CPU 上での XGBoost、LightGBM、CatBoost 推論の高速化

インテル® oneAPI データ・アナリティクス・ライブラリー（インテル® oneDAL）で XGBoost、LightGBM、CatBoost 推論ワークロードを高速化

Nikolay Petrov インテル コーポレーション AI ソフトウェア・エンジニアリング・マネージャー
Dmitry Razdoburdin インテル コーポレーション AI フレームワーク・エンジニア

最近は生成 AI の人気が高まっていますが、決定木で勾配ブースティングを使用することが表形式データを処理する最適な方法であることに変わりはありません。ほかの多くの技術やニューラル・ネットワークと比較した場合でも、より高い精度を提供します。XGBoost、LightGBM、および CatBoost 勾配ブースティングは、現実世界のさまざまな問題の解決、研究、Kaggle コンペティションで幅広く利用されています。これらのフレームワークは、デフォルトの設定でも優れたパフォーマンスを提供しますが、推論スピードには向上の余地があります。推論はマシンラーニング・ワークフローの最も重要なステージであることを考慮すると、パフォーマンス向上によってもたらされる利点は少なくありません。

次の例は、品質を損なうことなく、より高速な予測を得るためにモデルを変換する方法を示しています（**表 1**）。

```
import daal4py as d4p
d4p_model = d4p.mb.convert_model(xgb_model)
d4p_prediction = d4p_model.predict(test_data)
```

Dataset	n_estimators	daal.2023.2.0 vs xgb	daal.2023.2.0 vs lgbm	daal.2023.2.0 vs catboost
abalone	1000	3.28	4.93	7.94
airline-ohe	1000	8.38	13.76	2.50
higgs1m	100	3.85	13.66	5.06
higgs1m	300	3.67	12.97	3.37
higgs1m	1000	3.31	26.41	2.64
higgs1m	3000	3.40	26.14	2.38
higgs1m	10000	3.95	36.21	2.34
higgs1m	30000	3.85	44.28	2.21
mlsr	200	4.04	18.03	1.78
mortgage1Q	100	8.22	7.96	4.14
plasticc	60	2.63	6.21	3.71
santander	10000	2.58	11.58	1.75
airline	100	5.10	12.72	4.73
bosch	100	15.35	20.27	4.99
covtype	100	2.36	7.05	2.86
epsion	100	41.97	51.50	6.81
fraud	100	5.03	8.78	5.85
higgs	100	6.50	15.25	4.62
year_prediction_msd	100	12.27	14.79	6.60
geomean		**5.24**	**15.11**	**3.63**

表 1. daal4py の推論パフォーマンスと XGBoost、LightGBM、および CatBoost の比較。
色分けは各列の最良のケースと最悪のケースを示し、値が大きいほどパフォーマンスが優れていることを示します。

oneDAL のアップデート

推論の高速化に関する前回の記事 (43 号の「XGBoost と LightGBM 推論パフォーマンスの向上」) から数年が経過し、以下のような変更と改善が行われました。

1. API の簡素化および勾配ブースティング・フレームワークとの連携
2. CatBoost モデルのサポート
3. 欠損値のサポート
4. パフォーマンスの向上

API の簡素化および勾配ブースティング・フレームワークとの連携

メソッドが `convert_model()` と `predict()` のみになり、最小限の変更で既存のコードに簡単に統合できます。

一般的な XGBoost

```
from xgboost import xgb

xgb_model = xgb.fit(X_train, y_train)




xgb_prediction = xgb_model.predict(test_data)
```

daal4py モデルビルダー

```
from xgboost import xgb

xgb_model = xgb.fit(X_train, y_train)

from daal4py import d4p
d4p_model = d4p.mb.convert_model(xgb_model)

d4p_prediction = d4p_model.predict(test_data)
```

トレーニング済みモデルを daal4py に変換することもできます。

XGBoost :

```
d4p_model = d4p.mb.convert_model(xgb_model)
```

LightGBM :

```
d4p_model = d4p.mb.convert_model(lgb_model)
```

CatBoost :

```
d4p_model = d4p.mb.convert_model(cb_model)
```

scikit-learn* 形式の推定器もサポートされました。

```
from daal4py.sklearn.ensemble import GBTDAALRegressor
reg = xgb.XGBRegressor()
reg.fit(X, y)
d4p_predt = GBTDAALRegressor.convert_model(reg).predict(X)
```

アップデートされた API では、XGBoost、LightGBM、および CatBoost モデルをすべて 1 カ所で使用できます。メイン・フレームワークの場合と同様に、分類と予測の両方に同じ `predict()` メソッドを使用することもできます。詳細は、ドキュメント（英語）を参照してください。

CatBoost モデルのサポート

daal4py に CatBoost モデルのサポートを追加すると、勾配ブースティング・タスクの処理におけるライブラリーの汎用性と効率を大きく向上させました。CatBoost(Categorical Boosting の略)は、卓越したスピードとパフォーマンスで知られていますが、daal4py アクセラレーションを使用することで、さらに高速に処理できます（注 : カテゴリー特徴は推論ではサポートされていません）。CatBoost のサポートにより、daal4py は、最も一般的な 3 つの勾配ブースティング・フレームワークをサポートしました。

欠損値のサポート

欠損値は、現実世界のデータセットではよく発生します。欠損値は、ヒューマンエラー、データ収集の問題、観察メカニズムの制限など、さまざまな理由で発生する可能性があります。欠損値を無視したり不適切に処理すると、偏った解析が行われ、最終的にマシンラーニング・モデルのパフォーマンスが低下します。

欠損値のサポートは daal4py 2023.2 バージョンで追加されました。欠損値を含むデータでトレーニングされたモデルを使用したり、欠損値を含むデータを推論に使用することができます。データ・サイエンティストやエンジニアのデータ準備プロセスが合理化されるとともに、より正確で堅牢な予測が可能になります。

パフォーマンスの向上

前回のパフォーマンス比較以降、多くの最適化が oneDAL に追加されました（**表 2**）。

Datasets	n_estimators	**daal.2023.1.0** vs xgb	**daal.2023.2.0** vs xgb	**daal.2023.2.0** vs **daal.2023.1.0**
abalone	1000	2.09	3.28	1.57
airline-ohe	1000	8.44	8.38	0.99
higgs1m	100	3.33	3.85	1.16
higgs1m	300	3.35	3.67	1.10
letters	1000	0.90	1.87	2.07
mortgage1Q	100	7.25	8.22	1.13
airline	100	4.42	5.10	1.15
covtype	100	2.13	2.36	1.11
epsion	100	41.02	41.97	1.02
fraud	100	4.35	5.03	1.16
higgs	100	5.67	6.50	1.15
year_prediction_msd	100	10.73	12.27	1.14
geomean		**4.75**	**5.71**	**1.20**

表 2. 2023.1 および XGBoost と比較した 2023.2 バージョンのパフォーマンスの向上

すべてのテストは、AWS* EC2 c6i.12xlarge インスタンス（24 コアのインテル® Xeon® Platinum 8375C プロセッサーを含む）で動作している scikit-learn_bench（英語）と、Python* 3.11、XGBoost 1.7.4、LightGBM 3.3.5、CatBoost 1.2、daal4py 2023.2.1 を使用して実施されました。

新しいケースやその他の改善対象の拡大に、ぜひご協力ください。

daal4py の入手方法

daal4py は oneDAL（英語）ライブラリーの Python* インターフェイスで、PyPI*、conda-forge、および conda メインチャネルで利用できます。完全にオープンソースのプロジェクトであり、GitHub* リポジトリー（英語）経由での問題報告、リクエスト、貢献を受け付けています。PyPI* からライブラリーをインストールするには、次のコマンドを実行します。

```
pip install daal4py
```

conda-forge の場合は次のとおりです。

```
conda install -c conda-forge daal4py --override-channels
```

これらの結果は、コードに簡単な変更を加えることにより、現在のインテルのハードウェアで勾配ブースティング・データ解析タスクを大幅に高速化できることを示しています。これらのパフォーマンスの向上により、品質を犠牲にすることなく、コンピューティング・コストを低く抑え、より迅速に結果が得られるようになります。