Variants of Restricted Boltzmann Machines(RBMs)

Zhifei Zhang

Department of Computer Science and Technology
Tongji University

Nov. 27, 2013

Outline

Outline

Linear Equations

Example:

$3x_1 + 4x_2 = 8$, the solution is not unique.

Problem:

- Given $y \in \mathbb{R}^m$, $A \in \mathbb{Z}^{m \times n}$ (typically, $m \ll n$)
- Find a solution $x \in \mathbb{R}^n$, that satisfies $Ax = y$
- x is as sparse as possible

$$\min_x \|x\|_0 \quad \text{s.t.} \quad Ax = y \tag{1}$$

Solution:

$$\min_x \|x\|_1 \quad \text{s.t.} \quad Ax = y \tag{2}$$

$$\min_x (\|Ax - y\|_2^2 + \lambda \|x\|_1) \tag{3}$$

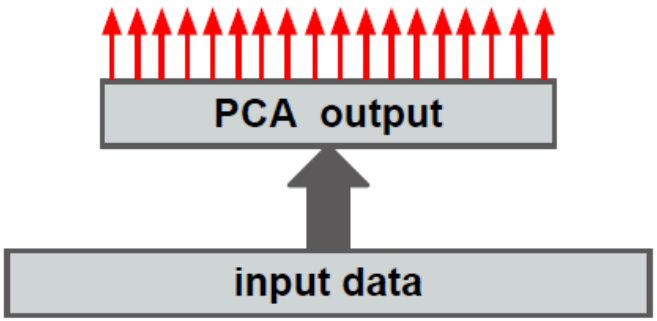
Sparse Coding [Olshausen and Field, VR 1997]

Given data $X \in \mathbb{R}^{d \times n}$, learn a dictionary $D \in \mathbb{R}^{d \times k}$ encoding $A \in \mathbb{R}^{k \times n}$ ($k > d$) which satisfies $X \approx DA$ ($\mathbf{x} = \sum_{i=1}^{k} a_i \phi_i$).

An overcomplete basis set is better able to capture structures and patterns inherent in the input data.

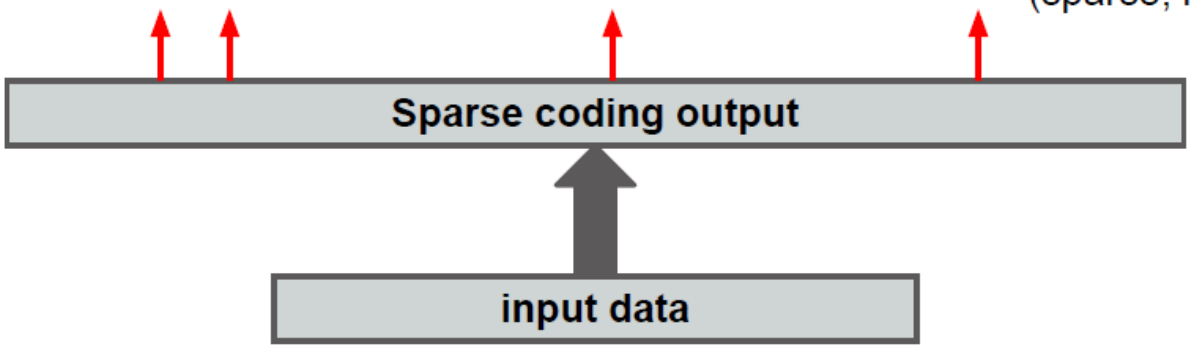
The coefficients a_i are no longer uniquely determined by the input vector $\mathbf{x}$. Thus, introduce an additional criterion of sparsity which means that each $\mathbf{x}$ is explained by few codewords.

Sparse Coding vs PCA



(dense, low-dimensional)

(sparse, high-dimensional)



Cost Function

$$\min_{a_i^{(j)},\phi_i} \sum_{j=1}^{n} \left\| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)}) \tag{4}$$

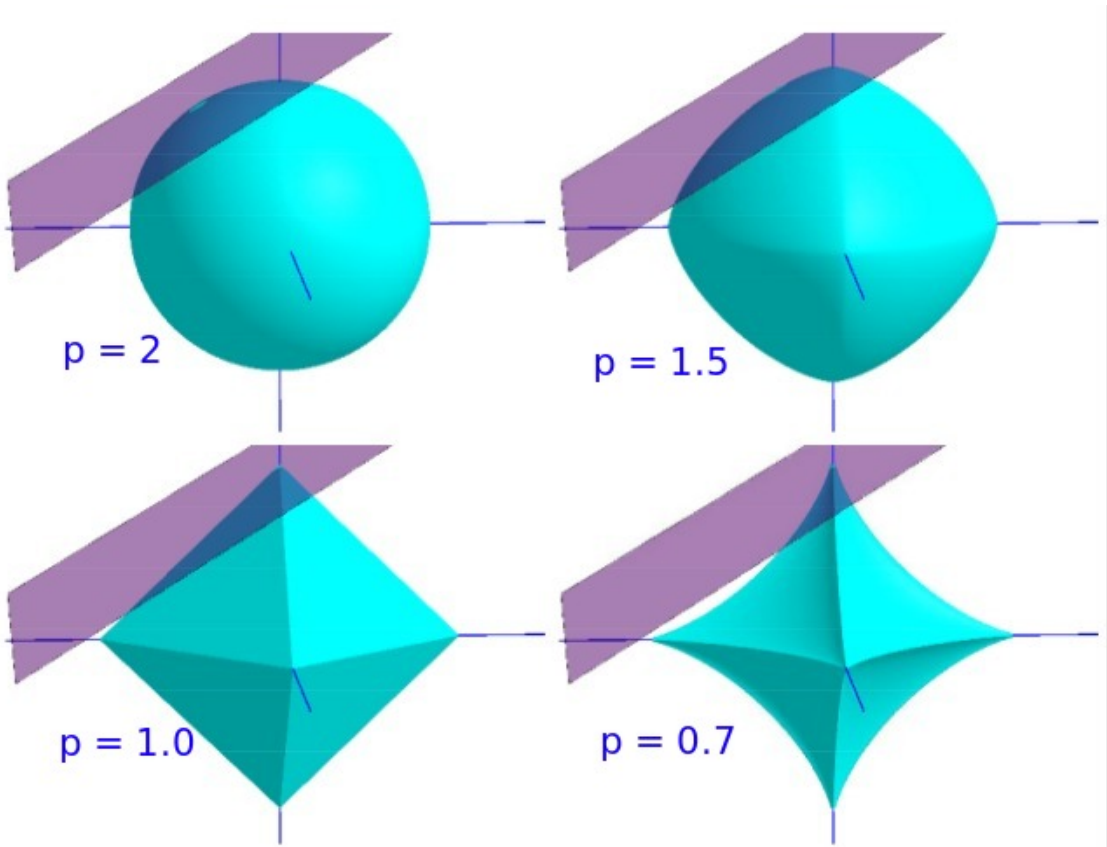
where $S(a_i) = \|a_i\|_1$ or $S(a_i) = \log(1 + \|a_i\|_2^2)$.

It's possible to make the sparsity penalty arbitrarily small by scaling down a_i and scaling ϕ_i up by a large constant.

$$\begin{aligned} \min_{a_i^{(j)},\phi_i} & \sum_{j=1}^{n} \left\| \mathbf{x}^{(j)} - \sum_{i=1}^{k} a_i^{(j)} \phi_i \right\|^2 + \lambda \sum_{i=1}^{k} S(a_i^{(j)}) \\ \text{s.t.} \quad & \|\phi_i\|_2^2 \leq C, \forall i = 1, ..., k \end{aligned} \tag{5}$$

Why using l_1-norm as sparsity penalty

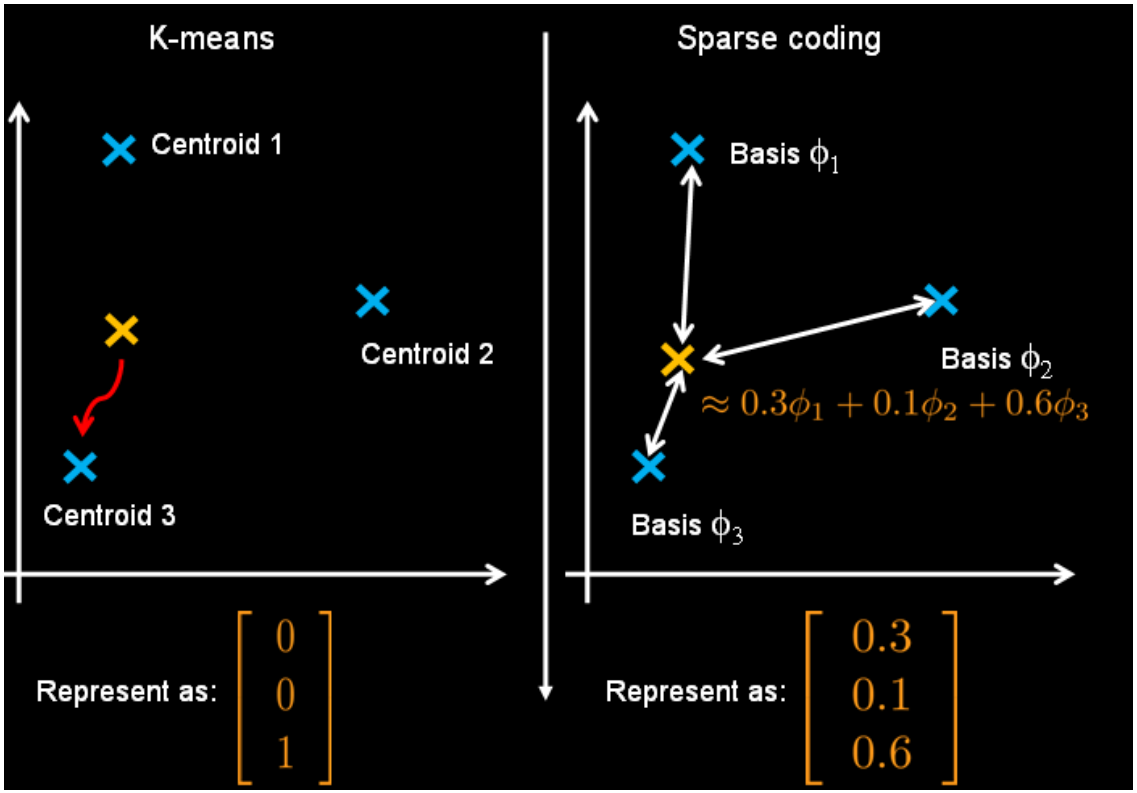
Although the most direct measure of sparsity is l_0-norm, it's non-differentiable and difficult to optimize in general.



Sparse Coding vs K-means

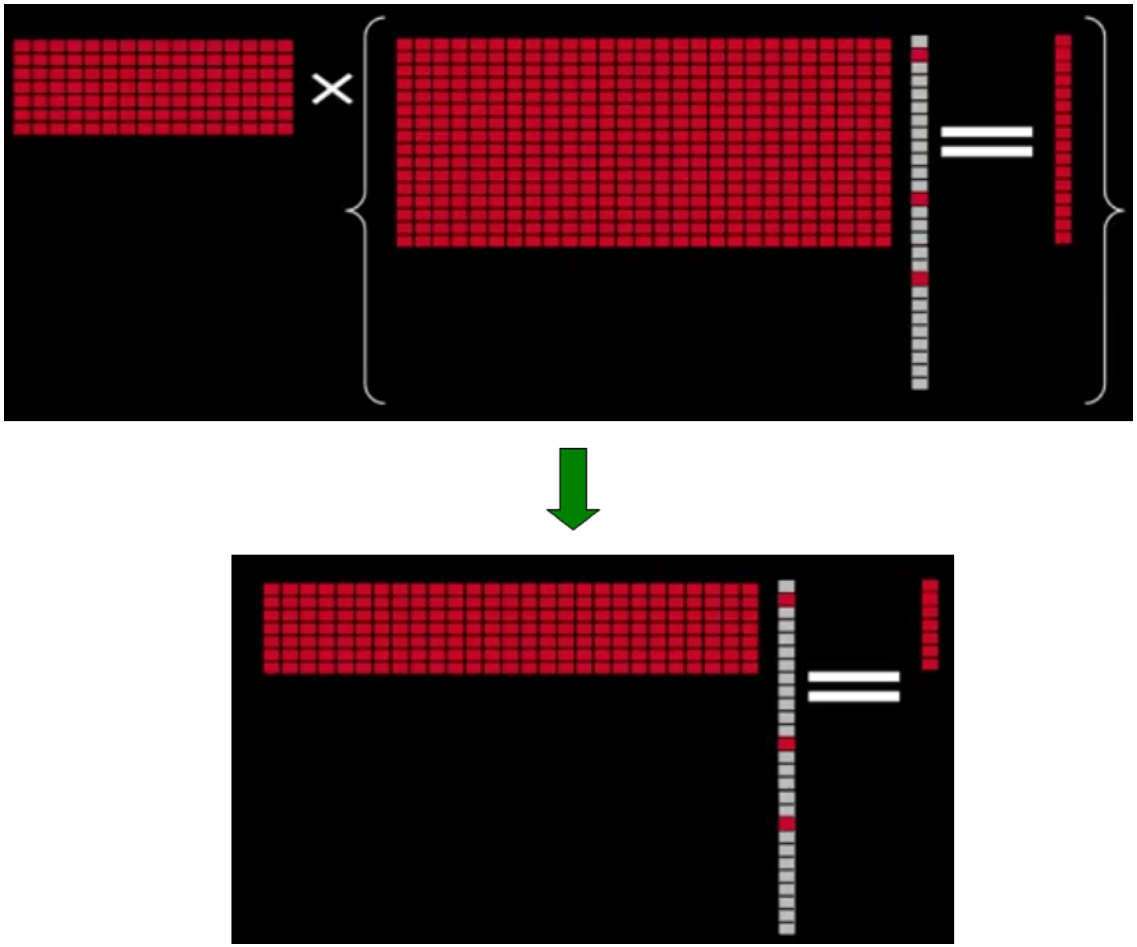
K-means:

$$\min_{U,V} \sum_{j=1}^{n} \left\| \mathbf{x}^{(j)} - \mathbf{u}^{(j)} V \right\|_2^2 \quad \text{s.t.} \quad \left\| \mathbf{u}^{(j)} \right\|_0 = 1 \tag{6}$$

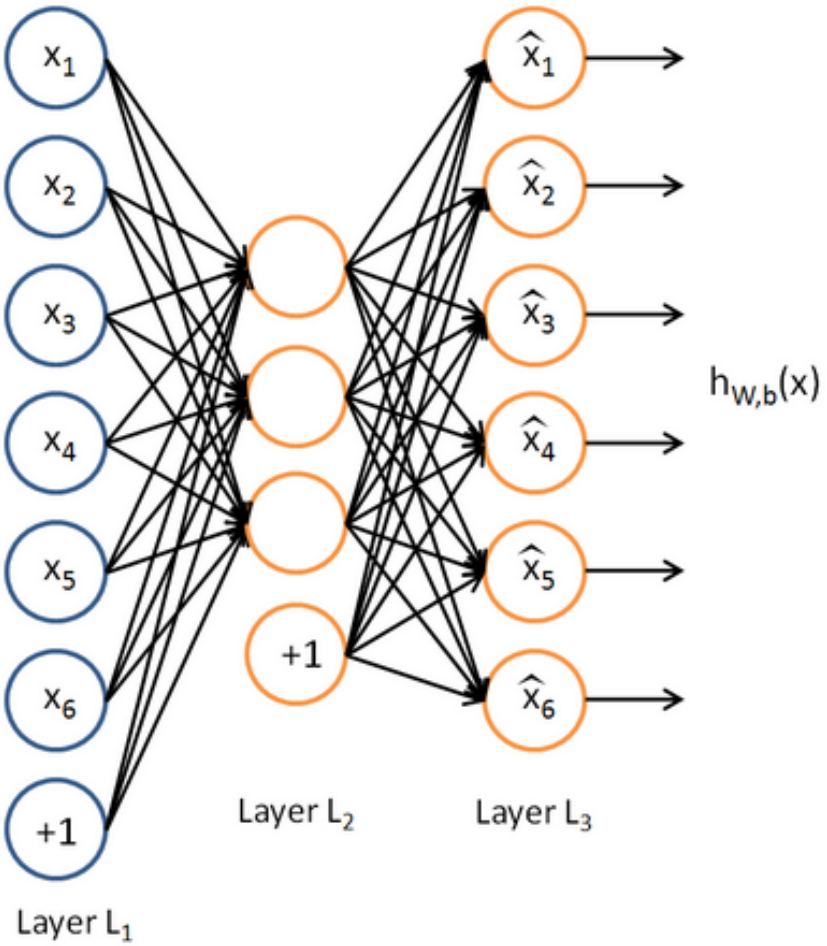


Sparse Coding vs Compressive Sensing

$$QDa = Qx \Rightarrow \tilde{D}a = \tilde{x}$$



An Example of Neural Network



Sparse Auto-Encoders [Ranzato et al., NIPS 2006]

$$J(W,b) = \left[\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}\left\|h_{W,b}(x^{(i)}) - x^{(i)}\right\|^2\right)\right] + \frac{\lambda}{2}\sum_{l=1}^{n_l-1}\sum_{i=1}^{s_l}\sum_{j=1}^{s_{l+1}}(W_{ji}^{(l)})^2 + \beta\sum_{j=1}^{s_2}\mathrm{KL}(\rho||\hat{\rho}_j) \tag{7}$$

$$\mathrm{KL}(\rho||\hat{\rho}_j) = \rho\log\frac{\rho}{\hat{\rho}_j} + (1-\rho)\log\frac{1-\rho}{1-\hat{\rho}_j} \tag{8}$$

where ρ is a sparsity parameter, which specifies our desired level of sparsity, typically a small value close to zero.

$$\hat{\rho}_j = \frac{1}{m}\sum_{i=1}^{m}[a_j^{(2)}(x^{(i)})] \tag{9}$$

Auto-Encoders vs PCA [Japkowicz et al., NECO 2000]

Nonlinear autoencoder is not equivalent to PCA

- Linear autoencoders emulate PCA and thus exhibit a flat or unimodal reconstruction error surface
- Autoencoders with nonlinearities in their hidden layer learn domains by building error reconstruction surfaces that, depending on the task, contain multiple local valleys.
- Nonlinear autoencoders can represent appropriate classifications of nonlinear multi-modal domains, in contrast to linear autoencoders which are inappropriate for such tasks.

Outline

Energy-Based Models [LeCun et al., 2006]

Energy-Based Models (EBMs) capture dependencies between variables by associating a scalar energy to each configuration of the variables.
Two tasks:

- Inference consists in setting the value of observed variables and finding configurations of the **hidden** variables that minimize the energy.
- Learning consists in finding an energy function in which observed configurations of the variables are given *lower* energies than hidden ones.

Any probability distribution can be cast as an energy-based model.

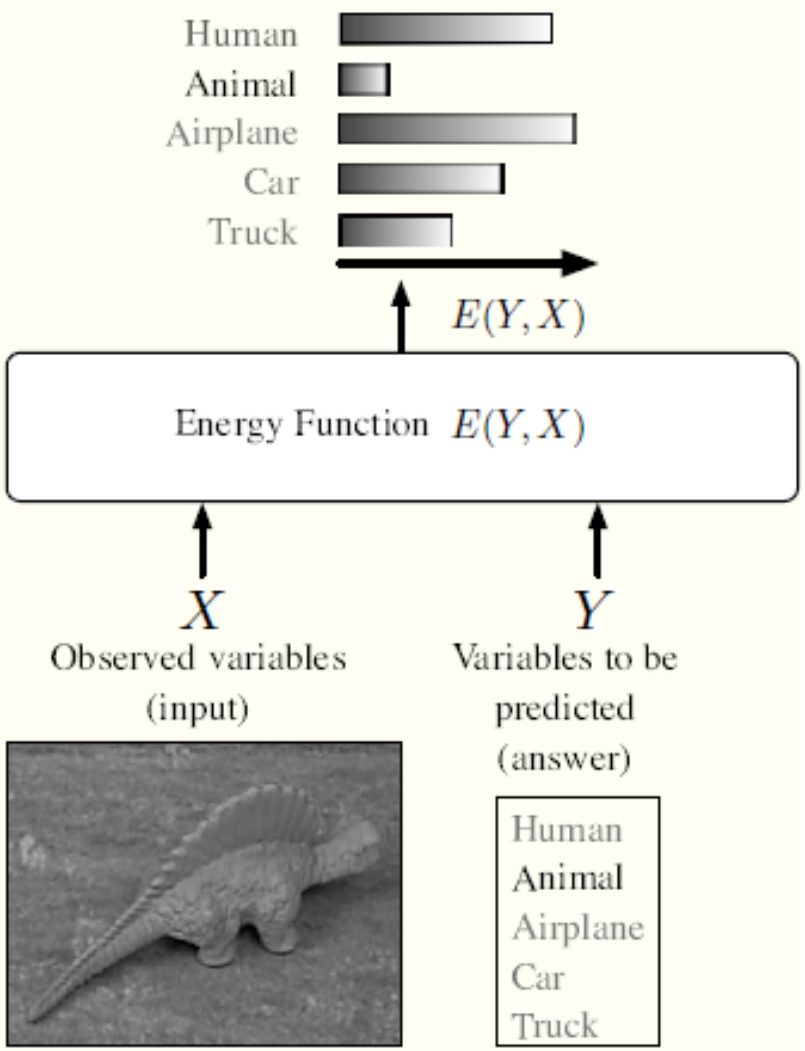
Energy-Based Models [LeCun et al., 2006]

Energy-Based Models (EBMs) capture dependencies between variables by associating a scalar energy to each configuration of the variables.
Two tasks:

- Inference consists in setting the value of observed variables and finding configurations of the **hidden** variables that minimize the energy.
- Learning consists in finding an energy function in which observed configurations of the variables are given *lower* energies than hidden ones.

Any probability distribution can be cast as an energy-based model.

Energy-Based Models (Cont')



Energy-Based Models (Cont')

Define a probability distribution through an energy function:

$$p(x) = \frac{e^{E(x)}}{Z} \tag{10}$$

where $Z = \sum_x e^{-E(x)}$ is called the *partition function*.

Negative log-likelihood loss:

$$l(\theta, D) = -L(\theta, D) = -\frac{1}{N} \sum_{x^{(i)} \in D} logp(x^{(i)}) \tag{11}$$

$$\Delta = \frac{\partial l(\theta, D)}{\partial \theta} = -\frac{1}{N} \frac{\partial \sum logp(x^{(i)})}{\partial \theta} \tag{12}$$

Introducing Hidden Variables

Consider an observed part x and a hidden part h,

$$P(x) = \sum_h P(x,h) = \sum_h \frac{e^{-E(x,h)}}{Z} \tag{13}$$

Introduce the notation of *free energy*,

$$F(x) = -log \sum_h e^{-E(x,h)} \tag{14}$$

$$P(x) = \frac{e^{-F(x)}}{Z} \tag{15}$$

where $Z = \sum_x e^{-F(x)}$.

Negative log-likelihood gradient:

$$\Delta = -\frac{\partial log p(x)}{\partial \theta} = -\frac{\partial(-F(x) - logZ)}{\partial \theta} = \frac{\partial F(x)}{\partial \theta} - \sum_{\hat{x}} p(\hat{x}) \frac{\partial F(\hat{x})}{\partial \theta} \quad (16)$$

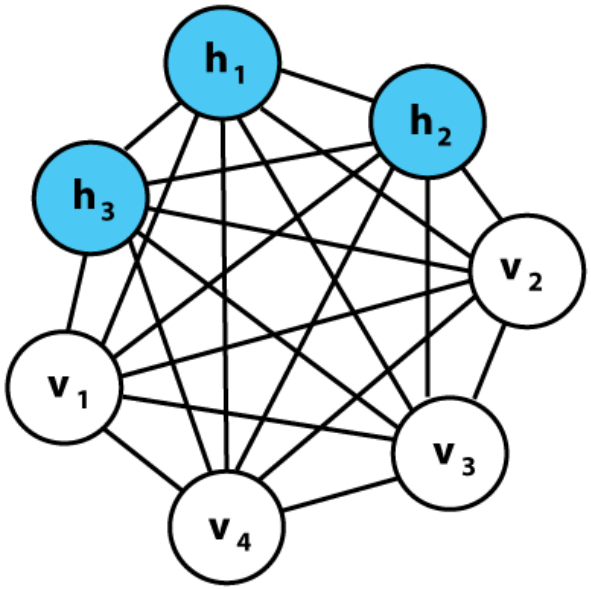
Positive phase: The first term increases the probability of **training** data (by reducing the corresponding free energy).
Negative phase: The second term decreases the probability of samples generated by the **model**.

It is usually difficult to determine this gradient analytically, as it involves the computation of $\mathbb{E}_P \left[\frac{\partial F(x)}{\partial \theta} \right]$. This is nothing less than an expectation over all possible configurations of the input (under the distribution P formed by the model).

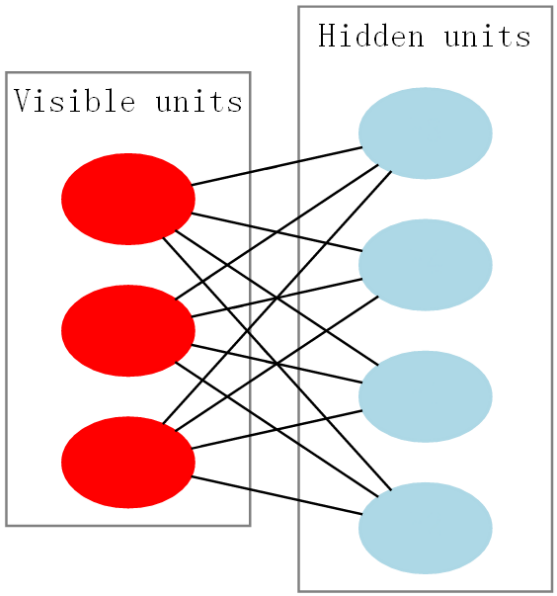
Boltzmann Machines vs Restricted Boltzmann Machines

BMs are a particular type of EBMs with hidden variables, and RBMs are a special form of BMs without visible-visible and hidden-hidden connections.

BMs[Ackley et al., CSJ 1985]:



RBMs[Hinton, NECO 2002]
(Harmonium [Smolensky, 1986]):



Boltzmann Machines [Ackley et al., CSJ 1985]

Energy function:

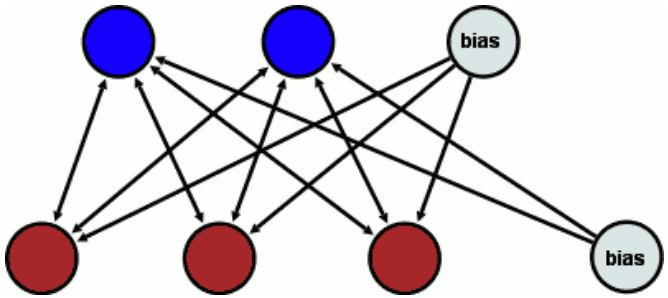
$$Energy(v,h) = -b'v - c'h - h'Wv - v'Hv - h'Vh \tag{17}$$

Two types of parameters: the offsets b_i and c_i (each associated with a single element of the vector v or h), and the wights W_{ij}, H_{ij} and V_{ij} (each associated with a pair of units).

Negative log-likelihood gradient:

$$\begin{aligned}\Delta &= -\frac{\partial \log P(v)}{\partial \theta} = -\frac{\partial \log \sum_h e^{-Energy(v,h)}}{\partial \theta} + \frac{\partial \log \sum_{\tilde{v},h} e^{-Energy(\tilde{v},h)}}{\partial \theta} \\ &= \sum\nolimits_h P(h|v)\frac{\partial Energy(v,h)}{\partial \theta} - \sum\nolimits_{\tilde{v},h} P(\tilde{v},h|v)\frac{\partial Energy(\tilde{v},h)}{\partial \theta}\end{aligned} \tag{18}$$

Restricted Boltzmann Machines [Hinton, NECO 2002]



$$Energy(v,h) = -b'v - c'h - h'Wv \tag{19}$$

$$F(v) = -b'v - \sum_i log \sum_{h_i} e^{h_i(c_i + W_i v)} \tag{20}$$

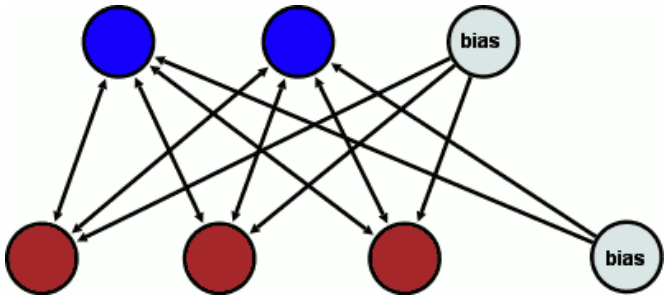
visible and hidden units are **conditionally independent**.

$$p(h|v) = \prod_i p(h_i|v) \tag{21}$$

$$p(v|h) = \prod_j p(v_j|h) \tag{22}$$

RBMs can represent any discrete distribution if enough hiddern units are used. [Roux and Bengio, NECO 2008]

Restricted Boltzmann Machines [Hinton, NECO 2002]



$$Energy(v,h) = -b'v - c'h - h'Wv \tag{19}$$

$$F(v) = -b'v - \sum_i log \sum_{h_i} e^{h_i(c_i + W_i v)} \tag{20}$$

visible and hidden units are **conditionally independent**.

$$p(h|v) = \prod_i p(h_i|v) \tag{21}$$

$$p(v|h) = \prod_j p(v_j|h) \tag{22}$$

RBMs can represent any discrete distribution if enough hiddern units are used. [Roux and Bengio, NECO 2008]

Learning RBMs

Contrastive Divergence [Hinton, NECO 2002]

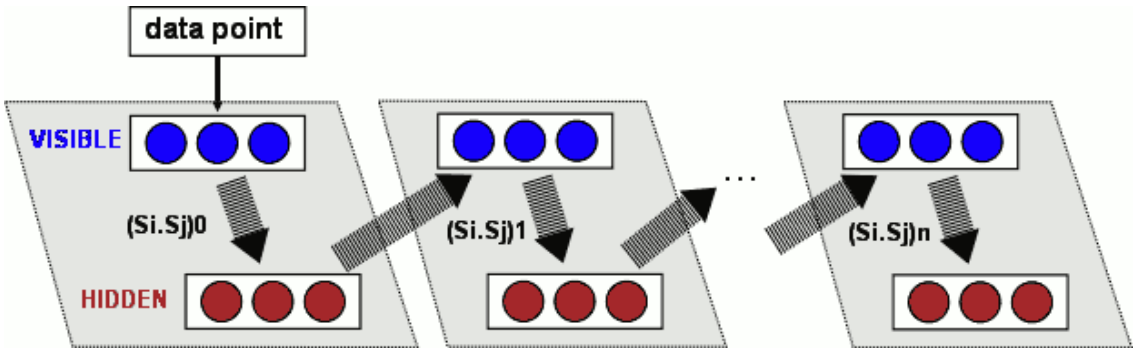
- Initialize the Markov chain with a **training example**;
- Samples are obtained after only **1-step** of Gibbs sampling.

$$v_i = x, h_i \sim P(h|v_i), v_{i+1} \sim P(v|h_i), h_{i+1} \sim P(h|v_{i+1})$$

$$\Delta_W = P(h_i|v_i)v_i' - P(h_{i+1}|v_{i+1})v_{i+1}' \quad (23)$$

$$\Delta_b = v_i - v_{i+1} \quad (24)$$

$$\Delta_c = P(h_i|v_i) - P(h_{i+1}|v_{i+1}) \quad (25)$$



Binary-Binary RBMs [Hinton, NECO 2002]

In the commonly studied case of using binary units (where v_j and $h_i \in \{0, 1\}$):

$$p(h_i = 1|v) = sigm(c_i + W_i v) \tag{26}$$

$$p(v_j = 1|h) = sigm(b_j + W'_j h) \tag{27}$$

$$F(v) = -b'v - \sum_i log(1 + e^{c_i + W_i v}) \tag{28}$$

$$-\frac{\partial log p(v)}{\partial W_{ij}} = E_v[p(h_i|v)v_j] - v_j^{(i)} sigm(W_i v^{(i)} + c_i) \tag{29}$$

$$-\frac{\partial log p(v)}{\partial c_i} = E_v[p(h_i|v)v_j] - sigm(W_i v^{(i)}) \tag{30}$$

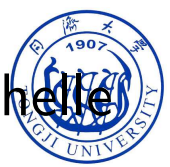
$$-\frac{\partial log p(v)}{\partial b_j} = E_v[p(h_i|v)v_j] - v_j^{(i)} \tag{31}$$

Outline

Variants of RBMs

v and h (given the other) in RBMs can be in any of the exponential family distribution. [Welling et al., NIPS 2004]

- Binary-Gaussian RBMs [Welling et al., NIPS 2004]
- Gaussian-Binary RBMs [Welling et al., NIPS 2004]
- Gaussian-Gaussian RBMs [Marks and Movellan, ICA 2001]
- Replicated Softmax [Salakhutdinov and Hinton, NIPS 2009]
- Rate-coded RBMs [Teh and Hinton, NIPS 2001]
- Lateral Connections [Osindero and Hinton, NIPS 2007]
- Conditional RBMs [Taylor and Hinton, ICML 2009; Salakhutdinov et al., ICML 2007]
- Temporal RBMs [Sutskever and Hinton, AISTATS 2007]
- Factored RBMs [Mnih and Hinton, ICML 2007]
- Sparse RBMs [Lee et al., NIPS 2007]
- Classification RBMs [Larochelle et al., JMLR 2012]



Gaussian-Binary RBMs [Welling et al., NIPS 2004]

Gaussian-Bernoulli RBMs ($v \in \mathbb{R}^I, h \in \{0,1\}^J$):

$$Energy(v,h) = \frac{1}{2}(v-b)'(v-b) - c'h - v'Wh \tag{32}$$

$$P(v|h) \propto exp(-\frac{1}{2}v'v + v'(b+Wh)) \tag{33}$$

$$P(h_j = 1|v) = sigm(c_j + v'W_{.j}) \tag{34}$$

Replicated Softmax [Salakhutdinov and Hinton, NIPS 2009]

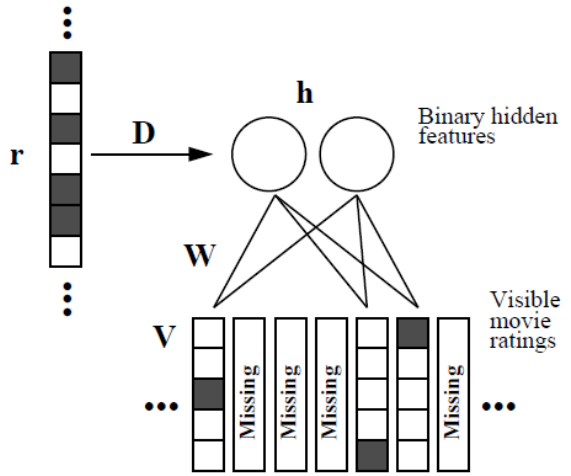
Categorical RBMs: using a 1-of-C encoding for categorical visible variables.

A **softmax** can be viewed as a set of binary units whose states are mutually constrained so that exactly one of the C states has value 1 and the rest have value 0.

The learning rule for the binary units in a softmax is identical to the rule for standard binary units.

$$p_j = \frac{e^{x_j}}{\sum_{i=1}^{C} e^{x_i}} \tag{35}$$

Conditional RBMs [Salakhutdinov et al., ICML 2007]



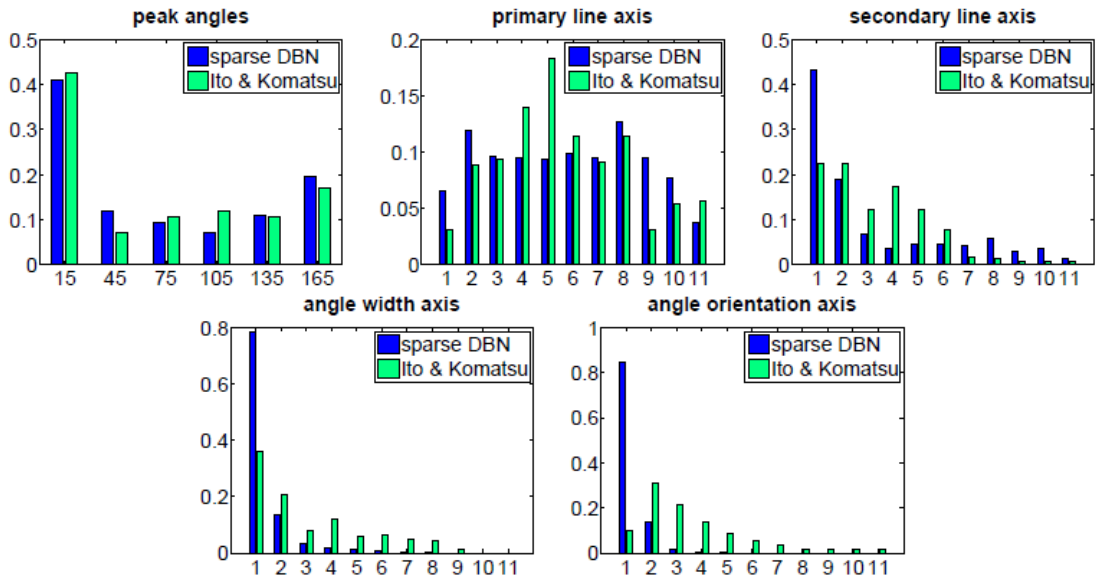
$r \in \{0,1\}^M$, indicating rated/unrated movies, affects binary states of the hidden units. D is a learned matrix that models the effect of r on h.

$$P(v_i^k = 1|h) = \frac{\exp(b_i^k + \sum_{j=1}^F h_j W_{ij}^k)}{\sum_{l=1}^K \exp(b_i^l + \sum_{j=1}^F h_j W_{ij}^l)} \tag{36}$$

$$P(h_j = 1|v,r) = sigm(b_j + \sum_{i=1}^m \sum_{k=1}^K v_i^k W_{ij}^k + \sum_{i=1}^M r_{ij} D_{ij}) \tag{37}$$

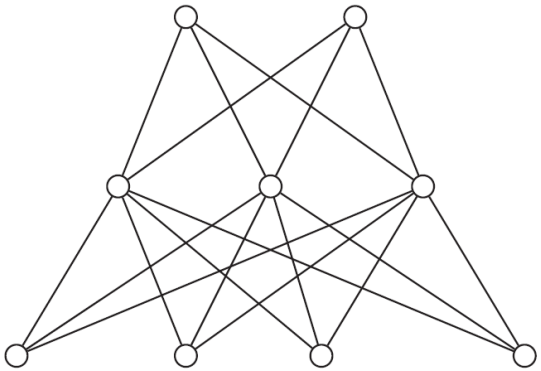
Sparse RBMs [Lee et al., NIPS 2007]

$$\min_{\{w_{ij},c_i,b_j\}} -\sum_{l=1}^{m}\log\sum_{h}P(v^{(l)},h^{(l)})+\lambda\sum_{j=1}^{n}\left|\rho-\frac{1}{m}\sum_{l=1}^{m}\mathbb{E}\left[h_j^{(l)}|v^{(l)}\right]\right|^2 \quad (38)$$

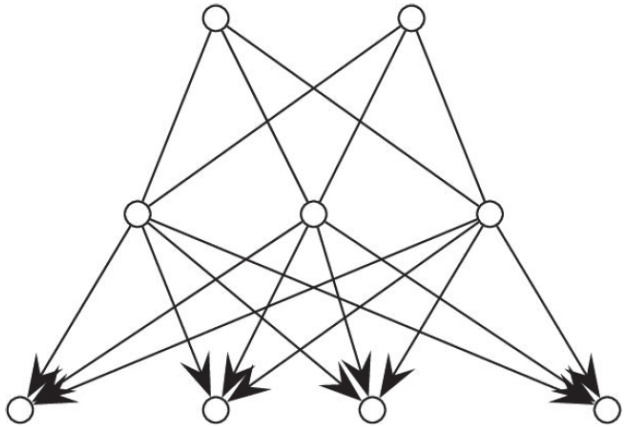


Deep Boltzmann Machines vs Deep Belief Networks

DBM [Salakhutdinov and Hinton, AISTATS 2009]:



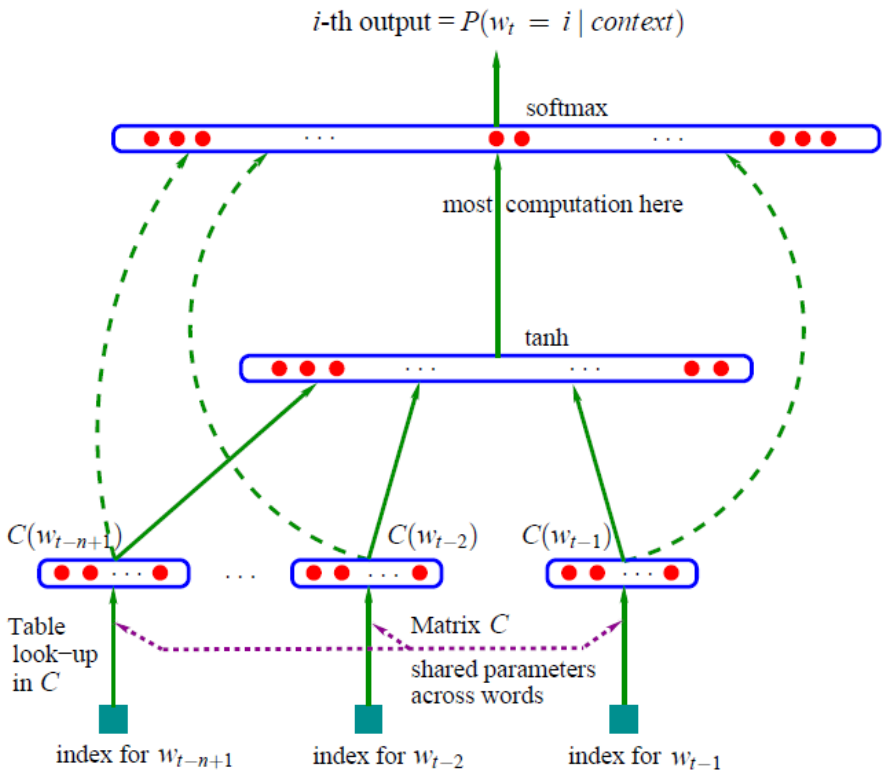
DBN [Hinton et al., NECO 2006]:



Outline

A Neural Probabilistic Language Model [Bengio et al., JMLR 2003]

Learn a distributed representation for words which learns simultaneously a distributed representation for each word along with the probability function for word sequences, expressed in terms of these representations.



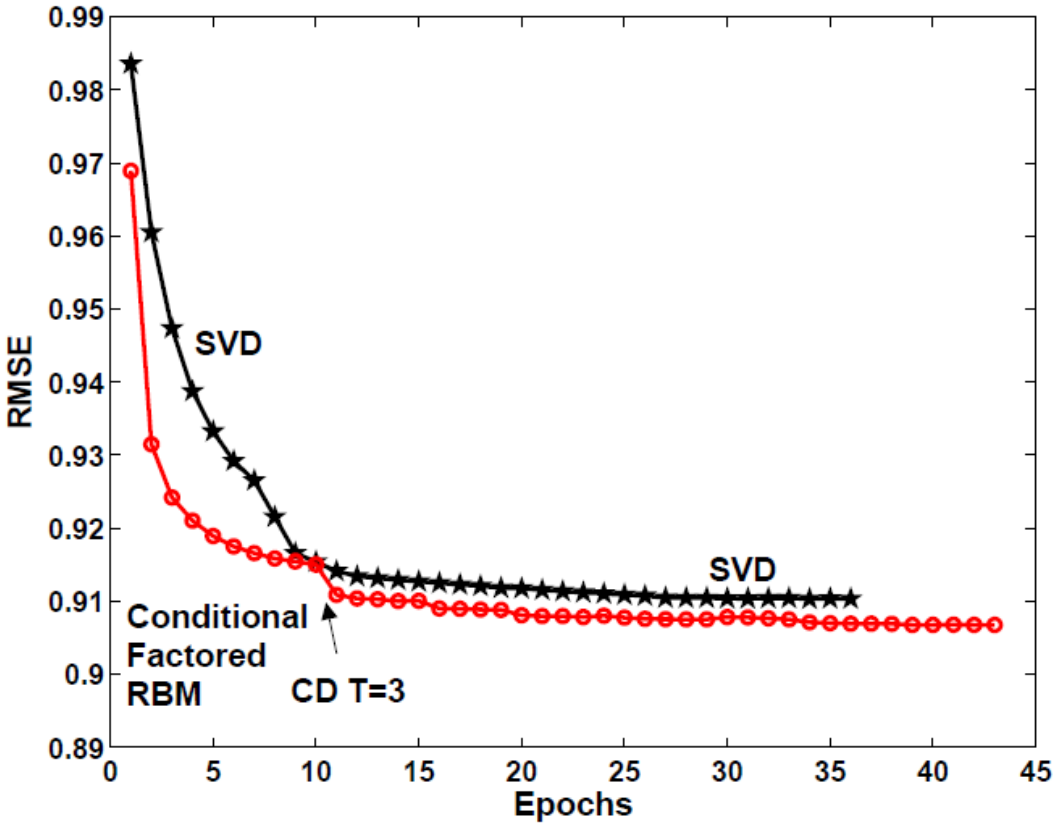
Three New Graphical Models for Statistical Language Modelling [Mnih and Hinton, ICML 2007]

Propose three new probabilistic language models (**Factored RBM**, **Temporal Factored RBM** and Log-Bilinear) that define the distribution of the next word in a sequence given several preceding words by using distributed representations of those words.

Model type	Context size	Model test score	Mixture test score
FRBM	2	169.4	110.6
Temporal FRBM	2	127.3	95.6
Log-bilinear	2	132.9	102.2
Log-bilinear	5	124.7	96.5
Back-off GT3	2	135.3	
Back-off KN3	2	124.3	
Back-off GT6	5	124.4	
Back-off KN6	5	116.2	

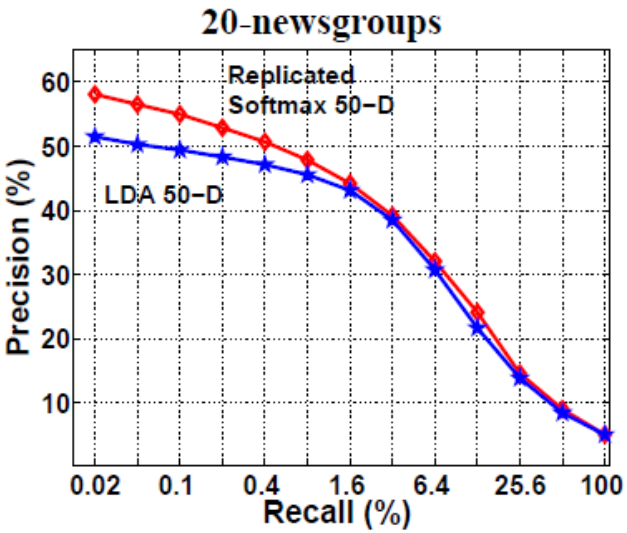
Restricted Boltzmann Machines for Collaborative Filtering [Salakhutdinov et al., ICML 2007]

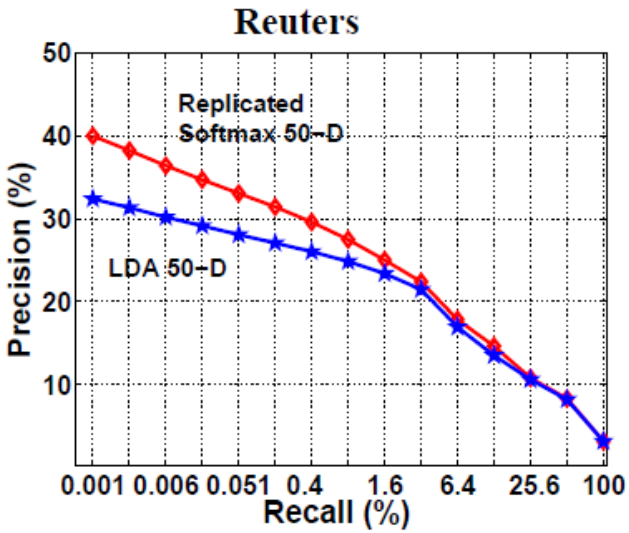
Show how RBMs can be used to model tabular data, such as user's ratings of movies and propose **Conditional Factored RBMs**.



Replicated Softmax: an Undirected Topic Model [Salakhutdinov and Hinton, NIPS 2009]

Two-layer undirected graphical model (**Replicated Softmax**): the top layer represents a vector of stochastic, binary topic features and the bottom layer represents softmax visible units. All visible units share the same set of weights, connecting them to binary hidden units.





Learning Sentence Representation for Emotion Classification on Microblogs [Tang et al., NLP&CC 2013]

- Learn the sentence representation through **Deep Belief Network** algorithm;
- Incorporate the Deep Belief Network based representation into basic features.

Method		Accuracy(%)
Text Feature	BOW	69.97
	BF	72.03
Learned Feature	PCA	70.54
	LDA	67.72
	DBN	**73.28**
Combined Feature	BF + PCA	72.46
	BF + LDA	70.19
	BF + DBN	**75.60**

Modeling Documents with a Deep Boltzmann Machine [Srivastava et al., UAI 2013]

Over-Replicated Softmax: the bottom layer represents softmax visible units, the middle layer represents binary latent topics, the top layer represents softmax hidden units. All visible and hidden softmax units share the same set of weights, connecting them to binary hidden units.

	20 News	**Reuters**
Training set size	11,072	794,414
Test set size	7,052	10,000
Vocabulary size	2,000	10,000
Avg Document Length	51.8	94.6
Perplexities		
Unigram	1335	2208
Replicated Softmax	965	1081
Over-Rep. Softmax ($M = 50$)	961	1076
Over-Rep. Softmax ($M = 100$)	**958**	**1060**

Thank you !

Q & A?

Email:tjzhifei@163.com
Weibo:@同济张_志飞