

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

<https://arxiv.org/abs/1703.04247>

0. Introduction

- 클릭률 예측(CTR prediction)은 온라인 광고와 추천 시스템에서 핵심 과제임.
- 기존 접근 방식은 Factorization Machine(FM)을 사용해 특징 간 상호작용(feature interactions)을 모델링했지만, 비선형 관계를 충분히 포착하지 못함.
- 딥러닝 기반 모델(예: FNN, PNN)은 비선형 표현 학습에 강하지만, 저차 상호작용을 효율적으로 처리하지 못함.
- 본 논문은 이 두 접근법의 장점을 결합한 DeepFM을 제안하여, FM의 자동 특성 조합 학습과 DNN의 고차 표현 학습을 동시에 수행함.

1. Overview

- DeepFM은 두 부분으로 구성됨:
 1. FM component: 저차(2차) feature interactions 학습
 2. Deep component: DNN으로 고차 비선형 feature interactions 학습
- 두 모듈은 동일한 embedding vector를 공유하며, 입력 피처를 임베딩하여 병렬로 처리함.
- 최종 출력은 두 컴포넌트의 출력을 합산한 후 sigmoid를 적용하여 CTR 확률을 계산함.
- 특징:
 - feature engineering 불필요
 - end-to-end 학습 가능
 - 기존 FM, DeepFM, Wide & Deep 간 성능 비교 실험 수행

2. Challenges

- CTR 예측에서의 주요 난제:
 - 희소 고차원(feature sparsity): one-hot encoding된 범주형 데이터로 인해 feature space가 매우 큼
 - feature interactions 선택: 어떤 feature 쌍이나 조합이 의미 있는지 자동으로 학습하기 어려움
 - 복잡한 비선형 관계: 단순 선형 모델(FM, LR)은 높은 수준의 상호작용을 학습하지 못함
 - 특징 엔지니어링 의존도: 기존 모델은 수작업 feature 조합에 크게 의존함
- 따라서 모델은 희소 입력 처리 + 저차 및 고차 상호작용 자동 학습을 동시에 수행해야 함

3. Method

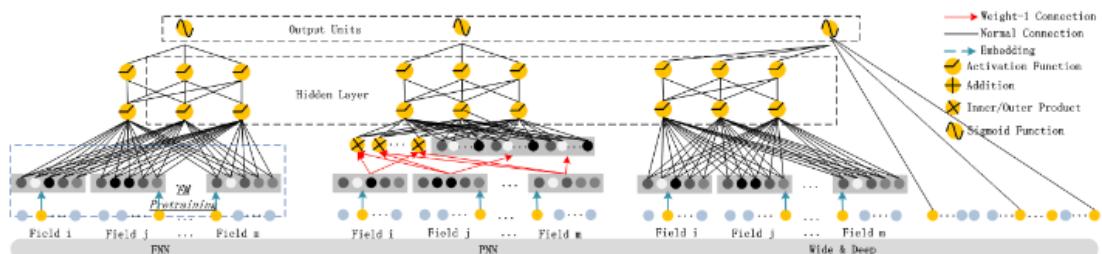


Figure 5: The architectures of existing deep models for CTR prediction: FNN, PNN, Wide & Deep Model

- 입력 표현:
 - 범주형 feature를 one-hot 인코딩 → embedding layer로 밀집 벡터 변환
 - embedding은 FM component와 Deep component가 공유(shared)
- FM component:

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

- Deep component:

- 입력: 모든 feature embedding을 concatenate한 벡터
- 다층 신경망(Multilayer Perceptron, MLP)을 통과
- 비선형 activation(ReLU) 사용
- 출력: (y_{Deep})
- 학습:
 - Binary cross-entropy loss
 - Adam optimizer 사용
 - end-to-end 방식으로 joint training 수행
- 특징: FM 부분은 low-order interaction, DNN 부분은 high-order interaction 학습
→ 자동 feature 조합 가능

4. Experiments

- Dataset:
 - Criteo Ad Click Logs (public benchmark)
 - Company's internal advertising dataset (proprietary, 구조 유사)
- Baseline models:
 - Logistic Regression (LR)
 - Factorization Machine (FM)
 - FNN (FM-based Neural Network)
 - PNN (Product-based Neural Network)
 - Wide & Deep
- Metrics:
 - AUC (Area Under ROC Curve)
 - Log Loss
- Training setup:
 - batch normalization 적용
 - hidden layer: 3~5 layers, each with 400 neurons

- ReLU activation
- dropout 사용 ($p=0.5$)

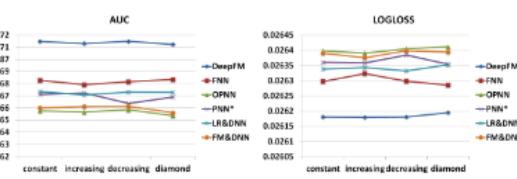
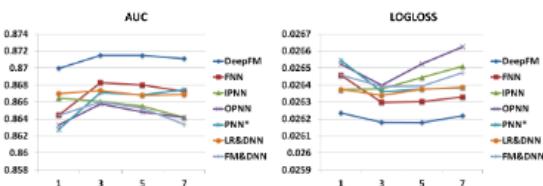
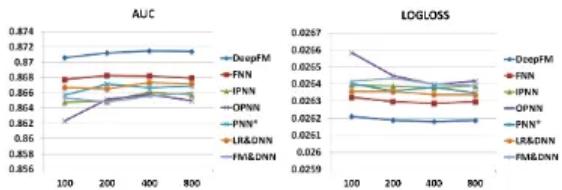
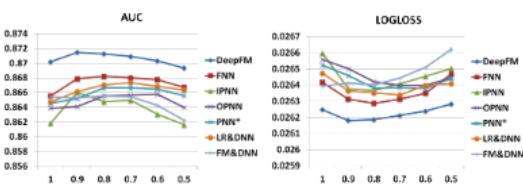
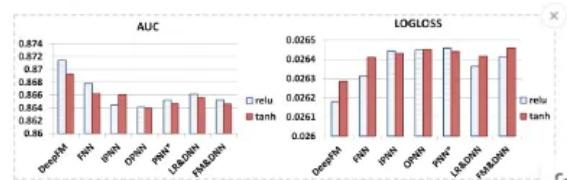
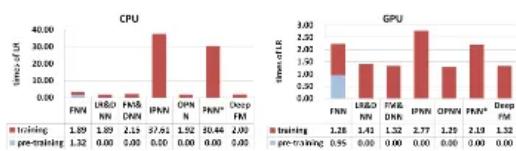
5. Results

Table 1: Comparison of deep models for CTR prediction

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

Table 2: Performance on CTR prediction.

	Company*		Criteo	
	AUC	LogLoss	AUC	LogLoss
LR	0.8640	0.02648	0.7686	0.47762
FM	0.8678	0.02633	0.7892	0.46077
FNN	0.8683	0.02629	0.7963	0.45738
IPNN	0.8664	0.02637	0.7972	0.45323
OPNN	0.8658	0.02641	0.7982	0.45256
PNN*	0.8672	0.02636	0.7987	0.45214
LR & DNN	0.8673	0.02634	0.7981	0.46772
FM & DNN	0.8661	0.02640	0.7850	0.45382
DeepFM	0.8715	0.02618	0.8007	0.45083



- 성능:
 - DeepFM은 Criteo dataset에서 AUC 0.8053으로 기존 Wide & Deep (0.7987)보다 향상
 - Logloss 또한 더 낮게 나타남

- Ablation Study:
 - shared embedding이 없을 경우 성능 하락 → embedding 공유가 효율적임을 검증
 - deep component depth를 늘릴수록 고차 표현 학습 효과 상승
- Feature engineering 불필요성:
 - Wide & Deep은 wide part에 handcrafted features 필요
 - DeepFM은 embedding 공유로 자동 feature interaction 학습 가능
- 학습 효율:
 - FM과 DNN을 병렬로 학습하므로 학습 속도도 비교적 빠름
- 결론: DeepFM은 feature engineering 없이도 CTR 예측에서 높은 성능 달성

6. Insight

- DeepFM은 FM의 장점(저차 상호작용)과 DNN의 장점(비선형 고차 표현)을 효율적으로 통합
- shared embedding 구조가 feature representation을 효율화함
- feature engineering 불필요 → 실무 적용 용이
- end-to-end 학습 구조 → 유지보수 및 확장성 높음
- 다양한 추천 및 광고 도메인에 적용 가능
- deep component 구조가 단순 MLP에 한정 → sequence/context 정보 반영 어려움
- embedding dimension과 depth에 따른 성능 변동 존재