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

- **Assignment 3** due date is Wednesday 11:59pm (Thursday?)
- **Assignment 4** is out, due Friday, February 15th @ 11:59pm

- Group **Projects** form completed
- Project proposals (in class on **Feb 26th**)
Final **Project** (50% of grade total) — **Reminder**

- Group project (groups of 3 are encouraged, but fewer maybe possible)
- Groups are self-formed, you will not be assigned to a group
- You need to come up with a project proposal and then work on the project as a group (each person in the group gets the same grade for the project)
- Project needs to be **research** oriented (not simply implementing an existing paper); you can use code of existing paper as a starting point though

---

Project proposal + class presentation: 15%
Project + final presentation: 35%
Presentation (~5 minutes irrespective of the group size)

1. Clear explanation of the overall problem you want to solve and relationship to the topics covered in class
2. What model/algorithms you planning to explore: at this can be somewhat abstract (e.g., CNN+RNN)
3. The dataset(s) you will use and how will you evaluate performance
4. List of papers you plan to read as references
5. How will you structure the project, who will do what and a rough timeline
Project proposal and class presentation — 15% of grade

**Presentation** (~5 minutes irrespective of the group size)

1. Clear explanation of the **overall problem** you want to solve and relationship to the topics covered in class
2. What **model/algorithms** you planning to explore: this can be somewhat abstract (e.g., CNN+RNN)
3. The **dataset(s)** you will use and how will you **evaluate** performance
4. List of **papers** you plan to read as references
5. How will you **structure the project**, who will do what and a rough timeline

After presentation you will get the feedback from me
Presentation (~5 minutes irrespective of the group size)

1. Clear explanation of the overall problem you want to solve and relationship to the topics covered in class
2. What model/algorithms you planning to explore: this can be somewhat abstract (e.g., CNN+RNN)
3. The dataset(s) you will use and how will you evaluate performance
4. List of papers you plan to read as references
5. How will you structure the project, who will do what and a rough timeline

After presentation you will get the feedback from me

Proposal

— Same as above but in more detail, with well defined algorithms and timeline
— Will be in the form of the PDF document (initial paper draft)
RNNs: Review

Key Enablers:

— Parameter sharing in computational graphs
— “Unrolling” in computational graphs
— Allows modeling **arbitrary length sequences**!
**RNNs: Review**

**Key Enablers:**

- Parameter sharing in computational graphs
- “Unrolling” in computational graphs
- Allows modeling **arbitrary length sequences**

![Vanilla RNN Diagram]

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

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

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

Key Enablers:

— Parameter sharing in computational graphs
— “Unrolling” in computational graphs
— Allows modeling arbitrary length sequences!

Vanilla RNN

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

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

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

Long-Short Term Memory (LSTM)

Vanishing or Exploding Gradients

\[ \begin{pmatrix} i \\ f \\ o \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix} \]

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

\[ h_t = o \odot \tanh(c_t) \]

Uninterrupted gradient flow!
RNNs: Review

**Key Enablers:**

- Parameter sharing in computational graphs
- “Unrolling” in computational graphs
- Allows modeling *arbitrary length sequences*

**Loss functions:** often cross-entropy (for classification); could be max-margin (like in SVM) or Squared Loss (regression)
Sequence Level Training

During training objective is different than at test time

- **Training:** generate next word given the previous
- **Test:** generate the entire sequence given an initial state

Optimize directly evaluation metric (e.g. BLUE score for sentence generation)

Set the problem as a Reinforcement Learning:

- RNN is an Agent
- Policy defined by the learned parameters
- Action is the selection of the next word based on the policy - Reward is the evaluation metric

[Ranzato et al., 2016]

* slide from Marco Pedersoli and Thomas Lucas
Let us look at some actual practical uses of RNNs
Applications: Skip-thought Vectors

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

[Kiros et al., 2015]
Applications: Google Language Translation

One model to translate from any language to any other language

[Johnson et al., 2017]
Applications: Google Language Translation

One model to translate from **any language** to any other language

[ Johnson et al., 2017 ]
Applications: Google Language Translation

One model to translate from *any language* to any other language

Flipped order encoding

Token designating target language

[Johnson et al., 2017]
Applications: Google Language Translation

One model to translate from **any language** to any other language

8! layer LSTM decoder and encoder

Flipped order encoding

Token designating **target** language

[Johnson et al., 2017]
Applications: Google Language Translation

One model to translate from **any language** to any other language

- Residual at other layers (ResNet style)
- Bi-directional at lower layers
- Flipped order encoding

Token designating **target** language

8! layer LSTM decoder and encoder

[ Johnson et al., 2017 ]
Applications: Google Language Translation

One model to translate from **any language** to any other language

- **Bi-directional** at lower layers
- **Residual** at other layers (ResNet style)
- **Flipped** order encoding

Token designating **target** language

8! layer LSTM decoder and encoder

[ Johnson et al., 2017 ]
Applications: Neural Image Captioning
Applications: Neural Image Captioning

Image Embedding (VGGNet)
Applications: Neural Image Captioning

Image Embedding (VGGNet)
Applications: Neural Image Captioning

Image Embedding (VGGNet)

Applications:
- Neural Image Captioning
- Slide from Dhruv Batra
Applications: Neural Image Captioning

Good results

A cat sitting on a suitcase on the floor
A cat is sitting on a tree branch
A dog is running in the grass with a frisbee
A white teddy bear sitting in the grass
Two people walking on the beach with surfboards
A tennis player in action on the court
Two giraffes standing in a grassy field
A man riding a dirt bike on a dirt track

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Neural Image Captioning

Failure cases

A woman is holding a cat in her hand

A person holding a computer mouse on a desk

A woman standing on a beach holding a surfboard

A bird is perched on a tree branch

A man in a baseball uniform throwing a ball

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Image Captioning with Attention

RNN focuses its attention at a different spatial location when generating each word

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

[Xu et al., ICML 2015]
Applications: Image Captioning with Attention

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

[Xu et al., ICML 2015]
Applications: Image Captioning with Attention

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Image Captioning with Attention

\[ z = \sum_{i=1}^{L} p_i v_i \]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Image Captioning with Attention

[Xu et al., ICML 2015]
Applications: Image Captioning with Attention

Xu et al., ICML 2015

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Image Captioning with Attention

[Xu et al., ICML 2015]
Applications: Image Captioning with Attention

[Xu et al., ICML 2015]
Applications: Image Captioning with Attention

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

[ Xu et al., ICML 2015 ]
Applications: Image Captioning with Attention

Good results

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Image Captioning with Attention

Failure results

A large white bird standing in a forest.

A woman holding a clock in her hand.

A man wearing a hat and a hat on a skateboard.

A person is standing on a beach with a surfboard.

A woman is sitting at a table with a large pizza.

A man is talking on his cell phone while another man watches.

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, cs231n Stanford
Applications: Typical Visual Question Answering (VQA)

Image

Question

“How many horses are in this image?”
Applications: Typical Visual Question Answering (VQA)

Image Embedding (VGGNet)

Question

“How many horses are in this image?”
Applications: Typical Visual Question Answering (VQA)

Image Embedding (VGGNet)

Question Embedding (LSTM)

“How many horses are in this image?”
Applications: Typical Visual Question Answering (VQA)

Image Embedding (VGGNet)

Question Embedding (LSTM)

“How many horses are in this image?”

Neural Network Softmax over top K answers

* slide from Dhruv Batra
Applications: Activity Detection

**Activity:** A collection of human/object movements with a particular semantic meaning

**Action Recognition:** Finding if a video segment contains such a movement

**Action Detection:** Finding a segment (beginning and start) and recognize the action in it

[ Ma et al., 2014 ]
Applications: Activity Detection

Early Detection: Recognize when an action starts and try to predict which action is performed as quickly as possible.

[ Ma et al., 2014 ]
Applications: Activity Detection

[ Ma et al., 2014 ]
Applications: Activity Detection

Penalty at every time step is the same

[ Ma et al., 2014 ]
Applications: Activity Detection

Penalty at every time step is the same

[ Ma et al., 2014 ]
Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

- Detecting the correct action class
- More confident that it is not the incorrect action class

[Ma et al., 2014]
Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

- Detecting the correct action class
- More confident that it is not the incorrect action class

[ Ma et al., 2014 ]
Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

- Detecting the correct action class
- More confident that it is not the incorrect action class
Applications: Activity Detection

As the detector sees more of an action, it should become more confident of

- Detecting the correct action class
- More confident that it is not the incorrect action class

[ Ma et al., 2014 ]
New Class of Loss Functions

Training loss at time $t$: 

$$\mathcal{L}^t = \mathcal{L}^t_c + \lambda_r \mathcal{L}^t_r$$

Classification loss at time $t$

Ranking loss at time $t$

$\mathcal{L}^t_r$ is one of the following:

- $\mathcal{L}^t_s$ ranking loss on detection score
- $\mathcal{L}^t_m$ ranking loss on discriminative margin

[ Ma et al., 2014 ]
**Ranking Loss** on Detection Score $\mathcal{L}^t_s$

Ideally what we want:

$$p^\gamma$$

Prediction score of the ground truth action label

[ Ma et al., 2014 ]
Ranking Loss on Detection Score $\mathcal{L}_s^t$

In Practice:

Prediction score of the ground truth action label

[Ma et al., 2014]
Ranking Loss on Detection Score $\mathcal{L}_s^t$

In Practice:

$$p_t^{*y_t} = \max_{t' \in [t_s, t-1]} p_{t'}^{y_t}$$

Prediction score of the ground truth action label

[Ma et al., 2014]
Ranking Loss on Detection Score $\mathcal{L}_s^t$

In Practice:

Prediction score of the ground truth action label

[ Ma et al., 2014 ]
LSTM-m LSTM trained using both classification loss and rank loss on *discriminative margin*.

LSTM-s LSTM trained using both classification loss and rank loss on *detection score*.

**Activity detection performance measured in mAP at different IOU thresholds**

<table>
<thead>
<tr>
<th>Model</th>
<th>$\alpha = 0.1$</th>
<th>$\alpha = 0.2$</th>
<th>$\alpha = 0.3$</th>
<th>$\alpha = 0.4$</th>
<th>$\alpha = 0.5$</th>
<th>$\alpha = 0.6$</th>
<th>$\alpha = 0.7$</th>
<th>$\alpha = 0.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heilbron <em>et al.</em></td>
<td>12.5%</td>
<td>11.9%</td>
<td>11.1%</td>
<td>10.4%</td>
<td>9.7%</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CNN</td>
<td>30.1%</td>
<td>26.9%</td>
<td>23.4%</td>
<td>21.2%</td>
<td>18.9%</td>
<td>17.5%</td>
<td>16.5%</td>
<td>15.8%</td>
</tr>
<tr>
<td>LSTM</td>
<td>48.1%</td>
<td>44.3%</td>
<td>40.6%</td>
<td>35.6%</td>
<td>31.3%</td>
<td>28.3%</td>
<td>26.0%</td>
<td>24.6%</td>
</tr>
<tr>
<td>LSTM-m</td>
<td>52.6%</td>
<td>48.9%</td>
<td>45.1%</td>
<td>40.1%</td>
<td>35.1%</td>
<td>31.8%</td>
<td>29.1%</td>
<td>27.2%</td>
</tr>
<tr>
<td>LSTM-s</td>
<td><strong>54.0%</strong></td>
<td><strong>50.1%</strong></td>
<td><strong>46.3%</strong></td>
<td><strong>41.2%</strong></td>
<td><strong>36.4%</strong></td>
<td><strong>33.0%</strong></td>
<td><strong>30.4%</strong></td>
<td><strong>28.7%</strong></td>
</tr>
</tbody>
</table>

[ Ma et al., 2014 ]
## Applications: Early Activity Detection

Activity early detection performance measured in mAP at different IOU thresholds

<table>
<thead>
<tr>
<th>Model</th>
<th>$\alpha = 0.1$</th>
<th>$\alpha = 0.2$</th>
<th>$\alpha = 0.3$</th>
<th>$\alpha = 0.4$</th>
<th>$\alpha = 0.5$</th>
<th>$\alpha = 0.6$</th>
<th>$\alpha = 0.7$</th>
<th>$\alpha = 0.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>27.0%</td>
<td>23.4%</td>
<td>20.4%</td>
<td>17.2%</td>
<td>14.6%</td>
<td>12.3%</td>
<td>11.0%</td>
<td>10.3%</td>
</tr>
<tr>
<td>LSTM</td>
<td>49.5%</td>
<td>44.7%</td>
<td>38.8%</td>
<td>33.9%</td>
<td>29.6%</td>
<td>25.6%</td>
<td>23.5%</td>
<td>22.4%</td>
</tr>
<tr>
<td>LSTM-m</td>
<td>52.6%</td>
<td>47.9%</td>
<td>41.5%</td>
<td>36.2%</td>
<td>31.4%</td>
<td>27.1%</td>
<td>24.8%</td>
<td>23.5%</td>
</tr>
<tr>
<td>LSTM-s</td>
<td>55.1%</td>
<td>50.3%</td>
<td>44.0%</td>
<td>38.9%</td>
<td>34.1%</td>
<td>29.8%</td>
<td>27.4%</td>
<td>26.1%</td>
</tr>
</tbody>
</table>

**Note:** first 3/10 of activity is seen by a detector

- **LSTM-m** LSTM trained using both classification loss and rank loss on *discriminative margin*.
- **LSTM-s** LSTM trained using both classification loss and rank loss on *detection score*.

**Take home:** Early detection is only 1-3% worse than sewing the whole sequence

[ Ma et al., 2014 ]
Applications: Activity Detection

Background: 0.484
Unloading the car: 0.385
Putting air in tires: 0.018

[ Ma et al., 2014 ]
Attention Models for Action Highlighting

[ Torabi & Sigal, 2017 ]