Effective Data Distillation for Tabular Datasets (Student Abstract)

Inwon Kang[1], **Parikshit Ram**[2], **Yi Zhou**[2], **Horst Samulowitz**[2], **Oshani Seneviratne**[1]

[1]Rensselaer Polytechnic Institute, Troy, New York, USA,
[2]IBM Research, Yorktown Heights, New York, USA
kangi@rpi.edu, Parikshit.Ram@ibm.com, Yi.Zhou@ibm.com, samulowitz@us.ibm.com, senevo@rpi.edu

Abstract

Data distillation is a technique of reducing a large dataset into a smaller dataset. The smaller dataset can then be used to train a model which can perform comparably to a model trained on the full dataset. Past works have examined this approach for image datasets, focusing on neural networks as target models. However, tabular datasets pose new challenges not seen in images. A sample in tabular dataset is a one dimensional vector unlike the two (or three) dimensional pixel grid of images, and Non-NN models such as XGBoost can often outperform neural network (NN) based models. Our contribution in this work is two-fold: 1) We show in our work that data distillation methods from images do not translate directly to tabular data; 2) We propose a new distillation method that consistently outperforms the baseline for multiple different models, including non-NN models such as XGBoost.

Introduction

With the ever growing availability of data and size of machine learning (ML) models, scalability is an important issue in the ML lifecycle. One way to tackle this issue is with dataset distillation. Dataset distillation refers to the technique of reducing a large dataset into a smaller representative dataset, which the target model can be trained on. It can lead to significant reduction in training time for various optimization tasks. It can also help enable data privacy by providing only the distilled version of the dataset and not the full original data.

This topic has been actively explored in terms of image datasets (Wang et al. 2020; Nguyen, Chen, and Lee 2021) and neural network (NN) based models. But it has not been as thoroughly explored in the space of tabular datasets. ML in tabular datasets can also suffer from the same scalability issues faced in image datasets. In addition, tree-based models such as XGBoost often outperform neural network based models in tabular data. Medvedev and D'yakonov (2020) discuss the properties of dataset distillation in tabular data in by exploring the method proposed by Wang et al. (2020). However, this work only considers a synthetic dataset with a NN which does not address the full scope of the problem. In this work, we propose a new distillation pipeline for tabular data and demonstrate its performance against a previous state of art approach (Nguyen, Chen, and Lee 2021).

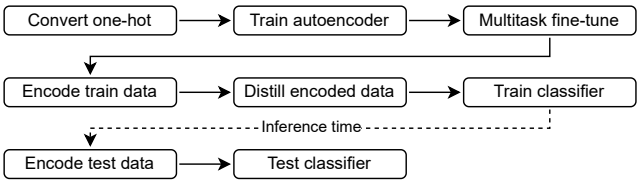

Figure 1: Overview of the final distillation method

Implementation

Dataset Our distillation pipeline takes a tabular dataset $X = \{x_1, ..., x_n\} | x_i \in R^d$ and its corresponding labels $Y = \{y_1, ..., y_n\} | y_i \in R$ as input. The goal is to generate some dataset $\hat{X} = \{\hat{x}_1, ..., \hat{x}_k\}$ and its labels $\hat{Y} = \{\hat{y}_1, ..., \hat{y}_k\}$ with $k < n$ which an arbitrary classifier can be trained on to output non-trivial performance on the original evaluation – i.e., perform better than random sampling. The data is then converted into a one-hot format to handle the categorical features. The continuous features are also binned to a one-hot format with 10 bins of identical widths. We only consider datasets with binary labels in our experiments. However, the distillation model is capable of handling multi-class datasets as well.

Autoencoder The distillation scheme we propose uses an autoencoder to project the dataset into a latent space in order to handle the sparsity of the one-hot encodings, in which we test different distillation methods. We consider two architectures of autoencoders in our work. The first version is a linear autoencoder composed of a multilayer perceptron (MLP) like structure, with ReLU activation after each hidden linear layer. The second version is a graph neural network (GNN) based autoencoder using the row-column bipartite graph formulation proposed in Wu, Yang, and Yan (2021).

Multihead Autoencoder We also consider a multi-head variant of each autoencoder architecture. To achieve this, we fine-tune the fully trained autoencoder with an MLP classifier head attached to the encoder for multi-task learning. The task for the fine-tuning step is to minimize the combined loss $L_r + \lambda L_c$, where L_c is the classification loss and λ is a weighting factor to balance the two tasks. While this

	XGB	LR	MLP	NNC	NBC
Base	0.8159	0.7736	0.7802	0.7395	0.6528
RND	0.6917	0.6894	0.6838	0.6480	0.6506
KIP+Base	0.6175	0.6643	0.6946	0.5700	0.6081
KM+Base	0.5698	**0.7285**	0.7057	0.7074	0.6658
KIP+MLP*	0.6886	0.7098	0.7186	0.6770	0.7006
KIP+GNN*	0.6858	0.7039	0.7116	0.6812	0.6954
KM+MLP*	**0.7072**	0.7267	**0.7279**	**0.7265**	**0.7195**
KM+GNN*	0.7038	0.7156	0.7241	0.7227	0.7125

Table 1: Average test metric (across 26 datasets) for model trained on a distilled data of size $N = 100$ across various distillation pipelines. Refer to text for additional details. The best performance among distillation schemes is in bold.

step may degrade the reconstruction performance of the decoder network, we find that the label information is effectively captured in the latent space by the encoder network.

Distillation Methods We consider a distance-based clustering method (KMeans or KM) and hierarchical clustering method (Agglomerative) as a naive baseline. To compare against the state of art in image data distillation, we consider kernel induced points (KIP) (Nguyen, Chen, and Lee 2021). We consider random sampling (RND) as our baseline. Each distillation algorithm is tested in the original one-hot space and the latent space of the four autoencoders. When the distillation is performed in the latent space, we also consider a version where the classifier is directly trained in the latent space or in the decoded space. For the target distillation size N for a dataset with l unique labels, the clustering algorithms identify N/l clusters for each label group and uses each cluster's euclidean centroid as the distilled output. Similarly, KIP randomly picks N/l points per each label group and optimizes them using the neural tangent kernel of an arbitrary network. Any distillation algorithm which includes randomness was repeated five times with different seeds and the average was taken as the final performance.

Optimization Due to the heterogenity of the datasets, we perform hyperparameter tuning for the 4 autoencoder architectures discussed above. The base autoencoder is trained for maximum of 200 epochs, while the multi-head fine-tuning is performed for maximum of 100 epochs. For all MLP components, the number of hidden layers and dropout rates are tuned. For the GNN encoder, we consider SAGE, GAT and GCN and also the number of stacked layers. The dimension of the latent space for every autoencoder was set to 16.

Results

We experiment with 26 different tabular datasets ranging from 10,000 to 110,000 rows and 7 to 80 features. Due to the high label imbalance in many datasets, we use balanced accuracy as the target metric. We first train the autoencoders for each dataset and apply the different combinations of distillation methods with the target distillation size N set to $20 \leq N \leq 200$ in steps of 20. Since many classical models perform well on tabular datasets, we consider four classical machine learning models – XGBoost (XGB), Logistic Regression (LR), Nearest Neighbor (NNC), Naive Bayes (NBC) – as well as an MLP classifier to train on the distilled dataset. We use python's scikit-learn and xgboost libraries for the experiments.

Table 1 shows the average balanced accuracy score (aggregated over the 26 datasets) of the different classifiers when trained on a dataset distilled to $N = 100$. **Base** is the performance of *a tuned model on the whole dataset in its original representation*. RND is the random sampling baseline. $\{\mathrm{X}\}$+Base is the application of distillation method or DM $\{\mathrm{X}\}$ on the original data representation. $\{\mathrm{X}\}$+$\{\mathrm{Y}\}$ is the application of DM $\{\mathrm{X}\}$ on the latent representations generated by the autoencoder or AE $\{\mathrm{Y}\}$. MLP* and GNN* are the multi-head variants of the AE.

With the exception of logistic regression, we find that using KMeans in the multi-head MLP encoder's *latent space* outputs the best performance. While using the multi-head encoder also helps the performance of KIP, we find that only effective for the MLP classifier. Agglomerative clustering also performed comparably to KMeans, but KMeans outputted a better performance in most datasets. Performance using the multi-head GNN encoder's latent space was comparable to that of the MLP encoder and even outperformed on some datasets, although the MLP autoencoder had a higher average performance. However, it is important to note that the GNN architecture uses far less parameters (between 5 to 10 % of the MLP encoder), which could lead to better scalability for large datasets. The GNN encoders have around 4,000 parameters on average while the MLP encoders have around 95,000.

Conclusion

We proposed and tested a new distillation method that works for tabular datasets. We tested our proposed pipeline with multiple autoencoder architectures and distillation methods on 26 different tabular datasets. In our experiments, we find that the multi-head MLP autoencoder can be used to effectively distill large tabular datasets. We also find that using naive clustering methods such as KMeans actually leads to better performance with classical ML models. This shows that the data distillation methods for image datasets which relies on the gradients produced by NNs may not translate directly to tabular datasets.

References

Medvedev, D.; and D'yakonov, A. 2020. New Properties of the Data Distillation Method When Working With Tabular Data. arXiv:2010.09839.

Nguyen, T.; Chen, Z.; and Lee, J. 2021. Dataset Meta-Learning from Kernel Ridge-Regression. arxiv:2011.00050.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2020. Dataset Distillation. arxiv:1811.10959.

Wu, Q.; Yang, C.; and Yan, J. 2021. Towards Open-World Feature Extrapolation: An Inductive Graph Learning Approach. arxiv:2110.04514.