Deep Learning – Autoencoder Models

INTELLIGENT SYSTEMS FOR PATTERN RECOGNITION (ISPR)
DAVIDE BACCIU – DIPARTIMENTO DI INFORMATICA - UNIVERSITA’ DI PISA
DAVIDE.BACCIU@UNIPI.IT
Lecture Outline

Autoencoders a.k.a. The first and the latest deep learning model

- Autoencoders and dimensionality reduction
- Deep neural autoencoders
  - Sparse
  - Denoising
  - Contractive
- Deep generative-based autoencoders
  - Deep Belief Networks
  - Deep Boltzmann Machines
- Application Examples
Basic Autoencoder (AE)

- Train a model to **reconstruct the input**
- Passing through some form of information bottleneck
  - $K \ll D$, or?
  - $h$ sparsely active
- Train by loss minimization
  $$L(x, \tilde{x}) = L(x, g(f(x)))$$
Neural Autoencoders

Generally, we would like to train nonlinear AEs, with possibly $K > D$, that do not learn trivial identity.

- Regularized autoencoders
  - Sparse AE
  - Denoising AE
  - Contractive AE
- Autoencoders with dropout layers
Sparse Autoencoder

Add a term to the cost function to penalize $h$ (want the number of active units to be small)

$$J_{SAE}(\theta) = \sum_{x \in S} (L(x, \tilde{x}) + \lambda \Omega(h))$$

Typically

$$\Omega(h) = \Omega(f(x)) = \sum_{j} |h_j(x))|$$
Probabilistic Interpretation (Oh No, Again!)

Training with regularization is MAP inference

$$\max \log P(h, x) = \max (\log P(x|h) + \log P(h))$$

$$P(h) = \frac{\lambda}{2} \exp\left(-\frac{\lambda}{2} |h|_1\right)$$

$$\Omega(h) = \lambda ||h||_1$$

Likelihood

Prior

Laplace
Denoising Autoencoder (DAE)

Train the AE to minimize the function

$$L(x, g(f(\hat{x})))$$

where $\hat{x}$ is a version of original input $x$ corrupted by some noise process $C(\hat{x}|x)$

Key Intuition - Learned representations should be robust to partial destruction of the input
Another Interpretation...

...yes, exactly the one you are thinking of

Learns the **denoising distribution**

\[ P(x|\hat{x}) \]

By minimizing

\[ -\log P_d(x|h = f(\hat{x})) \]
DAE as Manifold Learning

Learning a vector field (green arrows) approximating the gradient of the unknown data generating distribution

\[ g(h) - x \propto \frac{\partial \log p(x)}{\partial x} \]

\[ C(\hat{x}|x) = N(\hat{x}|x, \sigma^2) \]
The Manifold Assumption

Assume data lies on a lower dimensional non-linear manifold since variables in data are typically dependent.

Regularized AE can afford to represent only variations that are needed to reconstruct training examples.

AE mapping is sensitive only to changes in manifold direction.

Contractive Autoencoder

Penalize **encoding function** for input sensitivity

$$J_{CAE}(\theta) = \sum_{x \in S} (L(x, \tilde{x}) + \lambda \Omega(h))$$

$$\Omega(h) = \Omega(f(x)) = \left\| \frac{\partial f(x)}{\partial x} \right\|_F$$

You can as well penalize on higher order derivatives
Deep Autoencoder (AE)

- Unsupervised training
- Hierarchical autoencoder
- Extracts a representation of inputs that facilitates
  - Data visualization, exploration, indexing,...
  - Realization of a supervised task
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE

Any form of AE, e.g. those shown in previous slides
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE
Unsupervised Layerwise Pretraining

Incremental unsupervised construction of the Deep AE
Optional Fine Tuning

Fine tune the whole autoencoder to optimize input reconstruction.
You can use backpropagation, but it remains an unsupervised task.
Rearranging the Graphics

Does it look like something familiar?

A layered Restricted Boltzmann Machine

Can use RBM to perform layerwise pretraining and learn the matrices $W_i$
深信念网络（DBN）

一个成对的RBM堆叠

重要通知

DBM是一个深的自动编码器，但不是深的RBM

它是（主要是）有向的！
Deep Boltzmann Machine (DBM)

How do we get this?

Training requires some attention because of the recurrent interactions from higher layers to the bottom

\[
P(h_j^1 | x, h^2) = \sigma \left( \sum_i W_{ij}^1 x_i + \sum_m W_{jm}^2 h_m^2 \right)
\]

\[
P(x_i | h^1) = \sigma \left( \sum_j W_{ij}^1 h_j^1 \right)
\]
Pretraining DBM

How do we get this?

1) (Pre)training the first layer entails fitting this model

\[ P(h^1 \mid W^2) = \sum_{h^2} P(h^1, h^2 \mid W^2) \]

When putting things together, we need to average between the two

\[ P(h^1 \mid W^1) = \sum_{x} P(h^1, x \mid W^1) \]

2) (Pre)training the second layer changes \( h^1 \) prior by

\[ P(x \mid \theta) = \sum_{h^1} P(h^1 \mid W^1)P(x \mid h^1, W^1) \]
Pretraining DBM - Trick

Averaging the two models of $h^1$ can be approximated by taking half contribution from $W^1$ and half from $W^2$

- Using full $W^1$ and $W^2$ would double count $x$ contribution as $h^2$ depends on $x$

When training with more than two RBMs apply trick to first and last layers and halve weights (both direction) of intermediate RBM
The pretrained DBM matrices can be used to **initialize a deep autoencoder**

- Add input from $h^2$ to the first hidden layer
- Add output layer
- **Fine tuning** of the RBM matrices by backpropagation
Software - Deep Neural Autoencoders

- All deep learning frameworks offer facilities to build (deep) AEs
- Check out classic Theano-based tutorials for denoising autoencoders and their stacked version
- A variety of deep AE in Keras and their counterpart in Torch (plus a selection in Pytorch)
- Stacked autoencoders built with official Matlab toolbox functions
Matlab - Deep Generative Models

- **Matlab code** for the DBN paper with a demo on MNIST data
- **Matlab code** for Deep Boltzmann Machines with a demo on MNIST data
- **Deepmat** – Matlab library for deep generative models
- **DeeBNet** – Matlab/Octave toolbox for deep generative models with GPU support
Python - Deep Generative Models

- DBN and DBM implementations exist for all major deep learning libraries

- Deep Boltzmann machine implementation (Tensorflow-based) with image processing application, pre-trained networks and notebooks

- Deepnet – A Toronto based implementation of deep autoencoders (neural and generative)

- Check out classic Theano-based tutorials for deep belief networks and RBM
Visualizing complex data in learned latent space
Visualizing Sound

laughter
rustle
guitar
cat
helicopter
water_tap
child
speech
AE Applications – Image Restoration/Colorization

Apply autoencoder construction with advanced building blocks (e.g. CNN layers)
Multimodal DBM

Modality fusion layers

Modality 1

Modality K

N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014
## Multimodal DBM – Image and Text

<table>
<thead>
<tr>
<th>Image</th>
<th>Given Tags</th>
<th>Generated Tags</th>
<th>Input Tags</th>
<th>Nearest neighbors to generated image features</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Image" /></td>
<td>pentax, k10d, kangarooisland, southaustralia, sa, 300mm, australia, australiasealion</td>
<td>beach, sea, surf, strand, shore, wave, seascape, sand, ocean, waves</td>
<td>nature, hill, scenery, green, clouds</td>
<td><img src="image2.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Image" /></td>
<td>&lt; no text &gt;</td>
<td>night, lights, christmas, nightshot, nacht, nuit, notte, longexposure, noche, nocturna</td>
<td>flower, nature, green, flowers, petal, petals, bud</td>
<td><img src="image4.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image5.png" alt="Image" /></td>
<td>aberam, 0505, sarahc, moo</td>
<td>portrait, bw, blackandwhite, people, faces, girl, blackwhite, person, man</td>
<td>blue, red, art, artwork, painted, paint, artistic, surreal, gallery, bleu</td>
<td><img src="image6.png" alt="Image" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Image" /></td>
<td>unselpixel, naturey crap</td>
<td>fall, autumn, trees, leaves, foliage, forest, woods, branches, path</td>
<td>bw, blackandwhite, noiretblanc, bianconero, blanconegro</td>
<td><img src="image8.png" alt="Image" /></td>
</tr>
</tbody>
</table>

N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014
Multimodal DBM – Sampling

<table>
<thead>
<tr>
<th>Step 50</th>
<th>Step 100</th>
<th>Step 150</th>
<th>Step 200</th>
<th>Step 250</th>
</tr>
</thead>
<tbody>
<tr>
<td>travel</td>
<td>beach</td>
<td>sea</td>
<td>water</td>
<td>italy</td>
</tr>
<tr>
<td>trip</td>
<td>ocean</td>
<td>beach</td>
<td>canada</td>
<td>water</td>
</tr>
<tr>
<td>vacation</td>
<td>waves</td>
<td>island</td>
<td>bc</td>
<td>sea</td>
</tr>
<tr>
<td>africa</td>
<td>sea</td>
<td>vacation</td>
<td>britishcolumbia</td>
<td>boat</td>
</tr>
<tr>
<td>earthasia</td>
<td>sand</td>
<td>travel</td>
<td>reflection</td>
<td>italia</td>
</tr>
<tr>
<td>asia</td>
<td>nikon</td>
<td>ocean</td>
<td>alberta</td>
<td>mare</td>
</tr>
<tr>
<td>men</td>
<td>surf</td>
<td>caribbean</td>
<td>lake</td>
<td>venizia</td>
</tr>
<tr>
<td>2007</td>
<td>rocks</td>
<td>tropical</td>
<td>quebec</td>
<td>acqua</td>
</tr>
<tr>
<td>india</td>
<td>coast</td>
<td>resort</td>
<td>ontario</td>
<td>ocean</td>
</tr>
<tr>
<td>tourism</td>
<td>shore</td>
<td>trip</td>
<td>ice</td>
<td>venice</td>
</tr>
</tbody>
</table>

**Input tags**

<table>
<thead>
<tr>
<th>purple, flowers</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1.png" alt="Flowers" /></td>
</tr>
<tr>
<td><img src="image2.png" alt="Flowers" /></td>
</tr>
<tr>
<td><img src="image3.png" alt="Flowers" /></td>
</tr>
<tr>
<td><img src="image4.png" alt="Flowers" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>car, automobile</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5.png" alt="Car" /></td>
</tr>
<tr>
<td><img src="image6.png" alt="Car" /></td>
</tr>
<tr>
<td><img src="image7.png" alt="Car" /></td>
</tr>
<tr>
<td><img src="image8.png" alt="Car" /></td>
</tr>
</tbody>
</table>

N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014
Multimodal DBM – Multimodal Quering

<table>
<thead>
<tr>
<th>Multimodal Query</th>
<th>Top 4 retrieved results</th>
</tr>
</thead>
<tbody>
<tr>
<td>hongkong, causewaybay, shoppingcentre, building, mall</td>
<td>howell, bridge, genesee, river, rochester, downtown, building</td>
</tr>
<tr>
<td>me, myself, eyes, blue, hair</td>
<td>urban, me, abigfave, fiveflickrfavs</td>
</tr>
</tbody>
</table>

N. Srivastava, R. Salakhutdinov, Multimodal Learning with Deep Boltzmann Machines, JMLR 2014
Anomaly Detection

Statistics on the «normal» differences between \( x \) and \( \tilde{x} \)

Threshold
Take Home Messages

○ Regularized autoencoder
  ● Optimize reconstruction quality
  ● Constrain stored information

○ Autoencoder training is manifold learning
  ● Learn a latent space manifold where input data resides
  ● Store only variations that are useful to represent training data

○ Autoencoders learn a (conditional) distribution of input data \( P(\hat{x} \mid \ldots) \)

○ Deep AE: pretraining, fine tuning, supervised optimization

○ Use AE for finding new/useful data representations
  ● Or to learn its distribution
Next Lecture

Gated Recurrent Networks
- Learning with sequential data
- Gradient issues
- Gated RNN
  - Long-Short Term Memories (LSTM)
  - Gated Recurrent Units (GRU)
- Advanced topics
  - Understanding and exploiting memory encoding
  - Applications

PART I