IMPROVING SPIKING SPARSE RECOVERY VIA NON-CONVEX PENALTIES

Xiang Zhang\(^1\), Lei Yu\(^1\), Gang Zheng\(^2\)

\(^1\)School of Electronic and Information, Wuhan University, China
\(^2\)Inria Lille, 40 Avenue Halley, 59650 Villeneuve d’Ascq, France

ABSTRACT

Compared with digital methods, sparse recovery based on spiking neural networks has great advantages like high computational efficiency and low power-consumption. However, current spiking algorithms cannot guarantee more accurate estimates since they are usually designed to solve the classical optimization with convex penalties, especially the \(\ell_1\)-norm. In fact, convex penalties are observed to underestimate the true solution in practice, while non-convex ones can avoid the underestimation. Inspired by this, we propose an adaptive version of spiking sparse recovery algorithm to solve the non-convex regularized optimization, and provide an analysis on its global asymptotic convergence. Through experiments, the accuracy is greatly improved under different adaptive ways.

Index Terms— Non-convex regularized optimization, spiking neural network, sparse recovery

1. INTRODUCTION

As an inherent property of signals, sparsity has been widely exploited in many areas [1], and sparse recovery (SR) is one of the key problems. Given an \(M\)-dimensional stimulus \(s \in \mathbb{R}^M\) (e.g., an \(M\)-pixel image), SR algorithms aim at recovering a \(N\)-dimensional sparse signal \(a \in \mathbb{R}^N\) from \(s\), and one of the most popular methods is to solve the regularized optimization problem

\[
\min_{a \in \mathbb{R}^N} \frac{1}{2} \|s - \Phi a\|_2^2 + \lambda C(a), \quad \lambda > 0,
\]

where \(\Phi = [\phi_1, \ldots, \phi_N] \in \mathbb{R}^{M \times N}\) with \(M \ll N\) denotes the dictionary, and \(C(a) : \mathbb{R}^N \to \mathbb{R}_+\) is a sparsity-inducing penalty. As the most sparsity-inducing function among convex ones [2], \(\ell_1\)-norm is usually chosen as the penalty \(C(a) = \|a\|_1\), and many relevant iterative SR algorithms (ISR) have been proposed, including ISTA [3], BPDN [4], LASSO [5], etc. However, these aforementioned algorithms usually need many iterations to converge and require large computing resources, which limits their applications in practice. To tackle this, the spiking neural networks (SNNs) have been widely exploited because of their computational efficiency [6–9].

Inspired by the communication scheme that biological neurons use for efficient information transformation, spiking neurons (e.g., IAF [10]) operate autonomously and only communicate rarely with other neurons via asynchronous spikes [11]. These characteristics not only reduce the power consumption, which is desirable for practical usages, but also render the hardware implementation of SNN a potentially powerful computer. Theoretically, the computability of SNN has been proved in pioneering works [12–14], and several works have revealed the efficiency of SNN in solving specific problems such as the dictionary learning [15] and SR problems [6–9]. Specifically, Shapero et al. are the first to implement the spiking SR (SSR) algorithm on an FPAA chip [16]. Though the SSR algorithm is restricted to non-negative variables to mimic biophysical variables (e.g., firing rates), their experiments still verified the extremely high efficiency and low power-consumption of SNN in solving SR problems.

However, though SSR algorithms are far superior to iterative ones in efficiency, current SSR algorithms cannot guarantee more accurate estimates. This is because most SSR systems are designed to solve the classical \(\ell_1\)-norm regularized optimization. In fact, \(\ell_1\)-norm function tends to underestimate these high-amplitude components, and often results in biased estimates, while non-convex regularized optimization can lead to more accurate estimates since non-convex penalties will not punish these components excessively [17–19]. This characteristic hints that the use of non-convex penalties might be a breakthrough to improve the accuracy of the SSR algorithm, which is also our motivation.

In this work, we will investigate the improvement of SSR from the perspective of optimization problems, and establish a bridge between the SNN and the non-convex regularized optimization. Through an adaptive mechanism, the proposed system encourages the recovery of high-amplitude components, thereby obtaining more accurate estimates. Moreover, we prove that our adaptive SSR algorithm (A-SSR) is globally asymptotically convergent, and will converge to the critical point of the non-convex regularized optimization. Based on above discussion, the relationship between SR algorithms and optimization problems is illustrated in Fig. 1 for clarity.

2. ADAPTIVE SPIKING SPARSE RECOVERY

Under the assumption that each dictionary vector satisfies the unit norm \(\|\phi_i\|_2 = 1\), SSR is proved to converge to the solution of CLASSO problem [7]:

\[
\min_{a \geq 0} \frac{1}{2} \|s - \Phi a\|_2^2 + \lambda \|a\|_1.
\]
Following this, our target optimization can be formulated as
\[
\min_{a \geq 0} E(a) = \frac{1}{2} \|s - \Phi a\|^2 + \lambda \sum_{i=1}^{N} g(|a_i|),
\]
where \(g(\cdot)\) is a non-convex penalty. Next, we will first introduce the dynamic of the A-SSR algorithm, and then discuss about how to choose a suitable penalty \(g(\cdot)\). Finally, we will prove the convergence of the A-SSR system, and point out that A-SSR will converge to the critical point of (2).

2.1. The Dynamic of A-SSR Algorithm

Based on the IAF model, each neuron in A-SSR will go through three states: inactive, active and refractory period. Taking neuron-\(i\) as an example, its potential \(\nu_i(t)\) will be charged up according to the soma current \(\mu_i(t)\) under inactive states, and neuron-\(i\) will not become active until its potential exceeds the spiking threshold \(\nu^*\), i.e. \(\nu_i(t) > \nu^*\). When neuron-\(i\) is active, its potential \(\nu_i(t)\) will be immediately reset to the resting potential \(\nu^* = 0\), and simultaneously a spike signal will be sent to affect other neurons. Afterward, neuron-\(i\) will enter the refractory period \(t_{ref} > 0\), during which \(\nu_i(t)\) remains at \(\nu^*\). In the end, neuron-\(i\) will return to the inactive state and start charging again.

Similar to [7], we define the soma current of neuron-\(i\) as
\[
\begin{align*}
\mu_i(t) &= b_i - \sum_{j \neq i} \Omega_{ij}(\alpha s + \sigma_j)(t), \\
\tau \dot{\nu}_i(t) &= b_i - \mu_i(t) - \sum_{j \neq i} \Omega_{ij}\sigma_j(t),
\end{align*}
\]
where \(b_i \equiv \phi_i^T s\) denotes the bias current; \(\Omega_{ij} \equiv \phi_i^T \phi_j\) measures the influence of spikes according to the similarity of receptive fields between neuron-\(i\) and neuron-\(j\); \(\alpha(t) \equiv e^{-t/\tau} H(t)/\tau\) is a decay function, where \(\tau\) is the time constant and \(H(t)\) denotes the Heaviside step function; \(\sigma_j(t) \equiv \sum_k \delta(t - t_{j,k})\) represents the spike train of neuron-\(j\), and \(\{t_{j,k}\}\) is its ordered spiking time sequence. Equation (3) indicates that when no spike is generated, \(\mu_i(t)\) is equal to the bias current \(b_i\), and if other neurons become active, e.g., neuron-\(j\), the spike will be weighted by \(\Omega_{ij}\) first, and then affect the soma current of neuron-\(i\) with an exponential decay.

Then, based on \(\mu_i(t)\) and a non-convex penalty \(g(\cdot)\), we can define the potential with consideration of the resetting mechanism:
\[
\nu_i(t) \equiv \int_0^t (\mu_i(s) - \Lambda_i(t))ds - \nu^* \int_0^t \sigma_i(s)ds,
\]
where \(\Lambda_i(t) = \lambda g(|a_i(t)|)\) indicates the leakage current and \(g'(|a_i(t)|)\) is the first derivative of \(g(|a_i(t)|)\) w.r.t. \(|a_i(t)|\); \(\lambda\) is the same constant in (2), and \(a_i(t)\) is the firing rate of neuron-\(i\), which is defined as \(a_i(t) \equiv \frac{1}{T} \int_0^T \sigma_i(s)ds\). Apart from \(\mu_i(t)\), equation (4) hints that the speed of potential charging is also affected by the firing rate, since \(\Lambda_i(t)\) will update adaptively according to \(a_i(t)\). Moreover, equation (4) also establishes the connection between the chosen penalty \(g(\cdot)\) and the adaptive way of A-SSR.

Based on above, our A-SSR algorithm can be fully described by (3), (4) together with the definition of the firing rate \(a_i(t)\) and the spike train \(\sigma_i(t)\). For simplicity, we set the spiking threshold \(\nu^* = 1\) from now on, and additionally set the lower bound of potential \(\nu^- = 0\), so that the potential \(\nu_i(t)\) will be reset to \(\nu^-\) once \(\nu_i(t) < \nu^-\). In the next section, we will discuss the rules of choosing a suitable penalty \(g(\cdot)\).

2.2. Rules of Choosing Penalty \(g(\cdot)\)

Based on previous works [20–22], we summarize the rules as:

1. \(g(\cdot)\) is non-negative and subanalytic on \([0, +\infty)\).
2. The first derivative of \(g(\cdot)\) is continuous and non-negative on \([0, +\infty)\), i.e. \(g'(\cdot) \geq 0\).
3. The second derivative of \(g(\cdot)\) exists, and is bounded \(g''(x) \in (-\frac{1}{\lambda}, 0), \forall x \in (0, +\infty)\).

With rule-1, the objective \(E(a)\) is also subanalytic, thus the convergence of A-SSR system can be guaranteed (see Section 2.3 for details). Rule-2 ensures that the term \(\Lambda_i(t)\) will not go negative, in case \(\nu_i(t)\) is overcharged. Last but not least, rule-3 states that the choice of penalties should be related to parameter \(\lambda\), which is beneficial to the system stability. Also, rule-3 implies that \(g'(|a_i|)\) is monotonically decreasing over \(|a_i|\) in our case. Hence, these neurons with higher firing rates are more likely to spike according to (4). In other words, the penalty \(g(\cdot)\) will not raise as fast as convex ones, thus these large elements can be better estimated. Following above, we can write the convergence result now.

2.3. Convergence of A-SSR

Theorem 1. Under a suitable penalty \(g(\cdot)\) in Section 2.2, we additionally define the average soma current \(u_i(t)\) as
\[
u_i(t) \equiv \frac{1}{t} \int_0^t \mu_i(s)ds,
\]
and if other neurons become active, e.g., neuron-\(j\), the spike will be weighted by \(\Omega_{ij}\) first, and then affect the soma current of neuron-\(i\) with an exponential decay.
then the average soma current $u(t)$ and the firing rate $a(t)$ of A-SSR are globally asymptotically convergent, and $a(t)$ will converge to the critical point of (2) as time goes to infinity.

To prove it, we start with an auxiliary system designed as follows (the dependence on time is omitted for readability).

$$
\tau \dot{u}^a_i = b_i - u^a_i - \sum_{j \neq i} \Omega_{ij} a^a_j,
$$

$$
a^a_i = \max(u^a_i - \Lambda^a_i, 0) = \begin{cases}
  f(u^a_i), & \text{if } u^a_i > \Lambda^a_i, \\
  0, & \text{else},
\end{cases}
$$

where $f(u^a_i(t)) = u^a_i(t) - \Lambda^a_i(t) = \Lambda^a_i(t) = \lambda g'(|a^a_i(t)|)$ is defined on $[0, +\infty)$, and the variables are restricted to non-negative (i.e. $u^a_i(t) \geq 0$ and $a^a_i(t) \geq 0$). Based on this, the proof can be divided into four lemmas. Lemma 1 and Lemma 2 together prove the convergence property of the auxiliary system. Then, with the boundedness of $\mu(t)$ and $u(t)$ in Lemma 3, we can finally prove that A-SSR will converge to the auxiliary system as time goes to infinity, and the firing rate $a(t)$ will converge to the critical point of (2).

**Lemma 1.** With the objective $E(a^a_n)$ in (2), the auxiliary system satisfies that $\tau \dot{u}^a_i \in -\partial E(a^a_n)$.

**Proof.** By introducing the sign function $\text{sgn}(\cdot)$, we have

$$
a^a_i(t) - a^a_i(t) \in \Lambda^a_i(t) \text{sgn}(a^a_i(t))
$$

based on the following discussion: 1) When $u^a_i(t) > \Lambda^a_i(t)$, $\text{sgn}(u^a_i(t)) = 1$ holds since $\Lambda^a_i(t) \geq 0$ from the rule-2 in Section 2.2. Hence, we can derive the relationship $u^a_i(t) - \Lambda^a_i(t) = \Lambda^a_i(t) \text{sgn}(a^a_i(t))$. 2) When $u^a_i(t) \in [0, \Lambda^a_i(t)]$, we have $a^a_i(t) = 0$, therefore $u^a_i(t) - a^a_i(t) = u^a_i(t) \in \Lambda^a_i(t) \text{sgn}(a^a_i(t))$.

Then, according to (7), the state $u^a_i(t)$ in (6) can be rewritten as $\tau \dot{u}^a_i = b - \Omega u^a_i \text{sgn}(a^a_n)$, which is also equivalent to $\tau \dot{u}^a_i = \Phi^T s$ since $\Phi^T \Phi a^a_n = \lambda g'(\text{sgn}(a^a_n)) \times \text{sgn}(a^a_n)$ since $b = \Phi^T s$ and $\Omega = \Phi^T \Phi$. On the other hand, the sub-differentiation of $E(a^a_n)$ w.r.t. $a^a_n$ yields $\partial E(a^a_n) = -\Phi^T s + \Phi^T \Phi a^a_n + \lambda g'(\text{sgn}(a^a_n)) \times \text{sgn}(a^a_n)$, then it is straightforward that $\tau \dot{u}^a_i \in -\partial E(a^a_n)$ holds.

Next, we are going to prove the convergence of the auxiliary system. In fact, the global asymptotic convergence of such systems has been proved in [20] using ojasiewicz Inequality. Here, we focus on proving the conditions required by Theorem 1 and Theorem 2 in [20].

**Lemma 2.** The state $u^a_n(t)$ and the output $a^a_n(t)$ of the auxiliary system (2) are globally asymptotically convergent.

**Proof.** First of all, (7) corresponds to the condition (3) in [20], since $C(a^a_n(t)) = g(|a^a_n(t)|)$ in our case. Additionally, we have the first derivative of $f(u^a_i(t))$ when $u^a_i(t) > \Lambda^a_i(t)$:

$$
f'(u^a_i(t)) = 1 - \frac{d\Lambda_i}{du^a_i(t)} \frac{d\Lambda_i}{du^a_i(t) \text{sgn}(a^a_i(t))} = 1 - \lambda g'(\text{sgn}(a^a_i(t))) \times \text{sgn}(a^a_i(t)) f'(u^a_i(t)).
$$

With the rule-3 in Section 2.2, one can obtain that $f'(u^a_i(t)) = 1/(1 + \lambda g'(|a^a_i(t)|)) \times \text{sgn}(a^a_i(t)) > 0$, which corresponds to the condition (5b) in [20]. Also, $f'(\cdot)$ satisfies the condition (7) in [20], since one can always find an upper bound of $f'(\cdot)$ on bounded intervals. As for the condition (6) in [20], it is used to guarantee that $E(a)$ is subanalytic, which is also met by the rule-1 in Section 2.2. For the remaining conditions, there might be some small differences due to the one-side case we are concerned about, but they will not affect the final result of global asymptotic convergence.

**Lemma 3.** There exist an upper bound $B_+$ and a lower bound $B_-$ such that $\mu_i(t), u_i(t) \in [B_-, B_+], \forall i, t \geq 0$.

**Proof.** Based on the assumption that dictionary atoms are normalized to unit norm, we define the auxiliary parameters: $\Omega_{\text{max}} \defeq \text{max}_{j \neq i} |\phi_j^T \phi_i|$ and $\mu_{\text{max}} \defeq \text{max}_i |b_i|$. Then, with the refractory period in spiking neurons, the duration between spikes cannot be arbitrarily small whenever these two spike times exist, i.e. $t_{k+1} - t_k \geq t_* \epsilon, \forall k \geq 0$. Therefore, we have the inequality $\beta \defeq \frac{1}{N} \sum_{c=0}^{\infty} e^{-(ct_* \epsilon) / \tau} \geq (\alpha \ast \sigma) (t)$, and the following relationship can be obtained based on (3):

$$
\mu_i(t) \leq b_{\text{max}} + (N-1) \Omega_{\text{max}} \beta,
$$

where $N$ denotes the total number of neurons in A-SSR. Similarly, the relationship $\mu_i(t) \geq -b_{\text{max}} - (N-1) \Omega_{\text{max}} \beta$ can be derived. Since $u_i$ is the average of $\mu_i$, there exist bounds $B_+ = b_{\text{max}} + (N-1) \Omega_{\text{max}} \beta$ and $B_- = -b_{\text{max}} - (N-1) \Omega_{\text{max}} \beta$ such that $\mu_i(t), u_i(t) \in [B_-, B_+], \forall i, t \geq 0$.

**Lemma 4.** As $t \to \infty$, the average soma current $u(t)$ and the firing rate $a(t)$ of A-SSR is globally asymptotically convergent, and $a(t)$ will converge to the critical point of (2).

**Proof.** With the derivative derived from (3):

$$
\dot{u}_i(t) = \frac{\mu_i(t) - u_i(t)}{t},
$$

we have the equation below by applying $t^{-1} \int_0^t ds$ to (3):

$$
\tau \ddot{u}_i(t) = b_i - u_i(t) - \sum_{j \neq i} \Omega_{ij} a_j(t) - \tau \frac{u_i(t) - \mu(0)}{t}.
$$

On the other hand, based on $\nu^a = 1$ and the definition of $a_i(t)$, we can derive the following relationship from (3):

$$
a_i(t) = \frac{1}{t} \int_0^t (\mu(s) - \Lambda_i(s)) ds - \frac{\nu_i(t)}{t}
$$

Under $a_i(t) \geq 0$, then we can transfer to

$$
a_i(t) = \max(u_i(t) - \Lambda_i(t) - \nu_i(t), 0).
$$
A-SSR

Probability of Recovery

Sparsity

Measurement

A-SSR

Convergence

(d) Convergence comparison. (e) is the convergence comparison. (f) and (g) are the performance under different sparsity, noise, and measurement.

We investigated the improvement of SSR based on the optimization problem. By establishing the connection between the non-convex penalty and the adaptive mechanism, our A-SSR system can avoid the underestimation of high-amplitude components, thereby obtaining more accurate estimates.

3. NUMERICAL EXPERIMENT

Finally, to verify the performance of A-SSR, we present numerical experiments on Nengo, a neural engineering simulation platform. At first, we compare the working processes between SSR and A-SSR with exponential penalty. Since A-SSR converges to the auxiliary system, the A-SSR system satisfies $\tau \dot{u}(t) \in -\partial E(a(t))$ as $t \to \infty$. Combining (12), it is clear that $0 \in \partial E(a^\star)$, which means that as $t \to \infty$, $a(t)$ will converge to $a^\star$, the critical point of (2).

With above results, we complete the proof of Theorem 1.

With $\nu_i(t) \in [\nu^-, \nu^+]$ and Lemma 3, it is obvious that (9) and (11) will converge to (6) as $t \to \infty$, i.e. the A-SSR will converge to the auxiliary system. Therefore, with Lemma 2, the $u(t)$ and $a(t)$ of A-SSR are also globally asymptotically convergent when time goes to infinity.

Assume that $u^\star$ is the stable point of A-SSR and $a^\star$ is the corresponding output. With (8) and Lemma 3, we have

$$\dot{u}^\star = \lim_{t \to \infty} \dot{u}(t) = \lim_{t \to \infty} \frac{\mu(t) - u(t)}{t} = 0.$$ (12)

Since A-SSR converges to the auxiliary system, the A-SSR system satisfies $\tau \dot{u}(t) \in -\partial E(a(t))$ as $t \to \infty$. Combining (12), it is clear that $0 \in \partial E(a^\star)$, which means that as $t \to \infty$, $a(t)$ will converge to $a^\star$, the critical point of (2).

With above results, we complete the proof of Theorem 1.

4. CONCLUSION

We investigated the improvement of SSR based on the optimization problem. By establishing the connection between the non-convex penalty and the adaptive mechanism, our A-SSR system can avoid the underestimation of high-amplitude components, thereby obtaining more accurate estimates.
References

[1] Julien Mairal, “Sparse Modeling for Image and Vision Processing,” Foundations and Trends in Computer Graphics and Vision, vol. 8, no. 2-3, pp. 85–283, 2014.

[2] Alfred M. Bruckstein, David L. Donoho, and Michael Elad, “From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images,” SIAM Review, vol. 51, no. 1, pp. 34–81, Feb. 2009.

[3] I. Daubechies, M. Defrise, and C. De Mol, “An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint,” Communications on Pure and Applied Mathematics, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[4] Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders, “Atomic Decomposition by Basis Pursuit,” Siam Review, vol. 43, no. 1, pp. 129–159, 2001.

[5] Robert Tibshirani, “Regression Shrinkage and Selection Via the Lasso,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 58, no. 1, pp. 267–288, Jan. 1996.

[6] Joel Zylberberg, Jason Timothy Murphy, and Michael Robert DeWeese, “A Sparse Coding Model with Synaptically Local Plasticity and Spiking Neurons Can Account for the Diverse Shapes of V1 Simple Cell Receptive Fields,” PLoS Computational Biology, vol. 7, no. 10, pp. e1002250, Oct. 2011.

[7] Ping Tak Peter Tang, Tsung-Han Lin, and Mike Davies, “Sparse Coding by Spiking Neural Networks: Convergence Theory and Computational Results,” arXiv: Learning, May 2017.

[8] Ping Tak Peter Tang, “Convergence of LCA Flows to (C)LASSO Solutions,” arXiv: Numerical Analysis, Mar. 2016.

[9] Chi-Ning Chou, Kai-Min Chung, and Chi-Jen Lu, “On the Algorithmic Power of Spiking Neural Networks,” arXiv: Neural and Evolutionary Computing, Nov. 2018.

[10] Wolfram Gerstner and Werner M. Kistler, Spiking Neuron Models: Single Neurons, Populations, Plasticity, Cambridge University Press, 2002.

[11] Samanwoy Ghosh-Dastidar and Hojjat Adeli, “Spiking Neural Networks,” International Journal of Neural Systems, vol. 19, no. 4, pp. 295–308, 2009.

[12] Wolfgang Maass, “Lower bounds for the computational power of networks of spiking neurons,” Neural Computation, vol. 8, no. 1, pp. 1–40, 1996.

[13] Wolfgang Maass, “Networks of Spiking Neurons: The Third Generation of Neural Network Models,” Neural Networks, vol. 10, no. 9, pp. 1659–1671, Dec. 1997.

[14] Wolfgang Maass and Christopher Bishop, Pulsed Neural Networks, MIT Press, Jan. 1998.

[15] Tsung-Han Lin and Ping Tak Peter Tang, “Sparse Dictionary Learning by Dynamic Neural Networks,” ICLR 2019: 7th International Conference on Learning Representations, p. 20, 2019.

[16] Samuel Shapero, Christopher Rozell, and Paul Hasler, “Configurable hardware integrate and fire neurons for sparse approximation,” Neural Networks, vol. 45, pp. 134–143, Sept. 2013.

[17] Emmanuel J. Cands, Michael B. Wakin, and Stephen P. Boyd, “Enhancing Sparsity by Reweighted ℓ1 Minimization,” Journal of Fourier Analysis and Applications, vol. 14, no. 5-6, pp. 877–905, Dec. 2008.

[18] G. Gasso, A. Rakotomamonjy, and S. Canu, “Recovering Sparse Signals With a Certain Family of Nonconvex Penalties and DC Programming,” IEEE Transactions on Signal Processing, vol. 57, no. 12, pp. 4686–4698, Dec. 2009.

[19] Md. Mashud Hyder and Kaushik Mahata, “An Improved Smoothed ℓ0 Approximation Algorithm for Sparse Representation,” IEEE Transactions on Signal Processing, vol. 58, no. 4, pp. 2194–2205, Apr. 2010.

[20] Aurele Balavoine, Christopher J. Rozell, and Justin Romberger, “Convergence of a Neural Network for Sparse Approximation using the Nonsmooth ojasiewicz Inequality,” in The 2013 International Joint Conference on Neural Networks (IJCNN), Dallas, TX, USA, Aug. 2013, pp. 1–8, IEEE.

[21] Laming Chen and Yuantao Gu, “The Convergence Guarantees of a Non-Convex Approach for Sparse Recovery,” IEEE Transactions on Signal Processing, vol. 62, no. 15, pp. 3754–3767, Aug. 2014.

[22] Sophie M. Fosson, “A Biconvex Analysis for Lasso ℓ1 Reweighting,” IEEE Signal Processing Letters, vol. 25, no. 12, pp. 1795–1799, 2018.

[23] Trevor Bekolay, James Bergstra, Eric Hunsberger, Travis DeWolf, Terrence C. Stewart, Daniel Rasmussen, Xuan Choo, Aaron Russell Voelker, and Chris Eliasmith, “Nengo: a Python tool for building large-scale functional brain models,” Frontiers in Neuroinformatics, vol. 7, pp. 48–48, 2014.