Attention mechanisms

CS 585, Fall 2019
Introduction to Natural Language Processing

Mohit Iyyer
College of Information and Computer Sciences
University of Massachusetts Amherst

some slides from Richard Socher
stuff from last time

• Colab issues :(  
• HW1 time mixup, won’t count anyone who submitted before 11:59pm as late  
• Important dates:
  • Proposal due: Oct 4 (this Friday!!!)  
  • Milestone 1 due: Oct 24  
  • Midterm date: Oct 31  
  • Milestone 2 due: Nov 21  
  • HW 3 due: ???  
  • Poster presentations: Dec 10/12  
  • Final report due: Dec 19  
• Can we spend a lot of time on attention? maybe  
• Final exam instead of final project? NO!
Neural Machine Translation (NMT)

The sequence-to-sequence model

Encoding of the source sentence.
Provides initial hidden state for Decoder RNN.

Encoder RNN produces an encoding of the source sentence.
Neural Machine Translation (NMT)
The sequence-to-sequence model

Encoder RNN produces an encoding of the source sentence.

Encoding of the source sentence. Provides initial hidden state for Decoder RNN.

Source sentence (input)

les pauvres sont démunis

Target sentence (output)

<START> the poor don’t have any money <END>

Decoder RNN is a Language Model that generates target sentence conditioned on encoding.
Training a Neural Machine Translation system

$$J = \frac{1}{T} \sum_{t=1}^{T} J_t = J_1 + J_2 + J_3 + J_4 + J_5 + J_6 + J_7$$

= negative log prob of "the"

= negative log prob of "have"

= negative log prob of <END>

what are the parameters of this model?
Sequence-to-sequence: the bottleneck problem

Encoder RNN

Source sentence (input)
les pauvres sont démunis

Decoder RNN

Target sentence (output)
<START> the poor don’t have any money <END>
Sequence-to-sequence: the bottleneck problem

Encoding of the source sentence. This needs to capture *all information* about the source sentence. Information bottleneck!

Source sentence (input)

Target sentence (output)

<START> the poor don’t have any money <END>
“you can’t cram the meaning of a whole %&@#&ing sentence into a single $*(&@ing vector!”

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

Instead of:
les pauvres sont démunis =

Let’s try:
les pauvres sont démunis =
(all 4 hidden states!)
The solution: attention

• **Attention mechanisms** (Bahdanau et al., 2015) allow the decoder to focus on a particular part of the source sequence at each time step
  • Conceptually similar to *word alignments*
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 machine translation, what are the queries and keys?
Sequence-to-sequence with attention

Encoder RNN

Source sentence (input)

les pauvres sont démunis

<START>

Decoder RNN

Attention scores

dot product with keys (encoder hidden states)

Query 1:
decoder, first time step
Sequence-to-sequence with attention

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

Take softmax to turn the scores into a probability distribution

Source sentence (input): les pauvres sont démunis
Sequence-to-sequence with attention

Use the attention distribution to take a weighted sum of the encoder hidden states.

The attention output mostly contains information the hidden states that received high attention.

Source sentence (input)

les pauvres sont démunis

<START>
Sequence-to-sequence with attention

Encoder RNN

Source sentence (input)

les pauvres sont démunis

Decoder RNN

Attention distribution

Attention scores

Attention output

concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before

$\hat{y}_1$
Sequence-to-sequence with attention

Encoder RNN

RNN

Source sentence (input)

les pauvres sont démunis

Decoder RNN

Attention distribution

Attention scores

Attention output

poor

decoder, second time step

\( \hat{y}_2 \)
Attention is great

- Attention significantly improves NMT performance
  - It’s very useful to allow decoder to focus on certain parts of the source
- 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
| The agreement on the European Economic Area was signed in August 1992. |
Many variants of attention

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

- Bilinear product: \[ a(q, k) = q^T W k \]  
  Luong et al., 2015

- Dot product: \[ a(q, k) = q^T k \]  
  Luong et al., 2015

- Scaled dot product: \[ a(q, k) = \frac{q^T k}{\sqrt{|k|}} \]  
  Vaswani et al., 2017
Attention is not just for MT!
Here we have a standard seq2seq model for summarization
Here we have a seq2seq model with a **copy mechanism** for summarization

See et al., 2017
Target-side attention (in LMs or more complex MT models)

$$p(\text{Yellen}) = g \cdot p_{\text{vocab}}(\text{Yellen}) + (1 - g) \cdot p_{\text{ptr}}(\text{Yellen})$$
Image Captioning with Attention

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.

Xu et al., 2015
visual attention

• Use the question representation $q$ to determine where in the image to look

How many benches are shown?
attention over final convolutional layer in network: 196 boxes, captures color and positional information

How many benches are shown?
Hierarchical attention

Yang et al., 2016
Self-attention as an encoder!
(core component of Transformer)

Vaswani et al., 2017

figure: Graham Neubig
Attention variants
hard attention

attention over final convolutional layer in network: 196 boxes, captures color and positional information

we can use reinforcement learning to focus on just one box

How many benches are shown?

Xu et al., 2015
Multi-headed attention

• Intuition: $k$ different attentions, each of which is computed independently and focuses on different parts of the sentence

• Transformers = stacked layers of multi-headed self-attention
Self-attention

[Vaswani et al. 2017]
Self-attention

[Vaswani et al. 2017]
Self-attention

\[ \text{optics} \quad \text{advanced} \quad \text{who} \quad \text{Strickland} \quad \text{awards} \quad \text{committee} \quad \text{Nobel} \]

\[ Q \quad K \quad V \]

\[ \text{Layer } p \]

\[ \text{Nobel committee awards Strickland who advanced optics} \]

[Vaswani et al. 2017]
Self-attention

[optics, advanced, who, Strickland, awards, committee, Nobel]

[Q, K, V, A]

Layer p

Nobel, committee, awards, Strickland, who, advanced, optics

[Vaswani et al. 2017]
Self-attention

[Vaswani et al. 2017]
Self-attention

[Vaswani et al. 2017]
Self-attention

[Vaswani et al. 2017]
Self-attention

[Vaswani et al. 2017]
Multi-head self-attention

[Vaswani et al. 2017]
Multi-head self-attention

[Vaswani et al. 2017]
Multi-head self-attention

A

Layer p

M_H

Feed Forward

Q

K

V

Layer p+1

M_t

Nobel
c委员会
awards

who

Strickland

advances

optics

[Vaswani et al. 2017]
Multi-head self-attention

[Vaswani et al. 2017]
Multi-head self-attention

Layer 1

Layer p

Layer J

Nobel committee awards Strickland who advanced optics

Multi-head self-attention + feed forward

[Vaswani et al. 2017]