6. Attention-based networks

6.2. Transformers

Manel Martínez-Ramón
Meenu Ajith
Aswathy Rajendra Kurup

Self-attention models

Introduction

- A self-attention model is a mechanism where each token has its own query, keys and values.
- Each token attends to each other token based on their key vectors.
- This constructs a representation of the sequence of tokens which is based on a weighted sum over the rest of the tokens.
- This is also called an intra-attention model.

Self-attention models

Definitnion

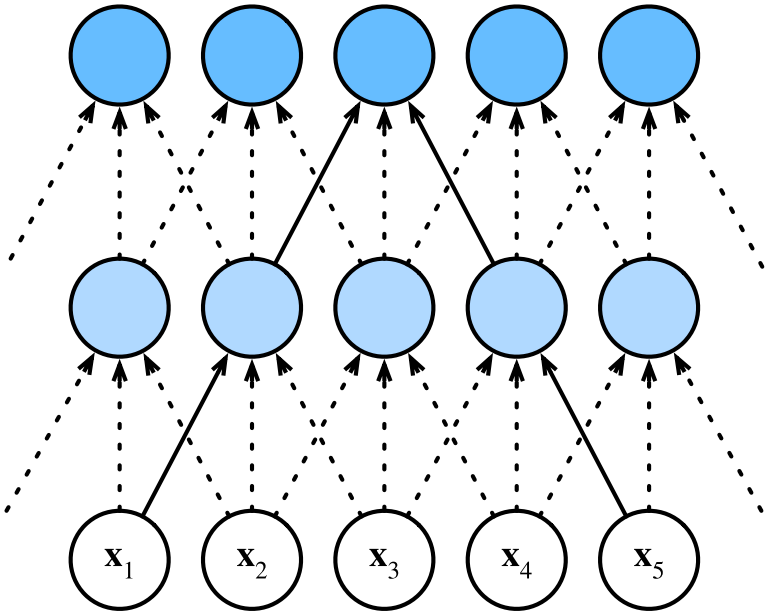
- Assume a sequence of input tokens $\mathbf{x}_1, \cdots \mathbf{x}_N$. A self-attention model outputs a sequence of N vectors $\mathbf{y}_1, \cdots \mathbf{y}_N$ of the same dimension.
- The equation of the model can be written as

$$\mathbf{y}_i = f(\mathbf{x}_i, (\mathbf{x}_1, \mathbf{x}_1), \ldots, (\mathbf{x}_n, \mathbf{x}_n)) \tag{1}$$

- This is a representation that is alternative to the RNN encoder-decoder seen in the previous lesson.

Self-attention models

Comparative complexities

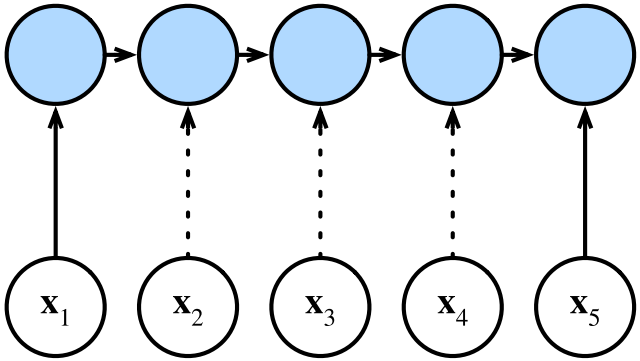


Convolutional neural network with kernels of size k.

- Computational complexity: $\mathcal{O}(kND^2)$.
- Sequential operations: $\mathcal{O}(1)$.
- Maximum path length between tokens $\mathcal{O}(N/k)$.

Self-attention models

Comparative complexities

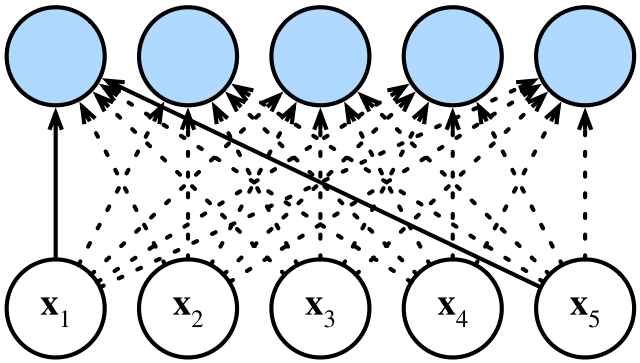


Recurrent neural networks.

- Computational complexity of hidden state computation: $\mathcal{O}(ND^2)$.
- Sequential operations: $\mathcal{O}(n)$.
- Maximum path length between tokens $\mathcal{O}(N)$.

Self-attention models

Comparative complexities



Self-attention models.

- Computational complexity: $\mathcal{O}(N^2 D)$.
- Sequential operations: $\mathcal{O}(1)$.
- Maximum path length between tokens $\mathcal{O}(1)$.

CNN and self-attention have parallel computing properties.
Self-attention has the shortest maximum path, but its N^2 is bad for long sequences

Positional encoding

- Self-attention avoids recurrent operation so parallel computing can be used.
- However, it does not pay *attention* to the order of the sequence.
- We need a mechanism to explain how the sequence arrived.
- This can be learned or fixed.
- The first positional encoding for transformers was the trigonometric positional encoding

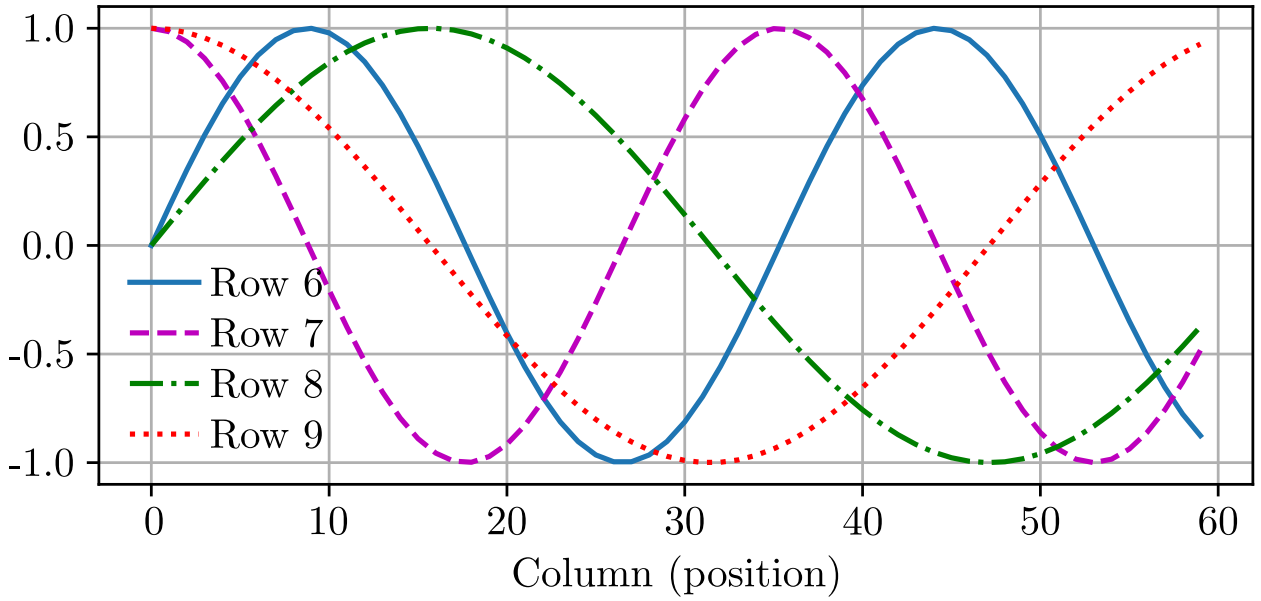
Positional encoding

- Assume that the embeddings $\mathbf{x}_1, \cdots, \mathbf{x}_N$ for the N tokens of a sequence are available.
- The positional encoding adds a vector $\mathbf{p}_n$ to each embedding with elements

$$\begin{aligned} p_{2j,n} &= \sin\left(\frac{n}{10000^{2j/D}}\right) \\ p_{2j+1,n} &= \cos\left(\frac{n}{10000^{2j/D}}\right) \end{aligned} \tag{2}$$

  where the first expression corresponds to elements with index even of vector $\mathbf{p}_n$ and the second to odd elements.
- The coded vector is $\mathbf{x}_n + \mathbf{p}_n$

Positional encoding



This can be compared to the binary coding

	Position															
A	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
B	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
C	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

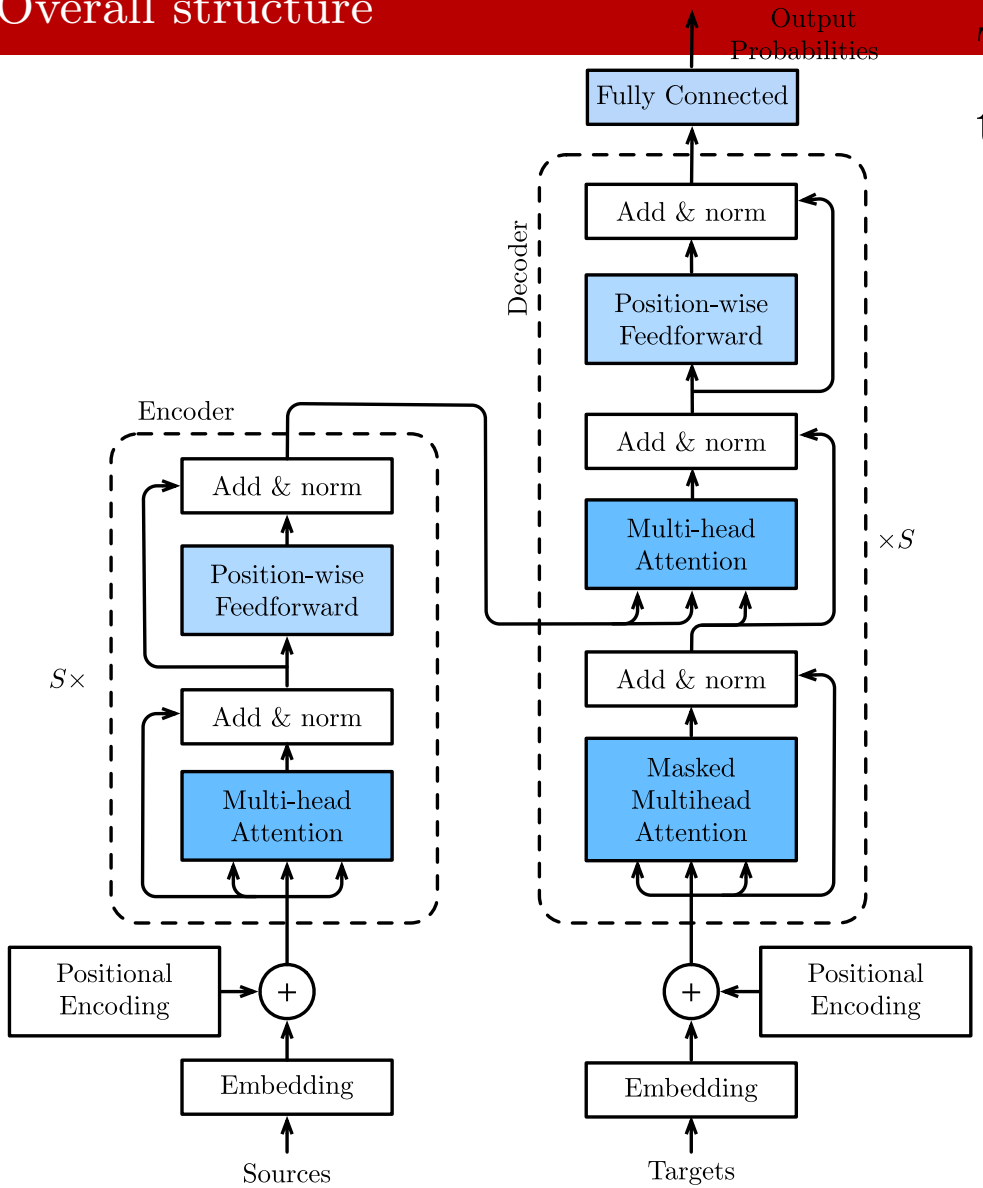
The transformer

Introduction

- Self-attention uses parallel computation and the shortest maximum path length.
- Deep structures for sequence encoding-decoding can be constructed with this strategy.
- The Transformer model is based only in self-attention mechanisms without CNN or RNN.
- Transformers are used in language, vision, speech or reinforcement learning.

The transformer

Overall structure

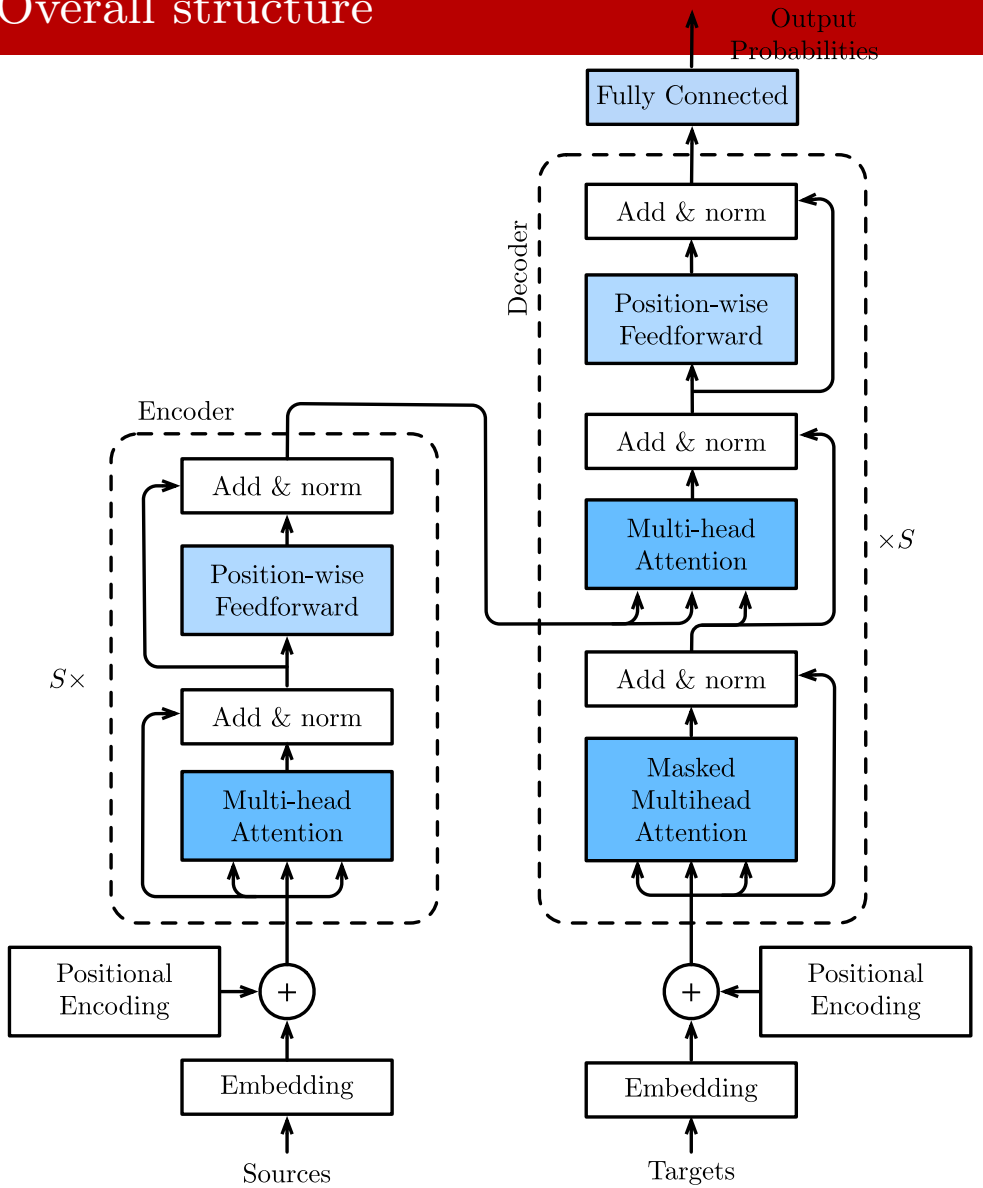


This is the overall structure of a transformer.

- The input and output sequences are added to a positional encoding.
- The encoder is a stack of identical layers.
- Each input is the output of the previous layer.
- The add & norm (residual) block adds the input and the output of the previous one, which has the same size.

The transformer

Overall structure



- The decoder is also a stack of multiple identical layers with residual and normalizations.
- The central sublayer inserts the encoder input as keys and values in a block called the encoder-decoder attention.
- The masked attention allows each position to attend to all positions until that position. That prevents a position to attend subsequent positions.
- Originally, S=6.

The transformer

Basic operation

- The transformer encoder maps a sequence $\mathbf{x}_1, \cdots, \mathbf{x}_N$ into a sequence of continuous representations $\mathbf{z}_1, \cdots, \mathbf{z}_N$.
- The decoder uses the sequence $\mathbf{z}_n$ to generate an output sequence $\mathbf{y}_1, \cdots, \mathbf{y}_M$ one element at a time.
- A position-wise feedforward network is a fully connected layer with ReLU activation followed by a linear one, expressed as

$$FF(\mathbf{u}) = \mathbf{W}^{(2)\top} \varphi \left(\mathbf{W}^{(1)\top} \mathbf{u} + \mathbf{b}_1 \right) + \mathbf{b}_2 \tag{3}$$

where $\mathbf{u}$ represents the FF input, $\mathbf{W}^{(2)} \in \mathbb{R}^{2048 \times 512}$ and $\mathbf{W}^{(1)} \in \mathbb{R}^{512 \times 2048}$

At each of the S layers, it is applied at each position separately and identically (with different parameters at each layer).

- The self-attention reduces the complexity per layer and increases parallelization.

Layer type	Complexity per layer	Sequential ops	Max. path length
Self-attention	$\mathcal{O}(N^2D)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Recurrent	$\mathcal{O}(ND^2)$	$\mathcal{O}(N)$	$\mathcal{O}(N)$
Convolutional	$\mathcal{O}(kND^2)$	$\mathcal{O}(1)$	$\mathcal{O}(\log_k(N))$
Self-attention (restricted)	$\mathcal{O}(rND)$	$\mathcal{O}(1)$	$\mathcal{O}(N/r)$

- The short path length allows the transformer to learn long-term dependencies.

- Training data:
  - WMT 2014 English-German dataset: 4.5M sentence pairs encoded with byte-pair encoding. WMT2014 English-French dataset: 36M sentences and split tokens into a 32000 word-piece vocabulary. Training batches with approximately 25000 source target tokens.
  - Base models trained with 100.000 training steps (12 hours), big models, 300.000 steps (3.5 days) with 8 GPUs.
  - Maximum BLEU of 41.0