Learning to Solve Linear Inverse Problems in Imaging with Neumann Networks

Greg Ongie
University of Chicago
gongie@uchicago.edu

Davis Gilton
University of Wisconsin, Madison
gilton@wisc.edu

Rebecca Willett
University of Chicago
willett@uchicago.edu

Abstract

Recent advances have illustrated that it is often possible to *learn* to solve linear inverse problems in imaging using training data that can outperform more traditional regularized least squares solutions. Along these lines, we present some extensions of the Neumann network, a recently introduced end-to-end learned architecture inspired by a truncated Neumann series expansion of the solution map to a regularized least squares problem. Here we summarize the Neumann network approach, and show that it has a form compatible with the optimal reconstruction function for a given inverse problem. We also investigate an extension of the Neumann network that incorporates a more sample efficient patch-based regularization approach.

1 Learning to solve inverse problems

We consider solving linear inverse problems in imaging in which a p-pixel image, $\boldsymbol{\beta}^\star \in \mathbb{R}^p$ (in vectorized form), is observed via m noisy linear projections as $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}^\star + \boldsymbol{\epsilon}$, where $\boldsymbol{X} \in \mathbb{R}^{m\times p}$ and $\boldsymbol{\epsilon} \in \mathbb{R}^m$ is a noise vector. The problem of estimating $\boldsymbol{\beta}^\star$ from $\boldsymbol{y}$ is referred to as *image reconstruction*, and a typical estimate is given by solving the regualarized least squares problem

$$\widehat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + r(\boldsymbol{\beta}). \tag{1}$$

where $r(\cdot)$ is a regularizer. Classical image reconstruction methods specify a choice of regularizer to promote piecewise smoothness of the reconstruction, sparsity in some dictionary or basis, or other geometric properties.

However, an emerging body of research explores the idea that training data can be used to learn to solve inverse problems using neural networks. At a high level, existing learning-based approaches to solving inverse problems can be categorized as either *decoupled* or *end-to-end*.

Decoupled approaches first learn a representation of the data that is independent of the forward model $\boldsymbol{X}$, followed by a reconstruction phase that uses $\boldsymbol{X}$ explicitly. Existing methods in this vein include using training images to learn a low-dimensional image manifold captured by the range of a generative adversarial network (GAN) and constraining the estimate $\widehat{\boldsymbol{\beta}}$ to lie on this manifold [1], or learning a denoising autoencoder that can be treated as a regularization step (*i.e.,* proximal operator) within an iterative reconstruction scheme [2, 3].

End-to-end approaches incorporate the forward model $\boldsymbol{X}$ directly into the network architecture during both training and testing, and are optimized for a specific $\boldsymbol{X}$ or class of $\boldsymbol{X}$'s. Many end-to-end approaches are based on "unrolling" finitely many iterations of an optimization algorithm for solving (1), where instances of the regularizer (or its gradient or proximal operator) are replaced by a neural network to be trained; see [4, 5, 6, 7, 8] among others.

The advantage of a decoupled approach is that the learned representation can be used for a wide variety of inverse problems without having to retrain. However, this flexibility comes with a high price in

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

terms of sample complexity. Learning a generative model or a denoising autoencoder fundamentally amounts to estimating a probability distribution and its support over the space of images; let us denote this distribution as $P(\boldsymbol{\beta})$. On the other hand, if $\boldsymbol{X}$ is known at training time, then we only need to learn the *conditional* distribution $P(\boldsymbol{\beta}|\boldsymbol{X}\boldsymbol{\beta})$, which can require far fewer samples to estimate [9].

To make this idea more precise, consider the problem of finding the MSE optimal reconstruction function in the noiseless setting:

$$\rho^* = \arg\min_{\rho} \mathbb{E}_{\boldsymbol{\beta}\sim P(\boldsymbol{\beta})} \left[\|\boldsymbol{\beta} - \rho(\boldsymbol{X}\boldsymbol{\beta}))\|^2 \right]. \tag{2}$$

Then ρ^* is characterized as follows.

Proposition 1. *Let $\boldsymbol{X} \in \mathbb{R}^{m\times p}$, $m \leq p$, be full rank, and let $\boldsymbol{X}_\perp \in \mathbb{R}^{p-m\times p}$ be a matrix whose rows form an orthonormal basis for the nullspace of $\boldsymbol{X}$. Then the MSE-optimal reconstruction function ρ^* in* (2) *is given by*

$$\rho^*(\boldsymbol{y}) = \boldsymbol{X}^+\boldsymbol{y} + \boldsymbol{X}_\perp^\top \mathbb{E}[\boldsymbol{X}_\perp\boldsymbol{\beta} \,|\, \boldsymbol{X}^+\boldsymbol{y}] \tag{3}$$

where $\boldsymbol{X}^+$ is the pseudoinverse of $\boldsymbol{X}$ and $\mathbb{E}[\boldsymbol{X}_\perp\boldsymbol{\beta} \,|\, \boldsymbol{X}^+\boldsymbol{y}]$ is the conditional expectation of $\boldsymbol{X}_\perp\boldsymbol{\beta}$ given $\boldsymbol{X}^+\boldsymbol{y}$.

We omit the proof of Proposition 1 for brevity, but the technique is similar to those used in [10] to derive the expressions for the MSE optimal autoencoder for a given data distribution.

This proposition shows that the optimal reconstruction function only requires estimating a conditional expectation of the component of the image in the nullspace of the linear forward model, or implicitly a conditional probability density rather than the full probability density over the space of all images. Therefore, in settings where training data is limited, end-to-end approaches are expected to outperform decoupled approaches due to their lower sample complexity. It also implies the end-to-end networks should have a structure compatible with (2) if they are to well-approximate the MSE optimal reconstruction function.

The focus of this work is on the recently-proposed Neumann network architecture as an end-to-end approach for learning to solve inverse problems [11]. Here we summarize the Neumann network architecture, and give it a new interpretation in light of Proposition 1. We also introduce an extension of Neumann networks to the case of a patch-based regularization strategy, which further improves the sample complexity of the approach.

2 Neumann Networks

Neumann networks are motivated by the regularized least squares optimization problem (1) in the special case where the regularizer r is quadratic. In particular, assume $r(\boldsymbol{\beta}) = \frac{1}{2}\boldsymbol{\beta}^\top\boldsymbol{R}\boldsymbol{\beta}$ so that $\nabla r(\boldsymbol{\beta}) = \boldsymbol{R}\boldsymbol{\beta}$ for some matrix $\boldsymbol{R} \in \mathbb{R}^{p\times p}$. A necessary condition for $\boldsymbol{\beta}^\star$ to be a minimizer of (1) in this case is

$$(\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{R})\boldsymbol{\beta}^\star = \boldsymbol{X}^\top\boldsymbol{y} \tag{4}$$

If the matrix on the left-hand side of (4) is invertible, the solution is given by

$$\boldsymbol{\beta}^\star = (\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{R})^{-1}\boldsymbol{X}^\top\boldsymbol{y}. \tag{5}$$

To approximate the matrix inverse in (5), the authors of [11] use a Neumann series expansion for the inverse of a linear operator [12], given by $\boldsymbol{A}^{-1} = \eta\sum_{k=0}^{\infty}(\boldsymbol{I} - \eta\boldsymbol{A})^k$, which converges provided $\|\boldsymbol{I} - \eta\boldsymbol{A}\| < 1$. Applying this series expansion to the matrix inverse appearing in (5), we have $\boldsymbol{\beta}^\star = \sum_{j=0}^{\infty}(\boldsymbol{I} - \eta\boldsymbol{X}^\top\boldsymbol{X} - \eta\boldsymbol{R})^j(\eta\boldsymbol{X}^\top\boldsymbol{y})$. Truncating this series to $B+1$ terms, and replacing multiplication by the matrix $\boldsymbol{R}$ with a general learnable mapping $R : \mathbb{R}^p \to \mathbb{R}^p$, motivates an estimator $\widehat{\boldsymbol{\beta}}$ of the form

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{y}) := \sum_{j=0}^{B}([\boldsymbol{I} - \eta\boldsymbol{X}^\top\boldsymbol{X}](\cdot) - \eta R(\cdot))^j(\eta\boldsymbol{X}^\top\boldsymbol{y}). \tag{6}$$

The estimator above becomes trainable by letting $R = R_{\boldsymbol{\theta}}$ be a neural network depending on a vector of parameters $\boldsymbol{\theta} \in \mathbb{R}^q$ to be learned from training data, along with the scale parameter η.

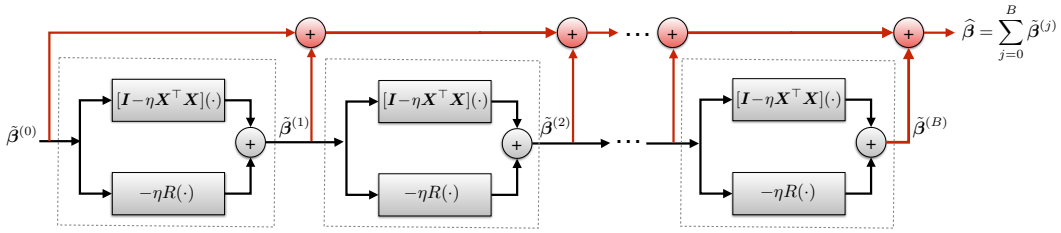


Figure 1: Neumann network architecture. Unlike other networks based on unrolling of iterative optimization algorithms, the series structure of Neumann networks lead naturally to additional "skip connections" (highlighted in red) that route the output of each dashed block to directly to the output layer.

Any estimator $\widehat{\boldsymbol{\beta}}(\boldsymbol{y}) = \widehat{\boldsymbol{\beta}}(\boldsymbol{y}; \boldsymbol{\theta}, \eta)$ specified (6) with trainable network $R = R_{\boldsymbol{\theta}}$ is called a *Neumann network* in [11]. Figure 1 shows a block diagram which graphically illustrates a Neumann network. The main architectural difference with Neumann networks over related unrolling approaches is the presence of additional "skip connections" that arise naturally due to the series structure. Empirical evidence in [11] suggests these additional skip connections may improve the optimization landscape relative to other architectures, and make Neumann networks easier to train.

2.1 Preconditioning

Efficiently finding a solution to the linear system (4) using an iterative method can be challenging when the matrix $\boldsymbol{X}^\top\boldsymbol{X} + \boldsymbol{R}$ is ill-conditioned. This suggests that the Neumann network, which is derived from a Neumann series expansion of the system in (4), may benefit from preconditioning.

Starting from (4), for any $\lambda > 0$ we have $(\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})\boldsymbol{\beta}^\star + (\boldsymbol{R} - \lambda\boldsymbol{I})\boldsymbol{\beta}^\star = \boldsymbol{X}^\top\boldsymbol{y}$. Applying $\boldsymbol{T}_\lambda := (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}$ to both sides and rearranging terms gives $(\boldsymbol{I} - \lambda\boldsymbol{T}_\lambda + \tilde{\boldsymbol{R}})\boldsymbol{\beta}^\star = \boldsymbol{T}_\lambda\boldsymbol{X}^\top\boldsymbol{y}$. where $\tilde{\boldsymbol{R}} = \boldsymbol{T}_\lambda\boldsymbol{R}$. Following the same steps used to derive the Neumann network gives the modified estimator

$$\widehat{\boldsymbol{\beta}}_{pc}(\boldsymbol{y}) = \sum_{j=0}^{B}(\lambda\boldsymbol{T}_\lambda(\cdot) - \tilde{R}(\cdot))^j\boldsymbol{T}_\lambda\boldsymbol{X}^\top\boldsymbol{y} \tag{7}$$

which is called a *preconditioned Neumann network* in [11]. Here $\tilde{R} = \tilde{R}_{\boldsymbol{\theta}}$ is a trainable mapping depending on parameters $\boldsymbol{\theta}$.

2.2 Connection with Proposition 1

The Neumann network estimators in (6) and (7) can be interpreted as approximating the MSE optimal reconstruction function in Proposition 1. To see this, observe that the pseudo-inverse $\boldsymbol{X}^+\boldsymbol{y} = (\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$ is given by the Neumann series $\boldsymbol{X}^+\boldsymbol{y} = \eta\sum_{k=0}^{\infty}(I - \eta\boldsymbol{X}^\top\boldsymbol{X})^k\boldsymbol{X}^\top\boldsymbol{y}$. The preconditioned Neumann network estimator $\widehat{\boldsymbol{\beta}}(\boldsymbol{y})$ has the form

$$\widehat{\boldsymbol{\beta}}(\boldsymbol{y}) = \eta\sum_{k=0}^{B}(I - \eta\boldsymbol{X}^\top\boldsymbol{X})^k\boldsymbol{X}^\top\boldsymbol{y} + \widehat{\boldsymbol{\beta}}_R(\boldsymbol{y}) \tag{8}$$

where $\widehat{\boldsymbol{\beta}}_R(\boldsymbol{y})$ collects all terms that depend on R. The preconditioned Neumann network more directly approximates $\rho^*(\boldsymbol{y})$ since the initial iterate $\tilde{\boldsymbol{\beta}}^{(0)} = \boldsymbol{T}_\lambda\boldsymbol{X}^\top\boldsymbol{y} = (\boldsymbol{X}^\top\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^\top\boldsymbol{y}$ already well-approximates $\boldsymbol{X}^+\boldsymbol{y}$ provided $\lambda > 0$ is small.

2.3 Patchwise Regularization

Here we present an extension to the Neumann network which incorporates a learned patchwise regularizer. For large images, learning an accurate regularizer may require more samples than are practical to gather due to cost or time constraints, leading to inaccurate reconstructions or overfitting.

However, empirical evidence suggests there is considerable low-rank and other subspace structure shared among small patches of natural images [13]. Redundancy and subspace structure across image

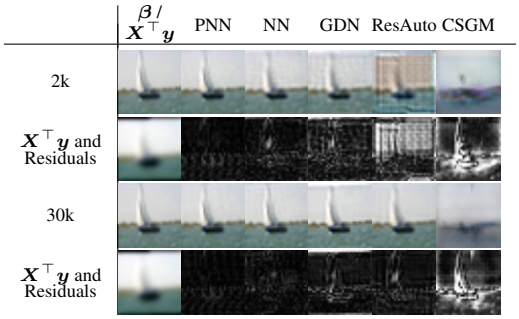


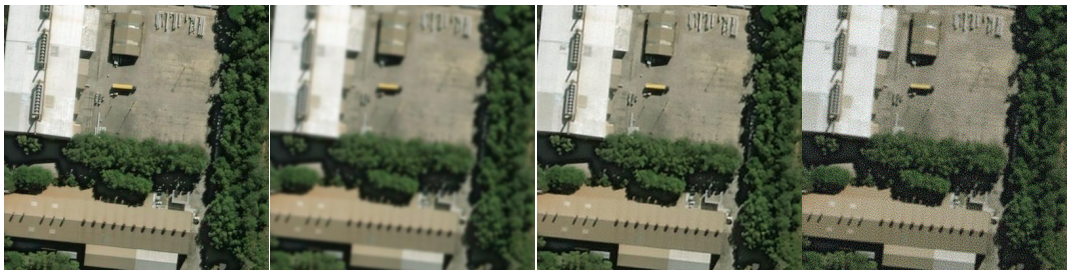
Figure 2: Reconstructions produced for the deblurring problem at different training set sizes, along with the associated residual images, which are scaled by 6x.

patches permits learning parameters of statistical models for image patches using training data, like Gaussian mixture models with low-rank covariance structure [14, 15]. We propose leveraging the highly structured nature of image patches in the learned component of the Neumann network.

Specifically, the patchwise learned regularizer first divides the input image into overlapping patches, subtracting the mean from each patch (a standard preprocessing technique in patch-based methods [16]), and passing each mean-subtracted patch through the learned component (*e.g.,* neural network). The original patch means are added to the regularizer outputs, which are recombined.

3 Experiments

Figure 2 compares the presented learning-based methods at different training set sizes. Methods that do not incorporate the forward model, like ResAuto and CSGM, appear not to perform well in the low-sample regime. We also demonstrate that patchwise regularization enables reconstruction of large images with very small training sets. In this experiment, the training set consists only of a single clean image, taken from the SpaceNet dataset [17].Test PSNR is 31.90 ± 1.42 dB for the 8x8 patchwise regularized NN, and 18.34 ± 1.31 for the full-image regularized NN across a test set of size 64. Fig. 3 contains some sample reconstructions of an image from the test set.



(a) Original (b) Blurred (c) 8x8 NN PSNR: 30.87 dB (d) Full NN PSNR: 17.75 dB

Figure 3: Single-image Training Sample Reconstruction. Local regularization enables competitive reconstruction quality even with a single training image, while unrestricted training on a single image results in a poor-quality, noisy reconstruction. The inverse problem in this case is Gaussian deblurring with kernel of size 9×9 and variance $\sigma^2 = 2$ with additive noise level 0.03.

4 Conclusion

This work explores the Neumann network architecture to solve linear inverse problems, which can be interpreted as an approximation of the MSE optimal reconstruction according to our Proposition 1. The Neumann network architecture also permits a learned patchwise regularizer, which learns the low-dimensional conditional distributions over image patches instead of the whole image. The Neumann network is empirically competitive with other state-of-the-art methods for inverse problems in imaging, and we demonstrate the ability to learn to regularize from a single training pair.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant Nos. 1740707, 1447449, and 0353079, as well as AFOSR FA9550-18-1-0166

References

[1] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis, "Compressed sensing using generative models," in *International Conference on Machine Learning (ICML)*, 2017, pp. 537–546.

[2] Tim Meinhardt, Michael Moeller, Caner Hazirbas, and Daniel Cremers, "Learning proximal operators: Using denoising networks for regularizing inverse imaging problems," in *IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1799–1808.

[3] Yaniv Romano, Michael Elad, and Peyman Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[4] Jian Sun, Huibin Li, and Zongben Xu, "Deep ADMM-Net for compressive sensing MRI," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2016, pp. 10–18.

[5] Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein, "Unrolled optimization with deep priors," *arXiv preprint arXiv:1705.08041*, 2017.

[6] Chris Metzler, Ali Mousavi, and Richard Baraniuk, "Learned D-AMP: Principled neural network based compressive image recovery," in *Advances in Neural Information Processing Systems*, 2017, pp. 1772–1783.

[7] Jonas Adler and Ozan Öktem, "Learned primal-dual reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[8] Hemant K Aggarwal, Merry P Mani, and Mathews Jacob, "MoDL: Model based deep learning architecture for inverse problems," *IEEE Transactions on Medical Imaging*, 2018.

[9] Sam Efromovich, "Conditional density estimation in a regression setting," *The Annals of Statistics*, vol. 35, no. 6, pp. 2504–2535, 2007.

[10] Guillaume Alain and Yoshua Bengio, "What regularized auto-encoders learn from the data-generating distribution," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3563–3593, 2014.

[11] Davis Gilton, Greg Ongie, and Rebecca Willett, "Neumann networks for inverse problems in imaging," *arXiv preprint arXiv:1901.03707*, 2019.

[12] Israel Gohberg and Seymour Goldberg, *Basic operator theory*, Birkhäuser, 2013.

[13] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, 2005, vol. 2, pp. 60–65.

[14] Jianbo Yang, Xuejun Liao, Xin Yuan, Patrick Llull, David J Brady, Guillermo Sapiro, and Lawrence Carin, "Compressive sensing by learning a Gaussian mixture model from measurements," *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 106–119, 2015.

[15] Antoine Houdard, Charles Bouveyron, and Julie Delon, "High-dimensional mixture models for unsupervised image denoising (HDMI)," *SIAM Journal on Imaging Sciences*, vol. 11, no. 4, pp. 2815–2846, 2018.

[16] Marc Lebrun, Miguel Colom, Antoni Buades, and Jean-Michel Morel, "Secrets of image denoising cuisine," *Acta Numerica*, vol. 21, pp. 475–576, 2012.

[17] Adam Van Etten, Dave Lindenbaum, and Todd M Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.