Lecture 14: Sequence to sequence models
CS 182/282A (“Deep Learning”)

2022/03/14
Today’s lecture

• Today is a short lecture covering sequence to sequence (seq2seq) models

• As the name suggests, we aim to convert input sequences to output sequences
  • Machine translation is the canonical example of such a task, and we will focus on this example today

• Prof. John DeNero will come in on Wednesday and tell you a lot more about natural language processing (NLP)!
Seq2seq models

- Typically, what we need to do is read in and process the entire input sequence before attempting to generate the output sequence.
Encoders and decoders

- Pretty much all seq2seq models follow an **encoder-decoder** architecture
- The encoder reads in the input sequence and encodes it into a representation
- The decoder conditions on this representation to decode the output
- Historically, these used to both be LSTMs (with separate parameters)
  - These days, the encoder and decoder are usually both transformers
- The rest of this lecture will sketch out the last ~8 years in seq2seq models
RNN (LSTM) seq2seq, the basic version
Multilayer RNNs

softmax
softmax
softmax

“un”  “chiot”  “mignon”  <start>  “a”  “cute”  “puppy”
What’s wrong with this seq2seq model?

- This naïve seq2seq model suffers from a *bottleneck problem* — all information about the source sequence has to pass through a “direct connection” between the encoder and decoder.

- This can make it difficult if, e.g., the last word (token) we want to decode corresponds to the first word (token) that we encoded.
Attention

softmax

v_1 \rightarrow k_1 

v_2 \rightarrow k_2 

v_3 \rightarrow k_3 

q_1 \rightarrow 

"un" \rightarrow "chiot" \rightarrow "mignon" \rightarrow <start> \rightarrow "a"
Details on, and variants of, attention

- With attention, the “direct connection” between encoder and decoder becomes much less important (though it is still sometimes used).
- We can also make the encoder a bidirectional RNN and attend over its outputs.
- Notice how the value function on the last slide is the identity function.
  - This does not have to be the case — we can also use a learned value function, e.g., linear as we saw previously.
  - We can also go in the other direction and also make the key and query functions identity functions.
Seq2seq transformers

![Diagram of Seq2seq transformers with input and output sequences labeled as "a", "cute", "puppy", and "un", "chiot", "mignon".](image)
Cross attention layers

\[ q_t = q(h_t) \]
\[ k_t = k(z_t) \]
\[ v_t = v(z_t) \]
A recent seq2seq model: T5

https://ai.googleblog.com/2020/02/exploring-transfer-learning-with-t5.html

- The text-to-text transfer transformer (T5) solves many different NLP tasks in a unified text input, text output framework

- It is trained on a massive dataset (called C4) and achieves a number of competitive and state-of-the-art results
One more reminder

- Are these things interesting to you? Do you want to learn more about NLP?
- If not: do you want to do as well as possible on the next midterm?
- If yes to any of the above: come to Prof. John DeNero’s Wed lecture!