Lightweight Self-Attentive Sequential Recommendation

Yang Li
The University of Queensland
yang.li@uq.edu.au

Tong Chen
The University of Queensland
tong.chen@uq.edu.au

Peng-Fei Zhang
The University of Queensland
mima.zpf@gmail.com

Hongzhi Yin
The University of Queensland
h.yin1@uq.edu.au

Introduction

- How to move data analytics from cloud servers to edge devices to ensure timeliness and privacy for recommendation system?

- How to effectively learns local and global user preference signals for accurate sequential recommendation?

Introduction

- A dynamic context-aware compositional embedding scheme was devised for where the item embedding was generated by the combination of base embeddings.

- A novel twin-attention sequential framework was proposed, which specializes the learning of long- and short-term user preference signals via a dedicated self-attention and convolution operation, respectively.

An Overview of LSAN (Lightweight Self-attentive network)

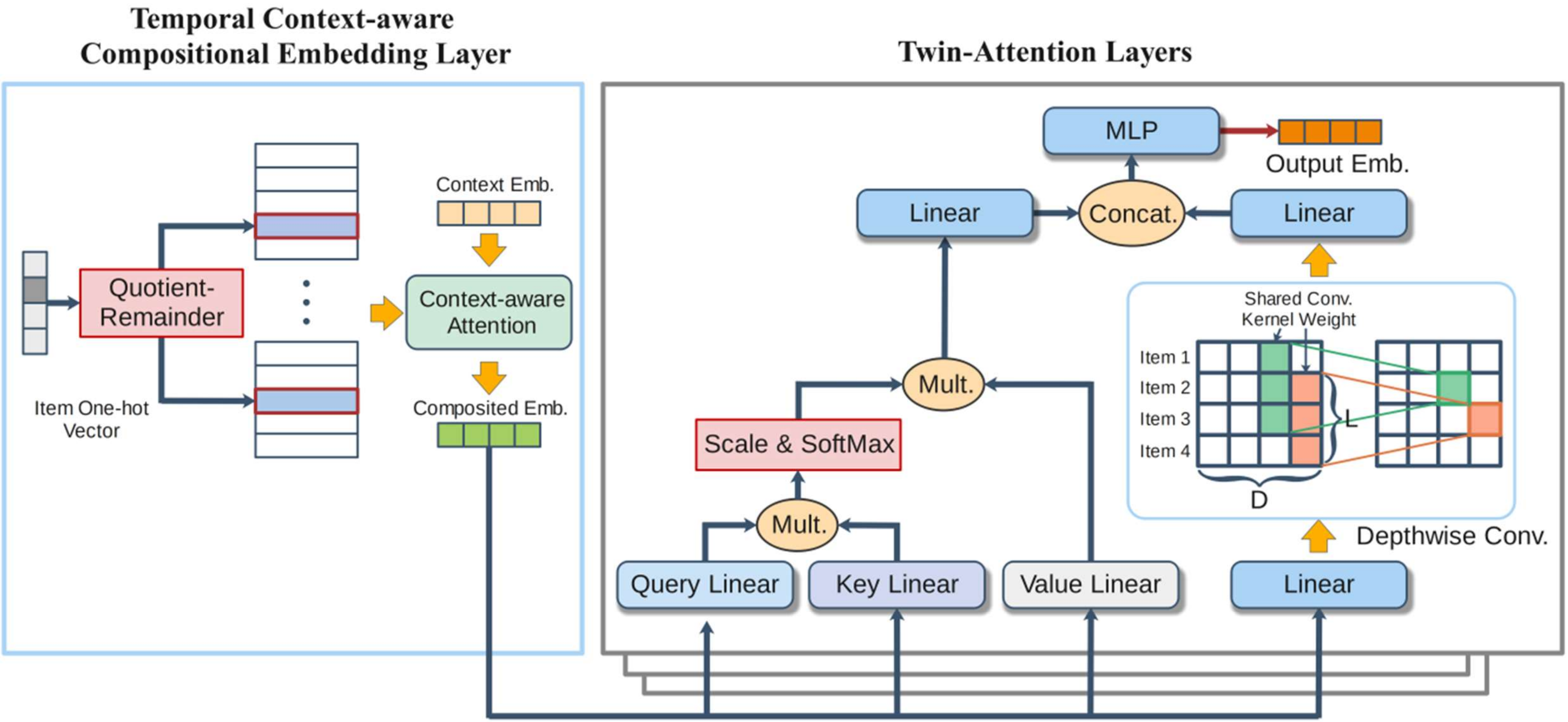

Figure 1: An overview of the proposed LSAN model.

Dynamic Context-aware Compositional Embedding

- Let N as the base embedding matrices, $N = \{\widetilde{E_1}, \dots, \widetilde{E_n}\}$. Denote m_n as the number of base embeddings in the n-th base embedding table $\widetilde{E_n}$. Let V as the item set. If $N \ll |V|, m_n \ll |V|\ and\ \prod_{n=1}^{N} m_n \gg |V|$, we could generate the item embedding by base embedding matrices, and ensure the uniqueness for each item.

Dynamic Context-aware Compositional Embedding

- *Quotient-remainder trick*

$$\widetilde{\boldsymbol{e}_i}^{\boldsymbol{1}} = \widetilde{\boldsymbol{E_1}}^{\boldsymbol{T}} \boldsymbol{R^1} \boldsymbol{f_i}$$

$$\boldsymbol{R^1_{i,j}} = \begin{cases} 1 \; if \; j \; mod \; m_1 = index(v_i) \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

Where $\boldsymbol{f_i} \in \mathbb{R}^{|\boldsymbol{V}|}$ is the one-hot encoding of v_i, $\boldsymbol{R^1} \in \mathbb{R}^{\boldsymbol{m_1} \times |\boldsymbol{V}|}$ is the hash table of embedding table $\widetilde{\boldsymbol{E_1}}$, $\widetilde{\boldsymbol{e}_i}^{\boldsymbol{1}}$ is the first base embedding of v_i.

$$\widetilde{\boldsymbol{e}_i}^{\boldsymbol{n}} = \widetilde{\boldsymbol{E_n}}^{\boldsymbol{T}} \boldsymbol{R^n} \boldsymbol{f_i}$$

$$\boldsymbol{R^n_{i,j}} = \begin{cases} 1 \; if \; j \; mod \; m_j = index(v_i) \backslash \prod_{n=1}^{n-1} m_n \\ 0 \qquad\qquad\qquad\qquad otherwise \end{cases}$$

$\widetilde{\boldsymbol{e}_i}^{\boldsymbol{n}}$ is the n-th base embedding of v_i.

Dynamic Context-aware Compositional Embedding

- *Base embeddings combination based on attention weight*

$$\alpha_n = \frac{\exp(\boldsymbol{r}_i^T SiLU(\boldsymbol{W}_a \widetilde{\boldsymbol{e}_i}^{\boldsymbol{n}}))}{\sum_{n=1}^{N} \exp(\boldsymbol{r}_i^T SiLU(\boldsymbol{W}_a \widetilde{\boldsymbol{e}_i}^{\boldsymbol{n}}))}$$

$$\boldsymbol{h_i} = \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \alpha_n \widetilde{\boldsymbol{e}_i}^{\boldsymbol{n}}$$

Where $r_i = (c_{i-1}, c_i, time(i))$ as a triplet of the categories of the previous and current items and the discrete time slot. A one-hot vector can be assigned for each r_i, then mapping the one-hot vector into a dense context embedding $\boldsymbol{r_i} \in \mathbb{R}^D$. $SiLU(x) = x \cdot sigmoid(x)$ is an activation function.

$$\widetilde{\boldsymbol{h_i}} = \mathrm{MLP}([\boldsymbol{h}_i; \boldsymbol{r}_i])$$

where [;] is the concatenation operation and $\mathrm{MLP}(\cdot): 2D \rightarrow D$ denotes a multi-layer perceptron.

Twin-Attention for User's Long- and Short-term Preferences

- *Convolution Branch for Local Patterns*

  Performing 1D convolution over the embedding matrix

$$\boldsymbol{H}_{i,d}^{conv} = \sum_{\boldsymbol{j}=\mathbf{1}}^{\boldsymbol{L}} \boldsymbol{W}_j^{conv} \boldsymbol{H}_{\left(i+j+\left\lceil \frac{L+1}{2} \right\rceil\right),d} \quad d = 1, \dots, D$$

Where $\boldsymbol{W}_j^{conv} \in \mathbb{R}^L$ is the kernel, $\boldsymbol{H}^{conv} \in \mathbb{R}^{T \times D}$ is the output matrix.

Twin-Attention for User's Long- and Short-term Preferences

- *Self-attention Branch for Global Patterns*

$$\widetilde{\boldsymbol{H}} = \boldsymbol{H} + \boldsymbol{P}$$

$$\widehat{\boldsymbol{H}} = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^T}{\sqrt{D/_H}}\right)\boldsymbol{V}$$

Where $\boldsymbol{P} \in \mathbb{R}^{T \times D}$ is a learnable position embeddings, $\boldsymbol{Q} = \boldsymbol{W_q}\widetilde{\boldsymbol{H}}, \boldsymbol{K} = \boldsymbol{W_k}\widetilde{\boldsymbol{H}}$ *and* $\boldsymbol{Q} = \boldsymbol{W_q}\widetilde{\boldsymbol{H}}$ are transformed item representations that are projection into query, key and value spaces, respectively.

- *Enhancing Expressiveness with Parallelism*

$$\boldsymbol{H}^{twin} = [\boldsymbol{H}_1^{conv}; \ldots, \boldsymbol{H}_H^{conv}; \boldsymbol{H}_1^{attn}; \ldots; \boldsymbol{H}_H^{attn}]$$

Each convolution and self-attention modules have H heads in parallel. $\boldsymbol{H}^{twin} \in \mathbb{R}^{T \times 2HD}$ is the final output.

Prediction Layer & Learning Objective

- *Prediction Layer*

$$\widehat{\boldsymbol{H}}^{twin} = GeLU\left(\boldsymbol{H}^{twin}\boldsymbol{W}_p^{(1)} + \boldsymbol{b}_p^{(1)}\right)\boldsymbol{W}_p^{(2)} + \boldsymbol{b}_p^{(2)}$$

$$\widehat{\boldsymbol{y}} = \boldsymbol{softmax}(\boldsymbol{W_o}\widehat{\boldsymbol{H}}^{twin} + \boldsymbol{b_o})$$

$GeLU(\cdot)$ denotes the Gaussian error linear unit. $\widehat{\boldsymbol{y}}$ is the items probability score distribution.

- *Learning Objective*

$$\mathcal{L} = -\frac{1}{S}\sum_{s=1}^{S} \boldsymbol{y}_s^T \log(\widehat{\boldsymbol{y}}_s) + \lambda\|\Psi\|_2^2$$

Experiments

Table 2: Comparison on sequential recommendation accuracy and model sizes. In each row, the best and second best results are highlighted in boldface and underlined, respectively. The parameter size of each model is obtained when $D = 128$.

Datasets	Metrics	FPMC	GRU4Rec	Caser	SASRec	BERT4Rec	**LSAN**$_{full.emb}$	Improv.	**LSAN**	Improv.
	HR@5	0.0149	0.0164	0.0205	0.0419	0.0312	0.0432	3.10%	**0.0492**	17.42%
	HR@10	0.0273	0.0283	0.0347	0.0650	0.0468	0.067	3.08%	**0.0785**	20.77%
	HR@20	0.0438	0.0479	0.0556	0.0872	0.0737	0.0992	13.76%	**0.1201**	37.73%
Beauty	nDCG@5	0.0096	0.0099	0.0131	0.0263	0.0223	0.0276	4.94%	**0.0316**	20.15%
	nDCG@10	0.0133	0.0137	0.0176	0.0337	0.0272	0.0352	4.45%	**0.041**	21.66%
	nDCG@20	0.0173	0.0187	0.0229	0.0372	0.0340	0.0433	16.4%	**0.0515**	38.44%
	#Parameters	8.26M	4.06M	8.42M	1.75M	4.29M	1.71M	-	1.11M	-
	HR@5	0.0099	0.0097	0.0166	0.0450	0.0136	0.045	0.00%	0.0437	-2.89%
	HR@10	0.0175	0.0176	0.0270	0.0650	0.0195	0.0676	4.00%	**0.0711**	9.38%
	HR@20	0.0273	0.0301	0.0420	0.0925	0.0333	0.097	4.86%	**0.1181**	27.68%
Toys	nDCG@5	0.0064	0.0059	0.0107	0.0300	0.0077	**0.0305**	1.67%	0.0283	-5.67%
	nDCG@10	0.0088	0.0084	0.0141	0.0370	0.0096	**0.0378**	2.16%	0.037	0.00%
	nDCG@20	0.0112	0.0116	0.0179	0.0436	0.0130	0.0452	3.67%	**0.0488**	11.93%
	#Parameters	7.77M	4.01M	7.9M	1.73M	4.24M	1.68M	-	1.28M	-
	HR@5	0.0088	0.0129	0.0116	0.0201	0.0139	0.0229	13.93%	**0.0314**	56.22%
	HR@10	0.0160	0.0204	0.0194	0.0314	0.0207	0.0366	16.56%	**0.0481**	53.18%
	HR@20	0.0259	0.0333	0.0314	0.0486	0.0438	0.0578	18.93%	**0.0759**	56.17%
Sports	nDCG@5	0.0055	0.0086	0.0072	0.0129	0.0085	0.0146	13.18%	**0.0211**	63.57%
	nDCG@10	0.0077	0.0110	0.0097	0.0164	0.0106	0.0191	16.46%	**0.0264**	60.98%
	nDCG@20	0.0100	0.0142	0.0126	0.0208	0.0162	0.0244	17.31%	**0.0334**	60.58%
	#Parameters	12.76M	5.83M	12.93M	2.55M	6.05M	2.51M	-	1.73M	-
	HR@5	0.0116	0.0152	0.0151	0.0210	0.0184	0.0251	19.52%	**0.0385**	83.33%
	HR@10	0.0211	0.0263	0.0253	0.0356	0.0259	0.0451	26.69%	**0.0682**	91.57%
	HR@20	0.0352	0.0439	0.0422	0.0575	0.0430	0.0744	29.39%	**0.1148**	99.65%
Yelp	nDCG@5	0.0074	0.0099	0.0096	0.0126	0.0114	0.0157	24.6%	**0.0205**	62.7%
	nDCG@10	0.0103	0.0134	0.0129	0.0176	0.0138	0.0221	25.57%	**0.0301**	71.02%
	nDCG@20	0.0137	0.0178	0.0171	0.0230	0.0181	0.0294	27.83%	**0.0417**	81.3%
	#Parameters	10.20M	5.32M	10.37M	2.32M	5.61M	2.27M	-	1.53M	-

Experiments

Table 3: A comparison of performance results and number of model parameters using different embedding compression rate m_1 on four datasets.

Datasets	Metrics	SASRec	LSAN(2x)	LSAN(3x)	LSAN(4x)	LSAN(5x)
Beauty	HR@20	0.0872	0.1201	0.0981	0.043	0.0456
	nDCG@20	0.0372	0.0515	0.0385	0.0158	0.0178
	#Parameters	1.75M	1.11M	0.71M	0.58M	0.5M
	Relative Size	100.00%	63.43%	40.57%	33.14%	28.57%
Toys	HR@20	0.0925	0.1181	0.0887	0.0618	0.0539
	nDCG@20	0.0436	0.0488	0.0341	0.0232	0.0211
	#Parameters	1.73M	1.28M	0.88M	0.75M	0.68M
	Relative Size	100.00%	73.99%	50.87%	43.35%	39.31%
Sports	HR@20	0.0486	0.0759	0.0551	0.0370	0.0311
	nDCG@20	0.0208	0.0334	0.0249	0.0159	0.0125
	#Parameters	2.55M	1.73M	1.34M	1.14M	1.03M
	Relative Size	100.00%	67.84%	52.55%	44.71%	40.39%
Yelp	HR@20	0.0575	0.1148	0.1087	0.0472	0.0436
	nDCG@20	0.023	0.0417	0.0434	0.0187	0.0161
	#Parameters	2.32M	1.53M	1.03M	0.85M	0.74M
	Relative Size	100.00%	65.95%	44.40%	36.64%	31.90%

Experiments

Table 4: Ablation study of different variants on four datasets.

Datasets	Variants			
	Metrics	$\text{LSAN}_{w/o.dynamic}$	$\text{LSAN}_{plain.attn}$	LSAN
Beauty	HR@20	0.977	0.108	0.120
	nDCG@20	0.038	0.045	0.051
Toys	HR@20	0.094	0.063	0.118
	nDCG@20	0.033	0.021	0.049
Sports	HR@20	0.076	0.066	0.076
	nDCG@20	0.033	0.030	0.033
Yelp	HR@20	0.104	0.011	0.115
	nDCG@20	0.038	0.046	0.042