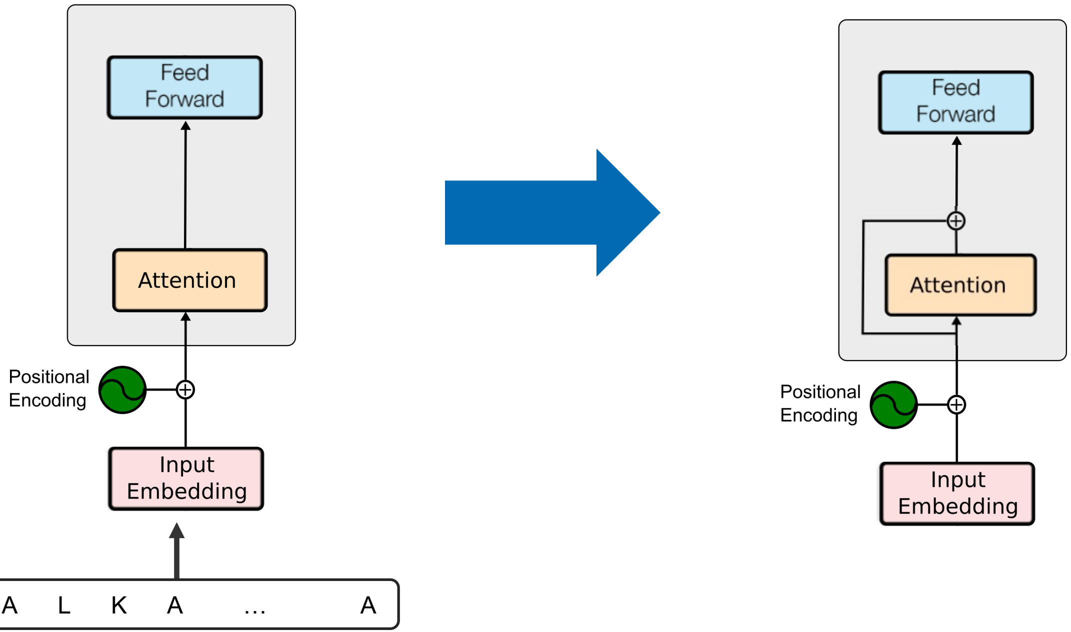
Heinrich Heine
Universität Düsseldorf

Transformer Network Encoders

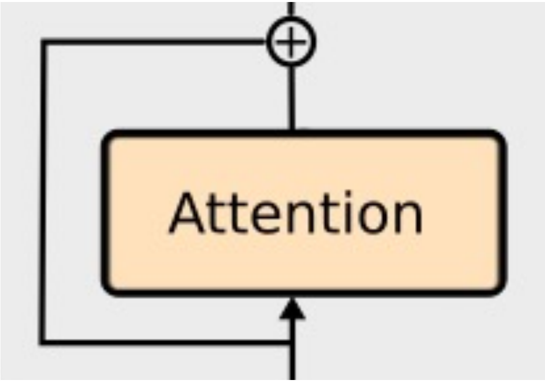
A more detailed view

Encoder



Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

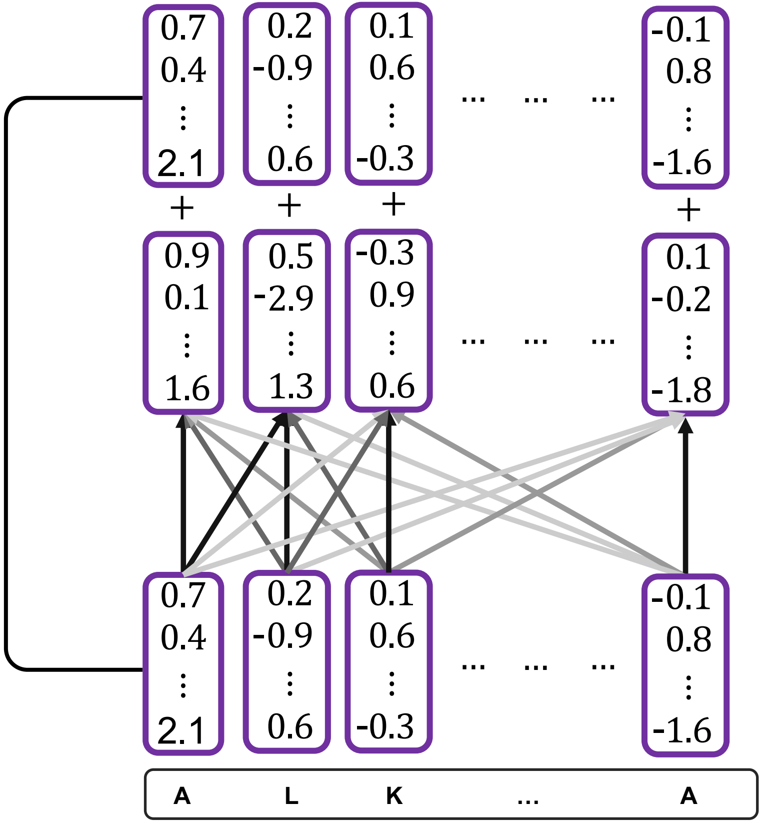
Residual connection



$$\vec{e}_i = Attention(\vec{e}_i, K, V)$$

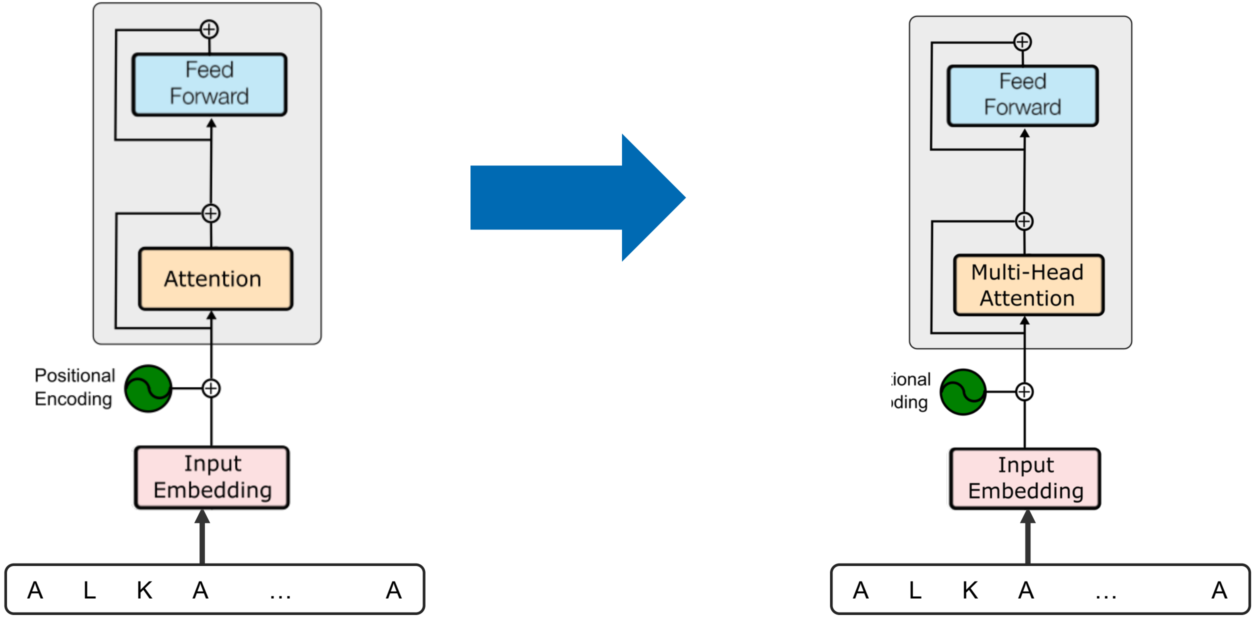
Adding residual connection

$$\vec{e}_i = Attention(\vec{e}_i, K, V) + \vec{e}_i$$



Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

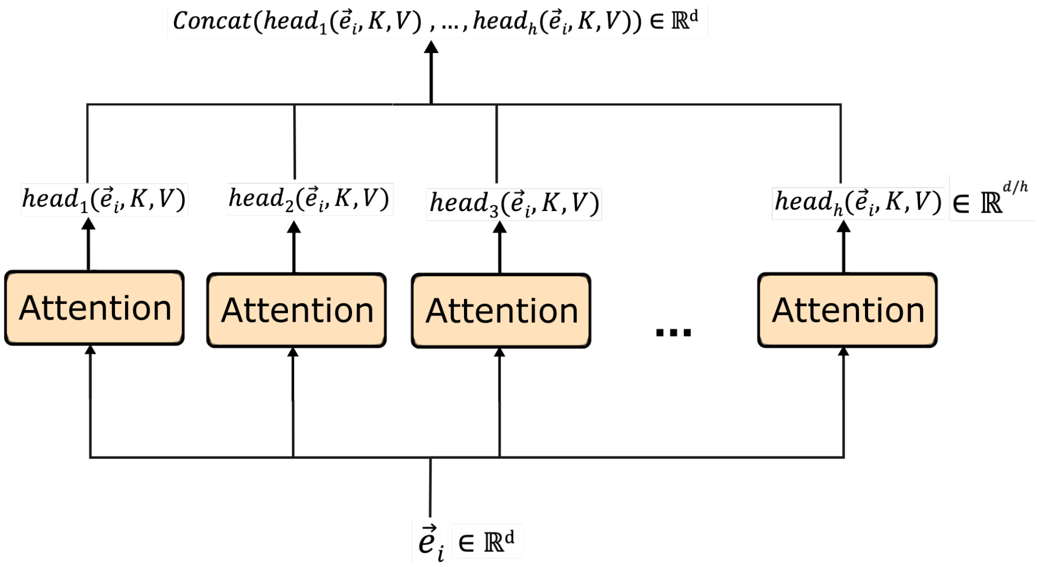
Encoder



Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

Multi-Head Attention

- Instead of applying attention function once, we apply it h times in parallel, e.g., $h = 8$.



- Each vector produced by one one the h attention heads should have dimension $m := d/h$
  - $W_E^j, W_E^j, W_V^j \in \mathbb{R}^{m \times d}$
  - $W_O \in \mathbb{R}^{d \times d}$

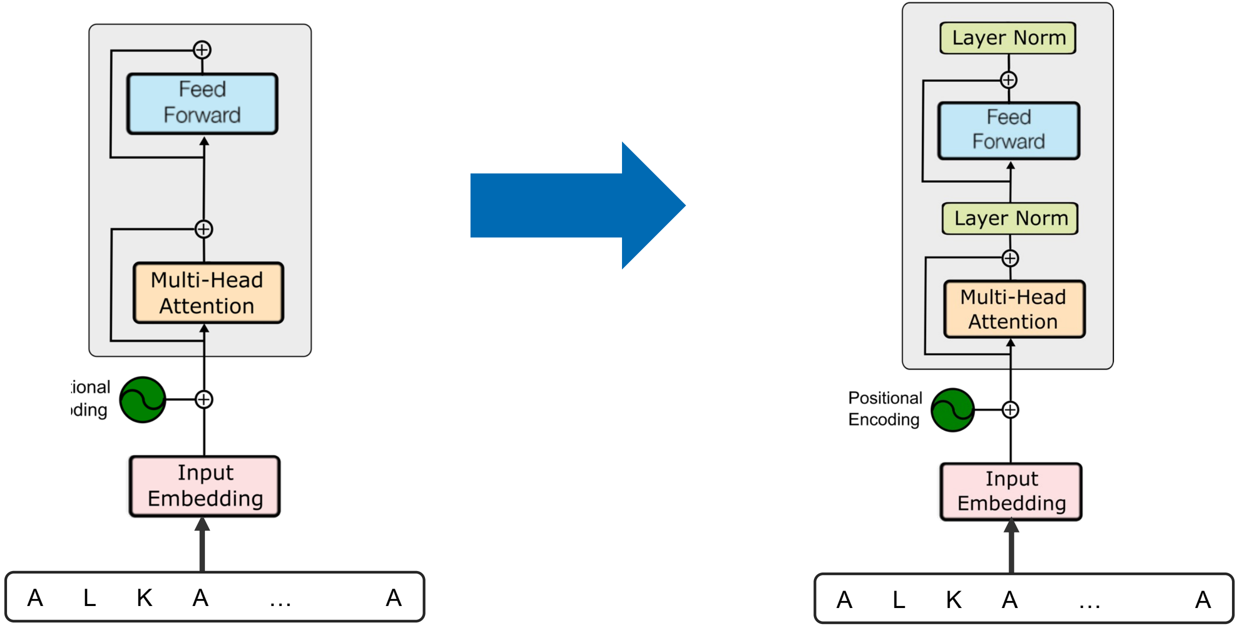
$$head_j(\vec{e}_i, K, V) = Attention\left(W_E^j \cdot \vec{e}_i, W_K^j \cdot K, W_V^j \cdot V\right), \qquad j = 1, \ldots, h; \quad \vec{e}_i \in \mathbb{R}^d$$

$$MultiHead(\vec{e}_i, K, V) = W_O \cdot Concat(head_1(\vec{e}_i, K, V), \ldots, head_h(\vec{e}_i, K, V)) \in \mathbb{R}^d$$

Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

Encoder

Heinrich Heine
Universität Düsseldorf



Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

Layout Normalization

- Example: $d = 3$, number of tokens in input sequence $N = 2$, and batch size 1
  - Output of Multi-Head attention

$$\vec{e}_1 = \begin{pmatrix} 0.5 \\ -3 \\ 8.5 \end{pmatrix} \quad \vec{e}_2 = \begin{pmatrix} -2 \\ 4 \\ 1 \end{pmatrix}$$

  - Calculating mean and standard deviation for each sample:

$$\mu_1 = \frac{1}{3}(0.5 - 3 + 8.5) = 2, \quad \mu_2 = \frac{1}{3}(-2 + 4 + 1) = 1$$

$$\sigma_1 = \sqrt{\frac{1}{3}\left[(0.5 - \mu_1)^2 + (-3 - \mu_1)^2 + (8.5 - \mu_1)^2\right]} = 2$$

$$\sigma_2 = \sqrt{\frac{1}{3}\left[(-2 - \mu_2)^2 + (4 - \mu_2)^2 + (1 - \mu_2)^2\right]} = 2$$

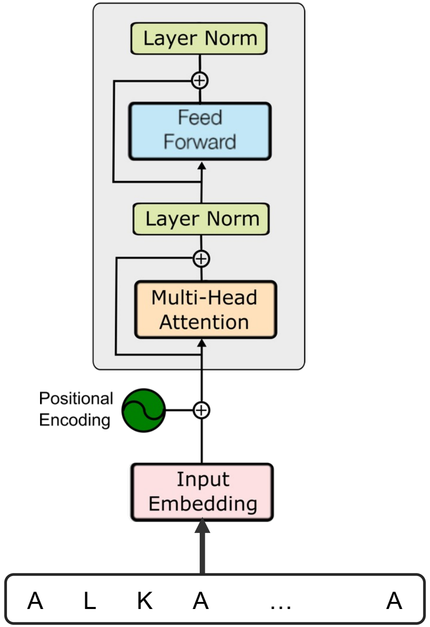
  - Normalizing all vectors

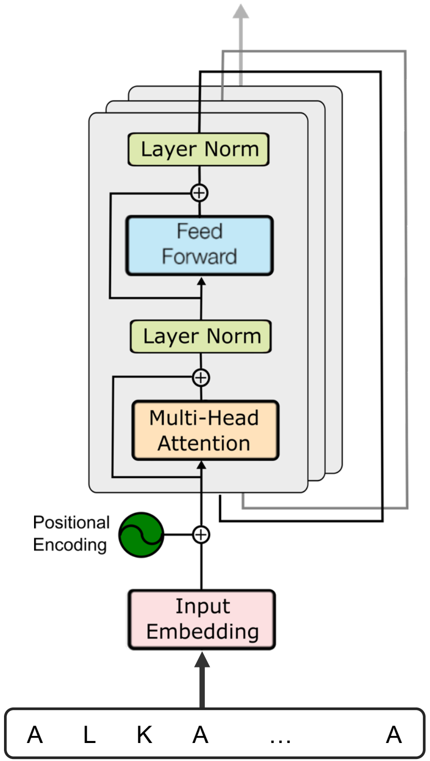
$$\frac{\vec{e}_1 - \mu_1}{\sigma_1} = \begin{pmatrix} -0.75 \\ -2.5 \\ 3.25 \end{pmatrix} \quad \frac{\vec{e}_2 - \mu_2}{\sigma_2} = \begin{pmatrix} -1.5 \\ 1.5 \\ 0 \end{pmatrix}$$

  - Adding learnable rescaling parameters $\gamma, \beta \in \mathbb{R}^3$

$$\gamma \odot \left(\frac{\vec{e}_1 - \mu_1}{\sigma_1}\right) + \beta \qquad \gamma \odot \left(\frac{\vec{e}_2 - \mu_2}{\sigma_2}\right) + \beta$$

Encoder





Modified figure from: Vaswani, A., et al. (2017). Attention is all you need. *Adv. Neural Inf. Process. Syst.*, *30*.

Parallelization of Attention Calculations

$$Attention(\vec{e}_i, K, V) = softmax(\frac{\vec{e}_i^{\,T} \cdot K}{\sqrt{d}}) \cdot V^T \in \mathbb{R}^{\mathrm{d}} \qquad K = (\vec{e}_1, \vec{e}_2, \dots, \vec{e}_7), \qquad V^T = \begin{pmatrix} \vec{e}_1^{\,T} \\ \vec{e}_2^{\,T} \\ \vdots \\ \vec{e}_7^{\,T} \end{pmatrix}$$

Parallelize

$$Attention(Q, K, V) = softmax(\frac{Q^T \cdot K}{\sqrt{d}}) \cdot V^T \in \mathbb{R}^{\mathrm{N\,x\,d}} \qquad Q^T = \begin{pmatrix} \vec{e}_1^{\,T} \\ \vec{e}_2^{\,T} \\ \vdots \\ \vec{e}_7^{\,T} \end{pmatrix}$$

$$Attention(W_E \cdot Q, W_K \cdot K, W_V \cdot V)$$