Why do tree-based models still outperform deep learning on tabular data?

Автор-докладчик: Стриженок Сергей

Рецензент-исследователь: Панеш Али

Хакер: Денисов Степан

ДОКЛАД

Введение

- Методы глубинного обучения хорошо работаю на текстовых и графических данных
- Древовидные модели остаются SOTA решениями на выборках порядка 10 тысяч примеров
- Превосходство глубинного обучения над древовидными моделями неочевидно

Идея

Провести набор экспериментов на воспроизводимых наборах данных со строго описанными условиями

Построение данных для эксперимента

Построено 45 выборок. Основные критерии:

- Неоднородные признаки
- Невысокая размерность
- Скрытие имен признаков, не потоковые данные
- Не искусственные данные
- Сложные данные
- Недетерминированный результат

Удаление побочных эффектов

- Сокращаем выборки до единого размера(10 000)
- Убираем пропуски
- Балансируем классы
- Убираем слишком узкие категориальные признаки и слишком широкие числовые

Подбор гиперпараметров

- Случайный поиск(400 итераций) на выборку
- Оцениваем с помощью бутстрапа

Оценка результатов

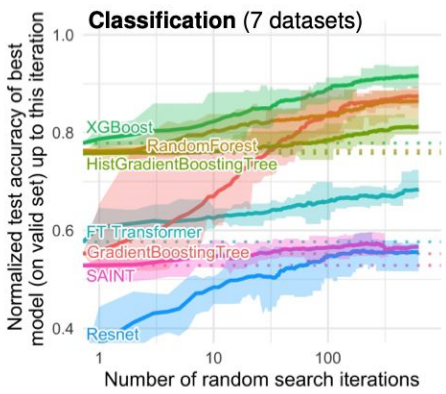
- Используем точность и R2 для соответствующих зада
- Оцениваем по среднему расстоянию до минимума

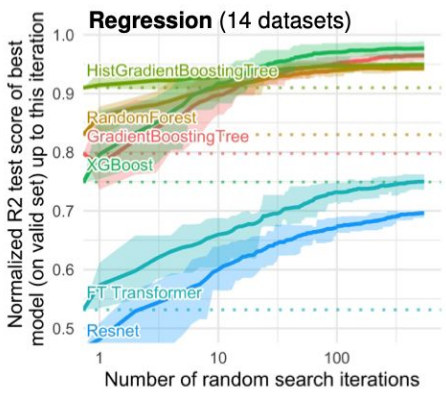
Небольшая подготовка данных

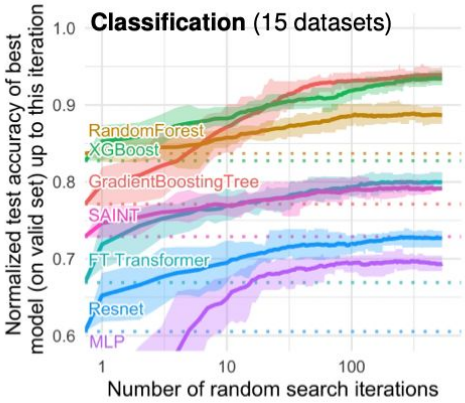
- Для сетей признаки проходят через QuantileTransformer
- Логарифмическое преобразование целевой переменной
- Кодирование категориальных признаков

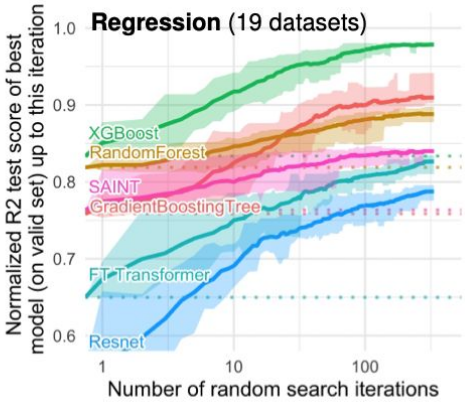
Эксперимент и результаты

Древовидные модели победили, даже без категориальных признаков.









Исследование

Какой inductive bias у древовидных моделей и у сетей?

- Нейронные сети склонны к чрезмерно гладким решениям
- Устойчивость относительно неинформативных признаков

Неинформативные признаки

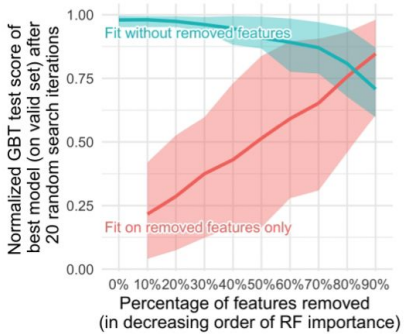
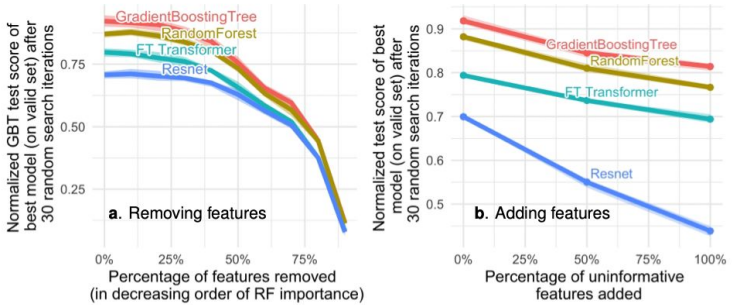


Figure 4: **Test accuracy of a GBT for varying proportions of removed features**, on our classification benchmark on numerical features. Features are removed in increasing order of feature importance (computed with a Random Forest), and the two lines correspond to the accuracy using the (most important) kept features (blue) or the (least important) removed features (red). A score of 1.0 corresponds to the best score across all models and hyperparameters on each dataset, and 0.0 correspond to random chance. These scores are averaged across 30 random search orders, and the ribbons correspond to the 80% interval among the different datasets.



Заключение

- Проведены эксперименты на выборках с понятными критериями
- Большая точка роста – исследование остальных выборок и дальнейшее изучение inductive bias

РЕЦЕНЗИЯ

Авторы

- Léo Grinsztajn - PHD в Soda, Inria Saclay
- Gaël Varoquaux - преподаватель в Soda, Inria Saclay, там же развивает skit-learn
- Edouard Oyallon - преподаватель в ISIR, CNRS, Sorbonne University, а также в École Polytechnique

Статья-родитель

Tabular Data: Deep Learning is Not All You Need
Shwartz-Ziv et al. 2021.

- Основное различие: датасеты

Статьи со сравнением

- Do we Need Hundreds of Classifiers to Solve Real World Classification Problems?
  Fernández-Delgado et al. 2014
- Comparison of machine learning techniques to predict all-cause mortality using fitness data: The Henry ford exercIse testing (FIT) project
  Sherif Sakr et al. 2017
- Comparison of Deep Learning With Multiple Machine Learning Methods and Metrics Using Diverse Drug Discovery Data Sets
  Korotcov et al. 2017
- Comparing different supervised machine learning algorithms for disease prediction
  Uddin et al. 2019

В чем проблема?

- Непонятные датасеты
- Мало датасетов на исследование
- В статье родителе - предложены датасеты ломающие модели

NN в таблицах

- Revisiting Deep Learning Models for Tabular Data
  Gorishniy et al. 2021
  Модели: MLP, RESNET, FT-Transformer

- Improved Neural Networks for Tabular Data via Row Attention and Contrastive Pre-Training
  Somepalli et al. 2021
  Модель: SAINT

NN в таблицах

$$\texttt{MLP}(x) = \texttt{Linear}\left(\texttt{MLPBlock}\left(\ldots\left(\texttt{MLPBlock}(x)\right)\right)\right)$$

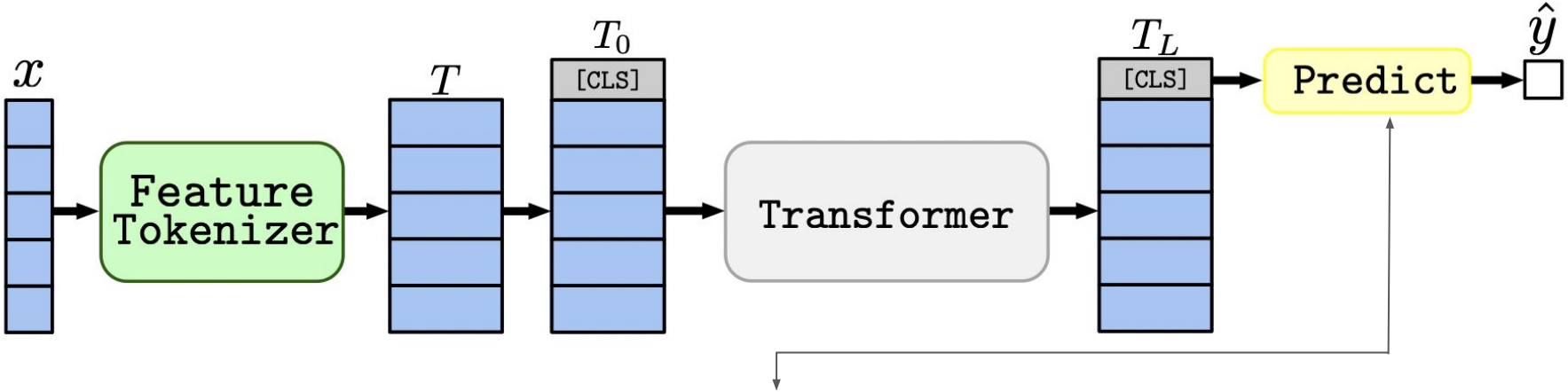
$$\texttt{MLPBlock}(x) = \texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(x)))$$

$$\texttt{ResNet}(x) = \texttt{Prediction}\left(\texttt{ResNetBlock}\left(\ldots\left(\texttt{ResNetBlock}\left(\texttt{Linear}(x)\right)\right)\right)\right)$$

$$\texttt{ResNetBlock}(x) = x + \texttt{Dropout}(\texttt{Linear}(\texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(\texttt{BatchNorm}(x))))))$$

$$\texttt{Prediction}(x) = \texttt{Linear}\left(\texttt{ReLU}\left(\texttt{BatchNorm}\left(x\right)\right)\right)$$

FT-Transformer



$$\hat{y} = \texttt{Linear}(\texttt{ReLU}(\texttt{LayerNorm}(T_L^{\texttt{[CLS]}}))).$$

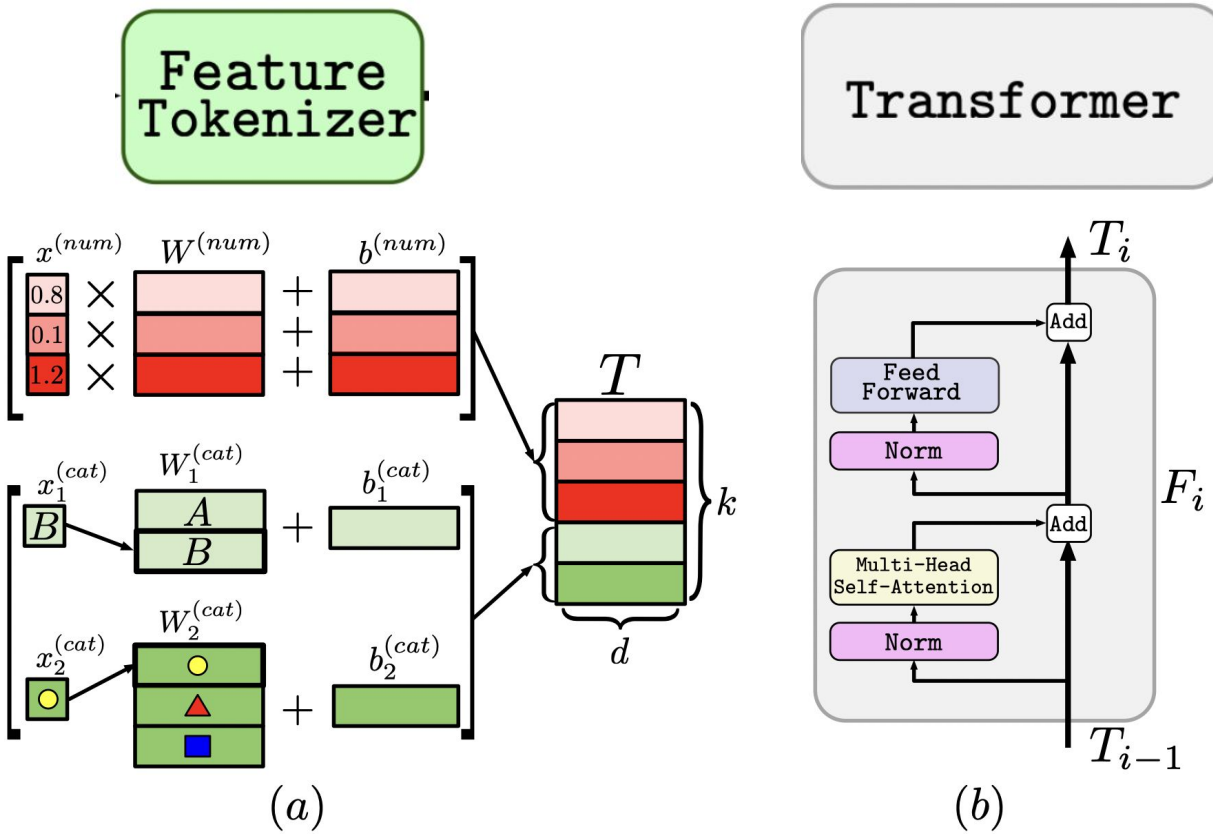
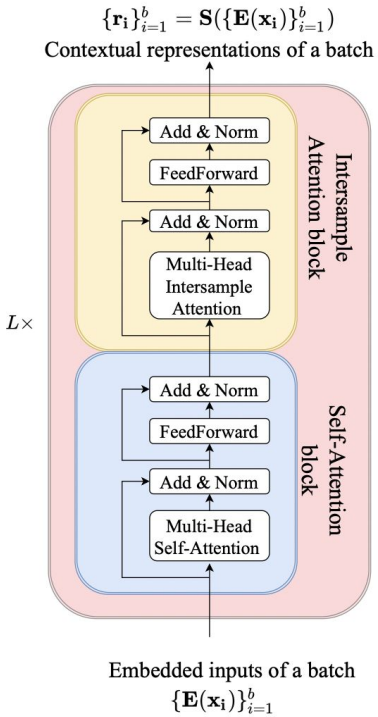
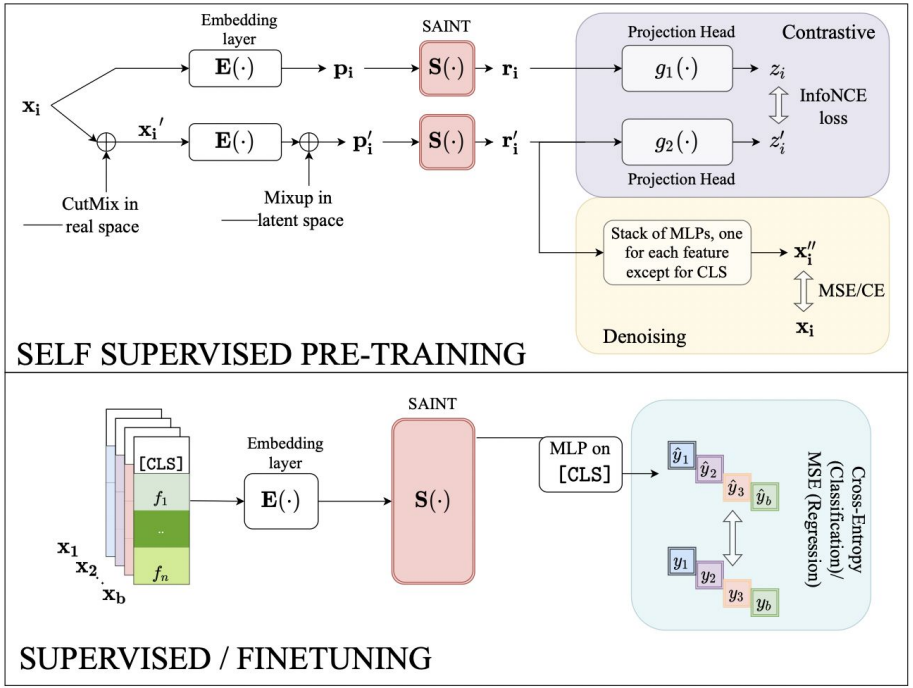


Figure 2: (a) Feature Tokenizer; in the example, there are three numerical and two categorical features; (b) One Transformer layer.

SAINT


$\{\mathbf{r_i}\}_{i=1}^{b} = \mathbf{S}(\{\mathbf{E}(\mathbf{x_i})\}_{i=1}^{b})$
Contextual representations of a batch
$L\times$
Add & Norm
FeedForward
Add & Norm
Multi-Head Intersample Attention
Intersample Attention block
Add & Norm
FeedForward
Add & Norm
Multi-Head Self-Attention
Self-Attention block
Embedded inputs of a batch
$\{\mathbf{E}(\mathbf{x_i})\}_{i=1}^{b}$

(a)


Embedding layer
SAINT
$\mathbf{x_i}$
$\mathbf{x_i}'$
$\mathbf{E}(\cdot)$
$\mathbf{p_i}$
$\mathbf{p_i}'$
$\mathbf{S}(\cdot)$
$\mathbf{r_i}$
$\mathbf{r_i}'$
CutMix in real space
Mixup in latent space
Projection Head
Contrastive
$g_1(\cdot)$
$g_2(\cdot)$
z_i
z_i'
InfoNCE loss
Projection Head
Stack of MLPs, one for each feature except for CLS
$\mathbf{x_i''}$
MSE/CE
$\mathbf{x_i}$
Denoising
SELF SUPERVISED PRE-TRAINING
[CLS]
f_1
f_n
$\mathbf{x_1}$
$\mathbf{x_2}$
$\mathbf{x_b}$
Embedding layer
$\mathbf{E}(\cdot)$
SAINT
$\mathbf{S}(\cdot)$
MLP on [CLS]
$\hat{y}_1$ $\hat{y}_2$ $\hat{y}_3$ $\hat{y}_b$
y_1 y_2 y_3 y_b
Cross-Entropy (Classification)/ MSE (Regression)
SUPERVISED / FINETUNING

(b)

Публикации

- Конференция NeurIPS 2022:
  Thirty-sixth Conference on Neural Information Processing Systems
  Datasets and Benchmarks Track.

- Встречена научным сообществом позитивно - 6.33

Сильные стороны

- Большое количество понятных, разнообразных табличных датасетов

- Неочевидные выводы: нарушение инвариантности может привести к лучшему перфомансу

Слабые стороны

- Подбор гиперпараметров: производился случайно
- Не используются “остановки” на древовидных моделях: не имеет смысла обучать маленькое количество деревьев
- RandomForest.feature_importance - не лучший показатель:
  Альтернативы - permutation_importances, dropcol_importances
  Beware Default Random Forest Importances, Parr et al. 2018
- Хочется смотреть и на другие типы датасетов

Предлагаемые улучшения

- Ввести более умный подбор гиперпараметров (optuna)
- Поменять RandomForest.feature_importance на предложенные метрики
- Поставить num_trees = 1000
- Добавить catboost
- Расширить датасеты датасетами большей размерностью, а также большей выборкой

Список литературы

- https://arxiv.org/abs/2207.08815
- https://arxiv.org/abs/2106.11959
- https://arxiv.org/abs/2106.01342
- https://explained.ai/rf-importance
- https://arxiv.org/abs/2106.03253
- https://jmlr.org/papers/volume15/delgado14a/delgado14a.pdf
- https://pubmed.ncbi.nlm.nih.gov/29258510/
- https://pubmed.ncbi.nlm.nih.gov/29096442/
- https://pubmed.ncbi.nlm.nih.gov/31864346/

ЭКСПЕРИМЕНТЫ

jupyter notebook