A Spike and Slab Restricted Boltzmann Machine

Aaron Courville **James Bergstra** **Yoshua Bengio**
DIRO, Université de Montréal, Montréal, Québec, Canada
{courvila,bergstrj,bengioy}@iro.umontreal.ca

Abstract

We introduce the *spike and slab* Restricted Boltzmann Machine, characterized by having both a real-valued vector, *the slab*, and a binary variable, *the spike*, associated with each unit in the hidden layer. The model possesses some practical properties such as being amenable to Block Gibbs sampling as well as being capable of generating similar latent representations of the data to the recently introduced mean and covariance Restricted Boltzmann Machine. We illustrate how the spike and slab Restricted Boltzmann Machine achieves competitive performance on the CIFAR-10 object recognition task.

1 Introduction

The prototypical Restricted Boltzmann Machine (RBM) is a Markov random field with a bipartite graph structure that divides the model variables into two layers: a visible layer consisting of binary variables representing the data, and a hidden (or latent) layer consisting of the latent binary variables. The bipartite structure excludes connections between the variates (or units) within each layer so that the units within the hidden layer are conditionally independent given the units of the visible layer, and the visible layer units are conditionally independent given the hidden layer units. This pair of conditionally factorial distributions permits a simple block Gibbs sampler, alternating between the dual conditionals P(visible layer | hidden layer) and P(hidden layer | visible layer). The ability to sample simply and efficiently from the RBM forms the basis for effective learning algorithms such as contrastive divergence [8, 2] and stochastic maximum likelihood [28, 23].

Appearing in Proceedings of the 14^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2011, Fort Lauderdale, FL, USA. Volume 15 of JMLR: W&CP 15.

While the RBM has proved effective in a range of tasks and data domains [11, 13, 20, 22, 21, 3, 6], it has not been as successful in modeling continuous multivariate data, and natural images in particular [17]. The most popular approach to modeling continuous observations within the RBM framework has been the so-called Gaussian RBM (GRBM), defined such that the conditional distribution of the visible layer given the hidden layer is a fixed covariance Gaussian with the conditional mean parametrized by the product of a weight matrix and a *binary* hidden vector. Thus the GRBM can be viewed as a Gaussian mixture model with the number of components being exponential in the number of hidden units.

The GRBM has proved unsatisfactory as a model of natural images, as the trained features typically do not represent sharp edges that occur at object boundaries and lead to latent representations that are not particularly useful features for classification tasks [17]. Ranzato and Hinton (2010) have argued that the failure of the GRBM to adequately capture the statistical structure apparent in natural images stems from the exclusive use of the model capacity to capture the conditional mean at the expense of the conditional covariance. While we agree that the GRBM provides a poor covariance model, we suggest that this deficiency has more to do with the binary nature of the hidden layer units than with the model's devotion to capturing the conditional mean.

Our perspective on the GRBM motivates us to reconsider the strategy of modelling continuous-valued inputs with strictly binary latent variables, and leads us to the spike and slab Restricted Boltzmann Machine (ssRBM). Like many RBM variants, the spike and slab RBM is restricted to a bipartite graph structure between two types of nodes. The visible layer units are modeled as real-valued variables as in the GRBM approach. Where our model departs from other similar methods is in the definition of the hidden layer latent variables. We model these as the element-wise product of a real-valued vector with a binary vector, i.e., each hidden unit is associated with a binary *spike* variable and the real vector valued *slab* variable. The name *spike and slab* is inspired from terminology in the statistics literature [12], 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.

In this paper, we show how the introduction of the slab variables to the GRBM leads to an interesting new RBM. By marginalizing out the slab variables, the conditional distribution of the spike variables given the input is very similar to the corresponding conditional of the recently introduced covariance RBM (cRBM) [18]. On the other hand, conditional on the spike variables, the ssRBM slab variables and input are jointly Gaussian and form conditionals with diagonal covariance matrices. Thus, unlike the cRBM or its extension the mean-covariance RBM (mcRBM), the ssRBM is amenable to simple and efficient Gibbs sampling. This property of the ssRBM makes the model an excellent candidate as a building block for the development of more sophisticated models such as the Deep Boltzmann Machine [19].

As we develop the model, we show that with multi-dimensional slab variables, feature "sum" pooling becomes a natural part of the model. In the experiments, we illustrate how maximum likelihood training of the ssRBM yields filters that capture natural image properties such as sharp edges. We also show how the model exhibits "disentangling" of color and edge features when trained on natural image patches and how the ssRBM can learn good features for the CIFAR-10 object classification dataset. [10].

2 The Inductive Bias of the GRBM

Before delving into the development of the ssRBM, we first elaborate on our perspective that the failure of the GRBM to model natural images is due to the use of binary hidden units. We argue this case by comparing the GRBM to a standard Gaussian factor model with a Gaussian distributed latent vector, $x \in \mathbb{R}^N$, and a Gaussian conditional distribution over the observations, $v \in \mathbb{R}^D$, given the latent variable. That is to say, $x \sim \mathcal{N}(0, \Sigma_x)$ and $v|x \sim \mathcal{N}(Wx, \sigma_v \mathrm{I})$, where W is a matrix $(D \times N)$ of weights. Under this model, variations in a single element x_i reflect covariance within the observation vector along the direction (in the input or v space) of $W_{:,i}$. Indeed marginalizing out the latent variables, we are left with the marginal distribution over the observation vector: $p_x(v) \sim \mathcal{N}(0, \sigma_v \mathrm{I} + W\Sigma_x W^T)$. Note that the weights W that parametrize the conditional mean serve also to parametrize the marginal covariance. The GRBM

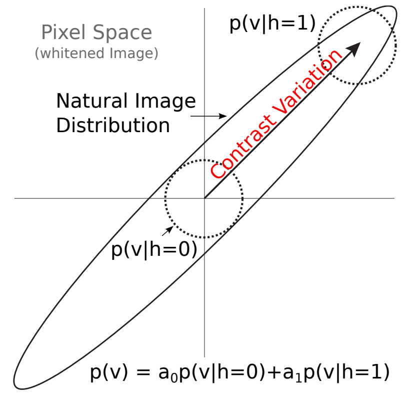

Figure 1: GRBM exhibits significant sensitivity to variation in contrast.

is different from the Gaussian factor model in a number of important ways, but most relevant for our purposes, the GRBM replaces the real-valued latent variables of the factor model with binary variables. If we replace the real-valued x in the factor model with simple binary variables h, the equivalence between parametrizing the conditional mean and parametrizing the marginal covariance breaks down. Instead of a single Gaussian with covariance $\sigma_v \mathrm{I} + W\Sigma_x W^T$, the marginal distribution $p(v)$ becomes the mixture of Gaussians: $p_h(v) = \sum_h P(v)\mathcal{N}(Wh, \sigma_v \mathrm{I})$.

This change from a real variable x to a binary variable h has an impact on the inductive bias of the model and consequently an impact on the suitability of the model to a particular data domain. Both the zero-mean Gaussian $p_x(v)$ and the mixture model $p_h(v)$ exhibit a preference (in the sense of higher probability density) for data distributed along the directions of the columns of their respective weight matrices. However, if the statistical structure of the data is such that density should be relatively invariant to overall scaling of v, then the inductive bias resulting from the binary h may be inappropriate. Figure 1 illustrates how the discrete mixture components in $p_h(v)$ are ill-suited to model natural images, where some of the most significant determiners of the norm of the data vector $\|v\|_2$ are the illumination conditions of the scene and the image contrast. Variation in contrast often bears little relevance to typical tasks of interest such as object recognition or scene understanding. This perspective on the GRBM, and especially its comparison to the standard Gaussian factor model, motivates us to consider alternatives to strictly binary latent variables and leads us to the spike and slab RBM.

3 The Spike and Slab RBM

Let the number of hidden units be N, and the dimensionality of the visible vector to be D: $v \in \mathbb{R}^D$. The ith hidden unit ($1 \leq i \leq N$) is associated with a binary *spike* variable: $h_i \in \{0,1\}$ and a real valued vector $s_i \in \mathbb{R}^K$, pooling over the K features.[1] The energy function for one example is:

$$E(v,s,h) = \frac{1}{2}v^T\Lambda v - \sum_{i=1}^{N}\left(v^T W_i s_i h_i + \frac{1}{2}s_i^T\alpha_i s_i + b_i h_i\right), \tag{1}$$

where W_i refers to the ith weight matrix of size $D \times K$, the b_i are the biases associated with each of the spike variables h_i, and α_i and Λ are diagonal matrices that penalize large values of $\|s_i\|_2^2$ and $\|v\|_2^2$ respectively. We will consider a joint probability distribution over v, $s = [s_1, \ldots, s_N]$ and $h = [h_i, \ldots, h_N]$ of the form:

$$p(v,s,h) = \frac{1}{Z}\exp\{-E(v,s,h)\} \times \mathbb{U}(v;R) \tag{2}$$

where, Z is the partition function that assures that $p(v,s,h)$ is normalized and $\mathbb{U}(v;R)$ represents a distribution that is uniform over a ball radius R, centered at the origin, that contains all the training data, i.e., $R > \max_t \|v_t\|_2$ (t indexes over training examples). The region of the visible layer space outside the ball has zero probability under the model. This restriction to a finite domain guarantees that the partition function Z remains finite. We can think of the distribution presented in equations 2 and 1, as being associated with the bipartite graph structure of the RBM with the distinction that the hidden layer is composed of an element-wise product of the vectors s and h.

With the joint distribution thus defined, we now turn to deriving the set of conditional distributions $p(v \mid s,h)$, $p(s \mid v,h)$, $P(h \mid v)$ and $p(v \mid h)$ from which we can gain some insight into the properties of the ssRBM. The strategy we will adopt is to derive the conditionals neglecting the $\mathbb{U}(v;R)$ factor, then during sampling we can correct for the omission via rejection sampling. This turns out to be very efficient as the number of rejections is expected to be very low as we will discuss later in section 4.

Let us first consider the conditional distribution $p(v \mid s,h)$. Taking into account the bounded domain of v, we have $p(v \mid s,h,\|v\|_2 > R) = 0$ and:

$$\begin{aligned} p(v \mid s,h,\|v\|_2 \leq R) &= \frac{1}{p(s,h)}\frac{1}{Z}\exp\{-E(v,s,h)\} \\ &= \frac{1}{B}\mathcal{N}\left(\Lambda^{-1}\sum_{i=1}^{N}W_i s_i h_i\ ,\ \Lambda^{-1}\right), \end{aligned}$$

where B is determined by integrating the Gaussian $\mathcal{N}\left(\Lambda^{-1}\sum_{i=1}^{N}W_i s_i h_i\ ,\ \Lambda^{-1}\right)$ over the ball $\|v\|_2 \leq R$. By isolating all terms involving v, the remaining terms are constant with respect to v and therefore the conditional distribution $p(v \mid s,h)$ has the form of a simple (truncated) Gaussian distribution and since the off-diagonal terms of the covariance are all zero, sampling from this Gaussian is straightforward, when using rejection sampling to exclude v outside the bounded domain. For convenience, we will adopt the notation $p^*(v \mid s,h)$ to refer to the un-truncated Gaussian distribution associated with $p(v \mid s,h)$; i.e., $p^*(v \mid s,h) = \mathcal{N}\left(\Lambda^{-1}\sum_{i=1}^{N}W_i s_i h_i\ ,\ \Lambda^{-1}\right)$

It is instructive to consider what happens if we do not assume we know s, i.e., considering the form of the distribution $p(v \mid h)$ where we marginalize out s:

$$\begin{aligned} p(v \mid h,\|v\|_2 \leq R) &= \frac{1}{P(h)}\frac{1}{Z}\int \exp\{-E(v,s,h)\}\ ds \\ &= \frac{1}{B}\mathcal{N}\left(0\ ,\ \left(\Lambda - \sum_{i=1}^{N}h_i W_i \alpha_i^{-1} W_i^T\right)^{-1}\right) \end{aligned} \tag{3}$$

The last equality holds only if the covariance matrix $\left(\Lambda - \sum_{i=1}^{N}h_i W_i \alpha_i^{-1} W_i^T\right)^{-1}$ is positive definite. By marginalizing over the "slab" variates, s, the visible vector v remains (truncated) Gaussian distributed, however the parametrization has changed significantly as a function of h. The distribution $p^*(v \mid s,h)$ uses h with s to parametrize the conditional mean, whereas in the case of $p^*(v \mid h)$, h parametrizes the conditional covariance. Another critical difference between these two distributions over v is that the covariance matrix of the Gaussian $p^*(v \mid h)$ is not diagonal. As such, sampling from $p^*(v \mid h)$ is potentially computationally intensive for large v as it would require a matrix inverse for every weight update. Fortunately, we will have no need to sample from $p^*(v \mid h)$.

We now turn to the conditional $p(s_i \mid v,h)$. The derivation is analogous to that leading to Eq. 3. The conditional $p(s \mid v,h)$ is Gaussian-distributed:

$$p(s \mid v,h) = \prod_{i=1}^{N}\mathcal{N}\left(h_i\alpha_i^{-1}W_i^T v\ ,\ \alpha_i^{-1}\right)$$

Here again, we see that the conditional distribution over s given v and h possess a diagonal covariance enabling simple and efficient sampling of s from this conditional distribution. The form of $p(s \mid v,h)$ indicates that, given $h_i = 1$, the expected value of s_i is linearly dependent of v.

Similar to $p(v \mid h)$, the distribution $p(h \mid v)$ is obtained by marginalizing out the slab variable s:

$$\begin{aligned} P(h_i = 1 \mid v) &= \frac{1}{p(v)}\frac{1}{Z_i}\int \exp\{-E(v,s,h)\}\ ds \\ &= \mathrm{sigm}\left(\frac{1}{2}v^T W_i \alpha_i^{-1} W_i^T v + b_i\right), \end{aligned} \tag{4}$$

[1] It is perhaps more natural to consider a scalar s_i, i.e., $K = 1$; however generalizing to vector valued s_i allows us to naturally implement a form of "sum" pooling.

where sigm represents a logistic sigmoid. As with the conditionals $p(v \mid s, h)$ and $p(s \mid v, h)$, the distribution of h given v factorizes over the elements of h. As a direct consequence of the marginalization of s, the influence of v on $P(h_i \mid v)$ is controlled by a term quadratic in $v^T W_i$, meaning that h_i is active when v exhibits significant "variance" in the direction of W_i.

A choice of data representations: The spike and slab RBM is somewhat unusual in that the use of dual latent variables, one continuous, and one binary, offers us a choice of data representations, to be used in the particular task at hand. One option is to marginalize over s, and use the binary h or its expectation $P(h \mid v)$ as the data representation. Another option is to use $[s_1 h_1, \dots, s_N h_N]$ or $[\|s_1\| h_1, \dots, \|s_N\| h_N]$ or the corresponding expectations. These options possess the property that, for active units, the model representation is equivariant to the intensity of the input variable (within the bounded domain). This is a property shared with the rectified linear units of Nair and Hinton [14], and is thought to be potentially beneficial in a range of vision tasks as it offers superior robustness to variations in image intensity.

4 ssRBM Learning and Inference

As is typical of RBM-style models, learning and inference in the ssRBM is dependent on the ability to efficiently draw samples from the model via Markov chain Monte Carlo (MCMC). Inspection of the conditionals $P(h \mid v)$, $p(v \mid h)$, $p(s \mid v, h)$ and $p(v \mid s, h)$ reveals some important property of the ssRBM model. First, let us consider the standard RBM sampling scheme of iterating between $P(h \mid v)$ and $p(v \mid h)$ with s marginalized out. Sampling from $P(h \mid v)$ is straightforward, as equation 4 indicates that the h_i are all independent given v. Under the assumption of a positive definite covariance matrix, the conditional distribution $p(v \mid h)$ is multivariate Gaussian with non-diagonal covariance: $\left(\Lambda - \sum_{i=1}^N h_i W_i \alpha_i^{-1} W_i^T\right)^{-1}$. Thus sampling from $p(v \mid h)$ requires the inversion of the covariance matrix with every weight update. For large input dimensionality D, this presents a challenging setting for learning. Fortunately, we need not sample from $p(v \mid h)$ directly, instead we can instantiate the slab variable s by sampling from $p(s \mid h, v)$ and then, given these s samples and the h sampled from $P(h \mid v)$, we can sample v from the conditional $p(v \mid s, h)$. Both these conditionals are Gaussian with diagonal covariance leading to simple and efficient sampling.

Taken all together the triplet $P(h \mid v)$, $p(s \mid v, h)$ and $p(v \mid s, h)$ form the basis of a block-Gibbs sampling scheme that allows us to sample efficiently from the ssRBM. Whenever a sample of v falls outside the ball $\|v\|_2 \le R$, we reject and resample from the conditional $p(v \mid s, h)$. The data likelihood gradient is

$$\begin{aligned}\frac{\partial}{\partial \theta_i}\left(\sum_{t=1}^T \log p(v_t)\right) = & - \sum_{t=1}^T \left\langle \frac{\partial}{\partial \theta_i} E(v_t, s, h) \right\rangle_{p(s,h|v_t)} \\ & + T \left\langle \frac{\partial}{\partial \theta_i} E(v, s, h) \right\rangle_{p(v,s,h)},\end{aligned}$$

i.e., of the same form as for a standard RBM, only with the expectations over $p(s, h \mid v_t)$ in the "clamped" condition, and over $p(v, s, h)$ in the "unclamped" condition. In training, we follow the stochastic maximum likelihood algorithm (also know as persistent contrastive divergence) [28, 23], i.e., performing only one or few updates of an MCMC chain between each parameter update.

The expectation of the gradient with respect to W_i in the "clamped" condition (also called the positive phase) is:

$$\left\langle \frac{\partial}{\partial W_i} E(v_t, s, h) \right\rangle_{p(s,h|v_t)} = -v_t \left(\mu_{i,t}^+\right)^T \hat{h}_{i,t}^+.$$

Here $\hat{h}_{i,t}^+ = p(h_i \mid v_t)$ and $\mu_{i,t}^+$ is the mean of the Gaussian density $p(s_i | h_i = 1, v_t)$. In the "unclamped" condition (negative phase) the expectation of the gradient with respect to W_i is given by:

$$\left\langle \frac{\partial}{\partial W_i} E(v, s, h) \right\rangle_{p(v,s,h)} \approx \frac{1}{M} \sum_{m=1}^M -\tilde{v}_m (\mu_{i,m}^-)^T \hat{h}_{i,m}^-.$$

Where $\hat{h}_{i,m}^- = p(h_i \mid \tilde{v}_m)$ and $\mu_{i,m}^-$, is the mean of the Gaussian density $p(s_i \mid h_i = 1, \tilde{v}_m)$. The $\tilde{v}_m$ are samples drawn from the model via Gibbs sampling. The expectation of the gradient with respect to b_i is identical to that of the GRBM. Finally, the expectation of the gradient with respect to Λ is given by:

$$\begin{aligned}\left\langle \frac{\partial}{\partial \Lambda} E(v_t, s, h) \right\rangle_{p(s,h|v_t)} &= \frac{1}{2} v_t^T v_t \\ \left\langle \frac{\partial}{\partial \Lambda} E(v, s, h) \right\rangle_{p(v,s,h)} &\approx \frac{1}{M} \sum_{m=1}^M \frac{1}{2} \tilde{v}_m^T \tilde{v}_m\end{aligned}$$

One could also imagine updating α to maximize likelihood; however in our experiments we simply treated α as a hyper-parameter.

As previously discussed, without the $\mathbb{U}(v; R)$ term in the joint density, the spike and slab model is not parametrized to guarantee that the model constitutes a well defined probability model with a finite partition function. To draw samples from the model, we rely on a rejection sampling scheme based on $\mathbb{U}(v; R)$. However, during training, we instead rely on a very important property of the likelihood gradient to suppress samples from the model that are drawn in regions of the data space unsupported by nearby data examples. As the parameters are updated, the model may approach instability. If this occurs, negative phase or "unclamped" samples are naturally drawn to the direction of the instability (i.e., outside the range of the training data) and through their influence act to return the model to a locally stable region of operation. Due to this stabilizing property of learning, we actually do not include the $\mathbb{U}(v; R)$ term to the joint likelihood during learning. Practically, training the model is straightforward provided the model is initialized in a stable regime of the parameter space. For example, the values of α and Λ must be sufficiently large to at least offset the initial values of W. We also use a decreasing learning rate that also helps maintain the model in a stable region of the parameter space. Training this way also ensures that the natural parametrization of the ssRBM (excluding the $\mathbb{U}(v; R)$) is almost always sufficient to ensure stability during sampling and renders our rejection sampling strategy highly efficient.

5 Comparison to Previous Work

There exist a number of papers that aim to address the issue of modeling natural images in the RBM context. The most relevant of these are the Product of Student's T-distribution (PoT) model [25] and the mean and covariance Restricted Boltzmann Machine (mcRBM) [17]. However before reviewing these models and their connections to the ssRBM, we note that the idea of building Boltzmann Machines with products of binary and continuous-valued variables was discussed in [26], [29], and [6]. We also note that the covariance structure of the ssRBM conditional $p(v \mid h)$ (equation 3) is essentially identical to the product of probabilistic principal components analysis (PoPPCA) model [27] with components corresponding to the ssRBM weight vectors associated with the active hidden units ($h_i = 1$).

5.1 Product of Student's T-distributions

The product of Student's T-distributions model [25] is an energy-based model where the conditional distribution over the visible units conditioned on the hidden variables is a multivariate Gaussian (non-diagonal covariance) and the complementary conditional distribution over the hidden variables given the visibles are a set of independent Gamma distributions. The PoT model is similar to our model in that it characterizes the covariance of real-valued inputs with real-valued hidden units, but in the case of the PoT model, the real-valued hidden units are Gamma-distributed rather than Gaussian-distributed as is the case for the ssRBM.

The most significant difference between the ssRBM and the PoT model is how they parametrize the covariance of the multivariate Gaussian over the visible units ($p(v \mid h)$ in the case of the ssRBM, equation 3). While the ssRBM characterizes the covariance as $\left(\Lambda - \sum_{i=1}^N h_i W_i \alpha_i^{-1} W_i^T\right)^{-1}$, the PoT model parametrized the conditional covariance as $\left(\sum_{i=1}^N u_i W_i W_i^T\right)^{-1}$, where the u_i are the Gamma-distributed latent variables. The PoT latent variables use their activation to maintain constraints, decreasing in value to allow variance in the direction of the corresponding weight vector. The spike and slab h_i variables use their activation to pinch the precision matrix along the direction specified by the corresponding weight vector. The two models diverge when the dimensionality of the hidden layer exceeds that of the input. 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. This is a property the ssRBM shares with sparse coding models [16, 7] where the sparse latent representation also encodes directions of variance above a nominal value. An over-complete PoT model has a different interpretation: with an over-complete set of constraints, variation of the input along a particular direction would require decreasing potentially all constraints with positive projection in that direction.

5.2 The Mean and Covariance RBM

One recently introduced and particularly successful approach to modeling real-valued data is the mean and covariance RBM. The mcRBM is a restricted Boltzmann machine designed to explicitly model both the mean and covariance of elements of the input. The mcRBM combines a variant of the earlier covariance RBM (cRBM) model [18] with a GRBM to capture the conditional "mean". Because of some surprising similarities between the cRBM and the ssRBM, we will review the cRBM in some detail.

We take the number of cRBM hidden units to be N_c: $h^c \in \{0,1\}^{N_c}$, and the dimensionality of the visible vector to be D: $v \in \mathbb{R}^D$. The cRBM model is defined via the energy function:

$$E^c(v, h^c) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} P_{ki} h_i^c \left(v^T C_{:,k}\right)^2 - \sum_{i=1}^{N} b_i^c h_i^c, \quad (5)$$

where P is a pooling matrix with non-positive elements($p \in \mathbb{R}^{K\times N}$), N is the number of hidden units, $C_{:,k}$ is the weight vector k ($C \in \mathbb{R}^{D\times K}$) and b^c is a vector of biases. Defining the energy function in this way allows one to derive the pair of conditionals for h and v respectively as:

$$P(h_i^c = 1 \mid v) = \mathrm{sigm}\left(\frac{1}{2}\sum_{k=1}^{K} P_{ki} h_i^c \left(v^T C_{:,k}\right)^2 - b_i^c\right),$$
$$p(v \mid h^c) = \mathcal{N}\left(0, \left(C \operatorname{diag}(Ph^c) C^T\right)^{-1}\right), \qquad (6)$$

where $\operatorname{diag}(v)$ is the diagonal matrix with vector v in its diagonal. That is, the conditional Gaussian distribution possess a non-diagonal covariance.

In relation to the ssRBM, the first thing to note about the cRBM is the similarity of the conditional for the binary latent variable, $P(h \mid v)$ in the case of the ssRBM (equation 4) and $P(h^c \mid v)$ in the case of the cRBM. Simplifying both models to pool over a single variable (setting the P matrix to the negative identity in the case of the cRBM and $K = 1$ in the ssRBM), both conditionals contain a $\frac{1}{2}(v^T W)^2$ term (with $C \equiv W$) and a constant bias. Remarkably, this occurs despite the two models sharing relatively little in common at the level of the energy function.

Despite the similarity in the conditional distribution over the binary latent variables, the two models diverge in their expressions for the complementary conditions over the visible variable v given the binary latents (comparing equations 3 and 6). While the the ssRBM parametrizes the covariance as $\left(\Lambda - \sum_{i=1}^{N} h_i W_i \alpha_i^{-1} W_i^T\right)^{-1}$; the cRBM parametrizes the covariance as $\left(C \operatorname{diag}(Ph^c) C^T\right)^{-1}$. Similar to the PoT model, the cRBM encodes the conditional covariance as a series of constraints to be actively enforced. As is the case for the PoT model, we suggest that this form of parametrization is not well suited to heavily over-complete models.

Despite different parametrizations of the conditional covariance, the ssRBM and the cRBM share the property that the conditional distribution over v given their respective binary latent variables is multivariate Gaussian with a non-diagonal covariance. In the ssRBM, we have recourse to a simple diagonal-covariance Gaussian conditional over v by instantiating the slab variables s, but there is no equivalent recourse for the cRBM. As a result, the cRBM and the mcRBM are not amenable to the kind of block Gibbs sampling available to the ssRBM and to more standard RBMs (a large matrix inversion would be required for each Gibbs step). In training the cRBM, samples are drawn using hybrid Monte Carlo (HMC) [15]. As an MCMC sampler, HMC has been shown to be very effective for some problems, but it suffers from a relatively large number of hyper-parameters that must be tuned to yield well-mixing samples from the target distribution.

The mcRBM combines a GRBM with a cRBM such that there are two kinds of hidden units, mean units h^m and covariance units h^c. The combined energy function of the mcRBM is given by:

$$E(v, h^c, h^m) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K} P_{ki} h_i^c \left(\frac{v^T}{\|v\|}\frac{C_{:,k}}{\|C_k\|}\right)^2$$
$$-\sum_{i=1}^{N} b_i^c h_i^c + \frac{1}{2} v^T v - \sum_{j=1}^{M} v^T W_{:,j} h_j^m - \sum_{j=1}^{M} b_j^m h_j^m$$

The mcRBM is not entirely equivalent to the combination of a cRBM and a GRBM, as its energy function includes a normalization of both the C_k weight vectors and the visible vector (to increase the robustness of the model to large contrast variations in the input image).

In deriving the conditionals for the ssRBM, we saw that by manipulating how we treat the slab variables s, we could fluidly move between modeling the conditional mean and modeling the conditional covariance. From this perspective it is revealing to think about the combination of the GRBM with the cRBM in the mcRBM. One can think about an equivalent model, within the spike and slab framework, where we take a subset of the ssRBM latent units and marginalize over the corresponding slab variables s – these unit would encode the conditional covariance. With the remaining units we model the equivalent conditional mean by imposing the constraint $s_i = \mathbf{1}$.

6 Experiments

We have run simulations with Theano [1] to illustrate three key ideas related to the ssRBM model: (a) it learns appealing filters to model natural images, (b) the spike variables are meaningfully used in a trained model, and (c) the latent image representation induced by the ssRBM makes the ssRBM a drop-in upgrade of the similar GRBM and cRBM models on CIFAR-10 image-labeling, and is competitive with the more complicated mcRBM.

6.1 Filters

The ssRBM learned qualitatively similar filters in the pooled and un-pooled models, but the pooling induced interesting structure to the set of filters.

Figure 2 illustrates the filters learned by an un-pooled ($K = 1$) ssRBM from a large number (one million) of PCA-whitened 8x8 RGB image patches drawn from the TinyImages dataset [24]. PCA-whitening retained 99% of the variance with 74 dimensions. These filters were obtained by stochastic maximum likelihood learning with the learning rate set to 10^{-4} for 20 000 training iterations using minibatches of size 128. Af-

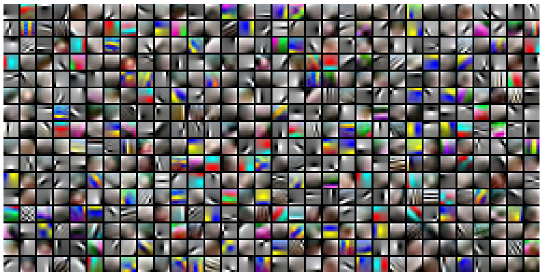

Figure 2: Filters learned by the unpooled ssRBM when applied to PCA-whitened 8x8 color image patches. Note how some filters care about color while others surprisingly do not, achieving a form of disentangling of color information from shape information.

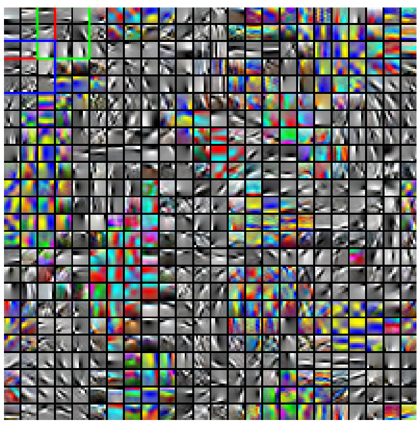

Figure 3: Filters learned by a pooled spike and slab RBM with topographically overlapping pools applied to PCA-whitened 8x8 color image patches. Pooling was done across 3x3 groups ($K = 9$) units s giving rise to a degree of continuity across the set of filters. Again, color and grey-level filters emerge separately.

ter 20 000 iterations the learning rate was reduced in inverse proportion to the iteration number. No sparsification or regularization was applied to the activations or model parameters. α was fixed to 1.5, the bias was initialized to -1, the weights were initialized from a zero-mean Gaussian with variance 10^{-4}.

Figure 3 illustrates the effect of pooling $K = 9$ scale variables s with each h. The pinwheel-like pattern was obtained by sharing columns $W_{:,i}$ between pools using the sort of topographic map used in [17]. Clean backgrounds in each filter were obtained by applying a small (10^{-4}) ℓ_1 penalty to the filter weights. All filters were brought into play by applying a small (.2) ℓ_1 penalty pushing each unit h_i to have a marginal mean of .1. The topographic map down-weighted the effective magnitude of each W column, so the initial range and learning rate on W were raised accordingly.

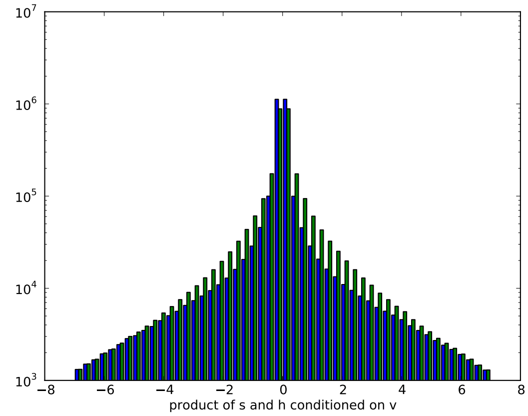

Figure 4: The spike and slab effect. In green is the marginal (over a large number of images) distribution over all s_i variables given $h_i = 1$, and in blue is the marginal distribution over all $s_i h_i$ products. The vertical axis is a log-scaled frequency of occurrence.

6.2 The Effect of Spike Variables

Figure 4 illustrates the effect of the binary spike variables (h). The effect of h_i is to suppress the influence of filter $W_{:,i}$ when the filter response is weak. Once h has been inferred from an observation v it induces a Gaussian conditional joint distribution $p(s, v \mid h)$ as well as a Gaussian conditional marginal $p(v \mid h)$ in which the covariance is determined by the filters that were unusually active.Figure 4 shows that the spike variables are indeed often 0, and eliminating potential directions of covariance in the conditional marginal.

6.3 Learning Features for Classification

To evaluate the latent variables of the ssRBM as features for object classification we adopted the testing protocol of [17] which looked at performance on CIFAR-10. CIFAR-10 comprises 40 000 training images, 10 000 validation images, and 10 000 test images. The images are 32-by-32 pixel RGB images. Each image is labeled with one of ten object categories (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck) according to the most prominent object in the image. We produced image features from an ssRBM model trained on patches using the same procedure as [17] - a 7-by-7 grid of (overlapping) 8-by-8 pixel patches was used to extract a 7-by-7 grid of mean-values for h. The ssRBM had N h variables, so the concatenation of the h vectors from each grid location yielded $49N$ features. We classified this feature vector using logistic regression, and we also experimented with backpropagating the error gradient into the ssRBM parameters and fine-tuning this "feature extractor" as if it and the classifier together were a single neural network.

Model	Classification Rate (%)
mssRBM (finetuned)	69.9 $\pm$ 0.9
mssRBM	68.7 $\pm$ 0.9
mcRBM	68.2 $\pm$ 0.9
ssRBM (finetuned)	69.2 $\pm$ 0.9
ssRBM	67.6 $\pm$ 0.9
cRBM (900 factors)	64.7 $\pm$ 0.9
cRBM (225 factors)	63.6 $\pm$ 0.9
GRBM	59.7 $\pm$ 1.0

Table 1: The performance of the pooled and unpooled ssRBM models relative to others on CIFAR-10. Confidence intervals are at 95% level. The mssRBM model is an ssRBM with 81 s units fixed at 1. The GRBM, cRBM and mcRBM results are copied from [17]

We optimized hyper-parameters for this task by drawing 200 hyper-parameter assignments randomly from a grid, performing 50 000 and 200 000 unsupervised training iterations, measuring classification error, and sorting all these unsupervised models by the validation set performance of the $P(h|v)$ feature vector. The random grid included variations in the number of unsupervised learning iterations (50K, 200K), learning rate (.0003, .0001), initial Λ (10, 15, 20), number of latent h variables N (100, 200,400), number of pooled s variables K per h (1,2,3), initial range for W (.05, .1, .2), initial bias on each h_i (-5, -4, -3), target sparsity for each h (.05, .1, .2), weight of sparsity regularization (0, .1, .2, .4). The initial value of α was fixed to 10.5. The best results with and without fine-tuning of the ssRBM weight matrix are given in Table 6.3 along with selected other results from the literature.

We also experimented with "mean" units as in [17] by adding $s_i h_i$ pairs in which the s_i were fixed to 1. Reusing the best-performing hyper-parameters, we simply added 81 mean units and repeated the training procedure. As in [17] we found that these additional mean units improved the performance of the model for classification beyond what was found by adding additional normal (unclamped) hidden units. This result, that a hidden layer consisting of a mix of mean and pooled units is better than either one alone, suggests that models with heterogenous latent states represent an interesting direction for future work. Indeed superior classification performance has been demonstrated by stacked binary RBMs on top of the mcRBM [17] (71.0%). In very recent work, other kinds of models with high accuracy have been advanced: a 4000-component patch-wise k-means [5] (79.6%), and an 8-layer neural network training on artificial translations of the data [4] (80.49%). However, convolutional training of Deep Belief Networks [9] (78.9%) has proved effective and we expect the ssRBM to be similarly improved by additional layers and convolutional training.

7 Discussion

In this paper we introduce a new *spike and slab* RBM model, which has a binary spike variable and a continuous slab variable associated with each hidden unit. These slab variables allow the model to capture covariance information while maintaining simple and efficient inference via a Gibbs sampling scheme.

Despite the similarity in the conditional distributions over the hidden binary variables between the ssRBM and the cRBM, there are a number of important distinctions. First, the ssRBM is amenable to Gibbs sampling whereas when sampling from the cRBM one must resort to hybrid Monte Carlo (HMC). While HMC is a practical algorithm for sampling in the RBM framework, the simplicity of Gibbs makes the ssRBM a more attractive option as a building block for more ambitious models such as the deep Boltzmann machine [19] and time-series models [22]. Another difference between the ssRBM and the cRBM is that the ssRBM induces sparse real-valued representations of the data. In our limited experiments using this data representation, we have not found it to be superior to using only $P(h \mid v)$, however recent work [14] has demonstrated the importance of sparse real-valued outputs in achieving superior classification performance.

As discussed previously, without any restriction on either the visible layer domain or the binary hidden unit combinations, the energy of the spike and slab model is not guaranteed to define a valid probability distribution. In practice this is fairly easily dealt with by imposing either a bounded domain on v, as we have done, or by applying a global penalty that is flat in the region of the training data and outside that region grows sufficiently fast to overcome any negative growth arising from the energy function (i.e., the term $-\sum_{i=1}^{N} v^T W_i s_i h_i$). As an alternative, one could restrict the covariance of $p(v \mid h)$ to remain positive definite and reject patterns of hidden unit activations that violate this constraint. Under the mixture model interpretation of the RBM, this approach may be interpreted as zeroing out the mixture components that violate the requirement that the mixture components be individually normalizable.

Acknowledgements

We acknowledge NSERC, FQRNT, RQCHP and Compute Canada for their financial and computational support; Chris Williams for pointing out the connection between the PoPPCA model and the ssRBM; and Charlie Tang for correcting a typo in an earlier manuscript.

References

[1] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *SciPy*, 2010.

[2] M. A. Carreira-Perpiñan and G. E. Hinton. On contrastive divergence learning. In *AISTATS 10*, pages 33–40, 2005.

[3] H. Chen and A. F. Murray. A continuous restricted Boltzmann machine with an implementable training algorithm. *IEE P-Vis. Image Sign.*, 150(3):153–158, 2003.

[4] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. High-performance neural networks for visual object classification. arXiv, February 2011.

[5] A. Coates, H. Lee, and A. Y. Ng. An analysis of single-layer networks in unsupervised feature learning. In *AISTATS 14*, 2011.

[6] Y. Freund and D. Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. Technical Report UCSC-CRL-94-25, University of California, Santa Cruz, 1994.

[7] R. Grosse, R. Raina, H. Kwong, and A. Y. Ng. Shift-invariant sparse coding for audio classification. In *UAI 2007*, 2007.

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

[9] A. Krizhevsky. Convolutional deep belief networks on cifar-10. Technical report, University of Toronto, Aug 2010.

[10] A. Krizhevsky and G. E. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[11] H. Larochelle and Y. Bengio. Classification using discriminative restricted Boltzmann machines. In *ICML 25*, pages 536–543. ACM, 2008.

[12] T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *J. Amer. Statistical Assoc.*, 83(404):1023–1032, 1988.

[13] V. Nair and G. E. Hinton. Implicit mixtures of restricted boltzmann machines. In *NIPS 21*, pages 1145–1152. MIT Press, 2009.

[14] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML 27*, pages 807–814. ACM, 2010.

[15] R. M. Neal. *Bayesian Learning for Neural Networks*. PhD thesis, Dept. of Computer Science, University of Toronto, 1994.

[16] B. A. Olshausen and D. J. Field. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vision Research*, 37:3311–3325, 1997.

[17] M. Ranzato and G. H. Hinton. Modeling pixel means and covariance using factorized third-order boltzmann machines. In *CVPR*, pages 2551–2558. IEEE Press, 2010.

[18] M. Ranzato, A. Krizhevsky, and G. E. Hinton. Factored 3-way restricted boltzmann machines for modeling natural images. In *AISTATS 13*, pages 621–628, 2010.

[19] R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *AISTAT 12*, pages 448–455, 2009.

[20] R. Salakhutdinov, A. Mnih, and G. E. Hinton. Restricted Boltzmann machines for collaborative filtering. In *ICML 24*, pages 791–798. ACM, 2007.

[21] I. Sutskever, G. E. Hinton, and G. Taylor. The recurrent temporal restricted boltzmann machine. In *NIPS 21*, pages 1601–1608. MIT Press, 2009.

[22] G. Taylor and G. E. Hinton. Factored conditional restricted Boltzmann machines for modeling motion style. In *ICML 26*, pages 1025–1032. ACM, 2009.

[23] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *ICML 25*, pages 1064–1071. ACM, 2008.

[24] A. Torralba, R. Fergus, and W. T. Freeman. 80 million tiny images: A large dataset for non-parametric object and scene recognition. *IEEE T. Pattern Anal.*, 30(11):1958–1970, 2008.

[25] M. Welling, G. E. Hinton, and S. Osindero. Learning sparse topographic representations with products of Student-t distributions. In *NIPS 15*, pages 1359–1366. MIT Press, 2003.

[26] C. K. I. Williams. Continuous-valued boltzmann machines. Unpublished Manuscript, March 1993.

[27] C. K. I. Williams and F. V. Agakov. Products of Gaussians and Probabilistic Minor Component Analysis. *Neural Computation*, 14(5):1169–1182, 2002.

[28] L. Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. In *Stochastics and Stochastics Models*, pages 177–228, 1998.

[29] R. S. Zemel, C. K. I. Williams, and M. Mozer. Directional-unit boltzmann machines. In *NIPS 5*, pages 172–179. MIT Press, 1993.