Summary: Mambular

April 14, 2024

Tabular deep learning has significantly progressed in recent years, slowly narrowing the performance gap with the traditionally dominant gradient-based decision trees [Gorishniy et al., 2021, Grinsztajn et al., 2022]. Although a small gap remains, transformer-based architectures, have consistently outperformed classical network structures across a variety of tabular problems [Arik and Pfister, 2021, Huang et al., 2020, Gorishniy et al., 2021, Hollmann et al., 2022, Grinsztajn et al., 2022, McElfresh et al., 2024]. Especially FT-Transformers [Gorishniy et al., 2021] have demonstrated robust results and can even outperform gradient-based decision trees in certain domains.

The introduction of the Mamba model [Gu and Dao, 2023], which has excelled in handling textual data, suggests promising applications in tabular deep learning. Ahamed and Cheng [2024] have already presented an approach by replacing traditional transformer blocks with Mamba blocks. However, they defer from the FT-Transformer architecture and treat numerical and categorical features identically, using ordinal encoding for numerical features, which could potentially result in information loss.

It is notable that most transformer-based deep learning architectures lack user-friendly Python libraries, which would make them as accessible as tools like XGBoost from scikit-learn [Pedregosa et al., 2011]. Neither Ahamed and Cheng [2024] nor Arik and Pfister [2021], nor Huang et al. [2020], offer easy-to-use, user-friendly packages. Furthermore, to the best of my knowledge, even the popular FT-Transformer is not available in a comprehensive Python package.

Mambular fills this void by offering a Python library that leverages the innovative Mamba architecture for deep learning tasks with tabular datasets [Gu and Dao, 2023]. Mambular adopts the principles from [Gorishniy et al., 2021], incorporating Mamba blocks instead of traditional transformer blocks. It also enhances model flexibility with adjustable embedding activation, pooling layers, and task-specific head architectures, thereby simplifying the implementation of the models as described in [Ahamed and Cheng, 2024]. Additionally, Mambular is designed to be user-centric; it integrates preprocessing within its architecture and provides a ready-to-use Python package that accommodates users with varying levels of technical expertise.

The default setup of Mambular mirrors the design of conventional tabular transformer models [Huang et al., 2020, Arik and Pfister, 2021, Gorishniy et al., 2021]. Numerical features that are integer-binned are processed as categorical and passed through an embedding layer. This would mirror the architecture presented by Ahamed and Cheng [2024]. Other numerical features, whether preprocessed or not, are handled through a single feed-forward dense layer with the same dimensionality as the embedding layers [Gorishniy et al., 2021]. Typically, no activation is applied to these embeddings, though users can adjust this setting. These embeddings are then processed through a series of Mamba layers and subsequently pooled, with average pooling as the default method. Alternatively, cls token embeddings can be used in place of pooling. Following pooling, RMS layer normalization from [Gu and Dao, 2023] is applied, culminating in a task-specific model head.

References

Md Atik Ahamed and Qiang Cheng. Mambatab: A simple yet effective approach for handling tabular data. *arXiv preprint arXiv:2401.08867*, 2024.

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.

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.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. Tabpfn: A transformer that solves small tabular classification problems in a second. *arXiv preprint arXiv:2207.01848*, 2022.

Xin Huang, Ashish Khetan, Milan Cvitkovic, and Zohar Karnin. Tabtransformer: Tabular data modeling using contextual embeddings. *arXiv preprint arXiv:2012.06678*, 2020.

Duncan McElfresh, Sujay Khandagale, Jonathan Valverde, Vishak Prasad C, Ganesh Ramakrishnan, Micah Goldblum, and Colin White. When do neural nets outperform boosted trees on tabular data? *Advances in Neural Information Processing Systems*, 36, 2024.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.