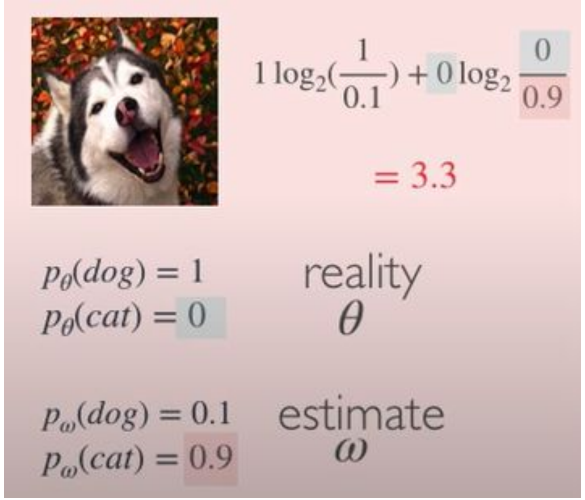
# LossFunction+Transformer

Nirbhay Sharma (B19CSE114)
(video: https://www.youtube.com/watch?v=wwb9XL1Gx_Y)

# KL - Divergence Loss Function

- Measure to see how differ two distributions are



$$1\log_2\left(\frac{1}{0.1}\right)+0\log_2\frac{0}{0.9}$$

$$= 3.3$$

$p_\theta(dog) = 1$ reality

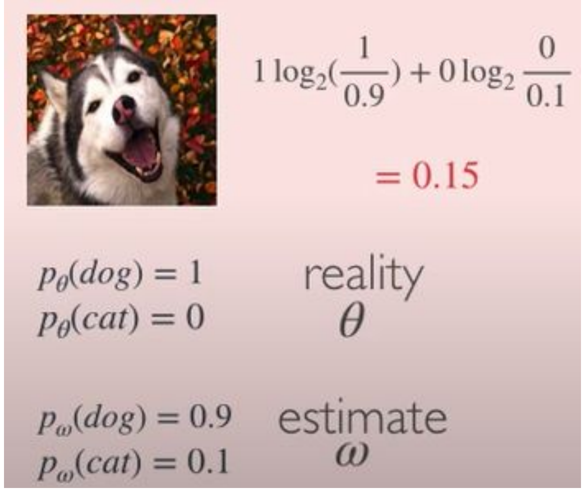
$p_\theta(cat) = 0$ $\theta$

$p_\omega(dog) = 0.1$ estimate

$p_\omega(cat) = 0.9$ $\omega$

$$\sum_{x \in Outcomes} p_\theta(x)\log_2\frac{p_\theta(x)}{p_\omega(x)}$$

$$L(y_{\text{pred}},\ y_{\text{true}}) = y_{\text{true}} \cdot \log\frac{y_{\text{true}}}{y_{\text{pred}}}$$



$$1\log_2\left(\frac{1}{0.9}\right)+0\log_2\frac{0}{0.1}$$

$$= 0.15$$

$p_\theta(dog) = 1$ reality

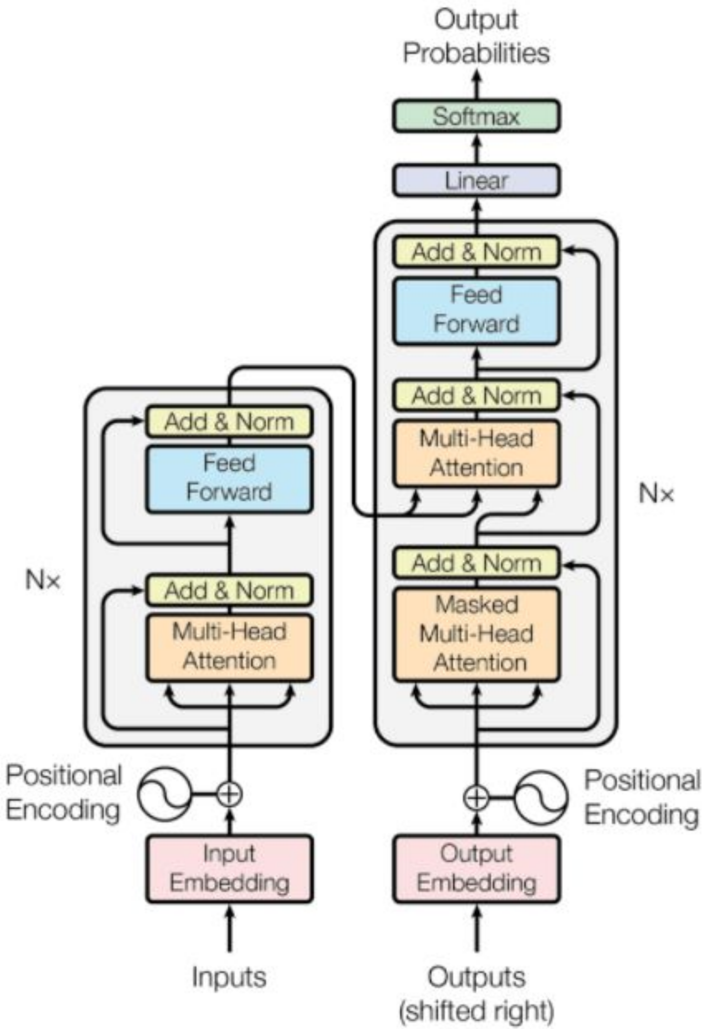
$p_\theta(cat) = 0$ $\theta$

$p_\omega(dog) = 0.9$ estimate

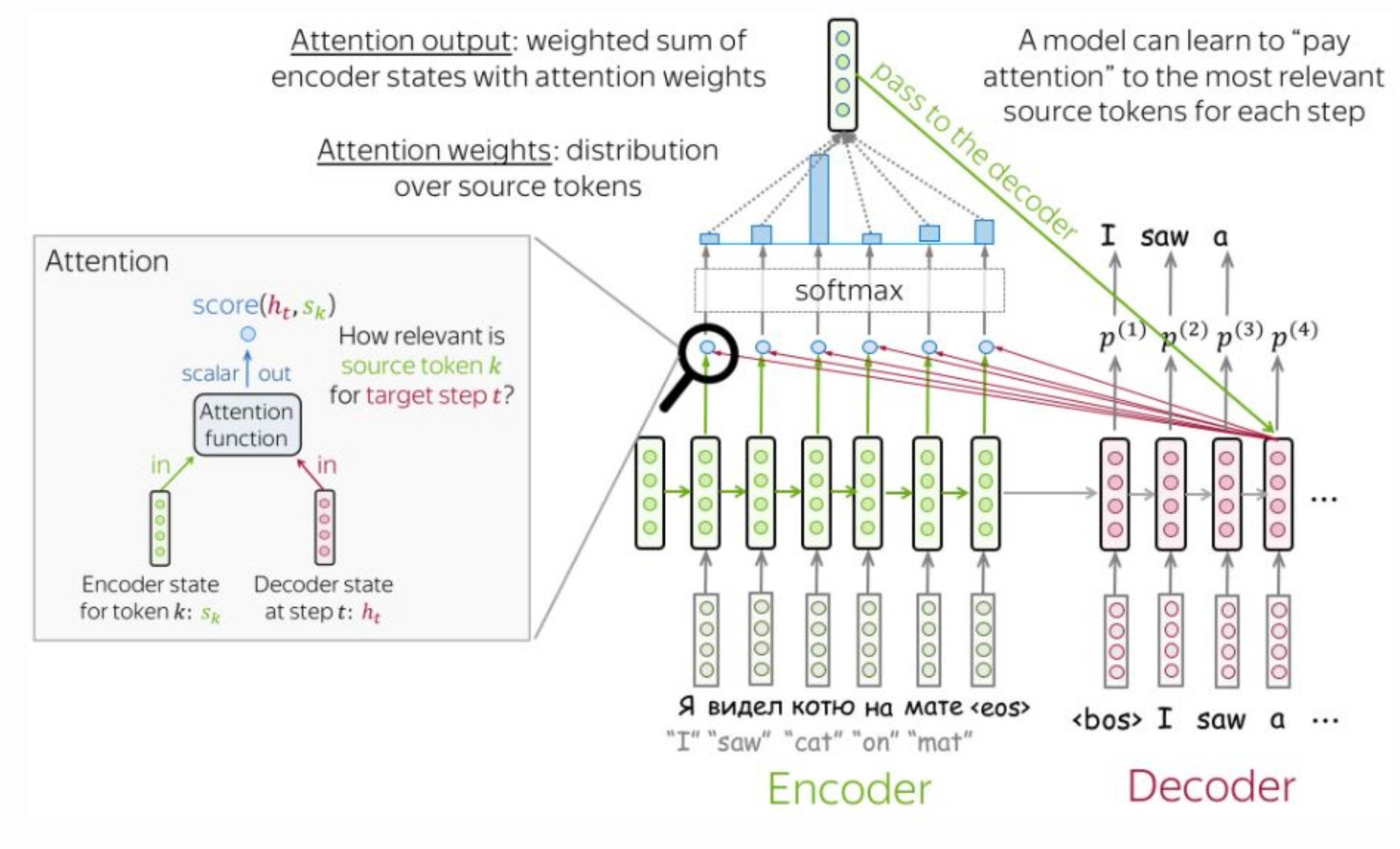
$p_\omega(cat) = 0.1$ $\omega$

Probable use cases - Multiclass classification, comparing prediction distribution with training data distribution

# Transformer Architecture



# Attention in Encoder Decoder Architecture



Attention output: weighted sum of encoder states with attention weights

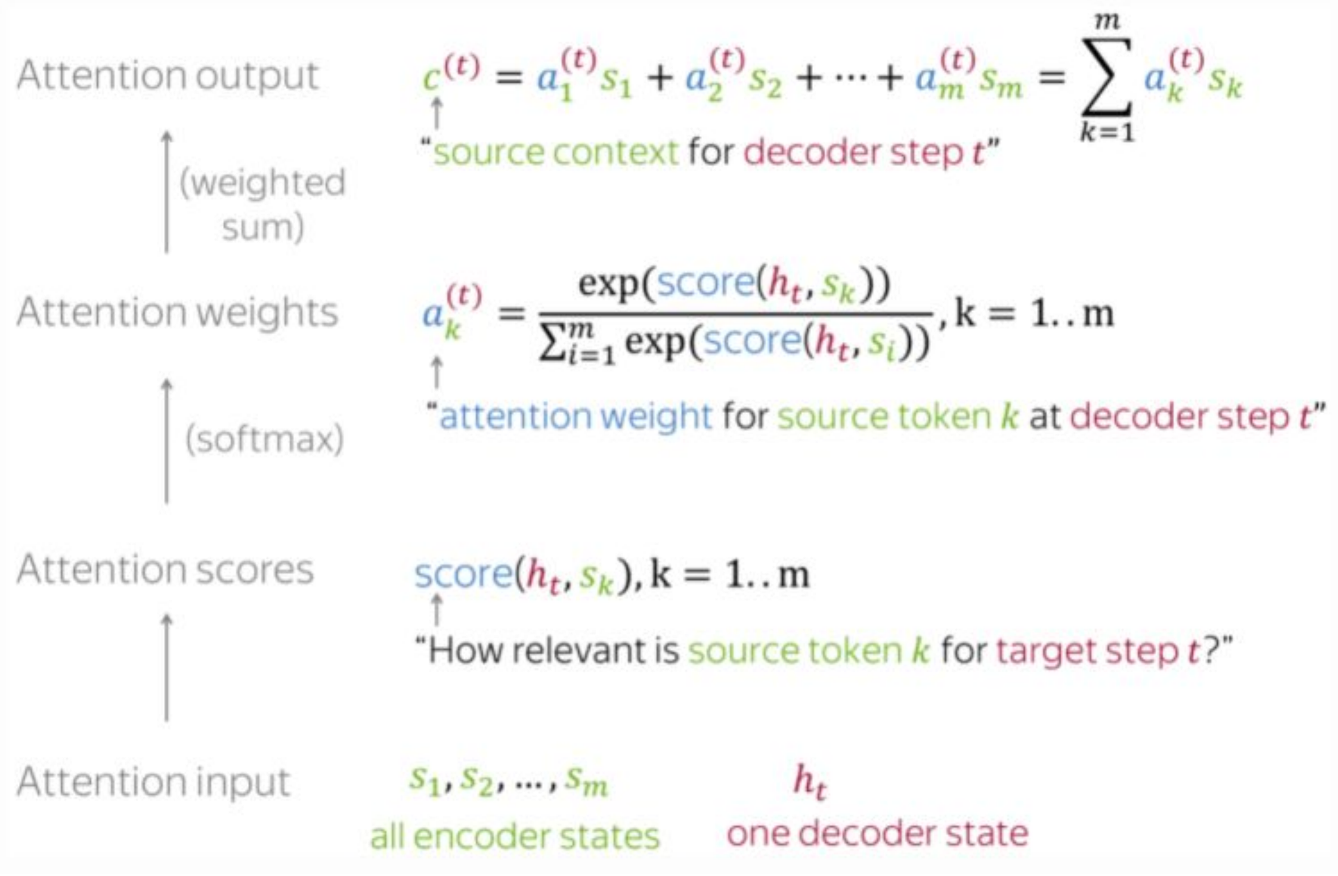
A model can learn to "pay attention" to the most relevant source tokens for each step

Attention weights: distribution over source tokens

How relevant is source token $k$ for target step $t$?

Attention
$\text{score}(h_t, s_k)$
scalar
out
Attention function
in
in
Encoder state for token $k$: $s_k$
Decoder state at step $t$: $h_t$
pass to the decoder
softmax
I saw a
$p^{(1)}$ $p^{(2)}$ $p^{(3)}$ $p^{(4)}$
...
Я видел котю на мате <eos>
"I" "saw" "cat" "on" "mat"
<bos> I saw a ...
Encoder
Decoder

# Attention in Encoder Decoder Architecture



Attention output

$$c^{(t)} = a_1^{(t)} s_1 + a_2^{(t)} s_2 + \dots + a_m^{(t)} s_m = \sum_{k=1}^{m} a_k^{(t)} s_k$$

"source context for decoder step $t$"

(weighted sum)

Attention weights

$$a_k^{(t)} = \frac{\exp(\text{score}(h_t, s_k))}{\sum_{i=1}^{m} \exp(\text{score}(h_t, s_i))}, \mathrm{k} = 1..\mathrm{m}$$

"attention weight for source token $k$ at decoder step $t$"

(softmax)

Attention scores

$$\text{score}(h_t, s_k), \mathrm{k} = 1..\mathrm{m}$$

"How relevant is source token $k$ for target step $t$?"

Attention input

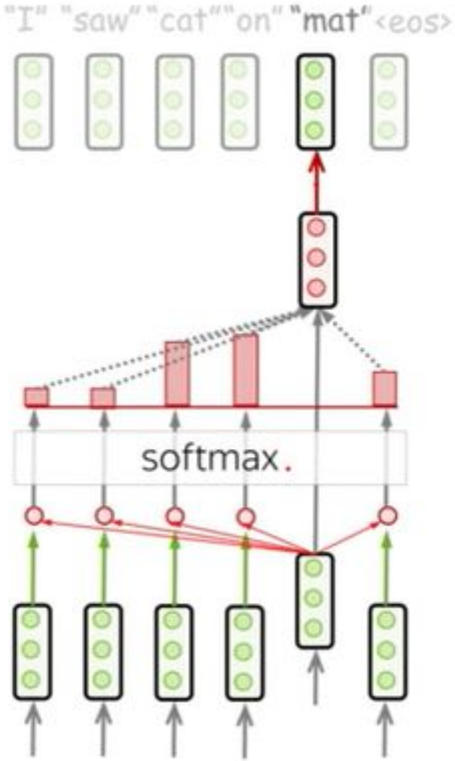
$s_1, s_2, \dots, s_m$

all encoder states

$h_t$

one decoder state

# Self-Attention



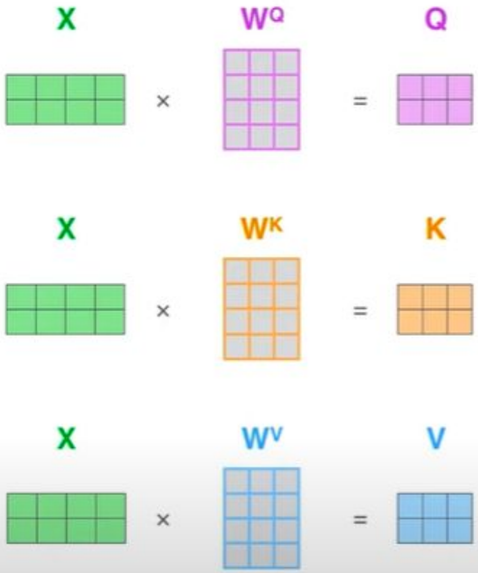
# Positional Encodings

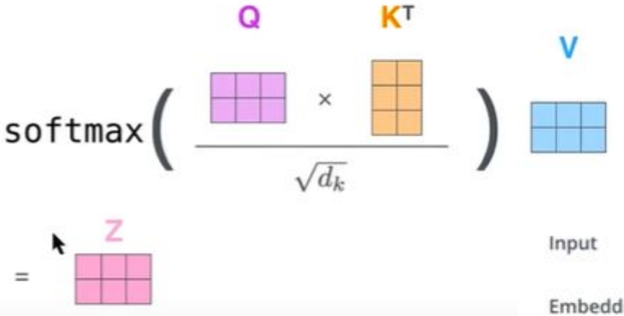
- Need ?
- Properties
  - Unique, bounded, deterministic

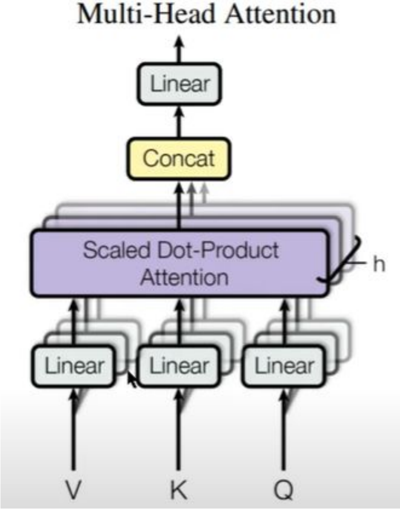
$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k + 1 \end{cases}$$

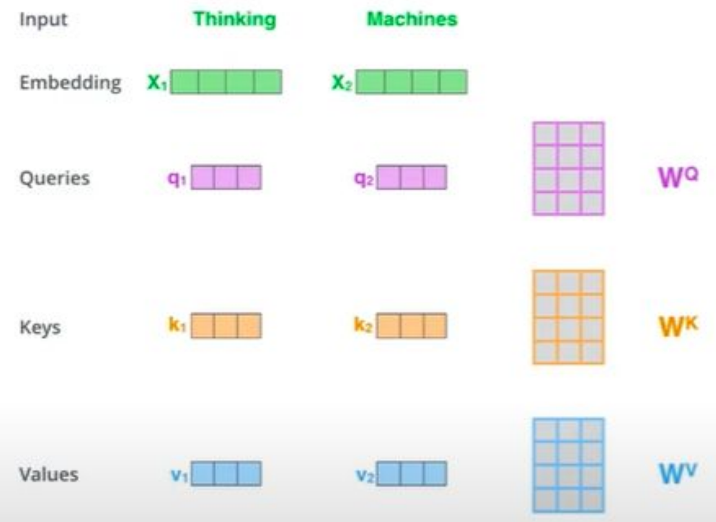
$$\omega_k = \frac{1}{10000^{2k/d}}$$

# Self-Attention

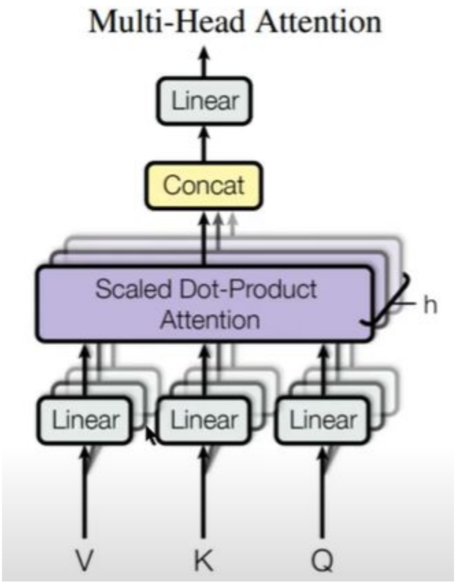


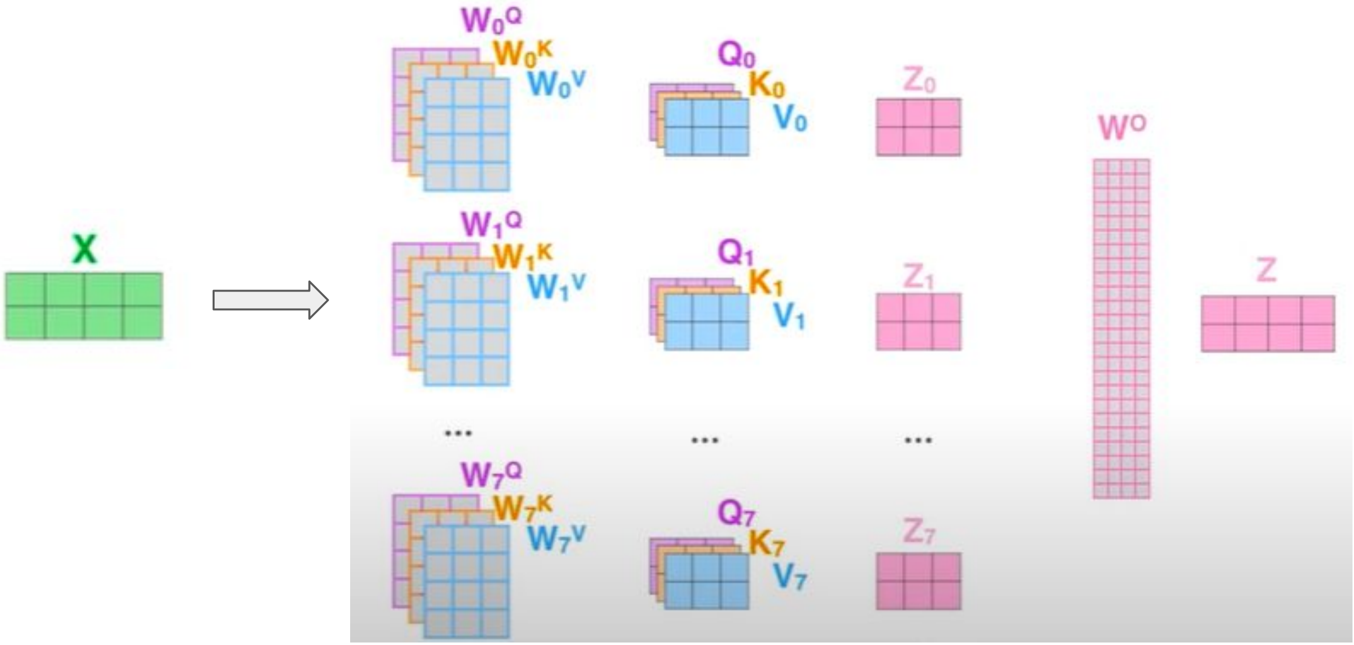




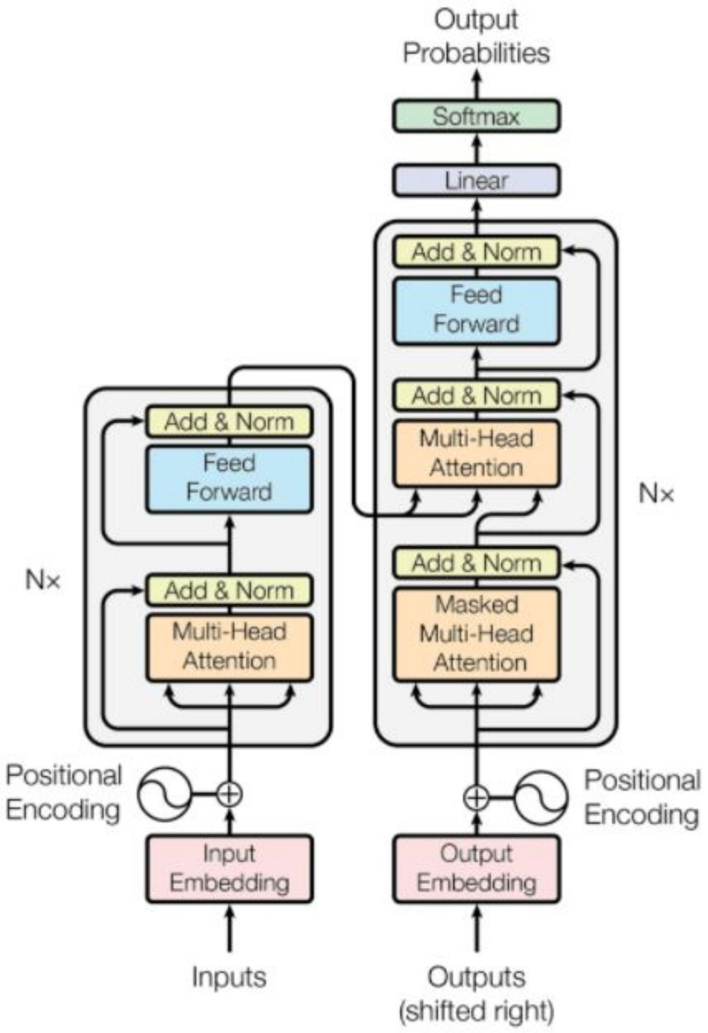


# Multihead-attention



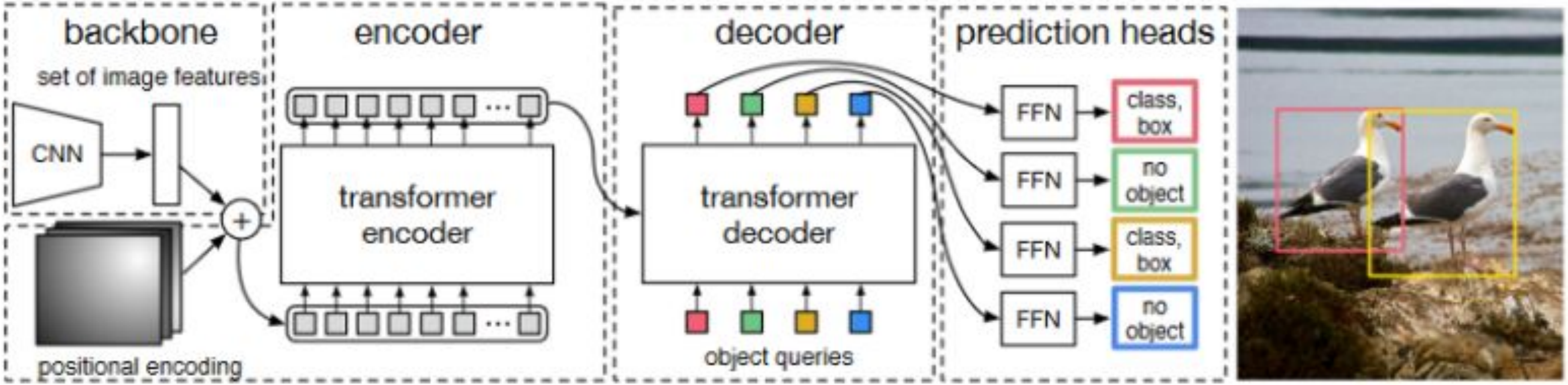


# Transformer Architecture



# Application of Transformer

- Vision Transformer
- Transformer for object detection (DETR)



# Credits and References

- https://lilianweng.github.io/posts/2018-06-24-attention/
- https://kazemnejad.com/blog/transformer_architecture_positional_encoding/
- https://lena-voita.github.io/nlp_course/seq2seq_and_attention.html

# Thanks