Multimodal Learning with Deep Belief Nets

Nitish Srivastava

Joint work with Ruslan Salakhutdinov
Multimodal Data

Images from MIR FLickr dataset (Creative Commons Licence)
Utilizing multimodal data

- **Improve Classification**

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

- **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 density model of data

Restricted Boltzmann Machine

• Markov Random Field
• Binary visible random variables $v \in \{0, 1\}^D$ connected to binary hidden random variables $h \in \{0, 1\}^F$

$$
E(v, h; \theta) = - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j
$$

$$
P_\theta(v, h) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))
$$
Building a density model of data

Restricted Boltzmann Machine

- Markov Random Field
- Binary visible random variables $v \in \{0,1\}^D$ connected to binary hidden random variables $h \in \{0,1\}^F$

$$E(v, h; \theta) = -\sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$$P_\theta(v, h) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))$$

$$P_\theta(h|v) = \prod_j P_\theta(h_j|v)$$

$$P_\theta(v|h) = \prod_i P_\theta(v_i|h)$$
Building a density model of data

Restricted Boltzmann Machine

- Markov Random Field
- Binary visible random variables $v \in \{0, 1\}^D$ connected to binary hidden random variables $h \in \{0, 1\}^F$

$$E(v, h; \theta) = - \sum_{i,j} W_{ij} v_i h_j - \sum_i b_i v_i - \sum_j a_j h_j$$

$$P_\theta(v, h) = \frac{1}{Z(\theta)} \exp(-E(v, h; \theta))$$

$$P_\theta(h_j = 1|v) = \frac{1}{1 + \exp(-\sum_i W_{ij} v_i - a_j)}$$

$$P_\theta(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)}$$
Model Learning

Restricted Boltzmann Machine

• Maximum Likelihood Learning

\[ P_\theta(v) = \frac{1}{Z(\theta)} \sum_h \exp(v^\top W h + a^\top h + b^\top v) \]

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

\[ \frac{\partial L(\theta)}{\partial W_{ij}} = E_{P_{data}}[v_i h_j] - E_{P_\theta}[v_i h_j] \]
Model Learning

Restricted Boltzmann Machine

• Maximum Likelihood Learning

\[ P_\theta(v) = \frac{1}{Z(\theta)} \sum_h \exp(v^\top W h + a^\top h + b^\top v) \]

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

\[ \frac{\partial L(\theta)}{\partial W_{ij}} = E_{P_{data}}[v_i h_j] - E_{P_\theta}[v_i h_j] \]

• Approximation with MCMC methods
  - Contrastive Divergence: Run Markov chain for a few steps to sample \((v,h)\) from \(P_\theta\)

Computationally intractable, in general
Generalization of RBMs

- Binary inputs

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

\[ P_\theta(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)} \]
Generalization of RBMs

• Binary inputs

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

\[ P_\theta(v_i = 1|h) = \frac{1}{1 + \exp(- \sum_j W_{ij} h_j - b_i)} \]

• Real-valued inputs

\[ E(v, h; \theta) = - \sum_{i,j} W_{ij} \frac{v_i h_j}{\sigma_i} + \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j a_j h_j \]

\[ P_\theta(v_i|h) = \mathcal{N}(b_i + \sigma_i \sum_j W_{ij} h_j, \sigma_i^2) \]
Generalization of RBMs

• Binary inputs

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

\[ P_\theta(v_i = 1|h) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)} \]

• Real-valued inputs

\[ E(v, h; \theta) = - \sum_{i,j} W_{ij} \frac{v_i h_j}{\sigma_i} + \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j a_j h_j \]

\[ P_\theta(v_i|h) = \mathcal{N}(b_i + \sigma_i \sum_j W_{ij} h_j, \sigma_i^2) \]

• Discrete Multinomial inputs

\[ E(v, h; \theta) = - \sum_{i,j,k} W_{ij}^k v_i^k h_j - \sum_{i,k} v_i^k b_i^k - \sum_j a_j h_j \]

\[ P_\theta(v_i^k = 1|h) = \frac{\exp(b_i^k + \sum_j W_{ij}^k h_j)}{\sum_q \exp(b_i^q + \sum_j W_{ij}^q h_j)} \]
Generalization of RBMs

• Binary inputs

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

\[ P_\theta(v_i = 1|\mathbf{h}) = \frac{1}{1 + \exp(-\sum_j W_{ij} h_j - b_i)} \]

• Real-valued inputs

\[ E(\mathbf{v}, \mathbf{h}; \theta) = - \sum_{i,j} W_{ij} \frac{v_i h_j}{\sigma_i} + \sum_i \frac{(v_i - b_i)^2}{2\sigma_i^2} - \sum_j a_j h_j \]

\[ P_\theta(v_i|\mathbf{h}) = \mathcal{N}(b_i + \sigma_i \sum_j W_{ij} h_j, \sigma_i^2) \]

• Discrete Multinomial inputs

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

\[ P_\theta(v_i^k = 1|\mathbf{h}) = \frac{\exp(b_i^k + \sum_j W_{ij}^k h_j)}{\sum_q \exp(b_i^q + \sum_j W_{ij}^q h_j)} \]
Building a better density model of data

Deep Belief Network

\[ P_\theta(v, h^{(1)}, h^{(2)}) = P_{\theta_1}(v|h^{(1)}) P_{\theta_2}(h^{(1)}, h^{(2)}) \]

Sigmoid Belief Net
Restricted Boltzmann Machine
Building a better density model of data

Deep Belief Network

\[ P_{\theta}(v, h^{(1)}, h^{(2)}) = P_{\theta_1}(v|h^{(1)}) P_{\theta_2}(h^{(1)}, h^{(2)}) \]

Sigmoid Belief Net
Restricted Boltzmann Machine
Building a better density model of data

Deep Belief Network

$$P_{\theta}(v, h^{(1)}, h^{(2)}) = P_{\theta_1}(v|h^{(1)})P_{\theta_2}(h^{(1)}, h^{(2)})$$
Building a better density model of data

Deep Belief Network

$$P_{\theta}(v, h^{(1)}, h^{(2)}) = P_{\theta_1}(v|h^{(1)})P_{\theta_2}(h^{(1)}, h^{(2)})$$
Multimodal Deep Belief Network

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence
Multimodal Deep Belief Network

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence
Multimodal Deep Belief Network

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence
Multimodal Deep Belief Network

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence
Multimodal Deep Belief Network

sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence
Generating text conditioned on image
Generating text conditioned on image
Generating text conditioned on image
Generating text conditioned on image
Generating text conditioned on image

Gibbs Sampling
Generating text conditioned on image

Gibbs Sampling
Generating text conditioned on image

Gibbs Sampling
Generating text conditioned on image

beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves
## Results: Generating text from images

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Given Text</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>sandbanks, lake, lakeontario, sunset, walking, beach, purple, sky, water, clouds, overtheexcellence</td>
<td>sunset, twilight, strand, wave, breathtaking, horizon, shore, seascape, surf, scenery</td>
</tr>
<tr>
<td><img src="image2.png" alt="Image" /></td>
<td>&lt;no text&gt;</td>
<td>night, notte, traffic, light, lights, parking, darkness, lowlight, nacht, glow</td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>pentax, k10d, kangarooisland southaustralia, sa australia australiansealion 300mm</td>
<td>beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves</td>
</tr>
</tbody>
</table>

Images from MIR Flickr dataset (Creative Commons Licence)
Results: Generating text from images

<table>
<thead>
<tr>
<th>Input Image</th>
<th>Given Text</th>
<th>Generated Text</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Butterfly" /></td>
<td>top20butterflies</td>
<td>flower, pink, flowers, petals, petal, macroflowerlovers, flowerotica, floral, roses</td>
</tr>
<tr>
<td><img src="image2.png" alt="Lightpainting" /></td>
<td>camera, jahdakine, lightpainting, reflection doublepaneglass wowiekazowie</td>
<td>blue, art, artwork, artistic, surreal, expression, original, artist, gallery, patterns</td>
</tr>
<tr>
<td><img src="image3.png" alt="Portrait" /></td>
<td>mickikrimmel, mickipedia, headshot</td>
<td>portrait, girl, woman, lady, blonde, pretty, gorgeous, expression, model</td>
</tr>
</tbody>
</table>

Images from MIR FLickr dataset (Creative Commons Licence)
Retrieving images conditioned on text

Gibbs Sampling

Nearest Neighbour

nature hill scenery
green clouds
Results: Retrieving images from text

nature hill scenery
green clouds

bw black and white
noiretblanc biancoenero
blancoynegro

flower macro nature green
flowers petal petals bud

blue red art artwork
painted paint artistic
surreal gallery bleu

portrait girl
woman lady
blonde
Improving classification
Improving classification on unimodal input
Improving classification on unimodal input
Improving classification

Multimedia Information Retrieval (MIR) Flickr Dataset - contains 1 Million images along with text tags, 25K annotated

On multimodal inputs

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Discriminant Analysis</td>
<td>0.492</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>0.475</td>
</tr>
<tr>
<td>Multimodal Deep Belief Network</td>
<td>0.566</td>
</tr>
</tbody>
</table>
## Improving classification

Multimedia Information Retrieval (MIR) Flickr Dataset - contains 1 Million images along with text tags, 25K annotated

### On multimodal inputs

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Discriminant Analysis</td>
<td>0.492</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>0.475</td>
</tr>
<tr>
<td>Multimodal Deep Belief Network</td>
<td>0.566</td>
</tr>
</tbody>
</table>

### On unimodal inputs: Image only

<table>
<thead>
<tr>
<th>Method</th>
<th>Mean Average Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear Discriminant Analysis</td>
<td>0.315</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>0.375</td>
</tr>
<tr>
<td>Unimodal Deep Belief Network</td>
<td>0.413</td>
</tr>
<tr>
<td>Multimodal Deep Belief Network</td>
<td>0.492</td>
</tr>
</tbody>
</table>
Thank you!
More results: Generating text from images

Images from MIR Flickr dataset (Creative Commons Licence)
More results: Generating text from images

Images from MIR Flickr dataset (Creative Commons Licence)
Extra slides: Math

Product of Experts : Marginals

\[ P(v; \theta) = \frac{\exp(\sum_i a_i v_i)}{Z(\theta)} \prod_j \left( 1 + \exp \left( \sum_i v_i W_{ij} + b_j \right) \right) \]
People working in machine learning typically concern themselves with models that are designed to work with data coming from a single modality. Text, speech, vision.

However, multimodal data, a few examples of which are shown here, is ubiquitous in real-world applications. In this talk I am going to talk about how to build a joint probabilistic model over such data.

There are several ways in which multimodal data poses interesting problems. One task is to classify images, using text to do a better job of that.

If data in some modality is missing, we may want to infer it conditioned on other modalities. Alternatively, one may want to retrieve data.

Before, I describe my model in detail, a short introduction to Restricted Boltzmann because they form the building blocks for our Deep Belief Net.

An RBM is a Markov Random Field - an undirected graphical model, which has 2 sets of random variables v and h which are connected by edges as shown.

We define an Energy function on the state space of the ensemble of random variables and use it to define a probability distribution, in this case, a Boltzmann Distribution.

The nice thing about such bipartite connectivity is that it makes conditional distributions factorial making inference easy.

In particular, inferring v and h is simple using bernoulli distribution.

So basically, we a latent variable model of the data that defines a probability density function on the space of hidden and visible variables.

We try to learn the model parameters (weights and biases) max maximizing likelihood.

However, doing this exactly is not feasible in general and requires computing an expectation with respect to the distribution defined by the model.

So we use MCMC - approximate the expectation by drawing samples from the distribution and computing the required statistics from the sample.

Ideally, we should run a Gibbs sampler till convergence, however it is found that running it for a few steps is good enough.

Till now we talked about modeling distributions over binary vectors. We would like to use a similar setup to model distributions over real-valued inputs. We can define an energy function that looks like this.

This leads to a Gaussian conditional distribution like...

In order to model distributions over variables that can take discrete values, we can use an energy function as follows -

This gives a softmax distribution over visibles.

One way of improving the model is to learn higher-order features. There is strong evidence from both theoretical and applied work that adding depth helps.

This can be done by a Deep Belief Network. The joint distribution over hidden and visible variables has this conditional independence structure which is depicted by this graphical model.

Learning proceeds by first learning theta 1 as an RBM. Then fixing it and learning theta 2. It can be shown that doing this improves a lower-bound on the log-likelihood of the data.

From now on, we will use this structure to depict a deep belieif net.