Chapter 7 Tensors for deep learning theory

Analyzing deep learning architectures via tensorization

7.1 Introduction

There are several prominent deep learning architecture classes.

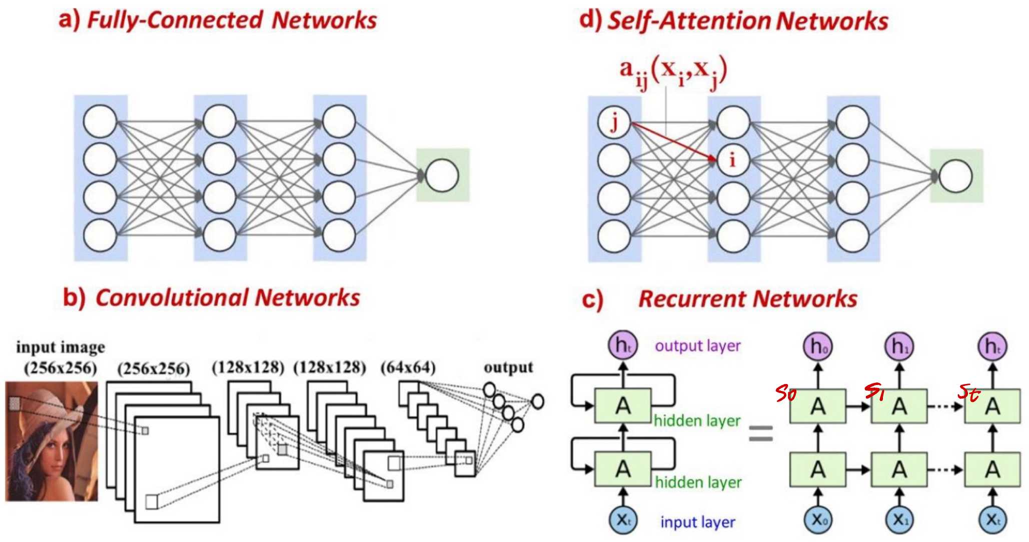


FIGURE 7.1

Prominent deep learning architecture classes. At each layer, fully connected networks connect all outputs to all inputs via learned weights, while in self-attention networks these weights are input-dependent. See a thorough presentation and analysis of self-attention in Section 7.3.1. Convolutional and recurrent networks are discussed in Section 7.4.

a) Fully-Connected Networks: MLP $x \to w_0 x + b_0 \to f \to w_1 x + b_1 \to f \to \cdots \to y$

b) Convolutional Networks: convolution kernal

c) Recurrent Networks: $\cdots \to x_t, x_{t-1}, S_{t-2} \to x_t, S_{t-1} \to S_t \to h_t$, sequence

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

Three high-level topics:

(1) Expressivity (2) Optimization (3) Generalization

7.2 Bounding a function's expressivity via tensorization

A scalar function y over N input vectors $\{\boldsymbol{x}^j\}_{j=1}^N$ in dimension d_x $y: (\mathbb{R}^{d_x})^N \to \mathbb{R}$

▷ A measure of capacity for modeling input dependencies

A partition (A,B) A and B are disjoint subsets of $[N] := \{1, \dots, N\}$

Separation rank of $y: (\mathbb{R}^{d_x})^N \to \mathbb{R}$ w.r.t. (A,B):

$$sep_{(A,B)}(y) := \min\{ R \in \mathbb{N} \cup \{0\} : \exists g_1^A, \dots, g_R^A, g_1^B, \dots, g_R^B : (\mathbb{R}^{d_x})^{N/2} \to \mathbb{R}, \; y(\boldsymbol{x}^1, \dots, \boldsymbol{x}^N) = \sum_{v=1}^{R} g_v^A(\{\boldsymbol{x}^i : i \in A\}) \, g_v^B(\{\boldsymbol{x}^i : i \in B\}) \}.$$

If $sep_{(A,B)}(y) = 1$, then y is seperable. (If y is pdf, independent)

$sep_{(A,B)}(y) \uparrow$, dependency between $\{x^j\}_{j\in A}$ and $\{x^j\}_{j\in B}$ $\uparrow$

An upper bound of it limits its ability to model input dependencies.

A lower bound of it guarantees that this ability is above a threshold.

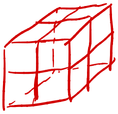
▷ Upper bound

$$y = \sum_{i=1}^{R} y_i \, , \quad \text{the } sep_{(A,B)}(y) \le R$$

▷ Lower bound (Bounding correlations with tensor matricization ranks)

- A tensor $\mathcal{A}$ of order N and dimension M_i in each mode $i \in [N]$

$\mathcal{A}_{j_1, \dots, j_N}$, where $j_i \in [M_i]$ $N = 3, \; M_1 = M_2 = M_3 = 2$



- Grid tensor (tensor-based multivariate function discretization)

A set of points on an exponentially large grid $\Longrightarrow$ a tensor

Fixing a set of template vectors $x^{(1)}, \cdots, x^{(M)} \in \mathbb{R}^{d_x}$,

the point on the grid are the set $\{(x^{(j_1)}, \cdots, x^{(j_N)})\}_{j_1,\cdots,j_N=1}^{M}$.

$\xrightarrow[y(x^1,\cdots,x^N)]{\text{given function}}$ grid tensor induced by y $\quad \mathcal{A}(y) \in (\mathbb{R}^M)^N$

$$\mathcal{A}(y)_{j_1\cdots j_N} \equiv y(x^1 = x^{(j_1)}, \cdots, x^N = x^{(j_N)})$$

A tensor $\mathcal{A}$ of order N and dimension M in each mode $i \in [N]$

- Matricization of $\mathcal{A}$ w.r.t. the balanced partition (A, B) $\quad [\![\mathcal{A}]\!]_{A,B} \in \mathbb{R}^{M^{N/2} \times M^{N/2}}$

The arrangement of the tensor elements as a matrix.

rows $\longrightarrow A$, columns $\longrightarrow B$

$A = \{a_1, \cdots, a_{N/2}\}$, $\quad a_1 < \cdots < a_{N/2}$

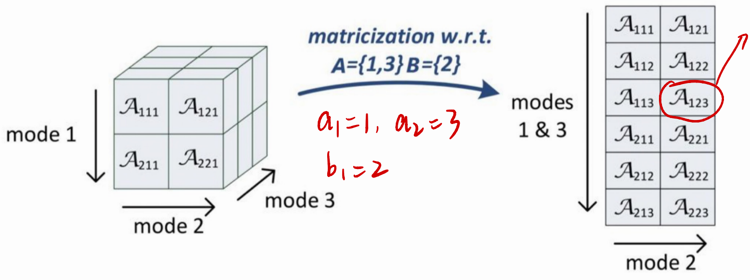
$B = \{b_1, \cdots, b_{N/2}\}$, $\quad b_1 < \cdots < b_{N/2}$

The entry $\mathcal{A}_{j_1\cdots j_N}$ is stored in the following matrix entry:

$$([\![\mathcal{A}]\!]_{A,B})_{row,\,col} = \mathcal{A}_{j_1\cdots j_N},$$

$$row = 1 + \sum_{t=1}^{N/2} (j_{a_t} - 1)\, M^{N/2-t}$$

$$col = 1 + \sum_{t=1}^{N/2} (j_{b_t} - 1)\, M^{N/2-t}$$


mode 1
$\mathcal{A}_{111}$
$\mathcal{A}_{121}$
$\mathcal{A}_{211}$
$\mathcal{A}_{221}$
mode 2
mode 3
matricization w.r.t.
A={1,3} B={2}
modes
1 & 3
$\mathcal{A}_{111}$ $\mathcal{A}_{121}$
$\mathcal{A}_{112}$ $\mathcal{A}_{122}$
$\mathcal{A}_{113}$ $\mathcal{A}_{123}$
$\mathcal{A}_{211}$ $\mathcal{A}_{221}$
$\mathcal{A}_{212}$ $\mathcal{A}_{222}$
$\mathcal{A}_{213}$ $\mathcal{A}_{223}$
mode 2

$a_1 = 1,\ a_2 = 3$

$b_1 = 2$

$j_1 = 1, j_2 = 2, j_3 = 3$

$row = 1 + \sum_{t=1}^{2} (j_{a_t} - 1)\cdot 2^{2-t}$

$= 1 + (3-1)\cdot 2^0 = 3$

$col = 1 + \sum_{t=1}^{1} (j_{b_t} - 1)\cdot 2^{1-t}$

$= 1 + (2-1)\cdot 2^0 = 2$

FIGURE 7.3

An illustrative example of a matricization of an order-3 tensor.

The following claim establishes a fundamental relation between a function's separation rank and the rank of the matrix obtained by the corresponding grid tensor matricization, which lower bounds it.

Claim 7.1. Let y be a scalar function over N inputs $\{x^j \in \mathbb{R}^{d_x}\}_{j=1}^{N}$ and let (A,B) be any partition of N. For any integer M and any set of template vectors $x^{(1)}, \dots, x^{(M)} \in \mathbb{R}^{d_x}$, we have

$$sep_{(A,B)}(y) \geq rank(\llbracket \mathcal{A}(y) \rrbracket_{A,B}),$$

where $\mathcal{A}(y)$ is the grid tensor of y with respect to the above template vectors.

Proof. ① $sep_{(A,B)}(y) = \infty$ ✓

② $sep_{(A,B)}(y) = R \in \mathbb{N}$ and $\{g_v^A, g_v^B\}_{v=1}^{R}$ the functions of decomposition

$$y(x^1, \dots, x^N) = \sum_{v=1}^{R} g_v^A(x^i : i \in A) \cdot g_v^B(x^i : i \in B) \quad \text{holds}$$

By def. of the grid tensor,

$$\begin{aligned}\mathcal{A}(y)_{j_1 \cdots j_N} &\equiv y(x^1 = x^{(j_1)}, \dots, x^N = x^{(j_N)}) \\ &= \sum_{v=1}^{R} g_v^A(x^{(j_i)}, i \in A) \cdot g_v^B(x^{(j_i)} : i \in B) \\ &\equiv \sum_{v=1}^{R} V^v_{j_i : i \in [A]} U^v_{j_i : i \in [B]} \qquad \text{holds}\end{aligned}$$

where V^v and U^v are the tensors holding the values of g_v^A, g_v^B, at the input points defined by the template vectors.

$\llbracket V^v \rrbracket_{A,B} \rightarrow$ column vectors (only have A part) $\rightarrow \mathbf{v}_v$

$\llbracket U^v \rrbracket_{A,B} \rightarrow$ row vectors (only have B part) $\rightarrow \mathbf{u}_v^T$

$$[\![A(y)]\!]_{A,B} = \sum_{v=1}^{R} v_v u_v^T, \qquad \text{holds}$$

$$\mathrm{Rank}(v_v),\ \mathrm{Rank}(u_v^T) \le 1$$

$$\mathrm{Rank}(v_v u_v^T) \le \min(\mathrm{Rank}(v_v), \mathrm{Rank}(u_v^T)) \le 1$$

$$\mathrm{Rank}\left(\sum_{v=1}^{R} v_v u_v^T\right) \le \sum_{v=1}^{R} \mathrm{Rank}(v_v u_v^T) = R = sep_{(A,B)}(y). \qquad \#$$

The entries of analyzed functions' grid tensors vary polynomially with learned weight Θ. Finding a single network weight configuration with the above property guarantees the bound for all configurations but a set of measure zero.

Claim 7.2. Let $M, N, K \in \mathbb{N}$, $1 \le R \le \min\{M, N\}$, and a polynomial mapping $A: \mathbb{R}^K \to \mathbb{R}^{M\times N}$ (i.e. for every $i \in [M]$, $j \in [N]$, $A_{ij}: \mathbb{R}^K \to \mathbb{R}$ is a polynomial function). If there exists a point $\Theta \in \mathbb{R}^K$ s.t. $\mathrm{rank}(A(\Theta)) \ge R$, then the set $\{\Theta \in \mathbb{R}^K \mid \mathrm{rank}(A(\Theta)) < R\}$ has zero measure.

Proof. $\mathrm{rank}(A(\Theta)) \ge R \iff \exists$ a nonzero $R \times R$ $\overset{\text{余子式}}{\underline{\text{minor}}}$ of $A(\Theta)$

is polynomial in the entries of $A(\theta)$ / θ

$c = \binom{M}{R} \cdot \binom{N}{R}$ be the number of minors in A, minors: $\{f_i(\Theta)\}_{i=1}^{c}$

Define the polynomial function $f(\Theta) = \sum_{i=1}^{c} f_i(\theta)^2$

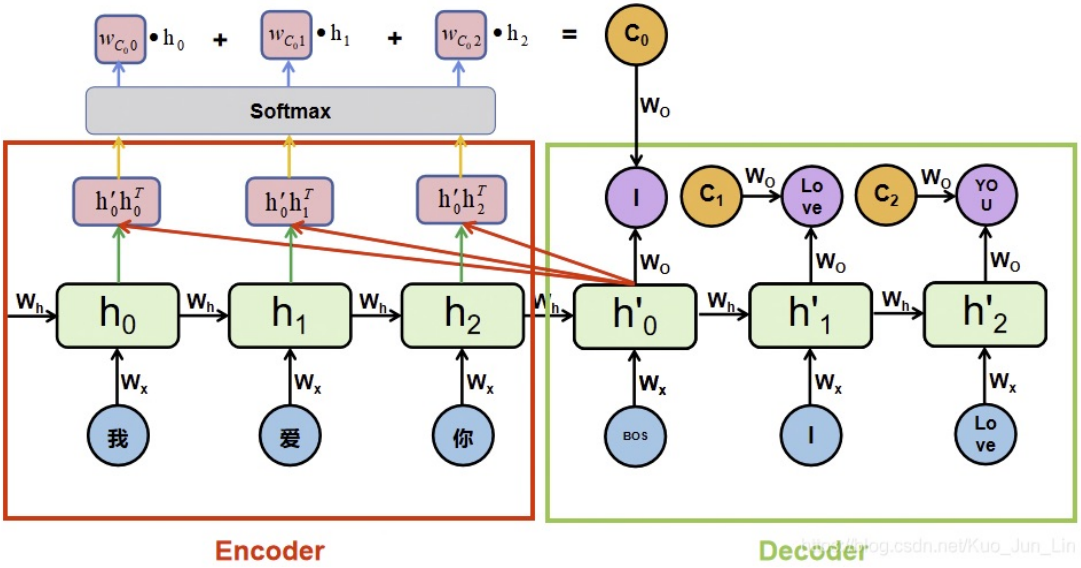
$f(\Theta) = 0 \iff$ For all $i \in [c]$, $f_i(\theta) = 0 \iff \mathrm{rank}(A(\theta)) < R$

By relying on Claim 7.2 above, we establish lower bounds on $\mathrm{rank}([\![\mathcal{A}(y)]\!]_{A,B})$ by choosing a simple-to-analyze assignment of the network's learned weights Θ and showing that for this value of the weights, there exist template vectors for which the rank of the grid tensor's matricization reaches a certain value. By relying on Claim 7.1 this in turn implies a lower bound on the separation rank of the analyzed architecture.

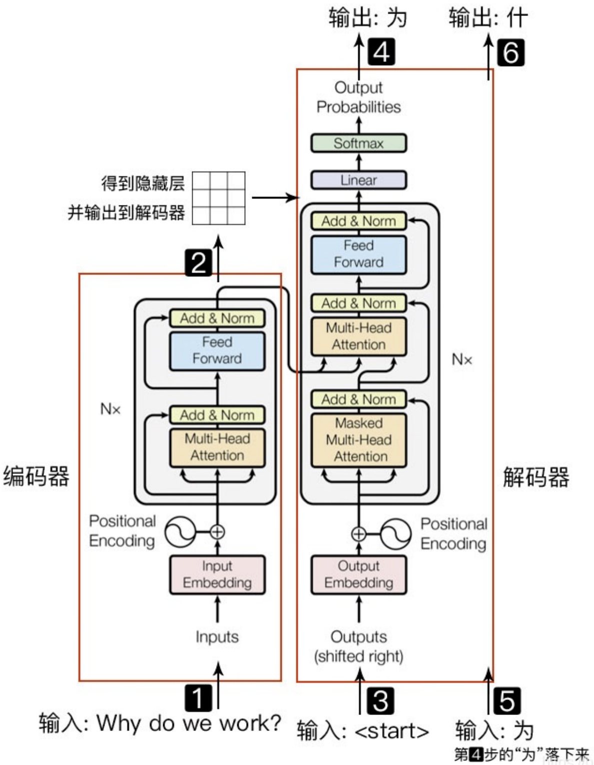
7.3 A case study: self-attention networks

7.3.1 The self-attention mechanism

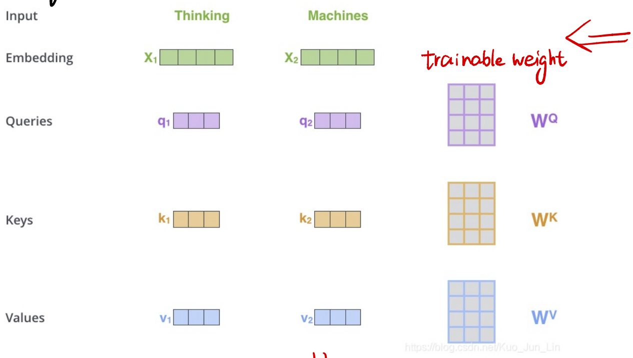
▷ Attention in RNN



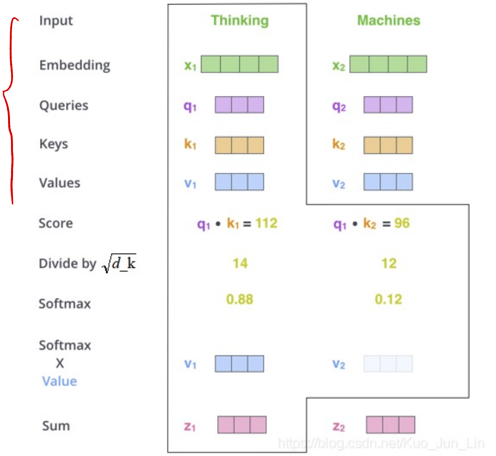
▷ Transformer



Single-head Attention

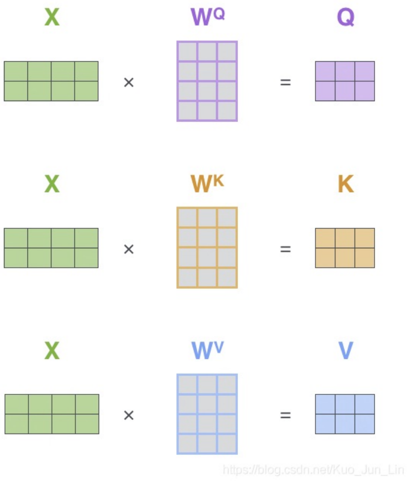


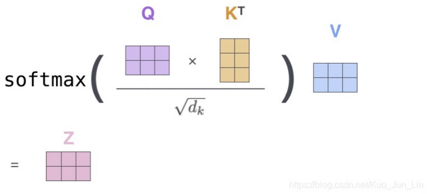
trainable weight



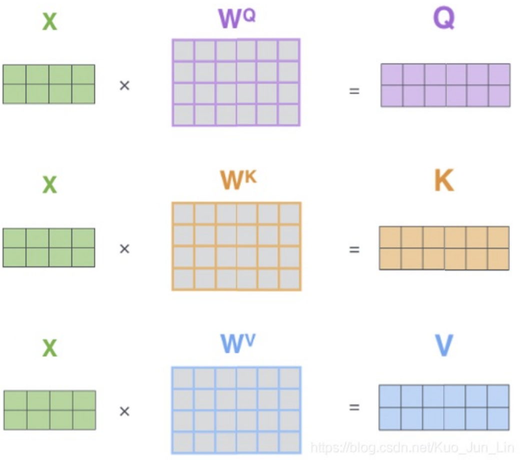
matrix

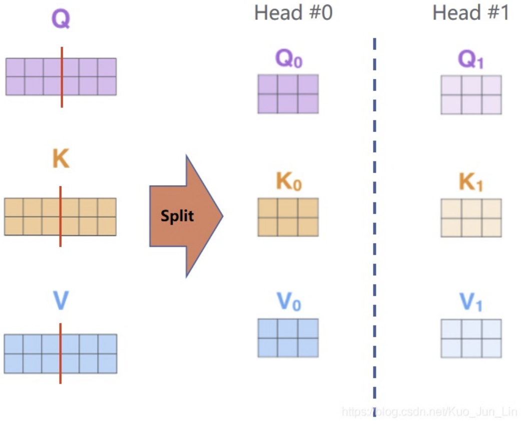
matrix

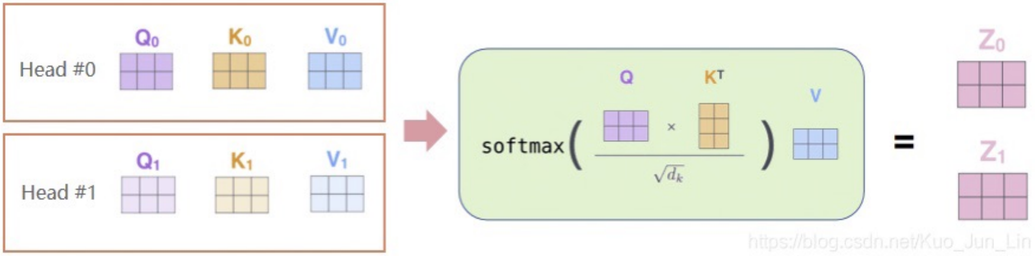


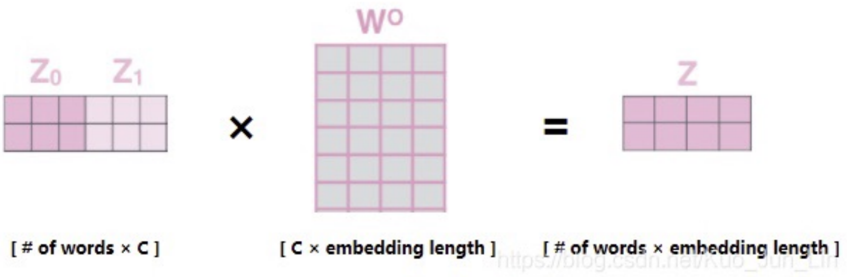


Multi-head Attention








Wo
Z0
Z1
Z
[# of words × C]
[C × embedding length]
[# of words × embedding length]
https://blog.csdn.net/Kuo_Jun_Lin

▷ The operation of a self-attention layer

depth L : the number of concatenated layers

N inputs : $\{x^j \in \mathbb{R}^{d_x}\}_{j=1}^{N} \xrightarrow[\text{layer}]{\text{first self-attention}}$ N outputs : $\{y^{1,j} \in \mathbb{R}^{\overset{\text{width}}{\overset{\uparrow}{\underline{d_x}}}}\}_{j=1}^{N}$

i-th output of layer $l \in [L]$: $y^{l,i} \in \mathbb{R}^{d_x}$

- Single-headed self-attention

$$\mathbf{y}^{l+1,i}(\mathbf{y}^{l,1}, \ldots, \mathbf{y}^{l,N}) = \sum_{j=1}^{N} a^i_j \left(\underline{W^{\mathrm{V},l}}\mathbf{y}^{l,j}\right), \qquad (7.5)$$

$$a^i_j = \langle \underline{W^{\mathrm{Q},l}}\mathbf{y}^{l,i}, \underline{W^{\mathrm{K},l}}\mathbf{y}^{l,j}\rangle, \quad \in \mathbb{R}^{d_x \times d_x}$$

- Multi-headed self-attention

$$\mathbf{y}^{l+1,i}(\mathbf{y}^{l,1}, \ldots, \mathbf{y}^{l,N}) = \sum_{h=1}^{H} \overset{\in \mathbb{R}^{d_x \times d_a}}{\underline{W^{\mathrm{O},l,h}}} \sum_{j=1}^{N} a^i_{hj} \left(\underline{W^{\mathrm{V},l,h}}\mathbf{y}^{l,j}\right), \qquad (7.6)$$

$$\in \mathbb{R}^{d_a \times d_x}, \quad d_a = d_x / H$$

$$a^i_{hj} = \langle \underline{W^{\mathrm{Q},l,h}}\mathbf{y}^{l,i}, \underline{W^{\mathrm{K},l,h}}\mathbf{y}^{l,j}\rangle,$$

By recursively applying Eq. (7.6) L times we attain a depth-L self-attention network. In Section 7.3.4 we prove that the recursive relation in Eq. (7.6) implies that the function realized by a network with representation dimension d_x and H attention heads per layer at output location $i \in [N]$ can be written as (Corollary 7.1)

$$\mathbf{y}^{i,L,d_x,H,\Theta}\left(\mathbf{x}^1, \ldots, \mathbf{x}^N\right) := \sum_{j_1,\ldots,j_C=1}^{N} \mathbf{g}^L\left(\mathbf{x}^i, \mathbf{x}^{j_1}, \ldots, \mathbf{x}^{j_C}\right), \tag{7.7}$$

where Θ stands for all $4LH$ learned weight matrices: $\forall (l,h) \in [L] \otimes [H]$: $W^{\mathrm{K},l,h}$, $W^{\mathrm{Q},l,h}$, $W^{\mathrm{V},l,h} \in \mathbb{R}^{d_a \times d_x}$, and $W^{\mathrm{O},l,h} \in \mathbb{R}^{d_x \times d_a}$, and the function $\mathbf{g}^L$ is a placeholder which integrates $C + 1 := \frac{3^L - 1}{2} + 1$ different network input vectors.

▷ Partition invariance of the self-attention separation rank

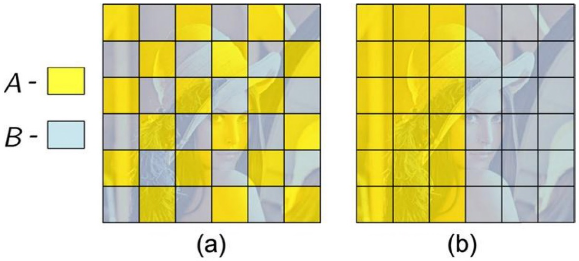


FIGURE 7.4

The separation rank of the function realized by self-attention networks is agnostic to the input partition. In contrast, the functions realized by convolutional and recurrent networks have exponentially higher separation ranks w.r.t. the interleaved partition (a) than w.r.t. the left-right partition (b).

Proposition 7.1. *For $p \in [d_x]$, let $y_p^{i,L,d_x,H,\Theta}$ be the scalar function computing the p-th entry of an output vector at position $i \in [N]$ of the depth-L self-attention network with embedding dimension d_x and H attention heads per layer, defined in Eqs. (7.6) and (7.7). Then, its separation rank w.r.t. balanced partitions, which obey $A \cup B = [N]$, $|A|, |B| = N/2$, is invariant to the identity of the partition, i.e., $\forall A \cup B = [N]$, $\tilde{A} \cup \tilde{B} = [N]$, s.t. $|A|, |B|, |\tilde{A}|, |\tilde{B}| = N/2$:*

$$\mathrm{sep}_{(A,B)}\left(y_p^{i,L,d_x,H,\Theta}\right) = \mathrm{sep}_{(\tilde{A},\tilde{B})}\left(y_p^{i,L,d_x,H,\Theta}\right). \tag{7.8}$$

Accordingly, for the discussion on self-attention, we will omit the specification of the partition, denoting $\mathrm{sep}(y_p^{i,L,d_x,H,\Theta})$ as the separation rank of $y_p^{i,L,d_x,H,\Theta}$ w.r.t. any balanced partition of the inputs.

Proof. $A=(a_1,\cdots,a_{N/2})$, $B=(b_1,\cdots,b_{N/2})$, $\breve{A}=(\breve{a}_1,\cdots,\breve{a}_{N/2})$, $\breve{B}=(\breve{b}_1,\cdots,\breve{b}_{N/2})$

By $\pi\in S_N$ the unique permutation that satisfies

$$\forall\, m\in\left[\frac{N}{2}\right],\quad \pi(a_m)=\breve{a}_m \wedge \pi(b_m)=\breve{b}_m$$

Assume that $a_1=\breve{a}_1=i$, where i is the output location.

Assume that $\mathrm{sep}(y^{i,L,d_x,H,\Theta}; A,B)=R$, there exist $g_1^A,\cdots,g_R^A;\ g_1^B,\cdots,g_R^B$, s.t.

$$\forall \mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\in\mathbb{R}^{d_x}:$$

$$y_p^{i,L,d_x,H,\Theta}\left(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\right)=\sum_{v=1}^{R} g_v^A\left(\mathbf{x}^{(a_1)},\ldots,\mathbf{x}^{(a_{\frac{N}{2}})}\right)g_v^B\left(\mathbf{x}^{(b_1)},\ldots,\mathbf{x}^{(b_{\frac{N}{2}})}\right).$$

Since both a_1 and $\pi(a_1)$ are equal to i, the summations over $j_1,\ldots,j_C$ in Eq. (7.7) imply that for any $\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\in\mathbb{R}^{d_x}$ we have

$$y_p^{i,L,d_x,H,\Theta}\left(\mathbf{x}^{(1)},\ldots,\mathbf{x}^{(N)}\right)=y_p^{i,L,d_x,H,\Theta}\left(\mathbf{x}^{(\pi(1))},\ldots,\mathbf{x}^{(\pi(N))}\right),\qquad x^{(i)}=x^{(\pi(i))}$$

and therefore

$$=\sum_{v=1}^{R} g_v\left(\mathbf{x}^{(\pi(a_1))},\ldots,\mathbf{x}^{(\pi(a_{\frac{N}{2}}))}\right)g_v'\left(\mathbf{x}^{(\pi(b_1))},\ldots,\mathbf{x}^{(\pi(b_{\frac{N}{2}}))}\right)$$

$$=\sum_{v=1}^{R} g_v\left(\mathbf{x}^{(\tilde{a}_1)},\ldots,\mathbf{x}^{(\tilde{a}_{\frac{N}{2}})}\right)g_v'\left(\mathbf{x}^{(\tilde{b}_1)},\ldots,\mathbf{x}^{(\tilde{b}_{\frac{N}{2}})}\right).$$

So we proved that

$$\mathrm{sep}\left(y_p^{i,L,d_x,H,\Theta};\tilde{A},\tilde{B}\right)\le\mathrm{sep}\left(y_p^{i,L,d_x,H,\Theta};A,B\right).$$

Finally, by switching the roles of $\tilde{A}$, $\tilde{B}$ and A, B, we can get the inverse inequality so we conclude that

$$\mathrm{sep}\left(y_p^{i,L,d_x,H,\Theta};\tilde{A},\tilde{B}\right)=\mathrm{sep}\left(y_p^{i,L,d_x,H,\Theta};A,B\right).\qquad\square$$

It does not integrate the input in a predefined pattern like convolutional networks, but rather computes the input integration in an input-dependent manner (a_j^i)

7.3.2 Self-attention architecture expressity questions

- The depth-to-width interplay in self attention

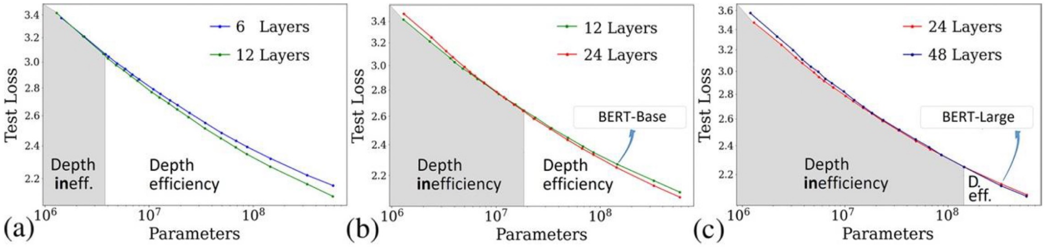


FIGURE 7.5

From [47], an experimental validation of the two depth efficiency/**in**efficiency regimes predicted by the tensorial framework of this chapter. [47] further showed that the transition between regimes occurs in exponentially larger network sizes as the networks gets deeper, in agreement with the theory in Section 7.3.3.1.

- The input embedding rank bottleneck in self-attention

When the dimensionality of the inputs is lower than the network width, the network width's contribution to expressivity is capped.

$\forall j$ the input $x^j \in \mathbb{R}^{d_x}$ can be written as

$$\mathbf{x}^j = M\mathbf{w}^j \tag{7.9}$$

$$\text{for } \; M \in \mathbb{R}^{d_x \times r} \; ; \; \mathbf{w}^j \in R^r$$

$$\text{s.t. } \; r < d_x ,$$

where r is referred to as the input embedding rank. For example, a naive representation of an image as a collection of N RGB vectors in $\mathbb{R}^3$ would imply that $r = 3$, and the identified bottleneck would come into play for $d_x > 3$.

- Mid-architecture rank bottlenecks in self-attention

eg. T5-11B, with $H \cdot d_a \gg d_x$

Low representation dimension caps the ability an excessive parameter increase in the Self-attention operation. (T5-11B is ~50% redundant.)

7.3.3 Results on the operation of self-attention

- The effect of depth L

  $L < \log_3(d_x)$, deepen $>$ widen (theorem 7.1)

  $L > \log_3(dx)$, deepen $\approx$ widen (theorem 7.2)

Theorem 7.1. *For $p \in [d_x]$, let $y_p^{i,L,d_x,H,\Theta}$ be the scalar function computing the p-th entry of an output vector at position $i \in [N]$ of the depth-L self-attention network with embedding dimension d_x and H attention heads per layer, defined in Eqs. (7.6) and (7.7). Let $\text{sep}(y_p^{i,L,d_x,H,\Theta})$ be its separation rank (Section 7.2). If L, d_x obey $L < \log_3(d_x)$, then the following holds almost everywhere in the network's learned parameter space, i.e., for all values of the weight matrices (represented by Θ) but a set of Lebesgue measure zero,*

$$3^{L-2}\big(\log_3(d_x - H) + a\big) \leq \log_3\big(\text{sep}\big(y_p^{i,L,d_x,H,\Theta}\big)\big) \leq \frac{3^L - 1}{2}\log_3(d_x + H) \quad (7.10)$$

with $a = -L + [2 - \log_3 2]$ (note that $\log_3(d_x - H) + a > 0$ in this regime of $L < \log_3(d_x)$).

depth $\sim$ sep : double exponentially , width $\sim$ sep : polynomially

H $\sim$ width : linear

Theorem 7.2. *For $y_p^{i,L,d_x,H,\Theta}$ as defined in Theorem 7.1, if $L > \log_3(d_x)$, then the following holds almost everywhere in the network's learned parameter space, i.e., for all values of the weight matrices (represented by Θ) but a set of Lebesgue measure zero:*

$$\frac{1}{2}d_x \cdot L + b_1 + b_2 \leq \log_3\big(\text{sep}\big(y_p^{i,L,d_x,H,\Theta}\big)\big) \leq 2d_x \cdot L + c_1 + c_2, \quad (7.11)$$

with corrections on the order of L: $b_1 = -L(\frac{H}{2} + 1)$, $c_1 = L$, and on the order of $d_x \log_3(d_x)$: $b_2 = -d_x(1 + \frac{1}{2}\log_3(\frac{d_x - H}{2}))$, $c_2 = -2d_x \cdot \log_3 {d_x}/{2\sqrt{2e}} + \log_3 d_x$.

depth $\sim$ sep & width $\sim$ sep : exponetially

- The effect of bottlenecks

The following theorem states that the network's capacity to model dependencies is harmed by a low-rank input embedding.

Theorem 7.3. *Let $y_p^{i,L,d_x,H,r}$ be the scalar function computing the pth entry of an output vector at position $i \in [N]$ of the H-headed depth-L width-d_x self-attention network defined in Eq. (7.6), where the embedding rank r is defined by Eq. (7.9). Let $sep(y_p^{i,L,d_x,H,r})$ denote its separation rank (Section 7.2.1). Then the following holds (upper bound):*

$$\log(\text{sep}(y_p^{i,L,d_x,H,r})) = \tilde{O}(L \cdot \min\{r, d_x\}). \tag{7.12}$$

Further assume that $L > \log_3 d_x$, $H < r$. Then for all values of the network weights but a set of Lebesgue measure zero, the following holds (lower bound):

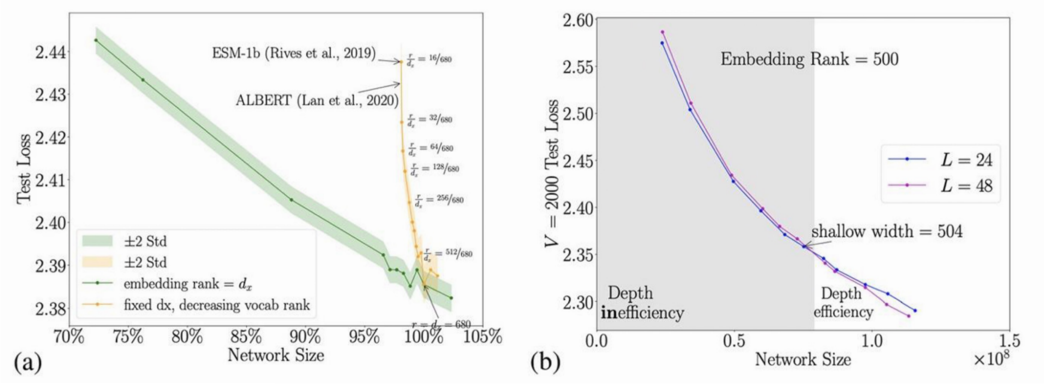
$$\log(\text{sep}(y_p^{i,L,d_x,H,r})) = \tilde{\Omega}(L \cdot (\min\{r, d_x\} - H)). \tag{7.13}$$

① The input embedding rank bottleneck.

A network with a low-rank embedding $r < d_x$ cannot express the operation of a full-rank $r = d_x$ network.

② Effect on the depth-to-width interplay

$d_x^{\text{shallow}} > r$, the deeper network is better.



③ A mid-architecture bottleneck — width caps the internal attention dimension.

Upper bound does not depend on H.