TL;DR: Tabular LoRA with Diffusion Models and BERT Embeddings

Minseo Yoon
Department of Data Science, 2021320322
DATA301: The Design and Analysis of Experiments, May 16, 2024
cooki0615@korea.ac.kr

Abstract

As large-scale models become more complex with dramatically increasing parameter counts, they face the significant challenge of inadequate training data. This highlights the urgent need for proficient data generation techniques. In response, our research leverages the capabilities of diffusion models and language models to generate tabular data. To enhance tuning efficiency, we implement Low-Rank Adaptation (LoRA). Initial findings suggest that the baseline approach can robustly synthesize realistic tabular data and ours was successfully reimplemented. We aim to test this across up to 15 datasets to thoroughly validate our methodology. The potential impact of our work is substantial, providing a scalable way to augment training datasets and significantly enhance the performance of machine learning models in data-scarce environments.

1. Introduction

In the era of big data, we navigate complex data types, from structured numerical databases to unstructured text and images. This diversity offers immense opportunities and challenges. For instance, Large Language Models (LLMs) like GPT-4 [21] have significantly increased model complexity. The upcoming GPT-5 exacerbates this by requiring massive training data due to its vast size, highlighting the paradox of abundant computational resources paired with data scarcity [18].

Historically, techniques such as data augmentation [10–12, 20, 22, 30] and contrastive learning [2, 4, 5, 13, 14] have enhanced dataset utility. The rise of generative AI, particularly diffusion models for images [1, 15, 25, 26] and LLMs for text [3, 6, 7, 9, 24, 27, 28], has opened new pathways for data generation. These models create robust datasets that can train other AI systems. Utilizing well-trained embeddings from these models allows for novel methods to synthesize tabular data, addressing data scarcity.

As model sizes grow, efficient fine-tuning becomes crucial, and LoRA [16] offers a solution by allowing significant updates without the need for extensive retraining. Our research introduces an approach leveraging diffusion models and advanced embeddings from pre-trained language models like BERT [9] to generate enriched tabular data. This aims to enhance the scalability and effectiveness of large-scale AI systems.

2. Methodology

2.1. Preliminaries

DDPM (Denoising Diffusion Probabilistic Models) [15] are generative models that transform a noise distribution into a data distribution over a sequence of iterative steps. The model's operation can be mathematically represented as follows:

$$x_t = \sqrt{\alpha_t} x_{t-1} + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \tag{1}$$

where x_t represents the data at step t, α_t are the variance schedules, and ϵ is Gaussian noise. The reverse process, crucial for generating data, involves learning to denoise:

$$\hat{x}_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t) \right) \tag{2}$$

Here, $\epsilon_\theta(x_t, t)$ is the neural network predicting noise, which is trained to minimize the difference from the actual noise used during the forward process.

BERT (Bidirectional Encoder Representations from Transformers) [9] leverages the Transformer, a deep learning model wherein attention mechanisms selectively focus on different parts of the input data. BERT's architecture allows it to consider the full context of a word by looking at the words that come before and after it in a sentence. This is captured by the Transformer's (multi-headed) attention function [29]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{3}$$

where Q, K, and V represent the query, key, and value components of the input data, respectively, and d_k is the dimensionality of the keys. BERT's pre-training involves two main tasks: masked language modeling and next sentence prediction, which allow it to develop a deep understanding of language context and structure.

LoRA (Low-Rank Adaptation) [16] introduces a low-rank approximation of updating parameters to pre-trained models, allowing efficient training and fine-tuning by updating only a small subset of model parameters. In the context of transformers, LoRA modifies the attention and feed-forward layers as follows:

$$W \leftarrow W + \Delta W = W + AB \tag{4}$$

where $W \in \mathbb{R}^{d\times d}$ is the original weight matrix, $\Delta W = AB$ represents the perturbation to the attention and feed-forward weights. The rank of matrices $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times d}$ are smaller ($r \ll d$), making the updates more parameter-efficient. LoRA achieves significant improvements in performance with minimal changes to the model architecture, optimizing for both computational efficiency and model effectiveness.

2.2. TL;DR: Tabular LoRA with Diffusion Models and BERT Embeddings

2.2.1 Baseline Model: TabDDPM

TabDDPM (the simplest design of **DDPM** for **tab**ular problems) [17] serves as the starting point for our research. Developed as an adaptation of standard DDPM, TabDDPM has been specifically optimized for handling tabular data. It employs a series of transformations that progressively denoise a data distribution, making it particularly effective for generating structured data that closely mimics real-world statistical properties.

2.2.2 Leveraging the Well-Trained Embedding

In the baseline model, a simple MLP architecture is used to model the reverse process in Equation 2.

$$\begin{aligned}\texttt{MLP}(x) &= \texttt{Linear}(\texttt{MLPBlock}(\ldots(\texttt{MLPBlock}(x))))\\ \texttt{MLPBlock}(x) &= \texttt{Dropout}(\texttt{ReLU}(\texttt{Linear}(x)))\end{aligned} \tag{5}$$

A tabular input x_{in}, a timestep t, and a class label y are processed as follows.

$$\begin{aligned}t_{\text{emb}} &= \texttt{Linear}(\texttt{SiLU}(\texttt{Linear}(\texttt{SinTimeEmb}(t))))\\ y_{\text{emb}} &= \texttt{Embedding}(y)\\ x &= \texttt{Linear}(x_{\text{in}}) + t_{\text{emb}} + y_{\text{emb}}\end{aligned} \tag{6}$$

where `SinTimeEmb` refers a sinusoidal time embedding.

The MLP architecture, though widely used, struggles with contextual awareness and modeling complex data dependencies. To address these shortcomings, our method utilizes the BERT structure. We leverage pre-trained embeddings from established models, demonstrating that the language model's structure and embeddings can be effectively utilized as well-trained embeddings beyond NLP tasks, similar to the approach used in BLIP-2's Q-Former [19].

$$\begin{aligned}\texttt{LanguageModel}(x) &= \texttt{Linear}(\texttt{TransformerBlock}(\\ &\quad\ldots(\texttt{TransformerBlock}(x))))\\ \texttt{TransformerBlock}(x) &= \texttt{LayerNorm}(\texttt{Add}(\\ &\quad\texttt{SelfAttention}(\texttt{Linear}(x)), x))\end{aligned} \tag{7}$$

2.2.3 Beyond the Hyperparameter Tuning and Gaussian Assumption

Due to the extensive number of parameters in language models, fine-tuning all weights is impractical. LoRA addresses this by focusing updates on small parameters while preserving the original parameters, reducing computational demands. In our model, the Transformers model the reverse process, capturing complex dependencies in tabular data more effectively. To boost efficiency, the attention weights are dynamically updated through LoRA, optimizing both performance and resource use. For example, the subset of attention weights $W \in \{W^Q, W^K, W^V, W^O\}$ are updated as follows to incorporate the benefits of LoRA in Equation 4 while r is not chosen by only tuning as the hyperparameter but by:

$$\begin{aligned}\text{SVD}_X &= U\Sigma V^T\\ \text{rank}(\Delta W) &= \min_r \left\{ r \,\middle|\, \frac{\sum_{i=1}^{r}\Sigma_{ii}}{\text{tr}(\Sigma)} \geq \alpha \right\}\end{aligned} \tag{8}$$

where $X \in \mathbb{R}^{n\times d}$ is the design matrix of dataset and $\alpha \in [0, 1]$ is possibly close to 1.

Furthermore, We advance diffusion by adjusting $\mathcal{N}(0, I)$ of the diffusion process in Equation 1 based on data statistics. This combination of techniques enhances the model's ability to generate relevant tabular data for practical applications.

2.3. Datasets

We evaluate the proposed model using a broad collection of at most 15 real-world public datasets. Most of the data can be obtained from the official code of the baseline model. These datasets are diverse in terms of size, type, feature count, and distribution patterns, and their related information can be found in Table 1. Examples of such datasets include the Abalone and California Housing datasets. This

Table 1. Summary of Datasets by Task Type. Range and average for # Num and # Cat denote the range and average number of numerical and categorical variables, respectively, for each task type.

Task Type	# Datasets	Range # Num	Range # Cat	Average # Num	Average # Cat
Regression	7	3-50	0-5	16.4	1.6
Binclass	8	5-50	0-8	15.9	2.0
Multiclass	2	4-32	0-5	18.0	2.5

variety ensures a comprehensive assessment across different data contexts, providing a robust test of our model's capabilities.

In order to perform exploratory data analysis, we plan to actively utilize the visualization techniques performed in assignments 1 and 2. If missing values exist, the imputation is planned to be performed through linear regression using the variable containing these missing values as a response variable and other variables as explanatory variables.

The divergence such as absolute difference or Frobenius norm between correlation matrices computed on real and synthetic datasets will be used as a main metric. We will also compute the ML efficiency of the state-of-the-art model, as of the publication date of the baseline paper. Specifically, we consider CatBoost [23] for evaluation.

3. Initial Findings

We have successfully reimplemented the official code. However, hyperparameter tuning on Colab is time-intensive, taking over five hours per dataset. Given these constraints, we plan to use the optimal hyperparameters provided in the official code. Although transitioning from an MLP to a BERT-based model might require adjustments to these parameters, time limitations within this project's scope have led us to proceed with the existing settings.

As part of our validation efforts, we compared our reimplementation's performance on the Churn Modelling dataset against the original paper's reported accuracy of 0.755; ours was 0.744. We conducted a Wilcoxon signed-rank test, which yielded a p-value of 0.827. This indicates no statistically significant difference in performance, suggesting that our reimplementation successfully replicates the original findings.

4. Challenges and Solutions

Advanced generative models like diffusion and language models present substantial challenges for individual researchers due to high computational demands and hardware requirements.

Diffusion models are resource-intensive, as their iterative generation process lengthens inference times, hindering rapid experimentation. Efficient variants such as DDIM (Denoising Diffusion Implicit Models) [26] reduce the number of timesteps without significantly affecting output quality, speeding up the generation process.

Language models often exceed the memory capacities of modest research setups due to their extensive parameters. Quantization reduces weight precision to 4-bit or 8-bit, lowering memory usage. When combined with LoRA through QLoRA [8], this approach enables efficient fine-tuning while maintaining manageable model size and reasonable performance.

5. Next Steps

With the midterm report due, our project will continue through Weeks 12 to 16 with focused activities each week. In Week 12, we aim to innovate by redesigning formulas for applying the LoRA [16] to tabular data. This will involve translating these novel formulations into code and initiating fine-tuning processes. Subsequently, in Week 13, our focus will shift to enhancing the diffusion model by exploring the statistical measures of data, moving beyond $\mathcal{N}(0, I)$. This will include formulating new mathematical expressions and incorporating these modifications into our model. During Week 14, we will conduct experiments to construct a comprehensive performance table, evaluating our model under different conditions to ensure robustness and efficacy. Finally, Weeks 15 and 16 will be dedicated to analyzing the results and drafting the final report, synthesizing insights from the experiments to formulate conclusions that highlight both the successes and limitations of our research.

6. Conclusion

In this project, our objective is to generate tabular data using diffusion models [15] and language models [9]. By employing LoRA [16], we aim to achieve a parameter-efficient methodology. We plan to validate our approach across various datasets to ensure robust performance. Following the midterm project, we intend to explore novel applications and enhancements in LoRA and diffusion priors. Ultimately, we expect that our work offers a scalable method to augment datasets and enhance machine learning performance in data-scarce environments.

References

[1] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In *NeurIPS*, 2021. 1

[2] Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 1

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1

[5] Xinlei Chen and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2103.07579*, 2021. 1

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. 1

[7] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *JMLR*, 2024. 1

[8] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In *NeurIPS*, 2023. 3

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1, 3

[10] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 1

[11] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NeurIPS*, 2014. 1

[12] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 1

[13] Kaiming He, Xinlei Chen, Saining Xie, Yang Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 1

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 1, 3

[16] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 2, 3

[17] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *ICML*, 2023. 2

[18] Seohee Lee. "gpt-5" *Gongburyang 8baena neuneunde... deo hakseupsikil deiteoga eopda.* , Apr 2024. 1

[19] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*. PMLR, 2023. 2

[20] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1

[21] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[22] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 1

[23] Liudmila Ostroumova Prokhorenkova, Gleb Gusev, Aleksandr Vorobev, Anna Veronika Dorogush, and Andrey Gulin. Catboost: unbiased boosting with categorical features. In *NeurIPS*, 2018. 3

[24] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. In *OpenAI Blog*, 2019. 1

[25] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1

[26] Yang Song and Stefano Ermon. Denoising diffusion implicit models. In *CVPR*, 2021. 1, 3

[27] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[28] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 1

[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[30] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 1