Preserving Optimal Step Size of unfolded ISTA

Zheng ZHOU[1,a) Rabi YAMADA[2,b)] Tomoya SAKAI[2,c)]

Abstract

We propose a strategy of sparse dictionary learning together with automatic tuning of the step size in the iterative soft thresholding algorithm (ISTA) via its unfolding to a computation graph. Our strategy makes ISTA trainable by back-propagation so as not only to fit the sparse dictionary to a training dataset but also to improve the speed of convergence. Our novelty is in the use of back-propable eigen decomposition in the unfolded ISTA to enable the step size tuning. Our work simplifies the sparse solver and outperforms the state-of-the-art unfolding techniques.

1. Introduction

Unfolding iterative algorithms into DNNs have won great attention in dealing with convex optimization problems these years [1, 2, 3, 4, 5]. LISTA[6] and its variants, *e.g.*, LISTA-CP[7] and ALISTA[8], are derived from unfolding the iterative soft thresholding algorithm (ISTA)[9]. They focus on training DNN weights and thresholds to speedup the convergence of ISTA.

In this paper, we propose tuning the step size of ISTA in training. The step size can be impressively important for speeding up the approximations of the sparse coding with just a few iterations or DNN layers.

This paper begins with our short survey on unfolding of ISTA. We experimentally evaluate and discuss the influences of keeping the optimal step size to unfolded iterative algorithms.

2. Unfolding ISTA

Consider the constrained minimization problem for a smooth $f\colon \mathbb{R}^n \longrightarrow \mathbb{R}$ and a convex function $g\colon \mathbb{R}^n \longrightarrow \mathbb{R}$ which may be non-differentiable:

$$\underset{x}{\text{Minimize}}\, f(x) + g(x) \tag{1}$$

If $f(x) = \frac{1}{2}\|b - Ax\|_2^2$ and $g(x) = \lambda\|x\|_1$, where $b \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, λ is a positive scaler, then the minimization problem in Eq. (1) is called the LASSO regression which we focus on in this paper. There exists a classical iterative shrinkage algorithm called ISTA (iterative shrinkage thresholding algorithm) for the LASSO regression:

$$x^{(k+1)} = \text{soft}\big(x^{(k)} + \alpha A^T\big(b - Ax^{(k)}\big),\ \alpha\lambda\big). \tag{2}$$

Here, $\text{soft}\colon \mathbb{R}^n \longrightarrow \mathbb{R}^n$ is a shrinkage operator defined as $\text{soft}(x,\ \theta) = \text{sign}(x)\max(|x| - \theta,\ 0)$. A positive constant α is the step size, of which optimal value is analytically known to be $1/\sigma_{max}(A^T A)$, *i.e.*, the reciprocal of the smallest Lipschitz constant L of ∇f.

The past decade has seen a renewed importance in taking iterative algorithms as DNNs for accelerate sparse solvers. LISTA, as learning-based ISTA, shows a representational way of unfolded iterative algorithms as neural networks. In LISTA, Eq. (2) is rewritten as

$$x^{(k+1)} = soft\big(W_e b + Sx^{(k)},\ \theta\big) \tag{3}$$

where $W_e = \frac{1}{L}A^T$, $S = I - \frac{1}{L}A^T A$, $\theta = \frac{\lambda}{L}$. The trainable parameters of LISTA are $\{(W_e, S, \theta)\}$.

To speedup the convergence of LISTA, an unshared and partial weight coupling algorithm (LISTA-CP) was proposed. LISTA-CP represents the ISTA as

$$x^{(k+1)} = \text{soft}\big(x^{(k)} - W_e^{(k)}\big(b - Ax^{(k)}\big), \theta^{(k)}\big) \tag{4}$$

with the trainable parameters $\left\{\big(W_e^{(k)},\ \theta^{(k)}\big)\right\}$.

LISTA-CP can be further simplified into ALISTA, which computes a shared weight W_e from a given dictionary A to use it as $W_e^{(k)} = \gamma^{(k)} W_e$, and just trains $\gamma^{(k)}$ and $\theta^{(k)}$.

Most studies tended to concentrate on making weight

[1] Fuzhou University
[2] Nagasaki University
[a)] N181120083@fzu.edu.cn
[b)] bb52120323@ms.nagasaki-u.ac.jp
[c)] tsakai@nagasaki-u.ac.jp

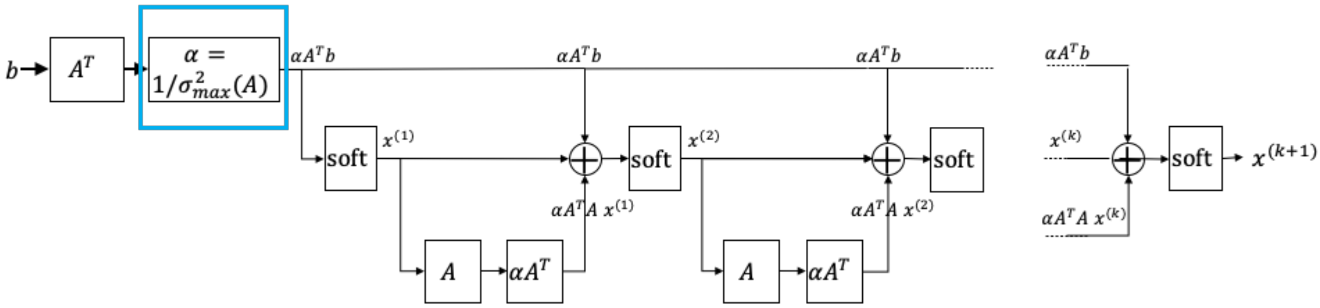


Figure. 1 block diagram of the tuning of step size method in unfolded ISTA. The blue block is the scheme of automatic compute step size α by using SVD, $\sigma^2_{max}(A)$ is the largest singular square value of A. The computed step size also used for soft shrinkage operator.

structure of the original ISTA to be holistic such as LISTA or LISTA-CP. As usual, the (smallest) Lipschitz constant L is taken as the largest eigenvalue of A^TA, and such L is included in weight or threshold set as the variable initialization for the model of the learned iterative algorithm. We argue that the step size in the original ISTA would be a critical point in the approximation of ground truth sparse solutions. If the step size was too small, the number of iterations, or the number of layers after unfolding ISTA, would be very large. Conversely, if the step size was too large, or L was too small, ISTA might be unstable and diverge. It turns out that a suitable and minimal Lipschitz constant L would be incredibly important in speeding-up estimations of sparse solutions.

We propose keeping the step size α to be the analytically optimal largest singular square value $\sigma^2_{max}(A)$ with the dictionary A being trainable. We fit this strategy into two different DNN architectures. The first one originally comes from the unfolded ISTA, and trainable parameters are $\{(A, \lambda)\}$, ISTA with the tuning of step size is named TSS-UISTA. The block diagram of the method is represented in figure 1. The second one is derived from the shared weight structure of LISTA-CP, and trainable parameters are $\{(W_e, A, \lambda)\}$, named TSS-LISTA-CP.

3. Automatic tuning of step size scheme

We introduce an automatic tuning of the step size scheme as follows and applied it to unfolded ISTA. Since step size plays an indispensable role when seeking fast approximation. As mentioned before, the optimal step size is always used as initial values in unfolded iterative algorithms. We focus on keeping computing optimal step size to pursue better performance of approximation of unfolded ISTA.

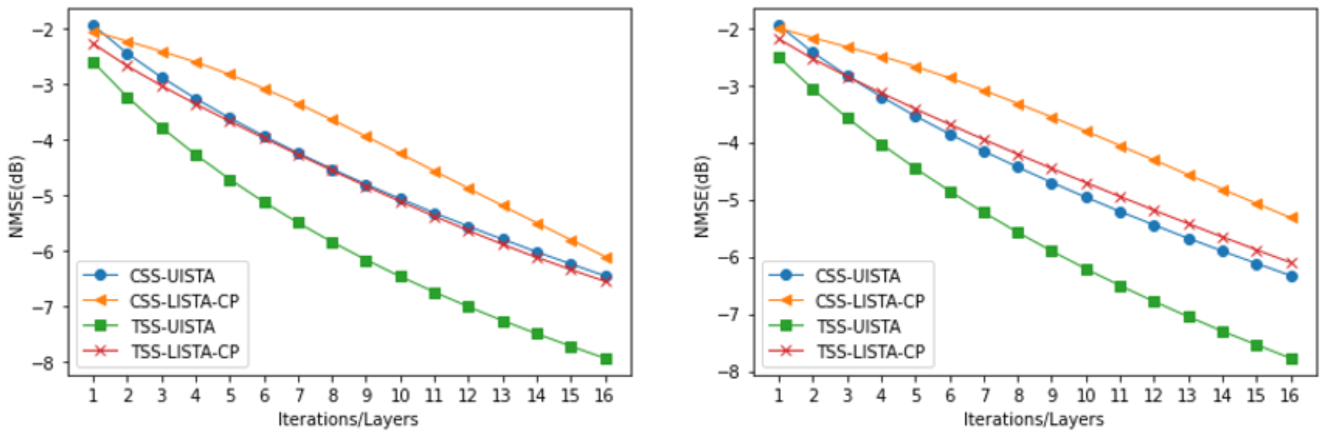
Firstly, we introduce TSS-UISTA. The main ideas are from dictionary learning and tuning of step size. With a trainable parameter A, the largest singular square value of A would be always changed during training. We adopt an automatic computed largest singular square value as step size, with this automatic step size scheme, we always keep the best step size. Trainable parameters are $\{(A, \lambda)\}$. Dictionary learning also helps simplify the weight structure compared to LISTA. We believe these solutions will help to save the number of iterations or layers to get a precise sparse solution.

Secondly, we apply this scheme to the shared weights LISTA-CP. The parameters are $\{(W_e, A, \lambda)\}$. Compared to the shared weights LISTA-CP to use a fixed A in their algorithm, a trainable A with the tuning of step size are preferred in our algorithm TSS-LISTA-CP.

4. Simulation Experiments

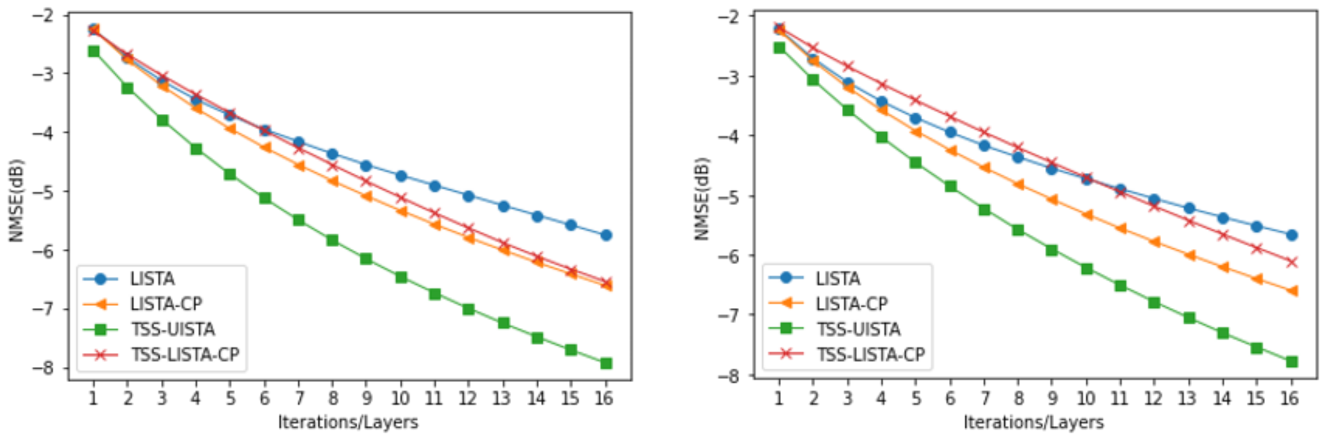
Our simulation set up bears a close resemblance to what Chen has done in [7]. The shape of A is $m = 250, n = 250$. A is sampled from standard Gaussian distribution, $A_{ij} \sim N(0,\ 1/m)$, and each column is normalized to unit length. In this experiment, the supports of ground truth equal to 10% of n. The Loss function is simply NMSE. The Optimizer is a simple stochastic gradient descent (SGD). λ is 0.16. We use TensorFlow-GPU 2.2.0-rc3 to model our experiments and the operating environment is based on google colab. All the hyperparameters are optimal in our simulation.

In the first set of experiments, two groups of simulations are created to evaluate our algorithms. To highlight the importance of computed step size, we compare the performances of using constant step size or using



(a) SNR = ∞ (b) SNR = 30

Figure. 2 Validation of constant step size α and computed step size α



(a) SNR = ∞ (b) SNR = 30

Figure. 3 Validation of performance between different algorithms

computed step size both in UISTA and LISTA-CP. UISTA with constant step size is simply named CSS-UISTA, LISTA-CP with a constant step size is named CSS -LISTA-CP. Results are shown in figure 2, step k can be equal to $x^{(2)}$, so the x axis is the value of k.

As shown in figure 2. TSS-UISTA showed the best performance of all methods. Compared to the constant step size (CSS-UISTA) and computed step size scheme (TSS-UISTA), it is clearly demonstrated that computed step size is necessary for unfolded ISTA. TSS-LISTA-CP also showed better than CSS-LISTA-CP.

In the second set of experiments, we compared LISTA, LISTA-CP, TSS-UISTA, and TSS-LISTA-CP. Results can be viewed in Figure 3. Interestingly, TSS-UISTA is outperformed than the other three schemes, even in the first iteration or layer. In figure 3(a), TSS-LISTA-CP shows better than LISTA after 6 iterations or layers and close to LISTA-CP with the increase of iterations or layers. As expected, TSS-UISTA shows a clear advantage over approaching ground truth.

5. Conclusions

We proposed using tuning of the step size strategy in unfolded ISTA. The step size would be a critical point in the approximation of ground truth sparse solution. Our research pointed out the importance of using sparse dictionary learning together with automatic tuning of the step size in the unfolded ISTA and we achieved great performances in simulation experiments. The results of the current study were limited by the time-consuming SVD calculations. Future work will concentrate on reducing the time-consuming problem and applying this method to real applications.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP19H04177.

References

[1] Zhangyang Wang, Qing Ling, and Thomas Huang. Learning

deep l0 encoders. In AAAI Conference on Artificial Intelligence, pages 2194–2200, 2016.

[2] Mark Borgerding, Philip Schniter, and Sundeep Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 2017.

[3] Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *IEEE CVPR*, 2018.

[4] Hillel Sreter and Raja Giryes. Learned convolutional sparse coding. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2191–2195. IEEE, 2018.

[5] Raja Giryes, Yonina C Eldar, Alex Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. IEEE Transactions on Signal Processing, 2018.

[6] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In Proceedings of the 27th International Conference on International Conference on Machine Learning, pages 399–406. Omnipress, 2010.

[7] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. arXiv preprint arXiv:1808.10038, 2018.

[8] Liu, J., Chen, X., Wang, Z., Yin, W.: ALISTA: analytic weights are as good as learned weights in LISTA. In: International Conference on Learning Representations(2019). https://openreview.net/forum?id=B1lnzn0ctQ

[9] I. Daubechies, M. Defrise, and C. D. Mol, An iterative thresholding algorithm for linear inverse problems with a sparsity constraint, Comm. Pure Appl. Math., 57 (2004), pp. 1413–1457.