Sequence Model

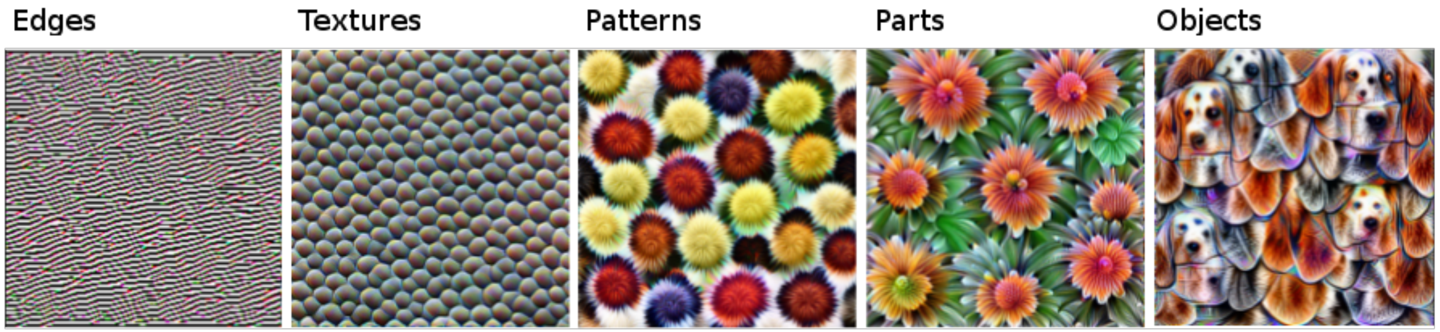
Announcements

1. Makeup exam Dec 11

2. We will release the last reading quiz today

Recap on Convolutional neural network

Learned feature representations in CNN



[https://distill.pub/2017/feature-visualization/]

Objective today

Understanding neural network structures that are suitable for natural language (i.e., sequences of words)

Outline today

1. Word-2-Vec embedding and positional embedding

2. Attention model

3. Putting things together: the Transformer model

Example: autocompletion

e.g., I went to the climbing gym and I ___

Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

$$y_1 \sim P(Y = \cdot \quad x_1, \ldots, x_n) \in \mathbb{R}^{100k}$$

Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

$$y_1 \sim P(Y = \cdot \quad x_1, \ldots, x_n) \in \mathbb{R}^{100k}$$

$$y_2 \sim P(Y = \cdot \quad x_1, \ldots, x_n, y_1)$$

Example: autocompletion

e.g., I went to the climbing gym and I ___

A Language model is a conditional probability model:

$$y_1 \sim P(Y = \cdot \quad x_1, \ldots, x_n) \in \mathbb{R}^{100k}$$

$$y_2 \sim P(Y = \cdot \quad x_1, \ldots, x_n, y_1)$$

$$y_m \sim P(Y = \cdot \quad x_1, \ldots, x_n, y_1, \ldots y_{m-1})$$

Word to Vector Embedding

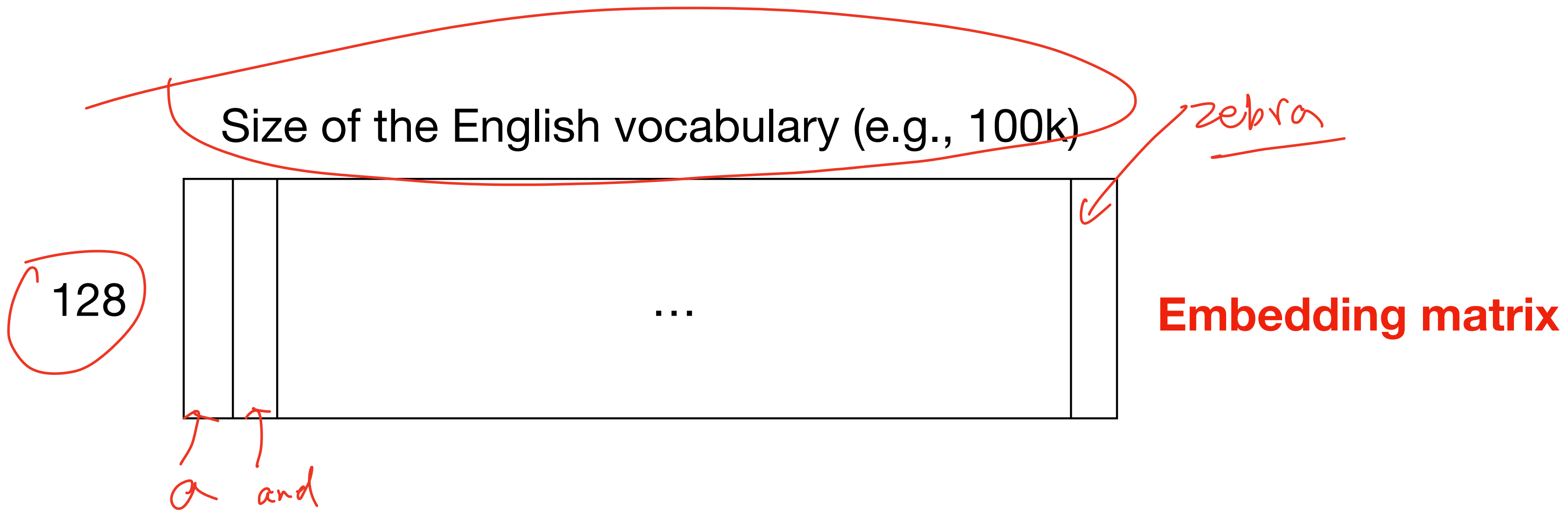
ML models only take vectors of real numbers as inputs…

e.g., I went to the climbing gym and I

Word to Vector Embedding

ML models only take vectors of real numbers as inputs…

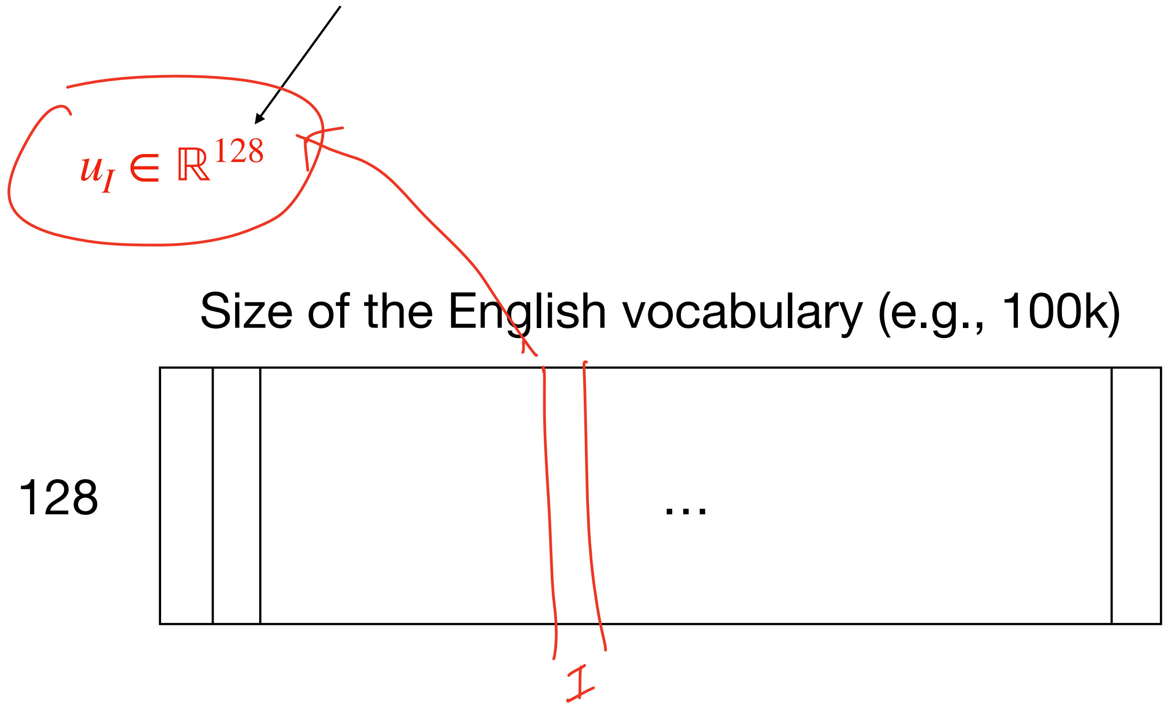
e.g., I went to the climbing gym and I



Word to Vector Embedding

ML models only take vectors of real numbers as inputs…

e.g., I went to the climbing gym and I

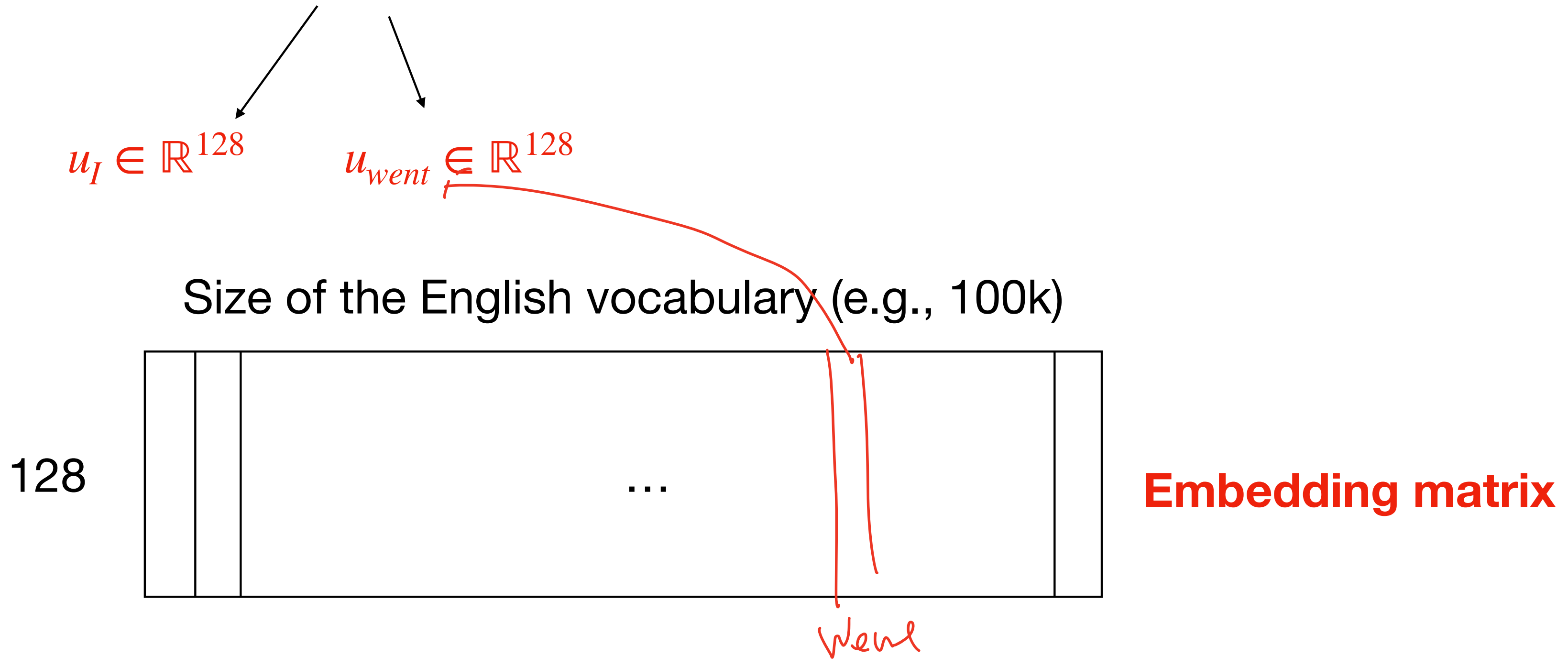

$u_I \in \mathbb{R}^{128}$
Size of the English vocabulary (e.g., 100k)
128
…
I

Embedding matrix

Word to Vector Embedding

ML models only take vectors of real numbers as inputs…

e.g., I went to the climbing gym and I

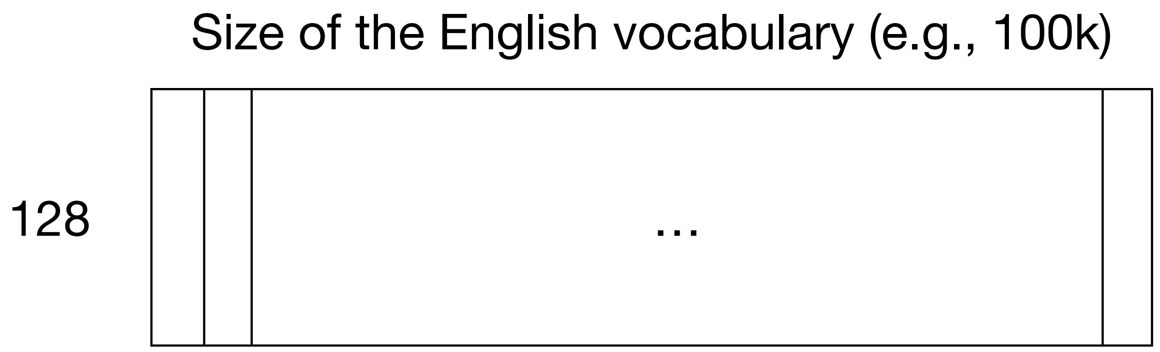

$u_I \in \mathbb{R}^{128}$
$u_{went} \in \mathbb{R}^{128}$
Size of the English vocabulary (e.g., 100k)
128
…
went
Embedding matrix

Word to Vector Embedding

ML models only take vectors of real numbers as inputs…

e.g., I went to the climbing gym and I

$u_I \in \mathbb{R}^{128}$ $u_{went} \in \mathbb{R}^{128}$ $u_{and} \in \mathbb{R}^{128}$ $u_I \in \mathbb{R}^{128}$



Embedding matrix

Positional embedding

Order of the words and their positions matter…

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} \in \mathbb{R}^{128}$

$u_{transformer}$

Positional embedding

Order of the words and their positions matter…

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} \in \mathbb{R}^{128}$

$\underline{+p_4} \in \mathbb{R}^{128}$

u_{mean}

$+ p_{11}$

$u_{transformer} \underline{+p_{13}} \in \mathbb{R}^{128}$

Positional embedding

Order of the words and their positions matter…

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} \in \mathbb{R}^{128}$

$+p_4 \in \mathbb{R}^{128}$

$u_{transformer} + p_{13} \in \mathbb{R}^{128}$

Create positional embedding using sin functions

Positional embedding

Order of the words and their positions matter…

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} \in \mathbb{R}^{128}$

$+p_4 \in \mathbb{R}^{128}$

$u_{transformer} + p_{13} \in \mathbb{R}^{128}$

$t \in \mathbb{N}^+$

t: position

Create positional embedding using sin functions

c_1 small

$$p_t = \begin{bmatrix} \sin(t/c_1) \\ \sin(t/c_2) \\ \dots \\ \sin(t/c_{128}) \end{bmatrix}$$

big

Positional embedding

Order of the words and their positions matter…

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} \in \mathbb{R}^{128}$

$+p_4 \in \mathbb{R}^{128}$

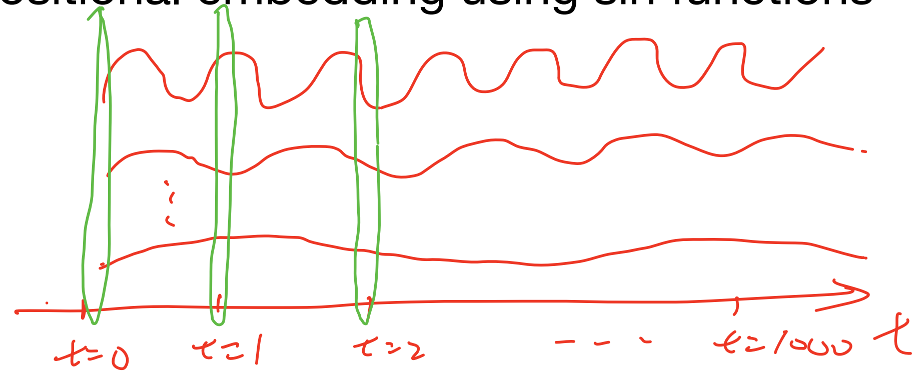
$u_{transformer} + p_{13} \in \mathbb{R}^{128}$

Create positional embedding using sin functions

High frequency

$$p_t = \begin{bmatrix} \sin(t/c_1) \\ \sin(t/c_2) \\ \cdots \\ \sin(t/c_{128}) \end{bmatrix}$$

Low frequency



Summary so far

We turn words into vectors of real numbers

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$$u_{transformer} + p_4 \qquad\qquad u_{transformer} + p_{13} \in \mathbb{R}^{128}$$

Summary so far

We turn words into vectors of real numbers

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

$u_{transformer} + p_4$ $\qquad$ $u_{transformer} + p_{13} \in \mathbb{R}^{128}$

Feature of the word + feature of the position

Outline today

1. Word-2-Vec embedding and positional embedding

2. Attention model

3. Putting things together: the Transformer model

Motivation

e.g., When I say <u>Transformer</u> in ML, I do not mean the <u>transformer</u> in the movies

e.g., When I say <u>Transformer</u>, I literally mean the <u>transformer</u> in the movies

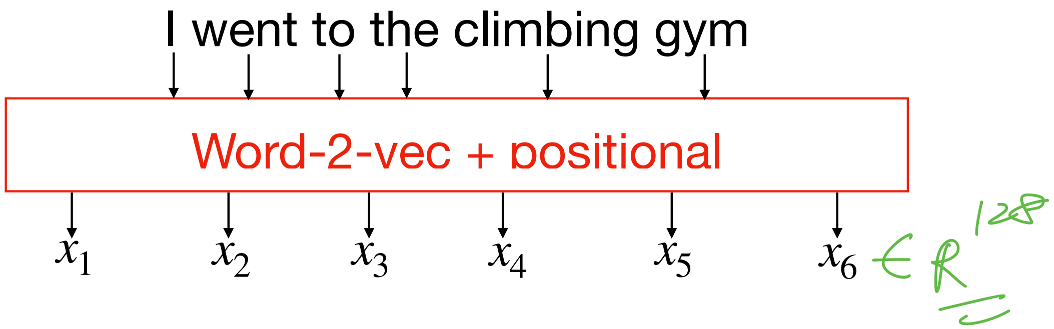
Motivation

e.g., When I say Transformer in ML, I do not mean the transformer in the movies

e.g., When I say Transformer, I literally mean the transformer in the movies

Contextual feature: feature of a word should depend on the context around it

Self-attention



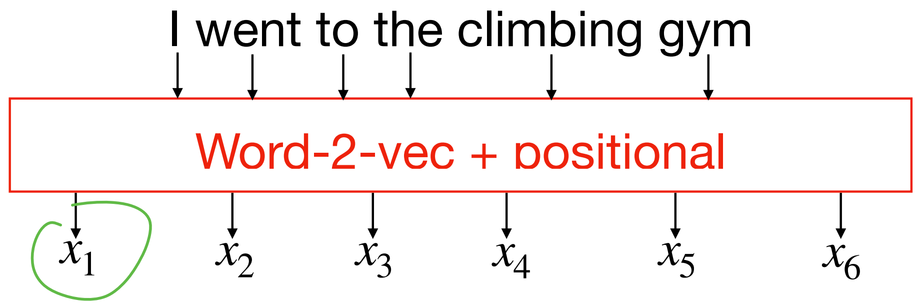
Self-attention

$R^{128\times128}$ $R^{128\times128}$ $R^{128\times128}$

Attention head:
three matrices:

W_q, W_k, W_v

x



Self-attention

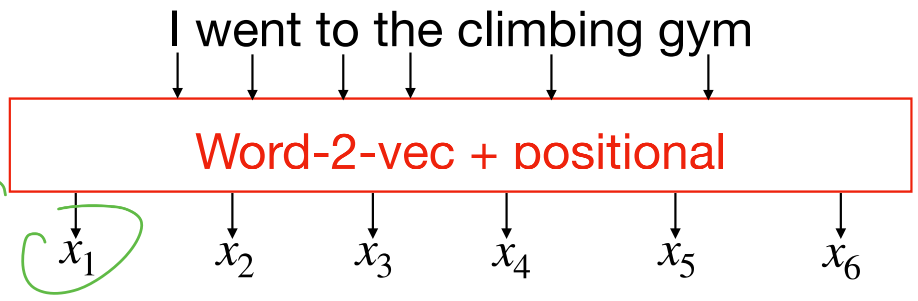
Attention head:
three matrices:

$$W_q, W_k, W_v$$

x $\quad$ x $\quad$ $x \in \mathbb{R}^{128}$

$$q = W_q x \qquad k = W_{\cancel{q}} x \; \leftarrow k \qquad v = W_{\cancel{q}} x \; v$$

Query $\qquad$ key $\qquad$ value



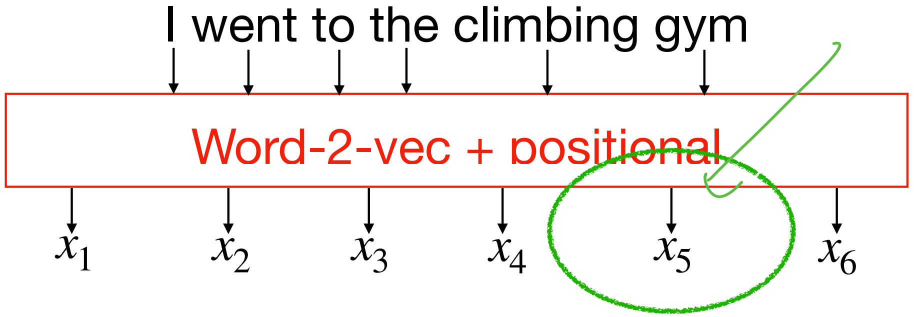
Self-attention

Attention head:
three matrices:

$$W_q, W_k, W_v$$

$q = W_q x$ $\quad$ $k = W_{\overset{k}{q}} x$ $\quad$ $v = W_{\overset{v}{q}} x$

Query $\quad$ key $\quad$ value



Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}
 & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\
q = W_q x & k = W_{\underset{k}{q}} x & v = W_{\underset{v}{q}} x
\end{tikzcd}$$

Query　　　key　　　value

I went to the climbing gym

$$\begin{tikzcd}
\text{I} \arrow[d] & \text{went} \arrow[d] & \text{to} \arrow[d] & \text{the} \arrow[d] & \text{climbing} \arrow[d] & \text{gym} \arrow[d] \\
\boxed{\text{Word-2-vec + positional}} \arrow[d] & \arrow[d] & \arrow[d] & \arrow[d] & \arrow[d] & \arrow[d] \\
x_1 & x_2 & x_3 & x_4 & x_5 & x_6
\end{tikzcd}$$

(q_1, k_1, v_1) — — (q_5, k_5, v_5) (q_6, k_6, v_6)

$q_1 = W_q x_1$

$k_1 = W_k x_1$

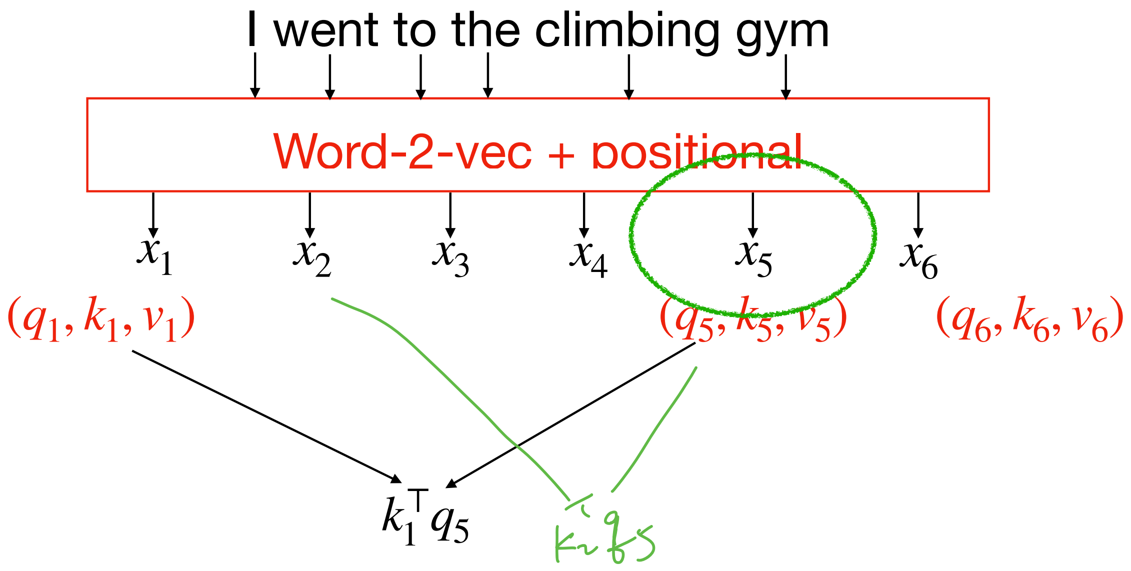
$v_1 = W_v \cdot x_1$

Self-attention

Attention head:
three matrices:

$$\begin{tikzcd} & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\ q = W_q x & k = W_q x & v = W_q x \end{tikzcd}$$

Query key value

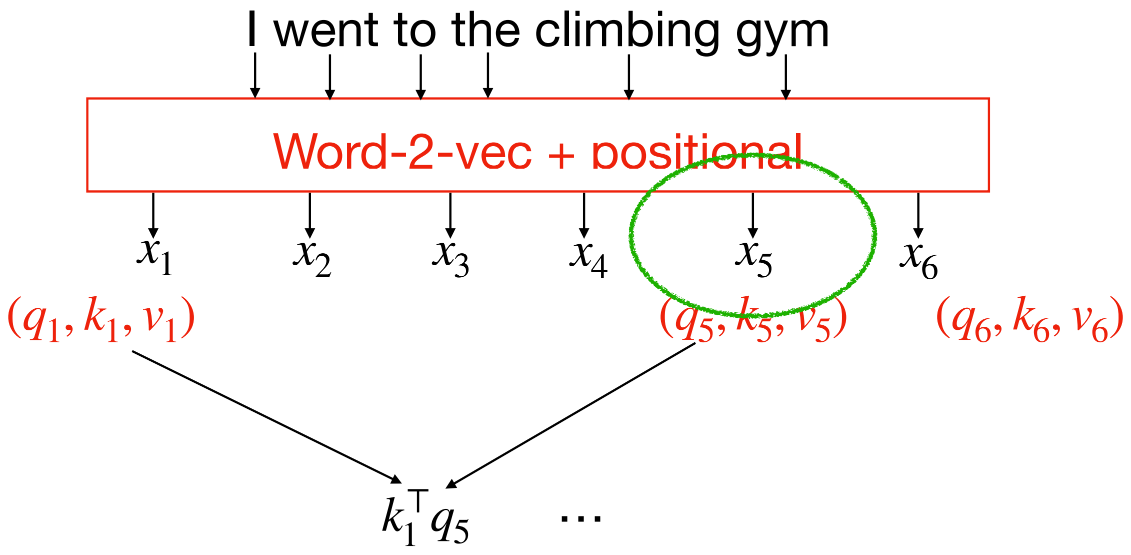


Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}
 & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\
q = W_q x & k = W_q x & v = W_q x
\end{tikzcd}$$

Query key value

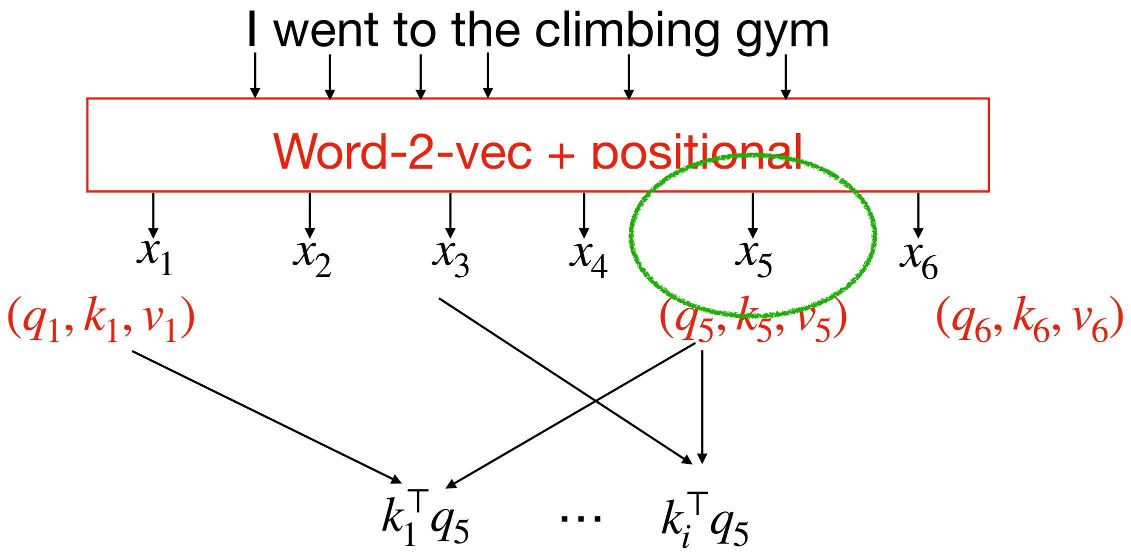


Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}
 & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\
q = W_q x & k = W_q x & v = W_q x
\end{tikzcd}$$

Query key value

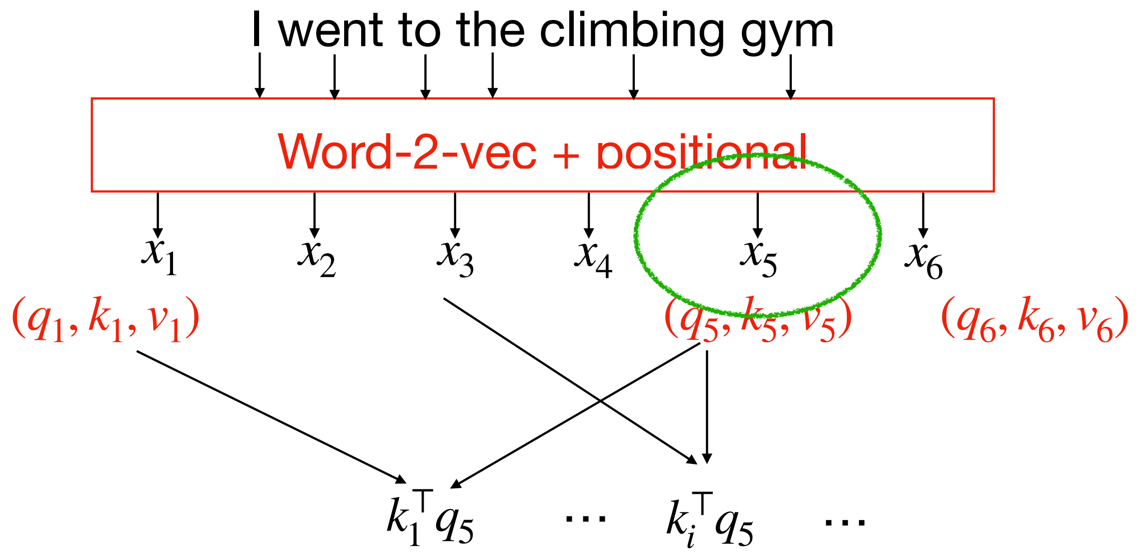

I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
(q_1, k_1, v_1)
(q_5, k_5, v_5)
(q_6, k_6, v_6)
$k_1^\top q_5$ $\cdots$ $k_i^\top q_5$

Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}
 & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\
q = W_q x & k = W_q x & v = W_q x
\end{tikzcd}$$

Query key value

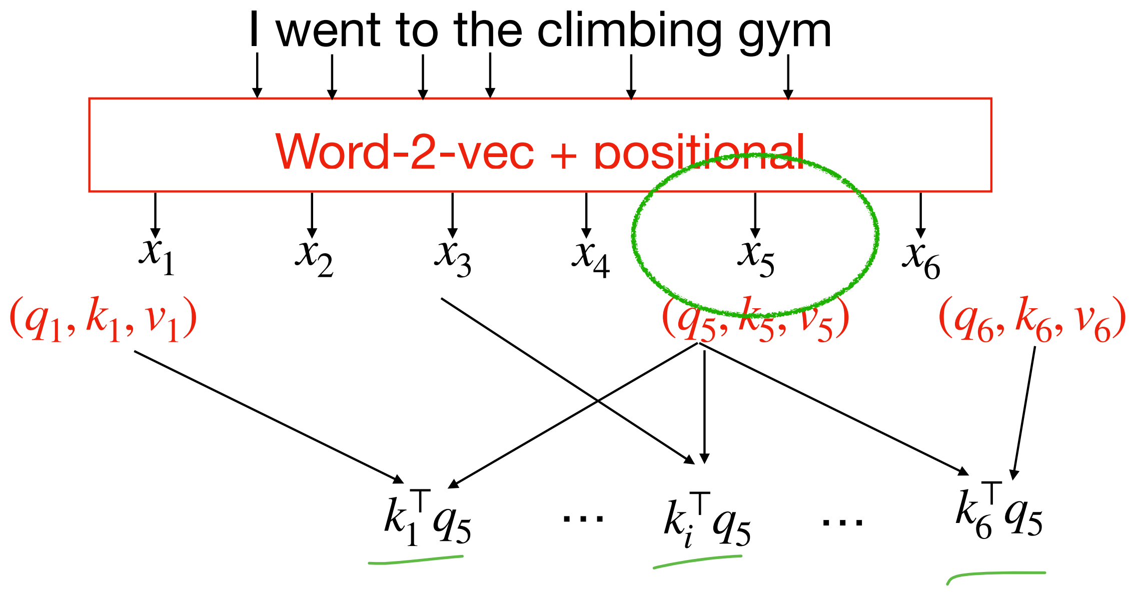


Self-attention

Attention head:
three matrices:

$$\begin{tikzcd} & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\ q = W_q x & k = W_q x & v = W_q x \end{tikzcd}$$

Query key value

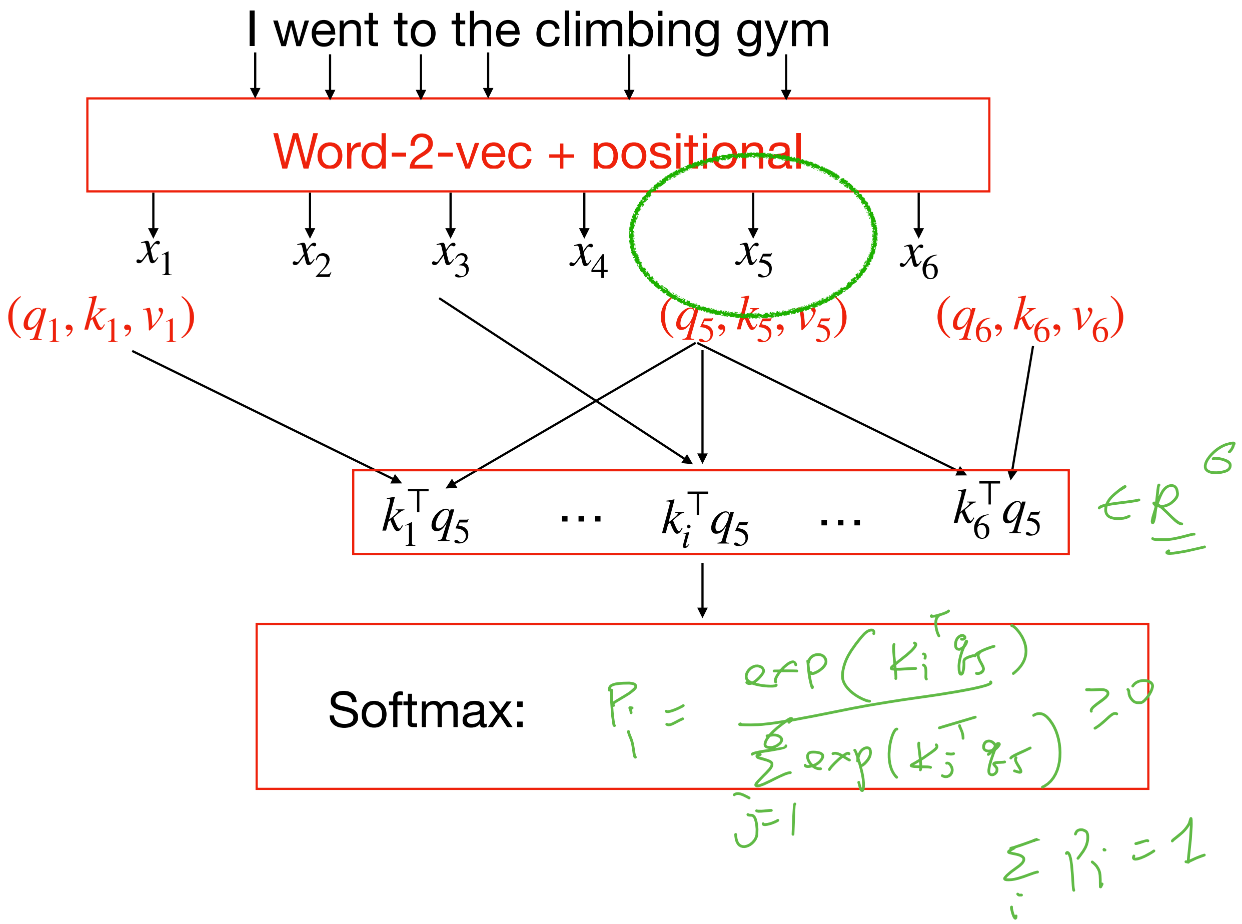


Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}[column sep=small] & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\ q = W_q x & k = W_q x & v = W_q x \end{tikzcd}$$

Query key value


I went to the climbing gym
Word-2-vec + positional
x_1
x_2
x_3
x_4
x_5
x_6
(q_1, k_1, v_1)
(q_5, k_5, v_5)
(q_6, k_6, v_6)
$k_1^\top q_5 \quad \cdots \quad k_i^\top q_5 \quad \cdots \quad k_6^\top q_5$
$\in \mathbb{R}^6$
Softmax:
$p_i = \frac{\exp(k_i^\top q_5)}{\sum_{j=1}^{6} \exp(k_j^\top q_5)} \geq 0$
$\sum_i p_i = 1$

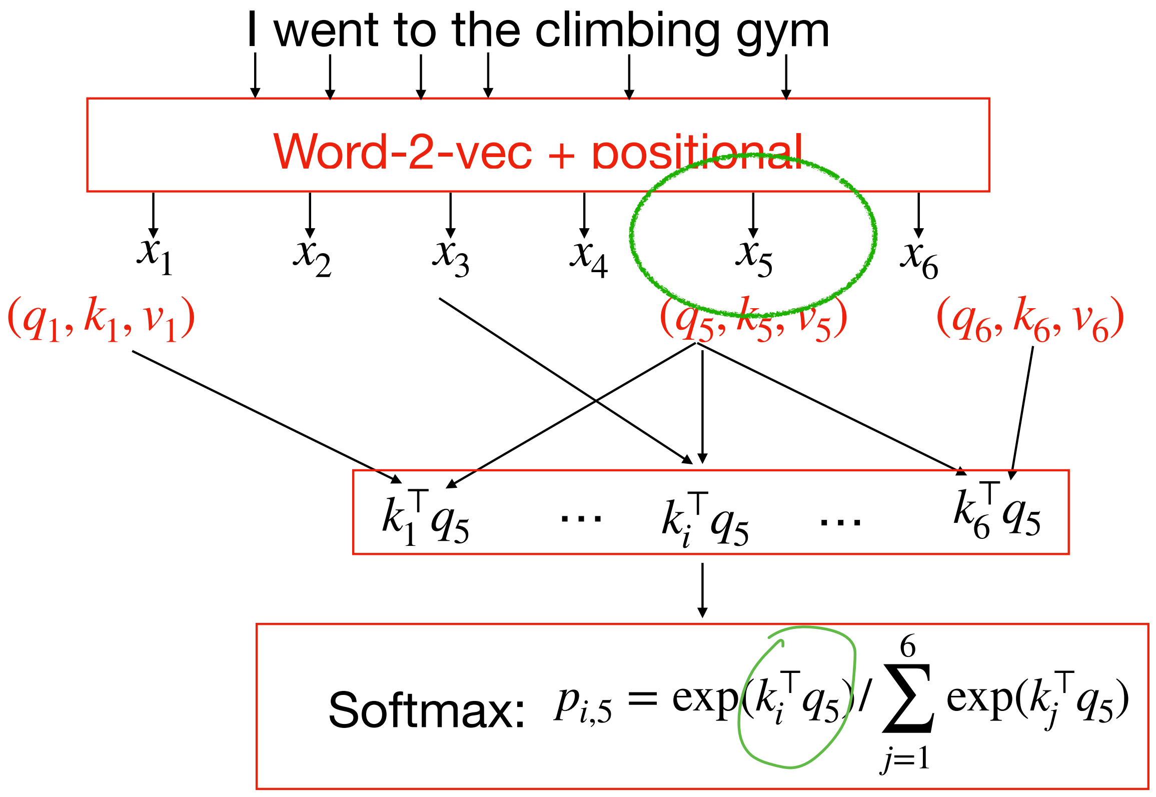
Self-attention

Attention head:
three matrices:

$$W_q, W_k, W_v$$

$$q = W_q x \qquad k = W_q x \qquad v = W_q x$$

Query key value



I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
(q_1, k_1, v_1)
(q_5, k_5, v_5)
(q_6, k_6, v_6)
$k_1^\top q_5 \quad \cdots \quad k_i^\top q_5 \quad \cdots \quad k_6^\top q_5$
Softmax: $p_{i,5} = \exp(k_i^\top q_5) / \sum_{j=1}^{6} \exp(k_j^\top q_5)$

$k_i^\top q_5 \to +\infty$

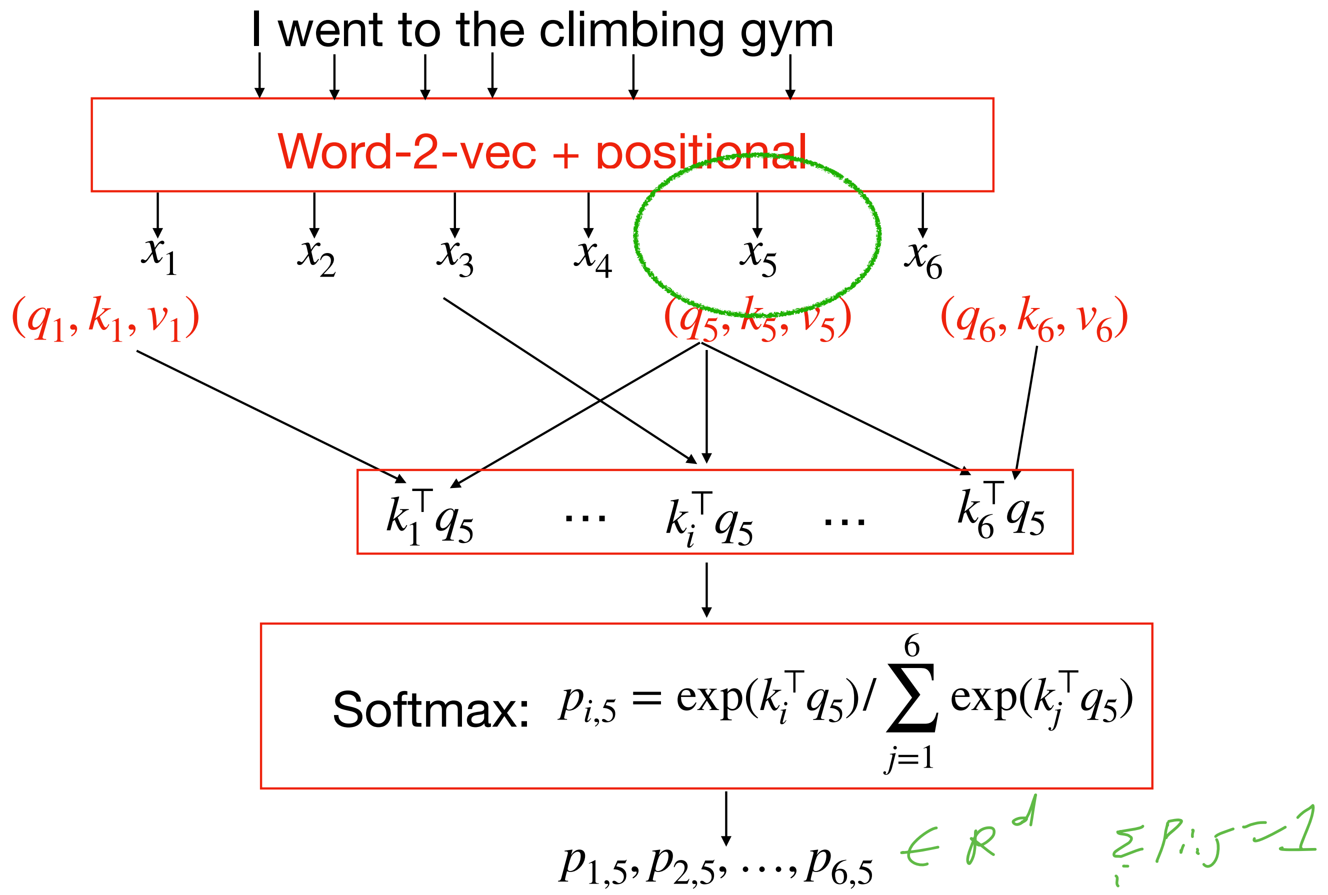
$p_{i,5} \to 1$

Self-attention

Attention head:
three matrices:

$$\begin{tikzcd}
 & W_q, W_k, W_v \arrow[dl] \arrow[d] \arrow[dr] & \\
q = W_q x & k = W_q x & v = W_q x
\end{tikzcd}$$

Query key value


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
(q_1, k_1, v_1)
(q_5, k_5, v_5)
(q_6, k_6, v_6)
$k_1^\top q_5 \quad \cdots \quad k_i^\top q_5 \quad \cdots \quad k_6^\top q_5$
Softmax: $p_{i,5} = \exp(k_i^\top q_5) / \sum_{j=1}^{6} \exp(k_j^\top q_5)$
$p_{1,5}, p_{2,5}, \ldots, p_{6,5}$
$\in \mathbb{R}^d$
$\sum_i p_{i,5} = 1$

Self-attention

Attention head:
three matrices:

$$W_q, W_k, W_v$$

$q = W_q x$ $\quad$ $k = W_q x$ $\quad$ $v = W_q x$

Query $\quad$ key $\quad$ value

I went to the climbing gym

Word-2-vec + positional

$x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6$

$(q_1, k_1, v_1) \quad (q_5, k_5, v_5) \quad (q_6, k_6, v_6)$

$k_1^\top q_5 \quad \cdots \quad k_i^\top q_5 \quad \cdots \quad k_6^\top q_5$

Softmax: $p_{i,5} = \exp(k_i^\top q_5) / \sum_{j=1}^{6} \exp(k_j^\top q_5)$

$p_{1,5}, p_{2,5}, \ldots, p_{6,5}$

$$x_5' = p_{1,5} v_1 + p_{2,5} v_2 + \ldots + p_{6,5} v_6$$

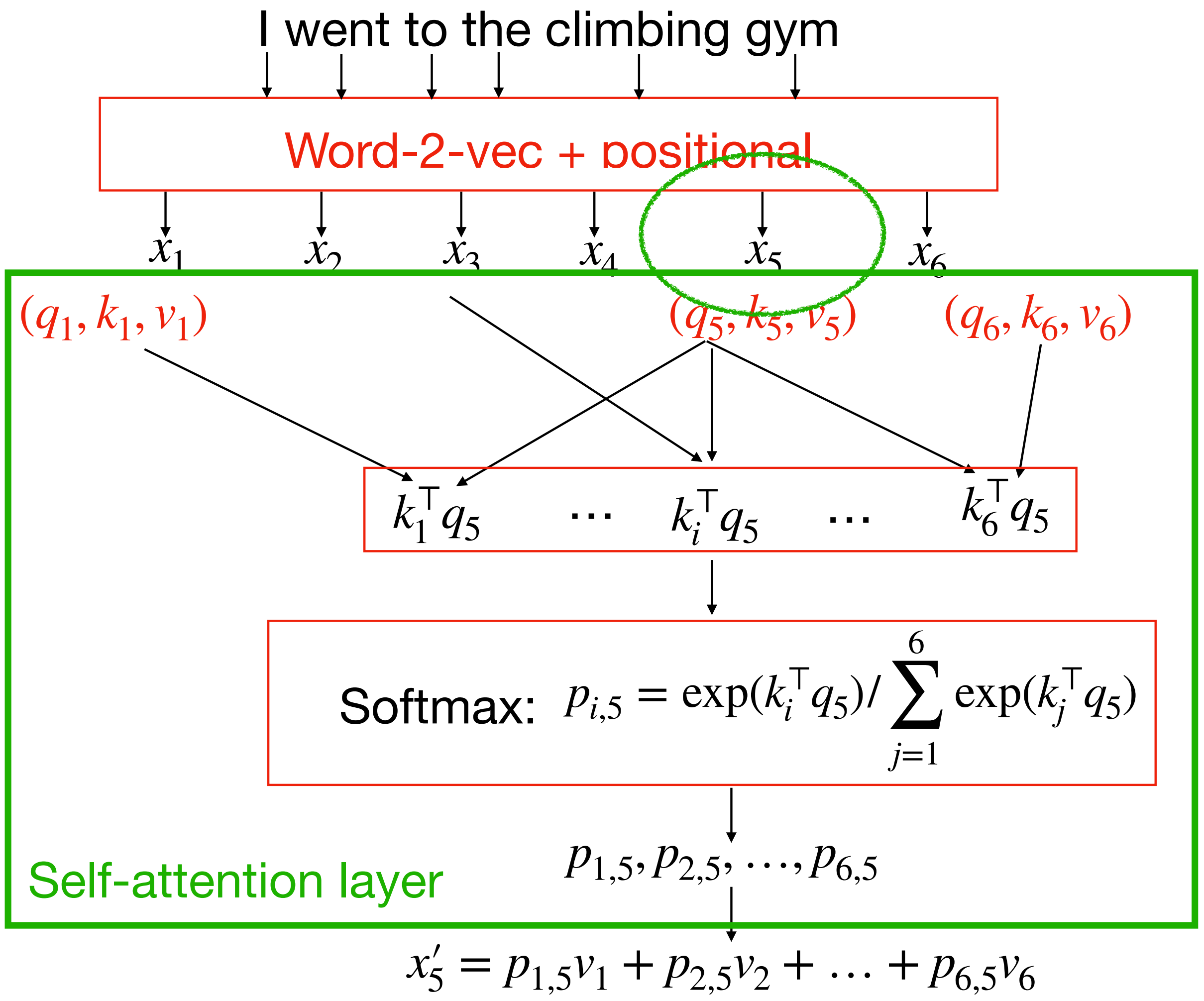
Self-attention

Attention head:
three matrices:

$$W_q, W_k, W_v$$

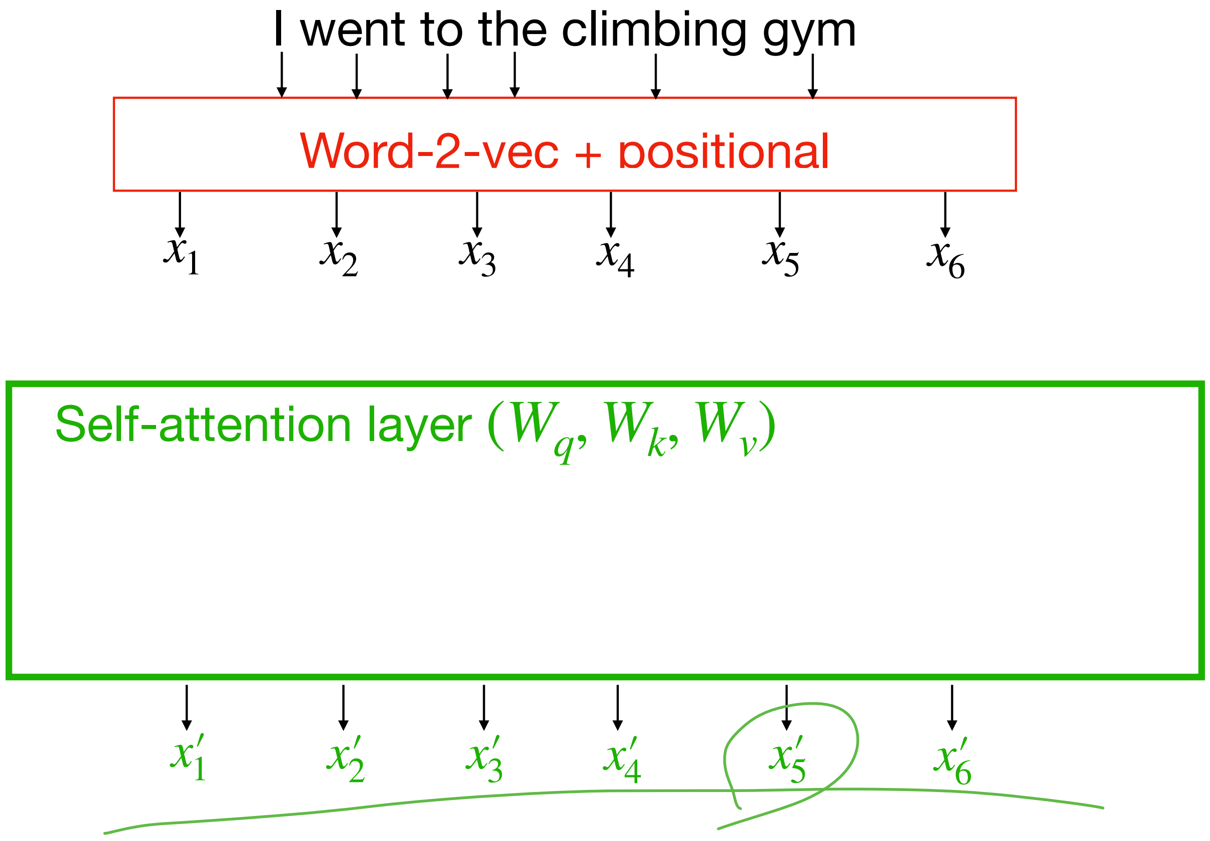
$q = W_q x$ $\quad k = W_q x$ $\quad v = W_q x$

Query $\qquad$ key $\qquad$ value



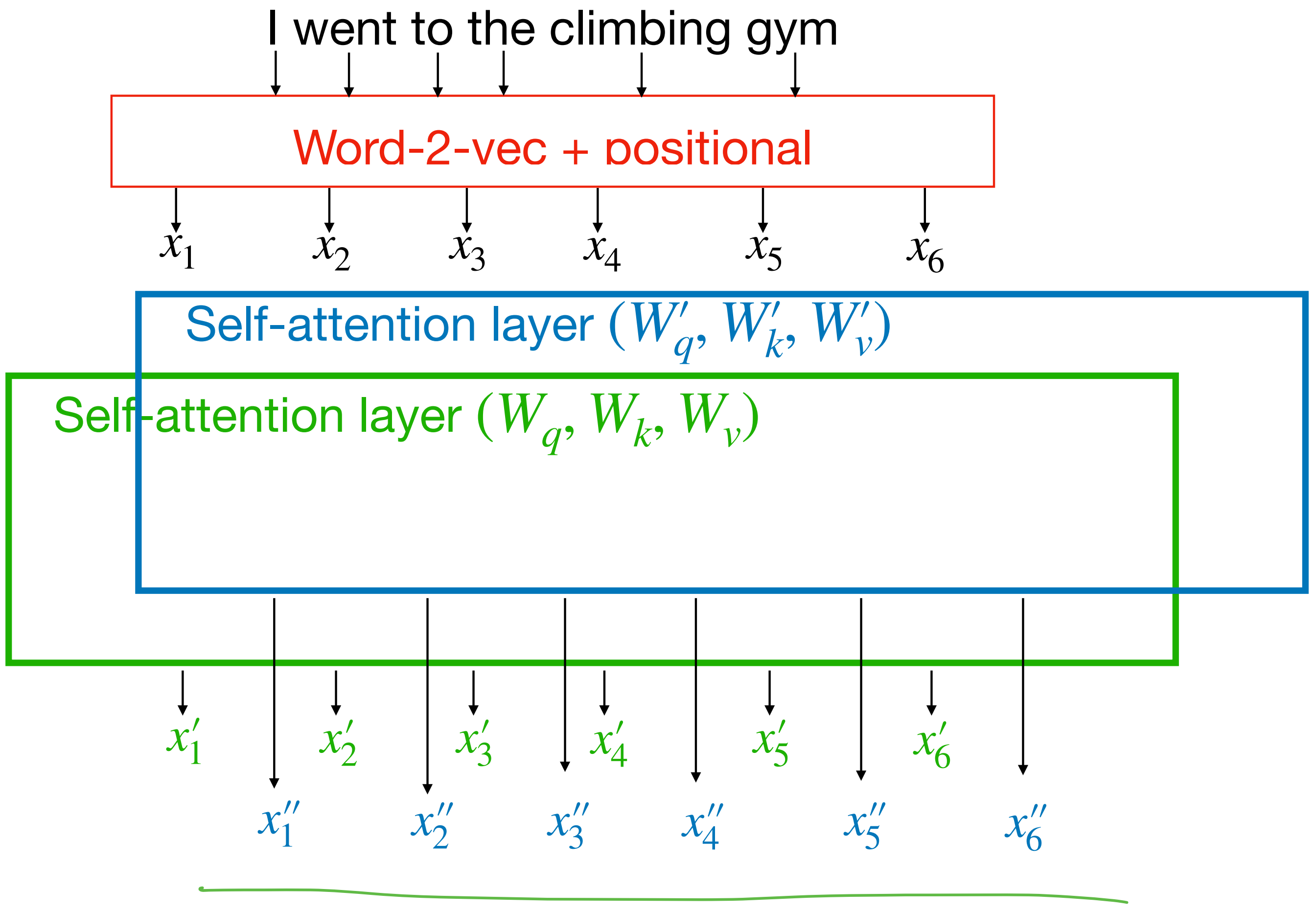
I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
(q_1, k_1, v_1)
(q_5, k_5, v_5)
(q_6, k_6, v_6)
$k_1^\top q_5 \quad \ldots \quad k_i^\top q_5 \quad \ldots \quad k_6^\top q_5$
Softmax: $p_{i,5} = \exp(k_i^\top q_5) / \sum_{j=1}^{6} \exp(k_j^\top q_5)$
$p_{1,5}, p_{2,5}, \ldots, p_{6,5}$
Self-attention layer
$x'_5 = p_{1,5} v_1 + p_{2,5} v_2 + \ldots + p_{6,5} v_6$

Multi-head self-attention

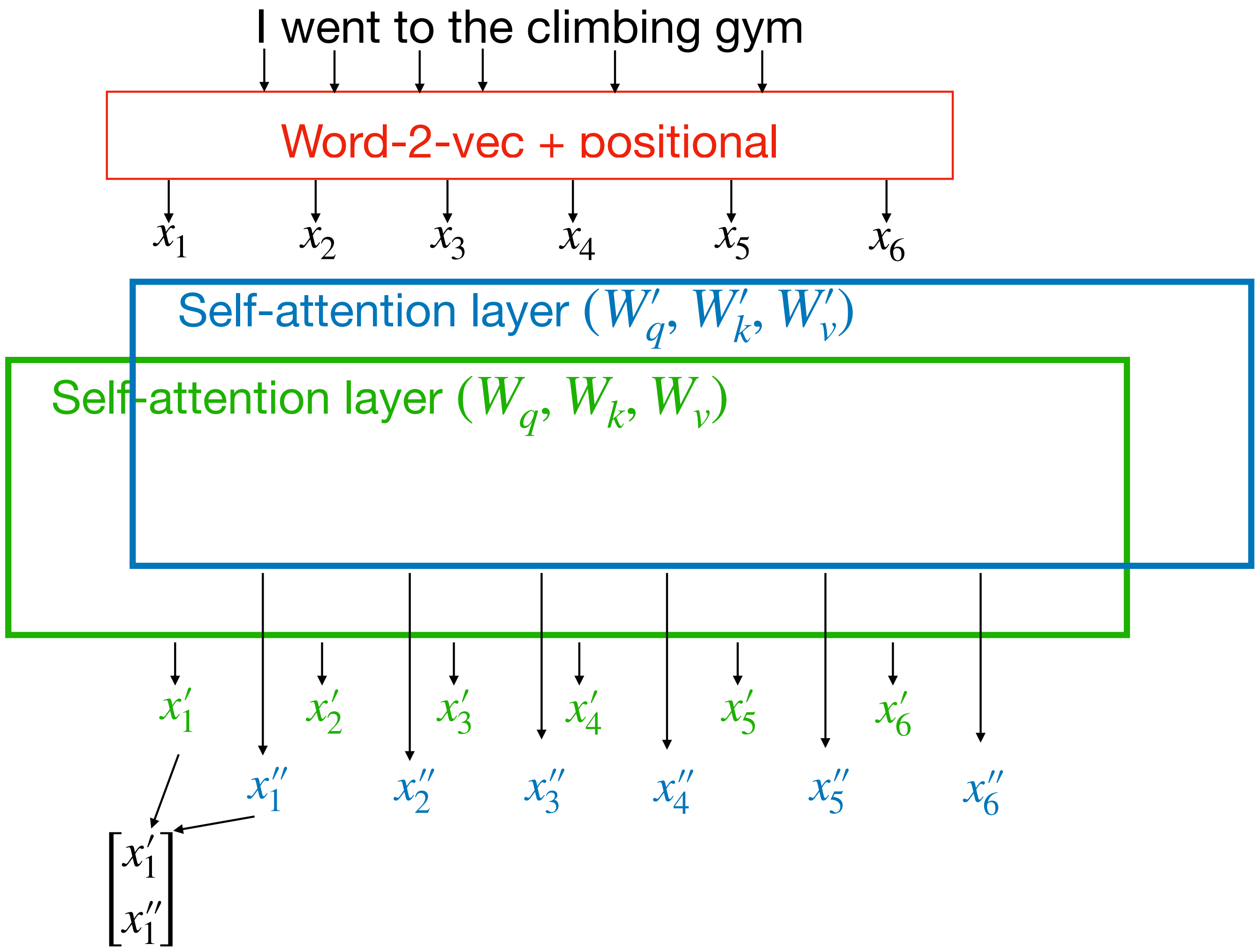


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Self-attention layer (W_q, W_k, W_v)
x_1' x_2' x_3' x_4' x_5' x_6'

Multi-head self-attention

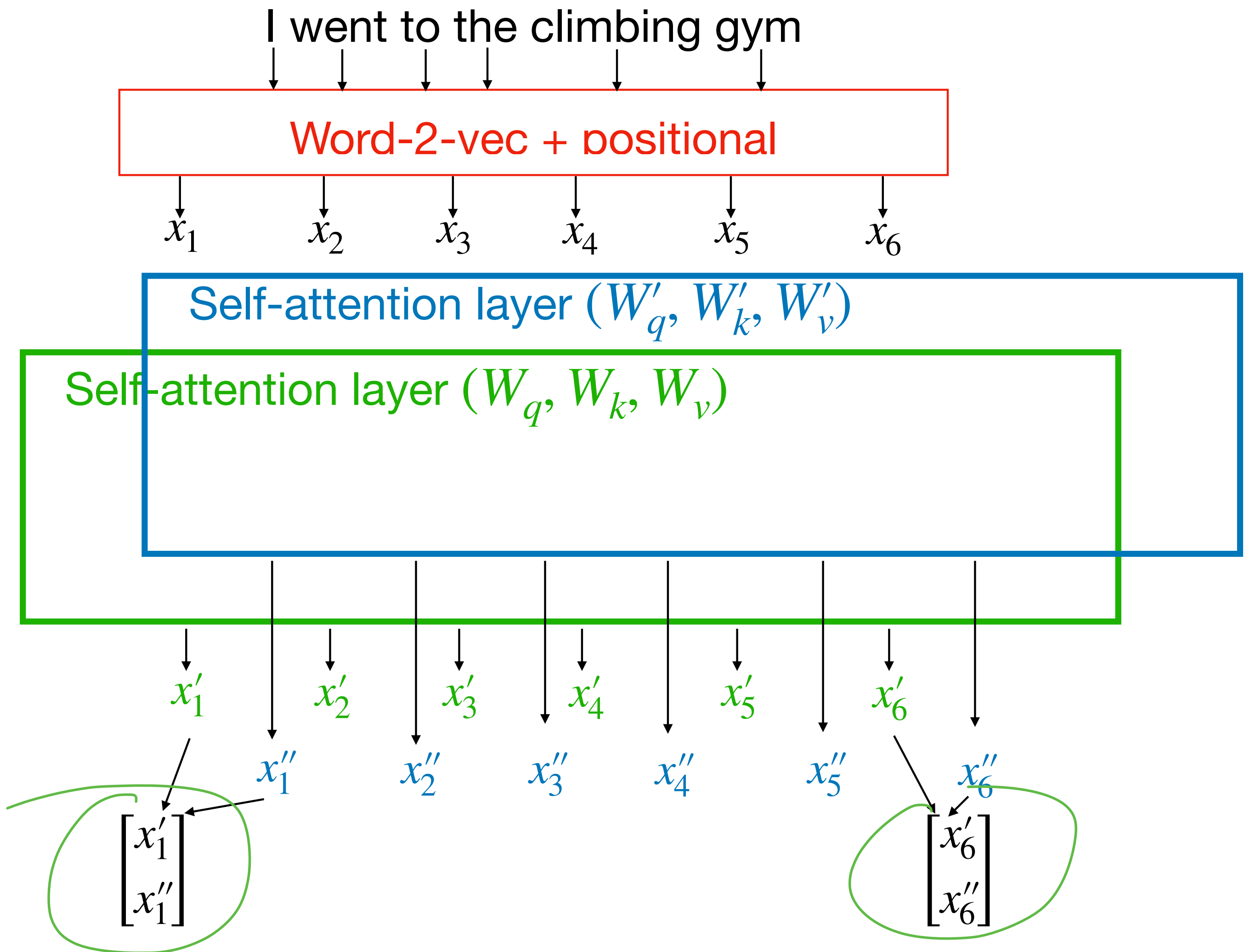

I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Self-attention layer (W'_q, W'_k, W'_v)
Self-attention layer (W_q, W_k, W_v)
x'_1 x'_2 x'_3 x'_4 x'_5 x'_6
x''_1 x''_2 x''_3 x''_4 x''_5 x''_6

Multi-head self-attention



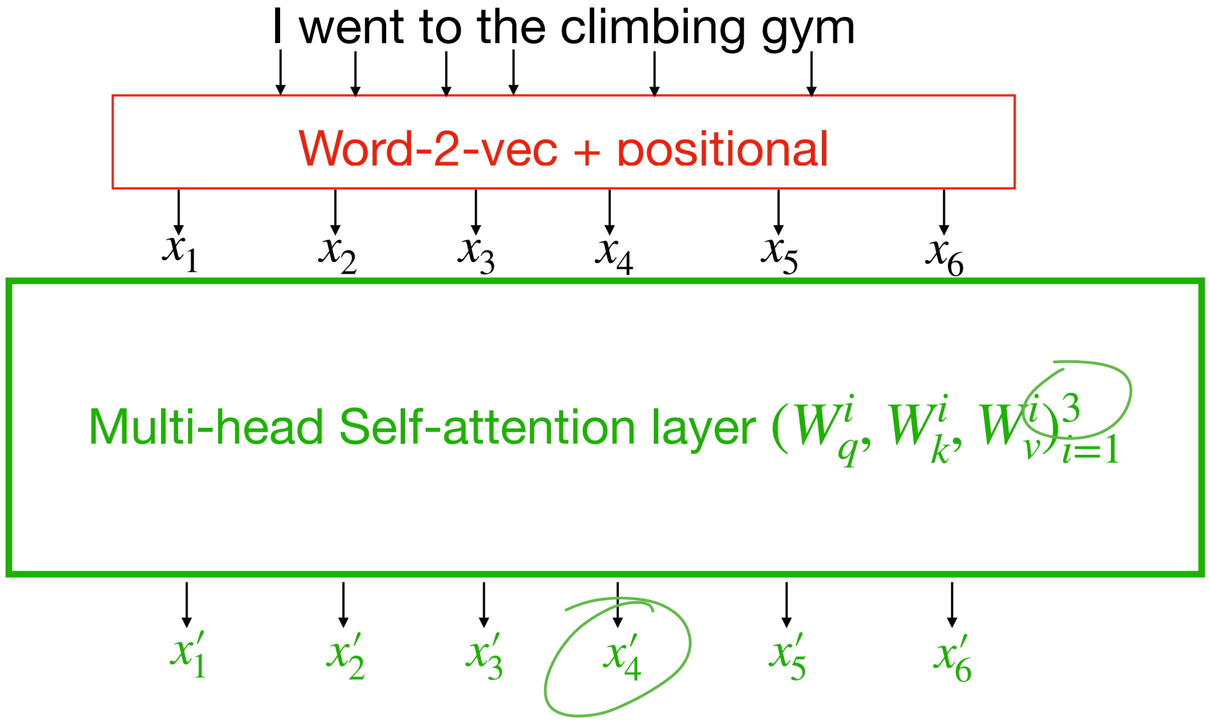
I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Self-attention layer (W'_q, W'_k, W'_v)
Self-attention layer (W_q, W_k, W_v)
x'_1 x'_2 x'_3 x'_4 x'_5 x'_6
x''_1 x''_2 x''_3 x''_4 x''_5 x''_6
$\begin{bmatrix} x'_1 \\ x''_1 \end{bmatrix}$

Multi-head self-attention



I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Self-attention layer (W'_q, W'_k, W'_v)
Self-attention layer (W_q, W_k, W_v)
x'_1 x'_2 x'_3 x'_4 x'_5 x'_6
x''_1 x''_2 x''_3 x''_4 x''_5 x''_6
$\begin{bmatrix} x'_1 \\ x''_1 \end{bmatrix}$
$\begin{bmatrix} x'_6 \\ x''_6 \end{bmatrix}$

Summary so far


I went to the climbing gym
Word-2-vec + positional
x_1
x_2
x_3
x_4
x_5
x_6
Multi-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
x'_1
x'_2
x'_3
x'_4
x'_5
x'_6

Contextual features: e.g., x'_4 encodes information from all words

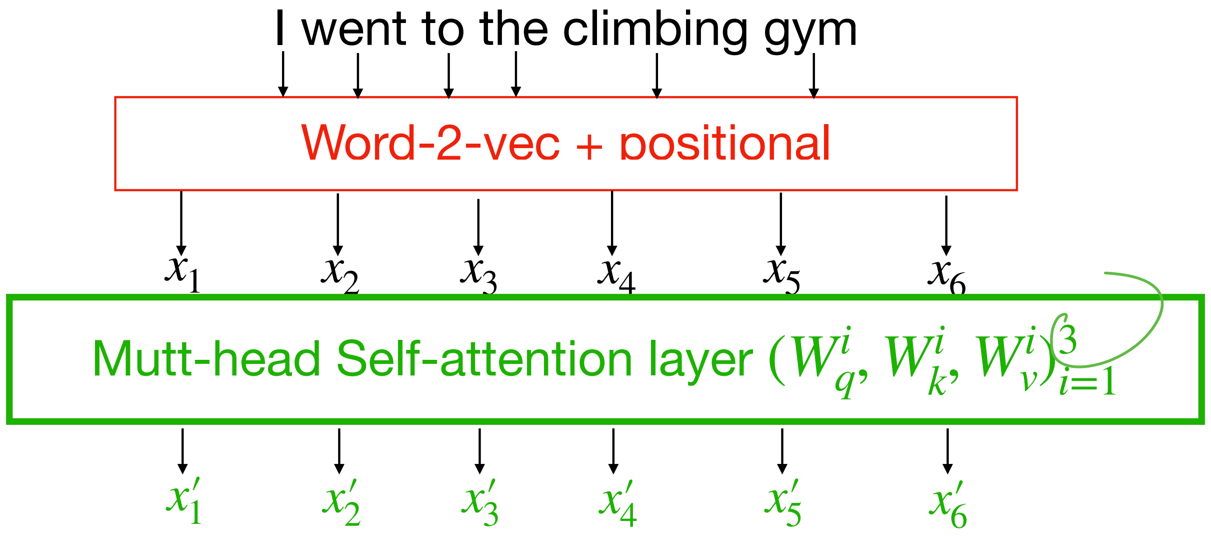
Outline today

1. Word-2-Vec embedding and positional embedding

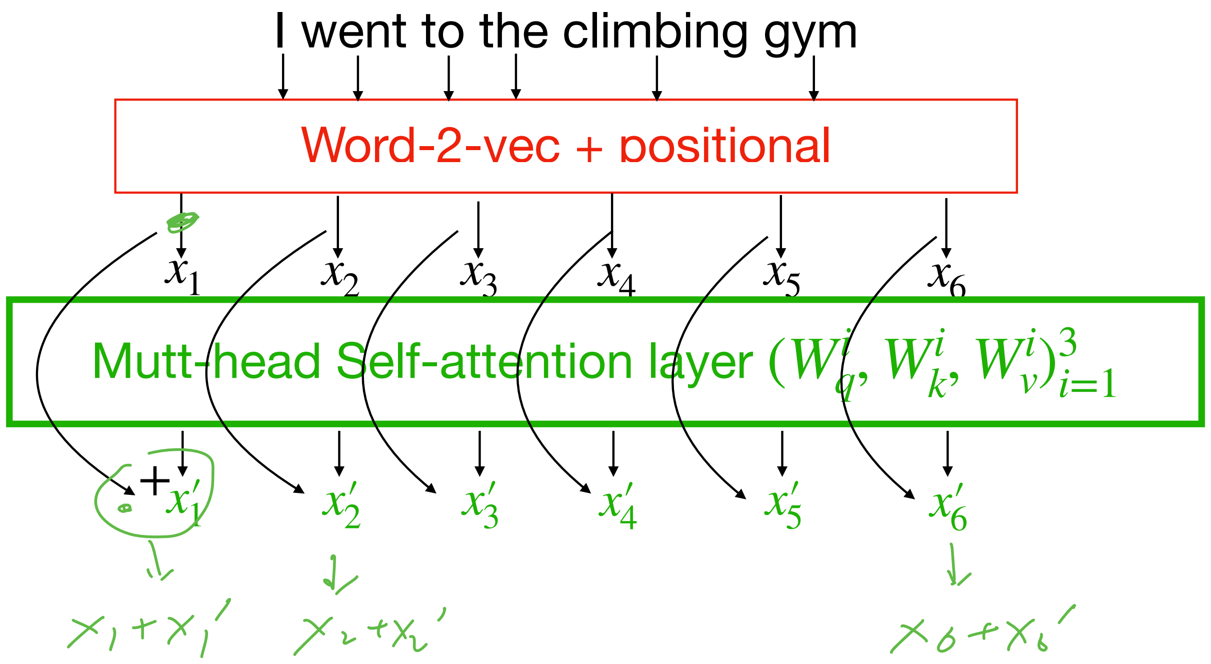
2. Attention model

3. Putting things together: the Transformer model

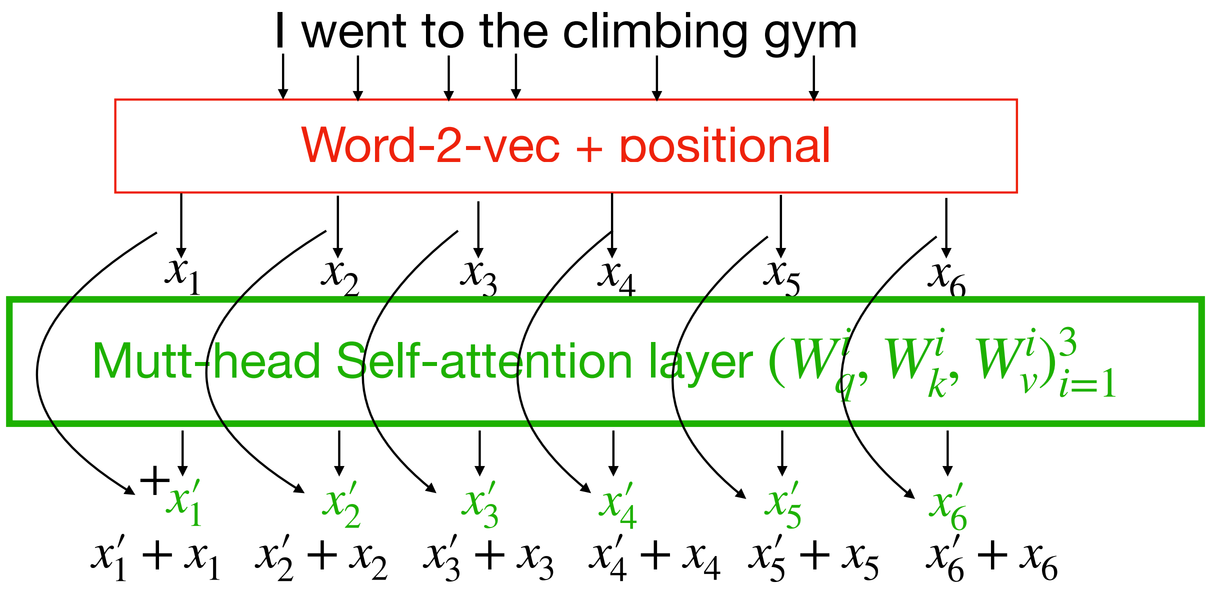
The Transformer model: encoder


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
x'_1 x'_2 x'_3 x'_4 x'_5 x'_6

The Transformer model: encoder

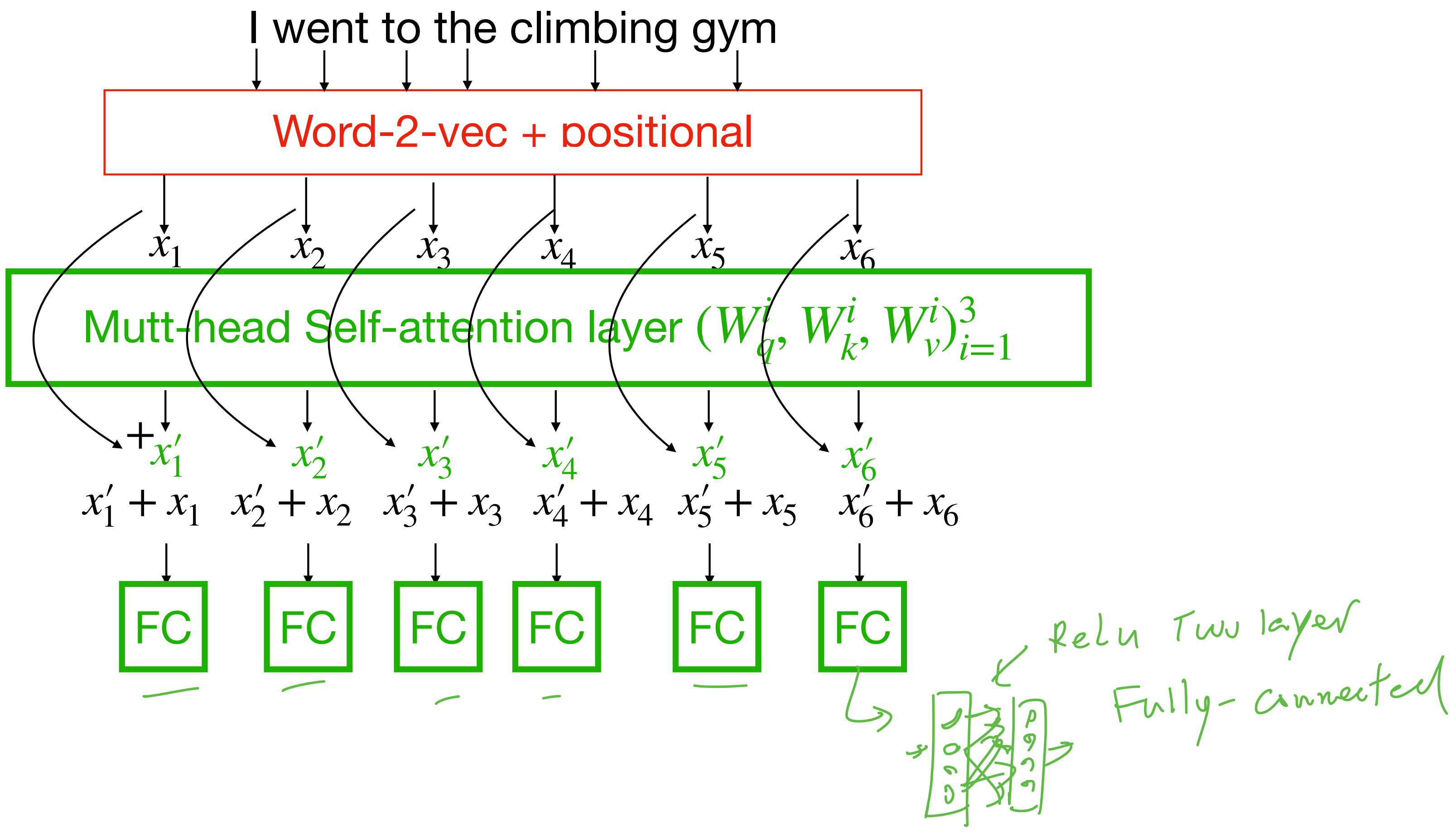

I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1 + x_1'$
$x_2 + x_2'$
$x_6 + x_6'$

The Transformer model: encoder

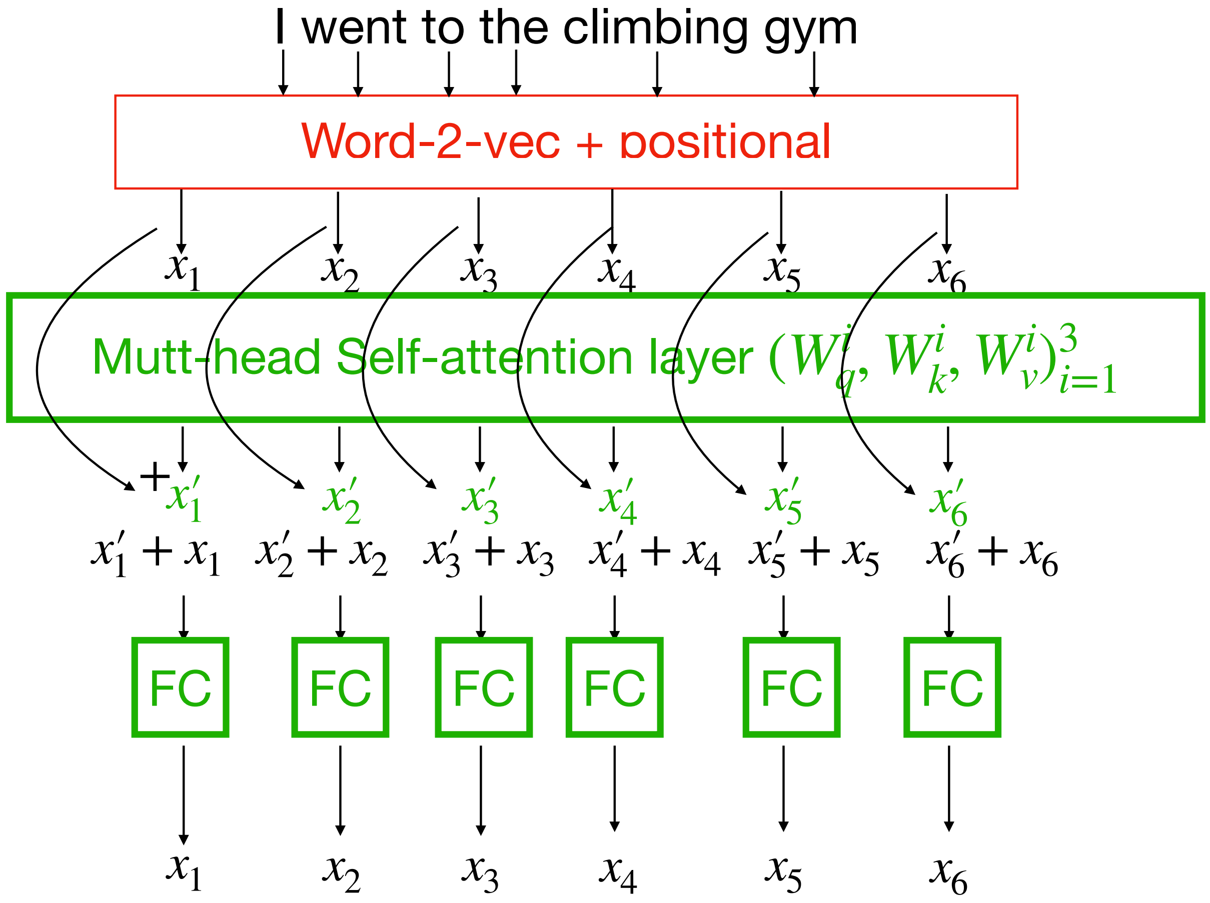


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1' + x_1$ $x_2' + x_2$ $x_3' + x_3$ $x_4' + x_4$ $x_5' + x_5$ $x_6' + x_6$

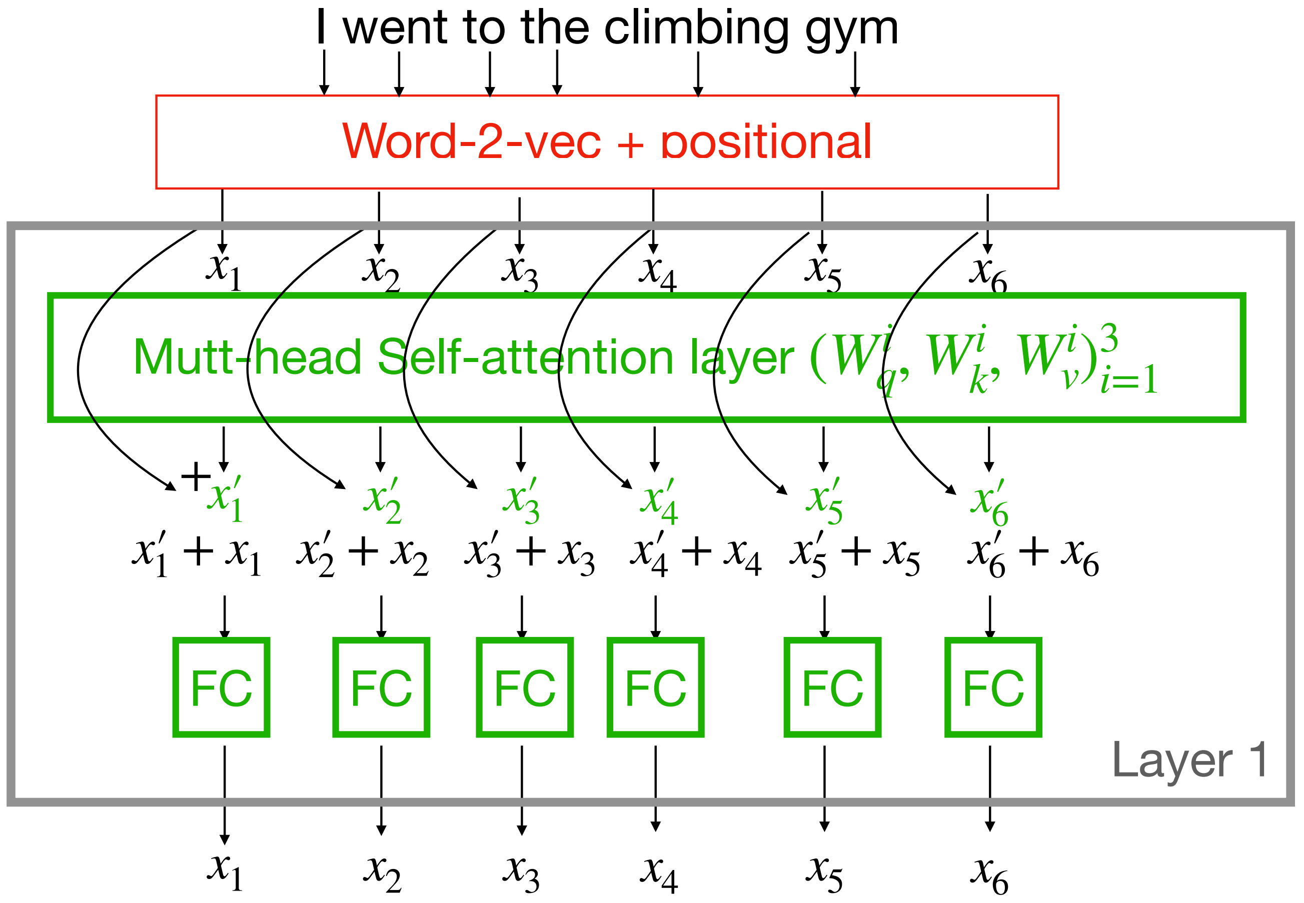
The Transformer model: encoder


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1' + x_1$ $x_2' + x_2$ $x_3' + x_3$ $x_4' + x_4$ $x_5' + x_5$ $x_6' + x_6$
FC FC FC FC FC FC
Relu Two layer
Fully-connected

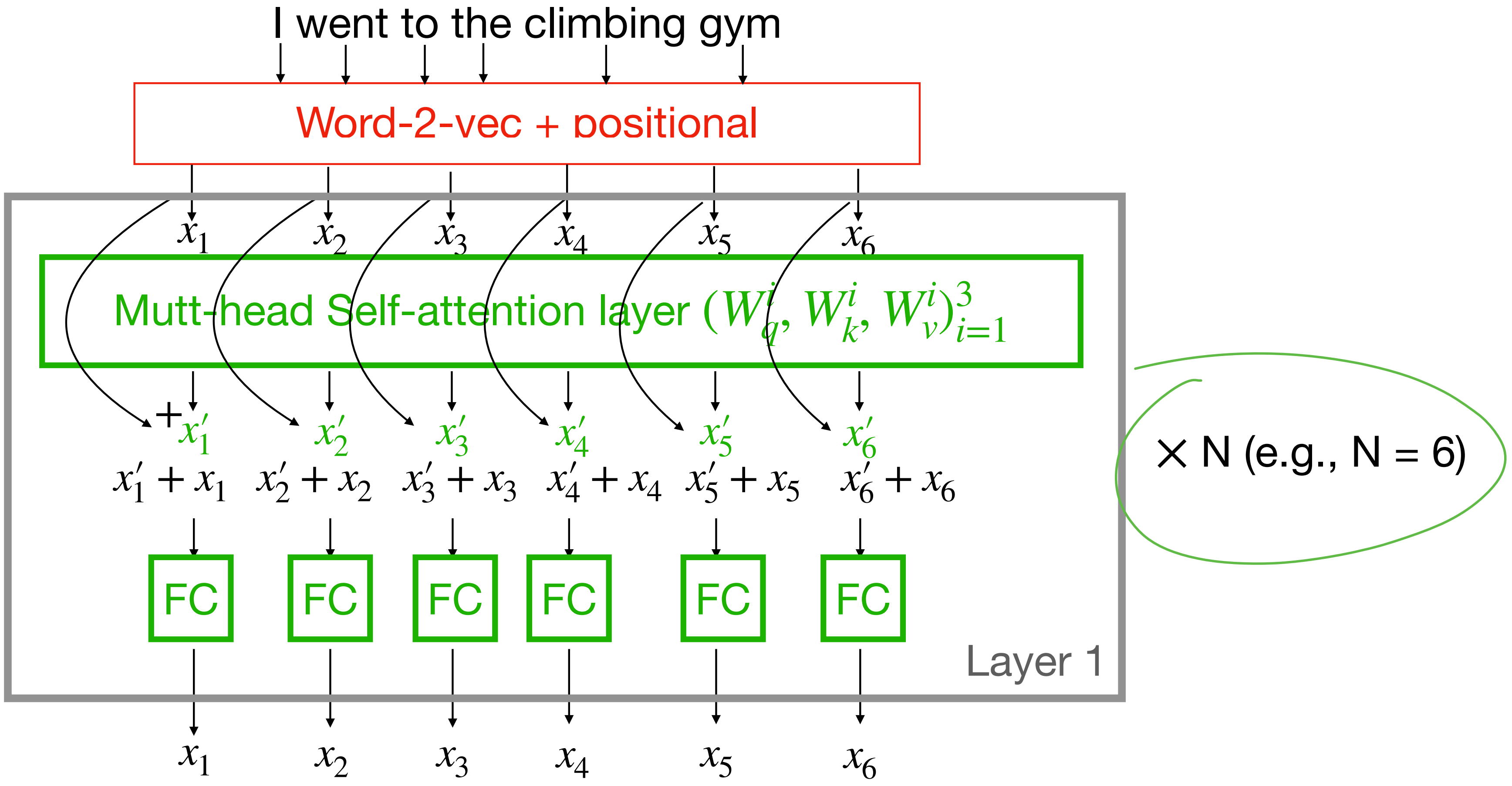
The Transformer model: encoder


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1' + x_1$ $x_2' + x_2$ $x_3' + x_3$ $x_4' + x_4$ $x_5' + x_5$ $x_6' + x_6$
FC FC FC FC FC FC
x_1 x_2 x_3 x_4 x_5 x_6

The Transformer model: encoder

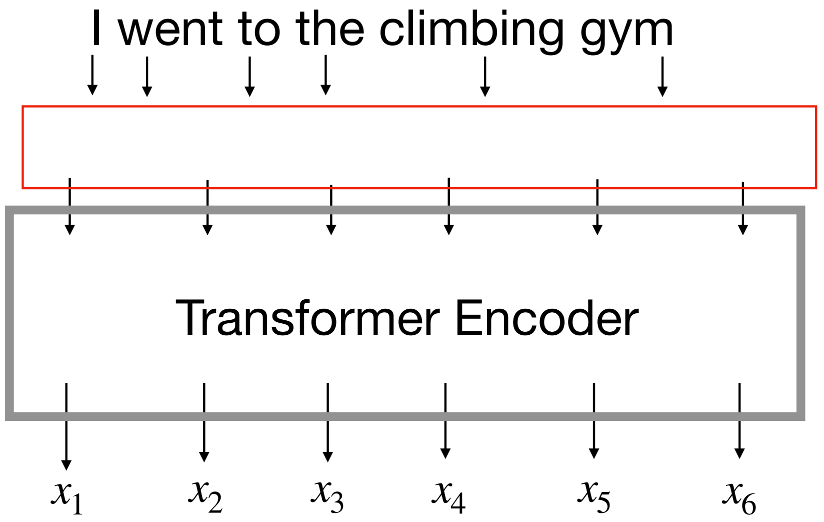

I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1' + x_1$ $x_2' + x_2$ $x_3' + x_3$ $x_4' + x_4$ $x_5' + x_5$ $x_6' + x_6$
FC FC FC FC FC FC
Layer 1
x_1 x_2 x_3 x_4 x_5 x_6

The Transformer model: encoder

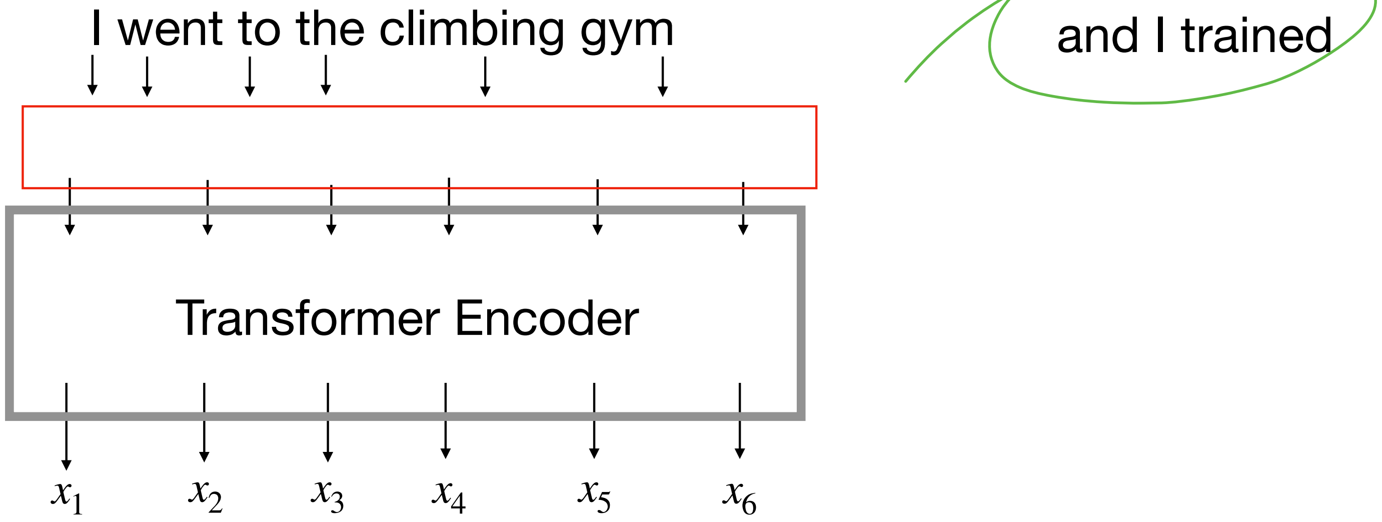


I went to the climbing gym
Word-2-vec + positional
x_1 x_2 x_3 x_4 x_5 x_6
Mutt-head Self-attention layer $(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+
x_1' x_2' x_3' x_4' x_5' x_6'
$x_1' + x_1$ $x_2' + x_2$ $x_3' + x_3$ $x_4' + x_4$ $x_5' + x_5$ $x_6' + x_6$
FC FC FC FC FC FC
Layer 1
x_1 x_2 x_3 x_4 x_5 x_6
× N (e.g., N = 6)

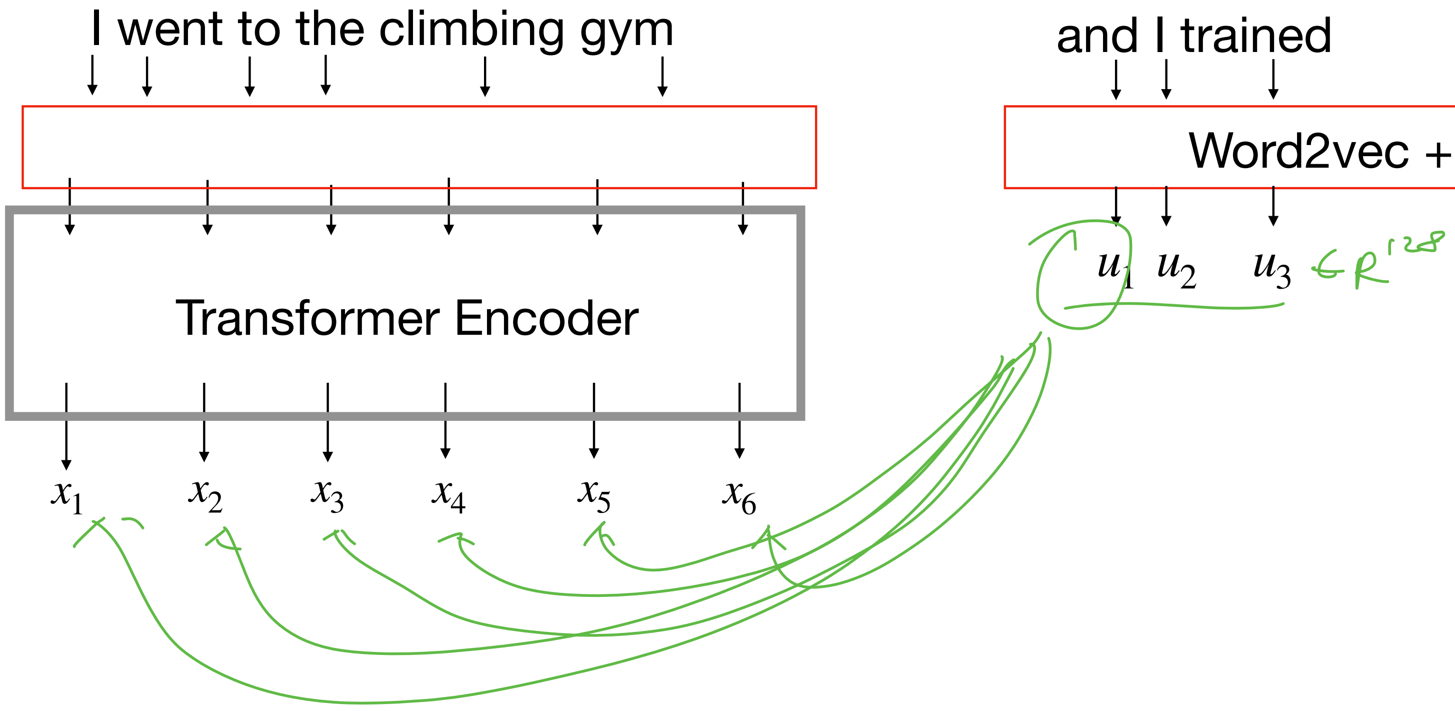
The Transformer model: decoder



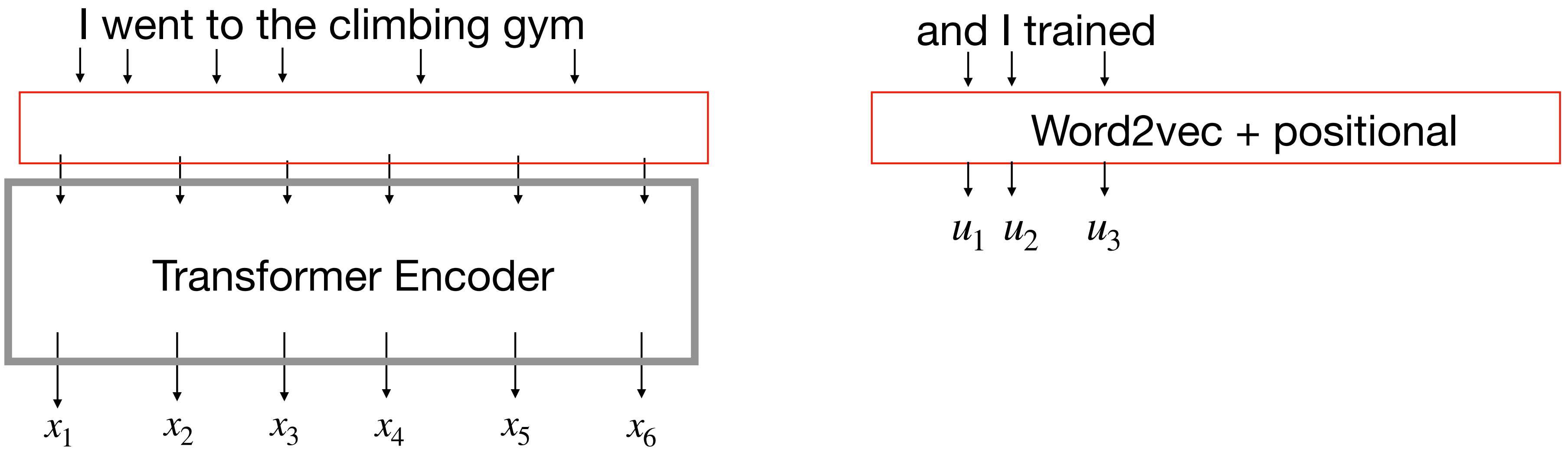
The Transformer model: decoder



The Transformer model: decoder

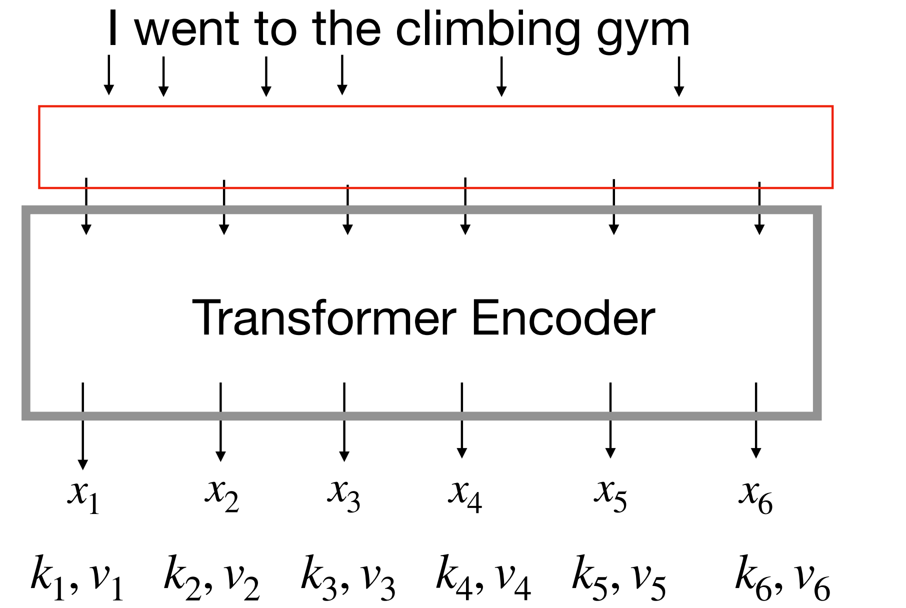

I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
u_1 u_2 u_3 $\in \mathbb{R}^{128}$

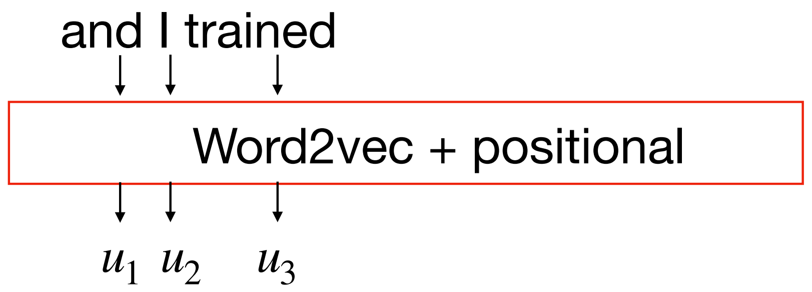
The Transformer model: decoder



cross-attention (W_q, W_k, W_v)

The Transformer model: decoder

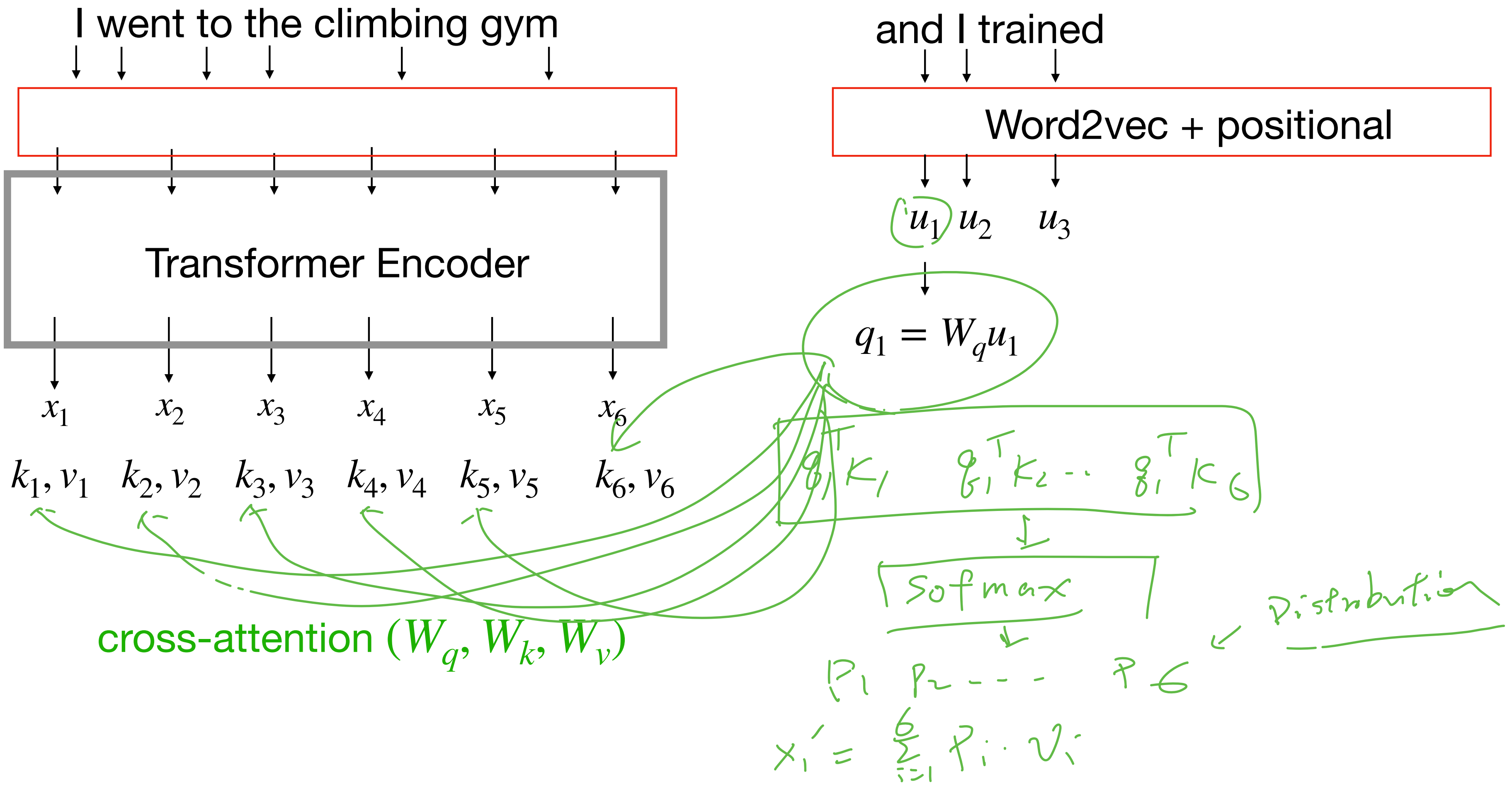

I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6



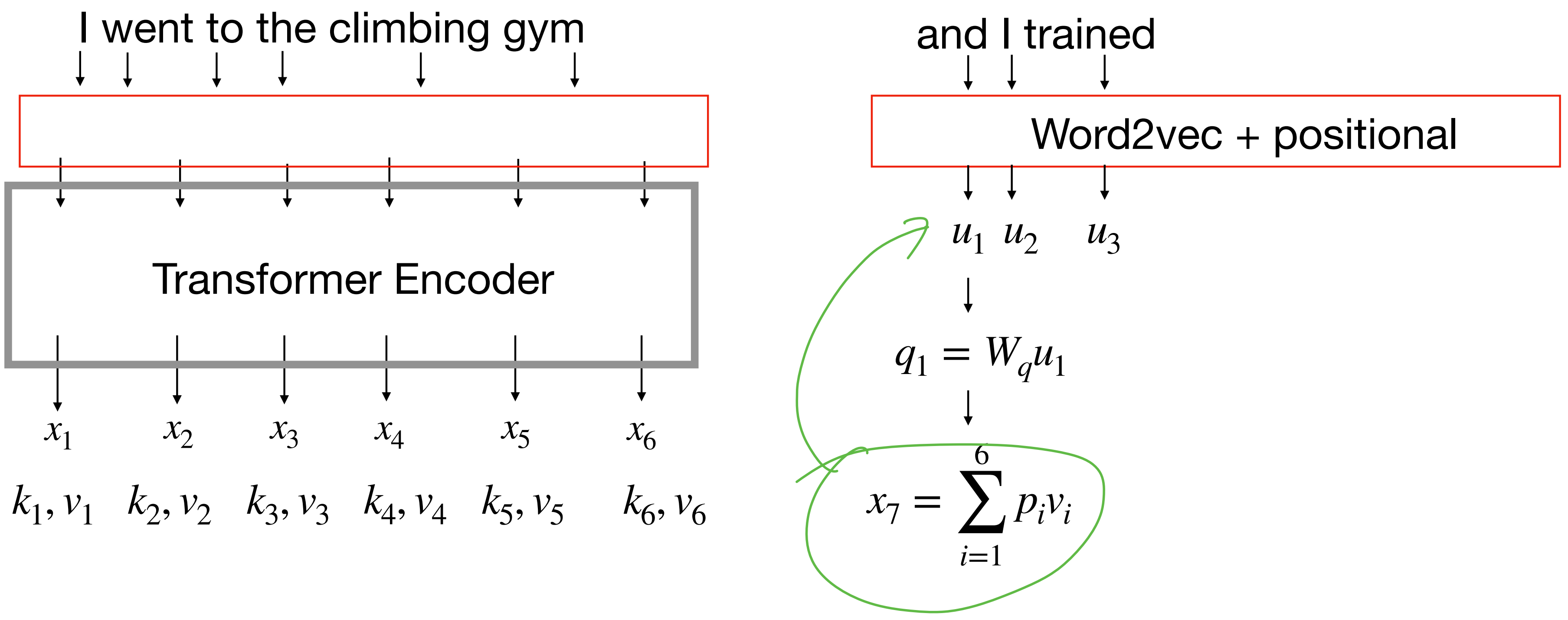
$K_1 = W_K \cdot x_1 \quad V_1 = W_V \cdot x_1$

cross-attention (W_q, W_k, W_v)

The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
cross-attention (W_q, W_k, W_v)
and I trained
Word2vec + positional
u_1 u_2 u_3
$q_1 = W_q u_1$
$q_1^T k_1$ $q_1^T k_2 \cdots$ $q_1^T k_6$
Softmax
p_1 p_2 - - - p_6
Distribution
$x_1' = \sum_{i=1}^{6} p_i \cdot v_i$

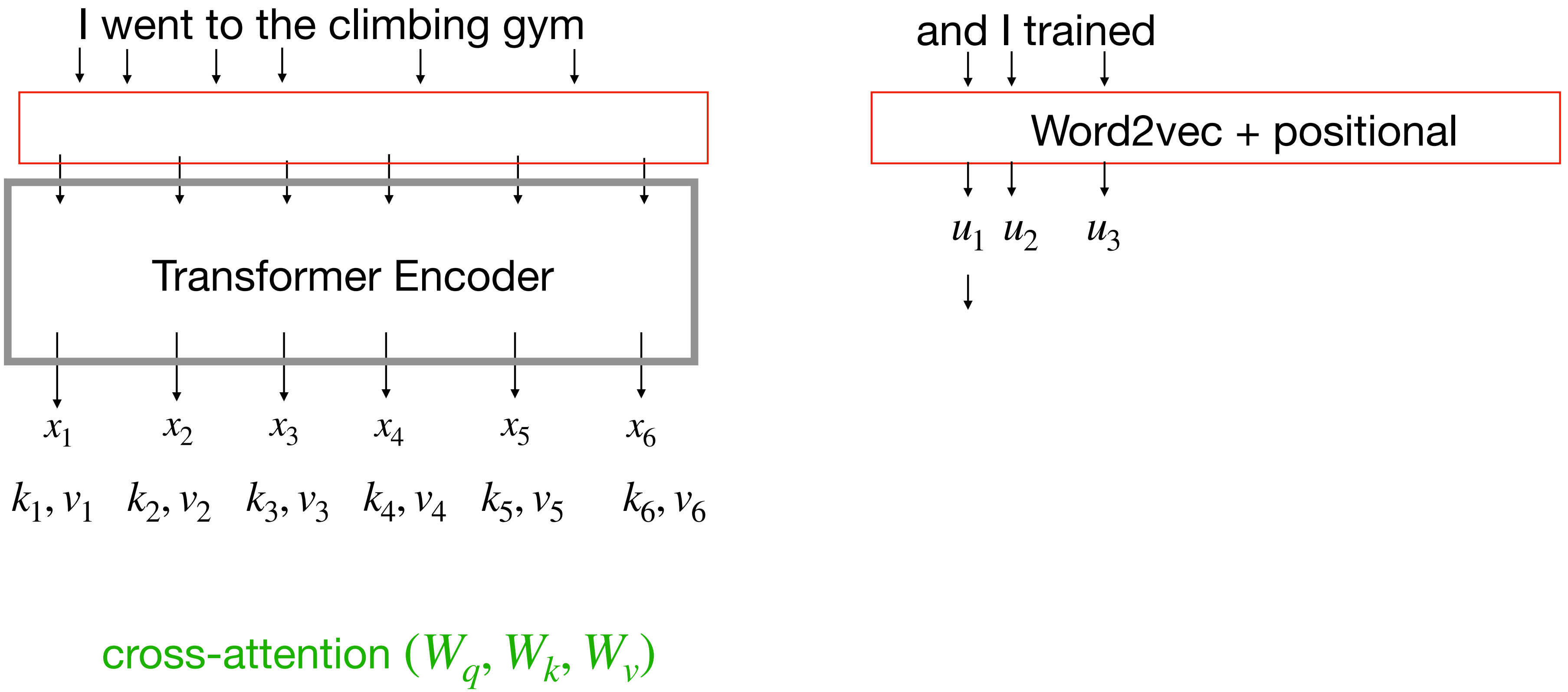
The Transformer model: decoder



I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
and I trained
Word2vec + positional
u_1 u_2 u_3
$q_1 = W_q u_1$
$x_7 = \sum_{i=1}^{6} p_i v_i$

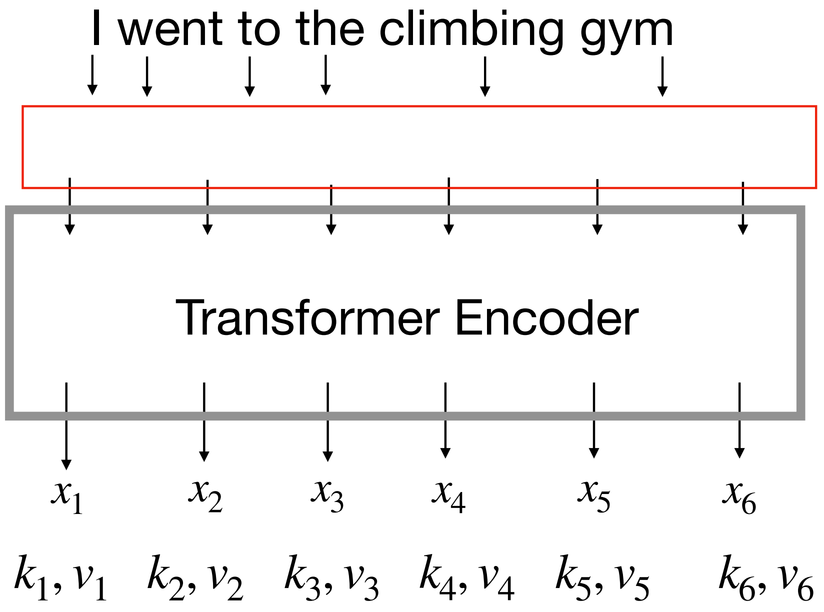
cross-attention (W_q, W_k, W_v)

The Transformer model: decoder

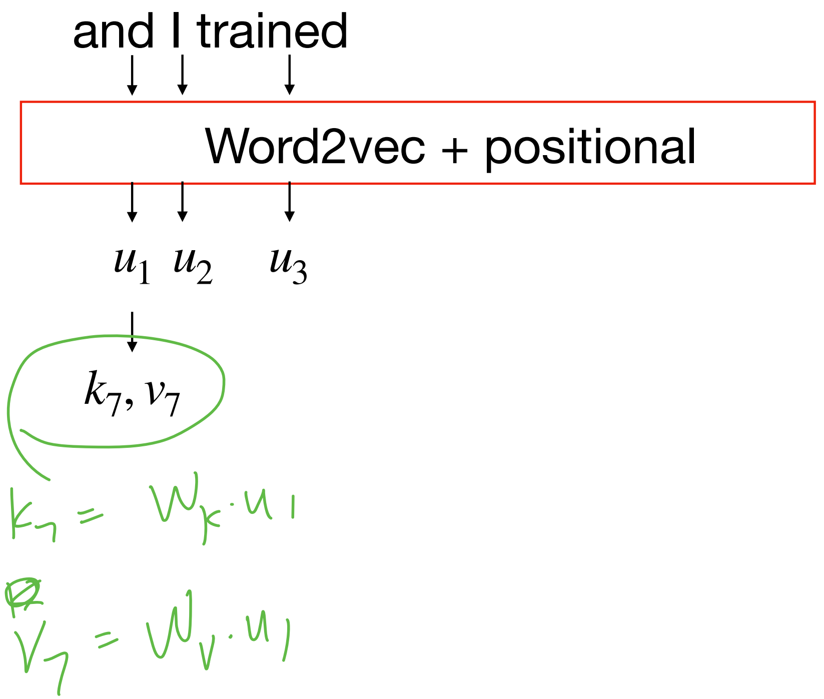


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
cross-attention (W_q, W_k, W_v)
and I trained
Word2vec + positional
u_1 u_2 u_3

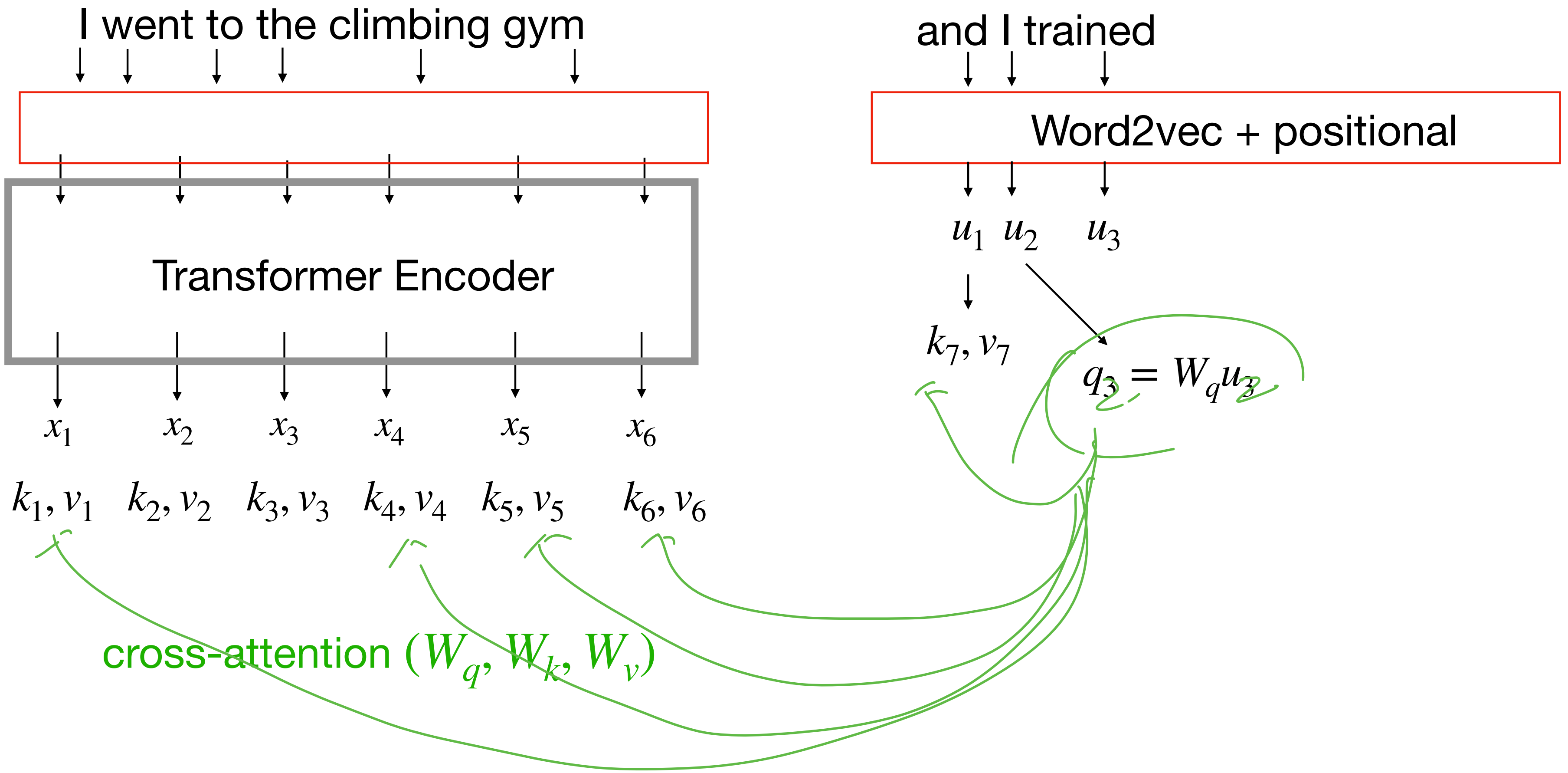
The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6

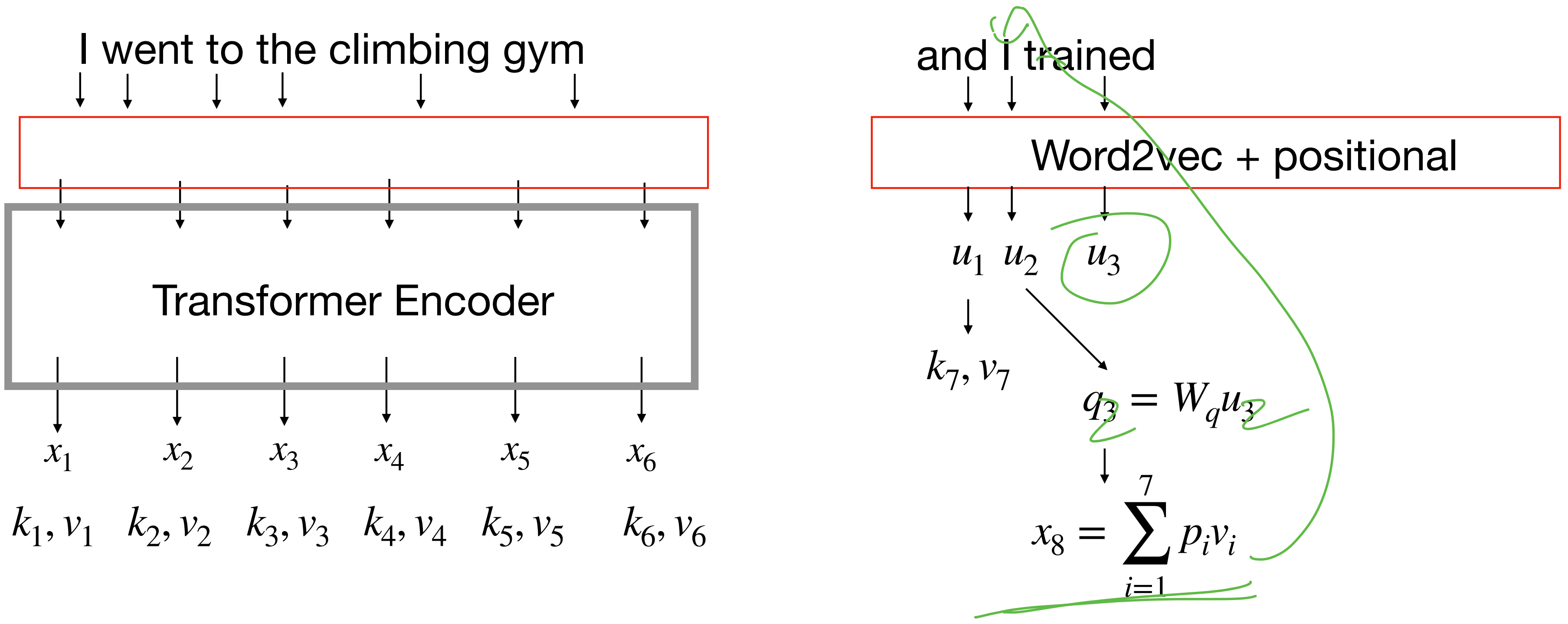
cross-attention (W_q, W_k, W_v)



The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
and I trained
Word2vec + positional
u_1 u_2 u_3
k_7, v_7
$q_3 = W_q u_3$
cross-attention (W_q, W_k, W_v)

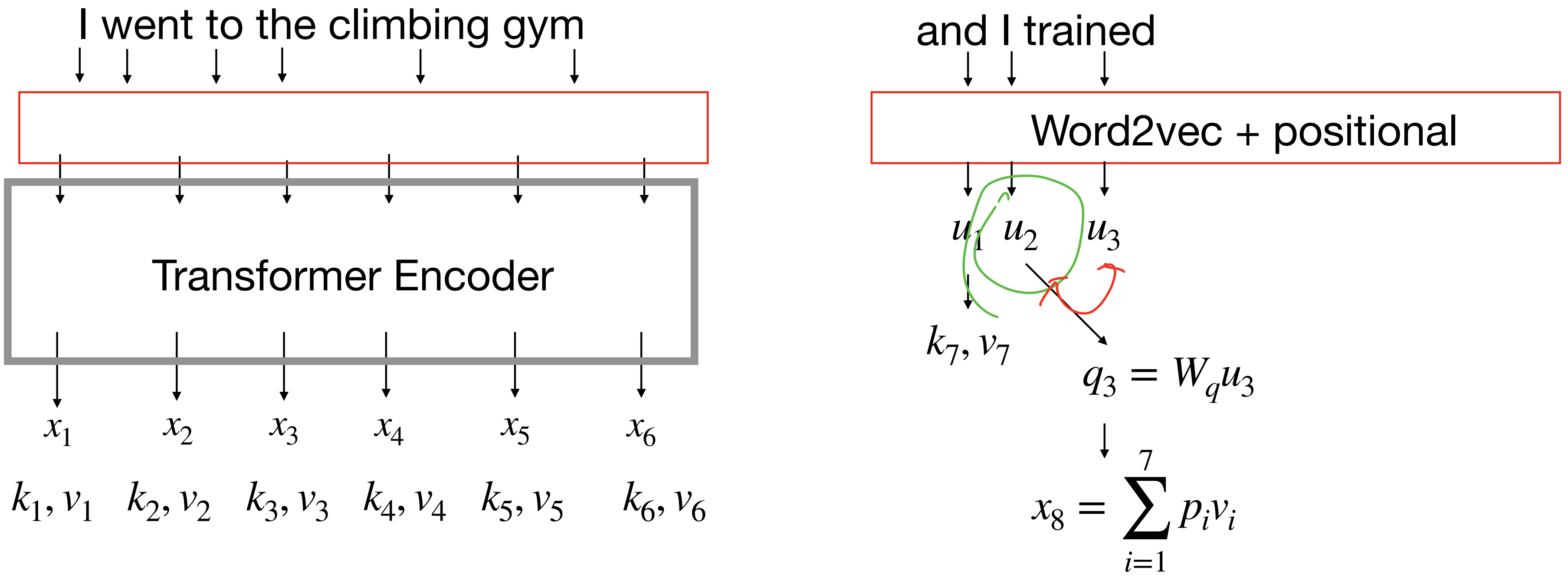
The Transformer model: decoder



I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
and I trained
Word2vec + positional
u_1 u_2 u_3
k_7, v_7
$q_3 = W_q u_3$
$x_8 = \sum_{i=1}^{7} p_i v_i$

cross-attention (W_q, W_k, W_v)

The Transformer model: decoder

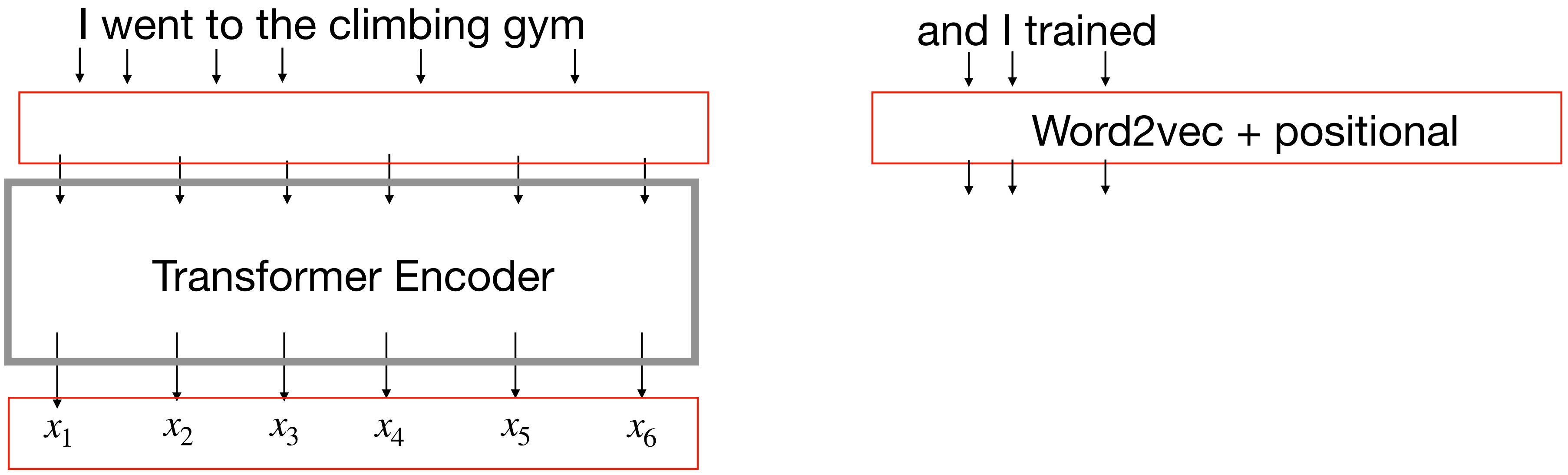


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
k_1, v_1 k_2, v_2 k_3, v_3 k_4, v_4 k_5, v_5 k_6, v_6
and I trained
Word2vec + positional
u_1 u_2 u_3
k_7, v_7
$q_3 = W_q u_3$
$x_8 = \sum_{i=1}^{7} p_i v_i$

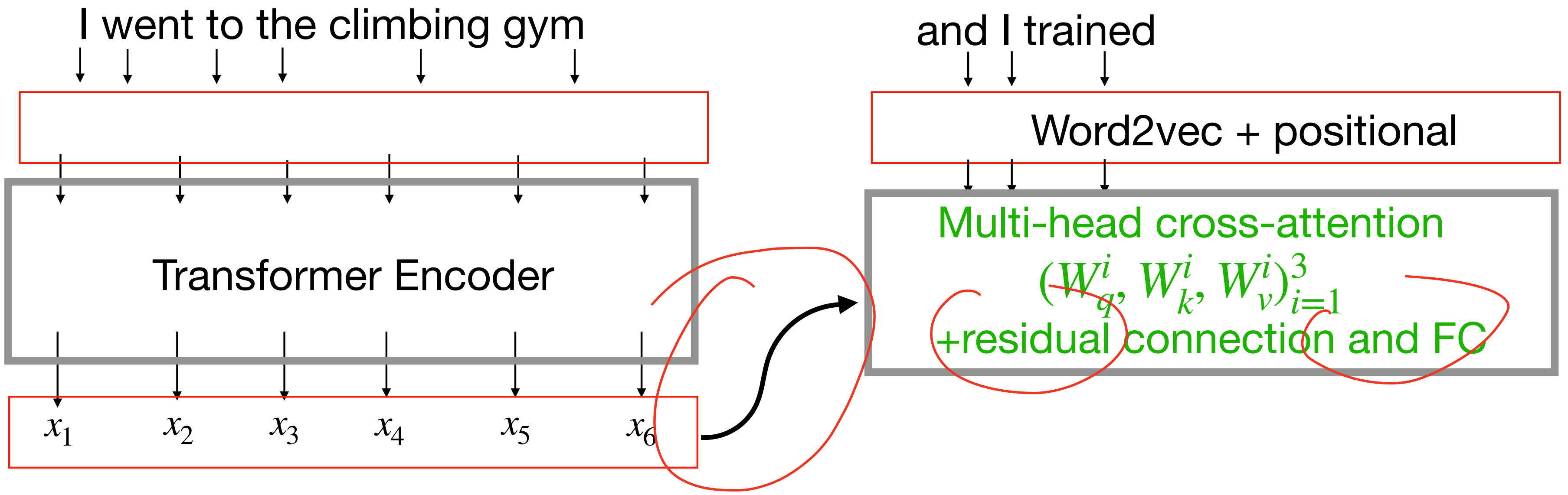
cross-attention (W_q, W_k, W_v)

Note: we do not pay attention to future words

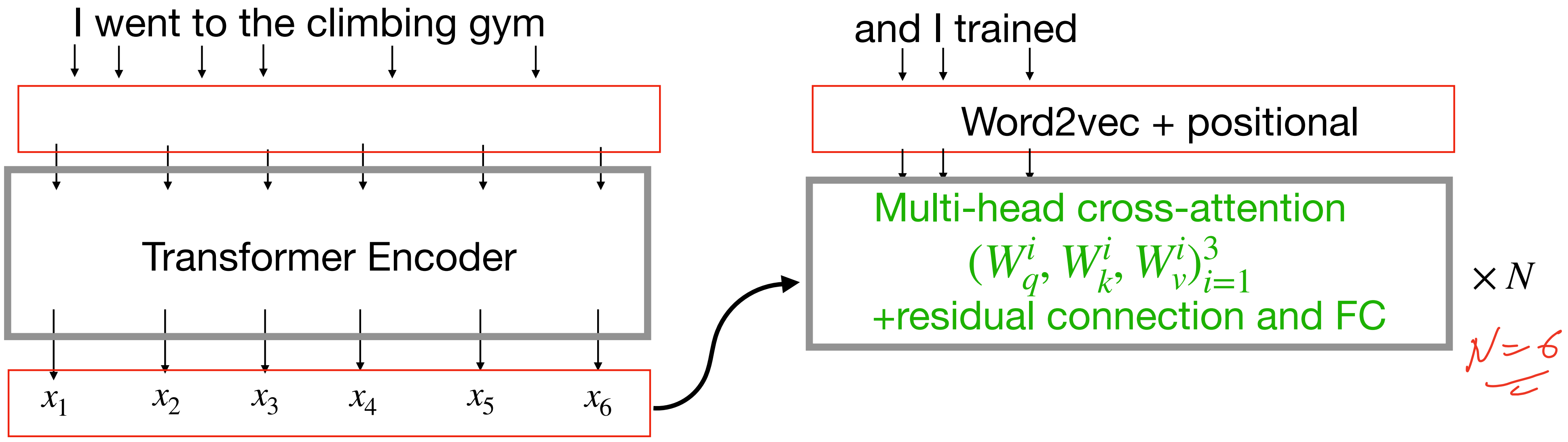
The Transformer model: decoder



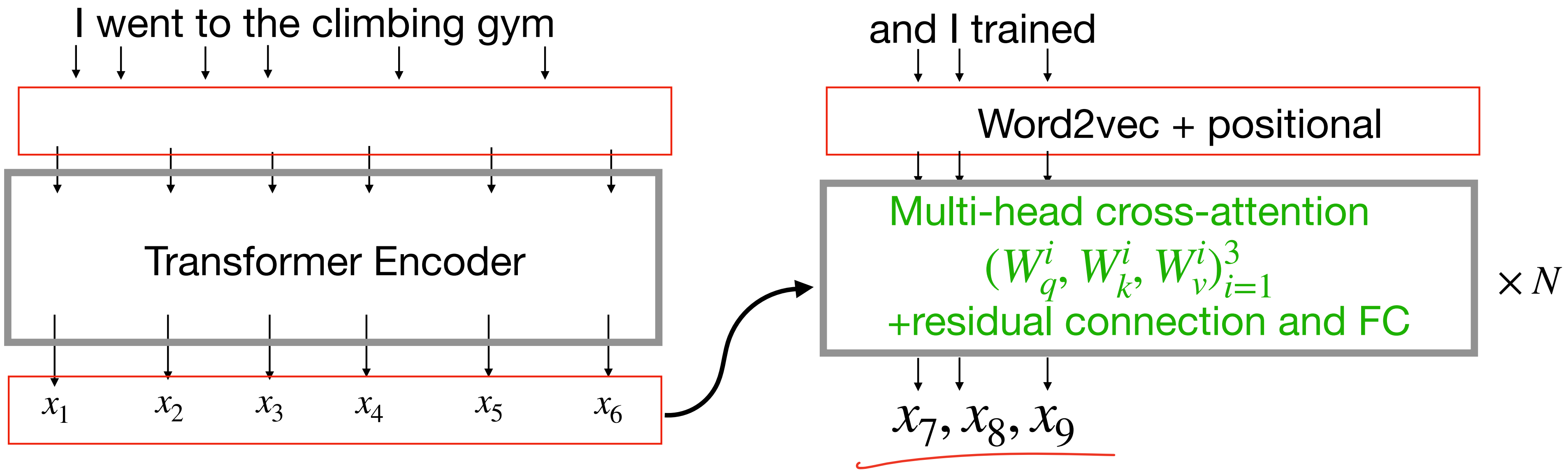
The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC

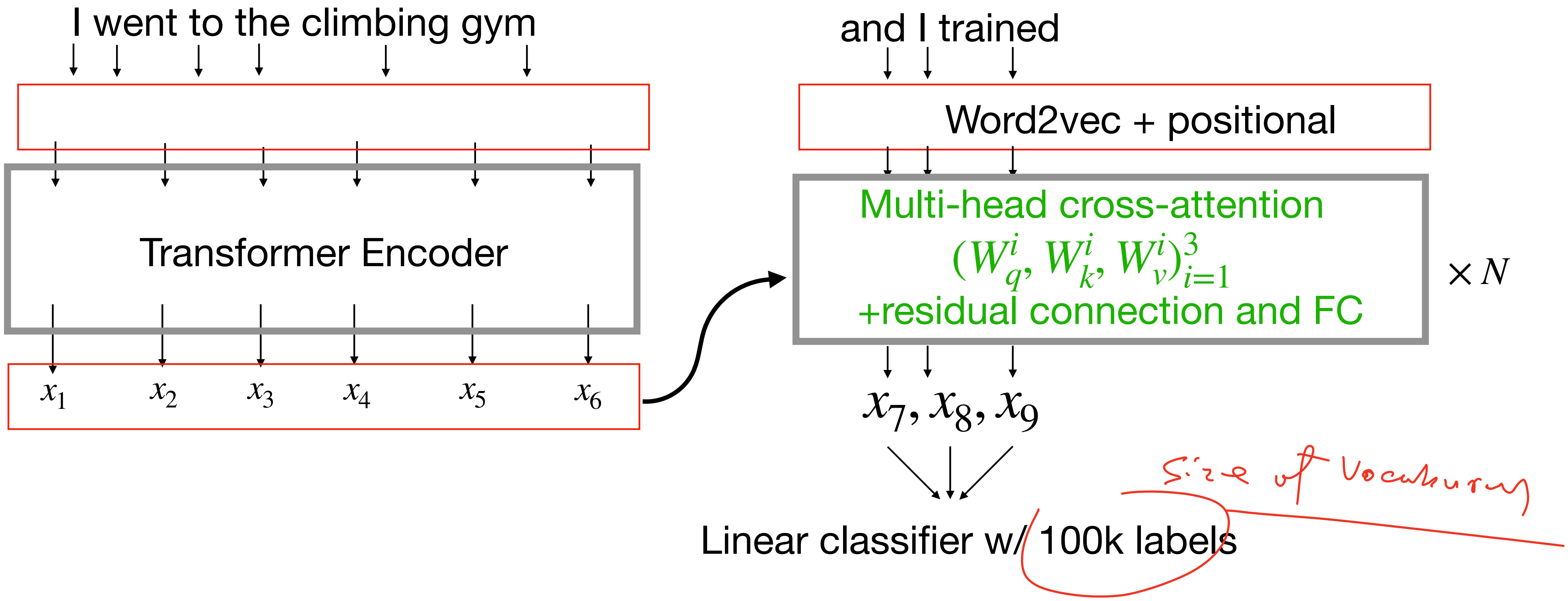
The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
$N=6$

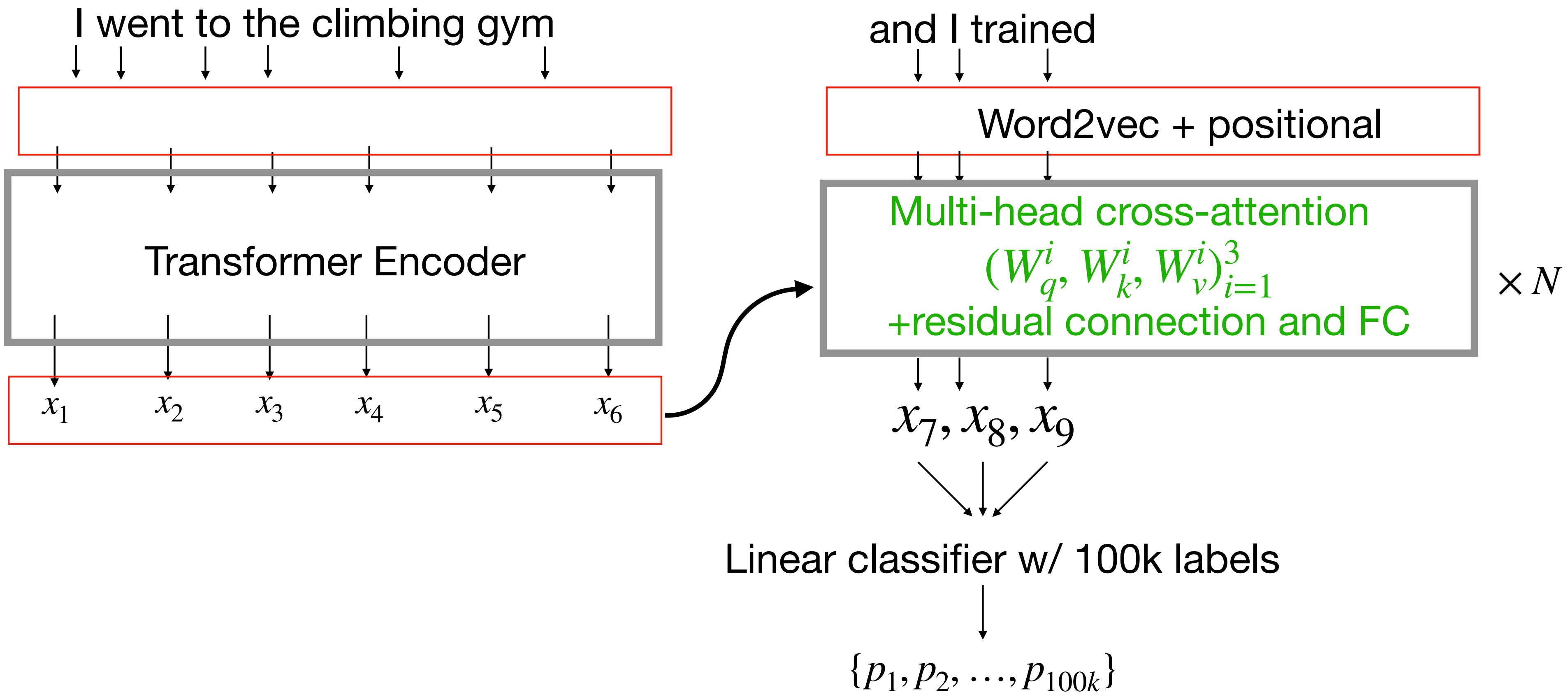
The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
x_7, x_8, x_9

The Transformer model: decoder

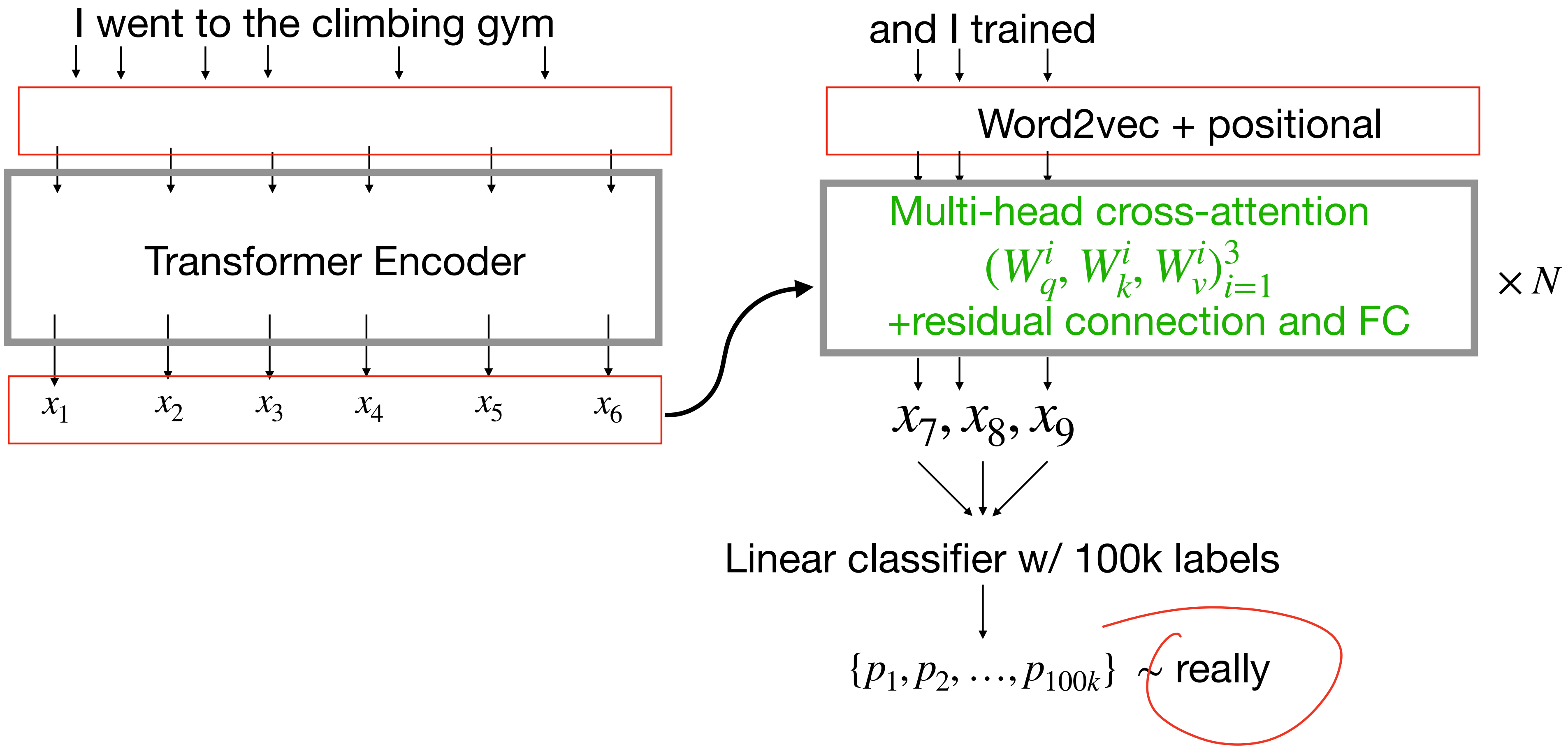

I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
x_7, x_8, x_9
Linear classifier w/ 100k labels
Size of Vocabulary

The Transformer model: decoder

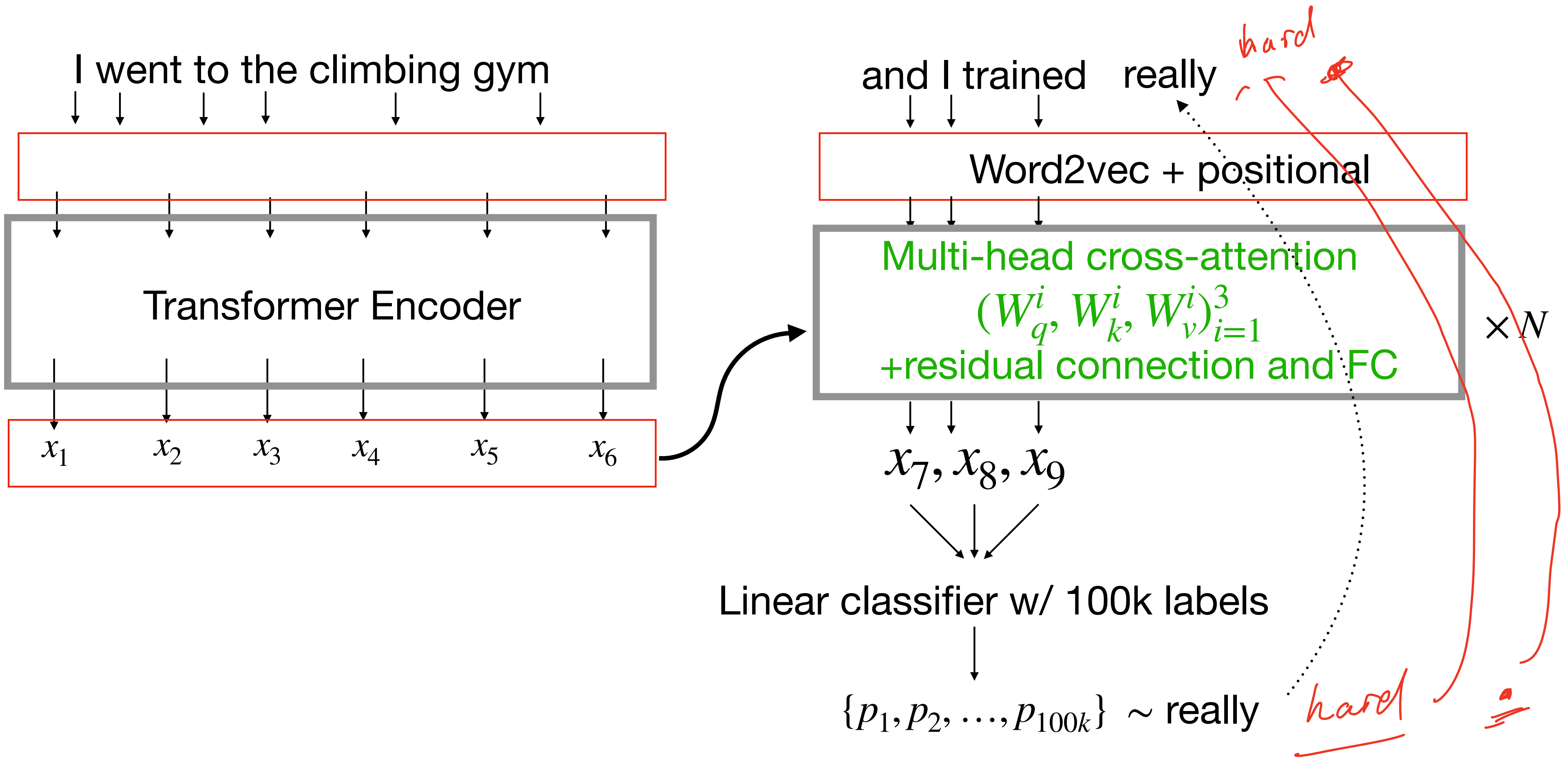


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
x_7, x_8, x_9
Linear classifier w/ 100k labels
$\{p_1, p_2, \ldots, p_{100k}\}$

The Transformer model: decoder


I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
x_7, x_8, x_9
Linear classifier w/ 100k labels
$\{p_1, p_2, \ldots, p_{100k}\} \sim$ really

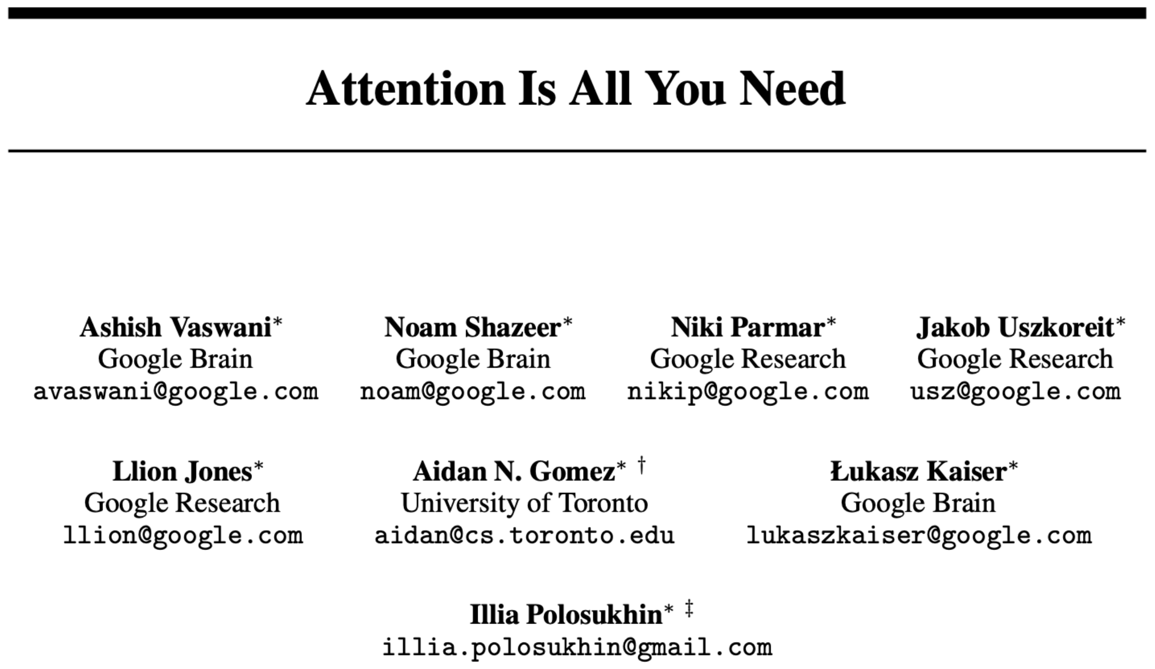
The Transformer model: decoder



I went to the climbing gym
Transformer Encoder
x_1 x_2 x_3 x_4 x_5 x_6
and I trained really
hard
Word2vec + positional
Multi-head cross-attention
$(W_q^i, W_k^i, W_v^i)_{i=1}^3$
+residual connection and FC
$\times N$
x_7, x_8, x_9
Linear classifier w/ 100k labels
$\{p_1, p_2, \ldots, p_{100k}\} \sim$ really
hard

Take home task:

Check out the the original paper (not too hard to read!)



Attention Is All You Need

Ashish Vaswani[*]
Google Brain
avaswani@google.com

Noam Shazeer[*]
Google Brain
noam@google.com

Niki Parmar[*]
Google Research
nikip@google.com

Jakob Uszkoreit[*]
Google Research
usz@google.com

Llion Jones[*]
Google Research
llion@google.com

Aidan N. Gomez[*] [†]
University of Toronto
aidan@cs.toronto.edu

Łukasz Kaiser[*]
Google Brain
lukaszkaiser@google.com

Illia Polosukhin[*] [‡]
illia.polosukhin@gmail.com