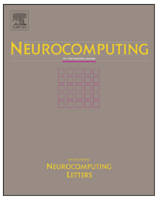
Neurocomputing 134 (2014) 198–205

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Modeling response properties of V2 neurons using a hierarchical K-means model[☆]

Xiaolin Hu [a,*], Jianwei Zhang [b], Peng Qi [a], Bo Zhang [a]

[a] *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*

[b] *Department of Informatics, University of Hamburg, Hamburg D-22527, Germany*

ARTICLE INFO

Article history:
Received 28 February 2013
Received in revised form
4 June 2013
Accepted 17 July 2013
Available online 24 January 2014

Keywords:
Neural network
Deep learning
K-means
V1
V2

ABSTRACT

Many computational models have been proposed for interpreting the properties of neurons in the primary visual cortex (V1). But relatively fewer models have been proposed for interpreting the properties of neurons beyond V1. Recently, it was found that the sparse deep belief network (DBN) could reproduce some properties of the secondary visual cortex (V2) neurons when trained on natural images. In this paper, by investigating the key factors that contribute to the success of the sparse DBN, we propose a hierarchical model based on a simple algorithm, K-means, which can be realized by competitive Hebbian learning. The resulting model exhibits some response properties of V2 neurons, and it is more biologically feasible and computationally efficient than the sparse DBN.

1. Introduction

The response properties of the neurons in the primary visual area (V1) have long been studied since the ground breaking discovery that the receptive fields of these neurons are edge-like [1]. Many computational models have been proposed to model such properties, among which two well-known proposals are sparse coding [2,3] and independent component analysis (ICA) [4]. Both approaches can be viewed as single-layer linear networks whose inputs are image pixels and outputs are assumed to be sparse. The outputs correspond to the responses of simple cells in V1. Sparse response means that the output units are silent most of the time and fire only occasionally. It has been demonstrated that the sparsity constraint plays a significant role in reproducing the edge-like receptive fields of V1 simple cells. Some recent nonlinear models enjoying the benefit of sparsity constraints are also able to replicate the edge-like receptive fields of V1 simple cells, including restricted Boltzmann machine (RBM) [5], auto-encoder [6], and K-means algorithm [7,8].

Another important type of neurons in V1, the complex cells, also attracted much interest. To reproduce the spatial phase invariance property of these cells, hierarchical models [9–11] have been proposed to model nonlinear statistical regularities in natural images. However, there have been fewer attempts to quantitatively model the properties of neurons beyond V1 along the cortical ventral pathway, such as V2 or V4. The famous hierarchical model HMAX [12] was shown to be able to reproduce some properties of V4 neurons [13], but the properties of the low level units of this model were handcrafted. What is more interesting to the computational neuroscience community is a model that emulates the visual pathway in a layer-wise fashion, and preferably employs the same learning method in different layers. Such a model would provide potentially better explanation for the learning procedure that takes place in the brain. The deep belief network (DBN) [14] is a candidate. A DBN consists of multiple layers of restricted Boltzmann machines (RBMs), and its learning starts from the bottom layer, and progresses layer-by-layer to the top layer in a similar fashion. It was found that with sparsity constraints on each layer, a two-layer DBN was able to replicate some properties of the receptive fields of both V1 neurons and V2 neurons [15,5]. However, RBM is a highly abstract model and its learning algorithm, namely, the contrastive divergence algorithm [16], is too complicated for biological systems. In addition, training RBM is computationally expensive, which has been a barrier to its wider use. Though some remedies have been proposed [17], it is still desirable to come up with some simple and efficient alternatives. This servers as our motivation for this work.

[☆] This work was supported by the National Basic Research Program (973 Program) of China (Grant Nos. 2013CB329403 and 2012CB316301), National Natural Science Foundation of China (Grant Nos. 61273023 and 91120011), Tsinghua University Initiative Scientific Research Program No. 20121088071, Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation and the DFG-MOE International Research Training Group IGK 1247 CINACS.

[*] Corresponding author.
E-mail address: xlhu@tsinghua.edu.cn (X. Hu).

0925-2312/$ - see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.07.052

Compared with previous models that were only capable of modeling V1 simple cell response properties by imposing sparsity constraints, the DBN owes its success largely to the nonlinearity on its first-layer output. In the present paper, we will demonstrate that the level of sparsity on the second-layer output also plays a key role in obtaining these results. More specifically, the second-layer model response should not be too sparse. Both factors should be taken into consideration if one seeks an alternative model for similar tasks.

Inspired by these observations, we propose a very simple yet effective model for V2 neurons. The model originates from the K-means clustering algorithm, which can be considered as an extremely sparse single-layer model where the input is image pixels and only one of the output units takes a non-zero value. To control the sparsity level of this algorithm, some modifications are necessary.

The rest of this paper is organized as follows. In Section 2 the sparse DBN is briefly reviewed and the critical factors that contribute to its success are discussed. In Section 3, a modified K-means algorithm is presented, which follows a hierarchical K-means model. Section 4 presents experimental results and Section 5 concludes the paper.

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 set of symmetric connection weights $\mathbf{W}$ between the two layers. The visible units and hidden units have biases, denoted by c_i and b_j, respectively [16]. The visible and hidden units are stochastic units which can only take 0 or 1. Given the parameters $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{c}$, an RBM defines the following joint distribution for its visible and hidden units:

$$p(\mathbf{v},\mathbf{h}) = \frac{1}{Z}\exp(-E(\mathbf{v},\mathbf{h})), \tag{1}$$

where $E(\mathbf{v},\mathbf{h}) = -\mathbf{v}^{\top}\mathbf{W}\mathbf{h} - \mathbf{c}^{\top}\mathbf{v} - \mathbf{b}^{\top}\mathbf{h}$ is called the energy function, and $Z = \int_{\mathbf{v},\mathbf{h}} \exp(-E(\mathbf{v},\mathbf{h}))$ is called the partition function. A standard RBM has no constraints on its visible and hidden units. See Fig. 1(a) for an illustration.

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:

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

over w_{ij}, c_i and b_j, where

$$-\log\ p(\mathbf{v},\mathbf{h}) = \frac{1}{2\sigma^2}\sum_i v_i^2 - \frac{1}{\sigma^2}\left(\sum_i c_i v_i + \sum_j b_j h_j + \sum_{i,j} v_i w_{ij} h_j\right) \tag{3}$$

and $\lambda, \sigma > 0$. In the above equations, $\langle \cdot \rangle$ denotes averaging over samples and $\mathbb{E}(\cdot)$ denotes the conditional expectation given the data. The parameter p is the desired firing probability of the hidden units, which controls the sparsity level of the hidden units. Also note that in these equations, the energy function is adjusted to use "Gaussian visible units", which takes real values instead of binary values to better accommodate the pixel intensities of natural images [18].

One advantage of RBM is that given the visible units, the hidden units are conditionally independent, and vice versa, which directly gives rise to the use of an efficient block Gibbs sampling method for learning. Specifically, the following probability distributions were used to sample the states of the stochastic units [5]:

$$p(v_i|\mathbf{h}) \sim \mathcal{N}\left(c_i + \sum_j w_{ij}h_j, \sigma^2\right),$$
$$P(h_j = 1|\mathbf{v}) = logistic\left(\frac{1}{\sigma^2}\left(b_j + \sum_i w_{ij}v_i\right)\right), \tag{4}$$

where $\mathcal{N}(\cdot)$ and $logistic(\cdot)$ denote the Gaussian distribution and logistic function, respectively.

With a modified contrastive divergence learning rule [5], the sparse RBM was able to learn Gabor-like weights on natural images resembling receptive fields of V1 simple cells. Fig. 2 visualizes the weights associated with 200 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 level was set as $p = 0.02$. Other parameter settings can be found in Section 4.4.

In order to reproduce V2 neuron-like response properties, 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 parameters (See Fig. 1(b)). The resulting model is called a *sparse deep belief network* or sparse DBN [5]. The receptive fields of the second-layer units are visualized in Fig. 3 as the weighted sum of the receptive fields of the first-layer units. It can be seen that with appropriate p, the receptive fields are like edge conjunctions or corners, in agreement with the V2 neuron properties (Fig. 3(b) and (c)). The nonlinearity in the first layer output, i.e. the binarization governed by the logistic function in (4), plays a significant role. This is because a two-layer linear model is equivalent to a single-layer linear model, and a linear model could at most reproduce V1 simple cell properties.

Fig. 3 shows that when the sparsity level p increases the receptive fields of the second-layer units become more and more complex. In fact, with $p = 0.02$ the receptive fields are visually

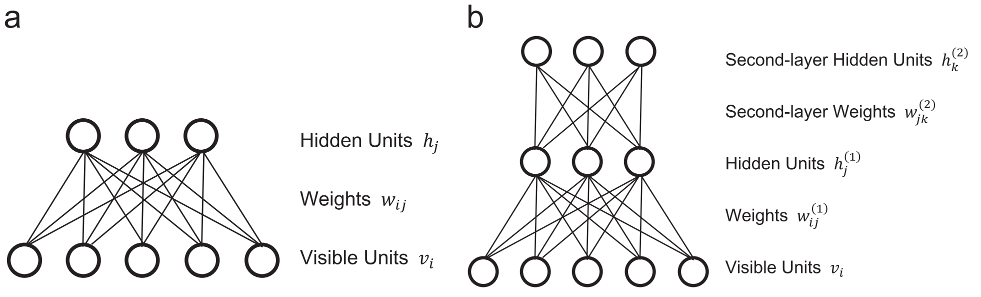


Fig. 1. Illustration of the models. (a) A single-layer structure. Both RBM and sparse RBM can be represented by this figure and the only difference is that the latter imposes a sparse firing constraint on the hidden units. The K-means and multiple firing K-means algorithms can also be represented by this figure, and the difference is that the former allows only one hidden unit fire at a time, while the latter allows more than one hidden units fire at a time and (b) a two-layer structure illustrating the sparse DBN and the hierarchical K-means model.

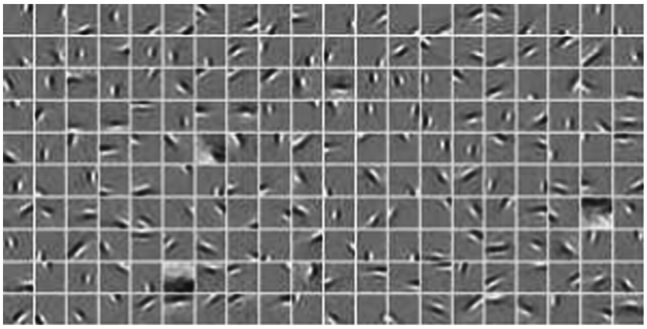


Fig. 2. Visualization of 200 first layer weight vectors of the sparse DBN. Each 14 × 14 patch corresponds to a weight vector.

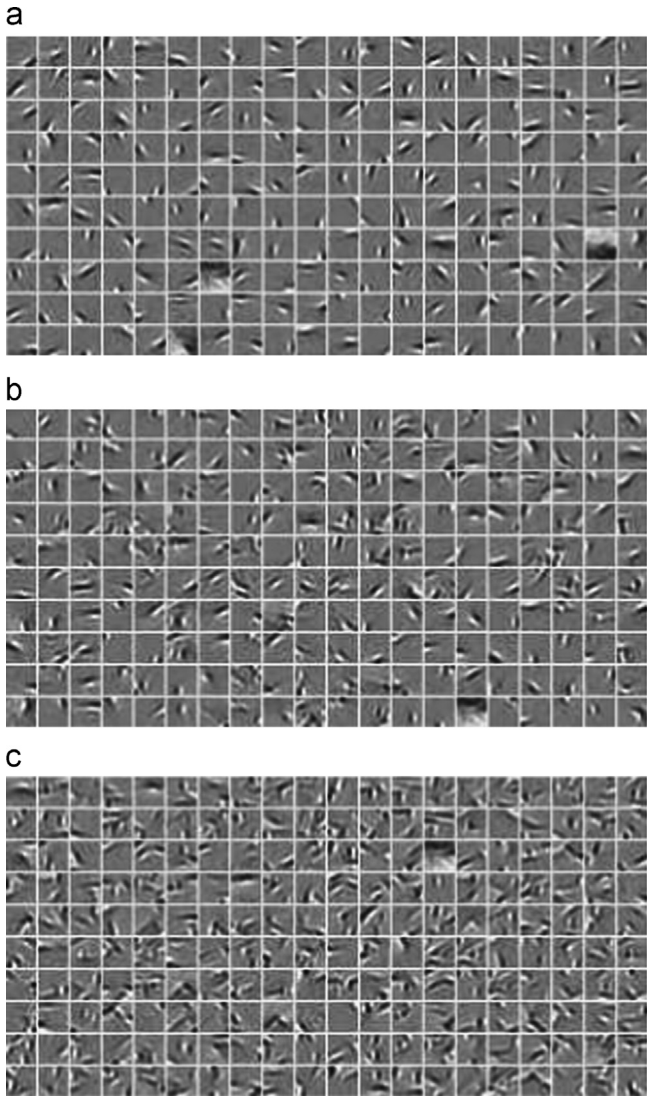


Fig. 3. Visualization of 200 second layer weight vectors of the sparse DBN. (a) $p=0.02$, (b) $p=0.03$, and (c) $p=0.04$.

similar to those of the first layer units. This observation suggests that the nonlinearity is not the only factor that contributes to the emergence of V2 neuron receptive fields in sparse DBN, and a relatively relaxed sparsity constraint on the second layer also makes critical contribution. Therefore, 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 algorithm

The goal of K-means algorithm is to partition the data set $\mathbf{v}_1,\dots,\mathbf{v}_N$ into K clusters. Denote the mean or centroid of cluster j by $\mathbf{w}_j$, where $j=1,\dots,K$, then the goal is to identify these $\mathbf{w}_j$. The learning algorithm is termed the *expectation maximization* (EM) algorithm, which consists of two iterative steps [19]:

- *E-step*: For each input $\mathbf{v}_n$ determine which cluster it belongs to. Mathematically, this amounts to determine $j^* = \arg\min_j \|\mathbf{v}_n - \mathbf{w}_j\|$.
- *M-step*. Update $\mathbf{w}_j$ for $j=1,\dots,K$ by taking the mean (centroid) of data assigned to cluster j. That is, $\mathbf{w}_j = \sum_{t=1}^{T}\mathbf{v}_t/T$ where $\mathbf{v}_t$ denotes the input assigned to cluster j.

After the algorithm converges, each data point $\mathbf{v}_n$ can be 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.

This algorithm can be implemented by biological systems with simple operations. In fact, the E-step can be implemented by a winner-takes-all circuits [20] and the M-step can be implemented by a Hebbian learning rule. To show the latter point, let us specify the following Hebbian plasticity rule for the change of a synaptic weight w_{ja} between h_j and v_{na} due to the presence of an input $\mathbf{v}_n$ [21]:

$$\Delta w_{ja} = h_j(v_{na} - w_{ja}).$$

Note that h_j can only take 0 or 1. Clearly, this rule modifies the weight w_{ja} for all a so they more match the input v_{na} when the output unit h_j is activated by $\mathbf{v}_n$. The synaptic weight is updated as follows:

$$w_{ja} = w_{ja} + \Delta w_{ja} = w_{ja} + (v_{na} - w_{ja}) = v_{na}.$$

For a set of inputs, the learning rule is

$$w_{ja} = w_{ja} + \frac{1}{T}\sum_{t=1}^{T}(v_{ta} - w_{ja}) = \sum_{t=1}^{T} v_{ta}$$

where t denotes the label of inputs that have activated unit h_j (making $h_j=1$). This is actually the E-step shown above.

3.2. Multiple firing K-means algorithm

Now we relax the sparsity constraint of the original K-means algorithm by allowing multiple hidden units fire together for an input. Specifically, for each input $\mathbf{v}_n$ we assign L clusters it belongs to, whose centroids are the nearest to the input. Mathematically, this amounts to determine a set $\Omega \subset V = \{1,\dots,K\}$ such that $|\Omega| = L$ and $\|\mathbf{v}_n - \mathbf{w}_s\| \le \|\mathbf{v}_n - \mathbf{w}_j\|$ for $s \in \Omega$ and $j \in V \backslash \Omega$.

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

$$h_j(\mathbf{v}_n) = \begin{cases} 1 & \text{if } \mathbf{v}_n \text{ belongs to cluster } j; \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Then there are always L hidden units firing, and given a desired sparsity level p, L can be simply determined as $L=pK$. For this reason this algorithm is called *multiple firing K-means algorithm*. The structure is also illustrated in Fig. 1(a). To learn this model, we take a similar EM approach as the original K-means algorithm. Convergence of this algorithm is stated in the following theorem.

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

$$J=\left\langle \sum_{j=1}^{K} h_j \|\mathbf{v}-\mathbf{w}_j\|^2 \right\rangle \tag{6}$$

until convergence.

Proof. In the E-step, $\mathbf{w}_j$ 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 the M-step, $\mathbf{h}$ is fixed. Notice that $\partial J/\partial \mathbf{w}_j = -2\langle h_j(\mathbf{v}-\mathbf{w}_j)\rangle$. Then this step is equivalent to taking $\partial J/\partial \mathbf{w}_j = 0$, which corresponds to minimization of J over $\mathbf{w}_j$. Therefore, each step results in a decrease of J until convergence.

Biologically implementation of this algorithm is similar to that for the standard K-means algorithm, as we discussed above. The only difference is that in the M-step, an L-winners-take-all circuit like those presented in [22,23] is necessary.

3.3. Hierarchical model

Similar to the sparse DBN, we can stack another multiple firing K-means model on top of the first-layer, i.e. it takes the output of the first layer as input and learns the second-layer centroids by freezing the first-layer centroids (see Fig. 1(b)). We term the resulting model a *hierarchical K-means model*.

4. Experiments

4.1. First layer results

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. We randomly extracted a large number of 14-by-14 patches from 10 natural images, which were preprocessed in the same way as in Section 2, that is, preprocessed by $1/f$ whitening and low-pass filtering in the frequency domain. At every iteration 50,000 patches were fed into the algorithm and the centroids got updated once. We determined whether the algorithm converged by checking the change of the loss function J in (6).

Fig. 4 illustrates the evolving history of J over iterations with L=3, 5, 7, 10, respectively. It is observed that after a few iterations the value of J reaches a relatively stationary state. The 200 centroids plotted in Fig. 5 were obtained with L=3 for 40 iterations. Clearly they are edge detectors, similar to the results of standard K-means [7,8], sparse DBN [5], and other sparse coding algorithms [2,4]. For L=5, 7, 10 the results were visually similar to this figure (results not shown).

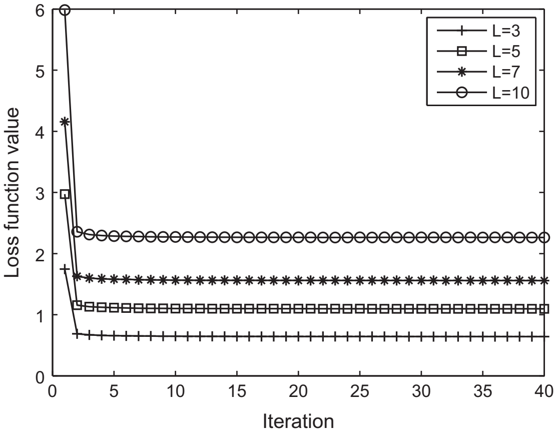


Fig. 4. Value of the loss function J over iterations.

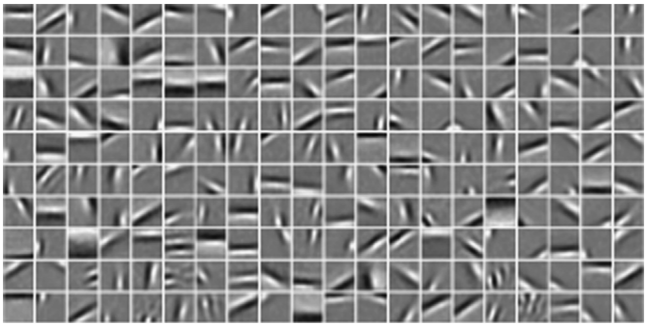


Fig. 5. Visualization of 200 first layer centroids of the hierarchical K-means model with L=3 (corresponding to p=0.015).

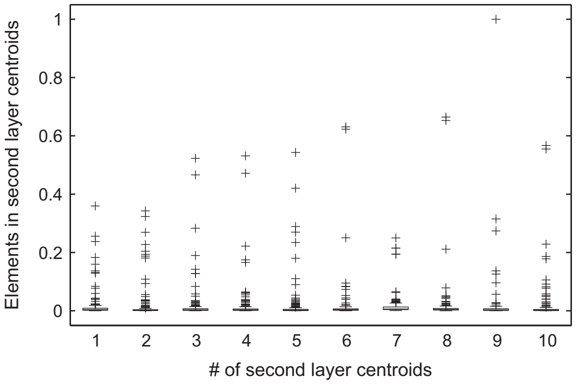


Fig. 6. Boxplot of elements in 10 random second-layer centroids.

4.2. Second layer results

We stacked a second layer of multiple firing K-means algorithm on top of the first layer. The second layer had 200 units and L was set to 10 (corresponding to p=0.05). After 20 iterations, the algorithm converged.

It was found that only a few elements in the learned second-layer centroids were significantly larger than zero. Fig. 6 shows the distribution of the elements in 10 randomly selected second-layer centroids with the *boxplot* method.[1] It is seen that most elements except a few are close to zero.

The second-layer centroids are visualized in Fig. 7 in the same manner as Fig. 3, i.e. they are visualized as weighted sums of the first-layer centroids. It is seen that the shapes of many second-layer centroids are like corners or conjunctions of edges, qualitatively in agreement with some V2 neuron properties [24].

4.3. Comparison with physiological results

To test the properties of the second-layer units obtained by the hierarchical K-means model, we generated a set of angle stimuli as shown in Fig. 8 [24]. Each stimulus was a 14-by-14 image patch representing an angle in $\{2\pi/M, 4\pi/M, \ldots, 2(M-1)\pi/M\}$ in

[1] The box has lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to show the extent of the rest of the data. Outliers denoted by "+" are data with values beyond the ends of the whiskers. A detailed description can be found in MATLAB documentation.

different orientations, which resulted in $M(M-1)$ different stimuli. See [24] for details. In addition, each stimulus was normalized to zero mean and unit variance. The angle stimuli in experiments were white (pixel value 1) in black background (pixel value 0). For better visualization, however, the pixel values are reversed in Fig. 8.

For each angle stimulus at a location, we first calculated the output of the first-layer units according to (5). As for the second-layer responses, we adjusted the definition of response for the purpose of comparing with physiological results. Instead of binarizing the distances between the second-layer weights and the first-layer outputs as in (5), we adopted the distances as the continuous output. In addition, to be consistent with the usual meaning of *responses* of real neurons (the more similar a stimulus to the receptive field, the higher responses will be induced by the stimulus), the output of a second layer unit with centroids $\mathbf{w}$ induced by an input $\mathbf{v}$ was defined as

$$r = C - \|\mathbf{w}-\mathbf{v}\|^2 \tag{7}$$

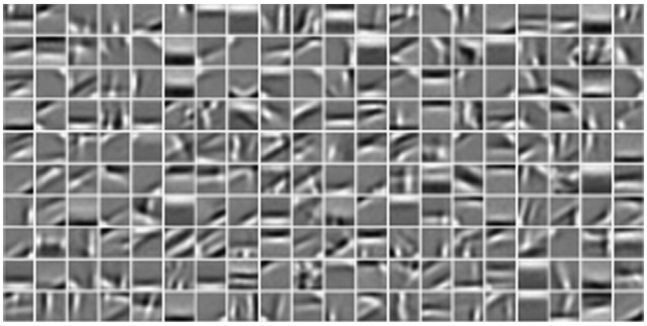


Fig. 7. Visualization of 200 s layer centroids of the hierarchical K-means model.

where $C = \max_{\mathbf{w},\mathbf{v}} \|\mathbf{w}-\mathbf{v}\|^2$ is a constant for ensuring non-negativity of the responses. The "max" operation was applied over all second-layer units on all angle stimuli at all positions.

To identify the "center" of the receptive field of each second-layer unit, we translated all stimuli densely over the 14×14 input image patch, and identified the position at which the maximum response was elicited. All measures were then taken with all angle stimuli centered at this position.

Fig. 8 shows the stimuli set with $M=24$ together with responses of six representative second-layer units. In each subplot, a small black square indicates the angle that induced overall peak response of the second-layer unit (or model V2 neuron) and the shaded patches indicate the angles that induced over 70% of the peak response. The angles are symmetric along the diagonal, so are the responses. There are neurons responsive to particular angles (top left), many angles (top middle and top right), a particular bar of an angle (bottom left) and either bar of an angle (bottom right). There are both single-group responses (bottom middle) and two-group responses (bottom right). We emphasize that these units are typical in our model.

To make a quantitative comparison between the simulation results and physiological results in [24], 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. 9. The definitions of the five quantities can be found in [24]. The physiological results and the sparse DBN results are also presented in the figure. It is seen that the hierarchical K-means model has produced similar results.

Another method for analyzing the properties of the model V2 neurons is to inspect their responses with small gratings covering different parts of the receptive fields. This is the method used in [25] where V2 neurons of monkeys were investigated. It was reported that some V2 neurons had uniform response

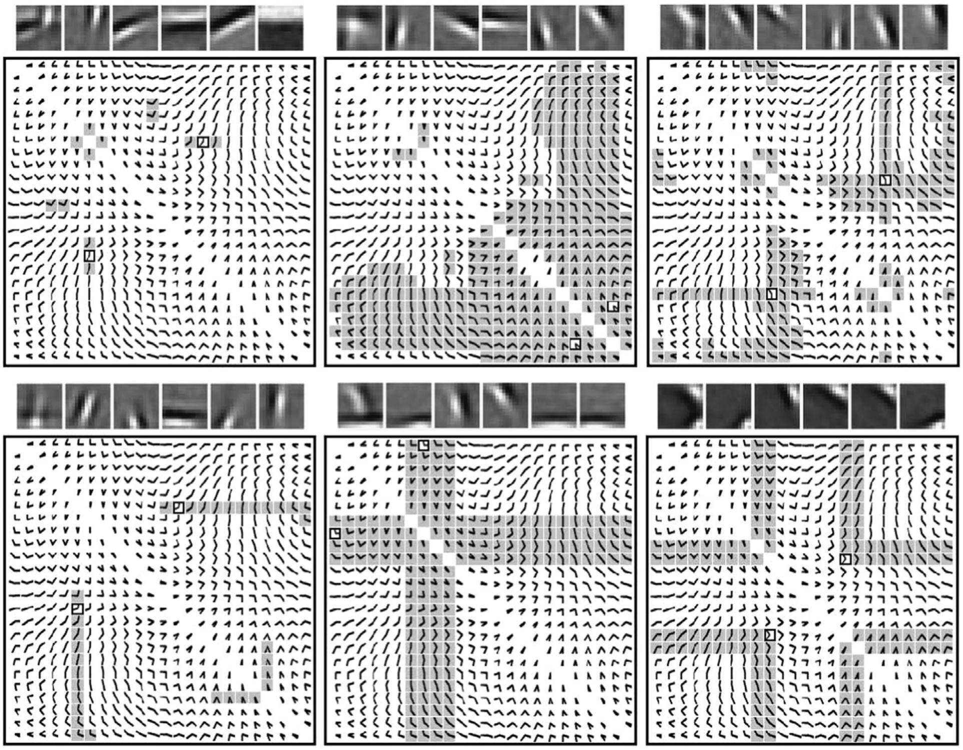


Fig. 8. Response profile of six example model V2 neurons on a set of angle stimuli. Top: The left most patch shows a model V2 neuron by taking the weighted sum of V1 simple cell receptive fields. The next five patches show the receptive fields of the model V1 simple cells that had strongest connections 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.

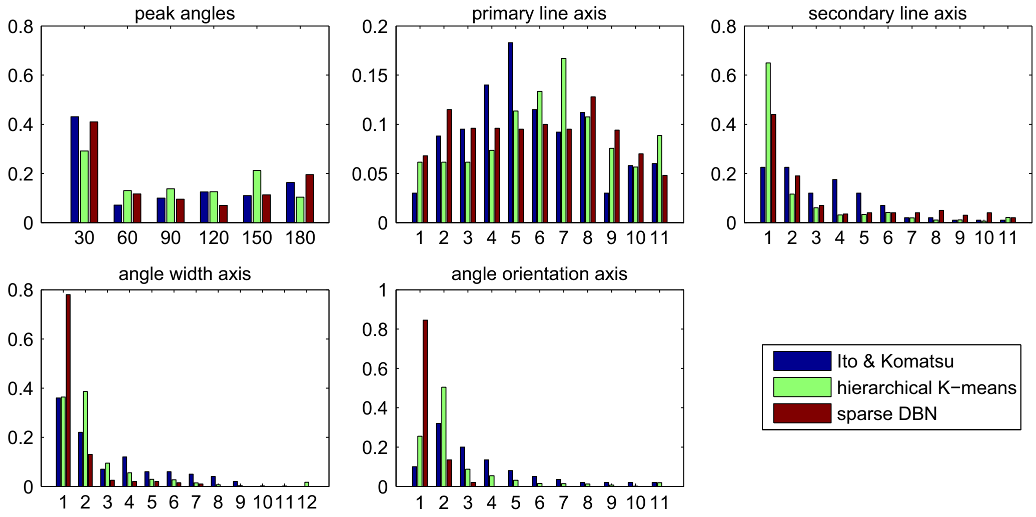


Fig. 9. Distribution of the response statistics over the angle stimuli. The five figures show respectively the distribution over (i) peak angle response, (ii) tolerance to primary line component, (iii) tolerance to secondary line component, (iv) tolerance to angle width, and (v) tolerance to angle orientation. See [5,24] for details. The physiological results and the sparse DBN results were extracted from Fig. 6 in [5]. Best viewed in color.

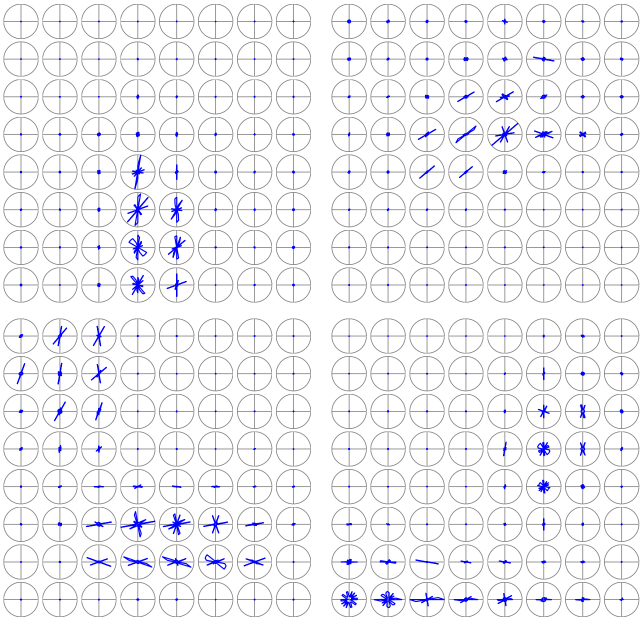


Fig. 10. Examples of space-orientation receptive field maps for four model V2 neurons. Responses of neurons are plotted in polar coordinate as a function of stimulus orientation at 64 positions arranged in square arrays in space. The center of each circle indicates the location in the receptive field and the radius indicates the maximum response. Orientation increases counterclockwise from 0° at the 12 o'clock position on each gray circle. Results for the orientation range between 0° to 180° were repeated to complete the polar plot in full circle.

characteristics across their receptive fields and some had nonuniform response characteristics.

We generated 18 sinusoidal gratings with different orientations (equal interval between 0° and 180°) in a circle with diameter 9 pixels. The surround of the receptive field was padded with zeros to make a larger patch of size 24-by-24, and each grating was translated within the patch, which would result in a 16-by-16 response matrix. Enlarging the receptive fields was to allow the gratings to move a little bit outside of the original receptive fields. At each location, 18 gratings were presented and consequently 18 responses were obtained. Responses of a model neuron were defined by (7) where C was the square of maximum distance of this neuron to all stimuli at all positions in the enlarged receptive field. Then the orientation tuning curves can be plotted in a polar coordinate (r, θ) where r stands for the response and θ stands for the orientation of the grating.

Table 1
Comparison of the computing time in seconds.

Models	V1	V2
Sparse DBN	2536.7 ± 21.0	2693.1 ± 37.3
Hierarchical K-means	68.9 ± 3.9	43.2 ± 4.2

Fig. 10 shows four example model V2 neurons. For each neuron only 8×8 tuning curves are presented, whose locations were sampled from the 16-by-16 response matrix with step size 2 on each side. Top left shows a model neuron with uniform response maps, that is, at all locations (except where there is little response), the neuron roughly tunes to a vertical grating. The other three neurons have nonuniform response maps. The top right one shows that at center location of the receptive field, the neuron tunes to a 135° grating but at the upper right corner it tunes to a 80° grating. The bottom two show two neurons whose receptive fields are “L” shape (left) and horizontally reversed “L” shape (right). These results are qualitatively consistent with the findings in the monkey brain [25]. However, these results contain more noise and the response maps are not as smooth as real V2 neurons' response maps (cf. Fig. 1 in [25]), which is partly due to the brutal nonlinearity used at the first layer output, that is, the binarization operation as described in (5).

4.4. *Computational efficiency*

Both the hierarchical K-means model and the sparse DBN can produce similar results, then how about their computational efficiency? This is not a question in computational neuroscience but is important in engineering applications, considering that deep learning networks have become a hot topic [14,26,27]. In fact, an excellent computer vision model, CDBN [26], was build on the sparse DBN.

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 significantly for producing visually similar results. Table 1 shows the computing time of the two algorithms on a computer (Intel Core i5-2320 3 GHz ×4, RAM 8GB), averaged over 10 trials, for producing visually similar results to Figs. 2, 3(c), 5 and 7, 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 [15]. 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, the first layer learning terminated after 40 iterations and the second-layer learning terminated after 20 iterations 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 model is tens of times faster than the sparse DBN.

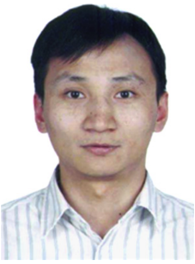
5. Concluding remarks

There are many models capable of producing edge-like structure that resembles the receptive fields of V1 neurons in the brain, but few have been shown capable of reproducing the edge conjunction structure of the receptive fields of V2 neuron, except the sparse DBN. In the paper a hierarchial K-means model is presented as an alternative to the sparse DBN for modeling the response properties of V2 neurons. After unsupervised training on natural images, the proposed model exhibited properties that qualitatively matched physiological data recorded in monkeys. Compared with the sparse DBN, the proposed model is more biologically feasible and computationally efficient.

Due to its biological plausibility, the proposed model may be employed to interpret the mechanisms of visual processing in the brain. Due to its computational efficiency, it is worth further investigations for computer vision. For example, it would be interesting to extend it to learn object parts, like the convolutional DBN [26]. We expect that integration of convolution into this model will lead to more powerful models, which might be useful on some challenging computer vision tasks.

References

[1] D.H. Hubel, T.N. Wiesel, Receptive fields and functional architecture in two nonstriate visual areas (18 and 19) of the cat, J. Neurophysiol. 28 (1965) 229–289.
[2] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (1996) 607–609.
[3] B.A. Olshausen, D.J. Field, Sparse coding with an overcomplete basis set: a strategy employed by V1? Vis. Res. 37 (23) (1997) 3311–3325.
[4] A.J. Bell, T.J. Sejnowski, The "independent components" of natural scenes are edge filters, Vis. Res. 37 (23) (1997) 3327–3338.
[5] H. Lee, C. Ekanadham, A. Ng, Sparse deep belief net model for visual area V2, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems, Vancouver, Canada, 2007.
[6] M. Ranzato, Y.-L. Boureau, Y. LeCun, Sparse feature learning for deep belief networks, in: J. Platt, D. Koller, Y. Singer, S. Roweis (Eds.), Advances in Neural Information Processing Systems (Advances in Neural Information Processing Systems), vol. 20, Vancouver, Canada, 2007.
[7] A. Coates, H. Lee, A.Y. Ng, An analysis of single-layer networks in unsupervised feature learning, in: G. Gordon, D. Dunson, M. Dudik (Eds.), Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS), Ft Lauderdale, FL, 2011.
[8] A.M. Saxe, M. Bhand, R. Mudur, B. Suresh, A.Y. Ng, Unsupervised learning models of primary cortical receptive fields and receptive field plasticity, in: J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 24, 2011, pp. 1971–1979.
[9] Y. Karklin, M.S. Lewicki, A hierarchical Bayesian model for learning nonlinear statistical regularities in nonstationary natural signals, Neural Comput. 17 (2005) 397–423.
[10] Y. Karklin, M.S. Lweicki, Emergence of complex cell properties by learning to generalize in natural scenes, Nature 457 (2009) 83–86.
[11] P. Qi, X. Hu, Learning nonlinear statistical regularities in natural images by modeling the outer product of image intensities, Neural Comput. accepted for publication.
[12] M. Riesenhuber, T. Poggio, Hierarchical models of object recognition in cortex, Nat. Neurosci. 2 (1999) 1019–1025.
[13] C. Cadieu, M. Kouh, A. Pasupathy, C.E. Connor, M. Riesenhuber, T. Poggio, A model of V4 shape selectivity and invariance, J. Neurophysiol. 98 (2007) 1733–1750.
[14] G.E. Hinton, S. Osindero, Y.-W. Teh, A fast learning algorithm for deep belief nets, Neural Comput. 18 (2006) 1527–1554.
[15] C. Ekanadham, Sparse deep belief net models for visual area V2 (Undergraduate honors thesis), Stanford University, 2007.
[16] G.E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Comput. 14 (2002) 1771–1800.
[17] K. Sohn, D.Y. Jung, H. Lee, A. O. Hero III, Efficient learning of sparse, distributed, convolutional feature representations for object recognition, in: Proceedings of the International Conference on Computer Vision, 2011.
[18] G.E. Hinton, A practical guide to training restricted Boltzmann machines, 2010.
[19] C.M. Bishop, et al., Pattern Recognition and Machine Learning, Springer, New York, 2006.
[20] R. Coultrip, R. Granger, G. Lynch, A cortical model of winner-take-all competition via lateral inhibition, Neural Netw. 5 (1) (1992) 47–54.
[21] P. Dayan, L.F. Abbott, Theoretical Neuroscience: Computational and Mathematical Modeling of Neural, The MIT Press, Cambridge, Massachusetts and London, England, 2001.
[22] X. Hu, J. Wang, An improved dual neural network for solving a class of quadratic programming problems and its k-winners-take-all application, IEEE Trans. Neural Netw. 19 (12) (2008) 2022–2031.
[23] X. Hu, B. Zhang, A new recurrent neural network for solving convex quadratic programming problems with an application to the k-winners-take-all problem, IEEE Trans. Neural Netw. 20 (4) (2009) 654–664.
[24] M. Ito, H. Komatsu, Representation of angles embedded within contour stimuli in area V2 of macaque monkeys, J. Neurosci. 24 (13) (2004) 3313–3324.
[25] A. Anzai, X. Peng, D.C.V. Essen, Neurons in monkey visual area V2 encode combinations of orientations, Nat. Neurosci. 10 (10) (2007) 1313–1321.
[26] H. Lee, R. Grosse, R. Ranganath, A. Y. Ng, Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations, in: Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada, 2009, pp. 609–616.
[27] Q.V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G.S. Corrado, J. Dean, A.Y. Ng, Building high-level features using large scale unsupervised learning, in: Proceedings of the 29th International Conference on Machine Learning, Edinburgh, Scotland, GB, 2012, pp. 81–88.



Xiaolin Hu received the B.E. and M.E. degrees in Automotive Engineering from Wuhan University of Technology, Wuhan, China, and the Ph.D. degree in Automation and Computer-Aided Engineering from The Chinese University of Hong Kong, Hong Kong, China, in 2001, 2004, 2007, respectively. He is now an Assistant Professor at the 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, China. His current research interests include artificial neural networks and computational neuroscience. He is an Associate Editor of the IEEE Transactions on Neural Networks and Learning Systems.

Jianwei Zhang is a Professor and the head of TAMS, Department of Informatics, University of Hamburg, Germany. He received both his Bachelor of Engineering (1986, with distinction) and Master of Engineering (1989) from the Department of Computer Science of Tsinghua University, Beijing, China, and his PhD (1994) at the Institute of Real-Time Computer Systems and Robotics, Department of Computer Science, University of Karlsruhe, Germany. His research interests are cognitive robotics, robot autonomous learning, robot vision, bio-inspired robot systems, service robot applications and human–robot interaction. In these areas he has published over 300 journal and conference papers, technical reports, six book chapters and two research monographs. He has been the PI and Coordinator of numerous EU and German National Projects on cognitive robotics and service robots. He has received several awards, including the IEEE ROMAN Best Paper Award in 2002 and the IEEE AIM Best Paper Award 2008. He is in the organization committee of numerous international conferences, including some future ones such as IEEE ICRA 2011 Program Co-chair, IEEE MFI 2012 General Chair, IROS 2015 General Chair. He was elected as the AdCom member of IEEE RAS 2013–2015.

Peng Qi received the B.E. degree in Computer Software from Tsinghua University, Beijing, China in 2012. He is currently a Research Assistant with the State Key Laboratory of Intelligent Technology and Systems, the Tsinghua National Laboratory for Information Science and Technology (TNList), and the Department of Computer Science and Technology, Tsinghua University, Beijing. His current research interests include artificial intelligence and computer vision.

Bo Zhang graduated from the Department of Automatic Control, Tsinghua University, Beijing, China, in 1958. He is now a Professor of the 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, China, and a fellow of Chinese Academy of Sciences. His research interests include artificial intelligence, robotics, intelligent control and pattern recognition. He has published about 150 papers and three monographs in these fields.