💚

TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED

> 💡 **정형** 데이터 분석에 딥러닝 모델을 활용해 보자!

1. Introduction

- Deep Neural Network는 이미지, 오디오, 텍스트와 같은 도메인에서 큰 성공을 거두었으나, 표 형식 데이터에 적용하는 데에는 어려움이 많았음
  - 표 형식 데이터는 의학/금융 등 다양한 분야에서 사용되며, 전통적으로 그래디언트 부스팅 결정 트리(GBDT)와 같은 기계 학습 방법이 우수한 성능을 보여왔음
  - 최근 몇몇 연구에서는 딥 네트워크가 표 형식 데이터에서 GBDT를 능가할 수 있다고 주장하지만, 일관된 비교가 어렵고, 최적화 과정도 동등하지 않음
- 이 연구는 다양한 표 형식 데이터셋에서 최근 제안된 deep model들과 XGBoost를 동일한 튜닝 프로토콜을 사용하여 평가
  - 연구 결과, 딥 모델들은 제안된 논문에서 사용된 데이터셋에서는 우수한 성능을 보이지만, 다른 데이터셋에서는 성능이 떨어짐
  - XGBoost는 대부분의 데이터셋에서 더 나은 성능을 보였음
  - 또한, XGBoost는 하이퍼 파라미터 검색 시간이 더 짧았음
- 긍정적인 측면으로, deep model과 XGBoost를 결합한 앙상블이 가장 좋은 성능을 보였음
  - 결론적으로, 딥 러닝이 표 형식 데이터에 대해 모든 문제를 해결하는 것은 아니며, 체계적인 연구가 필요함을 강조

2. Background

- 일반적으로 정형 데이터 분석에는 Gradient Boosting Decision Tree, XGBoost, LightGBM, 그리고 CatBoost가 사용됨

2-1. Deep Neural Models for Tabular Data

- 최근 몇몇 연구에서는 표 형식 데이터 도메인에 딥러닝을 적용하여 향상된 성능을 달성하기 위한 새로운 신경망 아키텍처가 도입되었음
- 크게 두 가지 범주로 나눌 수 있음
  1. 미분 가능한 트리
     - 결정 트리를 미분 가능하게 만드는 방법을 모색

       → for end-to-end training pipeline
     - 몇몇 연구에서는 트리 내부 노드의 결정 함수를 부드럽게 하여 트리 함수와 트리 라우팅을 미분 가능하게 만듦
  2. 어텐션 기반 모델
     - 표 형식 딥 네트워크에도 어텐션과 유사한 모듈을 사용하는 것을 제안하였음
     - 최근 연구에서는 주어진 샘플의 특징들이 서로 상호작용하는 샘플 간 어텐션(inter-sample attention)과 데이터 포인트들이 전체 행/샘플을 사용하여 상호작용하는 샘플 내 어텐션(intra-sample attention)이 제안되었음
- 최근 제안된 표 형식 데이터 학습을 위한 딥 모델 중에서 트리 앙상블을 능가한다고 주장되며 업계에서 큰 관심을 끈 **네 가지 모델**을 검토

TabNet

- 여러 데이터셋에서 우수한 성능을 보인 end-to-end 딥러닝 모델
- 희소 학습 마스크를 사용하여 특징을 인코딩하는 순차적 결정 단계를 포함한 인코더가 있으며, 각 행에 대해 관련 특징을 선택
- sparsemax 레이어를 사용하여 인코더가 소수의 특징만 선택하도록 강제
- learnable mask는 soft한 결정을 내릴 수 있음

  → 전통적 특징 선택 방법 완화

Neural Oblivious Decision Ensembles(NODE)

- 오류 그래디언트가 역전파될 수 있도록 미분 가능한 동일 깊이의 망각 결정 트리(ODTs)를 포함

- 전통적인 결정 트리와 마찬가지로 선택된 특징에 따라 데이터를 분할하고 각 특징을 학습된 임계값과 비교
    - 각 레벨에서는 하나의 특징만 선택됨

DNF-Net

- 아이디어: 딥 뉴럴 네트워크에서 합동 정상 형식(DNF)을 시뮬레이션하는 것
- 주요 특징
    - 완전 연결 레이어
    - literal에 대한 이항 결합의 soft version으로 형성된 DNNF 레이어로 구성된 결합 정상 뉴럴 폼(DNNF) 블록

  ⇒ 전체 모델은 이러한 DNNF의 앙상블임

1D-CNN

- 최근 tabular data에 대해 SOTA를 달성한 단일 모델
- CNN 구조가 특징 추출에 우수한 성능을 발휘한다는 아이디어에 기반
    - 그러나 특징의 순서에 지역성이 없기에 표 형식 데이터에 거의 사용되지 않음
- 해당 모델에서는 완전 연결 레이어를 사용하여 지역성 특징을 가진 더 큰 세트를 생성하고, 여러 1D-컨볼루션 레이어와 단축 연결(shortcut-like connections)을 사용함

2-2. Model Ensemble

- 앙상블 학습은 여러 모델을 훈련하고 그 예측을 결합하여 성능을 향상시키고 분산을 줄이는 방법임
    - 다양한 서브모델의 출력을 결합하여 보다 안정적이고 정확한 결과를 제공
    - 앙상블 학습은 서로 다른 기계 학습 방법이 다양한 상황에서 다르게 작동할 수 있다는 가정에 기반하며, 여러 방법을 결합하면 더 나은 결과를 얻을 수 있음
- 크게 두 가지 유형으로 나뉨
    - 랜덤화 기법: 랜덤 포레스트와 같이 각 구성원이 다른 초기 매개변수와 훈련 데이터를 사용
    - 부스팅 기법: 기본 학습자들을 순차적으로 맞추어 훈련
    - 대부분의 앙상블은 결정 트리를 기본 학습자로 사용
- 주요 앙상블 기법에는 배깅과 부스팅이 있음

    - 배깅은 무작위로 선택된 훈련 데이터 부분 집합을 사용하여 여러 결정 트리를 생성하고 투표를 통해 최종 결과를 결합
    - 부스팅은 각 훈련 반복 후에 샘플 가중치를 업데이트하고 가중 투표를 사용하여 출력을 결합하며, 새로운 모델을 추가하여 기존 모델의 오류를 조정
    - 또한, 스태킹은 신경망의 출력을 선형 회귀로 결합하는 방법임
- 앙상블을 수행하는 방법으로 두 가지를 제안
    1. 동일한 weight를 가지는 혼합 모델로 간주

       → 예측을 다음과 같이 결합

$$p(y|x) = \sum_{k=1}^{K} p_{\theta_m}(y|x, \theta_m)$$

    2. 각 모델에 대해 가중 평균을 계산(= 다른 가중치)

$$p(y|x) = \sum_{k=1}^{K} l_k^{\text{val}} p_{\theta_m}(y|x, \theta_m)$$

3. Comparing the Models

3-1. Experimental Setup

- 다양한 테이블 데이터셋에서 제안된 딥 모델의 장점을 조사
    - 실제 응용에서 모델은 정확하고, 효율적으로 훈련 및 추론하며, 최적화 시간이 짧아야 함
    - 이를 위해 딥 모델, XGBoost 및 앙상블의 성능을 다양한 데이터셋에서 평가하고, 앙상블의 구성 요소를 분석
- 또한, 앙상블을 위한 모델 선택 방법을 조사하고, 딥 모델이 좋은 결과를 내는 데 필수적인지, 또는 XGBoost, SVM, CatBoost와 같은 '클래식' 모델의 조합만으로도 충분한지 테스트

데이터셋 설명

- 11개의 다양한 분류 및 회귀 문제를 포함한 데이터셋을 사용

    - 각 데이터셋은 원 논문에서 설명한 대로 전처리 및 훈련되었으며, 평균이 0이고 분산이 1이 되도록 표준화됨
    - 사용된 데이터셋: Forest Cover Type, Higgs Boson, Year Prediction, Rossmann Store Sales, Gas Concentrations, Eye Movements, Gesture Phase, MSLR, Epsilon, Shrutime, Blastchar

구현 세부 사항

- 하이퍼파라미터 최적화를 위해 HyperOpt를 사용했으며, 각 데이터셋에서 1000단계 동안 검색을 실행했음
    - 모델의 하이퍼파라미터는 원 논문에서 가져왔음
- 데이터셋은 훈련, 검증, 테스트 세트로 분할되었으며, 분할 방법은 원 논문과 동일

메트릭 및 평가

- 이진 분류 문제에서는 교차 엔트로피 손실을, 회귀 문제에서는 평균 제곱근 오차를 활용
- 각 튜닝된 구성에 대해 다른 무작위 시드로 네 번의 실험을 수행하고, 테스트 세트에서 성능을 보고

통계적 유의성 테스트

- 프리드먼 테스트를 사용하여 모델 간의 성능 차이가 통계적으로 유의미한지 평가
- p-값이 0.05보다 크면 귀무 가설을 기각하지 않으며, 작으면 95% 신뢰 수준에서 귀무 가설을 기각

훈련

- 분류 데이터셋에서는 교차 엔트로피 손실을, 회귀 데이터셋에서는 평균 제곱근 오차를 최소화
- 딥 모델의 경우 Adam 최적화를 사용하고 학습률 스케줄을 사용하지 않음
- 검증 세트에서 100회 연속으로 개선이 없을 때까지 훈련을 지속

3-2. Results

Deep model이 다른 데이터셋에 대해 잘 일반화되는가

- 딥러닝 모델들에 대해 원래 포함되지 않은 데이터셋에서 성능이 얼마나 좋은지를 탐구하고 이를 XGBoost와 비교하였음

- 실험 결과, 대부분의 경우 딥 모델들은 원래 논문에 포함된 데이터셋보다 성능이 낮았고, XGBoost가 이러한 딥 모델들보다 더 나은 결과를 보였음
- 딥 모델들 중에서도 특정 데이터셋에서만 우수한 성능을 보이며, 어떤 딥 모델도 다른 모델들보다 일관되게 우월하지는 않았음

- 그러나 딥 모델들과 XGBoost의 앙상블은 대체로 다른 모델들보다 더 나은 성능을 보였음

⇒ 딥 모델들이 특정 데이터셋에 민감할 수 있음을 보여주며, 이는 선택 편향과 하이퍼파라미터 최적화의 차이로 설명 가능

XGBoost와 Deep Network가 둘 다 필요한가

- XGBoost를 딥 모델과 결합해야 하는지, 아니면 간단한 non-deep 모델의 앙상블이 유사한 성능을 낼 수 있는지에 대해 고민
- 이를 비교해보기 위해 XGBoost와 SVM, 그리고 Catboost의 non-deep model들의 앙상블을 훈련하였음

Model Name	Rossman	CoverType	Higgs	Gas	Eye	Gesture
XGBoost	490.18 ± 1.19	3.13 ± 0.09	21.62 ± 0.33	2.18 ± 0.20	**56.07**±0.65	80.64 ± 0.80
NODE	488.59 ± 1.24	4.15 ± 0.13	21.19 ± 0.69	2.17 ± 0.18	68.35 ± 0.66	92.12 ± 0.82
DNF-Net	503.83 ± 1.41	3.96 ± 0.11	23.68 ± 0.83	**1.44** ±0.09	68.38 ± 0.65	86.98 ± 0.74
TabNet	**485.12**±1.93	3.01 ± 0.08	**21.14**±0.20	1.92 ± 0.14	67.13 ± 0.69	96.42 ± 0.87
1D-CNN	493.81 ± 2.23	3.51 ± 0.13	22.33 ± 0.73	1.79 ± 0.19	67.9 ± 0.64	97.89 ± 0.82
Simple Ensemble	488.57 ± 2.14	3.19 ± 0.18	22.46 ± 0.38	2.36 ± 0.13	58.72 ± 0.67	89.45 ± 0.89
Deep Ensemble w/o XGBoost	489.94 ± 2.09	3.52 ± 0.10	22.41 ± 0.54	1.98 ± 0.13	69.28 ± 0.62	93.50 ± 0.75
Deep Ensemble w XGBoost	485.33 ± 1.29	**2.99** ±0.08	22.34 ± 0.81	1.69 ± 0.10	59.43 ± 0.60	**78.93** ±0.73
		TabNet			DNF-Net	

Model Name	YearPrediction	MSLR	Epsilon	Shrutime	Blastchar
XGBoost	77.98 ± 0.11	55.43±2e-2	11.12±3e-2	13.82 ± 0.19	20.39 ± 0.21
NODE	76.39 ± 0.13	55.72±3e-2	**10.39**±1e-2	14.61 ± 0.10	21.40 ± 0.25
DNF-Net	81.21 ± 0.18	56.83±3e-2	12.23±4e-2	16.8 ± 0.09	27.91 ± 0.17
TabNet	83.19 ± 0.19	56.04±1e-2	11.92±3e-2	14.94±, 0.13	23.72 ± 0.19
1D-CNN	78.94 ± 0.14	55.97±4e-2	11.08±6e-2	15.31 ± 0.16	24.68 ± 0.22
Simple Ensemble	78.01 ± 0.17	55.46±4e-2	11.07±4e-2	13.61±, 0.14	21.18 ± 0.17
Deep Ensemble w/o XGBoost	78.99 ± 0.11	55.59±3e-2	10.95±1e-2	14.69 ± 0.11	24.25 ± 0.22
Deep Ensemble w XGBoost	**76.19** ±0.21	**55.38**±1e-2	11.18±1e-2	**13.10**±0.15	**20.18**±0.16
		NODE		New datasets	

Table 2: **Test results on tabular datasets.** Presenting the performance for each model. MSE is presented for the YearPrediction and Rossman datasets, and cross-entropy loss (with 100X factor) is presented for the other datasets. The papers that used these datasets are indicated below the table. The values are the averages of four training runs (lower value is better), along with the standard error of the mean (SEM)

Subset of models

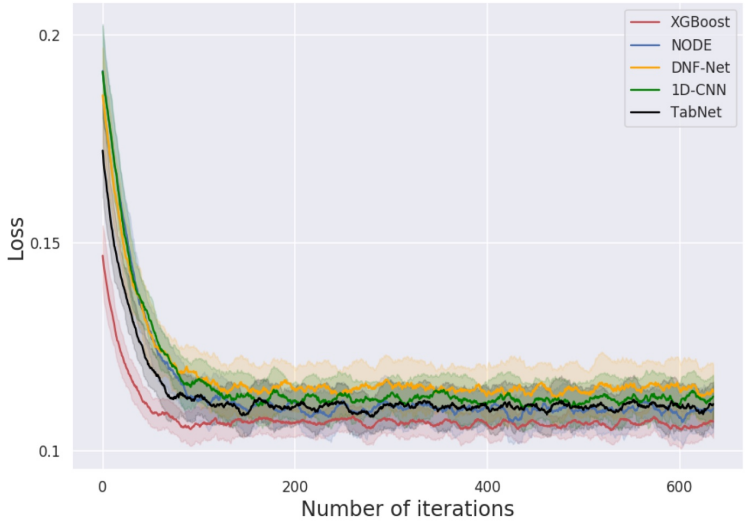
- XGBoost와 딥 모델들의 앙상블이 다양한 데이터셋에서 최고의 성능을 보임
  - 이에 따라 XGBoost와 다른 non-deep 모델들(SVM 및 CatBoost)의 앙상블이 비슷한 성능을 낼 수 있는지를 살펴보기 위해 실험을 진행

- 실험 결과, 전통적인 모델들의 앙상블이 deep 네트워크와 XGBoost의 앙상블보다 훨씬 낮은 성과를 보였음

- 또한, deep 모델들만으로 구성된 앙상블은 좋은 결과를 보이지 않았음
  - 이러한 결과는 모두 통계적으로 유의미했으며(모든 p값이 0.005 미만), 딥 모델과 XGBoost를 결합하는 것이 이 데이터셋에서의 성능 향상을 가져온다는 점을 시사

최적화의 어려움

- 실제 환경에서는 새로운 데이터셋에 대해 모델을 훈련하고 하이퍼 파라미터를 최적화하는 데 시간과 자원이 제한적임
  - 이에 대한 이해를 위해 각 모델이 이 작업을 어렵게 하는지를 평가하는 방법 중 하나는 모델이 필요로 하는 연산 수를 측정하는 것임
  - 이는 일반적으로 초당 부동 소수점 연산(FLOPS)으로 표시됨
- 다른 방법으로는 모델을 훈련하고 최적화하는 데 필요한 총 시간을 비교하는 것임
  - 실험 결과로는 XGBoost가 딥 네트워크보다 훨씬 빠르다는 것을 발견했으며, 이는 소프트웨어 최적화 수준에 따라 크게 달라질 수 있음
- 또 다른 접근 방식은 하이퍼파라미터 최적화 프로세스의 반복 횟수를 비교하는 것임
  - 이는 모델의 견고성과 초기 하이퍼파라미터 설정의 중요성을 보여줌
  - Figure 2에서는 Shrutime 데이터셋에 대한 이러한 접근 방식을 시각적으로 나타내고 있음



4. Discussion and Conclusions

- 최근에 제안된 딥러닝 모델들이 타블러 데이터에서 어떻게 성능을 발휘하는지를 조사
    - 연구 결과, 이러한 딥 모델들은 원래 논문에 포함되지 않은 데이터셋에서 성능이 XGBoost와 비교해 상대적으로 떨어짐을 보였음
    - 따라서 우리는 XGBoost와 딥 모델들을 함께 사용하는 앙상블을 제안했고, 이 앙상블은 개별 모델들보다 더 나은 성능을 나타내었음
    - 또한, 성능, 계산 리소스 소모, 하이퍼파라미터 최적화 시간 등의 요소 사이의 가능한 트레이드오프를 탐색하였음
- 우리의 분석은 딥 모델들의 성능을 평가할 때 신중함이 필요함을 강조
    - 다른 데이터셋에서도 비슷한 결과를 얻었으며, 딥 모델들을 새로운 데이터셋에서 최적화하는 것이 XGBoost보다 어려움을 확인하였음
    - 그러나 XGBoost와 딥 모델들을 결합한 앙상블이 우리가 연구한 데이터셋에서 가장 좋은 결과를 제공하였음

  ⇒ 연구자들이 실제 문제에 모델을 적용할 때 여러 요소를 고려해야 함을 시사
- 그러나 XGBoost만으로는 최상의 성능을 달성하기 어려울 수 있으며, 딥 모델들을 포함한 앙상블을 고려하여 성능을 극대화할 필요가 있음을 확인