Hierarchical Learning from Natural Images

Simon Berring Jeff Sun

CS229: Machine Learning, Autumn 2005
Stanford University
Stanford, CA 94305
sberring@stanford.edu, jsun@cs.stanford.edu

Abstract

In this paper, we apply unsupervised learning methods to construct response functions for V1 simple cells, V1 complex cells, and V2 simple cells from a set of natural images. To support this, we re-implement existing sparse coding methods with the use of commercial optimization software.

Introduction

The human visual cortex contains a small number of self-contained functional units that fit together in reasonably well-understood pathways. The ventral pathway, which is concerned with object recognition, has four stages: V1, V2, V4, and IT (the inferior temporal lobule). V1 consists of simple cells that resemble localized, oriented Gabor filters and complex cells that respond to identical stimuli independent of phase. V1 outputs information to V2, which contains cells thought to respond to broader image contours.

Since the landmark paper of Field & Olshausen (1996), it has been known that linear filters learned as sparse codings on datasets of natural images correspond almost exactly to the receptive fields of V1 simple cells. Hoyer & Hyvärinen (2000), among others, realized that learning an additional layer of nonlinear energies replicated the behavior of V1 complex cells. V1 has received the great majority of research focus in this area because its behavior is well understood and, perhaps, of suspicion that simple information-theoretic elements should not apply to higher levels. However, Hoyer & Hyvärinen (2002) demonstrates that similar sparse-coding techniques may yield the contour activation patterns one would expect in V2.

In this paper, we expand on these results in two ways. First, we introduce learned features at every level, feeding forward from the image to V1 and from V1 to V2. This is in contrast to the hand-coded V1 layers of previous results. Second, we improve existing implementations by treating sparse coding as a convex optimization problem, achieving notable speedup.

Problem Formulation

The problem we pose is that of learning in a three-level network, which we perform in two layers in a feed-forward manner. In the diagram below, the feed-forward direction is bottom up from pixels to V2 filters.

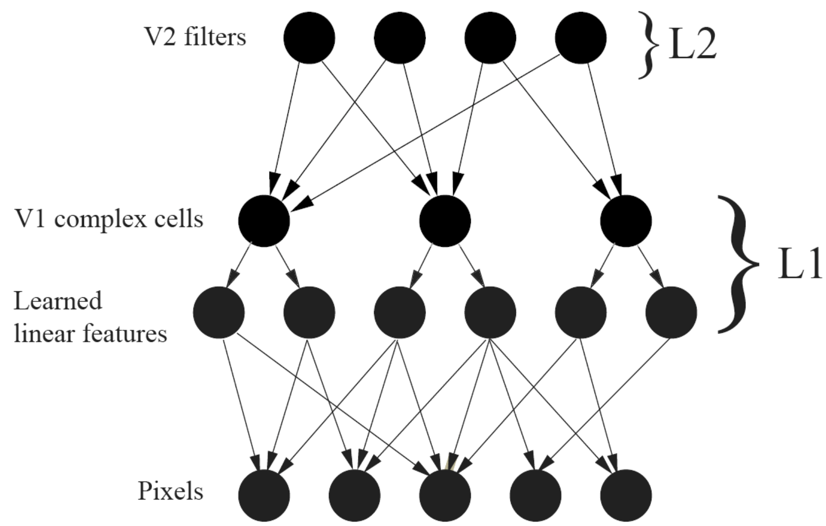

Figure 1: The Markov network representation of our three tiers of feature learning, adapted from H&H.

We begin with a set S of 13 large images. We normalize each image by subtracting the average pixel intensity of the image from every pixel and then normalizing the variance of the image's pixels. More precisely, we perform the following:

$$\forall s \in S\left[s := s - \bar{s}\right]$$

$$\forall s \in S\left[s := \frac{s}{\|S\|_2}\right]$$

We then randomly sample $M/13$ patches of size 16×16 from each image, resulting in M total patches. We represent these patches by a training matrix $X = \left[x^{(1)} \quad \ldots \quad x^{(M)}\right]$.

Finally, we use PCA to whiten X – reducing its dimensionality from 256 to 150. More precisely, we set each $x^{(i)}$ to its projection onto the 150 top eigenvectors of cov(X). This serves to denoise the data and also gives a speed boost to subsequent optimization steps.

Because learning V1 complex cells is a nonlinear problem, we employ an off-the-shelf implementation of Independent Subspace Analysis to perform the learning[1]. We then forward-feed our V1 complex cell responses as training input $\hat{X} = \left[\hat{x}^{(1)} \quad \ldots \quad \hat{x}^{(M)}\right]$ to learn sparse V2 filters. This brings us to our second layer of learning, which turns out to be a rather typical convex optimization problem.

We may formulate our convex optimization problem, penalized in favor of sparseness, as follows:

$$\min_{B,C} \sum_{i=1}^{M} \left(\hat{x}^{(i)} - Bc^{(i)}\right)^2 + \left\|c^{(i)}\right\|_1$$

We optimize this problem in *two stages*, which basically results in a flavor of coordinate descent.

Sparse Coding Stage: Given a matrix of basis vectors B, we learn a sparse representation $c^{(i)}$ for each input response $\hat{x}^{(i)}$ with the fast LSSOL optimizer package[2]. This step theoretically admits for tricks like Q-R factorization[3].

Basis Pursuit Stage: Given a matrix of sparse representations $C = \left[c^{(1)} \quad \ldots \quad c^{(M)}\right]$, we learn a matrix of basis vectors B with a fast Lagrange dual algorithm[4].

Results

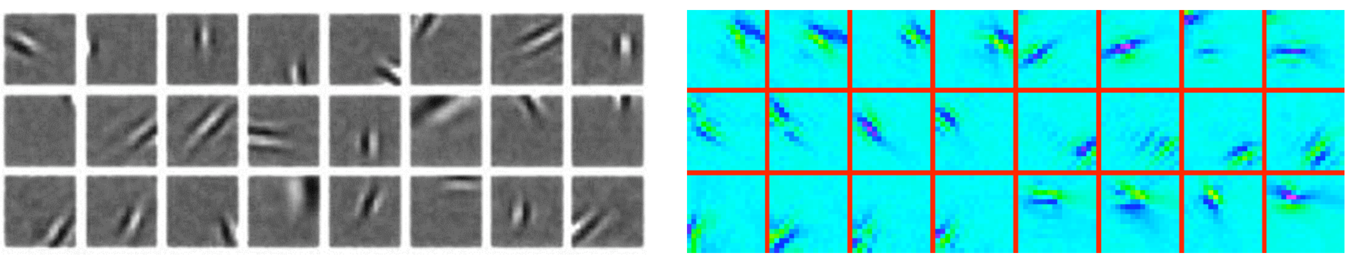

Figure 2: Canonical V1 features from Field and Olshausen (1998) *Left*; Our generated VI features *Right.*
Note the similarity to localized, oriented Gabor filters.

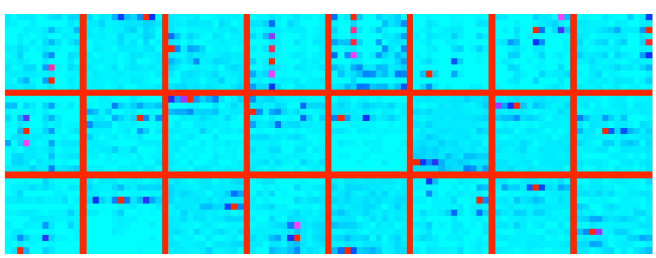

Figure 3: V2 features from non-learned V1
Note the elongated contours.

[1] Thanks to Honglak Lee's recommendation.
[2] Thanks to Professor Michael Saunders.
[3] We tried this but could not yield great results, so we dropped the effort.
[4] Developed by Honglak Lee and Andrew Ng.

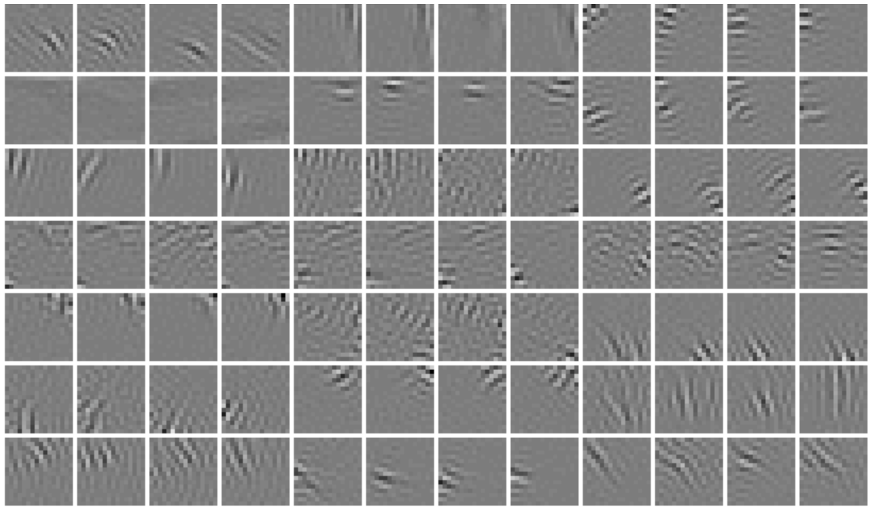

Figure 4: Our closest approximation to good end-to-end learning. We ran ISA on its own tweaked outputs. We found some features that seem characteristic of V2, but many poor ones as well. Note that the problem this methodology solves is not quite equivalent to the goal we specified.

Conclusions

Convex optimization, as always, is a useful paradigm for porting assorted problems into an extremely well-studied domain. By leveraging LSSOL, we were able to improve 3-fold on the speed of a research system, without applying any special domain knowledge to the algorithm. Of course, it is not surprising that we recently learned that Honglak Lee et al have just been able to write a more specialized algorithm that beats us by another factor of 3-10, but we are still pleased with the results.

As for our goal of implementing end-to-end learning from data. Well… we wish that had happened. Want to give us an extension?

Acknowledgements

We owe immense debts to several contributors. Honglak Lee provided Matlab code for basis optimization, and invaluable direction. Professor Michael Saunders showed remarkable patience working with us through the details of various optimization packages. Authors Chen, Donoho, Saunders; Field, Olshausen; and Hoyer, Hyvärinen deserve great credit for making the code used in their papers publicly available.

Sources

T. Serre, L. Wolf and T. Poggio. Object recognition with features inspired by visual cortex. *Proceedings of CVPR '05*. 2005.

P. Hoyer and A. Hyvärinen. A multi-layer sparse coding network learns contour coding from natural images. *Vision Research*, 42(12):1593-1605, 2002.

A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 12(7):1705-1720, 2000.

S. Chen, D. Donoho and M. Saunder. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33-61, 1998.

B. Olshausen and D. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37:3311-3325, 1997.

B. Olshausen and D. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607-609, 1996.