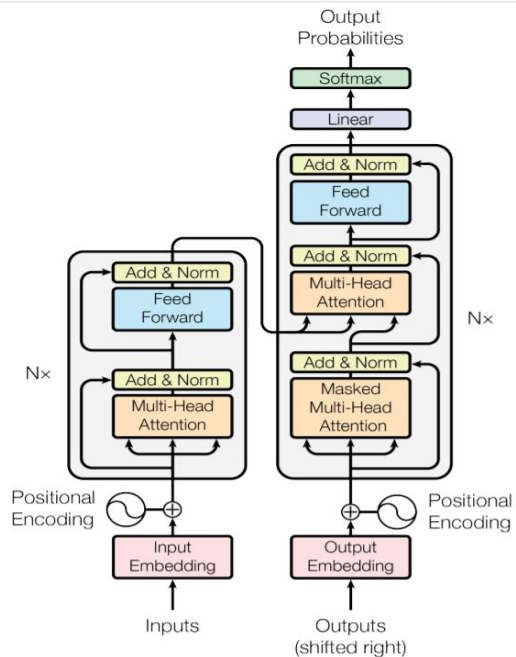


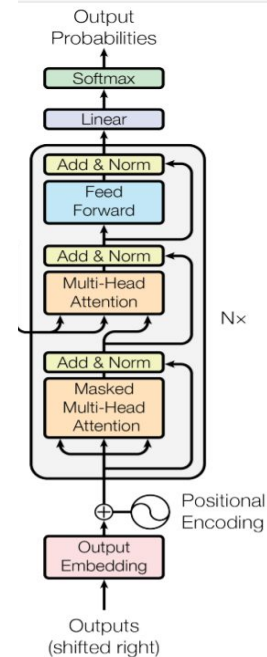
Трансформеры

и их улучшения

Модель Трансформеров

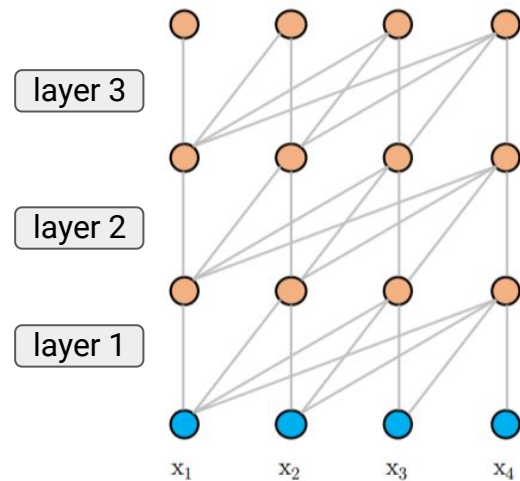
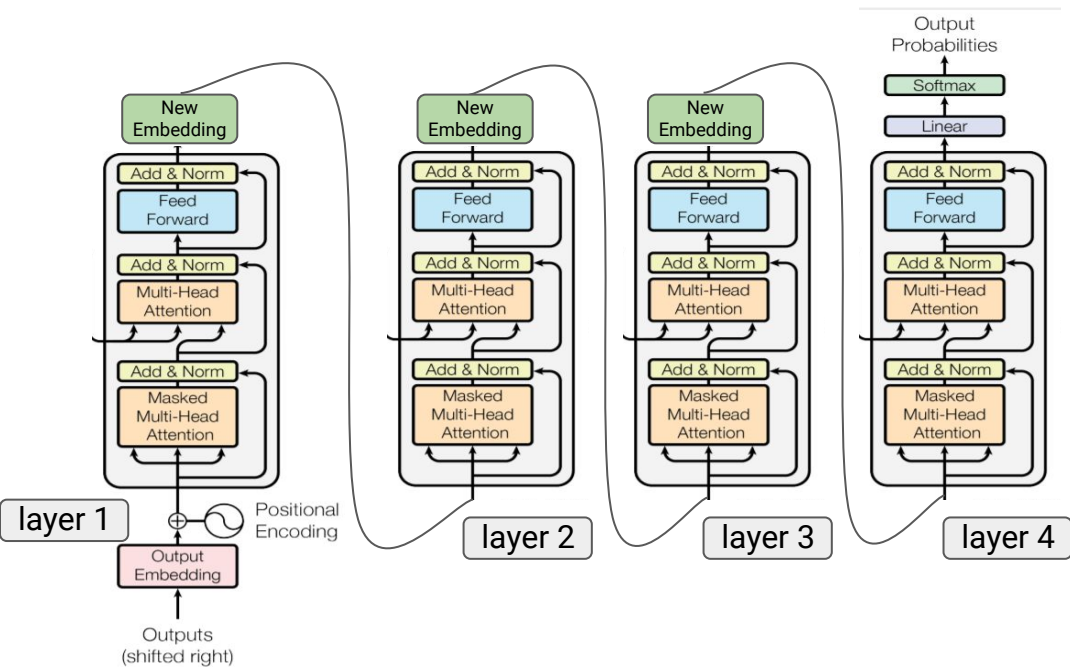


Перевод



Генерация

Модель Трансформеров



Генерация

Self-Attention

Step 1

$$n \begin{matrix} \text{X} \\ \begin{bmatrix} \text{green} & \text{green} & \text{green} & \text{green} \\ \text{green} & \text{green} & \text{green} & \text{green} \end{bmatrix} \end{matrix} \times \begin{matrix} \text{W}^Q \\ \begin{bmatrix} \text{purple} & \text{purple} & \text{purple} & \text{purple} \\ \text{purple} & \text{purple} & \text{purple} & \text{purple} \end{bmatrix} \end{matrix} = \begin{matrix} \text{Q} \\ \begin{bmatrix} \text{purple} & \text{purple} & \text{purple} \\ \text{purple} & \text{purple} & \text{purple} \end{bmatrix} \end{matrix}$$

$$\begin{matrix} \text{X} \\ \begin{bmatrix} \text{green} & \text{green} & \text{green} & \text{green} \\ \text{green} & \text{green} & \text{green} & \text{green} \end{bmatrix} \end{matrix} \times \begin{matrix} \text{W}^K \\ \begin{bmatrix} \text{orange} & \text{orange} & \text{orange} & \text{orange} \\ \text{orange} & \text{orange} & \text{orange} & \text{orange} \end{bmatrix} \end{matrix} = \begin{matrix} \text{K} \\ \begin{bmatrix} \text{orange} & \text{orange} & \text{orange} \\ \text{orange} & \text{orange} & \text{orange} \end{bmatrix} \end{matrix}$$

$$\begin{matrix} \text{X} \\ \begin{bmatrix} \text{green} & \text{green} & \text{green} & \text{green} \\ \text{green} & \text{green} & \text{green} & \text{green} \end{bmatrix} \end{matrix} \times \begin{matrix} \text{W}^V \\ \begin{bmatrix} \text{blue} & \text{blue} & \text{blue} & \text{blue} \\ \text{blue} & \text{blue} & \text{blue} & \text{blue} \end{bmatrix} \end{matrix} = \begin{matrix} \text{V} \\ \begin{bmatrix} \text{blue} & \text{blue} & \text{blue} \\ \text{blue} & \text{blue} & \text{blue} \end{bmatrix} \end{matrix}$$

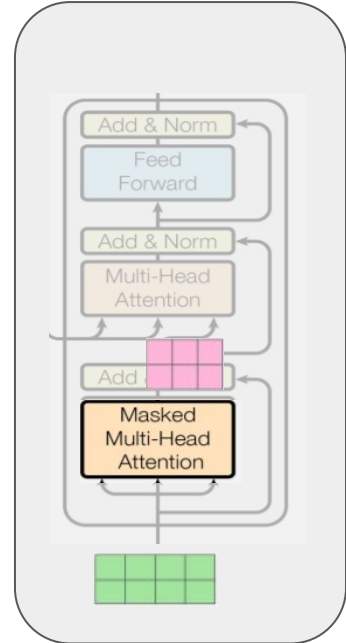
$\mathcal{O}(n^2)$

Step 2

$$\text{softmax} \left(\frac{\begin{matrix} \text{Q} \\ \begin{bmatrix} \text{purple} & \text{purple} & \text{purple} \\ \text{purple} & \text{purple} & \text{purple} \end{bmatrix} \end{matrix} \times \begin{matrix} \text{K}^T \\ \begin{bmatrix} \text{orange} & \text{orange} & \text{orange} \\ \text{orange} & \text{orange} & \text{orange} \end{bmatrix} \end{matrix}}{\sqrt{d_k}} \right) \begin{matrix} \text{V} \\ \begin{bmatrix} \text{blue} & \text{blue} & \text{blue} \\ \text{blue} & \text{blue} & \text{blue} \end{bmatrix} \end{matrix}$$

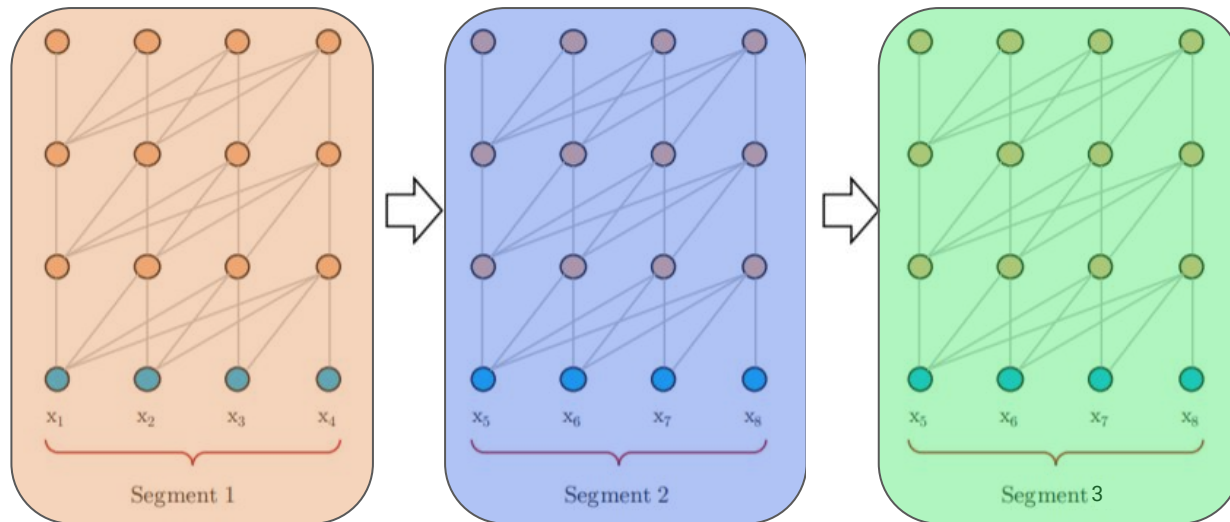
$$= \begin{matrix} \text{Z} \\ \begin{bmatrix} \text{pink} & \text{pink} & \text{pink} \\ \text{pink} & \text{pink} & \text{pink} \end{bmatrix} \end{matrix}$$

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



Процесс обучения

While we found the idea presented in the previous subsection very appealing

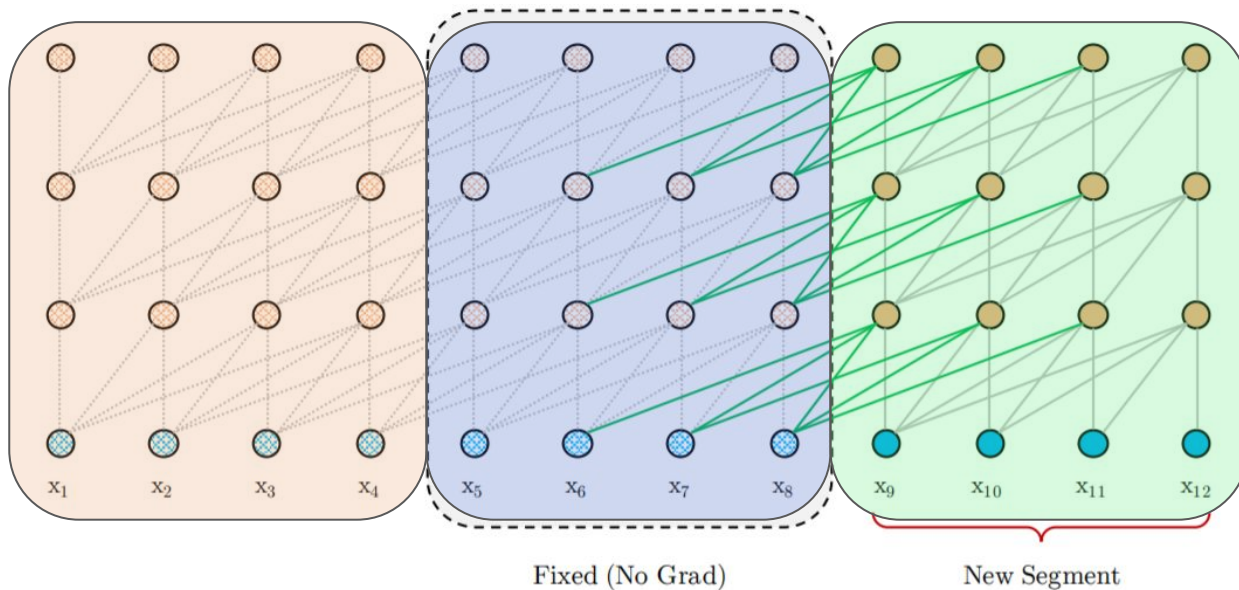


$$n = 12$$

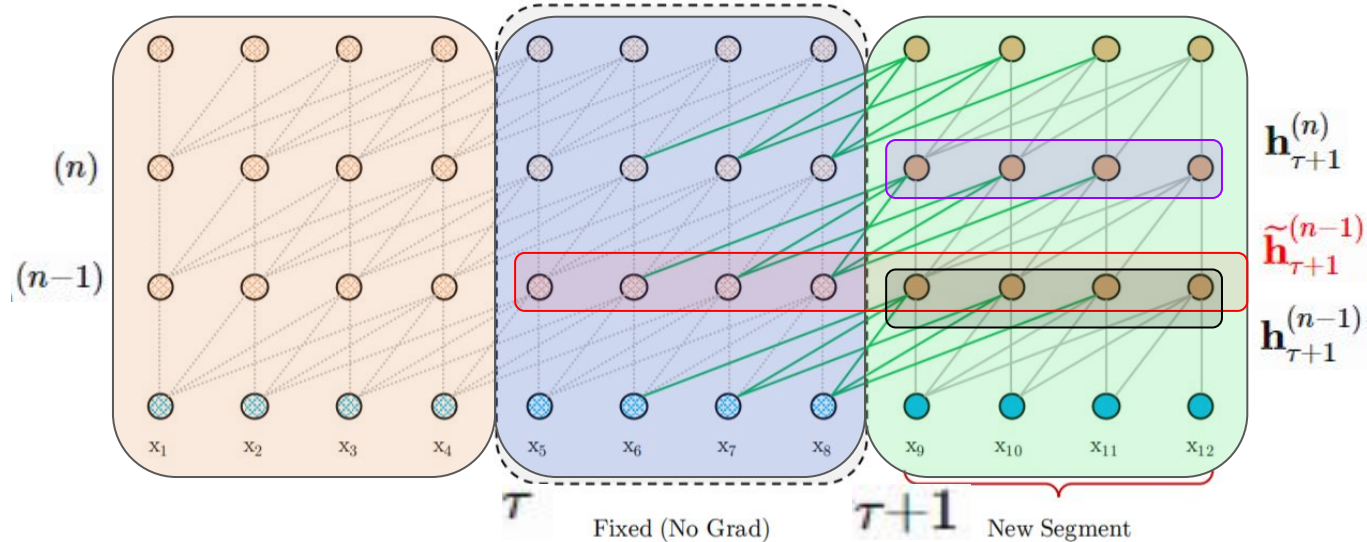
(a) Train phase.

Процесс обучения

While we found the idea presented in the previous subsection very appealing



Transformer-XL



$$\tilde{\mathbf{h}}_{\tau+1}^{(n-1)} = [\text{stop-gradient}(\mathbf{h}_{\tau}^{(n-1)}) \circ \mathbf{h}_{\tau+1}^{(n-1)}]$$

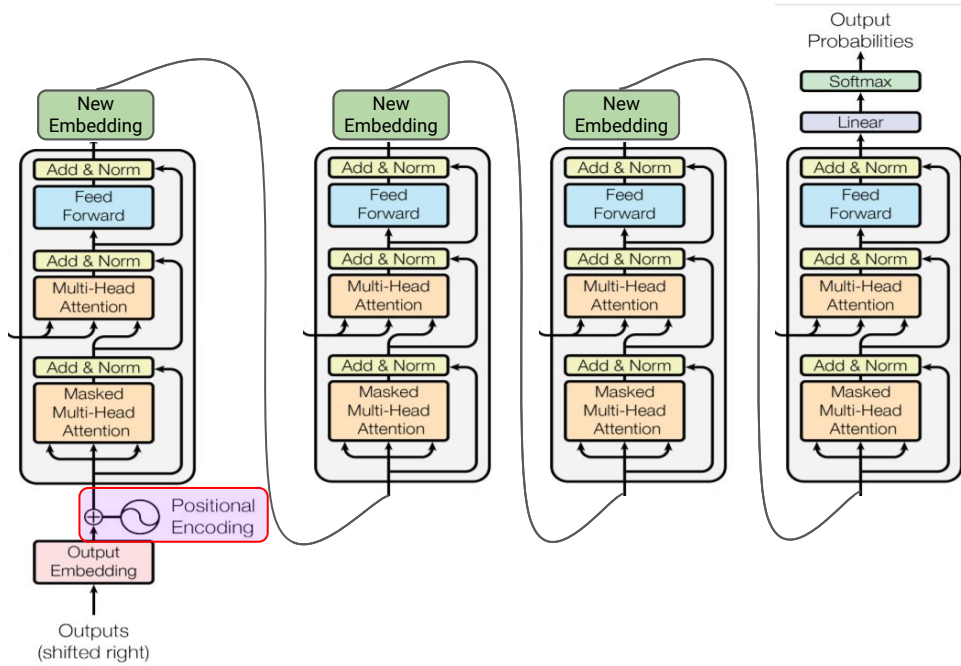
$$\mathbf{Q}_{\tau+1}^{(n)} = \mathbf{h}_{\tau+1}^{(n-1)} \mathbf{W}^q$$

$$\mathbf{K}_{\tau+1}^{(n)} = \tilde{\mathbf{h}}_{\tau+1}^{(n-1)} \mathbf{W}^k$$

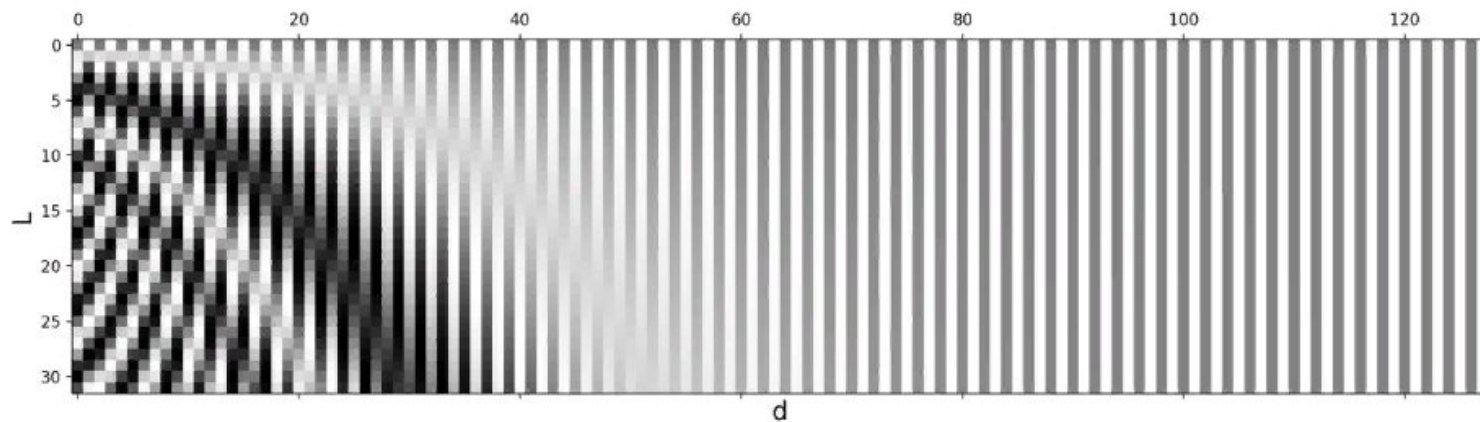
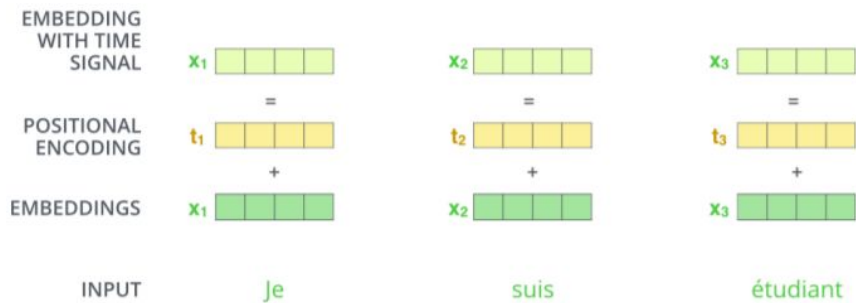
$$\mathbf{V}_{\tau+1}^{(n)} = \tilde{\mathbf{h}}_{\tau+1}^{(n-1)} \mathbf{W}^v$$

$$\mathbf{h}_{\tau+1}^{(n)} = \text{transformer-layer}(\mathbf{Q}_{\tau+1}^{(n)}, \mathbf{K}_{\tau+1}^{(n)}, \mathbf{V}_{\tau+1}^{(n)})$$

Positional Encoding

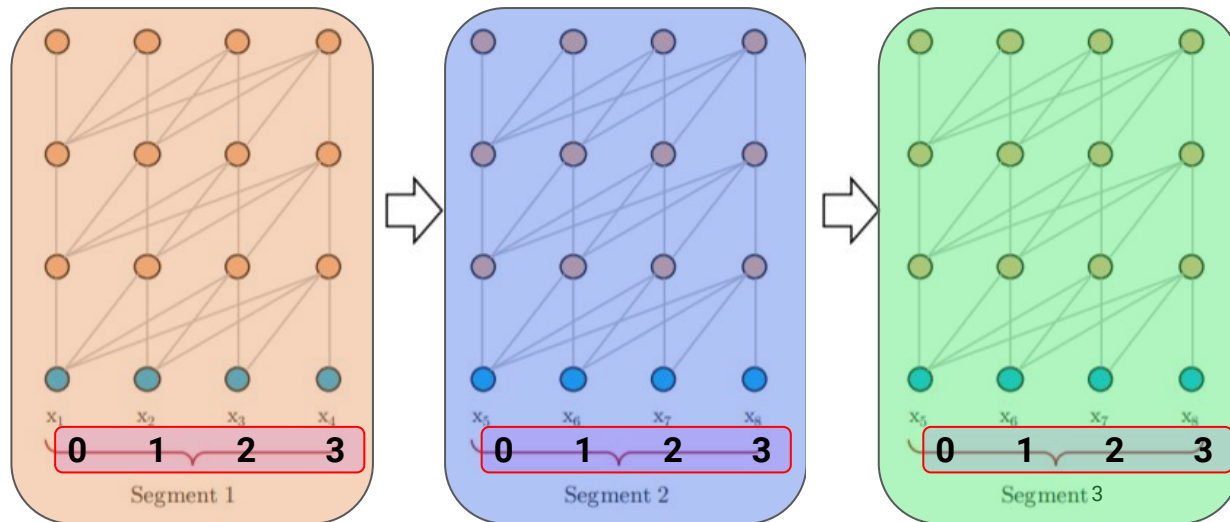


Positional Encoding



Positional Encoding

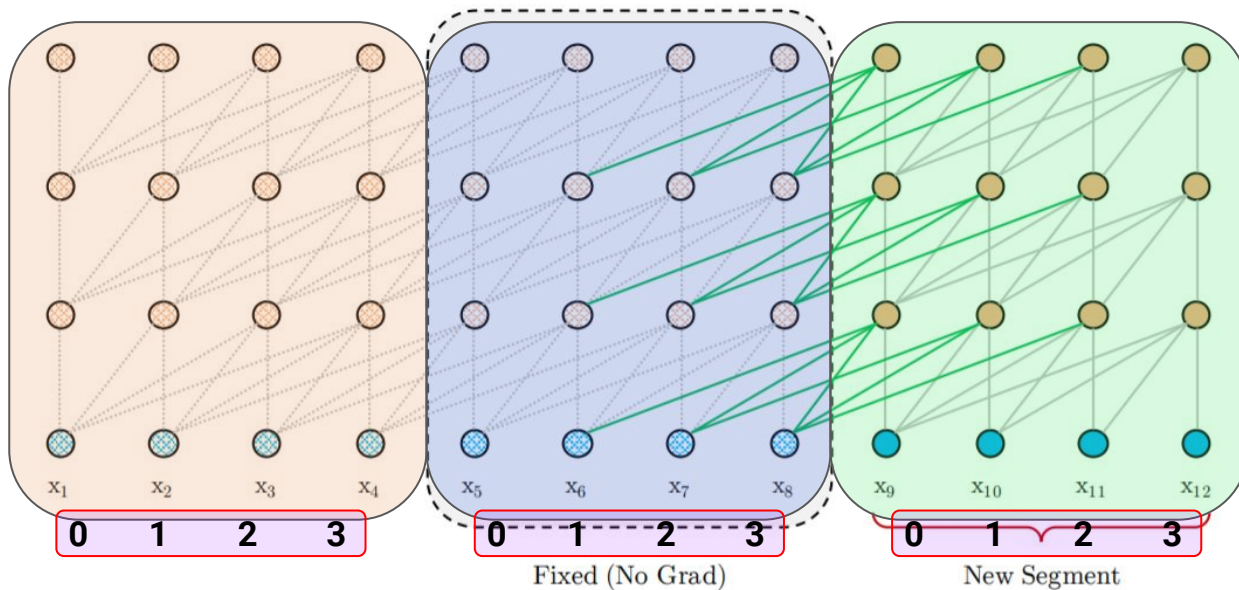
While we found the idea presented in the previous subsection very appealing



(a) Train phase.

Positional Encoding

While we found the idea presented in the previous subsection very appealing



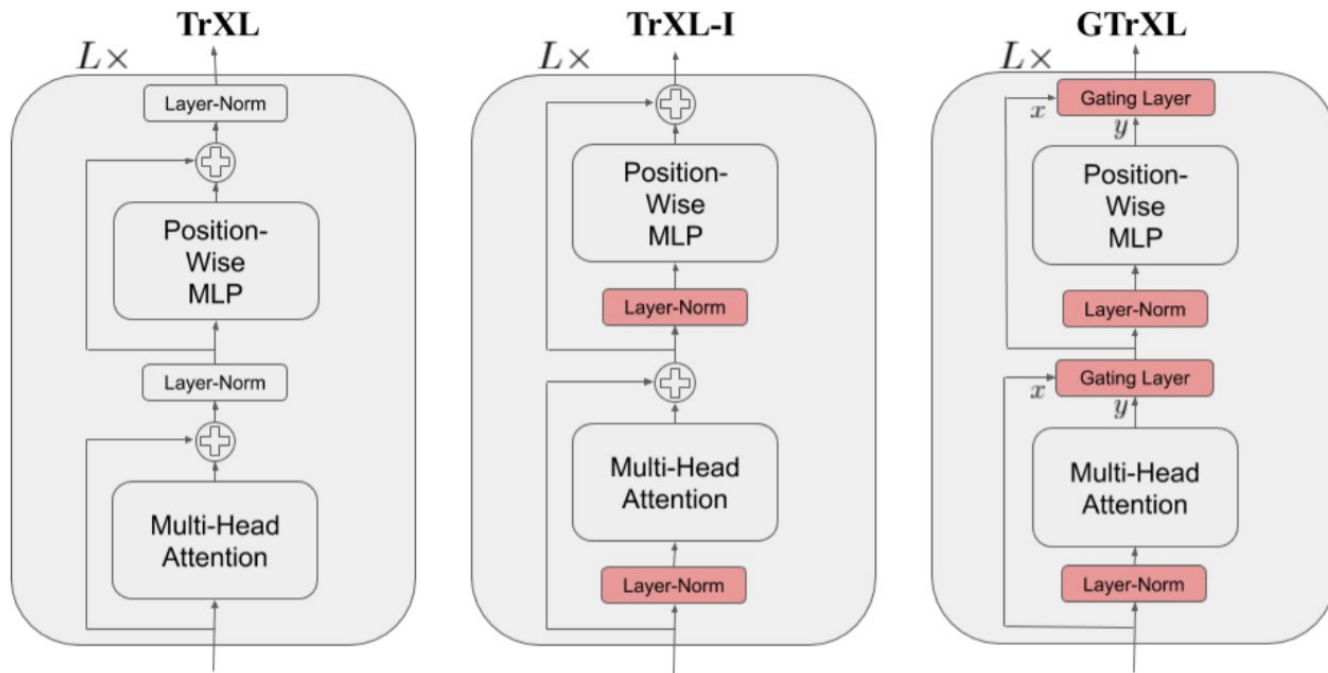
Positional Encoding

$$\begin{aligned} a_{ij} &= \mathbf{q}_i \mathbf{k}_j^\top = (\mathbf{x}_i + \mathbf{p}_i) \mathbf{W}^q ((\mathbf{x}_j + \mathbf{p}_j) \mathbf{W}^k)^\top \\ &= \mathbf{x}_i \mathbf{W}^q \mathbf{W}^{k\top} \mathbf{x}_j^\top + \mathbf{x}_i \mathbf{W}^q \mathbf{W}^{k\top} \mathbf{p}_j^\top + \mathbf{p}_i \mathbf{W}^q \mathbf{W}^{k\top} \mathbf{x}_j^\top + \mathbf{p}_i \mathbf{W}^q \mathbf{W}^{k\top} \mathbf{p}_j^\top \end{aligned}$$

Transformer-XL reparameterizes the above four terms as follows:

$$a_{ij}^{\text{rel}} = \underbrace{\mathbf{x}_i \mathbf{W}^q \mathbf{W}_E^{k\top} \mathbf{x}_j^\top}_{\text{content-based addressing}} + \underbrace{\mathbf{x}_i \mathbf{W}^q \mathbf{W}_R^{k\top} \mathbf{r}_{i-j}^\top}_{\text{content-dependent positional bias}} + \underbrace{\mathbf{u} \mathbf{W}_E^{k\top} \mathbf{x}_j^\top}_{\text{global content bias}} + \underbrace{\mathbf{v} \mathbf{W}_R^{k\top} \mathbf{r}_{i-j}^\top}_{\text{global positional bias}}$$

Stabilization for RL (GTrXL)



MultiHead Attention

1) This is our input sentence*

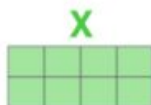
2) We embed each word*

3) Split into 8 heads. We multiply X or R with weight matrices

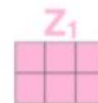
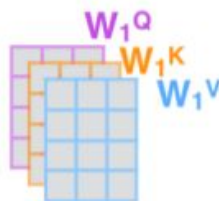
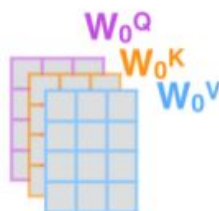
4) Calculate attention using the resulting $Q/K/V$ matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

Thinking
Machines



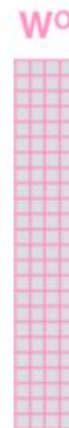
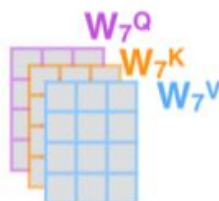
* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one



...

...

...



Adaptive Attention Span

$$m_z(x) = \text{clamp}\left(\frac{1}{R}(R + z - x), 0, 1\right)$$

where R is a hyper-parameter which defines the softness of m_z .

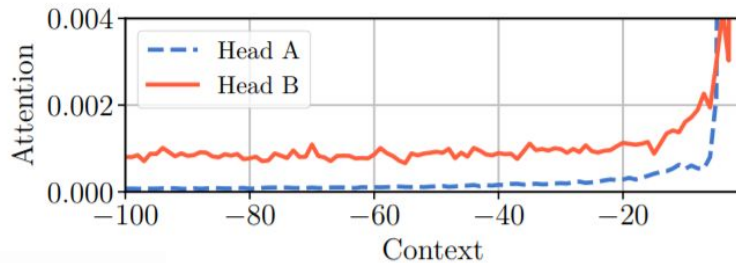
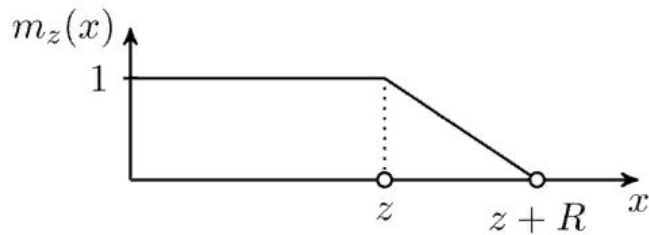
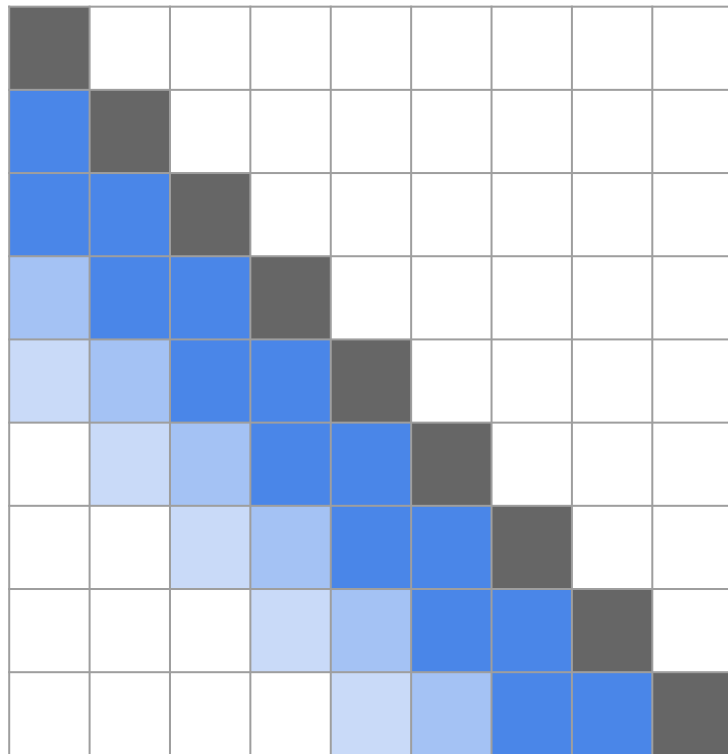
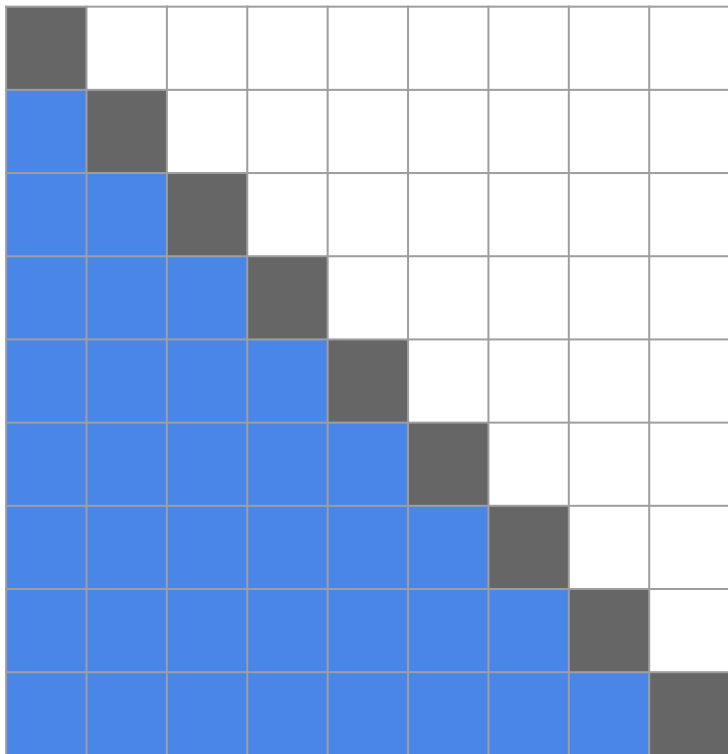


Fig. 8. The soft masking function used in the adaptive attention span. (Image source: [Sukhbaatar, et al. 2019.](#))

The soft mask function is applied to the softmax elements in the attention weights:

$$a_{ij} = \frac{m_z(i - j) \exp(s_{ij})}{\sum_{r=i-s}^{i-1} m_z(i - r) \exp(s_{ir})}$$

Adaptive Attention Span



Результаты

Model	#layers	Avg. span	#Params	#FLOPS	dev	test
<i>Small models</i>						
T12 (Al-Rfou et al., 2019)	12	512	44M	22G	-	1.18
Adaptive-Span ($S = 8192$)	12	314	38M	42M	1.05	1.11
<i>Large models</i>						
T64 (Al-Rfou et al., 2019)	64	512	235M	120G	1.06	1.13
T-XL (Dai et al., 2019)	24	3800	277M	438M	-	1.08
Adaptive-Span ($S = 8192$)	24	245	209M	179M	1.01	1.07

Литература

- <https://lilianweng.github.io/lil-log/2020/04/07/the-transformer-family.html>
- <https://arxiv.org/abs/1706.03762>
- <https://arxiv.org/abs/1901.02860>
- <https://arxiv.org/abs/1910.06764>
- <https://arxiv.org/abs/1905.07799>

