Efficient Learning of Sparse, Distributed, Convolutional Feature Representations for Object Recognition

Kihyuk Sohn Dae Yon Jung Honglak Lee Alfred O. Hero III
Dept. of Electrical Engineering and Computer Science
University of Michigan, Ann Arbor, MI 48109
{kihyuks,dyjung,honglak,hero}@umich.edu

Abstract

Informative image representations are important in achieving state-of-the-art performance in object recognition tasks. Among feature learning algorithms that are used to develop image representations, restricted Boltzmann machines (RBMs) have good expressive power and build effective representations. However, the difficulty of training RBMs has been a barrier to their wide use. To address this difficulty, we show the connections between mixture models and RBMs and present an efficient training method for RBMs that utilize these connections. To the best of our knowledge, this is the first work showing that RBMs can be trained with almost no hyperparameter tuning to provide classification performance similar to or significantly better than mixture models (e.g., Gaussian mixture models). Along with this efficient training, we evaluate the importance of convolutional training that can capture a larger spatial context with less redundancy, as compared to non-convolutional training. Overall, our method achieves state-of-the-art performance on both Caltech 101 / 256 datasets using a single type of feature.

1. Introduction

Object recognition poses a significant challenge due to the high pixel-level variability of objects in images. Therefore, having higher-level, informative image features is a necessary component for achieving state-of-the-art performance in object classification and detection. In the last decades, many efforts have been made to develop feature representations that can provide useful low-level information from images (e.g., [1, 2]). However, these feature representations are often hand-designed and require significant amounts of domain knowledge and human labor.

Therefore, there has been much interest in developing unsupervised and supervised feature learning algorithms for image representations that address these difficulties. Notable successes include clustering [3, 4, 5, 6], sparse coding [7, 8, 9], and deep learning methods [10, 11, 12, 13]. These methods are nonlinear encoding algorithms that provide new image representations from inputs. For instance, unsupervised learning algorithms (e.g., sparse coding [14]) can learn representations for low-level descriptors (e.g., SIFT) and provide discriminative features for visual recognition [9]. From another perspective, these methods can be viewed as generative models with latent variables that learn salient structures and patterns from inputs. In this view, the posterior probabilities of the latent variables can be used as features for discriminative tasks.

Although recently developed models provide powerful feature representations for visual recognition, some of these models are difficult to train, which has been a barrier to their wide use in many applications. For example, while the restricted Boltzmann machine [15] has rich expressive power and capability to build a deep network, it is difficult to train due to its intractable partition function and the need to tune many hyperparameters through expensive cross-validation.

In this paper, we investigate *black-box* training of restricted Boltzmann machines. Our main idea is to examine theoretical links among the unsupervised learning algorithms and take advantage of simple models to train more complicated models. We first provide a theoretical analysis showing the equivalence between Gaussian mixture models (GMMs) and Gaussian restricted Boltzmann machines [16] under specific constraints. This link has far-reaching implications on existing algorithms. For example, sparse RBMs [17] can be viewed as an approximation to a relaxation of clustering algorithms, and thus can provide richer image representations than clustering methods. Using these equivalence and implications, we enhance RBM training by utilizing K-means as a way of initializing RBMs. This allows for faster training and greater classification performance. We evaluate clustering methods and sparse RBMs on standard computer vision benchmarks, showing that sparse RBMs outperform clustering algorithms by allowing *distributed and less sparse* encoding.

Furthermore, we provide a simple connection between convolutional RBMs and non-convolutional RBMs. For ex-

ample, the convolutional RBM becomes equivalent to its non-convolutional counterpart when the convolution filter size is 1 (i.e., no spatial context). Not surprisingly, the convolutional RBM thus can capture larger spatial contexts and reduce the redundancy of feature representations.

Based on our efficient training method, we systematically evaluate the performance of convolutional RBMs on standard object recognition benchmarks, such as Caltech 101 and Caltech 256. We also provide an analysis of hyperparameters, such as sparsity and convolutional filter size, to demonstrate the effectiveness of sparse, distributed, convolutional feature learning. Overall, our approach leads to enhanced feature representations that outperform other learning-based encoding methods (e.g., ScSPM [9] and LLC [18]) and achieve state-of-the-art performance.

The main contributions of this paper are as follows:

- We provide a theoretical analysis showing an equivalence between mixture models and Gaussian restricted Boltzmann machines with specific constraints. We further show that sparse RBMs can be viewed as an approximation to a relaxation of such mixture models.
- Using these connections, we propose an efficient training method for sparse RBMs and convolutional RBMs. To the best of our knowledge, this is the first work showing that RBMs can be trained with almost no hyperparameter tuning to provide classification performance similar to or significantly better than GMMs.
- We evaluate the importance of convolutional training that can capture larger spatial contexts with less redundancy (compared to non-convolutional training). Specifically, we learn a feature representation based on SIFT and convolutional restricted Boltzmann machines. In the experiments, we show that such convolutional training provides a much better representation than its non-convolutional counterparts.
- Overall, our method achieves state-of-the-art performance on both Caltech 101 / 256 datasets using a single type of feature.

The rest of the paper is organized as follows. In Section 2, we revisit related works on unsupervised feature learning and convolutional learning. Section 3 describes preliminaries to the model, and subsequently in Section 4, we show the connection between Gaussian mixture models and softmax constrained Gaussian RBMs, as well as other connections regarding sparse RBMs and convolutional RBMs. We then introduce our proposed training method based on these connections. In Section 5, we evaluate our method on object recognition benchmark datasets. Finally, Section 6 presents conclusions of the paper.

2. Related work

Recently, researchers have tried to improve image features for object classification via unsupervised learning. A common unsupervised feature learning framework for classification is as follows: (1) densely extract image descriptors (e.g., SIFT [1] or HOG [2]); (2) train a feature mapping using an unsupervised learning algorithm; (3) once the feature mapping is learned, encode the descriptors to obtain "mid-level features" [7, 9]; (4) pool the single vector from multi-scaled sub-regions (e.g., spatial pyramid matching [5]) that characterize the entire image. The representations are then provided as inputs for linear or nonlinear classifiers (e.g., support vector machines). This pipeline is shown as a diagram in Figure 1.

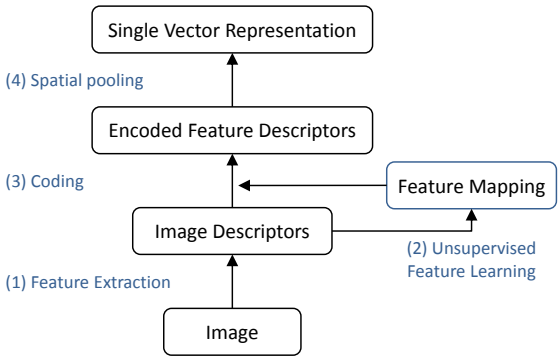

Figure 1. Pipeline for constructing features in object recognition.

Indeed, advanced encoding algorithms for image descriptors can provide significant improvements in object recognition. For example, ScSPM [9] applied patch-based (non-convolutional) sparse coding on densely extracted SIFT descriptors to obtain sparse feature representations. Similarly, Wang et al. [18] proposed Locality-constrained Linear Coding (LLC) based on locality. Boureau et al. [7] also used sparse coding, but they considered macrofeatures, which encode neighboring low-level descriptors to incorporate the spatial information. While these models are *not* trained convolutionally, we use the convolutional RBM [19] as an unsupervised learning algorithm on top of the SIFT descriptors. Our feature representation is robust to translation variations of images and effectively captures the larger spatial context, as shown in the experiments.

Convolutional extensions of unsupervised learning algorithms, such as sparse coding and RBMs, have been successful in developing powerful image representations. For example, Zeiler et al. [20] and Kavukcuoglu et al. [13] developed algorithms for convolutional sparse coding, which approximately solves the L_1-regularized optimization problem to minimize the reconstruction error between the data and the higher layer features convolved with the filters. Our approach is different from these methods in that we used convolutional RBMs instead of convolutional sparse coding. Further, we verified the advantage of convolutional training through the experimental comparison between convolutional and non-convolutional training.

Compared to sparse coding, RBMs can compute poste-

rior probabilities in a feedforward way, which is usually orders of magnitude faster. This computational efficiency provides a significant advantage over sparse coding since it scales up to a much larger number of codes. Furthermore, convolutional RBMs are amenable to GPU computation resulting in another order of magnitude speedup.

3. Preliminaries

3.1. Restricted Boltzmann machines

The restricted Boltzmann machine is a bipartite, undirected graphical model with visible (observed) units and hidden (latent) units. The RBM can be understood as an MRF with latent factors that explains the input visible data using binary latent variables. The RBM consists of visible data $\mathbf{v}$ of dimension L that can take real values or binary values, and stochastic binary variables $\mathbf{h}$ of dimension K. The parameters of the model are the weight matrix $\mathbf{W} \in \mathbb{R}^{L\times K}$ that defines a potential between visible input variables and stochastic binary variables, the biases $\mathbf{c} \in \mathbb{R}^L$ for visible units, and the biases $\mathbf{b} \in \mathbb{R}^K$ for hidden units.

When the visible units are real-valued, the model is called the Gaussian RBM, and its joint probability distribution can be defined as follows:

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

$$E(\mathbf{v},\mathbf{h}) = \frac{1}{2\sigma^2}\sum_i (v_i - c_i)^2 - \frac{1}{\sigma}\sum_{i,j} v_i W_{ij} h_j - \sum_j b_j h_j.$$

where Z is a normalization constant. The conditional distribution of this model can be written as follows:

$$P(h_j = 1|\mathbf{v}) = \text{sigm}(\frac{1}{\sigma}\sum_i W_{ij} v_i + b_j), \tag{2}$$

$$P(v_i|\mathbf{h}) = \mathcal{N}(v_i; \sigma\sum_j W_{ij} h_j + c_i, \sigma^2). \tag{3}$$

where $\text{sigm}(s) = \frac{1}{1+\exp(-s)}$ is the sigmoid function, and $\mathcal{N}(\cdot;\cdot,\cdot)$ is a Gaussian distribution. Here, the variables in a layer (given the other layers) are conditionally independent, and thus we can perform block Gibbs sampling in parallel.

The RBM can be trained using sampling-based approximate maximum-likelihood, e.g., contrastive divergence approximation [21]. After training the RBM, the posterior (Equation 2) of the hidden units (given input data) can be used as feature representations for classification tasks.

3.2. Gaussian convolutional RBMs

The Gaussian restricted Boltzmann machine is defined for input data in the form of vectors and does not model spatial context effectively. Thus, to make the RBMs scalable to more realistic and larger images, Lee et al. [19] proposed the convolutional restricted Boltzmann machine (CRBM). The CRBM can be viewed as a convolutional learning algorithm that can detect salient patterns in unlabeled (image) data. A schematic description of the Gaussian CRBM is provided in Figure 2, whose energy function is defined as

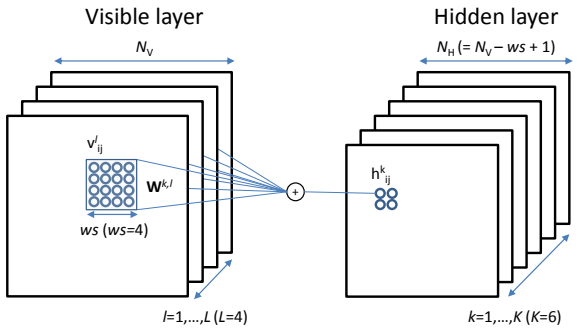

Figure 2. Illustration of a convolutional RBM. N_V and N_H refer to the size of visible and hidden layer, and ws to the size of convolution filter. The convolutional filter for the l-th channel (of size $ws \times ws$) corresponding to k-th hidden group is denoted as $\mathbf{W}^{k,l}$.

$$E(\mathbf{v},\mathbf{h}) = \frac{1}{2\sigma^2}\sum_{l=1}^{L}\sum_{i,j}(v^l_{i,j} - c_l)^2 - \sum_{k=1}^{K}\sum_{i,j} h^k_{i,j}\left(\sum_{l=1}^{L}\frac{1}{\sigma}(\widetilde{\mathbf{W}}^{k,l} * \mathbf{v}^l)_{i,j} + b_k\right), \tag{4}$$

where $\mathbf{v} \in \mathbb{R}^{N_V\times N_V\times L}$ denotes the visible nodes with L channels,[1,2] and $\mathbf{h} \in \mathbb{R}^{N_H\times N_H\times K}$ denotes the hidden nodes with K groups. The visible nodes and hidden nodes are related by the 4-d weight matrix $\mathbf{W} \in \mathbb{R}^{ws\times ws\times L\times K}$. More precisely, $\mathbf{W}^{k,l} \in \mathbb{R}^{ws\times ws}$ represents the connection between the units in k-th hidden group and l-th visible channel, and it is shared among the hidden units in the k-th group across all spatial locations. We define $\widetilde{\mathbf{W}}^{k,l}$ as the 2-d filter matrix $\mathbf{W}^{k,l}$ flipped vertically and horizontally, i.e., in Matlab notation, $\widetilde{\mathbf{W}}^{k,l} = fliplr(flipud(\mathbf{W}^{k,l}))$. The visible units in the l-th channel share the bias c_l, and the hidden units in the k-th hidden group share the bias b_k.

The conditional probability of the CRBM can be written as follows:

$$P(\mathbf{v}^l|\mathbf{h}) = \mathcal{N}\left(\mathbf{v}^l; \sigma\sum_k \mathbf{W}^{k,l} * \mathbf{h}^k + c_l, \sigma^2\mathbf{I}\right) \tag{5}$$

$$P(h^k_{i,j} = 1|\mathbf{v}) = \text{sigm}\left(\sum_l \frac{1}{\sigma}(\widetilde{\mathbf{W}}^{k,l} * \mathbf{v}^l)_{i,j} + b_k\right). \tag{6}$$

The convolutional RBM can be trained like the standard RBM using contrastive divergence. Since the CRBM is highly overcomplete, sparsity regularization [17] is used to encourage the hidden units to have sparse activations.

[1]For the simplicity of presentation, we assume that input images are "square" shaped; however, the algorithm is applicable to images of arbitrary aspect ratios.

[2]For example, L will be 128 when we use dense SIFT as an input.

4. Efficient training of RBMs

Although the RBM has shown promise in computer vision problems, it is not yet as commonly used as other unsupervised learning algorithms. We believe this is primarily due to its difficulty in training. To address this issue, we provide a novel training algorithm by exploiting the relationship between clustering methods and RBMs.

4.1. Equivalence between mixture models and RBMs with a softmax constraint

In this section, we show that a Gaussian RBM with softmax hidden units can be converted into a Gaussian mixture model, and vice versa. This connection between mixture models and RBMs with a softmax constraint completes the chain of links between K-means, GMMs, Gaussian-softmax RBMs, sparse RBMs, and convolutional RBMs. This chain of links will motivate an efficient training method for sparse RBMs and convolutional RBMs.

4.1.1 Gaussian mixture models

The Gaussian mixture model is a directed graphical model where the likelihood of visible units is expressed as a convex combination of Gaussians. The likelihood of a GMM with $K+1$ Gaussians can be written as follows:

$$P(\mathbf{v}) = \sum_{k=0}^{K} \pi_k \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \tag{7}$$

For the rest of the paper, we denote the GMM with shared spherical covariance as GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$), when $\boldsymbol{\Sigma}_k = \sigma^2\mathbf{I}$ for all $k \in \{0, 1, \ldots, K\}$. For the GMM with arbitrary positive definite covariance matrices, we will use the shorthand notation GMM($\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k$).

4.1.2 Gaussian-softmax RBMs

We define the Gaussian-softmax RBM as the Gaussian RBM with a constraint that at most one hidden unit can be activated at a time given the input, i.e., $\sum_j h_j \leq 1$. The energy function of the Gaussian-softmax RBM can be written in a vectorized form as follows:

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2}||\mathbf{v} - \mathbf{c}||^2 - \frac{1}{\sigma}\mathbf{v}^T\mathbf{W}\mathbf{h} - \mathbf{b}^T\mathbf{h} \tag{8}$$
$$\text{subj. to} \quad \sum_j h_j \leq 1$$

For this model, the conditional probabilities of visible or hidden units given the other layer can be computed as:

$$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\mathbf{v}; \sigma\mathbf{W}\mathbf{h} + \mathbf{c}, \sigma^2\mathbf{I}) \tag{9}$$

$$P(h_j = 1|\mathbf{v}) = \frac{\exp(\frac{1}{\sigma}\mathbf{w}_j^T\mathbf{v} + b_j)}{1 + \sum_{j'} \exp(\frac{1}{\sigma}\mathbf{w}_{j'}^T\mathbf{v} + b_{j'})}, \tag{10}$$

where $\mathbf{w}_j$ is the j-th column of the $\mathbf{W}$ matrix, often denoted as a "basis" vector for the j-th hidden unit. In this model, there are $K+1$ possible configurations (i.e., all hidden units are 0, or only one hidden unit h_j is 1 for some j).

4.1.3 Equivalence between Gaussian mixtures and Gaussian-softmax RBMs

As Equation 9 shows, the conditional probability of visible units given the hidden unit activations for Gaussian-softmax RBM follows a Gaussian distribution. From this perspective, the Gaussian-softmax RBM can be viewed as a mixture of Gaussians whose mean components correspond to possible hidden unit configurations.[3] In this section, we show an explicit equivalence between these two models by formulating the conversion equations between GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$) with $K+1$ Gaussian components and the Gaussian-softmax RBM with K hidden units.

Proposition 4.1. *The mixture of $K+1$ Gaussians with shared spherical covariance of $\sigma^2\mathbf{I}$ is equivalent to the Gaussian-softmax RBM with K hidden units.*

Proof. We prove by constructing the following conversions.
(1) From Gaussian-softmax RBM to GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$):
We begin by the decomposition using a chain rule:

$$P(\mathbf{v}, \mathbf{h}) = P(\mathbf{v}|\mathbf{h})P(\mathbf{h}),$$

where

$$P(\mathbf{h}) = \frac{1}{Z}\int d\mathbf{v} \exp(-E(\mathbf{v}, \mathbf{h})).$$

Since there are only a finite number of hidden unit configurations, we can explicitly enumerate the prior probabilities:

$$P(h_j = 1) = \frac{\int d\mathbf{v} \exp(-E(\mathbf{v}, h_j = 1))}{\sum_{j'} \int d\mathbf{v} \exp(-E(\mathbf{v}, h_{j'} = 1))}$$

If we define $\tilde{\pi}_j = \int d\mathbf{v} \exp(-E(\mathbf{v}, h_j = 1))$, then we have $P(h_j = 1) = \frac{\tilde{\pi}_j}{\sum_{j'} \tilde{\pi}_{j'}} \triangleq \pi_j$. In fact, $\tilde{\pi}_j$ can be analytically calculated as follows:

$$\begin{aligned}
\tilde{\pi}_j &= \int d\mathbf{v} \exp(-E(\mathbf{v}, h_j = 1)) \\
&= \int d\mathbf{v} \exp(-\frac{1}{2\sigma^2}||\mathbf{v} - \mathbf{c}||^2 + \frac{1}{\sigma}\mathbf{v}^T\mathbf{w}_j + b_j) \\
&= (\sqrt{2\pi}\sigma)^L \exp(b_j + \frac{1}{2}||\mathbf{w}_j||^2 + \frac{1}{\sigma}\mathbf{c}^T\mathbf{w}_j)
\end{aligned}$$

Using this definition, we can show the following equality:

$$P(\mathbf{v}) = \sum_j \pi_j \mathcal{N}(\mathbf{v}; \sigma\mathbf{w}_j + \mathbf{c}, \sigma^2\mathbf{I}).$$

(2) From GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$) to Gaussian-softmax RBM:
We will also show this by construction. Suppose we have the following Gaussian mixture with $K+1$ components and the shared spherical covariance $\sigma^2\mathbf{I}$:

$$P(\mathbf{v}) = \sum_{j=0}^{K} \pi_j \mathcal{N}(\mathbf{v}; \boldsymbol{\mu}_j, \sigma^2\mathbf{I}). \tag{11}$$

[3] In fact, the Gaussian RBM (without any constraints) can be viewed as a mixture of Gaussians with an exponential number of components. However, it is nontrivial to use this notion itself to develop a useful algorithm.

We can convert from this GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$) to a Gaussian-softmax RBM using the following transformations:

$$\mathbf{c} = \boldsymbol{\mu}_0 \tag{12}$$

$$\mathbf{w}_j = \frac{1}{\sigma}(\boldsymbol{\mu}_j - \mathbf{c}), j = 1, ..., K \tag{13}$$

$$b_j = \log\frac{\pi_j}{\pi_0} - \frac{1}{2}||\mathbf{w}_j||^2 - \frac{1}{\sigma}\mathbf{w}_j^T\mathbf{c}. \tag{14}$$

It is easy to see that the conditional distribution $P(\mathbf{v}|h_j = 1)$ can be formulated as a Gaussian distribution with mean $\boldsymbol{\mu}_j = \sigma\mathbf{w}_j + \mathbf{c}$, which is identical to that of the Gaussian-softmax RBM. Further, we can recover the posterior probability of hidden units given the visible units as follows:

$$\begin{aligned} P(h_j = 1|\mathbf{v}) &= \frac{\pi_j \exp(-\frac{1}{2\sigma^2}||\mathbf{v} - \sigma\mathbf{w}_j - \mathbf{c}||^2)}{\sum_{j'=0}^{K}\pi_{j'}\exp(-\frac{1}{2\sigma^2}||\mathbf{v} - \sigma\mathbf{w}_{j'} - \mathbf{c}||^2)} \\ &= \frac{\exp(\frac{1}{\sigma}\mathbf{w}_j^T\mathbf{v} + b_j)}{1 + \sum_{j'=1}^{K}\exp(\frac{1}{\sigma}\mathbf{w}_{j'}^T\mathbf{v} + b_j)} \end{aligned}$$

Therefore, a Gaussian mixture can be converted to an equivalent Gaussian RBM with a softmax constraint. □

Similarly, the Gaussian mixture with shared diagonal covariance is equivalent to the Gaussian-softmax RBM with a slightly more general energy function, where each visible unit v_i has its own noise parameter σ_i, as stated below.

Corollary 4.2. *The mixture of $K + 1$ Gaussians with a shared diagonal covariance matrix (with diagonal entries $\sigma_i^2, i = 1, ..., L$) is equivalent to the Gaussian-softmax RBM with the following energy function: $E(\mathbf{v}, \mathbf{h}) = \sum_i \frac{1}{2\sigma_i^2}(v_i - c_i)^2 - \sum_{i,j}\frac{1}{\sigma_i}v_i W_{ij}h_j - \sum_j b_j h_j$.*

Further, the equivalence between mixture models and RBMs can be shown for other settings. For example, the following corollaries can be derived from Proposition 4.1.

Corollary 4.3. *The binary RBM (i.e., when the visible units are binary) with a softmax constraint on hidden units and the mixture of Bernoulli models are equivalent.*

Corollary 4.4. *GMM($\mathbf{0}, \boldsymbol{\Sigma}_k$) with arbitrary covariance matrices and the factored 3-way RBM [22] with a softmax constraint on hidden units are equivalent.*

4.1.4 Implication

Proposition 4.1 has important ramifications. First, it is well known that K-means can be viewed as an approximation of a GMM with spherical covariance by letting $\sigma \to 0$ [23]. Compared to GMMs, the training of K-means is highly efficient; therefore, it is plausible to train K-means to provide an initialization of a GMM.[4] Then, the GMM is trained with EM, and we can convert it to an RBM with softmax units. As we discuss later, this can provide an efficient initialization for training sparse RBMs and convolutional RBMs.

4.2. Activation constrained RBMs, sparse RBMs, and convolutional RBMs

We can extend the Gaussian-softmax RBM to more general Gaussian RBMs that allow at most $\alpha \geq 1$ hidden units to be active for a given input example. We call this model the *activation constrained* RBM, and its energy function is written as follows:

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2}||\mathbf{v} - \mathbf{c}||^2 - \frac{1}{\sigma}\mathbf{v}^T\mathbf{W}\mathbf{h} - \mathbf{b}^T\mathbf{h} \tag{15}$$

$$\text{subj. to} \quad \sum_j h_j \leq \alpha$$

Note that the number of possible hidden configurations grows polynomially with α.[5] Therefore, such relaxation provides more expressive power than Gaussian mixture models. However, there is a trade-off between the expressive power (or capacity), and the tractability of exact inference and maximum-likelihood training. For example, an exact EM algorithm will require polynomial time complexity of $O(K^\alpha)$, which may be computationally expensive.

To address such difficulties, we approximate the activation constrained RBM to the sparse RBM [17]. Specifically, the sparse RBM is a variant of the RBM that is trained by adding a regularizer that encourages the average activation to be low (i.e., with target sparsity p_0) in the hidden representations. By setting $p_0 = \alpha/K$, the sparse RBM can be regarded as an approximation to the activation constrained RBM with a constraint $\sum_j h_j \leq \alpha$. The inference and training of sparse RBMs is much more efficient as α increases.

We further observe that the convolutional RBM is a generalization of the sparse RBM. Specifically, the two algorithms are equivalent when (1) the filter size of the CRBM is 1 (i.e., the convolution does not smooth the image); or (2) the filter size is the same as the image size (i.e., this is essentially equivalent to vectorizing the whole image, which is usually not interesting). For example, note that non-convolutional feature learning algorithms (e.g., sparse RBM) on SIFT descriptors would have a weight matrix of size $128 \times K$, which is equivalent to that of convolutional algorithms ($1 \times 1 \times 128 \times K$, i.e., no interactions between adjacent hidden units). In general, convolutional RBMs can model a larger spatial context using $ws \times ws \times L$ as weights for each hidden unit.

Predictions

Based on the connections described previously, we can make the following predictions:

[4] K-means learns cluster centroids and provides hard-assignment of training examples to the cluster centroids (i.e., each example is assigned to one centroid). This hard-assignment can be used to initialize GMM's parameters, such as π_k and σ, by running one M-step in the EM algorithm.

[5] If $\alpha \to \infty$ or there is no such constraint, the model is equivalent to the Gaussian RBM that has an exponential number of hidden configurations.

Algorithm 1 Efficient training algorithm for RBMs

1: Train $K+1$ centroids $\boldsymbol{\mu}_k$ via K-means.
2: Initialize GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$) parameters from K-means.
3: Train GMM($\boldsymbol{\mu}_k, \sigma^2\mathbf{I}$) via EM.
4: Initialize the RBM parameters (see Proposition 4.1).
5: Train sparse RBMs or convolutional RBMs (e.g., via contrastive divergence).

- K-means, GMM, and Gaussian-softmax RBM (with the same K) should have similar expressive power and show similar classification performance.
- When $\alpha > 1$, sparse RBMs can give better classification performance than K-means or Gaussian mixtures due to their increased expressive power.
- When convolutional filter size is larger than 1, convolutional RBMs can give better classification performance than non-convolutional RBMs since they can learn spatial context more efficiently.

These predictions will be verified with our efficient training method, which is described in the following section.

4.3. Algorithm and implementation details

The overall procedure for training sparse RBMs or convolutional RBMs is shown in Algorithm 1. In addition, we used the following methods to select hyperparameters.

4.3.1 Setting the σ automatically

As we see in the energy function of the Gaussian RBM (Equation 1), σ roughly controls the noise in the visible units. Typically, σ is fixed during training and treated as a hyperparameter that needs to be cross-validated. As an alternative, we used the following heuristic to automatically tune the σ value. Suppose that we are given a fixed set of hidden unit values $\hat{h}$, then we have the following conditional probability distribution for the Gaussian RBM:

$$P(\mathbf{v}|\mathbf{h}) = \mathcal{N}(\mathbf{v}; \sigma\mathbf{W}\hat{\mathbf{h}} + \mathbf{c}, \sigma^2 I) \tag{16}$$

If we apply the maximum likelihood estimation of σ given $\hat{\mathbf{h}}$ fixed, then σ should be the sample standard deviation of $\mathbf{v} - (\sigma\mathbf{W}\hat{\mathbf{h}} + \mathbf{c})$. Here, we use $\hat{\mathbf{h}}$ as the expectation of $\mathbf{h}$ given input $\mathbf{v}$. Thus, we update σ so that it becomes close to the reconstruction error of the training data.[6] The same method also applies to convolutional training.

4.3.2 Setting the L_2 regularization

When training RBMs, a hyperparameter for L_2 regularization (that penalizes high L_2 norm of the RBM's weight matrix W) typically has to be determined via cross validation. However, due to the connections between mixture models and RBMs discussed in Section 4.1.3, setting the L_2 regularization is straightforward. Specifically, the clustering-based initialization justifies using the L_2 regularization hyperparameter obtained from clustering models, which are often very small. In our experiments, we used 0.0001 without tuning.

In the following section, we show the efficacy of our training algorithm and provide experimental evidence for the above predictions. From the experiments, we find that a combination of moderately sparse ($1 < \alpha \ll K$) representations with a moderate amount of spatial convolution ($1 < ws \ll N_V$) performs the best for object recognition. Further, we show that our feature representation achieves state-of-the-art performance.

5. Experiments and discussions

In this section, we report classification results based on two datasets: Caltech 101 [4] and Caltech 256 [24]. In the experiments, we used SIFT as low-level descriptors, which were extracted densely from every 6 pixels with a patch size of 24. We resized the images to no larger than 300×300 pixels with a preserved aspect ratio for computational efficiency. After training the codebook, feature vectors were pooled from the 4×4, 2×2, and 1×1 subregions using max-pooling and then concatenated to single feature vectors. We used linear support vector machines [25] for training a linear classifier on randomly selected training images (with a fixed number of images per class) and then evaluated the classification accuracy on the rest of the images. We performed 5-fold cross-validation to determine hyperparameters on each randomly selected training set and reported the test accuracy averaged over 10 trials.

5.1. Caltech 101

The Caltech 101 dataset [4] is composed of 9,144 images split into 101 object categories, such as vehicles, artifacts, and animals, as well as one background category with significant variances in shape. The number of images in each class varies from 31 to 800. For fair comparisons, we performed experiments as in other studies [4, 9, 18]. Specifically, for each trial, we randomly selected $5, 10, \ldots, 30$ images from each class, including the background class, and trained a linear classifier. The remaining images from each class were tested, and the average accuracy over the classes (equal weight for each class) was reported.

We summarize the results from our proposed method and other existing methods in Table 1. Our algorithm clearly outperformed other state-of-the-art algorithms using a single type of feature. Specifically, our method breaks the record on the Caltech 101 dataset by 4.3% for 15 training images and 2.1% for 30 training images.

5.2. Caltech 256

We also tested our algorithm on a more challenging dataset, Caltech 256. Caltech 256 dataset [24] is composed

[6] Given training examples $\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(M)}\}$, we define the reconstruction error as $\sqrt{\frac{1}{LM}\sum_{i=1}^{M} ||\mathbf{v}^{(i)} - (\sigma\mathbf{W}\hat{\mathbf{h}}^{(i)} + \mathbf{c})||^2}$.

training images	5	10	15	20	25	30
Lazebnik et al. [5]	-	-	56.4	-	-	64.6
Griffin et al. [24]	44.2	54.5	59.0	63.3	65.8	67.6
Yang et al. [9]	-	-	67.0	-	-	73.2
Wang et al. [18]	51.2	59.8	65.4	67.7	70.2	73.4
Boureau et al. [7]	-	-	-	-	-	75.7
K-means (K=4096)	47.6	58.1	63.4	66.6	69.1	70.9
GMM (K=4096)	50.2	60.3	65.3	68.6	70.8	72.2
sparse RBM (K=4096)	54.2	64.0	68.6	71.2	73.1	74.9
CRBM (K=2048)	56.5	66.4	70.7	73.5	75.4	77.4
CRBM (K=4096)	**56.7**	**66.7**	**71.3**	**74.2**	**76.2**	**77.8**

Table 1. Average test classification accuracies for Caltech 101.

training images	15	30	45	60
Griffin et al. [24]	28.30	34.10	-	-
Gemert et al. [27]	-	27.17	-	-
Yang et al. [9]	27.73	34.02	37.46	40.14
Wang et al. [18]	34.36	41.19	45.31	47.68
CRBM (K=4096)	**35.09**	**42.05**	**45.69**	**47.94**

Table 2. Average test classification accuracies for Caltech 256.

of 30,607 images split into 256 object categories with more variabilities and finer classifications, as well as one "clutter" class of random pictures. Each class contains at least 80 images; the objects in each image are more variant in size, location, pose, etc., than those of Caltech 101 dataset. We followed the standard experimental settings from the benchmarks [24, 9], and the overall classification accuracies were averaged over 10 random trials. The summary of the results is reported in Table 2. Our algorithm performed slightly better than the LLC [18] algorithm, with considerably large margins to many other methods on Caltech 256 dataset.

5.3. Analysis of hyperparameters

To provide a better understanding of our proposed algorithm, we give a detailed analysis of the hyperparameters: sparsity and convolutional filter size. We performed the following experiments on Caltech 101 dataset while fixing the number of bases to 1024. In most cases, we observed significant improvements as the number of hidden bases increased, which is consistent with what others have reported [26]. All results reported in this section are validation accuracies (5-fold cross validation on the training set).

5.3.1 Sparsity (α/K)

The performance of sparse models, such as sparse coding and sparse RBMs, can vary significantly as a function of sparsity level. As we discussed in Section 4, the sparse RBM can be more expressive than K-means or GMMs. While K-means and GMM have sparsity of $1/K$ on average (i.e., allow only one cluster to be active for a given input), the sparse RBM can control sparsity by setting the target sparsity value $p_0 = \alpha/K$.

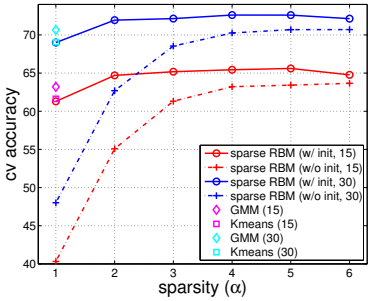

Figure 3. Average cross-validation accuracy on Caltech 101 dataset with 1024 bases using K-means, GMM, and sparse RBM with different sparsity values. The "sparse RBM (w/ init)" denotes the sparse RBM initialized from GMM as described in Section 4; the "sparse RBM (w/o init)" denotes the sparse RBM initialized randomly (baseline). Blue and cyan represent settings with 30 training images per class. Red and magenta represent settings with 15 training images per class.

In this experiment, we compared two settings for sparse RBM training—one by initializing from GMM as described in Section 4, and the other by initializing randomly (baseline). Figure 3 shows the average validation accuracies as a function of sparsity (α/K). Compared to the K-means and GMM, the sparse RBM with random initialization performed very poorly in the low α regime (i.e., when its corresponding number of activations is roughly 1). However, by using an efficient training method described in Section 4, the sparse RBM performs as well as K-means and GMM when the target sparsity is close to $1/K$, and significantly outperforms K-means and GMM when the representation is less sparse. Overall, the effect of accurate initialization is striking, especially in the high sparsity regime, and the best validation accuracy (maximum over α) was improved by 2% for both 15 and 30 training images.

5.3.2 Convolution filter size

Convolutional learning is powerful since it captures the spatial correlation between neighboring descriptors (e.g., pixels or dense SIFT) more efficiently than non-convolutional learning. The size of the filter, however, should be selected carefully. For instance, it is difficult to capture enough spatial information with small size filters; on the other hand, overly large filter size can result in severe over-smoothing of small details in the image. Therefore, we investigate how the filter size affects the performance.

In this experiment, we fixed the number of bases ($K = 1024$) and sparsity ($\alpha = 4$) while varying the convolutional filter size from 1 to 5. As shown in Figure 4, the filter size of 3 resulted in highest validation accuracies. We also observed improvements (up to 2%) using the clustering-based initialization method in the convolutional setting.

Overall, our experimental analysis confirms that a moderately sparse ($1 < \alpha \ll K$) representation that is convolutionally trained with sufficient spatial context ($1 < ws \ll$

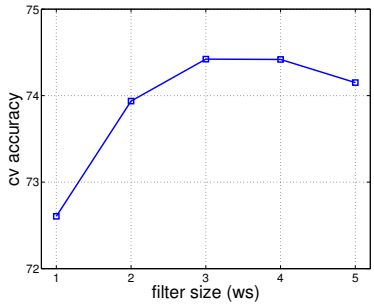

Figure 4. Average cross-validation accuracy on the Caltech 101 dataset with 1024 bases and different convolution filter sizes (ws).

N_V) outperforms both clustering methods (e.g., K-means and GMMs) and non-convolutional counterparts (e.g., non-convolutional sparse RBMs).

6. Conclusion

In this paper, we proposed a mid-level feature extraction method using convolutional RBMs. Our key idea is to investigate an efficient training method for sparse RBMs and convolutional RBMs through the connections between mixture models and RBMs. In our experiments, we show efficacy of our training algorithm, as well as the benefit of learning sparse, distributed, convolutional feature representations. Overall, our method achieves state-of-the-art performance in object recognition benchmarks.

Acknowledgments

This work was partially supported by a Google Faculty Research Award and ARO Grant W911NF-09-1-0310.

References

[1] D. Lowe, "Object recognition from local scale-invariant features," in *ICCV*, 1999. 1, 2

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005. 1, 2

[3] A. Agarwal and B. Triggs, "Hyperfeatures–multilevel local coding for visual recognition," *ECCV*, pp. 30–43, 2006. 1

[4] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories," in *CVPR*, 2004. 1, 6

[5] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR*, 2006. 1, 2, 7

[6] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *ICCV*, 2005. 1

[7] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *CVPR*, 2010. 1, 2, 7

[8] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *NIPS*, 2009. 1

[9] J. Yang, K. Yu, Y. Gong, and T. S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, pp. 1794–1801, 2009. 1, 2, 6, 7

[10] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006. 1

[11] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *NIPS*, 2007. 1

[12] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *NIPS*, pp. 1137–1144, 2006. 1

[13] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. LeCun, "Learning convolutional feature hierachies for visual recognition," in *NIPS*, 2010. 1, 2

[14] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996. 1

[15] P. Smolensky, "Information processing in dynamical systems: Foundations of harmony theory," *Parallel distributed processing: Explorations in the microstructure of cognition*, vol. 1, pp. 194–281, 1986. 1

[16] G. E. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006. 1

[17] H. Lee, C. Ekanadham, and A. Y. Ng, "Sparse deep belief network model for visual area V2," in *NIPS*, 2008. 1, 3, 5

[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010. 2, 6, 7

[19] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *ICML*, 2009. 2, 3

[20] M. Zeiler, D. Krishnan, G. Taylor, and R. Fergus, "Deconvolutional networks," in *CVPR*, 2010. 2

[21] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002. 3

[22] M. Ranzato, A. Krizhevsky, and G. E. Hinton, "Factored 3-way Restricted Boltzmann Machines for Modeling Natural Images," in *AISTATS*, 2010. 5

[23] C. Bishop, *Pattern recognition and machine learning*, vol. 4. Springer New York, 2006. 5

[24] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," tech. rep., California Institute of Technology, 2007. 6, 7

[25] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008. 6

[26] A. Coates, H. Lee, and A. Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," in *AISTATS*, 2011. 7

[27] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders, "Kernel codebooks for scene categorization," in *ECCV*, vol. 3, 2008. 7