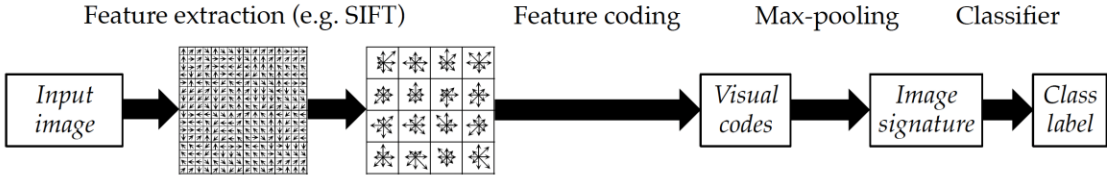
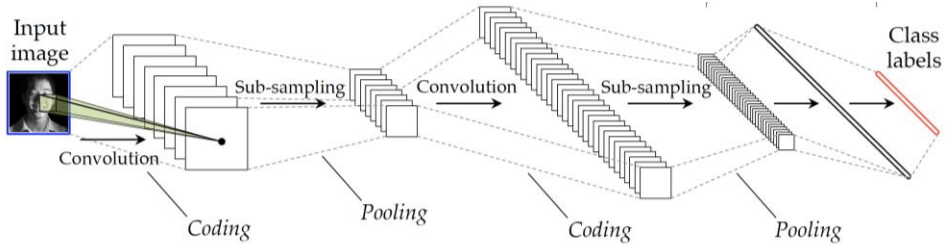
Image classification: where we are and what is missing?

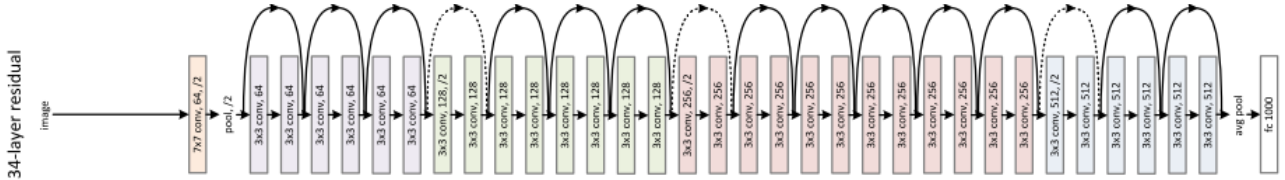
The 2000s: *BoWs + SVM*



The 2010s: Very *Large* ConvNets



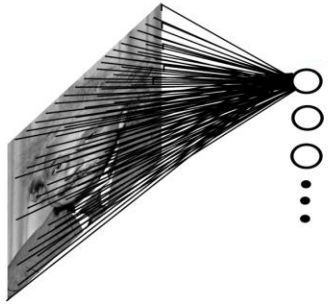
2020: The star: **ResNet**



Next step?

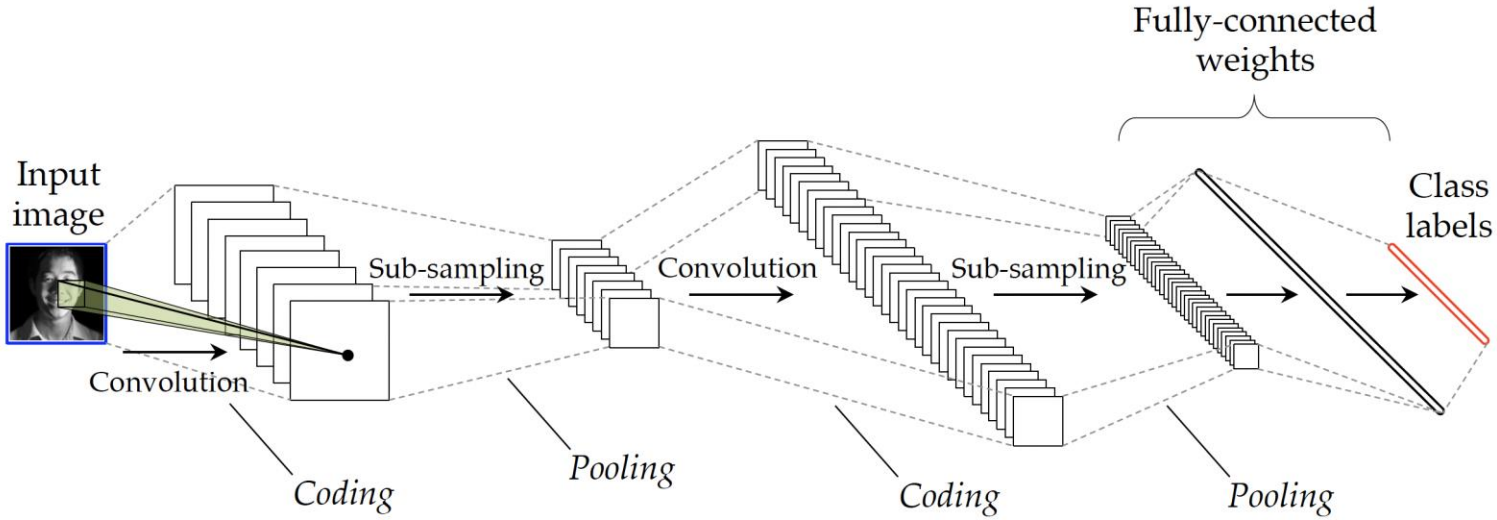
What is missing?

Attention process in ConvNets



In ConvNets, what information is shared between pixels (or features) in one block? => *2D spatial locality (typically 3x3) => attention is done locally*

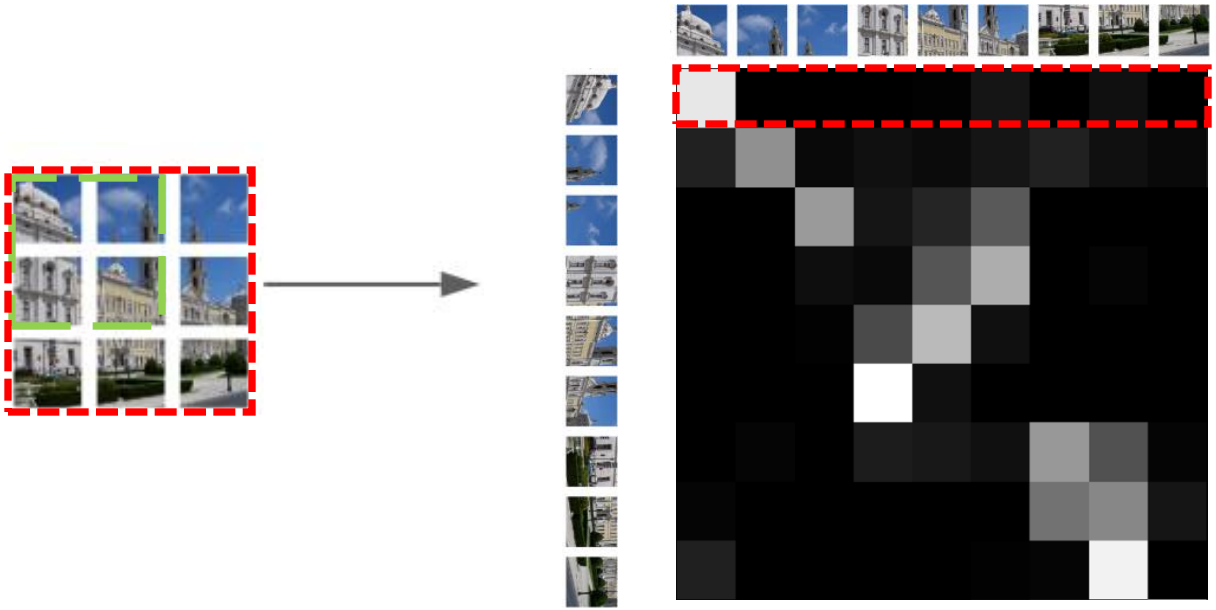
Rq: less local after many layers



Global (Self) attention

How to build a deep architecture with ~~local~~ global attention inside?
Meaning that one patch may interact with all others!

=> Different than convNet!



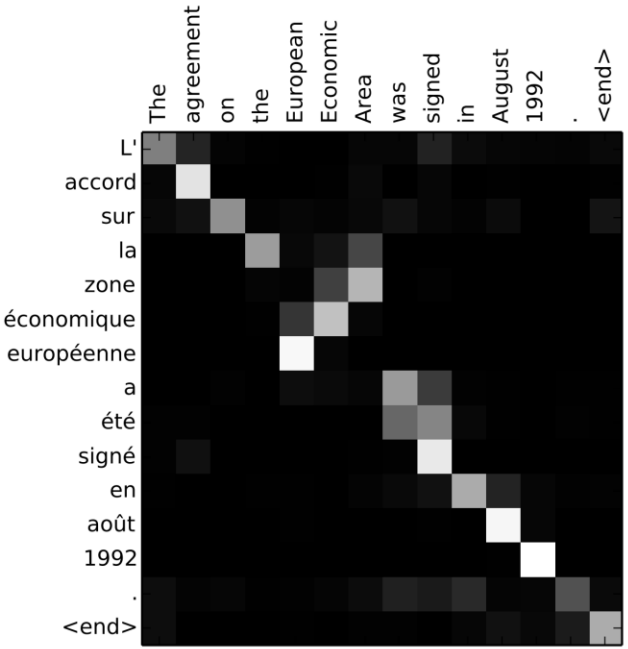
Vision Transformers

1. **NLP: Attention is all you need**
2. Transformer for Image Classification

Let’s see what they do in Natural Language Processing (NLP):

Attention between words in Machine translation process:

1. Computing of weights
2. Use them to compute new features

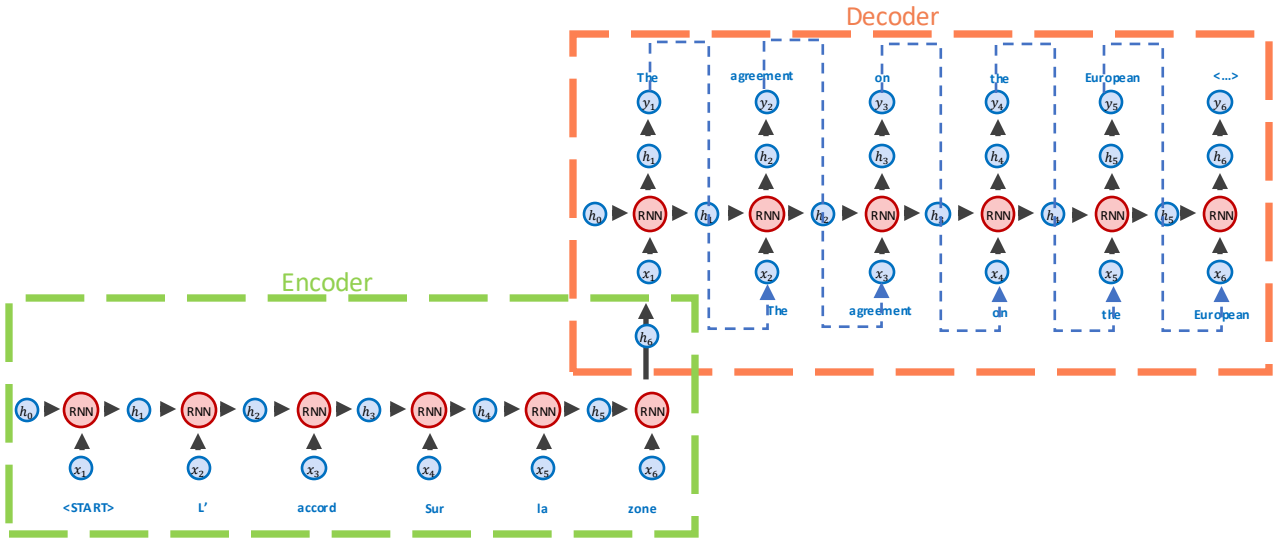


Attention process in NLP

Basic language translation models: Encoder/Decoder

Ex.: Seq2Seq -- RNNs2RNNs

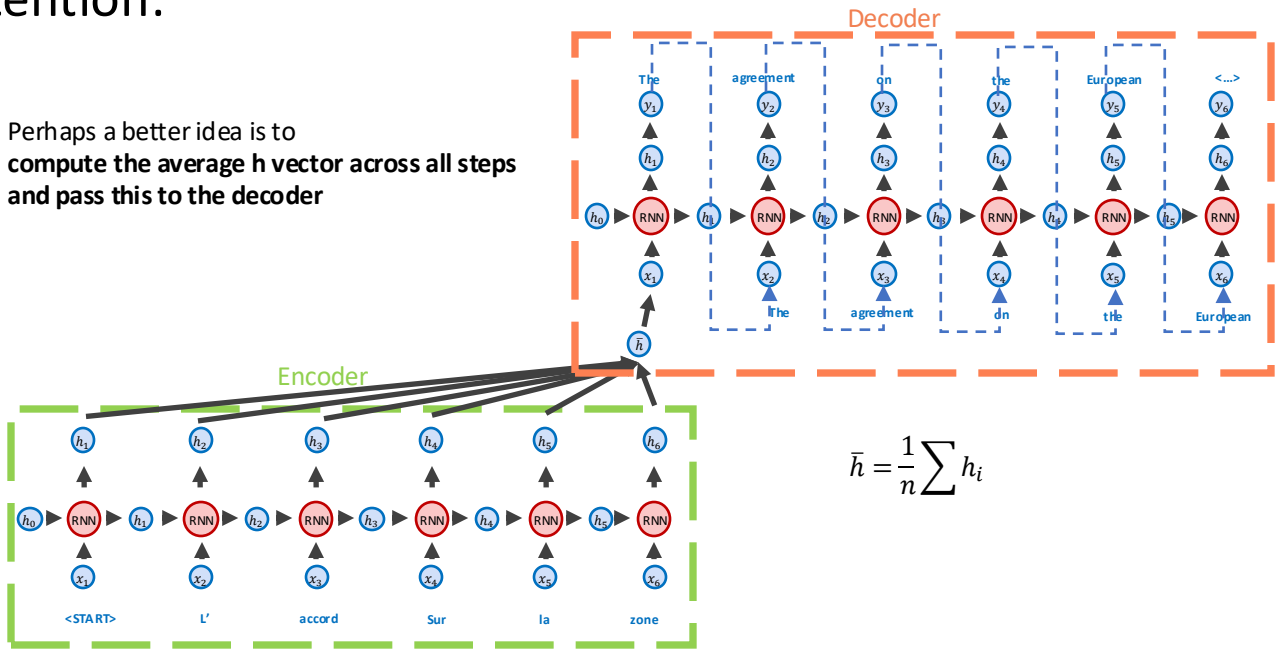
Cross-attention for language translation in at the end of Encoder



Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

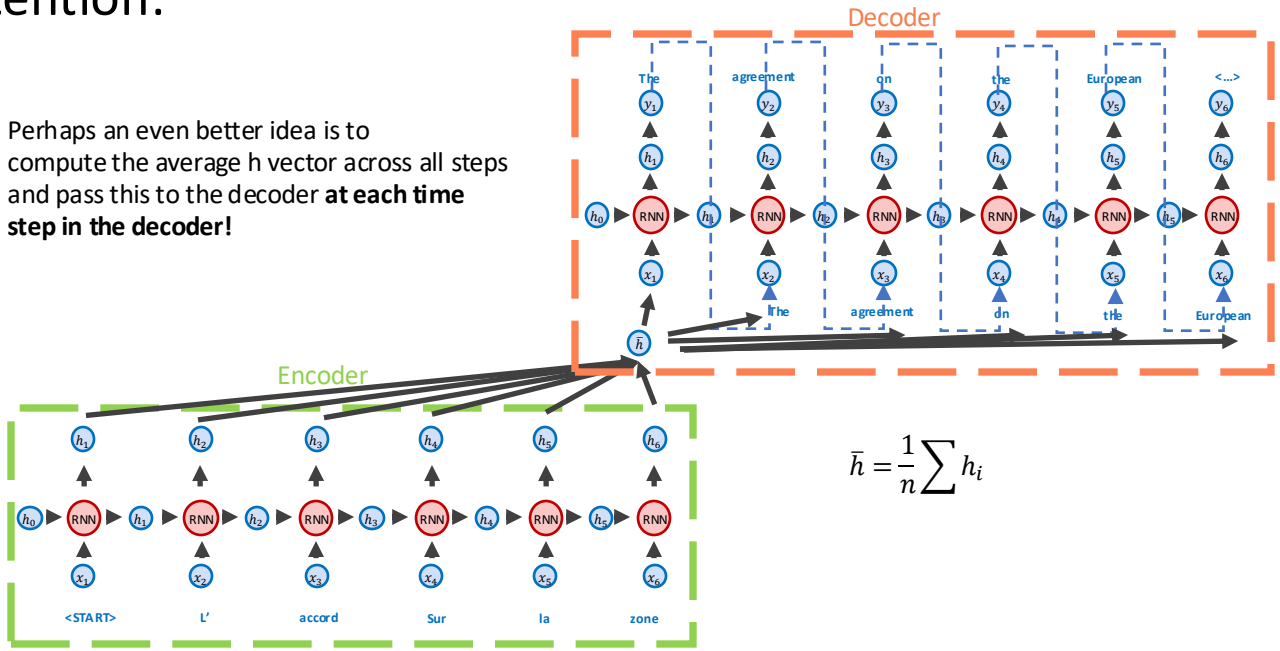


Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

Perhaps an even better idea is to compute the average h vector across all steps and pass this to the decoder **at each time step in the decoder!**



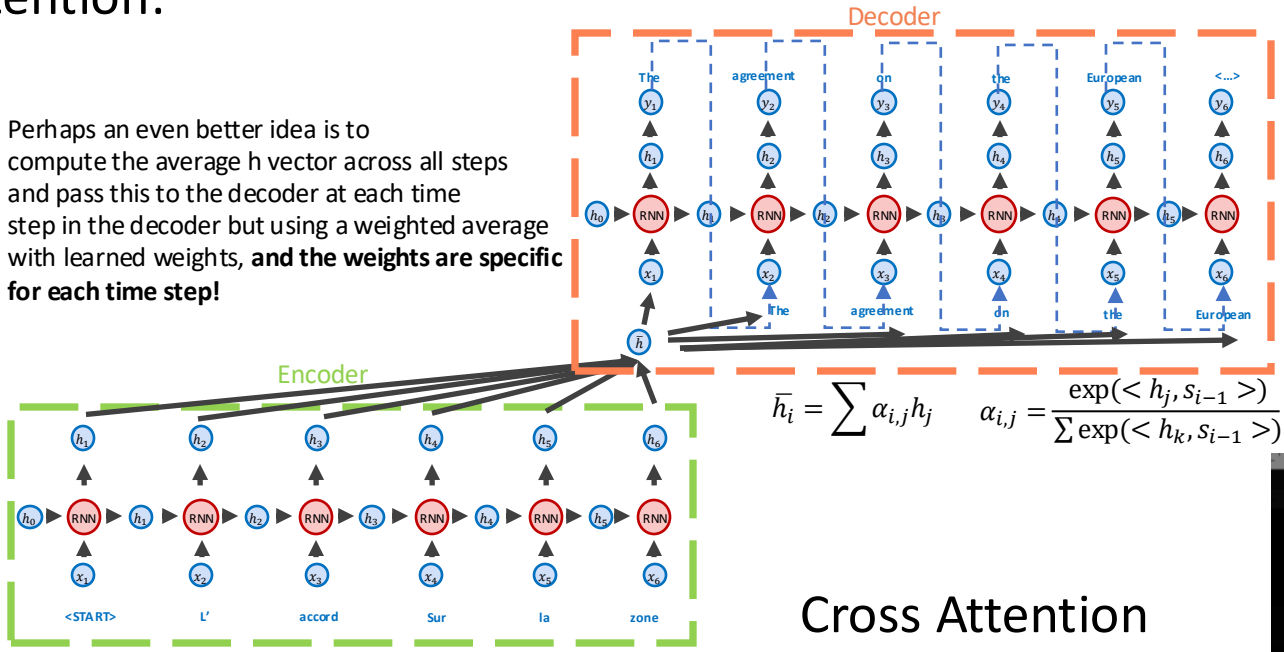
$$\bar{h} = \frac{1}{n}\sum h_i$$

Attention process in NLP

Basic language translation models: Encoder/Decoder

Cross-attention:

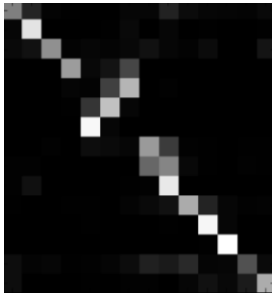
Perhaps an even better idea is to compute the average h vector across all steps and pass this to the decoder at each time step in the decoder but using a weighted average with learned weights, **and the weights are specific for each time step!**



$$\bar{h}_i = \sum \alpha_{i,j} h_j \qquad \alpha_{i,j} = \frac{\exp(< h_j, s_{i-1} >)}{\sum \exp(< h_k, s_{i-1} >)}$$

Cross Attention

Encoder/ Decoder

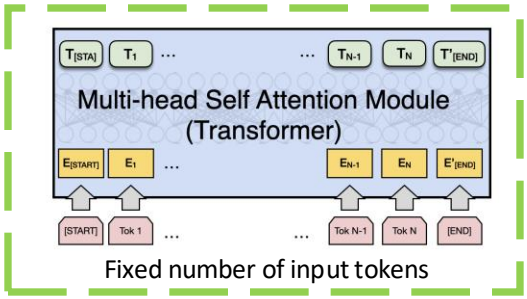


Attention process in NLP

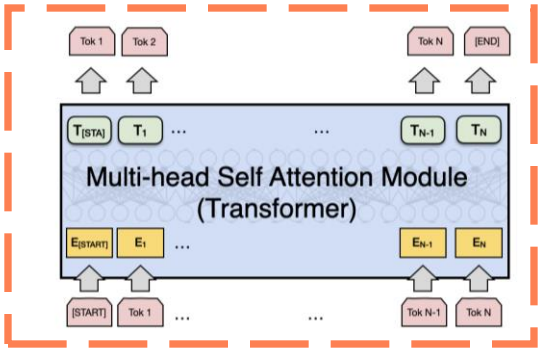
Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

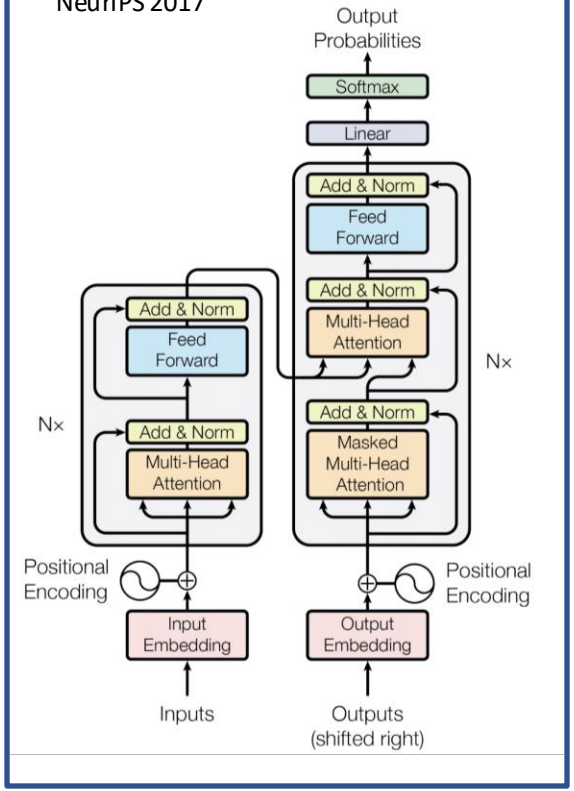
Encoder



Decoder



[Vaswani et al. Attention is all you need]
https://arxiv.org/abs/1706.03762
NeurIPS 2017

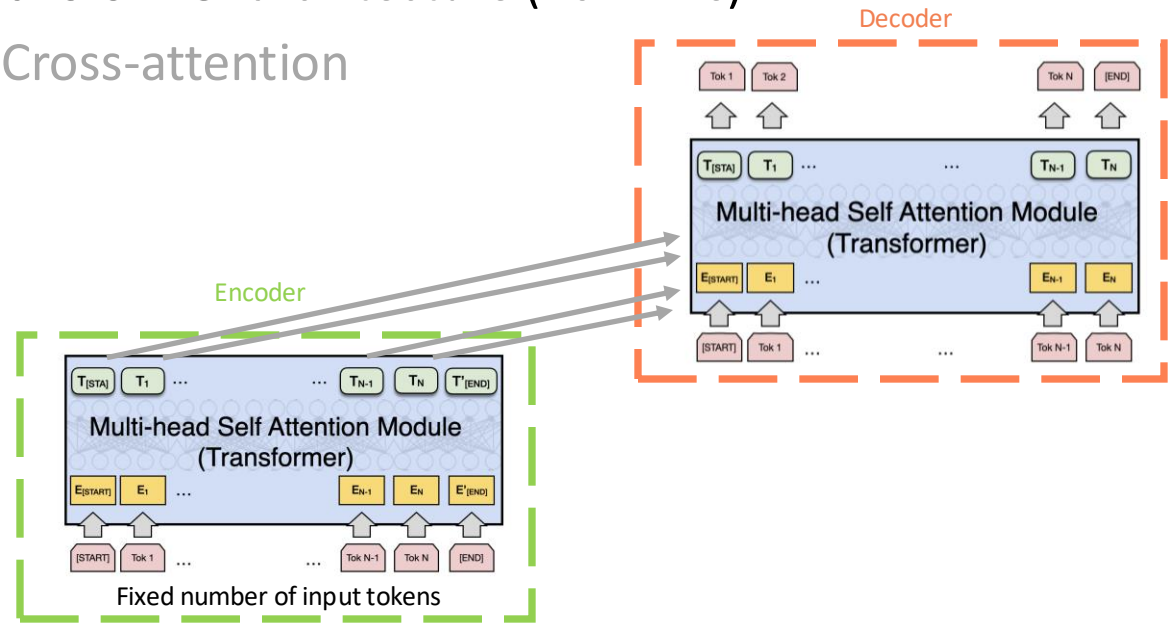


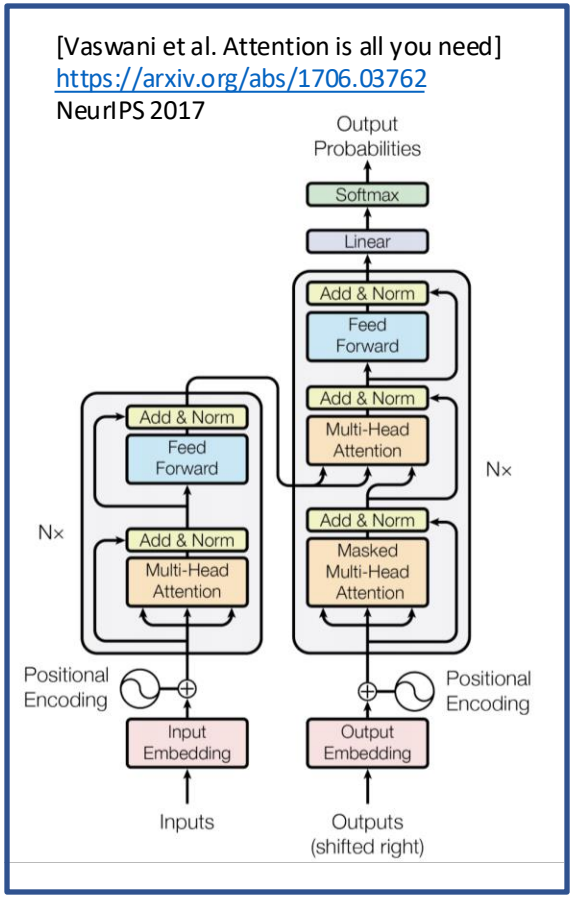
Attention process in NLP

Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

- Cross-attention



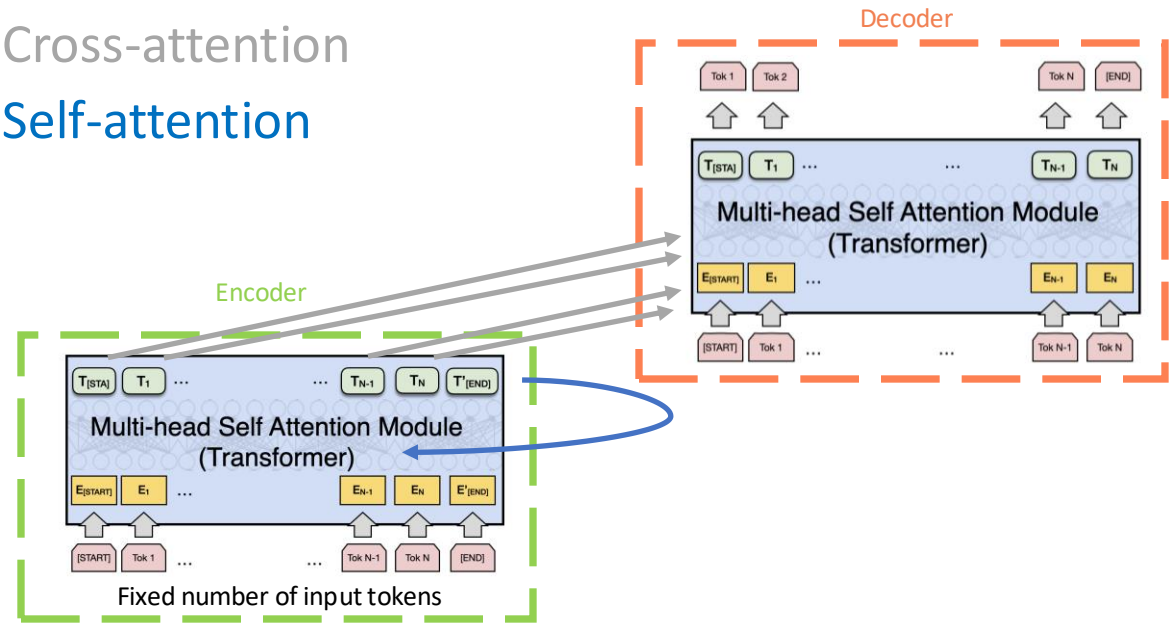


Attention process in NLP

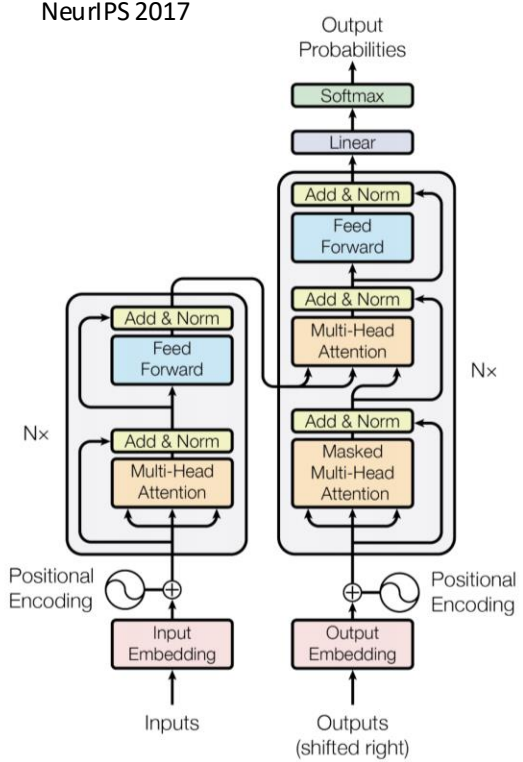
Basic language translation models: Encoder/Decoder

Transformer architecture (no RNNs)

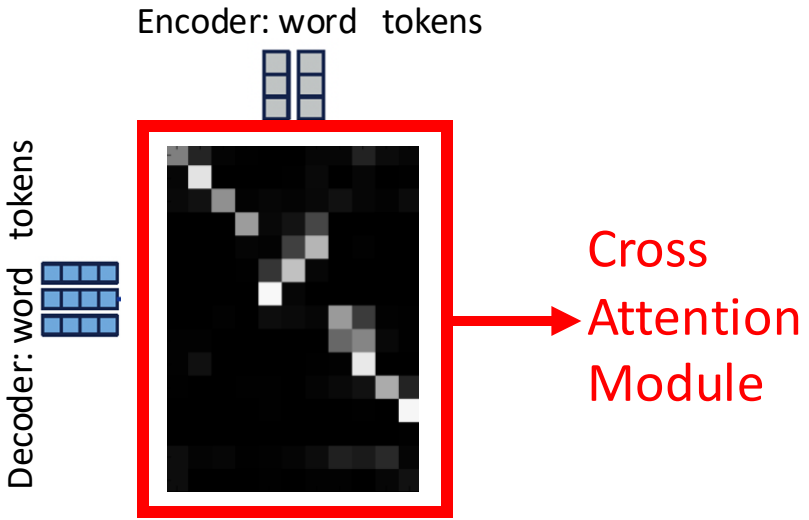
- Cross-attention
- Self-attention

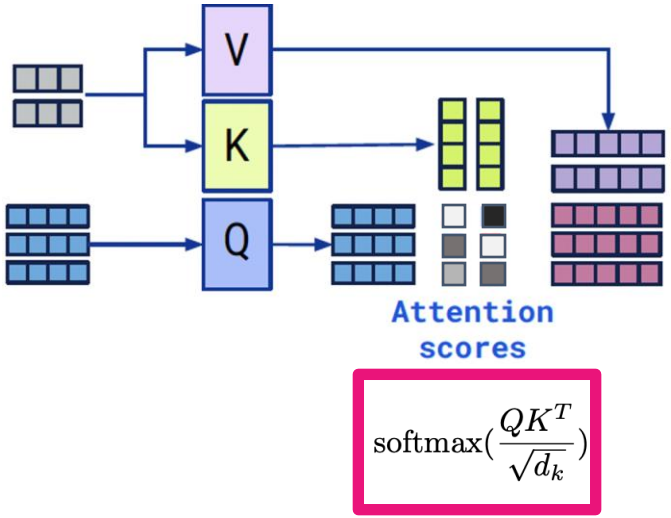


[Vaswani et al. Attention is all you need]
https://arxiv.org/abs/1706.03762
NeurIPS 2017



Attention process in NLP



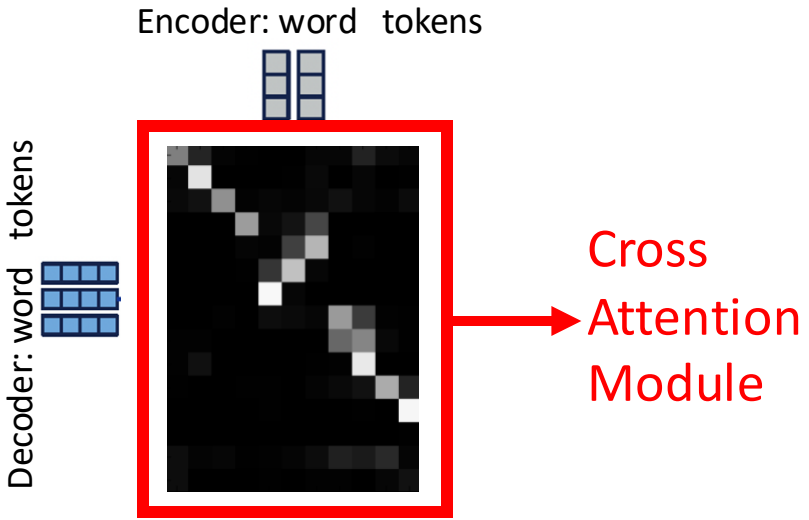


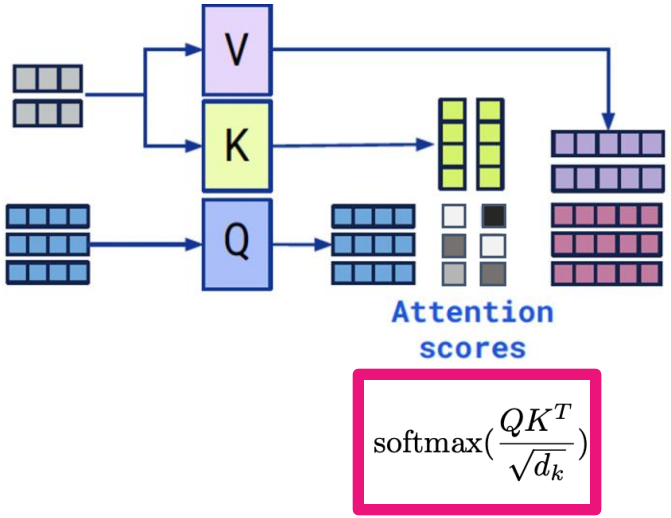
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Vision Transformers

1. NLP: Attention is all you need
2. **Transformer for Image Classification**

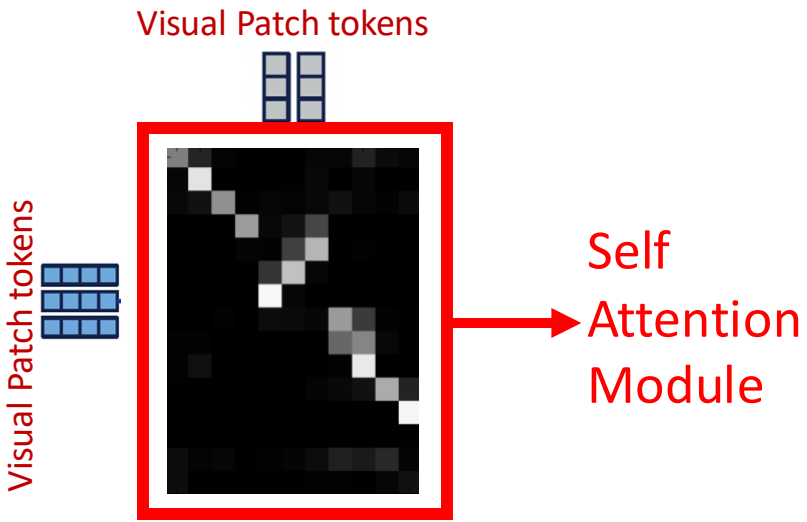
Attention process in NLP

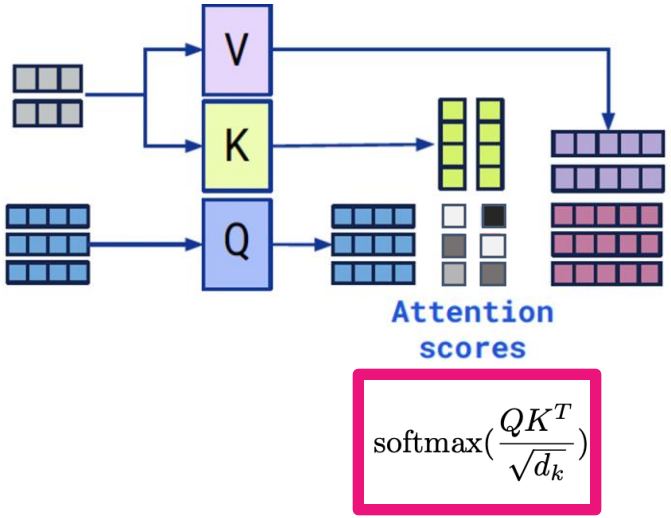




$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

Attention process in Vision





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

Very similar except that Visual token is definitively less natural than word for NLP

Attention process in Vision

Is it possible to mimic this attention-based architecture for vision processing?

Yes! **ViT** (Vision image Transformers) architecture

Published as a conference paper at ICLR 2021

AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE

Alexey Dosovitskiy[*,†], **Lucas Beyer**[*], **Alexander Kolesnikov**[*], **Dirk Weissenborn**[*], **Xiaohua Zhai**[*], **Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, Neil Houlsby**[*,†]
[*]equal technical contribution, [†]equal advising
Google Research, Brain Team
{adosovitskiy, neilhoulsby}@google.com