Expanded Convolutional Network for Tabular Data

Edson Luque Mamani [*1,2]

Abstract

Convolutional neural networks (CNNs) are widely recognized for their effectiveness in computer vision tasks, but their spatial information capturing ability does not directly apply to tabular datasets lacking spatial correlation. In this paper, a tailored approach called Expanded CNN (ExCNN) is proposed for tabular data analysis. Unlike common practices of transforming tabular data into images or using transformer architectures, ExCNN enhances feature dimensionality through a fully connected layer, harnessing the benefits of complex neural networks adapted to the tabular data domain. The performance of ExCNN is evaluated on various datasets, comparing it to existing architectures and benchmarking against Gradient Boosted Decision Trees. While no universally superior solution emerges, ExCNN demonstrates promise by leveraging the advantageous characteristics of CNNs for tabular data, outperforming certain deep learning architectures in specific metrics.

1. Motivation

In recent years, Convolutional Neural Networks (CNNs) have demonstrated great success in image analysis. However, non-image (tabular) data is prevalent in various fields, including bioinformatics (Bayat, 2002; Zhu et al., 2015), medicine (Topol, 2019; Rajkomar et al., 2018), and finance. While CNNs have shown excellent modelling capacity for image data, it is not always feasible to apply them directly to tabular data. This has led to the development of methods that transform tabular data into images to explicitly represent feature relationships (Sharma et al., 2019; Zhu et al., 2021; Bazgir et al., 2020; Ma & Zhang, 2018). However, such methods do not always preserve the spatial locality property, which is crucial for extracting meaningful feature representations.

The current state-of-the-art approach for tabular data problems centers around ensembles of decision trees, specifically Gradient Boosting Decision Tree (Friedman, 2001), which has consistently proven to be a popular choice in various machine learning competitions. Renowned GBDT libraries such as XGBoost, LightGBM, and CatBoost (Prokhorenkova et al., 2018) are widely adopted by machine learning researchers and practitioners. In recent years, there has been a notable surge in the development of deep learning models specifically designed for tabular data:

- (Arik & Pfister, 2021) with TabNet, that uses sequential attention and conventional feed-forward modules.
- (Badirli et al., 2020) with GrowNet, an architecture based on Gradient boosted and weaks MLPs
- (Klambauer et al., 2017) with SNN, an MLP-like architecture with the SELU activation.
- (Popov et al., 2019) with NODE, an ensemble of decision trees
- (Wang et al., 2021) with DCN, an MLP module and a combination of linear layers and multiplications.
- (Song et al., 2019) with AutoInt, transforms features to embeddings and applies attention-based transformations.
- (Prokhorenkova et al., 2018) with CatBoost, a GBDT implementation that uses oblivious decision trees

In fact, several studies have endeavored to categorize these models into three main groups: shallows, differentiable trees, and attention-based models (Gorishniy et al., 2021). Research such as the one cited above has served as inspiration for us to conduct a fair comparison of our model against the aforementioned existing architectures. These studies have demonstrated that a straightforward ResNet-like architecture can be an effective baseline for tabular data analysis.

We summarize the contributions of our paper as follows:

[1]Department of Engineering and Mathematics, University of La Salle, Arequipa, Perú [2]School of Computer Science, University of San Agustin, Arequipa, Peru. Correspondence to: Edson Luque <eluquem@ulasalle.edu.pe>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

- We propose a Expanded CNN network without transforming tabular data into images or generating complex embedding process.
- We thoroughly evaluate the main models for tabular Deep Learning models on a diverse set of tasks to investigate their relative performance.
- We reveal that our ExCNN model demonstrates promise by capitalizing on the advantageous characteristics of CNNs adapted for tabular data. In specific metrics, it outperforms certain families of deep learning architectures employed in tabular data methods

2. Proposal: Expanded Convolutional Network for Tabular Data

We can explore the possibility of utilizing a one-dimensional convolutional layer to apply a Convolutional Neural Network (CNN) to a tabular dataset. However, it is important to note that this layer assumes a spatial locality correlation between features, implying that adjacent columns are expected to exhibit spatial correlation. Unfortunately, this assumption does not hold true for most tabular datasets, as we typically lack prior knowledge about the spatial relationships between the columns. Therefore, directly feeding a collection of tabular data into a convolutional layer would not be appropriate, as the tabular entities lack spatial correlation. To address this, we propose incorporating an additional layer in our approach to reorder the tabular data, enabling its compatibility with the subsequent convolutional layer.

Introducing ExCNN network designed specifically for tabular data analysis. The network architecture begins by expanding the input size (number of rows) through a fully connected standard layer (depicted in green) see Figure 1. This layer then transforms into **'C'** channels, each containing **'S'-size** signals or **Sx1-size** images. To clarify, each signal represents a collection of **S** ordered characteristics, and we have **C** groups encompassing various combinations. It is worth noting that the values observed in the **Sx1** signals are not direct replicas of the original characteristics, but rather a result of nonlinear combinations. Subsequently, the features are extracted using a series of one-dimensional convolutional layers, bolstered by the incorporation of two skip connections to prevent performance degradation. Finally, the extracted features are utilized in predicting targets through a fully connected layer. Our model configuration boasts a setting of **C = 128** and **S = 8**, based on a random tuning process using optuna library, providing an optimal setup for effective tabular data analysis.

We consider supervised learning problems. Where $D = (x_i, y_i)_{i=1}^{n}$ denotes a dataset, where $x_i = (x_i^{num}, x_i^{cat}) \in \mathbb{X}$ represents numerical x_i^{num} and categorical x_i^{cat} features and $y_i \in \mathbb{Y}$ denotes the corresponding label. We consider three types of tasks: binary classification $\mathbb{Y} = \{0, 1\}$, multiclass classification $\mathbb{Y} = \{1, ..., m\}$ and regression $\mathbb{Y} = \mathbb{R}$.

Table 1. Dataset properties.

Dataset	Num. features	Classes	Metric	Abbreviation
California Housing (Kelley Pace & Barry, 1997)	8	-	RMSE	CA
Adult (Kohavi, 1996)	6	2	Accuracy	AD
Jannis (Guyon et al., 2019)	54	4	Accuracy	JA
Higgs (Baldi et al., 2014)	28	2	Accuracy	HI
Covertipe (Blackard & Dean, 1999)	54	7	Accuracy	CO
Year (McFee et al., 2012)	90	-	RMSE	YE
Yahoo (Chapelle & Chang, 2011)	699	-	RMSE	YA
Microsoft (Qin & Liu, 2013)	136	-	RMSE	MI
Kaggle's Hospital Israelita Albert Einstein (Albert, 2020)	111	2	Accuracy	COV

Finally we formalize the "ExCNN" architecture in Equation 1

$$
\begin{aligned}
Row &= (x_i^{num}, x_i^{cat}) \in \mathbb{X}_n^d \\
T &= expand\left[x_1^{num}, ..., x_3^{cat}, ..., x_n^{cat}\right] \in \mathbb{X}_n^{d\times S} \\
R &= reshape\left[T\right] \in \mathbb{X}_n^{S\times 1\times C} \\
ExCNN(R) &= Residuals + \\
&\quad Pooling(Conv(Pooling(Conv(...(R)))))
\end{aligned} \tag{1}
$$

3. Experiments

3.1. Datasets

We leverage a collection of nine distinct public datasets, carefully chosen for our experimentation. Each dataset undergoes a singular train-validation-test split, ensuring consistency across all algorithms employed. To provide an overview of these datasets, we present a summarized depiction in the accompanying Table 1.

3.2. Implementation

Data preprocessing plays a crucial role in the effectiveness of deep learning models. To ensure a fair comparison among all deep models, we applied the same preprocessing techniques to each dataset. For regression targets, we standardized the data across all algorithms. Categorical features were handled differently depending on the algorithm used. For CatBoost made use of its built-in support for categorical features. In the case of our model and other neural

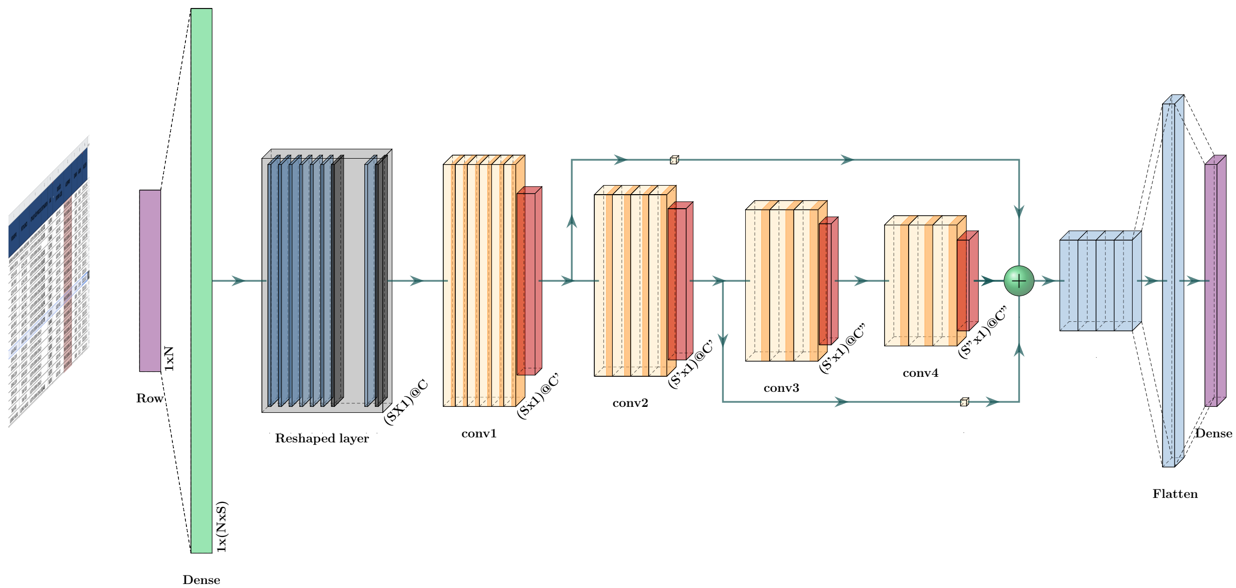

Figure 1. ExCNN network for tabular data. First, the input size (rows) is increased through a fully connected standard layer (green). This layer is then resized into ***C*** channels containing ***S***-size signals (or ***Sx1***-size images).

networks, we utilized embeddings of consistent dimensionality for all categorical features. Hyperparameter tuning is an essential step in optimizing model performance. For each dataset, we conducted a thorough tuning process for every model's hyperparameters. To achieve this, we utilized the Optuna library (Akiba et al., 2019), which employs Bayesian optimization techniques. Additionally, we iterated over predefined sets of configurations recommended by corresponding papers to ensure comprehensive exploration of the hyperparameter space.

3.2.1. Neural networks

To ensure a comprehensive evaluation of our model across different datasets, we employ specific optimization techniques tailored to the nature of the problem at hand. For classification tasks, we minimize cross-entropy, while for regression tasks, we utilize mean squared error as the objective function. In line with the original implementations of TabNet and GrowNet, we adopt the Adam optimizer proposed by (Kingma & Ba, 2014). Conversely, for all other algorithms, we utilize the AdamW optimizer introduced by (Loshchilov & Hutter, 2017), without incorporating any learning rate schedules.

Regarding batch sizes, we adhere to a predefined value for each dataset, unless specific instructions are provided in the corresponding papers. The training process continues until there have been no improvements on the validation set for patience + 1 consecutive epochs. For all algorithms, we have set the patience value to 10, ensuring a consistent approach across the board.

3.3. Evaluation

To obtain reliable results, we conduct a evaluation process for each tuned configuration. This involves running a total of 12 experiments using different random seeds. By performing multiple iterations, we mitigate the impact of random initialization and provide a comprehensive assessment of our model's performance on the test set. Table 2 shows the results for all architectures including our model.

4. Conclusion

From the results presented in Table 2, it is evident that ResNet proves to be a highly effective baseline. However, our ExCNN model consistently outperforms other models on numerous tasks, establishing itself as a powerful solution in the field of tabular data analysis.

Notably, NODE and SNN demonstrate high performance on multiple tasks as well. However, it is worth mentioning that these models, although more complex than ResNet, share similarities with it. Furthermore, they are not truly ”single” models, often comprising a significantly larger number of parameters compared to ResNet and ExCNN, and exhibiting an ensemble-like structure.

This study serves as a validation of the predictive capabilities of ExCNN, a novel model that leverages the unique characteristics of complex neural networks, such as CNNs adapted for tabular data. Our approach surpasses other deep

Table 2. Results for models, for each dataset, top results are in bold. The metric values averaged over 12 random seeds are reported.

Metric	Dataset	DL models								
		MLP	ResNet	GrowNet	DCN2	AutoInt	SNN	NODE	CatBoost	ExCNN
Accuracy	AD	0.843	0.856	0.847	0.861	0.859	0.854	0.857	**0.873**	0.868
	JA	0.701	0.728	-	0.718	0.721	0.709	0.727	0.724	**0.732**
	HI	0.713	**0.732**	0.709	0.721	0.725	0.722	0.716	0.728	0.698
	CO	0.962	0.954	-	0.966	0.934	0.961	0.958	0.910	**0.970**
	COV	0.853	0.841	0.827	0.801	0.793	0.787	0.843	0.868	**0.873**
Rmse	CA	0.476	**0.493**	0.455	0.484	0.474	**0.493**	0.464	0.428	0.478
	YE	8.853	8.831	8.827	8.890	8.882	8.889	**8.931**	8.885	8.901
	YA	0.757	0.742	0.765	0.765	0.768	0.768	**0.773**	0.749	0.752
	MI	0.747	0.728	0.751	0.749	0.750	**0.762**	0.745	0.744	0.760

learning solutions across a range of tasks, eliminating the need for intricate feature embedding processes commonly employed in deep learning models.

References

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2623–2631, 2019.

Albert, H. I. Diagnosis of covid-19 and its clinical spectrum, 2020. URL https://www.kaggle.com/datasets/einsteindata4u/covid19. Accessed on April 10 , 2023.

Arik, S. Ö. and Pfister, T. Tabnet: Attentive interpretable tabular learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 6679–6687, 2021.

Badirli, S., Liu, X., Xing, Z., Bhowmik, A., Doan, K., and Keerthi, S. S. Gradient boosting neural networks: Grownet. *arXiv preprint arXiv:2002.07971*, 2020.

Baldi, P., Sadowski, P., and Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nature communications*, 5(1):4308, 2014.

Bayat, A. Science, medicine, and the future: Bioinformatics. *BMJ: British Medical Journal*, 324(7344):1018, 2002.

Bazgir, O., Zhang, R., Dhruba, S. R., Rahman, R., Ghosh, S., and Pal, R. Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks. *Nature communications*, 11(1):1–13, 2020.

Blackard, J. A. and Dean, D. J. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and electronics in agriculture*, 24(3):131–151, 1999.

Chapelle, O. and Chang, Y. Yahoo! learning to rank challenge overview. In *Proceedings of the learning to rank challenge*, pp. 1–24. PMLR, 2011.

Friedman, J. H. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp. 1189–1232, 2001.

Gorishniy, Y., Rubachev, I., Khrulkov, V., and Babenko, A. Revisiting deep learning models for tabular data. *Advances in Neural Information Processing Systems*, 34: 18932–18943, 2021.

Guyon, I., Sun-Hosoya, L., Boullé, M., Escalante, H. J., Escalera, S., Liu, Z., Jajetic, D., Ray, B., Saeed, M., Sebag, M., et al. Analysis of the automl challenge series. *Automated Machine Learning*, 177, 2019.

Kelley Pace, R. and Barry, R. Sparse spatial autoregressions. *Statistics Probability Letters*, 33(3):291–297, 1997. ISSN 0167-7152. doi: https://doi.org/10.1016/S0167-7152(96)00140-X. URL https://www.sciencedirect.com/science/article/pii/S016771529600140X.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S. Self-normalizing neural networks. *Advances in neural information processing systems*, 30, 2017.

Kohavi, R. Scaling up the accuracy of naive-bayes classifiers: a. In *Proceedings of the Second International Conference on*, 1996.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ma, S. and Zhang, Z. Omicsmapnet: Transforming omics data to take advantage of deep convolutional neural network for discovery. *arXiv preprint arXiv:1804.05283*, 2018.

McFee, B., Bertin-Mahieux, T., Ellis, D. P., and Lanckriet, G. R. The million song dataset challenge. In *Proceedings of the 21st International Conference on World Wide Web*, pp. 909–916, 2012.

Popov, S., Morozov, S., and Babenko, A. Neural oblivious decision ensembles for deep learning on tabular data. *arXiv preprint arXiv:1909.06312*, 2019.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. Catboost: unbiased boosting with categorical features. *Advances in neural information processing systems*, 31, 2018.

Qin, T. and Liu, T.-Y. Introducing letor 4.0 datasets. *arXiv preprint arXiv:1306.2597*, 2013.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., et al. Scalable and accurate deep learning with electronic health records. *NPJ Digital Medicine*, 1(1):1–10, 2018.

Sharma, A., Vans, E., Shigemizu, D., Boroevich, K. A., and Tsunoda, T. Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture. *Scientific reports*, 9(1):1–7, 2019.

Song, W., Shi, C., Xiao, Z., Duan, Z., Xu, Y., Zhang, M., and Tang, J. Autoint: Automatic feature interaction learning via self-attentive neural networks. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pp. 1161–1170, 2019.

Topol, E. J. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25 (1):44–56, 2019.

Wang, R., Shivanna, R., Cheng, D., Jain, S., Lin, D., Hong, L., and Chi, E. Dcn v2: Improved deep & cross network and practical lessons for web-scale learning to rank systems. In *Proceedings of the web conference 2021*, pp. 1785–1797, 2021.

Zhu, Y., Xu, Y., Helseth Jr, D. L., Gulukota, K., Yang, S., Pesce, L. L., Mitra, R., Müller, P., Sengupta, S., Guo, W., et al. Zodiac: a comprehensive depiction of genetic interactions in cancer by integrating tcga data. *Journal of the National Cancer Institute*, 107(8):djv129, 2015.

Zhu, Y., Brettin, T., Xia, F., Partin, A., Shukla, M., Yoo, H., Evrard, Y. A., Doroshow, J. H., and Stevens, R. L. Converting tabular data into images for deep learning with convolutional neural networks. *Scientific reports*, 11 (1):1–11, 2021.