Supplement

Karol Gregor
Janelia Farm, HHMI
19700 Helix Drive
Ashburn, VA, 20147
karol.gregor@gmail.com

Arthur Szlam
The City College of New York
Convent Ave and 138th st
New York, NY, 10031
aszlam@courant.nyu.edu

Yann LeCun
New York University
715 Broadway, Floor 12
New York, NY, 10003
yann@cs.nyu.edu

1 Word tree

We trained the tree structure on frequency of words from NIPS proceedings. For each document we calculate a number of times each word appeared in the document and normalized. We only considered the 8274 most frequent words. The result is shown in the Figure 1. We see that dictionary elements correspond to reasonable NIPS topics and related categories closer in the tree are typically more related.

2 Bilinear structured sparse models

2.1 Patch based parts model

Each of the previous models has, at least cosmetically, a bilinear form. Suppose we have two data sets, X and Y. By setting

$$E(W, A, V, B, S, X, Y) =$$

$$\sum_j ||WA_j - X_j||^2 + ||VB_j - Y_j||^2 + |A_j|^T S|B_j|, \tag{1}$$

we get a model that modulates the sparsity of the representation of $x = X_j$ with the representation of $y = Y_j$ via S. The penalty term $|a|^T S|b|$ is "bilinear" in $b = B_j$ and $a = A_j$, and with a fixed, E reduces to a weighted basis pursuit sparse coding model

$$E_a(V, b, Y) = ||Vb - y||^2 + |b|_\Lambda,$$

where for any vectors c and Λ,

$$|c|_\Lambda = \sum |c_i| \cdot \Lambda_i,$$

and where $\Lambda = |a|^T S$. The equivalent statement holds with b fixed.

Note that equation (2) can be rewritten as

$$\sum_j ||Wz_j - p||^2 + |z_j|^T T|z_j|, \tag{2}$$

where $z_j = [a_j^T b_j^T]^T$,

$$W = \begin{pmatrix} W_1 & 0 \\ 0 & W_2 \end{pmatrix}, \quad p = \begin{pmatrix} x \\ y \end{pmatrix}$$

$$T = \begin{pmatrix} 0 & S^T \\ S & 0 \end{pmatrix}.$$

Thus this model is a special case of the previous ones.

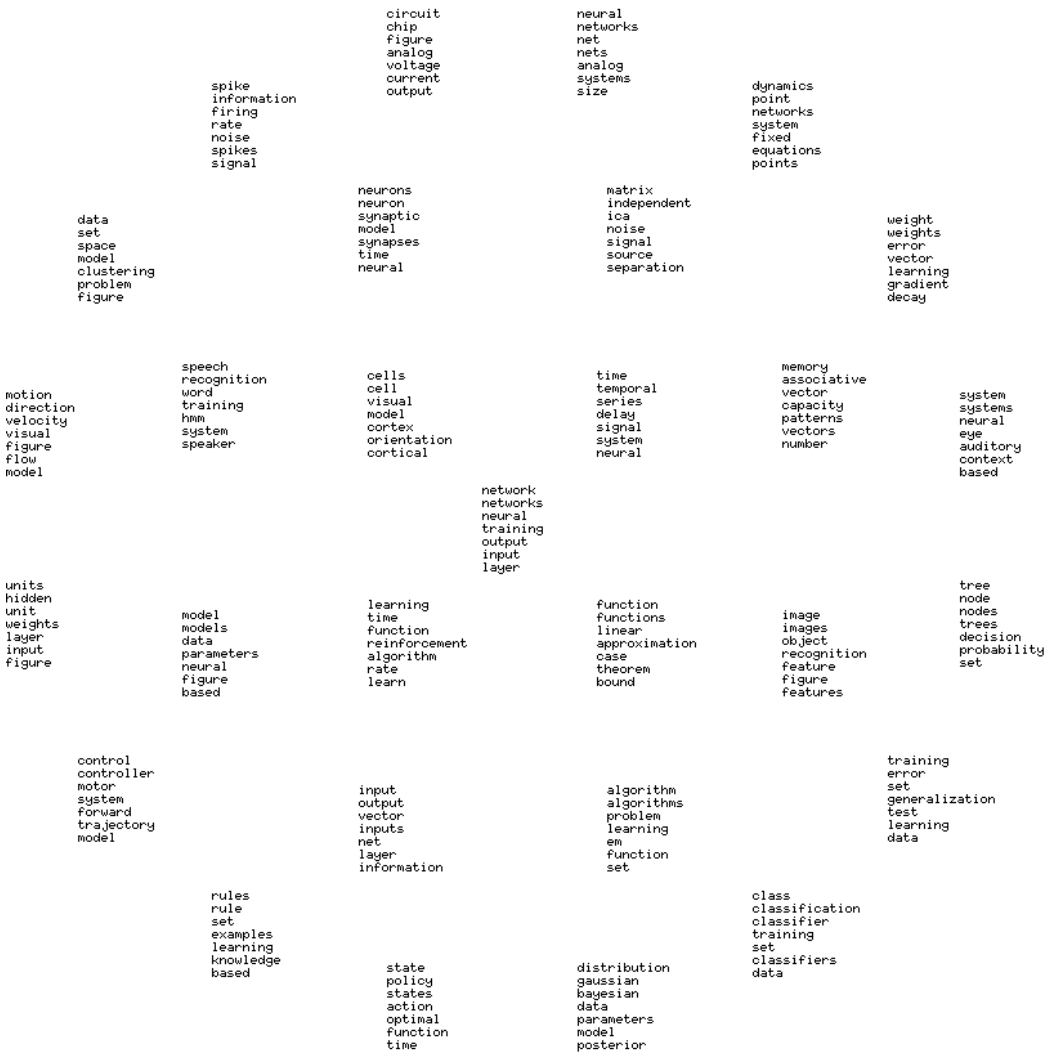

Figure 1: The coding units were placed on a tree. Displayed are the most common words of each dictionary element.

We give an example of learning S in a bilinear setting. We use the centered faces from the faces in the wild dataset, available at `http://vis-www.cs.umass.edu/lfw/`. From each of the 13233 images we extract the center 128×128 pixels, and resize by half. We then pick a random 48×48 window within the 64×64 square. The 48×48 window is broken up into 9 16×16 patches. We remove the mean from each patch, project it to the l_2 unit sphere, and write it as a 256 vector, producing 9 vectors $p_1, ..., p_9$. Finally, we construct the $8 \cdot 256$ vector $x = [p_1^T ... p_4^T \ p_6^T ... p_9^T]^T$ and $y = p_5$; that is, y is the center 16×16 patch and x is the 8 outside patches concatenated. We obtain data matrices X with dimensions $8 \cdot 256 \times 13233$ and Y with dimensions 256×13233. The purpose of the two layers of windowing is to insure that each image contains only parts of faces, but to allow the window to not always center on the same part of the face.

We then train a model minimizing the energy

$$E(W, Z_1, Z_2, S) =$$

$$\sum_{x \in X, \ y \in Y} \left(\frac{1}{8} ||\tilde{W} z_1 - x||^2 + ||W z_2 - y||^2 + z_1^T S z_2 \right),$$

such that W is a 256×400 dictionary (i.e. a dictionary with 400 atoms),

$$\tilde{W} = \begin{array}{c} \\ 256\{ \\ 256\{ \\ \\ 256\{ \end{array} \overset{\begin{array}{cccc} \overbrace{}^{400} & \overbrace{}^{400} & & \overbrace{}^{400} \end{array}}{\begin{pmatrix} W & 0 & \cdots & 0 \\ 0 & W & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & 0 & W \end{pmatrix}},$$

and $Z_1,\ Z_2 \geq 0$, $0 \leq S \leq .01003$, mean$(S) = .01$. The constraint on $\tilde{W}$ and W just forces each patch uses the same dictionary, regardless of where the patch lies on the 3×3 grid.

The energy is minimized via the batch procedure. The updates for Z are done via coordinate descent, the updates for W via least squares, and at each update, S is averaged with .1 of the solution to the linear program with fixed Z and renormalized. W is initialized via random patches from Y, and S is initialized as the all ones matrix. In Figure 2.1 the dictionary W is displayed.

To understand the S which is learned, we will try to use it to generate new images. The model allows the dictionary elements used in the patch in the middle to modulate the dictionary elements used in the reconstruction of each of the outer patches, and the dictionary elements used in the reconstruction of the patches on the outside to modulate the dictionary elements used for the center patch, but it does not directly allow generation of new patches without any data to reconstruct. We will ignore the reconstruction term, but to prevent the model from collapsing, we will specify that the code used to generate each patch has unit sum. In equations, we will try to minimize:

$$E_g(z_1, z_2, S) = z_1^T S z_2,$$

subject to

$$z_1,\ z_2 \geq 0$$

$$\sum_{j=k+1}^{k+400} z_1(j) = 1,\ k \in \{0, 1, ..., 7\}$$

and

$$z_2^T 1 = 1.$$

Note that at the minimum, one element in each block is turned on. We will start with z_2 chosen as uniform random vector normalized to have sum 1, and then alternate between solving for z_2 and z_1 until we hit a local minimum. Results of several runs are shown in Figure 2.1

The interesting thing about this experiment is the fact that no patch ever is allowed to see global information; the dictionary elements used in the reconstruction of the outside patches only get to see the value of the dictionary elements used in the center patch through S, and vice versa. Thus even though W is blind to anything larger than a 16×16 patch, and is the same for every location, the model is able to learn the global structure through S.

3 Learning the S on mean-zero patches

We use the model

$$\min_S \min_{W,Z} \sum_{x \in X} ||Wz - x||^2 + |z|^T S |z|, \tag{3}$$

$$Z \geq 0,\ ||W_j|| = 1\ \forall j,$$

$$0 \leq S \leq \beta,\ S = S^T,\ \text{and} |S_j|_1 = \alpha\ \forall j$$

on 50000 patches 8×8 patches taken from the Pascal dataset. The mean of each patch is removed, but they are otherwise unprocessed. Here, $\beta = 12$ and $\alpha = .13$. Note that these two numbers roughly specify the number of zeros in the solution of the S problem to be 800.

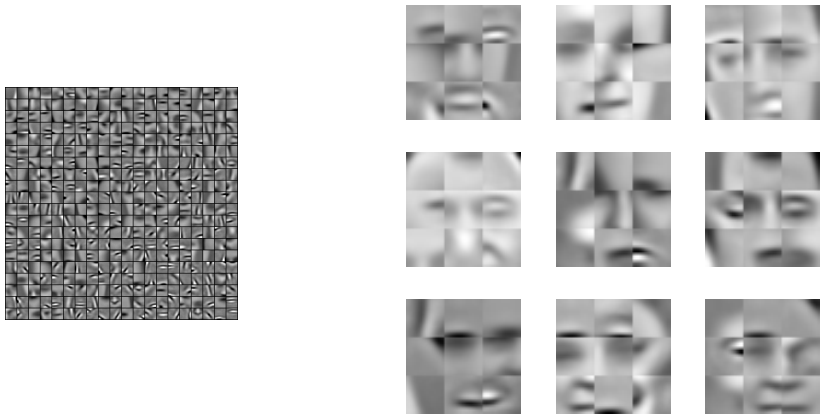

Figure 2: The dictionary W of 16×16 patches learned by the bilinear model on faces; Generated faces. Each patch is forced to have coefficients summing to 1.

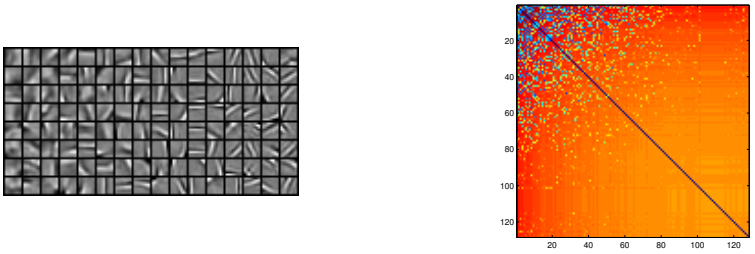

Figure 3: W and S, ordered by the sparsity of the interactions in S, as measured by l_1/l_2 of the row. S is on the right, with blue signifying small inhibition, and red signifying large inhibition; W is on the left, the atoms are arranged column major in the same order as the rows of S. Atoms corresponding to sparse rows of S have many low cost interactions with other elements (and pay for these with other, larger interactions); these tend to be low frequency. The non-sparse rows of S tend to correspond to high frequency atoms; these essentially always incur a cost to activate. Note that all the structure is learned from the data.

4 A Different way of learning the interaction matrix

In the main paper we provided an algorithm for learning the inhibitory interaction matrix S. The difficulty lies in the providing a contrastive term. In this section we give another model where learning is more naturally done by stochastic gradient descent. We write S simply as $S = e^{-P}$ where P is a matrix, exponential is done component-wise. We add a contrastive term proportional to the P^2 to the energy. The model becomes

$$\min_{S,W,Z} \sum_{x \in X} ||Wz - x||^2 + \alpha |z|^T e^{-P} |z| + \beta P^2, \tag{4}$$

$$||W_j|| = 1 \ \forall j.$$

We can understand the behavior as follows. The P_{ij} is pushed down by the L2 term and pushed up in proportion to the frequency and magnitude of co-activity of elements z_i, z_j. Because of the functional form the P will stay bounded and will be larger for units that co-occur together. We trained this system on natural image patches and looked which neighbors a given unit has strongest connections to. We found, in agreement with [GO08] that these are units with similar orientations, Figure 3.

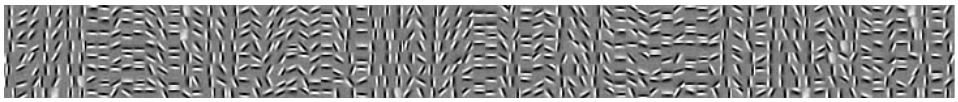

Figure 4: Each column shows the filters of the units that have strongest connections P to the unit with the first filter in the column in the model (4). Units of similar orientations tend do be connected (not inhibited).