Hierarchical Discriminative Sparse Coding via Bidirectional Connections

Zhengping Ji, Wentao Huang, Garrett Kenyon and Luis M. A. Bettencourt

***Abstract*— Conventional sparse coding learns optimal dictionaries of feature bases to approximate input signals; however, it is not favorable to classify the inputs. Recent research has focused on building discriminative sparse coding models to facilitate the classification tasks. In this paper, we develop a new discriminative sparse coding model via bidirectional flows. Sensory inputs (from bottom-up) and discriminative signals (supervised from top-down) are propagated through a hierarchical network to form sparse representations at each level. The ℓ_0-constrained sparse coding model allows highly efficient online learning and does not require iterative steps to reach a fixed point of the sparse representation. The introduction of discriminative top-down information flows helps to group reconstructive features belonging to the same class and thus to benefit the classification tasks. Experiments are conducted on multiple data sets including natural images, handwritten digits and 3-D objects with favorable results. Compared with unsupervised sparse coding via only bottom-up directions, the two-way discriminative approach improves the recognition performance significantly.**

I. Introduction

Sparse coding of visual inputs draws considerate research attention in recent years. Specially, it caused great interest to find the sparse visual representations and to learn the dictionary of feature bases from unlabeled raw data (e.g., [1]–[4]). The objective of sparse coding is to approximate an input signal $\mathbf{X} \in \mathbb{R}^{n\times 1}$ using a linear combination of over-complete bases $\mathbf{W} = [\mathbf{W_1}, \mathbf{W_2}, ..., \mathbf{W_m}] \in \mathbb{R}^{n\times m}$ with the sparse coefficient vector $\mathbf{Y} \in \mathbb{R}^{m\times 1}$ ($m \geq n$). The over-complete bases can either be a set of pre-defined functions, such as Gabor wavelets, or be developed from a set of input examples. For the case of given bases, the sparse coding is posed as an optimization problem to minimize $||\mathbf{Y}||_\mathbf{0}$ or $||\mathbf{Y}||_\mathbf{1}$ through the greedy search (e.g., Matching Pursuit (MP) [5], Orthogonal Matching Pursuit (OMP) [6]) or the convex optimization (e.g., Basis Pursuit (BP) [7], FOCUSS [8] and ℓ_1 regularization [9], [10]). The pre-defined bases, however, lack the dictionary adaptiveness to the data, thus do not perform well in the reconstructive manner.

A majority of sparse coding algorithms seek to learn both over-complete bases $\mathbf{W}$ and sparse vector $\mathbf{Y}$. A well-known sparse coding algorithm was proposed in Olshausen and Field 1997 [3] to minimize the reconstruction error $||\mathbf{X}-\mathbf{WY}||_\mathbf{2}^\mathbf{2}$ with a sparse constraint to independent prior $\mathbf{Y}$. Subsequently, a number of iterative batch learning algorithms have been developed for sparse coding, mainly focusing on various cost penalties and sparse constraints [11]–[14]. These studies are purely generative models, where learned dictionaries are used to reconstruct the input effectively. Even though some of the above sparse coding models have been applied to the discriminative tasks, e.g., image classification, the approach was limited to the simple combination of generative sparse coding and a stand-alone classifier. Recent research is aimed at integrating both input reconstruction and class discrimination within the sparse coding paradigm. Mairal et al. 2008 [15] proposed a supervised sparse learning framework, where a logistic loss function regarding input class was added to the reconstruction cost of sparse coding. Bradley and Bagnell [16] introduced a differential sparse prior rather than the conventional ℓ_1 norm to learn the dictionary of feature bases. Yet, no existing work used top-down connections as a natural information flow to construct the discriminative sparse coding model.

Another important progress of sparse coding is to model the hierarchical visual cortex [4], [17], [18]. Some aforementioned articles have demonstrated the development of V1-like features using natural images [3], [14], but limited sparse coding algorithms were extended to learn deep hierarchical structures, due to difficulties in inferring the states of the hidden layers. A few existing sparse coding networks [19], [20] approximate internal hidden states as a function of feed-forward and perhaps lateral connections, and performed greedy learning layer by layer. Still, discriminative information (e.g, top-down feedback) is not involved in the hierarchical learning of bases and sparse representation. Back propagation [21] once provided a powerful way to propagate the top-down signals, but it suffers the problems of local minimum, expensive computation and poor performance in multiple hidden layers.

In this paper, we develop a new discriminative sparse coding model via bidirectional information flows. Sensory inputs (from bottom-up) and discriminative signals (supervised from top-down) are propagated through a hierarchical network to form sparse representations at each level. A series of advances have been made in this paper: (1) The ℓ_0-constrained sparse coding model allows highly efficient online learning and does not require iterative steps to reach a fixed point of sparse representation. (2) Using discriminative top-down connections, reconstructive features belonging to the same class are grouped together, shaping the topographic

Zhengping Ji and Luis M. A. Bettencourt are with the Theoretical Division T-5, Los Alamos National Laboratory, Los Alamos, NM (email: jizhengp,lmbettencourt@gmail.com).

Wentao Huang is with the Department of Neuroscience, John Hopkins University, Baltimore, MD (email: hwtsch@gmail.com)

Garrett Kenyon is with the Physics Division P-21, Los Alamos National Laboratory, Los Alamos, NM (email: garkenyon@gmail.com)

This work was supported by the LANL LDRD program under grant 20090006DR.

areas to facilitate the classification tasks. (3) The proposed sparse coding model can be implemented in a divide-and-conquer manner within a hierarchical architecture, providing a solution to learn a deep network with feed-back connections.

The paper is organized as follows. Sec. II describes the network model and algorithms for the discriminate sparse coding. Sec. III discusses the development of deep hierarchical networks using the proposed learning procedure. Experimental results and conclusions are presented in Sections IV and V, respectively.

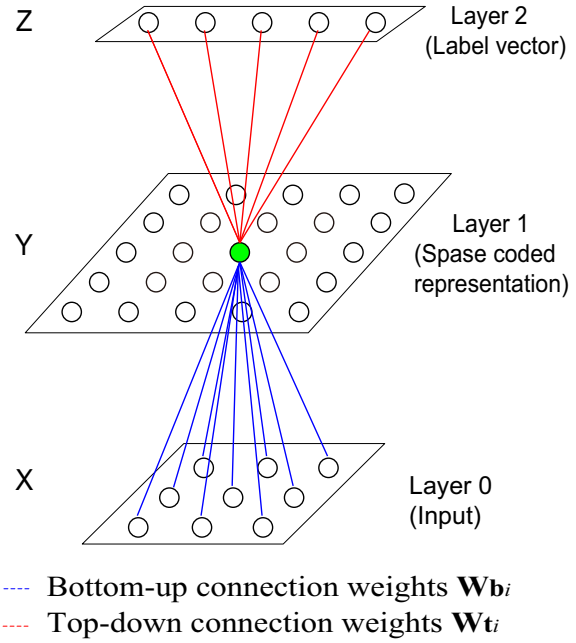


Fig. 1. A sparse coding network with one hidden layer (best viewed in color). Only connections to a centered cell are shown, but all the other cells in the hidden layer have the same default connections.

II. Bidirectional Discriminative Sparse Coding

The proposed sparse coding model contains a hierarchy of layers, each with a set of cells, arranged in a 2-D grid. It is a fully connected network from the sensory input to the corresponding label vector [1]. Local connectivity is possible and subject to the future studies. We first consider a simplest network structure with one hidden layer only, and learning of a deeper network structure will be addressed in Sec. III using the same sparse coding algorithm described here.

As shown in Fig. 1, each hidden cell i is connected with two types of connection weights:

1) Bottom-up weight vector $\mathbf{w_b}_i$ that links connections from the previous layer.
2) Top-down weight vector $\mathbf{w_t}_i$ that links connections from the next layer.

To learn an over-complete dictionary with m cells in the hidden layer, the sparse coding scheme minimizes the reconstruction error with both bottom-up input $\mathbf{X} \in \mathbb{R}^{n\times 1}$ and top-down input of label vector $\mathbf{Z} \in \mathbb{R}^{p\times 1}$, constrained by a fixed sparsity factor L.

$$\min_{\mathbf{Y},\mathbf{W_b},\mathbf{W_t}} \left\{ E = \frac{1-\alpha}{2}||\mathbf{X}-\mathbf{W_b}\mathbf{Y}||_2^2 + \frac{\alpha}{2}||\mathbf{Z}-\mathbf{W_t}\mathbf{Y}||_2^2 \right\} \quad \text{s. t.} \quad ||\mathbf{Y}||_0 \le L \tag{1}$$

where $0 \le \alpha \le 1$ is a constant parameter to control the influence of top-down contribution. $\mathbf{W_b} = [\mathbf{W_b}_1, \mathbf{W_b}_2, ..., \mathbf{W_b}_m] \in \mathbb{R}^{n\times m}$ and $\mathbf{W_t} = [\mathbf{W_t}_1, \mathbf{W_t}_2, ..., \mathbf{W_t}_m] \in \mathbb{R}^{p\times m}$ are bottom-up and top-down connection weights to be learned. $||\mathbf{Y}||_\mathbf{0}$ denotes the ℓ_0-norm of vector $\mathbf{Y}$, measuring sparsity by counting the number of non-zero elements in a vector.

It is noted that the overall problem of sparse coding with ℓ_0 constraint is non-convex, and in some existing studies the ℓ_0 norm was replaced by the ℓ_1 norm for convex optimization purpose, but yielding expensive computations. In this work, we used the ℓ_0 norm for the discriminative sparse coding based on its three advantages: (1) The ℓ_0 norm is the true sparsity measure of a representation, superior to ℓ_1-based sparsity in terms of similarity to receptive fields of neurons in the visual cortex [22]. (2) The ℓ_0 norm leads to an online close-form solution to infer the sparse representation without iterations. (3) The ℓ_0 norm provides a solution to learn a deep network with feed-back connections, which will be discussed in detail in Sec. III. In practice, moreover, extensive studies have established that if the sought solution $\mathbf{Y}$ is sparse enough, meaning that L is small, the ℓ_0 norm can recover the sub-optimality well (e.g., MOD [11] and K-SVD [13]). In this study, L is set to 1 accordingly.

A. Learning rule

Eq. 1 can be solved by a gradient descent strategy as follows:

$$\begin{cases} \dfrac{d\mathbf{Y}}{dt} = -\dfrac{\partial E}{\partial \mathbf{Y}} \\ \dfrac{d\mathbf{W_b}}{dt} = -\dfrac{\partial E}{\partial \mathbf{W_b}} \\ \dfrac{d\mathbf{W_t}}{dt} = -\dfrac{\partial E}{\partial \mathbf{W_t}} \end{cases} \tag{2}$$

where we have

$$\Delta\mathbf{Y} = (1-\alpha){\mathbf{W_b}}^T(\mathbf{X}-\mathbf{W_b}\mathbf{Y}) + \alpha{\mathbf{W_t}}^T(\mathbf{Z}-\mathbf{W_t}\mathbf{Y}) \tag{3}$$

In our case, $||\mathbf{Y}||_\mathbf{0}$ is constrained to be 1, meaning that only one component of vector $\mathbf{Y}$ is allowed to be active with non-zero response. We assume the k-th component is active and all others are set to 0. Eq. 3 can be rewritten as

$$\Delta y_k = (1-\alpha)\ \mathbf{W_b}_k^T\mathbf{X} + \alpha\ \mathbf{W_t}_k^T\mathbf{Z} - (1-\alpha)\ \mathbf{W_b}_k^T\mathbf{W_b}_k\ y_k - \alpha\ \mathbf{W_t}_k^T\mathbf{W_t}_k\ y_k \tag{4}$$

We normalize each column (i.e., $\mathbf{W_b}_i$ and $\mathbf{W_t}_i$) in ℓ_2 norm, such that $|\mathbf{W_b}_i||_2 = 1$ and $||\mathbf{W_t}_i||_2 = 1$. Thus

$$\Delta y_k = (1-\alpha)\mathbf{W_b}_k^T\mathbf{X} + \alpha\mathbf{W_t}_k^T\mathbf{Z} - y_k \tag{5}$$

[1]The label vector is composed of a number of motor cells, each presenting one discriminative class identity.

When $\mathbf{Y}$ reaches a fixed point, meaning that Δy_k=0,

$$\alpha \mathbf{W_b}_k^T \mathbf{X} + (1-\alpha) \mathbf{W_t}_k^T \mathbf{Z} - y_k = 0 \tag{6}$$

and equally

$$y_k = (1-\alpha) \mathbf{W_b}_k^T \mathbf{X} + \alpha \mathbf{W_t}_k^T \mathbf{Z} \tag{7}$$

Determination of the index k here entails a ranking in $\mathbf{Y}$ space, where the maximum component is pooled:

$$k = \arg \max_{1 \le i \le m} y_i(t) \tag{8}$$

The proposed sparse coding model has a closed-form solution for the hidden variable $\mathbf{Y}$ via the ℓ_0 constraint, where no iterations is required. In this sense, the learning efficiency is boosted dramatically.

The learning rule of connection weights $\mathbf{W_b}$ and $\mathbf{W_t}$ can be derived from Eq. 2 coordinately:

$$\begin{aligned} & \Delta \mathbf{W_b} = \eta \ (\mathbf{X} - \mathbf{W_b Y}) \ \mathbf{Y^T} \\ \Rightarrow \quad & \Delta \mathbf{W_b}_k = \eta \ (\mathbf{X} - \mathbf{W_b}_k \ y_k) \ y_k \end{aligned} \tag{9}$$

and

$$\begin{aligned} & \Delta \mathbf{W_t} = \eta \ (\mathbf{Z} - \mathbf{W_t Y}) \ \mathbf{Y^T} \\ \Rightarrow \quad & \Delta \mathbf{W_t}_k = \eta \ (\mathbf{Z} - \mathbf{W_t}_k \ y_k) \ y_k \end{aligned} \tag{10}$$

By controlling the parameter α for top-down influence, the sparse coding model can easily adjust the supervised or unsupervised learning mode. When $\alpha = 0$, it becomes a pure unsupervised learning.

The learning rate of weight adaption is determined by a plasticity function:

$$\eta = \frac{1 + \mu(n_k)}{n_k}, \tag{11}$$

where $\mu(n_k)$ is the plasticity function depending on the maturity of cell k. The cell maturity increments as $n_k \leftarrow n_k + 1$ every time a cell updates its weights, starting from zero. We use the following three-sectioned profile for $\mu(n_k)$:

$$\mu(n_k) = \begin{cases} 0 & \text{if } n_k \le t_1, \\ c(n_k - t_1)/(t_2 - t_1) & \text{if } t_1 < n_k \le t_2, \\ c + (n_k - t_2)/r & \text{if } t_2 < n_k, \end{cases} \tag{12}$$

in which, parameters $t_1 = 20$, $t_2 = 200$, $c = 2$, $r = 2000$ in our implementation. Given the small n, the multi-sectional function $\mu(n)$ performs straight average $\mu(n) = 0$ to reduce the error coefficient for earlier estimates. Then, $\mu(n)$ enters the rising section from t_1 to t_2 linearly, where cells compete for the different partitions by increasing their learning rates for faster convergence. Finally, n enters the third section for long-term adaptation section: $\mu(n)$ increases at a rate of $1/r$ constantly. As discussed in [23], this kind of plasticity scheduling is more suited for practical signals with unknown non-stationary statistics, where the distribution does not follow *i.i.d* assumption in all the temporal phase.

B. Online learning algorithm

Because the proposed discriminative sparse coding algorithm does not require iterative search in representation space nor compute the second order statistics, it has high learning efficiency. Given each n-dimensional input $\mathbf{x}(t)$ and p-dimensional label vector $\mathbf{z}(t)$, the system complexity for updating m neurons is $O(mn + mp)$. It is not even a function of the number of inputs t, due to the nature of online incremental learning. The overall online learning procedure is summarized in Algorithm 1. In this single hidden layer case, the update of responses and bottom-up weights is similar to what is described in [24].

Algorithm 1 Learning a single hidden layer of the proposed hierarchical discriminative network, at each time t

Require: The bottom-up input $\mathbf{x}(t)$ and top-down input $\mathbf{z}(t)$.
Require: The connection matrices $\mathbf{W_b}$ and $\mathbf{W_t}$.
Require: The layer-specific parameter α.

1: Normalize each column of connection matrices $\mathbf{W_b}$ and $\mathbf{W_t}$ and compute candidate response of each neuron i using Eq. 7.
2: Pool the neuron k with maximum response (i.e., Eq. 8) and set the responses of other neurons to be zero.
3: Update the number of hits (cell age) n_k for the winning neuron k: $n_k \leftarrow n_k + 1$, and compute $\mu(n_k)$ by the amnesic function in Eq. 12.
4: Determine the learning rate of the winning neuron k by Eq. 11.
5: Update the synaptic weights of winning neurons using Eqs. 9 and 10.
6: All other neurons keep their ages and weight unchanged.

C. Discriminative top-down connection

Top-down connections propagate discriminative information originated from the label vector. What is the advantage of the proposed discriminative sparse coding rather than the purely reconstructive ones? In this section, we will discuss the functional role of top-down connections, which was first described in [24].

Given the sparse coding model described in Sec. II-A, we assume that the bottom-up input space $\mathbb{X}$ is a high-dimensional manifold composed of relevant subspace $\mathbb{R}$ and irrelevant subspace $\mathbb{I}$, where the "relevance" is with respect to distinguish the data for class labels. Introducing the top-down connection provides a new subspace $\mathbb{Z}$ to boost relevant information and thus recruit more cells spread along the relevant space for the discrimination purpose.

Fig. 2 illustrates this top-down connection role. As shown in Fig. 2(d), after the variance boosting via top-down connections, the cells spread along in the way to partition the classes favorably. But before that, the classes in Fig. 2(b) are mixed in the bottom-up subspace $\mathbb{X}$.

D. Lateral spatial pooling

It is known that ℓ_0 norm max pooling without smoothness (similar to vector quantization) may cause a number of

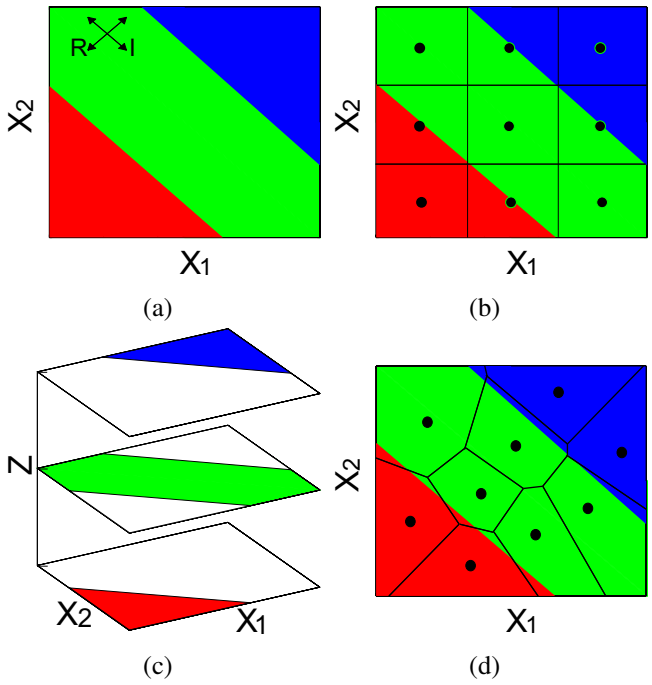


Fig. 2. Illustration of discriminative top-down connections (best viewed in color). (a) Consider two dimensional (x_1 and x_2) sensory inputs, which are assumed to have uniform densities and fall into one of the three class areas: "blue", "green" and "red". The "relevant" dimension (i.e., important to distinguish label outputs) and "irrelevant" dimensions are presented in the upper left legend, which are linear in this case. (b) The effect of sparse coding in bottom-up space: 9 cells are used to partition the space, however, resulting in the class boundaries mixed. (c) Top-down connections boost the variance of relevant subspace in the cell input, and thus recruit more cells along the relevant direction for better discrimination. (d) The effect of sparse coding in the boosted space. When embedding back into two dimensions, the partition boundaries now line up with the class boundaries and data that falls into a given partition is mostly from the same class. Figure is adapted from [24].

features to be recruited by the noises and outliers. An example is shown in Fig. 3. Consider two dimensional (x_1 and x_2) sensory inputs, which are assumed to have specific data densities along with a small number of noises and outliers. Four cells are used to model the data distribution. Given the max pooling of neural responses based on the correlation of feature vectors and data points (i.e., Eq. 7), two feature vectors are unfortunately recruited by a data point with noise (e.g., in "magenta") and outliers (e.g., in "cyan"), while the other two feature vector present the main density of data distribution (in "blue").

A regulation is thus in need to pool the distracted cells towards the main data distribution. In this paper, we use the lateral smoothness to reach this goal: when the max-pooled cell k is updated, its 3×3 neighboring cells becomes updated as well, using the same learning rule in Eq. 9. As the main density of data distribution (e.g., "blue" points in Fig. 3) updates their corresponding feature vectors frequently, the noise and outlier vectors are gradually pooled towards modeling the main data distribution. By this way, the proposed network gets immune to the noises and outliers.

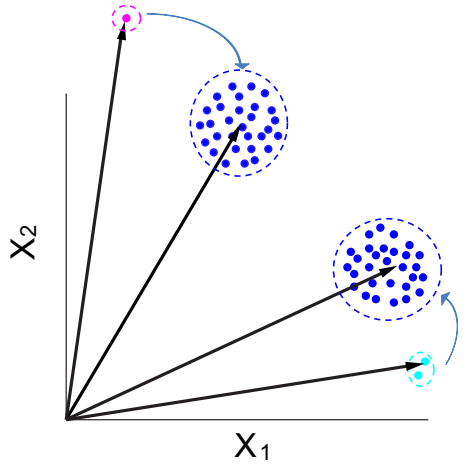


Fig. 3. An illustration of lateral pooling (best viewed in color).

Further recall the role of top-down connections in Sec II-C, which make the lateral pooling above apply within cells belonging to the same class, and thereby forms the class-specific topographic groups (discussed in detail in [24]). That is, based on the availability of cells, the weight features represented for the same class are grouped together to reduce complexity of discriminative boundaries and lead to a better recognition ability. Multiple experiments in Sec. IV are about to verify this grouping phenomenon.

III. Learning Deep Structure

Learning a deep network is known to be difficult. To develop the feature weights along layers, the good inference or approximation of layer representation is first required, yet encounter the obstacles such as "explaining way" and expensive computations [25]. Traditional learning strategy applies gradient descent search using back propagation, but empirically results in poor solutions for networks with multiple layers. Recently, the feed-forward layer-wise learning of deep networks has become popular, mainly based on the contrastive divergence learning of restricted Boltzmann machines (e.g., [26]) or the encoder-decoder architecture (e.g., [20]).

A. Divide and conquer

The proposed hierarchical architecture contains bidirectional (bottom-up and top-down) information flows, thus the feed-forward layer-wise learning is not applicable. In our case, inference of internal states becomes harder due to the recurrent structure: inference of the current layer state not only relies on the previous layer but also the next one. To solve this problem, we use a divide-and-conquer method. As Eq. 7 requires no searching iterations, we can divide the equation into two parts: bottom-up activation y_{k_b} and top-down activation y_{k_t}, where $k_b = \arg\max_{1\le i\le m}\{\mathbf{W_b}_i^T\mathbf{X}\}$ and $k_t = \arg\max_{1\le i\le m}\{\mathbf{W_t}_i^T\mathbf{Z}\}$. Note that the max pooling index k_b may not equal to k_t.

The network then freezes the connection weight and computes the bottom-up and top-down activations throughout the network via separate information flows. That means, originated from input, bottom-up activation of the current layer l is computed in terms of the bottom-up activation of the previous layer $l-1$. On the other hand, originated from class labels, top-down activation of the current layer l is computed in terms of the top-down activation of the previous layer $l+1$. After the two-way activations are generated at every layer, we add the two parts together given the parameter α and repeat the max pooling. The network weight is then updated given the post-pooling activation via Eqs. 9 and 10. Fig. 4 illustrates the divide-and-conquer process for the hierarchical deep network.

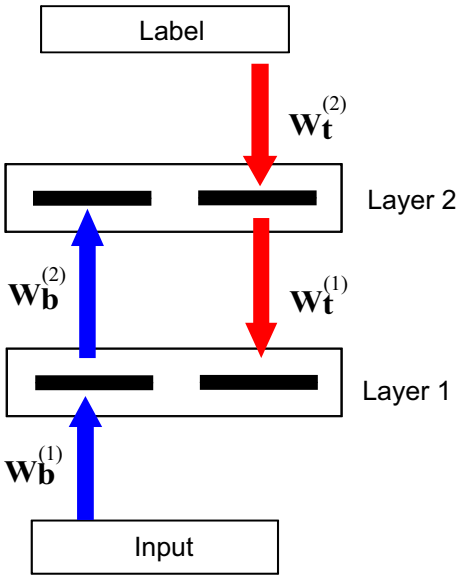


Fig. 4. Divide-and-conquer method to update hierarchical neural structures (best viewed in color). In each layer, the inference of internal states is divided into bottom-up part (originated from input) and top-down part (originated from label) separately. After the activation in each part is computed throughout all the layers, the network is updated based on their combined activation.

B. Initialization and pre-training

The network weights are initialized with random samples from the data. The number of layers and per-layer cells depends on a specific task. When a layer contains a large number of cells (e.g., 40×40), we need to pre-train the network with a smaller size (e.g., 20×20) first and then scale up the entire size. The pre-trained weights are copied to the cells in its neighborhood, which is determined by the scaling ratio (e.g., $40 \times 40 \ / \ (20 \times 20)$).

The pre-training procedure is necessary to explore cell resource and fit data manifold accurately. Since the data manifold here is described by a mixture of cell weights; when the network is initialized with too many cells, some initialized weights are possible to over-fit the data locally. As a result, the over-fitted cells can hardly be updated and only the rest cells are used to adapt to data distribution. The resource is wasted, and ultimately, the developed weights of these limited cells represent the data manifold in a very coarse manner.

IV. Experiments

Multiple experiments are conducted to evaluate the proposed sparse coding architecture, based on the data set of natural images, hand-written digits and 3-D object appearance.

A. Natural images

We apply the proposed sparse coding model to learn natural images [2] using only one layer with unsupervised learning ($\alpha = 0$). As natural images hold the vast inequities in variance along different directions of the input space, we should "sphere" the data by equalizing the variance in all directions [27]. This pre-processing is called whitening. The whitened sample vector $\mathbf{x}'$ is computed from the original sample $\mathbf{x}$ as $\mathbf{x}' = \mathbf{W}\mathbf{x}$, where $\mathbf{W} = \mathbf{V}\mathbf{D}$ is the whitening matrix. $\mathbf{V}$ is the matrix where each principal component $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$ is a column vector, and $\mathbf{D}$ is a diagonal matrix where the matrix element at row and column i is $\frac{1}{\sqrt{\lambda_i}}$ (λ_i is the eigenvalue of $\mathbf{v}_i$). Whitening is very beneficial to uncover the true correlations within the natural images since it avoids the derived features to be dominated by the larger components.

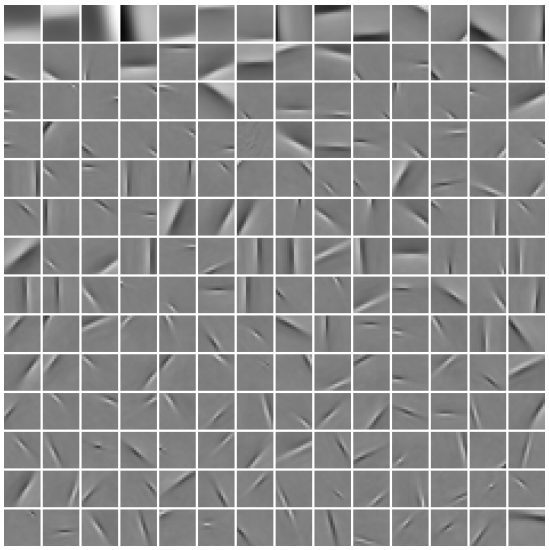
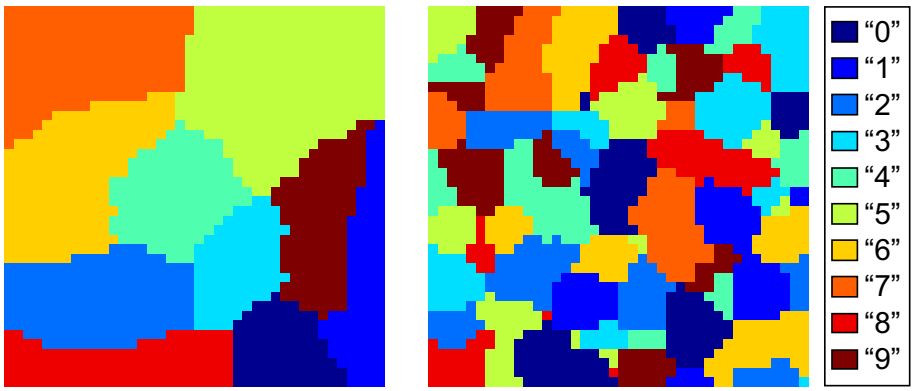


Fig. 5. A dictionary of feature weights developed from natural images (with whitening), ordered by the number of updating times of each cell. The cell with the most updates is at the top left of the image grid, and it progresses through each row until the one with the least updates, at the bottom right.

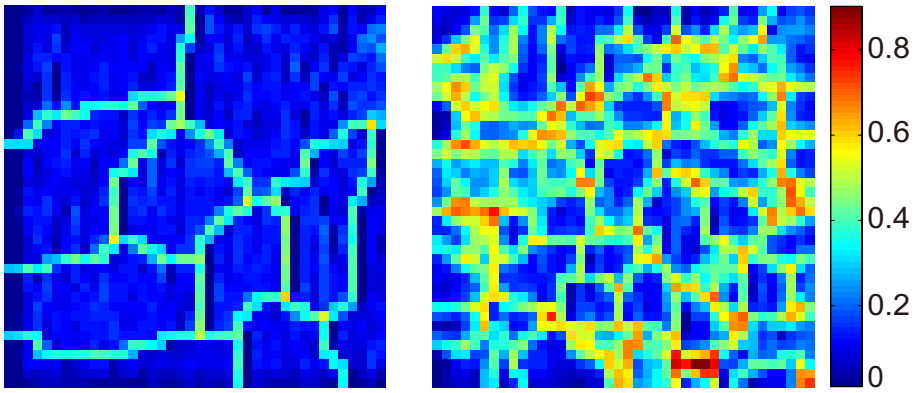
Figure 5 shows the developed learning weights using 256 cells and based on 500,000 whitened input samples with 16×16 dimensions. Each weight is reshaped to a 16×16 grid in the figure. The developed Gabor-like features resemble the

[2] available at http://www.cis.hut.fi/projects/ica/imageica/



(a) (b)

Fig. 6. 2D neural class map in the hidden layer (best viewed in color): (a) with top-down connections and (b) without top-down connections. Each cell is associated with one color, presenting a class with the largest empirical "probability" p_c.



(a) (b)

Fig. 7. Neural entropy in the hidden layer (best viewed in color): (a) with top-down connections and (b) without top-down connections.

orientation selective cells that were observed in V1 area [1], [28].

B. MNIST Handwritten Digits

Next, we will evaluate the discriminative tasks like object classification. MNIST is a well-known handwritten digit dataset [3] composed of 70,000 total images (60,000 training, 10,000 testing) with 10 classes of handwritten digits - from 0 to 9. Each image is size-normalized to $28 \times 28 = 784$ dimensions. All images have already been translation-normalized, so that each digit resides in the center of the image.

We developed a network with one hidden layer of 50×50 cells to train on this task, where $\alpha = 0.4$. To evaluate the discriminative sparse coding model compared to the unsupervised constructive one, we first define the empirical "probability" to evaluate a cell's updating experience across classes, as used in [23], [24]

$$p_c = \frac{n(c)}{\sum_{c=1}^{q} n(c)} \qquad c \in 1, 2, ..., q \tag{13}$$

where $n(c)$ is the updating times of a cell based on class c.

[3]available at http://yann.lecun.com/exdb/mnist/

We further use the "entropy" metric in [24] to measure the purity of a cell with respect to each class

$$\text{entropy} = -\sum_{c=1}^{q} p_c \log p_c. \tag{14}$$

where a cell with zero entropy learns inputs from the same class, while a cell with maximum entropy learns inputs with equal probability of all the classes.

To illustrate the maximum empirical "probability" of each cell, Fig. 6(a) plots a class map of the hidden layer with 50×50 cells. Given a cell's position, a color indicates a class holding the largest empirical "probability" p_c, and there are 10 colors in total. Based on the discriminative and pooling properties in Sec. II-C and Sec. II-D, cells tend to distribute along the classes (i.e., "relevant information"). When the number of available cells is larger than the number of classes, the cells representing the same class are grouped together, leading to the simpler boundaries for class decision and thus a better classification result as shown later. See [23], [24] for details and more examples. Without the discriminative top-down connections (i.e., $\alpha = 0$), however, no class-specific cell groups are observed and cells presenting the same class are scattered around the plane (see Fig. 6(b)).

Fig. 7(a) shows the corresponding entropy map in the hidden layer with 50×50 cells. Compared to the network without top-down discrimination (see Fig. 7(b)), the cell's entropy is much lower. In conclusion, the proposed network exhibits purer cell representations with respect to classes, and entails every cell with higher discriminative power for a specific class.

Table I summarizes the network performance with and without top-down discriminative connections, and compare to other state-of-art supervised models that deal with monolithic input. It shows that top-down discriminative connections boost the recognition performance in contrast to the network without top-down connections. Methods with local analysis and deformation processing (e.g., Convolutional Nets [29]) are known to be suited for the digit recognition problem with better performance. The proposed hierarchical structure compares favorably with the recognition rates achieved by K nearest neighbor (K-NN) and the back-propagation algorithms. The network performance is almost the same as the contrastive divergence approach applied to train restricted Bolzmann machines (RBM) [26]. The work of [26] further fine-tuned the RBM with deep structure using supervised gradient descent (called deep belief network) and reached the final performance with 1.25% error rate. Such a tuning is also applicable to our method and is investigated under the on-going studies.

TABLE I

SUMMARY OF RECOGNITION ERROR (%) ON MNIST DATASET. THE "IMPROVEMENT" ITEM SHOWS THE PERFORMANCE IMPROVEMENT DUE TO THE SUPERVISED TOP-DOWN CONNECTION.

K-NN (L2 Euclidean)	5.0
Back propagation (1000 hidden units)	4.5
Back propagation (500+150 hidden units)	2.95
Contrastive divergence	2.49
Deep belief network	1.17
Proposed work	2.64
Improvement	5.82

Using the trained network, we can also reconstruct an input providing the activation of one cell in the label layer, based on the random sampling of internal states. Fig. 8 shows certain examples of generated images for each class. This refers to the internal "imagination" of a neural network as is termed in [26].

C. NORB Objects

The normalized-centered NORB dataset[4] is a challenging dataset for 3D object recognition. It contains images of 50 different toy objects, and each 10 objects belong to one of five generic classes: cars, trucks, planes, animals, and humans. The training set contains 24,300 stereo image pairs of 25 objects (5 per class) while the test set contains remaining 24,300 stereo pairs with different 25 objects. We trained a network with two hidden layers, given both top-down-disabled and top-down-enabled configurations. There

[4]available at http://cs.nyu.edu/ ylclab/data/norb-v1.0/

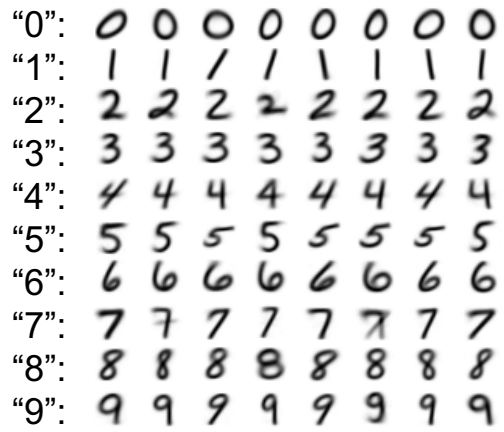


Fig. 8. Examples that are generated (i.e., "imagined") from the proposed model, with a particular label activated.

are 80×80 neurons used in the first layer and 40×40 in the second. $\alpha = 0.3$ in both layers.

TABLE II

SUMMARY OF RECOGNITION ERROR (%) ON NORB DATASET. THE "IMPROVEMENT" ITEM SHOWS THE PERFORMANCE IMPROVEMENT DUE TO THE SUPERVISED TOP-DOWN CONNECTION.

K-NN (L2 Euclidean)	18.4
Logistic Regression	19.6
SVM+Gaussian kernel	11.6
Deep belief network	11.9
Proposed work	12.1
Improvement	8.8

Tables II summarizes the performance with other well-known models. Our method compares favorably with other methods based on monolithic inputs. Better results are available by certain methods utilizing local analysis and supplementary training (e.g., aforementioned Convolutional Nets [29]). With top-down connections, our method outperforms K-nearest neighbor and provides a similar result regarding the deep belief network and SVM. However, SVM had to use significantly sub-sampled data (too slow to train with the original high dimensionality). And also, SVM lacks online learning capability and struggles when dealing with combinatorial data and expandable tasks. In that sense, our method is more scalable than any of the other methods, and new classes can be potentially added on the fly.

V. CONCLUSION

In this paper, we presented a hierarchical discriminative sparse coding via the propagation of bottom-up and top-down information flows. The sparse coding model minimizes the reconstruction errors of both bottom-up and top-down inputs. ℓ_0 norm is adopted for the sparse constraint, leading to an efficient online learning without iterative computation of the sparse representation. Due to this fact, the sparse coding model can be extended to learn deep structures in the divide-and-conquer fashion. The introduction of top-down connection and lateral pooling reorganizes cell distribution

for a class-specific grouping and facilitates the discriminative tasks. Experiments in visual recognition problems showed that the sparse coding network delivers similar performance with other comparable methods. Regarding the future work, we will extend the sparse coding algorithm to learn deeper networks with local analysis, and apply it in more complex problems with multiple sensory modalities.

VI. Acknowledgements

The authors would like to thank Matthew D. Luciw and John Weng for their helpful discussion and comments.

References

[1] B. A. Olshaushen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 13 1996.

[2] A. J. Bell and T. J. Sejnowski. The 'independent components' of natural scenes are edge filters. *Vision Research*, 37(23):3327–3338, 1997.

[3] B. A. Olshaushen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy used by V1? *Vision Research*, 37(23):3311–3325, 1997.

[4] W. E. Vinje and J. L. Gallant. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science*, 287(5456):1273–1276, 2000.

[5] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionsaries. *IEEE Trans. Signal Processing*, 41:3397–3415, 1993.

[6] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *The 27th Asilomar Conf. on Signals, Systems, and Computers*, 1993.

[7] S. Chen, D. Donoho, , and M. Saunders. Automatic decomposition by basis pursuit. *SIAM Journal of Scientific Computation*, 1(3):33–61, 1998.

[8] I. F. Gorodnitsky and B. D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *IEEE Transactions on Signal Processing*, 45(3):600–616, 1997.

[9] D. L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[10] E. Candes and J. Romberg. ℓ_1-magic : Recovery of sparse signals via convex programming. In *Technical Report California Institute of Technology*, Pasadena, California, 2005.

[11] K. Engan, S. O. Aase, and J. H. Husoy. Method of optimal directions for frame design. In *IEEE International Confrence on Acoustics, Speech, and Signal Processing*, 1999.

[12] M. S. Lewicki and T. J. Sejnowski. Learning overcomplete representations. *Neural Computation*, 12(2):337–365, 2000.

[13] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing*, 54(11):4311–4322, 2006.

[14] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *Advances in neural information processing systems*, pages 1137–1144, 2007.

[15] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Supervised dictionary learning. In *Advances in neural information processing systems*, pages 1033–1040, 2009.

[16] D. M. Bradley and J. A. Bagnell. Differentiable sparse coding. In *Advances in neural information processing systems*, pages 113–120, 2009.

[17] M. P. Young and S. Yamane. Sparse population coding of faces in the inferotemporal cortex. *Science*, 256:1327–1331, 1992.

[18] R. Baddeley, L. F. Abbott, M. C. Booth, F. Sengpiel, T. Freeman, E. A. Wakeman, and E. T. Rolls. Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proc. Biological Science*, 264:1775–1783, 1997.

[19] H. Lee, C. Ekanadham, and A. Y. Ng. Sparse deep belief net model for visual area v2. In *Advances in Neural Information Processing Systems*, pages 873–880, 2008.

[20] M. A. Ranzato, Y. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *Advances in Neural Information Processing Systems*, pages 1185–1192, 2007.

[21] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. *Parallel distributed processing: explorations in the microstructure of cognition*, 1:318–362, 1986.

[22] M. Rehn and F. T. Sommer. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of Computational Neuroscience*, 22(2):135–146, 2007.

[23] Z. Ji, M. Luciw, J. Weng, and S. Zeng. Incremental online object learning in a vehicular radar-vision fusion framework. *IEEE Transactions on Intelligent Transportation Systems*, PP(99):1–10, 2011.

[24] M. Luciw and J. Weng. Top-down connections in self-organizing hebbian networks: Topographic class grouping. *IEEE Trans. on Autonomous Mental Development*, 2(3):248–261, 2010.

[25] G. E. Hinton. Learning multiple layers of representation. *Trends in Cognitive Science*, 11(10):428–434, 2007.

[26] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[27] D. J. Field. Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America*, 4:2379–2394, 1987.

[28] D. H. Hubel and T. N. Wiesel. Receptive feilds of single neurons in the cat's striate cortex. *Journal of Physiology*, 148:574–591, 1959.

[29] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of IEEE*, 86(11):2278–2324, 1998.