Brytmind: A GPU Accelerated Machine Learning Framework for Relational Data Analytics

Jacek Karolczak[1,2], Albert Malewski[1], Bartłomiej Pokrzywa[1],
Piotr Nalewajko[1], Paweł Żurawski[1,3], Toby D. Young[1]

[1] Data Science Group, Brytlyt Ltd., Globe House, Eclipse Park, Kent, UK.

[2] Faculty of Computing and Telecommunications, Poznań University of Technology, Poznań, Poland.

[3] Faculty of Electrical Engineering, Automatics, Computer Science and Biomedical Engineering, AGH University of Science and Technology in Kraków, Kraków, Poland.

Introduction

Implementing Machine Learning (ML) and Deep Learning (DL) in a production environment is a complex challenge. Once a problem has been identified and a goal has been established, the development of an ML pipeline usually begins with an analysis of existing data sources. If the data is stored in a relational database, an appropriate connector can be naïvely provided by most data analysis tools. Next, the data is aggregated, and new features are engineered by executing Structured Query Language (SQL) queries using database management software.

The following stages of the ML pipeline require specialised software to enable communication between the database and the ML framework. This communication process can be time-consuming, as it involves transferring data between the CPU and GPU. Furthermore, implementation of the ML model requires an understanding of ML algorithms and their underlying concepts. Additionally, a dedicated software component is required for Machine Learning Operations (MLOps) to track experiments. After the model has been fitted to the data, another software component is needed to serve the model in production. All these steps require a significant amount of effort, expertise, and resources.

brytmind

A Machine Learning framework

Brytmind is a collection of GPU-enabled libraries suited to the rapid development of in-database ML and DL algorithms targeting BrytlytDB. Nowadays most analytic platforms and software come with ML connectors. Meanwhile, a dedicated package called Brytmind is in development for BrytlytDB. Brytmind encapsulates the data in BrytlytDB, stored on GPUs and exposes it as Torch tensors. Consequently, ML can be applied in situ, with neither extraction nor copying of the data. Brytmind's core functionalities were implemented in C++ as a matter of efficiency but Brytmind also has a Python API. The Python package implements numerous additional features. Brytmind is easily accessible as an ML development platform for beginners and advanced users. Some of the prominent features provided:

- Contains built-in model zoo, which is composed of several linear models (classification and regression) as well as DL models such as TabNet [1], DARNN [2], and SCINet [3].
- An exposed backend that enables the end user to take advantage of controlable, predefined trainng and testing schema that encaspulate their own custom models.
- Integration with MLflow [4] in order to address MLOps tasks, such as logging and comparison of ML experiments and model serving.

Listing 1: Training and tracking 'TabNet' as implemented in the model zoo using Brytmind tensors and MLFlow.

```python
import brytmind
import gmac

db_conn = gmac.Connection('gpumanager')
x = brytmind.FloatTensor('table_name', ['x1', 'x2'], db_conn)
y = brytmind.FloatTensor('table_name', ['y'], db_conn)

mlflow_logger = brytmind.modelzoo.callbacks.MLFlowLogger(
    tracking_uri, experiment_name
)
model = brytmind.modelzoo.TabNetClassifier(
    learning_rate=0.02, max_epochs=96, callbacks=mlflow_logger,
    x_transformations=brytmind.preprocessing.StandardScaler()
)
model.fit(x, y, train_size=0.3)
model.predict(x)
```

In-database model inference

A trained model can be loaded to BrytlytDB for use with the postgres command line. After loading inference can be performed by an executing SQL query as if the model were a regular table. SQL can thus be used as a part of a more complex workload and directly expose predictions from ML models to various applications.

Listing 2: Loading Brytmind model to BrytlytDB and running a simple inference using floats as input.

```
brytlyt=> /* Load model. */
brytlyt-> create foreign table mlp (
brytlyt(>     x1 real, x2 real, y real
brytlyt(> )
brytlyt-> server brytmind_fdw_server
brytlyt-> options (path 'models/mlp.bm');
brytlyt=> /* Run prediction. */
brytlyt-> select * from mlp where x1=2.5 and x2=3.5;
  x1  |  x2  | y
------+------+---
 2.5  | 3.5  | 6
(1 rows)
```

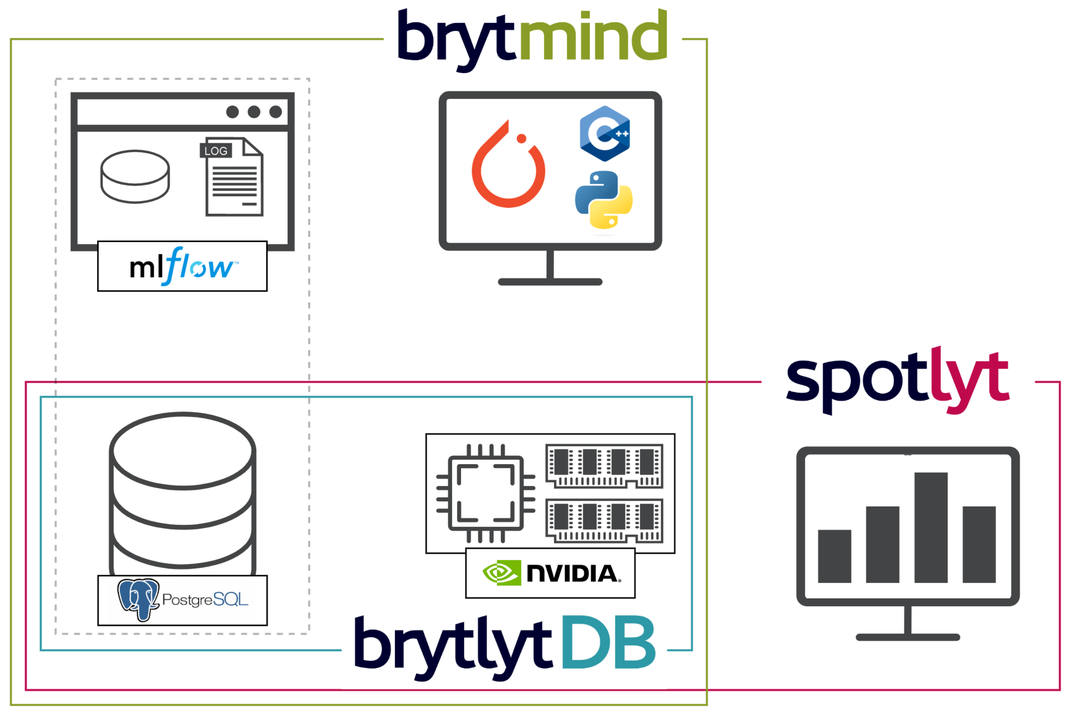

Figure 1: A schematic of the technologies that together compose Brytlyt's data analytics platform.

brytlytDB

GPU-accelerated database

BrytlytDB is a PostgreSQL database that uses GPU acceleration to enable speed of thought analytics on billion-row data. BrytlytDB can be deployed as an on-demand software solution, with GPU acceleration utilised only when a complex query is executed. This method of deployment reduces the carbon footprint while also minimising expenses.

spotlyt

Real-time visualisation

Prior to running ML algorithms, data exploration and analysis can be accomplished by utilising Spotlyt, a GPU-accelerated visualisation tool. Originally developed for use with BrytlytDB, Spotlyt is easily adaptable to the needs of users of any software that has a PostgreSQL connector, such as Amazon Redshift or CitusDB. Additionally, data processing for visualisation can be simply integrated with tools like Tableau and TIBCO Spotfire. Spotlyt can be utilised to create interactive dashboards that visualise predictions generated by Brytmind models.

The ML platform described in Figure 1 has proven value in the following sectors:

- Energy: for the simultaneous processing and visualisation of data from sensors, drilling, geolocalisation, and seismic activity.
- Telecommunications: to introduce intelligent benchmarking services.
- Public transport: for tracking metro, buses, trains, and taxi cabs in a smart city with a population of four million.

Summary

Brytmind is not a concept and is a high-end piece of software already capable of supporting the most demanding workloads. Brytmind takes applying AI to another level, making AI functionality fully embedded into the database. Using the tool is beneficial to new and advanced users, because of the wide range of utilities implemented in it.

Brytmind is under continuous development, to make it an even more powerful tool. Development plans include:

- Adding new models to the modelzoo to deliver current state-of-the-art algorithms.
- Improving support for more data sources, including CSV files and NVIDIA Rapids CuDF.
- Optimising modules and functions in terms of computational and memory complexity.

References

[1] S. Ö. Arik and T. Pfister. Tabnet: Attentive interpretable tabular learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8), 2021.

[2] Y. Qin, D. Song, H. Cheng, W. Cheng, G. Jiang, and G. W. Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[3] M. LIU, A. Zeng, M. Chen, Z. Xu, Q. LAI, L. Ma, and Q. Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. In *Advances in Neural Information Processing Systems*, 2022.

[4] MLFlow: An open source platform for the machine learning lifecycle. https://mlflow.org/.