Transformer-XL

Attentive Language Model beyond a Fixed-Length Context

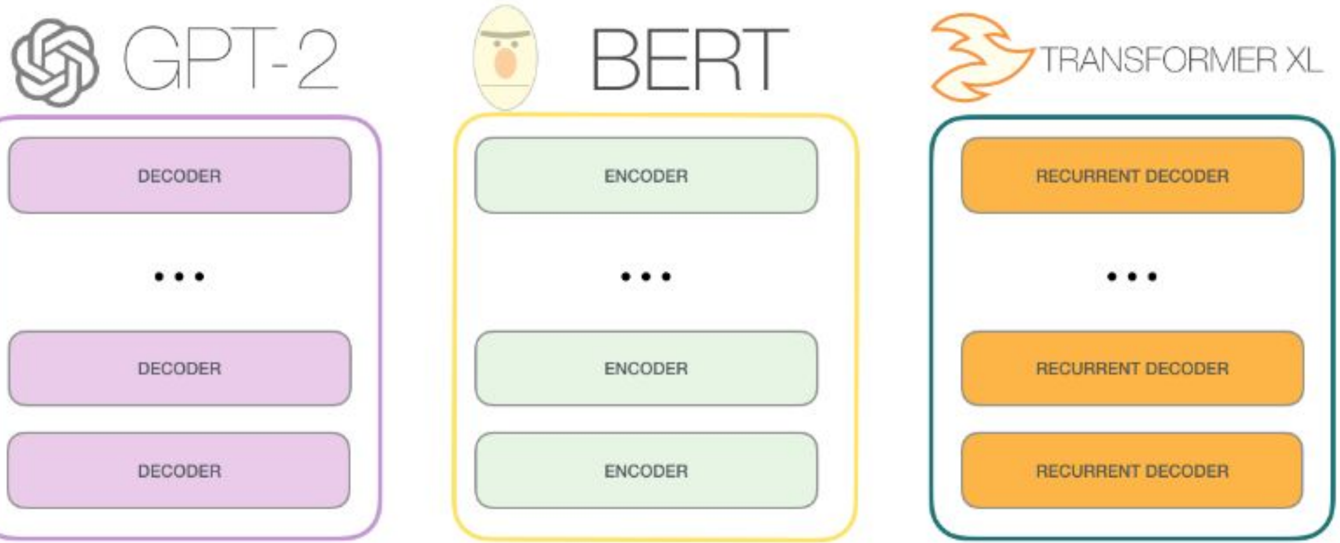
Language Modeling

Given a text sequence X, estimate the probability distribution P(x)

$$p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid \mathbf{x}_{<t})$$

Similar LM Models

Not Seq2Seq unlike the original Transformer



Predecessor & Successor

Deep transformer model with fixed context (AR)

Character-Level Language Modeling with Deeper Self-Attention (Al-Rfou et al.)

Transformer XL (AR)

Tranformer-XL: Attentive Language Models Beyond a Fixed-Length Context (Dai et al.)

XLNet (AR+AE)

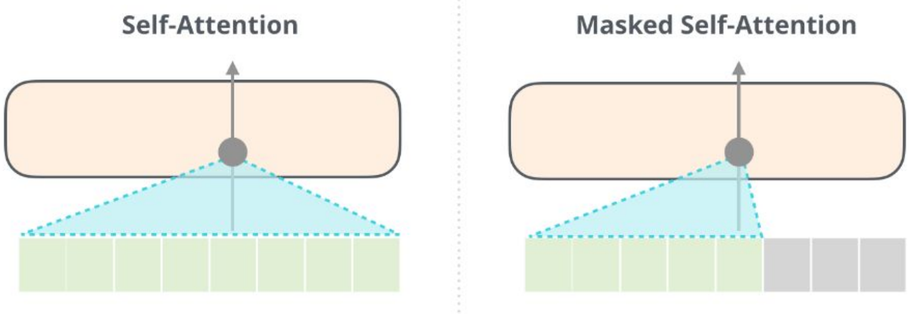
XLNet: Generalized Autoregressive Pretraining for Language Understanding (Yang et al.)

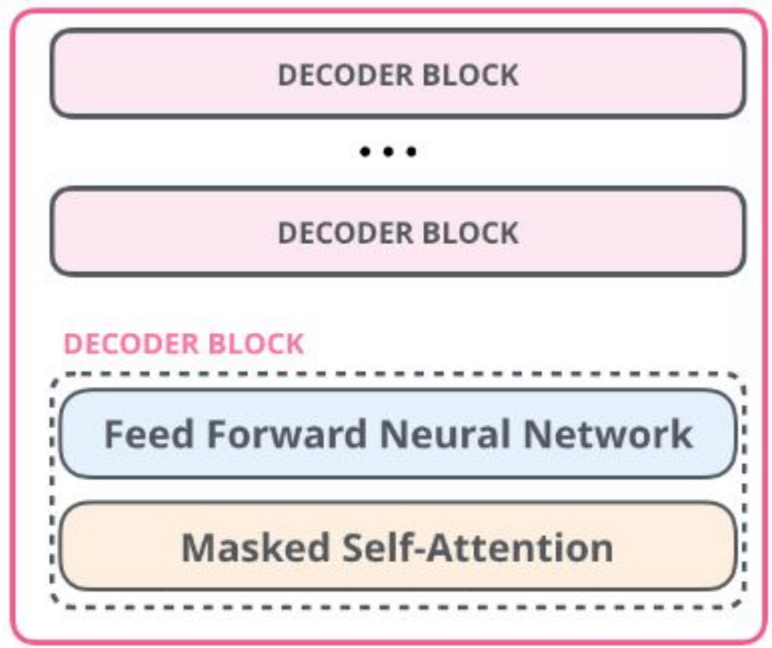
Vanilla Transformer LM

64 transformer layers (235 mil params)

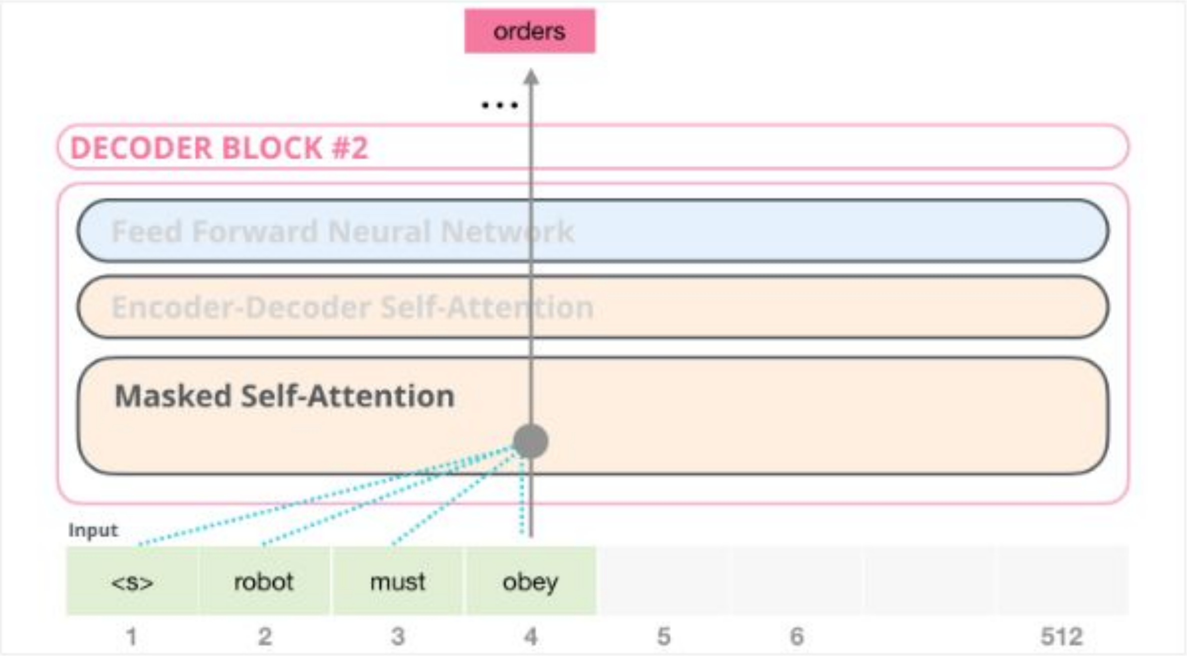
Fixed-context of 512 characters

Vanilla Transformer LM

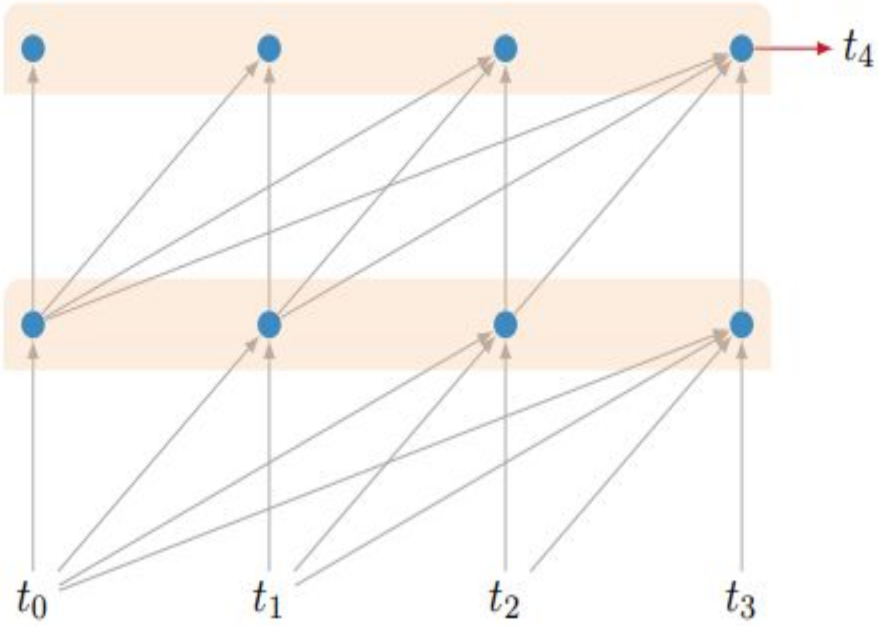




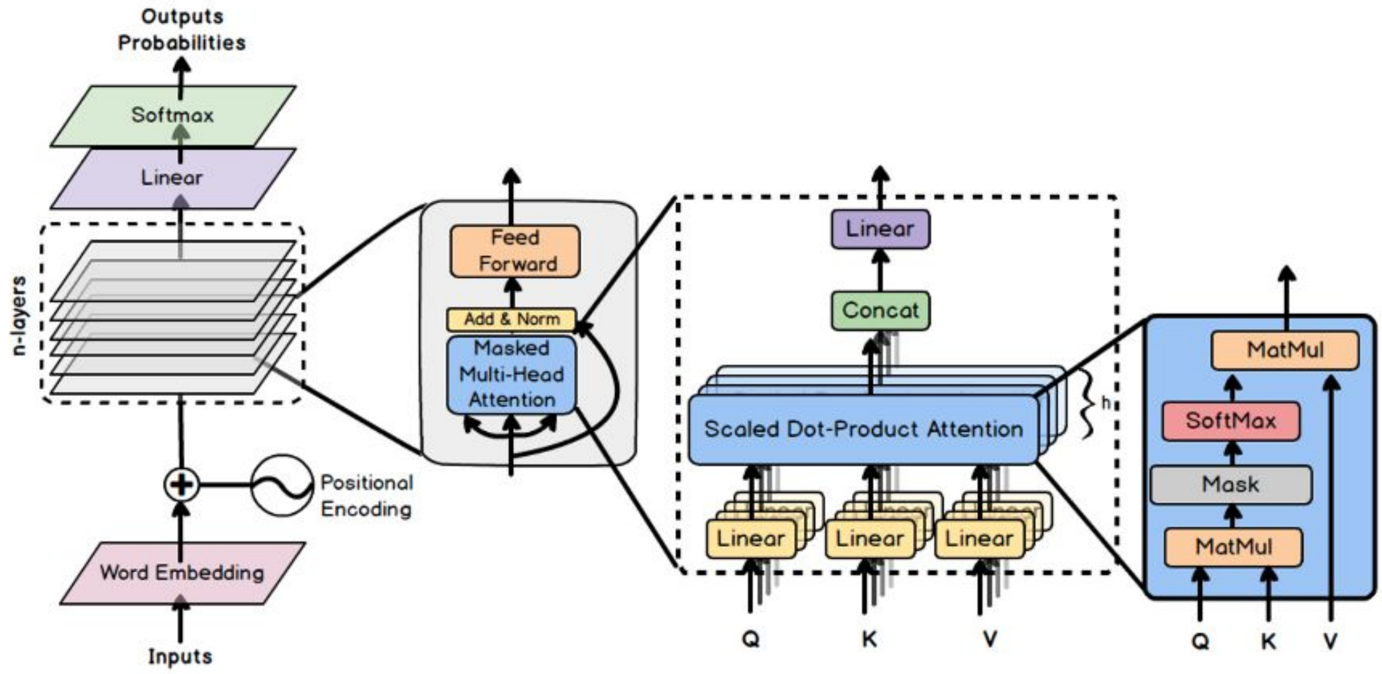
Vanilla Transformer LM



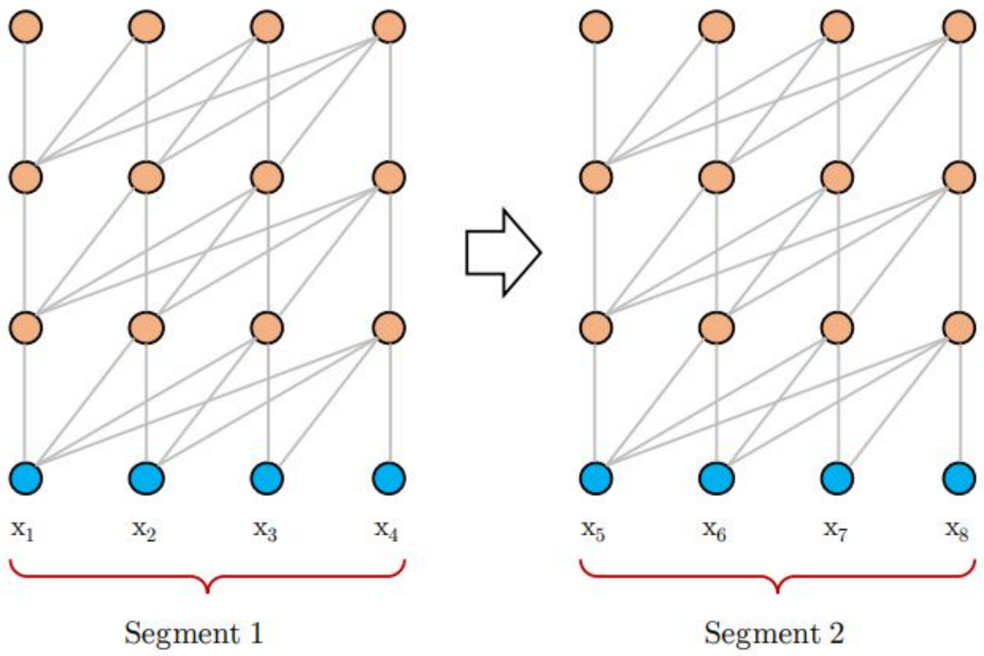
Vanilla Transformer LM



Vanilla Transformer LM



Vanilla Transformer LM: Training Phase



Vanilla Transformer LM: Training Phase

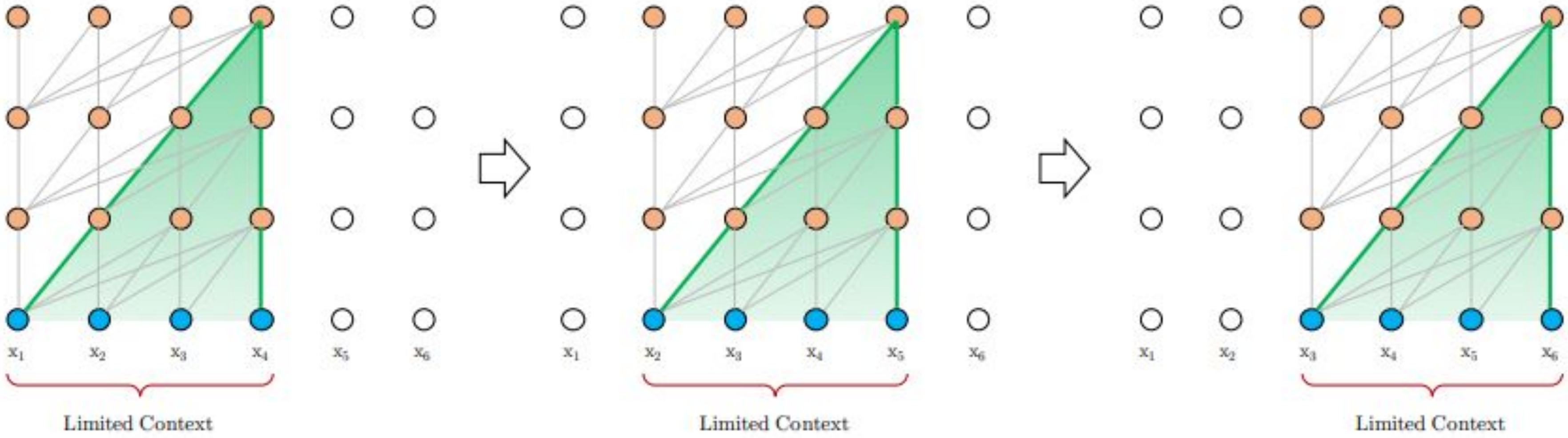
Short dependency length

- Upper-bounded by the segment length (a few hundred chars)

Context Fragmentation

- Non-semantic chunking
- Lacks necessary contextual info to predict the first few symbols

Vanilla Transformer LM: Evaluation Phase



Vanilla Transformer LM: Evaluation Phase

Expensive

- Need to process a new segment from scratch every position

Transformer XL

Segment-Level Recurrence

Relative Positional Encoding

Segment-Level Recurrence

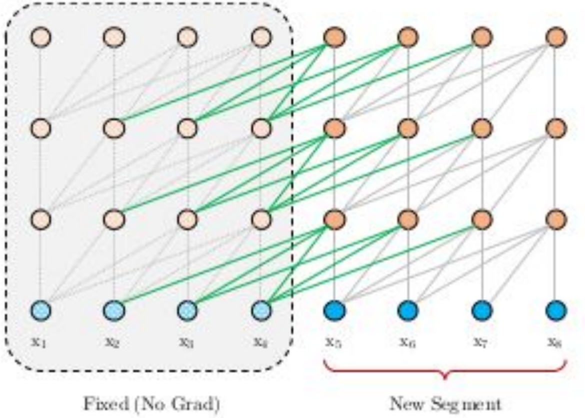
Re-introduce recurrence, but at the segment level

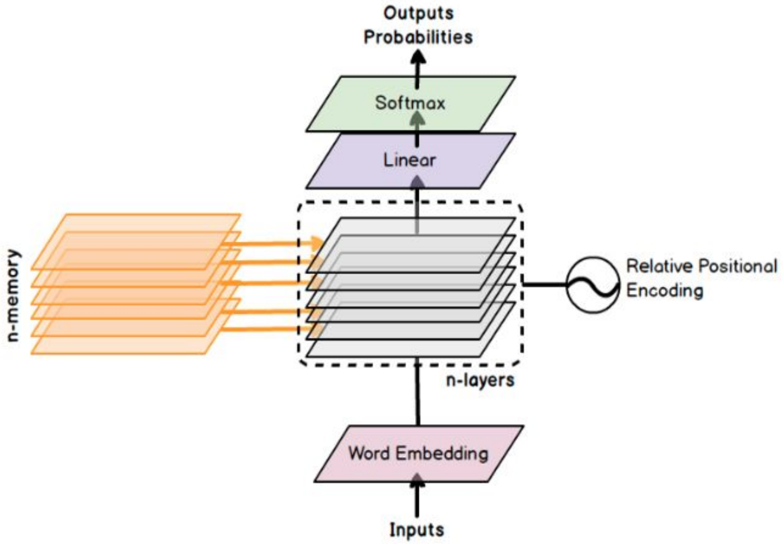
- Cache the hidden states of the previous segment
- Pass them as keys/values for the next

Segment-Level Recurrence

The n-th layer Key, Value are backed by the previous segment's n-1th hidden state

Note $h_{\tau}^{n-1} \rightarrow h_{\tau+1}^{n}$



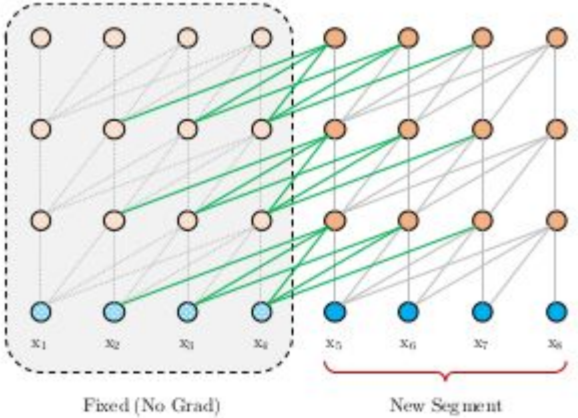


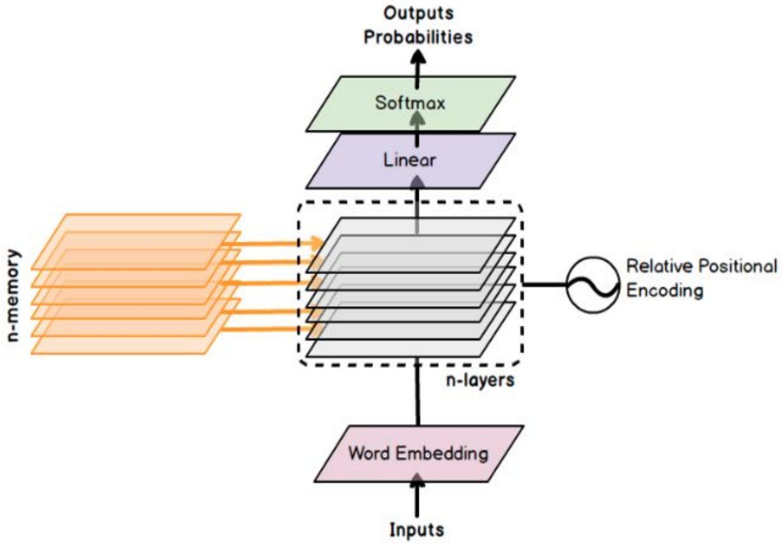
Segment-Level Recurrence

$$\widetilde{\mathbf{h}}_{\tau+1}^{n-1} = \left[\mathrm{SG}(\mathbf{h}_{\tau}^{n-1}) \circ \mathbf{h}_{\tau+1}^{n-1}\right],$$

$$\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n} = \mathbf{h}_{\tau+1}^{n-1}\mathbf{W}_q^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_k^{\top}, \widetilde{\mathbf{h}}_{\tau+1}^{n-1}\mathbf{W}_v^{\top},$$

$$\mathbf{h}_{\tau+1}^{n} = \text{Transformer-Layer}\left(\mathbf{q}_{\tau+1}^{n}, \mathbf{k}_{\tau+1}^{n}, \mathbf{v}_{\tau+1}^{n}\right)$$





Relative Position Encoding

Absolute Position Encoding leads to [0, …, L] [0, …, L]

- Incoherence in representing position
- Relatively less temporal info given longer context

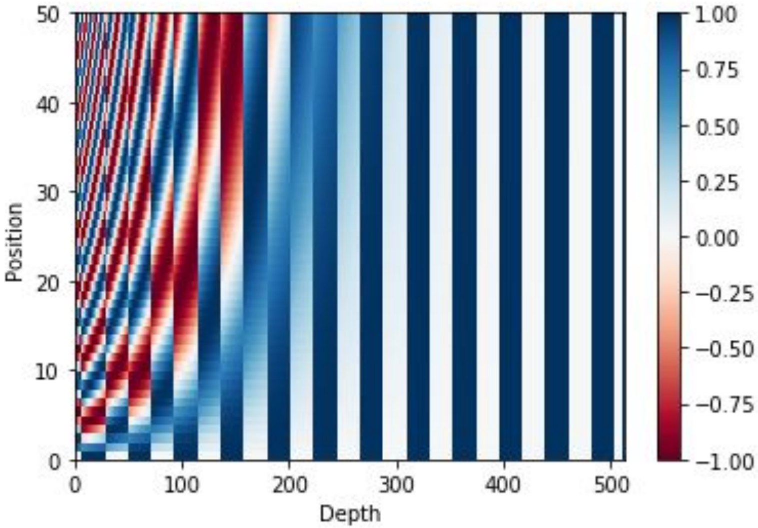
To reuse the previous states effectively

- Need to manage the positional information coherent

Absolute Position Encoding

$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{model}})$$

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



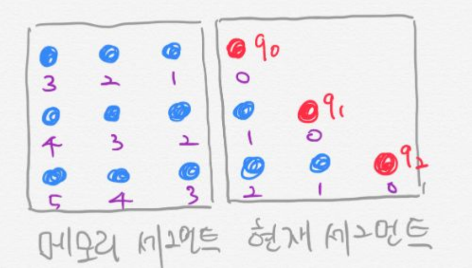
Reparameterization

$$\mathbf{A}_{i,j}^{\mathrm{abs}} = \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}_{x_i}^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(b)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}_i^\top \mathbf{W}_q^\top \mathbf{W}_k \mathbf{U}_j}_{(d)}.$$

$$A_{i,j} = q_i^T k_j + q_i^T k_{U,j} + q_{u,i}^T k_j + q_{u,i}^T k_{u,j}$$

(a) content_based addressing
(b) content-dependent positional bias
(c) global content bias
(d) global positional bias

Relative Position Encoding



$$\mathbf{A}^{\text{abs}}_{i,j} = \underbrace{\mathbf{E}^\top_{x_i}\mathbf{W}^\top_q\mathbf{W}_k\mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}^\top_{x_i}\mathbf{W}^\top_q\mathbf{W}_k\mathbf{U}_j}_{(b)}$$
$$+ \underbrace{\mathbf{U}^\top_i\mathbf{W}^\top_q\mathbf{W}_k\mathbf{E}_{x_j}}_{(c)} + \underbrace{\mathbf{U}^\top_i\mathbf{W}^\top_q\mathbf{W}_k\mathbf{U}_j}_{(d)}.$$

$$A_{i,j} = q_i^T k_j + q_i^T k_{U,j}$$
$$+ q_{u,i}^T k_j + q_{u,i}^T k_{u,j}$$

$$\mathbf{A}^{\text{rel}}_{i,j} = \underbrace{\mathbf{E}^\top_{x_i}\mathbf{W}^\top_q\mathbf{W}_{k,E}\mathbf{E}_{x_j}}_{(a)} + \underbrace{\mathbf{E}^\top_{x_i}\mathbf{W}^\top_q\mathbf{W}_{k,R}\mathbf{R}_{i-j}}_{(b)}$$
$$+ \underbrace{u^\top\mathbf{W}_{k,E}\mathbf{E}_{x_j}}_{(c)} + \underbrace{v^\top\mathbf{W}_{k,R}\mathbf{R}_{i-j}}_{(d)}.$$

$$A_{i,j} = q_i^T k_j + q_i^T k_{R,i-j}$$
$$+ u^T k_j + v^T k_{R,i-j}$$

Causal Attention Masking

$q_0^T k_3$	$q_0^T k_2$	$q_0^T k_1$	$q_0^T k_0$	0	0
$q_1^T k_4$	$q_1^T k_3$	$q_1^T k_2$	$q_1^T k_1$	$q_1^T k_0$	0
$q_2^T k_5$	$q_2^T k_4$	$q_2^T k_3$	$q_2^T k_2$	$q_2^T k_1$	$q_2^T k_0$

$q_0^T k_5$	$q_0^T k_4$	**$q_0^T k_3$**	**$q_0^T k_2$**	**$q_0^T k_1$**	**$q_0^T k_0$**
$q_1^T k_5$	**$q_1^T k_4$**	**$q_1^T k_3$**	**$q_1^T k_2$**	**$q_1^T k_1$**	**$q_1^T k_0$**
$q_2^T k_5$	**$q_2^T k_4$**	**$q_2^T k_3$**	**$q_2^T k_2$**	**$q_2^T k_1$**	**$q_2^T k_0$**

Attention Scores

Queries

robot	must	obey	orders

X

Keys

robot	must	obey	orders
robot	must	obey	orders
robot	must	obey	orders
robot	must	obey	orders

=

Scores (before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Scores (before softmax)

0.11	0.00	0.81	0.79
0.19	0.50	0.30	0.48
0.53	0.98	0.95	0.14
0.81	0.86	0.38	0.90

Apply Attention Mask →

Masked Scores (before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Masked Scores (before softmax)

0.11	-inf	-inf	-inf
0.19	0.50	-inf	-inf
0.53	0.98	0.95	-inf
0.81	0.86	0.38	0.90

Softmax (along rows) →

Scores

1	0	0	0
0.48	0.52	0	0
0.31	0.35	0.34	0
0.25	0.26	0.23	0.26

Result

80% longer than RNNs and 450% longer than vanilla Transformers

Up to 1,800+ times faster than a vanilla Transformer during evaluation

Better performance in perplexity

- on long sequences (because of the long-term dependency modeling)
- on short sequences (because of the context fragmentation problem)