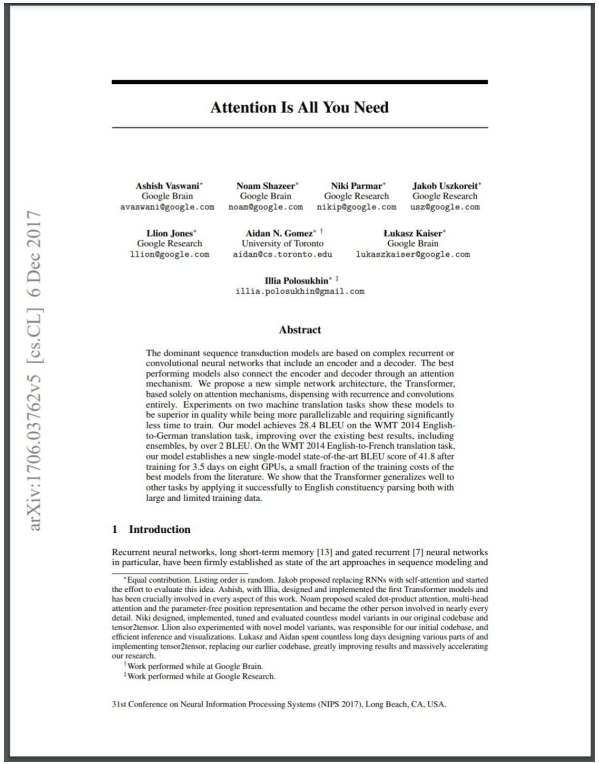
Attention is All You Need

A gentle introduction to Transformers

A **transformer** is a deep learning model that adopts the mechanism of **self-attention** to learn context in sequential data

"Attention is all you need" Ashish Vaswani et al. (2017)

Google Brain



arXiv:1706.03762v5 [cs.CL] 6 Dec 2017

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.

1 Introduction

Recurrent neural networks, long short-term memory [13] and gated recurrent [7] neural networks in particular, have been firmly established as state of the art approaches in sequence modeling and

*Equal contribution. Listing order is random. Jakob proposed replacing RNNs with self-attention and started the effort to evaluate this idea. Ashish, with Illia, designed and implemented the first Transformer models and has been crucially involved in every aspect of this work. Noam proposed scaled dot-product attention, multi-head attention and the parameter-free position representation and became the other person involved in nearly every detail. Niki designed, implemented, tuned and evaluated countless model variants in our original codebase and tensor2tensor. Llion also experimented with novel model variants, was responsible for our initial codebase, and efficient inference and visualizations. Lukasz and Aidan spent countless long days designing various parts of and implementing tensor2tensor, replacing our earlier codebase, greatly improving results and massively accelerating our research.

†Work performed while at Google Brain.

‡Work performed while at Google Research.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

2017 — **NLP**

Eg., machine translation, language modeling and named entity recognition

2020 — **Computer Vision**

E.g., image classification, object detection, image generation and video processing

2021 — **Audio**

E.g., speech recognition, speech synthesis, speech enhancement and music generation

2022 — **Time Series**

E.g., forecasting, anomaly detection, and classification

Outline

- Introduction
- General architecture
- Popular models
- Applications

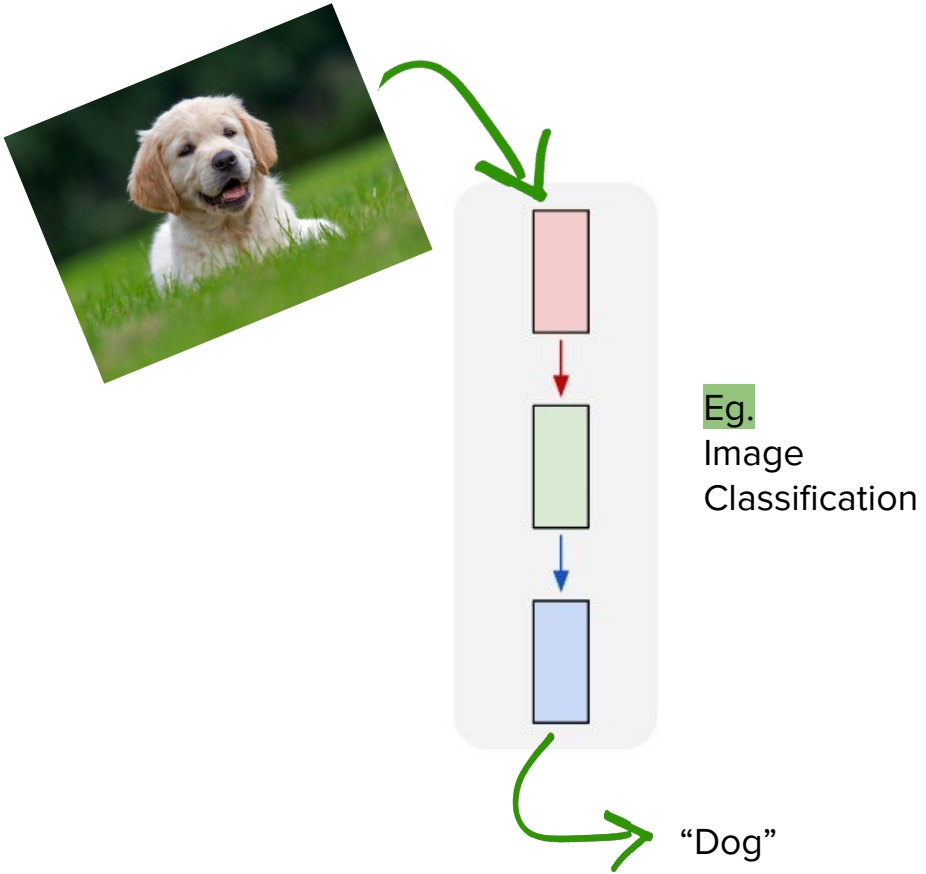
Introduction

Main schemes

- One to one
- One to many
- Many to one
- Many to many

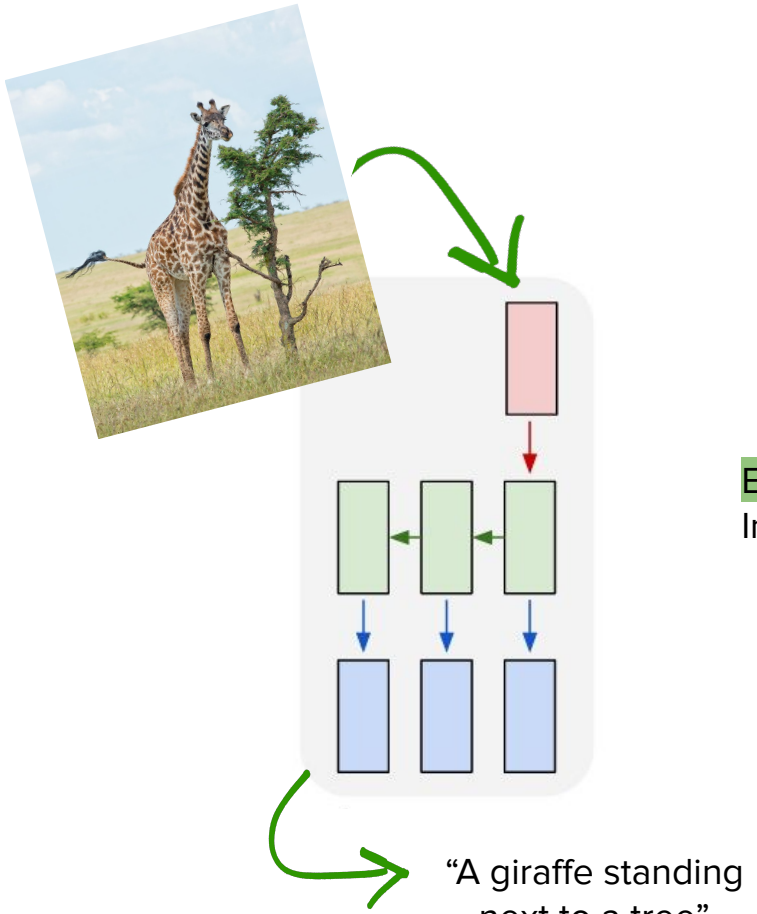
Main schemes

- One to one
- One to many
- Many to one
- Many to many



Main schemes

- One to one
- One to many
- Many to one
- Many to many

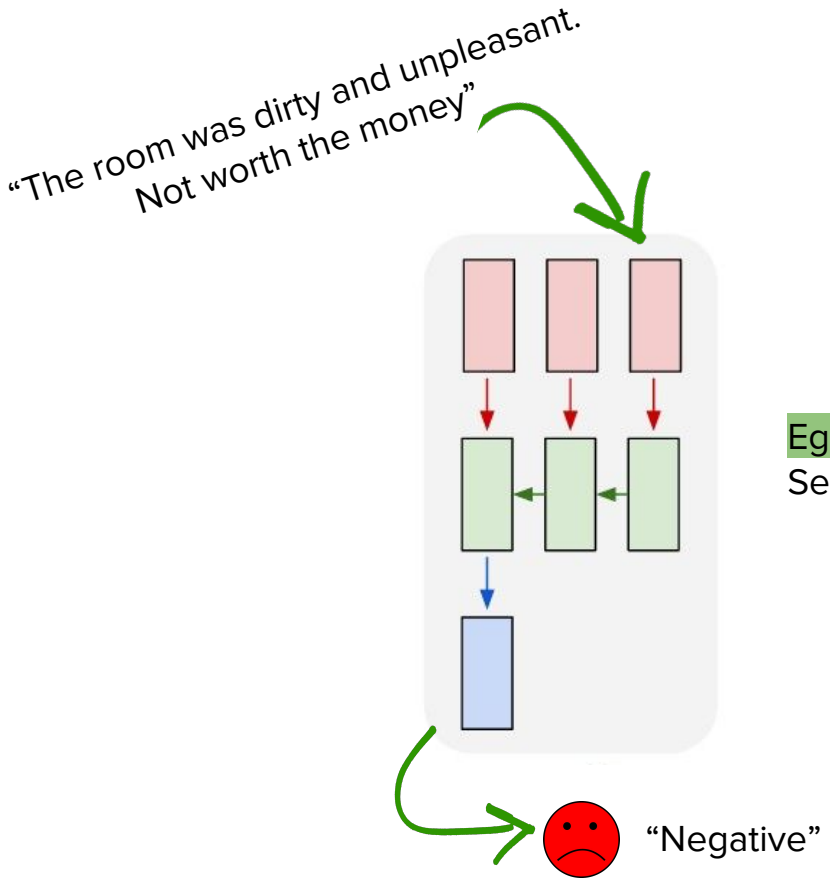


“A giraffe standing next to a tree”

Eg.
Image Captioning

Main schemes

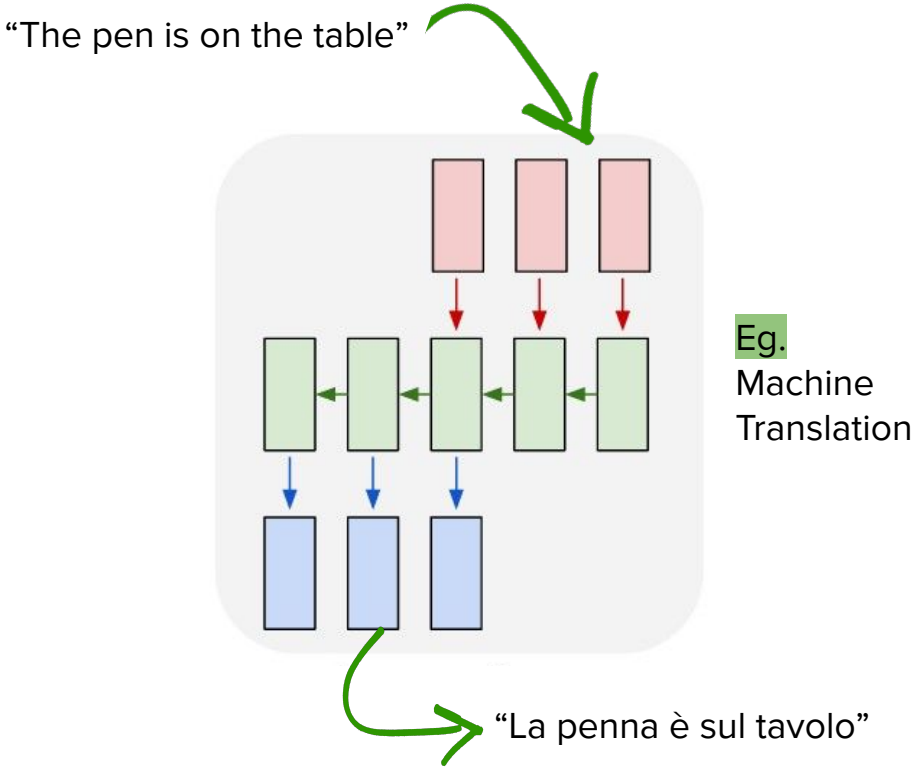
- One to one
- One to many
- Many to one
- Many to many



Eg.
Sentiment Analysis

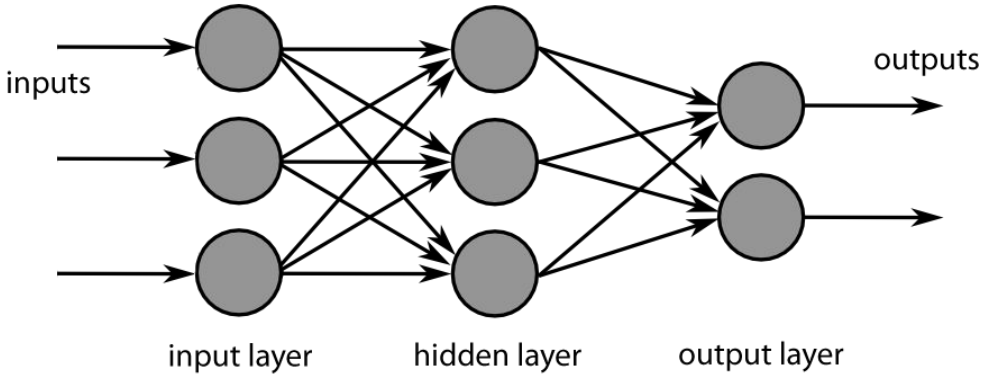
Main schemes

- One to one
- One to many
- Many to one
- Many to many



Eg.
Machine
Translation

How to deal with sequential data?

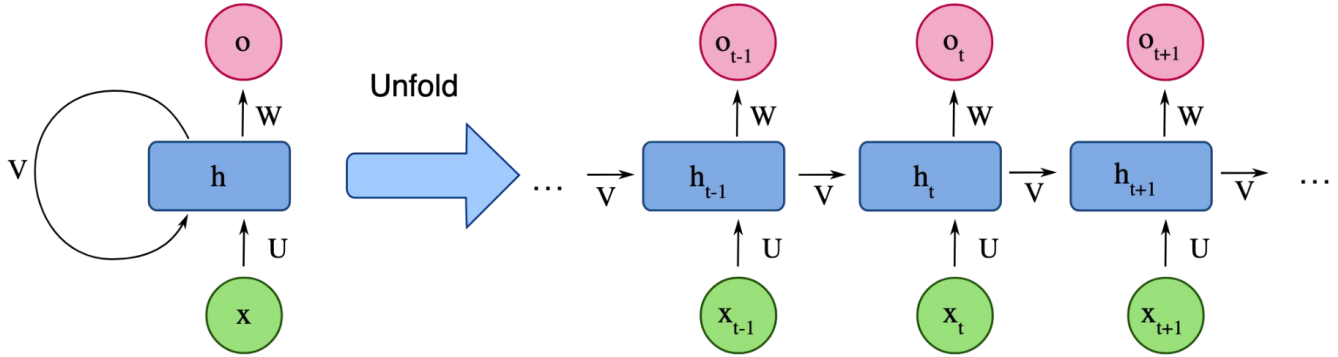


Feed-forward neural networks

Each input is mapped into an output

Not designed to keep track of sequential data

How to deal with sequential data?



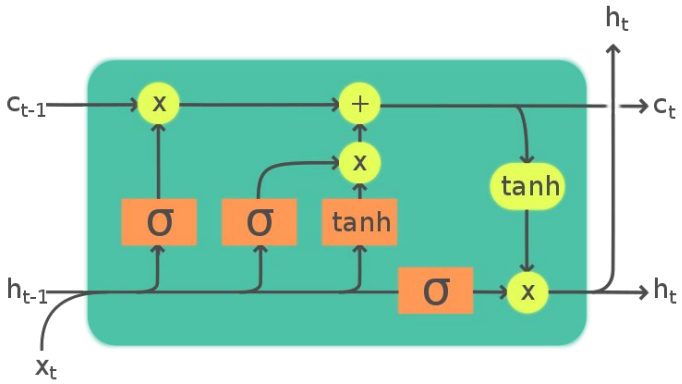
Recurrent neural networks

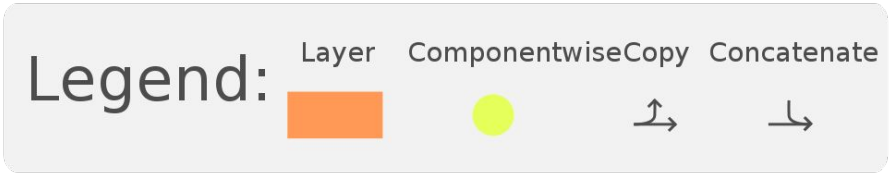
Keep track of
the entire sequence

Very slow
(data is processed
sequentially)

Cannot handle long sequences
(*vanishing gradients*)

How to deal with sequential data?



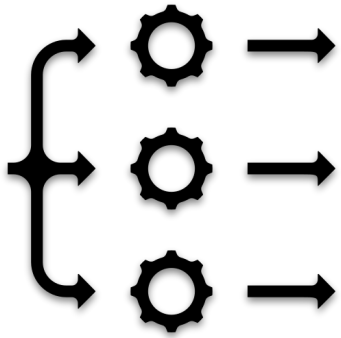


LSTMs

- Keep track of the entire sequence
- Solve the *vanishing gradient* problem

Slower than RNNs

Solution: use Transformers



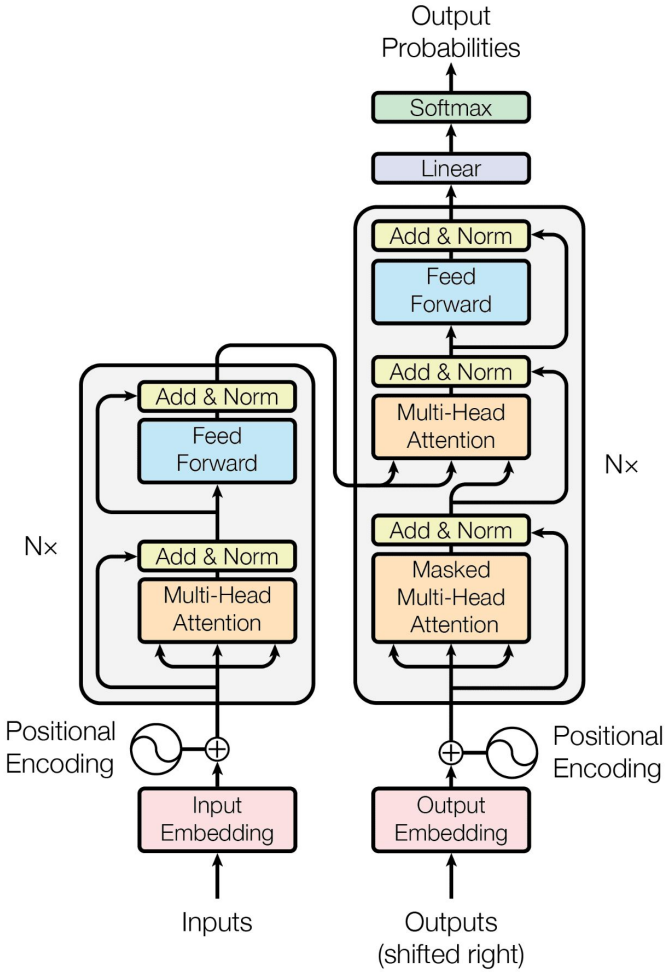
Sequences can be processed in **parallel**

“**Attention**” can track relations between items in very long sequences

General architecture

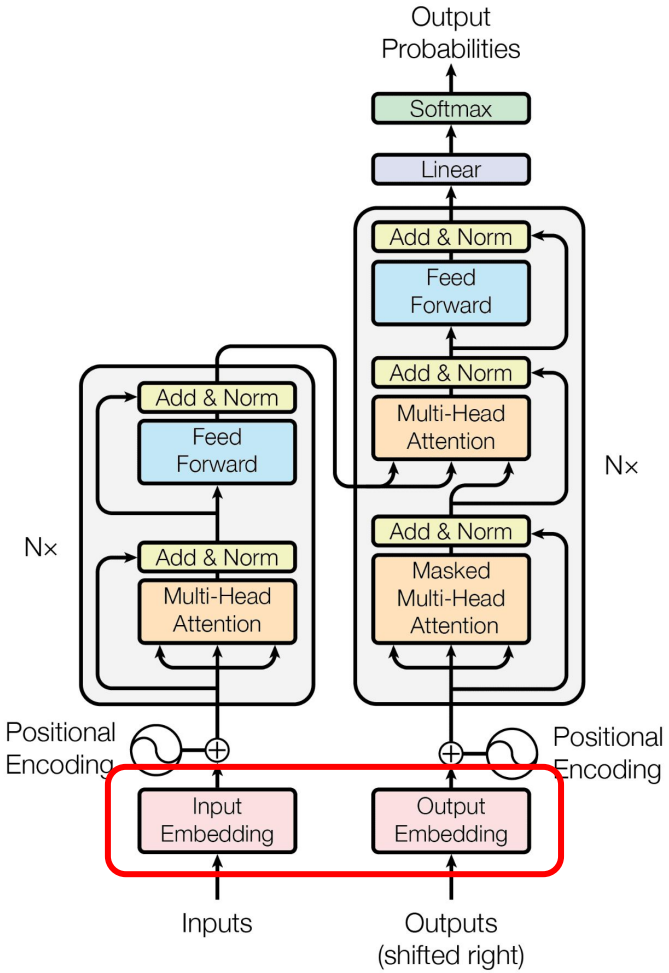
Model structure

- Encoder-decoder structure
- **Encoder**: input sequence ➔ sequence of continuous representations
- **Decoder**: output of the encoder & of the decoder at the previous step ➔ output sequence
- Does NOT rely on recurrence and convolution

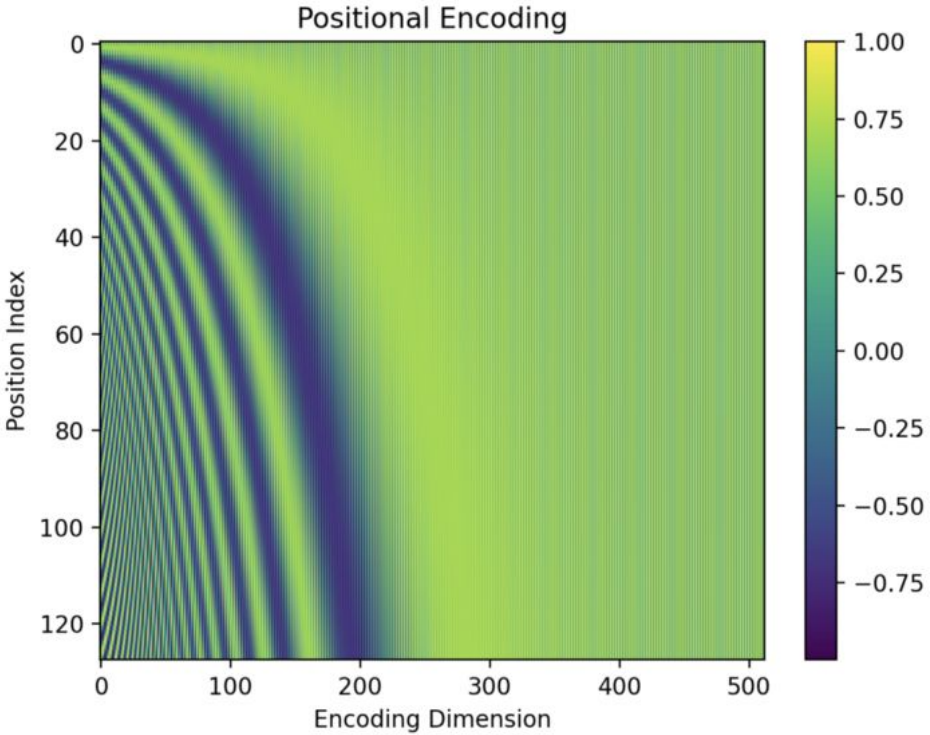


Model structure

- Encoder-decoder structure
- **Encoder**: input sequence ➔ sequence of continuous representations
- **Decoder**: output of the encoder & of the decoder at the previous step ➔ output sequence
- Does NOT rely on recurrence and convolution



Positional encoding



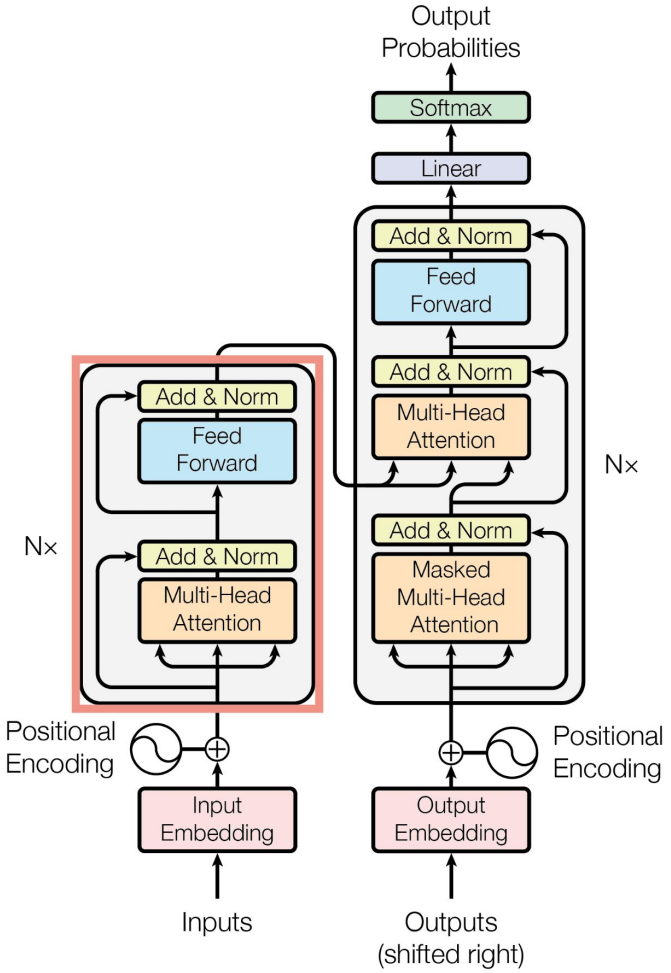
NO recurrence or convolution

Information about position must be injected with positional encoding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

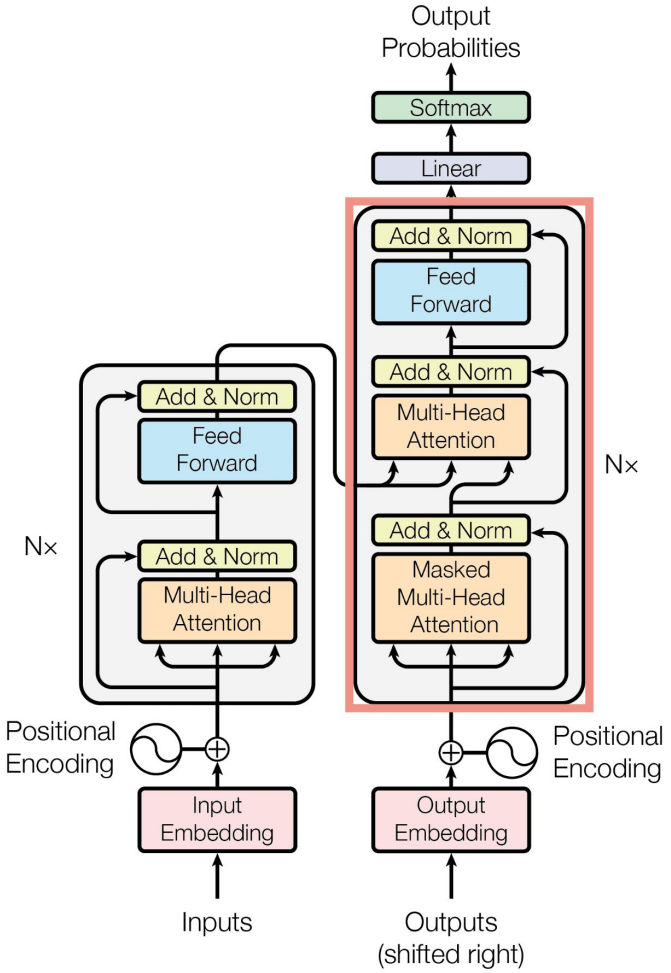
Encoder

- Consists of a stack of *N* identical layers
- Each layer is composed of:
  1. Multi-head self-attention
  2. Fully connected feed-forward network
- Each sublayer has a residual connection and is followed by a normalisation layer



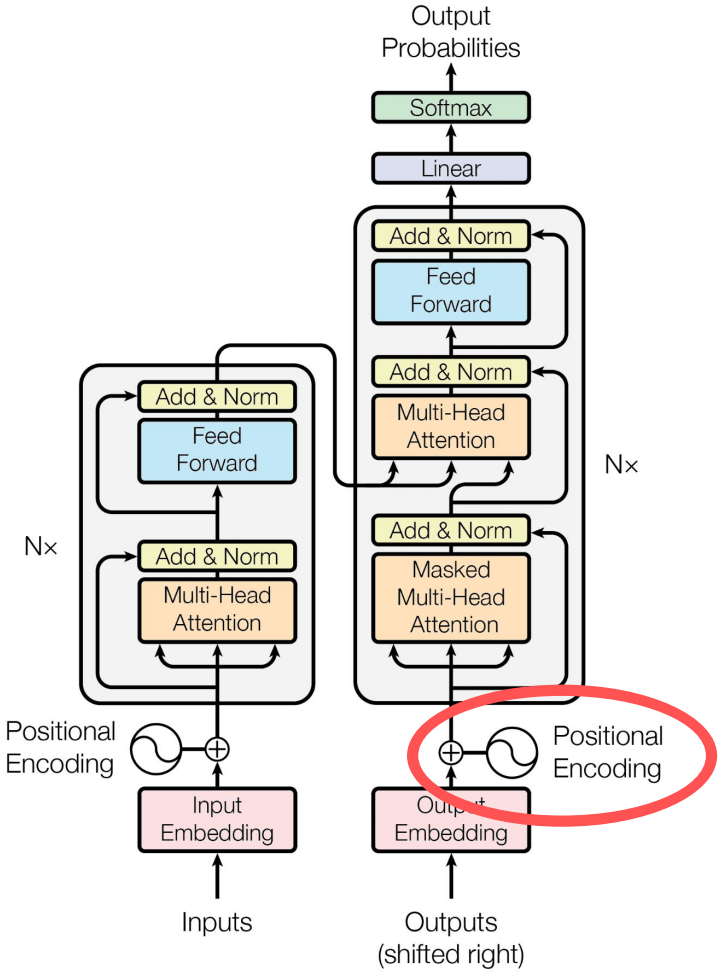
Decoder

- Consists of a stack of *N* identical layers
- Each layer is composed of:
  1. Masked multi-head self-attention
  2. Multi-head self-attention
  3. Fully connected feed-forward network
- Each sublayer has a residual connection and is followed by a normalisation layer



Decoder

- Consists of a stack of *N* identical layers
- Each layer is composed of:
  1. Masked multi-head self-attention
  2. Multi-head self-attention
  3. Fully connected feed-forward network
- Each sublayer has a residual connection and is followed by a normalisation layer



What is attention?

> *“In its most generic form, **attention** could be described as merely an overall level of alertness or ability to engage with surroundings.*
>
> – Attention in Psychology, Neuroscience, and Machine Learning, 2020

What is attention?

In neural networks, we give more importance to some parts of the data than to others depending on the **context**

Scaled Dot-Product Attention

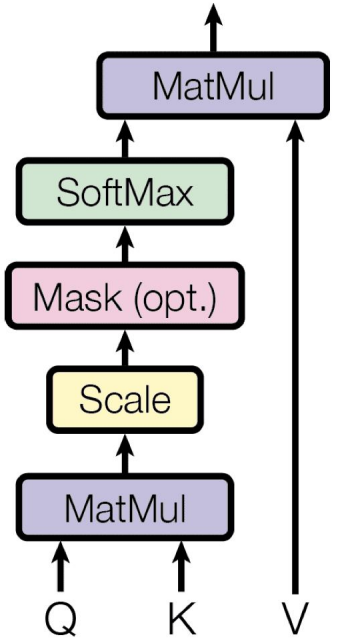
Q = query (one element in the sequence)

K = keys (all the elements in the sequence)

$$a = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)$$

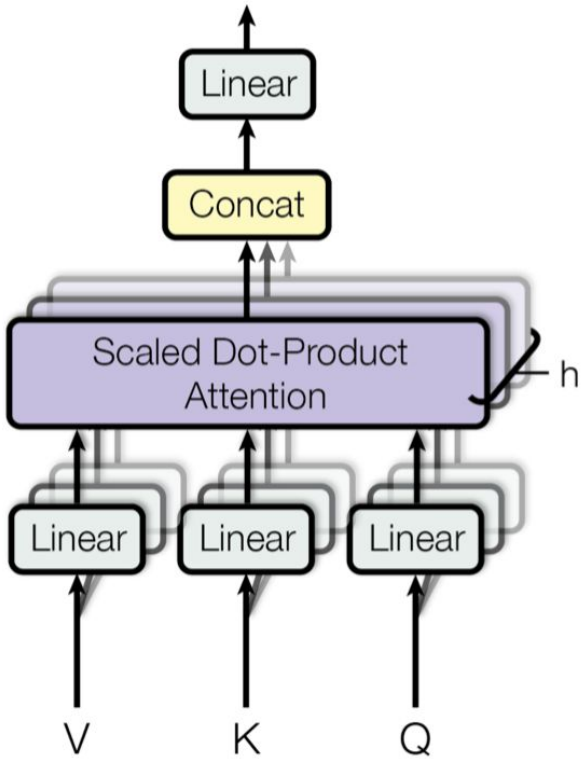
scale

V = values (all the elements in the sequence)



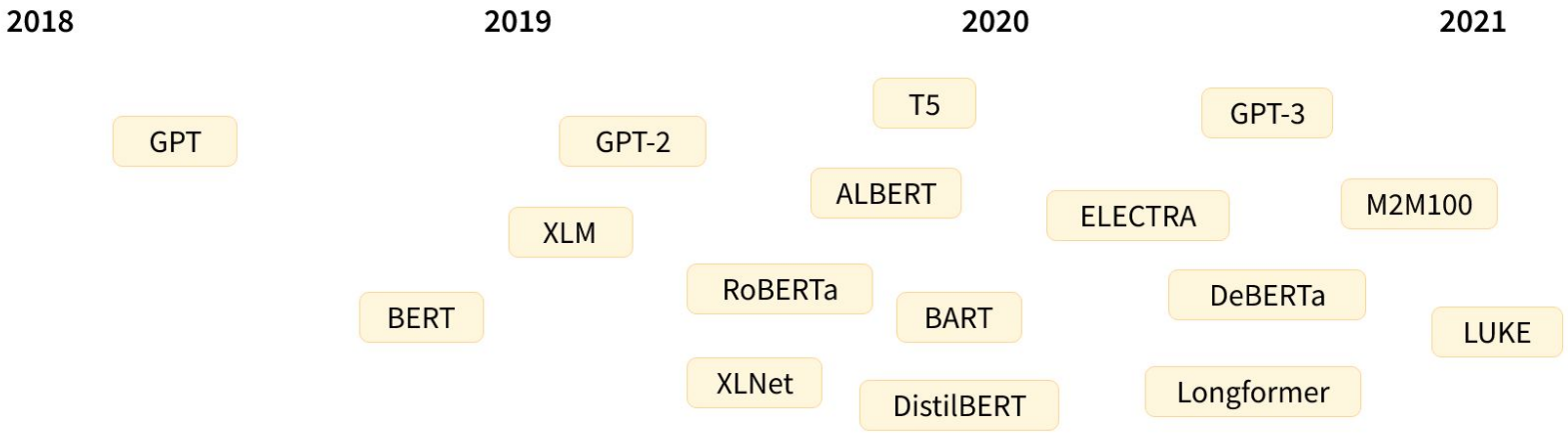
Multi-Head Attention

- Each layer has multiple *attention heads*
- Multiple attention heads can find different definition of **relevance**
- Multiple attentions heads encode relevance relations that are meaningful to humans



Popular models

(A bit of) Transformers history



https://huggingface.co

A recent example: DALL-E 2 (OpenAI, April 2022)

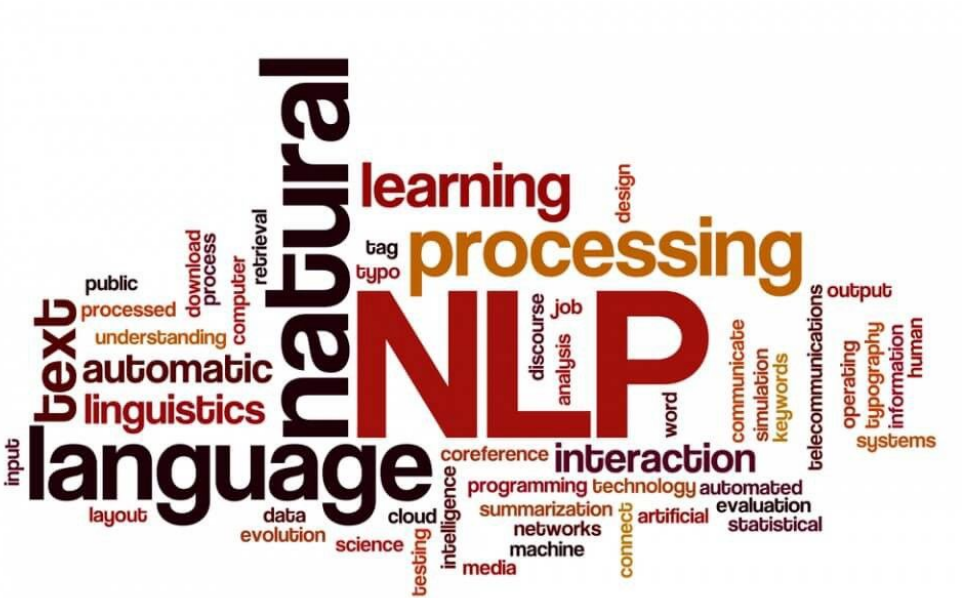
“An astronaut riding a horse in photorealistic style”

Create variations
inspired by the original

Applications

NLP

- Machine translation
- Text summarization
- Question-answering
- ...



Computer Vision

- Image Classification
- Object Detection
- Autonomous Driving
- Image Synthesis
- Video processing
- ...

Other applications...

- Speech recognition
- Time Series Forecasting
- Reinforcement learning

What’s next?