Structure, Attention, and BERT

Alexandre Allauzen

Fall 2024

Roadmap

Outline

The simplest classifier based on word embeddings

Input text is now a sequence of vectors (embeddings)

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

Derive a vector of features that represent the text

$$\mathbf{h} = \sum_{i=1}^{L} \underbrace{\mathbf{x}_i}_{\text{emb. of word } i}$$

Classification

$$\text{softmax}(\mathbf{W}^o\mathbf{h}) \text{ (multiclass) or } \sigma(\mathbf{w}^o\mathbf{h}) \text{ (binary)}$$

Limitations of BOW classifier

$$\mathbf{h} = \sum_{i=1}^{L} \underbrace{\mathbf{x}_i}_{\text{emb. of word } i}$$

Limitations

- Words are equally important
- Word order independent
- Miss contextual information (local/global)

Local contexts

the	end	is	very	bad	but	what	a	great	music

Local contexts

the	end	is	very	bad	but	what	a	great	music
			$\underbrace{\text{very} \quad \text{bad}}_{very \rightarrow bad++}$						

Local contexts

the	end	is	very	bad	but	what	a	great	music
			$\underbrace{\hphantom{very\ bad}}_{very \to bad++}$						
			$\underbrace{\hphantom{very\ bad\ but}}_{\textit{but}\text{ will change }\textit{bad}}$						

Local contexts

the	end	is	very	bad	but	what	a	great	music
			very→*bad*++						
			but will change *bad*						
		bad is for *end* not *music*						*great* is for *music* not fo *end*	

Motivations

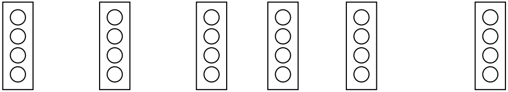
- Local contextualisation
- Global view of the sentence

Another view of a sentence

$=$ [this, movie, was, a, great, experience]

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

Look-up



$\mathbf{v}_{this} \quad \mathbf{v}_{movie} \quad \mathbf{v}_{was} \quad \mathbf{v}_{a} \quad \mathbf{v}_{great} \quad \mathbf{v}_{experience}$

$$\downarrow$$

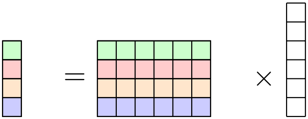
$$\begin{bmatrix} 1.7 & -0.3 & -0.5 & -2.7 & -0.0 & -0.3 \\ -0.5 & 0.3 & 0.4 & -1.1 & -0.9 & -0.5 \\ 0.7 & 0.6 & -1.3 & -1.1 & 0.7 & 1.6 \\ -0.0 & -0.7 & 1.1 & -0.3 & 0.7 & 1.5 \end{bmatrix}$$

Propose 2 solutions for an improved text classification

...

Matrix and Vector product

$$\mathbf{y} = \mathbf{W} \times \mathbf{x}$$



$$\begin{aligned} y_1 &= \mathbf{W}_{1,:} \times \mathbf{x} \\ y_2 &= \mathbf{W}_{2,:} \times \mathbf{x} \\ \dots & \quad \dots \quad \quad \dots \end{aligned}$$

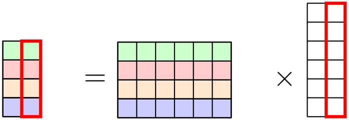
In terms of dimension:

$$\text{With} \begin{cases} \mathbf{x} & : (L_1 \times 1) \\ \mathbf{W} & : (L_2 \times C_2) \\ \boldsymbol{y} & : (L_2 \times 1) \end{cases} \Rightarrow (\mathbf{Wx}) : (L_2 {\times} 1) = (L_2 {\times} \underbrace{\cancel{C_2)(L_1}}_{C_2 = L_1} {\times} 1)$$

Matrix-matrix product

$\boldsymbol{X}$ is a matrix of 2 columns, 2 vectors as $\mathbf{x}$:

$$\boldsymbol{Y} \quad = \mathbf{W} \qquad\qquad \times\ \boldsymbol{X}$$



In terms of dimension:

$$\text{With} \begin{cases} \boldsymbol{X} & : (L_1 \times C_1) \\ \mathbf{W} & : (L_2 \times C_2) \\ \boldsymbol{y} & : (L_2 \times C_1) \end{cases} \rightarrow (L_2 \times C_2) = (L_2 \times \underbrace{\cancel{C_2})(\cancel{L_1}}_{C_2=L_1} \times C_1)$$

Outline

Draw attention for classification

Remind CBOW classifier

The classifier output:

$$\text{softmax}(\mathbf{W}^o\mathbf{h}) \text{ (multiclass) or } \sigma(\mathbf{w}^o\mathbf{h}) \text{ (binary)}$$

- What does represent a row of $\mathbf{W}^o$?
- The product $\mathbf{W}^o\mathbf{h}$?
- The softmax ?

Draw attention

Is a word vector related to the classification task ?

$$\mathbf{h} = \sum_{i=1}^{L} \underbrace{\mathbf{x}_i}_{\text{emb. of word } i} \longrightarrow \mathbf{h} = \sum_{i=1}^{L} \underbrace{\lambda_i}_{???} \mathbf{x}_i$$

Draw attention for classification (binary task)

$$\mathbf{X}\mathbf{q} = L\left\{\begin{array}{|c|c|c|c|}\hline \; & \; & \; & \; \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline & & & \\ \hline \end{array}\right. \times \begin{array}{|c|c|c|c|}\hline \; & \; & \; & \; \\ \hline \end{array} = \begin{array}{|c|}\hline \; \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \\ \hline \end{array} \in \mathbb{R}^L$$

$$(\mathbf{X}\mathbf{q})_i = \mathbf{x}_i^t \mathbf{q} \quad \text{(dot product)}$$

$$\mathbf{a} = \mathrm{softmax}(\mathbf{X}\mathbf{q})$$

- $\mathbf{a} = (a_i)$, $\sum_{i=1}^{L} a_i = 1$ 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 [11, 7]

Attention to weight inputs (binary task)

- $\mathbf{a} = \text{softmax}(\mathbf{X}\mathbf{q})$ is the attention vector

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

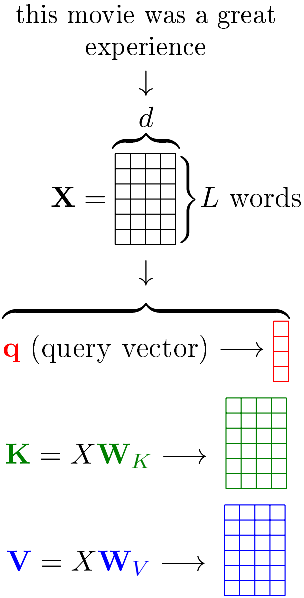
- A new vector, focused on the classification task ($\mathbf{q}$)
- To summarize:

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

Issues:

- Scale the dot product
- X is involved everywhere !

Basic attention mechanism for classification (binary task)



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

- X can be static emb.
- or contextualized embedding
- $\mathbf{q}$ is learnt as a target for selection
- $\mathbf{a} = \mathbf{Kq}$: selection in $\mathbf{V}$

Attention classifier: Going to multiclass

Exercise

- How to modify (parametrize) the model for multiclass classification ?
- Can we add more transformations ?

Outline

Contextualized word embeddings

Consider the word driver:

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

The context

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

Self attention: a first idea

Look at the "correlation" between words (embeddings)

- $\mathbf{X}\mathbf{X}^t$ is a $L \times L$ matrix, stores $(\mathbf{x}_i^t \mathbf{x}_j)$
- The i^{th} row stores the "correlation between" $\mathbf{x}_i$ and all the other words in the sentence
- For $i = 2$, we have the correlations with driver
- We can use this correlation as a weight

$$\mathbf{z}_2 = \mathbf{z}_{driver} = \sum_{j=1}^{L} \underbrace{\lambda_{2,j}}_{\mathbf{x}_2^t \mathbf{x}_j} \mathbf{x}_j$$

More (linear) transformations

Two different Transformations on $\mathbf{X}$

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

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

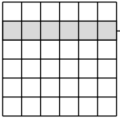
- with $\mathbf{W}_Q$ and $\mathbf{W}_K \in \mathbb{R}^{d \times d}$
- $\mathbf{Q}$ and $\mathbf{K}$ have the same dimensions as $\mathbf{X}$

$$\mathbf{A} = \mathbf{Q}\mathbf{K}^t = \underbrace{(\mathbf{Q}_{i,*}\mathbf{K}^t_{j,*})_{i,j}}_{L \times L} = (\mathbf{q}_i^{\mathbf{k}j}) = (\lambda_{i,j}),$$

with $\lambda_{i,j}$ the attention on "word" j to generate $\mathbf{z}_i$

Normalization of attention

Take the row-wise softmax:



$\longrightarrow \mathrm{softmax}_j(\mathbf{A}_{i,*})$

$$\sum_j \underbrace{\lambda_{i,j}}_{\text{or } a_{i,j}} = 1 \text{ and } \lambda_{i,j} \geq 0$$

Each row of $\mathbf{A}$ gives a convex combination

Self attention (overview)

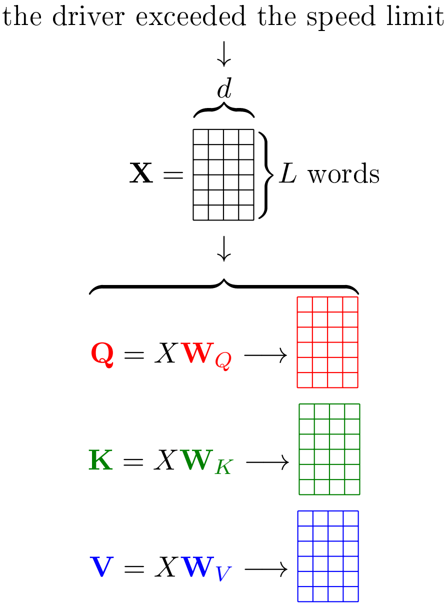
Consider the word driver:

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

Output: $\mathbf{z}_2 = f\left(\sum_j [\lambda_{2,j} \times g(\mathbf{x}_j)]\right)$

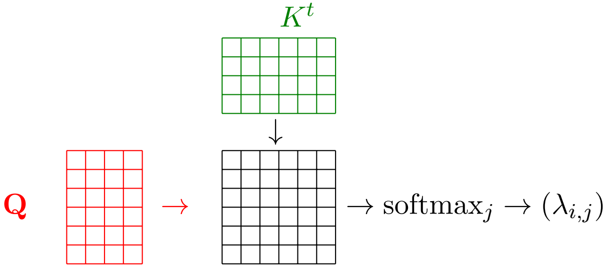
- $(\lambda_{i,j})$ are the attention coefficients, $\sum_j \lambda_{i,j} = 1$, and
- Reflects the influence of $\mathbf{x_j}$ on $\mathbf{x_i}$ (transformed version)

Transformer : Queries, Keys, Values



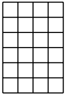
Tranformer : Attention matrix

The distance matrix between Q and K



Scaled Dot-Product Attention

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



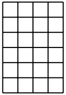
Q,K,V and Metric Learning

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

- 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)

$\mathbf{X} =$ 

$$\downarrow$$

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

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

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

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

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

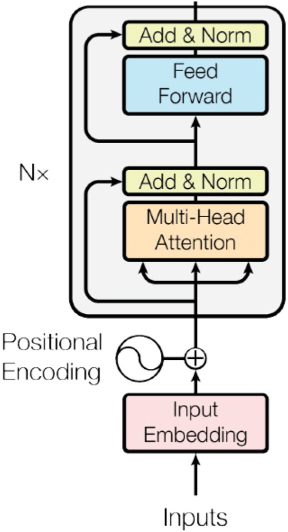
$$\mathbf{Z} = (\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)})\mathbf{W_o}$$

Putting all together (with more tricks)

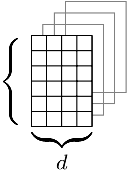
Transformer block

From [10]

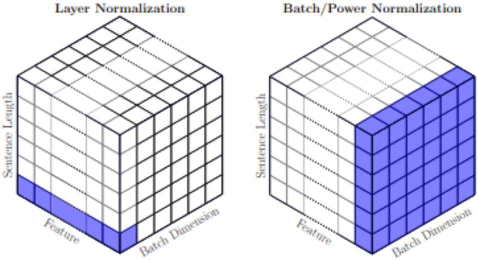
- Inputs is $\mathbf{X}$
- Positional embeddings
- Multihead attention
- Residual connections [6]
- Layer Normalization [2]
- Final filtering



Layer norm

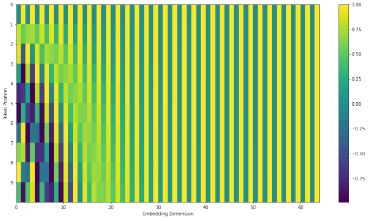
Assume $\mathbf{Z}$ a minibatch of sequences (B, L, D): $\mathbf{Z} = L$ 

Batch or Layer norm



[9]

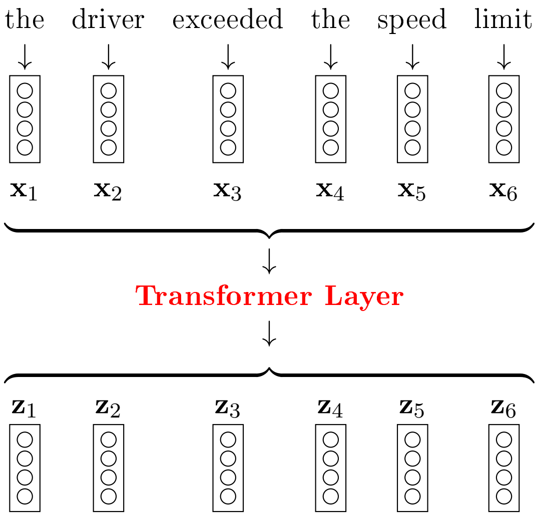
Positional embeddings



- Originally "absolute"
- Can be learnt [5, 1]
- Or relative [8]

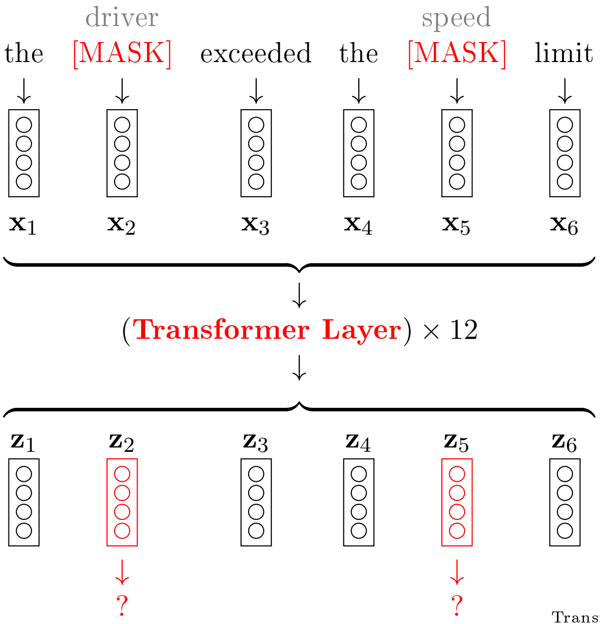
(figure generated by the following code
`https://github.com/jalammar/jalammar.github.io/blob/master/notebookes/transformer/transformer_positional_encoding_graph.ipynb`)

A Transformer layer

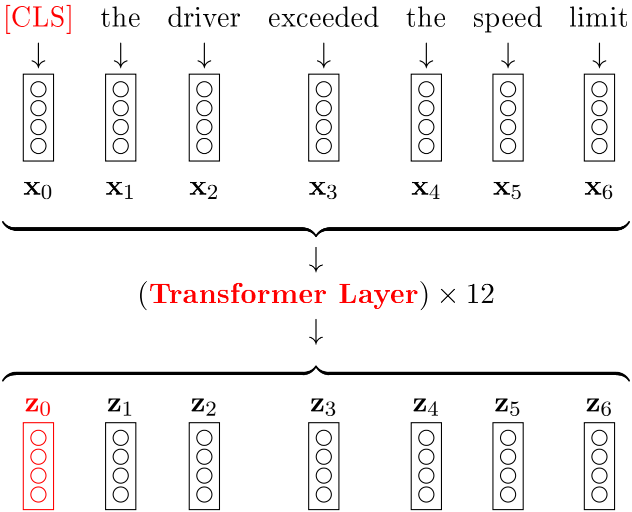


Transformer layers can be stacked !

Pre-training as a (Masked) language model



BERT Encoder for text classification



Outline

Summary

Attention, attention

- This mechanism allows the model to efficiently handle different kind of structure.
- Originally for machine translation, and with BI-GRU [4, 3].

Transformers

- Architecture proposed in [10]
- Nowadays state of the art component

Transformers are everywhere

State of the art encoder

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

Also a powerful generator

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

Outline

[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] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473 (2014). URL: http://arxiv.org/abs/1409.0473.

[4] Kyunghyun Cho et al. "Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1724–1734. URL: http://www.aclweb.org/anthology/D14-1179.

[5] 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.

[6] 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].

[7] 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.

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

[9] Sheng Shen et al. "PowerNorm: Rethinking Batch Normalization in Transformers". In: *Proceedings of the 37th International Conference on Machine Learning*. Ed. by Hal Daumé III and Aarti Singh. Vol. 119. Proceedings of Machine Learning Research. PMLR, 2020, pp. 8741–8751. URL: https://proceedings.mlr.press/v119/shen20e.html.

[10] 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.

[11] Zichao Yang et al. "Hierarchical Attention Networks for Document Classification". In: *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)06*. 2016.