

INTRODUCTION TO DEEP LEARNING

UPC TelecomBCN Barcelona (4th edition). Spring Edition.



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH



Instructors:



Xavier
Giró-i-Nieto



Ferran
Marqués



Ramon
Morros



Montse
Pardàs



Javier
Ruiz



Elisa
Sayrol



Veronica
Vilaplana



Gerard
Gallego



Albert
Mosella

Teaching Assistants:

Day 6 Lecture 4

The Transformer



Xavier Giro-i-Nieto



@DocXavi



xavier.giro@upc.edu

Associate Professor
Universitat Politècnica de Catalunya



Acknowledgments



Marta R. Costa-jussà

Associate Professor
Universitat Politècnica de Catalunya



Carlos Escolano

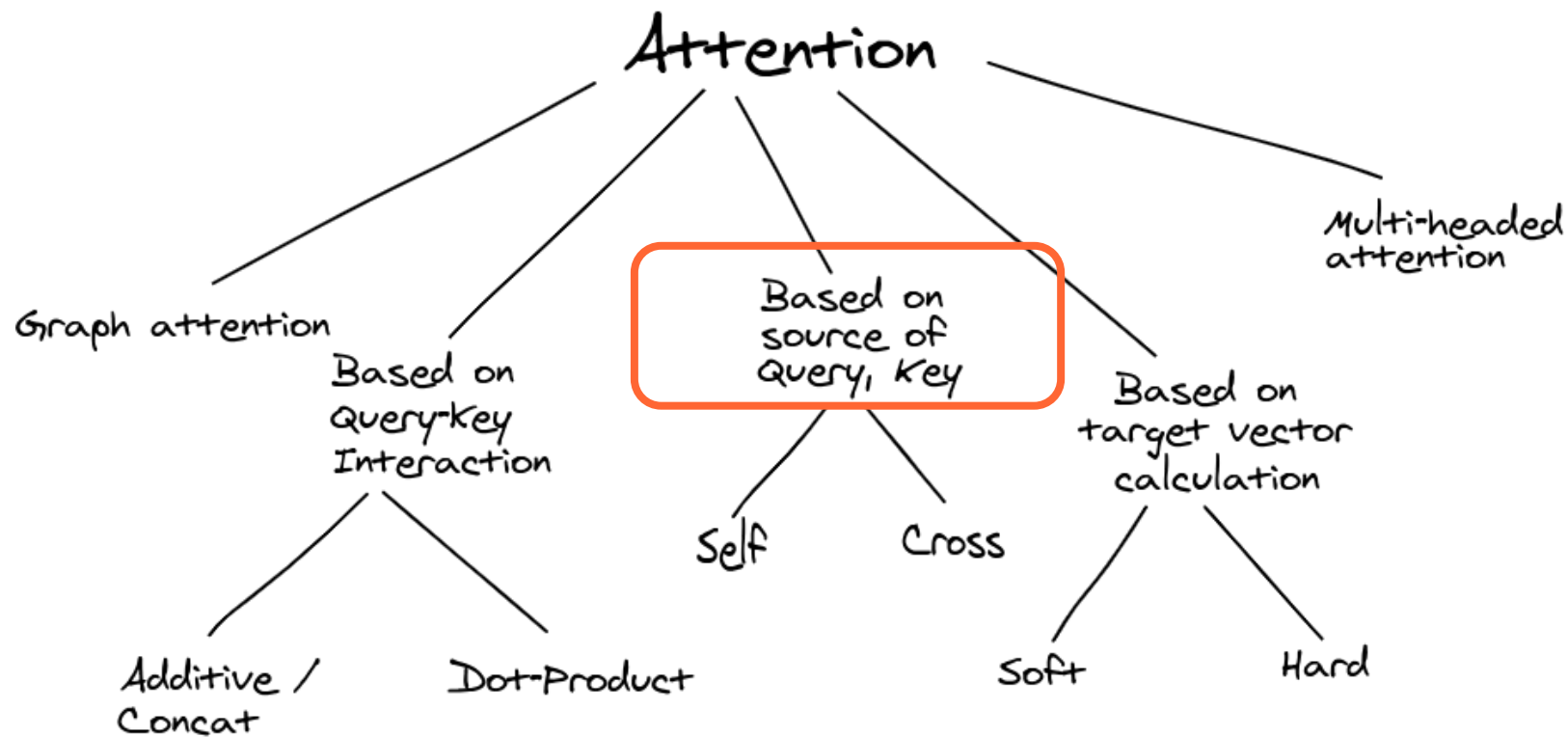
PhD Candidate
Universitat Politècnica de Catalunya



Outline

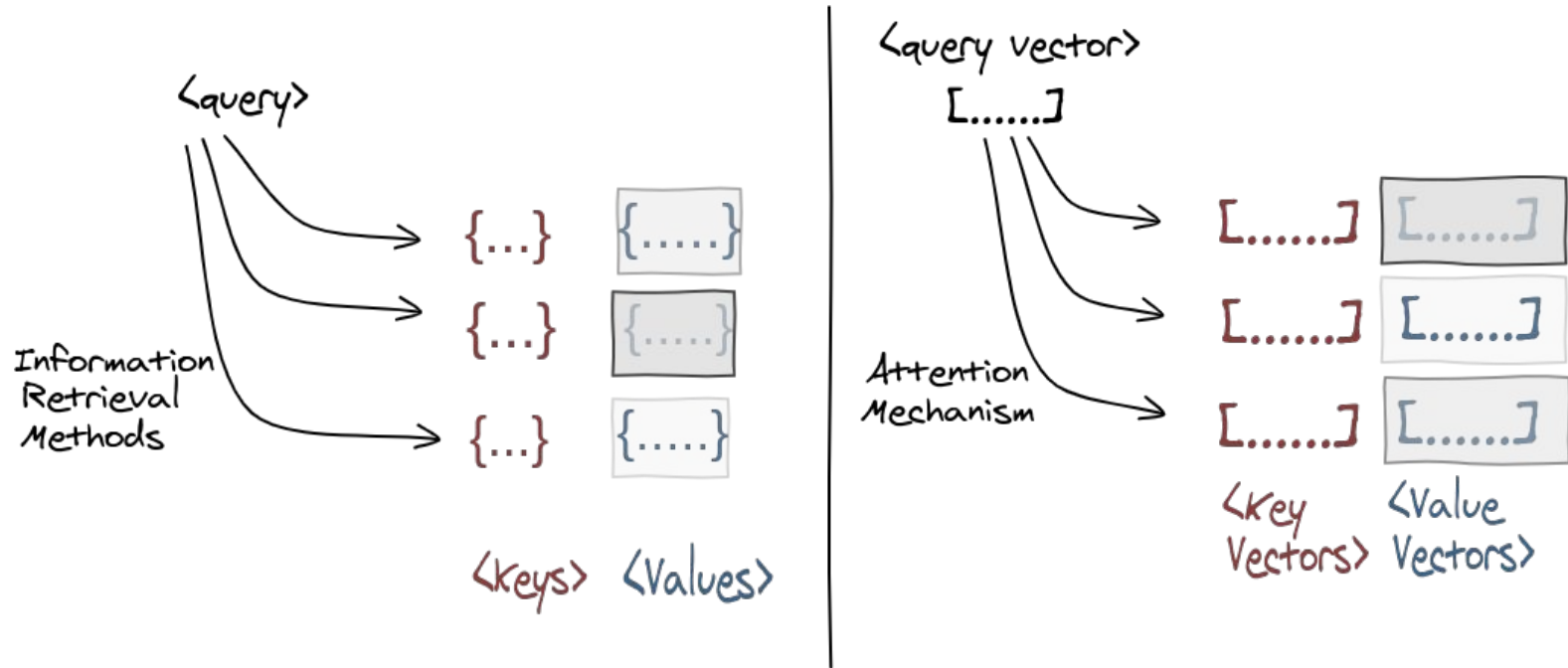
1. Reminders

Reminder

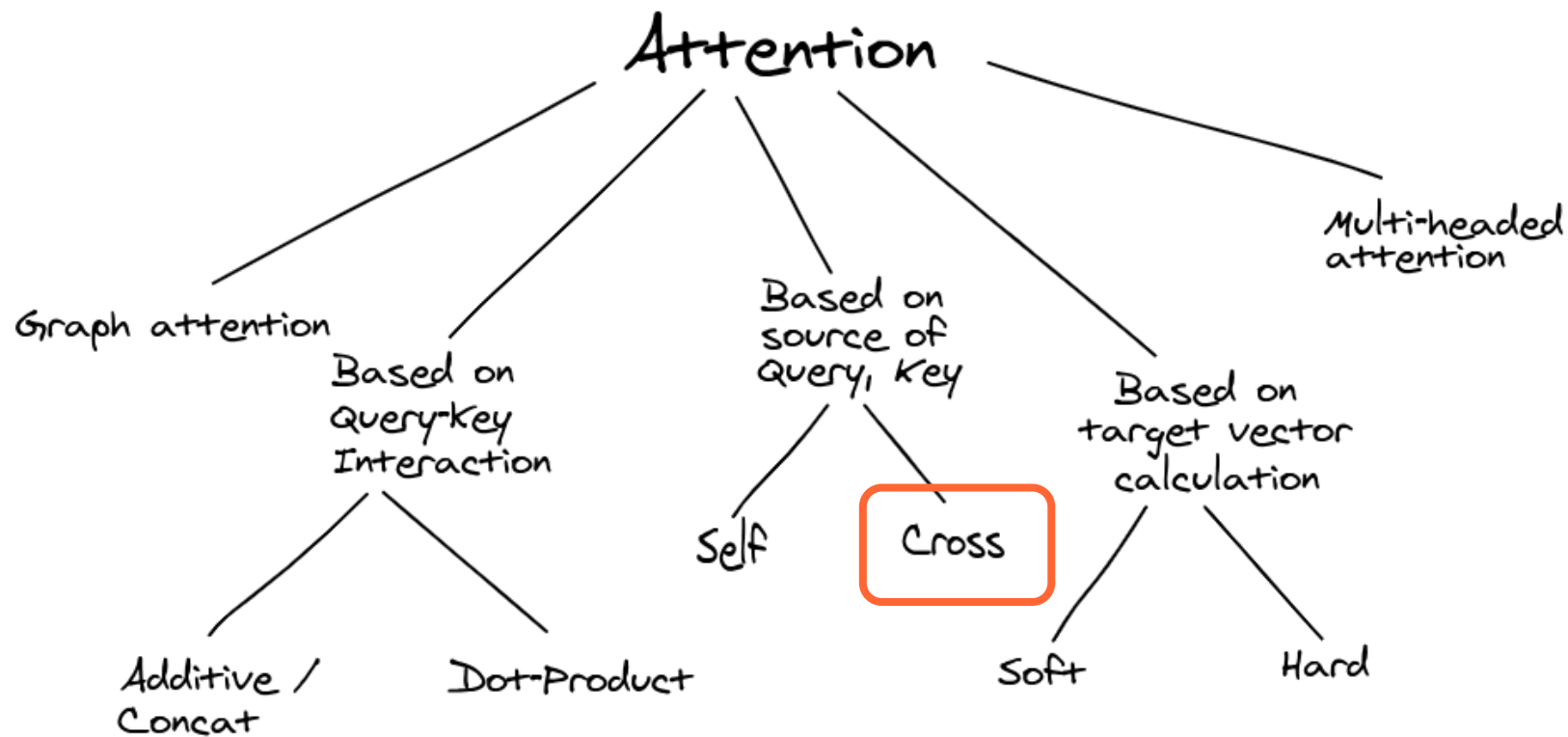


Reminder

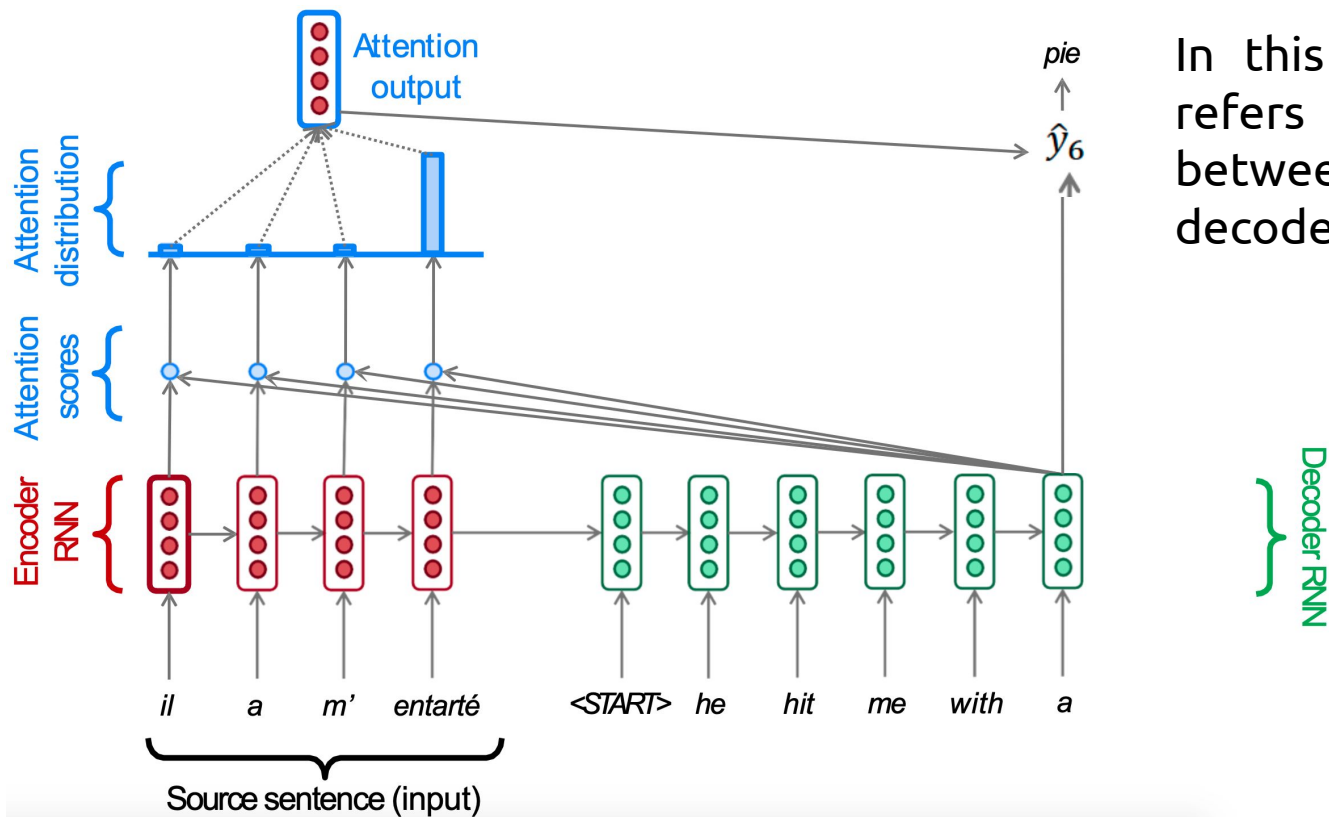
Attention is a mechanism to compute a context vector (c) for a **query (Q)** as a weighted sum of **values (V)**.



Reminder

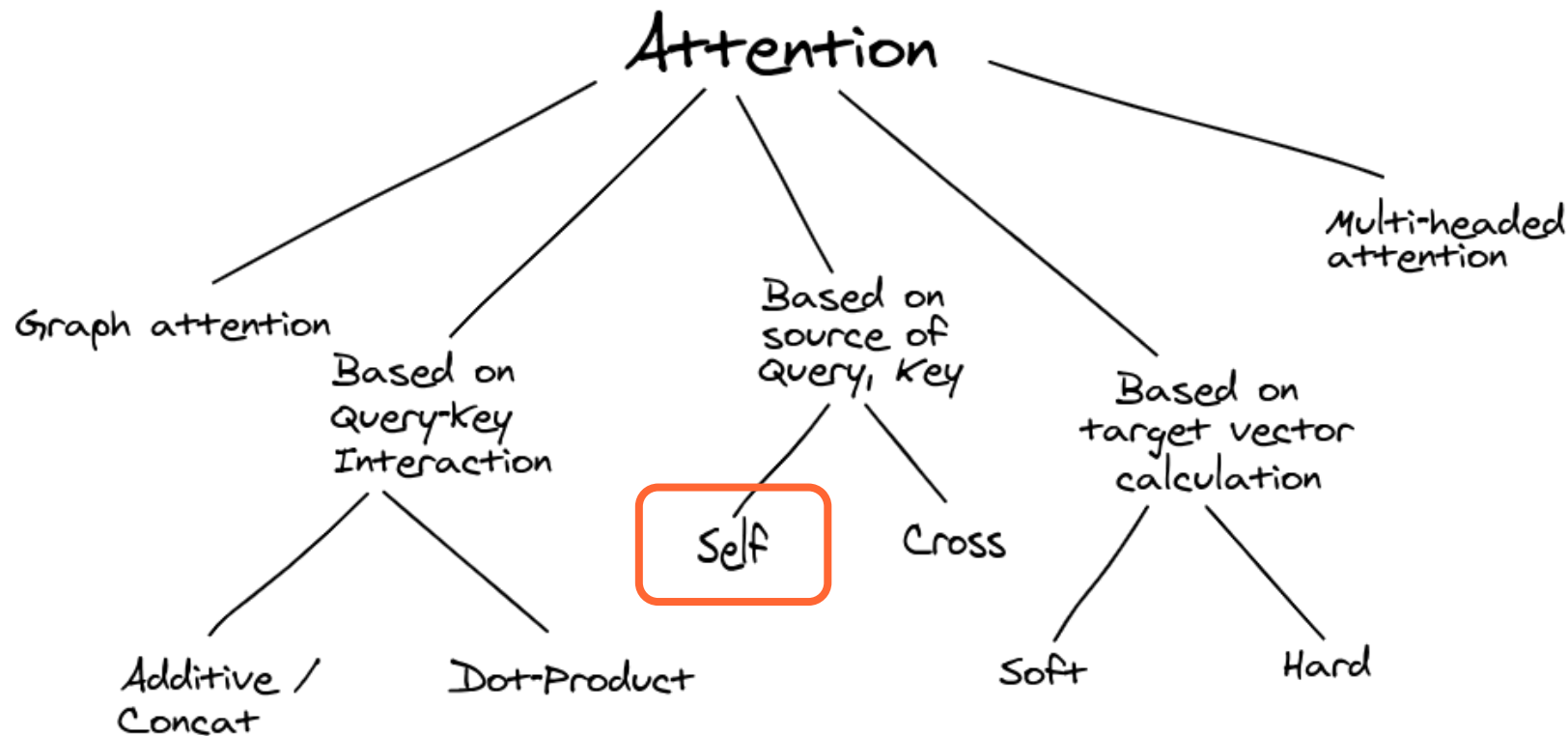


Reminder: Seq2Seq with Cross-Attention



In this case, **cross-attention** refers to the attention between the encoder and decoder states.

What may the term “self” refer to, as a contrast of “cross”-attention ?



Outline

1. Motivation

2. Self-attention

Self-Attention (or intra-Attention)

Self-attention refers to attending to other elements from the SAME sequence.

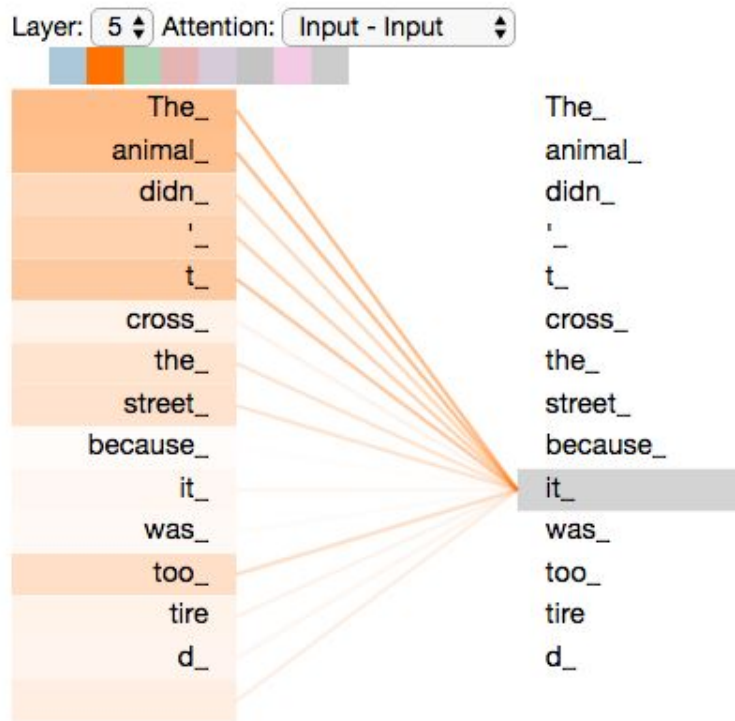
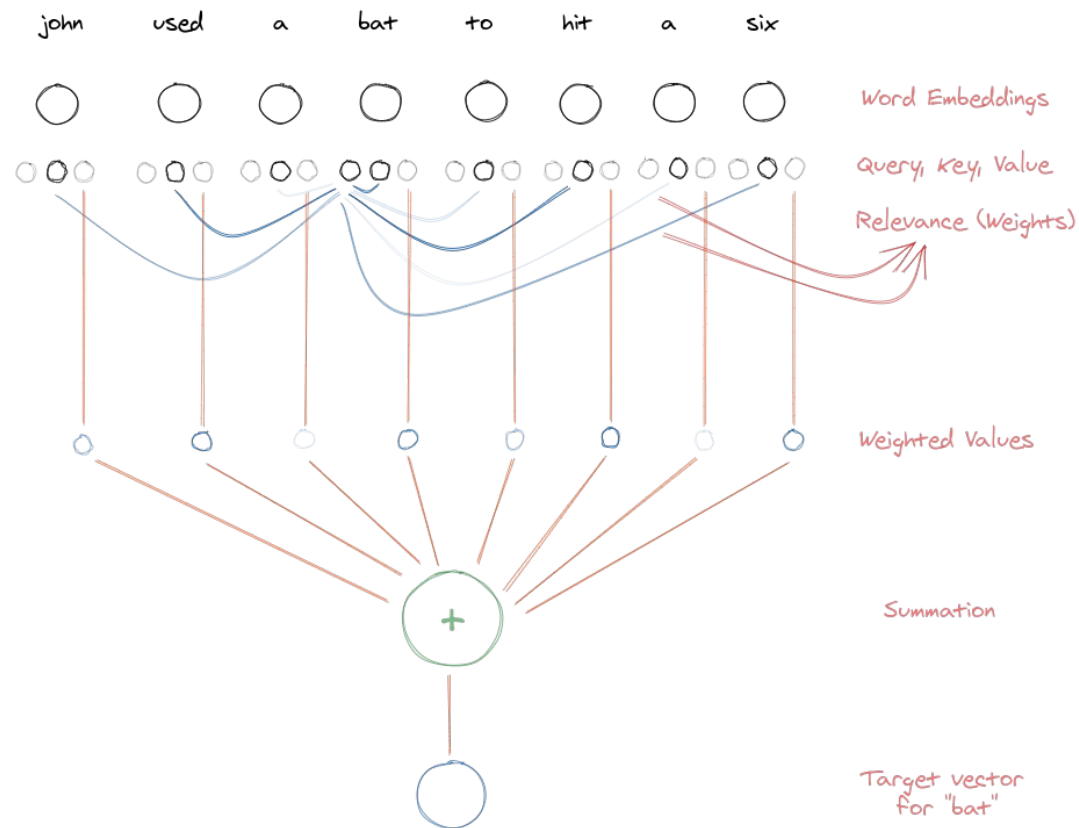


Figure:
Jay Alammar,
[“The Illustrated Transformer”](#)

Self-Attention (or intra-Attention)



Query (Q)
 $g(x) = W^Q x$

Key (K)
 $f(x) = W^K x$

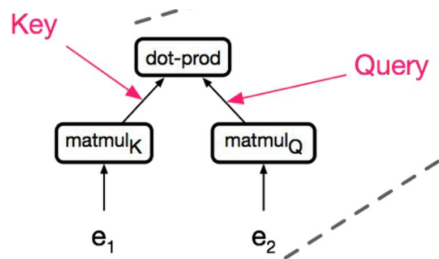
Value (V)
 $h(x) = W^V x$

W^Q , W^K and W^V are **projection layers** shared across all words.

Self-Attention (or intra-Attention)

Which steps are necessary to compute the contextual representation of a word embedding e_2 in a sequences of four words embeddings (e_1, e_2, e_3, e_4) ?

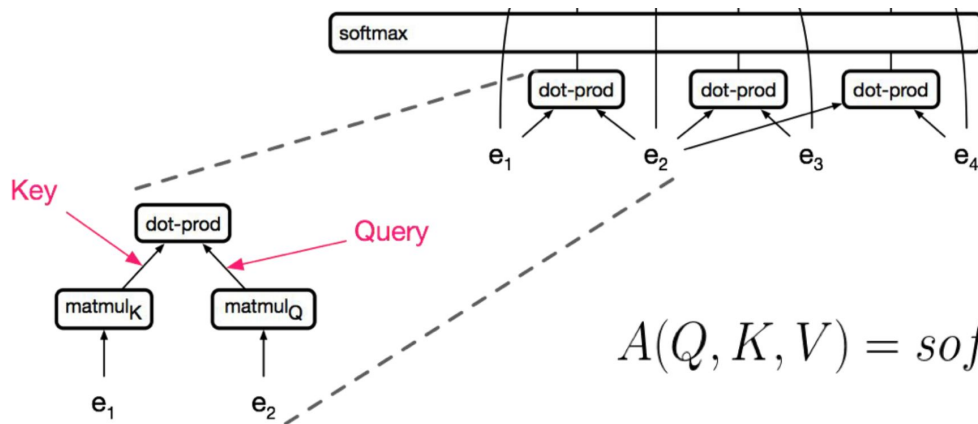
A (scaled) dot-product is computed between each pair of word embeddings (eg. e_1 and e_2)...



Self-Attention (or intra-Attention)

Which steps are necessary to compute the contextual representation of a word embedding e_2 in a sequences of four words embeddings (e_1, e_2, e_3, e_4) ?

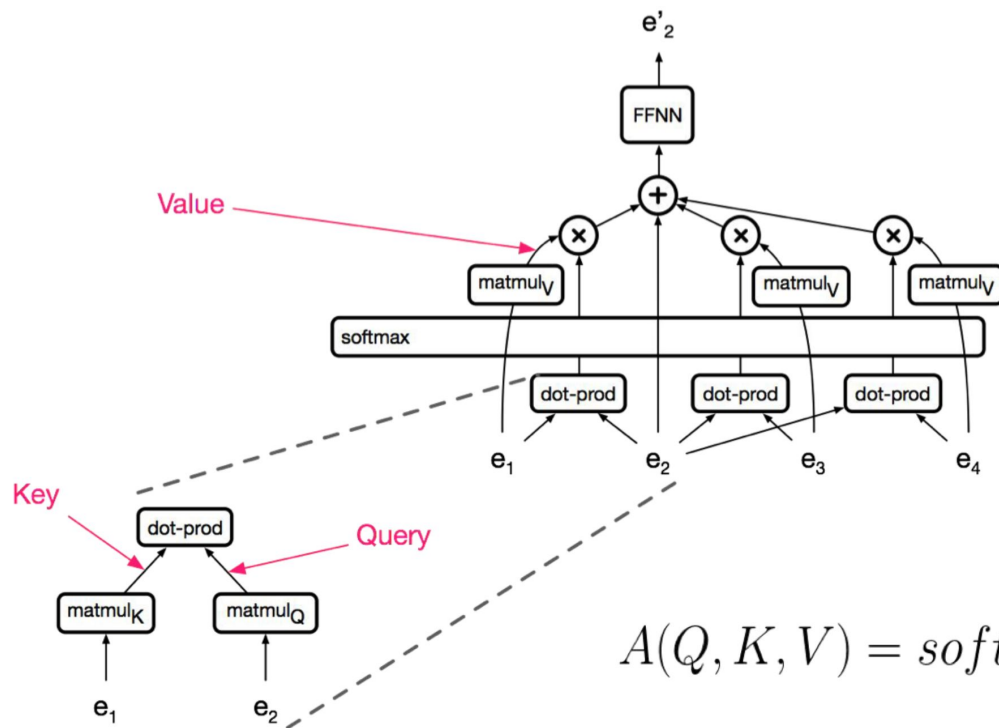
... a softmax layer normalizes the attention scores to obtain the attention distribution...



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

Self-Attention (or intra-Attention)

Which steps are necessary to compute the contextual representation of a word embedding e_2 in a sequences of four words embeddings (e_1, e_2, e_3, e_4) ?



...the same word embeddings are combined to obtain the contextual representation e'_2 .

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

Self-Attention (or intra-Attention)

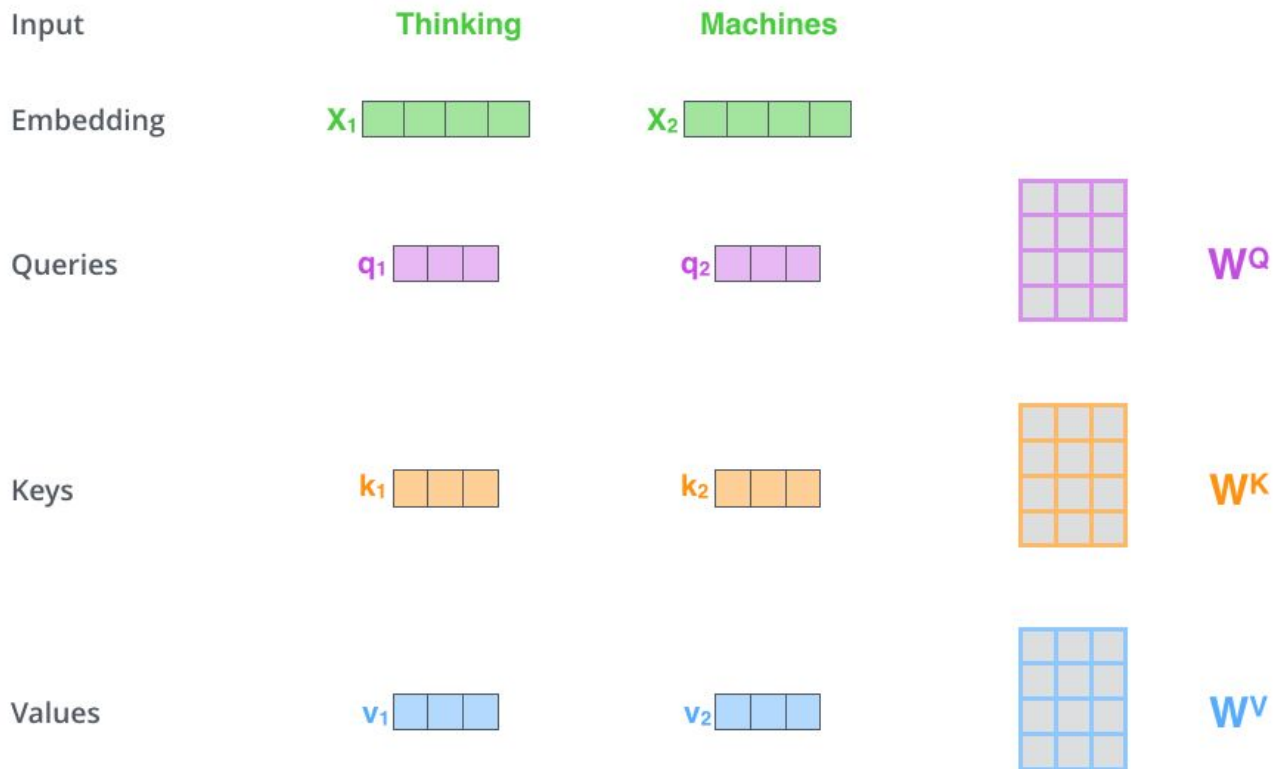
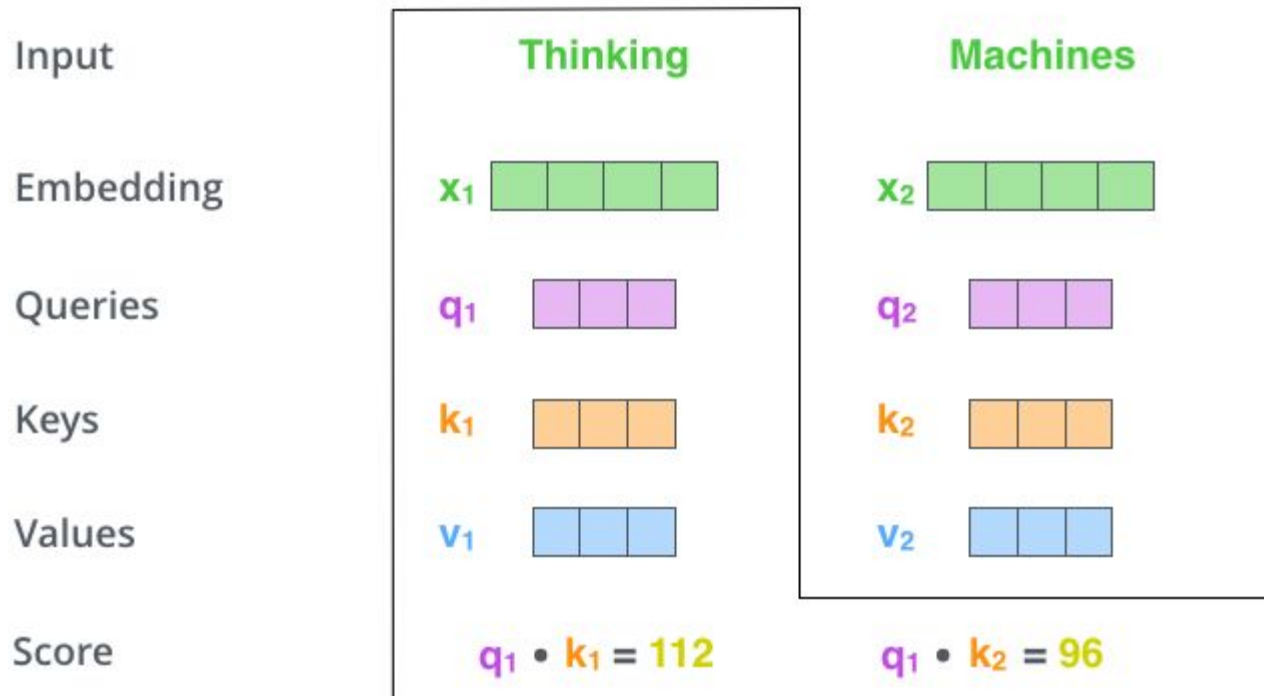


Figure: Jay Alammar, [“The illustrated Transformer”](#) (2018)

Self-Attention (or intra-Attention)



Self-Attention (or intra-Attention)

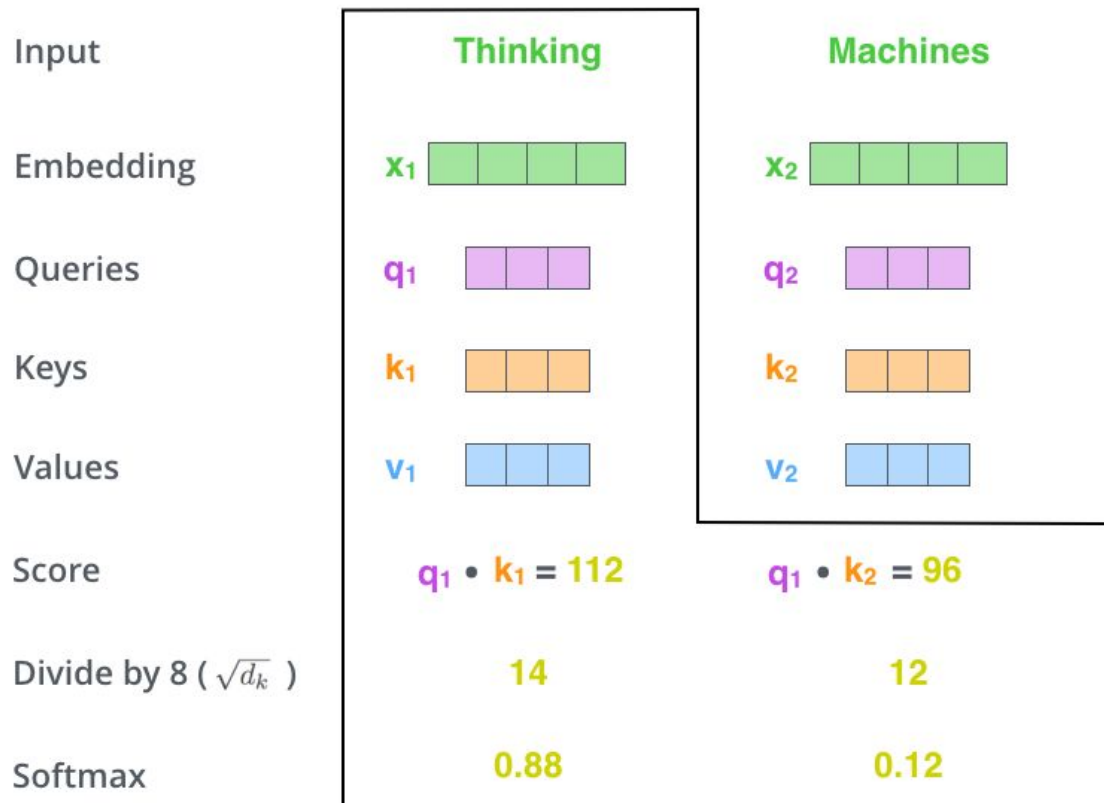


Figure: Jay Alammar, [“The illustrated Transformer”](#) (2018)

Self-Attention (or intra-Attention)

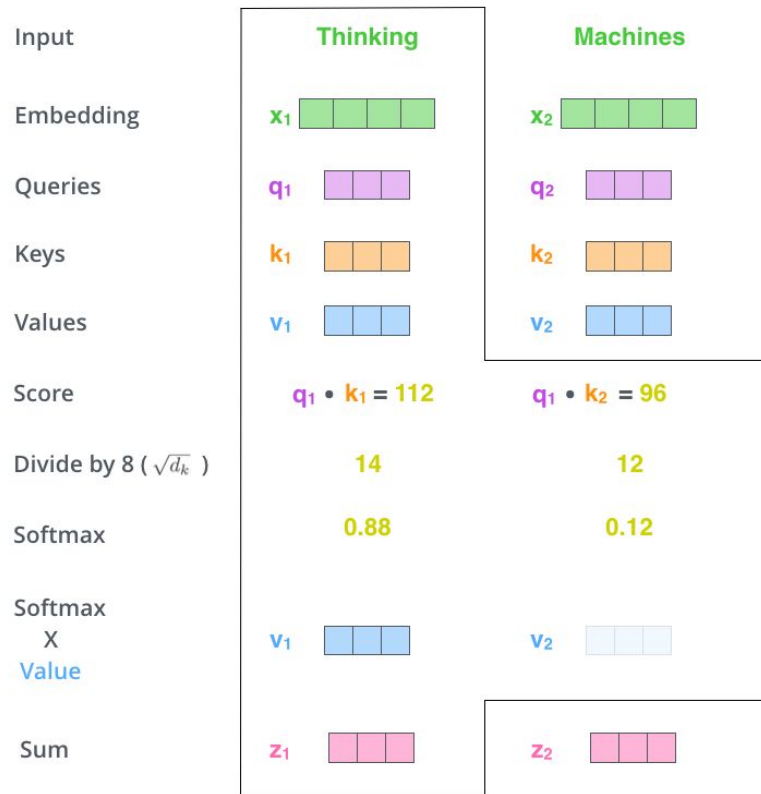


Figure: Jay Alammar, [“The illustrated Transformer”](#) (2018)

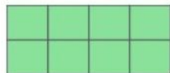
Self-Attention (or intra-Attention)

1) This is our
input sentence

2) We embed
each word

Thinking
Machines

E



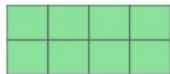
Self-Attention (or intra-Attention)

1) This is our
input sentence

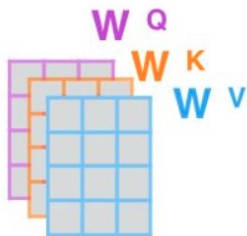
Thinking
Machines

2) We embed
each word

E



3) We multiply X with
weight matrices



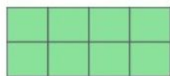
Self-Attention (or intra-Attention)

1) This is our input sentence

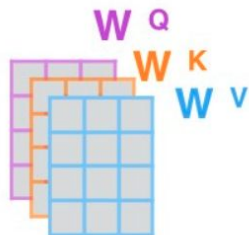
Thinking
Machines

2) We embed each word

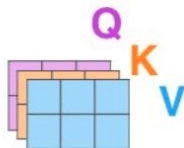
E



3) We multiply X with weight matrices



4) Calculate attention using the resulting Q/K/V matrices



$$\text{softmax}\left(\frac{\begin{matrix} \text{Q} \\ \text{4x4 grid} \end{matrix} \times \begin{matrix} \text{K}^T \\ \text{4x4 grid} \end{matrix}}{\sqrt{d_k}}\right) \begin{matrix} \text{V} \\ \text{4x4 grid} \end{matrix} = \begin{matrix} \text{E}' \\ \text{2x4 grid} \end{matrix}$$

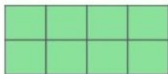
Self-Attention (or intra-Attention)

1) This is our
input sentence

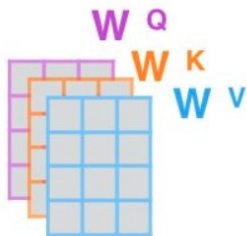
Thinking
Machines

2) We embed
each word

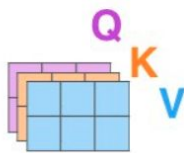
E



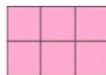
3) We multiply X with
weight matrices



4) Calculate attention
using the resulting
 $Q/K/V$ matrices



E'



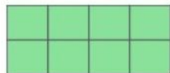
Self-Attention (or intra-Attention)

1) This is our input sentence

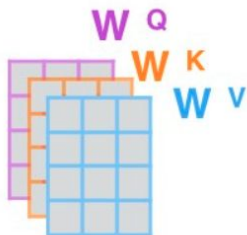
Thinking
Machines

2) We embed each word

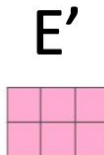
E



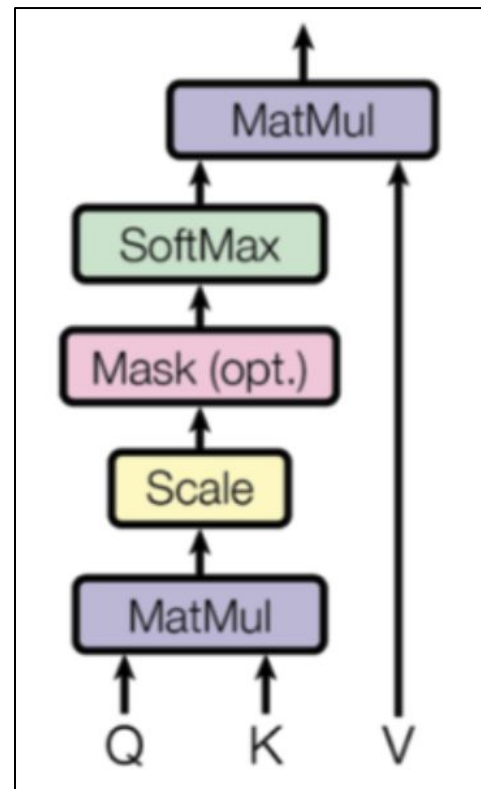
3) We multiply X with weight matrices



4) Calculate attention using the resulting $Q/K/V$ matrices



Scaled dot-product attention



Study case: Self-Attention in images

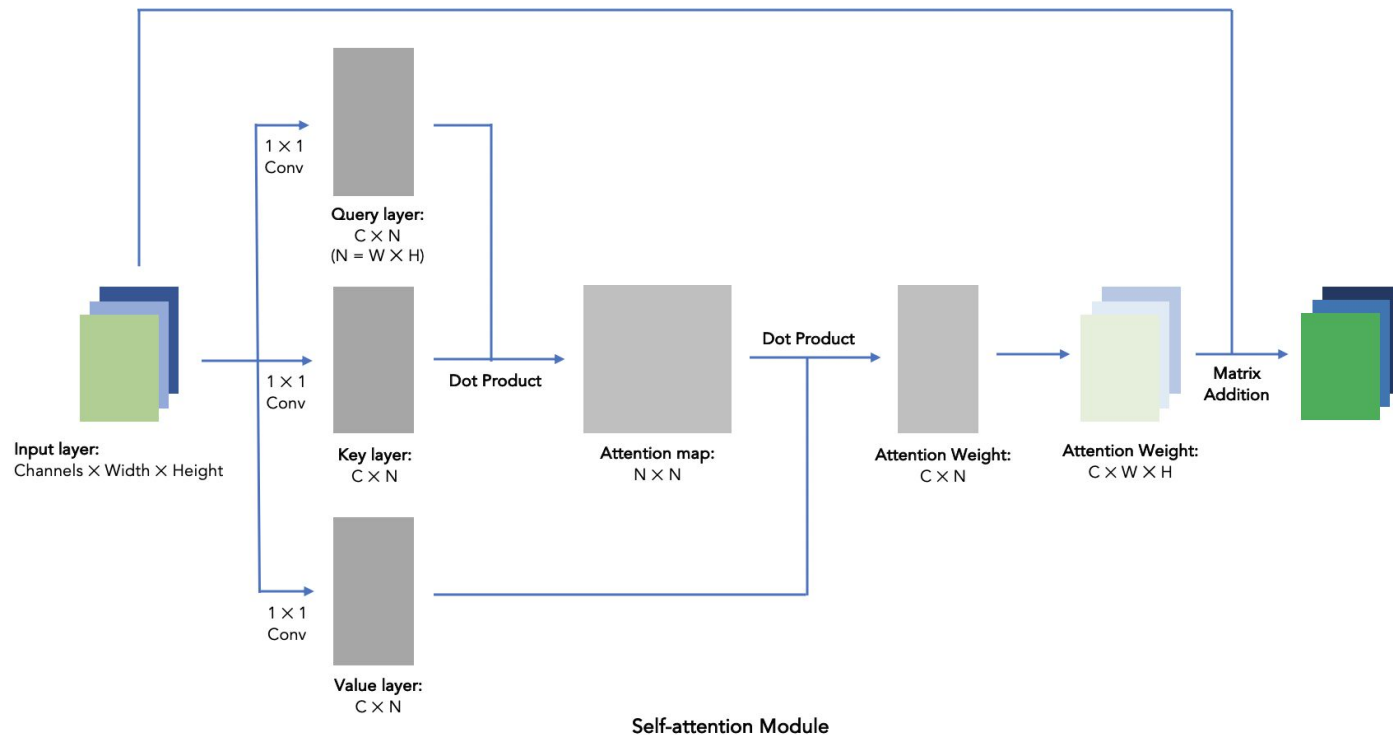


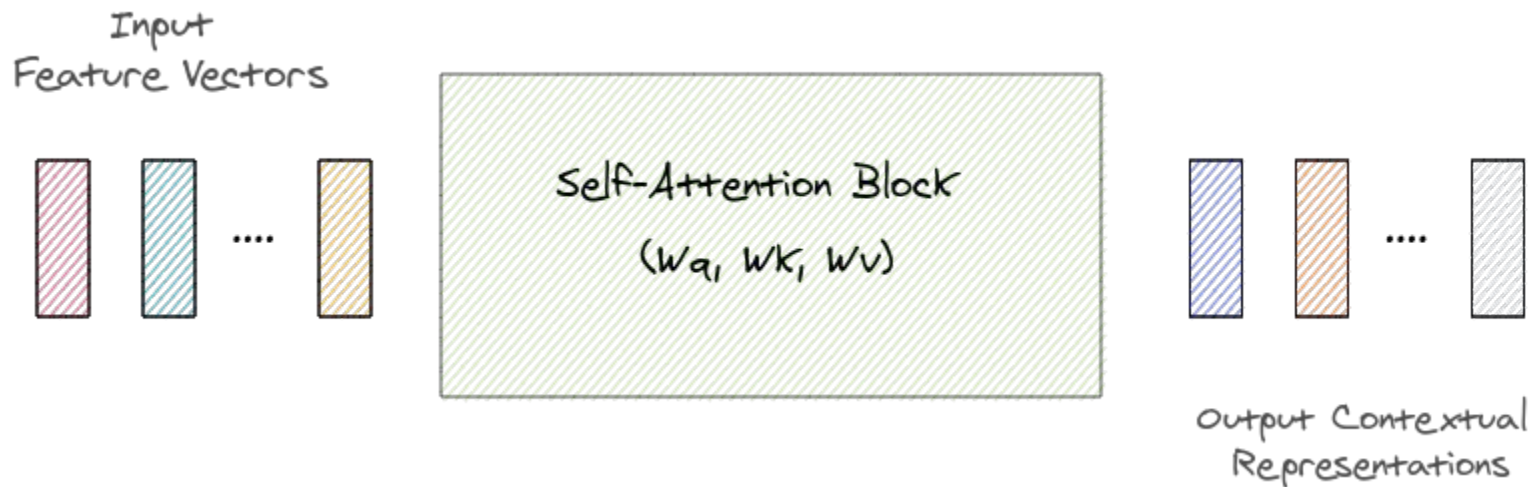
Figure:
[Frank Xu](#)

Outline

1. Motivation
2. Self-attention
3. **Multi-head Self-Attention (MHSA)**

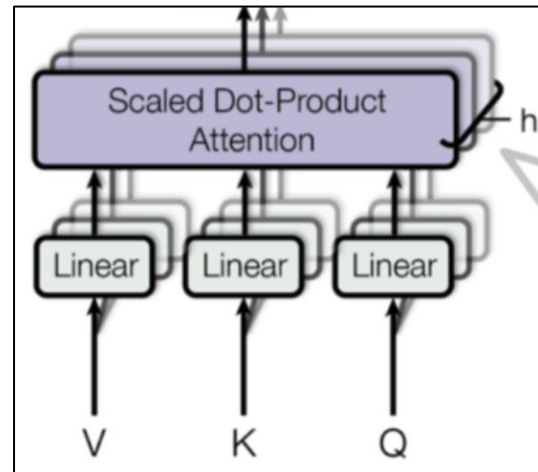
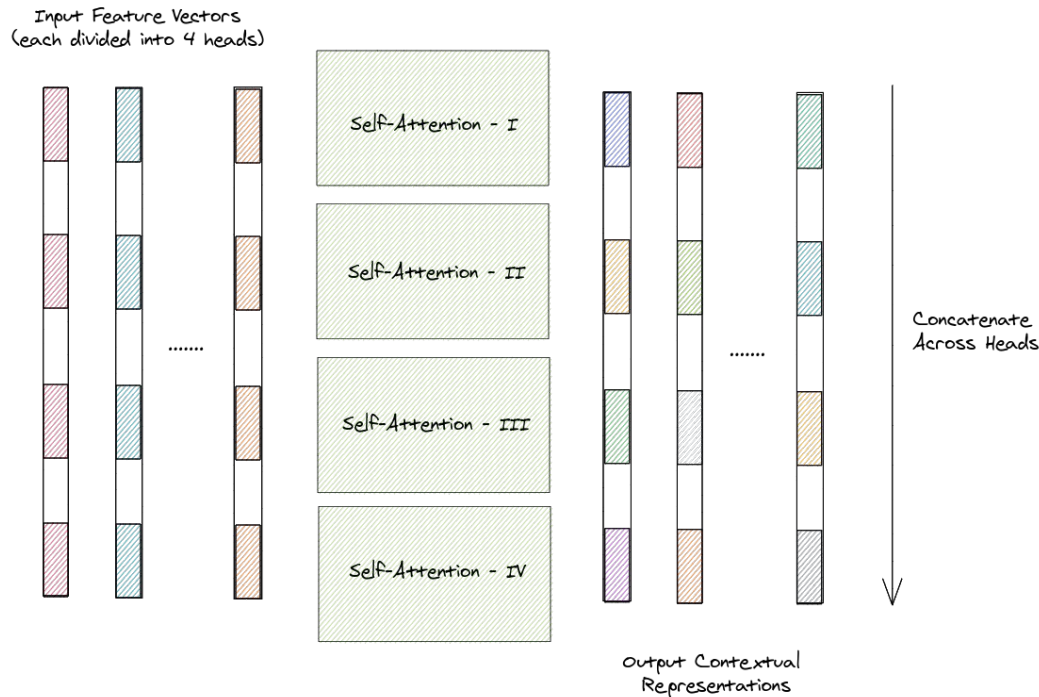
Multi-Head Self-Attention (MHSA)

In vanilla self-attention, a single set of projection matrices W^Q , W^K , W^V is used.



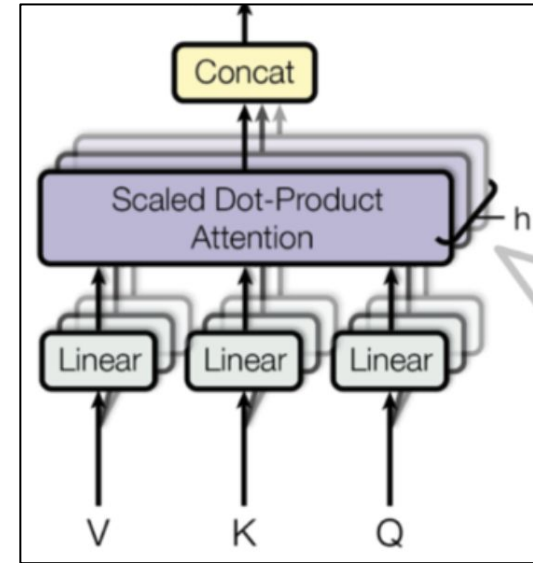
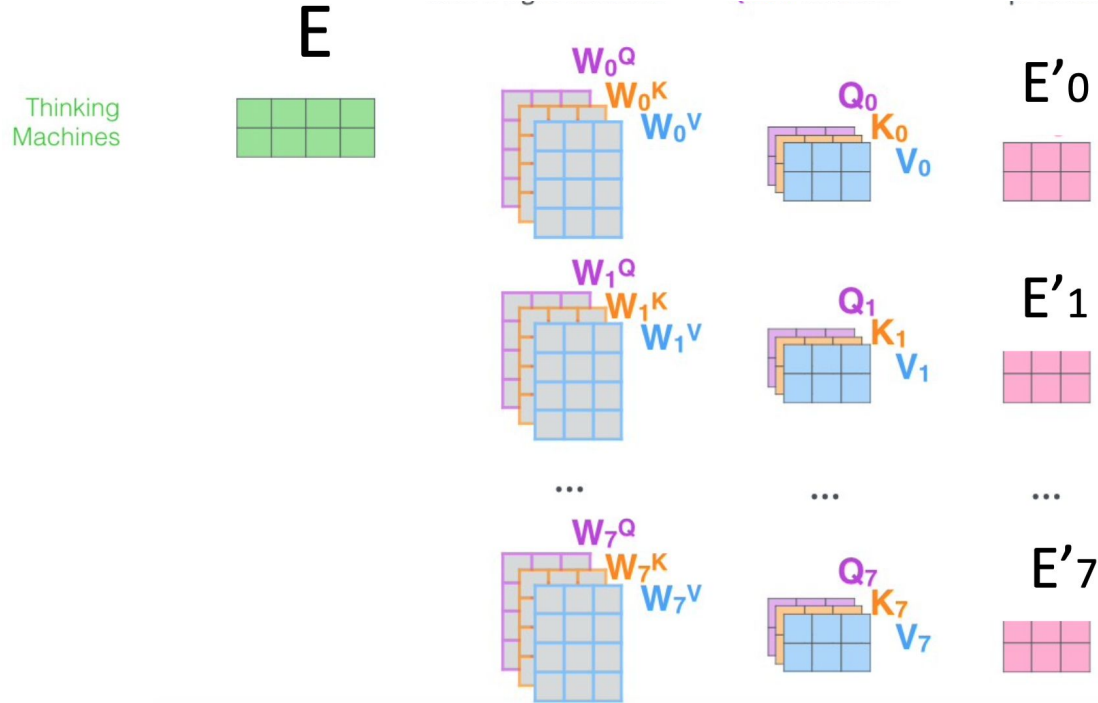
Multi-Head Self-Attention (MHSA)

In multi-head self-attention, multiple sets of projection matrices are used, and can provide different contextual representations for the same input token.



Multi-Head Self-Attention (MHSA)

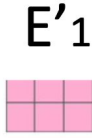
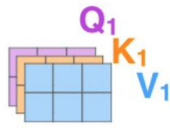
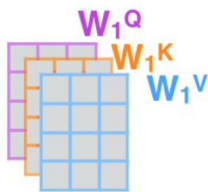
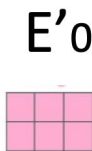
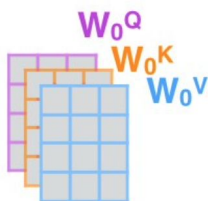
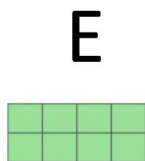
The multi-head self-attended E'_i matrixes are concatenated:



Multi-Head Self-Attention (MHSA)

A fully connected layer on top combines everything in a new E' .

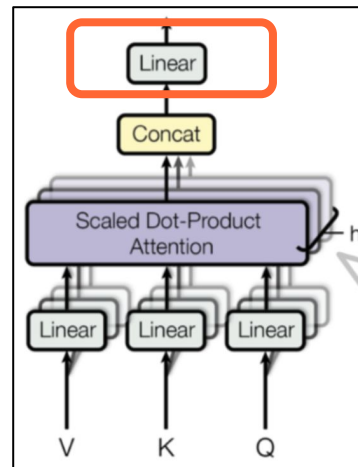
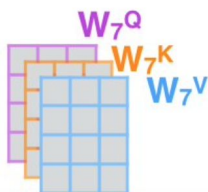
Thinking
Machines



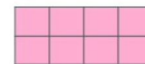
...

...

...

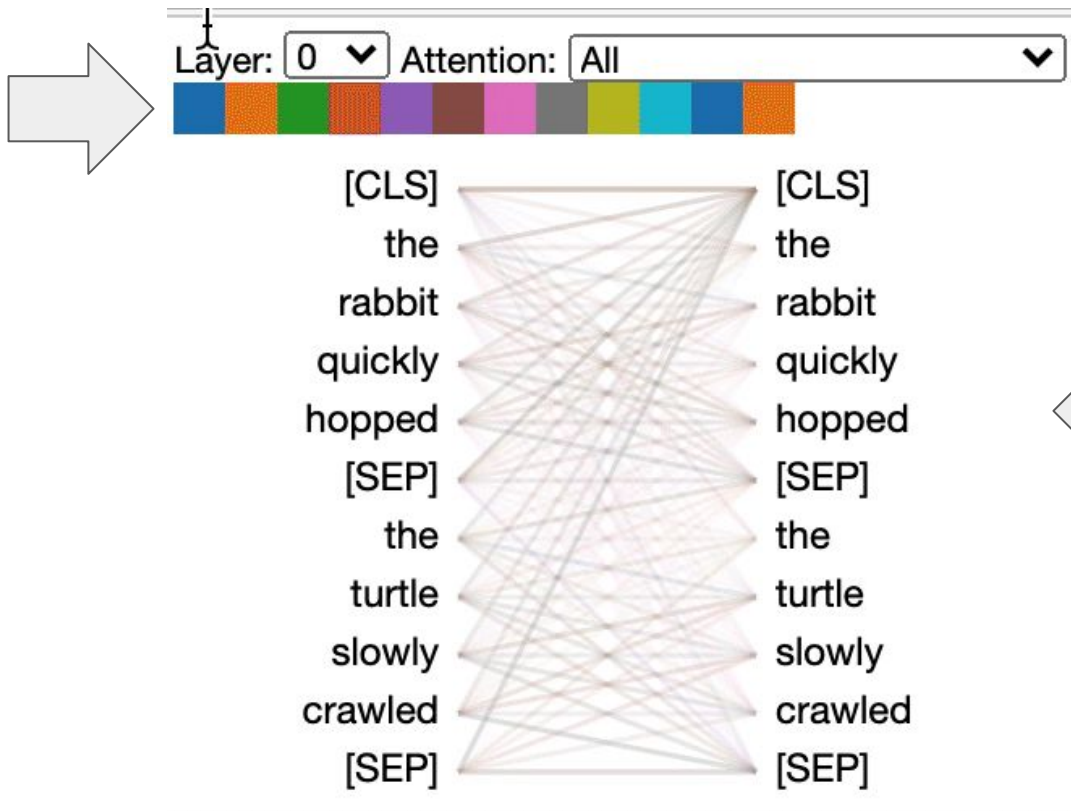


E'



Multi-head Self-Attention: Visualization

Each colour corresponds to a head.



Self-Attention and Convolutional Layers

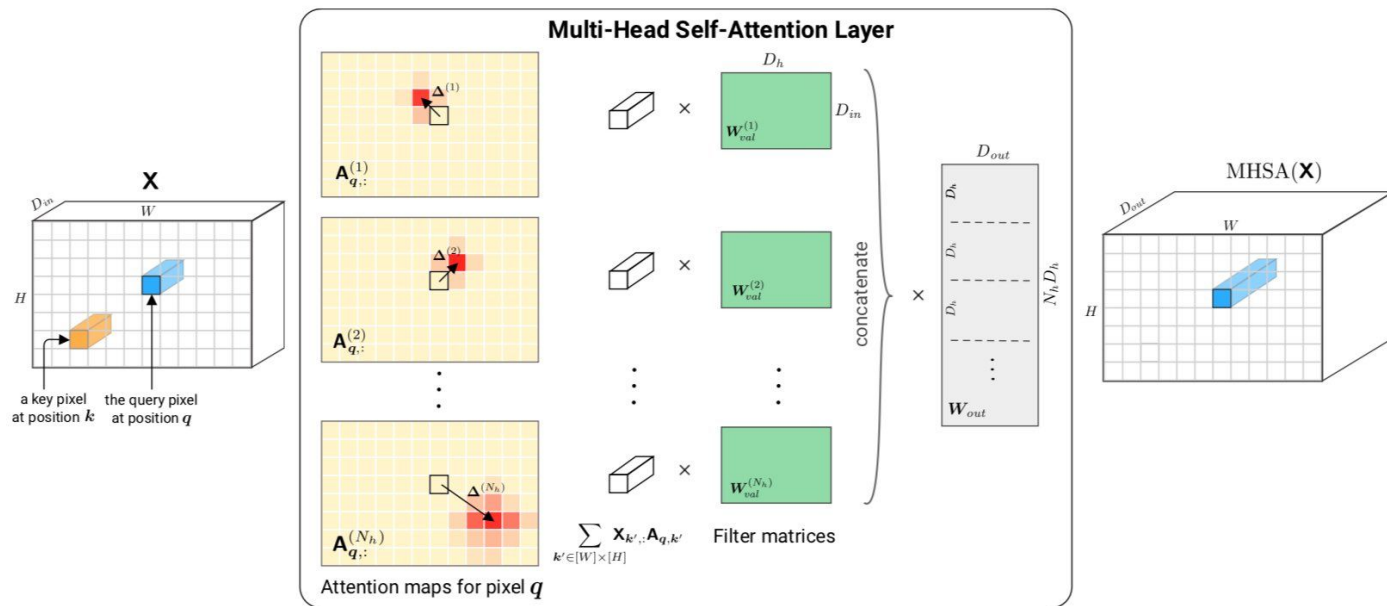


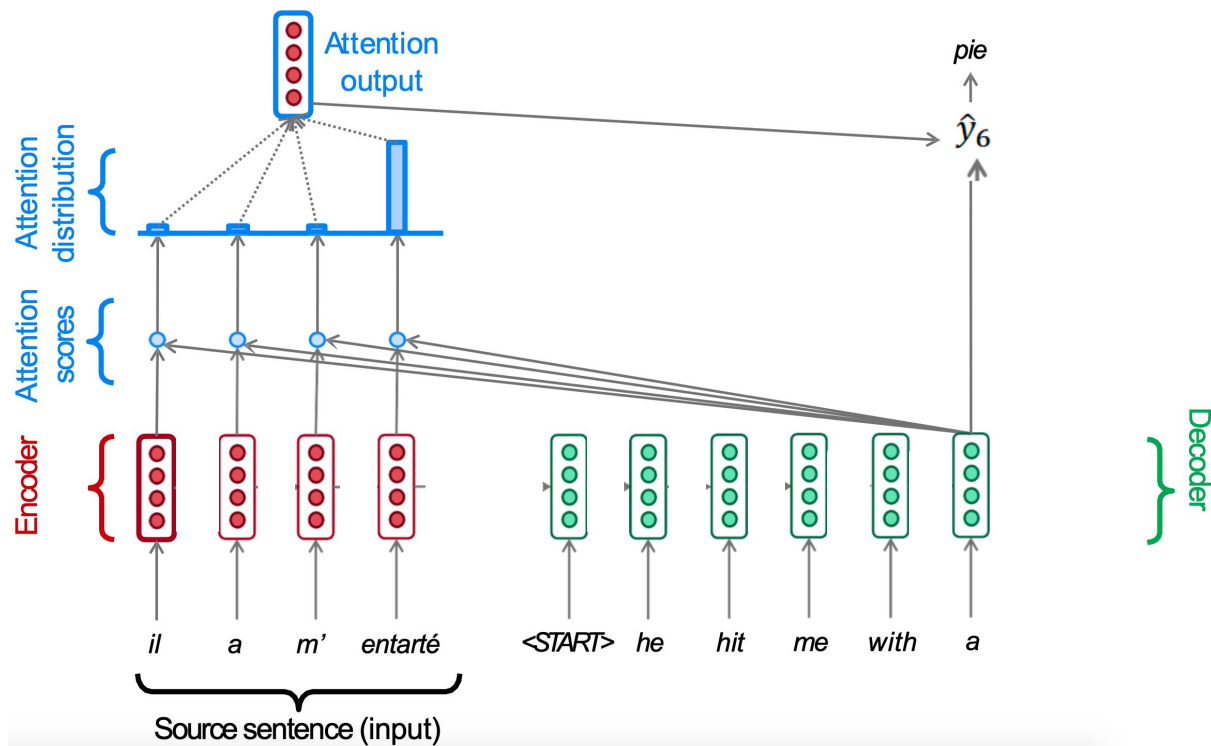
Figure 1: Illustration of a Multi-Head Self-Attention layer applied to a tensor image \mathbf{X} . Each head h attends pixel values around shift $\Delta^{(h)}$ and learn a filter matrix $\mathbf{W}_{val}^{(h)}$. We show attention maps computed for a query pixel at position q .

Outline

1. Motivation
2. Self-attention
3. Multi-head Attention
- 4. Positional Encoding**

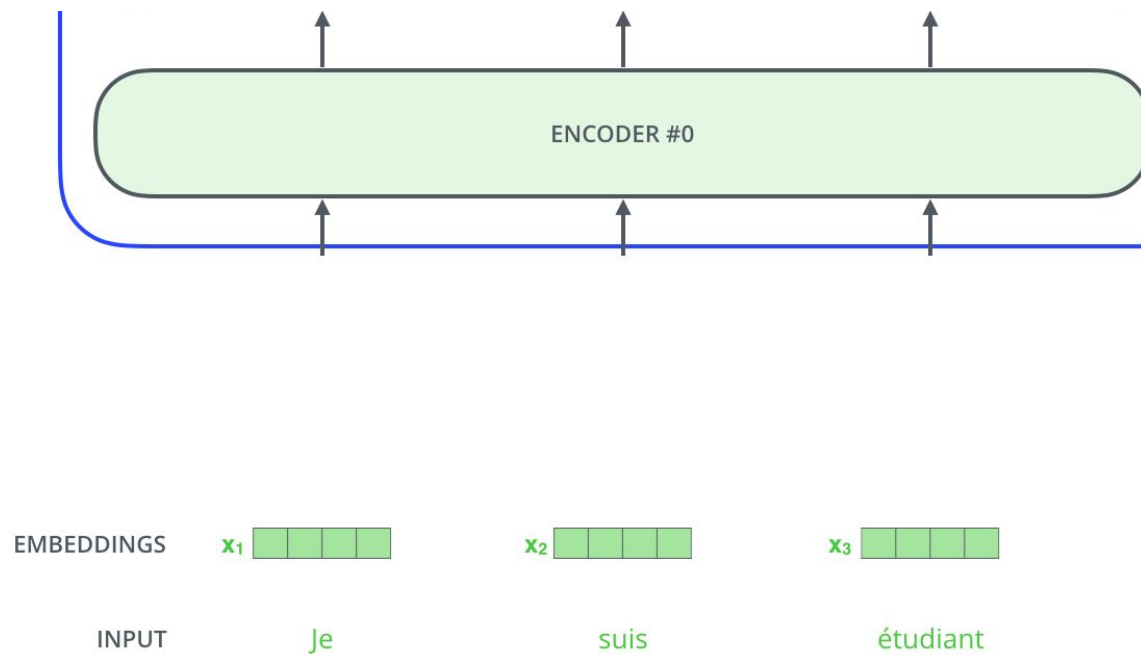
Positional Encoding

Given that the attention mechanism allows accessing all input (and output) tokens, we no longer need a memory through recurrent layers.



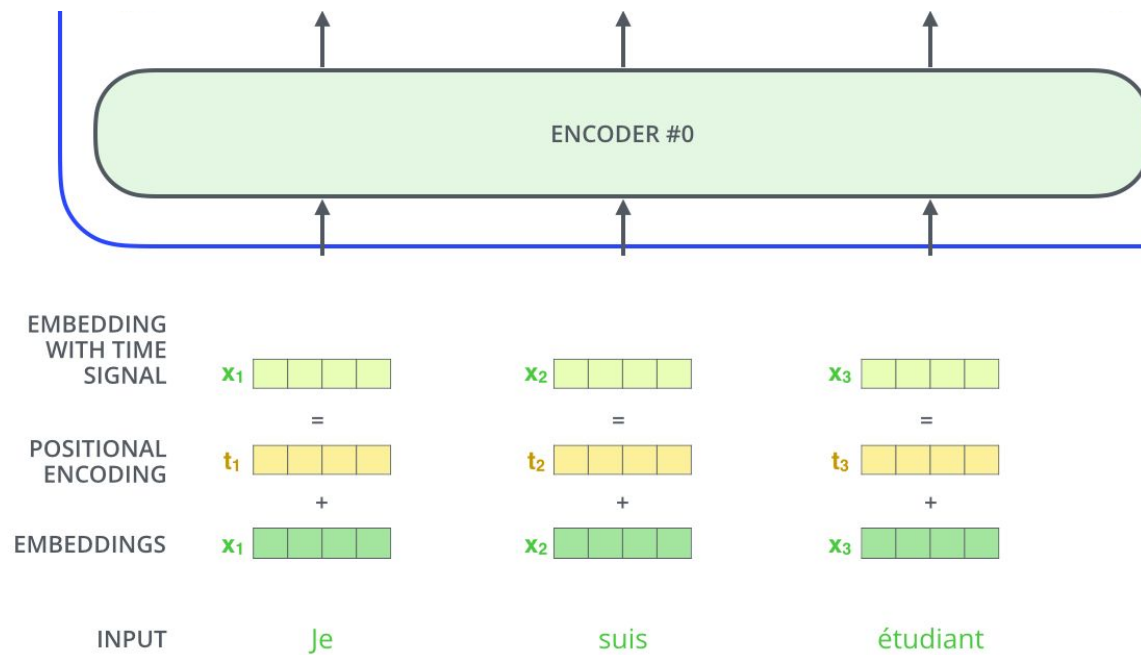
Positional Encoding

Where is the relative relation in the sequence encoded ?



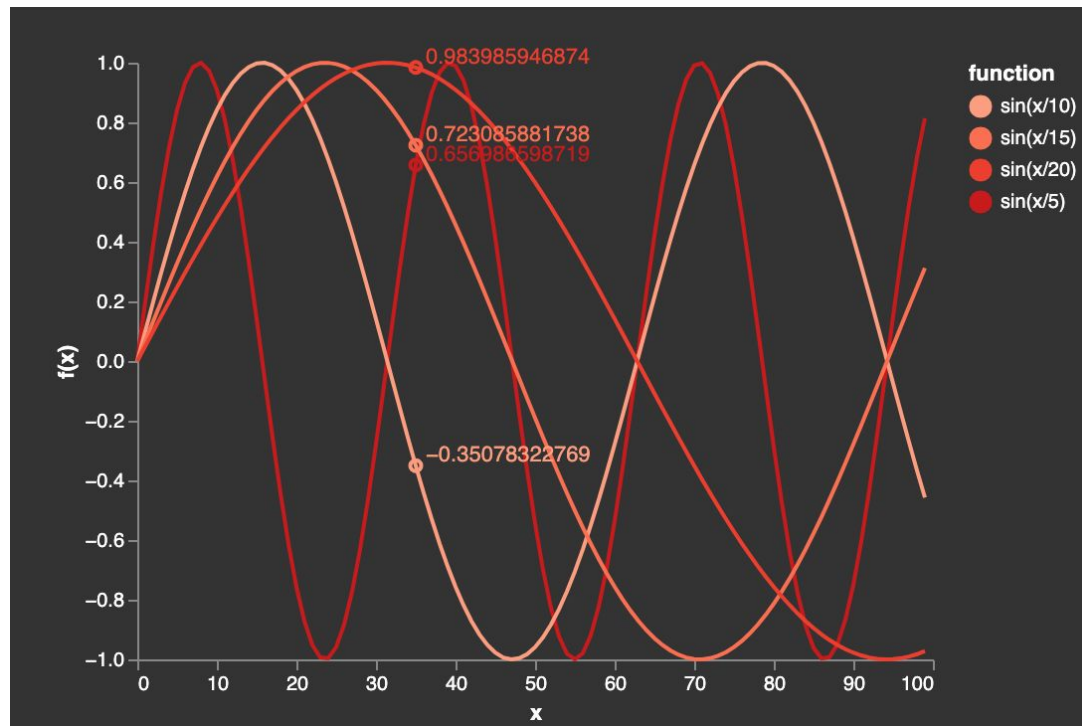
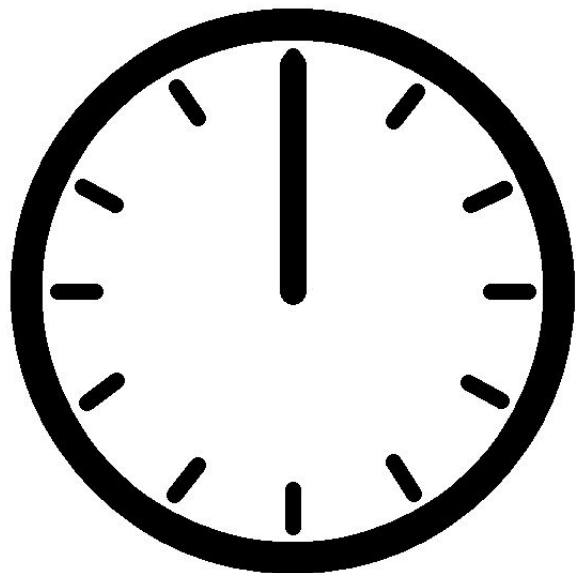
Positional Encoding

Where is the relative relation in the sequence encoded ?

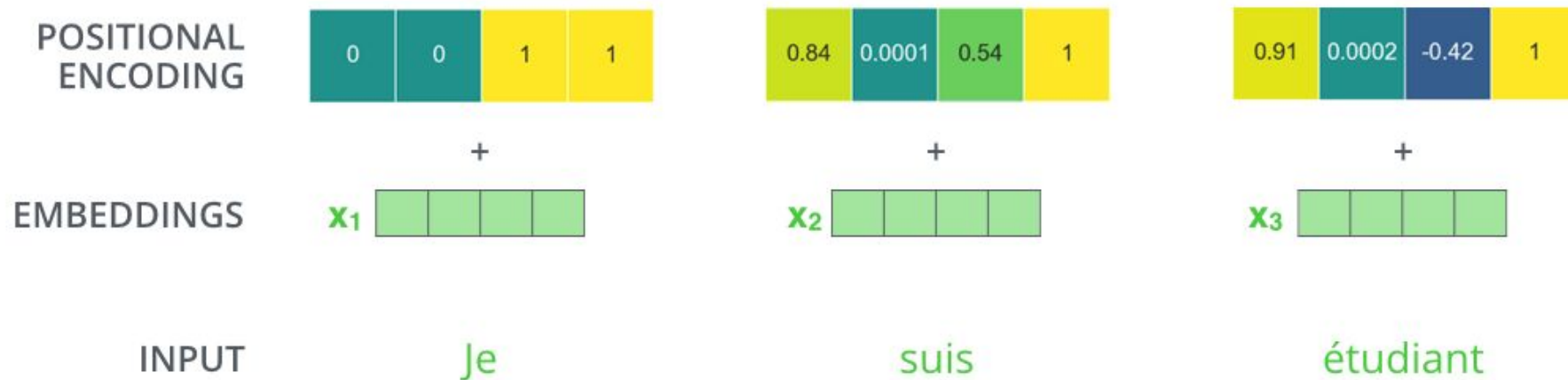


Positional Encoding

Sinusoidal functions are typically used to provide positional encodings.



Positional Encoding

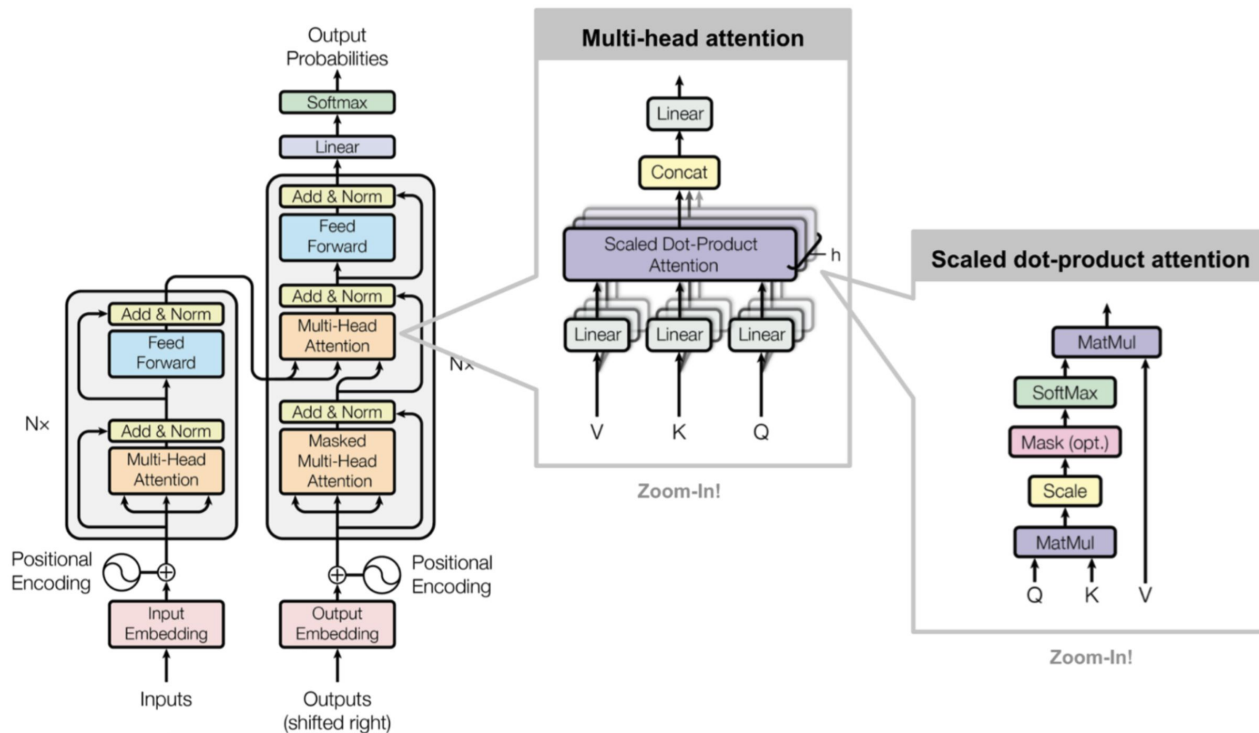


Outline

1. Motivation
2. Self-attention
3. Multi-head Attention
4. Positional Encoding
- 5. The Transformer**

The Transformer

The Transformer was a revolutionary architecture in machine translation.

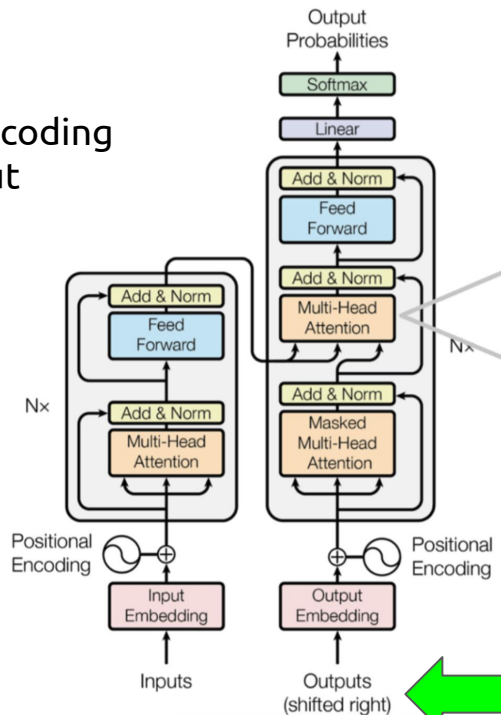
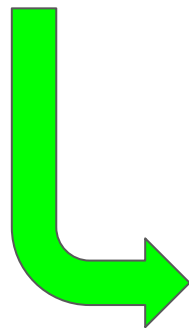


#Transformer Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I.. [Attention is all you need](#). NeurIPS 2017.

The Transformer

The Transformer removed the recurrent layer and adopted an auto-regressive approach (at test).

Positional Encoding
over the input
sequence.



Positional Encoding over the output
sequence.



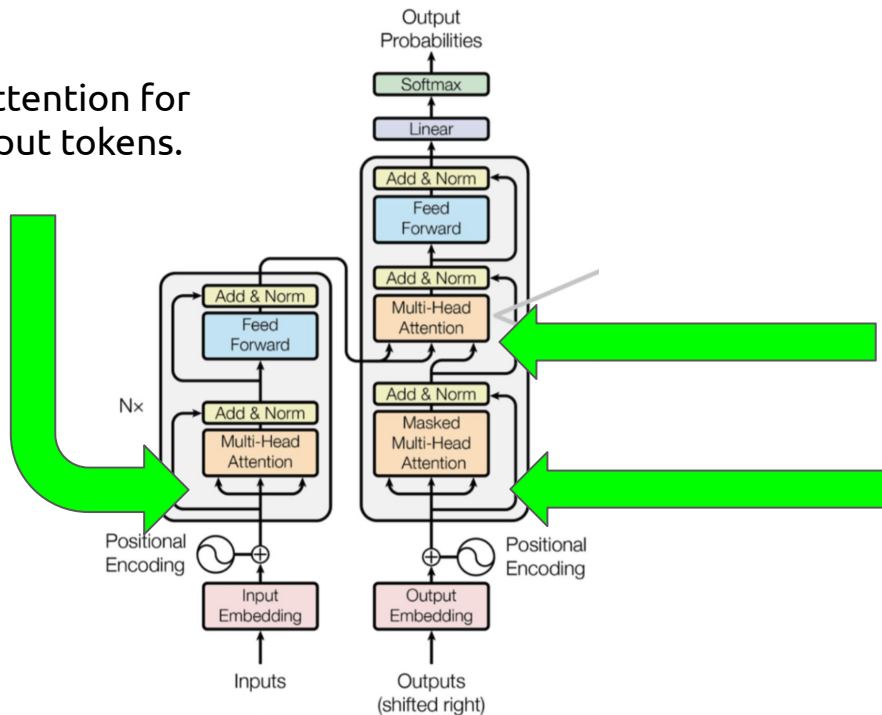
Auto-regressive.



The Transformer

The Transformer was a revolutionary architecture that removed the recurrent layers to process sequences.

Self-attention for the input tokens.

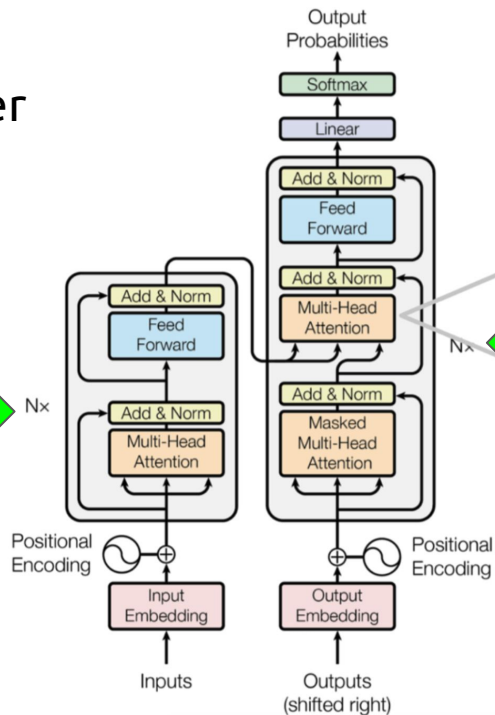


Cross-Attention (or inter-attention)
between input and output tokens

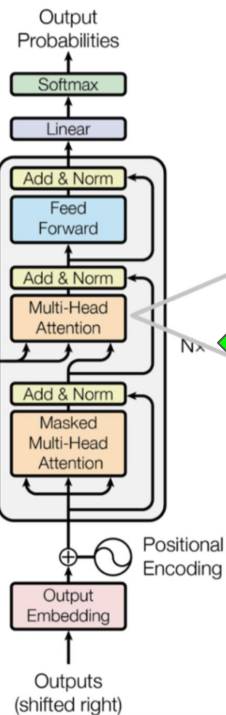
Self-attention for the output tokens.

The Transformer: Layers

N encoder
layers

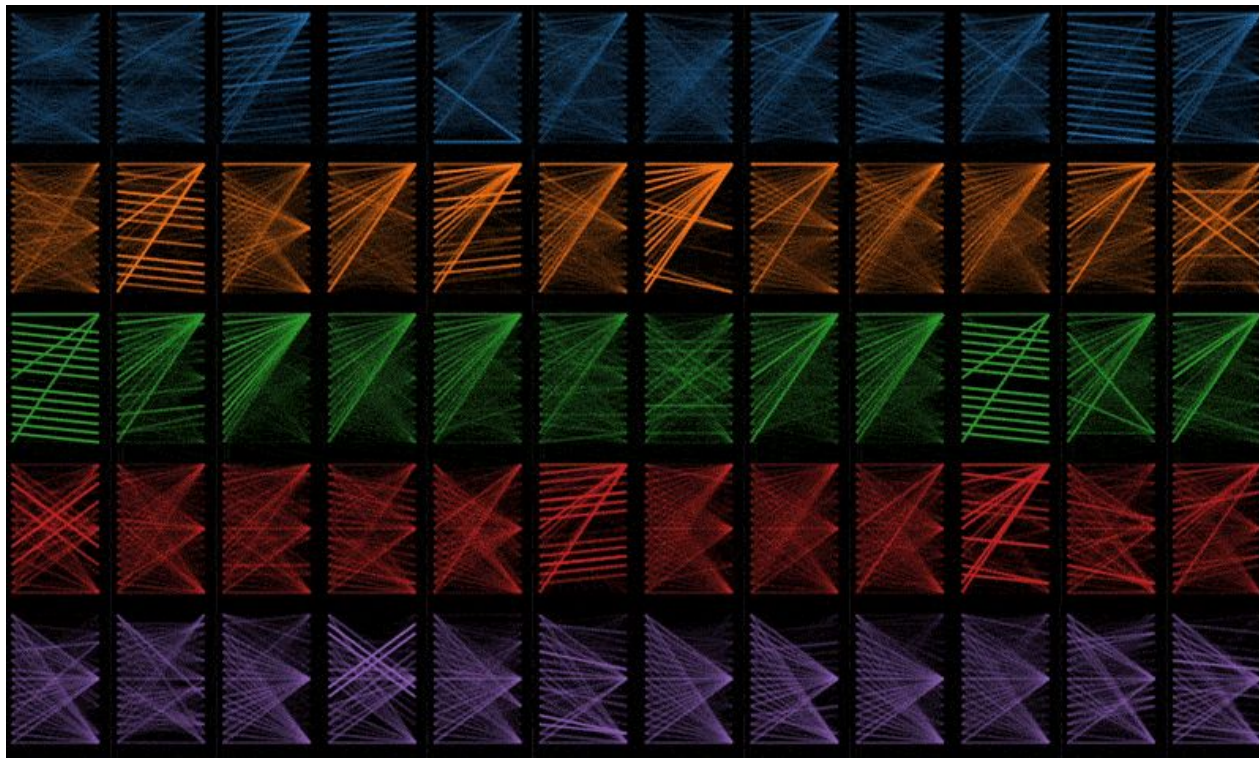


N decoder
layers

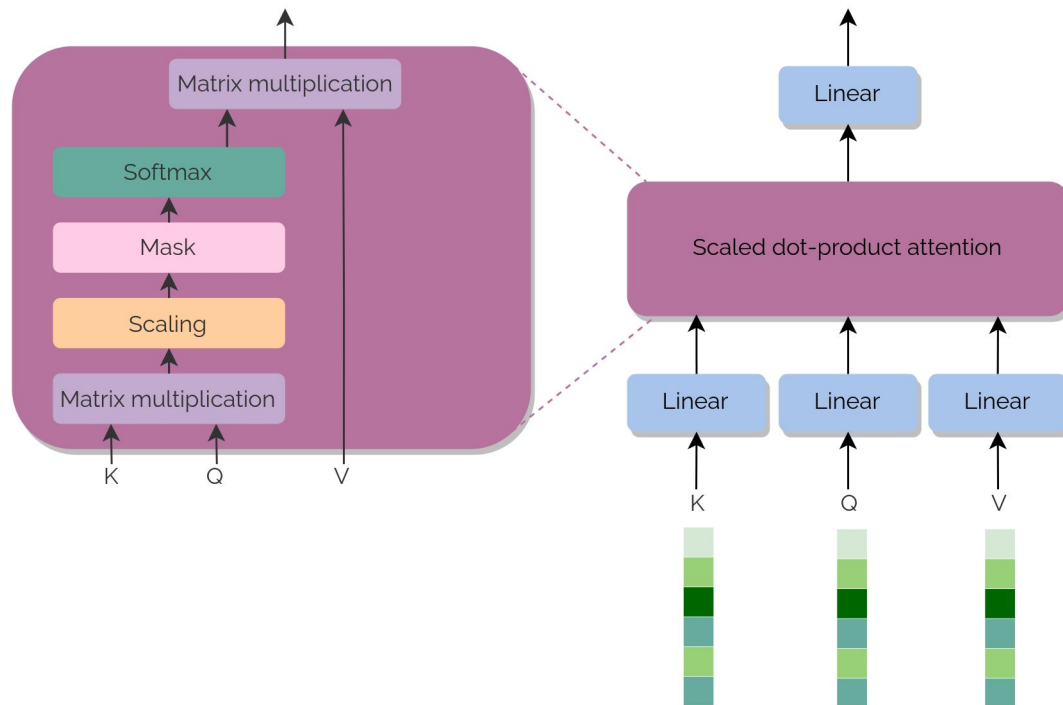


The Transformer: Layers

A birds-eye view of attention across all of the model's layers and heads



The Transformer: Visualization



Are Transformers for Language only ? NO !!

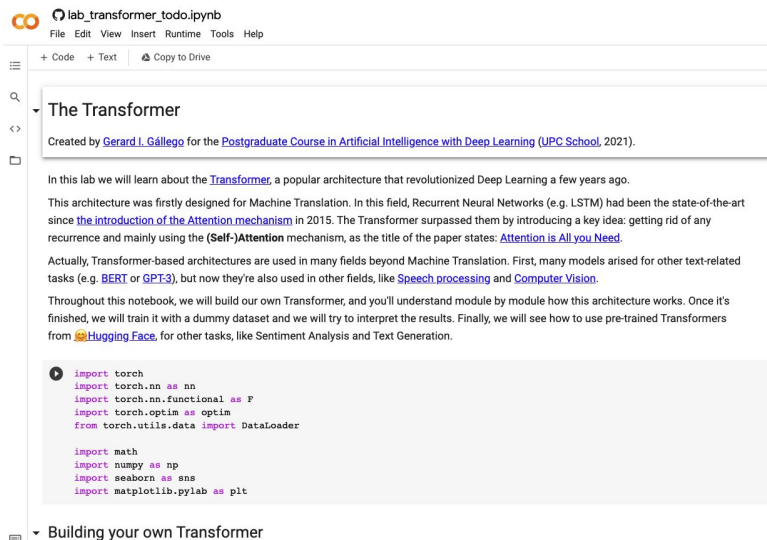


#ViT Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. ["An image is worth 16x16 words: Transformers for image recognition at scale."](#) ICLR 2021. [[blog](#)] [[code](#)]

Outline

1. Motivation
2. Self-attention
3. Multi-head Attention
4. Positional Encoding
5. The Transformer

(extra) PyTorch Lab on Google Colab



lab_transformer_todo.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text Copy to Drive

The Transformer

Created by [Gerard I. Gallego](#) for the [Postgraduate Course in Artificial Intelligence with Deep Learning \(UPC School, 2021\)](#).

In this lab we will learn about the [Transformer](#), a popular architecture that revolutionized Deep Learning a few years ago.

This architecture was firstly designed for Machine Translation. In this field, Recurrent Neural Networks (e.g. LSTM) had been the state-of-the-art since [the introduction of the Attention mechanism](#) in 2015. The Transformer surpassed them by introducing a key idea: getting rid of any recurrence and mainly using the [\(Self-\)Attention](#) mechanism, as the title of the paper states: [Attention is All you Need](#).

Actually, Transformer-based architectures are used in many fields beyond Machine Translation. First, many models arised for other text-related tasks (e.g. [BERT](#) or [GPT-3](#)), but now they're also used in other fields, like [Speech processing](#) and [Computer Vision](#).

Throughout this notebook, we will build our own Transformer, and you'll understand module by module how this architecture works. Once it's finished, we will train it with a dummy dataset and we will try to interpret the results. Finally, we will see how to use pre-trained Transformers from [Hugging Face](#), for other tasks, like Sentiment Analysis and Text Generation.

```
import torch
import torch.nn as nn
import torch.nn.functional as F
import torch.optim as optim
from torch.utils.data import DataLoader

import math
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Building your own Transformer

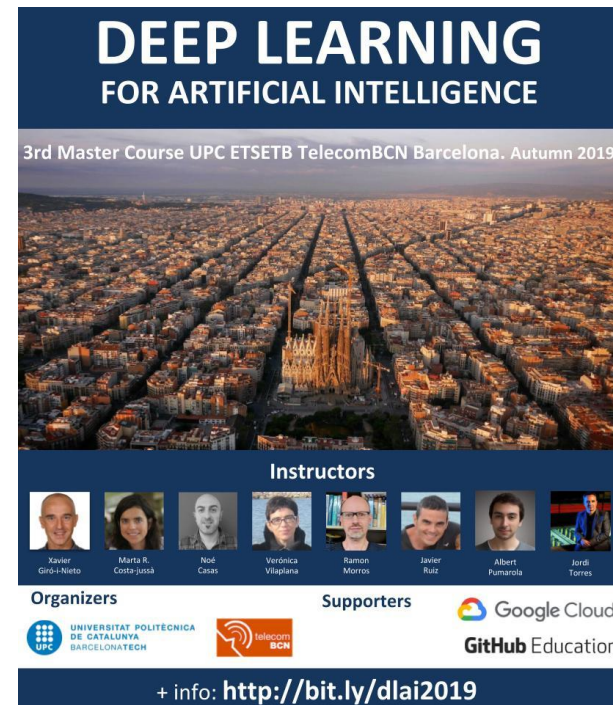


Gerard Gallego

gerard.ion.gallego@upc.edu


Student PhD

Universitat Politècnica de Catalunya
Technical University of Catalonia











DEEP LEARNING FOR ARTIFICIAL INTELLIGENCE



3rd Master Course UPC ETSETB TelecomBCN Barcelona. Autumn 2019.





Instructors

							
Xavier Giró-i-Nieto	Marta R. Costa-jussà	Noé Casas	Verónica Vilaplana	Ramon Morros	Javier Ruiz	Albert Pumarola	Jordi Torres

Organizers

 UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH	
--	---

Supporters

 Google Cloud	 GitHub Education
--	--

+ info: <http://bit.ly/dlai2019>

- [DL resources from UPC Telecoms:](#)
- [Lectures](#) (with Slides & Videos)
- [Labs](#)

Software

- [Transformers](#) in HuggingFace.
- [GPT-Neo](#) by EleutherAI
 - Similar results to GPT-3, but smaller and open source.
- Andrej Karpathy, [minGPT](#) (2020).



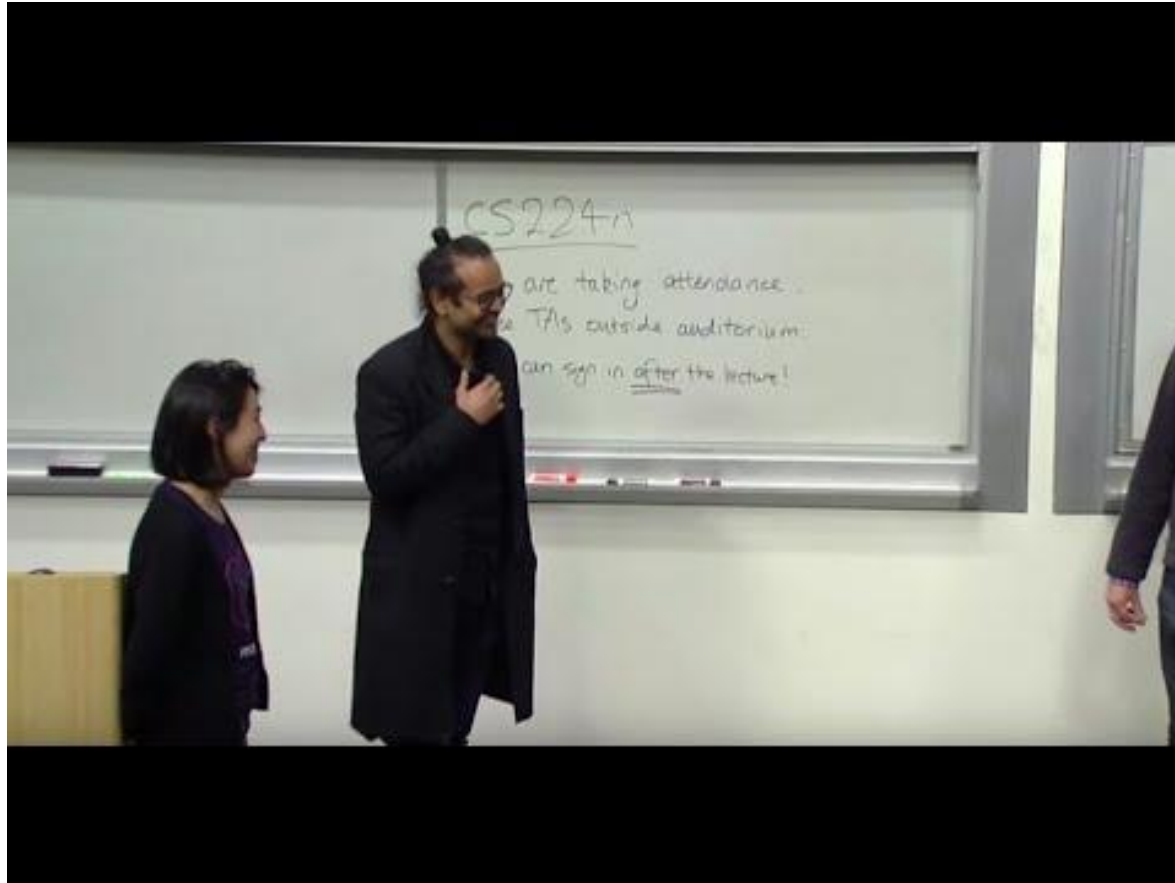
available GPT implementations



minGPT



Learn more



Ashihs Vaswani, Stanford CS224N 2019.

Learn more

- Tutorials
 - Sebastian Ruder, [Deep Learning for NLP Best Practices # Attention](#) (2017).
 - Chris Olah, Shan Carter, ["Attention and Augmented Recurrent Neural Networks"](#). distill.pub 2016.
- Demos
 - [François Fleuret](#) (EPFL)
- Twitter threads
 - [Christian Wolf](#) (INSA Lyon)
- Scientific publications
 - Katharopoulos, A., Vyas, A., Pappas, N., & Fleuret, F. (2020). [Transformers are rnns: Fast autoregressive transformers with linear attention](#). ICML 2020.
 - Siddhant M. Jayakumar, Wojciech M. Czarnecki, Jacob Menick, Jonathan Schwarz, Jack Rae, Simon Osindero, Yee Whye Teh, Tim Harley, Razvan Pascanu, ["Multiplicative Interactions and Where to Find Them"](#). ICLR 2020. [\[tweet\]](#)
 - Self-attention in language
 - Cheng, J., Dong, L., & Lapata, M. (2016). [Long short-term memory-networks for machine reading](#). arXiv preprint arXiv:1601.06733.
 - Self-attention in images
 - Parmar, N., Vaswani, A., Uszkoreit, J., Kaiser, Ł., Shazeer, N., Ku, A., & Tran, D. (2018). [Image transformer](#). ICML 2018.
 - Wang, Xiaolong, Ross Girshick, Abhinav Gupta, and Kaiming He. ["Non-local neural networks."](#) In CVPR 2018.
 - **#SAGAN** Zhang, Han, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. ["Self-attention generative adversarial networks."](#) ICML 2019. [\[video\]](#)

Questions ?

Undergradese

What undergrads ask vs. what they're REALLY asking

"Is it going to be an open book exam?"

Translation: "I don't have to actually memorize anything, do I?"

"Hmm, what do you mean by that?"

Translation: "What's the answer so we can all go home."

"Are you going to have office hours today?"

Translation: "Can I do my homework in your office?"

"Can i get an extension?"

Translation: "Can you re-arrange your life around mine?"

"Is this going to be on the test?"

Translation: "Tell us what's going to be on the test."

"Is grading going to be curved?"

Translation: "Can I do a mediocre job and still get an A?"

JORGE CHAM © 2008

