A Multi-task Learning Framework with Features Based on Behavioral Pattern Conversation

Bo Tang[1,2], Nan Wang[1,2(✉)], Jinbao Li[3,4(✉)], and Zhonghui Shen[1]

[1] College of Computer Science and Technology, Heilongjiang University, Harbin, China
2211889@s.hlju.edu.cn, wangnan@hlju.edu.cn
[2] Key Laboratory of Database and Parallel Computing, Heilongjiang University, Harbin, China
[3] Shandong Artificial Intelligence Institute, Qilu University of Technology, Jinan, China
lijinb@sdas.org
[4] School of Mathematics and Statistics, Qilu University of Technology, Jinan, China

Abstract. Deep neural network based multi-task learning has been widely successful in many real-world large scale applications, such as recommendation systems. A large number of commodity transaction data from e-commerce platforms show that users often go through a series of behavioral transitions such as impressed → click → add shopping cart before finally forming a purchase behavior, and only very few users click and then make a purchase directly. This phenomenon precisely follows the objective reality of power-law distribution. Based on this, this paper proposes a Multi-task learning framework with features based on Behavioral Pattern Conversation (BPCM). A feature tower model based on attribute information is constructed in the framework, which is able to control the fusion and screening process of features adaptively through the designed novel gate complementary mechanism. In addition, we designed several submodules with behavioral pattern conversation (BPC) applied to multi-task learning. The class of modules is not only able to adaptively model the sequentiality and dependencies between behavioral task transitions through information transfer, but also to effectively control the amount of information transferred between different tasks. Adequate experiments show that our BPCM obtains higher performance compared to more current advanced multi-task learning frameworks.

Keywords: Multi-task Learning · Behavioral Pattern Conversation · Recommendation System

Supported by National Natural Science Foundation of China (No. 62172243), Heilongjiang Provincial Natural Science Foundation of China (No. LH2021F047).

A. Zaslavsky et al. (Eds.): MobiQuitous 2023, LNICST 594, pp. 256–267, 2024.
https://doi.org/10.1007/978-3-031-63992-0_17

1 Introduction

In recent years deep neural network models have been successfully applied to many real-world, large-scale applications, such as recommendation systems [5]. In the real world, there is a rich variety of forms of user feedback on things. Take e-commerce as an example, users' behaviors such as clicking, collecting, adding shopping cart and buying can reflect different interests of users in items from different perspectives. As a result, much of the work in recommendation systems has focused on dealing with the preference of different behaviors of users for items [4]. However, different behavioral patterns of users are closely related to each other [6], and their behaviors are often ordered and interdependent, most likely affecting the strength of the user's acceptance of items at different steps. For example, users tend to browse items and then make a purchase decisions after adding shopping cart. The sequentiality and dependability between these actions is often overlooked.

As shown in the example in Fig. 1, a user has recently browsed apparel items (shoes, T-shirts, pants, etc.) and electronic items (cell phones, game consoles, etc.), and then the user further added apparel (shoes, T-shirts) and electronic items (cell phones) to the shopping cart, and finally the user purchased only two apparel items (shoes, T-shirts). By further obtaining information on the features of these two categories, we found that the user's current interests and purchase intentions are in fact governed by factors such as a certain "price" range and some "brands" of clothing, in addition to individual preferences. Therefore, fully considering the feature information of interactive goods can also effectively capture the potential interest of users. In addition, utilizing feature information between user behavior conversions can also rationalize the sequences and dependencies between user behavior patterns.

It is worth noting that recommender systems do not want users to display only a single goal, such as CTR (Click-Through-Rate); they often want users to display more goals to fulfill multiple user needs [2].

In addition, in the academic field, multi-task learning is a typical approach to solve the end-to-end conversion problem [17]. Since all tasks in the learning process use the same model and need to share the underlying data, it is equally important to handle the underlying data flow in a rational way. In recent years, modeling of inter-task relationships in multi-task learning has been partially achieved, and the most common approach is to use expert models [9,13].

The main contributions of this paper are as follows:

- 1. A multi-task learning framework with features based on behavioral pattern conversation (BPCM) is proposed. The framework effectively exploits the sequential relationships existing between user behavioral patterns and the interdependencies among the tasks processed to solve the multi-task learning problem under multiple behavioral patterns.
- 2. We proposed a feature tower model (Feature Tower) based on behavioral interaction information. In addition to making full use of the interaction sequence information of different behaviors, additional analysis takes into account the feature information of users and interacted items.

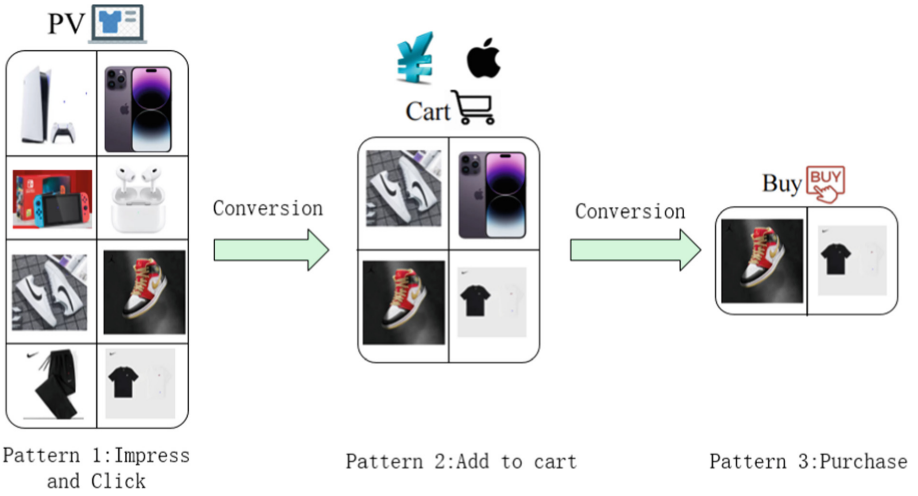

Fig. 1. Illustration of behaviors with sequential and dependent instructions through the user's history.

- 3. A Behavior Pattern Conversion (BPC) module is intelligently designed to adaptively learn the information to be passed between different behavioral task phases and the magnitude of the information. This module not only effectively guarantees the interdependence between different behavioral patterns, but also more strictly constrains the sequential nature in the behavioral pattern conversion task, and obtains more accurate prediction results.
- 4. Comprehensive experiments have verified that the model in this paper has better performance compared to recent more advanced models with the support of real large e-commerce platform datasets.

2 Related Works

In this section we introduce the related thesis of multi-task learning and prior behavior transformation, briefly describing the technical background and the current state of research.

2.1 Recommendation Model Based on Behavior Conversion

Several previous works have proposed some solutions to the end-to-end conversion rate problem and the sequential dependence problem among multiple tasks. For example, Ma et al. proposed the full-space task model (ESMM) [6]. Interlevel click-through rate (CTR) and conversion rate prediction (CVR) use dot product operations to pass the probabilities in the output layer. This method does not effectively utilize deep networks, and thus has average performance in more complex and task-unspecific application scenarios. The mixed sequence

expert model (MoSE) has also been proposed to model sequential user behavior in multi-task learning, but the lack of a tower model to guide the learning process between tasks leads to no information exchange in tasks and thus no improvement in the relationship of multiple behaviors [7]. Another method applied to advertisement analysis Adaptive Information Transfer Multitasking Framework (AITM) [17]. The framework is a more advanced and complete model for solving some of the behavior conversion problems, however, it does not filter and control the effective information and the amount of information between behaviors.

2.2 Multi-task Learning

Multi-task learning has been successful in many fields of machine learning, such as natural language processing, speech recognition, and computer vision. Multi-task models can learn commonalities and differences between different tasks in such a way that similar or identical parts between tasks can be handled equally, but at the same time there is differentiation to learn parts that are more different between tasks. This allows to improve the learning efficiency and model quality of each task [11]. Cauruana proposed a widely used multi-task learning model [3]. The model has a shared underlying model structure that can greatly reduce the risk of model overfitting. However, the variability between tasks may bring about conflicts in the parameter optimization section, and different tasks will inevitably interact with each other during the optimization process. Therefore, a more advanced way of underlying data sharing and task processing allocation is eagerly sought to handle the multi-task learning problem. Mixed expert models are often identified as an effective way to handle multi-task learning, with the basic idea that the underlying data is shared but each task is processed separately when multitasking.

3 Method

In this paper, a BPCM framework is constructed to solve the multi-task learning problem with multiple behavior patterns. The study is mainly based on an e-commerce platform and uses the generally accepted sequential relationship in this model to define a sequence of behavioral task conversions steps [17]. It is worth stating that the behaviors in this sequence of relations are sequential and monotonically ordered, except that all other task recurrence relations are not legal. Since the datasets of different platforms are very different, it is not possible to define a perfect sequentiality for all datasets, so it is sufficient to define behavioral inter-task recurrence relationships that fit the dataset and are reasonable [15]. The feature tower models consists of three components: 1) feature fusion self-attention module; 2) feature-aware aggregation module; and 3) feature tower complementary gating unit. The overall structure of BPCM is shown in Fig. 2.

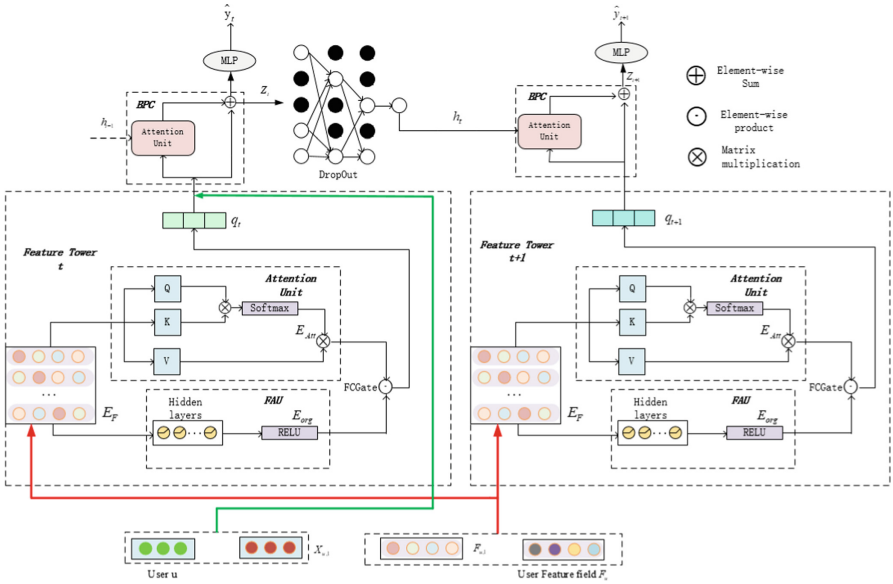

Fig. 2. Illustration of the proposed BPCM model. The middle figure shows the feature tower module corresponding to the current behavior task; the upper figure shows the Behavior Pattern Conversion (BPC) module and the information transfer process between behavior tasks; the lower figure illustrates the input and embedding module.

3.1 Problem Formulation

Given a set of users $U = \{u_1, u_2, \ldots, u_H\}$, where H is the number of users, and X_u is a set of the sequence of historical interactions of the user u, $X_u = \{X_{u,1}, X_{u,2}, ..., X_{u,I}\}$, where $X_{u,I}$ is the I-th item that the user interacted with and I is the number of items. $F_{u,i}$ denotes the set of features of item i that is interacted with user u, $F_{u,i} = \{f_{i,1}, f_{i,2}, ..., f_{i,N}\}$, where $f_{i,N}$ represents the N-th features of item i (e.g., the price of the item, the store it belongs to, etc.), N is the number of item feature information. F_u is the information about the user's features (e.g. user's age, spending power, etc.) denoted as $F_u = \{f_{u,1}, f_{u,1}, ..., f_{u,M}\}$, where M is the number of user feature information. y_t represents the label of the current task (e.g., 1 if clicked, 0 otherwise), and y_{t+1} represents the label of the next task.

3.2 Input Module

In order to avoid too long and redundant input data, we adopt the user-item pair as the model input, and append the feature information of users and items.

$$x_i = \{u_i, X_{u,i}, F_{u,i}, F_u\} \tag{1}$$

We generate the embedding matrix from the user id, item id and the respective set of corresponding features, which can transform the input of the model into a low-dimensional vector [16].

3.3 Feature Tower Module

Feature Fusion Self-attention Unit. The e-commerce data set is rich in Item features, such as price, brand, store name, category, region, etc. Empirically, different features have different effects on model performance [8]. In the BPCM framework, we introduce the self-attentive mechanism [14] to learn the relevance scores between the respective item features in the historical interaction sequence. The final output of the self-attentive unit is the integral feature matrix with adaptive relevance weights. The formula is as follows:

$$\begin{bmatrix} Q \\ K \\ V \end{bmatrix} = E_F \begin{bmatrix} W_Q \\ W_X \\ W_V \end{bmatrix} \tag{2}$$

$E_F \in \mathbb{R}^{I \times d}$ is the initial Embedding matrix of feature information in the input sequence x_i. The weight matrix $W_Q, W_K, W_V \in \mathbb{R}^{d \times d}$ represents the query vector, the key vector and the value vector, respectively. Then we use the softmax function to dot product the query vector (Q) with the key vector (K) to obtain the following formula for the attention matrix on the value vector (V).

$$E_{Att} = Attention(Q, K, V) = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}}\right)V \tag{3}$$

$E_{Att} \in \mathbb{R}^{I \times d}$ represents the output of the self-attentive mechanism. $\sqrt{d}$ plays a regulating role as a scaling factor so that the value of the inner product is not too large to affect subsequent calculations with smaller inner product values.

Feature-Aware Aggregation Unit (FAU). The self-attentive mechanism introduced in the previous section only uses pairwise feature interactions to represent the relative importance between features, but fails to further refine the learning of features using more comprehensive feature information. Therefore, we designed a feature-aware aggregation module component to aggregate the feature encoding relationships between users and items.

$$E_{\text{org}} = RELU\left(W_2\left(RELU\left(W_1 E_F + b_1\right)\right) + b_2\right) \tag{4}$$

where W_1 and W_2, b_1 and b_2 represent the trainable parameters of the first layer of the deep network and the trainable parameters of the second layer of the deep network, respectively. E_F stands for embedding the initial feature information of the input sequence as the input to the MLP. $E_{org} \in \mathbb{R}^{I \times d}$ then represents the native feature aggregation information of the input instance.

Feature Complementary Gate Unit (FCGate). Based on the GRU gate mechanism [3], we designed a novel complementary gating mechanism to control

the fusion and filtering process of all features. FCGate consists of three components: 1) feature fusion self-attention matrix E_{Att}; 2) native feature aggregation matrix E_{org}; 3) weight matrix W_b. $W_b \in \mathbb{R}^{I \times d}$ represents the learned gate signal.

$$W_b = \delta\left(W_C E_{org} + W_D E_{Att}\right) \tag{5}$$

δ represents the function Sigmoid, $W_C \in \mathbb{R}^{d \times d}$ and $W_D \in \mathbb{R}^{d \times d}$ represent trainable parameters in the gate control unit. The final output of the complementary gating unit is as follows:

$$q_t = E_{Att} \odot W_b + E_{\text{org}} \odot (1 - W_b) \tag{6}$$

where $q_t \in \mathbb{R}^{I \times d}$ represents the integral output of the feature tower module.

3.4 Behavior Pattern Conversion Module (BPC)

The purpose of the BPC module is to adaptive learn the transfer of information between behavioral tasks. Given T behavioral tasks, the output of the feature tower module corresponding to each task $t(1 \leq t \leq T)$ is q_t. For two adjacent tasks t and task $t+1$, since not all the information in the previous task t is worth passing to the next step task $t+1$. If all of them are imported, it will cause unavoidable information redundancy and overfitting, so the amount of information to be passed down needs to be traded off to solve this problem. Combining the previous methods of Layer Normalization [1] and Dropout [12] in Transformer [14]. We propose the following approach as a mode of passing between behavioral tasks.

$$Z_{t+1} = BPC\left(q_{t+1}, h_t\right) \tag{7}$$

$Z_t \in \mathbb{R}^I$ is the output corresponding to the task t in the BPC module.

$$h_t = msg\left(Z_t\right) \tag{8}$$

$msg\left(\right)$ is a function of how much information should be learned between task t and task $t+1$. The method is not unique, and in the paper the Dropout data augmentation method is used.

The BPC is designed as a module that adaptively assigns weights h_{t-1} to the passed information and the original information q_t of the current task. Previous work has also demonstrated that attention mechanisms are more effective in multi-tasking [18], and thus this effect can be achieved simply and effectively by reusing the self-attention mechanism in the feature tower model in this paper.

$$BPC() = \text{ Attention } \left(q_{t+1}, h_t\right) + q_{t+1} \tag{9}$$

In addition, for the first task, the output of the feature tower model is taken directly as the output of the first task in the initialization of the framework, that is $Z_1 = q_1$ when $t = 1$.

3.5 Prediction Module

In the prediction module we take a simple MLP approach with a final sigmoid function activation. It is additionally worth stating that the module output Z_t is used for both the current behavioral task prediction and the transmission to the next phase of the behavioral task. The predicted probability for the t-th task is

$$\hat{y}_t = \text{Sigmoid}\left(MLP\left(Z_t\right)\right) \tag{10}$$

3.6 Model Training and an Additional Compensation Loss

In the classification task, we choose the cross-entropy loss function to measure the model performance and training effect.

$$L_{Cl}(\theta) = -\frac{1}{N}\sum_{t=1}^{T}\sum_{(x_t,y_t\varepsilon D}^{N}\left(\left(y_t \log \hat{y}_t + (1-y_t)\log\left(1-\hat{y}_t\right)\right)\right) \tag{11}$$

N is the number of all samples in the sample space D that contains the inputs and labels. y_t is the true label under the t-th behavioral task and $\hat{y}_t$ is the predicted label under the t-th behavioral task.

θ is the setting hyper parameter in the BPCM framework. In addition, we introduce an additional calibrator $\mathcal{L}_{cr}(\theta)$ to minimize the task objective.

$$\mathcal{L}_{cr}(\theta) = -\frac{1}{N}\sum_{t=2}^{T}\sum_{(x_1\in D)}^{N}\max\left(\hat{y}_{t+1}-\hat{y}_t, 0\right) \tag{12}$$

Finally, we get an additional compensation Loss $\mathcal{L}(\theta)$, which combines the two components into a loss function.

$$\mathcal{L}(\theta) = \mathcal{L}_{cl}(\theta) + \alpha\mathcal{L}_{cr}(\theta) \tag{13}$$

α is controllable calibrator component weights.

4 Experiment

4.1 Datasets

– Ali-CCP Dataset is a public dataset called Ali-CCP (Alibaba Click and Conversion Prediction). This dataset is collected from the logs of mobile Taobao. Users can click on the products they are interested in from the impression results, or make further purchases. Thus the user's behavior can be summarized in a sequential pattern 'impression to click to purchase'. The dataset contains three main parts, the sample id part, the label part, and the feature part. In addition, we used all single-valued features in the feature domain and did not do any preprocessing on the features.

– TaoBao Dataset is a dataset of Taobao display ad click rate prediction. The dataset contains ad display/click log information of 1.14 million users randomly selected from the Taobao website over an 8-day period. It contains user id, ad group id, timestamp and other ad domain features. We use the data of the first seven days as training samples and the data of the eighth day as test samples.

The specific statistical information of the two datasets is shown in Table 1.

4.2 Evaluation Metrics

Following previous work, we used the widely adopted area under receiver operating characteristic curve (AUC) for evaluation in the classification task. Based on the work [14], we additionally introduced a metric called RelaImpr to more intuitively measure the degree of improvement of our model based on the baseline model. Since the AUC value of the stochastic strategy is about 0.5, the RelaImpr in this task is defined as

$$\text{RelaImpr} = \frac{AUC(\text{measured}) - 0.5}{AUC(\text{base}) - 0.5} - 1 \tag{14}$$

Table 1. Dataset statistics. "Positive" represents the percentage of positive samples in the train set.

DataSet	#Train	#Validation	#Test	Feature Num	%Positive
Ali-CCP	38M	4.2M	43M	18	3.89/0.02
TaoBao	16M	1.34M	13.4M	6	1.13/0.05

4.3 Baselines

We compare the proposed model with the following advanced and mainstream models. To demonstrate the effectiveness of our proposed BPCM, we compare the proposed model with the following advanced and mainstream models.

– MLP [10]: as the base baseline model, we use the basic structure of the AITM framework as a single task model. It is a multi-layer perceptron.
– OMoE [9]: the OMoE with an expert model bottom data sharing layer, which integrates the expert model by sharing the same gating across all tasks.
– MMoE [9]: the MMoE is an improvement of the OMoE and integrates the expert model through multiple gating on the basis of the OMoE.
– PLE [13]: the Progressive Layered Extraction (PLE) explicitly separates the task from the underlying data sharing layer using a progressive layered extraction approach based on an expert model.

– AITM [17]: the Adaptive Information Transfer Multi-task (AITM) has an advanced explicit expert model as the underlying data sharing, and also separates the tower model to handle the corresponding tasks separately.

We used MLP as the base comparison baseline model, and the results of the comparison experiment are shown in Table 2.

4.4 Ablation Study

In this section we conducted experiments on the Ali-CCP and TaoBao datasets, respectively. In this way, we analyze the significant impact of each component or design pattern in the BPCM model proposed in this paper on the multi-task prediction performance. As shown in Table 3.

– *w/o* Att denotes the removal of the feature fusion self-attentiveness mechanism, which makes the features between users and item no longer have relevance distinction.
– *w/o* Org denotes the removal of the feature-aware aggregation module. This makes the representation of the original characteristics of the multi-domain features lost in the initial feature fusion of the user with the item disappears.
– *w/o* Gate denotes the removal of the complementary gating unit part of the work. Instead, the approach is to directly sum the two parts of the representation that self-attentive fusion and initial feature fusion.
– *w/o* BPC indicates the removal of the behavior pattern conversion module. The experimental results show that the role of this module is essential. The purpose of this module is to coordinate between tasks based on different labels, while determining how much information to pass between multiple tasks.

Table 2. Performance comparisons on two real-world datasets with five baselines

Model	Ali-CCP				TaoBao			
	click AUC	purchase AUC	RelaImpr		impression AUC	click AUC	RelaImpr	
MLP	0.6035	0.5841	–	–	0.5771	0.5770	–	–
OMoE	0.6031	0.6405	−0.0386	+0.6706	0.5918	0.5887	+0.1906	+0.1519
MMoE	0.6042	0.6420	+0.0067	+0.6884	0.5922	0.5891	+0.1958	+0.1571
PLE	0.6039	0.6417	+0.0038	+0.6848	0.5852	0.5858	+0.1050	+0.1142
AITM	0.6051	0.6506	+0.0154	+0.7907	0.6191	0.6201	+0.5447	+0.5597
BPCM	**0.6192**	**0.6556**	**+0.1516**	**+0.8501**	**0.6245**	**0.6343**	**+0.6147**	**+0.7441**

4.5 Experimental Hyper-parameter Setting

The set of experiments is performed on the Ali-CCP dataset with the embedding size adjusted to $\{5, 16, 32, 64, 120\}$ and the loss function weights adjusted to $\{0.2, 0.4, 0.6, 0.8, 1.0\}$, respectively. The loss weights have a greater influence on

Table 3. Ablation study of BPCM.

Model	Ali-CCP		TaoBao	
	click AUC	purchase AUC	impression AUC	click AUC
w/o Att	0.6183	0.6240	0.6191	0.6183
w/o Org	0.6178	0.6286	0.6076	0.6096
w/o Gate	0.6185	0.6447	0.6187	0.6186
w/o BPC	0.6118	0.5324	0.5645	0.5514
BPCM	**0.6192**	**0.6556**	**0.6245**	**0.6342**

the experimental results, i.e., the influence of the loss function corrector is greater in the prediction process. It is inevitable that the prediction probability of the successor task is higher than the prediction probability of the precursor task due to the sparsity of the data during the model training. After graphical analysis, the model has the best comprehensive performance and the smoothest training process at embedding size $d = 5$ while $\alpha = 0.6$. At this point, the embedding dimension and loss function hyperparameters are optimally balanced.

5 Conclusion

In this work, we explore the problem of multi-task recommendation based on transitions between user behaviors. A new and improved end-to-end expert model framework BPCM is proposed, which realizes the goal by dealing with the underlying data sharing between different tasks and the information transfer between tasks. In addition, we introduce an additional compensation loss function during the training process to achieve an overall optimization of the model. Extensive experiments on two real e-commerce datasets show that our model significantly outperforms the other baseline models.

References

1. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization (2016)
2. Bansal, T., Belanger, D., Mccallum, A.: Ask the GRU: multi-task learning for deep text recommendations (2016)
3. Caruana, R.A.: Multitask learning: a knowledge-based source of inductive bias. Mach. Learn. Proc. **10**(1), 41–48 (1993)
4. Chen, C., Zhang, M., Liu, Y., Ma, S.: Social attentional memory network: modeling aspect- and friend-level differences in recommendation. In: The Twelfth ACM International Conference (2019)
5. Covington, P., Adams, J., Sargin, E.: Deep neural networks for YouTube recommendations. In: ACM Conference on Recommender Systems, pp. 191–198 (2016)
6. Guo, L., Hua, L., Jia, R., Zhao, B., Cui, B.: Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: The 25th ACM SIGKDD International Conference (2019)

7. Han, X., Hu, J., Ghosh, J.: MECATS: mixture-of-experts for quantile forecasts of aggregated time series. arXiv e-prints (2021)
8. Li, M., Lu, Z., Wu, Y., Li, Y.H.: BACPI: a bi-directional attention neural network for compound-protein interaction and binding affinity prediction. Bioinformatics **38**(7), 1995–2002 (2022)
9. Ma, J., Zhe, Z., Yi, X., Chen, J., Hong, L., Chi, E.H.: Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. ACM (2018)
10. Gardner, M.W., Dorling, S.R.: Artificial neural networks (the multilayer perceptron)-a review of applications in the atmospheric sciences. Atmos. Environ. **32**(14–15), 2627–2636 (1998)
11. Ruder, S.: An overview of multi-task learning in deep neural networks (2017)
12. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
13. Tang, H., Liu, J., Zhao, M., Gong, X.: Progressive layered extraction (PLE): a novel multi-task learning (MTL) model for personalized recommendations. In: RecSys 2020: Fourteenth ACM Conference on Recommender Systems (2020)
14. Vaswani, A., et al.: Attention is all you need. arXiv (2017)
15. Wan, M., Mcauley, J.: Item recommendation on monotonic behavior chains, pp. 86–94 (2018)
16. Wang, R., Shivanna, R., Cheng, D., Jain, S., Chi, E.: DCN V2: improved deep & cross network and practical lessons for web-scale learning to rank systems. In: WWW 2021: The Web Conference 2021 (2021)
17. Xi, D., Chen, Z., Yan, P., Zhang, Y., Chen, Y.: Modeling the sequential dependence among audience multi-step conversions with multi-task learning in targeted display advertising (2021)
18. Zhu, Y., et al.: Modeling users' behavior sequences with hierarchical explainable network for cross-domain fraud detection (2022)