One may argue that most DL models are minor variations of a few core architectures, such as the Transformer [VSP+17], so a reference augmented by a description of the changes should suffice. This would be true if (a) the changes were described precisely, (b) the reference architecture has been described precisely elsewhere, and (c) a reference is given to this description.

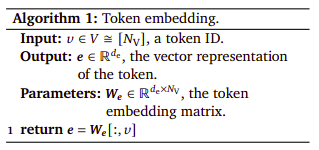
Transformers- good at modeling sequential data

* Sequence modeling (learn an estimation of the distribution)
  + Learn a distribution of a single token given its preceding tokens
  + Language modeling, music generation
* Sequence to sequence prediction (learn an estimation of the conditional distribution)
  + Text2speech, translation
* Classification (learn an estimate of the conditional distribution)
  + Sentiment classification, spam filtering

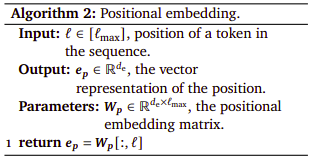
Tokenization

* Character level, word level, subword (most common)

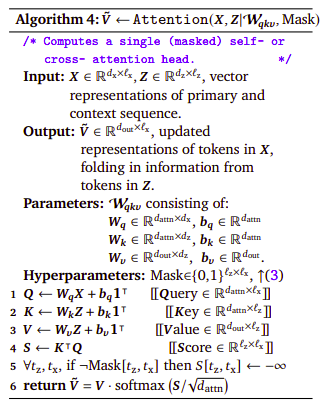
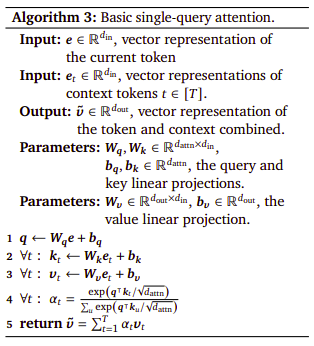
Token embedding



Positional embedding (learned require fixed max length while hard coded don’t such as original transformer paper)



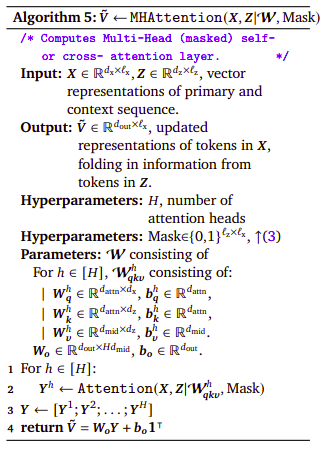
Attention



Bidirectional/ unmasked self- attention (all tokens as context)

Unidirectional/ masked self- attention (all preceding tokens as context)

Cross attention (attention to each token of primary token sequence, treating second token sequence as context)



Layer norm

Unembedding: convert a vector representation of a token and its context into a distribution over the vocabulary elements (sometimes learned, sometimes fixed)

BERT and GPT: main difference is attention masking

* Different activation
* Layer norms positioned differently

Encoder- decoder (seq2seq transformer) (original): used for machine translation which is why it is more complicated than its successors

1. Context sequence is encoded using bidirectional multi head attention
2. Output of this encoder is a vector representation of each context token, taking into account entire context sequence
3. Primary sequence is then encoded
4. Each token in this sequence uses information from encoded context sequence and primary sequence tokens that precede it

BERT (Encoder only)

* Bidirectional transformer trained on masked language modeling
  + Given text with some tokens masked out, recover the masked out tokens
* Doesn’t use mask parameter but each input token is replaced with probability by dummy token and evaluation is based on reconstruction to probability of tokens

GPT-2, GPT-3 Gopher (Decoder only)

* Given incomplete sentence or paragraph, predict next token (autoregressive language modeling)
* Unidirectional attention instead of bidirectional
* GPT-2 and 3 are similar except 3 is larger and other small differences

Gato (multi domain decoder only transformer)

* Each modality converted into a sequence prediction problem by a separate tokenization and embedding method

Tricks for improving performance

* Data preprocessing: cleaning, augmentation, adding noise, shuffling (besides tokenization and chunking)
* Architecture: sparse layers, weight sharing (besides attention)
* Training: improved optimizers, minibatches, batch normalization, learning rate scheduling, weight initialization, pretraining, ensembling, multi-task, adversarial (besides layer normalization)
* Regularization: weight decay, early stopping, cross-validation, dropout, adding noise
* Inference: scratchpad prompting, few-shot prompting, chain of thought, majority voting