Yformer: U-Net Inspired Transformer Architecture for Far Horizon Time Series Forecasting

Yformer

Paper by Kiran Madhusudhanan Johannes Burchert Lars Schmidt-Thieme
Presentation by Tobramycin

NCUT

2022.6

Contents

Background

ProbSparse Self-Attention[1]

Implement of the *ProbSparse* self-attention

We have implemented the *ProbSparse* self-attention in Python 3.6 with Pytorch 1.0. The pseudo-code is given in Algo.(1). The source code is available at https://github.com/zhouhaoyi/Informer2020. All the procedure can be highly efficient vector operation and maintains logarithmic total memory usage. The masked version can be achieved by applying positional mask on step 6 and using cmusum$(\cdot)$ in mean$(\cdot)$ of step 7. In the practice, we can use sum$(\cdot)$ as the simpler implement of mean$(\cdot)$.

Algorithm 1 ProbSparse self-attention

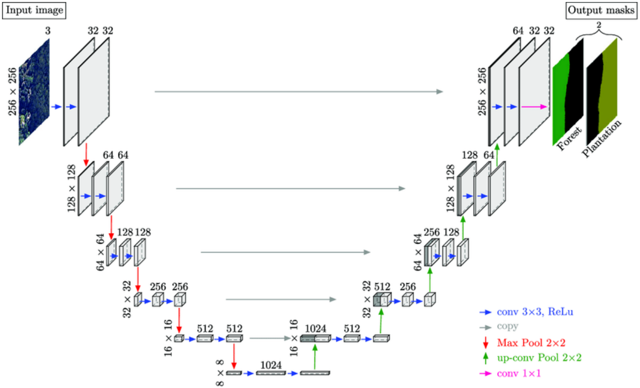
Require: Tensor $\mathbf{Q} \in \mathbb{R}^{m \times d}$, $\mathbf{K} \in \mathbb{R}^{n \times d}$, $\mathbf{V} \in \mathbb{R}^{n \times d}$

1: **print** set hyperparameter c, $u = c \ln m$ and $U = m \ln n$
2: randomly select U dot-product pairs from $\mathbf{K}$ as $\bar{\mathbf{K}}$
3: set the sample score $\bar{\mathbf{S}} = \mathbf{Q}\bar{\mathbf{K}}^\top$
4: compute the measurement $M = \max(\bar{\mathbf{S}}) - \text{mean}(\bar{\mathbf{S}})$ by row
5: set Top-u queries under M as $\bar{\mathbf{Q}}$
6: set $\mathbf{S}_1 = \text{softmax}(\bar{\mathbf{Q}}\mathbf{K}^\top/\sqrt{d}) \cdot \mathbf{V}$
7: set $\mathbf{S}_0 = \text{mean}(\mathbf{V})$
8: set $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_0\}$ by their original rows accordingly

Ensure: self-attention feature map $\mathbf{S}$.

[1] Xiaowu Zou et al. "Integration of residual network and convolutional neural network along with various activation functions and global pooling for time series classification". In: *Neurocomputing* 367 (2019), pp. 39–45.

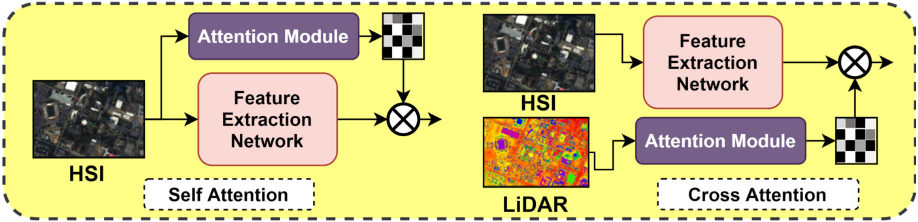
U-Net[2]



up-conv-1 up-conv-2

[2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

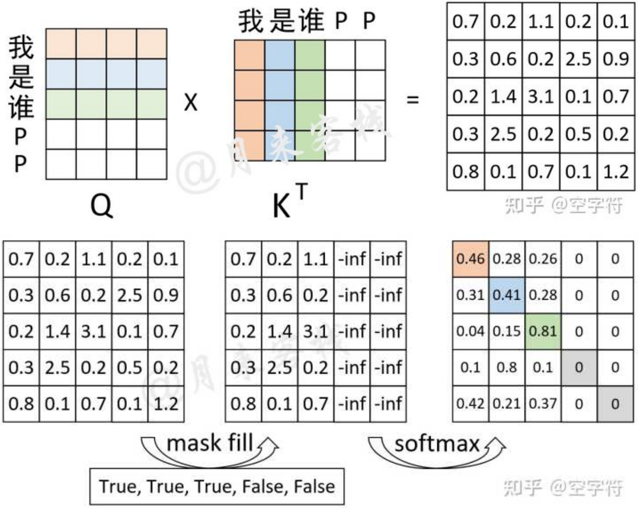
Cross-Attention



Self-Attention vs Cross-Attention[3]

[3] Satyam Mohla et al. "Fusatnet: Dual attention based spectrospatial multimodal fusion network for hyperspectral and lidar classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2020, pp. 92–93.

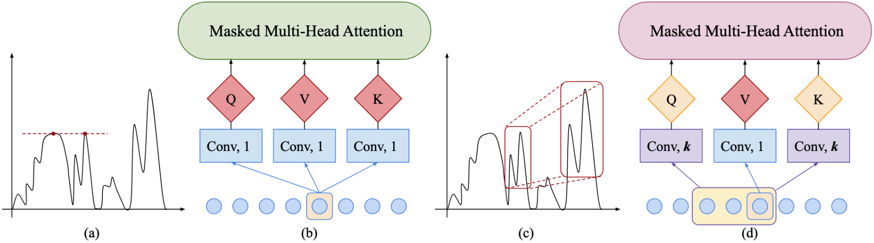
Masked Self-Attention[4]



[4] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

Canonical Self-Attention

Canonical Self-Attention[5] vs Convolutional Self-Attention[6]

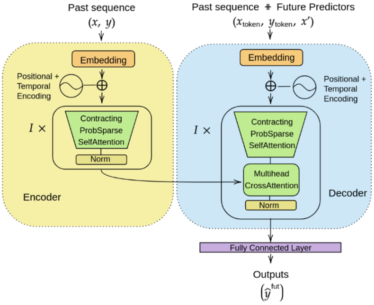


5 Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

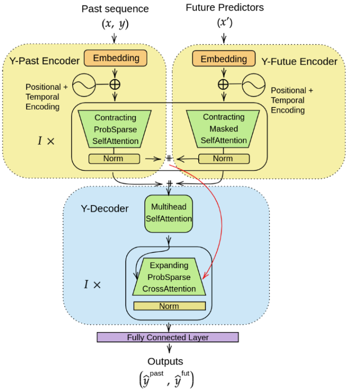
6 Shiyang Li et al. "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting". In: *Advances in neural information processing systems* 32 (2019).

Yformer

Framework

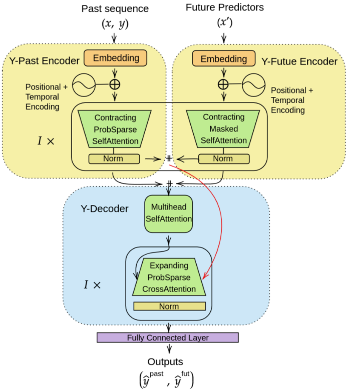


(a) Informer Architecture



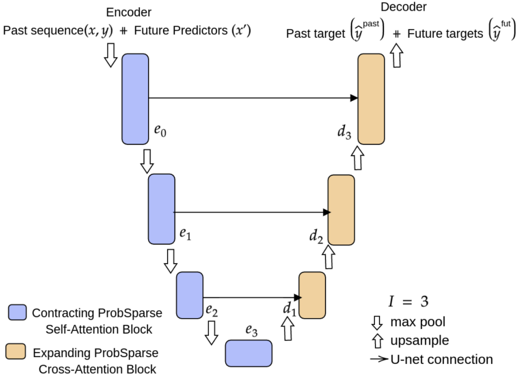
(b) Yformer Architecture

Y-Decoder



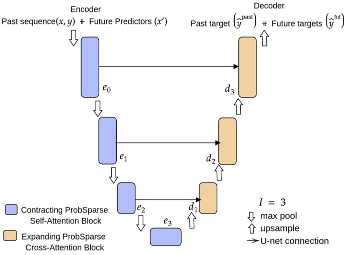
- ”... mitigates the issue of the redundant reprocessing of parts of the past sequence (x, y) used as tokens (xtoken, ytoken) in the Informer architecture.”
- ”... construct direct connections between the lower levels of the encoder and the corresponding symmetric higher levels of the decoder.”

U-net architecture



- "The U-net architecture is capable of compressing information by aggregating over the inputs and up-sampling embeddings to the same resolutions as that of the inputs from their compressed latent features."

Contracting ProbSparse Cross Attention



Algorithm 2 Contracting ProbSparse Self-Attention Block

Input : h_i

Output : h_{i+1}

$h_{i+1} \leftarrow \mathrm{ProbSparseAttn}(h_i, h_i)$

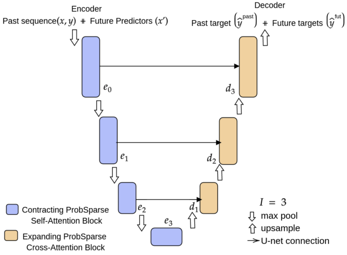
$h_{i+1} \leftarrow \mathrm{Conv1d}(h_{i+1})$

$h_{i+1} \leftarrow \mathrm{Conv1d}(h_{i+1})$

$h_{i+1} \leftarrow \mathrm{LayerNorm}(h_{i+1})$

$h_{i+1} \leftarrow \mathrm{MaxPool}(\mathrm{ELU}(\mathrm{Conv1d}(h_{i+1})))$

Expanding ProbSparse Cross Attention



Algorithm 1 Expanding ProbSparse Cross-Attention Block

Input : d_{I-i}, e_i
Output : d_{I-i+1}
$d_{I-i+1} \leftarrow \text{ProbSparseCrossAttn}(d_{I-i}, e_i)$
$d_{I-i+1} \leftarrow \text{Conv1d}(d_{I-i+1})$
$d_{I-i+1} \leftarrow \text{Conv1d}(d_{I-i+1})$
$d_{I-i+1} \leftarrow \text{LayerNorm}(d_{I-i+1})$
$d_{I-i+1} \leftarrow \text{ELU}(\text{ConvTranspose1d}(d_{I-i+1})))$

EXPERIMENTS

Datasets

Datasets	Datasets	# of sensors	Time Range
ETT	Electricity Transformer Temperature	6 features	70,080 data point
ECL	Electricity Consuming Load	370 clients	from 2011 to 201

Result

Dataset	Horizon (τ)	Methods					Improvement %
		LogTrans	LSTnet	Informer*	Informer	Yformer	
ETTh1	24	0.259	0.280	*0.246*	0.247	**0.230**	6.50
	48	0.328	0.327	0.322	*0.319*	**0.308**	3.45
	168	0.375	0.422	0.355	*0.346*	**0.268**	22.54
	336	0.398	0.552	*0.369*	0.387	**0.365**	1.08
	720	0.463	0.707	*0.421*	0.435	**0.394**	6.41
ETTh2	24	0.255	0.263	0.241	*0.240*	**0.221**	7.92
	48	0.348	0.341	*0.317*	**0.314**	0.334	−6.37
	168	0.422	0.414	0.390	*0.389*	**0.337**	13.37
	336	0.437	0.607	0.423	*0.417*	**0.391**	6.24
	720	0.493	0.58	0.442	*0.431*	**0.382**	11.37
ETTm1	24	0.202	0.243	0.160	*0.137*	**0.118**	13.87
	48	0.220	0.362	*0.194*	0.203	**0.173**	10.82
	96	0.386	0.496	0.384	*0.372*	**0.311**	16.40
	288	0.572	0.795	*0.548*	0.554	**0.316**	42.34
	672	0.702	1.352	0.664	*0.644*	**0.476**	26.09
ECL	48	0.429	*0.357*	0.368	0.359	**0.322**	9.80
	168	0.529	*0.436*	0.514	0.503	**0.361**	17.20
	336	0.563	*0.519*	0.552	0.528	**0.375**	27.75
	720	0.609	0.595	0.578	*0.571*	**0.479**	19.50
	960	0.645	0.683	0.638	*0.608*	**0.573**	16.11
# wins per method		0	0	0	1	19	
						Avg	13.62

Thank you!