عنوان

آشنایی با Large Data

تهیه کننده

سارا معصومی

تیر ۱۴۰۲

Large Data چیست؟

احتمالا با عبارت Big Data و ویژگی های آن آشنا هستید، اما بجز Big Data، دیتاست‌هایی وجود دارن که حجم داده‌های آن زیاد و روابط بین متغیرها پیچیده است بنابراین Ram / Memory سیستم ما توانایی پردازش آن حجم از اطلاعات را ندارد، اما با یکسری ترفندها میتوان تحلیل و بررسی این گونه از مجموعه داده‌ها را بر روی یک سیستم انجام داد و دیگر نیازی به ارتقاء سخت افزار یا توزیع دیتا بین چند سیستم نداریم. به این مدل از دیتاست‌ها Large Data میگویند.

سه راه‌حل کلی برای حل مسئله پردازش Large Data بر روی single computer وجود دارد:

۱) استفاده از الگوریتم‌های مناسب
۲) استفاده از ساختار داده مناسب
۳) انتخاب ابزار مناسب (سخت افزار و نرم افزار) برای کار با Large Data

هر کدام از راه‌حل‌های بالا به چند بخش تقسیم می‌شوند که در ادامه به آنها پرداخته خواهد شد.

۱) استفاده از الگوریتم‌های مناسب

- الگوریتم‌های آنلاین : Online Algorithm

در علوم کامپیوتر یک الگوریتم آنلاین ، الگوریتمی است که میتواند داده ها را تدریجا در بسته های کوچکتر به ترتیبی که به الگوریتم داده میشوند در لحظه پردازش کند بدون اینکه کل داده ها در دسترس باشند.بنابراین نیازی نیست کل داده ها با توجه به حجم بالای آنها یکجا پردازش شوند.

- الگوریتم های بلوکی : Blocked Algorithm

الگوریتم های خاصی برای کار با ماتریس‌های بزرگ هستند که میتوانند یک ماتریس کامل را به بلوک‌های کوچکتر تقسیم کنند و به جای ماتریس کامل با قسمت‌های کوچکتر کار کنند که در این حالت میتوان ماتریس‌های کوچکتری را در حافظه بارگزاری کرد و محاسبات را انجام داد. بنابراین از خطای کمبود حافظه جلوگیری خواهد شد.برای اینکار میتوان از کتابخانه های bcolz و Dask استفاده کرد.

- الگوریتم مپ ردیوس: MapReduce Algorithm

کاربرد این الگوریتم در پردازش‌های موازی میباشد در اینجا map به معنای نگاشت و reduce به معنای کاهش است.این الگوریتم از سه مرحله زیر تشکیل شده است:

۱.mapping

۲.group by key

۳.reduce

فرض کنید داده‌های ما متن ۱۰۰۰ خبر سیاسی باشد و ما میخواهیم تعداد کلمات بخصوصی را بررسی کنیم، الگوریتم ابتدا کلمات موجود در متون را به صورت زوج‌های کلید مقدار اصطلاحا map میکند. سپس مرحله گروه‌بندی آغاز میشود که مقادیر کلید(دراینجا کلمه) را بر اساس تعداد تکرار آن مینوسید، حال نوبت مرحله reduce میباشد در این فاز عملیات مشابهی بر روی کلیدهایی که مانند هم هستند انجام میشود برای مثال حاصل جمع مقدارهای هر کلید را بدست می‌آورد که خروجی این عملیات همان پاسخ تعداد تکرار کلمه مورد نظر در متن اخبار می‌باشد. هر کدام از مراحل ذکر شده میتوانند به صورت مستقل یا موازی اجرا شوند.

۲) استفاده از ساختار داده‌ای مناسب

- SPARSE STRUCTURES

فرض کنید یک دیتاست حجیم در اختیار دارید اما فقط تعداد بسیار کمی از این داده‌ها شامل اطلاعات هستند و باقی داده‌ها اطلاعاتی در اختیار ما قرار نمی‌دهند، برای مثال یک ماتریس بسیار بزرگ را در نظر بگیرید که تنها تعداد انگشت شماری از درایه‌های آن شامل مقدار باشد و باقی درایه‌ها ۰ یا تعریف نشده هستند، معمولا هنگام تبدیل داده‌های متنی به داده‌های باینری اینگونه ماتریس‌ها به‌دست می‌آیند، در چنین حالتی فضای ذخیره‌سازی سیستم ما صرف داده‌هایی خواهدشد که شامل اطلاعات مفیدی برای ما نیستند، رفتار درست در برابر چنین مجموعه داده‌ای فشرده‌سازی آن است به طوری که تنها اطلاعات مفید را حفظ و باقی حذف شوند. کتابخانه Pandas در پایتون روش‌هایی را برای فشرده‌سازی sparse data در اختیار ما قرار داده است که در صورت لزوم از آنها بهره ببریم.

- TREE STRUCTURES

درخت‌ها یکی از انواع ساختارهای‌داده هستند که برخلاف ساختارهای دیگر مانند لیست‌های پیوندی ، صف‌ها یا پشته‌ها، داده‌ها را به صورت خطی ذخیره نمی‌کنند بنابراین اطلاعات خیلی سریعتر از یک جدول بازیابی میشود چرا که برای دسترسی به یک داده‌ی مشخص به‌جای اسکن کل داده‌ها فقط فرزندان گره مربوطه را اسکن می‌کنند که در دنیای اطلاعات امروزه با توجه به پیچیدگی زمانی بالای داده‌ها استفاده از ساختمان داده‌ی درختی در افزایش سرعت بازیابی اطلاعات، تاثیر زیادی خواهد داشت. یک درخت دارای تعاریفی همچون ریشه ، گره والد، گره فرزند، یال و... است که برای کار با درخت‌ها لازم است با این عناوین آشنا شوید.

- HASH TABLES

جداول هش ساختارهای داده‌ای هستند که برای هر مقدار در داده‌های ما یک کلید محاسبه می‌کنند و کلیدها را در یک قسمت قرار می‌دهند. به این ترتیب وقتی می‌خواهیم روی داده‌ها کار کنیم با انتخاب کلید مناسب اطلاعات به سرعت بازیابی می‌شوند. دیکشنری‌ها در پایتون همان پیاده‌سازی جدول هش هستند. برای مثال زمانی که یک سیستم توصیه‌گر در یک پایگاه‌داده بسازیم، جداول هش به عنوان شاخص برای بازیابی سریع اطلاعات استفاده می‌شود.

۳) استفاده از ابزارهای مناسب جهت پردازش

- Cython

برای رایانه، مفهوم ۱- یا +۱ و یا -۱.۰۰ +۱.۰۰ متفاوت است. مثال اول نشان دهنده اعداد صحیح و مثال دوم اعداد اعشاری می‌باشد. که این محاسبات توسط قسمت‌های مختلف CPU انجام می‌شود. در پایتون لازم نیست نوع داده‌ای را که استفاده می‌کنید مشخص کنید، بنابراین کامپایلر پایتون خود باید آنها را استنتاج کند. اما استنتاج آنها یک عملیات کند محسوب میشود به همین دلیل پایتون یکی از سریع‌ترین زبان‌های موجود در دنیا نیست. خب Cython، ابرمجموعه‌ای از پایتون است که این مشکل را با مجبور کردن برنامه‌نویس برای تعیین نوع داده در حین توسعه برنامه حل می‌کند. هنگامی که کامپایلر این اطلاعات را

داشته‌باشد، برنامه‌ها را بسیار سریعتر اجرا می کند. برای اطلاعات بیشتر در مورد Cython می‌توانید به سایت https://cython.org مراجعه کنید.

- Numexpr

Numexpr یک ارزیابی‌کننده عبارات عددی برای NumPy است اما می‌تواند چندین برابر سریعتر از NumPy اصلی عمل کند. درواقع برای سرعت بخشیدن به هدفتان، عبارت شما را بازنویسی می‌کند و از یک کامپایلر داخلی استفاده می‌کند. برای جزئیات بیشتر از Numexpr می‌توانید به سایت https://github.com/pydata/numexpr مراجعه کنید.

- Bcolz

Bcolz به شما کمک می کند تا مشکل کمبود حافظه که ممکن است هنگام استفاده از NumPy رخ دهد را رفع کنید. درواقع می‌تواند آرایه‌ها را در یک فرم فشرده بهینه ذخیره و کار کند. این نه تنها نیاز به داده‌های شما را کاهش می‌دهد، بلکه از Numexpr در پس زمینه برای کاهش محاسبات مورد نیاز هنگام انجام محاسبات با آرایه‌های bcolz استفاده می‌کند. برای آشنا شدن با Bcolz به سایت http://bcolz.blosc.org مراجعه کنید.

- Blaze

اگر می‌خواهید از قدرت یک پایگاه داده همانند روش پایتونیکِ کار با داده‌ها استفاده کنید، Blaze ایده‌آل شماست. Blaze کد پایتون شما را به SQL ترجمه می‌کند، و می‌تواند ذخیره داده‌های بیشتری را نسبت به پایگاه داده‌های رابطه‌ای مدیریت کند. Blaze روشی یکپارچه برای کار با بسیاری از پایگاه‌های داده و کتابخانه‌های داده ارائه می‌دهد. البته Blaze هنوز در حال توسعه است، برای دسترسی به امکانات آن می‌توانید به سایت http://blaze.readthedocs.org/en/latest/index.html مراجعه کنید.

- Theano

Theano به شما این امکان را می‌دهد تا مستقیماً با واحد پردازش گرافیکی یعنی GPU کار کنید و بخشی از پردازش ها را به جای CPU به GPU بسپارید که البته همراه با یک کامپایلر عالی در اختیار شما قرار

میگیرد. علاوه بر این، یک کتابخانه مفید برای مفهوم ریاضی پیشرفته یعنی Tensor است، برای آشنایی با Theano می‌توانید به سایت http://deeplearning.net/software/theano مراجعه کنید.

- Dask

Dask شما را قادر می‌سازد تا محاسبات خود را بهینه و آن‌ها را به طور موثر اجرا کنید. همچنین به شما امکان می‌دهد محاسبات را بین بخش‌های مختلف توزیع کنید. درواقع قابلیت‌های محاسباتی parallel(همزمان) را فراهم میکند به طوری که داده‌های شما به قطعات کوچیک‌تر تقسیم و با به کارگیری پردازنده‌های مختلف بطور همزمان پردازش می‌شوند. برای آشنایی با Dask می‌توانید به سایت http://dask.pydata.org/en/latest مراجعه کنید.

" با تشکر از توجه شما "

منبع

Introducing Data Science: Big Data, Machine Learning, and more, using Python tools First Edition y Davy Cielen (Author), Arno Meysman (Author), Mohamed Ali (Author)
https://www.amazon.com/Introducing-Data-Science-Machine-Learning/dp/1633430030

دانلود مقاله‌های بیشتر از https://github.com/Sara0M