CSIT 6000M(L1) Reproduction and Improvement of xDeepFM

GROUP 13

ZHANG, Hexiao 20932780

ZHENG, Ke 20882179

CONTENT

- Background
- Previous Work
- Network Architecture
- Experiments
- Summary

BACKGROUND

- The explosive growth of online platforms, including e-commerce, social media, and online advertising, has made personalized recommendations ubiquitous in our daily lives
- 2 stages
  a. Retrieval system: find relevant items in a database
  b. Ranking system: rank the items based on predicted scores
- We focus on the score-predicting model of ranking system
  a. Predict the score $P(y \mid \boldsymbol{x})$: the probability of a user action label y given the features $\boldsymbol{x}$
  b. Click- through rate (CTR) prediction

BACKGROUND

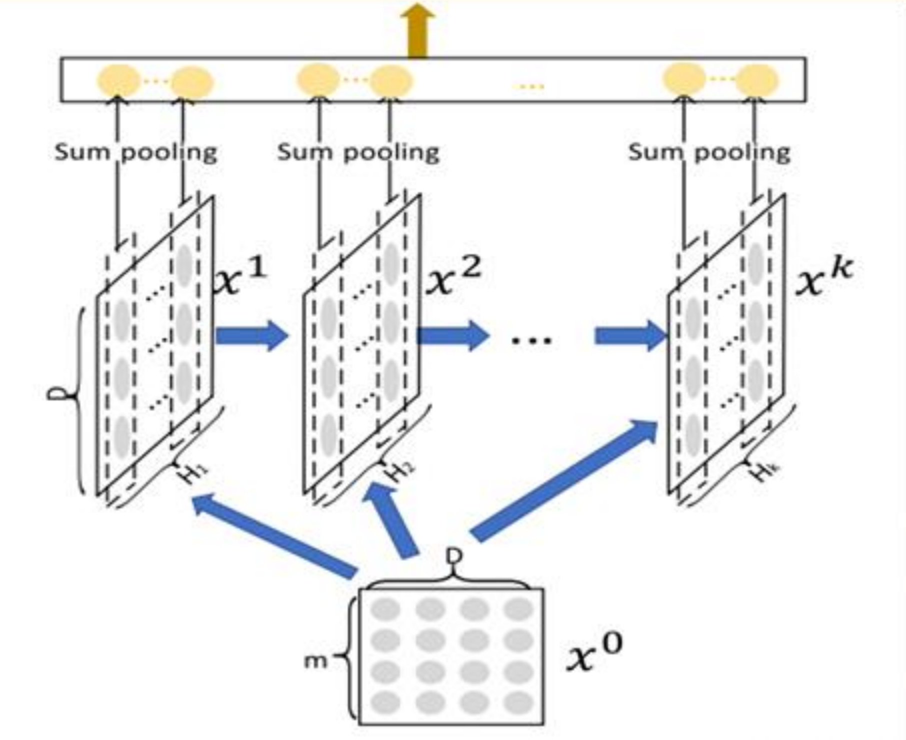
customer_id	zip_code	store_id	age	gender	previous_sales	target_sales
1215425	84342	425	34	0	4322.15	3119.64
8861344	86179	218	52	1	18.75	0.00
7886511	24306	101	42	1	341.17	440.91
...	...	...	...	...	...	...
4452336	84094	218	46	0	1312.66	1430.36
7788899	75011	604	47	0	19.97	177.05
1234687	80011	123	23	1	426.18	398.61

- Features are often highly sparse
- More complex combinations are needed
- Key problem: Feature Interactions

PREVIOUS WORK

Model	Learning fashion	Interaction level	Interaction degree	Characteristics
Manual Engineering				-High cost -Difficult in large-scale systems -Not generalize to unseen interactions
Factorization Machine (FM)	Explicit	Vector-wise	Low-order	-Include useless interactions
Deep Neural Networks (DNNS)	Implicit	Bit-wise	High-order	-Learn arbitrary function -Implicit -At the bit-wise level
Hybrid model (e.g., Wide & Deep, DeepFM)	Implicit	Bit-wise	Low-order & High-order	-Learn arbitrary function -Implicit -At the bit-wise level
Deep & Cross Network (DCN)	Explicit	Bit-wise	High-order	-Bit-wise level -Output is limited in a special form

CIN ARCHITECTURE

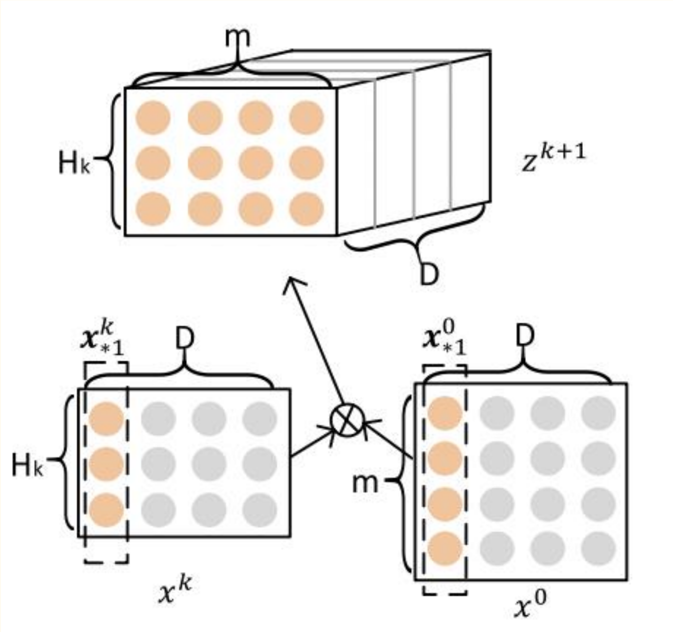


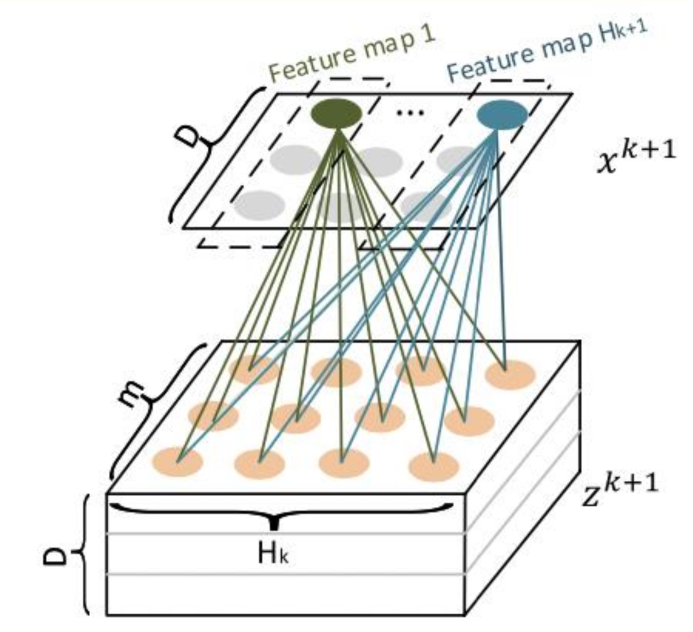
$$\mathbf{X}^k_{h,*} = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^{m} \mathbf{W}^{k,h}_{ij} (\mathbf{X}^{k-1}_{i,*} \circ \mathbf{X}^0_{j,*})$$

$$\mathbf{X}^0 \in \mathbb{R}^{m \times D}$$

CIN ARCHITECTURE

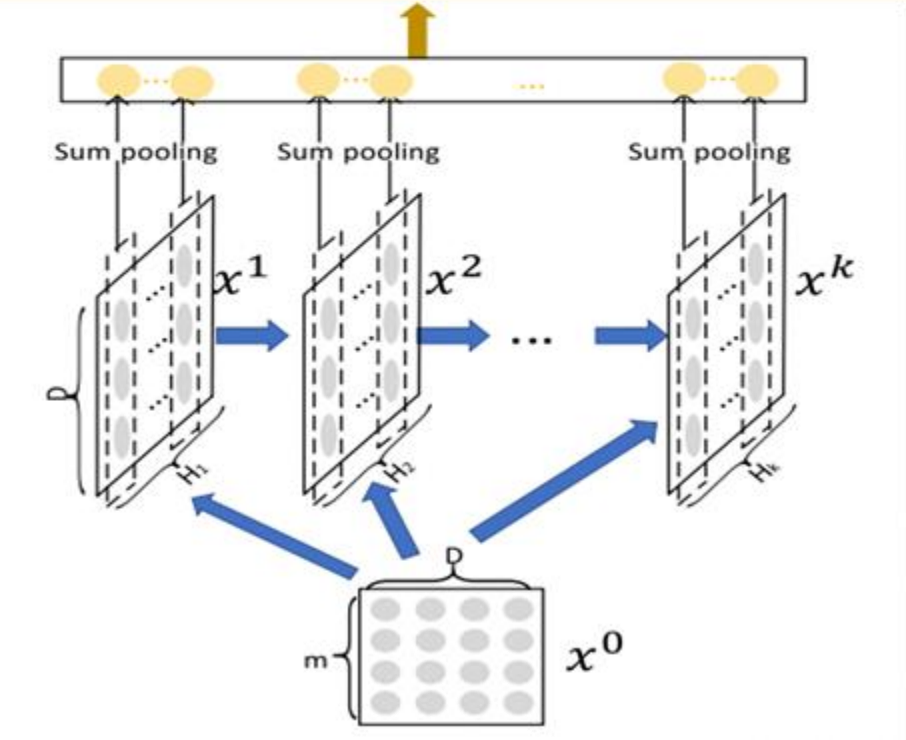
$$\mathbf{X}^{k}_{h,*} = \sum_{i=1}^{H_{k-1}} \sum_{j=1}^{m} \mathbf{W}^{k,h}_{ij} (\mathbf{X}^{k-1}_{i,*} \circ \mathbf{X}^{0}_{j,*})$$





$$\mathbf{Z}^{k+1} = \mathbf{X}^{k} \times \mathbf{X}^{0}$$

CIN ARCHITECTURE



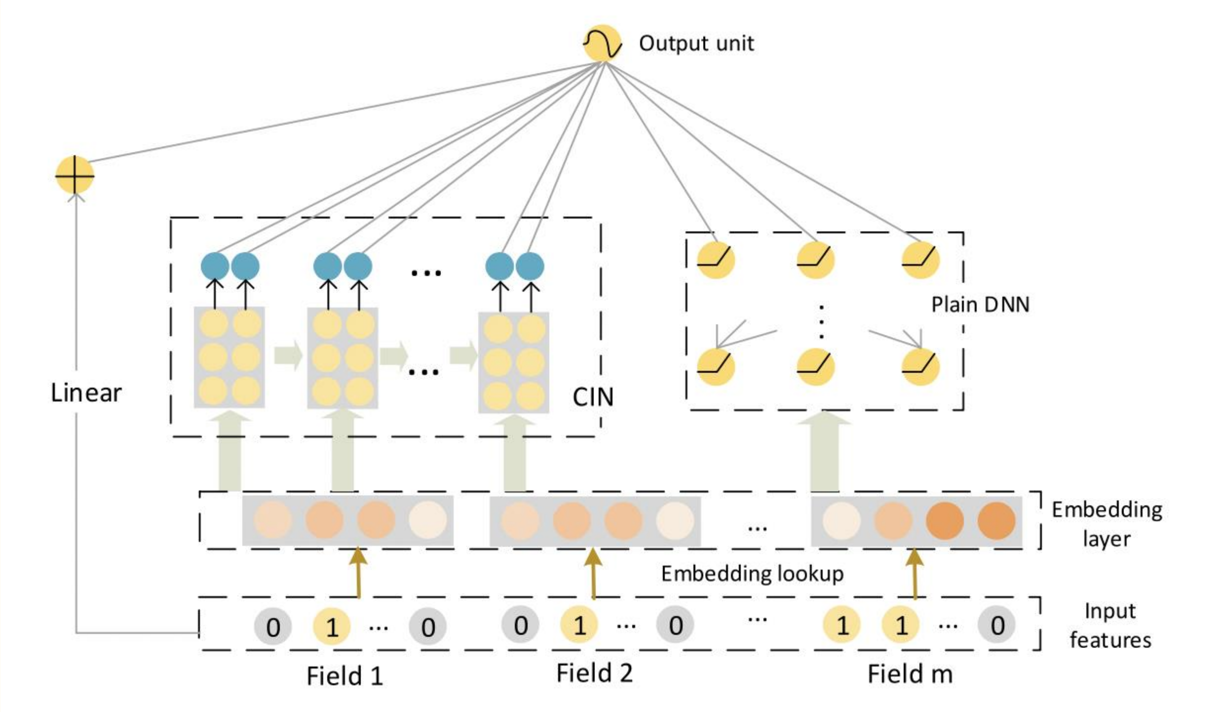
$$p_i^k = \sum_{j=1}^{D} \mathbf{X}_{i,j}^k$$

$$\mathbf{p}^k = [p_1^k, p_2^k, \ldots, p_{H_k}^k]$$

$$\mathbf{p}^+ = [\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^T] \in \mathbb{R}^{\sum_{i=1}^{T} H_i}$$

$$y = \frac{1}{1 + exp(\mathbf{p}^{+^{\mathrm{T}}} \boldsymbol{w}^o)}$$

xDeepFM ARCHITECTURE



- Embedding layer:

raw features → feature embeddings

- Linear part:

raw features → output
Low-order feature interactions

- DNN part:

feature embeddings → output
Implicit high-order, bit-wise level

- CIN part:

feature embeddings → output
Explicit high-order, vector-wise level

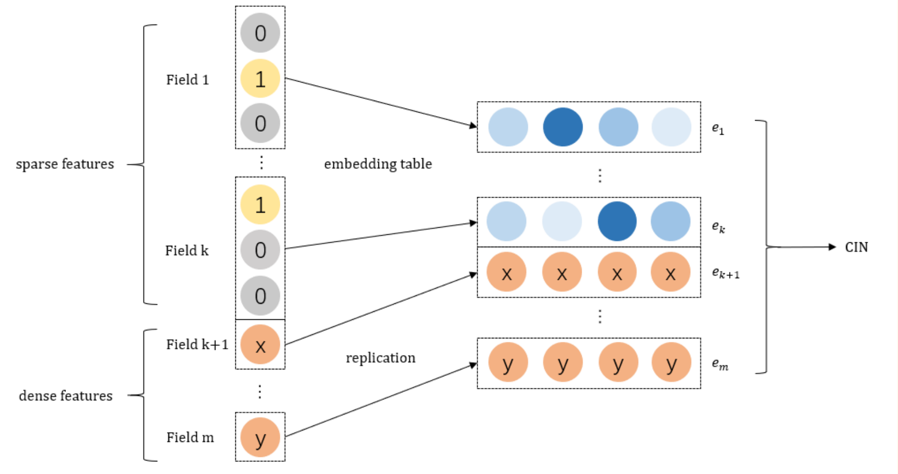
IMPROVEMENT

Insights

- Original CIN can only handle category (sparse) features.
- The interactions related to dense features are modeled by other components of xDeepFM.

Our Proposed Improvement

- Extend dense features into vectors by replication.



EXPERIMENTS

Criteo Dataset

- Classic dataset for predicting CTR
- Each sample contains its label, 26 sparse features and 13 dense features.
- Used 40% of training set due to limitation of computing resources.

  Training set: Val set: Test set = 64%: 16%: 20%

  Each epoch may take more than 1 hour.

Hyper Parameters

- Followed the settings of the original xDeepFM paper.

EXPERIMENTS

Evaluation

- AUC (Area Under the ROC curve)

  Measure the probability that a positive instance will be ranked higher than a random negative one.

- Logloss (Cross Entropy)

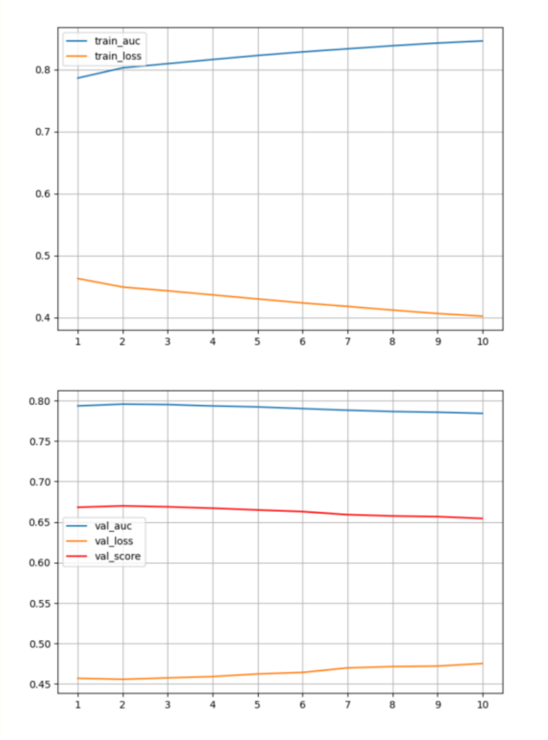
  Measure the distance between the predicted score and the true label.

EXPERIMENTS

Reproduction and Ablation Studies

- Slightly lower performance compared to original paper due to the division of the dataset.
- Validate the effectiveness of combining CIN with DNN in xDeepFM.

Model	AUC	Logloss (Cross Entropy)
Linear + CIN	0.7862	0.4635
Linear + DNN	0.7955	0.4561
xDeepFM	0.7959	0.4555



EXPERIMENTS

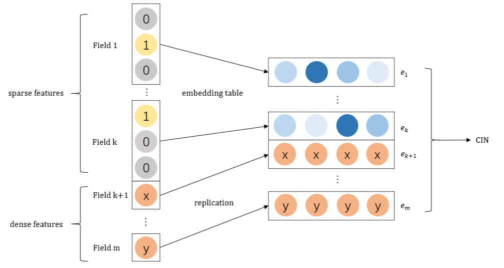
Improved Models

- The performance improved after adding dense embedding to CIN
- Since CIN can model the relationships associated with dense features now, the role of DNN is no longer significant.

Model	Adding Dense Features or Not	AUC	Logloss (Cross Entropy)
Linear + CIN	✗	0.7862	0.4635
	✓	0.7974	0.4538
xDeepFM (Linear + DNN + CIN)	✗	0.7959	0.4555
	✓	0.7974	0.4547

SUMMARY

- Reproduced xDeepFM on part of the Criteo dataset
- Proposed a simple method to vectorize dense features, enabling CIN to learn a comprehensive interactions for both sparse and dense features.



- Left for future work:

  More comprehensive experiments and methods