THURSDAY, SEPTEMBER 17, 2020

Using joblib to speed up your Python pipelines

Why joblib?

There are several reasons to integrate joblib tools as a part of the ML pipeline. There are major two reasons mentioned on their website to use it. However, I thought to rephrase it again:

Capability to use cache which avoids recomputation of some of the steps

Execute Parallelization to fully utilize all the cores of CPU/GPU.

Beyond this, there are several other reasons why I would recommend joblib:

Can be easily integrated

No specific dependencies

Saves cost and time

Easy to learn

1. Using Cached results

Basically, store the computed results in memory

```
from joblib import Memory


# Define a location to store cache

location = '~/Desktop/temp/cache_dir'

memory = Memory(location, verbose=0)


result = []
```

FOLLOWERS

Followers (3)

Follow

BLOG ARCHIVE

► 2025 (164)

► 2024 (303)

► 2023 (285)

► 2022 (238)

► 2021 (224)

▼ 2020 (240)

► December (40)

► November (55)

► October (70)

▼ September (41)

ThreeJS how to create a Plane, and a sphere, box s...

ThreeJS MeshLambertMaterial

ThreeJS what is point light and ambient light

ThreeJS what are different light sources

What is glTF file format

ThreeJS glLF loader

Threejs loading models

Data visualisation in Tableau , D3, R, Python

The mu editor for python programming

A quick threejs 3D model

Why should stick to react

Linux 2 : What does the file

```
# Function to compute square of a range of a number:

def get_square_range_cached(start_no, end_no):

    for i in np.arange(start_no, end_no):

        time.sleep(1)

        result.append(square_number(i))

    return result


get_square_range_cached =
memory.cache(get_square_range_cached)


start = time.time()

# Getting square of 1 to 50:

final_result = get_square_range_cached(1, 21)

end = time.time()


# Total time to compute

print('\nThe function took {:.2f} s to compute.'.format(end - start))

print(final_result)
```

To clear the cache results just do the below

```
memory.clear(warn=False)
```

2. Parallelization

As the name suggests, we can compute in parallel any specified function with even multiple arguments using “joblib.Parallel”. Behind the scenes, when using multiple jobs (if specified), each calculation

name red with black b...
Sails - Specify SSL certificate
Lets Encrypt, How to assign certificate to Certbot
Lets encrypt : DNS problem: NXDOMAIN looking up TX...
Vue 3 - what's in it?
What is Vue.js
Automatic recovery of EC2 instance using CloudWatch
Firebase Analytics, How to add custom parameter re...
Firebase Auth - Displays Domain Not verified
Are we able to attach SSL to IP address?
Firebase Analytics
Material UI How to set different color for typography
Some of the cool new UI designs
Using joblib to speed up your Python pipelines
Slide Model power point template
Android 11 - Supported phones
ThreeJS what is OrbitControl
Javascript How to implement Sleep
iOS Swift get device model number
Material UI - How to do localisation for MUiPicker
Javascript how to save the file to local PC
Javascript - Upload files using React JS
iOS 14 new changes:
Which all industries get affected by AR/VR
How do a Hex view of a file
What is Global In Node JS

does not wait for the previous one to complete and can use different processors to get the task done. For better understanding, I have shown how Parallel jobs can be run inside caching.

```
#Import package
from joblib import Parallel, delayed
from joblib import Memory

location = 'C:/Users/pg021/Desktop/temp/cache_dir'
memory = Memory(location, verbose=0)
costly_compute_cached = memory.cache(costly_compute)

def data_processing_mean_using_cache(data, column):
    """Compute the mean of a column."""
    return costly_compute_cached(data, column).mean()

start = time.time()
results = Parallel(n_jobs=2)(
    delayed(data_processing_mean_using_cache)(data, col)
    for col in range(data.shape[1]))
stop = time.time()

print('Elapsed time for the entire processing: {:.2f} s'
      .format(stop - start))
```

Here we can see that time for processing using the Parallel method was reduced by 2x.

3. Dump and Load

How much max bandwidth needed for VoIP?

How does lets encrypt work?

Material UI Styling

What is Higher order component ?

- ► August (14)
- ► July (11)
- ► April (2)
- ► January (7)

- ► 2019 (261)
- ► 2018 (259)
- ► 2017 (83)
- ► 2016 (124)
- ► 2015 (276)
- ► 2014 (200)
- ► 2012 (5)
- ► 2010 (3)
- ► 2009 (3)

ABOUT ME

LIVINGMOBILE

VIEW MY COMPLETE PROFILE

We often need to store and load the datasets, models, computed results, etc. to and from a location on the computer. Joblib provides functions that can be used to dump and load easily:

4. Compression methods

Supported ones are:

a. Simple Compression:

b. Using Zlib compression:

c. Using lz4 compression:

I find joblib to be a really useful library. I have started integrating them into a lot of my Machine Learning Pipelines and definitely seeing a lot of improvements.

References:

https://towardsdatascience.com/using-joblib-to-speed-up-your-python-pipelines-dd97440c653d

POSTED BY LIVINGMOBILE AT 10:45 AM

NO COMMENTS:

POST A COMMENT

To leave a comment, click the button below to sign in with Google.

SIGN IN WITH GOOGLE

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)