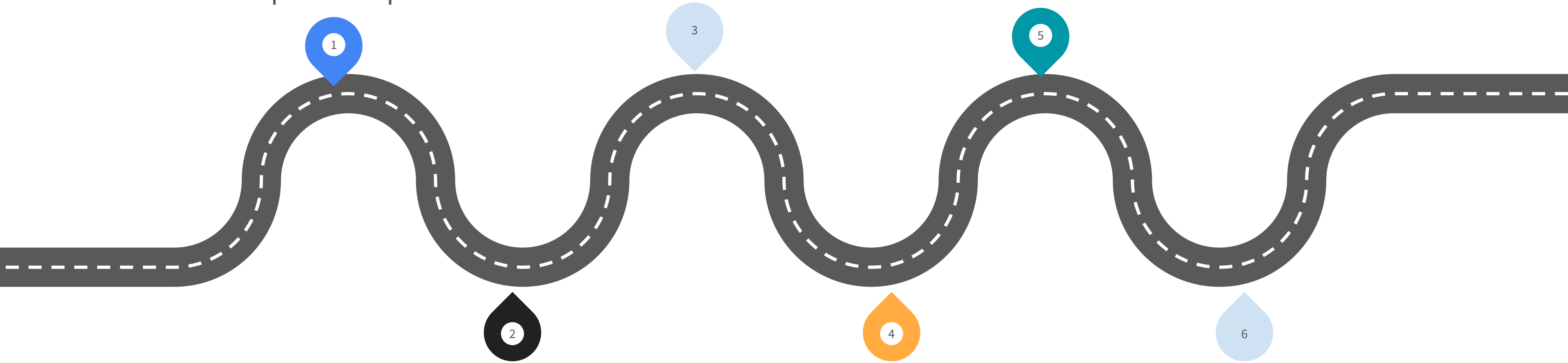
Attention and transformers

Сьогодні на лекції

Рекурентні нейронні мережі

Механізм уваги

Види уваги

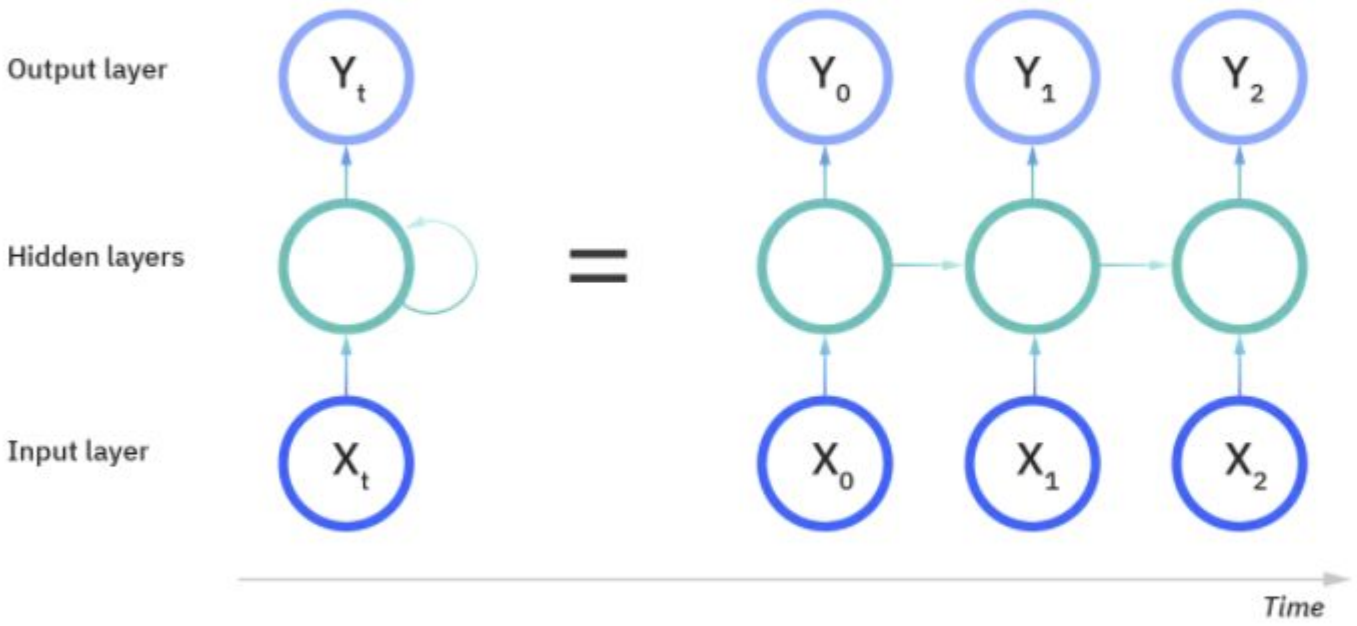


Моделі seq2seq

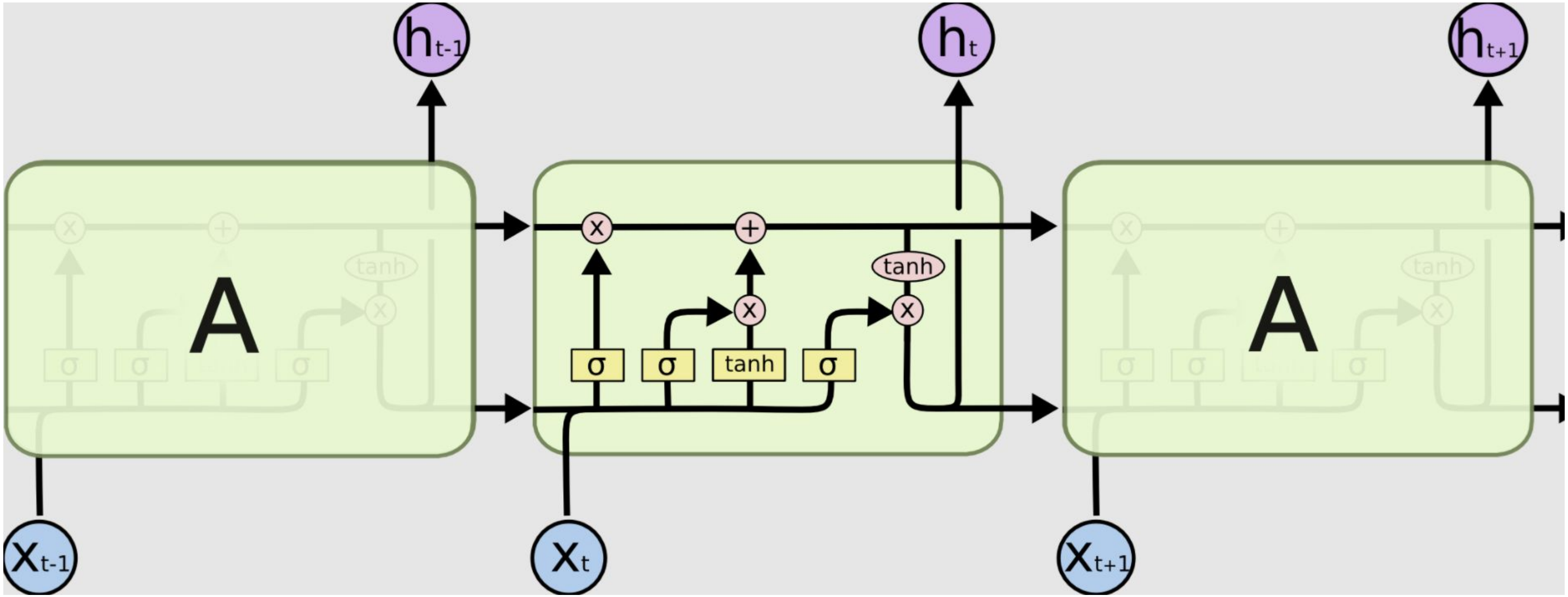
Трансформери

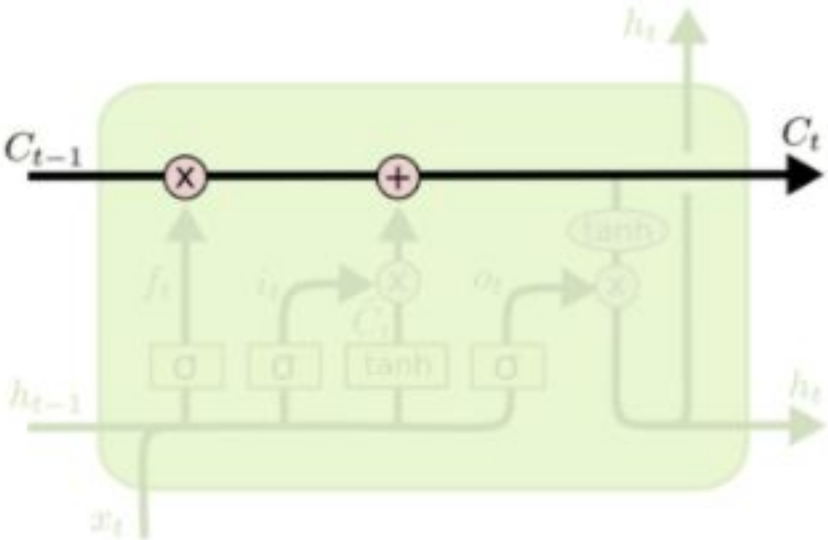
Зоровий трансформер

Recurrent Neural Networks

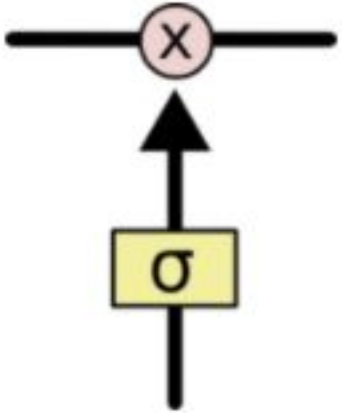


Long short-term memory (LSTM)

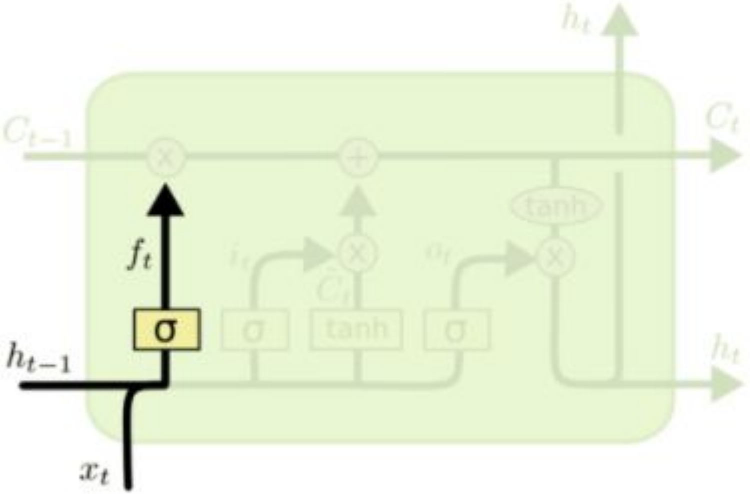




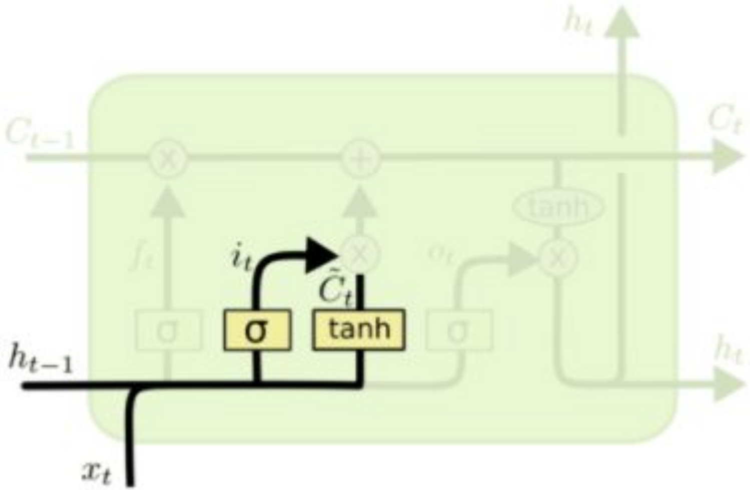
C_{t-1}
C_t
h_t
h_{t-1}
x_t
tanh



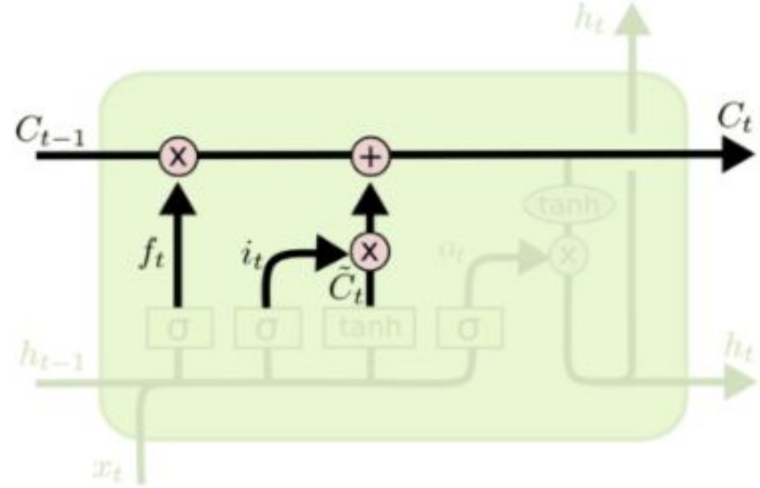
X
σ



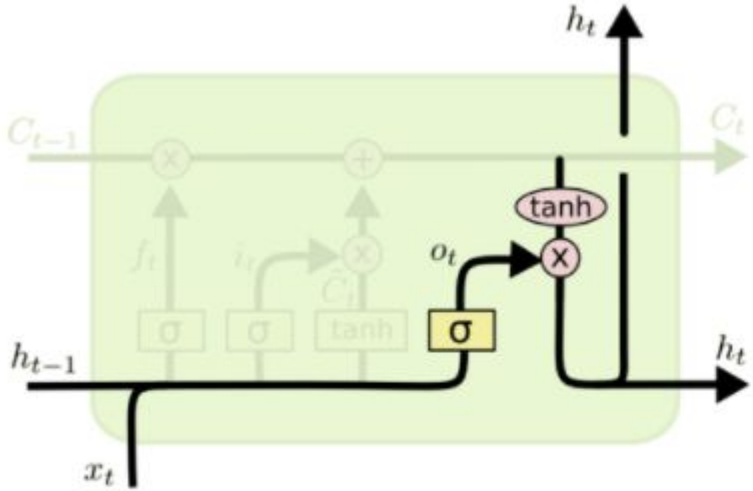
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$



$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

В чому перевага LSTM?

ɔже запам'ятати попередні стани

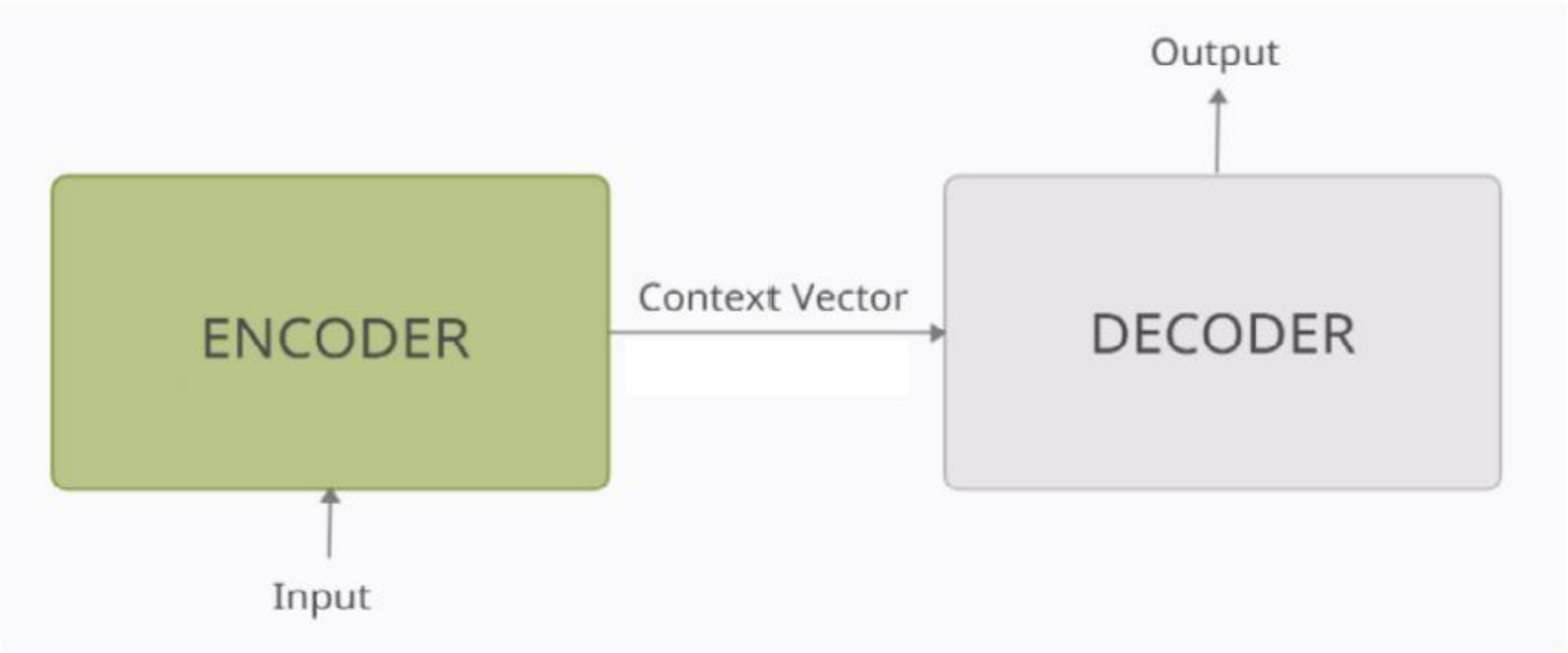
Може забути попередні стани

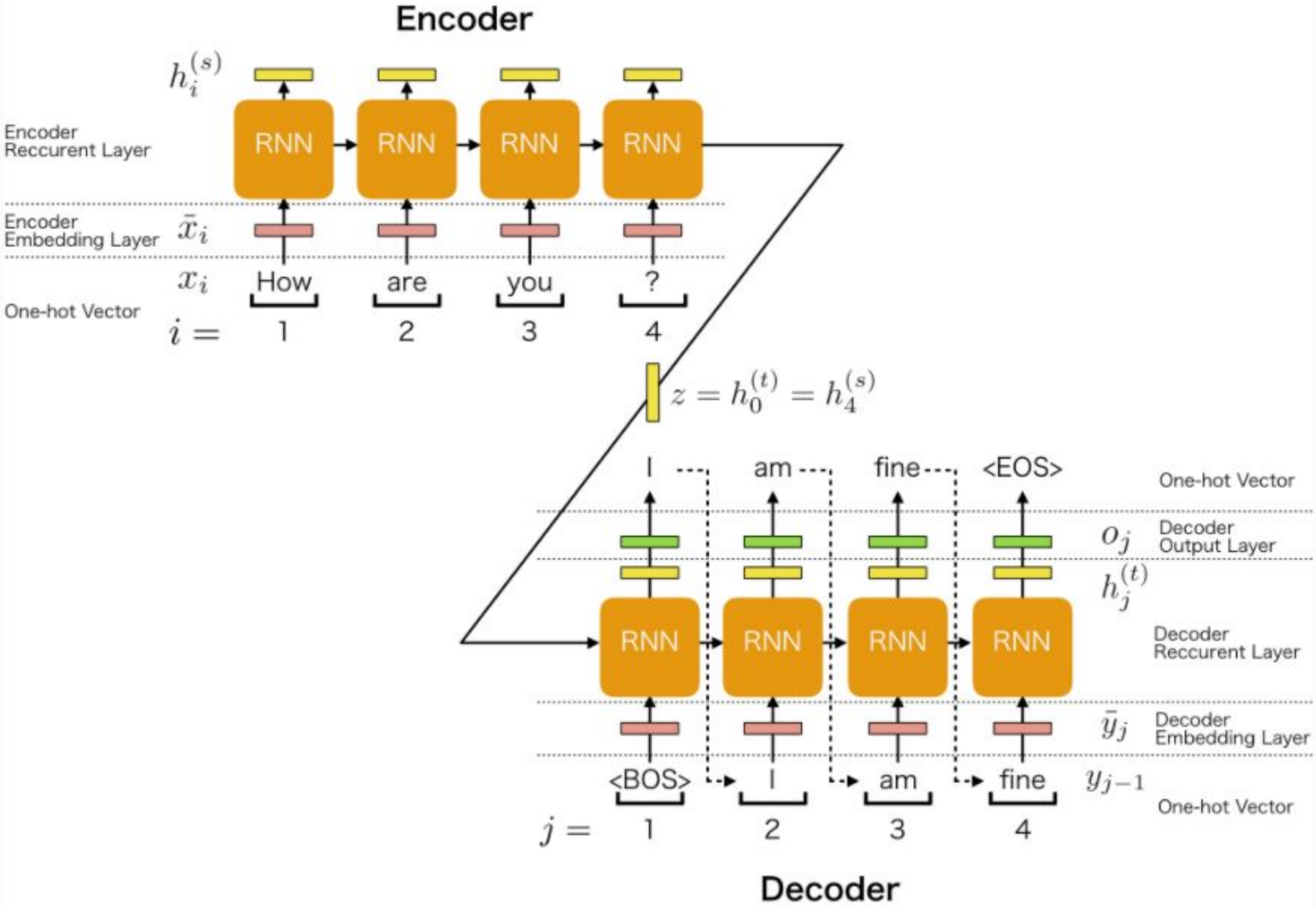
Може визначити нскільки треба запам'ятати попередні стани

Визначає на скільки ам'ятати/забути попередні стани

Все разом

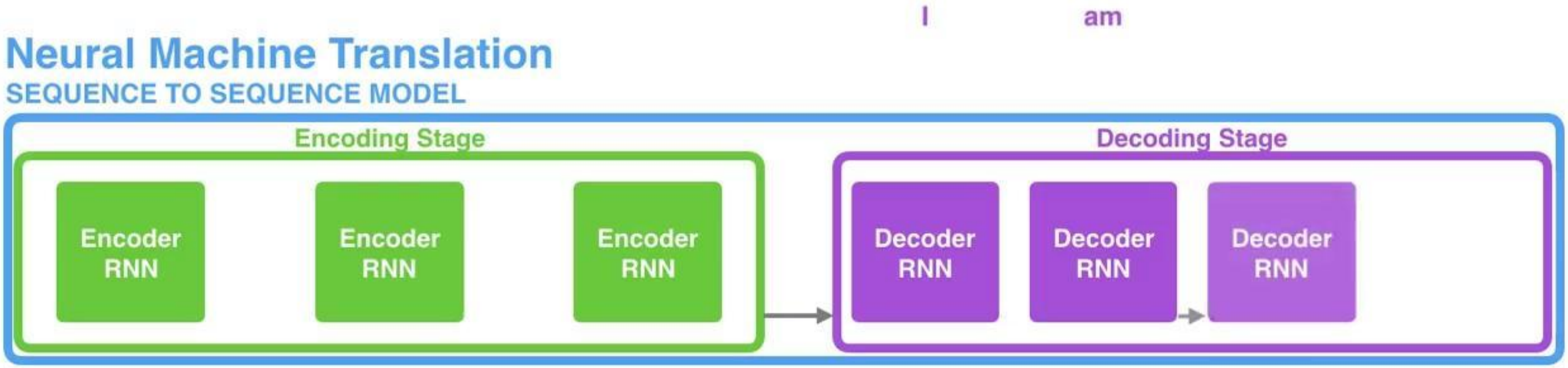
Seq2Seq models



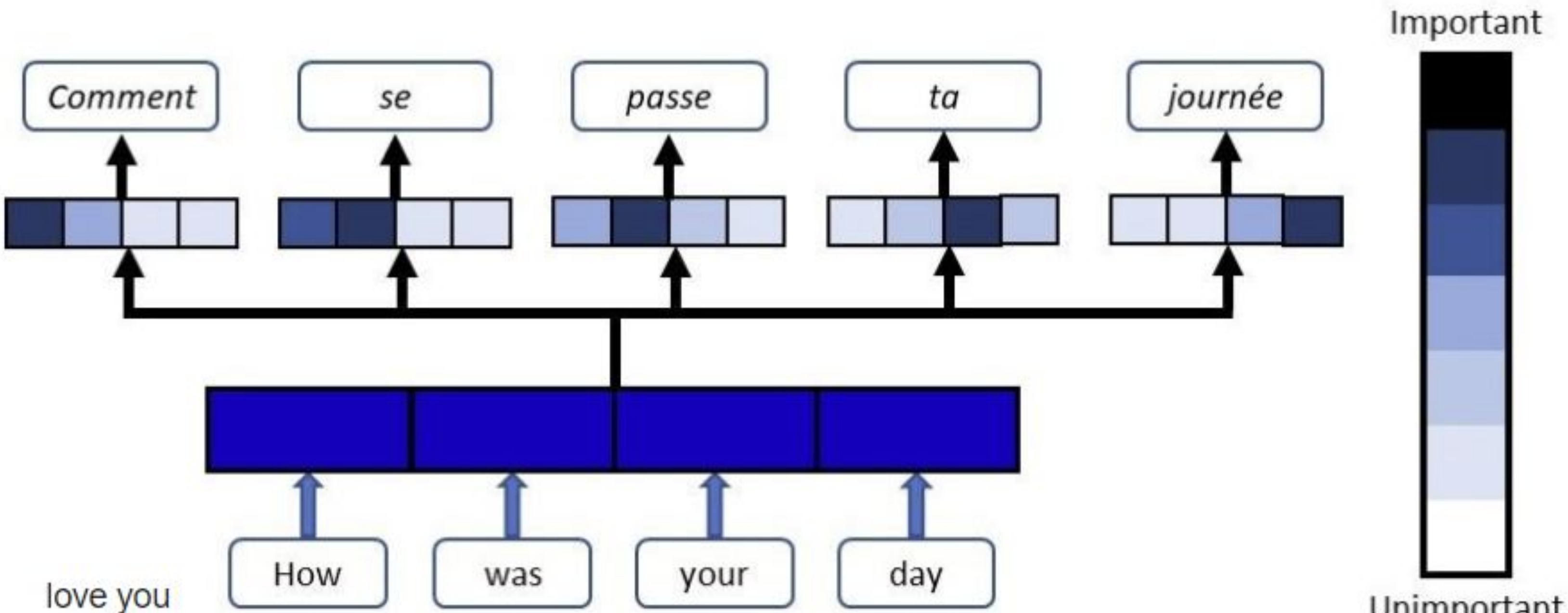

Encoder
$h_i^{(s)}$
Encoder
Reccurent Layer
RNN
RNN
RNN
RNN
Encoder
Embedding Layer
$\bar{x}_i$
x_i
How
are
you
?
One-hot Vector
$i =$
1
2
3
4
$z = h_0^{(t)} = h_4^{(s)}$
I
am
fine
<EOS>
One-hot Vector
o_j
Decoder
Output Layer
$h_j^{(t)}$
RNN
RNN
RNN
RNN
Decoder
Reccurent Layer
$\bar{y}_j$
Decoder
Embedding Layer
<BOS>
I
am
fine
y_{j-1}
One-hot Vector
$j =$
1
2
3
4
Decoder

Neural Machine Translation

SEQUENCE TO SEQUENCE MODEL



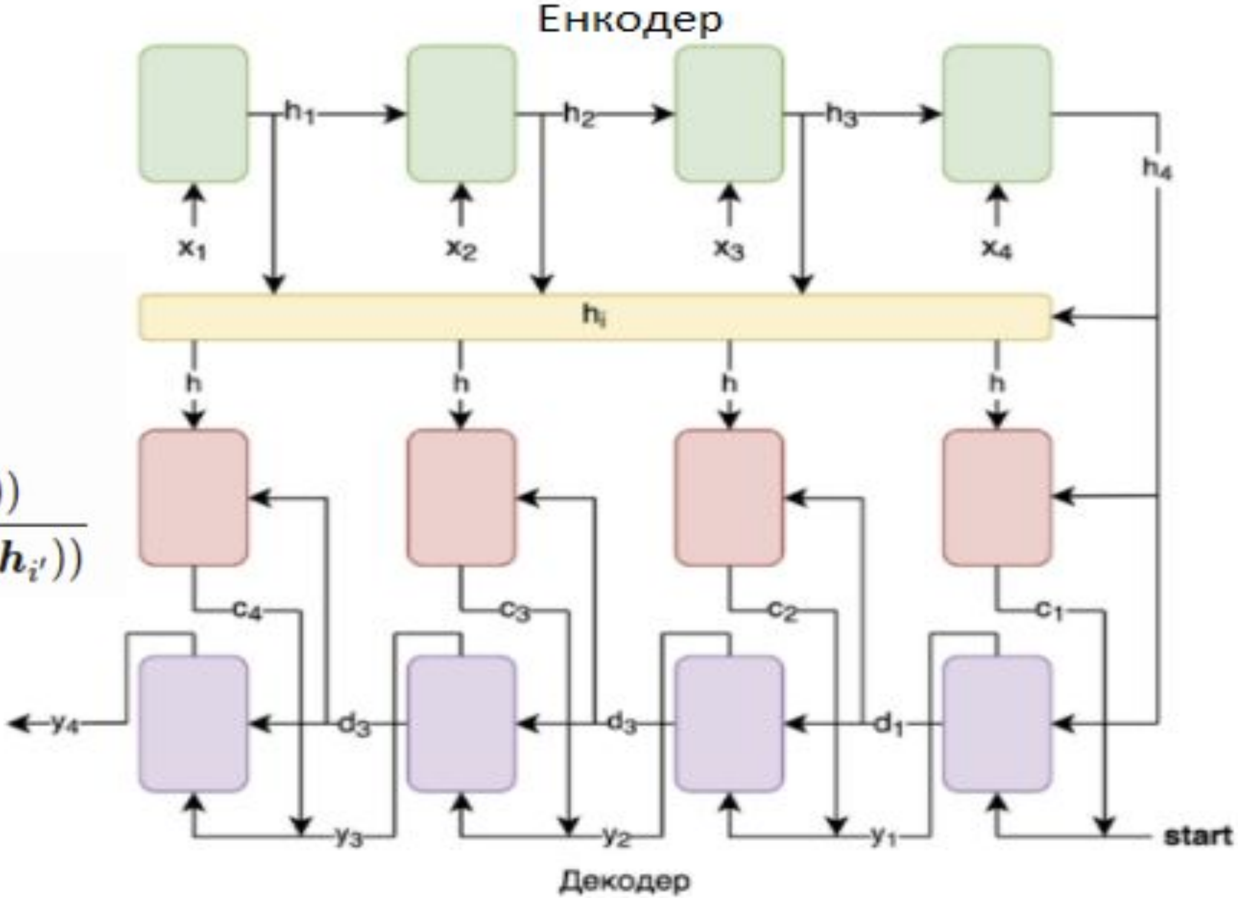
Attention mechanism



	I	love	you
Я	.94	.02	.04
тебе	.11	.01	.88
люблю	.03	.95	.02

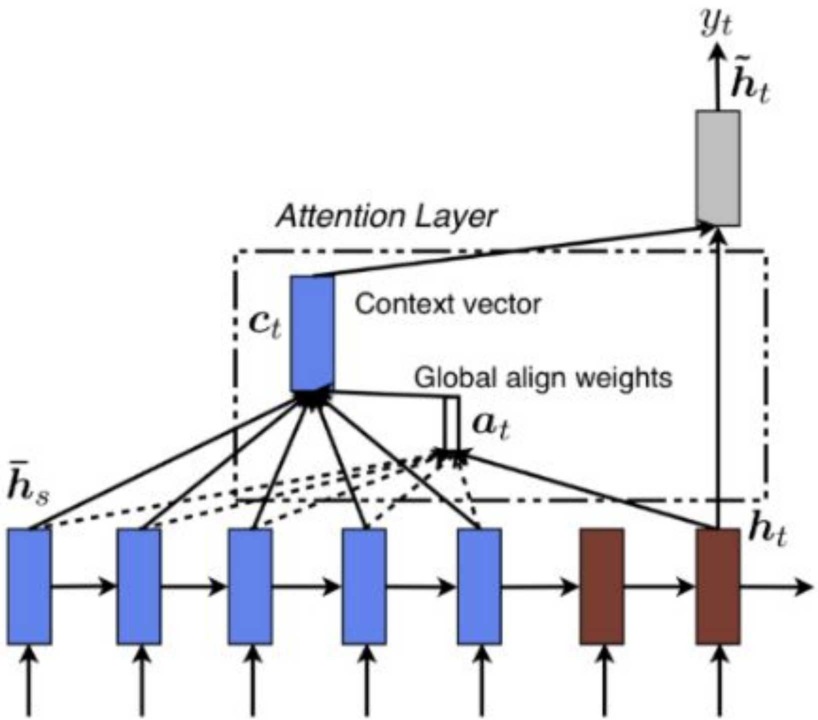
$$\mathbf{c}_t = \sum_{i=1}^{n} \alpha_{t,i} \boldsymbol{h}_i$$

$$\alpha_{t,i} = \text{align}(y_t, x_i) = \frac{\exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_i))}{\sum_{i'=1}^{n} \exp(\text{score}(\boldsymbol{s}_{t-1}, \boldsymbol{h}_{i'}))}$$

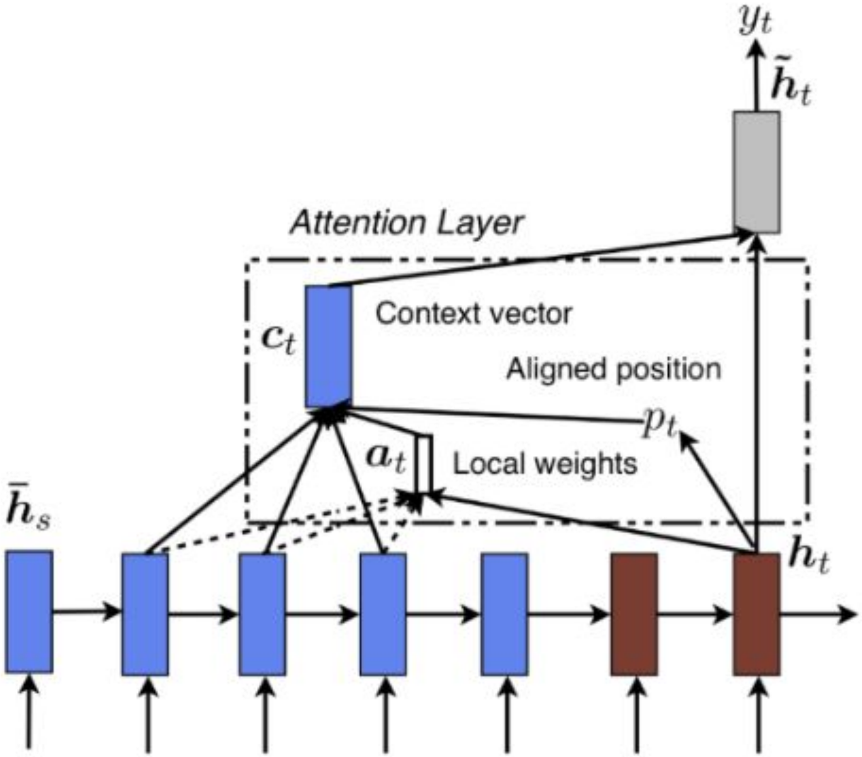


Content-base attention	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \text{cosine}[\boldsymbol{s}_t, \boldsymbol{h}_i]$
Additive(*)	$\text{score}(\boldsymbol{s}_t, \boldsymbol{h}_i) = \mathbf{v}_a^\top \tanh(\mathbf{W}_a[\boldsymbol{s}_t; \boldsymbol{h}_i])$
Location-Base	$\alpha_{t,i} = \text{softmax}(\mathbf{W}_a \boldsymbol{s}_t)$ Note: This simplifies the softmax alignment to only depend on the target position.

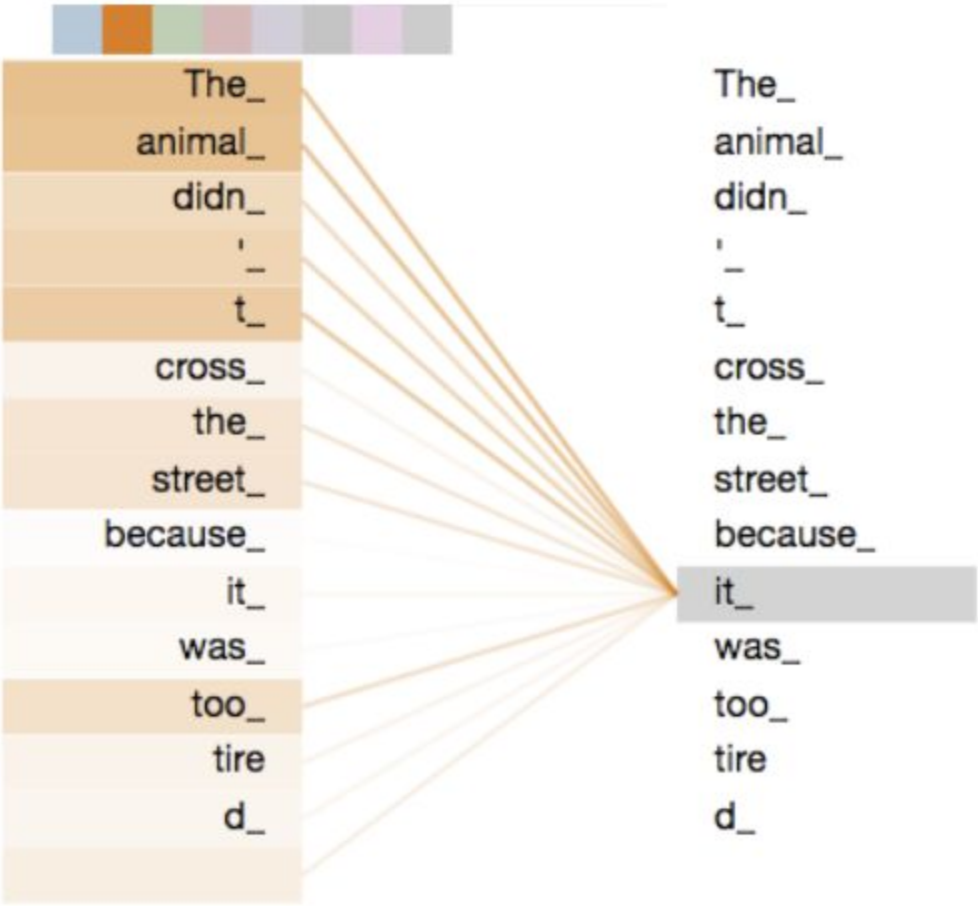
Self-Attention(&)	Relating different positions of the same input sequence. Theoretically the self-attention can adopt any score functions above, but just replace the target sequence with the same input sequence.
Global/Soft	Attending to the entire input state space.
Local/Hard	Attending to the part of input state space; i.e. a patch of the input image.



Global Attention Model



Local Attention Model


The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_
The_
animal_
didn_
'_
t_
cross_
the_
street_
because_
it_
was_
too_
tire
d_

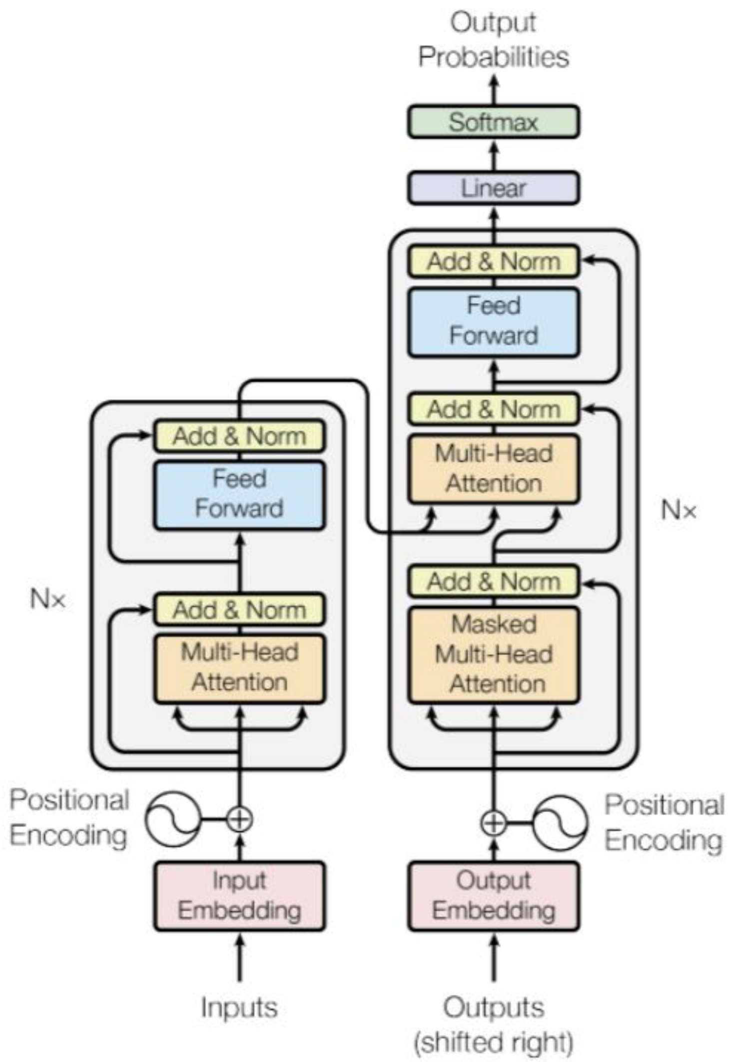
Що дає механізм уваги?

Transformers – Attention is All You Need

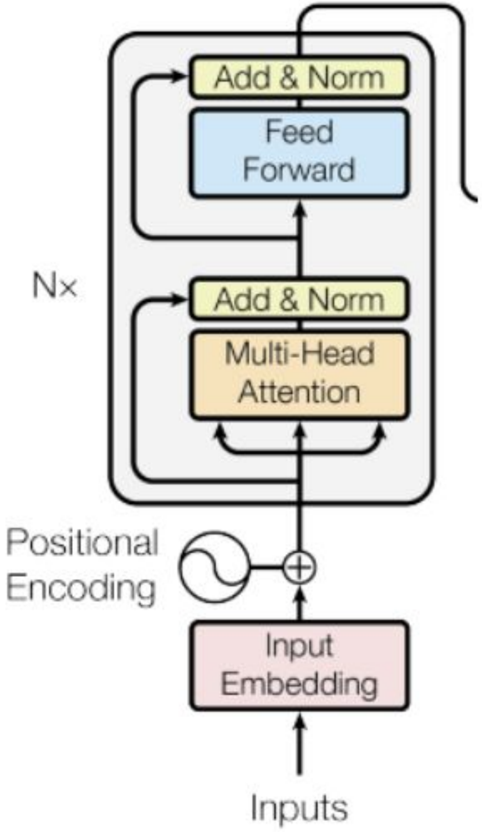
$$c = \sum_j \alpha_j h_j$$

$$e_{ij} = a(s_i, h_j), \qquad \alpha_{i,j} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

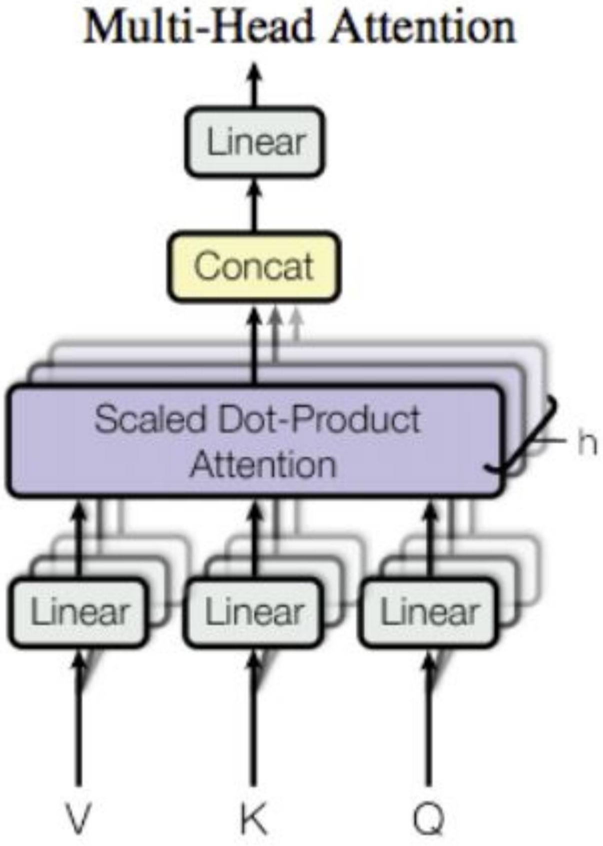
$$e_{ij} = f(s_i) g(h_j)^T$$

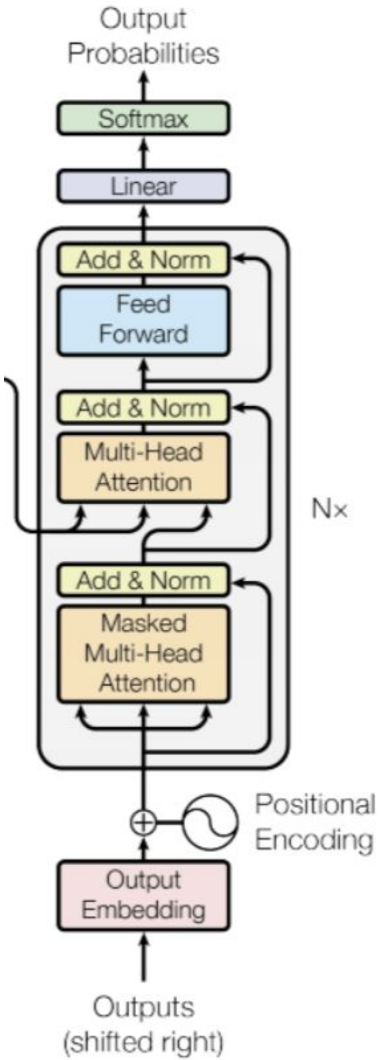


Attention Is All You Need

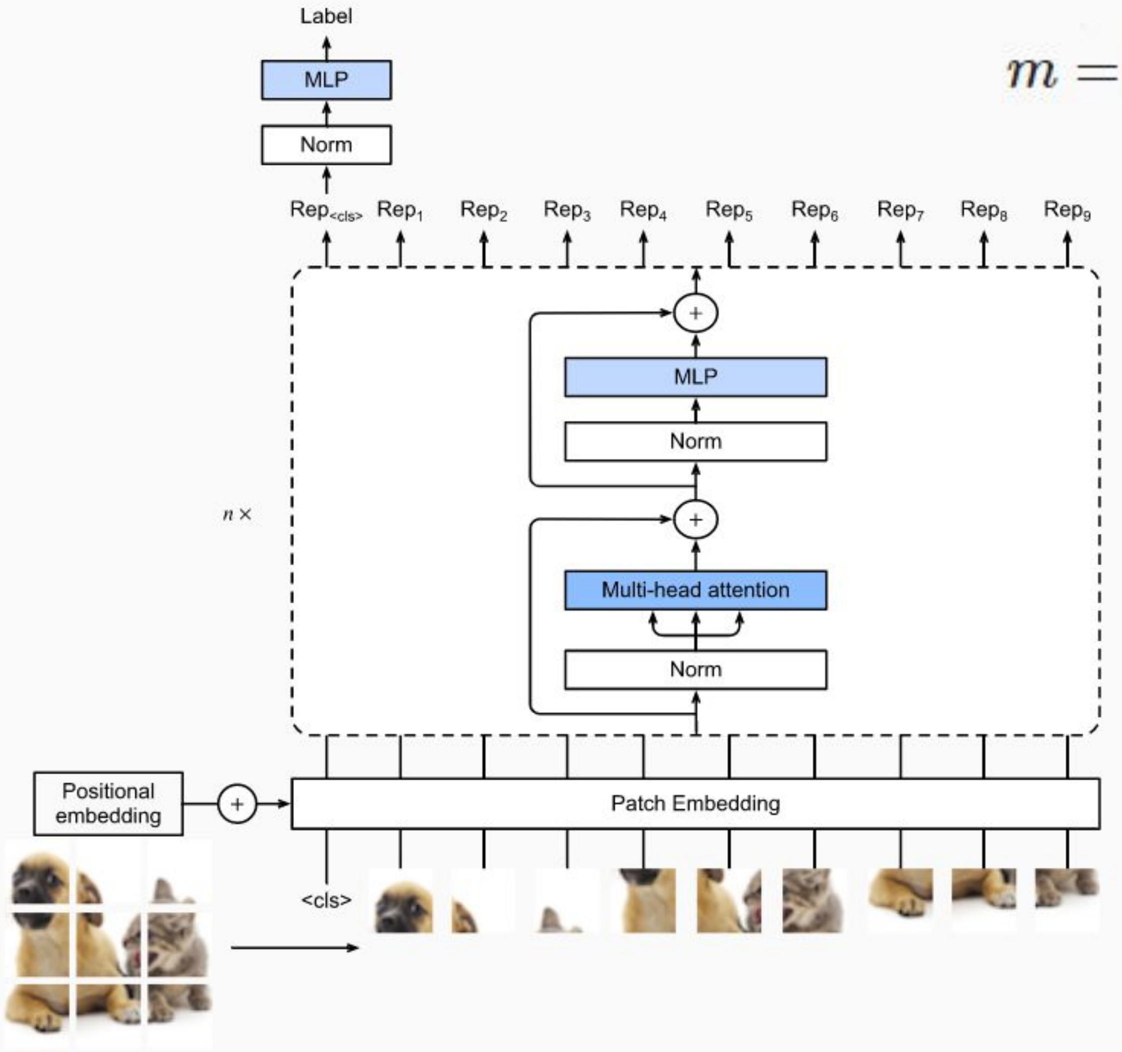


Multi-head attention





Зоровий трансформер



$$m = hw/p^2$$