I Want to Predict with Transformer

Reviewing ‘Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting’

Sungguk Cha

MINDs Lab

20th Algorithmic Trading Research Club
2021.06.26.

Contents

- Introduce *transformer*
- Introduce limitations of prior approaches
- Review ‘*Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting* (AAAI21)’
- Discussion

MINDs Lab

Transformer?

Transformer is

- proposed for *seq2seq* task.
- based on *attention* algorithm.

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} \times \begin{bmatrix} \text{Waa} & \text{Wba} & \text{Wca} \\ \text{Wab} & \text{Wbb} & \text{Wcb} \\ \text{Wac} & \text{Wbc} & \text{Wcc} \end{bmatrix} = \begin{bmatrix} a' \\ b' \\ c' \end{bmatrix}$$

Figure. Attention algorithm

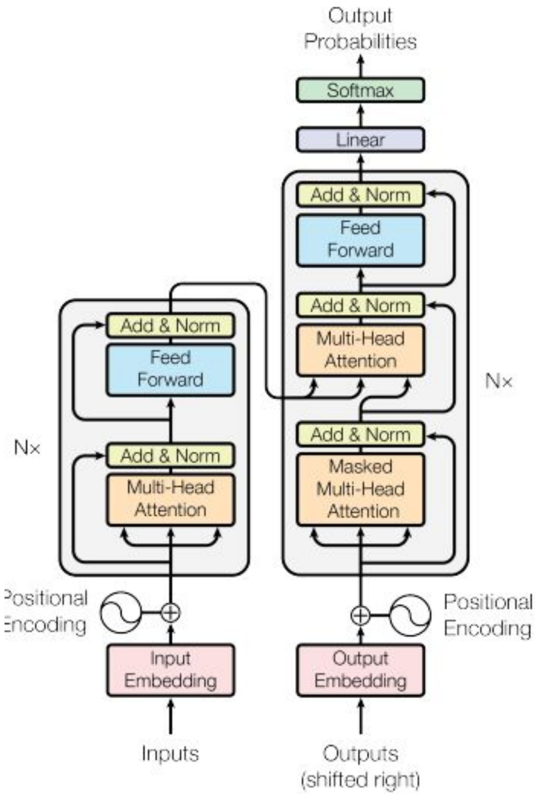


Figure 1: The Transformer - model architecture.

MINDs Lab

Figure 1 source: Vaswani et al., “Attention Is All You Need”, 2017

Transformer Is Overwhelming

Transformer has shown **overwhelming performances** over NLP, CV and multi-modal learning.

NLP: GPT-3 (text generation), BERT (language representation pretraining)

CV: ViT (image recognition)

Vision-NLP: CLIP (zero-shot classification), DALL E (text-to-image synthesis)

Vision-Audio: Lip reading, audio supervised object detection

MINDs Lab

Transformer Is Overwhelming

Transformer has shown **overwhelming performances** over NLP, CV and multi-modal learning.

NLP: GPT-3 (text generation), BERT (language representation pretraining)

CV: ViT (image recognition)

Vision-NLP: CLIP (zero-shot classification), DALL E (text-to-image synthesis)

Vision-Audio: Lip reading, audio supervised object detection

In a sense that transformer shows irreplaceable performance on seq2seq,
I believe it will be the same over price prediction task.

MINDs Lab

Limitations of Transformers

- The quadratic computation of self-attention: *O(L^2)*
- The memory bottleneck in stacking layers of *J* long inputs: *O(J*L^2)*
- The speed plunge in predicting long outputs.

MINDs Lab

Limitations of Transformers

- The quadratic computation of self-attention: *O(L^2)*
- The memory bottleneck in stacking layers of *J* long inputs: *O(J*L^2)*
- The speed plunge in predicting long outputs.

I will give you an example of BERT-base model w.r.t. the **space complexity**,

while the computation complexity is problematic as well.

MINDs Lab

Limitations of Transformers (cont.)

Example with a sequence with 512 tokens.

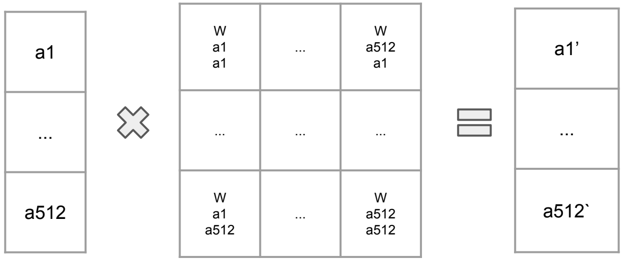


Figure. Attention algorithm

MINDs Lab

Limitations of Transformers (cont.)

Example with a sequence with 512 tokens.

A token is expressed with 4B * 768 = 3,072B (3KB).

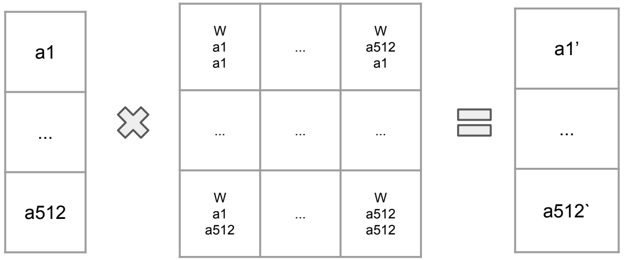


Figure. Attention algorithm

Limitations of Transformers (cont.)

Example with a sequence with 512 tokens.

A token is expressed with 4B * 768 = 3,072B (3KB).

A sequence is 3,072B * 512 = 1,572,864B(1.5MB).

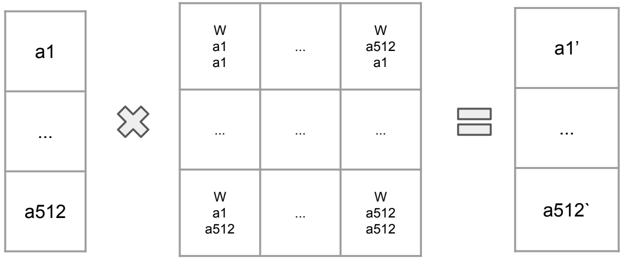


Figure. Attention algorithm

Limitations of Transformers (cont.)

Example with a sequence with 512 tokens.

A token is expressed with 4B * 768 = 3,072B (3KB).

A sequence is 3,072B * 512 = 1,572,864B(1.5MB).

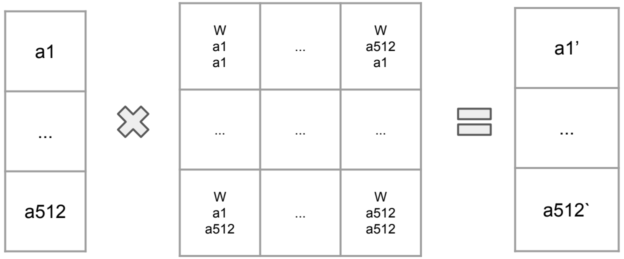


Figure. Attention algorithm

Attention matrix takes 512 * 512 * 4B * 768 = 805,306,368B (800MB)

Limitations of Transformers (cont.)

Example with a sequence with 512 tokens.

A token is expressed with 4B * 768 = 3,072B (3KB).

A sequence is 3,072B * 512 = 1,572,864B(1.5MB).

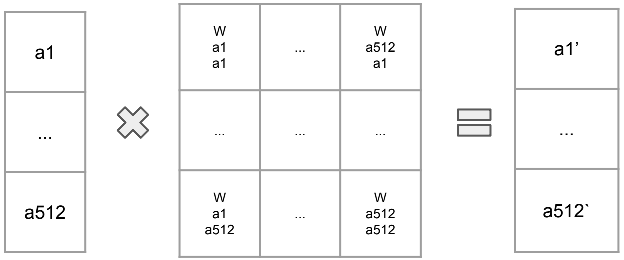


Figure. Attention algorithm

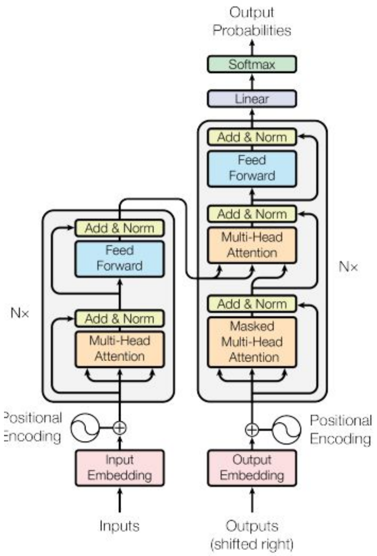


Figure 1: The Transformer - model architecture.

Transformer has 12 layers.
12 * 800 MB = 9.6GB.

1 batch requires at least about 10GB.

Training is conducted with thousands of batches.

I.e., **terabytes GPU memory** is needed.

Attention matrix takes 512 * 512 * 4B * 768 = 805,306,368B (800MB)

Figure 1 source: Vaswani et al., “Attention Is All You Need”, 2017

Informer!

1. The quadratic computation of self-attention: $O(L^2)$
2. The memory bottleneck in stacking layers of J long inputs: $O(J*L^2)$
3. The speed plunge in predicting long outputs.

MINDs Lab

Informer!

1. The quadratic computation of self-attention: *O(L^2)*

Heuristic approaches to obtain *O(L log L)* but the efficiency gain is limited.

Sparse Transformer (Child et al. 2019), LogSparse Transformer (Li et al. 2019), Longformer (Beltagy et al. 2020)

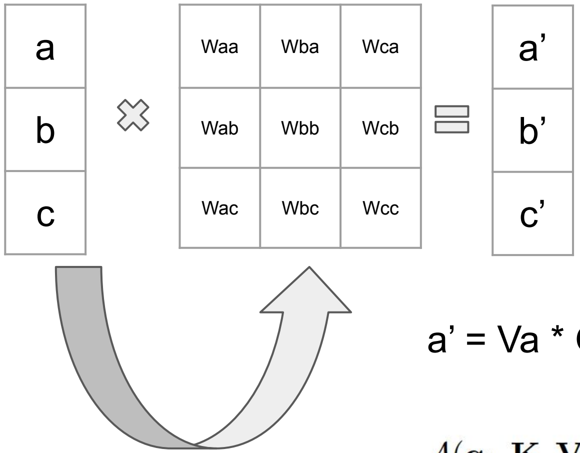
Reformer (Kitaev et al. 2019) achieves *O(L log L)* on extremely long sequences.

Linformer (Wang et al. 2020) conditionally obtains *O(L)* with risks of degradation to *O(L^2)*.

MINDs Lab

Efficient Self-Attention

Canonical Self-Attention Takes O(Lq Lk)



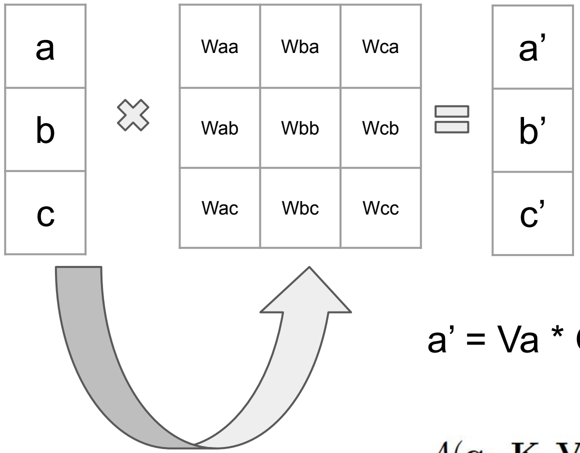
a’ = Va * QaKa + Vb * QaKb + Vc * QaKc

$$\mathcal{A}(\mathbf{q}_i, \mathbf{K}, \mathbf{V}) = \sum_j \frac{k(\mathbf{q}_i, \mathbf{k}_j)}{\sum_l k(\mathbf{q}_i, \mathbf{k}_l)} \mathbf{v}_j = \mathbb{E}_{p(\mathbf{k}_j|\mathbf{q}_i)}[\mathbf{v}_j]$$

MINDs Lab

Efficient Self-Attention

Proposed *ProbSparse* Self-attemtion takes O(Lk log Lq)



Constrain queries by ignoring sparse correlations.
| Q | -> *Lq*
| Q | -> *c log Lq*

a’ = Va * QaKa + Vb * QaKb + Vc * QaKc

$$\mathcal{A}(\mathbf{q}_i,\mathbf{K},\mathbf{V}) = \sum_j \frac{k(\mathbf{q}_i,\mathbf{k}_j)}{\sum_l k(\mathbf{q}_i,\mathbf{k}_l)}\mathbf{v}_j = \mathbb{E}_{p(\mathbf{k}_j|\mathbf{q}_i)}[\mathbf{v}_j]$$

MINDs Lab

Efficient Self-Attention

Measuring Sparsity

$p(\mathbf{k}_j|\mathbf{q}_i)$

If an attention value is closed to **1 / L**,
it means the attention is **as meaningless as uniform**.

MINDs Lab

Efficient Self-Attention

Measuring Sparsity

$p(\mathbf{k}_j|\mathbf{q}_i)$

If an attention value is closed to **1 / L**,
it means the attention is **as meaningless as uniform**.

$$M(\mathbf{q}_i, \mathbf{K}) = \ln \sum_{j=1}^{L_K} e^{\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}}$$

measuring KL divergence ….

MINDs Lab

Efficient Self-Attention

Measuring Sparsity

$p(\mathbf{k}_j|\mathbf{q}_i)$

If an attention value is closed to **1 / L**,
it means the attention is **as meaningless as uniform**.

$$M(\mathbf{q}_i, \mathbf{K}) = \ln \sum_{j=1}^{L_K} e^{\frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}}} - \frac{1}{L_K} \sum_{j=1}^{L_K} \frac{\mathbf{q}_i \mathbf{k}_j^\top}{\sqrt{d}}$$

measuring KL divergence ….

Waa	Wba	Wca
Wab	Wbb	Wcb
Wac	Wbc	Wcc

They are to use top- c log L correlations only according to their importance ranking (QiKj).

Computing relatively meaningful ones only.

| Q | -> *c log L*

MINDs Lab

Additional Engineering Contributions

- Reduce memory usage by self-attention distilling.
- Generates sequential output once
  - which can avoid error accumulation.

MINDs Lab

Experiment: Baselines

Methods	Training		Testing
	Time	Memory	Steps
Informer	$\mathcal{O}(L \log L)$	$\mathcal{O}(L \log L)$	1
Transformer	$\mathcal{O}(L^2)$	$\mathcal{O}(L^2)$	L
LogTrans	$\mathcal{O}(L \log L)$	$\mathcal{O}(L^2)$	$1^\star$
Reformer	$\mathcal{O}(L \log L)$	$\mathcal{O}(L \log L)$	L
LSTM	$\mathcal{O}(L)$	$\mathcal{O}(L)$	L

[1] The LSTnet is hard to present in a closed form.
[2] The $\star$ denotes applying our proposed decoder.

Table 4: L-related computation statics of each layer.

MINDs Lab

Experiment: Datasets

They experimented on their own dataset.

- Electricity Transformer Temperature (ETT)
- Electricity Consuming Load (ECL)
- Weather

MINDs Lab

Experimental Results

Methods		Informer		Informer†		LogTrans		Reformer		LSTMa		LSTnet	
Metric		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh$_1$	24	**0.577**	**0.549**	0.620	0.577	0.686	0.604	0.991	0.754	0.650	0.624	1.293	0.901
	48	**0.685**	**0.625**	0.692	0.671	0.766	0.757	1.313	0.906	0.702	0.675	1.456	0.960
	168	**0.931**	**0.752**	0.947	0.797	1.002	0.846	1.824	1.138	1.212	0.867	1.997	1.214
	336	1.128	0.873	**1.094**	**0.813**	1.362	0.952	2.117	1.280	1.424	0.994	2.655	1.369
	720	**1.215**	**0.896**	1.241	0.917	1.397	1.291	2.415	1.520	1.960	1.322	2.143	1.380
ETTh$_2$	24	**0.720**	**0.665**	0.753	0.727	0.828	0.750	1.531	1.613	1.143	0.813	2.742	1.457
	48	**1.457**	**1.001**	1.461	1.077	1.806	1.034	1.871	1.735	1.671	1.221	3.567	1.687
	168	3.489	**1.515**	3.485	1.612	4.070	1.681	4.660	1.846	4.117	1.674	**3.242**	2.513
	336	2.723	1.340	2.626	**1.285**	3.875	1.763	4.028	1.688	3.434	1.549	**2.544**	2.591
	720	**3.467**	**1.473**	3.548	1.495	3.913	1.552	5.381	2.015	3.963	1.788	4.625	3.709
ETTm$_1$	24	0.323	**0.369**	**0.306**	0.371	0.419	0.412	0.724	0.607	0.621	0.629	1.968	1.170
	48	0.494	0.503	**0.465**	**0.470**	0.507	0.583	1.098	0.777	1.392	0.939	1.999	1.215
	96	**0.678**	0.614	0.681	**0.612**	0.768	0.792	1.433	0.945	1.339	0.913	2.762	1.542
	288	**1.056**	**0.786**	1.162	0.879	1.462	1.320	1.820	1.094	1.740	1.124	1.257	2.076
	672	**1.192**	**0.926**	1.231	1.103	1.669	1.461	2.187	1.232	2.736	1.555	1.917	2.941
Weather	24	**0.335**	**0.381**	0.349	0.397	0.435	0.477	0.655	0.583	0.546	0.570	0.615	0.545
	48	0.395	0.459	**0.386**	**0.433**	0.426	0.495	0.729	0.666	0.829	0.677	0.660	0.589
	168	**0.608**	**0.567**	0.613	0.582	0.727	0.671	1.318	0.855	1.038	0.835	0.748	0.647
	336	**0.702**	**0.620**	0.707	0.634	0.754	0.670	1.930	1.167	1.657	1.059	0.782	0.683
	720	**0.831**	**0.731**	0.834	0.741	0.885	0.773	2.726	1.575	1.536	1.109	0.851	0.757
ECL	48	0.344	**0.393**	**0.334**	0.399	0.355	0.418	1.404	0.999	0.486	0.572	0.369	0.445
	168	0.368	0.424	**0.353**	**0.420**	0.368	0.432	1.515	1.069	0.574	0.602	0.394	0.476
	336	0.381	**0.431**	0.381	0.439	**0.373**	0.439	1.601	1.104	0.886	0.795	0.419	0.477
	720	0.406	0.443	**0.391**	**0.438**	0.409	0.454	2.009	1.170	1.676	1.095	0.556	0.565
	960	**0.460**	**0.548**	0.492	0.550	0.477	0.589	2.141	1.387	1.591	1.128	0.605	0.599
Count		33		14		1		0		0		2	

MINDs Lab

Conclusion

- We introduced Transformer and its limitations.

- We simply reviewed some approaches making transformer *lighter*.

MINDs Lab

Discussion

- Additionally, transformer is known to require ***extreme-scale*** data.
- Can we make awesome DNN for price prediction except temporarily dominating RL?

MINDs Lab