Deep Learning

Russ Salakhutdinov

Department of Computer Science
Department of Statistical Sciences
University of Toronto
Mining for Structure

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

Deep Learning Models that support inferences and discover structure at multiple levels.

• Develop statistical models that can discover underlying structure, cause, or statistical correlation from data in \textit{unsupervised} or \textit{semi-supervised} way.
• Multiple application domains.
Deep Boltzmann Machine

25,000 characters from 50 alphabets around the world.

- 3,000 hidden variables
- 784 observed variables (28 by 28 images)
- Over 2 million parameters

Model $P(\text{image})$

Sanskrit

Bernoulli Markov Random Field
Deep Boltzmann Machine

Conditional Simulation

P(image | partial image) Bernoulli Markov Random Field
Deep Boltzmann Machine

Conditional Simulation

Why so difficult?

$2^{28 \times 28}$ possible images!

Bernoulli Markov Random Field

$P(\text{image} | \text{partial image})$
Deep Generative Model

Model P(document)

Reuters dataset: 804,414 newswire stories: unsupervised

Bag of words
Convolutinal Deep Models for Image Recognition

- Learning multiple layers of representation.

(LeCun, 1992)
Learning Feature Representations

Input Space

pixel 1
pixel 2

Learning Algorithm

Segway
Non-Segway
Learning Feature Representations

Input Space

Feature Space

Feature Representation

Learning Algorithm
How is computer perception done?

Input Data → Low-level features → Learning Algorithm

Object detection
- Image
- Low-level vision features
- Recognition

Audio classification
- Audio
- Low-level audio features
- Speaker identification

Slide Credit: Honglak Lee
Computer vision features

SIFT

Spin image

HoG

RIFT

Textons

GLOH

Slide Credit: Honglak Lee
Audio features

Spectrogram

MFCC

Flux

ZCR

Rolloff
Audio features

Feature Learning: Can we learn meaningful features from unlabeled, partially labeled data?
Talk Roadmap

Part 1: Deep Networks

• Restricted Boltzmann Machines: Learning low-level features.
• Deep Belief Networks: Learning Part-based Hierarchies.


• Deep Boltzmann Machines
• Learning Structured and Robust Models
• Multimodal Learning
Restricted Boltzmann Machines

- Undirected bipartite graphical model

- Stochastic binary visible variables:
  \[ \mathbf{v} \in \{0, 1\}^D \]

- Stochastic binary hidden variables:
  \[ \mathbf{h} \in \{0, 1\}^F \]

The energy of the joint configuration:

\[
E(\mathbf{v}, \mathbf{h}; \theta) = - \sum_{ij} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j
\]

\[ \theta = \{W, a, b\} \text{ model parameters.} \]
Restricted Boltzmann Machines

Probability of the joint configuration is given by the Boltzmann distribution:

\[ P_\theta(v, h) = \frac{1}{\mathcal{Z}(\theta)} \exp \left( -E(v, h; \theta) \right) \]

\[ \mathcal{Z}(\theta) = \sum_{h,v} \exp \left( -E(v, h; \theta) \right) \]

Markov random fields, Boltzmann machines, log-linear models.
**Restricted Boltzmann Machines**

- **Bipartite Structure**: No interaction between hidden variables

Inferring the distribution over the hidden variables is easy:

\[
P(h|v) = \prod_j P(h_j|v) \quad P(h_j = 1|v) = \frac{1}{1 + \exp(-\sum_i W_{ij}v_i - a_j)}
\]

Similarly:

\[
P(v|h) = \prod_i P(v_i|h) \quad P(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_j W_{ij}h_j - b_i)}
\]

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

Observed Data
Subset of 25,000 characters

New Image:
\[ p(h_7 = 1|v) = \sigma \left( 0.99 \times \right) + 0.97 \times + 0.82 \times \cdots \]

 Observed Data
Subset of 1000 features

Learned W: “edges”

Most hidden variables are off

Represent: as
\[ P(h|v) = [0, 0, 0.82, 0, 0, 0.99, 0, 0 \ldots] \]
Model Learning

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

Maximize (penalized) log-likelihood objective:

\[
L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_{\theta}(\mathbf{v}^{(n)}) - \frac{\lambda}{N} \| W \|_F^2
\]
Model Learning

Maximize (penalized) log-likelihood objective:

\[ L(\theta) = \frac{1}{N} \sum_{n=1}^{N} \log P_\theta(v^{(n)}) - \frac{\lambda}{N} ||W||_F^2 \]

Regularization

Derivative of the log-likelihood:

\[
\frac{\partial L(\theta)}{\partial W_{ij}} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial}{\partial W_{ij}} \log \left( \sum_h \exp \left[ v^{(n)^T} W h + a^T h + b^T v^{(n)} \right] \right) - \frac{\partial}{\partial W_{ij}} \log Z(\theta) - \frac{2\lambda}{N} W_{ij}
\]

\[ = E_{P_{data}}[v_i h_j] - E_{P_\theta}[v_i h_j] - \frac{2\lambda}{N} W_{ij} \]
Model Learning

Derivative of the log-likelihood:

\[
\frac{\partial L(\theta)}{\partial W_{ij}} = \mathbb{E}_{P_{data}}[v_i h_j] - \mathbb{E}_{P_\theta}[v_i h_j] - \sum_{v, h} v_i h_j P_\theta(v, h)
\]

- Easy to compute exactly
- Difficult to compute: exponentially many configurations.
  Use MCMC

Approximate maximum likelihood learning

\[
P_{data}(v, h; \theta) = P(h|v; \theta)P_{data}(v)
\]

\[
P_{data}(v) = \frac{1}{N} \sum_{n} \delta(v - v^{(n)})
\]
Approximate Learning

• An approximation to the gradient of the log-likelihood objective:

$$\frac{\partial L(\theta)}{\partial W_{ij}} = \mathbb{E}_{P_{data}}[v_i h_j] - \mathbb{E}_{P_\theta}[v_i h_j]$$

$$\sum_{v, h} v_i h_j P_\theta(v, h)$$

• Replace the average over all possible input configurations by samples.
• Run MCMC chain (Gibbs sampling) starting from the observed examples.

- Initialize $v^0 = v$
- Sample $h^0$ from $P(h | v^0)$
- For $t=1:T$
  - Sample $v^t$ from $P(v | h^{t-1})$
  - Sample $h^t$ from $P(h | v^t)$
Approximate ML Learning for RBMs

Run Markov chain (alternating Gibbs Sampling):

\[
P(h|v) = \prod_j P(h_j|v) \quad P(h_j = 1|v) = \frac{1}{1 + \exp(-\sum_i W_{ij}v_i - a_j)}
\]

\[
P(v|h) = \prod_i P(v_i|h) \quad P(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_j W_{ij}h_j - b_i)}
\]
Contrastive Divergence

A quick way to learn RBM:

\[ P(h|v) \]

- Start with a training vector on the visible units.
- Update all the hidden units in parallel.
- Update the all the visible units in parallel to get a “reconstruction”.
- Update the hidden units again.

\[ P(v|h) \]

Update model parameters:

\[ \Delta W_{ij} = E_{P_{data}}[v_i h_j] - E_{P_1}[v_i h_j] \]

Implementation: \(~10\) lines of Matlab code.
RBM for Real-valued Data

\[ P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} + \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{F} a_j h_j \right) \]

Pair-wise

\[ \theta = \{ W, a, b \} \]

Unary

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

Gaussian-Bernoulli RBM:

- Stochastic real-valued visible variables \( v \in \mathbb{R}^D \).
- Stochastic binary hidden variables \( h \in \{0, 1\}^F \).
- Bipartite connections.
RBMs for Real-valued Data

\[ P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} + \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{F} a_j h_j \right) \]

\[ \theta = \{W, a, b\} \]

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

4 million unlabelled images

Learned features (out of 10,000)
RBMs for Real-valued Data

\[
P_{\theta}(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{j=1}^{F} W_{ij} h_j \frac{v_i}{\sigma_i} + \sum_{i=1}^{D} \frac{(v_i - b_i)^2}{2\sigma_i^2} + \sum_{j=1}^{F} a_j h_j \right)
\]

\[
\theta = \{ W, a, b \}
\]

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

4 million unlabelled images

Learned features (out of 10,000)

\[ p(h_{29} = 1|v) + 0.8 \ast \]
RBMs for Word Counts

Replicated Softmax Model: undirected topic model:

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

\[
P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{k=1}^{K} \sum_{j=1}^{F} W_{ij}^k v_i^k h_j + \sum_{i=1}^{D} \sum_{k=1}^{K} v_i^k b_i^k + \sum_{j=1}^{F} h_j a_j \right)
\]

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

(Salakhutdinov & Hinton, NIPS 2010, Srivastava & Salakhutdinov, NIPS 2012)
RBMs for Word Counts

\[
P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{i=1}^{D} \sum_{k=1}^{K} \sum_{j=1}^{F} W_{ij}^k v_i^k h_j + \sum_{i=1}^{D} \sum_{k=1}^{K} v_i^k b_i^k + \sum_{j=1}^{F} h_j a_j \right)
\]

\[
\theta = \{W, a, b\}
\]

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

Learned features: ```topics```

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

- Russian
- Russia
- Moscow
- Yeltsin
- Soviet
- Clinton
- House
- President
- Bill
- Congress
- Computer
- System
- Product
- Software
- Develop
- Trade
- Country
- Import
- World
- Economy
- Stock
- Wall
- Street
- Point
- Dow
Collaborative Filtering

\[ P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{ijk} W_{ijk} v_i^k h_j + \sum_{ik} b_i^k v_i^k + \sum_j a_j h_j \right) \]

- Binary hidden: user preferences
- Multinomial visible: user ratings

Learned features: `genre`

Netflix dataset:
- 480,189 users
- 17,770 movies
- Over 100 million ratings

- Fahrenheit 9/11
- Bowling for Columbine
- The People vs. Larry Flynt
- Canadian Bacon
- La Dolce Vita

- Independence Day
- The Day After Tomorrow
- Con Air
- Men in Black II
- Men in Black

- Friday the 13th
- The Texas Chainsaw Massacre
- Children of the Corn
- Child's Play
- The Return of Michael Myers

- Scary Movie
- Naked Gun
- Hot Shots!
- American Pie
- Police Academy

State-of-the-art performance on the Netflix dataset.

(Salakhutdinov, Mnih, Hinton, ICML 2007)
Different Data Modalities

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

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

• It is easy to infer the states of the hidden variables:
Product of Experts

The joint distribution is given by:

\[
P_\theta(v, h) = \frac{1}{Z(\theta)} \exp \left( \sum_{ij} W_{ij} v_i h_j + \sum_i b_i v_i + \sum_j a_j h_j \right)
\]

Marginalizing over hidden variables:

\[
P_\theta(v) = \sum_h P_\theta(v, h) = \frac{1}{Z(\theta)} \prod_i \exp(b_i v_i) \prod_j \left( 1 + \exp(a_j + \sum_i W_{ij} v_i) \right)
\]

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

(Salakhutdinov & Hinton, NIPS 2010)
Product of Experts

The joint distribution is given by:

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

Marginalizing over hidden variables:

$$P_{\theta}(v) = \sum_h P_{\theta}(v, h)$$

Distributed representations allow the topics "government", "corruption" and "oil" to combine to give very high probability to a word “Putin”.

- Reuters dataset
Multiple Application Domains

- Natural Images
- Text/Documents
- Collaborative Filtering / Matrix Factorization
- Video
- Motion Capture
- Speech Perception

Same learning algorithm --
multiple input domains.

Limitations on the types of structure that can be represented by a single layer of low-level features!
Talk Roadmap

Part 1: Deep Networks

• Restricted Boltzmann Machines: Learning low-level features.
• Deep Belief Networks: Learning Part-based Hierarchies.


• Deep Boltzmann Machines
• Learning Structured and Robust Models
• Multimodal Learning
Deep Belief Network

- Probabilistic Generative model.
- Contains multiple layers of nonlinear representation.
- Fast, greedy layer-wise pretraining algorithm.
- Inferring the states of the latent variables in highest layers is easy.
Deep Belief Network

Low-level features:
Edges

Built from **unlabeled** inputs.

(Hinton et.al. Neural Computation 2006)
Deep Belief Network

Higher-level features: Combination of edges

Low-level features: Edges

Built from unlabeled inputs.

Input: Pixels

Internal representations capture higher-order statistical structure

(Hinton et.al. Neural Computation 2006)
Deep Belief Network

- **Visible Layer (V)**
- **Hidden Layers (h^1, h^2, h^3)**
- **RBM**

Each layer is connected to the next layer through visible and hidden units. The network is characterized by its deep structure, enabling it to learn hierarchical representations.
Deep Belief Network

The joint probability distribution factorizes:

\[ P(v, h^1, h^2, h^3) = P(v|h^1)P(h^1|h^2)P(h^2, h^3) \]

\[ P(h^2, h^3) = \frac{1}{\mathcal{Z}(W^3)} \exp \left[ h^2^\top W^3 h^3 \right] \]

\[ P(h^1 = 1|h^2) = \frac{1}{1 + \exp \left( -\sum_k W^2_{jk} h^2_k \right)} \]

\[ P(v_i = 1|h^1) = \frac{1}{1 + \exp \left( -\sum_j W^1_{ij} h^1_j \right)} \]
Deep Belief Network

Approximate Inference

\[ Q(h^3|h^2) \]
\[ Q(h^2|h^1) \]
\[ Q(h^1|v) \]

Generative Process

\[ P(h^2, h^3) \]
\[ P(h^1|h^2) \]
\[ P(v|h^1) \]

\[ Q(h^t|h^{t-1}) = \prod_j \sigma \left( \sum_i W^t h_{i}^{t-1} \right) \]
\[ P(h^{t-1}|h^t) = \prod_j \sigma \left( \sum_i W^t h_{i}^t \right) \]
DBN Layer-wise Training

- Learn an RBM with an input layer \( v \) and a hidden layer \( h \).
DBN Layer-wise Training

- Learn an RBM with an input layer $v$ and a hidden layer $h$.
- Treat inferred values $Q(h^1|v) = P(h^1|v)$ as the data for training 2nd-layer RBM.
- Learn and freeze 2nd layer RBM.

$Q(h^1|v)$
DBN Layer-wise Training

- Learn an RBM with an input layer $v$ and a hidden layer $h$.

- Treat inferred values $Q(h^1|v) = P(h^1|v)$ as the data for training $2^{nd}$-layer RBM.

- Learn and freeze 2$^{nd}$ layer RBM.

- Proceed to the next layer.
DBN Layer-wise Training

- Learn an RBM with an input layer $v$ and a hidden layer $h$.
- Treat inferred values $Q(h^1|v) = P(h^1|v)$ as the data for training 2$^{nd}$-layer RBM.
- Learn and freeze 2$^{nd}$ layer RBM.
- Proceed to the next layer.

Layerwise pretraining improves variational lower bound
Why this Pre-training Works?

• Greedy pre-training improves variational lower bound!

• For any approximating distribution $Q(h^1|v)$

$$
\log P_\theta(v) = \sum_{h^1} P_\theta(v, h^1)
\geq \sum_{h^1} Q(h^1|v) \left[ \log P(h^1) + \log P(v|h^1) \right] + \mathcal{H}(Q(h^1|v))
$$
Why this Pre-training Works?

• Greedy training improves variational lower bound.

• RBM and 2-layer DBN are equivalent when $W^2 = W^1^\top$.

• The lower bound is tight and the log-likelihood improves by greedy training.

• For any approximating distribution $Q(h^1|v)$

$$
\log P_\theta(v) = \sum_{h^1} P_\theta(v, h^1) \\
\geq \sum_{h^1} Q(h^1|v) \left[ \log P(h^1) + \log P(v|h^1) \right] + \mathcal{H}(Q(h^1|v))
$$

Train 2\textsuperscript{nd}-layer RBM
Supervised Learning with DBNs

- If we have access to label information, we can train the joint generative model by maximizing the joint log-likelihood of data and labels

$$\log P(y, v)$$

- Discriminative fine-tuning:
  - Use DBN to initialize a multilayer neural network.
  - Maximize the conditional distribution:

$$\log P(y|v)$$
Sampling from DBNs

• To sample from the DBN model:

\[ P(v, h^1, h^2, h^3) = P(v|h^1)P(h^1|h^2)P(h^2, h^3) \]

• Sample \( h^2 \) using alternating Gibbs sampling from RBM.
• Sample lower layers using sigmoid belief network.
Learned Features

1\textsuperscript{st}-layer features

\[\text{Images of digit features from the 1\textsuperscript{st} layer.}\]

2\textsuperscript{nd}-layer features

\[\text{Images of digit features from the 2\textsuperscript{nd} layer.}\]
Learning Part-based Representation

Convolutional DBN

Faces
Groups of parts.

Object Parts

Trained on face images.

Lee et.al., ICML 2009
Learning Part-based Representation

Faces  Cars  Elephants  Chairs

Lee et al., ICML 2009
Learning Part-based Representation

Groups of parts.

Class-specific object parts

Trained from multiple classes (cars, faces, motorbikes, airplanes).

Lee et.al., ICML 2009
Deep Autoencoders
Deep Autoencoders

- We used 25x25 – 2000 – 1000 – 500 – 30 autoencoder to extract 30-D real-valued codes for Olivetti face patches.

- **Top**: Random samples from the test dataset.
- **Middle**: Reconstructions by the 30-dimensional deep autoencoder.
- **Bottom**: Reconstructions by the 30-dimentional PCA.
Information Retrieval

- The Reuters Corpus Volume II contains 804,414 newswire stories (randomly split into 402,207 training and 402,207 test).

- “Bag-of-words” representation: each article is represented as a vector containing the counts of the most frequently used 2000 words in the training set.

(Hinton and Salakhutdinov, Science 2006)
Semantic Hashing

• Learn to map documents into semantic 20-D binary codes.
• Retrieve similar documents stored at the nearby addresses with no search at all.

(Salakhutdinov and Hinton, SIGIR 2007)
Searching Large Image Database using Binary Codes

- Map images into binary codes for fast retrieval.

- Small Codes, Torralba, Fergus, Weiss, CVPR 2008
- Spectral Hashing, Y. Weiss, A. Torralba, R. Fergus, NIPS 2008
- Kulis and Darrell, NIPS 2009, Gong and Lazebnik, CVPR 2011
- Norouzi and Fleet, ICML 2011,
Talk Roadmap

Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.


- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning
Deep Boltzmann Machines

Low-level features:
Edges

Built from *unlabeled* inputs.

Input: Pixels
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

Image
DBNs vs. DBMs

Deep Belief Network

Deep Boltzmann Machine

DBNs are hybrid models:

- Inference in DBNs is problematic due to **explaining away**.
- Only greedy pretraining, **no joint optimization over all layers**.
- Approximate inference is feed-forward: **no bottom-up and top-down**.

Introduce a new class of models called Deep Boltzmann Machines.
Mathematical Formulation

\[ P_\theta(v) = \frac{P^*(v)}{\mathcal{Z}(\theta)} = \frac{1}{\mathcal{Z}(\theta)} \sum_{h^1, h^2, h^3} \exp \left[ v^T W^1 h^1 + h^1^T W^2 h^2 + h^2^T W^3 h^3 \right] \]

Deep Boltzmann Machine

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

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

\[ P(h^2_j = 1|h^1, h^3) = \sigma \left( \sum_k W^3_{kj} h^3_k + \sum_m W^2_{mj} h^1_m \right) \]

Unlike many existing feed-forward models: ConvNet (LeCun), HMAX (Poggio et.al.), Deep Belief Nets (Hinton et.al.)
Approximate Learning

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

(Approximate) Maximum Likelihood:

\[ \frac{\partial \log P_\theta(v)}{\partial W^1} = \mathbb{E}_{P_{data}} [vh^{1^T}] - \mathbb{E}_{P_\theta} [vh^{1^T}] \]

- Both expectations are intractable!

\[ P_{data}(v, h^{1}) = P_\theta(h^{1} | v) P_{data}(v) \]

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

Not factorial any more!
Approximate Learning

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

(Approximate) Maximum Likelihood:

\[ \frac{\partial \log P_\theta(v)}{\partial W^1} = \mathbb{E}_{P_{data}} [vh^{1^T}] - \mathbb{E}_{P_\theta} [vh^{1^T}] \]

Data

\[ P_{data}(v, h^1) = P_\theta(h^1 | v) P_{data}(v) \]

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

Not factorial any more!
Approximate Learning

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

(Approximate) Maximum Likelihood:

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

Variational Inference

Stochastic Approximation (MCMC-based)

Not factorial any more!
Many approaches for learning Boltzmann machines have been proposed over the last 20 years:

- Hinton and Sejnowski (1983),
- Peterson and Anderson (1987)
- Galland (1991)
- Kappen and Rodriguez (1998)
- Lawrence, Bishop, and Jordan (1998)
- Tanaka (1998)
- Welling and Hinton (2002)
- Welling and Teh (2003)
- Yasuda and Tanaka (2009)

Many of the previous approaches were not successful for learning general Boltzmann machines with hidden variables.

Real-world applications – thousands of hidden and observed variables with millions of parameters.

New Learning Algorithm

Posterior Inference

Conditional

Approximate conditional

\[ P_{data}(h|v) \]

Simulate from the Model

Approximate the joint distribution

\[ P_{model}(h, v) \]
New Learning Algorithm

Posterior Inference

Conditional

Approximate conditional

\[ P_{\text{data}}(h|v) \]

Data-dependent

\[ E_{P_{\text{data}}} [vh^\top] \]

Simulate from the Model

Approximate the joint distribution

\[ P_{\text{model}}(h, v) \]

Data-independent

\[ E_{P_{\text{model}}} [vh^\top] \]

Unconditional

Match

\[ \text{input} \]

Match
New Learning Algorithm

Posterior Inference

Conditional

Mean-Field

E_{P_{data}}[\mathbf{v}\mathbf{h}^\top]

Data-dependent: Variational Inference, mean-field theory

Key Idea of Our Approach:

Unconditional

Markov Chain Monte Carlo

E_{P_{model}}[\mathbf{v}\mathbf{h}^\top]

Data-independent: Stochastic Approximation, MCMC based
Variational Inference

(Approximate) Maximum Likelihood:

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

Variational Inference

Approximate intractable distribution \( P_\theta(h|v) \) with simpler, tractable distribution \( Q_\mu(h|v) \):

\[
\log P_\theta(v) \geq \log P_\theta(v) = \sum_h Q_\mu(h|v)
\]

\[
\text{KL}(Q||P) = \int Q(x) \log \frac{Q(x)}{P(x)} dx
\]

Variational Inference: Maximize the lower bound w.r.t. variational parameters \( \mu \).

Nonlinear fixed-point equations:

\[
\mu_k^{(2)} = \sigma \left( \sum_i W_{jk}^{2} \mu_j^{(1)} + \sum_m W_{km}^{3} \mu_m^{(3)} \right)
\]

\[
\mu_m^{(3)} = \sigma \left( \sum_k W_{km}^{3} \mu_k^{(2)} \right)
\]
Stochastic Approximation

Time \( t=1 \)

\[ \theta_1 \]

\[ x_1 \sim T_{\theta_1}(x_1 \leftarrow x_0) \]

Update \( \theta_1 \)

\[ h^1 \]

\[ v \]

\[ x_2 \sim T_{\theta_2}(x_2 \leftarrow x_1) \]

Update \( \theta_2 \)

\[ h^1 \]

\[ h^2 \]

\[ v \]

\[ x_3 \sim T_{\theta_3}(x_3 \leftarrow x_2) \]

Update \( \theta_3 \)

\[ h^1 \]

\[ h^2 \]

\[ v \]

Update \( \theta_t \) and \( x_t \) sequentially, where \( x = \{v, h^1, h^2\} \)

- Generate \( x_t \sim T_{\theta_t}(x_t \leftarrow x_{t-1}) \) by simulating from a Markov chain that leaves \( P_{\theta_t} \) invariant (e.g. Gibbs or M-H sampler)

- Update \( \theta_t \) by replacing intractable \( E_{P_{\theta_t}}[vh^\top] \) with a point estimate \( [v_t h_t^\top] \)

In practice we simulate several Markov chains in parallel.

L. Younes, Probability Theory 1989
Learning Algorithm

Update rule decomposes:

$$\theta_{t+1} = \theta_t + \alpha_t \left( \mathbb{E}_{P_{data}} [v h^\top] - \frac{1}{M} \sum_{m=1}^{M} v^{(m)} h^{(m)\top} \right) P_{\theta_t} [v h^\top]$$

- True gradient
- Perturbation term $\epsilon_t$

Almost sure convergence guarantees as learning rate $\alpha_t \to 0$

**Problem:** High-dimensional data: highly multimodal.

**Key insight:** The transition operator can be

**Fast Inference**

Learning can scale to millions of examples

Connections to the theory of stochastic approximation and adaptive MCMC.
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
Good Generative Model?

MNIST Handwritten Digit Dataset
## Handwriting Recognition

**MNIST Dataset**  
60,000 examples of 10 digits  

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>12.0%</td>
</tr>
<tr>
<td>K-NN</td>
<td>3.09%</td>
</tr>
<tr>
<td>Neural Net (Platt 2005)</td>
<td>1.53%</td>
</tr>
<tr>
<td>SVM (Decoste et.al. 2002)</td>
<td>1.40%</td>
</tr>
<tr>
<td>Deep Autoencoder (Bengio et. al. 2007)</td>
<td>1.40%</td>
</tr>
<tr>
<td>Deep Belief Net (Hinton et. al. 2006)</td>
<td>1.20%</td>
</tr>
<tr>
<td>DBM</td>
<td>0.95%</td>
</tr>
</tbody>
</table>

**Optical Character Recognition**  
42,152 examples of 26 English letters  

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>22.14%</td>
</tr>
<tr>
<td>K-NN</td>
<td>18.92%</td>
</tr>
<tr>
<td>Neural Net</td>
<td>14.62%</td>
</tr>
<tr>
<td>SVM (Larochelle et.al. 2009)</td>
<td>9.70%</td>
</tr>
<tr>
<td>Deep Autoencoder (Bengio et. al. 2007)</td>
<td>10.05%</td>
</tr>
<tr>
<td>Deep Belief Net (Larochelle et. al. 2009)</td>
<td>9.68%</td>
</tr>
<tr>
<td>DBM</td>
<td>8.40%</td>
</tr>
</tbody>
</table>

Permutation-invariant version.
Generative Model of 3-D Objects

24,000 examples, 5 object categories, 5 different objects within each category, 6 lightning conditions, 9 elevations, 18 azimuths.
3-D object Recognition

NORB Dataset: 24,000 examples

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logistic regression</td>
<td>22.5%</td>
</tr>
<tr>
<td>K-NN (LeCun 2004)</td>
<td>18.92%</td>
</tr>
<tr>
<td>SVM (Bengio &amp; LeCun 2007)</td>
<td>11.6%</td>
</tr>
<tr>
<td>Deep Belief Net (Nair &amp; Hinton 2009)</td>
<td>9.0%</td>
</tr>
<tr>
<td>DBM</td>
<td>7.2%</td>
</tr>
</tbody>
</table>

Pattern Completion
Learning Hierarchical Representations

Deep Boltzmann Machines:

Learning Hierarchical Structure in Features: combination of edges.

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

Need more structured and robust models
Talk Roadmap

Part 1: Deep Networks

• Restricted Boltzmann Machines: Learning low-level features.
• Deep Belief Networks: Learning Part-based Hierarchies.


• Deep Boltzmann Machines
• Learning Structured and Robust Models
• Multimodal Learning
Due to extreme illumination variations, deep models perform quite poorly on this dataset.
Deep Lambertian Model

Consider More Structured Models: undirected + directed models.

Combines the elegant properties of the Lambertian model with the Gaussian DBM model.

(Tang et. Al., ICML 2012, Tang et. al. CVPR 2012)
Lambertian Reflectance Model

- A simple model of the image formation process.

\[ I = a \times |\vec{\ell}| |\vec{n}| \cos(\theta) \]

- Albedo -- diffuse reflectivity of a surface, material dependent, illumination independent.

- Surface normal -- perpendicular to the tangent plane at a point on the surface.

- Images with different illumination can be generated by varying light directions
Deep Lambertian Model

\[ P(v|a, N, \ell) = \prod_{i \in \text{pixels}} \mathcal{N}(v_i|a_i(\hat{n}_i^\top \ell), \sigma^2) \]

\[ a \in \mathbb{R}^D, \quad N \in \mathbb{R}^{D \times 3}, \quad \ell \in \mathbb{R}^3 \]
Deep Lambertian Model

\[ P(v|a, N, \ell) = \prod_{i \in \text{pixels}} \mathcal{N}(v_i | a_i (\vec{n}_i^\top \ell), \sigma^2) \]

\[ a \in \mathbb{R}^D, \quad N \in \mathbb{R}^{D \times 3}, \quad \ell \in \mathbb{R}^3 \]

Inference: Variational Inference.
Learning: Stochastic Approximation
Yale B Extended Face Dataset

- 38 subjects, ~ 45 images of varying illuminations per subject, divided into 4 subsets of increasing illumination variations.
- 28 subjects for training, and 10 for testing.
Face Relighting

One Test Image

Observed
Inferred albedo

Face Relighting
Recognition Results

Recognition as function of the number of training images for 10 test subjects.

What about dealing with occlusions or structured noise?
Robust Boltzmann Machines

• Build more structured models that can deal with occlusions or structured noise.

\[ \log P(\tilde{v}, v, s, h, g) \sim \]
Robust Boltzmann Machines

• Build more structured models that can deal with occlusions or structured noise.

\[
\log P(\tilde{v}, v, s, h, g) \sim -\frac{1}{2} \sum_{i \in \text{pixels}} \frac{(v_i - b_i)^2}{\sigma_i^2} + v^\top W h
\]

- Gaussian RBM, modeling clean faces
- Binary RBM modeling occlusions

\[
-\frac{1}{2} \sum_{i \in \text{pixels}} \gamma_i s_i (v_i - \tilde{v}_i)^2
\]

- Binary pixel-wise Mask
- Gaussian noise model

(Tang et. Al., ICML 2012, Tang et. al. CVPR 2012)
Robust Boltzmann Machines

• Build more structured models that can deal with occlusions or structured noise.

\[
\log P(\tilde{v}, v, s, h, g) \sim -\frac{1}{2} \sum_{i \in \text{pixels}} \frac{(v_i - b_i)^2}{\sigma_i^2} + v^T Wh + s^T Ug
\]

Gaussian RBM, modeling clean faces

Binary RBM modeling occlusions

\[P(\tilde{v} \mid h, g)\text{ is a heavy-tailed distribution}\]

Binary pixel-wise Mask

Gaussian noise model

Inference: Variational Inference.

Learning: Stochastic Approximation
Recognition Results on AR Face Database

Inferred

Internal states of RoBM during learning.

# of iterations
Recognition Results on AR Face Database

Internal states of RoBM during learning.

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>Sunglasses</th>
<th>Scarf</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust BM</td>
<td>84.5%</td>
<td>80.7%</td>
</tr>
<tr>
<td>RBM</td>
<td>61.7%</td>
<td>32.9%</td>
</tr>
<tr>
<td>Eigenfaces</td>
<td>66.9%</td>
<td>38.6%</td>
</tr>
<tr>
<td>LDA</td>
<td>56.1%</td>
<td>27.0%</td>
</tr>
<tr>
<td>Pixel</td>
<td>51.3%</td>
<td>17.5%</td>
</tr>
</tbody>
</table>
Transfer Learning

How can we learn a novel concept – a high dimensional statistical object – from few examples.
Supervised Learning

Segway

Motorcycle

Test:
Transfer Learning

Background Knowledge

Millions of unlabeled images

Some labeled images

Bicycle
Dolphin
Elephant
Tractor

Learn to Transfer Knowledge

Learn novel concept from one example

Test:
What is this?
An Example

Structure in classes!

- Jaguar 1000
- Cheetah 3
- Leopard 1000
- Tiger 1000
- Car 1000
- Truck 3
- Tree

Slide Credit: Nitish Srivastava
Hierarchical-Deep Models
(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

**HD Models:** Integrate hierarchical Bayesian models with deep models.

**Hierarchical Bayes:**
- Learn **hierarchies of categories** for sharing abstract knowledge.

**Deep Models:**
- Learn **hierarchies of features**.
- **Unsupervised feature learning** – no need to rely on human-crafted input features.
Hierarchical-Deep Models
(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

- Tree hierarchy of classes is learned
- \( z \sim \text{nCRP} \) (Nested Chinese Restaurant Process) prior: a nonparametric prior over tree structures
- \( h^3 | z \sim \text{HDP} \) (Hierarchical Dirichlet Process) prior: a nonparametric prior allowing categories to share higher-level features, or parts.
- \( v | h^3 \sim \text{DBM} \) Deep Boltzmann Machine
  - Enforce approximate global consistency through many local constraints.
  - Incorporate prior knowledge to deal with occlusions, corrupted or missing data.
  - Images, Handwritten characters, Motion capture datasets.
CIFAR Object Recognition
(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

Tree hierarchy of classes is learned

Higher-level features

Lower-level generic features

Learned high-level features

DBM generic features

4 million Images
CIFAR Object Recognition
(Salakhutdinov, Tenenbaum, Torralba, NIPS 2011, PAMI 2013)

Tree hierarchy of classes is learned

Each image is made up of learned high-level features features.

Each higher-level feature is made up of lower-level features.

4 million Images
Learning Category Hierarchy

The model learns how to share the knowledge across many visual categories.

“aquatic animal” — dolphin, turtle, shark, ray  
“fruit” — apple, orange, pear  
“human” — girl, baby, man, woman  

Learned super-class hierarchy

Basic level class

Learned higher-level class-sensitive features

Learned low-level generic features
Given only 3 Examples

Willow Tree

Rocket

Generated Samples
Handwritten Character Recognition

Learned higher-level features

“alphabet 1”

“alphabet 2”

Handwritten Character Recognition

25,000 characters

Learned lower-level features

Edges
Simulating New Characters

Global

Super class 1

Super class 2

Class 1 Class 2

New class

Real data within super class

Simulated new characters
Simulating New Characters

Real data within super class

Simulated new characters
Simulating New Characters

Real data within super class

Simulated new characters
Simulating New Characters

Real data within super class

Simulated new characters
Simulating New Characters

Real data within super class

Simulated new characters
Simulating New Characters

Real data within super class

Simulated new characters
Motion Capture

Walk

Drunken Walk

Sexy Walk
The same model can be applied to speech, text, video, or any other high-dimensional data.
Talk Roadmap

Part 1: Deep Networks

- Restricted Boltzmann Machines: Learning low-level features.


- Deep Boltzmann Machines
- Learning Structured and Robust Models
- Multimodal Learning
Data – Collection of Modalities

- Multimedia content on the web - image + text + audio.
- Product recommendation systems.
- Robotics applications.

Modalities:
- Touch sensors
- Vision
- Audio
- Multimedia content on the web
- Product recommendation systems
- Robotics applications
Multi-Modal Input

• Improve Classification

  pentax, k10d, kangarooisland
  southaustralia, sa australia
  australiansealion 300mm

  SEA / NOT SEA

• Fill in Missing Modalities

  beach, sea, surf,
  strand, shore, wave,
  seascape, sand,
  ocean, waves

• Retrieve data from one modality when queried using data from another modality

  beach, sea, surf,
  strand, shore, wave,
  seascape, sand,
  ocean, waves
Building a Probabilistic Model

• Learn a joint density model:
  \[ P(h, v_{\text{image}}, v_{\text{text}}). \]

• \( h \): “fused” representation for classification, retrieval.

\[ P(h|v_{\text{image}}, v_{\text{text}}) \]

“Concept”

\[ h \]

\[ v_{\text{image}} \]

\[ v_{\text{text}} \]

sunrise, pacific ocean, baker beach, seashore, ocean
Building a Probabilistic Model

- Learn a joint density model:
  \[ P(h, v_{\text{image}}, v_{\text{text}}). \]

- \( h \): “fused” representation for classification, retrieval.

- Generate data from conditional distributions for
  - Image Annotation

\[ P(h, v_{\text{text}} \mid v_{\text{image}}) \]

“Concept”

\( h \)

\( v_{\text{image}} \)

\( v_{\text{text}} \)

Missing Data
Building a Probabilistic Model

• Learn a joint density model:
  \[ P(h, v_{\text{image}}, v_{\text{text}}) \].

• \( h \): “fused” representation for classification, retrieval.

• Generate data from conditional distributions for
  - Image Annotation
  - Image Retrieval

\[ P(h, v_{\text{image}} \mid v_{\text{text}}) \]

“Concept”

- sunset, pacificocean, bakerbeach, seashore, ocean

\[ v_{\text{image}} \]

\[ v_{\text{text}} \]
Challenges – I

Very different input representations

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

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

<table>
<thead>
<tr>
<th>Image</th>
<th>Text</th>
</tr>
</thead>
<tbody>
<tr>
<td>![Image](pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion)</td>
<td>pentax, k10d, pentaxda50200, kangarooisland, sa, australiansealion</td>
</tr>
<tr>
<td>![Image](mickikrimmel, mickipedia, headshot)</td>
<td>mickikrimmel, mickipedia, headshot</td>
</tr>
<tr>
<td><img src="%20no%20text" alt="Image" /></td>
<td>&lt; no text&gt;</td>
</tr>
<tr>
<td>![Image](unseulptpixel, naturey, crap)</td>
<td>unseulptpixel, naturey, crap</td>
</tr>
</tbody>
</table>

Noisy and missing data
<table>
<thead>
<tr>
<th>Image</th>
<th>Text</th>
<th>Text generated by the model</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="beach" /></td>
<td>pentax, k10, pentaxda50200, kangarooisland, sa, australianseal</td>
<td>beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves</td>
</tr>
<tr>
<td><img src="image2" alt="portrait" /></td>
<td>mickikrimmel, mickipedia, headshot</td>
<td>portrait, girl, woman, lady, blonde, pretty, gorgeous, expression, model</td>
</tr>
<tr>
<td><img src="image3" alt="no text" /></td>
<td>&lt; no text &gt;</td>
<td>night, notte, traffic, light, lights, parking, darkness, lowlight, nacht, glow</td>
</tr>
<tr>
<td><img src="image4" alt="nature" /></td>
<td>unseulpixel, naturey, crap</td>
<td>fall, autumn, trees, leaves, foliage, forest, woods, branches, path</td>
</tr>
</tbody>
</table>
A Simple Multimodal Model

- Use a joint binary hidden layer.
- **Problem**: Inputs have very different statistical properties.
- Difficult to learn cross-modal features.

![Diagram of multimodal model](image)
Multimodal DBM

Dense, real-valued image features

Gaussian model

Replicated Softmax

Word counts

$V_{\text{image}}$

$V_{\text{text}}$

(Srivastava & Salakhutdinov, NIPS 2012)
Multimodal DBM

Gaussian model

Dense, real-valued image features

V_{image}

h^1

Replicated Softmax

Word counts

V_{text}

(Srivastava & Salakhutdinov, NIPS 2012)
Multimodal DBM

Gaussian model

Dense, real-valued image features

$V_{\text{image}}$

$V_{\text{text}}$

(Srivastava & Salakhutdinov, NIPS 2012)
Dense, real-valued image features

Gaussian model

Bottom-up + Top-down

Replicated Softmax

Word counts

(V_{image} \rightarrow h^1 \rightarrow h^2 \rightarrow h^3 \rightarrow V_{text})

(Srivastava & Salakhutdinov, NIPS 2012)
Multimodal DBM

\[
P(v^m, v^t; \theta) = \frac{1}{Z(\theta, M)} \sum_h \exp \left( -\sum_i \frac{(v^m_i)^2}{2\sigma_i^2} + \sum_i \frac{v^m_i}{\sigma_i} W_{ij}^{(1m)} h^{(1m)}_j + \sum_j W_{jl}^{(2m)} h^{(1m)}_j h^{(2m)}_l \right) \\
+ \sum_{jk} W_{kj}^{(1t)} h^t_j v^m_k + \sum_{jl} W_{jl}^{(2t)} h^t_j h^{(2m)}_l + \sum_{lp} W^{(3t)} h^{(2m)}_l h^{(3)}_p + \sum_{lp} W^{(3m)} h^{(2m)}_l h^{(3)}_p
\]

Gaussian Image Pathway

Replicated Softmax Text Pathway

Joint 3rd Layer

(Srivastava & Salakhutdinov, NIPS 2012)
<table>
<thead>
<tr>
<th>Given</th>
<th>Generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog, cat, pet, kitten, puppy, ginger, tongue, kitty, dogs, furry</td>
<td>insect, butterfly, insects, bug, butterflies, lepidoptera</td>
</tr>
<tr>
<td>sea, france, boat, mer, beach, river, bretagne, plage, brittany</td>
<td>graffiti, streetart, stencil, sticker, urbanart, graff, sanfrancisco</td>
</tr>
<tr>
<td>portrait, child, kid, ritratto, kids, children, boy, cute, boys, italy</td>
<td>canada, nature, sunrise, ontario, fog, mist, bc, morning</td>
</tr>
<tr>
<td>Given</td>
<td>Generated</td>
</tr>
<tr>
<td>-------</td>
<td>-----------</td>
</tr>
<tr>
<td><img src="image1.png" alt="Lion" /></td>
<td>portrait, women, army, soldier, mother, postcard, soldiers</td>
</tr>
<tr>
<td><img src="image2.png" alt="Bird" /></td>
<td>obama, barackobama, election, politics, president, hope, change, sanfrancisco, convention, rally</td>
</tr>
<tr>
<td><img src="image3.png" alt="Sign" /></td>
<td>water, glass, beer, bottle, drink, wine, bubbles, splash, drops, drop</td>
</tr>
</tbody>
</table>
Images from Text

Given

water, red, sunset

nature, flower, red, green

blue, green, yellow, colors

chocolate, cake

Retrieved
MIR-Flickr Dataset

• 1 million images along with user-assigned tags.

Huiskes et. al.
Data and Architecture

\[ \approx 12 \text{ Million parameters} \]

- Image features: Gist, SIFT, MPEG-7 descriptors - 3857-dims.
- 200 most frequent tags.
- 25K labeled subset (15K training, 10K testing)
- 38 classes - sky, tree, baby, car, cloud …
## Results

- **Multimodal Inputs**

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>MAP</th>
<th>Precision@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.124</td>
<td>0.124</td>
</tr>
<tr>
<td>LDA [Huiskes et. al.]</td>
<td>0.492</td>
<td>0.754</td>
</tr>
<tr>
<td>SVM [Huiskes et. al.]</td>
<td>0.475</td>
<td>0.758</td>
</tr>
<tr>
<td>DBM-Labelled</td>
<td>0.526</td>
<td>0.791</td>
</tr>
</tbody>
</table>

Mean Average Precision

Similar Features, 15K labeled examples
## Results

- **Multimodal Inputs**

<table>
<thead>
<tr>
<th>Learning Algorithm</th>
<th>MAP</th>
<th>Precision@50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random</td>
<td>0.124</td>
<td>0.124</td>
</tr>
<tr>
<td>LDA [Huiskes et. al.]</td>
<td>0.492</td>
<td>0.754</td>
</tr>
<tr>
<td>SVM [Huiskes et. al.]</td>
<td>0.475</td>
<td>0.758</td>
</tr>
<tr>
<td>DBM-Labelled</td>
<td>0.526</td>
<td>0.791</td>
</tr>
<tr>
<td>DBM-Unlabelled+Dropout</td>
<td>0.641</td>
<td>0.888</td>
</tr>
<tr>
<td>MKL [Guillaumin et. al.]</td>
<td>0.623</td>
<td></td>
</tr>
</tbody>
</table>

- **Multiple Kernel Learning** uses 37,152 image features, compared to our model that uses 3,857 features.
Video and Audio

Cuave Dataset
Multi-Modal Models

Images

Video

Text & Language

Laser scans

Speech & Audio

Time series data

Develop learning systems that come closer to displaying human like intelligence

One of Key Challenges:

Inference
Summary

• Efficient learning algorithms for Hierarchical Models. Learning more adaptive, robust, and structured representations.

• Hierarchical models can improve 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 for learning RBMs, DBNs, and DBMs is available at:
http://www.utstat.toronto.edu/~rsalakhu/

Demo: http://deeplearning.cs.toronto.edu/
Generating Text from Images

Samples drawn after every 50 steps of Gibbs updates
Images from Text

Step 0
Sample drawn after every 50 steps of Gibbs sampling

Sample at step 0
Convolutinal Deep Models for Image Recognition

(Krizhevsky et. al., NIPS 2012)