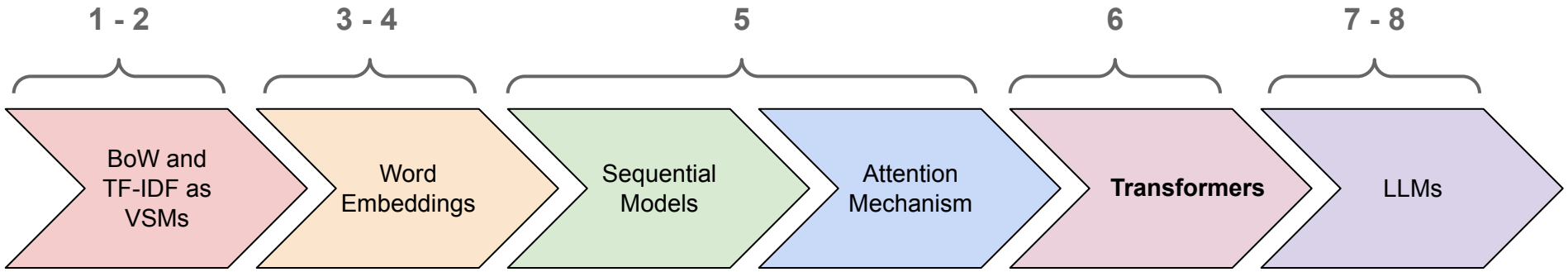
Code.Hub

The first Hub for Developers

Attention and Transformer Networks

Thanos Tagaris

NLP timeline up till today...

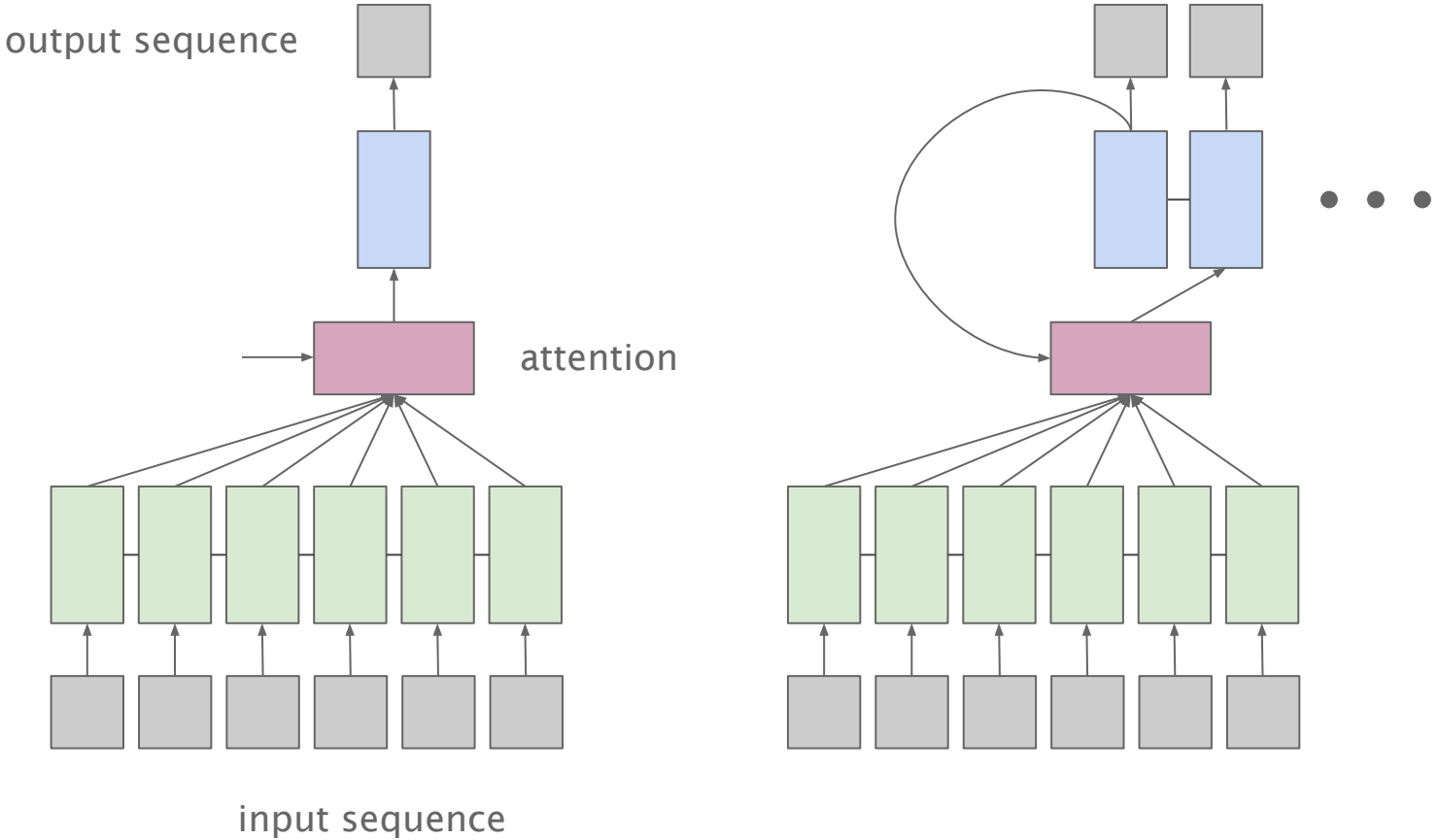


the attention mechanism was so influential that it gave birth to a new class of Neural Networks called **Transformers**

Recurrent Neural Networks

- Encoder-decoder architecture
- Can transform arbitrary-length sequential inputs to arbitrary-length sequential outputs
- Naturally utilize sequential nature of inputs
- Issues:
  - context vector needs to encode all information concerning the input sequence
  - this becomes challenging as sequences become longer and longer
- Attention to the rescue!

Attention Mechanism recap



Challenges with RNNs

- Modelling long-range dependencies
- Challenges in training: vanishing/exploding gradients
- Large number of training steps
- Recurrence **prevents parallel computation**

Idea: throw away all recurrent computation

Transformers

- If we throw away all recurrent layers what building blocks remain?
- Turns out…

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Transformers

- Encoder - Decoder architecture
- Has no recurrent layers
- Has only attention and FC layers
- Usually HUGE in size; require tremendous computation effort
- Highly parallelizable!

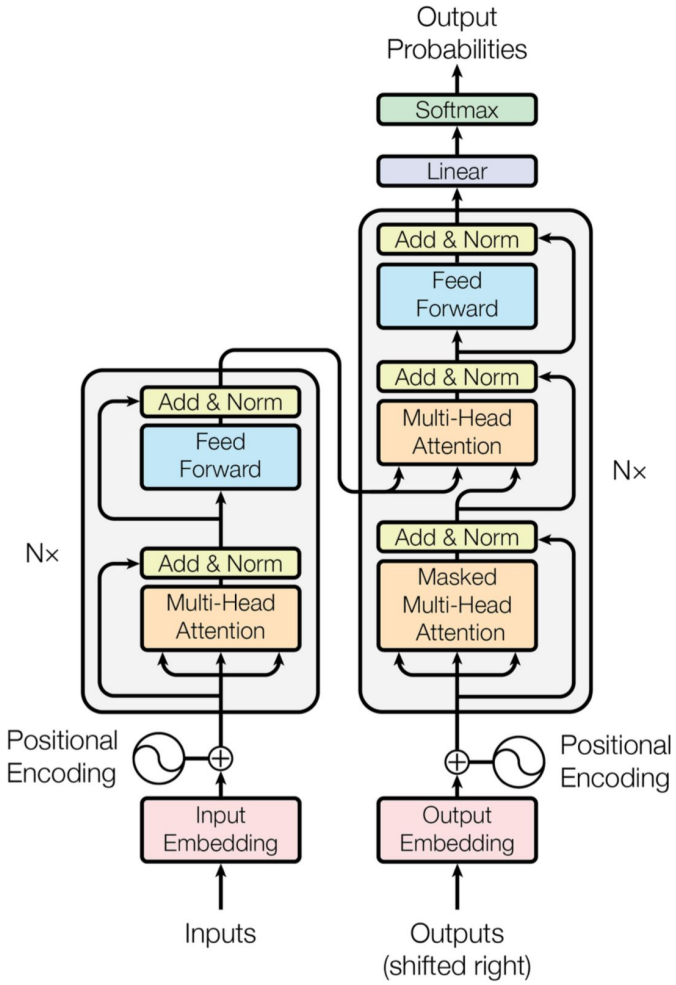


Figure 1: The Transformer - model architecture.

Transformers: Encoder

Nx means we have this block N times: first block looks at pairs of words, higher level blocks consider information from larger parts of the input

compute the attention between every word and every other word; essentially produces a better embedding by combining information from the rest of the sequence
This is referred to as **self-attention**

component that “embeds” the inputs (trainable embeddings)

input sequence

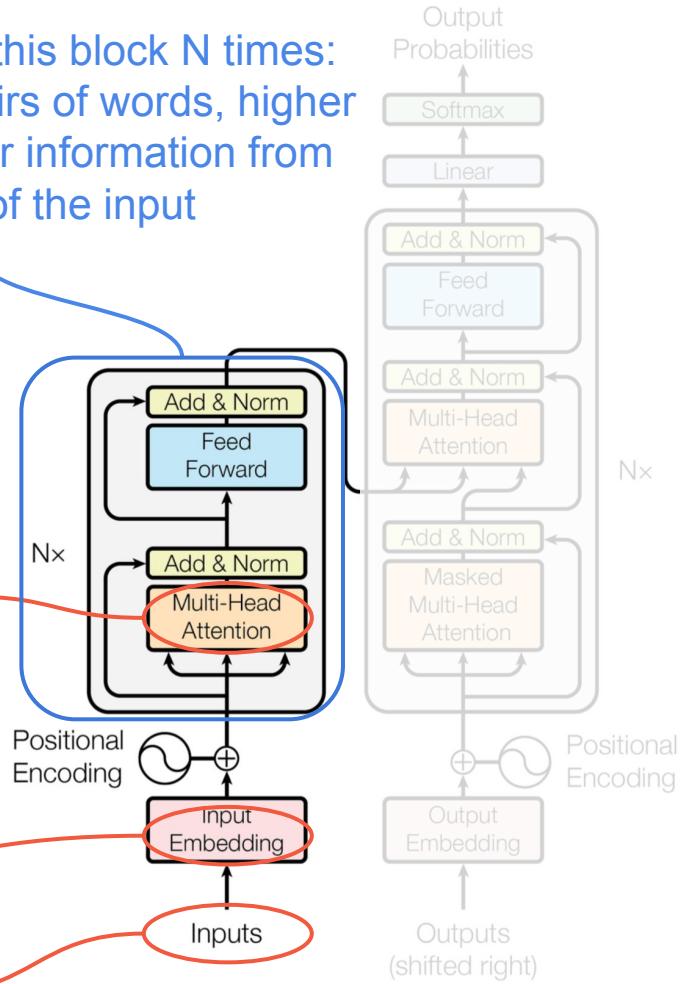
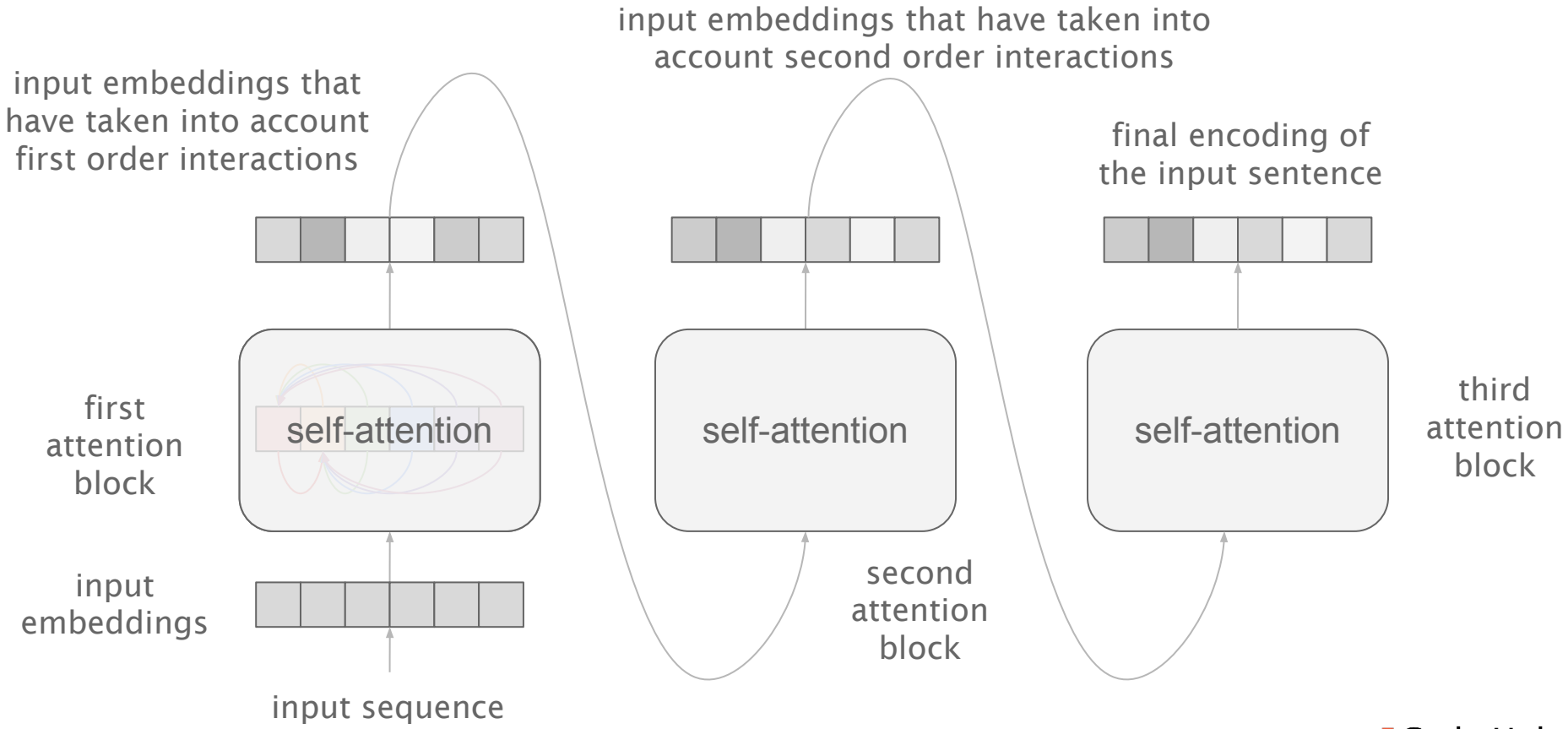


Figure 1: The Transformer - model architecture.

Transformers: Encoder (intuition)



Code.Hub

Transformers: Decoder

output is a label for each position

combine each output word with each input word; this is the type of attention we saw in the previous week

similar to encoder, combine each output word with the rest of the output words. *Note: because this is the output sequence, each word can only attend to its previous words, thus "**masked**"*

output sequence embeddings

output sequence

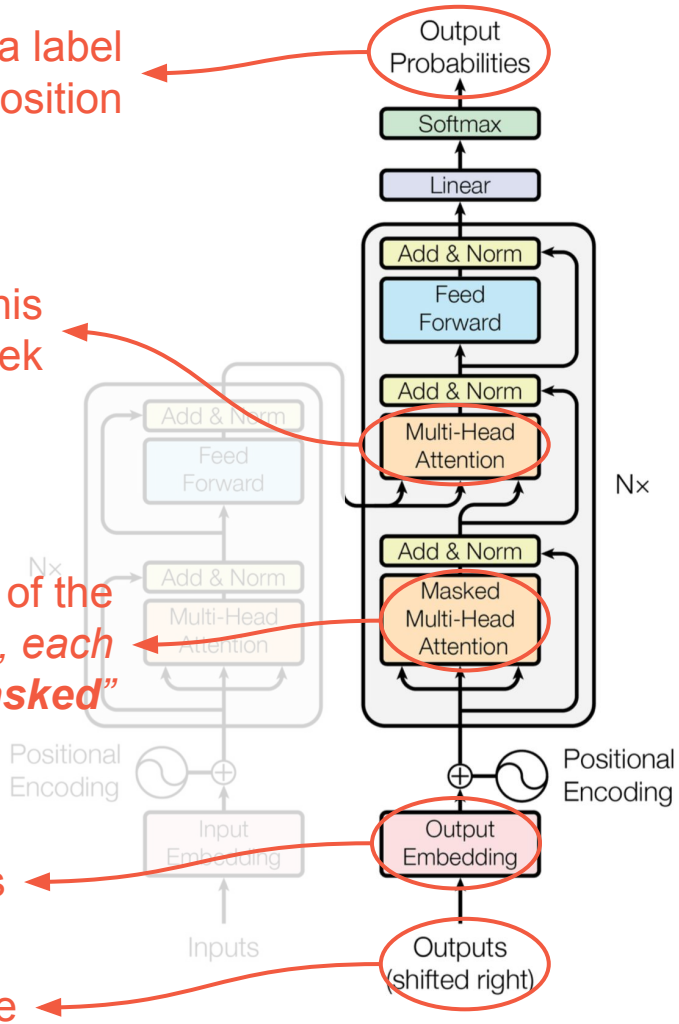
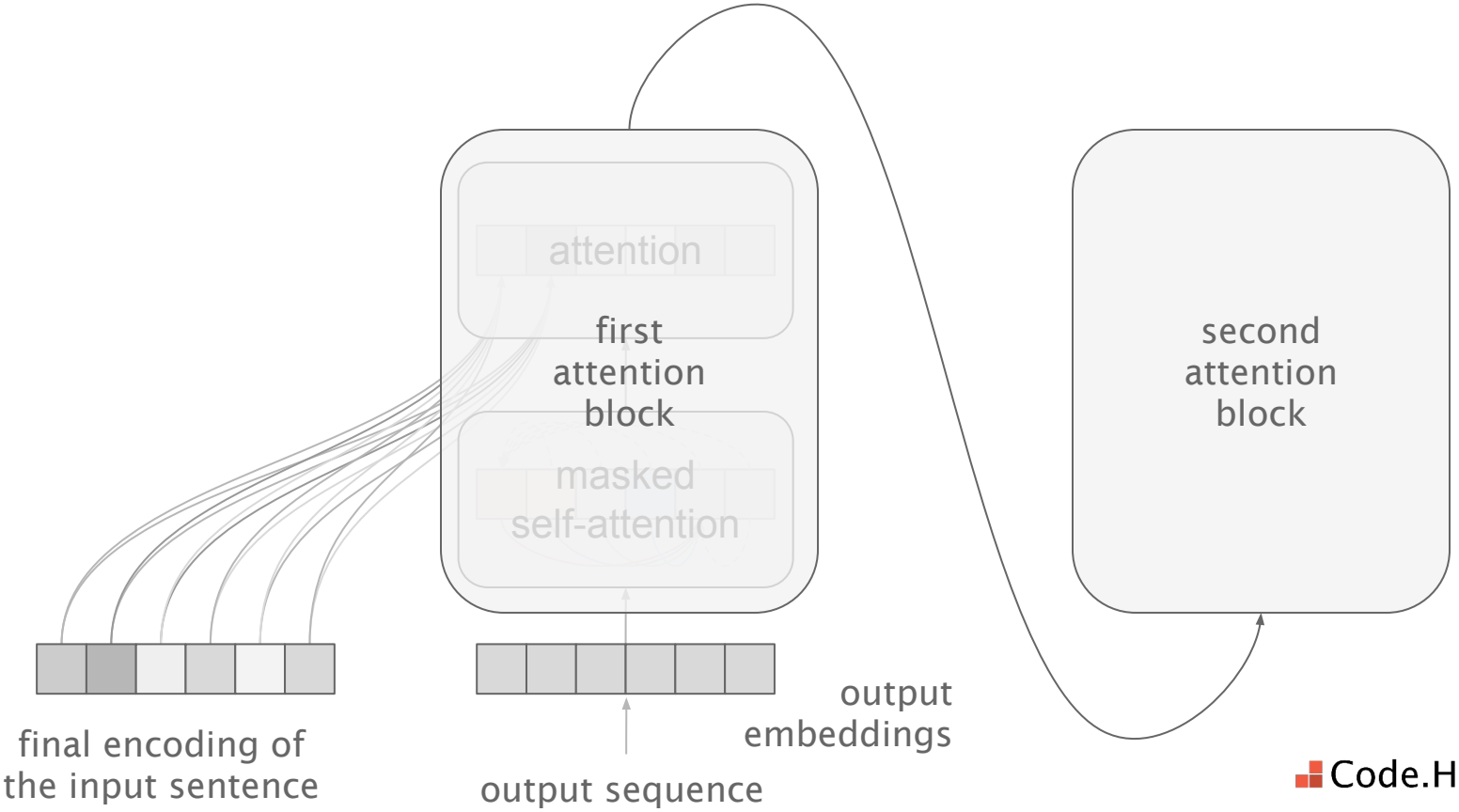


Figure 1: The Transformer - model architecture.

Transformers: Decoder (intuition)



Code.Hub

Transformers:
layer normalization

layer normalization

helps optimize convergence and stabilize training

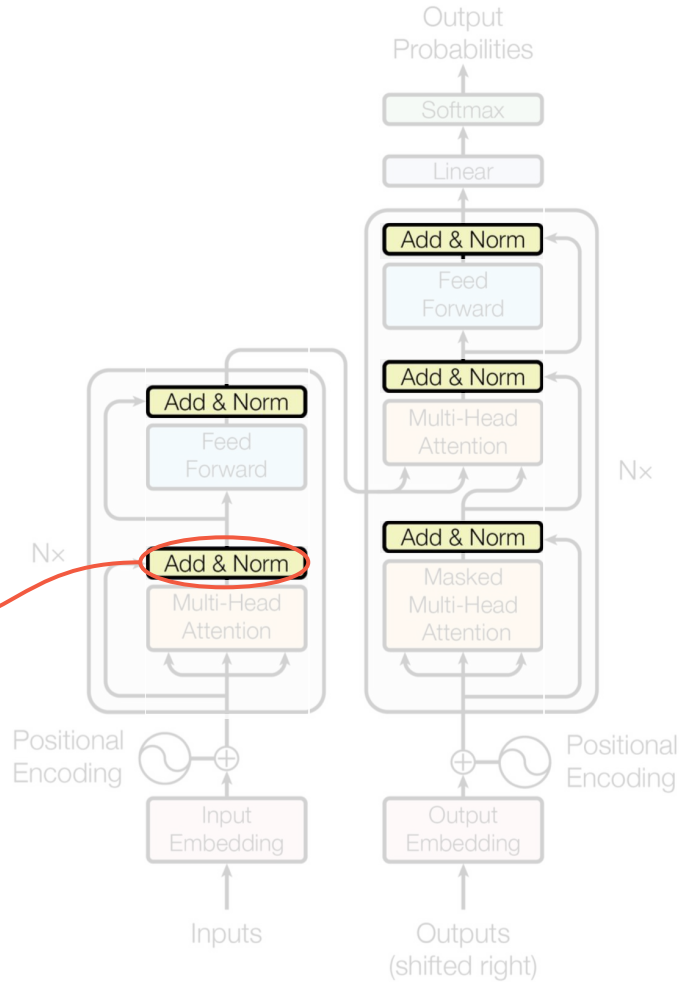


Figure 1: The Transformer - model architecture.

Transformers: positional encoding

- Attention layers on their own don't take into account word ordering

positional encoding

trick we do to change the embedding to take into account the position of the word

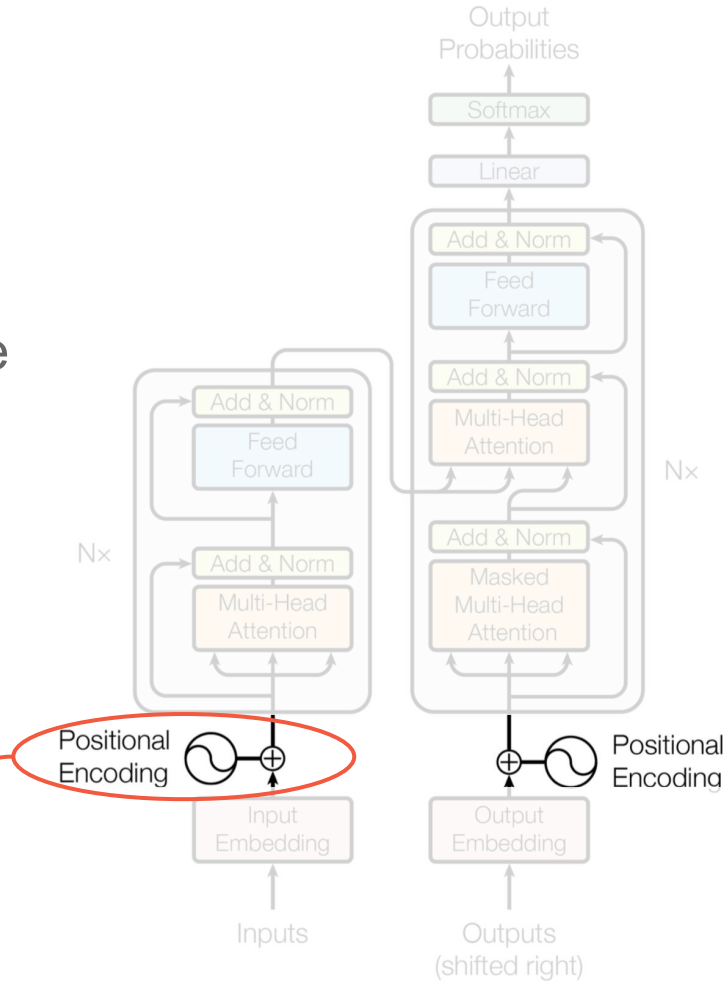
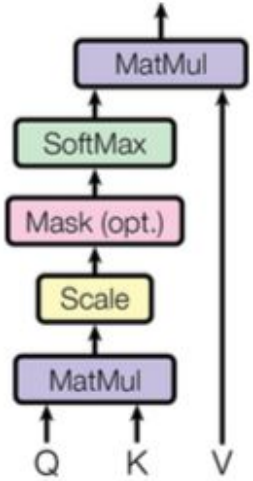


Figure 1: The Transformer - model architecture.

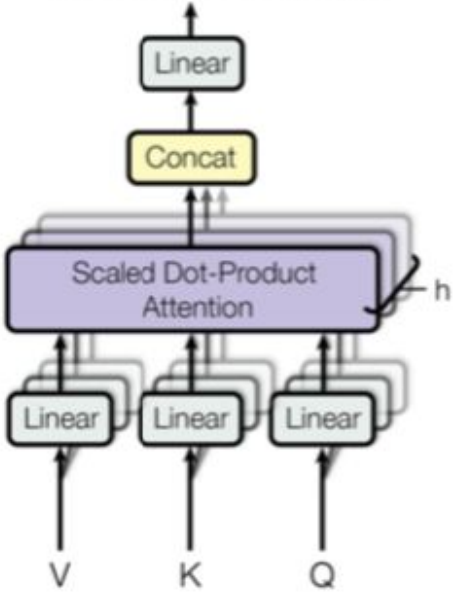
Attention: a deeper look…

$$\mathrm{A}(\mathrm{q},\mathrm{K},\mathrm{V}) = \sum_i \frac{\exp\left(e_{qk_i}\right)}{\sum_{\mathrm{j}} \exp\left(e_{qk_j}\right)} v_i$$

Scaled Dot-Product Attention



Multi-Head Attention



Code.Hub

Post-transformer landscape in NLP

Post-Transformer landscape in NLP

- Transformers introduced a new class of language models that did not rely on RNNs and thus could be parallelized
- This idea turned out to be very influential in NLP (and not only)

GPT and GPT-2

- Generative Pretrained Transformer (probably)
- GPT (117M params)
  - Language models are unsupervised multi-task learners
  - Pretext task: predict next word based on previous words
  - Because of this, no need for encoder in Transformer model is a stack of Transformer decoders
  - SOTA in zero-shot* setting for 7/8 language tasks
- GPT-2 (1.5B params)
  - Focused more on its generative capabilities

**no task learning, only unsupervised pre-training*

Bidirectional Encoder Representations from Transformers

a.k.a. BERT

- Architecture: stack of Transformer encoders
- Pretext tasks:
  - Masked Language Modelling: *predict word based on previous and next words*
  - Next sentence prediction: *given two sentences A and B, predict if B is likely to be the next sentence to A*
- Mask missing word with masked multi-head attention
- Methodology:
  - pre-train in an unsupervised manner
  - fine-tune on the task we want to solve
- This is what popularized SSL as the de-facto approach to language modelling
- Improved SOTA on 11 tasks

Beyond BERT: RoBERTa

- Robustly Optimized BERT Pretraining Approach
- Remove next sentence prediction
- Increase batch size via gradient accumulation
- Use larger vocabulary

Beyond BERT: ALBERT

- BERT models are huge:
  - BERT base → 108 million params
  - BERT large → 334 million params
- Is such a huge network necessary?
- ALBERT uses 2 ways to reduce model size:
  - Factorized embedding parametrization
  - Cross-layer parameter sharing

GPT-3

- Same architecture as GPT-2
- HUGE model (175B parameters)
- Trained on large corpora
- Nuff said…

Next steps...

- Circa 2021 researchers figured out that these LMs were much more capable than previously thought.
- They simply needed to be adapted to better serve user intent
- This gave rise to the chatbots we are familiar with

GPT-4

- Released on Tuesday! (Not yet available to general public)
- Context window 32768 tokens (compared to 2048 of GPT-3)
- Multimodal (can accept images as input along with text)
- Architectural details, training details, param count are not yet known

Some other notable models…

- DistilBERT
  - KD → 40% smaller, 60% faster, retains 97% NLU capabilities
- ELECTRA
  - sample-efficient pretraining
- LLaMA → Alpaca
  - Smaller GPT3.5 → ChatGPT counterparts by Meta and Stanford
- Bard
  - Google’s counterpart to ChatGPT