Deep Network Approach for Blind Calibration in Compressive Sensing

Gore Kao
Carnegie Mellon University
Pittsburgh, PA
gorek@andrew.cmu.edu

Abstract

The domain of Compressive Sensing (CS) is a signal processing technique for reconstructing a signal with a small number of measurements by utilizing a prior sparsity constraint. In an actual application setting, the calibration of a system may not be known (hence blind), such as the hardware implementation and acquisition. Therefore the topic of Blind Calibration within compressive sensing deals with added unknown parameters within a system that perturb the true measurements. This work presents a method through a deep learned network approach to perform calibration of the measurements as well reconstruction in a compressed sensing framework. The method builds upon the LISTA [1] framework to learn the calibration matrix successively across network layers as well as extract an estimate of the reconstructed signal. We show that with some network adjustments and additional fine tuning that this approach has potential in learning the parameters needed for calibration and compressive sensing.

1 Introduction

Compressive Sensing (CS) aims to reconstruct a signal by solving an underdetermined system of linear equations typically posed as $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{x} \in \mathbb{R}^N$ is the original signal of interest, $\mathbf{y} \in \mathbb{R}^M$ is the observed signal, and $\mathbf{A} \in \mathbb{R}^{M \times N}$ is the measurement operator. Traditional methods solve this formulation under a sparsity constraint of the signal $\mathbf{x}$ and utilize this prior to recover the original signal from far fewer samples required by the Nyquist–Shannon sampling theorem. Classical iterative as well unrolled learned methods have seen previous success when applied to traditional Compressive Sensing (CS) problems [1, 2, 3].

In a physical application setting, measurements are collected via physical sensors and can suffer from calibration errors. In addition to the measurement matrix, there is also an unknown perturbation vector, $\mathbf{g}$, that alters the original measurements. This turns the standard compressed sensing formulation into one that is ill posed. To address the calibration issue we consider an approach that relies on several observations within the uncalibrated system, utilizing the variance of the observed data to learn the parameters for calibration. Consider a system that captures p total observations within the uncalibrated compressive sensing setting. We can formulate each observation $\mathbf{y}_i$ and original vector $\mathbf{x}_i$ along with the calibration vector $\mathbf{g}$ and known measurement matrix $\mathbf{A}$ as follows in equation 1.

$$\mathbf{y}_i = \text{diag}(\mathbf{g})\mathbf{A}\mathbf{x}_i, \; \forall i = 1, \ldots, p \tag{1}$$

The diag operator here creates a square diagonal matrix from the unknown calibration vector $\mathbf{g}$. The goal then is to recover the original signal $\mathbf{x}$ and calibration vector $\mathbf{g}$ given several observations $\mathbf{y}$. We can pose equation 1 in a more concise matrix form considering $\mathbf{Y} \in \mathbb{R}^{M \times p}$ and $\mathbf{X} \in \mathbb{R}^{N \times p}$ shown in equation 2.

$$\mathbf{Y} = \text{diag}(\mathbf{g})\mathbf{A}\mathbf{X} \tag{2}$$

2 Related Work

There have been previous notable works utilizing deep networks to learn the parameters for CS reconstruction. Zhang et al. propose a novel structured deep network, ISTA-Net [3], inspired by the Iterative Shrinkage-Thresholding Algorithm (ISTA) [2] for compressive sensing reconstruction of natural images. The authors cast ISTA into a deep convolutional network and solve the proximal mapping associated with the sparsity-inducing regularizer using nonlinear transforms. Using ISTA-Net, the parameters for reconstruction can be learned rather than handcrafted. The results show ISTA-Net can outperform state-of-the art optimization and network based compressive sensing methods.

When adding the problem of blind calibration to compressive sensing, Gribonval et al. propose a convex optimization approach by minimizing the objective formulated in equation 3 [4].

$$(\hat{\mathbf{X}}, \hat{\mathbf{h}}) := \underset{\mathbf{X},\mathbf{h}}{\arg\min} \|\mathbf{X}\|_1 \text{ s.t. } \text{diag}(\mathbf{h})\mathbf{Y} = \mathbf{A}\mathbf{X},\ \mathbf{1}^\top \mathbf{h} = M \tag{3}$$

Here $\mathbf{h}$ is the diagonal of the inverse calibration matrix and the additional trace constraint is set to avoid a trivial zero solution. In addition, Shen et al. present a method of solving the problem using a cost function on a suitable manifold and perform a conjugate gradient method [5], rather than using a convex optimization approach.

Following the topic of blind calibration, Shulke et al. propose a message passing algorithm called calibration approximate message passing (cal-AMP) [6] that can address sensor calibrated imperfections. They express the blind sensor calibration problem in a CS framework, posing it as an inference problem using Bayes' rule and an *a priori* knowledge of the probability distribution of both the signal and calibration parameters. This leads to an *a posteriori* distribution of the solution combined with iterative updates of the signal estimates. Extending the work of cal-AMP [6] from the offline case, Gabrie et al. propose an online addition in which calibration is refined step by step for new incoming measurements [7].

More generally Gregor et al. propose a framework, dubbed LISTA (Learned ISTA), for very efficient reconstruction of sparse codes [1]. The method produces very fast approximations for reconstruction using a feed forward predictor trained with non-linear layers at a fixed depth. In a classical approach to solving sparsely coded problems, iterative methods have been employed with success that aim to minimize a quadratic reconstruction error with L_1 penalty term. This has shown to be slow in many applications and thus LISTA provides an efficient alternative through a deep unrolled network formulation. Our implementation therefore builds upon the LISTA framework to add a learned parameter for the calibration matrix as well as the reconstructed sparse code.

3 Method

The LISTA framework has shown great success as a learned method for solving a posed CS problem. To extend this idea to a blind calibration situation, this approach jointly learns estimates for reconstruction as well as the parameters for the inverse calibration matrix. Ideally we can efficiently learn a coupled version of diag$(\mathbf{h})$ through a fixed number of layers to produce a result diag$(\mathbf{h})$diag$(\mathbf{g}) = \mathbf{I}$. The identity result here would mean that the system has been perfectly recalibrated. The translation of the original iterative soft thresholding algorithm (ISTA) to the unrolled version of LISTA is shown in Figure 1 from the original work by Gregor et al.

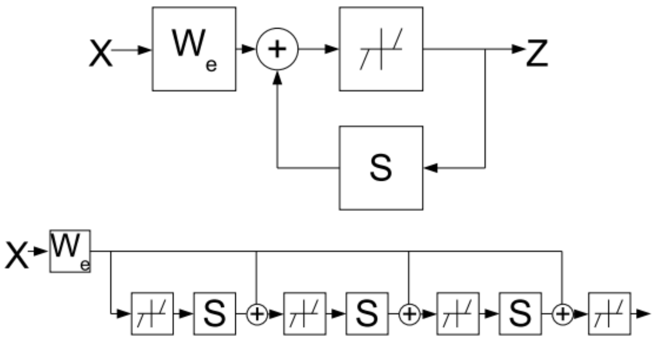

Figure 1: **Top:** Iterative Soft Thresholding Algorithm (ISTA). In the original implementation, $\mathbf{W_e}$ is the transpose dictionary matrix $\mathbf{W_d}$ and $\mathbf{S}$ is $\mathbf{W_d^T W_d}$. A shrinkage step is applied at each step which can be denoted as a function $z_\alpha(\cdot)$ with threshold parameter α. **Bottom:** A time unrolled version of ISTA, dubbed LISTA. Here the parameters $\mathbf{W_e}$ and $\mathbf{S}$ are learned instead. Original figure depiction shown by Gregor et al. [1].

In this problem, we consider the calibration settings of unknown gains and phase shifts. Included within the original LISTA framework at each layer is an additional parameter to learn the inverse calibration vector $\mathbf{h}$. Therefore at each layer an estimate $\hat{\mathbf{X}}$ is computed as well as a second estimate diag$(\mathbf{h})$. Following the constraint shown in equation 3, we can obtain two estimates, diag$(\hat{\mathbf{h}})\mathbf{Y}$ and $\mathbf{A}\hat{\mathbf{X}}$, such that the loss can then be calculated from equation 4.

$$\mathcal{L}(\hat{\mathbf{h}}, \hat{\mathbf{X}}) = L_p(\hat{\mathbf{h}}, \hat{\mathbf{X}}) = \|\text{diag}(\hat{\mathbf{h}})\mathbf{Y} - \mathbf{A}\hat{\mathbf{X}}\|_p \tag{4}$$

The p in equation 4 denotes a certain L_p loss rather than the number of observations. In this implementation we use the L_1 loss $(p = 1)$ to promote a more sparse estimate to diag$(\hat{\mathbf{h}})$. The forward propagation for Blind-LISTA is shown in Algorithm 1.

Algorithm 1 Blind LISTA: forward propagation

```
**Blind-LISTA::fprop**$(\mathbf{Y}, \mathbf{A}, \mathbf{S}, \mathbf{h}, \theta)$
$\mathbf{B} = (\text{diag}(\mathbf{h})\mathbf{A})^\top \mathbf{Y};\ \mathbf{S} = \mathbf{I} - \mathbf{B}\mathbf{A}$
Initialize $\hat{\mathbf{X}}^{(0)} = z_\theta(\mathbf{B}), \hat{\mathbf{h}}^{(t)} = \mathbf{h}$
Parameters $\hat{\mathbf{X}}^{(t)}, \mathbf{C}^{(t)}, \mathbf{B}, \hat{\mathbf{h}}^{(t)}$ saved for back prop.
  for $t = 1$ to $T$ do
      $\mathbf{C}^{(t)} = \mathbf{B} + \mathbf{S}\hat{\mathbf{X}}^{(t-1)}$
      $\hat{\mathbf{X}}^{(t)} = z_\theta(\mathbf{C}^{(t)})$
      $\hat{\mathbf{y}}_1^{(t)} = \text{diag}\ (\hat{\mathbf{h}}^{(t)})\mathbf{Y}$
      $\hat{\mathbf{y}}_2^{(t)} = \mathbf{A}\hat{\mathbf{X}}^{(t)}$
  end for
```

The loss is calculated for each layer, similarly following equation 4, as $L = \|\hat{\mathbf{y}}_1^{(t)} - \hat{\mathbf{y}}_2^{(t)}\|_1$ and summed up across all layers. Automatic differentiation [8] and backpropagation is then done across the mentioned parameters.

3.1 Measurement and Data Generation

The measurement matrix, $\mathbf{A}$, is generated by constructing a Discrete Fourier Transform (DFT) matrix of size $N \times N$ [9] and then randomly sampling M rows from the matrix to produce a measurement matrix of size $M \times N$. The ground truth signals, $\mathbf{X}$ of size $N \times p$, are generated from IID bernoulli-gaussian distribution with parameter q set for the distribution. The perturbation vector, $\mathbf{g}$, is generated from a random sample of gain in range $(0, \kappa)$ and random sample of the phase shift in range $[0, 2\pi]$. A batch of observations are then synthetically created as $\mathbf{Y} = \text{diag}(\mathbf{g})\mathbf{A}\mathbf{X}$.

3.2 Evaluation

We consider two possible evaluation metrics. The first approach directly compares $\mathrm{diag}(\mathbf{h})\mathrm{diag}(\mathbf{g})$ to the identity matrix $\mathbf{I}$ averaged across each layer and compared across each iteration to see the how close $\mathrm{diag}(\mathbf{h})\mathrm{diag}(\mathbf{g})$ can be reconstructed to identity. The mean squared error calculation is used here to calculate error between $\mathrm{diag}(\mathbf{h})\mathrm{diag}(\mathbf{g})$ and $\mathbf{I}$.

The second approach compares the normalized vectors of $1/\mathbf{h}$ and $\mathbf{g}$ to calculate a dot product. Since $\mathrm{diag}(\mathbf{h})$ is ideally the inverse of $\mathrm{diag}(\mathbf{g})$, we consider the inverse value in the vector $\mathbf{h}$ and the dot product ideally should approach 1. To deal with the shift variant possibility from $\mathbf{h}$ consider the metric formulation in equations 5 and 6.

$$\mathbf{h}_i^* = \Big[\mathrm{diag}(\hat{\mathbf{h}})\mathbf{A}\Big]_i, \ \forall i = 1, \dots, M \text{ columns} \tag{5}$$

$$\max\left(\left\langle \frac{\frac{1}{\mathbf{h}_i^*}}{\|\frac{1}{\mathbf{h}_i^*}\|}, \frac{\mathbf{g}}{\|\mathbf{g}\|} \right\rangle\right), \ \forall i = 1, \dots, M \tag{6}$$

3.3 Training Parameters

We set $M = 64, N = 256, p = 2048$. The probability parameter is set $q = 0.3$ for generating the bernoulli-gaussian distributed data and $\kappa = 50$ for generating $\mathbf{g}$. The thresholding parameter in the soft shrinkage function is set to 0.1 and we fix the number of network layers to be 8. Initial learning rate is 0.01 and halved at each epoch for 10 epochs. The batch size is set to 16. Following the input recalibration vector $\mathbf{h}$ in Algorithm 1, $\hat{\mathbf{h}}^{(t)}$ is initialized to be a vector of 1's so that $\mathrm{diag}(\hat{\mathbf{h}}^{(t)})$ is the identity matrix for each layer. Implementation is done in PyTorch [10] and is publicly available [1].

4 Results

The evaluation metrics mentioned in section 3.2 are shown in Figures 2 and 3 for both methods respectively. The metrics are plotted as averages across each layer for each iteration of propagation.

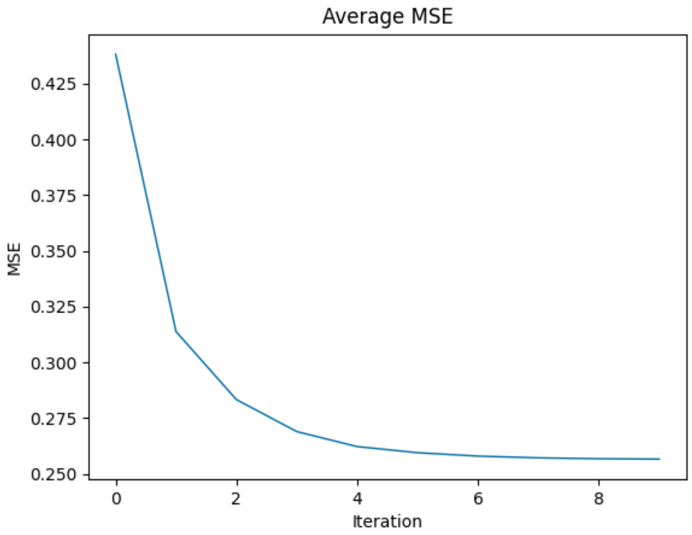

Figure 2: MSE reconstruction comparing $\mathrm{diag}(\mathbf{h})\mathrm{diag}(\mathbf{g})$ and $\mathbf{I}$.

[1] https://github.com/Gkao03/Blind-Calibration

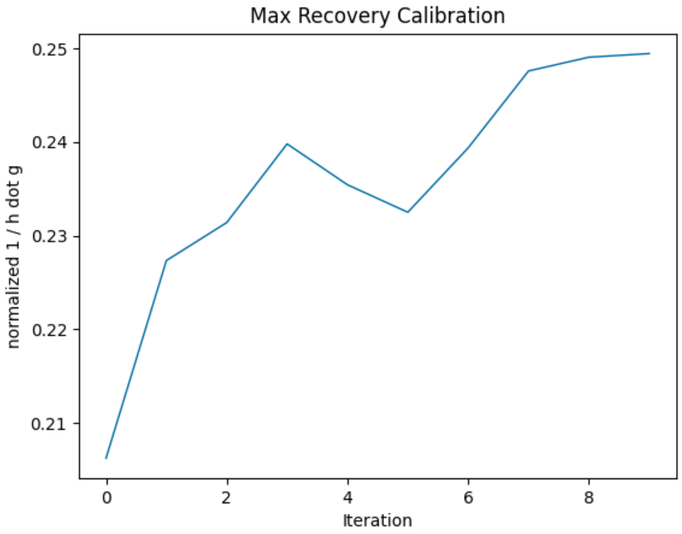

Figure 3: Recovery of dot product between normalized inverse of $\mathbf{h}$ and $\mathbf{g}$.

5 Discussion

The evaluation results show a trend towards convergence of decreasing error between $\mathrm{diag}(\mathbf{h})\mathrm{diag}(\mathbf{g})$ and $\mathbf{I}$ as well as increasing max recovery vector calibration between $\mathbf{h}$ and $\mathbf{g}$. It should be noted that running additional experiments showed the recovery calibration metric to be inconsistent at times compared to the MSE metric and generally plateaued around a value of 0.25. We provide both metrics here for consideration under the given problem formulation so future work can be compared fairly.

The initialization method for $\mathbf{h}$ also has a non-trivial effect on the recovery of the compressed signal as well as calibration vector. For this implementation a simple identity initialization is used, however future experiments may benefit from better initialization if an *a prior* distribution setting for pertubation $\mathbf{g}$ is known.

References

[1] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

[2] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” *SIAM journal on imaging sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[3] J. Zhang and B. Ghanem, “Ista-net: Interpretable optimization-inspired deep network for image compressive sensing,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1828–1837, 2018.

[4] R. Gribonval, G. Chardon, and L. Daudet, “Blind calibration for compressed sensing by convex optimization,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2713–2716, IEEE, 2012.

[5] H. Shen, M. Kleinsteuber, Ç. Bilen, and R. Gribonval, “A conjugate gradient algorithm for blind sensor calibration in sparse recovery,” in *2013 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–5, IEEE, 2013.

[6] C. Schülke, F. Caltagirone, and L. Zdeborová, “Blind sensor calibration using approximate message passing,” *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2015, no. 11, p. P11013, 2015.

[7] M. Gabrié, J. Barbier, F. Krzakala, and L. Zdeborová, “Blind calibration for compressed sensing: state evolution and an online algorithm,” *Journal of Physics A: Mathematical and Theoretical*, vol. 53, no. 33, p. 334004, 2020.

[8] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, “Automatic differentiation in pytorch,” 2017.

[9] J. Vora and A. Rajwade, “Compressive signal recovery under sensing matrix errors combined with unknown measurement gains,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5105–5109, IEEE, 2021.

[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” *Advances in neural information processing systems*, vol. 32, 2019.