

Machine Learning I

MATH80629A

Apprentissage Automatique I

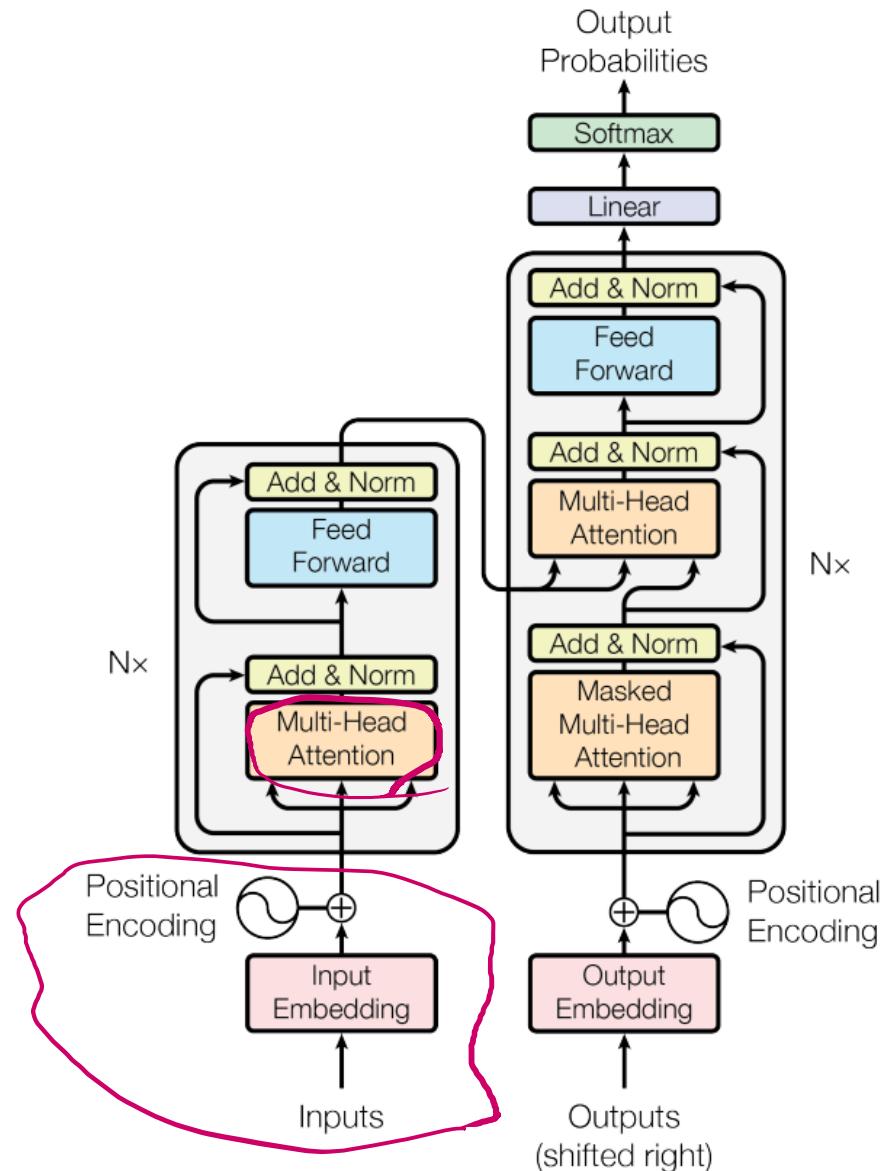
MATH80629

Attention and Transformers
— Week #6

Transformers

The architecture

Transformers

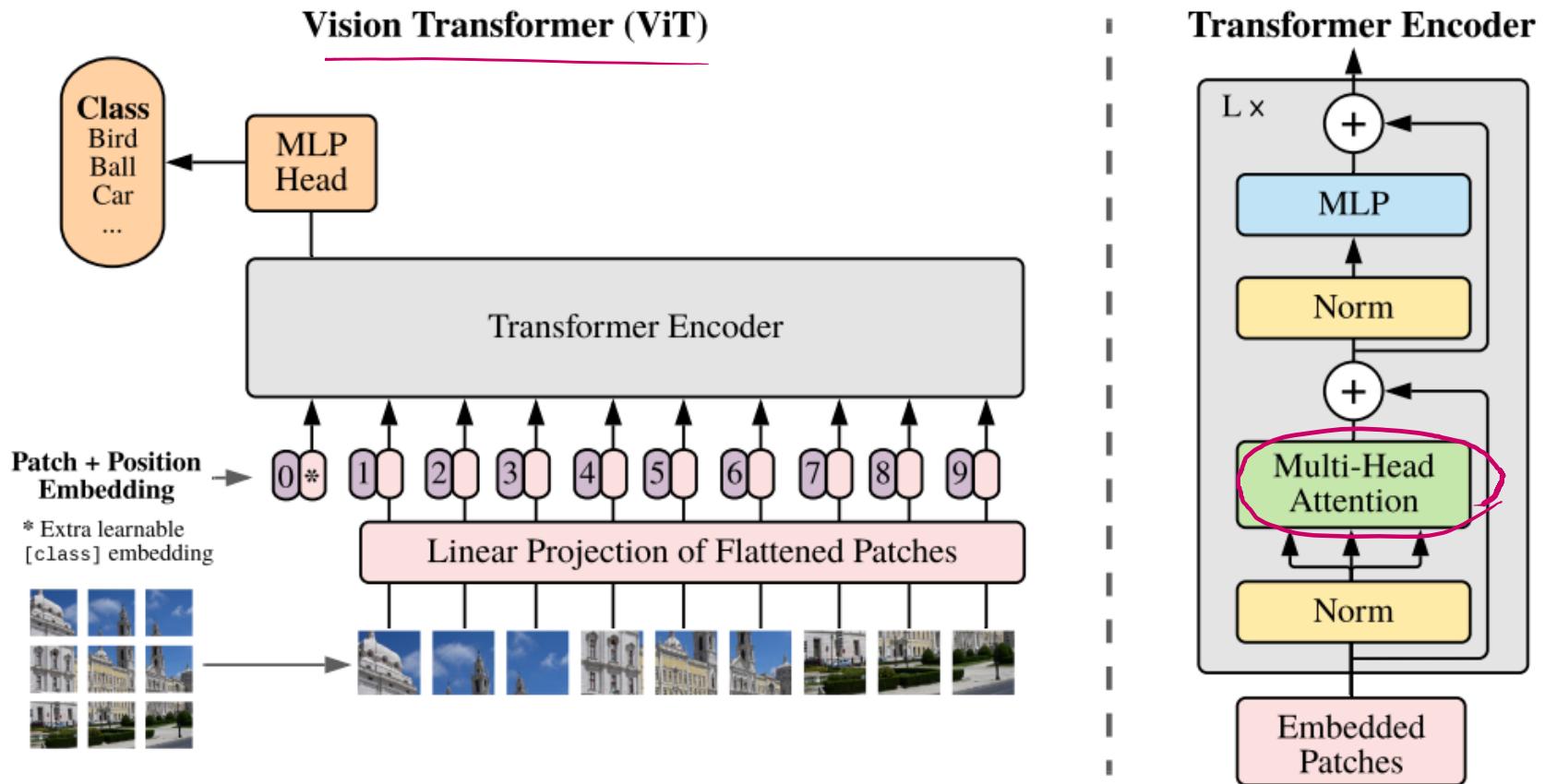


Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Transformers

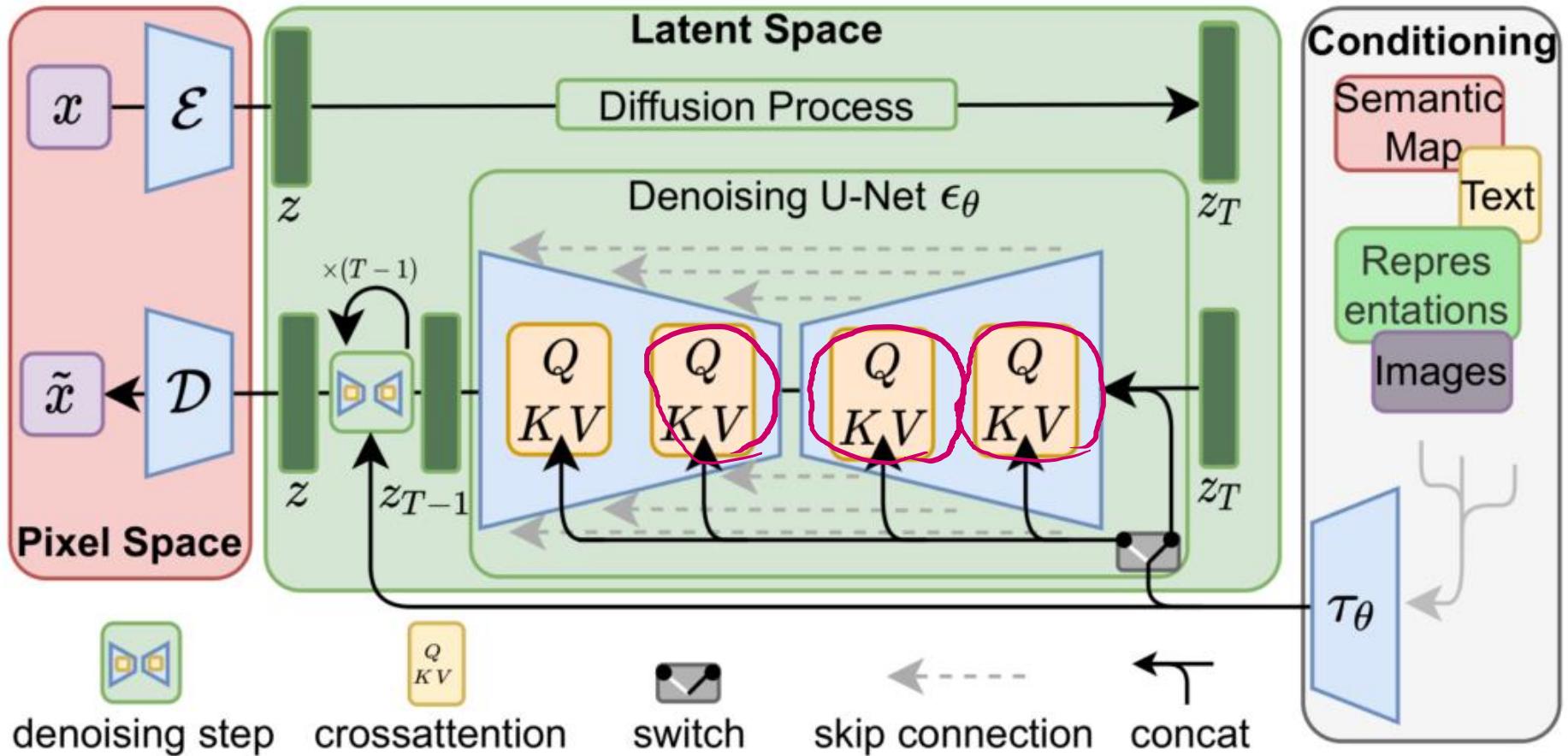
Applications

Vision Transformers



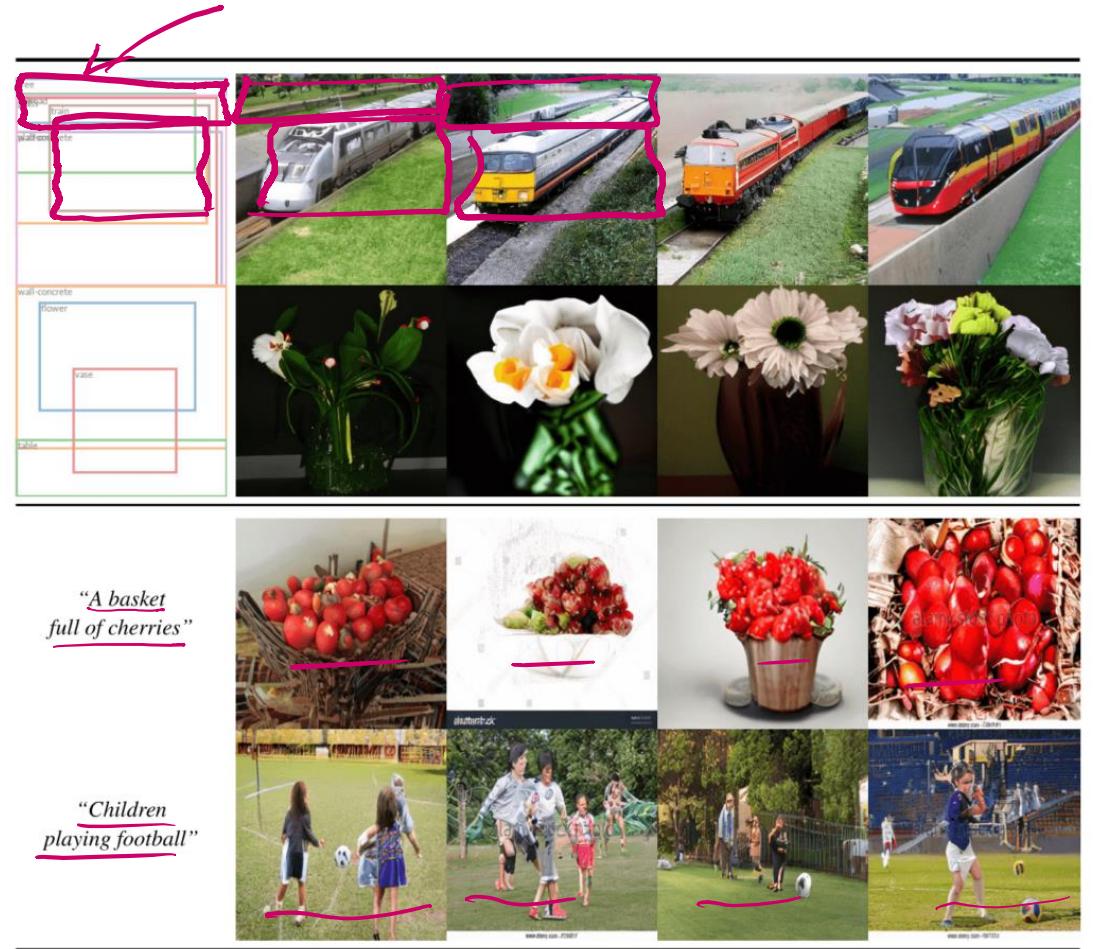
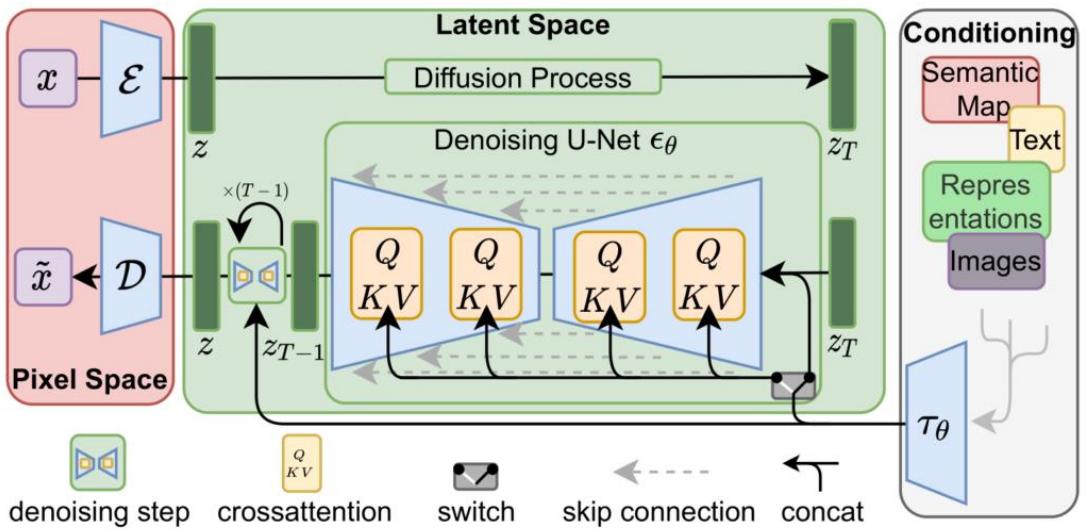
Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

Stable diffusion



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Stable diffusion



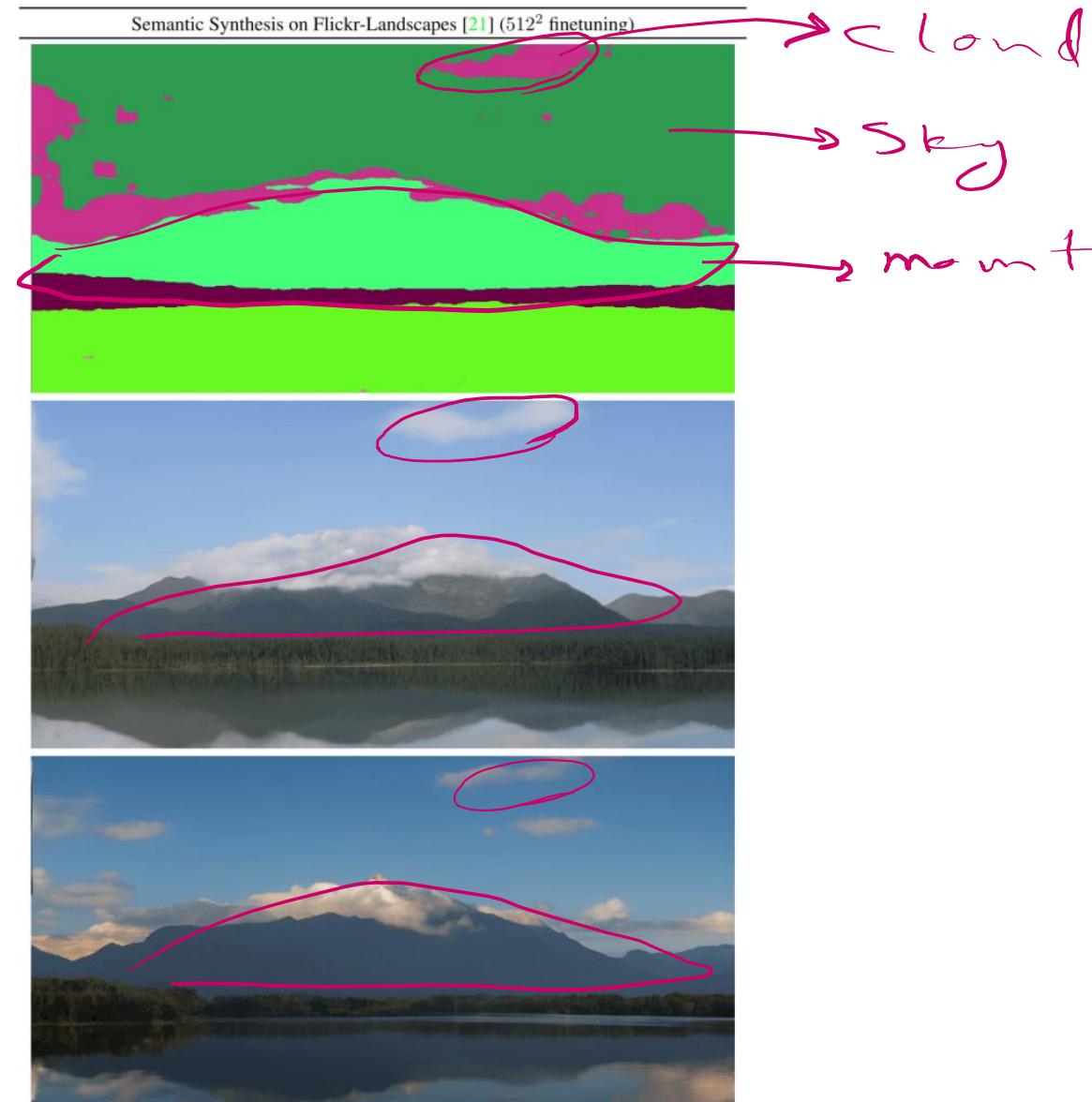
Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Stable diffusion



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Stable diffusion



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Stable diffusion

in-painting



Rombach, Robin, et al. "High-resolution image synthesis with latent diffusion models." *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022.

Speech



Bark is a universal text-to-audio model created by [Suno](www.suno.ai), with code publicly available [[here](#)](https://github.com/suno-ai/bark). Bark can generate highly realistic, multilingual speech as well as other audio - including music, background noise and simple sound effects. This demo should be used for research purposes only. Commercial use is strictly prohibited. The model output is not censored and the authors do not endorse the opinions in the generated content. Use at your own risk.

<https://github.com/suno-ai/bark>

Input Text

你好, I am a Transformer Model, uh – Excusez-moi
mes amies. My name is Bark.

Acoustic Prompt

Unconditional

✓ Unconditional

Announcer

Speaker 0 (en)

Speaker 1 (en)

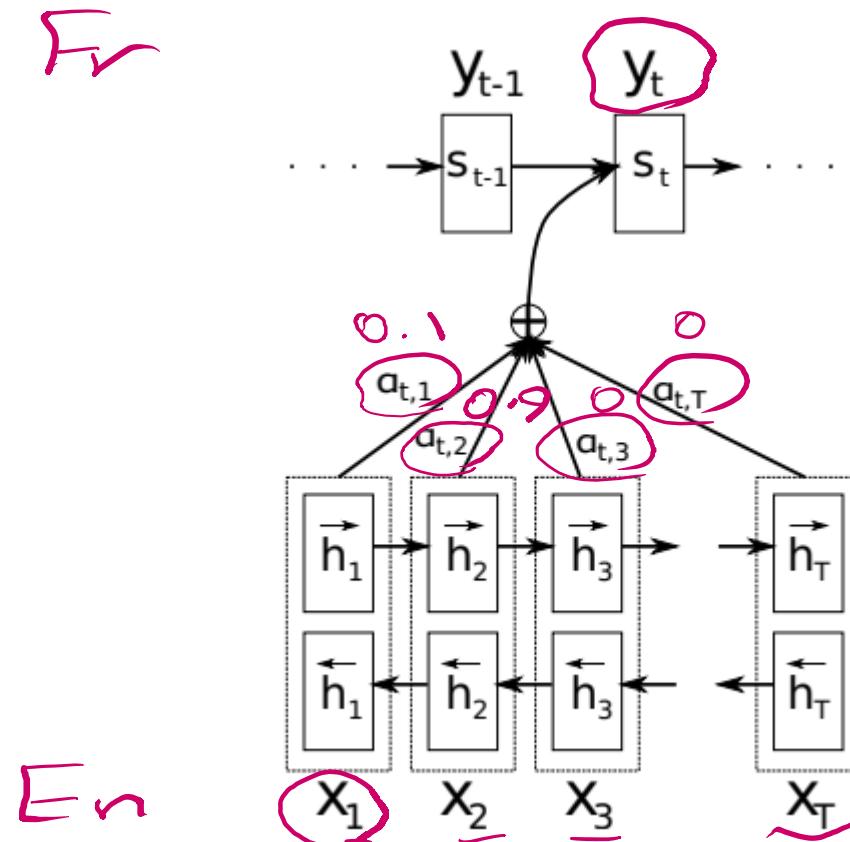


History

Attention layers to Transformers

Attention and RNNs

LSTM



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

Attention and RNNs

- RNN

LSTM

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

single hidden states

context vector

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

Attention and RNNs

- RNN

$$p(y_t | \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

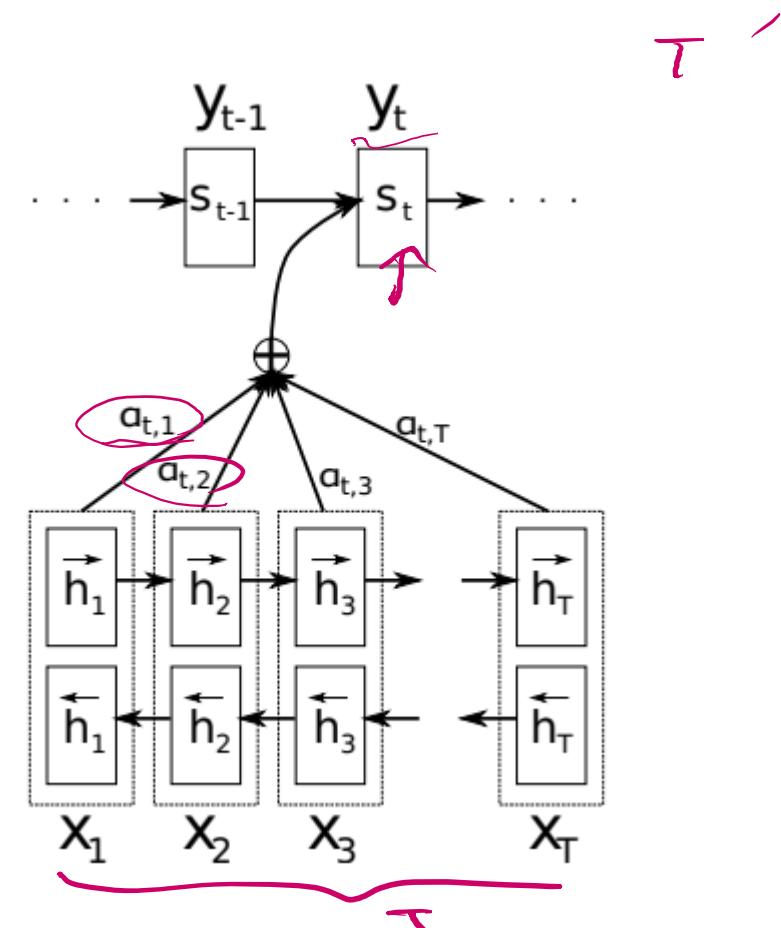
- Self-attention + RNN

$$p(y_i | y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j,$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

$c_i : T \times T'$



Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

Attention and RNNs

En

- RNN

$$p(y_t \mid \{y_1, \dots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c),$$

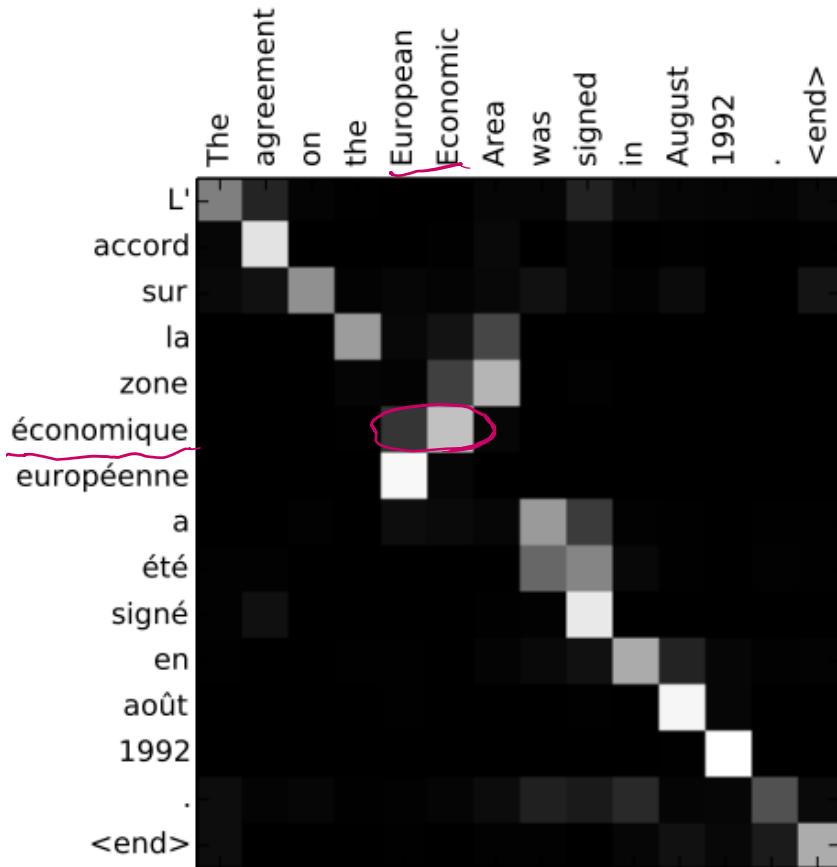
En

- Self-attention + RNN

$$p(y_i \mid y_1, \dots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

$$c_i = \sum_{j=1}^{T_x} \underbrace{\alpha_{ij}}_{\text{Attention weight}} h_j.$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

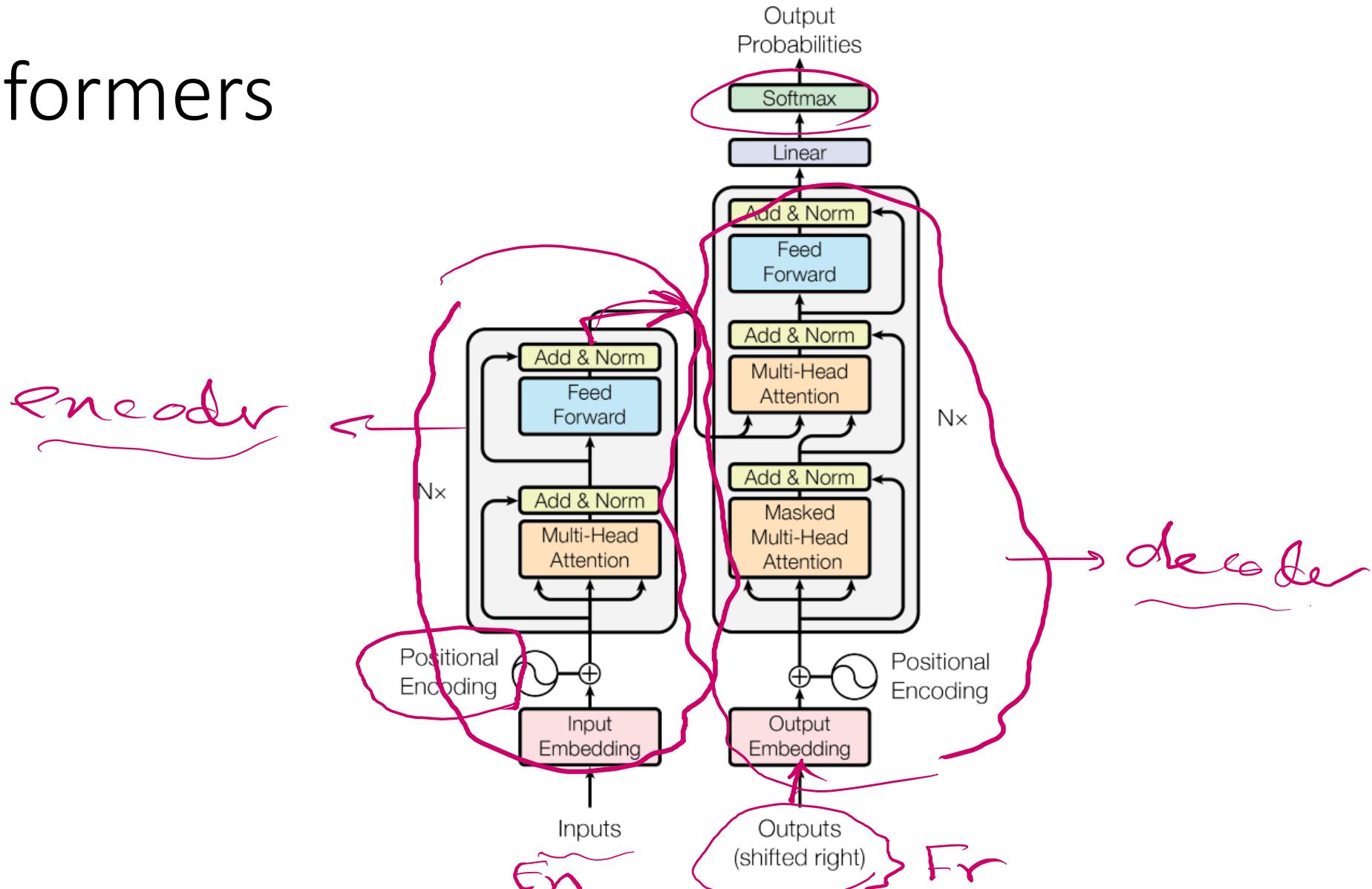


Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *arXiv preprint arXiv:1409.0473* (2014).

~~Attention and RNNs~~

- RNNs
 - Computation cost
 - Sequential (not parallel)
 - BPTT (retain computation graph)
- Alternative way to capture word dependency

Transformers



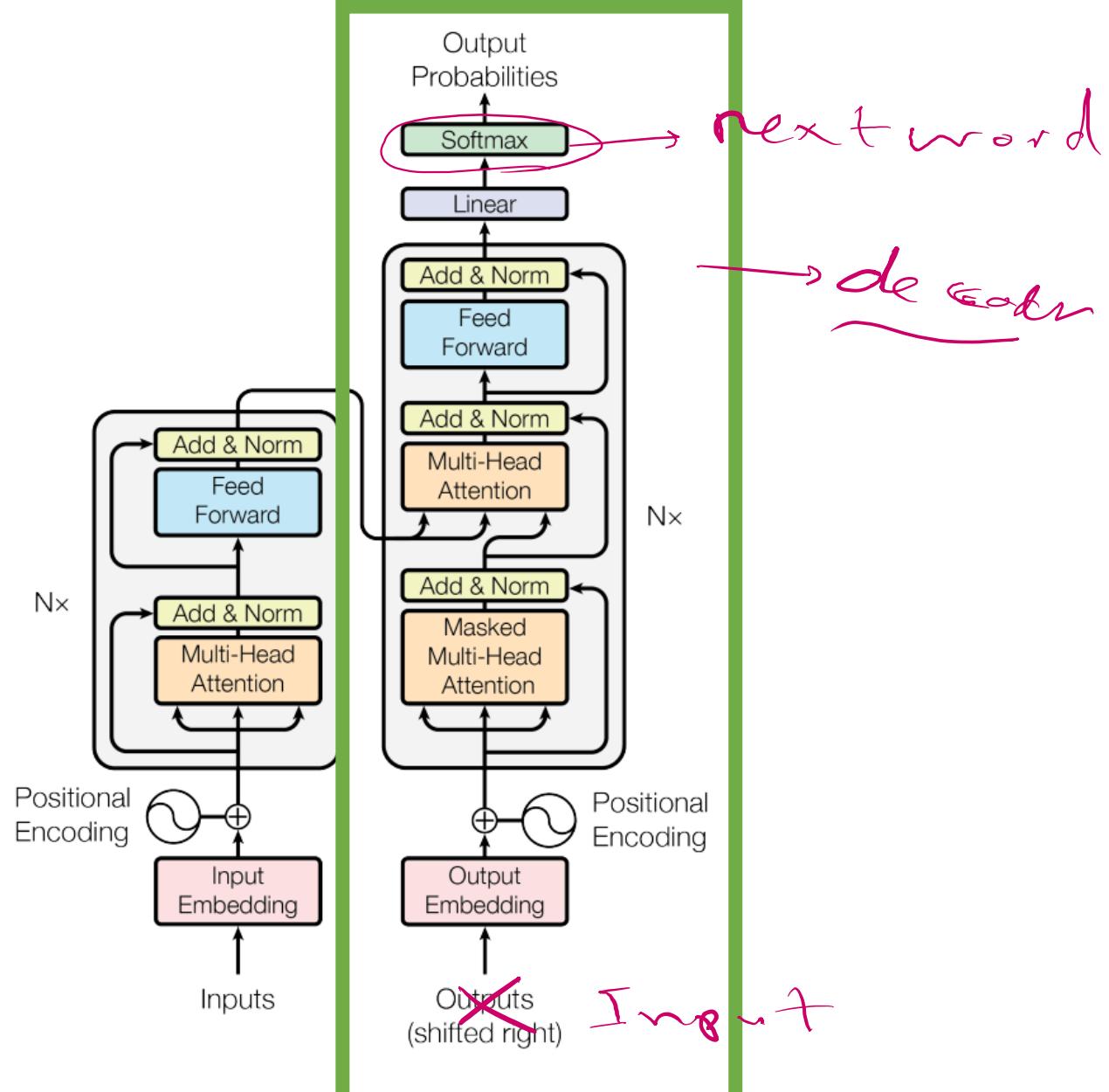
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

GPT

(Generative Pre-Training)

- Auto-regressive,
- Decoder language model

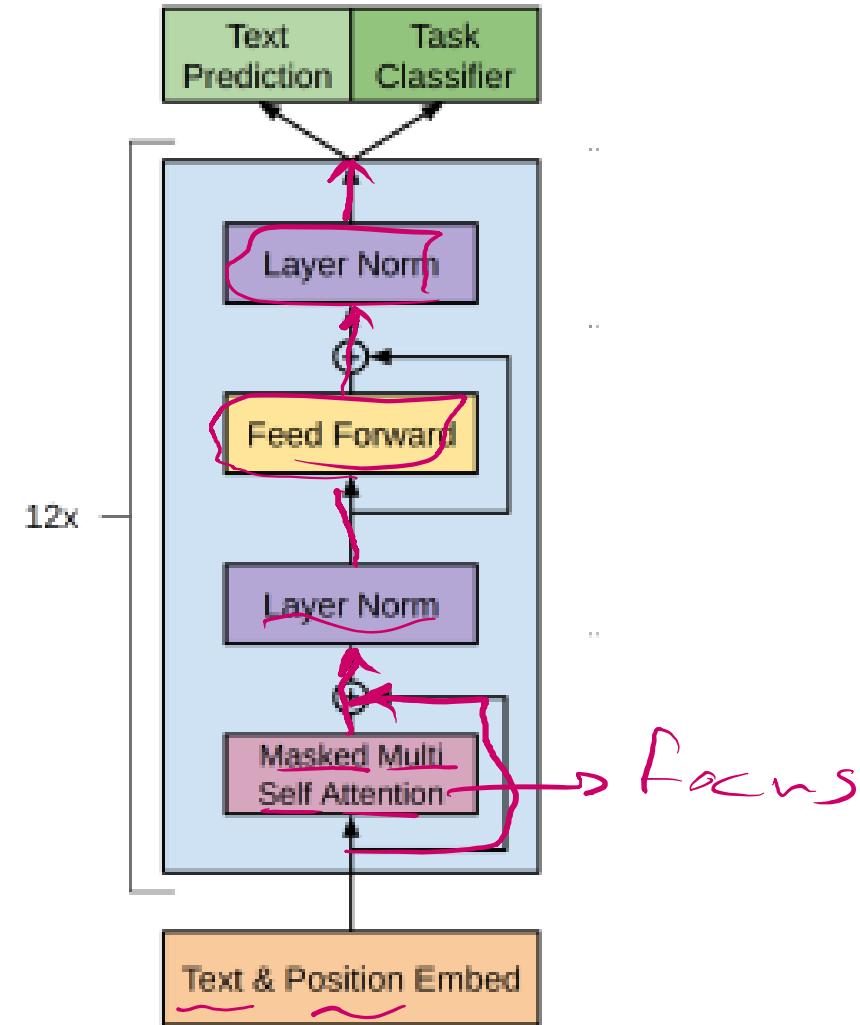
$\gamma_t \rightarrow \gamma_{t+1}$



Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT (Generative Pre-Training)

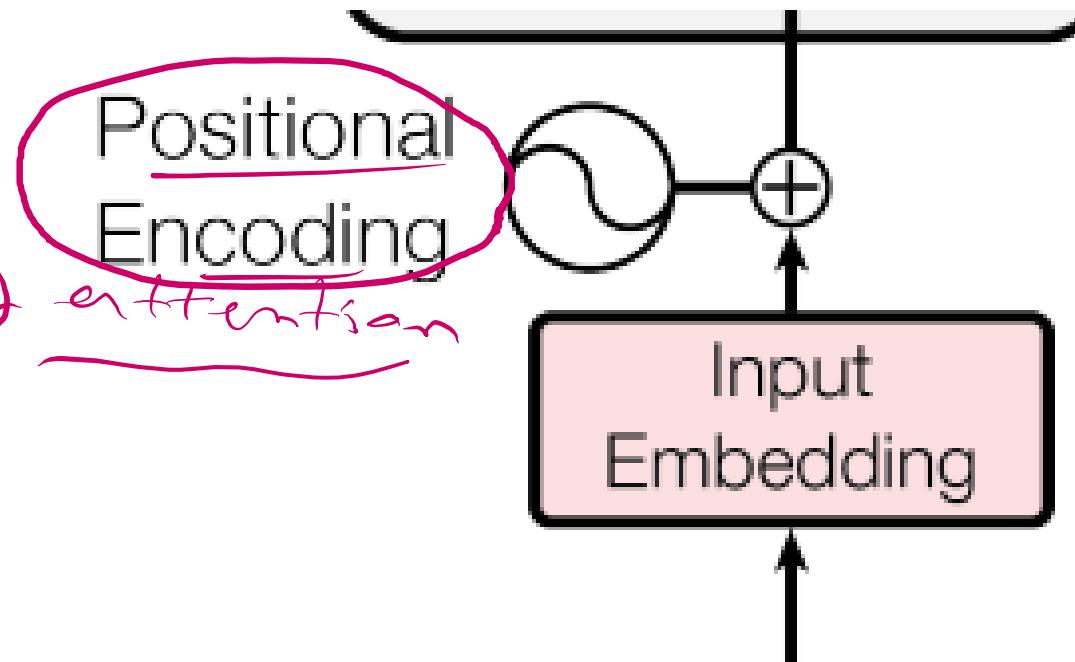
- Auto-regressive,
- Decoder language model



Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

GPT

RNN → Fully
attention



Inputs

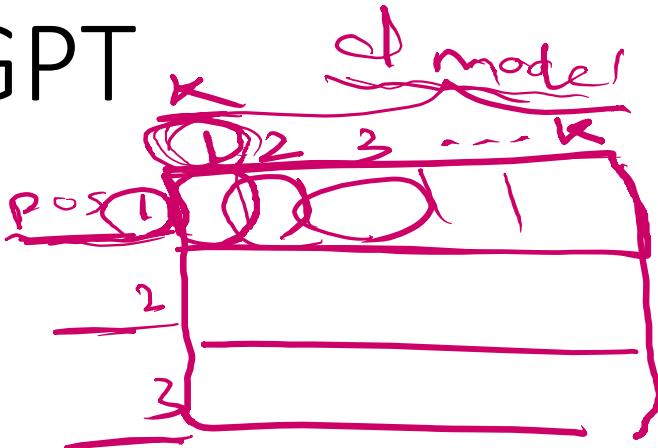
| 1 | 2 | 3 |
|Answer| the| following|
 $T = 3$

|1 | 2 | 3 |
|the| following| question|
 $T' = 3$

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

GPT

$T = 3$

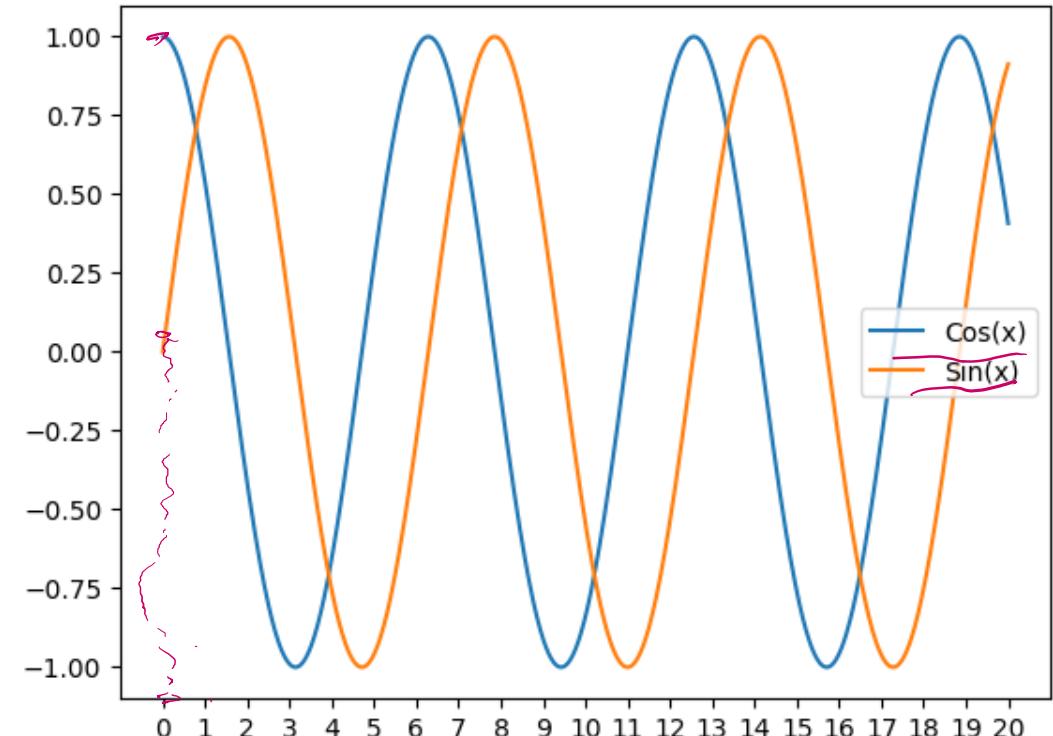


$$PE_{(pos, 2i)} = \sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos, 2i+1)} = \cos(pos/10000^{2i/d_{\text{model}}})$$

$i = 0$

| 1 | 2 | 3 |
|Answer | the | following |

$T = 3$

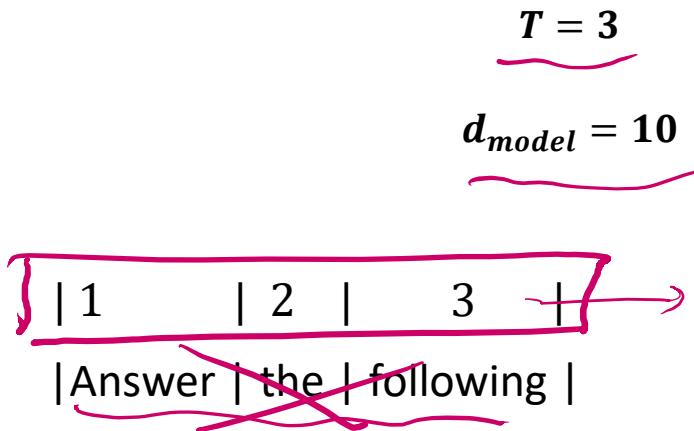


| 1 | 2 | 3 |
|the | following | question |

$T' = 3$

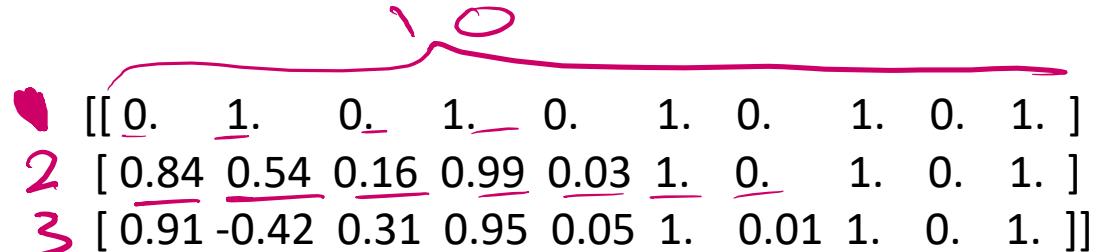
Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

GPT



$$\left. \begin{array}{l} PE_{(pos,2i)} = \sin(pos/10000^{2i/d_{model}}) \\ PE_{(pos,2i+1)} = \cos(pos/10000^{2i/d_{model}}) \end{array} \right\}$$

• Position embedding



1	[0.	1.	0.	1.	0.	1.	0.	1.	0.	1.]
2	[0.84	0.54	0.16	0.99	0.03	1.	0.	1.	0.	1.]
3	[0.91	-0.42	0.31	0.95	0.05	1.	0.01	1.	0.	1.]

GPT

$$T = 3$$

$$d_{model} = 10$$

$$vocab\ size = 6$$

- Input Embedding

| 1 | 2 | 3 |
| Answer | the | following |

$$d_{model} = 10$$

Answer	[-0.68 0.49 -0.08 0.57 0.22 -0.19 0.53 -1.19 0.75 -0.19]
Following	[0.09 0.04 -0.39 0.27 1.03 -0.01 0.07 -0.47 1.21 -0.28]
Here	[1.2 0.09 -0.44 0.12 0.06 0.08 1. -0.64 -0.52 0.05]
Please	[-1.28 -1.38 -0.26 -1.5 0.93 0.16 0.56 1.84 -0.41 -1.45]
Question	[-0.43 0.09 0.67 -0.06 -1.52 1.78 0.5 -1.38 -0.47 -2.]
The	[-0.8 0.54 1.66 -0.55 1.2 -2.73 1.02 -0.25 -0.36 -1.85]]

GPT

$$T = 3$$

$$d_{model} = 10$$

$$vocab\ size = 6$$

Input Embedding

+

Position Embedding

numpy | PyTorch emb $\in [1, 6, 2], \cdot$

| 1 | 2 | 3 |
+
| Answer | the | following |

Answer $[-0.68 \ 0.49 \ -0.08 \ 0.57 \ 0.22 \ -0.19 \ 0.53 \ -1.19 \ 0.75 \ -0.19]$

The $[-0.8 \ 0.54 \ 1.66 \ -0.55 \ 1.2 \ -2.73 \ 1.02 \ -0.25 \ -0.36 \ -1.85]$

Following $[0.09 \ 0.04 \ -0.39 \ 0.27 \ 1.03 \ -0.01 \ 0.07 \ -0.47 \ 1.21 \ -0.28]$

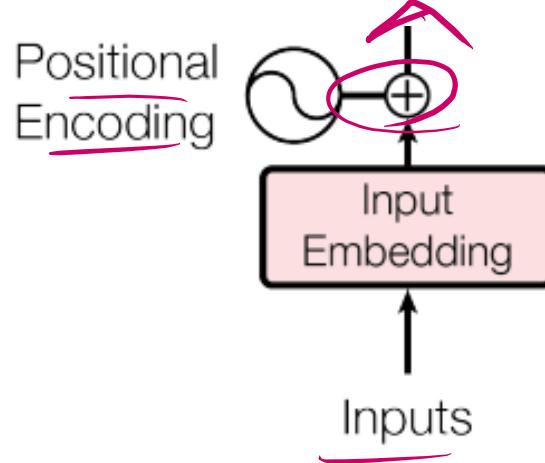
+

1 $[[0. \ 1. \ 0. \ 1. \ 0. \ 1. \ 0. \ 1. \ 0. \ 1.]$

2 $[0.84 \ 0.54 \ 0.16 \ 0.99 \ 0.03 \ 1. \ 0. \ 1. \ 0. \ 1.]$

3 $[0.91 \ -0.42 \ 0.31 \ 0.95 \ 0.05 \ 1. \ 0.01 \ 1. \ 0. \ 1.]]$

GPT



(○)

3 { Answer [-0.68 0.49 -0.08 0.57 0.22 -0.19 0.53 -1.19 0.75 -0.19]
The [-0.8 0.54 1.66 -0.55 1.2 -2.73 1.02 -0.25 -0.36 -1.85]
Following [0.09 0.04 -0.39 0.27 1.03 -0.01 0.07 -0.47 1.21 -0.28]]

X =

| 1 | 2 | 3 |

+

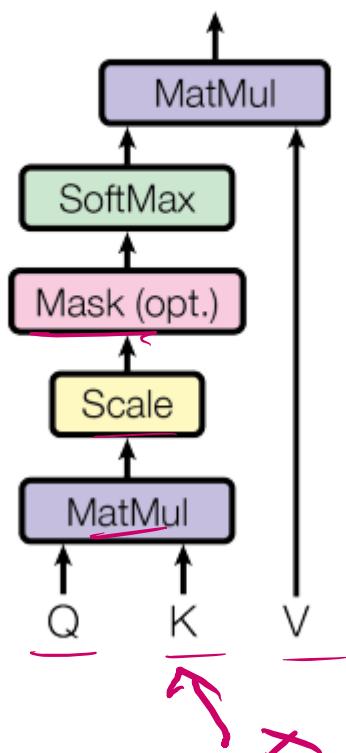
|Answer | the | following |

(○)

3 { 1 [[0. 1. 0. 1. 0. 1. 0. 1. 0. 1.]]
2 [0.84 0.54 0.16 0.99 0.03 1. 0. 1. 0. 1.]
3 [0.91 -0.42 0.31 0.95 0.05 1. 0.01 1. 0. 1.]]

Self-attention

Scaled Dot-Product Attention

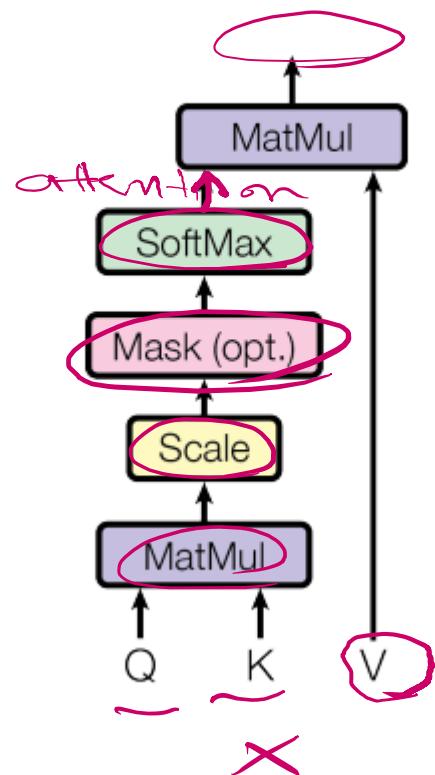


$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Self-attention

Scaled Dot-Product Attention



- Query
- Key
- Value
- Projections of input

Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

Self-attention

- $x = \underbrace{pos}_{(3 \times 1)} + \underbrace{token}_{(3 \times 10)}$
- Fully Connected (x)
 $(3 \times 10 * 3)$
- Reshape $\overbrace{T}^d_{model} \times 3$
 - $Q = (3 \times 10)$
 - $K = (3 \times 10)$
 - $V = (3 \times 10)$

- Q

$\left[\begin{array}{cccccccccc} 0.84 & 1.22 & -0.1 & 0.68 & -0.11 & -0.54 & -0.54 & -0.74 & -0.71 & 0.09 \\ -0.42 & 1.38 & 0.96 & 1.87 & -0.58 & 0.05 & 0.45 & -1.06 & 0.21 & -1.24 \\ -0.19 & 0.33 & -1.1 & -0.27 & -0.64 & 0.75 & -0.42 & 0.51 & -1.44 & -0.48 \end{array} \right]$

- K

$d_{model} = 10$

$\left[\begin{array}{cccccccccc} 0.16 & -1.08 & -0.9 & 0.33 & -0.65 & -0.09 & 0.83 & -0.3 & 0.51 & -0.03 \\ -0.22 & -0.74 & -1.61 & 0.89 & 1.08 & 1.66 & 0.2 & 1.18 & 0.24 & -1.74 \\ 0.15 & 0.49 & 0.43 & 0.16 & -1.04 & -0.7 & 0.33 & -0.61 & 0.94 & 0.5 \end{array} \right]$

- V

$\left[\begin{array}{cccccccccc} 0.54 & -0.7 & 0.14 & 0.96 & -0.74 & -0.04 & 0.11 & 0.33 & -0.03 & 0.6 \\ -0.02 & 0.49 & 1.3 & 1.81 & -0.94 & -0.6 & -0.02 & 1.24 & -0.47 & 0.89 \\ -0.04 & -1.8 & -0.44 & -0.74 & 0.05 & 0.19 & 0.35 & -0.85 & -0.43 & 0.03 \end{array} \right]$

Self-attention

$3 \times 10 \times 10 \times 3 \rightarrow 3 \times 3$

1. $\frac{QK^T}{\sqrt{d_{model}}}$

2. Mask

3. Softmax

1. QK^T

$$\begin{bmatrix} [-0.42 & -0.83 & 0.29] \\ [-0.19 & -0.1 & 0.71] \\ [-0.12 & 0.92 & -0.73] \end{bmatrix}$$

3. $1 / \sqrt{d_{model}}$

$$\begin{bmatrix} [-0.42 & \text{inf} & \text{inf}] \\ [-0.19 & -0.1 & \text{inf}] \\ [-0.12 & 0.92 & -0.73] \end{bmatrix}$$

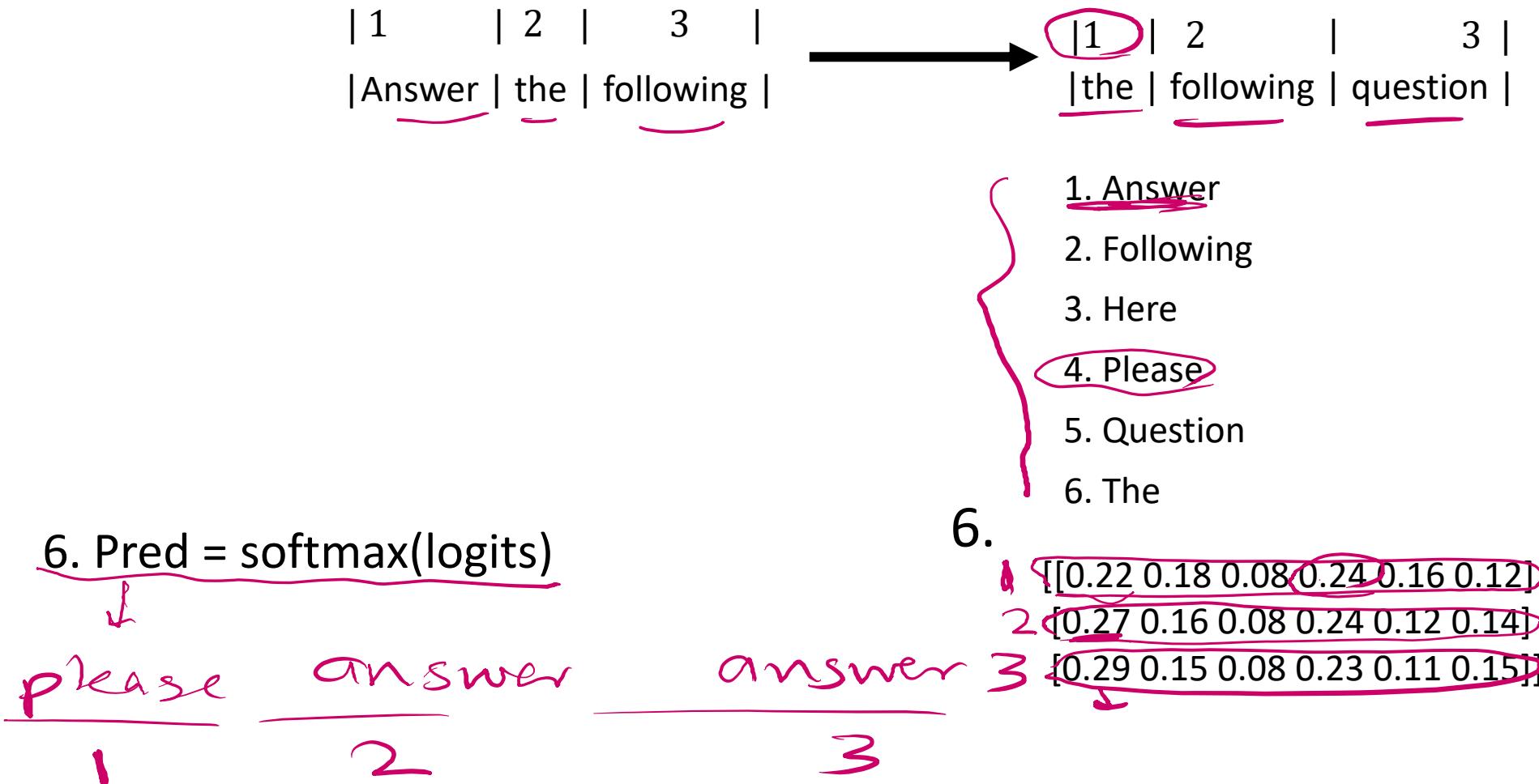
1 2 3
5. $\begin{bmatrix} [1. & 0. & 0.] \\ [0.48 & 0.52 & 0.] \\ [0.23 & 0.65 & 0.13] \end{bmatrix}$

Softmax

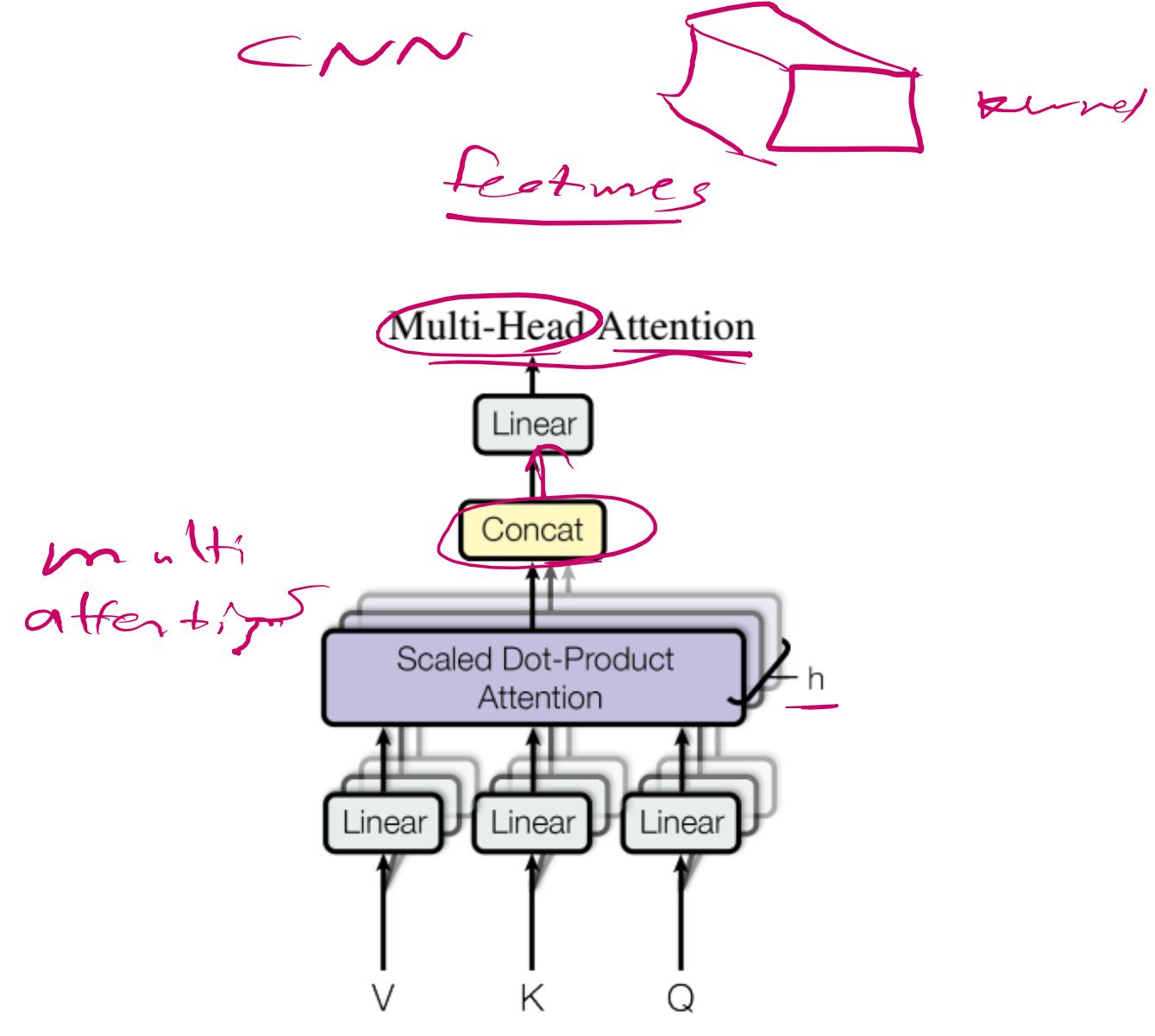
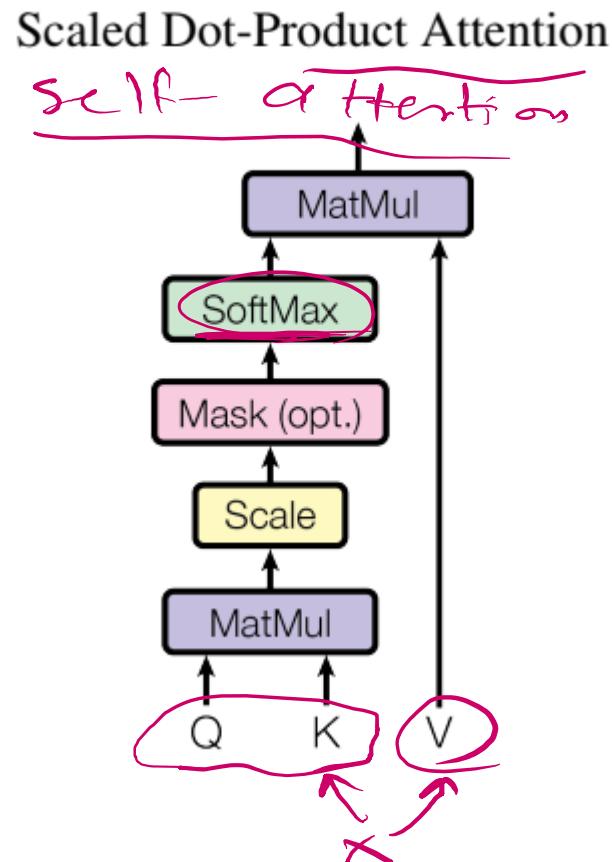
Self-attention

4. $\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$
5. Logits = Fully Connected (Attention)
 $(3 \times \text{vocab size})$
6. Pred = softmax(logits)
4. $\begin{matrix} 3 \times 3 & \times & 3 \times 10 \\ \uparrow & & \uparrow \\ \text{attention} & & d_{\text{model}} \\ & & = 3 \times 10 \end{matrix}$
4. $\begin{bmatrix} 0.54 & -0.7 & 0.14 & 0.96 & -0.74 & -0.04 & 0.11 & 0.33 & -0.03 & 0.6 \\ 0.24 & -0.08 & 0.75 & 1.4 & -0.84 & -0.33 & 0.04 & 0.81 & -0.26 & 0.75 \\ 0.1 & -0.07 & 0.82 & 1.3 & -0.77 & -0.37 & 0.06 & 0.77 & -0.36 & 0.72 \end{bmatrix}$
5. $\begin{bmatrix} 0.27 & 0.04 & -0.69 & 0.35 & -0.04 & -0.38 \\ 0.44 & -0.11 & -0.84 & 0.29 & -0.43 & -0.23 \\ 0.51 & -0.1 & -0.83 & 0.27 & -0.45 & -0.15 \end{bmatrix}$
6. $\begin{bmatrix} 0.22 & 0.18 & 0.08 & 0.24 & 0.16 & 0.12 \\ 0.27 & 0.16 & 0.08 & 0.24 & 0.12 & 0.14 \\ 0.29 & 0.15 & 0.08 & 0.23 & 0.11 & 0.15 \end{bmatrix}$

Self-attention



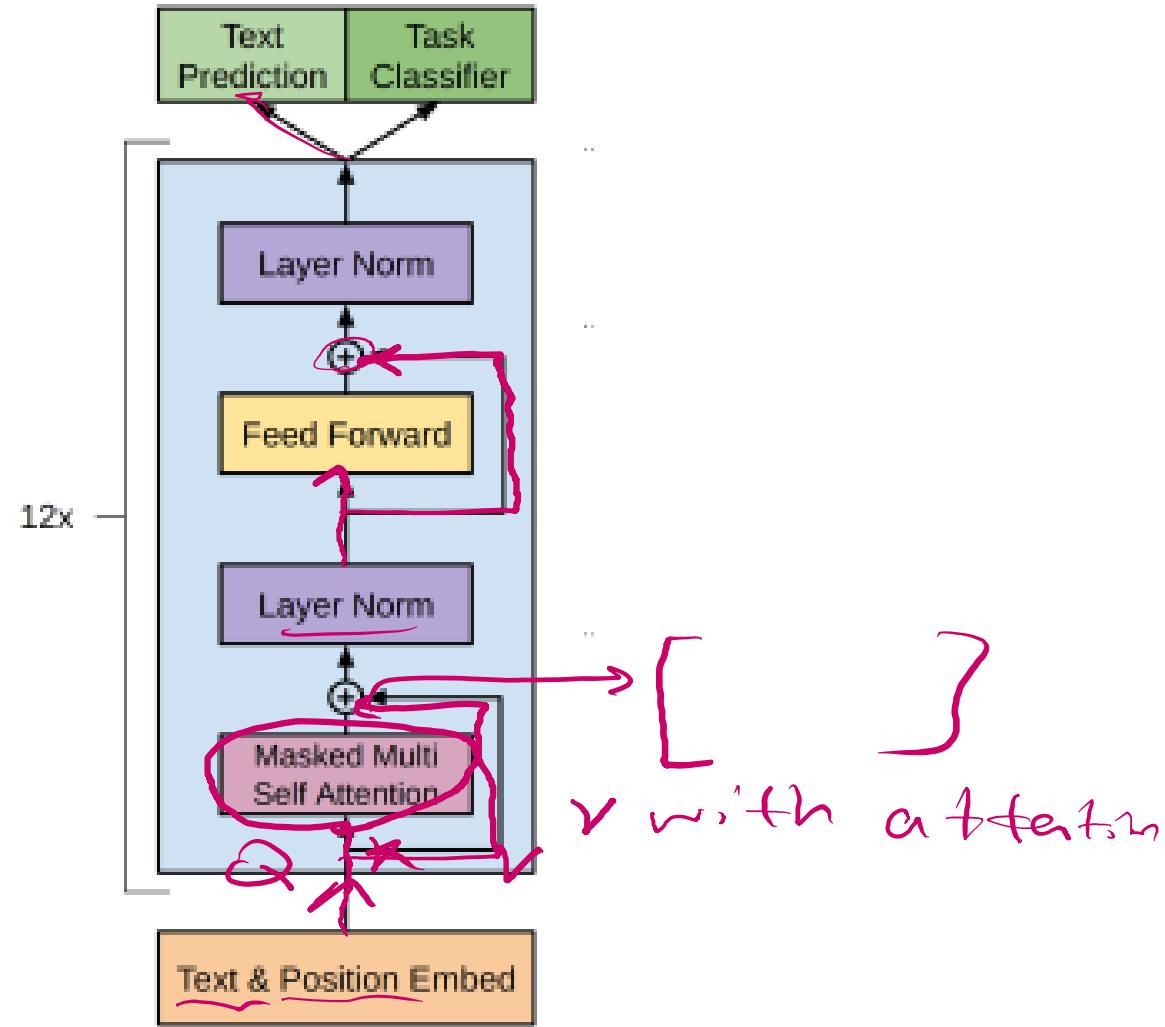
Multi-head attention



Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

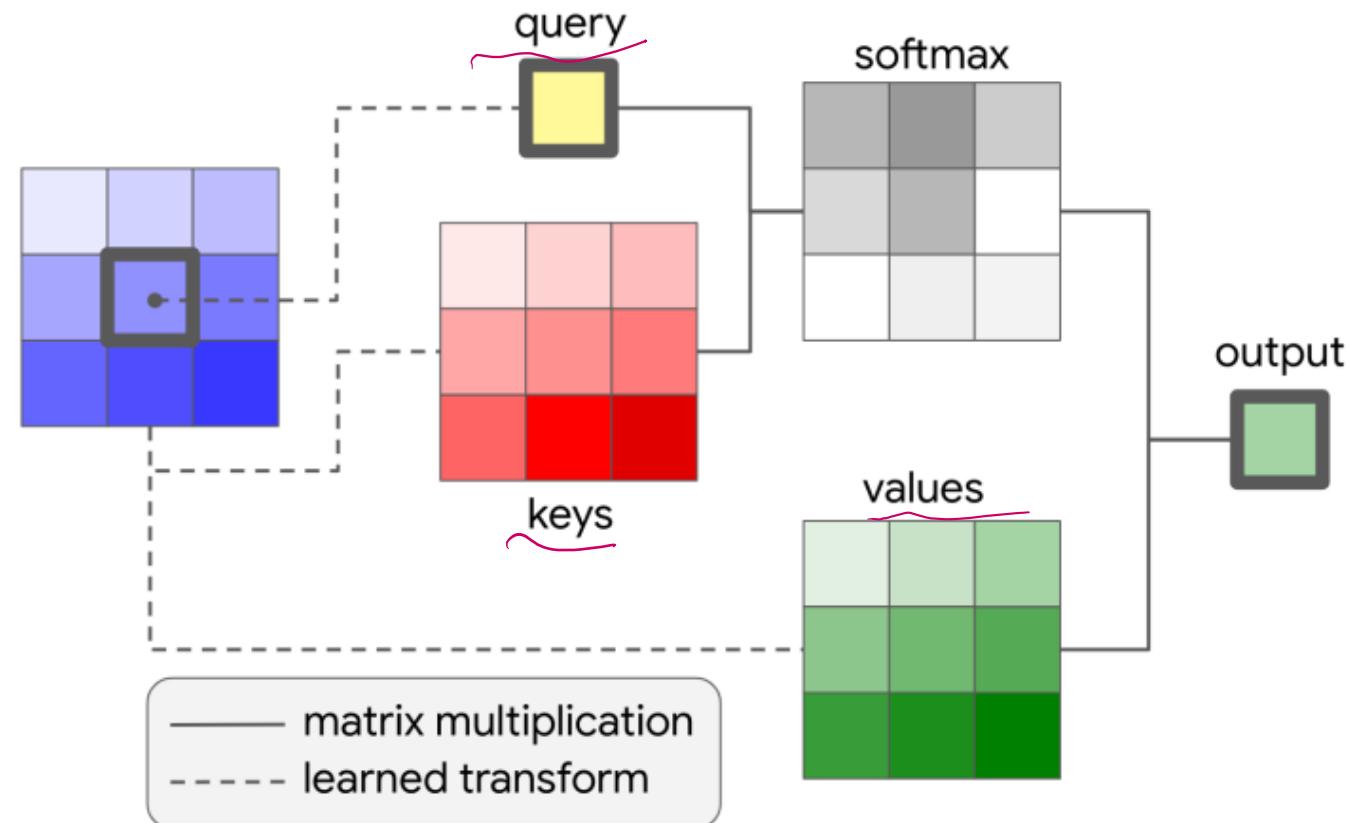
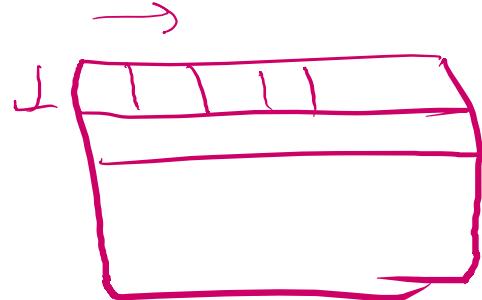
GPT (Generative Pre-Training)

- Auto-regressive,
- Decoder language model



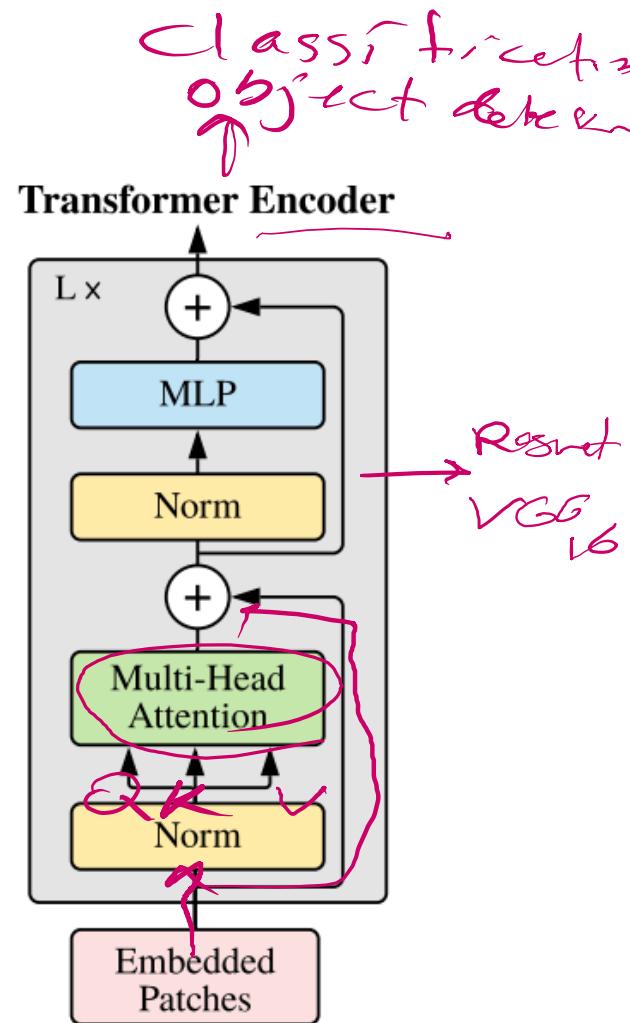
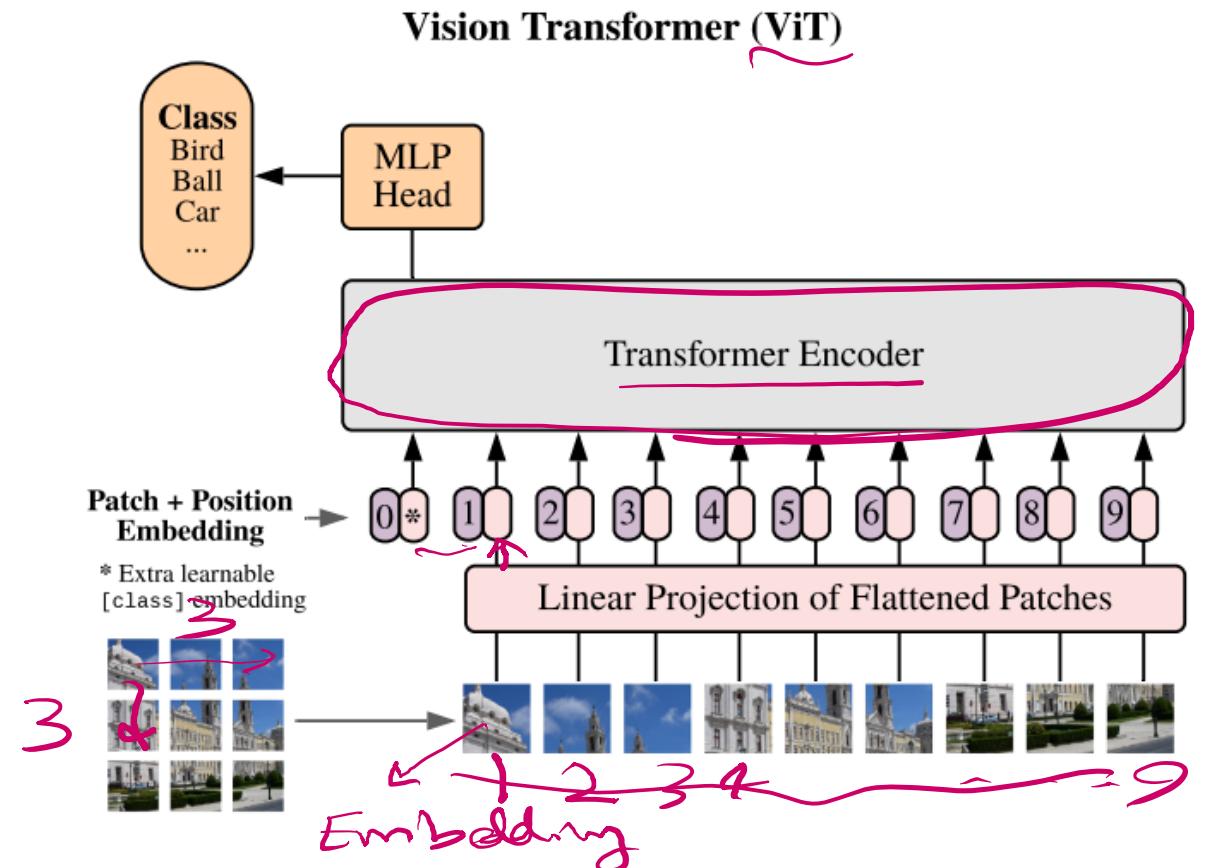
Radford, Alec, et al. "Improving language understanding by generative pre-training." (2018).

Attention to Vision



Ramachandran, Prajit, et al. "Stand-alone self-attention in vision models." *Advances in neural information processing systems 32* (2019).

Vision Transformers



Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).