Learning and Selecting Features Jointly with Point-wise Gated Boltzmann Machines

Kihyuk Sohn KIHYUKS@UMICH.EDU
Guanyu Zhou GUANYUZ@UMICH.EDU
Chansoo Lee CHANSOOL@UMICH.EDU
Honglak Lee HONGLAK@UMICH.EDU
Dept. of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109, USA

Abstract

Unsupervised feature learning has emerged as a promising tool in learning representations from unlabeled data. However, it is still challenging to learn useful high-level features when the data contains a significant amount of irrelevant patterns. Although feature selection can be used for such complex data, it may fail when we have to build a learning system *from scratch* (i.e., starting from the lack of useful raw features). To address this problem, we propose a *point-wise gated Boltzmann machine*, a unified generative model that combines feature learning and feature selection. Our model performs not only feature selection on learned high-level features (i.e., hidden units), but also *dynamic feature selection* on raw features (i.e., visible units) through a gating mechanism. For each example, the model can adaptively focus on a variable subset of visible nodes corresponding to the task-relevant patterns, while ignoring the visible units corresponding to the task-irrelevant patterns. In experiments, our method achieves improved performance over state-of-the-art in several visual recognition benchmarks.

1. Introduction

One fundamental difficulty in building algorithms that can robustly learn from complex real-world data is to deal with significant noise and irrelevant patterns. In particular, let's consider a problem of *learning from scratch*, assuming the lack of useful raw features. Here, the challenge is how to learn a robust representation that can distinguish important (e.g., task-relevant) patterns from significant amounts of distracting (e.g., task-irrelevant) patterns.

For constructing useful features, unsupervised feature learning (Hinton et al., 2006; Bengio et al., 2007; Ranzato et al., 2007; Bengio, 2009) has emerged as a powerful tool in learning representations from unlabeled data. In many real-world problems, however, the data is not cleaned up and contains significant amounts of irrelevant sensory patterns. In other words, *not all patterns are equally important*. In this case, the unsupervised learning methods may blindly represent the irrelevant patterns using the majority of the learned high-level features, and it becomes even more difficult to learn task-relevant higher-layer features (e.g., by stacking). Although there are ways to incorporate supervision (e.g., supervised fine-tuning), learning is still challenging when the data contains lots of irrelevant patterns, as shown in (Larochelle et al., 2007).

To deal with such complex data, one may envision using feature selection. Indeed, feature selection (Jain & Zongker, 1997; Yang & Pedersen, 1997; Weston et al., 2001; Guyon & Elisseeff, 2003) is an effective method for distinguishing useful raw features from irrelevant raw features. However, feature selection may fail if there are no good raw features to start with.

To address this issue, we propose to combine feature learning and feature selection coherently in a unified framework. Intuitively speaking, given that unsupervised feature learning can find partially useful high-level abstractions, it may be easier to apply feature selection on learned high-level features to distinguish the task-relevant ones from the task-irrelevant ones. Then, the task-relevant high-level features can be used to trace back where such important patterns occur. This information can help the learning algorithm to focus on these task-relevant raw features (i.e., visible units corresponding to task-relevant patterns), while

Proceedings of the 30^{th} International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

ignoring the rest.

In this paper, we formulate a generative feature learning algorithm called the *point-wise gated Boltzmann machine* (*PGBM*). Our model performs feature selection not only on learned high-level features (i.e., hidden units), but also on raw features (i.e., visible units) through a gating mechanism using stochastic "switch units." The switch units allow our model to estimate where the task-relevant patterns occur, and make only those visible units to contribute to the final prediction through multiplicative interaction. The model ignores the task-irrelevant portion of the raw features, thus it performs *dynamic feature selection* (i.e., choosing a variable subset of raw features depending on semantic interpretation of the individual example).

We evaluate our models in two ways: 1) recognizing handwritten digits in the irrelevant background, and 2) localizing and classifying objects in the natural scenes. In the first experiment, our method shows strong performance in learning features and distinguishing task-relevant features from task-irrelevant features. In the second experiment, our model shows promising results in distinguishing foreground objects from background scenes and localizing the object bounding boxes in a weakly-supervised way, which leads to an improved object recognition performance.

We summarize our main contributions as follows:

- We propose the PGBMs that jointly perform feature learning and feature selection in a unified framework.
- We propose the semi-supervised PGBM and show its effectiveness when given a small amount of labeled data and a large amount of unlabeled data.
- We show that the PGBM is an effective building block for constructing deep networks.
- We propose a convolutional extension of the PGBM. We further show that this can be used for weakly-supervised object localization. Using predicted bounding boxes of objects, we demonstrate state-of-the-art object recognition performance on the Caltech 101 dataset.
- We achieve a significant improvement over state-of-the-art on variations of the MNIST dataset.

2. Preliminaries

Our model can be viewed as a high-order extension of the restricted Boltzmann machine (RBM), and we briefly review it in this section. The RBM is an undirected graphical model that defines the distribution of visible units using binary hidden units. The joint distribution of binary visible units and binary hidden units is written as follows:

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

$$E(\mathbf{v},\mathbf{h}) = -\sum_{i=1}^{D}\sum_{k=1}^{K} v_i W_{ik} h_k - \sum_{k=1}^{K} b_k h_k - \sum_{i=1}^{D} c_i v_i$$

where $\mathbf{v} \in \{0,1\}^D$ are the visible (i.e., input) units, and $\mathbf{h} \in \{0,1\}^K$ are the hidden (i.e., latent) units. Z is the normalizing constant, and $\mathbf{W} \in \mathbb{R}^{D\times K}, \mathbf{b} \in \mathbb{R}^K, \mathbf{c} \in \mathbb{R}^D$ are the weight matrix, hidden and visible bias vectors, respectively. Since there are no connections between the units in the same layer, visible units are conditionally independent given the hidden units, and vice versa. The conditional probabilities of the RBM can be written as follows:

$$P(v_i = 1 \mid \mathbf{h}) = \sigma(\sum_k W_{ik} h_k + c_i),$$

$$P(h_k = 1 \mid \mathbf{v}) = \sigma(\sum_i W_{ik} v_i + b_k),$$

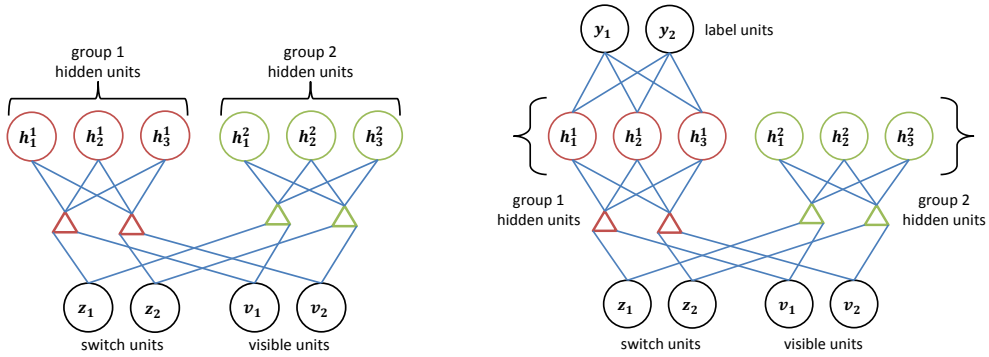
where $\sigma(x) = \frac{1}{1+\exp(-x)}$. Training the RBM corresponds to maximizing the log-likelihood of the data with respect to parameters $\{\mathbf{W}, \mathbf{b}, \mathbf{c}\}$. Although the gradient is intractable to compute, contrastive divergence (Hinton, 2002) can be used to approximate it.

3. Proposed Models

In this section, we propose the point-wise gated Boltzmann machine and its extensions. In Section 3.1, we describe the basic unsupervised PGBM that learns and groups features into semantically distinct components. In Section 3.2, we propose the supervised PGBM that uses class labels as a top-down feedback to partition the hidden units into the task-relevant and the task-irrelevant components. In Section 3.3, we propose the semi-supervised PGBM that uses unlabeled data as a regularizer when there are only a small number of labeled training examples. Furthermore, we construct a deep network using the PGBM as a building block, where we stack neural network layer(s) on top of the PGBM's task-relevant hidden units. Finally, we present the convolutional extension of the PGBM that can efficiently handle spatially correlated high-dimensional data.

3.1. Point-wise Gated Boltzmann machines

When we deal with complex data, it is desirable for a learning algorithm to distinguish semantically distinct patterns. For example, an object recognition algorithm may improve its performance if it can separate the foreground object patterns from the background clutters. To model this, we propose to represent each visible unit as a mixture model when conditioned on



(a) PGBM (b) supervised PGBM

Figure 1. Graphical model representation of the (a) PGBM and (b) supervised PGBM with two groups of hidden units. The Bernoulli switch unit z_i specifies which of the two components models the visible unit v_i. In other words, when $z_i = 1$, v_i is generated from the hidden units in the first group (shown in red); when $z_i = 2$, v_i is generated from the hidden units in the second group (shown in green).

the hidden units, where each group of hidden units can generate the corresponding mixture component.

Before going into details, we describe the generative process of the PGBM as follows: (1) The hidden units are partitioned into *components*, each of which defines a distinct distribution over the visible units. (2) Conditioning on the hidden units, we sample the *switch units*. (3) The switch units determine which *component* generates the corresponding visible units. A schematic diagram is shown in Figure 1(a) as an undirected graphical model.

The PGBM with R mixture components has a multinomial switch unit, denoted $z_i \in \{1, \cdots, R\}$,[1] for each visible unit v_i. The PGBM imposes element-wise multiplicative interaction between the paired switch and visible units, as shown in Figure 1(a). Now, we define the energy function of the PGBM as follows:

$$E^{\mathrm{U}}(\mathbf{v}, \mathbf{z}, \mathbf{h}) = -\sum_{r=1}^{R}\sum_{i=1}^{D}\sum_{k=1}^{K_r}(z_i^r v_i)\, W_{ik}^r h_k^r \qquad (1)$$
$$-\sum_{r=1}^{R}\sum_{k=1}^{K_r} b_k^r h_k^r - \sum_{r=1}^{R}\sum_{i=1}^{D}(z_i^r v_i)\, c_i^r,$$
$$\text{s.t.} \quad \sum_{r=1}^{R} z_i^r = 1,\ i = 1, \cdots, D.$$

Here, $\mathbf{v}$, $\mathbf{z}^r$ and $\mathbf{h}$ are the visible, switch and hidden unit binary vectors, respectively, and the model parameters W_{ik}^r, b_k^r, c_i^r are the weights, hidden biases, and the visible biases of r-th component. The binary-valued switch unit z_i^r is activated (i.e. takes value 1) if and only if its paired visible unit v_i is assigned to the r-th component, and its conditional probability given hidden units follows a multinomial distribution over R categories. The energy function can be written in matrix form as follows:

$$E^{\mathrm{U}}(\mathbf{v}, \mathbf{z}, \mathbf{h}) = -\sum_{r=1}^{R}(\mathbf{z}^r \odot \mathbf{v})^T \mathbf{W}^r \mathbf{h}^r$$
$$-\sum_{r=1}^{R}(\mathbf{b}^r)^T \mathbf{h}^r - \sum_{r=1}^{R}(\mathbf{c}^r)^T(\mathbf{z}^r \odot \mathbf{v}),$$

where the operator $\odot$ denotes element-wise multiplication, i.e., $(\mathbf{z}^r \odot \mathbf{v})_i = z_i^r v_i$.

The visible, hidden, and switch units are conditionally independent given the other two types of units, and the conditional probabilities can be written as follows:

$$P\left(h_k^r = 1 \mid \mathbf{z}, \mathbf{v}\right) = \sigma\left((\mathbf{z}^r \odot \mathbf{v})^T \mathbf{W}_{\cdot k}^r + b_k^r\right), \qquad (2)$$

$$P\left(v_i = 1 \mid \mathbf{z}, \mathbf{h}\right) = \sigma\left(\sum_r z_i^r\left(\mathbf{W}_{i\cdot}^r \mathbf{h}^r + c_i^r\right)\right), \qquad (3)$$

$$P\left(z_i^r = 1 \mid \mathbf{v}, \mathbf{h}\right) = \frac{\exp\left(v_i\left(\mathbf{W}_{i\cdot}^r \mathbf{h}^r + c_i^r\right)\right)}{\sum_s \exp\left(v_i\left(\mathbf{W}_{i\cdot}^s \mathbf{h}^s + c_i^s\right)\right)}, \qquad (4)$$

where we use $\mathbf{W}_{i\cdot}^r$ to denote i-th row, and $\mathbf{W}_{\cdot k}^r$ to denote k-th column of the matrix $\mathbf{W}^r$.

It is important to note that, while inferring the hidden units, our model *gates* (or re-weighs) each visible unit v_i according to the corresponding switch units z_i^r (Equation 2). In other words, the point-wise multiplicative interaction between the switch and the visible units allows the hidden units in each component to *focus* on a specific part of the data, and this makes the hidden units in one component to be robust to the patterns learned by other components. Moreover, the

[1] For convenience, we also use the vector representation for switch unit in boldface, i.e., $\mathbf{z}_i = [z_i^1, \cdots, z_i^R]^T \in \{0, 1\}^R$, where $\sum_{r=1}^R z_i^r = 1$, for each visible unit v_i.

top-down signal from the hidden units encourages assigning the same mixture component to semantically-related visible units during the switch unit inference, and therefore we can prune out the irrelevant raw features dynamically for each example.

It is worth noting that, when we tie all switch units (i.e., $\mathbf{z}_i = \mathbf{z}$ for all i), the PGBM becomes equivalent to the implicit mixture of restricted Boltzmann machine (Nair & Hinton, 2008). Furthermore, given that there is a multiplicative interaction between three types of variables, the PGBM can be understood in the context of higher-order Boltzmann machines (Sejnowski, 1987; Memisevic & Hinton, 2010).

We train the PGBM with stochastic gradient descent using contrastive divergence. Since the exact inference is intractable due to the three-way interaction, we use mean-field or alternating Gibbs sampling (i.e., sample one type of variables given the other two types using Equations (2),(3), and (4)) for approximate inference.

3.2. Generative feature selection with supervised PGBMs

Although the PGBM can learn to group distinct features for each mixture component, it doesn't necessarily learn discriminative features automatically since the generative training is done in an unsupervised way. One way to make the PGBM implicitly perform feature selection (i.e., distinguish features into different groups based on their relevance to the task) is to provide a good initialization of the model parameters. For example, we can pre-train the regular RBM and divide the hidden units into two groups based on the score from the simple feature selection algorithms such as the t-test[2] to initialize the weight matrices of the PGBM. As we will discuss in Section 5, this approach improves classification performance of the PGBMs.

Furthermore, to make use of class labels during the generative training, we propose a *supervised PGBM* that only connects the hidden units in the task-relevant component(s) to the label units. The graphical model representation is shown in Figure 1(b). By transferring the label information to the raw features through the task-relevant hidden units, the supervised PGBM can perform *generative feature selection* both at the high-level (i.e., using only a subset of hidden units for classification) and the low-level (e.g., dynamically blocking the influence of the task-irrelevant visible units) in a unified way.

For simplicity, we present the supervised PGBM with two mixture components, where we assign the first component to be task-relevant. The energy function is defined as follows:

$$E^{\mathrm{S}}(\mathbf{v}, \mathbf{z}, \mathbf{h}, \mathbf{y}) = E^{\mathrm{U}}(\mathbf{v}, \mathbf{z}, \mathbf{h}) - \mathbf{y}^T \mathbf{U} \mathbf{h}^1 - \mathbf{d}^T \mathbf{y} \quad (5)$$

subject to $z_i^1 + z_i^2 = 1, \ i = 1, \cdots, D$. The label vector $\mathbf{y} \in \{0, 1\}^L$ is in the 1-of-L representation. $\mathbf{U} \in \mathbb{R}^{L \times K_1}$ is the weight matrix between the task-relevant hidden units and the label units, and $\mathbf{d}$ is the label bias vector. The conditional probabilities can be written as follows:

$$P\left(h_k^1 = 1 \mid \mathbf{z}, \mathbf{v}, \mathbf{y}\right) = \sigma\left((\mathbf{z}^1 \odot \mathbf{v})^T \mathbf{W}_{\cdot k}^1 + b_k^1 + \mathbf{U}_{\cdot k}^T \mathbf{y}\right), \quad (6)$$

$$P\left(y_l = 1 \mid \mathbf{h}^1\right) = \frac{\exp\left(\mathbf{U}_{l\cdot} \mathbf{h}^1 + d_l\right)}{\sum_s \exp\left(\mathbf{U}_{s\cdot} \mathbf{h}^1 + d_s\right)}. \quad (7)$$

The conditional probabilities of the visible and switch units are the same as Equations (3) and (4). As we can see in Equation (6), the label information, together with the switch units, modulates the hidden unit activations in the first (task-relevant) component, and this in turn encourages the switch units z_i^1 to activate at the task-relevant visible units[3] during the iterative approximate inference.

We can train the supervised PGBM in generative criteria whose objective is to maximize the joint log-likelihood of the visible and the label units (Larochelle & Bengio, 2008). Similarly to that of PGBM, the inference can be done with alternating Gibbs sampling between Equations (3),(4),(6), and (7).

3.3. Variations of the model

3.3.1. Semi-supervised PGBMs

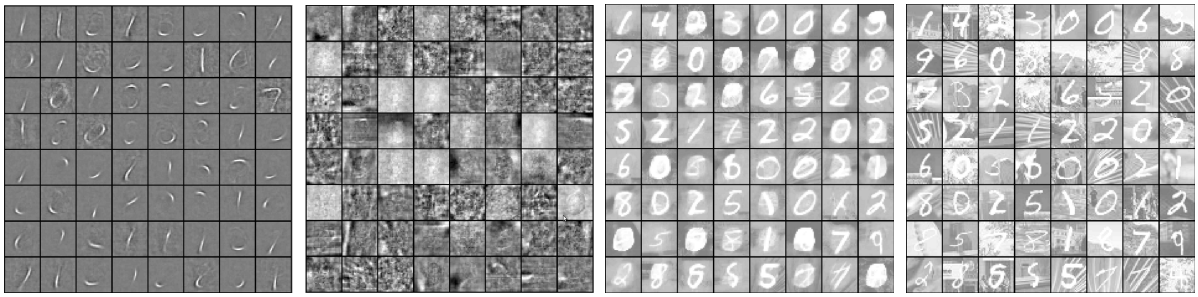
There are many classification tasks where we are given a large number of unlabeled examples in addition to only a small number of labeled training examples. For this scenario, it is important to include unlabeled examples during the training to generalize well to the unseen data. The supervised PGBM can be adapted to the semi-supervised learning framework. For example, we can regularize the joint log-likelihood $\log P^{\mathrm{S}}(\mathbf{v}, \mathbf{y})$ with the data log-likelihood $\log P^{\mathrm{S}}(\mathbf{v})$ defined on the unlabeled data (Larochelle & Bengio, 2008). We provide more details in Section 5 and the supplementary material.

3.3.2. Deep networks

The PGBM can be used as a building block of deep networks. For example, we can use it as a first layer

[2] http://featureselection.asu.edu/software.php

[3] Note: In our model, we call that a visible unit (a raw feature) is "task-relevant" (or "task-irrelevant") if its switch unit for the task-relevant (or task-irrelevant) component is active, respectively.



(a) task-relevant filters (b) task-irrelevant filters (c) switch unit activation (d) original images

Figure 2. (a, b) Visualization of filters corresponding to two components learned from the PGBM, (c) visualization of the activation of switch units, and (d) corresponding original images on *mnist-back-image* dataset. Specifically, (a) represents the group of hidden units that activates for the foreground digits (task-relevant), and (b) represents the group of hidden units that activates for the background images (task-irrelevant). See text for details.

block and stack neural networks on the hidden units of task-relevant components. Since the PGBM can select the task-relevant hidden units with supervision, the higher-layer networks can focus on the task-relevant information. In Section 5.1, we show that the two-layer model, where we stack a single-layer neural network on top of a PGBM's task-relevant component, was sufficient to outperform existing state-of-the-art classification performance on the variations of MNIST dataset with irrelevant backgrounds.

3.3.3. Convolutional PGBM

Convolutional models can be useful in representing spatially or temporally correlated data. The PGBM can be extended to a convolutional setting (Lee et al., 2011), where we share the filter weights over different locations in large images. In Section 5.2, we present the convolutional PGBM with an application to the weakly supervised foreground object localization problem. Furthermore, by locating the bounding box at the foreground object accurately, we achieved state-of-the-art recognition performance in Caltech 101. For more details, see the supplementary material.

4. Related Work

As mentioned in Section 3.1, the PGBM can be viewed as an extension of the implicit mixture of RBM (imRBM) (Nair & Hinton, 2008) that allows per-visible-unit switching. Although these two models look similar, the *per-visible-unit switching* property of the PGBM brings an important benefit over the imRBM because it allows the PGBM to represent data with multiple components, each of which focusing on different part of the raw features. In particular, the supervised PGBM represents the data with two groups of hidden units (one containing task-relevant hidden units and the other containing task-irrelevant hidden units). In contrast, the imRBM uses a single component to represent the data, and thus cannot distinguish between the relevant and irrelevant patterns when the data contains significant amounts of irrelevant patterns.

The discriminative RBM (discRBM) (Larochelle & Bengio, 2008) is another model that can learn discriminative features using class labels. We argue that, however, the PGBM can be more robust to noisy data since it can prune out (or re-weigh) the irrelevant features dynamically for each data instance using switch unit activations, whereas the discRBM accumulates the contribution from noisy visible units with the fixed weights applied to all data instances. In Section 5.1, we empirically show that the PGBM significantly outperforms both the imRBM and the discRBM in classifying handwritten digits in the presence of irrelevant background patterns.

Rifai et al. (2012) proposed the contractive discriminative analysis (CDA). Similarly to the PGBM, the CDA has two groups of hidden units, one of which is connected to labels. The difference is that the CDA is a feed-forward neural network which can learn distinct features for each group with a contractive penalty term, while the PGBM is a probabilistic model that performs *generative* feature selection through a multiplicative interaction between visible, hidden, and switch units.

The robust Boltzmann machine (RoBM) (Tang et al., 2012) shares its motivation with our work, though there are several technical differences. First, the RoBM models each background noise unit with a unimodal Gaussian distribution, whereas the PGBM models the background visible units with more complicated multimodal distribution with a group of hidden units. Furthermore, the PGBM can directly learn from the noisy data with class labels, but the RoBM requires clean data to pre-train the GRBM.

In terms of energy function, the unsupervised PGBM

Table 1. Test classification errors of (top) single-layer and (bottom) multi-layer models on MNIST variation datasets. We used 10,000/2,000/50,000 splits for train, validation and test sets, and report the test classification errors without retraining the model after hyperparameter search over the validation set. For all RBM variants including imRBM, discRBM, and PGBM, we used sparsity regularizer (Lee et al., 2008). The best performers among the single-layer models and the deep network models are both in bold.

Algorithm	*mnist-back-rand*	*mnist-back-image*	*mnist-rot-back-image*	*mnist-rot-back-rand*
RBM	11.39	15.42	49.89	51.97
imRBM	10.46	16.35	51.03	51.02
discRBM	10.29	15.56	48.34	48.28
RBM-FS	11.42	15.20	49.65	51.69
PGBM	7.27	13.33	45.45	45.53
supervised PGBM	**6.87**	**12.85**	**44.67**	**43.47**
DBN-3 (Vincent et al., 2008)	6.73	16.31	47.39	-
CAE-2 (Rifai et al., 2011)	10.90	15.50	45.23	-
PGBM+ DN-1	**6.08**	**12.25**	**36.76**	**30.41**

can be viewed as having a similar formulation to the masked RBM (Le Roux et al., 2011; Heess et al., 2011). However, our main motivation is to perform joint feature selection at both low-level and high-level. Specifically, the difference becomes clearer when we use class labels in supervised PGBM that performs generative feature selection, as discussed in Section 3.2.

5. Experiments

5.1. Recognizing handwritten digits in the presence of irrelevant background noise

We evaluated the capability of the proposed models in learning task-relevant features from noisy data. We tested the single-layer PGBMs and their extensions on the variations of MNIST dataset: *mnist-back-rand*, *mnist-back-image*, *mnist-rot-back-image*, and *mnist-rot-back-rand*.[4] The first two datasets use uniform noise or natural images as background patterns. The other two have rotated digits in front of the corresponding background patterns. We used the PGBM with two components of 500 hidden units, and initialized with the pre-trained RBM using the feature selection as described in Section 3.2. We used mean-field for approximate inference for all our experiments.[5]

In Figure 2, we visualize the filters and the switch unit activations for *mnist-back-image*. The foreground filters capture the task-relevant patterns resembling pen strokes (Figure 2(a)), while the background filters (Figure 2(b)) capture task-irrelevant patterns in the natural images. Further, the switch unit activations (the posterior probabilities that the input pixel belongs to the *foreground component*, Figure 2(c)) are high (colored in white) for the foreground digit pixels, and low (colored in gray) for the background pixels. This suggests that our model can *dynamically* separate the task-relevant raw features from the task-irrelevant raw features for each example.

For quantitative evaluation, we show test classification errors in Table 1. For all experiments with our single-layer models, we used the "task-relevant" hidden unit activations as the input for the linear SVM (Fan et al., 2008). The single-layer PGBM significantly outperformed the baseline RBM, imRBM, and discRBM.[6] We did a careful model selection to choose the best hyperparameters for each of the compared models. These results suggest that the point-wise mixture hypothesis is effective in learning task-relevant features from complex data containing irrelevant patterns.

5.1.1. Generative feature selection

As a control experiment, we compared our model to the two-step model which we call "RBM-FS," where we first trained the RBM and selected a subset of hidden units using feature selection. As we see in Table 1, the RBM-FS is only marginally better (or sometimes worse) than the baseline RBM. However, the PGBM significantly outperforms the RBM-FS, which demonstrates the benefit of the joint training.

5.1.2. Semi-supervised learning

The supervised PGBM can be trained in a semi-supervised way as described in Section 3.3.1. We used the same experimental setting as (Larochelle & Bengio, 2008), and provided labels for only 10 percent of training examples (100 labeled examples for each digit category). We summarize the classification errors of semi-supervised PGBM, supervised PGBM, RBM and

[4]The first three datasets are generated by Larochelle et al. (2007). We generated *mnist-rot-back-rand* following the procedure described in their paper.

[5]We have tested mean-field and alternating Gibbs sampling with 10-25 iterations, and they showed similar results.

[6]We used "hybrid" discriminative RBM whose objective is a weighted sum of the discriminative (conditional) and generative (joint) likelihood.

Table 2. The mean and the standard deviation of the test classification errors of semi-supervised PGBM, supervised PGBM, RBM, and RBM-FS. We repeated 5 times with randomly sampled 1,000 labeled training examples in addition to the remaining 9,000 unlabeled training examples. The best model and those within the standard deviation are in bold.

Algorithm	*mnist-back-rand*	*mnist-back-image*	*mnist-rot-back-image*	*mnist-rot-back-rand*
RBM	17.43 ± 0.36	23.71 ± 0.34	63.94 ± 0.50	63.17 ± 0.32
RBM-FS	17.15 ± 0.46	**20.22 ± 0.31**	61.76 ± 0.43	62.02 ± 0.81
supervised PGBM	16.15 ± 0.70	21.04 ± 0.18	**59.39 ± 0.58**	63.82 ± 0.68
semi-supervised PGBM	**11.98 ± 0.80**	**20.32 ± 0.15**	**59.19 ± 0.68**	**58.57 ± 0.49**

RBM-FS in Table 2. The semi-supervised PGBM consistently performed the best for all datasets, showing that semi-supervised training is effective in utilizing a large number of unlabeled examples.

5.1.3. Deep networks

Finally, we constructed a two-layer deep network by stacking one layer of neural network with 1,000 hidden units on the task-relevant component of the PGBM. We used softmax classifier for fine-tuning of the second layer neural network. Table 1 shows that our deep network (referred to as "PGBM+DN-1") outperforms the DBN-3 and the stacked contractive autoencoder by a large margin. In particular, the result of the DBN-3 on *mnist-back-image* implies that adding more layers to the DBN does not necessarily improve the performance when there are significant amounts of irrelevant patterns in the data. In contrast, the PGBM can block the task-irrelevant information from propagating to the higher layers, and hence it is an effective building block for deep networks. Finally, we note that, to the best of our knowledge, the PGBM+DN-1 achieved state-of-the-art classification performance on all datasets except *mnist-rot-back-image*, where the transformation-invariant RBM (Sohn & Lee, 2012) achieved 35.5% error by incorporating the rotational invariance.

5.2. Weakly supervised object segmentation with an application to object recognition

In this section, we extend our model to learn groups of task-relevant features (i.e., foreground patterns) from the images with higher resolution, and apply it to weakly supervised object segmentation.

5.2.1. Weakly supervised object segmentation

Lee et al. (2011) showed that the convolutional deep belief network (CDBN) composed of multiple layers of convolutional RBM (CRBM) can learn hierarchical feature representations from large images. In particular, the first layer of the CDBN mostly learns generic edge filters, and the higher layers learn not only complex generic patterns, such as corners or contours, but also semantically meaningful features, such as object parts (e.g., eyes, nose, or wheels) in the second layer or whole objects (e.g., human face or car) in the third layer. To learn and group related features from large images, we propose the point-wise gated convolutional deep network (CPGDN), where we use the convolutional extension of the PGBM (CPGBM) as a building block.

Specifically, we construct the two-layer CPGDN by stacking the CPGBM on the first layer CRBM. This construction makes sense because the first layer features are mostly generic, and the class-specific features emerge in higher layers (Lee et al., 2011). We train the CPGDN using greedy layer-wise training method, and perform feedforward inference in the first layer. We use mean-field in the second layer for approximate inference of switch and hidden units. Due to the space constraint, we put more technical details of the CPGDN in the supplementary material.

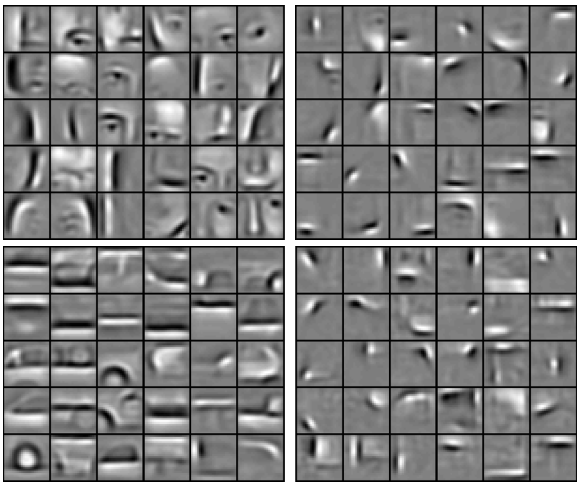


Figure 3. Visualization of the second layer CPGBM features from "Faces" (top row) and "Car side" (bottom row) classes. The left column shows the filters in the task-relevant components, and the right column shows the filters in task-irrelevant components.

We first trained a CPGDN with two mixture components only on the single class of images from Caltech 101 dataset (Fei-Fei et al., 2004). For this experiment, we randomly initialized the weights without pre-training. We visualize the second layer features trained on "Faces" and "Car side" classes in Figure 3. The CPGDN made a good distinction between the task-relevant patterns such as face parts and wheels, and the generic patterns. In Figure 4, we visualize

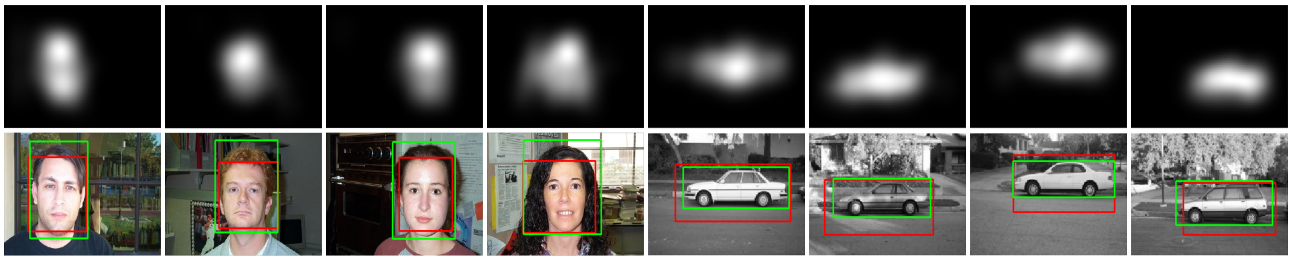


Figure 4. Visualization of (top) the switch unit activation map and (bottom) the images overlayed with the predicted (red) and the ground truth bounding boxes (green).

the switch unit activation map, which shows that the switch units are selectively activated at the most informative region in each image. Interestingly, using this activation map, we can segment the object region from the background reasonably well, though our model is not specifically designed for image segmentation.

5.2.2. Object recognition

Inspired by the CPGDN's ability to distinguish the foreground object from the background scene, we propose a novel object recognition pipeline on Caltech 101 dataset, where we first "crop" each image at the bounding box predicted using the switch unit activations of the CPGDN and perform classification using those cropped images. Specifically, we used the CPGDN with two mixture components, each of which is composed of 100 hidden units. To train the model efficiently from many different classes of images, we pretrain a set of second layer CRBMs with a small number of hidden units (e.g., 30) for each class to capture more diverse and class-specific patterns, and perform feature selection on those CRBM features from all object categories to initialize the weights of the second layer CPGBM. Once we train the model, we compute the posterior of switch units arranged in 2d. To predict the bounding box, we compute the row-wise and column-wise cumulative sum of switch unit activations and select the region containing (5,95) percentiles of the total activations as a bounding box. For classification, we followed the pipeline used in (Sohn et al., 2011), which uses the Gaussian (convolutional) RBMs with dense SIFT as input.

We first evaluated the bounding box detection accuracy. We declare that the bounding box prediction is correct when the average overlap ratio (the area of intersection divided by the union between the predicted and the ground truth bounding boxes) is greater than 0.5 (Everingham et al., 2010). We achieved average overlap ratio of 0.702 and detection accuracy of 88.3%.

Finally, we evaluated the classification accuracy using the cropped Caltech 101 dataset with CPGDN and summarize the results in Table 3. The object centered

Table 3. Test classification accuracy on Caltech 101.

Training images per class	15	30
Lazebnik et al. (2006)	56.4%	64.6%
Griffin et al. (2007)	59.0%	67.6%
Yang et al. (2009)	67.0%	73.2%
Boureau et al. (2010)	-	75.7%
Goh et al. (2012)	71.1%	**78.9**%
RBM (Sohn et al., 2011)	68.6%	74.9%
Our method + RBM	70.2%	76.8%
CRBM (Sohn et al., 2011)	71.3%	77.8%
Our method + CRBM	**72.4**%	**78.9**%

cropped images brought improvement in classification accuracies, such as 74.9% to 76.8% with RBM, and 77.8% to 78.9% with CRBM using 30 training images per class, respectively.[7] As a baseline, we also report the classification accuracy on the augmented dataset where we uniformly crop the center region across all the images with a fixed ratio. After cross-validating with different ratios, we obtain a worse classification accuracy of 75.8% with RBM using 30 training images per class. This suggests that the classification performance can be improved by localizing the object better than simply cropping the center region.

6. Conclusion

In this paper, we proposed a point-wise gated Boltzmann machine that can effectively learn useful feature representations from data containing irrelevant patterns. Our methods achieve state-of-the-art classification performance on several datasets that contain irrelevant patterns. We believe our method holds promise in building a robust algorithm that can learn from large-scale, complex, sensory input data.

Acknowledgments

This work was supported in part by NSF IIS 1247414 and a Google Faculty Research Award.

[7] We also performed the same experiment using different CPGDN model without pre-training. We obtained similar accuracy for the bounding box detection (0.697 for average overlap ratio, 90.2% for detection accuracy), but got slightly worse classification accuracy (76.4% with RBM using 30 training images per class).

References

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

Bengio, Y. Learning deep architectures for AI. *Foundations and Trends in Machine Learning*, 2(1):1–127, 2009.

Boureau, Y. L., Bach, F., LeCun, Y., and Ponce, J. Learning mid-level features for recognition. In *CVPR*, 2010.

Everingham, M., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88 (2):303–338, 2010.

Fan, R. E., Chang, K. W., Hsieh, C. J., Wang, X. R., and Lin, C. J. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9: 1871–1874, 2008.

Fei-Fei, L., Fergus, R., and Perona, P. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In *CVPR Workshop on Generative Model Based Vision*, 2004.

Goh, H., Thome, N., Cord, M., and Lim, J. H. Unsupervised and supervised visual codes with restricted Boltzmann machines. In *ECCV*, 2012.

Griffin, G., Holub, A., and Perona, P. Caltech-256 object category dataset. Technical report, California Institute of Technology, 2007.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Heess, N., Le Roux, N., and Winn, J. Weakly supervised learning of foreground-background segmentation using masked RBMs. In *ICANN*, 2011.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8): 1771–1800, 2002.

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

Jain, A. and Zongker, D. Feature selection: Evaluation, application, and small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, 1997.

Larochelle, H. and Bengio, Y. Classification using discriminative restricted Boltzmann machines. In *ICML*, 2008.

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. An empirical evaluation of deep architectures on problems with many factors of variation. In *ICML*, 2007.

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

Le Roux, N., Heess, N., Shotton, J., and Winn, J. Learning a generative model of images by factoring appearance and shape. *Neural Computation*, 23(3):593–650, 2011.

Lee, H., Ekanadham, C., and Ng, A. Y. Sparse deep belief network model for visual area V2. In *NIPS*, 2008.

Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. Unsupervised learning of hierarchical representations with convolutional deep belief networks. *Communications of the ACM*, 54(10):95–103, 2011.

Memisevic, R. and Hinton, G. E. Learning to represent spatial transformations with factored higher-order Boltzmann machines. *Neural Computation*, 22(6):1473–1492, 2010.

Nair, V. and Hinton, G. E. Implicit mixtures of restricted Boltzmann machines. In *NIPS*, 2008.

Ranzato, M., Poultney, C., Chopra, S., and LeCun, Y. Efficient learning of sparse representations with an energy-based model. In *NIPS*, 2007.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: Explicit invariance during feature extraction. In *ICML*, 2011.

Rifai, S., Bengio, Y., Courville, A., Vincent, P., and Mirza, M. Disentangling factors of variation for facial expression recognition. In *ECCV*, 2012.

Sejnowski, T. J. Higher-order Boltzmann machines. In *AIP Conference Proceedings on Neural Networks for Computing*, 1987.

Sohn, K. and Lee, H. Learning invariant representations with local transformations. In *ICML*, 2012.

Sohn, K., Jung, D. Y., Lee, H., and Hero, A. O. Efficient learning of sparse, distributed, convolutional feature representations for object recognition. In *ICCV*, 2011.

Tang, Y., Salakhutdinov, R., and Hinton, G. E. Robust Boltzmann machines for recognition and denoising. In *CVPR*, 2012.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P. A. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. Feature selection for svms. In *NIPS*, 2001.

Yang, J., Yu, K., Gong, Y., and Huang, T. S. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009.

Yang, Y. and Pedersen, J. O. A comparative study on feature selection in text categorization. In *ICML*, 1997.