Journal of Machine Learning Research 15 (2014) 2949-2980 Submitted 6/13; Revised 3/14; Published 9/14

# Multimodal Learning with Deep Boltzmann Machines

**Nitish Srivastava** NITISH@CS.TORONTO.EDU
*Department of Computer Science*
*University of Toronto*
*10 Kings College Road, Rm 3302*
*Toronto, Ontario, M5S 3G4, Canada.*

**Ruslan Salakhutdinov** RSALAKHU@CS.TORONTO.EDU
*Department of Statistics and Computer Science*
*University of Toronto*
*10 Kings College Road, Rm 3302*
*Toronto, Ontario, M5S 3G4, Canada.*

**Editor:** Aapo Hyvarinen

## Abstract

Data often consists of multiple diverse modalities. For example, images are tagged with textual information and videos are accompanied by audio. Each modality is characterized by having distinct statistical properties. We propose a Deep Boltzmann Machine for learning a generative model of such multimodal data. We show that the model can be used to create fused representations by combining features across modalities. These learned representations are useful for classification and information retrieval. By sampling from the conditional distributions over each data modality, it is possible to create these representations even when some data modalities are missing. We conduct experiments on bi-modal image-text and audio-video data. The fused representation achieves good classification results on the MIR-Flickr data set matching or outperforming other deep models as well as SVM based models that use Multiple Kernel Learning. We further demonstrate that this multimodal model helps classification and retrieval even when only unimodal data is available at test time.

**Keywords:** Boltzmann machines, unsupervised learning, multimodal learning, neural networks, deep learning

## 1. Introduction

Information in the real world comes through multiple input channels. Images are associated with captions and tags, videos contain visual and audio signals, sensory perception includes simultaneous inputs from visual, auditory, motor and haptic pathways. Each modality is characterized by very distinct statistical properties which makes it difficult to disregard the fact that they come from different input channels. Useful representations can potentially be learned for such data by combining the modalities into a joint representation that captures the real-world concept that the data corresponds to. For example, we would like a probabilistic model to correlate the occurrence of the words 'oak tree' and the visual properties of an image of an oak tree and represent them jointly, so that the model assigns high probability to one conditioned on the other.

Before we describe our model in detail, it is useful to understand why building such models is important. Different modalities can act like soft labels for each other. For example, consider bi-modal image-text data. If the same uncommon word was used in the context of several images, then there is some chance that all those images represent the same object. Conversely, if different words are used to describe similar looking images, then those words might mean the same thing. In other words, one modality might be somewhat invariant to large changes in another modality. This provides a rich learning signal. Moreover, different modalities typically carry different kinds of information. For example, people often caption an image to say things that may not be obvious from the image itself, such as the name of the person, place, or a particular object in the picture. Unless we do multimodal learning, it would not be possible to discover a lot of useful information about the world. We cannot afford to have discriminative models for every single concept. It would be useful, or at least elegant, to extract this information from unlabelled data.

In a multimodal setting, data consists of multiple modes, each modality having a different kind of representation and correlational structure. For example, text is usually represented as discrete sparse word count vectors, whereas an image is represented using pixel intensities or outputs of feature extractors which are real-valued and dense. Having very different statistical properties makes it much harder to discover relationships across modalities than relationships among features in the same modality. There is a lot of structure in the data but it is difficult to discover the highly non-linear relationships that exist between low-level features across different modalities. Moreover, the data is typically very noisy and there may be missing values.

A good multimodal learning model must satisfy certain properties. The joint representation must be such that similarity in the representation space implies similarity of the corresponding concepts so that the representation is useful for classification and retrieval. It is also desirable that the joint representation be easy to obtain even in the absence of some modalities. It should also be possible to fill-in missing modalities given the observed ones.

Our proposed multimodal Deep Boltzmann Machine (DBM) model satisfies the above desiderata. DBMs (Salakhutdinov and Hinton, 2009b) are undirected graphical models with bipartite connections between adjacent layers of hidden units. The key idea is to learn a joint density model over the space of multimodal inputs. Missing modalities can then be filled-in by sampling from the conditional distributions over them given the observed ones. For example, we use a large collection of user-tagged images to learn a joint distribution over images and text $P(\mathbf{v}_{img}, \mathbf{v}_{txt}; \theta)$. By drawing samples from $P(\mathbf{v}_{txt}|\mathbf{v}_{img}; \theta)$ and from $P(\mathbf{v}_{img}|\mathbf{v}_{txt}; \theta)$ we can fill-in missing data, thereby doing image annotation and image retrieval respectively, some of examples of which are shown in Figure 1.

There have been several approaches to learning from multimodal data. In particular, Huiskes et al. (2010) showed that using captions or tags, in addition to standard low-level image features significantly improves classification accuracy of Support Vector Machines (SVM) and Linear Discriminant Analysis (LDA) models. A similar approach of Guillaumin et al. (2010) based on the multiple kernel learning framework further demonstrated that an additional text modality can improve the accuracy of SVMs on various object recognition tasks. However, all of these approaches are discriminative by nature and cannot make use of large amounts of unlabelled data or deal easily with missing input modalities.

| Image | Given Tags | Generated Tags | Input Tags | Nearest neighbors to generated image features |
|---|---|---|---|---|
|  | pentax, k10d, kangarooisland, southaustralia, sa, 300mm, australia, australiansealion | beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves | nature, hill, scenery, green, clouds |   |
|  | < no text > | night, lights, christmas, nightshot, nacht, nuit, notte, longexposure, noche, nocturna | flower, nature, green, flowers, petal, petals, bud |   |
|  | aheram, 0505, sarahc, moo | portrait, bw, balckandwhite, people, faces, girl, blackwhite, person, man | blue, red, art, artwork, painted, paint, artistic, surreal, gallery, bleu |   |
|  | unseulpixel, naturey crap | fall, autumn, trees, leaves, foliage, forest, woods, branches, path | bw, blackandwhite, noiretblanc, bianconero, blancoynegro |   |

Figure 1: **Left:** Examples of text generated from a Deep Boltzmann Machine by sampling from $P(\mathbf{v}_{txt}|\mathbf{v}_{img};\theta)$. **Right:** Examples of images retrieved using features generated from a Deep Boltzmann Machine by sampling from $P(\mathbf{v}_{img}|\mathbf{v}_{txt};\theta)$.

On the generative side, Xing et al. (2005) used dual-wing harmoniums to build a joint model of images and text, which can be viewed as a linear Restricted Boltzmann Machine (RBM) model with Gaussian hidden units together with Gaussian and Poisson visible units. However, various data modalities will typically have very different statistical properties which makes it difficult to model them using shallow models. Most similar to our work is the recent approach of Ngiam et al. (2011) that used a deep autoencoder for speech and vision fusion. There are, however, several crucial differences. First, in this work we focus on jointly modelling very different data modalities: sparse word count vectors and real-valued dense image features. Second, we use a Deep Boltzmann Machine which is a probabilistic generative model as opposed to a feed-forward autoencoder. While both approaches have led to interesting results in several domains, using a generative model is important for applications we consider in this paper, as it allows our model to naturally handle missing and noisy data modalities.

## 2. Background: RBMs and Their Generalizations

Restricted Boltzmann Machines (RBMs) have been used effectively in modeling distributions over binary-valued data. Boltzmann machine models and their generalizations to exponential family distributions (Welling et al., 2005) have been successfully used in many application domains. For example, the Replicated Softmax model (Salakhutdinov and Hinton, 2009a) has been shown to be effective in modeling sparse word count vectors, whereas

Gaussian RBMs have been used for modeling real-valued inputs for image classification, video action recognition, and speech recognition (Lee et al., 2009; Taylor et al., 2010; Mohamed et al., 2011). In this section we briefly review these models, as they will serve as building blocks for our multimodal model.

### 2.1 Restricted Boltzmann Machines

A Restricted Boltzmann Machine (Smolensky, 1986) is an undirected graphical model with stochastic visible variables $\mathbf{v} \in \{0,1\}^D$ and stochastic hidden variables $\mathbf{h} \in \{0,1\}^F$, with each visible variable connected to each hidden variable. The model defines the following energy function $E : \{0,1\}^D \times \{0,1\}^F \to \mathbb{R}$

$$E(\mathbf{v}, \mathbf{h}; \theta) = -\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F} a_j h_j, \tag{1}$$

where $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}\}$ are the model parameters: $W_{ij}$ represents the symmetric interaction term between visible unit $i$ and hidden unit $j$; $b_i$ and $a_j$ are bias terms. The joint distribution over the visible and hidden units is defined by

$$P(\mathbf{v}, \mathbf{h}; \theta) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right), \quad \mathcal{Z}(\theta) = \sum_{\mathbf{v}}\sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right), \tag{2}$$

where $\mathcal{Z}(\theta)$ is the normalizing constant. The conditional distributions over hidden $\mathbf{h}$ and visible $\mathbf{v}$ vectors factorize and can be easily derived from Equations 1, 2 as

$$\begin{aligned} P(\mathbf{h}|\mathbf{v}; \theta) &= \prod_{j=1}^{F} p(h_j|\mathbf{v}), \quad \text{with} \quad p(h_j = 1|\mathbf{v}) = g\left(\sum_{i=1}^{D} W_{ij} v_i + a_j\right), \\ P(\mathbf{v}|\mathbf{h}; \theta) &= \prod_{i=1}^{D} p(v_i|\mathbf{h}), \quad \text{with} \quad p(v_i = 1|\mathbf{h}) = g\left(\sum_{j=1}^{F} W_{ij} h_j + b_i\right), \end{aligned}$$

where $g(x) = 1/(1+\exp(-x))$ is the logistic function. Given a set of observations $\{\mathbf{v}_n\}_{n=1}^{N}$, the derivative of the log-likelihood with respect to the model parameters can be obtained from Equation 2 as

$$\frac{1}{N}\sum_{n=1}^{N} \frac{\partial \log P(\mathbf{v_n}; \theta)}{\partial W_{ij}} = \mathbb{E}_{P_{\text{data}}}\left[v_i h_j\right] - \mathbb{E}_{P_{\text{model}}}\left[v_i h_j\right],$$

where $\mathbb{E}_{P_{\text{data}}}[\cdot]$ denotes an expectation with respect to the data distribution $P_{\text{data}}(\mathbf{h}, \mathbf{v}; \theta) = P(\mathbf{h}|\mathbf{v}; \theta) P_{\text{data}}(\mathbf{v})$, with $P_{\text{data}}(\mathbf{v}) = \frac{1}{N}\sum_n \delta(\mathbf{v} - \mathbf{v}_n)$ representing the empirical distribution, and $\mathbb{E}_{P_{\text{model}}}[\cdot]$ is an expectation with respect to the distribution defined by the model, as in Equation 2. We will sometimes refer to $\mathbb{E}_{P_{\text{data}}}[\cdot]$ as the *data-dependent expectation*, and $\mathbb{E}_{P_{\text{model}}}[\cdot]$ as the *model's expectation*.

### 2.2 Gaussian-Bernoulli RBM

RBMs were originally developed for modeling binary vectors. Gaussian-Bernoulli RBMs (Freund and Haussler, 1994; Hinton and Salakhutdinov, 2006) are a variant that can be used for modeling real-valued vectors such as pixel intensities and filter responses. Consider modeling visible real-valued units $\mathbf{v} \in \mathbb{R}^D$, and let $\mathbf{h} \in \{0,1\}^F$ be binary stochastic hidden units. The energy of the state $\{\mathbf{v}, \mathbf{h}\}$ of the Gaussian-Bernoulli RBM is defined as

$$E(\mathbf{v}, \mathbf{h}; \theta) = \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_{i=1}^{D}\sum_{j=1}^{F} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_{j=1}^{F} a_j h_j,$$

where $\theta = \{\mathbf{a}, \mathbf{b}, \mathbf{W}, \boldsymbol{\sigma}\}$ are the model parameters. The density that the model assigns to a visible vector $\mathbf{v}$ is given by

$$P(\mathbf{v}; \theta) = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right), \quad \mathcal{Z}(\theta) = \int_{\mathbf{v}} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}; \theta)\right) d\mathbf{v}.$$

Similar to the standard RBMs, the conditional distributions factorize as

$$\begin{aligned} P(\mathbf{h}|\mathbf{v}; \theta) &= \prod_{j=1}^{F} p(h_j|\mathbf{v}), \quad \text{with} \quad p(h_j = 1|\mathbf{v}) = g\left(\sum_{i=1}^{D} W_{ij} \frac{v_i}{\sigma_i} + a_j\right), \\ P(\mathbf{v}|\mathbf{h}; \theta) &= \prod_{i=1}^{D} p(v_i|\mathbf{h}), \quad \text{with} \quad v_i|\mathbf{h} \sim \mathcal{N}\left(b_i + \sigma_i \sum_{j=1}^{F} W_{ij} h_j, \sigma_i^2\right), \end{aligned} \tag{3}$$

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution with mean $\mu$ and variance $\sigma^2$. Observe that conditioned on the states of the hidden units (Equation 3), each visible unit is modeled by a Gaussian distribution, whose mean is shifted by the weighted combination of the hidden unit activations.

Given a set of observations $\{\mathbf{v}_n\}_{n=1}^N$, the derivative of the log-likelihood with respect to the model parameters takes a very similar form when compared to binary RBMs.

$$\frac{1}{N} \sum_{n=1}^{N} \frac{\partial \log P(\mathbf{v_n}; \theta)}{\partial W_{ij}} = \mathbb{E}_{P_{\text{Data}}}\left[\frac{v_i}{\sigma_i} h_j\right] - \mathbb{E}_{P_{\text{model}}}\left[\frac{v_i}{\sigma_i} h_j\right].$$

### 2.3 Replicated Softmax Model

The Replicated Softmax Model is useful for modeling sparse count data, such as word count vectors in a document (Salakhutdinov and Hinton, 2009a). This model is a type of Restricted Boltzmann Machine in which the visible variables, that are usually binary, have been replaced by multinomial one of a number of different states. Specifically, let $K$ be the dictionary size, $M$ be the number of words appearing in a document, and $\mathbf{h} \in \{0,1\}^F$ be binary stochastic hidden topic features. Let $\mathbf{V}$ be a $M \times K$ observed binary matrix with $v_{ik} = 1$ iff the multinomial visible unit $i$ takes on $k^{th}$ value (meaning the $i^{th}$ word in the document is the $k^{th}$ dictionary word). The energy of the state $\{\mathbf{V}, \mathbf{h}\}$ can be defined as

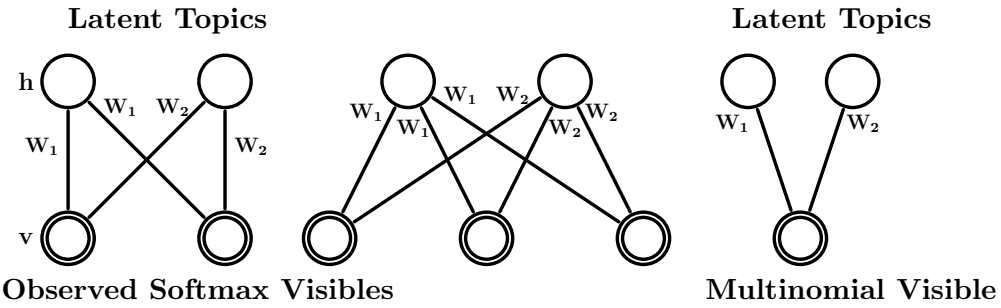



Figure 2: Replicated Softmax model. The top layer represents a vector $\mathbf{h}$ of stochastic, binary topic features and and the bottom layer represents softmax visible units $\mathbf{v}$. All visible units share the same set of weights, connecting them to binary hidden units. **Left:** The model for a document containing two and three words. **Right:** A different interpretation of the Replicated Softmax model, in which $M$ softmax units with identical weights are replaced by a single multinomial unit which is sampled $M$ times.

$$E(\mathbf{V},\mathbf{h}) = -\sum_{i=1}^{M}\sum_{j=1}^{F}\sum_{k=1}^{K} W_{ijk} v_{ik} h_j - \sum_{i=1}^{M}\sum_{k=1}^{K} b_{ik} v_{ik} - \sum_{j=1}^{F} a_j h_j,$$

where $\{\mathbf{W},\mathbf{a},\mathbf{b}\}$ are the model parameters: $W_{ijk}$ is a symmetric interaction term between visible unit $i$ that takes on value $k$, and hidden feature $j$, $b_{ik}$ is the bias of unit $i$ that takes on value $k$, and $a_j$ is the bias of hidden feature $j$. The probability that the model assigns to a visible binary matrix $\mathbf{V}$ is

$$P(\mathbf{V},\mathbf{h};\theta) = \frac{1}{\mathcal{Z}(\theta)}\exp\left(-E(\mathbf{V},\mathbf{h};\theta)\right), \quad \mathcal{Z}(\theta) = \sum_{\mathbf{V}}\sum_{\mathbf{h}}\exp\left(-E(\mathbf{V},\mathbf{h};\theta)\right).$$

The key assumption of the Replicated Softmax model is that for each document we create a separate RBM with as many softmax units as there are words in the document, as shown in Figure 2. Assuming that the order of the words can be ignored, all of these softmax units can share the same set of weights, connecting them to binary hidden units. In this case, the energy of the state $\{\mathbf{V},\mathbf{h}\}$ for a document that contains $M$ words is defined as

$$E(\mathbf{V},\mathbf{h}) = -\sum_{j=1}^{F}\sum_{k=1}^{K} W_{jk}\hat{v}_k h_j - \sum_{k=1}^{K} b_k \hat{v}_k - M\sum_{j=1}^{F} a_j h_j,$$

where $\hat{v}_k = \sum_{i=1}^{M} v_{ik}$ denotes the count for the $k^{th}$ word. Observe that the bias terms of the hidden variables are scaled up by the length of the document. This scaling is important as it allows hidden units to behave sensibly when dealing with documents of different lengths. In the absence of bias scaling, the scale of the weights would get adjusted to work optimally for a typical document length. Documents longer than this would tend to saturate the units

| Input | Reconstruction |
|---|---|
| chocolate, cake | cake, chocolate, sweets, dessert, cupcake, food, sugar, cream, birthday |
| nyc | nyc, newyork, brooklyn, queens, gothamist, manhattan, subway, streetart |
| dog | dog, puppy, perro, dogs, pet, filmshots, tongue, pets, nose, animal |
| flower, high, 花 | flower, 花, high, japan, sakura, 日本, blossom, tokyo, lily, cherry |
| girl, rain, station, norway | norway, station, rain, girl, oslo, train, umbrella, wet, railway, weather |
| fun, life, children | children, fun, life, kids, child, playing, boys, kid, play, love |
| forest, blur | forest, blur, woods, motion, trees, movement, path, trail, green, focus |
| españa, agua, granada | españa, agua, spain, granada, water, andalucía, naturaleza, galicia, nieve |

Table 1: Some examples of one-step reconstruction from the Replicated Softmax Model.

and shorter than this would lead to vague activations of the hidden units. The conditional distributions are given by

$$p(h_j = 1|\mathbf{V}) \;=\; g\left(Ma_j + \sum_{k=1}^{K} \hat{v}_k W_{jk}\right), \tag{4}$$

$$p(v_{ik} = 1|\mathbf{h}) \;=\; \frac{\exp\left(b_k + \sum_{j=1}^{F} h_j W_{jk}\right)}{\sum_{q=1}^{K} \exp\left(b_q + \sum_{j=1}^{F} h_j W_{jq}\right)}. \tag{5}$$

A pleasing property of using softmax units is that the mathematics underlying the learning algorithm for binary RBMs remains the same. Given a collection of $N$ documents $\{\mathbf{V}_n\}_{n=1}^{N}$, the derivative of the log-likelihood with respect to parameters $W$ is

$$\frac{1}{N}\sum_{n=1}^{N} \frac{\partial \log P(\mathbf{V_n})}{\partial W_{jk}} \;=\; \mathbb{E}_{P_{\text{data}}}\left[\hat{v}_k h_j\right] - \mathbb{E}_{P_{\text{model}}}\left[\hat{v}_k h_j\right].$$

The Replicated Softmax model can also be interpreted as an RBM model that uses a single visible multinomial unit with support $\{1, ..., K\}$ which is sampled $M$ times (see Figure 2, right panel).

For all of the above models, exact maximum likelihood learning is intractable because exact computation of the expectation $\mathbb{E}_{P_{\text{model}}}[\cdot]$ takes time that is exponential in $\min\{D, F\}$, i.e the number of visible or hidden units. In practice, efficient learning is performed by following an approximation to the gradient of a different objective function, called the "Contrastive Divergence" (CD) (Hinton, 2002).

One way to illustrate the working of the model is to look at one-step reconstructions of some bags of words. Table 1 shows some examples. The words in the left column were given as input to the model. Then Equation 4 was used to compute a distribution over hidden units. Using these probabilities as states of the units, Equation 5 was used to obtain a distribution over words. The second column shows the words with the highest probability in that distribution. This model was trained using text from the MIR-Flickr data set which we use later in our experiments. We can see that the model has learned a basic understanding of which words are probable given the input words. For example, *chocolate, cake* generalizes to *sweets, desserts, food*, etc. The model often makes interesting inferences. For example, *girl, rain, station, norway* extends to *oslo, train, wet, umbrella, railway*, which are very plausible in that context. The model also captures regularities about language,

discovers synonyms across multiple languages and learns about geographical relationships. This shows that the hidden units can capture these regularities and represent them as binary features.

## 3. Deep Boltzmann Machines (DBMs)

A Deep Boltzmann Machine (Salakhutdinov and Hinton, 2009b) is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $\mathbf{v} \in \{0,1\}^D$, and a sequence of layers of hidden units $\mathbf{h}^{(1)} \in \{0,1\}^{F_1}$, $\mathbf{h}^{(2)} \in \{0,1\}^{F_2}$,..., $\mathbf{h}^{(L)} \in \{0,1\}^{F_L}$. There are connections only between hidden units in adjacent layers, as well as between visible and hidden units in the first hidden layer. Consider a DBM with three hidden layers[1] (i.e., $L = 3$). The energy of the joint configuration $\{\mathbf{v}, \mathbf{h}\}$ is defined as

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}; \theta) \ = \ & -\sum_{i=1}^{D}\sum_{j=1}^{F_1} W_{ij}^{(1)} v_i h_j^{(1)} - \sum_{j=1}^{F_1}\sum_{l=1}^{F_2} W_{jl}^{(2)} h_j^{(1)} h_l^{(2)} - \sum_{l=1}^{F_2}\sum_{p=1}^{F_3} W_{lp}^{(3)} h_l^{(2)} h_p^{(3)} \\ & - \sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{F_1} b_j^{(1)} h_j^{(1)} - \sum_{l=1}^{F_2} b_l^{(2)} h_l^{(2)} - \sum_{p=1}^{F_3} b_p^{(3)} h_p^{(3)}, \end{aligned}$$

where $\mathbf{h} = \{\mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}\}$ is the set of hidden units and $\theta = \{\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}, \mathbf{b}, \mathbf{b}^{(1)}, \mathbf{b}^{(2)}, \mathbf{b}^{(3)}\}$ is the set of model parameters, representing visible-to-hidden and hidden-to-hidden symmetric interaction terms, as well as bias terms. Biases are equivalent to weights on a connection to a unit whose state is fixed at 1. The probability that the model assigns to a visible vector $\mathbf{v}$ is given by the Boltzmann distribution

$$P(\mathbf{v}; \theta) = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left(-E(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}; \theta)\right).$$

Observe that setting both $\mathbf{W}^{(2)}$=0 and $\mathbf{W}^{(3)}$=0 recovers the simpler Restricted Boltzmann Machine (RBM) model. The derivative of the log-likelihood with respect to the model parameters takes the form

$$\frac{\partial \log P(\mathbf{v}; \theta)}{\partial \mathbf{W}^{(1)}} \ = \ \mathbb{E}_{P_{\text{data}}}[\mathbf{v}\mathbf{h}^{(1)\top}] - \mathbb{E}_{P_{\text{model}}}[\mathbf{v}\mathbf{h}^{(1)\top}]. \tag{6}$$

The derivatives with respect to parameters $\mathbf{W}^{(2)}$ and $\mathbf{W}^{(3)}$ take similar forms but instead involve the cross-products $\mathbf{h}^{(1)}\mathbf{h}^{(2)\top}$ and $\mathbf{h}^{(2)}\mathbf{h}^{(3)\top}$ respectively. Unlike RBMs, the conditional distribution over the states of the hidden variables conditioned on the data is no longer factorial. The exact computation of the data-dependent expectation takes time that is exponential in the number of hidden units, whereas the exact computation of the model's expectation takes time that is exponential in the number of hidden and visible units.

1. For clarity, we use three hidden layers. Extensions to models with more than three layers is straightforward.

Deep Boltzmann Machines (DBMs) are interesting for several reasons. Firstly, like Deep Belief Networks (Hinton et al., 2006), DBMs can discover several layers of increasingly complex representations of the input, use an efficient layer-by-layer pretraining procedure (Salakhutdinov and Hinton, 2009b), can be trained on unlabelled data and can be fine-tuned for a specific task using (possibly limited) labelled data. Secondly, unlike other models with deep architectures, the approximate inference procedure for DBMs incorporates a top-down feedback in addition to the usual bottom-up pass, allowing Deep Boltzmann Machines to better incorporate uncertainty about missing or noisy inputs. Thirdly, and perhaps most importantly, parameters of all layers can be optimized jointly by following the approximate gradient of a variational lower-bound on the likelihood objective. As we show in our experimental results, this greatly facilitates learning better generative models, particularly when modeling the multimodal data.

## 4. Multimodal Deep Boltzmann Machines

Let us first consider constructing a multimodal DBM using an image-text bi-modal DBM as our running example. Let $\mathbf{v}^m \in \mathbb{R}^D$ denote a real-valued image input and $\mathbf{v}^t \in \{1, ..., K\}$ denote an associated text input containing $M$ words, with $v^t_k$ denoting the count for the $k^{th}$ word.

We start by modeling each data modality using a separate two-layer DBM. (see Figure 3). Let $\mathbf{h}^{(1m)} \in \{0,1\}^{F^m_1}$ and $\mathbf{h}^{(2m)} \in \{0,1\}^{F^m_2}$ denote the two layers of hidden units. The probability that the image-specific two-layer DBM assigns to a visible vector $\mathbf{v}^m$ is given by

$$
\begin{aligned}
P(\mathbf{v}^m;\theta^m) &= \sum_{\mathbf{h}^{(1m)},\mathbf{h}^{(2m)}} P(\mathbf{v}^m,\mathbf{h}^{(2m)},\mathbf{h}^{(1m)};\theta^m) \\
&= \frac{1}{\mathcal{Z}(\theta^m)} \sum_{\mathbf{h}^{(1m)},\mathbf{h}^{(2m)}} \exp\Bigg( -\sum_i \frac{(v^m_i - b^m_i)^2}{2\sigma^2_i} + \sum_{ij} \frac{v^m_i}{\sigma_i} W^{(1m)}_{ij} h^{(1m)}_j + \\
&\qquad \sum_{jl} W^{(2m)}_{jl} h^{(1m)}_j h^{(2m)}_l + \sum_j b^{(1m)}_j h^{(1m)}_j + \sum_l b^{(2m)}_l h^{(2m)}_l \Bigg).
\end{aligned}
$$

Note that conditioned on the states of $\mathbf{h}^{(1m)}$, the image-specific DBM uses Gaussian distribution to model the distribution over real-valued image features. Similarly, text-specific DBM uses Replicated Softmax to model the distribution over word count vectors. The corresponding probability that the text-specific two-layer DBM assigns to $\mathbf{v}^t$ is given by

$$
\begin{aligned}
P(\mathbf{v}^t;\theta^t) &= \sum_{\mathbf{h}^{(1t)},\mathbf{h}^{(2t)}} P(\mathbf{v}^t,\mathbf{h}^{(2t)},\mathbf{h}^{(1t)};\theta^t) \\
&= \frac{1}{\mathcal{Z}_M(\theta^t)} \sum_{\mathbf{h}^{(1t)},\mathbf{h}^{(2t)}} \exp\Bigg( \sum_{jk} W^{(1t)}_{k,j} h^{(1t)}_j v^t_k + \sum_{jl} W^{(2t)}_{jl} h^{(1t)}_j h^{(2t)}_l + \\
&\qquad \sum_k b^t_k v^t_k + M \sum_j b^{(1t)}_j h^{(1t)}_j + \sum_l b^{(2t)}_l h^{(2t)}_l \Bigg),
\end{aligned}
$$

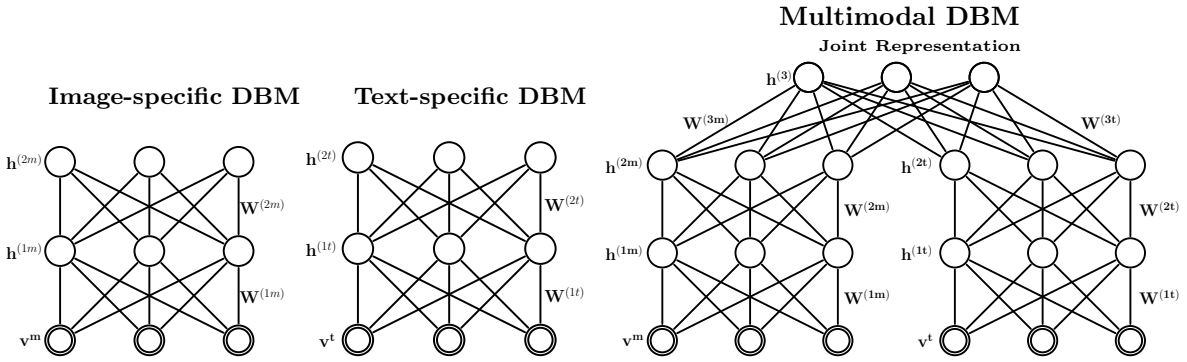



Figure 3: **Left:** Image-specific two-layer DBM that uses a Gaussian model to model the distribution over real-valued image features. **Middle:** Text-specific two-layer DBM that uses a Replicated Softmax model to model its distribution over the word count vectors. **Right:** A Multimodal DBM that models the joint distribution over image and text inputs. All layers but the first (bottom) layer use standard binary units.

where $\mathbf{h}^{(1t)} \in \{0,1\}^{F_1^t}$, $\mathbf{h}^{(2t)} \in \{0,1\}^{F_2^t}$ represent the two layers of hidden units.

To form a multimodal DBM, we combine the two models by adding an additional layer on top of them. The resulting graphical model is shown in Figure 3, right panel. The joint distribution over the multi-modal input, where $\mathbf{h} = \{\mathbf{h}^{(1m)}, \mathbf{h}^{(2m)}, \mathbf{h}^{(1t)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}\}$ denotes all hidden variables, can be written as

$$P(\mathbf{v}^m, \mathbf{v}^t; \theta) = \sum_{\mathbf{h}^{(2m)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}} P(\mathbf{h}^{(2m)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}) \Big( \sum_{\mathbf{h}^{(1m)}} P(\mathbf{v}_m, \mathbf{h}^{(1m)} | \mathbf{h}^{(2m)}) \Big) \Big( \sum_{\mathbf{h}^{(1t)}} P(\mathbf{v}^t, \mathbf{h}^{(1t)} | \mathbf{h}^{(2t)}) \Big)$$

$$= \frac{1}{\mathcal{Z}_M(\theta)} \sum_{\mathbf{h}} \exp \Bigg( \underbrace{\sum_{kj} W_{kj}^{(1t)} v_k^t h_j^{(1t)} + \sum_{jl} W_{jl}^{(2t)} h_j^{(1t)} h_l^{(2t)} + \sum_k b_k^t v_k^t + M \sum_j b_j^{(1t)} h_j^{(1t)} + \sum_l b_l^{(2t)} h_l^{(2t)}}_{\text{Replicated Softmax Text Pathway}}$$

$$\underbrace{- \sum_i \frac{(v_i^m - b_i^m)^2}{2\sigma_i^2} + \sum_{ij} \frac{v_i^m}{\sigma_i} W_{ij}^{(1m)} h_j^{(1m)} + \sum_{jl} W_{jl}^{(2m)} h_j^{(1m)} h_l^{(2m)} + \sum_j b_j^{(1m)} h_j^{(1m)} + \sum_l b_l^{(2m)} h_l^{(2m)}}_{\text{Gaussian Image Pathway}}$$

$$\underbrace{+ \sum_{lp} W^{(3t)} h_l^{(2t)} h_p^{(3)} + \sum_{lp} W^{(3m)} h_l^{(2m)} h_p^{(3)} + \sum_p b_p^{(3)} h_p^{(3)}}_{\text{Joint } 3^{rd} \text{ Layer}} \Bigg). \tag{7}$$

The normalizing constant depends on the number of words $M$ in the corresponding document, since the low-level part of the text pathway contains as many softmax units as there are words in the document. Similar to the Replicated Softmax model, the multimodal DBM can be viewed as a family of different-sized DBMs that are created for documents of different lengths that share parameters.

The conditional distributions over the visible and the five sets of hidden units are given by

$$
\begin{aligned}
p(h_j^{(1m)}=1|\mathbf{v}^m,\mathbf{h}^{(2m)}) &= g\left(\sum_{i=1}^{D} W_{ij}^{(1m)}\frac{v_i^m}{\sigma_i}+\sum_{l=1}^{F_2^m} W_{jl}^{(2m)}h_l^{(2m)}+b_j^{(1m)}\right), \qquad (8)\\
p(h_l^{(2m)}=1|\mathbf{h}^{(1m)},\mathbf{h}^{(3)}) &= g\left(\sum_{j=1}^{F_1^m} W_{jl}^{(2m)}h_j^{(1m)}+\sum_{p=1}^{F_3} W_{lp}^{(3m)}h_p^{(3)}+b_l^{(2m)}\right),\\
p(h_j^{(1t)}=1|\mathbf{v}^t,\mathbf{h}^{(2t)}) &= g\left(\sum_{k=1}^{K} W_{kj}^{(1t)}v_k^t+\sum_{l=1}^{F_2^t} W_{jl}^{(2t)}h_l^{(2t)}+Mb_j^{(1t)}\right),\\
p(h_l^{(2t)}=1|\mathbf{h}^{(1t)},\mathbf{h}^{(3)}) &= g\left(\sum_{j=1}^{F_1^t} W_{jl}^{(2t)}h_j^{(1t)}+\sum_{p=1}^{F_3} W_{lp}^{(3t)}h_p^{(3)}+b_l^{(2t)}\right),\\
p(h_p^{(3)}=1|\mathbf{h}^{(2)}) &= g\left(\sum_{l=1}^{F_2^m} W_{lp}^{(3m)}h_l^{(2m)}+\sum_{l=1}^{F_2^t} W_{lp}^{(3t)}h_l^{(2t)}+b_p^{(3)}\right),\\
p(v_{ik}^t=1|\mathbf{h}^{(1t)}) &= \frac{\exp\left(\sum_{j=1}^{F_1^t} h_j^{(1t)}W_{jk}^{(1t)}+b_k^t\right)}{\sum_{q=1}^{K}\exp\left(\sum_{j=1}^{F_1^t} h_j^{(1t)}W_{jq}^{(1t)}+b_k^t\right)},\\
v_i^m|\mathbf{h}^{(1m)} &\sim \mathcal{N}\left(\sigma_i\sum_{j=1}^{F_1^m} W_{ij}^{(1m)}h_j^{(1m)}+b_i^m,\sigma_i^2\right).
\end{aligned}
$$

Extending multimodal DBMs to other data modalities requires a simple modification of the first-layer modules. For example, consider modelling video-audio bi-modal data. Unlike image-text data, video-audio data can be represented as a sequence of real-valued vector pairs. Let $\mathbf{v}^m \in \mathbb{R}^D$ denote a real-valued input from the video stream (e.g., several consecutive image frames), and $\mathbf{v}^a \in \mathbb{R}^D$ denote an associated audio input (e.g., corresponding consecutive audio frames). We can easily construct the corresponding multimodal DBM with both pathways using Gaussian RBMs as the first layer. Compared to Equation 7, the the joint distribution over the multi-modal input variables, can be written as[2]

$$
\begin{aligned}
P(\mathbf{v}^m,\mathbf{v}^a;\theta) = \frac{1}{\mathcal{Z}(\theta)}\sum_{\mathbf{h}}\exp\Bigg(&\underbrace{-\sum_i\frac{(v_i^m)^2}{2\sigma_i^2}+\sum_{ij}\frac{v_i^m}{\sigma_i}W_{ij}^{(1m)}h_j^{(1m)}+\sum_{jl}W_{jl}^{(2m)}h_j^{(1m)}h_l^{(2m)}}_{\text{Gaussian Video Pathway}} \qquad (9)\\
&\underbrace{-\sum_i\frac{(v_i^a)^2}{2\sigma_i^2}+\sum_{ij}\frac{v_i^a}{\sigma_i}W_{ij}^{(1a)}h_j^{(1a)}+\sum_{jl}W_{jl}^{(2a)}h_j^{(1a)}h_l^{(2a)}}_{\text{Gaussian Audio Pathway}}+\underbrace{\sum_{lp}W^{(3t)}h_l^{(2t)}h_p^{(3)}+\sum_{lp}W^{(3m)}h_l^{(2m)}h_p^{(3)}}_{\text{Joint } 3^{rd} \text{ Layer}}\Bigg).
\end{aligned}
$$

2. We omit the bias terms for the hidden layers for clarity of presentation.

### 4.1 Approximate Inference and Learning

Exact maximum likelihood learning in this model is intractable, but efficient approximate learning of DBMs can be carried out by using a variational approach, where mean-field inference is used to estimate data-dependent expectations and an MCMC based stochastic approximation procedure is used to approximate the model's expected sufficient statistics.

#### 4.1.1 Estimating the Data-dependent Statistics

Consider any approximating distribution $Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu})$, parameterized by a vector of parameters $\boldsymbol{\mu}$, for the posterior $P(\mathbf{h}|\mathbf{v};\theta)$. Then the log-likelihood of the DBM model has the following variational lower bound,

$$\begin{aligned}\log P(\mathbf{v};\theta) &\geq \sum_{\mathbf{h}} Q(\mathbf{h}|\mathbf{v};\theta)\log P(\mathbf{v},\mathbf{h};\theta) + \mathcal{H}(Q) \\ &\geq \log P(\mathbf{v};\theta) - \mathrm{KL}(Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu})||P(\mathbf{h}|\mathbf{v};\theta)),\end{aligned} \tag{10}$$

where $\mathrm{KL}(Q||P)$ is the Kullback-–Leibler divergence between the two distributions, and $\mathcal{H}(\cdot)$ is the entropy functional. The bound becomes tight if and only if $Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) = P(\mathbf{h}|\mathbf{v};\theta)$.

We approximate the true posterior $P(\mathbf{h}|\mathbf{v};\theta)$, where $\mathbf{v} = \{\mathbf{v}^m, \mathbf{v}^t\}$, with a fully factorized approximating distribution over the five sets of hidden units $\{\mathbf{h}^{(1m)}, \mathbf{h}^{(2m)}, \mathbf{h}^{(1t)}, \mathbf{h}^{(2t)}, \mathbf{h}^{(3)}\}$:

$$Q(\mathbf{h}|\mathbf{v};\boldsymbol{\mu}) = \left(\prod_{j=1}^{F_1^m} q(h_j^{(1m)}|\mathbf{v}) \prod_{l=1}^{F_2^m} q(h_l^{(2m)}|\mathbf{v})\right)\left(\prod_{j=1}^{F_1^t} q(h_j^{(1t)}|\mathbf{v}) \prod_{l=1}^{F_2^t} q(h_l^{(2t)}|\mathbf{v})\right)\prod_{p=1}^{F_3} q(h_p^{(3)}|\mathbf{v}), \tag{11}$$

where $\boldsymbol{\mu} = \{\boldsymbol{\mu}^{(1m)}, \boldsymbol{\mu}^{(1t)}, \boldsymbol{\mu}^{(2m)}, \boldsymbol{\mu}^{(2t)}, \boldsymbol{\mu}^{(3)}\}$ are the mean-field parameters with $q(h_i^{(l)} = 1|\mathbf{v}) = \mu_i^{(l)}$ for $l = 1, 2, 3$.

For each training example, the variational bound of Equation 10 is maximized with respect to the variational parameters $\boldsymbol{\mu}$ for fixed parameters $\theta$. This results in the following mean-field fixed-point equations

$$\begin{aligned}
\mu_j^{(1m)} &\leftarrow g\Big(\sum_{i=1}^{D} W_{ij}^{(1m)}\frac{v_i^m}{\sigma_i} + \sum_{l=1}^{F_2^m} W_{jl}^{(2m)}\mu_l^{(2m)}\Big), \quad \mu_l^{(2m)} \leftarrow g\Big(\sum_{j=1}^{F_1^m} W_{jl}^{(2m)}\mu_j^{(1m)} + \sum_{k=1}^{F_3} W_{lk}^{(3m)}\mu_k^{(3)}\Big), \\
\mu_j^{(1t)} &\leftarrow g\Big(\sum_{k=1}^{K} W_{kj}^{(1t)}v_k^t + \sum_{l=1}^{F_2^t} W_{jl}^{(2t)}\mu_l^{(2t)}\Big), \quad \mu_l^{(2t)} \leftarrow g\Big(\sum_{j=1}^{F_1^t} W_{jl}^{(2t)}\mu_j^{(1t)} + \sum_{k=1}^{F_3} W_{lk}^{(3t)}\mu_k^{(3)}\Big), \\
\mu_p^{(3)} &\leftarrow g\Big(\sum_{l=1}^{F_2^m} W_{lp}^{(3m)}\mu_l^{(2m)} + \sum_{l=1}^{F_2^t} W_{lp}^{(3t)}\mu_l^{(2t)}\Big),
\end{aligned} \tag{12}$$

where $g(x) = 1/(1+\exp(-x))$ is the logistic function. To solve these fixed-point equations, we simply cycle through layers, updating the mean-field parameters within a single layer. The variational parameters $\boldsymbol{\mu}$ are then used to compute the data-dependent statistics in Equation 6. For example,

$$\mathbb{E}_{P_{\text{data}}}[\mathbf{v}^m \mathbf{h}^{(1m)\top}] = \frac{1}{N}\sum_{n=1}^{N} \mathbf{v}_n^m \boldsymbol{\mu}_n^{(1m)\top}$$
$$\mathbb{E}_{P_{\text{data}}}[\mathbf{h}^{(1m)} \mathbf{h}^{(2m)\top}] = \frac{1}{N}\sum_{n=1}^{N} \boldsymbol{\mu}_n^{(1m)} \boldsymbol{\mu}_n^{(2m)\top},$$

where the average on the RHS is over training cases. Note the close connection between the form of the mean-field fixed point updates and the form of the conditional distribution defined by Equation 8. In fact, implementing the mean-field updates requires no extra work beyond implementing the Gibbs sampler.

#### 4.1.2 Estimating the Data-independent Statistics

Given the variational parameters $\boldsymbol{\mu}$, the model parameters $\theta$ are then updated to maximize the variational bound using an MCMC-based stochastic approximation (Salakhutdinov and Hinton, 2009b; Tieleman, 2008; Younes, 1998). Remember that in out setting, we are learning a whole family of different-sized DBMs that depend on the number of words, or the number of replicated softmax variables (see Equation 7). Let us first assume that the text input only contains a set of $M$ words. Learning with stochastic approximation proceeds as follows. Let $\theta_t$ and $\mathbf{x}_t = \{\mathbf{v}_t^m, \mathbf{v}^t, \mathbf{h}_t^{(1m)}, \mathbf{h}_t^{(1t)}, \mathbf{h}_t^{(2m)}, \mathbf{h}_t^{(2t)}, \mathbf{h}_t^{(3)}\}$ be the current parameters and the state. Then $\mathbf{x}_t$ and $\theta_t$ are updated sequentially as follows:

- Given $\mathbf{x}_t$, sample a new state $\mathbf{x}_{t+1}$ from the transition operator $T_{\theta_t}(\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t)$ that leaves $P(\cdot; \theta_t)$ invariant. This can be accomplished by using Gibbs sampling (see Equation 8).
- A new parameter $\theta_{t+1}$ is then obtained by making a gradient step, where the intractable model's expectation $\mathbb{E}_{P_{\text{model}}}[\cdot]$ in the gradient is replaced by a point estimate at sample $\mathbf{x}_{t+1}$.

In practice, we typically maintain a set of $S$ "persistent" Markov chains $X_t = \{\mathbf{x}_{t,1}, ...., \mathbf{x}_{t,S}\}$, and use an average over those particles.

The overall learning procedure for DBMs is summarized in Algorithm 1. Extensions to the variable text input is trivial. For each $m = 1, ..., M_{max}$, where $M_{max}$ is the maximum number of words across all documents, we can create a corresponding multimodal DBM with $m$ replicated softmax variables and shared parameters. For each model $m$, we simply maintain a set of $S_m$ persistent Markov chains.[3] Learning then proceeds as discussed before.

Stochastic approximation provides asymptotic convergence guarantees and belongs to the general class of Robbins–Monro approximation algorithms (Robbins and Monro, 1951; Younes, 1998). Sufficient conditions that ensure almost sure convergence to an asymptotically stable point are given in Younes (1989, 1998); Yuille (2004). One necessary condition requires the learning rate to decrease with time, so that $\sum_{t=0}^{\infty} \alpha_t = \infty$ and $\sum_{t=0}^{\infty} \alpha_t^2 < \infty$.

3. Ideally, we would have each $S_m$ be as large as computationally feasible. However, given a fixed budget for the total number of chains, we could choose $S_m$ to be proportional to the number of documents containing $m$ words.

**Algorithm 1** Learning Procedure for a Multimodal Deep Boltzmann Machine.

Given: a training set of $N$ data vectors $\mathbf{v}_n = \{\mathbf{v}_n^m, \mathbf{v}_n^t\}$, $n = 1, ..., N$, and $S$, the number of persistent Markov chains. Let $\Lambda$ be a diagonal $D \times D$ matrix with $\Lambda_{ii} = 1/\sigma_i$.
Randomly initialize parameter vector $\theta_0$ and $S$ samples: $\{\tilde{\mathbf{v}}_{0,1}, \tilde{\mathbf{h}}_{0,1}\}, ..., \{\tilde{\mathbf{v}}_{0,S}, \tilde{\mathbf{h}}_{0,S}\}$, where we define $\tilde{\mathbf{h}} = \{\tilde{\mathbf{h}}^{(1m)}, \tilde{\mathbf{h}}^{(1t)}, \tilde{\mathbf{h}}^{(2m)}, \tilde{\mathbf{h}}^{(2t)}, \tilde{\mathbf{h}}^{(3)}\}$.
**for** $t = 0$ to $T$ (number of iterations) **do**

  // Variational Inference:
  **for** each training example $\mathbf{v}_n$, $n = 1$ to $N$ **do**
    Run the mean-field fixed-point updates until convergence using Equation 12.
    Set $\boldsymbol{\mu}_n = \boldsymbol{\mu}$.
  **end for**

  // Stochastic Approximation:
  **for** each sample $s = 1$ to $S$ (number of persistent Markov chains) **do**
    Sample $(\tilde{\mathbf{v}}_{t+1,s}, \tilde{\mathbf{h}}_{t+1,s})$ given $(\tilde{\mathbf{v}}_{t,s}, \tilde{\mathbf{h}}_{t,s})$ by running a Gibbs sampler for one step using Equation 8.
  **end for**

  // Parameter Update:

  // Image Pathway:

$$\mathbf{W}_{t+1}^{(1m)} = \mathbf{W}_t^{(1m)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \mathbf{v}_n^m \Lambda (\boldsymbol{\mu}_n^{(1m)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{v}}_{t+1,s}^m \Lambda (\tilde{\mathbf{h}}_{t+1,s}^{(1m)})^\top \right).$$

$$\mathbf{W}_{t+1}^{(2m)} = \mathbf{W}_t^{(2m)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\mu}_n^{(1m)} (\boldsymbol{\mu}_n^{(2m)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{h}}_{t+1,s}^{(1m)} (\tilde{\mathbf{h}}_{t+1,s}^{(2m)})^\top \right).$$

  // Text Pathway:

$$\mathbf{W}_{t+1}^{(1t)} = \mathbf{W}_t^{(1t)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \mathbf{v}_n^t (\boldsymbol{\mu}_n^{(1t)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{v}}_{t+1,s}^t (\tilde{\mathbf{h}}_{t+1,s}^{(1t)})^\top \right).$$

$$\mathbf{W}_{t+1}^{(2t)} = \mathbf{W}_t^{(2t)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\mu}_n^{(1t)} (\boldsymbol{\mu}_n^{(2t)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{h}}_{t+1,s}^{(1t)} (\tilde{\mathbf{h}}_{t+1,s}^{(2t)})^\top \right).$$

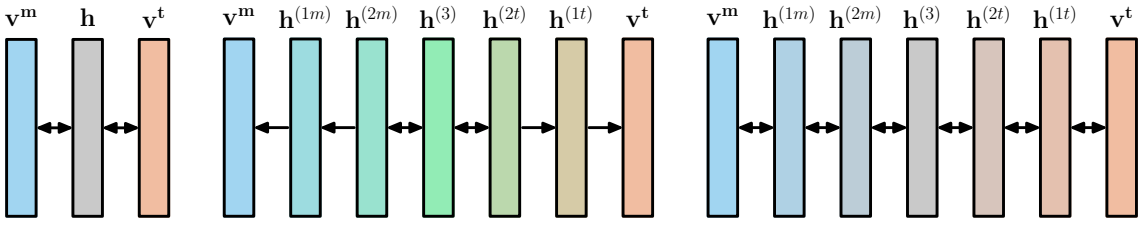
  // Joint Layer:

$$\mathbf{W}_{t+1}^{(3m)} = \mathbf{W}_t^{(3m)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\mu}_n^{(2m)} (\boldsymbol{\mu}_n^{(3)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{h}}_{t+1,s}^{(2m)} (\tilde{\mathbf{h}}_{t+1,s}^{(3)})^\top \right).$$

$$\mathbf{W}_{t+1}^{(3t)} = \mathbf{W}_t^{(3t)} + \alpha_t \left( \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{\mu}_n^{(2t)} (\boldsymbol{\mu}_n^{(3)})^\top - \frac{1}{S} \sum_{s=1}^{S} \tilde{\mathbf{h}}_{t+1,s}^{(2t)} (\tilde{\mathbf{h}}_{t+1,s}^{(3)})^\top \right).$$

  Decrease $\alpha_t$.
**end for**

This condition can, for example, be satisfied simply by setting $\alpha_t = a/(b + t)$, for positive constants $a > 0$, $b > 0$. Typically, in practice, the sequence $|\theta_t|$ is bounded, and the Markov chain, governed by the transition kernel $T_\theta$, is ergodic. Together with the condition on the learning rate, this ensures almost sure convergence of the stochastic approximation algorithm to an asymptotically stable point (Younes, 1998; Yuille, 2004).




(a) RBM (b) Multimodal DBN (c) Multimodal DBM

Figure 4: Different ways of modeling multimodal inputs.

#### 4.1.3 Greedy Layer-wise Pretraining

The learning procedure for Deep Boltzmann Machines described above can be used by initializing model parameters at random. However, the model performs much better if parameters are initialized sensibly. We therefore use a greedy layer-wise pretraining strategy by learning a stack of modified Restricted Boltzmann Machines (RBMs) (for details see Salakhutdinov and Hinton, 2009b). The pretraining procedure is quite similar to the pretraining procedure of Deep Belief Networks (Hinton et al., 2006), and it allows us to perform approximate inference by a single bottom-up pass. This fast approximate inference can also be used to initialize the mean-field, which then converges much faster than mean-field initialized at random.

## 5. Applying Multimodal DBMs to Different Tasks

There are several tasks that are of interest when working with multimodal data, such as generating missing modalities, inferring a joint representation or discriminative tasks that require classifying the multimodal input. In this section, we describe how DBMs can be used to solve these tasks. We also highlight the motivation behind the use of this approach.

### 5.1 Motivation

A Multimodal DBM can be viewed as a composition of unimodal undirected pathways. Each pathway can be pretrained separately in a completely unsupervised fashion, which allows us to leverage a large supply of unlabelled unimodal data. Any number of pathways each with any number of layers could potentially be used. The type of the lower-level RBMs in each pathway could be different, accounting for different input distributions. However, the hidden representations at the end of each pathway can be made to be of the same type (binary). Moreover, they can be encouraged to have nice statistical properties which we can control, such as having the same level of sparsity. These hidden representations are now much easier to combine than the low-level input representations.

The intuition behind our model is as follows. Each data modality has very different statistical properties which make it difficult for a single-layer model to directly find correlations in features across modalities. In our model, this difference is bridged by putting layers of hidden units between the modalities. The idea is illustrated in Figure 4c, which is just a different way of displaying Figure 3. Compared to the simple RBM (see Figure 4a), where the hidden layer $\mathbf{h}$ directly models the distribution over $\mathbf{v}^m$ and $\mathbf{v}^t$, the first layer of hidden

units $\mathbf{h}^{(1m)}$ in a DBM has an easier task to perform — that of modeling the distribution over $\mathbf{v}^m$ and $\mathbf{h}^{(2m)}$. Each layer of hidden units in the DBM makes a small contribution towards bridging $\mathbf{v}^m$ and $\mathbf{v}^t$. In the process, each layer learns successively higher-level representations and removes modality-specific correlations. Therefore, the middle layer in the network can be seen as a (relatively) "modality-free" representation of the input as opposed to the input layers which were "modality-full".

Another way of using a deep model to combine multimodal inputs is to use a Multimodal Deep Belief Network (DBN) (Figure 4b) which consists of an RBM at the center followed by directed belief networks leading out to the input layers. We emphasize that there is an important distinction between this model and the DBM model of Figure 4c. In a DBN model, and related autoencoder models, the responsibility for multimodal modeling falls entirely on the joint layer ($\mathbf{h}^{(2m)} \leftrightarrow \mathbf{h}^{(3)} \leftrightarrow \mathbf{h}^{(2t)}$). In the DBM, on the other hand, this responsibility is spread out over the entire network. From the generative perspective, states of low-level hidden units in one pathway can influence the states of hidden units in other pathways through the higher-level layers, which is not the case for DBNs.

### 5.2 Generating Missing Modalities

As argued in the introduction, many real-world applications will often have one or more modalities missing. The Multimodal DBM can be used to generate such missing data modalities by clamping the observed modalities at the inputs and sampling the hidden modalities by running the standard Gibbs sampler.

For example, consider generating text conditioned on a given image[4] $\mathbf{v}^m$. The observed modality $\mathbf{v}^m$ is clamped at the inputs and all hidden units are initialized randomly. Alternating Gibbs sampling is used to draw samples from $P(\mathbf{v}^t|\mathbf{v}^m)$ by updating each hidden layer given the states of the adjacent layers (see Equation 8). A sample drawn from this distribution describes a multinomial distribution over the text vocabulary. This distribution can then be used to sample words. This process is illustrated for a test image in Figure 5, showing the generated text after every 50 Gibbs steps. We see that not only does the sampler generate meaningful text, it shows evidence of jumping across different modes. For example, it generates *tropical*, *caribbean* and *resort* together, then moves on to *canada*, *bc*, *quebec lake*, *ice*, and then *italia*, *venizia* and *mare*. Each of these groups of words are plausible descriptions of the image. Moreover, each group is consistent within itself. This suggests that the model has been able to associate clusters of consistent descriptions with the same image. In other words, the model can capture multiple modes in the conditional distribution and access them by a Gibbs sampler.

The model can also be used to generate image features conditioned on text. Figure 6 shows examples of two such runs.

### 5.3 Inferring Joint Representations

The model can also be used to generate a joint representation of data by combining multiple data modalities. For inferring the joint representation, conditioned on the observed modalities, the observed modalities are clamped and Gibbs sampling is performed to sample from

4. Generating image features conditioned on text can be done in a similar way.

| Step 50 | Step 100 | Step 150 | Step 200 | Step 250 |
|---|---|---|---|---|
| travel | beach | sea | water | italy |
| trip | ocean | beach | canada | water |
| vacation | waves | island | bc | sea |
| africa | sea | vacation | britishcolumbia | boat |
| earthasia | sand | travel | reflection | italia |
| asia | nikon | ocean | alberta | mare |
| men | surf | caribbean | lake | venizia |
| 2007 | rocks | tropical | quebec | acqua |
| india | coast | resort | ontario | ocean |
| tourism | shore | trip | ice | venice |

Figure 5: Text generated by the DBM conditioned on an image by running a Gibbs sampler. Ten words with the highest probability are shown at the end of every 50 sampling steps.

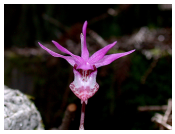
| Input tags | Step 50 | Step 100 | Step 150 | Step 200 | Step 250 |
|---|---|---|---|---|---|
| purple, flowers |  |  |  |  |  |
| car, automobile |  |  |  |  |  |

Figure 6: Images retrieved by running a Gibbs sampler conditioned on the input tags. The images shown are those which are closest to the sampled image features. Samples were taken after every 50 steps.

$P(\mathbf{h}^{(3)}|\mathbf{v}^m, \mathbf{v}^t)$ (if both modalities are present) or from $P(\mathbf{h}^{(3)}|\mathbf{v}^m)$ (if text is missing). A faster alternative, which we adopt in our experimental results, is to use variational inference to approximate posterior $Q(\mathbf{h}^{(3)}|\mathbf{v}^m, \mathbf{v}^t)$ or $Q(\mathbf{h}^{(3)}|\mathbf{v}^m)$ (see Section 4.1). The marginals of the approximate posterior over $\mathbf{h}^{(3)}$ (variational parameters $\boldsymbol{\mu}^{(3)}$) constitute the joint representation of the inputs.

This representation can then be used to do information retrieval for multimodal or unimodal queries. Each data point in the database (whether missing some modalities or not) can be mapped to this latent space. Queries can also be mapped to this space and an appropriate distance metric can be used to retrieve results that are close to the query.

### 5.4 Discriminative Tasks

After learning, the Multimodal Deep Boltzmann Machine can be used to initialize a multilayer neural network by partially unrolling the lower layers (Salakhutdinov and Hinton, 2009b). We can then use the standard backpropagation algorithm to discriminatively fine-tune the model. For each multimodal input vector $\mathbf{v} = \{\mathbf{v}^m, \mathbf{v}^t\}$, mean-field inference is

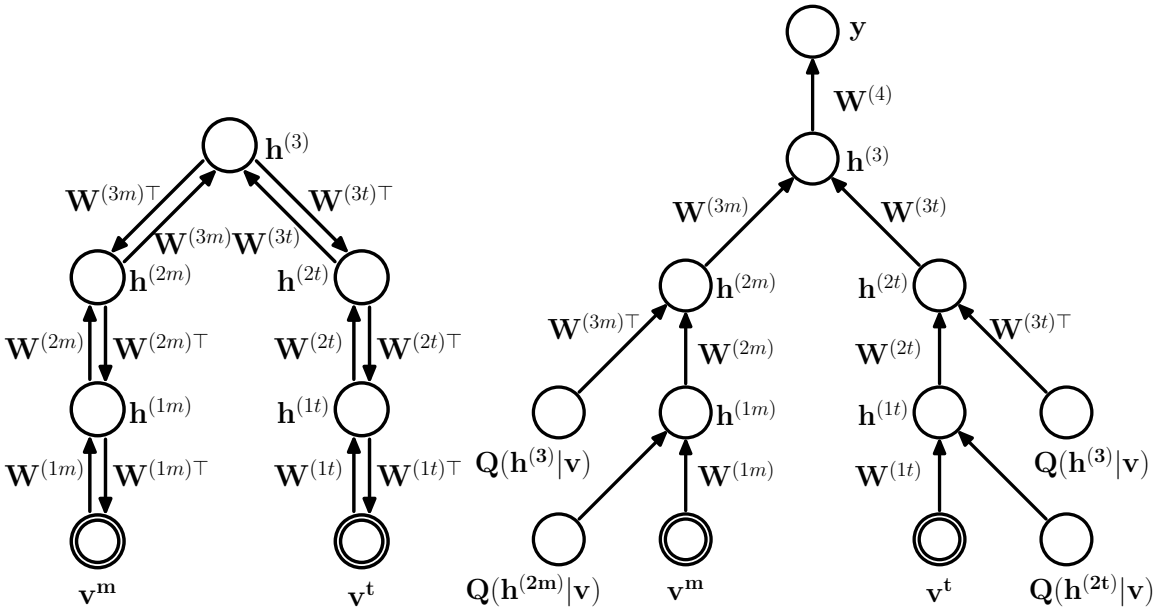



Figure 7: After learning, the DBM model as shown in (a) is used to initialize a multilayer neural network (b), where the marginals of approximate posterior $Q(h_i = 1|\mathbf{v})$ are used as additional inputs. The network is fine-tuned by backpropagation.

used to obtain an approximate posterior distribution $Q(\mathbf{h}|\mathbf{v})$. The marginals of this approximate posterior (variational parameters $\boldsymbol{\mu}$), together with the data, can be used to create an augmented input for this multimodal deep multilayer neural network, as shown in Figure 7. This augmented input is important because it helps maintain the scale of inputs that each hidden unit is expecting. For example, in Figure 7a, the conditional distribution over $\mathbf{h}^{(2m)}$, as defined by the DBM model (see Equation 8), takes the following form:

$$p(h_l^{(2m)} = 1|\mathbf{h}^{(1m)}, \mathbf{h}^{(3)}) \;=\; g\left(\sum_j W_{jl}^{(2m)} h_j^{(1m)} + \sum_p W_{lp}^{(3m)} h_p^{(3)} + b_l^{(2m)}\right).$$

Hence layer $\mathbf{h}^{(2m)}$ receives inputs from $\mathbf{h}^{(1m)}$ as well as $\mathbf{h}^{(3)}$. When this DBM is used to initialize a feed-forward network (Figure 7b), the augmented inputs $Q(\mathbf{h}^{(3)}|\mathbf{v})$ serve as a proxy for $\mathbf{h}^{(3)}$. This ensures that when the feed-forward network is fine-tuned, the hidden units in $\mathbf{h}^{(2m)}$ start off with receiving the same input as they would have received in a mean-field update during unsupervised pretraining. However, once the weights start changing during fine-tuning, the augmented inputs are no longer fixed points of the mean-field update equations and the model is free to use those inputs as it likes. The weights from $Q(\mathbf{h}^{(3)}|\mathbf{v})$ to $\mathbf{h}^{(2m)}$ are only initialized to $\mathbf{W}^{(3m)\top}$ and are not tied to the weights from $\mathbf{h}^{(2m)}$ to $\mathbf{h}^{(3)}$. This initialization scheme makes sure that the model starts fine-tuning from the same place where pretraining left off.

Note that the gradient-based fine-tuning may choose to ignore the marginals of the approximate posterior $Q(\mathbf{h}|\mathbf{v})$ by driving the corresponding weights to zero. This will result in a standard neural network, much like the neural network that is obtained from a Deep Belief Network or an autoencoder model. In practice, however, the network typically uses the entire augmented input for making predictions.

When using this model at test time, we first have to run the mean-field updates in the DBM to get the additional inputs and then use the fine-tuned feed-forward network to get the model's predictions. This creates an overhead in the running time. For all of the data sets in our experimental results, we typically used 5 mean-field updates, which was sufficient for the mean-field to settle down.

## 6. Experimental Results with Image-Text data

Our first data set consists of image-text pairs. Bi-modal data of this kind exemplifies a common real-world scenario where we have some image and a few words describing that image. There is a need to build representations that fuse this information into a homogeneous space, so that each data point can be represented as a single vector. This representation would be convenient for classification and retrieval problems.

### 6.1 Data Set and Feature Extraction

We used the MIR Flickr Data set (Huiskes and Lew, 2008) in our experiments. The data set consists of 1 million images retrieved from the social photography website Flickr along with their user assigned tags. The collection includes images released under the Creative Commons License. An example is shown in Figure 8. Among the 1 million images, 25,000 have been annotated using 24 labels including object categories such as, *bird*, *tree*, *people* and scene categories, such as *indoor*, *sky* and *night*. A stricter labeling was done for 14 of these classes where an image was annotated with a category only if that category was salient. This leads to a total of 38 classes where each image may belong to several classes. The data set also consists of an additional 975,000 unannotated images. From the 25,000 annotated images we use 10,000 images for training, 5,000 for validation and 10,000 for testing, following Huiskes et al. (2010). Mean Average Precision (MAP) is used as the performance metric. Results are averaged over 5 random splits of the 25,000 examples into train, validation and test sets.

There are more than 800,000 distinct tags in the data set. In order to keep the text representation manageable, each text input was represented using a vocabulary of the 2000 most frequent tags in the 1 million collection. After restricting to this vocabulary, the average number of tags associated with an image is 5.15 with a standard deviation of 5.13. There are 128,501 images which do not have any tags, out of which 4,551 are in the labelled 25K subset. Hence about 18% of the labelled data has images but is missing text.

Images were represented by 3857-dimensional features, that were extracted by concatenating Pyramid Histogram of Words (PHOW) features (Bosch et al., 2007), Gist (Oliva and Torralba, 2001) and MPEG-7 descriptors (EHD, HTD, CSD, CLD, SCD) (Manjunath et al., 2001). Each dimension was mean-centered and normalized to unit variance. PHOW features are bags of image words obtained by extracting dense SIFT features over multiple

| | | | | | |
|---|---|---|---|---|---|
| **Classes** | baby, female, people, portrait | plant life, river, water | clouds, sea, sky, transport, water | animals, dog, food | clouds, sky, structures |
| **Images** |  |  |  |  |  |
| **Tags** | claudia | ⟨ no text ⟩ | barco, pesca, boattosail, navegação | watermelon, hilarious, chihuahua, dog | colors, cores, centro, comercial, building |

Figure 8: Some examples from the MIR-Flickr data set. Each instance in the data set is an image along with textual tags. Each image has multiple classes.

scales and clustering them. We used publicly available code (Vedaldi and Fulkerson, 2008; Bastan et al., 2010) for extracting these features.[5]

### 6.2 Model Architecture and Implementation Details

The image pathway consists of a Gaussian RBM with 3857 linear visible units and 1024 hidden units. This is followed by a layer of 1024 binary hidden units. The text pathway consists of a Replicated Softmax Model with 2000 visible units and 1024 hidden units, followed by another layer of 1024 hidden units. The joint layer contains 2048 hidden units. All hidden units are binary. Each Gaussian visible unit was set to have unit variance ($\sigma_i = 1$) which was kept fixed and not learned.[6] Each layer of weights was pretrained using CD-$n$ where $n$ was gradually increased from 1 to 20. All word count vectors were normalized so that they sum to one. This way we avoid running separate Markov chains for each document length to get the model distribution's sufficient statistics, which makes it possible to have a fast GPU implementation.

We also experimented with training a proper generative model, that is, without normalizing the data. Remember, the image-text bimodal DBM can be viewed as a family of different-sized DBMs that are created for documents of different lengths that share parameters. In this setting, we used separate MCMC chains for different sized documents. However, the results were statistically indistinct from the case when we made the simplifying assumptions. This is probably because this data set does not have a huge variance in the number of words per image (5-15 tags per image).

After training the DBM model generatively, we applied it for classification and retrieval tasks. We compared different ways of using the model for classification. The simplest method is to extract the representation at the joint hidden layer and perform 1-vs-all classification using logistic regression. We compare this to fine-tuning the model discriminatively as described in Section 5.4. We also used dropout (Hinton et al., 2012; Srivastava et al., 2014) during fine-tuning to further improve the classification performance. For dropout, we retained each unit with probability $p = 0.8$.

5. The extracted features are publicly available at `http://www.cs.toronto.edu/~nitish/multimodal`.
6. We found that learning the variance made the training unstable.

| Model | MAP | Prec@50 |
|---|---|---|
| Random | 0.124 | 0.124 |
| SVM (Huiskes et al., 2010) | 0.475 | 0.758 |
| LDA (Huiskes et al., 2010) | 0.492 | 0.754 |
| DBM | 0.526 ± 0.007 | 0.791 ± 0.008 |
| DBM (using unlabelled data) | **0.585** ± 0.004 | **0.836** ± 0.004 |

Table 2: Multimodal Classification Results. Mean Average Precision (MAP) and precision@50 obtained by different models. A similar set of input features is used across all models.

### 6.3 Classification Tasks

We run two classification experiments to highlight two distinct capabilities of the proposed DBM model. In the first experiment, we train and test the model on multimodal inputs. This experiment is designed to evaluate the DBM's ability to *represent* multimodal data in a way that is useful for classification. In the second experiment, we train on multimodal inputs, but at test time we are only given images. This experiment is designed to evaluate the DBM's ability to *generate* useful text and use it as a substitute for real data.

Since examples in the data set may have multiple labels, classification accuracy is not very meaningful. Instead, we evaluate our models using Mean Average Precision (MAP) and precision at top-50 predictions (Prec@50). These are standard metrics used for multi-label classification and have been previously used to report results on this data set.

#### 6.3.1 Multimodal Inputs

In out first experiment, the task is to assign labels to image-text pairs. Huiskes et al. (2010) provided baselines for this data set with Linear Discriminant Analysis (LDA) and RBF-kernel Support Vector Machines (SVMs) using the labelled 25K subset of the data. They represent the multimodal input as a concatenation of image features and word counts. Table 2 shows the performance of these models. The image features did not include SIFT-based features. Therefore, to make a fair comparison, our model was first trained using the same amount of data and a similar set of features (i.e., excluding our SIFT-based features). Table 2 shows that the DBM model outperforms its competitor SVM and LDA models in terms of MAP and Prec@50. The DBM achieves a MAP of 0.526, compared to 0.475 and 0.492, achieved by SVM and LDA models.

Next, we tried to see how much gain can be obtained by using the 975,000 unlabelled examples. We trained a DBM using these examples and, not surprisingly, this improved the DBM's MAP to 0.585.

Having established that DBMs outperform simple linear models, we now compare DBMs to two other deep models—Deep Belief Nets (DBNs) (Hinton et al., 2006) and Denoising Autoencoders (DAEs) (Vincent et al., 2008). We found that further improvements can be obtained by using more image features. We added PHOW features, which use dense SIFT descriptors, to learn a feature dictionary. Table 3 shows results using this extended feature set. We use unlabelled data to pretrain these models. We also closely explore the benefits of full discriminative fine-tuning, regularizers that encourage sparse activations and dropout.

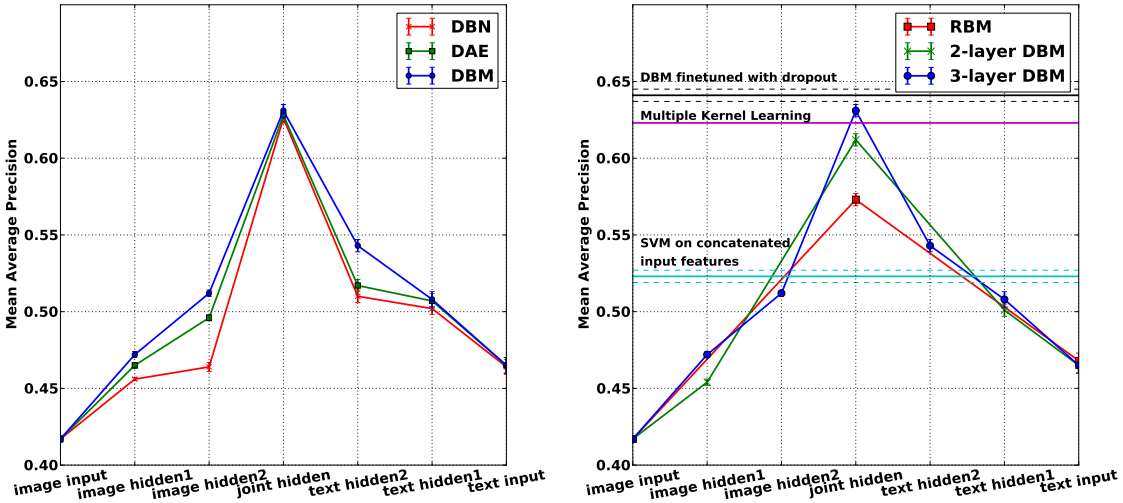
| Model | No Pretraining | DBN | DAE | DBM |
|---|---|---|---|---|
| Logistic regression on joint layer features | - | 0.599 ± 0.004 | 0.600 ± 0.004 | 0.609 ± 0.004 |
| Sparsity + Logistic regression on joint layer features | - | 0.626 ± 0.003 | 0.628 ± 0.004 | 0.631 ± 0.004 |
| Sparsity + discriminative fine-tuning | 0.482 ± 0.003 | 0.630 ± 0.004 | 0.630 ± 0.003 | 0.634 ± 0.004 |
| Sparsity + discriminative fine-tuning + dropout | 0.575 ± 0.004 | 0.638 ± 0.004 | 0.638 ± 0.004 | **0.641 ± 0.004** |

Table 3: Comparison of MAP across different deep models. Sparsity, full discriminative fine-tuning and dropout lead to improvements across all models. More input features were used compared to Table 2.

First, we apply simple logistic regression on the high-level joint representation learned by each of the three models. As shown in Table 3, the DBN and DAE obtain a MAP of 0.599 and 0.600 respectively, whereas the DBM gets 0.609. The error bars indicate that this improvement is statistically significant. The DBNs and DAEs give very similar results. Next we added a KL-sparsity regularizer (Olshausen and Field, 1996) during unsupervised pretraining of all the three models. This improved the performance across all models. In particular, the DBM achieved a MAP of 0.631. Full discriminative training further improved the DBM's MAP to 0.634. Next, we fine-tuned the model using the dropout technique proposed by Hinton et al. (2012). Using this we achieved the a MAP of **0.641**. The DBN and DAE also produce very close results. The Multiple Kernel Learning approach proposed by Guillaumin et al. (2010) obtained a MAP of 0.623 where they used a much larger set of image features (37,152 dimensions). TagProp (Verbeek et al., 2010) obtained a MAP of 0.640 which is comparable to DBMs again using a much larger set of features.

In terms of Prec@50, the DBM achieves a score of $0.888 \pm 0.004$. The DBN and DAE score $0.887 \pm 0.003$ and $0.888 \pm 0.004$ respectively. Therefore, all the deep models do about the same in terms of this metric.

Learning a deep hierarchy of features is widely believed to be the reason why deep models have been successful in a number of machine learning tasks. In order to better understand the properties of different layers in the network, we evaluate the quality of representation at each layer of the network. We do this by measuring MAP obtained by logistic regression classifiers on the representation at different layers of the network. We choose a simple classifier so that the MAP results represent a good measure of the representation's discriminative ability. Figure 9a compares different deep models. In all the models, MAP increases as we go from the input layer towards the joint hidden layer from either side. This shows that higher level representations become increasingly good at discovering useful features. It is interesting to note that the performance of the DBM's hidden layers increases rapidly with depth whereas that for the DBN seems to stagnate at the second layer. At the middle (joint) layer, the performance of both models increases tremendously. This behavior points to an important property of DBMs. Intuitively, the joint generative training of all the layers allows information to flow more readily between




(a) (b)

Figure 9: Mean Average Precision (MAP) obtained by applying logistic regression to representations learned at different layers. **Left :** Comparison of different deep models - Deep Belief Nets, Denoising Autoencoders and Deep Boltzmann Machines. All model have the same architecture and same number of parameters. **Right:** Comparison of DBMs of different depths with SVMs and MKL models. Observe that adding depth improves performance.

the image and text pathways. This comparison empirically verifies the intuition behind having undirected connections throughout the model as mentioned in Section 5.1.

Next, we investigate the effect of depth more closely on DBMs. The question that we try to answer here is how many intervening layers of hidden units should we put between the image and text modalities. It is useful to think of intervening layers as shown in Figure 4c. We could just have one intervening layer, creating an RBM (image input—joint hidden layer—text input). A two-layered DBM would have 3 intervening layers (image input—image hidden 1—joint hidden—text hidden 1—text input), and so on. Figure 9b compares the layer-wise performance of these models (RBM, 2-layer DBM and 3-layer DBM). The performance of other models is also indicated with horizontal lines. Comparing the performance of the joint hidden layer across the three models, we can see that having more intervening layers leads to better performance. The incremental utility of adding more layers seems to decrease.

### 6.3.2 Unimodal Inputs

In a multimodal data setting, it is very common for some data points to be missing some data modalities. For example, there may be images which do not have captions or tags. This raises interesting questions—Can we use a model that was trained on images and text, when we only have images at test time? Can this model do better than one that was trained on images alone? For multimodal DBMs the answer is affirmative. In this section, we describe an experiment to demonstrate this.

The task is the same as in the previous experiment. We trained a DBM using the unlabelled data and fine-tuned it for discrimination as before. The only difference is that

| Model | MAP | Prec@50 |
|---|---|---|
| Image LDA (Huiskes et al., 2010) | 0.315 | - |
| Image SVM (Huiskes et al., 2010) | 0.375 | - |
| Image DBN | 0.463 ± 0.004 | 0.801 ± 0.005 |
| Image DBM | 0.469 ± 0.005 | 0.803 ± 0.005 |
| Multimodal DBM (generated text) | **0.531 ± 0.005** | **0.832 ± 0.004** |

Table 4: Unimodal Classification Results. Mean Average Precision (MAP) and precision@50 obtained by different models. A similar set of input features is used across all models.

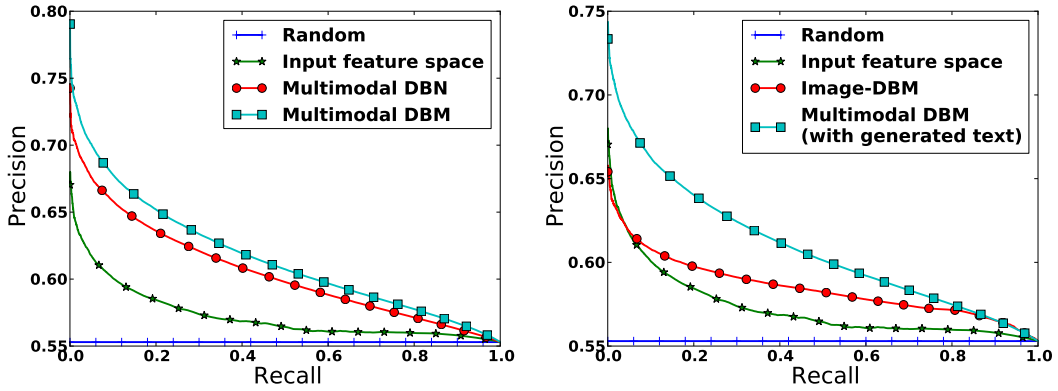
at test time, the model was given only image inputs and used the DBM to generate and fill in missing data. This was done by mean-field updates. We also tried Gibbs sampling and found that it work just as well but with more variance.

We compare the Multimodal DBM with models that were trained using only images. We compare with baseline (RBF-kernel) SVM and LDA results, using a restricted feature set which is similar to that used in Huiskes et al. (2010). Table 4 shows that the LDA and SVM models achieve a MAP of 0.315 and 0.375, respectively. A DBN trained on the similar image features improves this to 0.463. A DBM further improves this to 0.469. In both these cases, pretraining was done using images from the unlabelled set. Both models had the same number of layers and same number of hidden units in each layer. Next, we used a Multimodal DBM to infer the text input and hidden representations at each layer (using mean-field updates). At test time, these representations along with the image features were given as input to the discriminatively fine-tuned DBM. This achieved a significantly higher MAP of 0.531.

This result shows that the DBM can generate meaningful text that serves as a plausible proxy for missing data. This further suggests that *learning multimodal features helps even when some modalities are absent at test time*. The model learns much better features when it has access to multiple modalities because it is being asked to discover features that explain both modalities simultaneously. This can be interpreted as a regularization effect, where instead of the asking the model to be simple or sparse, we ask it to explain an alternative "view" of the data which lies on a very different manifold but shares essential discriminative characteristics with the original view.

### 6.4 Retrieval Tasks

The next set of experiments was designed to evaluate the quality of the learned joint representation for retrieval purposes. A database of images was created by randomly selecting 5000 image-text pairs from the test set. We also randomly selected a disjoint set of 1000 images to be used as queries. Each query contained both image and text modalities. Each data point has 38 labels. Using these, binary relevance labels were created by assuming that if any of the 38 labels overlapped between a query and a data point, then that data point is relevant to the query.




(a) Multimodal Queries (b) Unimodal Queries

Figure 10: Precision-Recall curves for Retrieval Tasks.

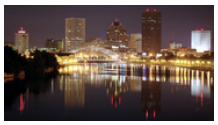
| **Multimodal Query** | **Top 4 retrieved results** | | | |
|---|---|---|---|---|
|  hongkong, causewaybay, shoppingcentre, building, mall |  howell, bridge, genesee, river, rochester, downtown, building |  london, uk, night, skyline, river, thames, lights, bridge |  edinburgh, scotland, dusk, bank |  arcoiris, fincadehierro, lluvia, sannicolas, valencia |
|  me, myself, eyes, blue, hair |  urban, me, abigfave, fiveflickrfavs, |  trisha, mynewcamera, lake, field, girl |  me, ofme, self, selfportrait |  pink, prettyinpink, explored |

Figure 11: Retrieval Results for Multimodal Queries from the DBM model.

### 6.4.1 Multimodal Queries

Figure 10a shows the precision-recall curves for the DBM, DBN, and DAE models (averaged over all queries). For each model, all queries and all points in the database were mapped to the joint hidden representation under that model. Cosine similarity function was used to match queries to data points.

The DBM model performs the best among the compared models achieving a MAP of 0.622. This is slightly better than the performance of the autoencoder and DBN models which achieve a MAP of 0.612 and 0.609 respectively. Figure 11 shows some examples of multimodal queries and the top 4 retrieved results. Note that even though there is little overlap in terms of text, the model is able to perform well.

| | | | | |
|---|---|---|---|---|
| **Image** |   |  |  |  |
| **Generated Tags** | water, glass, beer, bottle, drink, wine, bubbles, splash, drops, drop | portrait, women, army, soldier, mother, postcard, soldiers | nikon, d200, nikkor, d50, tamron, d300, d90, f28, sb600, d60 | obama, barackobama, election, politics, president, hope, change |

Figure 12: Examples where the DBM does not work well.

#### 6.4.2 Unimodal Queries

The DBM model can also be used to query for images alone. Figure 10b shows the precision-recall curves for the DBM model along with other unimodal models. Each model received the same set of test image queries as input. The joint hidden representation was inferred keeping the text input layer unclamped. Using this representation, the DBM model was able to achieve far better results than any unimodal method (MAP of 0.614 as compared to 0.587 for an Image-DBM and 0.578 for an Image-DBN).

### 6.5 When Does the Model Not Work?

In this section, we analyze the DBM model to understand when it fails to work and what exactly goes wrong. Figure 12 shows some examples where the model fails to generate meaningful text. To diagnose the problem, we looked at the Markov chains that lead to these results. By visual observation, it was clear that some of these chains got stuck in a region of space and never came out. This happened often when the text sampler reached the space of frequently occurring tags, such as those which refer to camera brands or lens specifications. These tags occur across all kinds of images and seem to take up a huge probability mass under the model independent of the image. There could be other more subtle causes of failure but they were hard to diagnose by visual inspection.

## 7. Experimental Results with Video-Audio Data

We next demonstrate our approach on video-audio bimodal data. We use data sets that consist of videos of lip movements along with the sound recordings of the words being spoken. This setting has been previously explored in the context of deep multimodal learning by Ngiam et al. (2011) using sparse denoising autoencoders.

### 7.1 Preprocessing and Data Sets

We represent the auditory information using 40 dimensional log-filter banks along with temporal derivatives to create a 120-d frame for 20 ms speech windows with a stride of 10 ms. Similar to Ngiam et al. (2011) we extract $60 \times 80$ mouth regions from the video using a simple object detector (Dalal and Triggs, 2005). The detections were cleaned by median filtering. The extracted mouth regions were compressed to 32 dimensions with

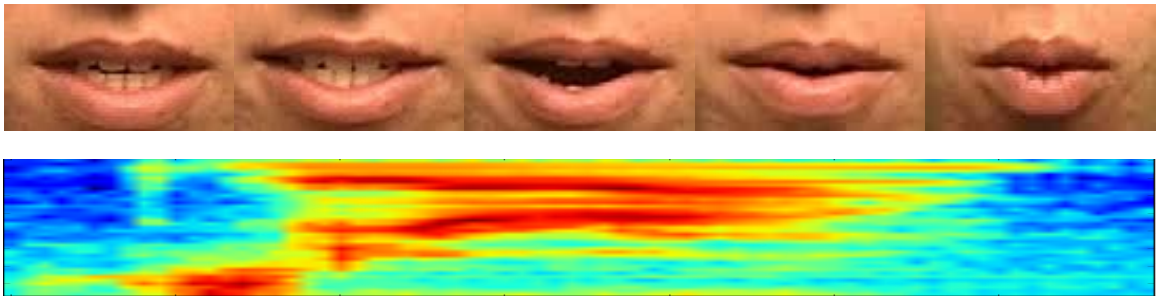


Figure 13: An example of audio-video data extracted from the CUAVE data set.

PCA. Temporal derivatives were then added to create a 96 dimensional representation for each frame. We combined several data sets in this experiment.

**CUAVE** (Patterson et al., 2002): This data set consists of 36 speakers speaking the digits 0 to 9. The data set has each speaker speaking with different facial orientations (front and sideways) and speaking speeds. We exclude the sideways oriented portions of the data set for simplicity. We use half the speakers for testing and the other half for training.

**AVLetters** (Matthews et al., 2002): This data set consists of 10 speakers speaking letters A-Z three times each. This data set does not come with raw audio and was used for unsupervised pretraining of the video pathway. We treat this data set as if it were missing audio and evaluate the DBM's ability of fill in the missing data and use it for classification.

**AVLetters 2** (Cox et al., 2008): This data set consists of high resolution recordings from 5 speakers speaking letters A-Z. The videos were down-sampled. This data set was used for unsupervised training of the entire model.

**TIMIT** (Fisher et al., 1986): This data set consists of recordings from 680 speakers covering 8 major dialects of American English reading ten phonetically-rich sentences in a controlled environment. We used this for the unsupervised pretraining of the auditory pathway.

In addition to these, Ngiam et al. (2011) use the Stanford Data Set which consists of 23 speakers speaking the letters A-Z and digits 0-9. However, this data set is not publicly available yet and we were unable to use it. Since all of the above data sets differ in terms of video recording environments, we use PCA in the hope to ameliorate some of these difference. In all the experiments, all available data was used for unsupervised pretraining.

### 7.2 Model Description

A Multimodal DBM was trained with 4 consecutive image frames and 10 consecutive audio frames since they roughly correspond to same amount of time. Both pathways used Gaussian RBMs as the first layer, as defined in Equation 9. The auditory pathway consisted of 1200 input units followed by 2 layers of 1024 hidden units. The visual pathway had 384 input units followed by 2 layers of 1024 hidden units. The joint layer had 2048 hidden units.

The task was to label each utterance with the digit or letter that was being uttered. Different utterances had different lengths. We obtained a fixed length representation by applying average pooling on the features obtained from the joint layer. In addition, we divided each utterance into 3 equal splits and average pooled the features over those separately.

| Method | Classification accuracy % |
|---|---|
| Concatenated video and audio features | 63.5 |
| Video RBM (Ngiam et al., 2011) | 65.4 ± 0.6 |
| Multimodal DAE (Ngiam et al., 2011) | 66.7 |
| Multimodal DBN | 67.2 ± 0.9 |
| Video DBM | 67.8 ± 1.1 |
| Video DAE (Ngiam et al., 2011) | 68.7 ± 1.8 |
| Multimodal DBM | **69.0 ± 1.5** |
| Discrete Cosine Transform (Gurban and Thiran, 2009) | 64 |
| Active Appearance Model (AAM) (Papandreou et al., 2007) | 75.7 |
| Fused Holistic + Patch (Lucey and Sridharan, 2006) | 77.08 |
| Visemic AAM (Papandreou et al., 2009) | 83 |

Table 5: Classification results on the CUAVE data set.

These 4 sets of pooled features were concatenated to form the multimodal representation of the input. We then used a linear SVM to classify based on these representations. This is the same as the method used in Ngiam et al. (2011). No discriminative fine-tuning of the DBM was performed.

### 7.3 Classification Results

We report the results of two classification experiments. In the first experiment, we classify utterances from the CUAVE data set into 10 digit classes. We use the DBM to extract features using both video and audio inputs. We compare this to a DBN, DAE (Ngiam et al., 2011) and various other methods. The results are shown in Table 5. Linear SVM on concatenated video and audio features serves as a baseline, which achieves 63.5% accuracy. A video-only RBM achieves 65.4%, which can be improved to 67.2% with a 3-layer DBN and to 67.8% with a 3-layer DBM. The denoising autoencoder achieves an even better performance of 68.7%. Ngiam et al. (2011) showed that adding audio features seems to hurt the performance of DAEs, reducing it down to 66.7%. The Multimodal DBM does not suffer from adding audio features and improves the performance slightly to 69.0%. However, this is not a significant improvement over the Video DAE. Note that the DBM was trained on less data compared to the Video DAE of Ngiam et al. (2011). The Multimodal DBM does improve significantly on the performance of the Video DBM trained on the same amount of data.

The performance of the deep models is much worse than that of Active Appearance Models (Papandreou et al., 2007, 2009) and Patch-based methods (Lucey and Sridharan, 2006). However, these models use a different train-test split and specialized image preprocessing techniques that are specifically designed for visual speech recognition tasks.

In our second experiment, we try to classify utterances from the AVLetters data set into 26 letter classes. In this case the audio input is considered missing and we use the DBM to infer the joint hidden representation keeping the audio input unclamped. We do the same for a DBN as well as compare to DAEs and other methods. Table 6 shows the results.

The baseline model which uses the preprocessed video features achieves 46.2% accuracy. An RBM on the same features achieves 54.2%, whereas a 3-layer DBM gets 61.8% and a DAE gets 64.4%. However, the Multimodal DAE again suffers from adding audio features at

| Method | Classification accuracy |
|---|---|
| Video features (Ngiam et al., 2011) | 46.2 |
| Video RBM (Ngiam et al., 2011) | 54.2 ± 3.3 |
| Multimodal DAE (Ngiam et al., 2011) | 59.2 |
| Video DBM | 61.8 ± 2.5 |
| Multimodal DBN | 63.2 ± 2.1 |
| Video DAE (Ngiam et al., 2011) | 64.4 ± 2.4 |
| Multimodal DBM | **64.7 ± 2.5** |
| Multiscale Spatial Analysis (Matthews et al., 2002) | 44.6 |
| Local Binary Pattern (Zhao et al., 2009) | 58.85 |

Table 6: Classification results on the AVLetters data set.

test time compared to a Video DAE, getting an accuracy of 59.2%. The Multimodal DBM, on the other hand, improves over the Video DBM and gets 64.7%, essentially matching the performance of the Video DAE (even though it used less data).

These experiments show that the Multimodal DBM model can effectively combine features across modalities. It consistently shows improvements over training on unimodal data, even when only unimodal inputs are given at test time.

## 8. Conclusion

We proposed a Deep Boltzmann Machine model for learning multimodal data representations. Large amounts of unlabelled data can be effectively utilized by the model. Pathways for each modality can be pretrained independently and "plugged in" together for performing joint learning. The model fuses multiple data modalities into a unified representation, which captures features that are useful for classification and retrieval. It also performs well when some modalities are absent and improves upon models trained on only the observed modalities. Our model performs well in terms of classification results on the bi-modal MIR-Flickr data set as well as on the CUAVE and AVLetters video-audio data sets, demonstrating the usefulness of this approach.

## Acknowledgments

This research was supported by Google, Samsung, and ONR Grant N00014-14-1-0232.

## References

M. Bastan, H. Cam, U. Gudukbay, and O. Ulusoy. Bilvideo-7: An MPEG-7- compatible video indexing and retrieval system. *IEEE Multimedia*, 17:62–73, 2010. ISSN 1070-986X.

A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. *IEEE 11th International Conference on Computer Vision (2007)*, 23:1–8, 2007.

S. Cox, R. Harvey, Y. Lan, J. Newman, and B. Theobald. The challenge of multispeaker lip-reading. In *International Conference on Auditory-Visual Speech Processing*, pages

179–184, September 2008.

N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 886–893, 2005.

W. M. Fisher, G. R. Doddington, and K. M. Goudie-Marshall. The DARPA speech recognition research database: Specifications and status. In *Proceedings of DARPA Workshop on Speech Recognition*, pages 93–99, 1986.

Y. Freund and D. Haussler. Unsupervised learning of distributions on binary vectors using two layer networks. Technical report, University of California at Santa Cruz, Santa Cruz, CA, USA, 1994.

M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 902 –909, 2010.

M. Gurban and J.-P. Thiran. Information theoretic feature extraction for audio-visual speech recognition. *IEEE Transactions on Signal Processing*, 57(12):4765–4776, 2009.

G. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.

G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1711–1800, 2002.

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

G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012.

M. J. Huiskes and M. S. Lew. The MIR Flickr retrieval evaluation. In *ACM International Conference on Multimedia Information Retrieval*, 2008.

M. J. Huiskes, B. Thomee, and M. S. Lew. New trends and ideas in visual concept detection: the MIR Flickr retrieval evaluation initiative. In *11th ACM International Conference on Multimedia Information Retrieval*, pages 527–536, 2010.

H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the 26th International Conference on Machine Learning*, pages 609–616, 2009.

P. Lucey and S. Sridharan. Patch-based representation of visual speech. In *Proceedings of the HCSNet workshop on Use of vision in human-computer interaction - Volume 56*, VisHCI '06, pages 79–85. Australian Computer Society, Inc., 2006.

B. Manjunath, J.-R. Ohm, V. Vasudevan, and A. Yamada. Color and texture descriptors. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):703 –715, 2001.

I. Matthews, T. F. Cootes, J. A. Bangham, S. Cox, and R. Harvey. Extraction of visual features for lipreading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):198–213, Feb. 2002.

A. Mohamed, G. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.

J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42:145–175, 2001.

B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.

G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Multimodal fusion and learning with uncertain features applied to audiovisual speech recognition. In *IEEE 9th Workshop on Multimedia Signal Processing*, pages 264–267, 2007.

G. Papandreou, A. Katsamanis, V. Pitsikalis, and P. Maragos. Adaptive multimodal fusion by uncertainty compensation with application to audiovisual speech recognition. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(3):423–435, 2009.

E. K. Patterson, S. Gurbuz, Z. Tufekci, and J. N. Gowdy. Cuave: A new audio-visual database for multimodal human-computer interface research. In *International Conference on Audio Speech and Signal Processing*, pages 2017–2020, 2002.

H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Stat.*, 22: 400–407, 1951.

R. Salakhutdinov and G. E. Hinton. Replicated softmax: an undirected topic model. In *Advances in Neural Information Processing Systems 22*, pages 1607–1614, 2009a.

R. R. Salakhutdinov and G. E. Hinton. Deep Boltzmann machines. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, volume 12, 2009b.

P. Smolensky. Information processing in dynamical systems: Foundations of harmony theory. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1*, pages 194–281. MIT Press, Cambridge, MA, USA, 1986.

N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *Eurpean Conference on Computer Vision*, 2010.

T. Tieleman. Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1064–1071, 2008.

A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

J. Verbeek, M. Guillaumin, T. Mensink, and C. Schmid. Image annotation with TagProp on the MIRFLICKR set. In *11th ACM International Conference on Multimedia Information Retrieval*, pages 537–546, 2010.

P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1096–1103, 2008.

M. Welling, M. Rosen-Zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In *Advances in Neural Information Processing Systems 18*, pages 1481–1488, 2005.

E. P. Xing, R. Yan, and A. G. Hauptmann. Mining associated text and images with dual-wing harmoniums. In *Uncertainty in Artificial Intelligence (UAI)*, pages 633–641. AUAI Press, 2005.

L. Younes. Parameter inference for imperfectly observed Gibbsian fields. *Probability Theory Rel. Fields*, 82:625–645, 1989.

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

A. L. Yuille. The convergence of contrastive divergences. In *Advances in Neural Information Processing Systems 17*, 2004.

G. Zhao, M. Barnard, and M. Pietikainen. Lipreading with local spatiotemporal descriptors. *IEEE Transactions on Multimedia*, 11(7):1254–1265, 2009.