Learning Responses of Cells in the Visual Cortex

Shashwat Shukla

A - Motivation

Hubel and Wiesel in 1962, reported their findings from recording spike-train data from neurons in the V1 cortex of monkeys, as they were shown moving bars of light on a screen [1]. Using these experiments they were able to determine the response curves of neurons in this area of the cortex. Their findings led to a search for explanations of the characteristic Gabor-wavelet like responses of Simple cells as well as the translation-invariant responses of Complex Cells and why these were chosen by nature and if they solve are the solution to some specific computational problem. Another relevant question to ask is what kind of learning rules can lead to such responses. We can take cues from the statistics of natural images, the observed connectivity and structure of the cortex, and rely on various theories like the Efficient Coding Hypothesis to constrain our search for learning algorithms.

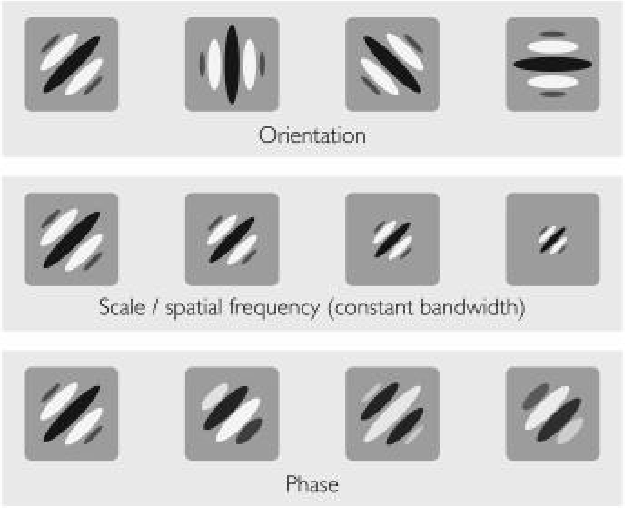
In this project, we explore various methods of learning a sparse representation for patches of natural images. We follow the methodology described in Olshausen and Field's '97 paper [2] in which they propose to learn a sparse, overcomplete representation for patches of natural images. The reason to assume sparsity is that a sparse representation can be compressed and hence supports the efficient coding hypothesis, and also experiments have shown that only a few neurons are firing at any given time, indicating sparsity. The reason for choosing an overcomplete basis is that the connectivity found in the cortex indicates that this is the case, and also because having an overcomplete basis allows for sparser representations (as any given vector can then be decomposed in a larger number ways into basis vectors).
We also explore other methods of inducing sparsity and compare the results we get with the setting in [2].

B - Theory

1. Simple and Complex Cells

Neuroscientists have found that the majority of cells in the V1 (first layer of the visual cortex) cortex can be classified as simple or complex and differ in the their responses. Simple cells respond maximally to a bar of light located in a specific position and at a specific orientation. Complex cells have responses that invariant to small shifts in the position of the bar of light. While there are multiple theories about how such an invariant response can be learnt, the leading theory is that the responses from simple cells with similar orientation preference are pooled together to learn the complex cell response.

The Gabor-like responses of simple cells have been found to vary in scale, orientation and frequency:



2. Relevance Vector Machine

It has previously been reported that Gaussians with variable variance when used as a prior, induce sparsity. For an intuitive explanation, please refer to Chap 7. 7. 2 of Bishop's book on Pattern Recognition. Hence we posit that in our use-case, this choice of prior should also lead to a sparse representation. The question then is if the sparse representation will form receptive fields similar to Gabor wavelets. The prior is given below:

$$\mathbf{X} = \mathbf{\Phi}\boldsymbol{\theta} + \boldsymbol{\xi} \qquad p(\boldsymbol{\theta}) = \prod_{i=1}^{d} \frac{1}{\sqrt{2\pi\lambda_i^{-1}}} e^{-\frac{\lambda_i}{2}\theta_i^2}$$

The following are the steps when the model is learnt using an EM style algorithm.

E-step:

$$\boldsymbol{\theta}_{\text{MAP}} = \arg\max \log p(\boldsymbol{\theta} \mid \mathbf{X})$$

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathbf{X}) &\equiv \log P(\mathbf{X} \mid \boldsymbol{\theta}) + \log p(\boldsymbol{\theta} \mid \lambda) + const. \\ &= -\frac{d}{2}\log 2\pi - \frac{1}{2}(\mathbf{X} - \mathbf{\Phi}\boldsymbol{\theta})'(\mathbf{X} - \mathbf{\Phi}\boldsymbol{\theta}) - \frac{1}{2}\sum_{i=1}^{d}\log 2\pi\lambda_i^{-1} - \sum_{i=1}^{d}\frac{\lambda_i}{2}\theta_i^2 + const\end{aligned}$$

$$\text{Newton-Raphson} \quad \boldsymbol{\theta}^{\text{new}} = \boldsymbol{\theta}^{\text{old}} + [-\nabla\nabla \log p(\boldsymbol{\theta} \mid \mathbf{X})]^{-1} \nabla \log p(\boldsymbol{\theta} \mid \mathbf{X})$$

M-step:

$$\frac{\partial Q(\mathbf{\Phi})}{\partial \mathbf{\Phi}} = \mathbf{X}\boldsymbol{\theta}'_{\mathrm{MAP}} - \mathbf{\Phi}(\mathrm{W} + \boldsymbol{\theta}_{\mathrm{MAP}}\boldsymbol{\theta}'_{\mathrm{MAP}})$$

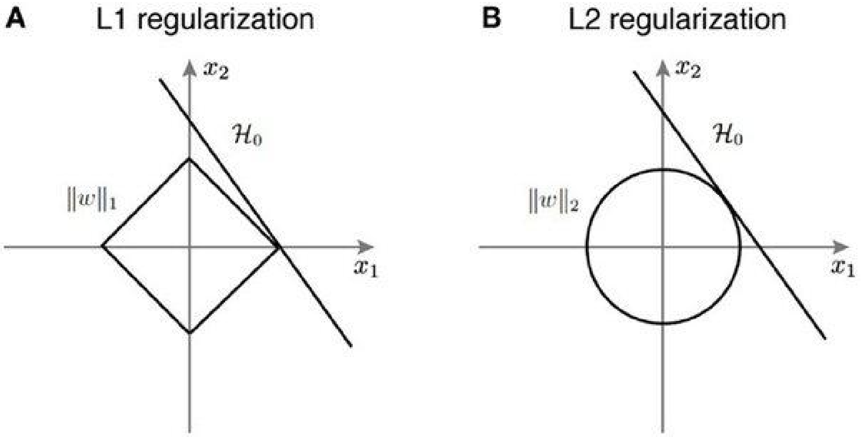
$$\mathrm{W} \equiv -\left[\frac{\partial \log p(\boldsymbol{\theta} \,|\, \mathbf{X}, \lambda)}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'}\right]^{-1}$$

$$\begin{aligned}[\lambda_1^{-1}, \ldots, \lambda_d^{-1}]' &= diag E_{\boldsymbol{\theta}|\mathbf{X}} \boldsymbol{\theta}\boldsymbol{\theta}' \\ &= diag[\mathrm{W} + \boldsymbol{\theta}_{\mathrm{MAP}}\boldsymbol{\theta}'_{\mathrm{MAP}}]\end{aligned}$$

$$\mathbf{\Phi}^{\mathrm{new}} = \mathbf{\Phi}^{\mathrm{old}} + \varepsilon \left.\frac{\partial Q(\mathbf{\Phi})}{\partial \mathbf{\Phi}}\right|_{\mathbf{\Phi}^{\mathrm{old}}}$$

3. Lasso and Ridge Regression

The following figure depicts why L1 penalty induces sparsity but L2 penalty does not. The idea is that a tangent for L1 will typically pass through the vertices which lie on the axis and hence induce sparsity. But the L2 contour has a continuously varying slope for points on an iso-contour and hence the point of intersection of the tangent will not in general pass through points on the axis and the solution will in general be dense. This can also be seen from the fact that L2 norm penalises large values as the square of their magnitude, a much higher penalty than that imposed by L1 and hence large values will be rare and instead many small ie a dense representation will be learnt.



The Conjugate Gradient method was implemented to solve the ridge regression problem efficiently. It is an efficient method of solving Ax = b (which is the form of the equation for ridge regression). The algorithm is:

```
$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
$\mathbf{p}_0 := \mathbf{r}_0$
$k := 0$
repeat
    $\alpha_k := \dfrac{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}$
    $\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$
    $\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$
    if $r_{k+1}$ is sufficiently small, then exit loop
    $\beta_k := \dfrac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}$
    $\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$
    $k := k + 1$
end repeat
The result is $\mathbf{x}_{k+1}$
```

The Coordinate Descent Algorithm was implemented to solve the Lasso problem. The corresponding equations are:

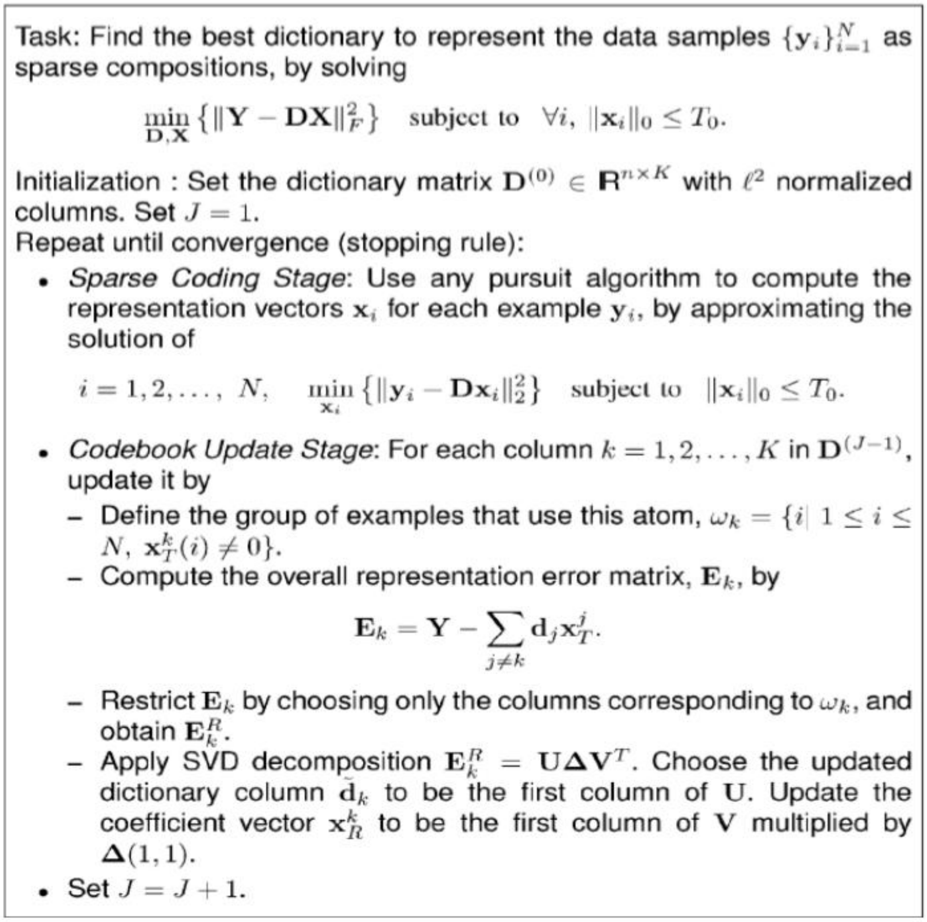
$$f(\beta) = \frac{1}{2n}\|\mathbf{y} - \mathbf{X}\beta\|_2^2 + \lambda\|\beta\|_1$$

$$\begin{aligned}\arg\min_{\beta_k} f(\beta) &= S\big(\frac{\langle \mathbf{r}_k, \mathbf{x}_k\rangle}{\|\mathbf{x}_k\|^2}, \frac{\lambda n}{\|\mathbf{x}_k\|^2}\big) \\ &= \frac{1}{\|\mathbf{x}_k\|^2} S\big(\langle \mathbf{r}_k, \mathbf{x}_k\rangle, \lambda n\big).\end{aligned}$$

This minimisation procedure is done for every coordinate in one iteration and then we iterate until convergence.

4. K-SVD and MOD

K-SVD is a dictionary learning algorithm for creating a dictionary for sparse representations, via a singular value decomposition approach. K-SVD is a generalization of the k-means clustering method, and it works by iteratively alternating between sparse coding the input data based on the current dictionary, and updating the atoms in the dictionary to better fit the data. The update of the dictionary columns is combined with an update of the sparse representations, thereby accelerating convergence. The K-SVD algorithm is flexible and can work with any pursuit method. We use OMP(Orthogonal Matching Pursuit) for this. Then we decompose the original error matrix into a sum of rank-1 matrices, out of which only the last term depends on the dictionary column to be updated. So we are trying to find a rank-1 approximation for the error matrix, and this can be done by computing the SVD of Ek, and using the singular vectors corresponding to the largest singular value. In order to ensure sparsity, which in not guaranteed inbuilt by the SVD, we considers only those data-points that actually use the k-th dictionary atom, effectively yielding a smaller matrix on which the SVD is performed. Though K-SVD convergence is not fully guaranteed and is susceptible to local minima and overfitting, it has been reported to work well for most of the applications.



Task: Find the best dictionary to represent the data samples $\{\mathbf{y}_i\}_{i=1}^{N}$ as sparse compositions, by solving

$$\min_{\mathbf{D},\mathbf{X}} \left\{ \|\mathbf{Y} - \mathbf{D}\mathbf{X}\|_F^2 \right\} \quad \text{subject to} \quad \forall i, \ \|\mathbf{x}_i\|_0 \le T_0.$$

Initialization : Set the dictionary matrix $\mathbf{D}^{(0)} \in \mathbf{R}^{n \times K}$ with ℓ^2 normalized columns. Set $J = 1$.
Repeat until convergence (stopping rule):

- *Sparse Coding Stage*: Use any pursuit algorithm to compute the representation vectors $\mathbf{x}_i$ for each example $\mathbf{y}_i$, by approximating the solution of

  $$i = 1, 2, \ldots, \ N, \quad \min_{\mathbf{x}_i} \left\{ \|\mathbf{y}_i - \mathbf{D}\mathbf{x}_i\|_2^2 \right\} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \le T_0.$$

- *Codebook Update Stage*: For each column $k = 1, 2, \ldots, K$ in $\mathbf{D}^{(J-1)}$, update it by
  - Define the group of examples that use this atom, $\omega_k = \{i|\ 1 \le i \le N,\ \mathbf{x}_T^k(i) \ne 0\}$.
  - Compute the overall representation error matrix, $\mathbf{E}_k$, by

    $$\mathbf{E}_k = \mathbf{Y} - \sum_{j \ne k} \mathbf{d}_j \mathbf{x}_T^j.$$

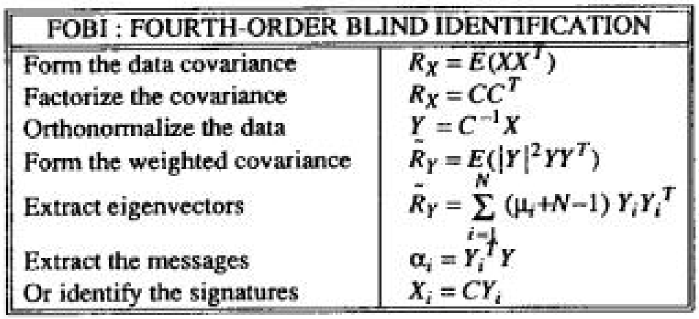
  - Restrict $\mathbf{E}_k$ by choosing only the columns corresponding to ω_k, and obtain $\mathbf{E}_k^R$.
  - Apply SVD decomposition $\mathbf{E}_k^R = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}^T$. Choose the updated dictionary column $\tilde{\mathbf{d}}_k$ to be the first column of $\mathbf{U}$. Update the coefficient vector $\mathbf{x}_R^k$ to be the first column of $\mathbf{V}$ multiplied by $\boldsymbol{\Delta}(1,1)$.
- Set $J = J + 1$.

The core idea of MOD it is to solve the minimization problem subject to the limited number of non-zero components of the representation vector: Minimize the frobenius norm of the error matrix subject to the constraint of sparsity on he coefficient vector for each datapoint. MOD alternates between getting the sparse coding using a method such as matching pursuit and updating the dictionary by computing the Moore-Penrose pseudoinverse. After this update the dictionary columns are renormalized to fit the constraints and the new sparse coding is obtained again. The process is repeated until convergence (or until a sufficiently small residue). MOD has proved to be a very efficient method for low-dimensional input data requiring just a few iterations to converge. However, due to the high complexity of the matrix-inversion operation, computing the pseudoinverse in high-dimensional cases is in many cases intractable.

5. PCA and ICA

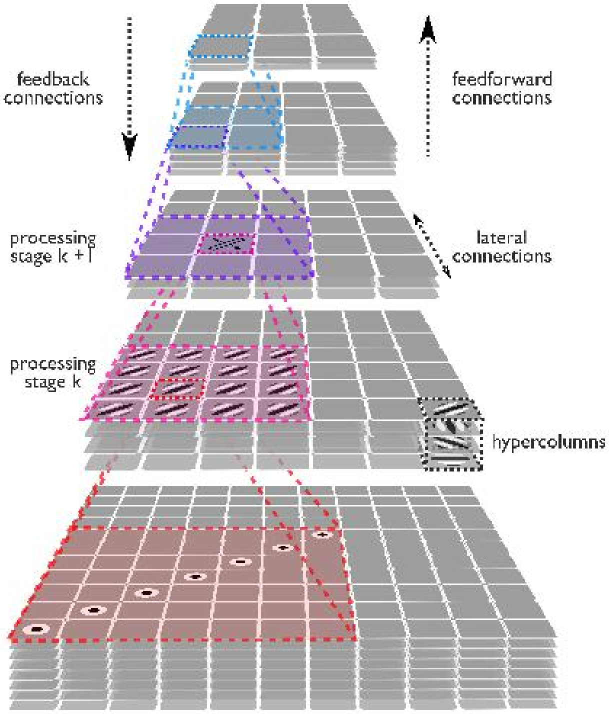
Principal Component Analysis (PCA) looks for a change of basis that whitens the data (so that the covariance matrix becomes an identity matrix). This is equivalent to looking for independent and orthogonal projections of the data if a Gaussian noise model is assumed (because for Gaussian vectors, uncorrelated is equivalent to being independent). The transformed vectors are orthogonal to each other.

With Independent Component Analysis, this constraint of orthogonality is relaxed and we look for independent components that are not necessarily orthogonal to one another. But true statistical independence can only be estimated from finite data with some model of the distribution in mind. And this has led to many different estimates for statistical independence and hence many corresponding ICA algorithms. fastICA is one of the most used ICA algorithms, and it uses KL divergence as the metric of independence and minimizes a cost function using gradient descent. We chose to instead implement FOBI-ICA (Fourth Order Blind Identification) ICA [3] which uses kurtosis (4th order moment) as a measure of statistical independence. The advantage of doing this is that there exists an analytic closed form solution for the FOBI-ICA model that can be computed very efficiently using matrix computations.
The FOBI algorithm is:



FOBI : FOURTH-ORDER BLIND IDENTIFICATION	
Form the data covariance	$R_X = E(XX^T)$
Factorize the covariance	$R_X = CC^T$
Orthonormalize the data	$\underset{\sim}{Y} = C^{-1}X$
Form the weighted covariance	$\tilde{R}_Y = E(\lvert Y \rvert^2 YY^T)$
Extract eigenvectors	$\tilde{R}_Y = \sum_{i=1}^{N} (\mu_i + N - 1)\, Y_i Y_i^T$
Extract the messages	$\alpha_i = Y_i^T Y$
Or identify the signatures	$X_i = CY_i$

6. Learning Responses of Complex Cells



In [4] they attempt to learn the transformation and the basis vectors jointly. The transformations are assumed to be linear and small in magnitude so that a first order approximation is valid. They learn Gabor-like filters, but the the translations considered are just 2-5 pixels and hence are very small. In [5], they follow a similar approach, modelling it as a bilinear model, with the first linear transform representing projection onto the basis vectors and the second one being used to account for translations, rotations etc.

In [6] the authors work with videos captured by mounting a camera on a cat's head and propose a hierarchical structure in which the middle layer neuron learn responses similar to simple cells while the top layer learns response close to that of complex cells. They propose a Hebbian learning rule inspired by Long Term Potentiation (LTP) and Long Term Depression (LTD).The middle layer incorporates a winner-take-all learning rule. In [7] they propose a learning rule that imposes a sparseness penalty as well as a slowness prior to learn a valid topographic map. This prior makes this method very similar to Slow Feature Analysis. In [8] the authors explicitly model the dynamics of neurons as Leaky Integrate and Fire neurons and the learning rule mimics STDP (Spike-Timing Dependent Plasticity). The various time constants used are critical to this model and can be seen as a spiking version of SFA.

In [11] the authors model the problem of learning invariance as finding a subspace in which the features are invariant to spatial shifts of the image and hence model this as a problem of Independent Subspace Analysis.

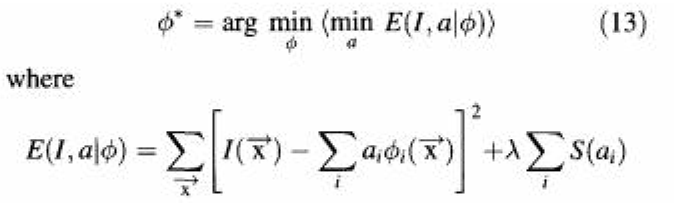
C - Methodology

The method we propose to learn the Gabor-wavelet like receptive field of simple cells is:

1) Take a dataset of 12x12 image patches extracted from natural images.
2) Whiten the patches
3) Initialise 100 (or more) basis vectors randomly.
4) The cost function is reconstruction error penalised by some sparsity inducing additive term. Note that the image is reconstructed as a linear weighted sum of the basis vectors
5) Minimize the cost function using an EM-style algorithm, learning the optimal basis vectors in the process.
6) Various penalties for sparsity tested

   Ridge
   Lasso
   RVM

   We also tried other methods of learning vectors: PCA, ICA, K-SVD, MOD
7) Verify that the learnt vectors look like Gabor wavelets.
8) Generate a large bank of Gabor wavelets and fit the learn basis vectors to these, and study the distribution of these wavelets in frequency, phase etc.
9) Verify that the representation obtained is sparse



$$\phi^* = \arg \min_{\phi} \langle \min_{a} E(I, a|\phi) \rangle \qquad (13)$$

where

$$E(I, a|\phi) = \sum_{\vec{x}} \left[I(\vec{x}) - \sum_i a_i \phi_i(\vec{x}) \right]^2 + \lambda \sum_i S(a_i)$$

Here S(a) is the sparsity penalty and phi is the set of learnt basis vectors

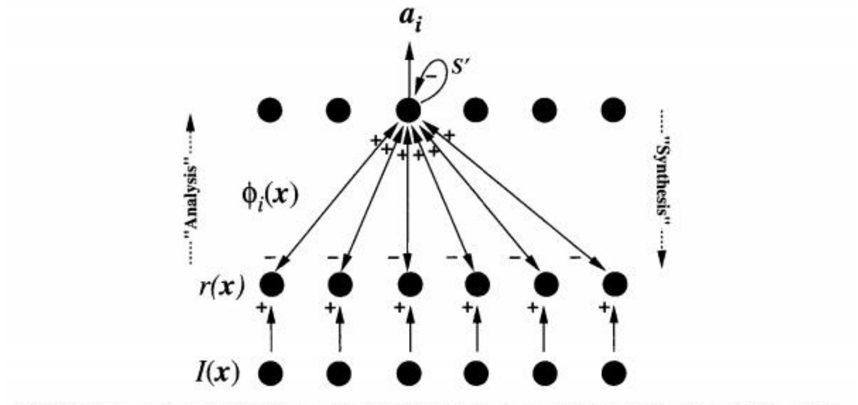


FIGURE 5. A network implementation for computing the a_i. Each output unit represents the value of a single coefficient, a_i. The output activities are fed back through the basis functions, ϕ_i, to form a reconstruction of the image. The reconstructed image is then subtracted from the input image, and the residual image is fed forward through the ϕ_i to drive each output, a_i, which is also being self-inhibited by S'. This process is analogous to the analysis–synthesis loop proposed by Mumford (1994) for performing inference on images. Learning is accomplished by doing a Hebbian update of the ϕ_i based on the average joint activity between the outputs (a_i) and the residual image computed via the negative feedback connections.

(From Olshausen and Field,1997)

D - Implementation Details

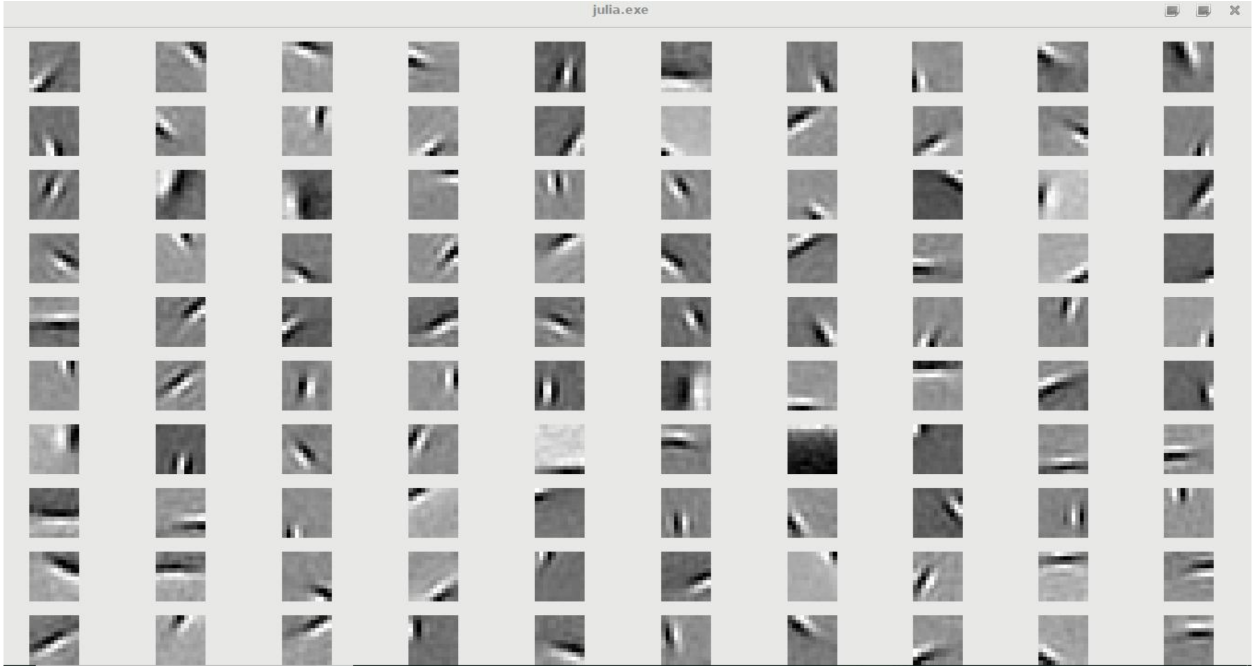
We are implementing all our algorithms from scratch, only using libraries to perform operations like matrix multiplication, finding inverses and eigenvalues, reading and writing to files etc.

We have implemented Automatic Relevance Determination for Sparse Bayesian Learning in Julia. Conjugate Descent was implemented for the choice of Gaussian prior (L2 penalty).
Coordinate Descent was implemented for LASSO for a Laplacian prior (L1 penalty).
PCA, ICA, K-SVD and MOD were also implemented from scratch.
We have also written code in Julia to generate and fit Gabor wavelets (along with some statistics) to the learnt filters.
We have also done an extensive literature review of how complex cell responses can be learnt.

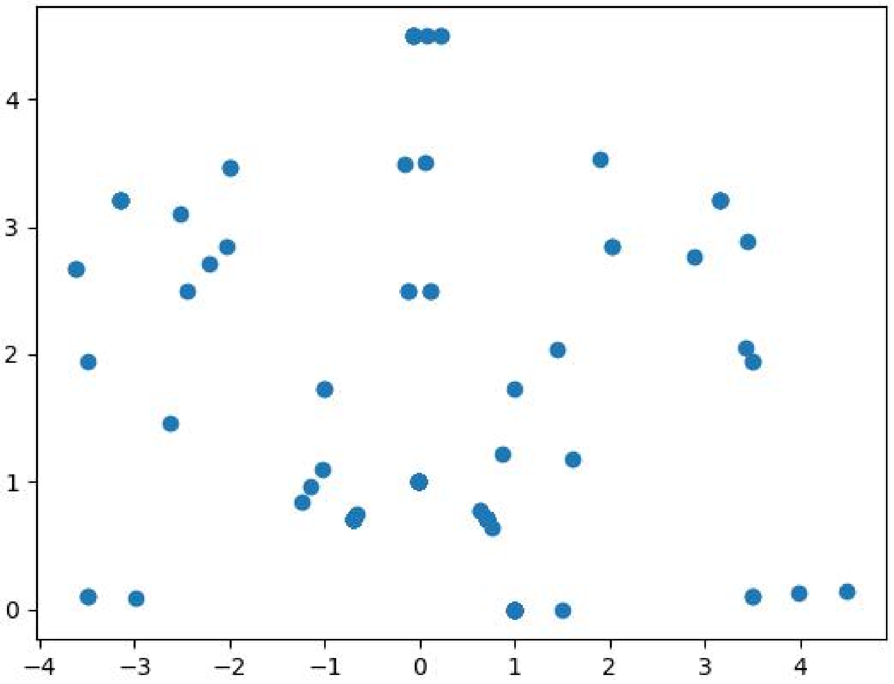
E - Experiments and Results

The dataset of (whitened) natural images used can be found at:
http://redwood.berkeley.edu/bruno/sparsenet/

As expected, imposing an L2 penalty leads to a dense representation and the learnt filters do not in any way resemble Gabor wavelets.
Below are the learnt filters using Automatic Relevance Determination. 400,000 iterations with a learning rate of 0.001 was used:

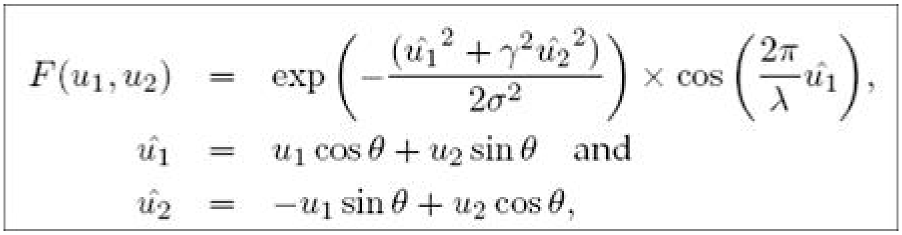


After fitting Gabor wavelets to these learnt filters, their spatial frequencies are plotted below:



X and Y axis respectively are the f_x and f_y, the spatial frequencies along X-axis and Y-axis

The Gabor wavelets used for fitting are based on Physiological parameters as reported in "Representation of higher-order statistical structures in natural scenes via spatial phase distributions" by MaBoudi et al. (http://dx.doi.org/10.1016/j.visres.2015.06.009)

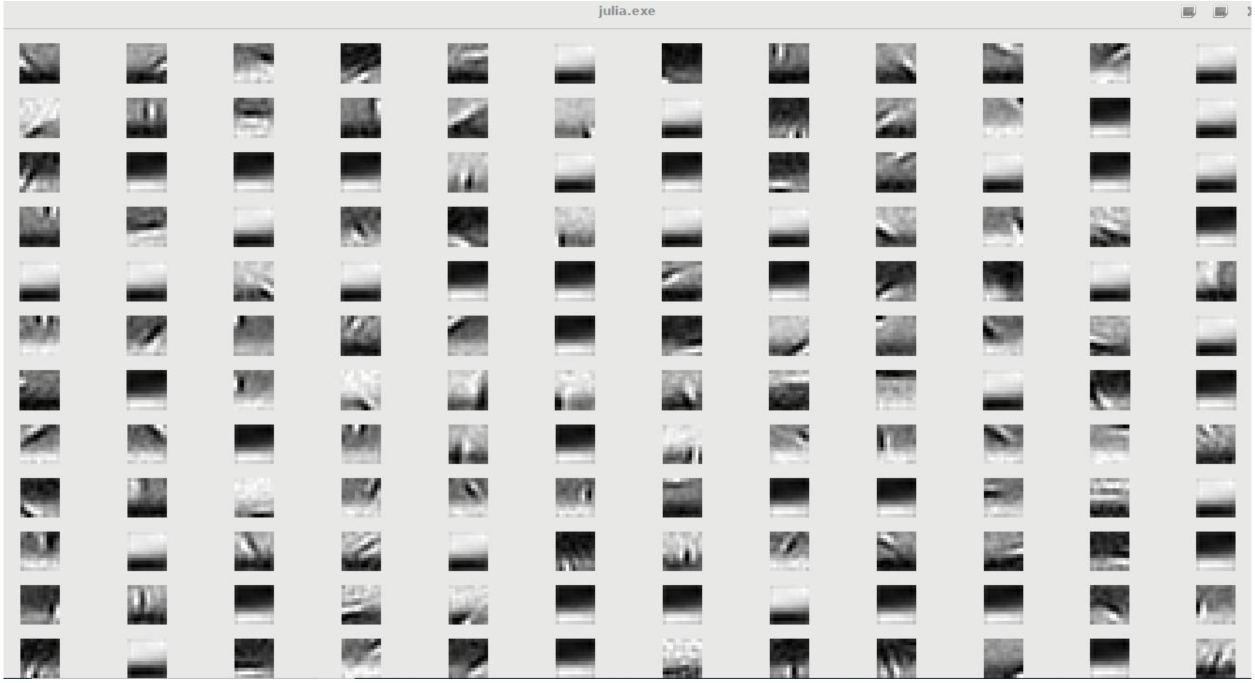


$$
\begin{aligned}
F(u_1, u_2) &= \exp\left(-\frac{(\hat{u_1}^2 + \gamma^2 \hat{u_2}^2)}{2\sigma^2}\right) \times \cos\left(\frac{2\pi}{\lambda}\hat{u_1}\right), \\
\hat{u_1} &= u_1 \cos\theta + u_2 \sin\theta \quad \text{and} \\
\hat{u_2} &= -u_1 \sin\theta + u_2 \cos\theta,
\end{aligned}
$$

Equation for a general Gabor wavelet

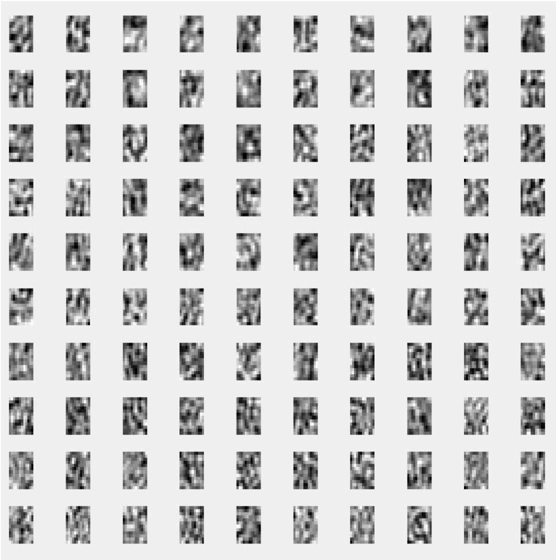
As per the above paper, after accounting for re-scaling based on image-patch size considered, we generated a total of 100000 Gabor Wavelets with parameters as:
σ_ = 0.375^2 * [1,0.5,0.25,0.125, 0.0625]
ω_= [1,1.5,2.0,2.5,3.0,3.5,4.0,4.5]
ψ_ = [-pi/2,-pi/4,0,pi/4,pi/2]
ab_ = [[3,9],[3,3],[6,6],[9,9],[9,3]] # Five positions for a 12x12 image
ρ = 0.60
α_ = pi * linspace(0,1,100)

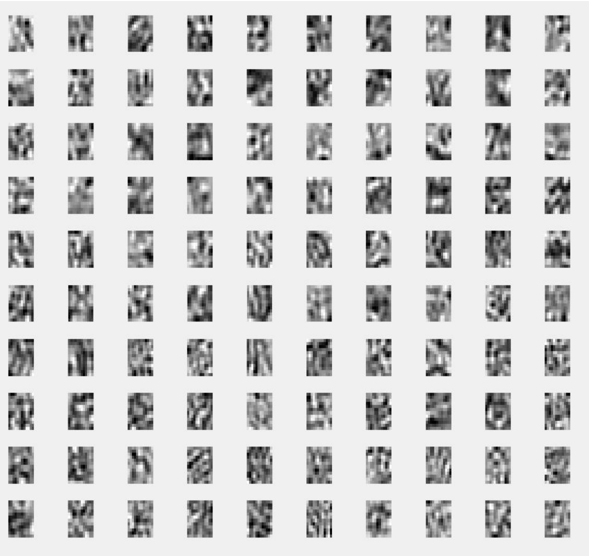
Similar results are also obtained on Lasso:



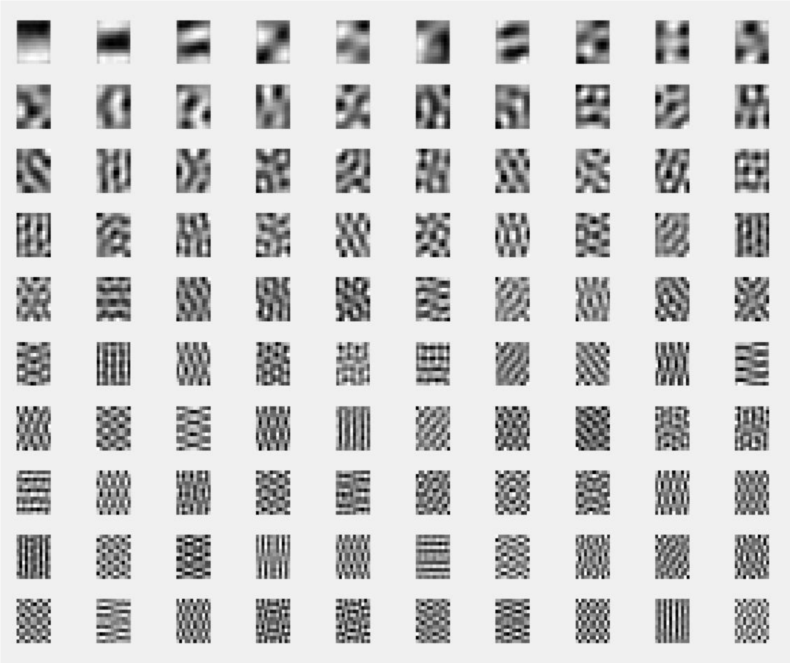
Note that here we have learnt 144 filters instead of 100 as above. The fact that many of the filters are almost identical leads us to conclude that 144 is greater than the “effective” rank of the natural images dataset.

The results on K-SVD and MOD are not great, but we can say that there is some kind of local features being learnt. Following are the basis vectors learnt by K-SVD and MOD method. As we can see, the results are better than PCA but worse than ICA. K-SVD and MOD try to learn globally optimum representation of the data rather than the local features like edges in different directions.

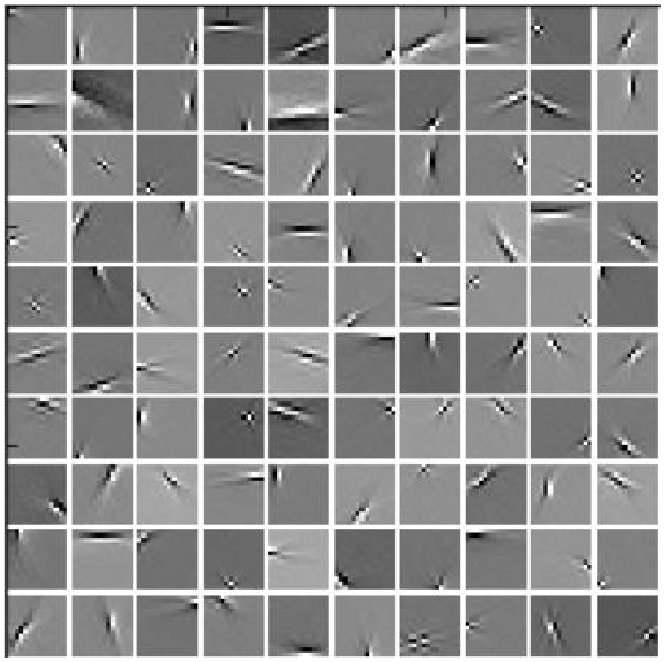

K-SVD


MOD

As expected, the results using PCA does not need to local Gabor wavelets being learnt as PCA learns globally optimum orthogonal features and sparsity is not induced in any way.The checkerboard pattern is clearly visible at high frequencies, as expected.



The results on ICA are very telling. The learnt filters are indeed Gabor like. ICA looks for independent components and the fact that they turn out to be the Gabor filters indicates that learning independent filters is a good strategy to encode information efficiently. As per the paper by Olhausen and Field, their method is equivalent to ICA when there is no noise and when a complete (instead of overcomplete) basis is desired.



F - Conclusions and Future Work

We successfully learnt a sparse and overcomplete basis using three methods: Relevance Vector Machines, Lasso and ICA. We also tested other methods like Ridge Regression, PCA, K-SVD and MOD. We also fit Gabor wavelets to the learnt basis vectors and found that there is a good match between empirically observed responses of simple cells and the vectors that we learnt. All algorithms were implemented from scratch.
We also studied literature on how responses of complex cells can be learnt.

In the future, it would be interesting to work with more general, hierarchical models like Predictive coding. Furthermore, introducing temporal dynamics in the learning process would be very important.

G - References

1. D. H. HUBEL AND T. N. WIESEL, RECEPTIVE FIELDS AND FUNCTIONAL ARCHITECTURE OF MONKEY STRIATE CORTEX, J. Physiol. (1968), 195, pp. 215-243
2. BRUNO A. OLSHAUSEN, DAVID J. FIELD, Sparse Coding with an Overcomplete Basis Set: A Strategy Employed by V1 ?, Vision Res., Vol. 37, No. 23, pp. 3311-3325, 1997
3. Jean-Francois Cardoso, Source separation using higher order moments, Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989
4. Rajesh P N Rao and Dana H Ballard, Development of localized oriented receptive fields by learning a translation-invariant code for natural images, Network: Comput. Neural Syst. 9 (1998) 219–234
5. David B. Grimes and Rajesh P. N. Rao, Bilinear Sparse Coding for Invariant Vision, Neural Computation 17, 47–73 (2005)
6. Wolfgang EinhaÈuser, Christoph Kayser, Peter KoÈnig and Konrad P. KoÈrding**,** Learning the invariance properties of complex cells from their responses to natural stimuli, European Journal of Neuroscience, Vol. 15, pp. 475-486, 2002
7. Wentao Huang, Zhengping Ji, Steven P. Brumby, Garrett Kenyon and Luis M. A. Bettencourt, Development of Invariant Feature Maps via A Computational Model of Simple and Complex Cells, WCCI 2012 IEEE World Congress on Computational Intelligence June, 10-15, 2012
8. Tao Louis and Cai David, Computational modeling of the dynamics of simple and complex cells in primary visual cortex, Acta Physiologica Sinica, October 25, 2011, 63(5): 401−411
9. Michal Aharon; Michael Elad; Alfred Bruckstein (2006), "K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation", IEEE Transactions on Signal Processing, 2006
10. Engan, K.; Aase, S.O.; Hakon Husoy, J. (1999-01-01). "Method of optimal directions for frame design". 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing, 1999
11. Aapo Hyv¨arinen, Urs K¨oster, Complex Cell Pooling and the Statistics of Natural Images
12. Rajesh P N Rao and Dana H Ballard, Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects, 1999 Nature America