Hierarchical K-Means Algorithm for Modeling Visual Area V2 Neurons

Xiaolin Hu, Peng Qi, and Bo Zhang

State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology (TNList), and Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
xiaolin.hu@gmail.com, pengrobertqi@163.com, dcszb@tsinghua.edu.cn

Abstract. Computational studies about the properties of the receptive fields of neurons in the cortical visual pathway of mammals are abundant in the literature but most addressed neurons in the primary visual area (V1). Recently, the sparse deep belief network (DBN) was proposed to model the response properties of neurons in the V2 area. By investigating the factors that contribute to the success of the model, we find that a simple algorithm for data clustering, K-means algorithm can be stacked into a hierarchy to reproduce these properties of V2 neurons, too. In addition, it is computationally much more efficient than the sparse DBN.

Keywords: Neural network, Deep learning, Visual area, V1, V2.

1 Introduction

Since Hubel and Wiesel [1] found that the receptive fields of many neurons in the primary visual cortex (V1) are edge detectors, a wealth of researches have attempted to interpret this ground breaking discovery. Two well-known proposals refer to sparse coding [2, 3] and independent component analysis (ICA) [4]. Both approaches can be understood as a single layer network where the inputs are image pixels and the outputs correspond to the responses of V1 simple cells, which are assumed to be sparse, i.e., the output units should keep silence or near silence most of the time and fire only occasionally. Sparsity is closely related to high-order statistics of natural images, which plays a significant role in reproducing the edge-like structure of the receptive fields of V1 simple cells. In fact, with sparsity constraint many other models such as the restricted Boltzmann machine (RBM) [5], auto-encoder [6] and K-means algorithm [7, 8] have been found to be able to learn the edge-like structure of the receptive fields of V1 neurons on natural images.

Hierarchical models [9,10] have been proposed for modeling the response properties of V1 complex cells, another important type of neurons in V1 area. However, there have been few attempts to quantitatively model the properties of neurons beyond V1 along the cortical visual pathway such as V2 or V4. The famous hierarchical model HMAX [11] was tested against V4 neurons and achieved remarkable results [12]. But the properties of its low level units are handcrafted

T. Huang et al. (Eds.): ICONIP 2012, Part III, LNCS 7665, pp. 373–381, 2012.

and what is more interesting to the computational neuroscience community is learning each layer in a similar way. The deep belief network [13] is such a model. It consists of multiple layers of RBMs, and learning starts from the bottom layer to the top layer in the sequel. It was found that a two-layer DBN is able to replicate some properties of the receptive fields of both V1 and V2 neurons by imposing a sparse firing constraint on each layer [5]. This model owes its success largely to its nonlinearity on the the first layer output. In the present paper, we will show that the difference between the sparsity degrees on the two layers are also critical for producing these results. To be more specifically, the second layer firing should not be as sparse as the first layer. If one seeks alternative models for doing similar task, neither of the two factors should be ignored.

In the paper, we will show that the K-means algorithm, a simple data clustering algorithm, can be stacked into a hierarchy to model V2 neurons. However, as the standard K-means algorithm is an extremely sparse model (for each input data only one hidden unit fires), to control its sparsity degree, some modifications are needed.

2 Sparse Deep Belief Network

A restricted Boltzmann machine (RBM) consists of a layer of visible units $\mathbf{v}$, a layer of hidden units $\mathbf{h}$ and a symmetric connections weights between the two layers represented by a matrix W. The visible units and hidden units have biases, denoted by c_i and b_j, respectively [14]. The sparse RBM imposes a sparse firing constraint on the hidden units [5]. With a set of training data $\mathbf{v}_1, \ldots, \mathbf{v}_N$ where $\mathbf{v}_n \in R^D$, the sparse RBM minimizes the following function

$$-N\langle \log \sum_{\mathbf{h}} P(\mathbf{v}, \mathbf{h})\rangle + \lambda \sum_{j=1}^{K} \|p - \langle E(h_j|\mathbf{v})\rangle\|^2$$

over w_{ij}, c_i and b_j, where

$$-\log P(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2}\sum_i v_i^2 - \frac{1}{\sigma^2}\left(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i w_{ij} h_j\right)$$

and $\lambda, \sigma > 0$. In above equations, $\langle\cdot\rangle$ denotes average over samples and $E(\cdot)$ denotes the conditional expectation given the data. The parameter p is the desired firing probability of the hidden units, which controls the sparsity degree of firing.

With a modified contrastive divergence learning rule [5], the sparse RBM can learn the gabor-like receptive fields of V1 simple cells on natural images. Fig. 1 visualizes 200 weights associated with the hidden units. They were learned on a large set of randomly selected 14-by-14 patches from ten 512-by-512 natural images [2], which were preprocessed by $1/f$ whitening and low pass filtering in the frequency domain. The sparsity parameter is set as $p = 0.02$.

We stacked another sparse RBM with 200 hidden units on top of the first layer, and trained the second layer weights and biases by freezing the first layer

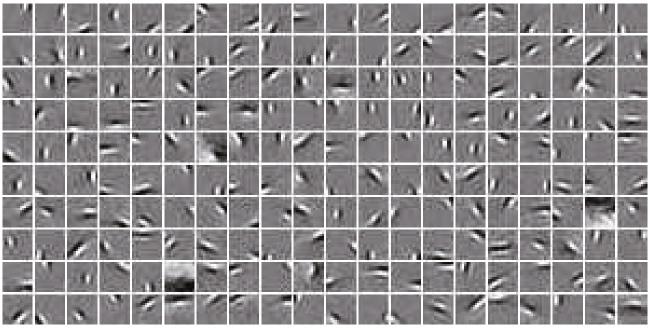
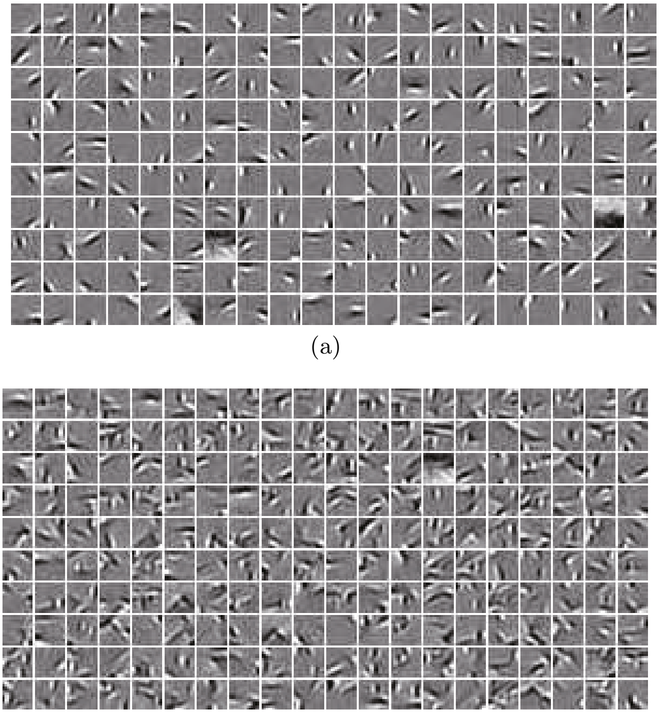


Fig. 1. Visualization of 200 first layer weight vectors of the sparse DBN. Each 14 × 14 patch corresponds to a weight vector.



(a)

(b)

Fig. 2. Visualization of 200 second layer weight vectors of the sparse DBN. (a) $p = 0.02$, (b) $p = 0.04$.

weights and biases. The resulting model is called *sparse deep belief network* or sparse DBN [5]. The receptive fields of the second layer units are visualized in Fig. 2 as weighted sum of the receptive fields of first layer units. It is seen that with $p = 0.02$ the receptive fields are visually similar to the receptive fields of the first layer; while with $p = 0.04$ the receptive fields are like edge conjunctions or corners, in agreement with the V2 neuron properties. In fact, with increasing p (greater than 0.02), our experiments showed that the structure of the receptive fields became more and more complex (data not shown). This observation suggests that the nonlinearity of the sparse RBM is not the only factor that contributes to the emergence of V2 neuron receptive fields, and the higher firing rate on the second layer than on the first layer is another critical factor. If one seeks alternative models for reproducing the V2 neuron properties, both factors should be considered.

3 Hierarchical K-Means Algorithms

3.1 K-Means Algorithms

The goal of K-means algorithm is to partition the data set $\mathbf{v}_1, \ldots, \mathbf{v}_N$ into K clusters. If we introduce a latent variable $\mathbf{w}_j$, the mean or centroid of cluster j, where $j = 1, \ldots, K$, then the goal is to identify $\mathbf{w}_j$. The algorithm consists of two iterative steps:

- For each input $\mathbf{v}_n$ determine which cluster it belongs to. Mathematically, this amounts to determine $j^* = \arg\min_j \|\mathbf{v}_n - \mathbf{w}_j\|$.
- Update $\mathbf{w}_j$ for $j = 1, \ldots, K$ by taking the mean (centroid) of data assigned to cluster j.

Each data point $\mathbf{v}_n$ is assigned a binary indicator vector $\mathbf{h}$ where $h_j = 1$ if this point belongs to cluster j and $h_j = 0$ otherwise. If the latent variables h_j are viewed as "neurons", then the firing pattern of these neurons is extremely sparse—for each input only one neuron fires.

3.2 Multiple Firing K-Means Algorithms

Now we relax the algorithm by allowing multiple hidden units fire together for an input in the first step. Specifically, for each input $\mathbf{v}_n$ we determine L clusters it belongs to. Mathematically, this amounts to determine a set $\Omega \subset V = \{1, \ldots, K\}$ such that $|\Omega| = L$ and $\|\mathbf{v}_n - \mathbf{w}_s\| \leq \|\mathbf{v}_n - \mathbf{w}_j\|$ for $s \in \Omega$ and $j \in V \backslash \Omega$.

For each input $\mathbf{v}_n$ set

$$h_j(\mathbf{v}_n) = \begin{cases} 1, & \text{if } \mathbf{v}_n \text{ belongs to cluster } j; \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Then there are always L hidden units firing. For this reason this algorithm is called *multiple firing K-means algorithm*. Its convergence results are stated in the following theorem.

Theorem 1. *Each step of the multiple firing K-means algorithm lowers the value of the function*

$$J = \langle \sum_{j=1}^{K} h_j \|\mathbf{v} - \mathbf{w}_j\|^2 \rangle \tag{2}$$

until convergence.

Proof. In step 1, $\mathbf{w}_j$ is fixed. It is easy to see that setting $h_j = 1$ for $j \in \Omega$ and $h_j = 0$ for $j \in V \backslash \Omega$ corresponds to the minimum of J over the binary vector $\mathbf{h}$ subject to the constraint that for each input there are always L elements equal to 1. In step 2, $\mathbf{h}$ is fixed. Notice that $\frac{\partial J}{\partial \mathbf{w}_j} = -2\langle h_j(\mathbf{v} - \mathbf{w}_j)\rangle$. Then step 2 is equivalent to taking $\frac{\partial J}{\partial \mathbf{w}_j} = 0$, which corresponds to minimization of J over $\mathbf{w}_j$. Therefore, each step results in a decrease of J until convergence.

3.3 Hierarchical Model

Similar to the sparse DBN, we can stack another multiple firing K-means algorithm on top of the first layer. It takes the output of the first layer as input and learns the centroids of the inputs by freezing the first layer centroids.

4 Experiments

It has been shown that the standard K-means algorithm can reproduce the gabor-like receptive fields of V1 cells [7,8]. Here we show that the multiple firing K-means algorithm has the same capability. A large number of 14-by-14 patches were randomly extracted from ten natural images, which were preprocessed in the same way as in Section 2. At every iteration 50,000 patches were input to the network and the centroids got updated once. After about 100 iterations the algorithm converged. The 200 centroids are plotted in Fig. 3 with $L = 3$. For other small values of L the results were visually similar to this figure (we tested $L = 5, 7, 10$).

We stacked a second layer multiple firing K-means algorithm to the output of the first layer. The second layer had 200 units and L was set to 10. After 100 iterations, the algorithm converged. It was found that only a few elements in the learned second layer centroids were significantly larger than zero (data not shown). The second layer centroids are visualized in Fig. 4 in the same manner as Fig. 2. It is seen that the shape of many second layer centroids are like corners or conjunctions of edges, in agreement with some V2 neurons properties [5].

To test the properties of the second layer units, we generated a set of angle stimuli as shown in Fig. 5 [15]. Each stimulus was a 14-by-14 image patch representing an angle in $\{\frac{2\pi}{M}, \frac{4\pi}{M}, \ldots, \frac{2(M-1)\pi}{M}\}$ in different orientations, which resulted in $M(M-1)$ different stimuli. See [15] for details. In addition, each stimulus was normalized to zero mean and unit variance. To identify the "center" of each second layer unit's receptive field, we translated all stimuli densely over the 14 × 14 input image patch, and identified the position at which the

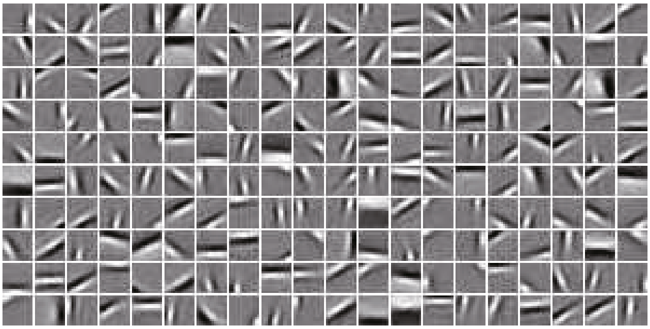


Fig. 3. Visualization of 200 first layer centroids of the hierarchical K-means

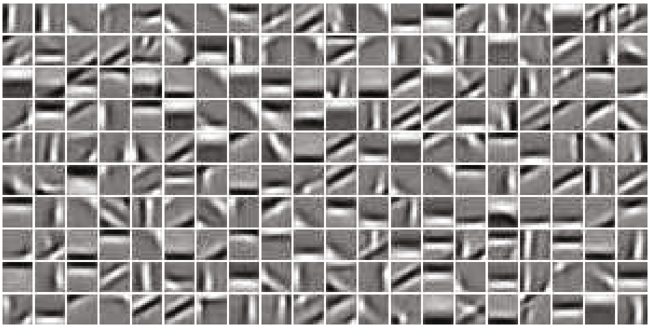


Fig. 4. Visualization of 200 second layer centroids of the hierarchical K-means.

maximum response was elicited. All measures were then taken with all angle stimuli centered at this position.

For each angle stimulus, we calculated the responses of the first layer units and second layer units sequentially. Fig. 5 shows the stimuli set with $M = 24$ together with responses of three representative second layer units. Note the similarity to Fig. 5 in [5]. And we emphasize that these units are typical in our model.

To make quantitative comparison between the simulation results and physiological results in [15], we then generated a stimuli set with $M = 12$. Five quantities about the statistics of the response profiles of the model neurons on the stimuli set were calculated and presented in Fig. 6. The physiological results and the model neurons by the sparse DBN are also presented in the figure. It is seen that the hierarchical K-means algorithm has produced similar results.

As the hierarchical K-means algorithm and the sparse DBN can produce similar results, then how about their computational efficiency? This is not a question in the computational neuroscience community but is important in engineering applications. One difficulty for such a comparison is that a common termination condition is lacked for the algorithms (notice that their final results are not the same, though qualitatively similar). Fortunately, our experiments showed that the computing time of the two algorithms differed much for producing visually

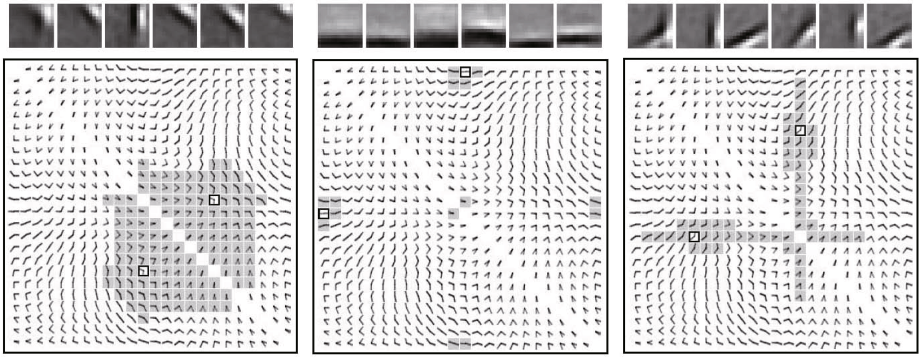


Fig. 5. Response profile of three example model V2 neurons on a set of angle stimuli. Top: the left most patch shows a model V2 neuron by taking the weighted sum of V1 simple cell receptive fields. The next five patches show the receptive fields of the model V1 simple cells that have strongest connections to this V2 neuron. Bottom: darkened patches represent stimuli to which the model V2 neuron responded strongly. A small black square indicates the overall peak response.

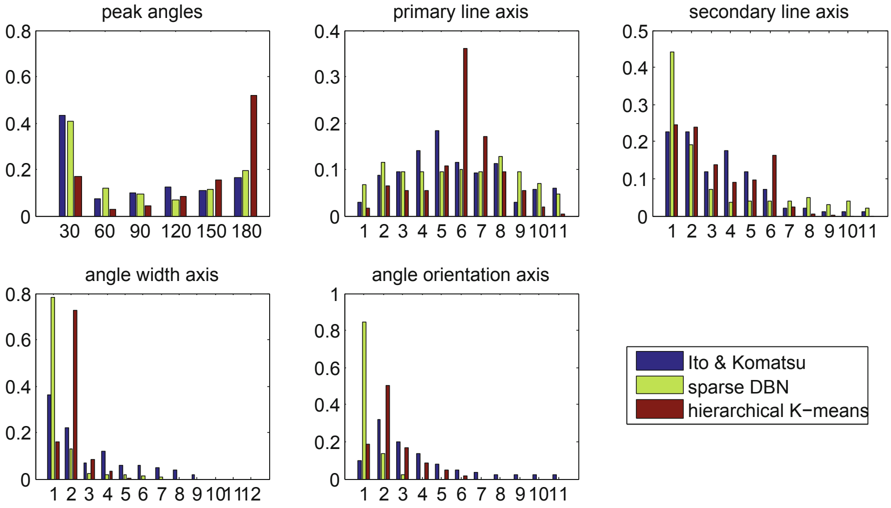


Fig. 6. Distribution of the response statistics over the angle stimuli. The five figures show respectively the distribution over (i) peak angle response, (ii) tolerance to primary line component, (iii) tolerance to secondary line component, (iv) tolerance to angle width, (v) tolerance to angle orientation. See [5, 15] for details. Best viewed in color.

similar results. Table 1 shows the computing time of the two algorithms on a computer (Intel Core i5-2320 3GHz × 4, RAM 8GB), averaged over 10 trials, for producing visually similar results to Figs. 1, 2(b), 3 and 4, respectively. For sparse DBN, p=0.02 in layer 1 and p=0.04 in layer 2. Moreover, in each layer σ decayed by a factor of 0.99 after every iteration with initial value 0.4, as suggested in [16]. Other parameters were tuned to achieve high efficiency. Learning terminated after 800 iterations for each layer and in every iteration 100,000 patches were input to the model in batches of 200. For hierarchical K-means, learning terminated after 100 iterations for each layer and in every iteration 50,000 patches were input to the model together. It is seen that learning in each layer of the hierarchical K-means algorithm is more than ten times faster than the sparse DBN.

Table 1. Comparison of the computing time in seconds

	V1	V2
sparse DBN	2536.7±21.0	2693.1±37.3
hierarchical K-means	164.3± 4.9	206.3±2.7

5 Conclusions

There are many models capable of reproducing edge-like structure of the receptive fields of V1 neurons, but few have shown to be capable of reproducing edge conjunction structure of the receptive fields of V2 neurons, except the sparse DBN. In the paper a hierarchial K-means algorithm is proposed as an alternative model for the visual area V2. The simulation results on natural images qualitatively matched physiological data recorded in monkeys. It was shown to be much more computationally efficient than the sparse DBN. A promising future direction of this research is to extend the hierarchical K-means algorithm to deep models for learning object parts for computer vision, like the convolutional DBN [17].

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grants 91120011 and 61134012, National Basic Research Program (973 Program) of China under Grant 2012CB316301, Cross-discipline Foundation and Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList).

References

1. Hubel, D.H., Wiesel, T.N.: Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat. Journal of Neurophysiology 28, 229–289 (1965)

2. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
3. Olshausen, B.A., Field, D.J.: Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision Research 37(23), 3311–3325 (1997)
4. Bell, A.J., Sejnowski, T.J.: The "independent components" of natural scenes are edge filters. Vision Research 37(23), 3327–3338 (1997)
5. Lee, H., Ekanadham, C., Ng, A.: Sparse deep belief net model for visual area V2. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, vol. 20 (2007)
6. Ranzato, M., Boureau, Y.L., LeCun, Y.: Sparse feature learning for deep belief networks. In: Platt, J., Koller, D., Singer, Y., Roweis, S. (eds.) Advances in Neural Information Processing Systems (NIPS), Vancouver, Canada, vol. 20 (2007)
7. Coates, A., Lee, H., Ng, A.Y.: An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft. Lauderdale, FL (2011)
8. Saxe, A.M., Bhand, M., Mudur, R., Suresh, B., Ng, A.Y.: Unsupervised learning models of primary cortical receptive fields and receptive field plasticity. In: Shawe-Taylor, J., Zemel, R., Bartlett, P., Pereira, F., Weinberger, K. (eds.) Advances in Neural Information Processing Systems (NIPS), vol. 24, pp. 1971–1979 (2011)
9. Karklin, Y., Lewicki, M.S.: A hierarchical bayesian model for learning nonlinear statistical regularities in nonstationary natural signals. Neural Computation 17, 397–423 (2005)
10. Karklin, Y., Lweicki, M.S.: Emergence of complex cell properties by learning to generalize in natural scenes. Nature 457, 83–86 (2009)
11. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nature Neuroscience 2, 1019–1025 (1999)
12. Cadieu, C., Kouh, M., Pasupathy, A., Connor, C.E., Riesenhuber, M., Poggio, T.: A model of V4 shape selectivity and invariance. Journal of Neurophysiology 98, 1733–1750 (2007)
13. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18, 1527–1554 (2006)
14. Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural Computation 14, 1771–1800 (2002)
15. Ito, M., Komatsu, H.: Representation of angles embedded within contour stimuli in area V2 of macaque monkeys. The Journal of Neuroscience 24(13), 3313–3324 (2004)
16. Ekanadham, C.: Sparse deep belief net models for visual area V2. Undergraduate honors thesis, Stanford University (2007)
17. Lee, H., Grosse, R., Ranganath, R., Ng, A.Y.: Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In: Proceedings of the 26th International Conference on Machine Learning (ICML), Montreal, Canada, pp. 609–616 (2009)