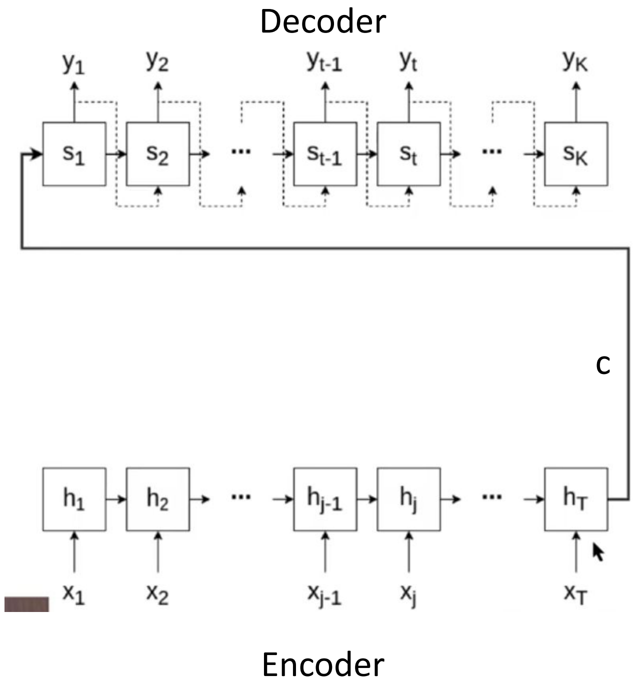
Introduction to transformers

Chang Bi

Sequence to Sequence Models



$$h_t = f\left(x_t, h_{t-1}\right)$$

$$c = q\left(\left\{h_1, \cdots, h_{T_x}\right\}\right)$$

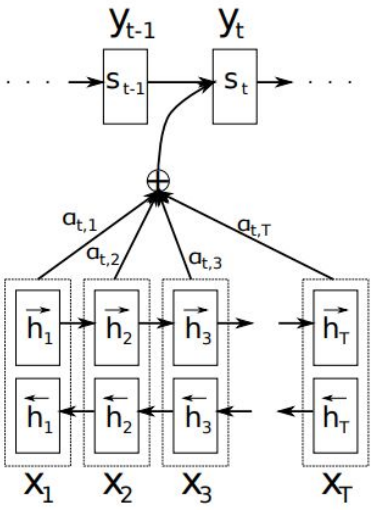
$$p(\mathbf{y}) = \prod_{t=1}^{T} p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c)$$

$$p(y_t \mid \{y_1, \cdots, y_{t-1}\}, c) = g(y_{t-1}, s_t, c)$$

The network summarize the input, $x_{\{1,\ldots,T\}}$, into a unique context vector c and infer output conditioned on c and previous sequence $y_{\{1,\ldots,t-1\}}$.

Bahdanau Attention

Decoder



Encoder

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

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

$$\alpha_{t,i} = \frac{\exp\left(\mathrm{score}(\mathbf{s}_{t-1}, \mathbf{h}_i)\right)}{\sum_{i'=1}^{n} \exp\left(\mathrm{score}(\mathbf{s}_{t-1}, \mathbf{h}_{i'})\right)}$$

Aside the bidirectional structure in the encoder part, this network has multiple context vectors that "tailor" for each output. Each c_i has a set of weights, a_{ij}, to guide it focusing on the most relevant part of the input to the target.

Attention mechanism of human

What color is the flower, and what color are the leaves?

Two ways to answer the question with the image:

1. The notes on the image before look at the image, then answer the question based on the notes.
2. Look at the image and question at the same time and only pay attention to the relevant parts of the image to the question.

What about JUST use the attention?

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com

Noam Shazeer* Google Brain noam@google.com

Niki Parmar* Google Research nikip@google.com

Jakob Uszkoreit* Google Research usz@google.com

Llion Jones* Google Research llion@google.com

Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu

Łukasz Kaiser* Google Brain lukaszkaiser@google.com

Illia Polosukhin* ‡ illia.polosukhin@gmail.com

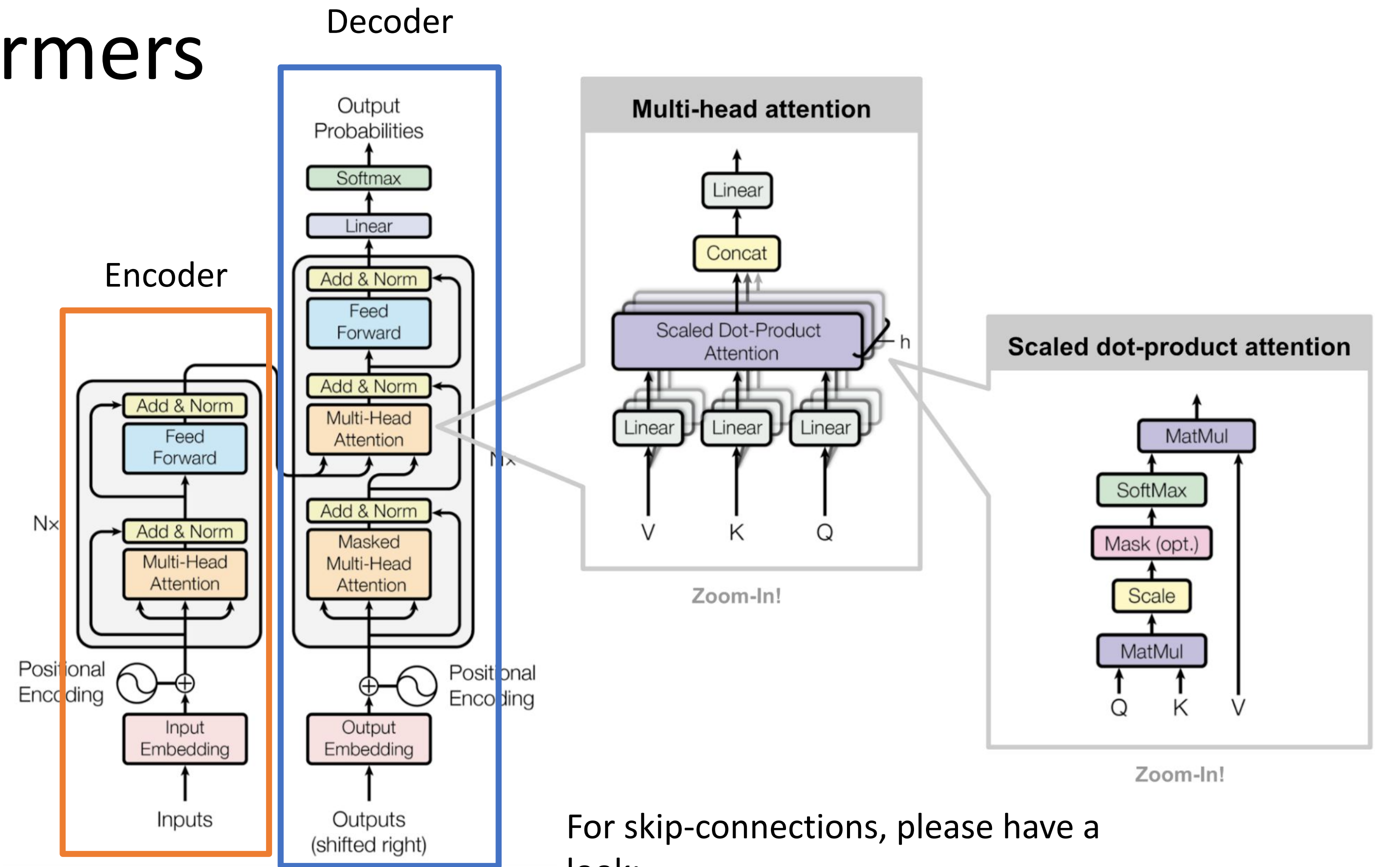
Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

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.

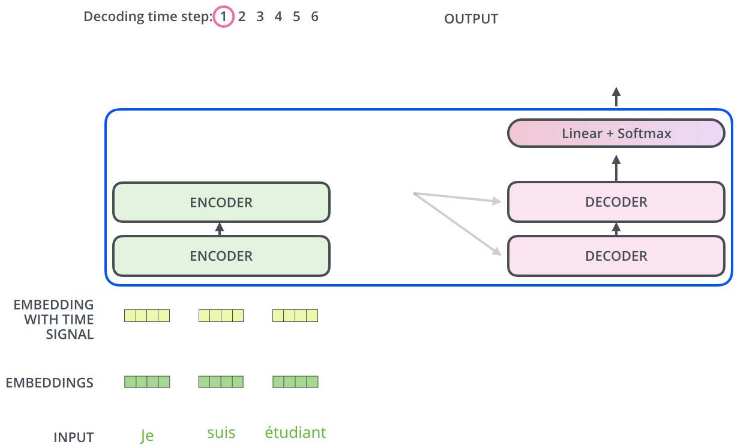
Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
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**	$2.3 \cdot 10^{19}$	

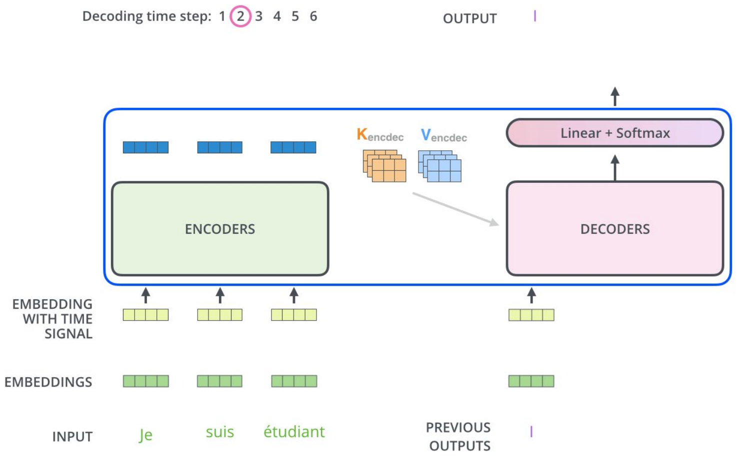
Transformers



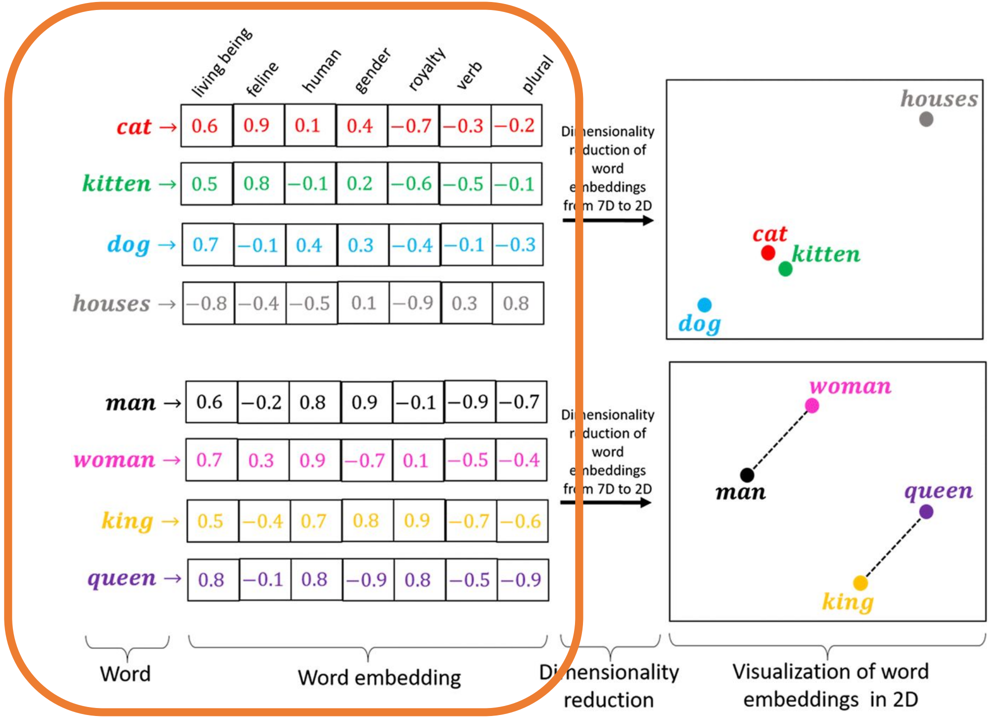
For skip-connections, please have a look: https://paperswithcode.com/method/resnet

Inputs, output and an high-level overview





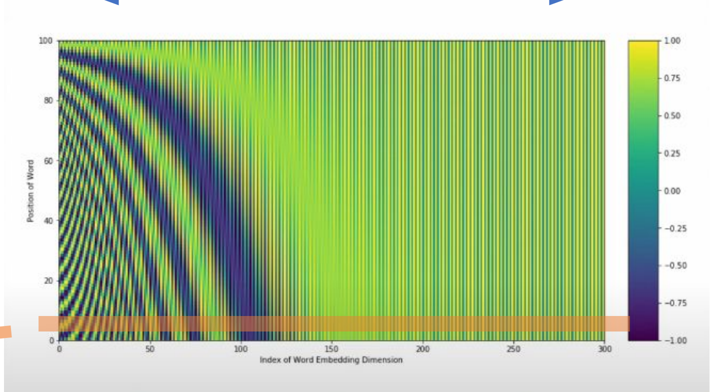
Background – word embedding



1. Positional encoding

Same length as the word vector

Same length as the sentence

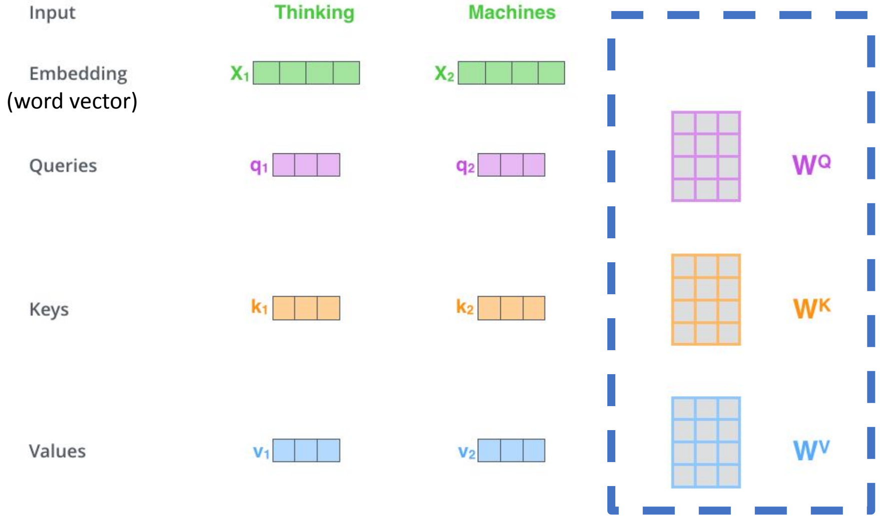


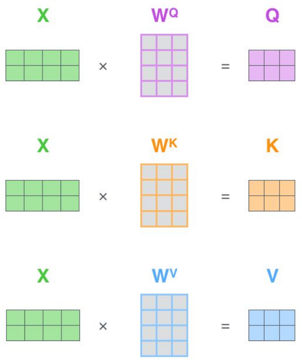
$$PE_{(pos,2i)} = sin(pos/10000^{2i/d_{\text{model}}})$$
$$PE_{(pos,2i+1)} = cos(pos/10000^{2i/d_{\text{model}}})$$

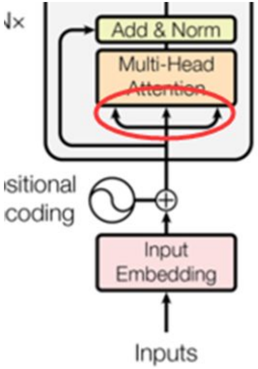
Who is that girl who is wearing a scarf?

2. Input embedding

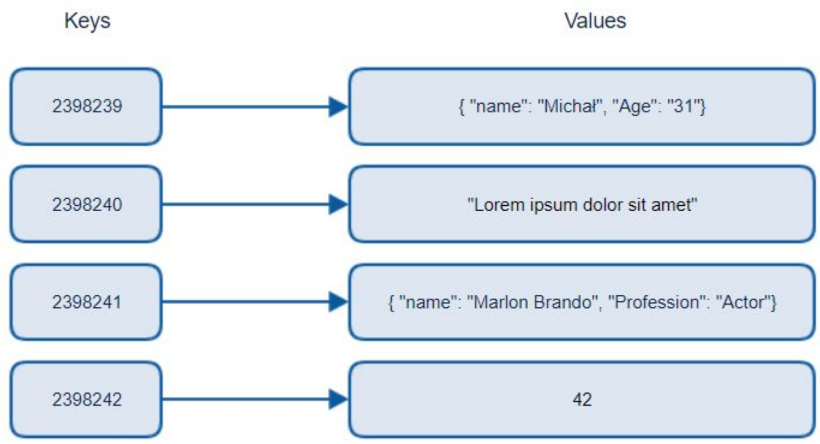
Learnable weights of the network







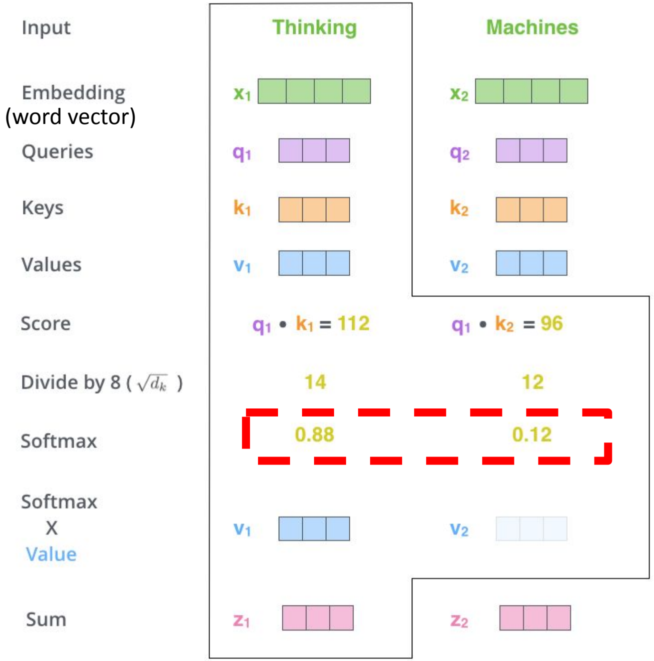
Key, value, and query in a data system



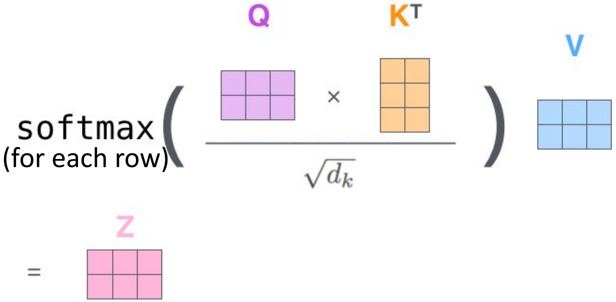
Keys: the location of the data in the database
Values: the content of the data
Queries: the request to retrieve data from the database

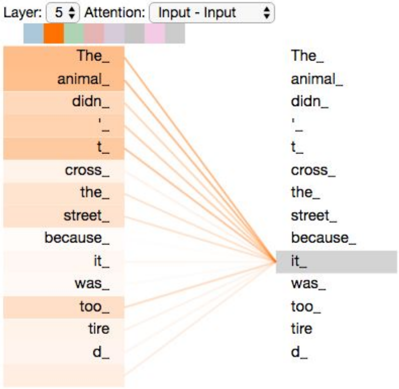
When a user send a query to a data system, the system compares the query with the key and return the corresponding value.

3. Self-attention



Attention weight, i.e. alignment scores





Encoding – context vector:
abstract representations of the context

Exercise 1

Input: [0,1,1,2,3,4,2,3,1]

Embedding method: one-hot

- E.g. 1 => (0,1,0,0,0)

W_q = 100*identity

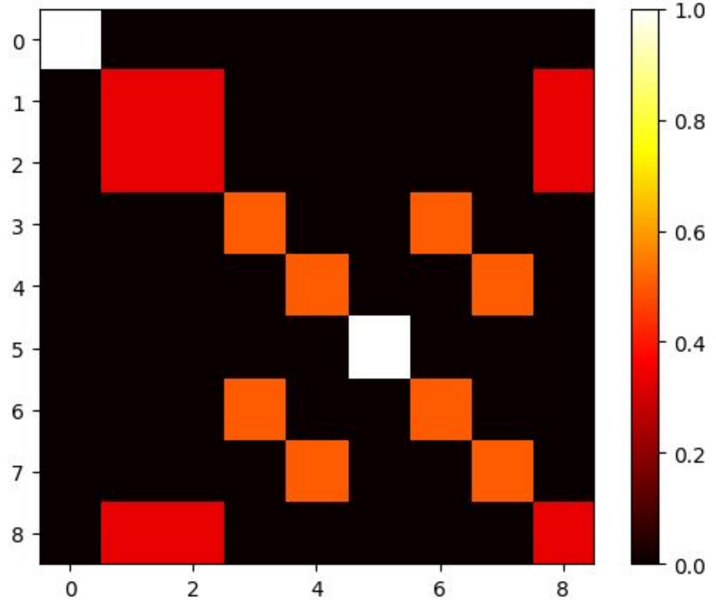
W_k = 100*identity

W_v = identity

d_k = 5

What does the attention weight matrix look like?

What is the context vector?



Exercise 2

Input: [0,1,1,2,3,4,2,3,1]

Embedding method:

0 => (1,0,0,0,0)

others => one-hot +(1,0,0,0,0)

- E.g. 1 => (1,1,0,0,0) 2=> (1,0,1,0,0,0)…

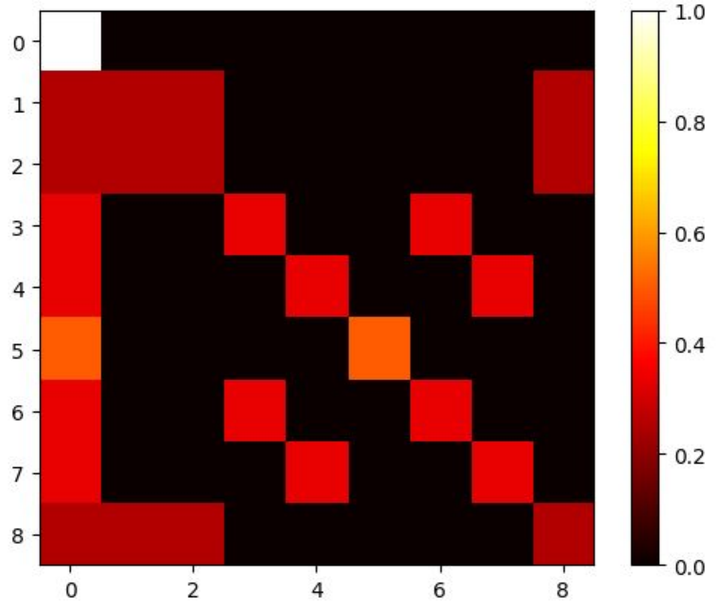
W_q = 100*identity

W_k = 100*identity

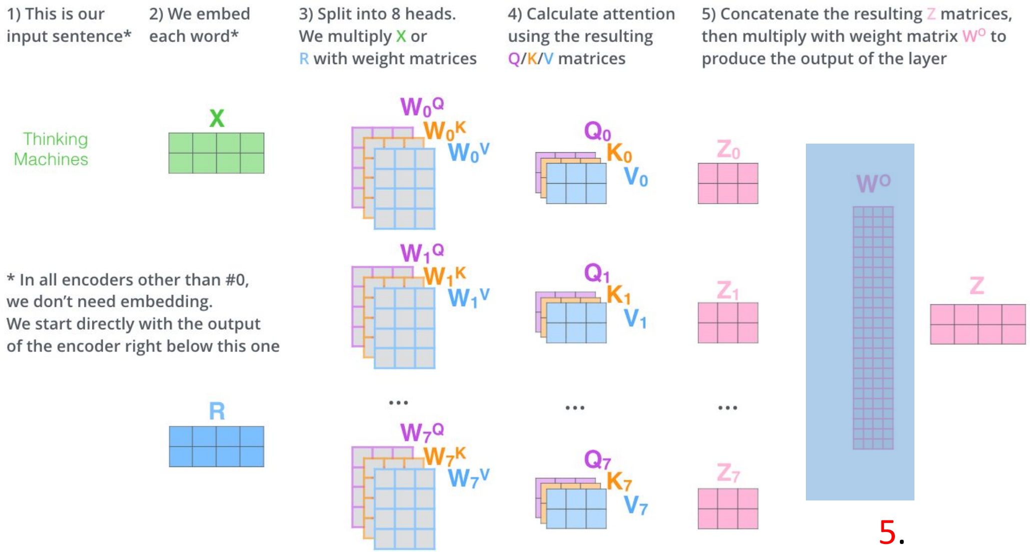
W_v = identity

d_k = 5

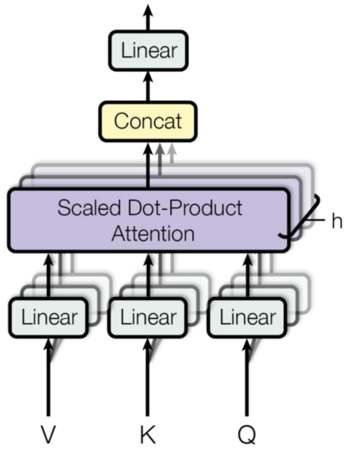
What does the attention weight matrix look like?



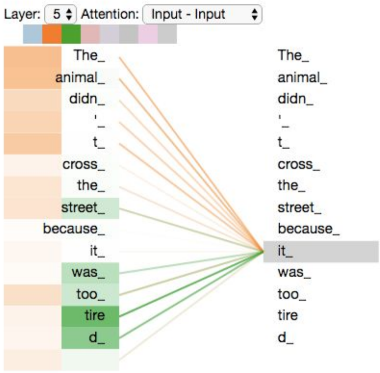
4. Multi-head attention



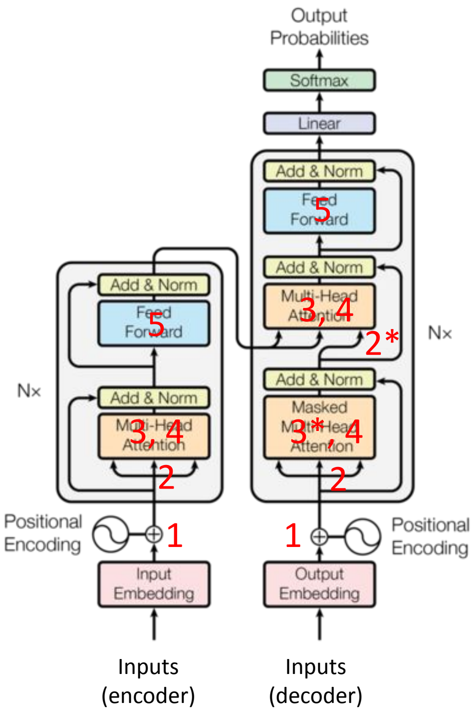
5. Feedforward module



Repeat 2 and 3 multiple times



Transformers

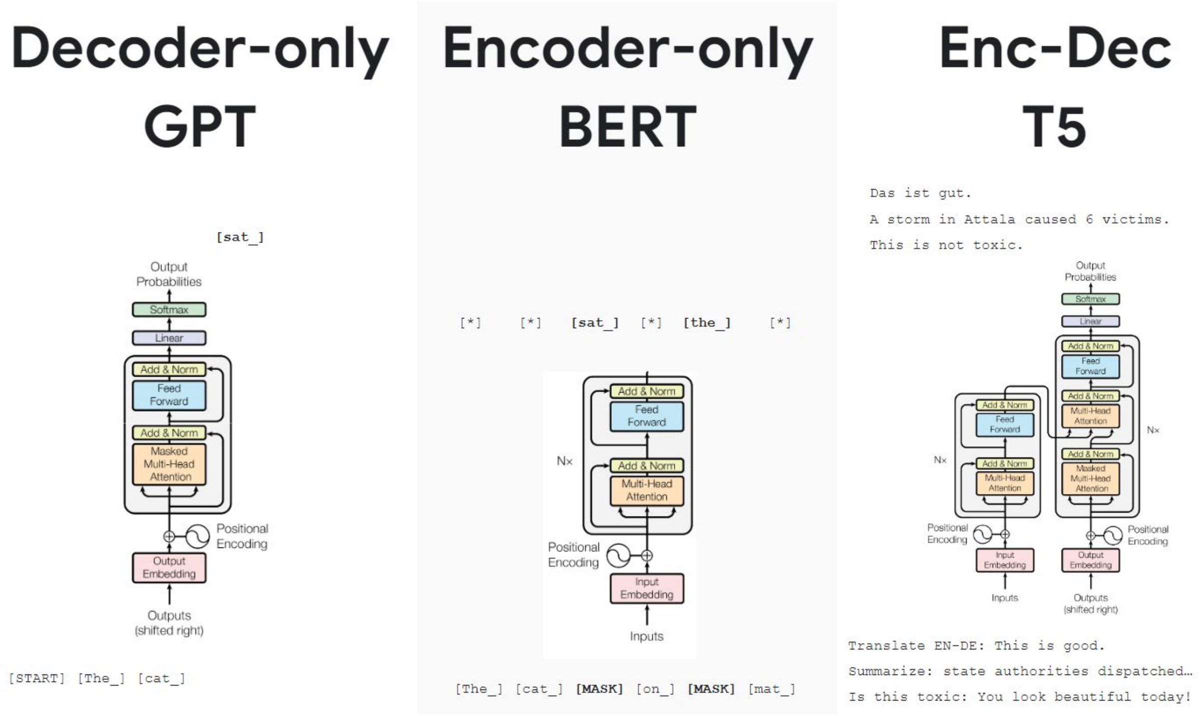


The numbers in the plot are corresponding to the mechanisms in the last few slides.

For a translation model:
Input(encoder): the **whole** sentence of language 1.
Input(decoder): the **partial** sequence of language 2.
Output: the probability distribution of the next word for the sequence of *Input(decoder).*

3*. a mask is applied to the later parts of the sentence for training efficiency.
2*. the key-value pair is learned from the encoding of *Input(encoder)*, while the query is learned from the encoding of *Input(decoder).* Therefore, this is *cross-attention* instead of self-attention.

Other architectures



Not limited to text!

Astronomy application – light curve encoding

RASTI **000**, 1–15 (2022) Preprint 29 November 2022 Compiled using rasti LATEX style file v3.0

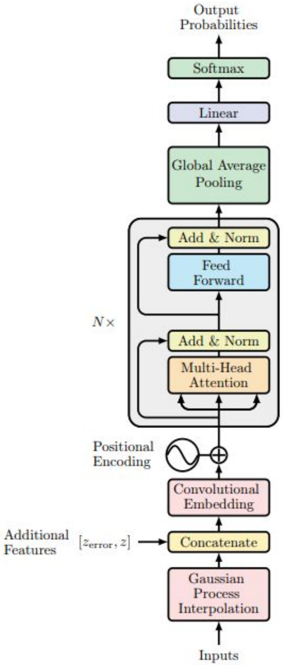
Paying Attention to Astronomical Transients: Introducing the Time-series Transformer for Photometric Classification

Tarek Allam Jr.,[1★] Jason D. McEwen[1]
[1]*Mullard Space Science Laboratory, University College London, Holmbury St Mary, Dorking, Surrey RH5 6NT, UK*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
Future surveys such as the Legacy Survey of Space and Time (LSST) of the Vera C. Rubin Observatory will observe an order of magnitude more astrophysical transient events than any previous survey before. With this deluge of photometric data, it will be impossible for all such events to be classified by humans alone. Recent efforts have sought to leverage machine learning methods to tackle the challenge of astronomical transient classification, with ever improving success. Transformers are a recently developed deep learning architecture, first proposed for natural language processing, that have shown a great deal of recent success. In this work we develop a new transformer architecture, which uses multi-head self attention at its core, for general multi-variate time-series data. Furthermore, the proposed time-series transformer architecture supports the inclusion of an arbitrary number of additional features, while also offering interpretability. We apply the time-series transformer to the task of photometric classification, minimising the reliance of expert domain knowledge for feature selection, while achieving results comparable to state-of-the-art photometric classification methods. We achieve a logarithmic-loss of 0.507 on imbalanced data in a representative setting using data from the Photometric LSST Astronomical Time-Series Classification Challenge (PLAsTiCC). Moreover, we achieve a micro-averaged receiver operating characteristic area under curve of 0.98 and micro-averaged precision-recall area under curve of 0.87.

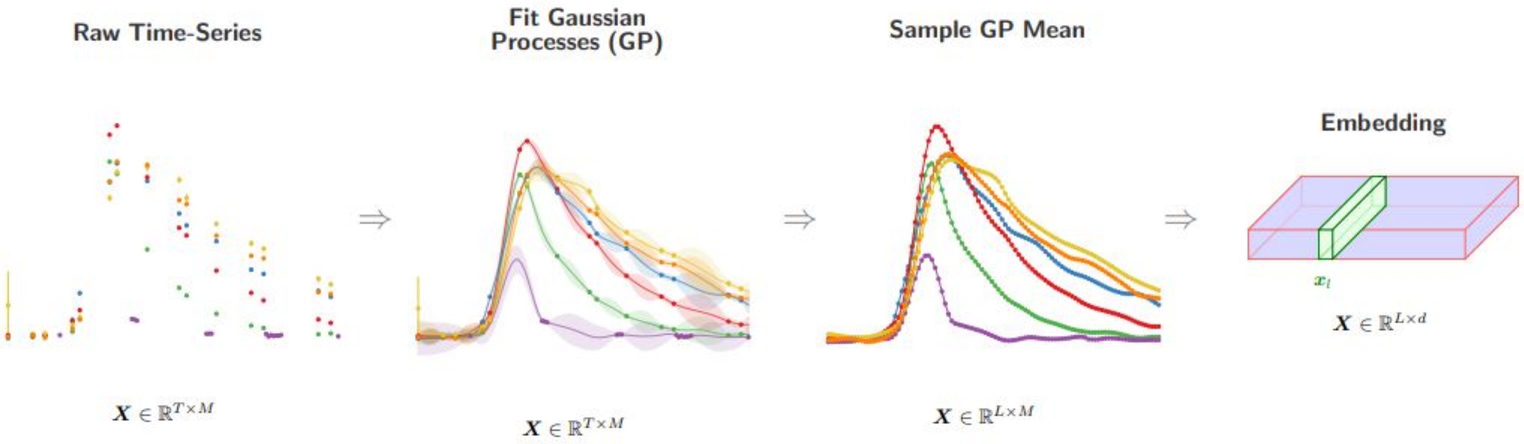
Key words: machine learning - software - data methods - time-series - transients - supernovae



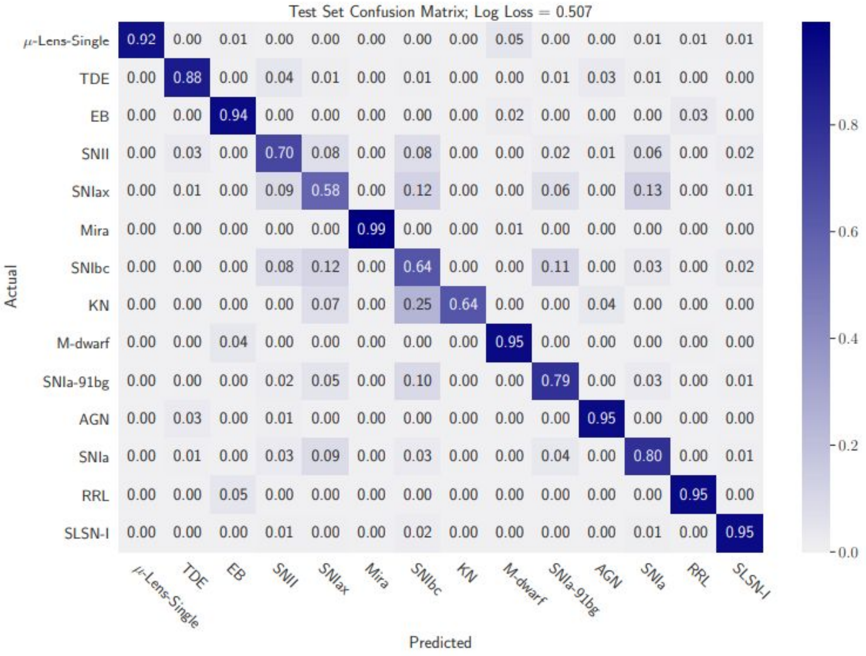
Time series transformer

Analog

- Magnitudes in different channels at a given timestep => word vector
- Multi-channel light curves => sentences



Performance in PLAsTiCC dataset



Runnable example

Toy encoder model(no positional embedding, exercises 1&2 and more):
https://colab.research.google.com/drive/1Z21lq1X_KwTr-O640k0V4IL8d47CVffL?usp=sharing

Detail transformer model:

https://github.com/harvardnlp/annotated-transformer/