A Common Network Architecture Efficiently Implements a Variety of Sparsity-based Inference Problems

Adam S. Charles[1], **Pierre Garrigues**[2], **Christopher J. Rozell**[1]
[1] School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA
[2]IQ Engines, Berkeley, CA

Keywords: Sparse approximation, sparse coding, optimization, inverse problems, analog architectures

Abstract

The sparse coding hypothesis has generated significant interest in the computational and theoretical neuroscience communities, but there remain open questions about the exact quantitative form of the sparsity penalty and the implementation of such a coding rule in neurally plausible architectures. The main contribution of this work is to show that a wide variety of sparsity-based probabilistic inference problems proposed in the signal processing and statistics literatures can be implemented exactly in the common network architecture known as the Locally Competitive Algorithm (LCA). Among the cost functions we examine are approximate ℓ_p norms ($0 \leq p \leq 2$), modified ℓ_p-norms, block-ℓ_1 norms, and re-weighted algorithms. Of particular interest is that we show significantly increased performance in re-weighted ℓ_1 algorithms by inferring all parameters jointly in a dynamical system rather than using an iterative approach native to digital computational architectures.

1 Introduction

New experimental approaches over the past decades have provided a closer look at how sensory nervous systems such as the visual cortex process information about their environment. Over this time it has become increasingly evident that the canonical *linear-nonlinear* model where cells encode visual information via linear filtering followed by a nonlinearity (e.g., thresholding and saturation) is inadequate to describe the complex processing performed by sensory cortex. For example, this type of linear-nonlinear model does not capture the rich variety of nonlinear response properties and contextual modulations observed in V1 (Seriès et al., 2003).

Many theoretical neuroscientists have postulated high level coding and computational principles for sensory cortex to attempt to further our understanding of these systems. In many cases, these proposals are based generally around probabilistic Bayesian inference (Doya, 2007) due to the natural fit with ecological goals and evidence from perceptual tasks in humans (Battaglia et al., 2003; Hürlimann et al., 2002). Many other researchers have postulated complimentary models based on the ideas of efficient coding where information is encoded by removing redundant aspects of the stimulus. A wide variety of interesting models have appeared related to this broad principle of efficient coding, with selected examples including models using predictive coding (Rao & Ballard, 1999; Spratling, 2011), divisive normalization (Schwartz & Simoncelli, 2001), and directly encoding statistical variations (Karklin & Lewicki, 2008; Coen-Cagli et al., 2012).

The sparse coding hypothesis is one interpretation of efficient coding that has generated significant interest in the computational and theoretical neuroscience communities. In this model, a population of cells performs Bayesian inference to determine the environmental causes of a stimulus, with a goal of using as few simultaneously active units in the encoding as possible. Distributed sparse neural codes have several potential benefits over dense linear codes, including explicit information representation and easy decodability at higher processing stages (Olshausen & Field, 2004), metabolic efficiency (due to the the significant cost of producing and transmitting action potentials (Lennie, 2003)), and increased capacity of associative and sequence memory models (Baum et al., 1988; Charles et al., 2012). The interest in the sparse coding model was originally generated when it was shown that this simple principle (combined with the statistics of natural images) is sufficient to explain the emergence of V1 receptive field shapes both qualitatively (Olshausen & Field, 1996) and quantitatively (Rehn & Sommer, 2007). More recently, electrophysiology experiments report results consistent with sparse coding (Haider et al., 2010; Vinje & Gallant, 2002), and simulation results have demonstrated that the sparse coding model can account for a wide variety of nonlinear response properties (called nonclassical receptive field effects) reported in single cells and population studies of V1 (Zhu & Rozell, 2012).

Despite this interest, there are many open fundamental questions related to the sparse coding model. First, what exactly is the proper notion of sparsity to use during inference? The original work in the computational neuroscience literature proposed several potential sparsity-inducing cost functions (Olshausen & Field, 1996), and recent work (motivated by strong theoretical results in the signal processing and applied mathematics communities) has seen people gravitate toward the ℓ_1 norm. While the main qualitative results appear to be relatively robust to the detailed choice of the sparsity-inducing cost function, the broader signal processing and statistics communities have proposed several alternative cost functions that have appealing computational or statistical properties and may be valuable alternatives. Second, how would such a coding principle be implemented in biologically plausible computational architectures? The computation necessary to implement an inference process with a sparsity penalty amounts to solving a non-smooth optimization problem that is notoriously challenging to solve (e.g., many gradient based methods are wildly inefficient due to the non-smooth nature of the objective). Recent theoretical work has demonstrated several network architectures that can efficiently compute sparse coefficients (Rehn & Sommer, 2007; Rozell

et al., 2010; Zylberberg et al., 2011; Perrinet et al., 2004). Interestingly, the sparse coding problem has become very prominent in modern signal processing (e.g., for use in inverse problems (Elad et al., 2010), computer vision (Wright et al., 2010), etc.), and there is also increasing interest in leveraging the computational benefits of analog neuromorphic architectures for these problems (Shapero et al., 2012a,b).

The main contribution of this work is to show that a wide variety of sparsity-based probabilistic inference problems can be implemented exactly in the common network architecture known as the Locally Competitive Algorithm (LCA) (Rozell et al., 2010). The LCA is a type of Hopfield network that is specifically designed to incorporate nonlinear thresholding elements that make it particularly efficient for solving the non-smooth optimization problems necessary for sparse coding. In particular, we examine sparsity-based approaches described in the recent statistics and signal processing literature to show that many proposed signal models based on sparsity principles can be implemented efficiently in this common neural architecture. Among the cost functions we examine are approximate ℓ_p norms ($0 \leq p \leq 1$), modified ℓ_p-norms that combine desirable properties of different statistical models, block-ℓ_1 norms for use in hierarchical models that impose correlations among the active variables, and re-weighted algorithms that use a hierarchical probabilistic model to achieve more efficient encodings. Of particular interest is that we show significantly increased performance in re-weighted ℓ_1 algorithms by inferring all parameters jointly in a dynamical system rather than using an iterative approach native to digital computational architectures. Preliminary results related to this work were reported in (Rozell & Garrigues, 2010).

2 Background and related work

2.1 Sparse Coding

In the sparse coding problem, we use probabilistic inference to find the smallest number of causes for an observed signal under a linear generative model

$$\boldsymbol{x} = \Phi \boldsymbol{a} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^M$ is the observed signal, $\boldsymbol{a} \in \mathbb{R}^N$ is the coefficient vector, $\Phi \in \mathbb{R}^{M \times N}$ is the dictionary of causes, and $\boldsymbol{\epsilon}$ is Gaussian noise. The coefficient vector is said to be sparse as we seek a solution with relatively few non-zero entries. The coefficients $\boldsymbol{a}$ are generally inferred via MAP estimation, which results in solving a non-linear optimization problem

$$\min_{\boldsymbol{a}} \; E = \frac{1}{2} \left|\left| \boldsymbol{x} - \Phi \boldsymbol{a} \right|\right|_2^2 + \lambda \widetilde{C}\left(\boldsymbol{a}\right), \tag{2}$$

where $\widetilde{C}\left(\cdot\right)$ is a cost function penalizing $\boldsymbol{a}$ based on its fit with the signal model, and λ is a parameter denoting the relative tradeoff between the data fidelity term (i.e., MSE, which arises from the log likelihood of the Gaussian noise) and the cost function. The cost function is the non-linear sparsity-inducing regularization term, corresponding to the log prior of the data model. More details about the formulation of this problem in the Bayesian inference framework can be found in (Olshausen & Field, 1997). Basic signal models frequently assume independence among the elements of $\boldsymbol{a}$, resulting in a

cost function that separates into a sum of individual costs $\left(\text{i.e., } \widetilde{C}(\boldsymbol{a}) = \sum_k C(a_k)\right)$. One common example is the ℓ_p norm, defined as $\widetilde{C}(\boldsymbol{a}) = \|\boldsymbol{a}\|_p^p = (\sum_i a_i^p)$.

2.2 Dynamical systems for ℓ_1 minimization

As mentioned above, recent work in computational neuroscience has shown that the LCA dynamical system provably solves the optimization programs in (2) and are efficient for solving the non-smooth problems of interest in sparse approximation. The LCA (Rozell et al., 2010) architecture is comprised of a network of analog nodes being driven by the signal to be approximated. Each node competes with neighboring nodes for a chance to represent the signal, and the steady-state response represents the solution to the optimization problem.

The LCA is a specific type of Hopfield neural network, which have a long history of being used to solve optimization problems (Hopfield, 1982). The LCA is a neurally plausible architecture, consisting of a network of parallel nodes that use computational primitives that are well-matched to individual neuron models. In particular, each node consists of a leaky integrator and a non-linear thresholding function, and it is driven by both feedforward and lateral (inhibitory and excitatory) recurrent connections. This architecture has been implemented in neuromorphic hardware, both as a purely analog system (Shapero et al., 2012a) and by using integrate and fire spiking neurons for each node (Shapero et al., 2012b). We also note that other types of network structures have also been proposed recently to approximately solve specific versions of the sparse approximation problem (Rehn & Sommer, 2007; Perrinet et al., 2004; Zylberberg et al., 2011; Hu et al., 2012).

Specifically, the k^{th} node of the LCA is associated with $\boldsymbol{\phi}_k$, the k^{th} column of Φ. Without loss of generality, we assume each column has unit norm. This node is described at a given time t by an internal state variable $u_k(t)$. The coefficients $\boldsymbol{a}$ are related to the internal states $\boldsymbol{u}$ via an activation (thresholding) function $\boldsymbol{a}(t) = \widetilde{T}_\lambda(\boldsymbol{u}(t))$ that is parametrized by λ. In the important special case when the cost function is separable, the output of each node k can be calculated independently of all other nodes by a pointwise activation function $a_k(t) = T_\lambda(u_k(t))$. Individual nodes are leaky integrators driven by an input proportional to $\langle \boldsymbol{\phi}_k, \boldsymbol{x} \rangle$, and competition between nodes occurs via lateral connections that allow highly active nodes to suppress nodes with less activity. The dynamics for node k are given by:

$$\dot{u}_k(t) = \frac{1}{\tau}\left[\langle \boldsymbol{x}, \boldsymbol{\phi}_k \rangle - u_k(t) - \sum_{\substack{j=1 \\ j \neq k}}^{N} \langle \boldsymbol{\phi}_k, \boldsymbol{\phi}_j \rangle a_j(t)\right], \tag{3}$$

where τ is the system time constant. In vector form, the dynamics for the whole network are given by:

$$\dot{\boldsymbol{u}}(t) = \frac{1}{\tau}\left[\Phi^t \boldsymbol{x} - \boldsymbol{u}(t) - \left(\Phi^t \Phi - I\right)\boldsymbol{a}(t)\right]. \tag{4}$$

In (Rozell et al., 2010) it was shown that for the energy surface E given in (2) with a separable, continuous and piecewise differentiable cost function, the path induced by

the LCA (using the outputs $a_k(t)$ as the optimization variable) ensures $\frac{dE(t)}{dt} \leq 0$ when the cost function satisfies:

$$\lambda \frac{dC\left(a_k\right)}{da_k} = u_k - a_k = u_k - T_\lambda\left(u_k\right) = T_\lambda^{-1}(a_k) - a_k, \tag{5}$$

where $T_\lambda\left(u_k\right)$ is non-decreasing. We use the notation $T_\lambda^{-1}(u_k)$ for convenience when the activation function is invertible, but this invertibility is not strictly required (i.e., the relationship in (5) involving just $T_\lambda\left(u_k\right)$ is sufficient). The same arguments also extend to the more general case of non-separable cost functions, ensuring $\frac{dE(t)}{dt} \leq 0$ when

$$\lambda \nabla_{\boldsymbol{a}} \widetilde{C}\left(\boldsymbol{a}\right) = \boldsymbol{u} - \boldsymbol{a} = \boldsymbol{u} - \widetilde{T}_\lambda\left(\boldsymbol{u}\right) = \widetilde{T}_\lambda^{-1}\left(\boldsymbol{a}\right) - \boldsymbol{a}. \tag{6}$$

Recent followup work (Balavoine et al., 2011) establishes stronger guarantees on the LCA, specifically showing that this system is globally convergent to the minimum of E (which may be a local minima if $C\left(\cdot\right)$ is not convex) and proving that the system converges exponentially fast with an analytically bounded convergence rate.

The relationship in (5) requires cost functions that are differentiable and activation functions that are invertible. However, the cost function for BPDN (the ℓ_1 norm) is non-smooth at the origin and the most effective sparsity-promoting activation functions will likely have non-invertible thresholding properties. In these cases, one can start with a smooth cost function that is a relaxed version of the desired cost and calculate the corresponding activation function. Taking the limit of the relaxation parameter in the activation function yields a formula for $T_\lambda\left(\cdot\right)$ that can be used to solve the desired problem. Specifically, in the appendix we use the log-barrier relaxation (Boyd & Vandenberghe, 2004) to show that the LCA solves BPDN when the activation function is the well-known soft thresholding function:

$$C\left(a_k\right) = \left|a_k\right| \quad \Longleftrightarrow \quad a_k = T_\lambda\left(u_k\right) = \begin{cases} 0 & \left|u_k\right| \leq \lambda \\ u_k - \lambda \text{sign}(u_k) & \left|u_k\right| > \lambda \end{cases}.$$

Similarly, the LCA can find a local minima to the non-convex optimization program that minimizes the ℓ_0 "norm" of the coefficients (i.e., number of non-zeros) by using the hard thresholding activation function (Rozell et al., 2010):

$$C\left(a_k\right) = I\left(a_k \neq 0\right) \quad \Longleftrightarrow \quad a_k = T_\lambda\left(u_k\right) = \begin{cases} 0 & \left|u_k\right| \leq \lambda \\ u_k & \left|u_k\right| > \lambda \end{cases},$$

where $I(\cdot)$ is the standard indicator function.

3 Alternate inference problems in the LCA architecture

Using the basic relationships described in (5) and (6), a variety of cost functions can be optimized in the same basic LCA structure by analytically determining the corresponding activation function.[1] These optimization programs include approximate ℓ_p norms,

[1] We also note that a cost function might be easily implementable even in the absence of an analytic formula for the activation function simply by using numerical integration to find a solution and fitting the resulting curve.

modified ℓ_p norms that attempt to achieve better statistical properties than BPDN, the group/block ℓ_1 norm that induces co-activation structure on the non-zero coefficients, re-weighted ℓ_1 and ℓ_2 algorithms that represent hierarchical statistical models on the coefficients, and classic Tikhonov regularization.

Before exploring specific alternate cost functions in the remainder of this section, it is worthwhile to make a technical note regarding the optimization programs that are possible to implement in the LCA architecture. The strong theoretical convergence guarantees established for the LCA (Balavoine et al., 2011) apply to a wide variety of possible systems, but do impose some conditions on the permissible activation functions. We will rely on these same conditions to analytically determine the relationship between the cost and activation functions for the examples in this section. Translated to conditions on the cost functions, the convergence results for the LCA (Balavoine et al., 2011) require that the cost functions be positive $\left(\widetilde{C}\left(\boldsymbol{a}\right) \geq 0\right)$, symmetric $\left(\widetilde{C}\left(-\boldsymbol{a}\right) = \widetilde{C}\left(\boldsymbol{a}\right)\right)$, and satisfy the condition that the matrix $\left(\lambda \nabla^2_{\boldsymbol{a}} \widetilde{C}\left(\boldsymbol{a}\right) + \boldsymbol{I}\right)$ is positive definite (i.e., $\lambda \partial^2 C\left(a_k\right) / \partial a_k^2 + 1 > 0$ for separable cost functions). This last condition can intuitively be viewed as requiring that the activation function resulting from (6) has only a single output for a given input.

Some of the cost functions considered here have non-zero derivatives at the origin, leading to a range of values around the origin where $T_\lambda\left(u_k\right)$ is not defined according to the relationship in (5). In these cases, the smallest value for which the threshold function is defined results in a zero-valued output (i.e., $T_\lambda\left(u_k\right) = 0$ at $u_k = \lim_{a_k \to 0^+} \lambda \partial C\left(a_k\right) / \partial a_k$). Since the second derivative condition on the cost function constrains the activation function to be non-decreasing, we can infer that the only allowable value of the activation function must be zero for the regions that are not well-characterized by the relationship in (5). Finally, we note that in most cases we will only consider the behavior of the activation function for $u_k \geq 0$ because the behavior for $u_k < 0$ is implied by the symmetry condition.

3.1 Approximate ℓ_p norms $(0 \leq p \leq 2)$

Perhaps the most widely used family of cost functions are the ℓ_p norms $\widetilde{C}\left(\boldsymbol{a}\right) = \|\boldsymbol{a}\|_p^p$. These separable cost functions include ideal sparse approximation (i.e., counting non-zeros), BPDN, and Tikhonov Regularization (Tikhonov, 1963) as special cases ($p = 0, 1$ and 2, respectively), and are convex for $p \geq 1$. Furthermore, recent research has shown some benefits of using non-convex ℓ_p norms ($p < 1$) for inverse problems with sparse signal models (Saab et al., 2008; Elad et al., 2007). While the ideal activation functions can be determined exactly for the three special cases mentioned above ($p = 0, 1$ and 2), it is not possible to analytically determine the activation function for arbitrary values of $0 \leq p \leq 2$. Elad et al. (Elad et al., 2007) recently introduced several parameterized approximations to the ℓ_p cost functions that are more amenable to analysis. In this section, we use these same approximations to determine activation functions for minimizing approximate ℓ_p norms for $0 \leq p \leq 2$.

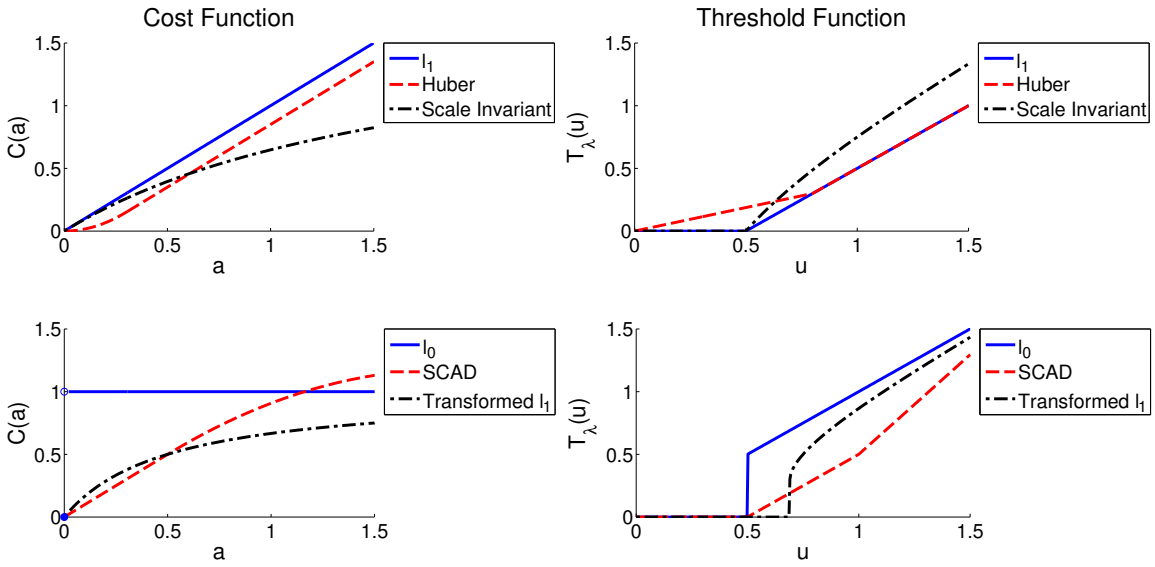

Figure 1: Cost functions and their corresponding thresholding functions. Left: The cost functions are compared for the (top) ℓ_1 with $\lambda = 0.5$, scale invariant Bayes with $\lambda = 0.5$, the Huber cost with $\lambda = 0.5$ and $\epsilon = 0.3$ and (bottom) ℓ_0 with $\lambda = 0.5$, SCAD with $\lambda = 0.5$ and $\kappa = 3.7$ and transformed ℓ_1 with thresh = 0.5 and $\beta = 2$. Right: The corresponding nonlinear activation function which can be used in the LCA to solve the regularized optimization program for each cost function.

Approximate ℓ_p for $1 \leq p \leq 2$

For $1 \leq p \leq 2$, Elad et al. (Elad et al., 2007) propose the approximate cost function

$$C(\boldsymbol{a}) = \sum_k \left[c|a_k| - cs \log\left(1 + \frac{|a_k|}{s}\right)\right],$$

as a good match for the true ℓ_p norm for some value of parameters s and c. In the limiting cases, $c = 1$ with $s \to 0$ yields the ℓ_1 norm and $c = 2s$ with $s \to \infty$ yields the ℓ_2 norm. Three intermediate examples for p = 1.25, 1.5 and 1.75 are shown in Figure 2. For any specific value of p, we find the best values of c and s by using standard numerical optimization techniques to minimize the squared error to the true cost function over the interval [0,2]. From this cost function, we can differentiate to obtain the relationship between each u_k and a_k as

$$u_k = a_k + \lambda \frac{ca_k}{s + a_k}.$$

We see from this relationship that with $c = 1$ and $s \to 0$, we obtain $a_k = u_k - \lambda$ for $u_k > \lambda$ (i.e., the soft-thresholding function for BPDN), while with $c = 2s$ and $s \to \infty$ we obtain $a_k = \frac{u_k}{1+2\lambda}$ (i.e., a linear amplifier for Tikhonov Regularization). Solving for a_k in terms of u_k (restricting the solution to be positive and increasing) yields a general relationship for the activation function

$$T_\lambda(u_k) = \frac{1}{2}\left[u_k - s - c\lambda + \sqrt{(u_k - s - c\lambda) + 4u_k s}\right].$$

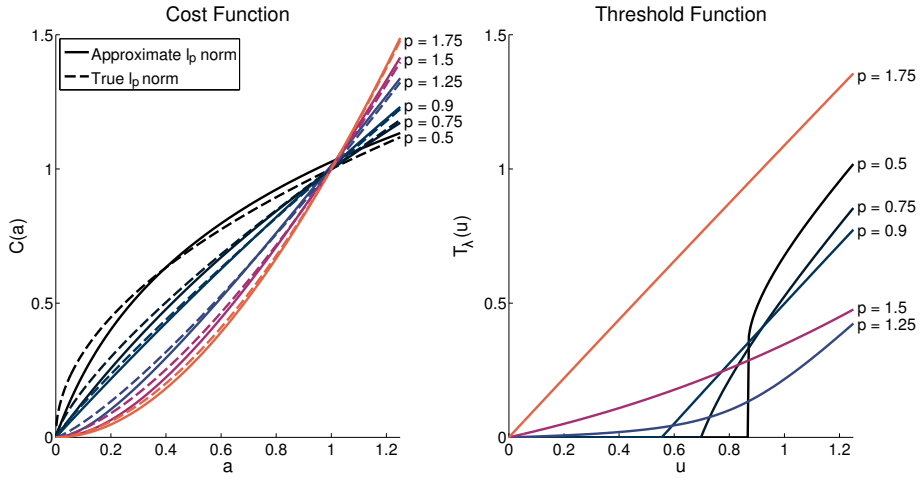

Figure 2: Approximate ℓ_p cost functions and their corresponding thresholding functions. Left: The cost functions are approximated over the parameters c, s for values of p ranging from 0 to 1 (top) and 1 to 2 (bottom). The true ℓ_p costs are shown as dotted lines in the same colors. Using these values of c and s, a nonlinear activation function that can be used in the LCA to solve the optimization is plotted (right) using the thresholding equations for $0 < p < 1$ (top) and $1 < p < 2$ (bottom). The thresholding functions clearly span the ranges between soft and hard thresholding for the lower range of p and between soft thresholding and linear amplification for the upper range of p.

This solution is shown in Figure 2 for p = 1.25, 1.5 and 1.75 for λ = 0.5.

Approximate ℓ_p for $0 \leq p \leq 1$

For $0 \leq p \leq 1$, Elad et al. (Elad et al., 2007) also propose the following approximate cost function as a good match for the true ℓ_p norm for some value of parameters s and c:

$$C\left(a_k\right) = cs \log\left(1 + \frac{|a_k|}{s}\right),$$

where the parameters $c > 0$ and $s > 0$ can be optimized as above to approximate different values of p. Three approximations for p = 0.5, 0.75 and 0.9 are shown in Figure 2. To determine the activation function, we again differentiate and find the appropriate relationship to be

$$a_k + \frac{\lambda cs}{s + a_k} = u_k.$$

Solving for a_k reduces to solving a quadratic equation, which leads to two possible solutions. As above, we restrict the activation function to only include the solution that is positive and increasing, resulting in the activation function

$$T_\lambda\left(u_k\right) = \frac{1}{2}\left(u_k - s + \sqrt{\left(u_k + s\right)^2 - 4\lambda cs}\right).$$

This activation function is only valid over the range where the output is a positive real number. If $c\lambda \leq s$, this condition reduces to $u_k \geq c\lambda$. More generally, this condition reduces to $u_k \geq 2\sqrt{2cs\lambda} - s$.

3.2 Modified ℓ_p norms

While the general ℓ_p norms have historically been very popular cost functions, many people have noted that this approach can have undesirable statistical properties in some instances (e.g., BPDN can result in biased estimates of large coefficients (Zou, 2006)). To address these issues, many researchers in signal processing and statistics have proposed modified cost functions that attempt to alleviate these statistical concerns. For example, hybrid ℓ_p norms smoothly morph between different norms to capture the most desirable characteristics over different regions. In this section we will demonstrate that many of these modified ℓ_p norms can also be implemented in the basic LCA architecture.

Smoothly Clipped Absolute Deviations

A common goal for modified ℓ_p norms is to retain the continuity of the cost function near the origin demonstrated by the ℓ_1 norm, while using a constant cost function for larger coefficients (similar to the ℓ_0 norm) to avoid statistical biases. One approach to achieving these competing goals is the smoothly clipped absolute deviations (SCAD) penalty (Fan, 1997; Antoniadis & Fan, 2001). The SCAD approach directly concatenates the ℓ_1 and ℓ_0 norms with a quadratic transition region, resulting in the cost function given by

$$C\left(a_k\right) = \begin{cases} a_k & 0 < a_k \leq \lambda \\ \frac{1}{(\kappa-1)\lambda}(a_k\kappa\lambda - \frac{a_k^2}{2} - \frac{\lambda^2}{2}) & \lambda < a_k \leq \kappa\lambda \\ \frac{\lambda}{2}(1+\kappa) & \kappa\lambda < a_k \end{cases},$$

for $\kappa \geq 1$ (κ defines the width of the transition region). An example of this cost function with $\lambda = 0.5$ and $\kappa = 3.7$ is shown in Figure 1.

To obtain the activation function we again solve $\lambda\frac{dC(a_k)}{da_k} + a_k = u_k$ for a_k as a function of u_k. For SCAD (and all of the piecewise cost functions we consider), the activation function can be determined individually for each region, paying careful attention to the ranges of the inputs u_k and outputs a_k to ensure consistency. For $0 < a_k \leq \lambda$, we have $\lambda + a_k = u_k$, implying that $a_k = 0$ for $u_k < \lambda$ and $a_k = u_k - \lambda$ over the interval $\lambda < u_k < 2\lambda$. For $\lambda < a_k \leq \kappa\lambda$, we have

$$\lambda\frac{(\kappa\lambda - a_k)}{(\kappa-1)\lambda} + a_k = u_k \implies a_k = \frac{(\kappa-1)u_k - \kappa\lambda}{\kappa - 2}$$

over the interval $2\lambda < u_k < \kappa\lambda$. Finally, for $\kappa\lambda < a_k$ we have $a_k = u_k$, giving the full activation function

$$a_k = T_\lambda\left(u_k\right) = \begin{cases} 0 & u_k \leq \lambda \\ u_k - \lambda & \lambda \leq u_k \leq 2\lambda \\ \frac{\kappa-1}{\kappa-2}u_k - \frac{\kappa\lambda}{\kappa-2} & 2\lambda \leq u_k \leq \kappa\lambda \\ u_k & \kappa\lambda \leq u_k \end{cases},$$

which is shown in Figure 1 for λ = 0.5 and $\kappa = 3.7$. Note that this activation function requires $\kappa \geq 2$ (Antoniadis and Fan recommend a value of $\kappa = 3.7$ (Antoniadis &

Fan, 2001)). While this is apparent from consistency arguments once the thresholding function has been derived, this restriction on κ can also be deduced from the condition $\lambda \partial^2 C\left(a_k\right) / \partial a_k^2 + 1 > 0$.

Transformed ℓ_1

Similar to the SCAD cost function, the transformed ℓ_1 cost (Antoniadis & Fan, 2001; Nikolova, 2000) attempts to capture something close to the ℓ_1 norm for small coefficients while reducing the penalty on larger coefficients. Specifically, transformed ℓ_1 uses the fractional cost function given by

$$C\left(a_k\right) = \frac{\beta |a_k|}{1 + \beta |a_k|},$$

for some $\beta > 0$. An example of this cost with $\beta = 2$ and λ = 0.5 is shown in Figure 1. After calculating the derivative of the cost function, the activation function can be found by solving

$$\frac{\lambda \beta}{(1 + \beta a_k)^2} + a_k = u_k$$

for a_k. Inverting this equation reduces to solving a cubic equation in a_k. The three roots can be calculated analytically, but only one root generates a viable thresholding function by being both positive and increasing for positive u_k. That root is given by

$$\begin{aligned} a_k &= \frac{\beta u_k - 2}{3\beta} + \frac{2^{\frac{2}{3}}}{6\beta} \left(6\beta u_k - 27\beta^2 \lambda + 6\beta^2 u_k^2 + 2\beta^3 u_k^3 \right. \\ &+ \left. 3\sqrt{3}\beta^3 \sqrt{-\frac{\lambda\left(4\beta^3 u_k^3 + 12\beta^2 u_k^2 - 27\lambda\beta^2 + 12\beta u_k + 4\right)}{\beta^4}} + 2 \right)^{\frac{1}{3}} \\ &+ \frac{\beta 2^{\frac{1}{3}} (\beta u_k + 1)^2}{3\left(6\beta u_k - 27\beta^2\lambda + 6\beta^2 u_k^2 + 2\beta^3 u_k^3 + 3\sqrt{3}\beta^3 \sqrt{-\frac{\lambda\left(4\beta^3 u_k^3 + 12\beta^2 u_k^2 - 27\lambda\beta^2 + 12\beta u_k + 4\right)}{\beta^4}} + 2 \right)^{\frac{1}{3}}} \end{aligned}.$$

This solution is viable only when a_k is real valued, which corresponds to the range $u_k \geq \left(3\left(\frac{\lambda}{4\beta}\right)^{1/3} - \frac{1}{\beta} \right)$. Outside of this range, no viable non-zero solution exists and so a_k = 0. The full thresholding function is shown in Figure 1 for λ = 0.5 and β = 2..

Huber Function

The Huber cost function (Huber, 1973) aims to modify standard ℓ_2 optimization to improve the robustness to outliers. This cost function consists of a quadratic cost function on smaller values and a smooth transition to an ℓ_1 cost on larger values, given by

$$C\left(a_k\right) = \begin{cases} \frac{a_k^2}{2\epsilon} & 0 \leq |a_k| \leq \epsilon \\ |a_k| - \frac{\epsilon}{2} & \epsilon < |a_k| \end{cases}.$$

An example of the Huber cost is shown in Figure 1 for λ = 0.5 and ϵ = 0.3. As in the case of other piecewise cost functions, we calculate the activation function separately over

each interval of interest by calculating the derivative of the cost function in each region. For the first interval, the relationship is given by $\frac{\lambda a_k}{\epsilon} = u_k - a_k$, which obviously gives the activation function $T_\lambda(u_k) = \frac{\epsilon u_k}{\epsilon+\lambda}$ for $|u_k| \leq \epsilon + \lambda$. For the second interval, we have $\lambda \frac{a_k}{|a_k|} = u_k - a_k$, which yields the activation function $T_\lambda(u_k) = u_k\left(1 - \frac{\lambda}{|u_k|}\right)$ for $|u_k| > \epsilon + \lambda$. Putting the pieces together, the full activation function (as expected) is a mixture of the Tikhonov regularization and the soft thresholding used for ℓ_1 optimization given by

$$a_k = T_\lambda(u_k) = \begin{cases} \frac{\epsilon u_k}{\epsilon+\lambda} & |u_k| \leq \epsilon + \lambda \\ u_k\left(1 - \frac{\lambda}{|u_k|}\right) & |u_k| > \epsilon + \lambda \end{cases},$$

which is shown in Figure 1 for λ = 0.5 and ϵ = 0.3. We can see that as $\epsilon \to 0$, the cost function converges to the ℓ_1 norm and the thresholding function correctly converges back to the soft-threshold function derived earlier using the log-barrier method.

Amplitude Scale Invariant Bayes Estimation

A known problem with using the ℓ_1 norm as a cost function is that it is not scale invariant, meaning that the results can be poor if the amplitude of the input signals changes significantly (assuming a constant value of λ). Many cost functions (including the ones presented above) are heuristically motivated, drawing on intuition and tradeoffs between the behavior of various ℓ_p norms. In contrast, Figueiredo and Nowak (Figueiredo & Nowak, 2001) approach the problem from the perspective of Bayesian inference with a Jeffreys' prior to determine a cost function with more invariance to amplitude scaling, similar to the non-negative Garrote (Gao, 2001). We consider here the cost function

$$C(\boldsymbol{a}) = \sum_k -\frac{a_k^2}{4\lambda} + \frac{a_k\sqrt{a_k^2+4\lambda^2}}{4\lambda} + \lambda \log\left(a_k + \sqrt{a_k^2+4\lambda^2}\right),$$

which is proportional to the one given by Figueiredo and Nowak (Figueiredo & Nowak, 2001) and is shown in Figure 1 for λ = 0.5.

Taking the derivative of this cost function, we end up with the relationship between u_k and a_k

$$u_k - a_k = -2\lambda\frac{a_k}{4\lambda} + \frac{2\lambda}{4\lambda}\sqrt{a_k^2+4\lambda^2}.$$

Solving for a_k as a function of u_k yields the following activation function,

$$a_k = T_\lambda(u_k) = \begin{cases} 0 & u_k \leq \lambda \\ (u_k^2 - \lambda^2)/u_k & u_k > \lambda \end{cases},$$

matching the results from Figueiredo and Nowak (Figueiredo & Nowak, 2001). This activation function is shown in Figure 1 for λ = 0.5.

3.3 Block ℓ_1

While all cost functions discussed earlier in this section have been separable, there is increasing interest in non-separable cost functions that capture structure (i.e., statistical

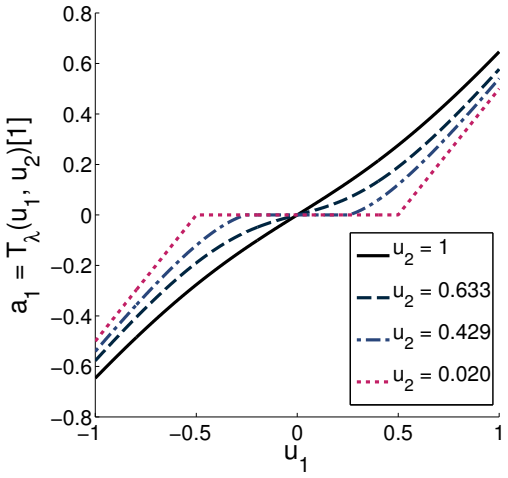

Figure 3: The nonlinear activation function used in the LCA to optimize the non-overlapping group LASSO cost function has multiple inputs and multiple outputs. The plot shows an example thresholding function for both elements in a group of size two ($\lambda = 0.5$), with each line illustrating the nonlinear effect on a_1 while u_2 is held constant.

dependencies) between the non-zero coefficients. For example, such structure would be important in performing inference in a complex cell energy model where the energies (i.e., magnitudes) are sparse in a complex-valued signal decomposition (e.g., (Cadieu & Olshausen, 2012)). Perhaps the most widely cited cost function discussed in this regard is the block ℓ_1 norm (also called the group ℓ_1 norm), which assumes that the coefficients representing $\boldsymbol{x}$ are active in known groups. In this framework, the coefficients are divided into blocks, $\mathcal{A}_l \subset \{a_k\}$ and each block of coefficients $\mathcal{A}_l$ is represented as a vector $\boldsymbol{a}^l$. For our purposes, we assume the blocks are non-overlapping but may have different cardinalities. The block ℓ_1 norm (Eldar et al., 2010) is defined as the ℓ_1 norm over the ℓ_2 norms of the groups,

$$\widetilde{C}\left(\boldsymbol{a}\right) = \sum_{l} \left\|\boldsymbol{a}^l\right\|_2,$$

essentially encouraging sparsity between the blocks (i.e., requiring only a few groups to be active) with no individual penalty on the coefficient values within a block. Because this cost is not separable, the activation function will no longer be a pointwise nonlinearity and will instead have multiple inputs and multiple outputs.

Following the same general approach as above, we calculate the gradient of the cost function for each block,

$$\nabla_{\boldsymbol{a}^l}\widetilde{C}\left(\boldsymbol{a}\right) = \frac{\boldsymbol{a}^l}{\left\|\boldsymbol{a}^l\right\|_2},$$

yielding the following relationship between the activation function inputs and outputs

$$\boldsymbol{u}^l = \boldsymbol{a}^l + \lambda \frac{\boldsymbol{a}^l}{\left\|\boldsymbol{a}^l\right\|_2}. \tag{7}$$

While directly solving this relationship for $\boldsymbol{a}^l$ appears difficult, we note that we can simplify the equation by expressing $\left\|\boldsymbol{a}^l\right\|_2$ in terms of $\left\|\boldsymbol{u}^l\right\|_2$. To see this, take the norm

of both sides of (7) to get $\left\|\boldsymbol{u}^l\right\|_2 = \left\|\boldsymbol{a}^l\right\|_2 + \lambda$. Substituting back into (7), the relationship simplifies to

$$\widetilde{T}_\lambda\left(\boldsymbol{u}^l\right) = \boldsymbol{a}^l = \boldsymbol{u}^l\left(1 - \frac{\lambda}{\left\|\boldsymbol{u}^l\right\|_2}\right)$$

over the range $0 \leq \left\|\boldsymbol{a}^l\right\|_2 = \left\|\boldsymbol{u}^l\right\|_2 - \lambda$, implying $\lambda \leq \left\|\boldsymbol{u}^l\right\|_2$.

This relationship yields the block-wise thresholding function

$$\boldsymbol{a}^l = \widetilde{T}_\lambda\left(\boldsymbol{u}^l\right) = \begin{cases} 0 & \left\|\boldsymbol{u}^l\right\|_2 \leq \lambda \\ \boldsymbol{u}^l\left(1 - \frac{\lambda}{\left\|\boldsymbol{u}^l\right\|_2}\right) & \left\|\boldsymbol{u}^l\right\|_2 > \lambda \end{cases}.$$

This activation function can be thought of as a type of shrinkage operation applied to an entire group of coefficients, with a threshold that depends on the norm of the group inputs. For the case of groups of two elements (with $\lambda = 0.5$), Figure 3 shows the nonlinearities for each of the two states as a function of the value of the other state.

3.4 Re-weighted ℓ_1 and ℓ_2

Recent work has also demonstrated that re-weighted ℓ_p norms can achieve better sparsity by iteratively solving a series of tractable convex programs (Wipf & Nagarajan, 2010; Chartrand & Yin, 2008; Candès et al., 2008; Garrigues & Olshausen, 2010). For example, re-weighted ℓ_1 (Candès et al., 2008) is an iterative algorithm where a single iteration consists of solving a weighted ℓ_1 minimization $\left(\widetilde{C}\left(\boldsymbol{a}\right) = \sum_k \lambda_k |a_k|\right)$, followed by a weight update according to the rule

$$\lambda_k \propto \frac{1}{|a_k| + \gamma}, \tag{8}$$

where γ is a small parameter. By having λ_k approximately equal to the inverse of the ℓ_1 norm of the coefficient from the previous iteration, this algorithm is more aggressive than BPDN at driving small coefficients to zero and increasing sparsity in the solutions. Similarly, re-weighted ℓ_2 algorithms (Wipf & Nagarajan, 2010) have also been used to approximate different p-norms with weights updated as

$$\lambda_k \propto \frac{1}{\left(a_k^2 + \gamma\right)^{\left(\frac{p}{2}-1\right)}}.$$

Such schemes have shown many empirical benefits over ℓ_p norm minimization, and recent work on re-weighted ℓ_1 has established theoretical performance guarantees (Khajehnejad et al., 2010) and interpretations as Bayesian inference in a probabilistic model (Garrigues & Olshausen, 2010).

One of the main drawbacks to re-weighted algorithms in digital architectures is the time required for solving the weighted ℓ_p program multiple times. Of course, it is also not clear that a discrete iterative approach such as this could be mapped to an asynchronous analog computational architecture. Because we have established earlier that the LCA architecture can solve the ℓ_p norm optimizations (and weighted norms are a

straightforward extension to those results), it would immediately follow that a dynamical system could be used to perform the optimization necessary for each iteration of the algorithm. While this would be a viable strategy, we show here that even more advantages can be gained by performing the entire re-weighted ℓ_1 algorithm in the context of a dynamical system. Specifically, we consider here a modified version of the LCA where an additional set of dynamics are placed on λ in order to simultaneously optimize the coefficients and coefficient weights in an analog system. While the ideas here are expandable to the general re-weighted case, we focus on results involving the re-weighted ℓ_1 as presented in (Garrigues & Olshausen, 2010).

The modified LCA is given by the system equations:

$$\begin{array}{c} \tau_u \dot{\boldsymbol{u}}(t) = \Phi^T \boldsymbol{x} - \boldsymbol{u}(t) - \left(\Phi^T \Phi - \boldsymbol{I}\right) \boldsymbol{a}(t) \\ \boldsymbol{a}(t) = T_{\boldsymbol{\lambda}}\left(\boldsymbol{u}(t)\right) \\ \tau_\lambda \dot{\lambda}_k(t) = \lambda_k^{-1}(t) - \nu^{-1}\left(|a_k(t)| + \gamma\right) \end{array} .$$

At steady state, $\dot{\boldsymbol{\lambda}} = 0$ which shows that λ_k (∞) abides by (8) with ν representing the proportionality constant. While the complete analysis of this expanded analog system is beyond the scope of this paper, we show in Figure 4a simulations which demonstrate that this system reaches a solution of comparable quality to digital iterative methods. Figure 4a plots the relative MSE from a compressed sensing recovery problem with length-1000 vectors from 500 noisy measurements with varying levels of sparsity. We sweep the parameter $\rho = S/M$ from zero to one and set the noise variance to 10^{-4}, with each plot representing the relative MSE averaged over 15 randomly chosen signals. Figure 4(a) plots the recovery quality for three systems: iterative re-weighted ℓ_1 (using GPSR (Figueiredo et al., 2007) to solve the ℓ_1 iterations), iterative re-weighted ℓ_1 (using the LCA to solve the ℓ_1 iterations), and dynamic re-weighted ℓ_1 which uses the modified LCA described above. It is clear that the three systems are achieving nearly the same quality in their signal recovery. Figure 4b plots the convergence of the recovery as a function of time (in terms of system time constants τ) for the iterative and dynamic re-weighted approaches using the LCA. The dynamically re-weighted system clearly converges more quickly, achieving its final solution in approximately the time it takes to perform two iterations of the traditional re-weighting scheme using the standard LCA.

4 Conclusions and future work

Sparsity-based signal models have played a significant role in many theories of neural coding across multiple sensory modalities. Despite the interest in the sparse coding hypothesis from the computational and theoretical neuroscience communities, the qualitative nature of much of the supporting evidence leaves significant ambiguity about the ideal form for a sparsity-inducing cost function. While recent trends favor the ℓ_1 norm due the emergence of guarantees in the signal processing literature, there are many sparsity-inducing signal models that may have benefits for neural computation and should be candidate models for neural coding. We have shown here that many of the sparsity-inducing cost functions proposed in the signal processing and statistics literatures can be implemented in a single unified dynamical system.

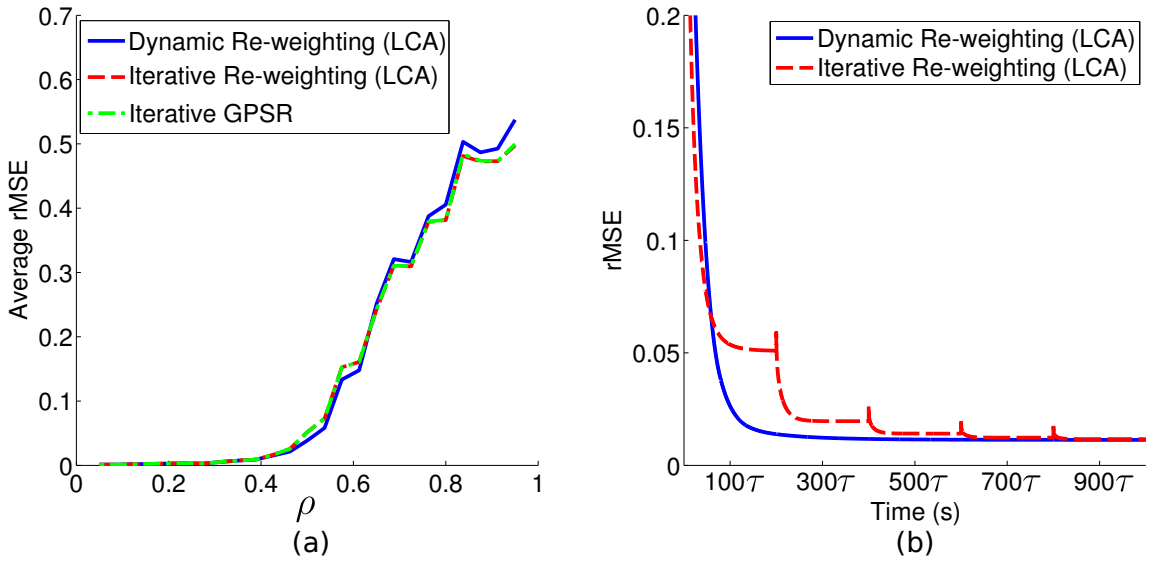

Figure 4: Re-weighted ℓ_1 optimization in digital algorithms and in a modified LCA. (a) Re-weighted ℓ_1 optimization for a signal with $N = 1000$ and $\delta = 0.5$, with ρ swept from 0 to 1. The traditional iterative re-weighting scheme is performed with both a standard digital algorithm (GPSR) and the LCA. For comparison, a dynamic re-weighting scheme where the LCA is modified to have continuous dynamics on the regularization parameter (rather than discrete iterations) is also shown. Each method is clearly achieving similar solutions. (b) The temporal evolution of the recovery relative MSE for a problem with $N = 1000$, $\delta = 0.6$ and $\rho = 0.45$. Solutions are shown for the amount of simulated time (in terms of number of time constants). The dynamically re-weighted system converges in approximately the time it takes to use the LCA to solve two iterations of the traditional re-weighted ℓ_1 algorithm.

From the results presented here, we conclude that neurally-plausible computational architectures can support a wide variety of sparsity-based signal models, and it is therefore reasonable to consider this broad family of models as reasonable candidates for theories of sensory neural coding. Furthermore, we have shown that even a relatively complex hierarchical probabilistic model resulting in a re-weighted ℓ_1 inference scheme can be implemented efficiently in a purely analog system. This observation is particularly interesting because it illustrates a fundamental potential advantage of analog computation over digital systems. Specifically, the analog approach to this problem is able to continuously infer two sets of variables jointly, rather than take an iterative approach that fundamentally must wait for the computations in each iteration for one variable to fully converge before inferring the other variable.

Beyond the applicability of these results to theories of neural computation, the recent shift toward optimization as a fundamental computational tool in the modern signal processing toolbox has made it difficult to implement many of these algorithms in applications with significant power constraints or real-time processing requirements. The results of this paper broaden the scope of problems that could potentially be approached through efficient neuromorphic architectures. The design and implementation of analog circuits has traditionally been difficult, but recent advances in reconfigurable analog cir-

cuits (Twigg & Hasler, 2009) have improved many of the issues related to the design of these systems. In fact, the reconfigurable platform described in (Twigg & Hasler, 2009) has been used to implement a small version of the LCA for solving BPDN (Shapero et al., 2012a,b), and preliminary tests of this implementation are consistent with simulations of the idealized LCA. These results lend encouragement to the idea that efficient analog circuits could be implemented for the variety of cost functions described in this paper.

Acknowledgments

The authors are grateful to B. Olshausen and J. Romberg for valuable discussions related to this work.

Appendix

Soft-threshold activation for BPDN using the log-barrier relaxation

We will first rewrite the desired BPDN problem (Equation (2) with the ℓ_1 cost funtion) in an extended formulation to make the variables non-negative. Define a new $M \times 2N$ matrix through the concatenation operation $\widetilde{\Phi} = [\Phi \ \ -\Phi]$. Similarly define a vector $\boldsymbol{z} = [\boldsymbol{z}_+ \ \boldsymbol{z}_-]$ of length $2N$ such that $z_i \geq 0$ and $\boldsymbol{a} = \boldsymbol{z}_+ - \boldsymbol{z}_-$. Essentially $\boldsymbol{z}$ represents the original variables $\boldsymbol{a}$ by separating them into two subvectors depending on their sign. We can then write a constrained optimization program that is equivalent to BPDN:

$$\min_{\boldsymbol{z}} \frac{1}{2} \left|\left| \boldsymbol{x} - \widetilde{\Phi}\boldsymbol{z} \right|\right|_2^2 + \lambda \sum_{k=1}^{2N} z_k \qquad \text{s.t.} \quad z_k \geq 0. \tag{9}$$

This reformulation is a standard way to show that ℓ^1 cost penalties are equivalent to a linear function in a constrained optimization program. One can then apply the standard log-barrier relaxation to convert the program in (9) to an approximately equivalent unconstrained program:

$$\min_{\boldsymbol{z}} \frac{1}{2} \left|\left| \boldsymbol{x} - \widetilde{\Phi}\boldsymbol{z} \right|\right|_2^2 + \lambda \sum_{k=1}^{2N} z_k + \left(\frac{1}{\gamma}\right) \sum_{k=1}^{2N} \log(z_k). \tag{10}$$

As $\gamma \to \infty$, this program approaches the desired program (9). This relaxation strategy underlies an interior point algorithm (called the barrier method) for solving convex optimization programs, where (10) is repeatedly solved with increasing values of γ (Boyd & Vandenberghe, 2004).

Note that the relaxed problem in (10) fits the form of the general optimization program stated in (2) with the differentiable cost function $C(z_k) = z_k - \frac{\log(z_k)}{\gamma\lambda}$. For a fixed value of γ, this cost function can be differentiated and used in the relationship given in (5) to solve for z_k in terms of u_k to find the corresponding invertible activation function:

$$z_k = T_\lambda(u_k) = \frac{1}{2}\left(\sqrt{\frac{4 + \gamma(\lambda - u_k)^2}{\gamma}} - (\lambda - u_k)\right).$$

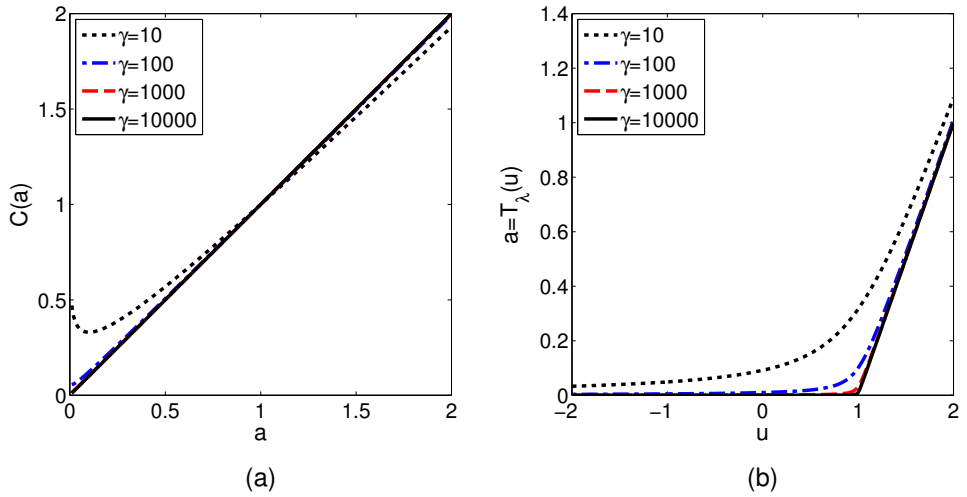

Figure 5: Log barrier relaxations of BPDN. (a) The cost function approaches the ideal ℓ^1 norm as the relaxation parameter is increased. (b) In a similar way, the nonlinear activation function derived for the LCA approaches the ideal soft-thresholding operator as the relaxation parameter is increased.

Finally it is straightforward to show that in the relaxation limit ($\gamma \to \infty$) where the program in (10) approaches BPDN, the desired activation function becomes the soft-thresholding function:

$$\lim_{\gamma\to\infty} \frac{1}{2}\left(\sqrt{\frac{4+\gamma(\lambda-u_k)^2}{\gamma}} - (\lambda - u_k)\right) = \frac{1}{2}\left(\sqrt{(\lambda-u_k)^2} - (\lambda-u_k)\right)$$
$$= \begin{cases} 0 & \text{when } u_k \leq \lambda \\ u_k - \lambda & \text{when } u_k > \lambda \end{cases}.$$

To illustrate the convergence of this relaxation to the desired ℓ^1 cost function and the corresponding soft-threshold activation function, Figure 5 plots $C(\cdot)$ and $T_\lambda(\cdot)$ in this relaxed problem for several values of γ. Note that in the extended formulation of BPDN given in (9), the variables occur in pairs where where only one of them can be nonzero at a time. Because the activation function is zero for all state values with magnitude less than threshold, it is possible to represent each of these pairs of variables in one LCA node that can take on positive and negative values and where the activation function is a two-sided soft-thresholding function (thereby reducing the number of nodes back down to N).

References

Antoniadis, A. & Fan, J. (2001). Regularization of wavelet approximations. Journal of the American Statistical Association, 96, 939–967.

Balavoine, A., Romberg, J., & Rozell, C. (2011). Convergence and rate analysis of neural networks for sparse approximation. In Press.

Battaglia, P., Jacobs, R., & Aslin, R. (2003). Bayesian integration of visual and auditory signals for spatial localization. JOSA A, 20, 1391–1397.

Baum, E., Moody, J., & Wilczek, F. (1988). Internal representations for associative memory. Biological Cybernetics, 59, 217–228.

Boyd, S. & Vandenberghe, L. (2004). Convex Optimization. (Cambridge University Press).

Cadieu, C. F. & Olshausen, B. A. (2012). Learning intermediate-level representations of form and motion from natural movies. Neural Computation, 24, 827–866.

Candès, E., Wakin, M., & Boyd, S. (2008). Enhancing sparsity by reweighted ℓ_1 minimization. Journal of Fourier Analysis and Applications, 14, 877–905.

Charles, A. S., Yap, H. L., & Rozell, C. J. (2012). Short-term memory capacity in recurrent networks via compressed sensing. In Computational and Systems Neuroscience (Cosyne) Meeting.

Chartrand, R. & Yin, W. (2008). Iteratively reweighted algorithms for compressive sensing. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 3869–3872.

Coen-Cagli, R., Dayan, P., & Schwartz, O. (2012). Cortical surround interactions and perceptual salience via natural scene statistics. PLoS Comput Biol, 8, e1002405.

Doya, K. (2007). Bayesian brain: Probabilistic approaches to neural coding. (The MIT Press).

Elad, M., Figueiredo, M., & Ma, Y. (2010). On the role of sparse and redundant representations in image processing. Proceedings of the IEEE, 98, 972–982.

Elad, M., Matalon, B., & Zibulevsky, M. (2007). Coordinate and subspace optimization methods for linear least squares with non-quadratic regularization. Applied and Computational Harmonic Analysis, 23, 346–367.

Eldar, Y. C., Kuppinger, P., & Bolcskei, H. (2010). Block-sparse signals: Uncertainty relationships and efficient recovery. IEEE Transactions on Signal Processing, 58, 3042–3054.

Fan, J. (1997). Comments on 'Wavelets in statistics: A review' by A. Antoniadis. Statistical Methods and Applications, 6, 131–138.

Figueiredo, M. A. T. & Nowak, R. D. (2001). Wavelet-based image estimation: An empirical Bayes approach using Jeffrey's noninformative prior. IEEE Transactions on Image Processing, 10, 1322–1331.

Figueiredo, M. A. T., Nowak, R. D., & Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. IEEE Journal of Selected Topics in Signal Processing.

Gao, H. (2001). Wavelet shrinkage denoising using the non-negative Garrote. Journal of Computational and Graphical Statistics, 7, 469–488.

Garrigues, P. & Olshausen, B. (2010). Group sparse coding with a laplacian scale mixture prior. Advances in Neural Information Processing Systems, pp. 1–9.

Haider, B., Krause, M., Duque, A., Yu, Y., Touryan, J., Mazer, J., & McCormick, D. (2010). Synaptic and Network Mechanisms of Sparse and Reliable Visual Cortical Activity during Nonclassical Receptive Field Stimulation. Neuron, 65, 107–121.

Hopfield, J. (1982). Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79, 2554.

Hu, T., Genkin, A., & Chklovskii, D. B. (2012). A network of spiking neurons for computing sparse representations in an energy efficient way. In Press.

Huber, P. J. (1973). Robust regression: Asymptotics, conjectures and Monte Carlo. The Annals of Statistics, 1, 799–821.

Hürlimann, F., Kiper, D., & Carandini, M. (2002). Testing the bayesian model of perceived speed. Vision research, 42, 2253–2257.

Karklin, Y. & Lewicki, M. (2008). Emergence of complex cell properties by learning to generalize in natural scenes. Nature, 457, 83–86.

Khajehnejad, M., Xu, W., Avestimehr, S., & Hassibi, B. (2010). Improved sparse recovery thresholds with two-step reweighted ℓ_1 minimization. Arxiv preprint arXiv:1004.0402.

Lennie, P. (2003). The cost of cortical computation. Current biology, 13, 493–497.

Nikolova, M. (2000). Local strong homogeneity of a regularized estimator. SIAM Journal on Applied Mathematics, 61, 633–658.

Olshausen, B. & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature, 381, 607.

Olshausen, B. & Field, D. (2004). Sparse coding of sensory inputs. Current opinion in neurobiology, 14, 481–487.

Olshausen, B. A. & Field, D. (1997). Sparse coding with an overcomplete basis set: A strategy employed by v1? Vision Research, 37, 3311–3325.

Perrinet, L., Samuelides, M., & Thorpe, S. (2004). Sparse spike coding in an asynchronous feed-forward multi-layer neural network using matching pursuit. Neurocomputing, 57, 125 – 134.

Rao, R. & Ballard, D. (1999). Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. Nature Neuroscience, 2, 79–87.

Rehn, M. & Sommer, F. T. (2007). A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. Journal of Computational Neuroscience, 22, 135–146.

Rozell, C. & Garrigues, P. (2010). Analog sparse approximation for compressed sensing recovery. In Proceedings of the 2010 ASILOMAR Conference on Signals, Systems and Computers, pp. 822–826.

Rozell, C. J., Johnson, D. H., Baraniuk, R. G., & Olshausen, B. A. (2010). Sparse coding via thresholding and local competition in neural circuits. Neural Computation, 20, 2526–2563.

Saab, R., Chartrand, R., & Yilmaz, O. (2008). Stable sparse approximations via nonconvex optimization. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal, pp. 3885–3888.

Schwartz, O. & Simoncelli, E. (2001). Natural signal statistics and sensory gain control. Nature neuroscience, 4, 819–825.

Seriès, P., Lorenceau, J., & Frégnac, Y. (2003). The silent surround of v1 receptive fields: theory and experiments. Journal of physiology-Paris, 97, 453–474.

Shapero, S., Charles, A., Rozell, C. J., & Hasler, P. (2012a). Low power sparse approximation on reconfigurable analog hardware. Submitted.

Shapero, S., Rozell, C. J., & Hasler, P. (2012b). Configurable hardware integrate and fire neurons for sparse approximation. Submitted.

Spratling, M. (2011). A single functional model accounts for the distinct properties of suppression in cortical area v1. Vision Research, 51, 563 – 576.

Tikhonov, A. (1963). Regularization of incorrectly posed problems. In Soviet Math. Dokl, vol. 4, pp. 1624–1627.

Twigg, C. & Hasler, P. (2009). Configurable analog signal processing. Digital Signal Processing, 19, 904–922.

Vinje, W. & Gallant, J. (2002). Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. Journal of Neuroscience, 22, 2904.

Wipf, D. & Nagarajan, S. (2010). Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. IEEE Journal of Selected Topics in Signal Processing, 4, 317–329.

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T., & Yan, S. (2010). Sparse representation for computer vision and pattern recognition. Proceedings of the IEEE, 98, 1031–1044.

Zhu, M. & Rozell, C. J. (2012). Visual nonclassical receptive field effects emerge from sparse coding in a dynamical system. Submitted.

Zou, H. (2006). The adaptive lasso and its oracle properties. Journal of the American Statistical Association, 101, 1418–1429.

Zylberberg, J., Murphy, J. T., & DeWeese, M. R. (2011). A sparse coding model with synaptically local plasticity and spiking neurons can account for the diverse shapes of v1 simple cell receptive fields. PLoS Comput Biol, 7, e1002250.