CS551: Introduction to Deep Learning

Transformer

Arijit Mondal
Dept. of Computer Science & Engineering
Indian Institute of Technology Patna
arijit@iitp.ac.in

Attention

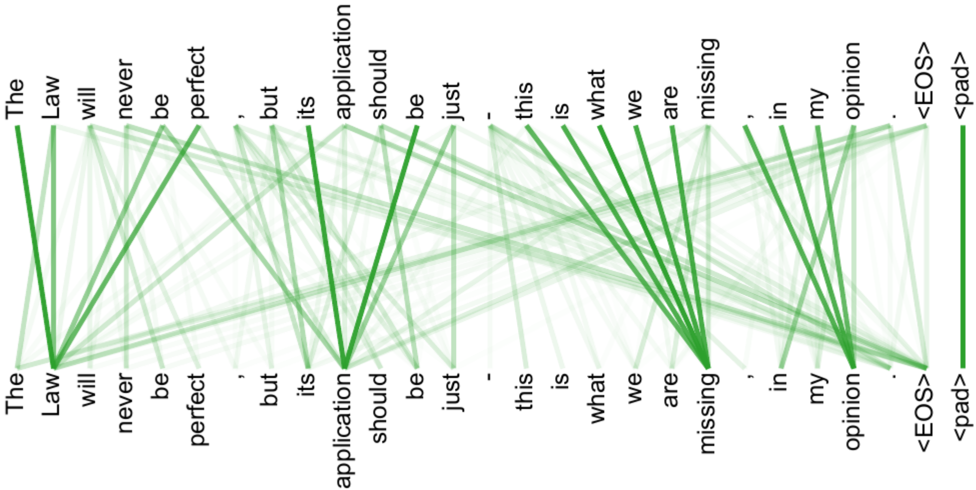
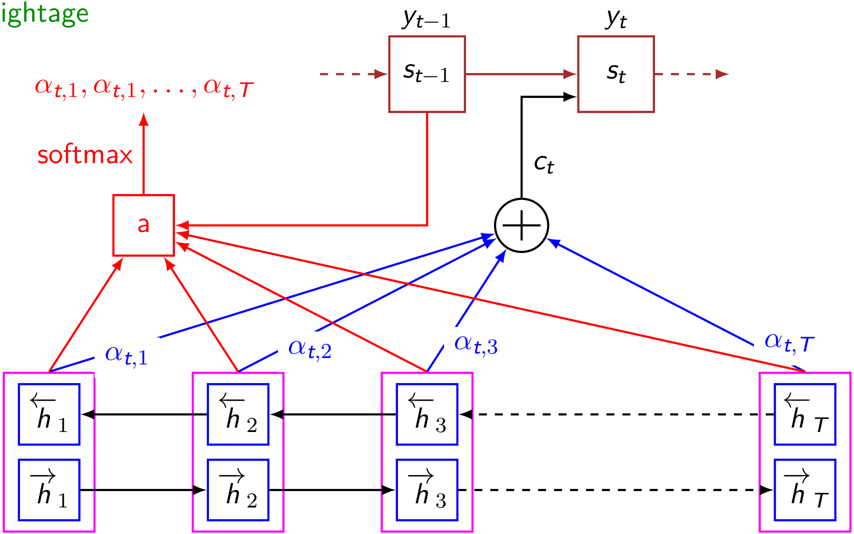


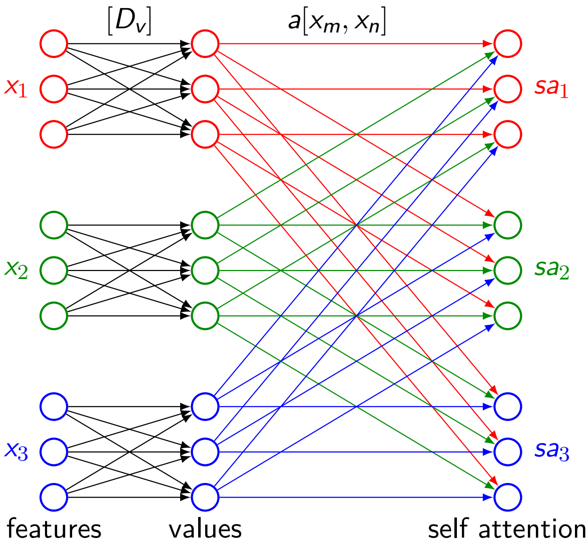
image source: Attention is all you need

Attention with RNN

- $\alpha_t = NN(s_{t-1}, h_t)$
- Softmax is used for weightage
- Context $= \sum_t \alpha_t h_t$



Self Attention

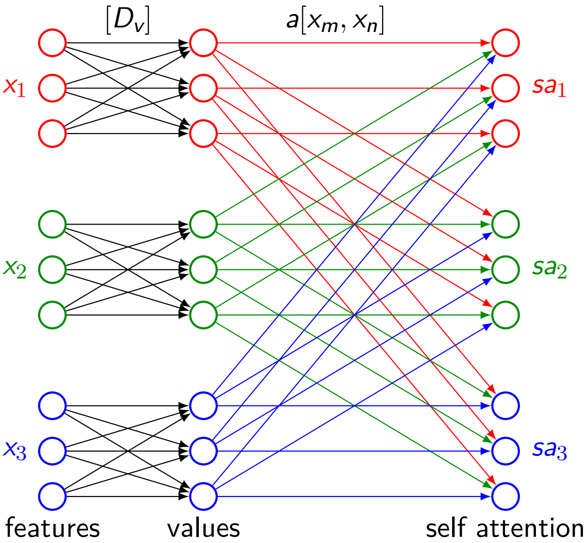


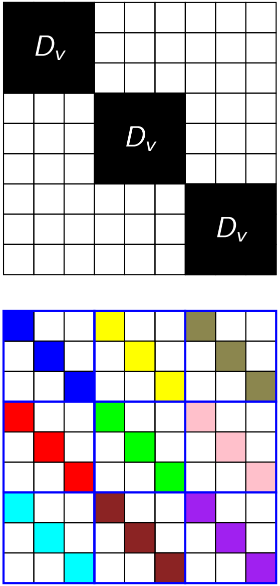
$$v_m = ReLU(b_v + D_v \times x_m)$$

$$sa_n[x_1, \ldots, x_N] = \sum_{m=1}^{N} a[x_m, x_n] \times v_m$$

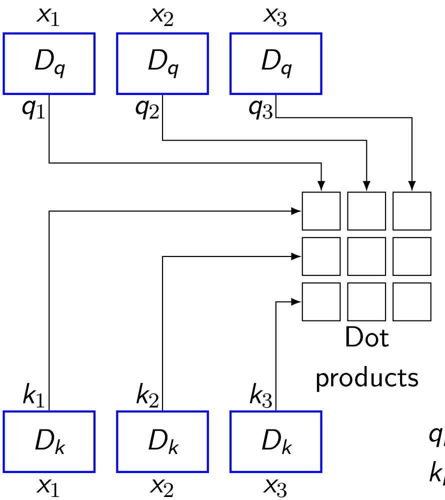
$$a[\bullet, x_n] \geq 0, \quad \sum_{m=1}^{N} a[x_m, x_n] = 1$$

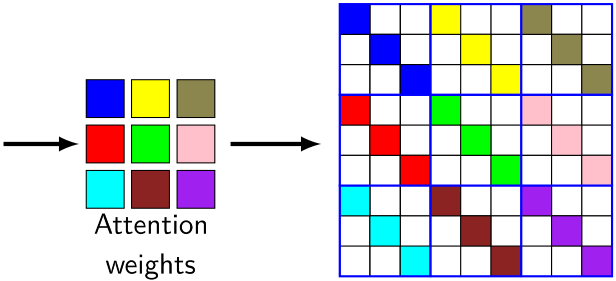
Self attention





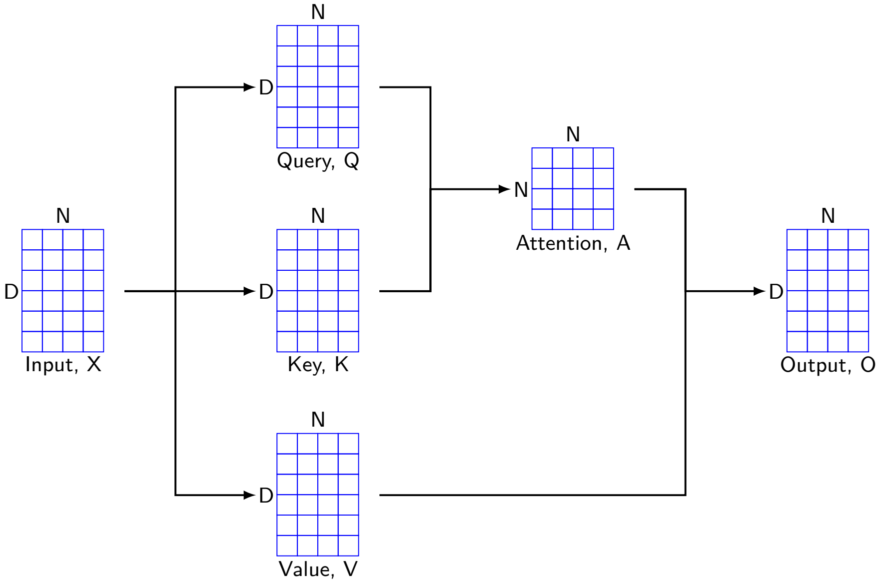
Computing attention weights



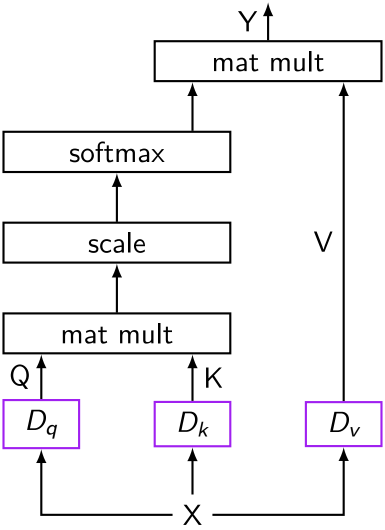


$$q_n = ReLU(b_q + D_q \times x_n)$$
$$k_m = ReLU(b_k + D_k \times x_m)$$
$$a[x_m, x_n] = \text{softmax}_m[k_\bullet^T q_n] = \frac{\exp[k_m^T q_n]}{\sum_{i=1}^N \exp[k_i^T q_n]}$$

Self-attention: Complete view



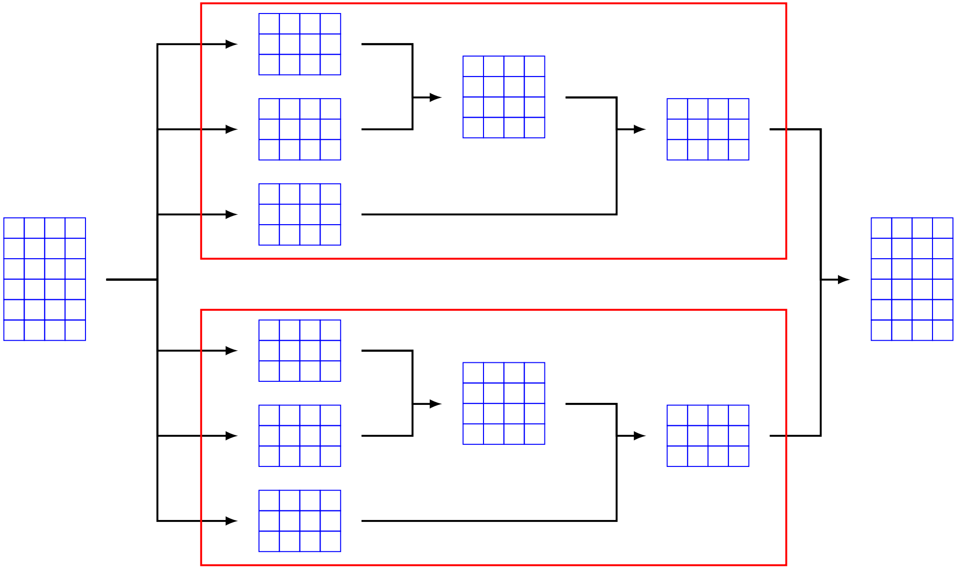
Self attention: Block diagram



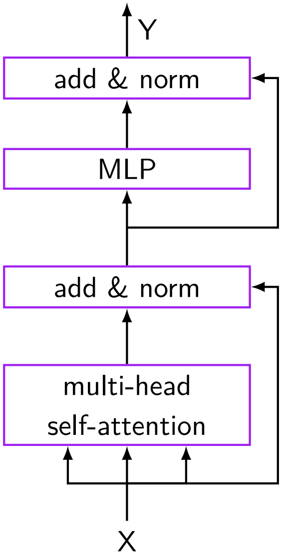
scale: $\left[\frac{K^T Q}{\sqrt{D}}\right]$

$$Y = Sa[X] = V \cdot \text{softmax}\left[\frac{K^T Q}{\sqrt{D}}\right]$$

Multi-head self attention



Transformer layer



Positional embedding

- Input $X \in \mathbb{R}^{N \times D}$ contains the D-dimensional embeddings for N tokens of a sequence
- The positional encoding outputs $X + P$ using a positional embedding matrix $P \in \mathbb{R}^{N \times D}$ of the same shape, whose element on the ith row and the $(2j)$th or the $(2j+1)$th column is

$$p_{i,2j} = \sin\left(\frac{i}{10000^{2j/D}}\right), \quad p_{i,2j+1} = \cos\left(\frac{i}{10000^{2j/D}}\right)$$

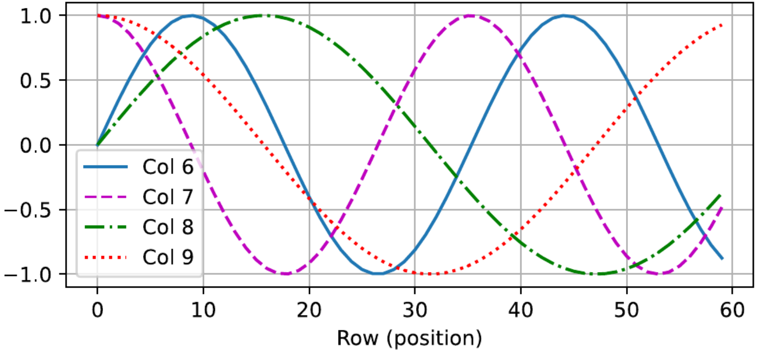


image source: Dive into Deep Learning

Transformer: Encoder

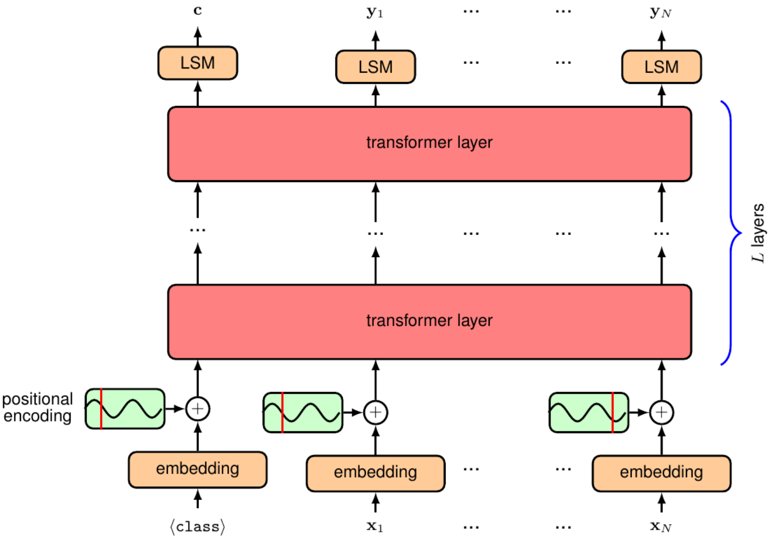


image source: Deep Learning Foundations and Concepts

Transformer: Decoder

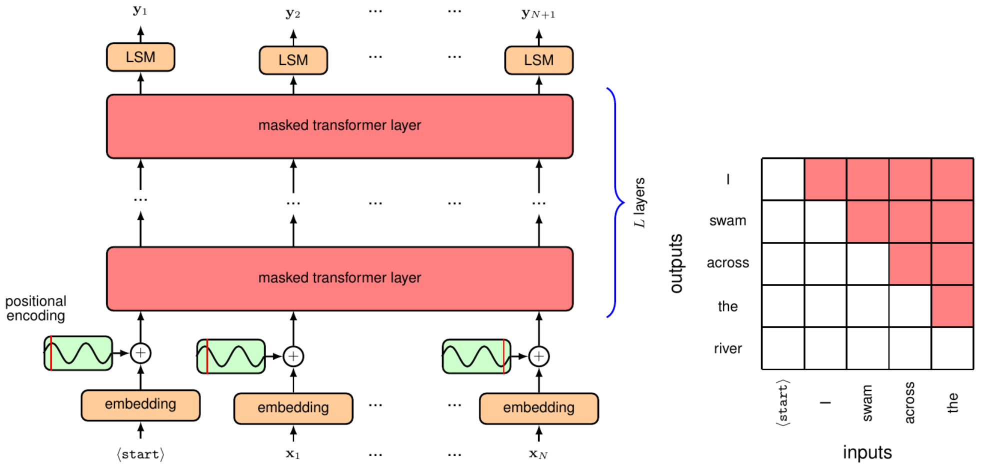


image source: Deep Learning Foundations and Concepts

Transformer: Cross attention

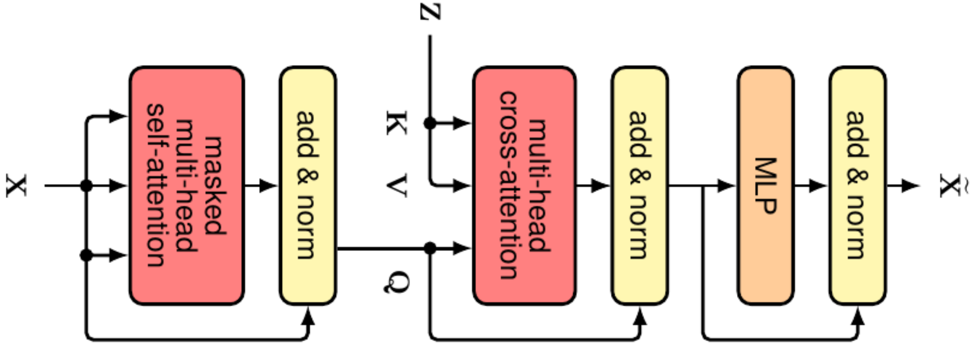


image source: Deep Learning Foundations and Concepts

Transformer: Encoder-Decoder

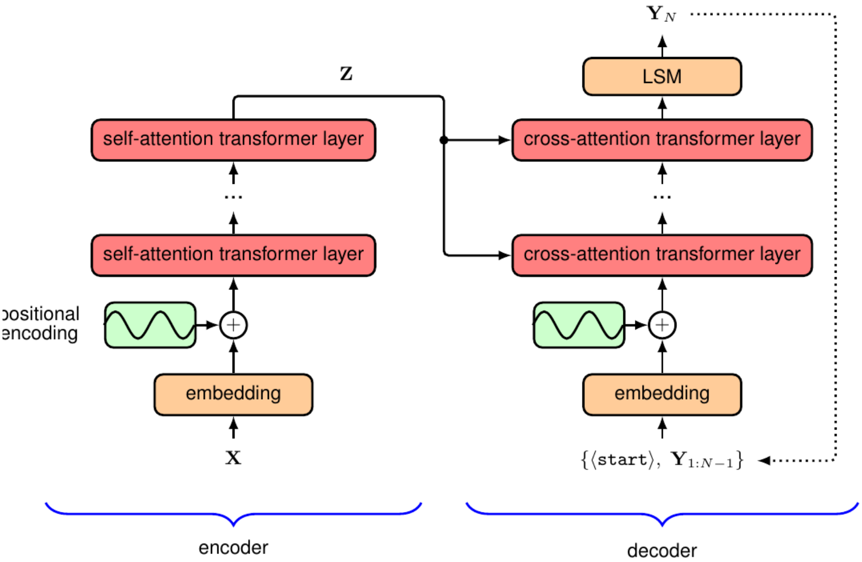


image source: Deep Learning Foundations and Concepts

Transformer: Encoder-Decoder

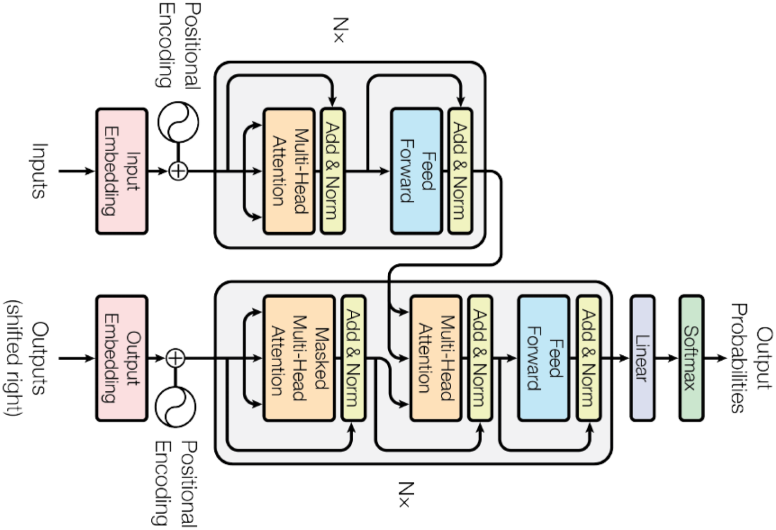


image source: Attention is all you need

Vision Transformer

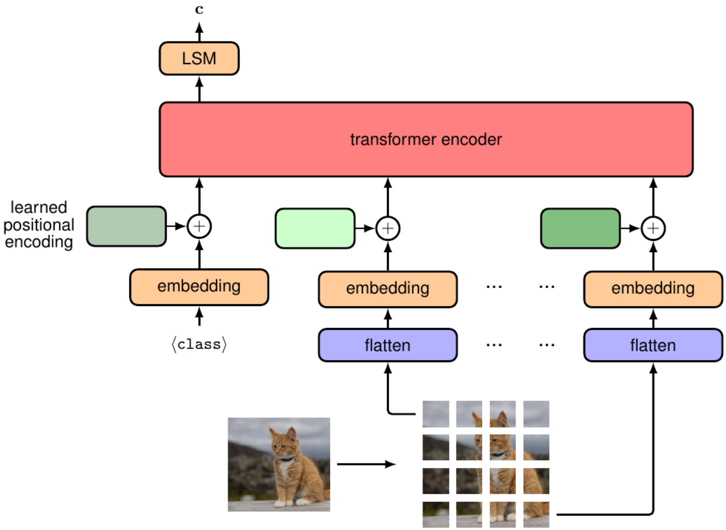


image source: Deep Learning Foundations and Concepts

Summary

- We have seen self-attention, multi-head attention and transformer layer
- Positional embedding is crucial
- Transformer has low complexity per layer
- Many computations can be parallelized
- It can handle long range dependencies in the text
- Performs very well for text data