

Глубокое обучение и вообще

Соловей Влад и Шигапова Фирюза

22 декабря 2021 г.

Посиделка 12: Transformers United

Agenda

Transformers для

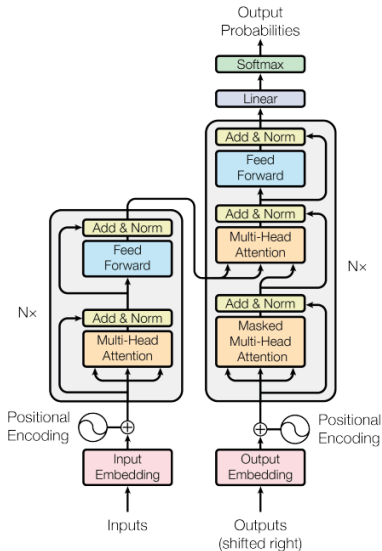
- Multivariate Time Series
- Tabular Data
- Images

Attention is all you need!

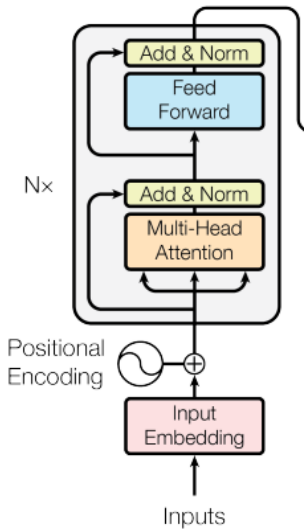
Attention is all you need

Развитие идеи внимания. Статья вышла в 2017 году и стала мамой всех текущих SOTA моделей.

Attention is all you need

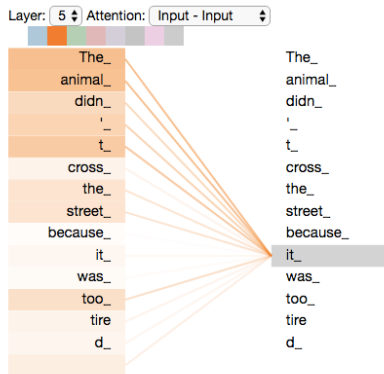


Encoder



Что мы хотим?

Есть предложение: "The animal didn't cross the street because it was too tired"



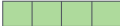
Абстракции!

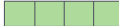
Input

Thinking


Machines


Embedding

x_1 

x_2 

Queries

q_1 


q_2 



W^Q

Keys


k_1 


k_2 



W^K

Values

v_1 

v_2 



W^V

А теперь тоже самое, но словами:

1. Query, key - ищем связи между словами. Ходим по всем со всеми смотрим насколько они связаны. Query - мое текущее слово, key - мое слово с которым я сравниваю себя.
2. Value - то, что мы знаем об этом слове

Mar 2

Input

Embedding

Queries

Keys

Values

Score

Thinking

x_1



q_1



k_1



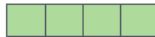
v_1



$$q_1 \cdot k_1 = 112$$

Machines

x_2



q_2



k_2



v_2



$$q_1 \cdot k_2 = 96$$

Mar 3

Input

Embedding

Queries

Keys

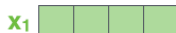
Values

Score

Divide by 8 ($\sqrt{d_k}$)

Softmax

Thinking

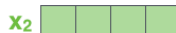


$$q_1 \cdot k_1 = 112$$

14

0.88

Machines

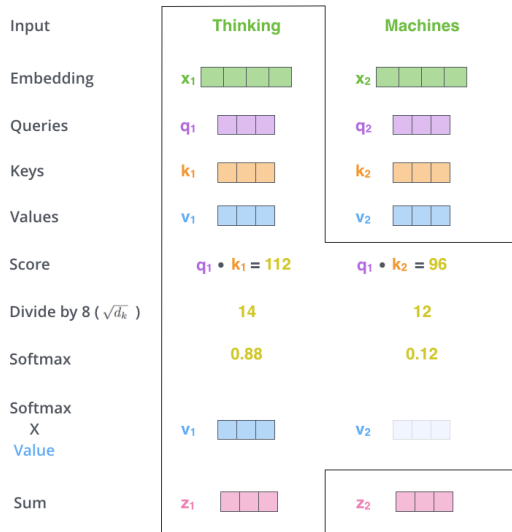


$$q_2 \cdot k_2 = 96$$

12

0.12

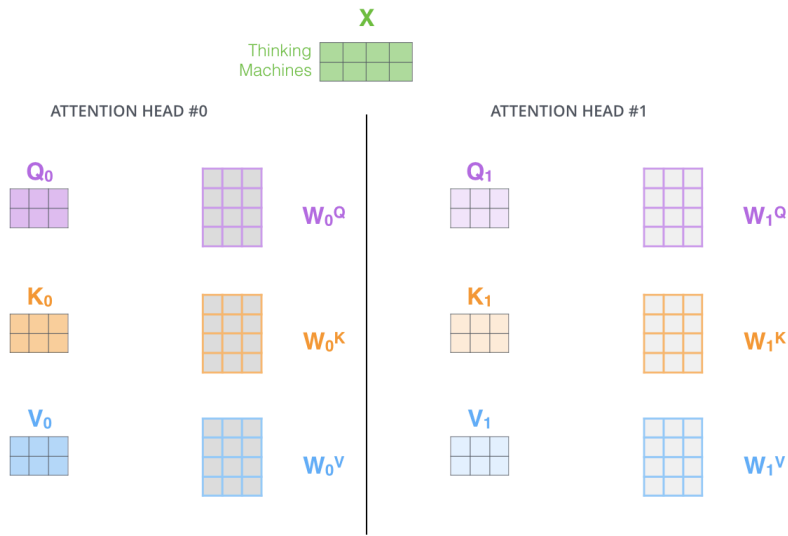
Mar 4



Шар 5



multi head attention



Соединяем!

1) Concatenate all the attention heads

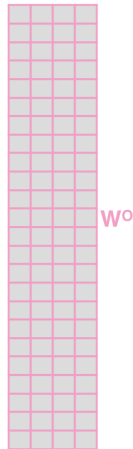


3) The result would be the Z matrix that captures information from all the attention heads. We can send this forward to the FFNN



2) Multiply with a weight matrix W^O that was trained jointly with the model

\times



Итого

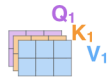
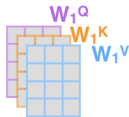
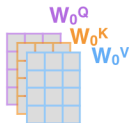
1) This is our input sentence*

Thinking
Machines

2) We embed each word*



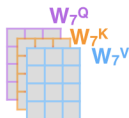
3) Split into 8 heads.
We multiply X or R with weight matrices



...

...

...



4) Calculate attention using the resulting $Q/K/V$ matrices

5) Concatenate the resulting Z matrices, then multiply with weight matrix W^O to produce the output of the layer

W^O



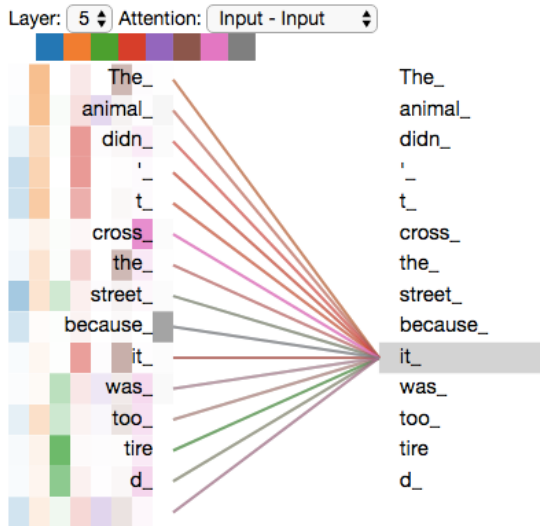
Z



* In all encoders other than #0, we don't need embedding. We start directly with the output of the encoder right below this one

R





Итого

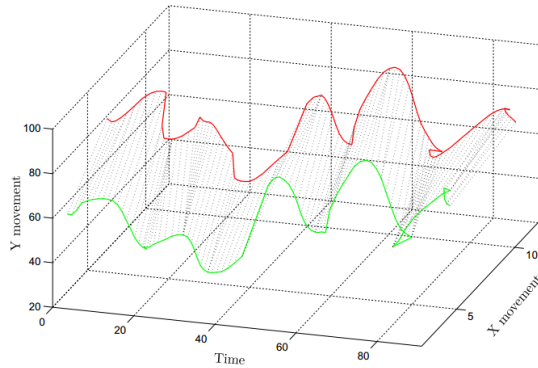
Выходом всего этого дела будут вектора key и value, которые позволят декодеру смотреть на нужные нам кусочки. И бежим смотреть гифки декодера!

Объяснение взято отсюда **английский оригинал** и отсюда **лекции мфти**

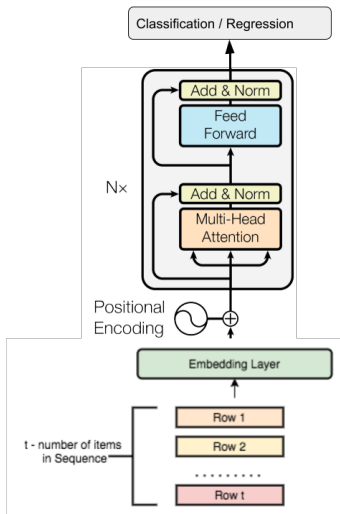
Итого

1. У нас нет никаких слоев, кроме dense
2. Учится очень классно, находит множество взаимосвязей
3. Positional Encoding позволяет учитывать позицию в тексте

Multivariate Time Series

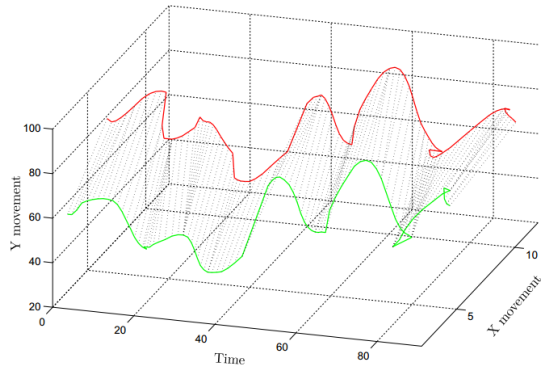


Multivariate Time Series

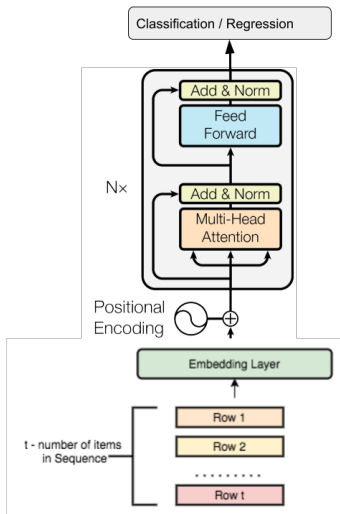


<https://arxiv.org/pdf/2010.02803.pdf>

Multivariate Time Series



Multivariate Time Series



<https://arxiv.org/pdf/2010.02803.pdf>

Multivariate Time Series - Positional Encoding

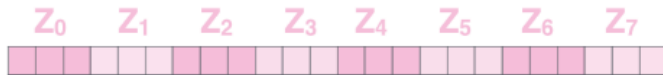
Positional Encoding позволяет учитывать позицию элемента в последовательности

1. Посчитать \cos - \sin и включить в качестве еще одной характеристики в вектор характеристик: $\text{concat}([u, \cos, \sin])$
2. Создать обучаемый вектор, равный размеру входного вектора характеристик, и сложить его с вектором: $u + W_{pos}$
3. Создать обучаемый скаляр и включить в качестве еще одной характеристики в вектор характеристик: $\text{concat}([u, \text{scalar}_{pos}])$

Multivariate Time Series - что можно после Encoder

MLP

Вытягиваем последовательность в один длинный вектор



Multivariate Time Series - что можно после Encoder RNN

Отдаем на вход RNN сети

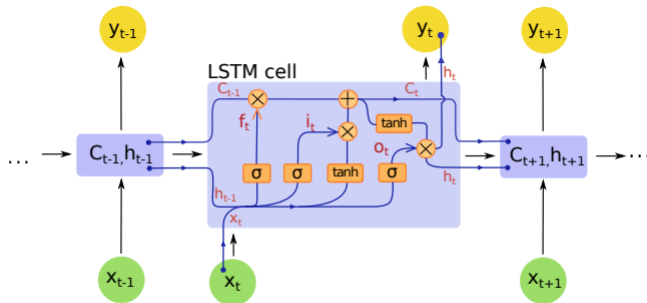
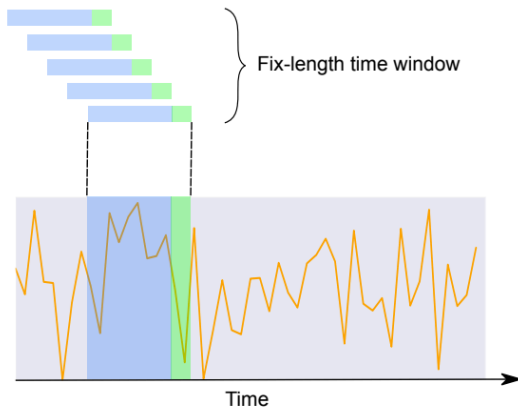


Figure 3. Long Short-Term Memory network and LSTM unit.

Multivariate Time Series - что можно после Encoder CNN

Отдаем на вход CNN сети



Tabular Data

	A	B	C	D	E	F	G	H	I	J	K	L
1	User	Card	Year	Month	Day	Time	Amount	Use Chip	Merchant Name	Merchant City	MCC	Is Fraud?
2	0	0	2019	4	21	9:47	\$6.81	Chip	Chevron	Brandon	5541	No
3	0	0	2019	4	21	10:38	\$10.07	Chip	Anwar Grocery	Brandon	5411	No
4	0	0	2019	4	22	3:53	\$42.61	Chip	Kelly Auto Repair	Brandon	7538	No
5	0	0	2019	4	22	7:28	\$47.66	Chip	Barnes & Noble	Brandon	5942	No
6	0	0	2019	4	22	10:30	\$9.73	Chip	Applebees	Brandon	5812	No
7	0	0	2019	4	23	15:02	\$121.47	Chip	Green Wholesale	Brandon	5300	No
8	0	0	2019	4	23	23:20	\$71.66	Online	Frontier Communications	ONLINE	4814	No
9	0	0	2019	4	24	10:13	\$11.05	Chip	Applebees	Brandon	5812	No
10	0	0	2019	4	24	10:17	\$11.05	Chip	Applebees	Brandon	5812	No

Tabular Data

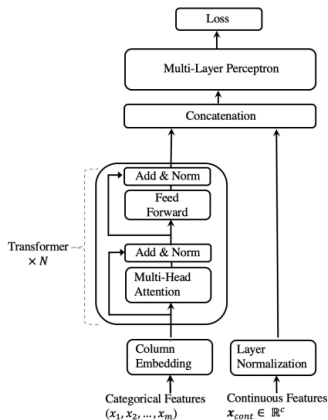


Figure 1: The architecture of TabTransformer.

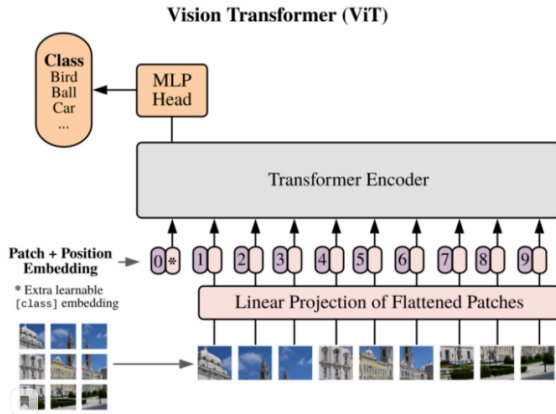
<https://arxiv.org/pdf/2012.06678.pdf>

Tabular Data

- Для каждой категориальной переменной ставим в соответствие Embedding
- Все категориальные признаки представляем в виде последовательности
- Созданная последовательность идет на вход Transformer

$$E(\mathbf{x}_{cat}) = e_1(x_1), \dots, e_m(x_m)$$

Vision Transformer



Picture by paper authors (Alexey Dosovitskiy et al.)

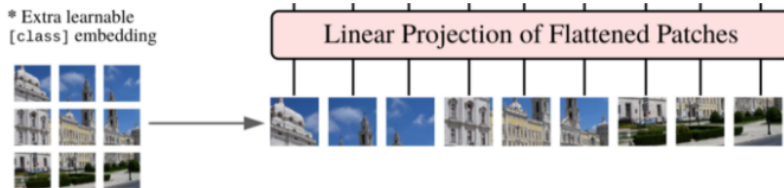
<https://arxiv.org/pdf/2010.11929.pdf>

Vision Transformer - Image to Patches



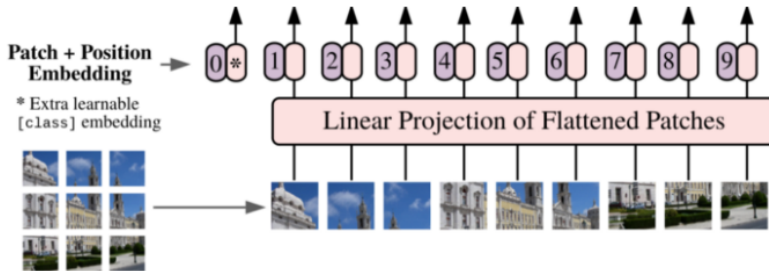
Picture by paper authors (Alexey Dosovitskiy et al.)

Vision Transformer - Image to Patches



Picture by paper authors (Alexey Dosovitskiy et al.)

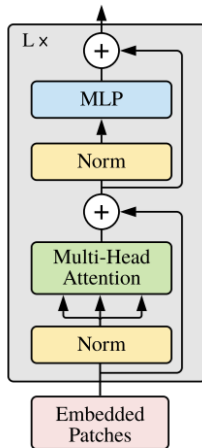
Vision Transformer - Position Embedding



Picture by paper authors (Alexey Dosovitskiy et al.)

Vision Transformer - Encoder

Transformer Encoder



Picture by paper authors (Alexey Dosovitskiy et al.)

Гибридные архитектуры

Вначале можно применить свертки (CNN) и только потом делать Patches.

Vision Transformer

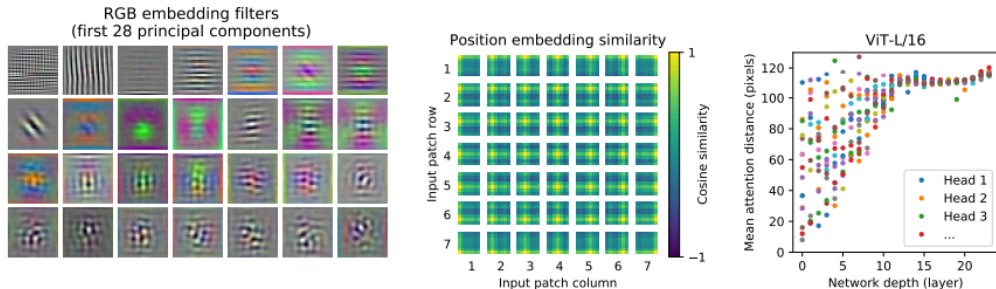


Figure 7: **Left:** Filters of the initial linear embedding of RGB values of ViT-L/32. **Center:** Similarity of position embeddings of ViT-L/32. Tiles show the cosine similarity between the position embedding of the patch with the indicated row and column and the position embeddings of all other patches. **Right:** Size of attended area by head and network depth. Each dot shows the mean attention distance across images for one of 16 heads at one layer. See Appendix D.7 for details.