TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED

AYŞE ASUDE ÇAKIN

Authors

- Ravid Shwartz-Ziv (ravid.ziv@intel.com) IT AI Group, Intel
- Amitai Armon (amitai.armon@intel.com) IT AI Group, Intel

Publish Date

November 24, 2021

Abstract

- Several deep learning models that recently proposed, claims that **deep learning models outperform XGBoost for some use cases**.
- This paper explores **whether deep learning models should be preferred instead of XGBoost** by rigorously comparing the proposed deep learning models to XGBoost.
- The paper also compares the required computation to tune hyperparameters for each model.

Abstract

The paper shows:

- XGBoost outperforms these deep learning models on various datasets, including the datasets used in the papers that proposed the deep models.
- XGBoost require less tuning.
- Ensemble of deep learning models and XGBoost performs the best.

Background

- Gradient-boosted decision trees(GBDT):
  - “weak” models
  - “strong” model that composed of “weak” models
  - XGBoost: New models are created from previous models’ residuals and then combined to make final prediction
  - GBDTs dominate tabular data applications

Background

- Challenges of deep neural networks when applied to tabular data:
  - Lack of locality
  - Data sparsity (missing values)
  - Mixed feature types (numerical, ordinal, and categorical)
- Deep neural networks are block boxes.

Purpose & Approach

- Whether any of the recently proposed deep models should indeed be a recommended choice for tabular dataset problems.
- Two parts of this question:
  - Are the models more accurate, especially for <u>unseen datasets</u>?
  - How long does it take to train and tune these models compared to other models?

Purpose & Approach

- Authors evaluate deep learning models and XGBoost on diverse tabular datasets with the same tuning protocol.
- They use 11 datasets 9 of which were used in these papers.

Proposed Deep Learning Models

- Authors examine four models that have been claimed to outperform tree ensembles and attracted significant industry attention: TabNet, NODE, DNF-Net, 1D-CNN.
- **TabNet:** TabNet includes an encoder, in which features are encoded into sparse learned masks and select relevant features for each row using the mask.
- **Neural Oblivious Decision Ensembles (NODE):** The NODE network contains equal depth oblivious decision trees, which are differentiable such that error gradients can backpropagate through them.

Proposed Deep Learning Models

- **DNF-Net:** DNF-Net replaces the hard Boolean formulas with soft, differentiable versions of them.
- **1D-CNN:** 1D-CNN is based on the idea that CNNs performs well on feature extraction.

Ensemble Models

- Ensemble learning enhances classifier performance by combining the multiple outputs from many submodels (base learners). Final prediction is obtained by combining the predictions of each submodel.
- Ensembles tend to improve the prediction performance, and reduce variance, leading to more stable and accurate results.

Ensemble Models – Authors' Approach

- Authors use five classifiers in their ensemble: TabNet, NODE, DNF-Net, 1D-CNN, and XGBoost.
- They also use ensembles of XGBoost and classical machine learning models.

Model Comparison Metrics

1. Perform accurately
2. Be trained and make inferences efficiently
3. Have a short optimization time

Experimental Setup

Datasets:

- Authors use 11 datasets that includes classification and regression problems.
- The datasets include 10 to 2.000 features, 1 to 7 classes, 7.000 to 1.000.000 samples.
- 9 of 11 datasets are used on the papers that proposed deep models.
- 2 datasets are "unseen" by any of the models.

Dataset	Features	Classes	Samples	Source	Paper
Gesture Phase	32	5	9.8k	OpenML	DNF-Net
Gas Concentrations	129	6	13.9k	OpenML	DNF-Net
Eye Movements	26	3	10.9k	OpenML	DNF-Net
Epsilon	2000	2	500k	PASCAL Challenge 2008	NODE
YearPrediction	90	1	515k	Million Song Dataset	NODE
Microsoft (MSLR)	136	5	964k	MSLR-WEB10K	NODE
Rossmann Store Sales	10	1	1018K	Kaggle	TabNet
Forest Cover Type	54	7	580k	Kaggle	TabNet
Higgs Boson	30	2	800k	Kaggle	TabNet
Shrutime	11	2	10k	Kaggle	New dataset
Blastchar	20	2	7k	Kaggle	New dataset

Table 1: Description of the tabular datasets

Experimental Setup

Optimization Process:

- Authors used HyperOpt, which uses Bayesian optimization.
- Initial hyperparameters were taken from the papers.

Experimental Setup

Metrics and Evaluation:

- For binary classification problems cross-entropy loss, for regression problem root mean square error is used.

Statistical Significance Test:

- Friedman's test is used to assess whether differences between models is indeed significant.

Experimental Setup

Training:

- Authors follow the original implementations and use Adam optimizer.
- Training is continued until there are 100 consecutive epochs without improvement on the validation set.

Results

- In most cases, the deep learning models perform worse on unseen datasets than do the datasets' original models.
- XGBoost generally outperformed the deep models.

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

Results

- To directly compare between the different models, authors calculated for each dataset the **relative performance** of each model compared to the best model for that dataset.

Name	Average Relative Performance (%)
XGBoost	3.34
NODE	14.21
DNF-Net	11.96
TabNet	10.51
1D-CNN	7.56
Simple Ensemble	3.15
Deep Ensemble w/o XGBoost	6.91
Deep Ensemble w XGBoost	**2.32**

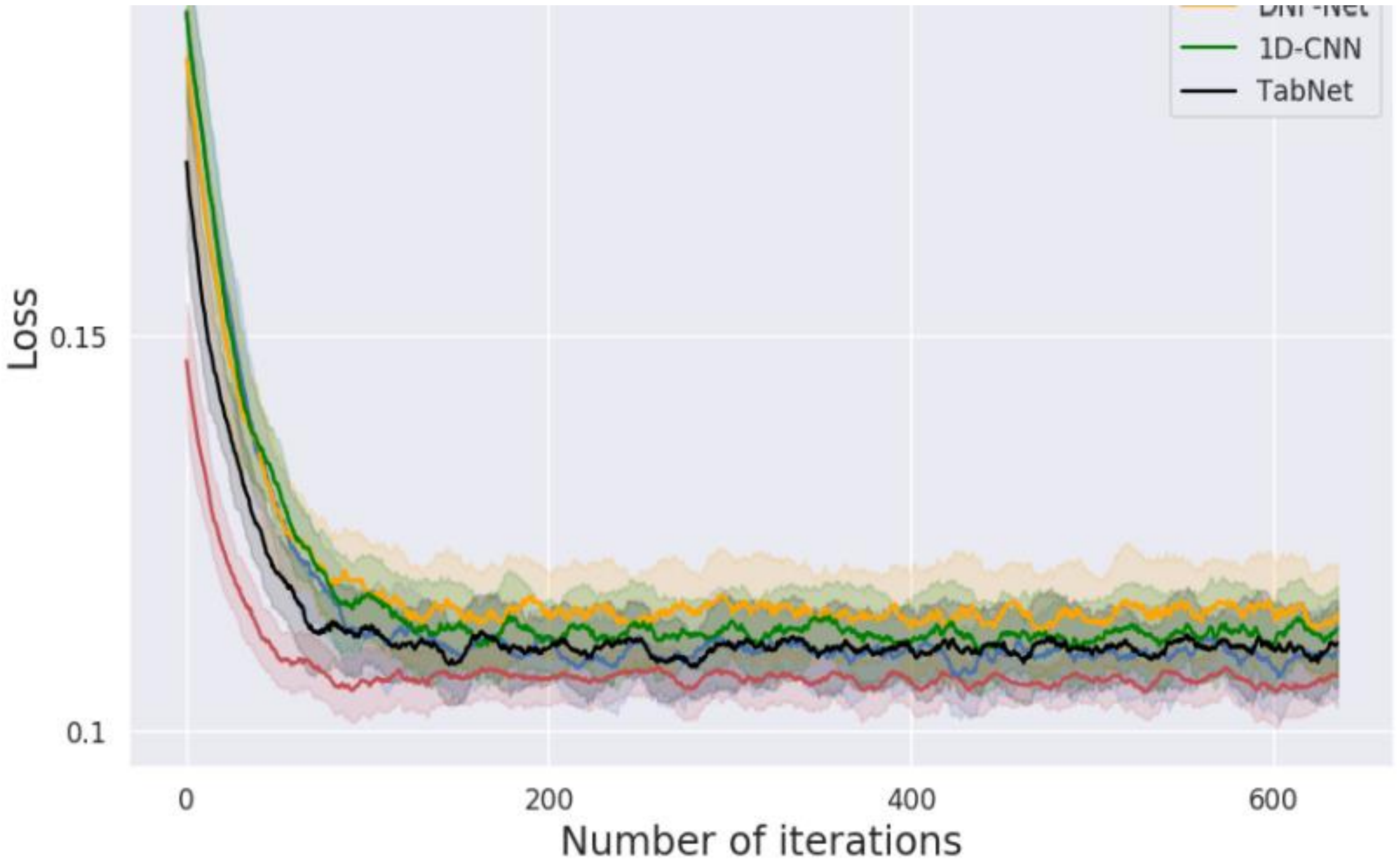
Table 3: Average relative performance deterioration for each model on its unseen datasets (lower value is better).

Possible reasons

- Selection bias: Each paper may have naturally demonstrated the model's performance on datasets with which the model worked well.
- Optimization of hyperparameters: Each paper may have set the model's hyperparameters based on a more extensive hyperparameter search.

Question: Do we need both XGBoost and deep networks?

How Difficult Is the Optimization?



Discussion and Conclusion

- The deep models were weaker on datasets that did not appear in their original papers, and they were weaker than XGBoost, the baseline model.
- Ensemble of XGBoost and deep models performed the best.
- **Take the reported deep models' performance with a grain of salt.**

Questions?