Classification using Discriminative Restricted Boltzmann Machines

Hugo Larochelle LAROCHEH@IRO.UMONTREAL.CA
Yoshua Bengio BENGIOY@IRO.UMONTREAL.CA
Dept. IRO, Université de Montréal C.P. 6128, Montreal, Qc, H3C 3J7, Canada

Abstract

Recently, many applications for Restricted Boltzmann Machines (RBMs) have been developed for a large variety of learning problems. However, RBMs are usually used as feature extractors for another learning algorithm or to provide a good initialization for deep feed-forward neural network classifiers, and are not considered as a standalone solution to classification problems. In this paper, we argue that RBMs provide a self-contained framework for deriving competitive non-linear classifiers. We present an evaluation of different learning algorithms for RBMs which aim at introducing a discriminative component to RBM training and improve their performance as classifiers. This approach is simple in that RBMs are used directly to build a classifier, rather than as a stepping stone. Finally, we demonstrate how discriminative RBMs can also be successfully employed in a semi-supervised setting.

1. Introduction

Restricted Boltzmann Machines (RBMs) (Smolensky, 1986) are generative models based on latent (usually binary) variables to model an input distribution, and have seen their applicability grow to a large variety of problems and settings in the past few years. From binary inputs, they have been extended to model various types of input distributions (Welling et al., 2005; Hinton et al., 2006). Conditional versions of RBMs have also been developed for collaborative filtering (Salakhutdinov et al., 2007) and to model motion capture data (Taylor et al., 2006) and video sequences (Sutskever & Hinton, 2007).

RBMs have been particularly successful in classification problems either as feature extractors for text and image data (Gehler et al., 2006) or as a good initial training phase for deep neural network classifiers (Hinton, 2007). However, in both cases, the RBMs are merely the first step of another learning algorithm, either providing a preprocessing of the data or an initialization for the parameters of a neural network. When trained in an unsupervised fashion, RBMs provide no guarantees that the features implemented by their hidden layer will ultimately be useful for the supervised task that needs to be solved. More practically, model selection can also become problematic, as we need to explore jointly the space of hyper-parameters of both the RBM (size of the hidden layer, learning rate, number of training iterations) and the supervised learning algorithm that is fed the learned features. In particular, having two separate learning phases (feature extraction, followed by classifier training) can be problematic in an online learning setting.

In this paper, we argue that RBMs can be used successfully as stand-alone non-linear classifiers alongside other standard classifiers like neural networks and Support Vector Machines, and not only as feature extractors. We investigate training objectives for RBMs that are more appropriate for training classifiers than the common generative objective. We describe Discriminative Restricted Boltzmann Machines (DRBMs), i.e. RBMs that are trained more specifically to be good classification models, and Hybrid Discriminative Restricted Boltzmann Machines (HDRBMs) which explore the space between discriminative and generative learning and can combine their advantages. We also demonstrate that RBMs can be successfully adapted to the common semi-supervised learning setting (Chapelle et al., 2006) for classification problems. Finally, the algorithms investigated in this paper are well suited for online learning on large datasets.

2. Restricted Boltzmann Machines

Restricted Boltzmann Machines are undirected generative models that use a layer of hidden variables to model a distribution over visible variables. Though they are most often trained to only model the inputs

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008.

of a classification task, they can also model the joint distribution of the inputs and associated target classes (e.g. in the last layer of a Deep Belief Network in Hinton et al. (2006)). In this section, we will focus on such joint models.

We assume given a training set $\mathcal{D}_{train} = \{(\mathbf{x}_i, y_i)\}$, comprising for the i-th example an input vector $\mathbf{x}_i$ and a target class $y_i \in \{1, \ldots, C\}$. To train a generative model on such data we consider minimization of the negative log-likelihood

$$\mathcal{L}_{gen}(\mathcal{D}_{train}) = -\sum_{i=1}^{|\mathcal{D}_{train}|} \log p(y_i, \mathbf{x}_i). \qquad (1)$$

An RBM with n hidden units is a parametric model of the joint distribution between a layer of hidden variables (referred to as neurons or features) $\mathbf{h} = (h_1, \ldots, h_n)$ and the observed variables made of $\mathbf{x} = (x_1, \ldots, x_d)$ and y, that takes the form

$$p(y, \mathbf{x}, \mathbf{h}) \quad \propto \quad e^{-E(y,\mathbf{x},\mathbf{h})}$$

where

$$E(y, \mathbf{x}, \mathbf{h}) \quad = \quad -\mathbf{h}^T\mathbf{W}\mathbf{x} - \mathbf{b}^T\mathbf{x} - \mathbf{c}^T\mathbf{h} - \mathbf{d}^T\vec{y} - \mathbf{h}^T\mathbf{U}\vec{y}$$

with parameters $\Theta = (\mathbf{W}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{U})$ and $\vec{y} = (1_{y=i})_{i=1}^{C}$ for C classes. This model is illustrated in Figure 2. For now, we consider for simplicity binary input variables, but the model can be easily generalized to non-binary categories, integer-valued, and continuous-valued inputs (Welling et al., 2005; Hinton et al., 2006). It is straightforward to show that

$$\begin{aligned} p(\mathbf{x}|\mathbf{h}) &= \prod_i p(x_i|\mathbf{h}) \\ p(x_i = 1|\mathbf{h}) &= \mathrm{sigm}(b_i + \sum_j W_{ji} h_j) \qquad (2) \\ p(y|\mathbf{h}) &= \frac{e^{d_y + \sum_j U_{jy} h_j}}{\sum_{y^*} e^{d_{y^*} + \sum_j U_{jy^*} h_j}} \qquad (3) \end{aligned}$$

where sigm is the logistic sigmoid. Equations 2 and 3 illustrate that the hidden units are meant to capture predictive information about the input vector as well as the target class. $p(\mathbf{h}|y, \mathbf{x})$ also has a similar form:

$$\begin{aligned} p(\mathbf{h}|y, \mathbf{x}) &= \prod_j p(h_j|y, \mathbf{x}) \\ p(h_j = 1|y, \mathbf{x}) &= \mathrm{sigm}(c_j + U_{jy} + \sum_i W_{ji} x_i). \end{aligned}$$

When the number of hidden variables is fixed, an RBM can be considered a parametric model, but when it is allowed to vary with the data, it becomes a non-parametric model. In particular, Freund and Haussler (1994); Le Roux and Bengio (2008) showed that an RBM with enough hidden units can represent any distribution over binary vectors, and that adding hidden units guarantees that a better likelihood can be achieved, unless the generated distribution already equals the training distribution.

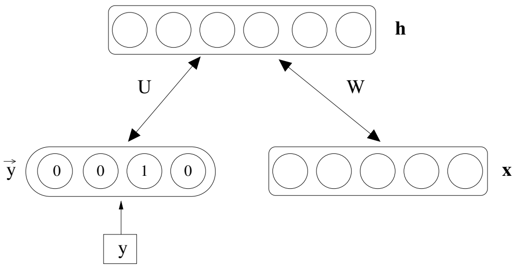


Figure 1. Restricted Boltzmann Machine modeling the joint distribution of inputs and target classes

In order to minimize the negative log-likelihood (eq. 1), we would like an estimator of its gradient with respect to the model parameters. The exact gradient, for any parameter $\theta \in \Theta$ can be written as follows:

$$\begin{aligned} \frac{\partial \log p(y_i, \mathbf{x}_i)}{\partial \theta} &= -\mathbb{E}_{\mathbf{h}|y_i,\mathbf{x}_i}\left[\frac{\partial}{\partial \theta} E(y_i, \mathbf{x}_i, \mathbf{h})\right] \\ &\quad + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\left[\frac{\partial}{\partial \theta} E(y, \mathbf{x}, \mathbf{h})\right]. \end{aligned}$$

Though the first expectation is tractable, the second one is not. Fortunately, there exists a good stochastic approximation of this gradient, called the contrastive divergence gradient (Hinton, 2002). This approximation replaces the expectation by a sample generated after a limited number of Gibbs sampling iterations, with the sampler's initial state for the visible variables initialized at the training sample $(y_i, \mathbf{x}_i)$. Even when using only one Gibbs sampling iteration, contrastive divergence has been shown to produce only a small bias for a large speed-up in training time (Carreira-Perpiñan & Hinton, 2005).

Online training of an RBM thus consists in cycling through the training examples and updating the RBM's parameters according to Algorithm 1, where the learning rate is controlled by λ.

Computing $p(y, \mathbf{x})$ is intractable, but it is possible to compute $p(y|\mathbf{x})$, sample from it, or choose the most probable class under this model. As shown in Salakhutdinov et al. (2007), for reasonable numbers of classes C (over which we must sum), this conditional distribution can be computed exactly and efficiently, by writing it as follows:

$$p(y|\mathbf{x}) \quad = \quad \frac{e^{d_y} \prod_{j=1}^{n} \left(1 + e^{c_j + U_{jy} + \sum_i W_{ji} x_i}\right)}{\sum_{y^*} e^{d_{y^*}} \prod_{j=1}^{n} \left(1 + e^{c_j + U_{jy^*} + \sum_i W_{ji} x_i}\right)}.$$

Algorithm 1 Training update for RBM over $(y, \mathbf{x})$ using Contrastive Divergence

```
Input: training pair $(y_i, \mathbf{x}_i)$ and learning rate $\lambda$
% Notation: $a \leftarrow b$ means $a$ is set to value $b$
%           $a \sim p$ means $a$ is sampled from $p$

% Positive phase
$y^0 \leftarrow y_i$, $\mathbf{x}^0 \leftarrow \mathbf{x}_i$, $\widehat{\mathbf{h}}^0 \leftarrow \mathrm{sigm}(\mathbf{c} + W\mathbf{x}^0 + \mathbf{U}\vec{y^0})$

% Negative phase
$\mathbf{h}^0 \sim p(\mathbf{h}|y^0, \mathbf{x}^0)$, $y^1 \sim p(y|\mathbf{h}^0)$, $\mathbf{x}^1 \sim p(\mathbf{x}|\mathbf{h}^0)$
$\widehat{\mathbf{h}}^1 \leftarrow \mathrm{sigm}(\mathbf{c} + W\mathbf{x}^1 + \mathbf{U}\vec{y^1})$

% Update
for $\theta \in \Theta$ do
  $\theta \leftarrow \theta - \lambda \left( \frac{\partial}{\partial \theta} E(y^0, \mathbf{x}^0, \widehat{\mathbf{h}}^0) - \frac{\partial}{\partial \theta} E(y^1, \mathbf{x}^1, \widehat{\mathbf{h}}^1) \right)$
end for
```

Precomputing the terms $c_j + \sum_i W_{ji} x_i$ and reusing them when computing $\prod_{j=1}^{n} \left(1 + e^{c_j + U_{jy^*} + \sum_i W_{ji} x_i}\right)$ for all classes y^* permits to compute this conditional distribution in time $O(nd + nC)$.

3. Discriminative Restricted Boltzmann Machines

In a classification setting, one is ultimately only interested in correct classification, not necessarily to have a good $p(\mathbf{x})$. Because our model's $p(\mathbf{x})$ can be inappropriate, it can then be advantageous to optimize directly $p(y|\mathbf{x})$ instead of $p(y, \mathbf{x})$:

$$\mathcal{L}_{disc}(\mathcal{D}_{train}) = - \sum_{i=1}^{|\mathcal{D}_{train}|} \log p(y_i|\mathbf{x}_i). \qquad (4)$$

We refer to RBMs trained according to $\mathcal{L}_{disc}$ as Discriminative RBMs (DRBMs). Since RBMs (with enough hidden units) are universal approximators for binary inputs, it follows also that DRBMs are universal approximators of conditional distributions with binary inputs.

A DRBM can be trained by contrastive divergence, as has been done in conditional RBMs (Taylor et al., 2006), but since $p(y|\mathbf{x})$ can be computed exactly, we can compute the exact gradient:

$$\begin{aligned} \frac{\partial \log p(y_i|\mathbf{x}_i)}{\partial \theta} &= \sum_j \mathrm{sigm}(o_{yj}(\mathbf{x}_i)) \frac{\partial o_{yj}(\mathbf{x}_i)}{\partial \theta} \\ &\quad - \sum_{j, y^*} \mathrm{sigm}(o_{y^*j}(\mathbf{x}_i)) p(y^*|\mathbf{x}_i) \frac{\partial o_{y^*j}(\mathbf{x}_i)}{\partial \theta} \end{aligned}$$

where $o_{yj}(\mathbf{x}) = c_j + \sum_k W_{jk} x_k + U_{jy}$. This gradient can be computed efficiently and then used in a stochastic gradient descent optimization. This discriminative approach has been used previously for fine-tuning the top RBM of a Deep Belief Network (Hinton, 2007).

4. Hybrid Discriminative Restricted Boltzmann Machines

The advantage brought by discriminative training usually depends on the amount of available training data. Smaller training sets tend to favor generative learning and bigger ones favor discriminative learning (Ng & Jordan, 2001). However, instead of solely relying on one or the other perspective, one can adopt a hybrid discriminative/generative approach simply by combining the respective training criteria. Though this method cannot be interpreted as a maximum likelihood approach for a particular generative model as in Lasserre et al. (2006), it proved useful here and elsewhere (Bouchard & Triggs, 2004). In this paper, we used the following criterion:

$$\mathcal{L}_{hybrid}(\mathcal{D}_{train}) = \mathcal{L}_{disc}(\mathcal{D}_{train}) + \alpha \mathcal{L}_{gen}(\mathcal{D}_{train}) \qquad (5)$$

where the weight α of the generative criterion can be optimized (e.g., based on the validation set classification error). Here, the generative criterion can also be seen as a data-dependent regularizer for a DRBM. We will refer to RBMs trained using the criterion of equation 5 as Hybrid DRBMs (HDRBMs).

To train an HDRBM, we can use stochastic gradient descent and add for each example the gradient contribution due to $\mathcal{L}_{disc}$ with α times the stochastic gradient estimator associated with $\mathcal{L}_{gen}$ for that example.

5. Semi-supervised Learning

A frequent classification setting is where there are few labeled training data but many unlabeled examples of inputs. Semi-supervised learning algorithms (Chapelle et al., 2006) address this situation by using the unlabeled data to introduce constraints on the trained model. For example, for purely discriminative models, these constraints are often imposed on the decision surface of the model. In the RBM framework, a natural constraint is to ask that the model be a good generative model of the unlabeled data, which corresponds to the following objective:

$$\mathcal{L}_{unsup}(\mathcal{D}_{unlab}) = - \sum_{i=1}^{|\mathcal{D}_{unlab}|} \log p(\mathbf{x}_i) \qquad (6)$$

where $\mathcal{D}_{unlab} = \{(\mathbf{x}_i)\}_{i=1}^{|\mathcal{D}_{unlab}|}$ contains unlabeled examples of inputs. To train on this objective, we can once again use a contrastive divergence approximation of the log-likelihood gradient:

$$\frac{\partial \log p(\mathbf{x}_i)}{\partial \theta} = -\mathbb{E}_{y,\mathbf{h}|\mathbf{x}_i}\left[\frac{\partial}{\partial \theta}E(y_i, \mathbf{x}_i, \mathbf{h})\right] + \mathbb{E}_{y,\mathbf{x},\mathbf{h}}\left[\frac{\partial}{\partial \theta}E(y, \mathbf{x}, \mathbf{h})\right]$$

The contrastive divergence approximation is slightly different here. The first term can be computed in time $O(Cn + nd)$, by noticing that it is equal to

$$\mathbb{E}_{y|\mathbf{x}_i}\left[\mathbb{E}_{\mathbf{h}|y,\mathbf{x}_i}\left[\frac{\partial}{\partial \theta}E(y_i, \mathbf{x}_i, \mathbf{h})\right]\right]$$

One could either average the usual RBM gradient $\frac{\partial}{\partial \theta}E(y_i, \mathbf{x}_i, \mathbf{h})$ for each class y (weighted by $p(y|\mathbf{x}_i)$), or sample a y from $p(y|\mathbf{x}_i)$ and only collect the gradient for that value of y. In the sampling version, the online training update for this objective can be described by replacing the statement $y^0 \leftarrow y_i$ with $y^0 \sim p(y|\mathbf{x}_i)$ in Algorithm 1. We used this version in our experiments.

In order to perform semi-supervised learning, we can weight and combine the objective of equation 6 with those of equations 1, 4 or 5

$$\mathcal{L}_{semi-sup}(\mathcal{D}_{train}, \mathcal{D}_{unlab}) = \mathcal{L}_{\text{TYPE}}(\mathcal{D}_{train}) + \beta\mathcal{L}_{unsup}(\mathcal{D}_{unlab}) \quad (7)$$

where TYPE $\in \{gen, disc, hybrid\}$. Online training according to this objective simply consists in applying the appropriate update for each training example, based on whether it is labeled or not.

6. Related Work

As mentioned earlier, RBMs (sometimes also referred to as harmoniums (Welling et al., 2005)) have already been used successfully in the past to extract useful features for another supervised learning algorithm. One of the main contributions of this paper lies in the demonstration that RBMs can be used on their own without relying on another learning algorithm, and provide a self-contained framework for deriving competitive classifiers. In addition to ensuring that the features learned by the RBM's hidden layer are discriminative, this approach facilitates model selection since the discriminative power of the hidden layer units (or features) can be tracked during learning by observing the progression of classification error on a validation set. It also makes it easier to tackle online learning problems relatively to approaches where learning features (hidden representation) and learning to classify are done in two separate phases (Hinton et al., 2006; Bengio et al., 2007). Gehler et al. (2006); Xing et al. (2005) have shown that the features learned by an RBM trained by ignoring the labeled targets can be useful for retrieving documents or classifying images of objects. However, in both these cases, the extracted features were linear in the input, were not trained discriminatively and had to be fed to another learning algorithm which ultimately performed classification. McCallum et al. (2006) presented Multi-Conditional Learning (MCL)[1] for harmoniums in order to introduce a discriminative component to harmoniums' training, but the learned features still had to be fed to another learning algorithm.

RBMs can also provide a good initialization for the parameters of neural network classifiers (Hinton, 2007), however model selection issues arise, for instance when considering the appropriate number of learning updates and the magnitude of learning rates of each training phase. It has also been argued that the generative learning aspect of RBM training was a key element to their success as good starting points for neural network training (Bengio et al., 2007), but the extent to which the final solution for the parameters of the neural network is influenced by generative learning is not well controlled. HDRBMs can be seen as a way of addressing this issue.

Finally, though semi-supervised learning was never reported for RBMs before, Druck et al. (2007) introduced semi-supervised learning in hybrid generative/discriminative models using a similar approach to the one presented in section 5. However, they worked with log-linear models, whereas the RBMs used here can perform non-linear classification. Log-linear models depend much more on the discriminative quality of the features that are fed as input, whereas an RBM can learn useful features using their hidden variables, at the price of non-convex optimization.

7. Experiments

We present experiments on two classification problems: character recognition and text classification. In all experiments, we performed model selection on a validation set before testing. For the different RBM models, model selection[2] consisted in finding good values for

[1] We experimented with a version of MCL for the RBMs considered in this paper, however the results did not improve on those of HDRBMs.

[2] Model selection was done with a grid-like search over λ (between 0.0005 and 0.1, on a log scale), n (50 to 6000), α for HDRBMs (0 to 0.5, on a log scale) and β for semi-supervised learning (0, 0.01 or 0.1). In general, bigger values of n were found to be more appropriate with more generative learning. If no local minima was apparent, the

the learning rate λ, the size of the hidden layer n and good weights for the different types of learning (generative and semi-supervised weights). Also, the number of iterations over the training set was determined using early stopping according to the validation set classification error, with a look ahead of 15 iterations.

7.1. Character Recognition

We evaluated the different RBM models on the problem of classifying images of digits. The images were taken from the MNIST dataset, where we separated the original training set into training and validation sets of 50000 and 10000 examples and used the standard test set of 10000 examples. The results are given in Table 1. The ordinary RBM model is trained generatively (to model (x, y)), whereas RBM+NNet is an unsupervised RBM used to initialize a one-hidden layer supervised neural net (as in (Bengio et al., 2007)). We give as a comparison the results of a Gaussian kernel SVM and of a regular neural network (random initialization, one hidden layer, hyperbolic tangent hidden activation functions).

First, we observe that a DRBM outperforms a generative RBM. However, an HDRBM appears able to make the best out of discriminative and generative learning and outperforms the other models.

We also experimented with a sparse version of the HDRBM model, since sparsity is known to be a good characteristic for features of images. Sparse RBMs were developed by Lee et al. (2008) in the context of deep neural networks. To introduce sparsity in the hidden layer of an RBM in Lee et al. (2008), after each iteration through the whole training set, the biases $\mathbf{c}$ in the hidden layer are set to a value that maintains the average of the conditional expected value of these neurons to an arbitrarily small value. This procedure tends to make the biases negative and large. We follow a different approach by simply subtracting a small constant δ value, considered as an hyper-parameter[3], from the biases after each update, which is more appropriate in an online setting or for large datasets.

This sparse version of HDRBMs outperforms all the other RBM models, and yields significantly lower classification error then the SVM and the standard neural network classifiers. The performance achieved by the sparse HDRBM is particularly impressive when compared to reported performances for Deep Belief Networks (1.25% in Hinton et al. (2006)) or of a deep neural network initialized using RBMs (around 1.2% in Bengio et al. (2007) and Hinton (2007)) for the MNIST dataset with 50000 training examples.

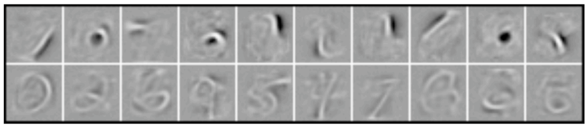


Figure 2. Filters learned by the HDRBM on the MNIST dataset. The top row shows filters that act as spatially localized stroke detectors, and the bottom shows filters more specific to a particular shape of digit.

Table 1. Comparison of the classification performances on the MNIST dataset. SVM results for MNIST were taken from http://yann.lecun.com/exdb/mnist/. On this dataset, differences of 0.2% in classification error is statistically significant.

Model	Error
RBM $(\lambda = 0.005,\ n = 6000)$	3.39%
DRBM $(\lambda = 0.05,\ n = 500)$	1.81%
RBM+NNet	1.41%
HDRBM $(\alpha = 0.01,\ \lambda = 0.05,\ n = 1500\ )$	1.28%
Sparse HDRBM (idem $+\ n = 3000,\ \delta = 10^{-4}$)	**1.16%**
SVM	1.40%
NNet	1.93%

The discriminative power of the HDRBM can be better understood by looking a the rows of the weight matrix $\mathbf{W}$, which act as filter features. Figure 2 displays some of these learned filters. Some of them are spatially localized stroke detectors which can possibly be active for a wide variety of digit images, and others are much more specific to a particular shape of digit.

7.2. Document Classification

We also evaluated the RBM models on the problem of classifying documents into their corresponding newsgroup topic. We used a version of the 20-newsgroup dataset[4] for which the training and test sets contain documents collected at different times, a setting that is more reflective of a practical application. The original training set was divided into a smaller training

grid was extended. The biases $\mathbf{b}$, $\mathbf{c}$ and $\mathbf{d}$ were initialized to 0 and the initial values for the elements of the weight matrices $\mathbf{U}$ and $\mathbf{W}$ were each taken from uniform samples in $\left[-m^{-0.5}, m^{-0.5}\right]$, where m is the maximum between the number of rows and columns of the matrix.

[3] To chose δ, given the selected values for λ and α for the "non sparse" HDRBM, we performed a second grid-search over δ (10^{-5} and 0.1, on a log scale) and the hidden layer size, testing bigger hidden layer sizes then previously selected.

[4] This dataset is available in Matlab format here: http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate-matlab.tgz

set and a validation set, with 9578 and 1691 examples respectively. The test set contains 7505 examples. We used the 5000 most frequent words for the binary input features. The results are given in Figure 3(a). We also provide the results of a Gaussian kernel SVM[5] and of a regular neural network for comparison.

Once again, HDRBM outperforms the other RBM models. However, here the generatively trained RBM performs better then the DRBMs. The HDRBM also outperforms the SVM and neural network classifiers.

In order to get a better understanding of how the HDRBM solves this classification problem, we first looked at the weights connecting each of the classes to the hidden neurons. This corresponds to the columns $\mathbf{U}_{\cdot y}$ of the weight matrix $\mathbf{U}$. Figure 3(b) shows a similarity matrix $\mathbf{M}(\mathbf{U})$ for the weights of the different newsgroups, where $\mathbf{M}(\mathbf{U})_{y_1 y_2} = \mathrm{sigm}(\mathbf{U}^T_{\cdot y_1} \mathbf{U}_{\cdot y_2})$. We see that the HDRBM does not use different neurons for different newsgroups, but shares some of those neurons for newsgroups that are semantically related. Another interesting visualization of this characteristic is given in Figure 3(c), where the columns of $\mathbf{U}$ were projected on their two principal components. In both cases, we see that the HDRBM tends to share neurons for similar topics, such as computer (`comp.*`), science (`sci.*`) and politics (`talk.politics.*`), or secondary topics such as sports (`rec.sports.*`) and other recreational activities (`rec.autos` and `rec.motorcycles`).

Table 2 also gives the set of words used by the HDRBM to recognize some of the newsgroups. To obtain this table we proceeded as follows: for each newsgroup y, we looked at the 20 neurons with the largest weight among $\mathbf{U}_{\cdot y}$, aggregated (by summing) the associated input-to-hidden weight vectors, sorted the words in decreasing order of their associated aggregated weights and picked the first words according to that order. This procedure attempts to approximate the positive contribution of the words to the conditional probability of each newsgroup.

7.3. Semi-supervised Learning

We evaluated our semi-supervised learning algorithm for the HDRBM on both the digit recognition and document classification problems. We also experimented with a version (noted MNIST-BI) of the MNIST dataset proposed by Larochelle et al. (2007) where background images have been added to MNIST digit images. This version corresponds to a much harder problem, but it will help to illustrate the advantage brought by semi-supervised learning in HDRBMs. The HDRBM trained on this data used truncated exponential input units (see (Bengio et al., 2007)).

In this semi-supervised setting, we reduced the size of the labeled training set to 800 examples, and used some of the remaining data to form an unlabeled dataset $\mathcal{D}_{unlab}$. The validation set was also reduced to 200 labeled examples. Model selection[6] covered all the parameters of the HDRBM as well as the unsupervised objective weight β of equation 7. For comparison purposes, we also provide the performance of a standard non-parametric semi-supervised learning algorithm based on function induction (Bengio et al., 2006b), which includes as a particular case or is very similar to other non-parametric semi-supervised learning algorithms such as Zhu et al. (2003). We provide results for the use of a Gaussian kernel (NP-Gauss) and a data-dependent truncated Gaussian kernel (NP-Trunc-Gauss) used in Bengio et al. (2006b), which essentially outputs zero for pairs of inputs that are not near neighbors. The experiments on the MNIST and MNIST-BI (with background images) datasets used 5000 unlabeled examples and the experiment on 20-newsgroup used 8778. The results are given in Table 3, where we observe that semi-supervised learning consistently improves the performance of the HDRBM.

The usefulness of non-parametric semi-supervised learning algorithms has been demonstrated many times in the past, but usually so on problems where the dimensionality of the inputs is low or the data lies on a much lower dimensional manifold. This is reflected in the result on MNIST for the non-parametric methods. However, for high dimensional data with many factors of variation, these methods can quickly suffer from the curse of dimensionality, as argued by Bengio et al. (2006a). This is also reflected in the results for the MNIST-BI dataset which contains many factors of variation, and for the 20-newsgroup dataset where the input is very high dimensional.

Finally, it is important to notice that semi-supervised learning in HDRBMs proceeds in an online fashion and hence could scale to very large datasets, unlike more standard non-parametric methods.

7.4. Relationship with Feed-forward Neural Networks

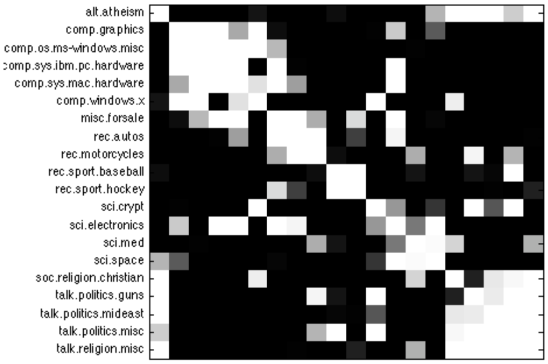
There are several similarities between discriminative RBMs and neural networks. In particular, the computation of $p(y|\mathbf{x})$ could be implemented by a single layer neural network with softplus and softmax acti-

[5] We used `libSVM` v2.85 to train the SVM model

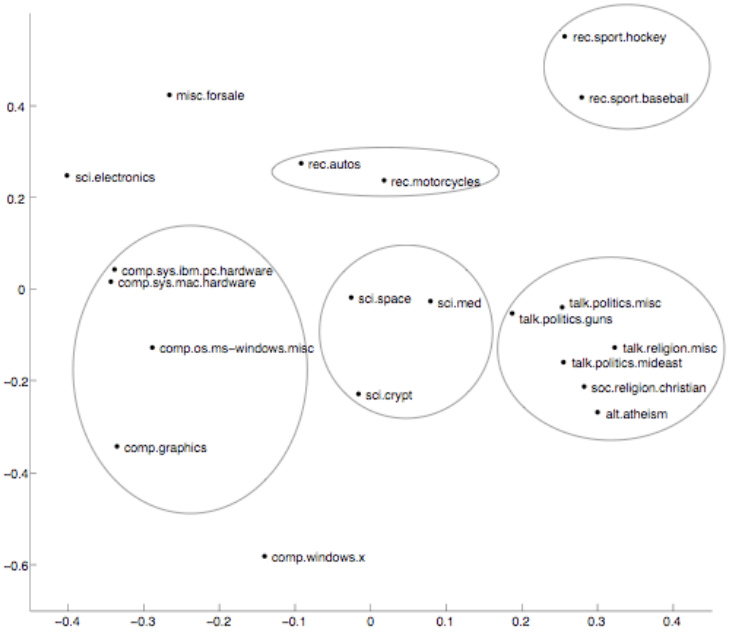
[6] $\beta = 0.1$ for MNIST and 20-newsgroup and $\beta = 0.01$ for MNIST-BI was found to perform best.

Model	Error
RBM ($\lambda = 0.0005$, $n = 1000$)	24.9%
DRBM ($\lambda = 0.0005$, $n = 50$)	27.6%
RBM+NNet	26.8%
HDRBM ($\alpha = 0.005$, $\lambda = 0.1$, $n = 1000$)	**23.8%**
SVM	32.8%
NNet	28.2%

(a) Classification performances



(b) Similarity matrix



(c) PCA embedding

Figure 3. Experiment on 20-newsgroup dataset. (Top left) Classification performance for the different models. The error differences between HDRBM and other models is statistically significant. (Bottom left) Similarity matrix of the newsgroup weights vectors $U_{\cdot y}$. (Right) Two dimensional PCA embedding of the newsgroup weights.

Table 2. Most influential words in the HDRBM for predicting some of the document classes

Class	**Words**	**Class**	**Words**
alt.atheism	bible, atheists, benedikt, atheism, religion	comp.graphics	tiff, ftp, window, gif, images, pixel
misc.forsale	sell, condition, floppy, week, am, obo	rec.autos	cars, ford, autos, sho, toyota, roads
sci.crypt	sternlight, bontchev, nsa, escrow, hamburg	talk.politics.guns	firearms, handgun, firearm, gun, rkba

Table 3. Comparison of the classification errors in semi-supervised learning setting. The errors in bold are statistically significantly better.

Model	MNIST	MNIST-BI	20-news
HDRBM	9.73%	42.4%	40.5%
Semi-sup HDRBM	8.04%	**37.5%**	**31.8%**
NP-Gauss	10.60%	66.5%	85.0%
NP-Trunc-Gauss	**7.49%**	61.3%	82.6%

vation functions in its hidden and output layers respectively, with a special structure in the output and hidden weights where the value of the output weights is fixed and many of the hidden layer weights are shared.

The advantage of working in the framework of RBMs is that it provides a natural way to introduce generative learning, which we used here to derive a semi-supervised learning algorithm. As mentioned earlier, a form of generative learning can be introduced in standard neural networks simply by using RBMs to initialize the hidden layer weights. However the extent to which the final solution for the parameters of the neural network is influenced by generative learning is not well controlled. This might explain the superior performance obtained by a HDRBM compared to a single hidden layer neural network initialized with an RBM (RBM+NNet in the tables).

8. Conclusion and Future Work

We argued that RBMs can and should be used as stand-alone non-linear classifiers alongside other standard and more popular classifiers, instead of merely being considered as simple feature extractors. We evaluated different training objectives that are more appropriate to train an RBM in a classification setting. These discriminative versions of RBMs integrate the process of discovering features of inputs with their use in classification, without relying on a separate classifier. This insures that the learned features are discriminative and facilitates model selection. We also presented a novel but straightforward semi-supervised learning algorithm for RBMs and demonstrated its usefulness for complex or high dimensional data.

For future work, we would like to investigate the use of discriminative versions of RBMs in more challenging settings such as in multi-task or structured output problems. The analysis of the target weights for the 20-newsgroup dataset seem to indicate that RBMs would be good at capturing the conditional statistical relationship between multiple tasks or in the components in a complex target space. Exact computation of the conditional distribution for the target is not tractable anymore, but there exists promising techniques such as mean-field approximations that could estimate that distribution. Moreover, in the 20-newsgroup experiment, we only used 5000 words in input because generative training using Algorithm 1 does not exploit the sparsity of the input, unlike an SVM or a DRBM (since in that case the sparsity of the input makes the discriminative gradient sparse too). Motivated by this observation, we intend to explore ways to introduce generative learning in RBMs and HDRBMs which would be less computationally expensive when the input vectors are large but sparse.

Acknowledgments

We thank Dumitru Erhan for discussions about sparse RBMs and anonymous reviewers for helpful comments.

References

Bengio, Y., Delalleau, O., & Le Roux, N. (2006a). The curse of highly variable functions for local kernel machines. In Y. Weiss, B. Schölkopf and J. Platt (Eds.), *Advances in neural information processing systems 18*, 107–114. Cambridge, MA: MIT Press.

Bengio, Y., Delalleau, O., & Le Roux, N. (2006b). Label propagation and quadratic criterion. In O. Chapelle, B. Schölkopf and A. Zien (Eds.), *Semi-supervised learning*, 193–216. MIT Press.

Bengio, Y., Lamblin, P., Popovici, D., & Larochelle, H. (2007). Greedy layer-wise training of deep networks. *Advances in Neural Information Processing Systems 19* (pp. 153–160). MIT Press.

Bouchard, G., & Triggs, B. (2004). The tradeoff between generative and discriminative classifiers. *IASC International Symposium on Computational Statistics (COMPSTAT)* (pp. 721–728). Prague.

Carreira-Perpiñan, M., & Hinton, G. (2005). On contrastive divergence learning. *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, Jan 6-8, 2005, Savannah Hotel, Barbados* (pp. 33–40). Society for Artificial Intelligence and Statistics.

Chapelle, O., Schölkopf, B., & Zien, A. (2006). *Semi-supervised learning*. Cambridge, MA: MIT Press.

Druck, G., Pal, C., Mccallum, A., & Zhu, X. (2007). Semi-supervised classification with hybrid generative/discriminative methods. *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 280–289). New York, NY, USA: ACM.

Freund, Y., & Haussler, D. (1994). *Unsupervised learning of distributions on binary vectors using two layer networks* (Technical Report UCSC-CRL-94-25). University of California, Santa Cruz.

Gehler, P. V., Holub, A. D., & Welling, M. (2006). The rate adapting poisson model for information retrieval and object recognition. *ICML '06: Proceedings of the 23rd international conference on Machine learning* (pp. 337–344). New York, NY, USA: ACM.

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

Hinton, G. (2007). To recognize shapes, first learn to generate images. In P. Cisek, T. Drew and J. Kalaska (Eds.), *Computational neuroscience: Theoretical insights into brain function*. Elsevier.

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

Larochelle, H., Erhan, D., Courville, A., Bergstra, J., & Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. *Twenty-fourth International Conference on Machine Learning (ICML'2007)*.

Lasserre, J. A., Bishop, C. M., & Minka, T. P. (2006). Principled hybrids of generative and discriminative models. *CVPR '06: Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 87–94). Washington, DC, USA: IEEE Computer Society.

Le Roux, N., & Bengio, Y. (2008). Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation, to appear*.

Lee, H., Ekanadham, C., & Ng, A. (2008). Sparse deep belief net model for visual area v2. In J. Platt, D. Koller, Y. Singer and S. Roweis (Eds.), *Advances in neural information processing systems 20*. Cambridge, MA: MIT Press.

McCallum, A., Pal, C., Druck, G., & Wang, X. (2006). Multi-conditional learning: Generative/discriminative training for clustering and classification. *Twenty-first National Conference on Artificial Intelligence (AAAI-06)*. AAAI Press.

Ng, A. Y., & Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *NIPS* (pp. 841–848).

Salakhutdinov, R., Mnih, A., & Hinton, G. (2007). Restricted boltzmann machines for collaborative filtering. *ICML '07: Proceedings of the 24th international conference on Machine learning* (pp. 791–798). New York, NY, USA: ACM.

Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. In D. Rumelhart and J. McClelland (Eds.), *Parallel distributed processing*, vol. 1, chapter 6, 194–281. Cambridge: MIT Press.

Sutskever, I., & Hinton, G. (2007). Learning multilevel distributed representations for high-dimensional sequences. *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics, March 21-24, 2007, Porto-Rico*.

Taylor, G., Hinton, G., & Roweis, S. (2006). Modeling human motion using binary latent variables. *Advances in Neural Information Processing Systems 20*. MIT Press.

Welling, M., Rosen-Zvi, M., & Hinton, G. E. (2005). Exponential family harmoniums with an application to information retrieval. *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press.

Xing, E. P., Yan, R., & Hauptmann, A. G. (2005). Mining associated text and images with dual-wing harmoniums. *UAI* (pp. 633–641). AUAI Press.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *ICML'2003*.