Recap: Recurrent Neural Networks
why is this good?

RNN **Advantages:**
- Can process any length input
- Model size doesn’t increase for longer input
- Computation for step $t$ can (in theory) use information from many steps back
- Weights are shared across timesteps $\rightarrow$ representations are shared

RNN **Disadvantages:**
- Recurrent computation is slow
- In practice, difficult to access information from many steps back

$\hat{y}^{(4)} = P(x^{(5)} | \text{the students opened their})$

Slides adapted from Mohit Iyyer
RNNs suffer from a **bottleneck** problem

The current hidden representation must encode all of the information about the text observed so far.

This becomes difficult especially with longer sequences.

\[ \hat{y}^{(4)} = P(x^{(5)}| \text{the students opened their}) \]

A representation of “the students opened their”

Slides adapted from Mohit Iyyer
“you can’t cram the meaning of a whole %&@#&ing sentence into a single $*(&@ing vector!”

— Ray Mooney (NLP professor at UT Austin)
Attention
idea: what if we use multiple vectors?

Instead of this, let’s try:

the students opened their = (all 4 hidden states!)

Slides adapted from Mohit Iyyer
The solution: **attention**

- **Attention mechanisms** (Bahdanau et al., 2015) allow language models to focus on a particular part of the observed context at each time step.
  - Originally developed for machine translation, and intuitively similar to *word alignments* between different languages.
How does it work?

- in general, we have a single query vector and multiple key vectors. We want to score each query-key pair

in a neural language model, what are the queries and keys?

---

Query: current time step
Keys: past time steps
Candidates for info to pay attention to

Slides adapted from Mohit Iyyer
Attention mechanisms in neural language models

- Source sentence (input): les pauvres sont démunis
- Decoder RNN
- Attention mechanisms in neural language models

Query 1: How much attention should I pay to each previous word? How similar is each previous word to me?

Keys:
- the
- students
- opened
- their

Query:
- books

Hidden state at current time step

Slides adapted from Mohit Iyyer
Attention mechanisms in neural language models

**Attention scores**

1. **Query 1:** How similar are each of these words to me?

2. **Query 2:** Hidden state at current time step

   - **Encoder RNN**
   - **Source sentence (input)**
     ```
     <START> les pauvres sont démunis
     ```
   - **Decoder RNN**
   - **Attention**
     - **Scores**
       - dot product with keys (encoder hidden states)
     - **Distribution**
       - Take softmax to turn the scores into a probability distribution

- The students opened their books

**Slides adapted from Mohit Iyyer**
Attention mechanisms in neural language models

On this decoder timestep, we’re mostly focusing on the first encoder hidden state ("les").

Compute softmax over the dot products to turn them into a probability distribution.

Reranking/Scoring the usefulness of previous hidden states.

the students opened their books

Slides adapted from Mohit Iyyer
Attention mechanisms in neural language models

At this time step, the attention distribution is focused on the first word of the sequence ("the")

Compute softmax over the dot products to turn them into a probability distribution
Attention mechanisms in neural language models

We use the attention distribution to compute a weighted average of the hidden states.

Intuitively, the resulting attention output contains information from hidden states that received high attention scores.

Slides adapted from Mohit Iyyer
Sequence-to-sequence with attention

- `les pauvres sont démunis` (Encoder RNN)
- `the students opened their books` (Decoder RNN)

Attention output

Attention distribution

Attention scores

Concatenate (or otherwise compose) the attention output with the current hidden state, then pass through a softmax layer to predict the next word.
Sequence-to-sequence with attention

Source sentence (input)

<START> les pauvres sont démunis

decoder, second time step

the students opened their books unwillingly

to

\( \hat{y}_2 \)

Slides adapted from Mohit Iyyer
• **Attention solves the bottleneck problem**
  • Attention allows decoder to look directly at source; bypass bottleneck

• **Attention helps with vanishing gradient problem**
  • Provides shortcut to faraway states

• **Attention provides some interpretability**
  • By inspecting attention distribution, we can see what the decoder was focusing on
  • We get alignment for free!
  • This is cool because we never explicitly trained an alignment system
  • The network just learned alignment by itself

---

Alignments: harder

The balance was the territory of the aboriginal people

Le reste appartenait aux autochtones

Many-to-one alignments

The poor don’t have any money

Les pauvres sont démunis

Many-to-many alignment

Phrase alignment

Alignment as a vector

Mary did not slap the green witch

Maria no dabatena la bruja verde

1 2 3 4 5 6 7

1 3 4 4 4 0 5 7 6

\(a_j = \begin{cases} 1 & \text{if } j \text{ is spurious} \\ 0 & \text{otherwise} \end{cases}\)

• Used in all IBM models
  • \(a_j\) is vector of length \(J\)
  • Maps indexes \(j\) to indexes \(i\)
  • Each \(a_j\) \(\{0, 1, \ldots, I\}\)

---

Slides adapted from Mohit Iyyer
Many variants of attention

• Original formulation: \( a(q, k) = w_2^T \tanh(W_1[q; k]) \)

• Bilinear product: \( a(q, k) = q^TWk \) \hspace{1cm} \text{Luong et al., 2015}

• Dot product: \( a(q, k) = q^Tk \) \hspace{1cm} \text{Luong et al., 2015}

• Scaled dot product: \( a(q, k) = \frac{q^Tk}{\sqrt{|k|}} \) \hspace{1cm} \text{Vaswani et al., 2017}

Slides adapted from Mohit Iyyer
Why dot product?

❖ Dot product provides a measure of similarity between keys and queries.

❖ But you might be wondering: *why do we want to pay attention to words that are similar to the current word?*
Why dot product?

❖ Dot product provides a measure of similarity between keys and queries.

❖ But you might be wondering: *why do we want to pay attention to words that are similar to the current word?*

Consider:

My brother, a **chemist**, was late yesterday because he missed the bus.
When he arrived, he was surprised to find that his lab ________
Why dot product?

- Dot product provides a measure of similarity between keys and queries.
- But you might be wondering: *why do we want to pay attention to words that are similar to the current word?*

Consider:

My brother, a **chemist**, was late yesterday because he missed the bus. When he arrived, he was surprised to find that his lab ________
**SELF-ATTENTION**

Goal: get rid of recurrent connections

If we get rid of them, we can compute in parallel.

![Diagram showing connections between variables](image)
Goal: parallelize computation of $h$

$f: \text{any non-linear function}$

$q_i = f(w_q c_i)$

$k_i = f(w_k c_i)$

$v_i = f(w_v c_i)$

1. Take dot product between $q_3$ & each key $<q_3 k_1, q_3 k_2, q_3 k_3>$

2. Softmax to obtain attention scores

3. Calculate weighted average $h_3 = 0.3v_1 + 0.5v_2 + 0.2v_3$
1. Compute the dot product between $q \cdot \text{all } k$

$$\langle q_2 k_1, q_2 k_2 \rangle$$

2. Softmax

3. Compute the weighted avg:

$$h_2 = 0.3v_1 + 0.7v_2$$