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

Lecture 10: RNNs (part 3)
-- Assignment 3 due date is Wednesday 11:59pm
Final Project – Reminder

• Group project (groups of 3 are encouraged, but fewer maybe possible)

• Groups are self-formed or random

• 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 and class presentation

**Presentation** (~3-5 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 proposal you will get the feedback from me**
Presentation (~3-5 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

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)
Long-Short Term Memory (LSTM)

**Vanilla RNN**

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

**LSTM**

\[
\begin{pmatrix}
  i \\
  f \\
  o \\
  g
\end{pmatrix} = \begin{pmatrix}
  \sigma \\
  \sigma \\
  \sigma \\
  \tanh
\end{pmatrix} W \left( \begin{array}{c} h_{t-1} \\ x_t \end{array} \right)
\]

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

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

four fully connected layers of size \(|h| \times (|x| + |h|)\) with sigmoid and tanh activation function

[ Hochreiter and Schmidhuber, NC 1977 ]

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

**Vanilla RNN**

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

**LSTM**

\[
\begin{pmatrix}
i \\ f \\ o \\ g
\end{pmatrix} = \begin{pmatrix}
\sigma & \sigma & \sigma & \tanh
\end{pmatrix} W \left( \begin{array}{c} h_{t-1} \\ x_t \end{array} \right)
\]

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

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

* four fully connected layers of size \(|h| \times (|x| + |h|)\) with sigmoid and tanh activation function

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

[Hochreiter and Schmidhuber, NC 1977]
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?

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

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

* slide from Dhruv Batra
**LSTM Intuition: Forget Gate**

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

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

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

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

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

* slide from Dhruv Batra

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

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

*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?

\[
0.1 \
-0.6 \
0.1 \
0.55 \
-0.67 \
0.4 \
0.01 \
0.7 \
... \
0.9 \\
\sigma
\]

\[
\begin{array}{ccc}
0.1 & 0.1 & \ldots \\
-0.6 & 0.6 & \ldots \\
0.1 & 0.5 & \ddots \\
0.55 & 0.3 & \ddots \\
-0.67 & 0.8 & \ddots \\
0.4 & 0.2 & \ddots \\
0.01 & 0.1 & \ddots \\
0.7 & 1 & \ddots \\
... & ... & \ddots \\
0.9 & 0.9 & \ddots \\
\end{array}
\]

\[
x_t = h_{t-1}, x_t + b_f)
\]

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

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

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

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**?

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

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

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

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

* slide from Dhruv Batra
LSTM Intuition: Memory Update

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

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

* slide from Dhruv Batra
LSTM Intuition: Memory Update

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

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

* slide from Dhruv Batra
LSTM Intuition: Output Gate

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

\[
o_t = \sigma(W_o [h_{t-1}, x_t] + b_o)
\]

\[
h_t = o_t \times \tanh(C_t)
\]

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

* slide from Dhruv Batra
**LSTM Intuition: Additive Updates**

Backpropagation from $c_t$ to $c_{t-1}$ only elementwise multiplication by $f$, no matrix multiply by $W$
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!
LSTM Variants: with Peephole Connections

Lets gates see the cell state / memory

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

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

* slide from Dhruv Batra
LSTM Variants: with Peephole Connections

Let's gates see the cell state / memory

\[ f_t = \sigma (W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f) \]
\[ i_t = \sigma (W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i) \]
\[ o_t = \sigma (W_o \cdot [C_t, h_{t-1}, x_t] + b_o) \]
**LSTM Variants:** with Coupled Gates

Only memorize new information when you’re forgetting old

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

* slide from Dhruv Batra
Gated Recurrent Unit (GRU)

No explicit memory; memory = hidden output

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

\( z = \) memorize new and forget old

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

* slide from Dhruv Batra
LSTM/RNN Challenges

- LSTM can remember some history, but not too long
- LSTM assumes data is regularly sampled
Phased LSTM

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

[ Neil et al., 2016 ]
Bi-directional RNNs/LSTMs

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

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

\[ h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h) \]
**Bi-directional RNNs/LSTMs**

\[
y_t = W_{hy}[\vec{h}_t, \overleftarrow{h}_t]^T + b_y
\]

\[
\vec{h}_t = f_W(\vec{h}_{t-1}, x_t)
\]

\[
\overleftarrow{h}_t = f_W(\overleftarrow{h}_{t+1}, x_t)
\]

\[
\vec{h}_t = \tanh(W_{hh} \vec{h}_{t-1} + W_{xh} x_t + \vec{b}_h)
\]

\[
\overleftarrow{h}_t = \tanh(W_{hh} \overleftarrow{h}_{t+1} + W_{xh} x_t + \overleftarrow{b}_h)
\]
Attention Mechanisms and RNNs

Consider a **translation task**: This is one of the first neural translation models.
Attention Mechanisms and RNNs

Consider a translation task with a bi-directional encoder of the source language.
Attention Mechanisms and RNNs

Consider a **translation task** with a bi-directional encoder of the source language:

\[ f = (\text{La, croissance, économique, s'est, ralentie, ces, dernières, années, .}) \]

\[ e = (\text{Economic, growth, has, slowed, down, in, recent, years, .}) \]

Attention Mechanisms and RNNs

Consider a **translation task** with a bi-directional encoder of the source language

Build a **small neural network** (one layer) with softmax output that takes

1. everything decoded so far and (encoded by previous decoder state $Z_i$)
2. encoding of the current word (encoded by the hidden state of encoder $h_j$)

and predicts **relevance of every source word** towards next translation

Attention Mechanisms and RNNs

Consider a **translation task** with a bi-directional encoder of the source language.

Build a **small neural network** (one layer) with softmax output that takes

1. everything decoded so far and (encoded by previous decoder state $Z_i$)
2. encoding of the current word (encoded by the hidden state of encoder $h_j$)

and predicts **relevance of every source word** towards next translation.

---

$\sum a_j = 1$

$c_i = \sum_{j=1}^{T} \alpha_j h_j$

[ Cho et al., 2015 ]

Soft **Attention** in details
Soft **Attention** in details

\[ \beta_{i,t} = \text{score}(h^{(enc)}_i, h^{(dec)}_t) \]

Relevance of encoding at token i for decoding token t
Soft **Attention** in details

\[ \beta_{i,t} = score(h_i^{(enc)}, h_t^{(dec)}) \]

Relevance of encoding at token i for decoding token t

\[ \alpha_{i,t} = \text{Softmax}(\beta_{i,t}) \]

Normalize the weights to sum to 1
Soft **Attention** in details

\[
\beta_{i,t} = \text{score}(h_i^{(\text{enc})}, h_t^{(\text{dec})})
\]

Relevance of encoding at token i for decoding token t

\[
\alpha_{i,t} = \text{Softmax}(\beta_{i,t})
\]

Normalize the weights to sum to 1

\[
c_t = \sum_i \alpha_{i,t} h_i^{(\text{enc})}
\]

Form a context vector that would simply be added to the standard decoder input

![Diagram of Soft Attention](image)
Soft **Attention** in details

\[ \beta_{i,t} = \text{score}(\mathbf{h}^{(enc)}_i, \mathbf{h}^{(dec)}_t) \]  
Relevance of encoding at token i for decoding token t

\[ \alpha_{i,t} = \text{Softmax}(\beta_{i,t}) \]  
Normalize the weights to sum to 1

\[ c_t = \sum_i \alpha_{i,t} \mathbf{h}^{(enc)}_i \]  
Form a context vector that would simply be added to the standard decoder input
Soft **Attention** in details

\[
\beta_{i,t} = \text{score}(h_i^{\text{enc}}, h_t^{\text{dec}}) \quad \text{Relevance of encoding at token } i \text{ for decoding token } t
\]

\[
\begin{align*}
\beta_{i,t} &= \text{score}(h_i^{\text{enc}}, x_t^{\text{dec}}) \\
\beta_{i,t} &= \text{score}(h_i^{\text{enc}}, h_{t-1}^{\text{dec}}) \\
\beta_{i,t} &= \text{score}(h_i^{\text{enc}}, [x_t^{\text{dec}}, h_t^{\text{dec}}])
\end{align*}
\]

\[
\alpha_{i,t} = \text{Softmax}(\beta_{i,t}) \quad \text{Normalize the weights to sum to } 1
\]

\[
c_t = \sum_i \alpha_{i,t} h_i^{\text{enc}} \quad \text{Form a context vector that would simply be added to the standard decoder input}
\]
**Soft Attention in details**

\[
\beta_{i,t} = \text{score}(h_i^{(enc)}, h_t^{(dec)})
\]

Relevance of encoding at token \(i\) for decoding token \(t\)

\[
\beta_{i,t} = \text{score}(h_i^{(enc)}, \mathbf{x}_t^{(dec)})
\]

\[
\beta_{i,t} = \text{score}(h_i^{(enc)}, h_{t-1}^{(dec)})
\]

\[
\beta_{i,t} = \text{score}(h_i^{(enc)}, [x_t^{(dec)}, h_{t-1}^{(dec)}])
\]

**Key:** \(K_i\)

\[
\alpha_{i,t} = \text{Softmax}(\beta_{i,t})
\]

Normalize the weights to sum to 1

\[
c_t = \sum_i \alpha_{i,t} h_i^{(enc)}
\]

Form a context vector that would simply be added to the standard decoder input

Query: \(Q_t\)

Value: \(V_i\)
# Soft Attention in details

<table>
<thead>
<tr>
<th>Name</th>
<th>Alignment score function</th>
<th>Citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content-base attention</td>
<td>$\text{score}(s_t, h_i) = \text{cosine}[s_t, h_i]$</td>
<td>Graves2014</td>
</tr>
<tr>
<td>Additive(*)</td>
<td>$\text{score}(s_t, h_i) = v_a^T \tanh(W_a[s_t, h_i])$</td>
<td>Bahdanau2015</td>
</tr>
<tr>
<td>Location-Base</td>
<td>$\alpha_{t,i} = \text{softmax}(W_a s_t)$</td>
<td>Luong2015</td>
</tr>
<tr>
<td>Note: This simplifies the softmax alignment to only depend on the target position.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>General</td>
<td>$\text{score}(s_t, h_i) = s_t^T W_a h_i$</td>
<td>Luong2015</td>
</tr>
<tr>
<td>Note: $W_a$ is a trainable weight matrix in the attention layer.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dot-Product</td>
<td>$\text{score}(s_t, h_i) = s_t^T h_i$</td>
<td>Luong2015</td>
</tr>
<tr>
<td>Scaled Dot-Product(^)</td>
<td>$\text{score}(s_t, h_i) = \frac{s_t^T h_i}{\sqrt{n}}$</td>
<td>Vaswani2017</td>
</tr>
<tr>
<td>Note: very similar to the dot-product attention except for a scaling factor; where $n$ is the dimension of the source hidden state.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Soft Attention in details

(Additive Attention)

$$\text{softmax}(\text{FFN}([Q; K]))$$

(Dot-Product Attention)

$$\text{softmax}(QK^T)$$

https://tzuruey.medium.com/attention-is-all-you-need-98d26aeb3517
Attention Mechanisms and RNNs

Economic growth has slowed down in recent years.

Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt.

Economic growth has slowed down in recent years.

La croissance économique s’est ralenti ces dernières années.
Self Attention

(Source-Target-Attention)

(Self-Attention)

https://tzuruey.medium.com/attention-is-all-you-need-98d26aeb3517
Transformers: Attention is all you need
**Transformers**: Attention is all you need (Encoder)

**Note**: for assignment you are **not** implementing transformer encoder
Transformers: Attention is all you need (Encoder)

Note: for assignment you are **not** implementing transformer encoder
Transformers: Attention is all you need (Encoder)

Note: for assignment you are not implementing transformer encoder
Transformers: Attention is all you need
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
The FBI is chasing a criminal on the run.
Regularization in RNNs

Standard dropout in recurrent layers does not work because it causes loss of long term memory!

* slide from Marco Pedersoli and Thomas Lucas
Regularization in RNNs

Standard dropout in recurrent layers does not work because it causes **loss of long term memory**!

- Dropout in input-to-hidden or hidden-to-output layers \([\text{Zaremba et al., 2014}]\)
- Apply dropout at sequence level (same zeroed units for the entire sequence) \([\text{Gal, 2016}]\)
- Dropout only at the cell update (for LSTM and GRU units) \([\text{Semeniuta et al., 2016}]\)
- Enforcing norm of the hidden state to be similar along time \([\text{Krueger & Memisevic, 2016}]\)
- Zoneout some hidden units (copy their state to the next timestep) \([\text{Krueger et al., 2016}]\)

* slide from Marco Pedersoli and Thomas Lucas