

---

# Learning to regularize, Application of Neumann Network to the fastMRI dataset

---

**Yves Greatti**  
New York University

**Soham Girish Tamba**  
New York University

## Abstract

Recent progress in Machine Learning and Data science has seen the development of efficient Neural Networks architecture to solve inverse-problems. Neural networks have surpassed more traditional approaches in many natural sciences, medicine and life sciences, challenging applications. Our goal in this work has been first to perform a quick survey of data-driven inverse problem methodologies, followed by an understanding of the key ideas at the core of a Neumann Network and its implementation and application to the task of image reconstruction using the fastMRI dataset, a very-large collection of undersampled Magnetic Resonance Imaging (MRI) measurements.

## 1 Introduction

In term of operator, an inverse problem is formalized as solving an operator equation

$$\mathbf{Y} = \mathcal{A}(\beta_{\text{true}}) + \epsilon \quad (1)$$

where  $\mathcal{A}$  is a forward operator from the model parameter space  $\mathbf{X}$  to the data space  $\mathbf{Y}$ ,  $\beta_{\text{true}}$  the ground truth image and  $\epsilon$  a random variable modeling the observation noise. The MRI (Magnetic Resonance Imaging) image reconstruction problem is an inverse problem, the forward operator is a discrete sampling operator concatenated with the Fourier transform. To eliminate image artifacts due to patient movements, an MRI machine acquires less data to reduce the scan time. For the fastMRI image reconstruction challenge, the undersampling consists in masking k-space lines from a fully-sampled acquisition. A difficulty in solving (1), is that the solution is sensitive to variations in the data, which is referred to ill-posedness. The notion of ill-posedness is attributed to Hadamard, who defined that, by contrast, a well-posed problem has three distinct properties: (1) it has a solution, that is unique (2), and that depends continuously on the data (3). Another way of describing ill-posedness is to understand the operator  $\mathcal{A}$  and when it is linear, look at its singular value decomposition. The decay of the spectral spectrum  $\sigma_k, k \in \mathcal{N}$  of  $\mathcal{A}$  is strongly related to the ill-posedness: faster decay implies a more ill-posed problem.

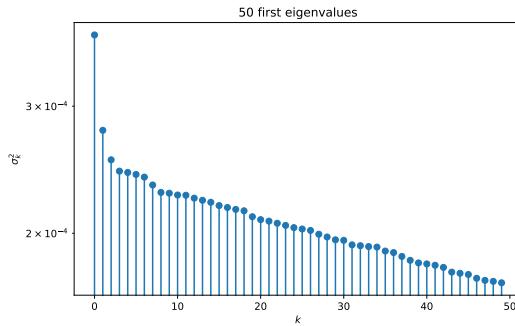


Figure 1: 50 first singular value of the k-space of an image from the fastMRI dataset after undersampling.

The problem defined in (1), can be formulated as an ordinary least squares problem (OLS) , given a set of training images  $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\}$ , and an operator  $\mathcal{A}$ , we want to compute the estimates  $\boldsymbol{\beta}$  which satisfy:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \|\mathbf{y} - \mathcal{A}\boldsymbol{\beta}\|_2^2$$

The aim of regularization theory is to provide a regularizer  $r(\cdot)$  which pushes the estimate  $\boldsymbol{\beta}$  to  $\boldsymbol{\beta}_{\text{true}}$ . We now set the inverse problem as:

$$\hat{\boldsymbol{\beta}} = \arg \min_{\boldsymbol{\beta}} \frac{1}{2} \|\mathbf{y} - \mathcal{A}\boldsymbol{\beta}\|_2^2 + r(\boldsymbol{\beta})$$

The regularization operator  $r(\cdot)$ , is a denoising encoder and one approach to solve these inverse problems is to learn  $r(\cdot)$ .

## 2 Theory

The core idea of a Neumann network, it is to use a denoising autoencoder as a proximal operator in an iterative scheme. The estimated reconstructed image is given by:

$$\text{prox}_r(\boldsymbol{\beta}) := \arg \min_{\boldsymbol{\beta}} r(\boldsymbol{\beta}) + \frac{1}{2} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|_2^2 \quad (2)$$

Assuming that  $r(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^T \mathbf{R} \boldsymbol{\beta}$  then taking the gradient of the (2), leads to the optimal solution  $\boldsymbol{\beta}^*$

$$\begin{aligned} (\mathbf{X}^T \mathbf{X} + \mathbf{R}) \boldsymbol{\beta}^* &= \mathbf{X}^T \mathbf{y} \\ \boldsymbol{\beta}^* &= (\mathbf{X}^T \mathbf{X} + \mathbf{R})^{-1} \mathbf{X}^T \mathbf{y} \end{aligned}$$

Using a well-established result, for any matrix,  $p \times p \mathbf{A}$ , the Neumann series,  $\sum_{k=0}^{\infty} \mathbf{A}^k$ , converges to the inverse of  $\mathbf{I} - \mathbf{A}$ , assuming  $\|\mathbf{A}\|_{l_i} < 1$  where  $l_i$  is an operator norm. We are now in position to replace the inversion defined in (2), by a finite sum of power of  $\mathbf{A}$ :

$$\begin{aligned} (\mathbf{I} - \mathbf{A})^{-1} &= \sum_{k=0}^{\infty} \mathbf{A}^k = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \dots \\ \mathbf{B}^{-1} &= \eta \sum_{k=0}^{\infty} (\mathbf{I} - \eta \mathbf{B})^k \text{ ancillary result} \\ \boldsymbol{\beta}^* &= \sum_{j=0}^{\infty} (\mathbf{I} - \eta \mathbf{X}^T \mathbf{X} - \eta \mathbf{R})^j (\eta \mathbf{X}^T \mathbf{y}) \\ \tilde{\boldsymbol{\beta}}(\mathbf{y}) &:= \sum_{j=0}^B \left( [\mathbf{I} - \eta \mathbf{X}^T \mathbf{X}](\cdot) - \eta \mathbf{R}(\cdot) \right)^j (\eta \mathbf{X}^T \mathbf{y}) \end{aligned}$$

The last equation has a natural recursive form which is easily interpretable as a deep learning architecture:

$$\begin{aligned} \tilde{\boldsymbol{\beta}}^{(j)} &:= \left( \mathbf{I} - \eta \mathbf{X}^T \mathbf{X} \right) \tilde{\boldsymbol{\beta}}^{(j-1)} - \eta \mathbf{R}(\tilde{\boldsymbol{\beta}}^{(j-1)}), j = 1, \dots, B \\ \tilde{\boldsymbol{\beta}}(\mathbf{y}) &:= \sum_{j=0}^B \tilde{\boldsymbol{\beta}}^{(j)} \end{aligned}$$

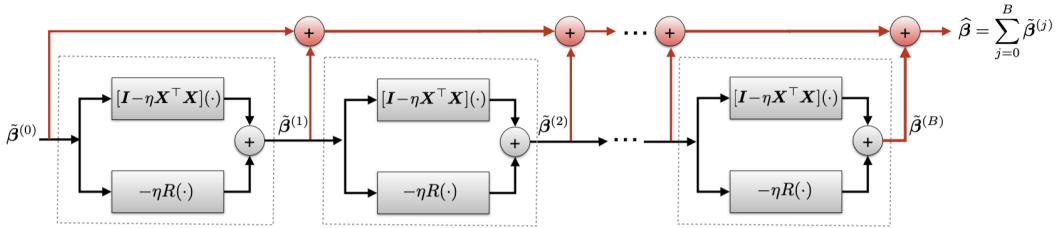


Figure 2: Neumann Network Architecture

One interesting lemma established by Gilton et Al ([7]), is that the finite sum  $\sum_{j=0}^B \tilde{\beta}^{(j)}$  provides indeed an estimate for the reconstructed image  $\tilde{\beta}(\mathbf{y})$  approximatively equal to the optimal reconstruction  $\beta^*$ , almost *independently* of the number of iterations, which is a result we recovered through our experiments.

Assuming that the autoencoder is linear with the input

$$\mathbf{R}(\beta) = \mathbf{R}\beta = -c_{\eta, B}(\mathbf{I} - \mathbf{X}^T \mathbf{X})\mathbf{U}(\mathbf{U}^T \mathbf{X}^T \mathbf{X}\mathbf{U})^{-1}\mathbf{U}^T \mathbf{X}^T \mathbf{X}$$

where  $\tilde{\beta}^{(0)} = \eta \mathbf{X}^T \mathbf{y}, \mathbf{y} = \mathbf{X}\beta^*$

then

$$\begin{aligned} \tilde{\beta}^{(j)} &= a_j \mathbf{X}^T \mathbf{X} \beta^* + b_j (\mathbf{I} - \mathbf{X}^T \mathbf{X}) \beta^* \\ \text{where } \sum_j a_j &\approx 1 \text{ and } \sum_j b_j \approx 1 \\ \hat{\beta}(\mathbf{y}) &= \sum_{j=0}^B \tilde{\beta}^{(j)} \\ &\approx \mathbf{X}^T \mathbf{X} \beta^* + (\mathbf{I} - \mathbf{X}^T \mathbf{X}) \beta^* = \beta^* \end{aligned}$$

### 3 MRI Experiments

#### 3.1 Data Preparation

The raw MRI data includes 8344 volumes, consisting of 167,375 slices or images of knees or brain examinations. The measurements are in the Fourier-space domain, known as *k-space*. In absence of noise or undersampling, the true image  $\beta_{\text{true}}$  can be fully recovered using an inverse multidimensional Fourier transform:  $\beta^* = \mathcal{F}^{-1}(\mathbf{y})$ . However to reduce acquisition time for various limiting factors (cost, patient experience), only part of the k-space samples are retained, the rest of the k-space lines are masked based on a parameter called *acceleration*. For example, with an acceleration of four, only 8% of all-kspace are included, during our experiments we selected a mask with random acceleration of four and eight. Random undersampling is usually chosen to provide a fair comparison with more traditional learned reconstruction algorithms.

	Volumes	Slices
Training	973	34,742
Validation	199	7,135
Test	108	3,903

Table 1: Single-coil: volumes and slices in each set

### 3.2 Deep learning architecture

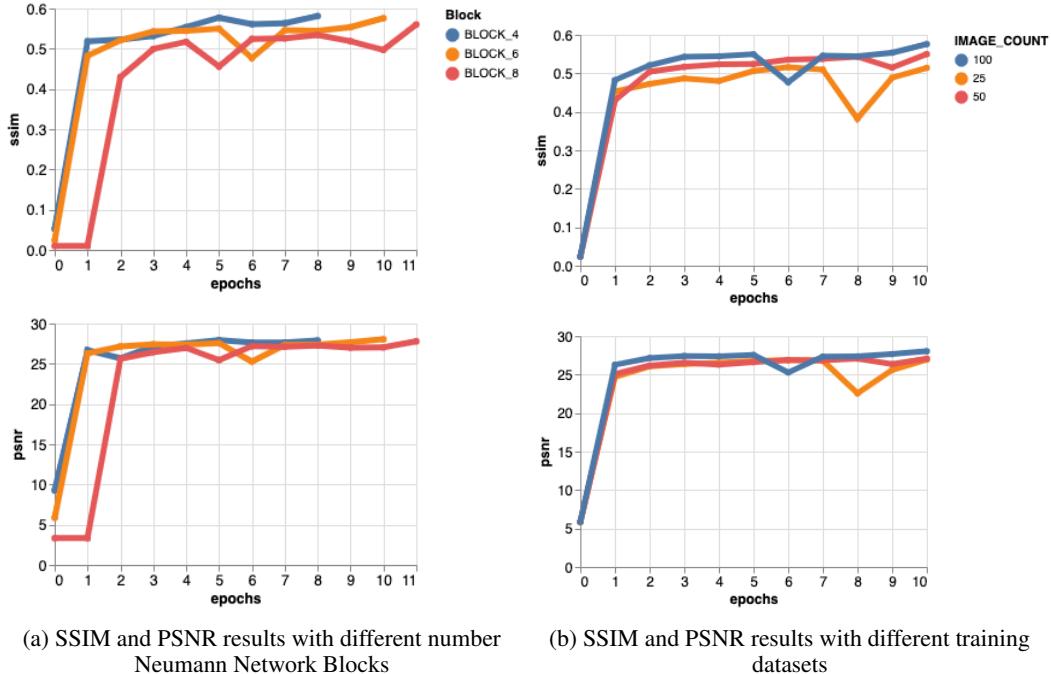
Two steps are critical when creating a Neumann network

- defining the forward operator  $\mathbf{X}$  and the gramian operator  $\mathbf{X}^T \mathbf{X}$
- selecting an autoencoder  $\mathbf{R}$  as regulizer

We implemented and tested various deep learning models for  $\mathbf{R}$  and finally settled on the U-Net architecture of the fastMRI challenge itself. If Neumann network has to improve any neural network regularizer, it should also make a difference on the baselines produced by this model. The U-Net single-coil baseline model consists of four deep convolutional networks, a down-sampling path followed by an up-sampling path. The training process used by Neumann network was

$$\arg \min_{\theta} \frac{1}{2} \sum_{j=1}^N \|\tilde{\beta}(\mathbf{y}) - \beta_{\text{true}}\|_1, N = \text{number of samples}$$

We ran various experiments with different number of iterations blocks  $B$ , network hyperparameter  $\theta$  and training sizes. We used RMSProp with a learning rate of 0.001, which was multiplied by 0.1 after 40 epochs when the model was initially trained for 50 epochs but realized that the network was quickly reaching its best performance metrics after few epochs. At the end of each epoch, we set up a validation step where we recorded the NMSE, SSIM, PSNR and the validation loss.



Compared to the plain U-Net, the performances of Neumann network are similar. We can detect some improvements when looking more closely at the reconstructed images 5c.

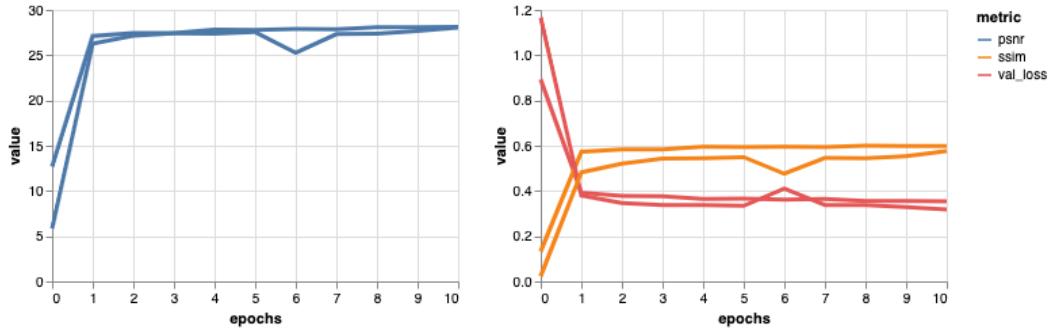
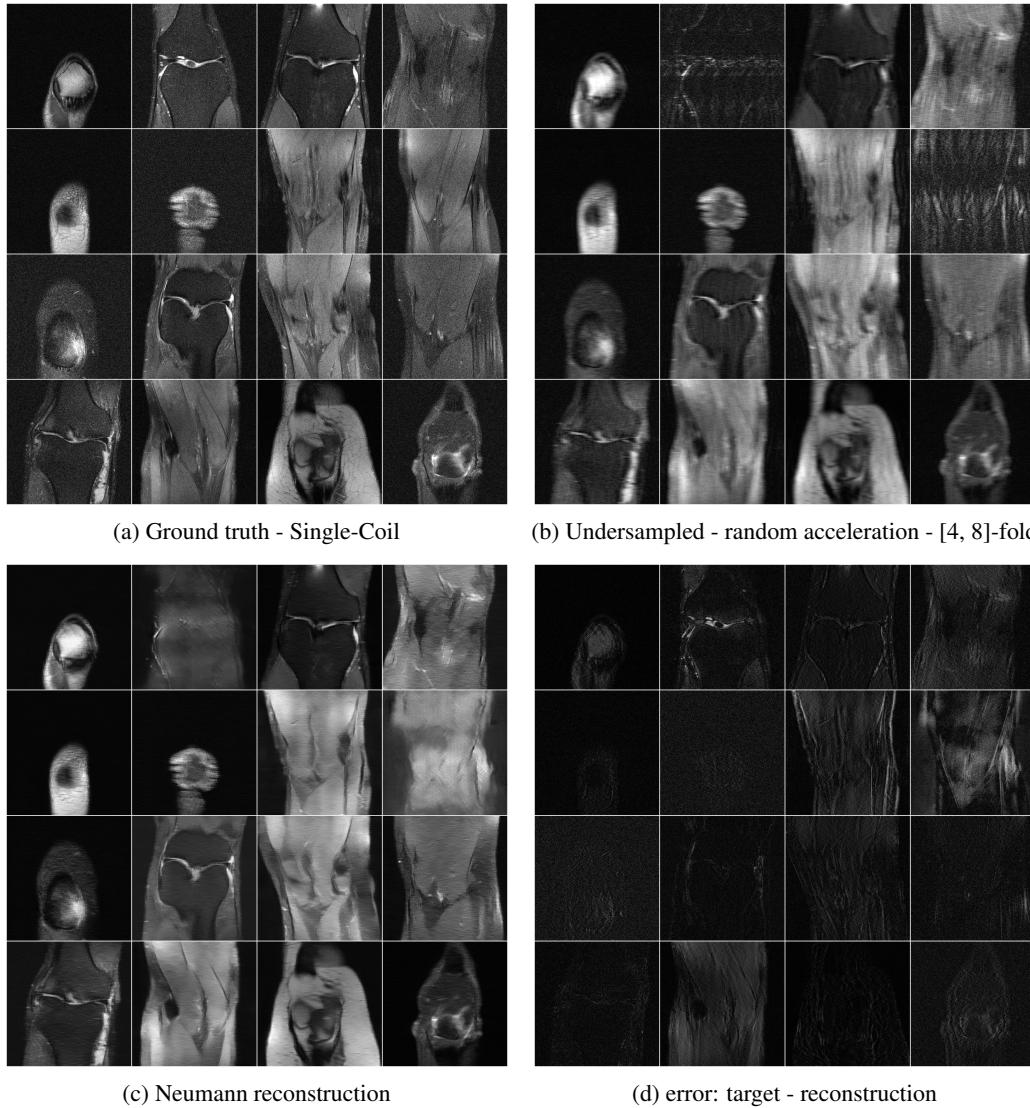


Figure 4: On the left PSNR for U-Net and Neumann and on the right SSIM and Validation loss for Neumann and U-Net.



## 4 Ill-posedness and non-linearity in the reconstruction process

When the matrix  $\mathbf{X}^T \mathbf{X} + \mathbf{R}$  is ill-conditioned, which happens in most cases, the authors, Gilton et Al ([7]), propose a preconditioning:

$$(\mathbf{X}^T \mathbf{X} + \mathbf{R})\beta^* = \mathbf{X}^T \mathbf{y}$$

$$(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})\beta^* + (\mathbf{R} - \lambda \mathbf{I})\beta^* = \mathbf{X}^T \mathbf{y}$$

Applying  $\mathbf{T}_\lambda := (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1}$  to both sides and rearranging the terms gives

$$(\mathbf{I} - \lambda \mathbf{T}_\lambda + \tilde{\mathbf{R}})\beta^* = \mathbf{T}_\lambda \mathbf{X}^T \mathbf{y}, \tilde{\mathbf{R}} = \mathbf{T}_\lambda \mathbf{R}$$

$$\hat{\beta}_{\text{pc}}(\mathbf{y}) = \sum_{j=0}^B \left( \lambda \mathbf{T}_\lambda(\cdot) - \tilde{\mathbf{R}}(\cdot) \right)^j \mathbf{T}_\lambda \mathbf{X}^T \mathbf{y}$$

In our experiments with the fastMRI datasets, a preconditioned Neumann network did not achieve higher quality results.

An attempt to address the non-linearities of the MRI process using an unrolled Gradient Descent scheme would be to use the Adomian decomposition method. In this method, the studied equation representing the non-linearities of the physical model is  $L(u) + R(u) + N(u) = q$  where  $L$  is a linear operator,  $N$  is an analytical nonlinear operator. Solving for  $L(u)$  gives:  $L(u) = g - R(u) - N(u)$ . Assuming  $L$  invertible, the expression becomes:

$$L^{-1}L(u) = L^{-1}g - L^{-1}R(u) - L^{-1}N(u) \quad (3)$$

leading to the solution of the initial equation as

$$u = \sum_{i=0}^{\infty} u_i \quad (4)$$

The non-linear term  $N(u)$  can be decomposed by the infinite series of Adomian polynomials

$$N(u) = \sum_{i=0}^{\infty} A_i \quad (5)$$

The  $A_i$ 's are given as:

$$A_0 = F(u_0)$$

$$A_1 = u_1 \frac{d}{du_0} F(u_0)$$

$$A_2 = u_2 \frac{d}{du_0} F(u_0) + \frac{u_1^2}{2!} \frac{d^2}{du_0^2} F(u_0)$$

$$\vdots$$

Substituting back (4) and (5) into (3) yields

$$\sum_{i=0}^{\infty} u_i = u_0 + L^{-1}R\left(\sum_{i=0}^{\infty} u_i\right) - L^{-1}\sum_{i=0}^{\infty} A_i \quad (6)$$

With a suitable  $u_0$  we obtain

$$u_1 = -L^{-1}R(u_0) - L^{-1}A_0$$

$$\vdots$$

$$u_{n+1} = -L^{-1}R(u_n) - L^{-1}A_n$$

Thus assuming the linear operator  $L$  and Adomian polynomials  $A_I, I = 1, \dots, n$  can be analytically determined, we can reuse a deep learning architecture similar to 2, the regularizer could be a FFN trained with observed data points  $\{x_i, u_i\}, i = 1, \dots, n$  using a mean square loss function (MSE).

## References

- [1] Jonas Adler and Ozan Öktem. Solving ill-posed inverse problems using iterative deep neural networks. *Inverse Problems*, 33(12):124007, Nov 2017. ISSN 1361-6420. doi: 10.1088/1361-6420/aa9581. URL <http://dx.doi.org/10.1088/1361-6420/aa9581>.
- [2] Hemant Kumar Aggarwal, Merry P. Mani, and Mathews Jacob. Modl: Model based deep learning architecture for inverse problems. *CoRR*, abs/1712.02862, 2017. URL <http://arxiv.org/abs/1712.02862>.
- [3] Tim Dockhorn. A discussion on solving partial differential equations using neural networks, 2019.
- [4] Ali Hasan, João M. Pereira, Robert Ravier, Sina Farsiu, and Vahid Tarokh. Learning partial differential equations from data using neural networks, 2019.
- [5] Ulugbek S. Kamilov and Hassan Mansour. Learning optimal nonlinearities for iterative thresholding algorithms. *CoRR*, abs/1512.04754, 2015. URL <http://arxiv.org/abs/1512.04754>.
- [6] Yaniv Romano, Michael Elad, and Peyman Milanfar. The little engine that could: Regularization by denoising (RED). *CoRR*, abs/1611.02862, 2016. URL <http://arxiv.org/abs/1611.02862>.
- [7] Willett et al. Neumann networks for inverse problems in imaging. *CoRR*, abs/1901.03707, 2019. URL <http://arxiv.org/abs/1901.03707>.
- [8] Jure Zbontar et al. fastmri: An open dataset and benchmarks for accelerated MRI. *CoRR*, abs/1811.08839, 2018. URL <http://arxiv.org/abs/1811.08839>.