ATRank: An Attention-Based User Behavior Modeling Framework for Recommendation

Alibaba Group

Key Laboratory of High Confidence Software Technologies, EECS, Peking University

2019.07

Background

- The user heterogeneous behaviors are hard to model
- The rating may only be related to very small parts of the user behaviors
  - Manually extract、RNN、CNN、LSTM could not perform well
- ATRank
  - Project all types of behaviors into multiple latent semantic spaces, where influence can be made among the behaviors via self-attention
  - Downstream applications then can use the user behavior vectors via vanilla attention

ATRank Framework

- Raw Feature Spaces
- Behavior Embedding Spaces
- Latent Semantic Spaces
- Self-Attention Layer
- Downstream Application Network

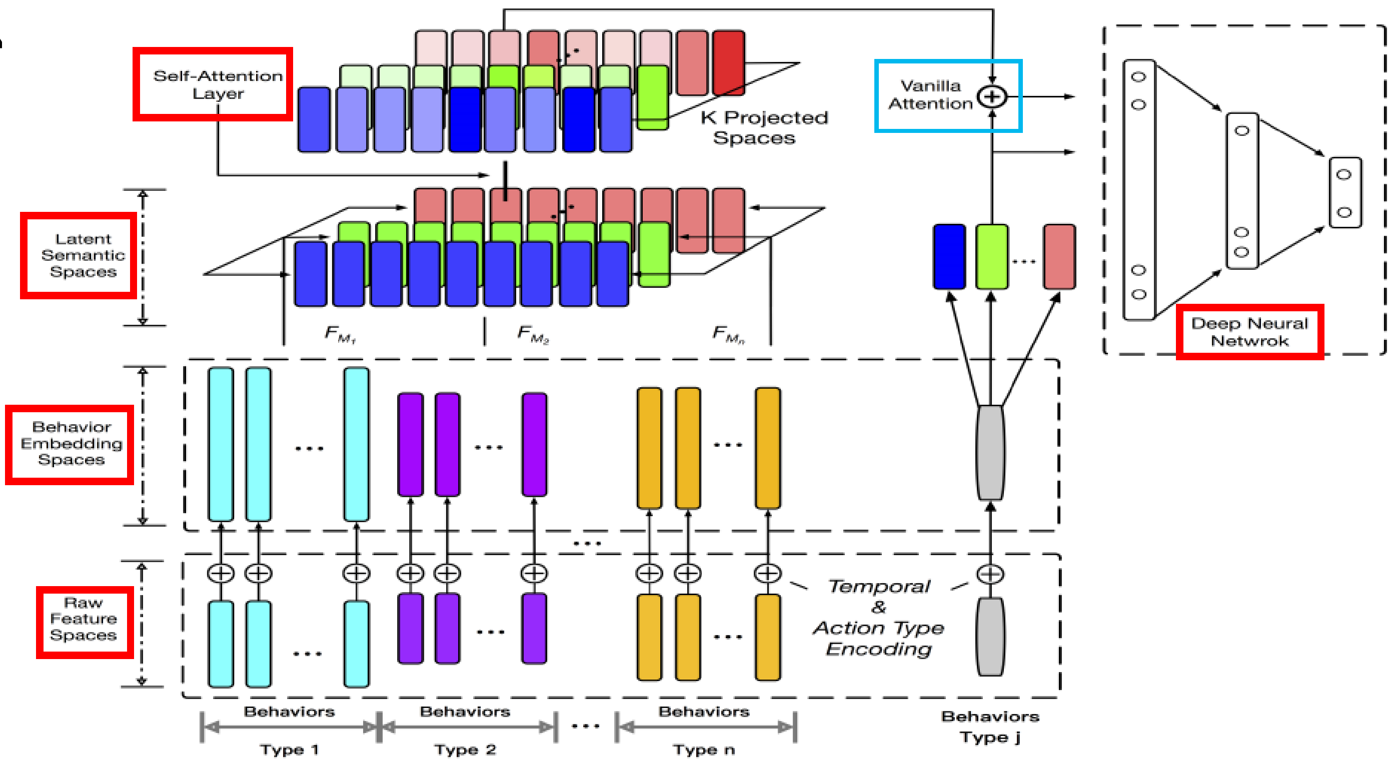


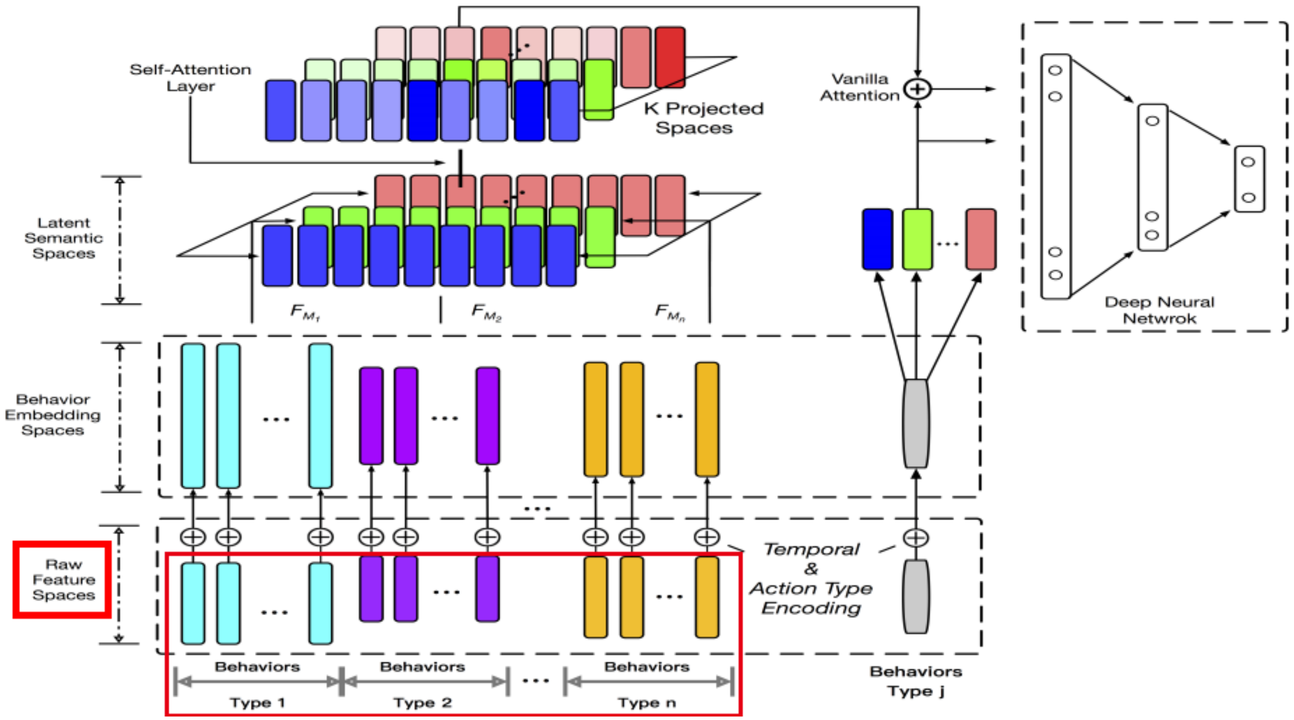
Figure 1: Attention-Based Heterogeneous Behaviors Modeling Framework

Definition

- A user behavior: a tuple{a,o,t} 三元组(动作类型，目标，时间)
  - a: action (E.g., 点击/收藏/加购、领取/使用)
  - o: object, is represented as all its belonging features (E.g., 商品，优惠券，关键字)
  - t: the timestamp
- A user can be represented as all his/her behaviors:
  - $\boxed{U = \{(a_j, o_j, t_j) | j = 1, 2, ..., m\}}$

Raw Feature Spaces

- Partition the user behavior tuples $U = \{(a_j, o_j, t_j) | j = 1, 2, ..., m\}$ into different behavior groups $G = \{bg_1, bg_2, ..., bg_n\}$ according to the target object types, where $bg_i \bigcap bg_j = \varnothing$ and $U = \bigcup_{i=1}^{n} bg_i$



Behavior Embedding Spaces

- Temporal & Behavior Type Encoding (用户的行为转换为嵌入向量)

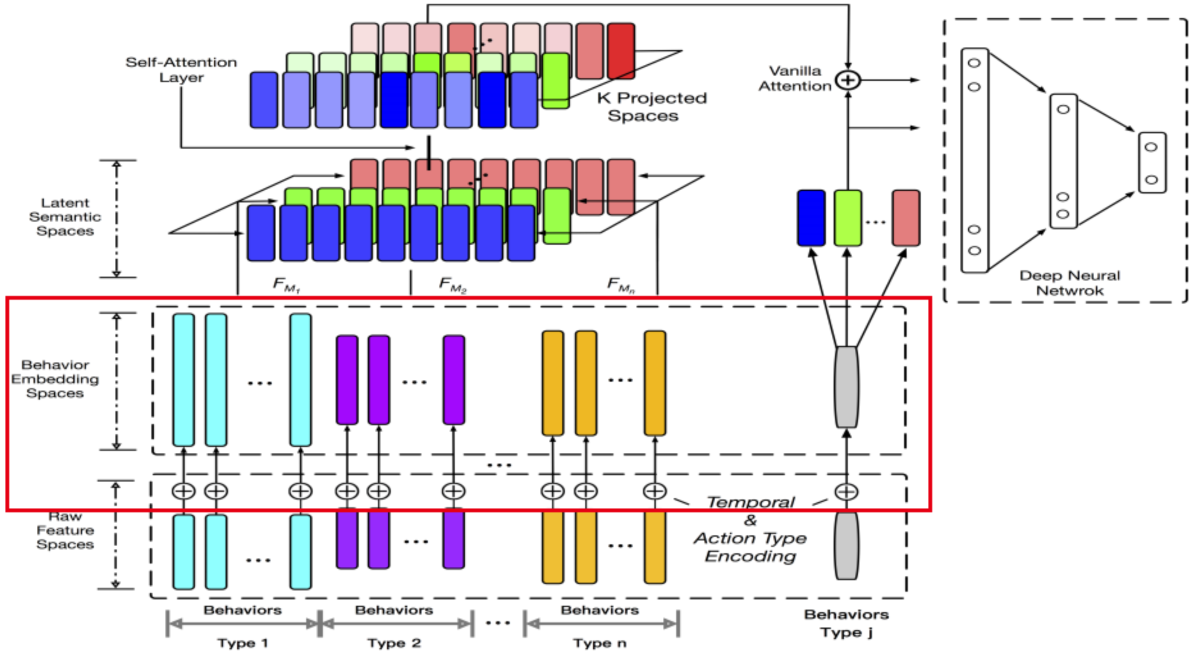
$$\boxed{u_{ij} = emb_i(o_j) + lookup_i^t(bucketize_i(t_j)) + lookup_i^a(a_j)}$$

  - A user behavior: $U = \{(a_j, o_j, t_j) | j = 1, 2, ..., m\}$
  - Behavior groups: $G = \{bg_1, bg_2, ..., bg_n\}$ $\quad bg_i \bigcap bg_j = \emptyset$ and $U = \bigcup_{i=1}^{n} bg_i$
  - All behavior groups: $B = \{u_{bg_1}, u_{bg_2}, ..., u_{bg_n}\}$ $\quad u_{bg_i} = concat_j(u_{ij})\ u_{ij} \in bg_i$

- Embedding Sharing (E.g., 店铺 id，类目 id)
  - 减少一定的稀疏性
  - 降低参数总量

Behavior Embedding Spaces

- The shape of the embeddings for each group may be different
  - The numbers of behaviors
  - The information in each type of behavior
    - E.g., an item behavior vs. a keyword search behavior

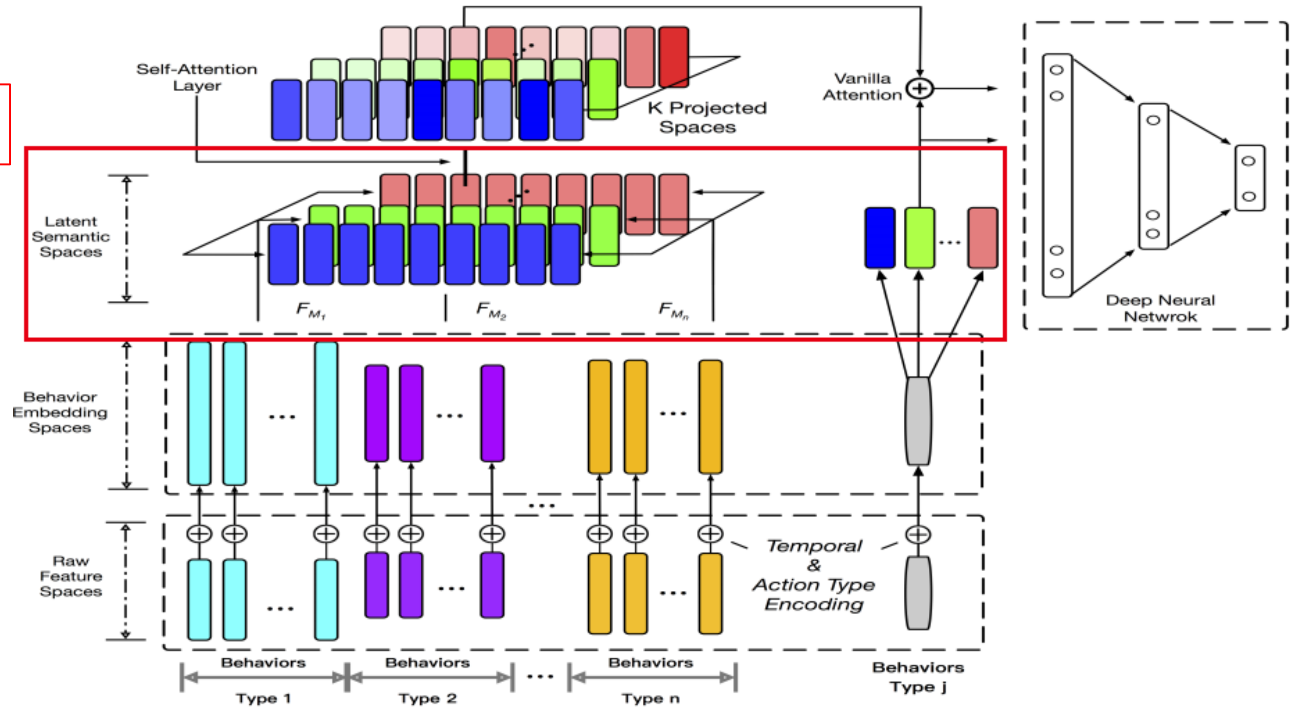


Latent Semantic Spaces

- Map each behavior into K latent semantic spaces
  - Heterogeneous behaviors embedding spaces: different sizes and meanings
- Project the variable-length behaviors in different groups into fix-length encoding vectors

$$S = concat^{(0)}(\mathcal{F}_{M_1}(u_{bg_1}), \mathcal{F}_{M_2}(u_{bg_2}), ..., \mathcal{F}_{M_n}(u_{bg_n}))$$

$$S_k = \mathcal{F}_{P_k}(S)$$

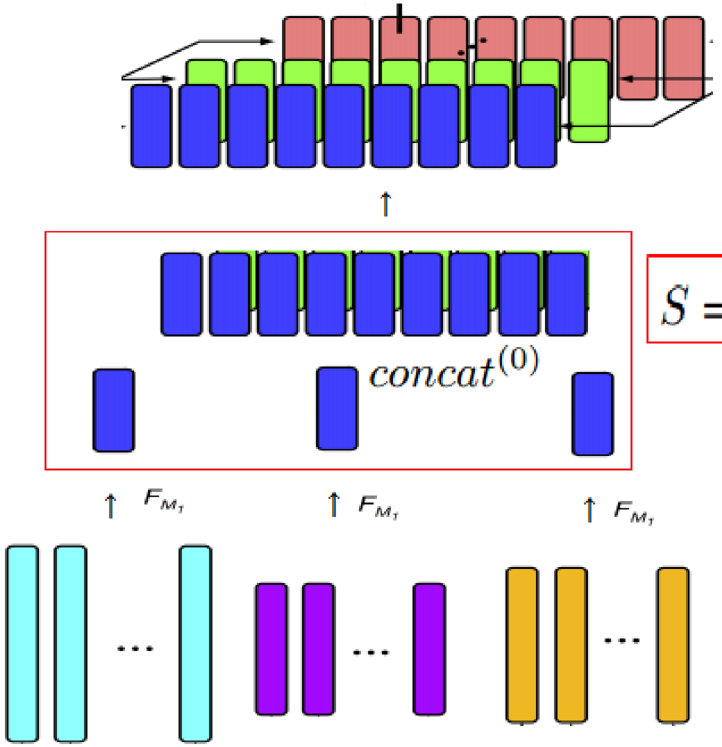


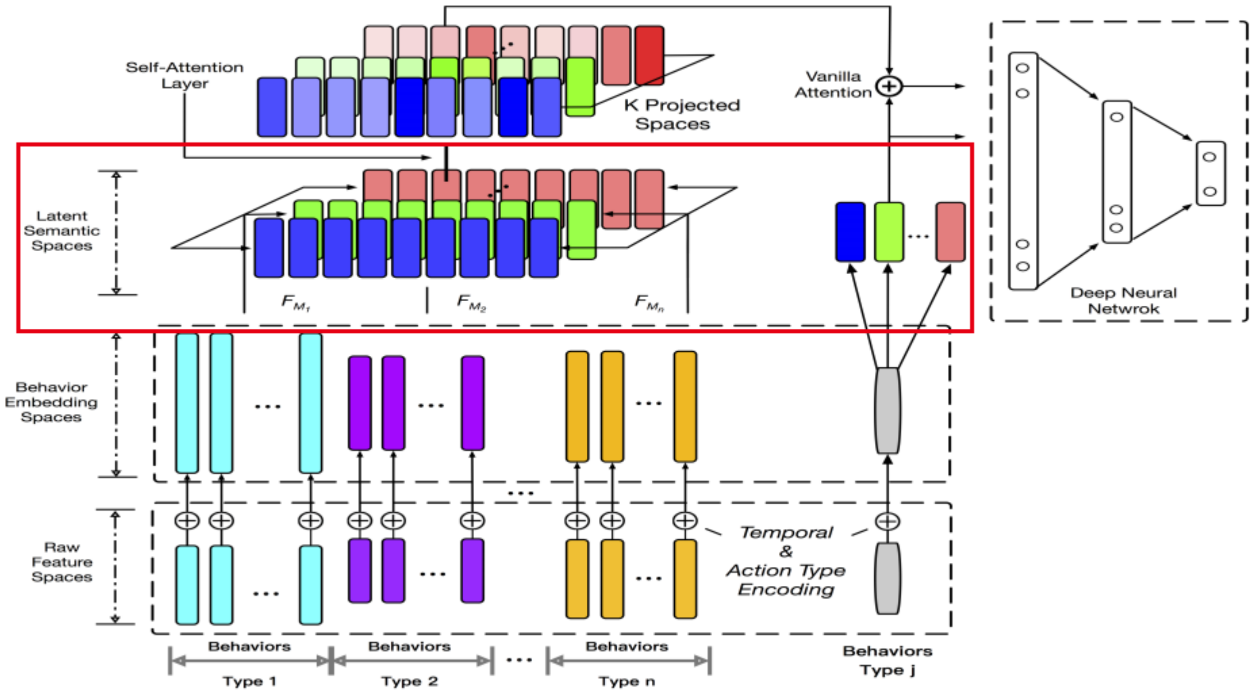
Latent Semantic Spaces

$\mathcal{F}_{P_k}$ is a projection function for the k-th semantic space

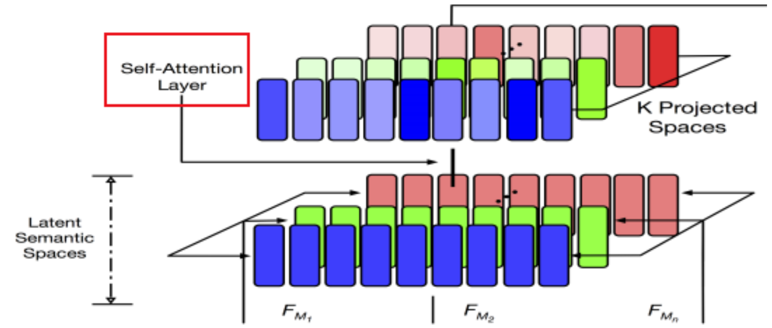
$$\boxed{S_k = \mathcal{F}_{P_k}(S)}$$

$$\boxed{S = concat^{(0)}(\mathcal{F}_{M_1}(u_{bg_1}), \mathcal{F}_{M_2}(u_{bg_2}), ..., \mathcal{F}_{M_n}(u_{bg_n}))}$$





Self-Attention Layer



- Capture the inner-relationships among each semantic space
- Self-attention mechanism
  - Calculate each attention score matrix A_k in the k-th semantic space as:

$$\boxed{\begin{array}{l} A_k = softmax(a(S_k, S; \theta_k)) \\ a(S_k, S; \theta_k) = S_k W_k S^T \end{array} \rightarrow \quad A_k = softmax(\ S_k W_k S^T\ )}$$

$$\text{Attention}(Q, K, V) = \boxed{\text{softmax}(\frac{QK^T}{\sqrt{d_k}})}V$$

  - The attention vectors of space k are:
    - $\mathcal{F}_{Q_k}$ is another projection function
    - $\mathscr{F}_{self}$ is a feedforward network with one hidden layer

$$\boxed{C_k = A_k \mathcal{F}_{Q_k}(S)}$$

$$\boxed{C = \mathscr{F}_{self}(concat^{(1)}(C_1, C_2, ..., C_K))}$$

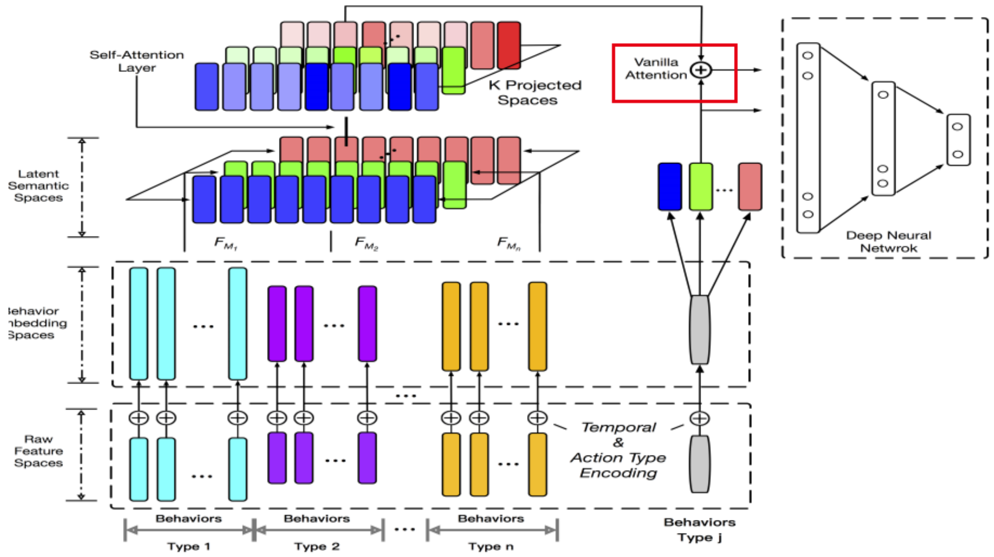
Downstream Application Network

- Vanilla Attention
  - 框架图中灰色的 bar 代表待预测的任意种类的行为
  - 将该行为也通过 embedding、projection 等转换，然后和用户表征产出的行为向量做 vanilla attention
  - 最后 Attention 向量和目标向量将被送入一个 Ranking Network

$$\vec{h_t} = \mathcal{F}_{M_{g(t)}}(\vec{q_t}), \quad \vec{s_k} = \mathcal{F}_{P_k}(\vec{h_t})$$

$$\vec{c_k} = softmax(a(\vec{s_k}, C; \theta_k))\mathcal{F}_{Q_k}(C)$$

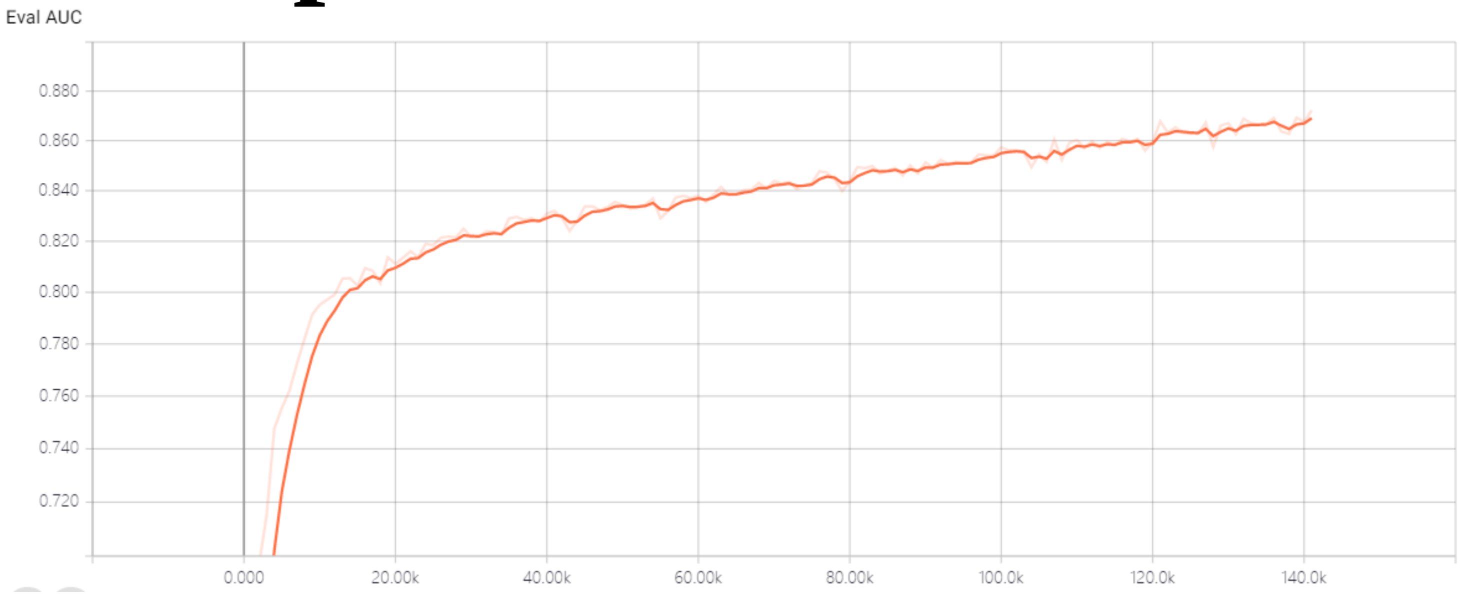
$$\vec{e_u^t} = \mathscr{F}_{vanilla}(concat^{(1)}((\vec{c_1}, \vec{c_2}, ..., \vec{c_K})))$$

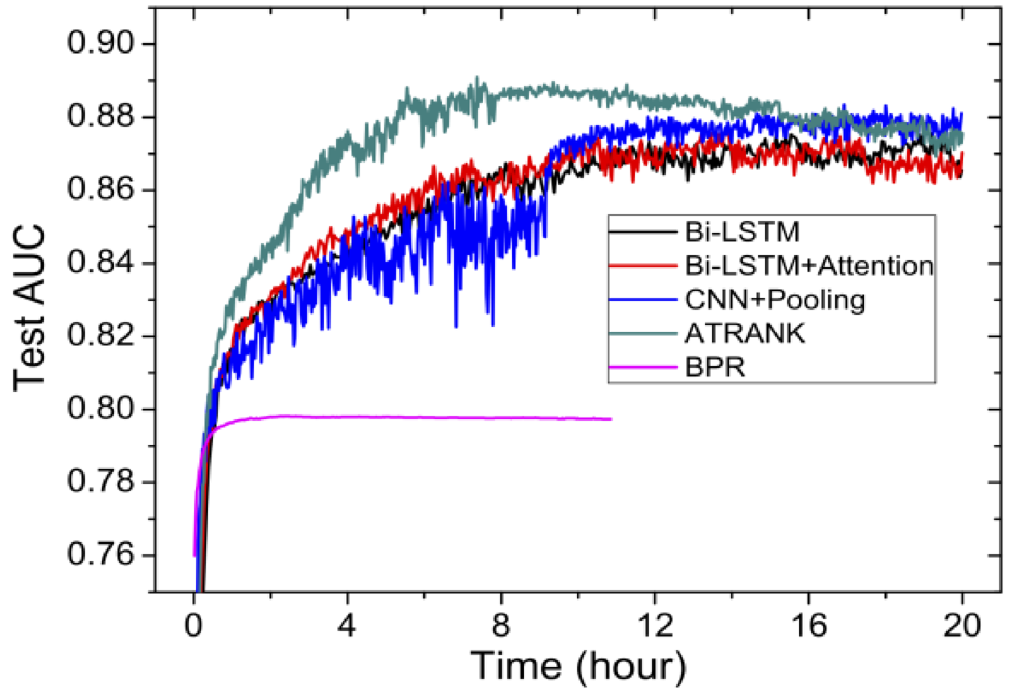


$$-\sum_{t,u} y_t \log \sigma(f(h_t, e_u^t)) + (1 - y_t) \log(1 - \sigma(f(h_t, e_u^t)))$$

Experiment

Dataset	# Users	# Items	# Cates	# Samples
Electro.	192,403	63,001	801	1,689,188
Clothing.	39,387	23,033	484	278,677






```
Epoch 1 Global_step 133000  Train_loss: 0.4485  Eval_AUC: 0.8669
Epoch 1 Global_step 134000  Train_loss: 0.4479  Eval_AUC: 0.8664
Epoch 1 Global_step 135000  Train_loss: 0.4469  Eval_AUC: 0.8667
Epoch 1 Global_step 136000  Train_loss: 0.4429  Eval_AUC: 0.8691
Epoch 1 Global_step 137000  Train_loss: 0.4395  Eval_AUC: 0.8637
Epoch 1 Global_step 138000  Train_loss: 0.4445  Eval_AUC: 0.8627
Epoch 1 Global_step 139000  Train_loss: 0.4433  Eval_AUC: 0.8693
Epoch 1 Global_step 140000  Train_loss: 0.4455  Eval_AUC: 0.8676
Epoch 1 Global_step 141000  Train_loss: 0.4418  Eval_AUC: 0.8722
Epoch 1 Global_step 142000  Train_loss: 0.4458  Eval_AUC: 0.8696
```

Dataset	Electro.	Clothe.
BPR	0.7982	0.7061
Bi-LSTM	0.8757	0.7869
Bi-LSTM + Attention	0.8769	0.7835
CNN + Max Pooling	0.8804	0.7786
ATRank	**0.8921**	**0.7905**

Table 3: AUC on Amazon Dataset