Дифференцируемый поиск нейросетевых архитектур для задач с табличными данными

Полюбин Арсений
ШАД
polyubinai@mai.ru

Чуб Вячеслав
ШАД
viacheslav.chub@math.msu.ru

Феоктистов Дмитрий
ШАД
feoktistovdd@my.msu.ru

2024

Аннотация

Работа с табличными данными остается той областью машинного обучения, в которой модели градиентного бустинга превосходят глубокие нейронные сети. Причина такого эффекта кроется в том, что нейронные сети хорошо работают с гомогенными данными, в то время как табличные данные являются гетерогенными. И если категориальные признаки сообщество умеет кодировать так, что нейронные сети хорошо их обрабатывают, то задача построения эмбеддингов для числовых признаков является значительно более сложной. В то же время методы автоматического поиска нейросетевых архитектур позволяют улучшить существующие архитектуры нейронных сетей. В данной работе авторами производится попытка использовать дифференцируемый поиск нейросетевых архитектур для улучшения существующих методов построения эмбеддингов для числовых признаков и, как следствие, улучшения архитектур нейронных сетей для табличных данных.

Ключевые слова: Поиск нейросетевых архитектур · Табличные данные · Эмбеддинги для числовых признаков

1 Введение

Глубокое обучение на табличных данных является наиболее сложной задачей в области. В то время как нейронные сети демонстрируют выдающиеся результаты в области компьютерного зрения и обработки естественного языка, в случае табличных данных таких прорывов не наблюдается. Несмотря на большое разнообразие архитектур [Arik and Pfister, 2021, Badirli et al., 2020], в большинстве задач они проигрывают ансамблям на основе решающих деревьев, хотя постепенно эта ситуация начинает меняться в пользу глубокого обучения [Gorishniy et al.].

Основной сложностью при работе с табличными данными является их неоднородность. Для категориальных признаков существует one-hot преобразование, которое решает эту проблемы и позволяет нейронным сетям хорошо обрабатывать данный тип признаков. В случае числовых признаков такого преобразования нет: при применении простых трансформаций, как standard scaler и quantile transform, теряется значительная часть информации. В большинстве архитектур табличных нейронных сетей выбираются весьма простые эмбеддинги для числовых признаков [Gorishniy et al., 2021, Guo et al., 2021], однако в новых работах показано, что правильное построение эмбеддингов числовых признаков может значительно увеличить качество модели [Gorishniy et al., 2022].

Методы автоматического поиска нейросетевых архитектур [Liu et al., 2018, Dong and Yang, 2019, Chen et al., 2021] зарекомендовали себя, как простой способ улучшить качество работы нейронных сетей для задач, которые не являются широко распространенными. Отдельно стоит отметить дифференцируемый поиск архитектур [Liu et al., 2018, Dong and Yang, 2019], так как его использование естественным образом соотносится с обучением нейронной сети.

Возникает идея использовать автоматический поиск нейросетевых архитектур для табличных данных. В данной работе мы решили сфокусироваться на улучшении следующих компонент нейронной сети: эмбеддинг слоя для числовых признаков и нейронной сети, предсказывающий целевое значение, используя полученные эмбеддинги.

2 Эмбеддинги для числовых признаков

Мы рассматриваем задачу обучения с учителем, обозначим датасет как $\{(x^j, y^j)\}_{j=1}^n$, где $y_j \in \mathbb{Y}$ – метка объекта, а $x^j = (x^{j(\text{num})}, x^{j(\text{cat})})$ – признаки объекта (числовые и категориальные соответственно).

Формализуем понятие эмбеддинга для числовых признаков следующим образом: $z_i = f_i(x_i^{(\text{num})}) \in \mathbb{R}^{d_i}$, где f_i – функция вычисляющая эмбеддинг для признака i, z_i – соответствующий эмбеддинг размерности d_i. Предполагается, что все эмбеддинги обучаются независимо, то есть разделение параметров не происходит, но при этом используется один функциональный вид для всех f_i.

В работе [Gorishniy et al., 2022] предложен PLR слой для построения эмбеддингов: первоначальные эмбеддинги получаются с помощью quantile-transform, после чего к ним применяется периодическая функция активации, а следом линейный слой и активация ReLU.

3 Дифференцируемый поиск нейросетевых архитектур

В работе мы опираемся на архитектуру из статьи [Liu et al., 2018], переписанную под свою модель. Составляется суперсеть, который представляет собой набор ячеек, где каждая ячейка является ациклическим графом, ребра которого- операции из пространства поиска, а узлы - это латентное пространство признаков. Все обучение строится на двухуровневой оптимизации параметров: α из пространтсва операций и weights - веса модели MLP.

Используются поочередные шаги градиентного спуска по α и weigts. Для быстроты вычисления второго градиента по α пользуемся апроксимацией оптимальных весов текущего слоя только одним шагом обучения. Таким образом сложность алгоритма упадет с $O(|\alpha| * |weights|)$ до $O(|\alpha| + |weights|)$

4 Пространство поиска

Мы будем использовать следующее пространство поиска:

1. Функции активации: ReLU(), Tanh(), Sigmoid(), Identity(), Zero(). Zero() тут добавлена для того, чтобы мы могли определить пустую операцию в ячейке, если у нас изначально не ребра между двумя узлами.
2. MLP слои
3. BatchNorm, dropout

5 Эксперименты

5.1 Поиск архитектуры

Функции активации применяются к весам модели на каждом шаге обучения в конкретной ячейке. Это и позволяет выбирать оптимальную архитектуру.

Описанный выше алгоритм помог нам найти следующую архитектуру на последней эпохе(Рис 1).

После получение параметров сети, было проведено повторное обучение для получение качества модели.

5.2 Сравнение найденной архитектуры с существующими

Разобьем нашу выборку на три части: обучающую (64% наблюдений), валидационную (16%) и тестовую (20%). Обучающую будем использовать для непосредственного обучения, валидационную для ранней остановки, а на тестовой будем замерять качество.

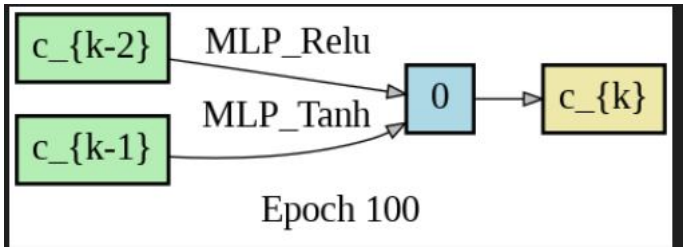

Рис. 1: California Housin

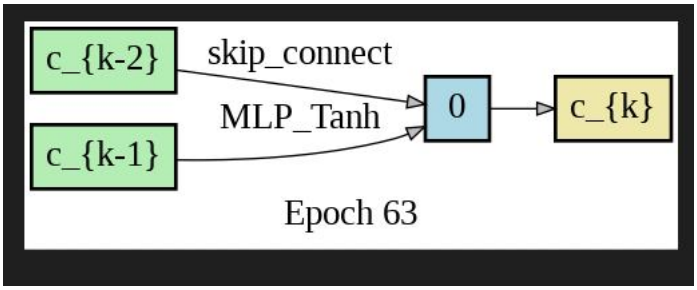

Рис. 2: Covertypes

Будем использовать два датасета: California Housing и Covertypes из sklearn. Выбор на эти датасеты пал в силу того, что в первом из них решается задача регрессии, а во втором задача классификации; в первом отсутствуют категориальные признаки, а во втором есть. Таким образом использование этих двух датасетов позволит посмотреть на работоспособность модели в сильно отличающихся задачах. Качество модели оценивалось по 10 запускам и RMSE в случае California Housing и 5 запускам и accuracy в случае Covertypes.

Будем использовать два бейзлайна: MLP и MLP-PLR. Если говорить конкретнее, то архитектура MLP выглядит так: Linear, ReLU, Dropout, Linear, ReLU, Dropout, Linear. Где скрытая размерность равна 384, вероятность в Dropout равна 0.4. В MLP-PLR для получения предсказания использовался тот же MLP для получения предсказаний, а размерность эмбеддинга для каждого признака равнялась 24. Обучение происходило с помощью оптимизатора Adam с lr=$3 \cdot 10^{-4}$. Результаты эксперимент представлены в табл. 1.

6 Выводы

Полученный лосс показал примерно такой же результат, что и baseline. Предположительно, для улучшения погрешности стоит более тщательно подобрать гиперпараметры градиентного спуска, а также пересмотреть пространство поиска.

7 Вклад участников

- Полюбин Арсений. Изучение архитектуры DARTS, Реализация алгоритма DARTS. придумывание пространства поиска, подготовка отчета
- Чуб Вячеслав. Изучение архитектуры DARTS, подготовка отчета.

Результаты экспериментов		
Модель	RMSE для California Housing	accuracy для Covertypes
MLP	0.466 ± 0.002	0.911 ± 0.001
MLP-PLR	0.413 ± 0.003	0.944 ± 0.001
Ours	$\mathbf{0.46 \pm 0.003}$	$\mathbf{0.918 \pm 0.01}$

Таблица 1: Результаты экспериментов.

- Феоктистов Дмитрий. Изучение области глубокого обучения на табличных данных, подготовка бейзлайнов и данных, придумывание пространства поиска, подготовка отчета.

Список литературы

Sercan Ö Arik and Tomas Pfister. Tabnet: Attentive interpretable tabular learning. In Proceedings of the AAAI conference on artificial intelligence, volume 35, pages 6679–6687, 2021.

Sarkhan Badirli, Xuanqing Liu, Zhengming Xing, Avradeep Bhowmik, Khoa Doan, and Sathiya S Keerthi. Gradient boosting neural networks: Grownet. arXiv preprint arXiv:2002.07971, 2020.

Yury Gorishniy, Ivan Rubachev, Nikolay Kartashev, Daniil Shlenskii, Akim Kotelnikov, and Artem Babenko. Tabr: Tabular deep learning meets nearest neighbors.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. Advances in Neural Information Processing Systems, 34:18932–18943, 2021.

Huifeng Guo, Bo Chen, Ruiming Tang, Weinan Zhang, Zhenguo Li, and Xiuqiang He. An embedding learning framework for numerical features in ctr prediction. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pages 2910–2918, 2021.

Yury Gorishniy, Ivan Rubachev, and Artem Babenko. On embeddings for numerical features in tabular deep learning. Advances in Neural Information Processing Systems, 35:24991–25004, 2022.

Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. arXiv preprint arXiv:1806.09055, 2018.

Xuanyi Dong and Yi Yang. Searching for a robust neural architecture in four gpu hours. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 1761–1770, 2019.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. arXiv preprint arXiv:2102.11535, 2021.