NanoTabPFN in ~10 minutes

Salih Bora Öztürk [1] Alexander Pfefferle [2,1] Frank Hutter [3,2,1]

[1]University of Freiburg [2]ELLIS Institute Tübingen [3]Prior Labs

TL;DR

modded-nanoTabPFN: This repository hosts the nanoTabPFN speedrun, in which we search for the fastest way to train a tabular foundation model that beats Random Forest on TabArena datasets.

Introduction

Motivation

- TabPFN have shown that pretraining on synthetic datasets can lead to strong performance... [1] [2]
- However, training these models takes long, and no one has time to wait...

Background

- GPT2 → nanoGPT → modded-nanoGPT (45 mins to 1.54 mins)
- TabPFNv2 → nanoTabPFN → modded-nanoTabPFN

Goal

- Pretrain a neural network to beat Random Forest (~0.8068 validation average ROC AUC) on subsampled TabArena datasets using 1 NVIDIA L40S.

This repo now contains a training algorithm which attains the target performance in:

- **9.26 minutes** on 1xL40S (baseline needed 74.32)
- **13184 synthetic datasets** (baseline needed 80576)

Rules

New records must:

1. Not modify the evaluation pipeline.
2. Not load any pretrained weights.
3. Run faster than prior record when baselined on the same hardware.

Other than that, anything and everything is fair game, including changes to the synthetic prior generation!

Timing protocol

- Counting only training wall-clock time (forward/backward/optimizer steps over synthetic batches).
- Evaluation runtime is excluded; the training timer is stopped while running validation.
- Prior generation time is excluded (using a pre-generated prior dump is allowed).

Evaluation details

Evaluation is on all 38 TabArena classification datasets:

- if >100 features, randomly select 100
- if >1000 rows, randomly select 1000 (stratified by class labels)
- 5-fold StratifiedKFold with shuffling, class labels are encoded with integers per fold
- average binary or one-vs-rest ROC AUC over all datasets

Improvement techniques

This improvement in training speed has been brought about by the following techniques:

- Muon optimizer [7]
- Batched Muon zeropower update for grouped QKV matrices
- Scaled Dot-Product Attention rewrite with explicit QKV [8]
- Pre-norm transformer blocks [9]
- Compile TransformerEncoderLayer forward [10]
- bfloat16 autocast in training and inference [11]
- Set float32 matmul precision to high [12]
- Increase learning rate from 10^{-4} to 10^{-3}
- Increase embedding size from 192 to 256
- Reduce attention heads from 6 to 4

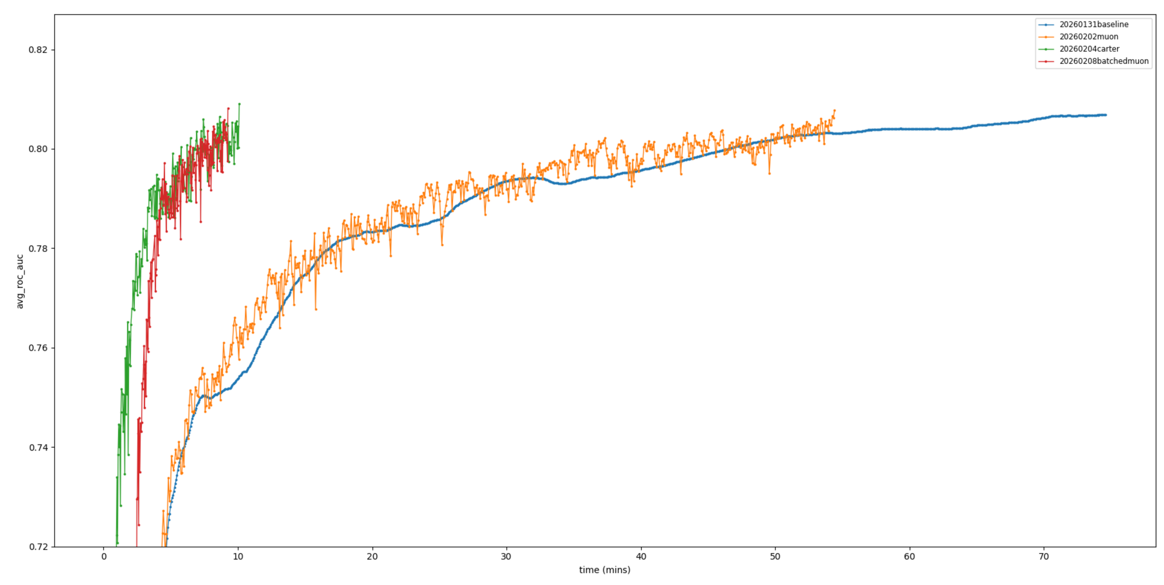
Work in progress:

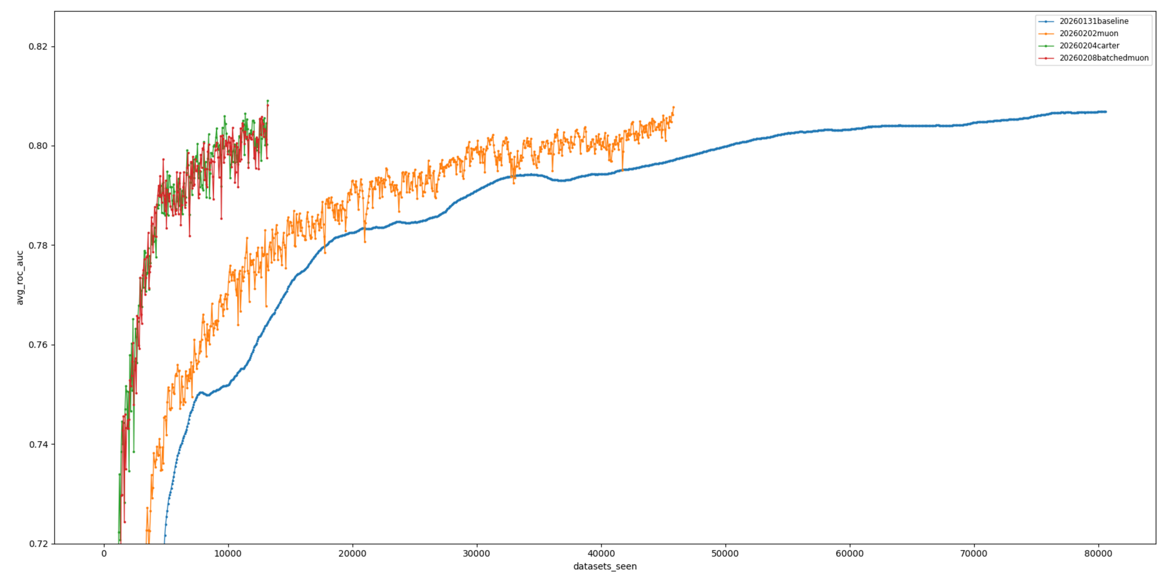
- GoLU activation function [13]
- RoRa [14]

The following techniques were evaluated but did not lead to improvements:

- Initialise linear layers with Xavier initialization [15]
- Disable all linear layer biases [9]

Record history





Record time	Description	Contributors
74.32 mins	Baseline	@borawhocodess, nano-tabpfn contributors
54.41 mins	Muon optimizer	@borawhocodess
10.10 mins	SDPA, bf16, higher LR, wider embeddings, fewer heads	@carterprince
9.26 mins	Batched Muon, compiled forward	@carterprince

Join the race!

Scan to join the speedrun on GitHub
github.com/borawhocodess/modded-nanotabpfn

References

[1] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

[2] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

[3] Samuel Müller, Noah Hollmann, Sebastian Pineda Arango, Josif Grabocka, and Frank Hutter. Transformers can do Bayesian inference. In *International Conference on Learning Representations (ICLR)*, 2022.

[4] Alexander Pfefferle, Johannes Hog, Lennart Purucker, and Frank Hutter. nanotabpfn: A lightweight and educational reimplementation of tabpfn, 2025.

[5] Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Mutalik Desai, David Salinas, and Frank Hutter. Tabarena: A living benchmark for machine learning on tabular data, 2025.

[6] Keller Jordan, Jeremy Bernstein, Brendan Rappazzo, @fernbear.bsky.social, Boza Vlado, You Jiacheng, Franz Cesista, Braden Koszarsky, and @Grad62304977. modded-nanogpt: Speedrunning the nanogpt baseline, 2024.

[7] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024.

[8] PyTorch Contributors. torch.nn.functional.scaled_dot_product_attention, 2026. Accessed: 2026-02-11.

[9] Jonas Geiping and Tom Goldstein. Cramming: Training a language model on a single gpu in one day, 2022.

[10] PyTorch Contributors. torch.compile, 2026. Accessed: 2026-02-11.

[11] PyTorch Contributors. Automatic mixed precision package - torch.amp, 2026. Accessed: 2026-02-11.

[12] PyTorch Contributors. torch.set_float32_matmul_precision, 2026. Accessed: 2026-02-11.

[13] Indrashis Das, Mahmoud Safari, Steven Adriaensen, and Frank Hutter. Gompertz linear units: Leveraging asymmetry for enhanced learning dynamics. *arXiv preprint arXiv:2502.03654*, 2025.

[14] ... Rora-tab: Geometric subspace selection for tabular learning via rotational rank adaptation. 2024.

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[16] PyTorch Contributors. Dealing with recompilations, 2026. Accessed: 2026-02-11.

ellis | INSTITUTE TÜBINGEN

AutoML.org

PRIOR