DEEP LEARNING INSTITUTE

DLI Accelerated Data Science Teaching Kit

Lecture 21.2 - Refactoring Workloads

Refactoring

CPU to GPU Data Science

- Large amounts of existing code in PyData (Numpy, pandas, scikit-learn, etc.)
- RAPIDS uses Pandas-like API
- Very easy and straightforward
- Simple changes in a few lines of code
- Replace import statements

```
import pandas as pd  →  import cudf
import numpy as np   →  import cupy as cp
```

- Use the new imports in place of previous libraries

Example 1

Pandas to cuDF

- Use the cudf df like pandas df
  - Examples: sort_values, concat, merge, unique, std, iloc, groupby

```
import pandas as pd
df = pd.read_csv('df.csv’)
df1 = pd.read_csv(‘df1.csv’)
pd.concat([df, df1])
df.fillna(0)
df.head(10)
```

```
import cudf
df = cudf.read_csv('df.csv’)
df1 = cudf.read_csv(‘df1.csv’)
cudf.concat([df, df1])
df.fillna(0)
df.head(10)
```

Same output, but faster!

Example 2

Numpy to cuPY

- Use the cupy array like numpy array
  - Examples: randint, arrange, zeros, shape, max, flatten, sort

```
import numpy as np
choices = range(6)

probs = np.random.rand(6)
s = sum(probs)
probs = [e / s for e in probs]
selected = np.random.choice(choices, 10000, p=probs)

print(selected.shape)
```

```
import cupy as cp
choices = range(6)

probs = cp.random.rand(6)
s = sum(probs)
probs = [e / s for e in probs]
selected = cp.random.choice(choices, 10000, p=probs)

print(selected.shape)
```

Same output, but faster!

Example 3

Scikit learn to cuML

- cuML has similar capabilities as sklearn
  - Examples: train_test_split, SVC, KMeans, LinearRegression, LabelBinarizer, NearestNeighbors

```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split


X_train, X-test, y_train, y_test =
train_test_split(X, y, random_state =0)

model = LinearRegression()

model.fit(X_train, y)
y_pred = model.predict(X_test)
```

→

```
import cuml.LinearRegression
from cuml.preprocessing.model_selection import
train_test_split

X_train, X-test, y_train, y_test =
train_test_split(X, y, random_state =0)

model = cuml.LinearRegression()

model.fit(X_train, y)
y_pred = model.predict(X_test)
```

Same output, but faster!

NVIDIA. DEEP LEARNING INSTITUTE | Georgia Tech | PRAIRIE VIEW A&M UNIVERSITY

DLI Accelerated Data Science Teaching Kit

Thank You