Signal recovery from Pooling Representations

Joan Bruna BRUNA@CIMS.NYU.EDU
Courant Institute of Mathematical Sciences, 715 Broadway New York NY 10003 USA

Arthur Szlam ASZLAM@CCNY.CUNY.EDU
Yann LeCun YANN@CS.NYU.EDU
Courant Institute of Mathematical Sciences, 715 Broadway New York NY 10003 USA

Abstract

In this work we compute lower Lipschitz bounds of ℓ_p pooling operators for $p = 1, 2, \infty$ as well as ℓ_p pooling operators preceded by half-rectification layers. These give sufficient conditions for the design of invertible neural network layers. Numerical experiments on MNIST and image patches confirm that pooling layers can be inverted with phase recovery algorithms. Moreover, the regularity of the inverse pooling, controlled by the lower Lipschitz constant, is empirically verified with a nearest neighbor regression.

1. Introduction

A standard architecture for deep feedforward networks consists of a number of stacked modules, each of which consists of a linear mapping, followed by an elementwise nonlinearity, followed by a pooling operation. Critical to the success of this architecture in recognition problems is its capacity for preserving discriminative signal information while being invariant to nuisance deformations. The recent works (Mallat, 2012; Bruna and Mallat, 2012) study the role of the pooling operator in building invariance. In this work, we will study a network's capacity for preserving information. Specifically, we will study the invertibility of modules with a linear mapping, the half rectification nonlinearity, and ℓ_p pooling, for $p \in \{1, 2, \infty\}$. We will discuss recent work in the case $p = 2$, and connections with the phase recovery problem of (Candes et al., 2013; Gerchberg and Saxton, 1972; Waldspurger et al., 2012).

1.1. ℓ_p pooling

The purpose of the pooling layer in each module is to give invariance to the system, perhaps at the expense of resolution. This is done via a summary statistic over the outputs of groups of nodes. In the trained system, the columns of the weight matrix corresponding to nodes grouped together often exhibit similar characteristics, and code for perturbations of a template (Kavukcuoglu et al., 2009; Hyvärinen and Hoyer, 2001).

The summary statistic in ℓ_p pooling is the ℓ_p norm of the inputs into the pool. That is, if nodes $x_{I_i}, ..., x_{I_l}$ are in a pool, the output of the pool is

$$(|x_{I_i}|^p + ... + |x_{I_l}|^p)^{1/p},$$

where as usual, if $p \to \infty$, this is

$$\max(|x_{I_i}|, ..., |x_{I_l}|).$$

If the outputs of the nonlinearity are nonnegative (as for the half rectification function), then $p = 1$ corresponds to average pooling, and the case $p = \infty$ is max pooling.

1.2. Phase reconstruction

Given $x \in \mathbb{R}^n$, a classical problem in signal processing is to recover x from the absolute values of its (1 or 2 dimensional) Fourier coefficients, perhaps subject to some additional constraints on x; this problem arises in speech generation and X-ray imaging (Ohlsson, 2013). Unfortunately, the problem is not well posed- the absolute values of the Fourier coefficients do not nearly specify x. For example, the absolute value of the Fourier transform is translation invariant. It can be shown (and we discuss this below) that the absolute value of the inner products between x and any basis of $\mathbb{R}^n$ are not enough to uniquely specify an arbitrary x; the situation is worse for $\mathbb{C}^n$. On the other hand, recent works have shown that by taking a redundant enough dictionary, the situation is different, and x can be recovered from the modulus of its inner products with the dictionary

(Balan et al., 2006; Candes et al., 2013; Waldspurger et al., 2012).

Suppose for a moment that there is no elementwise nonlinearity in our feedforward module, and only a linear mapping followed by a pooling. Then with a slightly generalized notion of phase, where the modulus is the ℓ_p norm of the pool, and the phase is the ℓ_p unit vector specifying the "direction" of the inner products in the pool, the phase recovery problem above asks if the module loses any information. The ℓ_2 case has been recently studied in (Cahill et al., 2013)

1.3. What vs. Where

If the columns of the weight matrix in a pool correspond to related features, it can be reasonable to see the entire pool as a "what". That is, the modulus of the pool indicates the relative presence of a grouping of (sub)features into a template, and the phase of the pool describes the relative arrangement of the subfeatures, describing "where" the template is, or more generally, describing the "pose" of the template.

From this viewpoint, phase reconstruction results make rigorous the notion that given enough redundant versions of "what" and throwing away the "where", we can still recover the "where".

1.4. Contributions of this work

In this work we give conditions so that a module consisting of a linear mapping, perhaps followed by a half rectification, followed by an ℓ_p pooling preserves the information in its input. We extend the ℓ_2 results of (Cahill et al., 2013; Balan and Wang, 2013) in several ways: we consider the ℓ_p case, take into account the half rectification nonlinearity, and we make the results quantitative in the sense that we give bounds on the lower Lipschitz constants of the modules. This gives a measure of the *stability* of the inversion, which is especially important in a multi-layer system. Using our bounds, we prove that redundant enough random modules with ℓ_1 or ℓ_∞ pooling are invertible.

We also show the results of numerical experiments designed to explore the gaps in our results and the results in the literature. We note that the alternating minimization method of (Gerchberg and Saxton, 1972) can be used essentially unchanged for the ℓ_p case, with or without rectification, and show experiments giving evidence that recovery is roughly equally possible for ℓ_1, ℓ_2, and ℓ_∞ using this algorithm; and that half rectification before pooling can make recovery easier. Furthermore, we show that with a trained initialization, the alternating method compares favorably with the state of the art recovery methods (for ℓ_2 with no rectification) in (Waldspurger et al., 2012; Candes et al., 2013), which suggests that the above observations are not an artifcact of the alternating method.

2. Injectivity and Lipschitz stability of Pooling Operators

This section studies necessary and sufficient conditions guaranteeing that pooling representations are invertible. It also computes upper and lower Lipschitz bounds, which are tight under certain configurations.

Let us first introduce the notation used throughout the paper. Let $\mathcal{F} = \{f_1, \dots, f_M\}$ be a real frame of $\mathbb{R}^N$, with $M > N$. The frame $\mathcal{F}$ is organized into K disjoint blocks $\mathcal{F}_k = \{f_j\}_{j \in I_k}$, $k = 1 \dots K$, such that $I_k \cap I_{k'} = \emptyset$ and $\bigcup_k I_k = \{1 \dots M\}$. For simplicity, we shall assume that all the pools have equal size $|I_k| = L$.

The ℓ_p pooling operator $P_p(x)$ is defined as the mapping

$$x \mapsto P_p(x) = \{\|\mathcal{F}_k^T x\|_p \ , \ k = 1 \dots K\} \ . \qquad (1)$$

A related representation, which has gained popularity in recent deep learning architectures, introduces a point-wise thresholding before computing the ℓ_p norm. If $\alpha \in \mathbb{R}^M$ is a fixed threshold vector, and $(\rho_\alpha(x))_i = \max(0, x_i - \alpha_i)$, then the ℓ_p *rectified pooling* operator $R_p(x)$ is defined as

$$x \mapsto R_p(x) = \{\|\rho_{\alpha_k}(\mathcal{F}_k^T x)\|_p \ , \ k = 1 \dots K\} \ , \qquad (2)$$

where α_k contains the coordinates I_k of α.

We shall measure the stability of the inverse pooling with the Euclidean distance in the representation space. Given a distance $d(x, x')$ in the input space, the Lipschitz bounds of a given operator $\Phi(x)$ are defined as the constants $0 \leq A \leq B$ such that

$$\forall \ x \ , \ x' \ , \ Ad(x, x') \leq \|\Phi(x) - \Phi(x')\|_2 \leq Bd(x, x') \ .$$

In the remainder of the paper, given a frame $\mathcal{F}$, we denote respectively by $\lambda_-(\mathcal{F})$ and $\lambda_+(\mathcal{F})$ its lower and upper frame bounds. If $\mathcal{F}$ has M vectors and $\Omega \subset \{1, \dots, M\}$, we denote $\mathcal{F}_\Omega$ the frame obtained by keeping the vectors indexed in Ω. Finally, we denote Ω^c the complement of Ω.

2.1. Absolute value and Thresholding nonlinearities

In order to study the injectivity of pooling representations, we first focus on the properties of the operators defined by the point-wise nonlinearities.

The properties of the *phaseless* mapping

$$x \mapsto M(x) = \{|\langle x, f_i \rangle| \ , \ i = 1 \dots m\} \ , \ x \in \mathbb{R}^n \ , \qquad (3)$$

have been extensively studied in the literature (Balan et al., 2006; Balan and Wang, 2013), in part motivated by applications to speech processing (Achan et al., 2003) or

X-ray crystallography (Ohlsson, 2013). It is shown in (Balan et al., 2006) that if $m > 2n - 1$ then it is possible to recover x from $M(x)$, up to a global sign change. In particular, (Balan and Wang, 2013) recently characterized the stability of the phaseless operator, that is summarized in the following proposition:

Proposition 2.1 ((Balan and Wang, 2013), Theorem 4.3) *Let $\mathcal{F} = (f_1, \ldots, f_M)$ with $f_i \in \mathbb{R}^N$ and $d(x, x') = \min(\|x - x'\|, \|x + x'\|)$. The mapping $M(x) = \{|\langle x, f_i\rangle|\}_{i \leq m}$ satisfies*

$$\forall\, x, x' \in \mathbb{R}^n \,,\ A_{\mathcal{F}}\, d(x, x') \leq \|M(x) - M(x')\| \leq B_{\mathcal{F}}\, d(x, x') \,, \tag{4}$$

where

$$A_{\mathcal{F}} = \min_{\Omega \subset \{1 \ldots M\}} \sqrt{\lambda_-^2(\mathcal{F}_\Omega) + \lambda_-^2(\mathcal{F}_{\Omega^c})} \,, \tag{5}$$

$$B_{\mathcal{F}} = \lambda_+(\mathcal{F}) \,. \tag{6}$$

In particular, $M(x)$ is injective if and only if for any subset $\Omega \subseteq \{1, \ldots, M\}$, either $\mathcal{F}_\Omega$ or $\mathcal{F}_{\Omega^c}$ is an invertible frame.

A frame $\mathcal{F}$ satisfying the previous condition is said to be *phase retrievable*.

We now turn our attention to the half-rectification operator, defined as

$$M_\alpha(x) = \rho_\alpha(\mathcal{F}^T x) \,. \tag{7}$$

For that purpose, let us introduce some extra notation. Given a frame $\mathcal{F} = \{f_1, \ldots, f_M\}$, a subset $\Omega \subset \{1 \ldots M\}$ is *admissible* if

$$\bigcap_{i \in \Omega} \{x \,;\, \langle x, f_i\rangle > \alpha_i\} \cap \bigcap_{i \notin \Omega} \{x \,;\, \langle x, f_i\rangle < \alpha_i\} \neq \emptyset \,. \tag{8}$$

We denote by $\overline{\Omega}$ the collection of all admissible sets, and V_Ω the vector space generated by Ω. The following proposition, proved in Section B, gives a necessary and sufficient condition for the injectivity of the half-rectification.

Proposition 2.2 *Let $A_0 = \min_{\Omega \in \overline{\Omega}} \lambda_-(\mathcal{F}_\Omega\big|_{V_\Omega})$. Then the half-rectification operator $M_\alpha(x) = \rho_\alpha(\mathcal{F}^T x)$ is injective if and only if $A_0 > 0$. Moreover, it satisfies*

$$\forall\, x, x' \,,\ A_0\|x - x'\| \leq \|M_\alpha(x) - M_\alpha(x')\| \leq B_0\|x - x'\| \,, \tag{9}$$

with $B_0 = \max_{\Omega \in \overline{\Omega}} \lambda_+(\mathcal{F}_\Omega) \leq \lambda_+(\mathcal{F})$.

The half-rectification has the ability to recover the input signal, without the global sign ambiguity. The ability to reconstruct from M_α is thus controlled by the rank of any matrix $\mathcal{F}_\Omega$ whose columns are taken from a subset belonging to $\overline{\Omega}$. In particular, if $\alpha \equiv 0$, since $\Omega \in \overline{\Omega} \Rightarrow \Omega^c \in \overline{\Omega}$, it results that $m \geq 2n$ is necessary in order to have $A_0 > 0$.

The rectified linear operator creates a partition of the input space into polytopes, defined by (8), and computes a linear operator on each of these regions. A given input x activates a set $\Omega_x \in \overline{\Omega}$, encoded by the sign of the linear measurements $\langle x, f_i\rangle - \alpha_i$.

As opposed to the absolute value operator, the inverse of M_α, whenever it exists, can be computed directly by locally inverting a linear operator. Indeed, the coordinates of $M_\alpha(x)$ satisfying $M_\alpha(x)_j > \alpha_j$ form a set $s(x)$, which defines a linear model $\mathcal{F}_{s(x)}$ which is invertible if $A_0 > 0$.

In order to compare the stability of the half-rectification versus the full rectification, one can modify M_α so that it maps x and $-x$ to the same point. If one considers

$$\widetilde{M}_\alpha(x) = \begin{cases} M_\alpha(x) & \text{if } \lambda_-(\mathcal{F}_{s(x)}) > \lambda_-(\mathcal{F}_{s(x)^c}) \,, \\ M_{-\alpha}(-x) & \text{otherwise} \,. \end{cases}$$

then $\tilde{M}_\alpha$ satisfies the following:

Corollary 2.3

$$\forall\, x, x' \in \mathbb{R}^n \,,\ \widetilde{A}\, d(x, x') \leq \|\widetilde{M}_\alpha(x) - \widetilde{M}_\alpha(x')\| \leq \widetilde{B}\, d(x, x') \,, \tag{10}$$

with

$$\widetilde{A} = \min_{\Omega \subset \overline{\Omega}} \max(\lambda_-^2(\mathcal{F}_\Omega),\, \lambda_-^2(\mathcal{F}_{\Omega^c})) \,, \tag{11}$$

$$\widetilde{B} = \max_{\Omega \subset \overline{\Omega}} \lambda_+(\mathcal{F}_\Omega) \leq \lambda_+(\mathcal{F}) \,, \tag{12}$$

and $d(x, x') = \min(x - x', x + x')$, so $\widetilde{A} \geq 2^{-1/2} A$ and $\widetilde{B} \leq B$. In particular, if M is invertible, so is $\tilde{M}_\alpha$.

It results that the bi-Lipschitz bounds of the half-rectification operator, when considered in under the equivalence $x \sim -x$, are controlled by the bounds of the absolute value operator, up to a factor $2^{-1/2}$. However, the lower Lipschitz bound (11) consists in a minimum taken over a much smaller family of elements than in (5).

2.2. ℓ_p Pooling

We give bi-Lipschitz constants of the ℓ_p Pooling and ℓ_p rectified Pooling operators for $p = 1, 2, \infty$.

From its definition, it follows that pooling operators P_p and R_p can be expressed respectively as a function of phaseless and half-rectified operators, which implies that for the pooling to be invertible, it is necessary that the absolute value and rectified operators are invertible too. Naturally, the converse is not true.

2.2.1. ℓ_2 POOLING

The invertibility conditions of the ℓ_2 pooling representation have been recently studied in (Cahill et al., 2013), where the authors obtain necessary and sufficient conditions on the frame $\mathcal{F}$. We shall now generalize those results, and derive bi-Lipschiz bounds.

Let us define

$$\mathcal{Q}_2 = \left\{(U_k \mathcal{F}_k)_{k\leq K} \,;\, \forall\, k \leq K \;,\; U_k \in \mathbb{R}^{d\times d} \;,\; U_k^T U_k = \mathbf{Id}\right\} . \tag{13}$$

$\mathcal{Q}_2$ thus contains all the orthogonal bases of each subspace $\mathcal{F}_k$.

The following proposition, proved in section B, computes upper and lower bounds of the Lipschitz constants of P_2.

Proposition 2.4 *The ℓ_2 pooling operator P_2 satisfies*

$$\forall\, x,\, x';\;,\; A_2 d(x,x') \leq \|P_2(x) - P_2(x')\| \leq B_2 d(x,x') \;, \tag{14}$$

where

$$\begin{aligned} A_2 &= \min_{\mathcal{F}'\in\mathcal{Q}_2} \min_{\Omega\subset\{1\dots M\}} \sqrt{\lambda_-^2(\mathcal{F}'_\Omega) + \lambda_-^2(\mathcal{F}'_{\Omega^c})} \;, \\ B_2 &= \lambda_+(\mathcal{F}) \,. \end{aligned} \tag{15}$$

This proposition thus generalizes the results from (Cahill et al., 2013), since it shows that $A_2 > 0$ not only controls when P_2 is invertible, but also controls the stability of the inverse mapping.

We also consider the rectified ℓ_2 pooling case. For simplicity, we shall concentrate in the case where the pools have dimension $d = 2$. For that purpose, for each x, x', we consider a modification of the families $\mathcal{Q}_2$, by replacing each sub-frame $\mathcal{F}_k$ by $\mathcal{F}_{I_k\cap s(x)\cap s(x')}$, that we denote $\widetilde{\mathcal{Q}}_{2,x,x'}$.

Corollary 2.5 *Let $d = 2$, and set $p(x,x') = s(x) \cup s(x')\backslash(s(x)\cap s(x'))$. Then the rectified ℓ_2 pooling operator R_2 satisfies*

$$\forall\, x,\, x';\;,\; \tilde{A}_2 d(x,x') \leq \|R_2(x) - R_2(x')\| \leq B_2 d(x,x') \;, \tag{16}$$

where

$$\begin{aligned} \tilde{A}_2 &= \inf_{x,x'} \min_{\mathcal{F}'\in\widetilde{\mathcal{Q}}_{2,x,x'}} \min_{\Omega\subset s(x)\cap s(x')} \Big(\lambda_-^2(\mathcal{F}_{p(x,x')}) + \\ &\quad \lambda_-^2(\mathcal{F}'_\Omega) + \lambda_-^2(\mathcal{F}'_{\Omega^c})\Big)^{1/2} \;, \end{aligned}$$

Proposition 2.4 and Corollary 2.5 give a lower Lipschitz bound which gives sufficient guarantees for the inversion of pooling representations. Corollary 2.5 indicates that, in the case $d = 2$, the lower Lipschitz bounds are sharper than the non-rectified case, in consistency with the results of section 2.1. The general case $d > 2$ remains an open issue.

2.2.2. ℓ_∞ Pooling

We give in this section sufficient and necessary conditions such that the max-pooling operator P_∞ is injective, and we compute a lower bound of its lower Lipschitz constant.

Given $x \in \mathbb{R}^N$, we define the *switches* $s(x)$ of x as the K vector of coordinates in each pool where the maximum is attained; that is, for each $k \in \{1,\dots,K\}$:

$$s(x)_k = \arg\max_{j\in I_k} |\langle x, f_j\rangle| ,$$

and we denote by $\mathcal{S}$ the set of all attained switches: $\mathcal{S} = \{s(x)\,;\, x \in \mathbb{R}^N\}$. This is a discrete subset of $\prod_k\{1,\dots,I_k\}$. Given $s\in\mathcal{S}$, the set of input signals having $s(x) = s$ defines a linear cone $\mathcal{C}_s \subset \mathbb{R}^N$:

$$\mathcal{C}_s = \bigcap_{k\leq K}\bigcap_{j\in I_k} \{x\,;\, |\langle x, f_{s_k}\rangle| \geq |\langle x, f_j\rangle|\} \;,$$

and as a result the input space is divided into a collection of Voronoi cells defined from linear equations. Restricted to each cone $\mathcal{C}_s$, the max-pooling operator computes the phaseless mapping $M(x)$ from equation (3) corresponding to $\mathcal{F}_s = (f_{s_1},\dots,f_{s_K})$.

Given vectors u and v, as usual, set the angle $\theta(u,v) = \arccos\left(\frac{|\langle u,v\rangle|}{\|u\|\|v\|}\right)$. For each $s, s'\in\mathcal{S}$ such that $\mathcal{C}_s\cap\mathcal{C}_{s'} = \emptyset$ and for each $\Omega\subset\{1\dots K\}$, we define

$$\beta(s,s',\Omega) = \min_{u\in\mathcal{F}_s\big|_\Omega(\mathcal{C}_s)\; v\in\mathcal{F}_{s'}\big|_\Omega(\mathcal{C}_{s'})} \theta(u,v).$$

This is a modified first principal angle between the subspaces $\mathcal{F}_s\big|_\Omega$ and $\mathcal{F}_{s'}\big|_\Omega$, where the infimum is taken only on the directions included in the respective cones. Set $\Lambda_{s,s',\Omega} = \lambda_-(\mathcal{F}\big|_\Omega)\cdot\sin(\beta(s,s',\Omega))$.

Given s, s', we also define $\mathcal{J}(s,s') = \{k\,;\, s_k = s'_k\}$. Recall L is the size of each pool. Set

$$\begin{aligned} A(s,s') &= \Big\{\min_{\Omega\subseteq\mathcal{J}(s,s')} \lambda_-^2(\mathcal{F}_\Omega) + \lambda_-^2(\mathcal{F}_{\mathcal{J}-\Omega}) + \\ &\quad \frac{1}{4L}\min_{\Omega\subseteq\mathcal{J}(s,s')^c} \Lambda^2_{s,s',\Omega} + \Lambda^2_{s,s',\Omega^c}\Big\}^{1/2} . \end{aligned}$$

The following proposition, proved in section B, gives a lower Lipschitz bound of the max-pooling operator.

Proposition 2.6 *For all x and x', the max-pooling operator P_∞ satisfies*

$$d(x,x')\left(\min_{s,s'} A(s,s')\right) \leq \|P_\infty(x) - P_\infty(x')\|, \tag{17}$$

where $d(x,x') = \min(\|x-x'\|, \|x+x'\|)$.

Propostion 2.6 shows that the lower Lipschitz bound is controlled by two different phenomena. The first one depends upon how the cones corresponding to disjoint switches are aligned, whereas the second one depends on the internal incoherence of each frame $\mathcal{F}_{\mathcal{J}(s,s')}$. One may ask how do these constants evolve in different asymptotic regimes. For example, if one lets the number of pools K be fixed but increases the size of each pool by increasing M. In that case, the set of possible switches $\mathcal{S}$ increases, and each cone $\mathcal{C}_s$ gets smaller. The bound corresponding to $\mathcal{C}_s \cap \mathcal{C}_{s'} \neq \emptyset$ decreases since the infimum is taken over a larger family. However, as the cones $\mathcal{C}_s$ become smaller, the likelihood that any pair $x \neq x'$ share the same switches decreases, thus giving more importance to the case $\mathcal{C}_s \cap \mathcal{C}_{s'} = \emptyset$. Although the ratio $\frac{1}{L}$ decreases, the lower frame bounds $\lambda_-(\mathcal{F}_\Omega)^2$, $\lambda_-(\mathcal{F}_{\Omega^c})^2$ will in general increase linearly with L. The lower bound will thus mainly be driven by the principal angles $\beta(s, s', \Omega)$. Although the minimum in (28) is taken over a larger family, each angle is computed over a smaller region of the space, suggesting that one can indeed increase the size of each pool without compromising the injectivity of the max-pooling.

Another asymptotic regime considers pools of fixed size L and increases the number of pools K. In that case, the bound increases as long as the principal angles remain lower bounded.

We also consider the stability of max-pooling with a half-rectification. By redefining the switches $s(x)$ accordingly:

$$s(x) = \{j\,;\, \langle x, f_j\rangle + \alpha_j > \max(0, \langle x, f_{j'}\rangle + \alpha_{j'}\,;\, \forall\, j' \in \mathrm{pool}(j)\}\} \tag{18}$$

the following proposition, proved in section B, computes a lower bound of the Lipschitz constant of R_∞.

Corollary 2.7 *The rectified max-pooling operator R_∞ satisfies*

$$\forall\, x, x'\,, \|x - x'\| \min_{s,s'} A(s,s') \leq \|R_\infty(x) - R_\infty(x')\|\,, \tag{19}$$

with

$$A(s,s') = \left\{\lambda_-^2(\mathcal{F}_{\mathcal{J}(s,s')}) + \frac{1}{4L}\Lambda^2_{s,s',\mathcal{J}(s,s')^c}\right\}^{1/2}$$

defined using the cones $\mathcal{C}_s$ obtained from (18).

2.2.3. ℓ_1 Pooling and Max-Out

Propostion 2.6 can be used to obtain a bound of the lower Lipschitz constant of the ℓ_1 pooling operator, as well as the Maxout operator (Goodfellow et al., 2013); see section B.4.2 in the supplementary material.

2.3. Random Pooling Operators

What is the minimum amount of redundancy needed to invert a pooling operator? As in previous works on compressed sensing (Candes and Tao, 2004) and phase recovery (Balan et al., 2006), one may address this question by studying random pooling operators. In this case, the lower Lipschitz bounds derived in previous sections can be shown to be positive with probability 1 given appropriate parameters K and L.

The following proposition, proved in Appendix B, analyzes the invertibility of a generic pooling operator constructed from random measurements. We consider a frame $\mathcal{F}$ where its M columns are iid Gaussian vectors of $\mathbb{R}^N$.

Proposition 2.8 *Let $\mathcal{F} = (f_1, \dots, f_M)$ be a random frame of $\mathbb{R}^N$, organized into K disjoint pools of dimension L. With probability 1 P_p is injective (modulo $x \sim -x$) if $K \geq 4N$ for $p = 1, \infty$ and if $K \geq 2N - 1$ for $p = 2$.*

The size of the pools L does not influence the injectivity of random pooling, but it affects the stability of the inverse, as shown in proposition 2.6. The half-rectified case requires extra care, since the set of admissible switches $\overline{\Omega}$ might contain frames with $M < N$ columns with non-zero probability, and is not considered in the present work.

3. Numerical Experiments

Our main goal in this section is to experimentally compare the invertibility of ℓ_p pooling for $p \in \{1, 2, \infty\}$, with and without rectification. Unlike in the previous sections, we will not consider the Lipschitz bounds, as we do not know a good way to measure these experimentally. Our experiments suggest that recovery is roughly the same difficulty for $p = 1, 2, \infty$, and that rectification makes recovery easier.

In the ℓ_2 case without rectification, and with $d = 2$, a growing body of works (Candes et al., 2013; Waldspurger et al., 2012) describe how to invert the pooling operator. This is often called phase recovery. A problem for us is a lack of a standard algorithm when $p \neq 2$ or with rectification. We will see that the simple alternating minimization algorithm of (Gerchberg and Saxton, 1972) can be adapted to these situations. However, alternating minimization with random initialization is known to be an inferior recovery algorithm for $p = 2$, and so any conclusions we will draw about ease of recovery will be tainted, as we would be testing whether the algorithm is equally bad in the various situations, rather than if the problems are equally hard. We will show that in certain cases, a training set allows us to find a good initialization for the alternating minimization, leading to excellent recovery performance, and that in this setting, or the random setting, recovery via alternating minimization is roughly as succesful for each of the p, suggesting invertibility is equally hard for each p. In the same way, we will see evidence that half rectification before pooling makes recovery easier.

3.1. Recovery Algorithms

3.1.1. Alternating minimization

A greedy method for recovering the phase from the modulus of complex measurements is given in (Gerchberg and Saxton, 1972); this method naturally extends to the case at hand. As above, denote the frame $\{f_1, ..., f_M\} = \mathcal{F}$, let $\mathcal{F}_k$ be the frame vectors in the kth block, and set I_k to be the indices of the kth block. Let $\mathcal{F}^{(-1)}$ be the pseudoinverse of $\mathcal{F}$; set $(P_p(x))_k = ||\mathcal{F}_k x||_p$. Starting with an initial signal x^0, update

1. $y^{(n)}_{I_k} = (P_p(x))_k \frac{\mathcal{F}_k x^{(n)}}{||\mathcal{F}_k x^{(n)}||_p}, k = 1 \dots K,$
2. $x^{(n+1)} = \mathcal{F}^{(-1)} y^{(n)}.$

This approach is not, as far as we know, guarantee to converge to the correct solution, even when P_p is invertible. However, in practice, if the inversion is easy enough, or if x_0 is close to the true solution, the method can work well. Moreover, this algorithm can be run essentially unchanged for each ℓ_p; for half rectification, we only use the nonegative entries in y for reconstruction.

In the experiments below, we will use random, Gaussian i.i.d. $\mathcal{F}$, but also we will use the outputs of dictionary learning with block sparsity. The $\mathcal{F}$ generated this way is not really a frame, as the condition number of a trained dictionary on real data is often very high. In this case, we will renormalize each data point to have norm 1, and modify the update $x^{(n+1)} = \mathcal{F}^{(-1)} y^{(n)}$ to

2. $x^{(n+1)} = arg \min_{||x||_2=1} ||\mathcal{F}x - y^{(n)}||^2.$

In practice, this modification might not always be possible, since the norm $\|x\|$ is not explicitly presented in P_p. However, in the classical setting of Fourier measurements and positive x, this information is available. Moreover, our empirical experience has been that using this regularization on well conditioned analysis dictionaries offers no benefit; in particular, it gives no benefit with random analysis matrices.

3.1.2. Phaselift and Phasecut

Two recent algorithms (Candes et al., 2013) and (Waldspurger et al., 2012) are guaranteed with high probability to solve the (classical) problem of recovering the phase of a complex signal from its modulus, given enough random measurements. In practice both perform better than the greedy alternating minimization. However, it is not obvious to us how to adapt these algorithms to the ℓ_p setting.

3.1.3. Nearest neighbors regression

We would like to use the same basic algorithm for all settings to get an idea of the relative difficulty of the recovery problem for different p, but also would like an algorithm with good recovery performance. If our algorithm simply returns poor results in each case, differences between the cases might be masked.

The alternating minimization can be very effective when well initialized. When given a training set of the data to recover, we use a simple regression to find x_0. Fix a number of neighbors q (in the experiments below we use $q = 10$, and suppose X is the training set). Set $G = P_p(X)$, and for a new point x to recover from $P_p(x)$, find the q nearest neighbors in G of $P_p(x)$, and take their principal component to serve as x_0 in the alternating minimization algorithm. We use the fast neighbor searcher from (Vedaldi and Fulkerson, 2008) to accelerate the search.

3.2. Experiments

We discuss results on the MNIST dataset, available at http://yann.lecun.com/exdb/mnist/, and on 16×16 patches drawn from the VOC dataset, available at http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc2012/. For each of these data sets, we run experiments with random dictionaries and adapted dictionaries. We also run experiments where the data and the dictionary are both Gaussian i.i.d.; in this case, we do not use adapted dictionaries.

The basic setup of the experiments in each case is the same: we vary the number of measurements (that is, number of pools) over some range, and attempt to recover the original signal from the ℓ_p pooled measurements, using various methods. We record the average angle between the recovered signal r and the original x, that is, we use $|r^T x|^2/(||r||^2||x||^2)$ as the measure of success in recovery.

In each case the random analysis dictionary $\mathcal{F}$ is built by fixing a size parameter m, and generating a Gaussian i.i.d. matrix $\mathcal{F}_0$ of size $2m \times n$, where $n = 100$ for MNIST, and $n = 256$ for VOC. Each pair of rows of $\mathcal{F}_0$ is then orthogonalized to obtain $\mathcal{F}$; that is we use groups of size 2, where the pair of elements in each group are orthogonal. This allows us to use standard phase recovery software in the ℓ_2 case to get a baseline. We used the ADMM version of phaselift from (Ohlsson et al., 2012) and the phasecut algorithm of (Waldspurger et al., 2012). For all of our data sets, the latter gave better results (note that phasecut can explicitly use the fact that the solution to the problem is real, whereas that version of phaselift cannot), so we report only the phasecut results.

In the experiments with adapted dictionaries, the dictionary is built using block OMP and batch updates with a K-SVD type update (Aharon et al., 2006); in this case, $\mathcal{F}$ is the transpose of the learned dictionary. We again use groups of size 2 in the adapted dictionary experiments.

We run two sets of experiments with Gaussian i.i.d. data and dictionaries, with $n = 20$ and $n = 40$. We consider m in the range from $n/2$ to $8n$. On this data set, phaselift outperforms alternating minimization; see the supplementary material.

For MNIST, we use the standard training set projected to $\mathbb{R}^{100}$ via PCA, and we let the number of dictionary elements range from 60 to 600 (that is, 30 to 300 measurements). On this data set, alternating minimization with nearest neighbor initialization gives exact reconstruction by 130 measurements; for comparison, Phaselift at $m = 130$ has mean square angle of .48; see the supplementary material.

We draw approximately 5 million 16×16 grayscale image patches from the PASCAL VOC data set; these are sorted by variance, and the largest variance 1 million are kept. The mean is removed from each patch. These are split into a training set of 900000 patches and a test set of 100000 patches. In this experiment, we let m range from 30 to 830. On this data set, by $m = 330$ measurements, alternating minimization with nearest neighbor initialization recovers mean angle of .97; for comparison, Phaselift at $m = 330$ has mean angle of .39; see the supplementary material.

3.3. Analysis

The experiments show (see figures 1 and 2) that:

- For every data set, with random initializations and dictionaries, recovery is easier with half rectification before pooling than without (green vs dark blue in figures).

- ℓ_∞, ℓ_1, and ℓ_2 pooling appear roughly the same difficulty to invert, regardless of algorithm (each column of figures, corresponding to an ℓ_p, is essentially the same).

- Good initialization improves performance; indeed, alternating minimization with nearest neighbor regression outperforms phaselift and phasecut (which of course do not have the luxury of samples from the prior, as the regressor does). We believe this of independent interest.

- Adapted analysis "frames" (with regularization) are easier to invert than random analysis frames, with or without regularization (the bottom row of each pair of graphs vs the top row of each pair in Figure 2).

Each of these conclusions is unfortunately only true up to the optimization method- it may be true that a different optimizer will lead to different results. With learned initializations and alternating minimization, recovery results can be better without half rectification. Note this is only up until the point where the alternating minimization gets better than the learned initialization without any refinement, and is especially true for random dictionaries. The simple interpretation is that the reconstruction step 2 of the alternating minimization just does not have a large enough span with roughly half the entries removed; that is, this is an effect of the optimization, not of the difference between the difficulty of the problems.

4. Conclusion

We have studied conditions under which neural network layers of the form (1) and (2) preserve signal information. As one could expect, recovery from pooling measurements is only guaranteed under large enough redundancy and configurations of the subspaces, which depend upon which ℓ_p is considered. We have proved conditions which bound the lower Lipschitz constants for these layers, giving quantitative descriptions of how much information they preserve. Furthermore, we have given conditions under which modules with random filters are invertible. We have also given experimental evidence that for both random and adapted modules, it is roughly as easy to invert ℓ_p pooling with $p = 1$, 2, and ∞; and shown that when given training data, alternating minimization gives state of the art phase recovery with a regressed initialization.

However, we are not anywhere near where we would like to be in understanding these systems, or even the invertibility of the layers of these systems. This work gives little direct help to a practicioner asking the question "how should I design my network?". In particular, our results just barely touch on the distribution of the data; but the experiments make it clear (see also (Ohlsson et al., 2012)) that knowing more information about the data changes the invertibility of the mappings. Moreover, preservation of information needs to be balanced against invariance, and the tension between these is not discussed in this work. Even in the setting of this work, without consideration of the data distribution or tension with invariance, Proposition 2.4 although tight, is not easy to use, and even though we are able to use 2.6 to get an invertibility result, it is probably not tight.

This work also shows there is much research to do in the field of algorithmic phase recovery. What are correct algorithms for ℓ_p inversion, perhaps with half rectification? How can we best use knowledge of the data distribution for phase recovery, even for the well studied ℓ_2 case? Is

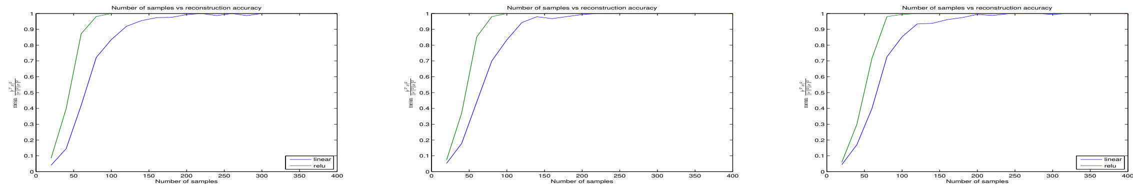


Figure 1. Average recovery angle using alternating projections on random data; each x is Gaussian i.i.d. in $\mathbb{R}^{40}$. The vertical axis measures the average value of $|r^T x|^2/(||r||^2||x||^2)$, where r is the recovered vector, over 50 random test points. The horizontal axis is the number of measurements (the size m of the analysis dictionary is twice the x axis in this experiment). The leftmost figure is ℓ_1 pooling, the middle ℓ_2, and the right max pooling. The dark blue curve is alternating minimization, and the green curve is alternating minimization with half rectification; both with random initialization.

it possible to guarantee that a well initialized alternating minimization converges to the correct solution?

References

Kannan Achan, Sam T. Roweis, and Brendan J. Frey. Probabilistic inference of speech signals from phaseless spectrograms. In *In Neural Information Processing Systems 16*, pages 1393–1400. MIT Press, 2003.

M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Trans. Sig. Proc.*, 54(11):4311–4322, November 2006. ISSN 1053-587X.

Radu Balan and Yang Wang. Invertibility and robustness of phaseless reconstruction, 2013.

Radu Balan, Pete Casazza, and Dan Edidin. On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, 20(3):345–356, May 2006.

J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions of PAMI*, 2012.

Jameson Cahill, Peter G. Casazza, Jesse Peterson, and Lindsey Woodland. Phase retrieval by projections, 2013.

E. Candes and T. Tao. Near Optimal Signal Recovery From Random Projections: Universal Encoding Strategies? *ArXiv Mathematics e-prints*, October 2004.

Emmanuel J. Candes, Thomas Strohmer, and Vladislav Voroninski. Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming. *Communications on Pure and Applied Mathematics*, 66(8):1241–1274, 2013.

R. W. Gerchberg and W. Owen Saxton. A practical algorithm for the determination of the phase from image and diffraction plane pictures. *Optik*, 35:237–246, 1972.

I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout Networks. *ArXiv e-prints*, February 2013.

A. Hyvärinen and P. Hoyer. A two-layer sparse coding model learns simple and complex cell receptive fields and topography from natural images. *Vision Research*, 41(18):2413–2423, August 2001. ISSN 00426989. doi: 10.1016/s0042-6989(01)00114-6.

Koray Kavukcuoglu, Marc'Aurelio Ranzato, Rob Fergus, and Yann LeCun. Learning invariant features through topographic filter maps. In *Proc. International Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 2009.

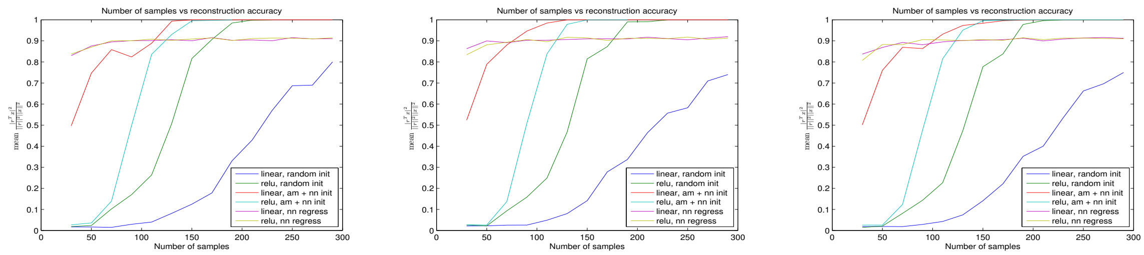
S. Mallat. Group Invariant Scattering. *Communications in Pure and Applied Mathematics*, 2012.

Henrik Ohlsson, Allen Y. Yang, Roy Dong, and S. Shankar Sastry. Cprl – an extension of compressive sensing to the phase retrieval problem. In Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger, editors, *NIPS*, pages 1376–1384, 2012.

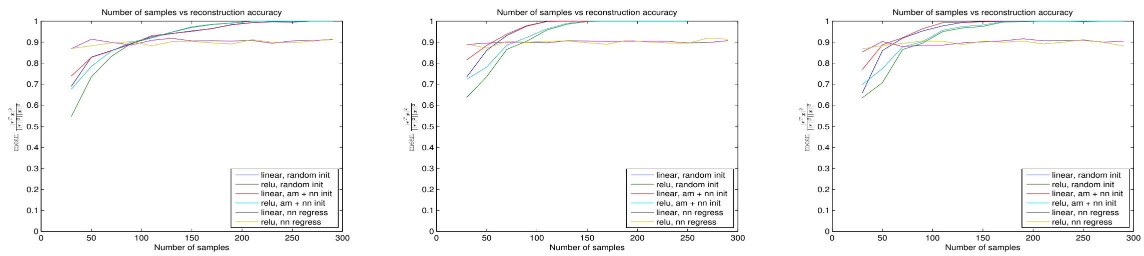
Y. Ohlsson, H. Eldar. On conditions for uniqueness for sparse phase retrieval, 2013.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

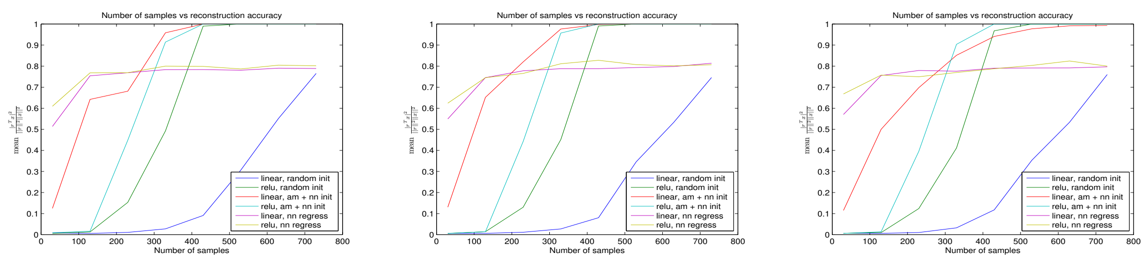
Irène Waldspurger, Alexandre d'Aspremont, and Stéphane Mallat. Phase recovery, maxcut and complex semidefinite programming, 2012.



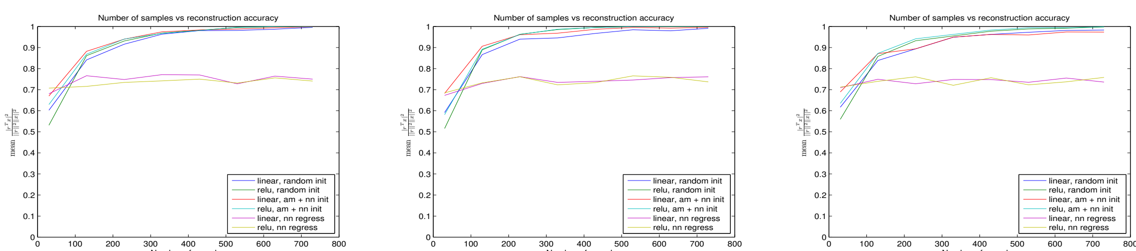
(a) MNIST, random filters



(b) MNIST, adapted filters



(c) Image patches, random filters



(d) Image patches, adapted filters

Figure 2. Average recovery angle using alternating projections on image patch data points. The vertical axis measures the average value of $|r^T x|^2/(||r||^2||x||^2)$, where r is the recovered vector, over 50 random test points. The horizontal axis is the number of measurements (the size of the analysis dictionary is twice the x axis in this experiment). The leftmost figure is ℓ_1 pooling, the middle ℓ_2, and the right max pooling. In the top row of each pair of rows the analysis dictionary is Gaussian i.i.d.; in the bottom row of each pair of rows, it is generated by block OMP/KSVD with 5 nozero blocks of size 2. The dark blue curve is alternating minimization, and the green curve is alternating minimization with half rectification; both with random initialization. The magenta and yellow curves are the nearest neighbor regressor described in 3.1.3 without and with rectification; and the red and aqua curves are alternating minimization initialized via neighbor regression, without and with rectification. See Section 3.3 for a discussion of the figures.

A. Comparison between phaselift and the various alternating minimization algorithms

Here we give a brief comparison between the phaselift algorithm and the algorithms we use in the main text. Our main goal is to show that the similarities between the ℓ_1, ℓ_2, ℓ_∞ recovery results are not just due to the alternating minimization algorithm performing poorly on all three tasks; however we feel that the quality of the recovery with a regressed initialization is interesting in itself, especially considering that it is much faster than either phaselift or phasecut.

In figures 3, and 4 we compare phaselift against alternating minimization with a random initialization and alternating minimization with a nearest neghbor/locally linear regressed initialization. Because we are comparing against phasecut, here we only show inversion of ℓ_2 pooling.

In figure of 3, we use random data and a random dictionary. As the data has no structure, we only compare against random initialization, with and without half rectification. We can see from figure 3 in this case, where we do not know a good way to initialize the alternating minimization, alternating minimization is significantly worse than phasecut. On the other hand, recovery after rectified pooling with alternating minimization does almost as well as phasecut.

In the examples where we have training data, shown in figure 4, alternating minimization with the nearest neighbor regressor (red curve) performs significantly better than phasecut (green and blue curves). Of course phasecut does not get the knowledge of the data distribution used to generate the regressor.

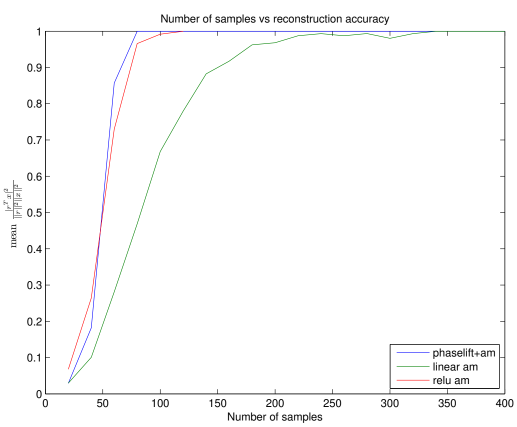


Figure 3. Average recovery angle using phaselift and alternating minimization on random data, Gaussian i.i.d. points in $\mathbb{R}^{40}$. The blue curve is phaselift followed by alternating minimization; the green curve is alternating minimization, and the red is alternating minimization on pooling following half rectification.

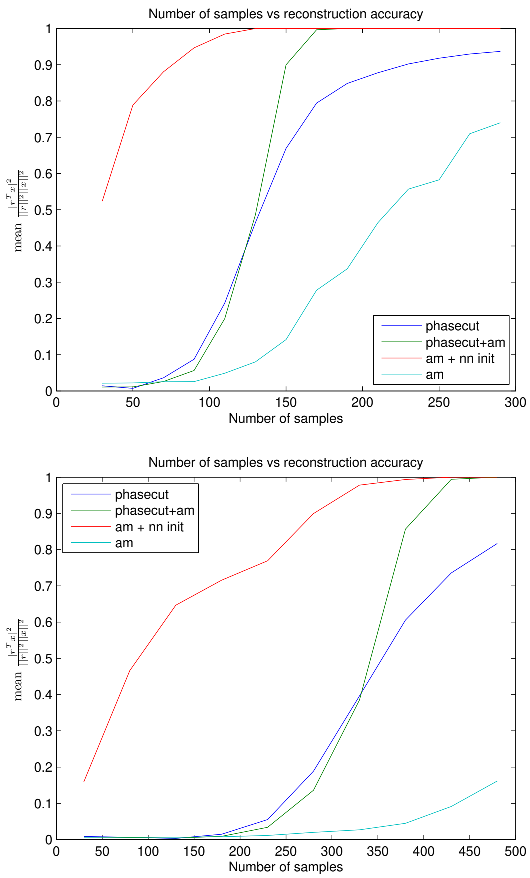


Figure 4. Average recovery angle using phaselift and alternating minimization on MNIST and patches data sets. Top: MNIST digits, projected via PCA to $\mathbb{R}^{100}$. Bottom: 16x16 image patches with mean removed. The red curve is alternating minimization with nearest neighbor initialization, the green is alternating minimization initialized by phasecut (this is the recommended usage of phasecut), the blue is phasecut with no alternating minimization, and the aqua is alternating minimization with a random initialization.

B. Proofs of results in Section 2

B.1. Proof of Proposition 2.2

Let us first show that $A_0 > 0$ is sufficient to construct an inverse of M_α. Let $x \in \mathbb{R}^N$. By definition, the coordinates of $M_\alpha(x) > \alpha$ correspond to

$$s(x) = \{i \ s.t. \langle x, f_i \rangle > \alpha_i\} \subset \{1, \dots, M\} ,$$

which in particular implies that x is known to lie in $V_{S(x)}$, the subspace generated by $s(x)$. But the restriction $\mathcal{F}_{s(x)}$ is a linear operator, which can be inverted in V_S as long as $\lambda_-(\mathcal{F}_{s(x)}\big|_{V_S}) \geq A_0 > 0$.

Let us now show that $A_0 > 0$ is also necessary. Let us suppose that for some S, $\mathcal{F}_S$ is such that $\lambda_-(\mathcal{F}_S\big|_{V_S}) = 0$. It results that there exists $\eta \in V_S$ such that $\|\eta\| > 0$ but $\|\mathcal{F}_S \eta\| = 0$. Since S is a cone, we can find $x \in S$ and $\epsilon \neq 0$ small enough such that $x + \epsilon e \in S$. It results that $M_\alpha(x) = M_\alpha(x + \epsilon e)$ which implies that M_α cannot be injective.

Finally, let us prove (9). If x, x' are such that $S = s(x) = s(x')$, then

$$\|M_\alpha(x) - M_\alpha(x')\| = \|\mathcal{F}_S(x - x')\| \geq A_0(x - x') .$$

If $s(x) \neq s(x')$, we have that $|M_\alpha(x)_i - M_\alpha(x')_i| = |\langle x - x', f_i \rangle|$ if $i \in s(x) \cap s(x')$ and $|M_\alpha(x)_i - M_\alpha(x')_i| \geq |\langle x - x', f_i \rangle|$ if $i \in s(x) \cup s(x')$, $i \notin s(x) \cap s(x')$. It results that

$$\|M_\alpha(x) - M_\alpha(x')\| \geq \|\mathcal{F}_{s(x) \cup s(x')}(x - x')\| \geq A_0 \|x - x'\| .$$

$\square$.

B.2. Proof of Proposition 2.4

The upper Lipschitz bound is obtained by observing that, in dimension d,

$$\forall\, y \in \mathbb{R}^d \ , \ \|y\|_1 \leq \sqrt{d}\|y\|_2 \ , \ \|y\|_\infty \leq d\|y\|_2 .$$

It results that

$$\begin{aligned} \|P_p(x) - P_p(x')\| &\leq \alpha_p \|P_2(x) - P_2(x')\| && (20) \\ &= \alpha_p \|M(x) - M(x')\| \leq \alpha_p \lambda_+(\mathcal{F}) . \end{aligned}$$

Let us now concentrate on the lower Lipschitz bound. Given $x, x' \in \mathbb{R}^n$, we first consider a rotation $\tilde{\mathcal{F}}_k$ on each subspace $\mathcal{F}_k$ such that $\langle x, \tilde{f}_{k,j} \rangle = \langle x', \tilde{f}_{k,j} \rangle = 0$ for $j > 2$, which always exists. If now we modify $\tilde{\mathcal{F}}_k$ by applying a rotation of the remaining two-dimensional subspace such that x and x' are bisected, one can verify that

$$\begin{aligned} (\|\mathcal{F}_k x\|_2 - \|\mathcal{F}_k x'\|_2)^2 &= (\|\widetilde{\mathcal{F}}_k x\|_2 - \|\widetilde{\mathcal{F}}_k x'\|_2)^2 \\ &= (|\langle x, \tilde{f}_{k,1} \rangle| - |\langle x', \tilde{f}_{k,1} \rangle|)^2 \\ &\quad + (|\langle x, \tilde{f}_{k,2} \rangle| - |\langle x', \tilde{f}_{k,2} \rangle|)^2 , \end{aligned}$$

which implies, by denoting $M(x) = (|\langle x, \widetilde{f}_{k,j} \rangle|)_{k,j}$, that $\|P_2(x) - P_2(x')\| = \|M(x) - M(x')\|$. Since $\widetilde{\mathcal{F}} \in \mathcal{Q}_2$, it results from Proposition 2.1 that

$$\begin{aligned} \|P_2(x) - P_2(x')\| &\geq d(x, x') \min_{S \subset \{1 \dots m\}} \sqrt{\lambda_-^2(\widetilde{\mathcal{F}}_S) + \lambda_-^2(\widetilde{\mathcal{F}}_{S^c})} \\ &\geq d(x, x') A_2 \ \square. && (21) \end{aligned}$$

B.3. Proof of Corollary 2.5

Given x, x', let I denote the groups I_k, $k \leq K$ such that $S_x \cap S_{x'} \cap I_k = I_k$. It results that

$$\begin{aligned} &\|R_p(x) - R_p(x')\|^2 \\ &= \sum_{k \in I} |R_p(x)_k - R_p(x')_k|^2 + \sum_{k \notin I} |R_p(x)_k - R_p(x')_k|^2 \\ &\geq \sum_{k \in I} |R_p(x)_k - R_p(x')_k|^2 + \sum_{k \notin I} (\|M_0(x)\big|_{I_k} - M_0(x')\big|_{I_k}\|)^2 . \end{aligned}$$

On the groups in I we can apply the same arguments as in theorem 2.4, and hence find a frame $\tilde{\mathcal{F}}$ from the family $\widetilde{\mathcal{Q}}_{p,x,x'}$ such that

$$\|R_p(x) - R_p(x')\|_I = \|M(x) - M(x')\| ,$$

with $M(x) = (|\langle x, \widetilde{f}_{k,j} \rangle|)_{k \in I, j}$ and $\{\widetilde{f}_{k,j}\} \in \widetilde{\mathcal{Q}}_{p,x,x'}$. Then, by following the same arguments used previously, it results from the definition of $\tilde{A}_p$ that

$$\|R_p(x) - R_p(x')\| \geq \tilde{A}_p d(x, x') .$$

Finally, the upper Lipschitz bound is obtained by noting that

$$\|M_\alpha(x) - M_\alpha(x')\| \leq \|\mathcal{F}(x - x')\| ,$$

and using the same argument as in (20) $\square$.

B.4. Proof of Proposition 2.6

Let $x, x' \in \mathbb{R}^N$, and let $\mathcal{J} = s(x) \cap s(x')$. Suppose first that $\mathcal{C}_{s(x)} \cap \mathcal{C}_{s(x')} \neq \emptyset$. Since $\|P_\infty x - P_\infty\| \geq \||\mathcal{F}_s x| - |\mathcal{F} s x'|\big|_{\mathcal{J}}\|$, it results that

$$d(x, x') A_{s(x), s(x')} \leq \|P_\infty x - P_\infty x'\| \qquad (22)$$

by Proposition 2.1 and by definition (**??**).

Let us now suppose $\mathcal{C}_{s(x)} \cap \mathcal{C}_{s(x')} = \emptyset$, and let $z = P_\infty x - P_\infty x'$. It results that $z = |\mathcal{F}_{s(x)} x| - |\mathcal{F}_{s(x')} x'| \in \mathbb{R}^K$, and hence we can split the coordinates $(1 \dots K)$ into Ω, Ω^c such that

$$\begin{aligned} z\big|_\Omega &= \mathcal{F}_{s(x)}\big|_\Omega(x) - \mathcal{F}_{s(x')}\big|_\Omega(x') , \\ z\big|_{\Omega^c} &= \mathcal{F}_{s(x)}\big|_{\Omega^c}(x) + \mathcal{F}_{s(x')}\big|_{\Omega^c}(x') . \end{aligned}$$

We shall concentrate in each restriction independently. Since $\mathcal{F}_{s(x')}\big|_{\Omega}(x') \in \mathcal{F}_{s(x')}\big|_{\Omega}(\mathcal{C}_{s(x')})$, it results that

$$\begin{aligned} \|z\big|_{\Omega}\| &\geq \inf_{y \in \mathcal{F}_{s(x')}\big|_{\Omega}} \|\mathcal{F}_{s(x)}\big|_{\Omega}(x) - y\| \\ &\geq \|\mathcal{F}_{s(x)}\big|_{\Omega}(x)\| \cdot |\sin(\beta(s(x), s(x'), \Omega))| \end{aligned} \quad (23)$$

Since by definition

$$\forall\, k\ , \ \sum_{j \in I_k} |\langle x, f_j \rangle|^2 \leq \frac{1}{|I_k|} |\langle x, f_{s(x)_k} \rangle|^2\ ,$$

it results, assuming without loss of generality that all pools have equal size ($|I_k| = \frac{M}{K}$),

$$\begin{aligned} \forall\, x \in \mathcal{C}_s\ , \ \|\mathcal{F}_{s(x)}\big|_{\Omega}(x)\| &\geq \sqrt{\frac{K}{M}} \|\mathcal{F}\big|_{\Omega}(x)\| \\ &\geq \sqrt{\frac{K}{M}} \lambda_{-}(\mathcal{F}\big|_{\Omega}) \|x\| \end{aligned} \quad (24)$$

Equivalently, since $\mathcal{F}_{s(x)}\big|_{\Omega}(x) \in \mathcal{F}_{s(x)}\big|_{\Omega}(\mathcal{C}_{s(x)})$ we also have

$$\|z\big|_{\Omega}\| \geq \sqrt{\frac{K}{M}} \lambda_{-}(\mathcal{F}\big|_{\Omega}) \cdot |\sin(\beta(s(x), s(x'), \Omega))| \cdot \|x'\|\ . \quad (25)$$

It follows that

$$\begin{aligned} \|z\big|_{\Omega}\| &\geq \sqrt{\frac{K}{M}} \lambda_{-}(\mathcal{F}\big|_{\Omega})\, |\sin(\beta(s(x), s(x'), \Omega))| \max(\|x\|, \|x'\|) \\ &\geq \sqrt{\frac{K}{4M}} \lambda_{-}(\mathcal{F}\big|_{\Omega})\, |\sin(\beta(s(x), s(x'), \Omega))|\, d(x, x')\ . \end{aligned} \quad (26)$$

By aggregating the bound for Ω and Ω^c we obtain (28) $\square$.

B.4.1. MAXOUT

These results easily extend to the so-called Maxout operator (Goodfellow et al., 2013), defined as $x \mapsto MO(x) = \{\max_{j \in I_k} \langle x, f_j \rangle\, ;\, k = 1 \dots K\}$. By redefining the switches of x as

$$s(x) = \{j\, ;\, \langle x, f_j \rangle > \max(\langle x, f_{j'} \rangle\, ;\, \forall\, j' \in \text{pool}(j)\}\ , \quad (27)$$

the following corollary computes a Lower Lipschitz bound of $MO(x)$:

Corollary B.1 *The Maxout operator MO satisfies (19) with $A(s, s')$ defined using the switches (27).*

B.4.2. ℓ_1 POOLING

Propostion 2.6 can be used to obtain a bound of the lower Lipschitz constant of the ℓ_1 pooling operator.

Observe that for $x \in \mathbb{R}^n$,

$$\|x\|_1 = \sum_i |x_i| = \max_{\epsilon_i = \pm 1} |\langle x, \epsilon \rangle|\ .$$

It results that $P_1(x; \mathcal{F}) \equiv P_\infty(x; \widetilde{\mathcal{F}})$, with

$$\widetilde{\mathcal{F}} = (\tilde{f}_{k,\epsilon} = \sum_i \epsilon(i) f_{k,i}\, ;\, k = 1 \dots, K\, ; \epsilon \in \{-1, 1\}^L\}\ .$$

Each pool $\widetilde{\mathcal{F}}_k$ can be rewritten as $\widetilde{\mathcal{F}}_k = H_L \mathcal{F}_k$, where H_L is the $L \times 2^L$ Hadamard matrix whose rows contain the ϵ vectors. One can verify that $H_L^T H_L = 2^L \mathbf{1}$, which implies that for any $\Omega \subseteq \{1 \dots K\}$, $\lambda_{-}(\widetilde{\mathcal{F}}\big|_{\Omega}) = 2^{L/2} \lambda_{-}(\mathcal{F}\big|_{\Omega})$. It results that

Corollary B.2 *The ℓ_1 pooling operator P_1 satisfies*

$$\forall\, x, x'\ ,\ d(x, x') \left(\min_{s, s'} \tilde{A}(s, s') \right) \leq \|P_1(x) - P_1(x')\|\ , \quad (28)$$

where $d(x, x') = \min(\|x - x'\|, \|x + x'\|)$ and

$$\begin{aligned} \tilde{A}(s, s') = \max \Big\{ &\min_{\Omega \subseteq \mathcal{J}(s,s')} \sqrt{\lambda_{-}^2(\widetilde{\mathcal{F}}_{\Omega}) + \lambda_{-}^2(\widetilde{\mathcal{F}}_{\mathcal{J}-\Omega})}\ , \\ &\frac{1}{2} \min_{\Omega \subseteq \{1 \dots K\}} \sqrt{\Lambda_{s,s',\Omega}^2 + \Lambda_{s,s',\Omega^c}^2} \Big\}\ , \end{aligned}$$

with s, s' and $\beta(s, s')$ are defined on the frame $\widetilde{\mathcal{F}}$.

Similarly as in Corollary 2.7, one can obtain a similar bound for the Rectified ℓ_1 pooling.

B.5. Proof of Corollaries 2.7 and B.1

The result follows immediately from Proposition 2.6, by replacing the phaseless invertibility condition of Propostion 2.1 by the one in Proposition 2.2. $\square$.

B.6. Proof of Proposition 2.8

Proposition 2.8 also extends to the maxout case. We restate it here with the extra result:

Proposition B.3 *Let $\mathcal{F} = (f_1, \dots, f_M)$ be a random frame of $\mathbb{R}^N$, organized into K disjoint pools of dimension L. Then these statements hold with probability 1:*

1. *P_p is injective (modulo $x \sim -x$) if $K \geq 4N$ for $p = 1, \infty$, and if $K \geq 2N - 1$ for $p = 2$.*
2. *The Maxout operator MO is injective if $K \geq 2N + 1$.*

Let us first prove (i), with $p = \infty$. Let $x, x' \in \mathbb{R}^N$ such that $P_\infty(x) = P_\infty(x')$, and let $s = s(x)$, $s' = s(x')$. The set of K pooling measurements is divided into the intersection $\mathcal{J}(s, s') = \{k\, ;\, s(x)_k = s(x')_k\}$ and its complement $\mathcal{J}(s, s')^c = \{k\, ;\, s(x)_k \neq s(x')_k\}$. Suppose first that $|\mathcal{J}(s, s')| \geq 2N - 1$. Then it results that we can pick $d = \lceil \frac{|\mathcal{J}(s,s')|}{2} \rceil \geq N$ elements of $\mathcal{J}(s, s')$ to form a frame

V, such that either $x - x' \in \text{Ker}(V)$ or $x + x \in \text{Ker}(V)$. Since a random frame of dimension $\geq N$ spans $\mathbb{R}^N$ with probability 1, it results that $x = \pm x'$. Suppose otherwise that $|\mathcal{J}(s, s')| < 2N - 1$. It follows that $|\mathcal{J}(s, s')^c| \geq 2N + 1$, and hence that any partition of $\mathcal{J}(s, s')^c$ into two frames will contain always a frame $\mathcal{F}\big|_{\Omega}$ with at least $N+1$ columns. Since two random subspaces of dimension N in $\mathbb{R}^M$ have nonzero largest principal angle with probability 1 as long as $K > N$, it results that $\Lambda_{s,s',\Omega} > 0$ and hence that $\text{Prob}(|P_\infty(x)\big|_{\mathcal{J}(s,s')^c}| = |P_\infty(x')\big|_{\mathcal{J}(s,s')^c}|) = 0$. The case $p = 1$ is proved identically thanks to Corollary B.2.

Finally, in order to prove (ii) we follow the same strategy. If $|\mathcal{J}(s, s')| \geq N$, then $MO(x) = MO(x') \Rightarrow x = x'$ with probability 1 since $\mathcal{F}\big|_{\mathcal{J}}$ spans $\mathbb{R}^N$ with probability 1. Otherwise it results that $|\mathcal{J}(s, s')^c| \geq N + 1$, which implies $MO(x) \neq MO(x')$, since two random subspaces of dimension N in $\mathbb{R}^{|\mathcal{J}(s,s')^c|}$ have 0 intersection with probability 1.

Let us now prove the case $p = 2$. We start drawing a random basis for each of the pools $F_1, \dots, F_K$. From proposition 2.4, it follows that we have to check that if $M \geq 2N$, the quantity

$$\min_{F'=U\,F\,,\,U^TU=\mathbf{1}} \min_{\Omega \subseteq \{1\dots M\}} \lambda_-^2(F'_\Omega) + \lambda_-^2(F'_Omega^c) > 0$$

with probability 1. If $M \geq 2N - 1$, it follows that either Ω has the property that it intersects at least N pools, either Ω^c intersects N pools. Say it is Ω. Now, for each pool with nonzero intersection, say F_k, we have that

$$\|(F'_k)^T y\| \geq \frac{1}{\sqrt(L)} |\langle f_{k,j}, y\rangle|$$

for some $f_{k,j}$ belonging to the initial random basis of F_k. It results that

$$\lambda_-^2(F'_\Omega) \geq \frac{1}{\sqrt(L)} \lambda_-^2(F^*) \,,$$

where F^* is a subset of N columns of the original frame $\mathcal{F}$, which means

$$\lambda_-^2(F'_\Omega) \geq \frac{1}{\sqrt(L)} \lambda_-^2(F*) > 0 \,.$$

$\square$.

C. Notes on changes from cycle 1

The mathematical results have been essentially rewritten, for clarity as well as to sharpen the bounds. The proofs are now in the supplementary material, as requested by the reviewers. We have used the extra space to expand the introduction, conclusion, and intro to the experiments, in part to explain the connections between the theoretical and experimental parts of the paper, as requested by the reviewers. We also added results on the invertibility of random modules.

We have edited the text in the experiments section and in the captions of the figures to clarify them. Each curve is described in the caption and the text; the graphs are also now specifically referenced in the analysis bullets in section 3.3.

The introduction and conclusion more explicitly address take messages. Note that the take home message is not of the form "this is how to design a network", but rather, "these conditions allow (stable) inversion". We are sympathetic to the reviewers desire for a take home message giving insight into the actual design of networks for practical applications. That is, of course, the ultimate goal of a mathematical analysis of a learning algorithm. However, if the standard for theoretical papers analyzing deep models is that they lead immediately to design suggestions with associated performance increases on benchmarks, it is unlikely that there will ever be a mature enough theory to give honest design suggestions.

Finally, we reprint larger versions of the figures below.

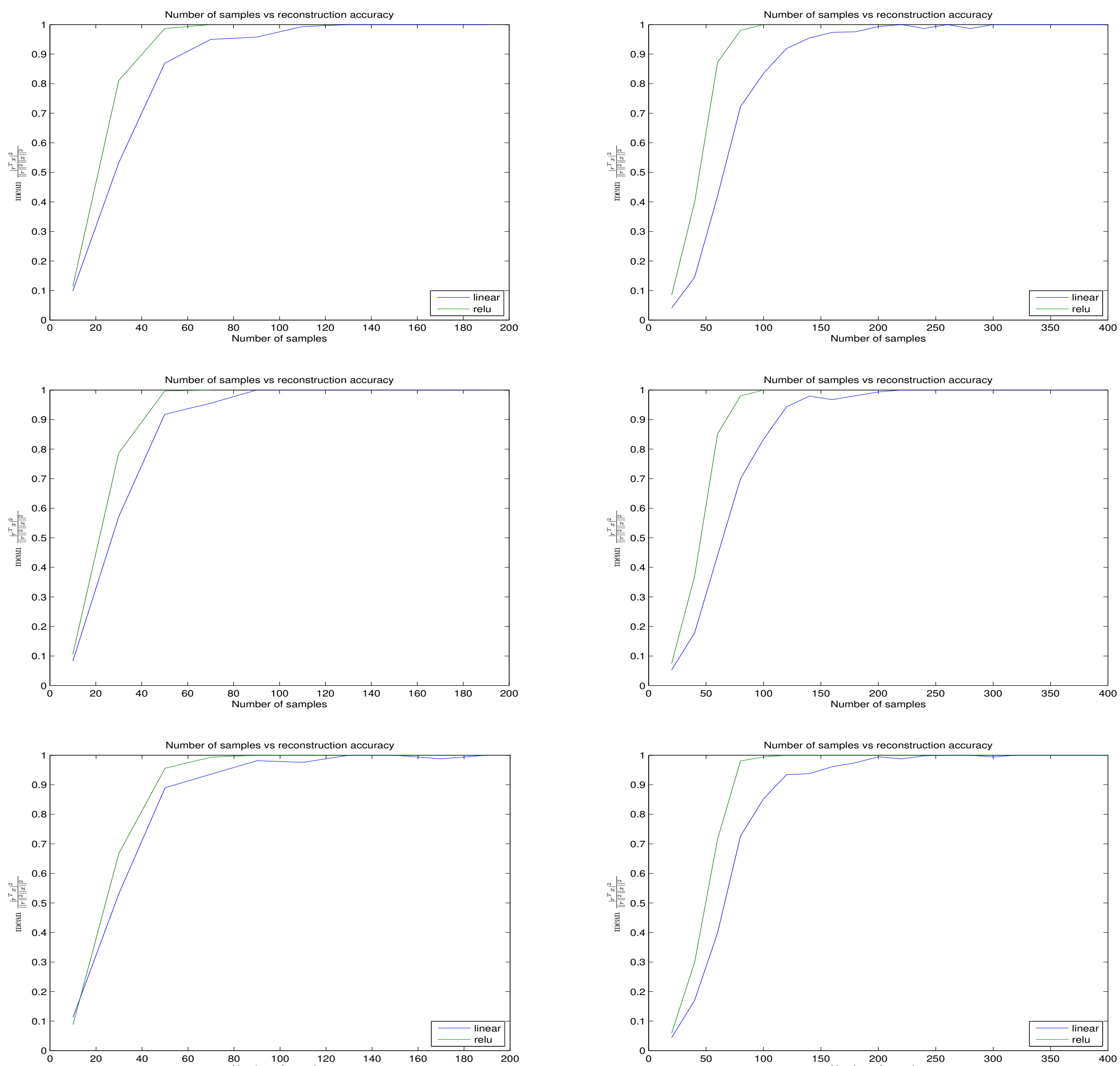


Figure 5. Average recovery angle using alternating projections on random data. The vertical axis measures the average value of $|r^T x|^2/(||r||^2||x||^2)$ over 50 random test points. The horizontal axis is the number of measurements (the size m of the analysis dictionary is twice the x axis in this experiment). The top row is ℓ_1 pooling, the middle ℓ_2, and the bottom max pooling. In the left column each x is Gaussian i.i.d. in $\mathbb{R}^{20}$, on the right, in $\mathbb{R}^{40}$. The dark blue curve is alternating minimization, and the green curve is alternating minimization with half rectification; both with random initialization.

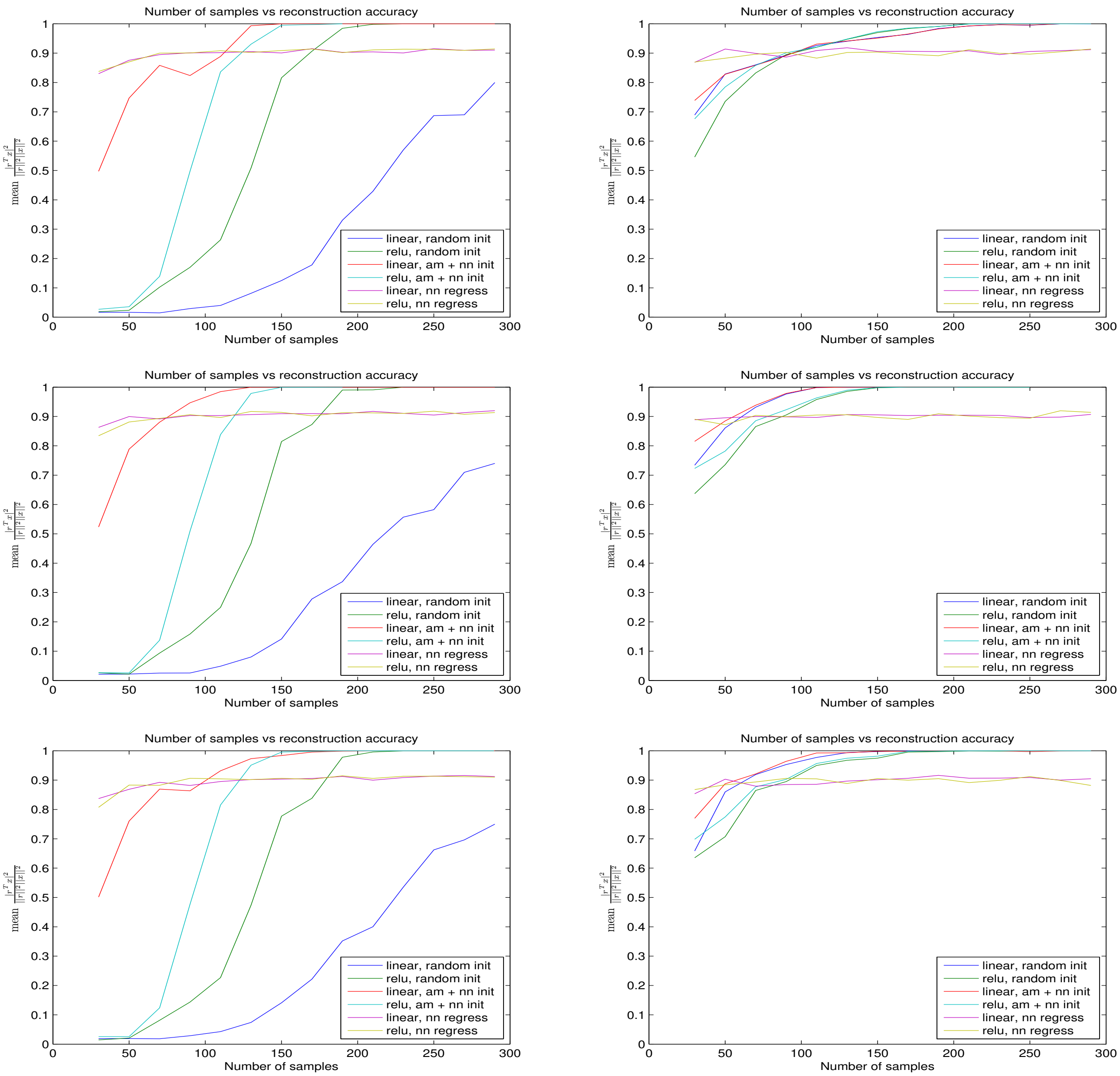


Figure 6. Average recovery angle using alternating projections on MNIST data points. The vertical axis measures the average value of $|r^T x|^2/(||r||^2||x||^2)$ over 50 random test points. The horizontal axis is the number of measurements (the size of the analysis dictionary is twice the x axis in this experiment). The top row is ℓ_1 pooling, the middle ℓ_2, and the bottom max pooling. In the left column the analysis dictionary is Gaussian i.i.d.; in the right column, generated by block OMP/KSVD with 5 nozero blocks of size 2. The dark blue curve is alternating minimization, and the green curve is alternating minimization with half rectification; both with random initialization. The magenta and yellow curves are the nearest neighbor regressor described in 3.1.3 without and with rectification ; and the red and aqua curves are alternating minimization initialized via neighbor regression, without and with rectification.

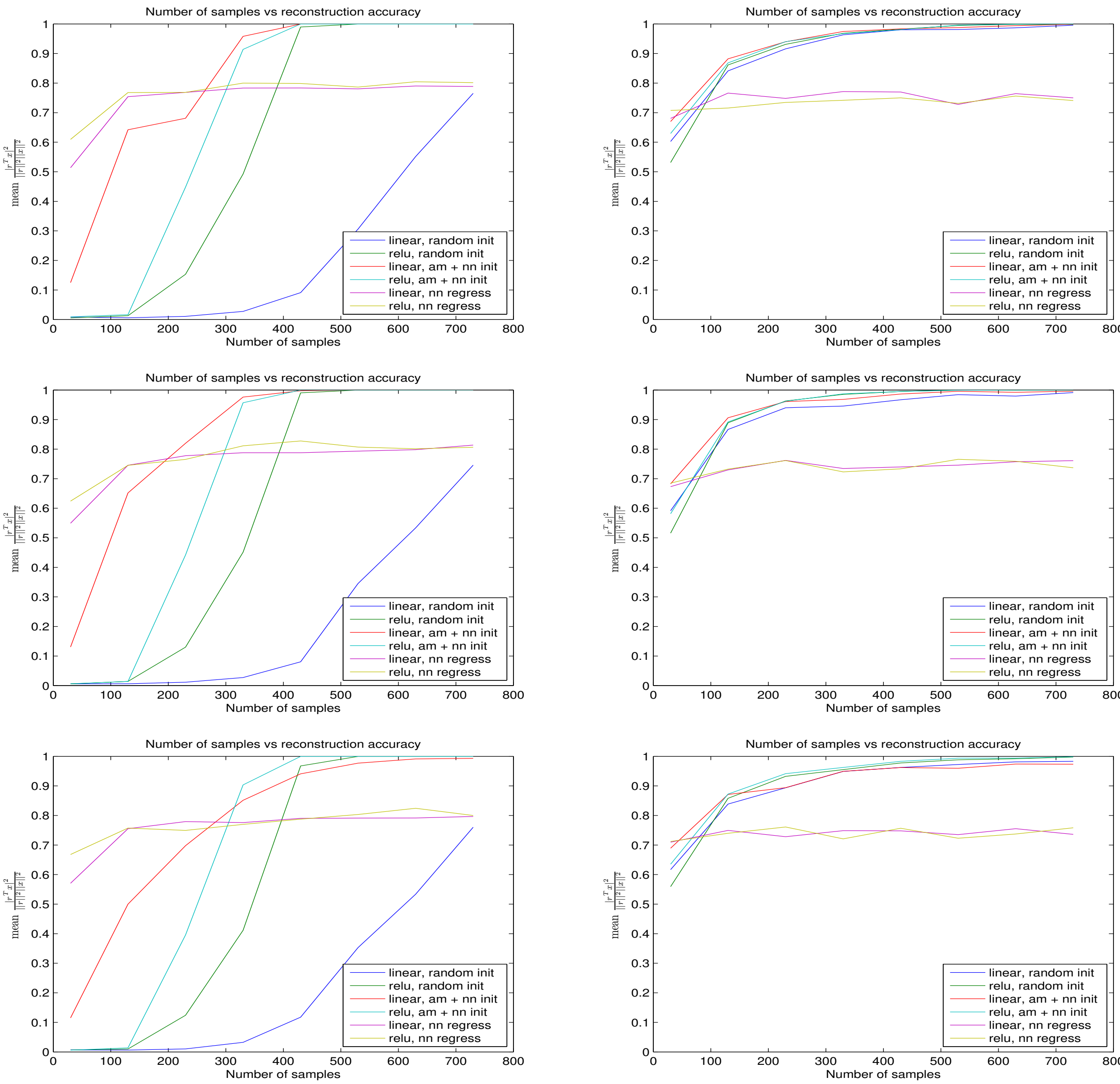
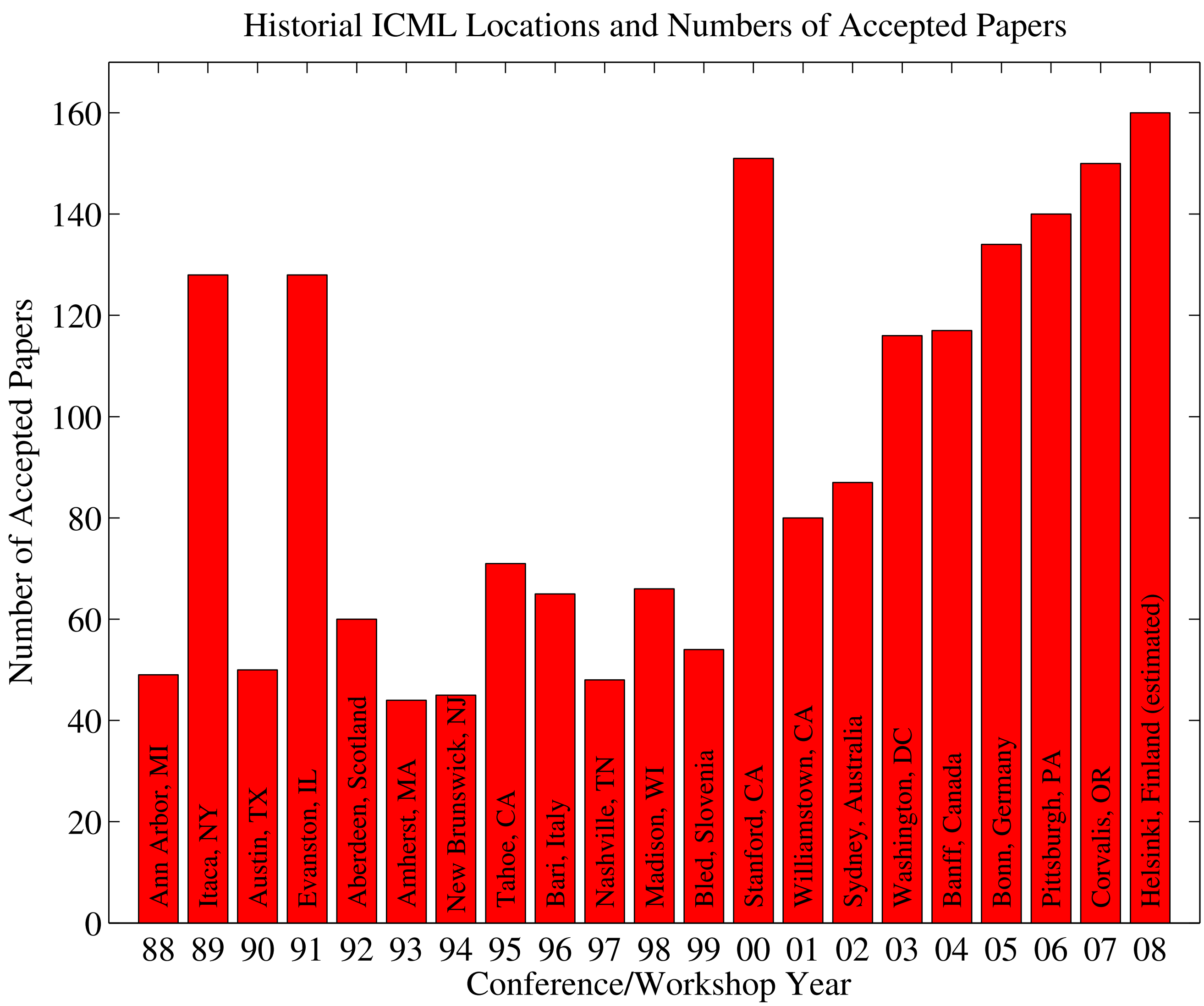


Figure 7. Average recovery angle using alternating projections on image patch data points. The vertical axis measures the average value of $|r^T x|^2/(||r||^2||x||^2)$ over 50 random test points. The horizontal axis is the number of measurements (the size of the analysis dictionary is twice the x axis in this experiment). The top row is ℓ_1 pooling, the middle ℓ_2, and the bottom max pooling. In the left column the analysis dictionary is Gaussian i.i.d.; in the right column, generated by block OMP/KSVD with 5 nozero blocks of size 2. The dark blue curve is alternating minimization, and the green curve is alternating minimization with half rectification; both with random initialization. The magenta and yellow curves are the nearest neighbor regressor described in 3.1.3 without and with rectification; and the red and aqua curves are alternating minimization initialized via neighbor regression, without and with rectification.



Historial ICML Locations and Numbers of Accepted Papers
Number of Accepted Papers
0
20
40
60
80
100
120
140
160
Ann Arbor, MI
Itaca, NY
Austin, TX
Evanston, IL
Aberdeen, Scotland
Amherst, MA
New Brunswick, NJ
Tahoe, CA
Bari, Italy
Nashville, TN
Madison, WI
Bled, Slovenia
Stanford, CA
Williamstown, CA
Sydney, Australia
Washington, DC
Banff, Canada
Bonn, Germany
Pittsburgh, PA
Corvalis, OR
Helsinki, Finland (estimated)
88 89 90 91 92 93 94 95 96 97 98 99 00 01 02 03 04 05 06 07 08
Conference/Workshop Year