Attention

- solution because context vectors forget earlier parts of sequence

Attention Mechanisms

Soft attention
- over each pixel, rating (0,1) ← continuous

Hard
- over patches, rating {0, 1} ← discrete

Global Attention
- similar to soft attention

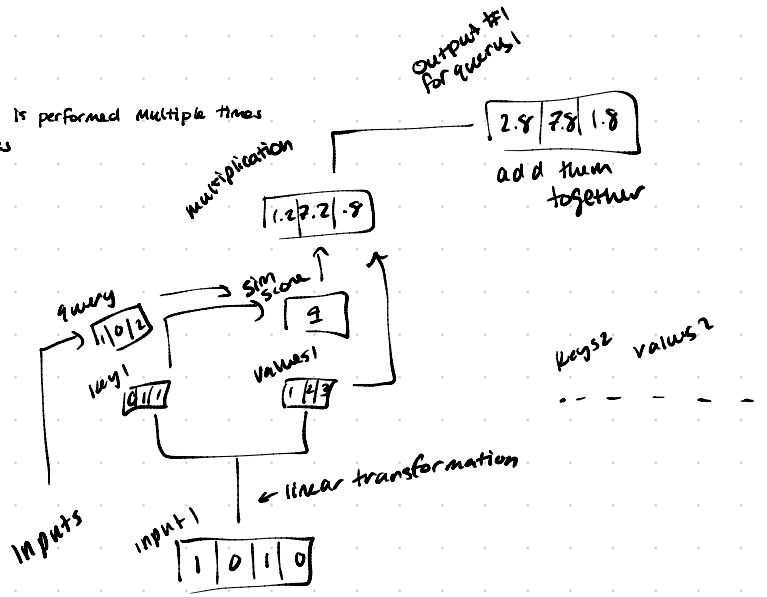
Local attention
- local hard w/ global soft
- only compute attention over small window

Self Attention
→ retrieval
given q, find keys k most similar to q and return the corresponding values v

Attention = $\sum$ sim(q, k) · v
- sim: scaled dot product as similarity function
- v: weight

Multihead self attention
- at every layer, self-attention is performed multiple times
- multiple representation subspaces
  - dif things / semantics

Head 1 → POS
Head 2 → syntax



+ Positional Representations

+ nonlinearities

+ masking : parallelize w/ out looking into the future
used in decoder
avoid data leakage

Encoder - Decoder

2017 Vaswani

translate eng to ital

encode entire sentence
decoder token by token

encoder
self attention
feed forward (nonlinear)
layer norm
residual connections btwn encoder blocks

self attention is multihead attention from prev layers of respective encoder + decoder

decoder
↳ like encoder but w/ multihead attention also using output of encoders
↳ masking to make sure not looking into future

Advantages
- constant path length btwn any two pos in seq
↳ no problem w/ long sequences
↳ wont loose context

- parrallization

Drawbacks
- quadratic time + space
↳ trying to make linear or quasi-linear

GPT-3

- only decoder blocks

$p(w_t \mid w_{1:t-1})$

predict last token given last + tokens

For downstream tasks:

→ train classifier on last hiddenstate
NLI, sentiment classification

→ use pretrained network as generator

In context learning

- few shot settings

ADA

- trained independently
- transformer (like BERT)
- matrioska representation learning
  to reduce space + lower dimensionality

Bidirectional encoder

- process text forward + backward to get all context

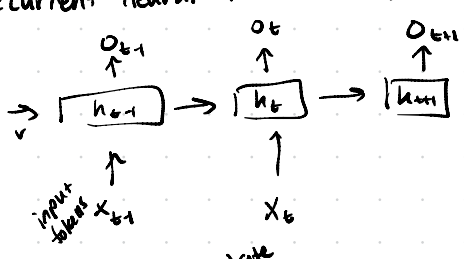
BERT

- transformer architecture
- only encoder ⇒ bc bidirectional can't pretrain naive language task
  - sees entire text ⇒ masked language modeling
    predict words based on context
  - new training objective
    ⇒ given 2 chunks predict if 2nd follows first
- downstream: additional classification layer
- embeddings from last layer corresponding to special token [EOS]

3-Gram Model (1951)

P(word | two preceding words) lookup table

Recurrent Neural Network (2011)



remember persistent state

Big LSTM (2016)

- built on top of RNN
- long term dependencies

Transformer

- long term dependencies improved
  - work =1 regardless of distance

GPT-2 (2019)

GPT-3 (2019)

very big transformer

Autoregressive Generative Model

Deep learning

supervised

AlexNet ⇒ classify images to labels (2012)

⇒ minimal risk

Unsupervised

⇒ unlabeled

⇒ no guarantees