Stacks of Convolutional Restricted Boltzmann Machines for Shift-Invariant Feature Learning

Mohammad Norouzi, Mani Ranjbar, and Greg Mori
School of Computing Science
Simon Fraser University
Burnaby, BC Canada
{mohammad,mra33,mori}@cs.sfu.ca

Abstract

In this paper we present a method for learning class-specific features for recognition. Recently a greedy layer-wise procedure was proposed to initialize weights of deep belief networks, by viewing each layer as a separate Restricted Boltzmann Machine (RBM). We develop the Convolutional RBM (C-RBM), a variant of the RBM model in which weights are shared to respect the spatial structure of images. This framework learns a set of features that can generate the images of a specific object class. Our feature extraction model is a four layer hierarchy of alternating filtering and maximum subsampling. We learn feature parameters of the first and third layers viewing them as separate C-RBMs. The outputs of our feature extraction hierarchy are then fed as input to a discriminative classifier. It is experimentally demonstrated that the extracted features are effective for object detection, using them to obtain performance comparable to the state-of-the-art on handwritten digit recognition and pedestrian detection.

1. Introduction

The success or failure of an object recognition algorithm hinges on the features used. Successful algorithms have been built on top of hand-crafted gradient response features such as SIFT [1] and histograms of oriented gradients (HOG) [2]. While their successes have been demonstrated in a variety of domains, they are fixed features that cannot adapt to model the intricacies of a particular problem. A competing approach, followed in this work, is to learn features specifically tuned for a particular object recognition task.

Consider the features shown in Figure 1. These features for pedestrian detection were automatically learned by our algorithm. The algorithm learns individual features corresponding to areas around the head, feet, and inverted-"V" patterns around the legs. Hand-crafting larger scale features such as these would be quite difficult. We demonstrate that combining these task-specific features with the generic HOG features leads to state-of-the-art performance on the challenging INRIA pedestrian detection benchmark.

Figure 1. Large-scale features learned by the proposed model for pedestrian detection. Each plate corresponds to a set of image patches with highest compatibility with a feature.

The algorithm we develop is based on the Restricted Boltzmann Machine (RBM) [3]. The RBM is a probabilistic model for a density over observed variables (e.g., over pixels from images of an object) that uses a set of hidden variables (representing presence of features). In the standard RBM all observed variables are related to all hidden variables by different parameters. While this model can be used to create features describing image patches, it does not explicitly capture the spatial structure of images. Instead, we incorporate ideas from the convolutional neural network (CNN) of LeCun et al. [4] and develop a model called *Convolutional RBM* (C-RBM). We define patterns of weight sharing amongst hidden variables that respect the spatial structure of an entire image, and pooling operations to aggregate these over areas of an image. Chaining these operations together in a multilayer hierarchy, we train stacks of C-RBMs that are able to extract large-scale features tuned to a particular object.

The main contribution of this paper is the development of the C-RBM model. We modify the standard RBM and learning algorithm to include spatial locality and weight sharing. We develop these in a generative framework for layerwise training of multilayer convolutional feature de-

978-1-4244-3991-1/09/$25.00 ©2009 IEEE

tectors. The framework learns a small set of stochastic features that model a distribution over images of a specific object class, which are then used in a discriminatively trained classifier. We demonstrate experimentally that this generative feature learning for a discriminative classifier is effective, using the learned features to obtain state-of-the-art performance on object recognition tasks.

2. Previous work

The idea of building hierarchical structures of features for object detection has deep roots in the computer vision literature, with the development of many such models inspired by allusions to the human visual system [5, 4, 6, 7, 8]. The HMAX model [6, 7] applies a layer of hand-crafted Gabor filters to the input image, followed by layers of maximum pooling and comparisons to a set of stored prototypes. Unlike stacks of C-RBMs developed in this paper, in the HMAX model learning is not performed on the feature extraction layers.

LeCun et al. [4] developed the convolutional neural network (CNN), a specialized type of neural network in which weight sharing is employed with the result that the learned weights play the role of convolution kernels and can be interpreted as a set of tuned feature detectors. In the original CNN work, the multilayer hierarchical model was learned using error back-propagation on a labeled training set. More recently, Ranzato et al. [9, 8] proposed an unsupervised energy-based algorithm for separately training layers of a CNN. This algorithm minimizes a loss function that involves encoding and decoding square errors, and regularization. Also, sparsity is incorporated via sparsifying logistic, and explicit shift-invariance is built into the model. In contrast to this line of work, the proposed C-RBM defines a probabilistic model over images and features, using the criterion of modeling the distribution over input images rather than a non-probabilistic loss minimization. Further, shift-invariance is implicitly handled in the C-RBM model.

Hinton et al. [10, 11] proposed a greedy layerwise procedure for training a multilayer belief network. Layers are trained separately, and bottom-up where each layer is seen as a RBM (more details below). This model has been applied to MNIST digits, faces, and natural image patches. The fundamental difference between the C-RBM model and the standard RBM is in the weight sharing and consequent reuse of filters. As we will describe below, the C-RBM is trained by modeling distributions over entire images rather than image patches. As such, the set of filters learned will implicitly tend to be shift-invariant since spatially-aware filters will be used to reconstruct entire images rather than individual patches.

Roth and Black [12] developed the *Fields of Experts* (FoE) model, for use in constructing image priors. The model defines a probability distribution over entire images as a product of patch potentials. While similar in spirit, FoEs and C-RBMs are different mainly because of their distinct patch potentials. Similarly, RBM and product of Student's t experts [13] are different. This difference is crucial because as noted in [14] the quadratic potential function of FoE model favors filters with close to zero responses on training images and non-zero responses elsewhere. In contrast, the C-RBM's linear patch potential favors filters with high response on training images. This leads to extracting frequent patterns of training images e.g., oriented edges in place of noisy filters learned in [12].

In this work we learn a set of features using a hierarchy of C-RBMs, which we pass into a supervised learning algorithm to learn a classifier for object detection. We perform experiments on two datasets: the MNIST handwritten digits and the INRIA pedestrian detection benchmark [2].

An enormous number of methods has been applied to the MNIST digit dataset. The aforementioned neural network-based approaches [9, 8] attain some of the best results on this dataset. For smaller numbers of training images, the *patchwork of parts* (PoP) model of Amit and Trouvé [15], which learns a deformable model of edge parts, attains excellent accuracy. Our method obtains results competitive with the state-of-the-art on MNIST.

A substantial volume of previous work on pedestrian detection also exists. State-of-the-art results on the INRIA pedestrian detection dataset include Tuzel et al. [16] and Maji et al. [17]. Tuzel et al. use a set of covariance descriptors describing the statistics of pixels in sub-regions of a pedestrian image, and develop a classifier for the Riemannian manifold on which such covariance descriptors lie. Our work focuses on automatically learning a set of features for detection, and obtains similar results with a generic SVM-based classifier. Maji et al. develop an efficient algorithm for using a *histogram intersection* kernel SVM on top of multi-scale HOG features. This kernel significantly improves results on the INRIA dataset, and could be combined with the features we learn using the C-RBMs.

3. Restricted Boltzmann Machine

The Restricted Boltzmann Machine (RBM) [3] is a two layer undirected graphical model that consists of a layer of observed and a layer of hidden random variables, with a full set of connections between them. It is a generative framework that models a distribution over visible variables by introducing a set of stochastic features. In this paper we associate the RBM's observed variables with image pixels and hidden variables with features.

A key characteristic of the RBM is that its stochastic hidden units are conditionally independent given the observed data. This property makes each hidden unit an independent expert of a specific feature. Therefore, the RBM is an instance of the product of experts model.

The probability of observed variables in an RBM with parameter set θ is defined according to a joint energy of visible and hidden units $E(\mathbf{v}, \mathbf{h}; \theta)$, as a Gibbs distribution

$$p(\mathbf{v}; \theta) = \frac{1}{Z(\theta)} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)} \ , \quad (1)$$

where $\mathbf{v}$ and $\mathbf{h}$ denote vectors of visible and hidden variables, and $Z(\theta)$ is the normalization constant.

Commonly RBMs refer to a model with binary hidden and binary visible random variables. In this paper we call such model a *binary RBM*. A modification of the binary RBM's energy function makes it suitable for modeling a density over continuous visible units, while hidden units are binary [18]. We call the second model *continuous RBM*. A continuous RBM is appropriate for modeling natural images at pixel level, while a binary RBM is applicable to observable variables corresponding to hidden layer of another RBM or quasi-binary images (e.g., handwritten digits).

The energy functions of the binary and continuous RBMs are defined as E_1 and E_2 respectively,

$$E_1(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i,j} v_i w_{ij} h_j - \sum_i b_i v_i - \sum_j c_j h_j \ , \quad (2)$$

$$E_2(\mathbf{v}, \mathbf{h}; \theta) = E_1(\mathbf{v}, \mathbf{h}; \theta) + \frac{1}{2} \sum_i {v_i}^2 \ , \quad (3)$$

where variables i and j iterate over observed and hidden units respectively, and $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$ is the model parameter set. The matrix $\mathbf{W}$ determines the symmetric interaction between pairs of hidden and visible units, and parameters $\mathbf{b}$ and $\mathbf{c}$ are bias terms that set the unary energy of the variables.

Inference in RBMs is straightforward. In the binary RBM conditionals are of the form

$$p(h_j = 1 | \mathbf{v}) = \sigma(c_j + \sum_i v_i w_{ij}) \ , \quad (4)$$

$$p_1(v_i = 1 | \mathbf{h}) = \sigma(b_i + \sum_j w_{ij} h_j) \ , \quad (5)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic sigmoid function. For the continuous RBM, (4) still holds, but the conditional distribution of visible units is a normal,

$$p_2(v_i | \mathbf{h}) = \mathcal{N}(b_i + \sum_j w_{ij} h_j, 1) \ , \quad (6)$$

where the variance is set to one because in a pre-processing stage visible units can be scaled with an arbitrary value.

3.1. Contrastive Divergence Learning

Ideally we want to learn RBM parameters by maximizing the likelihood in a gradient ascent procedure. The gradient of the log-likelihood for an energy-based model is

$$\frac{\partial}{\partial \theta} L(\theta) = -\left\langle \frac{\partial E(\mathbf{v}; \theta)}{\partial \theta} \right\rangle_{data} + \left\langle \frac{\partial E(\mathbf{v}; \theta)}{\partial \theta} \right\rangle_{model} \ , \quad (7)$$

where $E(\mathbf{v}; \theta)$ is the free energy of $\mathbf{v}$, and $\langle . \rangle_{data}$ and $\langle . \rangle_{model}$ denote expected value over all possible visible vectors $\mathbf{v}$ with respect to the data and model distribution. For an RBM $E(\mathbf{v}; \theta) = -\log \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h}; \theta)}$ and $\partial E(\mathbf{v}; \theta)/\partial \theta = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{v}; \theta) \partial E(\mathbf{v}, \mathbf{h}; \theta)/\partial \theta$. Unfortunately computing expected value regarding an RBM distribution involves an exponential number of terms, which makes it intractable. However, Hinton [19] proposed another objective function called *contrastive divergence* (CD) that can be efficiently minimized during training as an approximation to maximizing the likelihood.

During contrastive divergence learning a Gibbs sampler is initialized at each data point and is run for a small number of steps (n) to obtain an approximation of the model distribution. For an RBM the CD update rule of parameter w_{ij} becomes

$$w_{ij} = w_{ij} + \eta(\langle v_i^0 h_j^0 \rangle - \langle v_i^n h_j^n \rangle) \ , \quad (8)$$

where η is the learning rate, the random variable $\mathbf{v}^0$ takes value from the data distribution, $\mathbf{h}^0$ is obtained according to $p(\mathbf{h}^0|\mathbf{v}^0)$, random variable $\mathbf{v}^n$ takes value from sampled data generated by n full steps of Gibbs sampling, and $\mathbf{h}^n$ is again obtained from $p(\mathbf{h}^n|\mathbf{v}^n)$. It has been formally demonstrated that minimizing the contrastive divergence is an approximation of maximizing the likelihood [19].

3.2. Layerwise Training for Stacks of RBMs

Consider learning a fully connected multilayer belief network with a layer of observable variables $\mathbf{v}$ and a number of hidden layers $\mathbf{h_1}, \mathbf{h_2}, \ldots$. Discriminative or generative learning of this hierarchy has two major problems. First, the effect of the likelihood gradient on bottom layer parameters drastically decreases as depth increases. Second, employing more hidden units might not improve the model performance since some units might become inactive, which causes their error feedback to become zero, and they stay inactive.

To tackle these issues, Hinton et al. [20] proposed a greedy layerwise algorithm that views a multilayer belief network as a stack of RBMs. In this method parameters of the bottom-most layer, θ_1, are learned by training a single-layer RBM between $\mathbf{v}$ and $\mathbf{h_1}$. Subsequently, the first layer parameters are frozen and conditional probabilities of first layer hidden units, $p(\mathbf{h_1}|\mathbf{v}; \theta_1)$, are used to generate data for training a second RBM between $\mathbf{h_1}$ and $\mathbf{h_2}$. Additional layers can be added on top of the model similarly. If the size of hidden layers does not decrease, it can be proved that training additional layers increases a variational lower bound on the likelihood, though the likelihood might fall [20].

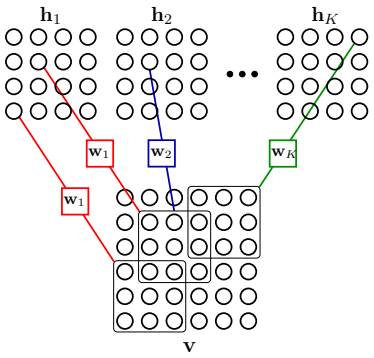


Figure 2. A C-RBM with 3×3 feature kernels. The hidden units are partitioned into K feature maps, each extracting a particular feature denoted by $\mathbf{w}_k$ from a 3×3 neighborhood of visible units.

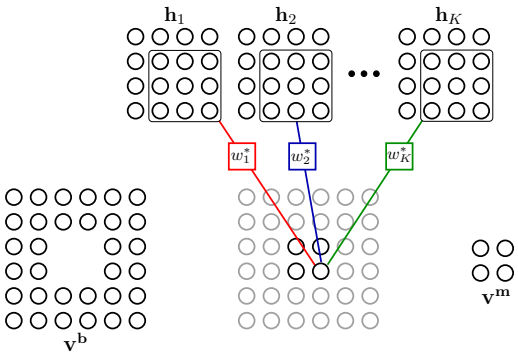


Figure 3. A C-RBM with 3×3 neighborhoods from the view of visible units. Observable units are divided into middle $\mathbf{v}^{\mathbf{m}}$ and boundary $\mathbf{v}^{\mathbf{b}}$ regions. We can sample from units of $\mathbf{v}^{\mathbf{m}}$ having the configuration of hidden unit using the flipped filters $\{\mathbf{w}_k^*\}$.

4. Convolutional RBM

In the standard RBM all observed variables are related to all hidden variables by different parameters. Using an RBM for extracting global features from complete images is not very helpful for object detection. Describing images in terms of local features needs fewer parameters, generalizes better, and offers re-usability as identical local features can be extracted from different locations of an image. Hence, an RBM can be trained on patches sampled from images to create local features. However, this approach does not respect the spatial relationship of image patches. Therefore, features extracted from neighboring patches become independent.

To tackle this problem, we introduce an extension of RBM, called C-RBM. In a C-RBM, features extracted from neighboring patches complement each other and cooperate to reconstruct the image. Unlike a patch-based RBM, a C-RBM is trained on complete images or large regions of them to exploit spatial structure of neighboring patches. A C-RBM has a visible and hidden layer that are connected by sets of local and shared parameters. This connection scheme, called convolutional, has been used in models such as [4], but a significant difference here is that convolutional connections are employed in a generative MRF architecture.

In the C-RBM, hidden units $\mathbf{h}$ are divided into K partitions, $\{\mathbf{h}_k\}_{k=1}^K$, each called a *feature map*. Variables of each feature map are connected via identical $x \times y$ filters to different $x \times y$ neighborhoods of observed variables e.g., pixels (Fig. 2). Hence, binary hidden units of each feature map represent the presence of a particular feature in different locations of an image. Let $\mathbf{w}_k$ denote parameters of the filter that connect $\mathbf{h}_k$ units and different neighborhoods of image $\mathbf{v}$.

In our notation, matrices are named by capital letters (or initial capital words) and lower-case of same names represent vectors obtained by concatenating elements of the corresponding matrices. Thus, W_k is the k^{th} filter (a $x \times y$ matrix) and $\mathbf{w}_k$ is its vector. To formulate the energy function of a C-RBM, we need to denote a certain subwindow of an image. Let $V_{(q)}$ be an $x \times y$ subwindow of image V with top-left corner at pixel q, and $\mathbf{v}_{(q)}$ be its corresponding vector. Later we will use $\mathbf{h}_{k(r)}$ to denote a $x \times y$ subwindow of the k^{th} feature map with top-left corner at r.

The joint energy of hidden and observed variables in a C-RBM is defined similar to that of an RBM as

$$E_3(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{k,q} h_{kq} \left(\mathbf{w}_k^{\mathsf{T}} \mathbf{v}_{(q)}\right) - \sum_i b v_i - \sum_{k,q} c_k h_{kq} \quad (9)$$

where q iterates over pixels of $\mathbf{v}$ with valid $\mathbf{v}_{(q)}$, $\theta = \{\{\mathbf{w}_k\}, b, \mathbf{c}\}$, and h_{kq} denotes a hidden unit of feature map H_k with coordinates q. Essentially, a continuous variant of C-RBM can be defined with an energy modification similar to (3). In the energy function of (9), identical bias terms are applied to all visible units and hidden units of each feature map. This is due to our assumption that different visible variables have similar statistics and different features are equally probable in different locations of images. When this is not the case, specific bias terms can be associated with different units in the energy function.

According to (9), the conditional probability of hidden units is given by

$$p(h_{kq}=1|\mathbf{v}) = \sigma\left(\mathbf{w}_k^{\mathsf{T}} \mathbf{v}_{(q)} + c_k\right). \quad (10)$$

However, the conditional probability of visible units needs more careful treatment, because the boundary units are within a smaller number of subwindows compared to the interior pixels. As an extreme case, the top left pixel only appears in K patches, while a middle pixel may contribute to Kxy features. This problem is caused by the asymmetry of the C-RBM connections from the viewpoint of visible units. To make the model symmetric, one may extend the

C-RBM's connections to an infinite field of hidden and visible variables, but this approach requires unbounded training signals. Instead, we divide the visible units into two partitions of boundary and middle variables. Let $\mathbf{v^b}$ denote a strip of boundary variables including $x-1$ margin pixels from left and right, and $y-1$ pixels form top and bottom of $\mathbf{v}$, while $\mathbf{v^m}$ represents the interior pixels. Clearly all nodes in $\mathbf{v^m}$ are connected with the same set of filters to different feature maps as depicted in Fig. 3. These filters, denoted by $\{\mathbf{w}_k^*\}$, are horizontally and vertically flipped version of the original filters.

The conditional probability of the interior observable units is

$$p(v^m_r=1|\mathbf{h}) = \sigma\left(\sum_k \mathbf{w}_k^{*\mathsf{T}}\mathbf{h}_{k(r)} + b\right) \ , \qquad (11)$$

where r iterates over pixels of $\mathbf{v^m}$. Furthermore, the conditional probabilities of boundary pixels cannot be computed accurately as noted above.

An important characteristics of filters learned using a C-RBM is that they are shift invariant, meaning that none of filters can be reconstructed by translating another filter. Shift invariance is a desired property of visual features because it can considerably decrease the number of features needed. In a C-RBM this property arises from the fact that each filter is applied to all overlapping neighborhoods of an image. Hence, learning two filters that are translations of each other, is unlikely since it does not increase the likelihood of filters given the training data.

4.1. Shift-invariant Feature Learning

The C-RBM parameters $\{\mathbf{w}_k\}$ and b are learned by minimizing the contrastive divergence (CD). We do the required Gibbs sampling from the C-RBM distribution by sampling from hidden variables given the visible ones, and next from observed variables given the hidden ones. However, we do not have enough features describing the boundary pixels, so we cannot sample from them precisely. Further, doing a number of Gibbs sampling steps might cause uninformative samples of the boundary pixels to propagate over other pixels.

Therefore, instead of maximizing the full data log-likelihood we (approximately) maximize the log-likelihood conditional on the image boundaries $\sum_{\mathbf{v}} \log p(\mathbf{v^m}|\mathbf{v^b};\theta)$. Under the new setup, CD learning is still possible since we can sample from the conditional distribution $p(\mathbf{v^m}|\mathbf{v^b};\theta)$ by n Gibbs sampling steps. This sampling is done by consecutive sampling from $\mathbf{h}$ given $\mathbf{v}$ (10) and from interior image region $\mathbf{v^m}$ given $\mathbf{h}$ (11). Then the image boundary is concatenated with the interior pixels to provide data for another sampling step.

Pseudocode for CD training of binary C-RBMs (using one step of Gibbs sampling) is provided in Alg. 1. Interestingly, most of the required computations can be expressed in terms of valid filtering operations. For instance, the gradient of the joint energy of a C-RBM (9) is given by

$$\frac{\partial E_3(V,H;\theta)}{\partial W_k} = Filter(V, H_k) \ , \qquad (12)$$

where $Filter(A,B)$ filters matrix A with filtering kernel B. This filtering is performed without extending the input matrix A, so the result will be smaller than A.

Algorithm 1 Stochastic parameter update of a binary C-RBM. Inputs are a 2D matrix $V0$, learning rate η, and K filters $\{W_k\}_{k=1}^K$.

for $k=1$ to K **do**
  $PH0_k \leftarrow \sigma(Filter(V0, W_k) + c_k)$
  $Grad0_k \leftarrow Filter(V0, PH0_k)$
  $H0_k \sim Bernoulli(PH0_k)$
end for
$V1^m \leftarrow \sigma(\sum_{i=1}^K Filter(H0_k, {W_k}^*) + b)$
$V1 \leftarrow Concatenate(V0^b, V1^m)$
for $k=1$ to K **do**
  $PH1_k \leftarrow \sigma(Filter(V1, W_k) + c_k)$
  $Grad1_k \leftarrow Filter(V1, PH1_k)$
  $W_k \leftarrow W_k + \eta(Grad0_k - Grad1_k)$
end for

4.2. Sparsity

An issue in learning C-RBMs is the overcompleteness of features. Note that because of the convolutional connections, the feature space of a C-RBM with K feature types is almost K times overcomplete. Although CD learning can deal with overcompleteness [21], our experiments indicate that it cannot handle this highly overcomplete representation (more feature maps is better). What happens is that after a few iterations of parameter update, sampled images become very close to the original ones, and the learning signal disappears. Increasing the number of Gibbs sampling steps is a solution to this problem; but it is time consuming. Alternatively, we add sparsity to the hidden features, which constrains the information content of each feature map. This helps the learning signal become stronger and more hidden units contribute in reconstruction of images.

Previously, Lee et. al [22] developed a sparse variant of RBMs. In their model the $\mathbf{c}$ parameters that control the sparsity of hidden units, are tuned at each learning iteration to obtain a fixed small activation probability for hidden units (using square loss). Also in [23], sparsity is added to RBMs by continuously decreasing a small constant from hidden bias terms c. Instead, In our experiments, we froze the $\mathbf{c}$ parameters at a negative fixed constant and kept them unchanged during training.

4.3. Multilayer C-RBMs

Our hierarchy of C-RBMs is trained in a layerwise and bottom-up procedure similar to a stack of RBMs. The only difference is that after each C-RBM feature extraction layer, a deterministic subsampling layer aggregates features over local areas of images. This subsampling is performed by taking the maximum conditional feature probability, $p(\mathbf{h_1}|\mathbf{v})$, over non-overlapping subwindows of feature maps. This deterministic max pooling layer makes the features invariant to small distortions and shifts. The next C-RBM layer is trained on the subsampled conditionals of the lower level features. Again, another deterministic max pooling layer is stacked on top of the feature detectors. In our model, we stop after the fourth layer, and use the final subsampled feature probabilities as the input of a discriminative classifier.

5. Experiments and Implementation Details

We experimentally evaluated the discriminative strength of the proposed shift-invariant feature learning method. We performed two sets of experiments on handwritten digit recognition (MNIST dataset) and pedestrian detection (INRIA person dataset). Obtaining state-of-the-art accuracy in these two different tasks demonstrates the robustness of our feature learning algorithm and its capability to extract task-dependent features.

For each task, we construct a separate four layer feature extractor in bottom-up layerwise manner. The second and fourth layers of this hierarchy are deterministic max pooling layers that do not have any free parameter except the subsampling widow size. The first and third layers are the convolutional connections that are tuned by CD learning of separate C-RBMs. Finally, a discriminative layer (SVM) is trained on the fourth layer outputs to do classification. For pedestrian detection, we combine our large-scale features with fine-scale HOG descriptors [2] and train the final SVM on the combination of these features. We employ the RBF kernel for digit recognition and the linear kernel for pedestrian detection.

Ranzato et. al [8] reported that although generative feature learning procedure benefits from a sparsifying non-linearity, the final discriminative classifier achieves better accuracy when the non-linearity is relaxed and features become less sparse. Our experiments supported this relaxation too. Thus, after the feature learning phase, we relaxed the bias parameters and scaled down the weights to obtain less sparse features.

For CD learning, we did batch gradient update with an additional momentum from the previous step's gradient. We subdivided the training data into batches of roughly 100 examples. In the gradient descent procedure, learning rate is important because some high and low values of η suppress some of C-RBM's feature maps to become always inactive, and in fact dismiss some of features. We tested a set of different values for η and selected the one with more number of active features. In the following we present results and specific details of MNIST and INRIA experiments.

Model	Error	Feature size
LeNet-5, rbf-SVM [24]	0.83%	150
Multilayer C-RBM, rbf-SVM	**0.67%**	1225
Large CNN, supervised [9]	0.60%	3200
Large CNN, unsupervised [8]	0.64%	3200

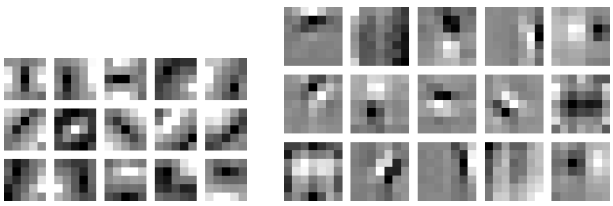
Table 1. MNIST Error rate when all the training images used for training. The models are not allowed to extend the training set by transforming the images.

Training	C-RBM	Large CNN[8]	PoP[15]
60000	0.67	0.64	0.68
20000	0.84	0.76	-
10000	1.11	0.85	.8
5000	1.45	1.52	1.52
2000	2.26	2.53	-
1000	2.86	3.21	2.14
300	5.18	7.18	3

Table 2. MNIST error rates as function of training set size used

5.1. MNIST handwritten digits

We used the full MNIST training images, without considering the digit labels, to train a four layer hierarchical feature detector. First layer filters, depicted in Fig. 4a, and third layer large-scale features, illustrated in Fig. 5, were learned. Next, different amount of labeled data was provided to an RBF SVM, which was trained using the topmost layer features. We evaluated all the models on the full test set. Table 1 shows the error rate for the case that all the labels used to train the models. It is hard to directly compare these error rates due to many details involved in each of the models. The feature size column gives an intuition about the size of the models. Table 2 shows the error rate as a function of the amount of labeled data.



(a) (b)

Figure 4. (a) First layer 5×5 filters obtained by training C-RBM on handwritten digit images (b) First layer 7×7 filters learned from INRIA positive training set

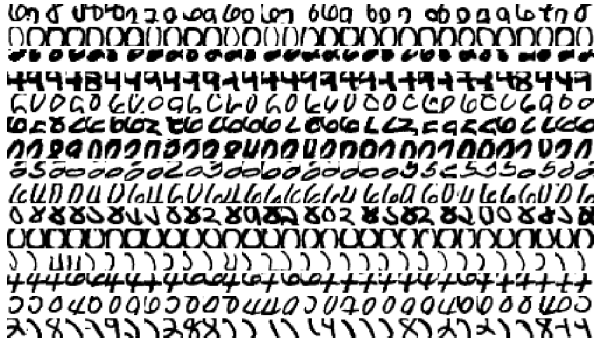


Figure 5. Each row corresponds to a set of 14×14 patches from different images that have highest compatibility with a third layer feature. The 10 illustrated out of 50 features are selected randomly.

The Ranzato et al. [8] model, similar to ours, is a four layer feature detector followed by a discriminative classifier. Their feature detector hierarchy is trained layerwise using an energy minimization approach. The model of [8] is about 2.5 times larger than ours, and a two layer neural network was used as the final classifier. Although our model delivers slightly higher error rate on MNIST, it is much smaller and achieves lower error when fewer labeled training data were accessible (Table 2). However, We believe by training larger models we will be able to improve the accuracy for the full MNIST task as well.

The input of our feature extractor is a 32×32 image obtained by evenly zero padding an original 28×28 mnist digit. As the first layer, we used 15 filters of 5×5 pixels, followed by a sigmoid non-linearity. Second layer includes maximum subsampling operators over 2×2 non-overlapping subwindows. The third layer filters are 3-dimensional, and operate on the 15 subsampled feature maps. Each filter has $15 \times 5 \times 5$ parameters that encode a combination of 5×5 patches of lower level feature maps. We employed 50 of these third layer features, followed by non-linearity and again 2×2 max pooling operations. As the final discriminative layer, We combined 10 one-vs-rest binary SVMs, and built a ten-class digit classifier.

We set the bias terms for both first and second C-RBMs to -6. This adds the desired sparsity to the features. In our experiments we observed that having a non-sparse separate feature helps the features to converge better. Hence, we added a feature with fixed 0 bias to the C-RBM features. This additional feature usually takes care of the background, and becomes active on the zero background regions. The parameters of the RBF SVM (c and g) were tuned by 5-fold cross validation for small training sets and using a validation set for larger trainings.

5.2. INRIA pedestrian detection benchmark

The INRIA person dataset is one of the most challenging datasets for pedestrian detection. This dataset includes 2416 positive training bounding boxes of size 128×64, and 1218 negative training images of different sizes. It includes 1132 positive test images of size 128×64, and 453 negative test images. This dataset involves extreme illuminations, occlusion, background clutter, and a range of human poses. The results on INRIA are reported as miss rate vs. different False Positive Per detection Window (FPPW) rates.

While in the digit recognition task background is very simple, in the INRIA dataset several types of background clutter appear in images including objects similar to human parts and parts of other humans in the background. Consequently, in addition to our part-like features, we need to have a template for the human figure. This template helps the model to rule out images containing spurious parts, and achieve highly accurate results. We combine our features learned from INRIA training set with well-known HOG features because of two reasons. First, HOG features are finer-scale and help the SVM to create the human figure template. Second, we examine the performance gain of our features over HOG features that are tuned to the INRIA dataset [2].

The feature learning is performed only on the INRIA positive training set, to make the model able to extract human part features. We learned 15 filters of 7×7 pixels, depicted in Fig. 4b, as the first layer. A continuous C-RBM is trained on contrast normalized 32×32 gray-scale image patches to obtain these filters. After a max pooling layer with 4×4 subsampling windows, we trained a binary C-RBM on top of lower level features extracted from full images. We learned 30 filter of size $15\times5\times5$ for the third layer, four of them illustrated in Fig. 1. The last max pooling window size was 2×2. Next, the fourth layer outputs were concatenated with HOG features into a feature vector. To combine the features we need to make the range of both types the same. Thus, since all features are bounded from below by zero, we only equalize their variance, and learn a linear SVM on them. Fig. 6 shows a significant improvement over HOG results when our features were concatenated.

In Table 3 we compare our results with two state-of-the-art approaches for pedestrian detection. our results are very close to Tuzel et al. [16]. The method proposed by Maji et al. [17] produces the best results. They proposed an efficient algorithm for learning histogram intersection kernel SVMs. This histogram intersection kernel is used on multi-scale HOG features to obtain these results on INRIA. However, this kernel can be used for other types of features e.g., our hierarchical filter responses, to improve the results.

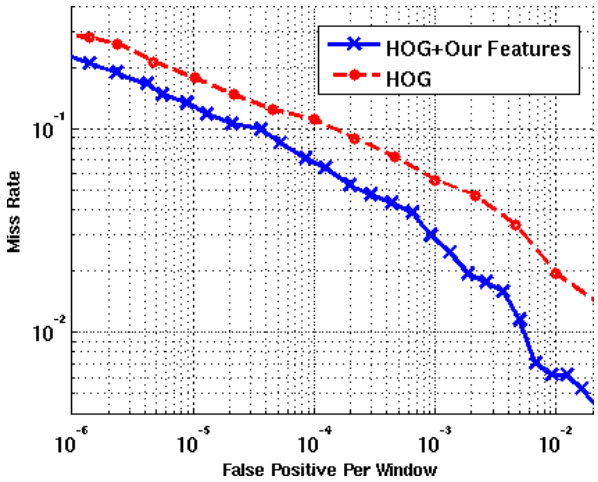


Figure 6. INRIA DET curves for HOG features and the combination of HOG and our learned features.

FPPW	HOG	HOG + Ours	Tuzel et al.	Maji et al.
10^{-4}	11.0%	6.6%	6.7%	2.5%
10^{-5}	17.8%	13.2%	11.4%	6.7%
10^{-6}	28.6%	22.7%	N/A	16.7%

Table 3. Miss rate on INRIA dataset as a function of FPPW.

6. Conclusion

In this paper, we have described an algorithm for learning features specific to an object class. The algorithm extends the Restricted Boltzmann Machine model by introducing weight sharing to define features that are replicated over spatial neighborhoods. By using this Convolutional Restricted Boltzmann Machine to model the distribution of a set of images, we learn a set of features which are tuned to represent a particular object class. These features are tested on the MNIST handwritten digits and INRIA pedestrian detection benchmark and obtain results comparable to state-of-the-art methods on both tasks.

References

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints", *Int. Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2005.

[3] P. Smolensky, *Parallel distributed processing: explorations in the microstructure of cognition*, vol. 1: foundations, chapter Information processing in dynamical systems: foundations of harmony theory, pp. 194–281, MIT Press, Cambridge, MA, USA, 1986.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, November 1998.

[5] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position", *Pattern Recognition*, vol. 15, no. 6, pp. 455–469, 1982.

[6] T. Serre, L. Wolf, S. Bileschi, and M. Riesenhuber, "Robust object recognition with cortex-like mechanisms", *IEEE Trans. PAMI*, vol. 29, no. 3, pp. 411–426, 2007, Member-Tomaso Poggio.

[7] J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2006.

[8] M. Ranzato, F. J. Huang, Y. Boureau, and Y. LeCun, "Unsupervised learning of invariant feature hierarchies with applications to object recognition", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2007.

[9] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model", in *Advances in Neural Info. Processing Systems*, J. Platt et al., Ed. 2006, MIT Press.

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

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

[12] S. Roth M. J. Black, "Fields of experts: A framework for learning image priors", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2005.

[13] M. Welling, G. Hinton, and S. Osindero, "Learning sparse topographic representations with products of student-t distributions", in *Advances in Neural Info. Processing Systems*, 2003, pp. 1383–1390.

[14] Y. Weiss and W. T. Freeman, "What makes a good model of natural images?", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2007, pp. 1–8.

[15] Y. Amit and A. Trouvé, "Pop: Patchwork of parts models for object recognition", *Int. Journal of Computer Vision*, vol. 75, no. 2, pp. 267–282, 2007.

[16] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds", *IEEE Trans. PAMI*, vol. 30, pp. 1713–1727, 2008.

[17] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient", in *Proc. IEEE Confrence Computer Vision and Pattern Recognition*, 2008.

[18] H. Chen and A. F. Murray, "Continuous restricted boltzmann machine with an implementable training algorithm", *IEEE Proc. Vision, Image and Signal Processing*, vol. 150, no. 3, pp. 153–159, 2003.

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

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

[21] Y. W. Teh, M. Welling, S. Osindero, and G. E. Hinton, "Energy-based models for sparse overcomplete representations", *The Journal of Machine Learning Research*, vol. 4, pp. 1235–1260, 2003.

[22] H. Lee, C. Ekanadham, and A. Ng, "Sparse deep belief net model for visual area v2", in *Advances in Neural Info. Processing Systems*, 2007, pp. 873–880.

[23] H. Larochelle and Y. Bengio, "Classification using discriminative restricted boltzmann machines", in *Proc. of the International Conference on Machine Learning*, 2008.

[24] F. Lauer, C. Y. Suen, and G. Bloch, "A trainable feature extractor for handwritten digit recognition", *Pattern Recognition*, vol. 40, no. 6, pp. 1816–1824, 2007.