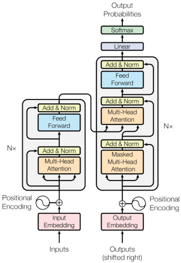
Transformer

Transformer-XL



Learning goals

- Understand the limitations for long sequences
- Understand the Segment Recurrence mechanism
- Understand relative positional encodings

LIMITATION OF THE TRANSFORMER

Table 1: Maximum path lengths, per-layer complexity and minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$
	not cool	cool	

Source: Vaswani et al. (2017)

Advantage:

- Every token can *directly* attend to each other token
- Cf. RNN: At worst *n* sequential operations (last to first token)

Severe Limitation:

- Every token attends to each other token (incl. itself)
  $\rightarrow$ We need to calculate n^2 attention weights
- Computational complexity of Transformer scales quadratically with the sequence length
  $\rightarrow$ Longer sequences are disproportionally expensive

TRANSFORMER-XL

Key facts:

- Objective: Autoregressive Language Modeling task
- Transformer decoder model
- Addresses long sequences
- Assumption: *No* infinite memory & compute; limited resources
- (Possible) Solution Vanilla Transformer:
  - Split corpus into shorter segments
  - Limited contextual information
- Solution Transformer-XL:
  - Segment-level recurrence mechanism
  - Able to model longer-term dependencies

TRANSFORMER-XL

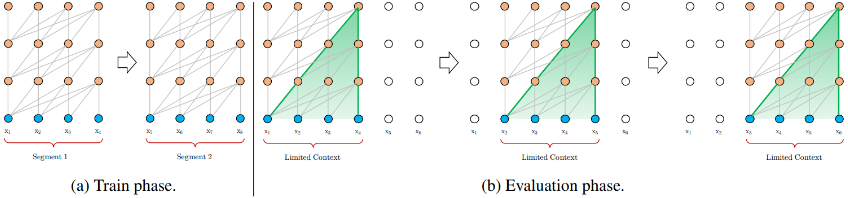


Figure 1: Illustration of the vanilla model with a segment length 4.

Source: Dai et al. (2019)

- Contextual information limited to segments
- Does not respect semantic or syntactic boundaries

TRANSFORMER-XL

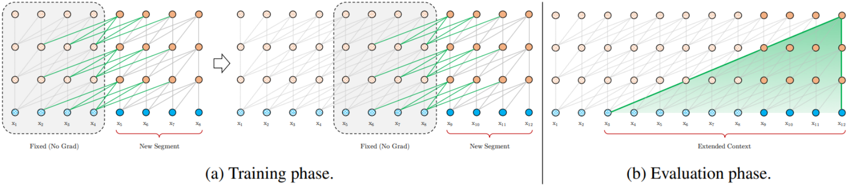


Figure 2: Illustration of the Transformer-XL model with a segment length 4.

Source: Dai et al. (2019)

- Caches hidden states from the previous segment
- Contextual information flows across segments

SEGMENT RECURRENCE

- Let $s_\tau = [x_{\tau,1}, \ldots, x_{\tau,L}]$ and $s_{\tau+1} = [x_{\tau+1,1}, \ldots, x_{\tau+1,L}]$ be two consecutive segments of length L.
- Let $h_\tau^n \in \mathbb{R}^{L \times d}$ denote the n-th layer hidden states for s_τ.
- Using segment recurrence, the n-th layer hidden states for the following segment $s_{\tau+1}$ are computed as follows:

$$\tilde{h}_{\tau+1}^{n-1} = Concat[SG(h_\tau^{n-1}), h_{\tau+1}^{n-1}]$$

$$q_{\tau+1}^n = h_{\tau+1}^{n-1} W_q^\mathsf{T}; \quad k_{\tau+1}^n = \tilde{h}_{\tau+1}^{n-1} W_k^\mathsf{T}; \quad v_{\tau+1}^n = \tilde{h}_{\tau+1}^{n-1} W_v^\mathsf{T}$$

$$h_{\tau+1}^n = Trafo(q_{\tau+1}^n, k_{\tau+1}^n, v_{\tau+1}^n),$$

where $SG(\cdot)$ stands for "‘stop-gradient"’.

RELATIVE POSITIONAL ENCODINGS

Problem:

- *Absolute* positional encodings (PE) would assign the same embedding to words in similar positions in both segments
- No positional difference between $x_{\tau,j}$ and $x_{\tau+1,j}$

Solution:

- Inject information about the relative distance between a query vector and the respective key vectors directly into the Attention mechanism
- *Comment:* Using relative PEs utterly necessary here, but also applicable independently of the segment recurrence

RELATIVE POSITIONAL ENCODINGS

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

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

RELATIVE POSITIONAL ENCODINGS

Solution:

- Replace all absolute PEs with relative ones (fixed $+$ sinusoidal)
  $\rightarrow$ $\mathbf{R}_{i-j}$ instead of $\mathbf{U}_j$ in (b) and (d)
- Replace all positional query vectors with single trainable embeddings
  $\rightarrow$ u and v instead of $\mathbf{U}_i^\top \mathbf{W}_q^\top$ in (c) and (d)
- Use separate weight matrices for linearly projecting $\mathbf{E}$
  $\rightarrow$ $\mathbf{W}_{k,E}$ and $\mathbf{W}_{k,R}$ instead of $\mathbf{W}_k$