DSCI 441 Final Project

Feature Interaction in Recommendation Systems

Cancan Zhang
Department of Engineering
Lehigh University
caz322@lehigh.edu

Date: May 1st, 2023

Abstract

Click-through rate (CTR) prediction is a critical task in recommendation systems and online advertising. Feature interaction is a key consideration for improving the accuracy of CTR prediction models. In this study, I present a benchmark to evaluate the performance of six state-of-the-art methods, including Factorization Machines (FM), Product-based Neural Network (PNN), Deep Factorization Machines (DeepFM), Deep Cross Network (DCN), Deep Cross Network v2 (DCNv2), and Attention Factorization Machines (AFM), on two widely used CTR prediction datasets: Criteo and Avazu. I also investigate the effect of embedding size on the best-performing model, PNN. Through my analysis of the experimental results, provide insights into the effectiveness of these methods in recommendation systems and advertising. The code can be found at https://github.com/CancanZhang/Recommendation_Systems.

1 Introduction

Click-through rate (CTR) prediction is a crucial task in recommendation systems and advertising, as it helps to identify which items or advertisements are most likely to be clicked by users. To improve the accuracy of CTR prediction models, it is essential to consider feature interaction, as it enables the modeling of complex relationships between different features. In this study, I present a benchmark for evaluating the performance of several state-of-the-art feature interaction methods, including Factorization Machines (FM) [9], Product-based Neural Network (PNN) [8], DeepFM [5], Deep Cross Network (DCN) [10], Deep Cross Network v2 (DCNv2) [11], and Attention Factorization Machines (AFM) [12], on two widely-used CTR prediction datasets: Criteo [4] and Avazu [2]. These datasets are complex and challenging, containing a large number of features and examples. By analyzing the performance of the different methods on these datasets, I aim to provide insights into

the effectiveness of these methods. Although my study is based on a small subset of the entire dataset, it still provides valuable insights into the direction of further studies.

2 Motivation and Contribution

- **Motivation**: Currently, there are numerous works in the field of recommendation systems that focus on feature interactions. However, these works have been tested on different datasets, some of which are even private. Therefore, in this project, I aim to create a benchmark that compares these algorithms using the same dataset.
- **Contribution**: I built on the code from the OptInter paper [7] and incorporated three additional models: DCN, DCNv2, and Attention FM (AFM). In addition to examining the impact of various feature interaction methods, I also analyzed the influence of embedding dimensions.
- **Future work**: Due to limited computing resources, I was only able to use 10% of the dataset for this benchmark. In the future, the full dataset can be evaluated. Furthermore, additional datasets and models could be included in this benchmark.

3 Models

3.1 Factorization Machines (FM)

FM [9] embeds features into dense features and use inner product to capture feature interactions as following:

$$y_{FM} = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d} \sum_{j=i+1}^{d} v_i^T v_j x_i x_j.$$

3.2 Product-based Neural Network (PNN)

Compared with FM, PNN [8] concate features interaction results and features. In this project, inner product was adopted as feature interactions.

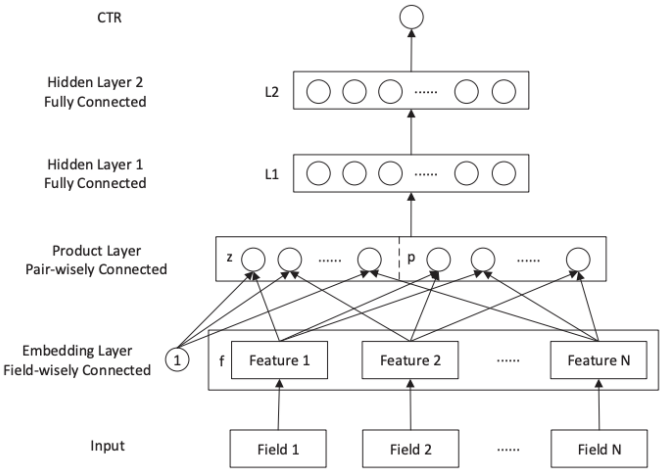


Figure 1: PNN Architecture

3.3 Deep Factorization Machines (DeepFM)

DeepFM [5] uses FM layer to capture low-order feature interactions while dnn layer to capture higher-order feature interactions as Figure 2 shows.

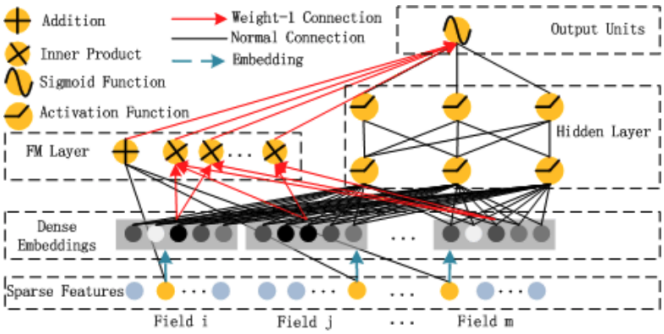


Figure 2: DeepFM Architecture

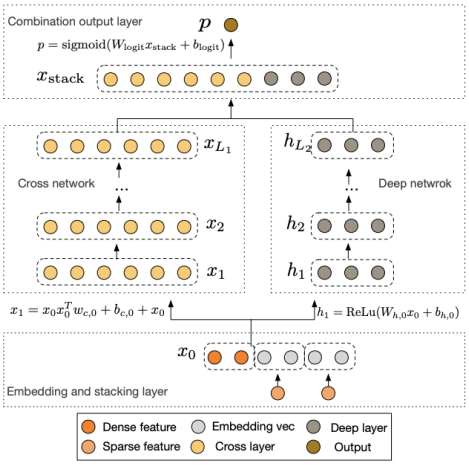
3.4 Deep Cross Network (DCN)

Similar to DeepFM, DCN [10] also have two parallel networks: a cross network and a deep network. The architecture of DCN is shown in Figure 3(a). Instead of using FM network to capture low-order feature interactions, DCN uses cross network to depict high-degree and explicit feature interactions.

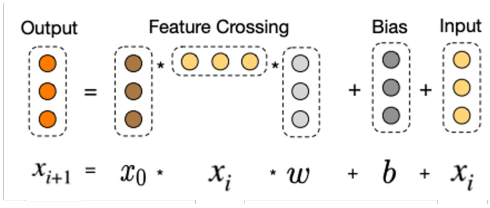
The layer l is computed as:

$$x_{l+1} = x_0 x_l^T w_l + b_l + x_l$$

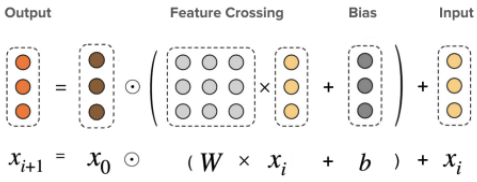
So the cross network comprises all the cross terms from 1 degree to $l+1$ degree.



(a) The architure of DCN



(b) The cross network of DCN



(c) The cross network of DCNv2

Figure 3: The architecture of DCN and DCNv2

3.5 Deep Cross Network v2 (DCNv2)

DCNv2 [11] updates the cross network using:

$$x_{l+1} = x_0 \odot (W x_l + b) + x_l,$$

and Figure 3(b) and 3(c) compares the cross network of DCN and DCNv2. As we can see, DCNv2 applies a matrix to calculate the weight of feature interactions while the DCN uses a vector weight.

3.6 Attention Factorization Machine (AFM)

AFM [12] uses an attention network to learn the attention for different feature interactions:

$$y_{AFM} = w_0 + \sum_{i=1}^{d} w_i x_i + p^T \sum_{i=1}^{d} \sum_{j=i+1}^{d} a_{ij} (v_i \odot v_j) x_i x_j$$

where

$$a_{ij}^{'} = h^T ReLU(W(v_i \odot v_j)x_i x_j + b),$$

$$a_{ij} = \frac{exp(a_{ij}^{'})}{\sum_{(i,j)} exp(a_{ij}^{'})}.$$

Instead of directly using inner product as the importance of feature interactions in FM, AFM applies the element-wise product between two feature latent vectors and uses an attention net to learn each element importance and Figure 4 shows its architecture.

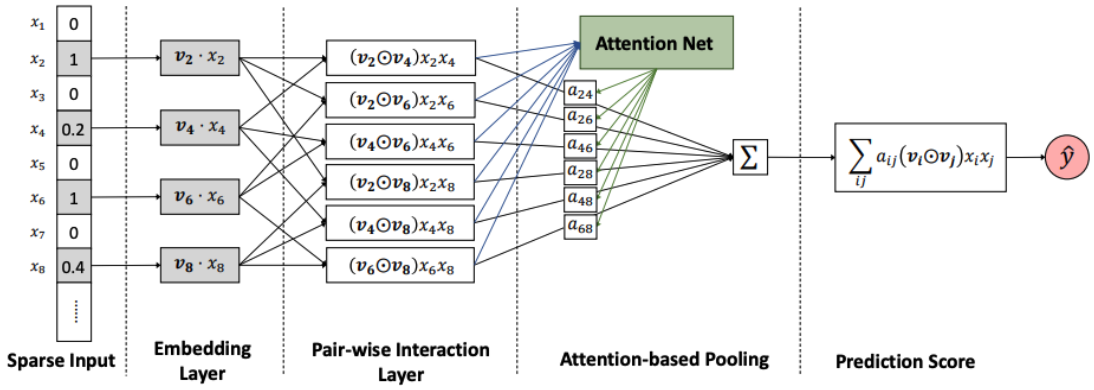


Figure 4: AFM Architecture

4 Datasets

I utilized two benchmark datasets for CTR prediction in advertising: Criteo [4] and Avazu [2]. The Criteo dataset comprises 7 days of Criteo traffic and includes 13 integer features and 26 categorical features, although the feature meanings are not disclosed. The Avazu dataset, on the other hand, consists of 23 categorical features, including 9 anonymized features. An example of Criteo dataset and Avazu dataset can be found in Table 1 and 2.

Table 1: **Criteo Dataset Example**. Criteo dataset [4] is provided by Criteo company for an online advertising click-through rate (CTR) prediction competition. 7 days data is provided in the training data. Generally speaking, it aims at predicting the probability that a user will click on an ad given this user and the page he or she is visiting. The dataset contains 13 columns of integer features (I1 to I13), and 26 columns of categorical features (C1 to C26). But the meaning of these columns are all anonymized and the categorical values themselves are also been hashed onto 32 bits to protect user privacy.

I1	**I2**	**I3**	**I4**	**I5**	**I6**	**I7**	**I8**	**I9**	**I10**	**I11**	**I12**	**I13**
1	6	3	3	443	6	2	6	6	1	2	NaN	3
1	1	25	11	934	72	1	24	25	1	1	NaN	11
1	3	NaN	0	28	6	1	0	0	1	1	NaN	0
2	5	59	2	67	2	2	36	2	1	1	NaN	2
0	144	NaN	2	4294	58	1	2	28	0	1	NaN	2
C1	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**	**C8**	**C9**	**C10**	**C11**	**C12**	**C13**
05db9164	38a947a1	87e2c1f1	b81eecac	25c83c98	7e0ccccf	ea32d016	56563555	a73ee510	935a36f0	755e4a50	a7a17c21	5978055e
8cf07265	4f25e98b	4156136e	932cdb5e	25c83c98	7e0ccccf	64917feb	37e4aa92	a73ee510	3b08e48b	f045731b	eec69081	252ee845
68fd1e64	52e9ecfc	88dca2c7	fd5bf4bf	25c83c98	fbad5c96	c9f171f9	0b153874	a73ee510	5df036eb	755e4a50	01d01675	5978055e
05db9164	2.87e+5	7e5fff77	8c3ca4dd	4cf72387	fe6b92e5	053b27f9	985e3fcb	a73ee510	22d5b25a	a3d2f3d0	66da8e94	e5186205
05db9164	e3a0dc66	6406abee	da13cca9	25c83c98	fbad5c96	ade953a9	1f89b562	a73ee510	6e607294	29e4ad33	ad1492b0	80467802
C14	**C15**	**C16**	**C17**	**C18**	**C19**	**C20**	**C21**	**C22**	**C23**	**C24**	**C25**	**C26**
b28479f6	46ed0b3c	7813d53d	07c540c4	2c6cb693	NaN	NaN	8efe33dc	NaN	32c7478e	b258af68	NaN	NaN
1adce6ef	17d9b759	2eb3e176	d4bb7bd8	7ef5affa	442e81c4	5840adea	052a8e16	NaN	32c7478e	3fdb382b	001f3601	d14e41ff
b28479f6	e2dd9a77	843fbae6	d4bb7bd8	1e42ba17	21ddcdc9	a458ea53	de07786a	ad3062eb	32c7478e	0ccc9397	f0f449dd	f17fe973
64c94865	f8526149	601d6f98	d4bb7bd8	8.92e+12	5e5ff12b	a458ea53	9b6ed758	ad3062eb	423fab69	3fdb382b	ea9a246c	49d68486
b28479f6	35679327	354a37d9	07c540c4	b608c073	NaN	NaN	7b877178	NaN	3a171ecb	f2e9f0dd	NaN	NaN

Table 2: **Avazu Dataset Example**. Avazu dataset [2] is provided by Avazu company for another online advertising click-through rate (CTR) prediction competition. 10 days data is provided in the training data. There are 23 columns and all the columns are categorical features: column id is an advertisement identifier, column click is either 0 or 1 meaning non-click for click, and column C1, and C14 to C21 are anonymized categorical variable for protecting user privacy.

id	click	hour	C1	banner_pos	site_id	site_domain	site_category
1000009418151094273	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd
10000169349117863715	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd
10000371904215119486	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd
10000640724480838376	0	14102100	1005	0	1fbe01fe	f3845767	28905ebd
10000679056417042096	0	14102100	1005	1	fe8cc448	9166c161	0569f928
app_id	app_domain	app_category	device_id	device_ip	device_model	device_type	device_conn_type
ecad2386	7801e8d9	07d7df22	a99f214a	ddd2926e	44956a24	1	2
ecad2386	7801e8d9	07d7df22	a99f214a	96809ac8	711ee120	1	0
ecad2386	7801e8d9	07d7df22	a99f214a	b3cf8def	8a4875bd	1	0
ecad2386	7801e8d9	07d7df22	a99f214a	e8275b8f	6332421a	1	0
ecad2386	7801e8d9	07d7df22	a99f214a	9644d0bf	779d90c2	1	0
C14	C15	C16	C17	C18	C19	C20	C21
15706	320	50	1722	0	35	-1	79
15704	320	50	1722	0	35	100084	79
15704	320	50	1722	0	35	100084	79
15706	320	50	1722	0	35	100084	79
18993	320	50	2161	0	35	-1	157

5 Experiment Settings

5.1 Dataset

To reduce the dataset size and speed up the training process, I used only 10% of the data, dividing it into an 80% training set and a 20% test set, as shown in Table 3.

Table 3: **Dataset Information.** The original Criteo and Avazu dataset contains about 30 million samples in training set. However, to spead up the training process, I only picked 10% of data from the original dataset. 80% of data is used as training set and another 20% is for test.

Dataset	Training Set Size	Test Set Size	#Continues Feaures	#Categorical Features	#Uniq Categorical Values
Criteo	2783531	695882	13	26	514798
Avazu	3234317	808579	0	23	1228795

5.2 Models

The details of network archtecure settings for different models is described in Tabel 4. Layer normalization [3] is applied after the forward neural network. The embedding dimension for Criteo dataset and Avazu dataset is set as 20 and 40 respectively. More detailed experiments about embedding size is explained in Section 6.2. For example, the DCN structure on Criteo and Avazu dataset is shown in Figure 5. Adam [6] optimizer is adopted for all the models. The learning rate for all the models are 0.001 except for AFM model which is slow to converge.

Table 4: **Network Settings.** For FM model, there is no dnn layer employed; For IPNN, DeepFM, DCN and DCNv2 models, the same network structure is used for the same dataset.

Model	Avazu	Criteo
FM	None	None
IPNN	2 hidden layers with size: [500,500]	2 hidden layers with size: [700,700]
DeepFM	2 hidden layers with size: [500,500]	2 hidden layers with size: [700,700]
DCN	2 hidden layers with size: [500,500]	2 hidden layers with size: [700,700]
DCNv2	2 hidden layers with size: [500,500]	2 hidden layers with size: [700,700]
AFM	attention layer size: 8	attention layer size: 8

6 Experiment Results

6.1 Model Performance

AUC metrics is employed to evaluate the model performance. It represents the possibility that the positive samples have higher scores than the negative samples [1].

$$AUC = \frac{\sum_{x_n \in D_n} \sum_{x_p \in D_p} 1[f(x_n) < f(x_p)]}{|D_n||D_p|},$$

where D_n and D_p represents the set of negative samples and positive samples respectively. It also represents the area under ROC curve of (True Positive Rate, False Postive Rate) pairs created by adjusting classification threshold.

The results of the experiments are shown in Table 5. As shown, the IPNN model achieves the best performance on both datasets. Although DCNv2 performs similarly to IPNN, it is a more complex model. Among FM, DeepFM, and AFM, the latter two perform better. Compared to DCN and DCNv2, DCNv2 is the better-performing model.

```
DCN(
  (cate_embeddings_table): Embedding(514798, 20)
  (fc_layers): ModuleList(
    (0): Linear(in_features=780, out_features=700, bias=True)
    (1-4): 4 x Linear(in_features=700, out_features=700, bias=True)
  )
  (norm_layers): ModuleList(
    (0-4): 5 x LayerNorm((700,), eps=1e-05, elementwise_affine=True)
  )
  (dnn_linear): Linear(in_features=1480, out_features=1, bias=True)
)
```

(a) Criteo Dataset

```
DCN(
  (cate_embeddings_table): Embedding(1228795, 40)
  (fc_layers): ModuleList(
    (0): Linear(in_features=960, out_features=500, bias=True)
    (1-4): 4 x Linear(in_features=500, out_features=500, bias=True)
  )
  (norm_layers): ModuleList(
    (0-4): 5 x LayerNorm((500,), eps=1e-05, elementwise_affine=True)
  )
  (dnn_linear): Linear(in_features=1460, out_features=1, bias=True)
)
```

(b) Avazu Dataset

Figure 5: The DCN structure on Criteo and Avazu Dataset

Table 5: Test Set AUC

Dataset	FM	IPNN	DeepFM	DCN	DCNv2	AFM
Criteo	0.7676	**0.7860**	0.7852	0.7854	0.7858	0.7744
Avazu	0.7564	**0.7573**	0.7566	0.7562	**0.7573**	0.7560

6.2 Effect of Embedding Size

In order to further understanding the effect of embedding size, I add experiment changing the embedding dimensions with: 5, 10, 20, 40 and 80. Figure 6 shows the experiment results on these two datasets. As the figure indicates, increasing the embedding size initially leads to an improvement in model performance. However, after a certain point, increasing the embedding size starts to have a negative effect on performance. In the case of the IPNN algorithm, the best performance on the Avazu dataset was achieved with an embedding size of 40, while on the Criteo dataset, an embedding size of 20 was optimal. This is likely due to the fact that Avazu has a higher number of unique categorical values (1228795) than Criteo (514798), and thus requires a larger embedding space.

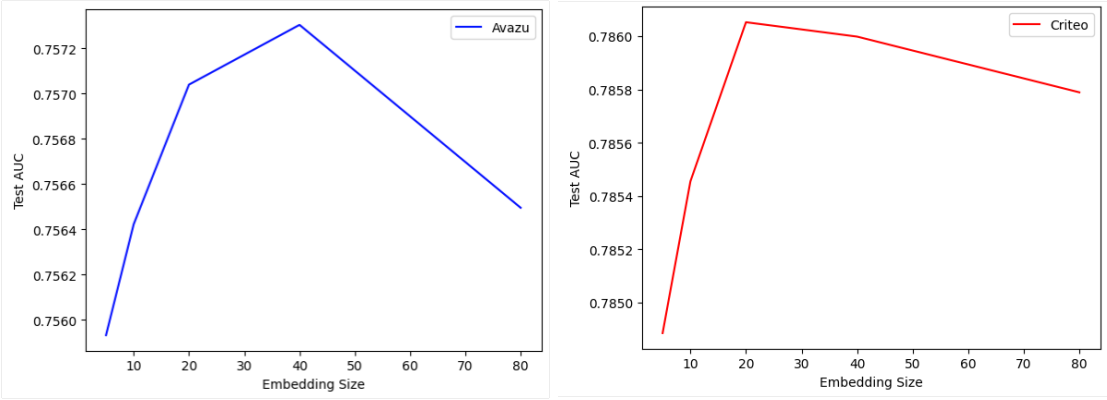


Figure 6: Effect of Embedding Size

7 Conclusion

In this study, I evaluated the performance of six state-of-the-art methods for click-through rate (CTR) prediction, including Factorization Machines (FM), Product-based Neural Network (PNN), DeepFM, Deep Cross Network (DCN), Deep Cross Network v2 (DCNv2), and Attention Factorization Machines (AFM), on two widely used CTR prediction datasets: Criteo and Avazu. I also investigated the effect of embedding size on the performance of the PNN algorithm. My results indicate that the simple IPNN algorithm achieves the best performance among all the methods evaluated in this study. Although more complex networks could perform better with larger number

of dataset, this experiment indicate that it is worthwhile to start the experiment of feature interactions from IPNN model. Moreover, for categories with higher number of unique values, it is better to adopt a larger embedding size.

References

[1] Receiver operating characteristic — Wikipedia, the free encyclopedia. [Online; accessed 06-May-2023].

[2] The Avazu Dataset. `https://www.kaggle.com/c/avazu-ctr-prediction/`. Accessed: 2023-04-16.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[4] Diemert Eustache, Meynet Julien, Pierre Galland, and Damien Lefortier. Attribution modeling increases efficiency of bidding in display advertising. In *Proceedings of the AdKDD and TargetAd Workshop, KDD, Halifax, NS, Canada, August, 14, 2017*, page To appear. ACM, 2017.

[5] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*, 2017.

[6] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[7] Fuyuan Lyu, Xing Tang, Huifeng Guo, Ruiming Tang, Xiuqiang He, Rui Zhang, and Xue Liu. Memorize, factorize, or be naive: Learning optimal feature interaction methods for ctr prediction. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 1450–1462. IEEE, 2022.

[8] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 1149–1154. IEEE, 2016.

[9] Steffen Rendle. Factorization machines. In *2010 IEEE International conference on data mining*, pages 995–1000. IEEE, 2010.

[10] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. 2017.

[11] Ruoxi Wang, Rakesh Shivanna, Derek Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed Chi. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pages 1785–1797, 2021.

[12] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617*, 2017.