

데이터분석프로그래밍 KDD 논문 review

**DeepGBM: A Deep Learning Framework Distilled by GBDT
for Online Prediction Tasks**

Smart I.O.T Lab
201511837 이상민 (학석사연계과정)

목차

What's GBDT?

1. ABSTRACT
2. INTRODUCTION
3. RELATED WORK
4. DEEPGBM
5. EXPERIMENT
6. CONCLUSION

What's GBDT(Gradient Boosting Decision Tree)?

Gradient Boosting은 컴퓨터공학의 알고리즘에서 자주 등장하는 greedy 전략을 decision tree에 적용한 것이라고 볼 수 있다

Boosting은 기존 모델이 잘 맞추지 못한 데이터에 대해 additional model을 이용해서 모델의 표현력을 계속해서 높이는 방법이며 기존 모델이 잘 못 맞춘 데이터에 대한 정보를 전달하는 방법이 error를 미분한 gradient에 기반하면 gradient boost가 된다.

MLP를 예로 들면, GB MLP을 만든다면, 경우 먼저 1개의 히든 뉴런만 있는 3층짜리 MLP로 데이터를 학습한다. 그 다음 error가 더이상 줄어들지 않으면, 새로운 히든 노드를 하나 추가하고 그 새로 추가된 노드의 weight를 같은 데이터로 다시 학습한다. 이를 error가 충분히 줄어들 때까지 계속 반복한다. 이 과정은 결국 하나의 히든 뉴런을 additional model로써 계속 추가하면서 표현력을 점진적으로 증가시키는 과정에 해당한다.

이러한 과정을 MLP가 아니라, Decision tree를 사용하게 되면 Gradient Boosting Tree(GBTree)가 되는 것이다.

1. Abstract

온라인 예측은 많은 실제 애플리케이션에서 가장 필수적인 작업 중 하나

표 입력 공간 : 몇 개의 범주로 나누어진 특징. (시간, 제목,,)

온라인 데이터 생성 : 동적 분포로 이어진 연속 데이터. (댓글작성시간)

표 형식 입력 공간을 이용한 효과적인 학습과

온라인 데이터 생성에 대한 빠른 적응은

온라인 예측 모델을 얻는 데 있어 중요한 과제이다.

1. Abstract

GBDT와 NN이 실제로 널리 사용되지만, 취약한 부분이 있다.

GBDT는 동적 온라인 데이터 생성에 거의 적응할 수 없으며 희박한 범주형 특징에 직면할 경우 비효율적인 경향이 있다.

NN은 밀도가 높은 수치 특징에 직면할 경우 만족스러운 성능을 달성하기가 매우 어렵다.

이러한 문제점들을 보완하기 위해 NN의 구성 요소를 사용하여 NN과 GBDT의 장점을 통합하는 새로운 학습 프레임 워크인 DeepGBM을 제안.

1. Abstract

1. CatNN : 희박한 범주형 특징 처리.
2. GBDT2NN : GBDT로 증류된 데이터를 가진 고밀도 수치 특징.

이러한 두가지의 Component를 사용하여서 DeepGBM은 효율적인 온라인 업데이트 기능을 유지하며, 희박한 범주형 특징 처리, 고밀도 수치 모두를 활용할 수 있다.

공개데이터 학습 결과, DeepGBM은 다양한 온라인 예측 작업에 대한 성능이 기존의 다른 프레임워크의 성능보다 앞서는 것을 볼 수 있다.

2. Introduction.

온라인 예측은 **클릭 예측, 웹 검색의 콘텐츠 순위, 추천 시스템의 콘텐츠 최적화, 이동 시간 추정**과 같은 실제 산업 애플리케이션에서 필수적인 역할을 하는 특정 유형의 작업을 의미.

일반적인 온라인 예측 작업은 표 입력 공간과 온라인 데이터 생성 측면 두가지를 산출하는데, 표 입력 공간은 후원 검색에서 클릭 예측 작업의 특징 공간은 일반적으로 쿼리와 광고 사이의 텍스트 유사성과 같은 숫자 범주 뿐만 아니라 광고 범주와 같은 범주형도 포함한다.

온라인 데이터 생성은 이러한 작업의 실제 데이터가 온라인에서 생성되고 데이터 배포가 실시간 동적이다.

뉴스 추천 시스템은 엄청난 양의 데이터를 실시간으로 생성하고, 끊임없이 등장하는 뉴스는 다른 시간에 동적 기능 배포를 발생시킨다.

2. Introduction.

온라인 예측 과제에 대한 효과적인 학습 기반 모델

1. 표 형식 입력 공간을 사용하여 효과적인 모델을 학습
2. 온라인 데이터 생성에 맞게 모델을 조정

현재 온라인 예측 작업을 해결하는데 널리 사용되는 기계 학습 모델 :
GBDT, NN이 있지만, 이는 두 가지 주요 과제를 동시에 해결 할 수 없다.
각자의 장단점이 존재하기 때문이다.

GBDT의 주요 장점은 고밀도 수치 특징을 효과적으로 처리할 수 있다.
트리를 구축하기 위해 통계 정보가 가장 큰 특징을 반복적으로 선택할 수 있
기 때문에, GBDT는 훈련 목표에 잘 맞도록 유용한 수치 특징을 자동으로 선
택하고 결합한다.

2. Introduction.

GBDT는 클릭 예측, 웹 검색 순위 및 기타 잘 알려진 예측 작업에서 효과를 입증했으나, 두 가지 주요 단점이 있다.

1. 학습된 트리를 변경할 수 없으므로 온라인 모드에서 업데이트하기가 어렵다.

이를 위해서는 재학습을 거쳐야하기때문에 비효율적이다.

이는 대규모 데이터를 학습하는 것을 방해한다.

학습을 위해 메모리에 엄청난 양의 데이터를 불러와야 하기 때문이다.

2. 희소 범주적 특징에 대한 학습의 비효율적이다.

One-hot encodings을 통한 변환 후 불균형 파티션과 비 파티션이 거의 같기 때문에 희소 형상에서 가는 통계 정보 이득이 적기 때문에 희소 기능을 사용하여 트리를 효과적으로 학습시키지 못한다.

2. Introduction.

NN은 희소 범주적 특징을 효과적으로 처리하고 온라인 데이터 생성에 효율적으로 적응할 수 있기 때문에 NN을 사용하여 클릭예측, 추천 시스템을 포함한 온라인 예측 작업에서 NN을 채택하지만, 밀도가 높은 수치 표 형상에 대해 학습하여 효과적인 모델을 도출하기는 여전히 어렵다.

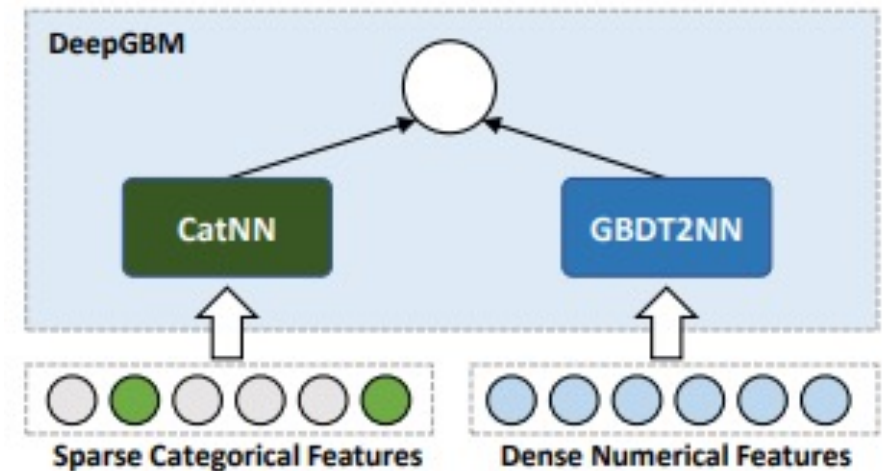
	NN	GBDT	GBDT+NN	DeepGBM
Sparse Categorical Feature	✓	✗	✓	✓
Dense Numerical Feature	✗	✓	✓	✓
Online update & Large-scale data	✓	✗	✗	✓

2. Introduction.

표 입력 공간과 온라인 데이터 생성의 두 가지 주요 과제를 동시에 해결하기 위해 NN과 GBDT의 장점을 통합하는 DeepGBM을 제안하였다.

CatNN은 범주형 형상의 입력이 있는 NN 구조이고 GBDT2NN은 수치 형상의 입력이 있는 또 다른 NN 구조이다.

수치적 특징을 학습하는 GBDT의 강점을 이용하기 위해 GBDT2NN은 GBDT가 학습한 지식을 사용하는 밀도가 높은 수치 형상의 입력이 있는 다른 NN 구조



2. Introduction.

GBDT2NN은 데이터 증류의 효과를 높이기 위해 사전 훈련된 GBDT의 출력 지식을 전달할 뿐만 아니라 획득한 트리의 트리 구조에 의해 암시되는 기능 중요성과 데이터 파티션에 대한 지식을 통합한다.

이러한 방식으로 GBDT와 유사한 성능을 달성하는 한편, NN 구조를 가진 GBDT2NN은 온라인 데이터 생성에 직면할 때 지속적으로 등장하는 데이터를 통해 쉽게 업데이트할 수 있다.

이러한 부분을 통해서 DeepGBM은 표 형식의 데이터를 사용하여 구성되어진 다양한 데이터 세트에 대해 기존의 예측 작업의 모델들을 능가하는 성능을 보인다.

3. Related work.

3.1 Applying GBDT for Online Prediction Tacks.

3.1.1 Online Update in Trees.

GBDT의 온라인 업데이트 문제를 해결하기 위한 관련 연구

스트리밍 데이터에서 트리 기반 모델을 훈련 시키려고 하였지만, 단일 트리 모델 또는 종속성이 없는 다중 병렬 트리를 위한 설계이므로 GBDT에 적용하기 어려우며, 한 번에 모든 데이터를 학습하는 것보다 성능이 떨어진다.

트리 구조를 고정시키고 새로운 데이터로 리프 출력을 업데이트하는 XGBoost, LightGBM은 만족할 만한 성능을 내지 못하며, 추가되는 데이터만으로 분할점을 시도하는 방법은 과거 데이터에 대한 통계 정보를 포함하지 않고 새로운 데이터에 의해 분할되기 때문에 정보의 편향이 발생하여 성능이 불안정하였다.

3. Related work.

3.1 Applying GBDT for Online Prediction Tacks.

3.1.2 Categorical Features in Trees

일부 다른 인코딩 방법을 통해 의사결정 트리가 잘 처리할 수 있도록 범주형 값을 고밀도 숫자 값으로 변환하려고 시도했다.

CatBoost도 범주형 특징에 유사한 숫자 인코딩 솔루션을 사용하였지만, 극도로 희박하고 고차원적인 특징은 불균형 파티션에서 매우 작은 통계 정보 이득을 야기할 수 있기 때문에 GBDT는 희소 특징을 효과적으로 사용하여 트리를 성장시킬 수 없다.

범주형 특성은 가능한 이진 파티션을 열거하여 트리 학습에도 직접 사용될 수 있지만, 이 방법은 범주에 데이터가 너무 적어서 통계 정보가 편향되기 때문에 범주적 특성이 희박한 경우 훈련 데이터에 과적합된다.

한마디로, 온라인 예측 작업에 GBDT를 적용하는 지속적인 노력이 있었지만, 대부분은 온라인 데이터 생성 처리 방법과 범주형 특징에 대한 학습 방법 측면에서 중요한 과제를 효과적으로 해결할 수 없다

3. Related work.

3.2 Applying NN for Online Prediction Tacks.

3.2.1 Prediction Tacks.

온라인 예측 작업에 NN은 밀도가 높은 수치 특징을 효과적으로 학습할 수 없다.

그동안의 NN을 채택한 많은 연구들은 희박한 범주적 특징에 초점을 맞추었기 때문에,
고밀도 수치 특징에 대한 연구가 부족하다.

전통적으로 FCNN은 종종 고밀도 수치 형상에 사용되지만, local optimums에 빠질 가능성이 크기 때문에 대부분 만족스러운 성능에 도달하지 못한다.

3. Related work.

3.3 Combining NN and GBDT

3.3.1. Tree-like NN.

Tree-like NN(예: GoogLeNet)은 어느 정도 나무와 같은 의사 결정 능력을 가지고 있다. 의사 결정 능력을 NN에 도입하는 다른 연구도 있었지만, 이러한 작업은 표 입력 공간이 있는 온라인 예측 작업에 주의를 기울이지 않고 주로 **컴퓨터 비전 작업에 초점**이 맞추어져 있다.

이 부분을 해결하기 위해서 **soft binning function**을 제안하였지만, 가능한 모든 의사 결정을 열거하기 때문에 **매우 비효율적**이다.

표 형식 데이터에서 학습을 개선하기 위해 **NNRF**를 제안하였지만, GBDT와 같은 훈련 데이터에 대한 **통계 정보를 활용하지 않고 무작위의 기능 조합을 사용**한다.

3. Related work.

3.3 Combining NN and GBDT

3.3.2. Convert Trees to NN.

훈련된 의사 결정 트리를 NN으로 변환하려고 시도하였지만, 작업이 중복되고 매우 희박한 NN을 사용하여 단순한 의사 결정 트리를 나타내기 때문에 비효율적이다.

트리가 매우 클 경우에는 변환 솔루션의 NN을 매우 넓게 구성해야하기 때문에 이는 현실적으로 사용이 불가능하다.

또한 하나의 트리를 변환하기 위해서 복잡한 규칙을 사용하므로 실제로 쉽게 사용할 수 없다.

3. Related work.

3.3 Combining NN and GBDT

3.3.3. Combining NN and GBDT.

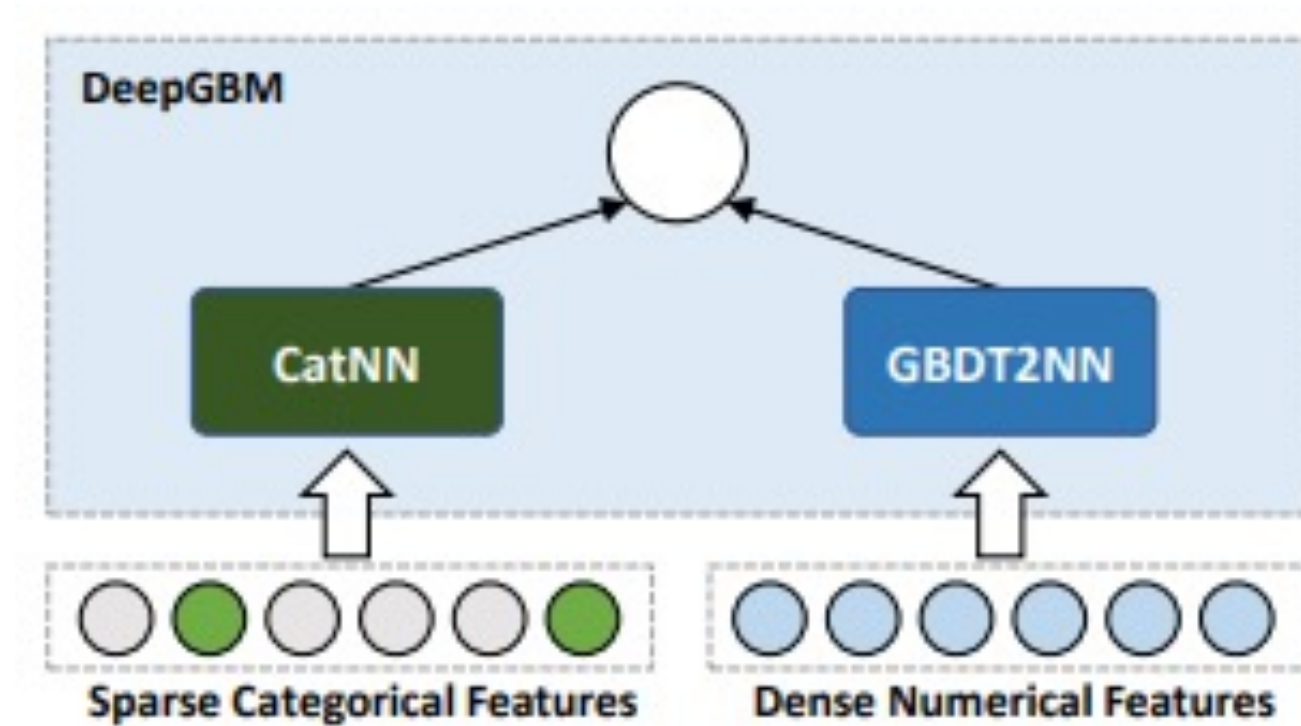
NN과 GBDT를 결합하려는 시도는 여럿 존재하였다.

Facebook, Microsoft등에서 문제 해결을 위해서 사용하였지만, GBDT의 온라인 업데이트 문제가 해결되지 않아 온라인에서 효율적으로 사용할 수 없다.

Facebook에서 GBDT 모델은 좋은 온라인 성능을 얻기 위해서 매일 재교육을 진행해야한다는 문제점을 지적하였다.

이러한 문제점들을 극복하기 위해서 NN과 GBDT를 더 잘 통합하기 위한 DeepGBM을 제안한다.

4. DeepGBM.



CatNN은 범주형 특징을 입력하는 NN 구조이고, GBDT2NN은 고밀도 수치 특징을 학습하는 데 초점을 맞춘 GBDT에서 증류된 또 다른 NN 구조이다.

4. DeepGBM.

4.1 CatNN for Sparse Categorical Features.

NN은 Wide&Deep, PNN, DeepFM 및 xDeepFM과 같은 범주적 기능에 대한 예측 모델을 학습하기 위해 널리 사용되었으며, CatNN의 학습 데이터가 이러한 작업과 동일하기 때문에 기존의 성능이 검증된 NN구조를 활용하여 CatNN에 활용하였다.

CatNN은 고차원 희소 벡터를 고밀도 벡터로 효과적으로 변환할 수 있는 임베딩 기술에 주로 의존한다.

CatNN은 FM, Deep구성 요소를 활용하여 특징에 대한 상호 작용을 학습하기 때문에 유사한 기능을 가진 다른 NN 구성 요소를 사용할 수 있기에 구성 요소에 의해 제한되지 않습니다.

다음과 같은 최종 출력 구조를 갖는다.

$$y_{Cat}(x) = y_{FM}(x) + y_{Deep}(x) . \quad (4)$$

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.1.1 Single Tree Distillation.

GBDT에서 학습된 트리를 NN 모델로 증류하는 방법에 대한 세부 정보 설명이다.

대부분의 이전 증류는 학습된 함수 측면에서 모델 지식만 전송하여 새 모델이 전송된 모델과 유사한 출력을 생성하도록 한다.

그러나 Tree와 NN은 기존 모델 증류가 자연적으로 다르기 때문에 **트리 모델에 더 많은 지식이 증류되어 NN으로 전달될 수 있다.**

특히 학습된 나무 구조에 의해 암시되는 데이터 파티션뿐만 아니라 학습된 나무의 특징 선택과 중요성은 실제로 나무의 다른 유형의 중요한 지식이다.

더 많은 정보가 잘 정제되어 다음 모델로 전송된다.

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.1.2 Tree-Selected Features.

NN과 비교하여 트리 기반 모델의 특별한 특징은 학습이 통계 정보를 기반으로 훈련 목표에 맞는 유용한 특징을 탐욕스럽게 선택하기 때문에 **모든 입력 기능을 사용하지 않을 수 있다**는 것이다.

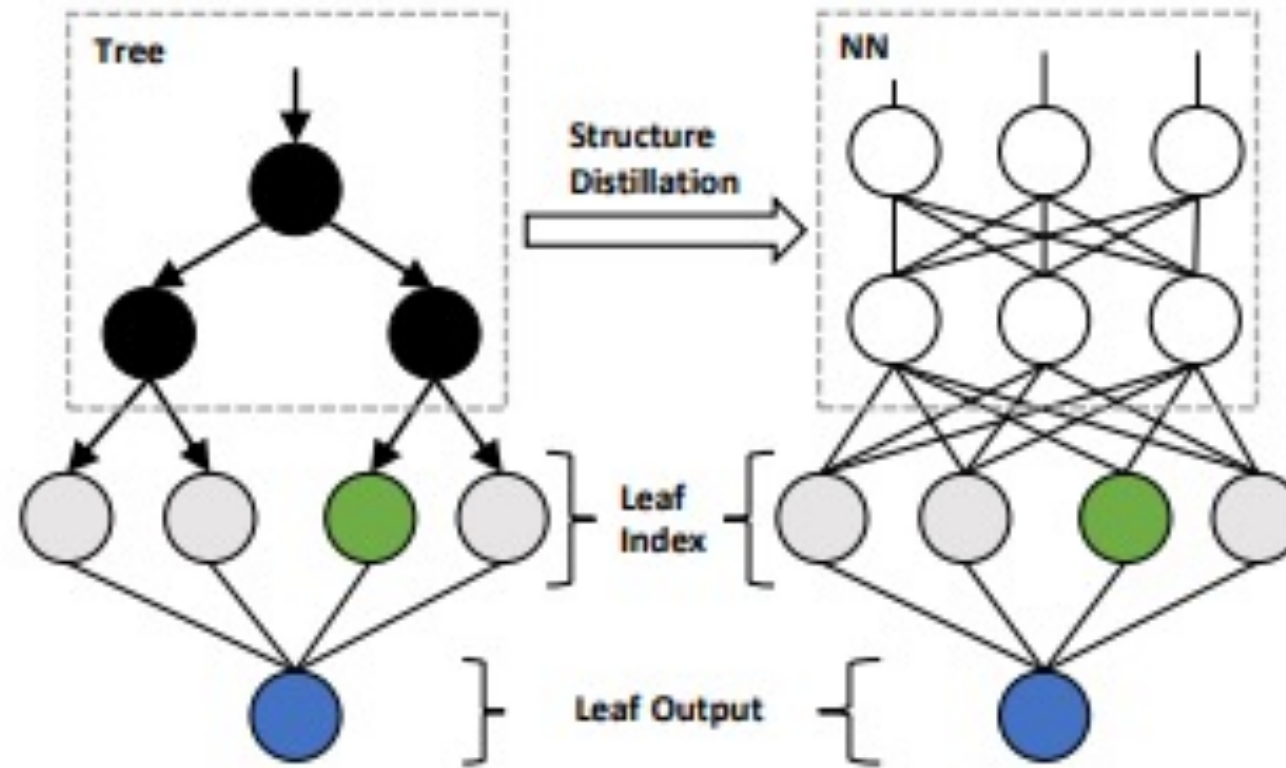
따라서 모든 입력 기능을 사용하는 대신 NN 모델의 **학습 효율성을 개선하기 위해**
트리 선택 기능을 활용할 수 있다.

특히 트리 선택 기능을 NN의 입력으로 사용할 수 있다.

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.1.3 Tree Structure.



4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.1.3 Tree Structure.

기본적으로 의사 결정 트리의 트리 구조에 대한 지식은 데이터를 중복되지 않는 많은 영역으로 분할하는 방법을 나타냅니다.

이러한 트리 구조를 NN으로 직접 전송하는 것은 본질적으로 다르기 때문에 어렵다.

NN 모델을 사용하여 트리 구조의 기능을 근사화하고 구조 지식 증류를 달성하였고,
그림 2에서 설명한 것처럼 NN을 사용하여 트리에서 생성된 클러스터 결과를 적합시켜 NN이 의사 결정 트리의 구조 기능을 근사하게 만들 수 있다.

NN의 강력한 표현 능력 때문에, 학습된 NN 모델은 의사결정 트리의 구조 기능과 완벽하게 유사해야 한다.

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

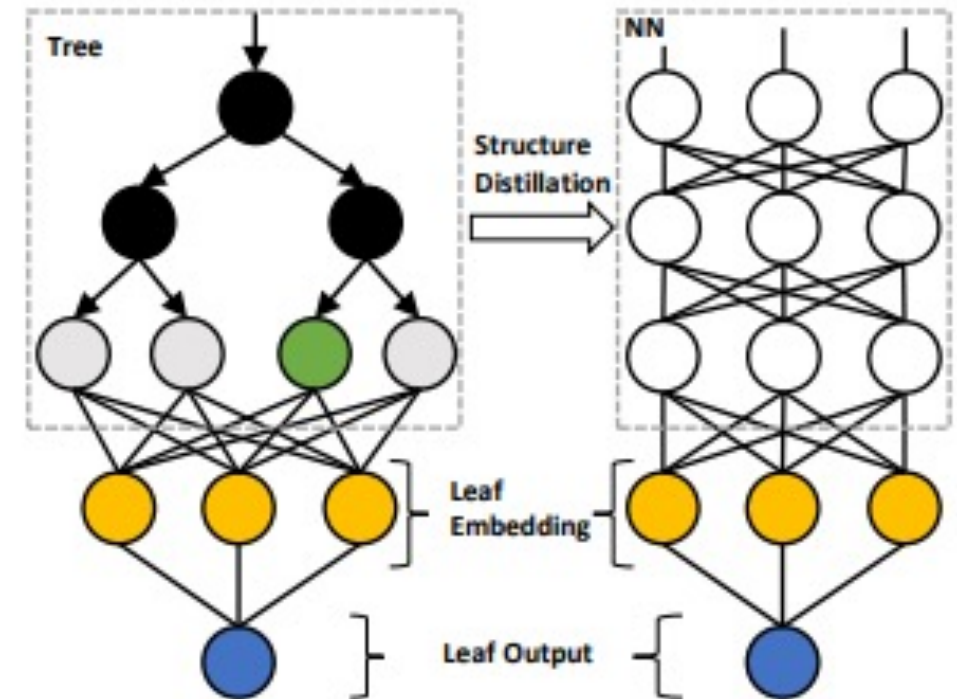
4.2.1.4 Tree Outputs.

Tree 입력에서 Tree 구조로의 매핑은 이전 단계에서 학습되었으므로 트리 출력을 증류하려면 Tree 구조에서 Tree 출력으로의 매핑만 알면 된다.

리프 인덱스에 해당하는 리프 값이 있으므로 이 매핑은 실제로 학습에 필요하지 않습니다.

Leaf를 내장하여 Tree 구조를 증류합니다. Leaf index는 먼저 Leaf Embedding으로 변환됩니다.

그런 다음 NN은 Leaf Embedding에 적합하여 트리 구조를 근사화한다.



4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.2.1 Multiple Tree Distillation.

GBDT에는 여러 트리가 있으므로 여러 트리의 증류액을 일반화해야 합니다.
각 모델은 하나의 트리를 사용한다.

그러나 이러한 구조는 높은 차원 때문에 비효율적이기 때문에
L을 줄여 효율성을 개선하기 위해서 Leaf embedding과 tree group을 제안
한다.

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.2.2 Leaf Embedding Distillation.

이전 단계의 데이터의 정확도를 높이고 L'' 의 차수를 줄이기 위해서 임베딩 기술을 채택하였으며, 임베딩이 하스 가저의 다음과 같다

$$\min_{w, w_0, \omega^t} \frac{1}{n} \sum_{i=1}^n \mathcal{L}^n \left(w^T \mathcal{H}(L^{t,i}; \omega^t) + w_0, p^{t,i} \right), \quad (7)$$

이 후 희박한 고차원 원 핫 표현 대신, 고밀도 임베딩을 대상으로 트리 구조의 기능을 근사화할 수 있으

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L} \left(\mathcal{N} \left(\mathbf{x}^i [\mathbf{I}^t]; \theta \right), H^{t,i} \right), \quad (8)$$

L 은 조밀한 임베딩 장착을 위한 L'' 손실과 같은 회귀 손실이다.

이는 훨씬 적은 NN 매개변수를 사용하기 때문에 더 효율적이다.

4. DeepGBM.

4.2 GBDT2NN for Dense Numerical Features.

4.2.2.3 Tree Grouping.

NN을 줄이기 위해 트리를 그룹화하고, NN 모델을 사용하여 트리 그룹에서 증류한다.

하지만 2가지의 문제점이 발생한다.

1. Tree를 그룹화하는 방법 2. Tree Group에서 증류하는 방법

Tree를 그룹화하는 방법의 해결책들 중 **균등하게 그룹화하는 방법**을 사용한다.

Tree Group에서 증류하기 위해서 **여러 트리에 대한 Leaf Embedding 증류를 확장**한다. Leaf Embedding 증류된 여러 결과 multiple one-hot leaf index vectors를 하나의 완전 연결 네트워크로 만든다.

결과적으로, Leaf 내장 증류 및 트리 그룹화로 인해 **GBDT2NN은 GBDT의 많은 트리를 소형 NN 모델로 효율적으로 증류**할 수 있으며, 트리 출력 외에도 **트리의 특징 선택과 구조 지식도 NN 모델로 효과적으로 증류**된다.

4. DeepGBM.

4.3 Training for DeepGBM.

4.3.1.1 End-to-End Offline Training.

DeepGBM의 학습을 위해서 먼저 **오프라인 데이터를 사용하여 GBDT 모델을 학습**한 후 GBDT의 Tree에 대한 Leaf embedding을 받아야 한다.

그래야지 DeepGBM을 처음부터 끝까지 교육할 수 있다.

실험에 사용된 데이터셋

#Num 숫자 형상의 수

#Cat 범주 형상의 수

#Sample : 데이터 세트의
세부 정보

Name	#Sample	#Num	#Cat	Task
Flight	7.79M	5	7	Classification
Criteo	45.8M	13	26	Classification
Malware	8.92M	12	69	Classification
AutoML-1	4.69M	51	23	Classification
AutoML-2	0.82M	17	7	Classification
AutoML-3	0.78M	17	54	Classification
Zillow	90.3K	31	27	Regression

4. DeepGBM.

4.3 Training for DeepGBM.

4.3.1.2 Online Update.

GBDT 모델은 오프라인으로 교육되므로 온라인 데이터에 학습을 포함할 경우 온라인 실시간 성능이 저하될 수 있다.

따라서 온라인 업데이트를 포함하지 않으며, end-to-end 손실만을 사용합니다.

DeepGBM을 온라인으로 사용할 경우 GBDT를 포함하고 처음부터 다시 교육하지 않고 L-online으로 모델을 업데이트하기 위해 새로운 데이터만을 필요로 한다.

이를 통해서 DeepGBM은 온라인 작업에 매우 효율적임을 알 수 있다.

또한 밀도가 높은 수치적 특징과 희박한 범주적 특징을 모두 다 처리할 수 있기 때문에 매우 효과적이다.

5. Experiment.

5.1 Experimental Setup.

5.1.1 Datasets

몇 개의 공개 표 형식 데이터 세트에 대해 DeepGBM에 대한 철저한 평가를 수행하고

그 성능을 널리 사용되는 여러 기준선 모델과 비교한다.

이후 오프라인 및 온라인 환경에서 DeepGBM의 성능을 분석하여 기준선 모델에 대한 효과와 이점을 입증하겠다.

Flight는 항공사 데이터 집합이며 비행이 지연되거나 지연되지 않는 것을 예측하는 데 사용된다. Criteo, Malware 및 Zillow은 Kaggle 대회의 데이터 세트이다.

AutoML-1, AutoML-2 및 AutoML-3은 NeurIPS 2018의 "AutoML for Lifelong Machine Learning" 챌린지의 데이터 세트이다.

이러한 데이터셋은 실제 작업에서 가져온 것이므로 범주형 및 숫자형 특징을 모두 포함하고 있다.

5. Experiment.

5.1 Experimental Setup.

Model	Binary Classification						Regression
	Flight	Criteo	Malware	AutoML-1	AutoML-2	AutoML-3	Zillow
LR	0.7234 \pm 5e-4	0.7839 \pm 7e-5	0.7048 \pm 1e-4	0.7278 \pm 2e-3	0.6524 \pm 2e-3	0.7366 \pm 2e-3	0.02268 \pm 1e-4
FM	0.7381 \pm 3e-4	0.7875 \pm 1e-4	0.7147 \pm 3e-4	0.7310 \pm 1e-3	0.6546 \pm 2e-3	0.7425 \pm 1e-3	0.02315 \pm 2e-4
Wide&Deep	0.7353 \pm 3e-3	0.7962 \pm 3e-4	0.7339 \pm 7e-4	0.7409 \pm 1e-3	0.6615 \pm 1e-3	0.7503 \pm 2e-3	0.02304 \pm 3e-4
DeepFM	0.7469 \pm 2e-3	0.7932 \pm 1e-4	0.7307 \pm 4e-4	0.7400 \pm 1e-3	0.6577 \pm 2e-3	0.7482 \pm 2e-3	0.02346 \pm 2e-4
PNN	0.7356 \pm 2e-3	0.7946 \pm 8e-4	0.7232 \pm 6e-4	0.7350 \pm 1e-3	0.6604 \pm 2e-3	0.7418 \pm 1e-3	0.02207 \pm 2e-5
GBDT	0.7605 \pm 1e-3	0.7982 \pm 5e-5	0.7374 \pm 2e-4	0.7525 \pm 2e-4	0.6844 \pm 1e-3	0.7644 \pm 9e-4	0.02193 \pm 2e-5
DeepGBM (D1)	0.7668 \pm 5e-4	0.8038 \pm3e-4	0.7390 \pm 9e-5	0.7538 \pm 2e-4	0.6865 \pm 4e-4	0.7663 \pm3e-4	0.02204 \pm 5e-5
DeepGBM (D2)	0.7816 \pm5e-4	0.8006 \pm 3e-4	0.7426 \pm5e-5	0.7557 \pm2e-4	0.6873 \pm3e-4	0.7655 \pm 2e-4	0.02190 \pm2e-5
DeepGBM	0.7943 \pm2e-3	0.8039 \pm3e-4	0.7434 \pm2e-4	0.7564 \pm1e-4	0.6877 \pm8e-4	0.7664 \pm5e-4	0.02183 \pm3e-5

5. Experiment.

5.1 Experimental Setup.

5.1.2 Compared Models.

GBDT(LightGBM) : 트리 기반 학습 알고리즘이며, 높은 효율성을 위해 LightGBM

LR : 로지스틱 회귀 분석

FM : 선형 모델과 FM 구성 요소를 포함합니다.

Wide&Deep : 얇은 선형 모델과 심층 신경망을 결합한 모델.

DeepFM : Wide&Deep에 FM 구성 요소를 추가하여 개선한 모델.

PNN : Probabilistic neural network(확률적 신경망)

DeepGBM : CatNN + GBDT2NN

DeepGBM(D1) : CatNN + GBDT

DeepGBM(D2) : GBDT2NN

5. Experiment.

5.1 Experimental Setup.

5.1.3 Experiments Settings.

LR, FM, Wide&DeepFM 및 PNN과 같은 수치 특징들 중 잘 처리되지 않는 모델의 수치 특징을 범주형 특징으로 구분한다.

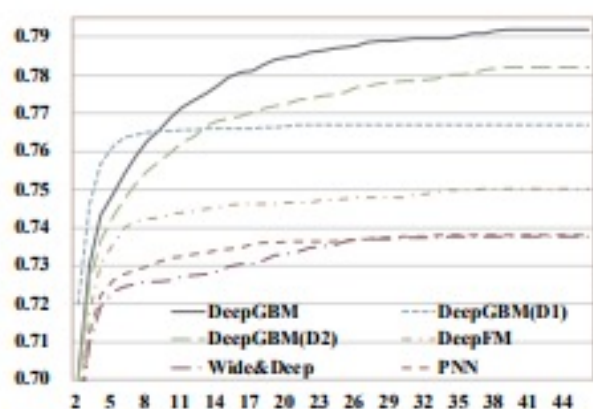
GBDT 및 이를 기반으로 하는 모델과 같이 범주형 특징을 잘 처리할 수 없는 모델의 경우 레이블 인코딩, 이진 인코딩을 통해 범주형 특징을 숫자형 특징으로 변환한다.

이러한 기반으로 모든 모델은 범주형 및 수치 형상 모두의 정보를 사용할 수 있기때문에 비교를 보다 신뢰할 수 있다.

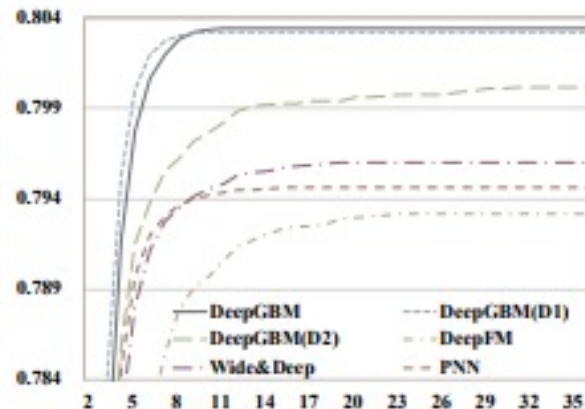
그리고 공정한 비교를 위해서 서로 다른 무작위 시드로 5회 실행되었다.

5. Experiment.

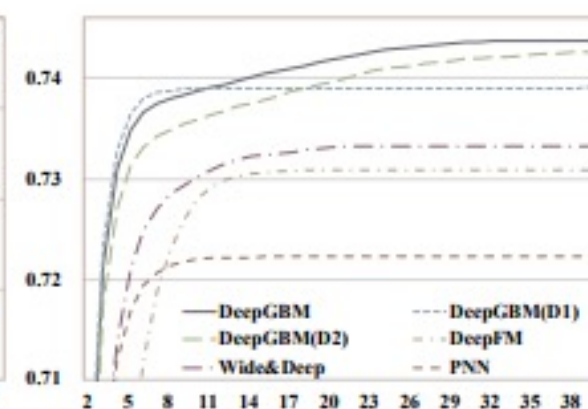
5.2 Offline Performance.



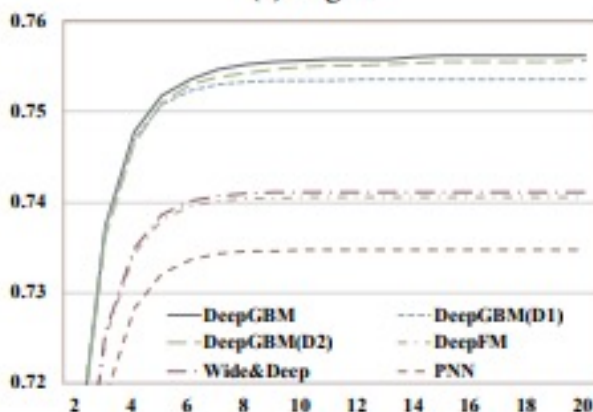
(a) Flight



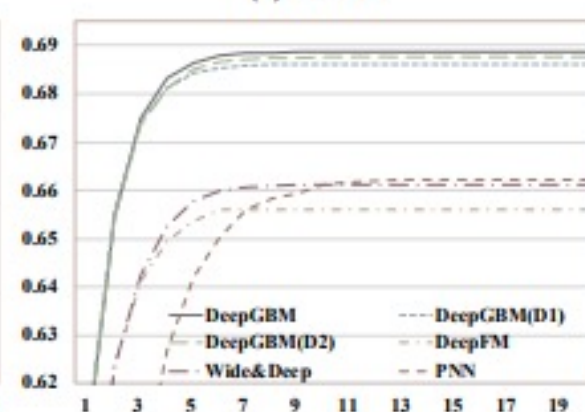
(b) Criteo



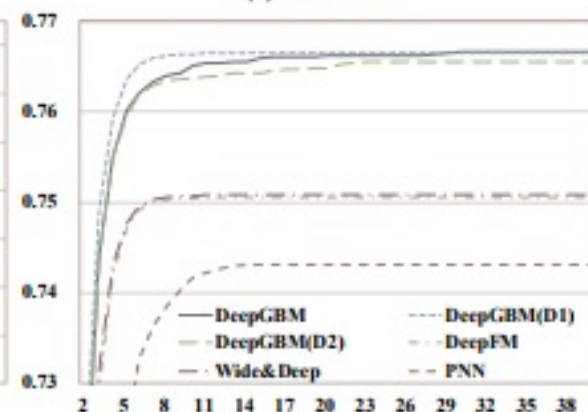
(c) Malware



(d) AutoML-1



(e) AutoML-2



(f) AutoML-3

5. Experiment.

5.2 Offline Performance.

실제 학습을 위해서 데이터셋을 훈련셋과 테스트 세트로 분할하고 학습을 진행하였으며, 학습에 오래된 데이터셋 90%가 사용되고, 새로운 샘플 10%가 사용된다.

전체 비교 결과는 표 3에서 확인 가능하며, 결과에 대한 분석은 다음과 같다.

GBDT는 다른 NN 기준선을 능가하며, GBDT 지식을 습득하면 DeepGBM에 확실히 도움이 된다는 것을 알 수 있다.

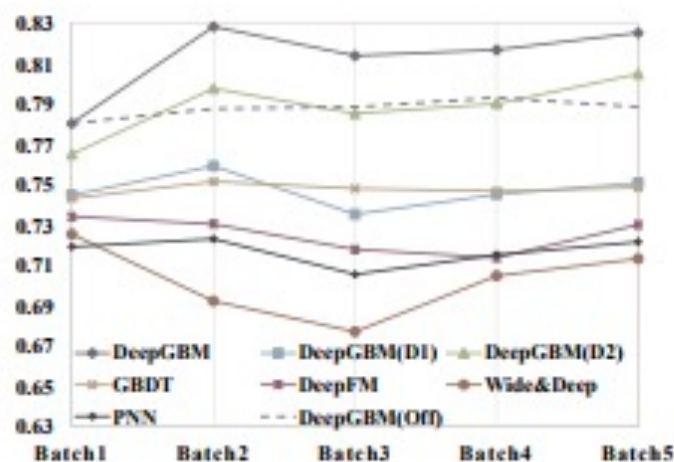
GBDT2NN(DeepGBM(D2))은 GBDT를 더욱 개선할 수 있으며, GBDT2NN이 훈련된 GBDT 모델을 NN으로 효과적으로 증류할 수 있음이 나타나며, 증류된 NN 모델을 더욱 개선하고 GBDT를 능가할 수 있음을 의미한다.

GBDT와 NN을 결합하면 성능을 더욱 향상시킬 수 있다. DeepGBM(D1)과 DeepGBM은 모두 단일 모델 기준선보다 더 나은 성능에 도달할 수 있었다.

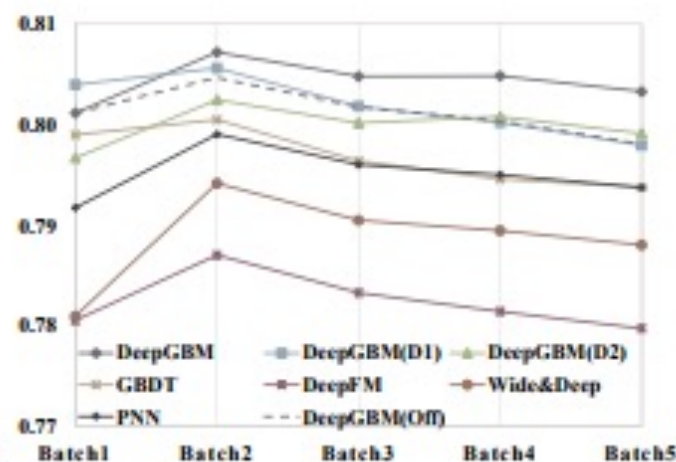
DeepGBM은 모든 데이터 세트의 모든 기준선을 능가하며, 다른 모델보다 훨씬 빠르게 수렴된다는 것을 알 수 있다.

5. Experiment.

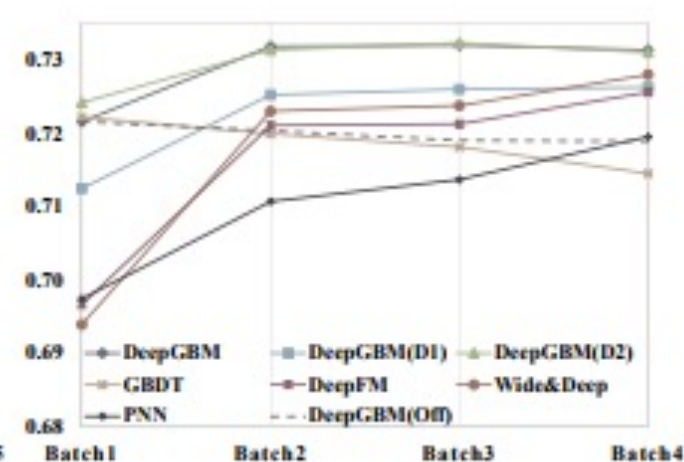
5.2 Online Performance.



(a) Flight



(b) Criteo



(c) AutoML-1

5. Experiment.

5.2 Online Performance.

온라인 성능을 평가하기 위해 Flight, Criteo, AutoML-1 데이터셋을 온라인 벤치마크로 사용한다.

데이터 세트를 여러 개의 연속적인 배치로 분할하며, 각 배치에 대한 모델을 가장 오래된 것부터 가장 최근의 것 순으로 훈련시킨다.

성능 결과를 분석한 결과, GBDT는 온라인 시나리오를 제대로 수행하지 못하며,

GBDT2NN(DeepGBM(D2))의 온라인 시나리오의 성능이 GBDT를 크게 능가한다.

또한 DeepGBM은 GBDT2NN 대신 GBDT를 사용하는 DeepGBM(D1)의 성능을 크게 능가합니다. 이는 GBDT에 의해 증류된 NN 모델이 온라인 시나리오에서 더욱 효과적임을 알 수 있다.

모든 실험결과 결과적으로 DeepGBM이 오프라인 및 온라인 시나리오 모두에서 모든 종류의 기준선을 크게 능가할 수 있음을 보인다.

6. Conclusion.

온라인 예측 작업에서 희박한 범주형 특징과 밀도가 높은 수치적 특징의 존재를 나타내는 표 입력 공간과 잠재적으로 동적 분포가 있는 연속적인 작업 생성 데이터를 의미하는 온라인 데이터 생성의 과제를 해결하기 위해 NN과 GBDT를 함께 사용한 새로운 프레임워크 DeepGBM을 제안한다.

DeepGBM은 CatNN(희소 범주 형상의 입력이 있는 NN 구조), GBDT2NN(밀도가 높은 수치 형상의 입력이 있는 또 다른 NN 구조)를 함께 사용하는 구조를 가진다.

이러한 DeepGBM은 두 가지 구성 요소를 이용하여 온라인 학습의 중요한 능력을 유지하면서 범주형 및 수치형 특징 모두에 대해 강력한 학습 용량을 처리할 수 있다.

그리고 종합적인 실험 결과 DeepGBM은 오프라인 및 온라인 시나리오 모두에서 다른 모델을 능가함을 보여주었다.