點選率預測資料處理與模型訓練

NetApp Solutions

NetApp
October 31, 2024

目錄

點選率預測資料處理與模型訓練

用於資料處理和模型訓練的程式庫

下表列出用來建置此工作的程式庫和架構。所有這些元件均已與Azure的角色型存取與安全控管功能完全整合。

程式庫/架構	說明
dask cuML	若要讓ML在GPU上運作、請使用 "CUML程式庫" 提供使用dask存取水流立方ML套件的功能。藉由以GPU為基礎的高效能實作、讓您在處理CPU的基礎上、能夠以高達100倍的速度加速、藉由使用NetApp的功能來實作熱門的ML演算法、包括叢集、維度減量及回歸方法。
dask coudf	CUDF包含多種其他功能、可支援GPU加速擷取、轉換、負載（ETL）、例如資料子設定、轉換、單一熱編碼等。水流團隊維持 "dAsk擁抱程式庫" 這包括使用dask和cuDF的輔助程式方法。
科學套件學習	SciPKit學習提供數十種內建的機器學習演算法和模型、稱為評估工具。每個 "預估工具" 可搭配使用的部分資料 "適合" 方法。

我們使用兩部筆記型電腦來建構ML管線進行比較、其中一部是傳統的「大大管」學習方法、另一部則是以「急水流」和「dask」進行分散式訓練。每一部筆記型電腦均可個別測試、以瞭解其在時間與擴充方面的效能表現。我們會個別介紹每一部筆記型電腦、以展示使用「水流」和「dask」進行分散式訓練的優點。

Load Criteo按一下「日誌第15天」（日誌第15天）、然後訓練sci套 件學習隨機樹系模式

本節說明我們如何使用Pandas和dask DataFrames從Criteo Terabyte資料集載入Click記錄資料。使用案例與數位廣告有關、可預測是否要點選廣告、或是Exchange未在自動化管道中使用準確的模式、藉此建立使用者的設定檔。

我們從Click Logs資料集載入第15天的資料、總計45GB。在Jupyter筆記型電腦「CTer-andasRF-collated .ipynb」中執行下列儲存格、會建立一個包含前5、000萬列的大圓子資料框架、並產生scipit-記憶 隨機樹系模型。

```
%%time
import pandas as pd
import numpy as np
header = ['col'+str(i) for i in range (1,41)] #note that according to
criteo, the first column in the dataset is Click Through (CT). Consist of
40 columns
first_row_taken = 50_000_000 # use this in pd.read_csv() if your compute
resource is limited.
# total number of rows in day15 is 20B
# take 50M rows
"""
Read data & display the following metrics:
1. Total number of rows per day
2. df loading time in the cluster
3. Train a random forest model
"""
df = pd.read_csv(file, nrows=first_row_taken, delimiter='\t',
names=header)
# take numerical columns
df_sliced = df.iloc[:, 0:14]
# split data into training and Y
Y = df_sliced.pop('col1') # first column is binary (click or not)
# change df_sliced data types & fillna
df_sliced = df_sliced.astype(np.float32).fillna(0)
from sklearn.ensemble import RandomForestClassifier
# Random Forest building parameters
# n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
rf_model = RandomForestClassifier(max_depth=max_depth,
n_estimators=n_trees)
rf_model.fit(df_sliced, Y)
```

若要使用受過訓練的隨機樹系模型來執行預測、請執行本筆記型電腦的下一段。我們從第15天開始採用最後一百萬列作為測試集、以避免任何重複資料。儲存格也會計算預測的準確度、定義為模型準確預測使用者是否點選廣告的發生百分比。若要檢閱此筆記型電腦中任何不熟悉的元件、請參閱 "正式的sci套 件學習文件"。

```
# testing data, last 1M rows in day15
test_file = '/data/day_15_test'
with open(test_file) as g:
    print(g.readline())

# dataFrame processing for test data
test_df = pd.read_csv(test_file, delimiter='\t', names=header)
test_df_sliced = test_df.iloc[:, 0:14]
test_Y = test_df_sliced.pop('col1')
test_df_sliced = test_df_sliced.astype(np.float32).fillna(0)
# prediction & calculating error
pred_df = rf_model.predict(test_df_sliced)
from sklearn import metrics
# Model Accuracy
print("Accuracy:",metrics.accuracy_score(test_Y, pred_df))
```

在dask中載入第15天、並訓練dask cuML隨機樹系模型

以類似上一節的方式、在Pandas中載入Criteo按一下「日誌第15天」、然後訓練scier-記憶隨機樹系模式。在此範例中、我們使用dAsk couDF執行DataFrame載入、並在dask cuML中訓練隨機樹系模型。我們比較了本節訓練時間與規模的差異 "「訓練時間比較」。"

criteo_dASk_RF.ipynb

本筆記型電腦會匯入「umpy」、「累計」及必要的「dask」程式庫、如下列範例所示：

```
import cuml
from dask.distributed import Client, progress, wait
import dask_cudf
import numpy as np
import cudf
from cuml.dask.ensemble import RandomForestClassifier as cumlDaskRF
from cuml.dask.common import utils as dask_utils
```

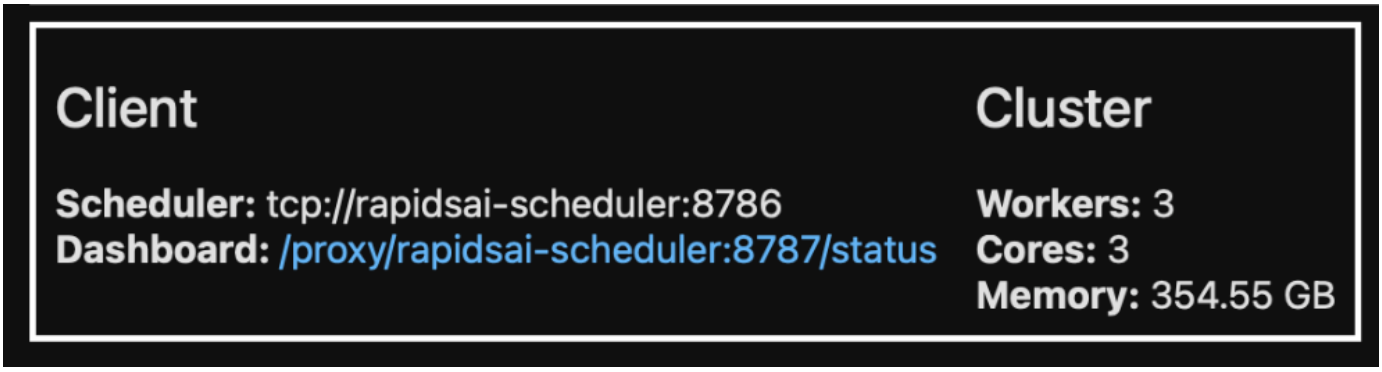
啟動dask Client()。

```
client = Client()
```

如果您的叢集設定正確、您可以看到工作節點的狀態。

```
client
workers = client.has_what().keys()
n_workers = len(workers)
n_streams = 8 # Performance optimization
```

在我們的高層叢集中、會顯示下列狀態：



請注意、dask採用的是閒置執行模式：dask並不是立即執行處理程式碼、而是建立直接執行的Acyclic圖表（DAG）。DAG包含一組工作及其互動、每位員工都需要執行這些工作。此配置表示在使用者指示dask以某種方式執行工作之前、工作不會執行。有了dask、您有三個主要選項：

- *在DataFrame上呼叫compacter()。*此呼叫會處理所有分割區、然後將結果傳回排程器、以供最終集合併轉換至cuDF DataFrame。除非排程器節點的記憶體不足、否則此選項應謹慎使用、且僅用於大幅減少的結果。
- * DataFrame上的呼叫持續（）。*此呼叫會執行圖表、但它不會將結果傳回排程器節點、而是將結果保留在整個叢集的記憶體中、讓使用者無需重新執行相同的處理、即可在管線中重複使用這些中繼結果。
- * DataFrame上的呼叫標頭（）。*就像cuDF一樣、此呼叫會傳回10筆記錄給排程器節點。此選項可用來快速檢查DataFrame是否包含所需的輸出格式、或記錄本身是否合理、視處理和計算而定。

因此、除非使用者撥打上述任一動作、否則工作人員會閒置等待排程器啟動處理。這種閒置執行模式在Apache Spark等現代化平行與分散式運算架構中相當常見。

以下段落使用dask cuML來訓練隨機樹系模型、以進行分散式GPU加速運算、並計算模型預測準確度。

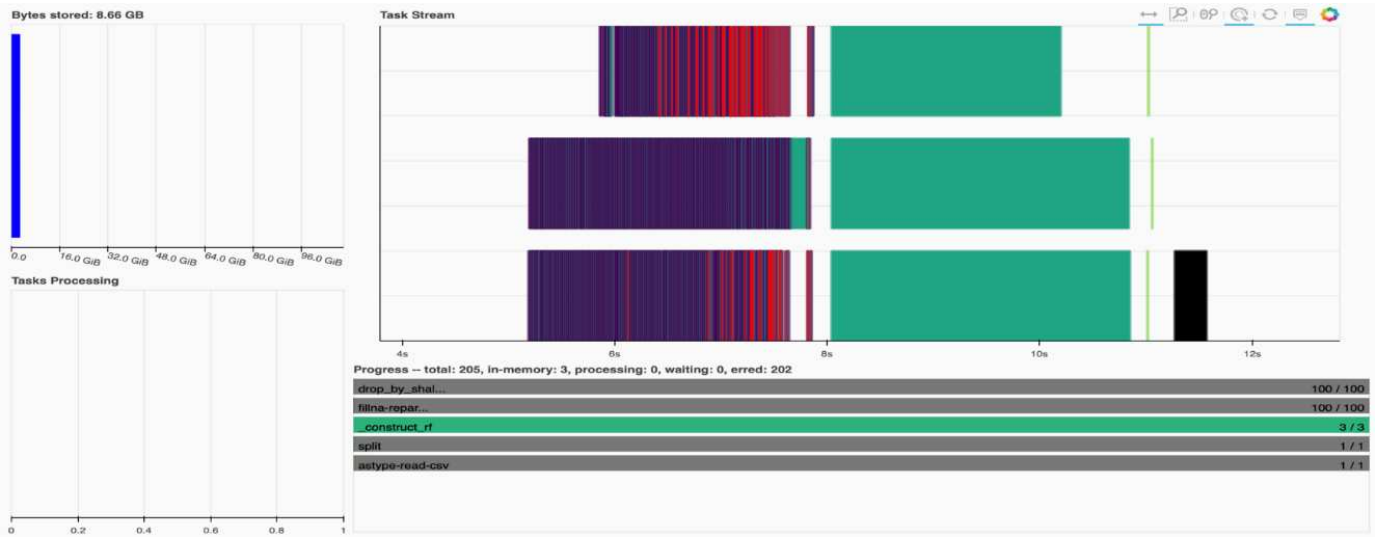
```
Adsf
# Random Forest building parameters
n_streams = 8 # optimization
max_depth = 10
n_bins = 16
n_trees = 10
cuml_model = cumlDaskRF(max_depth=max_depth, n_estimators=n_trees,
n_bins=n_bins, n_streams=n_streams, verbose=True, client=client)
cuml_model.fit(gdf_sliced_small, Y)
# Model prediction
pred_df = cuml_model.predict(gdf_test)
# calculate accuracy
cu_score = cuml.metrics.accuracy_score( test_y, pred_df )
```

使用原生工作串流儀表板監控dsask

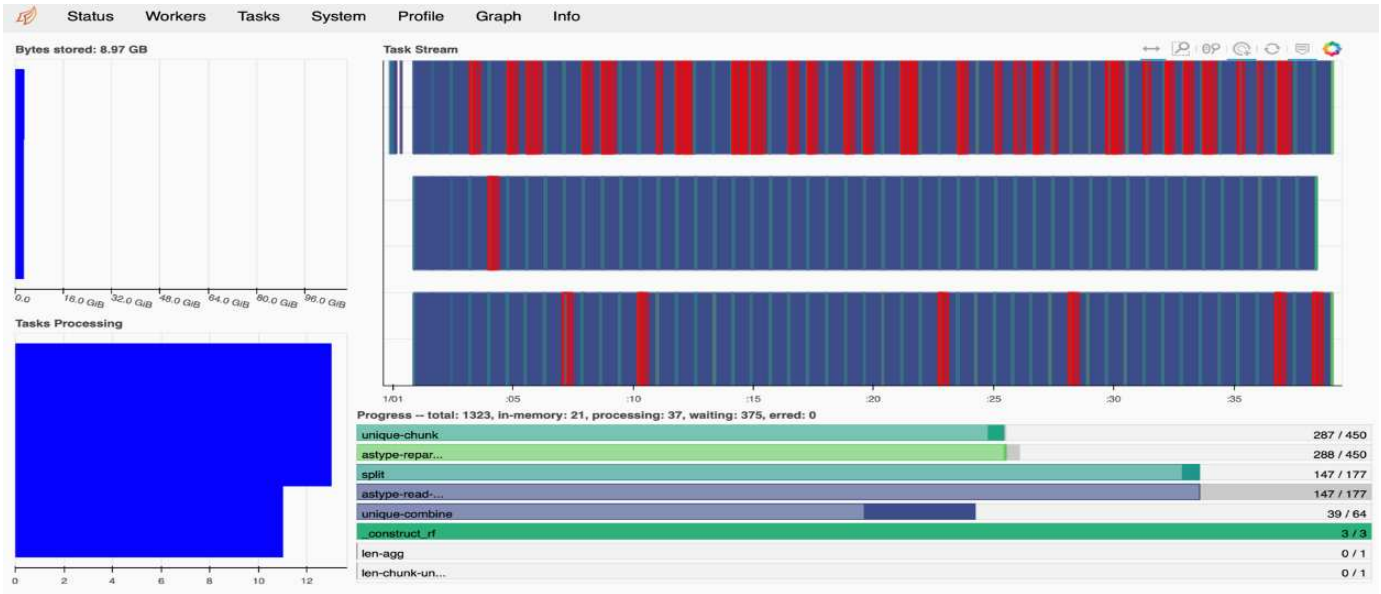
- "dask分散式排程器" 提供兩種形式的即時回饋：

  - 互動式儀表板包含許多繪圖和含有即時資訊的表格
  - 進度列適合在主控台或筆記型電腦中互動使用

在我們的案例中、下圖顯示如何監控工作進度、包括儲存的位元組、詳細細分串流數量的工作串流、以及執行相關功能的工作名稱進度。在我們的案例中、因為我們有三個工作節點、所以串流有三個主要區塊、而且色彩代碼會在每個串流中指出不同的工作。



您可以選擇分析個別工作、以毫秒為單位檢查執行時間、或找出任何障礙或阻礙。例如、下圖顯示隨機樹系模型擬合階段的工作串流。執行的功能相當多、包括用於DataFrame處理的獨特區塊、用於調整隨機樹系的_Constrature_RF等。大多數時間都花在DataFrame作業上、因為Criteo Click記錄中一天資料的大小（45GB）太大。



訓練時間比較

本節比較使用傳統Pandas的模型訓練時間與dask。對於Pandas而言、由於處理時間變慢、因此載入的資料量較少、以避免記憶體溢位。因此、我們將結果插補以提供公平的比較。

下表顯示隨機樹系模型使用的資料大幅減少時的原始訓練時間比較（資料集的20億個資料集中、有500萬列的資料）。此範例僅使用所有可用資料的0.25%以下。而對於dASK CUML、我們則針對所有200億個可用資料列、訓練隨機樹系模型。這兩種方法可提供相當的訓練時間。

方法	訓練時間
科學套件學習：在第15天只使用50M列做為訓練資料	47分21秒
「激流勇進」：將第15天的全部20B列當作訓練資料	1小時12分鐘11秒

如果我們以線性方式插補訓練時間結果、如下表所示、則搭配使用dask的分散式訓練將有顯著的優勢。傳統的「大作大作」學習方法需要13天的時間來處理和訓練45GB的資料、只需一天的點擊記錄、而「大浪」方法則能以相同數量的資料處理速度快262.39倍。

方法	訓練時間
科學套件-學習：將第15天的所有20B列當作訓練資料	13天、3小時、40分鐘及11秒
「激流勇進」：將第15天的全部20B列當作訓練資料	1小時12分鐘11秒

在上表中、您可以看到、透過使用配備DASK的PRUs將資料處理和模型訓練分散到多個GPU執行個體、相較於使用sciker-Learn模型訓練的傳統Pandas DataFrame處理、執行時間大幅縮短。此架構可在多節點、多GPU叢集內、在雲端及內部部署中進行橫向擴充。

利用**Prometheus**和**Grafana**監控**dask**和水流

部署所有項目之後、請針對新資料執行推斷。這些模型會根據瀏覽活動來預測使用者是否點選廣告。預測結果會儲存在dask couDF中。您可以使用Prometheus監控結果、並

在Grafana儀表板中以視覺化的方式呈現結果。

如需詳細資訊、請參閱 "中速漂流"。

使用NetApp DataOps Toolkit的資料集與模型版本管理

NetApp DataOps Toolkit for Kubernetes可將儲存資源與Kubernetes工作負載抽象化、直到資料科學工作區層級。這些功能均以簡單易用的介面進行封裝、專為資料科學家和資料工程師所設計。使用熟悉的Python程式形式、工具套件可讓資料科學家和工程師在短短數秒內配置及銷毀JupyterLab工作區。這些工作區可包含TB甚至PB的儲存容量、讓資料科學家能夠將所有訓練資料集直接儲存在專案工作區中。現在已經不再需要分別管理工作區和資料磁碟區了。

有關詳細信息，請訪問工具包 "GitHub儲存庫"。

Jupyter筆記型電腦供參考

本技術報告有兩部Jupyter筆記型電腦：

- "* CTR - pastasRF-collated .ipynb.*" 這款筆記型電腦會從Criteo TB Click日誌資料集載入第15天的資料、將資料處理及格式化為子網頁資料框架、訓練科學套件學習隨機樹系模型、執行預測並計算準確度。
- "* criteo_dASk_RF.ipynb.*" 本筆記型電腦會從Criteo TB載入第15天的內容、按一下「記錄資料集」、將資料處理及格式化為dask couDF、訓練dask cuML隨機樹系模型、執行預測並計算準確度。藉由運用GPU來運用多個工作節點、這種分散式資料與模型處理與訓練方法效率極高。您處理的資料越多、相較於傳統的ML方法、所節省的時間就越多。您可以將此筆記型電腦部署在雲端、內部部署或混合式環境、其中Kubernetes叢集包含不同位置的運算和儲存設備、只要您的網路設定能夠自由移動資料和模型發佈。