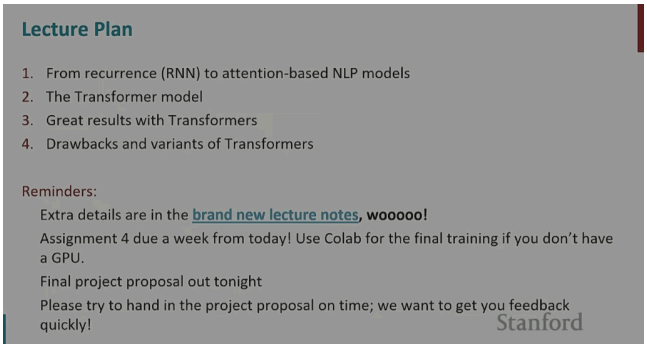
Notes

29 March 2024 08:00

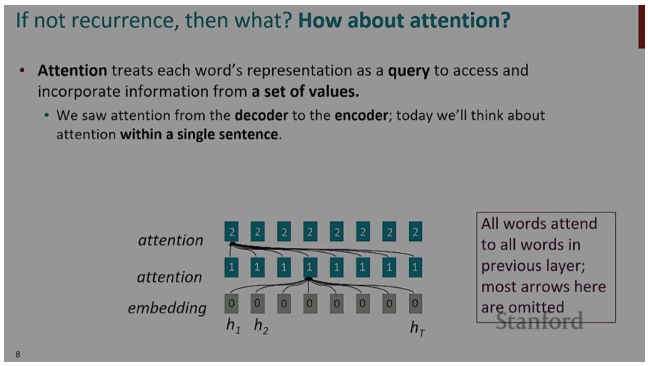
Outline

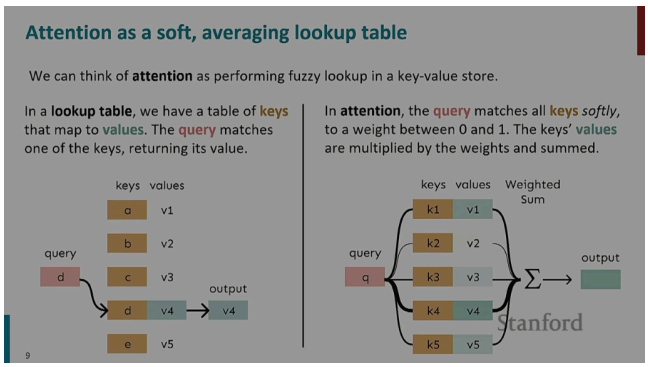


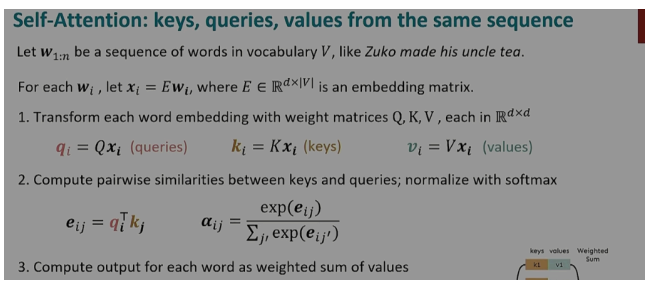
What's wrong with RNNs?

- After discovery of RNNs, the go to way for NLP problems was uni/bi-directional LSTM equipped with convolutional attention
- However RNNs suffer from:
  - Modelling Long-distance dependencies
  - Parallelizability: future hidden state calculations cannot be done before the past hidden states are calculated

Basic Idea of attention





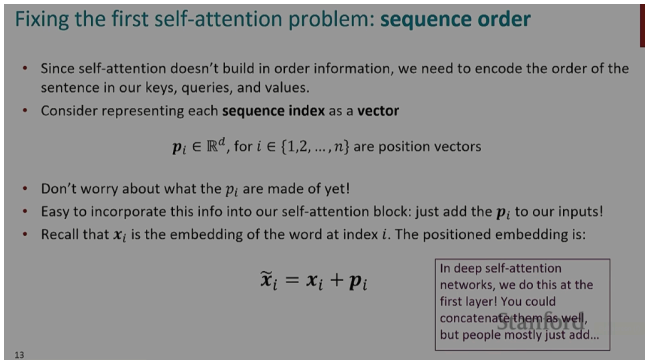


How attention solves problems of RNNs?

- Attention scores can be calculated for all the words in parallel
- (embeddings of) All the words have access to (embeddings of) all other words – so no linear dependency distance

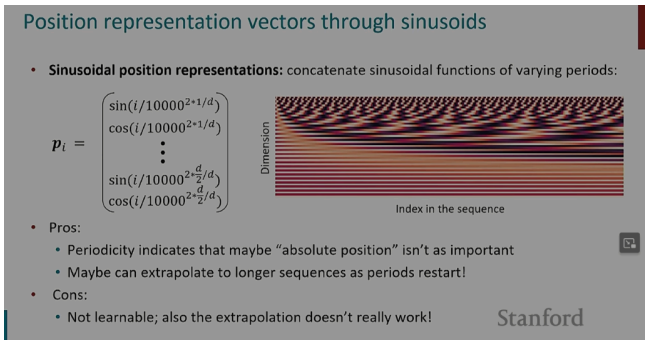
Potential problems with attention

1. Sequence of words is not taken into consideration: Then let's encode the positional information

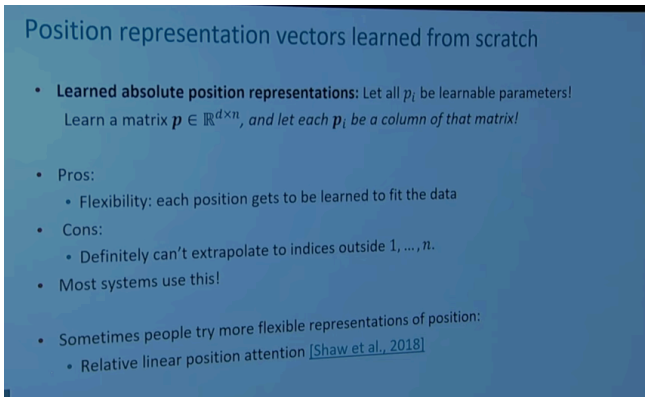


How to encode positional information?

Option 1: Sinusoidal encoding



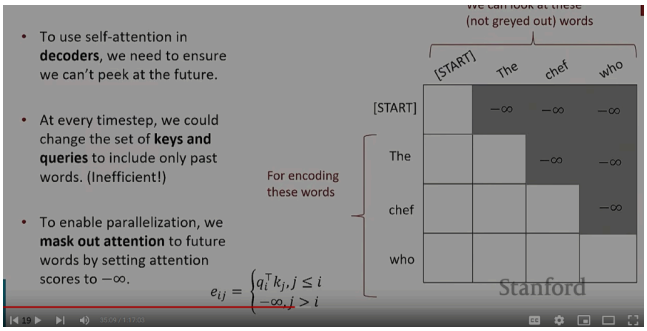
Option 2: Learned Positional encoding



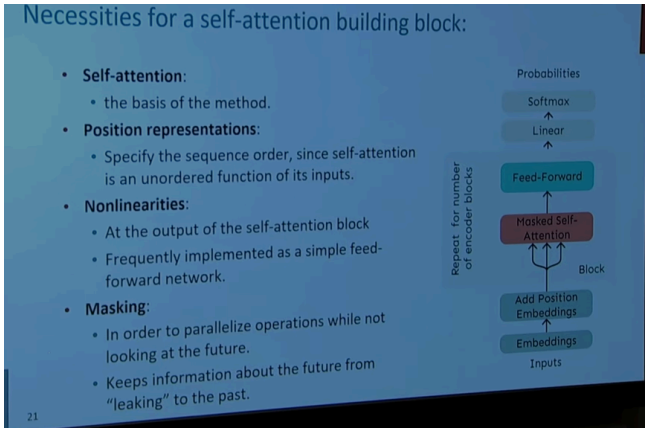
2. no non-linearity -- easy to fix by adding a feed forward network containing non-linearity

3. Not looking into future -- I.e. not looking at words that are yet to come (especially in decoding)

- Restricting q, k to only past words is possible but in-efficient
- Simply mask out words in future



Complete minimal design of self-attention block



Transformers

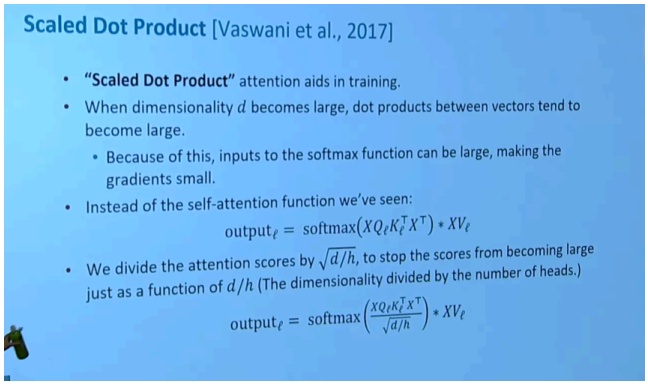
This section (from 40:00 – 56:00) discusses basics of attention/multi-head attention. I would like to refer the **Illustrated Transformer** article written by Jay Alammar - http://jalammar.github.io/illustrated-transformer/

Below are the notes of key modifications introduced in attention mechanism to make transformer architecture work better.

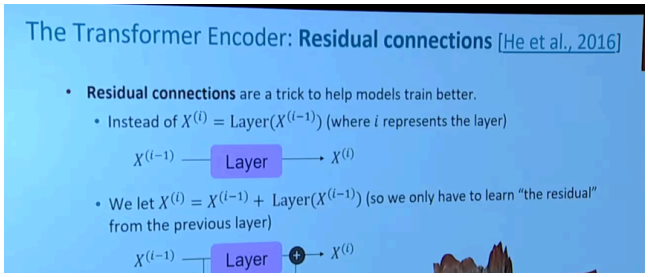
Scaled-dot product attention:

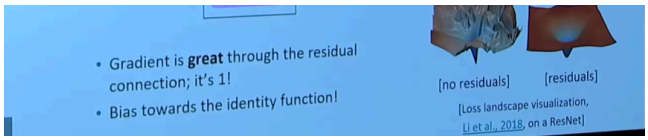
Q: Why to scale (down) the dot product of Q and K?

- Remember gradient of softmax function is softmax(input) x (1-softmax(input)



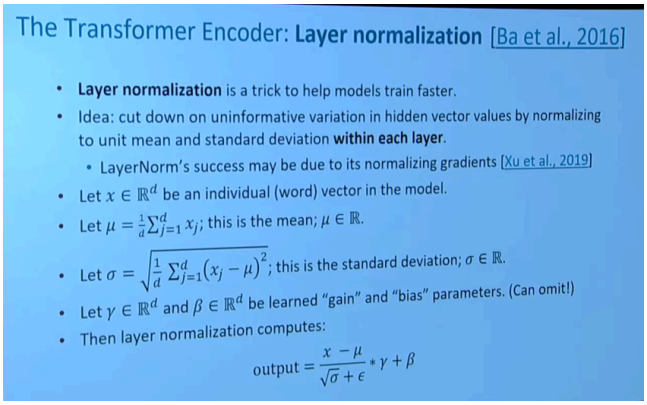
Q: Why to use residual connections in transformer ?





Q: Why to normalization in attention mechanism?

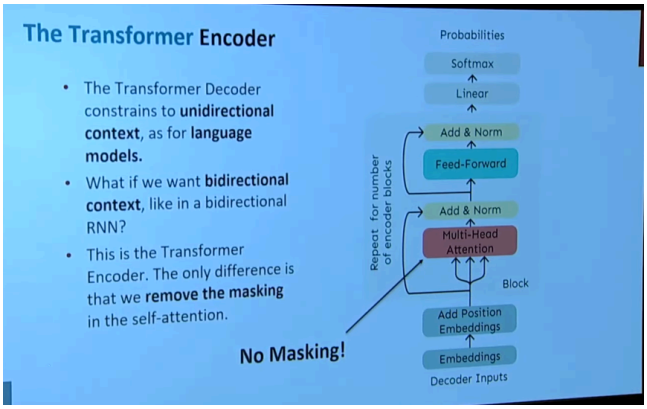
- All the matrix multiplications, non-linearities vastly impact the magnitude of values inside the NN.
- High-variance in magnitude of these values (either input values and/or resulting gradients) introduces numerical instability in training.
- That's why normalize the inputs so that we can maintain unit mean and variance



Transformer Encoder:

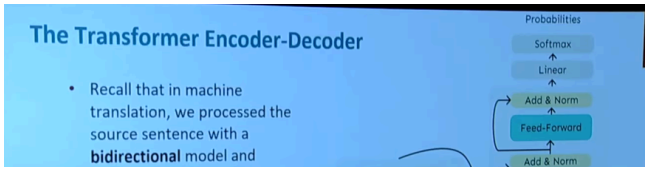
What changes from decoder?

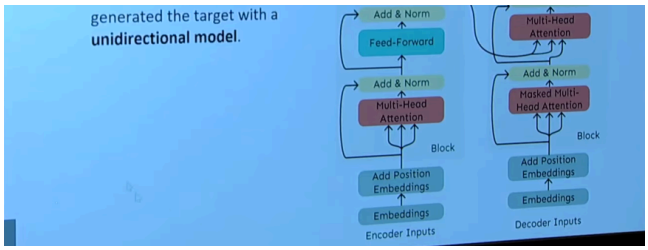
- Only the attention is non-masked (so future tokens/words) are visible to give a sense of bidirectional model



Complete Transformer

- Encoder + decoder put together.
- Still a uni-directional model, to introduce bi-directionality Cross attention is used. Cross attention also introduces a sense of memory (As decoder goes back to encoder again and again to see what encoder has produced and then using attention calculations decoder computes encoder features with high similarity for current word being decoded)
- Cross attention:
  - Using Key and Queries from encoder's output and values from decoder's features. I.e. Decoder using all the information available in encoder for decoding each word.





Drawbacks of transformer / self-attention:

Quadratic complexity of attention calculations wrt. The sequence length

- The RNNs/LSTMs only had linear complexity wrt. Sequence length