FAST APPROXIMATION OF NON-NEGATIVE SPARSE RECOVERY VIA DEEP LEARNING

Youye Xie[1], *Zifan Wang*[2], *Weiping Pei*[3], *and Gongguo Tang*[1]

[1]Department of Electrical Engineering, Colorado School of Mines, USA
[2]Ming Hsieh Department of Electrical Engineering, University of Southern California, USA
[3]Department of Computer Science, Colorado School of Mines, USA

ABSTRACT

Non-negative sparse recovery refers to recovering non-negative sparse source signals from linear observations. This model arises naturally in many image processing applications such as super-resolution and image inpainting. In this paper, we propose two efficient neural networks for fast approximation of non-negative sparse recovery. We also derive upper bounds on network sizes measured by the numbers of layers and neurons to achieve a specified approximation error. Numerical experiments demonstrate the effectiveness and robustness of the proposed networks and show their potential in solving more complicated signal recovery problems with non-stationary transformation process and noisy observation.

Index Terms— Deep learning, algorithm approximation, non-negative sparse recovery, compressive sensing

1. INTRODUCTION

1.1. Algorithm Approximation

Deep learning has found numerous applications [1, 2], among which one important field is algorithm approximation [3]. The basic idea is to unfold an iterative algorithm and transform the iteration process into a series of network layers. The network parameters are then trained with back-propagation. For example, [3] and [4] solve a sparse recovery problem without the non-negative constraint by approximating the Iterative Soft-Thresholding Algorithm (ISTA) [5] and Alternating Direction Method of Multipliers (ADMM) algorithm with neural networks, respectively. [6, 7] address the non-negative matrix factorization problem through algorithm approximation and [8] approximates the optimization algorithm for the network training. [9] considers the non-negative sparse recovery problem but their network contains a special integrator component and the networks in this paper have a unique skip connection design which can be seen as a variation of the skip connection in ResNet [10]. More importantly, few of the algorithm approximation literatures quantify the relation between the system performance and the network size as we do in this work.

1.2. Non-Negative Sparse Recovery

Throughout this paper, we consider the non-negative least square problem for sparse recovery which occurs naturally in many machine learning and image processing tasks [11, 12].

$$\begin{aligned} &\underset{\boldsymbol{x}}{\text{minimize}}\ \frac{1}{2}||\mathbf{A}\boldsymbol{x}-\boldsymbol{y}||_2^2 \\ &\text{subject to}\ \ \boldsymbol{x}\in\mathbf{R}^n_{\geq 0}\end{aligned} \tag{1.1}$$

where $\boldsymbol{y}=\mathbf{A}\boldsymbol{x}^*\in\mathbf{R}^m$ for some ground truth signal $\boldsymbol{x}^*\in\mathbf{R}^n_{\geq 0}$ is the measurement vector, and $\mathbf{A}\in\mathbf{R}^{m\times n}(m<n)$ is the sensing matrix. Since $m<n$, this problem is underdetermined and ill-posed. There are an infinite number of solutions $\hat{\boldsymbol{x}}$ such that $f(\hat{\boldsymbol{x}})=1/2||\mathbf{A}\hat{\boldsymbol{x}}-\boldsymbol{y}||_2^2=f(\boldsymbol{x}^*)$. Fortunately, in many scenarios, the ground truth signal is both sparse and non-negative. More precisely, we assume $\boldsymbol{x}^*$ only contains at most s positive entries and we call $\boldsymbol{x}^*$ a s-sparse vector.

Proposition 1. *[13, 14] If the matrix $\mathbf{A}\in\mathbf{R}^{m\times n}$ satisfies the self-regularizing condition and $(3/\tau^2,s)$-restricted eigenvalue condition, the convex optimization* (1.1) *has a unique s-sparse solution with overwhelming probability.*

A. Self-regularizing condition: there exists a constant $\tau>0$ such that

$$\max\left\{\sigma:\exists\boldsymbol{h}\in\mathbf{R}^m,||\boldsymbol{h}||_2\leq 1,\text{such that}\ \frac{\mathbf{A}^T\boldsymbol{h}}{\sqrt{m}}\succeq\sigma\boldsymbol{1}\right\}\geq\tau. \tag{1.2}$$

B. $(3/\tau^2,s)$-restricted eigenvalue condition: given τ from (1.2) *and sparsity s, the following inequality holds*

$$\min_{\substack{J\subseteq\{1,\dots,n\},\\|J|\leq s}}\ \min_{\substack{\delta\in\mathbf{R}^n\neq 0,\\||\delta_{J^c}||_1\leq 3/\tau^2||\delta_J||_1}}\frac{||\mathbf{A}\delta||_2}{\sqrt{m}||\delta_J||_2}>0 \tag{1.3}$$

where $|J|$ measures the cardinality of J and δ_J is the vector δ with all but entries whose indices $\notin J$ set to zero.

When $\mathbf{A}$ satisfies the conditions in Proposition 1, solving (1.1) equals to solving a non-negative sparse recovery problem and we assume $\mathbf{A}$ satisfies Proposition 1 throughout the

This work was supported by NSF grant CCF-1704204 and the DARPA Lagrange Program under ONR/SPAWAR contract N660011824020. Emails: {youyexie, weipingpei, gtang}@mines.edu, zifanw@usc.edu.

978-1-5386-6249-6/19/$31.00 ©2019 IEEE

paper. [13] shows that if the entries of $\mathbf{A}$ are sampled from i.i.d sub-Gaussian distribution on $\mathbf{R}_{x\geq 0}$, Proposition 1 is satisfied with overwhelming probability.

A classical approach to solve convex optimizations like (1.1) with a simple constrained set is the projected gradient descent (PGD) algorithm. Due to the convexity of the objective function and the uniqueness of the solution, starting from an arbitrary initial point, e.g. $\boldsymbol{x}_0 = \mathbf{0}$, PGD is guaranteed to converge to the ground truth solution. Given the sensing matrix $\mathbf{A}$ and the observation $\boldsymbol{y}$, PGD alternates between a gradient descent step and a projection step

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \mathrm{ReLU}(\boldsymbol{x}_k - \alpha(\mathbf{A}^T\mathbf{A}\boldsymbol{x}_k - \mathbf{A}^T\boldsymbol{y})) \\ &= \mathrm{ReLU}((\mathbf{I} - \alpha\mathbf{A}^T\mathbf{A})\boldsymbol{x}_k + \alpha\mathbf{A}^T\boldsymbol{y}) \\ &:= \mathrm{ReLU}(\mathbf{W}\boldsymbol{x}_k + \mathbf{S}\boldsymbol{y})\end{aligned} \tag{1.4}$$

where α is the step size and ReLU represents the projection onto the non-negative orthant defined as $\mathrm{ReLU}(\boldsymbol{x}) = \arg\min_{\boldsymbol{y}\in\mathbf{R}^n_{\geq 0}} ||\boldsymbol{x} - \boldsymbol{y}||_2 = \max\{0, \boldsymbol{x}\}$.

PGD can be accelerated with improved convergence rate to obtain the following accelerated projective gradient descent (APGD) [15, 16]:

$$\begin{aligned}\boldsymbol{x}_{k+1} &= \mathrm{ReLU}\left[\boldsymbol{y}_k - \alpha\nabla f(\boldsymbol{y}_k)\right] \\ \boldsymbol{y}_{k+1} &= \boldsymbol{x}_{k+1} + \gamma(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k).\end{aligned} \tag{1.5}$$

Substituting $\boldsymbol{y}_k = \boldsymbol{x}_k + \gamma(\boldsymbol{x}_k - \boldsymbol{x}_{k-1})$ and $\nabla f(\boldsymbol{y}_k) = \mathbf{A}^T\mathbf{A}\boldsymbol{y}^k - \mathbf{A}^T\boldsymbol{y}$ into $\boldsymbol{x}_{k+1}$ yields

$$\begin{aligned}\boldsymbol{x}_{k+1} = \mathrm{ReLU}\Big\{ &\left[(1+\gamma)\mathbf{I} - \alpha(1+\gamma)\mathbf{A}^T\mathbf{A}\right]\boldsymbol{x}_k \\ &+ \left(\alpha\gamma\mathbf{A}^T\mathbf{A} - \gamma\mathbf{I}\right)\boldsymbol{x}_{k-1} + \alpha\mathbf{A}^T\boldsymbol{y}\Big\} \\ := \mathrm{ReLU}&\left(\mathbf{W}_1\boldsymbol{x}_k + \mathbf{W}_2\boldsymbol{x}_{k-1} + \mathbf{S}\boldsymbol{y}\right).\end{aligned} \tag{1.6}$$

When initialized at $\boldsymbol{x}_0 = 0$, the PGD and APGD have block diagrams shown in Fig. 1 (a) and 2 (a).

The rest of the paper is organized as follows. In Section 2, we propose two neural networks for non-negative sparse recovery and derive bounds on the network sizes to achieve a specified reconstruction error. Section 3 is devoted to numerical experiments and the paper is concluded in Section 4.

2. DEEP LEARNING APPROXIMATION

In this section we propose two efficient neural networks for non-negative sparse recovery inspired by the algorithmic pipelines of PGD and APGD. In our design, only the ReLU activation function is used. We refer to the networks inspired by PGD and APGD algorithms as the learned projective gradient descent (LPGD) network and the learned accelerated projective gradient descent (LAPGD) network respectively. Specifically, we unfold the PGD and APGD algorithms and make their parameters, $\mathbf{W}$, $\mathbf{W}_1$, $\mathbf{W}_2$ and $\mathbf{S}$, trainable. The block diagrams and network structures of LPGD and LAPGD networks are shown in Fig. 1 and 2. $\boldsymbol{x}_k$ is the output of the k-th ReLU layer and we call the network whose output is $\boldsymbol{x}_k$ the k-depth network.

[redacted]

(a) The block diagram of the LPGD network.

[redacted]

(b) The network structure of the LPGD network.

Fig. 1: The learned projective gradient descent (LPGD) network.

In sparse recovery, samples $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ from a specific distribution are fed to networks to learn the mapping from $\boldsymbol{y}_i$ to $\boldsymbol{x}_i$ which is denoted as $g(\boldsymbol{y}_i, \mathbf{W})$, where $\mathbf{W}$ designates all the trainable parameters. Given N samples, the training process tries to minimize the Euclidean distance between the predicted and ground truth signals, $Loss(\mathbf{W}) = \frac{1}{N}\sum_{i=1}^{N}||\boldsymbol{x}_i^* - g(\boldsymbol{y}_i, \mathbf{W})||_2^2$. In addition, to better illustrate how the network size affects its performance, we derive the relation between the reconstruction error, $\frac{1}{2}||\mathbf{A}\cdot g(\boldsymbol{y}, \mathbf{W}) - \boldsymbol{y}||_2^2$, and the LPGD and LAPGD network sizes in terms of the the number of neurons and layers in Theorem 2.1.

Theorem 2.1. *Let F^* be the optimal value of the problem (1.1), then for any $\varepsilon > 0$, there exits an LPGD (or LAPGD) network, $g(\boldsymbol{y}, \mathbf{W})$, which outputs non-negative vectors and has $O(\log_{\frac{\beta-\theta}{\beta}}\left(\frac{\varepsilon}{C}\right))$ layers (including input, hidden and output layers) and $O(n\cdot\log_{\frac{\beta-\theta}{\beta}}\left(\frac{\varepsilon}{C}\right) + m)$ neurons, such that*

$$\frac{1}{2}||\mathbf{A}\cdot g(\boldsymbol{y}, \mathbf{W}) - \boldsymbol{y}||_2^2 \leq F^* + \varepsilon \tag{2.1}$$

where β is the square of the largest singular value of $\mathbf{A}$, θ is a quantity depending only on $\mathbf{A}$ and $C = ||\boldsymbol{y}||_2^2/2$.

Proof. The problem (1.1) can be reformulated as an unconstrained optimization,

$$\underset{\boldsymbol{x}}{\mathrm{minimize}}\, F(\boldsymbol{x}) = f(\boldsymbol{x}) + g(\boldsymbol{x}) \tag{2.2}$$

(a) The block diagram of the LAPGD network.

(b) The network structure of the LAPGD network.

Fig. 2: The learned accelerated projective gradient descent (LAPGD) network.

where $f(\boldsymbol{x}) = \frac{1}{2}||\mathbf{A}\boldsymbol{x} - \boldsymbol{y}||_2^2$ and $g(\boldsymbol{x})$ is the indicator function of the nonnegative orthant. Apparently f has β-Lipschitz continuous gradient with $\beta = \|\mathbf{A}\|_2^2 = \sigma_{\max}(\mathbf{A})^2$.

Furthermore, we argue that $F(\boldsymbol{x})$ satisfies the proximal-Polyak-Lojasiewicz (PL) inequality [17]:

$$\frac{1}{2}\mathcal{D}_g(\boldsymbol{x}, \beta) \geq \theta(F(\boldsymbol{x}) - F^*)$$

for some $\theta > 0$, where

$$\begin{aligned}&\mathcal{D}_g(\boldsymbol{x}, \beta)\\&= -2\beta \min_{\boldsymbol{z}} \left[\langle \nabla f(\boldsymbol{x}), \boldsymbol{z} - \boldsymbol{x}\rangle + \frac{\beta}{2}\|\boldsymbol{z} - \boldsymbol{x}\|^2 + g(\boldsymbol{z}) - g(\boldsymbol{x})\right].\end{aligned}$$

Then [17, Theorem 5] ensures that the proximal gradient algorithm with step size $\frac{1}{\beta}$ applied to (2.2), which reduces to the PGD applied to (1.1), has a linear convergence rate

$$F(\boldsymbol{x}_k) - F^* \leq \left(1 - \frac{\theta}{\beta}\right)^k (F(\boldsymbol{x}_0) - F^*). \tag{2.3}$$

Following the line of arguments in [17, Appendix F], one obtains that $\theta > 0$ can be taken as the Hoffman constant for a system of inequalities with a system matrix $\begin{bmatrix}A^T & -A^T & -I\end{bmatrix}^T$, which can be further upper bounded using the minimal singular values of certain submatrices of $\begin{bmatrix}A^T & -A^T & -I\end{bmatrix}^T$ [18, Theorem 4.2]. The choice of θ also implies $\theta < \beta$.

Therefore, for an arbitrary $\varepsilon > 0$, if we initialize $\boldsymbol{x}_0 = 0$ and set $C = ||\boldsymbol{y}||_2^2/2$, we have

$$\frac{1}{2}||\mathbf{A}\boldsymbol{x}_k - \boldsymbol{y}||_2^2 - F^* \leq \left(1 - \frac{\theta}{\beta}\right)^k C \leq \varepsilon \tag{2.4}$$

which results in

$$k \geq \log\left(\frac{\varepsilon}{C}\right) / \log\left(1 - \frac{\theta}{\beta}\right) = \log_{\frac{\beta-\theta}{\beta}}\left(\frac{\varepsilon}{C}\right). \tag{2.5}$$

Therefore, according to the structure of the LPGD network in Fig. 1, when $\mathbf{W} = \mathbf{I} - \frac{1}{\beta}\mathbf{A}^T\mathbf{A}$ and $\mathbf{S} = \frac{1}{\beta}\mathbf{A}^T$, the LPGD network requires $\lceil k \rceil + 1$ layers including the input and output layers and $n\lceil k \rceil + m$ neurons to minimize the reconstruction error below ε if $F^* = 0$. In addition, since the LAPGD network degenerates to the LPGD network when $\mathbf{W}_2 = 0$, the result applies to both LPGD and LAPGD networks. □

Note that the proposed networks and theorem are also applicable to (1.1) when $\mathbf{A}$ does not satisfy Proposition 1. But in that case, (1.1) could have more than one solution and we can no longer guarantee that $\hat{\boldsymbol{x}}$ converges to the $\boldsymbol{x}^*$.

3. NUMERICAL EXPERIMENTS

3.1. Fast Approximation for Sparse Recovery

In the first experiment, we compare the non-negative sparse recovery performance of the proposed networks with the PGD and APGD algorithms. Specifically, we synthesize 20000 data pairs ($\boldsymbol{x}_i \in \mathbf{R}^{20}, \boldsymbol{y}_i = \mathbf{A}\boldsymbol{x}_i \in \mathbf{R}^{10}$) for training and another 2000 data pairs for testing. The goal is to recover the high-dimensional signal $\boldsymbol{x}_i$ from observation $\boldsymbol{y}_i$ with known $\mathbf{A} \in \mathbf{R}^{10\times 20}$. For each ground truth vector, $\boldsymbol{x}_i$, we randomly select its sparsity from the set $\{1, 2, 3\}$ and choose the locations of the non-zero entries uniformly at random. Then the non-zero entries of $\boldsymbol{x}_i$ are sampled from the i.i.d uniform distribution on $[0, 100]$. Similarly, each entry of $\mathbf{A}$ is sampled from the i.i.d uniform distribution on $[0, 1]$.

The neural networks are trained using the Adam algorithm [19] with 10^{-4} initial learning rate. All weights of the network are initialized with i.i.d entries uniformly on $[0, 0.001]$. In addition, $\mathbf{W}$, $\mathbf{W}_1$ and $\mathbf{W}_2$ are initialized as symmetric matrices since they are symmetric in PGD and APGD algorithms. The batch size is 200 and the whole training process takes 10000 epoches. The LPGD and LAPGD networks with different depths are trained separately and we record their average recovery error, $||\boldsymbol{x}^* - g(\boldsymbol{y}, \mathbf{W})||_2$, on the testing set in Fig. 3. The PGD and APGD algorithms start with $\boldsymbol{x}_0 = 0$ and their step sizes are $\frac{1}{\beta}$ where $\beta = \|\mathbf{A}\|_2^2$ and $\gamma = 0.9$. We can observe that the LPGD and LAPGD networks manage to learn the sparse recovery process and outperforms PGD and APGD by a large margin with the same computational cost in the test set.

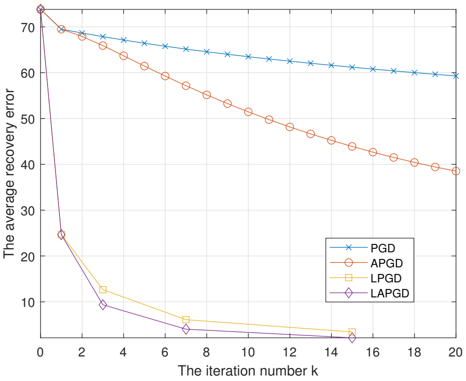


Fig. 3: The average recovery error on the testing set. For LPGD and LAPGD networks, k indicates their depths.

3.2. The Effectiveness of the Skip Connection

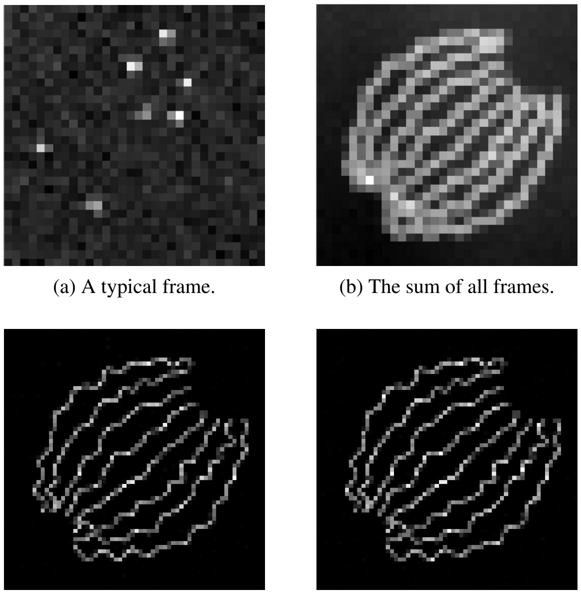
We refer to the connection that does not come from the last layer or comes from the last layer with an identity transformation as the skip connection. The second experiment illustrates the effectiveness of the skip connection in the LPGD and LAPGD networks by comparing the average recovery error of the 3-depth LAPGD, 3-depth LPGD, 3-depth LPGD without skip connections and a vanilla neural network with same number of layers and initializations. We adopt the same setup in the last experiment and the results are recorded in Table 1. Recall that, unlike the vanilla network, the weights between hidden layers in the LPGD network are the same and skip connection improves the network performance significantly. The LAPGD network with additional skip connections achieves better performance than LPGD network.

Table 1: The average recovery error on the testing set.

LAPGD	LPGD	Vanilla Network	LPGD w/o skip
8.89	12.59	12.71	14.52

3.3. Non-stationary Super Resolution

In this experiment, we examine the robustness of the LPGD and LAPGD networks when applied to the sparse recovery problem with non-stationary sensing matrix and noisy observation. This problem can no longer be solved by the PGD and APGD algorithms. Particularly, we apply the networks to the single molecule imaging, in which all sub-cellular structures are dyed with fluorophores before imaging by the microscope and in each observation, only a small portion of the fluorophores are activated for imaging. Thus each frame is composed of the activated fluorophores convolved with non-stationary point spread functions of the microscope with additive noise as shown in Fig. 4 (a). If we superpose all the frames, we obtain the low resolution image in Fig. 4 (b).



(a) A typical frame.

(b) The sum of all frames.

(c) Super-resolution result of the LPGD network.

(d) Super-resolution result of the LAPGD network.

Fig. 4: The single molecule imaging. The size of the images in (a) and (b) are 32×32 pixels with pixel size 200 nm×200 nm. (c) and (d) show the super-resolution results from LPGD and LAPGD networks whose sizes are 64×64 pixels.

The data comes from Single-Molecule Localization Microscopy grand challenge organized by ISBI [20] which contains 12000 imaging frames. With the same initialization from last experiment, we train the 7-depth LPGD and 7-depth LAPGD networks using 8000 imaging frames and implement the super-resolution on the rest 4000 frames. Thus, the training and testing datasets follow the same distribution but have different sparsity pattern and intensity for each frame. All data are pre-processed by subtracting the average intensity of the training set and the super-resolution results of the LPGD and LAPGD networks are presented in Fig. 4 (c) and (d).

4. CONCLUSION

In this paper, we propose two efficient neural networks for fast approximation of the non-negative sparse recovery. Specifically, we design the LPGD and LAPGD networks by unfolding the projected gradient descent and accelerated projective gradient descent algorithms and making their parameters trainable. Moreover, we derive an upper bound on the network sizes for a given approximation error. The experiments illustrate that the proposed networks are extremely efficient compared to the classical optimization algorithms and are capable of handling problems with non-stationary sensing matrix and noisy observation.

5. REFERENCES

[1] Youye Xie, Gongguo Tang, and William Hoff, "Chess piece recognition using oriented chamfer matching with a comparison to cnn," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 2001–2009.

[2] Weiping Pei, Youye Xie, and Gongguo Tang, "Spammer detection via combined neural network," in *International Conference on Machine Learning and Data Mining in Pattern Recognition*. Springer, 2018, pp. 350–364.

[3] Karol Gregor and Yann LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.

[4] Pablo Sprechmann, Roee Litman, Tal Ben Yakar, Alexander M Bronstein, and Guillermo Sapiro, "Supervised sparse analysis and synthesis operators," in *Advances in Neural Information Processing Systems*, 2013, pp. 908–916.

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

[6] Pablo Sprechmann, Alex M Bronstein, and Guillermo Sapiro, "Supervised non-euclidean sparse nmf via bilevel optimization with applications to speech enhancement," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*. IEEE, 2014, pp. 11–15.

[7] John R Hershey, Jonathan Le Roux, and Felix Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," *arXiv preprint arXiv:1409.2574*, 2014.

[8] Sepp Hochreiter, A Steven Younger, and Peter R Conwell, "Learning to learn using gradient descent," in *International Conference on Artificial Neural Networks*. Springer, 2001, pp. 87–94.

[9] Martijn Arts, Marius Cordts, Monika Gorin, Marc Spehr, and Rudolf Mathar, "A discontinuous neural network for non-negative sparse approximation," *arXiv preprint arXiv:1603.06353*, 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[11] Ron Zass and Amnon Shashua, "Nonnegative sparse pca," in *Advances in neural information processing systems*, 2007, pp. 1561–1568.

[12] Amnon Shashua and Tamir Hazan, "Non-negative tensor factorization with applications to statistics and computer vision," in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 792–799.

[13] Martin Slawski, Matthias Hein, et al., "Non-negative least squares for high-dimensional linear models: Consistency and sparse recovery without regularization," *Electronic Journal of Statistics*, vol. 7, pp. 3004–3056, 2013.

[14] Peter J Bickel, Ya'acov Ritov, Alexandre B Tsybakov, et al., "Simultaneous analysis of lasso and dantzig selector," *The Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[15] Yurii Nesterov et al., "Gradient methods for minimizing composite objective function," 2007.

[16] Sébastien Bubeck et al., "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[17] Hamed Karimi, Julie Nutini, and Mark Schmidt, "Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 795–811.

[18] Osman Güler, Alan J Hoffman, and Uriel G Rothblum, "Approximations to solutions to systems of linear inequalities," *SIAM Journal on Matrix Analysis and Applications*, vol. 16, no. 2, pp. 688–696, 1995.

[19] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[20] EPFL Biomedical Imaging Group, "Single-molecule localization microscopy," *http://bigwww.epfl.ch/smlm/*, 2013.