

Tabular DL

Подготовила: Парфенова Анна

Руководитель: Игнашин Игорь



Введение и предпосылки

- Классические GBDT модели превосходят исторически нейронные сети над табличными данными
- Для исследований DL это является последним рубежом
- В некоторых наборах данных и моделях остаются проблемы, включая шумные признаки, переобучение и трудности в обучении высокочастотных periodic embedding
- *Gorishniy, Y., Rubachev, I., & Babenko, A. (2023). On Embeddings for Numerical Features in Tabular Deep Learning. arXiv preprint arXiv:2203.05556*
- Такие методы как кусочно-линейное кодирование (piecewise linear encoding) и ансамблирование моделей показали преимущество нейронных сетей

Основная идея

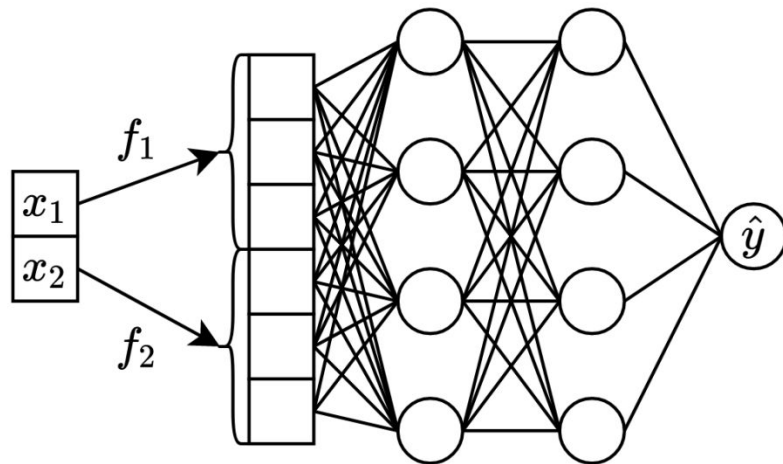
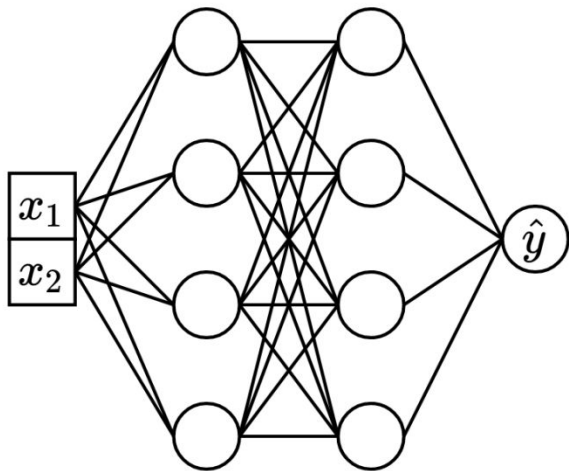
- Использование периодических эмбеддингов перед обучением MLP
- Вместо прямой подачи скалярных признаков — перевод их в векторы (эмбеддинги)
- Эти эмбеддинги затем обрабатываются стандартным backbone (MLP, Transformer и т.д.)
- Эмбеддинги можно строить различными способами: от простых линейных до периодических

- $\text{PLR} \mid \text{ReLU} \circ \text{Linear} \circ \text{Periodic}$

$$f_i(x) = \text{Periodic}(x) = \text{concat}[\sin(v), \cos(v)], \quad v = [2\pi c_1 x, \dots, 2\pi c_k x]$$

Первые эксперименты

Сравнение экспериментов на обычном MLP (первая картинка) и MLP-PLR (вторая картинка)



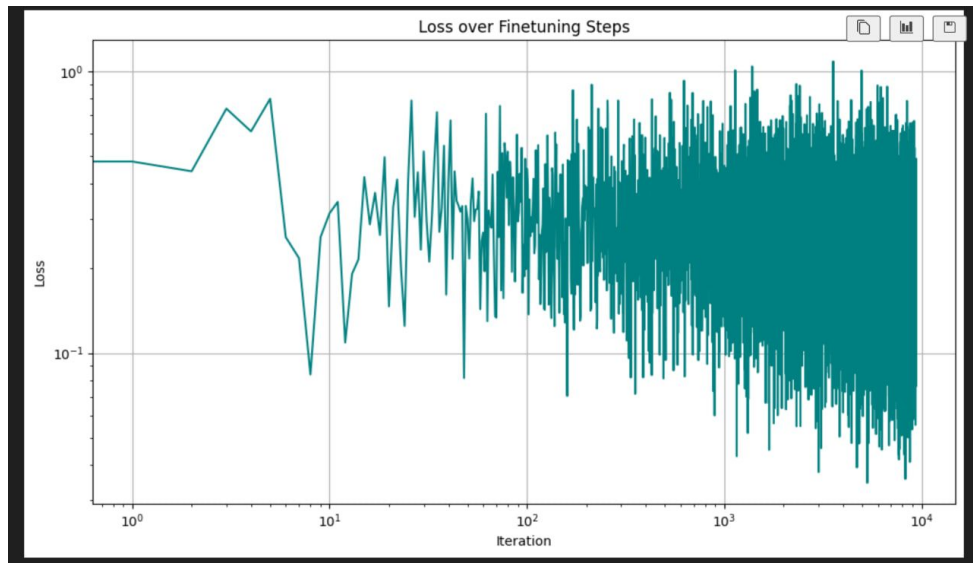
Какая стояла задача

Цель: улучшить бенчмарки, увеличить разрыв между GBDT-моделями и нейронными сетями над табличными данными

- за счет более обучаемых эмбеддингов
- применение разных датасетов (MNIST, CIFAR)
- использование разных моделей (MLP, Transformers)
- использование кастомных оптимайзеров
- регуляризация

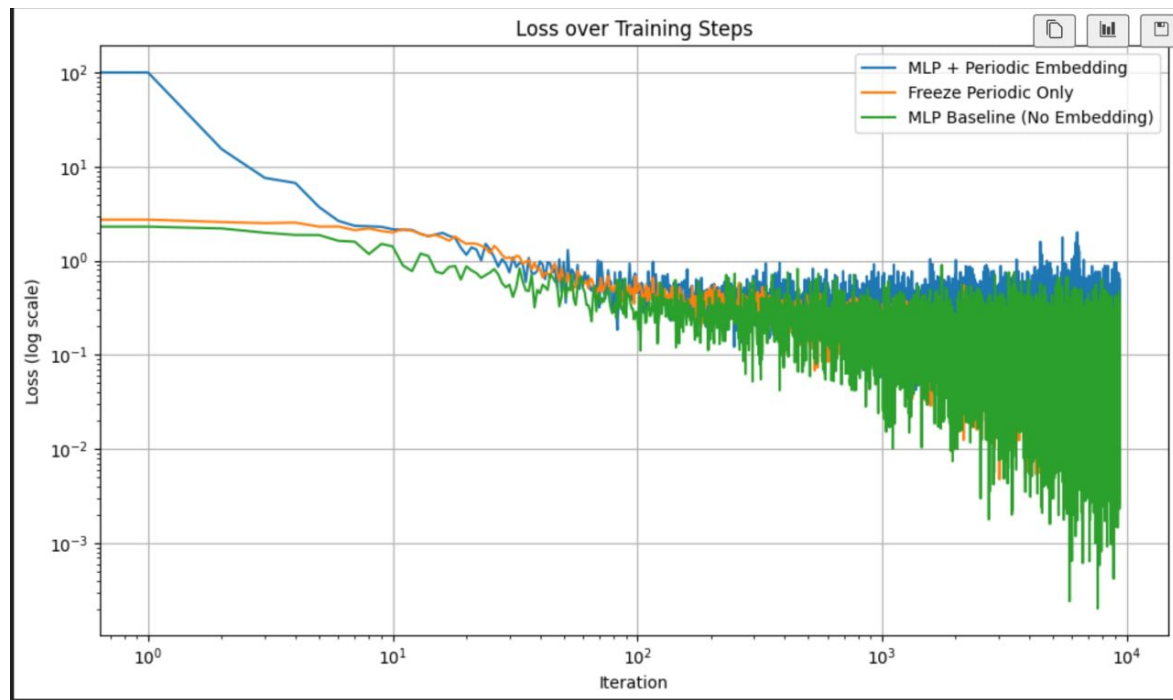
Первые результаты

- Обучение MLP-PLR на MNIST
- Дообучение только периодических весов на обученной модели



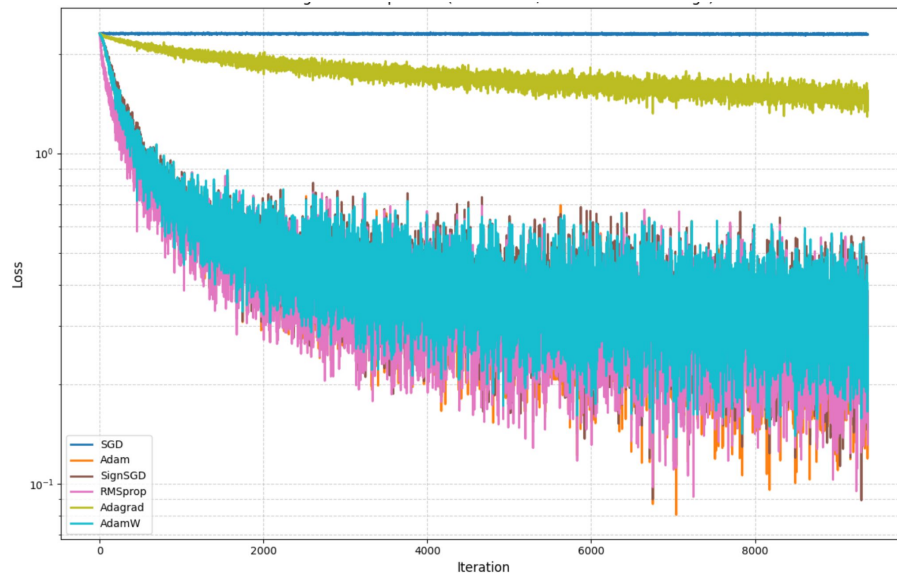
Обнаружение проблемы

- Запустили ту же модель и те же данные с замороженными весами, которые мы пытались оптимизировать
- Проблема: на выбранных данных работает плохо даже так



Возможное решение

- Во всех экспериментах ранее использовался только оптимизатор Adam, поэтому мы рассмотрели поведение MLP-PLR еще с оптимизаторами: SGD, Adam, RMSprop, AdaGrad, AdamW
- Все показали себя в итоге не очень хорошо, разве что Adam, RMSprop, AdamW показали себя чуть лучше остальных



Еще возможные решения в будущем

- Поменять датасет на CIFAR
- Попробовать с transformers
- Рассмотреть различные оптимайзеры другие
- Теоретически посмотреть, почему могут плохо учиться sin-cos