

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
Corso di Laurea in Informatica

Neural networks in the reconstruction of 3D topographic images

Relatore:
Chiar.ma Prof.ssa
Elena Loli Piccolomini

Presentata da:
Michele Cucci

Correlatrice:
Chiar.ma Dott.ssa
Elena Morotti

III Sessione
Anno Accademico 2020/2021

Introduzione

L'obiettivo di questa tesi è di presentare al lettore le possibili implementazioni di algoritmi supportati da reti neurali per la ricostruzione di immagini tomografiche 3D.

La ricostruzione di immagini tomografiche può essere osservata come la risoluzione di un problema inverso, dove il problema diretto è la scansione dell'oggetto desiderato.

Purtroppo, una normale scansione si può rivelare pericolosa per il paziente, poiché soggetto a potenziali massicce dosi di raggi X. Per cercare di ridurre il numero di radiazioni a scansione si stanno implementando nuovi metodi chiamati a *visione scarsa*, ossia ottenere una ricostruzione ottimale cercando di ridurre il numero di immagini effettuate durante una scansione compensando tramite computazioni matematiche.

Se in un ambiente perfetto è possibile ottenere la ricostruzione esatta di una immagine ciò non è possibile in un ambiente reale poiché sono presenti rumore e possibili artefatti. Quindi si implementano funzioni di ricostruzione, il cui risultato sarà il più vicino possibile al limite di convergenza. Ci sono molteplici algoritmi per la ricostruzione di immagini, ma i più comuni applicano una funzione di ottimizzazione e una di regolarizzazione per potersi avvicinare il più possibile al limite.

Tra questi metodi lentamente stanno prendendo piede i metodi iterativi, che permettono una discesa controllata fino ad arrivare al valore minimo. Purtroppo, i metodi iterativi hanno un alto costo computazionale per iterazione quindi un loro utilizzo in un ambiente medico o ospedaliero è estrema-

mente inefficiente per via dei lunghi tempi di attesa necessari per produrre una ricostruzione.

Per compensare a questi tempi d'attesa molte proposte sono state fatte, ma attualmente la più funzionale comporta l'utilizzo dell'intelligenza artificiale, in particolar modo delle reti neurali, per l'acceleramento del processo.

Il primo capitolo sarà dedicato alla tomografia computerizzata. Parleremo della sua evoluzione e delle formule che han portato alla creazione e realizzazione della tomografia come la conosciamo noi oggi.

Nel secondo capitolo verranno mostrati i vari algoritmi utilizzati per la ricostruzione tomografica. Osserveremo il loro funzionamento e le principali differenze tra **algoritmi iterativi** e no. Infine, compareremo i vari algoritmi per osservare quale abbia avuto il risultato migliore in un confronto generale. Il terzo capitolo comincia spiegando brevemente i principi alla base delle reti neurali e come esse funzionano. Dopo illustreremo i componenti principali delle **Reti Neurali Convolutionali**, una versione particolare di rete neurale utilizzata in questa tesi poiché particolarmente capace nell'imparare il riconoscimento delle immagini. Il capitolo si chiude con un'analisi della rete generata, confrontandola con i valori normalmente ottenibili da algoritmi iterativi.

Per concludere, condivideremo alcuni pensieri finali per riassumere e discutere i risultati ottenuti e come possano essere interpretati, seguiti dalla bibliografia.

Introduction

The aim of this thesis is to present to the reader the possible implementations of algorithms supported by neural networks for the reconstruction of 3D tomographic images.

The reconstruction of tomographic images can be observed as the resolution of an inverse problem, where the direct problem is the scanning of the desired object.

Unfortunately, a normal scan can prove to be dangerous for the patient, as it is subjected to potentially massive doses of X-rays. To try to reduce the number of scanned radiations, new methods called *poor vision* are being implemented, i.e., obtaining an optimal reconstruction while trying to reduce the number of images made during a scan by compensating through mathematical computations.

If in a perfect environment, it is possible to obtain an exact reconstruction of an image, this is not possible in a real environment since there is noise and possible artefacts. Then we implement reconstruction functions, the result of which will be as close as possible to the convergence limit. There are multiple algorithms for image reconstruction, but the most common ones apply an optimisation function and a smoothing function to be able to get as close as possible to the limit.

Among these methods, iterative methods are slowly gaining ground, which allow a controlled descent down to the minimum value. Unfortunately, iterative methods have a high computational cost per iteration, so their use in a medical or hospital environment is extremely inefficient due to the long

waiting times required to produce a reconstruction.

To compensate for these waiting times, many proposals have been made, but currently the most functional involves the use of artificial intelligence, especially neural networks, to speed up the process.

The first chapter will be devoted to computed tomography. We will talk about its evolution and the formulas that led to the creation and implementation of tomography as we know it today.

In the second chapter, the various algorithms used for tomographic reconstruction will be shown. We will observe how they work and the main differences between **iterative algorithms**. Finally, we will compare the various algorithms to see which one performed best in a general comparison.

The third chapter begins by briefly explaining the principles behind neural networks and how they work. We will then illustrate the main components of **Convolutional Neural Networks**, a particular version of a neural network used in this thesis as it is particularly capable of learning image recognition. The chapter closes with an analysis of the generated network, comparing it with the values normally obtainable from iterative algorithms.

To conclude, we will share some final thoughts to summarise and discuss the results obtained and how they can be interpreted, followed by the bibliography.

Contents

Introduzione	i
Introduction	iii
1 Computed Tomography	1
1.1 Sparse Tomography	4
1.2 Beer-Lambert Law	6
1.3 Radon transform	8
1.4 Solving the reconstruction problem	9
2 Tomographic reconstruction	11
2.1 Software used	11
2.1.1 CIL	12
2.1.2 ODL	12
2.1.3 ASTRA toolbox	12
2.2 Filtered Back Projection	13
2.3 Iterative reconstruction	14
2.3.1 Linear system	14
2.3.2 Least Squares	15
2.3.3 Total Variation	15
2.4 Iterative algorithms implemented	17
2.4.1 GD	17
2.4.2 CGLS	17

2.4.3 FISTA	18
2.4.4 SGP	19
2.5 Methods comparison	21
3 Neural Network	24
3.1 Convolutional neural network	30
3.2 Learned Post Processing	32
3.3 Technical Notes	35
3.4 Analysis of the network	37
Conclusions	43
Bibliography	45

Chapter 1

Computed Tomography

Computed tomography (CT) is a medical imaging technique that permits the performance of a noninvasive exam of a patient for the diagnosis and study of tumours and numerous other pathologies.

It is a radiological examination in which the data is collected by the passage of various X-ray beams in the affected area and then processed by a computer, in order to reconstruct a three-dimensional image of the scanned area. The first step towards computed tomography was made by the discovery of X-rays by the German physics professor Wilhelm Röntgen in 1895 and their potential application in the medical field.

Then in 1917 it was followed by the formulation of the **Radon transform** function which showcased a mathematical demonstrations of how a function $f(x,y)$ could be reconstructed from an infinite set of its projections. This function will be explained in more detail in chapter *1.1.2*.

Finally, the first prototype of a computed tomography scanner was conceived in 1967 by the English engineer Sir Godfrey Hounsfield, together with the South African physicist Allan Cormack. This discovery earned the two scientists the Nobel Prize for medicine in 1979. The prototype featured a scanning unit that rotated 180° , one degree at a time, around the patient's head, storing 160 images from each position, for a total of 28,800 images; this process

took from 5 to over 10 minutes.

The studies continued and, in 1975, culminated in the creation of the first CT machinery, the Automatic Computerised Transverse Axial Scanner (ACTA), capable of taking images of any part of the body. This machine had 30 photomultiplier tubes as detectors and was able to complete a scan in just nine translation and rotation cycles, a much faster speed than EMI's equipment. Nowadays, the sector is still developing but has had dramatic growth and significance over the past decades. For example, in 2008, the industry-leading *Siemens AG*, a German multinational conglomerate corporation, introduced a new generation of scanners capable of obtaining images in less than a second.

Modern tomographic scanners are primarily composed of an emitter of an X-ray beam that rotates around the patient and a detector, on the opposite side, that collects the image of a section of the patient.

But the size and type of either evolved considerably during the course of the years and thus scanners nowadays can be classified in four generations based around the changes of those parameters.

The first generation of scanners (see top left sub-figure of figure 1.1) were mainly composed of an X-ray tube that emitted a linear beam of X-rays. To perform a scan, the X-ray tube first performed a translation and then a rotation of one degree for 180 degrees, for a total of 180 angular projections \times 160 projection rays.

In the second generation, two major changes were introduced: the geometry of the radiant beam changed from a linear shape to a "fan-shaped" beam, as shown in top right sub-figure of figure 1.2, with an amplitude of around 20-30 degrees, and consequently, the number of detectors increased from 1 to a group of 20-30. With this generation, a single scan took only a few tens of seconds.

Third generations were introduced thanks to a general improvement in technology; now the "fan-shaped" beam changed to an amplitude of around 30-50 degrees, and the number of detectors also increased widely, reaching several

hundred elements. This meant that scans were able to include the entire anatomical portion to be studied.

A fourth generation based scanner presented detectors laid in a static ring placed all around the patient, replacing the rotating detector array (as shown in the bottom images of figure 1.1). This cut down the time required for a scan from approximately 5 seconds for a third generation scanner down to a few seconds.

Modern tomographs now present a fundamental characteristic, that of an acquiring spiral. In fact, in the unidirectional continuous rotation tomographs, the X-ray tube and the detectors are mounted on a rotating ring which is powered by "sliding contacts." This method allows the acquisition of images in a continuous way.

As the X-rays pass through the patient, they are attenuated differently by various tissues according to the tissue density defined by the *Beer-Lambert's law*. A visual representation of the raw data obtained is called a *Sinogram*. In conclusion, computed tomography (CT) use has increased dramatically over the past several decades. Integrating CT into routine care has improved patient health care dramatically, and CT is widely considered among the most important advances in medicine. However, CT delivers much higher radiation doses than conventional diagnostic x-rays. For example, a chest CT scan typically delivers more than 100 times the radiation dose of a routine frontal and lateral chest radiograph^[1].

Furthermore, radiation exposure from CT examinations has also increased, in part due to the increased speed of image acquisition, allowing vascular, cardiac, and multi-phase examinations, all associated with higher doses. Thus, greater use of CT has resulted in a concurrent increase in medical exposure to ionizing radiation^[2].

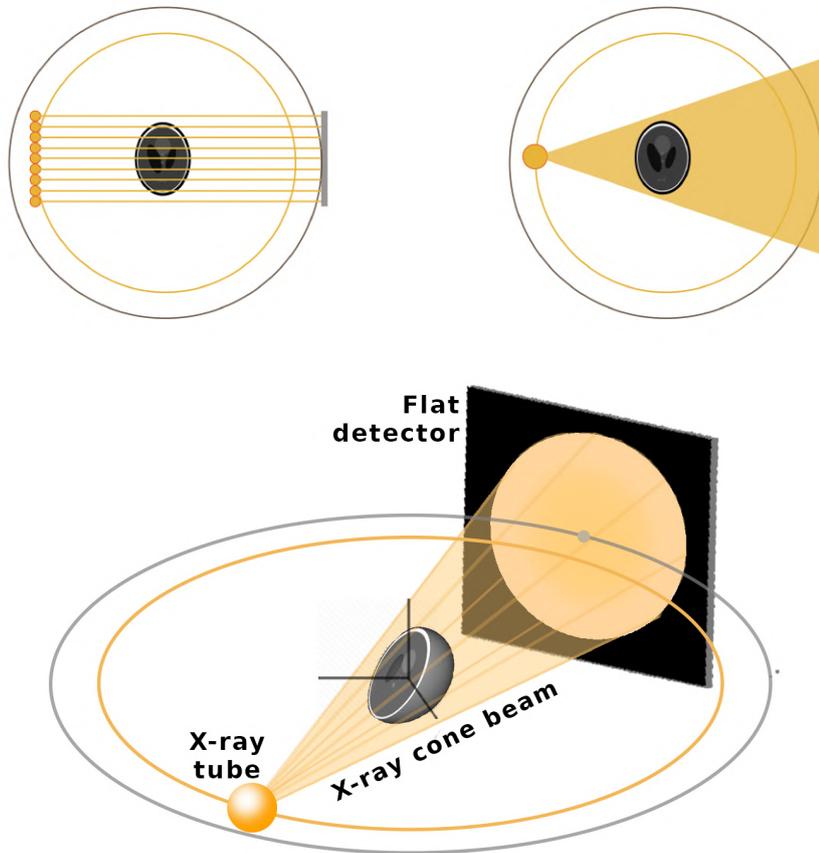


Figure 1.1: Sketches of tomographic devices, from the primordial technology with parallel X-rays scans (top left), to the most modern solution exploiting fan-beams for 2D (top right) and cone-beams (bottom) for 3D CT.

1.1 Sparse Tomography

As mentioned in the above paragraph, one of the biggest concerns regarding the use of CT scanners is how to reduce the radiation dose without compromising the image quality.

As one can imagine, higher radiation doses usually mean higher resolution images due to less attenuation caused by tissues and vice versa. However, increased dosages of radiation can increase the adverse side effects.

The patient surface dose from a typical abdominal CT is 200–300 times more than that of a typical chest radiograph, 20–30 times that of the mean glandular dose of a typical craniocaudal mammogram, and approximately 10–20 times that of typical abdominal radiography. [3]

As such, motivated by an increasing focus on the potentially harmful effects of X-ray radiation, a recent trend in CT research is the development of *few-views imaging techniques* [4], whose graphical drafts are schematically reported in Figure 1.2.

Hence, in classical full-dose CT (first row of Figure 1.2), up to one thousand projections are taken along the circular trajectory, hence, one way to reduce the total amount of radiation consists of decreasing the number of projections. The resulting protocol is labelled as *sparse-views full-angle tomography*.

A different CT geometry using few projections is the so-called *limited-angle tomography* where a further reduction of X-ray scans is made by limiting the source trajectory to a C-shaped path, i.e., by restricting the 360-degree angular scanning interval to a range smaller than 180 degrees (see second row of Figure 1.2). In some tomographic applications, human anatomy does not allow a complete circular motion to the X-ray source, thus the use of a reduced range is mandatory and the resulting technique is called *tomosynthesis*.

In the medical field, low-dose compared with full-dose protocols allows applying CT techniques to a wider class of examinations, including vascular, dental, orthopaedic, musculoskeletal, chest, and mammographic imaging. In particular, they are approved even for screening tests: safer routines are indeed led without compromising the reliability of their diagnosis [5]. In addition, they are widely used not only in medicine but also in materials science and cultural heritage, to prevent damage to the subject under study due to excessive radiation.

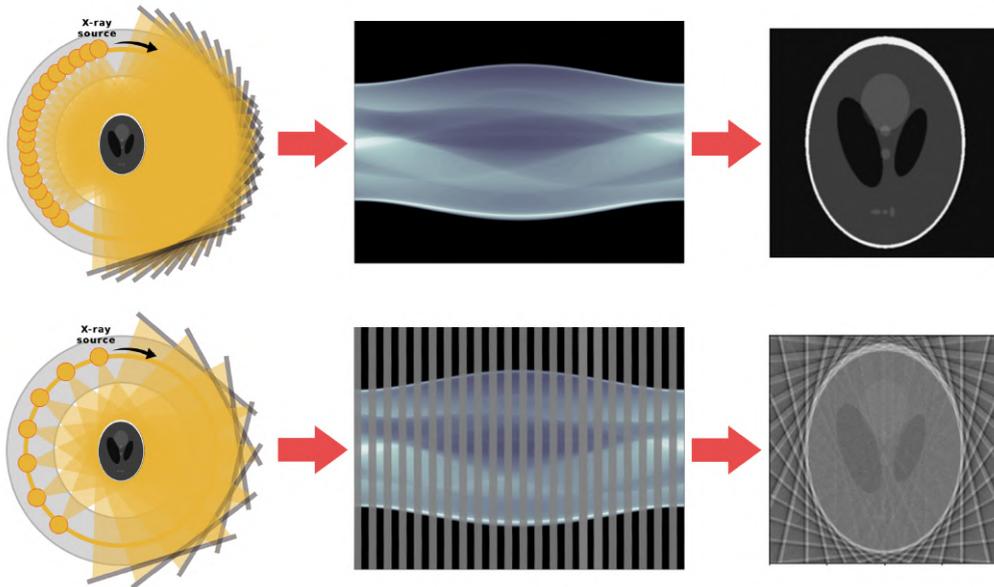


Figure 1.2: Types of tomographic reconstruction; top sketch illustrate a full-views techniques, bottom sketch a sparse-views technique

1.2 Beer-Lambert Law

As mentioned before, when an X-ray pass through an object, they are attenuated by the the type and density of the material they are passing through. From a physical point of view, the projection data reflect the absorption of the photons constituting the X-rays, and the image of the scanned object is a picture of the attenuation coefficient map in pseudo-colours.

The physical model describing photons absorption is described in the *Lambert Beer's law*.

The Lambert Beer's law relates the attenuation of light to the properties of the material through which the light is travelling.

In the most simple case, assuming a mono-energetic beam and a homogeneous sample of thickness dl (as shown in Figure 1.3), *Lambert-Beer's law of attenuation* can be expressed as:

$$m(w) = m_0 \cdot e^{-\mu w} \quad (1.1)$$

With μ being the linear attenuation coefficient for the crossed point w and imposing the initial condition of $m(0) = m_0$ where m_0 is the emitted photon count.

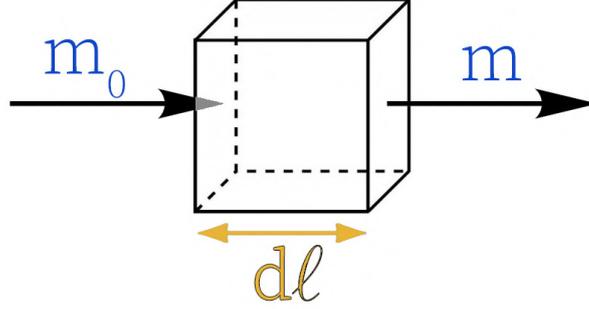


Figure 1.3: application of the Lambert-Beer's law through an object of length $d\ell$

In practice, the attenuation coefficient $\mu(w)$ is not constant along the ray path. So for inhomogeneous samples the equation above changes into:

$$m = m_0 \cdot e^{-\int_0^W \mu(w)dw} \quad (1.2)$$

With w being the beam direction. The integral in the exponential is also known as line integral $\mathcal{P}_{W\mu}$ and can be computed as:

$$\mathcal{P}_{W\mu} = -\ln\left(\frac{m}{m_0}\right) = \int_0^W \mu(w)dw \quad (1.3)$$

This equation represents the projection image, which is proportional to the integrated attenuation coefficient of a sample from a certain beam direction, furthermore by setting the plane coordinates as $w = (x, u)$ and renaming L the integration path we obtain:

$$\mathcal{P}_{W\mu} = -\ln\left(\frac{m}{m_0}\right) = + \int_L \mu(x, y)dw \quad (1.4)$$

1.3 Radon transform

Thanks to the application of Beer-Lambert's law, we can now obtain the projected distribution (profile) of the object, but what can be known about the actual distribution?

This problem has long been of interest to mathematicians, and with the advancements in experimental science and other applied fields, we also needed a solution to this issue. A solution to the reconstruction problem came with the formulation of the *Radon transform* and its inversion formula in 1917 by Johann Radon.

Initially, the formula was only for problems in \mathbb{R}^2 and \mathbb{R}^3 dimensions, but it was later expanded to resolve equations in \mathbb{R}^n .

The generalised Radon transform, Rf , is a function defined in the space of straight lines $L \subset \mathbb{R}^2$ by the line integral along each such line as:

$$Rf(L) = \int_L f(x)|dx| \quad (1.5)$$

The Radon transform of μ is defined as the map $\mathcal{R} : [0, 2\pi] \times \mathbb{R} \rightarrow \mathbb{R}$ such that:

$$\begin{aligned} (\mathcal{R}\mu)(\theta, t) &= \mathcal{P}_\theta\mu(t), \quad \forall \theta \in [0, 2\pi], \forall t \in \mathbb{R} \\ \mathcal{P}_\theta\mu(t) &= \int_L \mu(x, y)dw \end{aligned} \quad (1.6)$$

In other words, the Radon transform \mathcal{R} of an object slice described by μ is the set of projections acquired along the whole circular trajectory. The process defining full-dose tomography represents a discrete realization of the (continuous) Radon transform.

The Radon transform data is often called a *sinogram* because the Radon transform of an off-center point source is a sinusoid. Consequently, the Radon transform of a number of small objects appears graphically as a number of blurred sine waves with different amplitudes and phases.

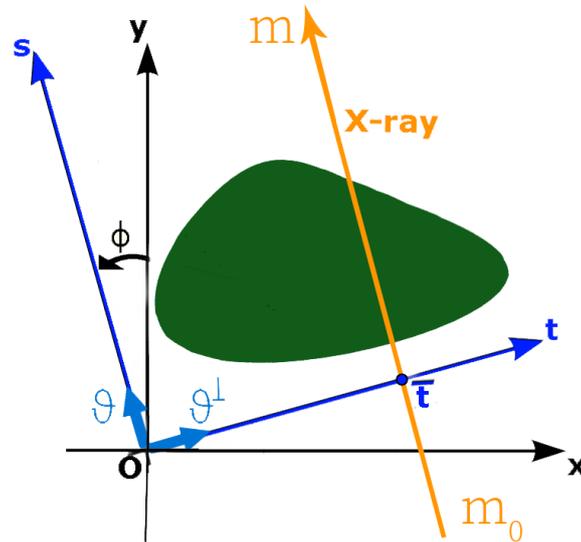


Figure 1.4: Geometry of the line integral associated with the Radon transform

1.4 Solving the reconstruction problem

In CT scanning, we can represent the acquisition of the sinogram from the object scanned by the machine as a forward problem, as it starts with the causes and then calculates the effects. To obtain the reconstruction of the object from the sinogram, the software reconstructs the digital image from the acquired projection data. Hence, tomographic image reconstruction represents an *inverse problem* mathematically.

In an inverse problem, which can be seen represented in Figure 1.5, only the measurements of an effect are known, as given data, whereas the causal factors represent the unknown that must be retrieved as a solution to the problem.

Hence, in computed tomography, the lines on which the parameter is integrated are straight lines; hence, the tomographic reconstruction of the parameter distribution is based on the inversion of the Radon transform, although the Radon transform and its generalisations still present many theoretical

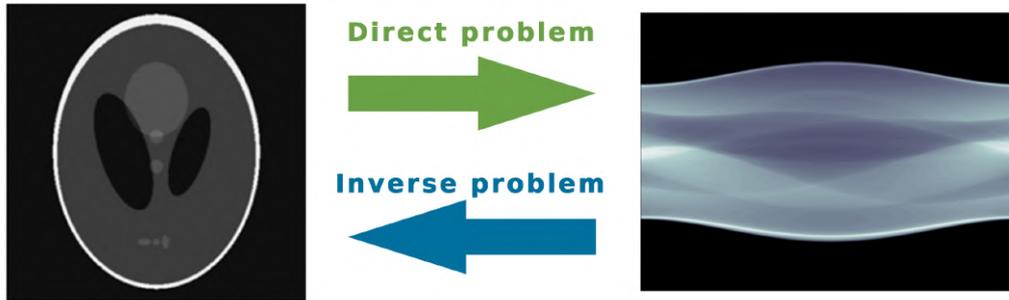


Figure 1.5: Inverse Problem

challenges, with questions of sufficiency of data still unresolved.

Mathematically, inverse problems are generally ill-posed in the sense of Hadamard [6], I.e., one of the following conditions is not satisfied:

1. At least one solution to the problem already exists.
2. The solution to the problem is unique.
3. The solution does not continuously depend on the data.

Traditional methods for CT can't face the ill-posedness and compute images with unwanted artefacts and noise. Textbf filtered backprojection and, as computing power has increased, textbf iterative reconstruction algorithms were investigated as solutions. This is possible thanks to a more recent approach that models the CT imaging process as an optimisation problem where the inverse problem is solved by inverting the discrete model, represented by a linear system, constrained by means of regularisation functions. Imposing regularisation also allows one to choose a good solution among the infinite possible ones.

Chapter 2

Tomographic reconstruction

Initially, CT scanners solved the tomographic reconstruction problem with linear algebra, but this approach was limited by its high computational complexity, especially given the computer technology available at the time.

To improve the quality of tomographic reconstructions starting with only a finite, limited number of projections, many algorithms have been proposed to help significantly reduce unwanted noise and artefacts. Modern used algorithms are primarily based on the inverse of the Radon transform and the geometry of the scanner.

This chapter will illustrate the tomographic reconstruction software implemented and present a brief description of the algorithms studied, how they work, and the differences between them.

2.1 Software used

Before we begin illustrating the algorithms, it's important to illustrate which software/toolboxes were utilised for repeatability purposes.

All the algorithms presented below were written in Python 3.9 using the toolboxes *CIL*, *ODL* and *ASTRA toolbox*.

2.1.1 CIL

The Core Imaging Library [\[7\]](#) (CIL) is an open-source Python framework for tomographic imaging with particular emphasis on the reconstruction of challenging datasets. CIL provides an extensive modular optimisation framework for prototyping reconstruction methods, including sparsity and total variation regularisation, as well as tools for loading, preprocessing, and visualising tomographic data.

2.1.2 ODL

The Operator Discretization Library (ODL) is a Python library for fast prototyping focusing on (but not restricted to) inverse problems. ODL is being developed at KTH Royal Institute of Technology, Stockholm, and Centrum Wiskunde and Informatica (CWI), Amsterdam. The main intent of ODL is to enable mathematicians and applied scientists to use different numerical methods on real-world problems without having to implement all the necessary parts from the bottom up. This is achieved by an Operator structure that encapsulates all application-specific parts, and a high-level formulation of solvers that usually expect an operator, data, and additional parameters.

2.1.3 ASTRA toolbox

ASTRA Toolbox [\[8\]](#) is an open platform for 3D image reconstruction in tomography. Most of the software tools that are currently used in electron tomography offer limited flexibility with respect to the geometrical parameters of the acquisition model and the algorithms used for reconstruction. The ASTRA Toolbox provides an extensive set of fast and flexible building blocks that can be used to develop advanced reconstruction algorithms, effectively removing these limitations. A series of experiments based on experimental dual-axis tilt series demonstrate this flexibility, the resulting reconstruction quality, and the computational efficiency of this toolbox.

2.2 Filtered Back Projection

One of the first algorithm ever introduced in the CT field was the *Filtered Back Projection* (FBP), which is a stabilised and discretized version of the inverse *Radon transform*.

As the name implies, it is mainly composed of two main steps: firstly it applies a *filter* to the data received as sinogram to reduce the blurring present and then *backproject* applying the inverse of the Radon transform.

With a sampled discrete system, the formula for the inverse Radon transform is [9]:

$$f(x) = \int_0^\pi (\mathcal{R}f(\cdot, \theta) * h)(\langle \mathbf{x}, \mathbf{n}_\theta \rangle) d\theta \quad (2.1)$$

In a real-world discrete setting, FBP assumes a perfect scanner and highly simplified physics, which is extremely difficult to implement. This can lead to a number of artefacts, high noise, and impaired image resolution. Nevertheless, the FBP algorithm is still implemented in many commercial systems, since it computes the output image in a very short time, which is a fundamental request in a medical setting.

A possible alternative is the implementation of a model-based iterative algorithm, which can be derived from the discretization of Lambert-Beer's Law. Compared with the Filtered Back-projection method, iterative reconstruction costs large computation time, limiting its practical use. However, due to the ill-posedness of Radon Inversion, the Filtered Back-projection method may be infeasible in the presence of discontinuity or noise. Iterative reconstruction methods provide metal artefact reduction, noise and dose reduction for the reconstructed result that attract much research interest around the world.

2.3 Iterative reconstruction

Iterative Image Reconstruction (**IIR**) method, or also commonly abbreviated as iterative algorithm are rising in popularity in the last periods due to their ability to provide images with improved resolution, reduced noise and fewer artefacts, as well as the ability to greatly reduce the radiation dose in certain circumstances, they can also introduce a priori information about the object scanned and exploit the *Compressed Sensing* (CS) theory for reconstructing a signal or an image from a reduced number of acquisitions with respect to the Nyquist theory [10].

Their main drawback is their higher computational cost with respect to the analytical methods, but due to a combination of new technologies and new algorithm using massive parallelism it's possible to realise fast iterative algorithm with acceptable results for medical settings [11].

2.3.1 Linear system

Whereas the FBP algorithm resolves a Radon transform for obtaining the reconstructed image, iterative methods require the implementation of a linear equation, but are applied to real-world settings.

When implementing a real discrete settings, we must first remember that the scanner and the scanned object are both discrete. Thus the attenuation coefficient $\mu(x, y)$ that we obtain from the application of the Beer-Lambert's law (1.4) is also discretized in an image of form $N = N_x \times N_y$, having n_p as the number of recording units for a detector and N_θ projections at equally spaced angles $\theta_k, \forall k = 1, \dots, N_\theta$

Now, using mathematical notation, we can describe the CT process using a linear system in the form:

$$Ax = b \tag{2.2}$$

where $A \in \mathbb{R}^{N_d} \times \mathbb{R}^N$ is the matrix describing the system geometry, having $N_d = N_\theta \times n_p$; $b \in \mathbb{R}^{N_d} \times \mathbb{R}^\theta (b > 0)$ is the vector of recorded projections

and $x \in \mathbb{R}^{N_v}$ is the discretisation of $\mu(x, y)$ in the N_v voxels of the objects. However, in a real-life discrete setting, the linear system [2.2] is under-determined, having $N > N_d$, hence it has infinite possible solutions. Moreover, due to the ill-posedness of the inverse problem and to a lack of data, noise and streaking artefacts corrupt the solutions, thus regularisation strategies are necessary.

A model-based approach is introduced to overcome these numerical controversies by adding some prior information. The resulting formulation can be stated as a minimisation problem involving a data-fitting function \mathcal{J} and a prior operator \mathcal{R} (acting here as a regulariser):

$$\underset{x}{\operatorname{argmin}} f(x) = \mathcal{J}(x) + \lambda \mathcal{R}(x) \quad (2.3)$$

2.3.2 Least Squares

For the implementation of the data-fitting function \mathcal{J} , one of the most commonly used approach is the *Least Squares* (LS) operator:

$$LS(x) = \| Ax - b \|_2^2 \quad (2.4)$$

The LS function perfectly fits with the assumption of Gaussian noise components on the available data b . The formula was then implemented in the SGP algorithm as [12]:

$$\mathcal{J}(x) = \frac{1}{2} \| Ax - b \|_2^2 \quad (2.5)$$

2.3.3 Total Variation

The total variation function [13] (TV), also known as total variation regularisation or total variation filtering, is currently the most widely used regularisation function for CT problems, so it was chosen as the perfect candidate for $\mathcal{R}(x)$.

It is based on the principle that signals with excessive and possibly spurious

detail have a high total variation, that is, the integral of the absolute image gradient is high. The concept was pioneered by Rudin, Osher, and Fatemi in 1992 and so is today known as the ROF model [14] :

$$TV(x) = \sum_{j=1}^N \|\nabla x_j\|_2 \quad (2.6)$$

or in its smoothed differentiable form as:

$$TV_\beta(x) = \sum_{j=1}^{N_v} \sqrt{\|\nabla x_j\|_2^2 + \beta^2} \quad (2.7)$$

where β is a small positive parameter.

2.4 Iterative algorithms implemented

2.4.1 GD

Gradient descent (GD) is an optimisation algorithm, based on convex or differentiable functions that tweaks its parameters iteratively to minimise the cost-given function to its local minimum.

It operates on the concept of *gradients*, which is the first derivative at a selected point or, in the case of a *multivariate function* the vector of derivatives in each main direction.

The GD algorithm works by iteratively calculating the next point using gradient at the current position, then scales it by a learning rate and subtracts obtained value from the current position (thus making a step). It subtracts the value because we want to minimise the function (to maximise it would be adding)(see Figure 2.1).

Formally, if we start at a point x_k and move a positive distance, α in the direction of the negative gradient, then our new and improved x_{k+1} will look like this:

$$x_{k+1} = x_k - \alpha_k \nabla f(x) = x_k + \alpha_k (b - Ax_k) \quad (2.8)$$

2.4.2 CGLS

The *conjugate gradient method* is an algorithm for the numerical solution of particular systems of linear equations, namely those whose matrix A is symmetric ($A^\top = A$) and positive definite ($x^\top Ax > 0$).

$$\begin{aligned} &\text{Solve } Ax = b \\ &\text{or minimise } \|Ax - b\|^2 \\ &\text{or solve } (A^\top A + sI)x = A^\top b, \end{aligned}$$

where the matrix A may be square or rectangular and s is a scalar (positive or negative). The method is stable if $s = 0$ or $s > 0$. More generally, it should

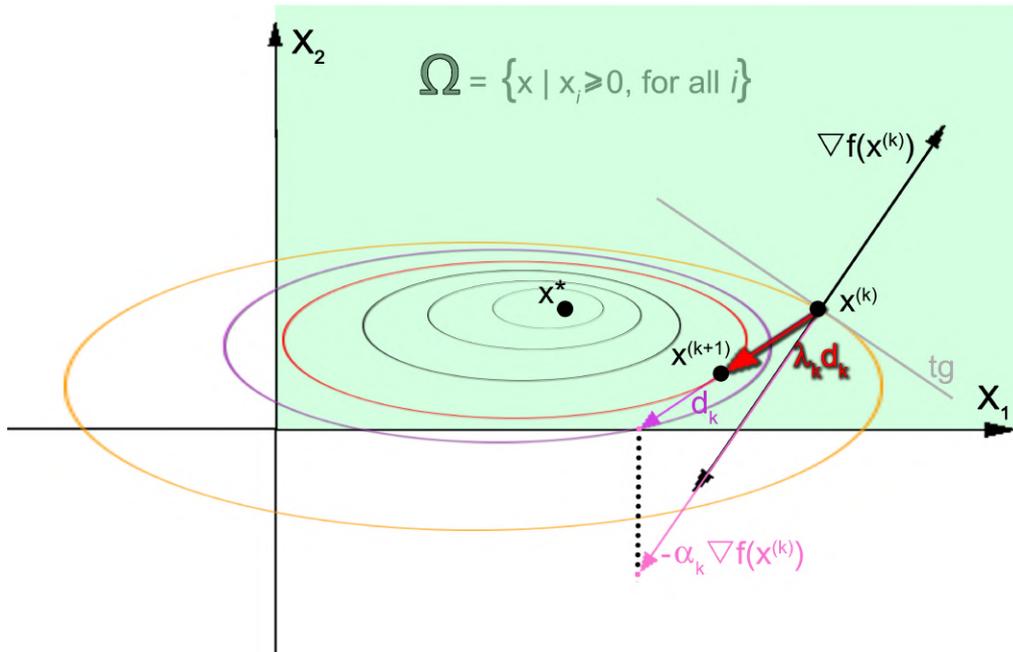


Figure 2.1: Method of the gradient descent

be stable if $A^T A + sI$ is positive definite. Otherwise, it may be unstable.

2.4.3 FISTA

The fast iterative shrinkage-thresholding algorithm (FISTA) [15] it's the evolution of the *iterative shrinkage-thresholding algorithms* (ISTA). FISTA preserves the computational simplicity of ISTA but with a global rate of convergence that is proven to be significantly better, both theoretically and practically. The concept is based around the general step of ISTA, which in linear form is:

$$x_{k+1} = \mathcal{T}_{\lambda t}(G(x_k)) \quad (2.9)$$

where $G(\cdot)$ stands for a gradient step of the fit-to-data LS function. FISTA address the main problem with ISTA algorithms, the slow convergence of x_k to a solution due to the fact that they are methods based on function values and gradient evaluations. The solution proposed in FISTA is to implement

a faster method similar to ISTA in the form:

$$x_{k+1} = \mathcal{T}_{\lambda t}(G(y_k)) \quad (2.10)$$

where the new point, y_k uses a very specific linear combination of the previous two points, x_{k-1}, x_{k-2} . Due to its minor role in this thesis, a detailed proof of the statement unfortunately exceeds the scope of this paper. We refer the reader to existing literature on the topic for details of this [15].

2.4.4 SGP

The *scaled projection* algorithm (SGP) is a scaled gradient projection algorithm accelerated by exploiting a scaling strategy for defining gradient-based descent directions and generalised Barzilai–Borwein rules for the choice of the step-lengths. [16] It solves a minimisation problem of the form:

$$\min_{x \geq 0} f(x) = LS(x) + \lambda TV_{\beta}(x) \quad (2.11)$$

SGP can be seen implemented in pseudo-code in the algorithm next page:

Algorithm 1 SGP($x_0, \maxIterations, toleranceGradient, \alpha = 1$)

$x = x_0 \wedge k = 0$

$grad = grad(LS(x) + \lambda * TV(x))$

$RelativeError = \frac{\|x - groundTruth\|}{\|groundTruth\|}$

$\rho = \sqrt{1 + 1e^{15}}$

$s = ScalingMatrix(x, \rho)$

CONTINUE $\leftarrow True$

while CONTINUE **do**

$directionDescent = Projection(x - \alpha * s * grad) - x$

$StepLength = BacktrackingStep(x, f(x), grad, directionDescent)$

$RelativeError = \frac{\|x - groundTruth\|}{\|groundTruth\|}$

$k += 1$

$x = x + StepLength * directionDescent$

$grad = \|grad\|$

if ($\|grad\| > toleranceGradient$) \wedge ($k < \maxIterations$) **then**

$grad_0 = grad$

$grad = grad(LS(x) + \lambda * TV(x))$

$\rho = \sqrt{1 + 1e^{15}/k^{2.1}}$

$s = ScalingMatrix(x, \rho)$

$\alpha = BarzilaiBorweinRules(s, x_0, x, grad_0, grad, alpha)$

else

 CONTINUE = *False*

end if

end while

Return ($x, k, grad, RelativeError, f(x)$)

2.5 Methods comparison

In this section, we will discuss the differences in the performance of each algorithm, looking at the quality of reconstructed images and *relative errors*. With relative error, it is intended as the result of a normalisation applied to the difference between the true image and the reconstructed one.

Each algorithm were tested each two times on a *Shepp-Logan phantom* of size $128 \times 128 \times 128$, the first time doing only 10 iterations whenever possible and the second time 100 iterations or till convergence.

The Sheep-Logan phantom is one of the most famous phantom used for testing reconstruction algorithms, it is based on ellipsoid vaguely resembling a human head (see figure 2.2),it was created by Larry Shepp and Benjamin F. Logan for their 1974 paper *The Fourier Reconstruction of a Head Section* [17].

To the phantom it was also applied *Gaussian noise*, which is statistical noise that can take on values that are Gaussian-distributed, with a standard deviation of $1e^{-3}$ to see how the various algorithms would reconstruct an image in a not-so-perfect setting.

Algorithm used	Relative error after 10 iterations	after 100 iterations
Fbp	0.23763588	0.23763588
Cgls	0.29899165	0.17062649
Gd	0.646686	0.3288964
Fista	0.48350394	0.09436152
Sgp	0.35354166	0.10564623

Table 2.1: Relative error of each algorithm

Looking at table 2.1, we can see that Fbp is the only non-iterative algorithm tested, proven by the fact that its relative error remains unchanged. Its relative error is one of the lowest overall, but the quality of the image is not very close to the original. The noise streaks are visible and quite apparent,

and two lines not present on the original can be seen adjacent to the reconstruction.

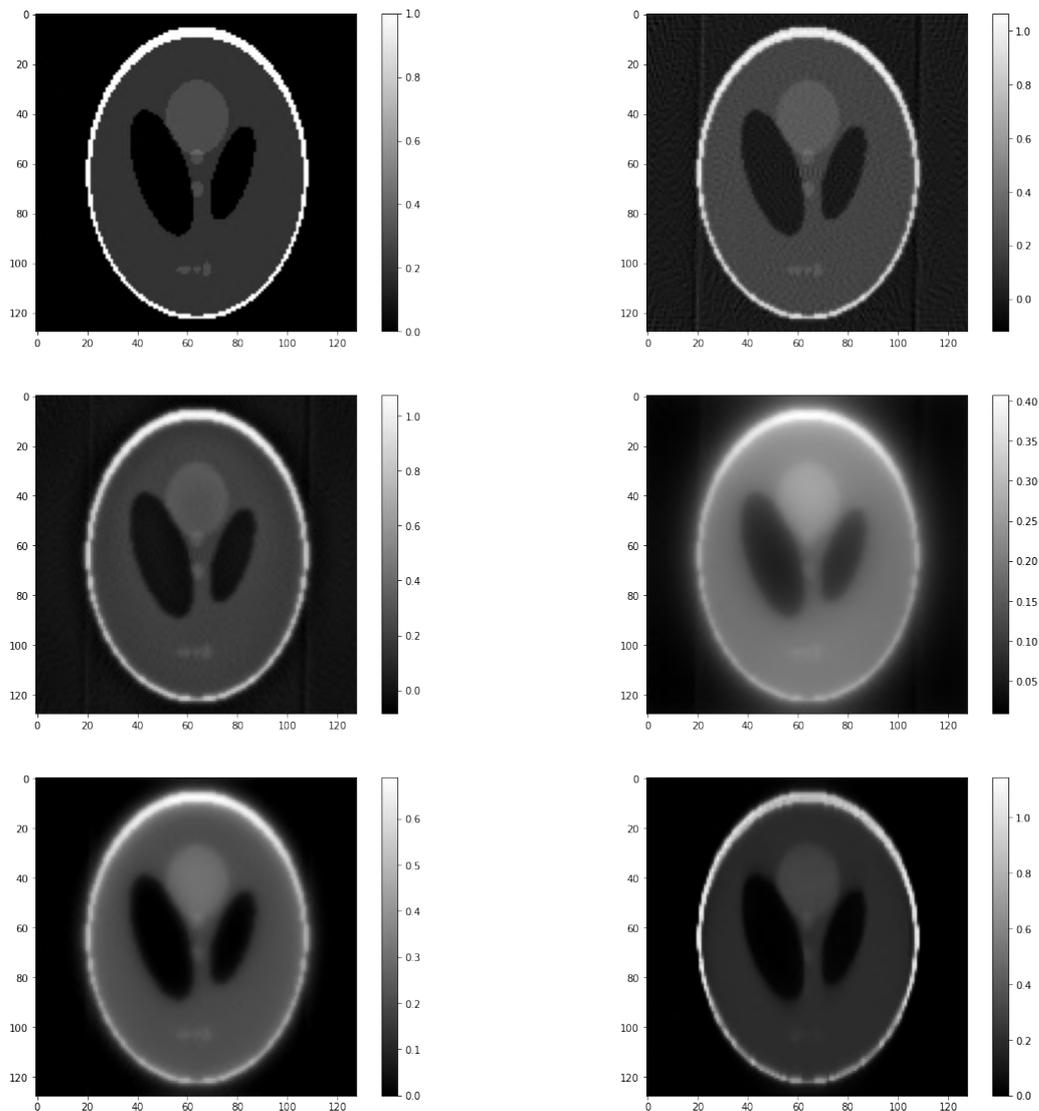


Figure 2.2: Reconstruction images of the phantom (top-left) after 10 iterations from top to bottom, left to right: phantom, fbp, cgl, gd, fista, sgp

CGLS is next and when looking at the relative errors we can see a small improvement, only halving the error after 90 iterations plus, but unfortunately

when looking at the image almost the exact situation reappear as for FBP, only reduced. Two lines that were not present on the original phantom are present adjacent to the reconstruction, but overall the image seems improved, although a little blurred.

Gradient Descent is the most particular case, having one of the highest relative errors, almost doubling the error of other algorithms after only 10 iterations. Even after 100 iterations, it still remains the highest, although reduced by almost half from 10 iterations. Looking at the original phantom and then the GD reconstruction, it appears that the Gradient Method not only bumped its brightness quite a lot, thus making the smaller ellipsoids disappear, but it also appears quite blurred, making it harder to recognise particular forms. While in this case it was not a cause of concern, in a medical setting it can be quite troublesome due to tissues not being quite recognisable.

Continuing further, we find Fista, which is one of the most recent algorithms published in this comparison, and its accuracy can be seen by looking at the table. Whilst the relative error after 10 iterations is the second highest of the group, after 100 iterations we can see it dropping down even below $1e^{-2}$, which is astounding. Looking at the image, we can see that, while a little blurred and brighter than most, it's one of the cleanest so far.

Finally, we arrive at Sgp. Sgp presents the best relative errors so far, while not the lowest of the group, they are quite close to having a relative error after 100 iterations equal to $1e^{-2}$. Instead, by looking at the image, we can see that, while it's difficult to recognise some of the smaller details, the contrast is quite clear and a lot less blurred than the other image.

Having confronted this algorithm, it was then decided to adopt Sgp as the method to continue working on, giving the clearest images whilst having some of the lowest errors.

Chapter 3

Neural Network

With *artificial neural networks*, or NN in short, we mean a sub-branch of artificial intelligence inspired by the biological neural networks that make up the brains of animals.

A neural network is a computational framework including a large number of *neurons* as basic computing units connected to each other with varying connection strengths known as "weights." This feed-forward, layered architecture is mainly intended to reflect the process of extracting visual features from biological neurons layer by layer. An artificial neural network is composed primarily of the following elements: neurons with weights and activation functions; network topology, or *connections*, which ought to be trained with training data according to some learning rules.

Neurons are the core of the network and the most elementary unit. They are capable of receiving information in input from other neurons, processing them, and then transmitting them to other neurons. This can be illustrated by a mathematical expression:

$$\begin{cases} v = \sum_{i=1}^m x_i w_i + b \\ y = \varphi(v) \end{cases}$$

where w_i represents the weight for each input signal component x_i , b is a bias, v is the sum of the input vector and the weight vector and y represents the output of the neuron after the application of a nonlinear *activation function* φ to v .

The activation function is an essential component of a neuron because it defines the output behaviour of the neuron and empowers the network thanks to its nonlinear mechanism. This non-linearity enables the artificial neural network to learn a complex nonlinear mapping from input to output signals without it, the network would be a linear system whose information processing capability would be very limited.

Generally speaking, the activation function delivers a single number via a "soft" thresholding operation as the final result of the information processing processed by the neuron. Two of the most commonly used functions are the *Sigmoid* and the *ReLU*.

The Sigmoid function sets the output value in a range between 0 and 1, where 0 represents not activated at all, and 1 represents fully activated.

$$\varphi(v) = \frac{1}{1 + e^{-v}}$$

Currently, the sigmoid nonlinearity is rarely used, because it has the major drawback of saturating when activated at either 0 or 1, and because the gradient in these regions is almost zero, this is undesirable for optimisation of the network parameters.

The *rectified linear unit* function (ReLU) works better than a smooth function like the Sigmoid, while also being significantly easier to compute. ReLU outputs 0 if its input is less than 0; otherwise, it just reproduces the input. The mechanism of ReLU is more like the biological neurons in the visual cortex.

$$\varphi(v) = \max(0, v)$$

There are two major merits of the ReLU function: there is no saturation zone for positive stimulation and there is no gradient. And, in the network

training process, the convergence speed of ReLU is much faster than that of Sigmoid.

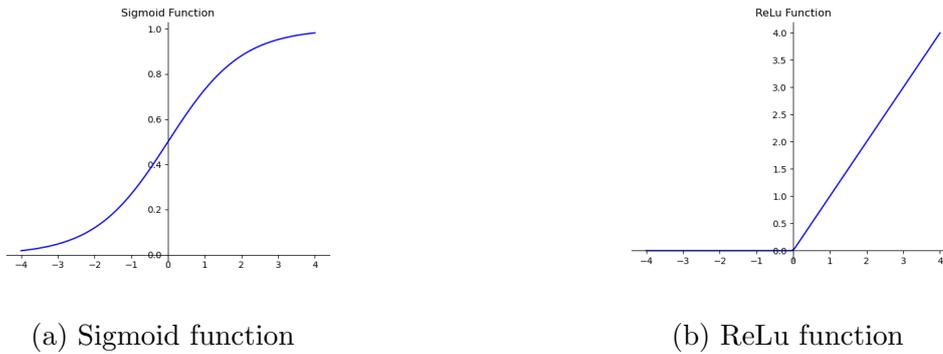


Figure 3.1: Activating function

Neurons combined form layers which can detect the type of feature of their input data. It could be the input layer or a hidden layer. Different types of networks can be created by combining different neurons each time, which will cause new layers to be created and old layers to be modified or even removed.

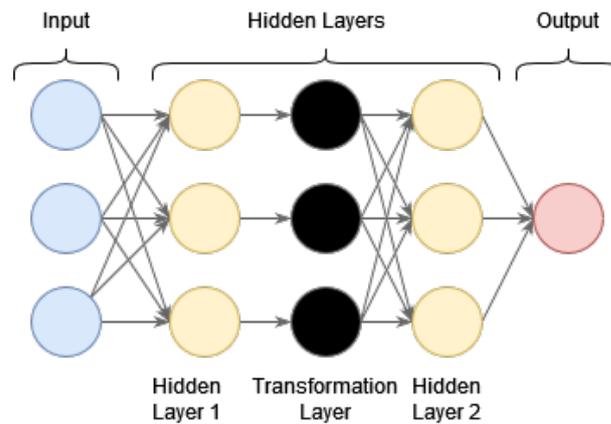


Figure 3.2: Graph of a three-layer model with activation function.

To understand how a neural network works, we must first take a look at the overall process of implementation and analyse each step. The whole process can be summarised into the following points:

1. *Design*: Based on the task, we design the neural network architecture, the topology and the details on *convolution*, *activation*, *loss function*, and choose the value for the *hyperparameters*.
2. *Forward projection*: Training samples are then fed into the network to produce some outputs. (Forward projection) The input samples are samples that originate from a training dataset used to train the network, while some samples are for creating a testing dataset to verify the level of convergence of the network.
3. *Backpropagation*: The weight of each neuron is then updated after checking the relative error using the backpropagation method.
4. *Repeat*: Repeat point 2 and 3 until the network converges. We have trained the network for one epoch.

The main concept behind the implementation and usage of a neural network is its ability to grow thanks to its capability to reinforce or weaken connections between neurons throughout the learning process by adjusting the weight of each neuron.

When a neural network is created, initially it sets all the weights randomly. As such, the initial iterations perform poorly. It still needs to adjust its weight iteratively, by comparing the error between the training values obtained and the predicted values, usually given by the developer, so as to obtain a converged or trained neural network with an appropriate internal representation, which can then map from an input to a desirable output. Such a training process is almost exclusively done using the backpropagation algorithm.

Backpropagation is a shorter term for ‘error backward propagation’ and is essentially a gradient descent optimisation. In backpropagation, it calculates

a loss function at the output layer of the network and then distributes it backwards throughout the network layers.

The loss function is used to measure the discrepancy between a predicted value \hat{y} and the corresponding label y . It is a non-negative function, whose minimisation drives the performance of the network to reach convergence in the training stage.

Training a neural network is to update the network parameters so that \hat{y} approaches y as closely as possible by some certain measure. The loss function can take a variety of forms, but here we will discuss the most commonly used: $L2$ and $L1$.

The mean squared error ($L2$) measures the average of the squares of the errors, that is, the average squared difference between the estimated values and the actual value. The standard form for L2 is:

$$L2 = \frac{1}{n} \sum_{i=1}^n (y^i - \hat{y}^i)^2$$

The mean absolute error ($L1$), similarly to L2, calculates the loss function as the sum of absolute differences between actual and predicted values. L1 does not have the normalising factor n (or $n - 1$). That is, the L1 loss is defined as:

$$L1 = \sum_{i=1}^n |y^i - \hat{y}^i|$$

While some parameters can change during a training session, there are some that cannot be changed by the algorithm and are decided initially by the developer of the network. Those parameters are called *hyperparameters*. For example, some hyperparameters are variables that determine the network structure and how the network is trained. By citing a few, we have the learning rate, the number of epochs for each training set, the regularisation parameter (aka which loss function), etc. etc.

With learning rate we intend how much of the error value has to be back propagated to the weights in the network in order to move in the direction of lower loss. Usually it looks like: $Weight_{new} = Weight_{current} - learningRate *$

gradient.

In order to train a neural network well and speed up the training process, there are several established strategies.

One of the most commonly implemented in neural networks with many layers is mini-batch. Mini-batch training means taking samples of a certain batch size to train the neural network during each iteration, instead of all available samples at the same time.

Batch normalisation is used to normalise the distribution of each layer's data to speed up the training speed. In batch normalisation, the input training data is processed in a batch, instead of for all the data. This slows down the training speed and thus requires lower learning rates and suitable parameter initialisation. This strategy is also useful to contrast *overfitting*.

An overfitting model is a statistical model that contains more parameters than can be justified by the data. The mechanism of overfitting is to unconsciously extract some residual changes (i.e., noise) as if that variation represented the underlying model structure. In contrast, under-fitting occurs when the statistical model does not adequately capture the underlying structure of the data. Then, when the model begins to 'remember' training data rather than 'learn', overfitting occurs when new data is processed. The direct consequence of overfitting is the poor performance of the network on the validation set.

See figure 3.3 for an illustration comparing an overfitted model to a fitted one.

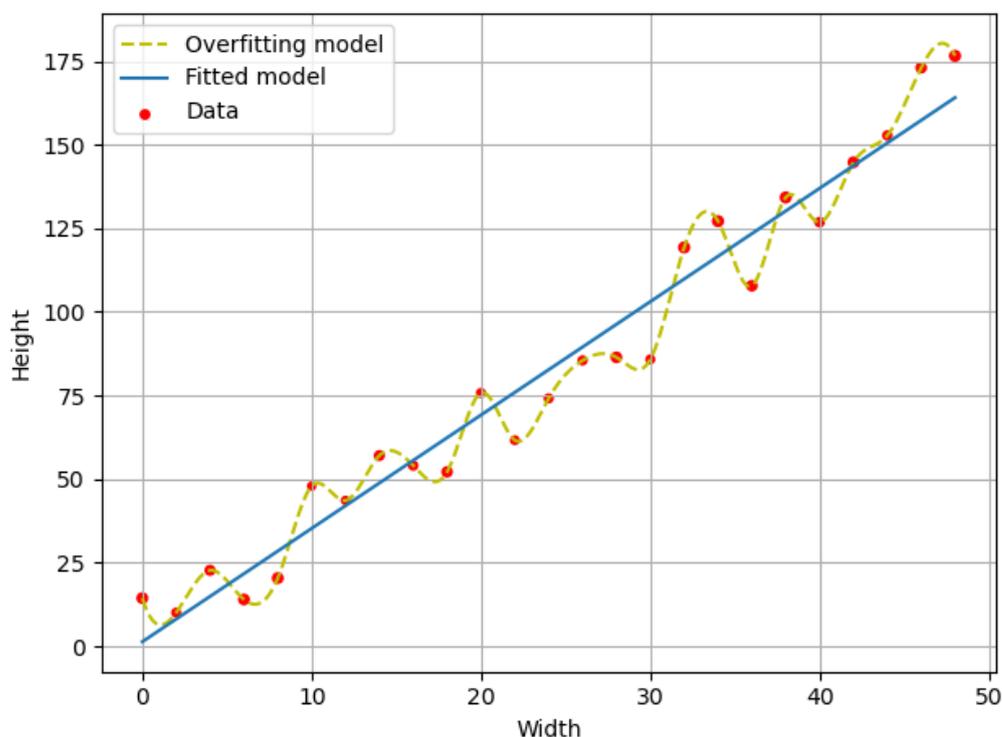


Figure 3.3: Noisy (roughly linear) data is fitted to a linear function and a polynomial function. Although the polynomial function is a perfect fit, the linear function is expected to perform better: if the two functions were used to extrapolate beyond the fitted data, the linear function should make better predictions.

3.1 Convolutional neural network

When we talked about how a neural network works, we focused primarily on the most commonly used net, the feedforward network, but there are various types of neural nets that are used for different use cases and data types.

For example, recurrent neural networks are commonly used for natural language processing and speech recognition, whereas *convolutional neural networks* (CNNs) are more often used for classification and computer vision tasks.

Convolutional neural networks were first discussed in a paper written by Dr.

David Hunter Hubel and Dr. Torsten Nils Wiesel in the 1960s, where they claimed that the visual cortex of cats and monkeys contains neurons that react individually to directional structures. Visual stimulus can affect the neighbourhood of a single neuron, known as the *receptive field*. Adjacent cells have similar and often overlapped receptive fields, the size and position of which vary, forming a complete visual spatial map.

This justified the use of local receptive fields in neural networks, and Hubel and Wiesel also proposed a cascading model for use in pattern recognition tasks. [18] Finally, in 1980, neocognition was proposed, marking the birth of the CNN, which introduced the concept of the receptive field in the artificial neural network. [19]

According to the concept of receptive fields, CNN exploits the spatial locality by enforcing a local connectivity between neurons of adjacent layers. This architecture ensures that the learned “filters” produce the strongest responses to spatially local input patterns of relevance. Stacking many such layers together forms nonlinear filters that become increasingly global as the depth goes deeper.

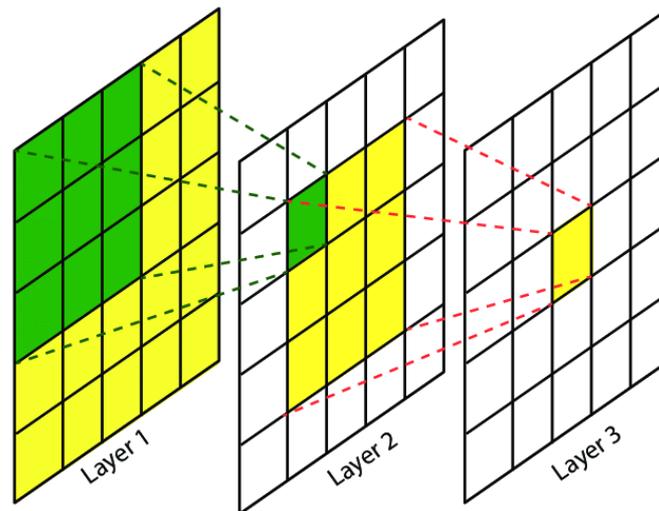


Figure 3.4: Convolution in a receptive field

In CNN, each filter is replicated across the entire visual field. These replicated units share the same parameters; that is, the same weight vector and bias are repeatedly used to produce a feature map. In a given convolutional layer, the features of interest for all neurons can be analysed by a shift-invariant correlation. Replicating units in this way allows for the same feature to be detected regardless of their position in the visual field.

Each CNN layer has neurons arranged in three dimensions: width, height, and depth. The width and height represent the size of a feature map. The depth represents the number of feature maps over the same receptive field, which offers different structural features in the same visual scope to respond to visual stimuli of various types, respectively. Finally, different types of layers, both locally and completely connected, are stacked to form the CNN architecture.

3.2 Learned Post Processing

Thanks to these discoveries, CNN has begun to achieve recognition in the field of reconstructive techniques as a potential successor to fully conventional iterative algorithms, but neural networks alone take little advantage of traditional algorithms that were developed through physical modeling and mathematical derivation, and there is no evidence, neither theoretically nor numerically, that neural network based algorithms solve the mathematical inverse problem modelling the tomographic reconstruction process.

This makes the network-based reconstruction results sub-optimal. So one possible solution proposed was to implement hybrids, to mix iterative reconstruction algorithms and deep learning techniques for improving performance whilst maintaining optimal results.

The current approach is called Learned Post Processing (LPP). First it produces a low quality image with artefacts and noise reconstructed with a fast method (typically a FBP) and then a neural network tries to improve the quality of the image by suppressing the artefacts. The network is then fed

as samples for comparison, a set of ground truth images reconstructed from full dose acquisitions. [20]

While LPP is extremely important as it pioneers a new approach to imaging iterative reconstruction, it is also being criticised for its lack of mathematical characterisation. [21]

One potential solution is the introduction of the *Rapid Iterative Solver with Iteration Network-based Gaining* (RISING) framework. [22] The RISING reconstructing procedure can be described by the two following steps executed in sequence:

- Starting from the projection data, a rapid iterative algorithm (IIR) produces a low quality reconstruction by solving the model-based problem of handling few-view CT with a few iterations. The execution of only a few iterations fits realistic time constraints.
- The previously computed rough reconstruction is processed by a deep neural network which aims at retrieving the unperformed iterations towards convergence. Its output is the RISING solution image.

Another widely implemented algorithm is the U-Net architecture. The U-Net is a generic deep learning solution for frequently occurring quantification tasks such as cell detection and shape measurements in biomedical image.

The U-Net is named to reflect its symmetric architecture and consists of a contracting part (left side) and an expansive part (right side).

The contracting part follows the typical architecture of a convolutional network. It repeatedly applies convolutions, each followed by the ReLU activation and a max pooling operation for down-sampling.

At each down-sampling step, U-Net doubles the number of feature channels. In the expansive part, each step is for up-convolution, which is known as transposed convolution. By the up-convolution, it finishes up-sampling feature maps and halves the number of feature channels. Then, U-Net concatenates these up-sampling feature maps with the correspondingly cropped feature maps from the contracting part of the U-Net.

Although the U-Net framework presents numerous advantages, like its practical adaptability, there are some disadvantages that limit its potential usage, like its deep structure that requires a very expensive training in terms of time and consumed energy.

One potential alternative is implementing a 'green' alternative architecture called ResUNet (residual-learning U-Net) that saves time and energy and also lowers computational time to reduce the cost of the hardware required to train the network.

The ResUNet is a fully convolutional neural network with a symmetric encoder decoder structure and pooling/unpooling operators to enlarge its receptive field. Each convolutional layer is composed of a Conv2D + Batch-Normalization + ReLU structure, as it is common in the literature, except for the last layer, where it was used as a tanh activation function.

Although this network is extremely important to understanding the evolution of neural networks, unfortunately, its complete explanation exceeds the scope of this paper, thus we refer the reader to "*A green prospective for learned post-processing in sparse-view tomographic reconstruction*"[\[23\]](#) for details of this.

3.3 Technical Notes

For our experiments, the implementation of a U-Net or ResUNet was too expensive, due to the sheer size of the phantom. As such, Colaboratory by Google was initially the platform to train our networks, but due to the constraints imposed on heavy-usage users, it was difficult to run the training samples without having the training abruptly stop halfway.

Due to this, it was then decided to run the training of the neural network on a machine generously given by the Department of *Scienze Informatiche*, which has a Geforce RTX A4000 with 16GB of memory.

The algorithm was written in Python 3.9 using the toolbox TensorFlow. Tensorflow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML-powered applications.

The samples on which the network was trained and then tested were not based on the Sheep-Logan phantom, which instead was the case for the comparison between iterative algorithms, due to its notoriety and its simplicity. Instead, it was decided to utilise an array of 120 phantoms, which were random objects generated containing ellipsoids, circles and other figures (see figure 3.5). Each phantom was generated as a 3D object with a size of 256x256x32.

While it was not possible to implement a complete U-Net, it was decided to implement a somewhat *partial* U-Net by implementing only a layer of convolution and up-convolution.

This left us with a greater degree of freedom compared to a normal U-Net and also helped to mitigate the enormous computing power usually required. Unfortunately, this choice came with a cost: by limiting the network's power, we also severed its ability to grow, making training till convergence impossible.

The network also implemented the *ADAM* optimiser, which is a stochastic

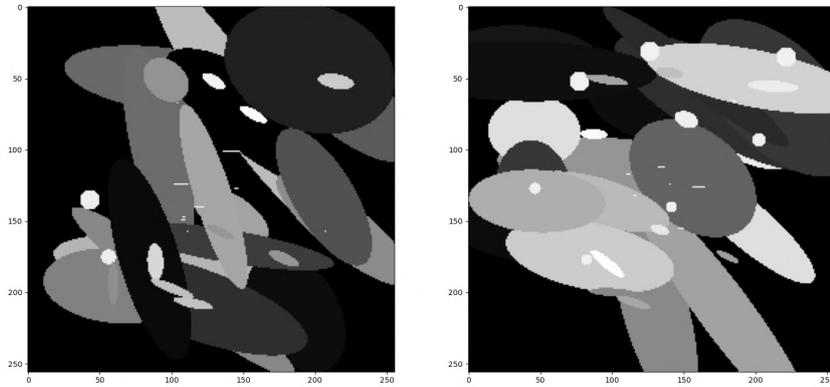


Figure 3.5: Two random phantoms generated for the network

gradient descent method that is based on adaptive estimation of first-order and second-order moments, to help improve the descent by manipulating the learning rate of the training. citekingma2017adam

The network was then trained three times, each time changing the size of the kernel to observe how the changes modified the behaviour of the network.

We define kernels as the stack of layers that are defined by the action of a number of filters on the input. For example, on a convolutional layer, the kernel size refers to the width \times height of the filter mask.

Layer Type	Output Shape
Conv3D	(256, 256, 32)
ReLU	(256, 256, 32)
BatchNormalisation	(256, 256, 32)
Conv3D [skip1]	(256, 256, 32)
ReLU	(256, 256, 32)
BatchNormalisation	(256, 256, 32)
MaxPooling3d	(128, 128, 16)
Conv3D	(128, 128, 16)
ReLU	(128, 128, 16)
BatchNormalisation	(128, 128, 16)
Conv3D	(128, 128, 16)
ReLU	(128, 128, 16)
BatchNormalisation	(128, 128, 16)
UpSampling3D	(256, 256, 32)
Concatenate(skip1)	(256, 256, 32)
Conv3D	(256, 256, 32)
ReLU	(256, 256, 32)
BatchNormalisation	(256, 256, 32)
Conv3D	(256, 256, 32)
ReLU	(256, 256, 32)
BatchNormalisation	(256, 256, 32)

Table 3.1: The 3D convolutional neural network’s structure

3.4 Analysis of the network

After training the neural network in each of its possible configurations for over 200+ epochs, we decided to test its ability with the taste samples. One of the most astounding feats that the network showcases is its ability to reduce the mean relative error (MRE) between the target result and the pro-

duced result. As we can see in Figure 3.6, by comparing the MRE's median of the sample not corrected by the network to the MRE's median produced by the neural network with kernel size 3, the network was able to reduce the mean relative error by almost half. Another surprise was its ability to reduce the number of outliers, as seen by the lengths of the *whiskers* in the first box plot compared to any other box plot.

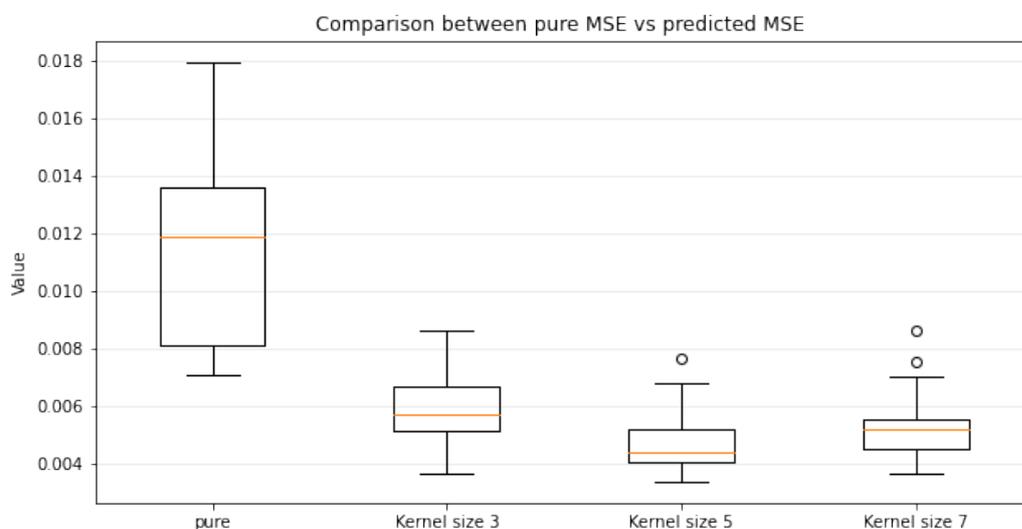


Figure 3.6: Median of mean relative errors

This illustrates that the network is quite able to identify potential errors present in an object and correct them, independently of the starter value, which was instead expected.

Another interesting situation occurs when looking at the kernel size. While the box plot for the neural network with kernel size 3 can be considered the largest amongst the three, it is also the only one that doesn't present any outliers outside of its whiskers (identified by the small dots above the box plots), when instead they appear when increasing the value of the kernel.

This, however, can be easily identified as being caused by the short duration of the network's training. By increasing the training duration, those outliers should be quite easily dealt with.

One tendency present in all of the reconstructed images is also its inability to recognise objects with similar shades next to each other, making it quite confusing to identify the borders.

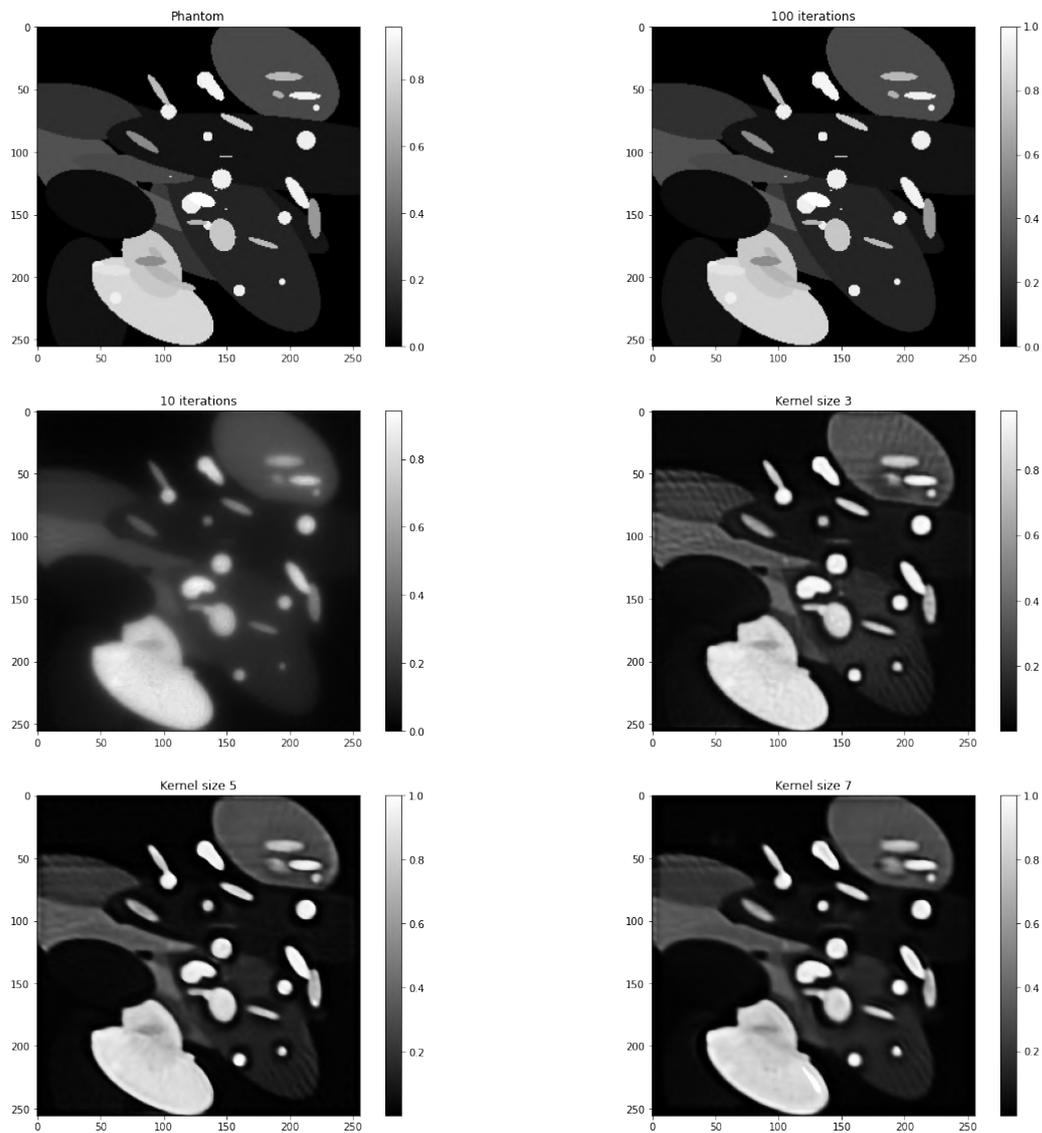


Figure 3.7: Comparison of objects between reconstructions

We then decided to look at the images reconstructed by the network and compare them against the ones obtained by only launching the iterative algorithm Sgp (see Figure 3.7 and 3.8).

When looking at the images obtained, we can clearly see that there are some issues that the network was unable to get rid of and also some that were even slightly worsened. The most noticeable issues are the streaks that appear in every reconstruction but seem more visible when the kernel size is equal to 3, and although there are still some present for the other two reconstructions, their visibility decreases noticeably.

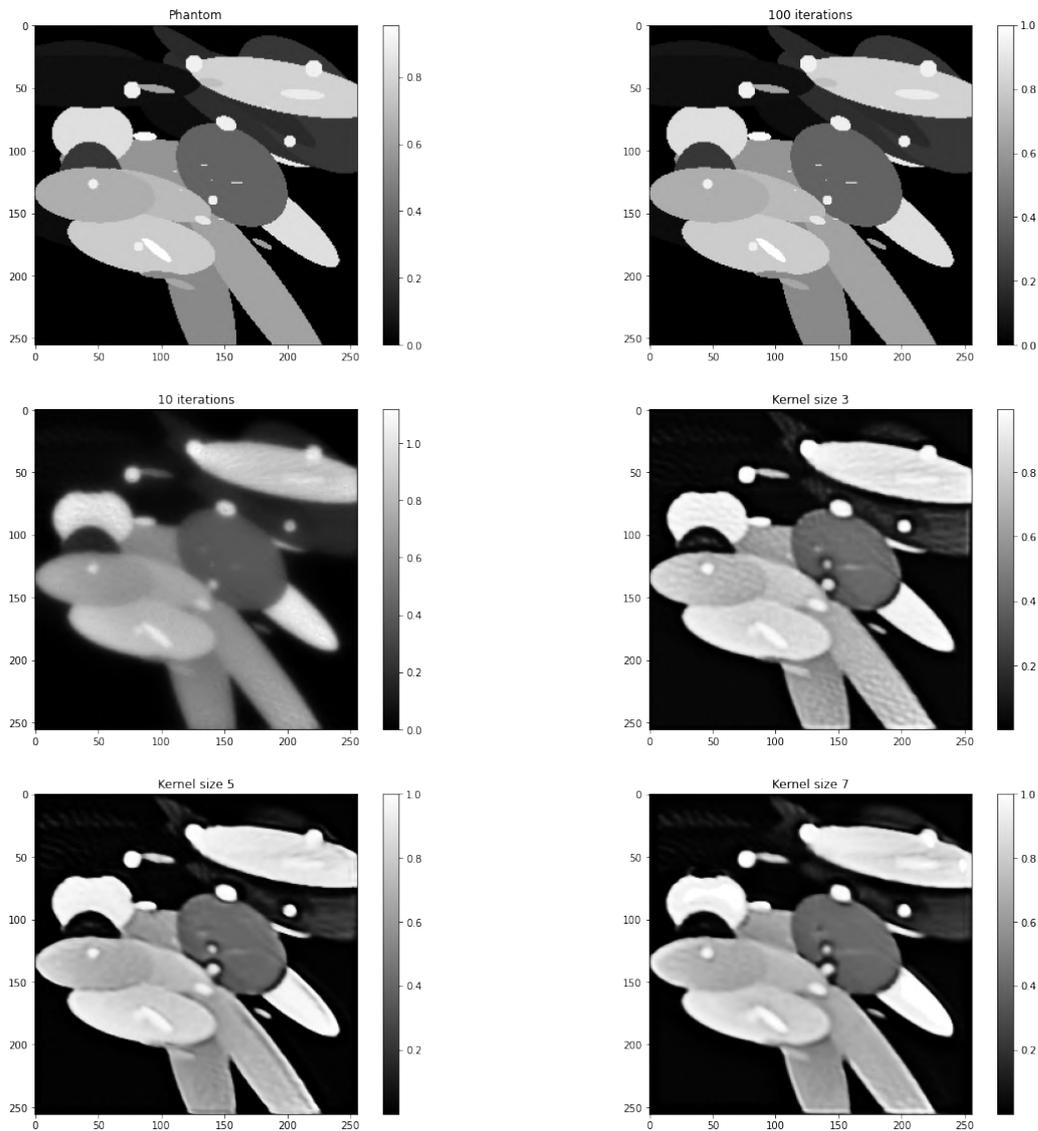


Figure 3.8: Second object comparison between reconstructions

Instead, when looking at the images obtained by the network with kernel size 7, we can see that there are still some streaks, but the most worrying issue that was encountered was its tendency to increase the size of particular points inside the phantoms. This could be cause for concern because, in a medical setting, the reconstructed image must be extremely close to the original to not cause some form of bad judgement.

When looking at the object from a side perspective view (see Figure 3.9), we can see that for this particular case, the Sgp algorithm after 10 iterations produced a beam originating from the object in a position of approximately (160, 15)

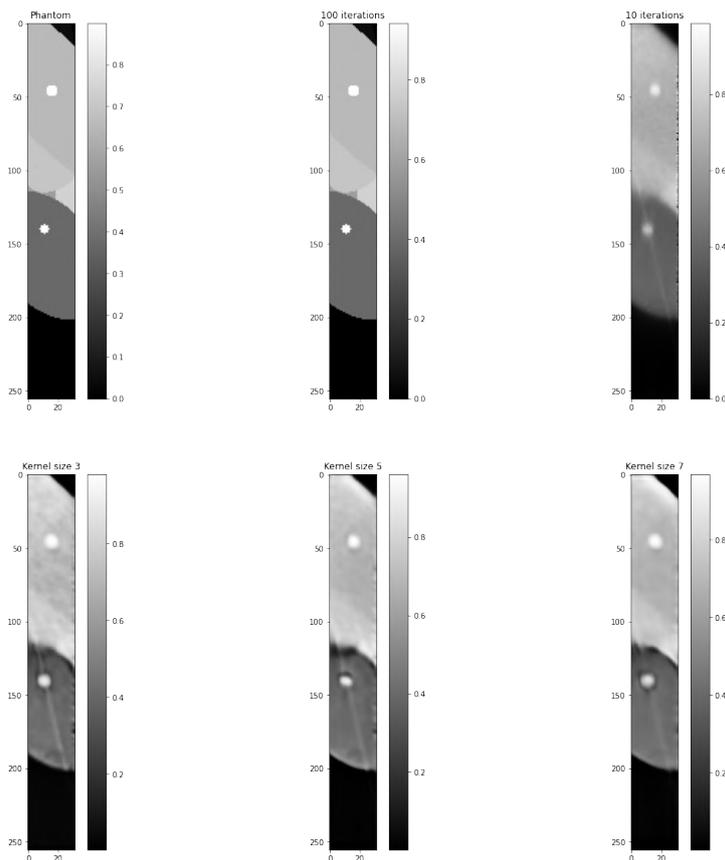


Figure 3.9: Side view of the first comparison of objects between reconstructions

This beam can be seen not present in the reconstruction after 100 iterations, but it appears in the images reconstructed by the network, albeit smaller for each increase in kernel size. This means that the network was able to identify it correctly as an error, but it wasn't able to correctly remove it.

One noteworthy mention we can add to this side perspective is the almost absent presence of streaks, signifying that the network is working correctly but requires more time, more samples, and possibly a more advanced form to continue to grow.

Finally, looking at the result, we can conclude that, although the reconstructed images weren't quite excellent, the network performed splendidly with the given resources.

Some changes could be made to improve its growth and reliability, like converting the network from a partial U-Net to a RedCNN [24] and also increasing the sample size to let it train with more unknown variables.

Conclusions

In this thesis, we showcased a possible approach to the implementation of a neural network for tomographic imaging techniques using *Convolutional neural network*.

Initially, it was shown how current CT scanners work and how the medical setting is undergoing a gradual change from full-view scanners to sparse-view ones, and how a sparse-view scanner can be implemented. It was then explained how modern imaging algorithms work and how they operate by solving the reconstruction problem of an object scanned, which can be classified mathematically as an inverse problem.

In the second chapter, we introduced the most commonly used algorithms nowadays, starting with the oldest and most basic ones, such as filtered back-projection and gradient descent, to analyse newly created iterative techniques, such as the Fast iterative shrinkage-thresholding algorithms (Fista) or the scaled projection algorithm (SGP).

They were then compared to observe their performance applied to a simulated real-world setting. The parameters compared were primarily relative errors and the quality of the reconstructed images. Each algorithm was launched two times, one to end after 100 iterations or if it converged, while the other was *early stopped* to simulate a real-world medical setting where it's unacceptable to wait for a long period of time for a reconstructed image and instead the image is required minutes after the scan.

After having compared the algorithms, we started to design a neural network with the intention of feeding it the images produced after 10 iterations as a

source and the images after 100 iterations as targets, with the objective of creating a network capable of creating images similar to the targets and thus removing over 90 iterations of the algorithm, increasing its efficiency and speed.

Various architectures were tested, and then it was decided to implement a partial U-Net due to its simplicity and quickness in the training phase. It was trained multiple times, every time changing the hyperparameters by a notch to observe potential differences and then implement only the ones with the best results obtained.

After testing, we can conclude that the networks created are all stable thanks to obtaining similar results even when the images were extremely different from each other. Even in the outlier's case, the error was reduced compared to the source image by quite a margin. Unfortunately, artefacts are still present inside the images and, in some cases, even worsened, amplifying streaks and increasing the size of objects, potentially making them a hazard in a possible medical setting.

Future development is still possible and welcomed. Current algorithms are primarily based on the reconstruction of two-dimensional layers, while three-dimensional reconstructions are quite young in comparison. The network described in this thesis is still young and potentially extremely accurate. Potential changes could integrate existing networks to form a better imaging reconstruction architecture.

Bibliography

Book Sources

- [5] Jennifer L Mueller and Samuli Siltanen. *Linear and nonlinear inverse problems with practical applications*. SIAM, 2012.
- [13] Curtis R Vogel. *Computational methods for inverse problems*. SIAM, 2002.

Other Sources

- [1] Rita F Redberg, Rebecca Smith-Bindman, et al. “We are giving ourselves cancer”. In: *New York Times* 30.1 (2014).
- [2] Rebecca Smith-Bindman et al. “Radiation dose associated with common computed tomography examinations and the associated lifetime attributable risk of cancer”. In: *Archives of internal medicine* 169.22 (2009), pp. 2078–2086.
- [3] Edward L Nickoloff and Philip O Alderson. “Radiation exposures to patients from CT: reality, public perception, and policy”. In: *American Journal of Roentgenology* 177.2 (2001), pp. 285–287.
- [4] Takeshi Kubo et al. “Radiation dose reduction in chest CT: a review”. In: *American journal of roentgenology* 190.2 (2008), pp. 335–343.

-
- [6] Jacques Hadamard. “Sur les problèmes aux dérivées partielles et leur signification physique”. In: *Princeton university bulletin* (1902), pp. 49–52.
- [7] Jakob S Jørgensen et al. “Core imaging library-part I: a versatile Python framework for tomographic imaging”. In: *Philosophical Transactions of the Royal Society A* 379.2204 (2021), p. 20200192.
- [8] Wim Van Aarle et al. “The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography”. In: *Ultramicroscopy* 157 (2015), pp. 35–47.
- [9] Emmanuel Candes. “Applied fourier analysis and elements of modern signal processing”. In: *Lecture 10, Stanford University* (2016).
- [10] Christian G Graff and Emil Y Sidky. “Compressive sensing in medical imaging”. In: *Applied optics* 54.8 (2015), pp. C23–C44.
- [11] Marcel Beister, Daniel Kolditz, and Willi A Kalender. “Iterative reconstruction methods in X-ray CT”. In: *Physica medica* 28.2 (2012), pp. 94–108.
- [12] Silvia Bonettini et al. “Scaling techniques for gradient projection-type methods in astronomical image deblurring”. In: *International Journal of Computer Mathematics* 90.1 (2013), pp. 9–29.
- [14] Leonid I Rudin, Stanley Osher, and Emad Fatemi. “Nonlinear total variation based noise removal algorithms”. In: *Physica D: nonlinear phenomena* 60.1-4 (1992), pp. 259–268.
- [15] Amir Beck and Marc Teboulle. “A fast iterative shrinkage-thresholding algorithm for linear inverse problems”. In: *SIAM journal on imaging sciences* 2.1 (2009), pp. 183–202.
- [16] E Loli Piccolomini et al. “Reconstruction of 3D X-ray CT images from reduced sampling by a scaled gradient projection algorithm”. In: *Computational Optimization and Applications* 71.1 (2018), pp. 171–191.
- [17] Lawrence A Shepp and Benjamin F Logan. “The Fourier reconstruction of a head section”. In: *IEEE Transactions on nuclear science* 21.3 (1974), pp. 21–43.

-
- [18] David H Hubel and Torsten N Wiesel. “Receptive fields and functional architecture of monkey striate cortex”. In: *The Journal of physiology* 195.1 (1968), pp. 215–243.
- [19] Kuniyuki Fukushima, Sei Miyake, and Takayuki Ito. “Neocognitron: A neural network model for a mechanism of visual pattern recognition”. In: *IEEE transactions on systems, man, and cybernetics* 5 (1983), pp. 826–834.
- [20] Yo Seob Han, Jaejun Yoo, and Jong Chul Ye. “Deep residual learning for compressed sensing CT reconstruction via persistent homology analysis”. In: *arXiv preprint arXiv:1611.06391* (2016).
- [21] Emil Y Sidky et al. “Do CNNs solve the CT inverse problem?” In: *IEEE Transactions on Biomedical Engineering* 68.6 (2020), pp. 1799–1810.
- [22] Davide Evangelista, Elena Morotti, and Elena Loli Piccolomini. “RISING a new framework for few-view tomographic image reconstruction with deep learning”. In: *arXiv preprint arXiv:2201.09777* (2022).
- [23] Elena Morotti, Davide Evangelista, and Elena Loli Piccolomini. “A green prospective for learned post-processing in sparse-view tomographic reconstruction”. In: *Journal of Imaging* 7.8 (2021), p. 139.
- [24] Hu Chen et al. “Low-dose CT with a residual encoder-decoder convolutional neural network”. In: *IEEE transactions on medical imaging* 36.12 (2017), pp. 2524–2535.

