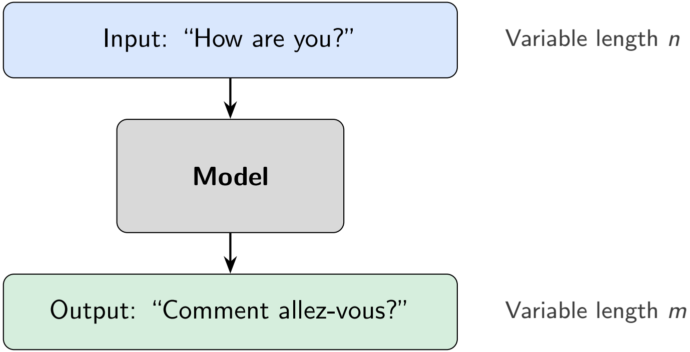
Sequence-to-Sequence Models & Transformers

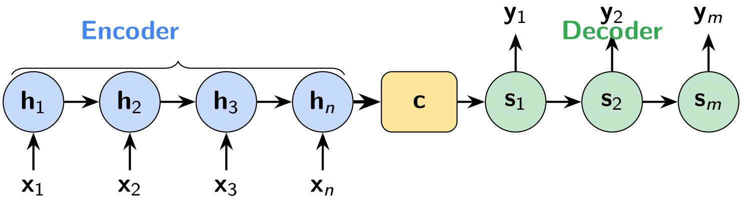
COS 484 Precept

The Seq2Seq Problem



Challenge: Input and output have **different lengths**

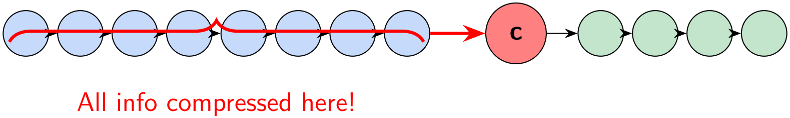
Encoder-Decoder Architecture



Encoder: $\mathbf{h}_t = f(\mathbf{h}_{t-1}, \mathbf{x}_t)$

Decoder: $\mathbf{s}_t = g(\mathbf{s}_{t-1}, \mathbf{y}_{t-1}, \mathbf{c})$

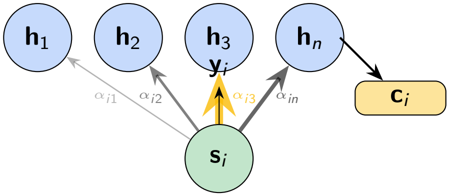
The Bottleneck Problem



Problem

- Fixed-size vector must encode **entire** input sequence
- Early information gets “forgotten”
- Performance degrades for long sequences

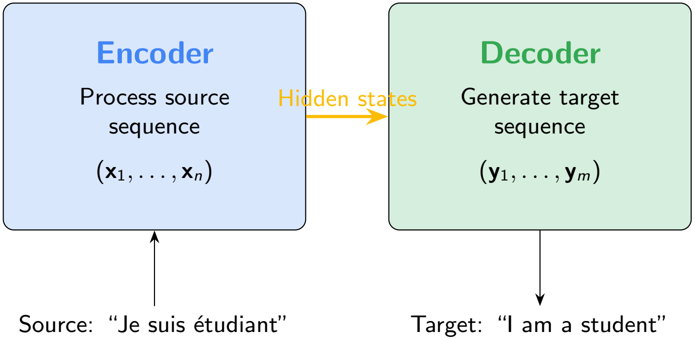
Attention: The Key Idea



Intuition: Each decoder step "looks back" at encoder states

Dynamic context $\rightarrow$ Different focus for each output

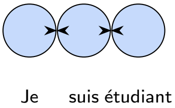
The Encoder-Decoder Paradigm



What Does Each Component Do?

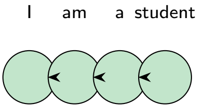
Encoder

- Reads the **entire** source sequence
- Creates **contextualized representations**
- Output: Hidden states $\mathbf{h}_1, \ldots, \mathbf{h}_n \in \mathbb{R}^{n \times d}$
- Bidirectional: can see past & future

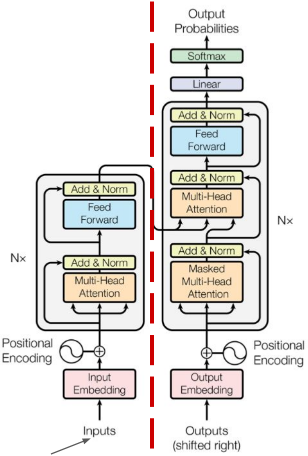


Decoder

- Generates output **one token at a time**
- Conditions on encoder output
- Output: $P(\mathbf{y}_t | \mathbf{y}_{<t}, \mathbf{x})$
- Autoregressive: only sees past



Transformer Architecture



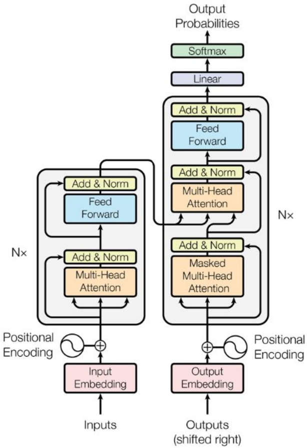
Encoder

Decoder

Source Sequence Input

Target Sequence (i.e. current progress generating words)

Transformer Encoder

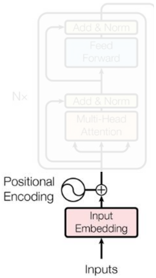


Source sequence $(x_1, \ldots, x_n)$

Transformer Encoder: Positional + Word Embedding

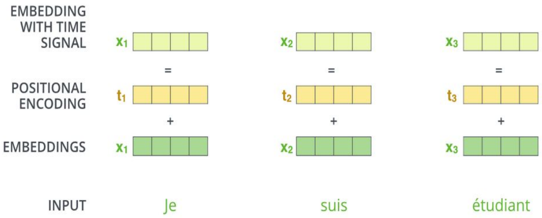
Embedded source sequence

$\mathbb{R}^{n \times d_1}$



Source sequence $(x_1, \ldots, x_n)$

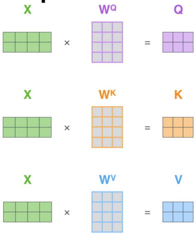
Input and Positional Embedding



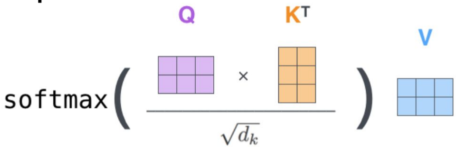
Transformer Encoder: Multi-Head Self Attention

Self-Attention: $W_i^Q \in \mathbb{R}^{d_1 \times d_q}, W_i^K \in \mathbb{R}^{d_1 \times d_k}, W_i^V \in \mathbb{R}^{d_1 \times d_v}$

Step 1:

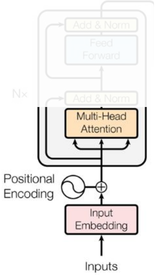


Step 2:



After Multi-Head Attention
$\mathbb{R}^{n \times d_2}$

Embedded source sequence
$\mathbb{R}^{n \times d_1}$



Source sequence
$(x_1, \ldots, x_n)$

MultiHead Attention: $W^O \in \mathbb{R}^{d \times d_2}$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \ldots, \mathrm{head_h})W^O$$

$$\mathrm{head}_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

Transformer Encoder: Multi-Head Self Attention

Self-Attention: $W_i^Q \in \mathbb{R}^{d_1 \times d_q}, W_i^K \in \mathbb{R}^{d_1 \times d_k}, W_i^V \in \mathbb{R}^{d_1 \times d_v}$

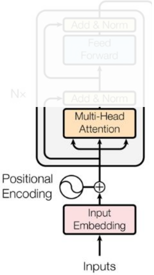
In practice, $d_1 = d_2$

After Multi-Head Attention

$\mathbb{R}^{n \times d_1}$

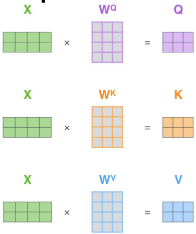
Embedded source sequence

$\mathbb{R}^{n \times d_1}$

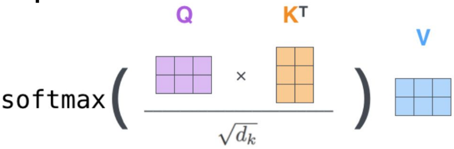


Source sequence $(x_1, \dots, x_n)$

Step 1:



Step 2:



"Self" attention means Q, K, V are all computed from a single sequence

MultiHead Attention: $W^O \in \mathbb{R}^{d \times d_2}$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head_1}, \dots, \mathrm{head_h})W^O$$

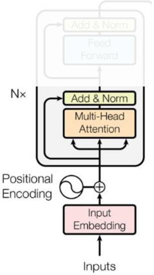
$$\mathrm{head}_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

Transformer Encoder: Add & Norm

After Add & Norm
$\mathbb{R}^{n\times d_1}$

After Multi-Head Attention
$\mathbb{R}^{n\times d_1}$

Embedded source sequence
$\mathbb{R}^{n\times d_1}$



Source sequence $(x_1, \ldots, x_n)$

Add & Norm:

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

LayerNorm

$$\boxed{y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta}$$

Transformer Encoder: Feed Forward

After Feed Forward

$\mathbb{R}^{n \times d_1}$

After Add & Norm

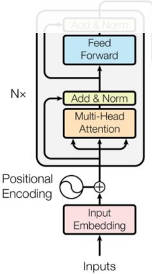
$\mathbb{R}^{n \times d_1}$

After Multi-Head Attention

$\mathbb{R}^{n \times d_1}$

Embedded source sequence

$\mathbb{R}^{n \times d_1}$



Source sequence $(x_1, \ldots, x_n)$

Feed Forward

$$\mathrm{FFN}(\mathbf{x}_i) = \mathrm{ReLU}(\mathbf{x}_i \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W_2} + \mathbf{b}_2$$

$$\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}, \mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$$

$$\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}, \mathbf{b}_2 \in \mathbb{R}^{d}$$

Compute transformation over each value in the sequence **independently**

Transformer Encoder: Final Add & Norm

After Final Add & Norm

$\mathbb{R}^{n\times d_1}$

After Feed Forward

$\mathbb{R}^{n\times d_1}$

After Add & Norm

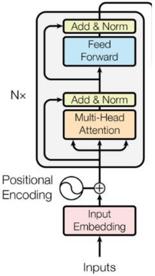
$\mathbb{R}^{n\times d_1}$

After Multi-Head Attention

$\mathbb{R}^{n\times d_1}$

Embedded source sequence

$\mathbb{R}^{n\times d_1}$



Source sequence $(x_1, \ldots, x_n)$

Add & Norm:

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

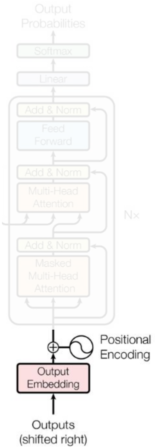
LayerNorm

$$\boxed{y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta}$$

Transformer Decoder:

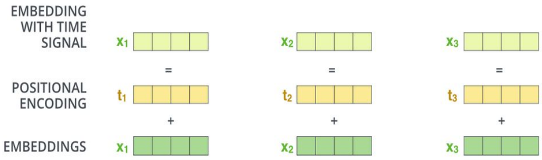
Embedded target sequence

$\mathbb{R}^{m \times d_1}$



Target sequence
(<bos>, x_1, …, x_m)

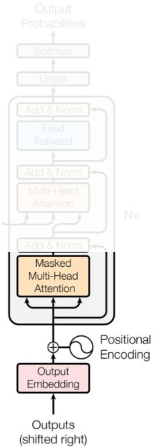
Output and Positional Embedding



Transformer Decoder: Masked Multi-Head Attention

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

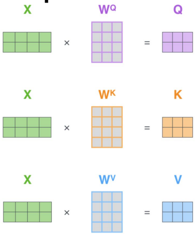
Embedded target sequence
$\mathbb{R}^{m \times d_1}$



Target sequence
(<bos>, x_1, …, x_m)

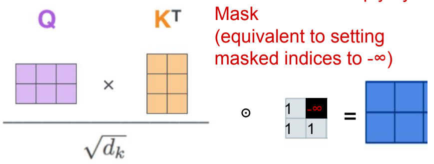
Masked Self-Attention: $W_i^Q \in \mathbb{R}^{d_1 \times d_q}, W_i^K \in \mathbb{R}^{d_1 \times d_k}, W_i^V \in \mathbb{R}^{d_1 \times d_v}$

Step 1:

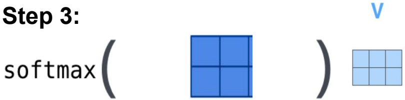


Step 2:

Elementwise Multiply by Mask
(equivalent to setting masked indices to -∞)



Step 3:



MultiHead Attention: $W^O \in \mathbb{R}^{d \times d_2}$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head}_h)W^O$$

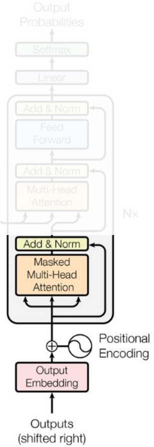
$$\mathrm{head}_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

Transformer Decoder:

After Add & Norm
$\mathbb{R}^{m \times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

Embedded target sequence
$\mathbb{R}^{m \times d_1}$



Target sequence
(<bos>, x_1, …, x_m)

Add & Norm:

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

LayerNorm

$$\boxed{y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta}$$

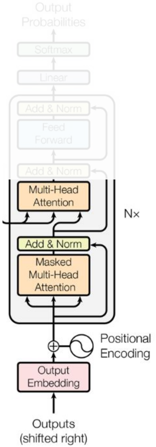
Transformer Decoder: Multi-Head (Cross) Attention

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

After Add & Norm
$\mathbb{R}^{m \times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

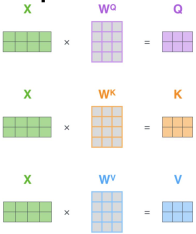
Embedded target sequence
$\mathbb{R}^{m \times d_1}$



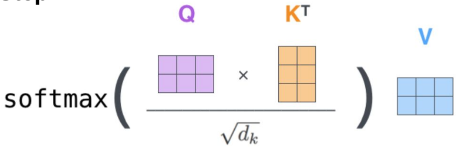
Target sequence
(<bos>, x_1, …, x_m)

Cross-Attention: $W_i^Q \in \mathbb{R}^{d_1 \times d_q}, W_i^K \in \mathbb{R}^{d_1 \times d_k}, W_i^V \in \mathbb{R}^{d_1 \times d_v}$

Step 1:



Step 2:



"Cross" attention means Q, K, V are computed from **separate** sequences

MultiHead Attention: $W^O \in \mathbb{R}^{d \times d_2}$

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head_h})W^O$$

$$\mathrm{head}_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

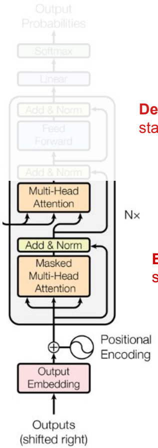
Transformer Decoder: Multi-Head (Cross) Attention

Masked Multi-Head Attention
$\mathbb{R}^{m\times d_1}$

After Add & Norm
$\mathbb{R}^{m\times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m\times d_1}$

Embedded target sequence
$\mathbb{R}^{m\times d_1}$

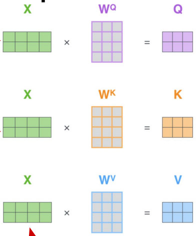


Target sequence
(<bos>, x_1, …, x_m)

Cross-Attention: $W_i^Q \in \mathbb{R}^{d_1\times d_q}, W_i^K \in \mathbb{R}^{d_1\times d_k}, W_i^V \in \mathbb{R}^{d_1\times d_v}$

Step 1:

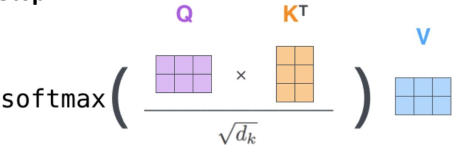
Decoder state



Encoder state

Encoder state

Step 2:



"Cross" attention means Q, K, V are computed from **separate** sequences

MultiHead Attention: $W^O \in \mathbb{R}^{d\times d_2}$

$$\mathrm{MultiHead}(Q,K,V) = \mathrm{Concat}(\mathrm{head}_1, ..., \mathrm{head_h})W^O$$

$$\mathrm{head}_i = \mathrm{Attention}(XW_i^Q, XW_i^K, XW_i^V)$$

Transformer Decoder: Add & Norm

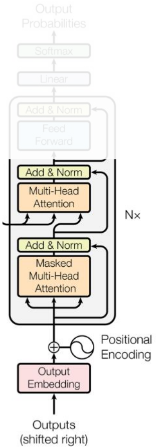
Add & Norm
$\mathbb{R}^{m \times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

After Add & Norm
$\mathbb{R}^{m \times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m \times d_1}$

Embedded target sequence
$\mathbb{R}^{m \times d_1}$



Target sequence
(<bos>, x_1, …, x_m)

Add & Norm:

$$\text{LayerNorm}(x + \text{Sublayer}(x))$$

LayerNorm

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

Transformer Decoder: Feed Forward

Feed Forward
$\mathbb{R}^{m\times d_1}$

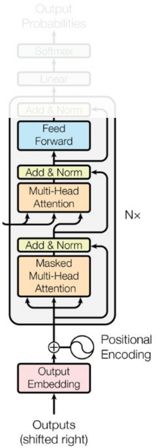
Add & Norm
$\mathbb{R}^{m\times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m\times d_1}$

After Add & Norm
$\mathbb{R}^{m\times d_1}$

Masked Multi-Head Attention
$\mathbb{R}^{m\times d_1}$

Embedded target sequence
$\mathbb{R}^{m\times d_1}$



Target sequence
(<bos>, x_1, …, x_m)

Feed Forward

$$\mathrm{FFN}(\mathbf{x}_i) = \mathrm{ReLU}(\mathbf{x}_i \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W_2} + \mathbf{b}_2$$

$$\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}, \mathbf{b}_1 \in \mathbb{R}^{d_{ff}}$$

$$\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}, \mathbf{b}_2 \in \mathbb{R}^{d}$$

Transformer Decoder: Add & Norm

Add & Norm

$\mathbb{R}^{m\times d_1}$

Feed Forward

$\mathbb{R}^{m\times d_1}$

Add & Norm

$\mathbb{R}^{m\times d_1}$

Masked Multi-Head Attention

$\mathbb{R}^{m\times d_1}$

After Add & Norm

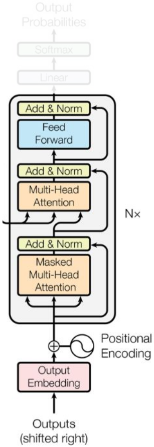
$\mathbb{R}^{m\times d_1}$

Masked Multi-Head Attention

$\mathbb{R}^{m\times d_1}$

Embedded target sequence

$\mathbb{R}^{m\times d_1}$



Target sequence

(<bos>, x_1, …, x_m)

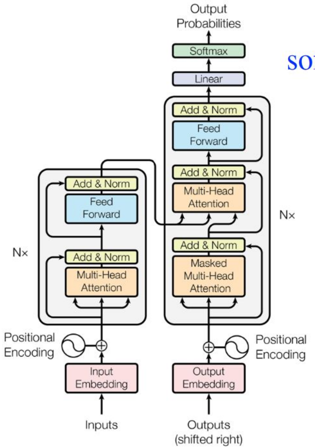
Add & Norm:

$$\mathrm{LayerNorm}(x + \mathrm{Sublayer}(x))$$

LayerNorm

$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

Transformer: Final output



$\mathrm{softmax}(\mathbf{W}_o\mathbf{h}_i)$

Compute transformation over **concatenated states**