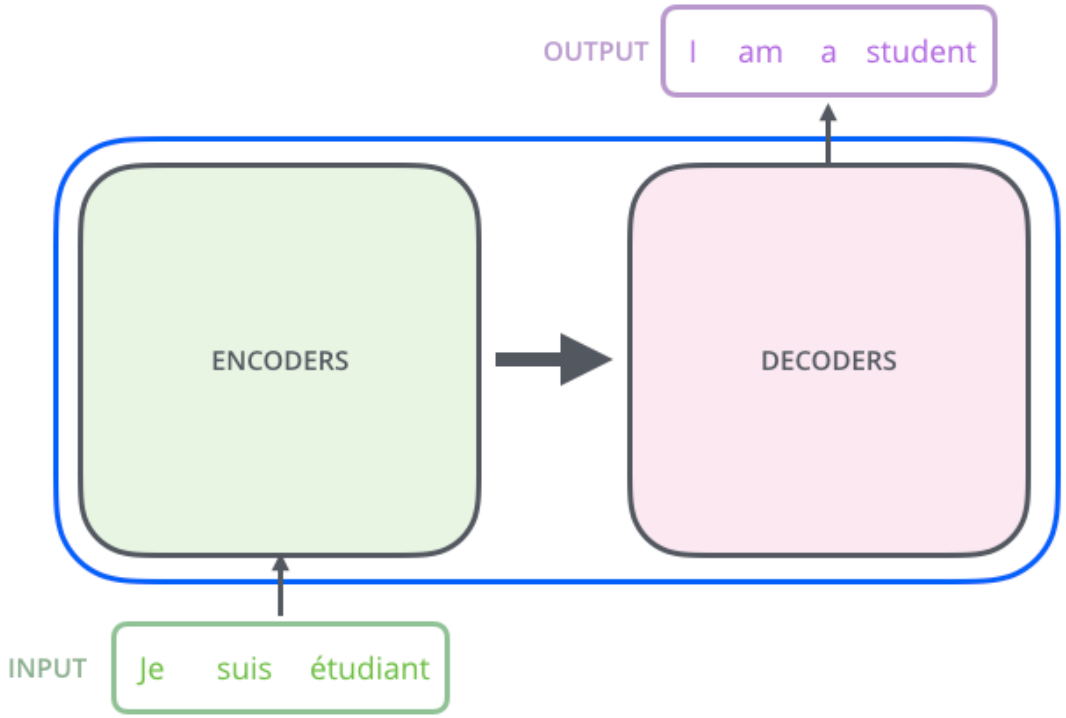
Transformers

Transformers

- Transformers wurden 2017 von Google Brain eingeführt
- „Attention is all you need!“
- Ursprünglich für Machine Translation
- Schnell für alle NLP-Tasks
- Dann auch für viele andere Bereiche …

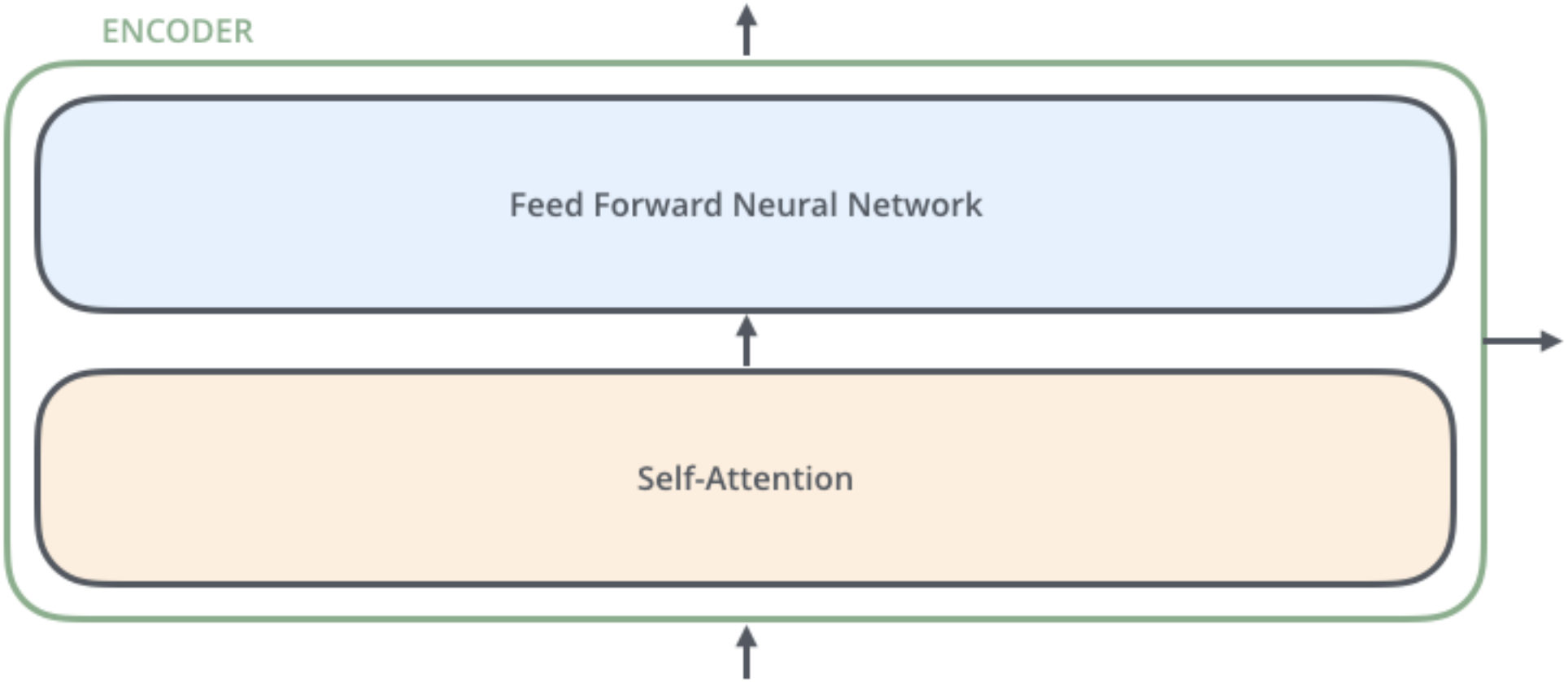
Encoder-Decoder Architektur

Source: https://jalammar.github.io/illustrated-transformer/



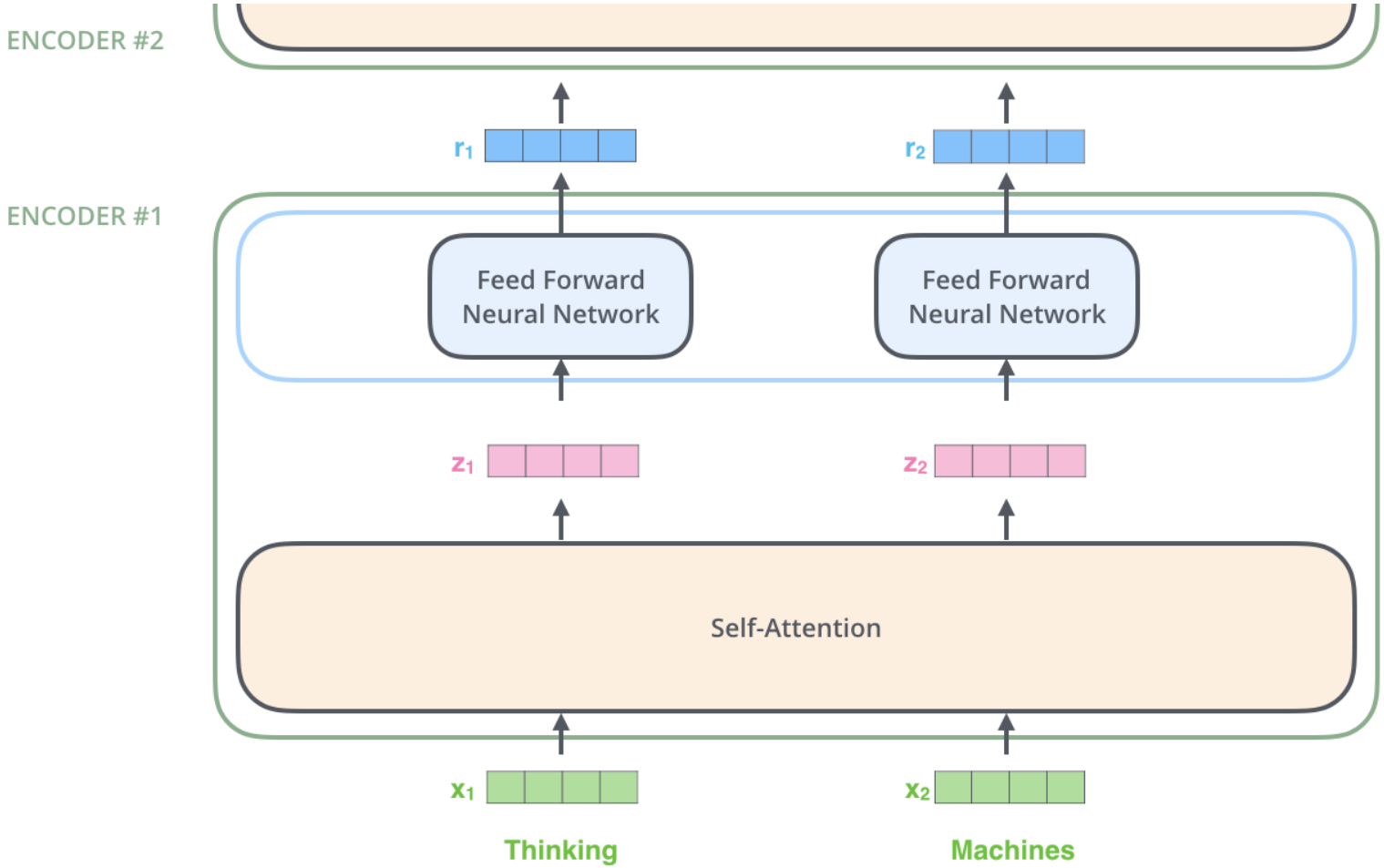
- Der ursprüngliche Transformer ist eine Encoder-Decoder Architektur
- Für Machine Translation ist diese Setup sehr intuitiv: Der Encoder encoded den Ausgangssatz … der Decoder nimmt diese Codierung und baut daraus die Übersetzung

Encoder



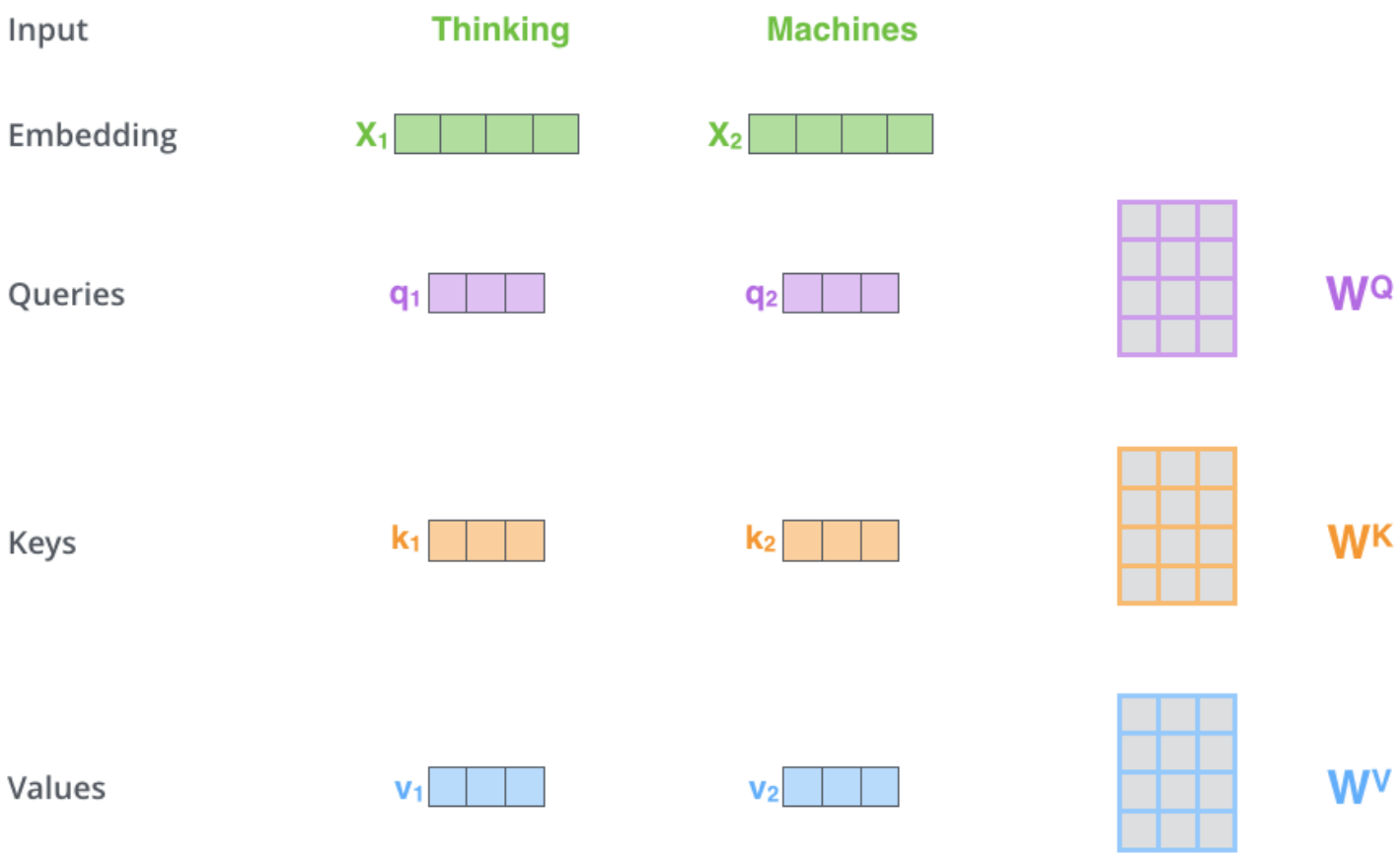
Source: https://jalammar.github.io/illustrated-transformer/

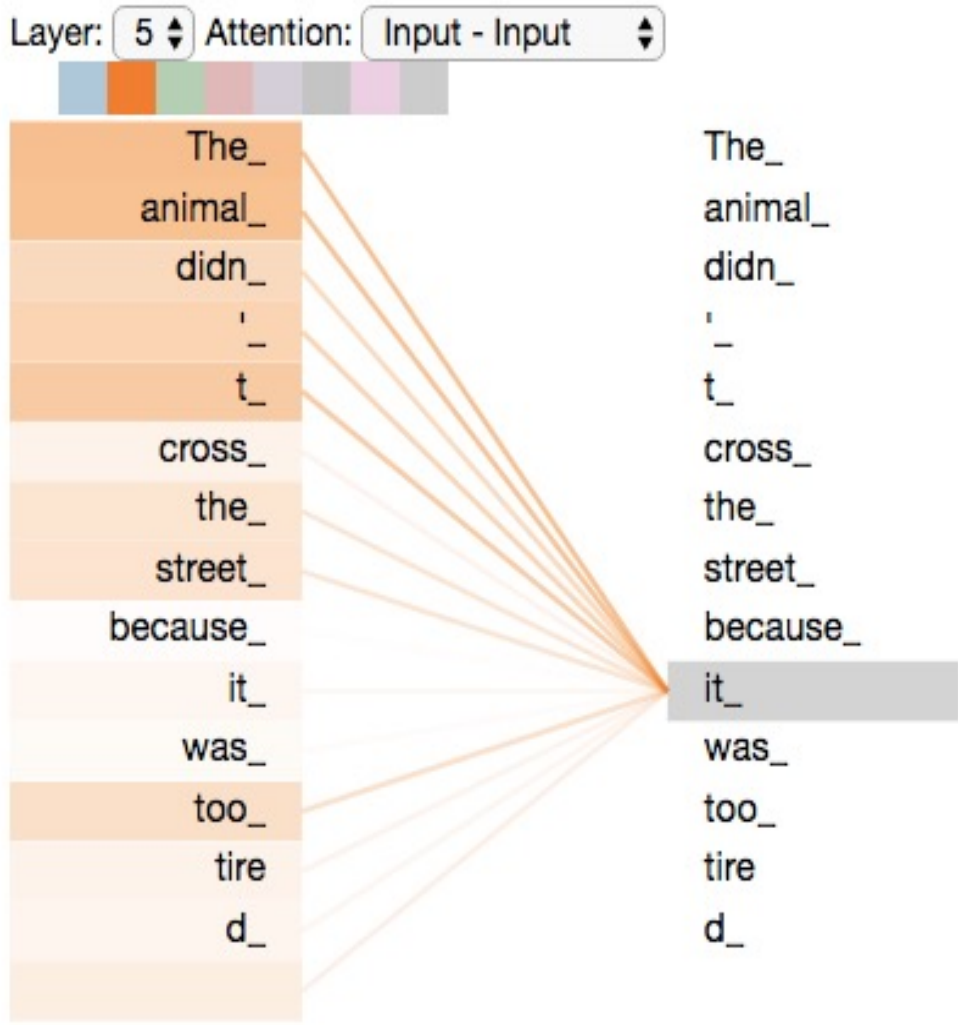
Encoder



Source: https://jalammar.github.io/illustrated-transformer/

Self-Attention



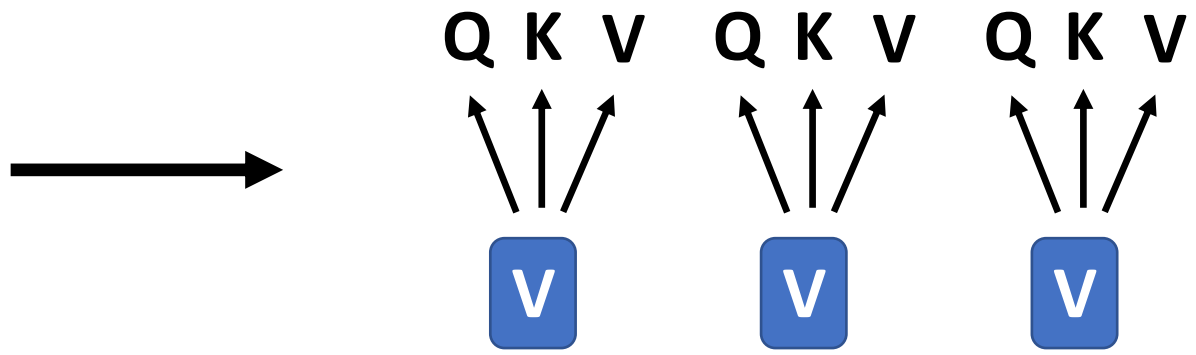


Source: https://jalammar.github.io/illustrated-transformer/

- Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems*. 2017. (10000 citations!)

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the **query Q**, **keys K**, **values V**, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key.

$$Q = vW^Q$$
$$K = vW^K$$
$$V = vW^V$$



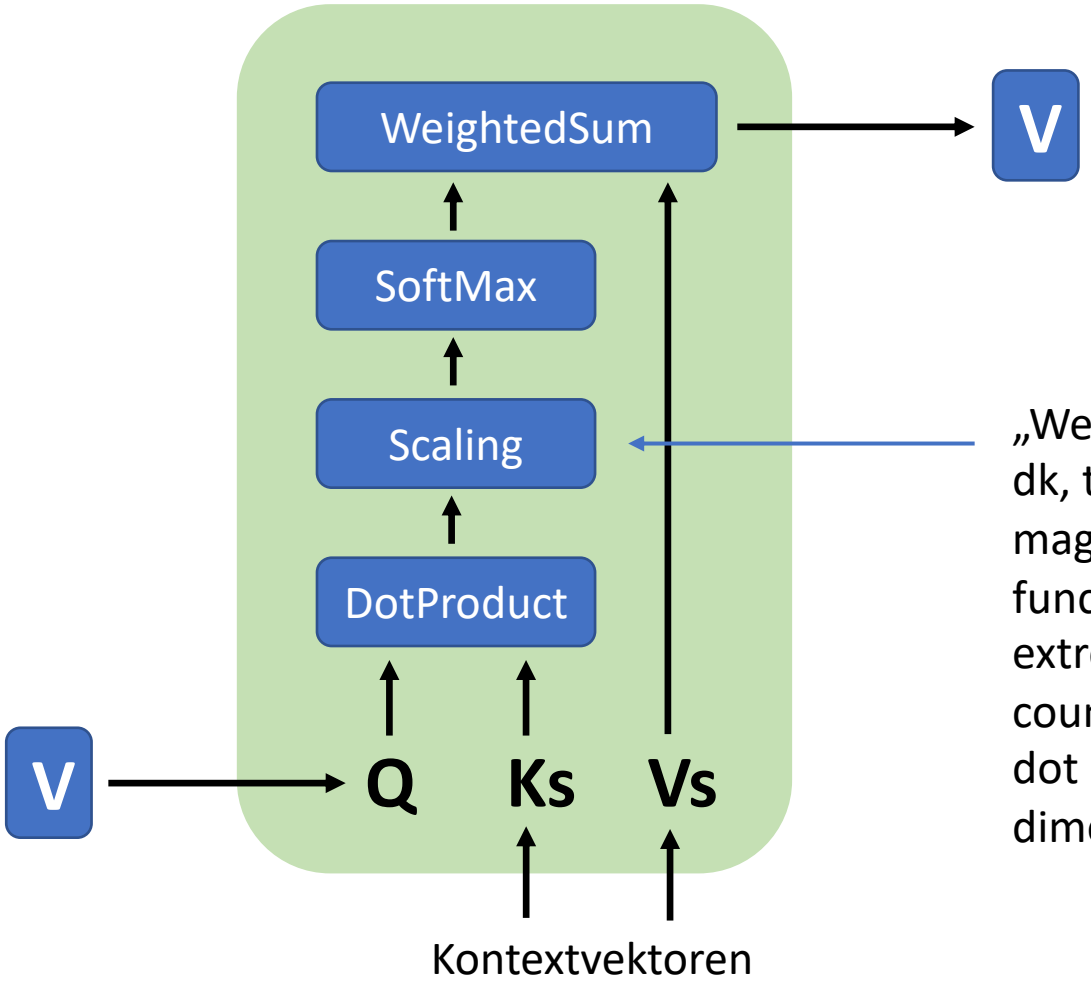
Wie wird der Vektor verwurstet?

1. Er wird in einen Vektor **Q** verwandelt.
2. Dann mit den **K**eys des Kontextes verglichen.
3. Und dann aus den **V**alues des Kontextes gewichtet neu berechnet.

$$Q = VW^Q$$

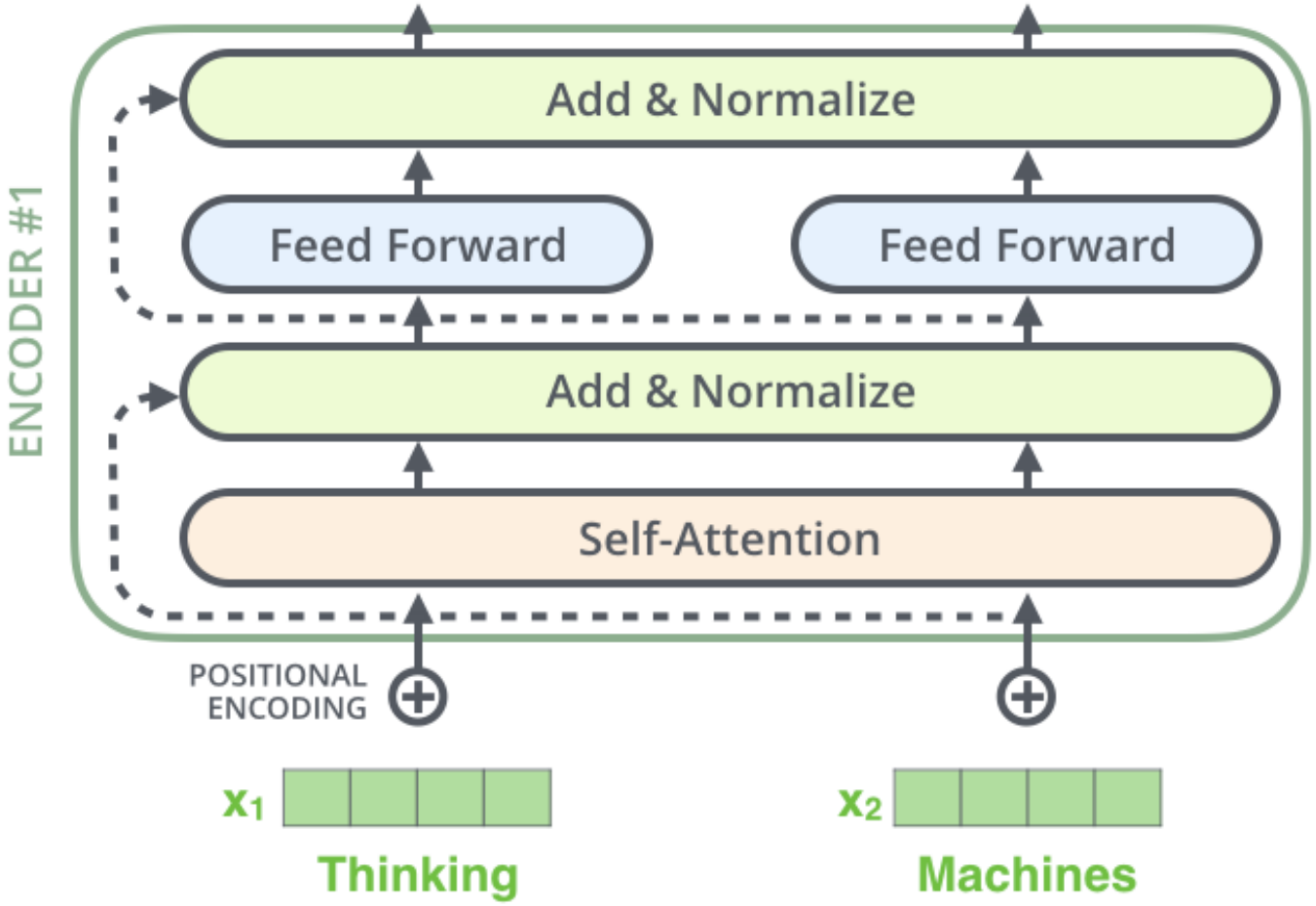
$$K = VW^K$$

$$V = VW^V$$



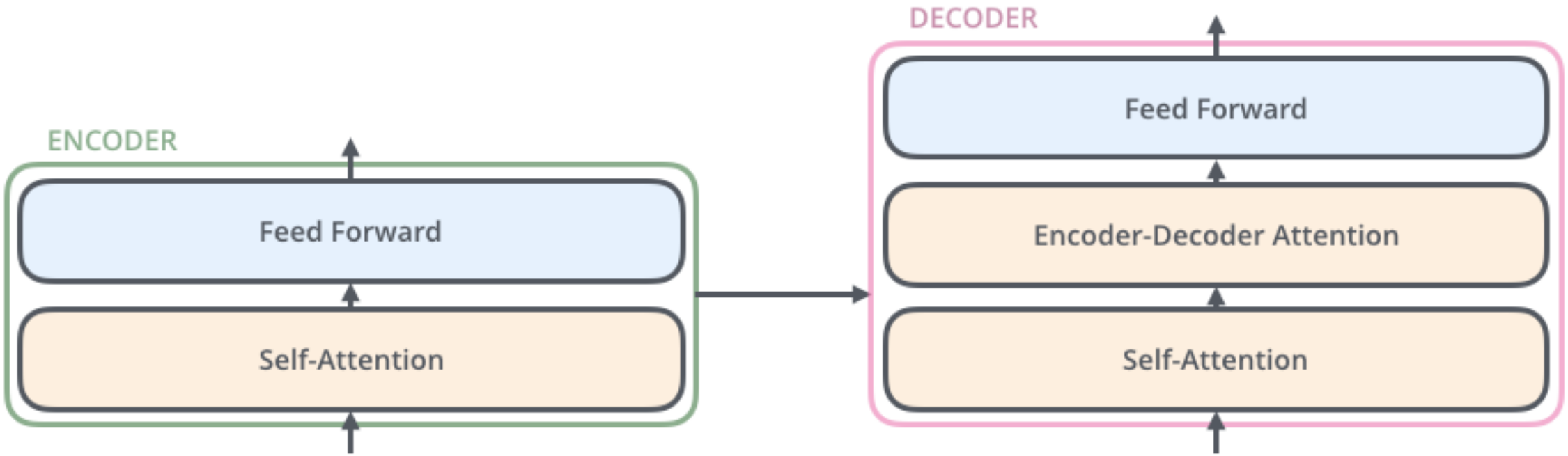
„We suspect that for large values of dk, the dot products grow large in magnitude, pushing the softmax function into regions where it has extremely small gradients. To counteract this effect, we scale the dot products by 1/√dk. (dk = dimension of Q and K)“

Normalisierung



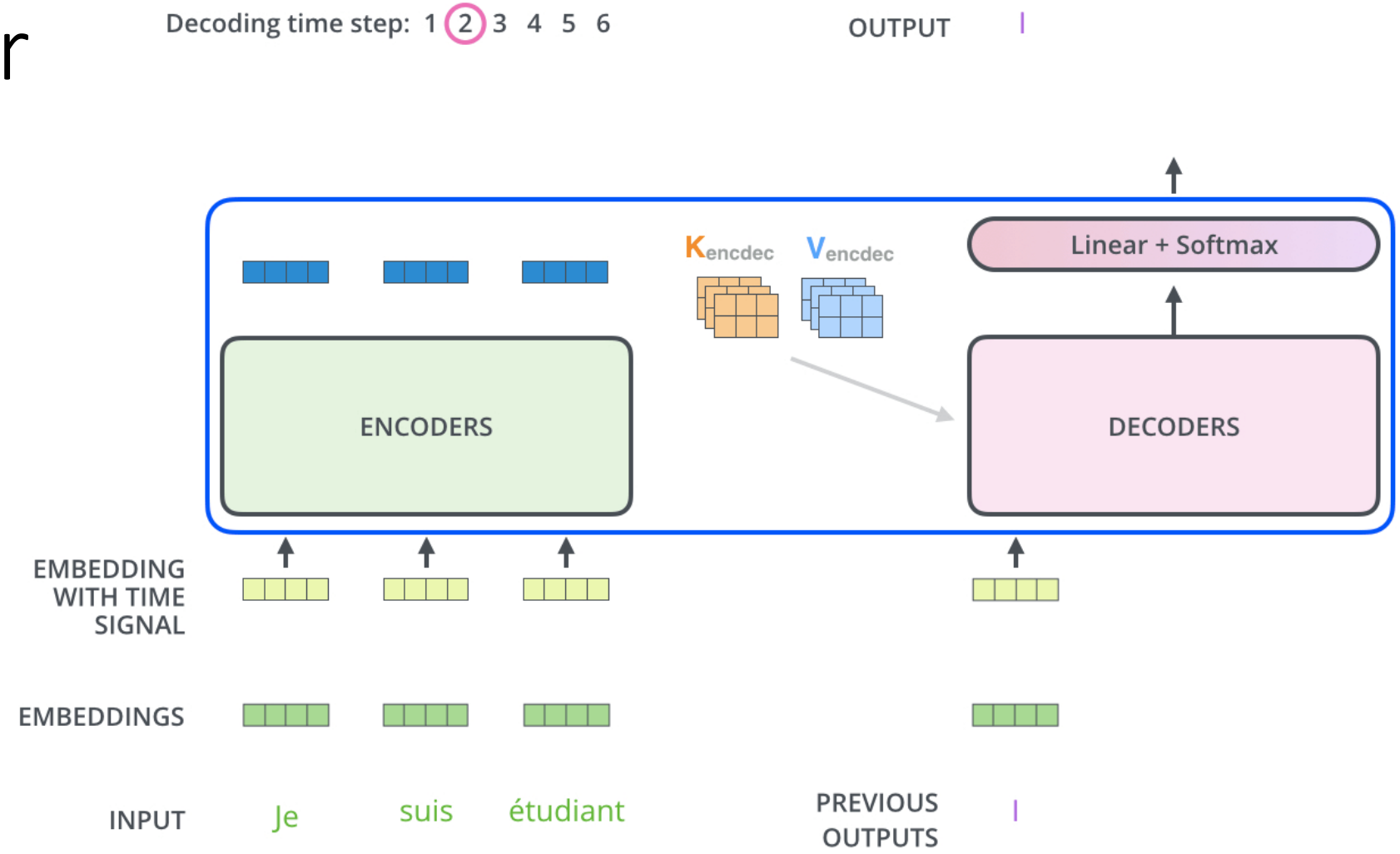
Source: https://jalammar.github.io/illustrated-transformer/

Decoder



Source: https://jalammar.github.io/illustrated-transformer/

Decoder



Source: https://jalammar.github.io/illustrated-transformer/