CS224N : Lecture 9 - Self-Attention and Transformers

From Recurrence (RNNs) to Attention-Based NLP Models

Issue with recurrent models

1. Recurrent Neural Networks

- RNNs are unrolled “left-to-right”
  - encodes linear locality (nearby words often affect each other’s meanings)
- Problem : RNNs take O(sequence length) steps for distant words pairs to interact
  - Hard to learn long-distance dependencies
  - Linear order of words is “baked in”

2. Lack of Parallelizability

- Forward and backward passes have O(sequence length) unparallelizable operations
  - Future RNN hidden states can’t be computed in full before past RNN hidden states have been computed → Inhibits traning on very large datasets

Word Window

- ~~Alternative for recurrence~~
- Words window models aggregate local contexts
  - Number of unparallelizable operations does not increase sequence length
- Stacking word window layers allows interaction betwen farther words

Attention

- Attention treats each word’s representation as a query to access and incorporate information from a set of values.
- Number of unparallelizable operations does not increase sequence length.

Self-attention

- Attention operates on queries, keys, and values.
- In Self-attention, the queires, ekys, and vlaues are drawn fro teh same source.

Self-attention as an NLP building block

- Can self-attention be a drop-in replacement for recurrence?
    - No, it has few issues.
        1. Self-attention is an operation on sets. It has no inherent notion of order.

Sequence order

- Solution for self-attention problem : self-attention deosn’t build in order information
- Representing each sequence index as a vector

Concatenation of sinusoids

- Sinusoidal position representations : concatenate sinusoidal functions of varyin periods
- Pros
    - Periodicity indicates that maybe “absolute position” isn’t as important
    - Maybe can extrapolate to longer sequences as periods restart
- Cons
    - Not learnable
    - Extrapolation doesn’t really work

Position representation

- Pros
  - Flexibility: each position gets to be learned to fit the data
- Cons
  - Can’t extrapolate to idices outside 1…

Nonlinearity

- At the output of the self-attention block
- Add a feed-forward network to post-process each output vector

Masking

- To enable parallelization → mask out attention to future words by setting attention score to -∞
- Keeps information about the future from “leaking” to the past

Understanding the Tranformer Model

The Transformer Encoder

1. Transformers
2. Multiheaded Attention
3. Residual Connetions
4. Layer Normalization
5. Scaled Dot Product

The Transformer Decoder

- Cross-attention

Great Results with Transformers

1. Machine Translation from the original Transformers papar
2. Document generation

- Transformer’s parallelizability allows for efficient pretraining, and have made them the de-facto standard