Inference on Boltzmann Machines Beyond Layer Architectures

—Probabilistic Continuous Relaxation
Hamiltonian Monte Carlo
Geometry

Yichuan Zhang

15th March
CamAIML 2018

A long time ago in a paper, before the rise of Deep Learning….

A Short History of Boltzmann Machines (1985-2009)

General Boltzmann Machines(1984-86)

COGNITIVE SCIENCE 9, 147–169 (1985)

A Learning Algorithm for Boltzmann Machines*

DAVID H. ACKLEY
GEOFFREY E. HINTON
Computer Science Department
Carnegie-Mellon University
TERRENCE J. SEJNOWSKI
Biophysics Department
The Johns Hopkins University

The computational power of massively parallel networks of simple processing elements resides in the communication bandwidth provided by the hardware connections between elements. These connections can allow a significant fraction of the knowledge of the system to be applied to an instance of a problem in a very short time. One kind of computation for which massively parallel networks appear to be well suited is large constraint satisfaction searches, but to use the connections efficiently two conditions must be met: First, a search technique that is suitable for parallel networks must be found. Second, there must be some way of choosing internal representations which allow the preexisting hardware connections to be used efficiently for encoding the constraints in the domain being searched. We describe a general parallel search method, based on statistical mechanics, and we show how it leads to a gen-

CHAPTER 7

Learning and Relearning in Boltzmann Machines

G. E. HINTON and T. J. SEJNOWSKI

Many of the chapters in this volume make use of the ability of a parallel network to perform cooperative searches for good solutions to problems. The basic idea is simple: The weights on the connections between processing units encode knowledge about how things normally fit together in some domain and the initial states or external inputs to a subset of the units encode some fragments of a structure within the domain. These fragments constitute a problem: What is the whole structure from which they probably came? The network computes a "good solution" to the problem by repeatedly updating the states of units that represent possible other parts of the structure until the network eventually settles into a stable state of activity that represents the solution.

One field in which this style of computation seems particularly appropriate is vision (Ballard, Hinton, & Sejnowski, 1983). A visual system must be able to solve large constraint-satisfaction problems rapidly in order to interpret a two-dimensional intensity image in terms of the depths and orientations of the three-dimensional surfaces in the world that gave rise to that image. In general, the information in the image is not sufficient to specify the three-dimensional surfaces unless the interpretive process makes use of additional plausible constraints about the kinds of structures that typically appear. Neighboring pieces of an image, for example, usually depict fragments of surface that have similar depths, similar surface orientations, and the same reflectance. The most plausible interpretation of an image is the one that satisfies

Hummel and Zucker (1983) and Hopfield (1982) have shown that some relaxation schemes have an associated "potential" or cost function and that the states to which the network converges are local minima of this function. This means that the networks are performing optimization of a well-defined function. Unfortunately, there is no guarantee that the network will find the best minimum. One possibility is to redefine the problem as finding the local minimum which is closest to the initial state. This is useful if the minima are used to represent "items" in a memory, and the initial states are queries to memory which may contain missing or erroneous information. The network simply finds the minimum that best fits the query. This idea was used by Hopfield (1982) who introduced an interesting kind of network in which the units were always in one of two states.[1] Hopfield showed that if the units are symmetrically connected (i.e., the weight from unit i to unit j exactly equals the weight from unit j to unit i) and if they are updated one at a time, each update reduces (or at worst does not increase) the value of a cost function which he called "energy" because of the analogy with physical systems. Consequently, repeated iterations are guaranteed to find an energy minimum. The global energy of the system is defined as

$$E = -\sum_{i<j} w_{ij} s_i s_j + \sum_i \theta_i s_i \qquad (1)$$

where w_{ij} is the strength of connection (synaptic weight) from the jth to the ith unit, s_i is the state of the ith unit (0 or 1), and θ_i is a threshold.

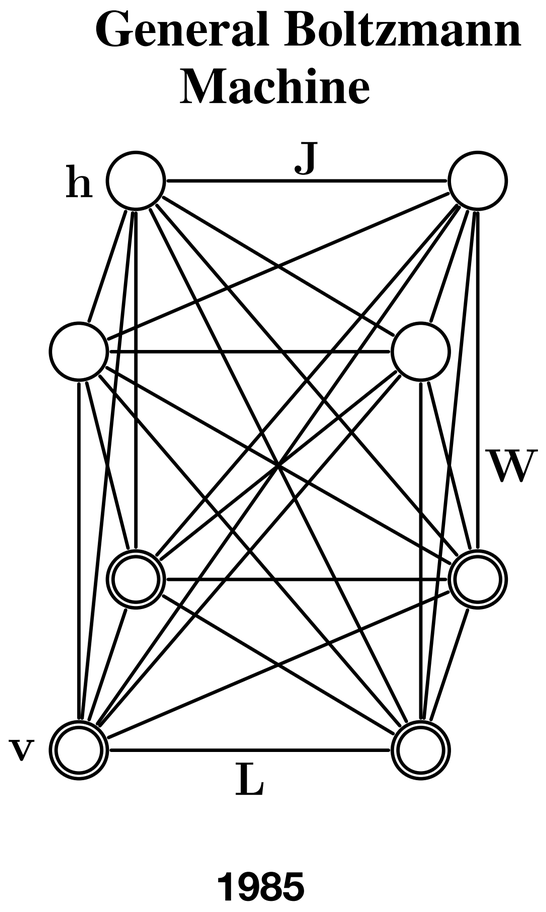
The updating rule is to switch each unit into whichever of its two states yields the lower total energy given the current states of the other units. Because the connections are symmetrical, the difference between the energy of the whole system with the kth hypothesis false and its energy with the kth hypothesis true can be determined locally by the kth unit, and is just

$$\Delta E_k = \sum_i w_{ki} s_i - \theta_k. \qquad (2)$$

Therefore, the rule for minimizing the energy contributed by a unit is to adopt the true state if its total input from the other units exceeds its threshold. This is the familiar rule for binary threshold units.

[1] Hopfield used the states 1 and −1 because his model was derived from physical systems called spin glasses in which spins are either "up" or "down." Provided the units have thresholds, models that use 1 and −1 can be translated into models that use 1 and 0 and have different thresholds.

The Evolution of Boltzmann Machines



1985

Salakhutdinov, R., & Larochelle, H. (2010, March). Efficient learning of deep Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 693-700).

Restricted Boltzmann Machines(1986)

CHAPTER 6

Information Processing in Dynamical Systems: Foundations of Harmony Theory

P. SMOLENSKY

INTRODUCTION

The Theory of Information Processing

At this early stage in the development of cognitive science, methodological issues are both open and central. There may have been times when developments in neuroscience, artificial intelligence, or cognitive psychology seduced researchers into believing that their discipline was on the verge of discovering the secret of intelligence. But a humbling history of hopes disappointed has produced the realization that understanding the mind will challenge the power of all these methodologies combined.

The work reported in this chapter rests on the conviction that a methodology that has a crucial role to play in the development of cognitive science is *mathematical analysis*. The success of cognitive science, like that of many other sciences, will, I believe, depend upon the construction of a solid body of theoretical results: results that express in a mathematical language the conceptual insights of the field; results that squeeze all possible implications out of those insights by exploiting powerful mathematical techniques.

This body of results, which I will call the *theory of information processing*, exists because information is a concept that lends itself to mathematical formalization. One part of the theory of information processing is already well-developed. The classical theory of computation provides powerful and elegant results about the notion of *effective*

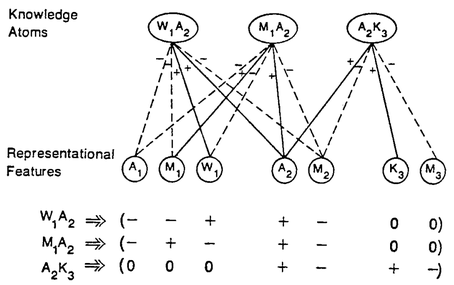


FIGURE 4. Each knowledge atom is a vector of +, −, and 0 values of the representational feature nodes.

line-segment units, which are like those in the original letter-perception model.

This simple model illustrates several points about the nature of knowledge atoms in harmony theory. The digraph unit W_1A_2 represents a pattern of values over the letter units: W_1 and A_2 on, with all other letter units for positions 1 and 2 off. This pattern is shown in Figure 4, using the labels +, −, and 0 to denote *on*, *off*, and *irrelevant*. These indicate whether there is an excitatory connection, inhibitory connection, or no connection between the corresponding nodes.[7]

Figure 4 shows the basic structure of harmony models. There are atoms of knowledge, represented by nodes in an upper layer, and a lower layer of nodes that comprises a representation of the state of the perceptual or problem domain with which the system deals. Each node is a *feature* in the representation of the domain. We can now view "atoms of knowledge" like W_1 and A_2 in several ways. Mathematically, each atom is simply a *vector* of +, −, and 0 values, one for each node in the lower, representation layer. This pattern can also be viewed as a *fragment* of a percept: The 0 values mark those features omitted in the fragment. This fragment can in turn be interpreted as a *trace* left behind in memory by perceptual experience.

[7] Omitted are the knowledge atoms that relate the letter nodes to the line segment nodes. Both line segment and letter nodes are in the lower layer, and all knowledge atoms are in the upper layer. Hierarchies in harmony theory are imbedded within an architecture of only two layers of nodes, as will be discussed in Section 2.

Point 5. Knowledge atoms are fragments of representations that accumulate with experience.

THE COMPLETION TASK

Having specified more precisely what the atoms of knowledge are, it is time to specify the task in which they are used.

Many cognitive tasks can be viewed as inference tasks. In problem solving, the role of inference is obvious; in perception and language comprehension, inference is less obvious but just as central. In harmony theory, a tightly prescribed but extremely general inferential task is studied: the *completion task*. In a problem-solving completion task, a partial description of a situation is given (for example, the initial state of a system); the problem is to complete the description to fill in the missing information (the final state, say). In a story understanding completion task, a partial description of some events and actors' goals is given; comprehension involves filling in the missing events and goals. In perception, the stimulus gives values for certain low-level features of the environmental state, and the perceptual system must fill in values for other features. In general, in the completion task some features of an environmental state are given as input, and the cognitive system must complete that input by assigning likely values to unspecified features.

A simple example of a completion task (Lindsay & Norman, 1972) is shown in Figure 5. The task is to fill in the features of the obscured portions of the stimulus and to decide what letters are present. This task can be performed by the model shown in Figure 3, as follows. The stimulus assigns values of *on* and *off* to the unobscured letter features. What happens is summarized in Table 1.

Note that which atoms are activated affects how the representation is

FIGURE 5. A perceptual completion task.

Restricted Boltzmann Machines(2002)

ARTICLE Communicated by Javier Movellan

Training Products of Experts by Minimizing Contrastive Divergence

Geoffrey E. Hinton
hinton@cs.toronto.edu
Gatsby Computational Neuroscience Unit, University College London, London WC1N 3AR, U.K.

It is possible to combine multiple latent-variable models of the same data by multiplying their probability distributions together and then renormalizing. This way of combining individual "expert" models makes it hard to generate samples from the combined model but easy to infer the values of the latent variables of each expert, because the combination rule ensures that the latent variables of different experts are conditionally independent when given the data. A product of experts (PoE) is therefore an interesting candidate for a perceptual system in which rapid inference is vital and generation is unnecessary. Training a PoE by maximizing the likelihood of the data is difficult because it is hard even to approximate the derivatives of the renormalization term in the combination rule. Fortunately, a PoE can be trained using a different objective function called "contrastive divergence" whose derivatives with regard to the parameters can be approximated accurately and efficiently. Examples are presented of contrastive divergence learning using several types of expert on several types of data.

1 Introduction

One way of modeling a complicated, high-dimensional data distribution is to use a large number of relatively simple probabilistic models and somehow combine the distributions specified by each model. A well-known example of this approach is a mixture of gaussians in which each simple model is a gaussian, and the combination rule consists of taking a weighted arithmetic mean of the individual distributions. This is equivalent to assuming an overall generative model in which each data vector is generated by first choosing one of the individual generative models and then allowing that individual model to generate the data vector. Combining models by forming a mixture is attractive for several reasons. It is easy to fit mixtures of tractable models to data using expectation-maximization (EM) or gradient ascent, and mixtures are usually considerably more powerful than their individual components. Indeed, if sufficiently many models are included in

Neural Computation **14, 1771–1800** (2002)

7 PoEs and Boltzmann Machines

The Boltzmann machine learning algorithm (Hinton & Sejnowski, 1986) is theoretically elegant and easy to implement in hardware but very slow in networks with interconnected hidden units because of the variance problems described in section 2. Smolensky (1986) introduced a restricted type of Boltzmann machine with one visible layer, one hidden layer, and no intralayer connections. Freund and Haussler (1992) realized that in this restricted Boltzmann machine (RBM), the probability of generating a visible vector is proportional to the product of the probabilities that the visible vector would be generated by each of the hidden units acting alone. An RBM is therefore a PoE with one expert per hidden unit.[7] When the hidden unit

[7] Boltzmann machines and PoEs are very different classes of probabilistic generative model, and the intersection of the two classes is RBMs.

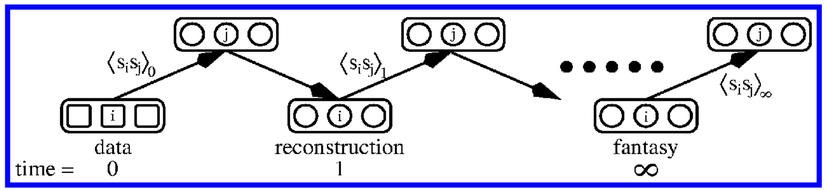
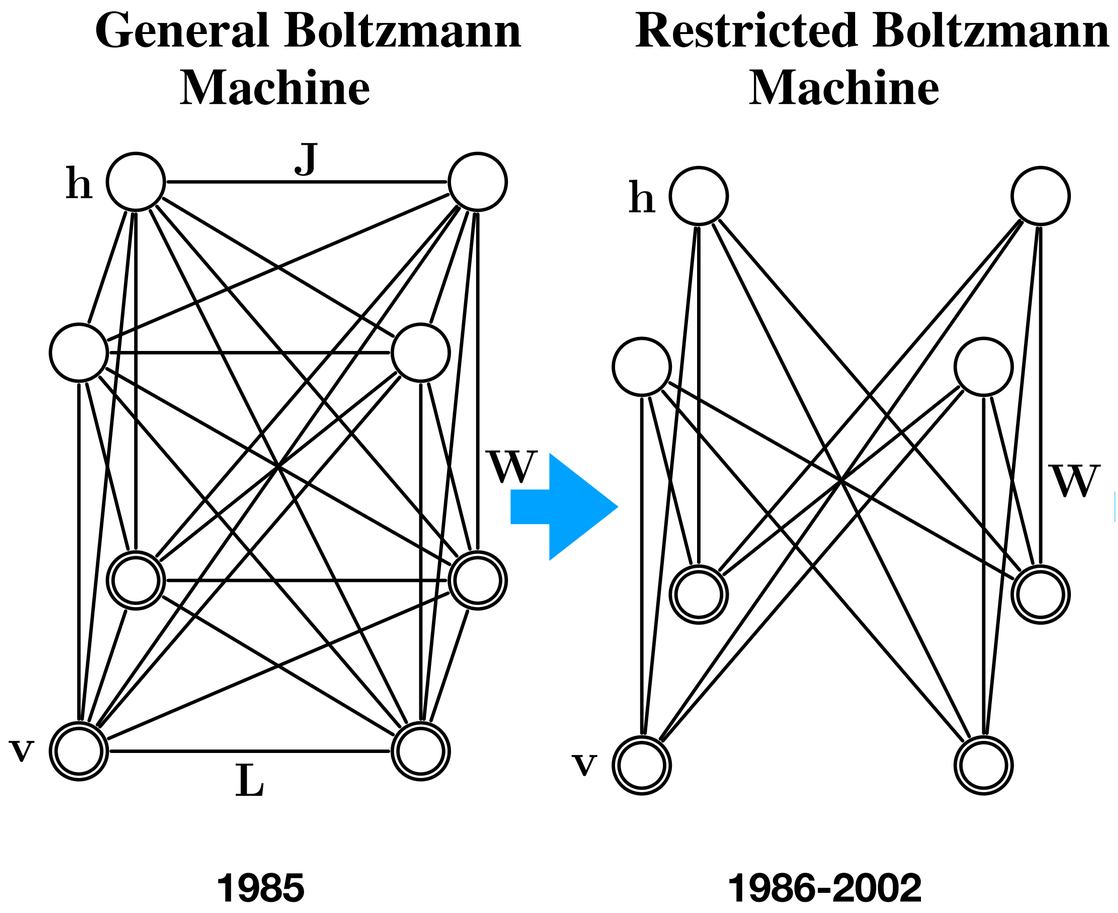


Figure 1: A visualization of alternating Gibbs sampling. At time 0, the visible variables represent a data vector, and the hidden variables of all the experts are updated in parallel with samples from their posterior distribution given the visible variables. At time 1, the visible variables are all updated to produce a reconstruction of the original data vector from the hidden variables, and then the hidden variables are again updated in parallel. If this process is repeated sufficiently often, it is possible to get arbitrarily close to the equilibrium distribution. The correlations $\langle s_i s_j \rangle$ shown on the connections between visible and hidden variables are the statistics used for learning in RBMs, which are described in section 7.

The Evolution of Boltzmann Machines



Salakhutdinov, R., & Larochelle, H. (2010, March). Efficient learning of deep Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 693-700).

Deep Belief Networks(2006)

A fast learning algorithm for deep belief nets *

Geoffrey E. Hinton and **Simon Osindero**
Department of Computer Science University of Toronto
10 Kings College Road
Toronto, Canada M5S 3G4
{hinton, osindero}@cs.toronto.c

Yee-Whye Teh
Department of Computer Science
National University of Singapore
3 Science Drive 3, Singapore, 117543
tchyw@comp.nus.edu.sg

Abstract

We show how to use "complementary priors" to eliminate the explaining away effects that make inference difficult in densely-connected belief nets that have many hidden layers. Using complementary priors, we derive a fast, greedy algorithm that can learn deep, directed belief networks one layer at a time, provided the top two layers form an undirected associative memory. The fast, greedy algorithm is used to initialize a slower learning procedure that fine-tunes the weights using a contrastive version of the wake-sleep algorithm. After fine-tuning, a network with three hidden layers forms a very good generative model of the joint distribution of handwritten digit images and their labels. This generative model gives better digit classification than the best discriminative learning algorithms. The low-dimensional manifolds on which the digits lie are modelled by long ravines in the free-energy landscape of the top-level associative memory and it is easy to explore these ravines by using the directed connections to display what the associative memory has in mind.

1 Introduction

remaining hidden layers form a directed acyclic graph that converts the representations in the associative memory into observable variables such as the pixels of an image. This hybrid model has some attractive features:

1. There is a fast, greedy learning algorithm that can find a fairly good set of parameters quickly, even in deep networks with millions of parameters and many hidden layers.
2. The learning algorithm is unsupervised but can be applied to labeled data by learning a model that generates both the label and the data.
3. There is a fine-tuning algorithm that learns an excellent generative model which outperforms discriminative methods on the MNIST database of hand-written digits.
4. The generative model makes it easy to interpret the distributed representations in the deep hidden layers.
5. The inference required for forming a percept is both fast and accurate.
6. The learning algorithm is local: adjustments to a synapse strength depend only on the states of the presynaptic and post-synaptic neuron.
7. The communication is simple: neurons only need to communicate their stochastic binary states.

LETTER Communicated by Terrence Sejnowski

Representational Power of Restricted Boltzmann Machines and Deep Belief Networks

Nicolas Le Roux
lerouxni@iro.umontreal.ca
Yoshua Bengio
bengioy@iro.umontreal.ca

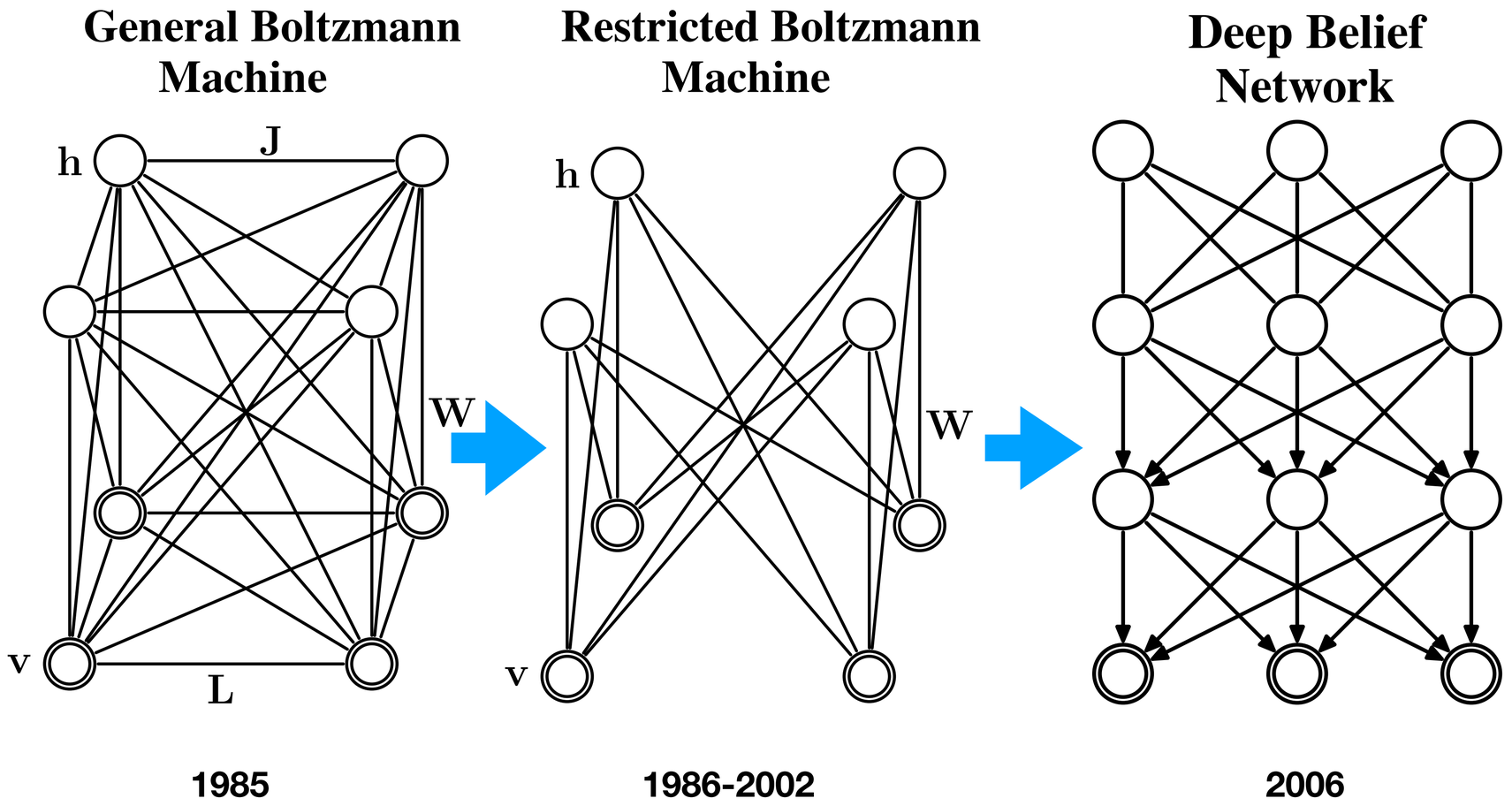
Département Informatique et Recherche Opérationnelle, Université de Montréal, Montréal, Québec, H3C 3J7, Canada

Deep belief networks (DBN) are generative neural network models with many layers of hidden explanatory factors, recently introduced by Hinton, Osindero, and Teh (2006) along with a greedy layer-wise unsupervised learning algorithm. The building block of a DBN is a probabilistic model called a restricted Boltzmann machine (RBM), used to represent one layer of the model. Restricted Boltzmann machines are interesting because inference is easy in them and because they have been successfully used as building blocks for training deeper models. We first prove that adding hidden units yields strictly improved modeling power, while a second theorem shows that RBMs are universal approximators of discrete distributions. We then study the question of whether DBNs with more layers are strictly more powerful in terms of representational power. This suggests a new and less greedy criterion for training RBMs within DBNs.

1 Introduction

Learning algorithms that learn to represent functions with many levels of

The Evolution of Boltzmann Machines



Salakhutdinov, R., & Larochelle, H. (2010, March). Efficient learning of deep Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 693-700).

Deep Boltzmann Machines(2009)

Deep Boltzmann Machines

Ruslan Salakhutdinov
Department of Computer Science
University of Toronto
rsalakhu@cs.toronto.edu

Geoffrey Hinton
Department of Computer Science
University of Toronto
hinton@cs.toronto.edu

Abstract

We present a new learning algorithm for Boltzmann machines that contain many layers of hidden variables. Data-dependent expectations are estimated using a variational approximation that tends to focus on a single mode, and data-independent expectations are approximated using persistent Markov chains. The use of two quite different techniques for estimating the two types of expectation that enter into the gradient of the log-likelihood makes it practical to learn Boltzmann machines with multiple hidden layers and millions of parameters. The learning can be made more efficient by using a layer-by-layer "pre-training" phase that allows variational inference to be initialized with a single bottom-up pass. We present results on the MNIST and NORB datasets showing that deep Boltzmann machines learn good generative models and perform well on handwritten digit and visual object recognition tasks.

units (Hinton, 2002). Multiple hidden layers can be learned by treating the hidden activities of one RBM as the data for training a higher-level RBM (Hinton et al., 2006; Hinton and Salakhutdinov, 2006). However, if multiple layers are learned in this greedy, layer-by-layer way, the resulting composite model is *not* a multilayer Boltzmann machine (Hinton et al., 2006). It is a hybrid generative model called a "deep belief net" that has undirected connections between its top two layers and downward directed connections between all its lower layers.

In this paper we present a much more efficient learning procedure for fully general Boltzmann machines. We also show that if the connections between hidden units are restricted in such a way that the hidden units form multiple layers, it is possible to use a stack of slightly modified RBM's to initialize the weights of a deep Boltzmann machine before applying our new learning procedure.

2 Boltzmann Machines (BM's)

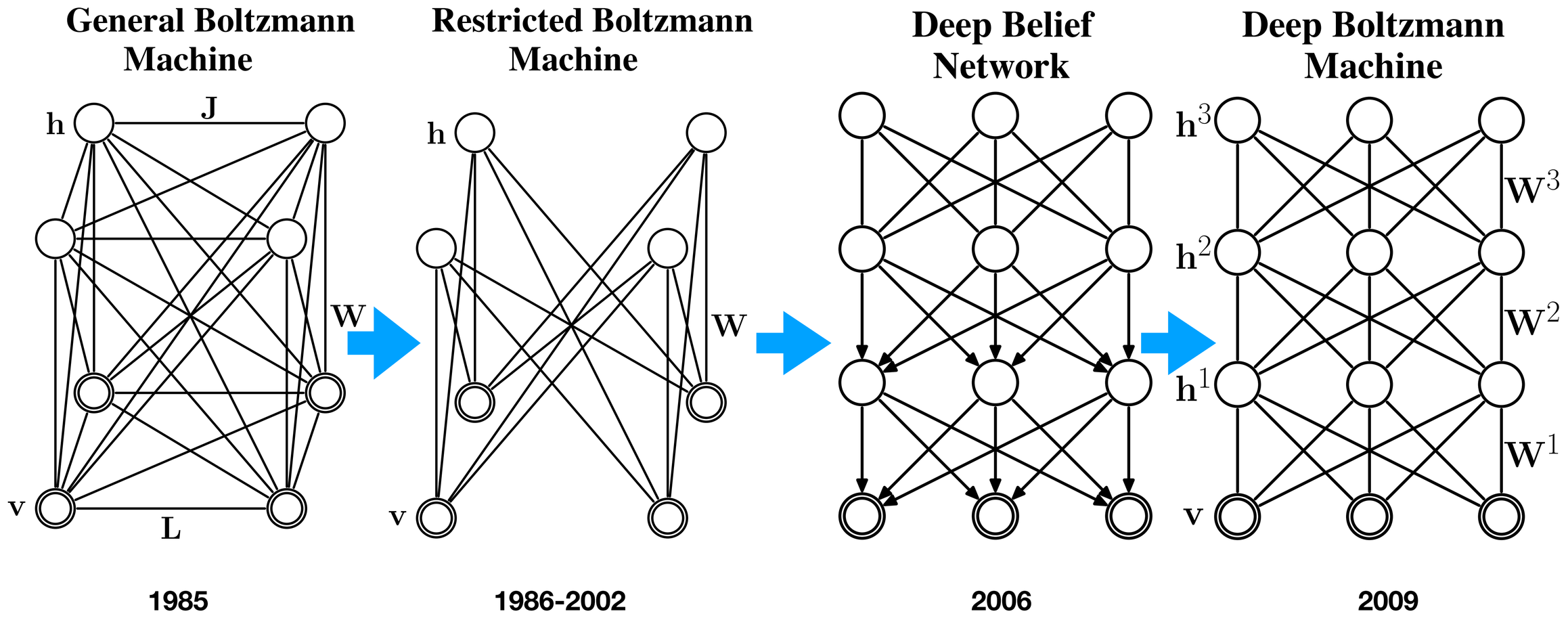
A Boltzmann machine is a network of symmetrically coupled stochastic binary units. It contains a set of visible units $\mathbf{v} \in \{0,1\}^D$, and a set of hidden units $\mathbf{h} \in \{0,1\}^P$ (see

2-layer BM

3-layer BM

Training Samples

The Evolution of Boltzmann Machines



Salakhutdinov, R., & Larochelle, H. (2010, March). Efficient learning of deep Boltzmann machines. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (pp. 693-700).

A Summary of the History

- General BMs was proposed as a type of cognitive models!
- RBMs became popular because of efficient training algorithm.
- Deep architecture arose to create more powerful BMs by stacking RBMs.
- DBMs are still one of the hardest types of neural networks to train today.

A Summary of the History

- Gene
- RBM
- Deep ng RBMs.
- DBM ain today.

Are there tractable learning algorithms for architectures that are stacks of RBMs?

The Challenge

General Boltzmann Machines

$$\mathbf{s} = (\mathbf{s}_h, \mathbf{s}_v) \qquad \theta = (\mathbf{a}, W)$$

$$p(\mathbf{s}) = \frac{1}{Z} \exp\left\{\mathbf{a}^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T W \mathbf{s}\right\}$$

$$Z = \sum_{\mathbf{s}} \exp\left\{\mathbf{a}^T\mathbf{s} + \frac{1}{2}\mathbf{s}^T W \mathbf{s}\right\}$$

Inference for Learning

$$\mathcal{D} = \{\mathbf{s}_v^{(i)}\}_{i=1}^N$$

$$-\partial_W \log p(\mathcal{D}|\theta) = \frac{1}{N} \sum_n \langle \mathbf{s}\mathbf{s}^T \rangle_{p_{\theta, \mathbf{s}_v^{(n)}}} - \langle \mathbf{s}\mathbf{s}^T \rangle_{p_\theta}$$

$$\mathbf{s}_h \sim p_\theta(\mathbf{s}_h|\mathbf{s}_v^{(n)}) \qquad \mathbf{s} \sim p_\theta(\mathbf{s})$$

Inference for Learning

$$-\partial_W \mathbf{L} \qquad \langle \qquad \rangle_{p_\theta}$$

$$-\partial_\theta \log p(\mathcal{D}|\theta) = 0 \text{ if } p_\theta(\mathbf{s}_v) = p(\mathcal{D})$$

$$\mathbf{s}_h \qquad p_\theta(\mathbf{s}_h|\mathbf{s}_v) \qquad \mathbf{s} \qquad p_\theta(\mathbf{s})$$

The Challenges of Inference

- The partition function is intractable.
- Unconstrained dependency structure.
- Gradient of discrete variables is not defined.
- Boltzmann machines can be highly multimodal.

The Challenges of Inference

- The p
- Unco
- Grad
- Boltz

All bad news

Probabilistic Continuous Relaxation

The Motivation

- Challenge 2 and 3
- Relaxing discrete variables to continuous ones in optimization

The Question

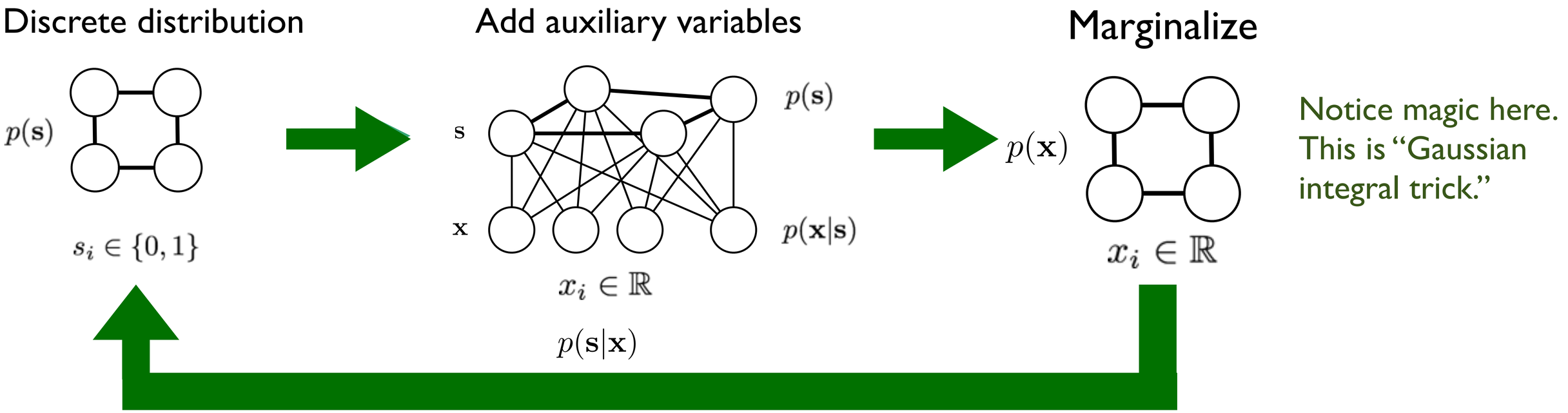
How to relax discrete random variables and preserve the distribution at the same time?

Probabilistic Relaxation

**Carefully add relaxed variables to the discrete variables,
so we can recover the discrete variables exactly.**

Probabilistic Relaxation

**Carefully add relaxed variables to the discrete variables,
so we can recover the discrete variables exactly.**



Probabilistic Relaxation for Boltzmann Machines

$$p(\mathbf{s}) = \frac{1}{Z} \exp\left\{ \mathbf{a}^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T W \mathbf{s} \right\}$$

$$p(s_i|\mathbf{x}) = \sigma\left(-a_i - x_i + \frac{d_i}{2}\right)^{1-s_i} \sigma\left(a_i + x_i - \frac{d_i}{2}\right)^{s_i}$$

Binary variables are Independent given relaxation

$$p(\mathbf{x}|\mathbf{s}) = \mathcal{N}(\mathbf{x}; A(W+D)\mathbf{s}, A(W+D)A^T)$$

Each configuration defines the mean of correlated Gaussian

$$p(\mathbf{x}) = Z^{-1} |2\pi(W+D)|^{-1/2} \exp\left\{ -\frac{1}{2} \mathbf{x}^T (W+D)^{-1} \mathbf{x} \right\} \prod_i \left(1 + \exp\left\{ a_i + x_i - \frac{d_i}{2} \right\}\right)$$

An Interpretation of the Relaxation of Boltzmann machines

$$p(\mathbf{x}) = Z^{-1}|2\pi(W+D)|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{x}^T(W+D)^{-1}\mathbf{x}\right\}\prod_i\left(1+\exp\left\{a_i+x_i-\frac{d_i}{2}\right\}\right)$$

An Interpretation of the Relaxation of Boltzmann machines

$$p(\mathbf{x}) = Z^{-1}|2\pi(W+D)|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{x}^T(W+D)^{-1}\mathbf{x}\right\}\prod_i\left(1+\exp\left\{a_i+x_i-\frac{d_i}{2}\right\}\right)$$

Gaussian density function

Log Concave

An Interpretation of the Relaxation of Boltzmann machines

$$p(\mathbf{x}) = Z^{-1}|2\pi(W+D)|^{-1/2}\exp\left\{-\frac{1}{2}\mathbf{x}^T(W+D)^{-1}\mathbf{x}\right\}\prod_i\left(1+\exp\left\{a_i+x_i-\frac{d_i}{2}\right\}\right)$$

Gaussian density function

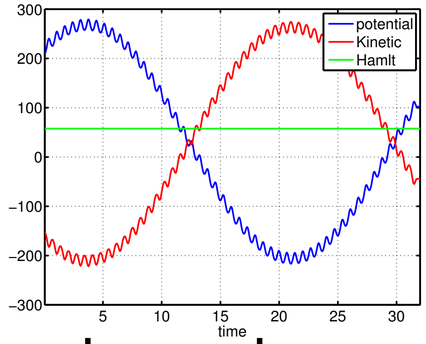
Log Concave

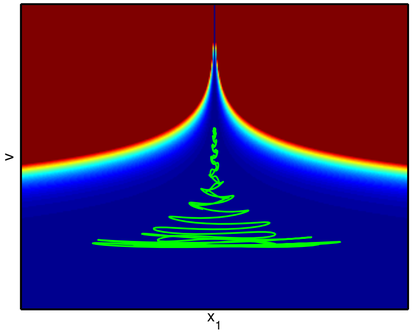
Normalization of logistic parameterised Bernoulli

Log Convex

Hamiltonian Monte Carlo

Hamiltonian Monte Carlo





- A MCMC sampler for continuous distributions
- Require derivative of gradients for simulating the dynamics
- Explore sample space by Hamiltonian dynamics rather than random walk
- Tolerant to strong correlation structure
- Outperform many other MCMC methods in high dimensional space

HMC for PR Boltzmann Machines

- The gradient of probabilistic relaxation of Boltzmann machines is easy to compute.
- HMC is available for Boltzmann machines now!
- Straightforward to adapt HMC with the structure of Boltzmann machines.
- But, the multimodality of relaxed Boltzmann machines makes HMC performs poorly in general.

Geometry

- Geometric perspective has been explored in slice sampling.
- Information geometry studies the space of probability distribution families.
- Recent advance in Manifold HMC reveals promising applications of Information geometry for Bayesian inference.
- Geometry is also crucial in sampling PR Boltzmann machines

Beyond Deep Boltzmann Machines

In a world not just deep

- The computation complexity of learning on general architecture
- Generalize learning algorithm for other undirected graphical models
- Continual learning with general Boltzmann machines
- Next inference challenge: Bayesian General Boltzmann machines!

Charles Sutton

Zoubin Ghahramani

Amos Storkey

Thank you
and
Questions!