Attention and Self-Attention

Tirtharaj Dash

Dept. of CS & IS and APPCAIR
BITS Pilani, Goa Campus

November 16, 2021

In the prev. lecture:

① seq2seq modeling → decoding

② attention mechanism.

real vector

↓

one-hot decoder output

↓

[ ]

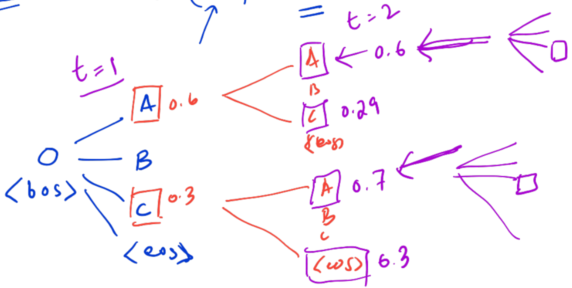
target sentence

greedy —

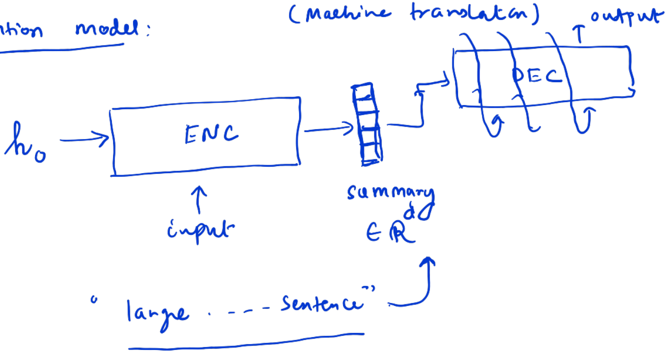
exhaustive —

beam search — $O(K\,|y|\,T_y)$

K=2



Attention model:

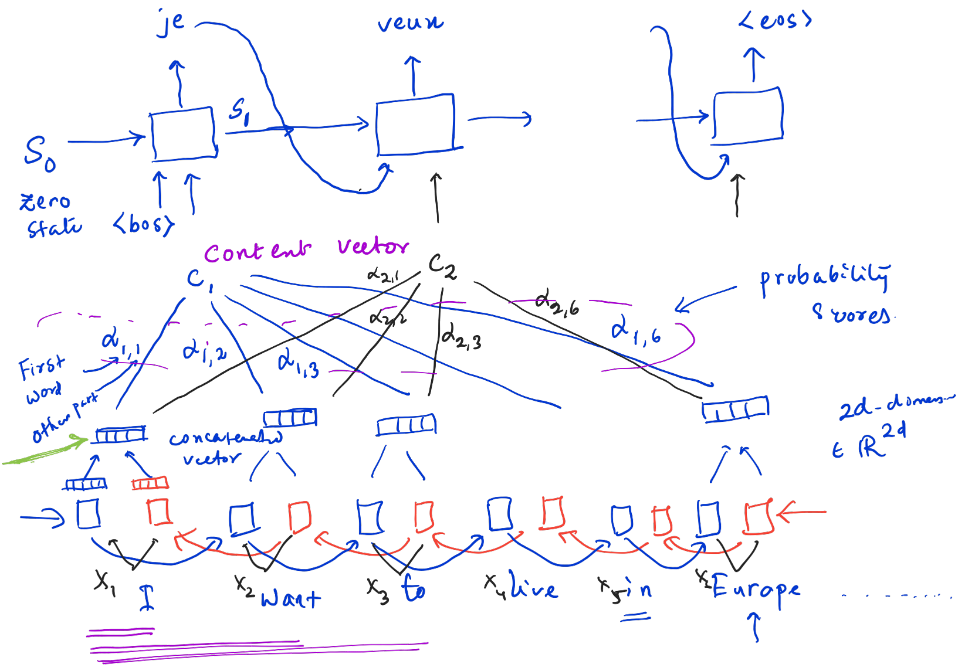


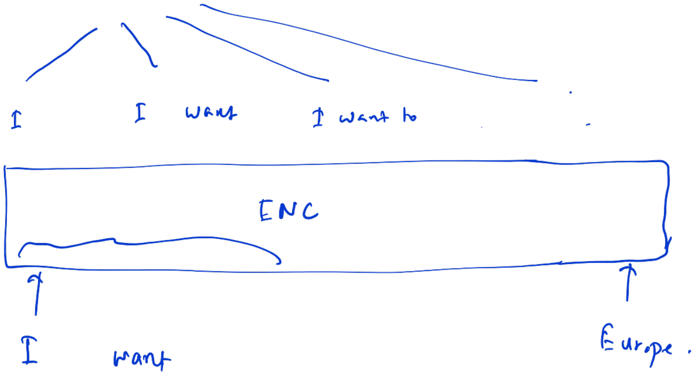
↓ mimic

Which parts are relevant (or should be given more attention to?)

↓

attention mechanism.


je
veux
<eos>
S_0
S_1
zero state
<bos>
Context vector
C_1
C_2
$\alpha_{2,1}$
$\alpha_{2,2}$
$\alpha_{2,3}$
$\alpha_{2,6}$
$\alpha_{1,1}$
$\alpha_{1,2}$
$\alpha_{1,3}$
$\alpha_{1,6}$
probability scores
First word
other part
concatenated vector
2d-dimension
$\in \mathbb{R}^{2d}$
x_1 I
x_2 Want
x_3 to
x_4 live
x_5 in
x_6 Europe


I
I want
I want to
ENC
I
want
Europe.

let $h_{t'} = (\overrightarrow{h}_{t'}, \overleftarrow{h}_{t'})$

hidden state (summary at $t=t'$) from the encoder

what we want

$$\sum_{t'} \alpha_{i,t'} = 1 \quad \text{and} \quad \alpha_{i,j} \geq 0$$

probability.

Content vector

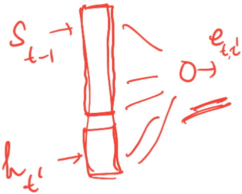
$$C_i = \sum_{t'} \boxed{\alpha_{i,t'}}\, h_{t'}$$

attention score or attention weight

attention weights

$$\alpha_{t,t'} = \text{softmax}(e_{t,t'})$$

$$= \frac{\exp(e_{t,t'})}{\sum_{t'=1}^{T_x} \exp(e_{t,t'})}$$

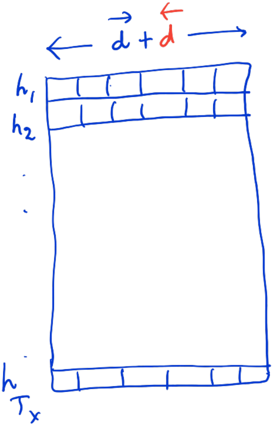


attention scores

$$e_{t,t'} = \tanh([s_{t-1}, h_{t'}] \times W_a + b_a)$$

s_{t-1} → hidden state / summary of the decoder at $t = t-1$

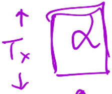
$h_{t'}$ = summary of the encoder at $t = t'$



$$[T_x \times 2d]$$

concat $\oplus$ $\longleftarrow d_1 \longrightarrow$ summary of the decoder (prev. state) s_{t-1}

attention weight



Softmax(tanh($[\,]$))

bias ?

$$\underbrace{[T_x \times (2d + d_1)]} \times W_a [T_x + 1] = \text{attention scores}$$

T_x

Transformer Network:

2017: A. Vaswani et. al. "Attention is all you need"

Idea: → attention based representation (content vectors)

\+

→ convolution style of processing.

parallel processing.

2 mechanism :→

① Self-attention ←

② Multi-head attention. ← Next class.

Self-attention:

$$A(q, K, V)$$

query, key, value.

attention-based representation. "vector" for a word (token).

↕

word : word-embedding (one-hot vector)

I want to live in Europe.

[0, 0, 0, 1 - - - , 0]

as a PostDoc ?

as a businessman ?

as a husband ?

[ ]

rich embedding

"Self-attention" is used to construct a "rich" (very information) embedding of a word (or <tokens>).

attention = Softmax ( )

$$= \frac{\exp(\boxed{})}{\sum \exp(\boxed{})}$$

(dot product)

attention for word or token "i" $= \sum_i \frac{\exp(q \cdot k_i)}{\sum_j \exp(q \cdot k_j)} v_i$

$A(q_i, K, V)$

query at i

Keys Values

p_i ← scalar

v_i ← vector

"self"

$$A(q_i, K, V) = \sum_i \boxed{\frac{\exp(q \cdot k_i)}{\sum_j \exp(q \cdot k_j)}} v_i$$

Softmax.

$\boxed{\text{I}}$ $\boxed{\text{want}}$ to live in $\boxed{\text{Europe.}}$

Person.

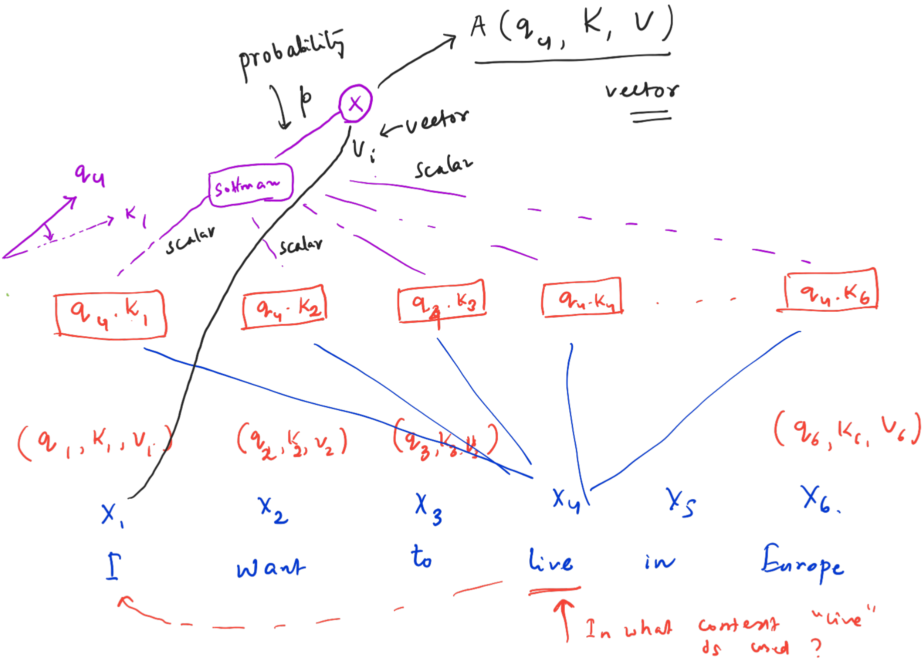
$t=1$ $t=2$ $t=3$ $t=4$

q_4

(query)

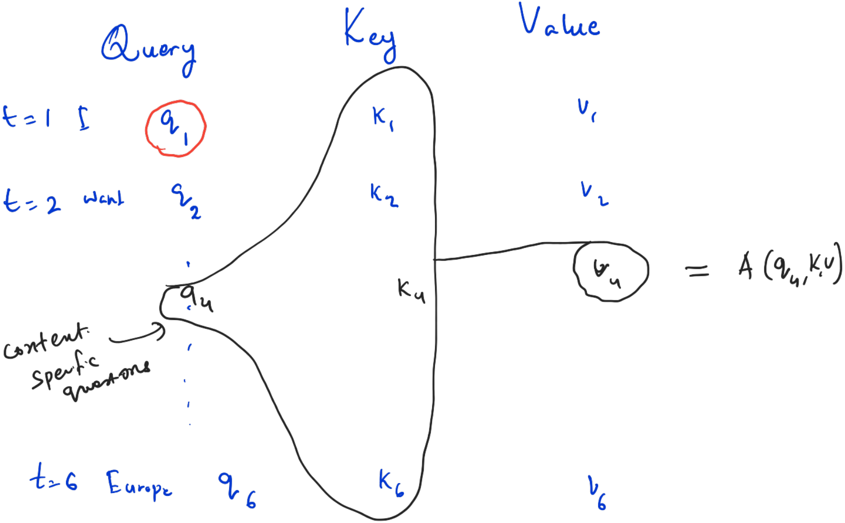
Sentence is looking at

itself.



A (q4, K, V)
vector
probability
p
vector
vi
scalar
q4
softmax
k1
scalar
scalar
q4·k1
q4·k2
q3·k3
q4·k4
q4·k6
(q1, k1, v1)
(q2, k2, v2)
(q3, k3, v3)
(q6, k6, v6)
x1
x2
x3
x4
x5
x6.
I
want
to
live
in
Europe
In what context "live" is used ?

→ associate each word with 3-tuple $\langle q, k, v \rangle$



for every $t = 1 \ldots T_x$:

Compute $A(q_t, K, V)$

$$\downarrow$$

"rich" representations.

word $t=1$ $[\quad A(q_1, K, V)\quad]$

$t=2$ $[\quad A(q_2, K, V)\quad]$

$\vdots$

$t=T_x$ $[\quad A(q_6, K, V)\quad]$

Let X_4 : be the word-embedding for the word "live"

(one-hot representation)

then

$$q_4 : W_Q \times X_4$$

$$k_4 : W_k \times X_4$$

$$v_4 : W_v \times X_v$$

parameters of the model

In vectorised form:

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

$q \cdot k_i$

used to control explosion of the numerator

(scaling parameter)

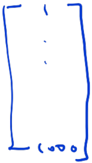
Scaled dot-product attention.

(Attention is all you need.)

example: I want to live in Europe.

sentence length = 7

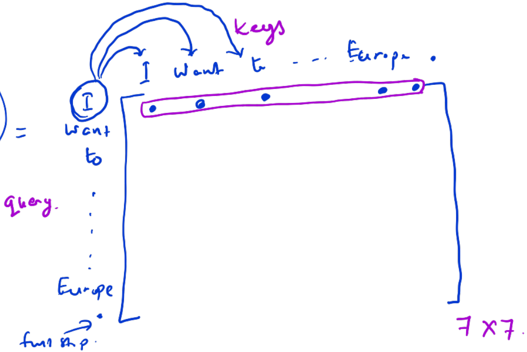
word embedding dimension = 1000



word dictionary.

size of Q, K, V = 7 x 1000

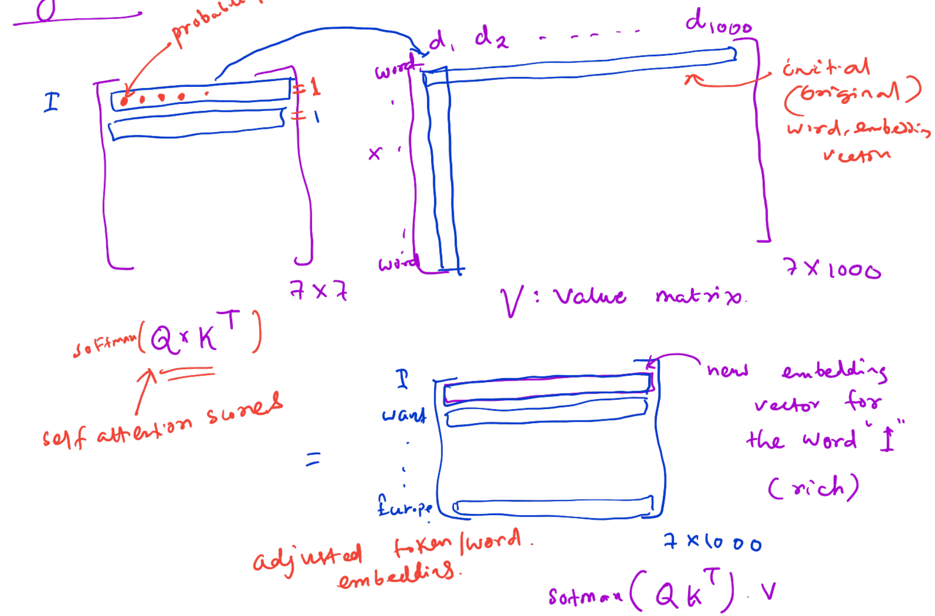
$\text{softmax}\left(Q \times K^T\right) =$



probability matrix

(attention-score matrix)

why V ?


probability
I
=1
=1
7 x 7
d1 d2 . . . . . . d1000
word
x
word
initial (original) word embedding vector
7 X 1000
V : Value matrix.
softmax(Q x K^T)
self attention scores
=
I
word
Europe
new embedding vector for the word "I" (rich)
7 x 1000
adjusted token/word embedding.
Softmax(Q K^T) . V