Attention and BERT

Alexandre Allauzen

Fall 2023

MILES
Machine Intelligence and Learning Systems

Dauphine | PSL
UNIVERSITÉ PARIS

ESPCI PARIS | PSL

Roadmap

Introduction to attention

Transformer architecture

Conclusion

References

Outline

Introduction to attention

Transformer architecture

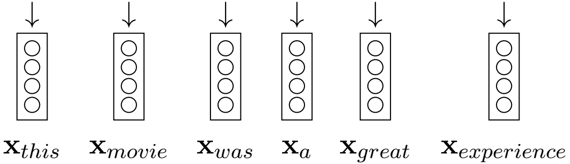
Conclusion

References

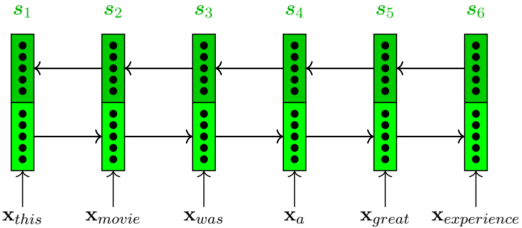
From Embeddings to Contextualized Embeddings

Static word embeddings

$$\mathbf{x} = [\ \text{this},\ \text{movie},\ \text{was},\ \text{a},\ \text{great},\ \text{experience}\ ]$$



Contextualized representation with bi-recurrent encoder:



Draw attention for classification

$$\mathbf{X} = \left.\overbrace{\begin{array}{|c|c|c|c|}\hline \; & \; & \; & \; \\ \hline \; & \; & \; & \; \\ \hline \; & \; & \; & \; \\ \hline \; & \; & \; & \; \\ \hline \; & \; & \; & \; \\ \hline \; & \; & \; & \; \\ \hline \end{array}}^{d}\right\} L \text{ words}$$

$$\times$$

$$\mathbf{q} = \begin{array}{|c|c|c|c|}\hline \; & \; & \; & \; \\ \hline \end{array} \quad : \text{ the query}$$

$$\downarrow$$

$$\text{softmax}$$

$$\downarrow$$

$$\mathbf{a} \in \mathbb{R}^L$$

- $\mathbf{a} = (a_i)$, $\sum_{i=1}^{L} a_i = 1$ and $0 \leq a_i \leq 1$
- $\mathbf{a}$: attention vector for the "query" $\mathbf{q}$ and the "keys" $\mathbf{X}$.
- $\mathbf{q}$ is a vector to be learnt [8, 5]

Attention to weight inputs

- $\mathbf{a} = X\mathbf{q}$ is the attention vector

$$\mathbf{h} = \sum_{i=1}^{L} a_i \mathbf{x}_i = \mathbf{a}\mathbf{X}$$

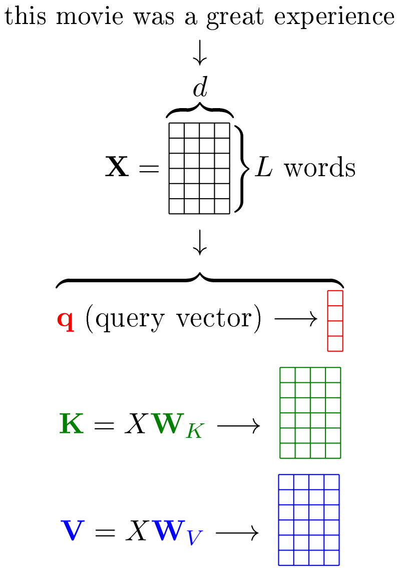
- A new vector, focused on the classification task ($\mathbf{q}$)
- To summarize:

$$\mathbf{h} = \text{softmax}(\mathbf{X}\mathbf{q})\mathbf{X} \rightarrow \text{ classification}$$

Issues:

- Scale the dot product
- X is involved everywhere !

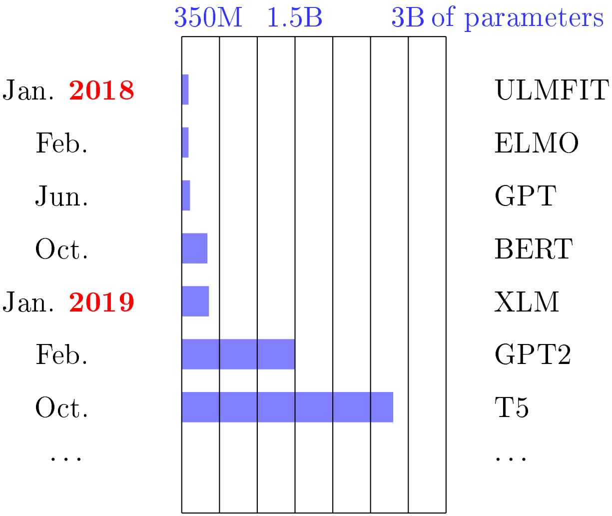
Basic attention mechanism for classification



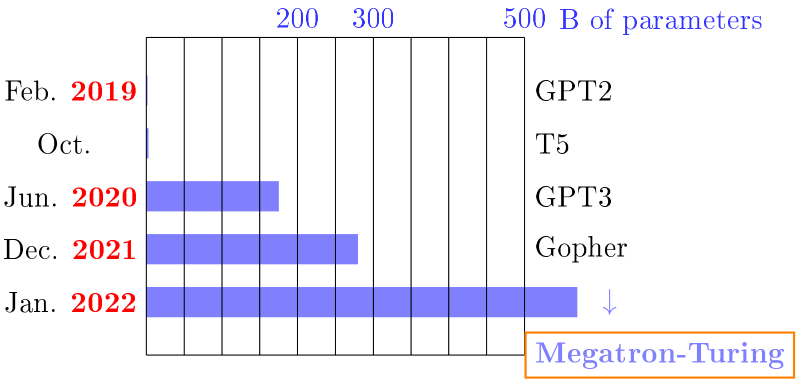
$$\mathbf{h} = \text{softmax}\left(\frac{\mathbf{K}\mathbf{q}}{\sqrt{d}}\right)^t \mathbf{V}$$

- X can be static emb.
- Derived from bi-LSTM
- $\mathbf{q}$ is learnt as a target for selection
- $pa = \mathbf{K}\mathbf{q}$: selection in $\mathbf{V}$

In a few dates



Bigger is . . .


200
300
500
B of parameters
Feb. 2019
Oct.
Jun. 2020
Dec. 2021
Jan. 2022
GPT2
T5
GPT3
Gopher
Megatron-Turing

Outline

Introduction to attention

Transformer architecture

Conclusion

References

Contextualized word embeddings

Consider the word driver:

the	audio	driver	is	really	outdated
the	driver	exceeded	the	speed	limit

Contextualized word embeddings

Consider the word driver:

the	audio	driver	is	really	outdated
the	driver	exceeded	the	speed	limit

The context

The	The	$\lambda_{1,2}$
audio	driver	$\lambda_{2,2}$
driver	exceeded	$\lambda_{3,2}$
is	the	$\lambda_{4,2}$
really	speed	$\lambda_{5,2}$
outdated	limit	$\lambda_{6,2}$

Self attention

Consider the word driver:

	the	driver	exceeded	the	speed	limit
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
Embeddings:	$\mathbf{x}_1$	$\mathbf{x}_2$	$\mathbf{x}_3$	$\mathbf{x}_4$	$\mathbf{x}_5$	$\mathbf{x}_6$
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
Attention:	$\lambda_{1,2}$	$\lambda_{2,2}$	$\lambda_{3,2}$	$\lambda_{4,2}$	$\lambda_{5,2}$	$\lambda_{6,2}$

Output:

$$\underbrace{\lambda_{1,2} \quad \lambda_{2,2} \quad \lambda_{3,2} \quad \lambda_{4,2} \quad \lambda_{5,2} \quad \lambda_{6,2}}_{\mathbf{z}_2 = f\left(\sum_i [\lambda_{i,2} \times g(\mathbf{x}_i)]\right)}$$

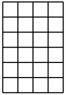
- $(\lambda_{i,j})$ are the attention coefficients, $\sum_i \lambda_{i,j} = 1$, and
- Reflects the influence of $\mathbf{x_i}$ on $\mathbf{x_j}$ (transformed version)

Transformer : Queries, Keys, Values

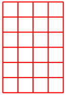
the driver exceeded the speed limit

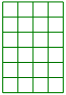
$\downarrow$

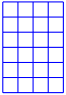
d

$\mathbf{X} =$  L words

$\downarrow$

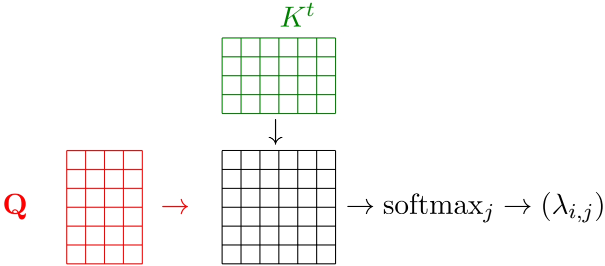
$\mathbf{Q} = X\mathbf{W}_Q \longrightarrow$ 

$\mathbf{K} = X\mathbf{W}_K \longrightarrow$ 

$\mathbf{V} = X\mathbf{W}_V \longrightarrow$ 

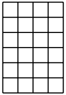
Tranformer : Attention matrix

The distance matrix between Q and K



Scaled Dot-Product Attention

$$\mathbf{Z} = \mathrm{softmax}\left(\frac{\mathbf{QK^t}}{\sqrt{d}}\right)\mathbf{V} =$$



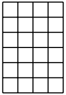
QKV and Metric Learning

$$\mathbf{Q}\mathbf{K}^t = \mathbf{X}\mathbf{W}_K \times (\mathbf{X}\mathbf{W}_K)^t = \mathbf{X}\mathbf{W}_Q \times (\mathbf{W}_K^t\mathbf{X}^t)$$
$$= \mathbf{X}\mathbf{M}\mathbf{X}^t$$

- If $\mathbf{M}$ would be PSD, it is a metric.
- Otherwise, it is a transformed similarity (bilinear similarity)

$\mathbf{M}$ is learnt: a transformer block learns its own similarity.

Multi-head attention (with 2 heads)

$\mathbf{X} =$ 

$$\downarrow$$

$$\mathbf{W}_Q^{(1)}, \mathbf{W}_K^{(1)}, \mathbf{W}_V^{(1)} \qquad \mathbf{W}_Q^{(2)}, \mathbf{W}_K^{(2)}, \mathbf{W}_V^{(2)}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\mathbf{Q}^{(1)}, \mathbf{K}^{(1)}, \mathbf{V}^{(1)} \qquad \mathbf{Q}^{(2)}, \mathbf{K}^{(2)}, \mathbf{V}^{(2)}$$

$$\downarrow \qquad\qquad \downarrow$$

$$\mathbf{Z}^{(1)} \qquad\qquad \mathbf{Z}^{(2)}$$

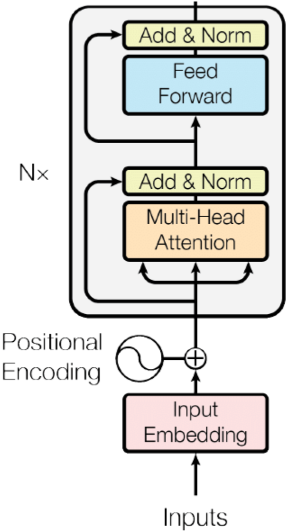
$$\mathbf{Z} = (\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)})\mathbf{W_o}$$

Putting all together (with more tricks)

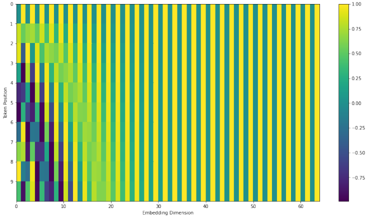
Transformer block

From [7]

- Inputs is $\mathbf{X}$
- Positional embeddings
- Multihead attention
- Residual connections [4]
- Layer Normalization [2]
- Final filtering



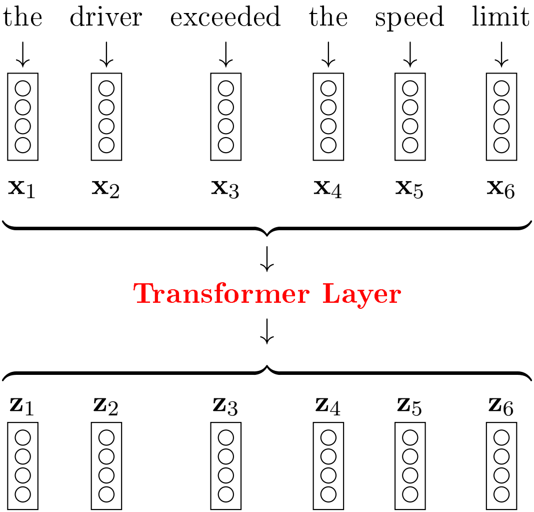
Positional embeddings



- Originally "absolute"
- Can be learnt [3, 1]
- Or relative [6]

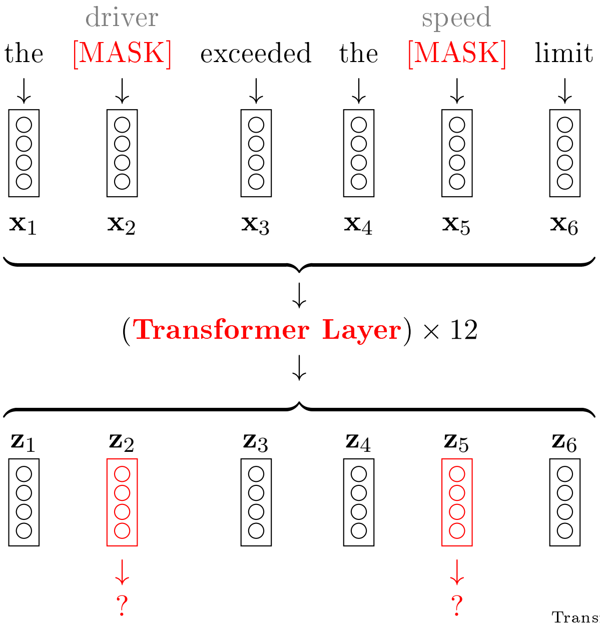
(figure generated by the following code `https://github.com/jalammar/jalammar.github.io/blob/master/notebookes/transformer/transformer_positional_encoding_graph.ipynb`)

A Transformer layer

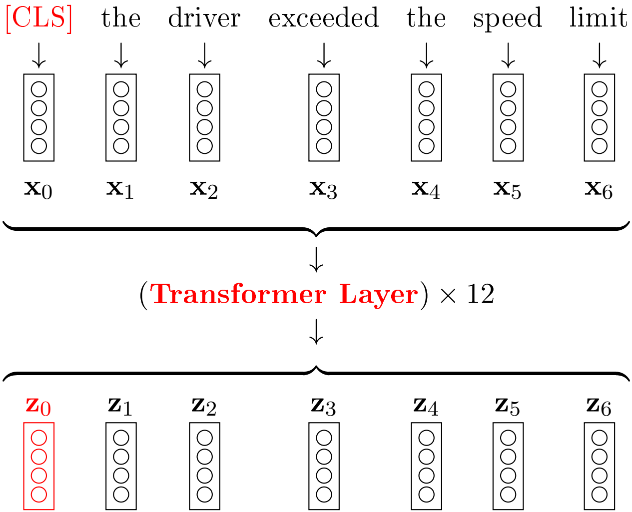


Transformer layers can be stacked !

Pre-training as a (Masked) language model

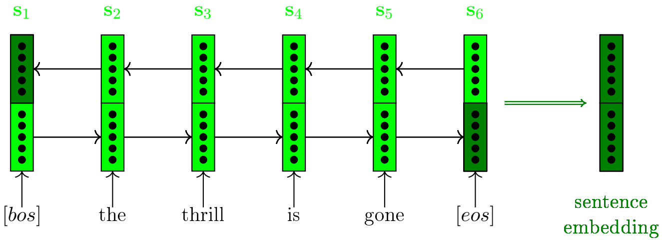


BERT Encoder for text classification



Transformers / bi-lstm encoders

Reminder of bi-recurrent encoder



The difference

- Two different ways to encode the dependence
- Richer for attention since we stack transformers
- all the deep-learning tricks $\Rightarrow$ over-parametrization

Outline

Introduction to attention

Transformer architecture

Conclusion

References

Transformers are everywhere

State of the art encoder

- For text !
- And also for speech, DNA, vision, ...

Also a powerful generator

- For text (GPT, ...)
- Speech, ... sequences

Outline

Introduction to attention

Transformer architecture

Conclusion

References

[1] Rami Al-Rfou et al. *Character-Level Language Modeling with Deeper Self-Attention.* 2018. arXiv: 1808.04444 [cs.CL].

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization.* 2016. arXiv: 1607.06450 [stat.ML].

[3] Jonas Gehring et al. "Convolutional Sequence to Sequence Learning". In: *CoRR* abs/1705.03122 (2017). arXiv: 1705.03122. URL: http://arxiv.org/abs/1705.03122.

[4] Kaiming He et al. "Deep Residual Learning for Image Recognition". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 770–778. arXiv: 1512.03385 [cs.CV].

[5] Zhouhan Lin et al. "A STRUCTURED SELF-ATTENTIVE SENTENCE EMBEDDING". In: *International Conference on Learning Representations.* 2017. URL: https://openreview.net/forum?id=BJC_jUqxe.

[6] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. *Self-Attention with Relative Position Representations.* 2018. arXiv: 1803.02155 [cs.CL].

[7] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 6000–6010. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

[8] Zichao Yang et al. "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL).*