



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

Doctor of Philosophy

OPTIMIZATION OF SELECTIVE LASER MELTING
PROCESS PARAMETERS FOR Ti6Al4V PARTS
USING DEEP LEARNING

The Graduate School

of the University of Ulsan

Department of Mechanical and Automotive Engineering

NGUYEN DINH SON

OPTIMIZATION OF SELECTIVE LASER MELTING
PROCESS PARAMETERS FOR Ti6Al4V PARTS
USING DEEP LEARNING

Supervisor: Professor CHANG MYUNG LEE

A Dissertation

Submitted to

The Graduate School of the University of Ulsan

In partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

NGUYEN DINH SON

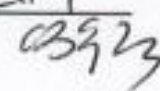
Department of Mechanical and Automotive Engineering

University of Ulsan, Korea


November 2019

OPTIMIZATION OF SELECTIVE LASER MELTING
PROCESS PARAMETERS FOR Ti6Al4V PARTS
USING DEEP LEARNING

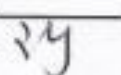
This certifies that the dissertation thesis of NGUYEN DINH SON is approved

Oak Taek LIM
Committee Chair Dr. 

Doo-Man Chun
Committee Member Dr. 

Changmyung Lee
Committee Member Dr. 

Wookjin Lee
Committee Member Dr. *Wookjin Lee*

Taeha Ha
Committee Member Dr. 

Department of Mechanical and Automotive Engineering

University of Ulsan, Korea

November 2019

OPTIMIZATION OF SELECTIVE LASER MELTING
PROCESS PARAMETERS FOR Ti6Al4V PARTS
USING DEEP LEARNING

지도교수 이장명

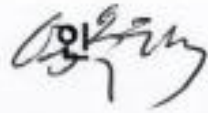
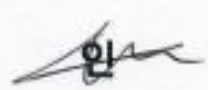

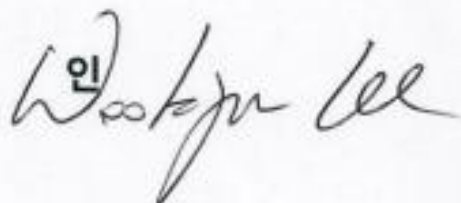

이논문을 공학박사 학위 논문으로 제출함

2019년 11월

울산대학교 대학원

기계자동차공학과

NGUYEN DINH SON 의 공학박사 학위 논문을 인준함

심사위원	심 옥택	
심사위원	천 주만	
심사위원	이 장명	
심사위원	이 옥진	
심사위원	하 태호	

울산대학교 대학원

2019 년 11 월

Acknowledgements

During the time in which this research was undertaken, I have had the best opportunity to work with excellent people. The thesis would not have been finished without their encouragement and kind help.

First and foremost, I would like to express my sincere gratitude to my advisors, Professor Chang Myung Lee and Professor Hong Seok Park for their support, motivation, and immense knowledge impartation during my study in the University of Ulsan. Their guidance helped me through the time of my research and writing of this thesis.

I would also love to thank Professor Ock Taeck Lim, Professor Doo Man Chun, Dr. Wook Jin Lee, Dr. Taeho Ha, for their insightful comments and encouragement, which enhance my research.

Last but not least, I would like to thank my laboratory members, my family, and friends. Their supports and encouragement became the infinite power for me to face difficulties during my study abroad.

Ulsan November 2019

Nguyen Dinh Son

OPTIMIZATION OF SELECTIVE LASER MELTING

PROCESS PARAMETERS FOR Ti6Al4V PARTS

USING DEEP LEARNING

Abstract

This thesis is a study geared towards optimizing Selective Laser Melting (SLM) crucial process parameters for manufacturing medical application parts based on deep learning. Titanium alloys, particularly Ti-6Al-4V ELI, are among the most commonly used materials in the medical industry, due to their superior biocompatibility resistance and specific strength. However, implant components are often a complex geometric shape that is difficultly fabricated by conventional methods such as formative and subtractive techniques. Additionally, that Titanium alloys are expensive materials leads to an increase in manufacturing cost because of waste material. Selective laser melting is an Additive Manufacturing method to fabricate complicated three-dimensional products, by solidifying successive layers with a layer of powder materials, based on a CAD model. The SLM is associated with full melting of powder material. There are many advantages achieved by SLM method, including high density and strength of printed part; negligible waste material by recycling the unmelted powder; printing complex shapes that the traditional manufacturing methods could not; no limitation during the design of a CAD model; customized products, among others. Despite the advantages of SLM technology, printing a product with properties satisfy a user requirement is difficult due to many considerations. There are many process parameters that affect the SLM processing and printed product, such as laser properties, material characteristics, printing conditions, machine properties. Additionally, the range value to modify these parameters are broad. So much time, expenditure and expert knowledge about the process and materials are therefore needed to operate the SLM, in order to achieve a desirable printed product. Thus, developing an optimization process parameters system for SLM processing is significantly necessary. Many optimizing methods have been researched, namely Taguchi, design of experiment (DOE), response surface method (RSM), genetic algorithm, and fuzzy method. However, they have

investigated individual parameters only, and the accuracy of these methods is affected strongly by non-linear problems.

In this thesis, deep learning that efficiently solves a non-linear issue was applied. Four significant essential process parameters, namely laser power, laser velocity, layer thickness and hatch distance, were investigated. Density and surface roughness of the product printed by the SLM technique were considered simultaneously. Density is an essential element of osteogenesis in implants, while surface roughness affects mechanical stability, osseointegration capabilities, interlocking due to bone growth, improve implant anchorage in a bone. Firstly, a deep neural network with backpropagation was applied, and a supervised learning algorithm was used for network learning. An adaptive momentum algorithm was utilized to solve some unwanted problems during the training of a neural network model, namely speed up training speed, convergence, and escape the local minimum. Regularization methods, such as weight decay and dropout, were handled by the cost function to avoid over-training of the neural network model. The deep neural network after building and validation was used to predict the properties of printed product from a given data set. A selection system was built to choose the optimal process parameters, of which properties of product satisfy a specific user requirement which are the inputs of the system. Due to the optimization system, an SLM operator does not need to be an experienced person, neither need to spend much time on the testing step to print a desirable product. The system is a significant relevant module for pre-processing of the SLM printing. Through this thesis, related literature will be introduced firstly. The influences of process parameters on the SLM process and printed product qualities are presented. Development of a deep neural network for predicting follows. Subsequent section is building an optimization system with a graphical user interface. The final part presents a case study used to verify the developed optimization system.

Table of Contents

Contents

Acknowledgements	5
Abstract.....	7
Table of Contents	9
Index of figures	12
Index of tables.....	16
Chapter 1. GENERAL INTRODUCTION	17
1.1 Motivation.....	17
1.2 State of the art	19
1.2.1 Additive Manufacturing	19
1.2.2 Metal Additive Manufacturing.....	24
1.2.3 Titanium Ti-6Al-4V alloy.....	31
1.2.4 Deep learning network.....	33
Chapter 2. METHODOLOGY	37
2.1 Material.....	37
2.2 Experiment.....	37
2.3 Data collection	41
2.3.1 Surface roughness measurement	41
2.3.2 Density measurement	43

Chapter 3. EFFECTS OF PROCESS PARAMETERS ON THE PRINTED PRODUCT QUALITIES	47
3.1. Consolidation behaviour during SLM.....	47
3.2. Penetration.....	48
3.3. Material ejection	50
3.4. Balling phenomenon	51
Chapter 4. DESIGNING OF THE DEEP NEURAL NETWORK.....	57
4.1. Artificial neuron mathematics	57
4.2. Data pre-processing	60
4.3. Deep neural network architecture.....	61
4.4. Activation functions.....	62
4.5. Cost function	63
4.6. Training a deep neural network.....	64
4.6.1. Deep neural network parameters initialization	64
4.6.2. Training algorithm.....	64
4.7. Neural network regularization	76
4.7.1. Weight decay regularization	76
4.7.2. Dropout.....	77
4.8. Deep neural network validation.....	79
Chapter 5. DESIGNING OF THE OPTIMIZATION SYSTEM	83
5.1. Generating data for prediction.....	83

5.2.	Optimization system algorithm	83
5.2.1.	Filtering condition	84
5.3.	Case study.....	87
5.4.	Implementation of the system.....	89
5.5.	Aspect benefits	90
Chapter 6.	CONCLUSION AND FUTURE WORK.....	91
References.....		95
Appendix 1:	An example of backpropagation calculation	108

Index of figures

Figure 1.1: Influence factors in Selective Laser Melting process.....	18
Figure 1.2: Stages of the Additive Manufacturing process [4].....	20
Figure 1.3: Schematic of fused deposition modelling (a) and a real machine (b). (<i>Image source: All3dp</i>)	20
Figure 1.4: Vat polymerization process with a submerged platform (left) and inversed (right) (<i>Image source: carbon3d.com</i>).....	21
Figure 1.5: Material jetting (left) and sheet laminated object manufacturing (<i>Image source: 3dhubs</i>).....	22
Figure 1.6: (a) A prosthetic arm fabricated by the FDM method, and (b) a prosthetic ear made by extrusion-based bioprinting (<i>Image source: CollPlant</i>)	22
Figure 1.7: Printing electronic circuit by the material jetting method: (a) printing on an existing platform of a device, and (b) printing on human skin (<i>Image source: Optomec, University of Minnesota</i>).....	23
Figure 1.8: (a) Blackbelt 3D printer concept, and (b) an oversize part printing (<i>Image source: Blackbelt-3d.com</i>).....	23
Figure 1.9: Working principle of powder bed fusion.....	24
Figure 1.10: The configuration of directed energy deposition of powder nozzle to feed the substrate: (a) coaxial feeding, (b) single feeding [4]	26
Figure 1.11: Hybrid Additive Manufacturing combined between DED (left) and then CNC process (right) (<i>Images source: DMG MORI</i>).....	27
Figure 1.12: Schematic of (a) gas atomized, (b) rotating electrode process, and plasma atomized [25]	28
Figure 1.13: Automotive parts applied AM technology (<i>Image source: metal-am.com</i>) .	29

Figure 1.14: (a) Porsche AlSi10MG turbo intercoolers with a 1.8mm wall thickness and (b) Porsche Inconel 625 turbo exhaust manifold with a 1mm wall (<i>Image source: MIMO Technik</i>).....	29
Figure 1.15: (a) An injection mold manufactured by SLM and its vertical cross-section revealing conformal cooling channel, (b) a sand casting mold produced by SLS technique and a casting product (<i>Image source: MODIA, Voxeljet</i>).....	30
Figure 1.16: The world’s first ship’s propeller manufacturing by metal AM (<i>Image source: RAMLAB</i>).....	30
Figure 1.17: Gas turbine blade with hollow structure (left), the world’s first brake calliper (middle), and fuel nozzles in passenger jet engine (right) printed by metal AM and Ti6Al4V (<i>Image source: Siemens, Bugatti, GE Aviation</i>).....	31
Figure 1.18: (a) Hip joint implant, (b) knee joint implant, and (c) craniomaxillofacial implant fabricated by Selective Laser Melting (<i>Image source: Farinia, Trumpf</i>)	32
Figure 1.19: Application of metal AM part in medical surgery.	32
Figure 1.20: Schematic of a biological neuron (a) and a perceptron (b) [50].....	34
Figure 2.1: (a) Morphology, and (b) Powder size distribution (Ti-6Al-4V ELI).	37
Figure 2.2: The WinforSys 150 SLM Printer	38
Figure 2.3: Schematic of the WinforSys 150 SLM printer.....	39
Figure 2.4: Layout of the experiment inside the processing chamber.....	40
Figure 2.5: Laser scanning strategy in which the hatch angle rotates 67 degrees after every layer	41
Figure 2.6: Surface roughness measurement principle (<i>Image source: Keyence.com</i>)	42
Figure 2.7: (a) The laser confocal microscope VK-200 (left) and its schematic, (b) Contact-type surface roughness instruments [74].....	43

Figure 2.8: (a) A picture of X-ray scanned samples and an image of a cross-section by microscopic analysis [81].....	44
Figure 2.9: Analytical balancer and Ad-1653 kit.....	45
Figure 3.1: Consolidation mechanisms during SLM processing: (a) solid-state sintering, (b) partial melting, and (c) full melting.....	47
Figure 3.2: (a) Schematic of a melt pool geometry and (b) geometry of a melt pool in practice	48
Figure 3.3: (a) Fluid dynamic of melting pool formed the keyhole, (b) bridge formation creating a bubble, (c) bubble flow [83], porosity resulted in keyhole effect in practice [84] .	49
Figure 3.4: Schematic of spattering and powder ejected during the SLM process [91] ...	50
Figure 3.5: Ejected particles upon (a) X-direction laser movement and (b) Y-direction laser movement.....	51
Figure 3.6: Schematic of the balling behaviour: (a) n th powder layer scrapping, (b) Balling formation after processing, and (c) risky pores in next layer printing.....	52
Figure 3.7: Morphology of single scanned track influenced by laser velocity (a, b) and laser power (c, d): (a) P = 150W; v = 800mm/s, (b) P = 150W; v = 1200mm/s, (c) P = 140W; v = 1000mm/s, and (d) P = 170W; v = 1000mm/s	53
Figure 3.8: Single scanned line morphology due to various laser power (P) and laser scanning speed (v).....	54
Figure 3.9: Surface roughness of a printed layer at a different layer thickness (a) t = 20μm, (b) t = 60μm ; and different hatch distance (b) h = 50μm, (c) h = 30μm. ...	54
Figure 3.10: Porosity of a printed part due to balling and spattering behavior.	55
Figure 3.11: (a) Aggravated balling (dash-black circles) and scratched lines on the powder bed surface, (b) a printed part affected by balling.....	56
Figure 4.1: A working of a perceptron.....	57

Figure 4.2: Working of a single layer.....	58
Figure 4.3: A example of deep network.....	59
Figure 4.4: Working of a deep neural network.....	59
Figure 4.5: ReLu (a) and sigmoid (b) activation function.....	63
Figure 4.6: An indication of how the Gradient Descent algorithms work by using derivative.	65
Figure 4.7: Schematic of backpropagation working. The $(l - 1)th$ layer and $(l + 1)th$ possibly are the input layer and the output layer, respectively.....	67
Figure 4.8: The convergence of a network model at learning rate $\eta = 0.01$	71
Figure 4.9: The convergence of a network model at learning rate $\eta = 0.5$	71
Figure 4.10: Gradient descent without the exponential average factor.....	73
Figure 4.11: Gradient descent with the exponential average factor.....	73
Figure 4.12: Correction direction actual step due to exponential weighed average factor	74
Figure 4.13: Gradient step without the exponentially weighted average (up) and with the exponentially weighted average (down) (<i>Image source: DataCamp</i>).....	74
Figure 4.14: (a) Underfitting, (b) good fit, and (c) overfitting problematic (<i>Image source: fundaml.com</i>).....	76
Figure 4.15: Neural network (a) before and (b) after applying dropout.....	78
Figure 4.16: Process of k-fold cross-validation	79
Figure 4.17: The flowchart of the training process of the deep neural network.....	80
Figure 4.18: Virtual mean absolute error of 10-fold cross-validation.....	81
Figure 5.1: Algorithm of the Optimization System	84

Figure 5.2: Schematic presenting the overlap zone during the SLM process.....	85
Figure 5.3: (a) Cracked and (b) broke samples during SLM processing	86
Figure 5.4: Bone structure	87
Figure 5.5: Graphical User Interface (GUI) of the developed optimization system.....	90

Index of tables

Table 2.1: Chemical composition of Ti-6Al-4V ELI alloy (wt.).....	37
Table 2.2: Specification data of the WinforSys 150 SLM printer	40
Table 2.3: SLM process parameters used for experiment dataset.....	41
Table 4.1: Mean absolute error during training and testing of the neural network.....	82
Table 5.1: Generated data for predicting	83
Table 5.2: Optimal process parameters corresponding the user requests.....	88
Table 5.3: Comparison of a user request and experiment result of which process parameters are generated from the optimization system.	88

Chapter 1. GENERAL INTRODUCTION

1.1 Motivation

The demand for implants has significantly increased in the last decades. However, the human parts have individual complicated shapes and structures that limits to fabricate by conventional manufacturing technologies. Additive Manufacturing is defined as the process of joining materials to make products from three-dimensional computer-aided design (CAD) model data, usually layer upon layer [1]. This technology has shown great ability in manufacturing pieces from powders for diverse applications, thus eliminating multiple manufacturing constraints and producing architecture with more complex geometry in contrast to traditional processes [2]. Due to its benefits, the metal AM has widely been used in industrial areas. In medical applications, the Selective Laser Melting (SLM) is the most attractive technique. Printed part by SLM can be achieved nearly at a full density, any geometric shape, and mechanical properties are similar to traditional methods. In addition, titanium and its alloys (preferred material in the medical industry) are ideal target material for SLM because it is an expensive material and problematic to process using traditional fabricating technologies. Titanium parts are challenging to produce by conventional methods due to the high cost of the removed material that adds to production cost. Therefore, applying SLM to fabricate a Titanium part is a promising approach, of which the advantages are the ability to print a net shape part, high material utilization, and minimal machining.

However, printing Titanium powder by SLM is not an easy task. SLM processing is affected by many factors. Figure 1.1 indicates some influence elements during SLM processing. Firstly, the crucial process parameters are layer power, laser scanning speed, the distance of two adjunct scanning line, the thickness of a powder layer, which all significantly affects the SLMed product. Additionally, the range of controlling value of these parameters are wide. While the laser power can reach hundreds of watts, the laser velocity can be modified to thousands of millimetre per second. Besides that, layer thickness and hatch distance can be changed without limitation. Moreover, while material powder characteristics affect the SLMed product through powder particle size, morphology, flowability, and recycled powder, the printer machine contributes by accuracy and durable processing of hardware. Manufacturing conditions such as environment temperature, humidity or shielding

gas changes the properties of the final part also. Thus, using SLM to print a desirable Titanium part is a nearly impossible mission.

Hence, developing an optimization system for selecting SLM parameters is a necessary demand. There are many optimization SLM process parameters approaches, such as the design of the experiment, Taguchi, response surface method, and genetic algorithms. However, they investigated parameters individually that could not figure out a generalization of SLM characteristic. Deep learning is currently widely considered as an optimization method. Deep networks, in general, are able to learn and generalize situations to produce meaningful solutions. It handles all kinds of data, be it experimental, empirical or theoretical. They can cope even in situations where the data is barely understood by humans. Moreover, it can learn and map the inputs to the output without making any assumption during model formulation. Therefore, investigating four crucial factors by using deep learning neural network is considered and applied in this thesis.

At the first section, this thesis gives state of the art approach relating to 3D printing, metal AM, Selective Laser Melting, and Titanium alloy material and deep neural network. The second section indicates the process of collecting data that is needed to train a neural network model. The relationship between process parameters and printed product quality will be determined via sensitivity analysis. Development of a deep neural network will be presented in the following section. The final section will show the progress of the optimization system based on neural network.

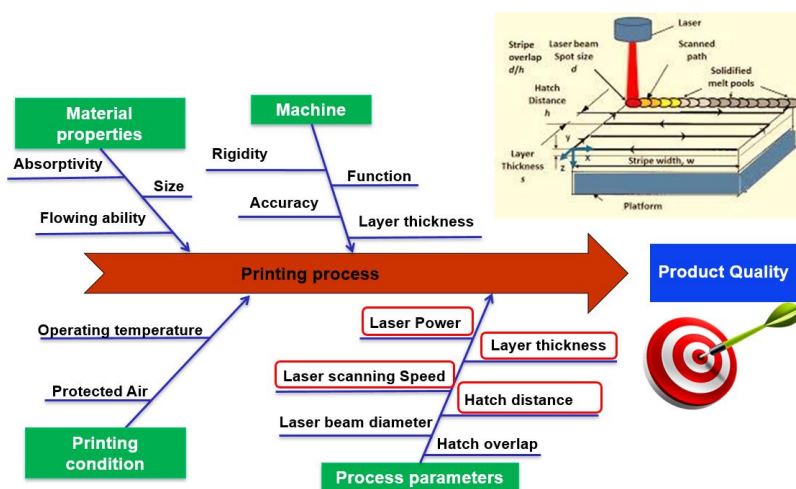


Figure 1.1: Influence factors in Selective Laser Melting process

1.2 State of the art

1.2.1 Additive Manufacturing

The Additive Manufacturing (AM) processes are a significantly simple process, which is by producing a complex 3D object directly from computer-aided design data. It works by adding material in layers [3]. Each layer is a thin cross-section of the part derived from the original CAD data. Traditional manufacturing methods require a careful and detailed plan of part geometry, tooling process, jig and fixtures. In contrast, AM seeks only for some dimensional features and a little background in AM machine working principle and material field. General steps of the AM process showed in Figure 1.2 are as follows:

- The AM process must start with a 3D CAD model that is created by any CAD software like Solidworks, Inventor, Catia, or by the scanning process.
- The CAD model is transferred to stereolithography (*.STL) format, which nearly every AM technologies use. The STL standard is a way of describing a CAD model in terms of its geometry alone. It works by removing any construction data and approximating the surfaces of the model with a series of triangular facets [4].
- The STL format file is sent to the AM machine where it can orient model, arrange position to allow to build on the platform, set up the support section, set up process parameters, allow printing of multi-part at a time.
- After setting up the AM machine, the model is built layer by layer. The printing process will repeat until the model completes.
- The support section supports the overhang areas and anchors the printed part to the platform. Therefore, they must be removed after the printing process.
- Depending on application purpose, the printed part could be post-processed in terms of abrasive finishing, polishing, heat treatment, etc.
- Finally, the printed part can be used after post-processing

There are several principles of AM processes such as material extrusion, vat polymerization, material jetting sheet lamination, powder bed fusion and directed energy deposition. The material extrusion working uses an effective 1D strip of material to fill in a 2D space to form a layer, which is printed on top of each other to create a fully 3D object. Schematic of fused deposition modelling (EDM) and a real FDM printer shown in Figure 1.3a), b) is a common

example of this technique. Material is drawn through a nozzle, where it is heated and the deposited layer by layer. Typically, the nozzle can move horizontally, and the platform moves vertically.

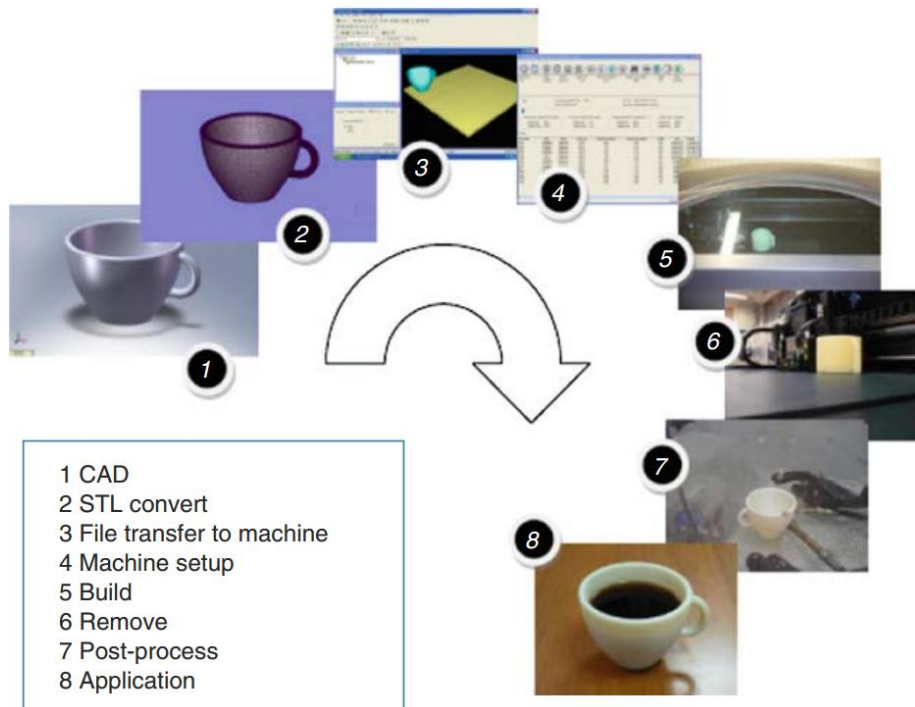


Figure 1.2: Stages of the Additive Manufacturing process [4]

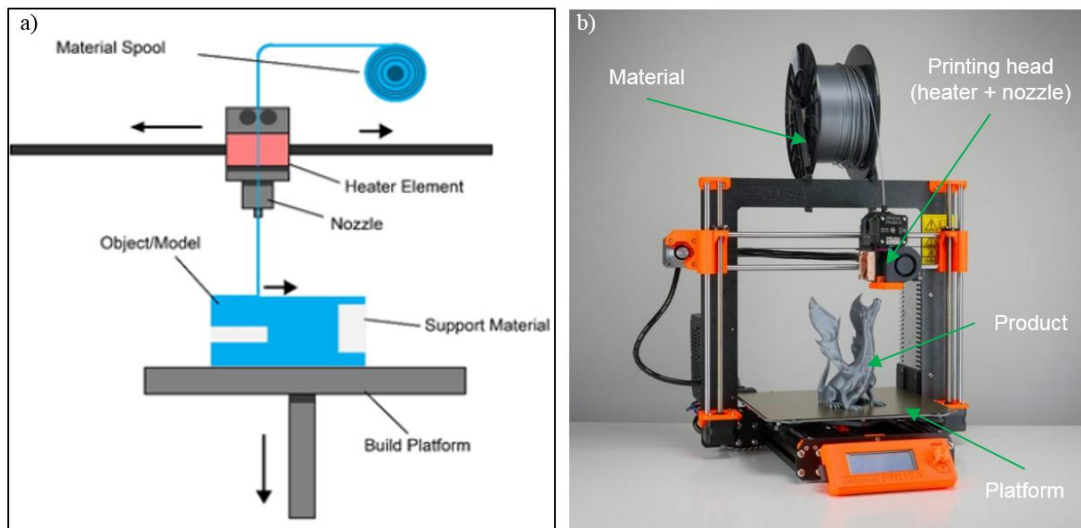


Figure 1.3: Schematic of fused deposition modelling (a) and a real machine (b). (*Image source: All3dp*)

The critical element of Vat polymerization process bases on photo-polymerization resin and resin exposure system [5]. The process could be configured by a submerged platform type or the opposite way, as shown in Figure 1.4, left and right, respectively. The exposure system polymerizes the layer of resin between the resin surface and the build plate until the entire 3D object is finished.

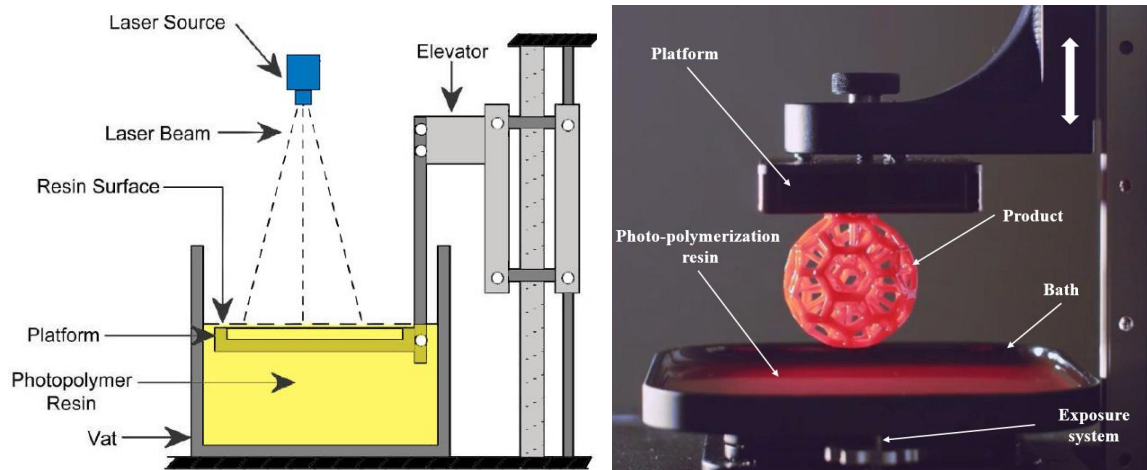


Figure 1.4: Vat polymerization process with a submerged platform (left) and inversed (right)
(Image source: carbon3d.com)

Material jetting works the same as a standard paper inkjet printer. Schematic of material jetting working principle is showed in Figure 1.5 (left), by using a single nozzle or arrays of nozzles to deposit droplets of material layer by layer on top of each other to form a solid part instead of on single layer of ink [6]. The material jetting offers higher scalability in productivity, and part dimension compared with the FDM. Another method is sheet lamination, referred to as Laminated Object Manufacturing (LOM). It operates based on chemically or mechanically bonding sheets of material (metals, paper, polymer) in a stack usually through heating or pressure, and cutting of the materials stack to the desired contour of a given layer [7]. The concept of the LOM process is indicated in Figure 1.5 (right). On one side of the bed is a roll of sheet-type material, with adhesive on the other side. The material is rolled over the whole bed and heated. A laser source traces over the cross-section of the model and then cross-hatches over all of the extraneous material so that it can be removed later.

AM processes are different from conventional manufacturing processes (forming, casting, and machining process). The core difference is, while traditional process shaping of materials

takes place across the entire physical domain of the desired part, the AM processes take place in the formation of the elements such as filaments or layers [8].

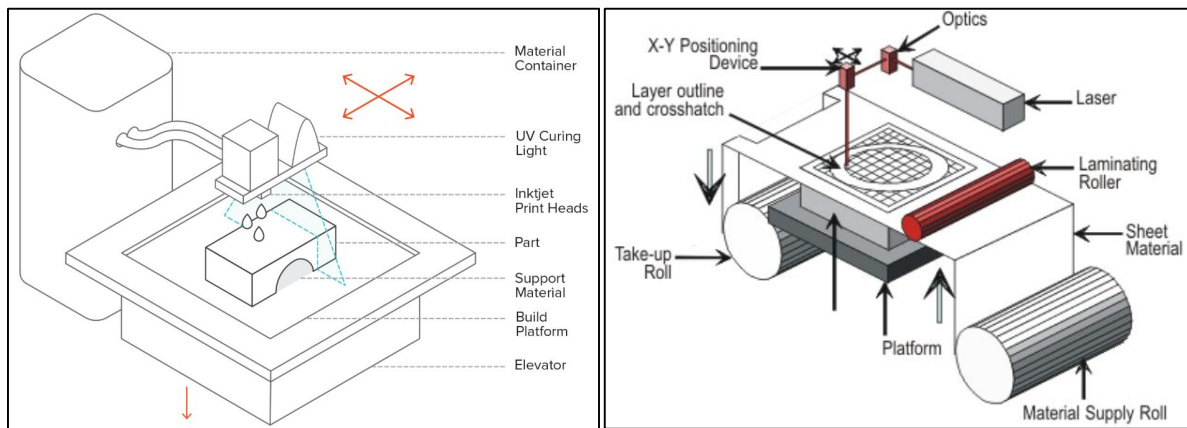


Figure 1.5: Material jetting (left) and sheet laminated object manufacturing (*Image source: 3dhubs*)

From application viewing, AM methods present a high degree of customization and personalization with a small impact on manufacturing complexity and cost [8]. An example is using the EDM method to print a prosthetic arm, as shown in Figure 1.6(a). Due to scanning and EDM, it can custom build each arm faster and cheaper than conventional fabrication methods. The additive manufacturing methods also are borrowed to print human body part [9]. Figure 1.6(b) shows an extrusion-based bioprinting technique. It is mostly used for fabricating tissue engineering. Bio-inks are the biomaterials laden with cells and other biological materials [10]. Due to the possible printing on a 3D surface, material jetting is used to print electronic circuit in smart devices [11] as shown in Figure 1.7(a) or even on human skin Figure 1.7(b).

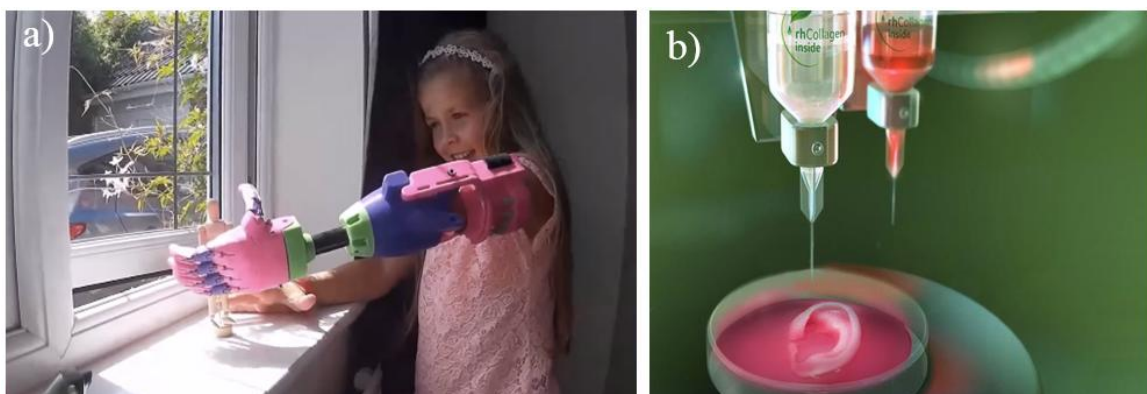


Figure 1.6: (a) A prosthetic arm fabricated by the FDM method, and (b) a prosthetic ear made by extrusion-based bioprinting (*Image source: CollPlant*)

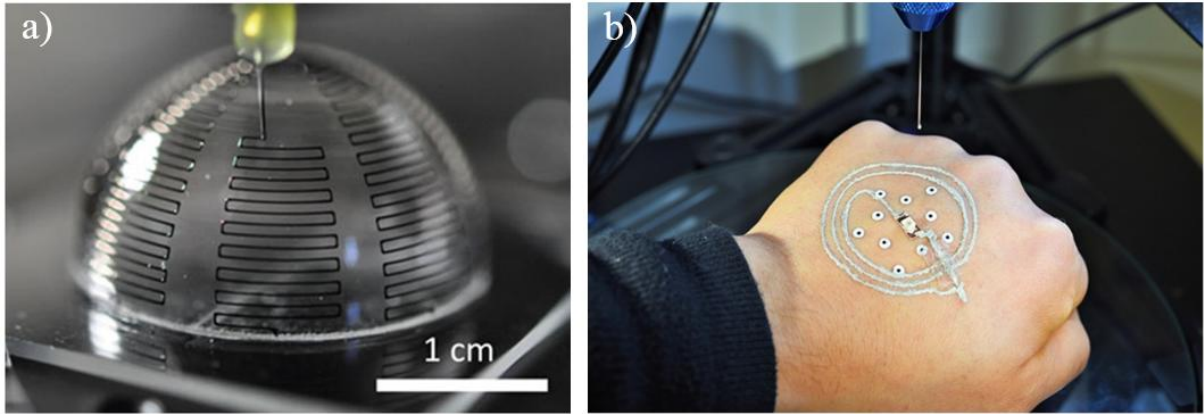


Figure 1.7: Printing electronic circuit by the material jetting method: (a) printing on an existing platform of a device, and (b) printing on human skin (*Image source: Optomec, University of Minnesota*)

Owing to many advantages, however, AM is not a method without drawbacks. The significant problem is the limitation of printed part size and printing productivity compared with traditional manufacturing methods. However, the disadvantage is fulfilling. Figure 1.8(a) indicates an endless printing process, called Blackbelt 3D printer. In a traditional EDM printer, the printing head moves in a frame that is parallel to the platform surface, as shown in Figure 1.3. The printing process must stop when it exceeds the distance. However, in the Blackbelt printer design, the frame where the head printing movement has inclined an angle with the platform that is a belt. Therefore, the printer can operate endlessly. It can print an oversize-part as long as required as shown in Figure 1.8(b)

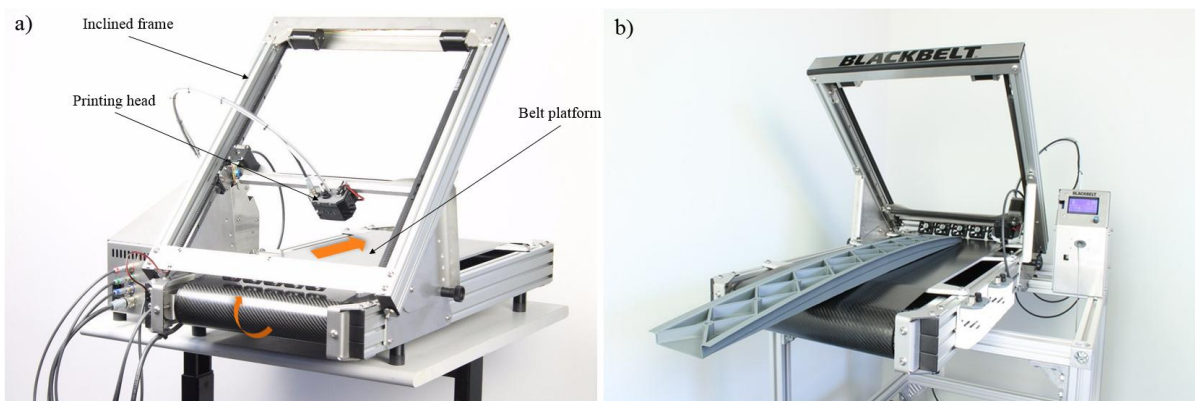


Figure 1.8: (a) Blackbelt 3D printer concept, and (b) an oversize part printing (*Image source: Blackbelt-3d.com*)

1.2.2 Metal Additive Manufacturing

Currently, metal Additive Manufacturing (AM) has been used as a promising technique to manufacture metallic parts because they can prevent the common problematics of traditional metal manufacturing methods. The metal AM can produce complicated products without limitation of designing. Similar to 3D printing generally, the metal AM techniques creates a three-dimensional object from a computer-aided design (CAD) model by building it up from a thin layer of material layer-by-layer [12][1]. Instead of removing material from a large stock, such as machining, the metal AM process fabricates a final part based on adding material principle. They make highly efficient use of the material and minimize waste material. While traditional manufacturing processes tend to impose various constraints on the design of a product, the ability to print a high-complex shape of the metal AM enables designers in building a 3D CAD model. The metal AM includes different types, namely powder bed fusion (PBF), directed energy deposition (DED), and hybrid additive manufacturing. They consist of three essential step: building a standard AM file format based on 3D CAD model; the part is processed by a software that can orient the part, set up support section, and set up process parameters such as power and velocity of heat source, layer thickness; the processed part is built layer by layer on an AM machine.

a) Powder Bed Fusion

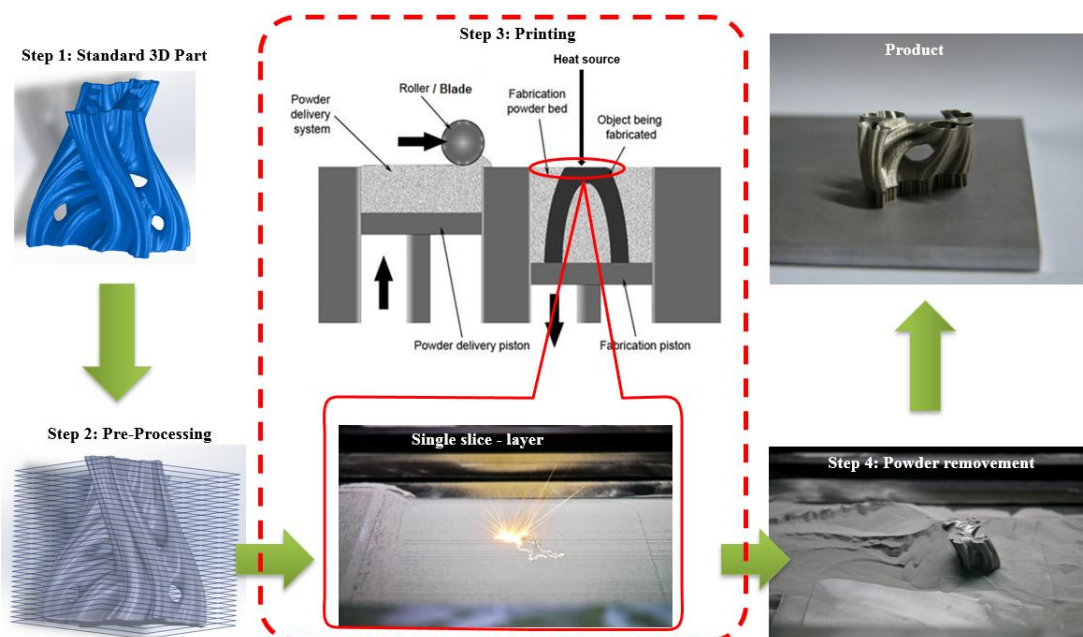


Figure 1.9: Working principle of powder bed fusion

This process uses as material a metallic powder deposited on a platform moving vertically and a laser beam or electron beam as the heat source. PBF is the most prevalent metal additive manufacturing method used to fabricate engineered products. The operating principle of the PBF is shown in Figure 1.9. Step 1 and step 2 is performed as same as a general AM technique. During processing, a layer of powder is spread on top of the fabrication piston area. Then, a laser source scans the powder layer based on a slice-form of the CAD model. After scanning, the fabrication platform lowers off the specified layer's depth, and a new quantity of powder, coming from the powder delivery piston is spread. The process is repeated until the product has been completed. After finishing, the excess powder is removed and recycled at step 4 [13] to get the final product.

Depending on the applied heat sources, the PBF includes electron-based PBF as Electron Beam Melting (EBM) or laser-based, namely Selective Laser Sintering (SLS) and Selective Laser Melting (SLM), the target technique in this thesis. While the EBM, as its name implies, uses electron source to manufacture parts with high temperature and vacuum environment, the SLM and SLS use a laser beam to melt the material powder with a shielding gas such as nitrogen, hydrogen or argon.

The SLS utilizes two-components metal powder combined as a sacrificial binder and the powder particles or single powder with two different grain sizes. During the SLS process, only binder material or smaller particle is melted, which binds the surrounding powder after cooling down [14]. The printed part will be applied to a heat treatment method to get the final product[15]. However, the SLM enables fabrication of a component with full melting material. The SLM does not need adding a lower-melting point material as the SLS technique. The SLM process requires higher laser power to melt powder completely. The most consideration during the SLM process is the residual stress resulting from the steep thermal gradient. However, it could be solved by applying pre-heating during printing [16]. The EBM, less widespread than the laser's one, uses an electron beam of 60kV and 3kW and needs a conductive material. This process can be more productive than the one using a laser as long as the electrons beam brings more energy to the powder, moves faster, and allow to work with thicker layer and larger powder particle size. However, the resulting surface states with this process are not as good as the ones with the laser's operation because the beam is more extensive and thus less precise.

b) Directed Energy Deposition, DED

This process delivers the material as a powder jet or a wire in the melting zone close to the surface instead of layering on the substrate [17][18]. The DED uses three types of material deposition to the substrate such as powder injection, pre-placed powder and wire extrusion. In the case of powder type, the DED feeds material powder by combining with a gas system via a nozzle, as shown in Figure 1.10. The high energy heat source is applied in the centre of the nozzle to melt material. Advantage of the DED method is high productivity. The flow rate of the powder is around $300\text{cm}^3/\text{h}$ with a powder size from 45 to 150 micrometres: thus, the manufacturing speed is more interesting than the PBF technique. Additionally, the DED could manufacture on the 3D surface instead of a 2D surface like the PBF. Typically, the supply of material is done from a nozzle moving on three axes on a platform of 2 axes. However, the axis of material supplier unit and substrate can be extended or combined with a robot arm. This mobility can simplify the manufacturing with the reduction of supports use. However, the DED parts have less accuracy because of stair effect. Therefore, the products fabricated by this method may additionally ask machining process to get a better final part, as indicated in Figure 1.11 [19][20]. The combination is called hybrid metal AM.

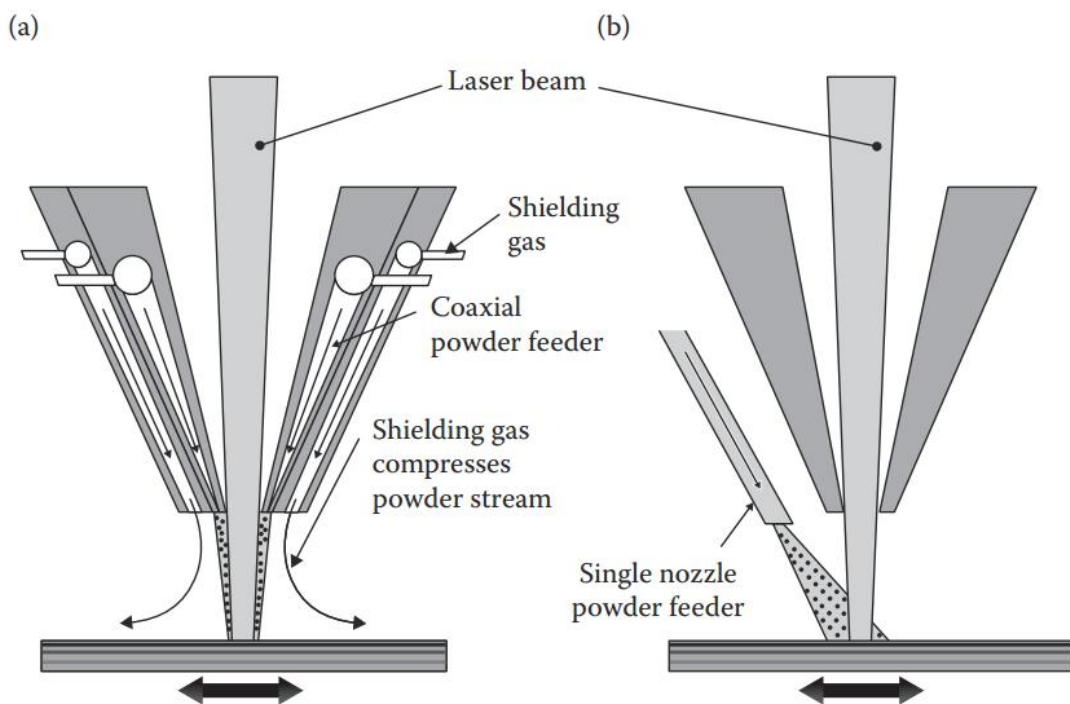


Figure 1.10: The configuration of directed energy deposition of powder nozzle to feed the substrate:

(a) coaxial feeding, (b) single feeding [4]



Figure 1.11: Hybrid Additive Manufacturing combined between DED (left) and then CNC process (right) (*Images source: DMG MORI*)

c) Powder metallurgy

Metal powders used for the SLM process require strictly the size, shape, and chemistry that are critical for a successful and repeatable process. Powders currently used for AM most commonly range from 10 to 105 microns in size and are generally spherical in shape. It allows to create delicate layers of powders in powder bed fusion process and to deliver an inert gas fed stream of powder smoothly without nozzle clogging, in case of powder directed energy deposition process.

Titanium alloy Ti-6-4 powder is produced by gas atomization, plasma atomization, and the plasma rotating electrode processes [21]. The size, shape, and chemical purity is different for each procedure and affects laser and electron beam processes, and powder bed or gas stream delivered [22][23][24]. Figure 1.12(a) shows a typical gas atomization process where the metal alloy is induction or furnace melted and atomized with a stream of gas, typically argon, collected and sieved into the desired range of sizes. The gas atomization process can produce high purity, spherical powders [25]. Figure 1.12(c) shows the plasma atomization process. Wire-type material is fed into the atomization chamber and melted by plasma heat sources to produce a highly spherical powder.

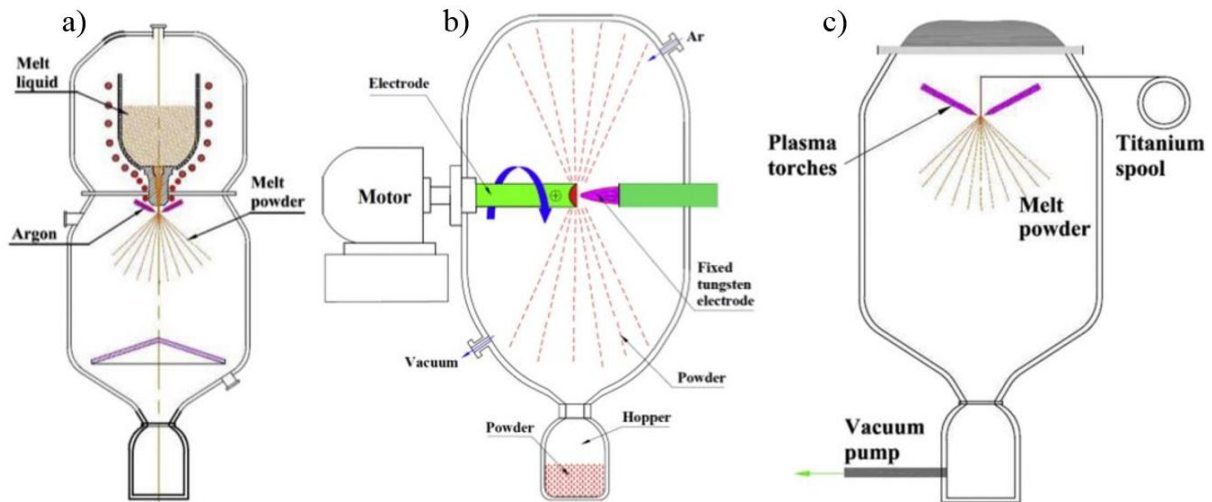


Figure 1.12: Schematic of (a) gas atomized, (b) rotating electrode process, and plasma atomized [25]

Another method is the rotating electrode process (REP) of which schematic is shown in Figure 1.12(b). In this process, a titanium bar is spun at about 18000 rpm in an inert gas-filled chamber. A heat source (either an electric arc or a plasma torch) melts the surface of the end of the spinning bar [26]. When the heat source is an electric arc, the powder is known as REP powder, and when it is a plasma torch, it is known as PREP powder. The molten titanium alloy at the surface is separated from the rotating electrode and impelled outward by the centrifugal force. Under the influence of surface tension, spherical droplets are formed, and these solidify in flight. The solid spherical pre-alloyed powder particles formed this way have a size that depends on the rotational velocity of the electrode [27]. The REP product has a larger average size than the gas atomized process.

d) Metal AM application

Comparing with the above mentioned AM method, the metal AM is widely applied in industrial fields such as aerospace, automotive, etc. Figure 1.13 shows how AMed parts are applied in a car. The ability to lightweight component benefits fuel economy, and lower inventory and handling costs in the supply chain. Therefore, Porsche used SLM technique to manufacture the turbo intercooler and turbo exhaust manifold as illustrated in Figure 1.14 (a) and (b), respectively. Because AM techniques can print any complex shapes, they are used to print any structures and shapes. Figure 1.15 demonstrates the application of metal AM in the molding industry. Figure 1.15(a) shows a mold (left) manufactured by SLM and the cooling channel inside (right). Instead of a straight channel fabricated by a conventional method, the

conformal channel can be designed in any shape. Therefore, the cooling step is faster and homogeneous in total injected part that improves product quality [28] [29][30]. The Selective Laser Sintering (SLS) is used to prepare a sand mold for the casting process, as shown in Figure 1.15(b). More attractively, the hybrid additive manufacturing combined between directed energy deposition and milling process is widely using for manufacturing large components. Figure 1.16 indicates the world's first propeller manufacturing by the hybrid AM. After finishing the DED process, the powder supplied nozzle is replace and milling tool is installed for the finishing part.

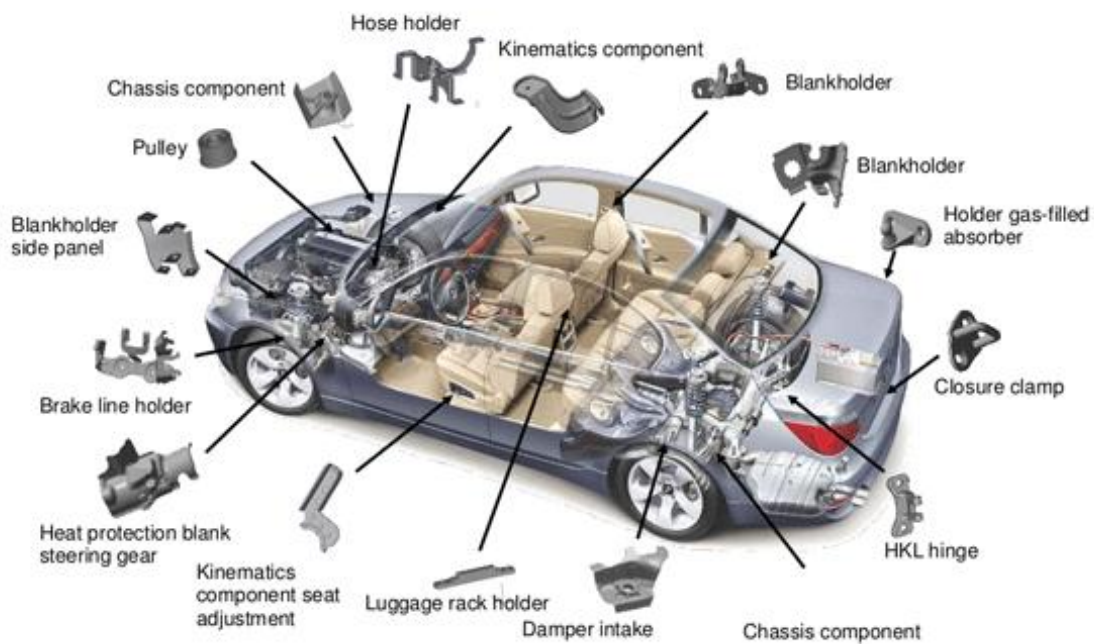


Figure 1.13: Automotive parts applied AM technology (*Image source: metal-am.com*)

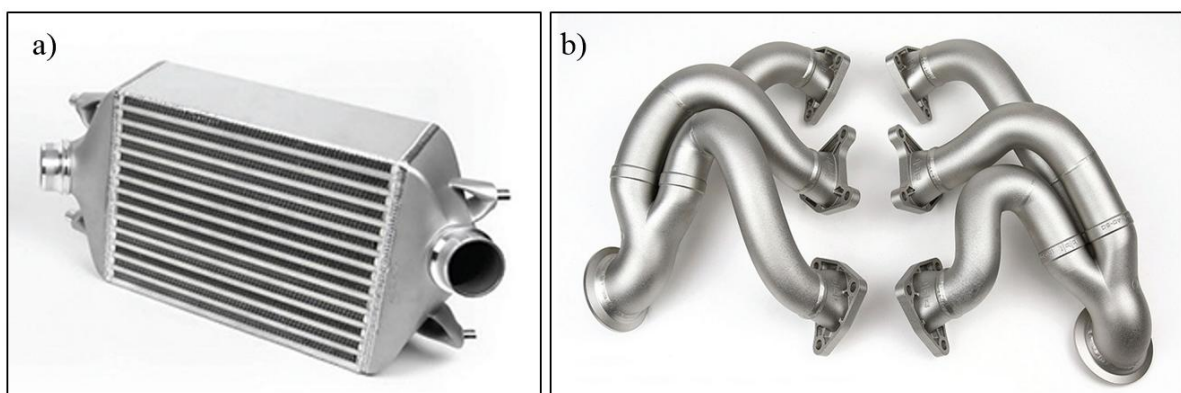


Figure 1.14: (a) Porsche AlSi10MG turbo intercoolers with a 1.8mm wall thickness and (b) Porsche Inconel 625 turbo exhaust manifold with a 1mm wall (*Image source: MIMO Technik*)



Figure 1.15: (a) An injection mold manufactured by SLM and its vertical cross-section revealing conformal cooling channel, (b) a sand casting mold produced by SLS technique and a casting product
(Image source: MODIA, Voxeljet)

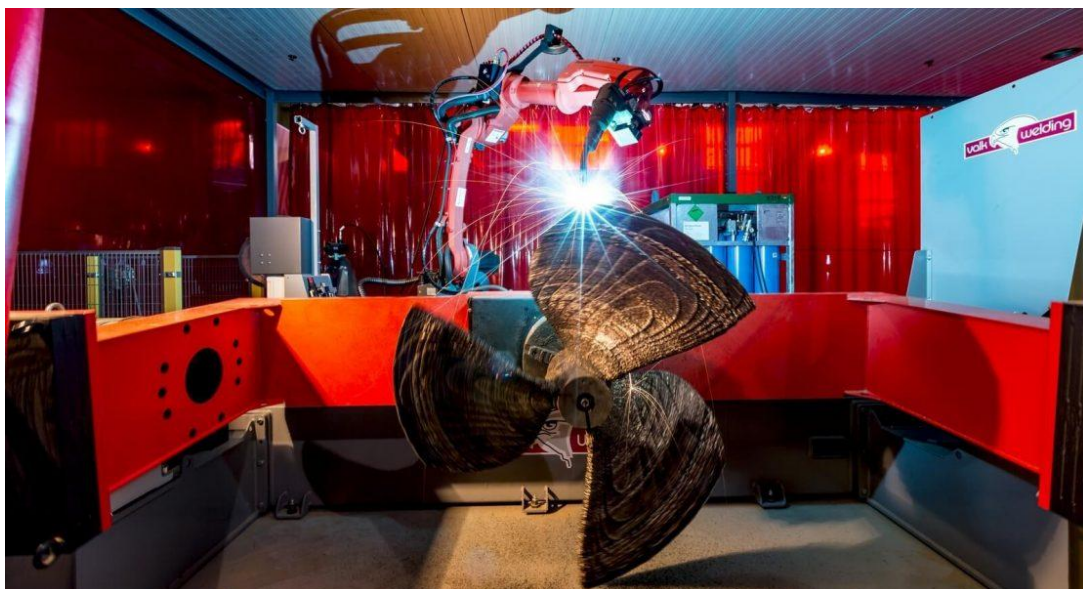


Figure 1.16: The world's first ship's propeller manufacturing by metal AM *(Image source: RAMLAB)*

1.2.3 Titanium Ti-6Al-4V alloy

Titanium alloys offer excellent strength, corrosion resistance, biocompatibility, low thermal expansion and lightweight. The most common alloy Ti-6Al-4V combines pure titanium with 6% aluminium and 4% vanadium. The Ti-6Al-4V extra low interstitial (ELI), the target material in this thesis, specifies a higher degree of purity for critical performance applications. It is often used in aerospace and medical implant applications where cost is secondary to in-service performance or biocompatibility [31]. The high price of these speciality powders limits their use in consumer products, but with the high utilization and reduced scrap rate while using metal AMs.

Titanium alloys are used to manufacture a blade of the gas turbine as shown in the left of Figure 1.17, or a fuel nozzle in a passenger jet engine in Figure 1.17 (right). Moreover, this advantage also applied to the manufacture component of a race car in the automotive industry. Middle of Figure 1.17 reveals a titanium brake calliper of Bugatti company. It is the world's first brake calliper made by metal SLM technology. It is stronger, lighter and better cooling compared to the conventional fabrication methods.



Figure 1.17: Gas turbine blade with hollow structure (left), the world's first brake calliper (middle), and fuel nozzles in passenger jet engine (right) printed by metal AM and Ti6Al4V (*Image source: Siemens, Bugatti, GE Aviation*)

Extra-low interstitial Titanium Ti6Al4V ELI has also been applied in biomedical applications because of its excellent biocompatibility [31]. It is widely used as structural and functional biomaterials for replacement of hard tissues in devices such as artificial total hip or knee replacement and dental implants [31]. The use of titanium in the biomedical field has become a well-established area because titanium fulfils the property requirements better than any competing material. The properties which are of interest for biomedical applications are

corrosion resistance, biocompatibility, bio-adhesion (bone ingrowth), modulus of elasticity, fatigue strength, and good processability (including joining and casting). There are numerous different medical devices using titanium materials, e.g. bone plates[32], screws, hip joint implants [33], stents [34] and various other kinds of fixtures also used in the dental area, as indicated in Figure 1.18 a, b, c. Figure 1.19 illustrates an example of skull parts printed by SLM.



Figure 1.18: (a) Hip joint implant, (b) knee joint implant, and (c) craniomaxillofacial implant fabricated by Selective Laser Melting (*Image source: Farinia, Trumpf*)

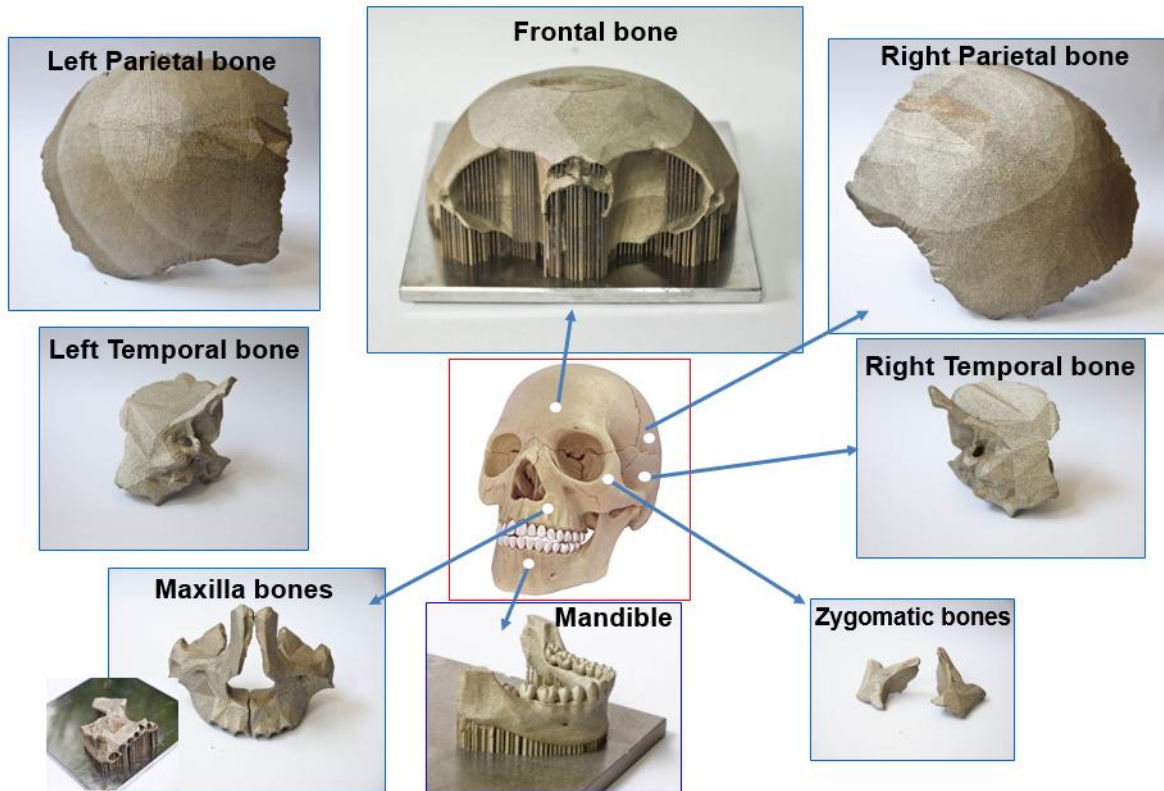


Figure 1.19: Application of metal AM part in medical surgery.

1.2.4 Deep learning network

Metal AM can fabricate parts with a density of nearly 100 percent and eliminates the need for post-processing. Although there are overwhelming advantages, it is not a method without limitation. They are complicated processes, and AMed parts are affected by many factors such as printing conditions, material powder characteristics, and laser properties. Therefore, the operator needs a good background of its technology and material science. Selecting wrong process parameters leads to unwanted behaviour namely high deformation, balling [35], cracking [36] [37], and high porosity [38] that result in a bad product. There are many optimization techniques applied in engineering problems, namely Taguchi, design of experiment, response surface method, fuzzy, [39] genetic algorithms, among others. They were used to optimize process parameters of engineering process such as milling [40][41], drilling [42], and turning processing [43] [44]. They also were applied to optimize metal AM manufacturing. Taguchi method was used to maximize density fabricated by AM methods [45] [46], and design of experiment which considered surface roughness as its target [47] [48] [49] were presented. However, those statistical methods are not suitable to use in this thesis because their accuracy is easily affected by non-linear data. Therefore, deep learning is used in this research instead.

a) Evolution of neurocomputing

By definition, deep learning is a structure comprised of densely interconnected adaptive simple processing elements that are capable of performing massively parallel computations for data processing and knowledge representation. The very first artificial networks were inspired by the biological neuron from which its structure and functioning have been mimicked extensively in modern computing. A biological neuron, (Figure 1.20(a)) consists of the cell body part that acts as the central command point, dendrites that act as transmitters and axon that connects the body part to the synapses. The human brain is composed of numerous interconnected nodes of neurons. Each node receives input signals from an external environment or neighbouring neurons and processes locally. If the processed signal is strong enough, it causes 'activation' to produce an output which is passed on to the next 'layer' of nodes or to external outputs (effectors) to trigger the response.

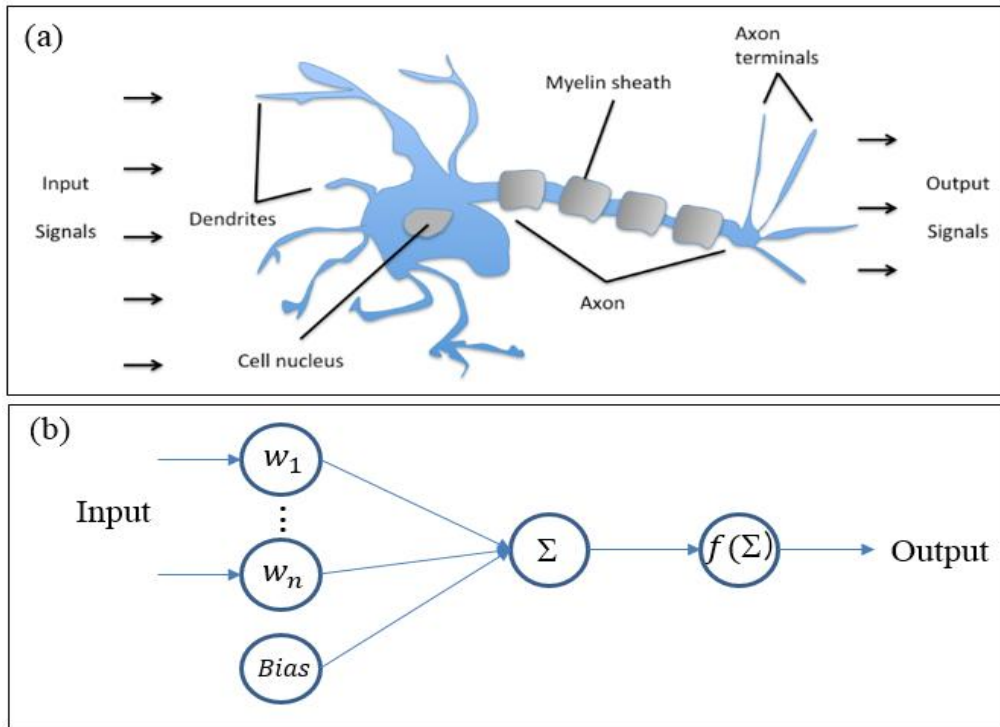


Figure 1.20: Schematic of a biological neuron (a) and a perceptron (b) [50]

An artificial neuron dates back to 1943 when psychiatrist Warren McCulloch and Mathematician Walter Pitts introduced a simple neuron, shown in Figure 1.20(b). Upon further studies, they discovered that biological neurons could be represented as conceptual circuit components that can be used to perform several computational tasks. A neural network is a technique used to map a random input vector into a corresponding random output vector without assuming that there is any persistent relationship between the two sets. A typical deep neural network has three layers, the input layer, hidden layer and the output layer. The arrangement of neurons in a layer and layers in a network is called neural network architecture. The number of neurons corresponds to the size of the input and output layers, while the hidden layer can be manipulated to suit the level of the desired output. The mapping process is achieved by first assigning each input with connection weights, which transmit the information to the next neuron or junction. The weights vector is first assigned randomly and subsequently fixed by ‘training’ the network. The training aims at achieving an optimal set of weights that minimizes the error. The system iteratively takes the dot product of the vector of connection weights and the input, and sum the total at a summing junction before sending it to the activation function of the network. It is also important to note that

each neuron functions independently as the whole network. Several activation functions can be used, depending on the nature of the inputs and the corresponding desired outputs.

In the present-day, neural networks cover a wide area of applications ranging from business, engineering, research and development as well as financial applications [51][52]. In engineering, deep neural networks have been used in the aerospace industry for fault diagnosis, autopilot enhancement [53][54]. In manufacturing, deep networks have been used to solve a wide range of problems including manufacturing process control, quality control and measurements, and many dynamic modelling of otherwise virtually understood problems. Deep learning was used to optimize the electrical discharge machine (EDM) process parameters [55][56], molding processing and molded product quality control [57][58], monitoring welding quality [59], waterjet machining process [60], predicting cutting force in turning process [61], etc. In metal Additive manufacturing, deep neural network was used for optimizing process parameters [62][63], prediction distortion [64], process monitoring [65], fault detection [66], and the powder bed fusion process.

Supervised learning algorithms are learning methods to associate some input, \mathbf{x} , and, for each one, a corresponding desired output, \mathbf{y} . The process is supervised at each step as to what it is expected to do. The supervised learning algorithm is often used in classification and regression problems. Unsupervised learning is a method where only the input data is presented without corresponding output. It is called so because there is no correct answer. The algorithms are left to their own devices to discover and present the interesting structure on the data. Unsupervised learning is applied in the clustering problem. Another algorithm is reinforcement. In reinforcement learning, it does not need examples of correct input-output pairs but needs a method for the machine to quantify its performance in the form of a reward sign.

b) The reason for selecting deep learning in this research

Deep neural networks, in general, are able to learn and generalize situations to produce meaningful solutions. It handles all kinds of data, experimental, empirical or theoretical. They can cope even with cases in which the data is fuzzy and barely understood by humans and are able to adapt the solutions to even the most dynamic circumstances [67]. Supervised learning is used in this research to conduct experiments involving input parameters and output of measured product properties. The process records by learning each at a time and comparing

the output of the model and the real target. The next layers are fully connected to the other layers, and thus an error obtained in one successful prediction is used as an input with an objective to minimize it in the next level. Feedforward networks (FFNN) have the following specific benefits that are owed to its wide applications: Since FFNN is data-driven; it can learn and map the inputs to the output without making any assumption during model formulation. Secondly, they are universal approximators of most dynamic and nonlinear models, which are useful for situations which are virtually or not understood at all. In metal AM, printed product qualities are affected by process parameters. Analysis influence of parameters independently will not represent the entire processing. However, researching interrelationship of multi-process parameters and fabricated product qualities is so complicated that conventional optimization methods could not generate a trustable result. Therefore, applying a deep network is a promising approach to optimize SLM process parameters in this thesis.

Chapter 2. METHODOLOGY

2.1 Material

In this study, Titanium alloy Ti6Al4V ELI material powder (SLM Solutions Co., Ltd, Germany) was used, of which element compositions, %wt. are shown in Table 2.1. The powder particle size of the Titanium alloy powder was 30-45 μ m. A scanning electron microscopy (SEM) image of the powder made by the gas atomization method is shown in Figure 2.1a. Figure 2.1b indicates the particle size distribution measured by the laser scattering analyzer LA-960.

Table 2.1: Chemical composition of Ti-6Al-4V ELI alloy (wt.%)

Element	Ti	Al	V	Fe	C	N	O	H	Others
Ti-6Al4V ELI	Bal.	5.50-6.50	3.50-4.50	0.25	0.08	0.03	0.13	0.0125	0.50

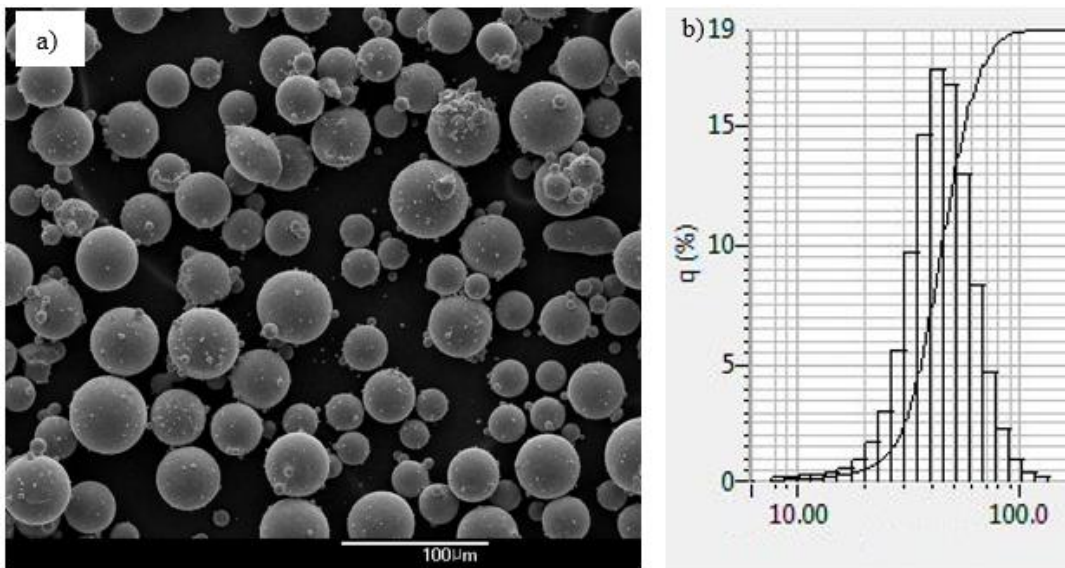


Figure 2.1: (a) Morphology, and (b) Powder size distribution (Ti-6Al-4V ELI).

2.2 Experiment

A selective laser melting printer (MetalSys 150, Winforsys co., Ltd) with a YLR-200-AC-Y11 IPG Ytterbium Fiber Laser, 200W maximum output power was used for the experiment. Figure 2.2 illustrates the printer's components and the technical parameters are shown in

Table 2.2. Figure 2.3 indicates the configuration of the printer. In the beginning, the argon gas was pumped into the chamber to maintain an oxygen level below 0.5% and then stopped. The “Scraper” moves back and forth between the “Feed container” and the substrate to create a layer of powder on the top of the “Removable substrate”. Then, the “Laser” source scans the powder layer based on a slice-form of the CAD model. After scanning, the substrate has lowered by a distance equal to the layer thickness controlled by “Step motor 1”, while the “Piston head” on the feed container has raised. The process is repeated until the product has been completed. After finishing, the excess powder was removed and recycled.

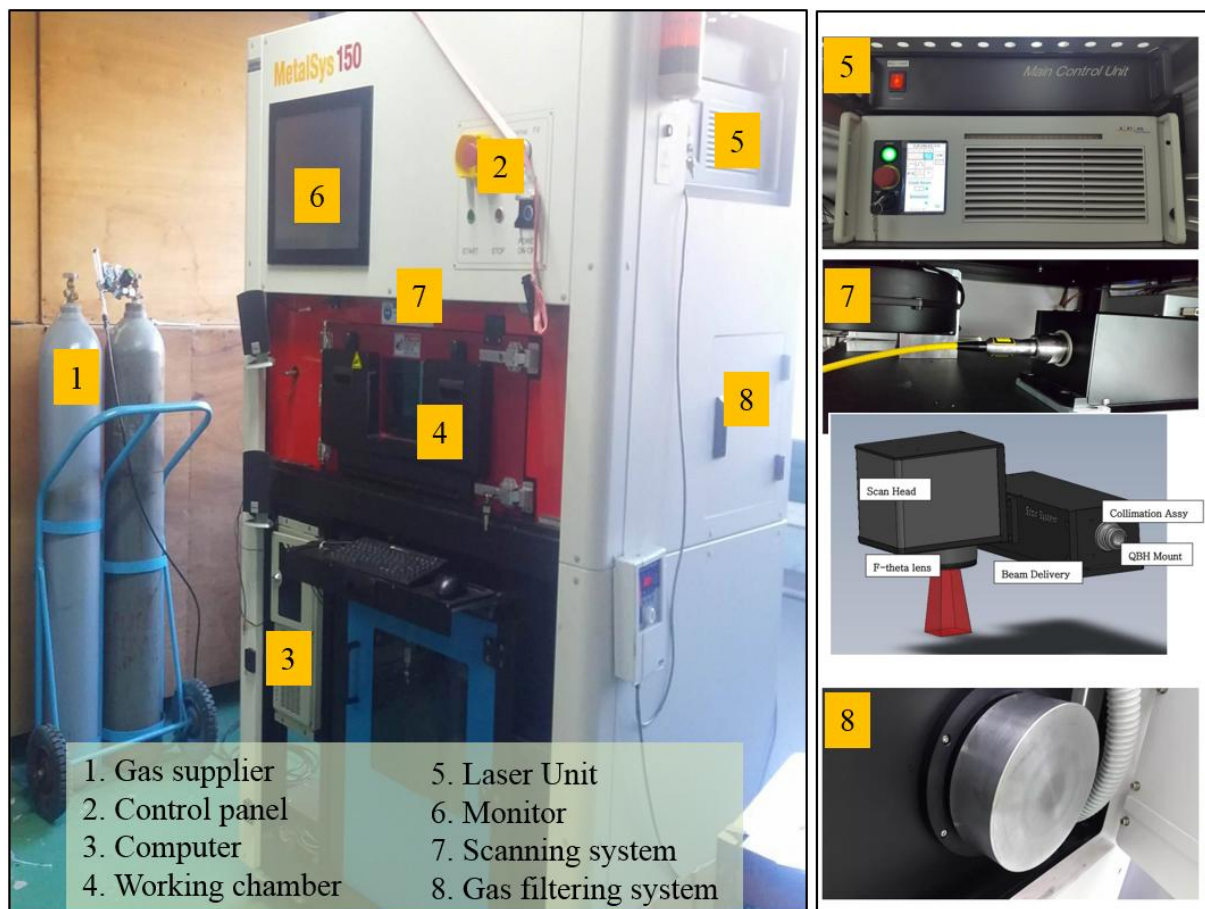


Figure 2.2: The WinforSys 150 SLM Printer

A series samples of $5 \times 5 \times 5 \text{ mm}^3$ boxes were arranged along the diagonal line of the substrate, as indicated in Figure 2.4, to avoid their influence on each other because of the gas flow effect [68][69][70]. The distance between each cube is 2 centimetres. The printing process used the meander laser scanning strategy in which the laser scan line in a layer are parallel and increased 67 degrees for next layer [71][72]. The scanning strategy is shown in Figure 2.5.

An arrangement of the experiment chamber and the gas circulation system, which was controlled by a fan (and shown in Figure 2.4) was running throughout product processing. The fume produced during the SLM process was removed from the chamber through the outlet, filtered, and then returned to the chamber. The fan was running continuously during the process to maintain a clear chamber atmosphere.

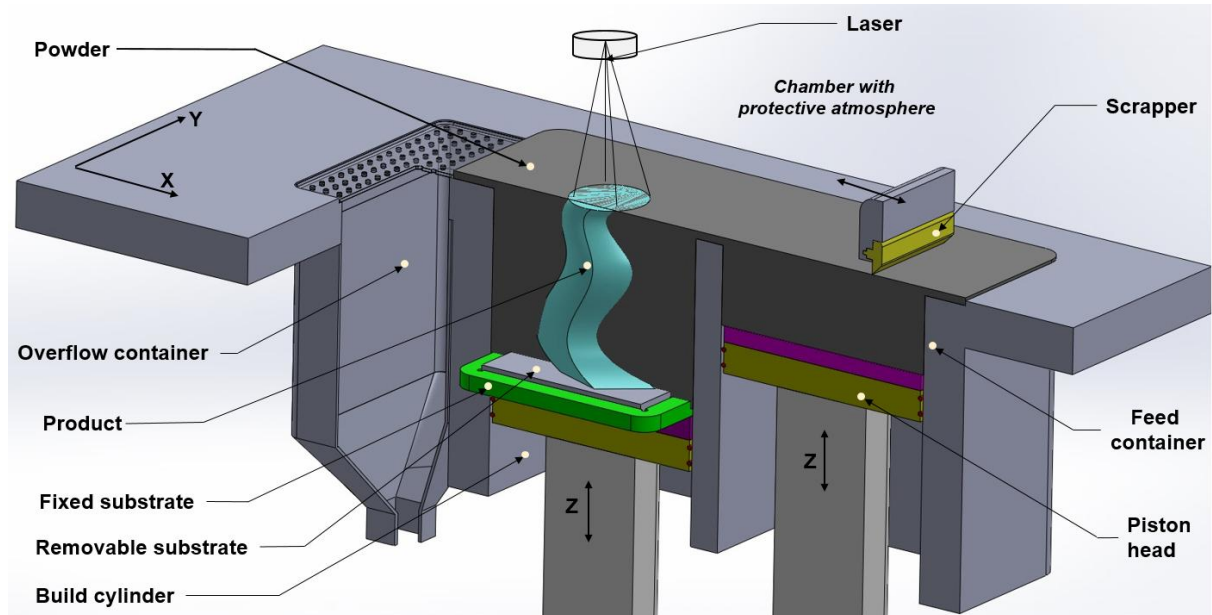


Figure 2.3: Schematic of the WinforSys 150 SLM printer

The selection of the experimental test cases to be measured is of great importance for the deep learning process. The dataset determines the applicability, the range of valid prediction and quality of the prediction. Therefore, the experimental dataset must be broad and representative of the problem to be modelled [73]. The experimental dataset in this thesis is the four investigated process parameters namely, laser power, laser scanning speed, hatch distance, and layer thickness. The laser spot diameter was fixed at $70\mu\text{m}$. The experimental dataset was generated by taking each of parameters and producing different variations of which include eight levels for laser power, laser scanning speed, hatch distance, and four levels for layer thickness. In this method, a total number of 2048 test cases representing the SLM 3D printing technique and the manufacturer cartography were obtained. Process parameters for experiments and their levels are shown in Table 2.3. After printing, SLM processed parts were tested for surface roughness and density.

Table 2.2: Specification data of the WinforSys 150 SLM printer

Item	Value
Laser wavelength	1,075nm
Laser Output power	200W
Pulse repetition	CW
Output power tunability	10 ~100%
Beam quality	M2 <1.1
Scanner positioning speed	7m/s
Building volume	150x150x150mm (X x Y x Z)
Beam spot	70 ~150um
Z-axis speed	100mm/s
Z-axis accuracy	+3um
Build room temperature	40 ~ 80°C
Process Gas	Argon gas
Powder feeder stroke	500mm
Blade	Silicon
System Max size (L x D x H)	850 x 1200 x 2000mm

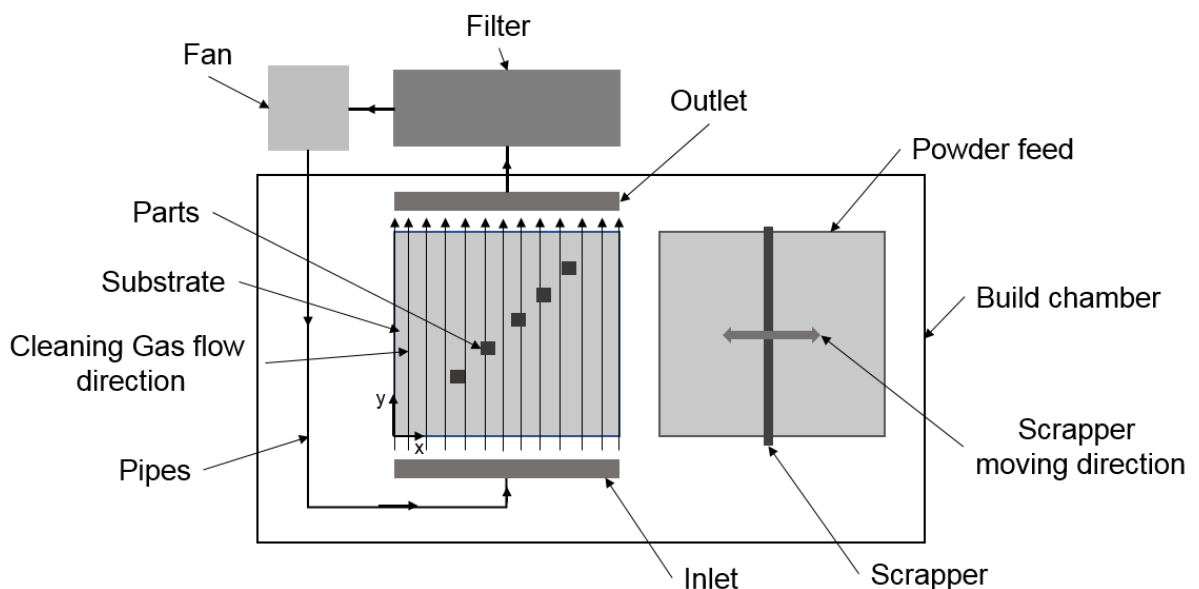


Figure 2.4: Layout of the experiment inside the processing chamber.

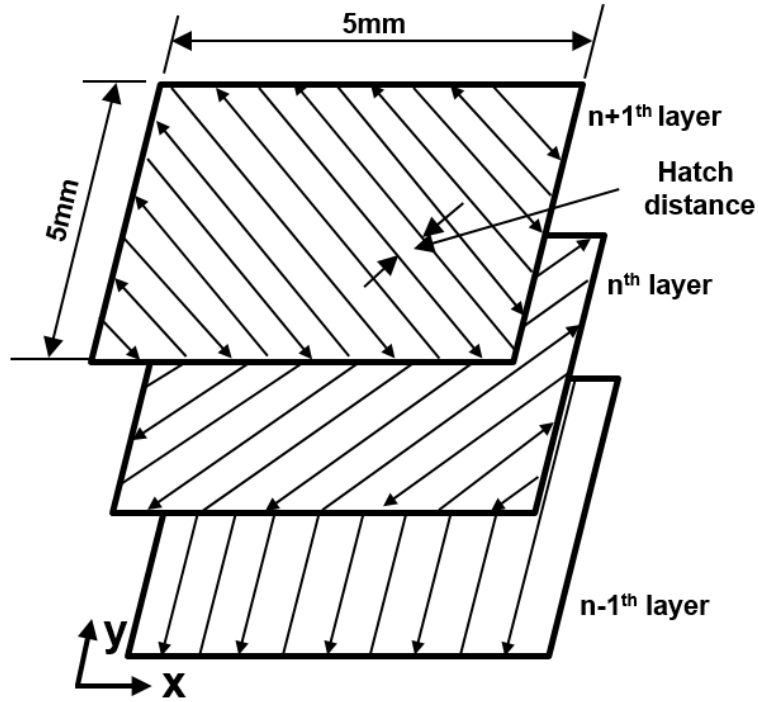


Figure 2.5: Laser scanning strategy in which the hatch angle rotates 67 degrees after every layer

Table 2.3: SLM process parameters used for experiment dataset

Factor	Unit	Level
Laser power	W	80, 100, 120, 140, 150, 160, 170, 180
Laser velocity	mm/s	800, 1000, 1200, 1400, 1600, 1800, 2000, 2500
Layer thickness	μm	20, 40, 60, 80
Hatch distance	μm	30, 40, 50, 60, 70, 80, 90, 100

2.3 Data collection

2.3.1 Surface roughness measurement

Surface roughness, S_a , is the extension of R_a (arithmetical mean height of a line) to a surface. It is the extrapolation of R_a . Instead of measuring many positions and taking an average of R_a value, S_a was preferred for examining the roughness of a printed surface. It expresses, as an absolute value, the difference in the height of each point compared to the arithmetical mean of the surface. This parameter is used to evaluate surface roughness at the top surface of SLMed products. In SLM, the printing part is surrounded by powder particle.

Therefore, the powder particles that do not belong to the scanning area is entrained in the melting area. They are not affected by laser energy but are stuck on the surface because of the melting material. Due to that reason, the roughness of side surfaces do not depend only on process parameters. Therefore, measuring the top surface can generalize the influence of investigating process parameters.

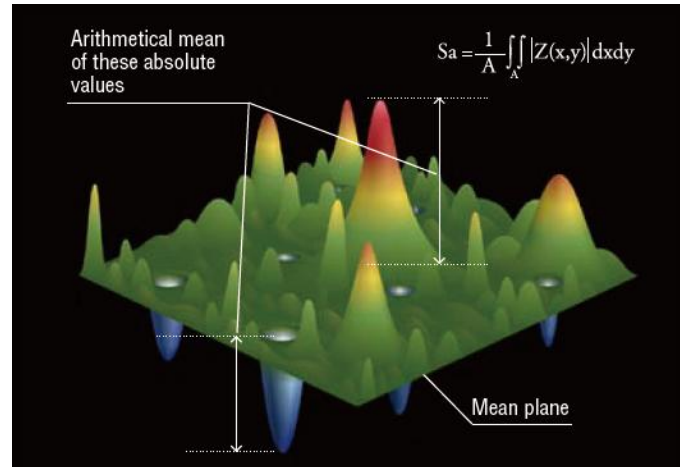


Figure 2.6: Surface roughness measurement principle (*Image source: Keyence.com*)

To measure the printed part, the laser scanning confocal microscope, VK-X200, Keyence Corp., Japan, is used. It is a device for imaging and measurement that enables sample observation using a large depth of focus and 3D measurement simultaneously. The device detects the height information based on the reflected light intensity from the sample. The lens scans the z-axis, and the software stores the data to recreate the 3D image. The objective lens is then driven in the z-axis direction, and the scanning process is repeated to obtain the reflected light intensity at the Z-axis position of each point. With this method, a fully focused light intensity image with height information can be captured. The schematic and real device of the VK-200K is shown in Figure 2.7 a). A pinhole in front of the position where the reflected light from the sample forms an image (photoreceptor) ensures that no light other than that which passes through the focal point of the objective lens reaches the photoreceptor. Use of this type of optics as a sensor allows for the maximum possible amount of light to reach the photoreceptor and allows changes in the focal point to be measured and used as height information. Additionally, the use of this technology in a microscope creates a device with almost no flare and significantly better contrast than a standard optical microscope. It also uses a two-way light source comprised of a laser light source and a white light source.

The two light sources together provide the colours, laser intensity and height information necessary to create deep field colour images, laser intensity images or height images.

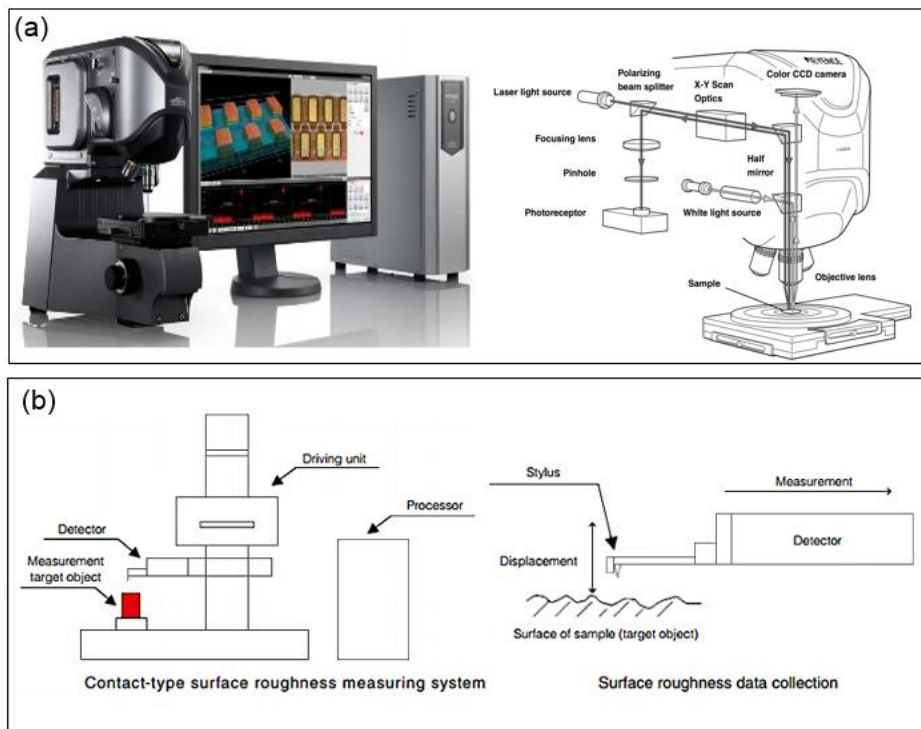


Figure 2.7: (a) The laser confocal microscope VK-200 (left) and its schematic, (b) Contact-type surface roughness instruments [74]

Another method that can be applied to measure surface roughness is a contact-type principle, as indicated in Figure 2.7 (b). With contact-type surface roughness instruments, a stylus tip makes direct contact with the surface of a sample. The detector tip is equipped with a stylus tip, which traces the surface of the sample and electrically detects the vertical motion of the stylus. The electrical signals go through amplification and digital conversion process to be recorded. However, this method requires much time compared to the former method. Therefore, the laser confocal laser scanning microscope is selected.

2.3.2 Density measurement

Theoretically, the density of a solid object is defined as the mass per unit volume of a substance: $\rho = M/V$ in which M and V present the mass and volume of the sample. However, there are some methods to measure it in practice, such as X-Ray CT scanning, microscopic analysis of the cross-section. The X-ray CT has been utilized as an innovative non-destructive image method for internal porosity by providing a complete analysis of size, shape, volume

and distribution of pores within the whole analyzed volume [75][76]. An X-ray system includes an x-ray source, a sample manipulator, X-ray detector and computational devices for data processing [77]. During CT scanning, a set of 2D x-ray projection is acquired at various angles. These projects are then used to reconstruct a 3D voxel model of the sample [75]. By applying a grey value threshold to distinguish between air and the sample material, the information about internal porosity can be revealed. Figure 2.8 (left) shows several results of X-ray CT scanning of samples printed by the SLM process.

The microscopic analysis of the cross-section is a simpler method to get insight into the distribution of the pores with any cross-section of a part. The sample is cut, embedded in epoxy resin, ground, sanded with abrasive paper and polished. The method uses an image processing software capable of automatic image stitching. A pre-elaboration of the image is required to remove any residual scratch of the polishing process and to get a binarized image after selecting an appropriate threshold value [78]. The porosity percentage can be calculated as the ratio between black pixels count and white pixels count, while the area of the pore can be evaluated by knowing the pixel of the image as shown in Figure 2.8(right). The method allows for the assessment of pore size and distribution. Another method is gas pycnometer that measures the volume of part and calculates density [79][80].

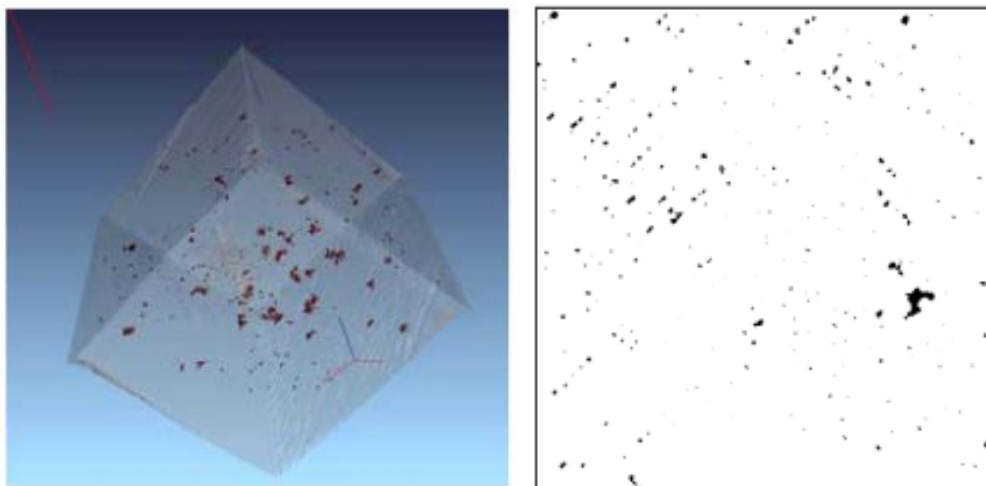


Figure 2.8: (a) A picture of X-ray scanned samples and an image of a cross-section by microscopic analysis [81]

However, the methods described above exist some disadvantages. The drawback of the X-Ray method is that it requires a high skill task for using devices. Significantly, the troubles of

using this method are high cost and high time usage. The cross-section analysis method confirms to specific sections of the specimen, resulting in not representing the entire part. Additionally, the technique demands a high cost also in terms of materials and time usage, namely mounting, and polishing. Because of these drawbacks, they could not be applied to collect data for deep neural network application that ask substantial training data. Therefore, for fast and reducing expenditure, the Archimedes' measurement method was used in this thesis.

Archimedes' principle is a method in which a body's immersion in a given fluid will produce a buoyant force acting upward on it [81] [82]. This force is equal to the weight of the displaced fluid. Advantages of the methods are, it is non-destructive, relatively inexpensive and quick. The density of a solid can be calculated by the weight of the sample in air, weight of the sample in liquid and the density of the liquid.

$$\rho_{measured} = \frac{A}{A-B}(\rho_0 - d) + d \quad 2.1$$

Where ρ : the density of the printed part; A : the weight of sample in the air; B : the weight of sample in liquid; ρ_0 : density of liquid; d : density of air (approximate 0.001 g/cm^3)



Figure 2.9: Analytical balancer and Ad-1653 kit

For more convenient, density measurement toolkit was used. A GR-200 balancer, A&D co., Ltd., combined with AD-1650 density determination kit, as shown in Figure 2.9. The measuring accuracy of the balancer has $\pm 0.0001\text{g}$. The device records weight of part put on the upper pan and the lower pan and calculates density and displays on the screen automatically.

The density ratio of printed parts was calculated as:

$$\text{density ratio} = \frac{|\rho_{\text{measured}} - \rho_{\text{standard}}|}{\rho_{\text{standard}}} \cdot 100 \quad 2.2$$

Chapter 3. EFFECTS OF PROCESS PARAMETERS ON THE PRINTED PRODUCT QUALITIES

3.1. Consolidation behaviour during SLM

SLM is technological processing. It includes a complex physical phenomenon that adds difficulties of interaction between process parameters. During the SLM process, there are three types of powder consolidation mechanism. The first is solid-state sintering that occurs when the temperature is below the melting temperature of material powder. The mechanism creates necks between adjacent particles, as shown in the dash-white rectangle (a) of Figure 3.1. The second is partial melting shown in dash-white rectangles (b) of Figure 3.1. It happens when the heat applied to powder particle is not enough to melt the whole particle. Only the shell is melted while the core of grain remains solid. And it also forms necks between particles. The mechanism also happens when the powder has a bi-modal distribution. The SLM powder exists in particle size distribution. While small particles are melted, the larger powders are still solid. The third primary consolidation mechanism in SLM is full melting. It is achieved when the heat source applied is enough to melt powder completely as illustrated in a dash-white rectangle (c) of Figure 3.1. Therefore, depending on the supplied energy that is represented by process parameters, three types of consolidation mechanism could happen. If energy provided is not enough, which is indicated by decreasing laser power or increasing laser scanning speed and layer thickness as well as hatch space, material powder suffers the first and the second type of solidification.

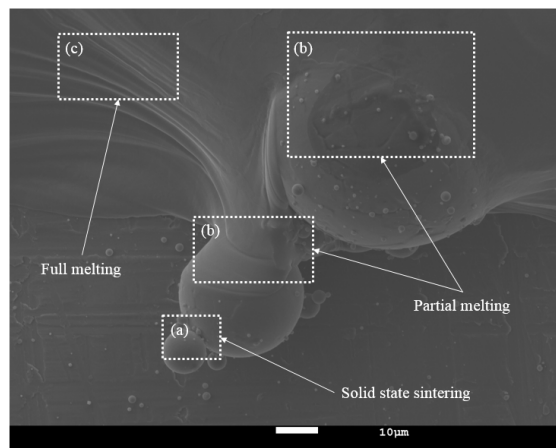


Figure 3.1: Consolidation mechanisms during SLM processing: (a) solid-state sintering, (b) partial melting, and (c) full melting.

3.2. Penetration

After melting a single line, the melt pool geometry, namely width and depth, was formed, as shown in Figure 3.2a). During the SLM process where multi-lines and layers were combined, depending on energy and interaction time of laser to material power, the current printing will penetrate the previously printed layer or not. Penetration is expressed by ratios of depth and width of melt.

If laser energy and interaction time with material powder are not sufficient, scanned lines are possibly melted but could not reach to the underneath layer, as indicated in Figure 3.2(b). Figure 3.2(b) reveals clearly each layer with large porosity. In this case, the penetration ratio is low. Moreover, if the laser energy is high enough and the interaction time short enough to create a depth of a full molten pool that is higher than layer thickness, the printing lines/layer will secure connection with the solid layer underneath as shown in Figure 3.2(c). Therefore, the penetration is accompanied by laser energy and reaction time between laser spot and material powder that was represented by laser power and laser scanning speed initially.

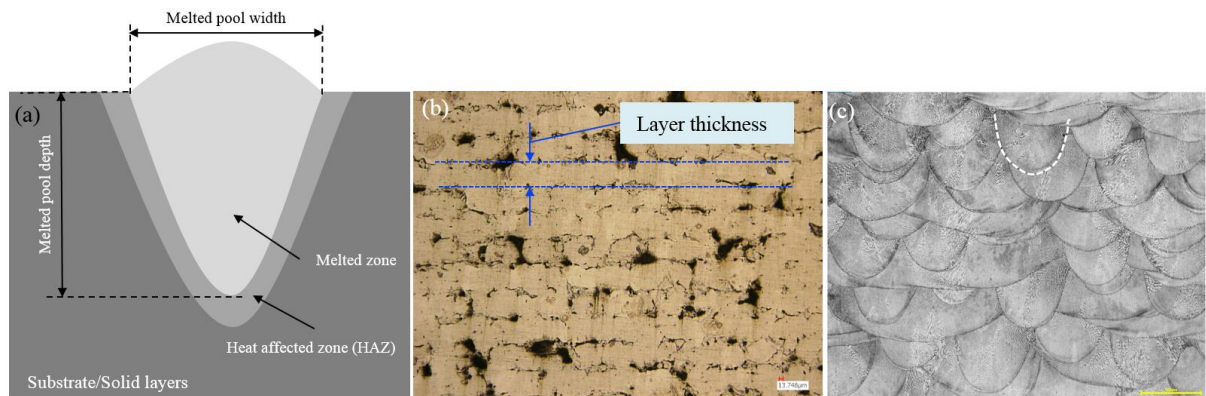


Figure 3.2: (a) Schematic of a melt pool geometry and (b) geometry of a melt pool in practice

However, too high laser energy combining with a low laser velocity creates an unwanted phenomenon that is called the keyhole, or deep penetration. Due to laser energy, the material will be heated and melted at the beginning. Increasing laser energy will make a molten material vapour that presents the recoil pressure. The recoil pressure depresses the molten pool that forms the keyhole. Moreover, the vapour pressure tends to push molten metal out from the keyhole what makes the keyhole deeper. Figure 3.3 indicates the process of keyhole formation. Due to the influence of surface tension gradient, the molten metal flows along the

melted pool edge at the rear part of the top surface, as shown in Figure 3.3(a). Due to vapour pressure, the molten metal at the bottom moves along the fusion line and then swerves to the centre of the melting pool on the back-keyhole wall. Meanwhile, the upper molten metal flows downward in the direction of the keyhole wall and returns to the bottom centre of the melting pool that creating a vortex. Because of the vortex, a bulge is formed, as illustrated in Figure 3.3(b). It will flow forward and touch the front keyhole wall to create a bridge. The bridge traps the bubbles. The trapped bubbles will follow the flow of molten fluid upward of the surface. If its speed is higher than the solidification rate, it will escape the liquid and distribute to the spattering phenomenon, as indicated in Figure 3.3(c). Otherwise, the bubble will be captured in the solid metal after laser passes away, as revealed in Figure 3.3(d) [83]. The porosity is mainly distributed at the bottom of the weld. Because at this location, the solidification rate and temperature gradient are larger than other positions.

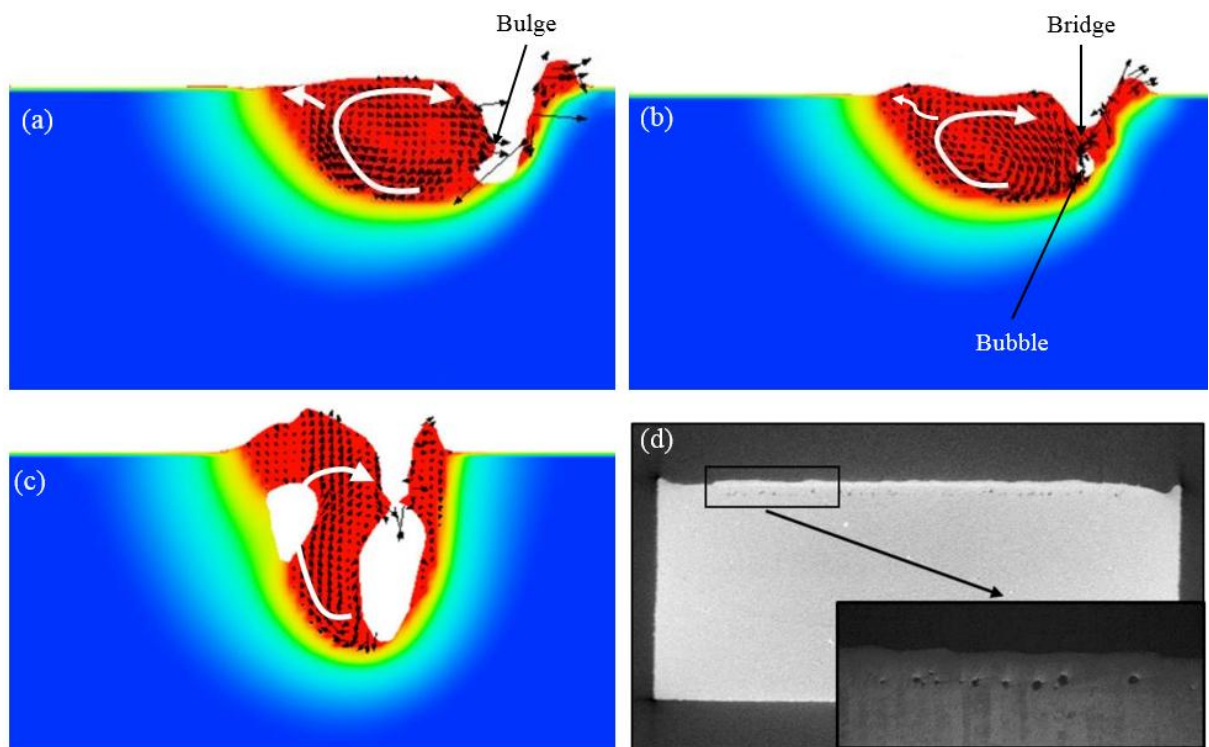


Figure 3.3: (a) Fluid dynamic of melting pool formed the keyhole, (b) bridge formation creating a bubble, (c) bubble flow [83], porosity resulted in keyhole effect in practice [84]

Before the keyhole formation, the laser is irradiated on the surface of the metal. However, when the keyhole is existing, the laser is irradiated through the vapour, and multiple reflections of the laser are achieved. Therefore, the evaporation increases the efficiency of laser energy absorption, which leads to intensifying of the keyhole depth. The penetration

associating with process parameters is gradually becoming shallower with increasing laser speed and deeper keyhole with an increase in laser power [85]. A shallow penetration result that babbles escapes more easily, which reduces porosity in the printed part [86] [87].

3.3. Material ejection

Material ejection of which schematic, shown in Figure 3.4, is a common phenomenon in powder bed fusion [88][89]. When the high laser energy is scanning on the surface of the powder bed, particles which are covered by the laser spot are molten, evaporates and forms a plume. The flow creates entrainment of powder particles. It is induced by the intensive evaporation that exists within the laser spot and pressure drop inside the associated vapour jet due to the Bernoulli effect [90]. Some surrounding particles can be consumed through direct contact with the liquid metal and capillary forces. The vapour-driven entrainment causes particles to not only be incorporated into the melt track but also be ejected vertically and rearward, relative to the laser scan direction [90]. The vaporized material is cooled down and condenses, forming particles with the size is bigger than the raw powder.

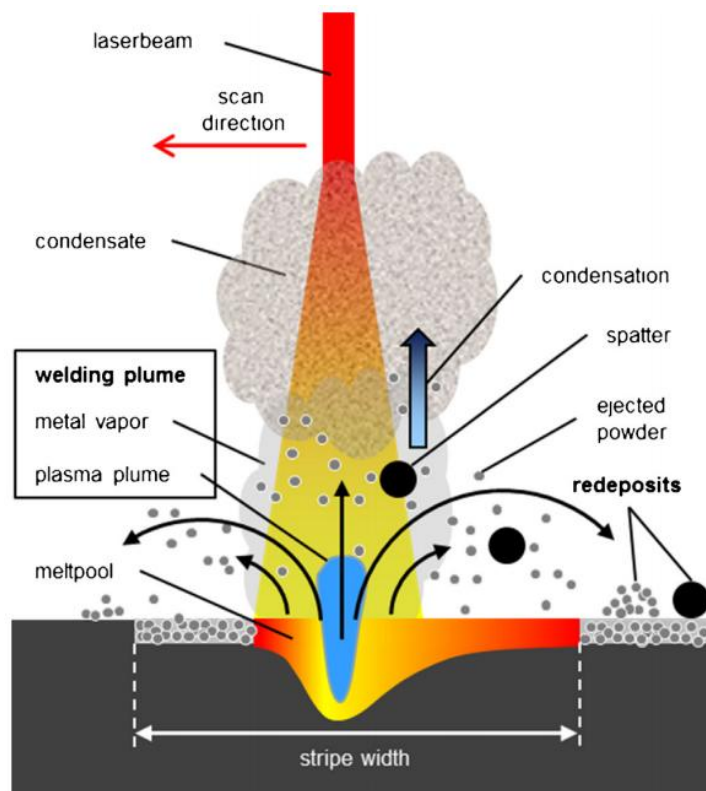


Figure 3.4: Schematic of spattering and powder ejected during the SLM process [91]

A Photron FASTCAM SA3 model 120k, ultra-high-speed frame rate camera, was applied to observe the ejection of powder. Figure 3.5a shows an observation in the X-direction scanning of the laser, while Figure 3.5b shows an observation in the Y-direction, indicating that powder particles were also ejected far away from the centre of the melt track.

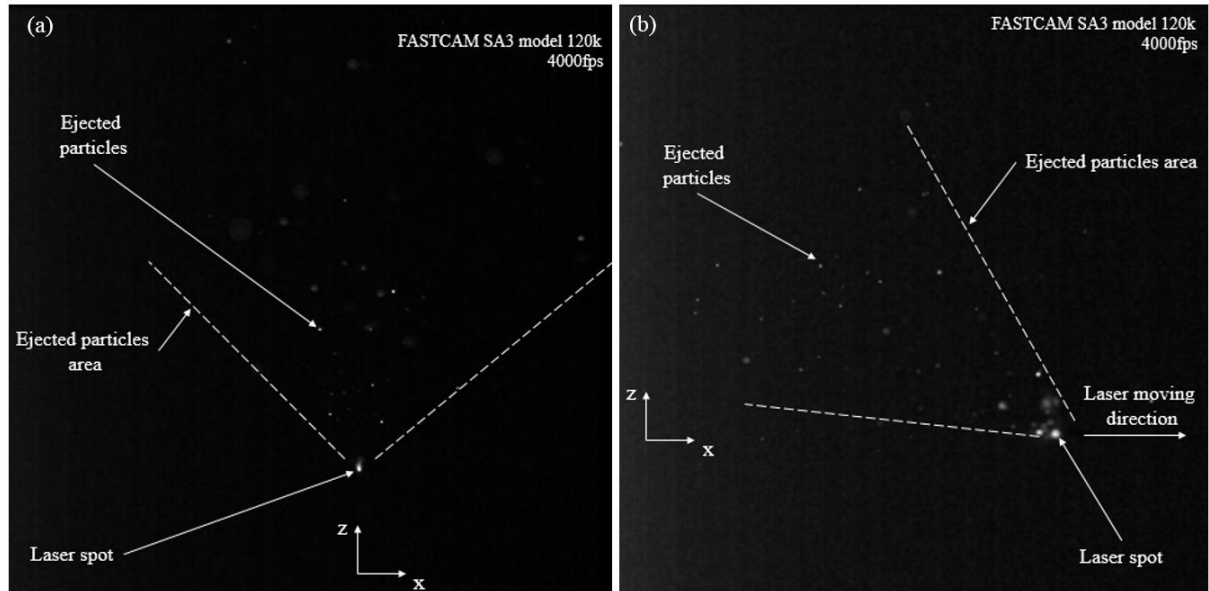


Figure 3.5: Ejected particles upon (a) X-direction laser movement and (b) Y-direction laser movement

Ejected particles located in the manufacturing area will contribute to the next layer of the manufacturing process. The redeposited material may lead to an increase in the layer thickness locally [91]. The increasing lack of fusion in the built material reduces the connection of the molten material to the previous layer, and finally increases the porosity of the printed part. Additionally, the redistribution of ejected particles from the powder bed surface aggravates the balling effect; a particular phenomenon in SLM [92]

3.4. Balling phenomenon

Balling formation is the experience of melted material solidification into spheres instead of a continuous line or solid layer [93], which results in poor surface roughness of the finished part. This happens when these parameters are not correctly chosen. Due to the surface tension, the molt material tends to form the sphere liquid instead of spreading on the underneath layers [94]. Figure 3.6 indicates a schematic of balling formation during the SLM process. At

a specific step, the n^{th} layer, a powder layer with a thickness of t is formed on the substrate, as shown in Figure 3.6 (a). After the laser spot passed, the balls were formed, Figure 3.6 (b). In the $(n + 1)^{\text{th}}$ layer, as a result of existing balls of the previous layer, there is some position nearby the balls that could lead pores potentially after laser impinging.

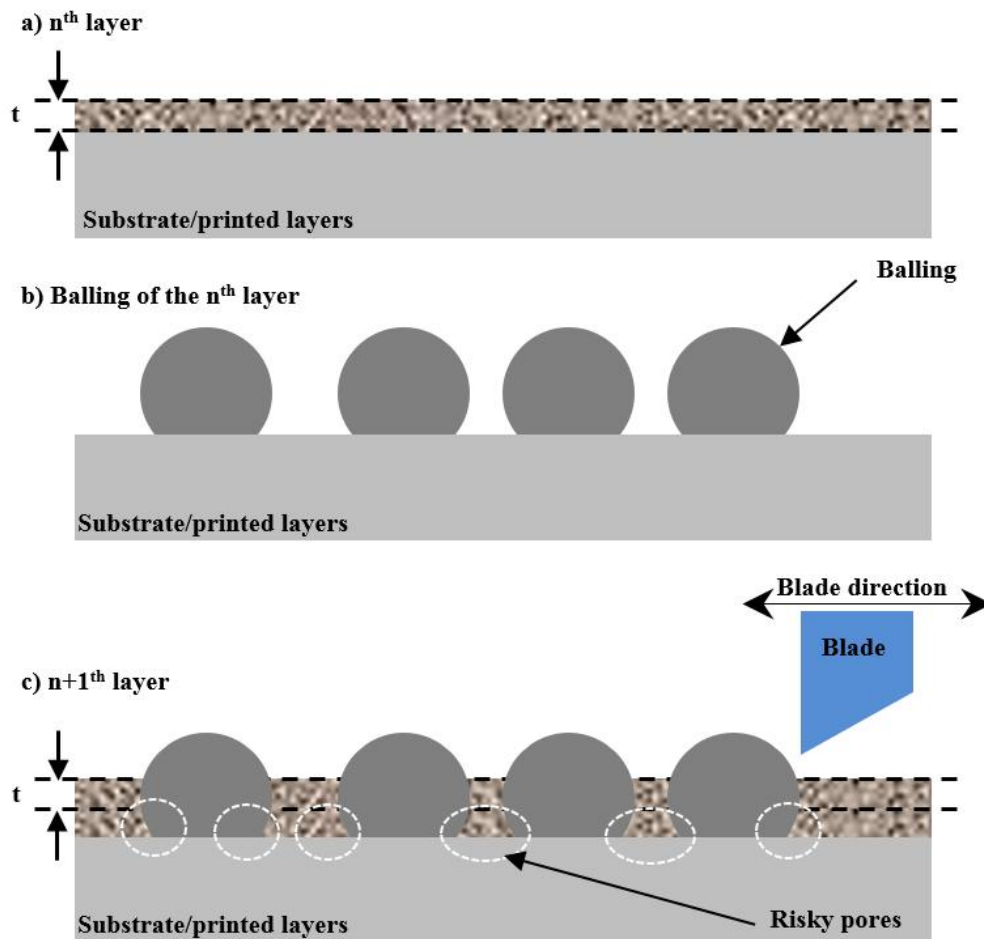


Figure 3.6: Schematic of the balling behaviour: (a) n^{th} powder layer scrapping, (b) Balling formation after processing, and (c) risky pores in next layer printing

As the laser scanning is performed line-by-line and the laser energy causes melting along a row of powder particle, the liquid can spread solid structure particle into a continuous shape. Due to an enhanced capillary instability effect caused by a high scanning speed, the liquid quickly collects at the contact between solid particle. A significant transverse shrinkage distortion tends to occur in inter-particles areas. Therefore, it results in the breakage of the liquid film. Additionally, the radial contraction of the liquid line will also occur. It leads to change in the direction of fluid flow from outward to inward. Eventually, it causes balling [95].

Figure 3.7 illustrates the SEM images of the single-track morphology at different laser power and laser speed. It shows that the single scanned line will form in discontinuous line when laser speed increases, shown in Figure 3.7 a, b or when laser power decreases, indicated in Figure 3.7 c, d.

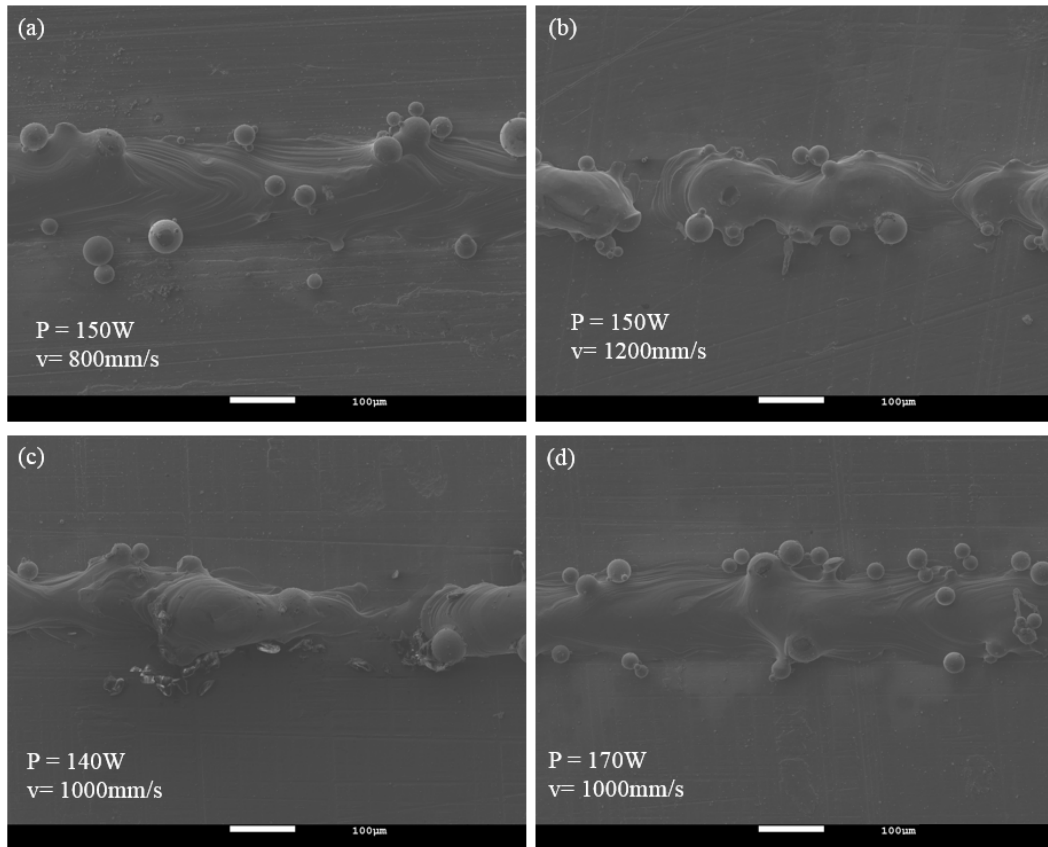


Figure 3.7: Morphology of single scanned track influenced by laser velocity (a, b) and laser power (c, d): (a) $P = 150\text{W}$; $v = 800\text{mm/s}$, (b) $P = 150\text{W}$; $v = 1200\text{mm/s}$, (c) $P = 140\text{W}$; $v = 1000\text{mm/s}$, and (d) $P = 170\text{W}$; $v = 1000\text{mm/s}$

The input energy presented by laser power and laser speed significantly affects the response of a powder bed system. During the analysis of single scanned track, it was figured out that the process parameters have a threshold to form a stable zone, as shown in Figure 3.8. The white zone and the orange zone present areas where the tracks are continuous and discontinuous, respectively. The first zone occurs at high laser speed or low laser power. It reveals that increasing laser power increases the range of laser velocity that makes continuous tracks. Additionally, decreasing laser velocity results in a broader range of the accepted laser power that could print good lines.

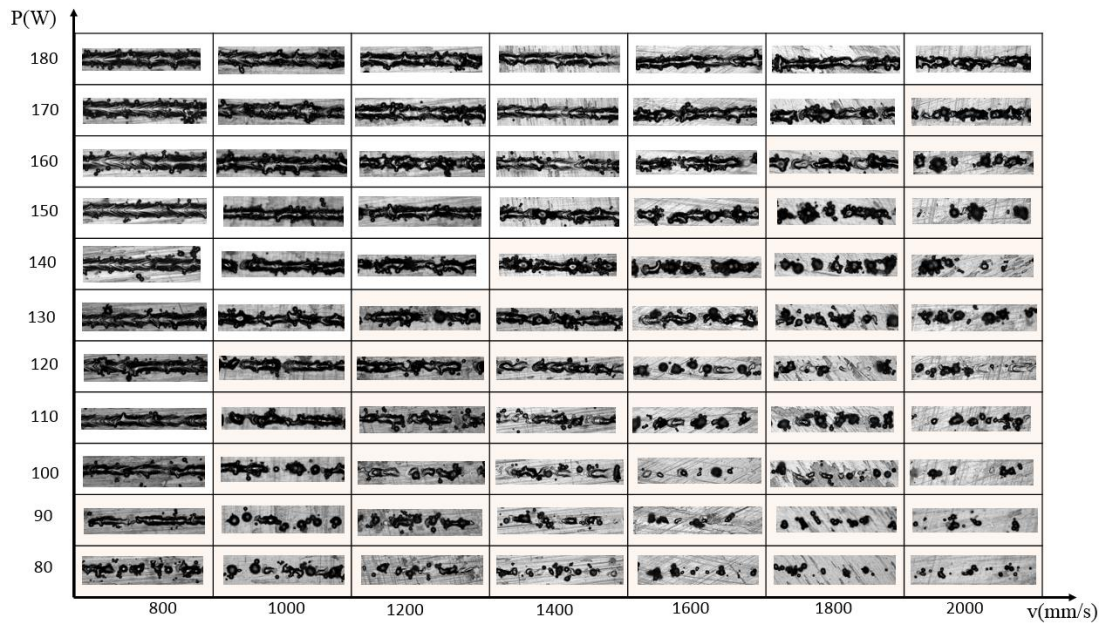


Figure 3.8: Single scanned line morphology due to various laser power (P) and laser scanning speed (v).

However, printing an entire product must consider the hatch distance, h , and layer thickness, t , and that the result showing in Figure 3.8 was not mentioned. At the same laser power $150W$, and laser speed $1200mm/s$, Figure 3.7 b indicated a discontinuous scanned line, but a smooth surface of a layer when scanning multi-lines was considered, as shown in Figure 3.9a. Figure 3.9a and b show a different morphology of surface scanned layers, of which layer thickness was $20\mu m$ and $60\mu m$, respectively. It exposes that layer thickness also affects the balling because increasing layer thickness reduces energy supplies to materials powder. The hatch distance (space between two adjacent lines) influences the portion of overlap melting of the current line to the previous line.

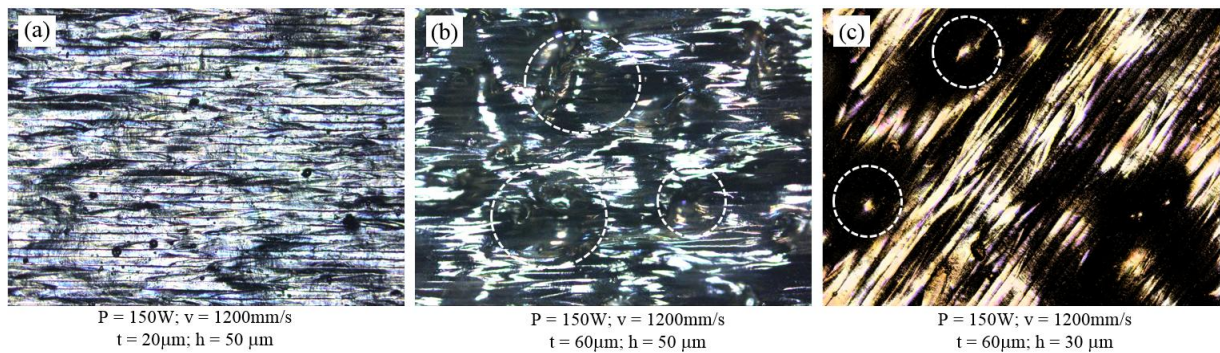


Figure 3.9: Surface roughness of a printed layer at a different layer thickness (a) $t = 20\mu m$, (b) $t = 60\mu m$; and different hatch distance (b) $h = 50\mu m$, (c) $h = 30\mu m$.

Figure 3.9(b) and (c) depict surface morphology of a scanned layer that used hatch space at $50\mu m$ and $30\mu m$. It clarifies that a smaller hatch distance will increase the re-melting area that results in reducing the balling. Therefore, investigating a single track could not be represented for the whole of the SLM process. Multi-parameters must be examined simultaneously.

Material ejection and balling result in a poor morphology of the scanned surface. If ejected particles or balling are small enough, they will be melted by the laser source as the initial powder. However, if their size exceeded the diameter of the laser spot, the laser energy will not melt them, and that decreases part density. This observation leads to porosity inside manufactured products, as explained in Figure 3.10. The dashed circles indicated the porosity that was created by the relocated particles. The surface roughness of the printed part also increases due to partially melted, and entrained powder particles that stick and agglomerate to the surface of the solidified melt pool [96][97].

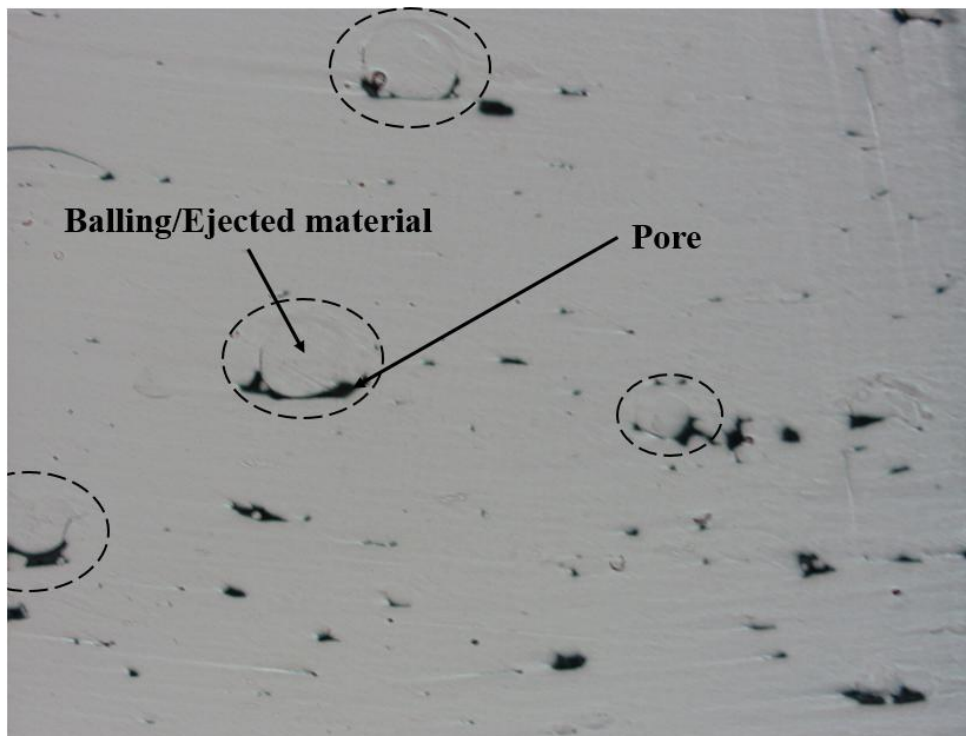


Figure 3.10: Porosity of a printed part due to balling and spattering behavior.

More importantly, the balling phenomenon could be aggravated. It leads to increasingly worse powder bed as well as the printed part quality, as demonstrated in Figure 3.11(a) and (b). In terms of technical processing, the balling could cause trouble for the printing process.

The process could be stopped due to conflict between the blade and humped section if the blade is made of metal. If it is a plastic blade, it could be torn, and result in a bad powder bed, as shown by scratched lines in Figure 3.11(a) and create a fail printed product as indicated in Figure 3.11(b).

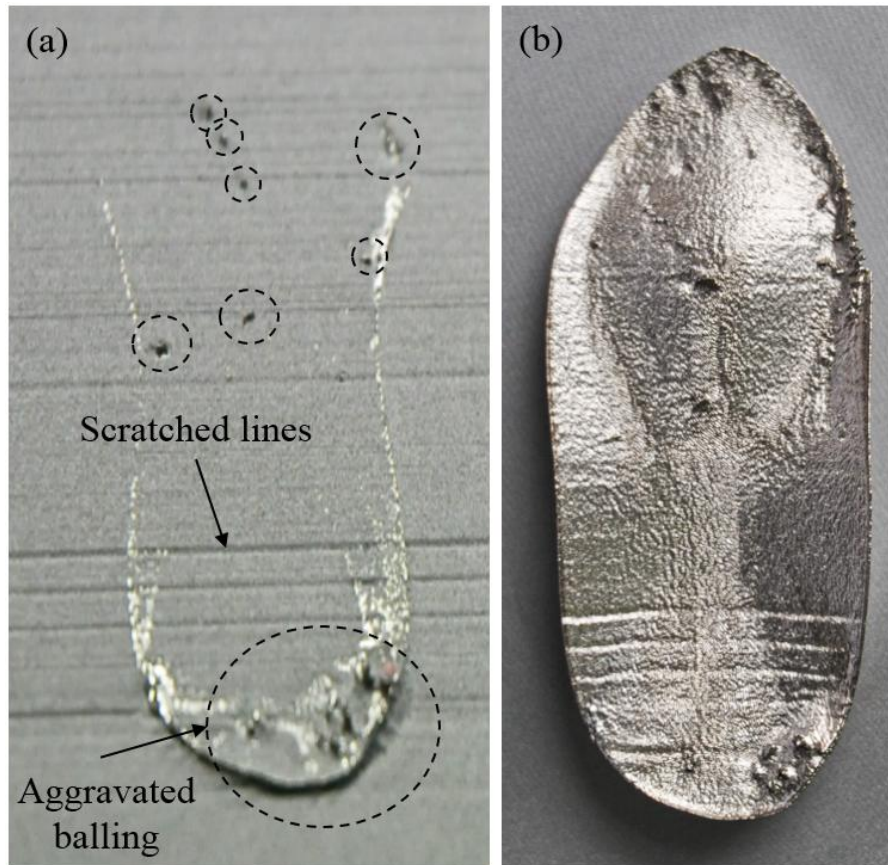


Figure 3.11: (a) Aggravated balling (dash-black circles) and scratched lines on the powder bed surface, (b) a printed part affected by balling.

Chapter 4. DESIGNING OF THE DEEP NEURAL NETWORK

In this chapter, building deep learning for prediction will be illustrated. Because each input data associates with target output data that were labelled, the supervised training is applied. The network is a fully connected layered feedforward with backpropagation in which every node in a layer is connected to all node in the next layer. The connection between neurons is in one direction. There is no connection between the units in the same layer. By a given input, the output of the network is compared with a given target, and the error between the target output and the model output through the system was backward. There are two phases to build a network: learning and generalization processes. The first stage is typically accomplished by using examples and finding suitable network parameters using learning algorithms. The learning process will stop once the network is trained to achieve the desired performance. After training, a net model represents a complicated relationship, and then it can be used directly.

4.1. Artificial neuron mathematics

a) Perceptron

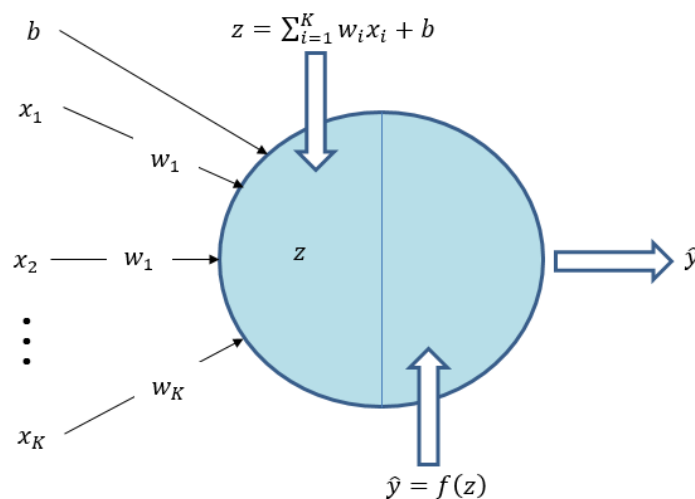


Figure 4.1: A working of a perceptron

$$z = \sum_{i=1}^K w_i x_i + b = w_1 x_1 + w_2 x_2 + \dots + w_i x_i = \mathbf{w}^T \cdot \mathbf{x} + b \quad \left. \vphantom{z} \right\} \quad 4.1$$

$$\hat{y} = f(z)$$

Perceptron [98] is the earliest and simplest neural network that includes a single input layer and one output unit without hidden layers. It is used for linear problems only. Each neuron receives a set of input values, x , and calculate the predicted value \hat{y} . Vector $\mathbf{x} = (x_1, \dots, x_K)$ contains the amount of the features in K examples from the training set. Each of the neurons has its own set of parameters, namely, weight vector, \mathbf{w} , and bias, b , which changes during the training process. In each iteration, the neuron calculates a weighted average of the value of vector \mathbf{x} , based on its current weight vector \mathbf{w} and adds bias. Finally, the calculated result is passed through an activation function, $f(\cdot)$. The working principle of a unit was shown in Figure 4.1 followed by Equation 4.1. Where \mathbf{w}^T is a transpose matrix.

b) Single-layer

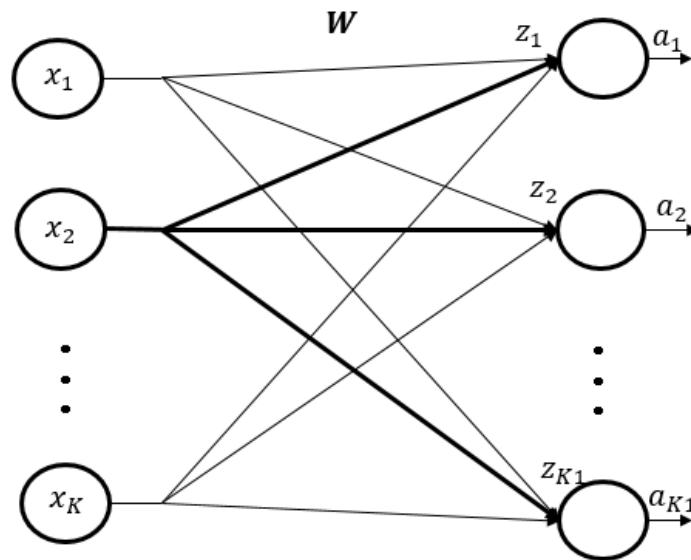


Figure 4.2: Working of a single layer

Figure 4.2 indicates the working in a single layer, and the system state is updated by:

$$\left. \begin{aligned} \mathbf{z} &= \mathbf{W}^T \mathbf{x} + \mathbf{b} \\ \mathbf{a} &= f(\mathbf{z}) \end{aligned} \right\} \quad 4.2$$

Where $\mathbf{a} = \{a_1, \dots, a_{K1}\}$ is a vector output of nodes; \mathbf{b} is a vector bias; \mathbf{z} is a vector input of neurons. The \mathbf{x} vector is the input layer. $f(\cdot)$ is the activation function. Units work a similar calculation:

$$\left. \begin{aligned} z_i &= \mathbf{w}_i^\top \mathbf{x} + b_i \\ a_i &= f(z_i) \end{aligned} \right\} \quad 4.3$$

c) Deep neural network

The deep neural network, shown in Figure 4.3, is a network that contains several hidden layers. The input layer is considered as 0^{th} layer.

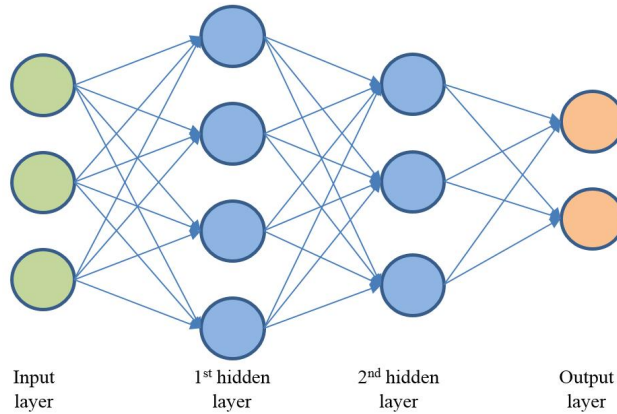


Figure 4.3: A example of deep network

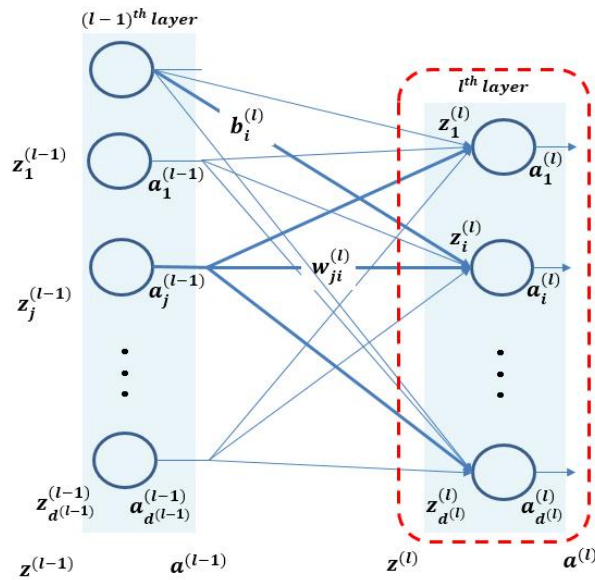


Figure 4.4: Working of a deep neural network

The working of the deep neural network was shown in Figure 4.4, accompanied by the following:

$$z_i^{(l)} = \sum w_{ji}^{(l)} a_j^{(l-1)} + b_i^{(l)} = \mathbf{w}_i^{(l)\top} \mathbf{a}_j^{(l-1)} + b_i^{(l)} \quad 4.4$$

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)\top} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \quad 4.5$$

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{z}^{(l)}) \quad 4.6$$

Whereby, $w_{ji}^{(l)}$ denotes connection from the node j^{th} in the $(l-1)^{th}$ to the node i^{th} in the l^{th} layer ; $z_i^{(l)}$ is a scalar input of the node i^{th} in the layer l^{th} ; $\mathbf{z}^{(l)} = \{z_i^{(l)}, \dots, z_{d^{(l)}}^{(l)}\}$ is a vector input of the l^{th} layer; $\mathbf{a}^{(l)} = \{a_i^{(l)}, \dots, a_{d^{(l)}}^{(l)}\}$ is an output vector of the l^{th} layer, $\mathbf{a}^{(l)} \in \mathbb{R}^{d^{(l)}}$; $a_i^{(l)}$ is the output of the node i^{th} in the layer l^{th} ; $d^{(l)}$ is the number of nodes in the l^{th} layer; $\mathbf{w}_i^{(l)}$ is weight vector of the node i^{th} in the l^{th} layer, while $\mathbf{W}^{(l)}$ is matrix-vector of the l^{th} layer, $\mathbf{W}^{(l)} \in \mathbb{R}^{d^{(l-1)} \times d^{(l)}}$; $\mathbf{b}^{(l)}$ is a bias vector of the l^{th} layer, $\mathbf{b}^{(l)} \in \mathbb{R}^{d^{(l)} \times 1}$

During the training process of a deep neural network, the training set is often vast data. Assuming the training data has N samples with n_x features each, putting together the vertical vectors \mathbf{w} , \mathbf{a} , \mathbf{z} of each layer creates the \mathbf{W} , \mathbf{A} , \mathbf{Z} matrices, respectively.

4.2. Data pre-processing

The inputs of the network (investigated SLM process parameters) have different units and scale values, that may increase the difficulty of the problem neural network is trying to model. The system will get poor performance during learning and high generalization error. Pre-processing of data is to convert the raw data to a new representation before being sent to a network. It significantly affects the generalization performance of a neural network. By removing the redundancy in the input space, the network could achieve a better generalization capability [99]. Learning algorithms work better or converge faster when features are on a relatively similar scale and are close to normally distributed. In this thesis, the Standard Scaler was used. For $\{\mathbf{x}_i | i = 1, \dots, N\}$, N : total training samples. The transformed data were defined by the following equation:

$$z = \frac{x_i - \mu}{\sigma} \quad 4.7$$

With mean:

$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i) \quad 4.8$$

and standard deviation:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2} \quad 4.9$$

N : the number of samples

4.3. Deep neural network architecture

Deciding the number of neurons in the hidden layers is a significant part of determining the overall deep network architecture. Though these layers do not directly interact with the external environment, they have a tremendous influence on the final output. Using a too thin network will result in underfitting. It occurs when there are too few neurons in the hidden layers to adequately detect the signal on a complicated data set. Similarly, using a too thick system can result in overfitting. Overfitting happens when the network has too much information processing capacity that the limited amount of information contained on the training set is not enough to train all the neurons in the hidden layers. Moreover, a large network can increase the time to train the network.

There are many methods for determining the suitable number of the hidden layers and neurons in the hidden layers. The quantity of neurons in the input layer is equal to the number of features that are the investigating parameters (laser power, laser scanning speed, hatch distance and layer thickness). The total of neurons in the output layer is equal to the number of analyzed printed parts properties as density and surface roughness. The number of hidden layers should be set between the size of the input and size of the output. The number of hidden nodes was calculated by:

$$N_h = \frac{N_s}{(\alpha(N_i + N_o))} \quad 4.10$$

N_i : number of input neurons; N_o : number of output neurons; N_s : number of samples in training data set; α : an arbitrary scaling factor, usually 2-10. With four input nodes and two output nodes, the maximum hidden units should not exceed 160 nodes.

Other techniques proved that the network capacity should be at least several times smaller than the size of the training set: $\frac{N_w}{N_y} \ll N$, where N_w , N_y , N are the total number of weights, number of output and the total number of training samples [100]. $N_w = (I + 1)H_1 + (H_1 + 1)H_2 + (H_2 + 1)O$, where I : total number of input; H_1 , H_2 : total hidden nodes in the first and the second hidden layer; O : number of output. With approximately 2000 collected samples for training, indicated in Table 2.3; four input nodes and two output nodes, the total hidden unit should be smaller than 120 units. In this thesis, initial architecture was designed to include two hidden layers and 60 hidden nodes for each hidden layer.

4.4. Activation functions

Activation functions are important for a deep network to learn and map between the features and response variable. Their main purpose is to convert an input or set of input signals of a node in a deep network to an output. That output signal uses input in the next layer in the stack. Particularly, in deep learning, we do the sum of products of inputs (X) and their corresponding weights (W), and apply an activation function ($f(x)$) to it to get an output of that layer, and then feed it as an input to the next layer. Without an activation function, the network would become a combination of linear functions and would not be able to learn and model other complicated data. Another important property of an activation function is that it should be differentiable, in order to compute the gradient of loss for optimizing network parameters that will be presented in the next section.

There are some popular types of activation function such as sigmoid, tanh and rectified linear units (ReLU). The ReLU – Rectified Linear Units, Figure 4.5a, was used in the hidden layers [101]. It is a piecewise linear function which prunes the negative part to zero and retains the positive part. The ReLU has become very popular in the past couple of years [102]. It was recently proved that it allows computing much faster than the sigmoid or tanh function [103]. It is very simple and efficient. The ReLU function is:

$$f(x) = \max(0, x) \quad 4.11$$

The derivative of ReLU function is given as 1 for $x > 0$ or 0 for $x < 0$. Therefore, the ReLU has a constant zero gradient whenever a unit is inactive.

The sigmoid activation function [104][105], Figure 4.5b, is traditionally a very popular activation function for neural networks [106]. The input to the function is transformed into a value between 0.0 and 1.0. Inputs that are much larger than 1.0 are transformed to the value 1.0. Similarly, values much smaller than 0.0 are snapped to 0.0. The shape of the function for all possible input is an S-shape from zero through 0.5 to 1.0. The sigmoid function was applied in the output layer only [107][108]. The sigmoid function is:

$$f(x) = \frac{1}{1+e^{-x}} \quad 4.12$$

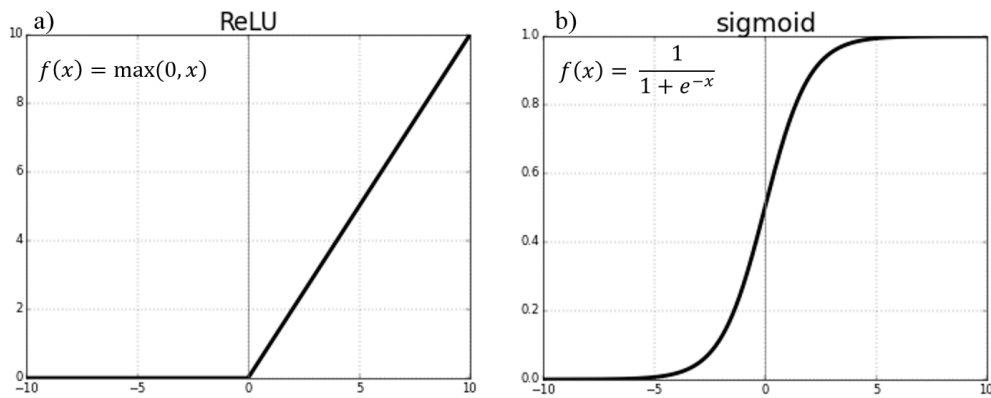


Figure 4.5: ReLU (a) and sigmoid (b) activation function

4.5. Cost function

Fore training the deep network, a loss function is used. The basic consideration of information on the training process is the evaluate of the loss function. The goal of the optimization network process (training state) is to find weights and biases that minimize the cost function. In this thesis, for an easier process in the training step that uses a gradient, the cost function uses the mean square error (MSE), calculated by:

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad 4.13$$

Where, N : number of training samples; y_i : experimental output; \hat{y}_i : the output of network model; \mathbf{W}, \mathbf{b} : weight matrix, bias matrix, respectively. For the validating step, the mean absolute error (MAE) is used. The formula of MAE is:

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad 4.14$$

4.6. Training a deep neural network

The training process is about minimizing the cost function by changing the values of the \mathbf{W} and \mathbf{b} parameters. The training process aims to learn a function $f: X \rightarrow Y$ from a given data set $\{x_i, y_i | i = 1, \dots, N\}$. Where $x_i \in X$ and $y_i \in Y$ is the known label of x_i . The process adjusts network parameters by direct comparison between the model output and desired output. The measured error, defined by Equation 4.13, is used to guide the learning process.

4.6.1. Deep neural network parameters initialization

Training a neural network is usually iterative and thus requires some specified initial point to begin the iteration. The choice of initial network parameters will influence whether the network model reaches a global minimum of the cost function. The update of the setting between two units depends on both derivatives of the activation function of the upper unit and the lower unit. Therefore, it should avoid initial parameters that would make the activation function or derivative of activation zero. A too large initial weight results in exploding values during forward-propagation or backpropagation calculation. Moreover, it leads the model to saturate where the derivative of the activation function (sigmoid, for example) has a very small value. And being too small causes low learning [109]. In this thesis, to initialize the weights, Glorot uniform initializer was applied [110]. It draws samples randomly from a uniform distribution centred on 0 with standard deviation:

$$std = \sqrt{\frac{6}{fan_in + fan_out}} \quad 4.15$$

Where fan_in , fan_out are the number of input units and output units in the weight tensor respectively. The approach for setting the biases must be coordinated with the approach for setting the weights. Setting the initial biases to zero is compatible with most weight initialization [103].

4.6.2. Training algorithm

a) Gradient descent algorithm

Gradient descent (GD) is one of the most popular algorithms to optimize the neural network. It is a first-order iterative optimization algorithm for finding the minimum of a function. To

find a minimum of a function using GD, one should take steps proportional to the negative of the gradient of the function at the current point. To indicate how GD works briefly, the below example is analyzed. Optimization is a task of minimizing function $f(x)$ by altering x . In the case of a neural network, it is a method to minimize cost function by adjusting network parameters. Assuming there is a parabola function $J(x) = \frac{1}{2}(x - 1)^2 - 2$ of which its graph in the Cartesian coordinate is shown in Figure 4.6.

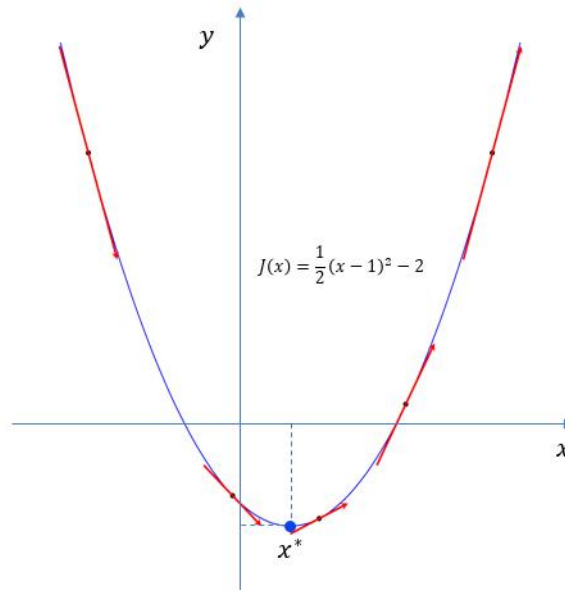


Figure 4.6: An indication of how the Gradient Descent algorithms work by using derivative.

The blue point is a local minimum which makes the $J(x)$ the smallest value, which is the global minimum. In general, the global minimum is a particular case of the local minimum. Note that, a derivation is a term that is calculated as the slope of the graph at a particular point. The hill is described by drawing a tangent line to the graph at the point. Therefore, if it can calculate this tangent line, it possibly is able to compute the desired direction to reach the minima. There are some indications in Figure 4.6:

- The blue point, x^* , makes the derivation of $J(x)$, $J'(x)$, equal to 0.
- Derivation values of the function at the points that are left of the x^* are negative values. Derivation values of the function at the points that are right of the x^* are positive values.
- Derivation value at a point is equal to the slope coefficient of the tangent line at that point.

From Figure 4.6, assuming x_t is a point at the t^{th} loop, the target is bringing the x_t to the x^* as close as possible. Additionally, x_t is on the right of x^* if $J'(x) > 0$ and vice versa. Therefore, the x_{t+1} should move to the left side to get closer the x^* : $x_{t+1} = x_t + \Delta$, in which the sign of Δ value is opposite of the derivation of the function. In general, the new point:

$$x_{t+1} = x_t - \eta J'(x_t) \quad 4.16$$

In which, η is a constant factor, and the minus sign means that the x_{t+1} should move to the opposite direction of the derivation value. This technique is named gradient descent. Applying the GD to the neural network, we have:

$$\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t) \quad 4.17$$

In which, θ : network parameters, w, b ; $\nabla_{\theta} J(\theta_t)$: derivation of the loss function with respect to θ at t^{th} loop; η is called the learning rate.

In deep learning, it always requires finding the global minimum of the cost function. However, the finding is very difficult or even impossible sometimes. Therefore, the solution is finding the local minimum points and then comparing them to find the global minimum. Moreover, the local minimum is a solution to solve the equation $J'(x) = 0$. However, solving the equation $J'(x) = 0$ is extremely difficult due to the derivation complexity, and the dimension of the data or data quantity. Therefore, the gradient descent is a solution to solve the mentioned problem. Starting at a certain point, a loop is applied to approach the global minimum point.

b) Gradient descent with backpropagation

To apply GD as Equation 4.17, the gradient of the cost function of weights matrices $\mathbf{W}^{(l)}$, and vector biases $\mathbf{b}^{(l)}$, must be computed. Assuming $J(\mathbf{W}, \mathbf{b})$ is a cost function, where \mathbf{W}, \mathbf{b} are weight matrices and bias vectors, respectively. Therefore, $\nabla_{\mathbf{W}^{(l)}} J, \nabla_{\mathbf{b}^{(l)}} J, l = \{1, \dots, L\}$ must be calculated. For easier calculation gradient of $J(\boldsymbol{\theta})$, $\nabla_{\boldsymbol{\theta}} J$ ($\boldsymbol{\theta}$ is variances of the cost function) it should be computed from the output layer because it relates to the output of the model. This technique is called back-propagation. Noting that backpropagation refers only to the method for computing the gradient while gradient descent is used to perform learning using gradient [103]. Figure 4.7 will be used to explain the backpropagation.

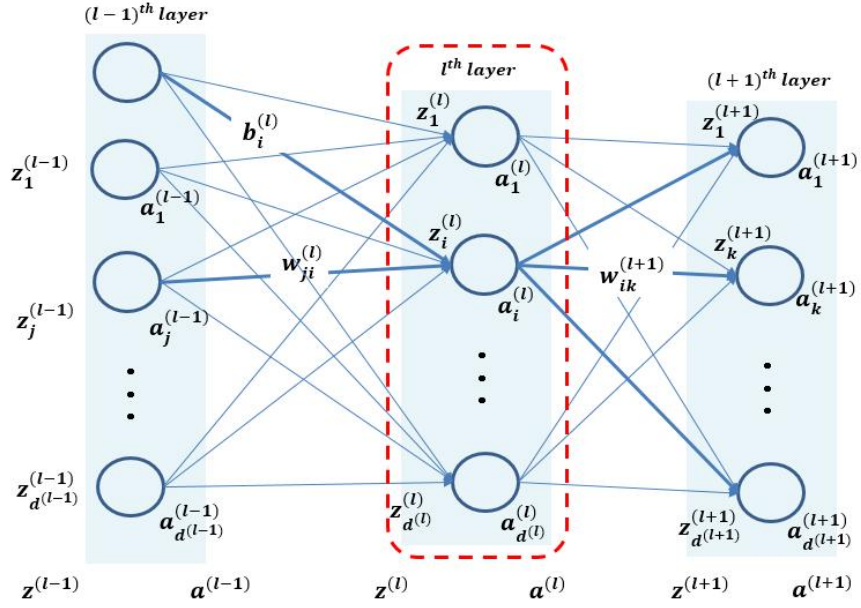


Figure 4.7: Schematic of backpropagation working. The $(l - 1)^{th}$ layer and $(l + 1)^{th}$ possibly are the input layer and the output layer, respectively

For the feedforward process:

- The output of the first layer is the vector input training data:

$$\mathbf{a}^{(0)} = \mathbf{x} \quad 4.18$$

- Input of the l^{th} layer is calculated by the following:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)\top} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)} \quad 4.19$$

- Output of the l^{th} layer is calculated by the following:

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{z}^{(l)}) \quad 4.20$$

- The output of the output layer, L^{th} :

$$\hat{\mathbf{y}} = \mathbf{a}^{(L)} \quad 4.21$$

- Calculation of the cost function:

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - \hat{\mathbf{y}}_i)^2 = \frac{1}{N} \sum_{i=1}^N (\mathbf{y}_i - \mathbf{a}_i^{(L)})^2 \quad 4.22$$

Where N is total training samples.

For the backward process:

At the output layer, L^{th} , the partial derivative of the cost function with respect to a weight w_{ji} is calculated, applying the chain rule, as shown in the following:

$$\frac{\partial J}{\partial w_{ji}^{(L)}} = \frac{\partial J}{\partial z_i^{(L)}} \cdot \frac{\partial z_i^{(L)}}{\partial w_{ji}^{(L)}} = e_i^{(L)} \cdot \frac{\partial z_i^{(L)}}{\partial w_{ji}^{(L)}} \quad 4.23$$

In which:

$$e_i^{(L)} = \frac{\partial J}{\partial z_i^{(L)}} \quad 4.24$$

$$z_i^{(L)} = \mathbf{w}_i^{(L)\top} \mathbf{a}_j^{(L-1)} + b_i^{(L)} \Rightarrow \frac{\partial z_i^{(L)}}{\partial w_{ji}^{(L)}} = a_j^{(L-1)} \quad 4.25$$

Therefore Equation 4.23 becomes:

$$\frac{\partial J}{\partial w_{ji}^{(L)}} = e_i^{(L)} \cdot a_j^{(L-1)} \quad 4.26$$

Similarity, the gradient of cost function by biases at the output layer:

$$\frac{\partial J}{\partial b_i^{(L)}} = \frac{\partial J}{\partial z_i^{(L)}} \cdot \frac{\partial z_i^{(L)}}{\partial b_i^{(L)}} = e_i^{(L)} \quad 4.27$$

For a specific layer $l < L$, as shown in Figure 4.7, we have:

$$\frac{\partial J}{\partial w_{ji}^{(l)}} = \frac{\partial J}{\partial z_i^{(l)}} \cdot \frac{\partial z_i^{(l)}}{\partial w_{ji}^{(l)}} = e_i^{(l)} \cdot a_j^{(l-1)} \quad 4.28$$

Where:

$$e_i^{(l)} = \frac{\partial J}{\partial z_i^{(l)}} = \frac{\partial J}{\partial a_i^{(l)}} \cdot \frac{\partial a_i^{(l)}}{\partial z_i^{(l)}} \quad 4.29$$

In which:

$$\frac{\partial a_i^{(l)}}{\partial z_i^{(l)}} = f^{(l)'}(z_i^{(l)}) \quad 4.30$$

$$\frac{\partial J}{\partial a_i^{(l)}} = \sum_{k=1}^{d^{(l+1)}} \frac{\partial J}{\partial z_k^{(l+1)}} \cdot \frac{\partial z_k^{(l+1)}}{\partial a_i^{(l)}} = \sum_{k=1}^{d^{(l+1)}} e_k^{(l+1)} \cdot w_{ik}^{(l+1)} \quad 4.31$$

Therefore:

$$e_i^{(l)} = \left(\sum_{k=1}^{d^{(l+1)}} e_k^{(l+1)} \cdot w_{ik}^{(l+1)} \right) f^{(l)'}(z_i^{(l)}) \quad 4.32$$

Similarity, we have:

$$\frac{\partial J}{\partial b_i^{(l)}} = e_i^{(l)} \quad 4.33$$

It reveals that calculating the $e_i^{(l)}$ is very important. Moreover, Equation 4.31 shows that to calculate $e_i^{(l)}$, $e_i^{(l+1)}$ needs to be computed. Therefore, following that condition, we need to calculate from the output layer. **Algorithm 1** describes the backpropagation calculation. **Appendix 1** indicates an example to better understand of the backpropagation calculation.

Algorithm 1: Calculation of partial deviation by backpropagation

Required:

+ Network depth: L + $\mathbf{W}^{(l)}$, $l = \{1, \dots, L\}$: the weight matrices of the model+ $\mathbf{b}^{(l)}$, $l = \{1, \dots, L\}$: the biases parameters of the model+ \mathbf{x} : the input in mini-batch to a process; \mathbf{y} : the target output in mini-batch to process

Forward computation:

$$\mathbf{a}^{(0)} = \mathbf{x}$$

for $l = 1, \dots, L$:

$$\mathbf{z}^{(l)} = \mathbf{W}^{(l)\top} \mathbf{a}^{(l-1)} + \mathbf{b}^{(l)}$$

$$\mathbf{a}^{(l)} = f^{(l)}(\mathbf{z}^{(l)})$$

end

$$\hat{\mathbf{y}} = \mathbf{a}^{(L)}$$

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{n'} \sum_{i=1}^{n'} (\mathbf{y} - \hat{\mathbf{y}})^2$$

Backpropagation:

Output layer, L :

$$\mathbf{e}^{(L)} = \nabla_{\mathbf{z}^{(L)}} J \in \mathbb{R}^{d^{(L)}}$$

$$\nabla_{\mathbf{W}^{(L)}} J = \mathbf{a}^{(L-1)} \mathbf{e}^{(L)} \in \mathbb{R}^{d^{(L-1)} \times d^{(L)}}$$

$$\nabla_{\mathbf{b}^{(L)}} J = \mathbf{e}^{(L)}$$

for $l = L - 1, L - 2, \dots, 1$:

$$\mathbf{e}^{(l)} = (\mathbf{W}^{(l+1)} \mathbf{e}^{(l+1)}) \odot f'(\mathbf{z}^{(l)}) \quad (\odot: \text{element-wise produce})$$

Update gradient for weights and biases:

$$\nabla_{\mathbf{b}^{(l)}} J = \mathbf{e}^{(l)}, \text{ biases}$$

$$\nabla_{\mathbf{W}^{(l)}} J = \mathbf{e}^{(l)} \mathbf{a}^{(l-1)\top}$$

end

There are three methods to use for training data. The first mode is that neural network parameters are updated for each sample of training data. Another approach is a form that network parameters are updated for the entire training dataset. However, the first approach leads to noise because it is affected by each and every sample. The second approach results in some bad situations; It is more computationally expensive, requires every iteration which will become very long and significantly longer to train the model. Therefore the third approach, a mini-batch M of n' samples from the training set were selected for every iteration, $n' < N$. During updating the network parameters, the gradient was computed from the mini-batch, rather than the training set and then update network parameters. The cost function at Equation 4.13 becomes:

$$J(\mathbf{W}, \mathbf{b}) = \frac{1}{n'} \sum_{i=1}^{n'} (y_i - \hat{y}_i)^2 \quad 4.34$$

c) Gradient descent with adaptive momentum

Recall Equation 4.17, $\theta_{t+1} = \theta_t - \eta \nabla_{\theta} J(\theta_t)$. Learning rate, η , reflects how much the neural network parameters follow the opposite direction of the gradient. Unfortunately, the hyper-parameters learning rate could be tough to set because it significantly affects the performance of the network. Figure 4.8 and Figure 4.9 indicate the effects of learning rate on the convergence of an example function: $f(x) = x^2 + 5\sin(x)$ of which derivative is $f'(x) = 2x + 5\cos(x)$ and update rule:

$$x_{t+1} = x_t - \eta(2x_t + 5\cos(x_t)) \quad 4.35$$

Figure 4.8 shows the convergence of a red-point of the function at the learning rate $\eta = 0.01$. It shows that if it is set too small, then the parameter update will be very slow, and it will take a very long time to reach the lowest point. It will never reach the minimum point if the iteration is not enough. Otherwise, if the learning rate is set too large at $\eta = 0.5$, as demonstrate in Figure 4.9, then the point will move to the lowest point very fast but over the minima and keep bouncing without ever reaching the minima.

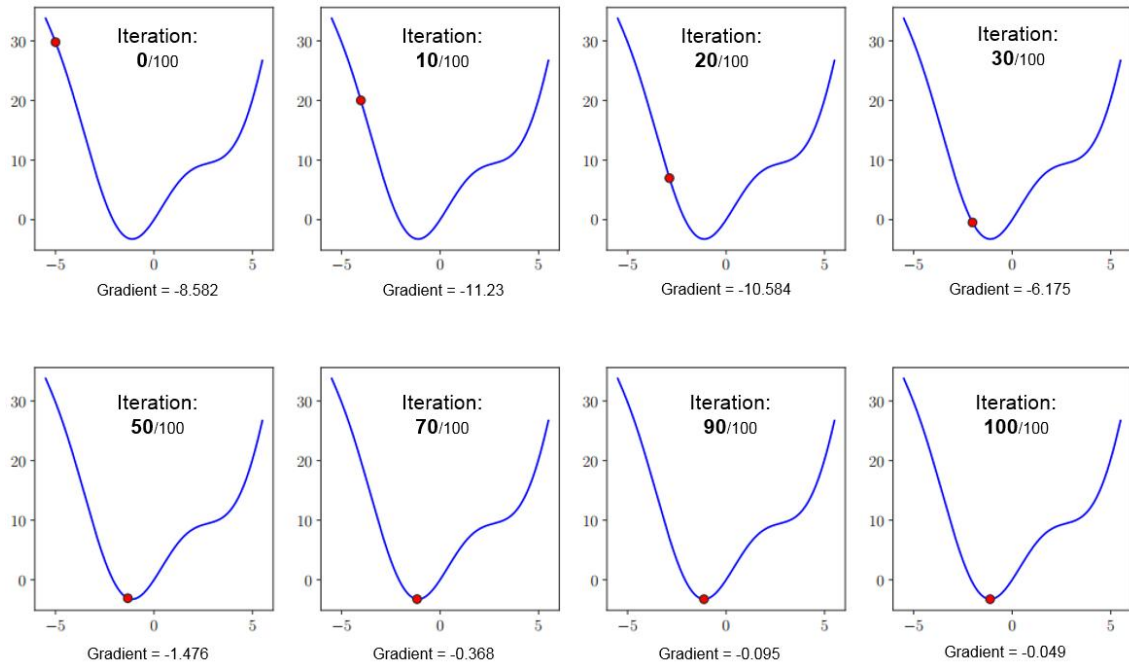


Figure 4.8: The convergence of a network model at learning rate $\eta = 0.01$

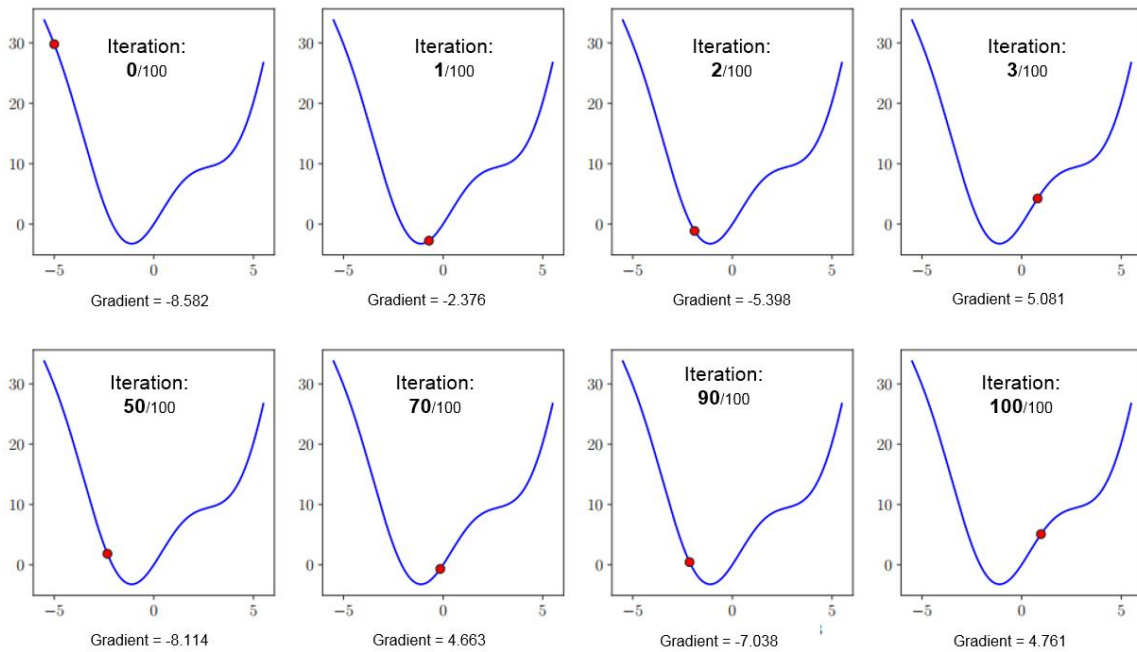


Figure 4.9: The convergence of a network model at learning rate $\eta = 0.5$

Therefore, it needs an algorithm that can adapt the learning rate themselves. [111], [112], [113]. The most popular algorithm is adaptive moment estimation [114]. It considers the exponentially decaying average of the past gradient, m_t , and the exponential decaying average of the past squared gradient, v_t :

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \quad 4.36$$

$$v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t \odot g_t \quad 4.37$$

Where: $g_t = \nabla_{\theta} J_t(\theta)$ is gradient on current mini-batch at timestep t that is calculated by backpropagation, described in **Algorithm 1**; and $g_t \odot g_t$ indicates the elementwise square; m_t and v_t denotes exponentially weighted averages of past gradient and the squared past gradient, respectively, initial $m_0 = 0$ and $v_0 = 0$; $\beta_1 = 0.9$ and $\beta_2 = 0.999$ are hyper-parameters controlling the exponential decay rates of these moving averages. The update rule when applying adaptive learning rate algorithm follows [114]:

$$\theta_{t+1} = \theta_t - \eta \frac{m_t}{\sqrt{v_t + \epsilon}} \quad 4.38$$

In which $\epsilon = 1e - 08$ is a small quantity preventing the division of zero.

- Considering Equation 4.36 with initial $m_0 = 0$, the subsequent update equations follows:

$$m_1 = (1 - \beta_1) * g_1$$

$$m_2 = \beta_1 * m_1 + (1 - \beta_1) * g_2 = \beta_1 * (1 - \beta_1) * g_1 + (1 - \beta_1) * g_2$$

$$m_3 = \beta_1 * m_2 + (1 - \beta_1) * g_3$$

$$= \beta_1 * [\beta_1 * (1 - \beta_1) * g_1 + (1 - \beta_1) * g_2] + (1 - \beta_1) * g_3$$

$$= \beta_1 * \beta_1 * (1 - \beta_1) * g_1 + \beta_1 (1 - \beta_1) * g_2 + (1 - \beta_1) * g_3$$

$$\Rightarrow m_t = (1 - \beta_1) \sum_{i=0}^{t-1} \beta_1^{t-i} * g_i$$

These above equations indicate that the previous gradients are also included in the current update. Therefore, it reveals that if the past gradients are in the same direction, the current gradient is reinforced. The reinforcement possibly helps network model that prevents a local minimum. The blue lines in Figure 4.10 and Figure 4.11 depict a function $f(x) = x^2 + 10 * \sin(x)$ of which derivation is: $f'(x) = 2x + 10 * \cos(x)$ and the moving of the red point is the movement of the solution based on the gradient descent method. The function $f(x)$ has a local minimum and global minimum. Without the exponential average factor, the red point converges early at the local minimum, which is not the expected solution, as shown in Figure

4.10. However, due to the exponential average factor, the red point passed the local minimum to converge at the global minima.

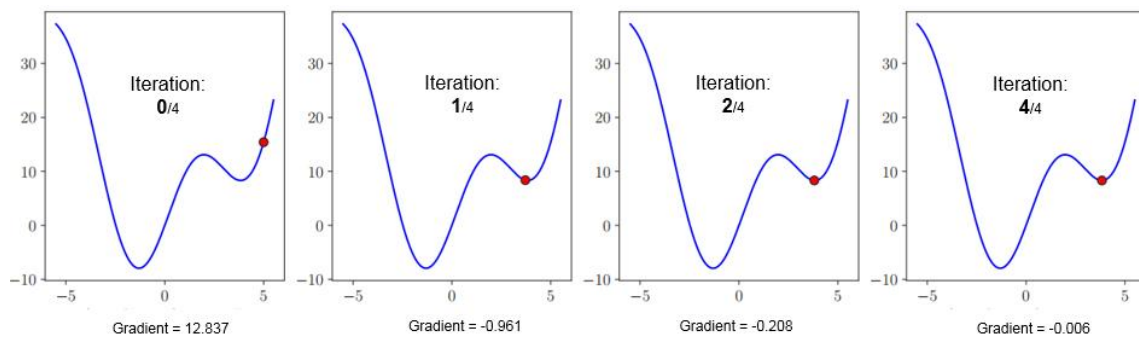


Figure 4.10: Gradient descent without the exponential average factor

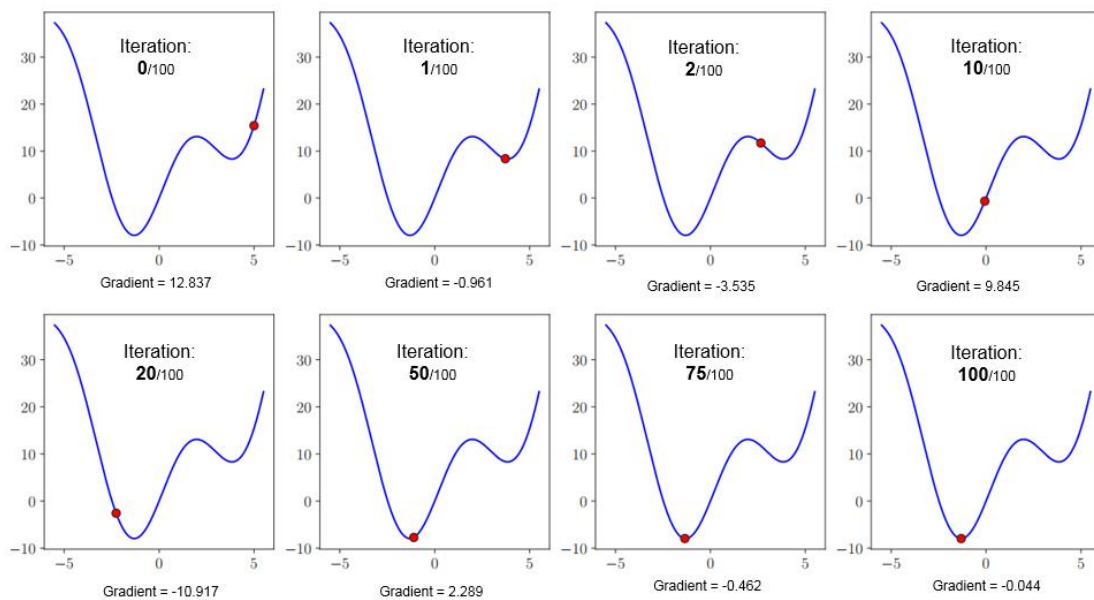


Figure 4.11: Gradient descent with the exponential average factor

If the past gradients are bounced off each other, that means the signal of the gradient is changed after every iteration, they will be cancelled out because of the sum. Finally, it reduces the oscillation and enhances the convergence of the network model. Another advantage is the effect of exponentially weighted average on the actual direction, which is demonstrated in Figure 4.12. Because derivatives of the cost function are noisy, it leads to possibly going in the wrong direction towards the minima. Therefore, the exponential average of the past gradients corrects the direction to the actual derivate, which is better than the noisy calculations.

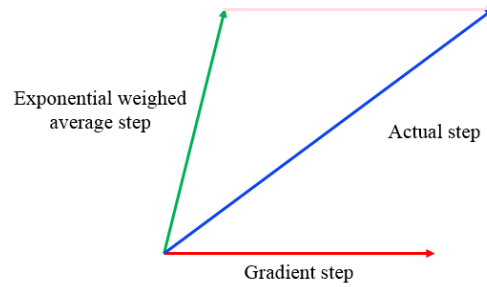


Figure 4.12: Correction direction actual step due to exponential weighed average factor

Equation 4.37 calculates an exponential average of the square of the gradients. By dividing it in Equation 4.38, the learning rate will be adaptively updated much larger for a small average and lesser for a large average of the past gradients.

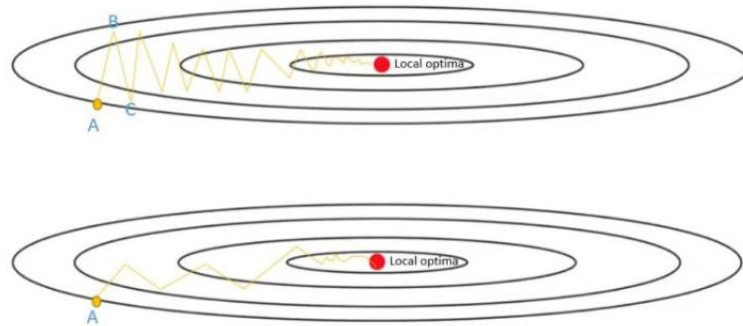


Figure 4.13: Gradient step without the exponentially weighted average (up) and with the exponentially weighted average (down) (Image source: DataCamp)

The benefit of using the exponential average is when the current point lies in ravines area where the surface of the cost function curves much more steeply in one dimension than in another [115]. Figure 4.13 illustrates contours of the quadratic cost function in which the red point is a minima point. Without the exponentially weighted average, the gradient will tend to oscillate across the ravines following point A to point B and then point C, as shown in Figure 4.13 (up) With each iteration of gradient descent, the update points move toward the optimal point in an upward and downward manner. The exponentially weighted average helps accelerate gradients in the right direction, as shown in Figure 4.13 (down). However, the first couple of iteration will provide a bad average because of the lack of values to average. Therefore, instead of using m_t and v_t , it uses the bias-corrected version of them: $\hat{m}_t = \frac{m_t}{1-\beta_1}$ and $\hat{v}_t = \frac{v_t}{1-\beta_2}$. Finally, the update rule using the adaptive learning rate algorithm follows:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad 4.39$$

By applying the adaptive momentum algorithm, learning rate for each neural network parameters will be adaptively updated individual, thus, increases training speed and convergence rate. The gradient descent with adaptive learning rate algorithm is described in **Algorithm 2**.

Algorithm 2: Gradient descent with adaptive momentum algorithm

Required:

- + Mini-batch
- + $m_{dw} = 0, v_{dw} = 0$, Initial exponential weighted average of weights
- + $m_{db} = 0, v_{db} = 0$, Initial exponential weighted average of biases
- + η : Initial learning rate
- + β_1, β_2

On iteration t :

+ Calculate exponential weighted average of gradient and squared gradient of weights and biases:

$$m_{dw} = \beta_1 * m_{dw} + (1 - \beta_1) * dw$$

$$m_{db} = \beta_1 * m_{db} + (1 - \beta_1) * db$$

$$v_{dw} = \beta_2 * v_{dw} + (1 - \beta_2) * dw^2$$

$$v_{db} = \beta_2 * v_{db} + (1 - \beta_2) * db^2$$

+ Calculate biased correction:

$$\hat{m}_{dw} = \frac{m_{dw}}{1 - \beta_1^t}$$

$$\hat{v}_{dw} = \frac{v_{dw}}{1 - \beta_2^t}$$

$$\hat{m}_{db} = \frac{m_{db}}{1 - \beta_1^t}$$

$$\hat{v}_{db} = \frac{v_{db}}{1 - \beta_2^t}$$

+ Update weights and biases:

$$w = w - \eta \frac{\hat{m}_{dw}}{\sqrt{\hat{v}_{dw} + \epsilon}}$$

$$b = b - \eta \frac{\hat{m}_{db}}{\sqrt{\hat{v}_{db} + \epsilon}}$$

4.7. Neural network regularization

The target of training a deep neural network is to ensure that the built model will generalize well for unseen data [116]. Many studies presented that the generalization ability of a neural network model is affected by a balance between the training examples and the size of the network [117] [118][119]. If the training data does not match the neural network complexity (such as when the network is very complex but with little training data) the network will present a bad generalization including underfitting and overfitting [120]. Due to overfitting, the network possibly produces excellent results as a small cost function for the training data but built a poor generalization capability. The model fits the noise in the data rather than the underlying function as shown in Figure 4.14c, where the blue line that presents for the trained model does not follow the green line that presents the correct model.

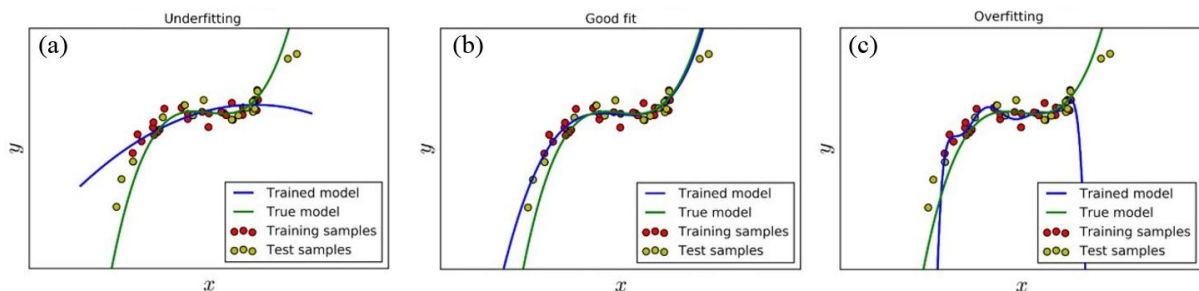


Figure 4.14: (a) Underfitting, (b) good fit, and (c) overfitting problematic (Image source: fundaml.com)

The overfitting often happens when a net model is too complicated. However, a thin network model possible leads to underfitting. It is a problem when a model could not learn enough from the training data and not capture the dominant trend, as shown in Figure 4.14a. The target is the development of a good model, as demonstrated in Figure 4.14b. There are several techniques used to regularize the network model based on preventing the capacity of a network model. In this thesis, the weight decay and dropout techniques are applied to prevent the overfitting.

4.7.1. Weight decay regularization

Weight decay regularization is a method to simplify the network model for achieving a better generalization by adding an additional term to the cost function that will penalize

complexity [116]. This method decreases the complexity of the model by limiting the growth of the weights, which is by adding a parameter norm penalty to the cost function:

$$\tilde{J}(w) = J(w) + \frac{1}{2}\lambda \sum_i w_i^2 = J(w) + \frac{1}{2}\lambda w^T w \quad 4.40$$

Where, $J(\theta)$ is the cost function, described in Equation 4.34 and $\tilde{J}(\theta)$ is the regularized cost function; w is a vector containing network parameters and w^T is its transpose; λ is called regularization parameter governing how strongly large weights are penalized. It means that λ denotes the relative contribution of the penalty term, $\sum_i w_i^2$, relative to the original cost function, $J(w)$. When $\lambda = 0$, it means no regularization, and large λ forces the weight to become smaller. The calculating gradient from Equation 4.40 follows:

$$\nabla_w \tilde{J}(w) = \nabla_w J(w) + \lambda w \quad 4.41$$

Following Equation 4.17, the weights are updated by:

$$w = w - \eta (\nabla_w \tilde{J}(w)) = w - \eta (\nabla_w J(w) + \lambda w) \quad 4.42$$

Written another way, the update is:

$$w = (1 - \eta\lambda)w - \eta \nabla_w J(w) \quad 4.43$$

Equation 4.43 reveals that the adding weigh decay modifies the learning rule by rescaling the weight vector on each step before performing the usual gradient update. Preventing the overfitting by applying the weight decay were presented [121][122]. The initial regularization parameter, λ , was selected at 0.0001.

4.7.2. Dropout

Dropout is another technique which addresses overfitting. It prevents overfitting and provides a way of approximately combining different neural network architectures [123][124][125]. Dropout means shutting down units randomly and temporarily in a neural network. It temporarily deactivates it from the network, as shown in Figure 4.15, which indicates a network model before (Figure 4.15a) and after (Figure 4.15b) dropout. The selection of ignored units is random. Each unit has remained with a fixed probability p ,

independent of another one. Form the original architecture, dropout results in a smaller network that contains all the remained units.

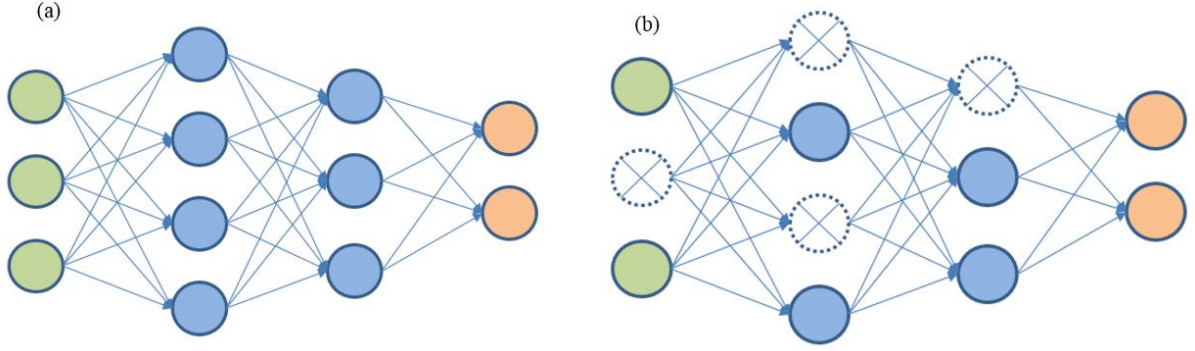


Figure 4.15: Neural network (a) before and (b) after applying dropout

Let $l = \{1, \dots, L\}$ index the hidden layers of a network that has L hidden layers; $\mathbf{z}_i^{(l)}$, $\mathbf{a}_i^{(l)}$ denote vector input and output of the l^{th} layer; $\mathbf{w}_i^{(l)}$, $\mathbf{b}^{(l)}$ denote weights and biases at the l^{th} layer. The feedforward of network with dropout follows:

$$\left. \begin{aligned} \mathbf{r}_j^{(l)} &\sim \text{Bernoulli}(p) \\ \tilde{\mathbf{a}}^{(l-1)} &= \mathbf{r}_j^{(l)} \odot \mathbf{a}^{(l-1)} \\ \mathbf{z}_i^{(l)} &= \mathbf{w}_i^{(l+1)} \tilde{\mathbf{a}}^{(l-1)} + \mathbf{b}_i^{(l)} \\ \mathbf{a}_i^{(l)} &= f(\mathbf{z}_i^{(l)}) \end{aligned} \right\} \quad 4.44$$

Where \odot : denote an element-wise product. For any layer l , $\mathbf{r}_j^{(l)}$ is a binary mask which has probability p of being 1. The $\mathbf{r}_j^{(l)}$ is multiplied element-wise with the output of that layer $\mathbf{a}^{(l)}$ to form a thinner output $\tilde{\mathbf{a}}^{(l)}$. The thin output $\tilde{\mathbf{a}}^{(l)}$ is applied as input to the next layer.

The training stage is used through the sub-network. After training, at the testing set, the network is used without dropout. When using dropout, the large neural network will be at a lesser risk of overfitting [120][126][127]. The initial value of hyperparameters p was recommended from 0.3 to 0.8. $p = 0.5$ was selected and will be tuned in this thesis. In addition, a combination of the dropout method and weight decay technique to improve the performance of the neural network was presented [128].

4.8. Deep neural network validation

During the training stage, the data splitting method has a significant impact on the performance of the final model. By partitioning the available data into other sets, the data will significantly reduce the number of samples which can be used for learning the model, and the results can depend on a particular random selection for the pair of sets. Therefore, in this thesis, the cross-validation technique was used [129]. Cross-validation is a statistical method of evaluating and comparing learning algorithms by dividing data into two subsets: one used to train a model and the other used to validate the model. The sets must cross-cover in successive rounds such that each data point has a chance of being validated against. The most common of this technique is k -fold cross-validation. Following this technique, the data is randomly partitioned into k equal (or nearly equal) size subsamples. Of the k subsets, a single sub-sample is retained as the validation data for testing the model, and the remaining $k - 1$ subsamples are used as training data. The training process is then repeated k times (the folds), with each of the k subsamples used exactly once as the validation data. The k results from the folds can then be averaged to produce a single estimation [130].

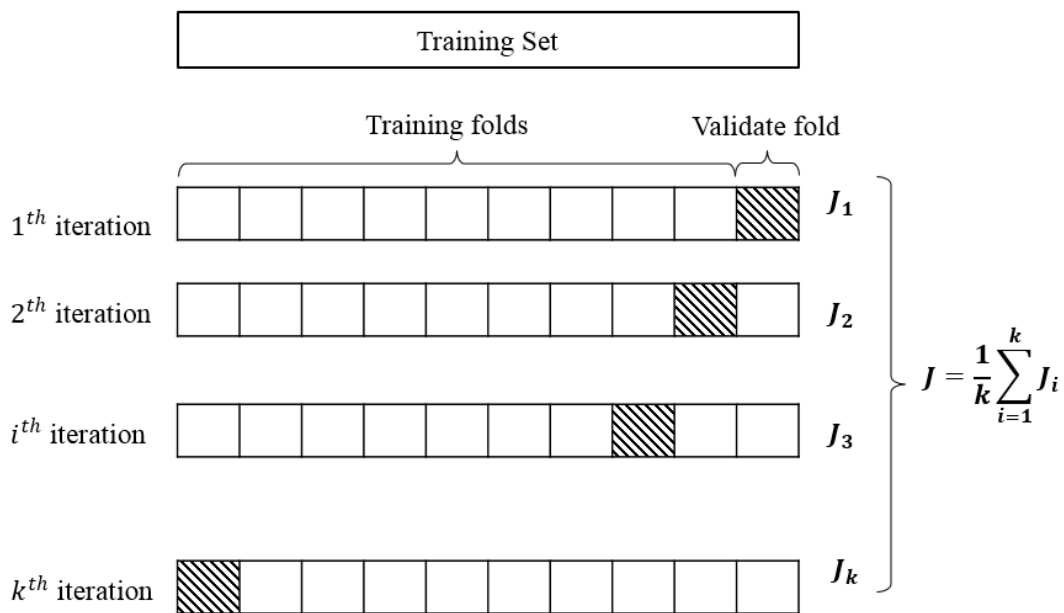


Figure 4.16: Process of k -fold cross-validation

Figure 4.16 illustrates the process of k -fold cross-validation, of which the white sections are used for training, and the diagonal stripe segments are used for validation. In each iteration, the training algorithm used $k - 1$ folds of data to train the network model, and subsequently,

the trained model is asked to predict the data in the validation fold [129]. The cost J is averaged from these iterations, $\frac{1}{k} \sum_{i=1}^k J_k$. The profit of this method is that all training data set are applied for both training and validation, and each observation is used for validation exactly once. Sub-data based on k -fold cross-validation should be independent and non-overlapping. If k value is too large, the training set is closer to the full data size but increasing the overlapping between iterations. This thesis uses $k = 10$ that has proved to be a good compromise [131] [129].

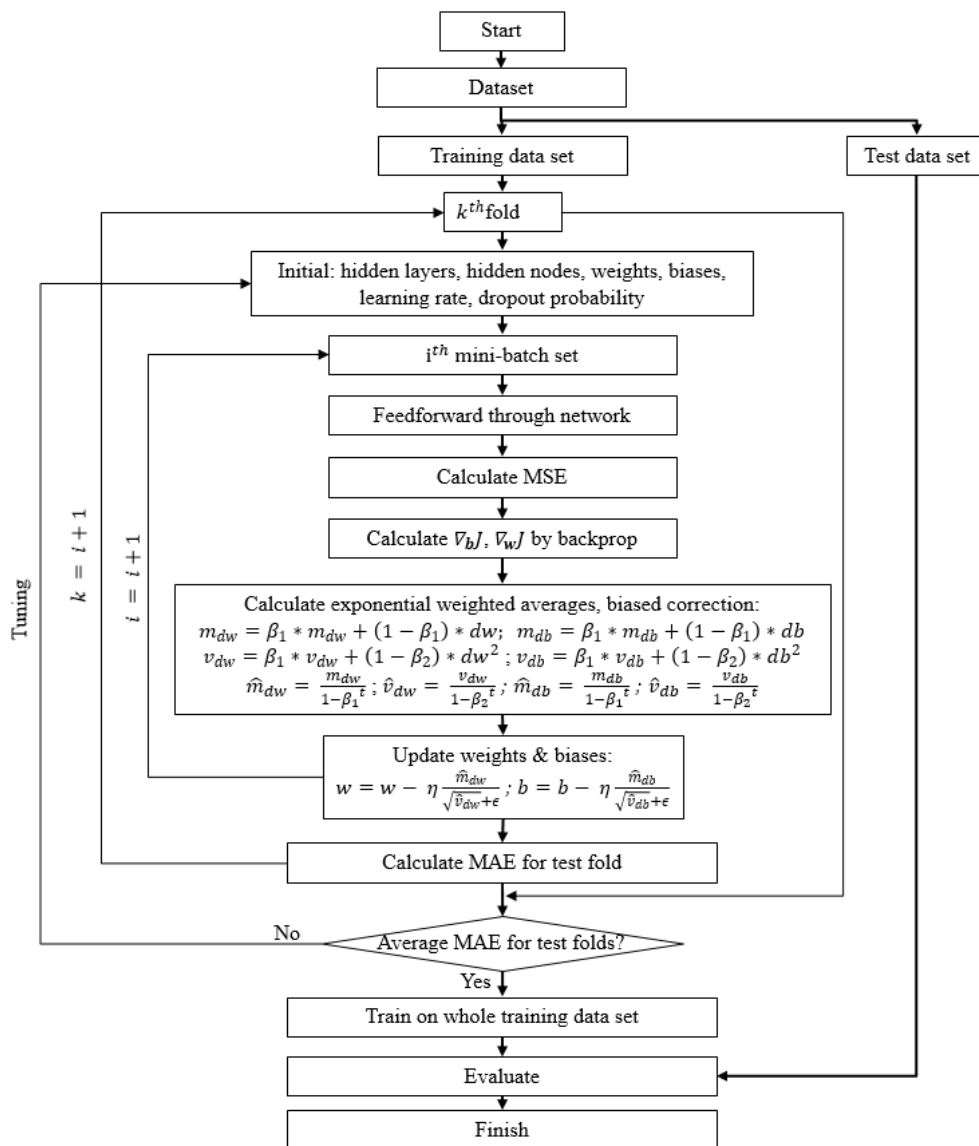


Figure 4.17: The flowchart of the training process of the deep neural network

Summarily, the total training algorithm of the deep neural network model is described by a flowchart in Figure 4.17. Firstly, the whole data collected by the experiment was divided into

data sets for training and testing. The training data set was split into the training folds and test fold. In each mini-batch partitioned from training folds, the input was fed to the network to calculate mean square error at the output layer. Each gradient of network parameter was computed by applying the backpropagation technique. Weights and biases were updated based on the adaptive learning rate algorithm. And the mean absolute error was computed. The process was repeated for all mini-batches and then all folds. After performing in all folds, the mean absolute error was averaged. The initial factors, such as neural network architecture, learning rate, dropout probability, were tuned to achieve a desirable average mean absolute error without overfitting. Next, the total training data set without k-fold splitting was finally used to train to gain the final neural network model. The final step is evaluating the network model by the test data set.

The neural network is implemented by using the TensorFlow of Google and Python programming language version 3.5. The MAE during the training process is shown in Table 4.1 and Figure 4.18. The figure reveals that the network model converges very fast at every fold. Moreover, values of train and validation process are close together, as indicated in Table 4.1. Average MAE of training and MAE of testing data are 2.4607 and 2.4127, respectively. It means that the developed model avoided the overfitting. Therefore, the developed neural network model is suitable for use.

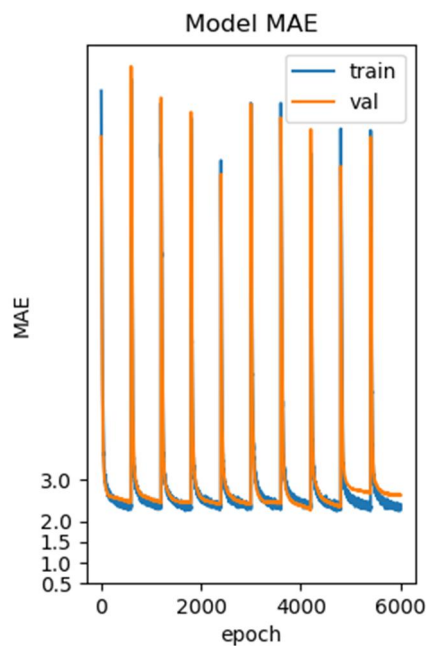


Figure 4.18: Virtual mean absolute error of 10-fold cross-validation

Table 4.1: Mean absolute error during training and testing of the neural network

Fold	Mean absolute error	
	Training	Validation
Fold #1	2.5037	2.4567
Fold #2	2.1269	2.4647
Fold #3	2.2973	2.4761
Fold #4	2.4010	2.4584
Fold #5	2.2642	2.4037
Fold #6	2.1313	2.3889
Fold #7	2.3995	2.2825
Fold #8	2.3395	2.4168
Fold #9	2.4497	2.5758
Fold #10	2.0782	2.6835
Average MAE	2.4706	
MAE of testing	2.4127	

Chapter 5. DESIGNING OF THE OPTIMIZATION SYSTEM

5.1. Generating data for prediction

The deep neural network, after being developed and verified, is applied to predict for new data. The data needing prediction were generated by a combination of the investigating SLM process parameters and their values. Laser power has 21 levels by changing from 80W to 180W with 5W variation. Laser scanning velocity factor has 18 levels, between 800mm/s to 2500mm/s with 100mm/s variation. Layer thickness has 13 levels by moving from 20 μ m to 80 μ m with 5 μ m variation. Hatch distance factor take 15 levels by changing from 30 μ m to 100 μ m with 5 μ m variation. Table 5.1 indicates process parameters that were used to predict by the developed deep neural network model.

Table 5.1: Generated data for predicting

Factor	Min value	Max Value	Interval	Level
Laser Power (W)	80	180	5	21
Laser scanning speed (mm/s)	800	2500	100	18
Layer thickness (μ m)	20	80	5	13
Hatch distance (μ m)	30	100	5	15

5.2. Optimization system algorithm

The optimization algorithm is shown in Figure 5.1. Firstly, the developed deep neural network is used to predict of generated data, including approximately 73 thousand input vectors, which is indicated in Table 5.1. The generated data includes inputs that represent the SLM process parameters, namely laser power, P , laser velocity, v , layer thickness, t , and hatch distance, h , without labeled output. The predicted values of the network model are stored as Predicted data as properties of the product printed by SLM, namely density ratio, d , and surface roughness, Sa . Once a user set desirable product qualities, the system will compare it with Predicted dataset. The point in Predicted data of which predicted values fit with the user input would be indexed and saved as Indexed data. If there is more than one point in Predicted data that are indexed, the Indexed data will be filtered. The filtered conditions include some thresholds to subtract only one point. The final point is the optimal process.

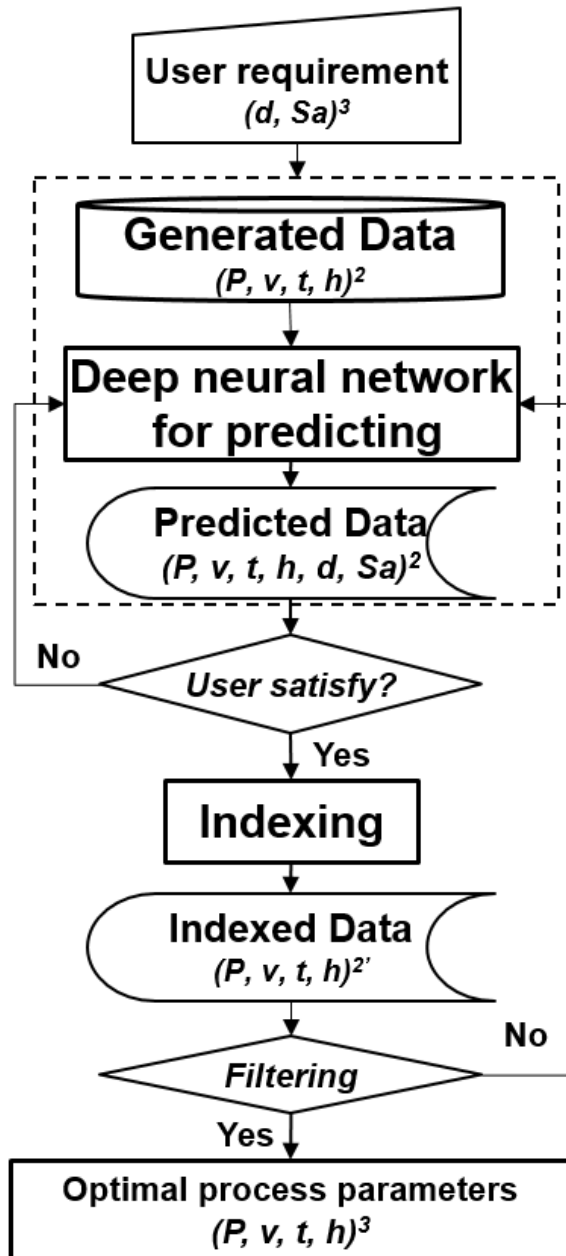


Figure 5.1: Algorithm of the Optimization System

5.2.1. Filtering condition

- Filter 1: Boundaries condition:

Although achieving a nearly full density of a product printed by SLM technology, component fabricated by this technique could not be higher than the bulk material. Moreover, a product with too high porosity is not acceptable in the SLM scope. Therefore, data of which predicted density exceeds 100 percentage or lower than 75 percentage must be filtered. In

considering surface roughness, the predicted data is limited between the minimum and maximum values of the original data used to train and test the neural network.

- Filter 2: Supplied energy condition:

Additionally, the experimental results were analysed to find the relationship between process parameters and printed product qualities. Previous studies presented that single melt lines have near semi-circular cross-section during the SLM process. Therefore, the SLM process can be simplified as the joining of overlapping areas. Figure 5.2 indicates an overlap area of two adjunct scan lines. h, t, r, τ are hatch distance, layer thickness, the radius of cross-section of single scanned line, and overlapping height, respectively. To achieve a full density part, $\tau \geq t$ [132]. Therefore, the radius value is calculated by:

$$r^2 = t^2 + \frac{h^2}{4} \quad 5.1$$

Therefore, the energy equation to account for the material volume is calculated as:

$$E = \frac{P}{v \cdot \frac{1}{2} \pi \cdot (t^2 + \frac{h^2}{4})} \quad 5.2$$

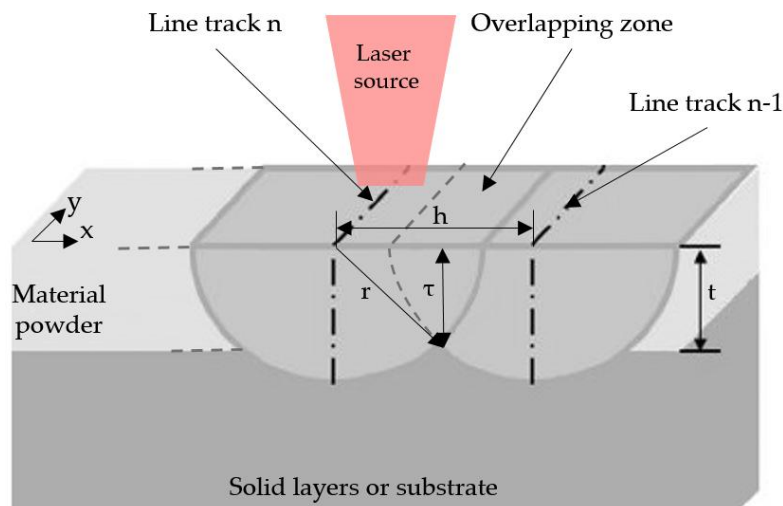


Figure 5.2: Schematic presenting the overlap zone during the SLM process.

Figure 5.3 (a) shows experiment results in which layer thickness and hatch distance are fixed at $20\mu m$ and $30\mu m$, respectively. The diagonal-stripe areas denote cracked samples when using corresponding parameters. Applying Equation 5.2 to calculate supplied energy illustrates that all the cracked samples happen when the supplied energy is higher than about

$152J.mm^{-3}$. These values are the smallest in the investigating range, resulting in the highest supplied laser energy, as mentioned in Table 5.1. Therefore, changing parameters generates a smaller than $152J.mm^{-3}$ of energy, which can avoid the cracking behavior. Moreover, the experiment at $80\mu m$ of layer thickness and $70\mu m$ of hatch distance reveals that printed were broke because of lacking fusion energy, as shown in the diagonal-stripes area Figure 5.3(b). Following the process described above, energy should be higher than $5J.mm^{-3}$ to print a successive product. Therefore, in consideration of supplied energy, the predicted set will be filtered data that makes laser energy in the range between $5J.mm^{-3}$ and $152J.mm^{-3}$.

- Filter 3: Productivity of the SLM process

The productivity of SLM is defined as $\dot{V} = v.h.t$. It is described by the material volume being created from material powder over time. It is a relationship between layer thickness, laser scanning speed and hatch space. If there are several data passing the Filter 1 and Filter 2, maximizing productivity is used as the terminal condition.

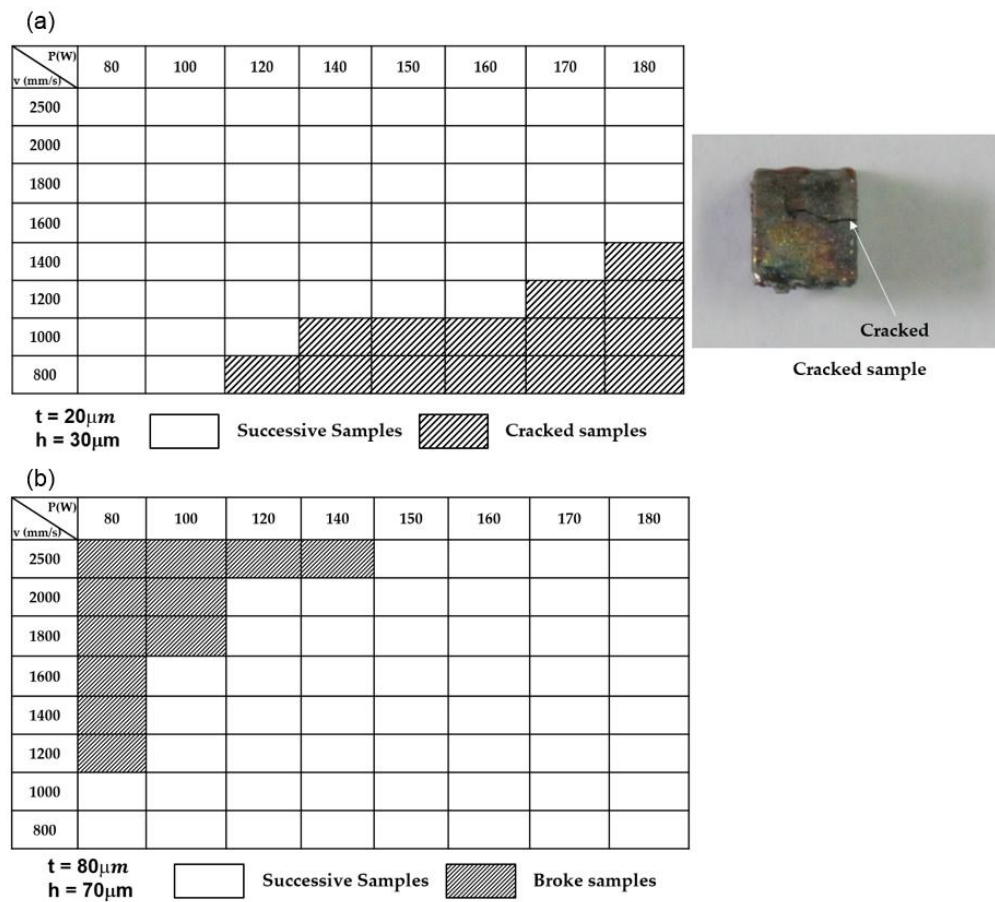


Figure 5.3: (a) Cracked and (b) broke samples during SLM processing

5.3. Case study

A bone (structure in Figure 5.4) is a porous composite material laid down by osteoblast cells and can be classified into the cortical bone and cancellous bone. The latter is a highly porous structure that normally is fabricated in a scaffold [133] [134] or gyroid structure [135] [136] by AM method. The former is the hard outer shell of bone with lower porosity. The density ratio of cortical bone is in 90% - 95% [137] [138] [139] [140].

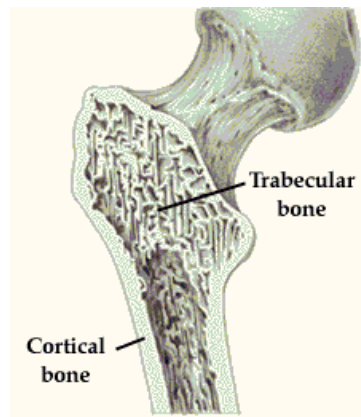


Figure 5.4: Bone structure

During an implant application, the rough surface exhibits better osseointegration and tissue integration performance than a smooth surface. The implant had high ability in both cell proliferation, and cell attachment, where the surface roughness, Sa was at $45\mu\text{m}$ and $53.25\mu\text{m}$ [141]. Therefore, 90% and 95% of the density ratio and $45\mu\text{m}$ and $53.25\mu\text{m}$ of surface roughness were selected to request the developed system. Density and surface roughness properties were then combined, resulting in four pairs of requests: Case 1 (90%, $45\mu\text{m}$); Case 2 (90%, $53.25\mu\text{m}$); Case 3 (95%, $45\mu\text{m}$) and Case 4 (95%, $53.25\mu\text{m}$). Table 5.2 indicates the optimal parameters corresponding to each requirement. These parameters were used to print samples.

The printed sample was taken to measure density and surface roughness. The density of the printed sample was measured by Archimedes' principle, while surface roughness was valued by the VK-200 confocal microscope scanning as described in Chapter 2. The measured results are indicated in Table 5.3; Case 1 had 90,85% of density ratio and $46\mu\text{m}$ of surface roughness; Case 2 had 90,59% of density ratio and $52\mu\text{m}$ of surface roughness; Case 3 had 95.755% of density and $52\mu\text{m}$ of surface roughness; and Case 4 had 95.285% of density

ratio and $52.5\mu\text{m}$ of surface roughness. The errors of density ratio and surface roughness between requirements and experiment are; 0.85% and $1\mu\text{m}$; 0.59% and $1.25\mu\text{m}$; 0.755% and $2\mu\text{m}$; 0.285% and $0.75\mu\text{m}$ for Case 1, Case 2, Case 3, and Case 4, respectively, as indicated in Table 5.3. The deviation between initial user requirements and the tested results prove that the results met the requested values and developed optimization system is valid for application.

Table 5.2: Optimal process parameters corresponding with the user requests

User request			Optimal process parameters			
Case study	Density ratio (%)	Surface roughness (μm)	Laser power, P, (W)	Laser scanning Speed, v, (mm/s)	Hatch distance, h, (μm)	Layer Thickness, t, (μm)
Case 1	90	45	180	2200	100	60
Case 2	90	53.25	180	1800	100	75
Case 3	95	45	180	1200	100	75
Case 4	95	53.25	180	1800	35	80

Table 5.3: Comparison of a user request and experiment result of which process parameters are generated from the optimization system.

Factor	User requirement	Experiment	Deviation
Case 1: P = 180W, v = 2200mm/s, h = 100 μm , t = 60 μm			
Density ratio (%)	90	90.85	0.85
Surface roughness(μm)	45	46	1
Case 2: P = 180W, v = 1800mm/s, h = 100 μm , t = 75 μm			
Density ratio (%)	90	90.59	0.59
Surface roughness(μm)	53.25	52	1.25
Case 3: P = 180W, v = 1200mm/s, h = 100 μm , t = 75 μm			
Density ratio (%)	95	95.755	0.755
Surface roughness(μm)	45	47	2
Case 4: P = 180W, v = 1800mm/s, h = 35 μm , t = 80 μm			
Density ratio (%)	95	95.285	0.285
Surface roughness(μm)	53.25	52.5	0.75

5.4. Implementation of the system.

In this thesis, the TensorFlow libraries combined with Python version 3.5 were used to implement and develop the neural network and the total optimization system. Python is an elegant and robust programming language that offers the power and general applicability of traditional compiled languages with the ease of use of simpler scripting and interpreted languages [142]. It combines remarkable expressive power with very clean, simple and compact syntax. Additionally, it is a functional language for performing mathematical calculations. TensorFlow is an open-source software machine learning library developed by Google. It uses tensor (multi-dimensional arrays) as its primary datatype and runs the learning algorithm as if data is flowing through the graph structure [143]. It was designed to allow efficient computation, especially in deep learning tasks [144] [145]. At first, the structure of neural is defined, then training data is feed into the structure to optimize the network model. Comparing predicted values with the labelled values is the final step to evaluate the developed network model. TensorFlow has become the tool of choice among professionals and researchers to implement machine learning solutions.

Additionally, *Tkinter*, an inbuilt Python module, was applied to build a graphical user interface (GUI) apps for more convenience during use. It provides a robust and platform-independent windowing toolkit by using *tkinter* package. Figure 5.5 demonstrates the graphical user interface of the optimization system. In this GUI, user requests are asked to input the two entries “Density ratio” and “Surface roughness”, correspondingly. The input must satisfy the Filter conditions. After that, the optimizing process is called by bottoming the “Optimizing”. The optimal process parameters will be displayed in four labels; “Laser power”, “Laser scanning speed”, “Hatch distance” and “Layer thickness”, respectively. The optimal parameters are exported and saved at *.txt format file by calling the “Save” function. To re-use the process, the “Reset” bottom needs to be called. These functions are also included in the Toolbar. In the end, the system is quitted by calling the “Exit” function.

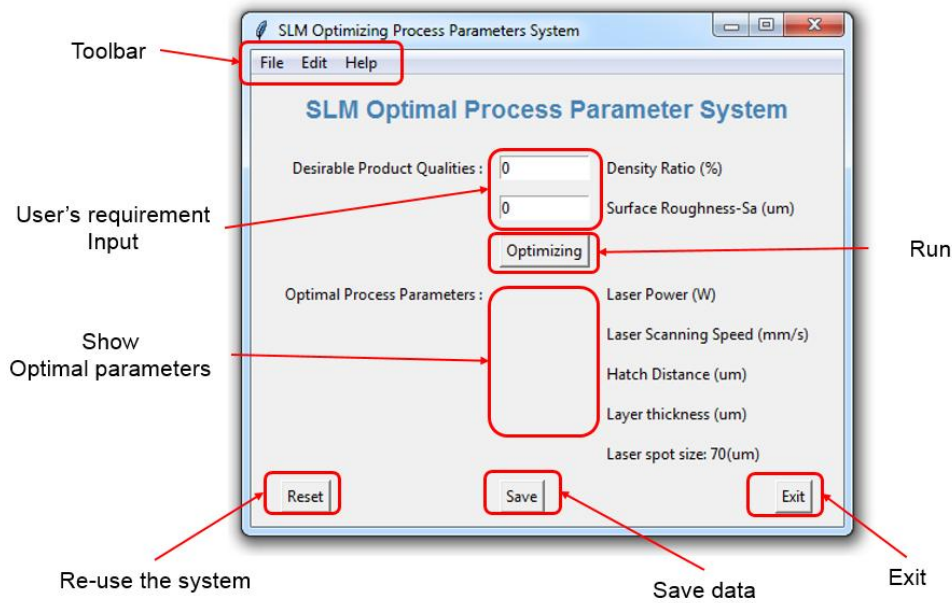


Figure 5.5: Graphical User Interface (GUI) of the developed optimization system

5.5. Aspect benefits

The optimization system operates based on deep learning. Deep neural networks have the ability to learn and model non-linear and complicated relationship. After teaching the network model by collected data, it can infer the unseen relationship on unseen data. In addition, the deep neural networks do not impose any restriction on the input variables. These advantages are very important because the relationship between SLM process parameters and its product are non-linear and complicated. Other statistical-based optimization widely used include, response surface methodology (RSM), Taguchi, design of experiment (DOE). However, any form of a non-linear relationship among the variables could not reduce the prediction accuracy. Therefore, optimization using deep learning algorithm is an advanced technique. Additionally, the metal Additive Manufacturing is a robust evolution in all industries, namely automotive, aerospace, medical, and so on. The expanding usage of metal AM requires much time to train for a new operator. Using the developed optimization system will suggest a suitable parameter at the beginning that reduces the training time. The system also reduces testing cost and time, getting a fail part, and expertise requirement that decreases manufacturing expenditure finally. Expanding research and applying the metal AM all over the world increasingly distributes researched data in society. Therefore, applying artificial intelligence in this thesis is paving the way for future research.

Chapter 6. CONCLUSION AND FUTURE WORK

This thesis is a study in optimizing Selective Laser Melting crucial process parameters for fabricating medical application parts based on deep learning. A system for optimizing process parameters of Selective Laser Melting Ti-6Al-4V parts manufacturing has been developed. The developed optimization system applied supervised learning deep neural network. The deep neural network architecture, as well as training algorithm, were built. These investigated parameters such as laser power, laser scanning speed, layer thickness and hatch distance were the input while printed product qualities namely density ratio and surface roughness collected from the experiment were the output of the deep neural network model. The gradient descent used to calculate the derivative of the mean square error function was combine with backpropagation applied to train the network model. In the training stage, the k-fold cross-validation method was applied. The adaptive momentum algorithm that required the exponentially weighted average was utilized. It individually updates the parameters of deep neural network separately. Therefore, the process of training network model was sped up, and prevent the noise of the cost function. For preventing the overfitting problematic resulting in a bed generalization, the network was regularized by weight decay and dropout methods. The weight decay method was used by adding an additional term that penalty the large weights. The dropout technique was performed by shutting down a neural node randomly to reduce the complexity of the network model. The errors of training, testing were 2.4607 and 2.4127, respectively. The results indicated an acceptable error of training. They were small and closed each other. Thus, the developed neural network model was not overfitting. Therefore, the network had a good generalization and a reliable model for prediction.

The deep network model after tuned and achieved an acceptable error was used to predict a new dataset. A selection system was built to index the optimal process parameters based on a user requirement as product properties. By analysing the experiment results, the relationship between process parameters and printed samples was considered as filtering conditions that were applied to the optimization system. The optimization system used desirable density, and surface roughness from a user as the inputs and generated process parameters as the output. The optimization system scans the predicted data, and then filters and indexed the optimal parameters that satisfy a user request. A graphical user interface was created for more

convenience while using the system. To verify the optimization system, the optimal parameters suggested by the system were used for the SLM process again. The error between the user requirement and the experiment confirmed the success of the developed system. The optimization system solved the difficulty of selecting an SLM suitable process parameters to manufacture the desired product qualities. It reduces the expenditure of SLM manufacturing by decreasing pre-processing time and cost.

For the future, training data will be collected further to improve performance of the system. More process parameters, as well as properties, will be investigated, which will be an improvement on the current four crucial process parameters. Additionally, other algorithms such as unsupervised learning, and semi-supervised learning and reinforcement learning will be considered instead of supervised learning.

Publications

□ Journal

1. **Dinh Son Nguyen**, Hong Seok Park & Chang Myung Lee (2019). **Effect of cleaning gas stream on products in selective laser melting**, *Materials and Manufacturing Processes*, 34:4, 455-461, DOI: [10.1080/10426914.2018.1512132](https://doi.org/10.1080/10426914.2018.1512132)

2. **Nguyen, D.-S.**; Park, H.-S.; Lee, C.-M. **Applying Selective Laser Melting to Join Al and Fe: An Investigation of Dissimilar Materials**. *Appl. Sci.* **2019**, *9*, 3031. DOI: [10.3390/app9153031](https://doi.org/10.3390/app9153031)

3. Ansari, M.J.; **Nguyen, D.-S.**; Park, H.S. **Investigation of SLM Process in Terms of Temperature Distribution and Melting Pool Size: Modeling and Experimental Approaches**. *Materials* **2019**, *12*, 1272. DOI:[10.3390/ma12081272](https://doi.org/10.3390/ma12081272)

4. Park, HS., Dang, XP., **Nguyen, DS.** et al. **Design of Advanced Injection Mold to Increase Cooling Efficiency** *Int. J. of Precis. Eng. and Manuf.-Green Tech.* (2019). DOI: [10.1007/s40684-019-00041-4](https://doi.org/10.1007/s40684-019-00041-4)

5. Park Hong-Seok, **Nguyen Dinh-Son**, **AI-based optimization of process parameters in Selective Laser Melting**. *IOS Press*. [10.3233/978-1-61499-902-7-119](https://doi.org/10.3233/978-1-61499-902-7-119)

6. H S Park, **Dinh Son Nguyen**, **Study on Flaking Behavior in Selective Laser Melting Process** *Procedia CIRP*, 2017, 63, 569-572

7. H S Park, N H Tran, **D S Nguyen**, **Development of a predictive system for SLM product quality**. 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* 227 012090

□ Program Registration:

Registration Number: 230171 – 0002982

Title: 선택적 레이저 용해 최적화 공정 매개 변수 시스템

□ Patents:

	Patent number	Application number	Title
1	10-1786458	10-2016 -0177959	3 차원 프린터용 소재 공급치 및 이를 구비하는 3차원 프린터
2	10-1895168	10-2017 -0067607	3 차원 프린팅용 및 이를 이용한 3 차원 프린팅 방법
3	10-1960594	10-2017 -0059271	잔류 응력에 따른 출력물의 변형을 방지하는 3차원 프린팅 장치 및 그것을 이용한 3차원 프린팅 방법
4	10-1962535	10-2017 -0067608	순환형 분말 적층 기반의 3차원 프린팅 장치 및 그 방법
5	10-1981859	10-2017 -0101040	3 차원 프린팅

References

- [1] ASTM INTERNATIONAL. ASTM F2792-12a. *West Conshohocken, PA* **2013**, 1–3.
- [2] Wohlers, A.; Wohlers Associates. Executive Summary of the Wohlers Report 2015. **2014**, 10.
- [3] Levy, G. N.; Schindel, R.; Kruth, J. P. Rapid Manufacturing and Rapid Tooling with Layer Manufacturing (LM) Technologies, State of the Art and Future Perspectives. *CIRP Ann. - Manuf. Technol.* **2003**.
- [4] Gibson, I.; Rosen, D. W.; Stucker, B. *Additive Manufacturing Technologies: Rapid Prototyping to Direct Digital Manufacturing*; 2010.
- [5] Lee, J. Y.; An, J.; Chua, C. K. Fundamentals and Applications of 3D Printing for Novel Materials. *Applied Materials Today*. 2017.
- [6] Yap, Y. L.; Wang, C.; Sing, S. L.; Dikshit, V.; Yeong, W. Y.; Wei, J. Material Jetting Additive Manufacturing: An Experimental Study Using Designed Metrological Benchmarks. *Precis. Eng.* **2017**.
- [7] Govil, K.; Kumar, V.; Pandey, D. P.; Praneeth, R.; Sharma, A. Additive Manufacturing and 3D Printing: A Perspective. In *Lecture Notes in Mechanical Engineering*; 2019.
- [8] Yang, L.; Hsu, K.; Baughman, B.; Godfrey, D.; Medina, F.; Menon, M.; Wiener, S. *Additive Manufacturing of Metals: The Technology, Materials, Design and Production*; 2017.
- [9] Kirkpatrick, K. 3D-Printing Human Body Parts. *Commun. ACM* **2017**, 60 (10), 15–17.
- [10] Tappa, K.; Jammalamadaka, U. Novel Biomaterials Used in Medical 3D Printing Techniques. *Journal of Functional Biomaterials*. 2018.
- [11] Lu, B.-H.; Lan, H.-B.; Liu, H.-Z. Additive Manufacturing Frontier: 3D Printing Electronics. *Opto-Electronic Adv.* **2018**.
- [12] ASTM International. *F2792-12a - Standard Terminology for Additive Manufacturing Technologies*; 2013.
- [13] Kruth, J.P., Dadbakhsh, S., Vrancken, B., et al. *Additive Manufacturing of Metal via Selective Laser Melting-Process Aspects and Material Development*, 1st Editio.;

- Srivatsan, T., Sudarshan, T., Eds.; CRC Press: Boca Raton, 2015.
- [14] Khaing, M. W.; Fuh, J. Y. H.; Lu, L. Direct Metal Laser Sintering for Rapid Tooling: Processing and Characterisation of EOS Parts. In *Journal of Materials Processing Technology*; 2001.
- [15] Kruth, J. P.; Wang, X.; Laoui, T.; Froyen, L. Lasers and Materials in Selective Laser Sintering. *Assem. Autom.* **2003**.
- [16] Mercelis, P.; Kruth, J. P. Residual Stresses in Selective Laser Sintering and Selective Laser Melting. *Rapid Prototyp. J.* **2006**.
- [17] Elahinia, M. H. *Shape Memory Alloy Actuators: Design, Fabrication and Experimental Evaluation*; 2015.
- [18] Andani, M. T.; Shayesteh Moghaddam, N.; Haberland, C.; Dean, D.; Miller, M. J.; Elahinia, M. Metals for Bone Implants. Part 1. Powder Metallurgy and Implant Rendering. *Acta Biomaterialia*. 2014.
- [19] Karunakaran, K. P.; Suryakumar, S.; Pushpa, V.; Akula, S. Low Cost Integration of Additive and Subtractive Processes for Hybrid Layered Manufacturing. *Robot. Comput. Integr. Manuf.* **2010**.
- [20] Li, L.; Haghighi, A.; Yang, Y. A Novel 6-Axis Hybrid Additive-Subtractive Manufacturing Process: Design and Case Studies. *J. Manuf. Process.* **2018**.
- [21] Sun, P.; Fang, Z. Z.; Zhang, Y.; Xia, Y. Review of the Methods for Production of Spherical Ti and Ti Alloy Powder. *JOM*. 2017.
- [22] Sutton, A. T.; Kriewall, C. S.; Leu, M. C.; Newkirk, J. W. Powders for Additive Manufacturing Processes: Characterization Techniques and Effects on Part Properties. *Solid Free. Fabr. Proc.* **2016**.
- [23] Sutton, A. T.; Kriewall, C. S.; Leu, M. C.; Newkirk, J. W. Powder Characterisation Techniques and Effects of Powder Characteristics on Part Properties in Powder-Bed Fusion Processes. *Virtual and Physical Prototyping*. 2017.
- [24] Gu, H.; Gong, H.; Dilip, J. J. S.; Pal, D.; Hicks, A.; Doak, H.; Stucker, B. Effects of Powder Variation on the Microstructure and Tensile Strength of Ti-6Al-4V Parts Fabricated by Selective Laser Melting. *Int. J. Powder Metall.* **2015**.

- [25] Chen, G.; Zhao, S. Y.; Tan, P.; Wang, J.; Xiang, C. S.; Tang, H. P. A Comparative Study of Ti-6Al-4V Powders for Additive Manufacturing by Gas Atomization, Plasma Rotating Electrode Process and Plasma Atomization. *Powder Technol.* **2018**.
- [26] Chen, G.; Zhao, S. yang; Tan, P.; Yin, J. ou; Zhou, Q.; Ge, Y.; Li, Z. feng; Wang, J.; Tang, H. ping; Cao, P. Shape Memory TiNi Powders Produced by Plasma Rotating Electrode Process for Additive Manufacturing. *Trans. Nonferrous Met. Soc. China (English Ed.* **2017**.
- [27] Vock, S.; Klöden, B.; Kirchner, A.; Weißgärber, T.; Kieback, B. Powders for Powder Bed Fusion: A Review. *Prog. Addit. Manuf.* **2019**.
- [28] Armillotta, A.; Baraggi, R.; Fasoli, S. SLM Tooling for Die Casting with Conformal Cooling Channels. *Int. J. Adv. Manuf. Technol.* **2014**.
- [29] Dimla, E. Design Considerations of Conformal Cooling Channels in Injection Moulding Tools Design: An Overview. *J. Therm. Eng.* **2015**.
- [30] Park, H.-S.; Dang, X.-P. Design and Simulation-Based Optimization of Cooling Channels for Plastic Injection Mold. In *New Technologies - Trends, Innovations and Research*; 2012.
- [31] Oshida, Y. *Bioscience and Bioengineering of Titanium Materials: Second Edition*; 2012.
- [32] Popov, V. V.; Muller-Kamskii, G.; Kovalevsky, A.; Dzhenzhera, G.; Strokin, E.; Kolomiets, A.; Ramon, J. Design and 3D-Printing of Titanium Bone Implants: Brief Review of Approach and Clinical Cases. *Biomedical Engineering Letters.* 2018.
- [33] Anatoliy, P.; Vadim, S.; Igor, P.; Evgenii, B.; Dmitriy, M. Producing Hip Implants of Titanium Alloys by Additive Manufacturing. In *Proceedings of the International Conference on Progress in Additive Manufacturing*; 2016.
- [34] Yang, L.; Chen, X.; Zhang, L.; Li, L.; Kang, S.; Wang, C.; Sun, W. Additive Manufacturing in Vascular Stent Fabrication. *MATEC Web Conf.* **2019**.
- [35] Zhang, J.; Song, B.; Wei, Q.; Bourell, D.; Shi, Y. A Review of Selective Laser Melting of Aluminum Alloys: Processing, Microstructure, Property and Developing Trends. *Journal of Materials Science and Technology.* 2019.
- [36] Qiu, C.; Chen, H.; Liu, Q.; Yue, S.; Wang, H. On the Solidification Behaviour and

- Cracking Origin of a Nickel-Based Superalloy during Selective Laser Melting. *Mater. Charact.* **2019**.
- [37] Santos, L. M. S.; Ferreira, J. A. M.; Borrego, L. P.; Costa, J. D.; Capela, C.; de Jesus, J. Fatigue Crack Propagation along Interfaces of Selective Laser Melting Steel Hybrid Parts. *Fatigue Fract. Eng. Mater. Struct.* **2019**.
- [38] Wally, Z. J.; Haque, A. M.; Feteira, A.; Claeysens, F.; Goodall, R.; Reilly, G. C. Selective Laser Melting Processed Ti6Al4V Lattices with Graded Porosities for Dental Applications. *J. Mech. Behav. Biomed. Mater.* **2019**.
- [39] ZIMMERMANN, H.-J. DESCRIPTION AND OPTIMIZATION OF FUZZY SYSTEMS†. *Int. J. Gen. Syst.* **1975**, 2 (1), 209–215.
- [40] Ghani, J. A.; Choudhury, I. A.; Hassan, H. H. Application of Taguchi Method in the Optimization of End Milling Parameters. *J. Mater. Process. Technol.* **2004**.
- [41] Kivak, T. Optimization of Surface Roughness and Flank Wear Using the Taguchi Method in Milling of Hadfield Steel with PVD and CVD Coated Inserts. *Meas. J. Int. Meas. Confed.* **2014**.
- [42] Siddiquee, A. N.; Khan, Z. A.; Goel, P.; Kumar, M.; Agarwal, G.; Khan, N. Z. Optimization of Deep Drilling Process Parameters of AISI 321 Steel Using Taguchi Method. *Procedia Mater. Sci.* **2014**.
- [43] Nalbant, M.; Gökkaya, H.; Sur, G. Application of Taguchi Method in the Optimization of Cutting Parameters for Surface Roughness in Turning. *Mater. Des.* **2007**.
- [44] Ranganath, M. S.; Vipin; Harshit. Optimization of Process Parameters in Turning Operation Using Response Surface Methodology : A Review. *Int. J. Emerg. Technol. Adv. Eng.* **2014**.
- [45] Sun, J.; Yang, Y.; Wang, D. Parametric Optimization of Selective Laser Melting for Forming Ti6Al4V Samples by Taguchi Method. *Opt. Laser Technol.* **2013**.
- [46] Taguchi, G. INTRODUCTION TO TAGUCHI METHODS. *Eng.* **1988**.
- [47] Carlotto, A.; Loggi, A.; Zito, D.; Sbornicchia, P.; Maggian, D.; Fockele, M.; Unterberg, P.; Molinari, A.; Cristofolini, I. Optimization of the Main Selective Laser Melting Technology Parameters in the Production of Gold Jewelry. *Int. J. Powder Metall.* **2015**.

- [48] Alrbaey, K.; Wimpenny, D.; Tosi, R.; Manning, W.; Moroz, A. On Optimization of Surface Roughness of Selective Laser Melted Stainless Steel Parts: A Statistical Study. *J. Mater. Eng. Perform.* **2014**.
- [49] Senthilkumaran, K.; Pandey, P. M.; Rao, P. V. M. Influence of Building Strategies on the Accuracy of Parts in Selective Laser Sintering. *Mater. Des.* **2009**.
- [50] Bose, N. K.; Liang, P. Neural Network Fundamentals with Graphs, Algorithms, and Applications. *McGraw-Hill Ser. Electr. Comput. Eng.* **1996**.
- [51] Vui, C. S.; Soon, G. K.; On, C. K.; Alfred, R.; Anthony, P. A Review of Stock Market Prediction with Artificial Neural Network (ANN). In *Proceedings - 2013 IEEE International Conference on Control System, Computing and Engineering, ICCSCE 2013*; 2013.
- [52] Naik, N.; Mohan, B. R. Optimal Feature Selection of Technical Indicator and Stock Prediction Using Machine Learning Technique. In *Communications in Computer and Information Science*; 2019.
- [53] de Sá, C. R.; Shekar, A. K.; Ferreira, H.; Soares, C. Building Robust Prediction Models for Defective Sensor Data Using Artificial Neural Networks. In *Advances in Intelligent Systems and Computing*; 2020.
- [54] Altun, A.; Onder Efe, M. Aircraft Control with Neural Networks; 2019.
- [55] Somashekhar, K. P.; Ramachandran, N.; Mathew, J. Optimization of Material Removal Rate in Micro-EDM Using Artificial Neural Network and Genetic Algorithms. *Mater. Manuf. Process.* **2010**.
- [56] Tsai, K. M.; Wang, P. J. Comparisons of Neural Network Models on Material Removal Rate in Electrical Discharge Machining. *J. Mater. Process. Technol.* **2001**.
- [57] Demirci, H. H.; Coulter, J. P. Neural Network Based Control of Molding Processes. *J. Mater. Process. Manuf. Sci.* **1994**.
- [58] Lilly, K. W.; Melligeri, A. S. Dynamic Simulation and Neural Network Compliance Control of an Intelligent Forging Center. *J. Intell. Robot. Syst. Theory Appl.* **1996**.
- [59] Mirapeix, J.; García-Allende, P. B.; Cobo, A.; Conde, O. M.; López-Higuera, J. M. Real-Time Arc-Welding Defect Detection and Classification with Principal Component Analysis and Artificial Neural Networks. *NDT E Int.* **2007**.

- [60] Çaydaş, U.; Hasçalik, A. A Study on Surface Roughness in Abrasive Waterjet Machining Process Using Artificial Neural Networks and Regression Analysis Method. *J. Mater. Process. Technol.* **2008**.
- [61] Özden, G.; Mata, F.; Öteyaka, M. Ö. Artificial Neural Network Modeling for Prediction of Cutting Forces in Turning Unreinforced and Reinforced Polyamide. *J. Thermoplast. Compos. Mater.* **2019**.
- [62] Marrey, M.; Malekipour, E.; El-Mounayri, H.; Faierson, E. J. A Framework for Optimizing Process Parameters in Powder Bed Fusion (PBF) Process Using Artificial Neural Network (ANN). *Procedia Manuf.* **2019**.
- [63] Jiang, J.; Hu, G.; Li, X.; Xu, X.; Zheng, P.; Stringer, J. Analysis and Prediction of Printable Bridge Length in Fused Deposition Modelling Based on Back Propagation Neural Network. *Virtual Phys. Prototyp.* **2019**.
- [64] Francis, J.; Bian, L. Deep Learning for Distortion Prediction in Laser-Based Additive Manufacturing Using Big Data. *Manuf. Lett.* **2019**.
- [65] Papazoglou, D. P. Additively Manufactured Lattices for Orthopedic Implants and Process Monitoring of Laser-Powder Bed Fusion Using Neural Networks. **2019**, No. May.
- [66] Okaro, I. A.; Jayasinghe, S.; Sutcliffe, C.; Black, K.; Paoletti, P.; Green, P. L. Automatic Fault Detection for Laser Powder-Bed Fusion Using Semi-Supervised Machine Learning. *Addit. Manuf.* **2019**.
- [67] Valle-Pérez, G.; Camargo, C. Q.; Louis, A. A. Deep Learning Generalizes Because the Parameter-Function Map Is Biased towards Simple Functions. **2018**, No. 2015, 1–35.
- [68] Nguyen, D. S.; Park, H. S.; Lee, C. M. Effect of Cleaning Gas Stream on Products in Selective Laser Melting. *Mater. Manuf. Process.* **2019**.
- [69] Dadbakhsh, S.; Hao, L.; Sewell, N. Effect of Selective Laser Melting Layout on the Quality of Stainless Steel Parts. *Rapid Prototyp. J.* **2012**.
- [70] Wauthle, R.; Vrancken, B.; Beynaerts, B.; Jorissen, K.; Schrooten, J.; Kruth, J. P.; Van Humbeeck, J. Effects of Build Orientation and Heat Treatment on the Microstructure and Mechanical Properties of Selective Laser Melted Ti6Al4V Lattice Structures. *Addit. Manuf.* **2015**.

- [71] Dimter, M.; Mayer, R.; Hummeler, L.; Salzberger, R.; Kotila, J. Method and Device for Manufacturing a Three-Dimensional Object, 2011.
- [72] Edwards, P.; Ramulu, M. Fatigue Performance Evaluation of Selective Laser Melted Ti-6Al-4V. *Mater. Sci. Eng. A* **2014**.
- [73] Alonso, J. M.; Alvarruiz, F.; Desantes, J. M.; Hernández, L.; Hernández, V.; Moltó, G. Combining Neural Networks and Genetic Algorithms to Predict and Reduce Diesel Engine Emissions. *IEEE Trans. Evol. Comput.* **2007**, *11* (1), 46–55.
- [74] Keyence. 3D Laser Scanning Microscope VK-X100K/X105/X110 VK-X200K/X210 User's Manual.
- [75] Khademzadeh, S.; Carmignato, S.; Parvin, N.; Zanini, F.; Bariani, P. F. Micro Porosity Analysis in Additive Manufactured NiTi Parts Using Micro Computed Tomography and Electron Microscopy. *Mater. Des.* **2016**.
- [76] Shrestha, S.; Starr, T.; Chou, K. Porosity Analysis in Metal Additive Manufacturing by Micro-CT. In *ASME International Mechanical Engineering Congress and Exposition, Proceedings (IMECE)*; 2018.
- [77] Carmignato, S.; Dreossi, D.; Mancini, L.; Marinello, F.; Tromba, G.; Savio, E. Testing of X-Ray Microtomography Systems Using a Traceable Geometrical Standard. *Meas. Sci. Technol.* **2009**.
- [78] Wits, W. W.; Carmignato, S.; Zanini, F.; Vaneker, T. H. J. Porosity Testing Methods for the Quality Assessment of Selective Laser Melted Parts. *CIRP Ann. - Manuf. Technol.* **2016**.
- [79] Chang, C. S. Measuring Density and Porosity of Grain Kernels Using a Gas Pycnometer. *Cereal Chemistry* **1987**, *65* (1), 13–15.
- [80] Handbook of Sol-Gel Science and Technology. In *Handbook of Sol-Gel Science and Technology*; Lisa Klein, Mario Aparicio, A. J., Ed.; Springer, Cham, 2018; pp 1399–1409.
- [81] Spierings, A. B.; Schneider, M.; Eggenberger, R. Comparison of Density Measurement Techniques for Additive Manufactured Metallic Parts. *Rapid Prototyp. J.* **2011**.
- [82] Hughes, S. W. Archimedes Revisited: A Faster, Better, Cheaper Method of Accurately Measuring the Volume of Small Objects. *Phys. Educ.* **2005**.

- [83] Huang, L.; Hua, X.; Wu, D.; Ye, Y. Role of Welding Speed on Keyhole-Induced Porosity Formation Based on Experimental and Numerical Study in Fiber Laser Welding of Al Alloy. *Int. J. Adv. Manuf. Technol.* **2019**.
- [84] Thanki, A.; Dewulf, W.; Witvrouw, A.; Yang, S. Study of Keyhole-Porosities in Selective Laser Melting Using X-Ray Computed Tomography. *iCT19* **2019**, No. January, 1–7.
- [85] Katayama, S.; Seto, N.; Mizutani, M.; Matsunawa, A. Formation Mechanism of Porosity in High Power YAG Laser Welding. *Int. Congr. Appl. Lasers Electro-Optics* **2000**, *2000* (1), C16–C25.
- [86] Chen, J.; Wei, Y.; Zhan, X.; Li, Y.; Ou, W.; Zhang, T. Melt Flow and Thermal Transfer during Magnetically Supported Laser Beam Welding of Thick Aluminum Alloy Plates. *J. Mater. Process. Technol.* **2018**.
- [87] Kawaguchi, I.; Tsukamoto, S.; Arakane, G.; Nakata, K. Formation Mechanism of Porosity in Deep Penetration Laser Welding - Study on Prevention of Porosity in High Power CO₂laser Welding (Report 2). *Yosetsu Gakkai Ronbunshu/Quarterly J. Japan Weld. Soc.* **2006**.
- [88] Zhao, C.; Fezzaa, K.; Cunningham, R. W.; Wen, H.; De Carlo, F.; Chen, L.; Rollett, A. D.; Sun, T. Real-Time Monitoring of Laser Powder Bed Fusion Process Using High-Speed X-Ray Imaging and Diffraction. *Sci. Rep.* **2017**, *7* (1).
- [89] Nakamura, H.; Kawahito, Y.; Nishimoto, K.; Katayama, S. Elucidation of Melt Flows and Spatter Formation Mechanisms during High Power Laser Welding of Pure Titanium. *J. Laser Appl.* **2015**, *27* (3), 032012-1-032012–10.
- [90] Matthews, M. J.; Guss, G.; Khairallah, S. A.; Rubenchik, A. M.; Depond, P. J.; King, W. E. Denudation of Metal Powder Layers in Laser Powder Bed Fusion Processes. *Acta Mater.* **2016**, *114*, 33–42.
- [91] Ladewig, A.; Schlick, G.; Fisser, M.; Schulze, V.; Glatzel, U. Influence of the Shielding Gas Flow on the Removal of Process By-Products in the Selective Laser Melting Process. *Addit. Manuf.* **2016**.
- [92] Khan, M.; Dickens, P. Selective Laser Melting (SLM) of Gold (Au). *Rapid Prototyp. J.* **2012**, *18* (1), 81–94.

- [93] Sames, W. J.; List, F. A.; Pannala, S.; Dehoff, R. R.; Babu, S. S. The Metallurgy and Processing Science of Metal Additive Manufacturing. *International Materials Reviews*. 2016.
- [94] Kruth, J. P.; Froyen, L.; Van Vaerenbergh, J.; Mercelis, P.; Rombouts, M.; Lauwers, B. Selective Laser Melting of Iron-Based Powder. In *Journal of Materials Processing Technology*; 2004.
- [95] Gu, D.; Shen, Y. Balling Phenomena during Direct Laser Sintering of Multi-Component Cu-Based Metal Powder. *J. Alloys Compd.* **2007**.
- [96] Tolochko, N. K.; Mozzharov, S. E.; Yadroitsev, I. A.; Laoui, T.; Froyen, L.; Titov, V. I.; Ignatiev, M. B. Balling Processes during Selective Laser Treatment of Powders. *Rapid Prototyp. J.* **2004**.
- [97] Mumtaz, K. A.; Hopkinson, N. Selective Laser Melting of Thin Wall Parts Using Pulse Shaping. *J. Mater. Process. Technol.* **2010**.
- [98] Rosenblatt, F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychol. Rev.* **1958**.
- [99] Yun, K.; Huyen, A.; Lu, T. Deep Neural Networks for Pattern Recognition. In *Advances in Pattern Recognition Research*; 2018.
- [100] Widrow, B.; Lehr, M. A. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *Proc. IEEE* **1990**.
- [101] Nair, V.; Hinton, G. E. Rectified Linear Units Improve Restricted Boltzmann Machines. In *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*; 2010.
- [102] Jin, X.; Xu, C.; Feng, J.; Wei, Y.; Xiong, J.; Yan, S. Deep Learning with S-Shaped Rectified Linear Activation Units. In *30th AAAI Conference on Artificial Intelligence, AAAI 2016*; 2016.
- [103] Ian Goodfellow, Yoshua Bengio, A. C. *Deep Learning*; 2015.
- [104] Narayan, S. The Generalized Sigmoid Activation Function: Competitive Supervised Learning. *Inf. Sci. (Ny)*. **1997**.
- [105] Rumelhart, D. E.; Durbin, R.; Golden, R.; Chauvin, Y. Backpropagation: The Basic

- Theory. In *Backpropagation: Theory, architectures, and applications.*; Developments in connectionist theory.; Lawrence Erlbaum Associates, Inc: Hillsdale, NJ, US, 1995; pp 1–34.
- [106] Cybenko, G. Approximation by Superpositions of a Sigmoidal Function. *Math. Control. Signals Syst.* **1989**, 2 (4), 303–314.
- [107] Funahashi, K. I. On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks* **1989**.
- [108] Xiang, C.; Ding, S. Q.; Lee, T. H. Geometrical Interpretation and Architecture Selection of MLP. *IEEE Trans. Neural Networks* **2005**.
- [109] Fausett, L. *Fundamentals Of Neural Network Architectures, Algorithms, and Applications*; 2017.
- [110] Glorot, X.; Bengio, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Journal of Machine Learning Research*; 2010.
- [111] Jacobs, R. A. Increased Rates of Convergence through Learning Rate Adaptation. *Neural Networks* **1988**.
- [112] Duchi, J.; Hazan, E.; Singer, Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In *COLT 2010 - The 23rd Conference on Learning Theory*; 2010.
- [113] Zeiler, M. D. ADADELTA: An Adaptive Learning Rate Method. **2012**.
- [114] Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. *Int. Conf. Learn. Represent. 2015* **2017**.
- [115] Sutton, R. S. Two Problems with Backpropagation and Other Steepest-Descent Learning Procedures for Networks. *Proc. 8th annual conf. cognitive science society.* 1986.
- [116] Wolpert, D. H.; Nowlan, S. J. Simplifying Neural Networks by Soft Weight Sharing. In *The Mathematics of Generalization*; 2018.
- [117] Schwartz, D. B.; Samalam, V. K.; Solla, S. A.; Denker, J. S. Exhaustive Learning. *Neural Comput.* **1990**, 2 (3), 374–385.
- [118] Tishby, N.; Levin, E.; Solla, S. A. Consistent Inference of Probabilities in Layered

- Networks: Predictions and Generalization. In *IJCNN Int Jt Conf Neural Network*; 1989.
- [119] Baum, E. B.; Haussler, D. What Size Net Gives Valid Generalization? *Neural Comput.* **1989**.
- [120] Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. R. Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors. **2012**, 1–18.
- [121] Gnecco, G.; Sanguineti, M. The Weight-Decay Technique in Learning from Data: An Optimization Point of View. *Comput. Manag. Sci.* **2009**.
- [122] MacKay, D. J. C. A Practical Bayesian Framework for Backpropagation Networks. *Neural Comput.* **1992**.
- [123] Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems*; 2012; pp 1097–1105.
- [124] Dahl, G. E.; Sainath, T. N.; Hinton, G. E. Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*; 2013.
- [125] Gao, W.; Zhou, Z. H. Dropout Rademacher Complexity of Deep Neural Networks. *Sci. China Inf. Sci.* **2016**.
- [126] Kingma, D. P.; Salimans, T.; Welling, M. Variational Dropout and the Local Reparameterization Trick. In *Advances in Neural Information Processing Systems*; 2015.
- [127] Wager, S.; Wang, S.; Liang, P. Dropout Training as Adaptive Regularization. In *Advances in Neural Information Processing Systems*; 2013.
- [128] Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**.
- [129] Refaeilzadeh, P.; Tang, L.; Liu, H. Cross-Validation. In *Encyclopedia of Database Systems*; LIU, L., ÖZSU, M. T., Eds.; Springer US: Boston, MA, 2009; pp 532–538.
- [130] James, G.; Witten, D.; Hastie, T.; Tibshirani, R. *An Introduction to Statistical*

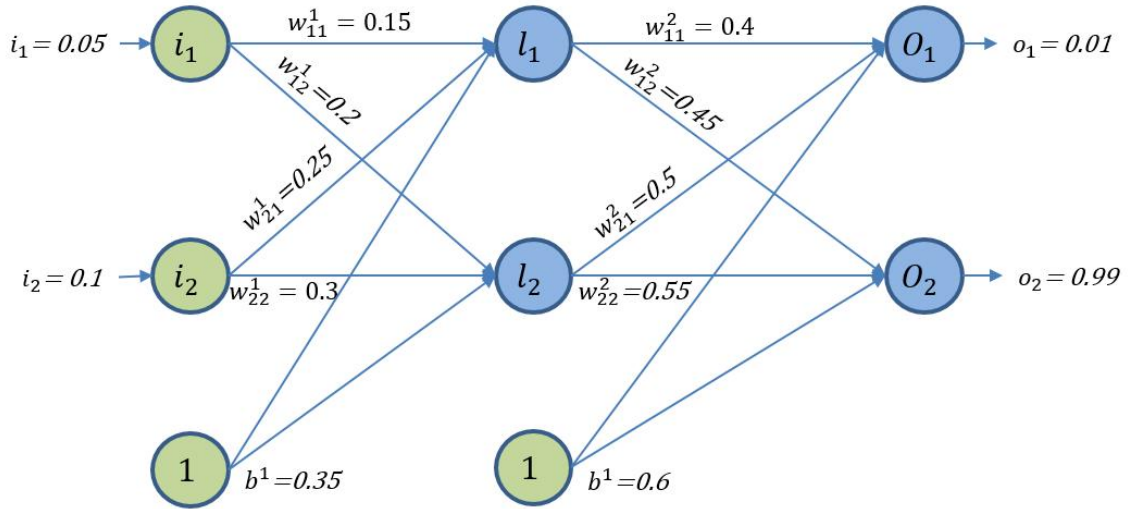
Learning; 2000.

- [131] Schenker, B.; Agarwal, M. Cross-Validated Structure Selection for Neural Networks. *Comput. Chem. Eng.* **1996**.
- [132] Yap, C. Y.; Chua, C. K.; Dong, Z. L. An Effective Analytical Model of Selective Laser Melting. *Virtual Phys. Prototyp.* **2016**.
- [133] O'Brien, F. J. Biomaterials & Scaffolds for Tissue Engineering. *Materials Today*. 2011.
- [134] Wieding, J.; Jonitz, A.; Bader, R. The Effect of Structural Design on Mechanical Properties and Cellular Response of Additive Manufactured Titanium Scaffolds. *Materials (Basel)*. **2012**.
- [135] Yáñez, A.; Cuadrado, A.; Martel, O.; Afonso, H.; Monopoli, D. Gyroid Porous Titanium Structures: A Versatile Solution to Be Used as Scaffolds in Bone Defect Reconstruction. *Mater. Des.* **2018**.
- [136] Wieding, J.; Lindner, T.; Bergschmidt, P.; Bader, R. Biomechanical Stability of Novel Mechanically Adapted Open-Porous Titanium Scaffolds in Metatarsal Bone Defects of Sheep. *Biomaterials* **2015**.
- [137] Atae, A.; Li, Y.; Fraser, D.; Song, G.; Wen, C. Anisotropic Ti-6Al-4V Gyroid Scaffolds Manufactured by Electron Beam Melting (EBM) for Bone Implant Applications. *Mater. Des.* **2018**.
- [138] Atae, A.; Li, Y.; Brandt, M.; Wen, C. Ultrahigh-Strength Titanium Gyroid Scaffolds Manufactured by Selective Laser Melting (SLM) for Bone Implant Applications. *Acta Mater.* **2018**.
- [139] Yan, C.; Hao, L.; Hussein, A.; Young, P. Ti-6Al-4V Triply Periodic Minimal Surface Structures for Bone Implants Fabricated via Selective Laser Melting. *J. Mech. Behav. Biomed. Mater.* **2015**.
- [140] Rajagopalan, S.; Robb, R. A. Schwarz Meets Schwann: Design and Fabrication of Biomimetic and Durataxic Tissue Engineering Scaffolds. *Med. Image Anal.* **2006**.
- [141] Zareidoost, A.; Yousefpour, M.; Ghasemi, B.; Amanzadeh, A. The Relationship of Surface Roughness and Cell Response of Chemical Surface Modification of Titanium. *J. Mater. Sci. Mater. Med.* **2012**.

- [142] Chun, W. *Core Python Programming*; 2006.
- [143] Nelli, F.; Nelli, F. Deep Learning with TensorFlow. In *Python Data Analytics*; 2018.
- [144] Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A System for Large-Scale Machine Learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*; 2016.
- [145] Abadi, M. TensorFlow: Learning Functions at Scale. *SIGPLAN Not.* **2016**, *51* (9), 1.

Appendix 1: An example of backpropagation calculation

Initial weights, the biases and the training input/outputs are described in Figure below; sigmoid activation function



Feedforward calculation

For the hidden layer:

Input of node l_1 : $z_{l_1} = w_{11}^1 * i_1 + w_{21}^1 * i_2 + b^1 * 1 = 0.3775$

Output of node l_1 : $a_{l_1} = \frac{1}{1 + e^{-z_{l_1}}} = \frac{1}{1 + e^{-0.3775}} = 0.5932$

Carry out the same process for node l_2 : $a_{l_2} = 0.5968$

Calculate the same process for output o_1 and o_2 :

$$z_{o_1} = w_{11}^2 * a_{l_1} + w_{21}^2 * a_{l_2} + b^1 * 1 = 1.1059$$

$$a_{o_1} = \frac{1}{1 + e^{-z_{o_1}}} = \frac{1}{1 + e^{-1.1059}} = 0.7513$$

Similarity: $a_{o_2} = 0.7729$

Total error: $J = J_{o_1} + J_{o_2}$

$$J_{o_1} = \frac{1}{2} (\text{target } o_1 - a_{o_1})^2 = \frac{1}{2} (0.01 - 0.7513)^2 = 0.2748$$

Similar: $J_{o_2} = 0.0235$

Therefore: $J = 0.2748 + 0.0235 = 0.2983$

Backward calculation:

The goal of backpropagation is to update each of the weights in the network that results in the actual output is closer to the target output. Considering how much a change in the weights w_{11}^2 affects the total error, $\frac{\partial J}{\partial w_{11}^2}$ must be calculated. $\frac{\partial J}{\partial w_{11}^2}$ is the partial derivative of J with respect to w_{11}^2 , or the gradient with respect to w_{11}^2 .

By applying the chain rule, we have:

$$\frac{\partial J}{\partial w_{11}^2} = \frac{\partial J}{\partial a_{o_1}} * \frac{\partial a_{o_1}}{\partial z_{o_1}} * \frac{\partial z_{o_1}}{\partial w_{11}^2}$$

$$\text{➤ } J = J_{o_1} + J_{o_2} = \frac{1}{2}(\text{target } o_1 - a_{o_1})^2 + \frac{1}{2}(\text{target } o_2 - a_{o_2})^2$$

$$\Rightarrow \frac{\partial J}{\partial a_{o_1}} = 2 * \frac{1}{2}(\text{target } o_1 - a_{o_1})^{2-1} * -1 + 0 = -(0.01 - 0.7513) = 0.7413$$

$$\text{➤ } a_{o_1} = \frac{1}{1+e^{-z_{o_1}}}$$

$$\Rightarrow \frac{\partial a_{o_1}}{\partial z_{o_1}} = a_{o_1}(1 - a_{o_1}) = 0.7513(1 - 0.7513) = 0.1868$$

$$\text{➤ } z_{o_1} = w_{11}^2 * a_{l_1} + w_{21}^2 * a_{l_2} + b^1 * 1$$

$$\Rightarrow \frac{\partial z_{o_1}}{\partial w_{11}^2} = a_{l_1} = 0.5932$$

Therefore:

$$\frac{\partial J}{\partial w_{11}^2} = 0.7413 * 0.1868 * 0.5932 = 0.0821$$

To decrease the error, gradient descent update rule is used (with learning rate is set to 0.5, for instance):

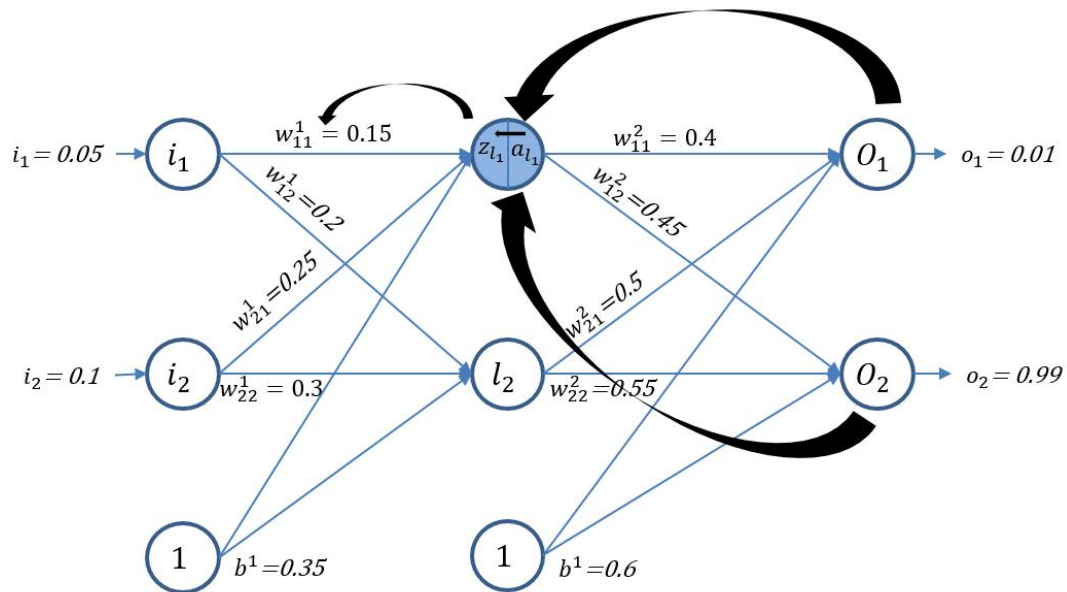
$$w_{11}^{2+} = w_{11}^2 - \eta * \frac{\partial J}{\partial w_{11}^2} = 0.4 - 0.5 * 0.0821 = 0.3589$$

Repeating this process to get the new weight:

$$w_{12}^{2+} = 0.4086; w_{21}^{2+} = 0.5113; w_{22}^{2+} = 0.5613$$

Hidden layer.

For calculating new values for $w_{11}^1, w_{12}^1, w_{21}^1, w_{22}^1$, the original weights are used, not the updated weights. Let consider the w_{11}^1 firstly.



$$\frac{\partial J}{\partial w_{11}^1} = \frac{\partial J}{\partial a_{l_1}} * \frac{\partial a_{l_1}}{\partial z_{l_1}} * \frac{\partial z_{l_1}}{\partial w_{11}^1} = \left(\frac{\partial J_{o_1}}{\partial a_{l_1}} + \frac{\partial J_{o_2}}{\partial a_{l_1}} \right) * \frac{\partial a_{l_1}}{\partial z_{l_1}} * \frac{\partial z_{l_1}}{\partial w_{11}^1}$$

Because a_{l_1} affects both a_{o_1} and a_{o_2} , therefore, $\frac{\partial J}{\partial a_{l_1}}$ needs to take into consideration its effect on both output neurons:

$$\triangleright \frac{\partial J}{\partial a_{l_1}} = \frac{\partial J_{o_1}}{\partial a_{l_1}} + \frac{\partial J_{o_2}}{\partial a_{l_1}}$$

$$+ \frac{\partial J_{o_1}}{\partial a_{l_1}} = \frac{\partial J_{o_1}}{\partial z_{o_1}} * \frac{\partial z_{o_1}}{\partial a_{l_1}}$$

$$\frac{\partial J_{o_1}}{\partial z_{o_1}} = \frac{\partial J_{o_1}}{\partial a_{o_1}} * \frac{\partial J_{a_{o_1}}}{\partial z_{o_1}} = 0.7413 * 0.1868 = 0.1384$$

$$z_{o_1} = w_{11}^2 * a_{l_1} + w_{21}^2 * a_{l_2} + b^1 * 1$$

$$\Rightarrow \frac{\partial z_{o_1}}{\partial a_{l_1}} = w_{11}^2 = 0.4$$

$$\Rightarrow \frac{\partial J_{o_1}}{\partial a_{l_1}} = 0.1384 * 0.4 = 0.0553$$

Following the same process for $\frac{\partial J_{o_2}}{\partial a_{l_1}}$:

$$\frac{\partial J_{o_2}}{\partial a_{l_1}} = -0.019$$

$$\frac{\partial J}{\partial a_{l_1}} = \frac{\partial J_{o_1}}{\partial a_{l_1}} + \frac{\partial J_{o_2}}{\partial a_{l_1}} = 0.0553 - 0.019 = 0.0363$$

$$\blacktriangleright a_{l_1} = \frac{1}{1+e^{-z_{l_1}}} \Rightarrow \frac{\partial a_{l_1}}{\partial z_{l_1}} = a_{l_1}(1 - a_{l_1}) = 0.5932 * (1 - 0.5932) = 0.2413$$

$$\blacktriangleright z_{l_1} = w_{11}^1 * i_1 + w_{21}^1 * i_2 + b^1 * 1 \Rightarrow \frac{\partial z_{l_1}}{\partial w_{11}^1} = i_1 = 0.05$$

Therefore:

$$\frac{\partial J}{\partial w_{11}^1} = 0.0363 * 0.2413 * 0.05 = 0.000437$$

w_{11}^1 is updated following gradient descent's rule:

$$w_{11}^{1+} = w_{11}^1 - \eta * \frac{\partial J}{\partial w_{11}^1} = 0.15 - 0.5 * 0.000437 = 0.1497$$

Repeating this process, we have:

$$w_{12}^{1+} = 0.1995;$$

$$w_{21}^{1+} = 0.2497$$

$$w_{22}^{1+} = 0.2995$$

Finally, all the weights are updated. The error was 0.2983 at the feedforward. After this first round of backpropagation, the total error is down to 0.2910. The error will be decreased after each iteration.