Outline

1. Introduction
   - Feature Learning
   - Correspondence in Computer Vision
   - Relational feature learning

2. Learning relational features
   - Sparse Coding Review
   - Encoding relations
   - Inference
   - Learning

3. Factorization, eigen-spaces and complex cells
   - Factorization
   - Eigen-spaces, energy models, complex cells

4. Applications
   - Applications
   - Conclusions
1 Introduction
- Feature Learning
- Correspondence in Computer Vision
- Relational feature learning

2 Learning relational features
- Sparse Coding Review
- Encoding relations
- Inference
- Learning

3 Factorization, eigen-spaces and complex cells
- Factorization
- Eigen-spaces, energy models, complex cells

4 Applications
- Applications
- Conclusions
Bag-Of-Warps
Collapsing all hidden representations at monocular SIFT keypoints (across all keypoints and time frames) and performing logistic regression yields 80.56% correct.
Convolutional GBM (Taylor et al., 2010):

<table>
<thead>
<tr>
<th>Prior Art</th>
<th>Accuracy</th>
<th>Convolutional architectures</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG3D-KM-SVM</td>
<td>85.3</td>
<td>32convGRBM_{16x16} -128F_{9x9} -R/N/A_{4x4} -log_reg</td>
<td>88.9</td>
</tr>
<tr>
<td>HOG/HOF-KM-SVM</td>
<td>86.1</td>
<td>32convGRBM_{16x16} -128F_{9x9} -R/N/A_{4x4} -mlp</td>
<td>90.0</td>
</tr>
<tr>
<td>HOG-KM-SVM</td>
<td>79.0</td>
<td>32F_{16x16} -R/N/A_{4x4} -128F_{9x9} -R/N/A_{4x4} -log_reg</td>
<td>79.4</td>
</tr>
<tr>
<td>HOF-KM-SVM</td>
<td>88.0</td>
<td>32F_{16x16} -R/N/A_{4x4} -128F_{9x9} -R/N/A_{4x4} -mlp</td>
<td>79.5</td>
</tr>
</tbody>
</table>

Convolutional GBM on Hollywood2:

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prior Art [27]:</td>
<td></td>
</tr>
<tr>
<td>HOG3D+KM+SVM</td>
<td>45.3</td>
</tr>
<tr>
<td>HOG/HOF+KM+SVM</td>
<td><strong>47.4</strong></td>
</tr>
<tr>
<td>HOG+KM+SVM</td>
<td>39.4</td>
</tr>
<tr>
<td>HOF+KM+SVM</td>
<td>45.5</td>
</tr>
<tr>
<td>convGRBM+SC+SVM</td>
<td><strong>46.6</strong></td>
</tr>
</tbody>
</table>
(Le, et al., 2011)

Velocity tuning of the higher-order features:

![Graph showing velocity tuning of higher-order features.](image-url)
ISA applied to action recognition

(Le, et al., 2011)

<table>
<thead>
<tr>
<th></th>
<th>KTH</th>
<th>Hollywood2</th>
<th>UCF</th>
<th>YouTube</th>
</tr>
</thead>
<tbody>
<tr>
<td>until 2011</td>
<td>92.1</td>
<td>50.9</td>
<td>85.6</td>
<td>71.2</td>
</tr>
<tr>
<td>hierarchical ISA</td>
<td>93.9</td>
<td>53.3</td>
<td>86.5</td>
<td>75.8</td>
</tr>
</tbody>
</table>
Analogy making

$A : A' :: B : ?$

1. Infer transformation from source images $x_{source}, y_{source}$:
   
   $z(x_{source}, y_{source})$

2. Apply the transformation to target image $x_{target}$:

   $y(z, x_{target})$
Filters learned from transforming faces:

- Filters learned from faces:
Learning a gated Boltzmann machine on changing facial expressions.
(Susskind, et al., 2011)
**Joint density training** allows for comparing compatibilities of pairs.
<table>
<thead>
<tr>
<th>Model/Task</th>
<th>TFD ID</th>
<th>TFD Exp</th>
<th>PUBFIG ID</th>
<th>AFFINE</th>
</tr>
</thead>
<tbody>
<tr>
<td>cosine</td>
<td>0.848</td>
<td>0.663</td>
<td>0.649</td>
<td>0.721</td>
</tr>
<tr>
<td>RBM</td>
<td>0.869</td>
<td>0.656</td>
<td>0.647</td>
<td>0.799</td>
</tr>
<tr>
<td>conditional</td>
<td>0.805</td>
<td>0.634</td>
<td>0.557</td>
<td>0.825</td>
</tr>
<tr>
<td>bilinear</td>
<td>0.905</td>
<td>0.637</td>
<td><strong>0.774</strong></td>
<td>0.812</td>
</tr>
<tr>
<td>3-way</td>
<td>0.932</td>
<td><strong>0.705</strong></td>
<td>0.771</td>
<td>0.930</td>
</tr>
<tr>
<td>3-way symm</td>
<td><strong>0.951</strong></td>
<td>0.695</td>
<td>0.762</td>
<td><strong>0.931</strong></td>
</tr>
</tbody>
</table>
Bi-linear classification

- Special case of a gated Boltzmann machine:
- Replace the output-“image” by a one-hot-encoded class-label.
- This is a classifier, where each label can blend in its own model!
Bi-linear classification

Marginalization is tractable in closed form

$$p(y|\mathbf{x}) = \sum_z p(y, z|\mathbf{x}) \propto \sum_z \exp(\mathbf{x}^t w_y z) = \sum_z \exp(\sum_{ik} w_{yik} x_i h_k)$$

$$= \prod_k (1 + \exp(\sum_i w_{yik} x_i))$$

It is also equivalent to a mixture of $2^K$ logistic regressors (Nair, 2008), (Memisevic, et al.; 2010), (Warrell et al.; 2010)
We can factorize parameters like before. This allows classes to share features. The activity of a factor, \( f \), given class \( j \), is now exactly equal to the parameter value \( w_{jf} \). Thus the weights can be thought of as the responses of **virtual class-templates**.
Rotated digit classification

- Data-set from the “deep learning-challenge” [Larochelle et al., 2007] like before.
- Learned rotation-invariant filters:
Deep Learning challenge (Larochelle et al., 2008).

<table>
<thead>
<tr>
<th>dataset/model:</th>
<th>SVMs</th>
<th>NNet</th>
<th>RBM</th>
<th>DEEP</th>
<th>GSM</th>
</tr>
</thead>
<tbody>
<tr>
<td>rectangles</td>
<td>SVMRBF 2.15</td>
<td>NNet 7.16</td>
<td>DBN1 4.71</td>
<td>DBN3 2.60</td>
<td>SAA3 2.41</td>
</tr>
<tr>
<td>rect.-images</td>
<td>SVMPOL 2.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mnistplain</td>
<td>3.03</td>
<td>3.69</td>
<td>4.69</td>
<td>3.94</td>
<td>3.11</td>
</tr>
<tr>
<td>convexshapes</td>
<td>19.13</td>
<td>19.82</td>
<td>32.25</td>
<td>19.92</td>
<td>18.63</td>
</tr>
<tr>
<td>mnistbackrand</td>
<td>14.58</td>
<td>16.62</td>
<td>20.04</td>
<td>9.80</td>
<td>6.73</td>
</tr>
<tr>
<td>mnistbackimg</td>
<td>22.61</td>
<td>24.01</td>
<td>27.41</td>
<td>16.15</td>
<td>16.31</td>
</tr>
<tr>
<td>mnistrotbackimg</td>
<td>55.18</td>
<td>56.41</td>
<td>62.16</td>
<td>52.21</td>
<td>47.39</td>
</tr>
<tr>
<td>mnistrot</td>
<td>11.11</td>
<td>15.42</td>
<td>18.11</td>
<td>14.69</td>
<td>10.30</td>
</tr>
</tbody>
</table>
To train energy models on *single images*:
- Plug in the same image left and right.
- Hiddens will model pixel covariance matrices.
- Eg., (Ranzato et al., 2010), (Karklin, Lewicki; 2008)
- Training can be finicky.
Extension to *less* than two frames

To train energy models on *single images*:
- Plug in the same image left and right.
- Hiddens will model pixel covariance matrices.
- Eg., (Ranzato et al., 2010), (Karklin, Lewicki; 2008)
- Training can be finicky. Use a relational auto-encoder.
We can combine this with standard hidden units in one model.
The combination tends to work better recognition (Ranzato et al., 2010).
The vanilla hidden units then plays the role of “higher-order-biases” (Memisevic, 2007).
Learning higher-order within-image structure has been suggested to address the fact that ICA does not really yield independent components...

- Add layers to model correlations of filter responses.
- Closely related to Deep Learning.
Some within image covariance and mean filters
Within-image correlations

- (Karklin, Lewicki; 2008), (Osindero et al., 2006), ...
- ISA itself used mainly for modeling within-image structure.
- (Ranzato et al., 2010) suggest combining covariance features and traditional “mean” features, for example to generate images with an MRF:
mcRBMs on TIMIT

- mcRBM applied to speech recognition (phones, speaker independent, TIMIT)
- (Dahl, et al.; 2010)
Hidden variables make extracting multiple, simultaneous motions easy.

When they fail they do so in a similar way as humans:

- Better discrimination at large angles, averaging at very small angles, “motion repulsion”.
- (e.g., Treue et al., 2000)
Depth as a latent variable

- Learning a dictionary for stereo:
- Generate left-right camera pairs with known disparities.
- *Predict* disparity from the hidden units.
- This gives rise to a three-layer network, that may be trained with Hebbian-like learning.

Rectified

Not rectified
Hiddens learn to encode disparities

Can use this to encode 3d-structure implicitly, for example, for multi-view recognition.
Norb stereo features

<table>
<thead>
<tr>
<th></th>
<th>NORB training subset:</th>
<th>NORB testset:</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBMmon</td>
<td>73.65</td>
<td>63.28</td>
</tr>
<tr>
<td>RBMbin</td>
<td>60.43</td>
<td>38.91</td>
</tr>
<tr>
<td>cc</td>
<td>34.85</td>
<td>36.80</td>
</tr>
<tr>
<td>cc+bin</td>
<td>31.48</td>
<td></td>
</tr>
</tbody>
</table>

Roland Memisevic (Uni Frankfurt)
Multiview Feature Learning
Tutorial at CVPR 2012
Transformations are transformation invariant.
The 2-D subspace projections, however, are at the same time affected by the aperture problem, so they are selective to other sources of variability, including object ID!
We can use the aperture effect to build invariant features:
Harnessing the aperture problem
Harnessing the aperture problem
Harnessing the aperture problem

$x_{i}\cdots x_{j}$
Harnessing the aperture problem
Harnessing the aperture problem

pose-independent, content-independent

pose-independent, content-dependent
Rotation “quadrature” filters
Rotation “quadrature” filters
Representing digits using rotation aperture features

- Learn rotation features. Represent digits using aperture features.
- No video available? Fill video buffer with copies of the same image: Represent the non-transformation.
Rotated MNIST error rates
Humans do not recognize still images but videos of objects. The way in which an object changes can convey useful information about the object, including 3-D structure.

→ **Learn features from videos not still images.** For example, (Lee and Soatto, 2011).
The “norbjects” video dataset
3-D rotation subspaces
3-D rotation subspaces
3-D rotation subspaces
3-D rotation subspaces
3-D rotation subspaces
3-D rotation subspaces
"Harnessing the aperture problem"
(Taylor, Hinton; 2009), (Taylor, et al.; 2010)

Learning models on mocap instead of images makes it possible to model motion style and to perform tracking.
More Tracking

(Bazzani et al.), (Larochelle, Hinton, 2011)
1 Introduction
- Feature Learning
- Correspondence in Computer Vision
- Relational feature learning

2 Learning relational features
- Sparse Coding Review
- Encoding relations
- Inference
- Learning

3 Factorization, eigen-spaces and complex cells
- Factorization
- Eigen-spaces, energy models, complex cells

4 Applications
- Applications
- Conclusions
Learning is a way to support simplicity and homogeneity of complex, intelligent systems.

Feature learning even more so.

Relational feature learning even more:

Learning “verbs”, not just “nouns”, can help address more tasks with a single kind of model.

This seems like a very good reason to have complex cells.

One reason, why looking for correspondences – across frames, across views, across modalities, etc. – is a common operation, is that mappings between modalities are often one-to-many.

The theory provides a strong inductive bias for products and/or squaring non-linearities when building deep learning models.
Thank you

More info, code, links, etc. at