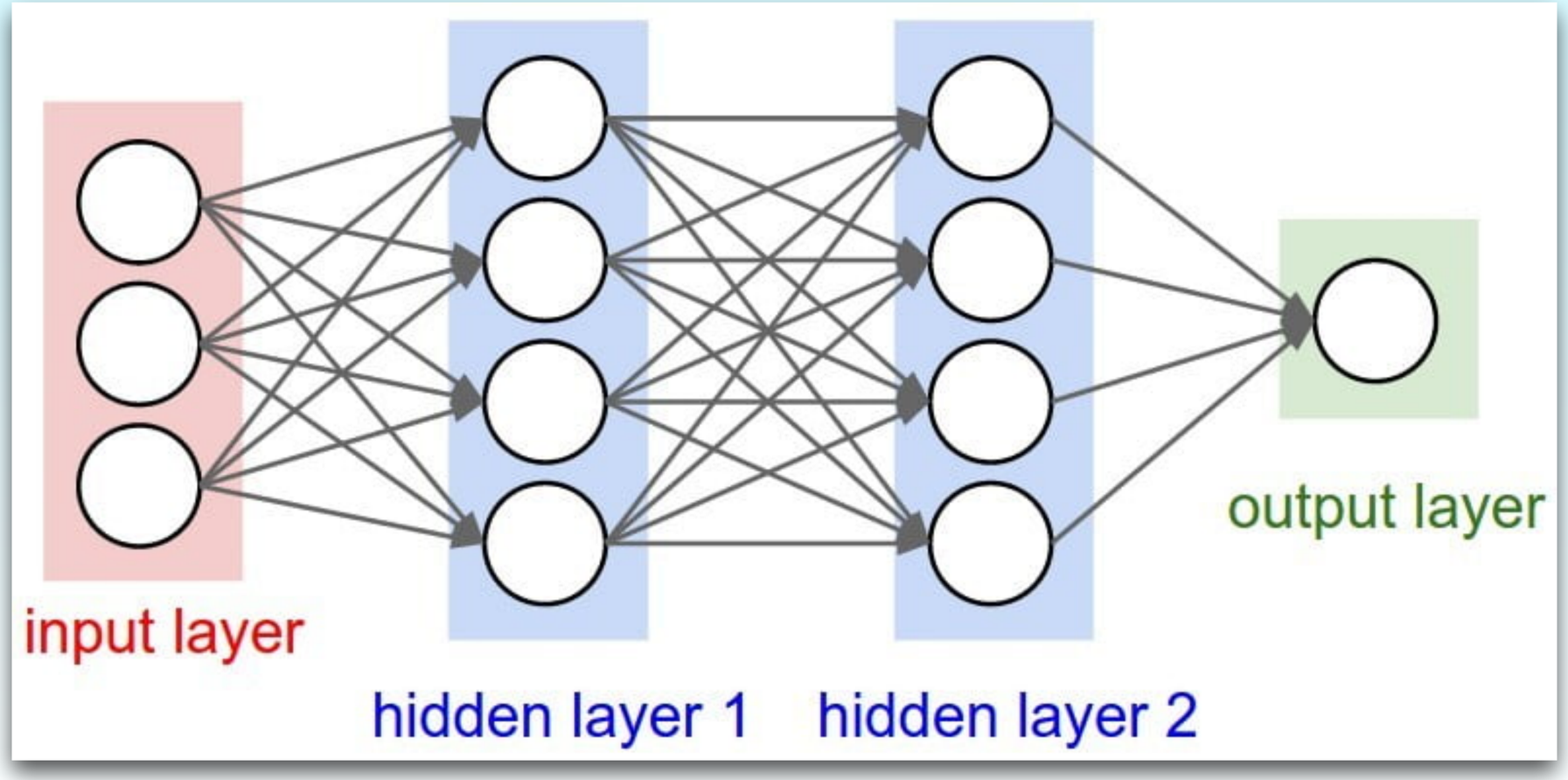
Transformers

LauzHack Deep Learning Summer Bootcamp

Seyed Parsa Neshaei — July 2024

Let’s have a recap!

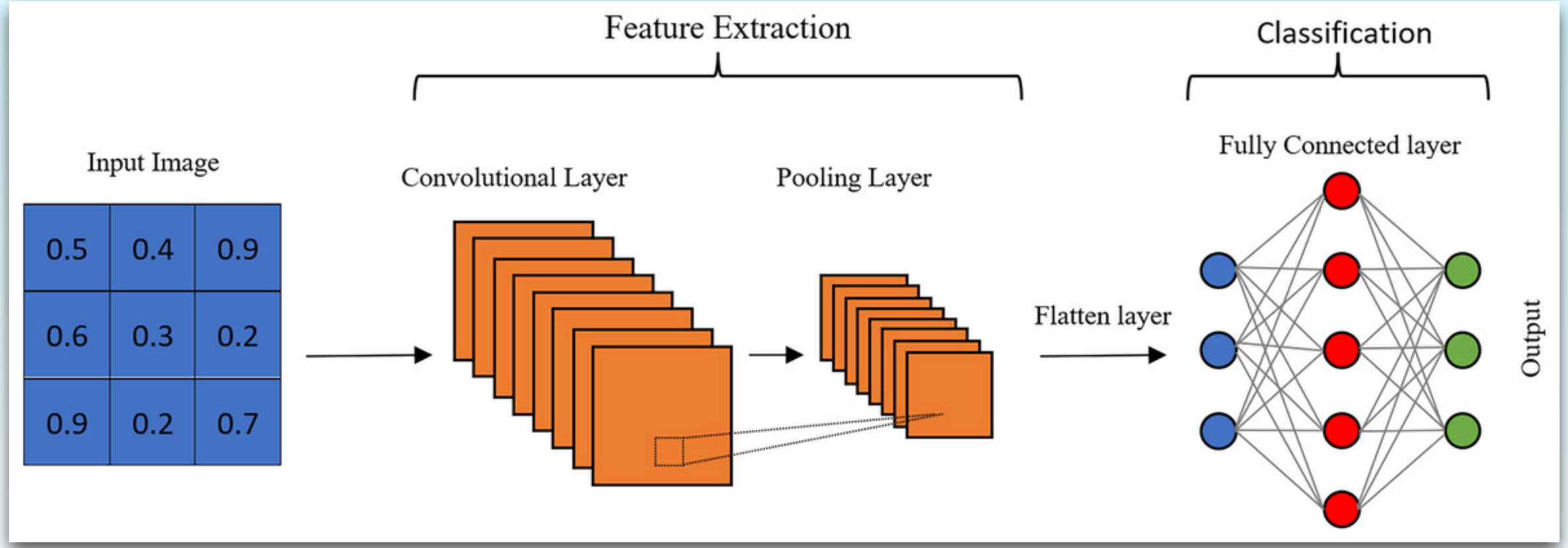
Deep Neural Networks



https://www.bmc.com/blogs/deep-neural-network/

Let’s have a recap!

Convolutional Neural Networks

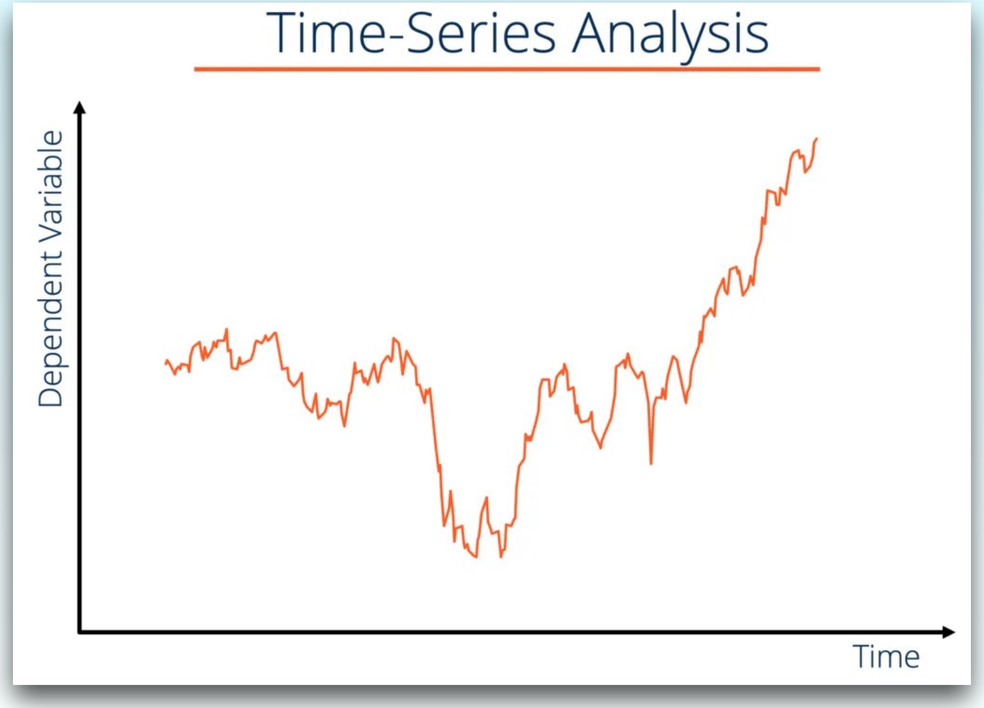


https://peerj.com/articles/cs-1395/

A new task!

Time series

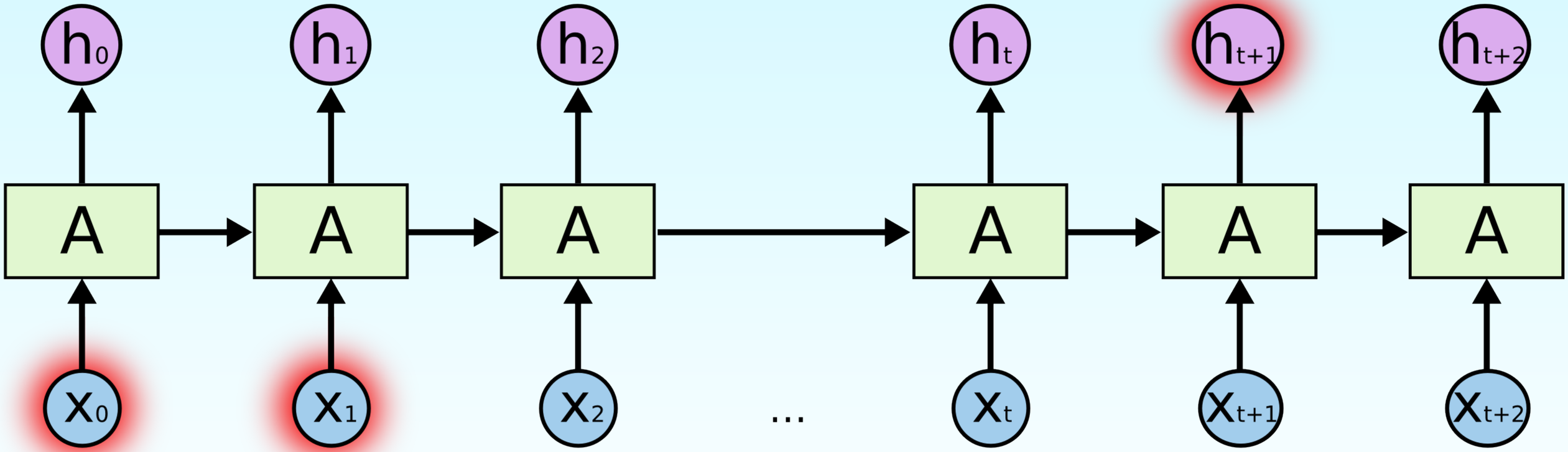
- Data measured at regular intervals, or step-by-step. Examples:
  - Hourly temperature in a city
  - Heights of ocean tides
  - Heartbeats per minute
  - The stock price of a company
  - Answers of students to exam questions
  - A written text (!): *EPFL I like* vs. *I like EPFL*
- Tasks: *classification* or *forecasting*
- Tip: if data semantics change by permutation



https://corporatefinanceinstitute.com/resources/data-science/time-series-data-analysis/

Let’s have a recap!

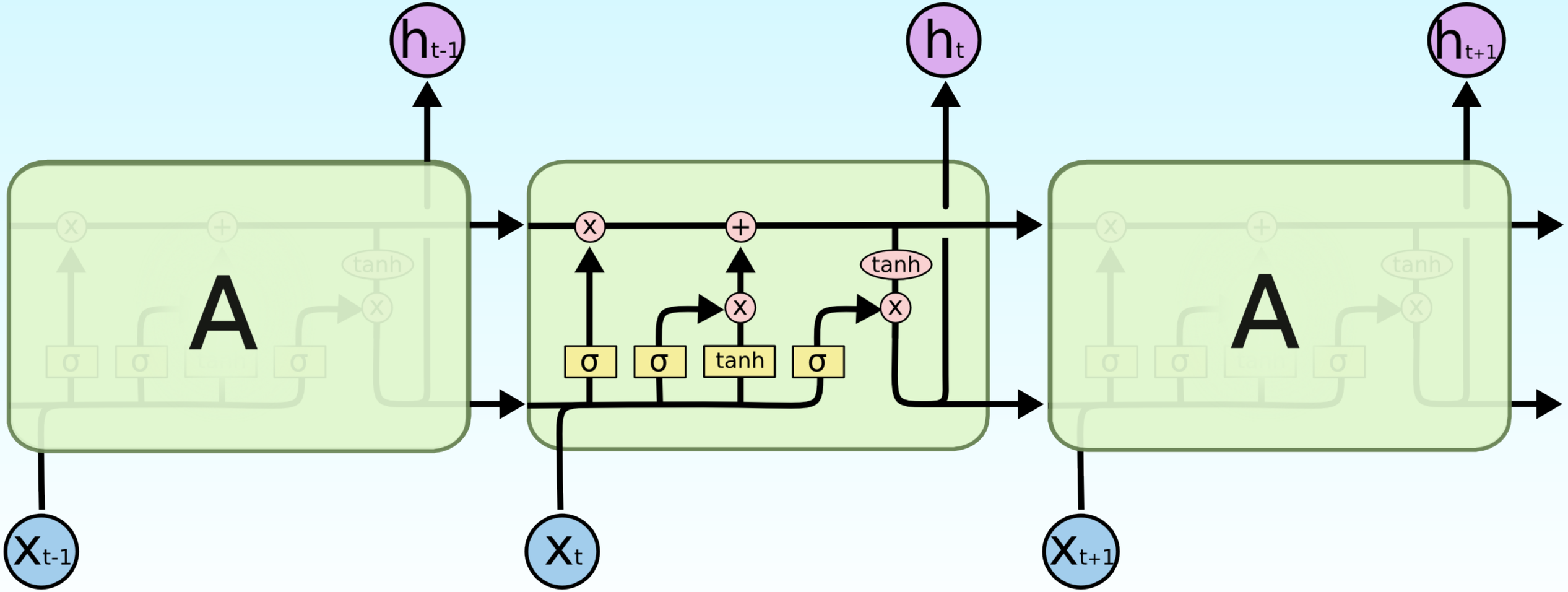
Recurrent Neural Networks



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

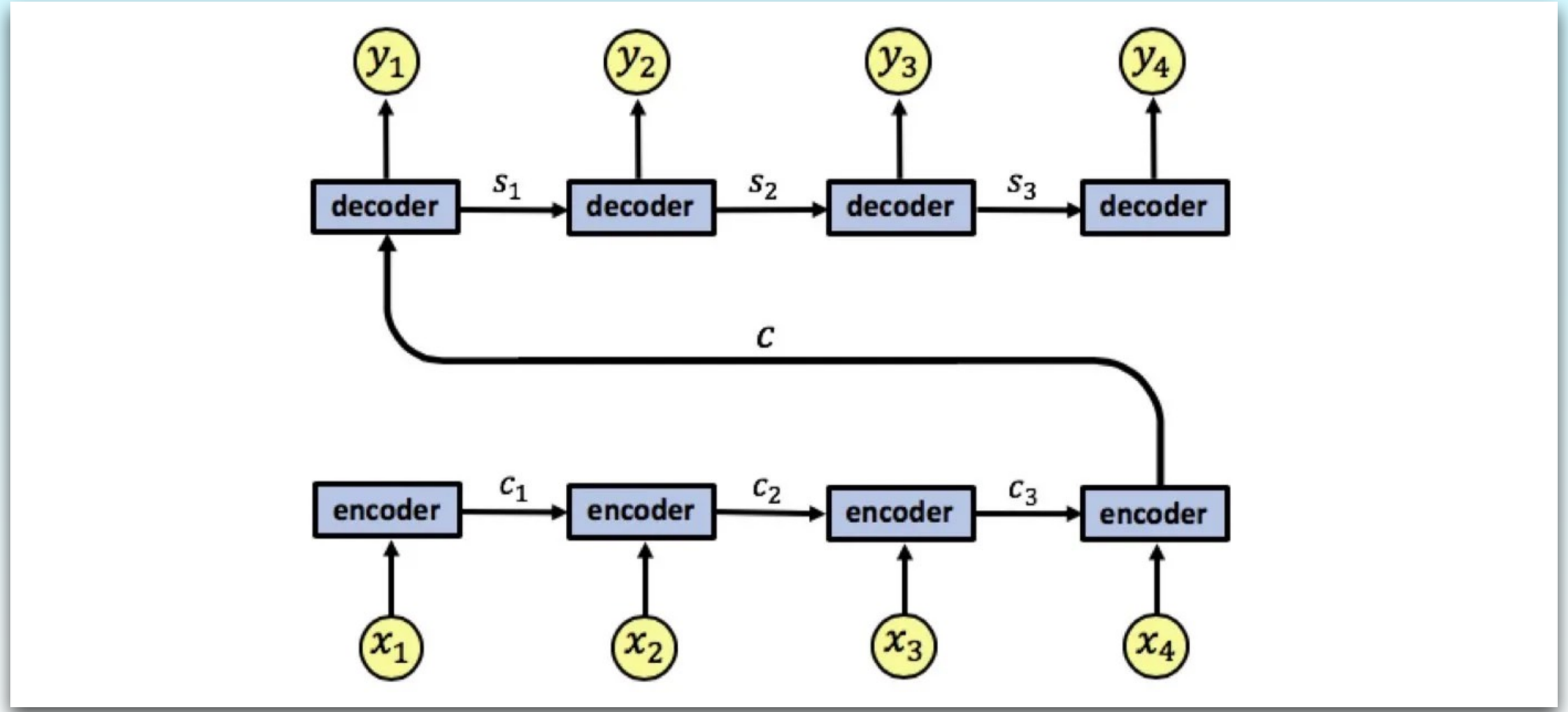
Let’s have a recap!

Long-Short Term Memory Networks



https://colah.github.io/posts/2015-08-Understanding-LSTMs/

Seq2seq Models



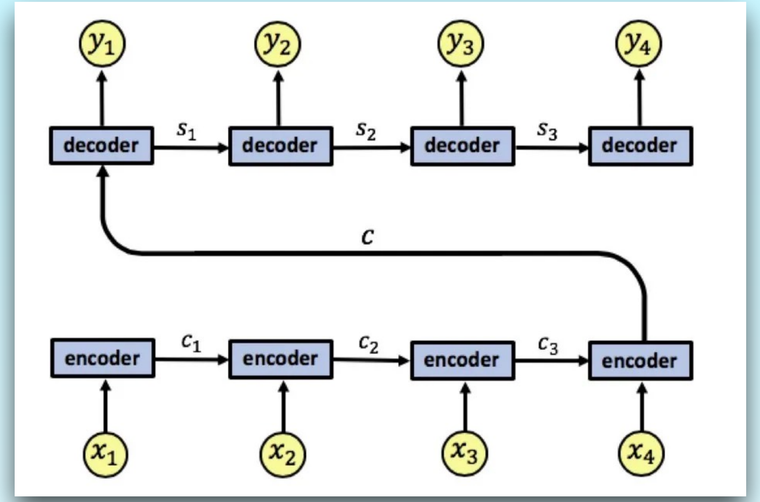
https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

A new method today!

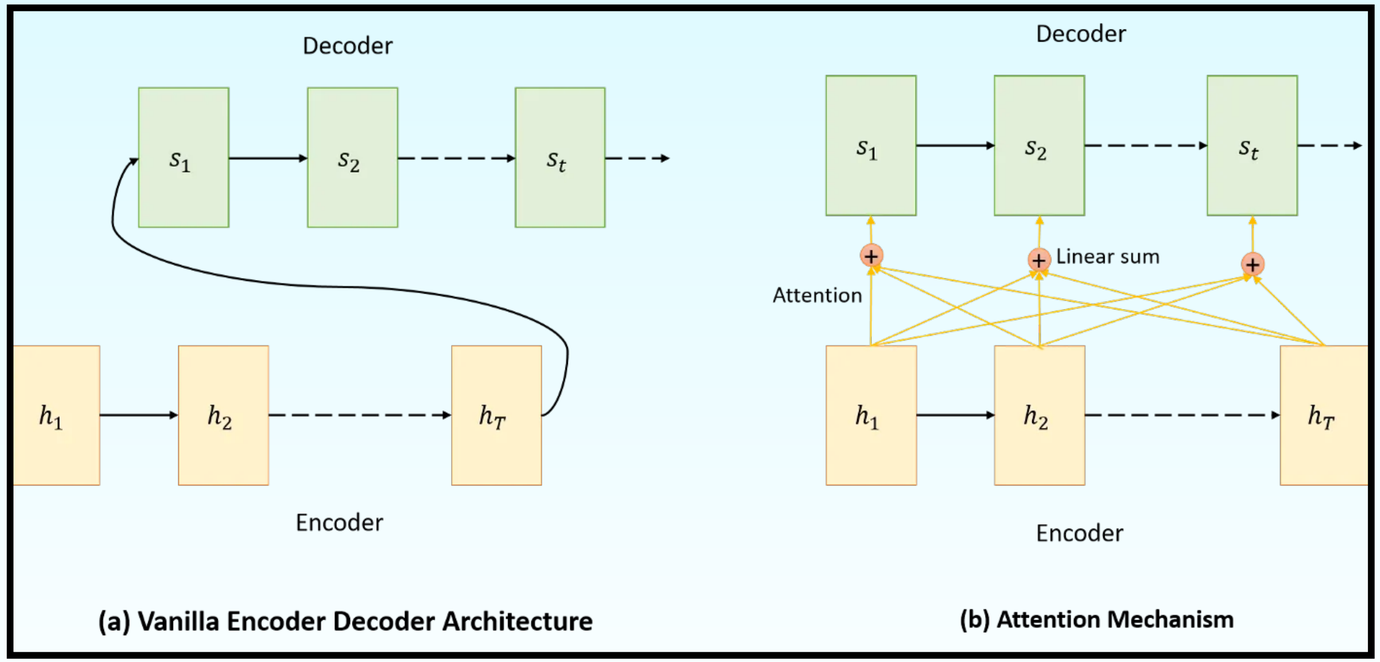
Question: which university/school is this?

https://www.epfl.ch/about/

Attention Mechanism

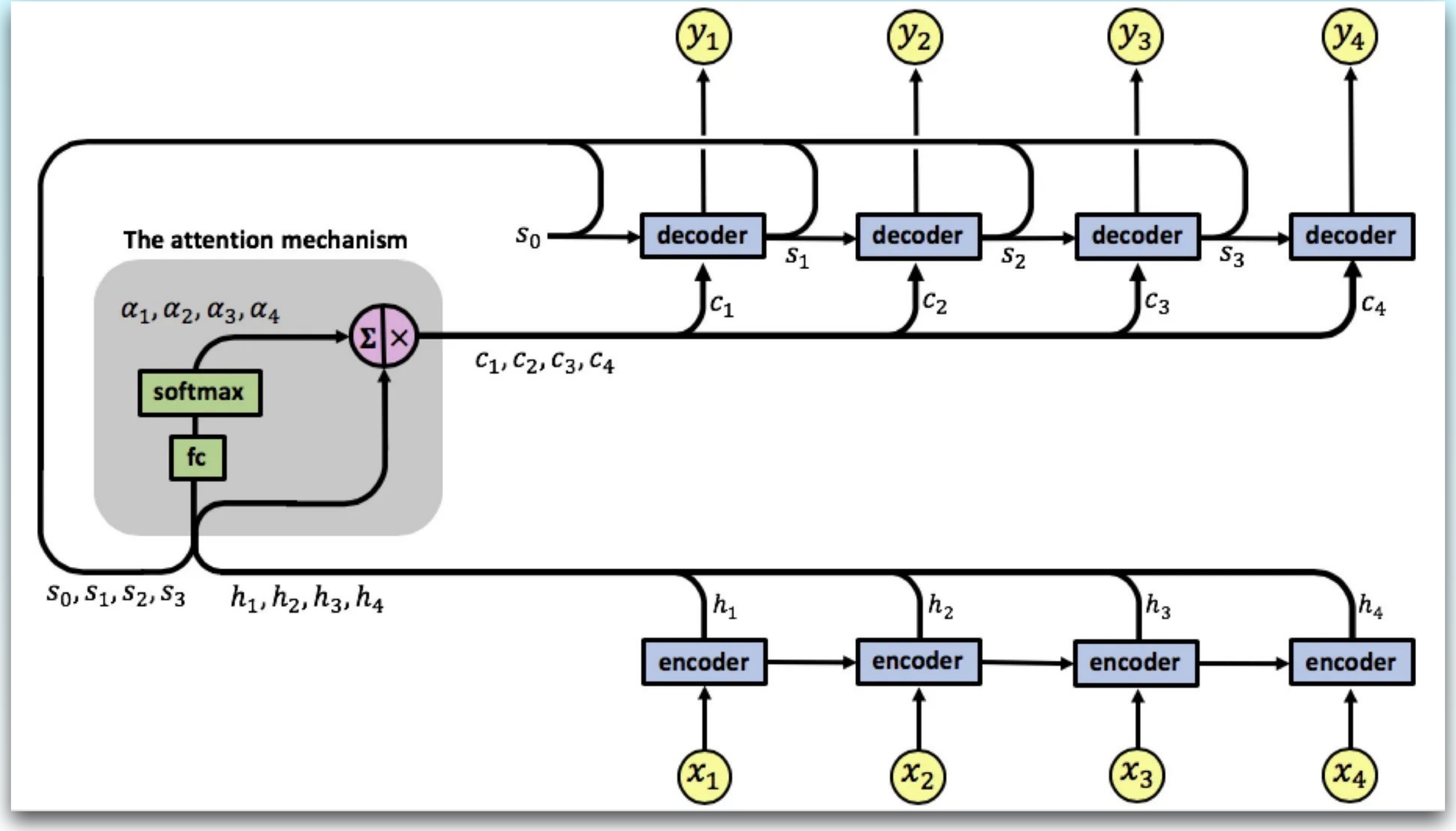


- Until now, the final encoding steps had to to represent the whole sequence into one vector
  - Information loss + the whole decoding from a single vector
- Idea: add an addition to the architecture, helping it to *attend* to the different inputs in the sequence
- Looks at all the encoder hidden states



Images from https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

Attention Mechanism

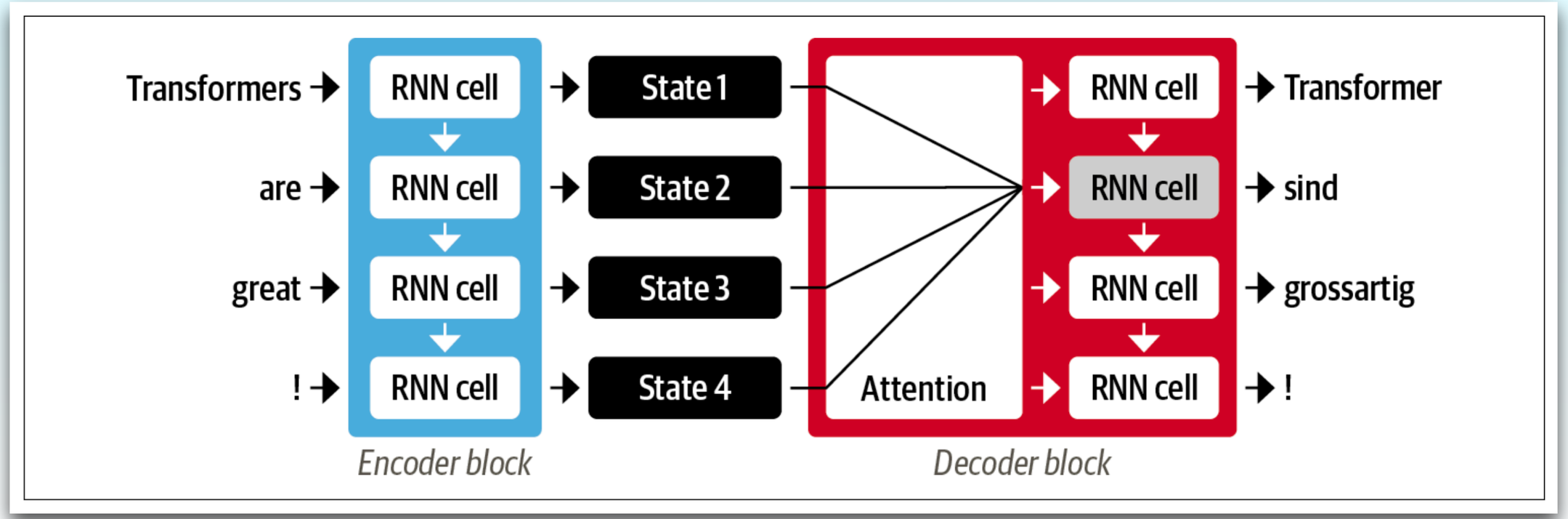


https://medium.com/swlh/a-simple-overview-of-rnn-lstm-and-attention-mechanism-9e844763d07b

Attention Mechanism

From Tunstall et al.:

> “The main idea behind attention is that instead of producing a single hidden state for the input sequence, the encoder outputs a hidden state at each step that the decoder can access.”



Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Attention Mechanism

Learns non-trivial input alignments

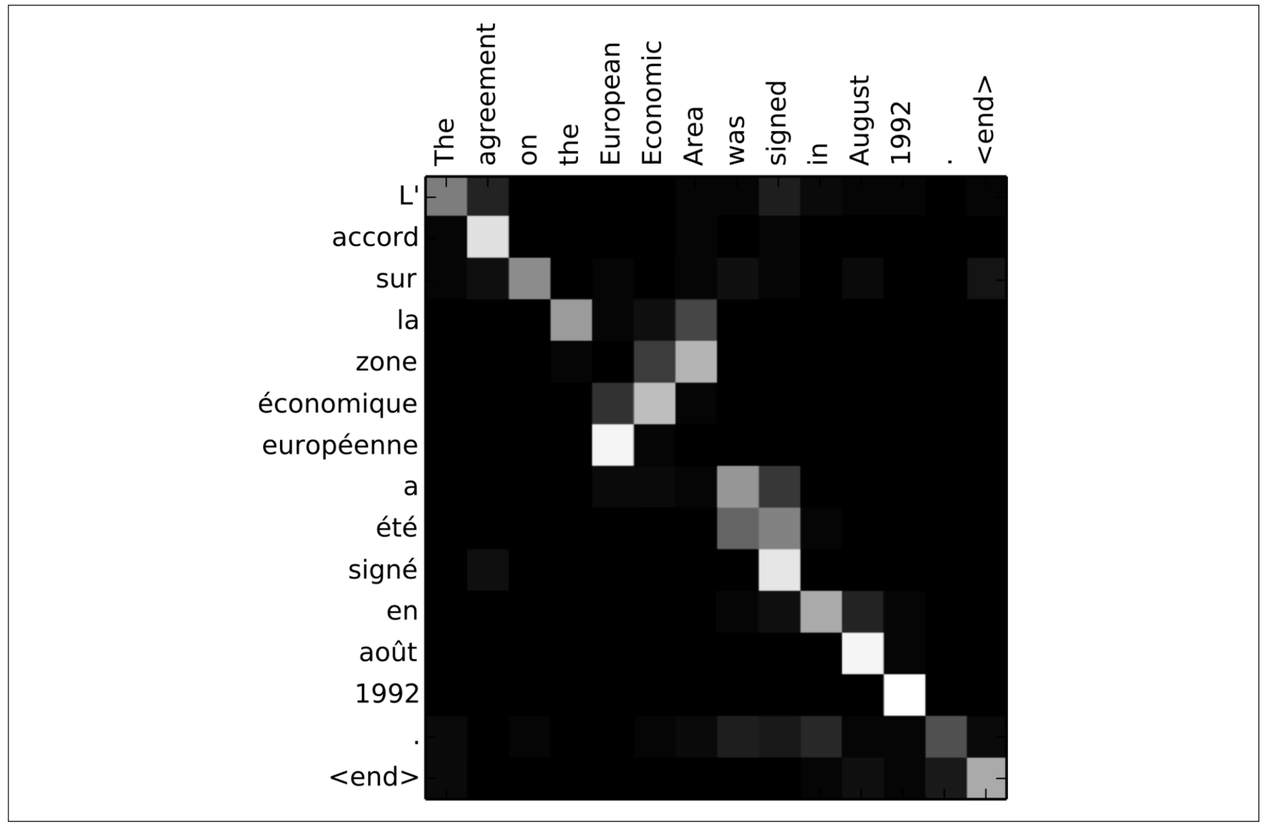
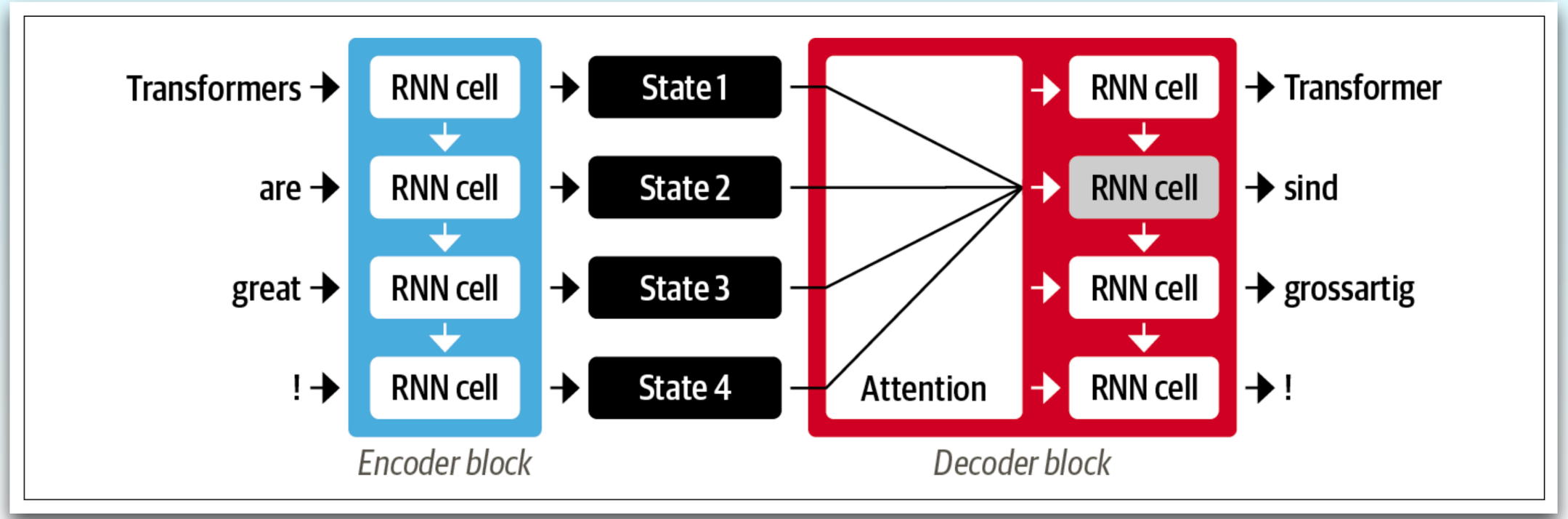


Figure 1-5. RNN encoder-decoder alignment of words in English and the generated translation in French (courtesy of Dzmitry Bahdanau)

Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Any downside that comes to mind?



Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Transformer

- Remove the recurrence! (Self-)attention is all you need!
- Encoder and Decoder (can be only one of them; see in NLP week!)
- Input can be text, image, … (but has small details to take care of!)

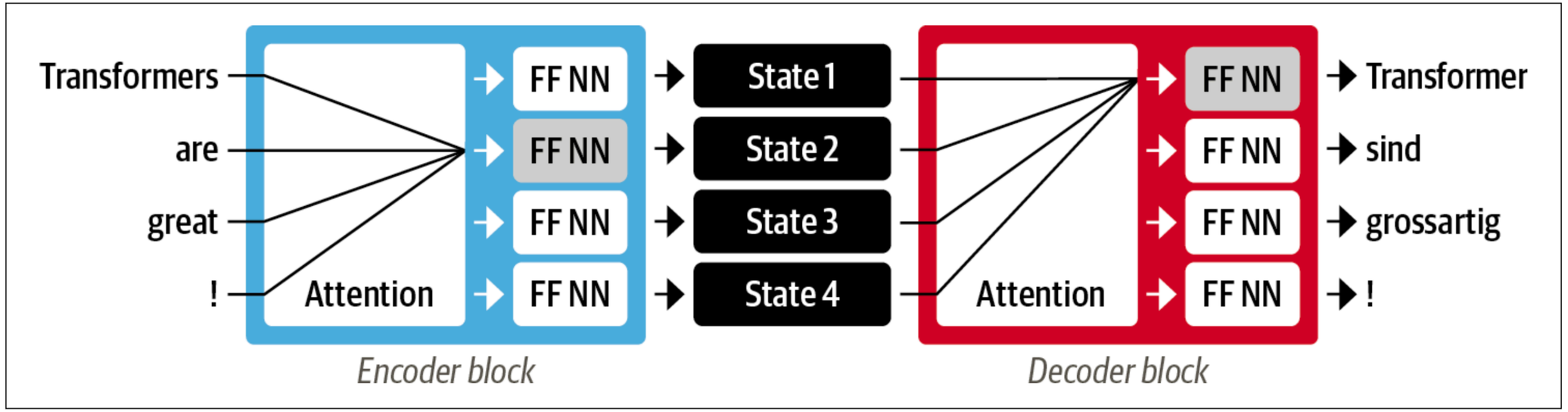


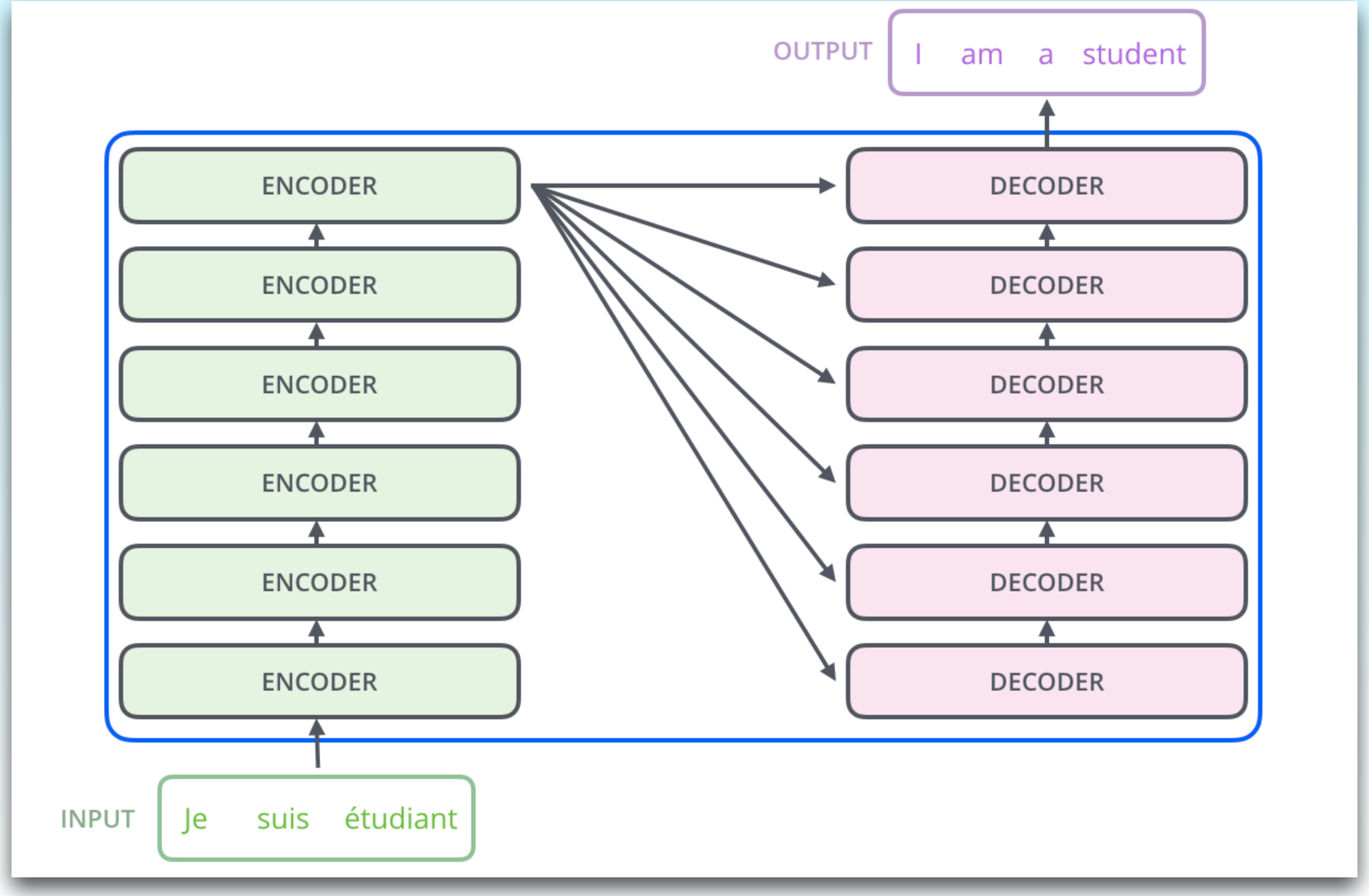
Figure 1-6. Encoder-decoder architecture of the original Transformer

Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Let’s explore how Transformers are designed!

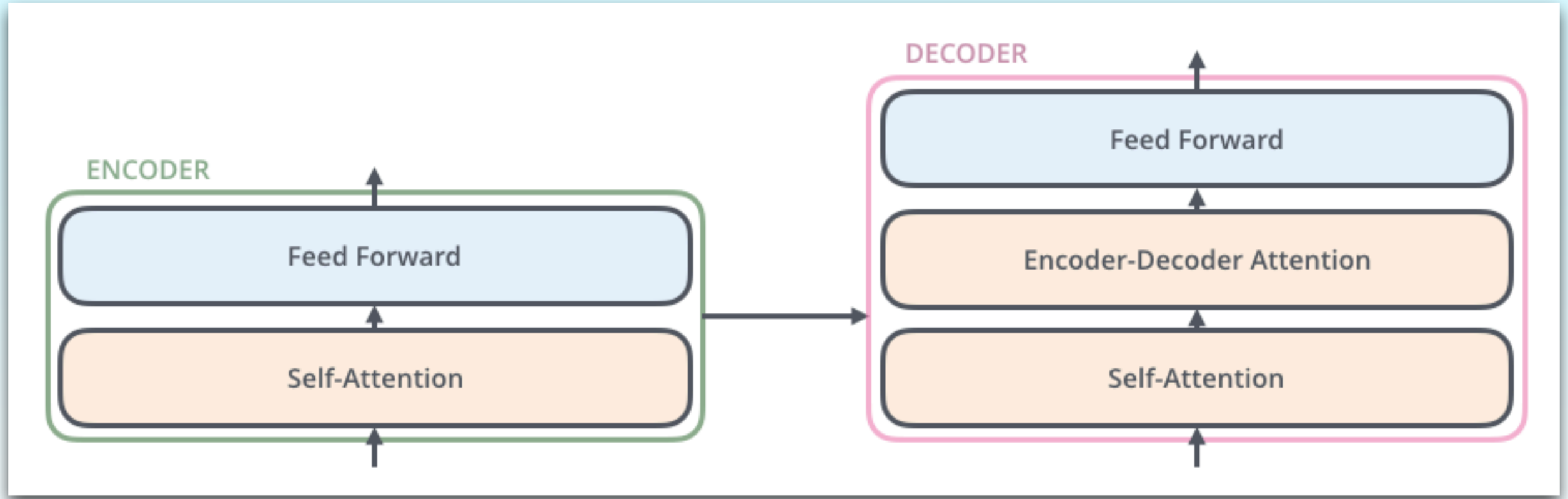
Images and content in this section is from https://jalammar.github.io/illustrated-transformer/

Transformer



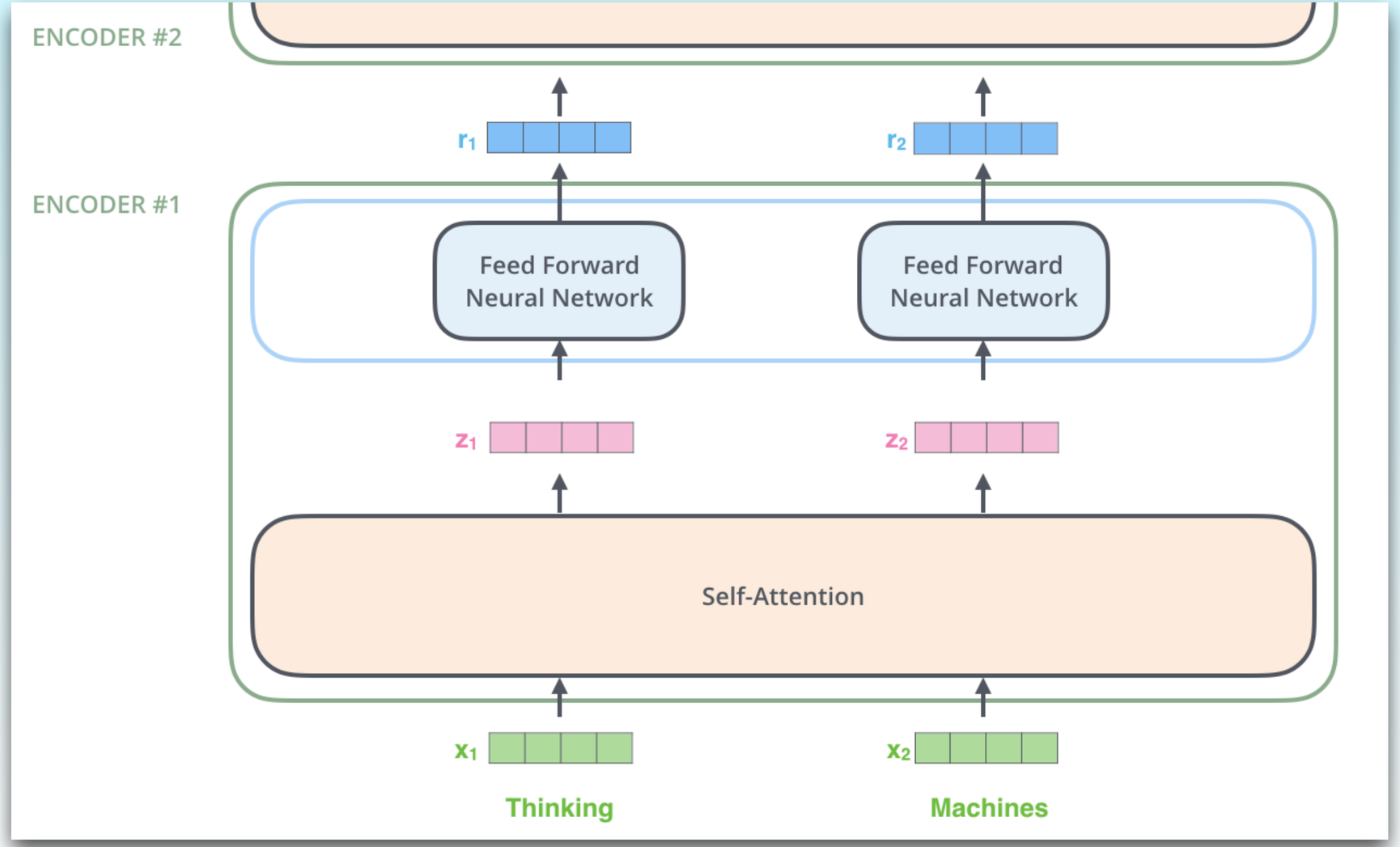
https://jalammar.github.io/illustrated-transformer/

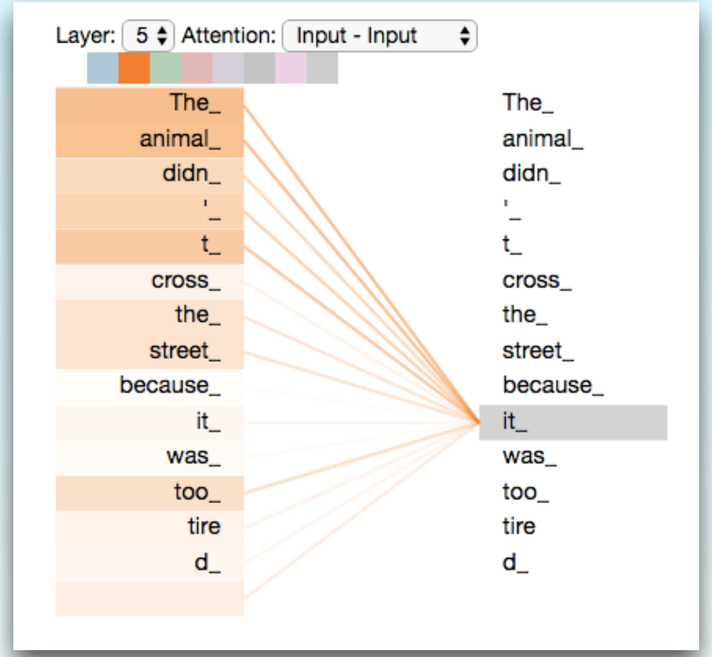
Transformer



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

Transformer Encoder





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

Self-attention — find the differences!

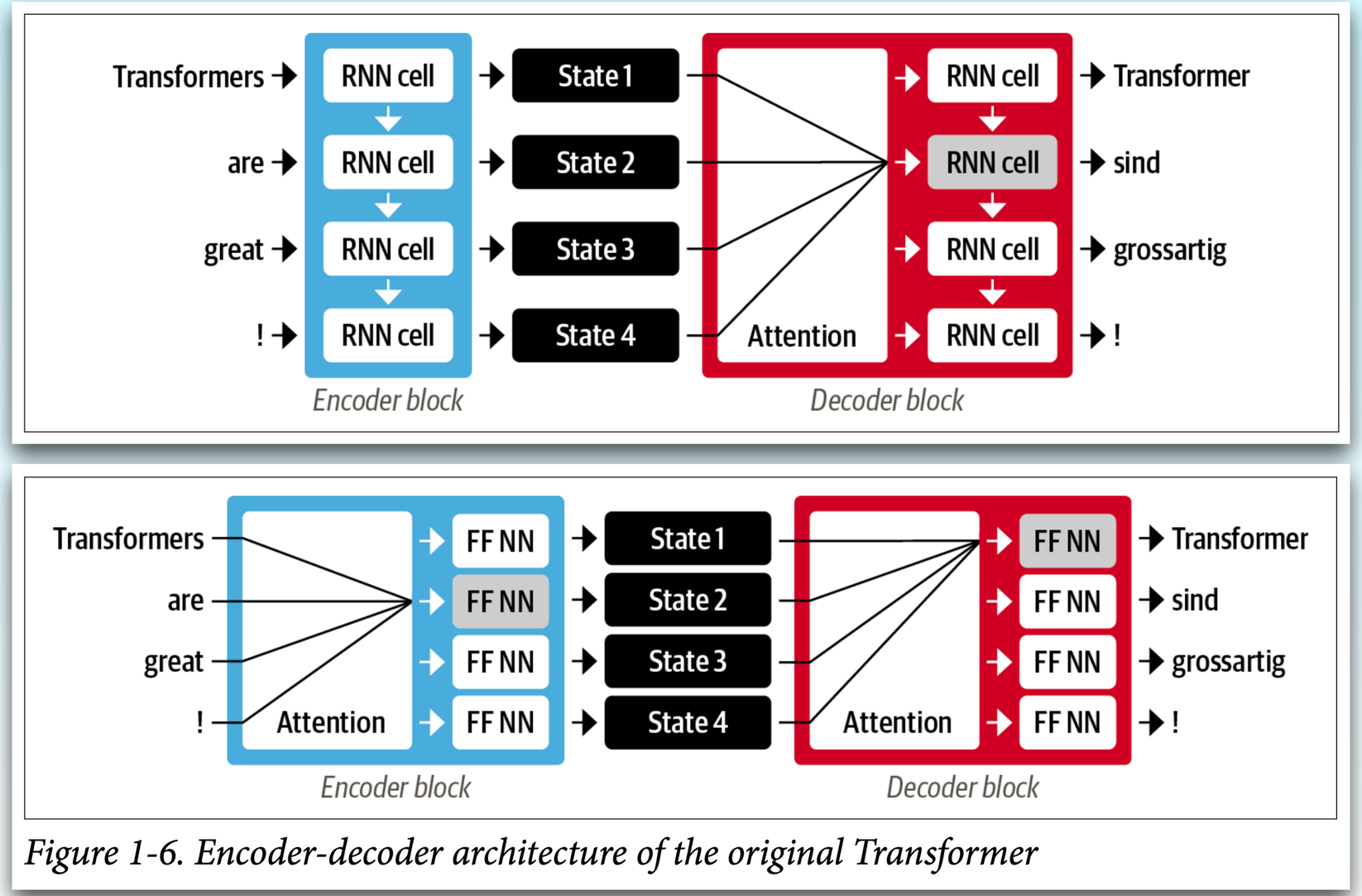


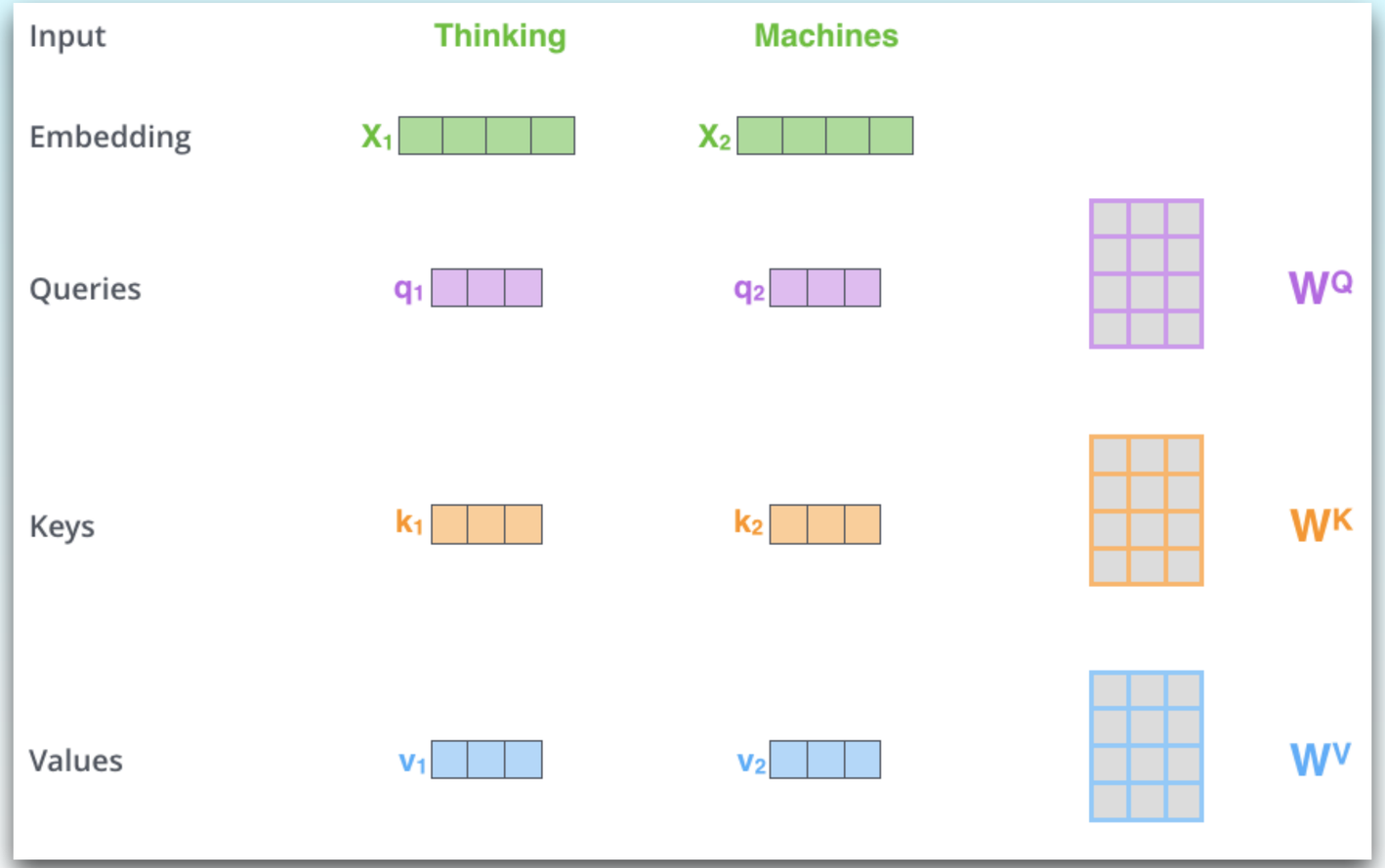
Figure 1-6. Encoder-decoder architecture of the original Transformer

Images from Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Now let’s get a bit more technical

Self-attention Process

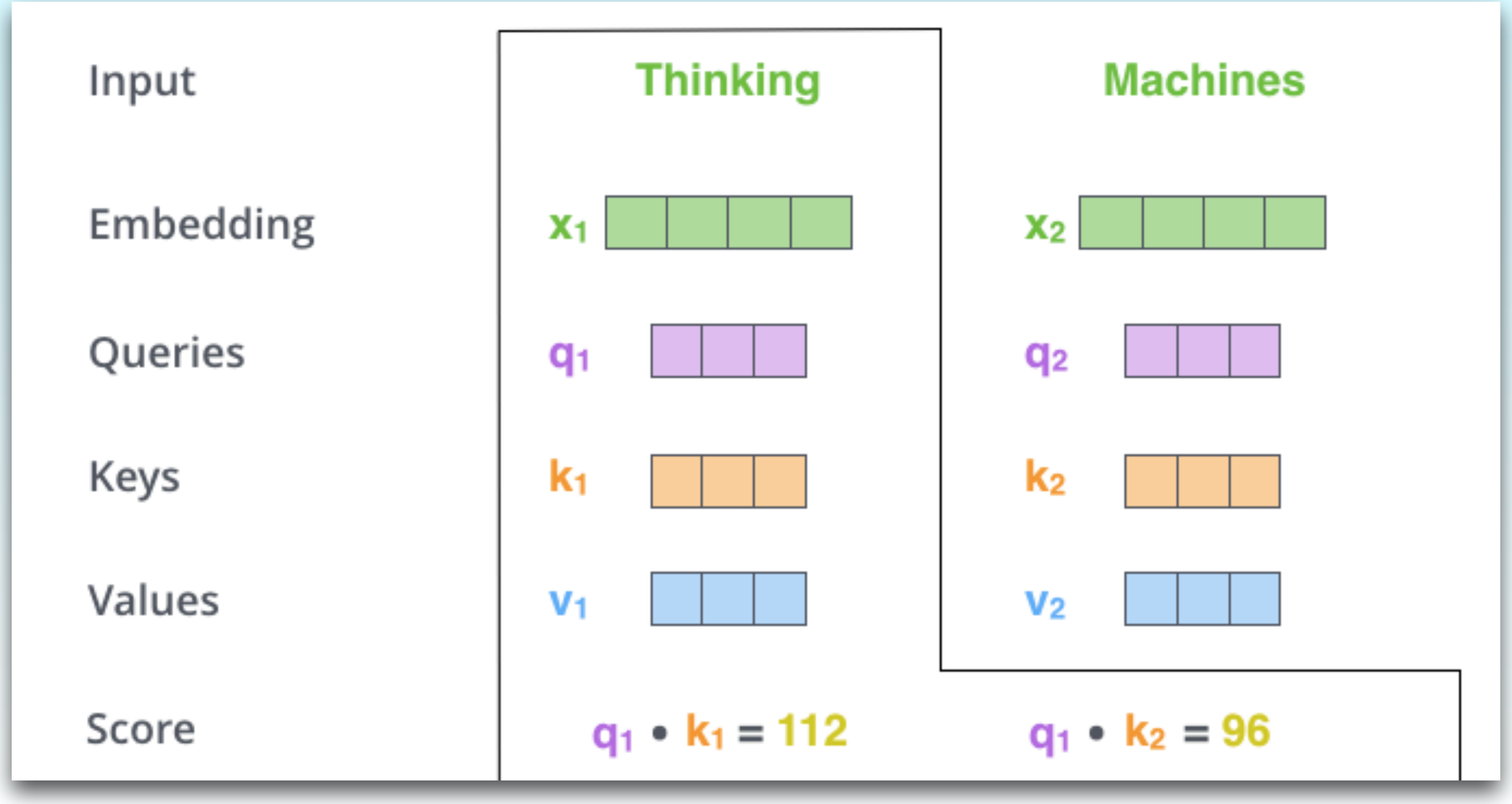
1) Create three vectors from each input of the encoder (can be smaller than input data!)



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

Self-attention Process

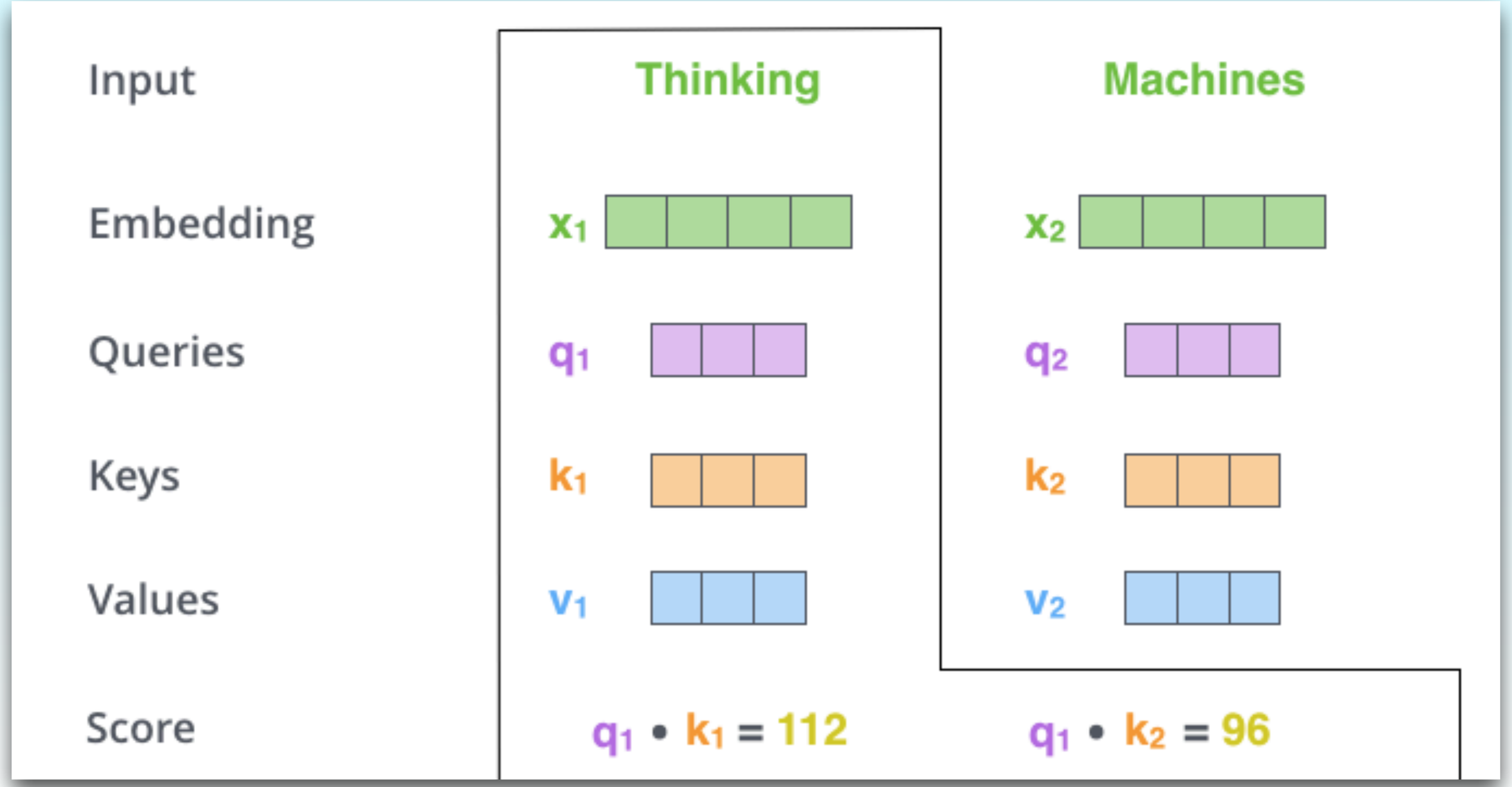
2) For each input, multiply its query with all the keys



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

Self-attention Process

3) Divide scores by sqrt of dim of key vectors (for stable gradients) and put into softmax



$$s\left(x_i\right) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

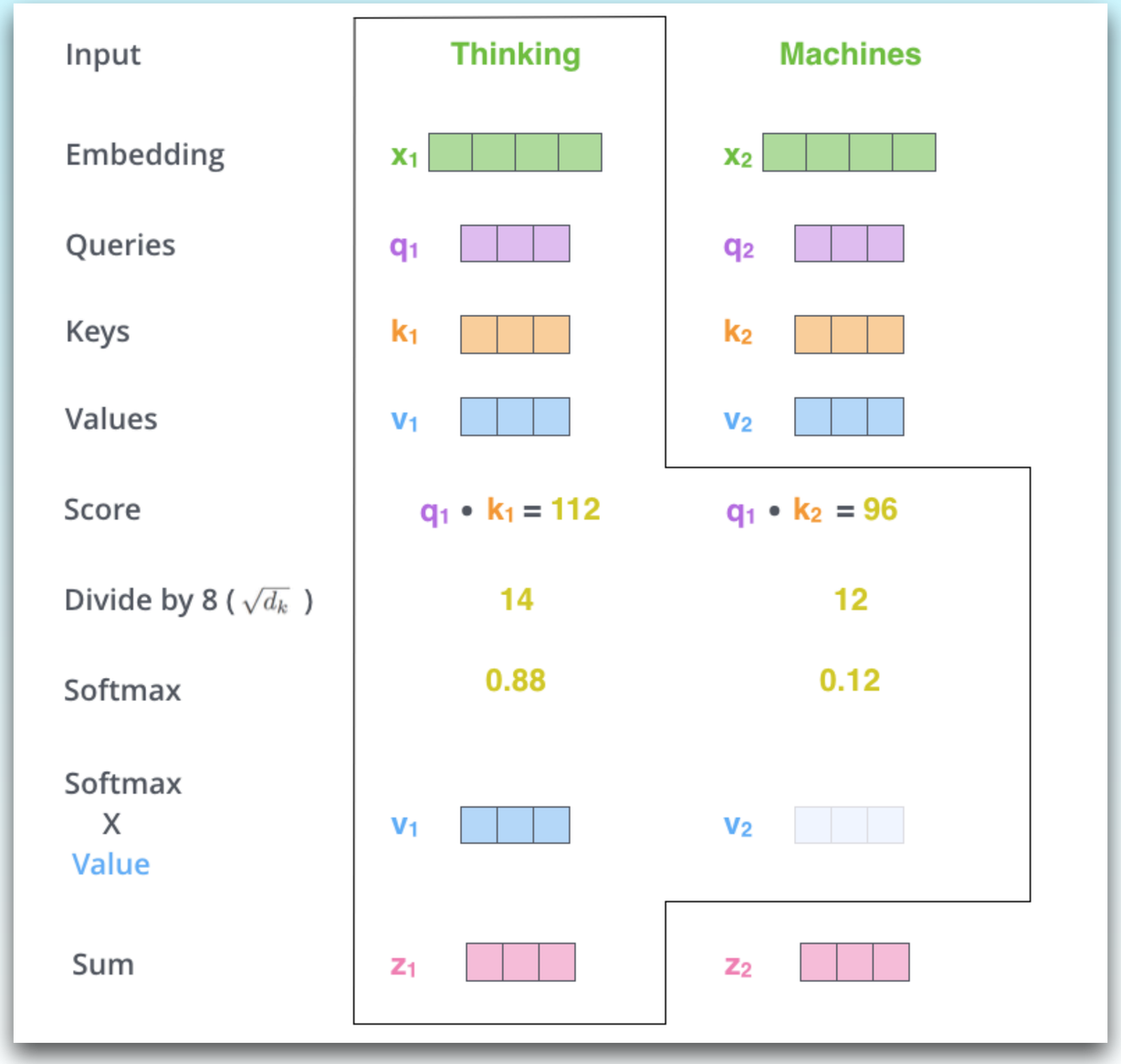
https://jalammar.github.io/illustrated-transformer/

https://developer.apple.com/documentation/accelerate/bnnsactivationfunction/2915301-softmax

Self-attention Process

4) Multiply *value* vectors by softmax (i.e., removing irrelevants), then sum up

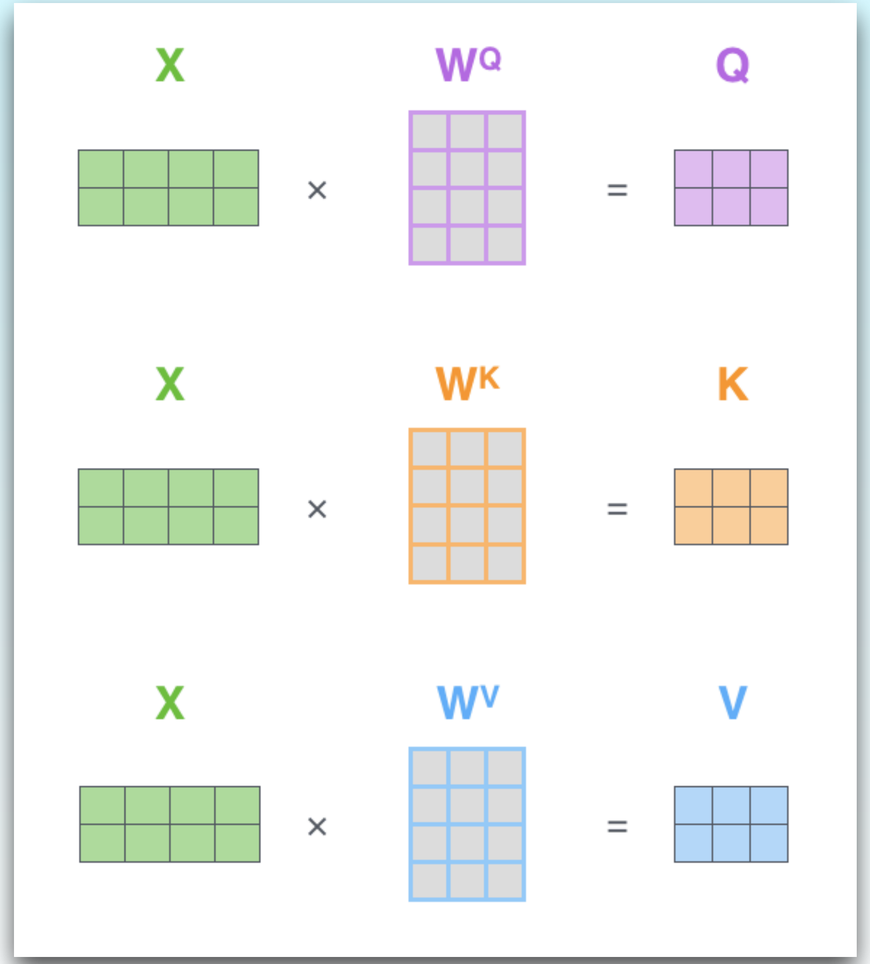
Done!

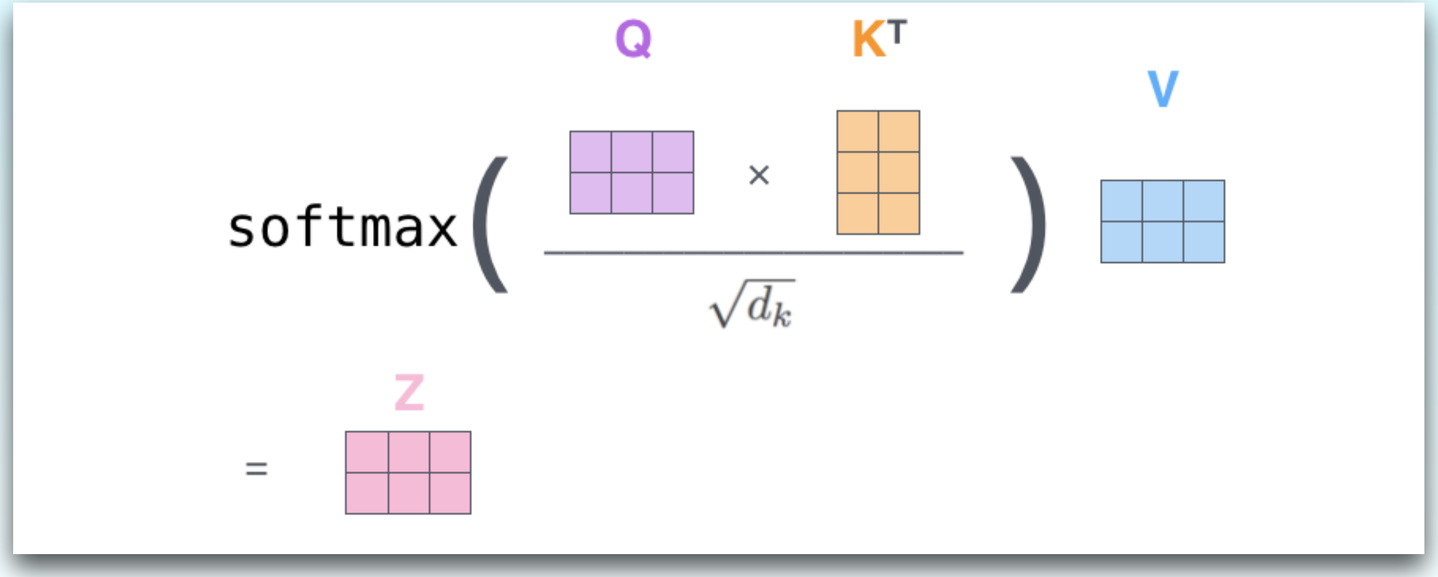


https://jalammar.github.io/illustrated-transformer/

Self-attention In Practice: Matrices

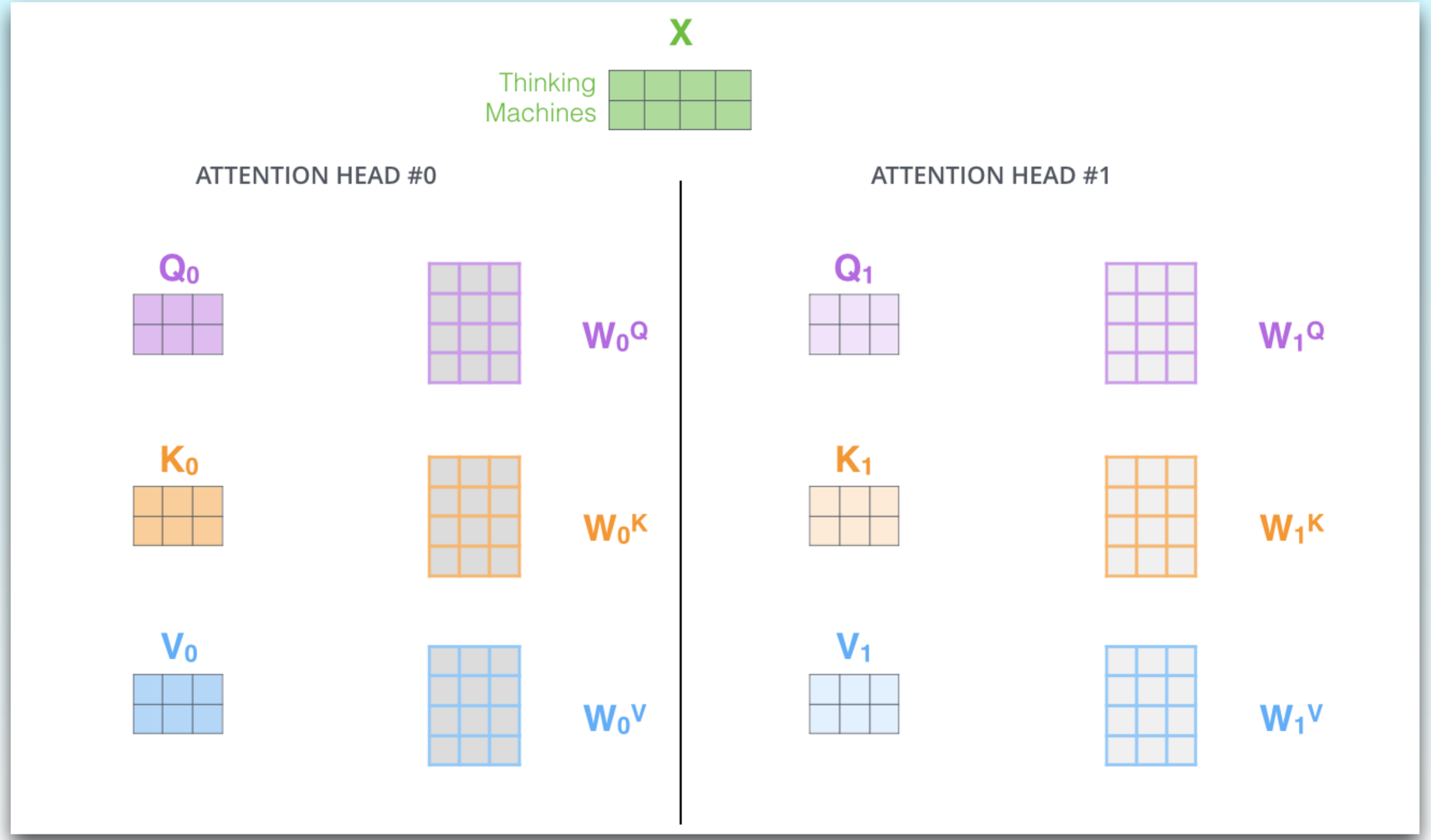
All embeddings into X …!





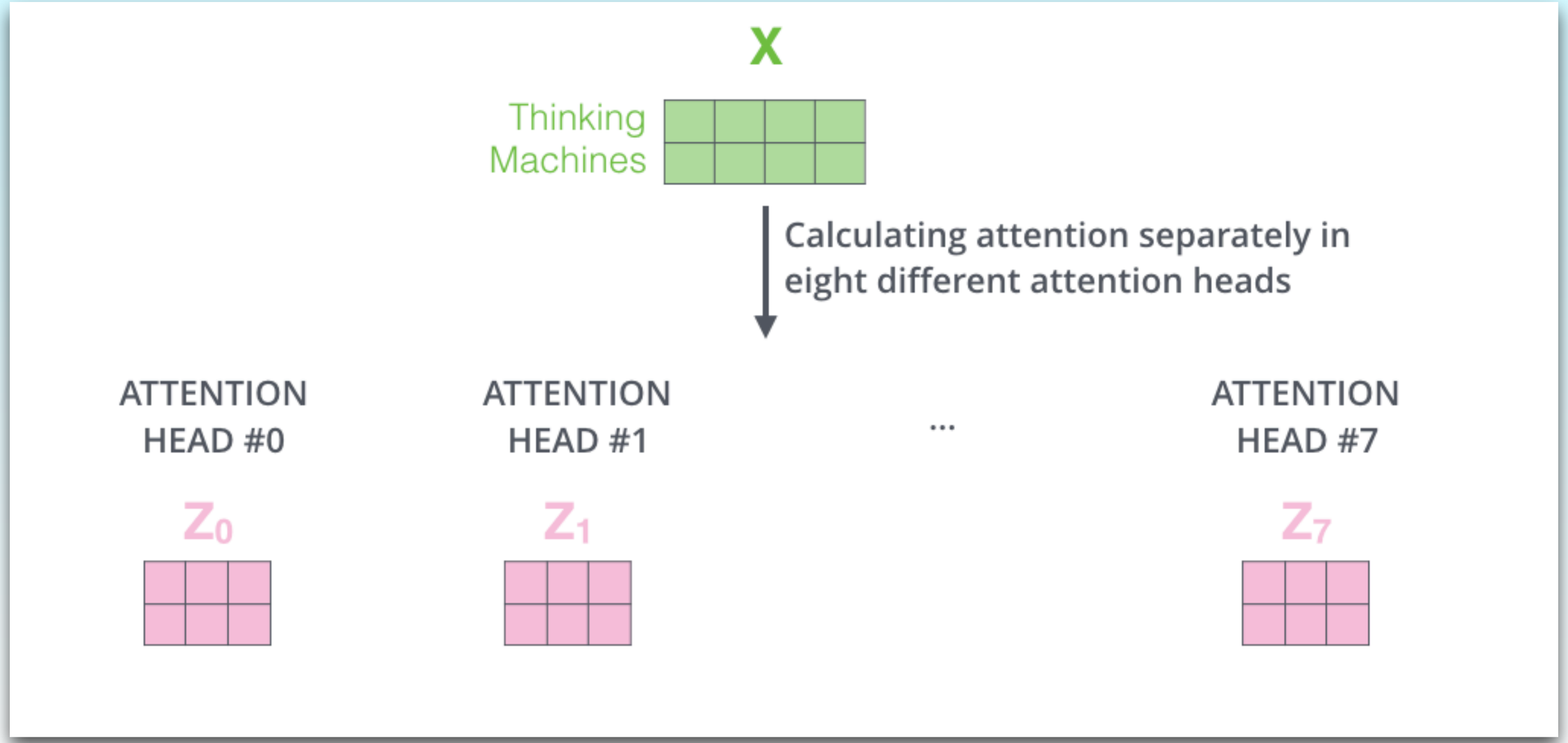
Images from https://jalammar.github.io/illustrated-transformer/

Can be multi-headed



Images from https://jalammar.github.io/illustrated-transformer/

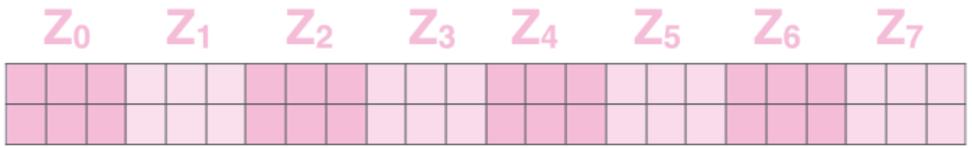
Can be multi-headed



Images from https://jalammar.github.io/illustrated-transformer/

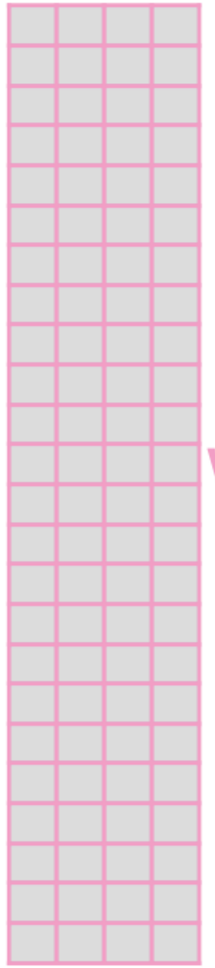
Can be multi-headed

1) Concatenate all the attention heads



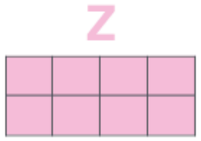
2) Multiply with a weight matrix W^O that was trained jointly with the model

X



3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN

=



Images from https://jalammar.github.io/illustrated-transformer/

What is one problem with this approach?

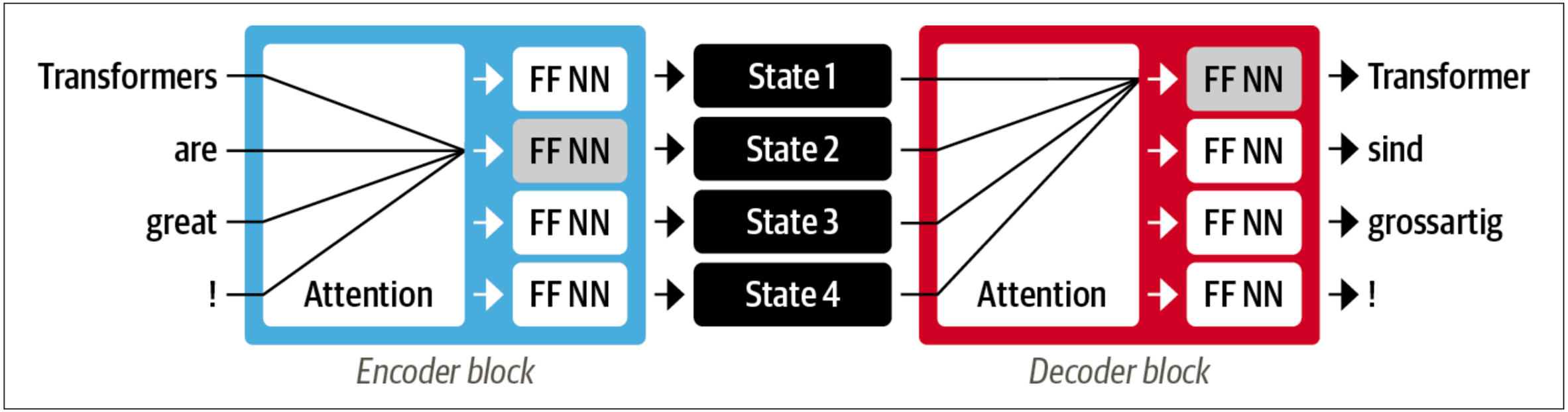
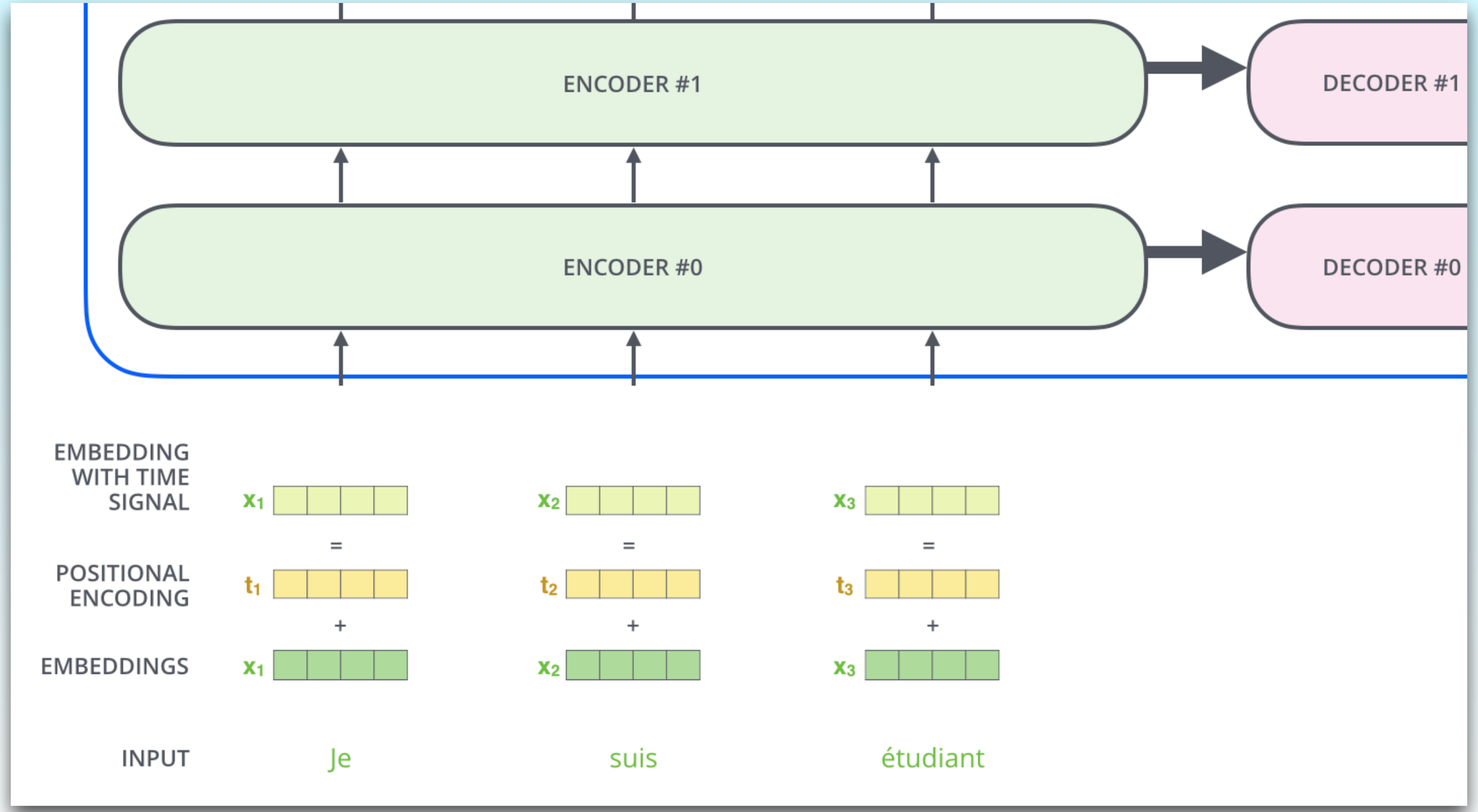


Figure 1-6. Encoder-decoder architecture of the original Transformer

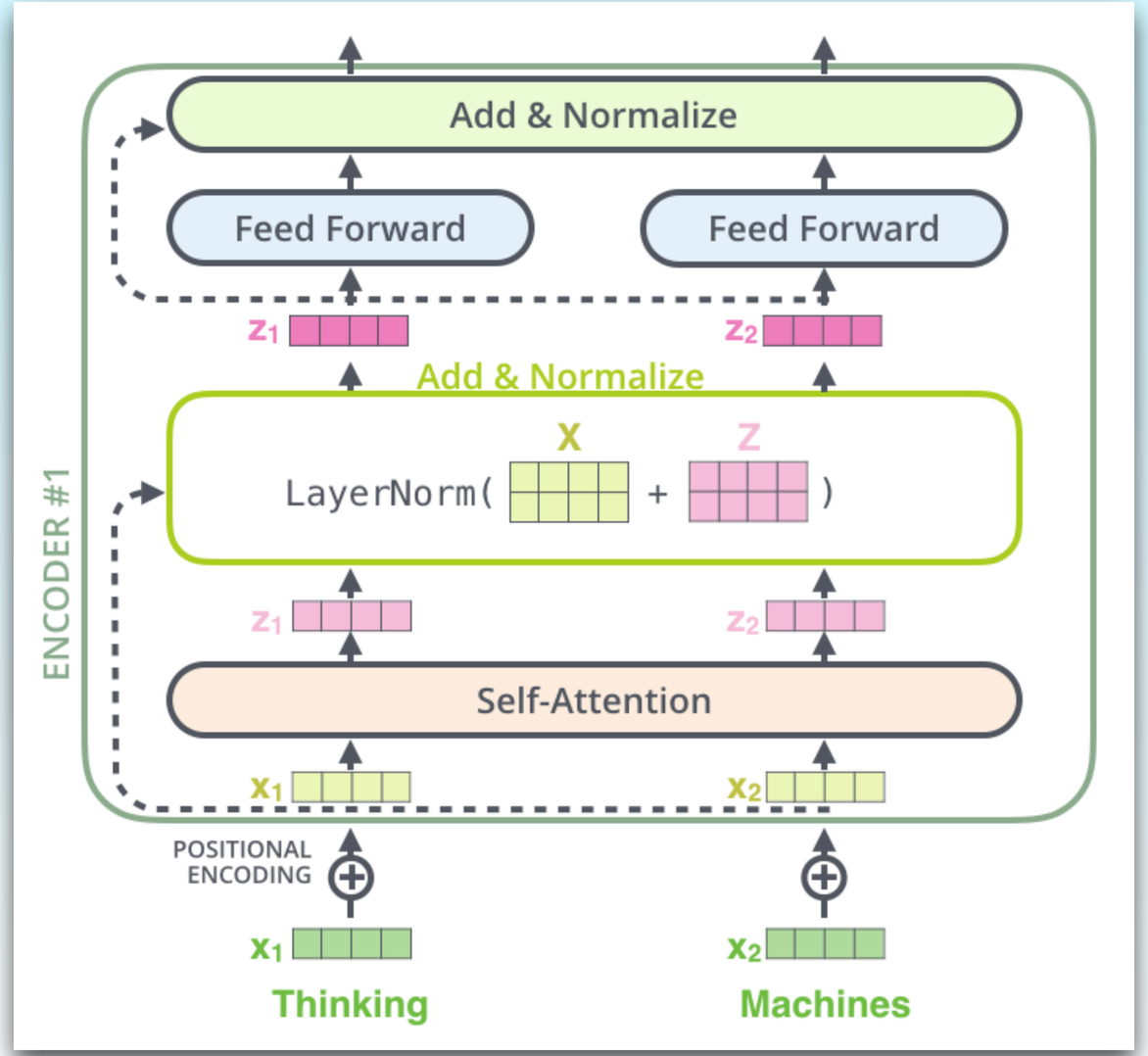
Images from Tunstall, Lewis, Leandro Von Werra, and Thomas Wolf. Natural language processing with transformers. " O'Reilly Media, Inc.", 2022.

Positional Encoding



Images from https://jalammar.github.io/illustrated-transformer/

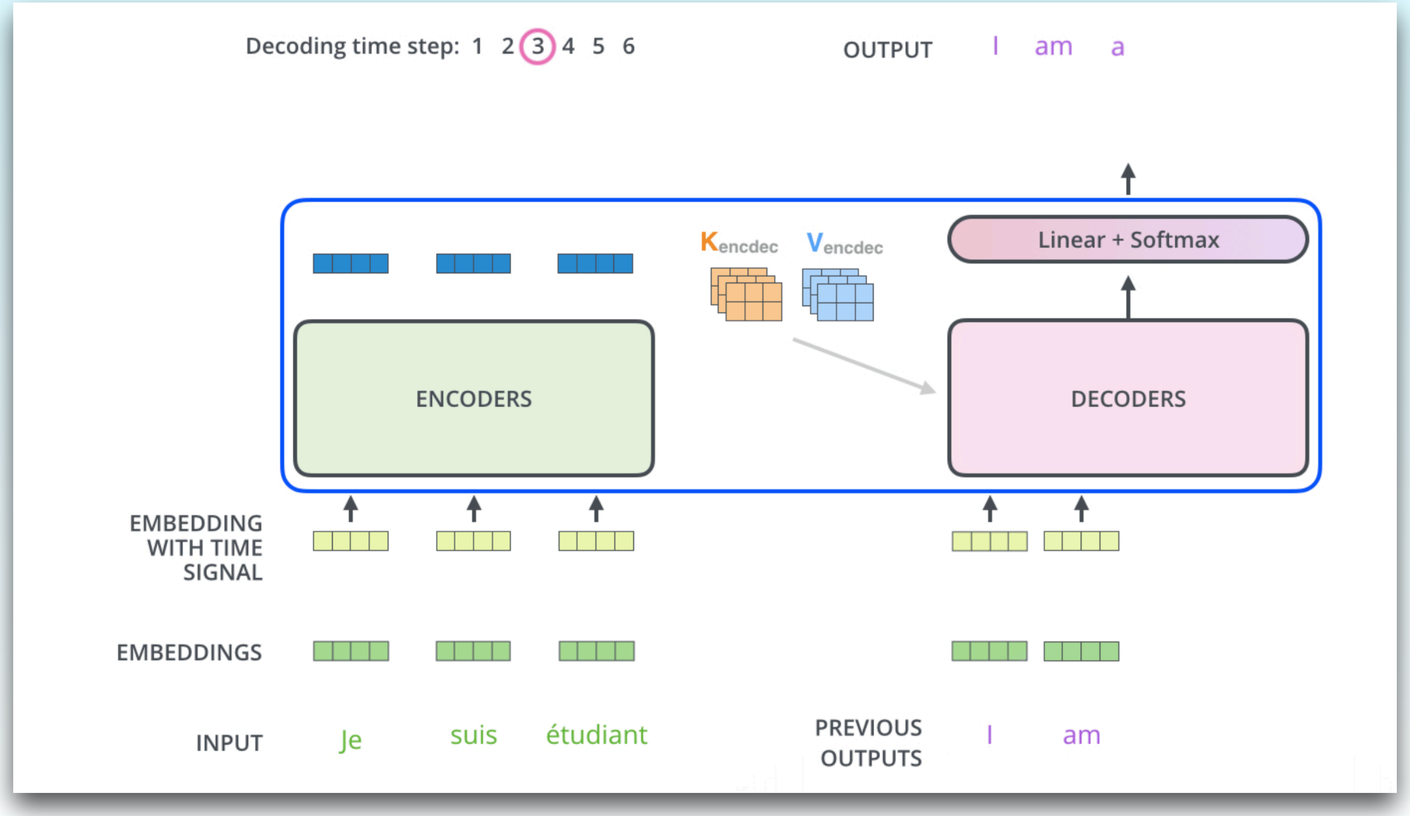
Full Encoder Together!



Images from https://jalammar.github.io/illustrated-transformer/

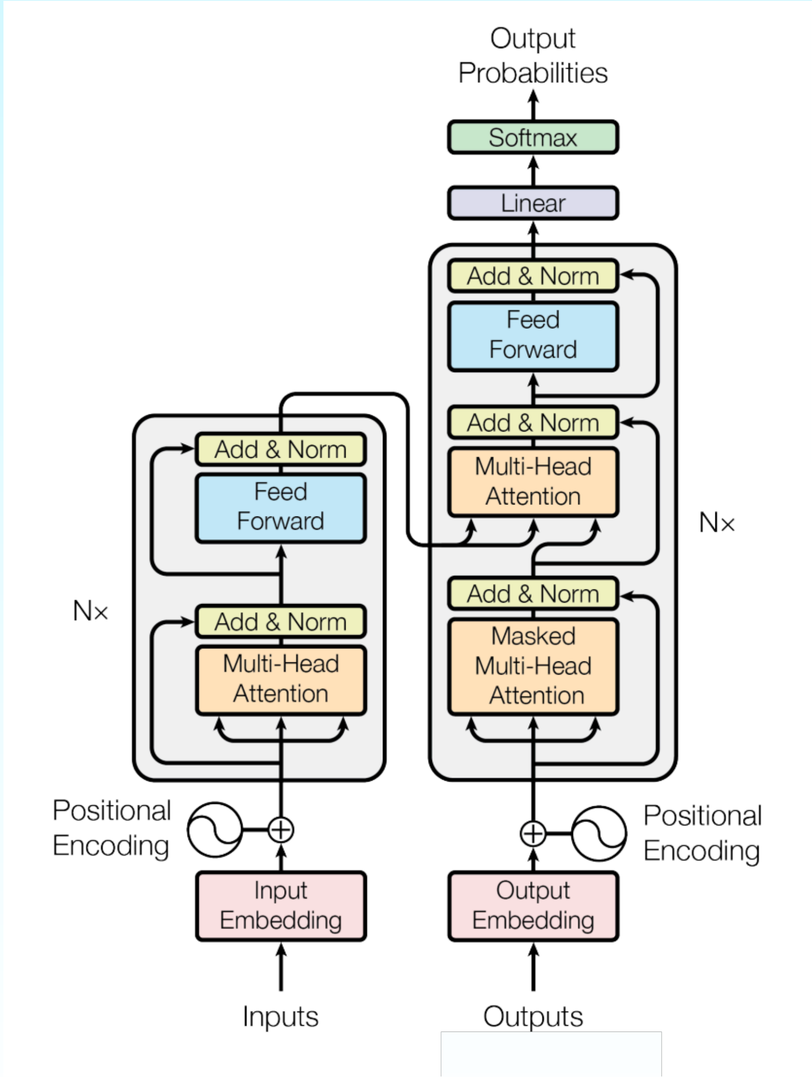
Transformer Decoder

The output of the last decoder turns into K and V — *masked self-attention + encoder-decoder attention (keys and values from encoder outputs)* -> each cell one word, then softmax



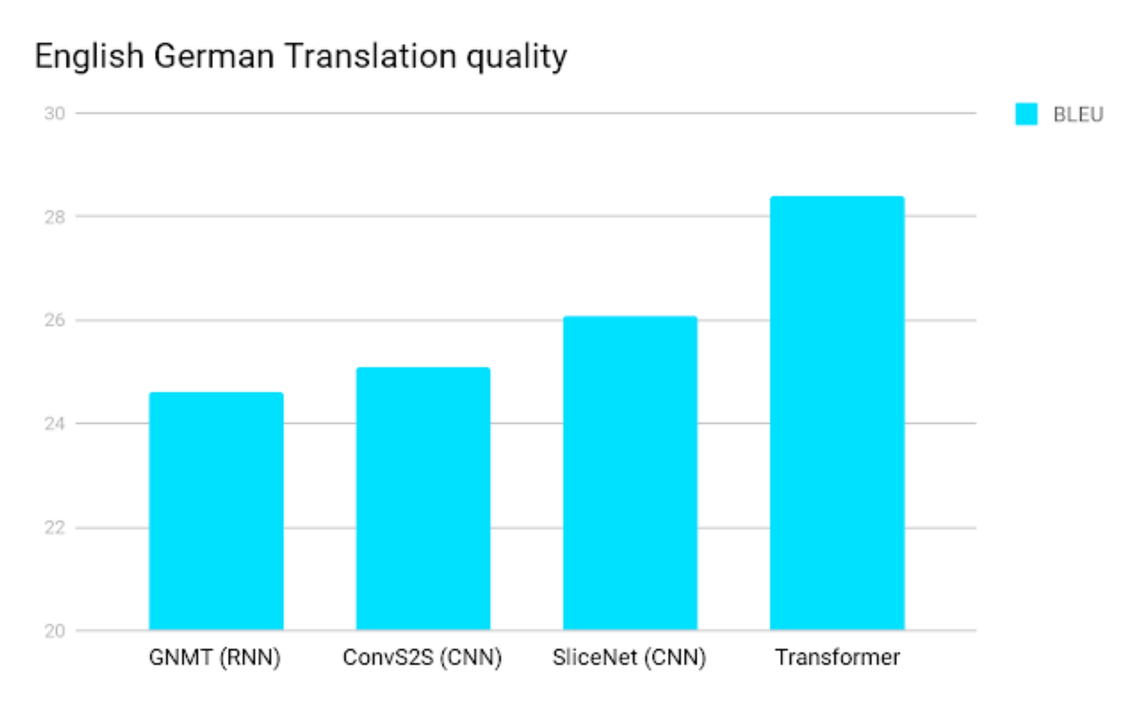
Images from https://jalammar.github.io/illustrated-transformer/

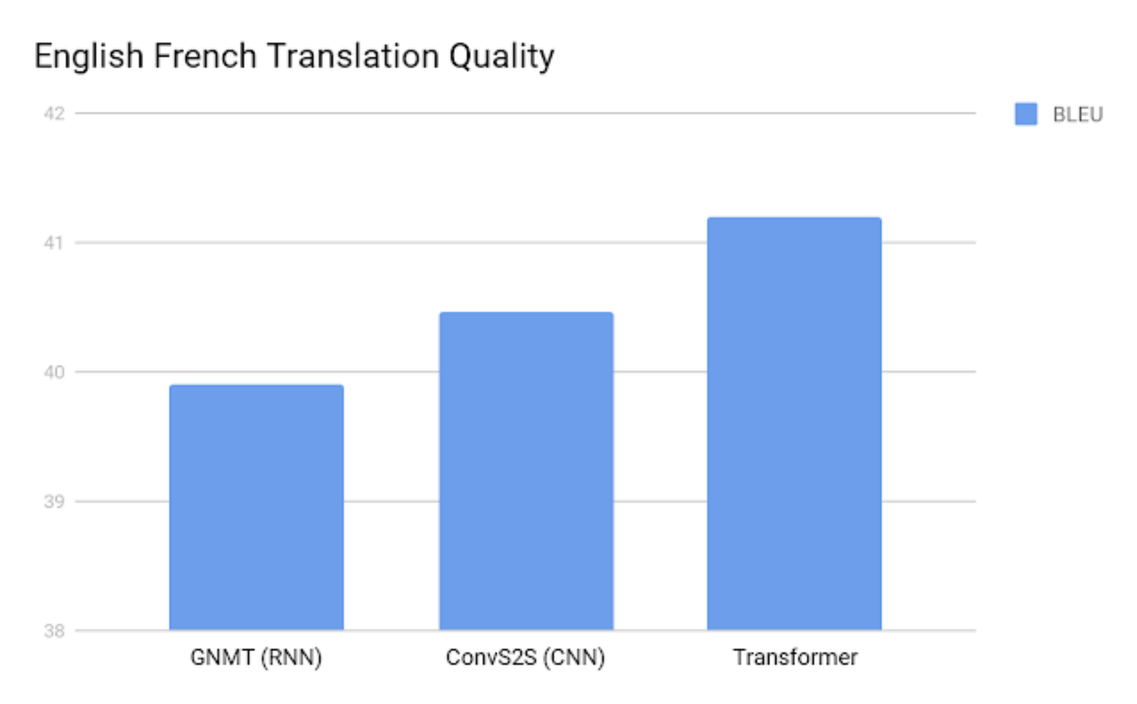
Transformer in one view!



From https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/ with changes

What did they achieve?





From https://heidloff.net/article/foundation-models-transformers-bert-and-gpt/ with changes

Let’s see a code together!