# NAJNOWSZE OSIĄGNIĘCIA W DZIEDZINIE PRZETWARZANIA JĘZYKA NATURALNEGO

Model Transformatora

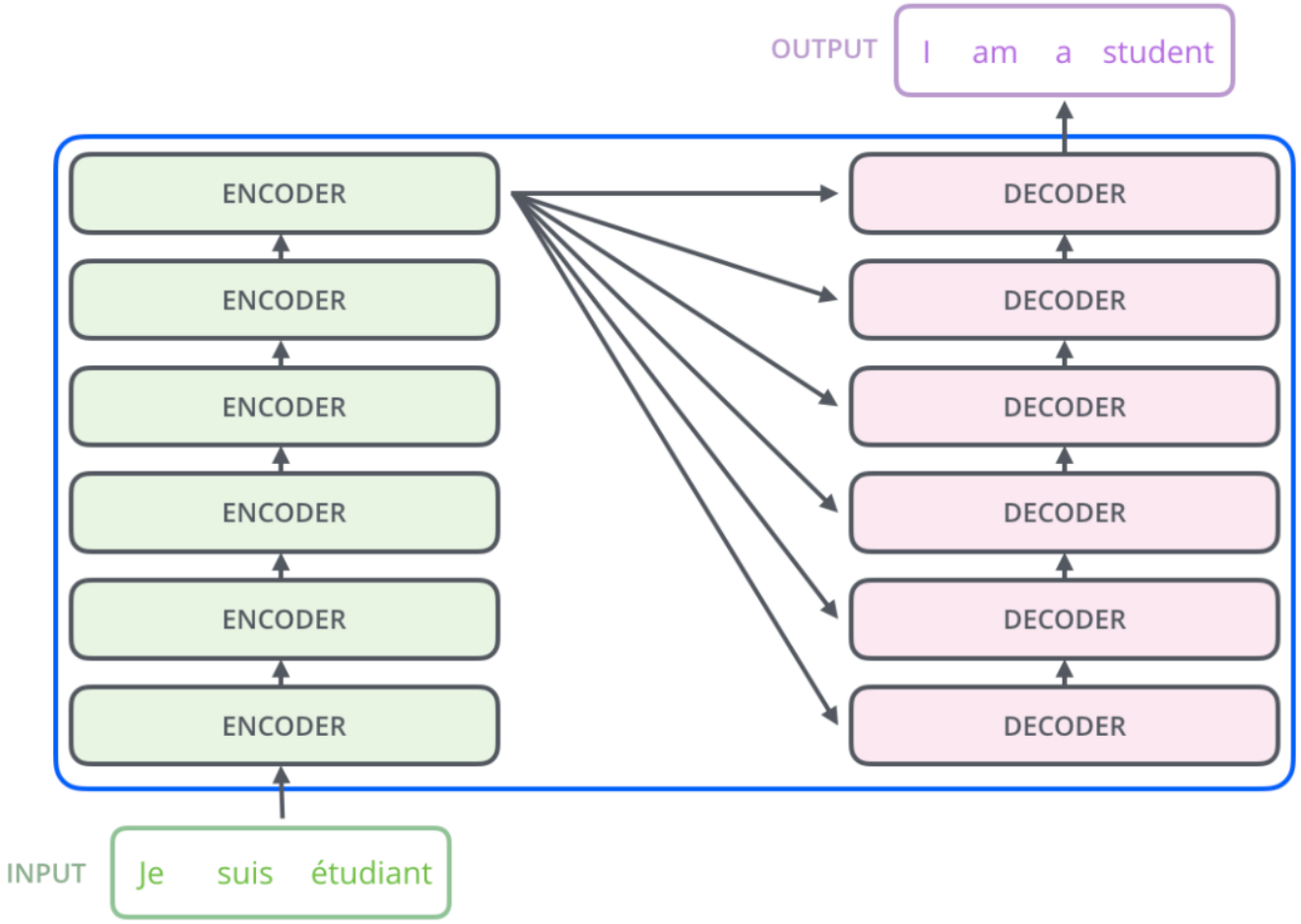
Pola Lubińska

# TRANSFORMATOR

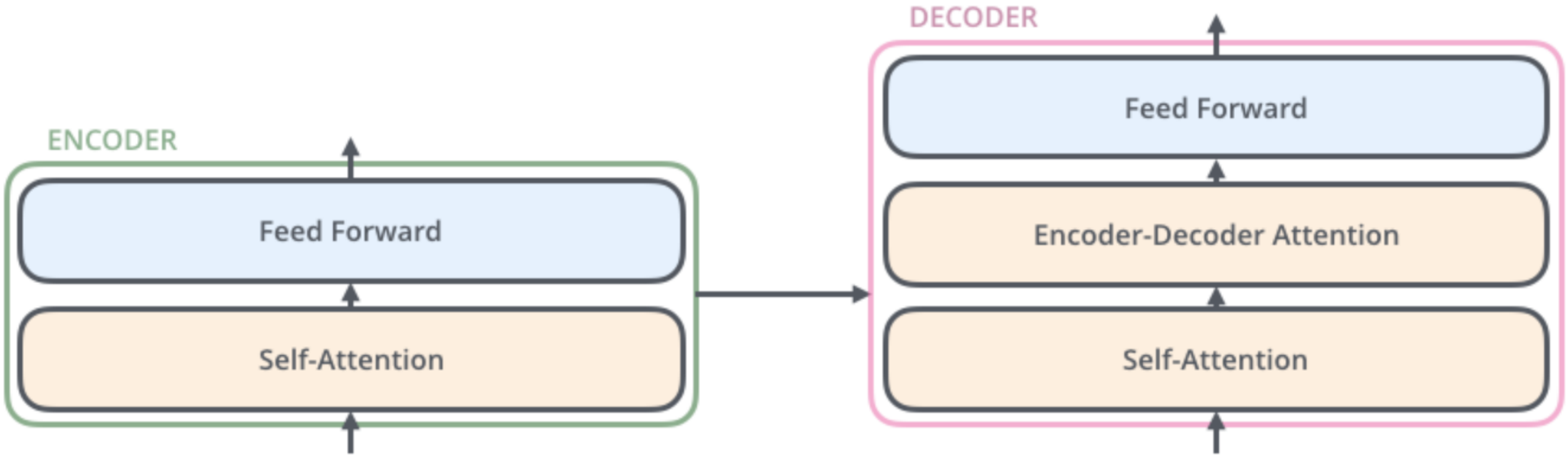
- Architektra Transformatora została poraz pierwszy przedstawiona w 2017 roku w artykule **"Attention is all you need",** który został opracowany przez naukowców pracujących w zespole Google Brain.
- Model transformatora opiera się całkowicie na **mechanizmie uwagi** (z ang. attention mechanism) i całkowicie wyeliminował rekurencyjność. Dzięki temu jest możliwość zrównoleglania obliczeń.

- Wiele najnowszych modeli językowych, takich jak BERT, GPT3 oraz T5 oparta jest na architekturze Transformatora.

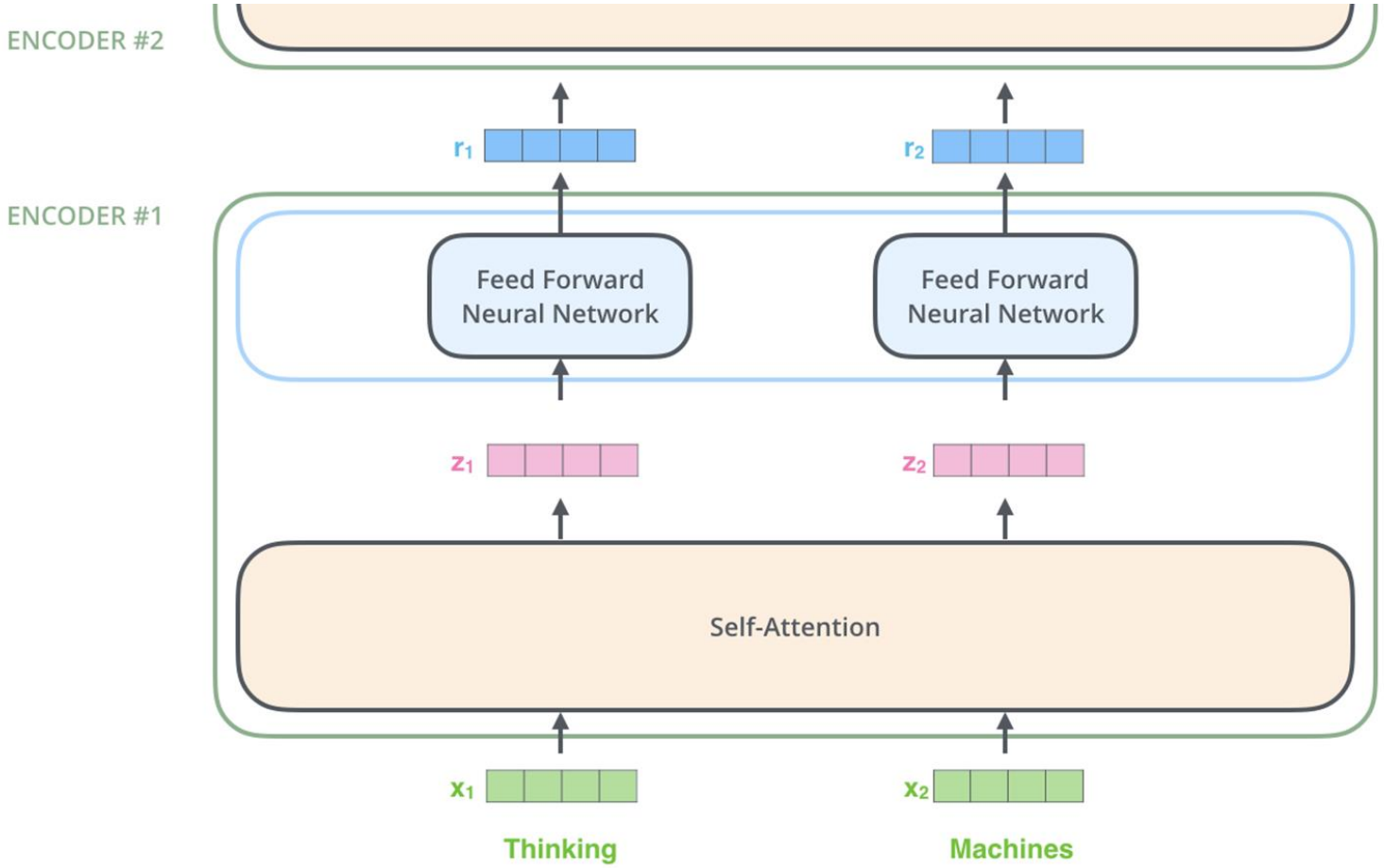
# ARCHITEKTURA TRANSFORMATORA



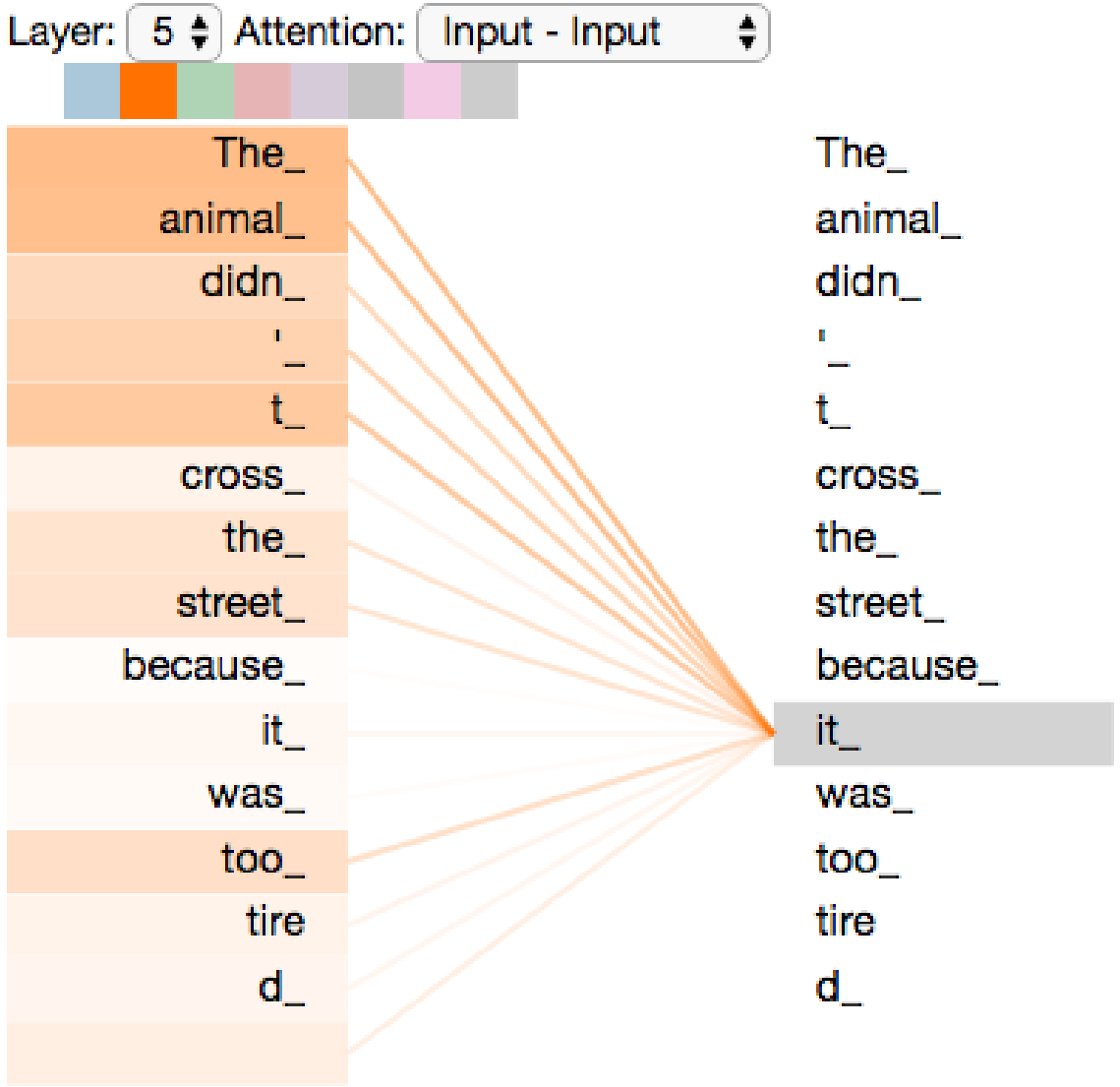
# ARCHITEKTURA TRANSFORMATORA



# STRUKTURA ENKODERA



# MECHANIZM SAMOUWAGI



# OBLICZENIE MECHANIZMU SAMOUWAGI

1. Policz wektor kwerendy $\mathbf{q}_i$ (z ang. *query vector*), wktor klucza $\mathbf{k}_i$ (z ang. key vector), oba wektory są wymiaru $d_k$. Policz również wektor wartości $\mathbf{v}_i$ (z ang. value vector), który jest wymiaru $d_v$. Każdy z trzech wektorów otrzymuje się poprzez przemnożenie początkowych wektorów osadzenia $\boldsymbol{x}_i \in \mathbb{R}^{d_{\text{model}}}$ dla elementu $i$ w sekwencji przez każdą z trzech macierzy wag:

$$\mathbf{q}_i = \mathbf{x}_i \mathbf{W}^Q, \quad \mathbf{W}^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}, \tag{3.1}$$

$$\mathbf{k}_i = \mathbf{x}_i \mathbf{W}^K, \quad \mathbf{W}^K \in \mathbb{R}^{d_{\text{model}} \times d_k}, \tag{3.2}$$

$$\mathbf{v}_i = \mathbf{x}_i \mathbf{W}^V, \quad \mathbf{W}^V \in \mathbb{R}^{d_{\text{model}} \times d_v}, \tag{3.3}$$

gdzie macierze wag $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V$ są uczone w trakcie procesu trenowania modelu.

2. Porównaj element $i$ względem wszystkich innych elementów w sekwencji, wykonując iloczyn skalarny jego wektora zapytania $\mathbf{q}_i$ ze wszystkimi wektorami kluczy $\mathbf{k}_j$ w sekwencji:

$$s_{ij} = \mathbf{q}_i \mathbf{k}_j, \qquad \forall j = 1, \ldots, N \tag{3.4}$$

3. Następnie, w celu przeskalowania wag, podziel wyniki z poprzedniego kroku przez pierwiastek kwadratowy z wymiaru wektora klucza $d_k$:

$$s'_{ij} = \frac{s_{ij}}{\sqrt{d_k}}, \qquad \forall j = 1, \ldots, N. \tag{3.5}$$

# OBLICZENIE MECHANIZMU SAMOUWAGI

4. Aby znormalizować wagi z poprzedniego kroku, użyj funkcji softmax dla każdego elementu $i$:

$$s''_{ij} = \frac{e^{s'_{ij}}}{\sum_{j=1}^{N} e^{s'_{ij}}}, \quad \forall j = 1, \ldots, N. \tag{3.6}$$

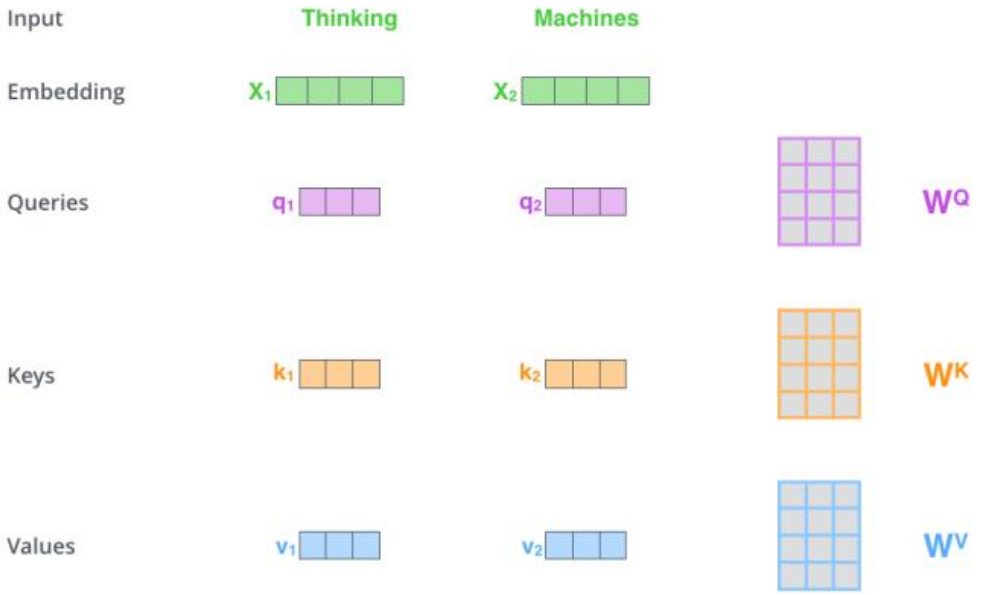
5. Pomnóż każdy wektor wartości $\mathbf{v}_j$ przez odpowiadający mu znormalizowany wynik:

$$\boldsymbol{v}'_{ij} = s''_{ij}\boldsymbol{v}_j, \quad \forall j = 1, \ldots, N. \tag{3.7}$$

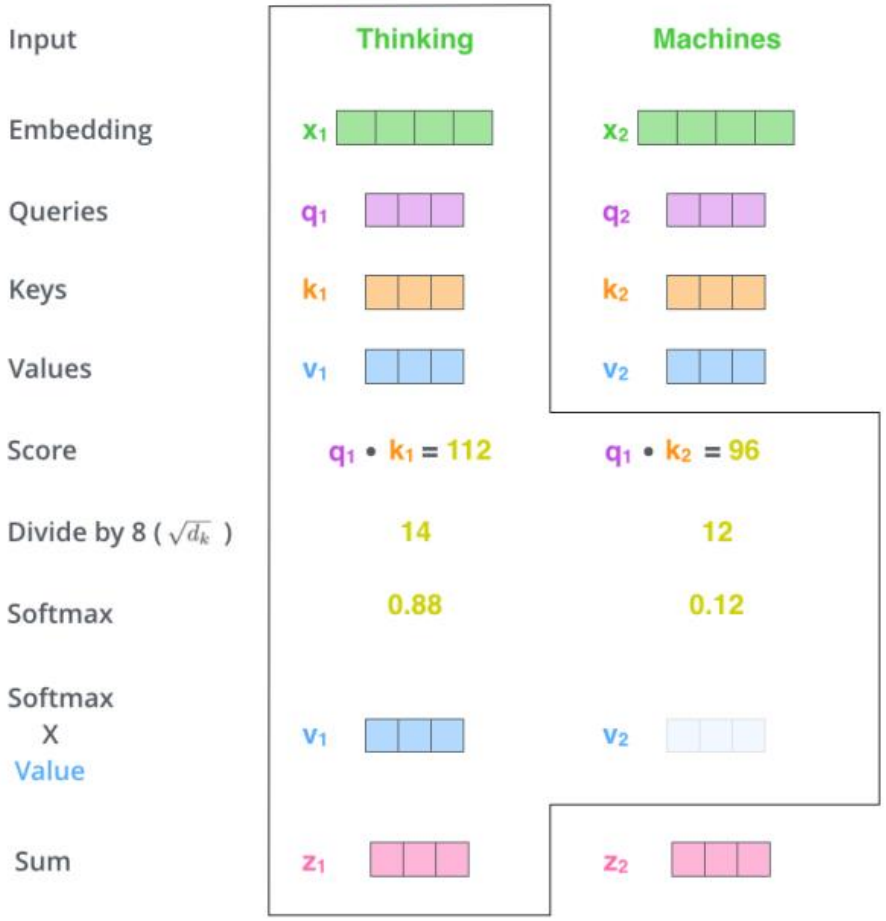
6. Zsumuj ważone wektory wartości jako końcowy wynik mechanizmu samouwagi:

$$\boldsymbol{z}_i = \sum_{j=1}^{N} \boldsymbol{v}'_{ij} \tag{3.8}$$

# OBLICZANIE MECHANIZMU SAMOUWAGI




(a) Obliczanie wektorów: kwerendy $\mathbf{q}_i$, klucza $\mathbf{k}_i$ oraz wartości $\mathbf{v}_i$.



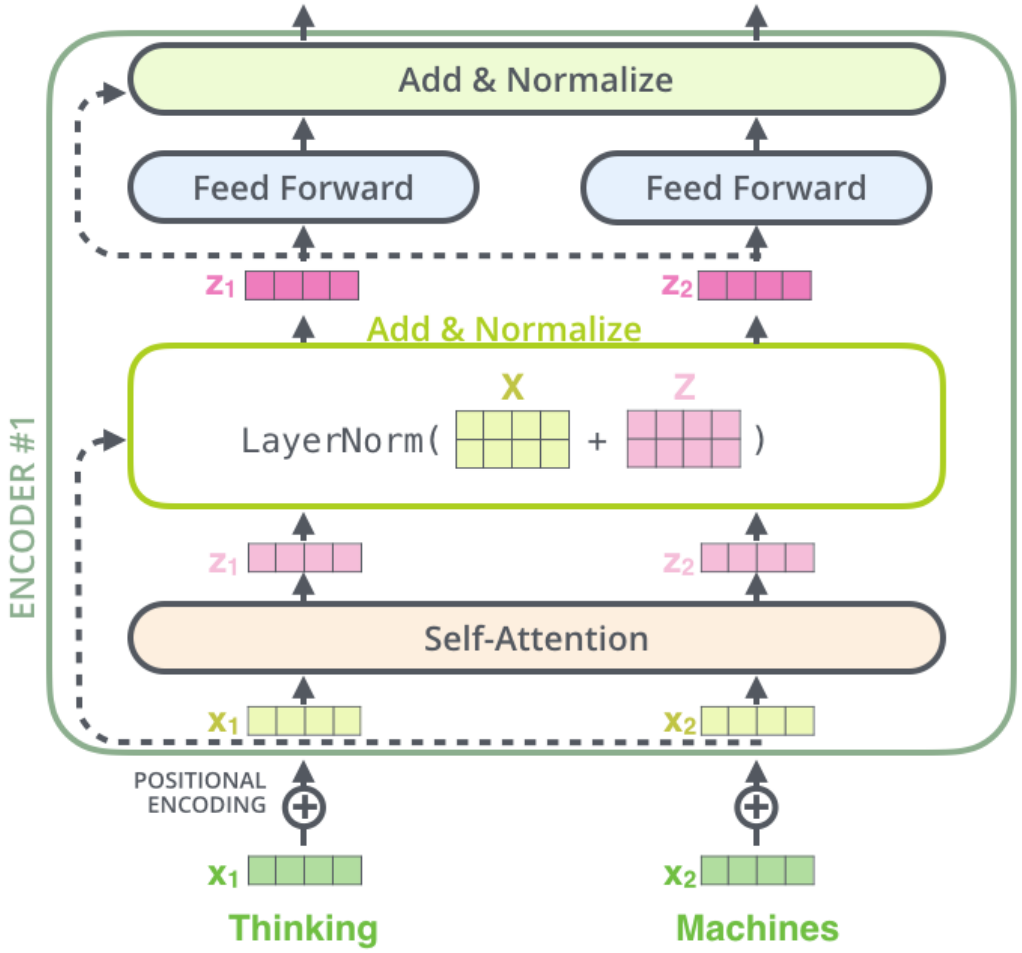

(b) Obliczanie wektorów $\mathbf{z}_i$ zwracanych przez mechanizm samouwagi.

# OBLICZENIE SCALE-DOT PRODUCT ATTENTION

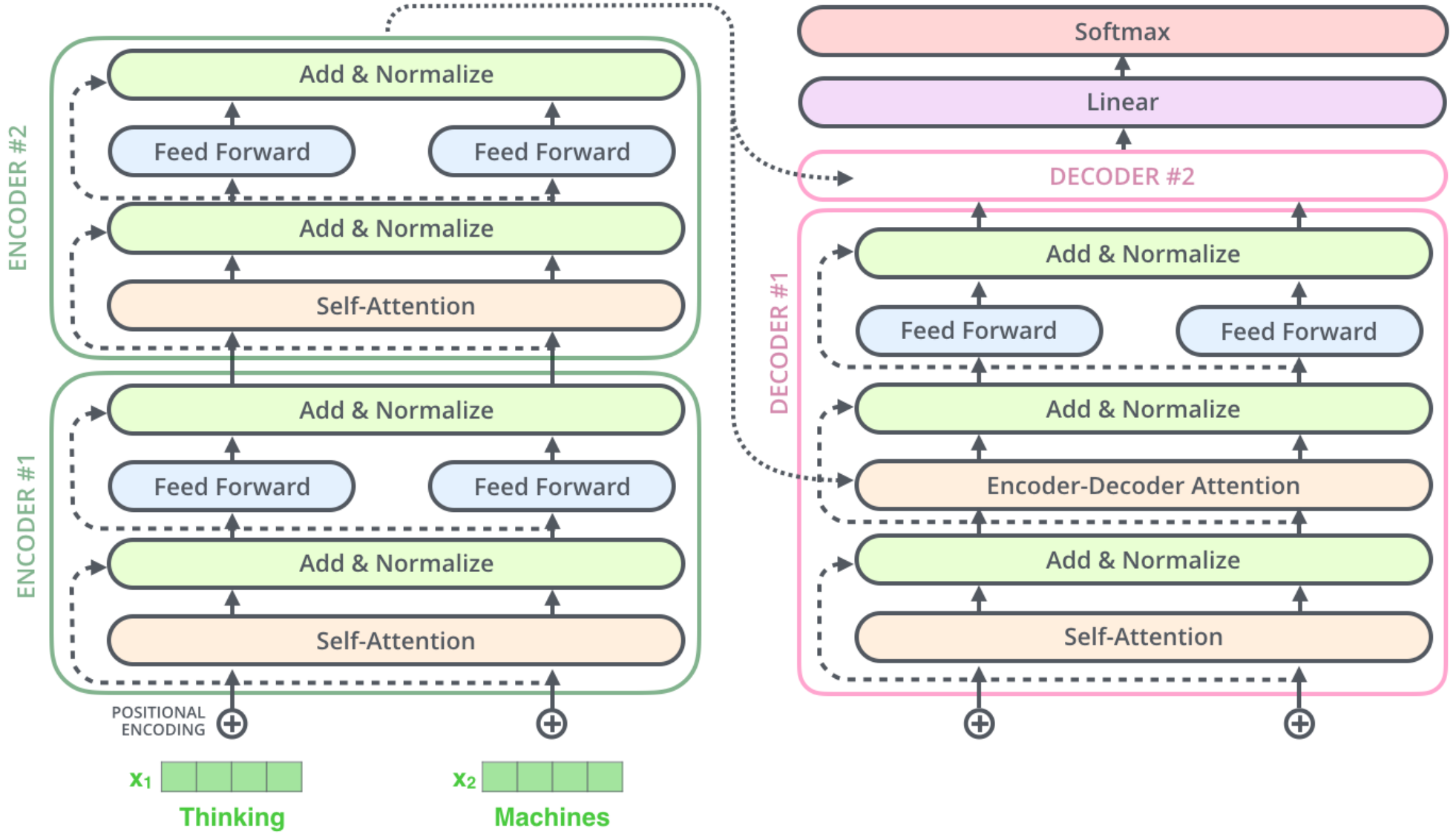
W praktyce mechanizm samouwagi jest liczony na zbiorze wszystkich wektorów kwerend jednocześnie. Wektory te są połączone razem jako jedna macierz $\mathbf{Q} \in \mathbb{R}^{N \times d_k}$. Wetory wartości oraz wektory kluczy również można zaprezentować w notacji macierzowej jako $\mathbf{K} \in \mathbb{R}^{N \times d_k}$ oraz $\mathbf{V} \in \mathbb{R}^{N \times d_v}$.
Korzystając z powyżej notacji, macierz wyjściowa zwracana przez mechanizm samouwagi obliczana jest w następujący sposób:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}. \tag{3.9}$$

# ENKODER


Add & Normalize
Feed Forward
Feed Forward
z1
z2
Add & Normalize
X
Z
LayerNorm( + )
ENCODER #1
z1
z2
Self-Attention
x1
x2
POSITIONAL ENCODING
x1
x2
Thinking
Machines

# TRANSFORMATOR



# DZIĘKUJĘ ZA UWAGĘ!