Transformers

Lucas Beyer

lucasb.eyer.be@gmail.com

@giffmana

http://lb.eyer.be/transformer

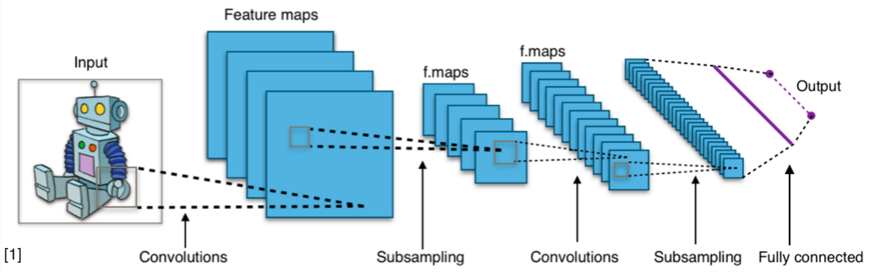
Google Research
Brain Team,
Zürich

The classic landscape:

One architecture per "community"

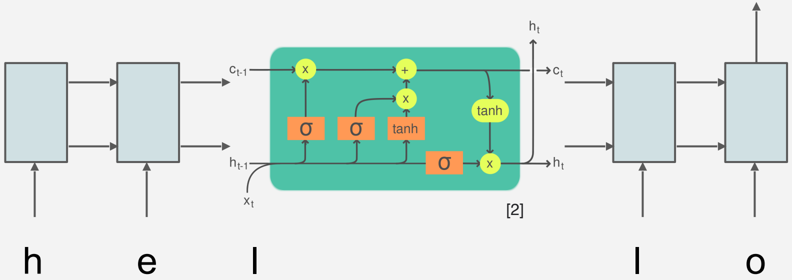
Computer Vision

Convolutional NNs (+ResNets)



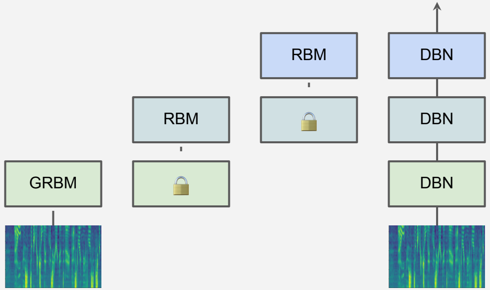
Natural Lang. Proc.

Recurrent NNs (+LSTMs)



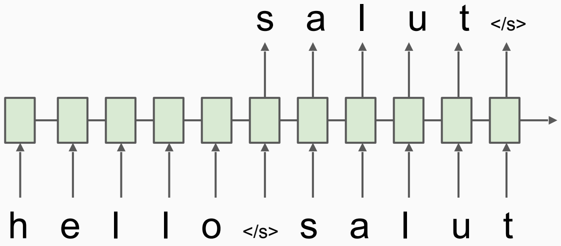
Speech

Deep Belief Nets (+non-DL)



Translation

Seq2Seq



RL

BC/GAIL

Algorithm 1 Generative adversarial imitation learning

1: **Input:** Expert trajectories $\tau_E \sim \pi_E$, initial policy and discriminator parameters θ_0, w_0
2: **for** $i = 0, 1, 2, \ldots$ **do**
3: Sample trajectories $\tau_i \sim \pi_{\theta_i}$
4: Update the discriminator parameters from w_i to w_{i+1} with the gradient

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_w \log(D_w(s,a))] + \hat{\mathbb{E}}_{\tau_E}[\nabla_w \log(1 - D_w(s,a))] \quad (17)$$

5: Take a policy step from θ_i to θ_{i+1}, using the TRPO rule with cost function $\log(D_{w_{i+1}}(s,a))$. Specifically, take a KL-constrained natural gradient step with

$$\hat{\mathbb{E}}_{\tau_i}[\nabla_\theta \log \pi_\theta(a|s) Q(s,a)] - \lambda \nabla_\theta H(\pi_\theta),$$
$$\text{where } Q(\bar{s}, \bar{a}) = \hat{\mathbb{E}}_{\tau_i}[\log(D_{w_{i+1}}(s,a)) \mid s_0 = \bar{s}, a_0 = \bar{a}] \quad (18)$$

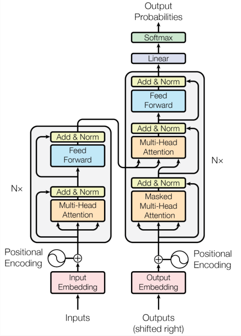
6: **end for**

[1] CNN image CC-BY-SA by Aphex34 for Wikipedia https://commons.wikimedia.org/wiki/File:Typical_cnn.png
[2] RNN image CC-BY-SA by GChe for Wikipedia https://commons.wikimedia.org/wiki/File:The_LSTM_Cell.svg

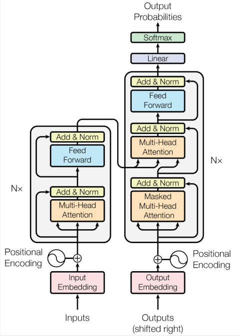
The Transformer's takeover:

One community at a time

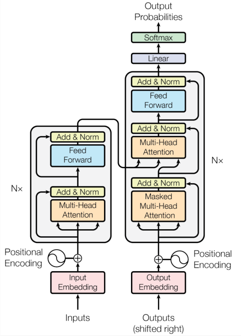
Computer Vision



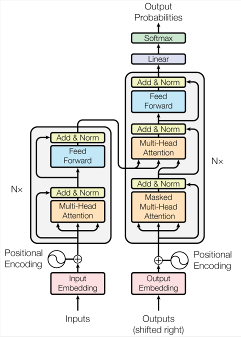
Natural Lang. Proc.



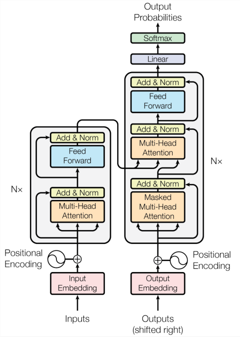
Reinf. Learning



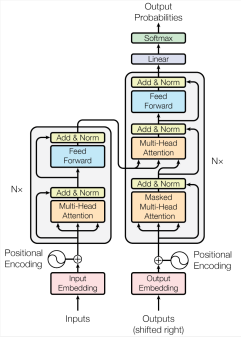
Speech



Translation



Graphs/Science



Transformer image source: "Attention Is All You Need" paper

The origins:

Translation, learned alignment

Neural Machine Translation by Jointly Learning to Align and Translate

2014, Dzmitry Bahdanau, KyungHyun Cho, Yoshua Bengio

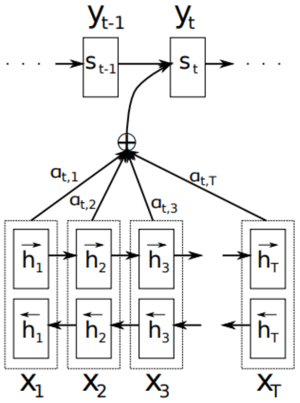
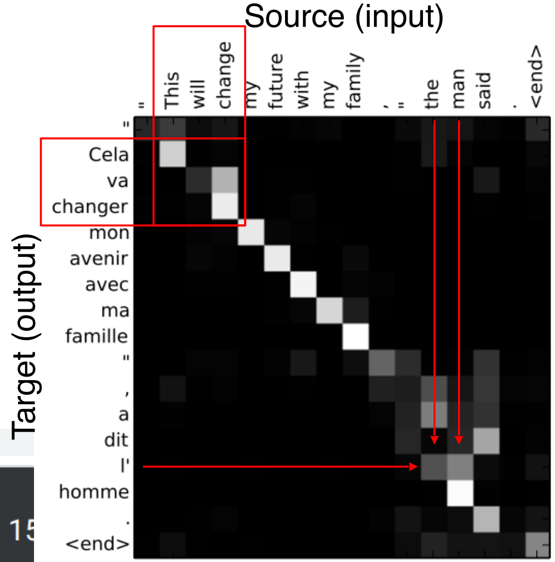


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.



(d)

4 / 15

attention 1/3

The probability α_{ij}, or its associated energy e_{ij}, reflects the importance of the annotation h_j with respect to the previous hidden state s_{i-1} in deciding the next state s_i and generating y_i. Intuitively, this implements a mechanism of attention in the decoder. The decoder decides parts of the source sentence to pay attention to. By letting the decoder have an attention mechanism, we relieve the encoder from the burden of having to encode all information in the source sentence into a fixed-length vector. With this new approach the information can be spread throughout the sequence of annotations, which can be selectively retrieved by the decoder accordingly.

Attention Is All You Need

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

Attention is a building block.

Think of it as a "soft" kv dictionary lookup:

1. Attention weights $a_{1:N}$ are query-key similarities:

$$\hat{a}_i = q \cdot k_i$$

Normalized via softmax: $a_i = e^{\hat{a}_i} / \sum_j e^{\hat{a}_j}$

2. Output z is attention-weighted average of values $v_{1:N}$:

$$z = \sum_i a_i v_i = a \cdot v$$

3. Usually, k and v are derived from the same input x:

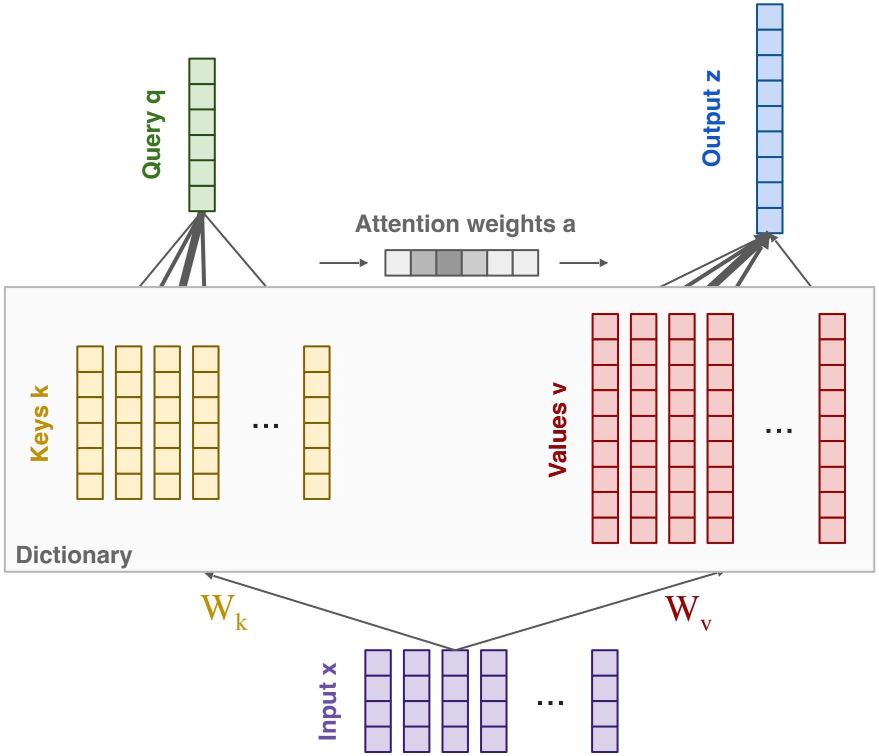
$$k = W_k \cdot x \qquad v = W_v \cdot x$$

The query q can come from a separate input y:

$$q = W_q \cdot y$$

Or from the same input x! Then we call it "self attention":

$$q = W_q \cdot x$$



Historical side-note: "non-local NNs" in computer vision and "relational NNs" in RL appeared almost at the same time and contain the same core idea!

Attention Is All You Need

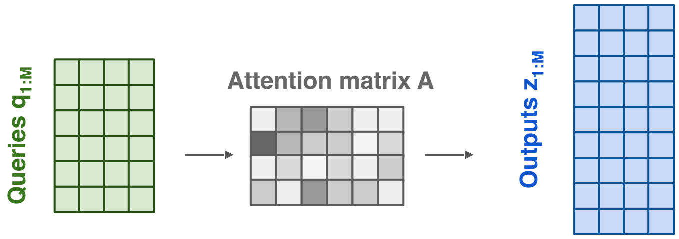
2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

But that's not actually it! There are a few more details:

1.We usually use **many queries** $q_{1:M}$, not just one.

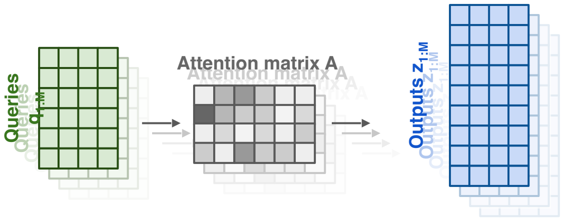
Stacking them leads to the Attention matrix $A_{1:N,1:M}$ and subsequently to many outputs:

$$z_{1:M} = \mathrm{Attn}(q_{1:M},x) = [\mathrm{Attn}(q_1,x) \mid \mathrm{Attn}(q_2,x) \mid \dots \mid \mathrm{Attn}(q_M,x)]$$



2. We usually use "**multi-head**" attention.
This means the operation is repeated K times and the results are concatenated along the feature dimension.
Ws differ.

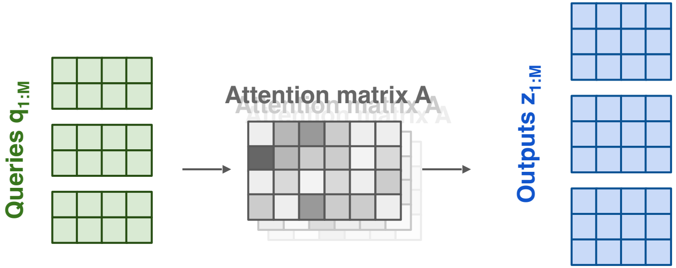
$$z_i = \begin{pmatrix} \mathrm{Attn}_1(q_i,x) \\ \mathrm{Attn}_2(q_i,x) \\ \dots \\ \mathrm{Attn}_K(q_i,x) \end{pmatrix}$$



3. The most commonly seen formulation:

$$z = \mathrm{softmax}(QK'/\sqrt{d_{key}})V$$

Note that the complexity is O(N²)



Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

Encoder

Task: read and “understand” the user’s input.

Decoder

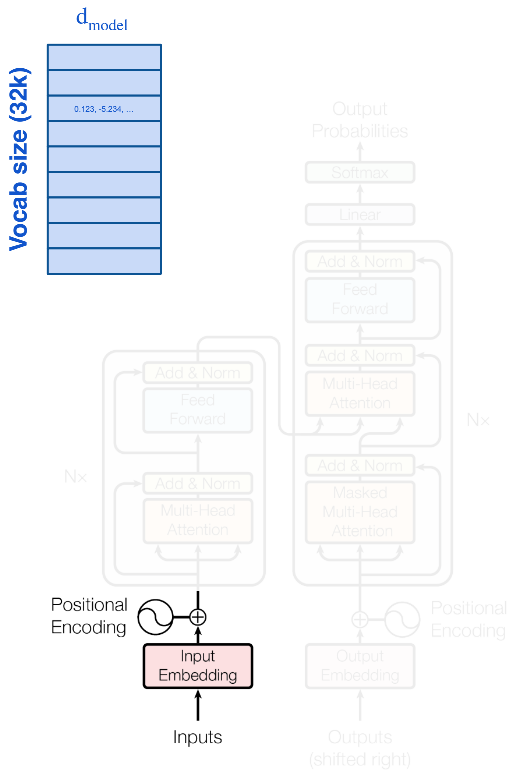
Task: generate the output (eg answer the user’s query).

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Input (Tokenization and) Embedding

Input text is first split into pieces. Can be characters, word, "tokens":

`"The detective investigated" -> [The_] [detective_] [invest] [igat] [ed_]`

Tokens are indices into the "vocabulary":

`[The_] [detective_] [invest] [igat] [ed_] -> [3 721 68 1337 42]`

Each vocab entry corresponds to a learned d_{model}-dimensional vector.

`[3 721 68 1337 42] -> [ [0.123, -5.234, ...], [...], [...], [...], [...] ]`

Positional Encoding

Remember attention is permutation invariant, but language is not!

("The mouse ate the cat" vs "The cat ate the mouse")

Need to encode position of each word; just add something.

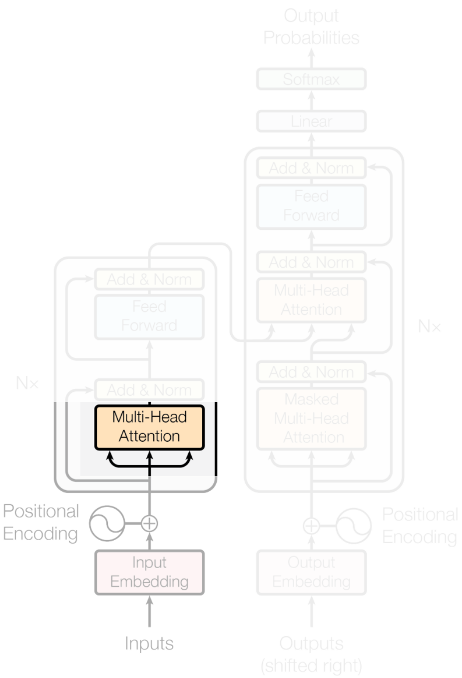
Think `[The_] + 10` `[detective_] + 20` `[invest] + 30` ... but smarter.

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

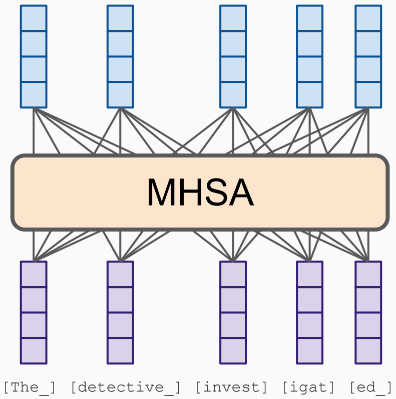
Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Multi-headed Self-Attention

Meaning the input sequence is used to create queries, keys, and values!
Each token can "look around" the whole input, and decide how to update its representation based on what it sees.

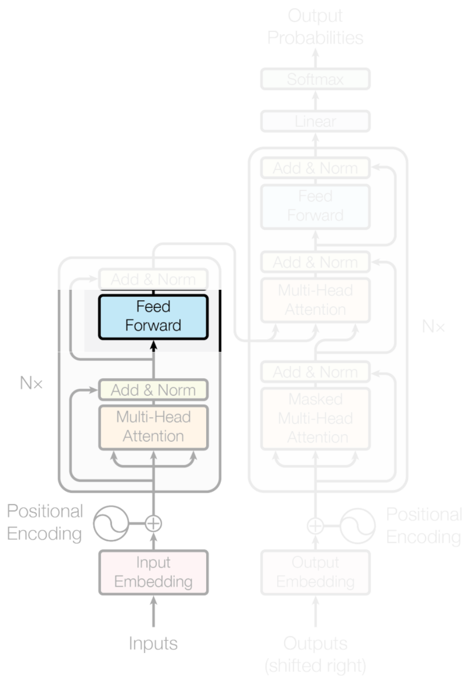


Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Point-wise MLP

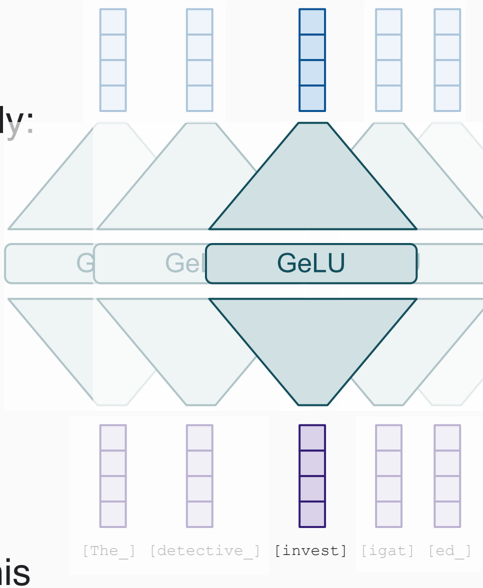
A simple MLP applied to each token individually:

$$z_i = W_2 \, \mathrm{GeLU}(W_1 x + b_1) + b_2$$

Think of it as each token pondering for itself about what it has observed previously. There's some weak evidence this is where "world knowledge" is stored, too.

It contains the bulk of the parameters. When people make giant models and sparse/moe, this is what becomes giant.

Some people like to call it 1x1 convolution.

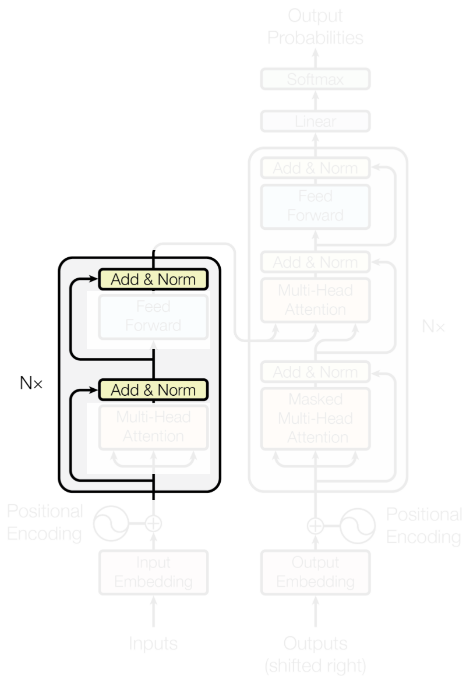


Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Residual/skip connections

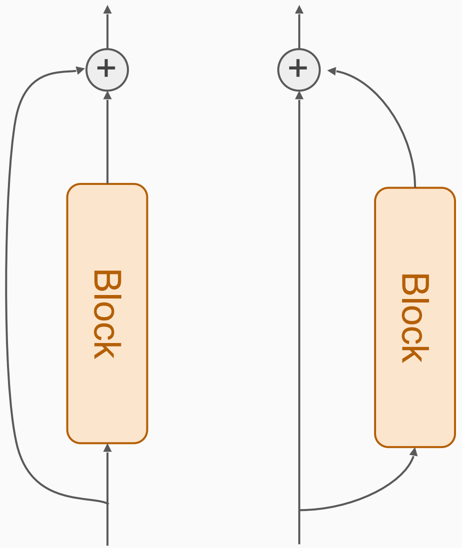
Each module's output has the exact same shape as its input.

Following ResNets, the module computes a "residual" instead of a new value:

$$z_i = \text{Module}(x_i) + x_i$$

This was shown to dramatically improve trainability.

"Skip connection" == "Residual block"



LayerNorm

Normalization also dramatically improves trainability.

There's **post-norm** (original) and **pre-norm** (modern)

$$z_i = \text{LN}(\text{Module}(x_i) + x_i)$$

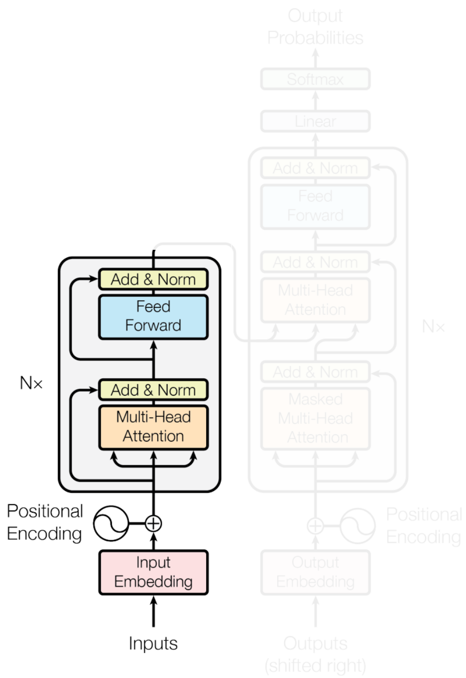
$$z_i = \text{Module}(\text{LN}(x_i)) + x_i$$

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Encoding / Encoder

Since input and output shapes are identical, we can stack N such blocks.

Typically, N=6 ("base"), N=12 ("large") or more.

Encoder output is a "heavily processed" (think: "high level, contextualized") version of the input tokens, i.e. a sequence.

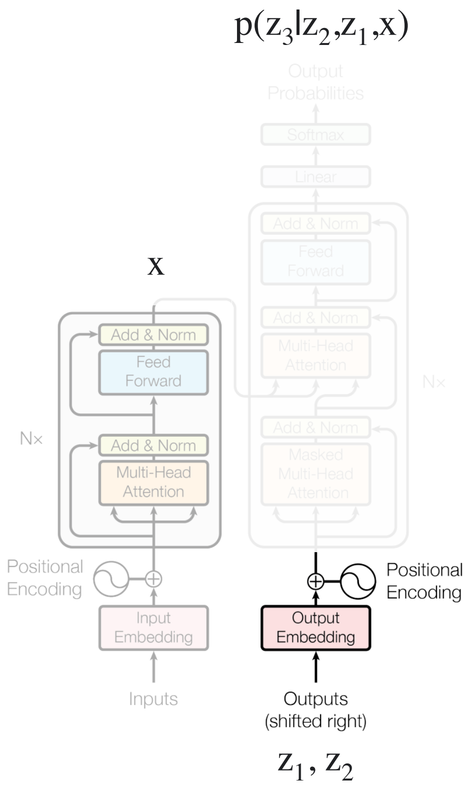
This has nothing to do with the requested output yet (think: translation). That comes with the decoder.

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Decoding / the Decoder **(alternatively Generating / the Generator)**

What we want to model: $p(z|x)$

for example, in translation: $p(z \mid \text{"the detective investigated"})\ \forall z$

Seems impossible at first, but we can *exactly* decompose into tokens:

$$p(z|x) = p(z_1|x)\ p(z_2|z_1,x)\ p(z_3|z_2,z_1,x)\ldots$$

Meaning, we can compute the likelihood of a given output z,
or generate/sample an answer z one token at a time.
Each p is a full pass through the model.

For generating $p(z_3|z_2,z_1,x)$:

x comes from the encoder,

z_1, z_2 is what we have predicted so far, goes into the decoder.

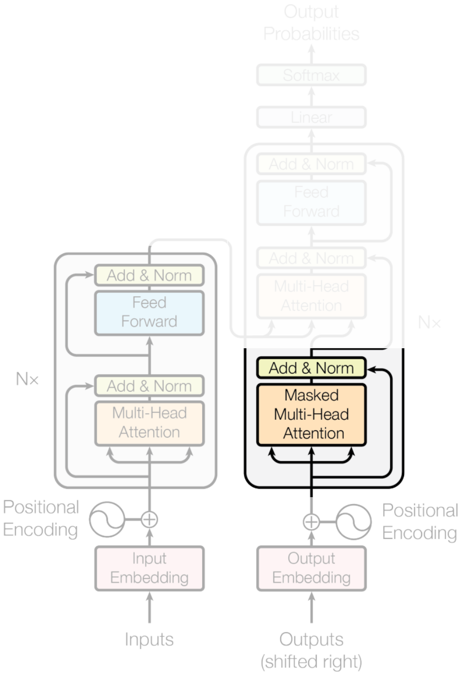
Once we have a $p(z_i|z_{:i},x)$ we still need to actually sample a sentence such as "le détective a enquêté". Many strategies: greedy, beam, ...

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



At training time: Masked self-attention

This is regular self-attention as in the encoder, to process what's been decoded so far, eg z_2, z_1 in $p(z_3 | z_2, z_1, x)$, but with a trick...

If we had to train on one single $p(z_3 | z_2, z_1, x)$ at a time: SLOW!

Instead, train on all $p(z_i | z_{1:i}, x)$ for all i simultaneously.

How? In the attention weights for z_i, set all entries $i{:}N$ to 0.

This way, each token only sees the already generated ones.

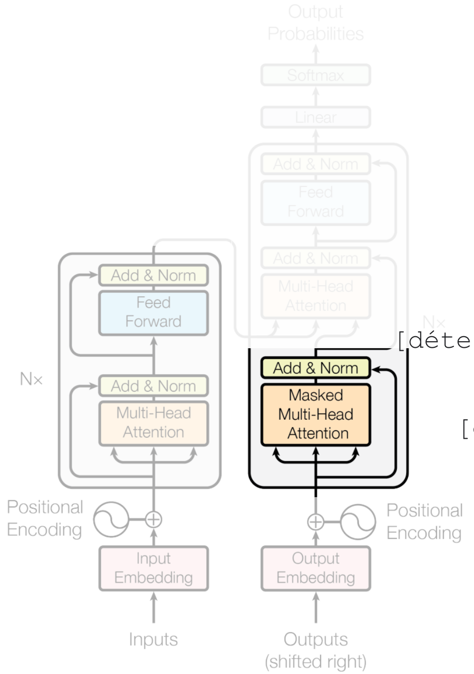
Let's see what that means…

Thanks to Basil Mustafa for slide inspiration

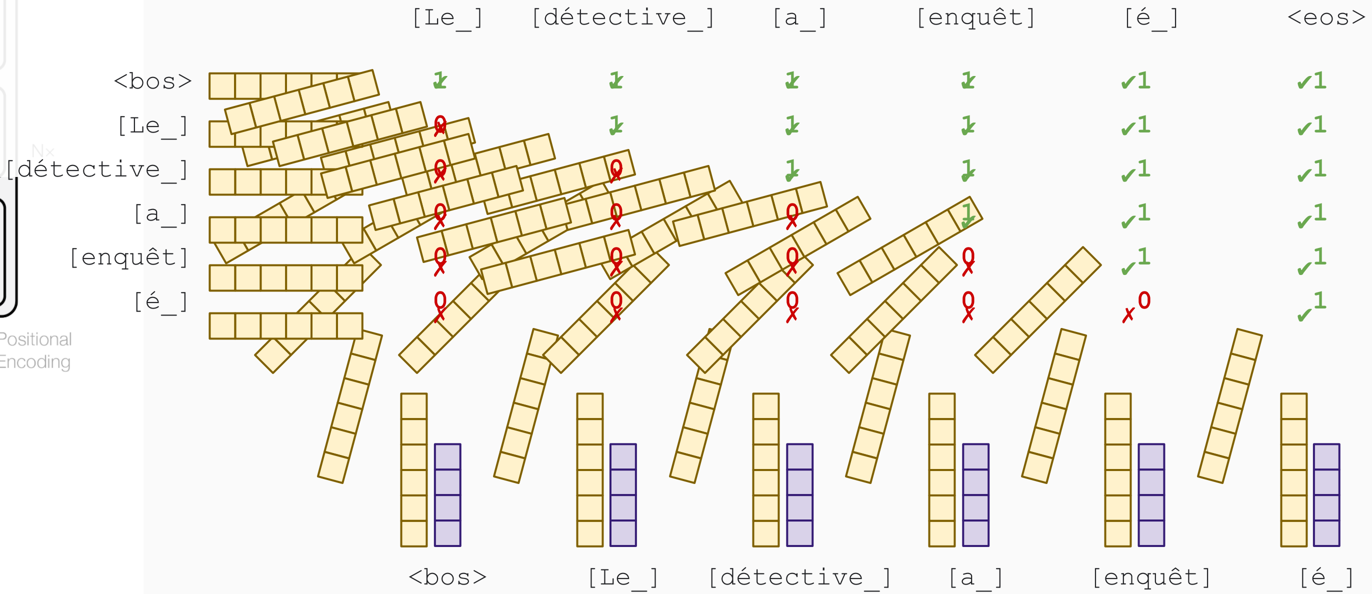
Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



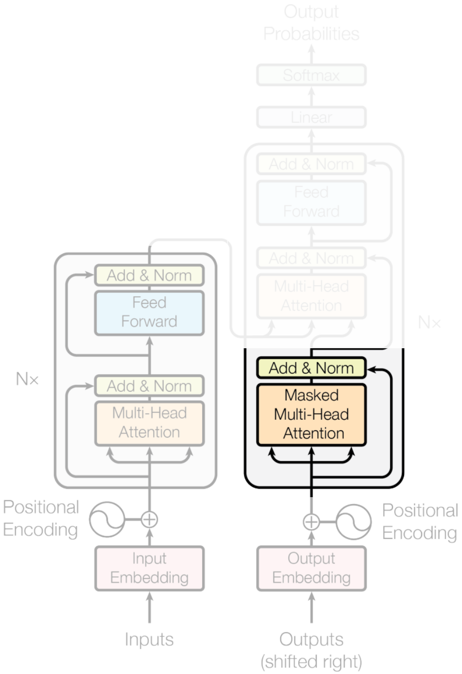
At training time: Masked self-attention



Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

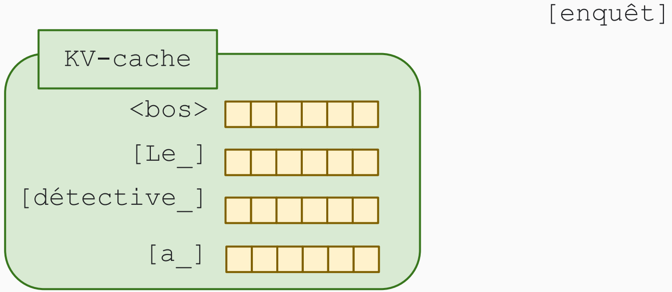
2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



At generation time

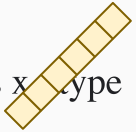
There is no such trick. We need to generate one z_i at a time. This is why autoregressive decoding is extremely slow.

But: we can cache k/v activations from previous tokens and re-use.



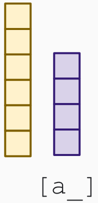
Note cache size:

(Seq x Heads x d_{model}) x Layers x dtype



Example Gemma-9B:

(4096 x 16 x 3584) x 42 x 4 = 36.75 GiB



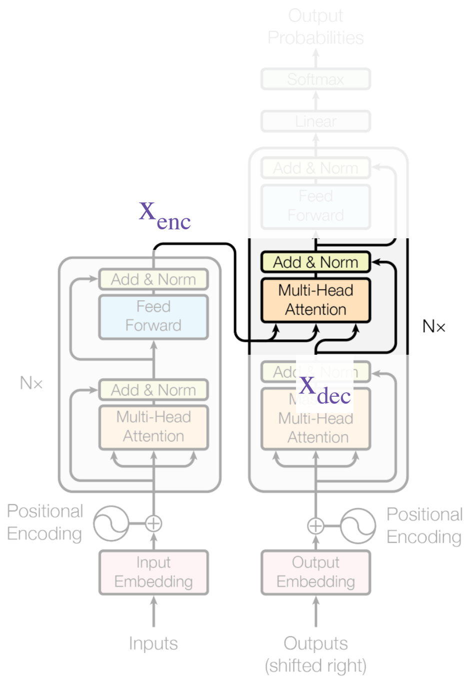
⇒ Research: GQA, MQA, kv-compression, …

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

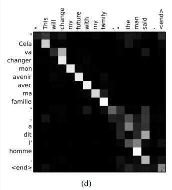
2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



"Cross" attention

Each decoded token can "look at" the encoder's output:

$$\mathrm{Attn}(q=W_q x_{dec},\ k=W_k x_{enc},\ v=W_v x_{enc})$$



This is the same as in the 2014 paper.

This is where $|x$ in $p(z_3|z_2,z_1,x)$ comes from.

Because self-attention is so widely used, people have started just calling it "attention".

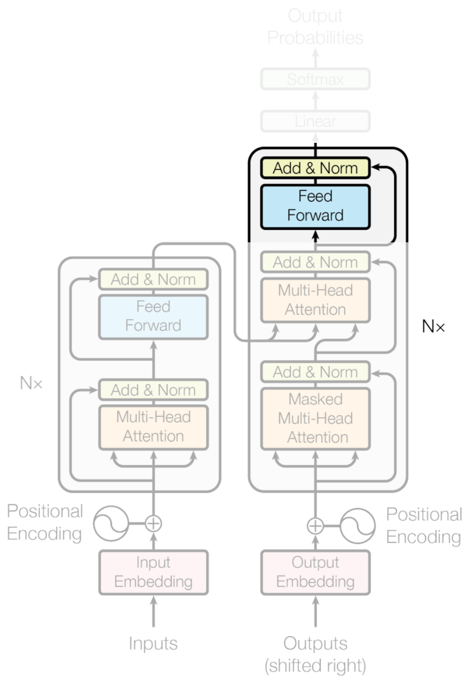
Hence, we now often need to explicitly call this "cross attention".

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper ; heatmap image source: "Neural Machine Translation by Jointly Learning to Align and Translate" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



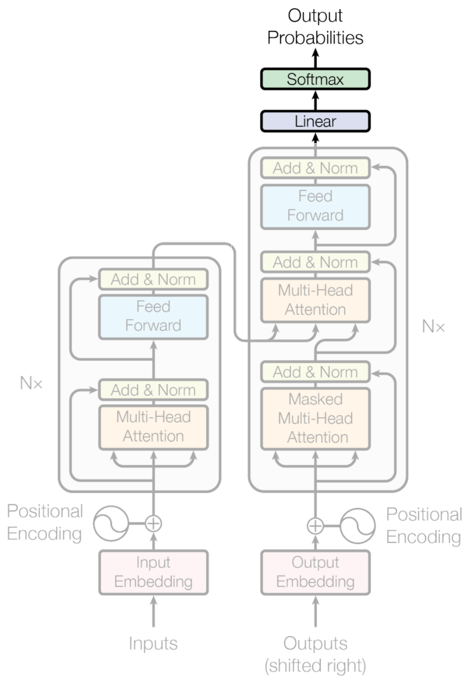
Feedforward and stack layers.

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - The Transformer architecture

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin



Output layer

Assume we have already generated K tokens, generate the next one.

The decoder was used to gather all information necessary to predict a probability distribution for the next token (K), over the whole vocab.

Simple:

- linear projection of token K
- SoftMax normalization

Thanks to Basil Mustafa for slide inspiration

Transformer image source: "Attention Is All You Need" paper

Attention Is All You Need - Summary and results

2017, Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

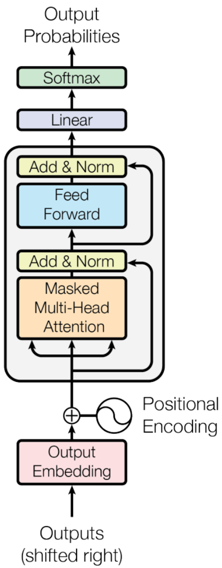
Animation source: https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html
Table source: "Attention Is All You Need" paper

The first (1.5th) big takeover:

Language Modeling / NLP

Decoder-only GPT

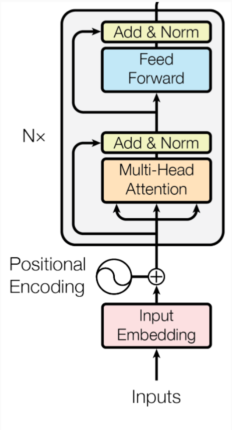
[sat_]



[START] [The_] [cat_]

Encoder-only BERT

[*] [*] **[sat_]** [*] **[the_]** [*]



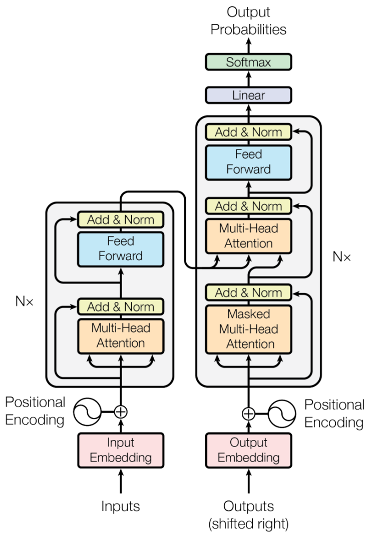
[The_] [cat_] **[MASK]** [on_] **[MASK]** [mat_]

Enc-Dec T5

Das ist gut.

A storm in Attala caused 6 victims.

This is not toxic.



Translate EN-DE: This is good.

Summarize: state authorities dispatched…

Is this toxic: You look beautiful today!

Transformer image source: "Attention Is All You Need" paper

The second big takeover:

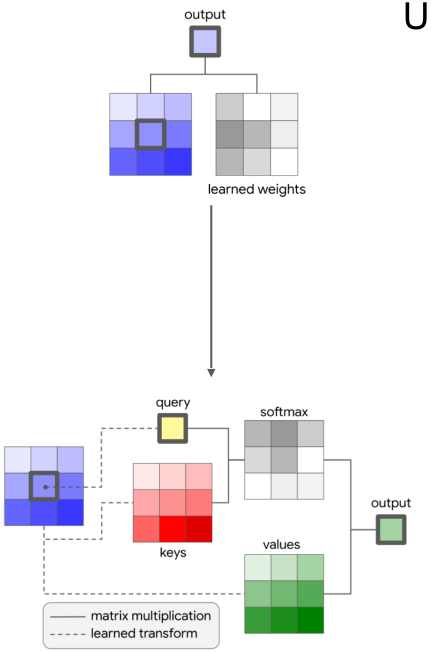
Computer Vision

Many prior works attempted to introduce self-attention at the pixel level.

For 224px², that's 50k sequence length, too much!

Previous approaches

1. On pixels, but locally or factorized



Usually replaces 3x3 conv in ResNet:

stage	output	ResNet-50		LR-Net-50 (7×7, m=8)	
res1	112×112	7×7 conv, 64, stride 2		**1×1, 64** **7×7 LR, 64, stride 2**	
res2	56×56	3×3 max pool, stride 2		3×3 max pool, stride 2	
		1×1, 64 3×3 conv, 64 1×1, 256	×3	1×1, 100 **7×7 LR, 100** 1×1, 256	×3
res3	28×28	1×1, 128 3×3 conv, 128 1×1, 512	×4	1×1, 200 **7×7 LR, 200** 1×1, 512	×4
res4	14×14	1×1, 256 3×3 conv, 256 1×1, 1024	×6	1×1, 400 **7×7 LR, 400** 1×1, 1024	×6
res5	7×7	1×1, 512 3×3 conv, 512 1×1, 2048	×3	1×1, 800 **7×7 LR, 800** 1×1, 2048	×3
	1×1	global average pool 1000-d fc, softmax		global average pool 1000-d fc, softmax	
# params		**25.5**×10^6		**23.3**×10^6	
FLOPs		**4.3**×10^9		**4.3**×10^9	

Results:

Are usually "meh", nothing to call home about

Do not justify increased complexity

Do not justify slowdown over convolutions

Examples:

Non-local NN (Wang et.al. 2017)

SASANet (Stand-Alone Self-Attention in Vision Models)

HaloNet (Scaling Local Self-Attn for Parameter Efficient...)

LR-Net (Local Relation Networks for Image Recognition)

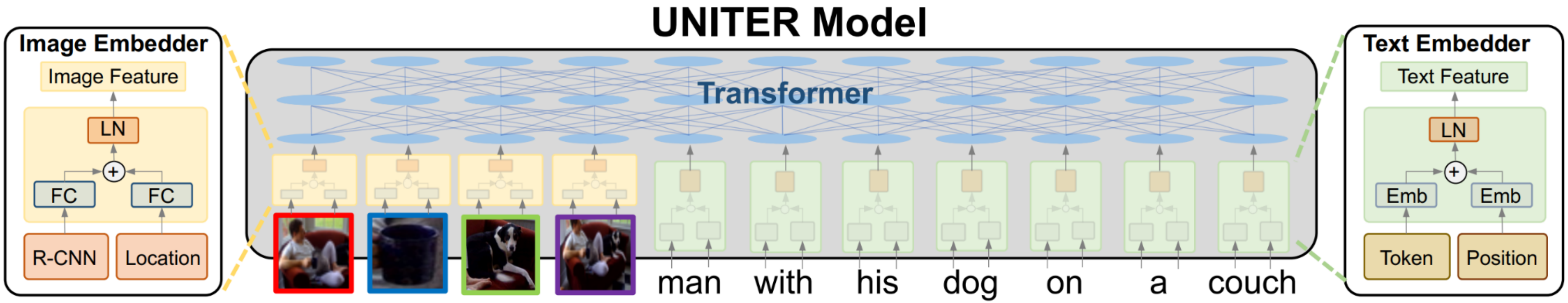
SANet (Exploring Self-attention for Image Recognition)

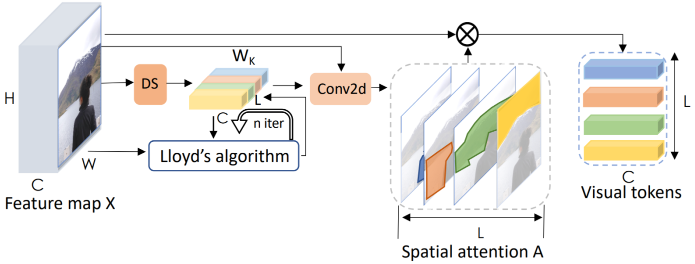
...

Image credit: Stand-Alone Self-Attention in Vision Models by Ramachandran et.al.
Image credit: Local Relation Networks for Image Recognition by Hu et.al.

Previous approaches

2. Globally, after/inside a full-blown CNN, or even detector/segmenter!





Cons:
result is highly complex, often multi-stage trained architecture.
not from pixels, i.e. transformer can't "learn to fix" the (often frozen!) CNN's mistakes.

Examples:
DETR (Carion, Massa et.al. 2020)
UNITER (Chen, Li, Yu et.al. 2019)
etc...
VisualBERT (Li et.al. 2019)
Visual Transformers (Wu et.al. 2020)
ViLBERT (Lu et.al. 2019)

Image credit: UNITER: UNiversal Image-TExt Representation Learning by Chen et.al.
Image credit: Visual Transformers: Token-based Image Representation and Processing for Computer Vision by Wu et.al.

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale

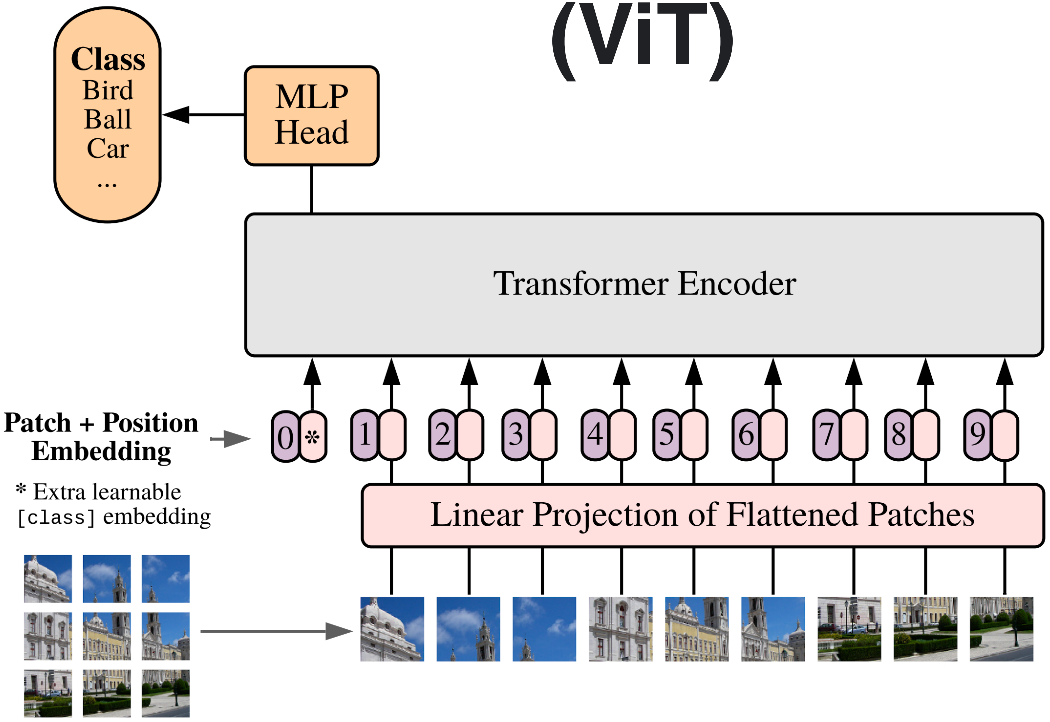
2020, A Dosovitskiy, L Beyer, A Kolesnikov, D Weissenborn, X Zhai, T Unterthiner, M Dehghani, M Minderer, G Heigold, S Gelly, J Uszkoreit, N Houlsby

Vision Transformer (ViT)

Many prior works attempted to introduce self-attention at the pixel level.

For 224px², that's 50k sequence length, too much!

Thus, most works restrict attention to local pixel neighborhoods, or as high-level mechanism on top of detections.



The **key breakthrough** in using the full Transformer architecture, standalone, was to **"tokenize" the image** by **cutting it into patches** of 16px², and treating each patch as a token, e.g. embedding it into input space.

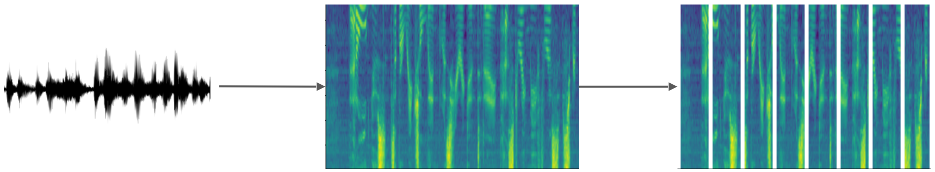
The third big takeover:

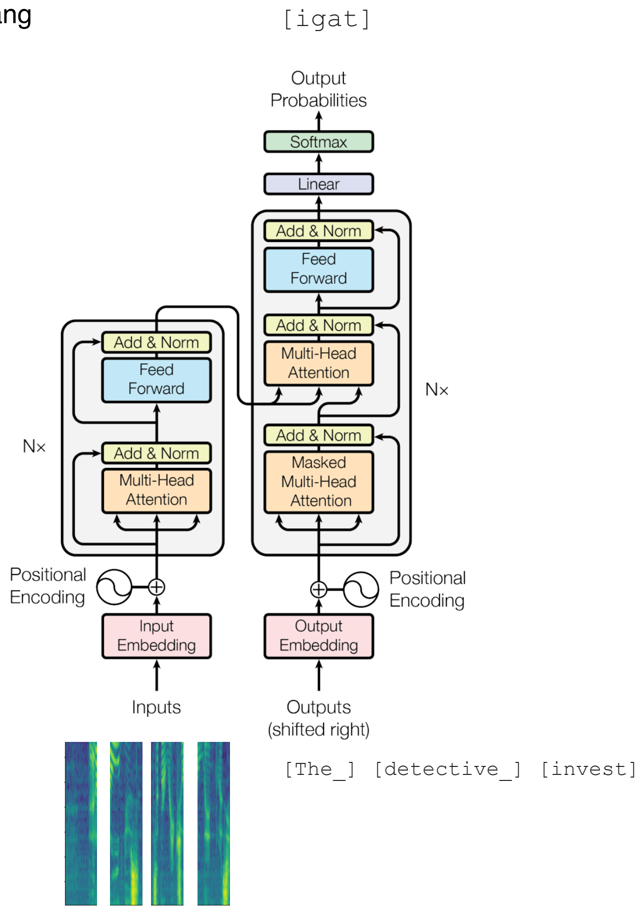
Speech

Conformer: Convolution-augmented Transformer for Speech Recognition

2020, A Gulati, J Qin, C-C Chiu, N Parmar, Y Zhang, J Yu, W Han, S Wang, Z Zhang, Y Wu, R Pang

Largely the same story as in computer vision.
But with spectrograms instead of images.





Conformer adds a third type of block using convolutions, and slightly reorder blocks, but overall very transformer-like.

Exists as encoder-decoder variant, or as encoder-only variant with CTC loss.

Sept 2022: Plain transformer model in “Whisper” by
A Radford, J W Kim, T Xu, G Brockman, C McLeavey, I Sutskever

Transformer image source: "Attention Is All You Need" paper

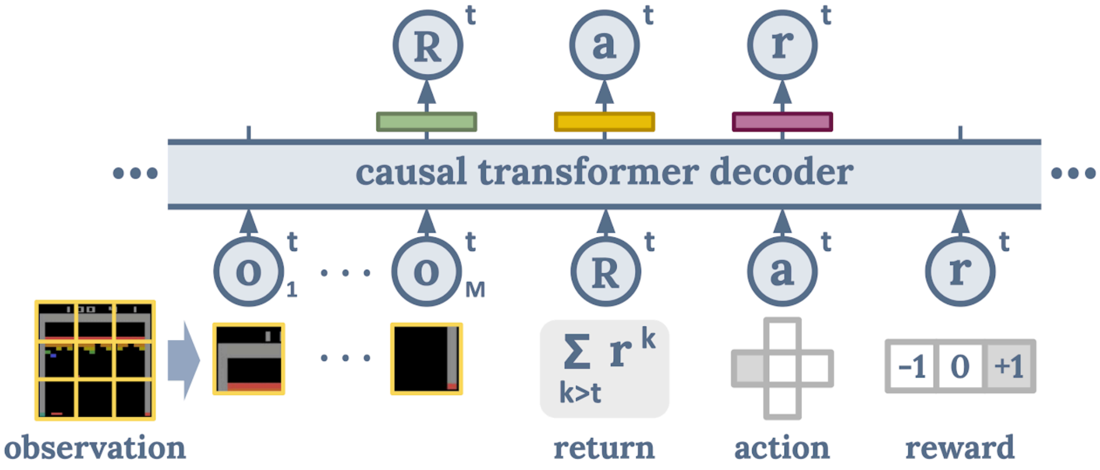
The fourth big takeover:

Reinforcement Learning

Decision Transformer: Reinforcement Learning via Sequence Modeling

2021, L Chen, K Lu, A Rajeswaran, K Lee, A Grover, M Laskin, P Abbeel, A Srinivas, I Mordatch

Cast the (supervised/offline) RL problem into a sequence ("language") modeling task:



Can generate/decode sequences of actions with desired return (eg skill)

The trick is prompting: `"The following is a trajectory of an expert player: [obs] ..."`

Slide credit: Igor Mordatch

The Transformer's

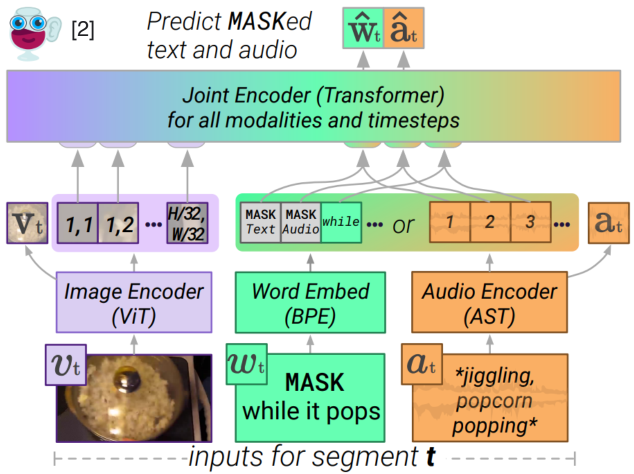
Unification of communities

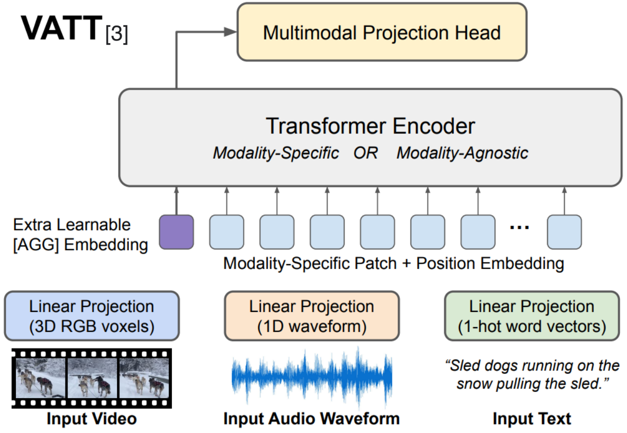
Anything you can tokenize, you can feed to Transformer

ca 2021 and onwards

Tokenize different modalities each in their own way (some kind of "patching"), and send them all jointly into a Transformer...

Seems to just work...
Currently an explosion of works doing this!





[1]

Images from:
[1] LIMoE by B Mustafa, C Riquelme, J Puigcerver, R Jenatton, N Houlsby
[2] MERLOT Reserve by R Zellers, J Lu, X Lu, Y Yu, Y Zhao, M Salehi, A Kusupati, J Hessel, A Farhadi, Y Choi
[3] VATT by H Akbari, L Yuan, R Qian, W-H Chuang, S-F Chang, Y Cui, B Gong

A note on

Efficient Transformers

A note on Efficient Transformers

The self-attention operation complexity is O(N²) for sequence length N.

We'd like to use large N:

- Whole articles or books
- Full video movies
- High resolution images

Many O(N) approximations to the full self-attention have been proposed in the past two years.

Unfortunately, none provides a clear improvement.
They always trade-off between speed and quality.

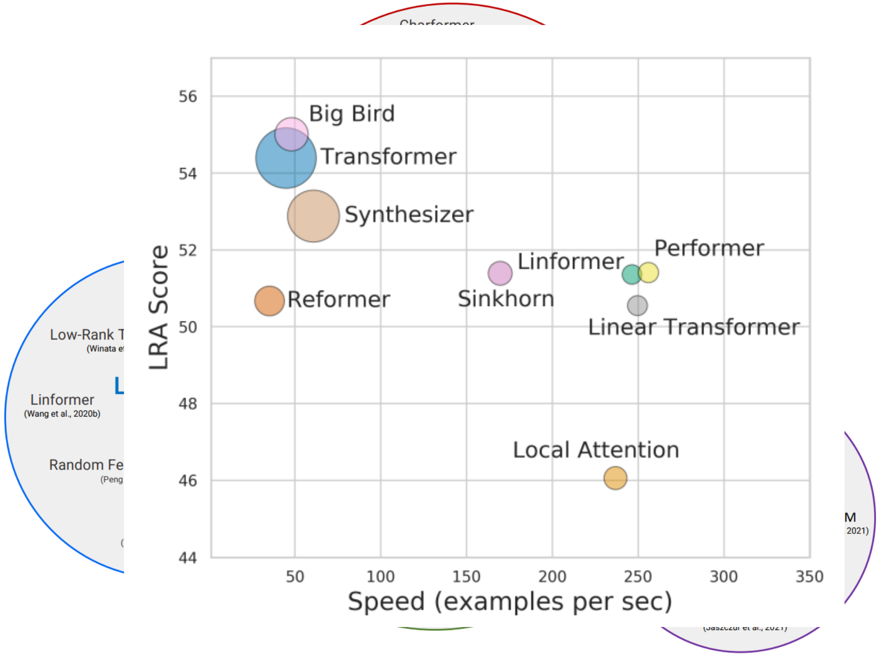


Figure 2: Taxonomy of Efficient Transformer Architectures.

Based on "Efficient Transformers: A Survey" by Yi Tay, Mostafa Dehghani, Dara Bahri, Donald Metzler
and "Long Range Arena: A Benchmark for Efficient Transformers" by Y Tay, M Dehghani, S Abnar, Y Shen, D Bahri, P Pham, J Rao, L Yang, S Ruder, D Metzler

Thanks for your... 🥁

Attention

Transformer image source: "Attention Is All You Need" paper