Shared Weights Yield Accurate Image Recovery with Scarce Training Labels: SURE Analysis

Anonymous Authors[1]

Abstract

Labeled data are often scarce in important applications, such as medicine, where expert judgments or ground truth measurements are expensive to acquire. Data scarcity impedes training standard deep neural network architectures to high accuracy, especially under compressive measurements. Recently, cross-layer *weight sharing* emerged as an effective regularization technique with promising generalization power. Leveraging Stein's Unbiased Risk Estimator (SURE) as a proxy for generalization risk, this paper analyzes train sample complexity and proves that in the low complexity regime, under certain conditions, weight sharing achieves a smaller risk than would typical optimization with weights separately varying. Supporting empirical observations are presented for linear inverse tasks including natural image denoising, deblurring, and compressed sensing MRI. These observations show that under the regime of low train sample size, weight sharing achieves a higher PSNR than non-sharing; the advantage decreases as the sample size increases. In analogy with scattering networks, frequency analysis of the unrolled network identifies weight sharing with bandpass filtering, while non-shared network alternate across layers between low and bandpass filters.

1. Introduction

Training deep neural networks typically demands abundant labeled data to achieve an acceptable generalization. Collecting valid labels, however, is costly for certain applications such as medical imaging due to physical constraints and privacy concerns. This paper deals with imaging from compressive measurements, where labels are high-quality images that in medical imaging drive diagnostic decisions.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Outside the scarce-label setting, much recent work adopts unrolled neural networks to learn the inversion map for recovering an underlying image from compressive and corrupted measurements; see e.g., (Gregor & LeCun, 2010; Zhang et al., 2017; Chan et al., 2017; Sun et al., 2016; Adler & Öktem, 2017; Diamond et al., 2017; Metzler et al., 2017; Kim et al., 2016; Schlemper et al., 2017) and references therein. The crux of unrolled schemes is to cast recovery as an image-to-image translation task mapping a low-quality image (e.g., found as linear estimate) to a high quality label image. It is essentially a cascade of alternating denoisers and data adjustment units with denoisers modeled via deep neural networks that typically vary over the iterations.

Building on that trend, the neural proximal algorithm (NPA) (Mardani et al., 2018; Parikh et al., 2014) was recently proposed; it learns a proximal map (denoiser) by training a recurrent neural network (Mardani et al., 2018). Modeling the proximal with a small residual network (ResNet) containing only a few residual blocks (RB), surprising performance was obtained in ill-posed inverse problems, such as compressed sensing MRI. Importantly, this scheme has shown itself to be much better adapted to the scarce label setting. It achieves faster training and better generalization performance than earlier unrolled schemes, where denoisers vary across iterations of the cascade.

The observed benefits of NPA seem due to *weight sharing* across denoiser layers; sharing reduces the number of parameters to estimate and apparently regularizes training. This paper aims to extensively study this phenomenon and reveals the influence of weight sharing on the generalization risk. It is worth noting that a similar phenomenon has been observed to achieve state-of-the-art quality for image superresolution tasks (Kim et al., 2016).

Contributions. In order to study the regularization effects of weight sharing we leverage the Stein's Unbiased Risk Estimator (SURE) (Stein, 1981; Donoho & Johnstone, 1995) as a proxy for the generalization MSE. In essence, SURE comprises two terms, residual sum of squares (RSS) plus *achievable* degrees of freedom (DOF), where RSS accounts for the prediction bias, while DOF captures the prediction variance. Adopting a single symmetric RB for the proximal, SURE is analyzed for the denoising task under both weight

sharing (WS) and weight changing (WC) scenaria. The results reveal an interesting trade off. WS achieves higher DOF but lower RSS than WC. The overall SURE for WC is however smaller.

We conducted extensive empirical evaluations for denoising natural images, confirming the theoretically-predicted behavior of SURE. The SURE gap between WS and WC schemes is shown to be significant for low sample sizes, but decreases as the sample size grows; eventually WS and WC agree as label scarcity abates. Further experiments for natural image deblurring and compressed sensing MRI under a realistic setting, show superior PSNR for WS vs. WC. We also compared the filtering behavior of the learned proximals for WS and WC inspired by deep scattering networks (Bruna & Mallat, 2013; Andén & Mallat, 2014). For this purpose we analyzed the frequency spectrum of different iterations that show WS performs bandpass filtering, while WC alternates between low and bandpass filtering to denoise the images.

In summary, these findings rest on several novel project contributions:

- Formal SURE analysis of WS and WC schemes comparing SURE terms analytically.
- Rigorous experimental evaluation with natural images for denoisng and deblurring tasks, and with realistic Knee MRI images for compressed sensing.
- Frequency analysis to assess the filtering behavior of learned proximal networks for WS and WC schemes.

Notations. In the paper $(.)^{\mathsf{H}}$, $\|\cdot\|_2$, $\mathbb{E}$, tr, $\circ$, and I_n refer to the matrix Hermitian, ℓ_2-norm, statistical expectation, trace, composition operator, and $n \times n$ identity matrix.

2. Preliminaries and Problem Statement

Consider the linear system

$$y = \Phi x_* + v \tag{1}$$

with $\Phi \in \mathbb{C}^{m\times n}$ and $m \leq n$, where the Gaussian noise $v \sim \mathcal{N}(o, \sigma^2/n)$ captures the noise and unmodeled dynamics. Suppose the unknown image x_* lies in a low-dimensional manifold. No information is known about the manifold besides the training samples $\mathcal{X} := \{x_i\}_{i=1}^N$ drawn from it, and the corresponding noisy observations $\mathcal{Y} := \{y_i\}_{i=1}^N$. Given a new undersampled observation y, the goal is to quickly recover a plausible image $\hat{x}$ that is close to x_*.

The stated problem covers a wide range of image recovery tasks. For instance, for image denosing $\Phi = I$ (Dabov et al., 2007; Dong et al., 2013; Zhang et al., 2017), for image deblurring Φ (Zoran & Weiss, 2011; Venkatakrishnan et al., 2013) signifies the local convolution operator, for image superresolution (Romano et al., 2017b; Bruna et al., 2015) Φ is the downsampling operator that averages out nonoverlapping image regions to arrive at a low resolution image, and for compressed sensing MRI (Lustig et al., 2007) Φ refers to the subsapmed Fourier operator.

2.1. Neural proximal algorithm

In order to invert the linear system 1 a variation of the proximal algorithm advocated in (Mardani et al., 2018) is adopted. Given a pre-trained proximal operator $\mathcal{P}_\psi$ (Parikh et al., 2014) modeled via a neural network, the overall iterative procedure evolves according to the state-space equations

$$\begin{aligned} &\textbf{step 1.} && \mathrm{s}^{\mathrm{t}+1} = \mathrm{g}\big(\mathrm{x}^{\mathrm{t}}; \mathrm{y}\big) \\ &\textbf{step 2.} && \mathrm{x}^{\mathrm{t}+1} = \mathcal{P}_\psi(\mathrm{s}^{\mathrm{t}+1}) \end{aligned}$$

for a fixed number of iterations, i.e., $t = 1, \ldots, T$. The first step invokes a linear operation that assures the state consistency with the measuremnts y. The second step executes the proximal mapping for denoising the image estimate. The recursion starts with the initial linear estimate $x_0 = \Phi^{\mathsf{H}} y$ as the match filtered input y. For the first step, we can perform a first-order gradient step as in (Mardani et al., 2018), or (preferably) a second-order least-squares step if computationally affordable. They are expressed as follows (for a learnable step size α):

- Gradient step

$$g(x^t; y) := \alpha \Phi^{\mathsf{H}} y + (I - \alpha \Phi^{\mathsf{H}} \Phi) x^t.$$

- Least-squares step

$$g(x^t; y) := (\alpha \Phi \Phi^{\mathsf{H}} + (1-\alpha) I)^{-1} \left(\alpha \Phi^{\mathsf{H}} y + (1-\alpha) x^t\right).$$

2.2. Proximal modeling with neural networks

A residual network (ResNet) (He et al., 2016) with K residual blocks (RB) is adopted to model the proximal map $\mathcal{P}_\psi$. Adopting the ReLU activation $\sigma(x) = D(x) \cdot x$, where $D(x) = 1_{x \geq 0}$, the outer iteration t (mapping x_{t-1} to x_t) can be decomposed as follows,

- $h_0^t = g(x^{t-1}; y)$
- $h_{k+1}^t = h_k^t + W_k^{\mathsf{H}} \sigma(\bar{W}_k h_k^t), \quad k = 1, \ldots, K$
- $x_t = h_K^t$.

Neural proximal algorithm is recurrent in nature to mimic the fixed point iteration for traditional proximal algorithm (Parikh et al., 2014). We thus shared weights $\{W_k\}_{k=1}^K$ for different outer iterations t. When $\bar{W}_k = W_k$ we call the model symmetric residual block, which provides further regularization through weight sharing.

However, we could also learn different weights $\{W_k^t\}_{k=1}^K$ for different t, which changes the hidden layers at t-th iteration to

$$h_{k+1}^t = h_k^t + W_k^{t,\mathsf{H}}\sigma(\bar{W}_k^t h_k^t), \quad k = 1, \dots, K \tag{2}$$

Pseudo linear representation. We adopt a pseudo-linear representation for the activation, where D_k is a diagonal mask matrix with binary values for ReLU. Note, during inference, the mask D_k is dependent on the input data examples, while W_k is fixed. Accordingly, we can write $h_{k+1}^t = M_k^t h_k^t$, where $M_k^t = I + W_k^{\mathsf{H}} D_k^t \bar{W}_k$. The overall proximal map M_t at t-th iteration then admits

$$x_{t+1} = \underbrace{M_t^K \dots M_t^2 M_t^1}_{:=M_t} s_{t+1}. \tag{3}$$

Apparently, the map M_t is input data dependent due to nonlinear activation.

Unrolling the T outer iterations of the proximal algorithm and the K inner iterations of the K RBs, the end-to-end recurrent map with input y_i yields

$$\hat{x}_i := x_T^i := (M_T \circ g) \circ \dots \circ (M_1 \circ g)(\Phi^{\mathsf{H}} y_i). \tag{4}$$

Training. training process then optimizes the network weights for the shared weights $\mathcal{W}_{sw} := \{W_k\}_{k=1}^K$, and changed weights $\mathcal{W}_{cw} := \{W_k^t,\ t = 1, \dots, T\}_{k=1}^K$ to fit $\hat{\mathcal{X}} := \{\hat{x}_i\}$ to $\mathcal{X} := \{x_{*,i}\}$ for the training population using pixel-wise empirical loss

$$\text{minimize}_{\mathcal{W}} \quad \frac{1}{N}\sum_{i=1}^N \ell(\hat{x}_i, x_{*,i}). \tag{5}$$

3. Risk Analysis

In order to ease the analytical exposition for the generalization risk we focus on the denoising task ($\Phi = I$),

$$y = x_* + v, \qquad v \sim \mathcal{N}(0, \sigma^2/n) \tag{6}$$

with $\|x_*\| = 1$. The derivation presented here could be generalized to an arbitrary Φ with noise following an exponential distribution using ideas presented in (Eldar, 2009). Let $x_T = h(y)$ denote the prediction obtained via neural proximal algorithm for the function $h(\cdot)$ in 4 with a test sample y as input argument. Assume h is weakly differentiable. The Stein's unbiased risk estimator (Stein, 1981; Donoho & Johnstone, 1995) for $h(y)$ is then expressed as

$$\text{SURE}(y) = -\sigma^2 + \underbrace{\|h(y;\Phi) - y\|_2^2}_{\text{RSS}(y)} + 2\frac{\sigma^2}{n}\underbrace{\nabla_y \cdot h(y)}_{\text{DOF}(y)}, \tag{7}$$

where ∇ is the divergence operator. Equation 7 is an unbiased estimate of the mean-squared-error (MSE) satisfying

$$\text{MSE} := \mathbb{E}\left[\|h(y) - x_*\|^2\right] = \mathbb{E}\left[\text{SURE}(y)\right]. \tag{8}$$

SURE in 7 comprises two main terms. The residual sum of squares (RSS) measures the error between the corrupted and denoised input, while the DOF measures the *achievable* degrees of freedom for the denoiser.

We are interested in SURE for a population of test samples $\{y\}$ in the inference phase. Accordingly, we define the average SURE

$$\text{MSE} = \mathbb{E}[\text{SURE(y)}] \tag{9}$$

In the sequel we analyze SURE for weight sharing and weight changing scenaria to compare their generalization power.

Lemma 1 *For the considered neural proximal algorithm with the end-to-end nonlinear map J_T and the model 6 it holds that (likewise for $\tilde{J}_T$)*

$$\begin{aligned} \text{DOF} &= \mathbb{E}\text{tr}(J_T) \\ \text{RSS} &= 1 + \sigma^2 + \mathbb{E}\left[\|x_T\|_2^2\right] - 2\mathbb{E}\text{tr}(y^{\mathsf{H}} x_T) \end{aligned}$$

A natural question then would pertain to the behavior of DOF and RSS terms as well as the overall SURE for WS and WC scenario. This is the subject of next section.

3.1. SURE comparison

To facilitate the SURE analysis, we consider the proximal map with a single RB. For WS we assume that RB is symmetric, meaning that the deconvolution operation is simply the convolution transposed. The corresponding proximals then simply admit $M_t = I + W_0^{\mathsf{H}} D_t W_0$ at t-th iteration for shared weight scheme, and $\tilde{M}_t = I + W_t^{\mathsf{H}} \tilde{D}_t \bar{W}_t$ for changed weight scheme (see 3). Accordingly, from 4 one can easily derive the end-to-end map (for shared weight) relating the iteration outputs x_{t+1} to the initial estimate $x_0 = y$ as

$$x_{t+1} = \underbrace{\Big(\sum_{k=0}^{t}\big(\prod_{j=1}^{k}(1-\alpha)(M_{t-j} - I)\big)\alpha M_{t-k-1}\Big)}_{:=J_{t+1}} y$$

Similarly, substituting M_t with $\tilde{M}_t$ results in $\tilde{J}_{t+1}$ for the changed weight scheme. One can then expand J_T as a linear combination of J_I's where there are in total 2^T different index choices for I, and upon defining $j = |I|$, I can be associated with an index array $(i_1, \dots, i_j)$. Similarly, we expand $\tilde{J}_T$ in the same form with the same coefficients. Accordingly,

$$\begin{aligned} J_T &= \textstyle\sum_I c_I J_I, \\ J_I &= W_0^T D_{i_j} W_0 W_0^T \dots W_0 W_0^T D_{i_1} W_0, \\ c_I &= (1-\alpha)^j \alpha^{T-j} \end{aligned}$$

Likewise

$$\tilde{J}_I = W_{i_j}^T \tilde{D}_{i_j} \bar{W}_{i_j} W_{i_{j-1}}^T \dots \bar{W}_{i_2} W_{i_1}^T \tilde{D}_{i_1} \bar{W}_{i_1}.$$

Theorem 1 *(SURE) For the single RB proximal model, and under Assumption 1 on the training, suppose that J_T and v are statistically independent and $\mathbb{E}[\|x_T\|_2^2] = 1$. If $\tilde{p} \leq p(1 + \sum_{s=1}^{j} \sigma_N^{2s}(1+\epsilon)^s)$, $1 \leq j \leq T$ for some small constant ϵ that $|\epsilon| \leq \delta \ll 1$, it then holds that*

$$\mathbb{E}\mathrm{tr}(J_T) \geq \mathbb{E}\mathrm{tr}(\tilde{J}_T)$$

and as a result

$$\begin{aligned} \mathrm{DOF}^{\mathrm{ws}} &\geq \mathrm{DOF}^{\mathrm{wc}}, \\ \mathrm{RSS}^{\mathrm{ws}} &\leq \mathrm{RSS}^{\mathrm{wc}}, \\ \mathrm{SURE}^{\mathrm{ws}} &\leq \mathrm{SURE}^{\mathrm{wc}} \end{aligned}$$

It is useful to recognize that for the setting where the path sparsities for WS and WC are close, i.e., $p \approx \tilde{p}$, then using smaller sample size for training (larger σ_N as per Assumption 1), $\mathbb{E}\mathrm{tr}(J_T) \geq \mathbb{E}\mathrm{tr}(\tilde{J}_T)$ indeed holds. As a results Theorem 1 predicts larger DOF, smaller RSS, and smaller SURE for WS than WC. Our extensive empirical evaluations in Section 4.2 corroborate this observation (see e.g., Figs. 1, 2, 3). In addition, for large sample sizes where $\sigma_N \to 0$, DOFs, RSSs, and SUREs for WS and WC will coincide, which is again quite evident empirically.

4. Empirical Evaluations

Extensive experiments were performed to assess our findings for three different inverse imaging tasks: denoising, deblurring, and compressed sensing. The first two tasks were tested on natural images while compressed sensing was examined on MRI data. In particular, we aimed to address the following important questions:

Q1. How would the RSS, DOF, and SURE behave empirically for WS and WC schemes?

Q2. How would MSE/PSNR behave as a function of the train sample size for WS and WC schemes?

Q3. Is there explainable filtering behaviour of the learned denoisers for WS and WC schemes?

4.1. Network Architecture and Training

To address the above questions, we adopted a ResNet with 2 RBs where each RB consists of two convolutional layers with 3×3 kernels and a fixed number (128) of feature maps, that were followed by batch normalization (BN) and ReLU activation. ResNet is used in the feature domain, and thus we add a convolutional layer with 3×3 kernels that lift up the image from previous iterations to 128 feature maps. Similarly, ResNet is followed by a convolutional layer with 1×1 kernels that lifts off the feature maps to create the next estimate. We used the Adam optimizer (Kingma & Ba, 2014) with the momentum parameter 0.9 and initial learning rate varying across the experiments. Training was performed with TensorFlow and PyTorch interface on NVIDIA Titan X Pascal GPUs with 12GB RAM. PSNR (dB) is used as the figure of merit that is simply related to MSE as $PSNR = -10\log_{10}(MSE)$ since the images are normalized to unity.

4.2. Denoising

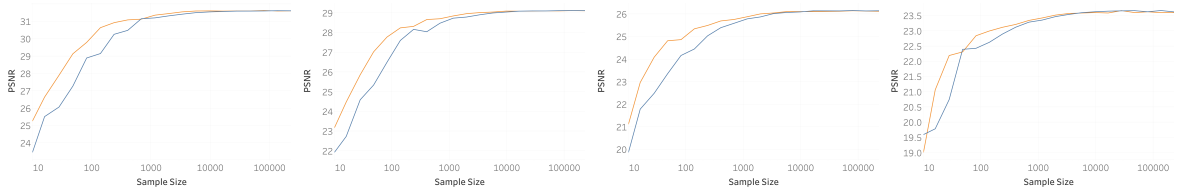
This section addresses Q1 and Q2 for natural image denoising task, where $\Phi = I$.

Dataset. 400 natural images of size 481×321 were selected from the Berkeley image segmentation dataset (a.k.a BSD68) (Martin et al., 2001). Patches of size 40×40 were extracted as labels, resulting in 230400 training samples. 68 full images were chosen for test data.

The ResNet architecture described before was adopted with $K = 2$ RBs and $T = 4$ iterations. It is trained for 50 epochs with minibatch size 256. The initial learning rate was annealed by a factor of 10 at the 40-th epoch. We run experiments independently (with random initialization) for several initial learning rates, namely $\{0.0075, 0.005, 0.0025, 0.001, 0.00075, 0.0005, 0.00025, 0.0001, 0.000075, 0.00005\}$, and pick the one leading to the best PSNR on test data during the last epoch. The aforementioned experiments were repeated for various noise levels $\sigma \in \{15, 25, 50, 100\}$. Moreover, the experiments were repeated with and without weight sharing.

We assess SURE, DOF, and RSS for NPA trained with sample sizes within the range $[10, 230400]$ (logarithmically spaced). It is first observed that the SURE estimate is in perfect agreement with the test MSE (or PSNR) when having the true labels available for validation purposes. We thus plot the PSNR evolution in Fig. 1 as the sample size grows for training. The orange line corresponds to the result obtained with weight sharing and the blue line to without. For all noise levels, we observe a consistent benefit from using weight sharing in sample sizes less than 1K. Interestingly after 1K they coincide and no benefit is observed for changing weights even for very large sample sizes in the order of 10^5. Note also that the non-smooth behavior of the curve is mainly attributed to Adam optimizer that may not necessarily converge to the globally optimum network weights.

Error bars for the individual SURE components including DOF and RSS are also plotted in Fig. 2 and 3, respectively. The upper (res. bottom) rows correspond to WC (res. WS). Fig. 2 depicts the evolution of normalized RSS, namely $\frac{1}{\sigma^2}\|y - h(y)\|^2$ over train sample size. Similarly, Fig. 3 plots the DOF $\nabla_y \cdot h(y)$. The blue dots correspond to 68 test image samples. Box-and-whisker plots also depict RSS percentile. It is first observed that for both WS and WC scenaria, DOF (res. RSS) tend to be increasing (res. decreasing) with the train sample size, where it finally sat-



($\sigma = 15$) ($\sigma = 25$) ($\sigma = 50$) ($\sigma = 100$)

Figure 1. Effects of weight sharing and sample size on the performance of NPA in the task of image denoising. Different columns correspond to different noise levels. Each panel depicts the test PSNR of NPA as function of train sample size. The orange line corresponds to the result obtained with weight sharing and the blue line to without.

urates at a limiting value that is identical for both WS and WC. The DOF for WS scheme, however, ramps off quickly, suggesting that fewer samples are required to construct the bases and attain the degrees of freedom embedded in the network. On the contrary, RSS would drop quickly for WS, which contributes to small SURE values.

Our second observation compares RSS and DOF values for WS and WC. It appears that, under different noise regimes, WS consistently achieves larger DOF. The achieved RSS however is much smaller which renders the overall SURE (or MSE) smaller for WS in low train sample complexity regimes. In addition, upon using suffient train samples, RSS converges to unity for all noise levels, which corresponds to $\|y - h(y;\Phi)\|_2^2 = n\sigma^2$. We explain this by noting that any sensible denoising algorithm should output estimated images within the noise level $n\sigma^2$ of the corrupted image.

4.3. Deblurring

This section addresses Q2 for natural image deblurring. The sensing matrix Φ is a linear operator that convoles an image with a Gaussian kernel of standard deviation 1.6. We prepared data as described for denoising, except that we extracted patches of size 50×50. The same ResNet as in the denoising case was adopted to model the proximal. However, instead of encouraging data consistency with a gradient step as for denoising (see 2.1), the full least square problem is simply solved after each proximal step. As a result, the state variable update is modified as

$$s_{t+1} = \left(\Phi\Phi^{\mathsf{H}} + \alpha I\right)^{-1}\left(\Phi^{\mathsf{H}} y + \alpha x_t\right). \qquad (10)$$

It is worth commenting that the approach of tackling a general image restoration problems by repeatedly denoising and solving a least-squares problem is a common practice in image processing; see e.g., (Romano et al., 2017a; Zoran & Weiss, 2011; Chan et al., 2017).

We train the architecture in the same way described in the previous subsection. The experiments are repeated for two noise levels $\sigma \in \{\sqrt{2}, 2\}$ at different panels. Each panel depicts the test PSNR of neural proximal algorithm as a function of training sample size. The orange (res. blue) line corresponds to WS (res. WC). We observe a consistent benefit from using weight sharing in sample sizes smaller than 50K.

4.4. Compressed sensing MRI

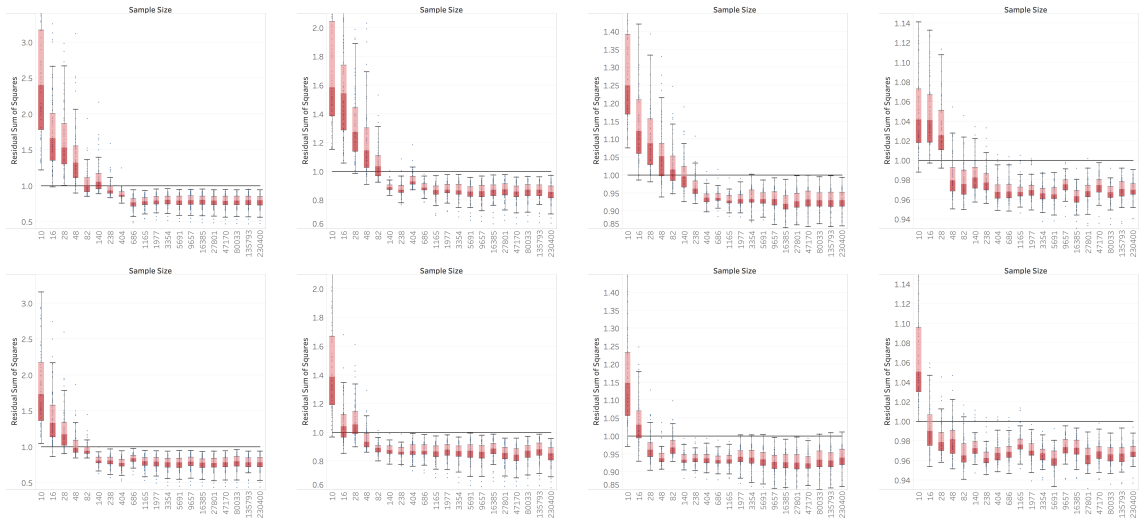
To further investigate Q2, we consider also the task of compressed sensing (Donoho, 2006) for MRI reconstruction. Looking at the linear model 1, compressed sensing (CS) assumes there are typically much less measurements than the unknown image pixels, i.e., $m \ll n$. A prime example for CS is reconstruction of MR images (Lustig et al., 2007), that is widely adopted in the clinical scanners. In essence, the MR scanner acquires a fraction of Fourier coefficients (k-space data) of the underlying image across various coils. We focused on a single-coil MR acquisition model, where for a patient the acquired k-space data admits

$$y_{i,j} = [\mathcal{F}(x)]_{i,j}, \quad (i,j) \in \Omega \qquad (11)$$

Here, $\mathcal{F}$ refers to the 2D Fourier transform, and the set Ω indexes the sampled Fourier coefficients. Just as in conventional CS MRI, we selected Ω based on variable-density sampling with radial view ordering that is more likely to pick low frequency components from the center of k-space (Lustig et al., 2007). Only 20% of Fourier coefficients were collected.

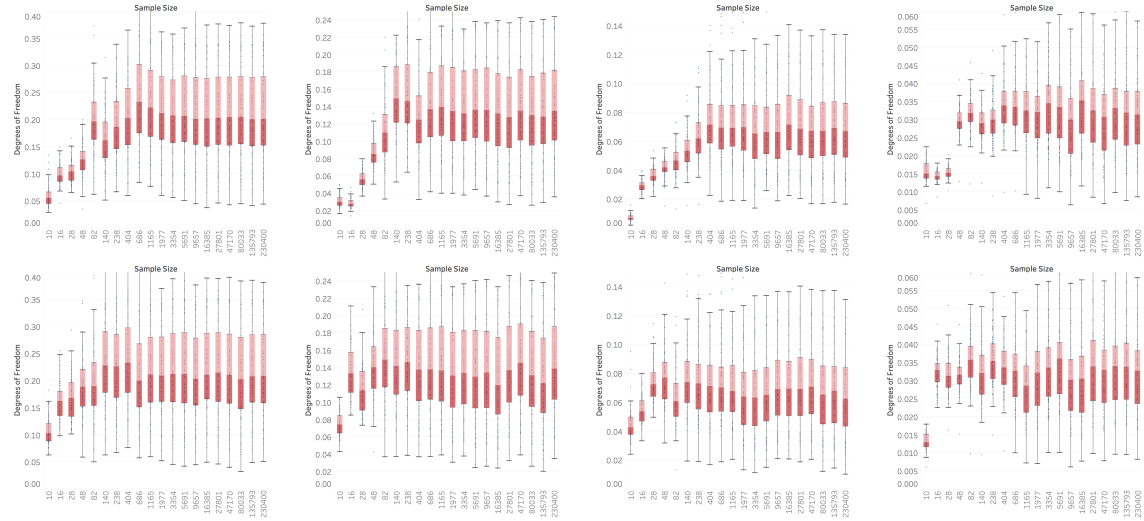
Dataset. It includes 19 subjects scanned with a 3T GE MR750 whole body MR scanner at the Stanford Lucile Packard Childrens Hospital. Fully sampled sagittal images were acquired with a 3D FSE CUBE sequence with proton density weighting including fat saturation. Other parameters include FOV=160mm, TR=1550 (sagittal) and $2,000$ (axial), TE=25 (sagittal) and 35 (axial), slice thickness 0.6mm (sagittal) and 2.5mm (axial). The dataset is publicly available at (mri). Each image is a complex valued 3D volume of size $320 \times 320 \times 256$. Axial slices of size 320×256 were considered as the input for train and test. 16 patients are used for training ($5,120$ image slices) and 3 patients for test (960 image slices).

Neural proximal algorithm with $T = 5$ iterations was run with minibatch of size 4. For any train sample size, training is performed for various learning rates $3 \times \{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$, choosing the one achieving the



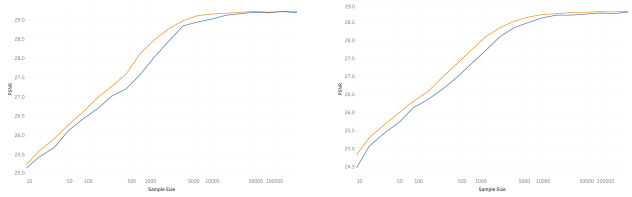
($\sigma = 15$) ($\sigma = 25$) ($\sigma = 50$) ($\sigma = 100$)

Figure 2. Effects of weight sharing and sample size on the residual sum of squares (RSS) of the NPA. Each column of panels corresponds to a different noise level. The upper row of panels corresponds to non-shared weights while the bottom to shared weights. Each panel depicts the normalized RSS $\frac{1}{\sigma^2}\|y - h(y;\Phi)\|_2^2$ of NPA as function of training sample size. The blue dots correspond to the 68 test images on which the RSS was computed. The box and whisker depict the percentiles of the RSS values.



($\sigma = 15$) ($\sigma = 25$) ($\sigma = 50$) ($\sigma = 100$)

Figure 3. Effects of weight sharing and sample size on the degrees of freedom (DOF) of the NPA. Each column of panels corresponds to a different noise level. The upper row of panels corresponds to non-shared weights while the bottom to shared weights. Each panel depicts the degrees of freedom $\nabla_y \cdot h(y;\Phi)$ of NPA as function of training sample size. The blue dots correspond to the 68 testing images on which the DOF was computed. The box and whisker depict the percentiles of the DOF values.



($\sigma = \sqrt{2}$) ($\sigma = 2$)

Figure 4. The effect of weight sharing and sample size on the performance of NPA in the task of natural image deblurring.

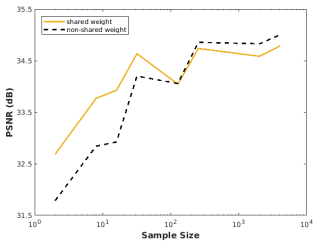


Figure 5. Effects of weight sharing and sample size on the performance of NPA for CS-MRI with 5-fold undersampling of the k-space data. The panel depicts the test PSNR as a function of training sample size. The orange line corresponds to the result obtained with weight sharing and the blue line to without.

highest PSNR. The input and output were complex-valued images of the same size and each included two channels for real and imaginary components. The input image $x_0 = \Phi^{\mathsf{H}} y$ was indeed the inverse 2D FFT of the k-space data where the missing frequencies were filled with zeros. It was thus severely contaminated with aliasing artifacts. The benefits of weight sharing for small sample sizes is evident, when using only a few MR images for training would lead to around 1 dB gain compared with the weight changing scheme. The gap kept decreasing with the train sample size and finally after 10^2 samples the weight changing scheme led to an slight improvement, possibly due to the larger representation capacity. Note also that compared with the denoising and deblurring experiments, the gap disappears for a smaller sample sizes, as train images are of large dimension with 320×256 training pixels.

4.5. Filtering interpretation of proximals

Along with Q3, it is of interest to *explain* how the cascade of learned proximals of WS and WC contribute to recover the input image from errors. To do so, we focus on the natural image denoising described in Section 4.2 with $T = 4$ proximal iterations. Recall the t-th proximal network mapping s_{t+1} to x_{t+1} through a ResNet with 2RBs. We focus on the first convolutional layer with 128 kernels collected as $\{f_{t,i}\}_{i=1}^{128}$ per iteration t. For visualization purposes, we propose to compute the two-dimensional Fourier transform for each of filters $\{f_{t,i}\}_{i=1}^{128}$ and then to sum over all filters the magnitude of the Fourier coefficients. We repeat this process for T iterations in WC case and the single set of filters in WS.

The results are shown in Fig. 6 for the noise level $\sigma = 100$. The first four panel columns represent the weights obtained in four iterations of WC network, while the fifth column represents WS. Each row corresponds to a different sample size. It is observed that for WS at high sample sizes the filters converge to a spectrum associated with a bandpass filter. The pattern observed for WC however is interesting; the odd iterations converge to a lowpasss, while the even iterations converge to bandpass filters. This is reminiscent of the scattering transform (Bruna & Mallat, 2013; Andén & Mallat, 2014), where a cascade of lowpasss and bandpass operations is applied.

In contrast with scattering networks however, our *learned* proximal network applies lowpasss filtering followed by bandpass filtering. The scattering transform applies several lowpasss filters, and then a highpasss filter at the last layer. We also observe that the shared weights converge to the final spectrum at approximately 686 examples, while the non-shared case requires more than 5691 samples to converge. Moreover, for WS case the filters in the first two iterations are very similar to their final versions already at 686 examples, meaning that the filters in the first iterations are trained first.

5. Conclusions

This paper studies weight sharing as a regularization technique for training unrolled proximal neural networks appearing in image translation tasks with scarce train labels. The Stein's Unbiased Risk Estimator (SURE) is adopted as a surrogate for the generalization risk, and analysis are carried out to assess SURE components for weight sharing and weight changing scenaria. Under mild conditions, it is proved that WS achieves smaller SURE value than WC. These findings are corroborated with extensive empirical observations indicating SURE values for WS are consistently smaller than WC under various regimes of train sample sizes and noise levels. In addition, experiments for natural image denoising and deblurring as well as compressed sensing of MR images show the benefit of WS over WC especially for low train sample sizes. Moreover, the frequency analysis of the learned filters *explains* the role of different units of the cascaded neural proximal network on image refinement. All in all, this is the first attempt to apply SURE for generalization risk analysis of deep neural networks in image recovery tasks.

There are still important avenues to explore that are left for future research. One such avenue pertains to generalizing the advocated SURE analysis to arbitrary sensing matrices.

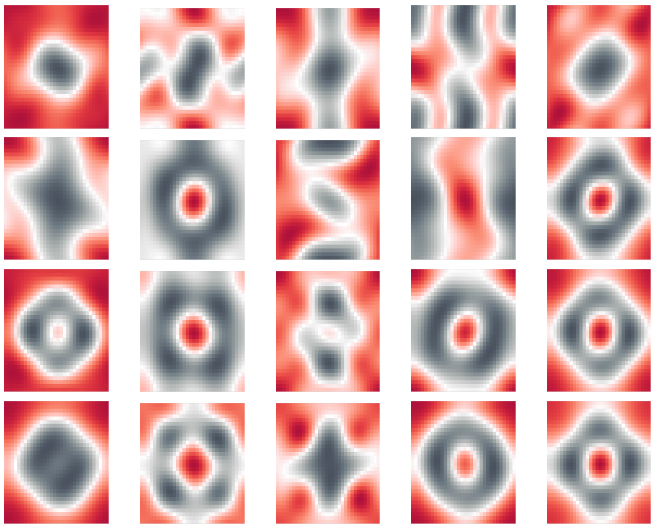


Figure 6. Frequency visualization of the learned proximals in the shared and non-shared case. The first four columns correspond to the weights obtained in the 4 iterations of the non-shared weights. The fifth column corresponds to the shared-weights case. Rows from top to bottom are for $10, 686, 5691, 135793$ train samples. Each panel depicts the summation of the magnitudes of the 2D Fourier transforms of the filters. The noise level is fixed to $\sigma = 100$. The color coding associates large magnitudes to black, and small ones to red.

Another one includes understanding the link between early stopping and weight sharing for training unrolled neural networks.

References

[online] http://mridata.org/fullysampled/knees.html. URL http://mridata.org/.

Adler, J. and Öktem, O. Learned primal-dual reconstruction. *arXiv preprint arXiv:1707.06474*, 2017.

Andén, J. and Mallat, S. Deep scattering spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

Bruna, J. and Mallat, S. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

Bruna, J., Sprechmann, P., and LeCun, Y. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666*, 2015.

Chan, S. H., Wang, X., and Elgendy, O. A. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2017.

Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8): 2080–2095, 2007.

Diamond, S., Sitzmann, V., Heide, F., and Wetzstein, G. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.

Dong, W., Zhang, L., Shi, G., and Li, X. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013.

Donoho, D. L. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

Donoho, D. L. and Johnstone, I. M. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

Eldar, Y. C. Generalized sure for exponential families: Applications to regularization. *IEEE Transactions on Signal Processing*, 57(2):471–481, 2009.

Gregor, K. and LeCun, Y. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pp. 399–406. Omnipress, 2010.

He, K., Zhang, X., Ren, S., and Sun, J. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645. Springer, 2016.

Kim, J., Kwon Lee, J., and Mu Lee, K. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1637–1645, 2016.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lustig, M., Donoho, D., and Pauly, J. M. Sparse MRI: The application of compressed sensing for rapid MR imaging. *Magnetic Resonance in Medicine*, 58(6):1182–1195, December 2007.

Mardani, M., Sun, Q., Vasawanala, S., Papyan, V., Monajemi, H., Pauly, J., and Donoho, D. Neural proximal gradient descent for compressiveimaging. In *Advances in Neural Information Processing Systems*, 2018.

Martin, D., Fowlkes, C., Tal, D., and Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pp. 416–423. IEEE, 2001.

Metzler, C., Mousavi, A., and Baraniuk, R. Learned D-AMP: Principled neural network based compressive image recovery. In *Advances in Neural Information Processing Systems*, pp. 1770–1781, 2017.

Parikh, N., Boyd, S., et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

Romano, Y., Elad, M., and Milanfar, P. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017a.

Romano, Y., Isidoro, J., and Milanfar, P. RAISR: rapid and accurate image super resolution. *IEEE Transactions on Computational Imaging*, 3(1):110–125, 2017b.

Schlemper, J., Caballero, J., Hajnal, J. V., Price, A., and Rueckert, D. A deep cascade of convolutional neural networks for MR image reconstruction. In *Proceedings of the 25st Annual Meeting of ISMRM, Honolulu, HI, USA*, 2017.

Stein, C. M. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pp. 1135–1151, 1981.

Sun, J., Li, H., Xu, Z., et al. Deep ADMM-net for compressive sensing MRI. In *Advances in Neural Information Processing Systems*, pp. 10–18, 2016.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, pp. 945–948. IEEE, 2013.

Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

Zoran, D. and Weiss, Y. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pp. 479–486. IEEE, 2011.