

IMPERIAL COLLEGE LONDON

DEPARTMENT OF COMPUTING

HiPEDS — LITERATURE SURVEY

A Tutorial Survey on Multimodal Deep Learning

Author:
Daniel COELHO DE CASTRO

Supervisor:
Dr. Ben GLOCKER

March 2, 2016

Contents

1	Introduction	3
2	Background	5
2.1	Energy-Based Models	5
2.2	Hopfield Networks	6
2.3	Boltzmann Machines	6
2.4	Restricted Boltzmann Machines	6
3	Restricted Boltzmann Machines	8
3.1	Generative RBMs	8
3.1.1	Replicated softmax	8
3.1.2	Gaussian RBM	9
3.1.3	Covariance models	9
3.1.4	Learning	11
3.1.5	Evaluation	14
3.1.6	Rectified linear units	14
3.1.7	Weight-sharing	15
3.2	Predictive RBMs	18
3.2.1	Discriminative RBM	18
3.2.2	Conditional RBM	19
4	Deep Models	20
4.1	Deep Belief Networks	20
4.1.1	Convolutional DBN	21
4.2	Deep Boltzmann Machines	22
4.2.1	Variational inference	23
4.2.2	Approximate maximum likelihood training	24
4.2.3	Multi-prediction training	24
4.3	Deep Energy Models	25
5	Multimodal Learning	26
5.1	Models	26
5.2	Learning	27
5.3	Application examples	27

6 Discussion	30
6.1 Case study	32
6.2 Future directions	33
References	35
A Derivations	44
A.1 RBM log-likelihood gradient	44
A.2 DBM mean-field fixed-point equations	45

Chapter 1

Introduction

Different data modalities typically reflect distinct properties of the underlying object or process. Instead of processing modalities independently, it is highly desirable to exploit this complementarity to obtain more representative models.

Multimodal learning is closely related to domain adaptation ([Pan and Yang, 2010](#)), a paradigm for transferring knowledge acquired from some data to other data which may come from a different distribution or live in a different feature space entirely. The crucial element for successfully learning in such a setting is having a good representation, which ideally will disentangle all important underlying variables of your data ([Bengio et al., 2012](#)). This means that we will have learned expressive modality-agnostic features, which can then be used for the task of interest or even transferred for a number of different purposes ([Caruana, 1997](#)).

Many multimodal applications have been tackled in vastly different settings: e.g. image and descriptive text ([Srivastava and Salakhutdinov, 2012](#); [Kiros et al., 2014](#)), audio and video for speech and emotion recognition ([Ngiam et al., 2011b](#); [Kahou et al., 2015](#)), semantic labelling of unstructured audio, video and text data from social media ([Fu et al., 2014](#)).

Remote sensing is an example of a mature field which has traditionally been attempting to integrate data of many different natures, such as multiresolution, multitemporal, multi- and hyperspectral and multi-sensor fusion ([Gomez-Chova et al., 2015](#)).

Another discipline that makes extensive use of multimodal data is medical imaging, which relies on different magnetic resonance imaging (MRI) sequences (e.g. T1, T2, T1-gad, FLAIR, PD), computed tomography (CT), diffusion tensor imaging (DTI), functional MRI (fMRI), positron emission tomography (PET) and others. Techniques have long existed for cross-modality image registration ([Hill et al., 1994, 2001](#)), and it is not uncommon to use different modalities to enhance medical image segmentation (e.g. [Geremia et al., 2013](#)).

Neuroscience, particularly, has demonstrated to benefit greatly from multi-

modal analysis (Toga et al., 2006; Friston, 2009). Studies have applied machine learning techniques to investigate brain dynamics (EEG-fMRI) (De Martino et al., 2010), to detect Alzheimer’s disease (MRI-PET) (Hinrichs et al., 2009) and ADHD (MRI-fMRI) (Colby et al., 2012), to name a few.

Novel multimodal medical image analysis methods have recently been developed, for example Schlegl et al. (2015) proposed creating semantic features from the accompanying textual reports to help in segmentation and Hor and Moradi (2015) presented a random forest-based algorithm for multimodal learning, applied to MRI and gene expression data. Cai et al. (2015) described a method for vertebra recognition from MRI and CT, and use a multimodal deep learning model to extract the cross-modal features.

For approaching all these different multimodal applications, there exists a wealth of techniques, such as multiple kernel learning (MKL) (Sonnenburg et al., 2006; McFee and Lanckriet, 2011) and several multimodal methods based on sparse coding (Monaci et al., 2007; Cao et al., 2013; Irie et al., 2013), dimensionality reduction (Lampert and Kroemer, 2010), latent attribute models (Fu et al., 2014), Markov random fields (Kadoury et al., 2011), random forests (Hor and Moradi, 2015) and classic multivariate statistical methods (Sui et al., 2012).

One approach which stands out as one of the most promising, due to its flexibility, expressive power and record-breaking results, is multimodal deep learning (e.g. Ngiam et al., 2011b; Srivastava and Salakhutdinov, 2012, 2013; Sohn et al., 2014).

As a motivating example, let us consider the following simplified hypothetical application: suppose we have a database with brain MRI and PET scans, alongside the corresponding textual clinical reports. Additionally, we are given the ultimate diagnosis for Alzheimer’s disease (AD) for these patients. We wish to build a model to look for signs of AD on new patients based only on the brain scans, for example to aid the radiologist or clinician in deriving diagnostic conclusions.

In this paper, we will review different practical aspects of designing and training restricted Boltzmann machine-based models, from the fundamental concepts to state-of-the-art techniques, in order to build deep multimodal architectures. This will give us the necessary tools to develop a concluding case study based on this example, on Section 6.1.

This survey is organized as follows: Chapter 2 briefly outlines the general framework on which restricted Boltzmann machines are built; Chapter 3 explores several RBM model variants along with aspects of arranging and training these models; in Chapter 4, we discuss different approaches to assembling deep architectures; Chapter 5 covers multimodal strategies and a few application examples from the literature; finally, Chapter 6 concludes with the mentioned case study and some comments on current and future research in the area.

Chapter 2

Background

2.1 Energy-Based Models

As shown in [LeCun et al. \(2006\)](#), many machine learning problems can be formulated in terms of energy models, such as regression, classification, detection, density estimation and others.

In this survey, we will cover some members of the family of Boltzmann machines, which have shown great promise in multimodal learning applications (e.g. [Cai et al., 2015](#); [Srivastava and Salakhutdinov, 2012](#); [Ngiam et al., 2011b](#); [Sohn et al., 2014](#)).

It is a class of probabilistic graphical models which, for a given state $\mathbf{s}$ of the network, define an energy function $E(\mathbf{s})$. This energy, in turn, determines the probability of that state according to the Boltzmann (or Gibbs) distribution¹

$$\mathbb{P}(\mathbf{s}) = \frac{1}{Z} e^{-E(\mathbf{s})}, \quad Z = \sum_{\mathbf{s}} e^{-E(\mathbf{s})} \quad (2.1)$$

The normalizing constant, Z , called the partition function, is typically intractable, because it involves a sum over all the exponentially many possible states of the model. As we will see, this entails some difficulties that need to be addressed when using these models in practice.

A related concept is that of products of experts (PoE) ([Hinton, 2002](#)), which define the probability as a normalized product of simple distributions (experts).² They can also be regarded as energy-based models, with an energy defined as minus the sum of the log-experts:

$$\mathbb{P}(\mathbf{s}) = \frac{1}{Z} \prod_i f_i(\mathbf{s}) = \frac{1}{Z} e^{-E(\mathbf{s})}, \quad E(\mathbf{s}) = - \sum_i \log f_i(\mathbf{s})$$

¹For the models discussed here, the inverse temperature parameter β of the physical Boltzmann distribution is set to one: $\mathbb{P}(\mathbf{s}) = e^{-\beta E(\mathbf{s})} / \sum_{\mathbf{s}'} e^{-\beta E(\mathbf{s}')}$

²Compare with mixture models, which are weighted *sums* of distributions.

2.2 Hopfield Networks

A Hopfield network is a fully connected binary neural network. The state of a unit is updated according to the following rule (Hopfield, 1982):

$$s_i \leftarrow \begin{cases} 1, & \text{if } \sum_j W_{ij}s_j + a_i \leq 0 \\ 0, & \text{otherwise} \end{cases}, \quad (2.2)$$

where W_{ij} are the weights of the connections and a_i is the bias of each unit. We define the nodes to have no connection to themselves ($W_{ii} = 0$) and the connections to be symmetrical ($W_{ij} = W_{ji}$).

Every configuration of the network has an associated potential energy, defined as

$$E(\mathbf{s}) = -\mathbf{s}^\top \mathbf{W} \mathbf{s} - \mathbf{a}^\top \mathbf{s} \quad (2.3)$$

This formulation belongs to the class of Ising models, a canonical type of Markov random field which is commonly used in Physics to describe spin glass lattices. It can be shown that stable (equilibrium) states of such a network correspond to local minima of the energy function (Little, 1974).

2.3 Boltzmann Machines

Boltzmann machines can be seen as a stochastic version of Hopfield networks. Additionally, we distinguish a set of visible units, which receive external input, and hidden units, which represent latent, unobserved variables (Ackley et al., 1985; Hinton and Sejnowski, 1986). Their behaviour is also governed by the energy function in Eq. (2.3) and the equilibrium probability of a given global state is given by Eq. (2.1).

For any given state of the network, the probability of activation of a single unit i given the states of the others is

$$\mathbb{P}(s_i = 1 \mid \mathbf{s}_{\setminus i}) = \sigma\left(\sum_{j \neq i} W_{ij}s_j + a_i\right), \quad (2.4)$$

where $\sigma(x) = 1/(1 + e^{-x})$ is the logistic sigmoid function.

2.4 Restricted Boltzmann Machines

Since learning in a sizeable BM is intractable due to the exponentially many terms found when marginalizing for a single unit, we commonly restrict its architecture to a bipartite graph, with no visible-to-visible or hidden-to-hidden connections (Fig. 2.1a). This type of model was originally called a harmonium, introduced by Smolensky (1986).

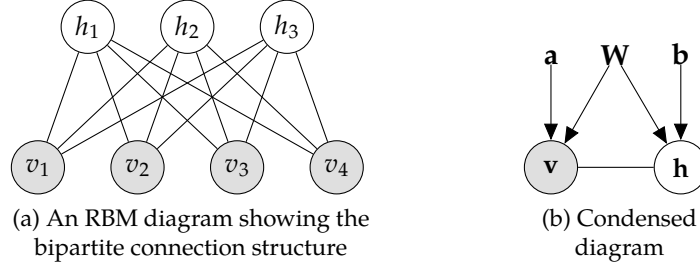


Figure 2.1: An RBM

In the restricted model, the energy becomes

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{a}^\top \mathbf{v} - \mathbf{b}^\top \mathbf{h}. \quad (2.5)$$

And we have, again, the Boltzmann probability distribution:

$$\mathbb{P}(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-E(\mathbf{v}, \mathbf{h})}, \quad Z = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \quad (2.6)$$

The key property of this model is that, given the states of one layer, the states of each unit in the other are independent of one another:

$$\mathbb{P}(\mathbf{v}|\mathbf{h}) = \prod_i \mathbb{P}(v_i|\mathbf{h}), \quad \mathbb{P}(\mathbf{h}|\mathbf{v}) = \prod_j \mathbb{P}(h_j|\mathbf{v}) \quad (2.7)$$

Because of this independence, it is possible to perform efficient learning by alternately sampling all hidden and all visible units in parallel (see Section 3.1.4).

Similarly to the BM's activation probabilities in Eq. (2.4), for an RBM we have

$$\mathbb{P}(v_i = 1|\mathbf{h}) = \sigma([\mathbf{W}\mathbf{h}]_i + a_i), \quad (2.8)$$

$$\mathbb{P}(h_j = 1|\mathbf{v}) = \sigma([\mathbf{W}^\top \mathbf{v}]_j + b_j). \quad (2.9)$$

Additionally, again in analogy to Physics, we can define a free energy function, which, for binary hidden units, is (Bengio, 2009; Mnih et al., 2012):

$$F(\mathbf{v}) = -\log \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} = -\sum_j \log(1 + e^{[\mathbf{W}^\top \mathbf{v}]_j + b_j}) - \mathbf{a}^\top \mathbf{v}, \quad (2.10)$$

such that the likelihood of the inputs can be written simply as

$$\mathbb{P}(\mathbf{v}) = \frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} = \frac{1}{Z} e^{-F(\mathbf{v})}, \quad Z = \sum_{\mathbf{v}} e^{-F(\mathbf{v})} = \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \quad (2.11)$$

These RBMs with binary inputs are sometimes also referred to as Bernoulli or binomial RBMs, to distinguish from other types of variables.

Chapter 3

Restricted Boltzmann Machines

3.1 Generative RBMs

In real-world settings, data is rarely binary, therefore in this section we will cover a few extensions of the Bernoulli RBM to different kinds of data, such as multinomial and several models for continuous variables.

3.1.1 Replicated softmax

When modelling text, a common simplifying approach is to disregard structure and order and consider just the number of occurrences of each word, which is known as a bag-of-words or unigram model. To deal with such input of multinomial nature, the replicated softmax RBM (RS) model was developed ([Salakhutdinov and Hinton, 2009b](#); [Srivastava et al., 2013](#)):

$$E(\mathbf{v}, \mathbf{h}) = -\mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{a}^\top \mathbf{v} - M \mathbf{b}^\top \mathbf{h}, \quad (3.1)$$

where $M = \sum_i v_i$ is the total number of words in the document.

The training procedure is the same as for a Bernoulli RBM (see Section [3.1.4](#)), and the conditional probabilities are given by

$$\mathbb{P}(v_i = 1 | \mathbf{h}) = \frac{\exp([\mathbf{W} \mathbf{h}]_i + a_i)}{\sum_{i'} \exp([\mathbf{W} \mathbf{h}]_{i'} + a_{i'})}, \quad (3.2)$$

$$\mathbb{P}(h_j = 1 | \mathbf{v}) = \sigma([\mathbf{W}^\top \mathbf{v}]_j + M b_j). \quad (3.3)$$

3.1.2 Gaussian RBM

One can also deal with continuous inputs at the visible layer by incorporating Gaussian noise into the energy function (Hinton, 2010; Hinton et al., 2012a)¹:

$$\begin{aligned} E(\mathbf{v}, \mathbf{h}) &= \sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_{ij} \frac{v_i}{\sigma_i} W_{ij} h_j - \sum_j b_j h_j \\ &= \frac{1}{2}(\mathbf{v} - \mathbf{a})^\top \mathbf{\Sigma}^{-1}(\mathbf{v} - \mathbf{a}) - \mathbf{v}^\top \mathbf{\Sigma}^{-\frac{1}{2}} \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{h} \end{aligned} \quad (3.4)$$

where $\mathbf{\Sigma}$ is a diagonal covariance matrix with σ_i^2 entries.

In this case, the conditional probabilities of v_i and h_j are given by

$$\mathbb{P}(v_i | \mathbf{h}) = \mathcal{N}(\sigma_i [\mathbf{W} \mathbf{h}]_i + a_i, \sigma_i^2), \quad (3.5)$$

$$\mathbb{P}(h_j = 1 | \mathbf{v}) = \sigma([\mathbf{W}^\top \mathbf{\Sigma}^{-\frac{1}{2}} \mathbf{v}]_j + b_j). \quad (3.6)$$

Krizhevsky (2009) derives the update rules for the variances, which can be estimated using the learning procedures described later. Another way to estimate σ_i^2 is to re-parametrize it as $z_i = \log \sigma_i^2$, naturally enforcing the positivity constraint on the variance (Cho et al., 2011).

In both cases, estimation can be quite costly. An easier and widely adopted alternative is to first normalize or whiten the inputs and simply consider unit variances (Hinton, 2010):

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2} \|\mathbf{v} - \mathbf{a}\|^2 - \mathbf{v}^\top \mathbf{W} \mathbf{h} - \mathbf{b}^\top \mathbf{h} \quad (3.7)$$

Bernoulli and Gaussian are the most commonly used distributions for the inputs, but more general distributions can also be considered. See Welling et al. (2004) for a discussion on how we can incorporate general exponential family distributions.

3.1.3 Covariance models

Even though Gaussian RBMs provide much richer representations of real-world data than Bernoulli RBMs, they are still very restrictive in that they model only the conditional means of the inputs. This limits their usefulness for modelling highly structured data such as images. For this reason, several new models were proposed which explicitly attempt to model the conditional covariance of the inputs.

¹This definition varies slightly among authors. Some use v_i/σ_i^2 in the second term (e.g. Cho et al., 2011), some use a single parameter σ (i.e. isotropic, e.g. Lee et al., 2008), some separate $\mathbf{a}$ into a bias term (e.g. Lee et al., 2009a).

Product of Student's t-distributions

One of the earliest such models is the product of Student's t-distributions (PoT) model (Welling et al., 2002). While not strictly an RBM, the PoT model is also energy-based – more specifically, a product of experts model (PoE) –, so most considerations in this text still apply. The conditional distribution of the visible units is a zero-mean Gaussian with full covariance matrix, but, unlike RBMs, its hidden units are conditionally Gamma-distributed.

Covariance RBM

Another covariance model is the covariance RBM (cRBM) (Ranzato et al., 2010a), which introduces three-way factors between hidden units and pairs of visible units. The conditional distribution of the visibles is then again multivariate Gaussian, with zero mean and a full covariance matrix modulated by the hidden units.

mPoT and mcRBM

PoT and cRBM are unable to model a non-zero conditional mean for the visible units. However, this can be achieved by combining them with a GRBM, yielding the mean-product of Student's t-distributions (mPoT) (Ranzato et al., 2010b) and the mean-covariance RBM (mcRBM) (Ranzato and Hinton, 2010) models, respectively. While mcRBMs have produced some state-of-the-art results (e.g. Dahl et al., 2010), their training is problematic, and they seem to have been supplanted by the mPoT model (Bengio et al., 2012). In a texture synthesis experiment, Kivinen and Williams (2012) show that a mean-less PoT completely failed to capture the subtleties of the patterns, and was vastly outperformed by mPoT and even GRBM.

Spike-and-slab RBM

A further member of this class is the spike-and-slab RBM (ssRBM), introduced in Courville et al. (2011a). Differently from the previous models, the ssRBM associates to each binary latent variable h_j ('spike') a continuous latent variable s_j ('slab'). The state of each hidden unit, given by their product, is then either real-valued or exactly zero. Here, again, the spikes directly modulate the conditional covariance of the visible variables (which is non-diagonal after marginalizing out the slabs).

Critically, both $\mathbb{P}(\mathbf{h}|\mathbf{v})$, $\mathbb{P}(\mathbf{s}|\mathbf{v}, \mathbf{h})$ and $\mathbb{P}(\mathbf{v}|\mathbf{h}, \mathbf{s})$ are factorial, which means that we can perform very efficient block Gibbs sampling, in the same manner as standard RBMs. Unfortunately, the proposed parametrization is too general, and constraints need to be imposed on the parameters so that they define positive definite covariances and thus proper distributions (Courville et al.,

2011b). [Courville et al. \(2011a, 2014\)](#) show that alternatively it suffices to limit the domain of the visible variables.

Further extensions of the ssRBM to discrete and to sparse spike-and-slab inputs were presented in [Courville et al. \(2014\)](#). The latter allows directly stacking ssRBMs to obtain deeper models (ssDBNs, see Chapter 4), which the authors show that produces superior results to a single ssRBM.

Table 3.1: Comparison of generative RBMs

Model	Input	Mean	Cov.	Sampling*	Main references
RBM	Binary	✓	✗	BG	Hinton (2010)
RS	Multinomial	✓	✗	BG	Salakhutdinov and Hinton (2009b)
GRBM	Continuous	✓	✗	BG	Hinton (2010)
PoT		✗	✓	HMC	Welling et al. (2002)
cRBM		✗	✓	HMC	Ranzato et al. (2010a)
mPoT		✓	✓	HMC	Ranzato et al. (2010b)
mcRBM		✓	✓	HMC	Ranzato and Hinton (2010)
ssRBM		✓	✓	BG	Courville et al. (2014)

*BG = Block Gibbs MCMC, HMC = Hybrid Monte Carlo (see Section 3.1.4)

3.1.4 Learning

The classic approach to training statistical models is maximum likelihood (ML), which here will involve the set of parameters, θ , of the energy function. The optimization can be carried out in a stochastic gradient descent (SGD) fashion, calculating the gradient over ‘mini-batches’ of data points, instead of the whole training set.

Taking the derivatives of the log-likelihood w.r.t. the model parameters yields (Appendix A.1)

$$\nabla_{\theta} \log \mathbb{P}(\mathbf{v}) = -\mathbb{E}_{\mathbb{P}(\mathbf{h}|\mathbf{v})}[\nabla_{\theta} E(\mathbf{v}, \mathbf{h})] + \mathbb{E}_{\mathbb{P}(\mathbf{v}, \mathbf{h})}[\nabla_{\theta} E(\mathbf{v}, \mathbf{h})] \quad (3.8)$$

The two terms are commonly referred to as positive and negative phase, respectively. Intuitively, this objective function will attempt to increase the likelihood (lower the energy) of the true observed data while lowering the likelihood (resp. increase the energy) of confabulations randomly generated by the model.² It can also be interpreted as minimizing the Kullback-Leibler divergence between the data distribution and the model’s equilibrium distribution ([Teh et al., 2003](#); [Hinton et al., 2006a](#)).

²This can be made evident if we rewrite the log-likelihood gradient in terms of the free energy: $\nabla_{\theta} \log \mathbb{P}(\mathbf{v}) = -\nabla_{\theta} F(\mathbf{v}) + \mathbb{E}_{\mathbb{P}(\mathbf{v})}[\nabla_{\theta} F(\mathbf{v})]$.

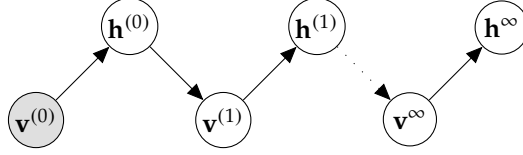


Figure 3.1: An illustration of the RBM Markov chain

For example, in a Bernoulli RBM, $-\nabla_{\mathbf{W}}E(\mathbf{v}, \mathbf{h}) = \mathbf{v}\mathbf{h}^\top$, $-\nabla_{\mathbf{a}}E(\mathbf{v}, \mathbf{h}) = \mathbf{v}$ and $-\nabla_{\mathbf{b}}E(\mathbf{v}, \mathbf{h}) = \mathbf{h}$, so we have the following update equations:

$$\begin{cases} \Delta \mathbf{W} &= \eta(\langle \mathbf{v}\mathbf{h}^\top \rangle_0 - \langle \mathbf{v}\mathbf{h}^\top \rangle_\infty) \\ \Delta \mathbf{a} &= \eta(\langle \mathbf{v} \rangle_0 - \langle \mathbf{v} \rangle_\infty) \\ \Delta \mathbf{b} &= \eta(\langle \mathbf{h} \rangle_0 - \langle \mathbf{h} \rangle_\infty) \end{cases} \quad (3.9)$$

where, for compactness, $\langle \cdot \rangle_0$ means expectation w.r.t. the data distribution, $\mathbb{P}(\mathbf{h}|\mathbf{v}^{(0)})$, $\langle \cdot \rangle_\infty$ means expectation w.r.t. the model distribution, $\mathbb{P}(\mathbf{v}, \mathbf{h})$, and η is the learning rate.

Unfortunately, calculating the full joint model expectation of the negative phase is intractable, because it involves summing or integrating over all possible states of the visible and hidden variables. In theory, we could run an infinite Gibbs sampling Markov chain³ until the equilibrium distribution was reached, and then use Monte Carlo to estimate the expectation (Fig. 3.1).

Contrastive divergence

In practice, this MCMC (Markov chain Monte Carlo) method need only be run for a finite number of iterations k : an approximation known as contrastive divergence (CD_k) (Hinton, 2002; Teh et al., 2003). As one might suspect, truncating the chain is a biased approximation, but this bias was proven to be small enough for practical applications (Carreira-Perpiñán and Hinton, 2005).

CD_1 (i.e. using the first reconstruction of the inputs, $\mathbf{v}^{(1)}$ and $\mathbf{h}^{(1)}$) is fast, has low variance, and is the most popular training algorithm for RBMs (Tieleman, 2008). A detailed guide on the practicalities of training RBMs can be found in Hinton (2010).

Improved results may be obtained with persistent contrastive divergence (PCD)⁴, which approximates the model's expectations by keeping a running Gibbs sampling chain (or a number of chains), instead of resetting it between parameter updates (Tieleman, 2008). Tieleman and Hinton (2009) have proposed a further refinement of PCD, called fast-weights PCD (FPCD), which promotes better mixing in the Gibbs chain and thus accelerates learning.

³As mentioned previously, because of the RBM's conditional independence property (Eq. (2.7)), we can perform block Gibbs sampling, i.e. starting with $\mathbf{v}^{(0)}$ clamped to the value of the data point and alternately sampling $h_j^{(t)} \sim \mathbb{P}(h_j|\mathbf{v}^{(t)})$ and $v_i^{(t)} \sim \mathbb{P}(v_i|\mathbf{h}^{(t-1)})$.

⁴Also known as stochastic approximation procedure (SAP) (e.g. Salakhutdinov and Hinton, 2009a) or stochastic maximum likelihood (SML) (e.g. Bengio et al., 2012).

[Fischer and Igel \(2014\)](#) provide a comparative overview of these techniques and discuss some of their statistical properties. They further draw a comparison with parallel tempering (PT) ([Salakhutdinov, 2009](#); [Cho et al., 2011](#)), a promising sampling method which involves running additional Markov chains which sample from tempered (i.e. smoothed) distributions, and has been shown to accelerate the mixing of the chains and improve the gradient approximations.

Contrastive backpropagation

For some models (e.g. PoT, cRBM and deep energy models (Section 4.3)), not all of the distributions factorize, so Gibbs sampling itself becomes challenging. In these cases, one could alternatively sample directly from the non-factorial likelihood by applying hybrid Monte Carlo (HMC) ([Duane et al., 1987](#); [Neal, 1993](#)) on the free energy. This strategy is known as contrastive backpropagation ([Hinton et al., 2006b](#)).

Score matching

A further alternative to all these sampling-based methods is score matching ([Hyvärinen, 2005, 2007](#)). This deterministic approach is particularly well-suited for energy-based models (or more generally any non-normalized probability model) because it avoids dealing with the partition function Z entirely, depending only on derivatives of the (free) energy w.r.t. the data ([Bengio, 2009](#); [Kingma and LeCun, 2010](#)).

Additional options

Sparsity Additionally to maximizing the log-likelihood, we can enforce sparsity of the hidden units' activations by adding a regularization term penalizing their deviations from a (low) fixed level ρ ([Lee et al., 2008](#)):

$$\min_{\theta} \left\{ - \sum_{n=1}^N \log \mathbb{P}(\mathbf{v}^n) + \lambda \sum_j \left(\rho - \frac{1}{N} \sum_{n=1}^N \mathbb{E}[h_j | \mathbf{v}^n] \right)^2 \right\}, \quad (3.10)$$

where n loops over the training instances.

The log-likelihood term can be optimized as before, e.g. using CD. For the second term, [Lee et al. \(2008\)](#) recommend doing standard gradient descent, but updating only the bias terms b_j , which relate directly to the units' activations.

Dropout Dropout is another kind of regularization procedure, first introduced in the context of feedforward neural networks ([Hinton et al., 2012b](#)). This technique prescribes that, during training, each hidden unit has a certain probability of being inactive, not participating in parameter updates. In

practice, this means that a different network architecture is sampled for each training input presented, resulting in a sort of ensemble learning. Because each unit is encouraged not to depend on the presence of the others, this prevents co-adaptation, which proves to be very effective for generalization.

[Srivastava et al. \(2014\)](#) proposed to extend dropout to RBM training as well. One simply associates a binary random variable to each hidden unit – driven by a global probability hyperparameter p –, governing whether it should be retained or dropped from the model. After training, all weights are scaled down by a factor of p , emulating their expected values during training. The authors show empirically that dropout also promotes sparsity of the learned features.

3.1.5 Evaluation

When building probabilistic models, one is often interested in comparing different architectures, settings of hyperparameters and training strategies, and this evaluation usually relies on the (log-) likelihood that the model assigns to the test data. For RBM-based models, unfortunately, this would involve calculating the intractable partition function.

[Salakhutdinov and Murray \(2008\)](#) showed that ratios of RBM partition functions – and ultimately the value of the partition function itself – can be estimated efficiently using annealed importance sampling (AIS) ([Neal, 2001](#)). They showed how AIS can be coupled with variational inference to estimate lower bounds in DBNs and [Salakhutdinov and Hinton \(2009a\)](#) extended it to DBMs (see Chapter 4). AIS has since been used as a reliable evaluation method in several applications (e.g. [Salakhutdinov and Hinton, 2009b](#); [Salakhutdinov and Larochelle, 2010](#); [Hinton and Salakhutdinov, 2012](#); [Srivastava et al., 2013](#); [Ngiam et al., 2011a](#)).

[Desjardins et al. \(2011\)](#) proposed an alternative way of estimating the partition function, which allows one to track its value during training. This real-time monitoring could be useful for early stopping (i.e. to control overfitting) and might qualify as a faster substitute for AIS ([Bengio et al., 2012](#)).

3.1.6 Rectified linear units

To give an RBM's units more expressive power, [Teh and Hinton \(2001\)](#) first proposed creating N replicas with same weights and bias. This enables them to represent not two, but $N + 1$ different levels, without changing the training procedure.

[Nair and Hinton \(2010\)](#) expand on this idea, presenting a formulation involving an infinite number of offset copies, together with a fast approximation which they named noisy rectified linear unit (NReLU). Letting $x = [\mathbf{W}^T \mathbf{v}]_j + b_j$, a NReLU replaces the usual activation, $\sigma(x)$, by $\max(0, x + \mathcal{N}(0, \sigma(x)))$.

The use of rectified linear units has since greatly contributed to some state-of-the-art results obtained by feedforward neural networks (e.g. [Krizhevsky et al., 2012](#)), notably because rectification is much faster to compute than the conventional logistic sigmoid. Furthermore, half-wave rectification enjoys some biological plausibility, because it is said to model the excitation behaviour of simple cells in the human visual cortex ([Malik and Perona, 1990](#)).

3.1.7 Weight-sharing

Convolution

Convolutional models are well-known to excel at computer vision tasks, because they incorporate the knowledge that image features should be local and translation-invariant. The idea of convolutional neural networks dates back to [Fukushima \(1980\)](#)'s Neocognitron.

[Lee et al. \(2009a\)](#) and [Desjardins and Bengio \(2008\)](#) independently proposed a convolutional form of RBM (CRBM). Here, $\mathbf{V}$ is the input image⁵ and the hidden units $\mathbf{H}$ are partitioned into K groups, which represent different feature maps. Each group has shared weight matrix $\mathbf{W}^k$ and bias b_k . Visible units share a single bias a . The energy function is defined as⁶

$$E(\mathbf{V}, \mathbf{H}) = - \sum_{k=1}^K \left(\mathbf{H}^k \bullet (\tilde{\mathbf{W}}^k * \mathbf{V}) - b_k \sum_{ij} H_{ij}^k \right) - a \sum_{ij} V_{ij} \quad (3.11)$$

and the conditional probabilities are given by

$$\mathbb{P}(V_{ij} = 1 | \mathbf{H}) = \sigma \left(\sum_k [\mathbf{W}^k \hat{*} \mathbf{H}^k]_{ij} + a \right), \quad (3.12)$$

$$\mathbb{P}(H_{ij}^k = 1 | \mathbf{V}) = \sigma ([\tilde{\mathbf{W}}^k * \mathbf{V}]_{ij} + b_k). \quad (3.13)$$

Intuitively, each $\tilde{\mathbf{W}}^k$ is a filter mask (kernel), meaning $\mathbb{P}(H_{ij}^k = 1 | \mathbf{V})$ is simply a function of the image response to filter k at pixel (i, j) .

For continuous inputs, we can simply add a quadratic term to the energy function, similarly to the GRBM ([Lee et al., 2009b](#)). The conditional distributions of the hidden units remain unchanged, but the visible units' become

$$\mathbb{P}(V_{ij} | \mathbf{H}) = \mathcal{N} \left(\sum_k [\mathbf{W}^k \hat{*} \mathbf{H}^k]_{ij} + a, 1 \right). \quad (3.14)$$

⁵Here, we assumed two-dimensional input, but CRBMs have also been applied to one-dimensional data such as audio ([Lee et al., 2009b](#)). One could also envision three-dimensional input, such as medical images (e.g. MRI).

⁶We adopt the notation of [Lee et al. \(2009a\)](#): $\bullet$ for element-wise product and summation (i.e. $\mathbf{A} \bullet \mathbf{B} = \text{tr}[\mathbf{A}^\top \mathbf{B}]$) and tilde for flipping rows and columns. Additionally, let $*$ denote valid convolution (i.e. computing only the fully-defined part), and $\hat{*}$ denote full convolution (i.e. with padding at the borders), such that $\mathbf{H}^k \bullet (\tilde{\mathbf{W}}^k * \mathbf{V}) = \mathbf{V} \bullet (\mathbf{W}^k \hat{*} \mathbf{H}^k)$.

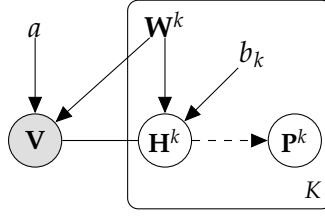


Figure 3.2: A CRBM with a probabilistic max-pooling layer

Desjardins and Bengio (2008) show how to adapt contrastive divergence for the convolutional setting.

Probabilistic max-pooling The conventional way of downscaling convolutional layers, borrowed from convolutional neural networks, is by (deterministic) max-pooling (Norouzi et al., 2009). This kind of operation brings to higher layers some amount of local translation invariance and also greatly reduces their computational load.

In order to better interface with the stochastic nature of CRBMs, Lee et al. (2009a) introduced a probabilistic form of max-pooling. We add to the energy function the constraint that at most a single unit in a pooling block, $B_\alpha = \{(i, j) | H_{ij} \in \text{block } \alpha\}$, should be active.

This results is a multinomial distribution, representing the fact that we expect only one unit to activate within each block α . Let $I(H_{ij}^k) = [\tilde{\mathbf{W}}^k * \mathbf{V}]_{ij} + b_k$ be the bottom-up signal from the input layer. The activation of the convolutional units becomes

$$\mathbb{P}(H_{ij}^k = 1 | \mathbf{V}) = \frac{\exp I(H_{ij}^k)}{1 + \sum_{(i', j') \in B_\alpha} \exp I(H_{i'j'}^k)}. \quad (3.15)$$

Finally, we may sample the units in the pooling layer according to the following distribution:

$$\mathbb{P}(P_\alpha^k = 0 | \mathbf{V}) = \frac{1}{1 + \sum_{(i', j') \in B_\alpha} \exp I(H_{i'j'}^k)} \quad (3.16)$$

Taylor et al. (2010) extend this model for processing video data. Their model can learn representations from pairs of successive images, effectively capturing the optical flow (i.e. motion features) in the frame sequence.

Roth and Black (2009) proposed a convolutional architecture for products of experts (PoE, of which PoT is a notable example), called field of experts (FoE). Kingma and LeCun (2010) have used FoE in image denoising and Kivinen and Williams (2012) investigated its application in texture synthesis.

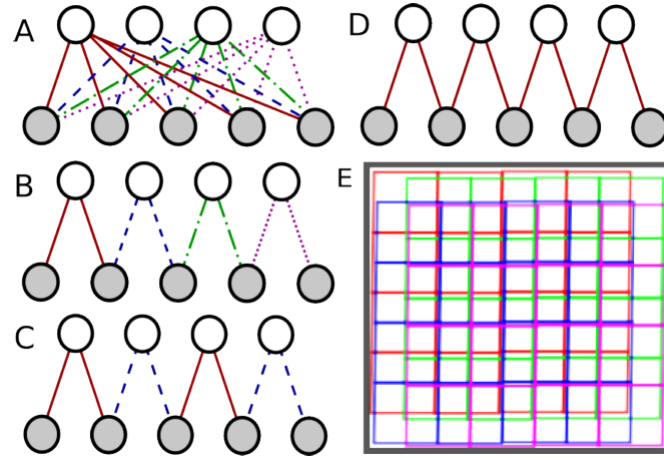


Figure 3.3: Connectivity patterns: global (A), local (B), tiled-convolutional (C) and convolutional (D). (E) illustrates a two-dimensional tiled convolution (from [Ranzato et al., 2010b](#))

Tiled convolution

Because they share a single set of parameters across the whole input, convolutional architectures are limited in what they can represent. The learned features are said to be translation-equivariant, in that a translation of the input equals a translation of the feature map, but the only invariance built into this model is that introduced by pooling ([Nair and Hinton, 2010](#)). Moreover, the representation is highly redundant and the hidden units are strongly correlated ([Ranzato et al., 2010b](#)).

More flexibility may be achieved by somewhat relaxing the weight-sharing, but maintaining the local connections. Using periodic replications of weights in a tiled configuration allows the model to learn more complex invariants, such as rotation and scale ([Gregor and LeCun, 2010](#); [Le et al., 2010](#)).

Tiling comes with an additional parameter, T , which is the size of the tiles, i.e. the period of replication of the filters. Note that tiled convolution reduces to standard convolution when $T = 1$, and turns into simple local connectivity when $T = N$ ([Le et al., 2010](#)). All of these weight-sharing schemes can naturally be extended to two, three or more dimensions.

Table 3.2: Number of parameters in each connectivity pattern

Pattern	# parameters	MRI example
Global	N^{2D}	$\sim 10^{16}$
Local	$(RN)^D$	$\sim 10^{11}$
Tiled-convolutional	$(RT)^D$	10^6
Convolutional	R^D	10^3

To get some insight into the savings in number of parameters, let us consider the following case: a D -dimensional input of size N in each dimension and feature map of the same size. Additionally, let R be the size of the receptive field along each dimension. The numbers of parameters for the different connectivity patterns are given in Table 3.2, with biases omitted for clarity.

It is clear that the convolutional and tiled-convolutional schemes scale much better to high dimensions. For example, a typical MRI volume has $512 \times 512 \times 512$ voxels ($D = 3$, $N = 512$). Table 3.2 shows, in the last column, the number of parameters with $10 \times 10 \times 10$ receptive fields and tiles ($R = T = 10$). In this setting, learning with global or even local connectivity would be outright infeasible, whereas the number of parameters of the other two is far more tractable.

Tiled convolution has met significant success with some of the models described here, including GRBM (Kivinen and Williams, 2012), mPoT (TmPoT, Ranzato et al., 2010b; Kivinen and Williams, 2012) and ssRBM (TssRBM, Luo et al., 2012).

3.2 Predictive RBMs

All RBM types described so far, together with the unsupervised ML training procedures, have been unsupervised *generative* models, which is to say they attempt to model the likelihood of the visible variables, $\mathbb{P}(\mathbf{v})$. They are therefore useful for extracting meaningful representations of the input data. If one is interested in a discriminative task, the extracted features can typically be fed into a separate classification model, e.g. logistic regression, SVM etc.

3.2.1 Discriminative RBM

In the above case, however, the RBM would have been trained oblivious to the ultimate goal of classification. Larochelle and Bengio (2008) proposed a new type of discriminative RBM (DRBM), which explicitly optimizes $\mathbb{P}(y|\mathbf{v})$, where y is a class label (Fig. 3.4a). The authors showed that the gradient of this discriminative objective function, unlike that of the log-likelihood, can be computed exactly.

They also presented hybrid DRBMs, which strike a balance between optimizing this predictive distribution and the generative joint likelihood, $\mathbb{P}(\mathbf{v}, y)$. Furthermore, a third form of training is introduced for leveraging unlabelled data, effectively enabling semi-supervised learning.

3.2.2 Conditional RBM

Taking the prediction concept one step further, conditional RBMs introduce an external set of *context* variables $\mathbf{u}$, and attempt to model the conditional distribution $\mathbb{P}(\mathbf{v}|\mathbf{u})$ (Fig. 3.4b). This makes it possible to predict generic structured outputs.

Conditional RBMs have been used in many applications, e.g. to model temporal data, using information from previous frames as context for the current frame (Taylor et al., 2007; Taylor and Hinton, 2009), and were usually trained with CD.

More recently, Mnih et al. (2012) argued that CD is not optimal for conditional models, and introduced a training procedure for them based on the generalized perceptron loss (LeCun et al., 2006) with a CD-like sampling approximation, called CD-PercLoss.

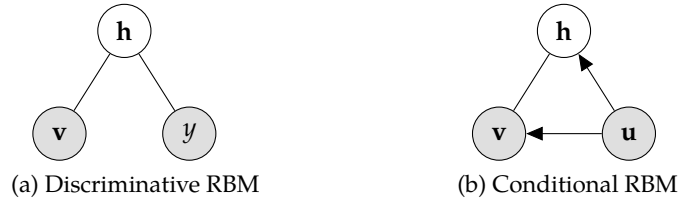


Figure 3.4: Predictive RBMs

Chapter 4

Deep Models

A much more powerful model can be built by stacking RBM hidden layers, because deeper layers can represent higher-order correlations between the input variables ([Deng, 2012](#)). Deep architectures have been shown to promote feature re-use, abstraction and invariance ([Bengio et al., 2012](#)).

4.1 Deep Belief Networks

A deep belief network (DBN) is the result of greedy layer-wise training of an RBM stack ([Hinton et al., 2006a](#)). After each RBM is trained, its parameters are frozen, and a new RBM is added on top of it, which is then trained using as inputs either the deterministic activations of the previous layer or samples from its posterior (Fig. 4.1). [Hinton et al. \(2006a\)](#) showed that, under certain assumptions, adding a new layer increases a lower bound on the log-likelihood of the data. The model is left with the undirected RBM on the last layer and a directed path leading to the visible layer (Fig. 4.2b).

Compare with sigmoid belief networks (SBNs), which are fully directed generative graphical models (Fig. 4.2a) ([Neal, 1992](#)). This type of model will not be covered here; the reader is referred to [Gan et al. \(2015\)](#) for details on SBN learning and inference.

Learning a DBN is often used as a pre-training step for building deep feed-forward neural networks, because of the RBM's ability to learn representative features in an unsupervised setting ([Hinton et al., 2012a](#)). After the DBN is trained, the probabilistic framework is dropped. The network is then 'unrolled' into a deep autoencoder and the parameters can be fine-tuned with conventional back-propagation ([Hinton and Salakhutdinov, 2006](#)).

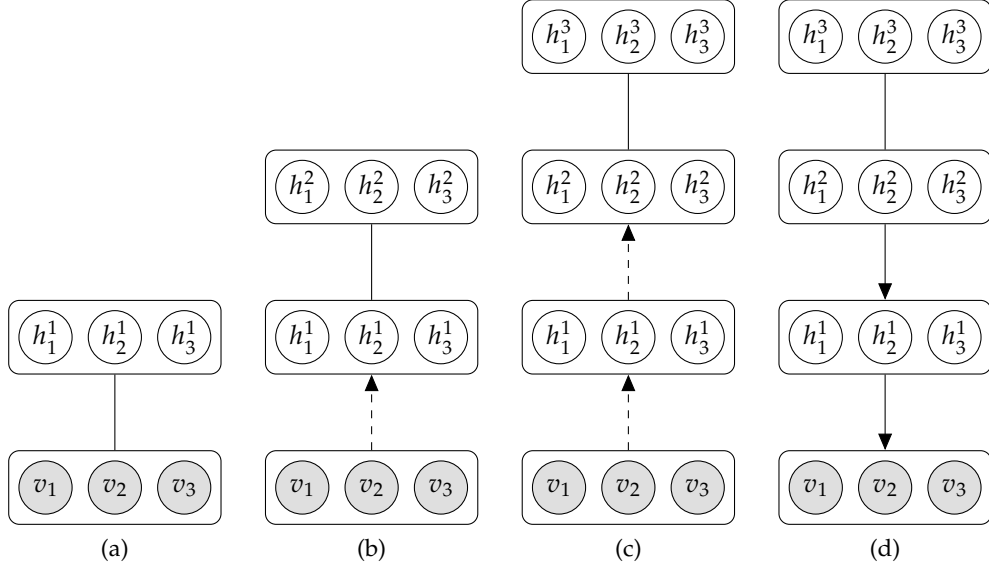


Figure 4.1: Deep belief network training. (a) Training of the first RBM layer. (b), (c) Adding and training deeper layers. (d) Final trained model. Dashed arrows represent deterministic relationships.

4.1.1 Convolutional DBN

Like regular RBMs, CRBM layers can be stacked to obtain more representative shift-invariant features (Lee et al., 2009a). The outputs from one pooling layer are used as input to the next convolution layer. Their deterministic counterparts, deep convolutional neural networks, enjoy great success in object recognition tasks (e.g. Krizhevsky et al., 2012; Szegedy et al., 2014).

Defining the bottom-up signal from the visible layer and the top-down signal from the following convolutional layer, respectively, as

$$I(H_{ij}^k) = [\tilde{\mathbf{W}}^k * \mathbf{V}]_{ij} + b_k \quad \text{and} \quad I(P_\alpha^k) = \sum_{\ell} [\Gamma^{k\ell} \hat{*} \mathbf{H}'^\ell]_\alpha,$$

where $\Gamma^{k\ell}$ and $\mathbf{H}'^\ell$ are the filters and hidden units of the layer above (Fig. 4.3), the activations of the convolutional and pooling layers take the following form:

$$\mathbb{P}(H_{ij}^k = 1 | \mathbf{V}, \mathbf{H}') = \frac{\exp(I(H_{ij}^k) + I(P_\alpha^k))}{1 + \sum_{(i', j') \in B_\alpha} \exp(I(H_{i'j'}^k) + I(P_\alpha^k))} \quad (4.1)$$

$$\mathbb{P}(P_\alpha^k = 0 | \mathbf{V}, \mathbf{H}') = \frac{1}{1 + \sum_{(i', j') \in B_\alpha} \exp(I(H_{i'j'}^k) + I(P_\alpha^k))} \quad (4.2)$$

As the name implies, a CDBN is built with greedy layer-wise training. Additionally, because the model is overcomplete, the authors have used sparse training (Lee et al., 2008) to prevent learning of trivial features (e.g. single

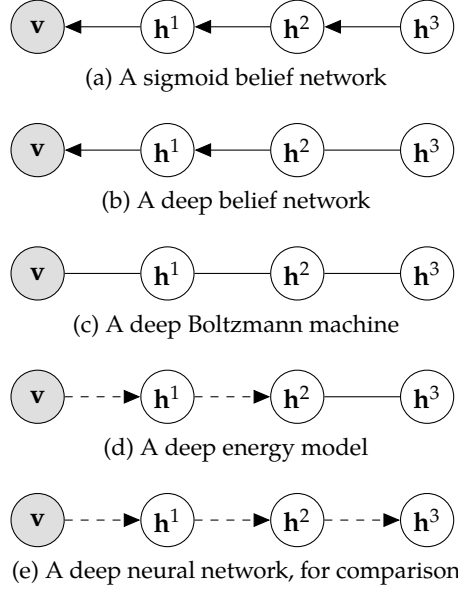


Figure 4.2: Deep graphical models. Dashed arrows represent deterministic relationships.

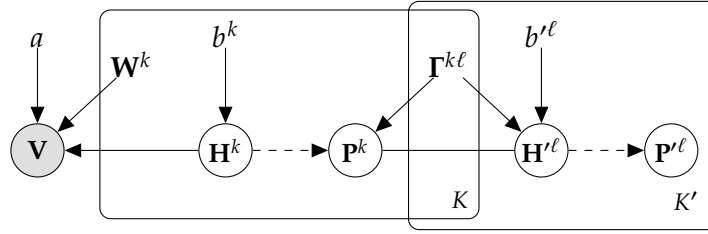


Figure 4.3: A two-layer CDBN

pixels). For inference, one can perform either Gibbs sampling or mean-field approximation.

4.2 Deep Boltzmann Machines

Contrary to DBNs, deep Boltzmann machines (DBMs) are fully undirected models (Fig. 4.2c). This means that a DBM incorporates both bottom-up and top-down relationships, allowing it to better represent uncertainty about ambiguous inputs (Salakhutdinov and Hinton, 2009a). In contrast to the efficient training of a DBN, training a DBM requires more sophisticated techniques which will be explored in more detail below.

In a DBM, the joint energy of the network is simply the sum of the energies of each layer. For example, the most common architecture consists of Bernoulli

hidden layers, in which case the joint energy is given by

$$E(\mathbf{v}, \mathbf{h}) = E_1(\mathbf{v}, \mathbf{h}^1) + \sum_{\ell=2}^L \left(-\mathbf{h}^{\ell-1\top} \mathbf{W}^\ell \mathbf{h}^\ell - \mathbf{b}^{\ell\top} \mathbf{h}^\ell \right), \quad (4.3)$$

where $\mathbf{h} = \{\mathbf{h}^\ell\}_{\ell=1}^L$, L is the number of hidden layers and E_1 is the energy function of the first layer, which may be any of the RBM types described above.

4.2.1 Variational inference

Once we begin to stack RBMs, the joint hidden unit posterior distribution no longer has a conveniently factorized closed form. To be able to perform inference on this model, we must therefore attempt to approximate the true posterior $\mathbb{P}(\mathbf{h}|\mathbf{v})$ by a fully factorized mean-field distribution $\mathbb{Q}(\mathbf{h}) = \prod_{\ell,j} \mathbb{Q}(h_j^\ell)$, where $\mathbb{Q}(h_j^\ell = 1) = \mu_j^\ell$ and thus $\mathbb{E}_{\mathbb{Q}}[\mathbf{h}] = \boldsymbol{\mu}$. The criterion used for finding the best approximating distribution is the Kullback-Leibler divergence:

$$\begin{aligned} KL[\mathbb{Q}(\mathbf{h})||\mathbb{P}(\mathbf{h}|\mathbf{v})] &= \sum_{\mathbf{h}} \mathbb{Q}(\mathbf{h}) \log \frac{\mathbb{Q}(\mathbf{h})}{\mathbb{P}(\mathbf{h}|\mathbf{v})} \\ &= \log \mathbb{P}(\mathbf{v}) + \sum_{\mathbf{h}} \mathbb{Q}(\mathbf{h}) \log \frac{\mathbb{Q}(\mathbf{h})}{\mathbb{P}(\mathbf{v}, \mathbf{h})} \\ &= \log \mathbb{P}(\mathbf{v}) - \mathcal{L}(\mathbb{Q}) \geq 0 \end{aligned} \quad (4.4)$$

By minimizing the variational free energy, $-\mathcal{L}(\mathbb{Q})$, we are thus minimizing the KL divergence and also maximizing a lower bound on the log-likelihood (Salakhutdinov and Hinton, 2009a), which is why $\mathcal{L}(\mathbb{Q})$ is often referred to as evidence lower bound (ELBO) in Statistics literature. This makes it a key ingredient for approximate ML training, described below. Setting $\nabla_{\boldsymbol{\mu}} \mathcal{L}(\mathbb{Q}) = 0$ yields the mean-field fixed-point equations (Appendix A.2):

$$\boldsymbol{\mu} = \sigma(-\nabla_{\boldsymbol{\mu}} E(\mathbf{v}, \boldsymbol{\mu})) \quad (4.5)$$

In our binary hidden layers example of Eq. (4.3), the fixed-point updates take the following form (Salakhutdinov and Larochelle, 2010):

$$\begin{cases} \mu^1 \leftarrow \sigma(-\nabla_{\mu^1} E_1(\mathbf{v}, \boldsymbol{\mu}^1) + \mathbf{W}^2 \boldsymbol{\mu}^2) \\ \mu^\ell \leftarrow \sigma(\mathbf{W}^{\ell\top} \boldsymbol{\mu}^{\ell-1} + \mathbf{W}^{\ell+1} \boldsymbol{\mu}^{\ell+1} + \mathbf{b}^\ell), \quad \ell = 2, \dots, L-1 \\ \mu^L \leftarrow \sigma(\mathbf{W}^L \boldsymbol{\mu}^{L-1} + \mathbf{b}^L) \end{cases}$$

We can initialize $\boldsymbol{\mu}$ to random values or zero, then update odd and even layers alternately with the above equations until convergence (Fig. 4.4). A better initialization can be obtained from a doubled bottom-up pass (single for the last layer), which makes the mean-field iterations converge much faster (Salakhutdinov and Hinton, 2009a). Furthermore, Salakhutdinov and Larochelle (2010) proposed training a separate recognition model to better initialize the mean-field parameters.

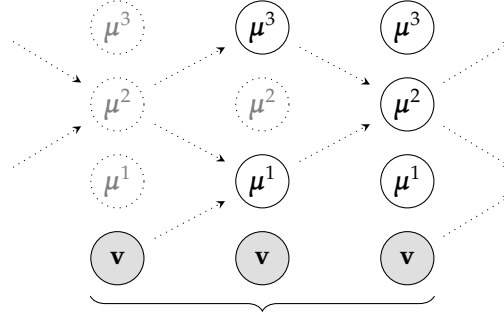


Figure 4.4: One iteration of the DBM mean-field updates

4.2.2 Approximate maximum likelihood training

As discussed previously, because the true posterior of a DBM is intractable, we must use the variational free energy as a surrogate for the log-likelihood. Its gradient w.r.t. the model parameters is (Bengio et al., 2012)

$$\nabla_{\theta} \mathcal{L}(\mathbb{Q}) = -\mathbb{E}_{\mathbb{Q}(\mathbf{h})}[\nabla_{\theta} E(\mathbf{v}, \mathbf{h})] + \mathbb{E}_{\mathbb{P}(\mathbf{v}, \mathbf{h})}[\nabla_{\theta} E(\mathbf{v}, \mathbf{h})]. \quad (4.6)$$

Note the resemblance to the log-likelihood gradient, Eq. (3.8), where we just replaced the posterior expectation with the mean-field expectation.¹

We can compute the first term with variational inference and, as before, $\mathbb{E}_{\mathbb{P}(\mathbf{v}, \mathbf{h})}[-\nabla_{\theta} E(\mathbf{v}, \mathbf{h})]$ can be approximated using contrastive divergence or its variants. Then, variational inference is performed again with the updated parameters, and the entire variational expectation-maximization is repeated until convergence. We will refer to a DBM trained with this approximate ML objective as ML-DBM.

Hinton and Salakhutdinov (2012) showed that this training algorithm can be greatly improved with DBN pre-training, instead of randomly initializing the parameters. Alternatively, Desjardins et al. (2012) proposed a regularized training scheme to allow joint training of the entire DBM, by attempting to limit the variance of the parameters during training.

Similarly to DBNs, DBMs themselves can also be fine-tuned for discriminative tasks by turning them into deterministic nets (Salakhutdinov and Hinton, 2009a).

4.2.3 Multi-prediction training

A very different approach to training DBMs, called multi-prediction (MP) training, is based on a variational approximation of the generalized pseudolikelihood, as opposed to the likelihood (Goodfellow et al., 2013). This method

¹Interestingly, because the true posterior of a single-layer RBM naturally factorizes, the result of its mean-field inference is exact: $\mathbb{Q}(h_j = 1) = \mu_j \equiv \mathbb{P}(h_j = 1 | \mathbf{v})$ (cf. Eq. (4.5)).

appears to circumvent the need for any pre-training, and thus the authors argue that it is much simpler to understand and implement.

The model is trained to predict any subset of variables given the remaining inputs, which makes it adept at answering arbitrary queries, even with missing inputs. However, in contrast to ML-DBMs, MP-DBMs are not optimized to generate realistic samples (i.e. as a generative model).

4.3 Deep Energy Models

RBM s are not constrained to be stacked only on top of other RBMs. Ngiam et al. (2011a) propose a class of models in which a single stochastic RBM layer is added to a deterministic feedforward neural network (Fig. 4.2d). In this way, rich representations may still be achieved from the deep neural network and, furthermore, the posterior distribution of the units in the last layer is factorial. In other words, it is a deep stochastic model with a tractable posterior.

In the proposed energy function,

$$E(\mathbf{v}, \mathbf{h}) = \frac{1}{2\sigma^2} \|\mathbf{v} - \mathbf{a}\|^2 - \mathbf{h}^\top \mathbf{W} \mathbf{g}_\theta(\mathbf{v}) - \mathbf{b}^\top \mathbf{h}, \quad (4.7)$$

$\mathbf{g}_\theta(\mathbf{v})$ represents the output of the feedforward neural network, and we can also recognize the Gaussian assumption on the inputs (cf. Section 3.1.2).

DEM training is carried out layer by layer, much like a DBN's. However, we optimize for the likelihood of the inputs of the bottom layer, and not for modelling the posteriors of the previous layer. This is possible because all previous layers are frozen and deterministic, absorbed into $\mathbf{g}_\theta$.

After this greedy pre-training step, all parameters can be unfrozen and trained jointly. As opposed to DBNs and DBMs, this is efficient because all intermediate layers are deterministic, so no sampling from them is required.

The authors propose as well a more general class of models, which can account for an arbitrary parametrization of the free energy function:

$$F(\mathbf{v}) = \frac{1}{2\sigma^2} \|\mathbf{v} - \mathbf{a}\|^2 + H(\mathbf{v}), \quad (4.8)$$

where $H(\mathbf{v}) = -\sum_i \log(1 + e^{[\mathbf{W} \mathbf{g}_\theta(\mathbf{v})]_i + a_i})$ for the neural network model. With this general formulation, other classes of models can also be represented, and the authors give PoT and cRBM as examples (see Section 3.1.3). The same learning procedure can also be applied to these general DEMs.

Chapter 5

Multimodal Learning

5.1 Models

The most straightforward approach to integrate different kinds of inputs in an RBM framework is to define a multimodal RBM, which is simply an RBM with visible units partitioned to represent separate input modalities:

$$E(\mathbf{v}_A, \mathbf{v}_B, \mathbf{h}) = -\mathbf{v}_A^\top \mathbf{W}_A \mathbf{h} - \mathbf{v}_B^\top \mathbf{W}_B \mathbf{h} - \mathbf{a}_A^\top \mathbf{v}_A - \mathbf{a}_B^\top \mathbf{v}_B - \mathbf{b}^\top \mathbf{h}$$

Here, again, the states of the visible units of each modality are conditionally independent of one another. This shallow model, however, cannot adequately capture cross-modality correlations (Ngiam et al., 2011b).

Deeper multimodal models can be built to address this issue. The easiest way of combining RBMs is to pre-train them separately for each modality, then add and train a multimodal RBM which joins the extracted features, forming a multimodal DBN (MDBN) (Ngiam et al., 2011b). In this specific work, the deep joint model was later ‘unrolled’ into a multimodal autoencoder, which was used for classification. Crucially, they used an augmented training dataset, in which data instances were replicated with either modality set to zero, and required the model to reconstruct both.

An alternative to this DBN-based autoencoder is to use a full multimodal DBM (MDBM). Srivastava and Salakhutdinov (2012) argue that a DBN would place all responsibility for the modality fusion on the joint layer – because the modality-specific pathways are trained independently –, whereas in a DBM it would be spread throughout all layers (Fig. 5.1). This implies that the states of hidden units on different pathways may influence each other, therefore every layer in the network plays a role in modelling the joint distribution of the multimodal inputs.

With a multimodal DBM, it is possible not only to extract joint representations (‘modality-free’ high-level features), but also to generate a missing modality. This can be done by either sampling from the appropriate distri-

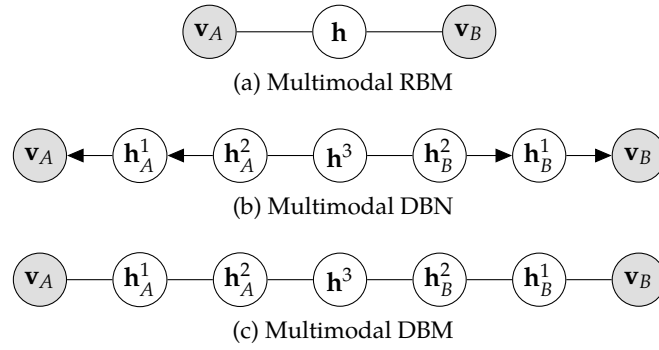


Figure 5.1: Multimodal architectures
(adapted from [Srivastava and Salakhutdinov, 2012](#))

bution with an alternating Gibbs sampler, as used for training, or by using variational inference ([Sohn et al., 2014](#)). Moreover, the learned representation can be used for answering queries with missing modalities ([Srivastava and Salakhutdinov, 2012](#)).

5.2 Learning

Training these multimodal deep models can be carried out in exactly the same fashion as unimodal ones. MDBNs can be trained greedily and for MDBMs we can apply the variational ML approach, as described in Section 4.2.2, potentially with MDBN pre-training ([Srivastava and Salakhutdinov, 2012](#)).

Recently, motivated by the challenge of dealing with missing data modalities, a new training objective for MRBMs has been proposed, based on variation of information (VI) ([Sohn et al., 2014](#)). This strategy explicitly seeks to improve the ability of the network to model the *conditional* distributions of one modality given the other: $\max_{\theta} \{\log \mathbb{P}(\mathbf{v}_A | \mathbf{v}_B) + \log \mathbb{P}(\mathbf{v}_B | \mathbf{v}_A)\}$.

This is in contrast to the traditional maximum likelihood training objective, which attempts to model the *joint* distribution. The authors propose two different training algorithms for the VI criterion: one based on the CD-PercLoss of conditional RBMs (Section 3.2.2) and another, more accurate, adapted from multi-prediction training (Section 4.2.3). In both cases, the authors combine the parameter updates calculated separately for $\mathbb{P}(\mathbf{v}_A | \mathbf{v}_B)$ and $\mathbb{P}(\mathbf{v}_B | \mathbf{v}_A)$.

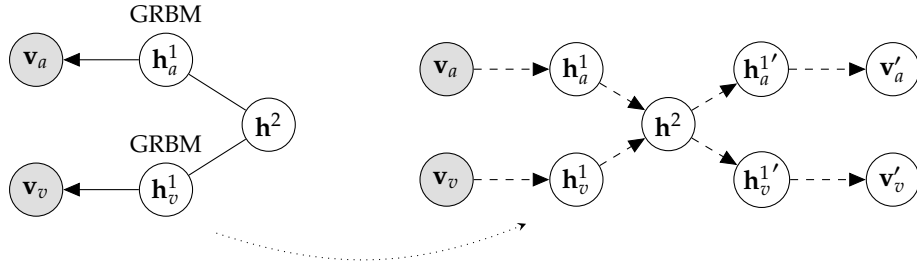
5.3 Application examples

For audio-visual speech classification, [Ngiam et al. \(2011b\)](#) combined separately pre-trained audio and video GRBMs with an added MRBM layer to form a MDBN, using spectrograms and extracted image features and respective time derivatives as inputs (Fig. 5.2a).

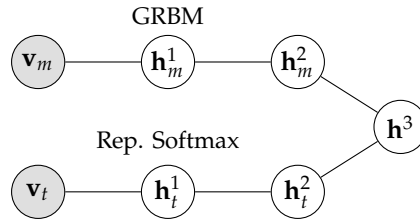
The MIR-Flickr data set ([Huiskes and Lew, 2008](#)) is commonly used for benchmarking models which relate image and text. It is composed of 1 million pictures, 25,000 of which are annotated with tags describing the image contents. For classification and retrieval tasks on this dataset, [Srivastava and Salakhutdinov \(2012, 2013\)](#) proposed using two-layer pathways – with a GRBM for extracted image features and a replicated softmax for the tags – and one joint MRBM (Fig. 5.2b). [Sohn et al. \(2014\)](#) went deeper, with three-layer pathways, and used a binary RBM instead of the replicated softmax (Fig. 5.2c).

In recent work on a medical imaging application, [Cai et al. \(2015\)](#) utilized a convolutional MDBN for detecting vertebrae locations from magnetic resonance (MR) and computed tomography (CT) scans. Both MR and CT pathways are composed of two convolutional layers, which are followed by two joint, fully-connected RBMs (Fig. 5.2d). The individual pathways (sequence of convolutional and fully-connected layers) are reminiscent of popular feedforward neural network architectures used in computer vision for object recognition (e.g. [Krizhevsky et al., 2012](#)).

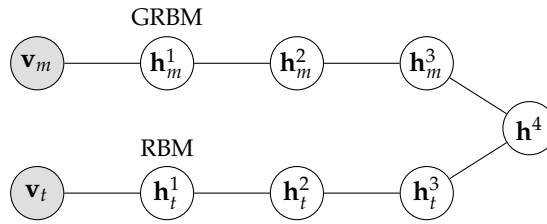
5.3. Application examples



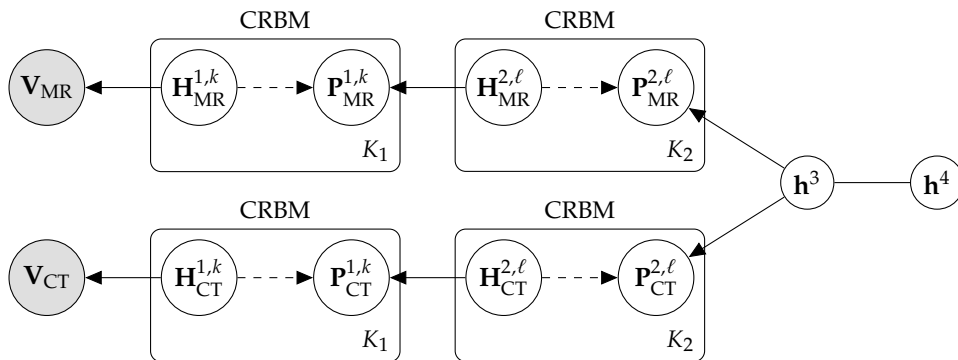
(a) The audio-video MDBN and unrolled autoencoder of [Ngiam et al. \(2011b\)](#)



(b) The image-text MDBM of [Srivastava and Salakhutdinov \(2012\)](#)



(c) The image-text MDBM of [Sohn et al. \(2014\)](#)



(d) The MR-CT convolutional MDBN of [Cai et al. \(2015\)](#)

Figure 5.2: Example multimodal architectures

Chapter 6

Discussion

In this survey, we have covered different classes of restricted Boltzmann machine models, tailored for various types of data and suitable for different tasks, such as representation and prediction. We have discussed how these models can be trained, along with considerations on enforcing some desirable properties, and also how we can arrange their connectivity to make them scalable to larger inputs and higher dimensions.

After dissecting these elementary building blocks, we analysed different strategies for constructing more representative models by moving to deep architectures. We broke down some fundamental properties of these deep models and reviewed practical considerations for training them.

Finally, we saw how all the presented models and techniques can be assembled for interesting applications involving multiple data modalities. Due to its flexibility, this class of multimodal probabilistic models opens up fertile grounds for investigation on real-world applications.

Table [6.1](#) summarizes the main categories of models covered in this text, putting into perspective the different training objectives and corresponding learning algorithms.

Table 6.1: Summary of different types of RBM-based models

Type of model	Objective ^a	Training strategy ^b	Main references
Generative RBMs	$\log \mathbb{P}(\mathbf{v})$	(Section 3.1.4)	(Table 3.1)
Discriminative RBM	$\log \mathbb{P}(y \mathbf{v})$	Exact	Larochelle and Bengio (2008)
Conditional RBM	$\log \mathbb{P}(\mathbf{v} \mathbf{u})$	CD-PercLoss	Mnih et al. (2012)
Deep belief network	$\log \mathbb{P}(\mathbf{h}^{L-1})$	Greedy layer-wise CD	Hinton et al. (2006a)
Deep energy model	$\log \mathbb{P}(\mathbf{v})$	Greedy layer-wise CB	Ngiam et al. (2011a)
Deep Boltzmann machine	$\log \mathbb{P}(\mathbf{v})$	Mean-field + CD	Salakhutdinov and Hinton (2009a)
	$\sum_s \log \mathbb{P}(\mathbf{v}_s \mathbf{v}_{\setminus s})$	Multi-prediction	Goodfellow et al. (2013)
Multimodal DBN	$\log \mathbb{P}(\mathbf{h}_A^{L-1}, \mathbf{h}_B^{L-1})$	Greedy layer-wise CD	Ngiam et al. (2011b)
Multimodal DBM	$\log \mathbb{P}(\mathbf{v}_A, \mathbf{v}_B)$	Mean-field + CD	Srivastava and Salakhutdinov (2012)
	$\log \mathbb{P}(\mathbf{v}_A \mathbf{v}_B) + \log \mathbb{P}(\mathbf{v}_B \mathbf{v}_A)$	2× CD-PercLoss or 2× Multi-prediction	Sohn et al. (2014)

^a y = class label, $\mathbf{u}$ = context variables, $\mathbf{v}_{A/B}$ = modality A/B^b CD = contrastive divergence (or PCD, FPCD), CB = contrastive backpropagation

6.1 Case study

To illustrate how these elements fit together, let us attempt to engineer a solution for our Alzheimer’s disease detection example, which is recapitulated in the following table:

Table 6.2: AD detection example

	Modality	Type	Train	Test
MRI	3D volume	Continuous	✓	✓
PET	3D volume	Continuous	✓	✓
Report	Text (unigram)	Multinomial	✓	✗
Diagnosis	Output	Binary	✓	✗

We’ve already seen in Section 3.1.7 that, for typical medical images, the only feasible options for connectivity are convolution and tiled convolution. To allow our model to learn richer image features than with convolution alone, we may settle for a tiled-convolutional pattern. Additionally, to allow some degree of translation/deformation invariance and greatly reduce the computational burden of the fusion layer, we may stack two convolutional layers with probabilistic max-pooling.

In this particular application, we speculate that we would have little added benefit from capturing the covariance between voxels. Furthermore, if we have a receptive field with width R (covering R^3 voxels), tile size T and K feature maps, modelling the covariance would involve $O(KT^3R^6)$ additional parameters, which might represent a prohibitive computational cost. With this in mind, we may opt for using the simple Gaussian RBM with diagonal covariance.

For the text processing pathway, it is a clear choice of the replicated softmax, upon which we may add an additional hidden layer for better representation.

We then add a fully-connected multimodal RBM layer to integrate all three pathways. Finally, we add a discriminative RBM on top of the entire model to perform the classification. See Fig. 6.1 for a global view of the model.

We can pre-train each pathway independently as DBNs (i.e. greedily train each successive layer with contrastive divergence), then add and train the multimodal fusion layer, again with CD. In addition, performing dropout might be helpful to reduce correlation between the learned features, promote sparsity and improve generalization to unseen patients. Finally, we may append the discriminative layer and jointly fine-tune the entire model with discriminative learning:

$$\nabla_{\theta} \log \mathbb{P}(y|\mathbf{v}) = -\mathbb{E}_{\mathbb{P}(\mathbf{h}|y,\mathbf{v})}[\nabla_{\theta} E(y, \mathbf{v}, \mathbf{h})] + \mathbb{E}_{\mathbb{P}(y,\mathbf{h}|\mathbf{v})}[\nabla_{\theta} E(y, \mathbf{v}, \mathbf{h})],$$

where y is the class label (diagnosis), $\mathbf{v}$ is the ensemble of inputs (MRI, PET and report) and $\mathbf{h}$ contains all hidden and pooling units.

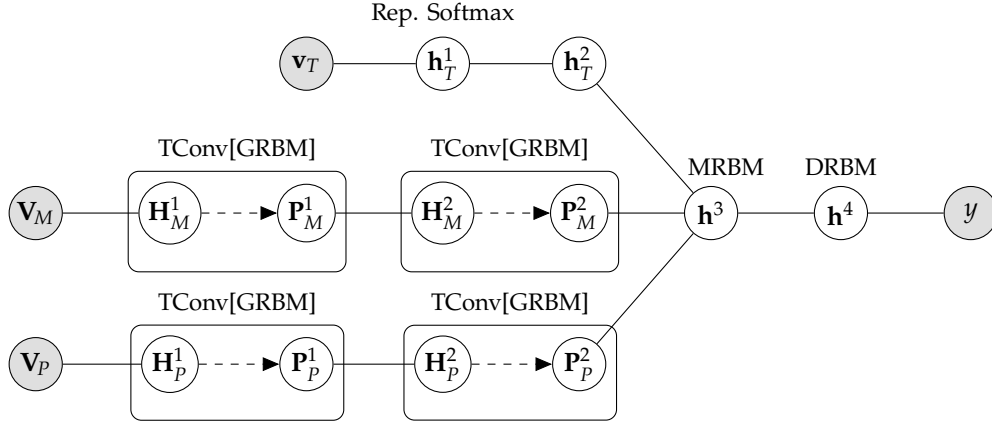


Figure 6.1: Graphical representation of the AD detection example

Because the training objective is discriminative, the whole learning procedure would be conditioned on the inputs, so no sampling is required. All we need are expectations on the class label and hidden units, which can be approximated with variational inference.

Critically, because in the end the model is trained jointly as an *undirected* DBM, we expect the textual pathway to have contributed to embedding the knowledge extracted from the reports across the entire network, such that it can still be leveraged when the report is absent at test time.

Alternatively, we could have trained the fusion MRBM in a conditional fashion, calibrating it to predict the clinical report based on the brain scans. This potentially makes the intermediate model (i.e. before appending the discriminative layer) more robust to the missing modality, and has the added benefit of producing a conditional generative model for the report.

6.2 Future directions

Deep learning is an extremely active research area in constant development. There was initially much interest in these probabilistic models (e.g. DBNs) because of their potential to improve the performance of deep neural networks by pre-training. Their potential as standalone models was gradually uncovered and they have taken off on their own.

There is currently much work on improving the training procedures themselves, such as refinements of SGD, adaptive learning rates, automated hyperparameter adjustment and new regularization schemes (Schmidhuber, 2014; Bengio et al., 2012), along with optimizing implementations for better hardware (LeCun et al., 2015).

Another very exciting line of research regards extending these probabilistic

models to incorporate more expressive power. For example, [Srivastava and Salakhutdinov \(2013\)](#) proposed adding priors on the topmost layer of a deep model to allow knowledge sharing in a hierarchy of classes. In a related effort, [Salakhutdinov et al. \(2013\)](#) presented a new approach combining DBMs with non-parametric hierarchical Bayesian models, enabling the models to generalize to previously unseen classes (i.e. one-shot learning).

References

- Ackley, D., Hinton, G. E., and Sejnowski, T. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, 9(1):147–169. [6](#)
- Bengio, Y. (2009). Learning Deep Architectures for AI. *Foundations and Trends(R) in Machine Learning*, 2(1):1–127. [7](#), [13](#)
- Bengio, Y., Courville, A., and Vincent, P. (2012). Representation Learning: A Review and New Perspectives. [3](#), [10](#), [12](#), [14](#), [20](#), [24](#), [33](#)
- Cai, Y., Osman, S., Sharma, M., Landis, M., and Li, S. (2015). Multi-Modality Vertebra Recognition in Arbitrary Views Using 3D Deformable Hierarchical Model. *IEEE Transactions on Medical Imaging*, 34(8):1676–93. [4](#), [5](#), [28](#), [29](#)
- Cao, T., Jojic, V., Modla, S., Powell, D., Czymmek, K., and Niethammer, M. (2013). Robust Multimodal Dictionary Learning. In Mori, K., Sakuma, I., Sato, Y., Barillot, C., and Navab, N., editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2013*, pages 259–266. [4](#)
- Carreira-Perpiñán, M. Á. and Hinton, G. E. (2005). On Contrastive Divergence Learning. In Cowell, R. and Ghahramani, Z., editors, *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics (AISTATS 2005)*, volume 10, pages 33–40. [12](#)
- Caruana, R. (1997). Multitask Learning. *Machine Learning*, 28(1):41–75. [3](#)
- Cho, K., Ilin, A., and Raiko, T. (2011). Improved Learning Algorithms for Restricted Boltzmann Machines. In *Artificial Neural Networks and Machine Learning - ICANN 2011*, pages 10–17. [9](#), [13](#)
- Colby, J. B., Rudie, J. D., Brown, J. A., Douglas, P. K., Cohen, M. S., and Shehzad, Z. (2012). Insights into multimodal imaging classification of ADHD. *Frontiers in Systems Neuroscience*, 6(AUG 2012):59. [4](#)
- Courville, A., Bergstra, J., and Bengio, Y. (2011a). A Spike and Slab Restricted Boltzmann Machine. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011)*, 15:233–241. [10](#), [11](#)
- Courville, A., Bergstra, J., and Bengio, Y. (2011b). Unsupervised Models of Images by Spike-and-Slab RBMs. *Proceedings of the 28th International Conference on Machine Learning (ICML 11)*, 10:8. [10](#)

- Courville, A., Desjardins, G., Bergstra, J., and Bengio, Y. (2014). The Spike-and-Slab RBM and Extensions to Discrete and Sparse Data Distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(9):1874–1887. [11](#)
- Dahl, G., Ranzato, M., Mohamed, A.-r., and Hinton, G. E. (2010). Phone Recognition with the Mean-Covariance Restricted Boltzmann Machine. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pages 469–477. [10](#)
- De Martino, F., Valente, G., de Borst, A. W., Esposito, F., Roebroek, A., Goebel, R., and Formisano, E. (2010). Multimodal imaging: an evaluation of univariate and multivariate methods for simultaneous EEG/fMRI. *Magnetic Resonance Imaging*, 28(8):1104–1112. [4](#)
- Deng, L. (2012). Three Classes of Deep Learning Architectures and Their Applications: A Tutorial Survey. *APSIPA Transactions on Signal and Information Processing*. [20](#)
- Desjardins, G. and Bengio, Y. (2008). Empirical Evaluation of Convolutional RBMs for Vision. Technical report. [15](#), [16](#)
- Desjardins, G., Bengio, Y., and Courville, A. C. (2011). On Tracking The Partition Function. In *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, pages 2501–2509. [14](#)
- Desjardins, G., Courville, A., and Bengio, Y. (2012). On Training Deep Boltzmann Machines. [24](#)
- Duane, S., Kennedy, A., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222. [13](#)
- Fischer, A. and Igel, C. (2014). Training restricted Boltzmann machines: An introduction. *Pattern Recognition*, 47(1):25–39. [12](#)
- Friston, K. J. (2009). Modalities, Modes, and Models in Functional Neuroimaging. *Science*, 326(5951):399–403. [4](#)
- Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2014). Learning Multimodal Latent Attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):303–316. [3](#), [4](#)
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202. [15](#)
- Gan, Z., Henao, R., Carlson, D., and Carin, L. (2015). Learning Deep Sigmoid Belief Networks with Data Augmentation. In Lebanon, G. and Vishwanathan, S. V. N., editors, *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS 2015)*, pages 268–276. [20](#)

- Geremia, E., Zikic, D., Clatz, O., Menze, B. H., Glocker, B., Konukoglu, E., Shotton, J., Thomas, O. M., Price, S. J., Das, T., Jena, R., Ayache, N., and Criminisi, A. (2013). Classification Forests for Semantic Segmentation of Brain Lesions in Multi-channel MRI. In Criminisi, A. and Shotton, J., editors, *Decision Forests for Computer Vision and Medical Image Analysis*, Advances in Computer Vision and Pattern Recognition, pages 245–260. Springer London. [3](#)
- Gomez-Chova, L., Tuia, D., Moser, G., and Camps-Valls, G. (2015). Multimodal Classification of Remote Sensing Images: A Review and Future Directions. *Proceedings of the IEEE*, 103(9):1560–1584. [3](#)
- Goodfellow, I., Mirza, M., Courville, A., and Bengio, Y. (2013). Multi-Prediction Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 548–556. [24](#), [31](#)
- Gregor, K. and LeCun, Y. (2010). Emergence of Complex-Like Cells in a Temporal Product Network with Local Receptive Fields. [17](#)
- Hill, D. L. G., Batchelor, P. G., Holden, M., and Hawkes, D. J. (2001). Medical image registration. *Physics in Medicine and Biology*, 46(3):R1–R45. [3](#)
- Hill, D. L. G., Studholme, C., and Hawkes, D. J. (1994). Voxel Similarity Measures for Automated Image Registration. *Visualization in Biomedical Computing*, 2359. [3](#)
- Hinrichs, C., Singh, V., Xu, G., and Johnson, S. (2009). MKL for robust multimodality AD classification. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2009*, pages 786–794. Springer. [4](#)
- Hinton, G. E. (2002). Training Products of Experts by Minimizing Contrastive Divergence. *Neural Computation*, 14(8):1771–1800. [5](#), [12](#)
- Hinton, G. E. (2010). A Practical Guide to Training Restricted Boltzmann Machines. [9](#), [11](#), [12](#)
- Hinton, G. E., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and Kingsbury, B. (2012a). Deep Neural Networks for Acoustic Modeling in Speech Recognition. *IEEE Signal Processing Magazine*, 29(6):82–97. [9](#), [20](#)
- Hinton, G. E., Osindero, S., and Teh, Y. W. (2006a). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554. [11](#), [20](#), [31](#)
- Hinton, G. E., Osindero, S., Welling, M., and Teh, Y.-W. (2006b). Unsupervised Discovery of Nonlinear Structure Using Contrastive Backpropagation. *Cognitive Science*, 30(4):725–31. [13](#)
- Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507. [20](#)

- Hinton, G. E. and Salakhutdinov, R. R. (2012). A Better Way to Pretrain Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 2447–2455. [14](#), [24](#)
- Hinton, G. E. and Sejnowski, T. (1986). Learning and relearning in Boltzmann machines. *Parallel Distributed Processing*, 1:282–317. [6](#)
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2012b). Improving neural networks by preventing co-adaptation of feature detectors. [13](#)
- Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79(8):2554–2558. [6](#)
- Hor, S. and Moradi, M. (2015). Scandent Tree: A Random Forest Learning Method for Incomplete Multimodal Datasets. In Nassir, N., Horneegger, J., Wells, W. M., and Frangi, A. F., editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, volume 9349, pages 694–701, Munich, Germany. Springer International Publishing. [4](#)
- Huiskes, M. J. and Lew, M. S. (2008). The MIR Flickr Retrieval Evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval (MIR '08)*, pages 39–43, New York, New York, USA. ACM Press. [28](#)
- Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6:695–708. [13](#)
- Hyvärinen, A. (2007). Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on Neural Networks*, 18(5):1529 – 1531. [13](#)
- Irie, G., Liu, D., Li, Z., and Chang, S.-F. (2013). A Bayesian Approach to Multimodal Visual Dictionary Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, pages 329–336. [4](#)
- Kadoury, S., Labelle, H., and Paragios, N. (2011). Automatic inference of articulated spine models in CT images using high-order Markov Random Fields. *Medical Image Analysis*, 15(4):426–437. [4](#)
- Kahou, S. E., Bouthillier, X., Lamblin, P., Gulcehre, C., Michalski, V., Konda, K., Jean, S., Froumenty, P., Dauphin, Y., Boulanger-Lewandowski, N., Ferrari, R. C., Mirza, M., Warde-Farley, D., Courville, A., Vincent, P., Memisevic, R., Pal, C., and Bengio, Y. (2015). EmoNets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, pages 1–13. [3](#)
- Kingma, D. P. and LeCun, Y. (2010). Regularized estimation of image statistics by Score Matching. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pages 1126–1134. [13](#), [16](#)

- Kiros, R., Salakhutdinov, R. R., and Zemel, R. (2014). Multimodal Neural Language Models. In *Proceedings of the 31st International Conference on Machine Learning (ICML 14)*, pages 595–603. [3](#)
- Kivinen, J. and Williams, C. (2012). Multiple Texture Boltzmann Machines. *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS 2012)*, 22:638–646. [10](#), [16](#), [18](#)
- Krizhevsky, A. (2009). *Learning Multiple Layers of Features from Tiny Images*. Master’s thesis, University of Toronto. [9](#)
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pages 1–9. [15](#), [21](#), [28](#)
- Lampert, C. H. and Kroemer, O. (2010). Weakly-Paired Maximum Covariance Analysis for Multimodal Dimensionality Reduction and Transfer Learning. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*, pages 566–579. Springer Berlin Heidelberg. [4](#)
- Larochelle, H. and Bengio, Y. (2008). Classification using discriminative restricted Boltzmann machines. In *Proceedings of the 25th International Conference on Machine Learning (ICML 08)*, pages 536–543, New York, New York, USA. ACM Press. [18](#), [31](#)
- Le, Q. V., Ngiam, J., Chen, Z., Chia, D., Koh, P. W., and Ng, A. Y. (2010). Tiled convolutional neural networks. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pages 1279–1287. [17](#)
- LeCun, Y., Bengio, Y., and Hinton, G. E. (2015). Deep learning. *Nature*, 521(7553):436–444. [33](#)
- LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., and Huang, F. J. (2006). A Tutorial on Energy-Based Learning. In Bakir, G., Hofmann, T., Schölkopf, B., Smola, A. J., Taskar, B., and Vishwanathan, S. V. N., editors, *Predicting Structured Data*, chapter 10, pages 191–246. MIT Press. [5](#), [19](#)
- Lee, H., Ekanadham, C., and Ng, A. Y. (2008). Sparse deep belief net model for visual area V2. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pages 873–880. [9](#), [13](#), [21](#)
- Lee, H., Grosse, R., Ranganath, R., and Ng, A. Y. (2009a). Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations. In *Proceedings of the 26th International Conference on Machine Learning (ICML 09)*, pages 1–8, New York, New York, USA. ACM Press. [9](#), [15](#), [16](#), [21](#)

- Lee, H., Pham, P., Largman, Y., and Ng, A. Y. (2009b). Unsupervised Feature Learning for Audio Classification using Convolutional Deep Belief Networks. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1096–1104. 15
- Little, W. A. (1974). The Existence of Persistent States in the Brain. *Mathematical Biosciences*, 19:101–120. 6
- Luo, H., Carrier, P. L., Courville, A., and Bengio, Y. (2012). Texture Modeling with Convolutional Spike-and-Slab RBMs and Deep Extensions. 18
- Malik, J. and Perona, P. (1990). Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7(5):923. 15
- McFee, B. and Lanckriet, G. (2011). Learning Multi-modal Similarity. *Journal of Machine Learning Research*, 12:491–523. 4
- Mnih, V., Larochelle, H., and Hinton, G. E. (2012). Conditional Restricted Boltzmann Machines for Structured Output Prediction. 7, 19, 31
- Monaci, G., Jost, P., Vandergheynst, P., Mailhe, B., Lesage, S., and Gribonval, R. (2007). Learning Multimodal Dictionaries. *IEEE Transactions on Image Processing*, 16(9):2272–2283. 4
- Nair, V. and Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. *Proceedings of the 27th International Conference on Machine Learning (ICML 10)*, (3):807–814. 14, 17
- Neal, R. M. (1992). Connectionist learning of belief networks. *Artificial Intelligence*, 56(1):71–113. 20
- Neal, R. M. (1993). Probabilistic Inference Using Markov Chain Monte Carlo Methods. Technical report, University of Toronto. 13
- Neal, R. M. (2001). Annealed Importance Sampling. *Statistics and Computing*, 11(2):125–139. 14
- Ngiam, J., Chen, Z., Koh, P. W., and Ng, A. Y. (2011a). Learning Deep Energy Models. In *Proceedings of the 28th International Conference on Machine Learning (ICML 11)*, pages 1105–1112. 14, 25, 31
- Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., and Ng, A. Y. (2011b). Multimodal Deep Learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML 11)*, pages 689–696. 3, 4, 5, 26, 27, 29, 31
- Norouzi, M., Ranjbar, M., and Mori, G. (2009). Stacks of convolutional Restricted Boltzmann Machines for shift-invariant feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, pages 2735–2742. IEEE. 16
- Pan, S. J. and Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359. 3

- Ranzato, M. and Hinton, G. E. (2010). Modeling pixel means and covariances using factorized third-order Boltzmann machines. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2010)*, pages 2551–2558. [10](#), [11](#)
- Ranzato, M., Krizhevsky, A., and Hinton, G. E. (2010a). Factored 3-Way Restricted Boltzmann Machines For Modeling Natural Images. *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, 13:621–628. [10](#), [11](#)
- Ranzato, M., Mnih, V., and Hinton, G. E. (2010b). Generating more realistic images using gated MRF’s. In *Advances in Neural Information Processing Systems 23 (NIPS 2010)*, pages 2002–2010. [10](#), [11](#), [17](#), [18](#)
- Roth, S. and Black, M. J. (2009). Fields of Experts. *International Journal of Computer Vision*, 82(2):205–229. [16](#)
- Salakhutdinov, R. R. (2009). Learning in Markov Random Fields using Tempered Transitions. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1598–1606. [13](#)
- Salakhutdinov, R. R. and Hinton, G. E. (2009a). Deep Boltzmann Machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS 2009)*, volume 5, pages 448–455. [12](#), [14](#), [22](#), [23](#), [24](#), [31](#)
- Salakhutdinov, R. R. and Hinton, G. E. (2009b). Replicated Softmax: an Undirected Topic Model. In *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, pages 1607–1614. [8](#), [11](#), [14](#)
- Salakhutdinov, R. R. and Larochelle, H. (2010). Efficient Learning of Deep Boltzmann Machines. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS 2010)*, volume 9, pages 693–700. [14](#), [23](#)
- Salakhutdinov, R. R. and Murray, I. (2008). On the quantitative analysis of deep belief networks. In *Proceedings of the 25th International Conference on Machine Learning (ICML 08)*, pages 872–879, New York, New York, USA. ACM Press. [14](#)
- Salakhutdinov, R. R., Tenenbaum, J. B., and Torralba, A. (2013). Learning with Hierarchical-Deep Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1958–71. [34](#)
- Schlegl, T., Waldstein, S. M., Vogl, W.-D., Schmidt-Erfurth, U., and Langs, G. (2015). Predicting Semantic Descriptions from Medical Images with Convolutional Neural Networks. In *Proceedings of the 24th International Conference on Information Processing in Medical Imaging (IPMI 2015)*, pages 437–448. [4](#)
- Schmidhuber, J. (2014). Deep Learning in Neural Networks: An Overview. *Neural Networks*, 61:85–117. [33](#)

- Smolensky, P. (1986). Information processing in dynamical systems: Foundations of harmony theory. *Parallel Distributed Processing*, 1:194–281. [6](#)
- Sohn, K., Shang, W., and Lee, H. (2014). Improved Multimodal Deep Learning with Variation of Information. In *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pages 2141–2149. [4](#), [5](#), [27](#), [28](#), [29](#), [31](#)
- Sonnenburg, S., Rätsch, G., Schäfer, C., and Schölkopf, B. (2006). Large Scale Multiple Kernel Learning. *Journal of Machine Learning Research*, 7:1531–1565. [4](#)
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958. [14](#)
- Srivastava, N. and Salakhutdinov, R. R. (2012). Multimodal Learning with Deep Boltzmann Machines. In *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, volume 15, pages 2231–2239. [3](#), [4](#), [5](#), [26](#), [27](#), [28](#), [29](#), [31](#)
- Srivastava, N. and Salakhutdinov, R. R. (2013). Discriminative Transfer Learning with Tree-based Priors. In *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, pages 2094–2102. [4](#), [28](#), [34](#)
- Srivastava, N., Salakhutdinov, R. R., and Hinton, G. E. (2013). Modeling Documents with Deep Boltzmann Machines. [8](#), [14](#)
- Sui, J., Adali, T., Yu, Q., Chen, J., and Calhoun, V. D. (2012). A review of multivariate methods for multimodal fusion of brain imaging data. *Journal of Neuroscience Methods*, 204(1):68–81. [4](#)
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., and Rabinovich, A. (2014). Going Deeper with Convolutions. [21](#)
- Taylor, G. W., Fergus, R., LeCun, Y., and Bregler, C. (2010). Convolutional Learning of Spatio-temporal Features. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Proceedings of the 11th European Conference on Computer Vision (ECCV 2010)*, Lecture Notes in Computer Science, pages 140–153, Berlin, Heidelberg. Springer Berlin Heidelberg. [16](#)
- Taylor, G. W. and Hinton, G. E. (2009). Factored conditional restricted Boltzmann Machines for modeling motion style. In *Proceedings of the 26th International Conference on Machine Learning (ICML 09)*, pages 1025–1032, New York, New York, USA. ACM Press. [19](#)
- Taylor, G. W., Hinton, G. E., and Roweis, S. T. (2007). Modeling Human Motion Using Binary Latent Variables. In *Advances in Neural Information Processing Systems 19 (NIPS 2006)*, pages 1345–1352. [19](#)

- Teh, Y.-W. and Hinton, G. E. (2001). Rate-coded Restricted Boltzmann Machines for Face Recognition. In *Advances in Neural Information Processing Systems 13 (NIPS 2001)*, pages 908–914. [14](#)
- Teh, Y. W., Welling, M., Osindero, S., and Hinton, G. E. (2003). Energy-based models for sparse overcomplete representations. *Journal of Machine Learning Research*, 4:1235–1260. [11](#), [12](#)
- Tieleman, T. (2008). Training Restricted Boltzmann Machines using Approximations to the Likelihood Gradient. In *Proceedings of the 25th International Conference on Machine Learning (ICML 08)*, pages 1064–1071, New York, New York, USA. ACM Press. [12](#)
- Tieleman, T. and Hinton, G. E. (2009). Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th International Conference on Machine Learning (ICML 09)*, pages 1–8, New York, New York, USA. ACM Press. [12](#)
- Toga, A. W., Thompson, P. M., Mori, S., Amunts, K., and Zilles, K. (2006). Towards multimodal atlases of the human brain. *Nature Reviews Neuroscience*, 7(12):952–966. [4](#)
- Welling, M., Hinton, G. E., and Osindero, S. (2002). Learning Sparse Topographic Representation with Products of Student-t-Distributions. In *Advances in Neural Information Processing Systems 15 (NIPS 2002)*, pages 1359–1366. [10](#), [11](#)
- Welling, M., Rosen-zvi, M., and Hinton, G. E. (2004). Exponential Family Harmoniums with an Application to Information Retrieval. In *Advances in Neural Information Processing Systems 17 (NIPS 2004)*, pages 1481–1488. [9](#)

Appendix A

Derivations

A.1 RBM log-likelihood gradient

$$\log \mathbb{P}(\mathbf{v}) = \log \left(\frac{1}{Z} \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} \right) = \log \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} - \log Z$$

For clarity, let us use $\mathbf{v}^*$ to explicitly denote the argument of the likelihood.

$$\begin{aligned} \nabla_{\theta} \log \mathbb{P}(\mathbf{v}^*) &= \frac{\sum_{\mathbf{h}} e^{-E(\mathbf{v}^*, \mathbf{h})} (-\nabla_{\theta} E(\mathbf{v}^*, \mathbf{h}))}{\sum_{\mathbf{h}} e^{-E(\mathbf{v}^*, \mathbf{h})}} - \frac{1}{Z} \sum_{\mathbf{v}, \mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})} (-\nabla_{\theta} E(\mathbf{v}, \mathbf{h})) \\ &= \sum_{\mathbf{h}} -\nabla_{\theta} E(\mathbf{v}^*, \mathbf{h}) \frac{e^{-E(\mathbf{v}^*, \mathbf{h})} / Z}{\sum_{\mathbf{h}'} e^{-E(\mathbf{v}^*, \mathbf{h}')} / Z} - \sum_{\mathbf{v}, \mathbf{h}} -\nabla_{\theta} E(\mathbf{v}, \mathbf{h}) \frac{e^{-E(\mathbf{v}, \mathbf{h})}}{Z} \\ &= \sum_{\mathbf{h}} -\nabla_{\theta} E(\mathbf{v}^*, \mathbf{h}) \frac{\mathbb{P}(\mathbf{v}^*, \mathbf{h})}{\mathbb{P}(\mathbf{v}^*)} - \sum_{\mathbf{v}, \mathbf{h}} -\nabla_{\theta} E(\mathbf{v}, \mathbf{h}) \mathbb{P}(\mathbf{v}, \mathbf{h}) \\ &= \sum_{\mathbf{h}} -\nabla_{\theta} E(\mathbf{v}^*, \mathbf{h}) \mathbb{P}(\mathbf{h} | \mathbf{v}^*) - \sum_{\mathbf{v}, \mathbf{h}} -\nabla_{\theta} E(\mathbf{v}, \mathbf{h}) \mathbb{P}(\mathbf{v}, \mathbf{h}) \\ &= \mathbb{E}_{\mathbb{P}(\mathbf{h} | \mathbf{v}^*)} [-\nabla_{\theta} E(\mathbf{v}^*, \mathbf{h})] - \mathbb{E}_{\mathbb{P}(\mathbf{v}, \mathbf{h})} [-\nabla_{\theta} E(\mathbf{v}, \mathbf{h})] \quad \text{Eq. (3.8) } \blacksquare \end{aligned}$$

Alternatively, in terms of the free energy:

$$\begin{aligned} \log \mathbb{P}(\mathbf{v}^*) &= -F(\mathbf{v}^*) - \log Z \\ \nabla_{\theta} \log \mathbb{P}(\mathbf{v}^*) &= -\nabla_{\theta} F(\mathbf{v}^*) - \frac{1}{Z} \sum_{\mathbf{v}} e^{-F(\mathbf{v})} (-\nabla_{\theta} F(\mathbf{v})) \\ &= -\nabla_{\theta} F(\mathbf{v}^*) - \sum_{\mathbf{v}} -\nabla_{\theta} F(\mathbf{v}) \frac{e^{-F(\mathbf{v})}}{Z} \\ &= -\nabla_{\theta} F(\mathbf{v}^*) - \sum_{\mathbf{v}} -\nabla_{\theta} F(\mathbf{v}) \mathbb{P}(\mathbf{v}) \\ &= -\nabla_{\theta} F(\mathbf{v}^*) - \mathbb{E}_{\mathbb{P}(\mathbf{v})} [-\nabla_{\theta} F(\mathbf{v})] \quad \blacksquare \end{aligned}$$

A.2 DBM mean-field fixed-point equations

The variational free energy can be rewritten as

$$\begin{aligned}\mathcal{L}(\mathbb{Q}) &= \mathbb{E}_{\mathbb{Q}(\mathbf{h})}[\log \mathbb{P}(\mathbf{v}, \mathbf{h}) - \log \mathbb{Q}(\mathbf{h})] \\ &= -E(\mathbf{v}, \boldsymbol{\mu}) - \log Z + \mathcal{H}(\mathbb{Q}),\end{aligned}$$

where

$$\mathcal{H}(\mathbb{Q}) = - \sum_{\ell, j} [\mu_j^\ell \log \mu_j^\ell + (1 - \mu_j^\ell) \log(1 - \mu_j^\ell)]$$

is the entropy of $\mathbb{Q}$, and its partial derivatives are given by

$$\frac{\partial \mathcal{H}(\mathbb{Q})}{\partial \mu} = -(\log \mu + 1 - \log(1 - \mu) - 1) = -\log \frac{\mu}{1 - \mu}.$$

Taking the derivative of $\mathcal{L}$ w.r.t. one of the variational parameters gives

$$\frac{\partial \mathcal{L}(\mathbb{Q})}{\partial \mu} = -\frac{\partial E(\mathbf{v}, \boldsymbol{\mu})}{\partial \mu} + \frac{\partial \mathcal{H}(\mathbb{Q})}{\partial \mu} = 0 \quad \therefore \quad \log \frac{\mu}{1 - \mu} = -\frac{\partial E(\mathbf{v}, \boldsymbol{\mu})}{\partial \mu}$$

Finally, solving for μ yields

$$\mu = \frac{\exp(-\frac{\partial}{\partial \mu} E(\mathbf{v}, \boldsymbol{\mu}))}{1 + \exp(-\frac{\partial}{\partial \mu} E(\mathbf{v}, \boldsymbol{\mu}))} = \sigma\left(-\frac{\partial E(\mathbf{v}, \boldsymbol{\mu})}{\partial \mu}\right) \quad \text{Eq. (4.5) } \blacksquare$$