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

Lecture 7: Word2Vec, Language Models and RNNs
— Assignment 1 grades (available on Connect) 
— Solutions will be posted over the weekend

— Assignment 2 was **due Yesterday**

— Assignment 3 will be out **Friday, January 26th**
— The due deadline will be extended

— Paper choices will be due **next week** (google form)
— **Projects** groups and short description (google form)
Representing a **Word**: One Hot Encoding

**Vocabulary**

dog

cat

person

holding

tree

computer

using

*slide from V. Ordonex*
Representing a **Word**: One Hot Encoding

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>2</td>
</tr>
<tr>
<td>person</td>
<td>3</td>
</tr>
<tr>
<td>holding</td>
<td>4</td>
</tr>
<tr>
<td>tree</td>
<td>5</td>
</tr>
<tr>
<td>computer</td>
<td>6</td>
</tr>
<tr>
<td>using</td>
<td>7</td>
</tr>
</tbody>
</table>

*slide from V. Ordonex*
Representing a **Word**: One Hot Encoding

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>one-hot encodings</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>[1, 0, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>cat</td>
<td>[0, 1, 0, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>person</td>
<td>[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>holding</td>
<td>[0, 0, 0, 1, 0, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>tree</td>
<td>[0, 0, 0, 0, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>computer</td>
<td>[0, 0, 0, 0, 0, 1, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>using</td>
<td>[0, 0, 0, 0, 0, 0, 1, 0, 0, 0]</td>
</tr>
</tbody>
</table>

*slide from V. Ordonex*
Representing **Phrases**: Bag-of-Words

**bag-of-words** representation

Vocabulary

<table>
<thead>
<tr>
<th>Word</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>2</td>
</tr>
<tr>
<td>person</td>
<td>3</td>
</tr>
<tr>
<td>holding</td>
<td>4</td>
</tr>
<tr>
<td>tree</td>
<td>5</td>
</tr>
<tr>
<td>computer</td>
<td>6</td>
</tr>
<tr>
<td>using</td>
<td>7</td>
</tr>
</tbody>
</table>

*slide from V. Ordonex*
Representing **Phrases**: Bag-of-Words

<table>
<thead>
<tr>
<th>Vocabulary</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>2</td>
</tr>
<tr>
<td>person</td>
<td>3</td>
</tr>
<tr>
<td>holding</td>
<td>4</td>
</tr>
<tr>
<td>tree</td>
<td>5</td>
</tr>
<tr>
<td>computer</td>
<td>6</td>
</tr>
<tr>
<td>using</td>
<td>7</td>
</tr>
</tbody>
</table>

**bag-of-words** representation

person holding dog \[ \{3, 4, 1\} \] \[1, 0, 1, 1, 0, 0, 0, 0, 0, 0\]
Representing **Phrases**: Bag-of-Words

**bag-of-words** representation

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Bag of Words</th>
<th>[1, 0, 1, 1, 0, 0, 0, 0, 0, 0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>person holding dog</td>
<td>{3, 4, 1}</td>
<td></td>
</tr>
<tr>
<td>person holding cat</td>
<td>{3, 4, 2}</td>
<td></td>
</tr>
</tbody>
</table>

**Vocabulary**

- dog 1
- cat 2
- person 3
- holding 4
- tree 5
- computer 6
- using 7

*slide from V. Ordonex*
## Representing Phrases: Bag-of-Words

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Indexes</th>
<th>Bag-of-Words Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>person holding dog</td>
<td>{3, 4, 1}</td>
<td>[1, 0, 1, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>person holding cat</td>
<td>{3, 4, 2}</td>
<td>[1, 1, 0, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>person using computer</td>
<td>{3, 7, 6}</td>
<td>[0, 0, 0, 1, 0, 1, 1, 0, 0]</td>
</tr>
</tbody>
</table>

### Vocabulary

<table>
<thead>
<tr>
<th>Word</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>dog</td>
<td>1</td>
</tr>
<tr>
<td>cat</td>
<td>2</td>
</tr>
<tr>
<td>person</td>
<td>3</td>
</tr>
<tr>
<td>holding</td>
<td>4</td>
</tr>
<tr>
<td>tree</td>
<td>5</td>
</tr>
<tr>
<td>computer</td>
<td>6</td>
</tr>
<tr>
<td>using</td>
<td>7</td>
</tr>
</tbody>
</table>

*slide from V. Ordonex*
Representing **Phrases**: Bag-of-Words

<table>
<thead>
<tr>
<th>Phrase</th>
<th>bag-of-words representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>person holding dog</td>
<td>{3, 4, 1} [1, 0, 1, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>person holding cat</td>
<td>{3, 4, 2} [1, 1, 0, 1, 0, 0, 0, 0, 0]</td>
</tr>
<tr>
<td>person using computer</td>
<td>{3, 7, 6} [0, 0, 0, 1, 0, 1, 1, 0, 0]</td>
</tr>
<tr>
<td>person using computer person holding cat</td>
<td>{3, 3, 7, 6, 2} [0, 1, 2, 1, 0, 1, 1, 0, 0]</td>
</tr>
</tbody>
</table>

*slide from V. Ordonex*
Distributional Hypothesis

— At least certain aspects of the meaning of lexical expressions depend on their distributional properties in the linguistic contexts

— The degree of semantic similarity between two linguistic expressions is a function of the similarity of the two linguistic contexts in which they can appear

* Adopted from slides by Louis-Philippe Morency [Lenci, 2008]*
What is the meaning of “bardiwac”? 

— He handed her glass of bardiwac.
— Beef dishes are made to complement the bardiwacs.
— Nigel staggered to his feet, face flushed from too much bardiwac.
— Malbec, one of the lesser-known bardiwac grapes, responds well to Australia’s sunshine.
— I dined off bread and cheese and this excellent bardiwac.
— The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

* Adopted from slides by Louis-Philippe Morency
What is the meaning of “bardiwac”?

— He handed her glass of bardiwac.
— Beef dishes are made to complement the bardiwacs.
— Nigel staggered to his feet, face flushed from too much bardiwac.
— Malbec, one of the lesser-known bardiwac grapes, responds well to Australia’s sunshine.
— I dined off bread and cheese and this excellent bardiwac.
— The drinks were delicious: blood-red bardiwac as well as light, sweet Rhenish.

bardic is an alcoholic beverage made from grapes

* Adopted from slides by Louis-Philippe Morency
Geometric Interpretation: Co-occurrence as feature

- Row vector describes usage of word in a corpus of text

- Can be seen as coordinates of the point in an n-dimensional Euclidian space

![Co-occurrence Matrix](image)

* Slides from Louis-Philippe Morency
Geometric Interpretation: Co-occurrence as feature

— Row vector describes usage of word in a corpus of text

— Can be seen as coordinates of the point in an n-dimensional Euclidian space

Co-occurrence Matrix

* Slides from Louis-Philippe Morency
Distance and Similarity

- Illustrated in two dimensions

- Similarity = spatial proximity (Euclidian distance)

- Location depends on frequency of noun (dog is 27 times as frequent as cat)

* Slides from Louis-Philippe Morency
Angle and Similarity

- direction is more important than location
- normalize length of vectors
- or use angle as a distance measure

* Slides from Louis-Philippe Morency
Geometric Interpretation: Co-occurrence as feature

- Row vector describes usage of word in a corpus of text

- Can be seen as coordinates of the point in an n-dimensional Euclidian space

Co-occurrence Matrix

* Slides from Louis-Philippe Morency
Geometric Interpretation: Co-occurrence as feature

- Row vector describes usage of word in a corpus of text

- Can be seen as coordinates of the point in an n-dimensional Euclidian space

Way too high dimensional!

Co-occurrence Matrix

* Slides from Louis-Philippe Morency
SVD for Dimensionality Reduction

\[
\begin{align*}
X &\approx U S V^T \\
\hat{X} &\approx \hat{U} \hat{S} \hat{V}^T
\end{align*}
\]
Learned Word Vector Visualization

We can also use other methods, like LLE here:


[ Roweis and Saul, 2000 ]
Issues with SVD

**Computational** cost for a $d \times n$ matrix is $O(dn^2)$, where $d < n$

- Makes it not possible for large number of word vocabularies or documents

It is hard to incorporate out of sample (new) words or documents

*slide from Vagelis Hristidis*
**word2vec**: Representing the Meaning of Words [Mikolov et al., 2013]

**Key idea**: Predict surrounding words of every word

**Benefits**: Faster and easier to incorporate new document, words, etc.
**word2vec**: Representing the Meaning of Words  

**Key idea:** Predict surrounding words of every word

**Benefits:** Faster and easier to incorporate new document, words, etc.

Continuous Bag of Words (**CBOW**): use context words in a window to predict middle word

**Skip-gram:** use the middle word to predict surrounding ones in a window

*slide from Vagelis Hristidis [Mikolov et al., 2013]*
CBOW: Continuous Bag of Words

Example: “The cat sat on floor” (window size 2)

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
CBOW: Continuous Bag of Words

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
CBOW: Continuous Bag of Words

Input layer

\[ x \in \mathbb{R}^{|V|} \]

\[ w \in \mathbb{R}^{|V|\times|N|} \]

\[ \hat{v} \in \mathbb{R}^{N} \]

Hidden layer

\[ w' \in \mathbb{R}^{N\times|V|} \]

Output layer

\[ y \in \mathbb{R}^{V} \]

\[ \hat{y} \in \mathbb{R}^{V} \]

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
CBOW: Continuous Bag of Words

Input layer

Hidden layer

Output layer

Parameters to be learned

\[ \mathbf{W}_{|V| \times |V|} \]

\[ \mathbf{W}'_{|N| \times |N|} \]

\[ \mathbf{W}_{|V| \times |N|} \]

\[ \mathbf{W}'_{|N| \times |V|} \]

\[ \hat{y} \in \mathbb{R}^{|N|} \]

\[ \hat{y} \in \mathbb{R}^{|V|} \]

\[ \mathbf{x} \in \mathbb{R}^{|V|} \]

\[ \mathbf{v} \in \mathbb{R}^{|N|} \]

\[ \mathbf{v} \in \mathbb{R}^{|V|} \]

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
CBOW: Continuous Bag of Words

**Input layer**

*Parameters to be learned*

**Hidden layer**

**Output layer**

- **x ∈ ℝ^|V|**
- **W_{|V|×|N|}**
- **y ∈ ℝ^|N|**
- **W'_{|N|×|V|}**
- **y' ∈ ℝ^|V|**

Size of the word vector (e.g., 300)

*slide from Vagelis Hristidis [Mikolov et al., 2013]*
CBOW: Continuous Bag of Words

\[ \text{Input layer} \]

\[ x_{\text{cat}} W_{|N|x|V|} \times x_{\text{cat}} = v_{\text{cat}} \]

\[ x_{\text{on}} W_{|N|x|V|} \times x_{\text{on}} = v_{\text{on}} \]

\[ x \in \mathbb{R}^{|V|} \]

\[ \hat{v} \in \mathbb{R}^{|N|} \]

\[ \hat{v} \times W_{|N|x|V|} \times \hat{x} = \hat{y} \]

\[ \hat{y} \in \mathbb{R}^{|V|} \]

[ Mikolov et al., 2013 ]

*slide from Vagelis Hristidis
CBOW: Continuous Bag of Words

Input layer

\[ W_{|V| \times |N|} \times x_{\text{cat}} = v_{\text{cat}} \]

*slide from Vagelis Hristidis [Mikolov et al., 2013]*
**CBOW**: Continuous Bag of Words

\[ \mathbf{x}_{cat} \times \mathbf{W}_{|V| \times |N|}^T \times \mathbf{x}_{on} = \mathbf{v}_{on} \]

\[ \begin{array}{cccccccc}
0.1 & 2.4 & 1.6 & 1.8 & 0.5 & 0.9 & \ldots & \ldots & 3.2 \\
0.5 & 2.6 & 1.4 & 2.9 & 1.5 & 3.6 & \ldots & \ldots & 6.1 \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
0.6 & 1.8 & 2.7 & 1.9 & 2.4 & 2.0 & \ldots & \ldots & 1.2 \\
\end{array} \]

*slide from Vagelis Hristidis [Mikolov et al., 2013]*
**CBOW: Continuous Bag of Words**

\[
x_{\text{cat}} \cdot W_{|N| \times |V|} \cdot x_{\text{cat}} = v_{\text{cat}}
\]

\[
x_{\text{on}} \cdot W_{|N| \times |V|} \cdot x_{\text{on}} = v_{\text{on}}
\]

\[
\hat{v} = \frac{v_{\text{cat}} + v_{\text{on}}}{2}
\]

\[
\hat{y} = \frac{v_{\text{cat}} + v_{\text{on}}}{2}
\]

\[
\hat{y} = \hat{v}
\]

\[
\hat{y} = v_{\text{cat}} + v_{\text{on}}
\]

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
CBOW: Continuous Bag of Words

Input layer

\[ x_{\text{cat}} \times x_{\text{cat}} = v_{\text{cat}} \]

\[ x_{\text{on}} \times x_{\text{on}} = v_{\text{on}} \]

\[ x \in \mathbb{R}^{V} \]

Hidden layer

\[ \hat{v} \in \mathbb{R}^{N} \]

Output layer

\[ \hat{y} = \text{softmax}(z) \]

\[ \hat{y}_{\text{sat}} \]

\[ \hat{y} \in \mathbb{R}^{V} \]

\[ \hat{v}' = W'_{|V| \times |N|} \times \hat{v} = z \]

*slide from Vagelis Hristidis

[ Mikolov et al., 2013 ]
**CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

- **Input layer**
  - $x_{\text{cat}}$ and $x_{\text{on}}$
  - $\mathbf{W}_{|V| \times |V|} \times x_{\text{cat}} = v_{\text{cat}}$
  - $\mathbf{W}_{|V| \times |N|} \times x_{\text{on}} = \mathbf{v}_{\text{on}}$

- **Hidden layer**
  - $\mathbf{v} \in \mathbb{R}^{|N|}$

- **Output layer**
  - $\mathbf{W}'_{|V| \times |N|} \times \mathbf{v} = \mathbf{z}$
  - $\hat{\mathbf{y}} = \text{softmax}(\mathbf{z})$
  - $\hat{y}_{\text{sat}}$
  - $\hat{\mathbf{y}} \in \mathbb{R}^{|V|}$

- Optimize to get close to 1-hot encoding

*slide from Vagelis Hristidis
**CBOW**: Continuous Bag of Words

\[
\begin{align*}
\hat{y} &= \text{softmax}(z) \\
\hat{y}_{\text{sat}} &= \text{softmax}(z) \\
\hat{y} &\in \mathbb{R}^{|V|} \\
\hat{y}_{\text{sat}} &\in \mathbb{R}^{|V|} \\
\end{align*}
\]

*slide from Vagelis Hristidis [Mikolov et al., 2013]*
**CBOW: Interesting Observation**

[ Mikolov et al., 2013 ]

*slide from Vagelis Hristidis*
Skip-Gram Model

[ Mikolov et al., 2013 ]
Comparison

- **CBOW** is not great for rare words and typically needs less data to train.
- **Skip-gram** better for rate words and needs more data to train the model.

<table>
<thead>
<tr>
<th>Model</th>
<th>Vector Dimensionality</th>
<th>Training words</th>
<th>Accuracy [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Semantic</td>
</tr>
<tr>
<td>Collobert-Weston NNLM</td>
<td>50</td>
<td>660M</td>
<td>9.3</td>
</tr>
<tr>
<td>Turian NNLM</td>
<td>50</td>
<td>37M</td>
<td>1.4</td>
</tr>
<tr>
<td>Turian NNLM</td>
<td>200</td>
<td>37M</td>
<td>1.4</td>
</tr>
<tr>
<td>Mnih NNLM</td>
<td>50</td>
<td>37M</td>
<td>1.8</td>
</tr>
<tr>
<td>Mnih NNLM</td>
<td>100</td>
<td>37M</td>
<td>3.3</td>
</tr>
<tr>
<td>Mikolov RNNLM</td>
<td>80</td>
<td>320M</td>
<td>4.9</td>
</tr>
<tr>
<td>Mikolov RNNLM</td>
<td>640</td>
<td>320M</td>
<td>8.6</td>
</tr>
<tr>
<td>Huang NNLM</td>
<td>50</td>
<td>990M</td>
<td>13.3</td>
</tr>
<tr>
<td>Our NNLM</td>
<td>20</td>
<td>6B</td>
<td>12.9</td>
</tr>
<tr>
<td>Our NNLM</td>
<td>50</td>
<td>6B</td>
<td>27.9</td>
</tr>
<tr>
<td>Our NNLM</td>
<td>100</td>
<td>6B</td>
<td>34.2</td>
</tr>
<tr>
<td>CBOW</td>
<td>300</td>
<td>783M</td>
<td>15.5</td>
</tr>
<tr>
<td>Skip-gram</td>
<td>300</td>
<td>783M</td>
<td><strong>50.0</strong></td>
</tr>
</tbody>
</table>
Interesting Results: **Word Analogies**

Test for linear relationships, examined by Mikolov et al. (2014)

$$d = \arg \max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

**man:woman :: king:?**

- king: [0.30 0.70]
- man: [0.20 0.20]
- woman: [0.60 0.30]
- queen: [0.70 0.80]
Interesting Results: **Word Analogies**

[ Mikolov et al., 2013 ]
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

* Slides from Louis-Philippe Morency*
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

* Slides from Louis-Philippe Morency
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

\[
\arg \max_{\text{wordsequence}} P(\text{wordsequence} | \text{acoustics}) =
\]

\[
\frac{P(\text{acoustics} | \text{wordsequence}) \times P(\text{wordsequence})}{P(\text{acoustics})}
\]

\[
\arg \max_{\text{wordsequence}} P(\text{acoustics} | \text{wordsequence}) \times P(\text{wordsequence})
\]

* Slides from Louis-Philippe Morency*
Language Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

\[
\text{arg max } P(\text{wordsequence} \mid \text{acoustics}) = \frac{\text{arg max } P(\text{acoustics} \mid \text{wordsequence}) \times P(\text{wordsequence})}{P(\text{acoustics})}
\]

\[
\text{arg max } P(\text{acoustics} \mid \text{wordsequence}) \times P(\text{wordsequence})
\]

* Slides from Louis-Philippe Morency*
Simple **Language Models**: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, \ldots, w_n] \)

We want to estimate \( p(w_{1:n}) \)

* Slides from Louis-Philippe Morency
Simple Language Models: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, \ldots, w_n] \)

We want to estimate \( p(w_{1:n}) \)

Using **Chain Rule** of probabilities:

\[
p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_{1:n-1})
\]
Simple **Language Models**: N-Grams

Given a word sequence: \( w_{1:n} = [w_1, w_2, \ldots, w_n] \)

We want to estimate \( p(w_{1:n}) \)

Using **Chain Rule** of probabilities:

\[
p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_{1:n-1})
\]

**Bi-gram Approximation:**

\[
p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-1})
\]

**N-gram Approximation:**

\[
p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-N+1:k-1})
\]

* Slides from Louis-Philippe Morency*
Estimating Probabilities

N-gram conditional probabilities can be estimated based on raw concurrence counts in the observed sequences

**Bi-gram:**

\[
p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}
\]

**N-gram:**

\[
p(w_n|w_{n-N-1:n-1}) = \frac{C(w_{n-N-1:n-1}w_n)}{C(w_{n-N-1:n-1})}
\]

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Mode

P(next word is "dog")
P(next word is "on")
P(next word is "the")
P(next word is "beach")

1-of-N encoding of "START"
1-of-N encoding of "dog"
1-of-N encoding of "on"
1-of-N encoding of "the"

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Mode

Problem: Does not model sequential information (too local)

* Slides from Louis-Philippe Morency
Neural-based Unigram Language Model

Problem: Does not model sequential information (too local)

* Slides from Louis-Philippe Morency
Why Model Sequences?

Foreign Minister. → FOREIGN MINISTER.

THE SOUND OF

\[ a_1=2 \quad a_2=0 \quad a_3=1 \quad a_4=3 \quad a_5=4 \quad a_6=2 \quad a_7=5 \]

\[ x = \text{bringen sie bitte das auto zurück}. \]

\[ y = \text{please return the car}. \]

Image Credit: Alex Graves and Kevin Gimpel

* slide from Dhruv Batra
Multi-modal tasks

A group of people shopping at an outdoor market.

There are many vegetables at the fruit stand.

[Vinyals et al., 2015]
Sequences where you don’t expect them …

Classify images by taking a series of “glimpses”

[ Gregor et al., ICML 2015 ]
[ Mnih et al., ICLR 2015 ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences where you don’t expect them …

Classify images by taking a series of “glimpses”

[ Gregor et al., ICML 2015 ]
[ Mnih et al., ICLR 2015 ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

one to one

Input: No sequence
Output: No seq.
Example: “standard” classification / regression problems

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequences in Inputs or Outputs?

- **Input:** No sequence  
  **Output:** No seq.  
  **Example:** “standard” classification / regression problems

- **Input:** No sequence  
  **Output:** No seq.  
  **Example:** Im2Caption

- **Input:** Sequence  
  **Output:** No seq.  
  **Example:** sentence classification, multiple-choice question answering

- **Input:** Sequence  
  **Output:** Sequence  
  **Example:** machine translation, video captioning, open-ended question answering, video question answering

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Key Conceptual Ideas

Parameter Sharing
- in computational graphs = adding gradients

“Unrolling”
- in computational graphs with parameter sharing

Parameter Sharing + “Unrolling”
- Allows modeling \textit{arbitrary length sequences}!
- Keeps number of parameters in check

* slide from Dhruv Batra
Recurrent Neural Network

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Recurrent Neural Network

usually want to predict a vector at some time steps

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
We can process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

where $h_t$ is the new state at time step $t$, $h_{t-1}$ is the old state at time step $t-1$, $x_t$ is the input vector at some time step, and $f_W$ is some function with parameters $W$. 

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Recurrent Neural Network

We can process a sequence of vectors $\mathbf{x}$ by applying a recurrence formula at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

**Note:** the same function and the same set of parameters are used at every time step.
(Vanilla) Recurrent Neural Network

\[ h_t = f_W(h_{t-1}, x_t) \]
(Vanilla) **Recurrent** Neural Network

\[
h_t = f_W(h_{t-1}, x_t)
\]

\[
h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)
\]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, *cs231n Stanford*
(Vanilla) **Recurrent Neural Network**

\[ y_t = W_{hy} h_t + b_y \]

\[ h_t = f_W(h_{t-1}, x_t) \]

\[ h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

\[
\begin{align*}
h_0 & \rightarrow f_W & \rightarrow h_1 \\
x_1 & \rightarrow & \\
\end{align*}
\]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

\[ h_0 \xrightarrow{f_W} h_1 \xrightarrow{f_W} h_2 \]

\[ x_1 \xleftarrow{ } x_2 \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph

h_0 \rightarrow f_W \rightarrow h_1 \rightarrow f_W \rightarrow h_2 \rightarrow f_W \rightarrow h_3 \rightarrow \ldots \rightarrow h_T

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Re-use the same weight matrix at every time-step

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**RNN Computational Graph: Many to Many**

\[
\begin{align*}
    &h_0 &\rightarrow &f_W &\rightarrow &h_1 &\rightarrow &f_W &\rightarrow &h_2 &\rightarrow &f_W &\rightarrow &h_3 &\rightarrow &\cdots &\rightarrow &h_T \\
    &x_1 &\rightarrow & & &x_2 &\rightarrow & & &x_3 &\rightarrow & & & & & & &
\end{align*}
\]

*slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford*
RNN Computational Graph: Many to Many

\[ \begin{align*}
    y_1 & \rightarrow L_1 \\
    y_2 & \rightarrow L_2 \\
    y_3 & \rightarrow L_3 \\
    \vdots & \\
    y_T & \rightarrow L_T \\
    h_0 & \rightarrow f_W \\
    h_1 & \rightarrow f_W \\
    h_2 & \rightarrow f_W \\
    h_3 & \rightarrow f_W \\
    \vdots & \\
    h_T & \\
    x_1 & \\
    x_2 & \\
    x_3 & \\
\end{align*} \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: Many to Many

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: Many to One

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
RNN Computational Graph: One to Many

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequence to Sequence: Many to One + One to Many

**Many to one:** Encode input sequence in a single vector

---

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Sequence to Sequence: Many to One + One to Many

**Many to one:** Encode input sequence in a single vector

**One to many:** Produce output sequence from single input vector

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Example: Character-level Language Model

Vocabulary:

[‘h’, ‘e’, ‘l’, ‘o’]

Example training sequence:

“hello”
**Example: Character-level Language Model**

**Vocabulary:**
['h', 'e', 'l', 'o']

Example training sequence: “hello”

\[ h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**Example:** Character-level Language Model

**Vocabulary:**

[‘h’, ‘e’, ‘l’, ‘o’]

**Example training sequence:**

“hello”

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**Example**: Character-level Language Model *(Sampling)*

**Vocabulary:**

['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Example: Character-level Language Model (Sampling)

Vocabulary:
[‘h’, ‘e’, ‘l’, ‘o’]

At test time sample one character at a time and feed back to the model.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
**Example**: Character-level Language Model (Sampling)

**Vocabulary:**

[‘h’, ‘e’, ‘l’, ‘o’]

At test time sample one character at a time and feed back to the model.
Example: Character-level Language Model (Sampling)

Vocabulary:
[‘h’, ‘e’, ‘l’, ‘o’]

At test time sample one character at a time and feed back to the model

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
BackProp Through Time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) chunks of the sequence, instead of the whole sequence.
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) chunks of the sequence, instead of the whole sequence.

Carry hidden states forward, but only BackProp through some smaller number of steps.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Truncated BackProp Through Time

Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Implementation: Relatively Easy

... you will have a chance to experience this in the Assignment 3
Learning to Write Like Shakespeare

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou art not prouder than the world's鮮花 ornament,
And only herald to the gaudy spring,
Within thine own bud burstiest thy content,
And tender churl mak'st waste in niggardling:
Pity the world, or else this gluton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Learning to Write Like Shakespeare ... after training a bit

at first:

"Tmont thithey" formesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
cioniogenc Phe lism thond hon at. MeiDimorotion in ther thize."

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Learning to Write Like Shakespeare ... after training

PANDARUS:
Alas, I think he shall be come approached and the day
When little strain would be attain’d into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
My fair nues begun out of the fact, to be conveyed,
Whose noble souls I’ll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I’ll drink it.

VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father’s world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master’s ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder’d at the deeds,
So drop upon your lordship’s head, and your opinion
Shall be against your honour.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTTHREAD_UNCCA) +
                ((count & 0x00000000fffff) & 0x0000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
            pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_contols(offset, idx, &offset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}
DopeLearning: Computational Approach to Rap Lyrics

Everybody got one
And all the pretty mommies want some
And what i told you all was
But you need to stay such do not touch
They really do not want you to vote
what do you condone
Music make you lose control
What you need is right here ahh oh
This is for you and me
I had to dedicate this song to you Mami
Now I see how you can be
I see u smiling i kno u hattig
Best I Eva Had x4
That I had to pay for
Do I have the right to take yours
Trying to stay warm

(2 Chainz - Extremely Blessed)
(Mos Def - Undeniable)
(Lil Wayne - Welcome Back)
(Common - Heidi Hoe)
(KRS One - The Mind)
(Cam’ron - Bubble Music)
(Missy Elliot - Lose Control)
(Wiz Khalifa - Right Here)
(Missy Elliot - Hit Em Wit Da Hee)
(Fat Joe - Bendicion Mami)
(Lil Wayne - How To Hate)
(Wiz Khalifa - Damn Thing)
(Nicki Minaj - Best I Ever Had)
(Ice Cube - X Bitches)
(Common - Retrospect For Life)
(Everlast - 2 Pieces Of Drama)

[ Malmi et al., KDD 2016 ]
Sunspiring: First movie generated by AI

Sunspiring, a short science fiction movie written entirely by AI, debuts exclusively on Ars today.
Multilayer RNNs

\[ h_t^l = \tanh \ W^l \begin{pmatrix} h_{t-1}^l \\ h_{t-1}^l \end{pmatrix} \]

\( h \in \mathbb{R}^n \quad W^l \ [n \times 2n] \)

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Vanilla RNN Gradient Flow

\[ h_t = \tanh(W_{hh}h_{t-1} + W_{hx}x_t) \]

\[ = \tanh \left( \begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \]

\[ = \tanh \left( W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right) \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford

[Bengio et al., 1994]
[ Pascanu et al., ICML 2013 ]
Vanilla RNN Gradient Flow

Backpropagation from $h_t$ to $h_{t-1}$ multiplies by $W$ (actually $W_{hh}^T$)

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$= \tanh \left( \begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

$$= \tanh \left( W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]
Vanilla RNN Gradient Flow

Computing gradient of $h_0$ involves many factors of $W$ (and repeated tanh)

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Vanilla RNN **Gradient Flow**

Computing gradient of $h_0$ involves many factors of $W$ (and repeated tanh)

**Largest singular value > 1:** 
**Exploding gradients**

**Largest singular value < 1:** 
**Vanishing gradients**

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]
Vanilla RNN Gradient Flow

**Computing gradient of** $h_0$ **involves many factors of** $W$ **(and repeated tanh)**

**Largest singular value > 1:** **Exploding gradients**

**Largest singular value < 1:** **Vanishing gradients**

**Gradient clipping:** Scale gradient if its norm is too big

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, *cs231n Stanford*
Vanilla RNN Gradient Flow

Largest singular value > 1: Exploding gradients

Largest singular value < 1: Vanishing gradients

Computing gradient of $h_0$ involves many factors of $W$ (and repeated tanh)

Change RNN architecture

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Long-Short Term Memory (LSTM)

Vanilla RNN

$$h_t = \tanh \left( W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \right)$$

LSTM

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

[Hochreiter and Schmidhuber, NC 1977]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Long-Short Term Memory (LSTM)

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
Long-Short Term Memory (LSTM)

Cell state / **memory**

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra*
LSTM Intuition: Forget Gate

Should we continue to **remember** this “bit” of information or not?

\[
f_t = \sigma \left( W_f \cdot [h_{t-1}, x_t] + b_f \right)
\]

* Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
LSTM Intuition: Forget Gate

Should we continue to **remember** this “bit” of information or not?

Intuition: memory and forget gate output multiply, output of forget gate can be though of as binary (0 or 1)

\[
f_t = \sigma (W_f \cdot [h_{t-1}, x_t] + b_f)
\]

* Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
* slide from Dhruv Batra
LSTM Intuition: Input Gate

Should we **update** this “bit” of information or not?
If yes, then what should we **remember**?

\[
i_t = \sigma \left( W_i \cdot [h_{t-1}, x_t] + b_i \right) \\
\tilde{C}_t = \tanh \left( W_C \cdot [h_{t-1}, x_t] + b_C \right)
\]
LSTM Intuition: Memory Update

Forget what needs to be forgotten + memorize what needs to be remembered

\[ C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \]

* slide from Dhruv Batra
LSTM Intuition: Output Gate

Should we output this bit of information (e.g., to “deeper” LSTM layers)?

\[
o_t = \sigma (W_o [h_{t-1}, x_t] + b_o) \\
h_t = o_t \times \text{tanh} (C_t)
\]

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)
**LSTM Intuition: Additive Updates**

Backpropagation from $c_t$ to $c_{t-1}$ only elementwise multiplication by $f$, no matrix multiply by $W$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
LSTM Intuition: Additive Updates

Uninterrupted gradient flow!

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
LSTM Intuition: Additive Updates

Uninterrupted gradient flow!

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
LSTM Variants: with Peephole Connections

Lets gates see the cell state / memory

\[ f_t = \sigma \left( W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f \right) \]
\[ i_t = \sigma \left( W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i \right) \]
\[ o_t = \sigma \left( W_o \cdot [C_t, h_{t-1}, x_t] + b_o \right) \]

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
LSTM Variants: with Coupled Gates

Only memorize new information when you’re forgetting old

\[ C_t = f_t \cdot C_{t-1} + (1 - f_t) \cdot \tilde{C}_t \]
Gated Recurrent Unit (GRU)

No explicit memory; memory = hidden output

\[ z_t = \sigma (W_z \cdot [h_{t-1}, x_t]) \]
\[ r_t = \sigma (W_r \cdot [h_{t-1}, x_t]) \]
\[ \tilde{h}_t = \tanh (W \cdot [r_t \ast h_{t-1}, x_t]) \]
\[ h_t = (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t \]

\[ z = \text{memorize new and forget old} \]

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra
Phased LSTM

Gates are controlled by **phased** (periodic) **oscillations**

[ Neil et al., 2016 ]
Skip-thought Vectors

word2vec but for sentences, where each sentence is processed by an LSTM

[ Kiros et al., 2015 ]