Fast approximate prediction of sparse codes

Karol Gregor, Yann LeCun
Department of Computer Science
New York University
715 Broadway Fl 12,
New York, NY, 10003
karol.gregor@gmail.com

Sparse coding is the problem of reconstructing input vectors using a linear combination of basis vectors with sparse coefficients. Sparse coding has become a popular method for extracting features representations from raw data. Finding the sparse code for a given input involves minimizing a quadratic reconstruction error with an L_1 penalty term. Consequently, a large amount of research has been devoted to efficiently solving this optimization problem [1–3,6–8]. Even so, these algorithms are often too slow for such applications as visual object recognition. Here we propose two versions of a very fast algorithm that produces approximate estimates of the optimal sparse code. While our method only produces approximate solutions, it can be used to compute good visual features, and can be used to initialize exact iterative algorithms. The main idea is to train a non-linear, feed-forward predictor with a particular architecture and a fixed depth to produce the best possible approximation of the sparse code. A version of the method, which can be seen as a trainable version of Li and Osher's coordinate descent method, is shown to produce approximate solutions with 10 times less computation than Li and Osher's for the same approximation error. Unlike previous proposals for sparse code predictors[4,5], the system allows a kind of approximate "explaining away" to take place during inference. The resulting approximator is differentiable and can be included into globally-trainable recognition systems.

The sparse codes are given by minimizing the following energy with respect to Z:

$$E_{W_d}(X, Z) = \frac{1}{2}||X - W_d Z||_2^2 + \alpha||Z||_1 \tag{1}$$

where W_d is an $n \times m$ dictionary matrix whose columns are the (normalized) basis vectors, α is a coefficient controlling the sparsity penalty. The two versions of our algorithm for are derived from two algorithms for obtaining optimal sparse codes, the (F)ISTA ((Fast) Iterative Shrinkage-Thresholding Algorithm)[1,2] and CoD (Coordinate Descent Algorithm)[7]. Both algorithms can be thought of in the framework of the Figure 1. In the simpler case of ISTA, the $W_e = \frac{1}{L}W_D^t$ and $S = I - \frac{1}{L}W_D^t W_D$ are matrices and the nonlinearity is the shrinking function $[h_\theta(V)]_i = \text{sign}(V_i)(|V_i| - \theta_i)_+$. The case of CoD is more complicated; at a given iteration only one code - the most optimal one - is updated. In this algorithm the following is repeated: Pick a coordinate and minimize (1) keeping other coordinates fixed. This has exact, explicit solution. The coordinate is picked that would produce the largest change in Z.

The basic idea of this paper is to truncate these two algorithms at a finite number of iterations and instead of using W_e, S given in terms of W_D (and the α in the shrinkage function given by the sparsity), we learn them, so that they produce the best approximations of the sparse codes given this architecture. To do this we backpropagate the gradients and use the stochastic gradient descent algorithm to train the parameters. At the same time we obtain the gradients with respect to the inputs. This allows these encoders to be included in a globally trained recognition system. We call these algorithms Learned ISTA (LISTA) and Learned CoD (LCoD).

The LISTA algorithm is a sequence of matrix multiplications and nonlinearities and consequently can be easily backpropagated through. In the LCoD we are updating one coordinate at a time. Nevertheless one can backpropagate through it. One only needs to save a small number of variables of the order O((number of iterations)+(dimensionality)), in particular the sequence of coordinates chosen.

We tested the algorithm on the natural image patches. Image patches of size 10×10 were extracted from Berkeley database. The mean was removed and the result was rescaled to have standard deviation one. The W_D in (1) was trained by stochastic gradient descent, where the codes were the optimal codes obtained using CoD. We only show the results for the coordinate descent version here. The result is shown in the Figure 2. We see that LCod gives much better prediction the CoD for small number of iterations. Similarly, we find that LISTA gives better predictions that FISTA.

---

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. *ICASSP'09*, pages 693–696, 2009.

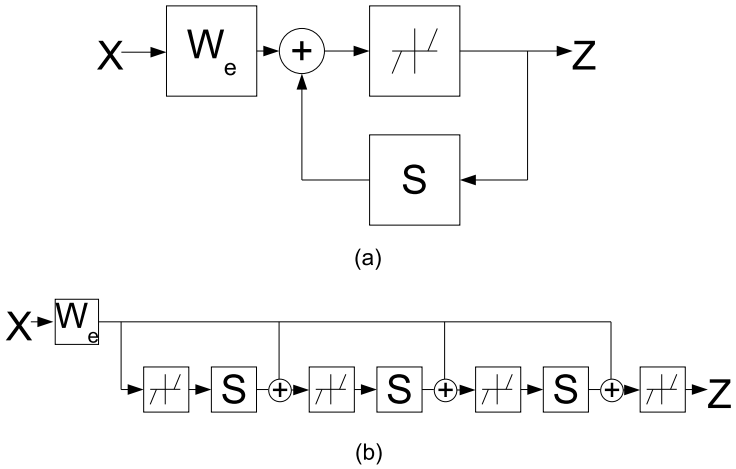

FIG. 1: **(a)** block diagram of the ISTA algorithm for sparse coding. The optimal sparse code is the fixed point of $Z(k+1) = h_\alpha(W_e X - SZ(k))$ where X is the input, h_α is a coordinate-wise shrinking function with threshold α, W_e is the transpose of the dictionary matrix W_d (whose columns are the basis vectors), and S is $W_d^T W_d$. **(b)** The proposed approximator "Learned ISTA", uses a time-unfolded version of the ISTA block diagram, truncated to a fixed number of iterations (3 here). The matrices W_e and S, are learned, so as to minimize the approxmation error to the optimal sparse code on a given dataset. The method allows us to impose restrictions on S so as to further reduce the computational burden (e.g. keeping many terms at 0, or using a low-rank factorized form). Another similar trainable encoder architecture based on the Coordinate Descent algorithm is also proposed.

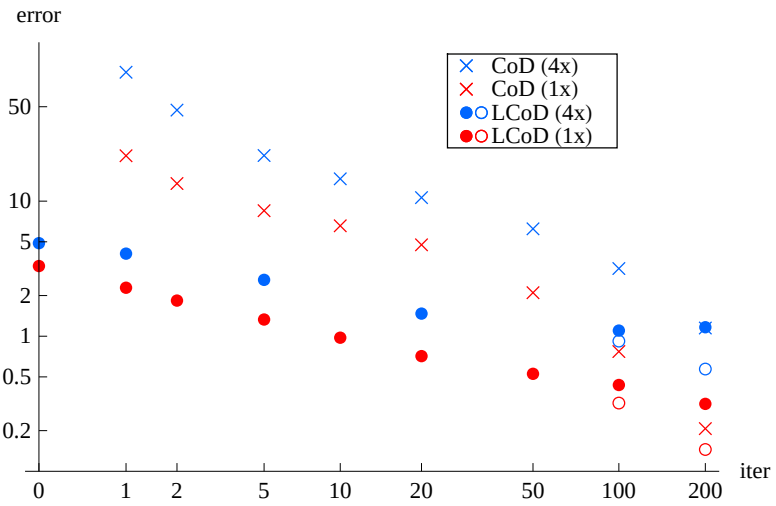

FIG. 2: Code prediction errors for CoD and LCoD for varying numbers of iterations. LCoD is about 20 times faster than CoD for small numbers of iterations. Initializing the matrices with their LCoD values before training (open circles) improve the performance in the high iteration regime, but seems to degrade it in the low iteration regime (data not shown).

[2] I Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Comm. on Pure and Applied Mathematics*, 57:1413–1457, 2004.

[3] E.T. Hale, W. Yin, and Y. Zhang. Fixed-point continuation for l1-minimization: Methodology and convergence. *SIAM J. on Optimization*, 19:1107, 2008.

[4] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In *ICCV'09*. IEEE, 2009.

[5] Koray Kavukcuoglu, Marc'Aurelio Ranzato, and Yann LeCun. Fast inference in sparse coding algorithms with applications to object recognition. Technical Report CBLL-TR-2008-12-01, Computational and Biological Learning Lab, Courant Institute, NYU, 2008.

[6] H. Lee, A. Battle, R. Raina, and A.Y. Ng. Efficient sparse coding algorithms. In *NIPS'06*, 2006.

[7] Y. Li and S. Osher. Coordinate descent optimization for l1 minimization with application to compressed sensing; a greedy algorithm. *Inverse Problems and Imaging*, 3(3):487–503, 2009.

[8] C.J. Rozell, D.H Johnson, R.G. Baraniuk, and B.A. Olshausen. Sparse coding via thresholding and local competition in neural circuits. *Neural Computation*, 20:2526–2563, 2008.