계산과학 데이터의 인공지능 분석을 위한 확장성 있는 특징 데이터 추출 자동화 시스템

안선일*
*한국과학기술정보연구원
siahn@kisti.re.kr

A scalable and automated feature data extraction system for AI analysis of computational science data

Sunil Ahn*
*Korea Institute of Science Technology and Information

요 약

AI 분석 과정에서 특징 데이터 추출은 분석 성능에 큰 영향을 미칠 뿐만 아니라 가장 많은 시간을 소요하는 과정 중의 하나이다. 계산과학 데이터는 HPC를 활용하여 생산되므로 데이터가 크고 복잡할 뿐 아니라 데이터의 수도 방대한 경우가 많다. 이 때문에 계산과학 데이터로부터 특징 데이터 추출하는 과정은 복잡성이 크고, 소요 시간도 매우 크다. 본 논문은 먼저 계산과학 데이터로부터 특징 데이터 추출하는 과정에 대한 요구사항과 이슈들을 분석한다. 그리고 확장성을 고려한 계산과학 데이터의 인공지능 분석을 위한 특징 데이터 추출 자동화 시스템을 제안한다.

1. 서론

최근 다양한 분야에서 인공지능의 도입이 활발한 가운데 계산과학 분야에서도 인공지능 기술 활용의 필요성이 부각되고 있다. 대표적인 활용 분야로는 소재 분야로써, 최근 소재 개발 시간을 더욱 단축하기 위해 계산과학 기반의 소재 물성 데이터베이스를 구축하고, 인공지능 방법을 적용한 서비스들을 제공하고 있다 [1].

높은 정확도의 인공지능 모델 개발을 위해서는 원시 데이터로부터 적절한 특징(feautre) 데이터를 선정하고 추출하는 과정이 무엇보다도 중요하다. 불필요한 특징들은 인공지능 모델 학습에서 메모리 공간을 낭비하고 학습시간을 길어지게 한다. 그러므로 결과에 중요한 영향을 줄 수 있는 특징들만을 가려내고 추출하는 과정이 무엇보다도 중요하다 [2].

특징 데이터의 추출은 한번 작업으로 끝나지 않고 원하는 성능의 인공지능 모델을 확보하기까지 특징들을 변경하면서 반복 수행한다. 이 때문에 특징 데이터 추출 과정은 많은 시간을 소요한다. 계산과학 데이터는 HPC를 활용하여 생산되므로 데이터가 크고 복잡할 뿐 아니라 데이터의 수도 방대한 경우가 많다. 이 때문에 계산과학 데이터로부터 특징 데이터 추출 과정은 복잡성이 크고, 소요 시간도 매우 크다.

본 논문은 먼저 계산과학 데이터로부터 특징 데이터 추출에 대한 요구사항과 이슈들을 분석한다. 그리고 확장성을 고려한 계산과학 데이터의 인공지능 분석을 위한 특징 데이터 추출 자동화 시스템을 제안한다. 마지막으로는 결론을 제시한다.

2. 계산데이터로부터 AI 분석을 위한 특징 데이터 추출에 대한 요구사항 분석

계산데이터로부터 특징 데이터 추출에 대한 요구사항은 다음과 같다. 첫째, 복잡하고 방대한 계산과학 데이터로부터 특징 데이터를 추출하는 소요 시간을 단축하기 위해 클러스터 자원의 활용이 필요하고, 특징 데이터 추출 작업들에 대한 관리가 용이해야 한다. 대용량의 동시 작업을 위한 일반적인 클러스터 자원 활용 방안은 배치 방식의 작업 제출이다. 이 경우 방대한 수의 계산과학 데이터, 예를 들어 수십만 건을 배치 작업으로 실행하는 경우, 스케줄러에 따라 사용자별 제출 가능한 최대 작업의 수에 제한이 있거나 스케줄러 부하를 유발할 가능성이 있

다. 또한, 배치 작업의 수가 수십만 건에 이르는 경우 작업들의 모니터링, 오류 확인, 오류 작업에 대한 재실행 등에 대한 복잡성은 직접 사람이 수작업으로 처리하기에는 불가능한 수준이다. MPI [3]와 같은 실행환경을 활용하여 특징 데이터를 추출하는 작업을 생성하는 경우 관리가 효율적이지만, 사용자가 MPI 병렬 코드를 직접 개발하는 것은 복잡하고 어려워하는 경우가 많다.

둘째, 특징 데이터를 추출하는 코드를 개발하는 환경과 대용량 데이터를 대상으로 특징 데이터를 추출하는 실행 환경 사이의 유연하고 원활한 연계가 요구된다. 최근 AI 관련 분석 코드의 개발은 웹 기반의 jupyter [4] 라는 인터액티브(interactive)한 툴을 흔히 활용하는 반면에, 대용량 데이터로부터 특징 데이터를 추출하는 실행 환경은 별도의 클러스터 환경인 경우가 많아서 두 개의 이질적 환경의 원활한 연계가 필요하다. 또한, 개발 환경과 실행환경 모두 되도록 동일한 실행환경으로 구성되는 것이 복잡성을 줄일 수 있다. 특징 데이터를 추출하는 과정은 특정 라이브러리나 툴의 설치를 요구하는 경우가 많다. 개발 환경에서 활용한 툴과 실행환경에서 활용되는 툴의 버전 등이 상이한 경우 올바른 실행이 보장되지 않는다.

셋째, 특징 데이터 추출 과정에서 활용된 코드는 인공지능 모델의 서비스 배포 과정에서 재활용이 가능해야 한다. 기존 인공지능 응용의 입력은 테스트 기반이거나 그림 파일인 경우가 대부분이고, 인공지능 서비스를 배포하는 측면에서 입력 데이터의 처리가 복잡하지 않다. 반면에 계산과학 데이터의 경우 이질적 파일들을 입력으로 하기 때문에 입력 데이터의 처리 과정이 복잡하고, 되도록 재활용이 필요하다.

3. 특징 데이터 추출 자동화 시스템

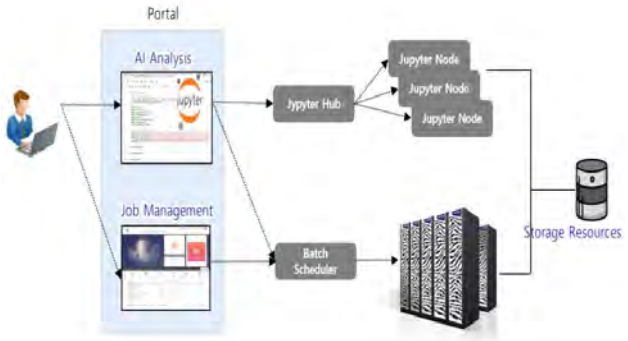
본 장에서는 확장성을 고려한 계산과학 데이터의 인공지능 분석을 위한 특징 데이터 추출 자동화 시스템을 제안한다. 제안하는 시스템의 주요 기능은 jupyter 기반 특징 데이터 추출 코드 개발환경, 클러스터를 활용한 작업 실행환경, 웹 기반의 작업 모니터링 및 관리환경 등을 포함한다. 사용자는 jupyter 환경에서 특징 데이터 추출 코드를 개발하고 소규모 데이터를 대상으로 시험한다. 개발이 완료된 특징 데이터 추출 코드는 대규모 데이터를 대상으로 클러스터 자원을 활용하여 실행된다. 그리고 웹 기반 환경을 통해 실행한 특징 데이터 추출 작업을 모니터링하고, 오류 혹은 최종 결과를 웹 기반 다운로드 할 수 있다.

우리는 먼저 대용량 계산과학 데이터를 대상으로 하여 확장성을 지원하는 것은 물론, 사용자가 직접 MPI 병렬 코드를 작성하는 복잡성을 제거하여 쉽고 편리하게 사용할 수 있는 특징 데이터 추출 자동화 시스템을 제안한다. 이 시스템은 클러스터에서 배치 스케줄러를 활용하여 MPI 기반 실행되기 때문에 기존 계산과학 클러스터를 그대로 활용할 수 있을 뿐 아니라, 각 데이터마다 작업을 생성하지 않고 한 MPI 작업으로 여러 원시 계산과학 데이터들을 처리하여 특성 데이터를 추출하기 때문에 관리 측면에서 효율적이다. 사용자는 jupyter 환경에서 개발한 python 코드, 처리하고자 하는 대상 데이터 목록, 활용하고자 하는 CPU 코어의 수를 입력하면 나머지 복잡한 과정들은 자동화 시스템을 통해 병렬 처리된다. 사용자의 작업은 python API와 웹 기반 GUI 인터페이스를 통해 시작된다. 시작된 작업은 MPI 기반 여러 병렬 태스크들을 실행시키고, 각 태스크들은 처리 대상 데이터들을 동일한 크기로 분할하여 처리한다. 그리고 모든 태스크들에서 특성 데이터 추출이 완료되면 결과들을 취합하여 하나의 CSV 형식 파일로 저장한다.

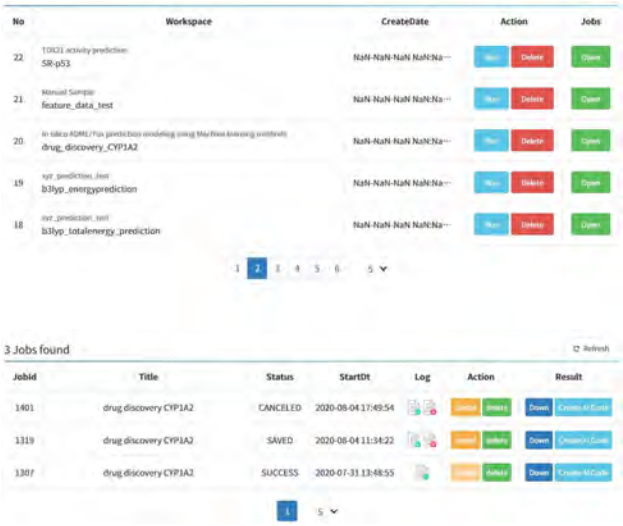
사용자의 특징 데이터 추출 코드는 세번째 요구사항인 재활용성을 높이기 위해 주어진 템플릿 함수를 활용하여 개발된다. 템플릿 함수는 특징을 생산하는 _process_input() 함수와, 레이블을 생산하는 _process_output() 함수로 구성된다. 각 함수의 입력은 계산과학 원시 데이터가 있는 디렉터리 위치이고, 함수들의 출력은 python 언어의 딕셔너리(dictionary) 형태를 가진 특징 데이터이다. 특징 데이터 추출 자동화 시스템은 처리할 데이터를 대상으로 사용자가 제공한 python 코드로부터 템플릿 함수들을 불러와 실행하고 그 결과를 CSV 파일로 저장한다. 템플릿 함수들을 불러와서 활용하는 것은 python ast [5] 모듈을 활용하였다.

우리는 두 번째 요구사항인 개발환경과 실행환경 이질성 문제를 해결하기 위해 컨테이너 기술을 활용한 특징 데이터 추출 환경 구성 방안을 제안한다. 이 방법은 개발환경과 실행환경이 동일한 시스템 환경을 가질 수 있도록 동일한 컨테이너 이미지를 생성하고 공유 활용하는 방안이다. 우리는 개발 환경 측면에서는 docker 컨테이너 이미지와 jupyterhub

스케줄러를 활용하였고, 실행환경을 위해서는 계산과학에서 널리 활용되는 배치 기반의 slurm 스케줄러와 컨테이너 이미지 관리의 복잡도를 최소화하기 위해 singularity [6]로 이미지를 변환하여 활용하였다. 개발환경과 실행환경 사이에 동일한 시스템 환경은 물론 동일한 사용자 환경을 갖는 것도 중요하다. 이를 위해 사용자 홈 디렉터리를 개발환경에서 jupyterhub가 실행하는 docker 컨테이너에 마운트하는 것은 물론, 실행환경에서 전체 클러스터와 singularity 컨테이너에 마운트될 수 있도록 구성하였다. 이를 통해 개발환경에서 사용자가 설치한 라이브러리가 실행환경에서 그대로 활용될 수 있다. 그림1은 시스템 구성도를 보여준다.



<그림1> 특징 데이터 추출 자동화 시스템 구성도



<그림2> 특징 데이터 추출 작업 관리 인터페이스

첫 번째 요구사항인 특징 데이터 추출 작업들에 대한 관리를 용이하게 할 수 있도록 사용자는 웹 인터페이스를 통해 특징 데이터 추출 작업을 실행할 수 있다. 특징 데이터 추출 작업의 실행은 몇 개의 CPU 코어 자원을 활용할 것인지와 대상으로 하는 원시 데이터 리스트, 그리고 특징 데이터 추출 코드를 입력으로 한다. 병렬 실행된 특징 데이터 추출 작업 역시 웹 기반 모니터링과 작업 취소가 가능하도록 하여 관리의 편이성을 높였다. 그림2는 웹 기반 특징 데이터 추출 작업 관리 인터페이스의 예를 보여준다.

개발된 시스템은 현재 EDISON [7] 서비스에 적용되어 활용 중에 있다. 우리는 개발된 시스템의 검증을 위해 약 2백만 건의 나노다공성(nanoporous) 소재 데이터로부터 특징 데이터를 추출하였다. 특성 데이터는 메타데이터와 토폴로지 분석을 위한 바코드 형태 특성 데이터를 포함한다. 작업처리 스케줄러의 부하를 낮추기 위해 하나의 배치 작업이 30건의 데이터를 처리하도록 설정하고, 평균 500 코어를 동시에 활용하였다. 각 소재 데이터로부터 특징 데이터를 추출하는데 소요되는 시간은 약 1분 내외의 시간이 소요되었고, 전체적으로 2백만 건 데이터의 처리에 약 3일 정도가 소요되었다. 이는 단일 CPU를 활용하는 경우 4년 이상 소요될 시간을 수백 배 단축한 것이다. 각 데이터의 처리는 서로 상관관계가 없이 독립적 처리가 가능하기 때문에 특징 데이터 추출에 소요되는 시간은 확보한 자원의 수에 반비례한다.

4. 결론

본 논문에서 제안한 특징 데이터 추출 자동화 시스템은 개발환경과 실행환경 이질성 문제를 해결하기 위해 컨테이너 기술을 활용한 특징 데이터 추출 환경 구성 방안을 제안했다는 점과, 사용자가 직접 MPI 병렬 코드를 작성하는 복잡성을 제거하여 쉽고 편리하게 원하는 특징 데이터를 추출할 수 있는 특징 데이터 추출 방안을 제안하였다는 점에서 의미가 있다. 그리고 향후 계산과학 데이터를 기반으로 인공지능 모델을 쉽게 개발하고, 웹 기반으로 쉽게 배포할 수 있는 계산과학, 데이터, 인공지능의 융합 연구환경 등에서 널리 활용될 것으로 기대한다.

* 이 논문은 한국과학기술정보연구원 과제번호(K-20-L02-C05)와 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업(No.NRF-2011-0020576)의 연구비 지원으로 작성되었습니다.

참고문헌

[1] Saal, James E., Anton O. Oliynyk, and Bryce Meredig. "Machine Learning in Materials Discovery: Confirmed Predictions and Their Underlying Approaches." Annual Review of Materials Research 50 (2020).

[2] Dong, Guozhu, and Huan Liu, eds. Feature engineering for machine learning and data analytics. CRC Press, (2018)

[3] Gabriel, Edgar, et al. Open MPI: Goals, concept, and design of a next generation MPI implementation, Springer, Berlin, Heidelberg (2004)

[4] Kluyver, Thomas, et al., Jupyter Notebooks—a publishing format for reproducible computational workflows, ELPUB (2016)

[5] ast https://github.com/python/cpython/blob/master/Lib/ast.py

[6] Kurtzer, Gregory M., Vanessa Sochat, and Michael W. Bauer., Singularity: Scientific containers for mobility of compute, PloS one, 12(5), (2017)

[7] Ahn, Sunil, et al., EDISON-DATA: A flexible and extensible platform for processing and analysis of computational science data, Software: Practice and Experience, 49(10), (2019)