The attention mechanism

Prof. Luca Cagliero
Dipartimento di Automatica e Informatica
Politecnico di Torino

Politecnico
di Torino

Lecture goal

- Intuition behind attention
- Preliminary examples in NLP and image processing
- Neural machine translation by jointly learning to align and translate

Intuition

- Attend to the most relevant parts of the input data
- Generate output mainly using the most salient input data parts
- Examples of applications
  - NLP
    - Machine Translation, Question Answering, ...
  - Image Processing
    - Object recognition, image classification, ...

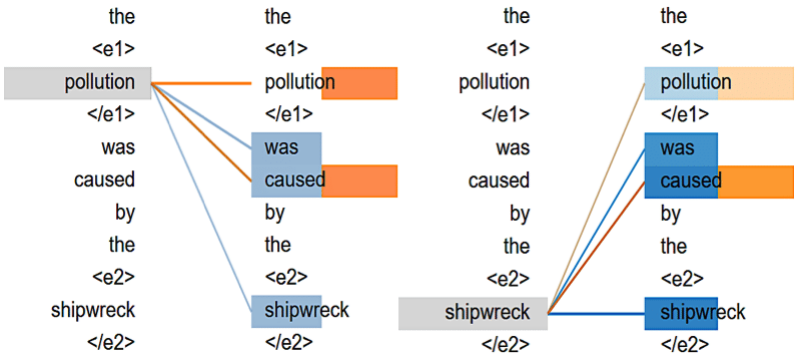
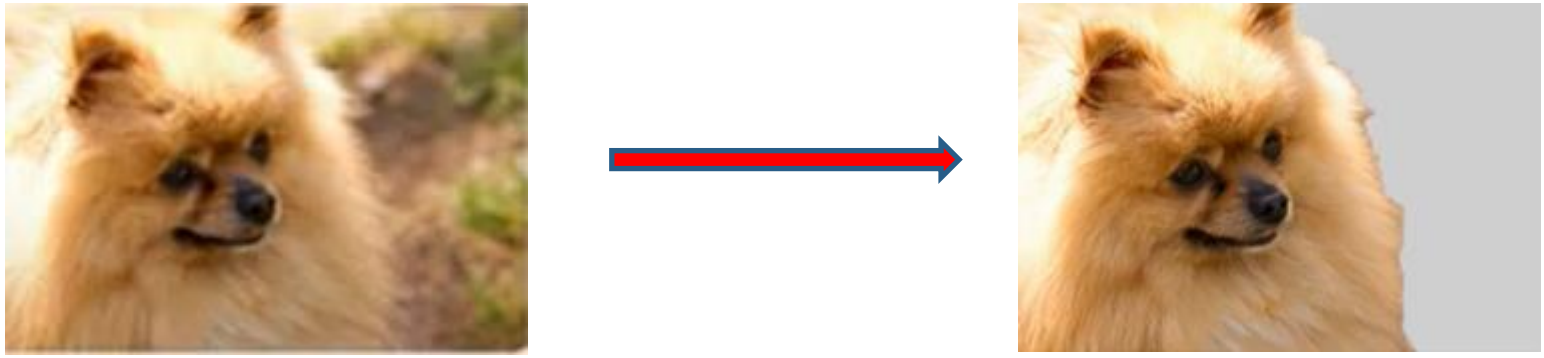


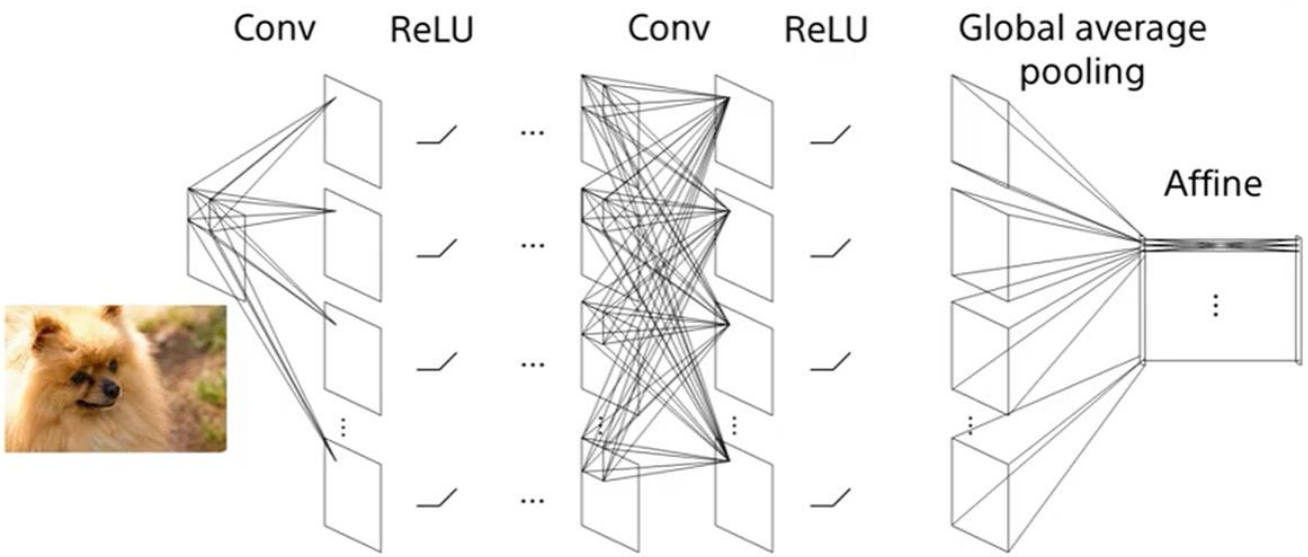
Image processing example: classification

- Mimic human reasoning
  - Focus on specific image portions
- Recognizing specific image regions is instrumental for classifying the main image subject ("dog")



Example: image classification

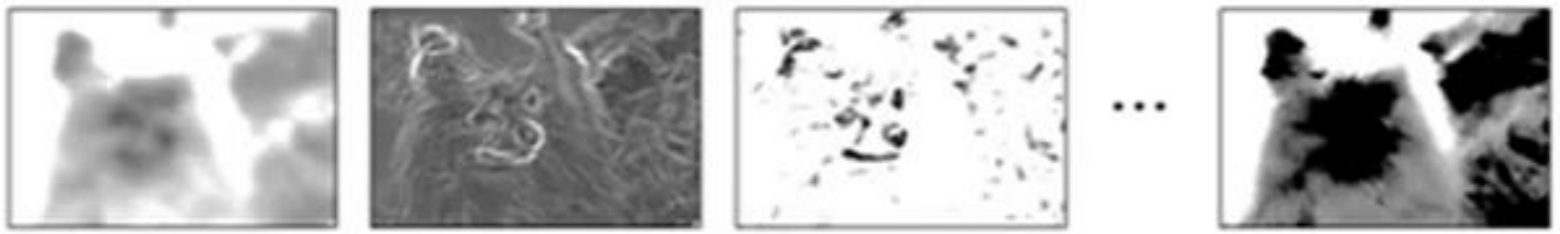
- Convolutional Neural Networks process the entire set of image features
- They repeat the process for the entire image



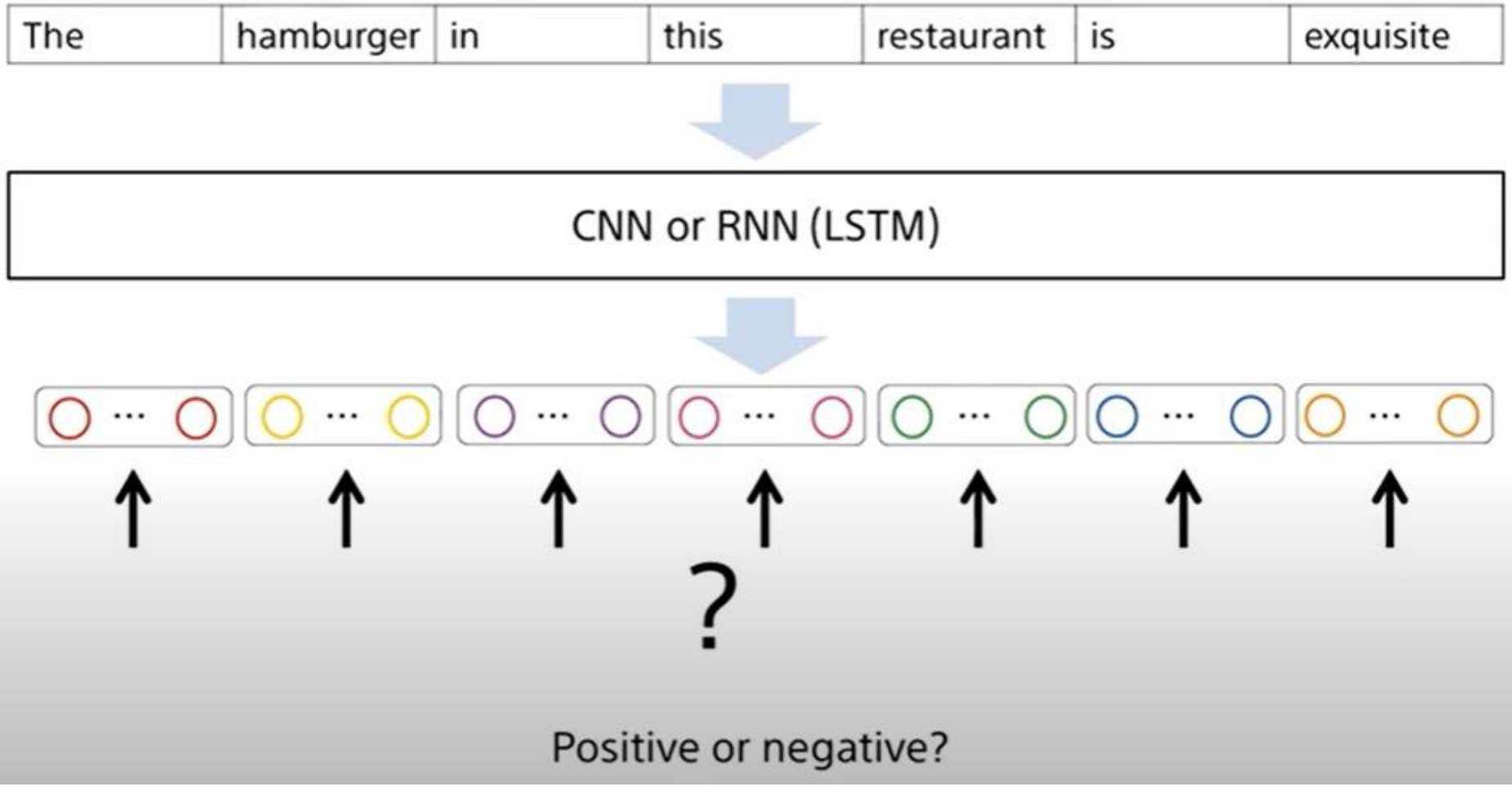
Sony Network Communication Inc. Introduction to Deep Learning. Attention Mechanism - Introduction to Deep Learning

Example: image classification

- Build a Neural Network that dynamically focuses on specific areas and features according to the input

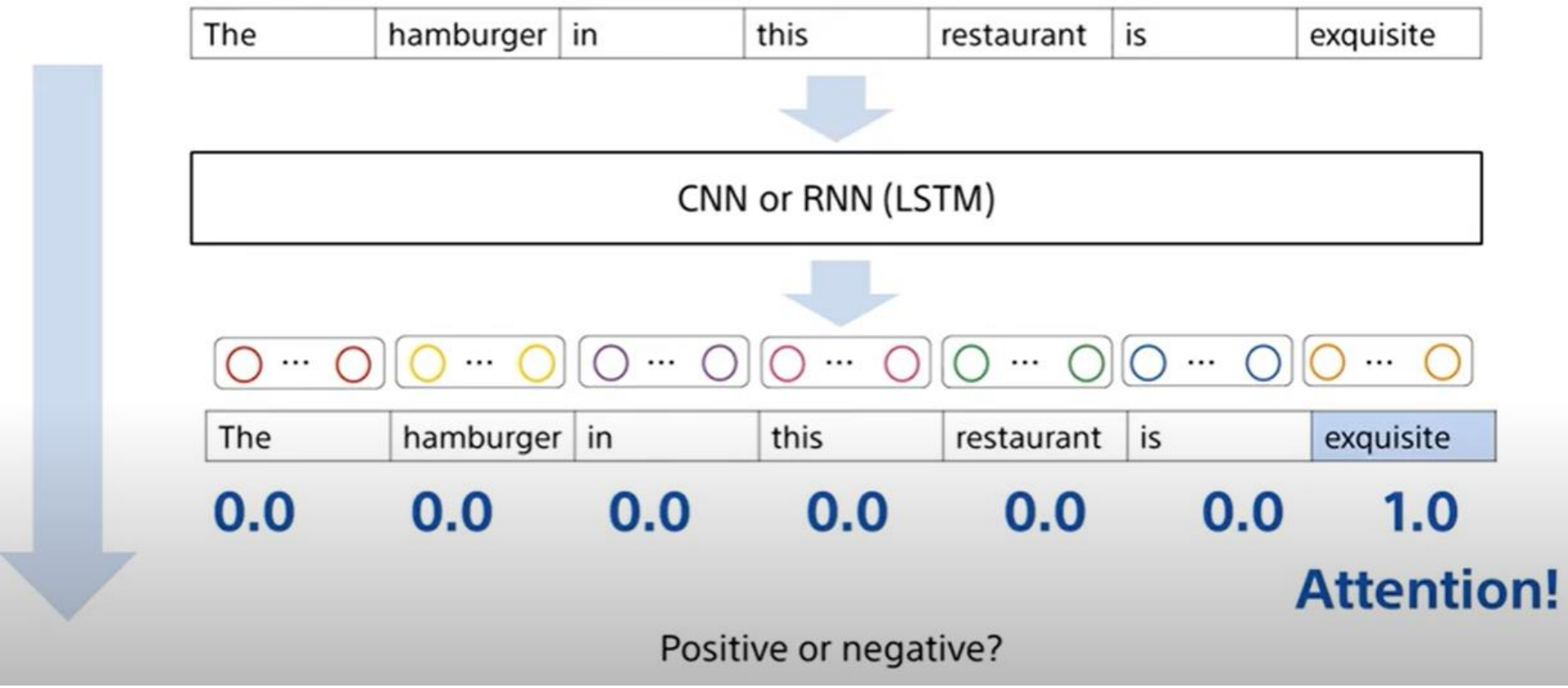


NLP example: sentiment analysis



Sony Network Communication Inc. Introduction to Deep Learning. Attention Mechanism - Introduction to Deep Learning

NLP example: sentiment analysis



Sony Network Communication Inc. Introduction to Deep Learning. Attention Mechanism - Introduction to Deep Learning

Attention mechanism in Neural Machine Translation

- Encoder-decoder architecture
  - An encoder neural network reads and encodes a source sentence into a fixed-length vector
  - A decoder then outputs a translation from the encoded vector
- The whole encoder–decoder system is jointly trained to maximize the probability of a correct translation given a source sentence
- Issue
  - All the necessary information of a source sentence need to be compressed into a fixed-length vector
  - Can be difficult to cope with long sentences
    - Especially those that are longer than the sentences in the training corpus

<u>Solution: align and translate jointly</u>

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

Attention mechanism in Neural Machine Translation

- Encoder-decoder architecture
  - Encoder: Bidirectional RNNs
  - Decoder: emulate search through a source sentence

- Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated.
- The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

NLP example: Machine Translation

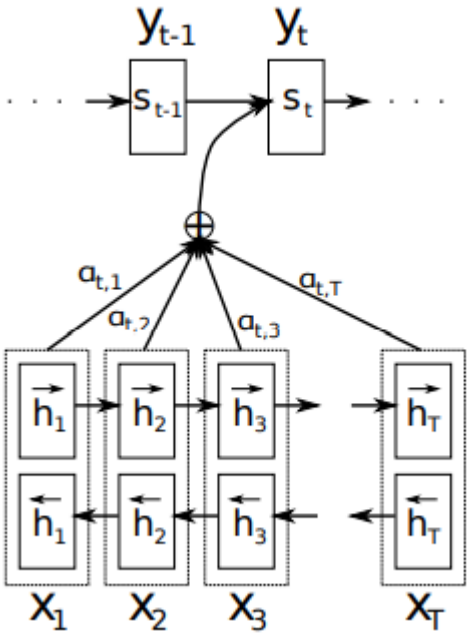
- Decoder translation probability:

$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c),$$

$$p(y_i|y_1, \ldots, y_{i-1}, \mathbf{x}) = g(y_{i-1}, s_i, c_i)$$

- RNN hidden state:

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



Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

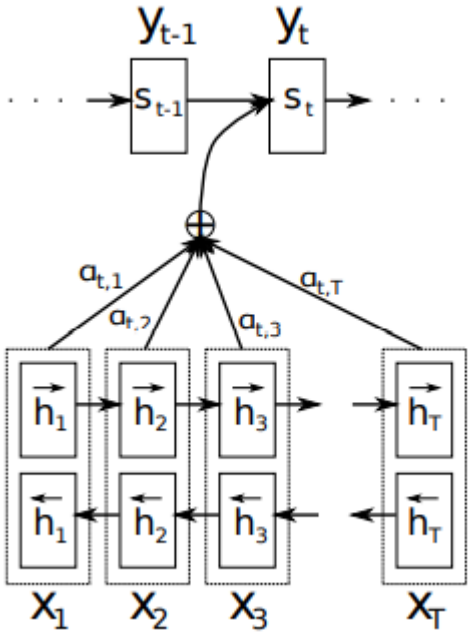
NLP example: Machine Translation

- Context vector c_i
  - Depend on a sequence of annotations to which an encoder maps the input sentence
- Annotation
  - Contains information about the whole sequence
  - Strong focus on the parts surrounding the i-th word of the input sequence

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$



Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

NLP example: Machine Translation

- Context vector c_i
  - Depend on a sequence of annotations to which an encoder maps the input sentence
- Annotation
  - Contains information about the whole sequence
  - Strong focus on the parts surrounding the i-th word of the input sequence

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j$$

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})}$$

$$e_{ij} = a(s_{i-1}, h_j)$$

Alignment model

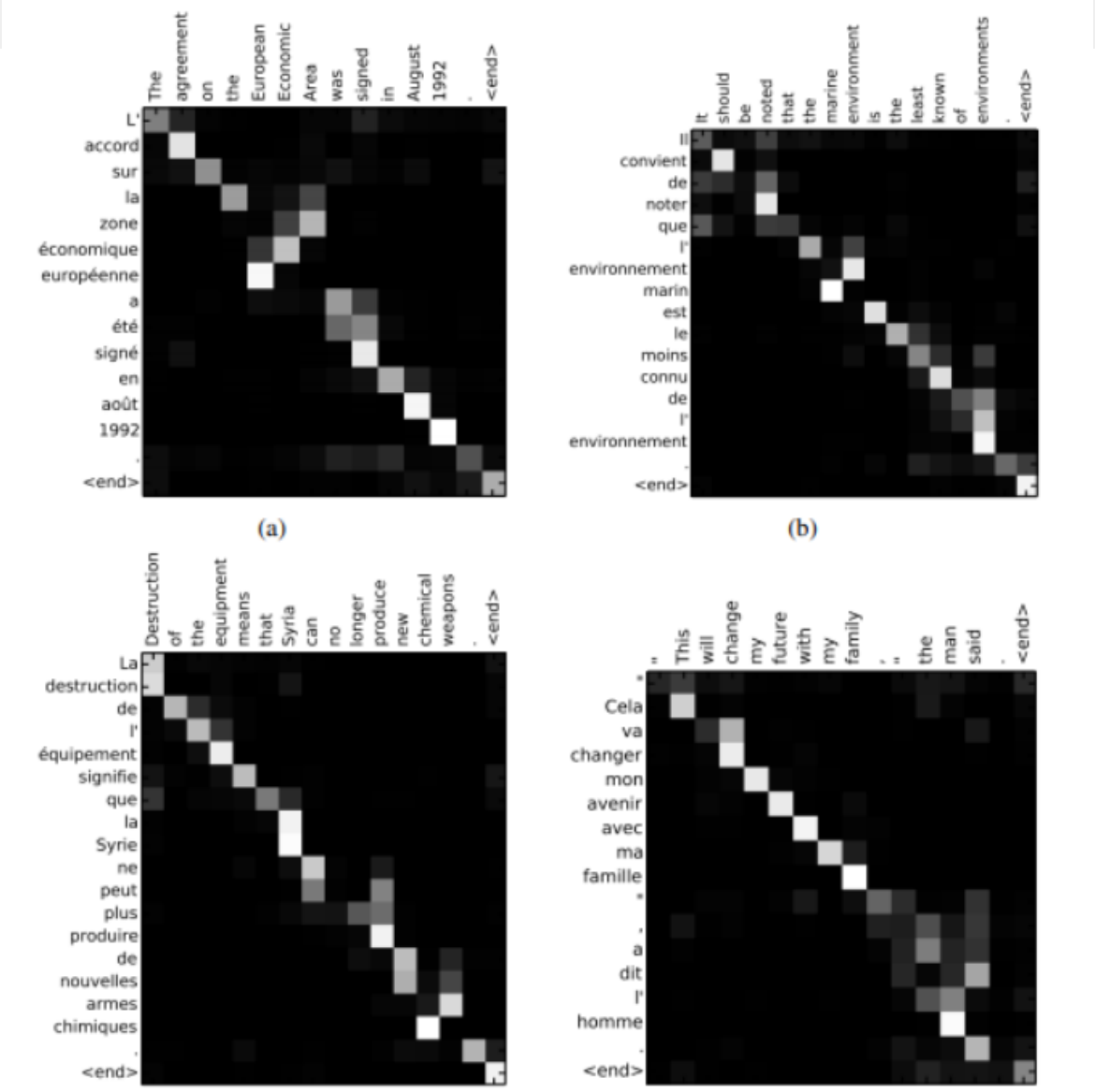
How well the inputs around position j and the output at position i match (based on state s_{i-1})

Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015

Saliency maps

- Plot used for getting insightful model explanations
- X- and Y-axis are words in the source and target sentences, respectively
- Each pixel shows the attention weight of the j-th source word for the i-th target word
  - Ranging from 0 (black) to 1 (white)

Saliency maps: example (English vs. French)



Additional reading on Neural Machine Translation

- Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. ICLR 2015
- Download and read the paper: https://arxiv.org/pdf/1409.0473.pdf

The attention mechanism

- Main steps
  - Identify which portions of the input text to attend to
  - Extract the features with higher attention

This step can be modelled as a search task!

The attention mechanism

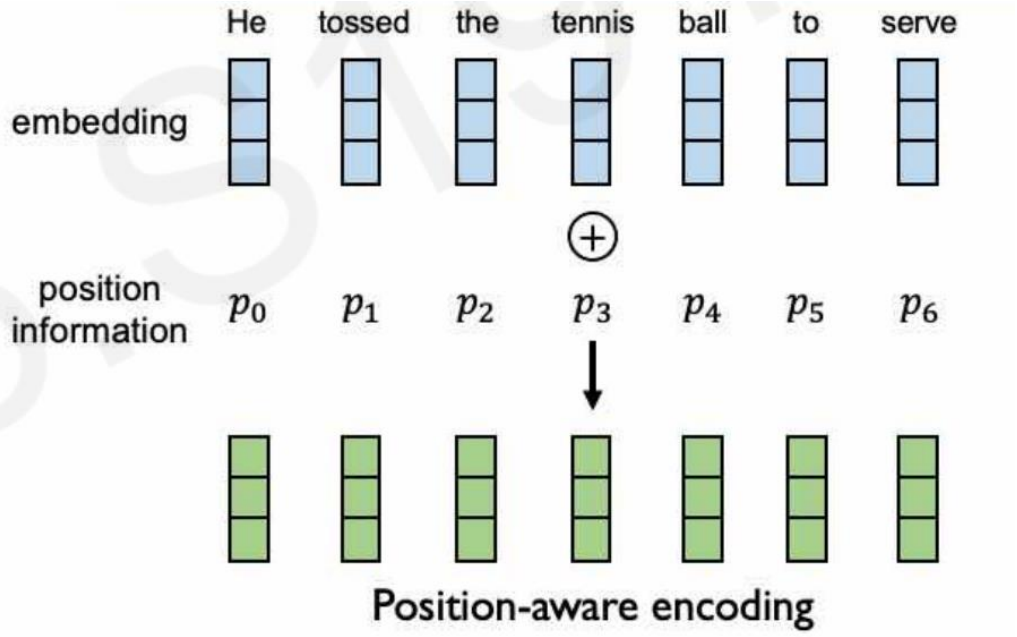
- Main steps
  - Identify which portions of the input text to attend to
  - Extract the features with higher attention

Steps

1. Compute the similarities between the query and each of the keys
2. store the similarities in the attention mask
3. Extract the values corresponding to the highest attention score

Learning self-attention

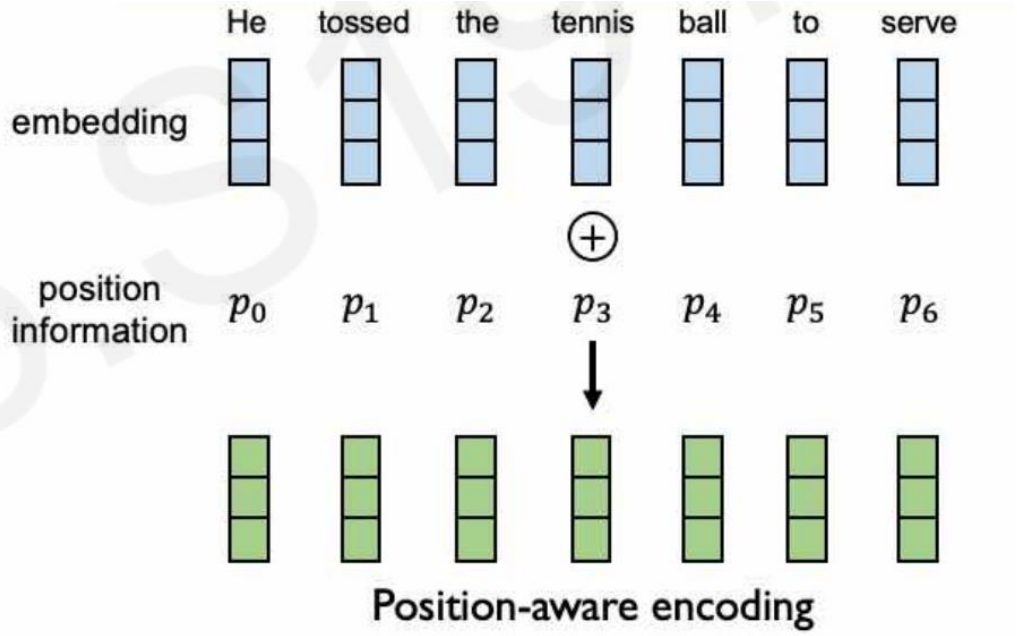
- Identify and attend the most relevant features in the input data



6.S191 Introduction to Deep Learning, MIT

Learning self-attention

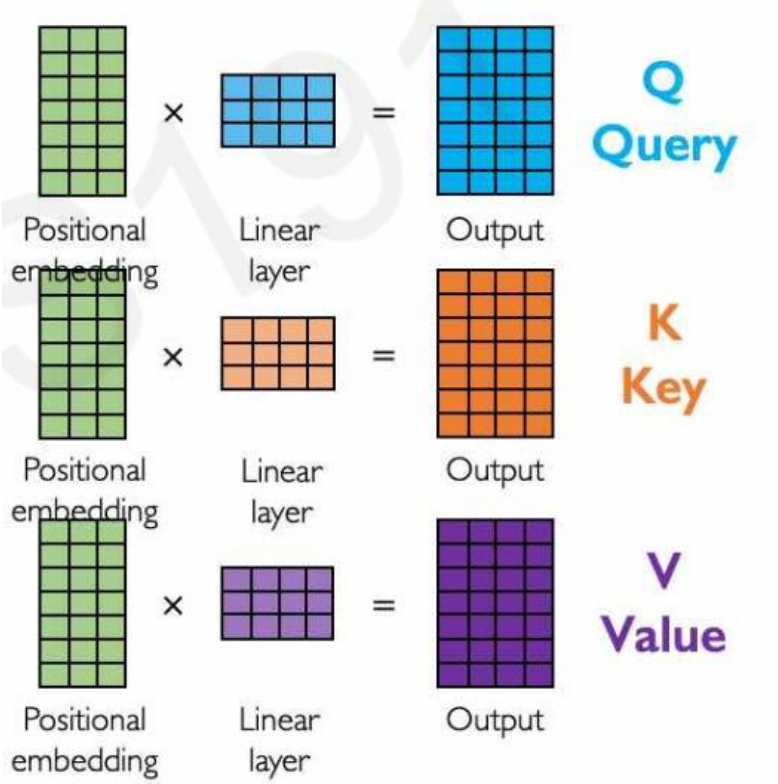
- Steps
  1) Encode position information to understand the order
     - Because data is fed in all at once!



6.S191 Introduction to Deep Learning, MIT

Learning self-attention

- Steps
  2) Extract query, key, value for search

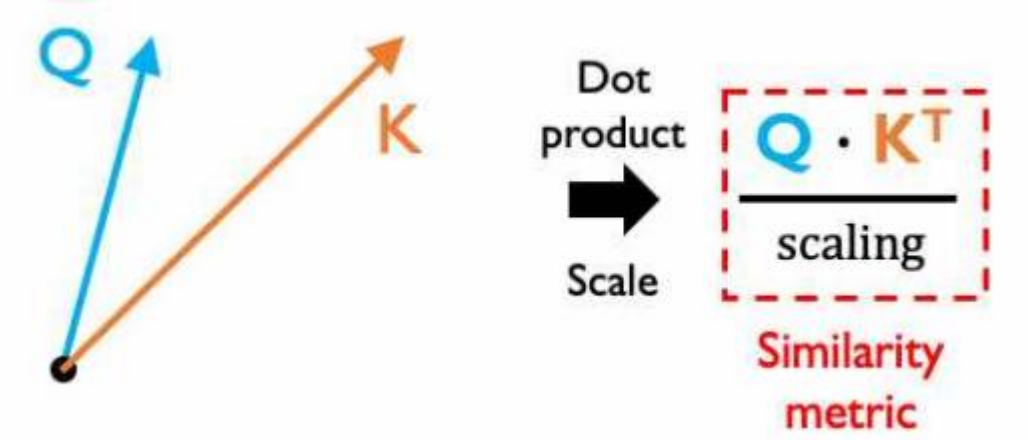


6.S191 Introduction to Deep Learning, MIT

Learning self-attention

- Steps

  3) Compute attention weights between each query and key using cosine similarity

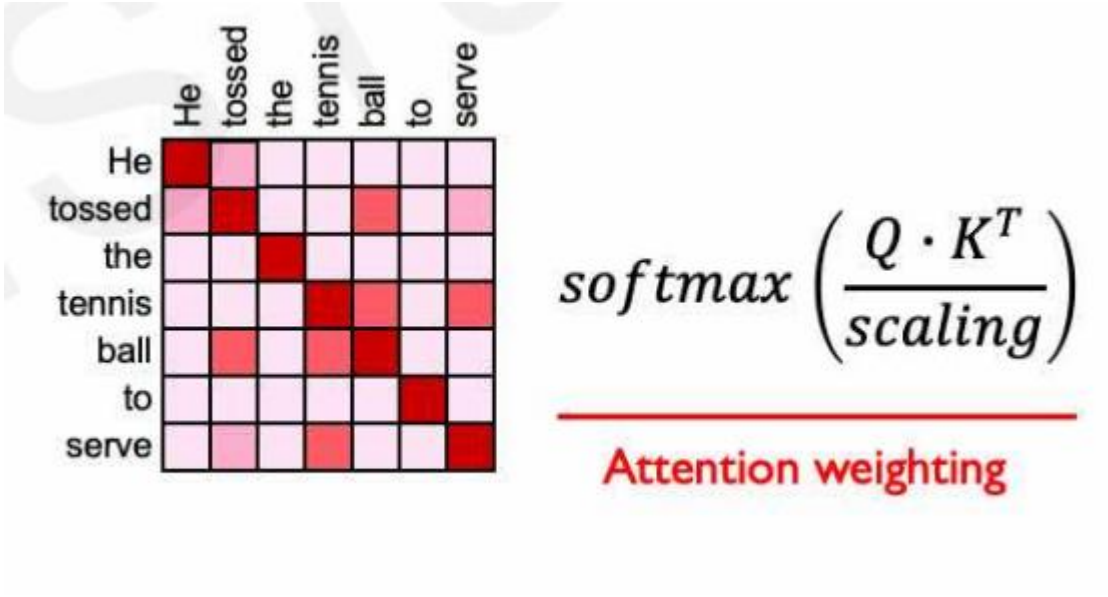


6.S191 Introduction to Deep Learning, MIT

Learning self-attention

- Steps

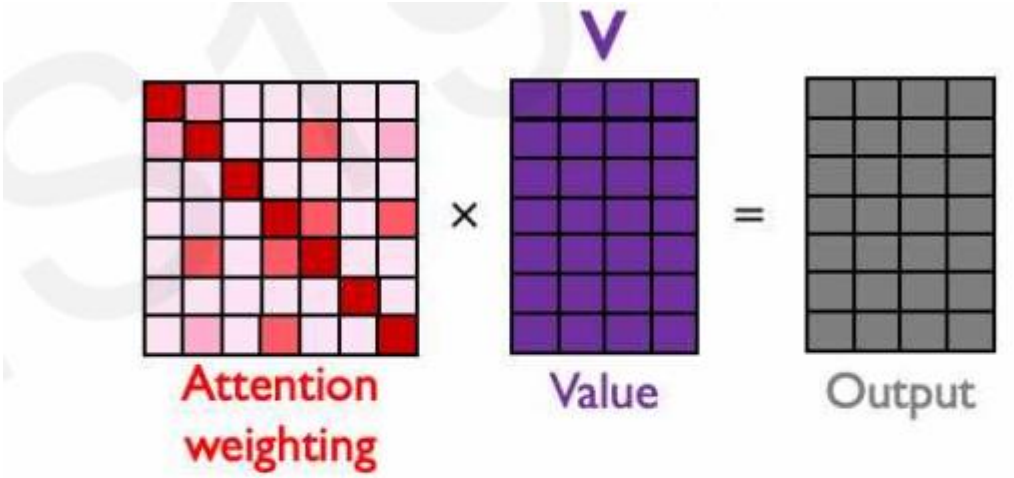
  4) Apply Softmax to get the attention weighting (probability distributions) indicating where to attend to



6.S191 Introduction to Deep Learning, MIT

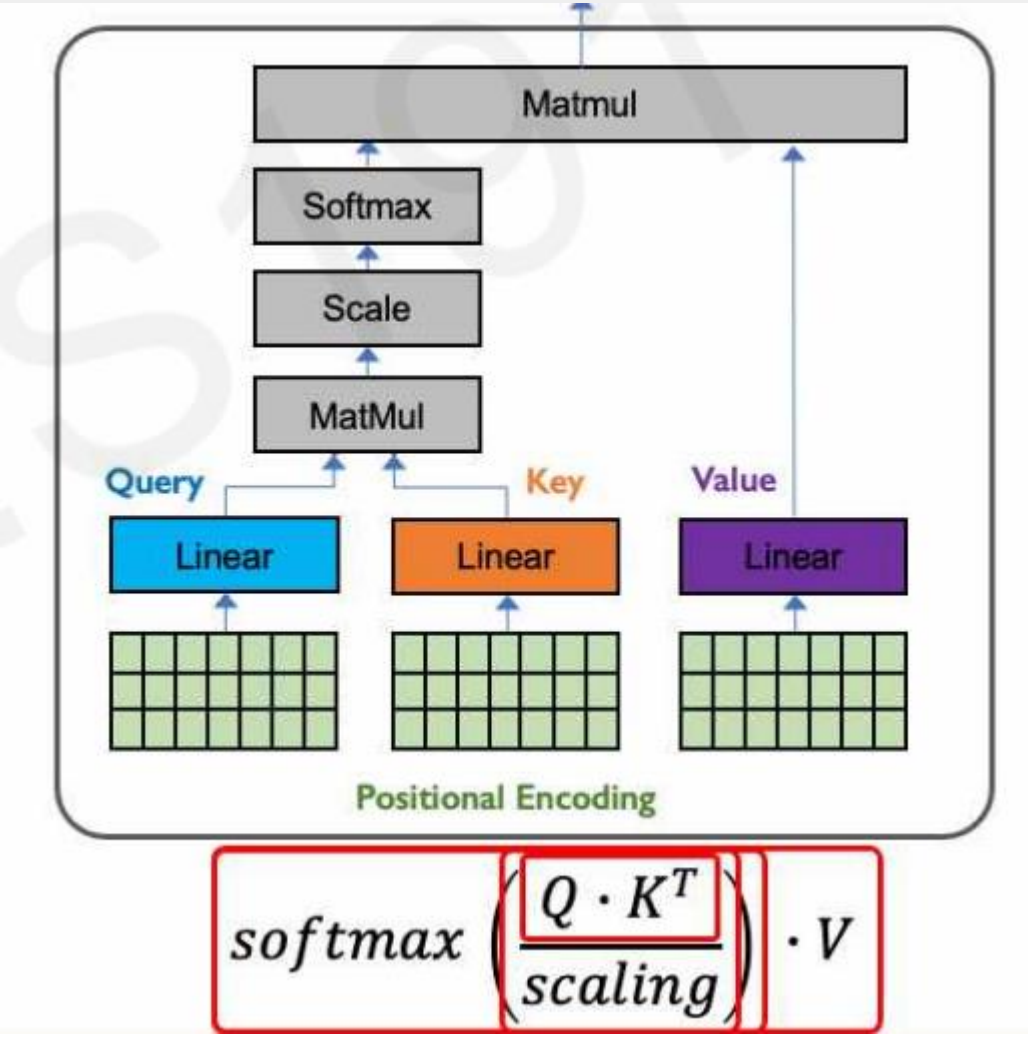
Learning self-attention

- Steps
  5) Extract features with high attention



6.S191 Introduction to Deep Learning, MIT

Self-attention head



Each head attend to a specific portion of the input data.

To attend to multiple portion (or to multiple data granularities):
a stack of multiple attention heads

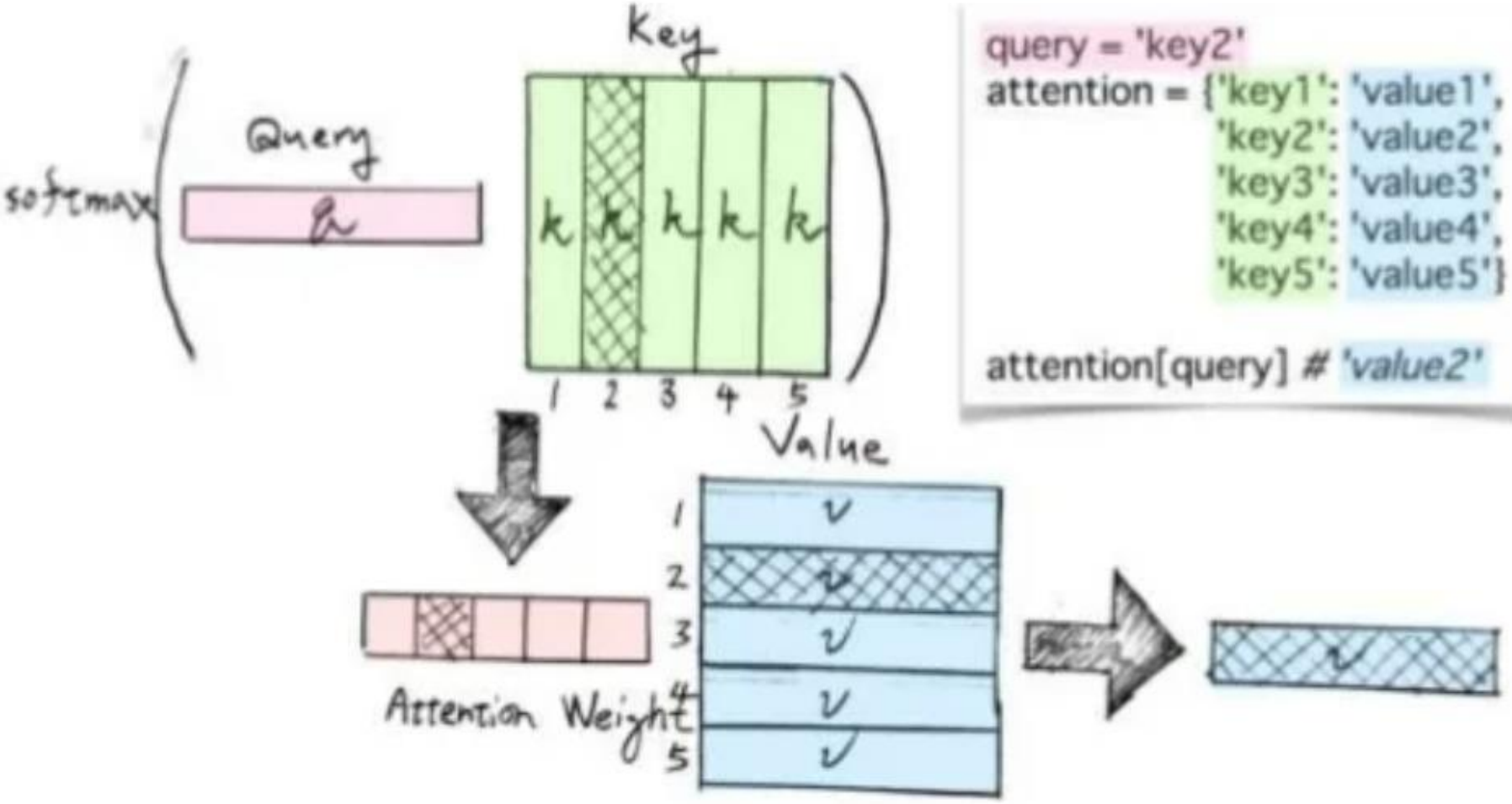
6.S191 Introduction to Deep Learning, MIT

The attention mechanism: problem statement

- Goal
  - Mimic the retrieval of a value v_i for a query q based on a key k_i in a database

$$\text{Attention}(q, k_i, v_i) = \sum_i \text{similarity}(q, k_i) \times v_i$$

The attention mechanism summarized

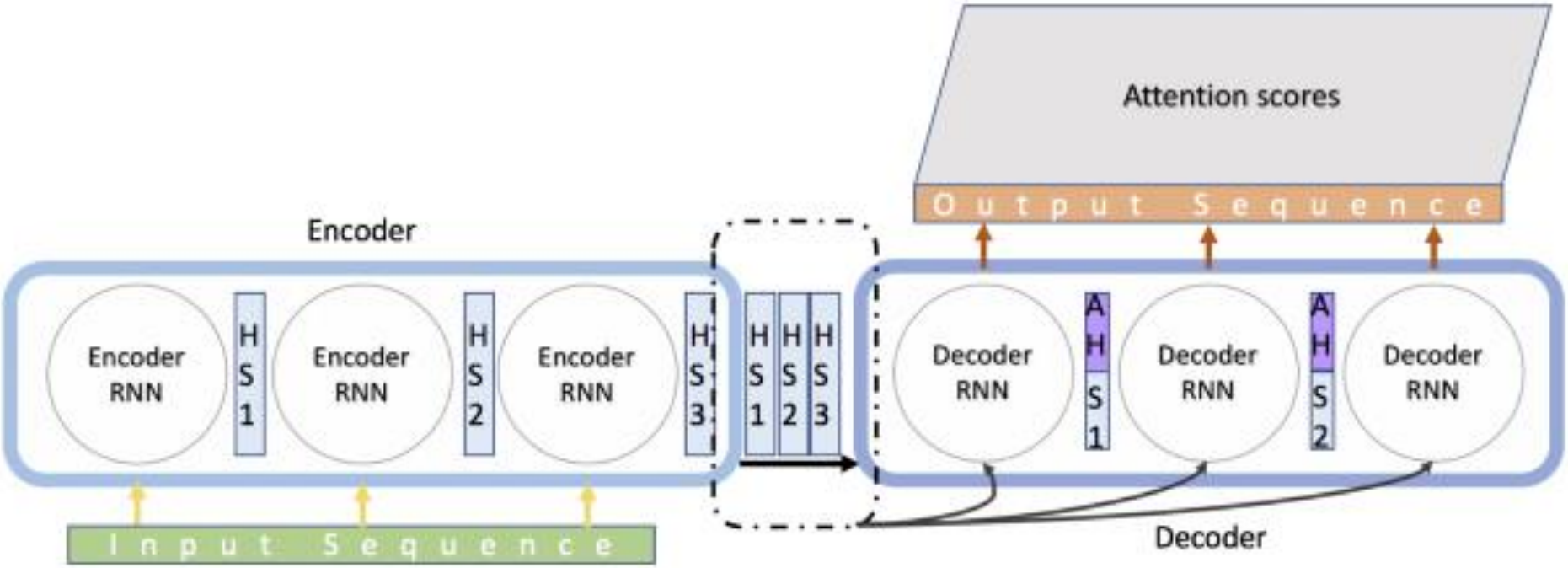


Daiki Tanaka. Attention is all you need at NIPS 2017. SlideShare.

The attention mechanism: key advantages

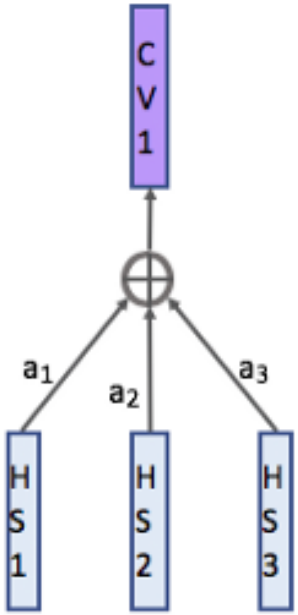
- Performance improvement
  - No recurrence -> facilitate parallelization (per layer)
  - Focus on specific text portions
  - Few training steps
- Overcome the vanishing gradient and explosion
  - Shortcut to faraway states
  - Facilitate long range dependencies
- Explain the model
  - Attention distribution indicates on which sentence portion the decoder is focusing on

The attention mechanism



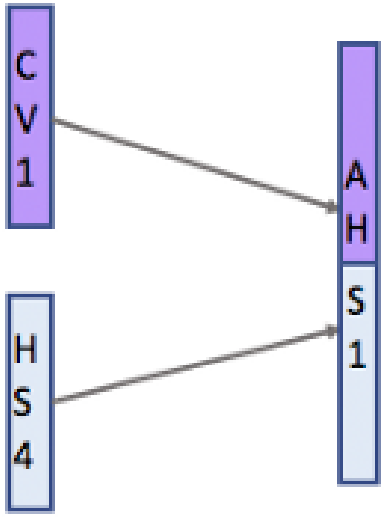
Attention mechanism: the context vector

- It is computed as the weighted sum of the input hidden state vectors



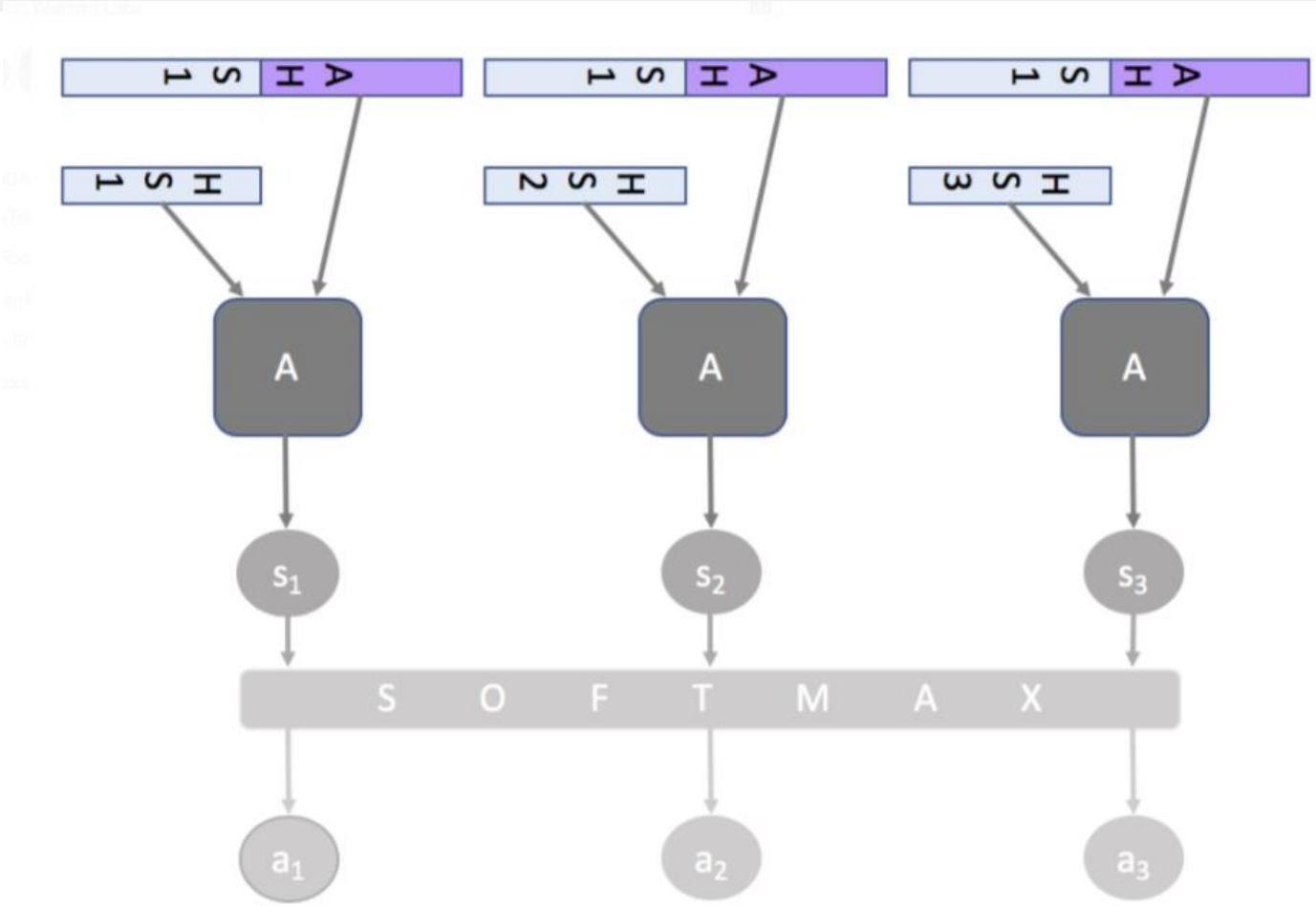
https://towardsdatascience.com/ (latest acces: June 2021)

Attention mechanism: the context vector



- The generated context vector is combined with the hidden state vector by concatenation to form a *attention hidden vector*
- The attention hidden vector is used to predict the output at that time instance

Attention mechanism: the alignment model



https://towardsdatascience.com/ (latest acces: June 2021)

Attention mechanism: the alignment model

- The alignment model is trained jointly with the seq2seq model initially
- It scores how well an input (represented by its hidden state) matches with the previous output (represented by attention hidden state) and quantifies this matching for every input with the previous output
- Then a softmax is taken over all these scores and the resulting number is the attention score for each input

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

The attention mechanism: the key steps

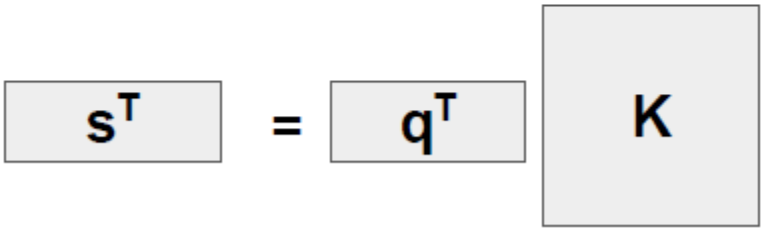
1. Given a query vector **q**, a key vector $\mathbf{k_i}$ representing value $\mathbf{v_i}$
   - S_i : similarity score between **q** and $\mathbf{k_i}$
2. Normalize the similarity scores to sum to 1
   - $P_i = \text{softmax}(S_i)$
3. Compute z as the weighted sum of the value vectors $\mathbf{v_i}$ weighted by their scores p_i

$$z = \sum_{i=1}^{L} p_i v_i$$

Attention types

- Given a query vector **q**, a key vector $\mathbf{k_i}$ representing value $\mathbf{v_i}$
  - S_i : similarity score between **q** and $\mathbf{k_i}$
- Additive attention
  - $Si = w_3 \tanh(\mathbf{w_2}^T\mathbf{q} + \mathbf{w_1}^T\mathbf{k_i})$
- Dot product attention
  - $S_i = \mathbf{q}^T\mathbf{k_i}$
- Scaled product attention
  - $S_i = \mathbf{q}^T\mathbf{k_i} / \mathbf{d_k}^{1/2}$

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$



Acknowlegdements and copyright license

- Copyright licence
  - Attribution + Noncommercial + NoDerivatives

- Acknowledgements
  - I would like to thank Dr. Moreno La Quatra, who collaborated to the writing and revision of the teaching content
- Affiliation
  - The author and his staff are currently members of the Database and Data Mining Group at Dipartimento di Automatica e Informatica (Politecnico di Torino) and of the SmartData interdepartmental centre
    - https://dbdmg.polito.it
    - https://smartdata.polito.it

Thank you!