Text, word embeddings, transformers

Alexandre Allauzen

Winter 2025

Machine Intelligence and Learning Systems

Roadmap

Outline

Text classification/rating

my wonderful friend took
me to see this movie
for our anniversary.
it was terrible.

→ ☹ : 0 ⋯ 1 : ☺

- How to represent the input text ?
- How to make classification ?

Bag of words (BOW)

this movie is just great , with a great music , while a bit long

Bag of words (BOW)

this movie is just great , with a great music , while a bit long

vocabulary	binary bag	count bag	tf.idf bag	...
awesome	0	0	0	...
great	1	2	1.9	...
long	1	1	2.5	...
the	0	0	0	...
this	1	1	0.1	...

A basic vectorial representation of text

$$\mathbf{x} = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^D \qquad \left.\begin{array}{r} \textit{awesome} \\ \textit{great} \\ \textit{long} \\ \textit{the} \\ \textit{this} \end{array}\right\} D$$

A simple problem

Assumptions

- Let define a finite set of known words: the vocabulary $\mathcal{V}$
- A text is a vector $\mathbf{x}$ of dimension $D = |\mathcal{V}|$
- Each component encodes the presence of a word

Then machine learning

- Naive Bayes
- SVM, Random Forrest, ...
- Logistic Regression

Outline

Back to logistic regression

$$\mathbf{x} = \begin{pmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 1 \end{pmatrix} \in \mathbb{R}^D \qquad \left.\begin{array}{r} \textit{awesome} \\ \textit{great} \\ \textit{long} \\ \textit{the} \\ \textit{this} \end{array}\right\} D$$

For one input text:

$$w_0 + \mathbf{w}^t\mathbf{x} = w_0 + 2 \times w_2 + w_3 + w_5$$

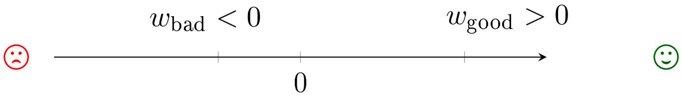
The class is positive $(y = 1)$ if

$$w_0 + 2 \times w_2 + w_3 + w_5 > 0$$

$$2 \times w_{great} + w_{long} + w_{this} + > -w_0$$

A limited representation of words

With the logistic regression model on a bag of words:



Consider the two following examples:

the end is **really** **bad**	☹	$\Rightarrow$ $w_{\text{bad}} \searrow$
the **bad** guy is *awesome*	☺	$\Rightarrow$ $w_{\text{bad}} \searrow, w_{\text{awesome}} \nearrow$

Multiple dimensions could help to:

- represent different usage
- consider the context
- leverage more from sparse, sometime ambigous observations.

A simple model for document classification - part 1

Idea

- The word representation could be shared among classes
- While their interpretation depends on the class

Input representation and composition

$$\mathbf{R} \times \mathbf{x} = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 & \mathbf{v}_4 & \mathbf{v}_5 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix} \times \begin{pmatrix} 0 \\ \mathbf{2} \\ \mathbf{1} \\ 0 \\ \mathbf{1} \end{pmatrix} = 2 \times \mathbf{v}_2 + \mathbf{v}_3 + \mathbf{v}_5 = \mathbf{d}$$

A simple model for document classification - part 2

Classification

$$
\begin{aligned}
P(y|\mathbf{x}) &= \text{softmax}(\mathbf{W^o d}) = \text{softmax}(\mathbf{W^o} \times \mathbf{Rx}), \text{ or} \\
&= \text{softmax}(\mathbf{W^o} \times f(\mathbf{Rx})),
\end{aligned}
$$

with f a non-linear activation function.

Parameters

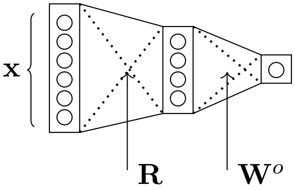
$$\boldsymbol{\theta} = (\mathbf{R}, \mathbf{W^o}) \rightarrow \textbf{to learn !!}$$

Reminder

If $\mathbf{y} = \text{softmax}(\mathbf{a})$, $\mathbf{y}$ is a vector and $\mathbf{a}$ is called the logit vector

$$y_i = \frac{e^{a_i}}{\sum_j e^{a_j}}$$

A first neural network



$$P(y = 1|\mathbf{x};\boldsymbol{\theta}) = \sigma(\mathbf{W}^o \times \mathbf{d})$$

- $\mathbf{x}$: $(|\mathcal{V}|, 1)$
- $\mathbf{R}$: $(K, |\mathcal{V}|)$
- $\mathbf{d}$: $(K, 1)$ $\qquad \mathbf{d} = \mathbf{R} \times \mathbf{x}$
- $\mathbf{W^o}$: $(1, K)$
- y : $(1, 1)$ $\qquad y = \sigma(\mathbf{W^o} \times \mathbf{d})$

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)

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

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

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

- $\mathbf{a} = (a_i)$, $\sum_{i=1}^{L} a_i = 1$ and $0 \le a_i \le 1$
- $\mathbf{a}$: 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

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

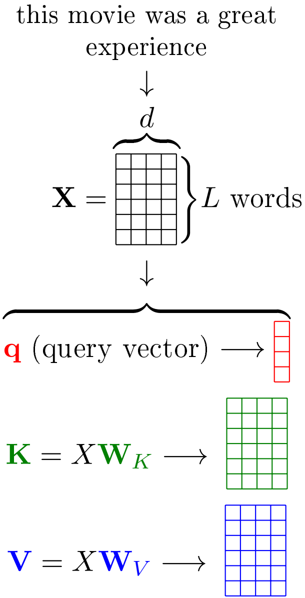
- A new vector, focused on the classification task ($\mathbf{q}$)
- To summarize:

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

Issues:

- Scale the dot product
- X is involved everywhere !

Basic attention mechanism for classification (binary task)



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

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

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 $\mathbf{X}$

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

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

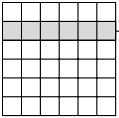
- with $\mathbf{W}_Q$ and $\mathbf{W}_K \in \mathbb{R}^{d\times d}$
- $\mathbf{Q}$ and $\mathbf{K}$ have the same dimensions as $\mathbf{X}$

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

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:

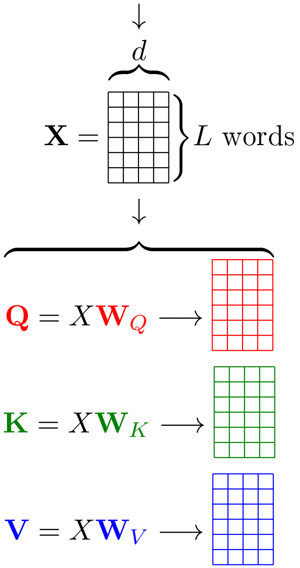
$$\begin{array}{lcccccc}
 & \text{the} & \text{driver} & \text{exceeded} & \text{the} & \text{speed} & \text{limit} \\
 & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{Embeddings:} & \mathbf{x}_1 & \mathbf{x}_2 & \mathbf{x}_3 & \mathbf{x}_4 & \mathbf{x}_5 & \mathbf{x}_6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

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

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

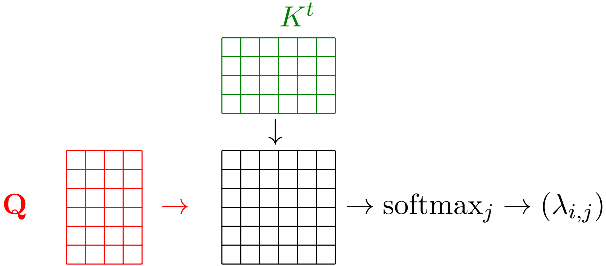
Transformer : Queries, Keys, Values

the driver exceeded the speed limit



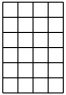
Tranformer : Attention matrix

The distance matrix between Q and K

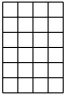


Scaled Dot-Product Attention

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



Multi-head attention (with 2 heads)

$$\mathbf{X} =$$ 

$$\downarrow$$

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

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

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

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

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

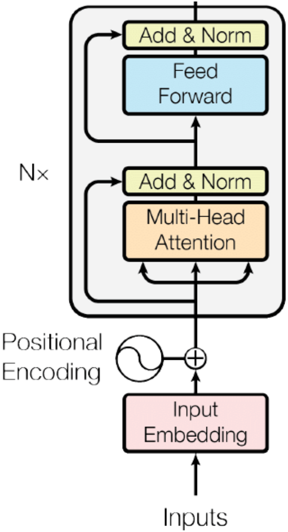
$$\mathbf{Z} = (\mathbf{Z}^{(1)}; \mathbf{Z}^{(2)})\mathbf{W_o}$$

Putting all together (with more tricks)

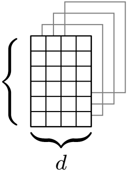
Transformer block

From [10]

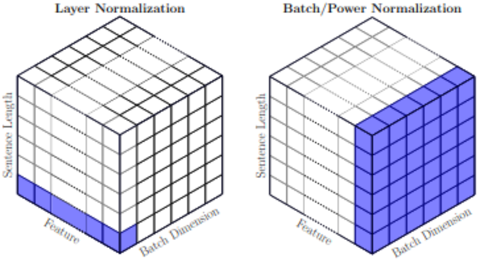
- Inputs is $\mathbf{X}$
- Positional embeddings
- Multihead attention
- Residual connections [6]
- Layer Normalization [2]
- Final filtering



Layer norm

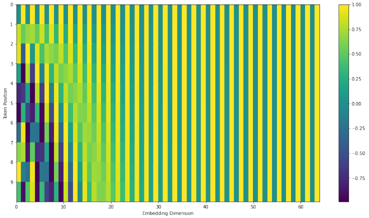
Assume $\mathbf{Z}$ a minibatch of sequences (B, L, D): $\mathbf{Z} = L$ 

Batch or Layer norm



[9]

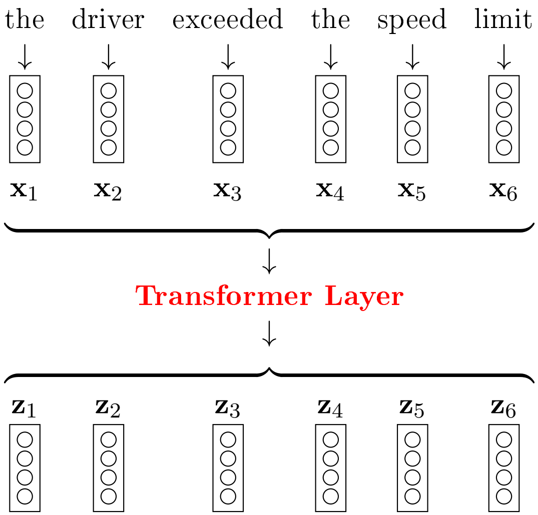
Positional embeddings



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

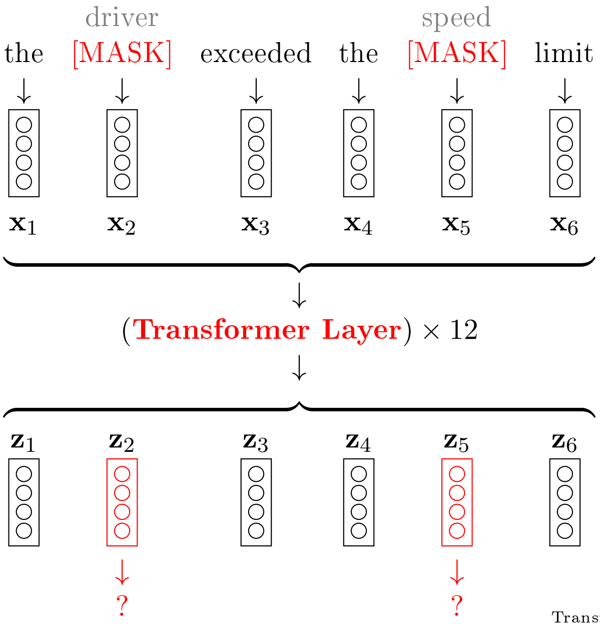
(figure generated by the following code
`https://github.com/jalammar/jalammar.github.io/blob/master/notebookes/transformer/transformer_positional_encoding_graph.ipynb`)

A Transformer layer

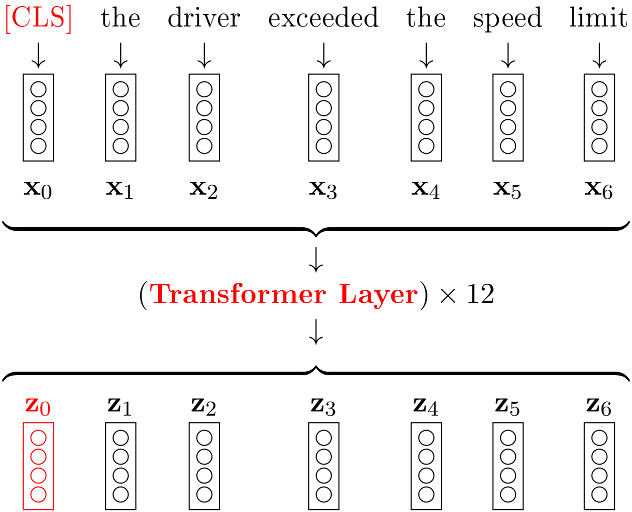


Transformer layers can be stacked !

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.