Convergence and Rate Analysis of Neural Networks for Sparse Approximation

Aurèle Balavoine, *Student Member, IEEE,* Justin Romberg, *Senior member, IEEE,*
and Christopher J. Rozell, *Member, IEEE,*

***Abstract*—We present an analysis of the Locally Competitive Algorithm (LCA), a Hopfield-style neural network that efficiently solves sparse approximation problems (e.g., approximating a vector from a dictionary using just a few non-zero coefficients). This class of problems plays a significant role in both theories of neural coding and applications in signal processing. However, the LCA lacks analysis of its convergence properties and previous results on neural networks for nonsmooth optimization do not apply to the specifics of the LCA architecture. We show that the LCA has desirable convergence properties, such as stability and global convergence to the optimum of the objective function when it is unique. Under some mild conditions, the support of the solution is also proven to be reached in finite time. Furthermore, some restrictions on the problem specifics allow us to characterize the convergence rate of the system by showing that the LCA converges exponentially fast with an analytically bounded convergence rate. We support our analysis with several illustrative simulations.**

***Index Terms*—Locally Competitive Algorithm, sparse approximation, global stability, exponential convergence, nonsmooth objective, Lyapunov function.**

I. Introduction

SPARSE approximation has generated substantial interest in a wide range of research communities over the last two decades, including signal processing, machine learning, statistics and computational neuroscience (e.g., see [2], [3] and references therein). Specifically, sparse approximation involves solving an optimization problem to represent a signal using just a few atoms from some specified (possibly overcomplete) dictionary. In addition to describing compelling models of neural coding for sensory information [4], [5], this approach has led to state of the art results in many types of inverse problems. One example of a regime that has leveraged this type of signal model is the theory of Compressed Sensing (CS) [6], [7]. In this domain, highly undersampled signals are recovered by solving a sparse approximation problem, thereby shifting the burden of data acquisition from the front-end sensor to a computationally intensive back end.

Because of the increasing interest in the sparse approximation paradigm, significant effort has been made to design efficient algorithms for solving (or approximately solving) least squares optimization problems that are regularized with a sparsity-inducing penalty (e.g., [8]–[10]). However, all of these algorithms are developed to run on digital computers in discrete time, which are both implausible for neural systems and suffer from several drawbacks as engineering approaches. In particular, the computational time required by these algorithms presents a barrier to real-time signal processing applications with high dimensional signals at significant datarates. Specifically, digital algorithms tend to have storage requirements and convergence times that scale unfavorably with the dimensionality of the signals being approximated. Additionally, the power consumption of digital solutions can be prohibitive for many applications. Given these considerations, a fast and low-power method for solving sparse approximation problems in a parallel architecture would be valuable both for engineering systems and viable models of neural coding.

Analog neural networks have long been proposed for solving optimization problems [11], with an early example being Hopfield's pioneering results [12] using networks to solve linear optimization problems. Analog neural networks offer several potential advantages over comparable digital algorithms, including their ability to be implemented in analog architectures that are highly parallel, fast and power efficient. Recent advances in VLSI reconfigurable analog chips [13] make the design of such systems more feasible and affordable than has often been true in the past. A recent neural network architecture, called a Locally Competitive Algorithm (LCA) [14], has been proposed to solve the types of nonsmooth optimization problems that come up in sparse approximation. These Hopfield-style networks appear to efficiently solve a whole family of sparse approximation problems by incorporating ideal thresholding nonlinearities in the network dynamics. The results in [14] provide encouraging evidence that biological systems and engineering applications can use neural networks to solve these important sparse approximation problems.

However, despite encouraging evidence of its performance, the LCA lacks strong convergence guarantees and estimates of the convergence rate. Furthermore, the specifics of the LCA architecture violate many of the assumptions used in previous analysis, making the extensive literature on neural network convergence inapplicable for this system. For example, in previous work the activation function is often assumed to be linear [15], piecewise linear [16], or nonlinear but increasing and bounded [17]–[20]. In contrast, the LCA activation function is nonlinear and unbounded. Other analyses assume an interconnection matrix which is positive definite [16], [21], [22] or non-singular [23], [24], while the LCA interconnection matrix may have negative eigenvalues as well as a non-

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0250 USA.
e-mail: {aurele.balavoine,jrom,crozell}@gatech.edu,

This work was partially supported by NSF grant CCF-0905346 and NIH grant R01-EY019965. Preliminary versions of portions of this work were presented in [1].

trivial nullspace (due to the approximation dictionary being overcomplete). Other analyses consider a nonsmooth objective function with constraints, but only shows convergence when the constraints are nonzero and convex [25], [26]. Finally, other relevant analyses assume the objective function is convex [27] or piecewise linear [28], while the work in [29] focuses on controllability of the network path.

The main contributions of this paper are to present a formal analysis of the LCA network architecture and its convergence properties, despite 1) an activation function that is nonsmooth and not necessarily bounded or increasing, and 2) a potentially singular interconnection matrix. Section III contains our first main result, which states that the fixed points of the LCA network correspond to critical points of the objective function. In the special case where the objective is convex, this set coincides with the global minima of the objective. In addition, we show that the network is globally stable and that the outputs are quasi-convergent, in the sense that they get infinitely close to a set of fixed points. Finally, in the case where the objective function has isolated minima, we show that the LCA converges to the solution of the sparse approximation problem for any initial point (i.e., a much stronger condition than just the non-increasing property of the energy function that was shown in previous work [14]). This section also shows that the LCA is well-behaved in that it converges in a finite number of switches (i.e., nodes crossing above or below threshold). Section IV expands on these general results to show that, under additional mild conditions on the problem specifics, the LCA actually converges *exponentially fast* to the solution. Furthermore, we give an analytic expression for this convergence rate that depends on the properties of the detailed approximation problem. Finally, Section V presents simulation results showing the correspondence of our analytic results with empirical observations of the network behavior.

II. Background

Before presenting our main results, in this section we briefly give a precise statement of the sparse approximation problems of interest, a description of the LCA architecture, and some preliminary observations on the LCA network dynamics that will be useful in the subsequent analysis.

A. *Sparse Approximation*

As mentioned above, sparse approximation is an optimization program that seeks to find the approximation coefficients of a signal on a prescribed dictionary, using as few non-zero elements as possible. To fix notation, we denote the input signal by $y \in \mathbb{R}^M$, the unit-norm dictionary elements by the columns of the $M \times N$ matrix $\Phi = [\Phi_1, \ldots, \Phi_N]$, and the coefficients by $a \in \mathbb{R}^N$. Generally, $M \ll N$ (i.e., the approximation dictionary is overcomplete), and the problem of recovering a from y is underdetermined. While similar in spirit to the well-known winner-take-all problem [30], sparse approximation problems are generally formulated as the solution to an optimization program because this approach can often yield strong performance guarantees in specific applications (e.g., recovery in a CS problem). In the most generic form, the objective function is the sum of a quadratic data fidelity term (i.e., mean squared error) and a regularization term that uses a sparsity-inducing cost function $C(\cdot)$:

$$\min_a \; V(a(t)) = \frac{1}{2}\left\|y - \Phi a\right\|_2^2 + \lambda \sum_{n=1}^{N} C(a_n), \qquad (1)$$

where the parameter λ is a tradeoff between the two terms in the objective function. The "ideal" sparse approximation problem has a cost function that simply counts the number of non-zero elements, resulting in a non-convex objective function that has many local minima [31].

One of the most widely used programs from this family is known as Basis Pursuit De-Noising (BPDN) [32], which is given by the objective function:

$$\min_a \; \frac{1}{2}\left\|y - \Phi a\right\|_2^2 + \lambda \left\|a\right\|_1. \qquad (2)$$

In BPDN, the ℓ_1-norm ($\|a\|_1 = \sum_n |a_n|$) is used as a convex surrogate for the idealized counting norm. This program has gained in popularity as researchers have shown that, in many cases of interest, substituting the ℓ_1 norm yields the same solution as using an idealized (and generally intractable) counting norm [33]. However, BPDN illustrates the canonical challenge of sparse approximation problems. Despite being convex, the BPDN objective function contains a nonsmooth nonlinearity that makes it considerably more difficult than a classic least-squares problem.

In the context of computational neuroscience, sparse approximation has been proposed as a neural coding scheme for sensory information. In one interpretation, programs such as BPDN can be viewed as Bayesian inference in a linear generative model with Gaussian noise and a prior with high kurtosis to encourage sparsity (e.g., the Laplacian prior in the case of BPDN) [5]. Given the prevalence of probabilistic inference as a successful description of human perception [34] and the theoretical benefits of sparse representations [3], it has long been conjectured that sensory systems may encode stimuli via sparse approximation. In fact, in classic results, sparse approximation applied to the statistics of natural stimuli in an unsupervised learning experiment has been shown to be sufficient to qualitatively and quantitatively explain the receptive field properties of simple cells in the primary visual cortex [4], [35] as well as the auditory nerve fibers [36]. Only recently have there been proposals of efficient neural networks that could efficiently solve the necessary optimization problems to implement this type of encoding [14], [35], [37], [38].

B. *LCA structure*

Our primary interest will be the LCA [14], an analog, continuous-time dynamical system that is a type of Hopfield-style network. In particular, each node in the LCA network is characterized by the evolution of a set of internal state variables, $u_n(t)$ for $n = 1, \ldots, N$, and uses a nonlinear activation function $T_\lambda(\cdot)$ to produce output variables $a_n(t)$ for $n = 1, \ldots, N$. The activation function is typically a nonsmooth nonlinear function (such as a thresholding function) to induce sparsity in the outputs. The dynamics of the internal

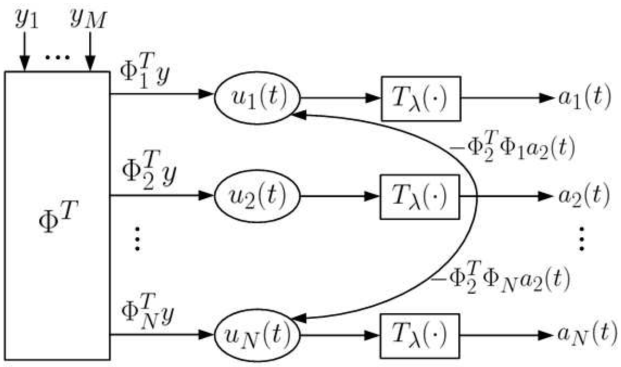

Fig. 1. Block diagram of the Locally Competitive Algorithm (LCA) [14], a Hopfield-style network architecture for solving sparse approximation problems.

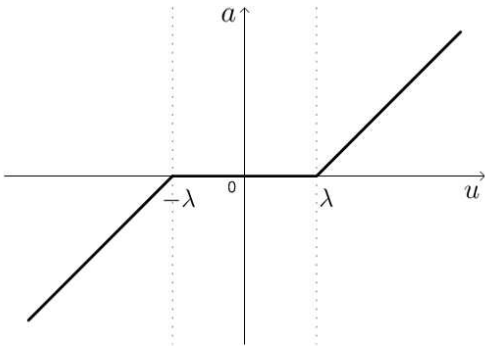

Fig. 2. The soft-thresholding activation function. When this is used for the thresholding function $T_\lambda(\cdot)$, the LCA solves the popular BPDN optimization problem used in many sparse approximation applications.

state variables are governed by a set of coupled, nonlinear ordinary differential equations (ODEs):

$$\begin{aligned} \tau \dot{u}(t) &= -u(t) - \left(\Phi^T \Phi - I\right) a(t) + \Phi^T y \\ a(t) &= T_\lambda(u(t)) \end{aligned} . \tag{3}$$

In this network, each node is associated with a single dictionary element Φ_n, $n = 1, \ldots, N$, and the node outputs will be shown to be the solution to the optimization problem of interest. The architecture of a typical LCA is shown in Fig. 1.

The inputs to the LCA network are feedforward connections computing the vector of driving synaptic inputs $\Phi^T y$, reflecting how well the signal y matches each dictionary element. The network also has recurrent inhibitory or excitatory connections between the nodes, modulated by weights corresponding to the interconnection matrix $G = \Phi^T \Phi - I$ (i.e., a modified Grammian matrix for the dictionary). The more overlap there is between a pair of nodes (characterized by the inner product between their dictionary elements), the stronger the potential inhibition between those nodes. While the modulating weight is symmetric between any pair of nodes, the total inhibition is not because it is also modulated by the activity of each individual node. Moreover, the matrix G potentially has both negative and positive eigenvalues, as well as a non-trivial nullspace. This inhibition structure ensures that nodes that carry the same information will not become active at the same time, thus meeting the goals of sparse approximation. The time constant τ is dependent on the physical properties of the solver implementing the ODEs. For our purposes, we will often assume $\tau = 1$ without loss of generality.

It was shown in [14] that the objective function in (1) is non-increasing along the LCA trajectory with the following relationship ($\forall a_n \in \mathbb{R}$ such that $a_n \neq 0$) between the cost penalty term $C(\cdot)$ and the activation function $T_\lambda(\cdot)$:

$$\lambda \frac{dC(a_n)}{da_n} = u_n - a_n = u_n - T_\lambda(u_n). \tag{4}$$

In the case of BPDN, the cost penalty is simply $C(x) = |x|$ and the activation function obtained from (4) is the soft-thresholding function (Fig. 2) defined by:

$$a_n(t) = T_\lambda(u_n(t)) = \begin{cases} 0, & |u_n(t)| \leq \lambda \\ u_n(t) - \lambda \ \mathrm{sign}(u_n(t)), & |u_n(t)| > \lambda \end{cases} .$$

This function often arises in connection to algorithms for minimizing the absolute value of the coefficients (e.g., see [39]). Generalizing from the soft-threshold, we focus on "thresholding" activation functions $T_\lambda(\cdot)$ of the form:

$$a_n(t) = T_\lambda(u_n(t)) = \begin{cases} 0, & |u_n(t)| \leq \lambda \\ f(u_n(t)), & |u_n(t)| > \lambda \end{cases} , \tag{5}$$

where the function $f(\cdot)$ is a real-valued function defined and continuous on the domain $\mathcal{D} = (-\infty, -\lambda] \cup [\lambda, +\infty)$, differentiable on the interior of $\mathcal{D}$, and satisfying the conditions:

$$f(-u_n) = -f(u_n), \qquad \forall u_n \in \mathcal{D} \text{ and } f(\lambda) = 0 \tag{6a}$$

$$f'(u_n) > 0, \qquad \forall u_n \in \mathcal{D} \tag{6b}$$

$$f(u_n) \leq u_n, \qquad \forall u_n \in \mathcal{D} \text{ s.t. } u_n \geq 0. \tag{6c}$$

Remark. Eq. (6a) ensures that $T_\lambda(\cdot)$ is continuous on all $\mathbb{R}$ and consequently, that $u(t)$ and $a(t)$ are continuous with respect to time. This ensures that the cost function $C(\cdot)$ is differentiable everywhere except at the origin. Eq. (6b) makes $f(\cdot)$ a bijection on $\mathcal{D}$. Finally, Eq. (6a) and (6c) ensure, by a simple computation, that $C(\cdot)$ is positive and non-decreasing with the absolute value of the coefficient. Notice that the ℓ_1-minimization objective function satisfies the three criteria in (6), in addition to being convex (which is not required, but guarantees that the system will find a global minima). In Fig. 3, we plot a more generic stylized activation function that satisfies the conditions (6).

As shown in Fig. 3, the activation function (5) is composed of two operating regions. The first region corresponds to the case where the internal state u_n is below the threshold λ, in which case the output a_n is zero. We call the nodes in this region *inactive nodes*. The second region corresponds to the case where the internal state u_n is above threshold, in which case (6b) guarantees that the output a_n is strictly increasing with u_n. We call the nodes that are above threshold *active nodes*, and we denote by Γ the *active set* (i.e., the set of indices corresponding to those nodes, $\Gamma = \{n \in [1, N] : |u_n(t)| > \lambda\}$). On the contrary, we denote the set of indices corresponding to nodes that are below threshold by Γ^c and call them the *inactive set*. While these two sets do change with time as the network evolves, for the sake of readability

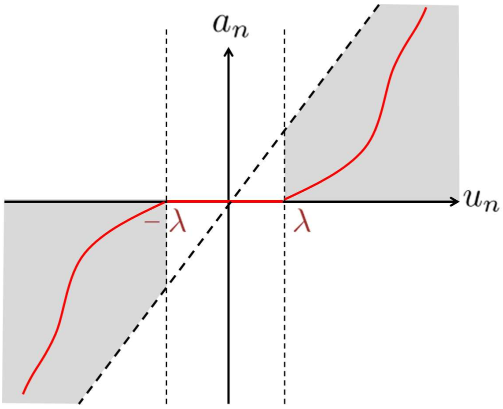

Fig. 3. Generic activation function satisfying conditions (6). The area in gray represents where the activation function must lie in to satisfy conditions (6b) and (6c).

we omit the dependence on time in the notation.

Due to the nonsmooth nature of the objective (1), we require the generalized notion of a subgradient (see [40] for more details). Note that $V(a)$ is differentiable everywhere except at points a that contain zeros, due to the non-differentiability of $C(\cdot)$ at the origin noted above for the cases of interest in sparse approximation. The subgradient $\partial V(\cdot)$ extends the classic notion of a gradient $\nabla V(\cdot)$ at those points of discontinuity. The properties in (6) allow us to use the following simple definition:

$$\partial V(a) = co\left\{\lim_{i\to\infty} \nabla V(a_i) : a_i \to a, a_i \notin \mathcal{S}, a_i \notin \Omega_V\right\},$$

where co is the convex hull, Ω_V is the set of point where V fails to be differentiable and $\mathcal{S}$ is any set of Lebesgue measure 0 in $\mathbb{R}^N$. In other words, it is the smallest convex set containing the gradients of the function as we approach the discontinuity from any direction. Note that when a is not a point of discontinuity, this simply reduces to $\nabla V(a)$.

C. LCA Node Dynamics

Because nodes can cross threshold and go from the inactive set to the active set (and vice versa), the LCA can be thought of as a switched system [41] where the ODEs change at each switching time (i.e., when a node crosses above or below threshold). In between two switching times, the active set Γ and inactive set Γ^c are fixed. Therefore, the dynamics on the two sets can be considered separately to facilitate analysis. In the following, $\Phi_{\mathcal{T}}$ is the matrix composed of the columns of Φ indexed by the set $\mathcal{T}$. Similarly, $u_{\mathcal{T}}$ and $a_{\mathcal{T}}$ refer to the elements in the original vectors indexed by $\mathcal{T}$. We also denote by $\{t_k\}_{\{k\in\mathbb{N}\}}$ the sequence of times for which the system switches from the set of active nodes Γ_{k-1} to Γ_k. The notation $T_\lambda(u(t))$ refers to the vector $\left[T_\lambda(u_1(t)),\ldots,T_\lambda(u_N(t))\right]^T$, and we denote by $F'(t)$ the associated Jacobian matrix with respect to the internal state variables u. Because the activation function has the form in (5), the matrix $F'(t)$ is diagonal with diagonal elements equal to zero for indices in the inactive set Γ^c and equal to $f'(u_n)$ for indices n in the active set Γ.

Using the chain rule and (5), we can calculate the derivative with respect to time of the outputs on the active set Γ:

$$\dot{a}_\Gamma(t) = F'_\Gamma(t)\dot{u}_\Gamma(t) = \text{diag}\Big(f'\left(u_n(t)\right)\Big)_{\{n\in\Gamma\}}\dot{u}_\Gamma(t). \quad (7)$$

As a consequence, the ODE (3) can be rewritten for nodes in the active set as follows:

$$\dot{a}_\Gamma(t) = F'_\Gamma(t)\left[-u_\Gamma(t) + a_\Gamma(t) - \Phi_\Gamma^T\Phi_\Gamma a_\Gamma(t) + \Phi_\Gamma^T y\right]. \quad (8)$$

The active nodes follow this ODE until the next switch occurs, changing the sets of active and inactive nodes. Similarly, we can rewrite the set of differential equations acting on the inactive nodes. Since the output of the activation function for nodes in Γ^c is zero, the ODE (3) on the inactive set becomes:

$$\dot{u}_{\Gamma^c}(t) = -u_{\Gamma^c}(t) - \Phi_{\Gamma^c}^T\Phi_\Gamma a_\Gamma(t) + \Phi_{\Gamma^c}^T y. \quad (9)$$

Separating the system dynamics into two separate characterizations for the active nodes Γ and inactive nodes Γ^c yields two sets of differential equations that are partially decoupled, and this will be crucial for our subsequent analysis. This partial decoupling is possible because only the active nodes in the system produce inhibitory signals, coupling their dynamics to the dynamics of each node in the inactive set via the interconnection matrix. However, because inactive nodes do not inhibit active nodes, the dynamics on the active set are independent of the inactive set.

III. Convergence Results

In [14], the authors show that the LCA network trajectory is non-increasing on the energy surface corresponding to the desired objective function. While this is necessary behavior for a network that solves an optimization problem, it is not sufficient to actually show that the network converges to a fixed point (or set of points), let alone reaches a minimum of that desired objective function. Both of these are important guarantees to make before relying on this system as a viable model of neural processing or as an implementation in engineering applications.

In Section III-A, we analyze the general convergence properties of the LCA. In particular, in this section the interconnection matrix $\Phi^T\Phi - I$ may have many negative or zero eigenvalues, which complicates the analysis. The main results of this first section show that the fixed points of the LCA correspond to critical points of the objective (1) and the outputs of the network converge from any initial state to the set of fixed points. In the case of isolated critical points of (1), we show that the LCA is globally convergent in the sense that starting from any initial state, the system converges to a fixed point. Elaborating on these results, Section III-B shows that the LCA (even under very general assumptions) is very well-behaved, converging in a finite number of switches (i.e., nodes crossing above or below threshold) and therefore recovering the solution support (i.e., the set of non-zero elements) in finite time. Due to the difficulties described above, our approach relies on classic results from nonsmooth analysis, such as subgradients and a generalized chain rule (see Appendix A).

A. Global Asymptotic Convergence

To begin, we recall some useful definitions of stability and concepts from Lyapunov theory (see [42] for more background). There exist several notions of stability associated with dynamical systems that describe the evolution of the nodes or their outputs both locally and globally. For a neural network described by a differential equation of the form

$$\dot{x} = G(x),\ x \in \mathbb{R}^N, \tag{10}$$

with outputs $z = T_\lambda(x)$ defined as in (5), we say that a constant vector $x^* \in \mathbb{R}^N$ is a **fixed point** of (10) if and only if

$$G(x^*) = 0. \tag{11}$$

On the other hand, the outputs of a system reach a **critical point** a^c of the objective function $V(\cdot)$ when they satisfy the inclusion:

$$0 \in \partial V(a^c). \tag{12}$$

Note that if $V(\cdot)$ is convex, then the critical points correspond exactly to the global minima of V. We say that the system (10) is (Lyapunov) **stable** at x^* if for each $\epsilon > 0$, there exists $R > 0$ such that for all x_0 with $\|x_0 - x^*\| < R$, and all the solutions $x(\cdot)$ with initial state x_0,

$$\|x(t) - x^*\| < \epsilon,\ \forall t > 0. \tag{13}$$

This is clearly a local notion of stability, guaranteeing that once the trajectory is close to a fixed point, it remains nearby. But this type of stability is insufficient to guarantee global convergence, which means that trajectories approach a fixed point as time goes to infinity. The following notion of stability is slightly stronger, guaranteeing that trajectories approach at least a set of fixed points regardless of the initial state.

Definition 1. *We say that the outputs of* (10) *are* ***globally quasi-convergent*** *if there exists a set $\mathcal{E} = \{z^* \in \mathbb{R}^N : \dot{z}^* = 0\}$ such that for all $x_0 \in \mathbb{R}^N$, the outputs $z(\cdot) = T_\lambda(x(\cdot))$ with initial state x_0 satisfy $\lim_{t \to +\infty} z(t) \in \mathcal{E}$.*

Finally, the strongest and most desirable form of convergence, which guarantees that the nodes converge to a single fixed point, is stated in the next definition.

Definition 2. *We say that (10) is* ***globally convergent****, or equivalently* ***globally asymptotically stable****, if there exists a fixed point x^* at which the system is stable, and if for all initial states $x_0 \in \mathbb{R}^N$ the solutions $x(\cdot)$ satisfy: $\lim_{t \to +\infty} x(t) = x^*$.*

Typically, global convergence is established through the use of a so-called Lyapunov function V. The notation $\dot{V}$ refers to the derivative with respect to time, i.e. dV/dt.

Definition 3. *A function $V : \mathbb{R}^N \mapsto \mathbb{R}$ is a* ***weak Lyapunov function*** *on $\mathbb{R}^N$ if:*
(i) $V(x) > 0$, $\forall x \neq 0$;
(ii) V is continuous on $\mathbb{R}^N$;
(iii) $\dot{V}(x) \leq 0$, $\forall x \in \mathbb{R}^N$; and
(iv) V is radially unbounded:

$$\lim_{\|x\| \to +\infty} V(x) = +\infty.$$

Similarly, a function is called a ***strict Lyapunov function*** if it meets the above conditions, with the exception of having a strict inequality in condition (iii) (i.e., $\dot{V}(x) < 0, \forall x \in \mathbb{R}^N$).

Remark. When it is possible to find a weak Lyapunov function for a given dynamical system, the first theorem of Lyapunov [42] guarantees that any fixed point of the system is stable in the sense of (13). However, to show global convergence of a system, the second theorem of Lyapunov requires a strict Lyapunov function.

One can check that the objective function in (1) is a weak Lyapunov function for the system (3), thus guaranteeing that the LCA is Lyapunov stable (which improves on the stability result obtained in [14]). However, it is not a strict Lyapunov function. Indeed, (1) only depends on the active nodes, meaning that it could stop decreasing while subthreshold nodes are still evolving. Continued evolution of the inactive nodes could cause a node in Γ^c to become active, thereby causing the objective function to start decreasing again. As a consequence, condition (iii) in Definition 3 is not satisfied with strict inequality, and the standard approach of using the objective function as a Lyapunov function is not sufficient to show global convergence of the LCA. To show global convergence of the system to a fixed point, it is necessary to account for the dynamics on both active and inactive nodes.

Our main convergence theorem guarantees the global quasi-convergence of the LCA towards a set of critical points of the objective function. In the case of isolated critical points, this implies global convergence.

Theorem 1. *The LCA system defined in (3), with an activation function of the form (5) satisfying conditions* (6)*:*

1) *has fixed points that are critical points of the objective function defined in (1);*
2) *has globally quasi-convergent outputs; and*
3) *is globally convergent, provided that the critical points of* (1) *are isolated.*

Note that part 2 of the above theorem only relates to the output variables, while part 3 is a much stronger condition on the entire dynamical system (including subthreshold states). In the highly relevant case of convex objective functions with unique minima (e.g., BPDN), part 3 of the above theorem applies directly and we have the strongest possible notion of convergence. In fact, for most dictionaries Φ (e.g., random Gaussian), the minimum of the ℓ_1-minimization objective function (2) is unique [43]. Recent results on subgradient dynamical systems [26], [44] lead us to believe that part 3 could still hold in the case where the fixed points of the system are not isolated and there exists a subspace of solutions to (1), but this conjecture is beyond the scope of this work. Section V-A illustrates the convergence behavior of the LCA in simulation. The proof of this theorem is in Appendix A, and relies on generalized notions of a subgradient due to the nonsmooth nature of the objective.

B. Convergence in a Finite Number of Switches

In the theorem below, we strengthen the previous result to establish that under some mild conditions, the support of the final solution is reached with a finite number of switches. To prove this result, it is sufficient to assume that no node in the solution lies exactly on the threshold λ. This assumption precludes unwanted infinite oscillation behavior on the boundaries, known as Zeno behavior. In other words, we assume that there exists a margin ($r > 0$) above and below the threshold which contains no node in u^*. One would expect this condition to hold with near certainty for any signal that was not pathologically constructed.

Theorem 2. *If the system* (3) *converges to a fixed point u^* such that there exists $r > 0$:*

$$\begin{array}{ll} \forall n \in \Gamma_*, & |u_n^*| \geq \lambda + r \\ \forall n \in \Gamma_*^c, & |u_n^*| \leq \lambda - r \end{array},$$

then the system converges after a finite number of switches.

This implies that the neural network recovers the support of the solution a^* in finite time. Section V-A explores the number of switches during the convergence of the LCA in simulation.

Proof: Let Γ_* be the set of active nodes in u^*. By contradiction, assume that the sequence of switching times $\{t_k\}_{k\in\mathbb{N}}$ is infinite. Since the LCA converges to u^*, we have:

$$u(t_k) \underset{k\to+\infty}{\longrightarrow} u^*.$$

As a consequence, for $r > 0$, there exists $K \in \mathbb{N}$ such that $\forall k \geq K$, $\|u(t_k) - u^*\|_2 < r$. To begin, we show that for all $k \geq K$, the state variables $u(t_k)$ are in the subsystem Γ_*. For all $k \geq K$, we have two cases:

- Nodes that are above threshold in u^* are above threshold in $u(t_k)$. Indeed, $\forall n \in \Gamma_*$, we have:

$$r > |u_n(t_k) - u_n^*| \geq |u_n^*| - |u_n(t_k)| \geq \lambda + r - |u_n(t_k)|$$
$$\Rightarrow \ |u_n(t_k)| > \lambda.$$

  Moreover, nodes are active with the correct sign, otherwise, we would have:

$$r > |u_n(t_k) - u_n^*| = |u_n(t_k)| + |u_n^*| > \lambda + \lambda + r$$
$$\Rightarrow \ 0 > \lambda,$$

  which is a contradiction.
- Nodes that are below threshold in u^* are below threshold in $u(t_k)$. Indeed, $\forall n \in \Gamma_*^c$, we have:

$$|u_n(t_k)| - \lambda \leq |u_n(t_k)| - |u_n^*| - r \leq |u_n(t_k) - u_n^*| - r < 0$$
$$\Rightarrow \ |u_n(t_k)| < \lambda.$$

As a consequence, for all $k \geq K$, $\quad \Gamma_k = \Gamma_*$. However, Γ_k and Γ_{k+1} must be different to define the switching time t_{k+1} and we reach a contradiction. This proves that after a finite number of switches K, there cannot be any switching out of subsystem Γ_*. ∎

IV. Exponential Convergence Rate

With Theorem 1 showing that the LCA is globally convergent, the most pressing issue remaining is to determine the convergence rate of the network. Such a bound will be especially important for implementations, which must guarantee solution times. In this section, we show that under some additional conditions on the problem's specifics (given in (16)), the LCA network converges exponentially fast to a unique fixed point u^*. We also give an analytic characterization of the convergence speed.[1]

To begin this section, we recall the definition of exponential convergence:

Definition 4. *The dynamical system in* (10) *is* ***exponentially convergent*** *to the solution x^* if there exists a constant $c > 0$ such that for any initial point $x(0)$, there exists a constant $\kappa_0 > 0$ (which may depend on $x(0)$) for which the trajectory $x(t)$ of the system satisfies:*

$$\|x(t) - x^*\| \leq \kappa_0 e^{-ct}, \qquad \forall t \geq 0.$$

The constant c is referred to as ***convergence speed*** *of the system.*

In order to state the main theorem of this section, we define the two following quantities. The first constant is denoted by α and provides a bound on the derivative of $f(\cdot)$ in (5):

$$\forall t \geq 0, \forall n = 1, \ldots, N \qquad |f'(u_n(t))| \leq \alpha. \qquad (14)$$

Note that the constant α is always well defined since the trajectories $u_n(t)$ are bounded and the function $f(\cdot)$ is continuous. The second constant, denoted by δ, is the smallest positive constant such that for any active set Γ visited by the algorithm and any vector x in $\mathbb{R}^N$ with active set $\widetilde{\Gamma} = \Gamma \cup \Gamma_*$, where Γ_* is the active set of the solution to (1), we have:

$$(1-\delta)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1+\delta)\|x\|_2^2. \qquad (15)$$

The constant δ depends on the singular values of the matrix $\Phi_{\widetilde{\Gamma}}$ and on the sequence of active sets visited by the system. It may not be well defined for any matrix Φ or any input y. However, in many interesting cases in CS, the constant δ is close to 0 and the dictionary elements are almost orthogonal for any small enough active set [45]. The following theorem shows that this constant directly relates to the convergence speed of the neural network.

Theorem 3. *Under conditions* (6) *on the activation function in* (5)*, and provided that the constants α and δ, defined in* (14) *and* (15) *respectively, satisfy:*

$$\delta < \min\left(1, \frac{1}{(2\alpha-1)^2}\right), \qquad (16)$$

the LCA system defined in (3) *is globally exponentially convergent to a unique equilibrium, with convergence speed $(1-\delta)/2\tau$.*

If condition (16) is satisfied, the expression given for the

[1] We reintroduce the time constant τ in this discussion of the LCA, since it appears in the expression for the convergence speed.

convergence speed is positive and thus meaningful. It depends on the eigenvalues of the matrix $\Phi_\Gamma^T\Phi_\Gamma$, which vary with the active set Γ. A careful analysis of the sequence of active sets visited by the network is required to obtain a good estimate of δ. Since such a study is application dependent, we do not address this question here. Note that in the very interesting case of the soft-threshold function, $\alpha = 1$ and so, condition (16) reduces to $\delta < 1$. The time constant τ of the physical solver implementing the LCA neural network appears in the expression of the speed of convergence. Lowering this time constant means the system will converge faster. Analog systems can have smaller time constants than their digital counterparts that scale better with the problem size. Section V-B explores the convergence rate bounds for the LCA in simulation.

To establish the expression of the convergence speed, we use the following Lure-Postnikov-type Lyapunov function:

$$E(t) = \frac{1}{2}\left\|\widetilde{u}(t)\right\|_2^2 + \sum_{n=1}^{N}\int_0^{\widetilde{u}_n(t)} g_n(s)ds, \tag{17}$$

where we again redefine the output and state variables in terms of the distance from any arbitrary fixed point u^*

$$\begin{aligned}&\widetilde{u}_n(t) = u_n(t) - u_n^*,\\ &\widetilde{a}_n(t) = a_n(t) - a_n^* = T_\lambda(\widetilde{u}_n(t) + u_n^*) - T_\lambda(u_n^*),\end{aligned} \tag{18}$$

and the function $g_n(s) = T_\lambda(s + u_n^*) - T_\lambda(u_n^*)$, which captures how much a perturbation of the internal state affects the output. Because the second term in (17) is non-negative, we have trivially that this energy function bounds the mean-squared error of the current state values to their steady-state values, $\frac{1}{2}\left\|\widetilde{u}(t)\right\|_2^2 \leq E(t)$. The theorem will prove that u^* is unique. The properties presented in the following Lemma (see Appendix B for proof) are useful to prove the main result of this section.

Lemma 1. *Assume that the activation function* (5) *satisfies the conditions* (6). *Then, the set of variables* $\widetilde{\mathrm{u}}$ *and* $\widetilde{\mathrm{a}}$ *defined in* (18) *satisfy the properties:*

(i) $\mathrm{sign}(\widetilde{a}_n) = \mathrm{sign}(\widetilde{u}_n)$.

(ii) $\left|\widetilde{a}_n\right| \leq \alpha\left|\widetilde{u}_n\right|$.

(iii) $\widetilde{a}_{\mathcal{T}}^T\widetilde{a}_{\mathcal{T}} \leq \alpha\widetilde{u}_{\mathcal{T}}^T\widetilde{a}_{\mathcal{T}} \leq \alpha^2\widetilde{u}_{\mathcal{T}}^T\widetilde{u}_{\mathcal{T}}$ *for any* $\mathcal{T}$ *(in particular for* $\mathcal{T} = \widetilde{\Gamma}$*).*

(iv) $\displaystyle\sum_{n=1}^{N}\int_0^{\widetilde{u}_n(t)} g_n(s)ds \leq \widetilde{u}^T\widetilde{a}$.

Armed with this lemma, we can now give the proof of the main theorem.

Proof of Theorem 3: We begin by noting that since u^* and a^* are the internal state variables and the outputs at a fixed point of (3), we can rewrite the dynamics in terms of the new variables:

$$\tau\dot{\widetilde{u}}(t) = -\widetilde{u}(t) - \left(\Phi^T\Phi - I\right)\widetilde{a}(t). \tag{19}$$

Using the chain rule, we find the expression for the derivative with respect to time of the energy function (17):

$$\tau\dot{E}(t) = \tau\frac{dE(\widetilde{u}(t))}{d\widetilde{u}(t)}^T\frac{d\widetilde{u}(t)}{dt} = \Big(\widetilde{u}(t) + \widetilde{a}(t)\Big)^T\tau\dot{\widetilde{u}}(t)t.$$

Using (19), this becomes:

$$\tau\dot{E}(t) = -\Big(\widetilde{u}(t) + \widetilde{a}(t)\Big)^T\Big(\widetilde{u}(t) + \left(\Phi^T\Phi - I\right)\widetilde{a}(t)\Big).$$

Using the properties in Lemma 1 leads to the desired expression (and removing the term t to increase readability):

$$\begin{aligned}\tau\dot{E}(t) &= -\Big(\widetilde{u} + \widetilde{a}\Big)^T\Big(\widetilde{u} + \left(\Phi^T\Phi - I\right)\widetilde{a}\Big)\\ &= \left(\widetilde{u} + \widetilde{a}\right)^T\left(-\widetilde{u} + \widetilde{a} - \Phi^T\Phi\widetilde{a}\right)\\ &= -\widetilde{u}^T\widetilde{u} + \widetilde{a}^T\widetilde{a} - \widetilde{a}^T\Phi^T\Phi\widetilde{a} - \widetilde{u}^T\Phi^T\Phi\widetilde{a}.\end{aligned}$$

Adding and subtracting the term $\frac{1}{2}\widetilde{u}\Phi^T\Phi\widetilde{u}$, results in:

$$\begin{aligned}\tau\dot{E}(t) = {}&-\widetilde{u}^T\widetilde{u} + \widetilde{a}^T\widetilde{a} - \widetilde{a}^T\Phi^T\Phi\widetilde{a} + \frac{1}{2}\left(\widetilde{u} - \widetilde{a}\right)^T\Phi^T\Phi\left(\widetilde{u} - \widetilde{a}\right)\\ &- \frac{1}{2}\widetilde{u}\Phi^T\Phi\widetilde{u} - \frac{1}{2}\widetilde{a}^T\Phi^T\Phi\widetilde{a}\\ = {}&-\widetilde{u}^T\widetilde{u} + \widetilde{a}^T\widetilde{a} - \frac{3}{2}\left\|\Phi\widetilde{a}\right\|_2^2 + \frac{1}{2}\left\|\Phi\left(\widetilde{u} - \widetilde{a}\right)\right\|_2^2 - \frac{1}{2}\left\|\Phi\widetilde{u}\right\|_2^2.\end{aligned}$$

We separate the vectors into their components onto the active set $\widetilde{\Gamma} = \Gamma \cup \Gamma^*$ of $\widetilde{a}$ and its complement $\widetilde{\Gamma}^c$:

$$\begin{aligned}&\frac{1}{2}\left\|\Phi\left(\widetilde{u} - \widetilde{a}\right)\right\|_2^2 - \frac{1}{2}\left\|\Phi\widetilde{u}\right\|_2^2 = \frac{1}{2}\left\|\Phi_{\widetilde{\Gamma}}\left(\widetilde{u}_{\widetilde{\Gamma}} - \widetilde{a}_{\widetilde{\Gamma}}\right)\right\|_2^2 + \frac{1}{2}\left\|\Phi_{\widetilde{\Gamma}^c}\widetilde{u}_{\widetilde{\Gamma}^c}\right\|_2^2\\ &\quad - \frac{1}{2}\left\|\Phi\widetilde{u}\right\|_2^2 = \frac{1}{2}\left\|\Phi_{\widetilde{\Gamma}}\left(\widetilde{u}_{\widetilde{\Gamma}} - \widetilde{a}\right)\right\|_2^2 - \frac{1}{2}\left\|\Phi_{\widetilde{\Gamma}}\widetilde{u}_{\widetilde{\Gamma}}\right\|_2^2.\end{aligned}$$

We can now use the definition of δ in (15):

$$\begin{aligned}\tau\dot{E}(t) \leq {}&-\widetilde{u}^T\widetilde{u} + \widetilde{a}^T\widetilde{a} - \frac{3}{2}(1-\delta)\left\|\widetilde{a}\right\|_2^2\\ &+ \frac{1}{2}(1+\delta)\left\|\widetilde{u}_{\widetilde{\Gamma}} - \widetilde{a}\right\|_2^2 - \frac{1}{2}(1-\delta)\left\|\widetilde{u}_{\widetilde{\Gamma}}\right\|_2^2\\ = {}&-\frac{1}{2}(1-\delta)\widetilde{u}^T\widetilde{u} - (1-\delta)\widetilde{u}^T\widetilde{a} - \frac{1}{2}(1-\delta)\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{u}_{\widetilde{\Gamma}}\\ &- 2\delta\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{a} + 2\delta\widetilde{a}^T\widetilde{a} - \frac{1}{2}(1+\delta)\left\|\widetilde{u}_{\widetilde{\Gamma}^c}\right\|_2^2\\ \leq {}&-\frac{1}{2}(1-\delta)\widetilde{u}^T\widetilde{u} - (1-\delta)\widetilde{u}^T\widetilde{a} - \frac{1}{2}(1-\delta)\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{u}_{\widetilde{\Gamma}}\\ &- 2\delta\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{a} + 2\delta\widetilde{a}^T\widetilde{a}.\end{aligned}$$

Now, using property (iii) of Lemma 1, the expression becomes:

$$\begin{aligned}\tau\dot{E}(t) \leq {}&-\frac{1}{2}(1-\delta)\left\|\widetilde{u}\right\|_2^2 - (1-\delta)\widetilde{u}^T\widetilde{a}\\ &+ \left[-\frac{1}{2\alpha}(1-\delta) - 2\delta + 2\delta\alpha\right]\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{a}\\ = {}&-\frac{1}{2}(1-\delta)\left\|\widetilde{u}\right\|_2^2 - (1-\delta)\widetilde{u}^T\widetilde{a}\\ &- \frac{1}{2\alpha}\left[1 - \delta\left(2\alpha - 1\right)^2\right]\widetilde{u}_{\widetilde{\Gamma}}^T\widetilde{a}.\end{aligned}$$

From (16), we know that $1 - \delta\left(2\alpha-1\right)^2 \geq 0$, and so:

$$\tau\dot{E}(t) \leq -\frac{1}{2}(1-\delta)\left\|\widetilde{u}\right\|_2^2 - (1-\delta)\widetilde{u}^T\widetilde{a}.$$

Finally, using property (iv) of Lemma 1 and the fact that $\delta \leq 1$, we arrive at:

$$\begin{aligned}\tau\dot{E}(t) &\leq -\frac{1}{2}(1-\delta)\left\|\widetilde{u}\right\|_2^2 - (1-\delta)\sum_{n=1}^{N}\int_0^{\widetilde{u}_n(t)} g_n(s)ds\\ &= -\left(1-\delta\right)E(t).\end{aligned}$$

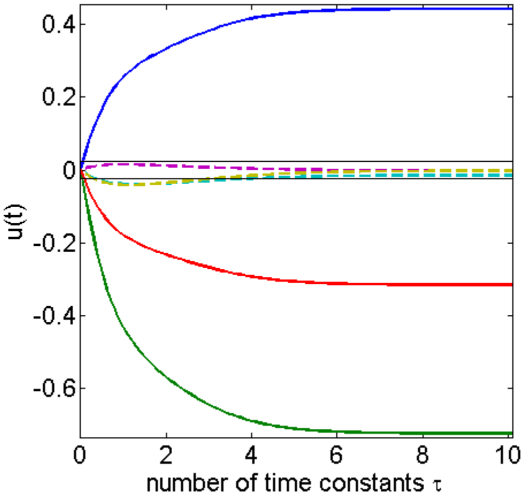

Fig. 4. Plot of the evolution with respect to time of several LCA nodes $u_k(t)$. The plain lines correspond to nodes that are active in the final solution and the dashed lines correspond to nodes that are inactive in the final solution.

Knowing already that $E(t)$ is an upper bound on the quantity of interest, we can integrate both sides of the above inequality (i.e., apply Gronwall's inequality) to get a bound on $E(t)$,

$$\frac{1}{2}\left\|u(t)-u^*\right\|_2^2 \leq E(t) \leq e^{-(1-\delta)t/\tau}E(0), \tag{20}$$

proving that the LCA is exponentially convergent to a unique fixed point u^*. Taking the square root, this expression also shows that the convergence speed in the exponential is $(1-\delta)/(2\tau)$, as stated in the theorem. ■

V. Simulations

In this section, we illustrate the previous theoretical results in simulation. Each plot is based on the following canonical sparse approximation problem. We generate a "true" sparse vector $a_0 \in \mathbb{R}^N$, with $N = 512$ and $s = 5$ non-zero entries. We select the locations of the nonzeros uniformly at random and draw their amplitudes from a normal gaussian distribution (then normalizing to have unit norm). We choose a dictionary Φ as a union of the canonical basis and a sinusoidal basis having dimensions $M \times N$ with $M = 256$. The vector of measurements is $y = \Phi a_0 + \eta \in \mathbb{R}^M$, where η is Gaussian random noise with standard deviation 0.0062. We use an LCA with a soft-threshold activation function, with a threshold set to $\lambda = 0.025$ and $u(0) = 0$. We simulate the LCA dynamical equations (3) through a discrete approximation in Matlab with a step size of 0.001 and a solver time constant chosen to be equal to $\tau = 0.01$.

A. Convergence Results

From Theorem 1, the LCA should converge and recover the solution to the sparse approximation problem (1), which has a unique minimizer. Since the signal a_0 to recover is sparse, the outputs of the neural network are expected to converge to a solution close to the initial signal a_0. Fig. 4 shows the evolution of a few nodes $u_n(t)$ selected at random. We see that both active and inactive nodes converge relatively quickly. Fig. 5 shows the fixed point reached by the LCA system and compares it to the initial signal a_0 and to the solution obtained using a standard digital solver for the same optimization program [8]. The solution reached by the network possesses exactly 5 non-zero entries that correspond to the non-zero entries in a_0. The recovered amplitudes are very close to the initial amplitudes (it cannot be exact due to the added measurement noise) and to the ones produced by the reference digital solver. However, the LCA produces a sparse vector while the digital solver returns many small but non-zero entries that would have to be removed by postprocessing.

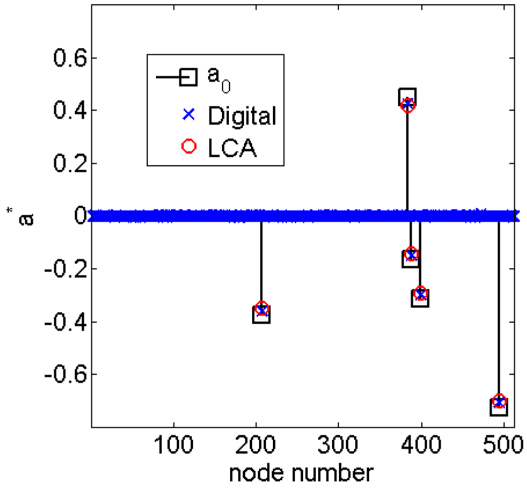

Fig. 5. Output a^* of the LCA after convergence. Only non-zero elements are plotted. The fixed point reached by the system is very close to the initial sparse vector used to create the measurement vector (it cannot be exact due to noise). The solution is also very close to a standard digital solver [8] run using the same inputs. Note that the LCA produces many coefficients that are exactly zero (therefore not plotted).

To illustrate the global convergence behavior, we also ran the LCA for 30 randomly generated initial points. We selected two nodes from the final active set and plotted the trajectories in the space defined by those two nodes. Fig. 6 clearly shows that the solution is attractive for any of those initial points.

To illustrate the number of switches used by the system (see Theorem 2), we generate 1000 sparse vectors a_0 and measurements y and simulate the LCA dynamics. Fig. 7 shows a histogram of the number of switches needed for the system to converge. The figure illustrates that the number of switches before convergence of the neural network is finite and of the order of the dictionary size. This illustrates that this solver takes an efficient path towards the solution.

B. Convergence Rate Results

To illustrate the convergence rate result in Theorem 3, it is necessary to find an expression for the convergence speed $(1-\delta)/\tau$ that appears in the exponential term in (20). This term bounds the error squared:

$$\frac{1}{2}\left\|u(t)-u^*\right\|_2^2,$$

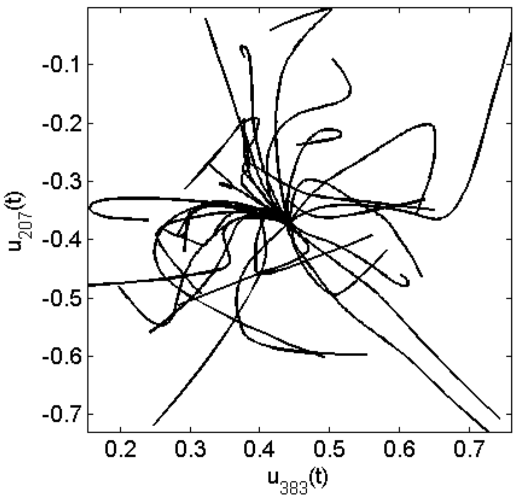

Fig. 6. Trajectories of $u(t)$ in the plane defined by two nodes chosen randomly from the active set. Trajectories are shown for 30 random initial states.

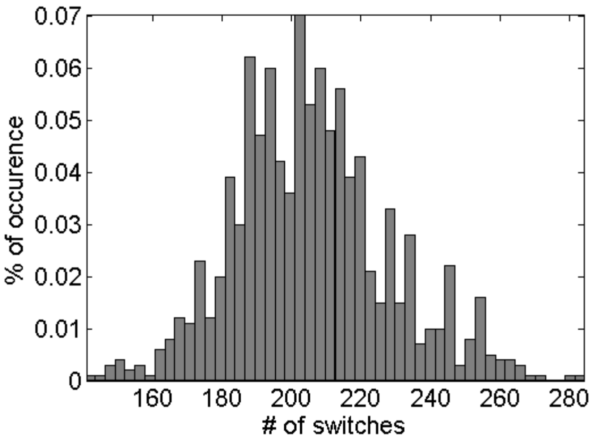

Fig. 7. Histogram (in percentage) of the number of switches the LCA requires before convergence over 1000 trials.

which is normalized to have initial value at $t = 0$ of 1 and plotted using a log-scale on Fig. 8. Note that the constant δ defined in (15) depends on the sequence of active sets Γ_k visited by the system. However, it is very difficult to predict for a given input signal what sequence of active sets the algorithm is going to visit. To estimate this upper bound, we compute the constant δ using the matrix Φ_{Γ_*} composed of the dictionary elements that are active in the final solution. The corresponding upper bound on the decay $e^{-(1-\delta^*)t/\tau}$ is plotted in Fig. 8. During the transient phase, the number of active nodes is actually larger and thus, we expect $1 - \delta$ to be smaller than this estimate and the nodes to converge slower at some point during the transient phase. As a consequence, we keep track of the largest support visited by the network and compute the corresponding δ. This second upper bound $e^{-(1-\delta_{\max})t/\tau}$ on the convergence rate is plotted on the same figure. As expected, the theoretical decay computed with the maximum support visited is an upper bound for the convergence speed. However, it can be seen that this estimate is very pessimistic and that the bound computed with the final active set is a better estimate. This simulation illustrates that the theoretical exponential convergence appears to capture the essential system behavior.

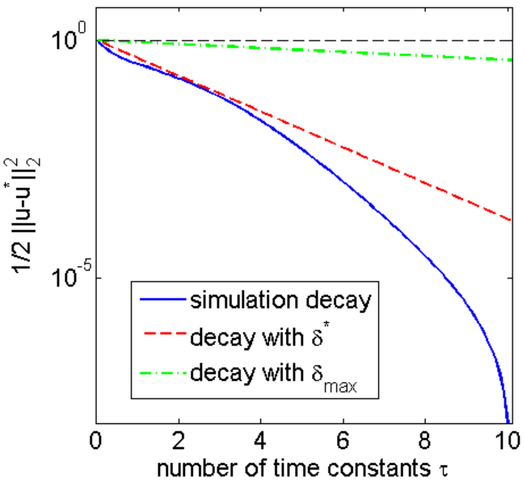

Fig. 8. Convergence behavior of $\frac{1}{2}\|u(t) - u^*\|_2^2$. The dashed line shows the theoretical decay in (20) with δ^* computed by using the final solution support. The dash-dot line shows the theoretical decay with $\delta_{\max}$ computed on the largest support visited. While both estimates are showing theoretically correct behavior, the estimate of the rate based on δ^* is more empirically accurate than the conservative estimate based on $\delta_{\max}$.

VI. Conclusion

This paper presents a mathematical analysis of the convergence properties and convergence rate of the LCA, a neural network designed specifically for the challenging sparse approximation problem. Despite a nonsmooth activation function and possibly singular interconnection matrix that prevent the application of existing analysis approaches, we have shown that the system is globally convergent to the optimal solution. In addition, under some mild assumptions on the solution, we have shown that the trajectories follow a reasonable path and reach the final active set in finite time. Finally, under slightly stronger assumptions on the problem specifics (applicable at least in CS recovery problems), we established that the LCA is exponentially convergent and with a convergence rate that depends on problem-specific parameters.

This collection of results and analysis leads us to conclude that performance guarantees can be made for the LCA system that make it plausible for implementation in engineering applications and as a model of biological information processing. Indeed, providing such guarantees makes it easier to justify the expense associated with developing analog VLSI implementations, which could eventually result in significant improvements to the speed and power consumption necessary for real-time signal processing applications. Our future work will concentrate on finding reasonable estimates of the theoretical convergence speed (especially in well-studied special cases such as CS recovery), and further characterizations of the LCA system dynamics that may open up new applications of this system for time-varying input signals.

Appendix A
Proof of Theorem 1

Building on the earlier definition of a subgradient to give a notion that non-differentiable functions can still be well-behaved, a function $g : X \to \mathbb{R}$ (where X is a Banach space) is said to be *regular* [40, Def. 2.3.4] at x in X if

(i) For all $v \in X$, the usual one-sided directional derivative
$$g'(x; v) = \lim_{t \downarrow 0} \frac{g(x + tv) - g(x)}{t} \qquad \text{exists.}$$

(ii) For all $v \in X$, $g'(x; v) = g^\circ(x; v)$, where $g^\circ(x; v)$ is the generalized directional derivative
$$g^\circ(x; v) = \limsup_{\substack{y \to x \\ t \downarrow 0}} \frac{g(y + tv) - g(y)}{t}.$$

It can easily been seen that, since the function $C(\cdot)$ is defined from $\mathbb{R}$ to $\mathbb{R}$, is differentiable on $\mathbb{R}\backslash\{0\}$ and the function $T_\lambda(\cdot)$ is continuous on all $\mathbb{R}$, then $C(\cdot)$ admits left and right derivatives and is clearly regular on $\mathbb{R}$. This implies that $V(\cdot)$ is regular on $\mathbb{R}^N$, and by [40, Prop. 2.3.3] we have that:

$$\partial V(a(t)) = -\Phi^T y + \Phi^T \Phi a(t) + \lambda \partial C(a(t)), \tag{21}$$

where $\partial C(a(t)) = \left[\partial C(a_1(t)), \ldots, \partial C(a_N(t))\right]^T$. We also recall the following result [40, Th. 2.3.9 (iii)], which is a generalization of the chain rule for regular functions.

Lemma 2. *Suppose that $V(a) : \mathbb{R}^N \to \mathbb{R}$ is regular in $\mathbb{R}^N$ and that $a(t) : [0, +\infty) \to \mathbb{R}^N$ is strictly differentiable on $[0, +\infty)$. Then, $V(a(t))$ is regular on $\mathbb{R}^N$, and we have*

$$\frac{d}{dt} V(a(t)) = \zeta^T \dot{a}(t) \qquad \forall \zeta \in \partial V(a(t)). \tag{22}$$

Note that from this theorem, since $V(a(t))$ is regular, we can choose any element in $\partial V(a(t))$ to compute the time derivative of $V(\cdot)$ along the trajectories of the neural network. Armed with these tools, we proceed with the proof of Theorem 1.

Proof of Theorem 1: Beginning with part 1 of the theorem, we first show that any fixed point of system (3) is a critical point of the objective in (1). From (11), any fixed point u^* of (3) satisfies the relationship $\dot{u}(t) = 0$. Let Γ_* be the active set at the fixed point u^*. Eq. (8) and (9) yield:

$$-\, u^*_{\Gamma_*} + f(u^*_{\Gamma_*}) - \Phi^T_{\Gamma} \Phi_{\Gamma} a^*_{\Gamma_*} + \Phi^T_{\Gamma_*} y = 0 \tag{23}$$
$$-\, u^*_{\Gamma^c_*} - \Phi^T_{\Gamma^c_*} \Phi_{\Gamma_*} a^*_{\Gamma_*} + \Phi^T_{\Gamma^c_*} y = 0. \tag{24}$$

According to (12), and using (21), a point a^* is a critical point of $V(\cdot)$ if and only if:

$$\Phi^T y - \Phi^T \Phi a^* \in \lambda \partial C(a^*). \tag{25}$$

For the nodes in the active set $n \in \Gamma_*$, $C(a_n)$ admits a usual gradient, as defined in (4) and thus, (25) yields

$$\Phi^T_{\Gamma_*} y - \Phi^T_{\Gamma_*} \Phi_{\Gamma_*} a^*_{\Gamma_*} = u^*_{\Gamma_*} - f(u^*_{\Gamma_*}).$$

This is the same as condition (23). For nodes in the inactive set Γ^c_*, we need to determine the subgradient of $C(\cdot)$ at zero. To do so, note that using (4) and the continuity of the function $f(\cdot)$ at λ (condition (6a)), we have:

$$\lim_{a \downarrow 0} \lambda \nabla C(a) = \lim_{u \downarrow \lambda} u - f(u) = \lambda.$$

As a consequence, we find that:

$$\lim_{a \downarrow 0} \nabla C(a) = 1 \qquad \text{and} \qquad \lim_{a \uparrow 0} \nabla C(a) = -1.$$

From the definition of the subgradient, these limits show that: $\partial C(0) = [-1, 1]$. Using this, the condition in (25) restricted to the set of inactive nodes is equivalent to:

$$\Phi^T_{\Gamma^c_*} y - \Phi^T_{\Gamma^c_*} \Phi_{\Gamma_*} a^*_{\Gamma_*} \in [-\lambda, \lambda].$$

We immediately see that this condition is the same as (24), since by definition of the inactive nodes $u^*_{\Gamma^c_*} \in [-\lambda, \lambda]$. This shows that the fixed points a^* coincide with the critical points of the objective function (1).

Moving on to establishing the convergence result in part 2 of the theorem, we first note that from our analysis above, the same reasoning leads to the conclusion that:

$$-\dot{u}(t) = u(t) - f(u(t)) + \Phi^T \Phi\, a(t) - \Phi^T y \in \partial V(a(t)). \tag{26}$$

Using the chain rule (22) with $\zeta = -\dot{u}(t) \in \partial V(a(t))$ from (26) and using the expression in (7), we get the relationship:

$$\begin{aligned} \frac{dV(a(t))}{dt} &= -\dot{u}(t)^T \dot{a}(t) \\ &= -\dot{u}(t)^T F'(t) \dot{u}(t) \\ &= -\sum_{n \in \Gamma} f'(u_n(t)) \left|\dot{u}_n(t)\right|^2 . \end{aligned} \tag{27}$$

This expression is valid for all time $t \geq 0$. In addition, because the function $f(\cdot)$ satisfies (6b), $f'(u_n(t)) > 0$, and thus

$$\begin{aligned} &\frac{dV(a(t))}{dt} \leq 0 \quad \text{for all } t \geq 0, \text{ and} \\ &\frac{dV(a(t))}{dt} < 0 \quad \text{for all } t \geq 0 \text{ such that } \left\|\dot{u}_\Gamma(t)\right\|_2 \neq 0. \end{aligned}$$

This means that $V(a(t))$ is non-increasing for all $t \geq 0$. Since $V(a(t))$ is continuous, bounded below by zero, and non-increasing, $V(a(t))$ converges to a constant value V^*, and its time derivative $\dot{V}(a(t))$ tends to zero as $t \to \infty$. Using equations (27) and (6b), we conclude that: $\lim_{t \to +\infty} \left\|\dot{u}_\Gamma(t)\right\|_2 = 0$. As a consequence, we also have $\lim_{t \to +\infty} \left\|\dot{a}(t)\right\|_2 = 0$, so the outputs converge to the set $E = \{a : \text{ s.t. } \dot{a}(t) = 0\}$, and the LCA outputs are quasi-convergent.

Moving on to part 3 of the theorem, we assume that the critical points of (1) are isolated. We need to show that both active and inactive nodes converge to a single fixed point. From part 2, we know that since the nodes converge to E, after some time, the active nodes will be within a ball of radius R around one element $a^* \in E$. However, since critical points of (1) are isolated, there exists a ball $\mathcal{B}_\epsilon(a^*)$ of radius $\epsilon > 0$ around a^* that does not contain any critical point:

$$a^* \in E \qquad \text{and} \qquad \forall a \in \mathcal{B}_\epsilon(a^*), a \neq a^* \Rightarrow a \notin E.$$

Since the system is stable, we know that once the trajectory gets close enough (within a ball of radius R) to one element in E, it cannot leave a ball of radius ϵ around this fixed point (see

(13)). As a consequence, the outputs remain within the ball $\mathcal{B}_\epsilon(a^*)$, which contains only the fixed point a^*. This proves that the active nodes converge to the point a^* in E:

$$\lim_{t\to+\infty} a(t) = a^*. \tag{28}$$

This implies that the ODE (3) can be written in terms of the distance $\widetilde{a}(t) = a(t) - a^*$ of the outputs from the solution:

$$\dot{u}(t) = -u(t) - \Phi^T\Phi a^* + \Phi^T y + a^* - \Phi^T\Phi\widetilde{a}(t) + \widetilde{a}(t).$$

We let

$$u^* = -\Phi^T\Phi a^* + \Phi^T y + a^*,$$

and rewrite the ODE (3) as:

$$\dot{u}(t) = -u(t) + u^* - \left(\Phi^T\Phi - I\right)\widetilde{a}(t)$$

Solving this ODE for all $t \geq 0$ yields:

$$u(t) = u^* + e^{-t}\left(u(0) - u^*\right) + e^{-t}\int_0^t e^s\left(\Phi^T\Phi - I\right)\widetilde{a}(s)ds.$$

While it is difficult to say anything directly about the trajectory of the system, it is helpful to consider a surrogate trajectory that is a straight line in the state-space: $u^* + e^{-t}\left(u(0) - u^*\right)$. This linear path obviously converges to the fixed point u^*, and if we are able to show that the actual trajectory $u(t)$ asymptotically approaches this idealized linear path, we will have established that the system converges to u^*. To take this approach, we examine the quantity

$$h(t) = u(t) - u^* - e^{-t}\left(u(0) - u^*\right),$$

which is the deviation from the linear path. Consider the norm of this deviation:

$$\begin{aligned}\left\|h(t)\right\|_2 &= \left\|u(t) - u^* - e^{-t}\left(u(0) - u^*\right)\right\|_2 \\ &= \left\|e^{-t}\int_0^t e^s\left(\Phi^T\Phi - I\right)\widetilde{a}(s)ds\right\|_2 \\ &\leq e^{-t}\left\|\Phi^T\Phi - I\right\|_2\int_0^t e^s\left\|\widetilde{a}(s)\right\|_2 ds.\end{aligned}$$

To show convergence to zero, we split the integral into two parts. Since $\widetilde{a}(t) \xrightarrow[t\to+\infty]{} 0$, then for any $\widetilde{\epsilon} > 0$, there exists a time $t_c \geq 0$ such that $\forall t \geq t_c$, $\left\|\widetilde{a}(t)\right\|_2 \leq \widetilde{\epsilon}$. Moreover, since $\left\|\widetilde{a}(t)\right\|_2$ is continuous and goes to zero as t goes to infinity, it admits a maximum μ, $\forall t \in \mathbb{R}$. This yields, for all $t \geq 2t_c$:

$$\begin{aligned}\left\|h(t)\right\|_2 &\leq e^{-t}\left\|\Phi^T\Phi - I\right\|_2\mu\int_0^{t_c} e^s ds \\ &\qquad + e^{-t}\left\|\Phi^T\Phi - I\right\|_2\widetilde{\epsilon}\int_{t_c}^t e^s ds \\ &\leq \left\|\Phi^T\Phi - I\right\|_2\mu\left[e^{-t+t_c} - e^{-t}\right] \\ &\qquad + \left\|\Phi^T\Phi - I\right\|_2\widetilde{\epsilon}\left[1 - e^{-t+t_c}\right] \\ &\leq \left\|\Phi^T\Phi - I\right\|_2\mu\left[e^{-t/2} - e^{-t}\right] + \left\|\Phi^T\Phi - I\right\|_2\widetilde{\epsilon}.\end{aligned}$$

Since the left term converges to 0 and $\widetilde{\epsilon}$ can be chosen arbitrarily small, this shows that the trajectory $u(t)$ converges to the trajectory $u^* + e^{-t}\left(u(0) - u^*\right)$ as t goes to infinity, and thus, we can conclude that $u(t) \xrightarrow[t\to+\infty]{} u^*$.

Because we have shown separately that both the active and inactive nodes converge for any initial state, it concludes our proof that the system is globally convergent. ■

Appendix B
Proof of Lemma 1

Proof: Each of the four cases will be treated separately.

(i) For any $\widetilde{u}_n \in \mathbb{R}$, let $z_n = \operatorname{sign}(\widetilde{u}_n)$. Since the thresholding function is non-decreasing (6b), and $T_\lambda(-u_n) = -T_\lambda(u_n)$ from (6a), we have:

$$\begin{aligned}z_n = \operatorname{sign}(\widetilde{u}_n) &\Rightarrow 0 \leq z_n\widetilde{u}_n \\ &\Rightarrow z_n u_n^* \leq z_n\widetilde{u}_n + z_n u_n^* \\ &\Rightarrow T_\lambda\left(z_n u_n^*\right) \leq T_\lambda\left(z_n\widetilde{u}_n + z_n u_n^*\right) \\ &\Rightarrow z_n T_\lambda\left(u_n^*\right) \leq z_n T_\lambda\left(\widetilde{u}_n + u_n^*\right) \\ &\Rightarrow 0 \leq z_n\left[T_\lambda\left(\widetilde{u}_n + u_n^*\right) - T_\lambda\left(u_n^*\right)\right] \\ &\Rightarrow 0 \leq z_n\widetilde{a}_n \\ &\Rightarrow \operatorname{sign}(\widetilde{a}_n) = z_n = \operatorname{sign}(\widetilde{u}_n).\end{aligned}$$

(ii) We can separate this proof into four cases.
If $|\widetilde{u}_n + u_n^*| \leq \lambda$ and $|u_n^*| \leq \lambda$, we have: $|\widetilde{a}_n| = |T_\lambda(\widetilde{u}_n + u_n^*) - T_\lambda(u_n^*)| = 0 \leq \alpha\left|\widetilde{u}_n\right|$. If $|\widetilde{u}_n + u_n^*| \leq \lambda$ and $|u_n^*| > \lambda$, according to the mean value theorem, since the function $f(\cdot)$ is continuous on $[\ \operatorname{sign}(u_n^*)\lambda, u_n^*]$, differentiable on $(\ \operatorname{sign}(u_n^*)\lambda, u_n^*)$ and $f(\ \operatorname{sign}(u_n^*)\lambda) = 0$, there exist $c \in (\ \operatorname{sign}(u_n^*)\lambda, u_n^*)$ such that:

$$\begin{aligned}|\widetilde{a}_n| = |f(u_n^*)| &= |f'(c)\left(u_n^* - \operatorname{sign}(u_n^*)\lambda\right)| \\ &= |f'(c)|\left(|u_n^*| - \lambda\right) \quad (\text{since } |u_n^*| > \lambda) \\ &\leq \alpha\left(|u_n^*| - \lambda\right) \\ &\leq \alpha\left|\widetilde{u}_n\right| \quad (\text{since } |u_n^*| - |\widetilde{u}_n| \leq |\widetilde{u}_n + u_n^*| \leq \lambda).\end{aligned}$$

If $|\widetilde{u}_n + u_n^*| > \lambda$ and $|u_n^*| \leq \lambda$, according to the mean value theorem, there exists $c \in (\ \operatorname{sign}(\widetilde{u}_n + u_n^*)\lambda, \widetilde{u}_n + u_n^*)$ such that:

$$\begin{aligned}|\widetilde{a}_n| &= |f(\widetilde{u}_n + u_n^*)| \\ &= |f'(c)\left(\widetilde{u}_n + u_n^* - \operatorname{sign}(\widetilde{u}_n + u_n^*)\lambda\right)| \\ &= |f'(c)|\left(|\widetilde{u}_n + u_n^*| - \lambda\right) \\ &\leq \alpha\left(|\widetilde{u}_n| + |u_n^*| - \lambda\right) \leq \alpha\left|\widetilde{u}_n\right|.\end{aligned}$$

Finally, if $|\widetilde{u}_n + u_n^*| > \lambda$ and $|u_n^*| > \lambda$, according to the mean value theorem, there exists $c \in (u_n^*, \widetilde{u}_n + u_n^*)$ such that $|\widetilde{a}_n| = |f(\widetilde{u}_n + u_n^*) - f(u_n^*)| = |f'(c)\widetilde{u}_n| \leq \alpha\left|\widetilde{u}_n\right|$.

(iii) Using property (i) and (ii), we have:

$$\begin{aligned}\widetilde{a}_{\mathcal{T}}^T\widetilde{a}_{\mathcal{T}} &= \sum_{n\in\mathcal{T}}\widetilde{a}_n\widetilde{a}_n = \sum_{n\in\mathcal{T}}|\widetilde{a}_n|\,|\widetilde{a}_n| \\ &\leq \sum_{n\in\mathcal{T}}\alpha\left|\widetilde{u}_n\right|\left|\widetilde{a}_n\right| = \alpha\sum_{n\in\mathcal{T}}\widetilde{u}_n\widetilde{a}_n = \alpha\widetilde{u}_{\mathcal{T}}^T\widetilde{a}_{\mathcal{T}} \\ &\leq \sum_{n\in\mathcal{T}}\alpha\left|\widetilde{u}_n\right|\alpha\left|\widetilde{u}_n\right| = \alpha^2\sum_{n\in\mathcal{T}}\widetilde{u}_n\widetilde{u}_n = \alpha^2\widetilde{u}_{\mathcal{T}}^T\widetilde{u}_{\mathcal{T}}.\end{aligned}$$

(iv) Since the soft-threshold function is non-decreasing, then the function $g_n(s) = T_\lambda(s + u_n^*) - T_\lambda(u_n^*)$ is also non-decreasing. Moreover $g_n(0) = 0$ and $g_n(\widetilde{u}_n(t)) = \widetilde{a}_n(t)$.

As a consequence, we can bound the integral by:

$$\int_0^{\widetilde{u}_n(t)} g_n(s)ds \leq \Big(g_n(\widetilde{u}_n(t)) - g_n(0)\Big)\widetilde{u}_n(t) = \widetilde{a}_n\widetilde{u}_n.$$

As a consequence:

$$\sum_{n=1}^{N}\int_0^{\widetilde{u}_n(t)} g_n(s)ds \leq \sum_{n=1}^{N}\widetilde{a}_n\widetilde{u}_n = \widetilde{a}^T\widetilde{u}.$$

■

References

[1] A. Balavoine, C. J Rozell, and J. Romberg, "Global convergence of the locally competitive algorithm," in *2011 IEEE Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE)*. Jan. 2011, pp. 431–436, IEEE.

[2] M. Elad, M. A.T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, June 2010.

[3] B. A. Olshausen and D. J. Field, "Sparse coding of sensory inputs," *Current Opinion in Neurobiology*, vol. 14, no. 4, pp. 481–487, Aug. 2004, PMID: 15321069.

[4] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, June 1996.

[5] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," *Vision Research*, vol. 37, no. 23, pp. 3311–3325, Dec. 1997, PMID: 9425546.

[6] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489– 509, Feb. 2006.

[7] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[8] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An Interior-Point method for Large-Scale L1-Regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[9] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–598, Dec. 2007.

[10] 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, Aug. 2004.

[11] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, Wiley, 1993.

[12] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554 –2558, Apr. 1982.

[13] C. Twigg and P. Hasler, "Configurable analog signal processing," *Digital Signal Processing*, vol. 19, no. 6, pp. 904–922, Dec. 2009.

[14] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neural Computation*, vol. 20, no. 10, pp. 2526–2563, Oct. 2008.

[15] J. P Hespanha, "Uniform stability of switched linear systems: Extensions of LaSalle's invariance principle," *IEEE Trans. Autom. Control*, vol. 49, no. 4, pp. 470– 482, Apr. 2004.

[16] S. Liu and J. Wang, "A simplified dual neural network for quadratic programming with its KWTA application," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1500–1510, Nov. 2006.

[17] H. Yang and T. S. Dillon, "Exponential stability and oscillation of hopfield graded response neural network," *IEEE Trans. Neural Netw.*, vol. 5, no. 5, pp. 719–729, Sept. 1994, PMID: 18267846.

[18] X.-B. Liang and J. Wang, "A recurrent neural network for nonlinear optimization with a continuously differentiable objective function and bound constraints," *IEEE Trans. Neural Netw.*, vol. 11, no. 6, pp. 1251–1262, Nov. 2000.

[19] M. Forti and A. Tesi, "Absolute stability of analytic neural networks: an approach based on finite trajectory length," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 12, pp. 2460– 2469, Dec. 2004.

[20] W. Lu and J. Wang, "Convergence analysis of a class of nonsmooth gradient systems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 11, pp. 3514–3527, Dec. 2008.

[21] Y. Xia, G. Feng, and J. Wang, "A recurrent neural network with exponential convergence for solving convex quadratic program and related linear piecewise equations," *Neural Networks*, vol. 17, no. 7, pp. 1003–1015, Sept. 2004, PMID: 15312842.

[22] L. V. Ferreira, E. Kaszkurewicz, and A. Bhaya, "Support vector classifiers via gradient systems with discontinuous righthand sides," *Neural Networks*, vol. 19, no. 10, pp. 1612–1623, Dec. 2006, ACM ID: 1223046.

[23] M. Forti and A. Tesi, "New conditions for global stability of neural networks with application to linear and quadratic programming problems," *IEEE Trans. Circuits Syst. I, Fundam. Theory Appl.*, vol. 42, no. 7, pp. 354–366, July 1995.

[24] L. Li and L. Huang, "Dynamical behaviors of a class of recurrent neural networks with discontinuous neuron activations," *Applied Mathematical Modelling*, vol. 33, no. 12, pp. 4326–4336, Dec. 2009.

[25] M. Forti, P. Nistri, and M. Quincampoix, "Generalized neural network for nonsmooth nonlinear programming problems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 51, no. 9, pp. 1741– 1754, Sept. 2004.

[26] M. Forti, P. Nistri, and M. Quincampoix, "Convergence of neural networks for programming problems via a nonsmooth Łojasiewicz inequality," *IEEE Trans. Neural Netw.*, vol. 17, no. 6, pp. 1471–1486, Nov. 2006.

[27] X. Xue and W. Bian, "Subgradient-Based neural networks for nonsmooth convex optimization problems," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 8, pp. 2378–2391, Sept. 2008.

[28] Q. S. Liu and J. Wang, "Finite-Time convergent recurrent neural network with a Hard-Limiting activation function for constrained optimization with Piecewise-Linear objective functions," *IEEE Trans. Neural Netw.*, vol. 22, no. 4, pp. 601–613, Apr. 2011.

[29] H. Lin and P. J. Antsaklis, "Stability and stabilizability of switched linear systems: A survey of recent results," *IEEE Trans. Autom. Control*, vol. 54, no. 2, pp. 308–322, Feb. 2009.

[30] W. Maass, "On the computational power of Winner-Take-All," *Neural Computation*, vol. 12, no. 11, pp. 2519–2535, 2000.

[31] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227–234, 1995.

[32] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, vol. 43, no. 1, pp. 129–159, Mar. 2001.

[33] D. L. Donoho and J. Tanner, "Sparse nonnegative solutions of underdetermined linear equations by linear programming," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 27, pp. 9446–9451, 2005.

[34] D. C. Knill and R. Pouget, "The Bayesian brain: The role of uncertainty in neural coding and computation," *Trends in Neuroscience*, vol. 27, pp. 712–719, 2004.

[35] M. Rehn and F. T. Sommer, "A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields," *Journal of Computational Neuroscience*, vol. 22, no. 2, pp. 135–146, Apr. 2007, PMID: 17053994.

[36] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, Feb. 2006.

[37] S. Fischer, R. Redondo, L. Perrinet, and G. Cristbal, "Sparse approximation of images inspired from the functional architecture of the primary visual areas," *EURASIP Journal on Advances in Signal Processing*, vol. 2007, pp. 1–17, 2007.

[38] L. Perrinet, "Feature detection using spikes: The greedy approach," *Journal of Physiology (Paris)*, vol. 98, no. 4-6, pp. 530–539, July 2004.

[39] J. M Bioucas-Dias and M. A.T. Figueiredo, "A new TwIST: Two-Step iterative Shrinkage/Thresholding algorithms for image restoration," *IEEE Trans. Image Process.*, vol. 16, no. 12, pp. 2992–3004, Dec. 2007.

[40] F. H. Clarke, *Optimization and Nonsmooth Analysis*, Society for Industrial Mathematics, Jan. 1987.

[41] R. A. DeCarlo, M. S. Branicky, S. Pettersson, and B. Lennartson, "Perspectives and results on the stability and stabilizability of hybrid systems," *Proc. IEEE*, vol. 88, no. 7, pp. 1069–1082, July 2000.

[42] A. Bacciotti and L. Rosier, *Liapunov functions and stability in control theory*, Springer, 2005.

[43] J.-J. Fuchs, "On sparse representations in arbitrary redundant bases," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1341–1344, June 2004.

[44] J. Bolte, A. Daniilidis, and A. Lewis, "The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems," *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 1205–1223, 2007.

[45] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, Aug. 2006.