The Spike-and-Slab RBM and Extensions to Discrete and Sparse Data Distributions

Aaron Courville, Guillaume Desjardins, James Bergstra, and Yoshua Bengio

Abstract—The *spike-and-slab* restricted Boltzmann machine (ssRBM) is defined to have both a real-valued “slab” variable and a binary “spike” variable associated with each unit in the hidden layer. The model uses its slab variables to model the conditional covariance of the observation—thought to be important in capturing the statistical properties of natural images. In this paper, we present the canonical ssRBM framework together with some extensions. These extensions highlight the flexibility of the spike-and-slab RBM as a platform for exploring more sophisticated probabilistic models of high dimensional data in general and natural image data in particular. Here, we introduce the subspace-ssRBM focused on the task of learning invariant features. We highlight the behaviour of the ssRBM and its extensions through experiments with the MNIST digit recognition task and the CIFAR-10 object classification task.

Index Terms—Feature learning, unsupervised learning, restricted boltzmann machines, natural image modeling

1 Introduction

Unsupervised feature learning for natural images is presently the subject of intense research. Approaches to object recognition [4], [5], [6], scene analysis [41] and activity recognition [24] have largely converged on a classification pipeline that begins with at least one feature extraction phase. While standard feature extraction schemes such as SIFT [26] are popular, superior performance has been demonstrated by incorporating *learned features*.

A large variety of modeling paradigms have been applied to the problem, including autoencoders [39], [44], sparse coding [34], and energy-based models. One of the most popular energy-based modeling paradigms for unsupervised feature learning is the restricted Boltzmann machine (RBM). An RBM is a Markov random field with a bipartite graph structure consisting of a visible layer and a hidden layer. The bipartite structure excludes connections between the variables within each layer so that the latent variables are conditionally independent given the visible variables and vice versa. The factorial nature of these conditional distributions enables efficient Gibbs sampling which forms the basis of the most popular RBM learning algorithms such as contrastive divergence [13] and stochastic maximum likelihood [42].

Both as a feature learning scheme [35] and especially as a generative model of natural images [19], [37], [38], RBM-based methods have shown considerable promise. As the canonical energy model for real-valued data, the Gaussian RBM has long been popular as a means of extracting features from natural image data. However, recently [36] have argued that the Gaussian RBM inductive bias is not well suited to the statistical variations present in natural image data. In response to these insights, several alternative models have been proposed to better account for the kinds of variation we see in natural images. These include the mean and covariance RBM (mcRBM) and the mean-product of t-distribution (mPoT) model. Unlike the Gaussian RBM which uses its hidden units to encode the conditional mean of pixels, these models use their hidden units to encode the conditional covariance of the pixels. One drawback of both of these models is that, unlike the standard RBM, the conditional distribution over the observation given the hidden units is not factorial. As a result, the usual (and efficient) inference and training strategies are not compatible with these models.

In this paper, we develop the spike-and-slab RBM (ssRBM). The ssRBM is defined as having each hidden unit associated with the product of a binary *spike* latent variable and a real-valued *slab* latent variable. The name *spike-and-slab* is inspired from terminology in the statistics literature [30], where the term refers to a prior consisting of a mixture between two components: the spike, a discrete probability mass at zero; and the slab, a density (typically uniformly distributed) over a continuous domain.

Spike-and-slab models have previously been explored in the context of factor analyzer-like directed graphical models [9], [10], [27], [31], [43], [49] as well as in a hierarchical extension of such models [14]. The primary advantage that our ssRBM offers over these directed spike-and-slab models is its comparative ease of inference. The ssRBM shares the RBM’s well-known efficient posterior computation, making it an appropriate basis for scalable representation learning.

As a model of natural images, the ssRBM is interesting in that, like the mcRBM and the mPoT model, its binary hidden units encode the conditional covariance of the pixels while simultaneously maintaining the simple conditional independence structure that underlies efficient learning and

- *A. Courville is with the Department of Computer Science and Operations Research, University of Montreal, Montreal, Quebec, Canada. E-mail: aaron.courville@umontreal.ca.*
- *G. Desjardins and Y. Bengio are with the DIRO, University of Montreal, 2920 chemin de la Tour, Montreal, QC H3T1J4, Canada. E-mail: desjagui@iro.umontreal.ca, Yoshua.Bengio@umontreal.ca.*
- *J. Bergstra is with the Centre for Theoretical Neuroscience, University of Waterloo, Waterloo, Ontario, Canada. E-mail: james.bergstra@gmail.com.*

Manuscript received 18 Jan. 2013; revised 6 July 2013; accepted 22 Sept. 2013. Date of publication 4 Dec. 2013; date of current version 29 July 2014.
Recommended for acceptance by M. Welling.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2013.238

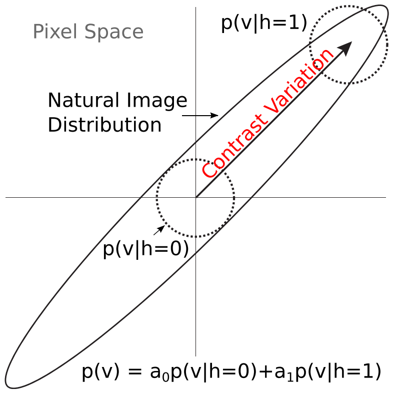

Pixel Space
p(v|h=1)
Natural Image Distribution
Contrast Variation
p(v|h=0)
p(v) = a$_0$p(v|h=0)+a$_1$p(v|h=1)

Fig. 1. An illustration of the Gaussian RBM inductive bias (with one hidden unit) and the statistical structure of natural images. The Gaussian RBM exhibits undesirable sensitivity to local variation in contrast.

inference in the traditional RBM. In the ssRBM, this is accomplished by exploiting real-valued latent slab variables, as was done in [29]. Marginalizing over these variables results in the conditional covariance being parametrized by the binary hidden units. However, conditioning on the slab variables recovers the traditional RBM conditional independence structure.

In this paper, we present the canonical ssRBM framework together with several extensions of the model. These extensions demonstrate the flexibility of the spike-and-slab RBM as a platform for exploring more sophisticated probabilistic models of high dimensional data. In particular, we present variations of the ssRBM model to binary data and to sparse real-valued data that can be represented as spike-and-slab data (either real-valued or exactly zero). Finally, we also present an extension of the ssRBM termed the subspace-ssRBM that ties single binary "spike" variables to subspaces of the observation space. These subspaces are defined through sets of feature vectors, each associated with a slab variable, that span the subspaces. We demonstrate how learned feature subspaces can improve the performance of the extracted features, particularly when the number of training examples is low.

1.1 The Gaussian RBM as a Model of Images

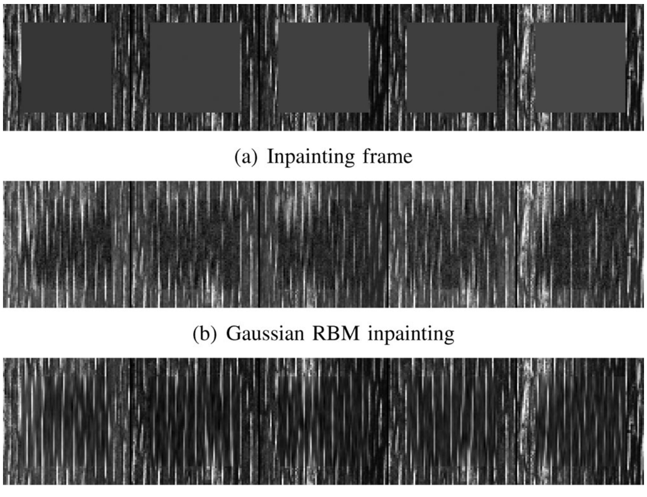
The Gaussian RBM is the simplest RBM that models real-valued data. Taking the number of hidden units to be N and dimensionality of the input to be D, we let $h_i \in \{0,1\}$ for i ranging from 1 to N denote the binary hidden units and $x \in \mathbb{R}^D$ denote a single real-valued observation vector. Assuming that x is drawn from a centered distribution, the Gaussian RBM is specified by the energy function:

$$E(x,h) = \frac{1}{2}x^T\Lambda x - \sum_{i=1}^{N} x^T W_i h_i - \sum_{i=1}^{N} b_i h_i, \tag{1}$$

where W_i is the weight vector of the ith hidden unit, Λ is the diagonal precision matrix on the visible units, and b are the hidden unit biases. From $E(x,h)$ we can derive the conditional distribution over the inputs given the hidden units:

$$p(x \mid h) = \mathcal{N}(Wh, \sigma\mathbf{I}), \tag{2}$$

where $\mathcal{N}(\mu,\Sigma)$ denotes a Gaussian distribution with mean μ and covariance Σ. The scalar σ represents an isometric variance parameter. We can interpret the Gaussian RBM marginal as a Gaussian mixture model with each setting of the hidden units specifying the position of a mixture component. While the number of mixture components scales exponentially with N, these components share a set of parameters that only scales linearly with N.


(a) Inpainting frame

(b) Gaussian RBM inpainting

(c) Spike-and-slab RBM inpainting

Fig. 2. (a) Inpainting frames of Brodatz texture D68. (b) Inpainting with the Gaussian RBM (trained tile-convolutionally—see [28] for details). (c) Inpainting with the ssRBM model in the identical training and text configuration as the Gaussian RBM.

Ranzato and Hinton [35] suggest that the Gaussian RBM is unsatisfactory as a model of natural images because of the model's constant conditional covariance. They argue that the relevant statistics of natural images are captured by the covariance of pixel values rather than absolute pixel values. This point is supported by the widespread use of preprocessing methods that standardize the global scaling of pixel values across each image in a data set. Fig. 1 illustrates the mismatch between the Gaussian RBM inductive bias and a simplified view of the kind of variation we would expect to see in natural images.

The inductive bias of the Gaussian RBM has real consequences on its ability to model natural images. Fig. 2 illustrates the effect of this bias on an inpainting task. The model was trained on a natural texture, shown in Fig. 2a, in a tiled-convolutional configuration (see [28] for details). Fig. 2b shows the results of inpainting with the Gaussian RBM. The inpainted regions exhibit a clearly visible mismatch in contrast at the border of the inpainted region of the image. The spike-and-slab RBM, on the other hand, is better able to match the contrast at the boundary of the inpainted region, as shown in Fig. 2c.

Despite its disadvantages, the Gaussian RBM has the significant advantage that it preserves the classic RBM property that the two conditionals, $p(x \mid h)$ and $P(h \mid x)$ are factorized. Alternative energy formulations such as [36] and [35] trade this important property away in order to model more interesting (at least, non-diagonal) covariance between pixel values. Consequently, the Gaussian RBM remains a standard approach to modeling images and other continuous data in the RBM framework.

2 The Spike-and-Slab RBM

In this section we present the spike-and-slab restricted Boltzmann machine: an extension of the RBM framework designed to improve on the Gaussian RBM as a model of natural images. The ssRBM specifies interactions between three random vectors: the vector x representing the observed data, the latent binary "spike" variables h and the latent real-valued "slab" variables s. The ith hidden unit is associated with the product of the spike element h_i and the slab element s_i of the real-valued variable. Combined, their product gives each hidden unit a sparse value, i.e., either real-valued or zero. Let there be N hidden units: $h \in \{0,1\}^N$, $s \in \mathbb{R}^N$ and a observation vector of dimension D: $x \in \mathbb{R}^D$. The ssRBM model is defined via the energy function:

$$\begin{aligned} E(x,s,h) = &-\sum_{i=1}^N x^T W_i s_i h_i \ + \frac{1}{2} x^T \left(\Lambda + \sum_{i=1}^N \Phi_i h_i \right) x \\ &+ \frac{1}{2} \sum_{i=1}^N \alpha_i s_i^2 \ - \sum_{i=1}^N \alpha_i \mu_i s_i h_i \ - \sum_{i=1}^N b_i h_i \\ &+ \sum_{i=1}^N \alpha_i \mu_i^2 h_i, \end{aligned} \tag{3}$$

where, as with the Gaussian RBM W_i denotes the ith weight vector ($W_i \in \mathbb{R}^D$), b_i is the bias of spike h_i, and Λ is a diagonal precision matrix on the observations x. The ssRBM, though, introduces additional parameters beyond those in the Gaussian RBM. Namely, each $\alpha_i > 0$ is a scalar precision parameter for the real-valued slab variable s_i; each Φ_i is a non-negative diagonal matrix that defines an h-dependent quadratic penalty on x; and each μ_i is a mean parameter for the slab variable s_i. With the energy function thus defined, the joint probability distribution over the model variables is given by:

$$p(x,s,h) = \frac{1}{Z} \exp\{-E(x,s,h)\}, \tag{4}$$

where Z is the normalizing partition function. The ssRBM joint distribution has the very important property that it corresponds to the standard RBM bipartite graph structure with the distinction that the hidden units are considered to form N cliques consisting of paired spike and slab variables h_i and s_i.

One way to understand the ssRBM model is to consider the form of its various conditional distributions. We will begin by comparing the conditional form of $p(x \mid h)$ as it arises in the Gaussian RBM (recall Eq. (2)) and the ssRBM models. In the ssRBM, we recover this conditional by marginalizing out the slab variables s:

$$\begin{aligned} p(x \mid h) &= \frac{1}{P(h)} \frac{1}{Z} \int \exp\{-E(x,s,h)\}\, ds \\ &= \mathcal{N}\left(C_{x\mid h} \sum_{i=1}^N W_i \mu_i h_i,\ C_{x\mid h} \right), \end{aligned} \tag{5}$$

where $C_{x\mid h} = (\Lambda + \sum_{i=1}^N \Phi_i h_i - \sum_{i=1}^N \alpha_i^{-1} h_i W_i W_i^T)^{-1}$. The last equality holds only if the covariance matrix $C_{x\mid h}$ is positive definite, which is not guaranteed from the parametrization. In Section 3, we will discuss a few strategies, via constraints on Λ and Φ, to ensure positive definiteness of $C_{x\mid h}$.

From Eq. (5) we see that, like the Gaussian RBM, the conditional $p(x \mid h)$ is Gaussian-distributed, however unlike the Gaussian RBM, the hidden units not only encode the conditional mean but also specify the conditional covariance $C_{x\mid h}$. Delving a bit deeper into the parametrization of the conditional $p(x \mid h)$, we see that the conditional mean of x given h and principal axis of conditional covariance are related: active hidden variables ($h_i = 1$) for which μ_i is relatively large will tend to align the mean with the principal axes of variance, whereas hidden variables for which μ_i is close to zero will only affect the directions of conditional variance. This flexibility for the model to *adaptively* assign capacity to model either the conditional mean or the conditional variance represents an innovation of the ssRBM over previous work.

While the conditional $p(x \mid h)$ demonstrates that the ssRBM is appropriately parametrized for natural image modeling in that its conditional covariance is fully general (i.e., not restricted to be diagonal as in the Gaussian RBM), the non-diagonal covariance has another immediate consequence for learning: unlike the Gaussian RBM, the elements of the ssRBM conditional $p(x \mid h)$ are not independent. This implies we cannot sample easily and efficiently from this conditional using block Gibbs sampling. It might seem we have gained modeling power at the expense of a more challenging sampling scenario (options include Hybrid Monte Carlo [33], HMC). However, in the case of the ssRBM, another option is available.

In addition to marginalizing out the slab variables s, it is also enlightening to condition on them, so that:

$$\begin{aligned} p(x \mid s,h) &= \frac{1}{p(s,h)} \frac{1}{Z} \exp\{-E(x,s,h)\} \\ &= \mathcal{N}\left(C_{x\mid s,h} \sum_{i=1}^N W_i s_i h_i,\ C_{x\mid s,h} \right), \end{aligned} \tag{6}$$

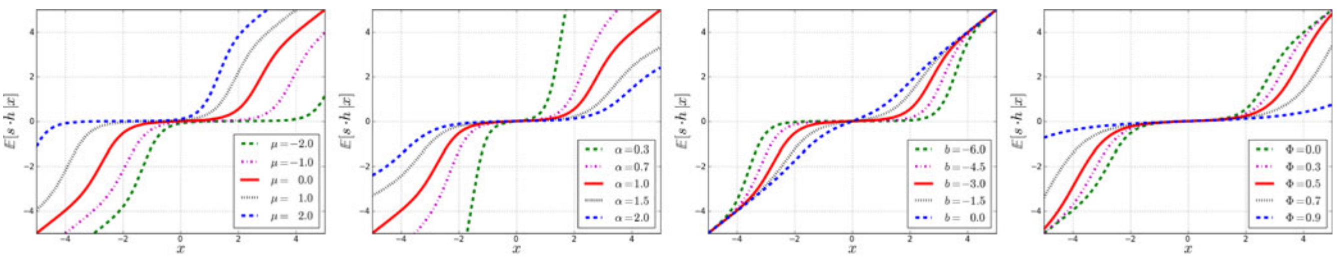
where $C_{x\mid s,h} = (\Lambda + \sum_{i=1}^N \Phi_i h_i)^{-1}$. That is, the conditional distribution of x given both s and h is, once again, Gaussian distributed. Critically though, with diagonal Λ and Φ_i ($\forall\ i \in [1,N]$), the conditional covariance $C_{x\mid s,h}$ is also diagonal. The diagonal covariance allows us to sample from $p(x \mid s,h)$ using block Gibbs sampling. Eq. (19) also shows the role played by Φ_i in augmenting the precision with the activation of h_i. Indeed, hidden unit i contributes a component not only to the mean proportional to $W_i s_i$, but also to the global scaling of the conditional mean.

The next conditional distribution we consider is the conditional over the slabs s given the spikes h and the observation x:

$$p(s \mid x,h) = \prod_{i=1}^N \mathcal{N}\big((\alpha_i^{-1} x^T W_i + \mu_i) h_i,\ \alpha_i^{-1} \big). \tag{7}$$

As was the case with $p(x \mid s,h)$, the conditional $p(s \mid x,h)$ is once again Gaussian distributed with diagonal covariance. Eq. (7) also shows how the mean of the slab variable s_i, given $h_i = 1$, is linearly dependent on x, and as the precision $\alpha_i \to \infty$, s_i converges in probability to μ_i.

Finally we consider the conditional distribution over the latent spike variables h given the observations. When we



(a) s mean param. μ (b) s precision param. α (c) h bias b (d) x precision param. Φ

Fig. 3. The sensitivity of the spike-and-slab RBM feature activation curves, given by $\mathbb{E}[s \cdot h \,|\, x]$, to changes to model parameters (in the case of a one dimensional input v). Default values for the model parameters are: $\mu = 0.0$, $\alpha = 1.0$, $b = -3.0$ and $\Phi = 0.0$. Unless specified by the legend, all parameter are at their default values.

marginalize over s, this time we find that the conditional does factorize, i.e., $P(h \,|\, x) = \prod_i^N P(h_i \,|\, x)$ with

$$\begin{aligned} P(h_i = 1 \,|\, x) &= \frac{1}{p(x)} \frac{1}{Z_i} \int \exp\{-E(x, s, h)\} \; ds \\ &= \operatorname{sigm}\left(\frac{1}{2} \alpha_i^{-1} (x^T W_i)^2 + x^T W_i \mu_i - \frac{1}{2} x^T \Phi_i x + b_i \right), \end{aligned} \tag{8}$$

where sigm indicates a logistic sigmoid. Eq. (8) shows the interaction between three data-dependent terms. The first term, $\frac{1}{2}\alpha_i^{-1}(x^T W_i)^2$, is the contribution due to the variance in s about its mean (note the scaling with α_i^{-1}) and appears in the sigmoid as a result of marginalizing out s. This term is always non-negative, so it always acts to increase $P(h_i \,|\, x)$. Countering this tendency to activate h_i is the other term quadratic in x, $-\frac{1}{2}x^T \Phi_i x$, that is always a non-positive contribution to the sigmoid argument. In addition to these two quadratic terms, there is the term $x^T W_i \mu_i$ whose behaviour mimics the data-dependent term in the analogous Gaussian RBM version of the conditional distribution over h: $P_{\text{GRBM}}(h_i \,|\, x) = \operatorname{sigm}\,(x^T W_i + b_i)$.

One of the interesting aspects of the model is that, when using the model as a feature learning mechanism, the ssRBM offers a choice in the feature representation of the data. One natural choice is the expected product of the spike and slab variables, $\mathbb{E}[s \cdot h \,|\, x]$. This choice has the potentially advantage that, like the RBM with rectified linear hidden units [32], the feature activations scale with the intensity of activation, through the action of the slab variable s. In this way, the feature response can be said to be *equivariant* with respect to changes in intensity. Fig. 3 shows how the quantity $\mathbb{E}[s \cdot h \,|\, x]$ changes as a function of the model parameters. On the other hand, sometimes it is desirable to be *invariant* with respect to changes in the intensity of activation. By taking $P(h \,|\, x)$ as the feature representation, the model can encode correlation patterns while remaining relatively insensitive to local changes in intensity. In the case of natural images, intensity often reflects illumination conditions and contrast levels—factors that are often irrelevant to tasks of interest such as object classification [2].

3 Positive Definite Constraints

The conditional $p(x \,|\, h)$ is only a well-defined Gaussian distribution if the covariance matrix $C_{x \,|\, h}$ is positive definite (PD). However, as previously noted the form of $C_{x \,|\, h}$ (in Eq. (5)) is not parametrized to guarantee that this condition is met. In particular, if there exists an x such that $x^T C_{x \,|\, h} x \leq 0$, then the covariance matrix is not PD.

One way to deal with this issue is to restrict the domain of x to a finite box or ball that encompasses all training data. In the case of the box constraint, this would imply replacing the conditional Gaussian distribution of the visible variables in eq. (19) with a truncated Gaussian distribution. In some cases (including the case of pixel arrays from a CCD sensor) this restriction may be natural, because there are physical reasons why observation vectors must necessarily be limited in magnitude. Generally though, it would be preferable not to fix the modeling domain a priori, and instead permit the training algorithm free reign over all of $\mathbb{R}^D$. To that end, this section examines several techniques for constraining the basic ssRBM so that the conditional covariance $C_{x \,|\, h}$, or equivalently the conditional precision matrix, remains PD.

Specifically, we wish to constrain:

$$x^T C_{x \,|\, h}^{-1} x \; > \; 0 \quad \forall x \neq \mathbf{0}.$$

That is, we need to ensure that $\Lambda + \sum_{i=1}^N \Phi_i h_i$ is, in some sense, large enough to offset $\sum_{i=1}^N \alpha_i^{-1} W_i W_i^T h_i$. Here we consider two basic strategies: (1) define Λ to be large enough to offset a worst-case setting of the h; and (2) define the Φ_i to ensure that the contribution of each active h_i is itself PD.

3.1 Constraining Λ

One option to ensure that $C_{x \,|\, h}$ remains PD for all patterns of h activation is to constrain Λ to be large enough. In setting a constraint on Λ, we will ignore the contribution of the Φ_i terms (which leads to non-tightness of the constraint). Since the contribution of every $\alpha_i^{-1} W_i W_i^T h_i$ term is negative semi-definite, the worst case setting of the h would be to have $h_i = 1$ for all $i \in [1, N]$. This implies that Λ must be constrained such that:

$$x^T \left(\Lambda - \sum_{i=1}^N \alpha_i^{-1} W_i W_i^T \right) x \; > \; 0 \quad \forall x \neq \mathbf{0}. \tag{9}$$

If we constrain Λ to be a scalar matrix, i.e., $\Lambda = \lambda I$, then the problem of enforcing a PD precision matrix reduces to ensuring that λ is greater than the maximum eigenvalue, ρ, of $\sum_{i=1}^N \alpha_i^{-1} W_i W_i^T$. In practice we use the power iteration method to quickly estimate an upper bound on the maximum eigenvalue, and then constrain $\lambda > \rho$ throughout training.

3.2 Constraining Φ_i

Another option to ensure that $C_{x\,|\,h}$ remains PD for all patterns of h activation is to constrain Φ_i to be large enough to ensure that the contribution of each h_i is PD. Let W_{ij} be the jth element of the filter W_i (or equivalently, the ijth element of the weight matrix W) and let Φ_{ij} denote the jjth element of the diagonal Φ_i matrix. We can ensure that $C_{x\,|\,h}$ is positive definite if we constrain Φ_{ij} either as[1]:

$$\Phi_{ij} = \zeta_{ij} + \alpha^{-1} \sum_j W_{ij}^2 I, \tag{10}$$

where Φ_{ij} takes the form of a scalar matrix, or as

$$\Phi_{ij} = \zeta_{ij} + \alpha^{-1} D W_{ij}^2, \tag{11}$$

where the jth elements on the diagonal of Φ_i is scaled with W_{ij}^2 (recall D is the dimension of the observation vector). In both cases, the parameter $\zeta_{ij} > 0$ provides an extra degree of freedom to Φ_{ij} to be estimated through maximum likelihood learning. These are of course not the only option for parametrization of Φ_i. However they are particularly simple constraints that offer complimentary behaviour. In the case of Φ_{ij} parametrized as in Eq. (10), the presence of the $\sum_i^N \Phi_i h_i$ as a scaling term implies that the activation of any h_i will have an effect on the scaling of the mean across the entire observation vector irrespective of how localized is the corresponding filter W_i. Unsurprisingly, use of this parametrization tends to encourage both sparse activation of h_i and W_i having relatively large receptive fields. The parametrization of Φ_{ij} as in Eq. (11) has the property that the Φ_i receptive fields are steered in the direction of W_i. Where W_i is near zero, Φ_i is relatively unconstrained. This is an appealing property for modeling images or other data that give rise to sparse receptive fields W_i. We will empirically explore these constraints and their effect on the ssRBM as a feature learning and extraction algorithm.

4 Learning in the ssRBM

As is typical of RBM-styled models, learning in the ssRBM is rooted in the ability to efficiently draw samples from the model via block Gibbs sampling. As previously discussed, the conditionals $P(h\,|\,x)$, $p(x\,|\,h)$, $p(s\,|\,x,h)$ and $p(x\,|\,s,h)$ possess some important properties with regard to sampling. First, consider the standard RBM Gibbs sampling scheme of iteratively sampling from $P(h\,|\,x)$ and $p(x\,|\,h)$ with s marginalized out. Sampling from $P(h\,|\,x)$ is straightforward, as Eq. (8) indicates that the h_i are all conditionally independent given x. Under the assumption of a positive definite covariance matrix, the conditional distribution $p(x\,|\,h)$ is multivariate Gaussian with non-diagonal covariance $C_{x\,|\,h}$. As previously discussed, sampling from $p(x\,|\,h)$ would require the calculation of the covariance matrix (via matrix inverse) with every weight update. Fortunately, in the case of the ssRBM, rather than sampling directly from $p(x\,|\,h)$, we can sample the slab vector from $p(s\,|\,h,x)$, which is Gaussian distributed with diagonal covariance. Then, given both s and h, we can sample x from the conditional $p(x\,|\,s,h)$, which is also Gaussian distributed with diagonal covariance. Taken all together the triplet $P(h\,|\,x)$, $p(s\,|\,x,h)$ and $p(x\,|\,s,h)$ forms a three-phase block-Gibbs sampling scheme that allows us to sample efficiently from the ssRBM.

In training, we use the stochastic maximum likelihood algorithm (SML, also known as persistent contrastive divergence) [42]. We follow the data log likelihood gradient $\frac{\partial}{\partial\theta_i}(\sum_{t=1}^T \log p(x_t))$, is:

$$-\sum_{t=1}^{T} \left\langle \frac{\partial}{\partial\theta_i} E(x_t, s, h) \right\rangle_{p(s,h\,|\,x_t)} + T \left\langle \frac{\partial}{\partial\theta_i} E(x, s, h) \right\rangle_{p(x,s,h)}.$$

The log likelihood gradient takes the form of a difference between two expectations, over $p(s,h\,|\,x_t)$ in the "clamped" condition, and over $p(x,s,h)$ in the "unclamped" condition. As with the standard RBM, the expectations over $p(s,h\,|\,x_t)$ are amenable to analytic evaluation. The expectations over the model distribution $p(x,s,h)$ is approximated by samples drawn from the ssRBM three-phase Block Gibbs sampler. Typically in SML, only one or a few Markov Chain (Gibbs) simulations are performed between each parameter update.

5 Related Models of Conditional Covariance

As discussed in the introduction, there are other Boltzmann Machine-based models with the goal of modeling the kind of statistical structure found in natural images. For instance, as previously discussed, RBMs with rectified linear hidden units [32] possess a similar equivariance of intensity as the ssRBM slab variables. However, unlike the model of [32], the ssRBM is expressed naturally as a simple energy function. This allows us to consider simple extensions of the model that we consider in later sections. The models that are most closely related to the ssRBM are the mcRBM [35] and the mPoT model [37]. Here we briefly review these models and compare them to the spike-and-slab RBM.

5.1 The Mean and Covariance RBM

Similar to the ssRBM, the mean and covariance RBM is a Boltzmann machine that models the observation x as a multivariate Gaussian distributed quantity with general covariance structure. However it does so via a very different mechanism. The mcRBM uses its hidden layer to independently parametrize both the mean and covariance of the data through two sets of binary hidden units. The model combines the covariance RBM (cRBM) [36] with the Gaussian RBM. The cRBM components model the conditional covariance structure, with the Gaussian RBM capturing the conditional mean. With N_c covariance units: $h^c \in \{0,1\}^{N_c}$, and N_m mean units: $h^m \in \{0,1\}^{N_m}$, the mcRBM model is defined via the energy function:

[1] Courville et al. [7] present the details of the derivation of this constraint.

$$E_{\mathrm{mc}}(x, h^c, h^m) = -\frac{1}{2}\sum_{j=1}^{N^c} h_j^c \big(x^T C_j\big)^2 - \sum_{j=1}^{N^c} b_j^c h_i^c + E_m(x, h^m), \tag{12}$$

where C_j is the weight vector associated with covariance unit h_i^c and b^c is a vector of covariance unit biases. The energy function defines a conditional distribution over the observations given h^m and h^c with a fully general multivariate Gaussian distribution:

$$p_{\mathrm{mc}}(x \mid h^m, h^c) = \mathcal{N}\left(\Sigma\left(\sum_{j=1}^{N^m} W_j h^m\right), \Sigma\right), \tag{13}$$

with covariance matrix $\Sigma = (\sum_{j=1}^{N^c} h_j C_j C_j^T + \mathbf{I})^{-1}$. The conditional distributions over the binary hidden units h_i^m and h_i^c form the basis for the feature representation in the mcRBM and are given by:

$$P_{\mathrm{mc}}(h_i^m = 1 \mid x) = \mathrm{sigm}\left(\sum_{i=1}^{N^m} W_i h_i^c - b_i^m\right),$$

$$P_{\mathrm{mc}}(h_j^c = 1 \mid x) = \mathrm{sigm}\left(\frac{1}{2}\sum_{j=1}^{N^c} h_j^c \big(x^T C_j\big)^2 - b_j^c\right).$$

The mcRBM can be trained using contrastive divergence or SML which require the ability to draw samples from the model. However, due to its non-diagonal conditional covariance structure, sampling from $p_{\mathrm{mcRBM}}(x \mid h^m, h^c)$ would require computing the Σ^{-1} at every iteration of learning. This leads to an impractical computational burden for even moderately sized observations. Ranzato and Hinton [35] avoid direct sampling from the conditional in Eq. (13) by sampling directly from the marginal $p(x)$ using hybrid Monte Carlo [33] on the mcRBM free energy.

5.2 Mean-Product of Student's T-Distributions (mPoTs)

The mean-product of Student's t-distribution model [37] extends the PoT model [45] in a manner similar to how the mcRBM extends the cRBM. Specifically, by including nonzero Gaussian means by the addition of Gaussian RBM-like hidden units. The PoT model is an energy-based model where the conditional distribution over the observation is a multivariate Gaussian (non-diagonal covariance) and the complementary conditional distribution over the hidden variables are a set of conditionally independent Gamma distributions. The mPoT energy function is given as:

$$E_{\mathrm{mp}}(x, h^m, h^c) = E_m(x, h^m) + \sum_j \left(h_j^c\left(1 + \frac{1}{2}\big(C_j^T x\big)^2\right) + (1 - \gamma_j)\log h_j^c\right), \tag{14}$$

where C_j is the weight vector associated with covariance unit h_j^c. The mPoT model energy function specifies a multivariate Gaussian conditional distribution over x with non-diagonal covariance. While the covariance units h^c are conditionally Gamma-distributed:

$$P_{\mathrm{mp}}(h_j^c \mid x) = \mathcal{G}\left(\gamma_j, 1 + \frac{1}{2}\big(C_j^T x\big)^2\right). \tag{15}$$

As both the mPoT model and mcRBM give rise to multivariate Gaussian conditional distributions over the observations with non-diagonal covariance structure, it is unsurprising that mPoT parameter learning encounters the same difficulties as encountered with the mcRBM. Ranzato et al. [37] also advocate direct sampling of $p(x)$ via hybrid Monte Carlo.

5.3 Comparing the ssRBM to the mcRBM, mPoT

The mcRBM and the mPoT model differ from the ssRBM in a number of interesting ways. First, while both the mcRBM and mPoT models resort to hybrid Monte Carlo, the design of the ssRBM admits a simple and efficient Gibbs sampling scheme. It remains to be determined if this difference impacts their relative feasibility though it seems likely that the ssRBM might prove a more flexible framework for further extensions. The later sections of the paper set out to highlight this aspect of the model.

Another difference between these models is how the conditional covariance of the observation is parametrized. The mcRBM and mPoT both model the covariance structure of the observation as $(\sum_{j=1}^{N^c} h_j^c C_j C_j^T + \mathbf{I})^{-1}$, using the activation of the hidden units $h_j > 0$ to enforce constraints on the conditional covariance in the direction C_j. In contrast, the ssRBM specifies the conditional covariance of the observations as $(\Lambda + \sum_{i=1}^{N} \Phi_i h_i - \sum_{i=1}^{N} \alpha_i^{-1} h_i W_i W_i^T)^{-1}$, i.e., using the hidden spike activations $h_i = 1$ to pinch the precision matrix along the direction specified by the corresponding weight vector. In fact, the covariance structure of the ssRBM conditional $p(x \mid h)$ (Eq. (5)) is very similar to the product of probabilistic principal components analysis (PoPPCA) model [46] with components corresponding to the ssRBM weight vectors associated with the active hidden units ($h_i = 1$). In the over-complete setting, sparse activation with the ssRBM parametrization permits significant variance (above the nominal variance given by Λ^{-1}) only in the select directions of the sparsely activated h_i. In the case of the mPoT model or the mcRBM, an over-complete set of constraints on the covariance implies that capturing arbitrary covariance along a particular direction of the input requires removing potentially all constraints with positive projection in that direction. This would suggest that these models are less well suited in the overcomplete setting.

6 Exp. I: ssRBM on Natural Images

In this section, we demonstrate the utility of the ssRBM on the CIFAR-10 data set [22] by classifying images and by sampling from the model. Our experiments explore the roles of μ and Φ and the effects of the Λ and Φ PD constraints.

In this first set of experiments we use the CIFAR-10 image classification data set consisting of 40K training images, 10K validation images, and 10K test images. The images are 32-by-32 pixel RGB images. Each image is

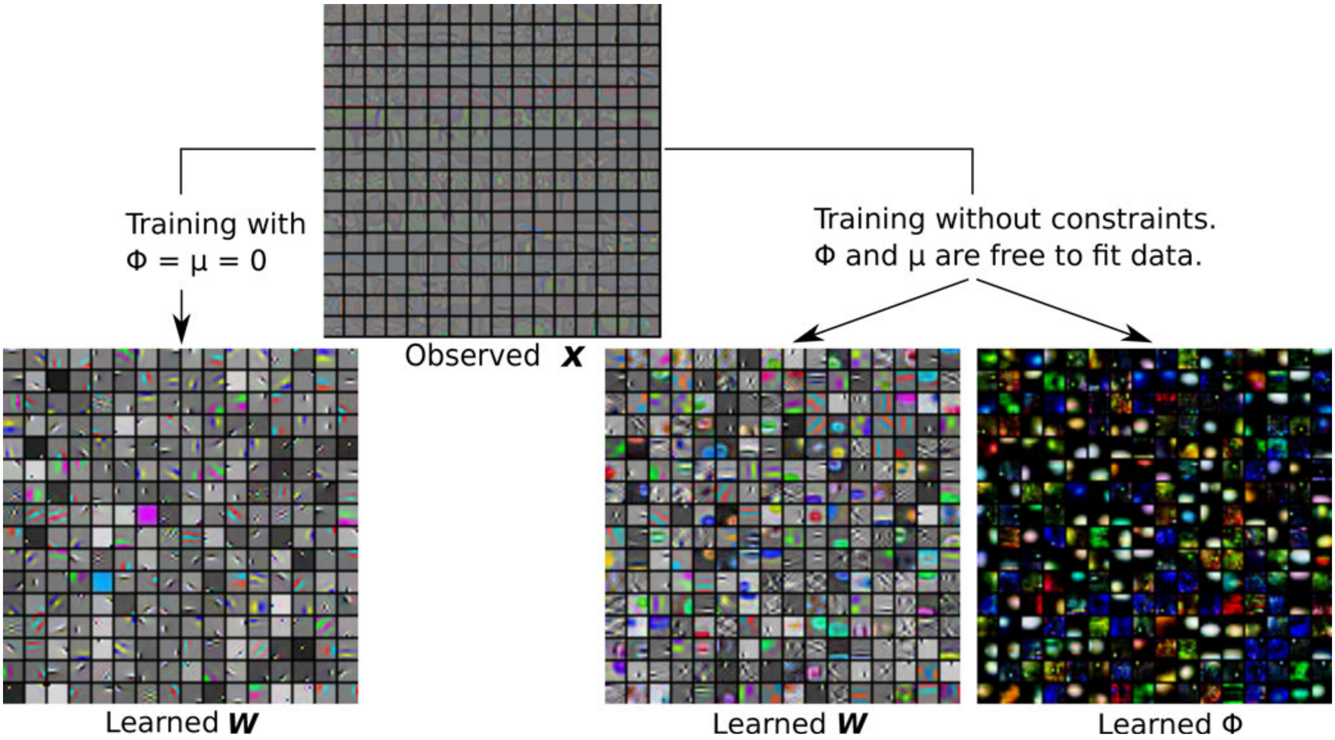


Fig. 4. ZCA-whitened training data and parameters learnt by the ssRBM reveal filters that neatly separate luminance and color oriented edges. The combination of W and Φ gives individual units more representational flexibility, and yields a wider variety of features.

labelled with one of ten object categories (aeroplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck) according to the most prominent object in the image.

6.1 Classification

We evaluate the ssRBM as a feature-extraction scheme by plugging it into the classification pipeline developed by [6]. Broadly, the ssRBM is fit to (192-dimensional) 8×8 RGB image patches, and then applied convolutionally to the 32×32 images. The image patches (starting from pixels between 0 and 255) on which the ssRBM was trained were centered, and then normalized by dividing by the square root of their variance plus a noise-cancelling constant ($c = 10$). The normalized patches were whitened by ZCA [17] with a small positive constant (0.1) added to all eigenvalues. The resulting patches (Fig. 4, top left) are mostly grey with high spatial frequencies amplified, and lower spatial frequencies attenuated. Our models were trained from the 16 non-overlapping 8×8 patches from each of the first 10K training set images in CIFAR-10 (for a total of 160K training examples).

Models were trained for one hundred thousand minibatches of 100 patches. On an NVIDIA GTX 285 GPU this training took on the order of 15 minutes for most models. We used SML training [47], [42]. Classification was done with an ℓ_2-regularized SVM. The SVM was applied to the conditional mean value of latent spike (h) variables, extracted from every 8×8 image patch in the 32×32 CIFAR-10 image. Prior to classification, our conditional h values were spatially pooled into nine regions, analogous to the four quadrants employed in [6]. For a model with N hidden units, the classifier operated on a feature vector of $9N$ elements.

Fig. 4 shows the filters W and Φ for the trained ssRBM. When $\Phi = \mathbf{0}$, the ssRBM filters display the characteristic gabor-like edge detectors and look similar to filters learned using a variety of methods such as sparse coding. When Φ is free to be estimated via approximate likelihood maximization, they tend to form localized receptive fields that match those of the corresponding filters W. Comparing the filters W, between the $\Phi = \mathbf{0}$ and Φ free, shows that Φ can have a significant impact on the evolution of the filters. When Φ is free, the filters W display significantly more variety of form.

Table 1 shows the results of an ablative analysis on the ssRBM model. For this comparison all variants were trained with a mild sparsity penalty aimed at maintaining 15 percent activity, and were configured with 256 hidden units. The sparsity penalty is a KL-divergence penalty penalizing average spike variable activity above and below 15 percent activity. This penalty was done to ensure that all of the hidden units are engaged by the model. The

TABLE 1
The Performance of ssRBM Variants with 256 Hidden Units in CIFAR-10 Image Classification
(± 95 Percent Confidence Intervals)

Model		Accuracy (%)
no PD, μ free, Φ free		73.1 ± 0.9
no PD, μ free, $\Phi = \mathbf{0}$		71.43 ± 0.9
no PD, $\mu = \mathbf{0}$, Φ free		71.19 ± 0.9
no PD, $\mu = \mathbf{0}$, $\Phi = \mathbf{0}$		68.92 ± 0.9
PD by Diag. W	(Eqn. 11)	69.1 ± 0.9
PD by Λ	(Eqn. 9)	68.3 ± 0.9
PD by scal. mat.	(Eqn. 10)	67.1 ± 0.9

"no PD" = without PD constraint.

TABLE 2
The Performance of ssRBM Relative to Other Generative Feature-Learning Models in the Literature for CIFAR-10 (± 95 Percent Confidence Interval)

Model	Accuracy (%)
ssRBM (4096 units)	76.7 ± 0.9
ssRBM (1024 units)	76.2 ± 0.9
ssRBM (512 units)	74.1 ± 0.9
ssRBM (256 units)	73.1 ± 0.9
Deep net, learned RFs (3200)	82.0 ± 0.9
conv. trained DBN	78.9 ± 0.9
mcRBM (225 factors)	68.2 ± 0.9
cRBM (900 factors)	64.7 ± 0.9
cRBM (225 factors)	63.6 ± 0.9
Gaussian RBM	59.7 ± 1.0

strength of regularization was picked to be just strong enough to have the desired effect, of engaging the full set of hidden units. We experimented with simplifications to the energy function $\mu = \mathbf{0}$ and $\Phi = \mathbf{0}$. We found that the full model, with both μ and Φ, without any constraint to operate in a strictly PD regime, worked the best. Removing the μ or Φ term from the energy function cost about 1.5 percent classification accuracy, and removing both cost about 3 percent. The constraints that the model operate in a strictly PD regime also detracted from classification performance by between 4 and 6 percent.

Table 2 situates the performance of the ssRBM in the literature of results on CIFAR-10. The ssRBM outperforms related energy models GRBM, cRBM, and mcRBM as a feature extractor for classification on CIFAR-10. Although some of the differences in performance may almost certainly be attributable to differences in the pre-processing and classification details [35], as we've argued in Section 5.3, since the mcRBM models the data in terms of constraints rather than directions of variance, they are less well suited to the sparse and overcomplete regime where we see the best performance for the ssRBM. In this task, it appears that the ability to model the conditional mean, exhibited by both the ssRBM and the mcRBM is important factor in improving performance over models such as the cRBM that are not able to model the conditional mean with its hidden units. The "conv. trained DBN" result is the convolutionally trained two-layer Deep Belief Network (DBN) with rectified linear units, reported in [21]. Very recently this approach has been improved using maxout networks [11] to achieve just over 87.0 percent accuracy. The "Deep net, learned RFs" result is from [5], a deep neural network is formed by greedy layer-wise unsupervised training, using vector quantization (clustering) to learn the filters and a correlation-based mechanism to learn receptive fields for higher-layer units. Earlier work by [4] has shown that even simple dictionary-learning algorithms such as K-means can yield highly effective feature extractors for CIFAR-10, if only they are used to extract sufficiently many features (thousands). Training the ssRBM with thousands of hidden units yielded less marginal gain than was observed in the case of K-means, but this is possibly because we did not properly optimize the hyper-parameters for this case.

6.2 Model Samples

We also trained a version of the ssRBM convolutionally, following the convolutional RBM described in [21]. Our convolutional implementation of the ssRBM included 1,000 fully-connected units to capture global structure, and 64 hidden units for every image position using 9×9 RGB filters. The model was trained on the CIFAR data set, centered and globally contrast normalized. Filters W and Φ were shared across the image, though independent scalar-parameters μ_i, α_i, and hidden unit bias b_i were allocated for each individual hidden unit.

Fig. 5 illustrates some samples drawn from the model. The samples are taken from the negative phase at the end of training, with the learning rate annealed to near zero, ($\approx 10^{-6}$). These samples exhibit global coherence, and sharp region boundaries. Qualitatively, these samples compare favorably with samples from similar energy-based models, such as those featured in [37] with samples drawn from the mPoT model. Much like we see in binary and Gaussian RBMs, the negative phase Gibbs sampler for a thoroughly trained ssRBM can mix very slowly, as it does in this convolutionally-trained version. As is the case with these other models, we can turn to established methods, such as tempering to overcome this challenge to sampling [3], [8], [40].

7 ssRBM for Discrete Data

Unlike related models such as the mcRBM and mPoT, the ssRBM provides a natural framework for capturing

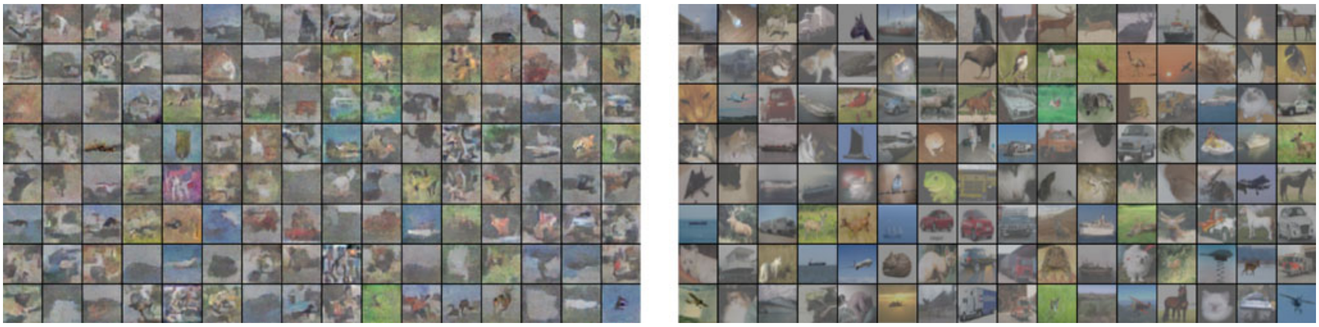


Fig. 5. (Left) Samples drawn from a convolutionally trained ssRBM (Left), and (Right) closest matching images from the CIFAR-10 training set (L2 distance with contrast normalized training images). Samples have the general appearance of data set images, and the dissimilarity to corresponding training images indicates that the model has not memorized training points.

conditional covariance in discrete data. Here we will outline an ssRBM model for binary observations $v \in \{0,1\}^D$. A formulation for multinomial-valued observations ($v \in \{1,\dots,c\}^D$) would be similar.

7.1 Model

The spike-and-slab energy function serves perfectly well when v is binary:

$$\begin{aligned} E(v,s,h) = &-\sum_{i=1}^{N}\sum_{j=1}^{D} v_j W_{ij} s_i h_i - \sum_{j=1}^{D} \rho_j v_j \\ &+ \frac{1}{2}\sum_{i=1}^{N} \alpha_i s_i^2 - \sum_{i=1}^{N} \alpha_i \mu_i s_i h_i - \sum_{i=1}^{N} b_i h_i, \quad (16) \end{aligned}$$

except that compared with the original energy function (Eq. (3)) the quadratic in x is redundant, and in Eq. (16) it collapses to the linear term $\sum_{j=1}^{D} \rho_j v_j$ with visible biases ρ_j, for $j \in [1,\dots,D]$. Note that we have set $\Phi_i = 0$ to simplify the parametrization of the model. This is the case for all of the extensions we will consider in the remainder of this paper. Even with binary data, the real-valued slab variables s are meaningful. Their variation will capture covariance information in the binary v, just as in the case of real-valued x.

With regards to the conditionals, the first thing to note is that changing x from a real-valued vector to the binary vector v does not affect the conditional distributions over s or h that arise from conditioning on v. The conditionals $p(s \mid v,h)$ and $P(h \mid v)$ are identical to their form in the case of real-valued x and are given by Eqs. (7) and (5) respectively. The conditional distributions over v are of course affected by its binary nature. It is straightforward to show that, given s, the conditional distribution over the binary visible vector v factorizes:

$$\begin{aligned} p(v \mid s,h) &= \frac{p(v,s,h)}{p(s,h)} \\ &= \frac{1}{Z'}\prod_{j=1}^{D} \exp\left\{\sum_{k=1}^{M}\sum_{i=1}^{N} v_j W_{ij} s_i h_i + \sum_{k=1}^{M} \rho_j v_j\right\}, \end{aligned}$$

so that the conditionals over the individual elements of v can be expressed by:

$$p(v_j = 1 \mid s,h) = \mathrm{sigmoid}\left(\sum_{i=1}^{N} W_{ij} s_i h_i + \rho_j\right).$$

This conditional is similar to the standard RBM model over binary data, with the addition of real-valued s_i variable.

We can glean some insight into the role of the s_i by considering the conditional over v with s marginalized out:

$$\begin{aligned} p(v \mid h) &= \int p(v,s \mid h)\, ds \\ &\propto \exp\left\{\begin{array}{l} \displaystyle\sum_{j=1}^{D} \rho_j v_j \\ \displaystyle\quad + \sum_{i=1}^{N} \frac{1}{2}\alpha_i \left(\sum_{j=1}^{D} v_j \alpha_i^{-1} W_{ij} + \mu_i\right)^2 h_i \end{array}\right\}. \end{aligned}$$

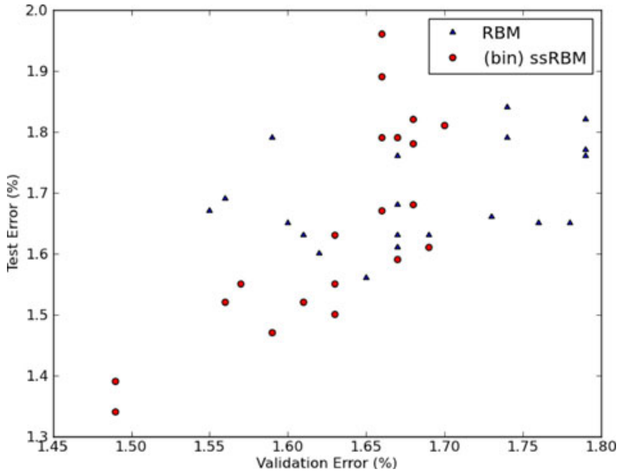


Fig. 6. MNIST classification results. Plot shows test versus validation error for the 20-best RBM and (binary) ssRBM models.

While this distribution is less familiar than the Gaussian that emerged as $p(x \mid h)$ in the real-valued case (Eq. (5)), marginalizing out s critically renders the elements of the input v conditionally *dependent*. An ssRBM on binary-valued observations is not a reparametrized vanilla RBM. Sampling $p(v \mid h)$ directly would be difficult, perhaps requiring sequential Gibbs sampling from the elements of v. Unlike in the real-valued setting, a discrete domain for v removes any concern regarding the potential for a non-PSD conditional covariance.

7.2 Exp. II: Binary Visible Data

We evaluate the binary extension of the ssRBM on MNIST, the hand-written character recognition data set [25]. As with our previous classification experiments, we first perform unsupervised learning using SML and in a second phase, use the latent representation of the RBM (conditional mean of the latent variables h) as input to an ℓ_2-regularized linear SVM.

We compare the ssRBM to a traditional binary-binary RBM, choosing the hyper-parameters through random-search [1]. For each model family, we ran 100 different experiments varying the hyper-parameters as follows: number of hidden units in $[500, 1{,}500]$, initial weights sampled from a zero-mean normal distribution with standard deviation in $\{10^{-3}, 10^{-2}, 10^{-1}, 0.5\}$, sparse activation targets (as described in [12]) in $\{0.05, 0.1, 0.2\}$ and giving the sparsity regularization term a weight in $\{0, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}\}$. Learning rates were chosen to have a linear decreasing schedule, with start and end points randomly sampled[2] from $\{10^{-2}, 10^{-3}, 10^{-4}\}$ and performed up to 500k updates with a mini-batch of size 64. While the unsupervised training was performed on the entire 60k training set, the SVM was trained on the first 50k labels of the MNIST training set only, using the last 10k to select the hyper-parameters. Test set error is reported after retraining the optimal SVM on the entire training set.

The 20 best resulting models are shown in Fig. 6. Overall, the lowest classification error obtained by the binary ssRBM

[2] We additionally constrain the learning rate endpoint to be smaller or equal to the starting learning rate.

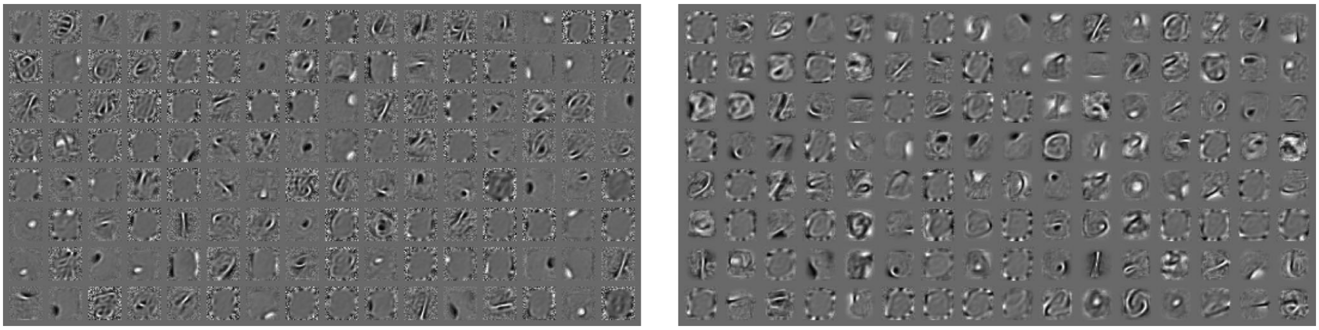


Fig. 7. Random subset of filters drawn from the best performing (Left) RBM and (Right) binary-visible ssRBM models of the MNIST classification experiments of Section 7.2. ssRBM filters are noticeably different and appear more diverse, capturing both local and global structure in the input.

is 1.39 percent, while the RBM achieves 1.67 percent. In comparison, Local Coordinate Coding achieves 1.64 percent error [48]. Interestingly, the filters obtained by the best performing ssRBM are noticeably different than those obtained with an RBM. A random subset of filters is shown in Fig. 7. The ssRBM filters appear more diverse, capturing both local (penstrokes, letter-boxing artifacts) and some more global filters (digit outlines, as well as high-frequency circular gratings).

8 ssRBM for Spike-and-Slab Data

One variation of the ssRBM framework that may be of particular interest is how it can be used to model sparse, real-valued data. In particular, we consider a spike-and-slab RBM on spike-and-slab modeled data (S4RBM). That is we consider an observation pair $[x, v]$ where x is a real-valued slab vector and v are binary spike variables. Obviously this setting is interesting from the perspective of stacking the ssRBM to form a spike-and-slab deep belief net (ssDBN). We can train an ssRBM to model the hidden unit activations of the ssRBM in the layer below. However it might also be an excellent way to model sparse real-valued data where elements of the observation vector are either exactly zero or otherwise real-valued. In this setting, the observation is modeled as the element-wise product $x \circ v$. When $v_j = 0$, one may easily impute the missing x (as it is not directly observed).

8.1 Model

The ssRBM with spike-and-slab observations is defined by the energy function:

$$\begin{aligned} E(x, v, s, h) = &-\sum_{i=1}^{N}\sum_{j=1}^{D} x_j v_j W_{ij} s_i h_i - \sum_{j=1}^{D} \rho_j v_j + \frac{1}{2}\sum_{j=1}^{D} \lambda_j x_j^2 \\ &- \sum_{j=1}^{D} \lambda_j \eta_j x_j v_j + \frac{1}{2}\sum_{j=1}^{D} \lambda_j \eta_j^2 v_j + \frac{1}{2}\sum_{i=1}^{N} \alpha_i s_i^2 \\ &- \sum_{i=1}^{N} \alpha_i \mu_i s_i h_i + \frac{1}{2}\sum_{i=1}^{N} \alpha_i \mu_i^2 h_i - \sum_{i=1}^{N} b_i h_i, \end{aligned} \tag{17}$$

where the model parameters are defined as before, with ρ_j for $j \in [1, \dots, D]$ acting as a bias on v and λ_j is the precision weight for x_j. Note: while it is technically straightforward to include the terms involving the Φ_i to the S4RBM energy function, we have suppressed them to simplify the model.

As one might expect, the conditionals reveal a symmetry between the pair $P(v \mid s, h)$ and $p(x \mid v, s, h)$ and the pair $P(h \mid x, v)$ and $p(s \mid x, v, h)$, all of which factorize with the conditional distributions over the elements given by:

$$\begin{aligned} &P(v_j = 1 \mid s, h) \\ &\quad = \operatorname{sigm}\left(\rho_j + \frac{1}{2}\sigma_j^{-1}\left(\sum_{i=1}^{N} W_{ij} s_i h_i\right)^2 + \eta_j \sum_{i=1}^{N} W_{ij} s_i h_i\right), \\ &p(x_j \mid v, s, h) = \mathcal{N}\left(\left[\eta_j + \sigma_j^{-1}\sum_{i=1}^{N} W_{ij} s_i h_i\right] v_j,\ \sigma^{-1}\right), \\ &P(h_i = 1 \mid x, v) \\ &\quad = \operatorname{sigm}\left(b_i + \frac{1}{2}\alpha_i^{-1}\left(\sum_{j=1}^{D} W_{ij} x_j v_j\right)^2 + \mu_i \sum_{j=1}^{D} W_{ij} x_j v_j\right), \\ &p(s_i \mid x, v, h) = \mathcal{N}\left(\left[\mu_i + \alpha_i^{-1}\sum_{j=1}^{D} W_{ij} x_j v_j\right] h_i,\ \alpha_i^{-1}\right). \end{aligned}$$

In the case of spike-and-slab observations, the block Gibbs sampling scheme becomes a four-phase algorithm iteratively drawing samples from the conditionals $P(h \mid x, v)$, $p(s \mid x, v, h)$, $P(v \mid s, h)$ and then $p(x \mid v, s, h)$.

As revealed in the ssRBM experiments in Section 6, using parameter constraints to ensure that the marginal distribution of the data remains well defined (with a positive definite covariance matrix) resulted in a decrease in classification performance. For this reason, our experiments with the S4RBM used no such constraint. Provided we initialized the model in a stable (PD) regime and used a sufficiently small learning rate ($<=$0.001), we did not experience any difficulty in maintaining the model in this regime.

8.2 Exp. III: Spike-and-Slab Data

We trained stacked S4RBMs on the output of the best one-layer model of Section 7.2. We employed the typical greedy layer-wise training procedure of deep belief networks and used the concatenation of all binary latent variables as input to a ℓ_2-regularized linear SVM. MNIST results are shown in Table 3.

The two-layer ssDBN achieves an impressive test error of 1.21 percent, which is comparable to the original Deep Belief Network results of [15], but were obtained without the need for fine-tuning. The rich latent representation learnt by the ssRBM thus seems better suited at capturing discriminative information present in the input.

TABLE 3
Classification Error Obtained by ssDBNs on MNIST

Model	Validation (%)	Test (%)
ssRBM (1363 units)	1.49	1.39
ssDBN (1363-1000 units)	1.27	**1.21**
ssDBN (1363-1000-1000 units)	1.29	**1.21**

9 Subspace Spike-and-Slab RBM

The principle that invariant features can actually *emerge*, using only unsupervised learning, was first established in the ASSOM model [20]. Since then, the same basic strategy has reappeared in a number of different models and learning paradigms [16], [23], [18], [35]. The strategy is to group filters together, for example, by using a variable (the *pooling* feature) that *gates* the activation of all elements of the group. This gated activation mechanism causes the filters within the group to share a common window on the data set, which in turn leads to filter groups composed of mutually complementary filters. In the end, the span of the filter vectors defines a subspace which specifies the directions in which the pooling feature is invariant. Somewhat surprisingly, this basic strategy has repeatedly demonstrated that useful invariant features can be learned in a strictly unsupervised fashion, using only the statistical structure inherent in the data.

In this section we explore how the spike-and-slab model can be straightforwardly extended to a subspace feature learning method: the subspace-ssRBM. We arrive at the subspace-ssRBM by simply generalizing the slab variable associated with hidden unit i to a slab vector of dimension L: $s_i \in \mathbb{R}^L$ and associating an independent weight vector W_{il} with each element of the slab vector. What this extension implies is that each binary spike variable h_i is associated with a set of L slab variables and their associated weight vectors. Modifying the original ssRBM energy function to incorporate this extension is a trivial matter of converting the relevant scalar operations to vector and matrix operations (not shown). All conditionals are equivalent except the conditional $P(h_i = 1 \,|\, x)$ which must incorporate all interactions between the observations and the weight vectors associated with h_i:

$$\begin{aligned} &P(h_i = 1 \,|\, x) \\ &\quad = \operatorname{sigm}\left(\frac{1}{2}\sum_{l=1}^{L} \alpha_{il}^{-1}(x^T W_{\cdot,il})^2 + \sum_{l=1}^{L} v^T W_{\cdot,il}\mu_{il} - \frac{1}{2} v^T \Phi_i v + b_i\right). \end{aligned} \tag{18}$$

In contrast to the standard ssRBM or the S4RBM, when applied to real-valued data, subspace-ssRBM is more susceptible slipping outside the parameter regime in which the marginal distribution over the visible units is assured to be positive definite. Rather than enforce the parameter constraints we explored in Section 3, we opt for a simple box constraint on the visible variables that enclosed the training data. For each data point all dimensions that fall outside the interval are clipped to the interval boundary. In order to enforce the box constraint for the negative phase samples (see Section 4), for each dimension of x, we use a truncated normal ($\mathcal{TN}$) distribution:

TABLE 4
MNIST Classification Error of a Subspace-ssRBM with (a) $N = 500$ Hidden Units and a Pooling Size $L \in \{1, 3, 5\}$ and (b) $L = 1$ with $N \in \{500, 1.5\text{K}, 2.5\text{K}\}$ as a Function of the Number of Supervised Training Examples

Number of	N=500	N=500		L=1	
Labels	L=1	L=3	L=5	N=1500	N=2500
10	18.21	*14.51*	**13.44**	19.17	22.00
100	5.82	*5.22*	**5.03**	6.32	6.70
1000	2.94	2.70	*2.69*	**2.64**	2.99

$$p(x_j \,|\, s, h) = \mathcal{TN}\left(\mathcal{C}_j \sum_{i=1}^{N}\left(\sum_{i=1}^{L} W_{il} s_{il}\right) h_i \;\; \mathcal{C}_j, \text{lb}, \text{ub}\right), \tag{19}$$

where $\mathcal{C}_j = \left[(\Lambda + \sum_{i=1}^{N} \Phi_i h_i)^{-1}\right]_{jj}$, i.e., the jth element along the diagonal of the covariance matrix. This distribution defines a (re-normalized) Gaussian distribution for the interval: $\text{lb} < x_j < \text{ub}$, while ensuring that $P((x_j >= \text{ub}) \cup (x_j <= \text{lb})) = 0$.

9.1 Exp. IV: Subspace-ssRBM

We now attempt to quantify the effect of pooling on classification performance. In particular, we try to determine whether for a fixed network capacity (i.e., number of filters), it is better to use pooling ($L > 1$) or simply increase the number of hidden units to $L \times N$.

MNIST: We trained various subspace-ssRBM models on MNIST (binary input-valued), measuring classification error as a function of both pooling size and number of training examples used by the SVM. Models were chosen to have either $N = 500$ with $L \in \{1, 3, 5\}$ or $L = 1$ with $N \in \{500, 1.5\text{K}, 2.5\text{K}\}$ hidden units. The number of training labels was restricted to $\{10, 100, 1\text{K}\}$ (per class). Hyperparameters were otherwise chosen from a range similar to Section 7.2, with the exception of learning rates which were held constant in $\{10^{-1}, 10^{-2}, 10^{-3}\}$.

The results are presented in Table 4. We can see that for models having equivalent capacity, pooling is always beneficial in the low-labeled data regime (shown in italic). When limiting ourselves to 10 training labels, the best pooling model ($N = 500, L = 5$) achieves 13.44 percent classification error, a reduction of 29.9 percent compared to the 19.17 percent error achieved by the best un-pooled configuration ($N = 1{,}500, L = 1$). Pooling remains beneficial when using 100 labels, decreasing the error of the best non-pooled model from 6.32 to 5.03 percent with pooling, a decrease of about 20.4 percent. When using 1k labels, the benefits of pooling seem to be offset by the benefits of using a larger output layer: a model with $N = 1{,}500$ hidden units and no-pooling achieved 2.64 percent error, compared to 2.69 percent with pooling.

These results should not come as a surprise. It is a fairly well known result that large over-complete representations are best when using simple linear classifiers [6]. In the low-data regime however, training a large output layer becomes problematic due to overfitting. By allowing each hidden unit to be invariant to a larger subspace of the input, pooling can yield a richer representation, while restricting the dimensionality of the output.

A random subset of filters from a competitive pooling model with $L = 3$ is shown in Fig. 8(left). We can see

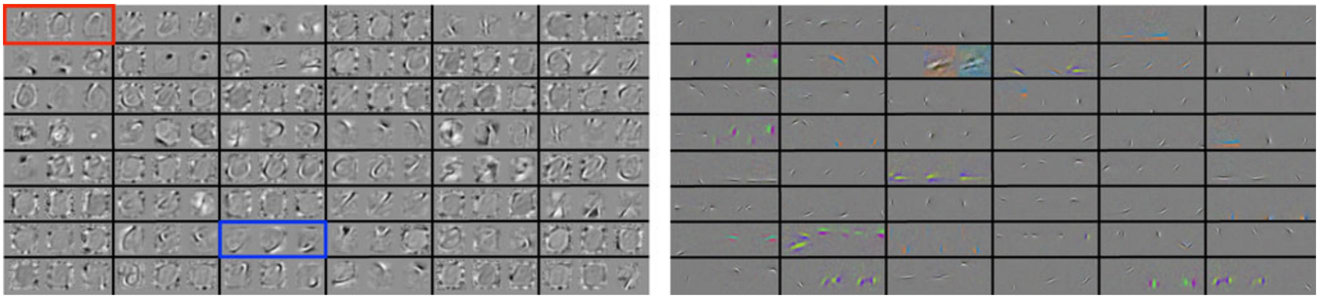


Fig. 8. Random subset of filters drawn from competitive pooled (Left) binary-input ssRBM with $L = 3$, trained on MNIST, and (Right) ssRBM with $L = 3$ trained on global contrast-normalized and whitened CIFAR-10 images. Filters belonging to the same pool are arranged in contiguous blocks of L filters, with two examples shown with a blue and red outline.

that filters belonging to the same pool (consecutive groups of three filters, e.g., outlined in red or blue) tend to learn similar pen-strokes, often with offsets in position, curvature or phase.

CIFAR-10: We now perform similar experiments on the CIFAR-10 data set. Prior to training, the images were preprocessed by first performing global contrast normalization, followed by a ZCA whitening transform. We again compare models with $N = 500$ and $L \in \{1, 3, 5\}$ to models with no-pooling and $N \in \{500, 1.5\text{K}, 2.5\text{K}\}$ and vary the number of training labels in $\{500, 1\text{K}, 2\text{K}\}$.[3] The results are shown in Table 5. With $(N = 500, L = 5)$, we achieve 51.1 percent error, an 11 percent reduction in error compared to the 57.37 percent achieved with $(N = 2{,}500, L = 1)$. As with MNIST, this relative boost in performance drops as we increase the number of training labels at our disposal: an 8.74 percent reduction in error with 1k labels, and 2.3 percent using 2k labels. Note that the goal of this experiment was to perform a comparative analysis. We did not employ convolutional architectures, depth nor large models of 10k units. This explains the gap with other published results [21], [48].

Fig. 8(right) shows a random subset of filters, obtained with $L = 3$. We can clearly see a topological structure, which emerges from the pooling. Filters belonging to the same pool are similar but span rich subspaces through subtle shifts in phase, curvature and orientation of the Gabor-like filters. Interestingly, we also see filters with two (sometimes overlapping) edge detectors, which was not observed with $L = 1$ (not shown).

10 Conclusion

The spike-and-slab RBM offers a powerful framework for modeling real-valued input data, in particular, we have explored its suitability for natural images. Unlike the Gaussian RBM which is limited to modeling diagonal conditional covariances, the slab variables affords the ssRBM rich modeling capacity. Contrary to the mcRBM and mPoT models however, this does not come at the expense of the simple RBM conditional dependency structure: the ssRBM allows for an efficient blocks Gibbs sampling algorithm, by conditioning on the slab variables when sampling from the visible units. One potential drawback of the ssRBM parametrization however, is that its energy function does not guarantee that the conditional covariance on units x be positive definite. This issue can be side-stepped however by imposing additional constraints on Λ and Φ. On the competitive CIFAR-10 data set, the ssRBM was shown to outperform many of its competing methods. Since it does not rely on HMC for generating visible samples, the ssRBM can also be naturally extended to modeling covariances in binary data and even sparse data.

In this paper we have also proposed two novel variants of the spike-and-slab latent variable framework that target (1) discrete (especially binary) data and (2) spike-and-slab data. We use the first of these to show how the spike-and-slab framework can yield superior performance as a feature extractor compared to the standard RBM for classification on the MNIST data set. We then showed how we can use the model variant with spike-and-slab data to support the stacking of spike-and-slab RBMs in a spike-and-slab deep belief network. This model was shown to provide fairly competitive results on MNIST classification without the use of fine-tuning the model parameters for the discriminative task. We expect that fine-tuning the ssDBN would result in further improvements in performance. Finally, by extending slab variables to be vector-valued, one can learn features (spikes) which are invariant to a subspace spanned by the set of filters associated with each slab. We demonstrate the utility of this modeling framework on the MNIST and CIFAR-10 data set in the low-data limit. We should note that in the low-labeled-training-data regime, that organizing the model capacity in the form of these pools leads to a significant boost in classification performance.

Acknowledgments

The authors acknowledge NSERC, FQRNT, CIFAR, RQCHP and Compute Canada for their financial and computational support; and Chris Williams for pointing out the connection to the PoPPCA model. They also

TABLE 5
CIFAR-10 Classification Error of a Fully-Connected Subspace ssRBM with (a) $N = 500$, $L \in \{1, 3, 5\}$ and (b) $L = 1$ with $N \in \{500, 1.5\text{K}, 2.5\text{K}\}$ As a Function of the Number of Supervised Training Examples

Number of	N=500	N=500		L=1	
Labels	L=1	L=3	L=5	N=1500	N=2500
500	57.23	*53.16*	**51.10**	58.09	57.37
1000	54.29	*50.42*	**48.26**	53.56	52.88
2000	52.34	*48.95*	**46.57**	49.64	47.66

[3] Using fewer labels yielded a significantly worse performance, regardless of model capacity and pooling size.

thank the anonymous reviewers for their valuable suggestions for improving the paper.

References

[1] J. Bergstra and Y. Bengio, "Random Search for Hyper-Parameter Optimization," *The J. Machine Learning Research*, vol. 13, pp. 281-305, 2012.

[2] J. Bergstra, Y. Bengio, and J. Louradour, "Suitability of V1 Energy Models for Object Classification," *Neural Computation*, vol. 23, no. 3, pp. 774-790, Mar. 2011.

[3] K. Cho, T. Raiko, and A. Ilin, "Parallel Tempering Is Efficient for Learning Restricted Boltzmann Machines," *Proc. Int'l Joint Conf. Neural Networks (IJCNN '10)*, 2010.

[4] A. Coates and A.Y. Ng, "The Importance of Encoding versus Training with Sparse Coding and Vector Quantization," *Proc. 28th Int'l Conf. Machine Learning (ICML '11)*, 2011.

[5] A. Coates and A.Y. Ng, "Selecting Receptive Fields in Deep Networks," *Proc. Advances in Neural Information Processing Systems (NIPS '11)*, 2011.

[6] A. Coates, H. Lee, and A.Y. Ng, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," *Proc. Int'l Conf. Artificial Intelligence and Statistics (AISTATS '11)*, 2011.

[7] A. Courville, J. Bergstra, and Y. Bengio, "Unsupervised Models of Images by Spike-and-Slab RBMs," *Proc. Int 'l Conf. Machine Learning (ICML '11)*, 2011.

[8] G. Desjardins, A. Courville, Y. Bengio, P. Vincent, and O. Delalleau, "Tempered Markov Chain Monte Carlo for Training of Restricted Boltzmann Machine," *Proc. Int'l Conf. Artificial Intelligence and Statistics (AISTATS)*, vol. 9, pp. 145-152, 2010.

[9] P. Garrigues and B. Olshausen, "Learning Horizontal Connections in a Sparse Coding Model of Natural Images," *Proc. Advances in Neural Information Processing Systems (NIPS '20)*, 2008.

[10] I. Goodfellow, A. Courville, and Y. Bengio, "Large-Scale Feature Learning with Spike-and-Slab Sparse Coding," *Proc. 29th Int'l Conf. Machine Learning (ICML '12)*, 2012.

[11] I.J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio, "Maxout Networks," *Proc. Int'l Conf. Machine Learning (ICML '13)*, 2013.

[12] G. Hinton, "A Practical Guide to Training Restricted Boltzmann Machines," Technical Report 2010-003 Univ. of Toronto, 2010.

[13] G.E. Hinton, "Training Products of Experts by Minimizing Contrastive Divergence," *Neural Computation*, vol. 14, pp. 1771-1800, 2002.

[14] G.E. Hinton, B. Sallans, and Z. Ghahramani, "A Hierarchical Community of Experts," *Proc. NATO Advanced Study Inst. on Learning in Graphical Models*, pp. 479-494, 1998.

[15] G.E. Hinton, S. Osindero, and Y.W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, pp. 1527-1554, 2006.

[16] A. Hyvärinen and P. Hoyer, "Emergence of Phase and Shift Invariant Features by Decomposition of Natural Images into Independent Feature Subspaces," *Neural Computation*, vol. 12, no. 7, pp. 1705-1720, 2000.

[17] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*. Wiley-Interscience, 2001.

[18] K. Kavukcuoglu, M.-A. Ranzato, R. Fergus, and Y. LeCun, "Learning Invariant Features through Topographic Filter Maps," *Proc. IEEE Conf. Computer Vision anad Pattern Recognition (CVPR '09)*, 2009.

[19] J.J. Kivinen and C.K.I. Williams, "Multiple Texture Boltzmann Machines," *Proc. 15th Int'l Conf. Artificial Intelligence and Statistics (AISTATS '12)*, 2012.

[20] T. Kohonen, "Emergence of Invariant-Feature Detectors in the Adaptive-Subspace Self-Organizing Map," *Biological Cybernetics*, vol. 75, pp. 281-291, 1996.

[21] A. Krizhevsky, "Convolutional Deep Belief Networks on CIFAR-10," technical report, U. Toronto, 2010.

[22] A. Krizhevsky and G. Hinton, "Learning Multiple Layers of Features from Tiny Images," technical report, Univ. of Toronto, 2009.

[23] Q. Le, J. Ngiam, Z. Chen, D.Jin hao Chia, P.Wei Koh, and A. Ng, "Tiled Convolutional Neural Networks," *Proc. Advances in Neural Information Processing Systems (NIPS '10)*, 2010.

[24] Q.V. Le, W.Y. Zou, S.Y. Yeung, and A.Y. Ng, "Learning Hierarchical Invariant Spatio-Temporal Features for Action Recognition with Independent Subspace Analysis," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '11)*, 2011.

[25] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient Based Learning Applied to Document Recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.

[26] D. Lowe, "Object Recognition from Local Scale Invariant Features," *Proc. IEEE Seventh Int'l Conf. Computer Vision (ICCV '99)*, 1999.

[27] J. Lücke and A.-S. Sheikh, "A Closed-Form EM Algorithm for Sparse Coding," arXiv:1105.2493, 2011.

[28] H. Luo, P.L. Carrier, A. Courville, and Y. Bengio, "Texture Modeling with Convolutional Spike-and-Slab RBMs and Deep Extensions," *Proc. Int'l Conf. Artificial Intelligence and Statistics (AISTATS '13)*, 2013.

[29] J. Martens and I. Sutskever, "Parallelizable Sampling of Markov Random Fields," *Proc. Int'l Conf. Artificial Intelligence and Statistics (AISTATS '10)*, 2010.

[30] T.J. Mitchell and J.J. Beauchamp, "Bayesian Variable Selection in Linear Regression," *J. Am. Statistical Assoc.*, vol. 83, no. 404, pp. 1023-1032, 1988.

[31] S. Mohamed, K. Heller, and Z. Ghahramani, "Bayesian and L1 Approaches to Sparse Unsupervised Learning," arXiv:1106.1157, 2011.

[32] V. Nair and G.E Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," *Proc. 27th Int'l Conf. Machine Learning*, 2010.

[33] R.M. Neal, "Probabilistic Inference Using Markov Chain Monte-Carlo Methods," Technical Report CRG-TR-93-1 Dept. of Computer Science, Univ. of Toronto, 1993.

[34] B.A. Olshausen and D.J. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature*, vol. 381, pp. 607-609, 1996.

[35] M. Ranzato and G.H. Hinton, "Modeling Pixel Means and Covariances Using Factorized Third-Order Boltzmann Machines," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '10)*, pp. 2551-2558, 2010.

[36] M. Ranzato, A. Krizhevsky, and G.E. Hinton, "Factored 3-Way Restricted Boltzmann Machines for Modeling Natural Images," *Proc. Int'l Conf. Artificial Intelligence and Statistics (AISTATS '10)*, 2010.

[37] M. Ranzato, V. Mnih, and G. Hinton, "Generating More Realistic Images Using Gated MRF's," *Proc. Advances in Neural Information Processing Systems (NIPS '10)*, 2010.

[38] M.A. Ranzato, J. Susskind, V. Mnih, and G. Hinton, "On Deep Generative Models with Applications to Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '11)*, 2011.

[39] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive Auto-Encoders: Explicit Invariance During Feature Extraction," *Proc. 28th Int'l Conf. Machine Learning (ICML '11)*, 2011.

[40] R. Salakhutdinov, "Learning in Markov Random Fields Using Tempered Transitions," *Proc. Advances in Neural Information Processing Systems (NIPS '10)*, 2010.

[41] R. Socher, C. Manning, and A.Y. Ng, "Parsing Natural Scenes and Natural Language with Recursive Neural Networks," *Proc. Int'l Conf. Machine Learning (ICML '11)*, 2011.

[42] T. Tieleman, "Training Restricted Boltzmann Machines Using Approximations to the Likelihood Gradient," *Proc. Int'l Conf. Machine Learning (ICML '08)*, pp. 1064-1071, 2008.

[43] M.K. Titsias and M. Lázaro-Gredilla, "Spike and Slab Variational Inference for Multi-Task and Multiple Kernel Learning," *Proc. Advances in Neural Information Processing Systems (NIPS '11)*, 2011.

[44] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and Composing Robust Features with Denoising Autoencoders," *Proc. 25th Int'l Conf. Machine Learning (ICML '08)*, 2008.

[45] M. Welling, G.E. Hinton, and S Osindero, "Learning Sparse Topographic Representations with Products of Student-T Distributions," *Proc. Advances in Neural Information Processing Systems (NIPS '02)*, 2003.

[46] C.K.I. Williams and F.V. Agakov, "Products of Gaussians and Probabilistic Minor Component Analysis," *Neural Computation*, vol. 14, no. 5, pp. 1169-1182, 2002.

[47] L. Younes, "On the Convergence of Markovian Stochastic Algorithms with Rapidly Decreasing Ergodicity Rates," *Stochastics and Stochastics Models*, pp. 177-228, 1998.

[48] K. Yu and T. Zhang, "Improved Local Coordinate Coding Using Local Tangents," *Proc. Int'l Conf. Machine Learning (ICML '10)*, 2010.

[49] M. Zhou, H. Chen, J.W. Paisley, L. Ren, G. Sapiro, and L. Carin, "Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations," *Proc. Advances in Neural Information Processing Systems (NIPS '09)*, pp. 2295-2303, 2009.

Aaron Courville is an assistant professor in the Department of Computer Science and Operations Research at the University of Montreal, Canada. His recent research interests have been focused on the development of deep learning models and methods. He is particularly interested in developing probabilistic models and novel inference methods.

Guillaume Desjardins is currently working toward the doctoral studies at the University of Montreal, Canada, under the direction of Yoshua Bengio and Aaron Courville. His main research interests are in probabilistic models and deep neural networks, applied to natural signal processing.

James Bergstra completed doctoral studies at the University of Montreal, Canada, in 2011 under the direction of Yoshua Bengio. He codeveloped Theano, an open source optimizing compiler that can make use of GPUs for high-performance computation. He was a postdoctoral researcher at Harvard University with David Cox and the University of Waterloo with Chris Eliasmith. Currently, he holds a Banting Fellowship at the Centre for Theoretical Neuroscience at the University of Waterloo.

Yoshua Bengio is a full professor in the Department of Computer Science and Operations Research and the head of the Machine Learning Laboratory (LISA) at the University of Montreal, Canada, CIFAR fellow in the Neural Computation and Adaptive Perception program, Canada Research chair in Statistical Learning Algorithms, and he also holds the NSERC-Ubisoft industrial chair. His main research ambition is to understand principles of learning that yield intelligence.