Supplementary Material: Unsupervised learning models of primary cortical receptive fields and receptive field plasticity

Andrew Saxe, Maneesh Bhand, Ritvik Mudur, Bipin Suresh, Andrew Y. Ng
Department of Computer Science
Stanford University
{asaxe, mbhand, rmudur, bipins, ang}@cs.stanford.edu

1 Modeling approach

1.1 Learning algorithms

Independent component analysis (ICA) The ICA algorithm has been applied successfully to modeling V1 simple cell receptive fields [1, 2]. It is closely related to sparse coding methods, and can be cast in terms of a simple generative model [3]: We suppose that our data $x \in \mathbb{R}^n$ is an unknown linear mixture of independent, non-Gaussian sources, i.e. $x = As$ where $A \in \mathbb{R}^{n \times n}$ is unknown. During learning, ICA adapts the weights to unmix these sources. At the end of learning the recovered weights are the pseudoinverse of the mixing matrix A, and the model neuron activities are the values of the independent sources s. We use the standard "FastICA" Matlab implementation [4] which we modified to run on GPU hardware, and we use the log-cosh nonlinearity. The basic ICA method itself has no parameters, and the only parameter we consider here is the number of principal components of the data kept before running the algorithm.

Sparse autoencoder (SAE) The sparse autoencoder is an artificial neural network with a single hidden layer trained to reproduce its input at its output, as described in [5]. We constrain the hidden unit activations to be sparse. The autoencoder must therefore compress the structure present in the input data into this sparse representation. In our implementation we use hyberbolic tangent activation functions and adapt the weights using backpropagation to minimize the sum-squared error between input examples and reconstructions. To impose sparsity, we use the method of [6] and constrain the expected activation of the hidden units to be sparse. Our implementation has five parameters: the number of hidden units; the sparsity level or target mean activation of the hidden units; the strength of weight decay; the learning rate used in stochastic gradient descent; and the number of principal components retained.

Restricted Boltzmann machine (RBM) The RBM is an undirected graphical model that learns a generative model of the input data [7]. During learning, the weights are adjusted to approximately maximize the log-likelihood of the data. Our implementation makes use of the standard contrastive divergence algorithm, and we again constrain the hidden units to be sparse. We use Gaussian visible units, and Bernoulli hidden units. has six parameters: the number of hidden units; the target mean activation; the strength of weight decay; the learning rate used in contrastive divergence updates; a scale factor controlling the variance in visible unit activations; and the number of principal components retained.

K-means The K-means algorithm is a method for clustering data that iteratively updates a number of cluster centroids by averaging the data points nearest to each centroid. We consider two parameters: the number of cluster centroids (comparable to the number of hidden units in the other algorithms), and the number of principal components retained.

Sparse Coding The sparse coding algorithm has been applied successfully to modeling V1 simple cells [8]. It alternates between solving two minimization problems, first to obtain a set of sparse neuron activities that represent the input, and second to improve the dictionary of receptive fields. Although the encoding process is nonlinear, here we only analyze the linear component of the response, which was shown by [8] to be very close to the learned dictionary weights. Our implementation has two parameters: the number of hidden units, and a weight controlling the tradeoff between sparsity and reconstruction error in the optimization objective.

1.2 Audio data processing details

We transform the raw sound waveform into a representation of its frequency content over time meant to approximate the response of the cochlea [9]. In particular, we pass the input sound signal to an equivalent rectangular bandwidth (ERB) gammatone filterbank with log-spaced center frequencies, which approximates auditory nerve fiber responses [9, 10]. Our filterbank consists of 64 center frequencies ERB-spaced from 300–8000 Hz. The energy of the filter responses is then summed within fixed time-bins at regular intervals, yielding a representation akin to a spectrogram. For each frequency the energy of the filtered signal is then integrated within fixed time-bins of 25ms, at intervals of 10ms. Tests with linearly spaced center frequencies yielded similar results. After rescaling the datasets to have the same mean amplitude, we randomly extract 250 ms segments from the gammatonegrams of the sounds. Our auditory training data comprises 50,000 samples from each dataset. We used the Gammatonegram toolbox [11].

To eliminate the possibility that our results are an artifact of the adaptive stimulus procedure employed in O'Connor et al., we replicated their procedure and used it to recover the preferred amplitude spectra from our model neurons. Because the results were similar, and the optimization is computationally expensive, we used a simple average over time to estimate the prefered amplitude spectrum for our analyses.

1.3 Touch dataset acquisition and processing details

To extract the contact point information, the blue glove was first extracted from the green background using a simple RGB filter. Grasps were made with both right and left hands, with left-hand pictures being mirrored later in software. The images were then cropped to the hand and aligned to one another. Next we built a mask to separate image pixels on the glove from those in the background. To generate the mask, we found the mean of all aligned images and used an RGB threshold to identify all background pixels. Next a neutral image was used to capture the true color of the blue glove without any powder on it. This image is then used as a reference to detect the white powder on the gloves. The mask is then applied to each contact point image and this forms the raw data for our learning algorithms. t test of this prediction.

We exactly replicated the smoothing and thresholding of the receptive fields used in [12], and restricted our analysis to the fingertip areas of our bases, although we did not restrict the analysis to digit 2 alone due to the limited number of bases learned by our algorithms.

1.4 Parameter fitting

The metric used to determine the best-fitting vision and somatosensory parameters was the mean of the L1 distances between the data histograms and the model histograms, each normalized to sum to one. For the receptive field shape data in V1, since the data lie roughly on a one dimensional line, we projected the values onto the best fit line for the experimental data and computed a histogram of the resulting one dimensional positions.

Because the audio fits were not similar histograms, and instead varied widely in scale, we determined the best fit by scaling each error metric by its standard deviation over the set of models we trained. We selected the parameters that minimized the sum of these normalized scores. This procedure was meant to approximately put equal weight on each comparison.

The figures below show our quantitative fit metric as a function of hyperparameters for two example algorithms. The complete results of our gridsearches are available at `http://www.stanford.edu/~asaxe/rf_plasticity.html`.

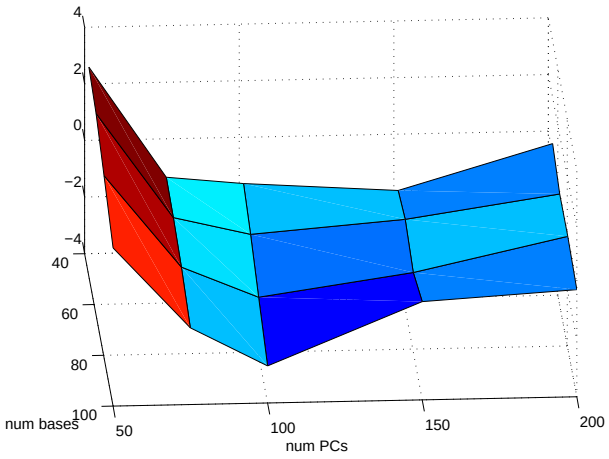

Figure 1: Z-scores v/s number of bases and prinicipal components used in training K-means experiments on our audio dataset

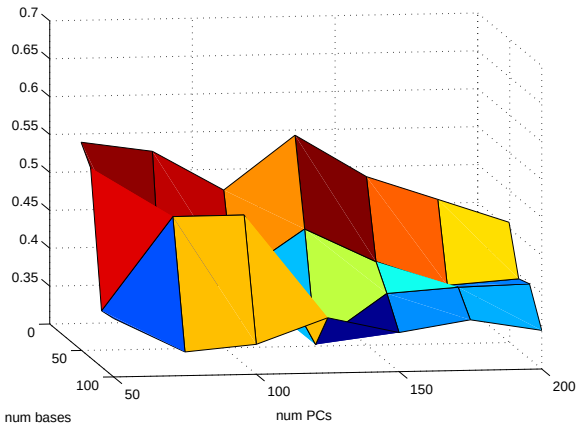

Figure 2: L1 scores (squared sum of L1 distance to DiCarlo's histograms [12]) v/s number of bases and prinicipal components used in training Sparse-coding experiments on our touch dataset

1.4.1 Cross-validation experiment

To check the degree to which our parameter fitting methodology is overfitting to the data, we found the parameters that achieved the best fit for one modality and applied it to another. Below are three example plots from the results. Overall, the qualtitative pattern of results remains largely unchanged, and algorithms with fewer parameters (ICA, K-means, and sparse coding) tended to fare better.

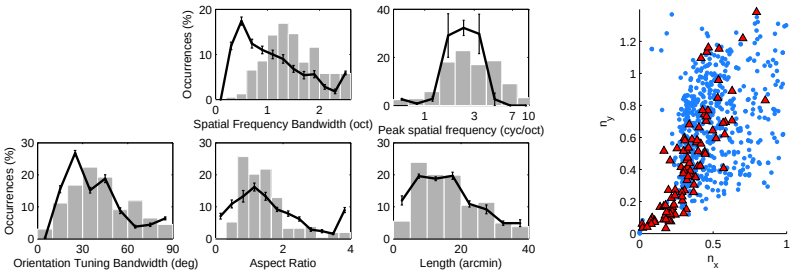

Figure 3: Fit to data in visual cortex using the best somatosenosry parameters, here for the K-means algorithm.

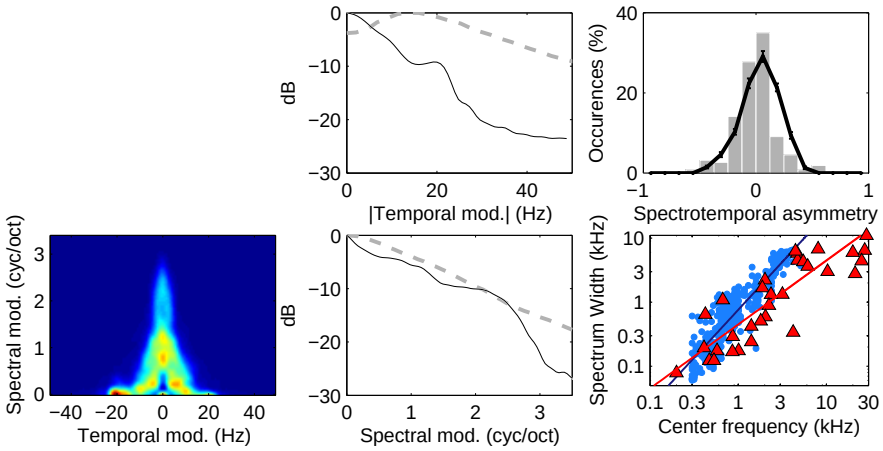

Figure 4: Fit to data in auditory cortex using the best vision parameters, here for the sparse rbm algorithm.

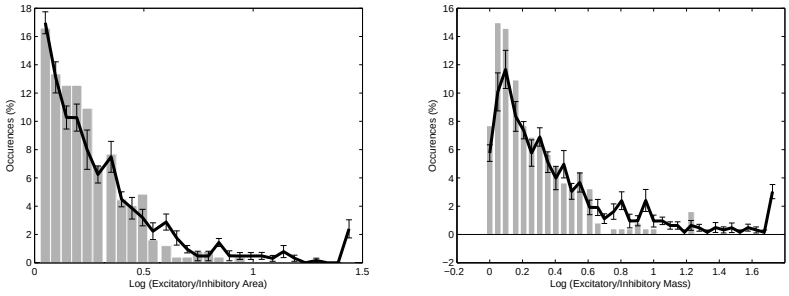

Figure 5: Fit to data in somatosensory cortex using the best vision parameters, here for the sparse autoencoder algorithm. Left: Ratio of excitatory to inhibitory receptive field area. Right: Ratio of excitatory to inhibitory receptive field mass. Black, model distribution. Gray, experimental data from [12].

2 Naturalistic experience and normal receptive field properties

2.1 Primary visual cortex

The figures below display the example bases learned by algorithms other than K-means on our natural image dataset.

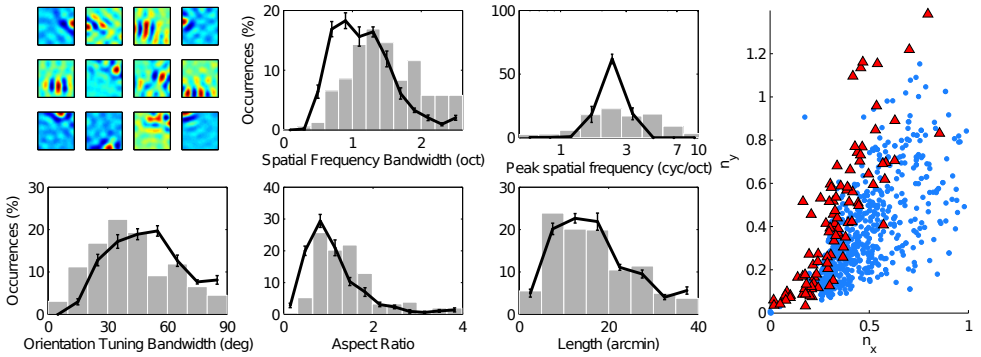

Figure 6: Top left: Sparse auto-encoder bases learned from natural images. Histograms: Black lines show population statistics for bases, gray bars show V1 simple cell data from Macaque. Far right: Distribution of receptive field shapes; Red triangles are V1 simple cells from [13], blue circles are bases.

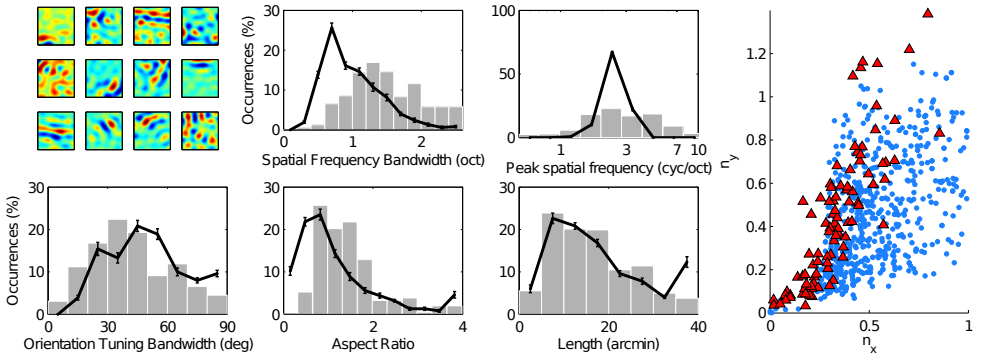

Figure 7: Top left: Sparse-RBM bases learned from natural images. Histograms: Black lines show population statistics for bases, gray bars show V1 simple cell data from Macaque. Far right: Distribution of receptive field shapes; Red triangles are V1 simple cells from [13], blue circles are bases.

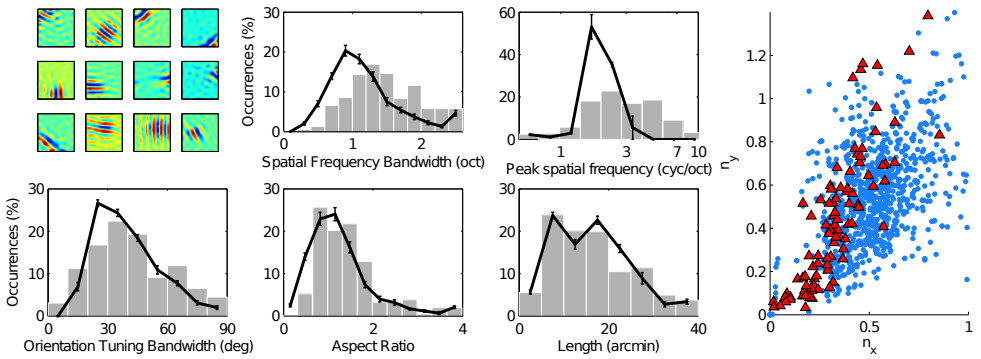

Figure 8: Top left: ICA bases learned from natural images. Histograms: Black lines show population statistics for bases, gray bars show V1 simple cell data from Macaque. Far right: Distribution of receptive field shapes; Red triangles are V1 simple cells from [13], blue circles are bases.

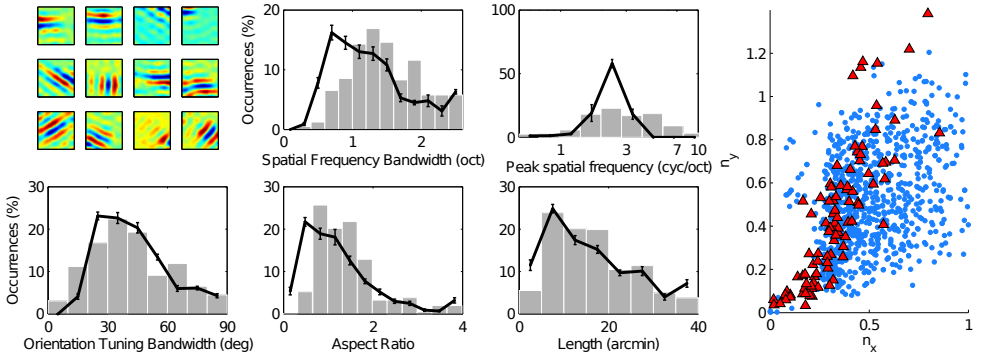

Figure 9: Top left: Sparse-coding bases learned from natural images. Histograms: Black lines show population statistics for bases, gray bars show V1 simple cell data from Macaque. Far right: Distribution of receptive field shapes; Red triangles are V1 simple cells from [13], blue circles are bases.

2.2 Primary auditory cortex

The figures below display the example bases learned by algorithms other than Sparse-RBM on our natural sound dataset.

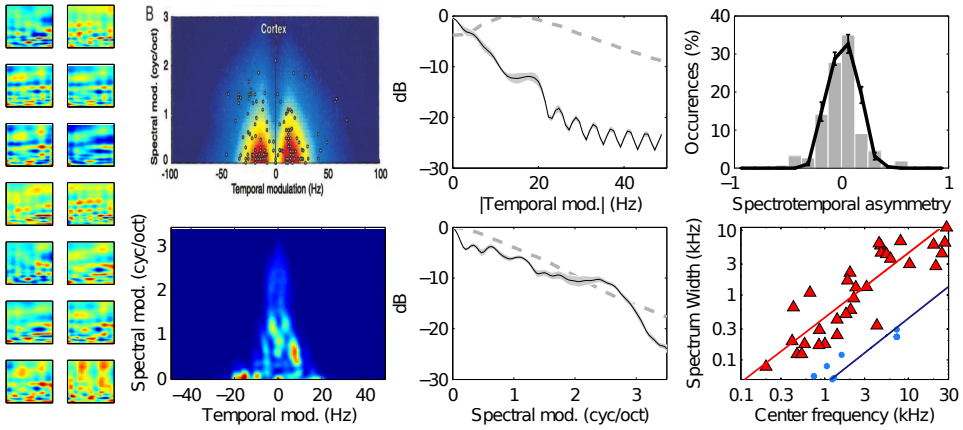

Figure 10: Clockwise from top left: Sparse autoencoder bases; Composite MTF in cat A1, reproduced from [14]; Composite MTF for sparse autoencoder; temporal MTF in A1 (dashed gray) and for our model (black); spectral MTF; frequency sweep preference; Spectrum width vs center frequency for A1 neurons (red triangles) and model neurons (blue circles).

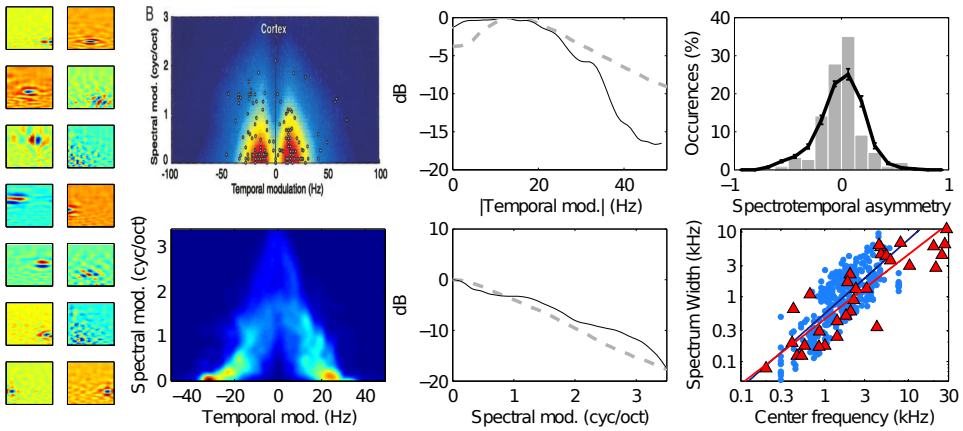

Figure 11: Comparison to A1. Clockwise from top left: ICA bases; Composite MTF in cat A1, reproduced from [14]; Composite MTF for algorithm; temporal MTF in A1 (dashed gray) and for our model (black); spectral MTF; frequency sweep preference; Spectrum width vs center frequency for A1 neurons (red triangles) and model neurons (blue circles).

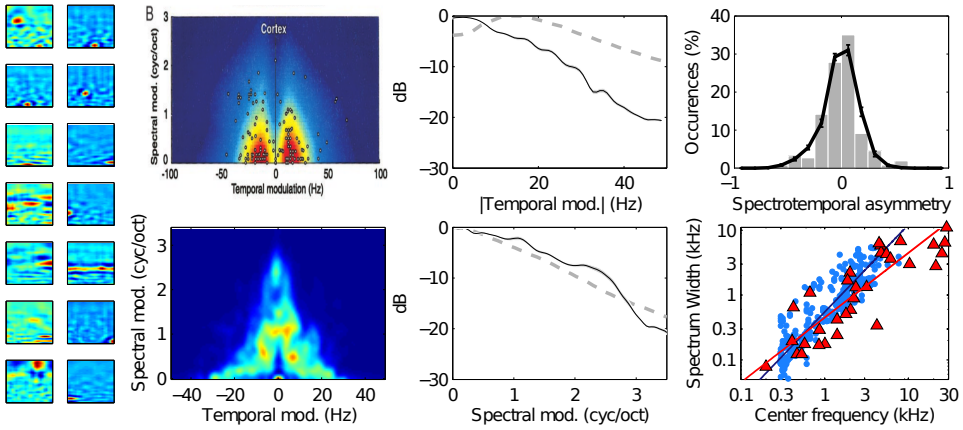

Figure 12: Comparison to A1. Clockwise from top left: K-means bases; Composite MTF in cat A1, reproduced from [14]; Composite MTF for algorithm; temporal MTF in A1 (dashed gray) and for our model (black); spectral MTF; frequency sweep preference; Spectrum width vs center frequency for A1 neurons (red triangles) and model neurons (blue circles).

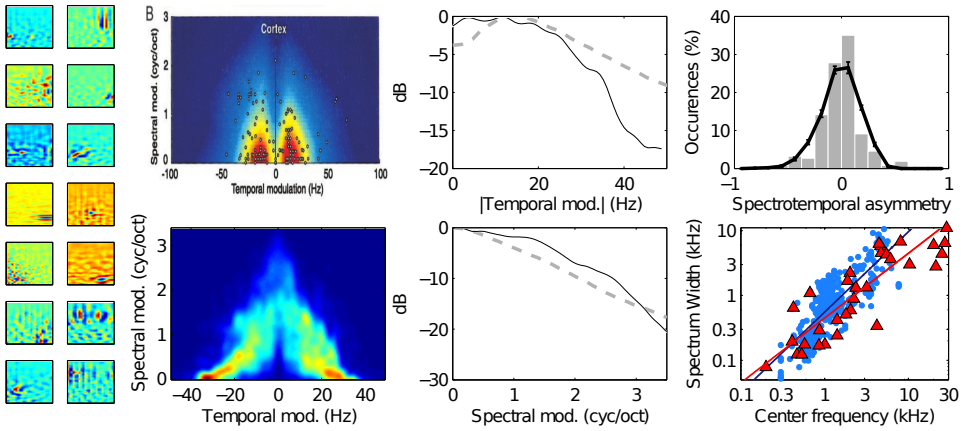

Figure 13: Comparison to A1. Clockwise from top left: Sparse-coding bases; Composite MTF in cat A1, reproduced from [14]; Composite MTF for algorithm; temporal MTF in A1 (dashed gray) and for our model (black); spectral MTF; frequency sweep preference; Spectrum width vs center frequency for A1 neurons (red triangles) and model neurons (blue circles).

2.3 Primary somatosensory cortex

The figures below display the example bases learned by algorithms other than Sparse-autoencoders on our natural touch dataset.

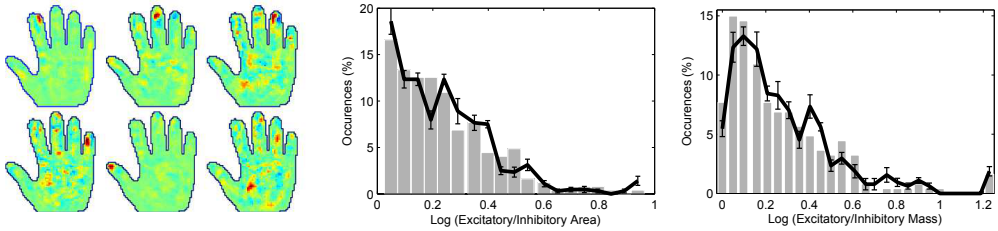

Figure 14: Left: Sparse-RBM bases. Middle and Right: Histograms of receptive field structure for the algorithm. Black, model distribution. Gray, experimental data from [12].

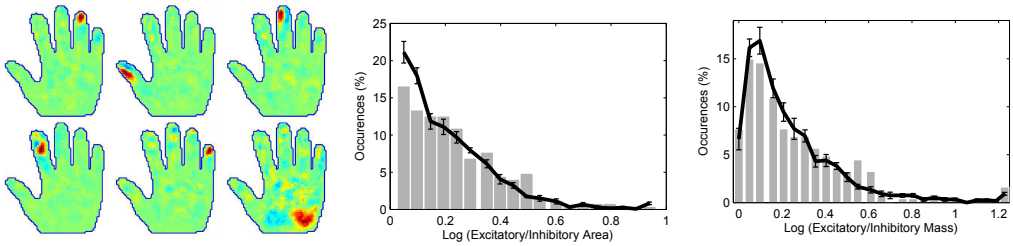

Figure 15: Left: ICA bases. Middle and Right: Histograms of receptive field structure for the algorithm. Black, model distribution. Gray, experimental data from [12].

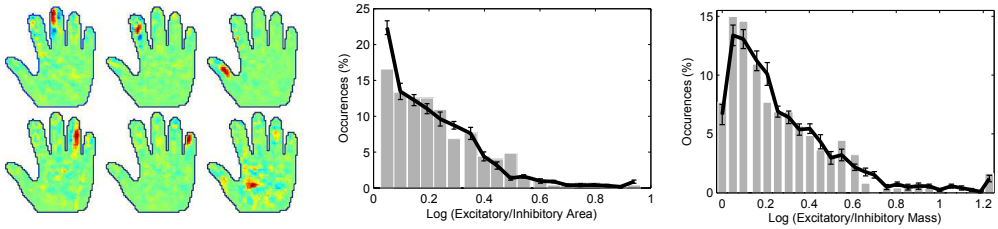

Figure 16: Left: K-means bases. Middle and Right: Histograms of receptive field structure for the algorithm. Black, model distribution. Gray, experimental data from [12].

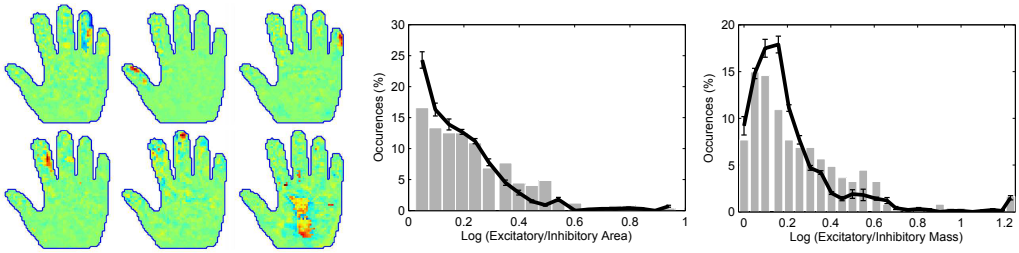

Figure 17: Left: Sparse coding bases. Middle and Right: Histograms of receptive field structure for the algorithm. Black, model distribution. Gray, experimental data from [12].

3 Adaptation to altered environmental statistics

3.1 Goggle-rearing alters V1 orientation tuning

The figures below display the orientation histograms for all the other learning algorithms.

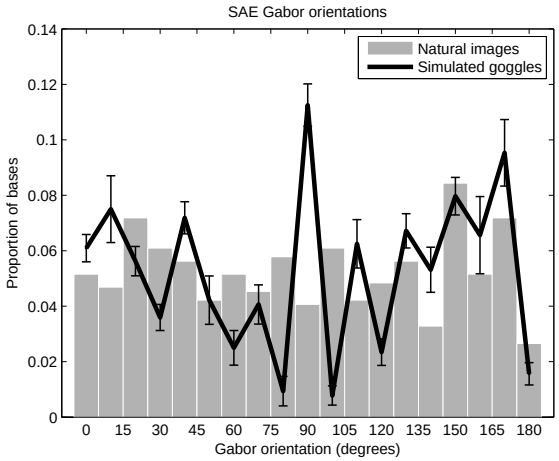

Figure 18: Orientation histogram for sparse-autoencoder (biased towards goggle orientation) ($90°$).

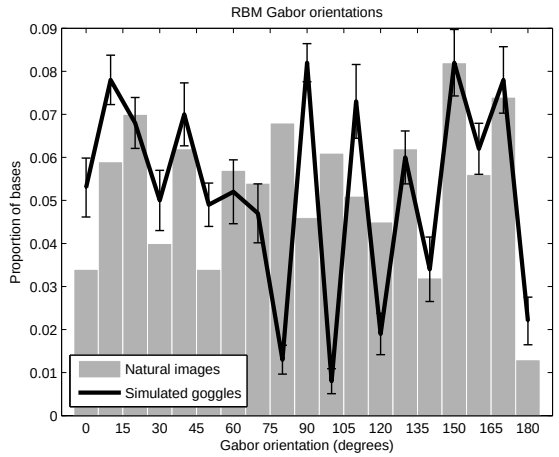

Figure 19: Orientation histogram for sparse-rbm (biased towards goggle orientation) ($90°$).

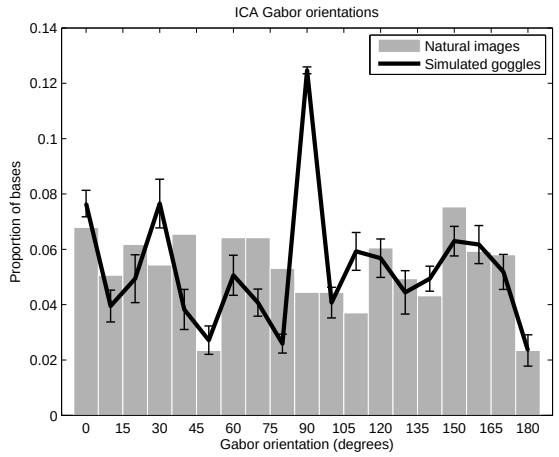

Figure 20: Orientation histogram for ICA (biased towards goggle orientation) ($90°$).

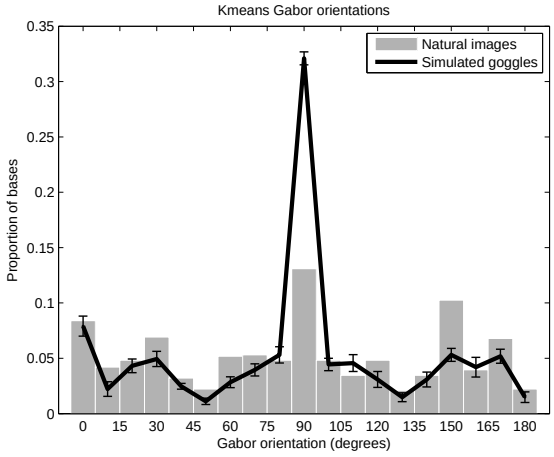

Figure 21: Orientation histogram for K-means (biased towards goggle orientation) ($90°$).

3.2 Pulsed-tone rearing alters A1 frequency tuning

The figures below display the population histograms for all the other learning algorithms.

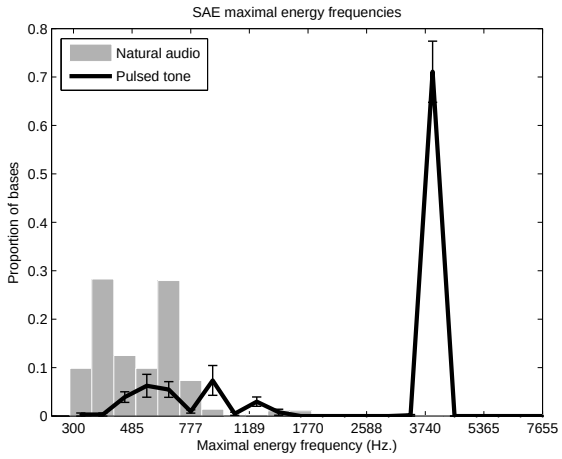

Figure 22: Sparse-autoencoder bases: Population histograms of preferred frequency (reveals a strong preference for the pulsed-tone frequency of 4kHz).

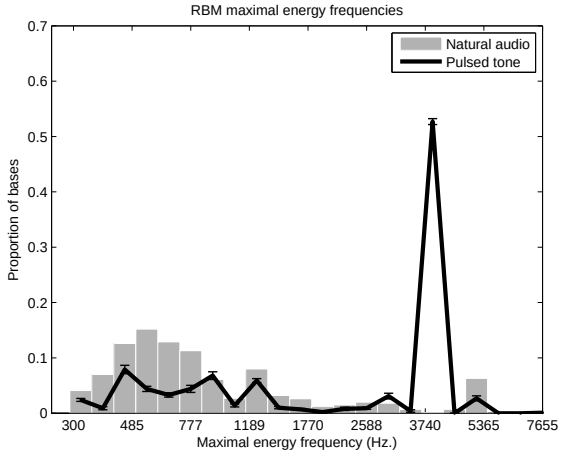

Figure 23: Sparse-RBM bases: Population histograms of preferred frequency (reveals a strong preference for the pulsed-tone frequency of 4kHz).

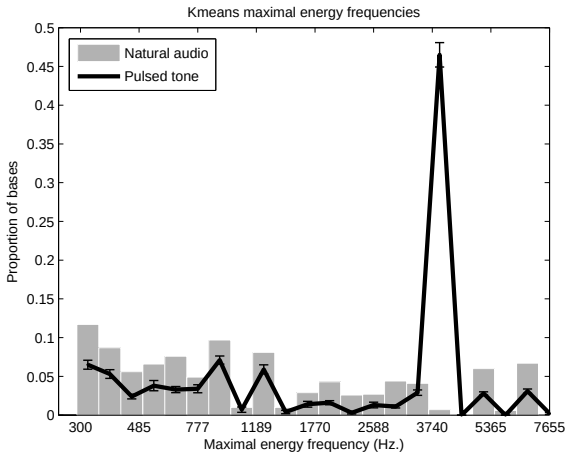

Figure 24: K-means bases: Population histograms of preferred frequency (reveals a strong preference for the pulsed-tone frequency of 4kHz).

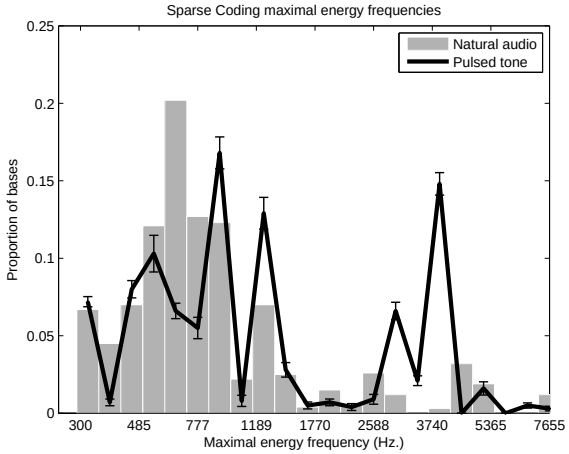

Figure 25: Sparse-coding bases: Population histograms of preferred frequency (reveals a strong preference for the pulsed-tone frequency of 4kHz).

3.3 Artificial digital syndactyly

The figure below displays sample syndactyly bases for all other learning algorithms (plus PCA).

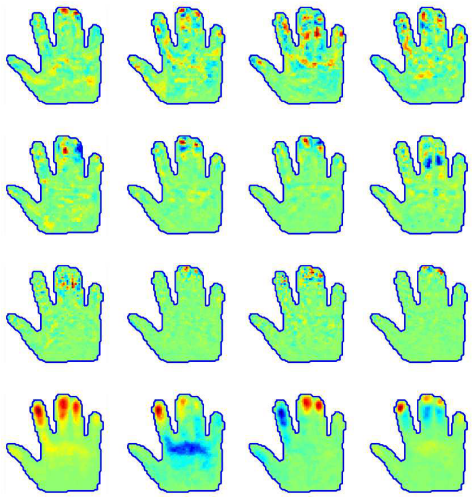

Figure 26: Bases trained on artificial syndactyly data learn double digit receptive fields. Top to bottom:row: Sparse-RBM, ICA, Sparse-coding and PCA (first 4 components)

References

[1] A.J. Bell and T.J. Sejnowski. The "independent components" of natural scenes are edge filters. *Vision Res.*, 37(23):3327–38, December 1997.

[2] J.H. van Hateren and D.L. Ruderman. Independent component analysis of natural image sequences yields spatio-temporal filters similar to simple cells in primary visual cortex. *Proc. R. Soc. Lond. B*, 265(1412):2315–20, December 1998.

[3] A. Hyvärinen, J. Hurri, and P.O. Hoyer. *Natural Image Statistics*. Springer, London, 2009.

[4] A. Hyvärinen and E. Oja. A fast fixed-point algorithm for independent component analysis. *Neural Comp.*, 9:1483–1492, 1997.

[5] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *ICML*, 2008.

[6] H. Lee, C. Ekanadham, and A.Y. Ng. Sparse deep belief net model for visual area V2. In *NIPS*, 2008.

[7] G.E. Hinton, S. Osindero, and Y.-W. Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.*, 18:1527–1554, 2006.

[8] B.A. Olshausen and D.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–9, 1996.

[9] R.D. Patterson, K. Robinson, J. Holdsworth, D. McKeown, C. Zhang, and M. Allerhand. Complex sounds and auditory images. In *Adv. Biosci.*, 1992.

[10] M. Slaney. An Efficient Implementation of the Patterson-Holdsworth Auditory Filter Bank, 1993.

[11] D.P.W. Ellis. Gammatone-like spectrograms, 2009.

[12] J.J. DiCarlo, K.O. Johnson, and S.S. Hsiao. Structure of receptive fields in area 3b of primary somatosensory cortex in the alert monkey. *J. Neurosci.*, 18(7):2626–45, April 1998.

[13] D.L. Ringach. Spatial structure and symmetry of simple-cell receptive fields in macaque primary visual cortex. *J. Neurophysiol.*, 88(1):455–63, July 2002.

[14] L.M. Miller, M.A. Escabí, H.L. Read, and C.E. Schreiner. Spectrotemporal receptive fields in the lemniscal auditory thalamus and cortex. *J. Neurophysiol.*, 87(1):516–27, January 2002.