Learning Image Representations from the Pixel Level via Hierarchical Sparse Coding

Kai Yu[†] Yuanqing Lin[†] John Lafferty[‡]

[†]NEC Laboratories America, Cupertino, CA 95014
[‡]Carnegie Mellon University, Pittsburgh, PA 15213

Abstract

We present a method for learning image representations using a two-layer sparse coding scheme at the pixel level. The first layer encodes local patches of an image. After pooling within local regions, the first layer codes are then passed to the second layer, which jointly encodes signals from the region. Unlike traditional sparse coding methods that encode local patches independently, this approach accounts for high-order dependency among patterns in a local image neighborhood. We develop algorithms for data encoding and codebook learning, and show in experiments that the method leads to more invariant and discriminative image representations. The algorithm gives excellent results for hand-written digit recognition on MNIST and object recognition on the Caltech101 benchmark. This marks the first time that such accuracies have been achieved using automatically learned features from the pixel level, rather than using hand-designed descriptors.

1. Introduction

Sparse coding refers to a general class of techniques that automatically select a sparse set of vectors from a large pool of possible bases to encode an input signal. While originally proposed as a possible computational model for the efficient coding of natural images in the visual cortex of mammals [17], sparse coding has been successfully applied to many machine learning and computer vision problems, including image super-resolution and image restoration. More recently, it has gained popularity among researchers working on image classification, due to its state-of-the-art performance on several image classification problems [18, 19, 5, 20, 15, 1].

Many image classification methods apply classifiers based on a Bag-of-Words (BoW) image representation [6], where vector-quantization (VQ) is applied to encode the pixels or descriptors of local image patches, after which the codes are linearly pooled within local regions. In this approach, prior to encoding, a codebook is learned via unsupervised k-means, which summarizes the distribution of signals by a set of "visual words." The method is very intuitive because the pooled VQ codes represent the image through the frequencies of these visual words.

Sparse coding can easily be plugged into the BoW framework as a replacement for vector quantization. Raina et al. [18] describe an approach that uses sparse coding to construct high-level features, showing that the resulting sparse representations perform much better than conventional representations, e.g., raw image patches. Yang et al. [20] propose a two stage approach where sparse coding model is applied over hand-crafted SIFT features, followed by a spatial pyramid max pooling. When applied to general image classification tasks, this approach has achieved state-of-the-art performance on several benchmarks when used with a simple linear classifier. However, this is achieved using sparse coding on top of hand-designed SIFT features. It is desirable to develop fully automatic methods to learn features from the pixel level.

A limitation of the above approaches is that they encode local patches independently, ignoring the spatial neighborhood structure of the image. In this paper we propose a two-layer sparse coding model to overcome this limitation, by modeling the higher-order dependency of patches in the same local region of an image. The first layer encodes individual patches, and the second layer then jointly encodes the set of patches that belong to the same group (i.e., image or image region). Accordingly, the model has two levels of codebooks, one for individual patches, and another for sets of patches. In the codebook learning phase, our model learns the two codebooks jointly, where each code in the higher-

level codebook represents a dependency pattern among the lower-level code words.

This approach offers several advantages in terms of both modeling and computation. Because the individual patches of the same group are jointly encoded, the first-layer codebook yields a more invariant representation compared with standard sparse coding. Moreover, the use of a higher-level codebook, whose codewords directly model the statistical dependency of the first layer codewords, allows the method to encode more complex visual patterns. Computationally, the encoding optimization is jointly convex over both layers. Finally, our method generates sparse representations at the image pixel level, which shows the feasibility of learning features fully automatically.

We evaluate the new hierarchical sparse coding algorithm on the well-known MNIST digit recognition benchmark and the Caltech-101 object recognition benchmark. Our results show that the unsupervised two-layer coding scheme generates image representations that are more invariant and discriminative than those obtained through one-layer coding, leading to improved accuracies for both image classification tasks.

The remainder of the paper is organized as follows. In Section 2, we introduce the two-layer coding scheme, and describe the optimization procedure for data encoding. We then describe the codebook learning algorithm in Section 3, followed by a description of a classification method that uses the new coding scheme in Section 4. The experimental results are presented in Section 5, and concluding remarks are made in Section 6.

2. Hierarchical Sparse Coding

Let $x_1, \ldots, x_n \in \mathbb{R}^d$ be a set of n patches within an image. For now we ignore the spatial information of the patches; as we show later, it is straightforward to incorporate a dependence on location. Our goal is to obtain a sparse representation for this set of patches. Let $X = [x_1\, x_2 \cdots x_n] \in \mathbb{R}^{d\times n}$ be the set of patches in matrix form. Let $B \in \mathbb{R}^{d\times p}$ be a dictionary of codewords for the first level, which we also call the patch-level, as in standard sparse coding. In addition, we introduce a second level or set-level dictionary $\Phi = (\phi_1\phi_2\cdots\phi_q) \in \mathbb{R}_+^{p\times q}$, where each element of Φ is non-negative. The set-level codebook Φ will be used to model the statistical dependencies among the representations of the patches x_i in the patch-level.

We obtain sparse representations simultaneously at the patch-level and the set-level by carrying out the following optimization:

$$(\widehat{W}, \widehat{\alpha}) = \arg\min_{W,\alpha} \quad L(W,\alpha) + \frac{\lambda_1}{n}\|W\|_1 + \gamma\|\alpha\|_1 \quad (1)$$
$$\text{subject to} \quad \alpha \succeq 0,$$

where the loss function $L(W,\alpha)$ is given by

$$\frac{1}{n}\sum_{i=1}^{n}\left\{\frac{1}{2}\left\|x_i - Bw_i\right\|^2 + \lambda_2 w_i^\top \Omega(\alpha) w_i\right\}.$$

Here $W = (w_1\, w_2 \cdots w_n) \in \mathbb{R}^{p\times n}$ is the patch-level representation, $\alpha \in \mathbb{R}^q$ is the set-level representation, and

$$\Omega(\alpha) \equiv \left(\sum_{k=1}^{q}\alpha_k \mathrm{diag}(\phi_k)\right)^{-1}.$$

The ℓ_1 penalty on each w_i and on α encourages sparsity in the representations at both levels.

To gain some intuition for what this optimization is doing, first note that taking $\lambda_2 = 0$ reduces the procedure to standard sparse coding, which encodes each patch independently. On the other hand, if $\lambda_2 > 0$, then the term involving α implements a type of weighted ℓ_2 regularization of w_i. Note, however, that pooling these terms together results in an expression of the form

$$\frac{1}{n}\sum_{i=1}^{n} w_i^\top \left(\sum_{k=1}^{q}\alpha_k\ \mathrm{diag}(\phi_k)\right)^{-1} w_i = \mathrm{tr}\Big(S(W)\,\Omega(\alpha)\Big)$$

where

$$S(W) \equiv \frac{1}{n}\sum_{i=1}^{n} w_i w_i^T \in \mathbb{R}^{p\times p}$$

is the sample covariance of the patch-level representations. Thus, the loss function $L(W,\alpha)$ may be written more succinctly as

$$L(W,\alpha) = \frac{1}{2n}\|X - BW\|_F^2 + \frac{\lambda_2}{n}\,\mathrm{tr}\Big(S(W)\,\Omega(\alpha)\Big) \quad (2)$$

If the w_i vectors were sampled independently from a Gaussian with covariance matrix $\Sigma(\alpha) = \Omega(\alpha)^{-1}$, the log-likelihood of W would be $\mathrm{tr}\Big(S(W)\,\Omega(\alpha)\Big)$, plus a constant that doesn't depend on W. Thus, the set-level code can be seen to model the covariance structure of the patch-level representations.

Note that hierarchical sparse coding, as defined above, is similar to but fundamentally different from the group sparse coding procedure recently proposed by Bengio et al. [1]. The method in [1] incorporates a group lasso penalty $\|W\|_2$ to encourage similar sparsity patterns for the patches in a group. However, there

is no second codebook that is constructed at a higher level. As will be seen in our experimental results, the set-level codebook that results in a hierarchical coding scheme that is interpretable, where the set-level codebook is effectively a shift-invariant representation of correlated patch-level bases.

Importantly, the encoding optimization problem above is jointly convex in both W and α. To see this, recall that the matrix-fractional function $f(x, Y) = x^T Y^{-1} x$ is jointly convex as a function of the vector x and the positive-semidefinite matrix Y (see [4]), and $\sum_{k=1}^{q} \alpha_k \text{diag}(\phi_k)$ is affine in α.

It is convenient to use an alternating optimization procedure to actually compute the solution, by iteratively optimizing W with α fixed, and then optimizing α with W fixed. The details of these optimizations are described next.

2.1. Optimization of patch-level representation W

The optimization of W for fixed α can be seen as a modified elastic net problem, using a weighted ℓ_2 norm regularization. Specifically, the optimization

$$\min_{W} \frac{1}{n} \sum_{i=1}^{n} \Big\{ \frac{1}{2} \|x_i - B w_i\|_2^2 + \lambda_1 \|w_i\|_1 + \lambda_2 w_i^\top \Omega(\alpha) w_i \Big\} \tag{3}$$

is a generalized elastic net problem. It can be transformed into a canonical lasso problem as

$$\min_{W} \frac{1}{n} \sum_{i=1}^{n} \Big\{ \frac{1}{2} \Big\| \widetilde{x}_i - \widetilde{B} w_i \Big\|_2^2 + \lambda_1 \|w_i\|_1 \Big\}$$

where

$$\widetilde{x}_i = \begin{bmatrix} x_i \\ 0_{p \times 1} \end{bmatrix}, \quad \widetilde{B} = \begin{bmatrix} B \\ \left(\lambda_2 \sum_{k=1}^{q} \alpha_k \, \text{diag}(\phi_k)\right)^{-\frac{1}{2}} \end{bmatrix}$$

and $0_{p \times 1}$ denotes a vector of p zeros. Fast algorithms based on iterative soft thresholding are available for efficiently solving this quadratic program.

2.2. Optimization of set-level representation α

The optimization problem for updating α with W fixed is

$$\min_{\alpha \succeq 0} \quad \frac{1}{n} \sum_{i=1}^{n} \left\{ \lambda_2 w_i^\top \left(\sum_{k=1}^{q} \alpha_k \text{diag}(\phi_k) \right)^{-1} w_i \right\} + \gamma \|\alpha\|_1, \tag{4}$$

Again, we transform the optimization problem in order to take the advantage of well-developed lasso solvers, as

$$\min_{\alpha \succeq 0, \Sigma \succeq 0} \quad \frac{\lambda_2}{n} \sum_{i=1}^{n} w_i^\top \Sigma^{-1} w_i + \lambda_3 \left[\|\sigma - \Phi \alpha\|_2^2 + \lambda_4 \|\alpha\|_1 \right], \tag{5}$$

where $\text{diag}(\Sigma) = \sigma$ and $\lambda_4 = \gamma / \lambda_3$. This optimization is jointly convex with respect to Σ and α. As $\lambda_3 \to \infty$, this formulation formulation is equivalent to the original one. In our implementation we set λ_3 to a very large number.

Here again, we adopt an alternating minimization procedure, which alternates between the updates of σ and α. For fixed α, the optimization for each element of σ can be done independently, and the resulting one-dimensional problems can be efficiently solved. On the other hand, the optimization for α is a standard nonnegative lasso problem, which can also be efficiently solved.

3. Codebook Learning

Effective image coding requires high-quality codebooks B and Φ. Now we describe algorithms to learn the codebooks to capture the structural information of data.

Let $\mathcal{X} = (X_1, \ldots, X_m)$ be m image patch sets, obtained from local regions of training images. The formulation of codebook learning aims at solving the following optimization problem.

$$\begin{aligned} \min_{B, \Phi} & \left\{ \frac{1}{m} \sum_{j=1}^{m} \min_{W^j, \alpha^j} L(W^j, \alpha^j, \sigma^j, B, \Phi) \right\} \\ \text{subject to} \quad & \|B_i\| \le 1, \ \|\phi_k\|_1 \le 1, \\ & i = 1, 2, ..., p, \ k = 1, 2, ..., q \\ & \sigma^j \succeq 0, \Phi \succeq 0 \end{aligned} \tag{6}$$

where

$$\begin{aligned} & L(W^j, \alpha^j, \sigma^j, B, \Phi) \\ & = \sum_{i=1}^{n} \left[\frac{1}{2n} \|x_i^j - B w_i^j\|^2 + \frac{\lambda_1}{n} \|w_i^j\|_1 + \frac{\lambda_2}{n} (w_i^j)^\top \Sigma_j^{-1} w_i^j \right] \\ & \quad + \lambda_3 \left[\|\sigma_j - \Phi \alpha_j\|_2^2 + \lambda_4 \|\alpha\|_1 \right] \end{aligned}$$

where Σ_j is a diagonal matrix and $\text{diag}(\Sigma_j) = \sigma_j$. It is easy to see that the above objective function is the same as the one in the coding phase if the codebooks are given. One important feature of the above formulation is that the set-level dictionary Φ is required to be nonnegative.

The optimization problem can be solved by iteratively alternating the following two steps: 1) given the

codebooks B and Φ, compute the optimal coding using the methods described in Section 2; 2) given the new coding, re-optimize the codebooks. For the coding step (Step 1), we have described the algorithms in Section 2. For the Step 2), B and Φ can be optimized independently.

For solving B, the optimization problems can be solved via their dual formulation [12], which becomes a convex optimization with solely nonnegative constraints. We developed a projected Newton method for efficiently solving the resulting optimization. The projected Newton method can be shown to have superlinear convergence rate under fairly mild conditions [3].

Optimizing Φ is a bit more tricky due to the extra nonnegativity constraint on its elements. Fortunately, the optimization is still convex. We developed a projected gradient algorithm for solving the optimization problem [3]. For the projected gradient, each iteration step consists of two sub-steps. First, each column of ϕ_k goes one step along the gradient direction

$$(\phi_k)_{1/2} = \phi_k - \eta \nabla_{\phi_k} \tag{7}$$

where ∇_{ϕ_k} is the gradient of ϕ_k, and η is a stepsize that needs to be determined by line search. Then the projection step is to find the point in the constrained domain that is closest to $(\phi_k)_{1/2}$. The projection can be done by independently solving the following optimization problem on each column of Φ:

$$\min_{\phi_k} \|\phi_k - (\phi_k)_{1/2}\|^2 \tag{8}$$

$$\text{subject to} \quad \sum_{l=1}^{p} \phi_{lk} = 1, \ \phi_{lk} \geq 0 \tag{9}$$

where ϕ_{lk} is the l^{th} element of ϕ_k. This optimization is to project $(\phi_k)_{1/2}$ onto a probabilistic simplex, and it can be solved very efficiently, i.e. in $\mathcal{O}(p)$ as described in [7].

4. Application to Image Classification

The proposed hierarchical sparse coding is readily applicable to learning image representations for classification. As revealed by the data encoding procedure in Section 2.1 and 2.2, the whole model operates on a set X of image patches in a local region, first nonlinearly mapping each x from the region to its sparse code w, and then (implicitly) pooling the codes of the set to obtain Σ, which is akin to the sample (diagonal) covariance of the sparse codes in that region, and corresponds to a way of "energy pooling". In the next level, the model encodes Σ nonlinearly to obtain the sparse code α for the set X. The encoding procedure is implemented by solving a joint convex optimization problem (1), which is also visualized by Figure 1.

4.1. Modeling Spatial Dependence

A slight modification can lead to a more general formulation, in the sense that Σ acts as not the sample covariance for only one region, but for several neighboring regions jointly. Then the learned bases Φ will capture the spatial dependence among several regions. Without loss of generality, let's consider a joint model for 2×2 local regions. Suppose each region contains n patches, let X and W denote all the $4 \times n$ patches and their first-layer codes in these 4 regions. Then $L(W, \alpha)$ in (2) is modified as

$$\frac{1}{n}\|X - BW\|_F^2 + \frac{\lambda_2}{n} \sum_{s,t} \text{tr}\Big(S(W^{(s,t)})\, \Omega^{(s,t)}(\alpha)\Big)$$

where

$$\Omega^{(s,t)}(\alpha) \equiv \left(\sum_{k=1}^{q} \alpha_k \text{diag}\left(\phi_k^{(s,t)}\right)\right)^{-1} \tag{10}$$

is the inverse diagonal covariance for the (s,t)-th region, $s = 1, 2$, $t = 1, 2$. In this model, each local descriptor has its own first-level coding, while the 2×2 regions share the joint second-layer coding α. Each basis $\phi_k = [\phi_k^{(1,1)}, \phi_k^{(1,2)}, \phi_k^{(2,1)}, \phi_k^{(2,2)}] \in \mathbb{R}^{p \times 4}$ describes a spatial co-occurrence pattern across 2×2 regions.

4.2. Hierarchical Convolution Coding

A further improvement is to convolve the above joint model over the image. Again, without loss of generality, let the image be partitioned into 4×4 regions, indexed by (s,t). Then convolution of the two-layer hierarchical coding over every 2×2 region neighborhood leads to 3×3 coding results $\alpha^{(u \times v)} \in \mathbb{R}^q$, where $u = 1, 2, 3$ and $v = 1, 2, 3$. Here each (u,v) indexes a "receptive field" of the hierarchical coding. Let X and W denote all the patches and their first-layer codes in the image. Then $L(W, \alpha)$ in (2) is modified as

$$\frac{1}{n}\|X - BW\|_F^2 + \frac{\lambda_2}{n} \sum_{s,t} \sum_{u,v} \varphi\left(W^{(s,t)}, \alpha^{(u,v)}\right) \tag{11}$$

where $\varphi(W^{(s,t)}, \alpha^{(u,v)}$ is defined to be zero if the (s,t)-region is not in the (u,v) receptive field, otherwise

$$\varphi\left(W^{(s,t)}, \alpha^{(u,v)}\right) = \text{tr}\Big(S(W^{(s,t)})\, \Omega^{(s,t)}(\alpha^{(u,v)})\Big)$$

where

$$\Omega^{(s,t)}\left(\alpha^{(u,v)}\right) \equiv \left(\sum_{k=1}^{q} \alpha_k^{(u,v)} \text{diag}\left(\phi_k^{r(s,t,u,v)}\right)\right)^{-1}. \tag{12}$$

Here $r(s,t,u,v)$ indexes the relative position of the (s,t) region in the (u,v) receptive field. The coding algorithm and codebook learning algorithm are basically the same as those described in the previous section.

4.3. Image Representation

We follow the standard procedure to sample image patches densely at a grid of locations. We partition the patches into different non-overlapping regions based on their spatial locations, and then treat each window of several regions as a receptive field. For example, a typical setting can be

- Each patch is 4×4 pixels, sampled from a grid with step size 2 pixels;
- Each non-overlapping region contains 4×4 patches;
- Each receptive field contains 4×4 such non-overlapping regions, with a step size 1.

Finally each receptive field will give rise to a q-dimensional second-layer code vector. We pool the second-layer code vectors by using max pooling. In order to obtain better shift and scale invariance, we partition each image in different scales, for example, into 1×1 and 2×2 blocks, and pool the second-layer codes within each block. In the end, we concatenate the block-wise results to form the image representation.

Finally, we note that it is straightforward to extend the current two-layer model to multi-layer ones, because it it naturally extends to a general hierarchical approach by having a codebook at each level, and including a covariance operator that regularizes the codebook coefficients of the appropriate groups of vectors in the lower level. This suggests a connection to "deep learning" [8]. Recently learning feature hierarchies from unlabeled data has become an active research area [8, 16, 2, 13]. Most of the works are based on restrictive Boltzmann machines or autoencoder neural networks. Our work provides one way to potentially learn a stack of sparse coding models.

4.4. Connection to Sparse Coding on SIFT

The architecture of two-layer convolution coding has an interesting analogy to sparse coding on a SIFT feature vector [14]. For each SIFT descriptor, its receptive field contains 4×4 smaller non-overlapping regions – within each region, responses of a 8-dimensional coding, corresponding to a histogram of 8 orientations, are pooled together. A SIFT descriptor is then resulted from concatenating the 4×4 pooling results, outputting a 128 dimensional vector. Then sparse coding is the second-layer coding applied on top of SIFT. It has been shown in [20] that sparse coding on SIFT leads to state-of-the-art results on a number of image classification benchmarks. The method presented here follows a similar processing architecture, but is a fully automatic approach learning features from raw pixels.

5. Experiments

5.1. MNIST dataset

Our first experiment is based on the MNIST handwritten digit recognition benchmark, where there are 70000 data examples, and each is a 28×28 gray image. All the images are pre-normalized into a unitary 784-dimensional vector. The data set is divided into a training set with 60000 images and a test set with 10000 images.

For each digit image, we densely sample 12×12 patches at every location (with padding), and partition the image in two different scales, i.e., 1×1 and 2×2. Therefore the second-layer coding is independently performed for each of the 5 regions, and the coding results are concatenated to form the image representation. The codebook sizes are 1024 for B and 2048 for Φ. We train the codebooks based on the unlabeled training set. We initialize the first-layer codebook by learning the first-layer codebook without the second layer coding, and then start the iterative procedure to learn the both codebooks. The regularization parameters are chosen via evaluating the classification performance on a small holdout set of the training data.

We first visualize the jointly learned codebooks in Fig. 2, where each row corresponds a random second-layer basis in Φ, and in each row we show the top associated first-layer basis in B. The figure shows that each basis in Φ captures the dependency of related B patterns at different locations. This implies that though the first layer bases are less invariant, the second-layer coding can be more shift-invariant.

We also find that the sparse codes based on hierarchical coding are more discriminative for classification. In this investigation, we compare our method with the popular one-layer sparse coding approach that first encodes local patches and then obtains image representations via square-root of average energy pooling (we also tried max-pooling, which produced similar results on MNIST). In order to make the comparison possible, for hierarchical coding we extract the feature representation by pooling its first-layer codes in the same way. For the two compared image representations, we compute dimension by dimension separately the *fisher discriminant score*, which is the ratio of within-class variance over between-class variance, as used by linear discriminant analysis (LDA). Let $F_1(d)$ be the score

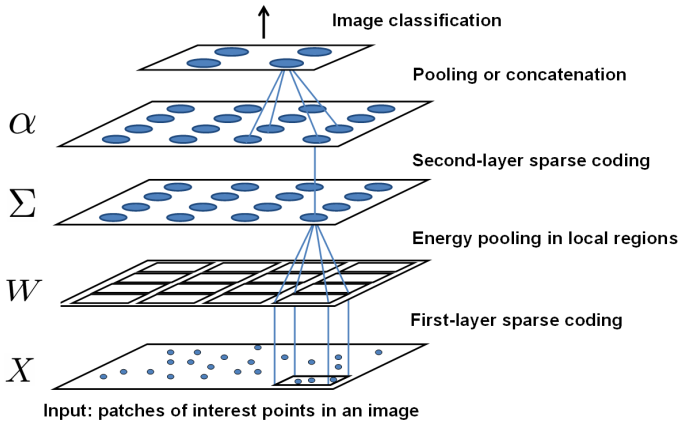

Figure 1. Illustration of hierarchical sparse coding applied to image classification: the figure visualizes the case where we partition the image into 4×4 regions in one scale, and apply the hierarchical coding on the 16 patch groups.

for the d-th dimension of the representation obtained by the one-layer sparse coding, and $F_2(d)$ be the d-th dimension of the representation obtained by our two-layer sparse coding. Then we compute the ratio $R(d) = F_1(d)/F_2(d)$ — if $R(d) > 1$, it implies that two-layer sparse coding produces a more discriminative representation on the d-th dimension. Fig. 3 shows the distribution of this ratio values. As we can see, the majority of the ratios are greater than one, indicating that the two-layer coding is more discriminative than the one-layer coding.

In the next we apply the learned representation for image classification using linear SVMs. The results are shown in Table 1, which compared the result with those of competitors. including unsupervised and supervised sparse coding methods, with and without convolution, and also convolution neural networks which actually learn image representations in a supervised way. We note that convolution neural network is the state-of-the-art method for hand-written digit recognition. Our method outperforms other competitors in terms of classification accuracy, given the fact that the presentation is obtained from a purely unsupervised learning approach.

5.2. Caltech-101 Object Recognition

The Caltech-101 dataset contains 9144 images of 101 object categories, including animals, vehicles, flowers, etc., plus one background category. The number of images per category varies from 31 to 800. Most images are medium resolution , i.e. about 300×300 pixels. We follow the common experiment setup for Caltech-101, to repeat 10 times, each time training on 30 random images per category and testing on the rest. Average classification accuracy normalized by class frequency is used for evaluation. We treat all the images as gray

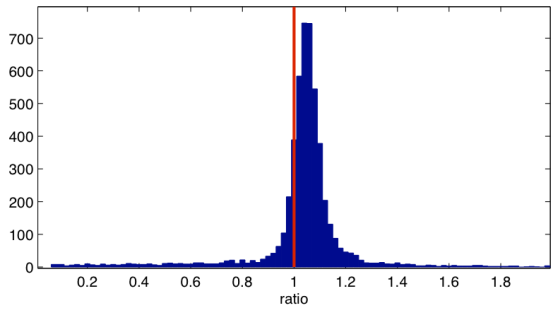

Figure 3. Illustration of discrimination power of hierarchical coding vs. one-layer coding for MNIST digit images,

images

We apply the hierarchical convolution sparse coding here, and implement two architectures. In the first architecture, we extract every 4×4 patches with sampling step size being 1 pixel, and let each region contain $4 \times 4 = 16$ such patches. The first-layer dictionary is set to be 8; for the second-layer, the joint coding is performed on every receptive field containing $4 \times 4 = 16$ regions, with the dictionary size being 2048, which means the coding is for a $16 \times 8 = 128$-dimensional vector space; In the second architecture, we increase the complexity of the model by letting the patch size be 8×8, the first-layer codebook size be 64, and the second-layer codebook size be 4096. In the end, we pool the second-layer codes by following the spatial pyramid structure 1×1, 2×2, and 4×4, which has been commonly used for this particular benchmark. We visualize the learned first-layer bases of the second architecture in Figure 4. It is non-trivial in this case to visualize the second-layer bases.

The object recognition results are shown in Table 2, with a comparison to those reported in literature by

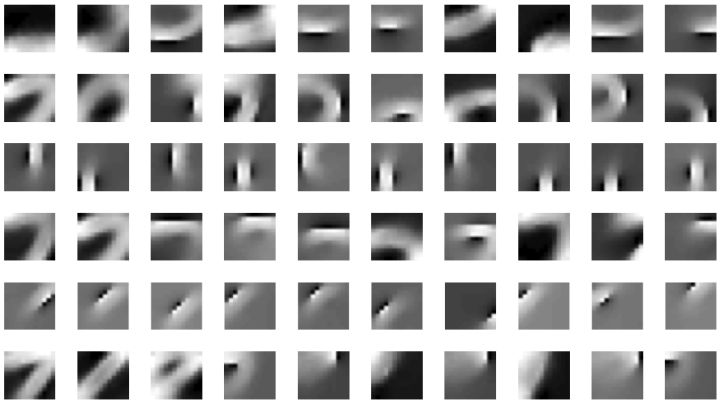

Figure 2. Illustration of bases learned from MNIST digit images, where each row corresponds a random second-layer basis in Φ, and in each row we show the top associated first-layer basis in B.

Methods	Error rate (%)
Sparse coding (unsupervised)	2.10
Local coordinate coding (unsupervised) [21]	1.90
Extended local coordinate coding (unsupervised) [21]	1.64
Differentiable sparse coding (supervised) [5]	1.30
Discriminative sparse coding (supervised) [15]	1.05
One-layer sparse coding (unsupervised)	0.98
Convolutional neural network (supervised) [11]	0.82
Hierarchical sparse coding (unsupervised)	**0.77**

Table 1. Classification error rate on MNIST by different sparse coding approaches, all operating on pixels.

using various unsupervised feature learning methods. These methods can be roughly put into two categories: one is feature learning on top of hand-crafted SIFT features, like [10, 20]; the other is feature learning directly on image pixels, e.g. [18, 13]. We note that feature learning from pixels has been known as a challenging problem. Even though researchers in machine learning have been pursuing hard to develop methods to automatically learn better features, as shown in Table 2, systems using SIFT features still outperform fully automatic methods by a big margin. But our results show that, at least on this well-known benchmark, hierarchical sparse coding can achieve very competitive results, which is encouraging. Between the two architectures we implemented, the one with higher complexity performs significantly better, achieving 74% accuracy. To the best of our knowledge, this is one of the best results without using feature combination or multi-kernel learning.

6. Conclusion

In this paper we introduce a new method to learn image representations via a two-layer sparse coding network at the pixel level. The first layer encodes local patches while the second layer forms a joint representation that effectively models the covariance of the patches from the neighboring image region. Unlike traditional one-layer sparse coding methods, this new approach learns higher-order dependencies among related image patterns, which leads to sparse yet more invariant and discriminative image representations. We develop algorithms for two-layer encoding and learning of two-layer codebooks from unlabeled data. Our experiments show that hierarchical sparse coding produces image representations that improve accuracy on

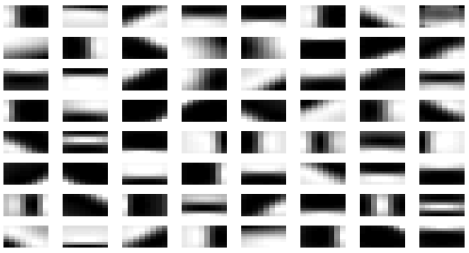

Figure 4. Illustration of the learned bases from Caltech images, the 64 first-layer bases of the second architecture.

Methods	Accuracy (%)
VQ coding on SIFT (nonlinear SVM) [10]	64.4
Sparse coding on SIFT [20]	73.2
One-layer sparse coding on pixels [18]	46.6
One-layer convolution deep belief network on pixels [13]	60.5
Two-layer convolution deep belief network on pixels [13]	65.4
Two-layer convolutional neural network on pixels [9]	66.3
Hierarchical sparse coding on pixels - architecture I	**70.8**
Hierarchical sparse coding on pixels - architecture II	**74.0**

Table 2. Normalized classification accuracy on Caltech101 object recognition benchmark by different single-layer and multiple-layer unsupervised feature learning approaches, in the setting of 30 training examples per class.

both the MNIST digit recognition problem and the Caltech101 object recognition benchmark. The results show that automatically learning features from image pixels is a promising research direction.

References

[1] S. Bengio, F. Pereira, Y. Singer, and D. Strelow. Group sparse coding. In *NIPS*, pages 82–89. 2009. 1713, 1714

[2] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *NIPS*, 2007. 1717

[3] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 2003. 1716

[4] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 1715

[5] D. M. Bradley and J. A. Bagnell. Differential sparse coding. In *NIPS*, 2008. 1713, 1719

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004. 1713

[7] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l_1-ball for learning in high dimensions. In *ICML*, 2008. 1716

[8] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 1717

[9] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. LeCun. Learning convolutional feature hierachies for visual recognition. In *NIPS*, 2010. 1720

[10] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1719, 1720

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1719

[12] H. Lee, A. Battle, R. Raina, and A. Y. Ng. Efficient sparse coding algorithms. In *NIPS*, 2006. 1716

[13] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *ICML*, 2009. 1717, 1719, 1720

[14] D. G. Lowe. Distinctive image features from scale invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 1717

[15] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Supervised dictionary learning. *NIPS*, 21, 2008. 1713, 1719

[16] F. MarcAurelio Ranzato, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies with applications to object recognition. In *CVPR*, 2007. 1717

[17] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381, 1996. 1713

[18] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng. Self-taught learning: Transfer learning from unlabeled data. In *ICML*. 1713, 1719, 1720

[19] M. Ranzato, Y.-L. Boureau, and Y. LeCun. Sparse feature learning for deep belief networks. In *NIPS*, 2007. 1713

[20] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 1713, 1717, 1719, 1720

[21] K. Yu and T. Zhang. Improved local coordinate coding using local tangents. In *ICML*, 2010. 1719