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

Lecture 11: RNNs (Part 3), Applications
Course Logistics

- Assignment 3 due date is Monday -> Wednesday
- Assignment 4 is released Monday
- Assignment 1 & 2 solutions are out
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**

\[
y_t = W_y h_t + b_y
\]

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

\[
h_t = \tanh(W_h h_{t-1} + W_x x_t + b_h)
\]
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

\[
y_t = W_yh_t + b_y
\]

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

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

### Long-Short Term Memory (LSTM)

**Vanishing or Exploding Gradients**

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

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

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

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)
Soft **Attention** in details

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

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

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

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

\[ c_t = X_i \]

**Attention Layer**

- Context vector
- Global align weights

\[ h^{(\text{enc})}_i \]

\[ y_t, h^{(\text{dec})}_t \]
Soft **Attention** in details

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

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

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

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

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

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

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

Form a context vector that would simply be added to the standard decoder input.
Encoder (English)
You <SOS>
You are my best friend...
You are my best friend...
You are my best friend...
You are my best friend...
You are my best friend...
You are my best friend…
You are my best friend...
You are my best friend…

Encoder (English)

Decoder (Spanish)

Summary Vector
You are my best friend…

Encoder (English)

Summary Vector

Decoder (Spanish)
You are my best friend…

Encoder (English)

Decoder (Spanish)
You are my best friend…
You are my best friend…
You are my best friend...
\[ \text{You are my best friend...} \]
You are my best friend

= Score( $h_1^{\text{enc}}$, $h_1^{\text{dec}}$ )
\[ \beta_{1,1} = \text{Score}(h_1^{(\text{enc})}, h_1^{(\text{dec})}) \]
\[ = \text{Score}(h^{(\text{enc})}_1, h^{(\text{dec})}_1) \]

\[ \beta_{1,1} \]

\[
\begin{array}{cccccc}
\vdots & \vdots & \vdots & \vdots & \vdots & \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \\
1 & 0 & \cdots & 0 & 0 & \\
0 & 0 & \cdots & 1 & 0 & \\
\vdots & \vdots & \cdots & \vdots & \vdots & \\
\end{array}
\]

\[ <\text{SOS}> \quad \text{You} \quad \text{are} \quad \text{my} \quad \text{best} \quad \text{friend} \]

\[ <\text{SOS}> \]
$= \text{Score}(h_2^{(\text{enc})}, h_1^{(\text{dec})})$
\[ \beta_{2,1} = \text{Score}(h_2^{\text{enc}}, h_1^{\text{dec}}) \]
You are my best friend…

\[ = \text{Score}(h_2^{(\text{enc})}, h_1^{(\text{dec})}) \]
You are my best friend…

1 0 0
0 0 1
0 0 0
0 1 0
0 0 0
1 0 0

You are my best friend
You are my best friend...
You are my best friend...
You are my best friend
\[ C_1 = h_1^{(enc)} + h_2^{(enc)} + h_3^{(enc)} + h_4^{(enc)} + h_5^{(enc)} + h_6^{(enc)} \]

\[ \alpha_{1,1} \uparrow + \alpha_{2,1} \uparrow + \alpha_{3,1} \uparrow + \alpha_{4,1} \uparrow + \alpha_{5,1} \uparrow + \alpha_{6,1} \uparrow \]

Softmax

\[ \beta_{1,1} \uparrow + \beta_{2,1} \uparrow + \beta_{3,1} \uparrow + \beta_{4,1} \uparrow + \beta_{5,1} \uparrow + \beta_{6,1} \uparrow \]

\[ h_{1}^{(dec)} \]

Context Vector

<table>
<thead>
<tr>
<th>1</th>
<th>0</th>
<th>...</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>...</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>...</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<\text{SOS}> You are my best friend <\text{SOS}>
\[
C_1 = h_1^{(enc)} + h_2^{(enc)} + h_3^{(enc)} + h_4^{(enc)} + h_5^{(enc)} + h_6^{(enc)}
\]

\[
\begin{align*}
\alpha_{1,1} & \quad \alpha_{2,1} & \quad \alpha_{3,1} & \quad \alpha_{4,1} & \quad \alpha_{5,1} & \quad \alpha_{6,1} \\
\beta_{1,1} & \quad \beta_{2,1} & \quad \beta_{3,1} & \quad \beta_{4,1} & \quad \beta_{5,1} & \quad \beta_{6,1} \\
\end{align*}
\]

\[
\begin{align*}
h_0^{(enc)} & \rightarrow h_1^{(enc)} & \rightarrow h_2^{(enc)} & \rightarrow h_3^{(enc)} & \rightarrow h_4^{(enc)} & \rightarrow h_5^{(enc)} & \rightarrow h_6^{(enc)} \\
\end{align*}
\]
\[ \beta_{1,2} = \text{Score}(h_1^{(enc)}, h_2^{(dec)}) \]
$\beta_{1,2} = \text{Score}(h_1^{(enc)}, h_2^{(dec)})$

We don't have this (we need a proxy)
\[ \beta_{1,2} = \text{Score}(h^{(enc)}_1, \ldots) \]

We don’t have this (we need a proxy)
\[
\beta_{1,2} = \text{Score} ( h^{(\text{enc})}_1, h^{(\text{dec})}_1 )
\]
\[ \beta_{1,2} = \text{Score}(h^{(enc)}_1, \ldots) \]
\[ \beta_{1,2} = \text{Score}(h^{(\text{enc})}_1, h^{(\text{dec})}_1) \]
Additive Attention

$$\beta_{1,2} = \text{Score}(h_{1}^{\text{enc}}, h_{1}^{\text{dec}}) = \text{NN}(\quad)$$
Additive Attention

\[ \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{4,2}, \beta_{5,2}, \beta_{6,2} = \text{NN}( \text{query (replicated)} ) \]

\[ \text{NN} \left( \begin{array}{ccccccc}
\cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots \\
\cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots \\
\cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots \\
\cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots & h_1^{(dec)} & \cdots \\
\cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots \\
\cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots \\
\cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots \\
\cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots & h_1^{(enc)} & \cdots \\
\end{array} \right) \]
Additive Attention

\[ \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{4,2}, \beta_{5,2}, \beta_{6,2} = \text{NN}( \ldots, h_1^{\text{enc}}, h_2^{\text{enc}}, h_3^{\text{enc}}, h_4^{\text{enc}}, h_5^{\text{enc}}, h_6^{\text{enc}}, \ldots ) \]
Dot-product Attention

$$\beta_{1,2} = \text{Score}(h^{(\text{enc})}_1, h^{(\text{dec})}_1) = h^{(\text{enc})}_1$$

Key Query
Dot-product Attention

\[ \beta_{1,2} = \text{Score}(h^{(\text{enc})}_1, h^{(\text{dec})}_1) = h^{(\text{enc})}_1 \]

key \quad \text{query}

\[ \beta_{1,2}, \beta_{2,2}, \beta_{3,2}, \beta_{4,2}, \beta_{5,2}, \beta_{6,2} = h^{(\text{enc})}_1, h^{(\text{enc})}_2, h^{(\text{enc})}_3, h^{(\text{enc})}_4, h^{(\text{enc})}_5, h^{(\text{enc})}_6 \]

keys
General Dot-product Attention

\[ \beta_{1,2} = \text{Score}(W_k, h^{(\text{enc})}_1, W_q, h^{(\text{dec})}_1) \]
Scaled General Dot-product Attention

\[ \beta_{1,2} = \text{Score}(W_k, W_q, h_i^{(\text{enc})}, h_i^{(\text{dec})}) \]

\[ \hat{\beta}_{1,2} = \frac{\beta_{1,2}}{\sqrt{n}} \]

\[ n = \text{Length}(h_i^{(\text{enc})}) \]
Scaled General Dot-product Attention

\[ \beta_{1,2} = \text{Score}( \mathbf{W}_k, \mathbf{W}_q ) \]

\[
\hat{\beta}_{1,2} = \frac{\beta_{1,2}}{\sqrt{n}} \quad n = \text{Length } ( h_i^{(\text{enc})} ) \\
\hat{\beta}_{1,2} = \frac{\beta_{1,2}}{\sqrt{n}} \quad n = \text{Length } ( h_i^{(\text{dec})} )
\]
# 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>General</td>
<td>$\text{score}(s_t, h_i) = s_t^T W_a h_i$</td>
<td>Luong2015</td>
</tr>
<tr>
<td></td>
<td>where $W_a$ is a trainable weight matrix in the attention layer.</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></td>
<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>
</tr>
</tbody>
</table>
Forming a **Context Vector**

\[ C_1 = \alpha_{1,1} \alpha_{2,1} \alpha_{3,1} \alpha_{4,1} \alpha_{5,1} \alpha_{6,1} \]

\[ \beta_{1,1} \beta_{2,1} \beta_{3,1} \beta_{4,1} \beta_{5,1} \beta_{6,1} \]

**Context Vector**
Forming a **Context Vector**

\[
C_1 = \alpha_{1,1} h_1^{(enc)} + \alpha_{2,1} h_2^{(enc)} + \alpha_{3,1} h_3^{(enc)} + \alpha_{4,1} h_4^{(enc)} + \alpha_{5,1} h_5^{(enc)} + \alpha_{6,1} h_6^{(enc)}
\]

\[
\beta_{1,1} + \beta_{2,1} + \beta_{3,1} + \beta_{4,1} + \beta_{5,1} + \beta_{6,1}
\]

**Softmax**
Forming a **Context Vector**

\[
C_1 = h_1^{\text{enc}} + h_2^{\text{enc}} + h_3^{\text{enc}} + h_4^{\text{enc}} + h_5^{\text{enc}} + h_6^{\text{enc}}
\]

**Context Vector**

\[
\alpha_{1,1}, \alpha_{2,1}, \alpha_{3,1}, \alpha_{4,1}, \alpha_{5,1}, \alpha_{6,1}
\]

**Softmax**

\[
\beta_{1,1}, \beta_{2,1}, \beta_{3,1}, \beta_{4,1}, \beta_{5,1}, \beta_{6,1}
\]

**W_v**
Forming a **General Context Vector**

\[ c_1 = h_1^{(enc)} + h_2^{(enc)} + h_3^{(enc)} + h_4^{(enc)} + h_5^{(enc)} + h_6^{(enc)} \]

where \( c_1 \) is the Context Vector, \( h_i^{(enc)} \) are the encoded hidden states, and \( \alpha_i, \beta_i \) are the weights for each hidden state before and after the Softmax layer, respectively.
**Soft Attention in details**

Relevance of encoding at token $i$ for decoding token $t$

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

Query: $Q_t$

Key: $K_i$

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

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

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

Normalized weights to sum to 1

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

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

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

Value: $V_i$
Generalized Soft **Attention** in details

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

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

\[ \beta_{i,t} = \text{score}(W_k h^{(enc)}_i, W_q x^{(dec)}_t) \]
\[ \beta_{i,t} = \text{score}(W_k h^{(enc)}_i, W_q h^{(dec)}_t) \]
\[ \beta_{i,t} = \text{score}(W_k h^{(enc)}_i, W_q [x^{(dec)}_t, h^{(dec)}_{t-1}]) \]

**Key:** \( K_i \)

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

Normalize the weights to sum to 1

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

**Value:** \( V_i \)

Form a context vector that would simply be added to the standard decoder input
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 ralentie ces dernières années.
Self Attention

(SOURCE-TARGET-ATTENTION)

(Self-Attention)

https://tzuruey.medium.com/attention-is-all-you-need-98d26aeb3517
You are my best friend…

Encoder (English)

\[ h_0^{\text{enc}} \rightarrow h_1^{\text{enc}} \rightarrow h_2^{\text{enc}} \rightarrow h_3^{\text{enc}} \rightarrow h_4^{\text{enc}} \rightarrow h_5^{\text{enc}} \rightarrow h_6^{\text{enc}} \]

<\text{SOS}> \quad \text{You} \quad \text{are} \quad \text{my} \quad \text{best} \quad \text{friend}
You are my best friend...
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  (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 (Encoder)

Normalize by sqrt of dimensionality
(leads to more stable gradients)
Transformers: Attention is all you need (Encoder)
Transformers: Attention is all you need (Encoder)
Transformers: Attention is all you need (Encoder)

In practice, we use multiple self-attention heads.
Transformers: Attention is all you need (Encoder)

In practice, we use multiple self-attention heads
Transformers: Attention is all you need (Encoder)

Residual connection with LayerNorm
Transformers: Attention is all you need (Encoder)

Residual connection with LayerNorm
Transformers: Attention is all you need (Encoder)
Transformers: Attention is all you need (Encoder)
Transformers: Attention is all you need (Encoder)
**Transformers**: Attention is all you need
Transformers: Attention is all you need
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.
Benefits of Transformers

1. Tokens are processes in parallel in both encoder and decoder, which is much faster than RNN or LSTM

2. Can (in principle) model infinite history, unlike RNN or LSTM that typically only carries context for relatively small number of steps

3. No gradient flow issues, due to residual architecture design of Transformer layers — similar to LSTM in some sense.
**Benefits** of Transformers

**Note:** In principle Transformer can model RNN-line or LSTM-like recursion by using causal mask and computing relevance based on “positional” information stored in a token representation and context based on “content” information stored in a token

(in other words, it is more or less strict generalization)
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

![Diagram of a neural network model for language translation.](image)

- **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

Flipped order encoding

**Why?**

Token designating **target** 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

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

8! layer LSTM decoder and encoder

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

Node 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]