2021.05.17

DeepFM

DeepFM: A Factorization-Machine based Neural Network for CTR Prediction

Huifeng Guo[*1], **Ruiming Tang**[2], **Yunming Ye**[†1], **Zhenguo Li**[2], **Xiuqiang He**[2]
[1]Shenzhen Graduate School, Harbin Institute of Technology, China
[2]Noah's Ark Research Lab, Huawei, China
[1]huifengguo@yeah.net, yeyunming@hit.edu.cn
[2]{tangruiming, li.zhenguo, hexiuqiang}@huawei.com

26th International Joint Conference on Artificial Intelligence 2017

Abstract

Learning sophisticated feature interactions behind user behaviors is critical in maximizing CTR for recommender systems. Despite great progress, existing methods seem to have a strong bias towards low- or high-order interactions, or require expertise feature engineering. In this paper, we show that it is possible to derive an end-to-end learning model that emphasizes both low- and high-order feature interactions. The proposed model, DeepFM, combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture. Compared to the latest Wide & Deep model

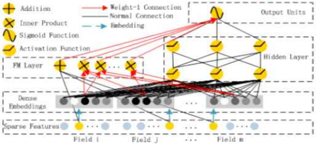


Figure 1: Wide & deep architecture of DeepFM. The wide and deep component share the same input raw feature vector, which enables DeepFM to learn low- and high-order feature interactions simultaneously from the input raw features.

Abstract

Learning sophisticated feature interactions behind user behaviors is critical in maximizing CTR for recommender systems. Despite great progress, existing methods seem to have a strong bias towards low- or high-order interactions, or require expertise feature engineering. In this paper, we show that it is possible to derive an end-to-end learning model that emphasizes both low- and high-order feature interactions. The proposed model, DeepFM, combines the power of factorization machines for recommendation and deep learning for feature learning in a new neural network architecture. Compared to the latest Wide & Deep model from Google, DeepFM has a shared input to its "wide" and "deep" parts, with no need of feature engineering besides raw features. Comprehensive experiments are conducted to demonstrate the effectiveness and efficiency of DeepFM over the existing models for CTR prediction, on both benchmark data and commercial data.

1. DeepFM모델은 Low와 High-order interactions모두 학습할 수 있다.
2. Factorization Machine과 Deep Learning의 장점을 모두 합친 모델이다.
3. 추가 feature engineerin없이 raw feature를 그대로 사용할 수 있다.
4. 벤치마크 데이터와 commercial데이터에서 실험을 완료했다.

1. Introduction

1. CTR(click through rate)이란 user가 추천된 항목을 click할 확률을 예측하는것. -> CTR에 의해 user가 선호할 item랭킹을 부여한다.
2. CTR을 정확히 예측하는데 있어 **가장 중요한 것은 feature interaction을 효과적으로 모델링하는 것**이다.
   - App category와 Timestamp관계 : 음식 배달 어플은 식사시간 근처에 다운로드가 많다.
   - App categort와 User gender, Age관계 : 남자 청소년들은 슈팅과 RPG게임을 선호한다.
   - 맥주와 기저귀를 함께 구매하는 사람들이 많다 -> 쉽게 이해되는 interaction과 달리 Hidden interaction도 있다.

   -> **low와 high-order feature interactio을 모두 고려해야 한다.**

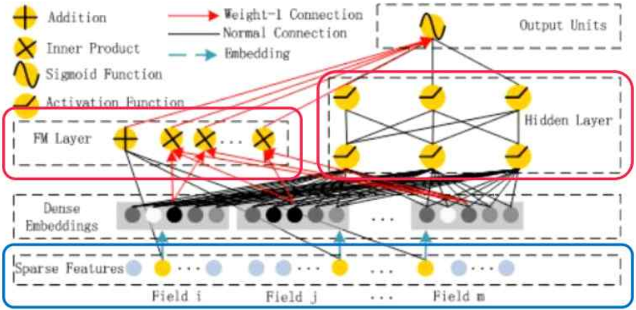
   -> **Explicit과 implicit features를 모두 모델링할 수 있어야 한다.**

1. Introduction

- We propose a new neural network model DeepFM (Figure 1) that integrates the architectures of FM and deep neural networks (DNN). It models low-order feature interactions like FM and models high-order feature interactions like DNN. Unlike the wide & deep model [Cheng *et al.*, 2016], DeepFM can be trained end-to-end without any feature engineering.
- DeepFM can be trained efficiently because its wide part and deep part, unlike [Cheng *et al.*, 2016], share the same input and also the embedding vector. In [Cheng *et al.*, 2016], the input vector can be of huge size as it includes manually designed pairwise feature interactions in the input vector of its wide part, which also greatly increases its complexity.
- We evaluate DeepFM on both benchmark data and commercial data, which shows consistent improvement over existing models for CTR prediction.

1. DeepFM 모델 구조를 제안한다.
   - Low order는 FM part에서, High-order는 DNN part에서 모델링한다.
   - End-to-end 학습 가능 (pretrain이 필요없다.)
2. DeepFM은 다른 비슷한 모델보다 더 효율적으로 학습할 수 있다.
   - Input과 embedding vector를 share한다.
   - Wide부분에서 따로 pairwised feature interaction을 만들어 줄 필요가 없다!
3. DeepFM은 benchmark와 commercial데이터의 CTR predictio에서 의미있는 성능향상을 이루었다.

2.1 DeepFM – architechure



1. DeepFM은 FM component와 deep component 로 구성되고, 같은 input을 공유한다.

Prediction output

$$\hat{y} = sigmoid(y_{FM} + y_{DNN})$$

2. 예측해야 하는 값 y_hat은 predicted CTR 이고 0또는 1값이다.
3. FM layer, Deep layer에서 나온 output을 concat해서 이 값에 sigmoid함수를 적용한 것이 최종 y예측값이 된다.

2.1 DeepFM – architechure

FM Component

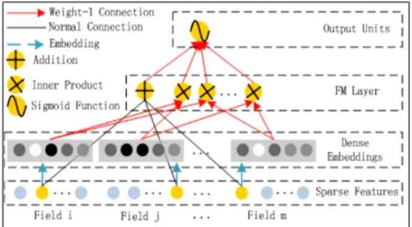


Figure 2: The architecture of FM.

Deep component

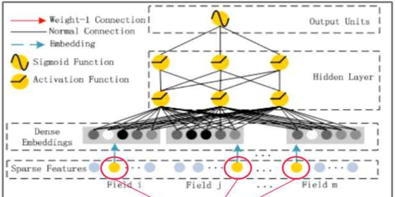


Figure 3: The architecture of DNN.

모든 embedding vector을 모두 합친 것이 input이 된다

$$y_{FM} = \langle w, x \rangle + \sum_{j_1=1}^{d} \sum_{j_2=j_1+1}^{d} \langle V_i, V_j \rangle x_{j_1} \cdot x_{j_2}$$

- High-order interaction을 학습

- FM의 결과는 unit의 addition과 많은 unit의 내적들의 합산이다.
- Addition unit의 <w,x>는 order-1 feature의 중요성을 나타냄.
- **임베딩 벡터의 내적을 이용해 order-2 의 부분도 해결한다는 것이 포인트!**

2.1 DeepFM – architechure

Input Part

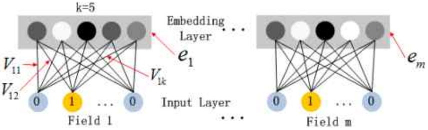


Figure 4: The structure of the embedding layer

$$\boxed{a^{(0)} = [e_1, e_2, \dots, e_m]}$$

$$a^{(l+1)} = \sigma\left(W^{(l)} a^{(l)} + b^{(l)}\right)$$

$$y_{DNN} = \sigma(W^{|H|+1} \cdot a^H + b^{|H|+1})$$

- Input layer를 embedding layer로 변환하는 structure이다.
- 1) input field의 길이는 다 달라도 그들의 embeddin은 같은 사이즈(k)
- 2) V가 pre-train되어야 할 필요를 없애고 공동적으로 한번에 전체적인 network가 학습될 수 있도록 하였다.

- m : field의 개수
- l : layer depth
- 알파 : activation function
- H : hidden layer 개수

2.2 Relationship with the other Neural Networks

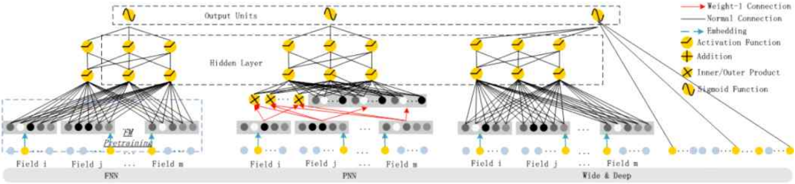


Figure 5: The architectures of existing deep models for CTR prediction: FNN, PNN, Wide & Deep Model

Table 1: Comparison of deep models for CTR prediction

	No Pre-training	High-order Features	Low-order Features	No Feature Engineering
FNN	×	√	×	√
PNN	√	√	×	√
Wide & Deep	√	√	√	×
DeepFM	√	√	√	√

Conclusion

1. DeepFM
   - Deep componen와 FM component를 합쳐서 학습한다.
   - Pre-train이 필요하지 않다.
   - High와 Low-order feature interactions 둘다 모델링한다.
   - Input과 embedding vector를 share한다.
2. Experiment
   - CTR task에서 더 좋은 성능을 얻을 수 있다.
   - 다른 SOTA모델보다 AUC와 Logloss에서 성능이 뛰어나다.
   - DeepFM이 가장 efficient하다.