

Information Maximizing Generative Adversarial Network Regularized ADMM for CS-MRI Reconstruction

A thesis submitted
in partial fulfillment for the award of the degree

Master of Technology
in
Digital signal processing

by

Deepak Raya

Under the guidance of:

Dr. J. Sheeba Rani

Dr. Deepak Mishra



Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, India

June 2021

Certificate

This is to certify that the thesis titled *Information Maximizing Generative Adversarial Network Regularized ADMM for CS-MRI Reconstruction* submitted by **Deepak Raya**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Master of Technology in Digital signal processing** is a bona fide record of the original work carried out by him/her under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. J. Sheeba Rani
Associate Professor

Dr. Deepak Mishra
Head of the Department, Avionics
Associate Professor

Place: Thiruvananthapuram

Date: June 2021

Declaration

I declare that this thesis titled *Information Maximizing Generative Adversarial Network Regularized ADMM for CS-MRI Reconstruction* submitted in partial fulfillment for the award of the degree of **Master of Technology in Digital signal processing** is a record of the original work carried out by me under the supervision of **Dr. J. Sheeba Rani** and **Dr. Deepak Mishra**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgments have been made wherever the findings of others have been cited.

Place: Thiruvananthapuram

Date: June 2021

Deepak Raya

(SC19M048)

This thesis is dedicated to my parents, Smt. Shri. Ratna Devi and Shankar Rao

Acknowledgements

I would like to express my deep sense of gratitude to my thesis supervisors **Dr. J Sheeba Rani**, Associate Professor, Department of Avionics, IIST and **Dr. Deepak Mishra**, Associate professor, Department of Avionics, IIST for their constant support and invaluable guidance during this project. I would like to give a special thanks to **Dr. Vineeth B.S**, Assistant Professor, Department of Avionics, IIST for taking all the responsibility of conducting project update meetings and guiding us through the process and evaluations.

I am extending my gratitude to **Shri. Sarath Babu**, PhD Research scholar, Department of Avionics, IIST for his resources about report formatting and research guidelines. And to **Dr. Sudeesh Chethil**, Associate Professor, Department of Physics, IIST and **Mrs. Divya Mohan S**, Technical Assistant, IIST for providing the computational resources and remote access for the experiments and simulation.

I am extremely grateful to **my family**, without whom I could not have accomplished this project during this hard pandemic times. I am fortunate to have open minded and helpful classmates and seniors, throughout my journey during my Master's, I specially thank **Avinash Ch**, PhD Research scholar; **Ganesh Kumar**, **Amith Singh**, and **Mahesh Pal**, M.Tech(DSP).

Deepak Raya

Abstract

Compressed sensing theory has provided a great potential for reconstructing signals from significantly under-sampled measurements. The algorithms developed for CS reconstruction are of iterative nature, which hinders its usage to time-critical applications such as medical imaging. But, recent advances in deep Neural Networks and their enormous capability in data driven learning, has been extended to solve ill-posed inverse problems, such as compressed sensing. There have been many instances in which deep learning has provided success in CS reconstruction. Besides the deep Feed-forward Neural Nets, the success of Deep Generative Models, such as VAEs and GANs, in learning the underlying distribution and statistics of data, will further aid the efficacy of reconstruction procedure.

The main aim of this project is to reduce MRI acquisition time taken for collecting entire k-space measurements and constructing the image, i.e. to construct a good approximation of the MR image only with fewer samples in k-space. This thesis presents an algorithm, which combines both iterative optimization and Deep unsupervised learning based method. The interesting thing about this approach is instead of searching among the sparse priors, we search in a space constrained by the output of generator network, which eliminates the need for sparsity assumption in a hand-crafted basis.

The algorithm makes use of ADMM based iterative optimization, with the statistical prior of MR image learned through info-GAN trained on MR brain images data set and a projector which trained to structure the latent space. It performs denoising like update with a composition of generator and projector. Although this work has been directly inspired by f-CSR [1] algorithm, there are some crucial aspects that are tailored according to the need of CS-MRI, first the under-sampling is done in k-space and not in pixel domain, second the images are of size 256×256 , hence the neural networks architecture has been scaled and modified accordingly; furthermore, the images are not vectored, in order to eliminate the requirement of huge under-sampling matrix. Also the under-sampling is slightly different from traditional randomized sampling used in general CS and is physically feasible for MRI modality. This algorithm reconstructs the image within ten iterations of ADMM updates, and the SSIM scores are above 0.6 and raises up to 0.75 and in rare cases goes around 0.5.

Contents

List of Figures	x
List of Algorithms	xii
Abbreviations	xiii
Nomenclature	xiv
1 Introduction	1
1.1 Related Work	1
1.2 Overview of the approach	2
1.3 Contributions	3
2 Literature Survey	4
2.1 Compressed sensing using generative model	4
2.2 Deep De-Aliasing GAN	4
2.3 GAN with cyclic loss	5
2.4 Structure Preserving CS-MRI reconstruction	5
2.5 Fast compressed sensing recovery using generative networks	5
3 Information Maximizing GAN	6
3.1 Introduction to GANs	6
3.2 GANs learn the underlying data distribution	6
3.3 Disentangling the representations using information maximizing GAN	9
3.3.1 Encouraging the mutual information maximization	9
3.3.2 The control codes	11
3.3.3 Algorithm and implementation	12
3.4 Summary	12

4	Alternating Direction Method of Multipliers	13
4.1	Primal to Dual	13
4.2	Dual Decomposition	14
4.3	Augmented Lagrangian	15
4.4	ADMM	15
4.5	ADMM for Image recovery	16
4.6	Summary	17
5	Methods and Contributions	18
5.1	Data set	18
5.2	Latent Space Design	19
5.3	Info-GAN: generation of data like samples	19
5.4	Projector Network: Constraining the latent space	23
5.5	Under-sampling of k-space: CS-MRI	25
5.6	ADMM+info-GAN+Projector: CS-MRI Reconstruction	25
5.7	Summary	27
6	Results, Discussion and Conclusion	29
6.1	Results and Discussion	29
6.2	Conclusion	33
6.3	Future Work	36
	Bibliography	36
	Appendices	40
A	Principles of MRI and MRI acquisition	40
A.1	Net Magnetization and T ₁ ,T ₂ Processes	40
A.1.1	T ₁ Process	41
A.1.2	Precession and T ₂ Process	41
A.1.3	Time Domain NMR signal	41
A.2	From NMR signals to Image	42
A.2.1	The Three main systems of MRI	42
A.2.2	Fourier Transform Tomography	44
B	Compressive Sensing	48
B.1	Compressed Sensing reconstruction formulation	48

B.1.1	Requirements for a successful Compressed Sensing	49
B.2	Algorithms for Compressed Sensing Reconstruction	51
B.2.1	Matching Pursuit	52
B.2.2	Iterative Thresholding	55
B.2.3	Approximate Message Passing	56

List of Figures

1.1	Flow of the approach	3
3.1	Flow-chart depicting the adversarial learning in GANs [11]	7
3.2	Generative Advesarial Network Model: The expression for standard GAN loss function is shown in Red and the expressions in black show the binary cross entropy loss for each network	8
3.3	Info-GAN modality: C is the latent code word, Z is the noise latent variable (flow: bottom to top)	9
5.1	A sample of MCCAI dataset images	18
5.2	Generator Network: Each is followed by a ReLU non-linear activation and Batch Normalization, except for the last layer which uses Tanh activation layer. (Size of output mentioned below each layer, and kernel(k), stride(s) and padding(p) mentioned between the layers)	20
5.3	Discriminator Network: Each followed by Leaky-ReLU and a Batch Normalization, except for last which used sigmoid activation	21
5.4	Auxiliary network: bifurcates from sixth layer of discriminator, this network predicts the latent code that had generated the image. The intermediate layer is followed by ReLU and Batch Normalization, and last layer is a linear	21
5.5	Discriminator loss(green) and Generator loss(orange) Vs iteration plot: An adversarial game between generator and discriminator can be observed here. The loss is a binary cross entropy loss	22
5.6	Auxiliary Network Loss function plot Vs iteration	22
5.7	Sample of generated images after the training	23
5.8	The projector network: maps the image to latent code. (from right to left)	23

5.9	Projector training modality: The projector will try to find the latent variable close to the latent variable that generated a particular image, in MSE sense .	24
5.10	Projector loss Vs Training iteration: MSE loss function is being optimized .	25
5.11	undersampling masks and corresponding matrices	25
5.12	ADMM iteration for CS-MRI reconstruction	26
6.1	Reconstruction results with compression ratio 1.33; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR .	30
6.2	SSIM and MSE between the true image and reconstructed image for compression ratio 1.33: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric	30
6.3	Reconstruction results with compression ratio 2; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR . . .	31
6.4	SSIM and MSE between the true image and reconstructed image for compression ratio 2: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric	31
6.5	Reconstruction results with compression ratio 2.66; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR .	32
6.6	SSIM and MSE between the true image and reconstructed image for compression ratio 2.66: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric	32
A.1	Linear gradient of magnetic field	43
A.2	43
A.3	Frequency Encoding: (a) and (b) represent the frequency encoding when the sample is placed in different orientations, (back projection reconstruction is performed by stage wise complete rotation)	44
A.4	Phase encoding: 7 repetitions of phase gradient activations with distinct gradient magnitudes are shown here	44
A.5	The timing diagram for Fourier transform imaging sequence	45
A.6	Slice Selection	46

B.1 Compressively sensed signal with transform sparsity $S = 4$ (adopted from [2])	50
--	----

List of Algorithms

3.1	GAN training algorithm using Mini-batch SGD	8
3.2	Info-GAN training algorithm	12
5.1	info-GAN regularized ADMM for CS-MRI reconstruction	27
B.1	Orthogonal Matching Pursuit	53
B.2	Compressive sampling matching pursuit	54
B.3	Iterative Hard Thresholding	55
B.4	Iterative Soft Thresholding	56

Abbreviations

CS	Compressed Sensing
GAN	Generative Adversarial Network
Info-GAN	Information Maximization GAN
ADMM	Alternating direction Method of Multipliers
VAE	Variational Autoencoders
BCE	Binary Cross Entropy loss
MSE	Mean Squared error
NRMSE	Normalized Root Mean Squared Error
PSNR	Peak signal to noise ratio
SSIM	Structural similarity index measure
MRI	Magnetic Resonance Imaging
ZFR	Zero Filled Reconstruction
f-CSR	fast Compressed sensing recovery using generative models
LASSO	Least Absolute Shrinkage
PnP ADMM	Plug and Play ADMM
MICCAI	Medical Image Computing and Computer Assisted Intervention
ADAM	Adaptive moment estimation
CS-MRI	Compressed sensing MRI
FID	Free Induced Decay
NMR	Nuclear Magnetic Resonance
RF	Radio Frequency
FT	Fourier Transform
SNR	Signal to Noise Ratio
FOV	Field of view

Nomenclature

μ	mean
σ^2	variance
σ_{xy}	covariance between x and y
\mathcal{F}	Fourier Transform
\mathcal{F}^{-1}	Inverse Fourier transform
l_1	L-1 Norm
l_2	L-2 Norm
D_{KL}	Kullback Leibler divergence
$I(x, y)$	Mutual Information between random variables x and y
$H(x)$	Entropy of random variable x
p_x	Probability distribution of random variable x
\mathbb{E}	Expectation
$V(D, G)$	GAN loss

Chapter 1

Introduction

Magnetic Resonance imaging is a widely used safe and non-invasive medical imaging technique, with a reliable resolution and tissue contrast. One of the important drawback of Magnetic Resonance Imaging is longer acquisition times which causes patient discomfort, delay in critical situations and also there is high probability for movement induced artifacts. One way to overcome this drawback is to apply compressed sensing technique to MR images(CS-MRI), i.e. to reconstruct the MRI image from significantly fewer measurements, in contrast to Shannon-Nyquist constraint. This project explores a deep unsupervised learning based approach for CS-MRI reconstruction.

1.1 Related Work

With traditional CS, sparsity in pixel or a transform domain is required for reconstruction, which is formulated as a regularized optimization with the sparsity constraint. This assumption requires handcrafted basis that promote sparsity, and in most cases the signals are not exactly sparse but approximately sparse, therefore this assumption might not result in accurate reconstruction and requires an expert knowledge for handcrafting a sparse basis based on the nature of the signal. On the other hand there are learning based approaches for CS reconstruction. This section discusses some of the learning based compressed sensing MRI reconstruction methods.

One form of learning based method is Dictionary learning MRI reconstruction (DLMRI)[3], which adaptively learns the sparsifying basis and simultaneously reconstructs the image from under-sampled measurements, but this takes long computational time due to iterative optimization nature.

Another approach deep ADMM net [4] for CS-MRI reconstruction defines an unrolled deep neural network over the data flow graph derived from iterative ADMM optimization

method for CS based MRI and learns the parameters of the algorithm from data. Accelerating magnetic resonance imaging [5] is a deep learning based method in which a deep convolutional neural network learns to map under-sampled to and true images and tries to reconstruct the true image, when an under-sampled MR image is presented. Bora et al. [6] has proposed the use of generative models for compressed sensing, in which the CS optimization is regularized by constraining the latent space, this has inspired the use of generative models for CS. First generative model based CS-MRI reconstruction is proposed in DAGAN [7] for CS-MRI, uses a conditional GAN conditioned on zero filled reconstructed MR image, with U-net based architecture and modified loss function. Structure preserving CS-MRI reconstruction using GAN, similar to DAGAN uses a zero filled reconstruction conditioned GAN with the generator trained based on GAN loss augmented with pixel domain l_1 loss and a SSIM based loss with a "residual in residual blocks" [8] in a U-net fashion. A network called RefineGAN is introduced in [9] The generator contains two networks one which reconstructs the image and the other which refines the reconstruction and they propose a novel loss called cyclic loss. "Fast compressive sensing recovery using generative models with structured latent variables" [1] proposes f-CSR algorithm (fast compressed sensing reconstruction using generative models) which uses a trained GAN and projector network in a ADMM setup to perform general scenario CS reconstruction. This is the primary inspiration for the method used in this project.

1.2 Overview of the approach

The approach presented in [1] has been extended for the case of CS-MRI reconstruction, an Info-GAN (Information maximizing GAN) is trained on MRI brain images with the standard GAN loss, which learns the statistics of the data and generate brain like images i.e. it maps a latent variable to a brain like MR image. Then a projector network is trained using a mean squared error loss in order to learn the latent vector that had generated a MR image. Once the training of info-GAN and projector is complete, we use composition of generator and projector as a denoiser in plug and play ADMM [10] optimizer. After few iteration, around ten the output of the ADMM optimizer is the reconstructed MR image.

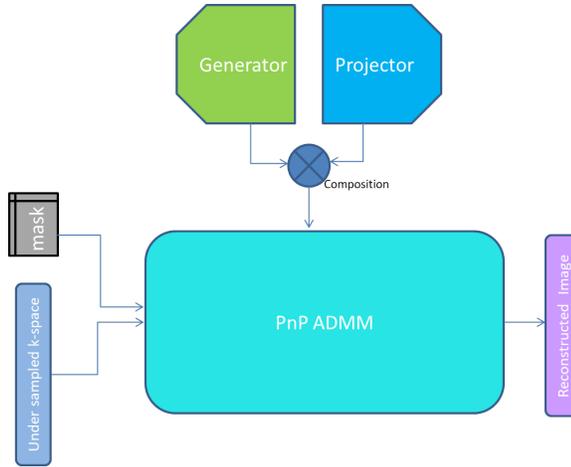


Figure 1.1: Flow of the approach

1.3 Contributions

The projector network architecture is a fully convolutional neural network unlike the fully connected network used in f-CSR algorithm. A continuous code word is used as the latent code. Since the compressed sensing on MRI images is performed in acquisition domain, we perform the least square optimization step, the first step of ADMM in frequency domain and denoising is performed in pixel domain and hence there is inter conversion from frequency to spatial domain between the first and second step of ADMM algorithm. The Least squares optimization step is not performed in vectored form of images, rather we propose a stacked way of optimization, in which we perform column wise least squares and stack all the optimized columns before being further processed.

Other deep learning methods use either a supervised approach or a unrolled iterative algorithm into a neural network or a VAE like conditional GAN, conditioned on zero filled reconstruction, this method reconstructs directly from under-sampled k-space measurements and structured latent space without the need for zero filled reconstruction in an unsupervised fashion, and is faster compared to unrolled algorithms.

The results and details of the implementation and description of the data set is presented in chapter 6, although we couldn't get perfect reconstruction, one can subjectively notice a good amount of pattern being reconstructed. we are restricted by the quality of images being generated by the generator. Most of the reconstructed SSIM scores for compression ratios 1.33, 2 and 2.66 are above 0.6, sometimes goes upto 0.8, in rare cases it goes to around 0.5.

Chapter 2

Literature Survey

2.1 Compressed sensing using generative model

This method was proposed by Bora et al [6], in which instead of employing a sparsity constraint and optimizing to find a reconstruction they used a generative models VAE and GAN. The proposed CSGM algorithms reconstructs the image by performing gradient descent to optimize the latent space variable \mathbf{z} , so that the corresponding image $G(\mathbf{z})$ has minimum measurement error $\|\mathbf{A}G(\mathbf{z}) - \mathbf{y}\|^2$.

A fully connected neural network architectures with vectored inputs were used, which might not be efficient for large size MRI data of 256×256 . Moreover, though they are using a constraint on norm of latent vector, the latent space is still entangled to generate a proper reconstruction. This work can provide some inspiration to our algorithm except for constraining the optimization objective based on norm of the latent word.

2.2 Deep De-Aliasing GAN

DAGAN a refinement learning based U-net inspired generator is proposed in [7] for CS-MRI reconstruction. Which is an end to end network that reduces the artifacts in zero filled reconstruction of under-sampled k-space data. This method also augments content based loss to the standard GAN loss, the content based loss include pixel domain and frequency domain MSE and perceptual loss.

It uses a conditional GAN paradigm conditioned on ZFR images, and thus similar to a auto-encoder architecture. DAGAN needs to be trained with a data-set containing both true MRI images and ZFR images, which may acts as a bottle neck if the data is huge, especially when the data is huge it takes a lot of storage space and makes it difficult to use when the

memory requirement is constrained.

2.3 GAN with cyclic loss

Similar to DAGAN, this method generates full MRI image from its corresponding zero filled reconstruction, but it proposes a two fold chain networks called reconGAN and refine-GAN [9] with a cyclic data consistency loss. It is a variant of convolutional auto-encoder and GAN with fully residual connections.

This model came up with a novel loss, cyclic data consistency loss in order to overcome the extreme case problem of wrong mapping from ZFR to any existing fully reconstructed image, although this model works very well the implementation complexity and memory requirements are huge.

2.4 Structure Preserving CS-MRI reconstruction

This method [8] uses a zero filled reconstruction conditioned GAN with multiple residual in residual dense blocks and uses a patch based discriminator, instead of discriminating the whole image. The model also incorporates SSIM loss and pixel domain l_1 loss to the Wasserstien GAN loss.

The interesting things about this method is it uses structure preserving components such as patch based discriminator and SSIM loss augmentation. But the network architecture is too deep with multiple residual in residual blocks that is difficult to train in computationally and storage constrained environments.

2.5 Fast compressed sensing recovery using generative networks

This project is the extension of this f-CSR algorithm [1] to CS-MRI modality. It uses a composition convolutional GAN based generator models and a fully connected projector network as a denoiser in a plug and play ADMM optimizer. This algorithm claims best results with information maximization GAN due to its disentangled latent space.

This method is particularly attractive because of its complete unsupervised nature, without any requirement for a zero filled reconstruction for each point in the data set and simplicity in principle and implementation.

Chapter 3

Information Maximizing GAN

3.1 Introduction to GANs

Generative Adversarial Networks(GANs) are a deep unsupervised generative models, which try to generate or mimic training data like objects generally from a low dimensional noisy latent vector. The basic architecture of GANs contain a Generator deep neural network and a discriminator network, the two networks play a min-max adversarial game in order to learn and generalize the underlying true data distribution i.e. Discriminator will try to do its best to properly discriminate the true data points from generated(fake) data points, On the other hand generator will try its best to fool the discriminator network.

A properly learned GAN's generator can produce output similar to but not same as data points. The output samples are also representative of the underlying factors of variation in the training distribution, for example digits with unseen strokes faces with unseen poses etc. GANs are useful for augmenting the data-set to make it more exhaustive and expand it in a useful manner, they are used to learn the statistics of data distribution from few samples of data in an unsupervised way without any parametric assumptions, they are used for super-resolution, they are used to generate photo realistic images and cartoon characters and for many other impressive and useful applications.

3.2 GANs learn the underlying data distribution

The generator network and discriminator network play two player min-max game and the two networks are adversaries to each other. This kind of competitive framework enables GAN to learn the data distribution. The standard GAN loss and learning is described by

the following equation,

$$\min_G \max_D V(D, G) = \min_G \max_D \mathbb{E}_{x \sim p_{data}} [\log D(x)] + \mathbb{E}_{z \sim p_z} [\log(1 - D(G(z)))] \quad (3.1)$$

Discriminator(D) will try to maximize the log likelihood of binary classification problem, Generator(G) tries to minimize the log-probability of its samples being classified as 'fake' by the discriminator(D), thus promoting the adversarial game.

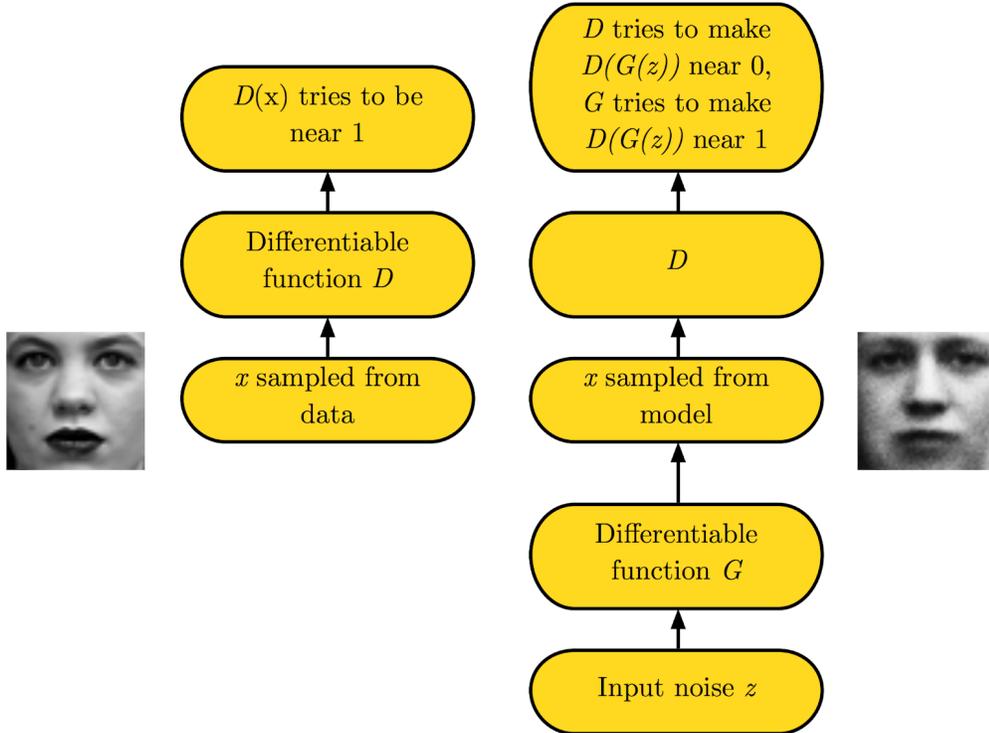


Figure 3.1: Flow-chart depicting the adversarial learning in GANs [11]

The learning is performed through mini-batch stochastic descent, for each passage of single update iteration for k-steps the discriminator is updated with both generated data and true data. after few discriminator update steps the generator is updated. The number of discriminator update steps(k-steps) is a hyper-parameter, which is usually set to one.

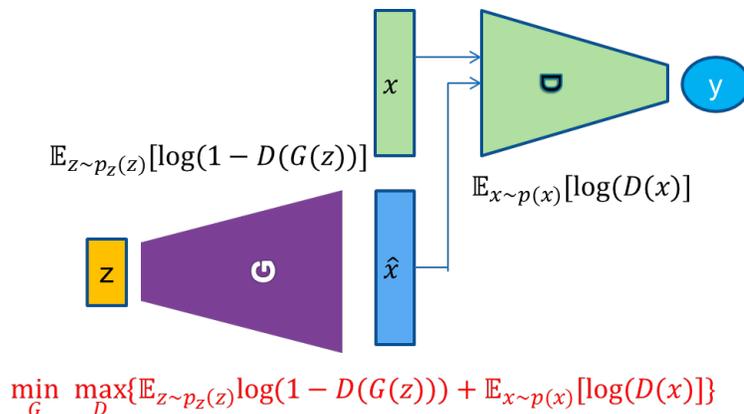


Figure 3.2: Generative Adversarial Network Model: The expression for standard GAN loss function is shown in Red and the expressions in black show the binary cross entropy loss for each network

Algorithm 3.1: GAN training algorithm using Mini-batch SGD

- 1 **for** each training iteration **do**
- 2 **for** k steps **do**
- 3 Sample m noise samples $\{z^1, \dots, z^m\}$ from noise prior $p_g(z)$
- 4 Sample m training data examples $\{x^1, \dots, x^m\}$
- 5 Update discriminator by ascending its gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^i)] + [\log(1 - D(G(z^i)))]$$

- 6 **end**
- 7 Sample m noise samples $\{z^1, \dots, z^m\}$ from noise prior $p_g(z)$
- 8 update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^m [\log(1 - D(G(z^i)))]$$

- 9 **end**
-

However in practical implementation aspects a tweak to loss function and update is performed, where instead of using the GAN loss as it is, the labels are flipped for generated images to fool the discriminator.

3.3 Disentangling the representations using information maximizing GAN

The standard GAN model doesn't promote control of output with random noise input, the input and output of the generator are entangled. This problem can be solved by Information maximizing GAN or Info-GAN, in short. Info-GAN is an information theoretic extension[12] to GAN framework, in which the control variables are introduced along with noisy latent space and control variables are learned automatically by the model.

Apart from Generator and discriminator, the Info-GAN uses a auxiliary network that predicts the control variables that generates a particular instance of modeled data. The auxiliary network learns the control codes through minimizing the mutual information based loss function.

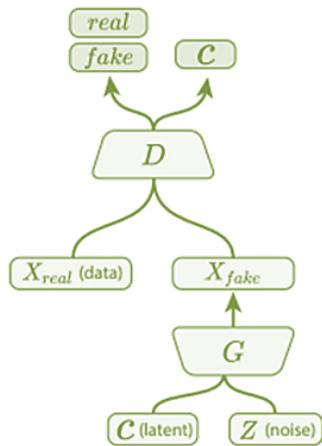


Figure 3.3: Info-GAN modality: C is the latent code word, Z is the noise latent variable (flow: bottom to top)

3.3.1 Encouraging the mutual information maximization

Maximization of mutual information between the control codes and the generated images is an important trick for learning meaningful representations. Mutual information between two random variables quantifies the shared information content of the two variables, Intuitively it is the reduction in entropy of a variable given the knowledge of another random variable.

$$I(c; G(z, c)) = H(c) - H(c|G(z, c)) \quad (3.2)$$

The equation (3.2) gives the expression for mutual information required for the case of Info-GAN, $H(\cdot)$ represents the entropy, $G(\cdot)$ represents the Generator function, z and c are the latent noise vector and code vector respectively

Combining (3.2) and (3.1), we can reformulate the Info-GAN min-max game as

$$\min_G \max_D V_I(D, G) = \min_G \max_D V(D, G) - I(\hat{c}; G(z, c)) \quad (3.3)$$

Calculating true mutual information is often intractable[13], techniques such as variational information maximization is used in [12], in order to train the generator through Mutual Information maximization a auxiliary network is introduced, which shares the same weights as that of discriminator, to interpret the generated image, but with an extra overhead that predicts the control code that generated the image.

The variational information maximization simplifies the (3.3) to training auxiliary model, as shown by the following equations. Finding $I(c; G(z, c))$ directly is not possible since it requires the posterior $P(c|x)$ in prior which is not possible, instead a lower bound is obtained by an approximating auxiliary distribution $Q(c|x)$.

$$\begin{aligned} I(c; G(z, c)) &= H(c) - H(c|G(z, c)) \\ &= \mathbb{E}_{x \sim G(x, c)} [\mathbb{E}_{\hat{c} \sim P(c|x)} [\log P(\hat{c}|x)]] + H(c) \\ &= \mathbb{E}_{x \sim G(x, c)} [\underbrace{D_{KL}(P||Q)}_{\geq 0} + \mathbb{E}_{\hat{c} \sim P(c|x)} [\log Q(\hat{c}|x)]] + H(c) \\ &\geq \mathbb{E}_{x \sim G(x, c)} [\mathbb{E}_{\hat{c} \sim P(c|x)} [\log Q(\hat{c}|x)]] + H(c) \end{aligned} \quad (3.4)$$

Using the above variational information maximization technique we can describe the Info-GAN min-max game (3.3) as

$$\begin{aligned} L_I(G, Q) &= \mathbb{E}_{x \sim G(x, c)} [\mathbb{E}_{\hat{c} \sim P(c|x)} [\log Q(\hat{c}|x)]] + H(c) \\ \min_{G, Q} \max_D V_{InfoGAN}(D, G, Q) &= V(D, G) - \lambda L_I(G, Q) \end{aligned} \quad (3.5)$$

In practical implementation, the auxiliary distribution is parameterized as a Neural Network, called Auxiliary Network or Q network. In most cases the Q network shares all the weights with the discriminator, except for a last fully connected layer which predicts the output. The addition of this network doesn't increase the computational cost of GAN model[12].

3.3.2 The control codes

The type and distribution of the control codes play crucial role in learning the class of semantic features pertaining to the data, there are two main types of control codes used, one is categorical control variables and the other one is continuous control variables. The mutual information based loss for training depends on the type and distribution of the control codes.

Categorical control codes

This may be used to select the class or type of generated image, for example in MRI brain images it can control the slice type. The codes is a one hot encoded vector with length of the code equal to number of classes. we do not explicitly choose the code, instead we give a randomly sampled batch of codes and the model learns to map them automatically, although not in any particular order of encoding. For categorical codes the Q model also should output a one hot encoding, which is enabled by using a softmax non-linearity in the output layer of auxiliary model.

The loss function can be directly used from mutual information maximization, according to (3.5), but it can be simplified as a categorical cross entropy between the predicted code and input code, since the code entropy is drawn from a known distribution and it is usually a small constant value and lambda is used to scale the mutual information loss function, which can be set to one.

Continuous control codes

These type of codes may control the style of images generated, usually the codes are sampled from uniform or Gaussian distribution, the output of the Q network tries to predict the mean and variance of the distribution, for mean, a linear output is used and for variance a sigmoid function is used, since it is a positive value.

The mutual information loss function for optimization is inferred from the predicted mean and variance. Alternatively, in the case of Gaussian there is no need for this sophistication, instead we can use the mean squared error loss with linear output.

The algorithm for training the Info-GAN is presented in 3.2

3.3.3 Algorithm and implementation

Algorithm 3.2: Info-GAN training algorithm

```
1 for each training iteration do
2   for  $k$  steps do
3     Sample  $m$  latent samples  $\{z^1, \dots, z^m\}$  from noise prior  $p_g(z)$  and control
       codes
4     Sample  $m$  training data examples  $\{x^1, \dots, x^m\}$ 
5     Update discriminator by ascending its gradient:
       
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x)] + [\log(1 - D(G(z)))]$$

6   end
7   Sample  $m$  latent samples  $\{z^1, \dots, z^m\}$  from noise prior  $p_g(z)$  and control codes
8   Update the generator weights by descending its stochastic gradient and Q
       network head weights:
       
$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log(1 - D(G(z)))]$$

9   The loss update for Q network head is based on type and distribution of control
       code being used.
10 end
```

3.4 Summary

GANs learn the underlying data distribution from the samples presented to them through a mini-max adversarial game between generator and discriminator network. An improved version of GAN that can generate the data points in a disentangled fashion is the Info-GAN model, which learns the mapping between latent space to output via a information maximization scheme, fundamentally it tries to predict the code corresponding to the generated image through a network bifurcating from discriminator called as auxiliary network. This Info-GAN network is used as the generative model in this project which has maximum structure with respect to information between generated images and latent space in its latent space.

Chapter 4

Alternating Direction Method of Multipliers

Most important optimization problems, such as LASSO regression combine two terms

$$\text{minimize } F_1(\mathbf{x}) + F_2(\mathbf{x}) \text{ for } x \text{ in a convex set} \quad (4.1)$$

F_1 involves l^1 type norm and F_2 involves l^2 type norm, though both are convex, they don't mix well, i.e. the optimization doesn't occur among the two terms in a balanced way. The solution for this problem is to split the algorithm and alternate between the two. Split algorithms are more efficient than mixed approach[14].

One such split algorithm is ADMM(Alternating direction method of multipliers), the main forward steps that describe this algorithm are dual decomposition, augmented Lagrangian or method of multipliers.

The flow starts from the primal problem and its Lagrangian, next goes to dual problem and dual decomposition of it, and then comes the augmented Lagrangian which makes the Lagrangian more robust and ends at ADMM.

4.1 Primal to Dual

The primal problem is

$$\text{Minimize } f(\mathbf{x}) \text{ subject to } \mathbf{Ax} = \mathbf{b} \quad (4.2)$$

Now the primal problem is combined with the constraint and optimized using Lagrangian multiplier \mathbf{y} . The Lagrangian of above problem is

$$L(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) \quad (4.3)$$

The optimal points \mathbf{x}^* and \mathbf{y}^* form the saddle points of (4.3): $\min_x \max_y L$, The key is to separate the above into primal and dual problem, *Minimize* $L(\mathbf{x}, \mathbf{y})$ over \mathbf{x} and *Maximize* $m(\mathbf{y}) = L(\mathbf{x}^*(\mathbf{y}), \mathbf{y})$ respectively.

The primal update and dual update can be sequenced using steepest ascent as shown:

$$\begin{aligned} \mathbf{x}_{k+1} &= \operatorname{argmin} L(\mathbf{x}, \mathbf{y}_k) \\ \mathbf{y}_{k+1} &= \mathbf{y}_k + s_k(\mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}) \end{aligned} \quad (4.4)$$

In short the above equation is, find \mathbf{x}_{k+1} using some initial dual variable \mathbf{y}_k and then follow(ascend) $\nabla m = \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}$ with a step size s_k , and iterate this sequence until convergence.

4.2 Dual Decomposition

Instead of performing one large minimization of (4.3), multiple mini-optimizations can be performed if the objective function $f(\mathbf{x})$ is separable, i.e if it can be decomposed into $f_1(\mathbf{x}_1) + \dots + f_N(\mathbf{x}_N)$ where \mathbf{x}_i 's are sub-vectors of \mathbf{x} . We can partition the columns of $\mathbf{A} = [A_1 \dots A_N]$ in a similar fashion. The Lagrange's function now decomposes into N smaller Lagrangian s:

$$f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = \sum_1^N L_i(\mathbf{x}_i, \mathbf{y}) = \sum_1^N [f_i(\mathbf{x}_i) + \mathbf{y}^T A_i \mathbf{x}_i - \frac{1}{N} \mathbf{y}^T \mathbf{b}] \quad (4.5)$$

Now these N-minimizations can be solved in parallel, the decomposed dual problem with N dual problems solved in parallel.

$$\begin{aligned} \mathbf{x}_i^{k+1} &= \operatorname{argmin} L_i(\mathbf{x}_i, \mathbf{y}^k) \\ \mathbf{y}^{k+1} &= \mathbf{y}^k + s_k(\mathbf{A}\mathbf{x}^{k+1} - \mathbf{b}) \end{aligned} \quad (4.6)$$

The new \mathbf{x}_i^{k+1} are concatenated into $(\mathbf{A}\mathbf{x}^{k+1})$, then the resulting \mathbf{y} is distributed into N processors that execute separate minimizations in the next iteration of (4.6), which gives an enormous time advantage compared to single large minimization.

4.3 Augmented Lagrangian

To relax strict convexity on $f(\mathbf{x})$ and make the Lagrangian robust, it is penalized by augmenting it with a penalty term scaled by a scalar ρ .

$$L_\rho(\mathbf{x}, \mathbf{y}) = f(\mathbf{x}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

The augmented Lagrangian is also known as method of multipliers, this improves the convergence guarantees.

The main drawback of this augmented Lagrangian is the penalty term makes the Lagrangian inseparable even if the objective function is separable, which otherwise could give a performance advantage.

4.4 ADMM

The drawback of augmented Lagrangian can be avoided by using a split of $f(\mathbf{x})$ (possibly a $l^1 - l^2$ split), and substituting a new variable \mathbf{z} in one of the split with a new constraint $\mathbf{x} = \mathbf{z}$. The new constraint joins the original and forms $\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} = \mathbf{0}$, and the optimization objective is $f(\mathbf{x}) + g(\mathbf{z})$

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}} f(\mathbf{x}) + g(\mathbf{z}) \\ \text{Such that : } \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} = \mathbf{0} \end{aligned} \tag{4.7}$$

The augmented Lagrangian is

$$L_\rho(x, z, y) = f(\mathbf{x}) + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c}\|^2 \tag{4.8}$$

which is scaled to a simple form by using a dual variable \mathbf{y} scaled by $\frac{1}{\rho}$, thus reducing the augmented Lagrangian to, which leads to scaled ADMM[15]

$$\begin{aligned} L_\rho(\mathbf{x}, \mathbf{z}, \mathbf{y}) = f(\mathbf{x}) + g(\mathbf{z}) + \frac{\rho}{2}\|\mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{z} - \mathbf{c} + \mathbf{u}\|^2 \\ ; \text{ where, } u = \frac{\mathbf{y}}{\rho} \end{aligned}$$

Finally, the ADMM update steps are following, which include extra \mathbf{z} variable update apart from primal and dual update, which is called as auxiliary update:

$$\mathbf{x}^{(k+1)} = \underset{x}{\operatorname{argmin}}(f(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{Ax} + \mathbf{Bz}^{(k)} - \mathbf{c} + \mathbf{u}^{(k)}\|^2) \quad (4.9)$$

$$\mathbf{z}^{(k+1)} = \underset{z}{\operatorname{argmin}}(g(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{Ax}^{(k+1)} + \mathbf{Bz} - \mathbf{c} + \mathbf{u}^{(k)}\|^2) \quad (4.10)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{Ax}^{(k+1)} + \mathbf{Bz}^{(k+1)} - \mathbf{c} \quad (4.11)$$

ADMM can achieve acceptable accuracy quickly, but to reach high exact accuracy in could be slow and take time.

4.5 ADMM for Image recovery

The image recovery problem as formulated in B.1 can be solved using the ADMM method. The set of equations for using ADMM for image reconstruction are:

$$\underset{x}{\operatorname{argmin}} F(\mathbf{x}) + \lambda G(\mathbf{x}) \quad (4.12)$$

$$\underset{x,z}{\operatorname{argmin}} F(\mathbf{x}) + \lambda G(\mathbf{z}) ; S.T. \mathbf{x} = \mathbf{z} \quad (4.13)$$

Where $F(\cdot)$ is a fidelity term and $G(\cdot)$ is a prior or constraint that signal must satisfy, for example it could be the l^1 -norm for a sparsity prior in traditional compressed sensing reconstruction.

The augmented Lagrangian for the above problem 4.13 is

$$L(\mathbf{x}, \mathbf{z}, \mathbf{u}) = F(\mathbf{x}) + \lambda G(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z} + \mathbf{u}\|^2 \quad (4.14)$$

and the ADMM updates are:

$$\mathbf{x}^{k+1} = \underset{x}{\operatorname{argmin}}(F(\mathbf{x}) + \frac{\rho}{2} \|\mathbf{x} - \mathbf{z}^k + \mathbf{u}^k\|^2) \quad (4.15)$$

$$\mathbf{z}^{k+1} = \underset{z}{\operatorname{argmin}}(\lambda G(\mathbf{z}) + \frac{\rho}{2} \|\mathbf{x}^{k+1} - \mathbf{z} + \mathbf{u}^k\|^2) \quad (4.16)$$

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{x}^{k+1} - \mathbf{z}^{k+1} \quad (4.17)$$

Here the update step one is an inversion step since the $F(x)$ is the fidelity term, a norm error; the second step can be seen as denoising step, in which we are finding a z close to noisy x given by $x + u$. This modular structure makes ADMM favorable to use a variant in which it is not required to mention $G(z)$ ahead of running the ADMM iterations and instead replace the second step with an off the shelf denoising algorithm $\mathbf{z}^{(k+1)} = D_\sigma(\mathbf{x}^{k+1} + \mathbf{u}^k)$ this variant is called as plug and play ADMM[10], which is used for image restoration applications such as Denoising, and Super-resolution.

Pros and Cons of Plug and Play ADMM

Although the Plug and Play ADMM is of ad-hoc nature, empirically it is observed to perform better than some state of the art methods.

The main challenge is imposed on theoretical analysis, since the denoiser \mathcal{D}_σ is most of the times non-linear and doesn't have closed form expression. For a general \mathcal{D}_σ convergence is not known, since it requires the prior $G(\cdot)$ to be convex and closed to ensure proper convergence. But for off the shelf denoiser \mathcal{D}_σ one can not point out a corresponding prior $G(\cdot)$, and also it is unclear if it can be derived or found.

4.6 Summary

ADMM points out a way to effectively optimize problems that involve unbalanced objective that can be split into separate objectives, for example LASSO regression. It makes use of dual decomposition and augmented Lagrangian for optimizing the problem of interest. A plug and play nature of ADMM was proposed for image recovery applications in which one can plug in a off the shelf denoiser into the second step, although it seems to be of an ad-hoc nature, it is observed to provide descent results for image recovery problems.

Chapter 5

Methods and Contributions

5.1 Data set

The data set which was used for training and testing is MICCAI grand challenge 2013 T1 weighted MRI brain images data set, which contain subject wise files in ".nii.gz" (NifTI) format [16] each containing around 265 slices (sliced along transverse plane).

For extracting the image data from NifTI format, a python Neuro-imaging library, "Nifabel" [17] was used and the extracted image data is dumped into a pickle format, first hundred files are selected from the entire folder and from each 160 useful and informative slices were taken. The training data set contains 16000 images (i.e. 160×100), and a similar method is followed for collecting testing images from a separate training images folder. While converting from ".nii.gz" format the image gray scale is normalized to the range $[-1, 1]$.

At each iteration of training a batch of 32 slices are selected at random from among 160 slices and a file is chosen at random from a set of 100 files without replacement. Each epoch is trained with 30 batches of data.

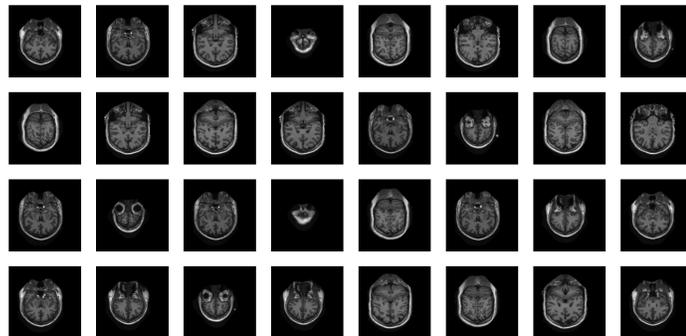


Figure 5.1: A sample of MICCAI dataset images

5.2 Latent Space Design

A infoGAN or GAN maps a latent noise space to the data to be generated. In an InfoGAN setup since information maximization is performed over code vector, the latent space is the concatenation of noise vector and code vector or vectors.

Different dimensions of latent vectors are tested and empirically it was found that a length of 256 dimension code provides a better performance. If the dimension is too small it gets difficult to train the network and if it is too large there are problems with projector network learning.

A 256 dimension latent vector is used with first 192 dimension being noise and remaining 64 dimensions being the code word. Each component of noise is sampled from a standard normal distribution and the code vector used is of continuous nature, each component is sampled according to uniform distribution of range $[-1, 1]$.

A discrete code was tested with 160 dimension one hot encoding, assuming the discrete feature to be the slice index, but it is found in the data that there are no exact correspondence of slices across files. Hence this did not work well for information maximization and moreover a discrete type feature is not easily found apart from the slice index, therefore we used continuous codes without any assumption of code and image correspondence. Another possible way that was hypothesized, but not tested is to design the code according to the distribution of first 64 principle component coefficients.

5.3 Info-GAN: generation of data like samples

The generator of the info-GAN maps a 256 dimensional latent space to 256×256 gray scale image, which is a brain like MRI image after training.

The generator architecture is shown in the figure 5.2, it contains all transposed convolutional layers, with kernel size 4×4 , each is followed by ReLU non-linear activation and batch normalization, except the last layer which is activated with Tanh non-linearity. First layer is convolved with stride 1 and padding 0, remaining layers are convolved with stride 2 and padding 1.

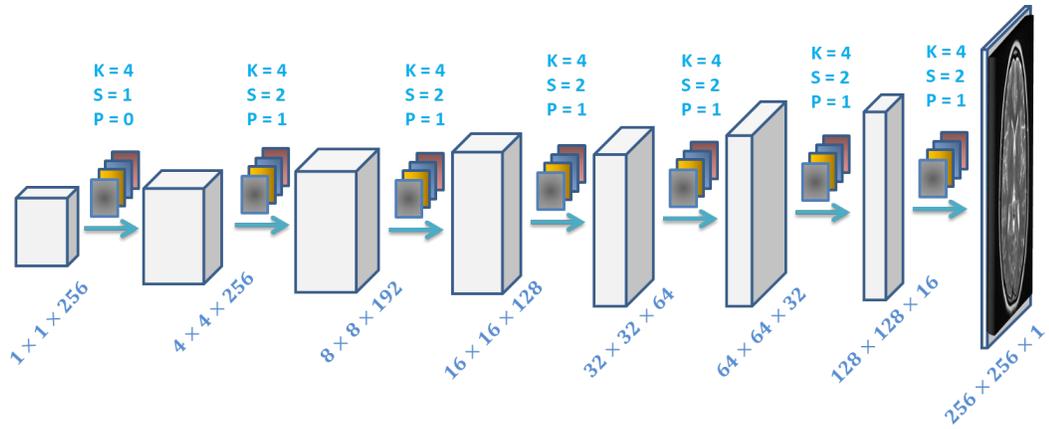


Figure 5.2: Generator Network: Each is followed by a ReLU non-linear activation and Batch Normalization, except for the last layer which uses Tanh activation layer. (Size of output mentioned below each layer, and kernel(k), stride(s) and padding(p) mentioned between the layers)

The discriminator is a classification network, that tries to distinguish between true images and generated images, this uses all convolutional layers with kernel size 4×4 , for last but one layer the kernel size is 1×1 , each is followed by Leaky-ReLU activation and then a Batch Normalization, except the last layer which is with sigmoid non linearity. All the layers use stride of 2 and padding of 1, last but one layer and final layer uses stride of 1 and no padding.

The discriminator is bifurcated into two, one is for classification as discussed above and other branch is connected to a auxiliary network, that learns to predict the code word that had generated the image.

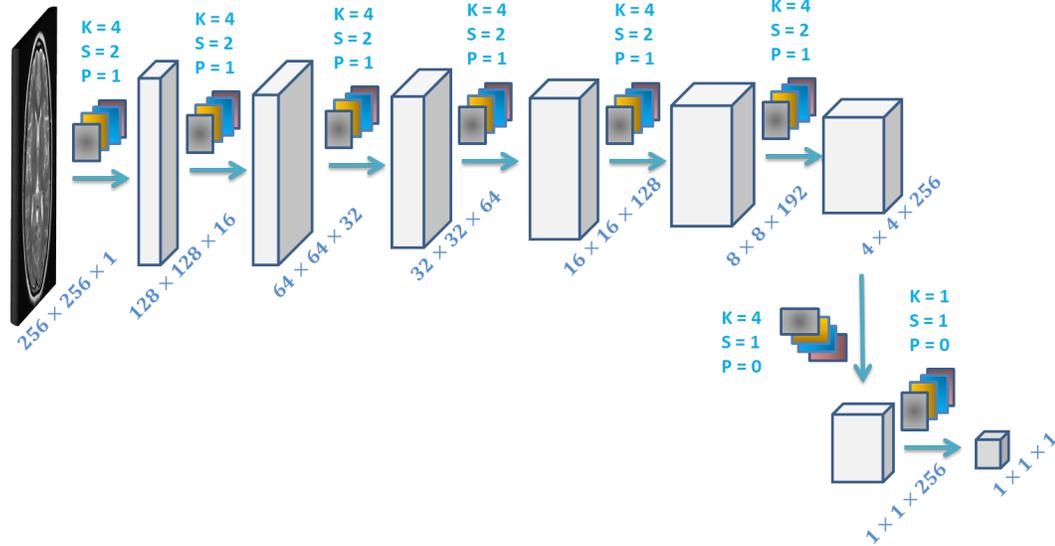


Figure 5.3: Discriminator Network: Each followed by Leaky-ReLU and a Batch Normalization, except for last which used sigmoid activation

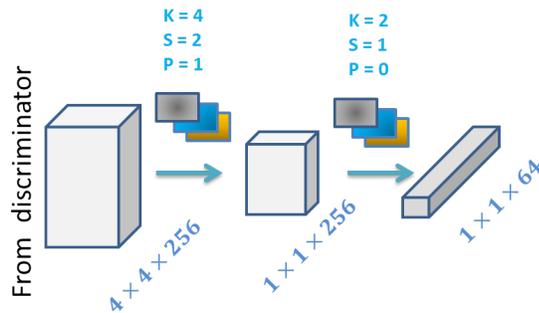


Figure 5.4: Auxiliary network: bifurcates from sixth layer of discriminator, this network predicts the latent code that had generated the image. The intermediate layer is followed by ReLU and Batch Normalization, and last layer is a linear

Training the Info-GAN

The GAN learns through optimizing the standard GAN loss 3.1, and along with stepping the gradient through generator, the auxiliary network is trained, the loss function used for optimizing the auxiliary network is Mean squared Loss between the output and latent code that generated the image. The classic algorithm for training the info-GAN is present in the algorithm 3.2.

The training is carried out with multiple restarts for first 100 epochs a learning rate of 0.002 was used, then for next 20 epochs a learning rate of 0.0001, for next 20 epochs a learning

rate of 1×10^{-5} and final 30 epochs are trained with a learning rate of 5×10^{-6} . ADAM optimizer is used for both GAN and auxiliary network gradient descent with $\beta_1 = 0.5$ and $\beta_2 = 0.999$.

The plot of generator, discriminator and auxiliary network loss function Vs training iteration is shown in the following figures 5.5 and 5.6

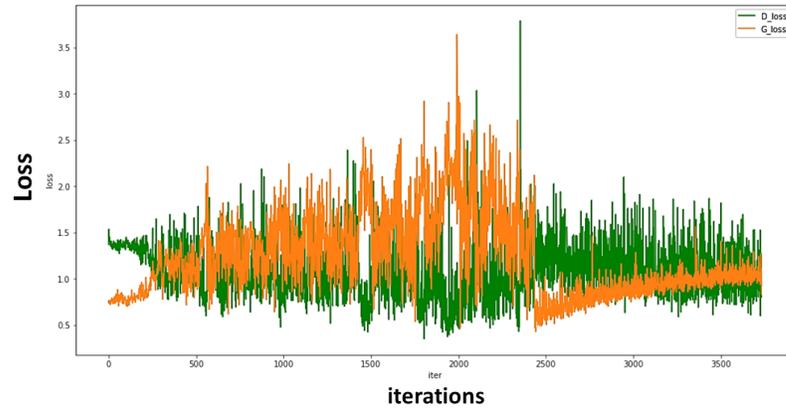


Figure 5.5: Discriminator loss(green) and Generator loss(orange) Vs iteration plot: An adversarial game between generator and discriminator can be observed here. The loss is a binary cross entropy loss

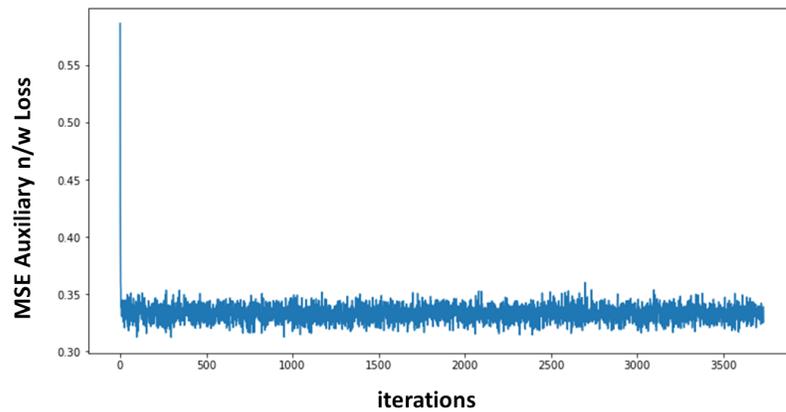


Figure 5.6: Auxiliary Network Loss function plot Vs iteration

Generated images after training the info-GAN are shown in the following figure 5.7

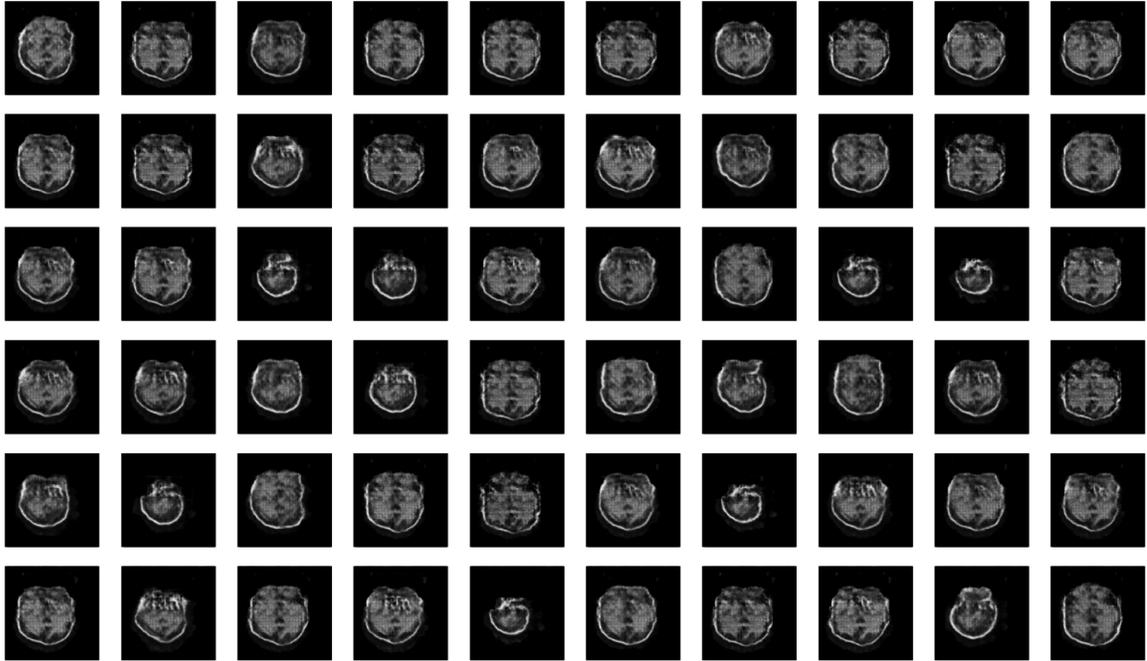


Figure 5.7: Sample of generated images after the training

5.4 Projector Network: Constraining the latent space

The projector network learning the mapping between MRI image and the corresponding 256 dimensional latent code, The architecture is almost the reverse of generator architecture just with a small change in number of feature maps in each layer. The architecture of projector network is depicted in the figure 5.8.

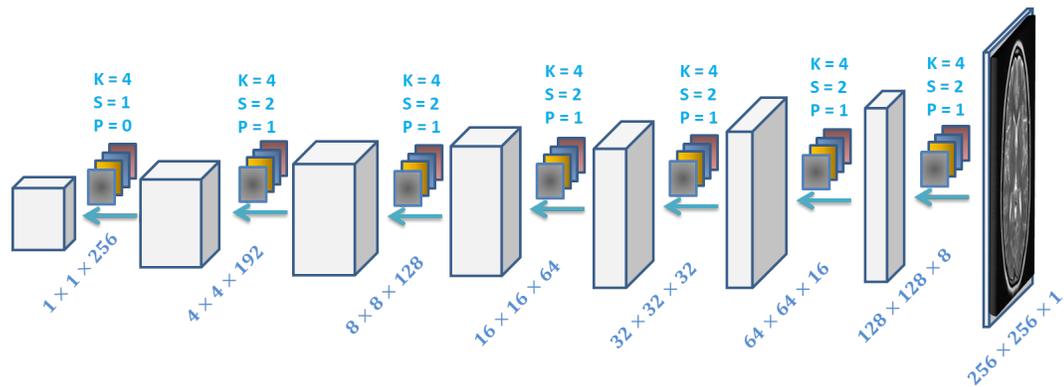


Figure 5.8: The projector network: maps the image to latent code. (from right to left)

Training the projector network

The objective cost function is mean squared error loss between the predicted latent variable and the true latent variable that had generated the image. ADAM optimizer with default beta values is used for gradient descent. For first 150 epochs a learning rate of 0.0001. was used and for final 50 epochs of training the learning rate is set to 10^{-4} .

First projector training modality in f-CSR algorithm [1] is used 5.9, in which a latent variable is sampled from its distribution and passed through the trained generator, and then the output of generator along with a small noise added, acts as an input the projector network which try to minimize the mean squared error between predicted latent variable and true latent variable.

The projector loss function plot is given in the following figure 5.10

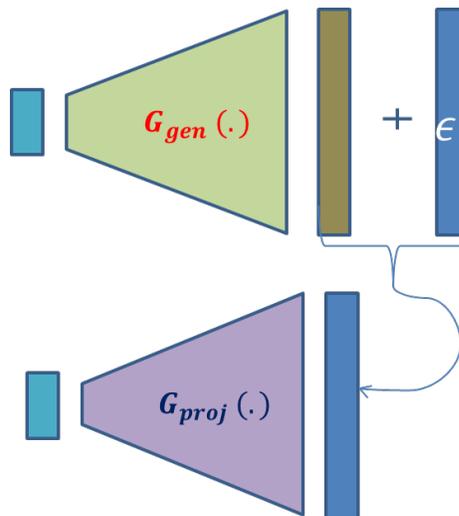


Figure 5.9: Projector training modality: The projector will try to find the latent variable close to the latent variable that generated a particular image, in MSE sense

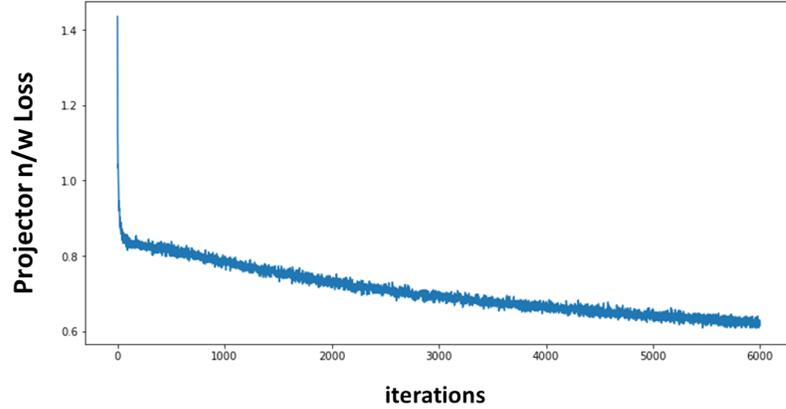


Figure 5.10: Projector loss Vs Training iteration: MSE loss function is being optimized

5.5 Under-sampling of k-space: CS-MRI

Under-sampling in k-space domain is performed using a linear mask, i.e. the mask randomly selects few rows from among 256 rows. The number of rows that are randomly selected based on the compression ratio. The under-sampling masks and corresponding matrix are shown below 5.11

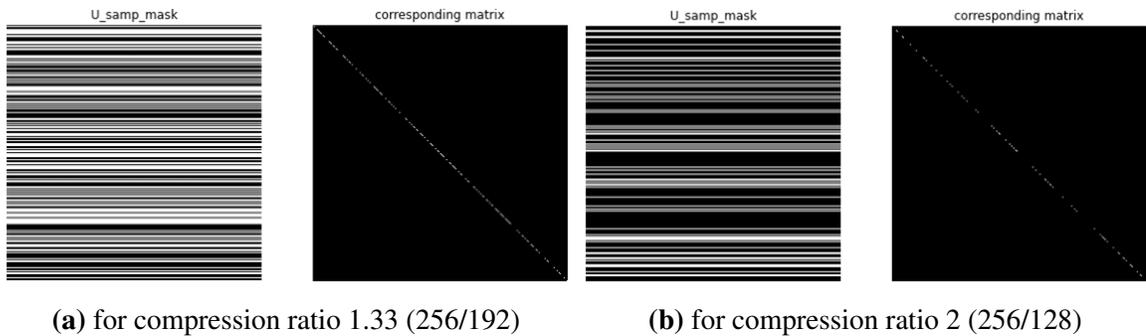


Figure 5.11: undersampling masks and corresponding matrices

5.6 ADMM+info-GAN+Projector: CS-MRI Reconstruction

After training the info-GAN and projector network, we combine the generator $G_{gen}(\cdot)$, projector $G_{proj}(\cdot)$ with ADMM 4.13 algorithm for the reconstruction of MRI Images from under-sampled k-space measurements.

The algorithm takes under-sampled k-space measurements \mathbf{y} and under-sampling matrix Φ as input, the trained generator and projector are used in the PnP ADMM iteration denoising step. Since the CS-MRI is performed in Fourier space we apply the ADMM optimization in Fourier domain and then perform inverse Fourier transform before the denoising step in the ADMM iteration. ADMM steps for CS-MRI reconstruction are shown in the figure 5.12 below.

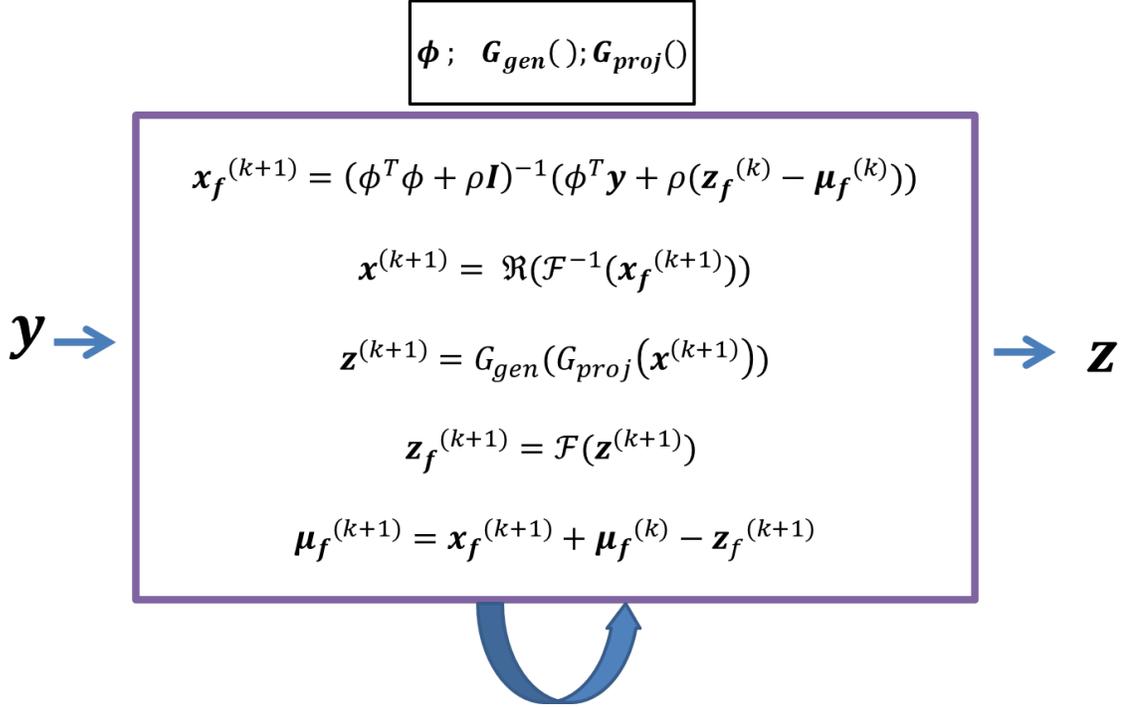


Figure 5.12: ADMM iteration for CS-MRI reconstruction

We have a trained generator $G_{gen}()$, Projector $G_{proj}()$ and the under-sampling matrix Φ , \mathbf{y} is under-sampled k-space image. First step is a least squares step performed in Fourier domain, then we pass the real part of inverse Fourier transform of the first step output to composition of generator and projector, this is the denoising step. We again Fourier transform the denoised output in previous step so that it can be used in first step for next iteration, and the final step is dual variable update. The reconstructed MR image is present in \mathbf{z} after the completion of iterations, we choose the \mathbf{z} such that its under-sampled k-space has minimum error with the measurements \mathbf{y} in a mean squared error sense. It is observed that the minimum error occurs within ten iterations of ADMM, therefore maximum number of iterations is set to 10.

The first least squares step of ADMM is performed in a column wise fashion without vectoring the 256×256 k-space image, and finally stack the optimized columns together, we

call this process as stacked optimization. The entire algorithm of info-GAN regularized ADMM for CS-MRI reconstruction is presented in the algorithm 5.1

Algorithm 5.1: info-GAN regularized ADMM for CS-MRI reconstruction

```

1 train the info-GAN according to and save the generator model
2 train the projector network according to and save projector model
3 get test images from the data set
4 generate under-sampling mask  $U$  based on required compression ratio
5 construct denoiser (i.e. composition of saved generator and projector models from
   step 1 and 2)
6 for each  $test\_image$  do
7     Function  $u_{samp}(test\_image, U)$  :
8          $k_{space} = \mathcal{F}(test\_image)$ 
9          $y = U \times k_{space}$ 
10        return  $y$ 
11         $\rho = \frac{1}{\sqrt{256}}$ 
12         $x_f^{(0)}$  = a random  $256 \times 256$  image
13         $max\_iter = 10$ 
14         $z_f^{(0)} = \text{zeros}(256 \times 256)$ , dtype = complex
15         $u_f^{(0)} = \text{zeros}(256 \times 256)$ , dtype = complex
16        for  $iter \leq max\_iter$  do
17            perform ADMM steps as in 5.12
18            store  $z^{(iter)}$  in a list
19        end
20         $\hat{z} = \underset{z}{\text{argmin}} y - u_{samp}(z, U)$ 
21 end

```

5.7 Summary

Information maximizing GAN regularized ADMM algorithm makes uses two important networks, one the generator from GAN model and a projector network. Here we used a fully convolutional projector network unlike the fully connected network in f-CSR algorithm. In this scenario convolution layers found to work well compared to fully connected layer, with fully connected layer the MSE converges and saturates at around 0.8 whereas with convolutional layers it progressively decreases and reaches upto 0.5 MSE.

The ADMM denoiser uses composition of generator and projector functions, which was derived through the formulation of regularizing the optimization problem with respect to latent variable and generator output. The results of this algorithm are described in the next chapter.

Chapter 6

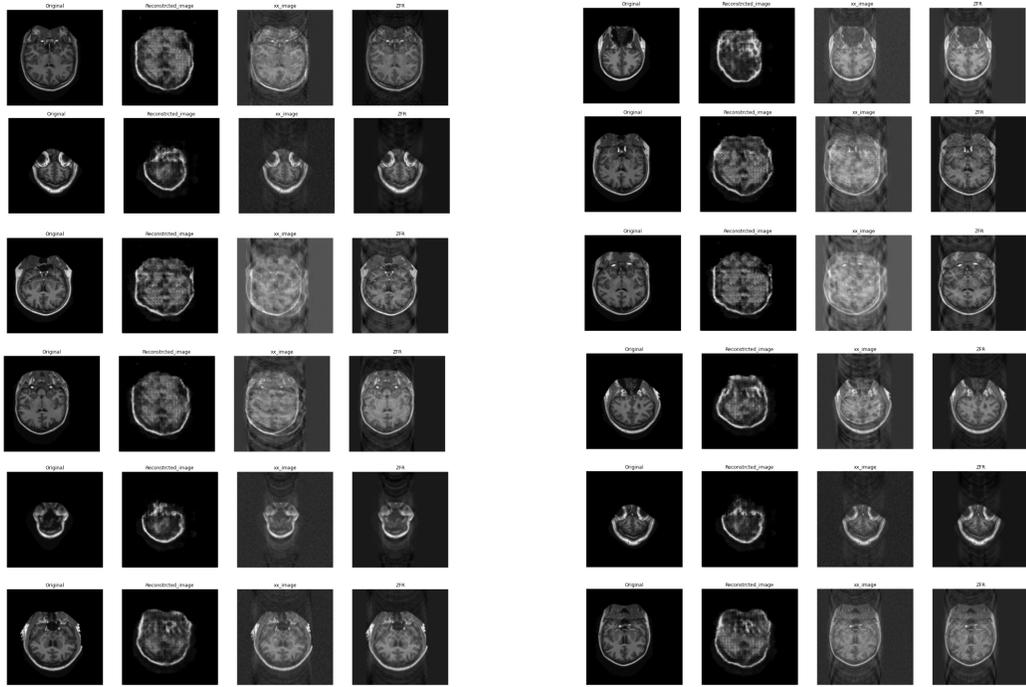
Results, Discussion and Conclusion

6.1 Results and Discussion

The reconstructed results performed for compression ratios 1.33, 2, 2.66 are presented in this section. Each result of reconstruction shown in the following figures contain four columns, the first column represent original test images, second column contain the reconstructed results, third column contain the intermediate image that corresponds to x variable in the ADMM algorithm 5.12 and last column shows the zero filled reconstructions i.e. the inverse Fourier transform of under-sampled k -space image. Along with the reconstructed image results, SSIM (structural similarity index measure) and mean squared error plots taken for five trails of reconstruction experiments are shown, each trail consists of reconstruction of randomly selected ten test images(slices) from the test data set.

The first compression ratio 1.33 corresponds to a under-sampling mask of 192 randomly selected rows, similarly compression ratios of 2 and 2.66 corresponds to under-samplings masks of 128 and 96 randomly selected rows respectively, mean SSIM metric which is performed with 11×11 Gaussian window is chosen as the best metric to evaluate the reconstruction of MR image since it takes account of local statistics and structural properties of an image unlike the global evaluation metrics such as Mean squared error and PSNR. Although the MSE metric is also evaluated to show the contrast.

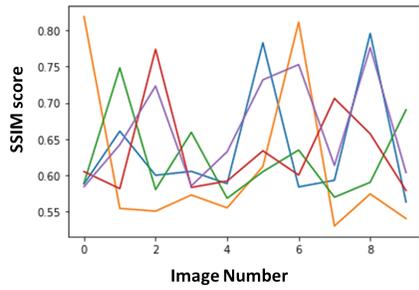
We could not objectively evaluate the image generation by info-GAN, since we do not have any explicit classes or measures to evaluate metrics like inception score, and evaluation of GANs is still an open problem and varies from application to application. the only way to evaluate the generation is to subjectively analyze the generated images. One claim is that we can not get reconstructions better than the quality of generated images, we are upper bound by the generated images quality.



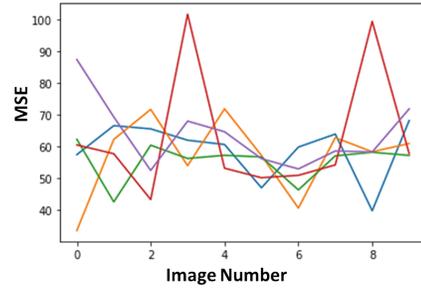
(a)

(b)

Figure 6.1: Reconstruction results with compression ratio 1.33; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR



(a) 5 snapshots of SSIM



(b) 5 snapshots of MSE

Figure 6.2: SSIM and MSE between the true image and reconstructed image for compression ratio 1.33: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric

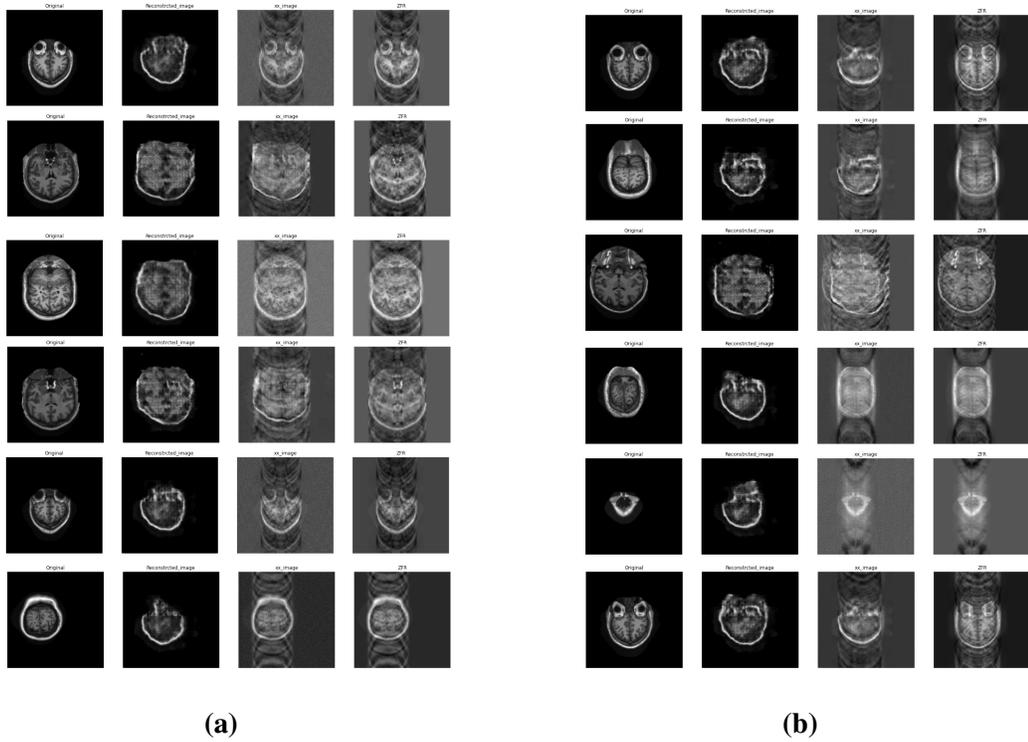
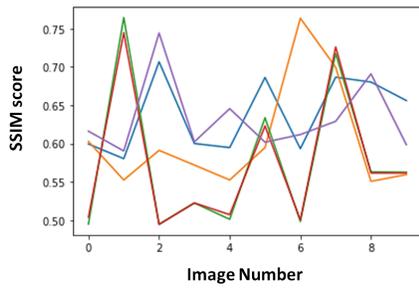
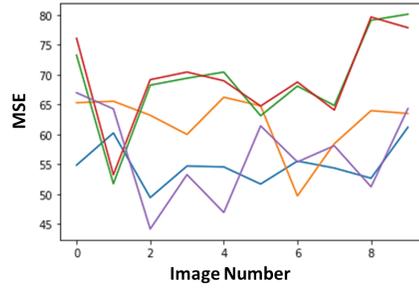


Figure 6.3: Reconstruction results with compression ratio 2; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR

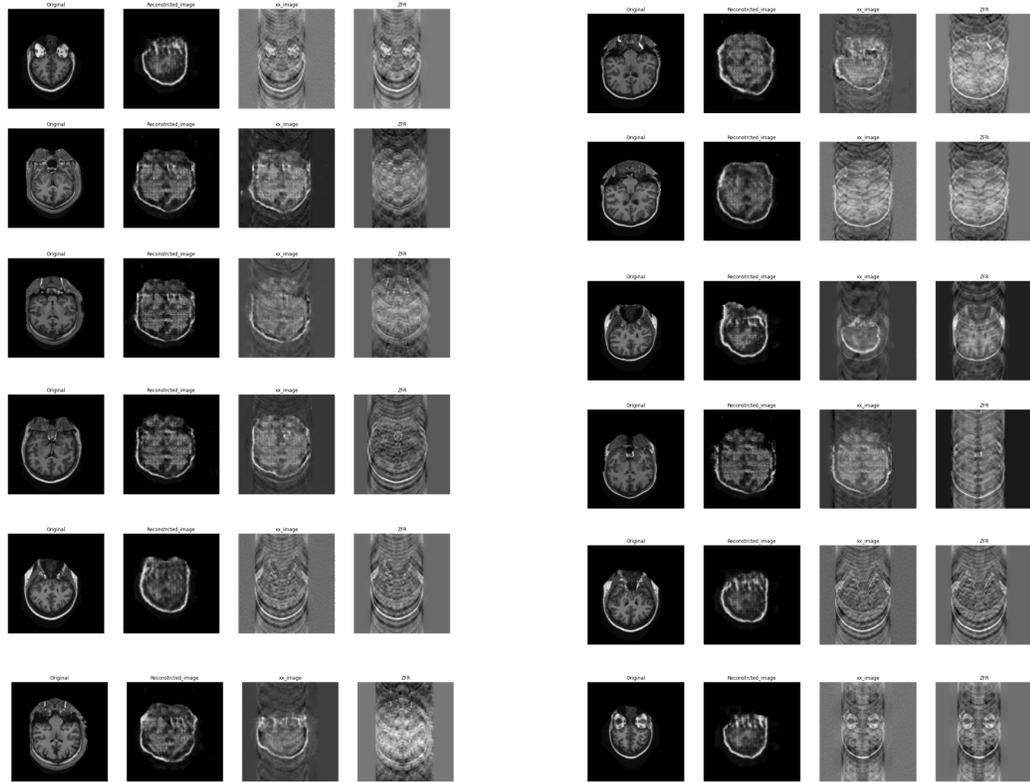


(a) 5 snapshots of SSIM



(b) 5 snapshots of MSE

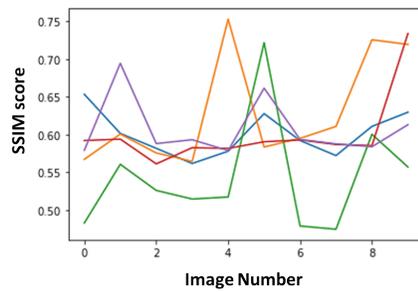
Figure 6.4: SSIM and MSE between the true image and reconstructed image for compression ratio 2: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric



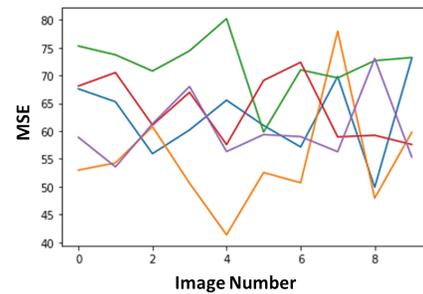
(a)

(b)

Figure 6.5: Reconstruction results with compression ratio 2.66; column 1: original, column 2: Reconstruction, column 3: intermediate image, column 4: ZFR



(a) 5 snapshots of SSIM



(b) 5 snapshots of MSE

Figure 6.6: SSIM and MSE between the true image and reconstructed image for compression ratio 2.66: each color represents a trial of reconstruction, the horizontal axis is the test image number in each trial, vertical axis shows the score of evaluation metric

6.2 Conclusion

Information maximizing GAN regularized ADMM is directly inspired from the fast compressed sensing reconstruction using generative model (f-CSR) algorithm [1], although that was applied for pixel domain compressed sensing this project has extended its application for compressing sensing in Fourier domain, since the main vision of this project is to reduce the acquisition times, therefore compressed sensing on MRI needs to be performed in acquisition domain, i.e. the k-space (Fourier domain).

There was no open access raw k-space data set found, we assumed the equivalence between the true k-space data and 2D-Fourier transform of MR image, and performed the under-sampling on the Fourier Transform image.

The reconstruction results are not so effective, although one can easily see a good pattern is learned and reconstructed. Since the effectiveness of reconstruction is constrained by the quality of generated images by the generator, i.e. the quality of reconstruction can not be better than the quality of generated images. If one could generate a good quality images the quality of results would be more and hence will lead to a SSIM score close to one for all test images.

For the above results SSIM score is above 0.55 and extends up to 0.8 for compression ratio 1.33; the SSIM score is above 0.5 and extends up to 0.75 SSIM score for compression ratio 2; and the SSIM score is above 0.45 and extends upto 0.75 for compression ratio 2.66. With increasing compression ratio the SSIM score is decreasing slightly.

In the second step of ADMM 5.12 only the real part is taken into consideration, and hence there is some loss of phase information, If we could incorporate the imaginary part, the results are expected to improve.

Only one type of CS-MRI under-sampling mask was evaluated, but there are many other types of CS-MRI under-sampling paradigms, this mask was chosen because most of the MRI acquisition is carried out in the linear scan fashion along horizontal or vertical direction and it is easier to implement and test this type of mask

Other consideration to be taken into account include the choice of evaluation metrics, theoretical bounds and recovery guarantees, which are discussed below.

Choice of Evaluation Metrics

Evaluation metrics play an important role for analyzing the results and determining whether the results justify conclusions. Similarly, in analyzing the reconstruction quality of various

compressed sensing recovery approaches it plays a key role. Often times the metrics are biased to show the efficiency over other existing methods or sometimes the metrics are not useful for certain application domains.

To overcome this, rather than only computing favorable metrics, it is important to evaluate wide variety of metrics and get an unbiased view of the results and in turn conclusions.

In this project for evaluating the results we have used SSIM measure and Mean squared error, We chose SSIM to be an important quality metric for MRI and mean squared error is often times misleading metric.

This section discusses about various metrics useful for analyzing the reconstructed signals.

Reconstruction Error

It is the average Euclidean distance between the recovered images and ground truth images, this is an average error in direct pixel domain, thus it may or may not provide a holistic metric about the results.

PSNR: Peak Signal to Noise Ratio

PSNR is a most commonly used metric for measuring the quality of reconstruction of compressed images. Higher the PSNR better is the quality of reconstruction of the image.

$$PSNR = 10 \log_{10} \left(\frac{MAX_I^2}{MSE} \right) \quad (6.1)$$

Where I is the original image and K is the reconstructed image the mean squared error(MSE) is given the average norm error between each pixel value in original and reconstructed image.

SSIM: Structural Similarity Index Measure

SSIM is a perception based model, in which degradation of image is considered as change in structural information, unlike the absolute error metrics such as PSNR and MSE, this measures the structural information, which is encoded as the statistical similarity between neighborhood pixels in an image.

SSIM is calculated over various windows (block) of an image, let say x and y are two same location windows in original and reconstructed image then the SSIM is given by 6.2

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_1)} \quad (6.2)$$

where; μ is the mean with respect to subscript,
 σ is the variance(var and covariance) of subscript variables
 c_1 and c_2 are division stability constant,to stabilize when a weak denominator occurs

NRMSE: Normalized root mean squared error

Root mean squared error is most commonly used error metric in estimation problems. It is the square root of second sample moment of the differences between observed and predicted values.

Classification accuracy or domain specific measures

Sometimes it is appropriate to use or device metrics that are very specific to application domain. for example if we take the MNIST data set the reconstruction quality can be analyzed by performing the classification task and reporting the accuracy on reconstructed images. Which is a more informed way of result analysis, but it can turn out to be difficult to device this type of metrics in many cases, and also can make comparisons with other existing work difficult, although useful for an application.

Theoretical Recovery Guarantees

Intuitively, we can't use any dimension of latent space or any deep network for reconstruction. Theoretical recovery guarantees can set the bounds for which recovery can be guaranteed. They generally involve proving the upper bound on probability of reconstruction and minimum number of required measurements dependent on the dimension of latent space given as the input to generative adversarial network.

The proof of those is usually determined using Lipchitz assumption of Deep Neural Network parametric function, Johnson Linderstrauss Lemma, Restricted Isometry property and structure of measurement matrix. Laying such theoretical bounds will make a CS-MRI reconstruction method more principled, which are not analyzed in this project

Other consideration that needs to be addressed are robustness of the learning algorithm, complexity of implementation and feasibility of deployment of GANs. they are a recent innovation and designing its hyper-parameters is more of a art than a science (it is based on lot of experience and empirical analysis by experts in this domain), and evaluating the GAN performance is still an open problem.

Although the reconstruction results of the project are not perfect enough to be used in real

time, one can see the enormous capabilities of using this method for CS-MRI reconstruction for improved speeds of reconstruction without any requirement for hand crafted basis and supervised data.

6.3 Future Work

Some of the future work, that is expected to improve the quality of reconstructions is discussed here. First regarding the info-GAN design, by looking at some networks used on or for MRI images, if we can have residual connections or skip connections across the corresponding layers of generator and discriminator, the quality of generated images might be improved, at the same time one must keep in mind during the implementation is that the generator network should be comfortably detached and used in ADMM at later stages.

Second, as discussed in latent space design section, using a principled approach such as the distribution of first n principal components, and if possible to have a kind of data set that has one to one slice correspondence across the subject so that we can have codes that has both discrete and continuous components, which can have improved effect at denoising step of ADMM.

Finally, along with above improvements if one can derive the theoretical guarantees and test with a range of under-sampling masks it would make the work more wholesome and principled.

⁰The codes for this project and simulation can be found at: https://github.com/DeepakRaya/CS_MRI_infoGAN_ADMM

Bibliography

- [1] S. Xu, S. Zeng, and J. Romberg, “Fast compressive sensing recovery using generative models with structured latent variables,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 2967–2971.
- [2] Y. Zhang, Y. Huang, H. Li, P. Li, X. Fan *et al.*, “Conjugate gradient hard thresholding pursuit algorithm for sparse signal recovery,” *Algorithms*, vol. 12, no. 2, p. 36, 2019.
- [3] S. Ravishankar and Y. Bresler, “Mr image reconstruction from highly undersampled k-space data by dictionary learning,” *IEEE Transactions on Medical Imaging*, vol. 30, no. 5, pp. 1028–1041, 2011.
- [4] Y. Yang, J. Sun, H. Li, and Z. Xu, “Deep admm-net for compressive sensing mri,” in *Proceedings of the 30th international conference on neural information processing systems*, 2016, pp. 10–18.
- [5] S. Wang, Z. Su, L. Ying, X. Peng, S. Zhu, F. Liang, D. Feng, and D. Liang, “Accelerating magnetic resonance imaging via deep learning,” in *2016 IEEE 13th International Symposium on Biomedical Imaging (ISBI)*. IEEE, 2016, pp. 514–517.
- [6] A. Bora, A. Jalal, E. Price, and A. Dimakis, “Compressed sensing using generative models,” in *ICML*, 2017.
- [7] G. Yang, S. Yu, H. Dong, G. Slabaugh, P. L. Dragotti, X. Ye, F. Liu, S. Arridge, J. Keegan, Y. Guo *et al.*, “Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction,” *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1310–1321, 2017.
- [8] P. Deora, B. Vasudeva, S. Bhattacharya, and P. M. Pradhan, “Structure preserving compressive sensing mri reconstruction using generative adversarial networks,” in

Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, 2020, pp. 522–523.

- [9] T. M. Quan, T. Nguyen-Duc, and W.-K. Jeong, “Compressed sensing mri reconstruction using a generative adversarial network with a cyclic loss,” *IEEE transactions on medical imaging*, vol. 37, no. 6, pp. 1488–1497, 2018.
- [10] S. H. Chan, X. Wang, and O. A. Elgendy, “Plug-and-play admm for image restoration: Fixed-point convergence and applications,” *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 84–98, 2016.
- [11] I. J. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” *CoRR*, vol. abs/1701.00160, 2017. [Online]. Available: <http://arxiv.org/abs/1701.00160>
- [12] X. Chen, Y. Duan, R. Houthoofd, J. Schulman, I. Sutskever, and P. Abbeel, “Infogan: Interpretable representation learning by information maximizing generative adversarial nets,” *Advances in neural information processing systems*, vol. 29, pp. 2172–2180, 2016.
- [13] J. Browlee, “Generative adversarial networks with python: deep learning generative models for image synthesis and image translation,” *Machine Learning Mastery*, 2019.
- [14] G. Strang, *Linear algebra and learning from data*. Wellesley-Cambridge Press Cambridge, 2019.
- [15] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.
- [16] C. M. M. Henry Knipe and et al, “Nifti (file format).” [Online]. Available: <https://radiopaedia.org/articles/nifti-file-format>
- [17] C. Markiewicz, “Nibabel: Access a cacophony of neuro-imaging file formats.” [Online]. Available: <https://nipy.org/nibabel/gettingstarted.html>
- [18] J. P. Hornak, “The basics of mri.” [Online]. Available: <https://www.cis.rit.edu/htbooks/mri/index.html>
- [19] S. Foucart and H. Rauhut, “A mathematical introduction to compressive sensing,” *Bull. Am. Math*, vol. 54, pp. 151–165, 2017.

- [20] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.
- [21] D. L. Donoho, A. Maleki, and A. Montanari, “Message-passing algorithms for compressed sensing,” *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

Appendix A

Principles of MRI and MRI acquisition

Magnetic Resonance imaging is a modality based on the phenomenon of nuclear magnetic resonance (NMR), primarily used for medical imaging, it is non invasive and possess good tissue contrast to look inside the human body.

The key basis of MRI is the NMR signal associated with the hydrogen atoms present as water and fat in the Human body. The NMR signal is produced by the nucleus of hydrogen atom due to the spin property.

Spin can be thought of as a small magnetic moment vector, which can align itself to external magnetic field (\mathbf{B}) and absorbs photons of frequency, $\nu = \gamma\mathbf{B}$, this frequency is termed as Larmour frequency. Two states of particles will be present when set of spins are placed in an external magnetic field one with lower energy spin and other with higher energy spin, at equilibrium, according to Boltzmann statistics lower energy spin state slightly outnumber the higher energy spins. when the group of spins are probed with a photon of Larmour frequency the lower energy spin will transition to higher energy spin, it is found that in NMR the Larmour frequency in Radio-Frequency Range.

A.1 Net Magnetization and T1,T2 Processes

The collection spins when placed in a magnetic field the net magnetic moment causes net magnetization vector to point out in the same direction of external magnetic field, if \mathbf{B}_0 in positive z direction, the net Magnetization $\mathbf{M}_{net} = M_0 \hat{\mathbf{z}}$, at equilibrium.

When sufficient amount of energy is incident on to the group of spins, the photons can change the net magnetization vector and it can even saturate the magnetization vector to zero.

A.1.1 T_1 Process

The time constant that accounts for the saturation of longitudinal magnetization (M_{net}) to its equilibrium value is termed as spin-lattice relaxation time (T_1).

$$M_z = M_0(1 - e^{-t/T_1}) \quad (\text{A.1})$$

A.1.2 Precession and T_2 Process

If net magnetization vector is forced down to $X - Y$ plane, it rotates about z axis with a precession rate equal to Larmour frequency, the net magnetization starts to dephase due to slight differences in magnetic field experienced among different spin sets.

The time constant that accounts for the transfer of transverse magnetization M_{xy} to its equilibrium value is called spin-spin relaxation constant T_2

$$M_{xy} = M_{xy0} e^{-t/T_2} \quad (\text{A.2})$$

Both process T_1 and T_2 occur simultaneously, with $T_1 \geq T_2$.

A.1.3 Time Domain NMR signal

If a coil is placed around X axis the transverse magnetization induces current in the coil, this current as a function of time is sinusoidal with decaying amplitude with time constant T_2 this signal is called Free induced decay (FID) signal, which is the NMR signal.

90-FID sequence

The sequence of RF-pulses applied in order produce particular NMR signal from a sample is called as pulse sequence. When 90 FID pulse sequence is activated the net magnetization vectors rotates on to $X - Y$ plane, and precess about Z axis, with decaying magnitude.

Spin-Echo Sequence

Here similar to FID pulse sequence a 90° pulse is applied first, which rotates the magnetization vector in $X - Y$ plane, which starts to dephase, then at some point in time a 180° pulse is applied, which tries to rephase the magnetization vector and produces a signal called echo.

Inversion Recovery Sequence

This is another pulse sequence for recording an NMR spectrum, first a 180° pulse is activated which forces the net magnetization vector in negative z direction, which then undergoes spin-relaxation towards positive z direction, but before reaching the equilibrium a 90° pulse is applied which rotates the magnetization vector onto $X - Y$ plane, which results in FID signal.

A.2 From NMR signals to Image

Previous section described about the principles of MRI and signal formation, this section deals with the formation of MRI image from NMR signal measurements (MRI acquisition).

A.2.1 The Three main systems of MRI

The three main systems of MRI are:

- Static Magnetic field
- Gradient Magnetic field
- The RF system

The static magnetic field gives rise to the resonance frequency of the spin called Larmour frequency and net Magnetization vector.

$$\nu_0 = \gamma \mathbf{B}_0 \quad (\text{A.3})$$

The gradient system introduces a form of regional encoding, so that one can uniquely associate a NMR measurement to a particular volume element. The Gradient magnetic field is a contrast in the magnitude of magnetic field with respect to the position, most common variation used for MR imaging is Linear Gradient, i.e.

$$\mathbf{B}_x = \mathbf{B}_0 + \mathbf{G}_x x \quad (\text{A.4})$$

Where a gradient along a particular direction is denoted with G_x , G_y , and G_z , one can now uniquely identify the a spatial encoding relation ship, since the resonance frequency varies with position.

$$\nu = \nu_0 + \gamma G_x x; \quad \text{where } \nu_0 = \gamma \mathbf{B}_0 \quad (\text{A.5})$$

$$x = \frac{(\nu - \nu_0)}{\gamma G_x} \quad (\text{A.6})$$

equations A.5 and A.6 represent the spatial encoding and frequency encoding due to gradient magnetic field applied in the X direction.

The RF system consists of transceiver coils which can transmit a RF pulse for slice selection and receive the FID or Echo signal from the excited spin system.

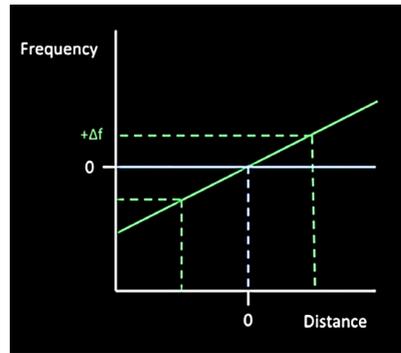
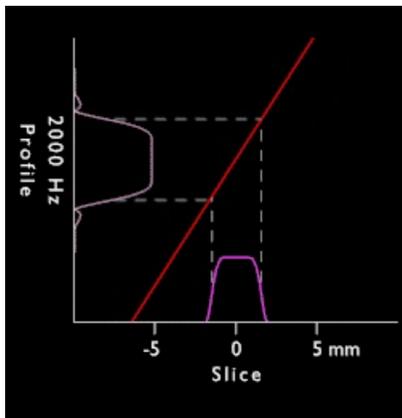
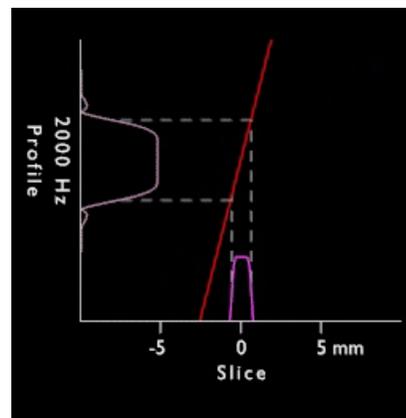


Figure A.1: Linear gradient of magnetic field



(a) slice selection(wide slice)



(b) slice selection(narrow slice)

Figure A.2

A.2.2 Fourier Transform Tomography

One of the early forms of MR imaging is to use back projection reconstruction of the selected slice, using the spatial and frequency encoding with the gradient magnetic fields. But with another gradient called phase encoding gradient one can reconstruct the MR image more effectively and efficiently.

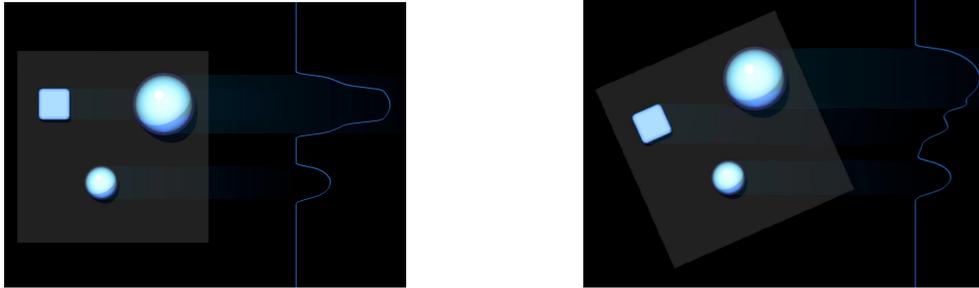


Figure A.3: Frequency Encoding: (a) and (b) represent the frequency encoding when the sample is placed in different orientations, (back projection reconstruction is performed by stage wise complete rotation)

Phase encoding gradient

It is the gradient in \mathbf{B}_0 which imparts certain phase angle to transverse magnetization vector. Initially all the magnetization vectors rotate at Larmor frequency, but when a gradient is applied along a transverse direction, each spin will precess at different frequency governed by the gradient field, after some time when the gradient is switched off the magnetization vectors rotate at same frequency but with non-identical phase angles, the amount of phase shift is dependent on the magnitude of the gradient and time for which it is applied. This step gives a unique phase angle for each spin in the system.

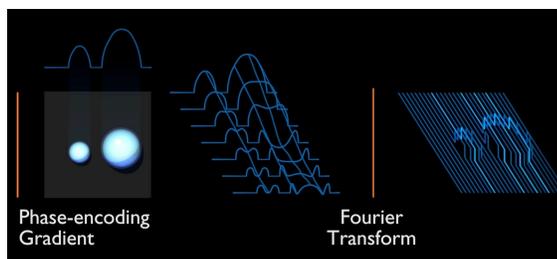


Figure A.4: Phase encoding: 7 repetitions of phase gradient activations with distinct gradient magnitudes are shown here

FT Tomographic Imaging

This explains the simplest form Fourier Transform imaging sequence for converting the received signal from the spin system to image. The timing diagram for the FT imaging is shown in A.5.

The first event is to activate the slice selection gradient and simultaneously slice selection RF pulse is applied. The slice selective RF pulse is an modified sinc function shaped burst of RF energy. Once the RF pulse is complete the slice selection gradient is put off and the phase encoding gradient is activated. Once the phase encoding gradient has been turned off a frequency encoding gradient is turned on and the FID signal is recorded.

This sequence is reiterated 128 or 256 times, depending on image size and resolution, to accumulate enough data required to produce an image. After each repetition time (TR) the magnitude of the phase encoding gradient is changed in regular steps between the maximum amplitude and the minimum amplitude of the gradient, so that it can have enough unique data to construct the image.

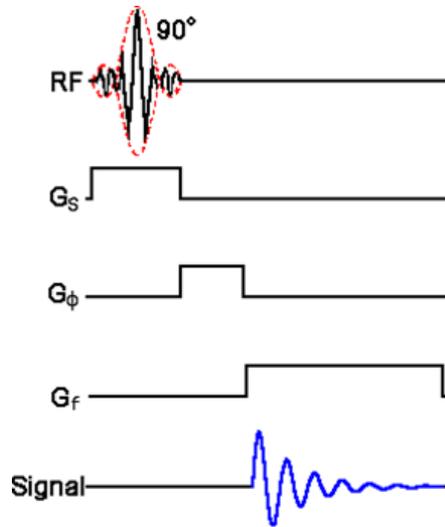


Figure A.5: The timing diagram for Fourier transform imaging sequence

The direction of applied field gradients is indicated in the following table A.1.

Table A.1: Direction of Gradient Fields

Slice	G_ϕ	G_s	G_f
XZ	Y	X or Z	Z or X
XY	Z	X or Y	Y or X
YZ	X	Y or Z	Z or Y

Consider a cube of spins as a the system to be imaged, this cube is consists of several volume elements each with its own net magnetization [18]. Suppose we want to image in $X - Y$ plane, the static magnetic field B_0 is along Z direction and a slice selection gradient is applied along Z direction, simultaneously a RF pulse is applied which rotates only the spins that satisfy resonance condition. The slice plane selected is given by

$$Z = \frac{\Delta\nu}{\gamma G_s} \quad (\text{A.7})$$

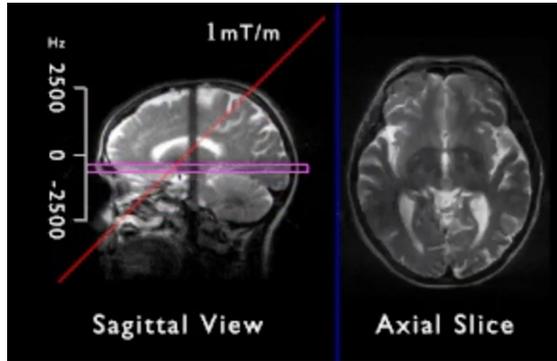


Figure A.6: Slice Selection

For simplicity let the slice selected be 3×3 element space, all the net magnetization vectors rotate at same Larmour frequency, Once a Phase Gradient along X direction is turned on, each element rotates at different frequency along the gradient direction and then if it is turned off, each slice element along X direction rotates with same frequency but with different phases.

Once the phase encoding gradient is turned off the frequency encoding gradient is switched on along Y direction, now different elements precess with different frequency depending on their y location. Finally this timing sequence imparts unique phase and frequency to each and every volume element of the slice. This implies that if we can determine the phase and frequency of the signal from the net magnetization vector we could position the

volume element in the slice.

If there were only one volume element in a slice we can perform this task by observing the spectrum of received FID signal. But if there are many elements we need multiple phase encoding gradients, i.e one for each location along the phase encoding gradient. If there are 256 phase encoding direction locations we need 256 discrete phase gradient amplitudes and 256 distinct FIDs will be recorded.

Image Resolution

The resolvability or distinguishability of two features in an image is called image resolution. Resolution quantifies the quality of an image. In MRI the resolution is dependent on T_2 , slice thickness, image size, SNR and sampling frequency.

Field of view is set by sampling interval Δk , by the relation $FOV = \frac{1}{\Delta k}$, voxel(volume element) size is set by maximum spectral frequency k_{max} , given by $\Delta x = \frac{1}{2k_{max}}$.

If two features are separated by a distance within $\frac{FOV}{N}$, where N number of data point across a dimension of MR image, they can not be resolved. Therefore pixel size and the parameters that it depend on must be chosen accordingly.

Image Contrast selection

To differentiate between tissues, we need to modulate the contrast based on some coupled properties of recorded signal and tissue. The magnitude of contrast (S) is dependent on type of pulse sequences being used.

The following are the contrast equations for different pulse sequences.

- $S = k\rho(1 - e^{(-TR/T1)})e^{(-TE/T2)}$ (: for Spin Echo)
- $S = k\rho(1 - 2e^{(-TI/T1)} + e^{(-TR/T1)})$ (: for 180-90 inversion recovery)
- $S = k\rho(1 - 2e^{(-TI/T1)} + e^{(-TR/T1)})e^{(-TE/T2)}$ (: for 180-90-180 inversion recovery)

Appendix B

Compressive Sensing

Compressive sensing is a framework for reconstruction of signal from far fewer samples compared to the traditional requirement of Shannon-Nyquist sampling, which requires dense samples and doesn't exploit the structure and sparsity, that is inherent to many natural signals.

The idea of compressed sensing comes down to the fact that there are only few degrees of freedom left after compressing; therefore, it is possible to take comparably less number of measurements than the complete dimension of the signal and reconstruct it approximately, if the signal is sparse in its pixel domain or in a transformed domain.

B.1 Compressed Sensing reconstruction formulation

The compressed sensing signal recovery can be formulated as a regularized optimization problem, as shown below:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} F(\mathbf{x}) + \lambda J(\mathbf{x}) \quad (\text{B.1})$$

$F(\mathbf{x})$ is a fidelity term with respect to measurements and $J(\mathbf{x})$ is a property term that a signal must satisfy, i.e. its minimization results in a output that belongs to the signal subspace. The scalar λ controls fidelity and property trade-off.

Usually the fidelity term is a norm error between true measurements(\mathbf{y}) and predicted measurement(\mathbf{Ax}), and the property term in traditional compressed sensing paradigms is a l_0 norm or a relaxed form of it, the l_1 norm [19] of the signal.

$$\mathbf{x}^* = \underset{x}{\operatorname{argmin}} \|\mathbf{y} - \mathbf{Ax}\|^2 + \lambda \|\mathbf{x}\|_1 \quad (\text{B.2})$$

The solution for B.2 gives a sparsest vector \mathbf{x} that has high fidelity with the observed measurements

B.1.1 Requirements for a successful Compressed Sensing

Compressed sensing relies on two main principles, one sparsity which depends on the signal of interest and two incoherence which depends on the sensing paradigm [20]. Signals have a concise representation when expressed in proper basis Ψ which is generally termed as sparsity. Incoherence between the representation basis and sensing basis imply that the sensing waveforms or basis must have dense representation in transform basis Ψ .

Let $f \in \mathbb{R}^n$ be the signal of interest and is sensed using m sensing waveforms ϕ_k $k \in 1, \dots, m$ i.e

$$y_k = \langle f, \phi_k \rangle \quad k = 1, \dots, m \quad (\text{B.3})$$

In B.3 y_k 's represent the measurements and $m \ll n$, the number of linear measurements of signal are far less than true signal dimension. let $\mathbf{A}_{m \times n}$ be a sensing matrix with rows as $\phi_1, \phi_2, \dots, \phi_m$, the complete sensing equation can be expressed as,

$$\mathbf{y} = \mathbf{A}\mathbf{f} \quad (\text{B.4})$$

which is a ill-posed problem, since $m \ll n$

Sparsity

A successful CS relies on assumption of sparsity of signal in its acquired domain or transformed domain, so that the search space to the ill posed problem B.4 can be properly restricted to sparse solutions.

Let the signal of interest f be sparse in ψ basis,

$$\mathbf{f} = \sum_{i=1}^n x_i \psi_i \quad (\text{B.5})$$

where x_i 's are the coefficients in transformed basis, given by

$$x_i = \langle \mathbf{f}, \psi_i \rangle$$

The signal \mathbf{f} , can be expressed as

$$\mathbf{f} = \Psi \mathbf{x} \quad (\text{B.6})$$

when the signal is sparse, small coefficients can be discarded, without much loss in the information content, if a signal is S -sparse, we can write B.6 as

$$\mathbf{f}_s = \Psi \mathbf{x}_s \quad (\text{B.7})$$

$$\mathbf{y}_k = A \mathbf{x}_s, \quad A = \Phi \Psi \quad (\text{B.8})$$

where \mathbf{x}_s , represents all, except S dominant values of \mathbf{x} , as zero. Now one can solve for the \mathbf{x}_s using the optimization program B.2.

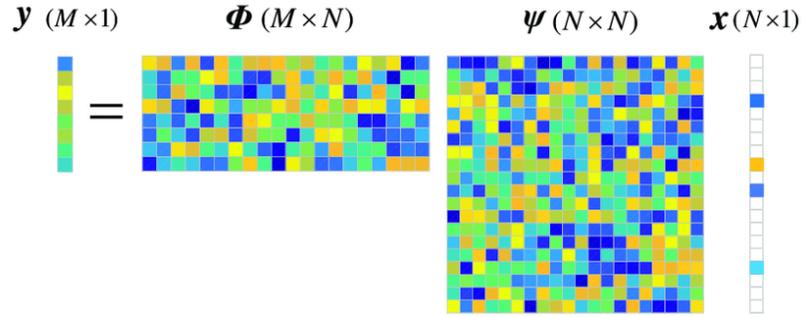


Figure B.1: Compressively sensed signal with transform sparsity $S = 4$ (adopted from [2])

Incoherence

Coherence between the pair of bases (Φ, Ψ) , where Ψ is the transform basis, and Φ is the sensing basis is defined as maximum correlation between any two elements of Φ and Ψ [20].

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{1 \leq k, j \leq n} |\langle \phi_k, \psi_j \rangle| \quad (\text{B.9})$$

for successful reconstruction we need pairs with lower coherence, i.e. incoherent pairs.

Random sensing matrices are found to be largely incoherent with any transformation basis Ψ , most commonly used sensing matrices for compressed sensing are the matrices with Gaussian *iid* entries or random binary (± 1)

Lower bound on Measurements

It is shown that number of measurements required for successful compressed sensing depends on incoherence between the sensing and transformation basis, sparsity of the underlying signal, and the true dimension of the signal. The lower bound on number of required measurements is given by,

$$m \geq C.\mu(\Phi, \Psi)^2.S.\log(n) \tag{B.10}$$

C is a constant, μ is the coherence, S is the sparsity of the signal and n is the total true dimension of the signal of interest.

Minimum number of measurements required for successful reconstruction increases logarithmically with the true dimension of the signal. Higher incoherence(lower coherence) and higher sparsity decreases the the lower bound on required measurement vector dimension.

It is also shown that one can infer the signal, with probability greater than $1 - \delta$, if the number of measurements $m \geq C.\mu(\Phi, \Psi)^2.S.\log(\frac{n}{\delta})$.

Restricted Isometry Property

One of the "key property" to study the robustness of Compressed sensing is "Restricted Isometry property"(RIP).

Isometry constant (δ_s) of a matrix \mathbf{A} is the smallest number such that,

$$(1 - \delta_s)\|\mathbf{x}\|^2 \leq \|\mathbf{Ax}\|^2 \leq (1 + \delta_s)\|\mathbf{x}\|^2 \tag{B.11}$$

holds for all S -sparse vectors \mathbf{x} , the matrix \mathbf{A} is said to obey the RIP of order s if δ_s is not very close to one [20]. RIP implicitly ensures that the S -sparse vectors can not be in Null space of \mathbf{A} .

B.2 Algorithms for Compressed Sensing Reconstruction

There are various classical algorithms for CS reconstruction, that use traditional formulation of sparsity prior on the signal, such as Orthogonal matching pursuit, basis pursuit, LASSO regression, iterative Thresholding and Approximate message passing.

On the other hand if the data of signals or images of interest are available one can use

various machine learning algorithms to learn the property term and then use a iterative optimization to reconstruct the signal, this can replace the assumption of sparsity and remove the need for designing a sparse basis for the signal under consideration. Another way is to learn a parametric function, usually a Deep Neural Network, which directly maps the input measurements to the reconstructed signal, this requires a supervised learning paradigm where the data is set of both signal data and corresponding measurements. Recent developments and enormous capability of the unsupervised learning paradigms and generative models such as GAN and VAE paves another effective way for learning the signal property term $J(\cdot)$, makes an attractive choice compared to traditional and supervised learning methods.

The classical l_1 minimization (Basis Pursuit) is already discussed in the introductory session of this chapter. This section deals with other main algorithms for CS Reconstruction, that are computationally fast and efficient, further some deep neural network based algorithms are also presented.

B.2.1 Matching Pursuit

"Matching Pursuit is an iterative algorithm", which decomposes the signal into a linear combination of functions that form a set called dictionary. At each iteration dictionary elements are chosen in a greedy-optimal way, so that the Linear expansion best approximates the signal of interest.

Let $\mathcal{D} = \{g_\gamma\}$ be a family of vectors with norm one, this set is called a dictionary, and f be the signal of interest. In Matching Pursuit f is decomposed as B.12, for any $g \in \mathcal{D}$

$$f = \langle f, g \rangle g + r^{(g)} \tag{B.12}$$

where $r^{(g)}$ is the residual left after approximation of f by g . For greedy option we want to choose the best approximation, since the residual is orthogonal to f and according to B.13, we want to minimize the l_2 -magnitude of residual, implicitly we want maximal $|\langle f, g \rangle|^2$.

$$\|f\|^2 = |\langle f, g \rangle|^2 + \|r^{(g)}\|^2 \tag{B.13}$$

Or it is also sufficient to choose the dictionary element that is almost optimal, i.e B.14 for any $\alpha \in (0, 1]$.

$$|\langle f, g \rangle|^2 \geq \alpha \sup_{\gamma \in \Gamma} |\langle f, g_\gamma \rangle|^2 \tag{B.14}$$

At each iteration matching pursuit will select the best approximation for the present residual from the dictionary, and progressively the residual term converges to zero.

Orthogonal Matching Pursuit

Orthogonal matching pursuit fits the original signal onto selected dictionary elements, in prior via Least squares, rather than taking the simple scalar products of residual with new dictionary elements for coefficients.

Algorithm B.1: Orthogonal Matching Pursuit

input:
 Φ : Measurement Matrix
 \mathbf{y} : Measurements Vector
 k : Assumed sparsity of the signal \mathbf{x}

output:
 $\hat{\mathbf{x}}$: Reconstruction
 Λ_k : set of non zeros position indices
 \mathbf{a}_k : measurements approximation
 $\mathbf{r} = \mathbf{y} - \mathbf{a}_k$: The residual

- 1 $\mathbf{r}^0 \leftarrow \mathbf{y}$
- 2 $\Lambda \leftarrow \emptyset$
- 3 **for** $i = 1, 2, \dots, n$ **do**
- 4 $\lambda^{(i)} \leftarrow \operatorname{argmax}_{j=1, \dots, n} |\langle \mathbf{r}^{(i-1)}, \phi_j \rangle|$
- 5 $\Lambda^{(i)} \leftarrow \Lambda^{(i-1)} \cup \lambda^{(i)}$
- 6 $\Phi^{(i)} \leftarrow \Phi^{(i-1)} \phi_{\lambda^{(i)}}$
- 7 $\mathbf{x}^{(i)} \leftarrow \operatorname{argmin}_{\hat{\mathbf{x}}} \|\Phi^{(i)} \hat{\mathbf{x}} - \mathbf{r}^{(i-1)}\|$
- 8 $\mathbf{a}^{(i)} \leftarrow \Phi^{(i)} \mathbf{x}^{(i)}$
- 9 $\mathbf{r}^{(i)} \leftarrow \mathbf{y} - \mathbf{a}^{(i)}$
- 10 **end**
- 11 $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{(k)}$

return:
 $\hat{\mathbf{x}}, \Lambda^k, \mathbf{a}^{(k)}, \mathbf{r}^{(k)}$

CoSaMP

This is an extension to orthogonal matching pursuit algorithm, and is shown to have tight bounds with respect to its convergence.

CoSaMP contains five main steps:

- **Identification:** Finds the significant $2s$ components in the proxy signal \mathbf{z} .
- **Support Merge:** merges support of proxy with support of the previous iterate of solution.
- **Estimation:** estimates the solution via constrained least squares on particular support
- **Pruning:** takes the estimate of the solution and compresses it to the required support
- **Sample Update:** updates the residual in Φ -space

Algorithm B.2: Compressive sampling matching pursuit

input:

Φ : Measurement Matrix

\mathbf{y} : Measurements Vector

s : sparsity of the signal \mathbf{x}

output:

$\hat{\mathbf{x}}$: Reconstruction

```

1  $\mathbf{x}^0 \leftarrow \mathbf{0}$ 
2  $\mathbf{v} \leftarrow \mathbf{y}$ 
3  $k \leftarrow 0$ 
4 while stopping condition not met do
5    $k \leftarrow k + 1$   $\mathbf{z} \leftarrow \Phi^T \mathbf{v}$ 
6    $\Omega \leftarrow \text{supp}(\mathbf{z}^{2s})$ 
7    $\Gamma \leftarrow \Omega \cup \text{supp}(\mathbf{x}^{(k-1)})$ 
8    $\bar{\mathbf{x}} \leftarrow \text{argmin}_{\bar{\mathbf{x}}: \text{supp}(\bar{\mathbf{x}})=\Gamma} \|\Phi \bar{\mathbf{x}} - \mathbf{y}\|_2$ 
9    $\mathbf{x}^{(k)} \leftarrow \bar{\mathbf{x}}$ 
10   $\mathbf{v} \leftarrow \mathbf{y} - \Phi \mathbf{x}^{(k)}$ 
11 end
12  $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{(k)}$ 
return:
 $\hat{\mathbf{x}}$ 

```

B.2.2 Iterative Thresholding

This class of algorithms, at each iteration step they perform some kind of thresholding operation \mathcal{T} , B.15, with parameters α and g is a function that acts on previous step output.

$$\mathbf{x}^{(i)} = \mathcal{T}_\alpha(g(\mathbf{x}^{(i-1)})) \quad (\text{B.15})$$

Iterative Hard Thresholding

Hard thresholding uses a function \mathcal{H}_s , which sets all except s significant components to zero.

$$\mathcal{H}_s = \begin{cases} \mathbf{x}_i & |\mathbf{x}_i| \geq |\zeta| \\ 0 & \text{otherwise} \end{cases} \quad (\text{B.16})$$

Algorithm B.3: Iterative Hard Thresholding

input:
 Φ : Measurement Matrix
 \mathbf{y} : Measurements Vector
 s : sparsity of the signal \mathbf{x}
output:
 $\hat{\mathbf{x}}$: Reconstruction

- 1 $\mathbf{x}^0 \leftarrow \mathbf{0}$
- 2 **for** i, \dots **do**
- 3 $\mathbf{x}^{(i)} \leftarrow \mathcal{H}_s(\mathbf{x}^{(i-1)} + \Phi^T(\mathbf{y} - \Phi\mathbf{x}^{(i-1)}))$
- 4 **end**
- 5 $\hat{\mathbf{x}} \leftarrow \mathbf{x}^{(k)}$

return:
 $\hat{\mathbf{x}}$

Iterative Soft Thresholding

Soft Thresholding function performs B.17 component wise

$$\mathcal{S}_\alpha = \begin{cases} x - \alpha & x > \alpha \\ 0 & |x| \leq \alpha \\ x + \alpha & x < -\alpha \end{cases} \quad (\text{B.17})$$

Soft thresholding is used to minimize LASSO or basis pursuit kind of objectives

$$F_\alpha(x) = \|\Phi x - y\|_2^2 + 2\alpha\|x\|_1 \quad (\text{B.18})$$

and the solution to B.18 is given as

$$Lt_{k \rightarrow \infty} x^{(k+1)} = \mathcal{S}_\alpha(x^{(k)} + \Phi^T y - \Phi^T \Phi x^{(k)}) \quad (\text{B.19})$$

Algorithm B.4: Iterative Soft Thresholding

input:

Φ : a Matrix with Eigen values < 1

y : Measurements Vector

a Radius R

output:

A vector $\mathbf{x}_R \in \mathbb{R}^n$ that is the minimiser in B_R of $\mathcal{D}(\mathbf{x}) = \|\Phi x - y\|_2$

1 $\mathbf{x}^0 \leftarrow \mathbf{0}$

2 **for** i, \dots **do**

3 Select $\beta^{(k)}$

4 $\mathbf{x}^{(k)} \leftarrow \mathcal{P}_R[\mathbf{x}^{(k-1)} + \beta^{(k)}\Phi^T(y - \Phi\mathbf{x}^{(k-1)})]$

5 **end**

6 $\mathbf{x}_R \leftarrow \mathbf{x}^{(k)}$

B_R is the l_1 ball in \mathbb{R}^n with l_1 radius R , and \mathcal{P}_R is the projection of point \mathbf{x} to a point closest, in l_2 sense onto the convex set B_R .

B.2.3 Approximate Message Passing

This method[21] is a modification on iterative thresholding algorithms, with a important term derived based on "belief propagation theory" and found to improve "sparsity-undersampling trade-off". measurements taken according to $\mathbf{y} = \Phi\mathbf{x}$, we initially start with $\mathbf{x}^{(0)} \leftarrow \mathbf{0}$ and the approximate message passing algorithms look like

$$\mathbf{z}^{(k)} = \mathbf{y} - \Phi\mathbf{x}^{(k)} + \frac{1}{\delta}\mathbf{z}^{(k-1)}\langle\eta_{k-1}(\Phi^*\mathbf{z}^{(k-1)} + \mathbf{x}^{(k-1)})\rangle \quad (\text{B.20})$$

$$\mathbf{x}^{(k+1)} = \eta_k(\Phi^*\mathbf{z}^{(k)} + \mathbf{x}^{(k)}) \quad (\text{B.21})$$

The special crucial term $\frac{1}{\delta}\mathbf{z}^{(k-1)}\langle\eta_{k-1}(\Phi^*\mathbf{z}^{(k-1)} + \mathbf{x}^{(k-1)})\rangle$ is what makes this algorithm more attractive over general thresholding algorithm.