Learning Multimodal Deep Models

Russ Salakhutdinov

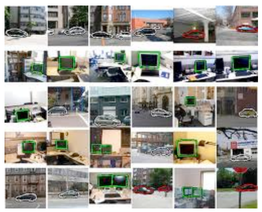
Department of Computer Science
Department of Statistics
University of Toronto
Canadian Institute for Advanced Research

Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.

Images & Video

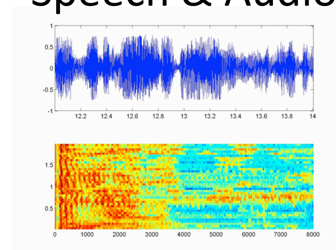
flickr Google YouTube



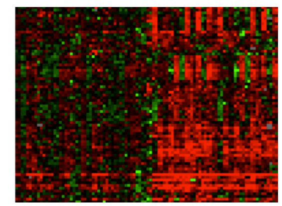
Text & Language

Wikipedia The Free Encyclopedia REUTERS Associated Press

Speech & Audio



Gene Expression

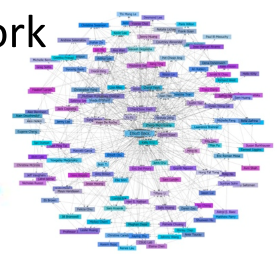


Product Recommendation

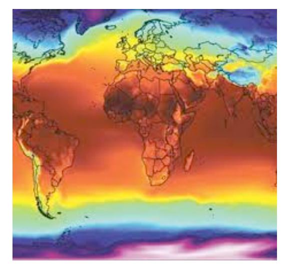
amazon NETFLIX ebay

Relational Data/ Social Network

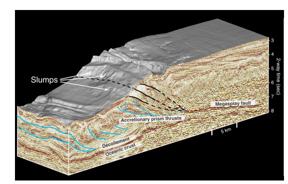
facebook twitter



Climate Change



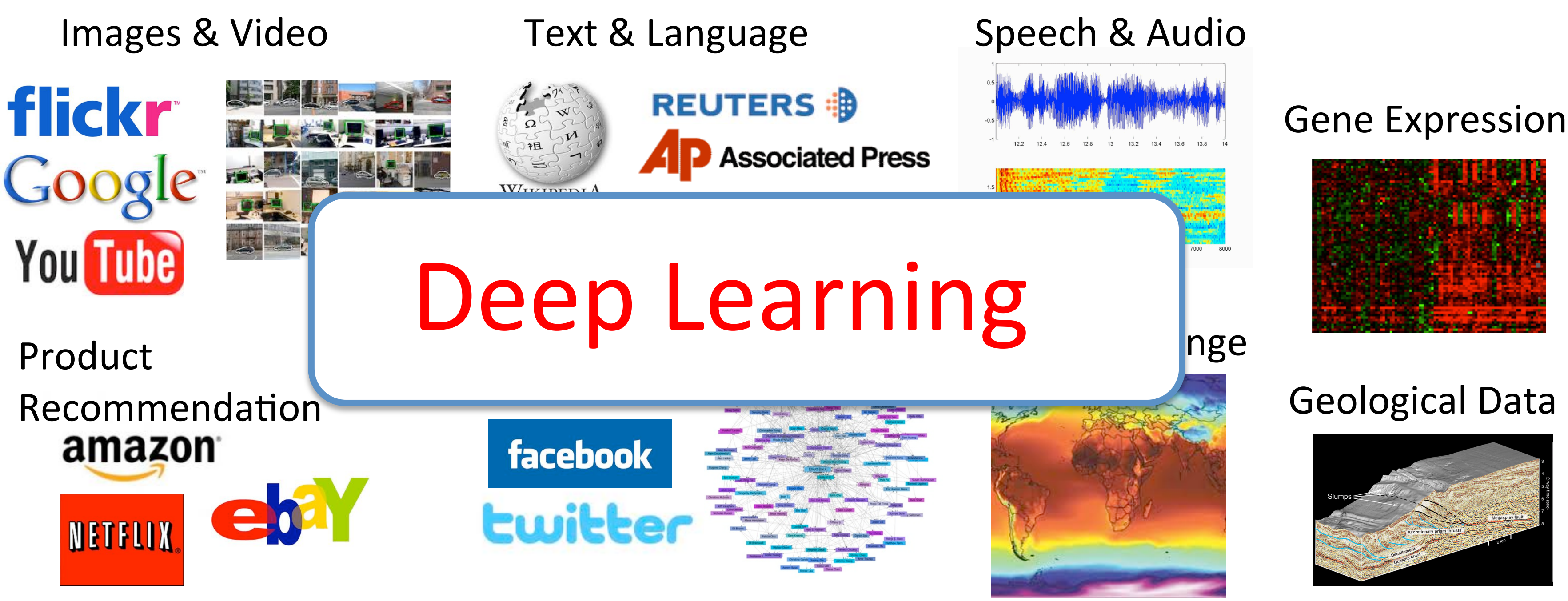
Geological Data



- Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
- Multiple application domains.

Mining for Structure

Massive increase in both computational power and the amount of data available from web, video cameras, laboratory measurements.



• Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in **unsupervised** or **semi-supervised** way.
• Multiple application domains.

Example: Understanding Images

TAGS:

strangers, coworkers, conventioneers, attendants, patrons

Nearest Neighbor Sentence:

people taking pictures of a crazy person

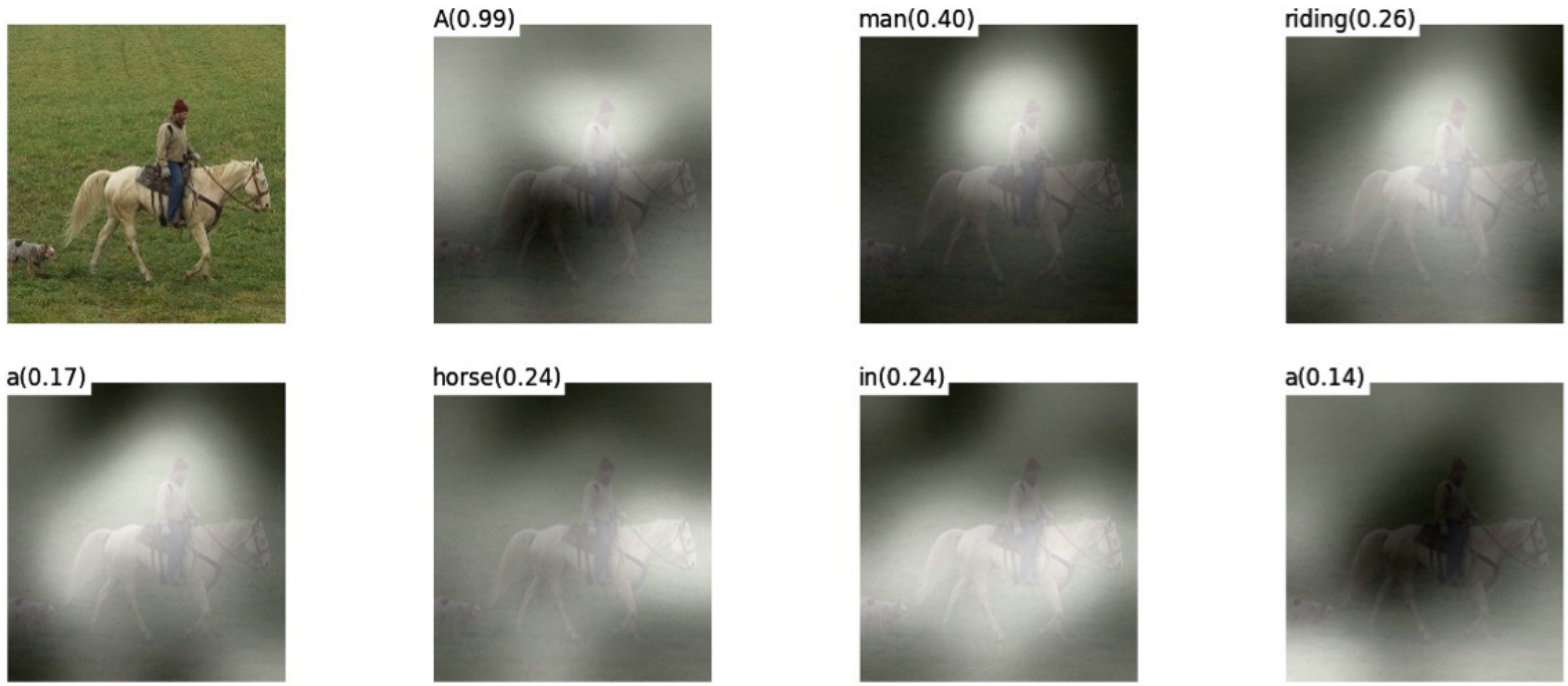
Model Samples

- a group of people in a crowded area .
- a group of people are walking and talking .
- a group of people, standing around and talking .
- a group of people that are in the outside .

Caption Generation with Visual Attention

A man riding a horse
in a field.

Caption Generation with Visual Attention



A man riding a horse
in a field.

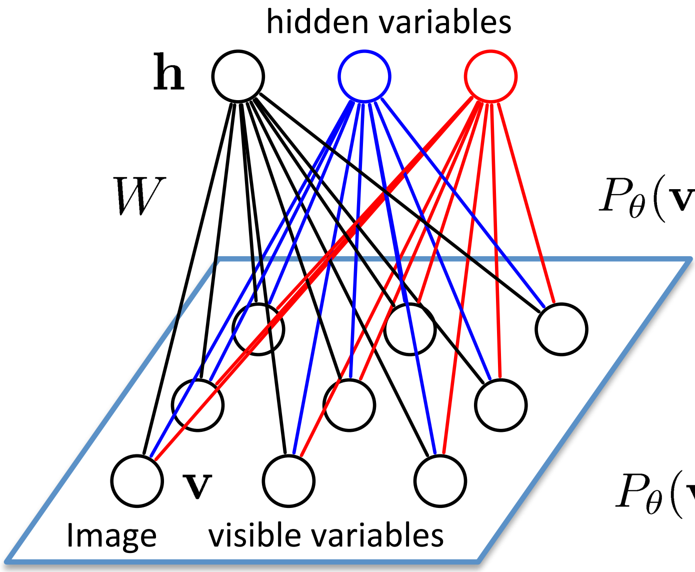
Xu et.al., ICML 2015

Talk Roadmap

- Learning Deep Models
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines

- Multi-Modal Learning

Restricted Boltzmann Machines



$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\underbrace{\sum_{i=1}^{D}\sum_{j=1}^{F} W_{ij} v_i h_j}_{\textbf{Pair-wise}} + \underbrace{\sum_{i=1}^{D} v_i b_i + \sum_{j=1}^{F} h_j a_j}_{\textbf{Unary}} \right)$$

$$\theta = \{W, a, b\}$$

$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \frac{1}{1+\exp(-\sum_{j=1}^{F} W_{ij} v_i h_j - b_i)}$$

RBM is a Markov Random Field with:

- Stochastic binary visible variables $\mathbf{v} \in \{0,1\}^D$.
- Stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.
- Bipartite connections.

Markov random fields, Boltzmann machines, log-linear models.

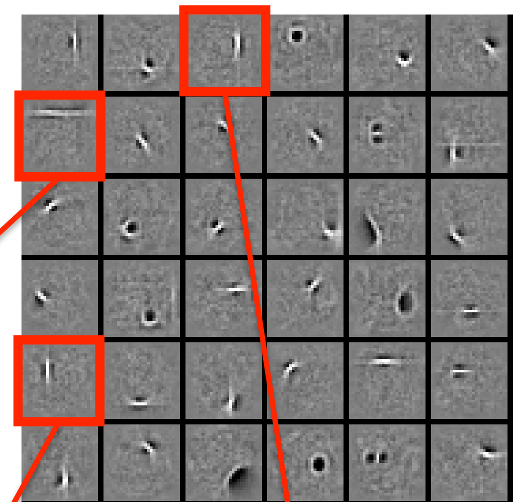
Learning Features

Observed Data

Subset of 25,000 characters

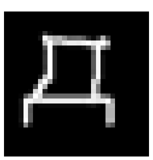
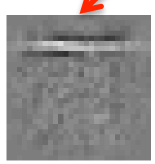
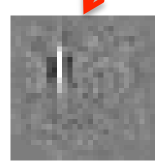
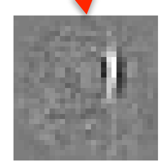
Learned W: “edges”

Subset of 1000 features



Sparse representations

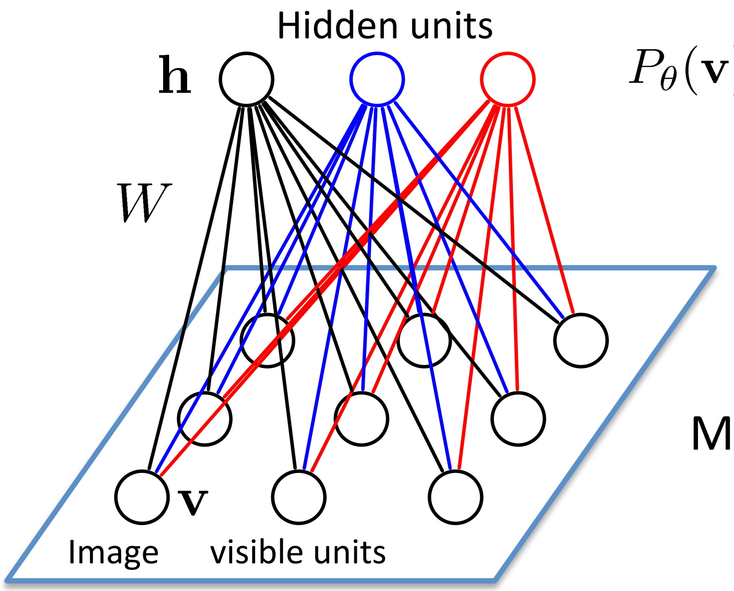
New Image: $p(h_7 = 1|v)$ $p(h_{29} = 1|v)$

 $= \sigma\Big(0.99 \times$  $+\ 0.97 \times$  $+\ 0.82 \times$  $\cdots\Big)$

$$\sigma(x) = \frac{1}{1+\exp(-x)}$$

Logistic Function: Suitable for modeling binary images

Model Learning



$$P_\theta(\mathbf{v}) = \frac{P^*(\mathbf{v})}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{\mathbf{h}} \exp\left[\mathbf{v}^\top W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}\right]$$

Given a set of *i.i.d.* training examples $\mathcal{D} = \{\mathbf{v}^{(1)}, \mathbf{v}^{(2)}, ..., \mathbf{v}^{(N)}\}$, we want to learn model parameters $\theta = \{W, a, b\}$.

Maximize log-likelihood objective:

$$L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(\mathbf{v}^{(n)})$$

Derivative of the log-likelihood:

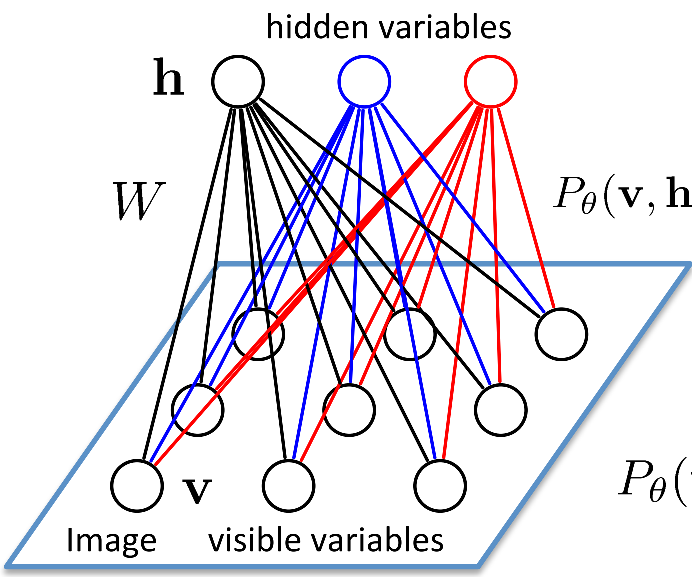
$$\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log\left(\sum_{\mathbf{h}} \exp\left[\mathbf{v}^{(n)\top} W \mathbf{h} + \mathbf{a}^\top \mathbf{h} + \mathbf{b}^\top \mathbf{v}^{(n)}\right]\right) - \frac{\partial}{\partial W_{ij}} \log \mathcal{Z}(\theta)$$

$$= \mathrm{E}_{P_{data}}[v_i h_j] - \underbrace{\mathrm{E}_{P_\theta}[v_i h_j]}_{\text{Difficult to compute: exponentially many configurations}}$$

$$P_{data}(\mathbf{v}, \mathbf{h}; \theta) = P(\mathbf{h}|\mathbf{v}; \theta) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N} \sum_{n} \delta(\mathbf{v} - \mathbf{v}^{(n)})$$

RBMs for Real-valued Data

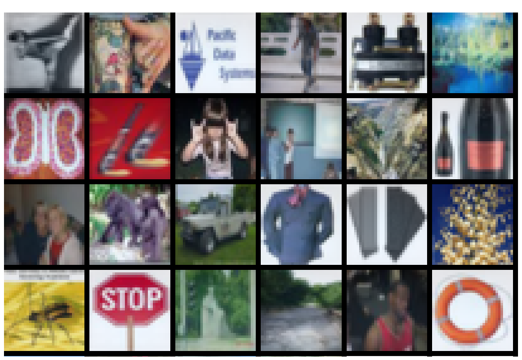


$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\sum_{i=1}^{D}\sum_{j=1}^{F} \overbrace{W_{ij} h_j \frac{v_i}{\sigma_i}}^{\text{Pair-wise}} + \sum_{i=1}^{D} \overbrace{\frac{(v_i - b_i)^2}{2\sigma_i^2}}^{\text{Unary}} + \sum_{j=1}^{F} \overbrace{a_j h_j} \right)$$

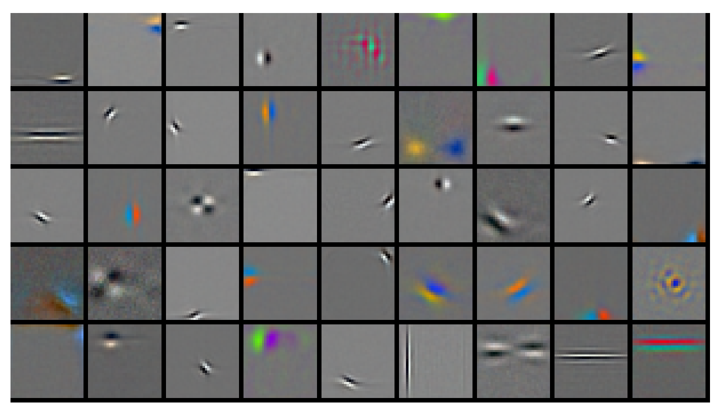
$$\theta = \{W, a, b\}$$

$$P_\theta(\mathbf{v}|\mathbf{h}) = \prod_{i=1}^{D} P_\theta(v_i|\mathbf{h}) = \prod_{i=1}^{D} \mathcal{N}\left(b_i + \sum_{j=1}^{F} W_{ij} h_j, \sigma_i^2 \right)$$

4 million **unlabelled** images

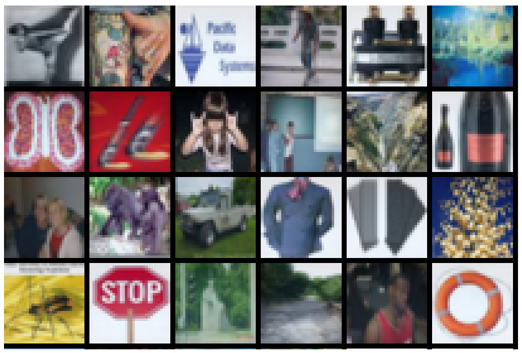


Learned features (out of 10,000)

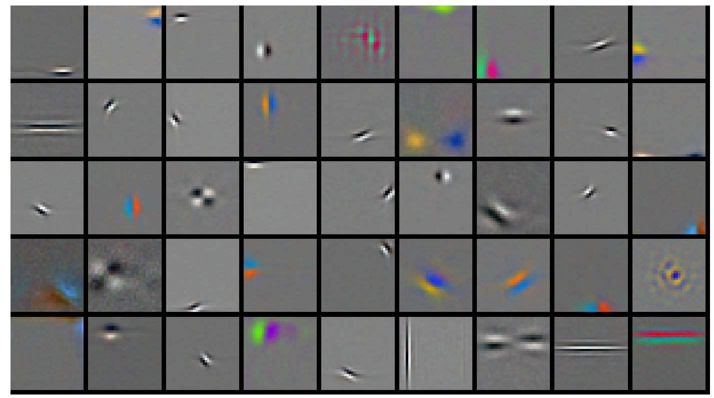


RBMs for Real-valued Data

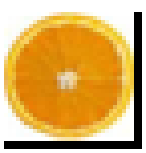
4 million **unlabelled** images



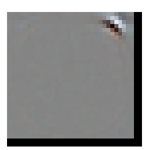
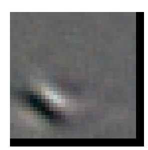
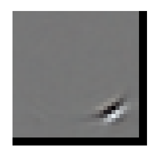
Learned features (out of 10,000)



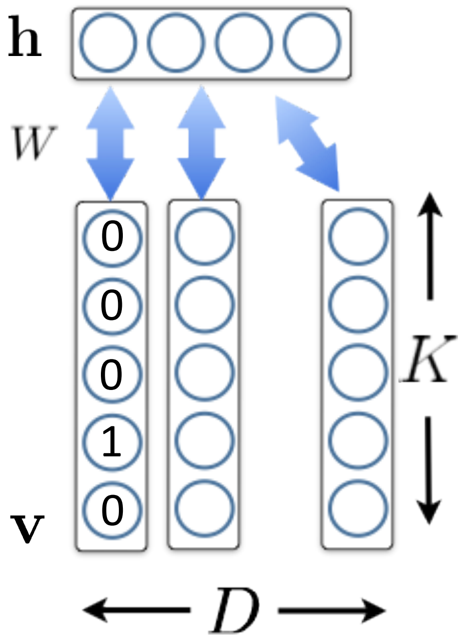
$p(h_7 = 1|v)$ $\quad$ $p(h_{29} = 1|v)$



New Image

$=\ 0.9\ *$  $+\ 0.8\ *$  $+\ 0.6\ *$  $\ldots$

RBMs for Word Counts



$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\overbrace{\sum_{i=1}^{D}\sum_{k=1}^{K}\sum_{j=1}^{F}}^{\textbf{Pair-wise}} W_{ij}^{k} v_i^k h_j + \overbrace{\sum_{i=1}^{D}\sum_{k=1}^{K}}^{\textbf{Unary}} v_i^k b_i^k + \overbrace{\sum_{j=1}^{F}}^{\textbf{Unary}} h_j a_j \right)$$

$$\theta = \{W, a, b\}$$

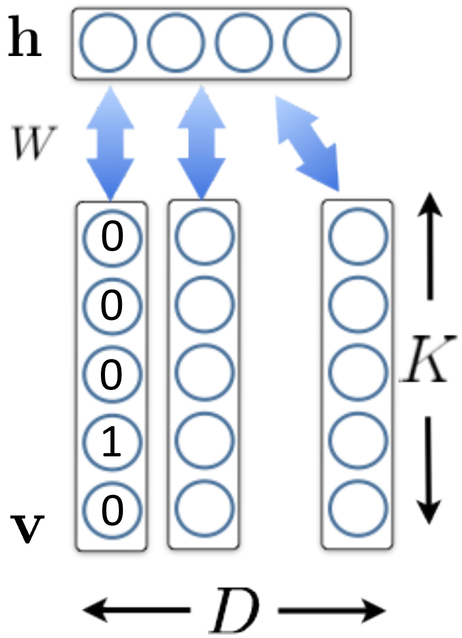
$$P_\theta(v_i^k = 1|\mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k\right)}{\sum_{q=1}^{K} \exp\left(b_i^q + \sum_{j=1}^{F} h_j W_{ij}^q\right)}$$

Replicated Softmax Model: undirected topic model:

- Stochastic 1-of-K visible variables.
- Stochastic binary hidden variables $\mathbf{h} \in \{0,1\}^F$.
- Bipartite connections.

(Salakhutdinov & Hinton, NIPS 2010, Srivastava & Salakhutdinov, NIPS 2012)

RBMs for Word Counts



$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left(\overbrace{\sum_{i=1}^{D}\sum_{k=1}^{K}\sum_{j=1}^{F}}^{\text{Pair-wise}} W_{ij}^{k} v_i^k h_j + \overbrace{\sum_{i=1}^{D}\sum_{k=1}^{K}}^{\text{Unary}} v_i^k b_i^k + \overbrace{\sum_{j=1}^{F}}^{} h_j a_j \right)$$

$$\theta = \{W, a, b\}$$

$$P_\theta(v_i^k = 1|\mathbf{h}) = \frac{\exp\left(b_i^k + \sum_{j=1}^{F} h_j W_{ij}^k\right)}{\sum_{q=1}^{K} \exp\left(b_i^q + \sum_{j=1}^{F} h_j W_{ij}^q\right)}$$

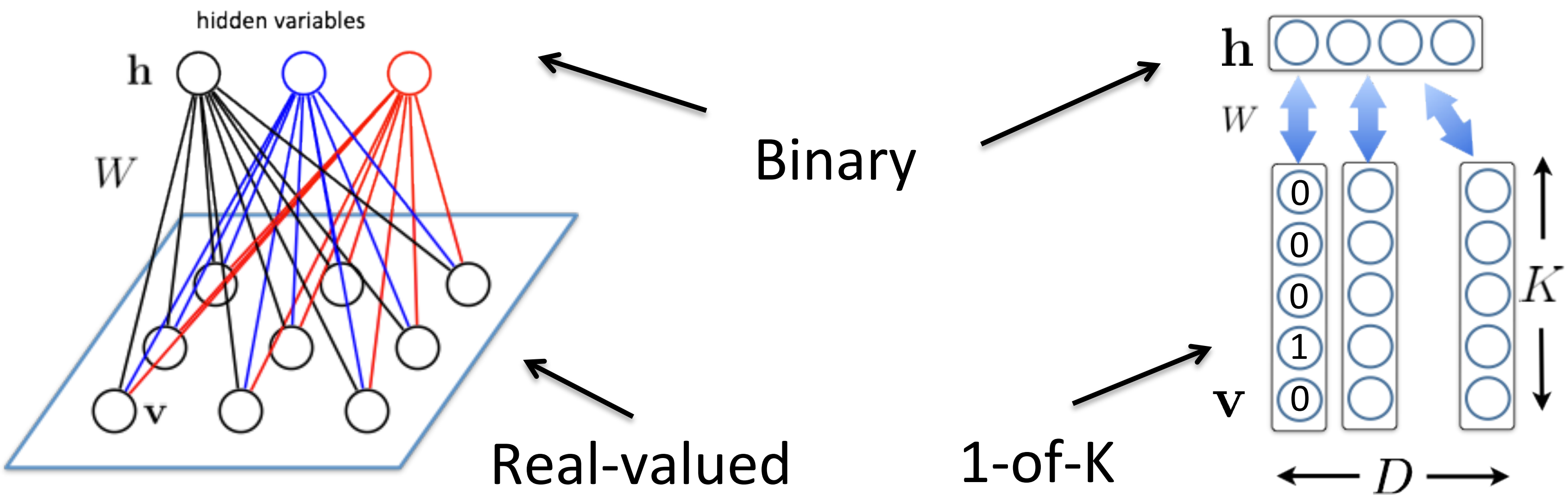
Reuters dataset:
804,414 **unlabeled** newswire stories
Bag-of-Words

Learned features: ``topics''

russian	clinton	computer	trade	stock
russia	house	system	country	wall
moscow	president	product	import	street
yeltsin	bill	software	world	point
soviet	congress	develop	economy	dow

Different Data Modalities

- Binary/Gaussian/Softmax RBMs: All have binary hidden variables but use them to model different kinds of data.



- It is easy to infer the states of the hidden variables:

$$P_\theta(\mathbf{h}|\mathbf{v}) = \prod_{j=1}^{F} P_\theta(h_j|\mathbf{v}) = \prod_{j=1}^{F} \frac{1}{1+\exp(-a_j - \sum_{i=1}^{D} W_{ij}v_i)}$$

Product of Experts

The joint distribution is given by:

$$P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\Big(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\Big)$$

Marginalizing over hidden variables:

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}} P_\theta(\mathbf{v},\mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \prod_i \exp(b_i v_i) \prod_j \overbrace{\Big(1 + \exp(a_j + \sum_i W_{ij} v_i)\Big)}^{\textbf{Product of Experts}}$$

government	clinton	bribery	oil	stock	…
auhority	house	corruption	barrel	wall	
power	president	dishonesty	exxon	street	
empire	bill	putin	putin	point	
putin	congress	fraud	drill	dow	

Putin

Topics “government”, ”corruption” and ”oil” can combine to give very high probability to a word “Putin”.

(Srivastava & Salakhutdinov, NIPS 2012)

Product of Experts

The joint distribution is given by:

$$P_\theta(\mathbf{v}, \mathbf{h}) = \frac{1}{\mathcal{Z}(\theta)} \exp\Big(\sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j\Big)$$

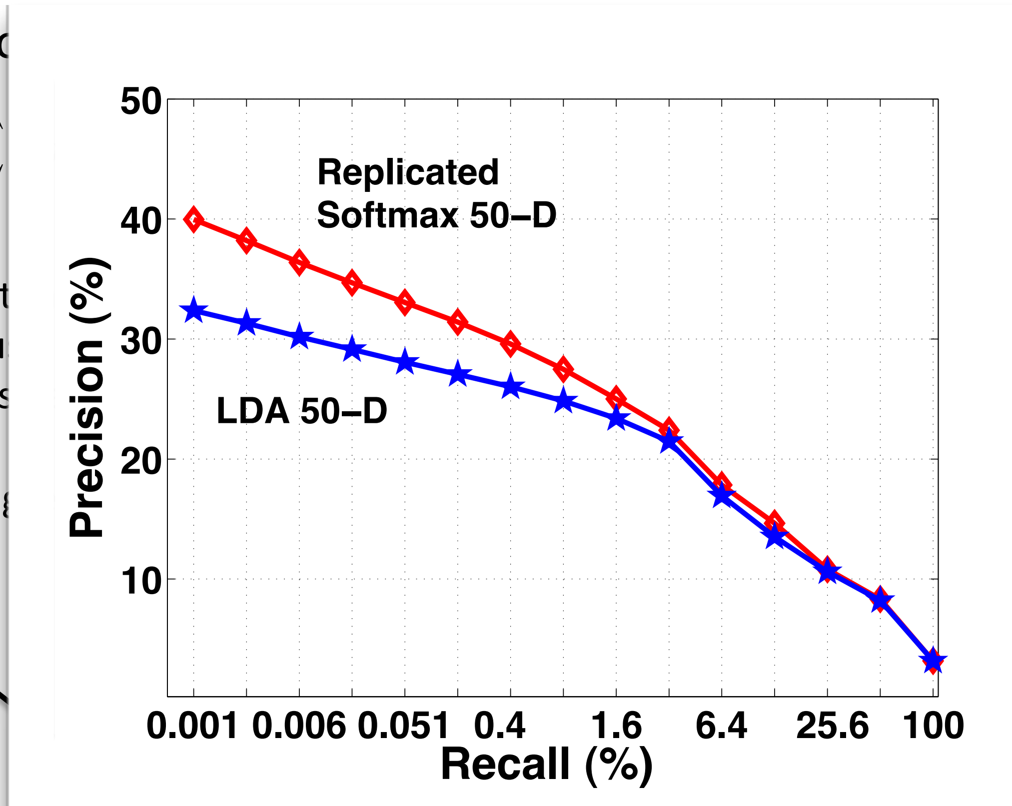
Marginalizing c

$$P_\theta(\mathbf{v}) = \sum_{\mathbf{h}}$$

duct of Experts

$$W_{ij} v_i)\Big)$$

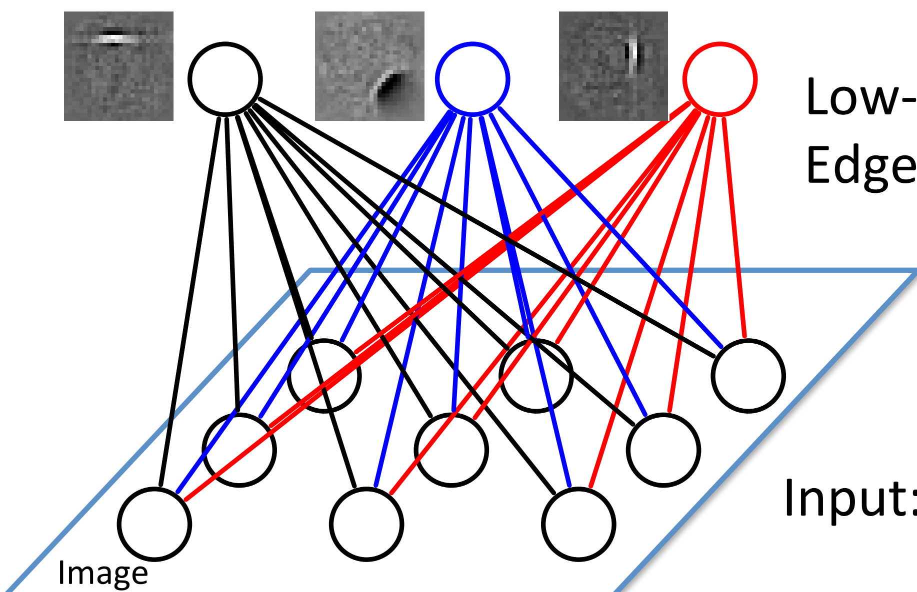
government	clint
auhority	hou
power	pres
empire	bill
putin	cong



", "corruption"
e to give very high
"Putin".

(Srivastava & Salakhutdinov, NIPS 2012)

Deep Boltzmann Machines



Low-level features:
Edges

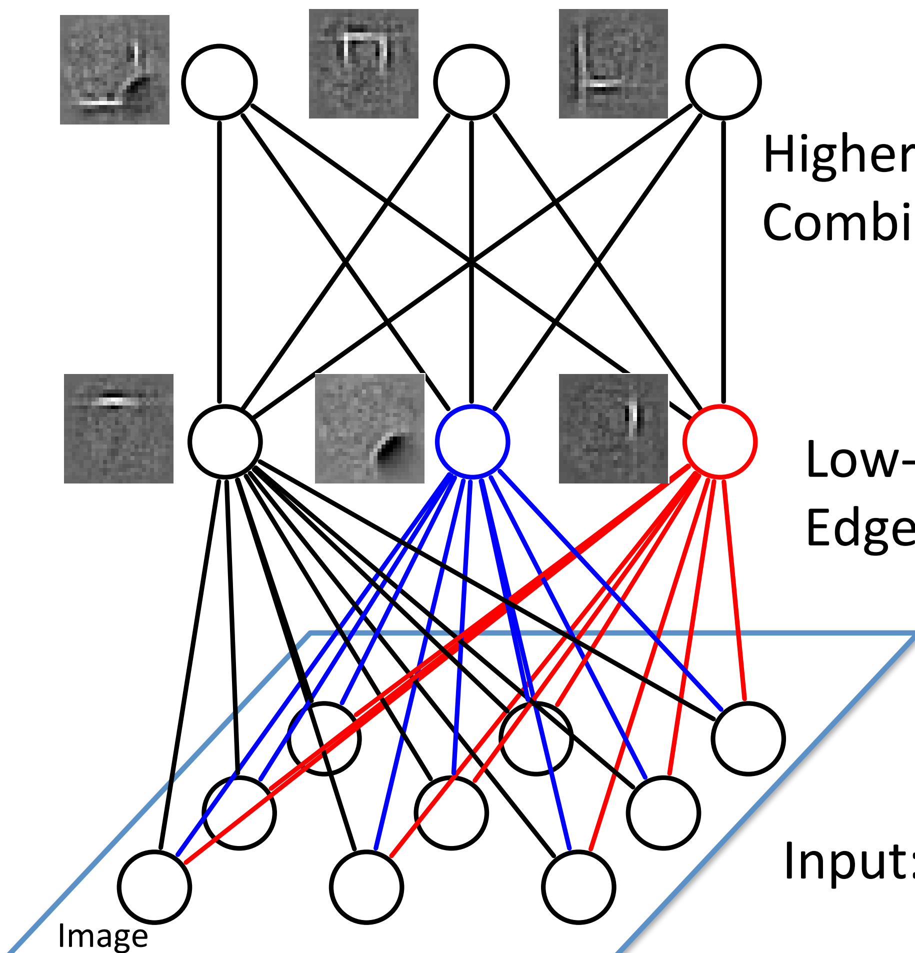
Built from **unlabeled** inputs.

Input: Pixels

(Salakhutdinov & Hinton, Neural Computation 2012)

Deep Boltzmann Machines

Learn simpler representations,
then compose more complex ones



Higher-level features:
Combination of edges

Low-level features:
Edges

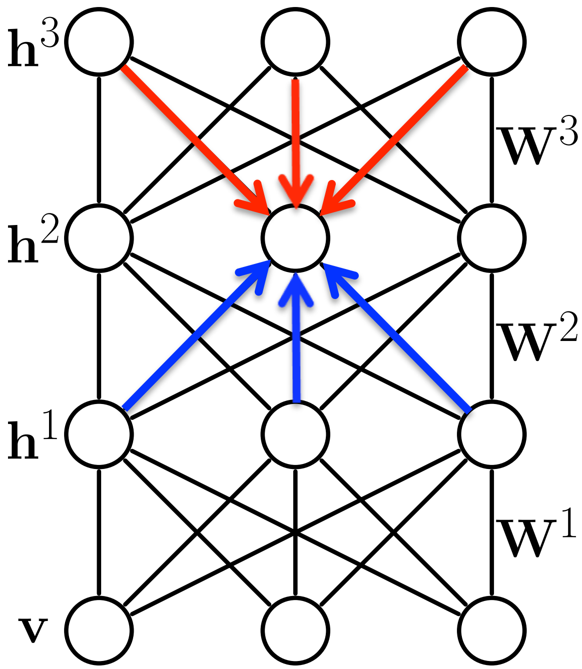
Built from **unlabeled** inputs.

Input: Pixels

(Salakhutdinov 2008, Salakhutdinov & Hinton 2012)

Model Formulation

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\underbrace{\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)}}_{\textbf{Same as RBMs}} + \underline{\mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)}} + \underline{\mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}}\right]$$



Input

$\theta = \{W^1, W^2, W^3\}$ model parameters

- Dependencies between hidden variables.
- All connections are undirected.
- Bottom-up and Top-down:

$$P(h_j^2 = 1|\mathbf{h}^1, \mathbf{h}^3) = \sigma\Big(\sum_k W_{kj}^3 h_k^3 + \sum_m W_{mj}^2 h_m^1\Big)$$

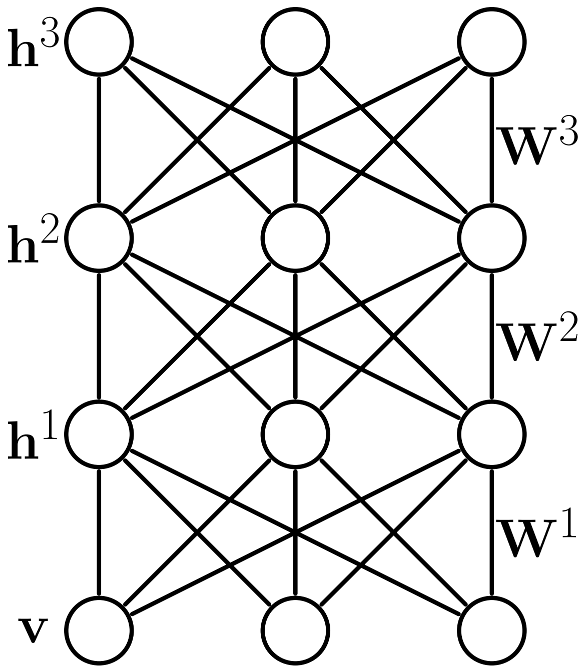
Top-down

Bottom-up

- Hidden variables are dependent even when **conditioned on the input**.

Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)} \mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)} \mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)} \mathbf{h}^{(3)}\right]$$



(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h^1}^\top] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h^1}^\top]$$

- Both expectations are intractable!

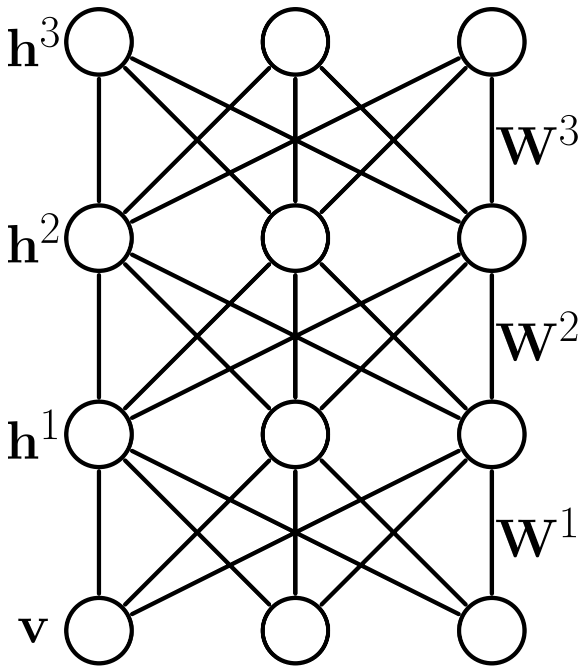
$$P_{data}(\mathbf{v}, \mathbf{h^1}) = P_\theta(\mathbf{h^1}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N} \sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

Not factorial any more!

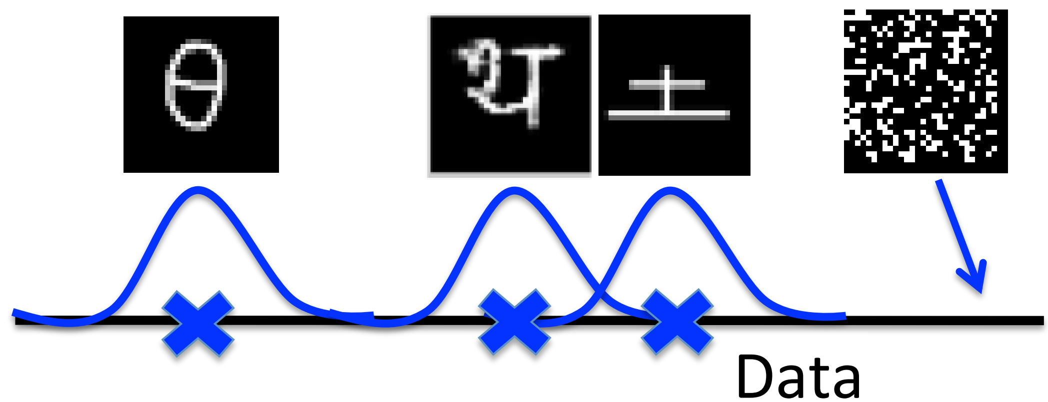
Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)} \mathbf{h}^{(1)} + {\mathbf{h}^{(1)}}^\top W^{(2)} \mathbf{h}^{(2)} + {\mathbf{h}^{(2)}}^\top W^{(3)} \mathbf{h}^{(3)}\right]$$



(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}{\mathbf{h}^{\mathbf{1}}}^\top] - \mathbb{E}_{P_\theta}[\mathbf{v}{\mathbf{h}^{\mathbf{1}}}^\top]$$



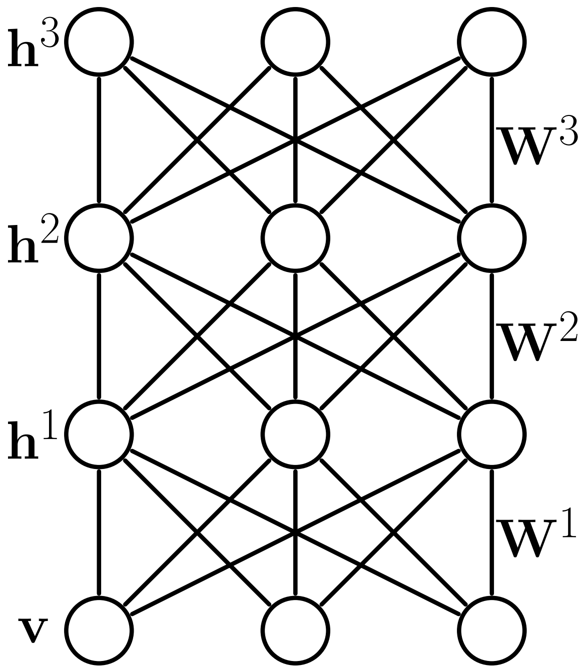
$$P_{data}(\mathbf{v}, \mathbf{h}^{\mathbf{1}}) = P_\theta(\mathbf{h}^{\mathbf{1}}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

Not factorial any more!

Approximate Learning

$$P_\theta(\mathbf{v}, \mathbf{h}^{(1)}, \mathbf{h}^{(2)}, \mathbf{h}^{(3)}) = \frac{1}{\mathcal{Z}(\theta)} \exp\left[\mathbf{v}^\top W^{(1)}\mathbf{h}^{(1)} + \mathbf{h}^{(1)\top} W^{(2)}\mathbf{h}^{(2)} + \mathbf{h}^{(2)\top} W^{(3)}\mathbf{h}^{(3)}\right]$$



(Approximate) Maximum Likelihood:

$$\frac{\partial \log P_\theta(\mathbf{v})}{\partial W^1} = \mathbb{E}_{P_{data}}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}] - \mathbb{E}_{P_\theta}[\mathbf{v}\mathbf{h}^{\mathbf{1}\top}]$$

Variational Inference

Stochastic Approximation (MCMC-based)

$$P_{data}(\mathbf{v}, \mathbf{h}^{\mathbf{1}}) = P_\theta(\mathbf{h}^{\mathbf{1}}|\mathbf{v}) P_{data}(\mathbf{v})$$

$$P_{data}(\mathbf{v}) = \frac{1}{N}\sum_{n=1}^{N} \delta(\mathbf{v} - \mathbf{v_n})$$

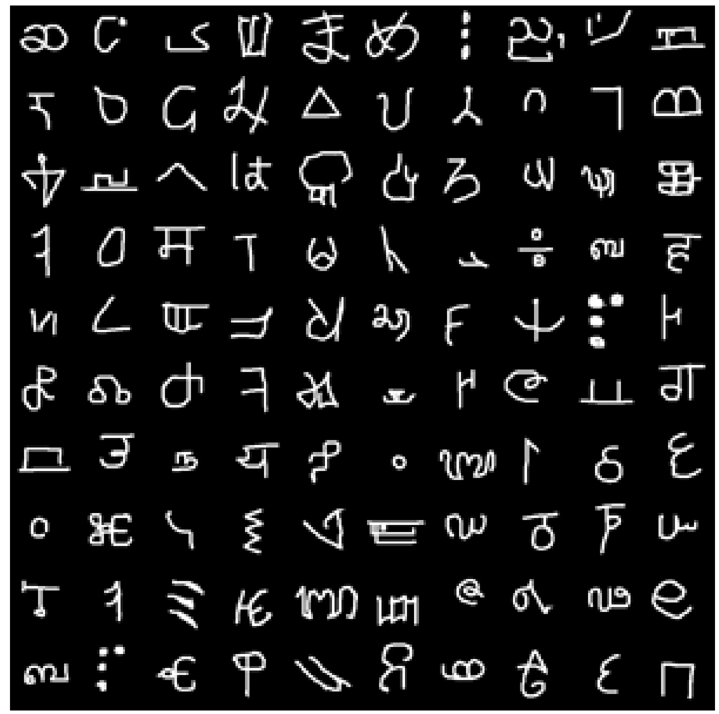
Not factorial any more!

Good Generative Model?

Handwritten Characters

Good Generative Model?

Handwritten Characters



Good Generative Model?

Handwritten Characters

Simulated

Real Data

Good Generative Model?

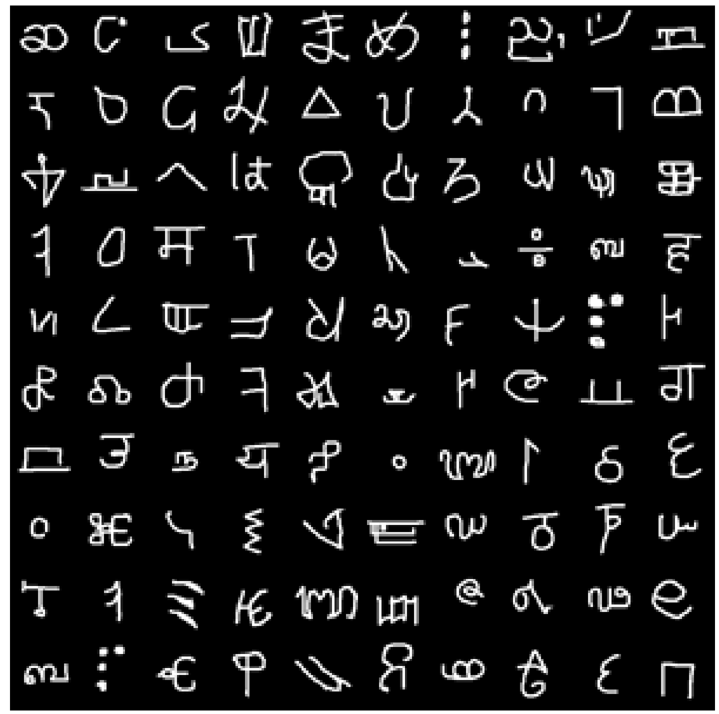
Handwritten Characters

Real Data

Simulated

Good Generative Model?

Handwritten Characters



Handwriting Recognition

MNIST Dataset
60,000 examples of 10 digits

Learning Algorithm	Error
Logistic regression	12.0%
K-NN	3.09%
Neural Net (Platt 2005)	1.53%
SVM (Decoste et.al. 2002)	1.40%
Deep Autoencoder (Bengio et. al. 2007)	1.40%
Deep Belief Net (Hinton et. al. 2006)	1.20%
DBM	**0.95%**

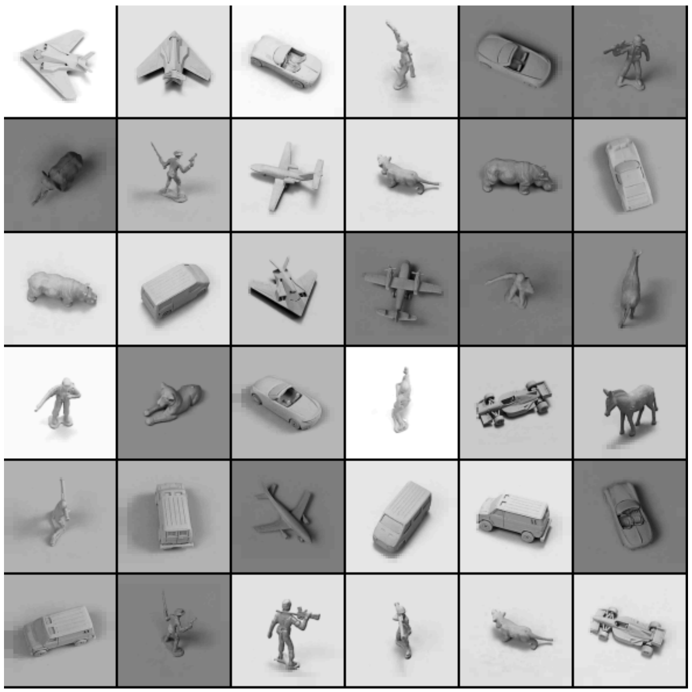
Optical Character Recognition
42,152 examples of 26 English letters

Learning Algorithm	Error
Logistic regression	22.14%
K-NN	18.92%
Neural Net	14.62%
SVM (Larochelle et.al. 2009)	9.70%
Deep Autoencoder (Bengio et. al. 2007)	10.05%
Deep Belief Net (Larochelle et. al. 2009)	9.68%
DBM	**8.40%**

Permutation-invariant version.

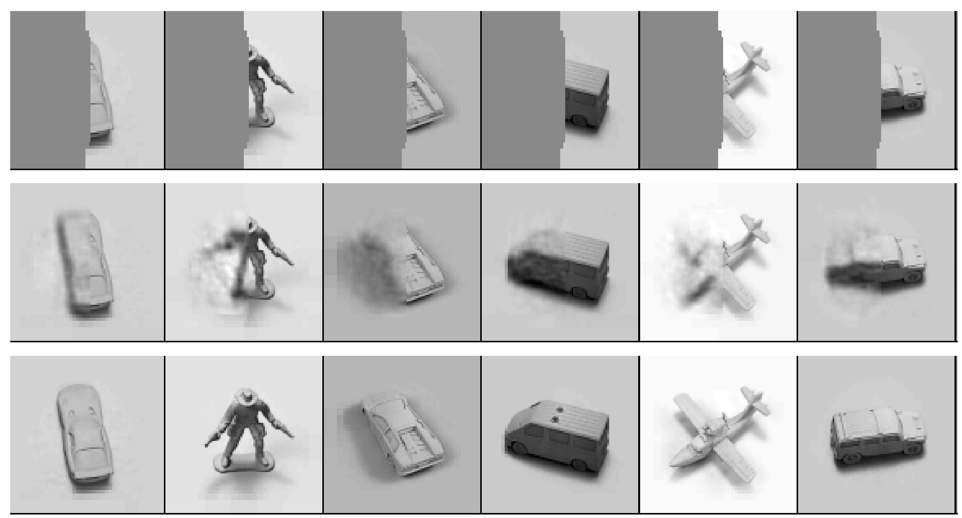
3-D object Recognition

NORB Dataset: 24,000 examples



Learning Algorithm	Error
Logistic regression	22.5%
K-NN (LeCun 2004)	18.92%
SVM (Bengio & LeCun 2007)	11.6%
Deep Belief Net (Nair & Hinton 2009)	9.0%
DBM	**7.2%**

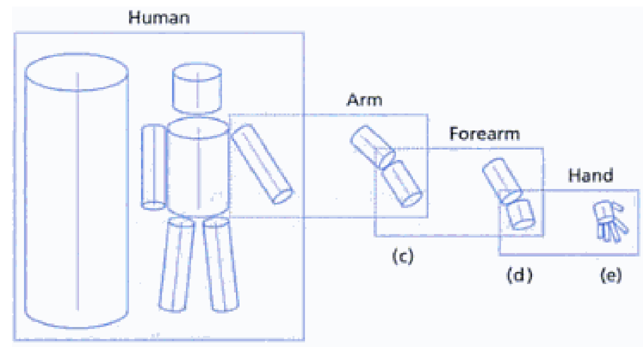
Pattern Completion



Learning Hierarchical Representations

Deep Boltzmann Machines:

Learning Hierarchical Structure in Features: edges, combination of edges.



- Performs well in many application domains
- Fast Inference: fraction of a second
- Learning scales to millions of examples

Talk Roadmap

- Learning Deep Models
  - Restricted Boltzmann Machines
  - Deep Boltzmann Machines

- Multi-Modal Learning

Data – Collection of Modalities

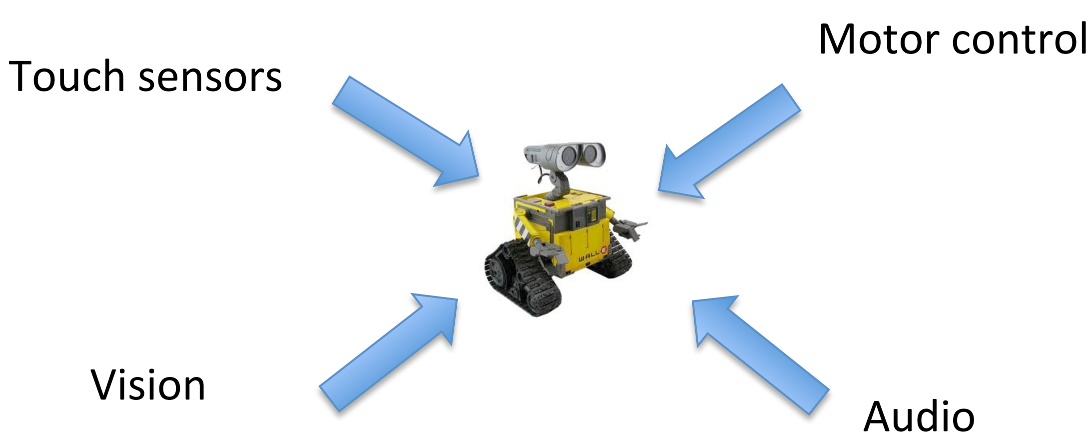
- Multimedia content on the web - image + text + audio.

- Product recommendation systems.

- Robotics applications.

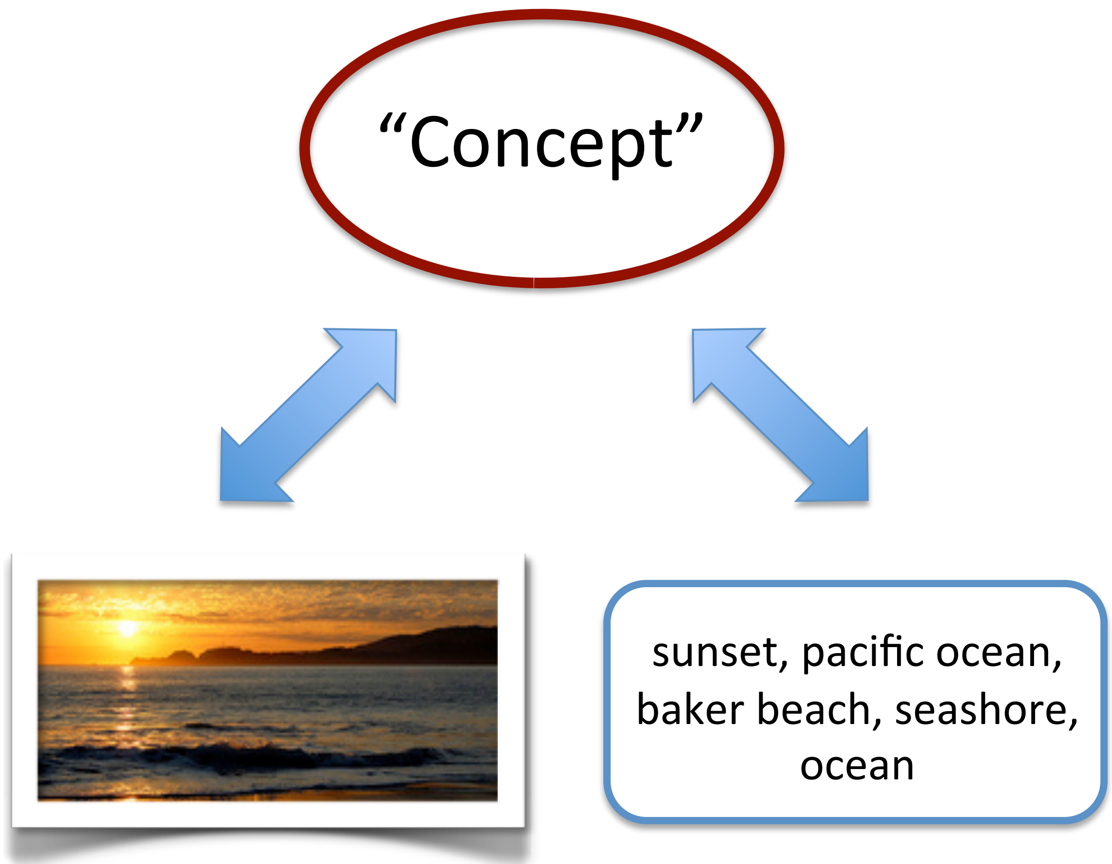

car,
automobile


sunset,
pacificocean,
bakerbeach,
seashore, ocean



Shared Concept

"Modality-free" representation



"Modality-full" representation

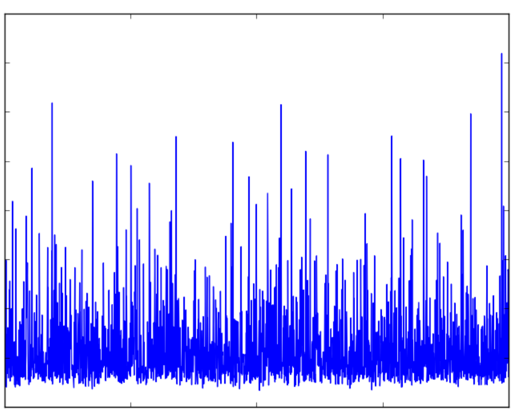
Challenges - I

Image

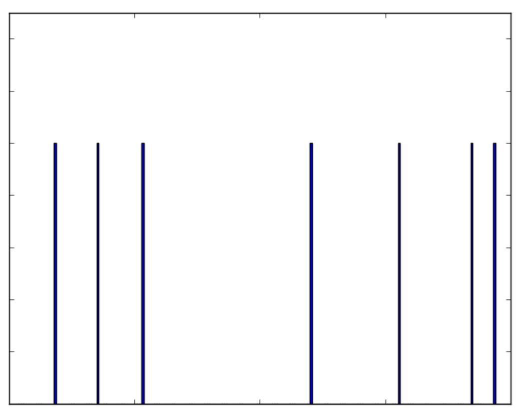
Text

sunset, pacific ocean, baker beach, seashore, ocean

Dense



Sparse



Very different input representations

- Images – real-valued, dense
- Text – discrete, sparse

Difficult to learn cross-modal features from low-level representations.

Challenges - II

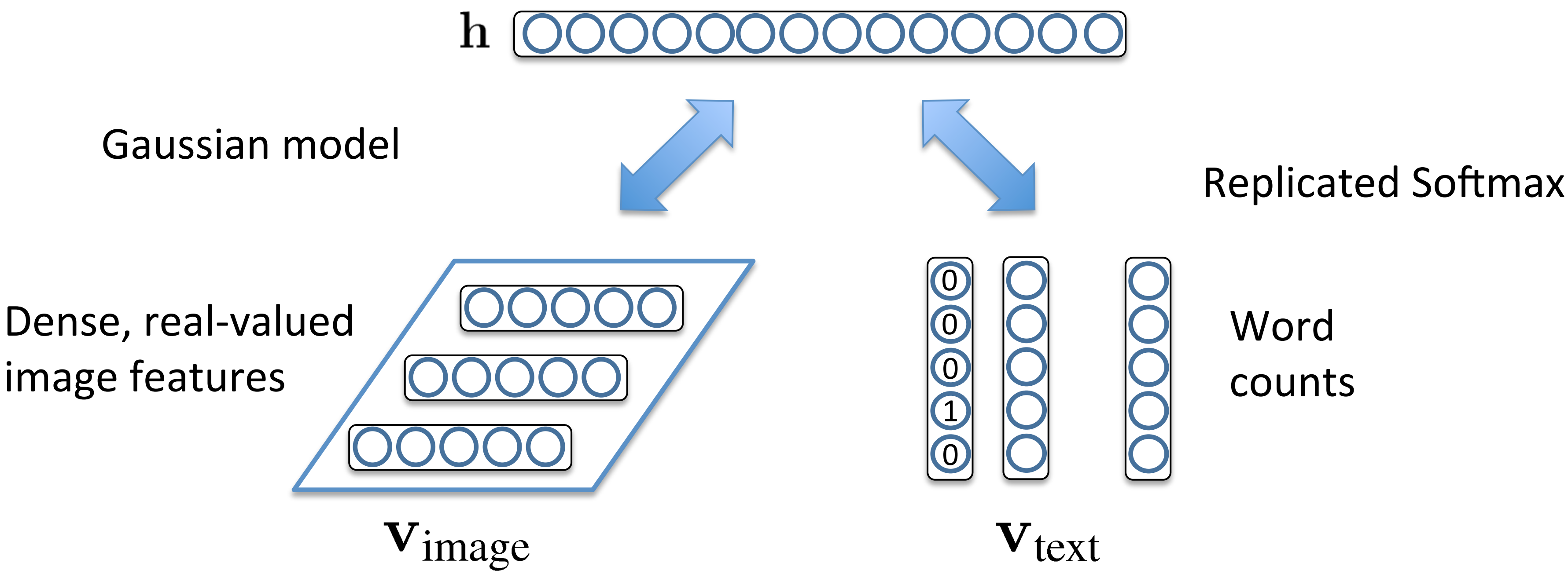
Image	Tags
	pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion
	mickikrimmel, mickipedia, headshot
	< no text>
	unseulpixel, naturey

Noisy and missing data

Challenges - II

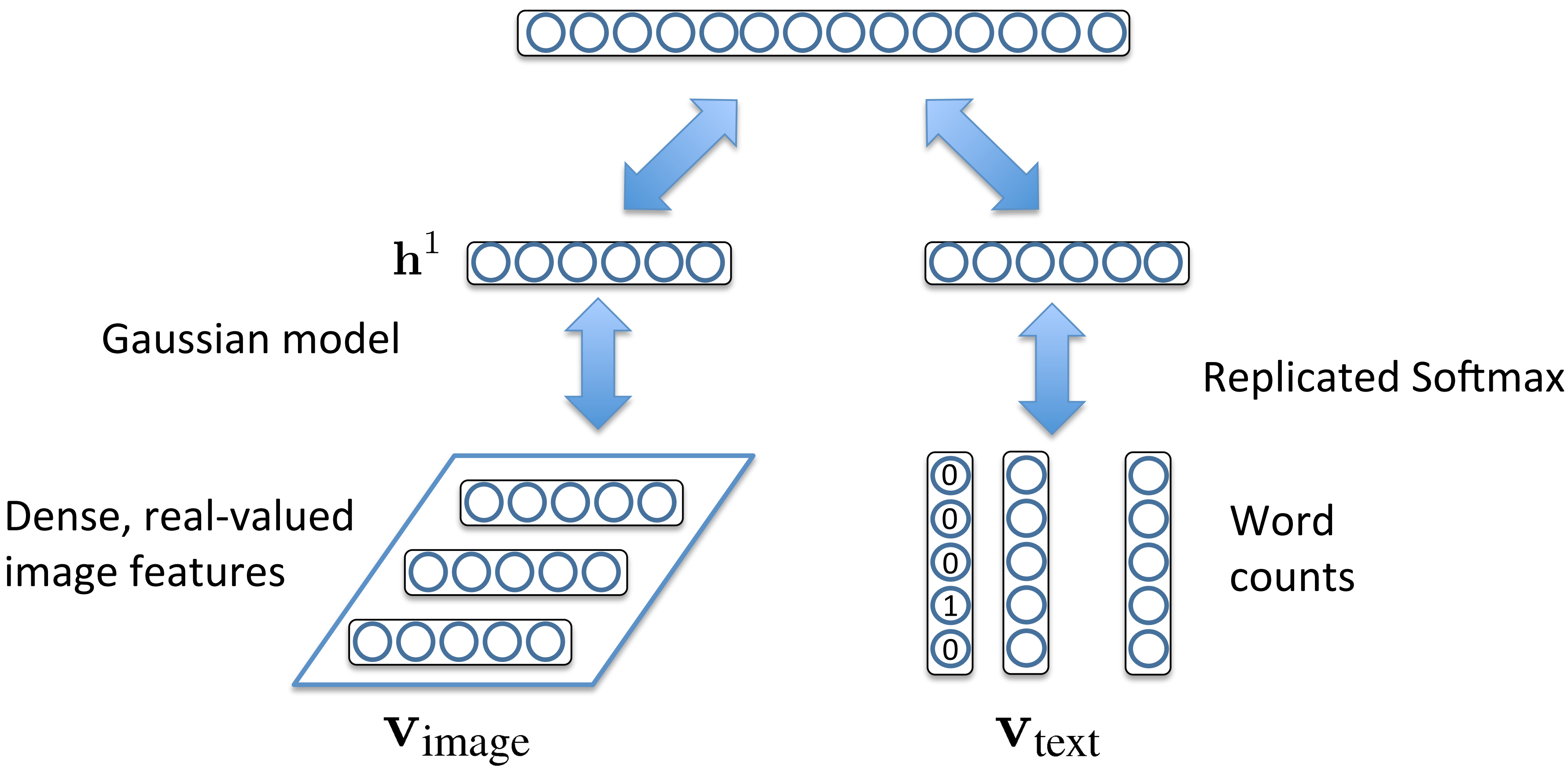
Image	Tags	Tags generated by the model
	pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion	beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves
	mickikrimmel, mickipedia, headshot	portrait, girl, woman, lady, blonde, pretty, gorgeous, expression, model
	< no text>	night, notte, traffic, light, lights, parking, darkness, lowlight, nacht, glow
	unseulpixel, naturey	fall, autumn, trees, leaves, foliage, forest, woods, branches, path

Multimodal DBM



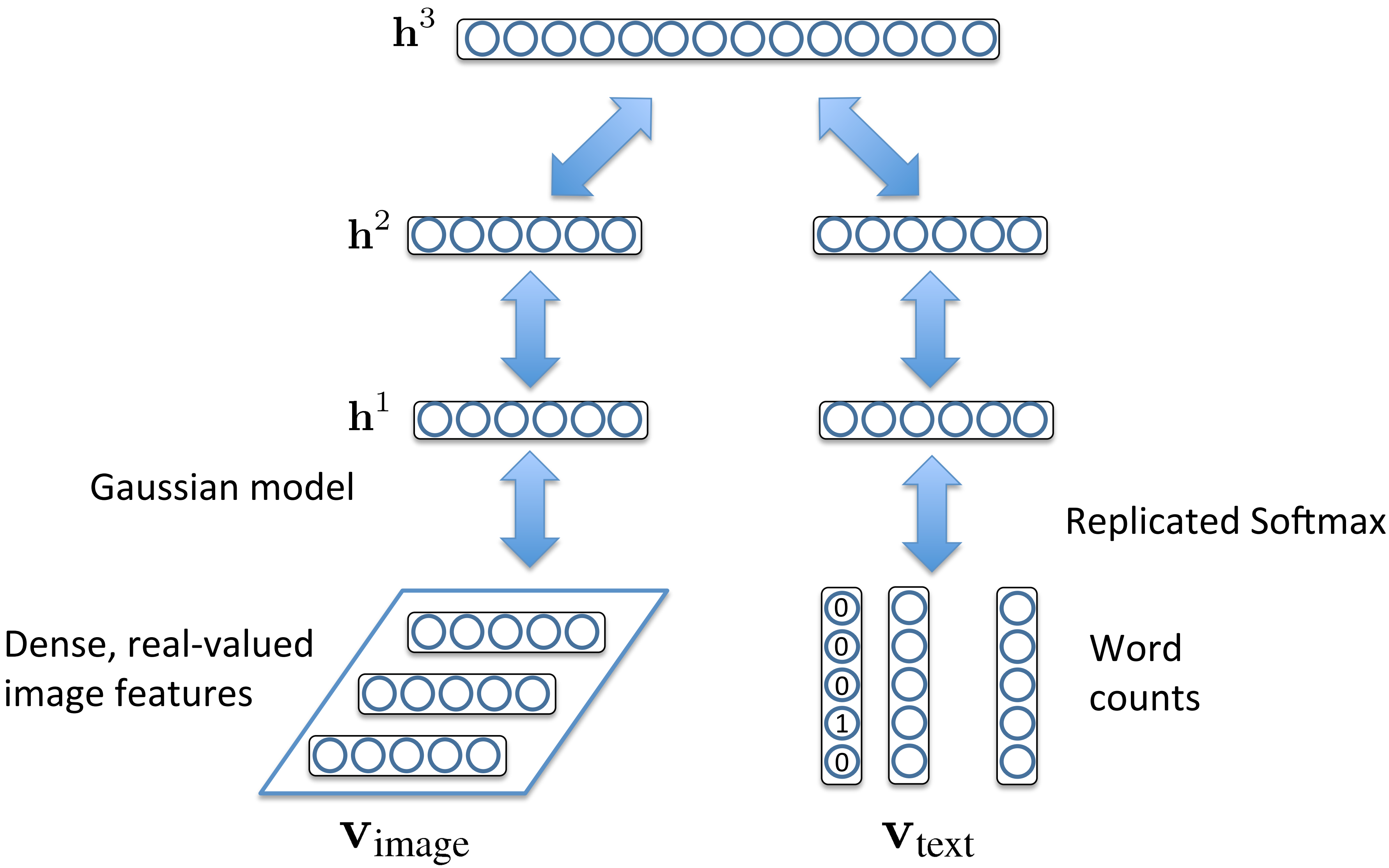
(Srivastava & Salakhutdinov, NIPS 2012, JMLR 2014)

Multimodal DBM



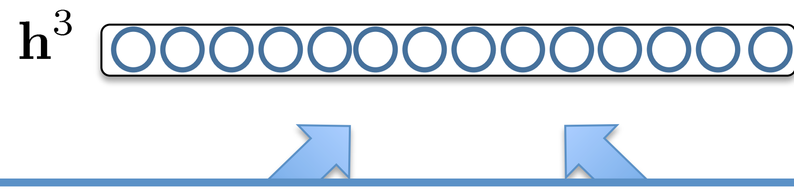
(Srivastava & Salakhutdinov, NIPS 2012, JMLR 2014)

Multimodal DBM



(Srivastava & Salakhutdinov, NIPS 2012, JMLR 2014)

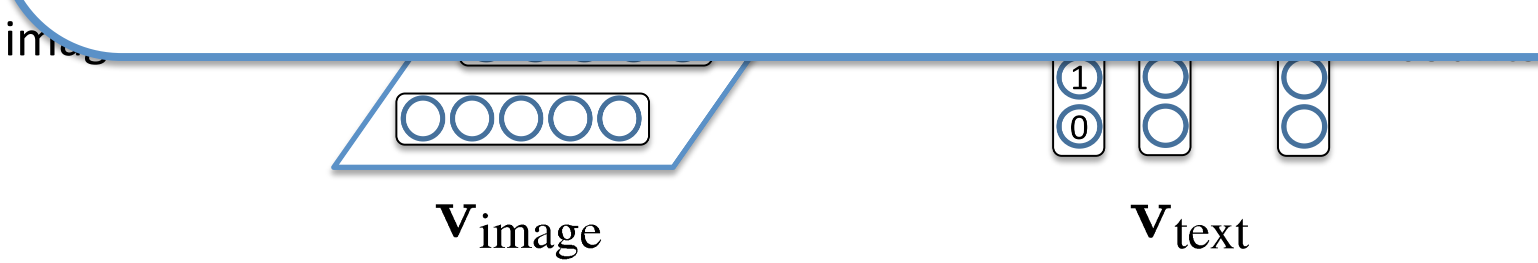
Multimodal DBM



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

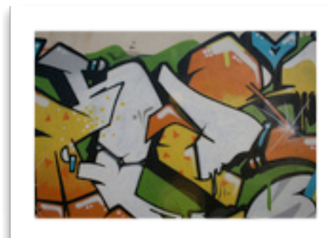
$$\frac{1}{\mathcal{Z}(\theta, M)} \sum_{\mathbf{h}} \exp \Bigg(\underbrace{ - \sum_i \frac{(v_i^m)^2}{2\sigma_i^2} + \sum_{ij} \frac{v_i^m}{\sigma_i} W_{ij}^{(1m)} h_j^{(1m)} + \sum_{jl} W_{jl}^{(2m)} h_j^{(1m)} h_l^{(2m)} }_{\text{Gaussian Image Pathway}}$$

$$\underbrace{ + \sum_{jk} W_{kj}^{(1t)} h_j v_k^t + \sum_{jl} W_{jl}^{(2t)} h_j^{(1t)} h_l^{(2t)} }_{\text{Replicated Softmax Text Pathway}} + \underbrace{ \sum_{lp} W^{(3t)} h_l^{(2t)} h_p^{(3)} + \sum_{lp} W^{(3m)} h_l^{(2m)} h_p^{(3)} }_{\text{Joint } 3^{rd} \text{ Layer}} \Bigg)$$



Text Generated from Images

Given	Generated
	dog, cat, pet, kitten, puppy, ginger, tongue, kitty, dogs, furry
	sea, france, boat, mer, beach, river, bretagne, plage, brittany
	portrait, child, kid, ritratto, kids, children, boy, cute, boys, italy

Given	Generated
	insect, butterfly, insects, bug, butterflies, lepidoptera
	graffiti, streetart, stencil, sticker, urbanart, graff, sanfrancisco
	canada, nature, sunrise, ontario, fog, mist, bc, morning

Text Generated from Images

Given | Generated

portrait, women, army, soldier,
mother, postcard, soldiers

obama, barackobama, election,
politics, president, hope, change,
sanfrancisco, convention, rally

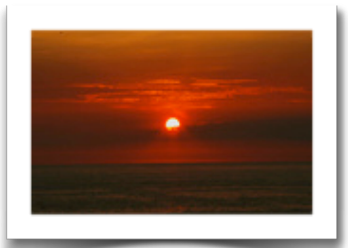
water, glass, beer, bottle,
drink, wine, bubbles, splash,
drops, drop

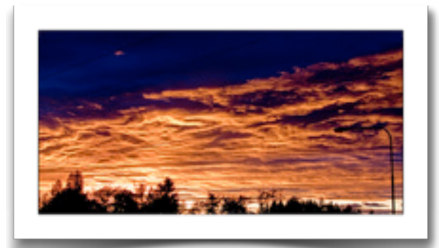
Images from Text

Given

Retrieved

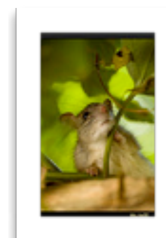
water, red, sunset





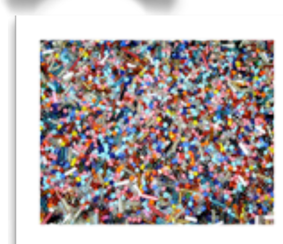
nature, flower, red, green



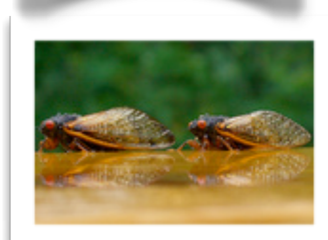



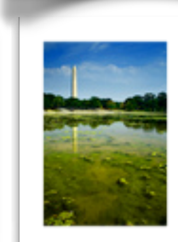
blue, green, yellow, colors



chocolate, cake



MIR-Flickr Dataset

- 1 million images along with user-assigned tags.

sculpture, beauty, stone

d80

nikon, abigfave, goldstaraward, d80, nikond80

food, cupcake, vegan

anawesomeshot, theperfectphotographer, flash, damniwishidtakenthat, spiritofphotography

nikon, green, light, photoshop, apple, d70

white, yellow, abstract, lines, bus, graphic

sky, geotagged, reflection, cielo, bilbao, reflejo

Huiskes et. al.

Results

- Logistic regression on top-level representation.
- Multimodal Inputs

Mean Average Precision

Learning Algorithm	MAP	Precision@50
Random	0.124	0.124
LDA [Huiskes et. al.]	0.492	0.754
SVM [Huiskes et. al.]	0.475	0.758
DBM-Labelled	0.526	0.791
Deep Belief Net	0.638	0.867
Autoencoder	0.638	0.875
DBM	0.641	0.873

Labeled 25K examples

+ 1 Million unlabelled

Generating Sentences

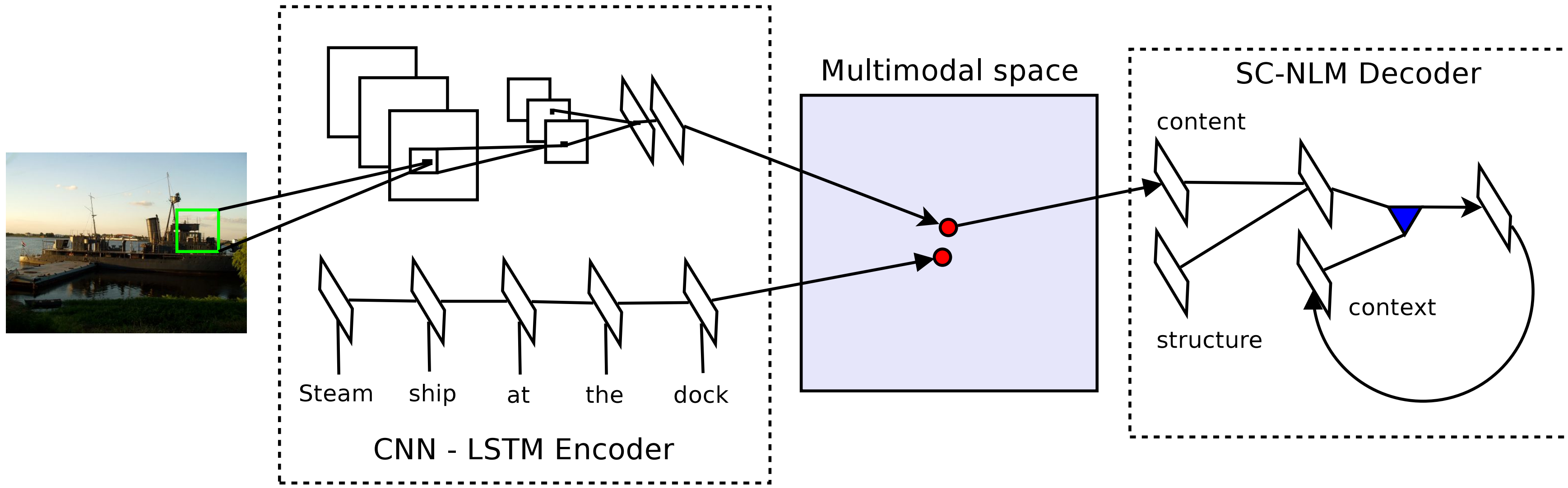
- More challenging problem.
- How can we generate complete descriptions of images?

Input

Output

A man skiing down the snow covered mountain with a dark sky in the background.

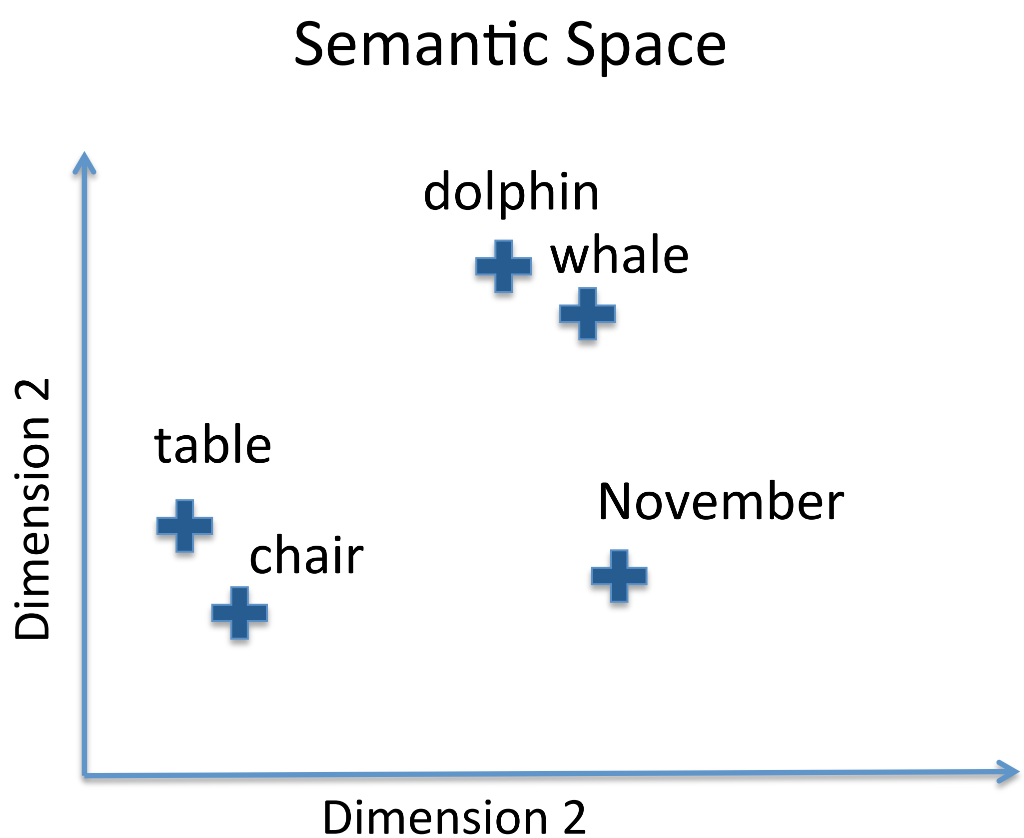
Encode-Decode Framework



- Encoder: CNN and Recurrent Neural Net for a joint image-sentence embedding.
- Decoder: A neural language model that combines structure and content vectors for generating a sequence of words

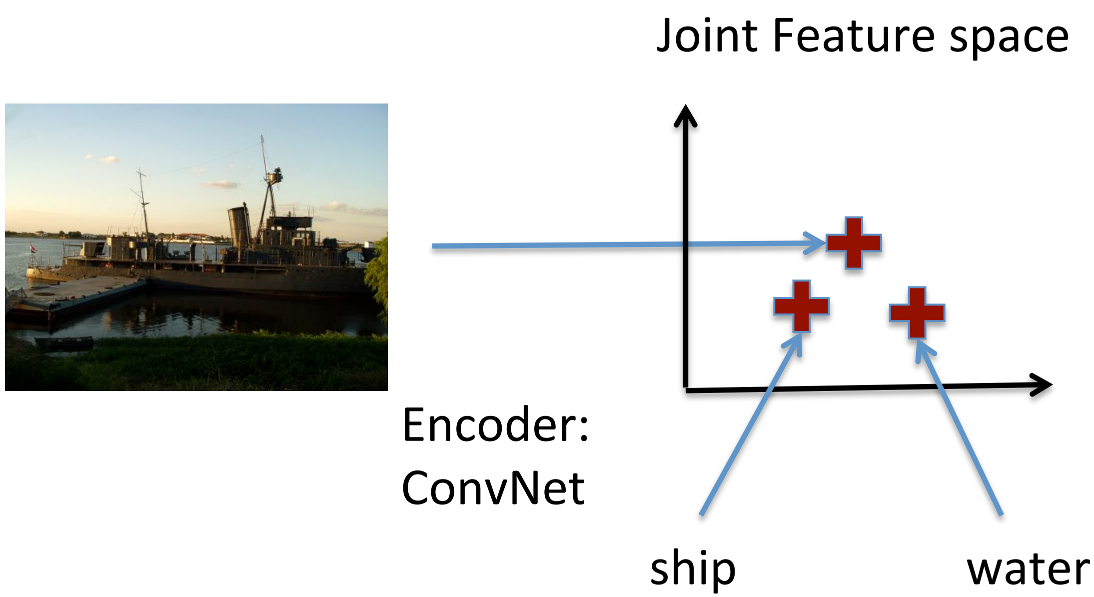
Representation of Words

- Key Idea: Each word w is represented as a D-dimensional real-valued vector $r_w \in R^K$.



Bengio et.al., 2003, Mnih et. al., 2008, Mikolov et. al., 2009, Kiros et.al. 2014

An Image-Text Encoder



- Learn a joint embedding space of images and text:
  - Can condition on anything (images, words, phrases, etc)
  - Natural definition of a scoring function (inner products in the joint space).

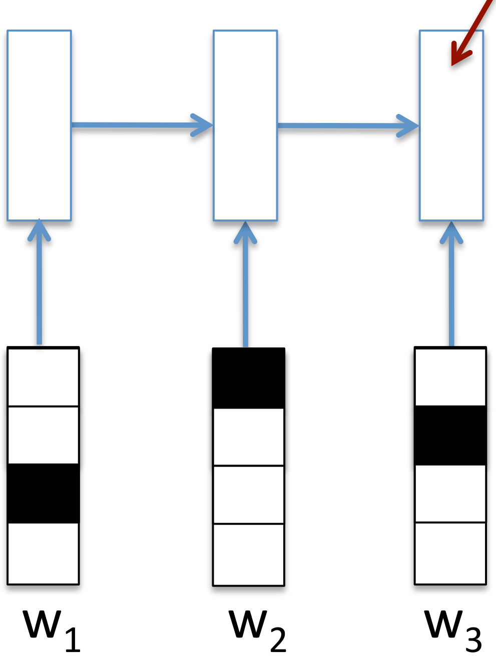
Socher 2013, Frome 2013, Kiros 2014

An Image-Text Encoder

Sentence Representation

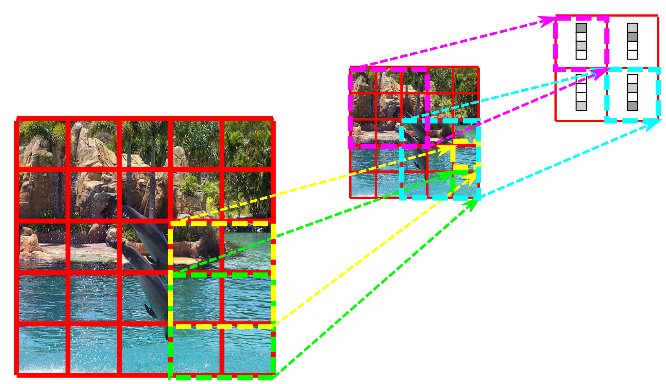
See Skip-Thought Vectors, (Kiros et.al. arXiv 2015)

Image Representation



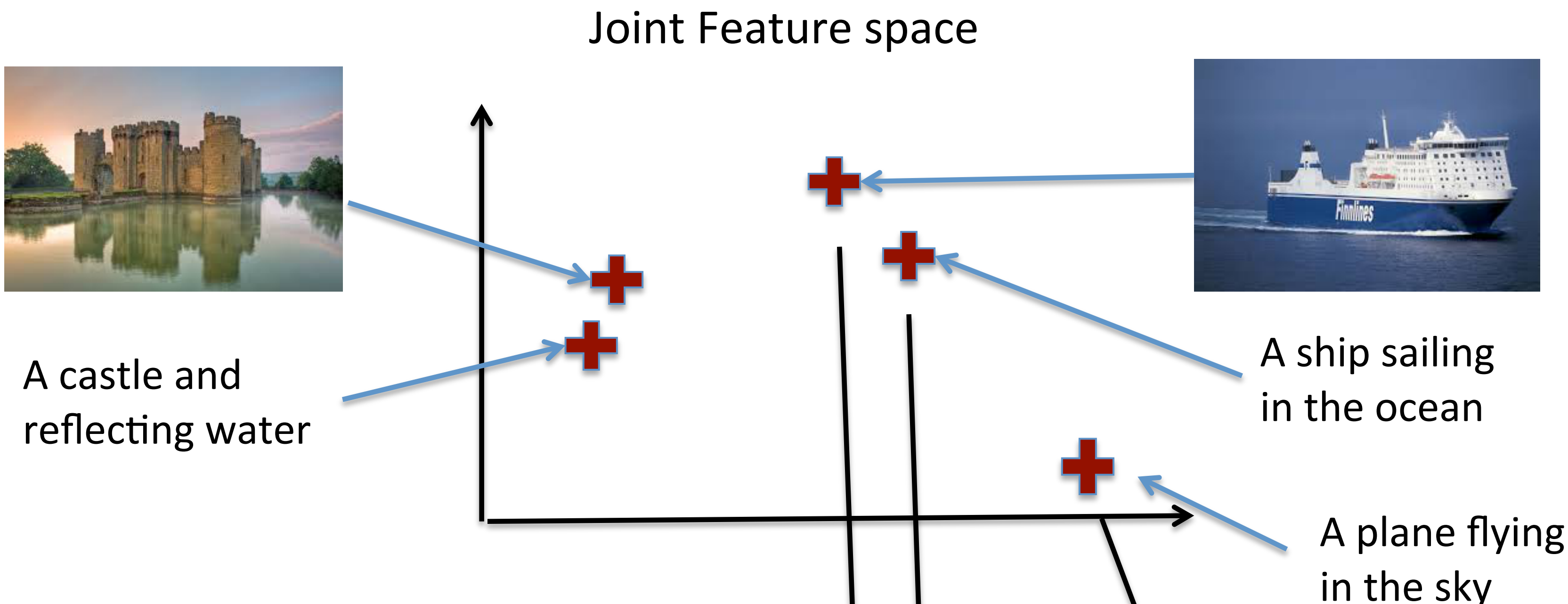
1-of-V encoding of words

Recurrent Neural Network (LSTM)



Convolutional Neural Network

An Image-Text Encoder



Minimize the following objective:

Images: $$\sum_{\mathbf{x}} \sum_{k} \max\{0, \alpha - s(\mathbf{x}, \mathbf{v}) + s(\mathbf{x}, \mathbf{v}_k)\} \; +$$

Text: $$\sum_{\mathbf{v}} \sum_{k} \max\{0, \alpha - s(\mathbf{v}, \mathbf{x}) + s(\mathbf{v}, \mathbf{x}_k)\}$$

Retrieving Sentences for Images

The dogs are in the snow in front of a fence .

Four men playing basketball , two from each team .

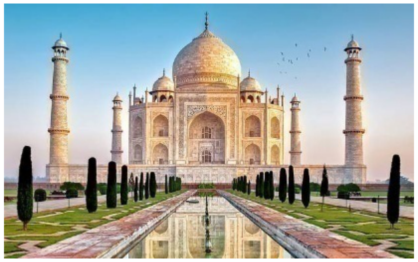
A boy skateboarding

Two men and a woman smile at the camera .

Women participate in a skit onstage .

A man is doing tricks on a bicycle on ramps in front of a crowd .

Tagging and Retrieval



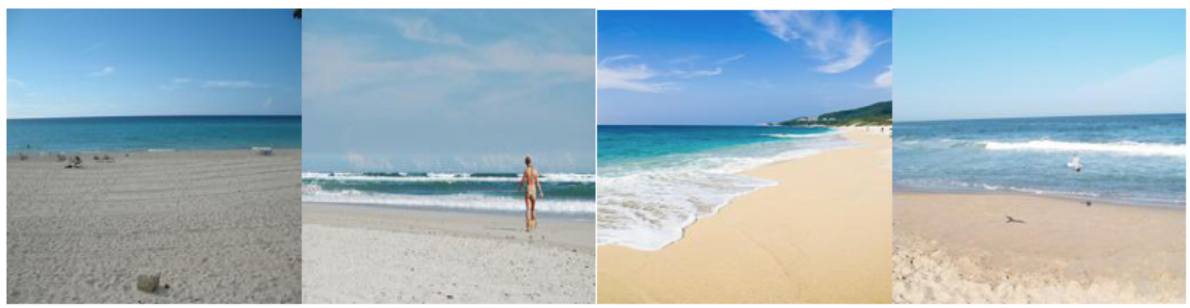
mosque, tower, building, cathedral, dome, castle

ski, skiing, skiers, skiiers, snowmobile

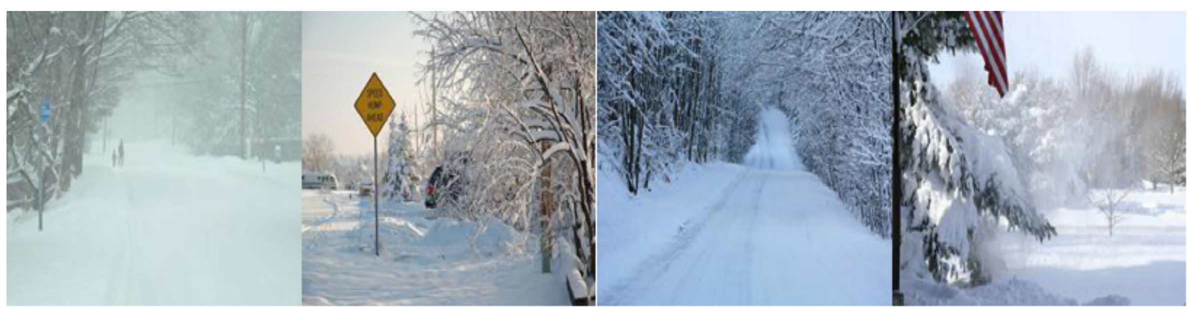
kitchen, stove, oven, refrigerator, microwave

bowl, cup, soup, cups, coffee

beach



snow



Retrieval with Adjectives

fluffy

delicious

Multimodal Linguistic Regularities

Nearest Images

- blue + red =

- blue + yellow =

- yellow + red =

(Kiros, Salakhutdinov, Zemel, TACL 2015)

Multimodal Linguistic Regularities

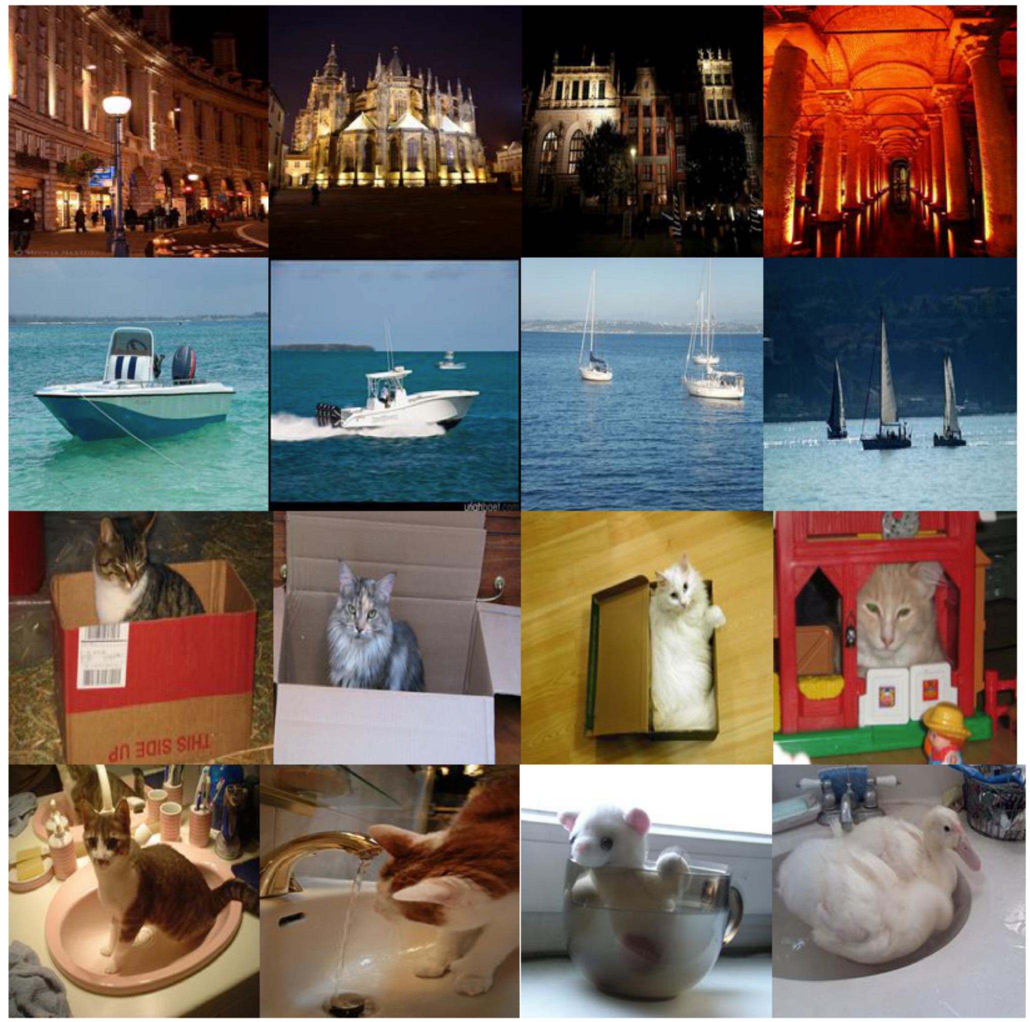
- day + night =

- flying + sailing =

- bowl + box =

- box + bowl =

Nearest Images



(Kiros, Salakhutdinov, Zemel, TACL 2015)

How About Generating Sentences!

Input

Output

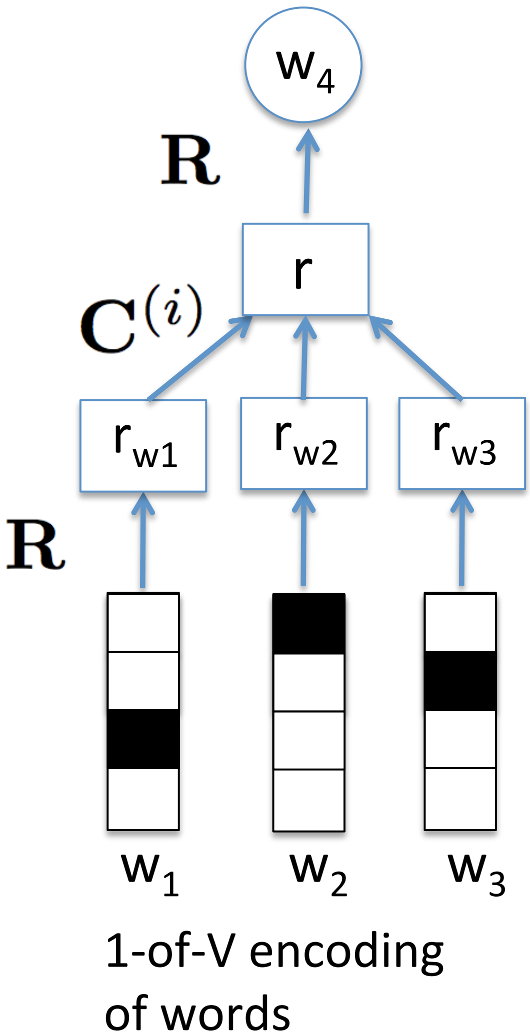
A man skiing down the snow covered mountain with a dark sky in the background.

Need to model:

$$p(w_1, w_2, ..., w_n) =$$

$$p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)...p(w_n|w_1, w_2, ..., w_{n-1})$$

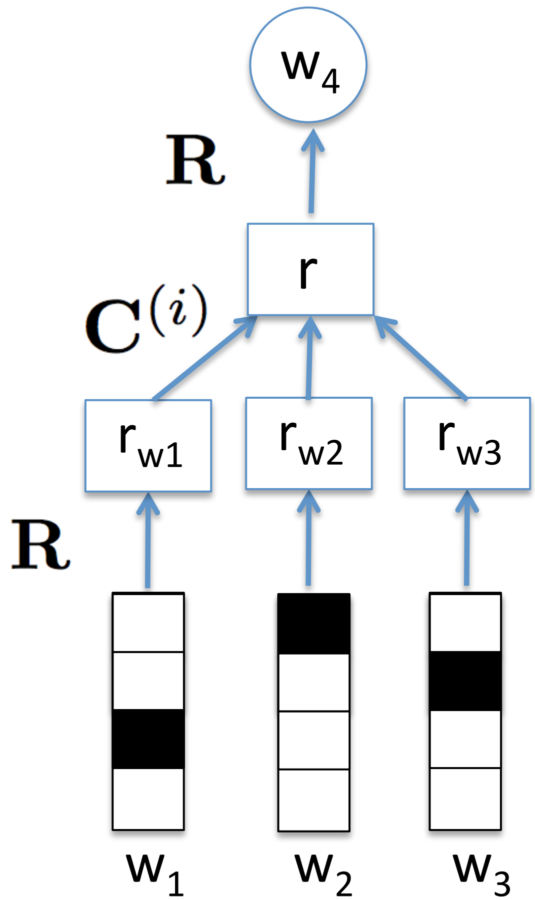
Log-bilinear Neural Language Model



- Feedforward neural network with a single linear hidden layer.
- Each word w is represented as a K-dim real-valued vector $r_w \in R^K$.
- **R** denote the V $\times$ K matrix of word representation vectors, where V is the vocabulary size.
- $(w_1, \ldots, w_{n-1})$ is tuple of n-1 words, where n-1 is the context size. The next word representation becomes:

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \underbrace{\mathbf{C}^{(i)}}_{K \times K \text{ context parameter matrices}} \mathbf{r}_{w_i},$$

Log-bilinear Neural Language Model



1-of-V encoding of words

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{r}_{w_i},$$

Predicted representation of r_{wn}.

- The conditional probability of the next word given by:

$$P(w_n = i | w_{1:n-1}) = \frac{\exp(\hat{\mathbf{r}}^T \mathbf{r}_i + b_i)}{\underbrace{\sum_{j=1}^{V} \exp(\hat{\mathbf{r}}^T \mathbf{r}_j + b_j)}_{\text{Can be expensive to compute}}}$$

Bengio et.al. 2003

Multiplicative Model

- We represent words as a tensor:

$$\boldsymbol{\mathcal{T}} \in \mathbb{R}^{V \times K \times G}$$

  where G is the number of tensor slices.

- Given an attribute vector u $\in$ R^G (e.g. image features), we can compute attribute-gated word representations as:

$$\boldsymbol{\mathcal{T}}^u = \sum_{i=1}^{G} u_i \boldsymbol{\mathcal{T}}^{(i)}$$

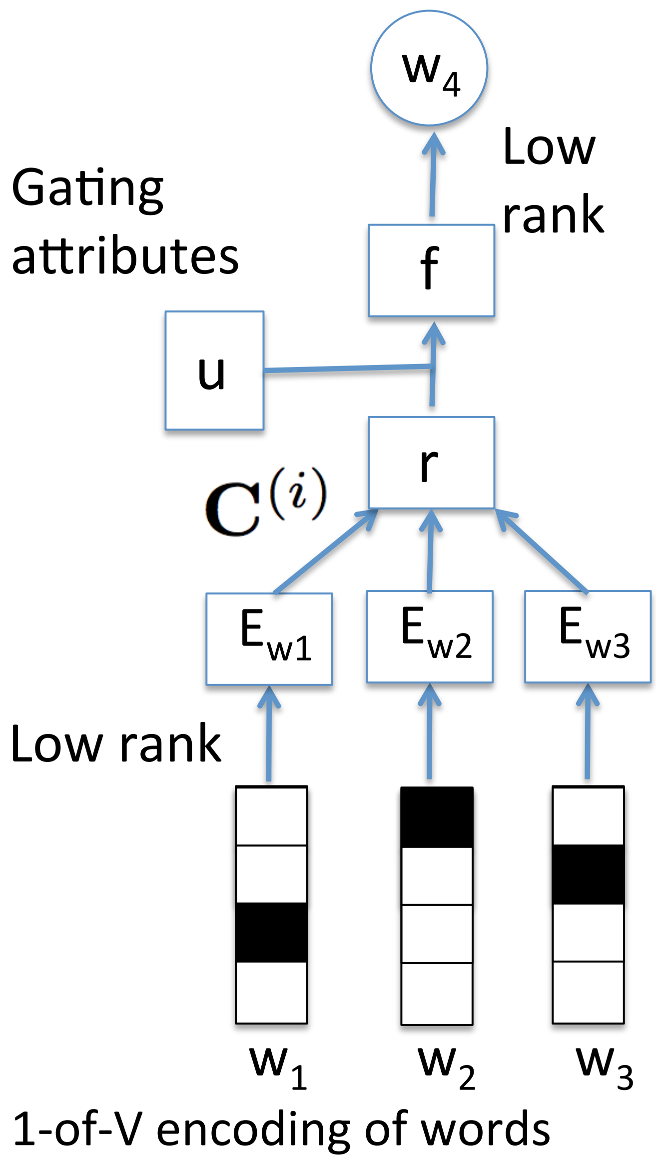
- Re-represent Tensor in terms of 3 lower-rank matrices (where F is the number of pre-chosen factors):

$$\mathbf{W}^{fk} \in \mathbb{R}^{F \times K}, \mathbf{W}^{fd} \in \mathbb{R}^{F \times G} \quad \mathbf{W}^{fv} \in \mathbb{R}^{F \times V}$$

$$\boldsymbol{\mathcal{T}}^u = (\mathbf{W}^{fv})^\top \cdot \mathrm{diag}(\mathbf{W}^{fd}\mathbf{u}) \cdot \mathbf{W}^{fk}$$

(Kiros, Zemel, Salakhutdinov, NIPS 2014)

Multiplicative Log-bilinear Model



- Let $\mathbf{E} = (\mathbf{W}^{fk})^\top \mathbf{W}^{fv}$ denote a folded K $\times$ V matrix of word embeddings.

- Then the predicted next word representation is:

$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{E}(:, w_i)$$

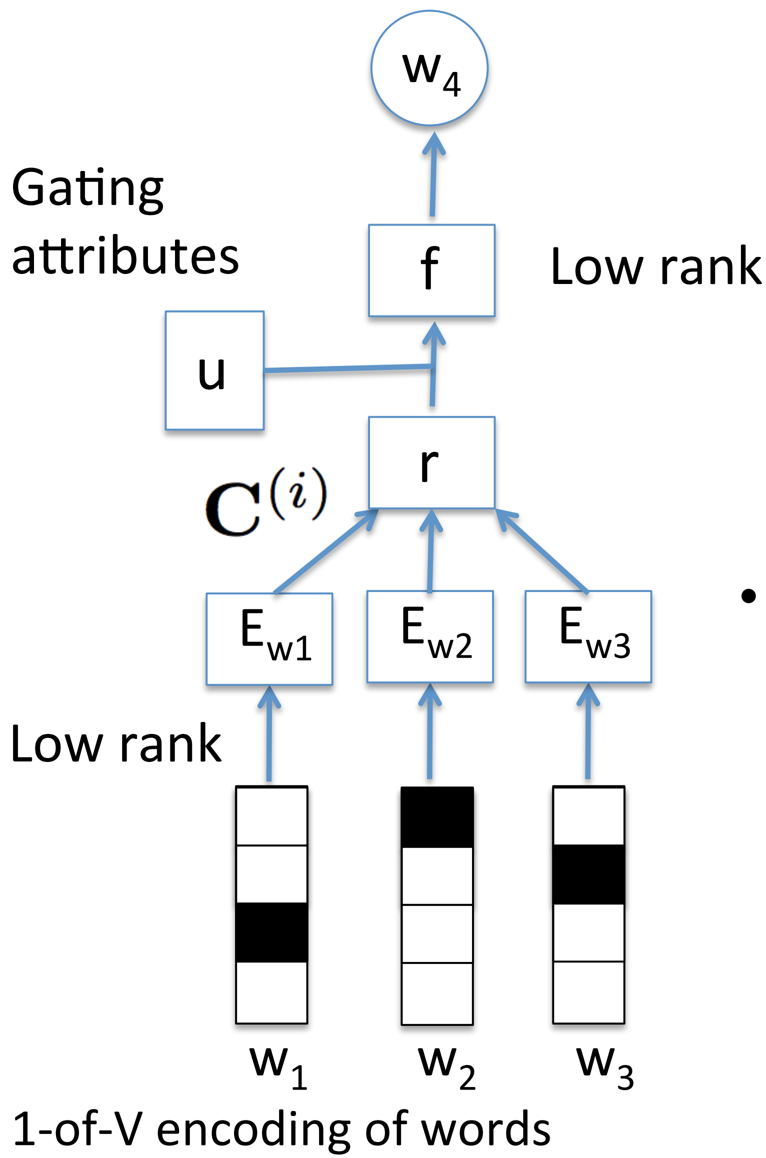
- Given next word representation r, the factor outputs are:

$$\mathbf{f} = (\mathbf{W}^{fk}\hat{\mathbf{r}}) \bullet (\mathbf{W}^{fd}\mathbf{x})$$

Component-wise product

(Kiros, Zemel, Salakhutdinov, NIPS 2014)

Multiplicative Log-bilinear Model



$$\mathbf{E} = (\mathbf{W}^{fk})^\top \mathbf{W}^{fv}$$

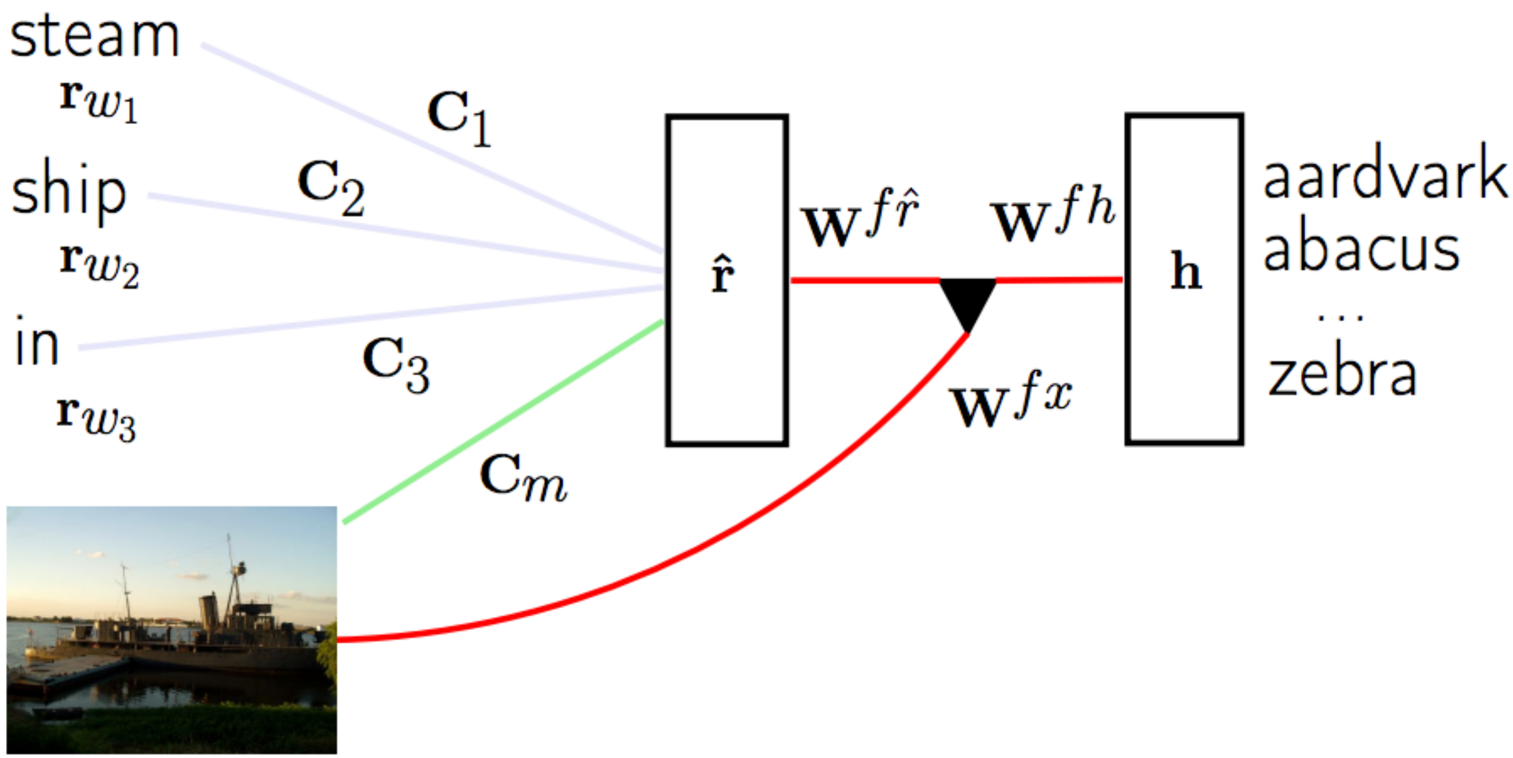
$$\hat{\mathbf{r}} = \sum_{i=1}^{n-1} \mathbf{C}^{(i)} \mathbf{E}(:, w_i)$$

$$\mathbf{f} = (\mathbf{W}^{fk}\hat{\mathbf{r}}) \bullet (\mathbf{W}^{fd}\mathbf{x})$$

- The conditional probability of the next word given by:

$$P(w_n = i | w_{1:n-1}, \mathbf{u}) = \frac{\exp\big((\mathbf{W}^{fv}(:, i))^\top \mathbf{f} + b_i\big)}{\sum_{j=1}^{V} \exp\big((\mathbf{W}^{fv}(:, j))^\top \mathbf{f} + b_j\big)}$$

Decoding: Neural Language Model



- Image features are gating the hidden-to-output connections when predicting the next word.
- We can also condition on POS tags when generating a sentence.

(Kiros, Salakhutdinov, Zemel, ICML 2014)

Caption Generation

a car is parked in
the middle of nowhere .

a wooden table and chairs
arranged in a room .

there is a cat sitting on a shelf .

a ferry boat on a marina
with a group of people .

a little boy with a bunch
of friends on the street .

Caption Generation

the two birds are trying
to be seen in the water .
(can't count)

a giraffe is standing next
to a fence in a field .
(hallucination)

a parked car while
driving down the road .
(contradiction)

Caption Generation

the two birds are trying to be seen in the water . (can't count)

a giraffe is standing next to a fence in a field . (hallucination)

a parked car while driving down the road . (contradiction)

the handlebars are trying to ride a bike rack . (nonsensical)

a woman and a bottle of wine in a garden . (gender)

Caption Generation

TAGS:

colleagues waiters waiter entrepreneurs busboy

Model Samples

- Two men in a room talking on a table .
- Two men are sitting next to each other .
- Two men are having a conversation at a table .
- Two men sitting at a desk next to each other .

Caption Generation with Visual Attention

A woman is throwing a frisbee in a park.

Xu et.al., ICML 2015

Caption Generation with Visual Attention

A woman is throwing a frisbee in a park.

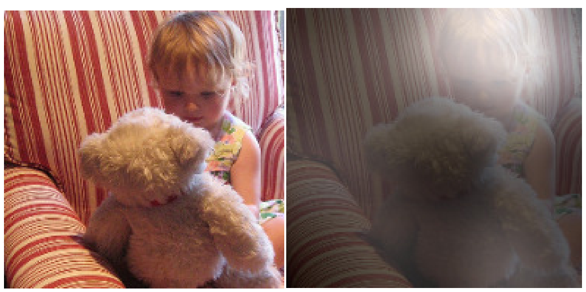
Xu et.al., ICML 2015

Caption Generation with Visual Attention

A woman is throwing a frisbee in a park.

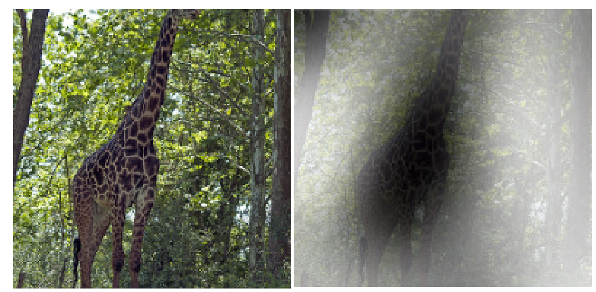
A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.



A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.



A giraffe standing in a forest with trees in the background.

Xu et.al., ICML 2015

Results

Flickr30K								
	Image Annotation				Image Search			
Model	**R@1**	**R@5**	**R@10**	**Med *r***	**R@1**	**R@5**	**R@10**	**Med *r***
Random Ranking	0.1	0.6	1.1	631	0.1	0.5	1.0	500
† DeViSE [5]	4.5	18.1	29.2	26	6.7	21.9	32.7	25
† SDT-RNN [6]	9.6	29.8	41.1	16	8.9	29.8	41.1	16
† DeFrag [15]	14.2	37.7	51.3	10	10.2	30.8	44.2	14
† DeFrag + Finetune CNN [15]	16.4	40.2	54.7	8	10.3	31.4	44.5	13
m-RNN [7]	18.4	40.2	50.9	10	12.6	31.2	41.5	16
Our model	14.8	39.2	50.9	10	11.8	34.0	46.3	13
Our model (OxfordNet)	**23.0**	**50.7**	**62.9**	**5**	**16.8**	**42.0**	**56.5**	**8**

- R@K is Recall@K (high is good).
- Med r is the median rank (low is good).

Caption Generation with Visual Attention

- Montreal/Toronto team takes 3rd place on Microsoft COCO caption generation competition, finishing slightly behind Google and Microsoft. This is based on the human evaluation results.

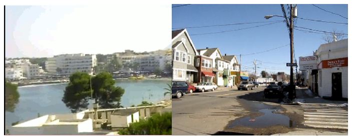
Table-C5 Table-C40 **Table-human** Last update: June 8, 2015. Visit CodaLab for the latest results.

	M1	M2	M3	M4	M5
Human[5]	0.638	0.675	4.836	3.428	0.352
Google[4]	0.273	0.317	4.107	2.742	0.233
MSR[8]	0.268	0.322	4.137	2.662	0.234
Montreal/Toronto[10]	0.262	0.272	3.932	2.832	0.197
MSR Captivator[9]	0.250	0.301	4.149	2.565	0.233
Berkeley LRCN[2]	0.246	0.268	3.924	2.786	0.204
m-RNN[15]	0.223	0.252	3.897	2.595	0.202
Nearest Neighbor[11]	0.216	0.255	3.801	2.716	0.196

Xu et.al., ICML 2015

Multi-Modal Models

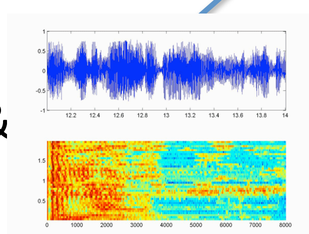
Images



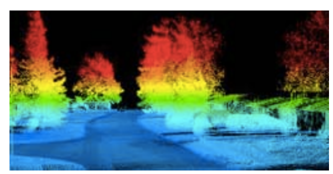
Text & Language

Video

Speech & Audio



Lidar



Develop learning systems that come closer to displaying human like intelligence

Summary

- Efficient learning algorithms for Deep Learning Models. Learning more adaptive, robust, and structured representations.

Text & image retrieval / Object recognition

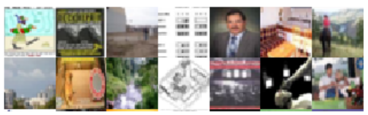
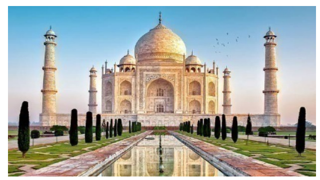
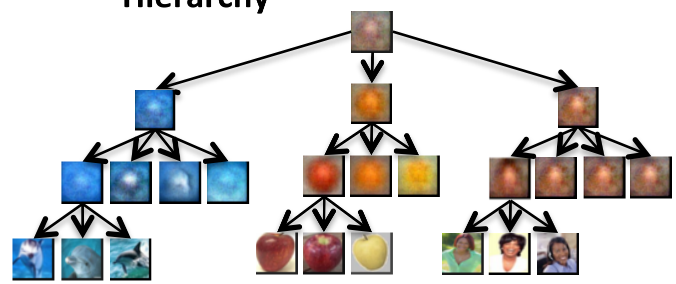


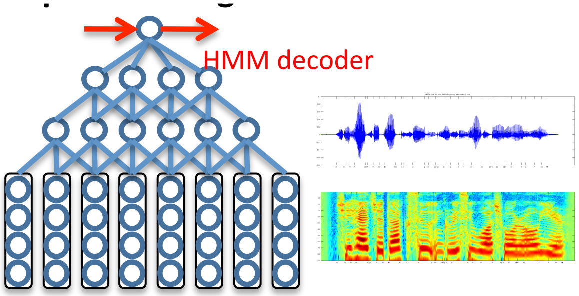
Image Tagging



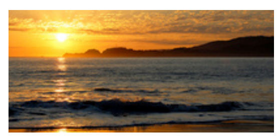
mosque, tower, building, cathedral, dome, castle

Learning a Category Hierarchy

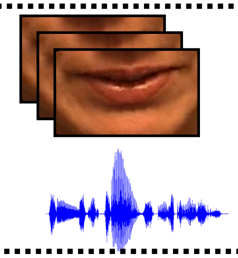




Multimodal Data



sunset, pacific ocean, beach, seashore



Caption Generation

a car is parked in the middle of nowhere .

a ferry boat on a marina with a group of people .

a little boy with a bunch of friends on the street .

- Deep models improve the current state-of-the art in many application domains:
  - Object recognition and detection, text and image retrieval, handwritten character and speech recognition, and others.

Thank you

Code is available at:
http://deeplearning.cs.toronto.edu/