Topics in AI (CPSC 532S): Multimodal Learning with Vision, Language and Sound

Lecture 14: Unsupervised Learning, Autoencoders [Part 2]
Logistics

- **Assignment 1 & 2** grades are posted

- **Assignment 3** handed in, solutions are give

- **Assignment 4** is out (due last day before the break) — Do Part 1!

- **Project pitches next week**
  9 groups per class (~8 minutes / group, 5-6 min presentation + questions)
Review — Autoencoders

Self (i.e. self-encoding)

— Feed forward network intended to reproduce the input
— Encoder/Decoder architecture
  Encoder: $f = \sigma(Wx)$
  Decoder: $g = \sigma(W'h)$
— Score function

$$x' = f(g(x))$$

$$\mathcal{L}(x', x)$$

*slide from Louis-Philippe Morency*
**Review — De-noising Autoencoder**

**Idea:** add noise to input but learn to reconstruct the original

- Leads to better representations
- Prevents copying

**Note:** different noise is added during each epoch
Review — Context Encoders

[Pathak et al., 2016]
Spatial Context Networks

[ Wu, Sigal, Davis, 2017 ]
Spatial Context Networks

<table>
<thead>
<tr>
<th>Method</th>
<th>Initialization</th>
<th>Supervision</th>
<th>Pretraining time</th>
<th>Classification</th>
<th>Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Gaussian</td>
<td>random</td>
<td>N/A</td>
<td>&lt; 1 minute</td>
<td>53.3</td>
<td>43.4</td>
</tr>
<tr>
<td>Wang et al. [32]</td>
<td>random</td>
<td>motion</td>
<td>1 week</td>
<td>58.4</td>
<td>44.0</td>
</tr>
<tr>
<td>Doersch et al. [3]</td>
<td>random</td>
<td>context</td>
<td>4 weeks</td>
<td>55.3</td>
<td>46.6</td>
</tr>
<tr>
<td>*Doersch et al. [3]</td>
<td>1000 class labels</td>
<td>context</td>
<td>–</td>
<td>65.4</td>
<td>50.4</td>
</tr>
<tr>
<td>Pathak et al. [21]</td>
<td>random</td>
<td>context inpainting</td>
<td>14 hours</td>
<td>56.5</td>
<td>44.5</td>
</tr>
<tr>
<td>Zhang et al. [36]</td>
<td>random</td>
<td>color</td>
<td>–</td>
<td>65.6</td>
<td>46.9</td>
</tr>
<tr>
<td>ImageNet [21]</td>
<td>random</td>
<td>1000 class labels</td>
<td>3 days</td>
<td>78.2</td>
<td>56.8</td>
</tr>
<tr>
<td>*ImageNet</td>
<td>random</td>
<td>1000 class labels</td>
<td>3 days</td>
<td>76.9</td>
<td>58.7</td>
</tr>
<tr>
<td>SCN-EdgeBox</td>
<td>1000 class labels</td>
<td>context</td>
<td>10 hours</td>
<td>79.0</td>
<td>59.4</td>
</tr>
</tbody>
</table>

[Wu, Sigal, Davis, 2017]
A Little Theory: **Information Bottleneck** [Tishbi et al., 1999]

Every layer could be treated as a random variable, then entire network is a Markov Chain

**Data processing theorem:** if the only connection between $X$ and $Y$ is through $T$, the information that $Y$ gives about $X$ cannot be bigger than the information that $T$ gives about $X$.

\[ I(X;Y) \leq I(T_1;Y) \leq I(T_2;Y) \leq \cdots \leq I(Y;Y) \]
A Little Theory: **Information Bottleneck** [Tishbi et al., 1999]

**Observation:** In the information plane layers first increase the mutual information between themselves and the output and then reduce information between themselves and the input (which leads to “forgetting” of irrelevant inputs and ultimately generalization)
A Little Theory: **Information Bottleneck** [Tishbi et al., 1999]

50 networks of same topology being optimized
A Little Theory: **Information Bottleneck**  
[Tishbi et al., 1999]

50 networks of same topology being optimized
A Little Theory: **Information Bottleneck** [Tishbi et al., 1999]

**Limitation:** Does not seem to work for non-Tanh activations (e.g., ReLU)
What is a good multimodal representation?

- **Similarity** in the representation (somehow) implies similarity in corresponding concepts (we saw this in word2vec)

- **Useful** for various discriminative tasks (retrieval, mapping, fusion, etc.)

- Possible to obtain in absence of one or mere modalities

- Fill in missing modalities given others (map or translate between modalities)
Multimodal Representation Types

**Joint** representations:

- Simplest version: **modality concatenation** (early fusion)
- Can be learned **supervised** or **unsupervised**

*slide from Louis-Philippe Morency*
**Multimodal Representation Types**

### Joint representations:

- Simplest version: **modality concatenation** (early fusion)
- Can be learned **supervised** or **unsupervised**

### Coordinated representations:

- **Similarity-based** methods (e.g., cosine distance)
- **Structure constraints** (e.g., orthogonality, sparseness)
- Examples: CCA, joint embeddings

*slide from Louis-Philippe Morency*
Multimodal Representation Types

Joint representations:

- Simplest version: modality concatenation (early fusion)
- Can be learned supervised or unsupervised

*slide from Louis-Philippe Morency*
Joint Representation: Simple Multimodal Autoencoders

**Concatenating** modalities is fine, but requires both modalities at test time.

No ability to ensure there is indeed **sharing** in the representations space.
Joint Representation: Deep Multimodal Autoencoders

[ Ngiam et al., 2011 ]

Each **modality** can be pre-trained
- using denoising autoencoder

To train the model, **reconstruct both modalities** using
- both Audio & Video
- just Audio
- just Video

*slide from Louis-Philippe Morency*
Multimodal Research: Historical Perspective

The McGurk Effect

McGurk Effect (1976)

* Adopted from slides by Louis-Philippe Morency

* video credit: OK Science
Multimodal Research: Historical Perspective

The McGurk Effect

McGurk Effect (1976)

* video credit: OK Science

* Adopted from slides by Louis-Philippe Morency
Joint Representation: Deep Multimodal Autoencoders

Table 3: McGurk Effect

<table>
<thead>
<tr>
<th>Audio / Visual Setting</th>
<th>Model prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>/ga/, Audio /ga/</td>
<td>82.6% 2.2% 15.2%</td>
</tr>
<tr>
<td>/ba/, Audio /ba/</td>
<td>4.4% 89.1% 6.5%</td>
</tr>
<tr>
<td>/ga/, Audio /ba/</td>
<td>28.3% 13.0% 58.7%</td>
</tr>
</tbody>
</table>

*slide from Louis-Philippe Morency"
Useful when you know you may only be conditioning on one modality at test time

Can be regarded as a form of **regularization**

*slide from Louis-Philippe Morency*
Supervised Joint Representation

For supervised leaning tasks, we need to join unimodal representations

- Simple **concatenation**
- Element-wise **multiplicative** interactions
- many many others

Encoder-decoder Architectures

*e.g. Sentiment*  
softmax

$h_m$

$h_x$ & $h_y$

Text $X$ & Image $Y$

*slide from Louis-Philippe Morency*
Multi-modal Sentiment Analysis

For supervised learning tasks, we need to join unimodal representations

— Simple **concatenation**

MOSI dataset (Zadeh et al, 2016)

- 2199 subjective video segments
- Sentiment intensity annotations
- 3 modalities: text, video, audio

\[ h_m = \sigma(W \cdot [h_x, h_y, h_z]^T) \]

*slide from Louis-Philippe Morency*
Bilinear Pooling

For supervised learning tasks, we need to join unimodal representations

- Simple **concatenation**
- Element-wise **multiplicative** interactions

\[ h_m = h_x \otimes h_y \]

[Tenenbaum and Freeman, 2000]

*slide from Louis-Philippe Morency*
For supervised learning tasks, we need to join unimodal representations

- Simple **concatenation**
- Element-wise **multiplicative** interactions

\[
\mathbf{h}_m = \begin{bmatrix} \mathbf{h}_x \\ 1 \end{bmatrix} \otimes \begin{bmatrix} \mathbf{h}_y \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_x & \mathbf{h}_x \otimes \mathbf{h}_y \end{bmatrix}
\]

[ Zadeh, Jones and Morency, EMNLP 2017 ]
For supervised learning tasks, we need to join unimodal representations

- Simple *concatenation*
- Element-wise *multiplicative* interactions

\[
\mathbf{h}_m = \left[ \begin{array}{c} \mathbf{h}_x \\ 1 \end{array} \right] \otimes \left[ \begin{array}{c} \mathbf{h}_y \\ 1 \end{array} \right] \otimes \left[ \begin{array}{c} \mathbf{h}_z \\ 1 \end{array} \right]
\]

[ Zadeh, Jones and Morency, EMNLP 2017 ]
Low-rank Tensor Fusion

Tucker tensor decomposition leads to MUTAN fusion

[Ben-younes et al., ICCV 2017]

*slide from Louis-Philippe Morency
For supervised learning tasks, we need to join unimodal representations

- Simple *concatenation*
- Element-wise *multiplicative* interactions

**Encoder-decoder** Architectures
Multimodal Representation Types

Coordinated representations:

- **Similarity-based** methods (e.g., cosine distance)
- **Structure constraints** (e.g., orthogonality, sparseness)
- Examples: CCA, joint embeddings

*slide from Louis-Philippe Morency*
Data with **Multiple Views**

\[ x_1^{(i)} \quad x_2^{(i)} \]

- **demographic properties**
  - Image of people
- **responses to survey**
  - Logo of political parties
- **audio features at time \( i \)**
  - Heatmap image
- **video features at time \( i \)**
  - Image of person speaking

*slide from Andrew, Arora, Bilmes, Livescu*
Correlated Representations

Goal: Find representations $f_1(x_1), f_2(x_2)$ for each view that maximize correlation:

$$\text{corr}(f_1(x_1), f_2(x_2)) = \frac{\text{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\text{var}(f_1(x_1)) \cdot \text{var}(f_2(x_2))}}$$

*slide from Andrew, Arora, Bilmes, Livescu*
Correlated Representations

**Goal:** Find representations \( f_1(x_1), f_2(x_2) \) for each view that maximize correlation:

\[
\text{corr}(f_1(x_1), f_2(x_2)) = \frac{\text{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\text{var}(f_1(x_1)) \cdot \text{var}(f_2(x_2))}}
\]

Finding correlated representations can be **useful** for

- Gaining insights into the data
- Detecting of asynchrony in test data
- Removing noise uncorrelated across views
- Translation or retrieval across views

*slide from Andrew, Arora, Bilmes, Livescu*
Correlated Representations

**Goal:** Find representations $f_1(x_1), f_2(x_2)$ for each view that maximize correlation:

$$
corr(f_1(x_1), f_2(x_2)) = \frac{\text{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\text{var}(f_1(x_1)) \cdot \text{var}(f_2(x_2))}}
$$

Finding correlated representations can be **useful** for

- Gaining insights into the data
- Detecting of asynchrony in test data
- Removing noise uncorrelated across views
- Translation or retrieval across views

Has been **applied widely** to problems in computer vision, speech, NLP, medicine, chemometrics, metrology, neurology, etc.

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

Classical technique to find **linear** correlated representations, i.e.,

\[ f_1(x_1) = \mathbf{W}_1^T x_1 \]
\[ f_2(x_2) = \mathbf{W}_2^T x_2 \]

where

\[ \mathbf{W}_1 \in \mathbb{R}^{d_1 \times k} \]
\[ \mathbf{W}_2 \in \mathbb{R}^{d_2 \times k} \]
CCA: Canonical Correlation Analysis

Classical technique to find linear correlated representations, i.e.,

\[ f_1(x_1) = W_1^T x_1 \]  \quad \text{where} \quad W_1 \in \mathbb{R}^{d_1 \times k} \\
\[ f_2(x_2) = W_2^T x_2 \]  \quad W_2 \in \mathbb{R}^{d_2 \times k} \\

The first columns \((w_{1,:1}, w_{2,:1})\) of the matrices \(W_1\) and \(W_2\) are found to maximize the correlation of the projections:

\[ (w_{1,:1}, w_{2,:1}) = \arg \max \text{corr}(w_{1,:1}^T X_1, w_{2,:1}^T X_2) \]

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

Classical technique to find **linear** correlated representations, i.e.,

\[ f_1(x_1) = W_1^T x_1 \quad \text{where} \quad W_1 \in \mathbb{R}^{d_1 \times k} \]

\[ f_2(x_2) = W_2^T x_2 \quad \text{where} \quad W_2 \in \mathbb{R}^{d_2 \times k} \]

The first columns \((w_{1,1}, w_{2,1})\) of the matrices \(W_1\) and \(W_2\) are found to maximize the **correlation of the projections**:

\[
(w_{1,1}, w_{2,1}) = \arg \max \text{corr}(w_{1,1}^T X_1, w_{2,1}^T X_2)
\]

Subsequent pairs are constrained to be **uncorrelated with previous components** (i.e., for \( j < i \))

\[
\text{corr}(w_{1,i}^T X_1, w_{1,j}^T X_1) = \text{corr}(w_{2,i}^T X_2, w_{2,j}^T X_2) = 0
\]

*slide from Andrew, Arora, Bilmes, Livescu*
CCA Illustration

\[ f_1(X_1) = w_1^T X_1 \quad \text{max corr} \quad f_2(X_2) = w_2^T X_2 \]

\[ X_1 \in \mathbb{R}^2 \quad X_2 \in \mathbb{R}^2 \]

Two views of each instance have the same color

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

1. Estimate **covariance matrix** with regularization:

\[
\Sigma_{11} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1)(x_1^{(i)} - \bar{x}_1)^T + r_1 I
\]

\[
\Sigma_{12} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1)(x_2^{(i)} - \bar{x}_2)^T
\]

\[
\Sigma_{22} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_2^{(i)} - \bar{x}_2)(x_2^{(i)} - \bar{x}_2)^T + r_2 I
\]

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

1. Estimate **covariance matrix** with regularization:

\[
\begin{align*}
\Sigma_{11} &= \frac{1}{N-1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1)(x_1^{(i)} - \bar{x}_1)^T + r_1 I \\
\Sigma_{12} &= \frac{1}{N-1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1)(x_2^{(i)} - \bar{x}_2)^T \\
\Sigma_{12} &= \frac{1}{N-1} \sum_{i=1}^{N} (x_2^{(i)} - \bar{x}_2)(x_1^{(i)} - \bar{x}_1)^T + r_2 I \\
\Sigma_{22} &= \frac{1}{N-1} \sum_{i=1}^{N} (x_2^{(i)} - \bar{x}_2)(x_2^{(i)} - \bar{x}_2)^T + r_2 I
\end{align*}
\]

\[
\Sigma = \begin{bmatrix}
\Sigma_{11} & \Sigma_{12} \\
\Sigma_{12} & \Sigma_{22}
\end{bmatrix}
\]

\[
W_1^* \quad W_2^*
\]
CCA: Canonical Correlation Analysis

1. Estimate **covariance matrix** with regularization:

\[
\Sigma_{11} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1) (x_1^{(i)} - \bar{x}_1)^T + r_1 I
\]

\[
\Sigma_{12} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_1^{(i)} - \bar{x}_1) (x_2^{(i)} - \bar{x}_2)^T
\]

\[
\Sigma_{22} = \frac{1}{N - 1} \sum_{i=1}^{N} (x_2^{(i)} - \bar{x}_2) (x_2^{(i)} - \bar{x}_2)^T + r_2 I
\]

2. Form **normalized covariance** matrix: \( T = \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2} \) and its singular value decomposition \( T = UDV^T \)

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

1. Estimate **covariance matrix** with regularization:

\[
\Sigma_{11} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{1}^{(i)} - \bar{x}_1)(x_{1}^{(i)} - \bar{x}_1)^T + r_1 I \\
\Sigma_{12} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{1}^{(i)} - \bar{x}_1)(x_{2}^{(i)} - \bar{x}_2)^T \\
\Sigma_{22} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{2}^{(i)} - \bar{x}_2)(x_{2}^{(i)} - \bar{x}_2)^T + r_2 I
\]

2. Form **normalized covariance** matrix: \( T = \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2} \) and its singular value decomposition \( T = U D V^T \)

3. **Total correlation** at \( k \) is \( \sum_{i=1}^{k} D_{ii} \)

*slide from Andrew, Arora, Bilmes, Livescu*
CCA: Canonical Correlation Analysis

1. Estimate **covariance matrix** with regularization:

\[
\Sigma_{11} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{1}^{(i)} - \bar{x}_{1})(x_{1}^{(i)} - \bar{x}_{1})^T + r_1 I \\
\Sigma_{12} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{1}^{(i)} - \bar{x}_{1})(x_{2}^{(i)} - \bar{x}_{2})^T \\
\Sigma_{12} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{1}^{(i)} - \bar{x}_{1})(x_{2}^{(i)} - \bar{x}_{2})^T + r_2 I \\
\Sigma_{22} = \frac{1}{N-1} \sum_{i=1}^{N} (x_{2}^{(i)} - \bar{x}_{2})(x_{2}^{(i)} - \bar{x}_{2})^T + r_2 I
\]

2. Form **normalized covariance** matrix: \( T = \Sigma_{11}^{-1/2} \Sigma_{12} \Sigma_{22}^{-1/2} \) and its singular value decomposition \( T = UDV^T \)

3. **Total correlation** at \( k \) is \( \sum_{i=1}^{k} D_{ii} \)

4. The optimal projection matrices are:

\[
W_1^* = \Sigma_{11}^{-1/2} U_k \\
W_2^* = \Sigma_{22}^{-1/2} V_k
\]

where \( U_k \) is the first \( k \) columns of \( U \).

*slide from Andrew, Arora, Bilmes, Livescu*
There may be non-linear functions $f_1(x_1), f_2(x_2)$ that produce more highly correlated (better) representations than linear projections.

**Kernel CCA** is a principal method for finding such functions:
- Learns functions from any reproducing kernel Hilbert space
- May use different kernels for each view

Using RBF (Gaussian) kernel in KCCA is akin to finding sets of instances that form clusters in both views.

*slide from Andrew, Arora, Bilmes, Livescu*
KCCA vs. CCA

Pros:
— More complex function space of KCCA can yield dramatically higher correlations

Cons:
— KCCA is slower to train
— For KCCA training set must be stored and referenced at test time
— KCCA model is more difficult to interpret

*slide from Andrew, Arora, Bilmes, Livescu*
Deep CCA

Canonical Correlation Analysis

View 1

View 2

*slide from Andrew, Arora, Bilmes, Livescu
Benefits of Deep CCA

Pros:

– Better suited for natural, real-world data
– **Parametric model**
  – The training set can be disregarded once the model is learned
  – Computational speed at test time is fast

*slide from Andrew, Arora, Bilmes, Livescu*
Deep CCA: Training

Training a Deep CCA model:

1. **Pretrain** the layers of each side individually

2. **Jointly fine-tune** all parameters to maximize the total correlation of the output layers. Requires computing correlation gradient:
   - Forward propagate activations on both sides.
   - Compute correlation and its gradient w.r.t. output layers.
   - Backpropagate gradient on both sides.

*slide from Andrew, Arora, Bilmes, Livescu*
Training a Deep CCA model:

1. **Pretrain** the layers of each side individually

2. **Jointly fine-tune** all parameters to maximize the total correlation of the output layers. Requires computing correlation gradient:
   - Forward propagate activations on both sides.
   - Compute correlation and its gradient w.r.t. output layers.
   - Backpropagate gradient on both sides.

Correlation is a population objective, so instead of one instance (or minibatch) training, requires L-BFGS second-order method (with full-batch)

*slide from Andrew, Arora, Bilmes, Livescu*
Deep Canonically Correlated Autoencoders (DCCAE)

Jointly optimize for DCCA and auto encoders loss functions

— A trade-off between multi-view correlation and reconstruction error from individual views

[Wang et al., ICML 2015]
Multimodal Representation Types

Coordinated representations:

- **Similarity-based** methods (e.g., cosine distance)
- **Structure constraints** (e.g., orthogonality, sparseness)
- Examples: CCA, joint embeddings

*slide from Louis-Philippe Morency*
Correlated Representations vs. Joint Embeddings

**Correlated Representations:** Find representations $f_1(x_1), f_2(x_2)$ for each view that maximize correlation:

$$\text{corr}(f_1(x_1), f_2(x_2)) = \frac{\text{cov}(f_1(x_1), f_2(x_2))}{\sqrt{\text{var}(f_1(x_1)) \cdot \text{var}(f_2(x_2))}}$$

**Joint Embeddings:** Models that minimize distance between ground truth pairs of samples:

$$\min_{f_1, f_2} D\left(f_1(x_1^{(i)}), f_2(x_2^{(i)})\right)$$
Joint Embeddings

Distance(s,t)

Image features s

Text: a parrot rides a tricycle
Joint Embeddings

Image features $s$

Text: *a parrot rides a tricycle*
Joint Embeddings

Nearest images

- blue + red =
- blue + yellow =
- yellow + red =
- white + red =

[ Kiros et al., Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models, 2014 ]
Joint Embeddings

- day + night =
- flying + sailing =
- bowl + box =
- box + bowl =

[ Kiros et al., Unifying Visual-Semantic Embeddings with Multimodal Neural Language Models, 2014 ]