Encoder-Decoder Architectures and Neural Attention

Artificial Intelligence for Digital Health (AID)

M.Sc. in Digital Health – University of Pisa

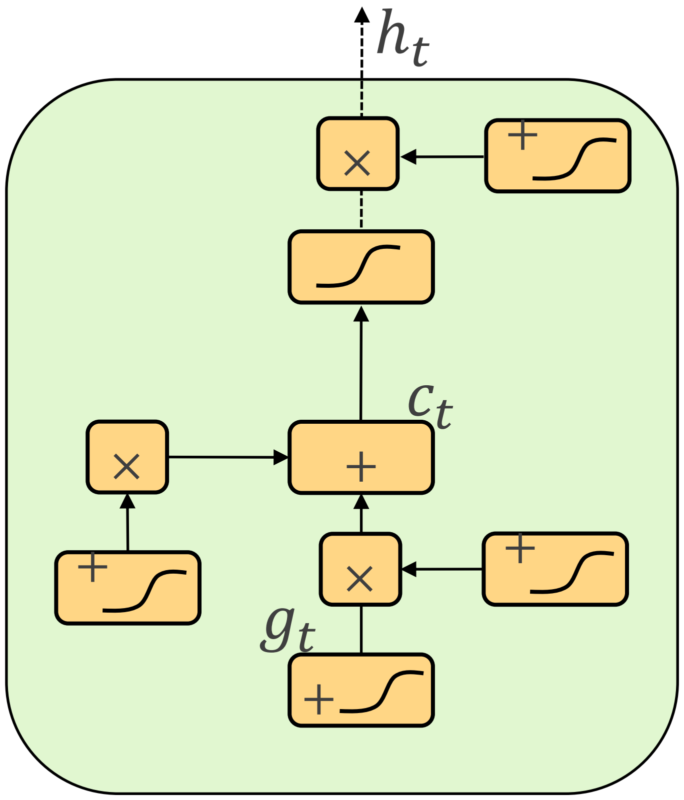
Davide Bacciu (davide.bacciu@unipi.it)

IN SUPREMÆ DIGNITATIS · 1343 ·

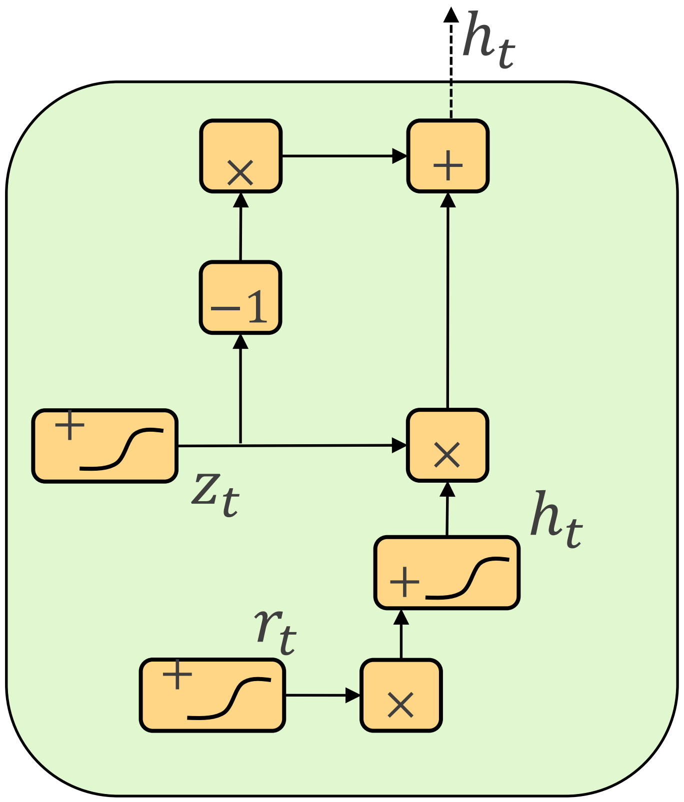
Lecture Outline

- Sequence-to-sequence learning
- Encoder-decoder architectures
- Neural attention
  - Cross-attention
  - Self-attention
- Transformers and vision transformers

Gated RNN Refresher

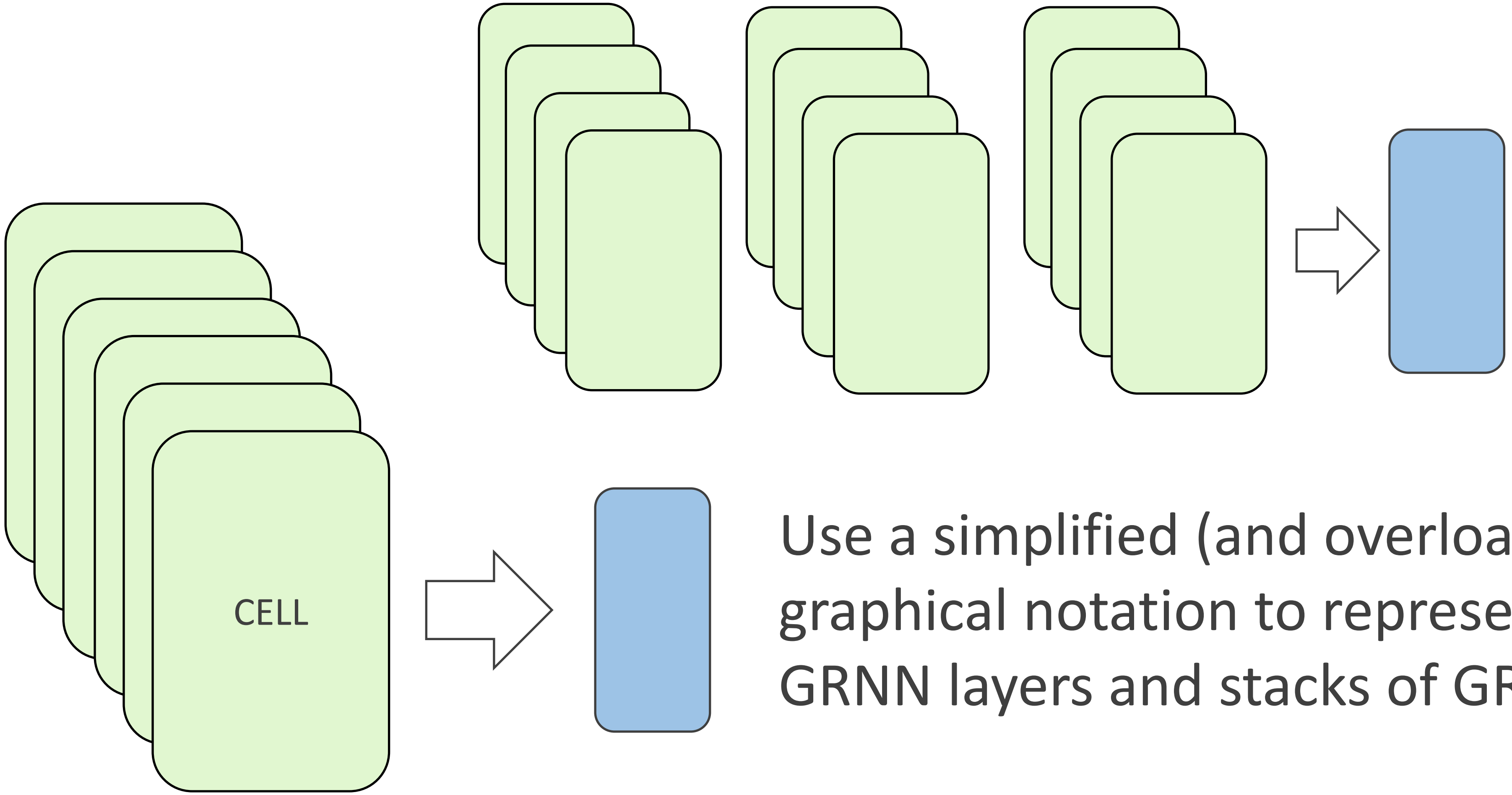


LSTM Cell



GRU Cell

Graphical Notation for Compositionality



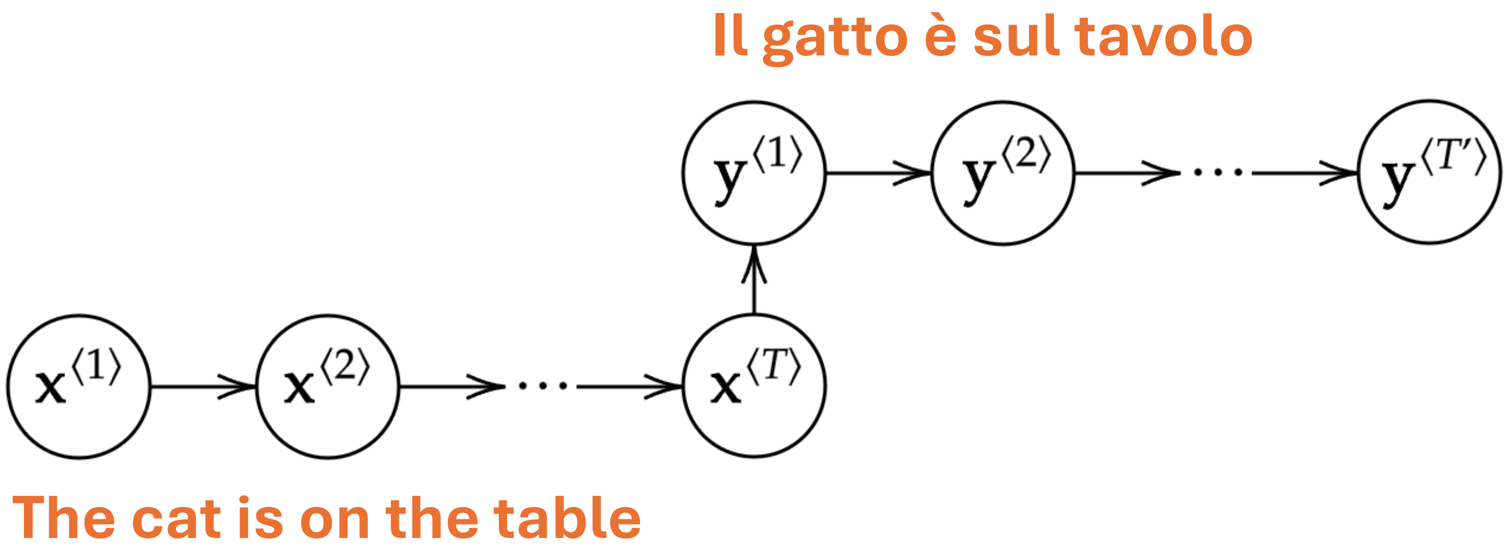
Use a simplified (and overloaded) graphical notation to represent GRNN layers and stacks of GRNN

Dealing with Compound Data

- GRNN are excellent to handle size/topology varying data in input
  - How can we handle size/topology varying outputs?
  - Sequence-to-sequence
- Structured data is compound information
  - Efficient processing needs the ability to focus on certain parts of such information
  - Attention mechanism

Sequence Transduction

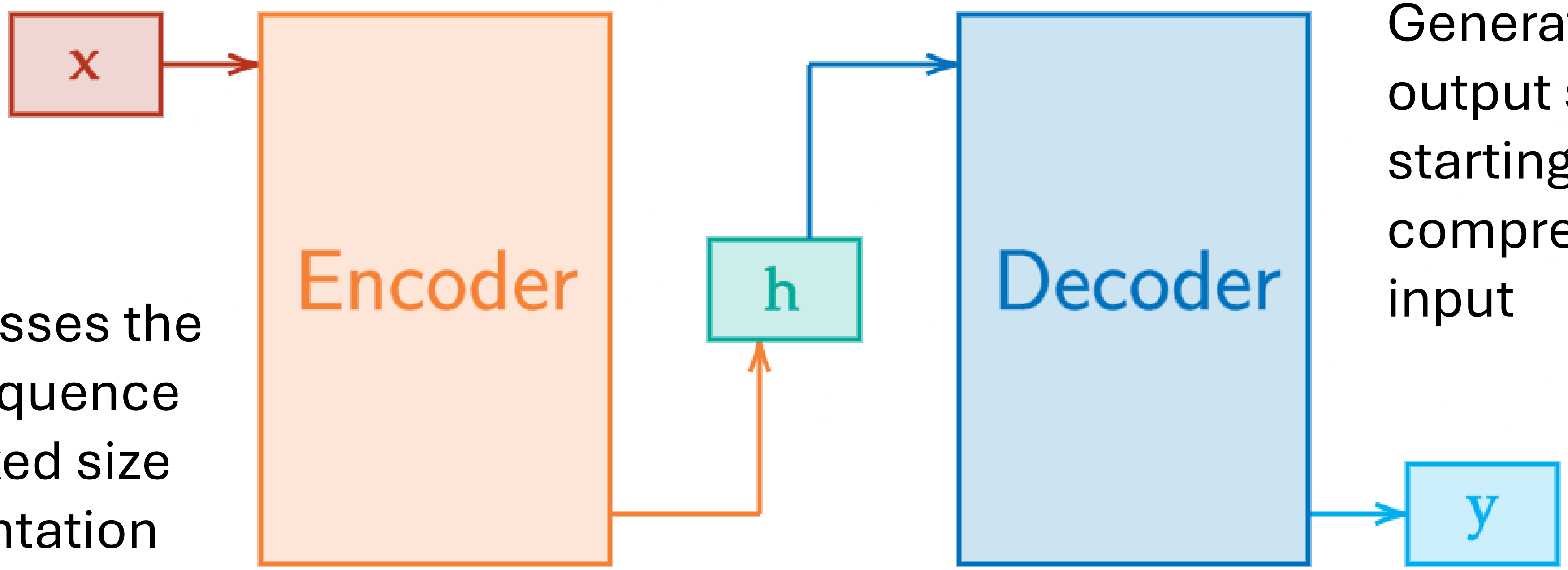
- A new class of learning problems over sequences
- Input and output are both sequences
  - They may have different lengths
  - They are not-aligned



Sequence-to-Sequence Learning

Solution is based on an encoder-decoder scheme

Compresses the input sequence into a fixed size representation

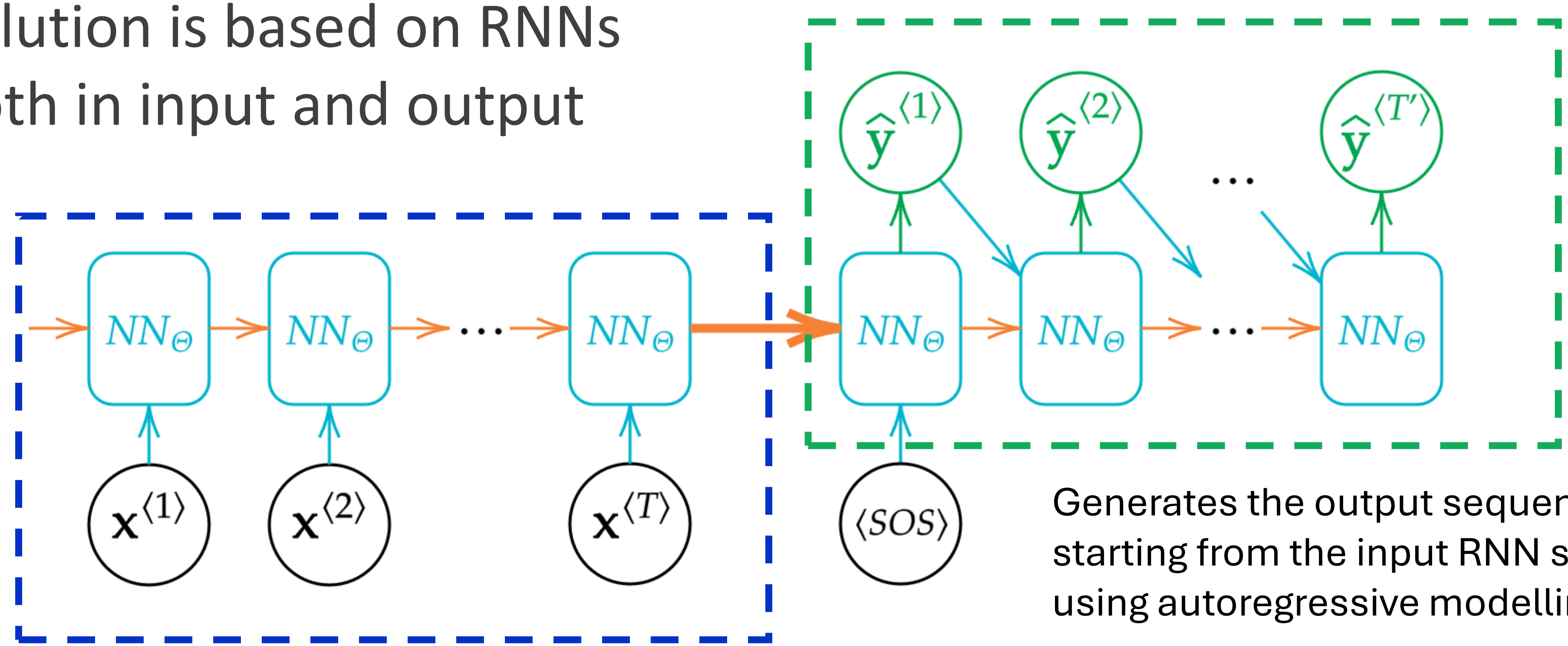


Generates the output sequence starting from the compressed input

Early Encoder-Decoder Architectures

Recurrent Sequence-to-Sequence Learning

Solution is based on RNNs both in input and output

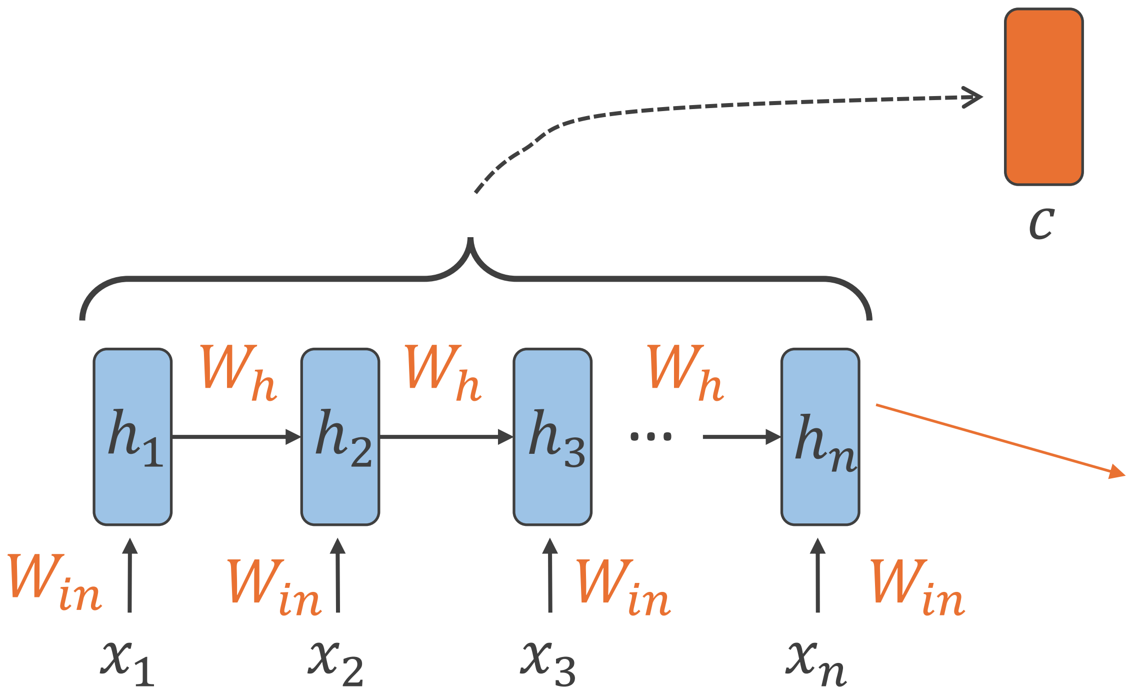


Generates the output sequence starting from the input RNN state using autoregressive modelling

Compresses the input sequence into a the RNN state

Encoder

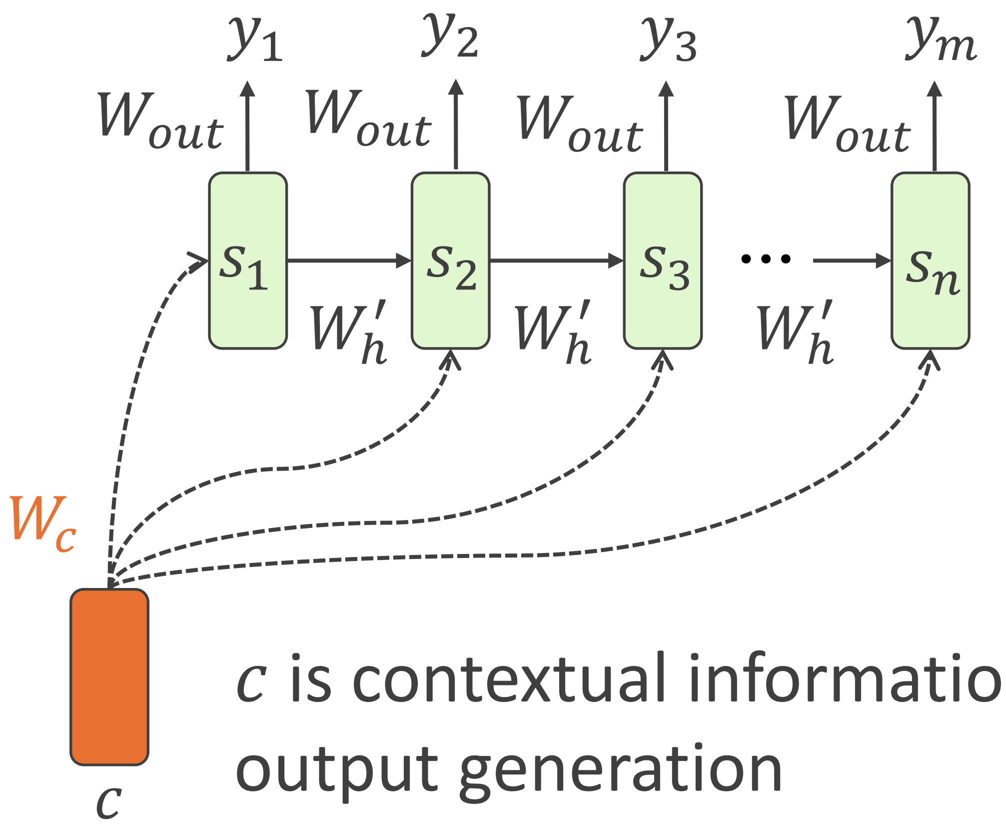
Produce a compressed and fixed length representation c of all the input sequence $x_1, \dots, x_n$



Originally $c = h_n$

Activations of an LSTM/GRU layer of K cells

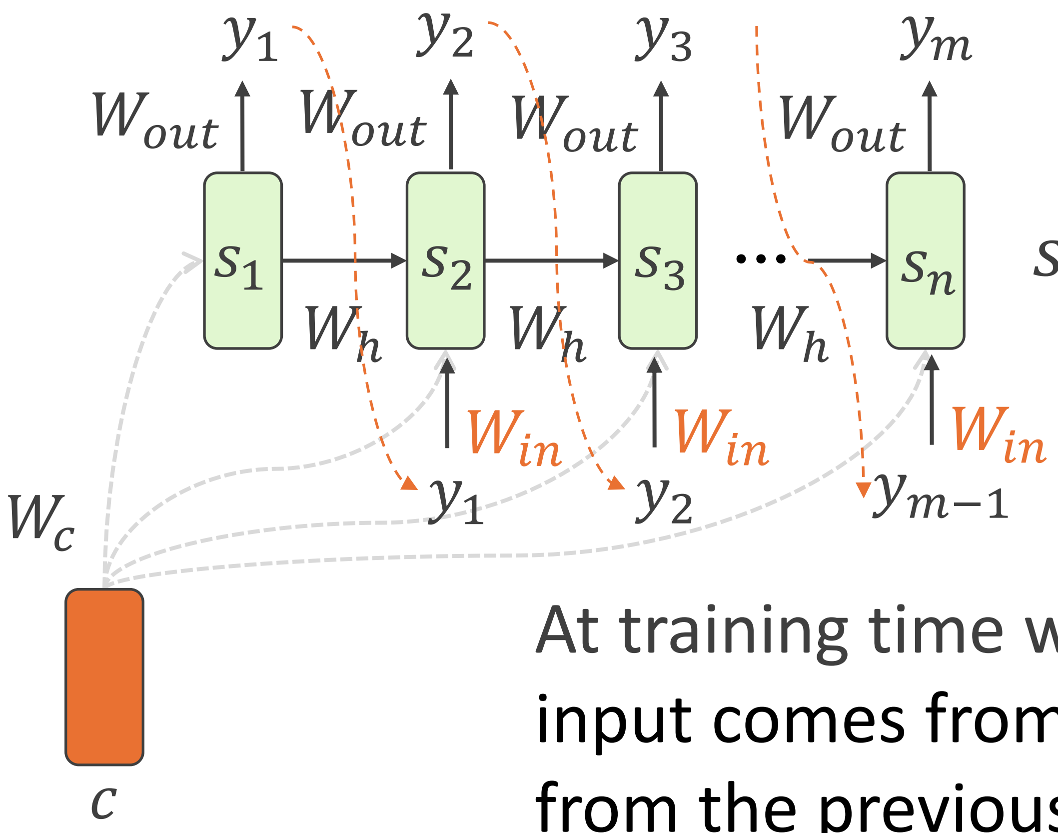
Decoder



A LSTM/GRU layer of K cells seeded by the context vector c

c is contextual information kept throughout output generation

Autoregressive Decoding



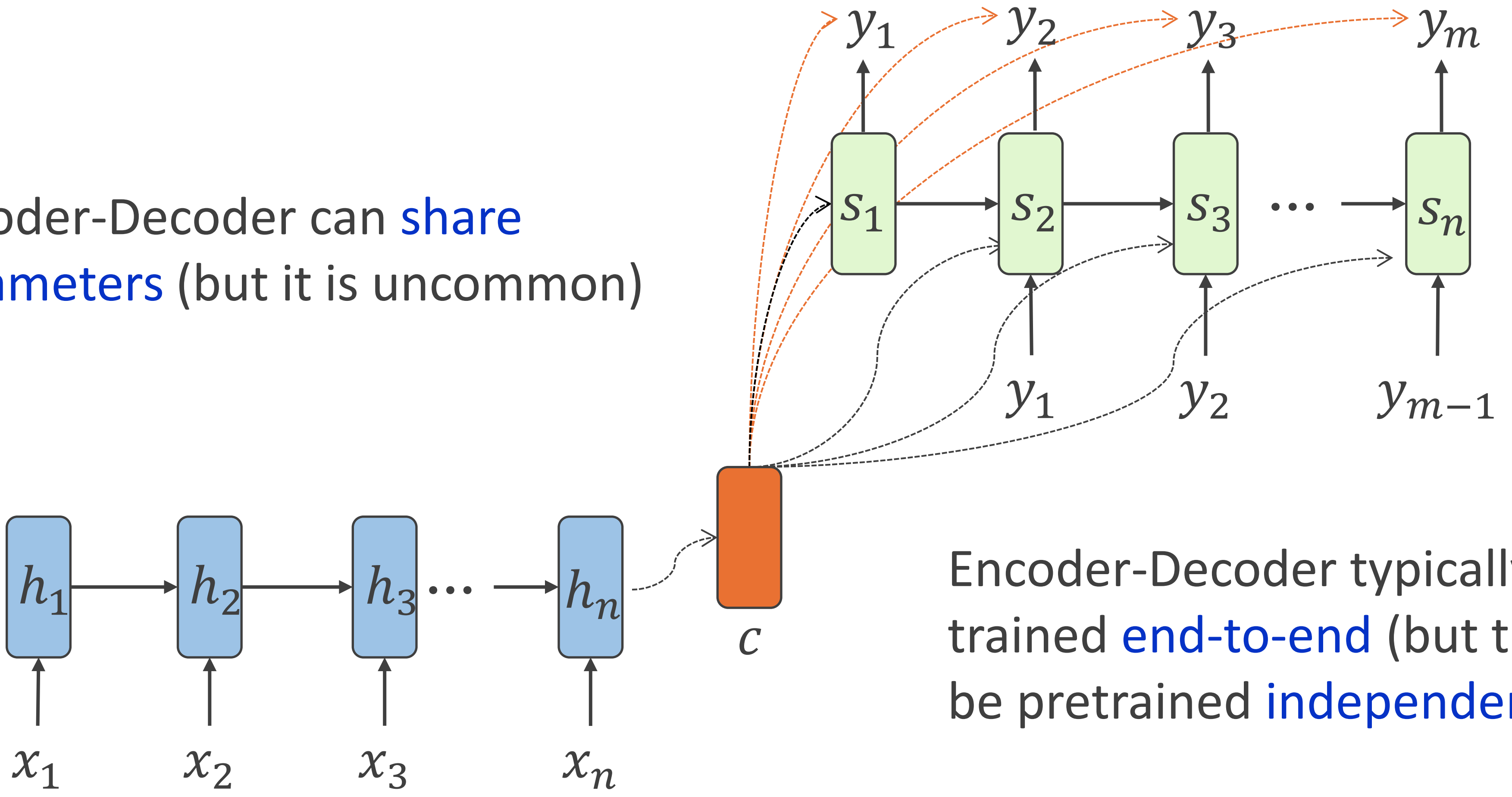
The output at the previous step is used as input to the current step (autoregressive generation)

$$s_i = f(c, s_{i-1}, y_{i-1})$$

At training time we use teacher forcing: current input comes from the ground truth rather than from the previous output (which can be wrong)

Sequence-To-Sequence Learning

Encoder-Decoder can share parameters (but it is uncommon)



Encoder-Decoder typically trained end-to-end (but they can be pretrained independently)

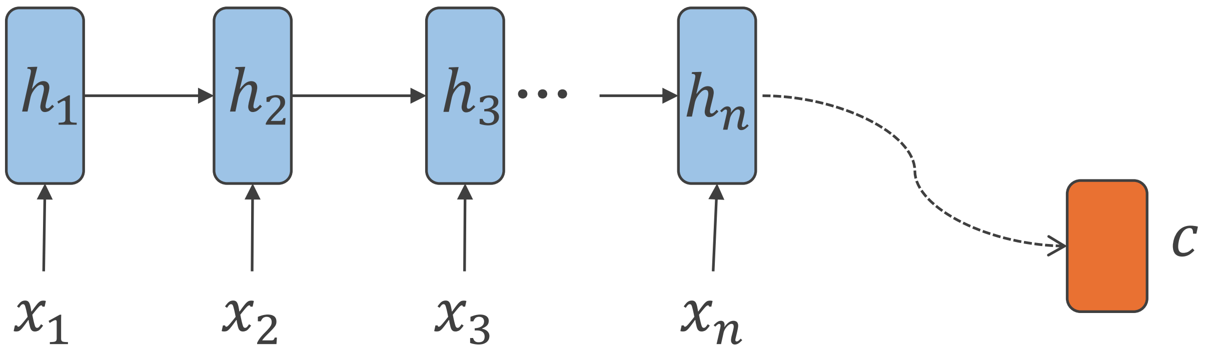
Attention

A Motivating Example

The cat is on the table

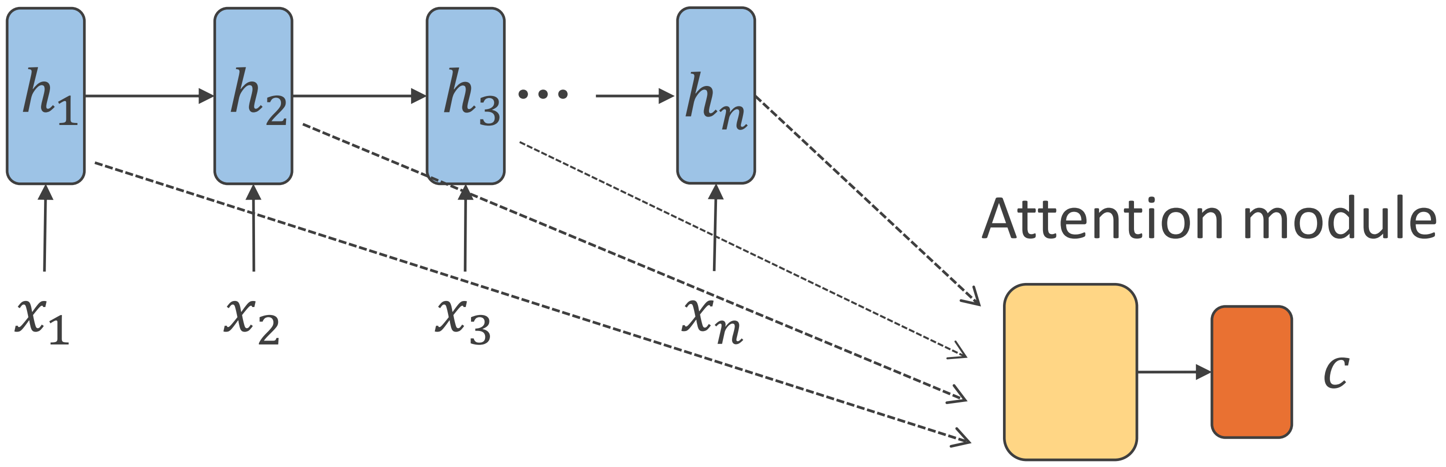
Il gatto è sul tavolo

On the Need of Paying Attention



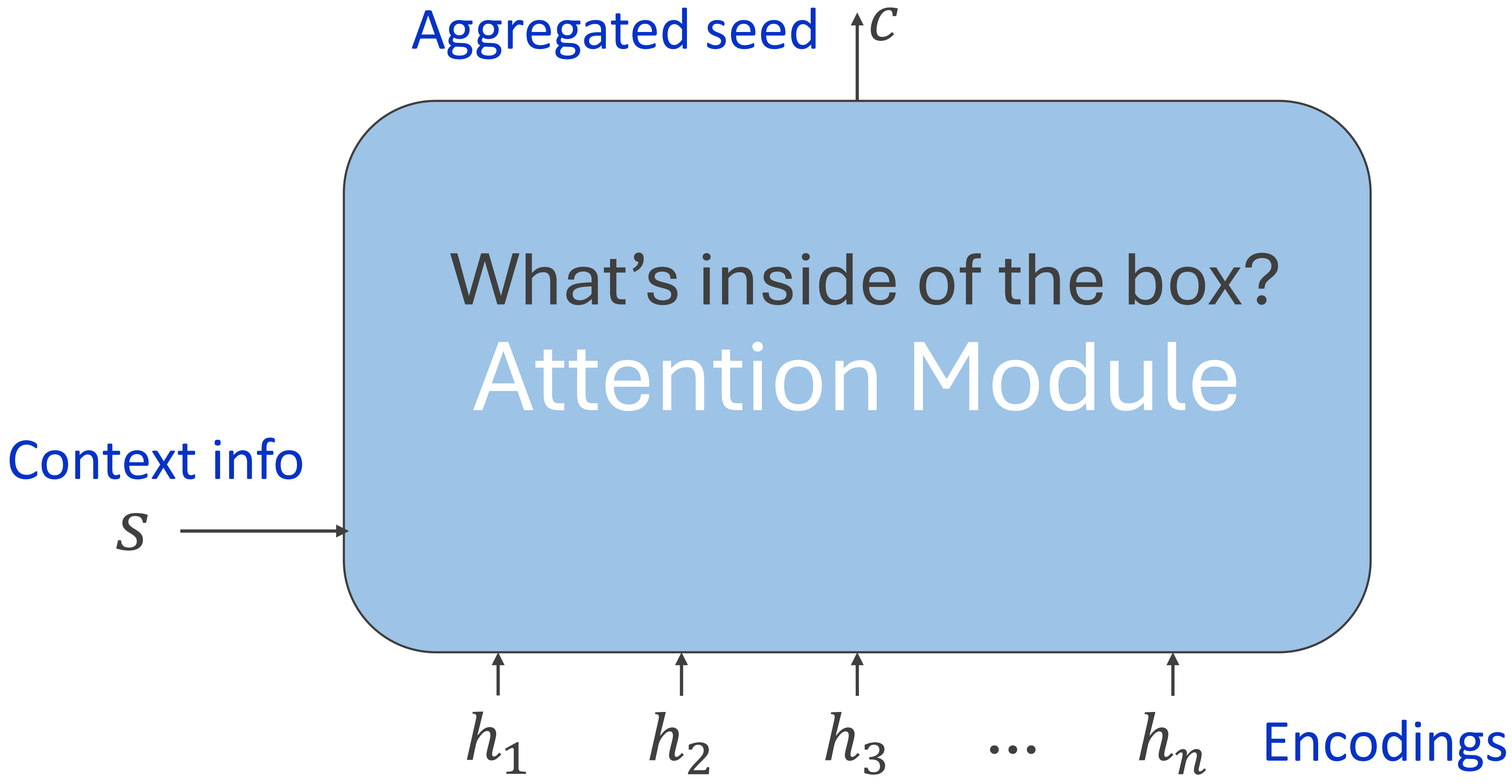
- Encoder-Decoder scheme assumes the hidden activation of the last input element summarizes sufficient information to generate the output
  - Bias toward most recent past
- Other parts of the input sequence might be very informative for the task
  - Possibly elements appearing very far from sequence end

On the Need of Paying Attention



- Attention mechanisms select which part of the sequence to focus on to obtain a good c

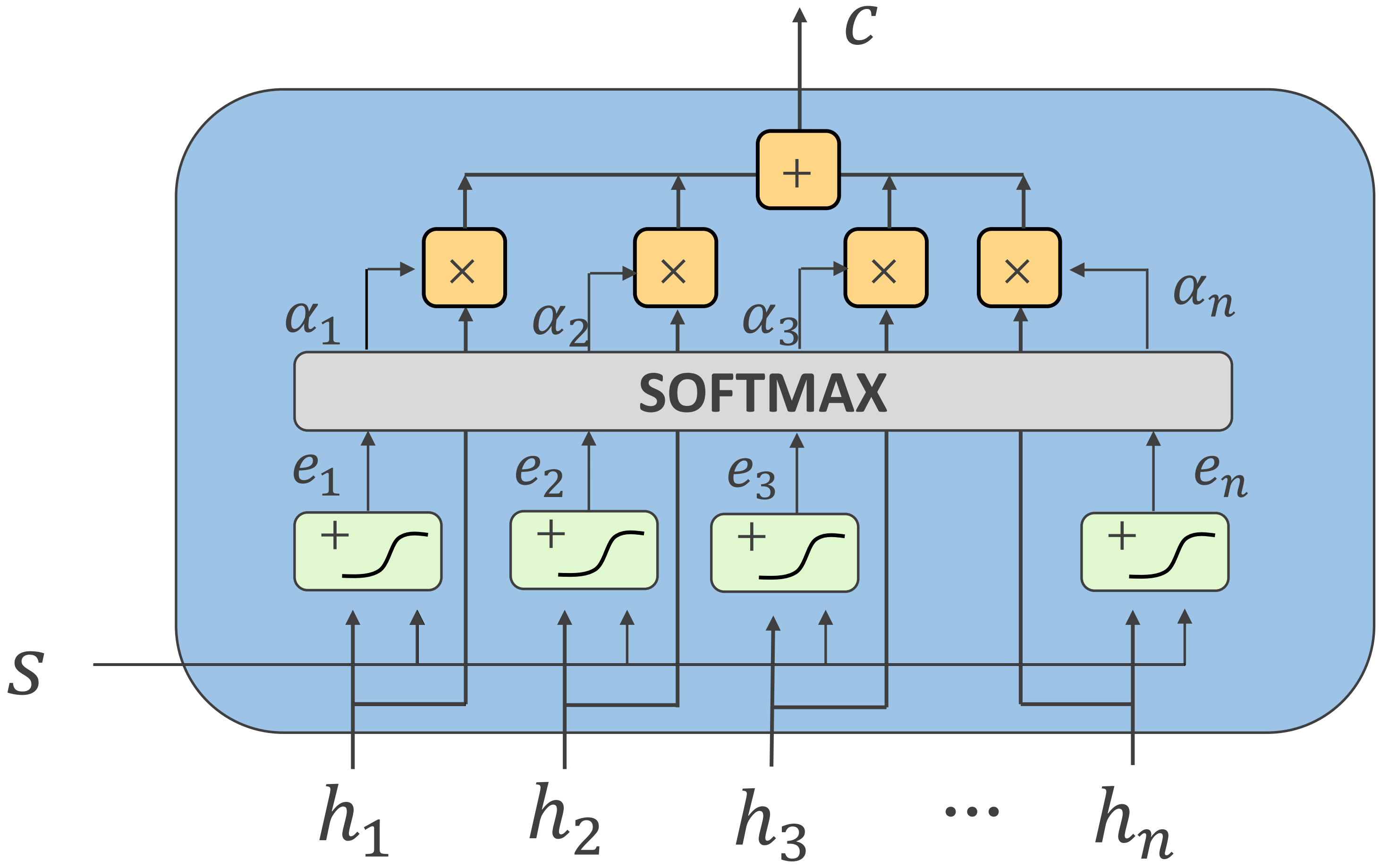
Attention Mechanisms – Blackbox View



What's inside of the box?

The Revenge of the Gates!

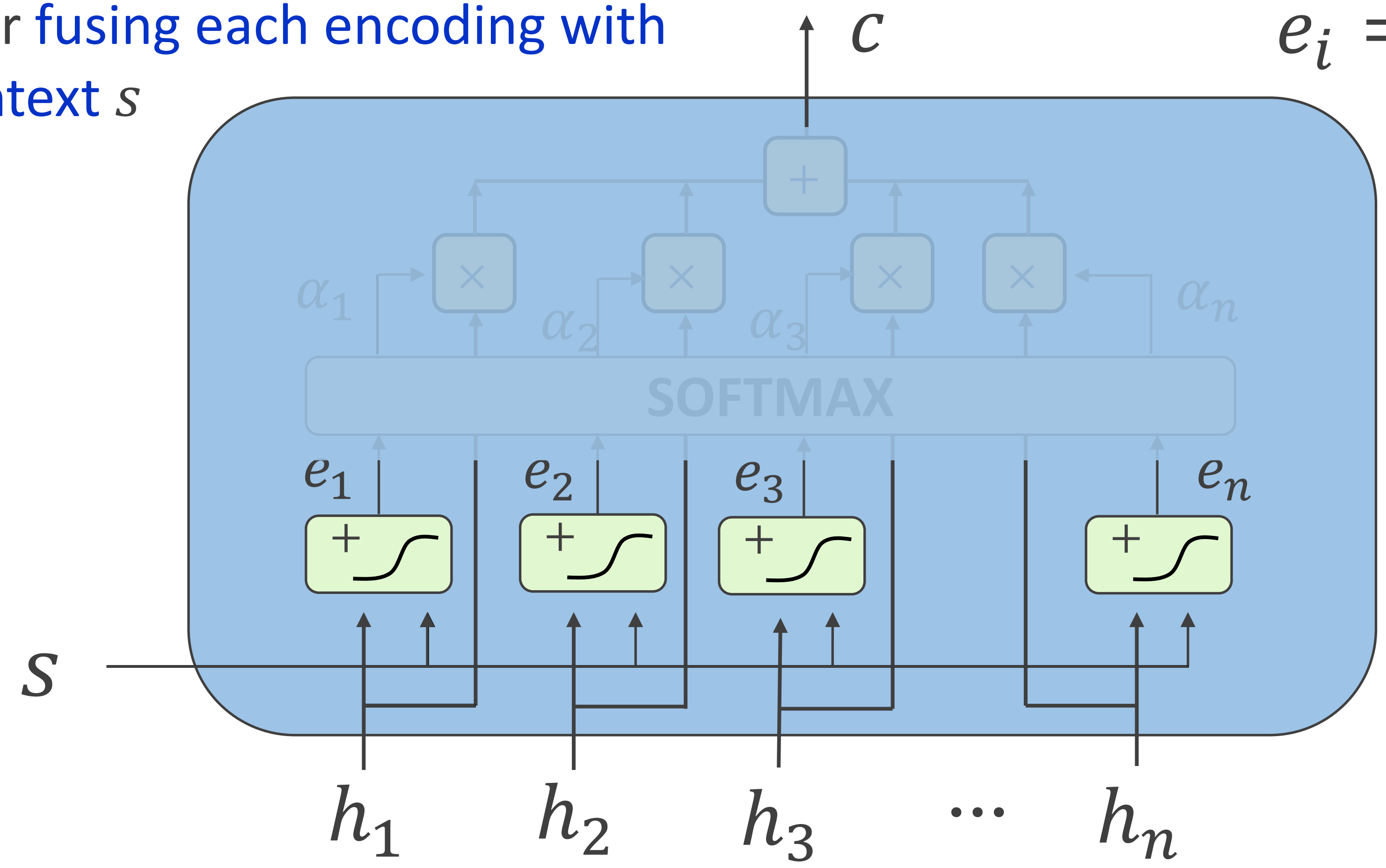
Opening the Box – Cross Attention



Opening the Box – Relevance

Neural layer fusing each encoding with current context s

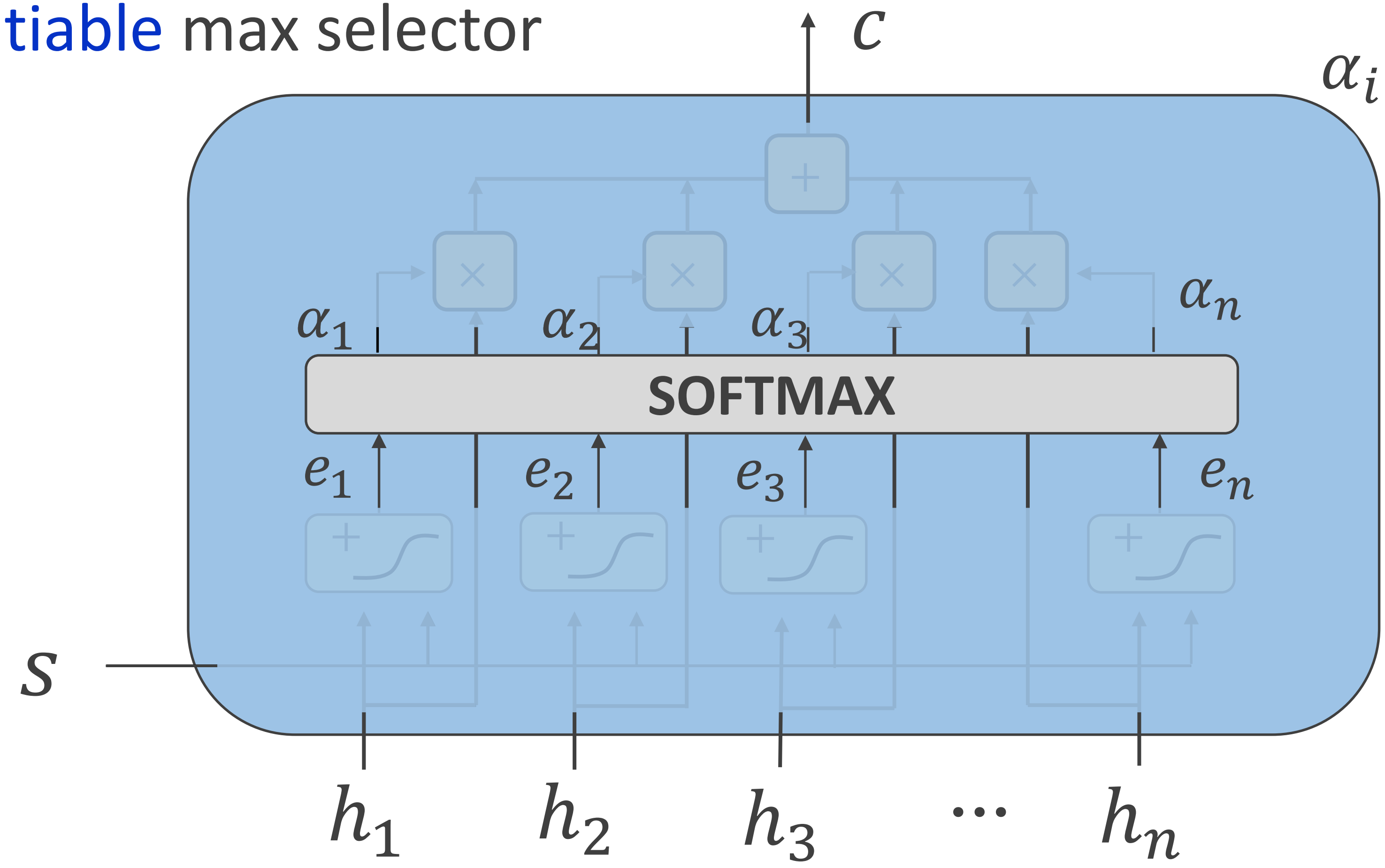
$$e_i = \mathrm{a}(s, h_i)$$



Opening the Box – Softmax

A differentiable max selector operator

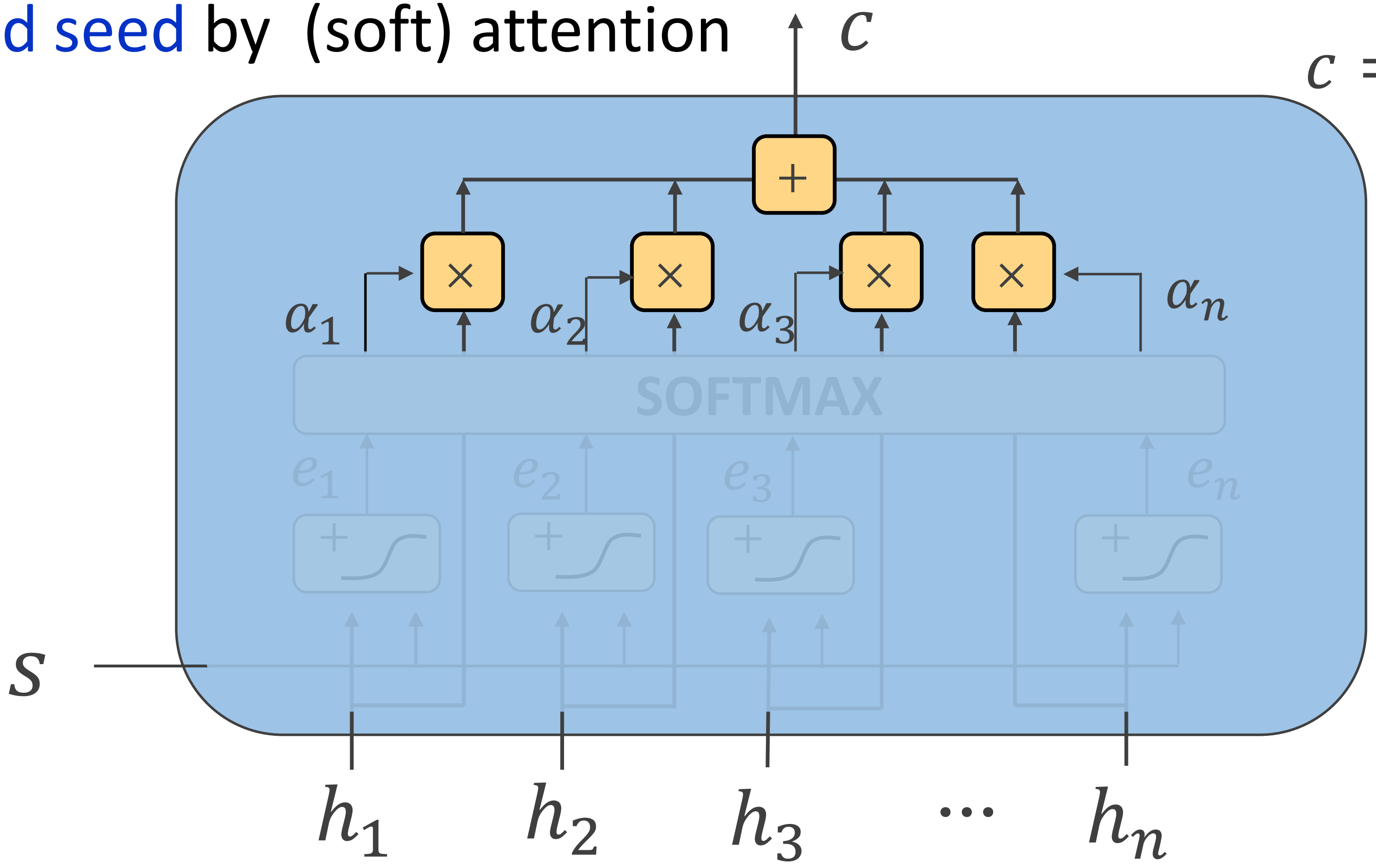
$$\alpha_i = \frac{\exp(e_i)}{\sum_j \exp(e_j)}$$



Opening the Box – Voting

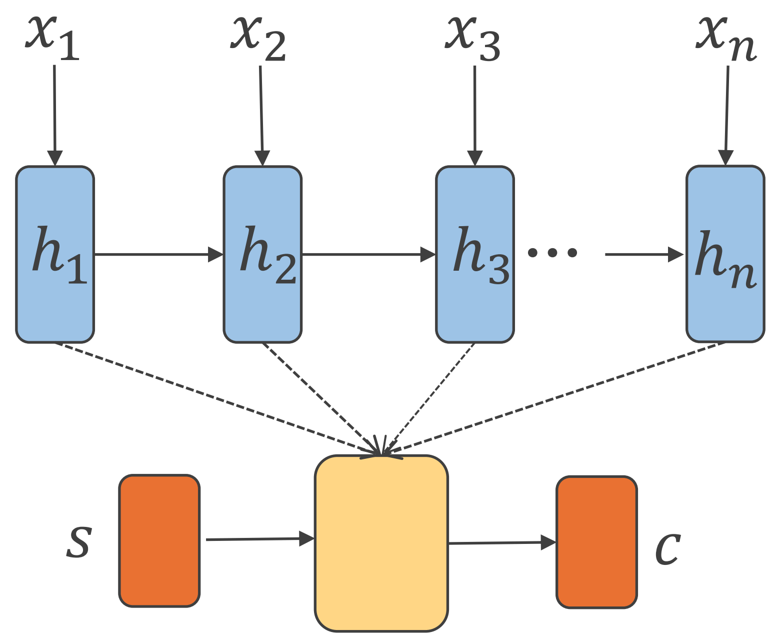
Aggregated seed by (soft) attention voting

$$c = \sum_i \alpha_i h_i$$



Attention - Equations

- Relevance: $e_i = \mathrm{a}(s, h_i)$
- Normalization: $\alpha_i = \dfrac{\exp(e_i)}{\sum_j \exp(e_j)}$
- Aggregation: $c = \sum_i \alpha_i h_i$

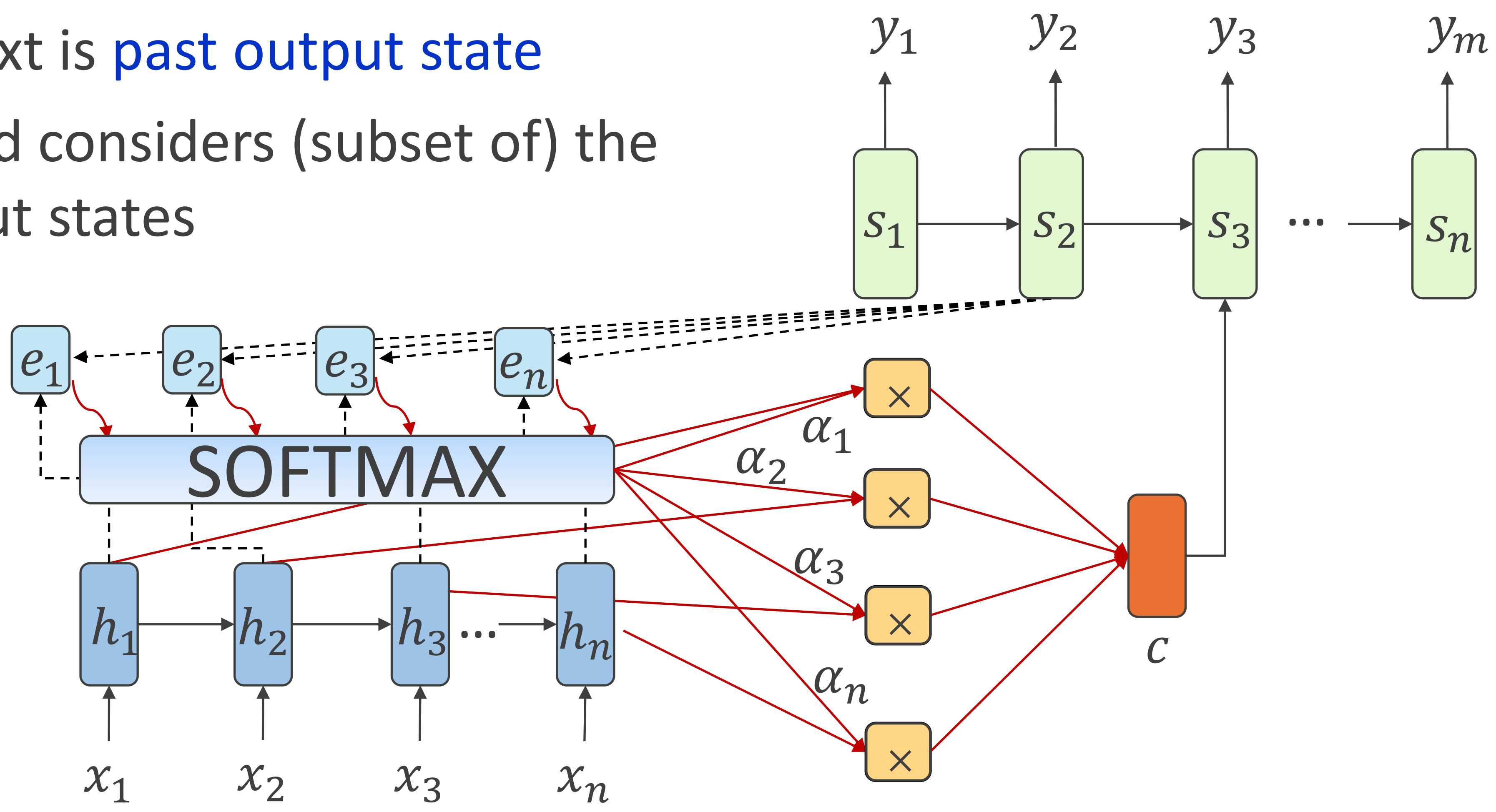


Attention module

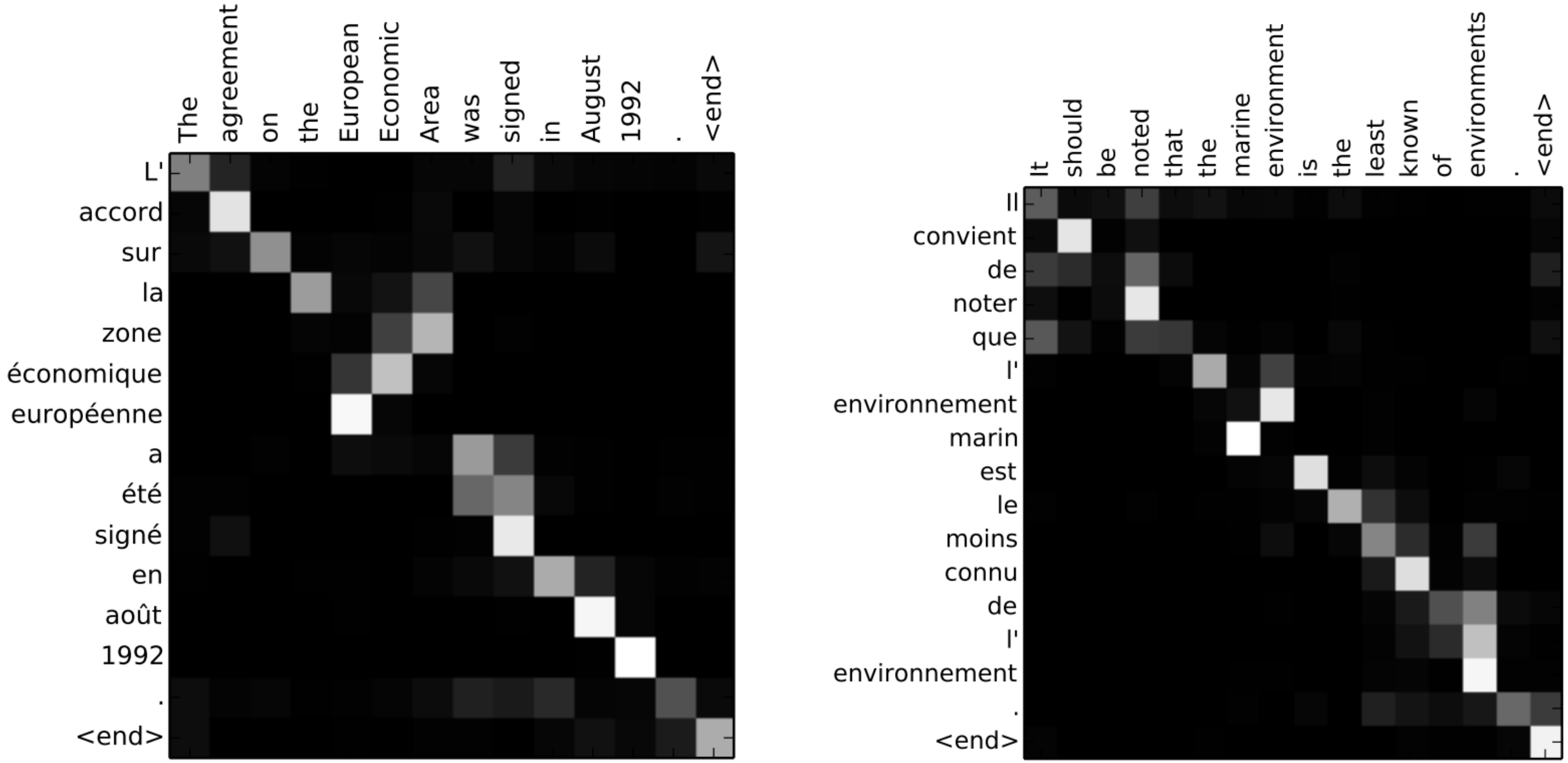
Attention in Seq2Seq

Context is past output state

- Seed considers (subset of) the input states



Learning to Translate with Attention



Bahdanau et al, Show, Neural machine translation by jointly learning to align and translate, ICLR 2015

Transformers

Limitations of the Recurrent Approach

- Sequence-to-sequence RNNs opened the way “difficult” tasks such as machine translation and question answering
- They also popularized the encoder-decoder architecture which as been used in multiple concretizations: e.g., image-to-sequence, sequence-to-image, image-to-imagesequence, ...
- Still RNNs retain the issues seen previously
  - Difficulty in learning long-range dependencies
  - Gradient issues

We need a different approach, that does not use recurrence to capture relationships between the elements of the sequence

Self Attention – The Intuition

- **Previously** - Hope that the input at position i remains in memory up to position j in order to learn the relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$
- **Now** – Compute explicitly the relationship between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ for all choices of i and j
- To achieve this, we will need to transform each element $\boldsymbol{x}_i$ of the input sequence into three vectors
  - **K**ey
  - **Q**uery
  - **V**alue

Self Attention – K,V,Q Generation

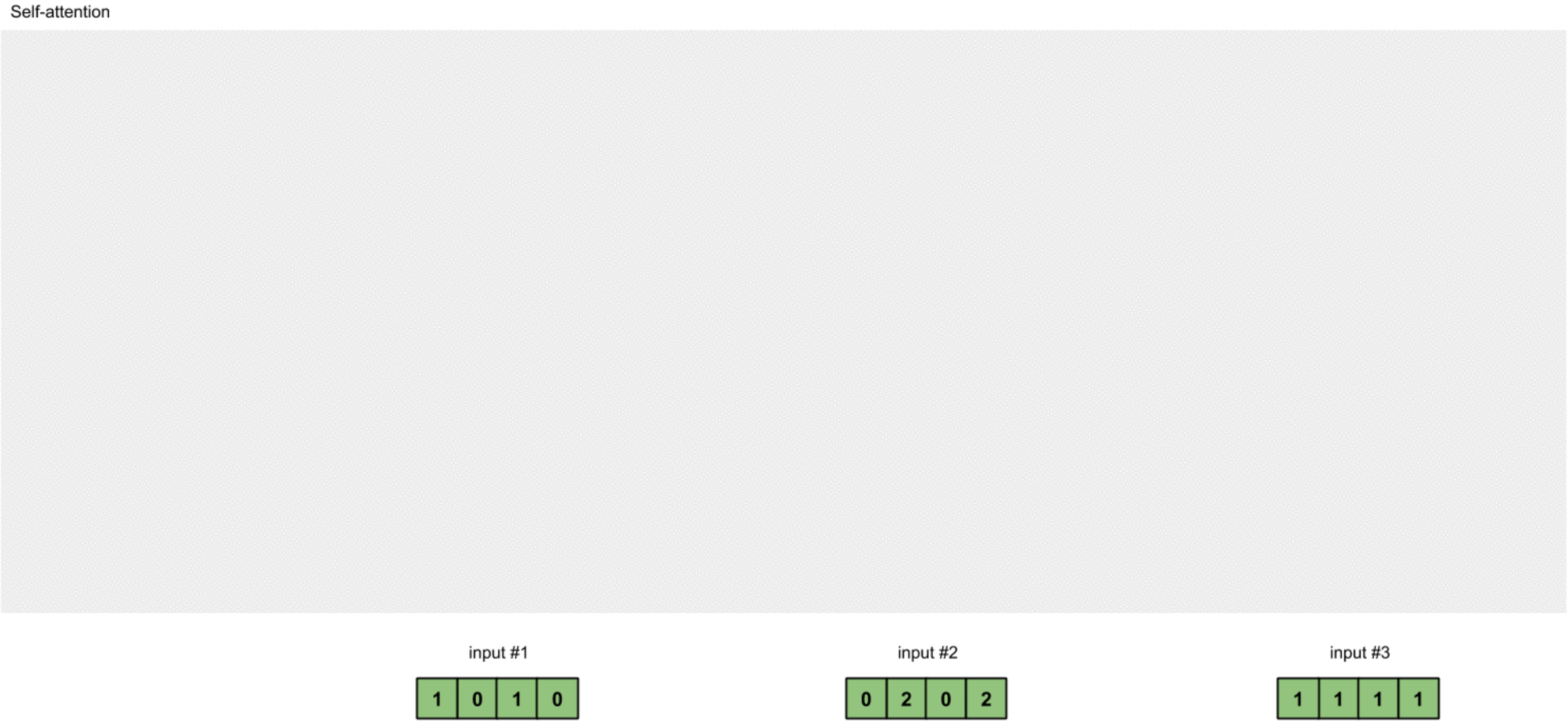
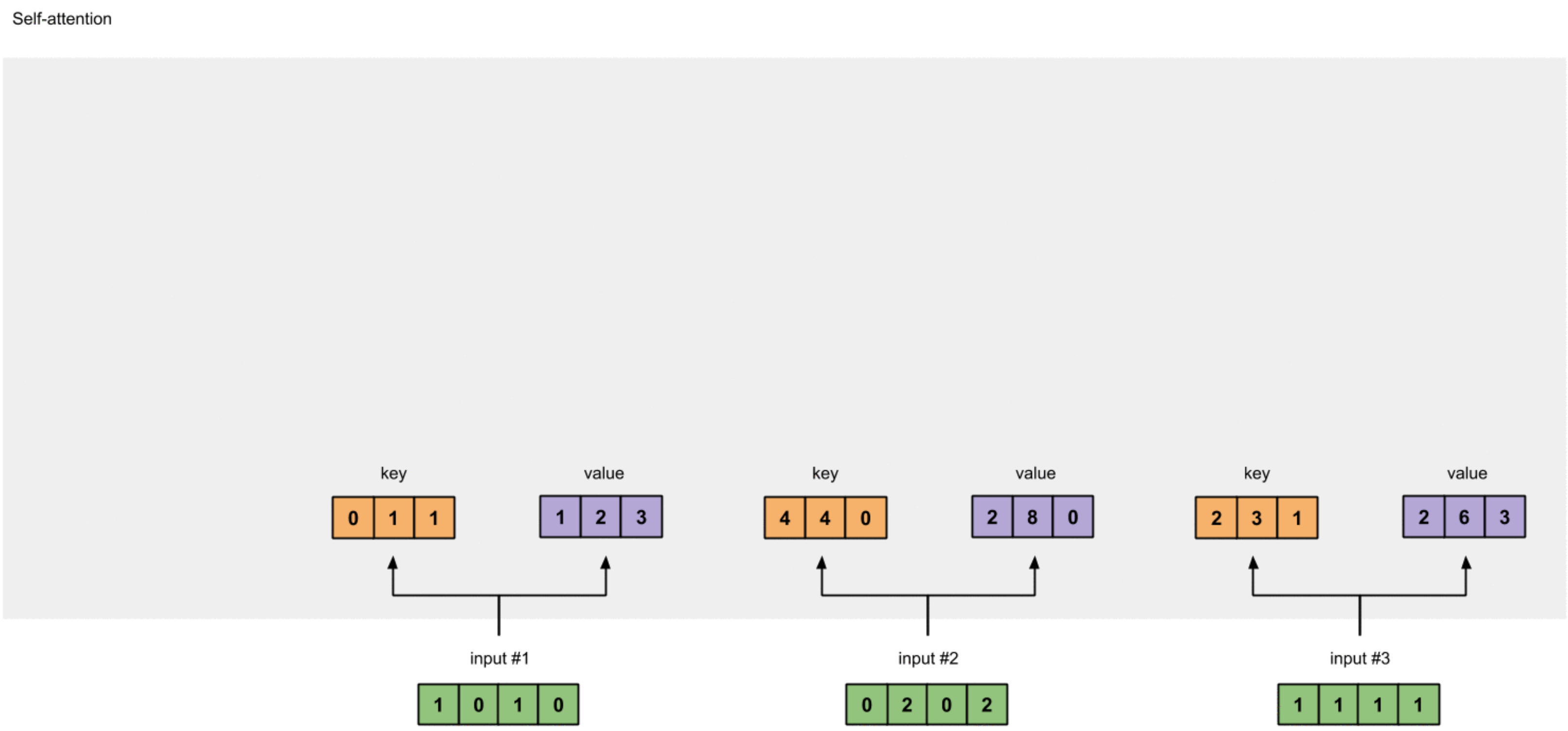


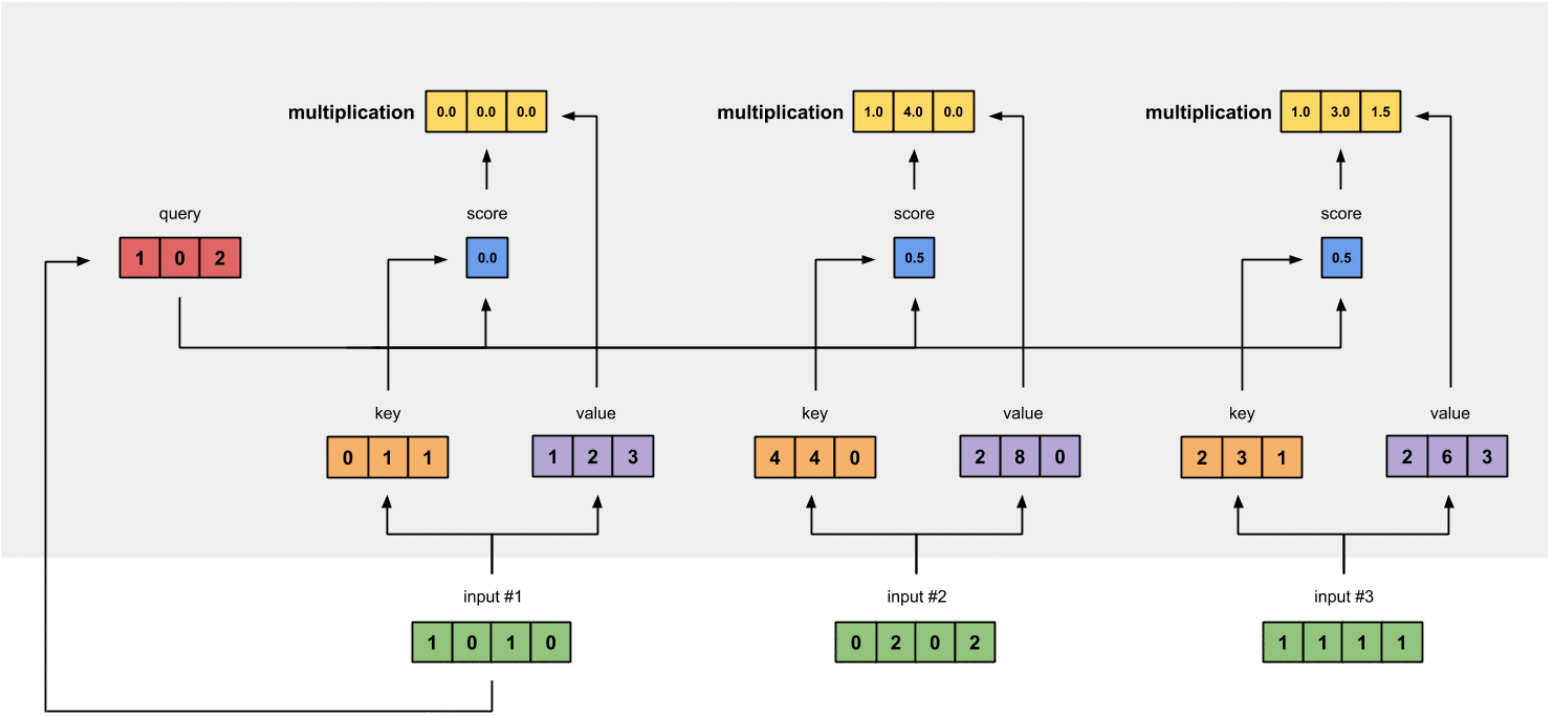
Figure credit to this [article]

Self Attention – Compute Attention Score



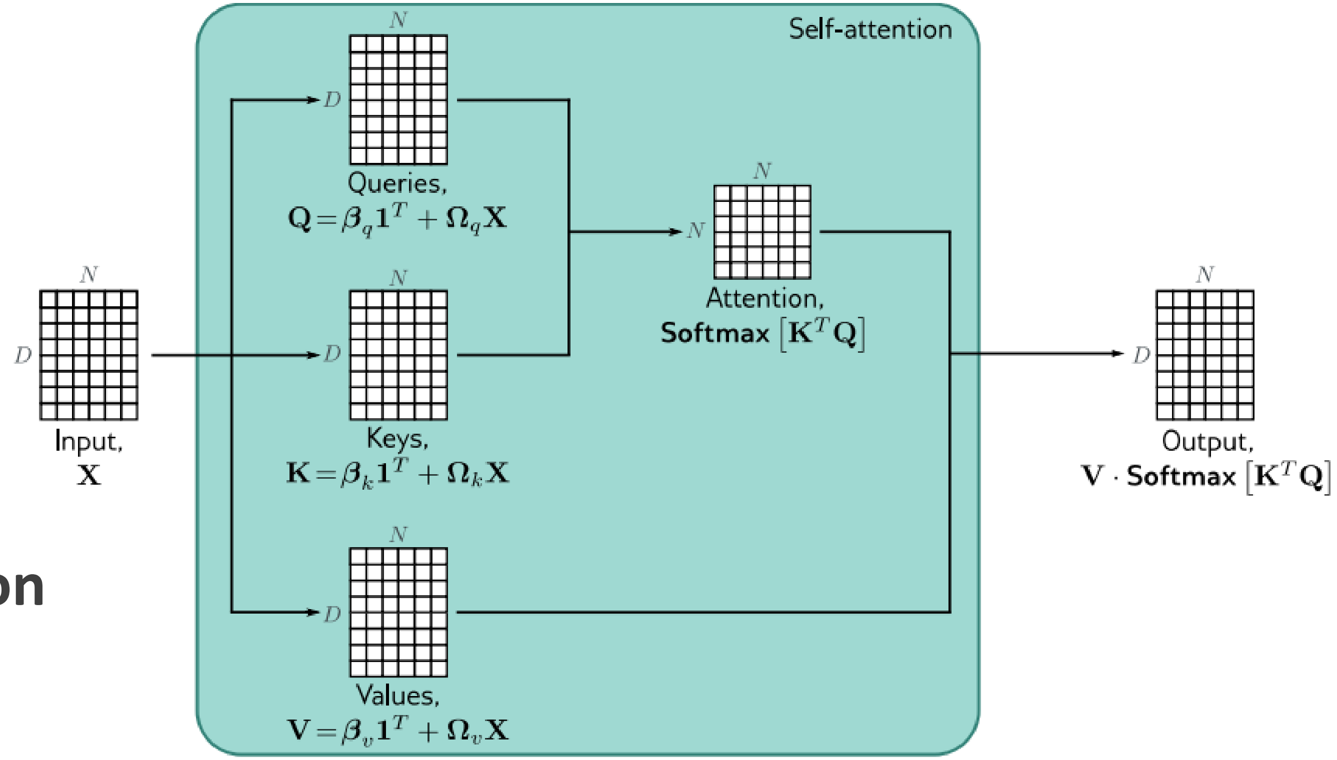
Self Attention – Produce Output

Self-attention



Self Attention

Each element of an input sequence x_i projects into 3 vectors: **query**, **key** and **value**



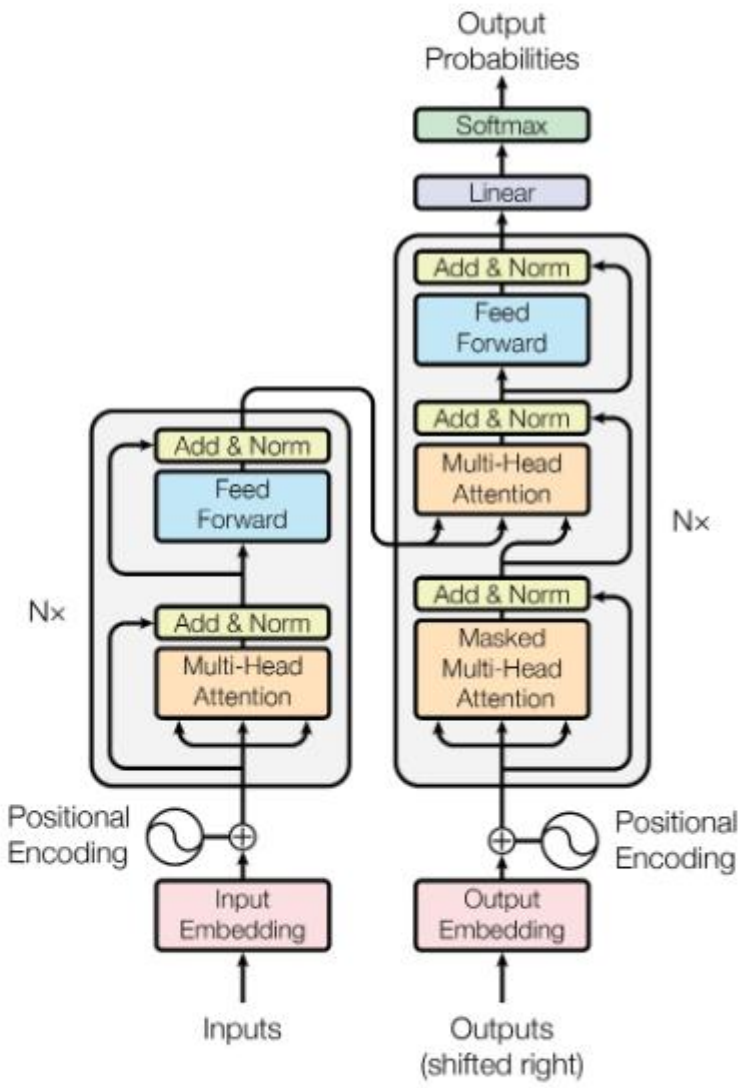
Scaled (multiplicative) self-attention

$$\sum_j softmax_j \left(\frac{Q_i \cdot \boldsymbol{K}^T}{\sqrt{d_k}} \right) V_j$$

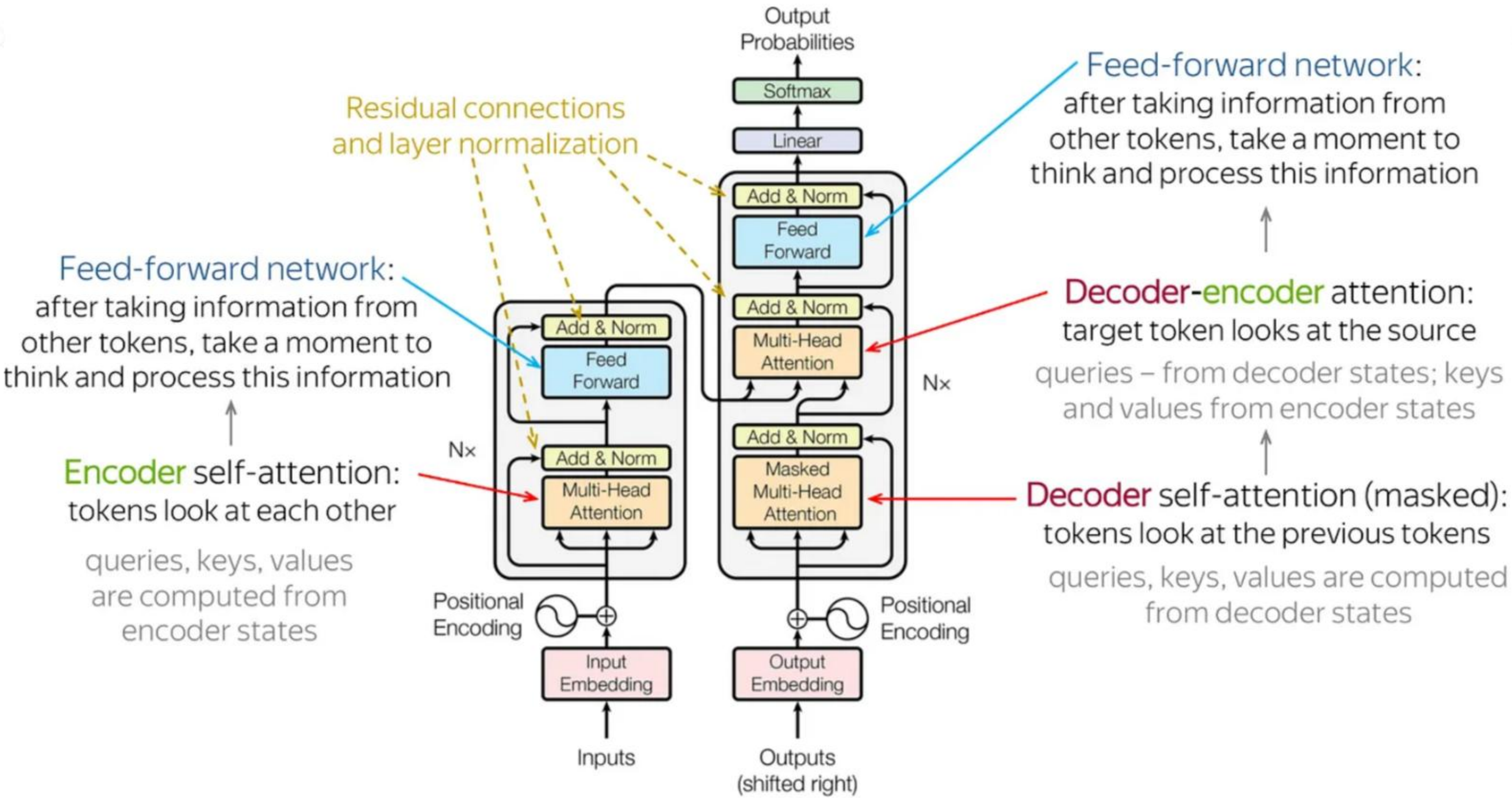
Attention Is All You Need,
https://arxiv.org/pdf/1706.03762.pdf

Transformers

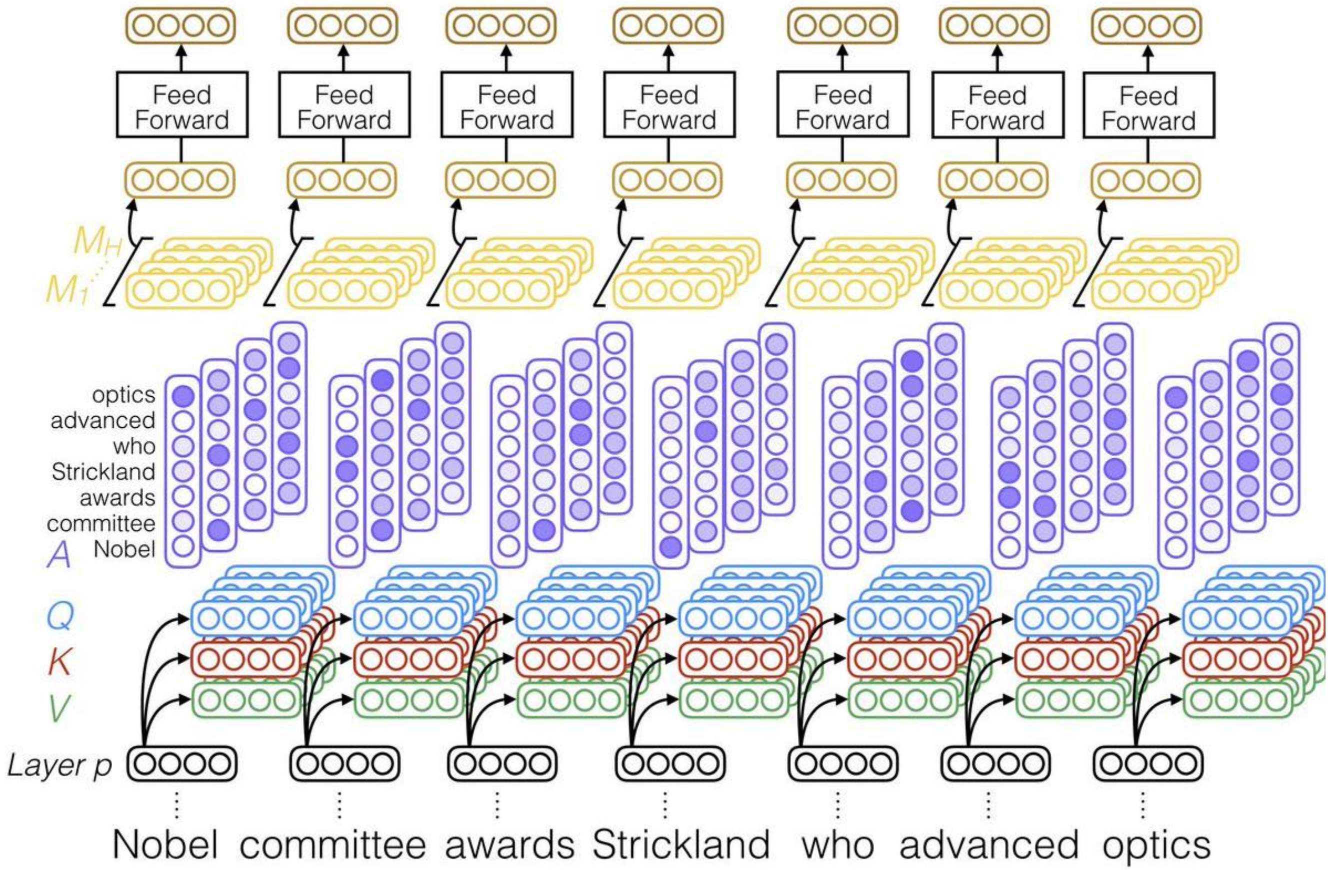
- Encoder-decoder architecture
- First pure attention-based model
- Self-attention is place of recurrence



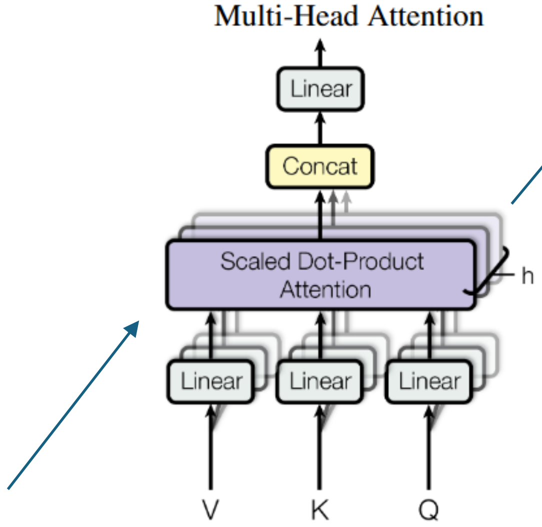
Transformer Architecture

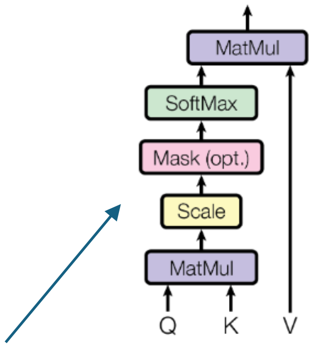


Self Attention – MultiHead



Strubell et al, Linguistically-Informed Self-Attention for Semantic Role Labeling, EMNLP 2018





concat

Attention Is All You Need, https://arxiv.org/pdf/1706.03762.pdf

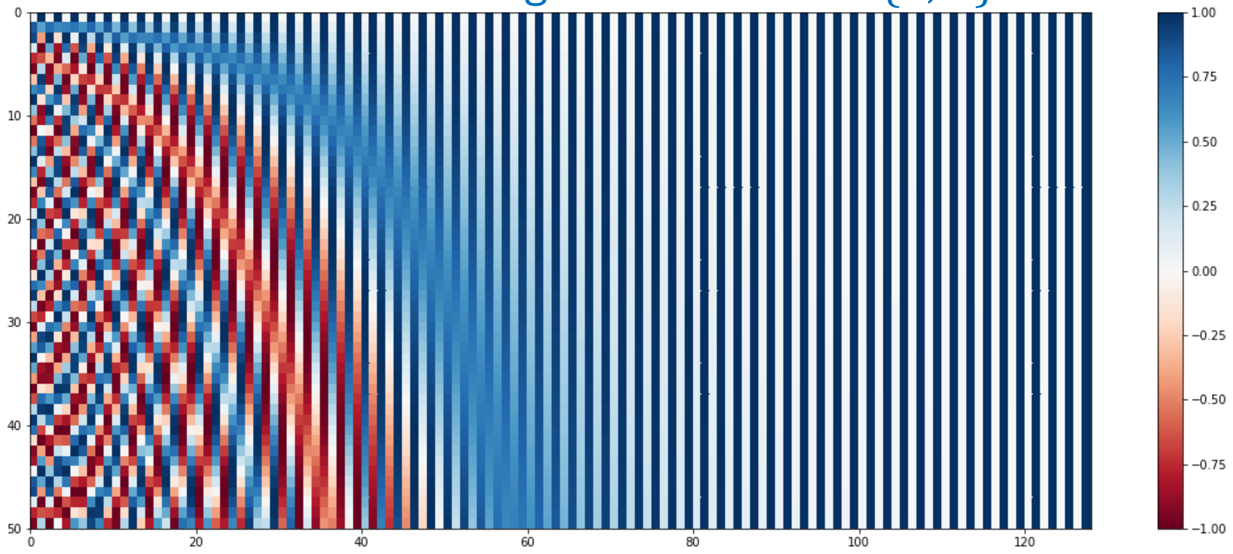
(Absolute) Positional Encoding

- Self-attention is order-independent
- But in sequences we need ordering information
- Positional embeddings are vectors associating unique values to each position in the sequence
- They are summed to the original embedding: input embedding + positional embedding

Their computation uses sines and cosines functions, for reasons we do not cover here

embedding dimensions $i \in \{1, d\}$

Positions p



Encoder Components

- **Input embedding** – Transforms discrete input tokens (e.g. words) into dense vectorial representations
- **Positional encoding** – Adds position information to the embeddings
- **Multi-head self-attention** – Updates the input embedding adding information from the context in which a specific input occurs within the sequence
- **Add & Norm** – Residual connection (Add) plus layer normalization to prevents gradient issues
- **FeedforwardNN/MLP** – The bit on nonlinearity we always need (same network applied to each input element, usually)

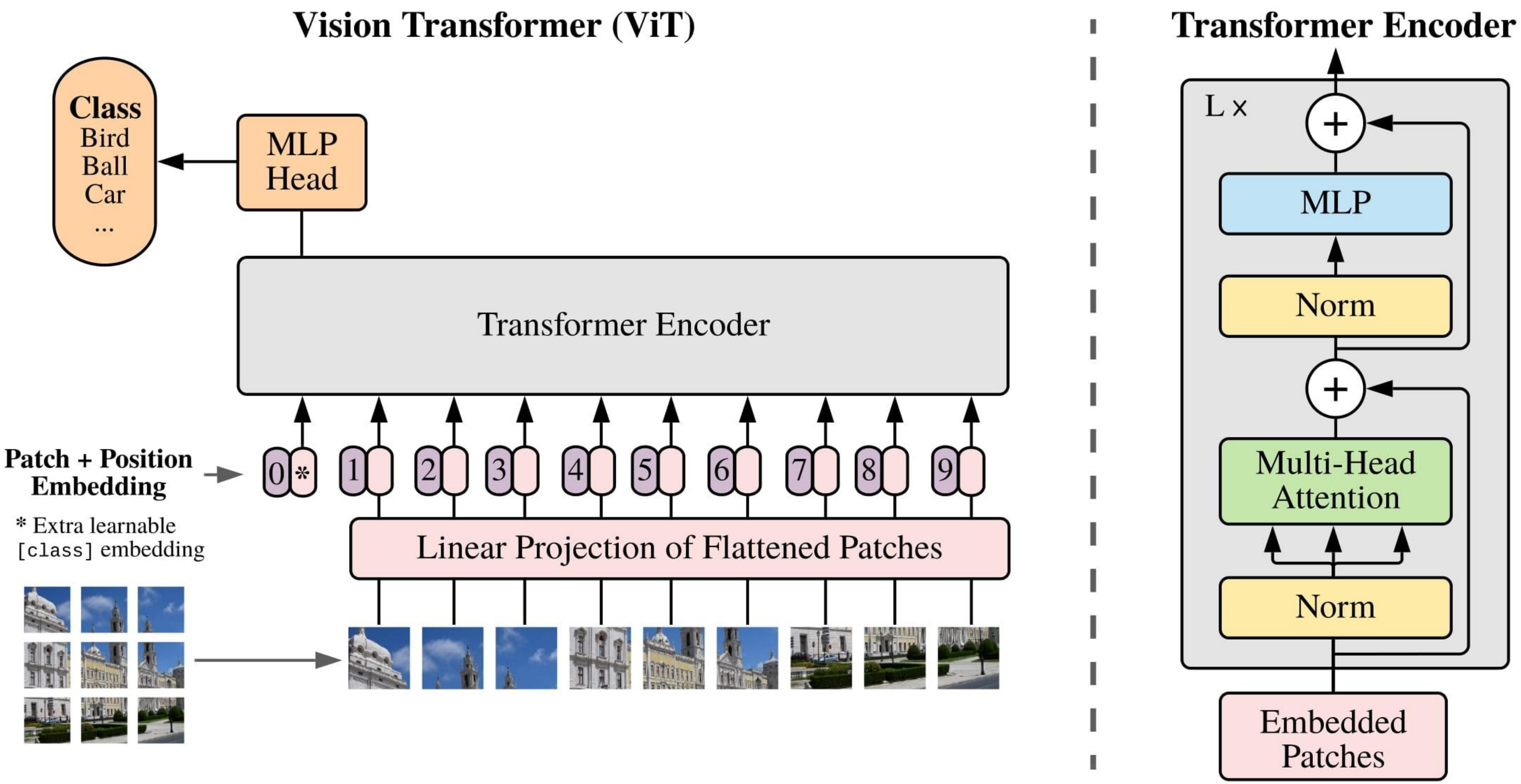
Decoder Components

Same as for the encoder, plus

- **Masked Multi-head self attention** – Like the standard one but we are not allowed to look on the right of the current element (because it was not generated yet)
- **Multi-head cross attention** – Adds the context from the encoder, just like in RNN sequence-to-sequence
- **Output layer** – Predict the current output item (typically a softmax for textual sequences)

The Vision Transformer (ViT)

A. Dosovitskiy et al, ICLR 2021



Wrap-up

Take Home Lessons

- Attention as a powerful tool to obtain context dependent neural representations (embeddings) of elements composing my data
  - Self-attention: relationship between one element of the input and all the input elements (including itself)
  - Cross-attention: relationship between each element of the input and an external context
- Encoder-Decoder scheme
  - A general architecture to compose heterogeneous models and data
  - Decoding allows sampling complex predictions from an encoding conditioned distribution
- Transformers as low-inductive bias architectures
  - Need huge amounts of data to generalize

Next Lectures

- The (silently) missing bit: how do we deal with textual sequences, from natural or biological languages?
- Representing textual information
  - Word embeddings
  - Skip-grams
- Tackling textual modelling tasks
  - Masked language modelling
  - Relevant language model architectures
  - Pretraining and foundation models
- Language modelling in healthcare