Alternating Direction Method of Multipliers and Machine Learning for Computed-Tomography

Dominic Afuwape

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Master of Science

of

University College London.

Department of Physics and Astronomy
University College London
MSc Scientific Computing

May 7, 2020

I, Dominic Afuwape, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Abstract

High quality reconstructions of image projections is highly sought after within the field of medical imaging. Medical image reconstruction poses an ill-posed problem and therefore typical reconstruction schemes rely on iterative optimisation with additional regularisation. One such optimisation scheme, which is the method explored in this report, is alternating method of multipliers. This report looks specifically at the problem of Computed Tomography reconstruction. In this report a total variation ADMM scheme is implemented which solves linear system of equations iteratively to minimise. The performance of the method is analysed and the report examines suggested tolerances, and various iterative solvers are also compared for performance. Another development within the field of medical imaging are schemes which use neural networks to learn part or all of the optimisation scheme. This is highly useful as it combines traditional model-driven approaches with data-driven approaches. Therefore, in this paper two new fully learned optimization schemes based on ADMM are proposed which build upon current state of the art learned schemes. The report finds that these methods are highly performant with ”memory-less” learned ADMM methods surpassing the performance of the state of the art ”memory-less” Learned Primal dual methods while needing to learn fewer parameters with improvements of up 7dB peak signal-to-noise ratio. This report also contains a learned ADMM with ”memory” that matches the performance of current Learned Primal Dual methods”

Acknowledgements

I would like to thank Dr. Marta Betcke and Francesc Rullan for amazing support supervising my project. I would also like to thank the department of Physics and Astronomy for providing a great experience. Code for this project is available here: https://github.com/DominicUCL/ADMM-for-Computed-Tomography

Contents

Chapter 1

Introduction to CT problems

The techniques of medical imagery can be used to gain visualisation of the internal structures and geometries of the body. It is a wide-ranging field with a number of different technologies, methods and applications. First introduced in the 1970s, one widely used technology for medical imagery (though also with industrial usage) is the CT scan (Computed-Tomography) whereby, typically through the use of X-rays, non-destructive imaging can be performed. Specifically, within the field of medical imaging, CT scans can be used for diagnosis through examining material composition, abnormal geometries and density variation [1]. Examples of diagnoses that could be achieved thought the use of CT scans includes detecting cancers,lung disease, blood clots and infections among others [2]. The basic method of a CT scan is to use a coupled X-ray source and X-ray detector constrained on a rotary table. As the X-ray source and detector processes around the body at various angle positions it acquires cross sectional image projections. At each angle, the X-rays will undergo interactions (i.e attenuation, scattering, passing through) and the measurements relate to the internal and external properties of the observed object which are then used in reconstruction of an image.

While this is a powerful technique, a few issues arise. Due to the dependence on exposing the examined organic bodies to potentially harmful radiation, it is important to find ways to minimise this dose. There exists a trade-off between radiation exposure and detail/accuracy of the reconstruction as for example more accurate reconstructions may require increased measurements of the object in ques-

Figure 1.1: Patient entering C-T Scanner

tion. The Filtered Back Projection is the traditional method for reconstruction of the image from the projections and is suitable for problems with full data, however in problems with limited data, the use of variational regularisation is often required to compensate for the missing data. This has led to the application of techniques such as Total-Variation denoising, for example, that exploit knowledge of priors and apply this knowledge to optimisation to achieve better quality reconstruction with limited projections. Alternatively the application of U-Net and ResFCN neural networks to improving CT scans provided high quality results is also popular. Overall, the approaches to improving the quality of CT scans are wide ranging and deserving of much analysis.

Chapter 2

Introduction to inverse problems

The problem of reconstructing CT scans using the acquired projections comes under a wider branch of problems termed ”Inverse problems”. The solution to an inverse problem results in computing an unknown quantity based on indirect measurements with causal factors that are linked to the unknown quantity itself. CT reconstruction can to stated to be ill-posed, as with most cases of inverse problems. A well-posed problem can be defined as:

1. a solution to the problem exists in a given set of “admissible” solutions (existence condition),

2. the solution is unique (uniqueness condition),

3. the solution depends continuously on input data (stability condition). [3]

In a more mathematical sense, the inverse problem is the solution for f to the following equation (2.1)

$$\mathcal{K}(f) = g \tag{2.1}$$

Where f is the unknown quantity and g is the data which can be acquired directly through measurement. Here, K : X $\rightarrow$ Y denotes an operator mapping (forward operator) from the X to the Y. The *well - posedness* of a problem can now be interpreted as:

1. For all input data there exists a solution to the problem, i.e. for all $g \in Y$ there exists an $f \in X$ with $\mathcal{K}(f)$ = g .

2. For all input data this solution is unique, i.e. $f \neq v$ implies $\mathcal{K}(v) \neq g$.

3. The solution of the problem depends continuously on the input datum, i.e. for all $\{f_k\}_{k \in N}$ with $\mathcal{K}(f_k) \to f$ we have $f_k \to f$. [4]

While in theory the inverse of the forward operator may exist leading to a solution

$$f = \mathcal{K}^{-1}(g)$$

The problem becomes more complicated when we add a noise or error term δg especially in CT problems where the popular Filtered Back Projection method enhances image noise.

$$\mathcal{K}(f) + \delta g = g$$

at which point methods that can implement prior knowledge of the uncorrupted image show their strength. Certainly in any case, by these definitions, the CT scan problem is not well posed.

It is of interest to understand the physical interpretation of all the terms in the optimisation problem in particular the operator $\mathcal{K}$ which is applied to the image/ slice f.

2.1 Radon Transform

The Radon transform is a type of integral transform used in Computed-Tomography reconstruction and when in the 2-dimensional space. Referencing back to the introduction and operator $\mathcal{K}$, for the CT problem we select this operator to be the Radon transform R. The Radon transform can be thought of as the forward projection of the unknown image. The Radon transform of an image written as a function $f(x, y)$ when looking at 2-dimensional images is defined as the line integral of f along a line L inclined at an angle θ with distance s from the origin.

$$Rf(\theta, s) = \int_L f(x, y)du$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y)\delta(xcos\theta + ysin\theta - s)\,dxdy$$

where $\delta(x)$ is the Dirac delta distribution [5]

Specifically in this project, the parallel beam geometry is used meaning that, for any given projection, the angle of the beams becomes fixed and the distances from the origin of the parallel beams are varied meaning s would be a set of m distances from the origin.

In real cases, the transform will be discretized not continuous, therefore it does not capture the full information about the image within these forward projections.

2.2 Back-projection

If we were to measure using continuous data Fourier transforms of the various parallel projections over a full π interval, the measured object can be completely reconstructed in the Fourier space. [6]

$$\mathcal{F}_t Rf(\theta, s) = \mathcal{F}_2 f(-psin\theta, pcos\theta)$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \mathcal{F}_2 f(p_x, p_y)\mathrm{e}^{j2\pi(xp_x + yp_y)} dp_x dp_y$$

Then, by changing the integration variables from $dp_x dp_y$ to $|p|dpd\theta$ and using further simplification, the equation for the original image in terms of the forward projection data $Rf(\theta, s)$ can be shown as

$$f(x,y) = \frac{1}{2}\int_{0}^{2\pi} (Rf * g)(\theta, ycos\theta - xsin\theta)$$

where

$$g(t) = \frac{1}{2}\int_{-\infty}^{\infty} |p|\mathrm{e}^{j2\pi pt} dp$$

can be viewed as a ramp filter hence the term **filtered** back-projection[6]. A ramp filter emphasises high frequency components which can amplify unwanted noise, motivating the need for schemes that rectify this.

2.3 Variational Regularisation

The reconstruction problem can be solved by a model driven approach which minimises a loss function (2.2) shown as

$$\min_{f \in X} \mathcal{L}(\mathcal{K}(f), g) \tag{2.2}$$

This however is not suitable for ill-posed problems such as CT reconstruction as an instablity exists meaning that a small error induced in measurement $\mathcal{K}(f)$ leads to a large change in the reconstructed uncertain quantity. A suitable development on this is to then introduce a regularisation functional $S : X \to R$ that encodes a priori knowledge that is known about f_{true} through penalties. The knowledge that this functional can encode is widely ranging, popular functional include L-p norms which encode knowledge about the expected magnitudes of the reconstruction and similarly total variation which encodes knowledge about the expected magnitudes of the gradients of f_{true}. Now the full optimisation problem can be written as

$$\min_{f \in X} \mathcal{L}(\mathcal{K}(f), g) + \lambda S(f) \tag{2.3}$$

where λ is some regularisation parameter that governs the amount of penalisation the regularisation functional applies. The optimal regularisation functional varies from problem to problem and much work has been done to produce performant regularisers. There exists alternative approaches to solving this problem. The problem (2.3) could also be formulated as a denoising problem with a maximum a posteriori (MAP) estimation.

$$\operatorname*{argmax}_{f} p(f \mid g)$$

$$\operatorname*{argmin}_{f} -log\, p(g \mid f) - log\, p(f)$$

If we note that $-log\, p(f)$ represents the prior distribution of f, an equivalence can be made with the variational regularisation problem's regularisation functional $-log\, p(f) = \lambda S(f)$ which has lead to the use of sequential applications of denoising steps in ”Plug and Play” type scheme optimization schemes which use readily available high performance denoisers [7] and later on this was expanded upon with ”Regularisation by Denoising ” where for certain denoisers with symmetric jacobians, the regularisation function $\lambda S(f)$ could be explicitly written in terms of a denoiser functional $D(f)$ as

$$S(f) = \frac{1}{2} f^T (f - D(f))$$

[8] . While the above regularisation is only valid for denoisers with symmetric jacobians further research has gone on to demonstrate how a more general class of denoisers can be used in optimization schemes interpreted as ”score-matching” between f and $D(f)$ [9]. The type of denoiser used can take on many forms even the use of the previously mentioned UNet and other **learned denoisers** which forms the basis for learned regularisation schemes.

Chapter 3

Solving the optimisation problem

Having outlined the optimisation problems and examined current developments, it helps to understand how an optimisation problem might be solved. Take an equation

$$\min_{x} r(x) + h(Kx)$$

In order to solve this problem, it is common to resort to variable splitting methods. To do this, r and h can be minimised separately with an auxiliary variable z for the function h with a constraint that.

$$z = Kx$$

this then becomes a constrained optimisation problem with the **Lagrangian** formulation

$$\mathcal{L}(x, z, \lambda) = r(x) + h(z) + \langle \lambda, Kx - z \rangle$$

where λ is called the **Lagrange multiplier**. The problem finds the minimum $w.r.t$ x and z while maximising w.r.t λ (which is is $\lambda = +\infty$). From the Lagrangian, we find the **Augmented Lagrangian** .

$$\mathcal{L}_{\rho}(x, z, \lambda) = r(x) + h(z) + \langle \lambda, Kx - z \rangle + \frac{\rho}{2} \|Kx - z\|_2^2$$

where it can be seen that another penalty has been applied.

In the **ADMM** (Alternating Direction Method of Multipliers) method, the functionals r and h are minimised $w.r.t$ to the variables x and z which are updated sequentially in the following steps.

$$\begin{aligned} x_{i+1} &= \operatorname*{argmin}_x r(x) + \frac{\sigma}{2}\|Kx - z_i + \frac{\lambda_i}{\sigma}\|_2^2 \\ z_{i+1} &= \operatorname*{argmin}_z h(z) + \frac{\tau}{2}\|Kx_{i+1} - z + \frac{\lambda_i}{\tau}\|_2^2 \\ \lambda_{i+1} &= \lambda_i + Kx_{i+1} + z_{i+1} \end{aligned} \tag{3.1}$$

ADMM has a number of useful properties, for example, each step does not require an exact solution and it is possible to adapt this method to have separate constraints on each variable. The CT optimization problem requires specific choices on the functions r and h and the operator K. Commonly, h has been chosen to be a L2 data fidelity term and r is often a total variation norm or any other suitable regulariser, and K is the Radon transform.

3.1 Non-Learned Solvers

The following section lays out two popular methods of solving the CT reconstruction problem with both being forms of the ADMM method previously mentioned. In this section the following notation is used, $\mathcal{F}(f) = ||\mathcal{K}(f) - g||_2^2$ and $\mathcal{G}(f) = ||\nabla f||_1$.

3.1.1 Linearised ADMM

When attempting to minimise through the ADMM method it helps now to define the **proxmial operator**.

$$prox(x)_{\rho f} = \underset{z}{\operatorname{argmin}}\, f(z) + \frac{\rho}{2}||x - z||_2^2$$

There is a clear connection between the ADMM method (3.1) and the proximal operator as the z step can be put in proximal form by stating the step as

$$z_{i+1} = prox_{\tau\mathcal{F}}(Kx_{i+1} + \frac{\lambda_i}{\tau})$$

and it may be possible to write the first step in a proximal form (in cases where the constraint $Kx = z$ is chosen).

The approach used in the Linearised ADMM to write the x update in proximal form is to use an approximation around $\mathcal{G}$ for the first update step

$$x_{i+1} = \underset{x}{\operatorname{argmin}}\, \mathcal{G}(x) + \frac{1}{2\sigma}||x - (x_i - \frac{\sigma}{\tau}K^T(Kx_i - z_i + \lambda_i))||_2^2$$

This makes the solution for x_{i+1} a proximal mapping of $\mathcal{G}$

$$x_{i+1} = prox_{\sigma\mathcal{G}}(x_i - \frac{\sigma}{\tau}K^T(Kx_i - z_i + \lambda_i))$$

Both the x and z update steps are sub-problems. The x subproblem can be solved by Nesterov's fast gradient projection (FGP) method for total variation noising.

The z update step can be solved by substituting $Kx_{i+1} + \frac{\lambda_i}{\tau}$ with f and instead solving the proximal operator of $||f - g||_2^2$ for penalty parameter σ which has the elementwise solution

$$prox_{\sigma\|f-g\|_2^2} = \frac{f + 2\sigma g}{1 + 2\sigma}$$

which is also how the solution is implemented in the ODL library that will be mentioned later.

Algorithm 1 Linearised ADMM

```
Given:$x_0 \in \mathcal{X}, \lambda_0 \in \mathcal{U}, z_0 \in \mathcal{U}$
for $i = 1, ...., I$ do
    $x_{i+1} \leftarrow prox_{\sigma\mathcal{G}}(x_i - \frac{\sigma}{\tau}K^T(Kx_i - z_i + \lambda_i))$
    $z_{i+1} \leftarrow prox_{\tau\mathcal{F}}(Kx_{i+1} + \lambda_i)$
    $\lambda_{i+1} \leftarrow \lambda_i + Kx_{i+1} - z_{i+1}$
```

3.1.2 System Solve ADMM

When the constraint $x = z$ is chosen instead, the step for calculating x_{i+1} corresponds exactly to the x_{i+1} step in the traditional ADMM Lasso problem [10]. The proximal for the least-sqaures function $\mathcal{F}$ has a closed-form solution.

$$x_{i+1} = (K^T K + \rho I)^{-1}(K^T g + \rho(z_k - \frac{\lambda_k}{\rho}))$$

For large problems such as this Computed Tomography problem it is clearly not advisable to compute exactly the inverse $(K^T K + \rho I)^{-1}$. This can then be solved with an equation solver for example an iterative solver like LSQR or GMRES leading to an ADMM solution with an outer loop and an inner loop for the iterative solver of the fidelity function. Important aspects to know about this problem is that the condition number of $(K^T K + \rho I)$ has dependency on the number of projections along with the choice of value ρ which has implications for how easy it is to solve the system of equations.

Algorithm 2 System Solve

```
Given:$f_0 \in \mathcal{X}, h_o \in \mathcal{U}$
for $i = 1, ...., I$ do
    $x_{i+1} \leftarrow (K^T K + \rho I)^{-1}(K^T g + \rho(z_k - \frac{\lambda_k}{\rho}))$
    $z_{i+1} \leftarrow prox_{\tau\mathcal{F}}(x_{i+1} + \lambda_i)$
    $\lambda_{i+1} \leftarrow \lambda_i + x_{i+1} - z_{i+1}$
```

Chapter 4

Learned Optimisation

While the previously shown TV regularisation method does produce satisfactory results for certain applications an alternative would be to consider the idea of a **learned optimisation**. Research has shown that one can choose to learn iterative reconstruction schemes where proximal operators can be replaced by other operators which are not proximal operators. Typically, the proximal operator for the 'regularisation functional' is replaced as this encodes the a priori information and furthermore, typically the data fidelity term proximal has a closed form solution which provides good results if the forward operator is a good estimate[11]. By replacing the regularisation proximal with a neural network for example, using large datasets, a priori information is encoded into the scheme without the need to specify the regularisation functional beforehand. Another direction researchers who wish to apply iterative schemes have taken is to learn not just the regularisation component but the full iterative reconstruction scheme. The learned optimisation schemes developed in this thesis are motivated by and follow a similar scheme to work produced on Learned Primal-dual Reconstruction[12]. Therefore, the Learned Primal Dual Hybrid Gradient and Learned Primal Dual algorithms will be laid out here for context. These algorithms learn **two** proximal operators (in the data space and reconstruction space) as in the fashion of a full learned iterative reconstruction scheme. It is useful to outline the original non-learned Primal Dual Hybrid Gradient algorithm which these two learn methods are based on.

4.1 Primal Dual Hybrid Gradient

The Primal Dual Hybrid Gradient method is a splitting method for constrained optimisation also known as the Chambolle-Pock algorithm. Note that here $\mathcal{K}$ is the forward operator and $\partial\mathcal{K}$ is the adjoint of the forward operator with image f.

Algorithm 3 Primal Dual Hybrid Gradient

```
Given:$\sigma, \tau > 0 \, s.t \, \sigma\tau\|\mathcal{K}\|^2 < 1, \gamma \in [0,1] \, and \, f_0 \in \mathcal{X}, h_o \in \mathcal{U}$
for $i = 1, ....$ do
    $h_{i+1} \leftarrow prox_{\sigma\mathcal{F}^*}(h_i + \sigma\mathcal{K}(\bar{f}_i))$
    $f_{i+1} \leftarrow prox_{\tau\mathcal{G}}(f_i - \tau[\partial\mathcal{K}(f_i)] * (h_{i+1}))$
    $\bar{f}_{i+1} \leftarrow f_{i+1} + \gamma(f_{i+1} - f_i)$
```

4.2 Learned Primal Dual Hybrid Gradient

Algorithm 4 Learned Primal Dual Hybrid Gradient

```
Given:$f_0 \in \mathcal{X}, h_o \in \mathcal{U}$
for $i = 1, ...., I$ do
    $h_{i+1} \leftarrow \Gamma_{\theta^d}(h_i + \sigma\mathcal{K}(\bar{f}_i))$
    $f_{i+1} \leftarrow \Lambda_{\theta^p}(f_i - \tau[\partial\mathcal{K}(f_i)] * (h_{i+1}))$
    $\bar{f}_{i+1} \leftarrow f_{i+1} + \theta(f_{i+1} - f_i)$
```

With this algorithm, traditionally in a non learned environment various step-size/penalty parameters would need to be selected, however due to this being a learned method, these can be left as variables to be determined by the network. The parameters of the algorithm are the step lengths σ , τ, the over-relaxation parameter, θ and the general parameters of the primal proximal network θ^p and the dual proximal network θ^d.

4.3 Learned Primal Dual

Algorithm 5 Learned Primal Dual

```
Given:$f_0 \in \mathcal{X}, h_o \in \mathcal{U}$
for $i = 1, ...., I$ do
    $\mathrm{h}_{i+1} \leftarrow \Gamma_{\theta_i^d}(h_i, \mathcal{K}(\bar{f}^{(2)}_{i-1}), g)$
    $f_{i+1} \leftarrow \Lambda_{\theta_i^p}(f_i, [\partial\mathcal{K}(f^{(1)}_{i-1})] * (h^{(1)}_{i+1}))$
```

The Learned Primal Dual method expands on the PDHG method in number of ways.

1. First the network expands the primal space to add memory between iterations

$$f = [f^{(1)}, f^{(2)}, ..., f^{(N_{primal})}] \in X^{N_{primal}}$$

2. and similarly it extends the dual space U to $U^{N_{dual}}$.

3. The network allows the network to choose how to combine the dual variable h_i and the operator transformed primal variable $\mathcal{K}(\bar{f}_i)$ for the first proximal

4. The network learns how to combine f_{i-1} and $\partial\mathcal{K}(f^{(2)}_{i-1})$

5. The last change is to allow the learned proximal operators to vary at each iteration $i = 1, ..., I$

4.3.1 Learned ADMM

The learned ADMM method which has been developed for this thesis has many similarities to the Learned PDHG. It takes inspiration from the Linearised - ADMM. Like PDHG, there are two learned proximal operator both in the image and sinogram space however this method adapts the way the dual variable (u in this case) is updated in the network. Similarly to the Learned Primal Dual Hybrid Gradient, parameters σ and τ are learned along with the general parameters of the primal proximal network θ^p and the dual proximal network θ^d.

Algorithm 6 Learned ADMM

```
Given:$x_0 \in \mathcal{X}, \lambda_o \in \mathcal{U}$
for $i = 1, ...., I$ do
    $x_{i+1} \leftarrow \Gamma_{\theta_i^1}(x_i - \tau K^T(Kx_i - z_i + \frac{\lambda_i}{\tau}))$
    $z_{i+1} \leftarrow \Lambda_{\theta_i^2}(\sigma K x_{i+1} + \frac{\lambda_i}{\sigma}, g)$
    $\lambda_{i+1} \leftarrow \lambda_i + \gamma(Kx_{i+1} - z_{i+1})$
```

4.3.2 Learned ADMM +

Following the essence of the Learned Primal Dual method, compared to the Learned Primal Dual Hybrid Gradient method, the Learned ADMM method is expanded by adding memory channels for the two primal variables x and z . The network is also allowed to choose how to combine x_i and $\frac{\sigma}{\tau}K^*(Kx_i - z_i + u_i))$ and how to combine Kx_i and z_i. Finally, the scheme is also allowed to learn different proximal operators for each iteration.

Algorithm 7 Learned ADMM +

```
Given: $f_0 \in \mathcal{X}, h_o \in \mathcal{U}$
for $i = 1, ...., I$ do
    $x_{i+1} \leftarrow \Gamma_{\theta_i^1}(x_i, \frac{\sigma}{\tau}K^T(Kx_i - z_i, \lambda_i))$
    $z_{i+1} \leftarrow \Lambda_{\theta_i^2}(Kx_{i+1} + \lambda_i, g)$
    $\lambda_{i+1} \leftarrow \lambda_i + Kx_{i+1} - z_{i+1}$
```

There is in fact a clear relationship between Linearised ADMM and Primal Dual Hybrid Gradient. With certain preconditioners, the Linearised ADMM is equivalent to Primal Dual Hybrid Gradient applied to the dual[13] however this equivalence does not translate to the basic learned schemes (Learned ADMM and Learned PDHG) as in the Learned ADMM scheme the update of the dual λ is kept as a separate step and the dual is also strictly divided by the penalty parameters when used in the input for the x and z updates. However, as the extended learned methods are allowed to learn how to combine the variables the two methods become incredibly similar which was something that was revealed during the development of the Learned ADMM + scheme and the similarity of the results is demonstrated later in this paper.

Chapter 5

Implementation

5.1 Operator Discretization Library (ODL)

In order to solve the aforementioned problems, this project makes extensive use of the Operator Discretization Library. ODL was developed of fast prototyping of inverse problems and contains a number of useful tools. Examples of these include linear and non-linear operators for tomography problems with implementations that use CUDA for GPU acceleration, the ability to generate the Shepp-Logan phantom, the implementation of popular solvers for inverse problems such as ADMM and PDHG and Filtered Back Projection, and interfaces for TensorFlow. For this particular project, ODL is used to generate the phantoms, to solve the total variation ADMM problem which is used as a baseline and serves as a starting point for the System Solve ADMM. ODL is integrated with the SciPy python scientific computing package which contains a number of methods for solving systems of equations. ODL also provides the Radon transform (Forward projection) and the transpose of the Ray transform (Back-projection) which can then be made into TensorFlow layers.

5.2 Ellipse Phantoms: Training Data

Having outlined the fundamentals of the CT problem, there still lies the question of what data do we wish to solve the problems with and also train the neural networks as neural networks work best with large amounts of training data. One method that answers these questions for the 2-D problem is to train the network on large amounts of clinically realistic simulated human data. However, in order to test whether the new schemes are valid propositions, a good starting point would be to work with a dataset of images formed from compositions of ellipses in various configurations [12]. This has the advantage of being able to generate training data sets of whatever size is needed. The data set generated in this case consists of images of many overlapping phantoms with different grayscale values. The images have pixel dimensions specified to be 128x128. The corresponding sinogram data is obtained using a total of 60 different angles (equidistributed over 180 degrees) for projections per image with the forward operator and Gaussian noise with magnitude of 5% of the mean magnitude of the sinogram data is applied.

5.2.1 Shepp-Logan Phantom: Validation Data

For validation step the Shepp-Logan phantom is used. The Shepp-Logan phantom was developed in 1974 is a set formation of ellipses used as a standard for CT reconstruction metrics. Though it may not be seen as a realistic phantom in a modern setting, it serves as a good first test. What has been noted about this image is that similarly to organic medical images there is sparse gradients meaning that there are significant continuous areas of constant values followed by edge-discontinuities [14] which relates back to the original choice of total variation penalisation.

5.3 Evaluating the results (SSIM/PSNR)

In order to evaluate the results, two metrics are used.

Peak Signal to Noise ratio:

PSNR is a long-standing metric however it is noted that PSNR does not always relate directly to *perceived* visual quality. It relies on a simple summation of error across the image which is not how the human visual system processes error.

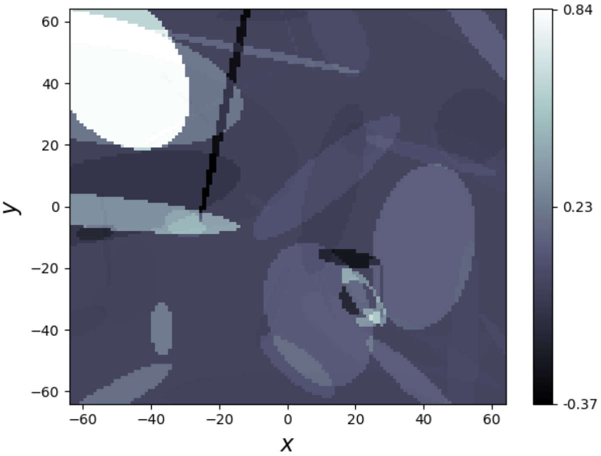


Figure 5.1: Ellipses Phantom

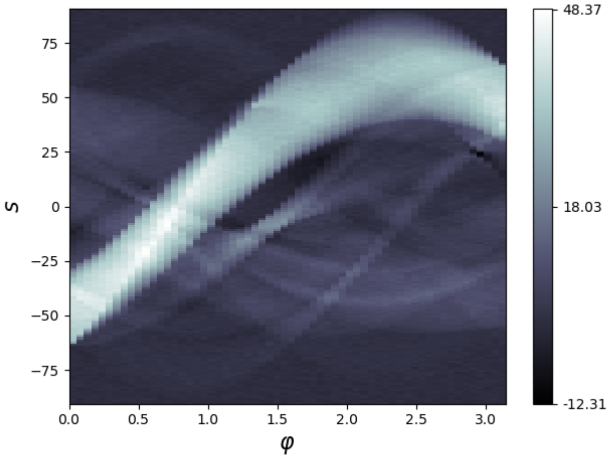


Figure 5.2: Sinogram with 5% noise

Therefore a more appropriate indicator of quality was needed.

$$PSNR = -10 \cdot log_{10}(\frac{MSE}{MAX^2})$$

Where MSE is the mean squared error $\frac{1}{n}\sum_{i=1}^{n}(x_{result} - x_{true})^2$, MAX is the highest possible pixel value and n is the number of pixels.

SSIM:

Conversely SSIM which was designed specifically for measuring image quality is based on the change in structural information in the image. It uses three comparisons;contrast, structure and luminance to create a composite metric [15]. This

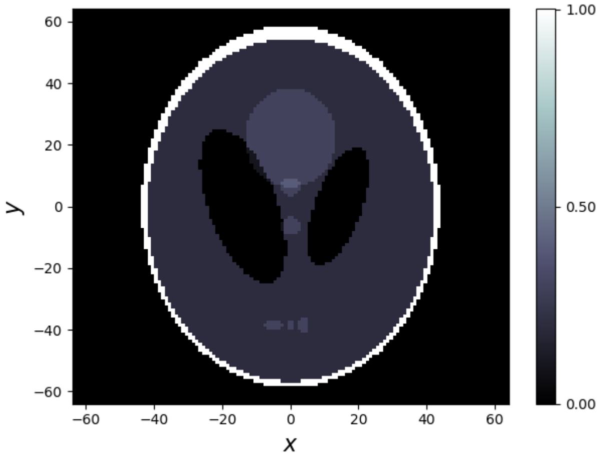


Figure 5.3: Shepp Logan Phantom

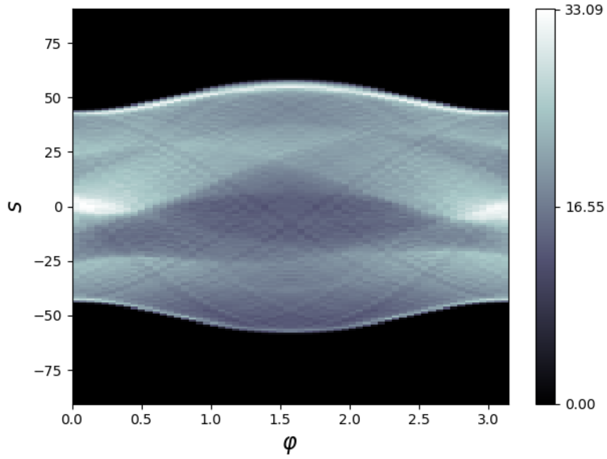


Figure 5.4: Sinogram of Shepp Logan Phantom with 5% noise

metric aligns more closely with the human visual perception system using windows x and y with dimensions $N * N$

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

- σ_x^2 is the variance of x
- σ_y^2 is the variance of y
- σ_{xy} is the covariance of x and y
- μ_x is the average value of x

- μ_y is the average value of y
- L is the pixel dynamic range
- k_1 and $k_2 << 1$
- $c_1 = (k_1 L)^2 c_2 = (k_2 L)^2$ are variables added for stability when the other denominator values are very close to zero.

5.4 Neural Network

The machine learning framework through which this method is implemented is TensorFlow, specifically TensorFlow 2.0 which runs natively in python. TensorFlow is a robust tool which has integration with many of the tools needed for this project such as ODL operators that function as TensorFlow layers. The back-end can also be GPU accelerated, in the case of using the tensorflow-gpu package.

The training was done using a Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz which has 12 cores and 24 threads cpu and more importantly a nVidia Tesla K40c with has 12GB GDDR5 and 2880 CUDA cores. The time taken to train the various networks was in the range of 10-18 hours depending on the network being trained.

Alder et al notes that for CT reconstructions, many of the useful properties for the forward operator and prior are translation invariant meaning the reconstruction operator should be translation invariant which suggests use of convolutional neural networks as building blocks for the neural network[16].

Two schemes are chosen for the forms of the learned proximal operators, firstly a series of convolutions and non linearities and secondly an identity plus a series of convolutions and non linearities. While the former is more to keep the learned operators more in the spirit of true non-learned proximal operators, the second choice is in the form of a residual network which has the benefit that each update does not need to learn the whole solution but rather only an update which has been shown to be easy to train.

$$Id + W_{w3,b3} * A_{c2} * W_{w2,b2} * A_{c1} * W_{w1,b1}$$

Still following from the format used for Learned Primal Dual Reconstruction [16] Affine operators parametrised by weights w_j and biases b_j are used which map

$$\mathcal{W}_{w_j b_j} : X^n \to X^m$$

and the kth component is given by

$$(\mathcal{W}_{w_j,b_j}([f^{(1)}, ..., f^{(n)}])^{(k)} = b_j^{(k)} + \sum_{l=1}^{n} w_j^{(l,k)} f^{(l)}$$

The activation functions used are Parametric Rectified Linear Unit (PReLU) functions.

$$A_{c_j}(x) = \begin{Bmatrix} x, & \text{if } x \geq 0 \\ -c_j x, & \text{else} \end{Bmatrix}$$

The difference between the PReLU activation and the popular ReLU is that rather than having the slope of negative portion being predefined, this value $-c_j x$ is learned by the network. This activation function has very little additional computational cost while typically producing superior performance. [17]

The learned proximal networks relies on convolutions of kernel size 3 x 3 pixels and each of the learned proximal networks use three layers of convolutional neural networks. For the Learned ADMM, the number of input channels are limited compared to the learned ADMM + as only one input channel is used for first learned proximal network leading to the channel structure $1 \to 32 \to 32 \to 32 \to 1$ and for the second proximal network two input channels are used leading to the channel structure (due to also using the corresponding sinogram data as an input for data fidelity) $2 \to 32 \to 32 \to 32 \to 1$. In total, this amounts to approximately 2.4 x 10^4 learned parameters.

In comparison, the Learned ADMM + has a channel structure of $6 \to 32 \to 32 \to 32 \to 5$ for the first and second proximal (which includes the corresponding sinogram data as an input) networks as additional channels are added to the primal variables to act as memory in the neural network, note that this is one less input channel than used in the Learned Primal-Dual scheme meaning fewer parameters are learned. 2.2 x 10^5 compared to 2.4 x 10^5

The network was trained to learn 10 ADMM iteration with each iteration having 2 ”proximal” steps and one dual update for each of the proximal methods shown in this project. For the learned ADMM method the learned parameters and proximal operators were kept consistent across each iteration while for the learned ADMM + method at each iteration a new set of learned parameters are learned.

In order to train the network an appropriate loss functional must be chosen. In this case the loss is chosen to be a mean squared error function of the reconstructed image and the true image. For the n length image variable x with loss $L(\theta)$

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}(x_{result} - x_{true})^2$$

This thesis also contains a promising alternative which is to consider the L1 norm for the objective function as this encourages sparsity in the difference between the learned and true image.

$$L(\theta) = \frac{1}{n}\sum_{i=1}^{n}|x_{result} - x_{true}|$$

For both the Learned ADMM and the Learned ADMM + 10,000 total batches were used i.e each time the network was trained on 10,000 different ellipses patterns. The schedule of the learning rate was a cosine decay meaning

$$r_t = \frac{r_0}{2}(1 + cos(\pi\frac{t}{t_{max}}))$$

The optimiser that was chosen is ”ADAM”. Introduced in 2015, the ADAM optimizer is highly popular and has a number of important distinctions from say a typical Stochastic Gradient optimiser. ADAM is relatively inexpensive as it only requires first-order gradients. The method computes individual adaptive learning rates for the different parameters from approximations of first and second moments of the gradients. This method combines advantages of two popular methods; AdaGrad and RMS Prop and it's performance training CNNs is known to be better. [18] The learning rate was set to 10^{-3} , batch size was set to five and the gradient norms were clipped to 1 for training stability.

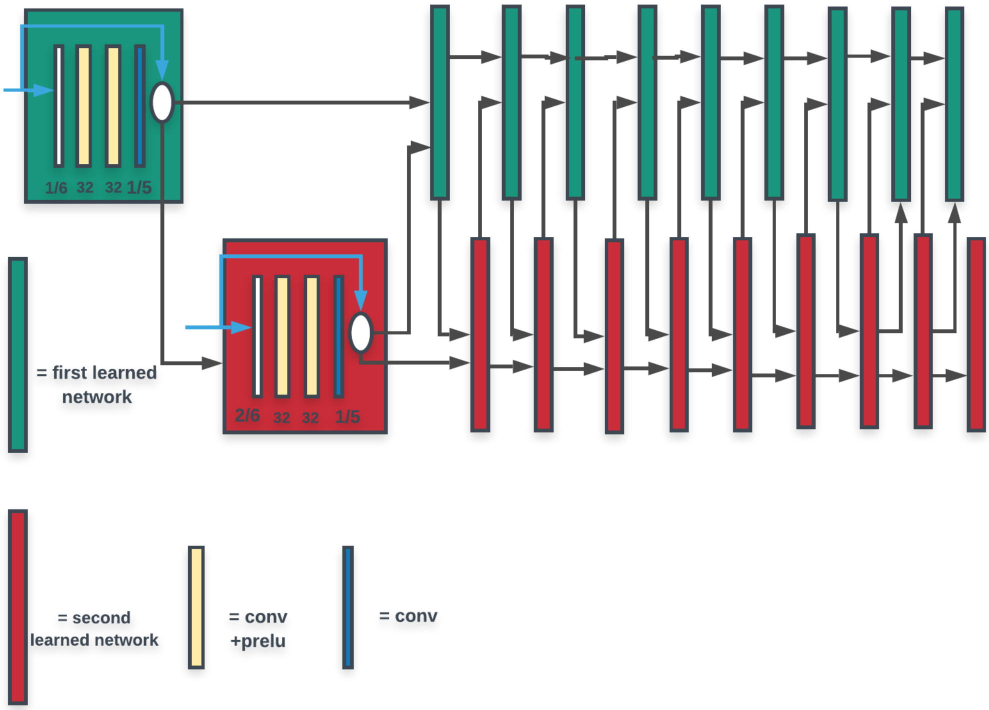


Figure 5.5: Diagram of neural network structure.The inputs to the first network 1 or 6 channels and the inputs to the second network have 2 or 6 channels depending on whether we implement Learned ADMM or Learned ADMM + respectively

Chapter 6

Results

As a baseline, to examine methods developed and implemented within this thesis, results are compared to two popular methods. The widely-established Filtered Back Projection and the Total Variation solution, solved using Linearised ADMM.

6.1 Filtered Back Projection

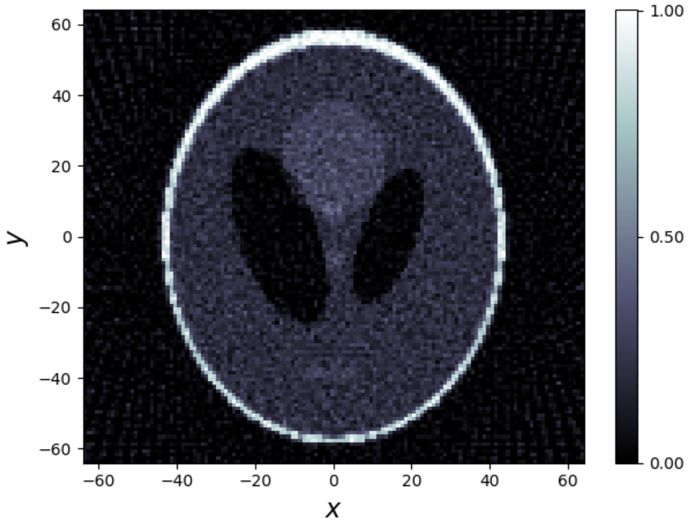


Figure 6.1: Filtered Back Projection

As a baseline Filtered Back Projection results are shown. The figure clearly shows a noisy image with artifacts due to the lack of sufficient projections and features which cannot be discerned such as the dots which should be present near

SSIM	0.5939
PSNR	21.9679
runtime	0.120s

Table 6.1: Filtered Back Projection Metrics

the bottom. The poor result is reflected in the quality metrics; the reconstruction however, is almost instantaneous.

6.2 Total Variation

For the ADMM optimisation, considering later on it will be compared to methods which learn optimal parameters for solving the problem, in order to have reasonable comparison the step size/quadratic penalty parameters σ, τ and γ and the regularisation parameter for the total variation function were tuned to provide optimal performance in terms of PSNR and SSIM. This method is also allowed a boosted starting point by choosing the filtered back projection for the initial value x_0 which serves as a good initial approximation and dramatically improves results. Rather than using only 10 iterations as used in the learned methods, 100 ADMM iterations are used as this provides better results that start to approach the performance of the learned methods.

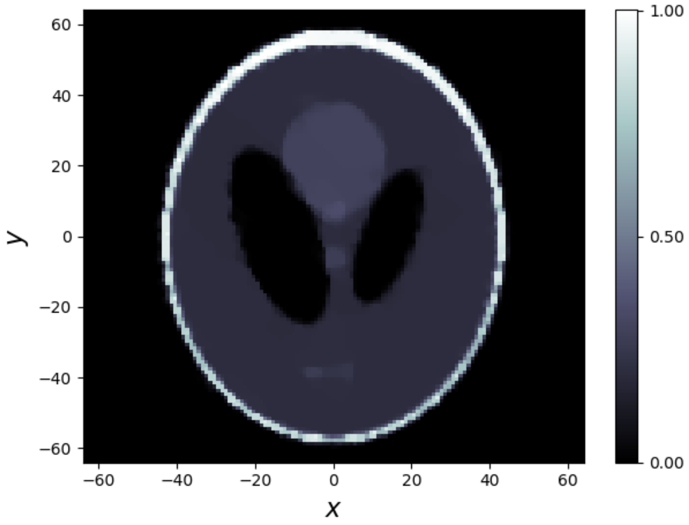


Figure 6.2: Total Variation ADMM

SSIM	0.9709
PSNR	26.8363
runtime	4.390

Table 6.2: TV-ADMM metrics

It should be noted that these results suggest very good performance with regards to SSIM, the PSNR results are subpar compared to the learned methods. Visually, this result presents a noticeable blurriness around the round edges of objects in the phantom.

6.3 Total Variation with Linear System Solve

The total variation when solving the linear system poses its own problem. It is not advisable to solve the linear system exactly due to the high expense of inverting the matrix $K^T K + \rho I$ in many cases as the system can become very large however, when using an iterative solver this inversion can be avoided. Another consideration is to what tolerance should the linear system of equations be solved specifically from the CT reconstruction problem as the tolerance of the solve has implications for the noise and data fidelity as well as convergence properties. Initially the choice was to solve the linear system using LSQR however this produced suboptimal performance. Therefore, this thesis examines the performance of various solvers for the linear system of equations in terms of time taken to achieve a tolerance of 6 x 10^{-5}. For the best performing solver, the quality of the reconstruction as a function of the accuracy of the system solver is also explored.

6.3.1 LSQR

This method is congugate gradient type method based on Golub-Kahan bidiagonalization. The LSQR solver has relatively low storage requirements and has better results for ill- conditioned matrices than simple least squares methods.[19]

6.3.2 Bicgstab

The biconjugate gradient stabilized method combines ideas of the conjugate gradient squared method with successive overrelaxation given by a three-term recurrence relation. It also requires more limited memory space than GMRES though this may not be a huge concern for a problem of this size.

6.3.3 GMRES

GMRES is a minimal residual method. This method has some benefits for stability, convergence as a minimal residual method, convergence to the required tolerance is guaranteed and does not have problems with early termination. The GMRES solver provides the best solution being over 10 seconds faster than the next closest, BIGSTAB.

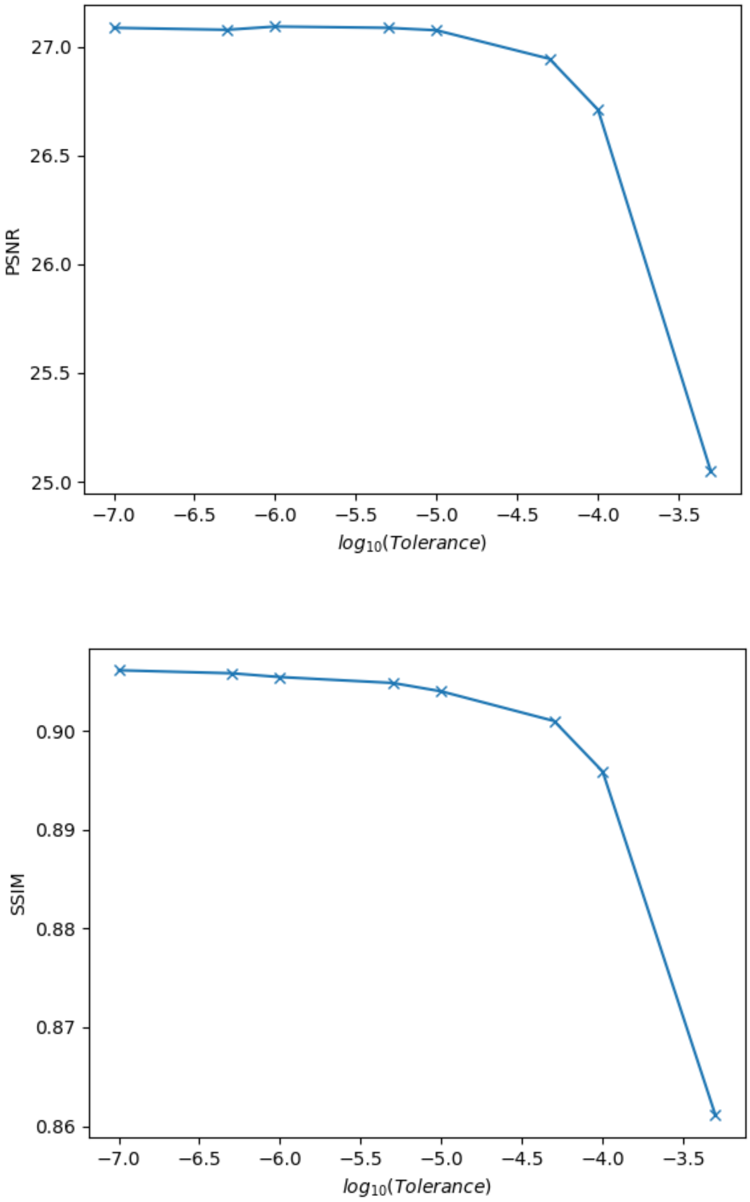
runtime of LSQR	99.989s
runtime of BICGSTAB	40.828s
runtime of GMRES	**31.384s**

Table 6.3: Runtime of various iterative solver with time in seconds

6.3.4 Reconstruction as a function of tolerance

The relationship between reconstruction performance, in terms of both PSNR and SSIM, and the tolerance to which the system of equations is solved is outlined in this graph. This is highly important as increased accuracy increases the time to solve the system so we would like to understand the minimum tolerance that still provides adequate properties.

Using a range of different tolerances for the linear solver the ADMM scheme was run for 50 iterations and the final SSIM and PSNR results were recorded. The graph shows clear diminishing returns. Interestingly, the PSNR shows diminishing returns much earlier compared with SSIM re-emphasising the benifits of SSIM as a metric.



(a)

(b)

Figure 6.3

Figure 6.4: Log base 10 of Tolerance against (a) PSNR (b) SSIM

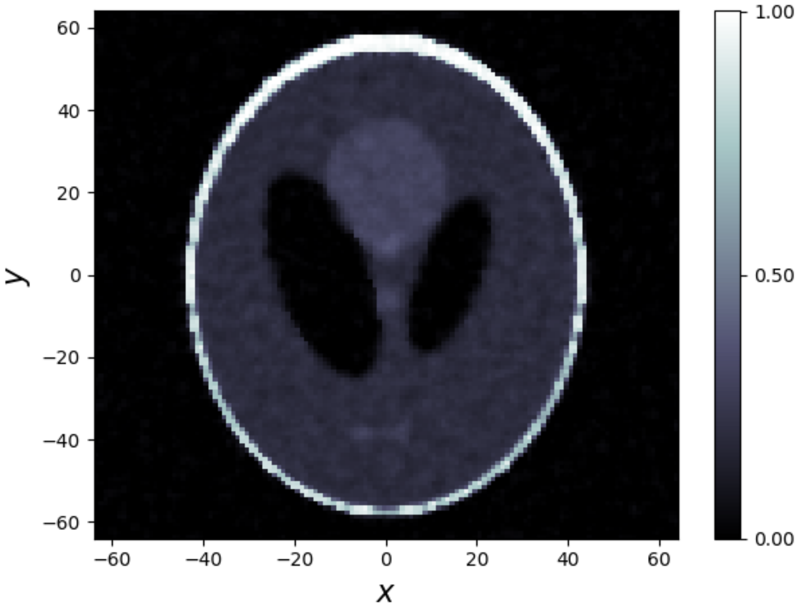


Figure 6.5: Reconstruction using System Solving ADMM

The result for this ADMM variant has a grainier quality and poor SSIM for the given number of iterations and slower run time which is reflected in the poor metrics.

SSIM	0.932783303572
PSNR	27.6081991647
runtime	7.080s

Table 6.4: Results for ADMM with system solver

6.4 Learned ADMM

The Learned ADMM reconstruction marks a significant improvement over the non-learned methods. It should be noted that until the last iteration, the reconstruction was relatively poor and the error is not monotonically decreasing with each iteration, similar to what was seen in *Alder* et al [16] which implies that rather than the expected iterative scheme the network performs some image enhancements and then applies reconstruction step towards the end of the scheme. What can be noted qualitatively about the reconstruction when compared to the non-learned methods is vastly superior quality of edges implying some edge-preserving steps are taken, though there is some localised contrast inconsistency which is the only noticeable downside to this method compared to Learned ADMM +. One possible reason for this could be that while it is hoped that the network can learn some gradient based step similar to the total variation norm proximal step it is not hard encoded into the network so there may not be much penalisation for high gradients (implying larger localised changes in value). The following figures show one step from the learned scheme. Intuitively, when we compare the output of this step to the euclidean norm of the contrast gradient of the Shepp-Logan phantom in the x and y directions there appears to be some similarity.

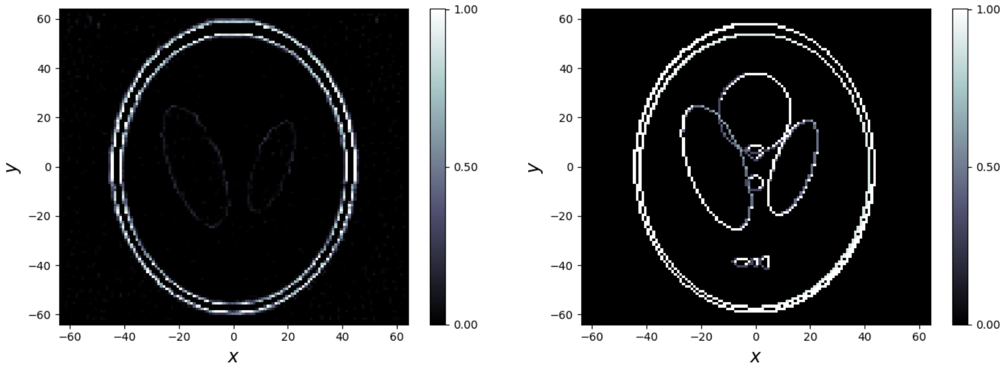


Figure 6.6: 3rd step of Learned ADMM compared to sum of the gradient of the Shepp-Logan phantom

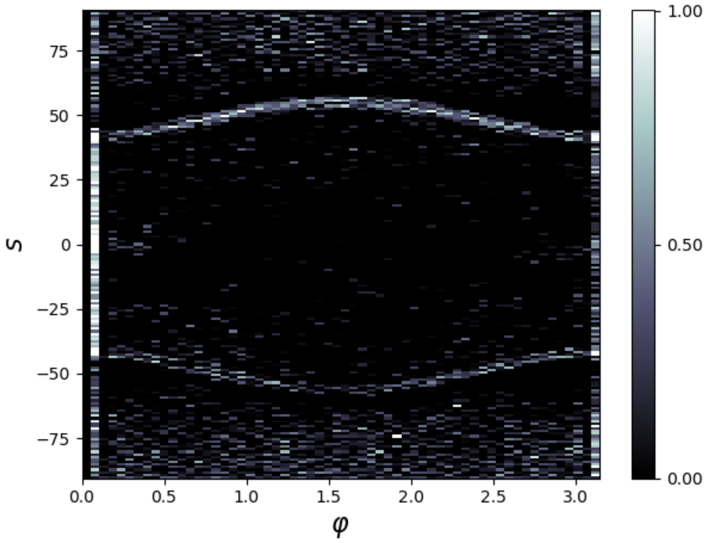


Figure 6.7: Sinogram of 3rd step

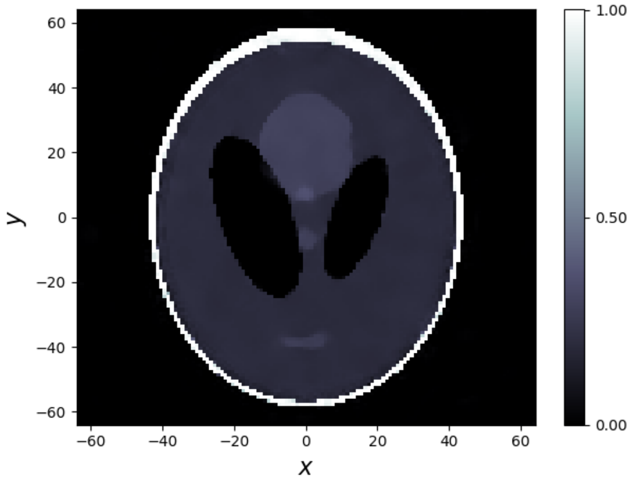


Figure 6.8: Learned ADMM with L2 Loss function

SSIM	0.98302235622
PSNR	35.4152460377
runtime	0.283

This method is further built on by attempting to solve the issue of colour inconsistency by training using an L1 norm loss function to encourage sparsity in the error. The results are very promising showing excellent SSIM results with a visibly crisper image though it is interesting to note that this use of the L1 norm was insignificant in improving the PSNR metrics.

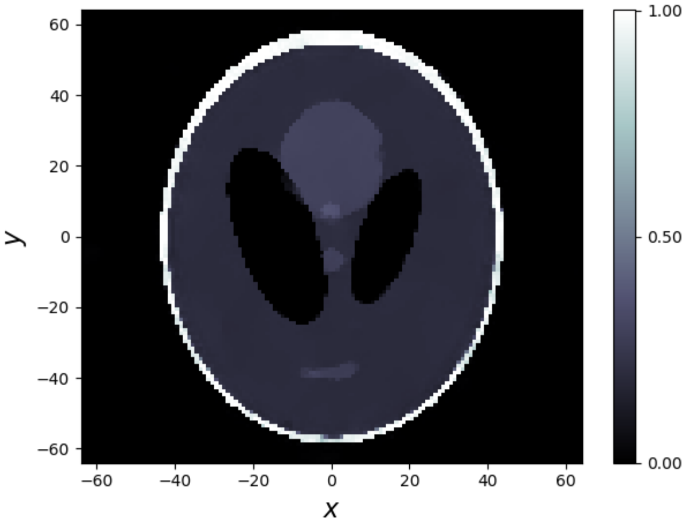


Figure 6.9: Learned ADMM with L1 Loss function

SSIM	0.990524696686
PSNR	35.11833905687
runtime	0.340

6.5 Learned ADMM +

The Learned ADMM + method further improves on the previous learned method when using the L2 norm loss function. By incorporating memory and learning an update, the SSIM and PNSR values improve further with PNSR increasing over 3dB. It should be noted that unlike the base Learned ADMM, the residual for this method reduces with each iteration, more similar to a traditional ADMM scheme. The problem of localised colour inconsistency that was seen in the Learned ADMM is noticeably reduced with Learned ADMM +, which is reflected in the SSIM score.

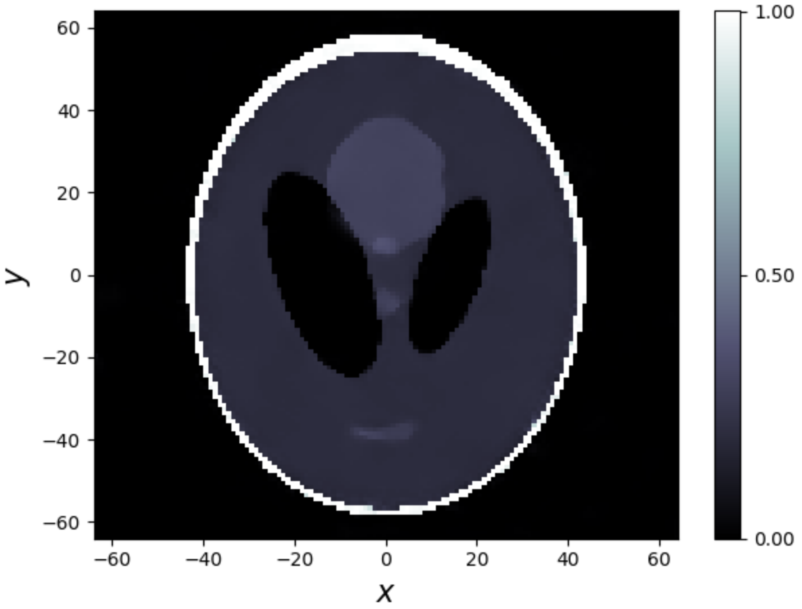


Figure 6.10: Learned ADMM + reconstruction

SSIM	0.987537365257
PSNR	38.8156018375
runtime	0.291

Table 6.5: Learned ADMM + metrics

6.6 Final Results Table

In this table, results for the Primal Dual methods are also included for comparison (in italics) and are taken from the Learned Primal Dual Reconstruction paper [16]. The runtime results for these methods were not included as the comparison is not useful because different hardware was used. The results show that the ADMM methods produce better overall results in both SSIM and PSNR score though as expected, the Learned ADMM+ and Learned Primal Dual show similar performance. All of the learned methods are much faster that the non-learned methods as they require far less applications of the forward and back projection operators.

Name	SSIM	PSNR	Runtime
FBP	0.5939	21.9679	0.120
PDHG-TV	0.9290	28.06	N/A
ADMM-TV	0.9709	26.8363	4.390
ADMM-SS	0.9328	27.6082	7.080
Learned PDHG	0.9090	28.32	N/A
Learned Primal-Dual	0.9890	38.28	N/A
Learned ADMM	0.9830	35.4152	0.283
Learned ADMM L1	**0.9905**	35.1183	0.340
Learned ADMM +	0.9876	**38.8156**	0.291

Table 6.6: Full Table of Results

Chapter 7

Discussion

The results show that all methods implemented in this thesis show significant improvements over the baseline Filtered Back-projection. The results also show that the use of Linearised ADMM with the total variation regularisation provides noticeable improvements over previous implementations of baseline iterative methods (Primal Dual Hybrid Gradient) while exhibiting a similar runtime. The Linearised ADMM also provides high quality results in terms of structural similarity, though not necessarily in terms of PNSR performance which is very similar to what was noticed about the PDHG method in [16]. The obvious downside to iterative methods such as this are the increases in time to run the method (more than 30x longer than FBP) and the need to tune the parameters for the problem, however it may be argued that compared to the length of time needed to train a learned method for instance, the time taken for the tuning of parameters is insignificant and even though it takes 30x longer than FBP, FBP itself is extremely fast. The results were also not significantly worse than the learned methods as there is equal localised constrast consistency. However the clearer edges provided by the learned methods might be more beneficial to the task of recognizing anomalies in medical scenarios so the trade off between developing a learned method compared to easily implementable methods such as Linearised ADMM must be considered.

With regards to the linear system solving ADMM, the proposed alternative solvers (BICGSTAB , GMRES) provide significant improvements over the LSQR solver which is typically suggested for this type of problem. This method does however have significant drawbacks. Even with the improved speed of the linear system solve, within the Python SciPy enviroment, construction and solving the linear system still represents a significant overhead over elementwise functions which are sometimes used for the L2 norm proximal and so it would be useful to explore how to make this more efficient but as presently constituted in this thesis the linear system solve method is not performant enough to be competitive with current methods. With regards to the learned methods, the most significant benefits were in the Learned ADMM method. The closest point of comparison would be the Learned PDHG which has almost exactly the same number of parameters to learn and a similar formulation which also does not use ”memory channels”. Even with these similarities, the Learned ADMM method has significantly better performance with 7db improvements in PSNR and a 0.8 improvement in SSIM (due to how SSIM scales an increase of 0.8 represents a noticeable improvement) and it much is closer in performance to learned primal dual methods that incorporate memory. Some of the iteration reconstructions observed in the method had some unusual properties for example the 3-4th step detects and highlights the outer rings, the 9-10th step performs a colour inversion. The significant difference in the type of results produced by the network at each ADMM iteration is most likely due to not using a residual network (i.e learning the whole update) for each step.

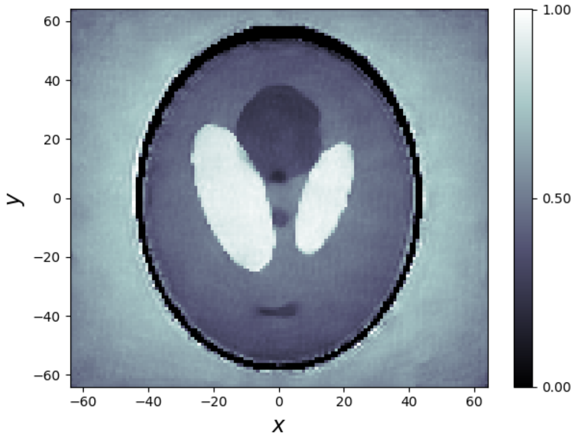


Figure 7.1: Step 9 of Learned ADMM

The thesis also shows significant improvements to the results of the training can be found by using an L1 norm learning objective to induce sparsity in the error. Previous work has noted the issues with the L2 norm as it assumes that the significance of noise is independent on the local characteristics of the image and that the noise is white Gaussian and suggested novel loss functions based on SSIM + L1 or L1 minimisation followed by L2 minimisation which may improve even on the L1 norm performance [20].

Finally, looking at the Learned ADMM + method, while the method does provide similar results to the Learned Primal-Dual method, the Learned ADMM + is not a significant improvement. This is likely because in Linearized ADMM which the Learned ADMM methods are based on can be seen as optimally preconditioned PDHG methods in many cases, so as more of the parameters and combinations become learned, the difference between the two methods becomes less defined. Because the Learned ADMM + learns a residual update to the current update, it exhibits vastly different properties to the Learned ADMM in terms of how the reconstruction changes iteration to iteration. This method shows an improvement in reconstruction quality at each iteration and the type of the reconstruction does not appear to change between ADMM steps. However, the Learned ADMM + actually has slighty worse performance than the Learned ADMM trained on the L1 loss function even when the Learned ADMM + is trained on L1 loss function, while at the same time the Learned ADMM uses around 1/6th of the parameters.

Interestingly all reconstruction methods fail to produce separation of the three circles at the bottom part of the Shepp-Logan phantom, so it might be assumed that this is a intrinsic limit due to lack of projections and the noise present.

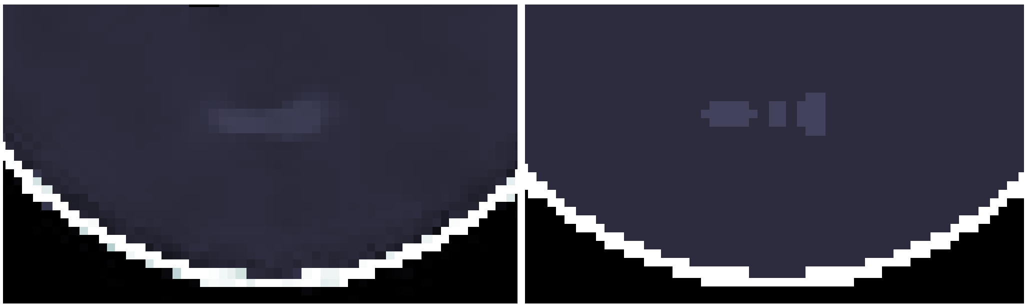


Figure 7.2: Close up of bottom of reconstruction

Chapter 8

General Conclusions

8.1 Conclusions

This report manages to explore various iterative solvers for the system of equations needed for the data fidelity proximal operator and provides suggestions for appropriate solvers which show improved performance over the widely used LSQR iterative solver with these results suggesting the use of GMRES while also examining how the quality of the reconstruction is affected by the degree of accuracy chosen for the system solver. The report also provides the experimental results which give clear guidance as to the required tolerance to solve the system in order to provide convergent results with 60e-5 being a good guidance point for problems of this size allowing for the method to use optimal run time by not solving the system to excessive levels. This report also shows that by using the Linearized ADMM as the base framework for the neural network, a high performance reconstruction scheme can be learned while using significantly fewer learned parameters when compared to the Learned Primal-Dual method. The report also finds that while the Learned ADMM shows far better results that it's counterpart Learned Primal Dual Hybrid Gradient, the extended Learned ADMM + scheme does not show any significant improvements over the learned Primal-Dual scheme. Finally, when evaluating the reconstruction quality based on SSIM which is widely deemed to be a superior metric compared to PSNR, the use of the L1 norm as a loss function for Learned methods produces superior results, even comparing to more complex networks trained on the L2 norm loss function.

8.2 Future Work

While the learned methods has showed promising results, there are many aspects that could be considered for improvement. Firstly, rather than having a full iterative reconstruction scheme which needs to learn two proximal networks, the use of non-learned methods to solve the data fidelity update should be considered as, so long as the assumptions on the noise model and forward operator are correct, learning this step is unnecessary [11]. This could not be implemented in this project due to time constraints. Another possible area of exploration is refining the neural network, in this project a relatively simple structure is used with no network regularisation such as dropout, furthermore the depth of the CNN could be increased with possibly some use of recurrent networks to more natively handle keeping memory between iterations.

Finally, looking at the results produced by learned methods compared to the non learned methods, one area in which the non learned methods exceeded some of the learned method is the ability to capture the sparse gradient nature of solution/contrast consistency in contiguous regions. Therefore it may be advisable to extend the Learned ADMM to a multiblock system which directly incorporates the gradient of the image rather than indirectly hoping the network will learn to encode a gradient.

Appendix A

Reconstruction Images

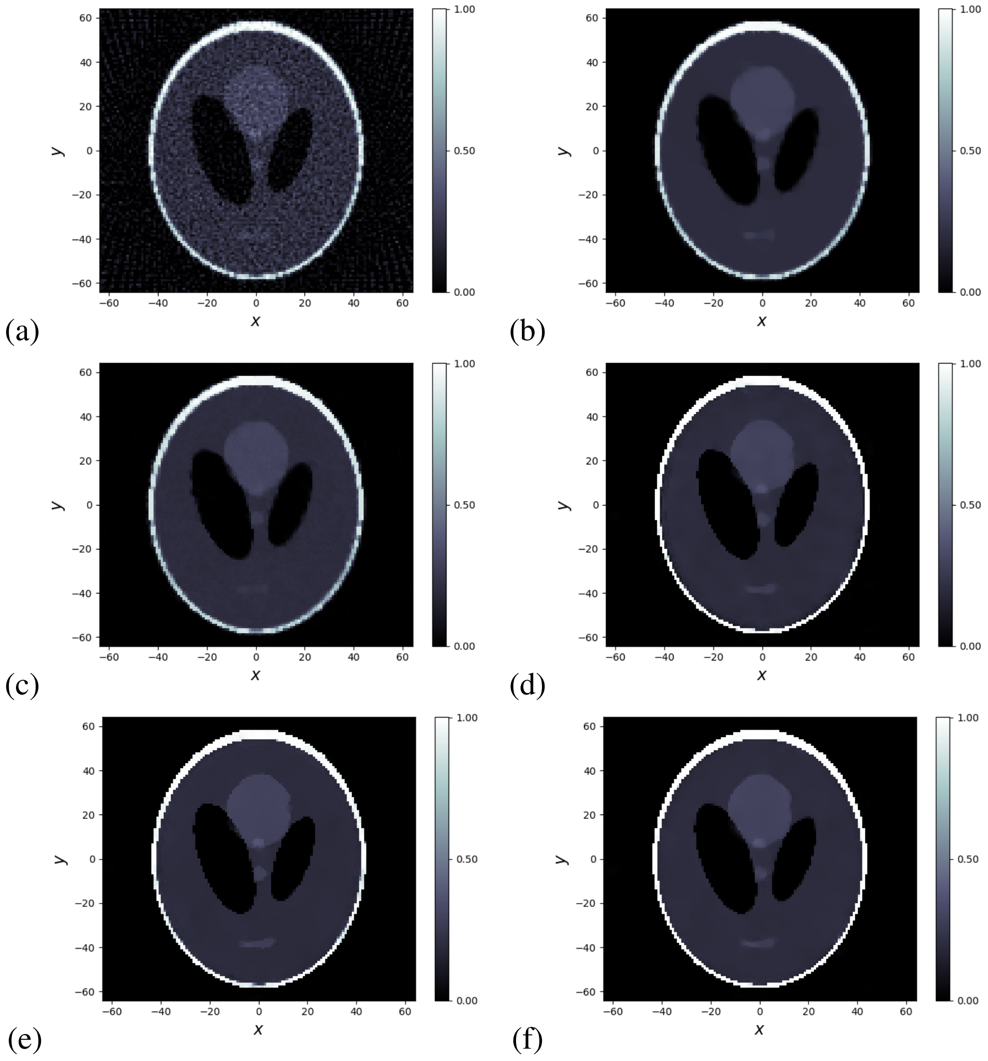


Figure A.1: (a)FBP (b) TV-ADMM (c)ADMM-SS (d) Learned ADMM (e) Learned ADMM L1 (f) learned ADMM +

Bibliography

[1] Angela Cantatore and Pavel Müller. *Introduction to computed tomography*. DTU Mechanical Engineering, 2011.

[2] CT Scan (CAT Scan): Purpose, Procedure, Risks, Side-Effects, Results.

[3] R. Courant and D. Hilbert. *Methods of Mathematical Physics*. 2008.

[4] Martin Benning. Inverse Problems. Technical report.

[5] Amir Averbuch, Ilya Sedelnikov, and Yoel Shkolnisky. CT reconstruction from parallel and fan-beam projections by a 2-D discrete radon transform. *IEEE Transactions on Image Processing*, 2012.

[6] Henrik Turbell. Cone-beam reconstruction using filtered backprojection /. 2001.

[7] Stanley H. Chan, Xiran Wang, and Omar A. Elgendy. Plug-and-Play ADMM for Image Restoration: Fixed-Point Convergence and Applications. *IEEE Transactions on Computational Imaging*, 2016.

[8] Yaniv Romano, Michael Elad, and Peyman Milanfar. The Little Engine That Could: Regularization by Denoising (RED). *SIAM Journal on Imaging Sciences*, 2017.

[9] Edward T. Reehorst and Philip Schniter. Regularization by Denoising: Clarifications and New Interpretations. *IEEE Transactions on Computational Imaging*, 2018.

[10] Hangrui Yue, Qingzhi Yang, Xiangfeng Wang, and Xiaoming Yuan. Implementing the ADMM to Big Datasets: A Case Study of LASSO. Technical report, 2017.

[11] Tim Meinhardt, Michael Moeller, Caner Hazirbas, and Daniel Cremers. Learning Proximal Operators: Using Denoising Networks for Regularizing Inverse Imaging Problems. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[12] Jonas Adler and Ozan Öktem. Learned Primal-Dual Reconstruction. *IEEE Transactions on Medical Imaging*, 2018.

[13] Ernie Esser, Xiaoqun Zhang, and Tony F. Chan. A General Framework for a Class of First Order Primal-Dual Algorithms for Convex Optimization in Imaging Science. *SIAM J. Imaging Sciences*, 3:1015–1046, 2010.

[14] Ray Maleh and Martin Strauss. Efficient Sparse Approximation Methods for Medical Imaging. 2009.

[15] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 2004.

[16] Jonas Adler and Ozan Öktem. Learned Primal-Dual Reconstruction. *IEEE Transactions on Medical Imaging*, 2018.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. Technical report.

[18] Diederik P Kingma and Jimmy Lei Ba. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. Technical report.

[19] Christopher C. Paige and Michael A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software (TOMS)*, 1982.

[20] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss Functions for Image Restoration with Neural Networks. Technical report.