. Nous nous attachons à comprendre la relation complexe entre les variables d'entrée et le SOC, et à exploiter la puissance des réseaux neuronaux pour capturer et modéliser efficacement cette relation.

Les résultats de notre étude contribuent non seulement au domaine des systèmes de gestion des batteries, mais éclairent également la compréhension plus large de l'estimation de l'état de charge dans le contexte des piles au lithium. En évaluant les performances de différentes architectures de réseaux neuronaux et en analysant les résultats obtenus, nous obtenons des informations précieuses sur l'optimisation des performances des batteries et l'amélioration de l'efficacité globale des véhicules électriques et des robots mobiles.

Mots-Clés: Estimation de l'état de charge, batterie au lithium, intelligence artificielle, réseaux neuronaux profonds.

الملخص

تلعب تقنية البطارية عاملا رئيسيا في تقدم السيارات الكهربائية والروبوتات. تعتبر الإدارة الفعالة للبطارية ، وخاصة التقدير الدقيق لحالة الشحن ، ذات أهمية قصوى في كل من البحث النظري والتطبيقات العملية. ومع ذلك ، فإن تقديرها مهمة صعبة بسبب اختلاف الوقت المتأصل وخصائصها غير الخطية.

في هذه الدراسة ، نتعمق في الطبيعة المعقدة لخلايا الليثيوم ونستكشف إمكانيات الشبكات العصبية للتنبؤ بحالة الشحن. على وجه التحديد ، نقارن بين ثلاثة بنيات مختلفة للشبكة العصبية باستخدام مجموعة البيانات ، والتي توفر رؤى قيمة حول سلوك خلايا الليثيوم.

من خلال تحليل نظرية الشبكات العصبية وإجراء تجارب شاملة ، نهدف إلى تحسين دقة تقدير حالة الشحن. ينصب تركيزنا على فهم العلاقة المعقدة بين متغيرات الإدخال و حالة الشحن، وتسخير قوة الشبكات العصبية لالتقاط هذه العلاقة ونمذجتها بشكل فعال.

لا تساهم نتائج دراستنا فقط في مجال أنظمة إدارة البطاريات ولكن أيضا تسلط الضوء على الفهم الأوسع لتقدير حالة الشحن في سياق خلايا الليثيوم. من خلال تقييم أداء بنيات الشبكات العصبية المختلفة وتحليل النتائج التي تم الحصول عليها ، نكتسب رؤى قيمة لتحسين أداء البطارية وتعزيز الكفاءة الإجمالية للمركبات الكهربائية والروبوتات المتنقلة.

الكلمات المفتاحية: تقدير حالة الشحن ؛ بطارية ليثيوم ؛ الذكاء الاصطناعي؛ الشبكات العصبية العميقة.

Abstract

Battery technology plays a key factor in the advancement of electric cars and mobile robots. Efficient battery management, particularly accurate estimation of the state of charge (SOC), is of paramount importance in both theoretical research and practical applications. However, SOC estimation for batteries is a challenging task due to their inherent time-varying and non-linear characteristics.

In this study, we delve into the complex nature of lithium cells and explore the potential of neural networks for SOC prediction. Specifically, we compare three different neural network architectures using the LG HG2 dataset, which provides valuable insights into lithium cell behavior.

By analyzing the theory of neural networks and conducting comprehensive experiments, we aim to improve the accuracy of SOC estimation. Our focus is on understanding the intricate relationship between input variables and SOC, and harnessing the power of neural networks to capture and model this relationship effectively.

The results of our study not only contribute to the field of battery management systems but also shed light on the broader understanding of SOC estimation in the context of lithium cells. By evaluating the performance of different neural network architectures and analyzing the obtained results, we gain valuable insights into optimizing battery performance and enhancing the overall efficiency of electric vehicles and mobile robots.

Keywords: State of charge estimation; Lithium Battery; Artificial Intelligence; Deep neural networks.

Contents

List of Tables

List of Figures

Abbreviation

This document contains a number of abbreviations which we define here.

SOC	State Of Charge
BP	Backpropagation
SOH	State Of Health
BMS	Battery Management System
RBF	Radial Basis Function
SVM	Support Vector Machine
OCV	Open Circuit Voltage
FNN	Feed Forward Neural Network
ANN	Artificial Neural Network
BNN	Biological Neural Network
RNN	Recurrent Neural Network
UDDS	Urban Dynamometer Driving Schedule
HWFET	Highway Fuel Economy Test
US06	high acceleration aggressive driving schedule
RMSE	Root Mean Square Error
MSE	Mean Square Error
MAE	Mean Absolute Error
MAF	Moving Average Filter

CSV	Comma Separated values
LCO	Lithium Cobalt Oxide
LMO	Lithium Manganese Oxide
NMC	Nickel Manganese Cobalt Oxide
NCA	Lithium Nickel Cobalt Aluminium Oxide
LFP	Lithium Iron Phosphate
LTO	Lithium Titanate
LMO	Lithium Manganese Oxide
LMO	Lithium Manganese Oxide
LMO	Lithium Manganese Oxide
SGD	Stochastic Gradient Descent
Adam	Adaptive Moment Estimation

Symbols

$	Dollar
A	Ampere
Ah	Ampere-hour
mAh	Milliampere-hour
C	Celsius
mA	Milliamperes
V	Volt
Wh	Watt-hour
W	Watt
KWh	Kilo-Watt-hour
Ω	Ohm
mΩ	Milliohms
kg	kilogram

General Introduction

With the advancements in technology and the increasing demand for electric vehicles and portable electronics, lithium batteries have become a popular choice due to their high energy density and long cycle life. However, accurately estimating the state of charge of lithium cells has become a significant challenge. The SOC of a battery represents the amount of available energy, and knowing this information is crucial for optimizing battery usage, preventing over-discharge or over-charge conditions, and ensuring efficient and reliable operation. Inaccurate SOC estimation can lead to reduced performance, shortened battery lifespan, and even potential safety hazards. Therefore, developing reliable SOC estimation methods for lithium cells has garnered significant attention in research and industry. By addressing this problem, we can enhance the usability, efficiency, and safety of lithium-based systems, driving the progress of sustainable energy storage technologies [1].

Our project aims to tackle the challenge of accurately estimating the state of charge of lithium cells by leveraging the behavior and features of these batteries and applying artificial intelligence (AI) techniques. Recognizing the complexity and time-varying nature of lithium cells, we propose the use of neural networks, a powerful AI tool, to develop SOC estimation models. We explore three different architectures of neural networks to identify the most effective approach. By training these models on a comprehensive dataset of lithium cell behavior, we aim to achieve an impressive level of accuracy [2].

This document is made up of four (04) chapters, the first of which gives a general presentation of lithium cells, their types and characteristics, as well as a presentation of the problems involved in state-of-charge estimation. The second chapter is devoted to the state of the art of state-of-charge estimation methods. The third chapter presents artificial intelligence and neural network theory. The fourth chapter shows the path followed to obtain the optimal model and the comparison between 3 different classifiers, as well as the validation of the hypotheses.

Chapter 1

State of the Art Battery Technologies

Rechargeable batteries have come a long way in recent years and there have been many developments in their design and technology. One of the most popular types of rechargeable batteries is the lithium-ion battery, which is commonly used in portable electronic devices such as smartphones, laptops, and tablets. These batteries have a high energy density, which means they can store a lot of energy in a small space, and they have a relatively long lifespan compared to other types of rechargeable batteries. They are also relatively lightweight and have a low self-discharge rate, meaning they can hold their charge for a long time when not in use [1].

Another type of rechargeable battery is the nickel-metal hybrid battery, which is often used in hybrid and electric vehicles. These batteries have a higher capacity than lithium-ion batteries and are more resistant to damage caused by overcharging. However, they are typically heavier and more expensive than lithium-ion batteries.

There are also a number of newer technologies that are being developed for rechargeable batteries. For example, solid-state batteries, which use solid electrolytes rather than liquid ones, have the potential to be safer and have higher energy densities than traditional lithium-ion batteries. Additionally, researchers are working on developing batteries that use new materials such as silicon, graphene, and lithium-sulfur, which have the potential to improve the performance and lifespan of rechargeable batteries [3].

Overall, rechargeable batteries continue to evolve and improve, and it is likely that we will see further developments and innovations in this field in the coming years.

1.1 Types of Lithium Cells

1.1.1 Lithium Cobalt Oxide(LiCoO2) — LCO

Lithium Cobalt Oxide (LiCoO2), also known as LCO, is a popular choice for mobile phones, laptops, and digital cameras due to its high specific energy. The battery is made up of a cobalt oxide cathode and a graphite carbon anode, with lithium ions moving from the anode to the cathode during discharge and reversing during charge. However, LCO has a relatively short lifespan, low thermal stability, and limited load capabilities. Newer systems include nickel, manganese, and/or aluminum to improve performance and reduce cost. Li-cobalt should not be charged or discharged at a current higher than its C-rating to avoid overheating and stress.

1.1.2 Lithium Manganese Oxide (LiMn2O4) — LMO

LiMn2O4 or LMO is a Li-ion cell that improves ion flow and has low internal resistance, enabling fast charging and high-current discharging. LMO has high thermal stability and enhanced safety, but limited cycle and calendar life. It has roughly one-third lower capacity than Li-cobalt but can be optimized for longevity, maximum load current or high capacity. LMO is used for power tools, medical instruments, and hybrid and electric vehicles.

1.1.3 Lithium Nickel Manganese Cobalt Oxide (LiNiMnCoO2) — NMC

Nickel-manganese-cobalt (NMC) is a successful Li-ion system that can serve as Energy Cells or Power Cells. NMC combines nickel and manganese to enhance their respective strengths and cobalt stabilizes nickel. The cathode combination typically consists of one-third each of nickel, manganese, and cobalt. NMC is used in power tools, e-bikes, and electric powertrains. Its capacity and discharge current can be optimized for specific power or energy.

1.1.4 Lithium Iron Phosphate(LiFePO4) — LFP

Li-phosphate has a lower nominal voltage of 3.2V/cell, making its specific energy lower than cobalt-blended lithium-ion. It is more tolerant to full charge conditions and elevated temperatures, but has a higher self-discharge and requires cleanliness in manufacturing. Li-phosphate is often used as a lead acid starter battery replacement and can be topped up while driving, but overcharging for a prolonged time can stress the battery. Cold temperature can also affect performance and cranking ability.

1.1.5 Lithium Nickel Cobalt Aluminum Oxide (LiNiCoAlO2) — NCA

NCA has high specific energy, good specific power, and a long life span, but is less safe and more expensive compared to other lithium-ion batteries. It is a development of lithium nickel oxide and adding aluminum improves its stability.

1.1.6 Lithium Titanate (Li2TiO3) — LTO

Li-titanate batteries replace graphite in the anode with a spinel structure, have a nominal voltage of 2.40V, and can be fast charged with a high discharge current of 10C. They are safe, have excellent low-temperature discharge characteristics, and have a higher cycle count than regular Li-ion batteries. While they have thermal stability under high temperature, they are expensive and have a low specific energy of 65Wh/kg. They are used in electric powertrains, uninterruptible power supply, and solar-powered street lighting. Table 1.1 represents a general comparison between different lithium-ion battery chemistries

Table 1.1: Comparison of different lithium-ion battery chemistries. [1]

	Operating Voltage (V)	**Energy** (Wh/kg)	**Charge** (C)	**Discharge** (C)	**Cycle Life** (cycles)	**Cost** ($/kWh)
LCO	3.0-4.2	150-200	0.7C-1C	1C	500-1000	200
LMO	3.0-4.2	100-150	0.7C-1C	1C-10C	1000	150
NMC	3.0-4.2	150-220	0.7C-1C	1C-2C	1000	420
LFP	2.5-3.65	90-120	1C	1C-25C	2000	580
NCA	3.0-4.2	200-260	0.7C	1C	500	350
LTO	1.8-2.85	50-80	1C-5C	10C	3000-7000	1,005

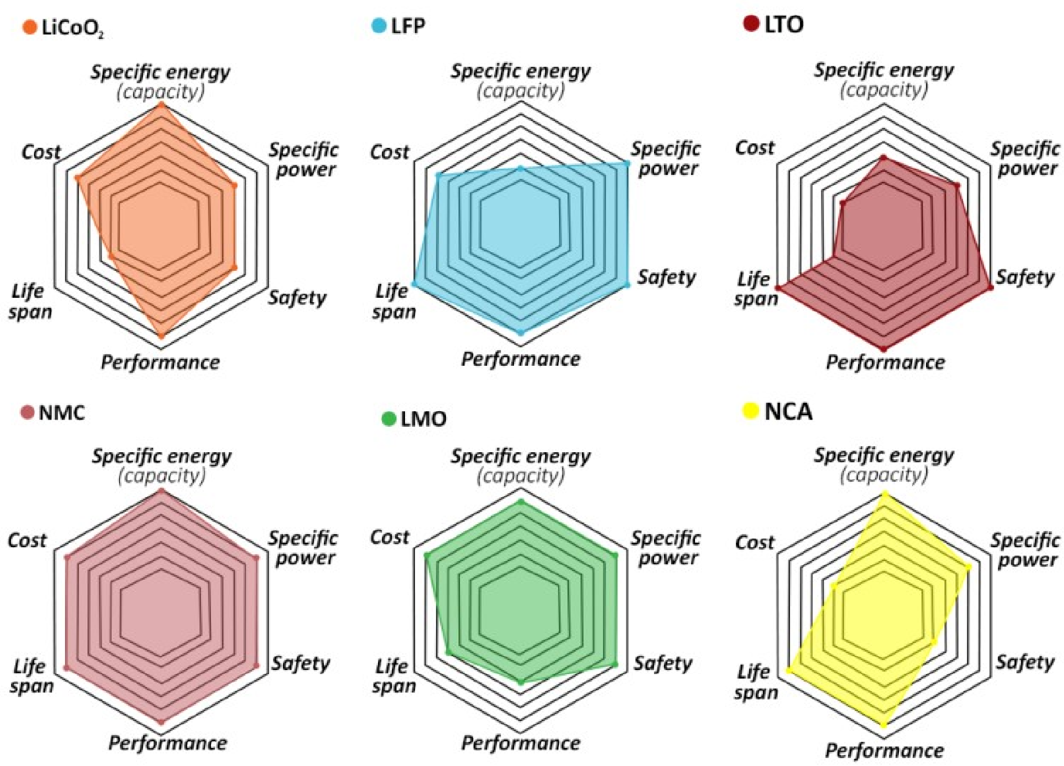

Figure 1.1: Comparison between different types of cathodes [4]

.

1.2 Batteries Characteristics

1.2.1 Nominal Voltage

The nominal voltage of a battery varies according to the technology with which it has been developed, it depends on the nature and concentration of the chemical species present in the in the battery, its value is only an average, since at the start of a discharge, the voltage is higher, and as the discharge progresses, the voltage drops, gradually decreases as the discharge progresses [5].

1.2.2 Capacity

This refers to the amount of energy that a battery can store, typically measured in milliampere-hours (mAh) or ampere-hours (Ah). The capacity of a battery is an important factor to consider, as it determines how long a device can operate on a single charge. For example, a battery with a higher capacity will typically be able to power a device for a longer period of time than a battery with a lower capacity.

1.2.3 Internal Resistance

The internal resistance of a battery is a disadvantageous characteristic because it causes the voltage at its terminals to drop as the current it delivers increases. This value is partly due to internal connections, chemical reaction inertia, built-in protection circuits, and the aging state of the battery. As a result, internal resistance is an important parameter in quantifying a battery's SOH. Its value is generally a few hundred milliohms (mΩ) [6].

1.2.4 Life Cycle

The life cycle of a battery represents the total number of charge-discharge cycles it can endure before reaching the end of its useful lifespan. It is a crucial consideration as it directly affects the overall longevity of the battery. Batteries with a high cycle life can withstand numerous charge-discharge cycles before requiring replacement, whereas batteries with a low cycle life may need to be replaced more frequently.

1.2.5 Charge/Discharge Rate

The rate, also known as the C-rate,at which a battery can be charged or discharged is referred to the charge/discharge rate, commonly measured in amperes (A), for example 1C refers to 3A in 3000 mAh battery. Significantly impacts both the overall performance and lifespan of a battery. Charging a battery too rapidly can lead to a reduced lifespan, while discharging it too quickly can result in diminished performance. Therefore, maintaining an appropriate C-rate is important for optimizing the efficiency and longevity of the battery.

1.2.6 Operating Temperature

This refers to the range of temperatures over which a battery can be used without damage or significant loss of performance. The operating temperature of a battery can have a significant impact on its performance and lifespan, as high temperatures can result in decreased performance and decreased lifespan, while low temperatures can result in decreased performance [7].

1.3 State of Charge

The state of charge (SOC) is one of the most important parameters for batteries, but its definition presents many different issues . In general, the SOC of a battery is defined as the ratio of its current capacity the nominal capacity, which is given by the manufacturer and represents the maximum amount of charge that can be stored in the battery. The SOC can be defined as follows:

$$SOC = \frac{Q_{used}}{Q_{rated}} \times 100\% \tag{1.1}$$

where SOC is the State of Charge, Q_{used} is the amount of charge discharged from the battery, and Q_{rated} is the rated capacity of the battery.

The accurate estimation of SOC remains a significant challenge in battery usage. A precise SOC estimation provides information about the remaining capacity, enabling the application to implement efficient control strategies for energy conservation, protection against over-discharge which means enhancement of battery lifespan. However, a battery functions as a source of chemical energy storage, and the chemical energy cannot be directly measured, which poses a challenge in estimating the state of charge. The complexity of accurately estimating the SOC stems from limitations in battery models and parametric uncertainties. These limitations result in instances of low accuracy and unreliable SOC estimates in practical applications [8].

1.3.1 Dependencies Of Measurements State Of Charge

1.3.2 Open Circuit Voltage

The open Circuit Voltage OCV of a battery cell is the potential difference between the positive electrode and the negative electrode when no current flows and the electrode potentials are at equilibrium. A battery undergoing charge or discharge does not exhibit this potential since it is modified by kinetic effects . Open circuit voltage, as a nonlinear function of state of charge of lithium ion battery, commonly obtained through offline OCV test at certain ambient temperatures and aging stages. The OCV-SOC relationship may be inaccurate in real application due to the difference in operation conditions.[5]

1.3.3 Discharge Current

The discharge current has a direct effect on the internal resistance of a lithium cell, which in turn affects the measurement of SOC. As the discharge current increases, the internal resistance of the battery also increases. This increased resistance leads to a higher voltage drop across the battery during discharge [9].

When measuring SOC based on voltage, the voltage drop caused by the internal resistance can result in an underestimation of the actual SOC. This is because the measured voltage will be lower than expected due to the voltage drop across the internal resistance. To accurately measure SOC, it is important to account for the influence of discharge current on the internal resistance. This can be done by implementing compensation techniques or using advanced algorithms that consider the dynamic behavior of the battery's internal resistance during discharge. By taking into account the discharge current and its impact on the internal resistance, more accurate SOC measurements can be obtained [9]. Image 1.2 shows the relationship between internal resistance and discharge current .

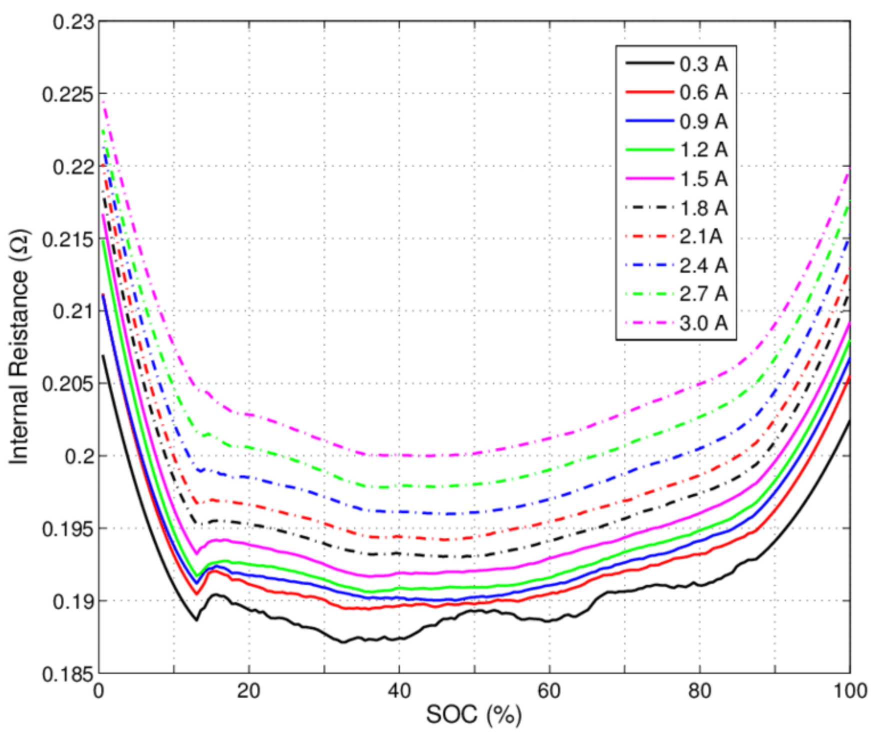

Figure 1.2: Internal resistance variation for different discharge currents and different SOC levels. [9]

.

1.3.4 Temperature

Temperature has a significant impact on battery SOC measurement. Higher temperatures increase SOC, while lower temperatures decrease it due to temperature-dependent battery capacity and performance. A study on lithium-ion batteries [7] developed a simplified battery model integrating an OCV-SOC-temperature table. The model improved SOC estimation accuracy, yielding smaller RMSE errors at temperatures ranging from 0°C to 50°C. See the figure below for the OCV-SOC relationship in different temperature tests.

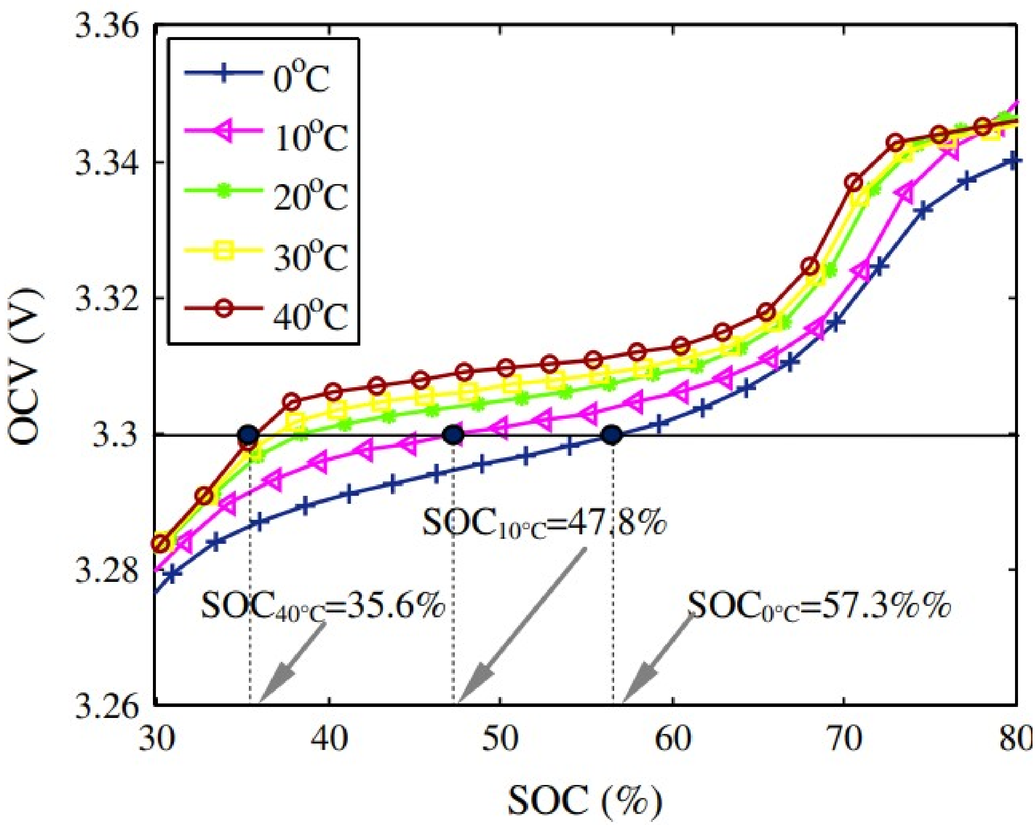

Figure 1.3: OCV–SOC curves between 30% and 80% SOC at different temperatures[7]

1.4 Conclusion

In conclusion, this chapter provides an overview of rechargeable batteries, with a particular focus on lithium-ion batteries and their various chemistries. It discusses the advancements in rechargeable battery technology and the different types of lithium cells, including their characteristics and applications.

Lithium-ion batteries, have different properties and performance characteristics based on chemistries. They vary in terms of operating voltage, energy density, charge/discharge rate, cycle life, and cost. Each chemistry has its own advantages and disadvantages, making them suitable for specific applications such as smartphones, laptops, electric vehicles, power tools, and renewable energy systems.

The chapter also discusses important battery characteristics, including nominal voltage, capacity, internal resistance, life cycle, charge/discharge rate, and operating temperature. These characteristics play a key role in determining the performance and lifespan of batteries.

Additionally, the chapter highlights the challenge of accurately estimating the state of charge of a battery and the dependencies of SOC measurements on factors such as open circuit voltage, discharge current, and temperature.

Overall, this chapter provides a comprehensive introduction to rechargeable batteries and their key features. The advancements in rechargeable batteries continue to drive innovation in various industries, and further research and development are expected to enhance their performance, safety, and cost-effectiveness in the future.

Chapter 2

State of Charge Estimation Methods

2.1 Introduction

This chapter provides an overview of state of charge estimation methods for lithium batteries. Various techniques for estimating the SOC are discussed, including direct measurement methods, book-keeping methods, adaptive systems, and hybrid methods. The objective of this chapter is to present a comprehensive understanding of these estimation methods and their applicability in determining the SOC of batteries. The advantages and limitations of each method are explored, and insights are provided to guide the selection and implementation of suitable SOC estimation techniques for different battery systems [8]. By delving into the intricacies of these methods, this chapter aims to contribute to the advancement of accurate and reliable SOC estimation for battery applications.

2.2 Direct Measurement

Direct measurement methods involve measuring specific physical characteristics of a battery, such as terminal voltage and impedance. Various direct measurement techniques have been used, including the open circuit voltage method, terminal voltage method, impedance measurement method, and impedance spectroscopy method [8].

2.2.1 Open Circuit Voltage Method

OCV method is a direct measurement method that estimates the SOC of a battery based on the OCV of the battery. Figure 1 shows the typical relationship between the SOC and OCV of a Li-ion battery. However, it should be noted that the relationship between the OCV and SOC may vary among batteries, and the conventional OCV-SOC relationship may not apply to all batteries [10]. Although the OCV method can provide accurate SOC estimation, it has some limitations. The OCV method is based on the voltage measurement of external battery terminals when they are disconnected from the loads for more than two hours, and such a long disconnection time may not be feasible for many battery applications. Therefore, the OCV method may not be suitable for applications where frequent SOC estimation is required [5].

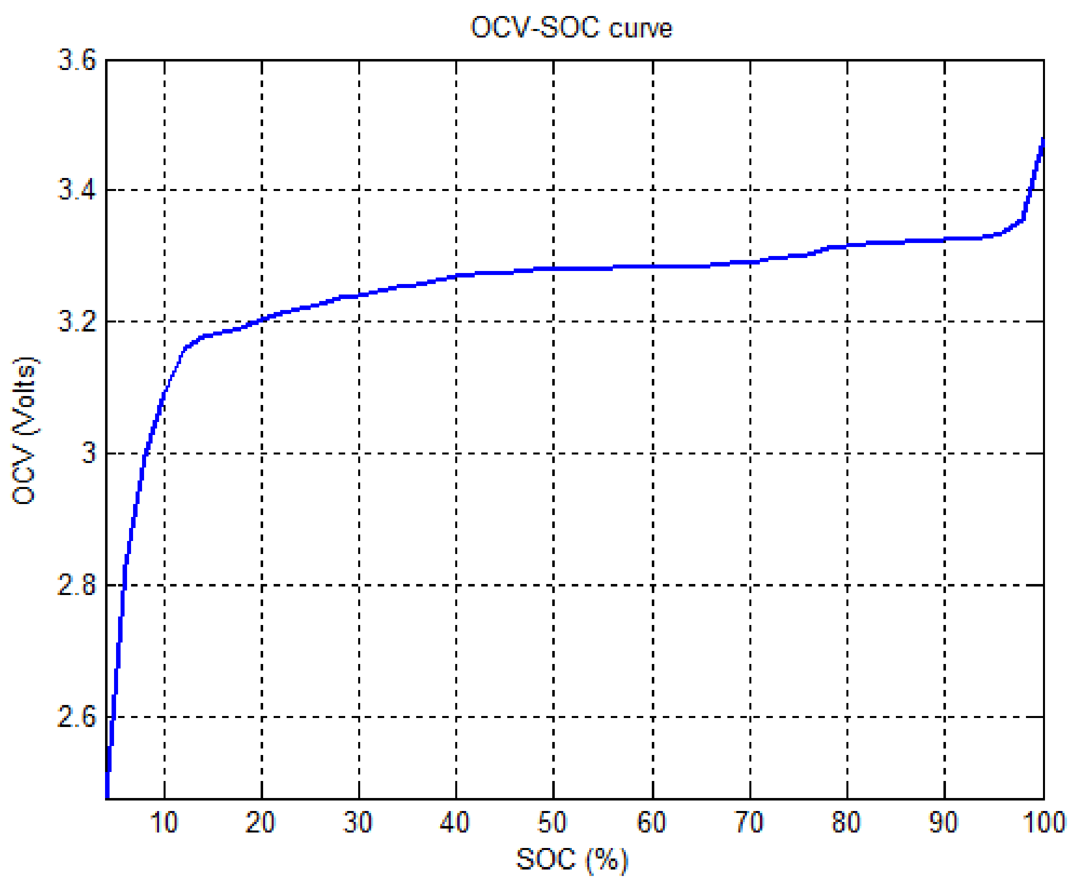

Figure 2.1: Experimental OCV-SOC curve for LFP cell [11].

2.2.2 Terminal Voltage Method

The Terminal Voltage Method for SOC estimation is based on the relationship between the terminal voltage drops and the internal impedance of the battery during discharge. The electromotive force of the battery is proportional to the terminal voltage. This method has been implemented under various discharge currents and temperatures. However, it should be noted that the accuracy of the terminal voltage method is affected by the internal resistances of the battery, which may change with usage and aging. Thus, this method may not always provide accurate SOC estimates [12].

2.2.3 Impedance Spectroscopy Method

Impedance measurements can give us information about a battery's SOC. However, different batteries may have different impedance parameters and their variations with SOC. Therefore, a wide range of impedance experiments are needed to identify and use these parameters to estimate the SOC. To estimate the SOC using impedance measurements, it can be found the values of model impedance by fitting measured impedance values to known impedance at various SOC levels [13].

2.3 Book-keeping Method

The book-keeping estimation method uses the battery's discharging current data as input. This method considers some internal battery effects such as self-discharge, capacity-loss, and discharging efficiency. Two types of book-keeping estimation methods are used: Coulomb counting method and modified Coulomb counting method.

2.3.1 Coulomb Counting Method

Coulomb counting is another method for estimating SOC. This method uses the principle that the amount of charge that flows into or out of a battery is equal to the product of the current and the time [14]. By measuring the current and integrating it over time, the amount of charge that has flowed in or out of the battery can be calculated, and thus the SOC can be estimated by the following equation :

$$SOC(t) = SOC(t-1) + \frac{\int_{t-\Delta}^{t} I(\tau)d\tau}{Qn} \tag{2.1}$$

where $SOC(t)$ is the state of charge at time t, $SOC(t-1)$ is the state of charge at the previous time step, $I(t)$ is the current at time t, Q is the battery capacity, n is the number of series-connected cells, and Δ is the sampling time interval.

The precision of the coulomb counting method relies on the precision of the initial SOC estimate and the accuracy of battery current measurement (dependent on the accuracy of current sensors), potentially causing error accumulation. While this method is not well-suited for real-time SOC estimation, it can be used to validate the precision of SOC estimations generated by other methods [3].

The modified Coulomb counting method is a variation of the Coulomb counting method that takes into account the non-linear relationship between the battery's voltage and SOC. This method uses a correction factor based on the battery's voltage to adjust the SOC estimate, resulting in a more accurate estimation.

2.4 Adaptive Systems

Adaptive systems are a class of methods for estimating the SOC of lithium-ion batteries that can self-design and automatically adjust the SOC estimation algorithm based on different discharging conditions. Several new adaptive systems for SOC estimation have been developed in recent years, including.

Feed Forward Neural Network

The FNN, a type of artificial neural network, is popular for its ability to perform nonlinear mapping and self-learning. It is commonly used for SOC estimation, as the relationship between input and target is complex and nonlinear. The network has an input layer for current, voltage, and temperature, one or more hidden layers for nonlinear mapping, and an output layer for SOC. Nodes between layers are interconnected, and the hidden and output layers use activation functions for processing. Figure 2.2 shows the structure of a feed-forward neural network for SOC estimation. The output of a processing node j in the hidden or output layer is given by:

$$y_j = F(u_j) = F\left(\sum_i x_{ij} w_{ji} + b_j\right) \tag{2.2}$$

where y_j is the output of processing node j, F is the activation function, u_j is the weighted sum of inputs to node j, x_{ij} is the input from node i to node j, w_{ji} is the weight of the connection between nodes i and j, and b_j is the bias term for node j.

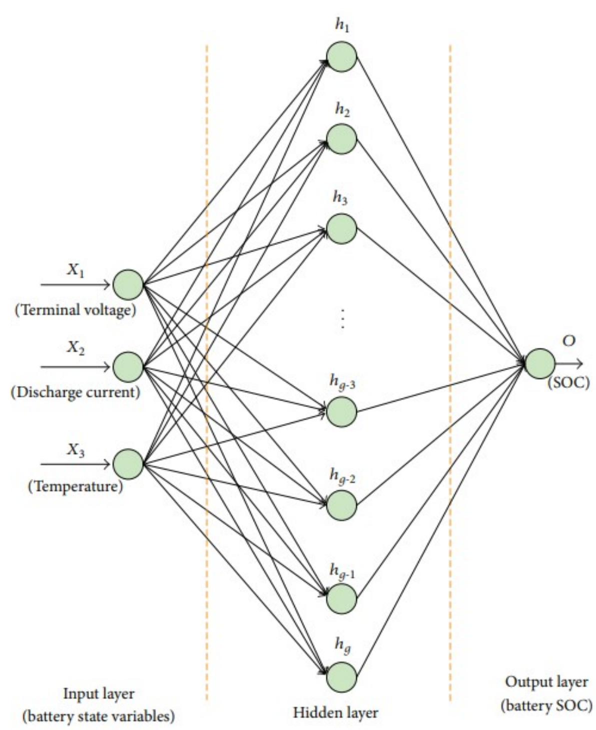

Figure 2.2: The architecture of the SOC estimating Feed forward neural network [11].

2.4.1 RBF Neural Network

Due to its nonlinear mapping characteristics, self-organized study ability, fast training, and ability to converge to global optimization and approach the function in the best way. Thus, RBF NN is used to estimate the SOC of the batteries, given its significant advantages. The RBF NN structure includes one input layer, one hidden layer, and one output layer.The output $h_j(X)$ of the j-th neuron in the hidden layer is given by :

$$h_j(X) = \varphi(|X - C_j|), \tag{2.3}$$

where $\varphi(\cdot)$ is the radial basis function, $C_j \in R^n$ is the center of the j-the hidden neuron, and $|\cdot|$ is the Euclidean norm.The output Y (X), which is designed as the SOC estimation, is the linearly combined signal of the outputs $h_j(X)$ from the hidden layer with the synaptic weights. $Y(X) = \sum_{j=1}^{m} w_j h_j(X)$ [15] .

If $\varphi(\cdot)$ is adopted to be the Gaussian function, $h_j(X)$ is described by

$$h_j(X) = \exp\left(-\frac{|X - C_j|^2}{2\sigma_j^2}\right), \tag{2.4}$$

where σ_j is the spread of the j^{th} neuron for the i^{th} input signal.

2.4.2 Support Vector Machine

Support vector machines (SVM) are supervised learning methods used for classification and regression that have the ability to approximate any multivariate function to any level of accuracy. Originally developed for solving classification problems. The models produced by SVM rely on a subset of the training data as the cost function used in building the model disregards points beyond the margin and training data that are close (within a threshold) to the model prediction, respectively [16].

Kalman filter

this method uses Electrical circuit model, which aim to reproduce the behaviour of the electrical quantities in the battery (SOC, Voltage and current) and allows them to be easily formulated into mathematical formula. This model, as shown in Figure, use standard electrical components such as resistors, capacitors and voltage source. The values of the model are obtained from experimentation [17].

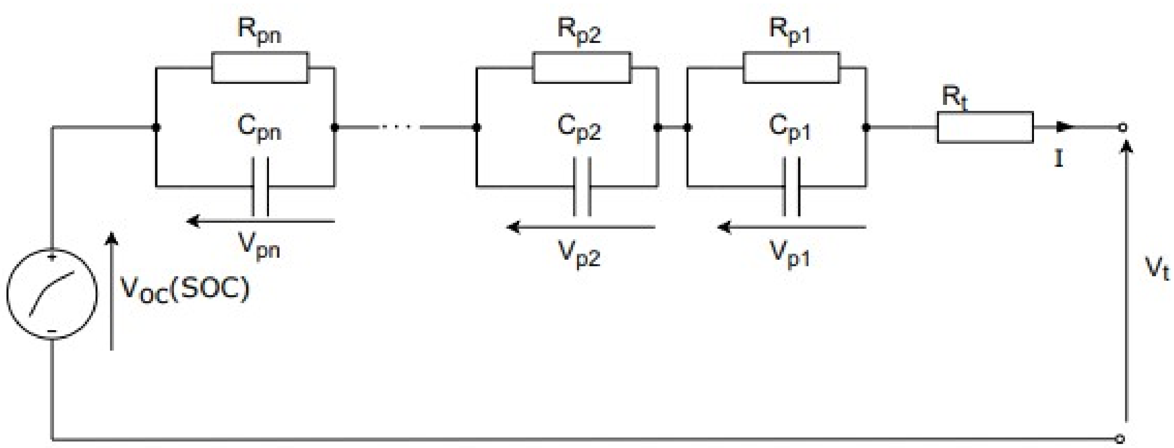

Figure 2.3: RC complex equivalent electrical model for lithium batteries

The state equation describing this electrical system is given as follow :

$$
\begin{cases}
\begin{bmatrix} \dot{SOC} \\ \dot{V}_{p1} \\ \dot{V}_{p2} \\ \vdots \\ \dot{V}_{pn} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ 0 & \frac{-1}{R_{p1}C_{p1}} & 0 & \cdots & 0 \\ 0 & 0 & \frac{-1}{R_{p2}C_{p2}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \frac{-1}{R_{pn}C_{pn}} \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_{p1} \\ V_{p2} \\ \vdots \\ V_{pn} \end{bmatrix} + \begin{bmatrix} \frac{1}{C_{actual}} \\ \frac{1}{C_{p1}} \\ \frac{1}{C_{p2}} \\ \vdots \\ \frac{1}{C_{pn}} \end{bmatrix} \cdot I \\
V_t = \begin{bmatrix} a & 1 & 1 & \cdots & 1 \end{bmatrix} \cdot \begin{bmatrix} SOC \\ V_{p1} \\ V_{p2} \\ \vdots \\ V_{pn} \end{bmatrix} + R_t \cdot I + b
\end{cases}
\tag{2.5}
$$

The system dynamics are first translated into a model, which is then used in conjunction with a Sliding Mode Observer to estimate the state of charge. However, the estimated SOC is subject to noise, so a Kalman filter is employed to remove the noise and improve the accuracy of the SOC estimation [17].

2.4.3 Hybrid Methods

Hybrid methods for estimating the SOC of lithium batteries combine the advantages of multiple SOC estimation methods to achieve better accuracy and performance. By integrating multiple methods, such as Coulomb counting, Kalman filter, and neural network, hybrid models can effectively compensate for the limitations and uncertainties of individual methods.

The literature has shown that hybrid methods generally produce more accurate SOC estimations compared to single methods. However, developing a hybrid model requires careful selection and optimization of each method, as well as determining the appropriate weighting and fusion strategy for combining the results [8].

Conclusion

This chapter provides an overview of state of charge estimation methods for lithium batteries. It covers direct measurement methods, such as open circuit voltage, terminal voltage, and impedance spectroscopy. The book-keeping method, including Coulomb counting and modified Coulomb counting, is also discussed. Additionally, adaptive systems like feed-forward neural networks, radial-basis-function neural networks, support vector machines, and Kalman filters are explored. The chapter concludes by highlighting the benefits of hybrid methods that combine multiple techniques for improved SOC estimation. Overall, the chapter offers valuable insights into various SOC estimation methods and their suitability for different battery systems.

Chapter 3

Artificial Intelligence And Theory Of Neural Networks

3.1 Introduction

Neural networks (NN) have revolutionized the field of artificial intelligence by simulating the complex workings of the human brain. The concept of neural networks traces its roots back to the 1940s, when Warren McCulloch and Walter Pitts developed the first artificial neuron model [18]. However, it was in the 1950s and 1960s that significant progress was made, with Frank Rosenblatt's perceptron algorithm becoming a cornerstone of neural network research . The field faced setbacks in the 1970s and 1980s, known as the "AI winter," but a breakthrough came in 1986 with the development of the backpropagation algorithm by Geoffrey Hinton, David Rumelhart, and Ronald Williams . Since then, neural networks have experienced exponential growth, leading to advancements in various domains such as image and speech recognition, natural language processing, and autonomous vehicles [19].

3.2 Neurons: Building Blocks of Neural Networks

The fundamental units that form the basis of neural networks are neurons, which are inspired by the structure and functionality of biological neurons in the human brain. In this section, the key differences between biological neurons and artificial neurons will be explored, the structure and functionality of artificial neurons will be examined, and the crucial role they play in neural networks will be understood [20].

3.2.1 Biological Neurons vs. Artificial Neurons

Biological neurons, also known as nerve cells, are the core components of the nervous system in living organisms. They receive and transmit electrochemical signals, enabling the communication and processing of information within the brain and throughout the body. Artificial neurons, on the other hand, are mathematical abstractions designed to mimic the behavior and functionality of biological neurons [2].

While biological neurons are highly complex and intricate in structure, artificial neurons are simplified mathematical models that perform computations based on input signals. Artificial neurons are primarily used in artificial neural networks to process information and make predictions or decisions [2].

3.2.2 Structure and Functionality of Artificial Neurons

Artificial neurons consist of three main components: inputs, weights, and an activation function. The inputs represent the information received by the neuron, which can be numerical values or binary signals. Each input is associated with a weight, which determines the strength of its influence on the neuron's output. The weights signify the importance or relevance of each input in the overall computation [2]. Image 3.1 represents a biological and an artificial neuron.

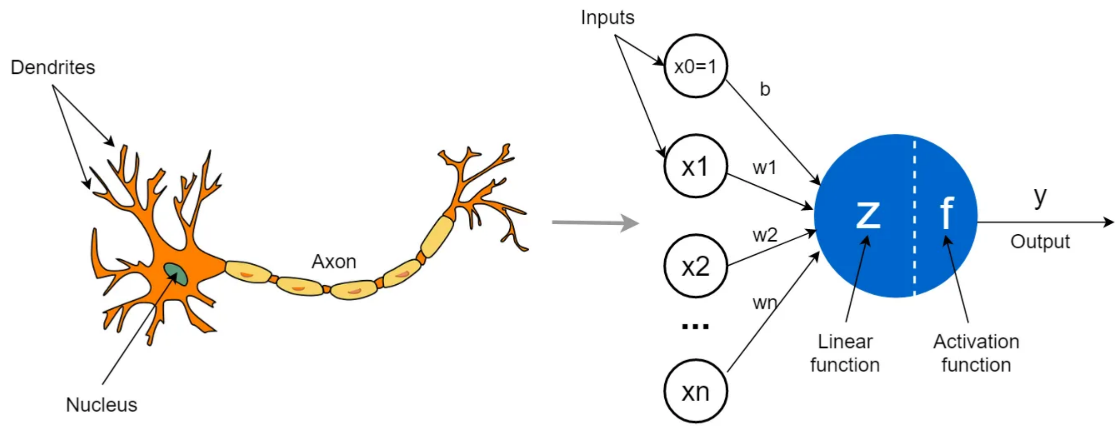

Figure 3.1: Biological vs, artificial neurons [11].

3.2.3 Role of Neurons in Neural Networks

Neurons are the fundamental building blocks of NN, which are computational models inspired by the interconnectedness of biological neurons in the brain. Neural networks are composed of multiple layers of interconnected neurons, forming a network architecture. Each neuron in a neural network receives inputs from the previous layer's neurons, processes the information, and passes the output to the next layer.

The role of neurons in neural networks is to perform computations and transformations on the input data, enabling the network to learn and make predictions. Neurons work collectively to capture complex patterns, relationships, and representations within the data. Through an iterative process known as training, the neural network adjusts the weights of the neurons based on the provided data, optimizing its ability to generalize and make accurate predictions on new, unseen data.

The connections between neurons, determined by the weights, form a vast web of interconnected information flow within the neural network. This interconnectedness allows neural networks to model and solve a wide range of problems, including image recognition, natural language processing, and even state of charge estimation of lithium cells [2].

3.3 Weights and Biases

3.3.1 Importance of Weights and Biases

The weights and biases in a neural network are of paramount importance. They significantly influence the behavior and performance of the network. The weights determine the strength of connections between neurons, while the biases provide an additional level of flexibility by adjusting the output of each neuron. Proper initialization and adjustment of weights and biases can greatly impact the network's ability to learn and make accurate predictions.

3.3.2 Initialization and Update Methods

Sselecting appropriate methods for weight and bias initialization and update is crucial for optimizing the performance of a neural network. Techniques like random initialization, Xavier initialization, and He initialization help set suitable initial values for weights, aiding in the network's learning process. Update methods such as gradient descent, stochastic gradient descent, and Adam optimization algorithm determine how the network adjusts its weights and biases during training [2]. Careful selection and utilization of these methods can significantly impact the network's convergence, learning speed, accuracy, and generalization capabilities. Poor initialization or improper update methods can hinder learning and lead to suboptimal solutions, while well-initialized and properly updated weights and biases contribute to efficient learning and improved model performance on unseen data [2].

3.4 Types of Learning in Neural Networks

3.4.1 Supervised Learning

Supervised learning is a machine learning technique where a model learns from labeled training data. In this approach, each training example consists of an input and its corresponding output label. The goal of supervised learning is to train a model that can accurately predict the output labels for new, unseen inputs [21].

During the training process, the model learns the relationship between the input features and the corresponding output labels by minimizing a predefined loss function. This is achieved by iteratively adjusting the model's parameters, such as weights and biases, using optimization algorithms like gradient descent. The model's performance is evaluated using evaluation metrics, such as accuracy or mean squared error, on a separate validation set.

Supervised learning is widely used in various applications, including image classification, sentiment analysis, and speech recognition. It enables the model to generalize from the provided labeled examples and make predictions on unseen data [20].

3.4.2 Unsupervised Learning

Unsupervised learning is a machine learning technique where a model learns patterns and structures from unlabeled data. Unlike supervised learning, there are no explicit output labels associated with the input data. Instead, the model discovers hidden patterns, structures, or relationships within the data.

Unsupervised learning can provide valuable insights into the underlying structure of the data and help identify hidden patterns or anomalies. It is widely used in applications such as customer segmentation, anomaly detection, and recommender systems [22].

3.4.3 Reinforcement Learning

Reinforcement learning is a type of machine learning that focuses on learning through interaction with an environment. In this approach, an agent learns to make sequential decisions to maximize a cumulative reward signal [22].

Reinforcement learning is a process where an agent interacts with the environment, taking actions and receiving rewards or penalties. Its goal is to learn a policy that maximizes long-term rewards. Through trial-and-error, the agent explores actions and learns from the outcomes. Value functions estimate rewards, and policy optimization finds the best policy based on these estimates [22].

Reinforcement learning has been successfully applied in various domains, including robotics, game playing, and autonomous systems. It allows the agent to learn from experience and improve its decision-making capabilities over time [22]. Figure 3.2 resumes basic types of machine learning models.

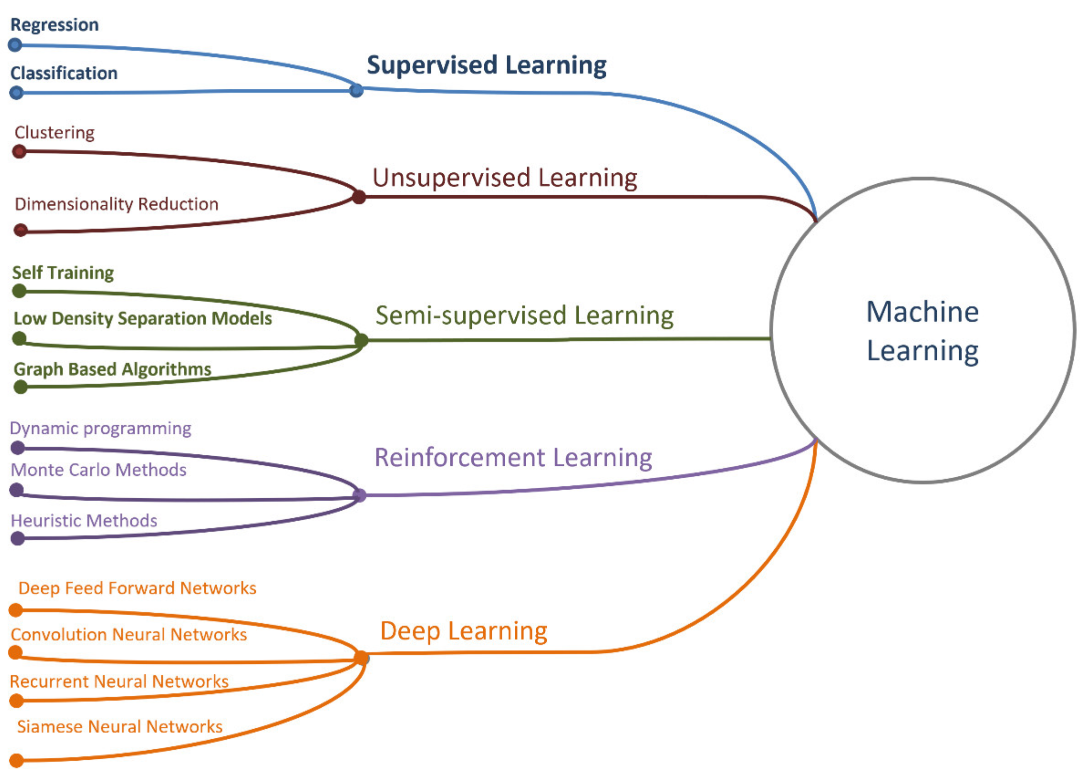

Figure 3.2: Basic types of machine learning models [22]

3.5 Activation Functions

3.5.1 Role and Importance of Activation Functions

Activation functions allows introducing non-linearity to the model. They determine the output of a neuron and help in mapping the input to the desired range or activating specific neurons based on the input. Activation functions are responsible for introducing complex behaviors and decision-making capabilities into the neural network [2].

The activation function is applied to the weighted sum of the inputs and biases of a neuron to compute its output. It adds non-linearity to the model, enabling neural networks to learn and represent complex relationships between inputs and outputs. Without activation functions, a neural network would simply be a linear regression model, incapable of modeling complex patterns and nonlinear relationships.

The choice of activation function is essential as it can impact the network's performance, convergence speed, and ability to generalize. Different activation functions exhibit distinct properties, and their selection depends on the nature of the problem and the characteristics of the data [23].

3.5.2 Common Activation Functions

Sigmoid

The sigmoid activation function, also known as the logistic function, is widely used in neural networks. It squeezes the input values between 0 and 1, resulting in a smooth, S-shaped curve. The sigmoid function is differentiable and allows for the interpretation of the output as a probability. However, it suffers from the vanishing gradient problem, which can hinder the training process, especially in deep networks.

Rectified Linear Unit (ReLU)

The Rectified Linear Unit (ReLU) is one of the most popular activation functions in deep learning. It returns the input directly if it is positive, and zero otherwise. ReLU has a simple implementation and provides faster training compared to sigmoid and tanh [2].

Tansig

The tansig activation function, also known as the hyperbolic tangent sigmoid function, is commonly used in neural networks. It maps the input values to the range (-1, 1), similar to the tanh function. The tansig function exhibits an S-shaped curve and provides stronger non-linearity compared to the sigmoid function. It is useful for tasks where the outputs need to be bounded within a specific range and can be advantageous in certain network architectures.

The tansig activation function can be defined as:

$$\text{tansig}(x) = \frac{2}{1 + e^{-2x}} - 1$$

Other Activation Functions

Besides the sigmoid, ReLU, and tanh functions, there are other activation functions used in neural networks. Some examples include:

- Softmax: This activation function is commonly used in the output layer of classification tasks. It normalizes the outputs into a probability distribution, allowing the model to make class predictions.

- Leaky ReLU: This variation of the ReLU function addresses the dying ReLU problem by introducing a small slope for negative inputs, preventing the neurons from becoming completely inactive.

- Parametric ReLU (PReLU): PReLU generalizes ReLU by introducing learnable parameters for the negative part of the input, providing more flexibility and reducing the likelihood of dead neurons.

- Exponential Linear Unit (ELU): ELU function is similar to ReLU for positive inputs but allows negative inputs to have non-zero outputs. It helps alleviate the dying

ReLU problem and can lead to improved performance.

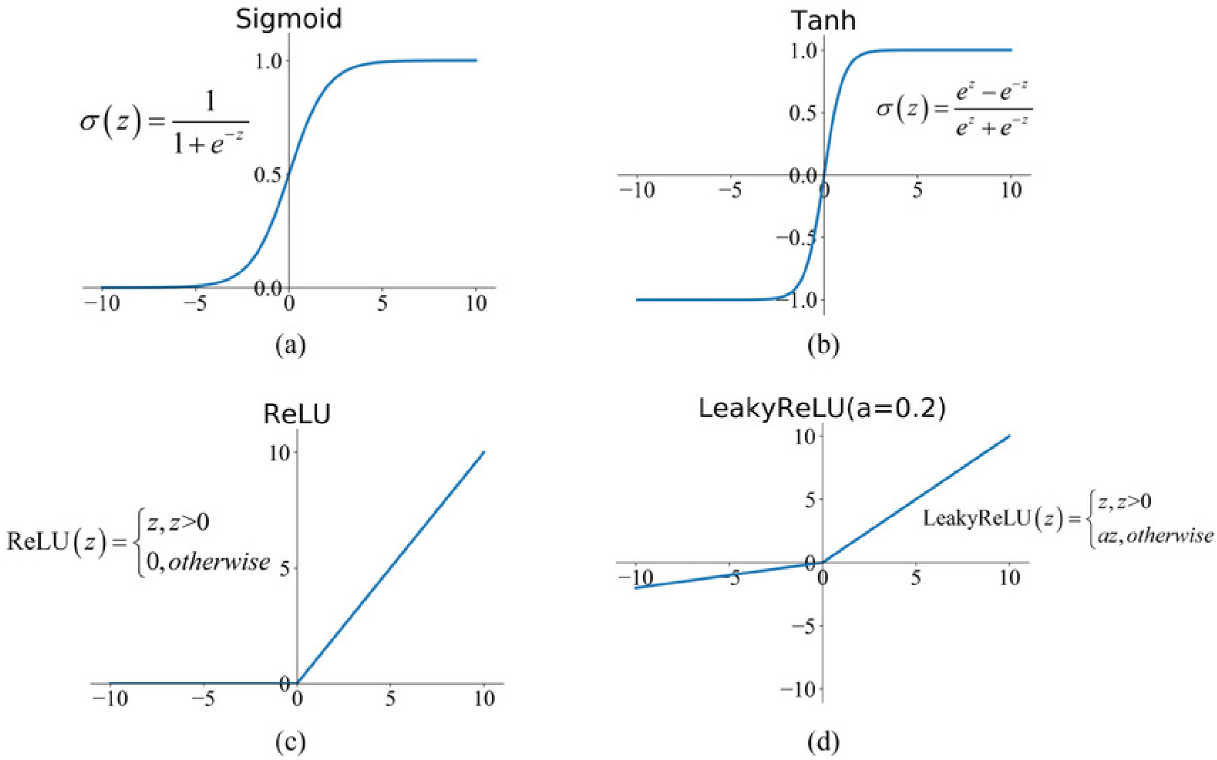

Figure 3.3: Commonly used activation functions: (a) Sigmoid, (b) Tanh, (c) ReLU, and (d) LReLU [23]

3.6 Gradient and Backpropagation

Gradient descent is an optimization algorithm widely used in training neural networks. The goal is to minimize the loss function by iteratively adjusting the model's parameters based on the gradient of the loss with respect to those parameters. The gradient represents the direction of steepest ascent, by moving in the opposite direction, the minimum of the loss function can be descended towards [24].

3.6.1 Cost Function

The cost function, also known as the loss function or objective function, quantifies the difference between the predicted outputs of the neural network and the actual values. It measures the model's performance and provides a way to evaluate how well the network is learning. The choice of the cost function depends on the specific task at hand. For example, in regression problems, the mean squared error (MSE) or mean absolute error (MAE) are commonly used, while in classification tasks, the cross-entropy loss function is often employed [2].

The Mean Squared Error (MSE) is given by:

$$\text{MSE}\ (y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

where y represents the actual values, $\hat{y}$ represents the predicted values, and n is the number of data points.

The Mean Absolute Error (MAE) is given by:

$$\text{MAE}\ (y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where y represents the actual values, $\hat{y}$ represents the predicted values, and n is the number of data points.

3.6.2 Understanding Gradient Descent

Gradient descent is an optimization algorithm widely used in training neural networks. The goal is to minimize the cost function by iteratively adjusting the model's parameters based on the gradient of the cost function with respect to those parameters. The gradient represents the direction of steepest descent, by moving in the opposite direction, the minimum of the cost function can be descended towards [24].

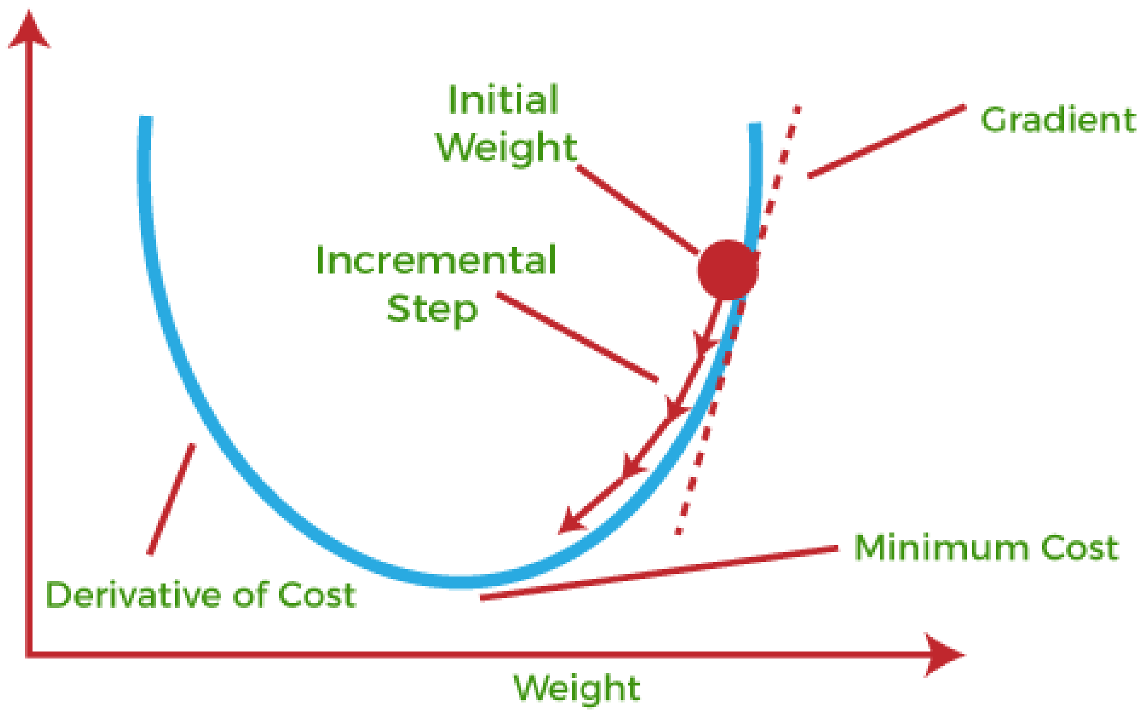

Figure 3.4: Gradient Descent in Machine Learning [24]

3.6.3 Backpropagation Algorithm

Backpropagation is a fundamental algorithm in training neural networks that plays a crucial role in optimizing their performance. It enables the efficient computation of gradients with respect to the weights and biases in each layer of the network. By propagating the error backward through the network, backpropagation allows us to calculate the gradients layer by layer, leveraging the chain rule of calculus [24].

The process of backpropagation starts with the computation of the output values of the neural network given a set of input data. These outputs are then compared to the true values using a cost function, which quantifies the discrepancy between the predicted and actual outputs. The goal is to minimize this cost function.

To achieve this, backpropagation works by iteratively adjusting the weights and biases of the network using gradient descent. The gradients of the cost function with respect to the weights and biases are calculated during the backward pass. The chain rule is utilized to efficiently compute these gradients by propagating the error from the output layer to the input layer.

During the backward pass, the error at each layer is determined by considering the contribution of that layer's weights and biases to the overall error. The gradients are then computed by multiplying this error by the partial derivatives of the activation function with respect to the weighted inputs of the neurons in the layer. These gradients indicate the direction and magnitude of the changes required to minimize the cost function [24].

With the gradients in hand, the weights and biases are updated in the opposite direction of the gradient using the gradient descent algorithm. The magnitude of the update is controlled by the learning rate, which determines the step size taken in the direction of the negative gradient. By repeatedly applying this process over multiple iterations, the network gradually converges to a set of weights and biases that minimize the cost function [2].

By leveraging backpropagation, neural networks can learn from data and adjust their parameters to improve their performance on various tasks such as classification, regression, and pattern recognition. The ability to efficiently compute gradients and update the network's weights and biases through backpropagation has been instrumental in the success of deep learning and the training of complex neural network architectures [2].

3.6.4 Optimization and Challenges

Gradient descent and backpropagation present optimization challenges in NN training. One common issue is the vanishing or exploding gradients problem, where the gradients become extremely small or large during backpropagation. This can impede learning or cause instability in the training process. Techniques like weight initialization, gradient clipping, and using activation functions that alleviate gradient issues can help address these challenges [19].

Another challenge is finding an appropriate learning rate for gradient descent. A learning rate that is too small can lead to slow convergence, while a learning rate that is too large can cause overshooting and divergence. Various learning rate scheduling strategies, adaptive optimization algorithms, and regularization techniques are employed to optimize the learning process and improve convergence speed [19].

Furthermore, avoiding overfitting is crucial in training neural networks. Overfitting occurs when the model learns the training data too well but fails to generalize to unseen data. Regularization techniques such as L1 and L2 regularization, dropout, and early stopping are commonly employed to prevent overfitting and enhance the generalization ability of the network.

3.7 Learning Rate and Its Implications

3.7.1 Definition and Significance of Learning Rate

The learning rate is a hyperparameter that determines the magnitude of parameter updates during the training process. It controls the speed at which the model converges to an optimal solution. By adjusting the learning rate, we can influence the pace of learning and the convergence behavior of the neural network [25].

3.7.2 Effect of Learning Rate on Model Training

The choice of learning rate significantly impacts the training dynamics and the quality of the resulting model. An excessively high learning rate can cause the loss function to oscillate or diverge, preventing convergence. On the other hand, an overly small learning rate can result in slow convergence, especially in the early stages of training.

It is important to strike a balance between convergence speed and accuracy. An optimal learning rate allows the model to converge steadily towards the optimal solution while avoiding overshooting or getting stuck in suboptimal regions. Different learning rates may be suitable for different stages of training, with higher rates initially for rapid progress and lower rates later for fine-tuning [25].

3.7.3 Strategies for Choosing an Appropriate Learning Rate

- **Manual Tuning:** Start with a reasonable learning rate and observe the training dynamics. If the loss function is not converging, decrease the learning rate. If the convergence is too slow, increase it. Iterate this process until a satisfactory result is obtained [25].

- **Learning Rate Schedules:** Gradually reduce the learning rate over time. This allows for more substantial updates in the early stages and fine-grained adjustments later on, facilitating convergence to an optimal solution [2].

- **Adaptive Learning Rate Methods:** Utilize adaptive learning rate methods like Adam. These algorithms dynamically adjust the learning rate based on the gradient's magnitude, effectively reducing the need for manual tuning and improving convergence [2].

- **Learning Rate Warm-up:** Start training with a low learning rate and gradually increase it. This helps to prevent convergence to poor local optima and enables the model to explore a larger portion of the parameter space before settling down [2].

3.8 Data: Fueling Neural Networks

Data plays a significant role in training neural networks and fueling their performance. Here are the key aspects related to data in neural network training:

3.8.1 Role of Data in Neural Network Training

Data serves as the foundation for training neural networks. It provides the necessary information for the network to learn and make predictions. The quality, quantity, and diversity of the data directly impact the network's ability to generalize and accurately solve the task at hand.

3.8.2 Data Collection and Preprocessing

Data collection involves acquiring relevant information or samples to train a neural network. This process may involve gathering data from various sources, such as databases, sensors, or online repositories. Data preprocessing steps, including cleaning, normalization, and feature extraction, are applied to ensure the data is in a suitable format for training. This prepares the data for effective learning by the neural network [26].

3.8.3 Training, Validation, and Testing Data Split

To assess the performance and generalization capabilities of a neural network, the available data is typically split into three subsets: training, validation, and testing. The training set is used to optimize the network's parameters, while the validation set helps fine-tune hyperparameters and monitor the model's performance. The testing set provides an unbiased evaluation of the final model's performance on unseen data. Proper data splitting ensures reliable assessment and helps avoid overfitting [26].

3.9 Optimization Techniques and Optimizers

3.9.1 Overview of Optimization Techniques

Optimization techniques play a crucial role in training neural networks, enabling them to converge to optimal solutions efficiently. Several optimization algorithms have been developed to improve the convergence and efficiency of neural network training. Here are some widely used optimizers:

Stochastic Gradient Descent (SGD)

SGD is a popular optimization algorithm that updates the parameters based on the gradients computed from a randomly selected mini-batch of training data. It offers a computationally efficient approach for large-scale training but can be sensitive to the learning rate and may require careful tuning.

Adam Optimizer

Adam (Adaptive Moment Estimation) combines the advantages of AdaGrad and RMSprop optimizers. It maintains adaptive learning rates for different parameters and uses momentum to accelerate convergence. Adam is widely used due to its effectiveness and ease of use, requiring minimal hyperparameter tuning.

Levenberg-Marquardt

Levenberg-Marquardt is an optimization algorithm commonly used in training neural networks with a specific architecture, such as feedforward networks with a single hidden layer. It is based on the Gauss-Newton method and utilizes the Levenberg-Marquardt parameter to balance the trade-off between the gradient descent and Gauss-Newton approaches.

Bayesian Regularization

Bayesian regularization, also known as weight decay or weight regularization, is a technique that introduces a penalty term to the loss function based on the magnitude of the weights. It encourages the model to have smaller weights, preventing overfitting and promoting better generalization to unseen data.

3.10 Conclusion

In conclusion, this chapter has provided an overview of artificial intelligence and the theory of neural networks. Neural networks, inspired by the structure and functionality of biological neurons, have revolutionized the field of artificial intelligence.

The chapter then delved into the building blocks of neural networks, the neurons. It explained the differences between biological neurons and artificial neurons and explored the structure and functionality of artificial neurons. Neurons play a crucial role in neural networks by processing information and making predictions or decisions. They work collectively to capture complex patterns and relationships within the data.

Weights and biases were identified as essential components of neural networks. The chapter emphasized their importance and discussed methods for their initialization and update. Proper initialization and adjustment of weights and biases significantly impact the network's learning process and performance.

The chapter also introduced different types of learning in neural networks, including supervised learning, unsupervised learning, and reinforcement learning. Each type serves different purposes and is applied in various domains. Supervised learning learns from labeled data, unsupervised learning discovers patterns in unlabeled data, and reinforcement learning focuses on learning through interaction with an environment.

Activation functions were discussed as crucial elements in neural networks, introducing non-linearity and enabling complex behaviors. The chapter highlighted the role and importance of activation functions and presented common activation functions such as sigmoid, ReLU, and tanh, along with their properties and applications.

Gradient descent and backpropagation were presented as fundamental techniques in training neural networks. Gradient descent optimizes the network's parameters by minimizing the cost function, and backpropagation efficiently computes the gradients using the chain rule of calculus. These techniques enable the network to learn from data and improve its performance.

In summary, this chapter provided a comprehensive understanding of artificial intelligence, neural networks, neurons, weights and biases, types of learning, activation functions, gradient descent, and backpropagation. These concepts form the foundation of neural networks and are instrumental in their ability to learn, generalize, and make accurate predictions.

Chapter 4

Evaluating Different NN Model Architectures and Parameters for SOC Estimation

4.1 Introduction

This chapter focuses on evaluating various neural network (NN) models for SOC estimation. The chapter introduces the objectives and approach of the study, followed by a description of the experimental setup. It discusses data acquisition, cell specifications, and the dataset used for the study.

The chapter explains the data selection and normalization techniques used, including the application of the Min-Max normalization algorithm and a moving average filter. It then delves into training two types of NN models: a Feed Forward Neural Network (FNN) and a Recurrent Neural Network (RNN). The properties and training functions of each model are described.

4.2 Experimental Setup

- Selection of the input and the output data for the supervised BP learning.
- Normalization of the input and the output data.
- Training of the normalized data using BP learning.
- Testing the goodness of fit of the model.
- Comparing the predicted output with the desired output.

The following Figure summarize the experimental setup leading to the final NN model.

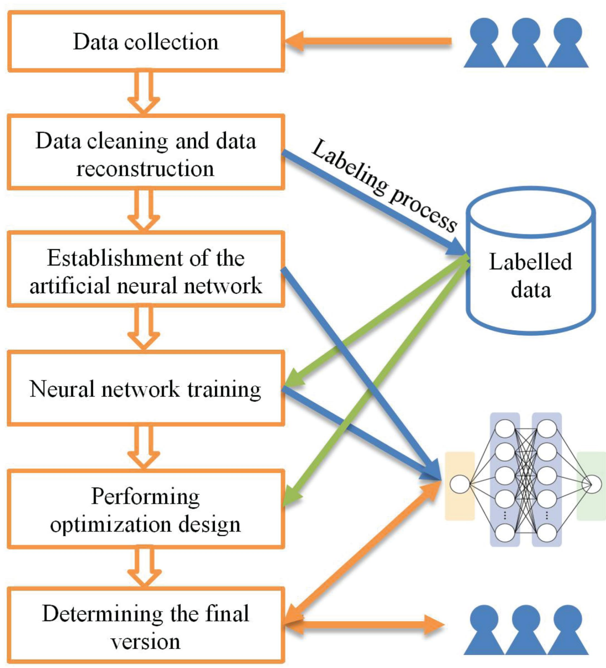

Figure 4.1: Experimental Setup

4.3 data acquisition and cell specification

In [26], an 0.23 m^3 thermal chamber was used to test a brand new LG HG2 cell with a capacity of 3Ah. The testing was conducted using a Digatron Firing Circuits Universal

Battery Tester channel, which had a 75 A, 5 V capacity and a voltage and current accuracy of 0.1% of full scale.

This work involves testing cells using power profiles that were calculated for an electric vehicle undergoing three industry standard drive cycles: US06, UDDS, and LA92. The dataset used in this study was obtained from the LG HG2 lithium cell, which provides information on the measured cell voltage, current, cell temperature, and the amp-hours consumed from the cell. The training data consisted of mixed profiles (1-8), while the testing data were drive cycles. All tests were conducted in a thermal chamber that was set to 25°C.

Table 4.1: Characteristics of LG HG2 Lithium Cell

Capacity	3000 mAh
Chemistry	Lithium-ion
Operating Voltage	2.8 V - 4.2 V
Max Discharge Current	20 A
Energy Density	240 Wh/Kg

4.4 data selection and normalization

The preprocessing steps applied to both the training and testing data will be described in this section. The following steps were performed using Python.

1. The data was combined into a single file to simplify processing.

2. The Voltage, Current, Temperature, and SOC data were extracted from the file and stored in separate lists for further analysis.

3. The Min-Max Normalization algorithm was applied to each list to scale the data between 0 and 1.

4. The normalized lists were saved in CSV format, organized by columns.

To further process the voltage and current data, Matlab was utilized:

5. The voltage, current, and temperature data were selected and organized into a $3 \times Q$ matrix using data import tools.

6. A moving average filter was applied to the voltage and current data to obtain a $5 \times Q$ matrix.

4.4.1 Min Max Normalization

Min-max normalization is a commonly used data preprocessing technique that scales the values of a dataset to a fixed range, typically between 0 and 1. It is useful when the range of values across different features of a dataset is significantly different, as it can improve the performance of machine learning models by reducing the impact of outliers and ensuring that all features are on the same scale. Additionally, it can help with faster convergence during training and avoid issues caused by numerical instability [21].

$$\text{min-max normalization}(x_i) = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

In this equation, x_i represents the i element in the vector x, and $\min(x)$ and $\max(x)$ represent the minimum and maximum values in the vector x, respectively.

4.4.2 Moving Average Filter

In signal processing, the use of a moving average filter (MAF) is a common technique to mitigate noise and remove undesired fluctuations from a signal. This technique involves computing the mean value of a small segment of consecutive data points within the signal and substituting each data point with the computed mean value. This process is repeated iteratively for all the data points, leading to a smoothed signal with reduced noise [27].

The plot of the training data after normalization is represented in figure 4.2.

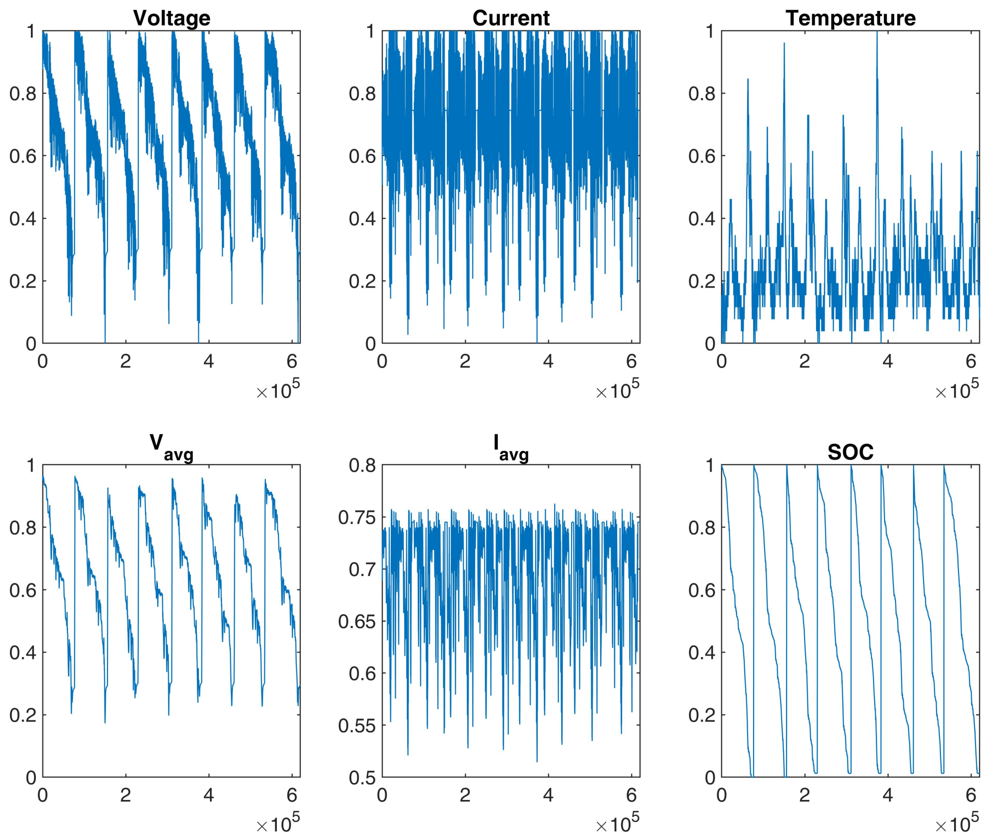

Figure 4.2: Training data plot

4.5 Training The Neural Network Model

In this section, our objective was to determine the optimal architecture and parameters for SOC estimation using neural networks. To achieve this, we experimented with three different NN classifiers: FNN and RNN models implemented in "NFTOOl," and deep learning using the Deep Network Designer in MATLAB.

4.5.1 Feed Forward Neural Network

Using "nntool" – open network/data manager in Matlab, network properties are shown in the following table :

Table 4.2: FNN model properties

Neural Network Properties	
Network Type	Feedforward Backprop
Training Function	Levenberg-Marquardt (trainlm)
Adaption Learning Function	LearnGDM
Performance Function	MSE
Activation Function	Tansig
Number of Epochs	30

The Levenberg-Marquardt "trainlm" training function was used with the Tansig activation function, which gave us the best results in terms of accuracy and the minimum noise. The performance of the neural network was measured using MSE. To obtain the most precise model, an iterative process was followed in order to adjust the number of neurons in the hidden layer. To ensure stability during the training process and avoid over-fitting, a careful examination of both the regression and gradient plots is required. Visualizing these plots allows a fine-tune the model and achieve the highest level of accuracy.

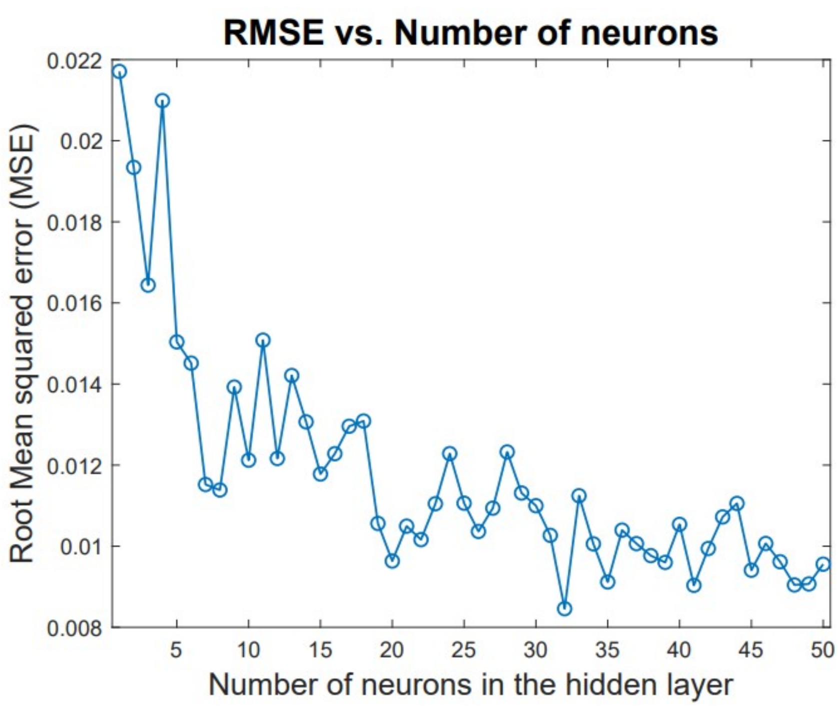

Figure 4.3: RMSE in function of number of neurons

The figure clearly demonstrates that the RMSE attains its minimal values by employing more than 30 neurons in the hidden layer, whereas the model fails to exhibit any significant improvement by utilizing more than 50 neurons.

In the Neural Fitting Tool of MATLAB "NFTOOL", the regression curve is a graphical representation of the fit between the actual and predicted values of the output variable. It is a plot of the predicted values against the actual values, where the ideal curve is a straight line that passes through the origin with a slope of 1. The regression is plotted in Figure 4.4.

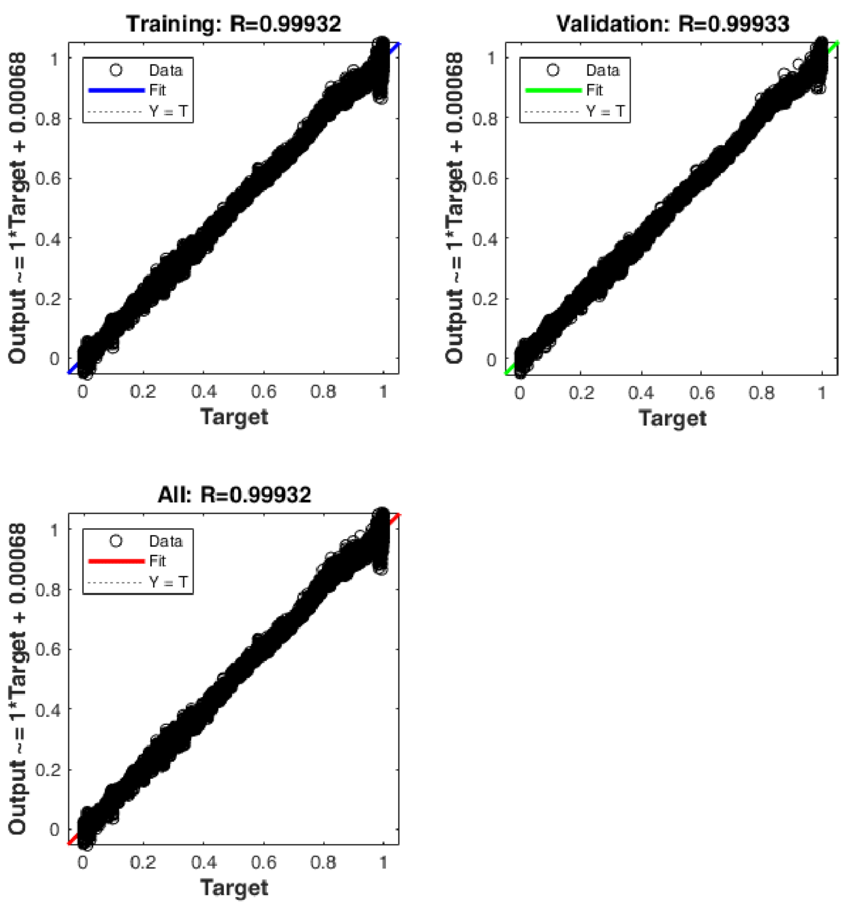

Figure 4.4: Regression Plot

The gradient plot is a visual representation of the magnitude of the gradients of the weights with respect to the error function during the training process. The plot shows the average gradient magnitude for each layer of the network as a function of training time or epoch. The gradient plot is useful for monitoring the progress of training and diagnosing potential issues, such as slow convergence or oscillations in the weights.

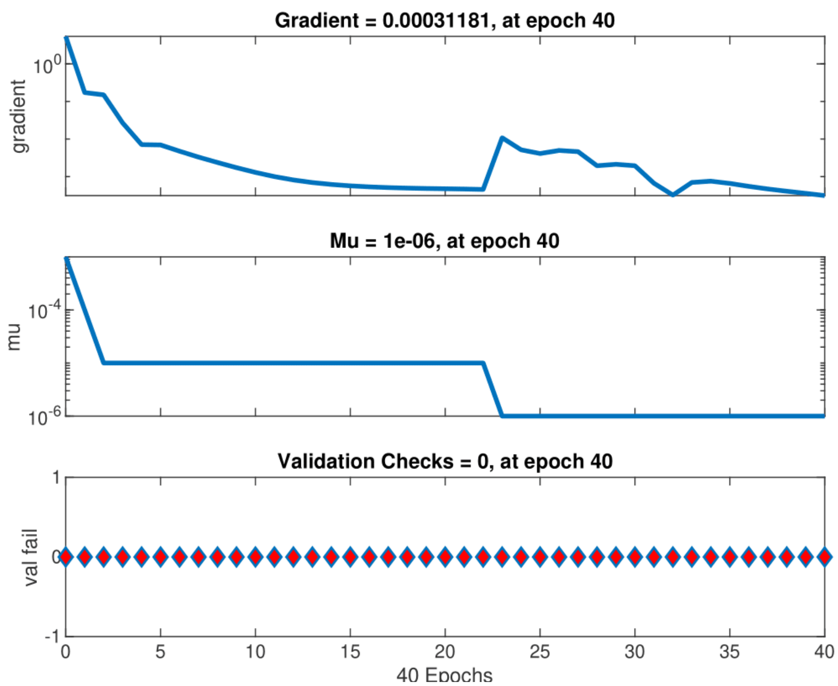

Figure 4.5: Training State

Upon analyzing 4.5, we observed a steady decrease in the plot, indicating that the weights are adjusting significantly to improve the training's fit. However, at epoch 22, there was a rise in the gradient plot due to the decrease in the learning rate.

Figure 4.6 represents a comparison graph between the actual SOC and the predicted SOC

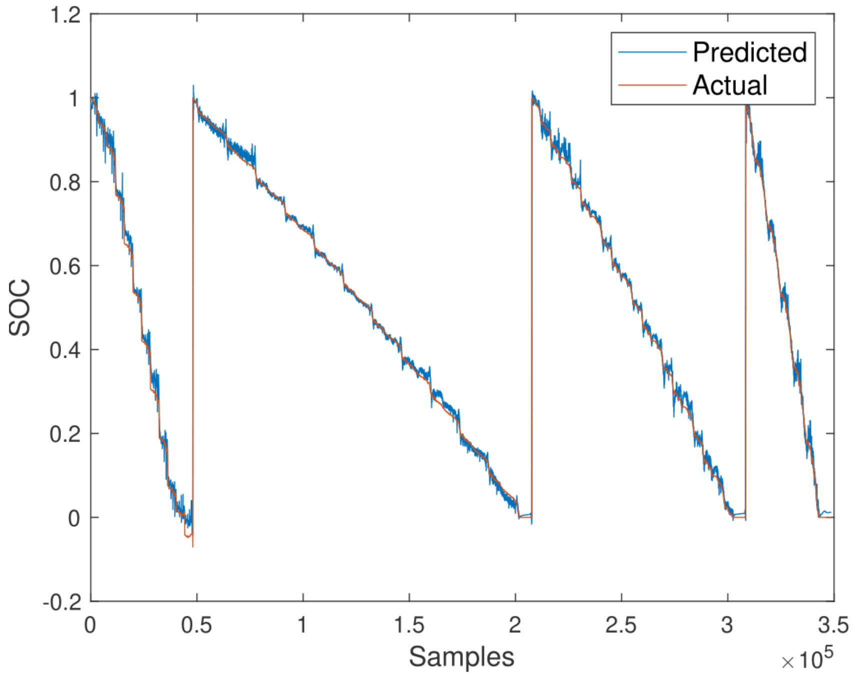

Figure 4.6: Estimated Values vs real values

Based on 4.6, we obtained fairly good results with a high degree of accuracy, as indicated by the MAE value of 0.0095. However, the graph also shows some noise that can be reduced by applying a MAF.

4.5.2 Recurrent Neural Network

Recurrent Neural Networks are a type of artificial neural network that are well-suited for processing sequential data, such as time-series data or natural language.

The key feature of RNN is that they allow for the processing of sequences of inputs by maintaining an internal state that is updated at each time step and serves as a memory of the past inputs. This makes them particularly useful for tasks such as sequence prediction, where the goal is to predict the next element in a sequence based on previous elements.

In more detail, the internal state of an RNN is a vector of values that represents the network's "memory" of the previous inputs. At each time step, the RNN takes in an input vector and the current state vector, and computes a new state vector and an output vector. The output vector can be used to make a prediction or classification based on the current input, while the state vector is used to keep track of the context of the previous inputs.

The state vector is updated based on a combination of the current input and the previous state vector. This update is governed by a set of weights and biases that are learned during training [2]. The following image represents the architecture of RNN in Matlab toolbox .

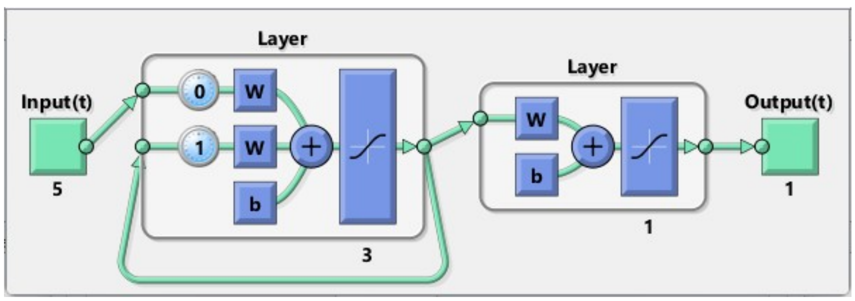

Figure 4.7: Architecture of RNN with 3 neurons in hidden layer

Table 4.3 the MAE of the testing data as a function of the number of neurons in the hidden layer of a RNN model, which was trained on the previous data-set to find the optimal number of neurons in the hidden layer.

Table 4.3: MAE vs Number of Neurons in RNN model

Number of Neurons	MAE
1	0.0291
2	0.0128
3	0.0131
5	0.0120
10	0.0117
20	0.0112

The RNN model uses the previous estimated sample of State of Charge as input in the hidden layer to predict the actual SOC. The table of MAE values as a function of the number of neurons in the hidden layer shows that there is no significant impact on the MAE when using 2 or more neurons. This indicates that the RNN model can effectively predict SOC without requiring a large number of neurons in the hidden layer. The training state of a RNN model with 5 neurons is shown in 4.9.

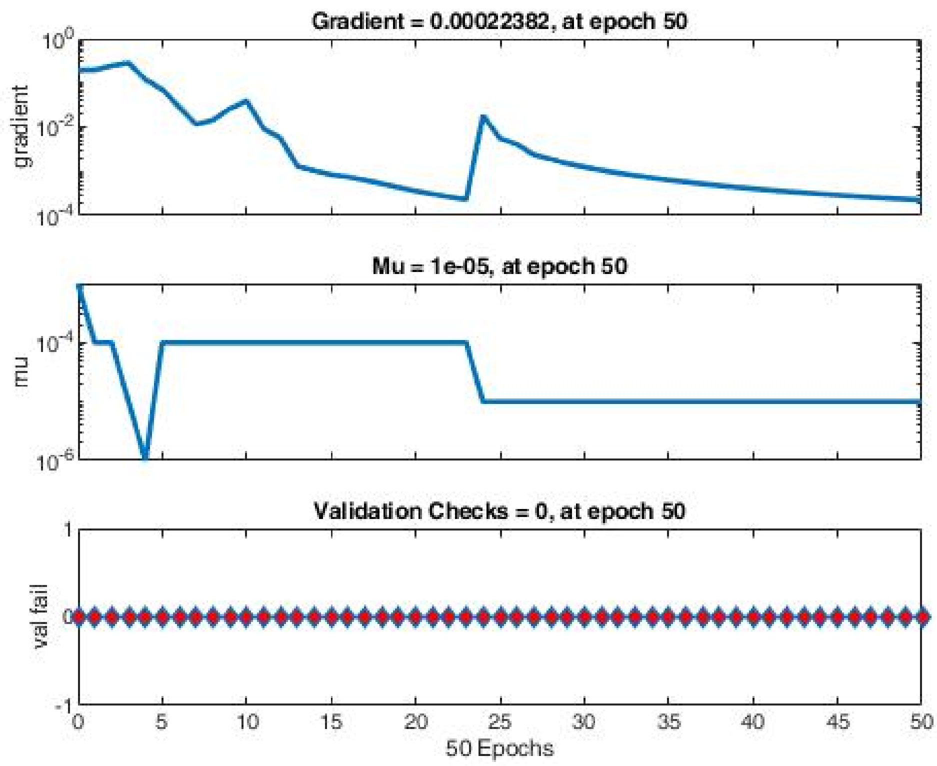

Figure 4.8: Training State

In addition figure 4.9 represents a comparison graph between the actual SOC and the predicted SOC .

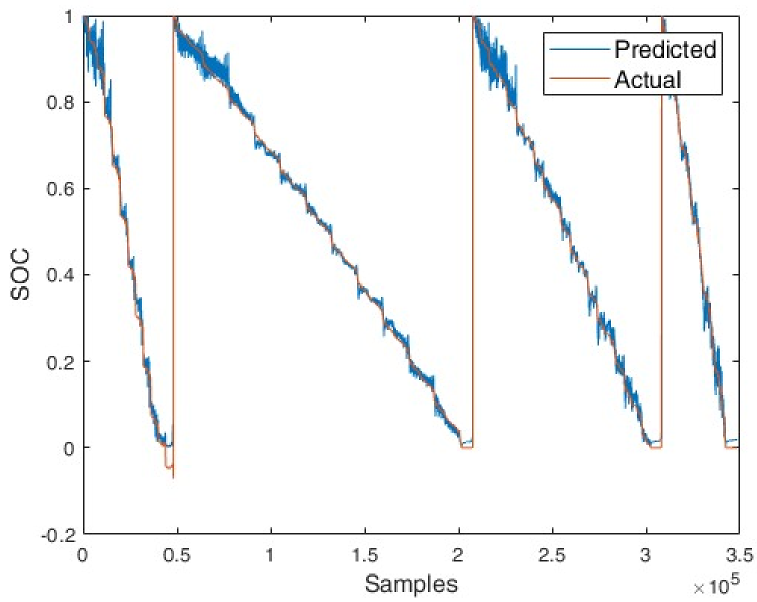

Figure 4.9: Estimated Values vs real values of 5 neurons RNN model

According to the above graph, the results obtained were quite satisfactory with a high level of accuracy, which is reflected in the low MAE value of 0.012. However, the graph also exhibits some noise, especially from 1 to 0.8 comparing to the results from FFNN model, which could be also reduced by applying MAF.

4.5.3 Deep Neural Network

Traditionally, machine learning techniques used one or two layers of non-linear and linear transformations to extract features from input data. However, as computational power increased and more data became available, researchers began investigating deeper architectures. Deep learning refers to the use of neural networks with multiple layers of non-linear transformations to learn representations of data. By adding more layers to a neural network, the network can learn increasingly complex and abstract features. This can lead to significant improvements in performance for many applications, including image and speech recognition, natural language processing, and game playing [28].

In this Part we are going to try the model proposed by [26], using Matlab Deep

Learning toolbox, figure 4.10, 4.11, 4.12, represents respectively the architecture of our model, training performance, and testing plot.

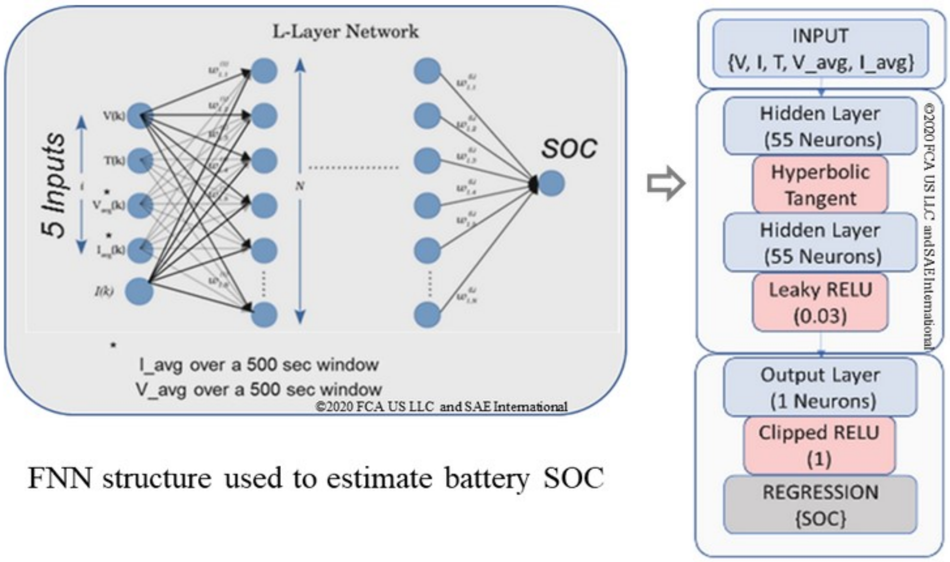

Figure 4.10: Deep learning model architecture

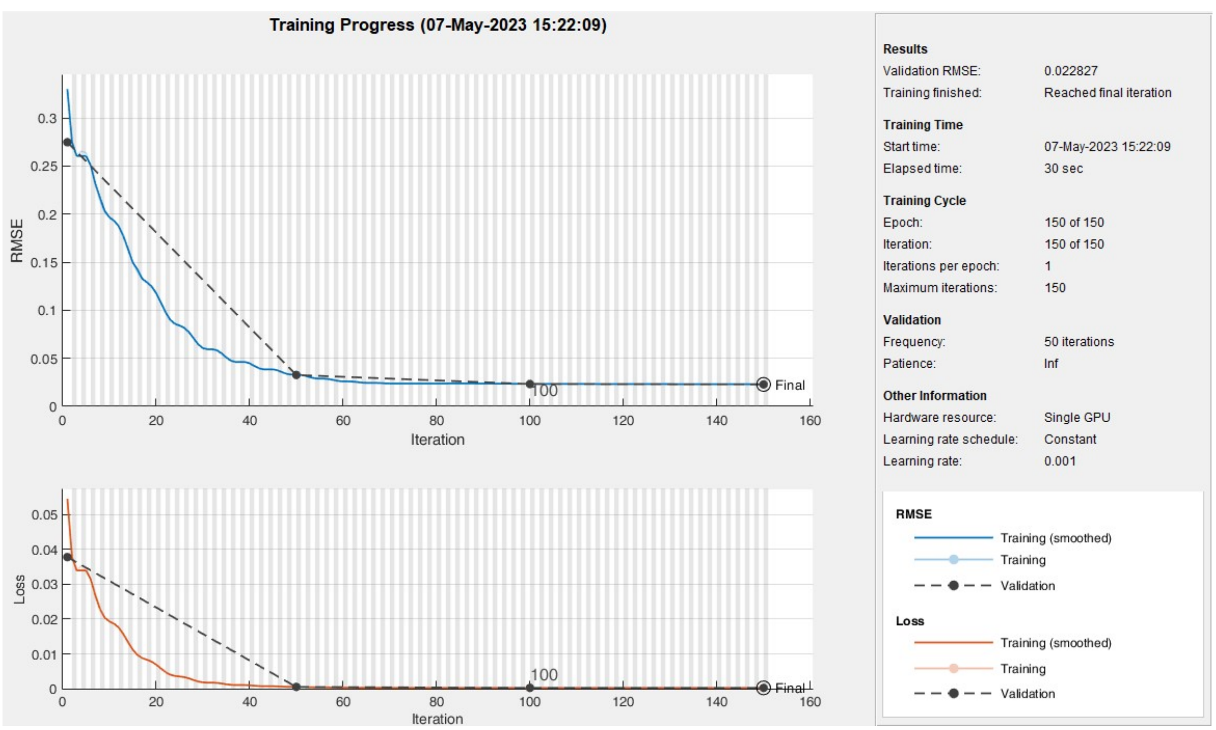

Figure 4.11: Performance curve of deep learning model

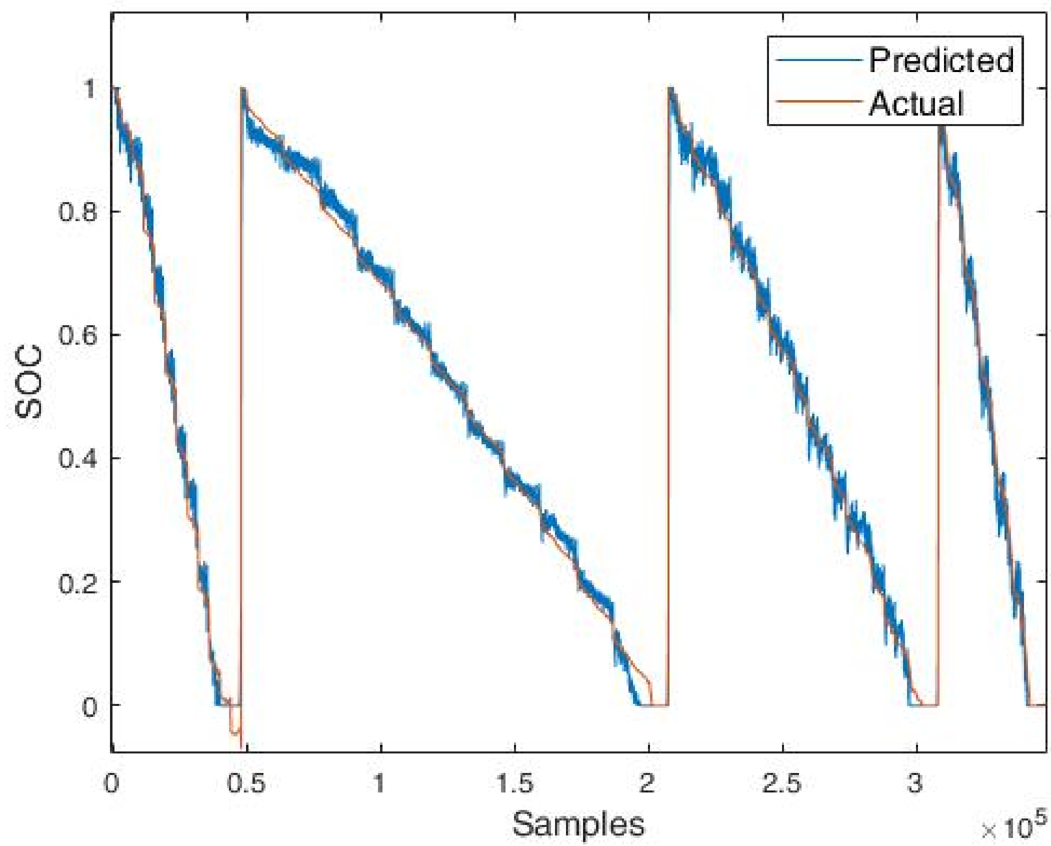

Figure 4.12: Predicted values using deep learning model

4.6 Exploring Different Approaches

4.6.1 Temperature's Impact on Model Performance

The suggestion was made to remove temperature from our dataset in order to observe the effect of temperature on this cell measurement. Theoretically, the variance of temperature between 23 and 27 degrees is insignificant. However, by testing this theory on our model, it will be proven that the features of our lithium cell have been learned. The same architecture of the feed-forward example was utilized, and the following results were obtained.

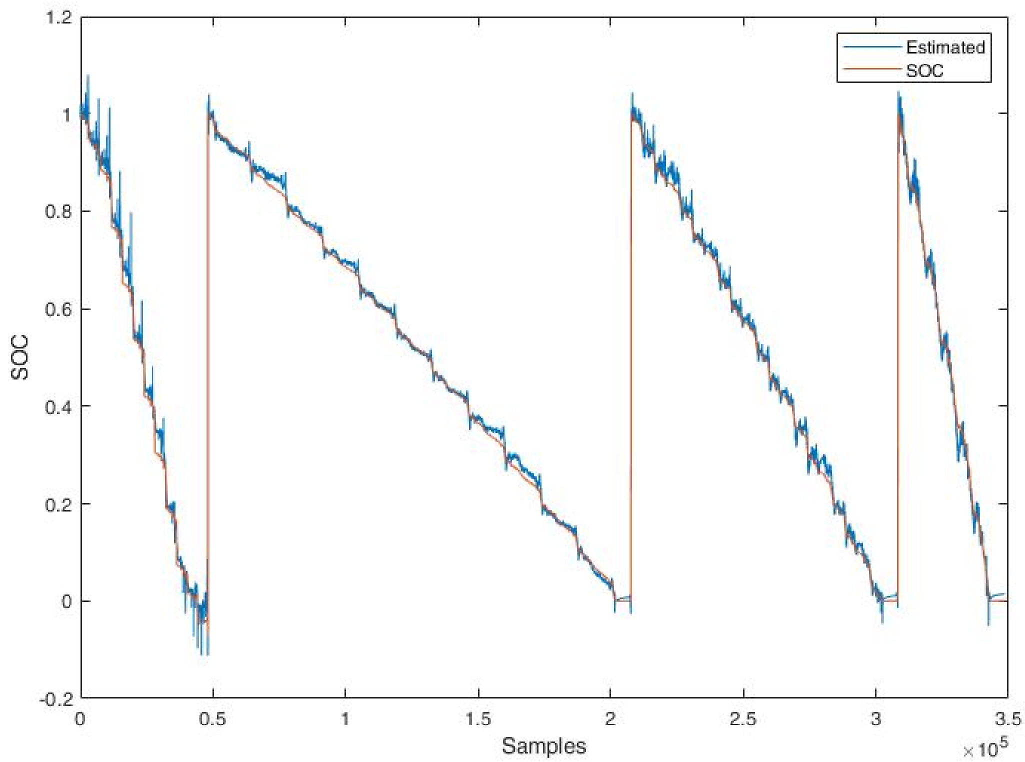

Figure 4.13: Results of the FNN model without temperature

the error between predicted and real values can be ploted in the following figure where the MAE is 0.0103 :

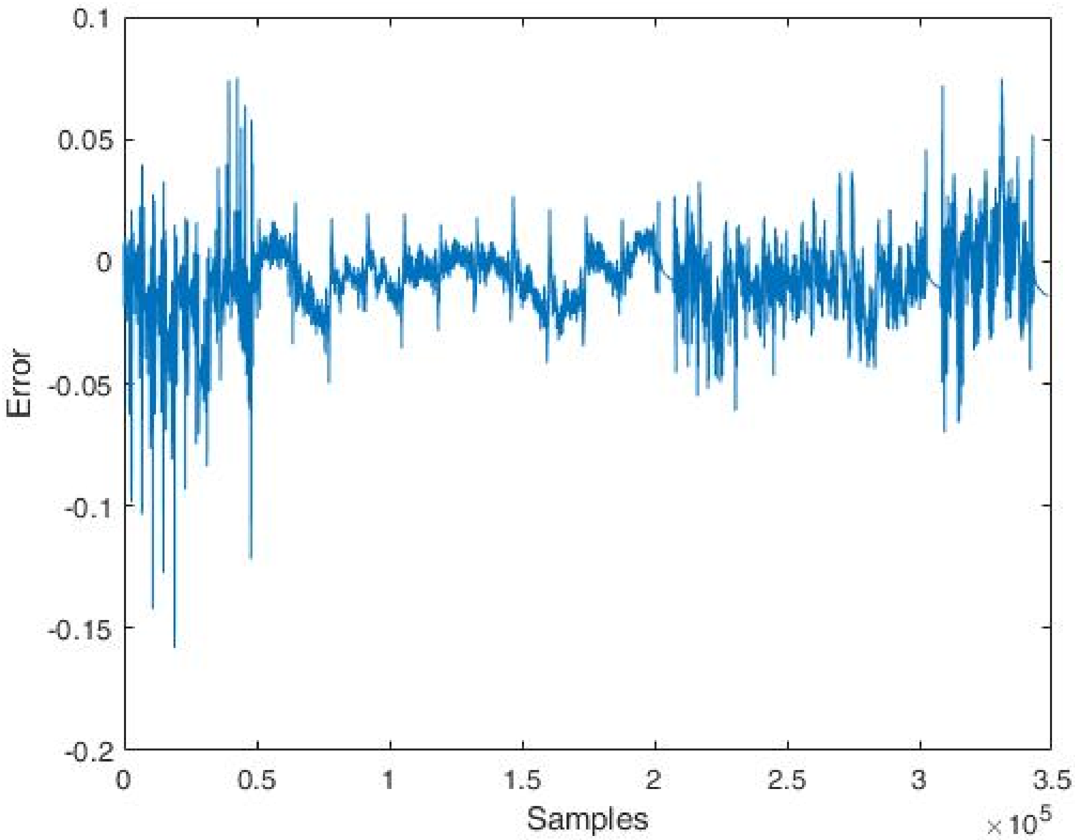

Figure 4.14: MAE error for no temperature FNN model

By comparing the testing plots of the FFNN model with and without temperature, it can be inferred that the temperature variation is negligible in SOC prediction, indicating that there are no features to be extracted from the temperature range of 23 to 27 degrees.

4.6.2 Effect of $V_{\mathbf{avg}}$ and $I_{\mathbf{avg}}$ on Model Performance

The data input selection was based on the behavior of the lithium cell and its dependencies for SOC estimation. Additionally, the inclusion of average values of voltage and current was inspired by the work of [26], where the Principal Component Analysis (PCA) algorithm was implemented.

PCA is a versatile and powerful method that provides a comprehensive understanding of complex multivariate data. It enables the identification of relationships between variables and samples, detection of outliers, pattern identification, generation of new hypotheses, and various other analytical tasks.

This attempt highlights the significance of data analysis in selecting the appropriate data inputs for the neural network model. Three different data input selections were explored, namely Voltage, Current, and Temperature. Figure 4.15 illustrates the predicted values of the testing data for the FFNN model without average inputs, while Figure 4.16 displays the MAE between the actual and predicted SOC values.

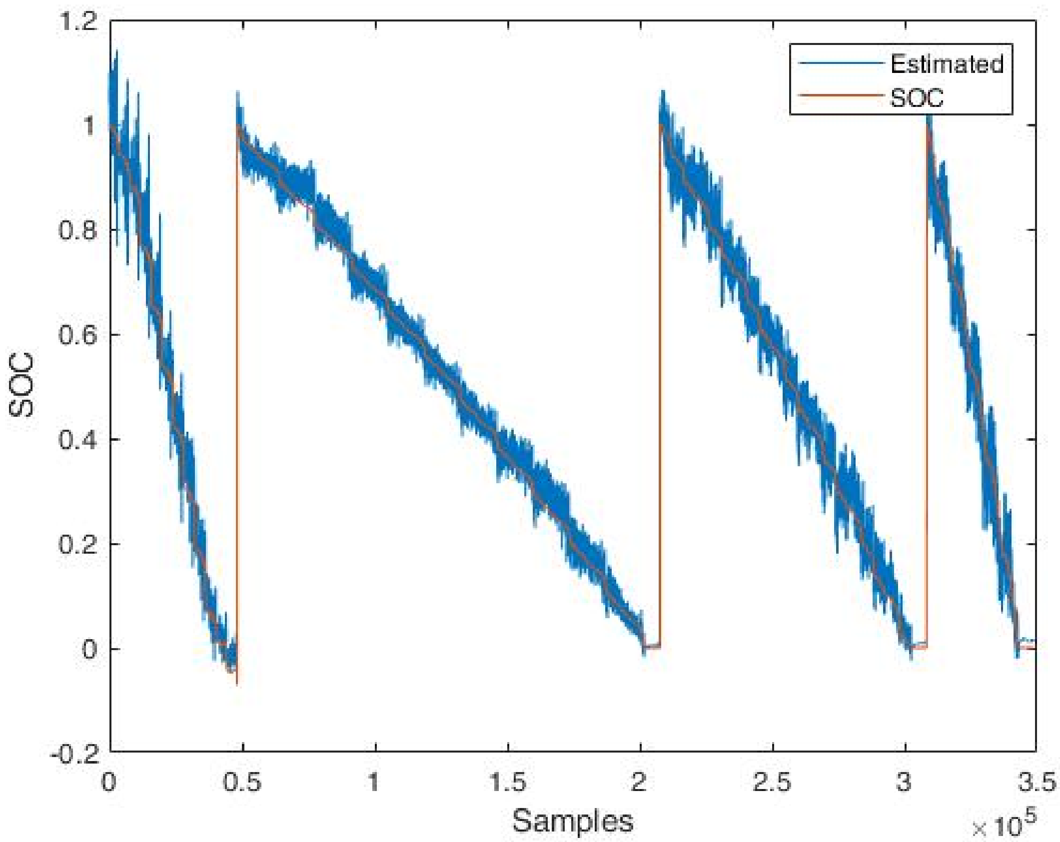

Figure 4.15: FNN model results without V_{avg} and I_{avg} inputs

the error between predicted and real values can be ploted in the following figure where the MAE is 0.019 :

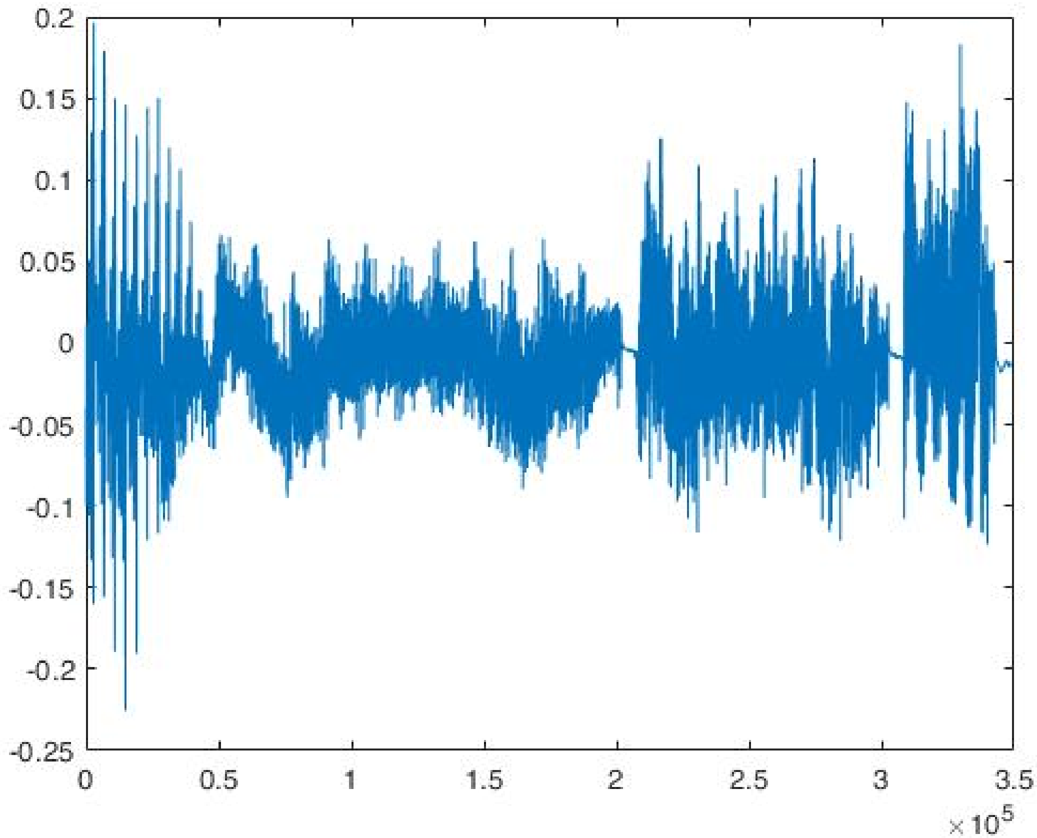

Figure 4.16: MAE for the FNN model without average inputs

The plot of predicted values exhibits significant noise compared to 4.6, highlighting the fundamental role of data analysis and preprocessing in data input selection, which can be seen as the fuel for NN.

4.6.3 Noise Elimination Using Moving Average Filter

After examining the earlier findings, it is observed that the curve representing the predicted values of SOC exhibits some level of noise. To address this issue, a potential solution involves implementing MAF to mitigate the impact of minor SOC fluctuations during the discharge process of the lithium cell, as facilitated by the battery management system. Figure 4.17 illustrates the filtered predicted values for 4.6,and 4.18 shows the error between filtered and original SOC, showcasing the effectiveness of this approach in reducing noise.

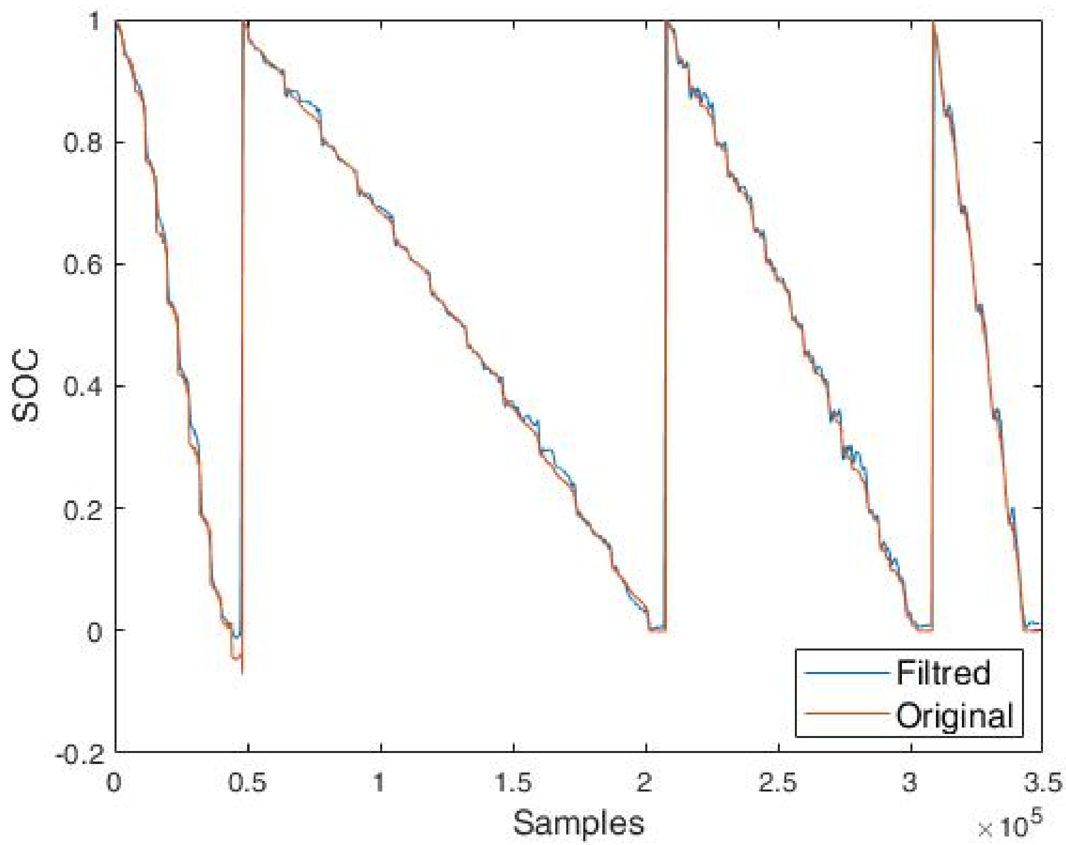

Figure 4.17: Filtered values using MAF

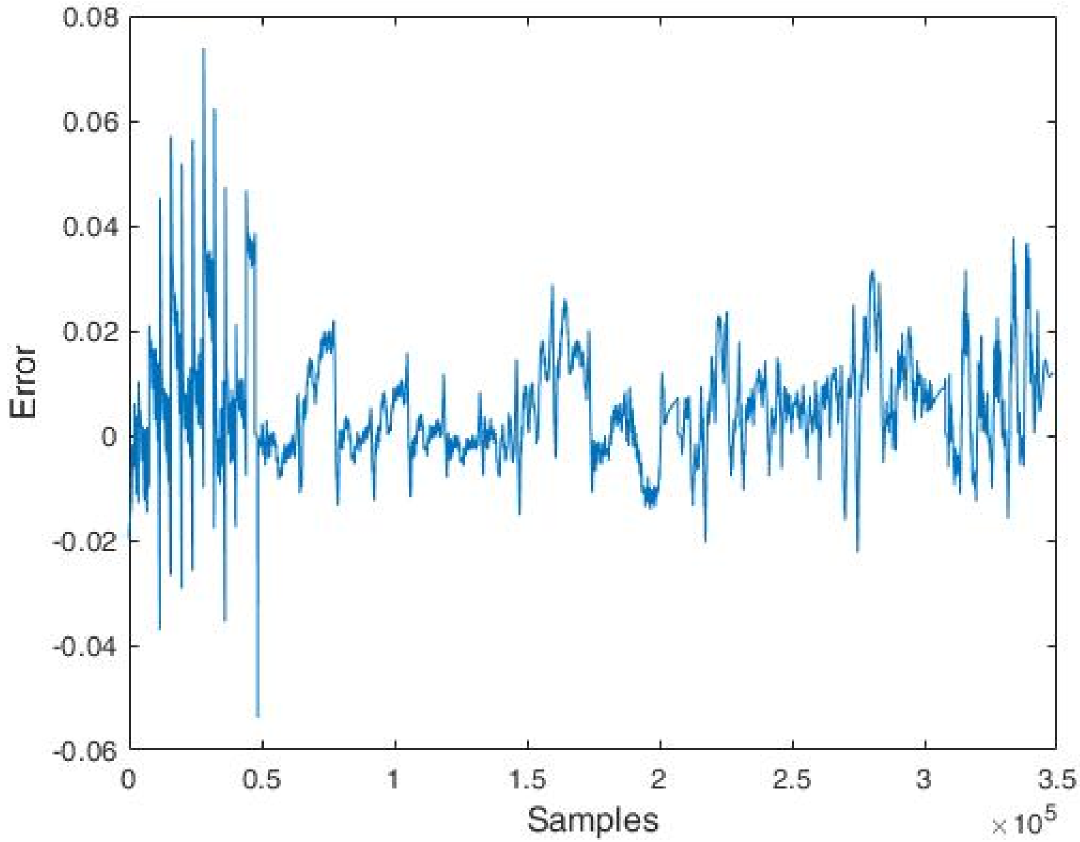

Figure 4.18: Error between Filtered SOC values and actual SOC

4.7 Conclusion

In this chapter, an evaluation of different neural network model architectures and parameters for state of charge estimation has been made. The experimental setup involved the selection of input and output data for supervised backpropagation learning, normalization of the data, training of the normalized data, and testing the model's goodness of fit.

The data acquisition and cell specification section described the testing setup using a thermal chamber and a Digatron Firing Circuits Universal Battery Tester. The dataset used in the study was obtained from an LG HG2 lithium cell, which provided information on measured cell voltage, current, temperature, and coulombs consumed. The training data consisted of mixed profiles, while the testing data consisted of drive cycles.

Data selection and normalization were performed using Python and Matlab. The data was combined into a single file, and the voltage, current, temperature, and SOC data were extracted and normalized using the min-max normalization algorithm. A moving average filter was applied to the voltage and current data.

Three different NN classifiers: feedforward neural network, recurrent neural network, and deep neural network have been trained. For the FNN model, the number of neurons in the hidden layer was adjusted iteratively to achieve the highest accuracy. The Levenberg-Marquardt training function and Tansig activation function were used. The RNN model was trained to predict SOC using the previous estimated sample as input in the hidden layer. The DNN model was trained using the Deep Network Designer in MATLAB.

The results showed that both the FNN and RNN models achieved good accuracy in predicting SOC, with low mean absolute error values. The deep NN model also showed promising results, with a well-designed architecture and good performance on the testing data.

In conclusion, this chapter demonstrated the effectiveness of different NN model architectures and parameters for SOC estimation. The FNN, RNN, and deep NN models all showed promising results, with the ability to accurately predict SOC based on input data.

General conclusion and prospects

In general, this paper provides a comprehensive overview of rechargeable batteries, state of charge estimation methods, artificial intelligence, and neural networks. Each chapter focuses on specific aspects and contributes to a deeper understanding of these topics.

The first chapter gives an introduction to rechargeable batteries, with a particular emphasis on lithium-ion batteries. It discusses various chemistries, their characteristics, and applications. The chapter also highlights important battery characteristics and the challenge of accurately estimating the state of charge.

The second chapter focuses on SOC estimation methods for lithium batteries. It covers direct measurement methods and bookkeeping methods such as Coulomb counting. It also explores adaptive systems like neural networks and hybrid methods. The chapter provides valuable insights into different SOC estimation techniques and their suitability for battery systems.

The third chapter delves into the theory of neural networks and their role in artificial intelligence. It explains the structure and functionality of artificial neurons, the importance of weights and biases, and different types of learning. The chapter also introduces activation functions, gradient descent, and backpropagation, which are fundamental techniques in training neural networks.

The fourth chapter evaluates different neural network model architectures and parameters for SOC estimation. It describes the experimental setup, data acquisition, and cell specifications. The chapter presents the training and testing process using various neural network models, including feedforward neural networks, recurrent neural networks, and deep neural networks. The results demonstrate the effectiveness of these models in accurately predicting SOC.

We are satisfied with our work but aim to further enhance the system. Here are some ideas and perspectives we have put forward to improve the system:

- Apply Kalman Filter instead of MAF : Investigate the use of Kalman Filter as an alternative to MAF for SOC estimation. The Kalman Filter is an optimal recursive estimator that can be combined with model's predicted values. Assess the performance of the Kalman Filter in terms of SOC estimation accuracy and compare it with MAF and other existing techniques.

- Online Adaptation and Learning : Develop adaptive SOC estimation methods that can continuously learn and adapt to changing battery conditions. This can involve real-time monitoring and adjustment of model parameters, allowing for improved SOC estimation during different battery operating conditions.

- Validation and Verification Studies : Conduct extensive validation and verification studies to assess the performance of the developed SOC estimation models in real-world applications. This includes testing the models with diverse battery chemistries, operating conditions, and environmental factors to ensure their robustness and reliability.

- Integration with BMS : Explore the integration of advanced SOC estimation techniques into battery management systems. This can involve developing algorithms that can provide real-time SOC updates, enable better battery monitoring, and optimize battery usage and performance.

Bibliography

[1] B. University, *BU-205: Types of Lithium-ion*, Accessed on 10 April 2023, 2019. [Online]. Available: https://batteryuniversity.com/article/bu-205-types-of-lithium-ion.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016, http://www.deeplearningbook.org.

[3] A. M. S. M. H. S. Attanayaka, J. P. Karunadasa, and K. T. M. U. Hemapala, "Estimation of state of charge for lithium-ion batteries - a review," *AIMS Energy*, vol. 7, no. 2, pp. 186–210, 2019. DOI: 10.3934/energy.2019.2.186.

[4] R. M. Salgado, F. Danzi, J. E. Oliveira, A. El-Azab, P. P. Camanho, and M. H. Braga, "The latest trends in electric vehicles batteries," *Molecules*, vol. 26, no. 11, p. 3188, 2021.

[5] F. Elmahdi, L. Ismail, and M. Noureddine, "Fitting the ocv-soc relationship of a battery lithium-ion using genetic algorithm method," in *E3S Web of Conferences*, EDP Sciences, vol. 234, 2021, p. 00 097.

[6] R. Tsiriry, "Etude de l'amelioration de la longevite de la performance des batteries lithium-ion pour voitures electriques," Ph.D. dissertation, Université d'Antananarivo, 2017.

[7] Y. Xing, W. He, M. Pecht, and K. L. Tsui, "State of charge estimation of lithium-ion batteries using the open-circuit voltage at various ambient temperatures," *Applied Energy*, vol. 113, pp. 106–115, 2014.

[8] W.-Y. Chang, "The state of charge estimating methods for battery: A review," *International Scholarly Research Notices*, vol. 2013, 2013.

[9] D. Kim, K. Koo, J. J. Jeong, T. Goh, and S. W. Kim, "Second-order discrete-time sliding mode observer for state of charge determination based on a dynamic resistance li-ion battery model," *Energies*, vol. 6, no. 10, pp. 5538–5551, 2013.

[10] S. Lee, J. Kim, J. Lee, and B.-H. Cho, "State-of-charge and capacity estimation of lithium-ion battery using a new open-circuit voltage versus state-of-charge," *Journal of Power Sources*, vol. 185, no. 2, pp. 1367–1373, 2008.

[11] A. Sidhu, A. Izadian, and S. Anwar, "Adaptive nonlinear model-based fault diagnosis of li-ion batteries," *IEEE Transactions on Industrial Electronics*, vol. 62, pp. 1002–1011, Feb. 2015. DOI: 10.1109/TIE.2014.2336599.

[12] H. Anbuky and P. Pascoe, "Vrla battery state-of-charge estimation in telecommunication power systems," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 3, pp. 565–573, 2000.

[13] K. Bundy, M. Karlsson, G. Lindbergh, and A. Lundqvist, "An electrochemical impedance spectroscopy method for prediction of the state of charge of a nickel-metal hydride battery at open circuit and during discharge," *Journal of Power Sources*, vol. 72, no. 2, pp. 118–125, 1998.

[14] K. Movassagh, A. Raihan, B. Balasingam, and K. Pattipati, "A critical look at coulomb counting approach for state of charge estimation in batteries," *Energies*, vol. 14, no. 14, p. 4074, 2021.

[15] J. Bi, S. Shao, W. Guan, and L. Wang, "State of charge estimation of li-ion batteries in an electric vehicle based on a radial-basis-function neural network," *Chinese Physics B*, vol. 21, no. 11, p. 118 801, 2012.

[16] J. Á. Antón, P. G. Nieto, F. de Cos Juez, F. S. Lasheras, M. G. Vega, and M. R. Gutiérrez, "Battery state-of-charge estimator using the svm technique," *Applied Mathematical Modelling*, vol. 37, no. 9, pp. 6244–6253, 2013.

[17] A. A. Chellal, "Energy management of li-po batteries in the mobile robotics domain," Ph.D. dissertation, Instituto Politécnico de Bragança, Portugal, 2021.

[18] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *The bulletin of mathematical biophysics*, vol. 5, pp. 115–133, 1943.

[19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[20] A. Almaghrebi, F. Aljuheshi, M. Rafaie, K. James, and M. Alahmad, "Data-driven charging demand prediction at public charging stations using supervised machine learning regression methods," *Energies*, vol. 13, no. 16, p. 4231, 2020.

[21] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013. DOI: `10.1109/TPAMI.2013.50`.

[22] R. I. Mukhamediev, A. Symagulov, Y. Kuchin, K. Yakunin, and M. Yelis, "From classical machine learning to deep neural networks: A simplified scientometric review," *Applied Sciences*, vol. 11, no. 12, p. 5541, 2021.

[23] J. Feng, X. He, Q. Teng, C. Ren, H. Chen, and Y. Li, "Reconstruction of porous media from extremely limited information using conditional generative adversarial networks," *Physical Review E*, vol. 100, no. 3, p. 033 308, 2019.

[24] JavaTpoint, *Gradient descent in machine learning*, `https://www.javatpoint.com/gradient-descent-in-machine-learning`.

[25] S. Lau, *Learning rate schedules and adaptive learning rate methods for deep learning*, `https://towardsdatascience.com/learning-rate-schedules-and-adaptive-learning-rate-methods-for-deep-learning-2c8f433990d1`, 2017.

[26] P. Kollmeyer, C. Vidal, M. Naguib, and M. Skells, *LG 18650HG2 Li-ion Battery Data and Example Deep Neural Network xEV SOC Estimator Script*, version V3, 2020. DOI: `10.17632/cp3473x7xv.3`.

[27] J. G. Proakis and D. G. Manolakis, *Digital signal processing: Principles, algorithms, and edition*, 1995.

[28] E. Chemali, P. J. Kollmeyer, M. Preindl, and A. Emadi, "State-of-charge estimation of li-ion batteries using deep neural networks: A machine learning approach," *Journal of Power Sources*, vol. 400, pp. 242–255, 2018.