Filter-enhanced MLP is All You Need for Sequential Recommendation

Kun Zhou[1,4†], Hui Yu[2,5†], Wayne Xin Zhao[3,4*] and Ji-Rong Wen[3,4]
[1]School of Information, Renmin University of China, China
[2]University of Chinese Academy of Sciences, China
[3]Gaoling School of Artificial Intelligence, Renmin University of China, China
[4]Beijing Key Laboratory of Big Data Management and Analysis Methods
[5]Key Laboratory of Petroleum Resources Research, Institute of Geology and Geophysics, Chinese Academy of Sciences
francis_kun_zhou@163.com, ishyu@outlook.com, batmanfly@gmail.com, jrwen@ruc.edu.cn

WWW 2022
Code: https://github.com/RUCAIBox/FMLP-Rec.
Dataset: Beauty Sports Toys Yelp Nowplaying Retailrocket Tmall Yoochoose

gesis Leibniz-Institut für Sozialwissenschaften

Forschungszentrum L3S Research Center

Reported by liang li

Motivation&Solution

Motivation(Take SASRec as an example):

- logged user behavior data is inevitable to contain noise
- and deep recommendation models are easy to overfit on these logged data

Solution (Take SASRec as an example)：

To tackle this problem, we borrow the idea of filtering algorithms from signal processing that attenuates the noise in the frequency domain

By replacing the self-attention components from Transformers to a novel filter layer

In a filter we perform Fast Fourier Transform (FFT) to convert the input representations into the frequency domain and an inverse FFT procedure recovers the denoised representations.

Problem Statement

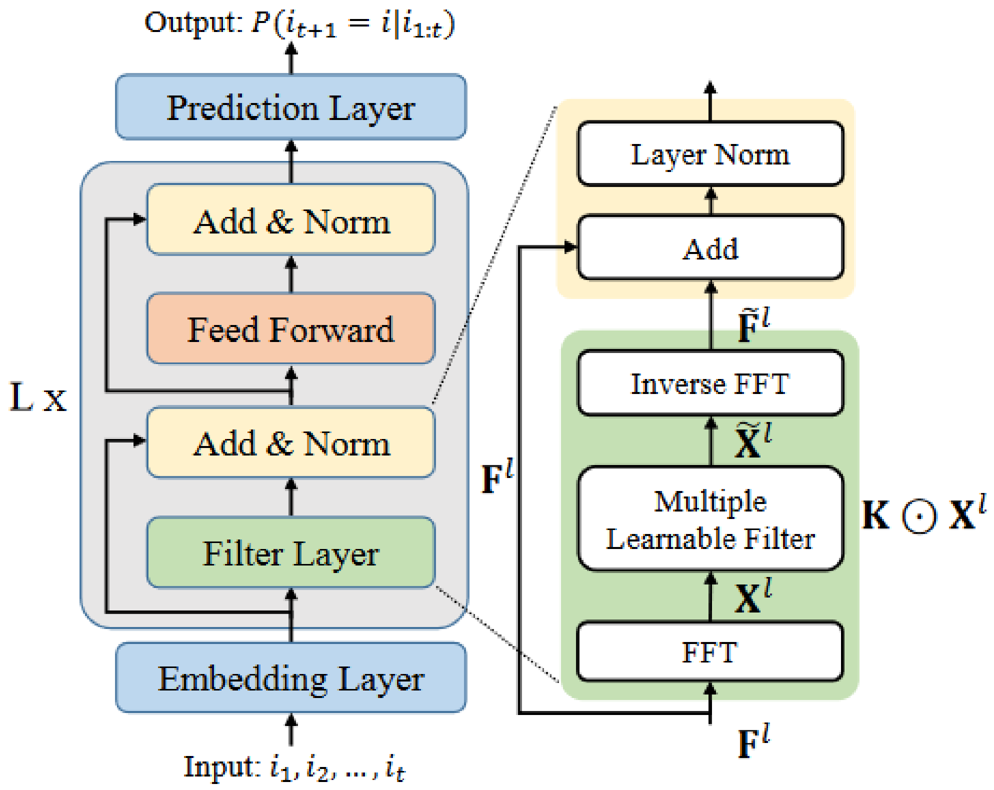

Figure 2: The overview of our FMLP-Rec, an all-MLP model that stacks multiple learnable filter-enhanced blocks.

Assume that we have a set of users and items, denoted by $\mathcal{U}$ and $\mathcal{I}$, respectively, where $u \in \mathcal{U}$ denotes a user and $i \in \mathcal{I}$ denotes an item. The numbers of users and items are denoted as $|\mathcal{U}|$ and $|\mathcal{I}|$, respectively. For sequential recommendation with implicit feedback, a user u has a context c, a chronologically-ordered interaction sequence with items: $c = \{i_1, \cdots, i_n\}$, where n is the number of interactions and i_t is the t-th item that the user u has interacted with. For convenience, we use $i_{j:k}$ to denote the subsequence, *i.e.*, $i_{j:k} = \{i_j, \cdots, i_k\}$ where $1 \leq j < k \leq n$.

Based on the above notations, we now define the task of sequential recommendation. Formally, given the contextual item sequence of a user $c = \{i_1, \cdots, i_n\}$, the task of sequential recommendation is to predict the next item that the user is likely to interact with at the $(n+1)$-th step, denoted as $p(i_{n+1}|i_{1:n})$.

Method

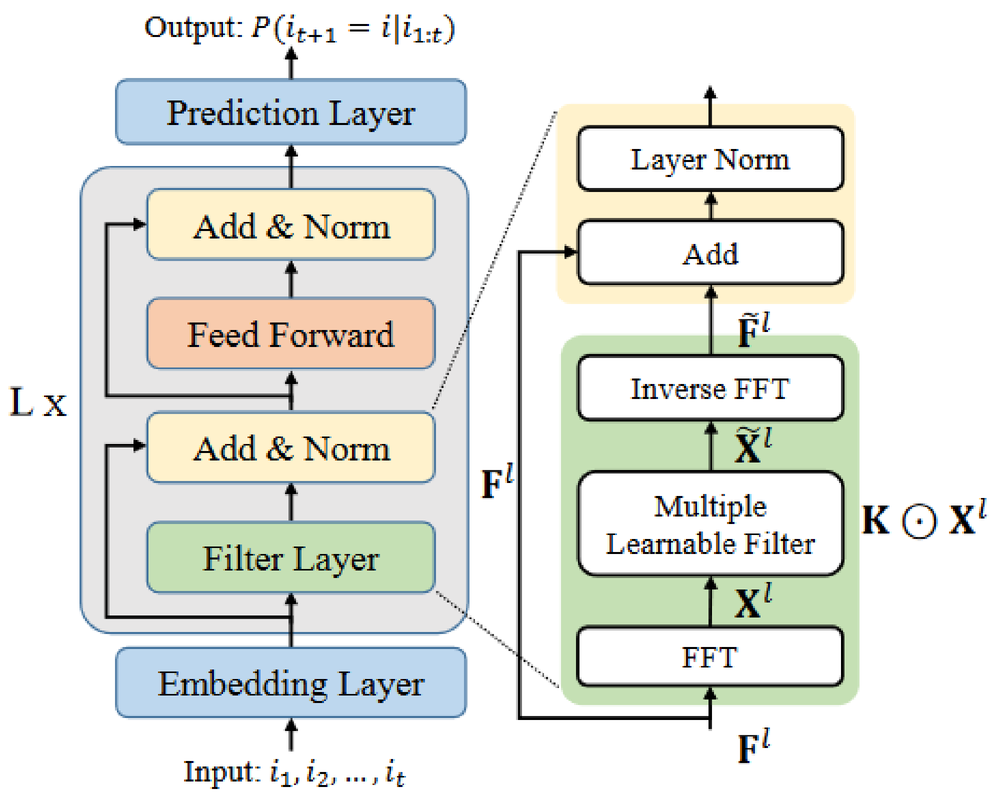

Figure 2: The overview of our FMLP-Rec, an all-MLP model that stacks multiple learnable filter-enhanced blocks.

$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} nk}, \quad 0 \le k \le N-1. \tag{1}$$

$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} nk}. \tag{2}$$

$$\mathbf{E}_I = Dropout(LayerNorm(\mathbf{E} + \mathbf{P})). \tag{3}$$

$$\mathbf{X}^l = \mathcal{F}(\mathbf{F}^l) \in \mathbb{C}^{n \times d} \tag{4}$$

$$\tilde{\mathbf{X}}^l = \mathbf{W} \odot \mathbf{X}^l, \tag{5}$$

$$\tilde{\mathbf{F}}^l \leftarrow \mathcal{F}^{-1}(\tilde{\mathbf{X}}^l) \in \mathbb{R}^{n \times d}. \tag{6}$$

$$\tilde{\mathbf{F}}^l = \mathrm{LayerNorm}(\mathbf{F}^l + \mathrm{Dropout}(\tilde{\mathbf{F}}^l)) \tag{7}$$

$$\mathrm{FFN}(\tilde{\mathbf{F}}^l) = (\mathrm{ReLU}(\tilde{\mathbf{F}}^l \mathbf{W}_1 + \mathbf{b}_1))\mathbf{W}_2 + \mathbf{b}_2, \tag{8}$$

$$P(i_{t+1} = i|i_{1:t}) = \mathbf{e}_i^\top \mathbf{F}_t^L,$$

$$L = -\sum_{u \in \mathcal{U}} \sum_{t=1}^{n} \log \sigma \Big(P(i_{t+1}|i_{1:t}) - P(i_{t+1}^-|i_{1:t}) \Big), \tag{10}$$

Experiments

Table 1: Performance comparison of SASRec and GRU4Rec with different filtering algorithms.

Methods	Filter	Beauty		Sports	
		HR@10	NDCG@10	HR@10	NDCG@10
GRU4Rec		0.4106	0.2584	0.4299	0.2527
	+HPF	0.3828	0.2228	0.3654	0.2063
	+LPF	0.4351	**0.2689**	**0.4481**	**0.2578**
	+BSF	**0.4372**	0.2658	0.4432	0.2563
SASRec		0.4696	0.3156	0.4622	0.2869
	+HPF	0.4544	0.3037	0.4530	0.2785
	+LPF	0.4941	0.3320	0.5040	0.3138
	+BSF	**0.5011**	**0.3334**	**0.5115**	**0.3172**

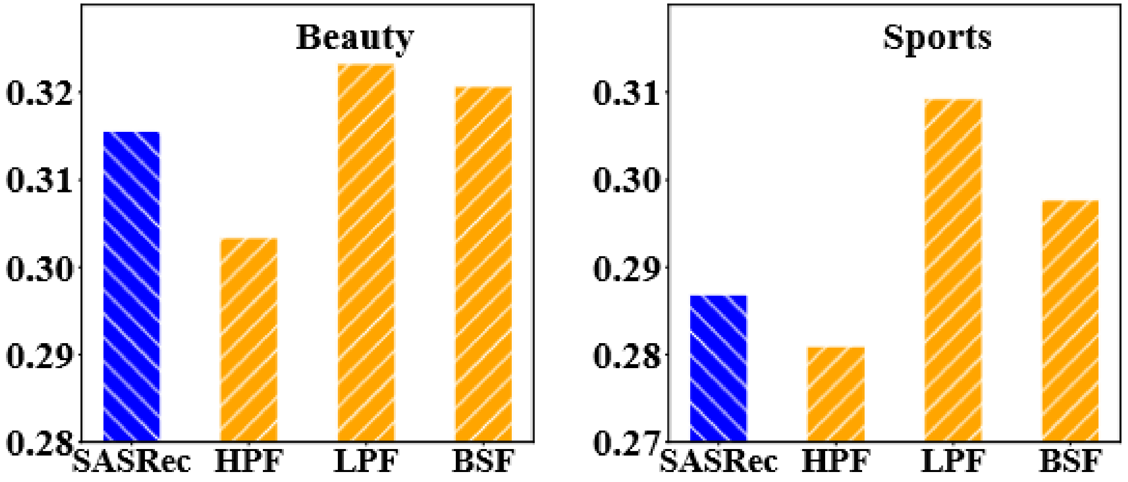

Figure 1: Performance (NDCG@10) comparison of all-MLP variants of SASRec with different filtering algorithms.

Experiments

Table 2: Time complexity and receptive field of the proposed FMLP-Rec with Caser and SASRec, where n and k denote the sequence length and convolution kernel size, respectively. For simplicity, we omit other same terms (*e.g.*, hidden size) and highlight the difference at the sequence level.

	Time Complexity per Layer	Receptive Field
Caser	$O(kn)$	k
SASRec	$O(n^2)$	n
FMLP-Rec	$O(n \log n)$	n

Table 3: Statistics of the datasets after preprocessing.

Dataset	# Sequences	# Items	# Actions	# Sparsity
Beauty	22,363	12,101	198,502	99.93%
Sports	25,598	18,357	296,337	99.95%
Toys	19,412	11,924	167,597	99.93%
Yelp	30,431	20,033	316,354	99.95%
Nowplaying	145,612	59,593	1,085,410	99.99%
Retailrocket	321,032	51,428	871,637	99.99%
Tmall	66,909	37,367	427,797	99.98%
Yoochoose	470,477	19,690	1,434,349	99.98%

Experiments

Table 4: Performance comparison of different methods on four datasets containing user transaction records. The best performance and the second best performance methods are denoted in bold and underlined fonts respectively.

Datasets	Metric	PopRec	FM	AutoInt	GRU4Rec	Caser	HGN	RepeatNet	CLEA	SASRec	BERT4Rec	SRGNN	GCSAN	FMLP-Rec
Beauty	HR@1	0.0678	0.0405	0.0447	0.1337	0.1337	0.1683	0.1578	0.1325	0.1870	0.1531	0.1729	0.1973	**0.2011**
	HR@5	0.2105	0.1461	0.1705	0.3125	0.3032	0.3544	0.3268	0.3305	0.3741	0.3640	0.3518	0.3678	**0.4025**
	NDCG@5	0.1391	0.0934	0.1063	0.2268	0.2219	0.2656	0.2455	0.2353	0.2848	0.2622	0.2660	0.2864	**0.3070**
	HR@10	0.3386	0.2311	0.2872	0.4106	0.3942	0.4503	0.4205	0.4426	0.4696	0.4739	0.4484	0.4542	**0.4998**
	NDCG@10	0.1803	0.1207	0.1440	0.2584	0.2512	0.2965	0.2757	0.2715	0.3156	0.2975	0.2971	0.3143	**0.3385**
	MRR	0.1558	0.1096	0.1226	0.2308	0.2263	0.2669	0.2498	0.2376	0.2852	0.2614	0.2686	0.2882	**0.3051**
Sports	HR@1	0.0763	0.0489	0.0644	0.1160	0.1135	0.1428	0.1334	0.1114	0.1455	0.1255	0.1419	**0.1669**	0.1646
	HR@5	0.2293	0.1603	0.1982	0.3055	0.2866	0.3349	0.3162	0.3041	0.3466	0.3375	0.3367	0.3588	**0.3803**
	NDCG@5	0.1538	0.1048	0.1316	0.2126	0.2020	0.2420	0.2274	0.2096	0.2497	0.2341	0.2418	0.2658	**0.2760**
	HR@10	0.3423	0.2491	0.2967	0.4299	0.4014	0.4551	0.4324	0.4274	0.4622	0.4722	0.4545	0.4737	**0.5059**
	NDCG@10	0.1902	0.1334	0.1633	0.2527	0.2390	0.2806	0.2649	0.2493	0.2869	0.2775	0.2799	0.3029	**0.3165**
	MRR	0.1660	0.1202	0.1435	0.2191	0.2100	0.2469	0.2334	0.2156	0.2520	0.2378	0.2461	0.2691	**0.2763**
Toys	HR@1	0.0585	0.0257	0.0448	0.0997	0.1114	0.1504	0.1333	0.1104	0.1878	0.1262	0.1600	**0.1996**	0.1935
	HR@5	0.1977	0.0978	0.1471	0.2795	0.2614	0.3276	0.3001	0.3055	0.3682	0.3344	0.3389	0.3613	**0.4063**
	NDCG@5	0.1286	0.0614	0.0960	0.1919	0.1885	0.2423	0.2192	0.2102	0.2820	0.2327	0.2528	0.2836	**0.3046**
	HR@10	0.3008	0.1715	0.2369	0.3896	0.3540	0.4211	0.4015	0.4207	0.4663	0.4493	0.4413	0.4509	**0.5062**
	NDCG@10	0.1618	0.0850	0.1248	0.2274	0.2183	0.2724	0.2517	0.2473	0.3136	0.2698	0.2857	0.3125	**0.3368**
	MRR	0.1430	0.0819	0.1131	0.1973	0.1967	0.2454	0.2253	0.2138	0.2842	0.2338	0.2566	0.2871	**0.3012**
Yelp	HR@1	0.0801	0.0624	0.0731	0.2053	0.2188	0.2428	0.2341	0.2102	0.2375	0.2405	0.2176	0.2493	**0.2727**
	HR@5	0.2415	0.2036	0.2249	0.5437	0.5111	0.5768	0.5357	0.5707	0.5745	0.5976	0.5442	0.5725	**0.6191**
	NDCG@5	0.1622	0.1333	0.1501	0.3784	0.3696	0.4162	0.3894	0.3955	0.4113	0.4252	0.3860	0.4162	**0.4527**
	HR@10	0.3609	0.3153	0.3367	0.7265	0.6661	0.7411	0.6897	0.7473	0.7373	0.7597	0.7096	0.7371	**0.7720**
	NDCG@10	0.2007	0.1692	0.1860	0.4375	0.4198	0.4695	0.4393	0.4527	0.4642	0.4778	0.4395	0.4696	**0.5024**
	MRR	0.1740	0.1470	0.1616	0.3630	0.3595	0.3988	0.3769	0.3751	0.3927	0.4026	0.3711	0.4006	**0.4299**

Experiments

Table 5: Performance comparison of different methods on four session-based datasets. Since these datasets do not have attribute information and the item sequences are usually shorter, we remove several improper baseline methods.

Datasets	Metric	PopRec	GRU4Rec	Caser	HGN	RepeatNet	CLEA	SASRec	SRGNN	GCSAN	FMLP-Rec
Nowplaying	HR@1	0.0757	0.4035	0.3435	0.3491	0.3350	0.2728	0.4396	0.3819	0.4447	**0.4731**
	HR@5	0.2197	0.6829	0.6267	0.6026	0.5257	0.5575	0.7042	0.6028	0.6728	**0.7262**
	NDCG@5	0.1480	0.5536	0.4942	0.4835	0.4355	0.4226	0.5812	0.4986	0.5658	**0.6094**
	HR@10	0.3318	0.7720	0.7318	0.6992	0.6110	0.6766	0.7968	0.7007	0.7616	**0.8081**
	NDCG@10	0.1841	0.5825	0.5283	0.5149	0.4631	0.4612	0.6113	0.5304	0.5946	**0.6360**
	MRR	0.1602	0.5314	0.4752	0.4677	0.4297	0.4070	0.5615	0.4892	0.5520	**0.5895**
Retailrocket	HR@1	0.0817	0.7202	0.6871	0.6432	0.6744	0.3410	0.7588	0.7408	**0.7773**	0.7736
	HR@5	0.2315	0.8597	0.8348	0.7650	0.7724	0.6139	0.8769	0.8373	0.8650	**0.8810**
	NDCG@5	0.1577	0.7982	0.7689	0.7098	0.7270	0.4853	0.8250	0.7937	0.8259	**0.8338**
	HR@10	0.3388	0.8925	0.8719	0.8036	0.8112	0.7296	0.9024	0.8684	0.8901	**0.9060**
	NDCG@10	0.1922	0.8089	0.7809	0.7223	0.7395	0.5227	0.8333	0.8037	0.8340	**0.8419**
	MRR	0.1694	0.7858	0.7563	0.7029	0.7234	0.4702	0.8145	0.7878	0.8199	**0.8247**
Tmall	HR@1	0.1025	0.4178	0.3288	0.4467	**0.5556**	0.2895	0.4045	0.4026	0.4650	0.5173
	HR@5	0.2264	0.5855	0.5066	0.5793	0.6090	0.4573	0.5478	0.5335	0.5940	**0.6565**
	NDCG@5	0.1647	0.5062	0.4230	0.5168	0.5826	0.3768	0.4792	0.4710	0.5329	**0.5911**
	HR@10	0.2967	0.6636	0.5943	0.6471	0.6494	0.5478	0.6275	0.6082	0.6591	**0.7206**
	NDCG@10	0.1874	0.5315	0.4513	0.5386	0.5956	0.4060	0.5049	0.4950	0.5538	**0.6118**
	MRR	0.1723	0.5021	0.4209	0.5168	**0.5894**	0.3775	0.4804	0.4736	0.5328	0.5879
Yoochoose	HR@1	0.1794	0.7208	0.7041	0.6278	0.7450	0.5106	0.7611	0.7575	**0.7855**	0.7749
	HR@5	0.4990	0.8950	0.8818	0.8425	0.8673	0.7628	0.8976	0.8840	0.9073	**0.9084**
	NDCG@5	0.3432	0.8189	0.8026	0.7460	0.8123	0.6465	0.8375	0.8280	**0.8535**	0.8499
	HR@10	0.6574	0.9273	0.9172	0.8927	0.9001	0.8436	0.9273	0.9137	0.9320	**0.9359**
	NDCG@10	0.3948	0.8294	0.8141	0.7623	0.8230	0.6727	0.8471	0.8376	**0.8616**	0.8588
	MRR	0.3276	0.8004	0.7838	0.7246	0.8019	0.6257	0.8240	0.8164	**0.8414**	0.8364

Experiments

Table 6: Ablation study of our FMLP-Rec, we report NDCG@10 on Beauty and Sports datasets.

	Beauty		Sports	
	HR@10	NDCG@10	HR@10	NDCG@10
FMLP-Rec	**0.4998**	**0.3385**	**0.5059**	**0.3165**
w/o Filter Layer	0.4317	0.2914	0.4243	0.2604
w/o FFN	0.4607	0.3067	0.4594	0.2838
w/o Add & Norm	0.4654	0.2919	0.4831	0.2860
+HPF	0.4567	0.3034	0.4595	0.2810
+LPF	0.4847	0.3233	0.4949	0.3093
+BSF	0.4822	0.3207	0.4835	0.2977

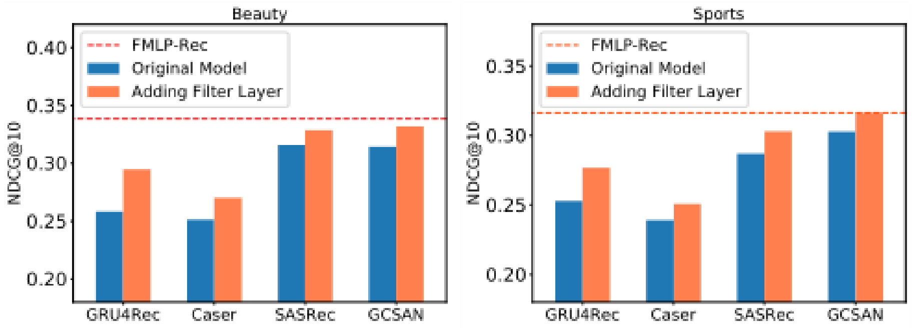

Figure 3: Performance (NDCG@10) comparison of different models enhanced by our learnable filters.

Thanks