실시간 데이터 예측을 위한 인공지능 분석 방법 연구

홍필두

한국폴리텍대학 분당융합기술교육원

A Study on the Analysis Method of Artificial Intelligence for Real-Time Data Prediction.

Phil-Doo, Hong

Korea Polytechnics, BCTC

E-mail : iamhpd@kopo.ac.kr

요 약

인공지능 분석에서 모델을 만들고 이를 검증하는 과정은 이미 생성된 데이터를 가지고 수행하는 Batch Processing이기에 연산 처리시간이 필요한 작업이다. 우리는 주식이나 국방 정보와 같은 실시간으로 발생하는 데이터를 바로 앞에서 발생한 데이터를 가지고 실시간으로 모델을 세우고 검증하여 예측하는 것이 필요하다. 이를 위한 해결책으로, 인공지능 모델링 작업에 필요한 데이터를 시간 처리 순으로 분할하고 데이터를 여러 프로세스에서 분산 처리하는 기법을 적용하여 해결하였다.

ABSTRACT

In Artificial Intelligence analysis, the process of creating a model and verifying it is a task that requires computational processing time because it is Batch Processing performed with already generated data. We need to model, validate, and predict real-time data, such as stocks and defense information, with data generated directly in front of us. As a solution to this, we solve it by applying techniques to segment the data required for artificial intelligence modeling tasks in order of time processing and distribute the data across multiple processes.

키워드

Artificial Intelligence, Realtime Deep Learning Process,Data Prediction.

Ⅰ. 서 론

인간의 지능으로 할 수 있는 사고, 학습, 자기 개발 등을 컴퓨터가 할 수 있도록 하는 방법을 연구하는 분야[1]인 인공지능 분야를 활용하고자 하는 시도는 많은 분야에서 매우 활발하게 이루어지고 있다. 이러한 인공지능 분야를 좀 더 세분화해 보면 현재의 데이터를 분석 학습하여 얻어지는 예측 모델을 가지고 수치를 예측하고, 사건의 발생 여부를 판단하는 이진 분류를 처리하거나, 또 주어진 데이터를 바탕으로 동식물의 종류를 분류하는 등 다중분류를 진행할 수 있다. 또 텍스트를 분류하거나, 이미지를 인식할 수도 있다. 이러한 Machine Learning이나 Deep Learning의 이론적인 내용과 모델을 찾고 예측하는 알고리즘은 매우 복잡하다고 볼 수 있다.[2] 하지만 최근 등장한 Python과 같은 프로그래밍 언어의 기반에서 활용할 수 있는 add-on library인 TensorFlow, Keras, PyTorch 등을 사용한다면[3], 인공지능 분야를 대중적으로 쉽게 다룰 수 있고, 더욱 많은 인공지능을 응용한 애플리케이션을 활성화할 수 있게 되었다 또 벡터 등 다차원 기반 연산의 특화된 GPU 연산을 활용할 수 있는 CUDA 라이브러리등을 사용한다면 획기적인 연산 실행속도도 얻을 수 있다.

인공지능 분석에서 모델을 만들고 이를 검증하는 과정은 이미 생성된 데이터를 가지로 이루어진다. 해당 모델을 이용하여 발생된 데이터를 모델

에 적용시켜 예측값을 얻는 작업을 한다. 모델을 만들고 이를 검증하는 과정은 일정부분 데이터가 모여야 실행할 수 있는 Batch Processing 이기에 연산처리 시간이 필요한 작업이다. 물론 현재의 하드웨어와 GPU연산이 과거에 비해 처리속도를 현저히 줄여주었다. 우리의 제안은 이를 보다 발전시켜 예를 들어 주식이나 비트코인 국방 정보와 같은 실시간으로 발생하는 데이터를 바로 앞에서 발생된 데이터를 가지고 모델을 세우고 검증하여 예측하고자 하는 노력이다. 이를 위하여 우리는 인공지능 모델링 작업을 시분할로 처리하고 분산처리하는 기법을 적용하여 해결하였다.

이에, 본 논문에서는 2장 인공지능 모델링 예측으로 일반적인 인공지능 분석절차에 대하여 설명하고 3장에서 우리의 시분할 및 분산처리 방식에 대하여 설명한다. 마지막으로 본 논문의 결론을 맺었다.

II. 인공지능 모델링 예측

일반적인 Deep learning 문제를 처리하기 위하여 가장 먼저 고려하여야 하는 부분은 주어진 데이터의 연관성을 설명할 수 있는 AI모델을 찾는 것이다. 이 모델의 깊숙한 내부 지식이 없더라도 이미 잘 만들어진 Deep learning library를 사용하여 쉽게 Deep learning 문제를 처리할 수 있다. 이러한 문제를 처리하는 데 있어서 대부분은 다음의 절차로 처리하고 있다.

첫째, 문제를 분석하고 예측할 자료를 수집하는 일이다. 대부분 데이터는 이미 발생한 데이터를 가져다 사용한다. 때로는 시뮬레이션 등으로 생성할 수도 있다. 이 데이터들을 분석하기 위하여 모델을 훈련하는 training data set, 훈련된 모델이 overfitting또는 underfitting을 판단하여 적절한 정확도를 판단하기 위한 validation data set, 이후 모델이 만들어 진후 해당 모델을 평가하기 위한 test data set으로 나누는 작업을 한다. 그리고 모델 내에서 사용할 수 있도록 reshape, one-hot encoding등 형태를 변환하는 일도 수행한다. 두 번째, 모델을 정의하는 일이다. sequential 처리 후 layer를 추가하여 모델을 구성한다. 기존의 잘 만들어진 모델이 있다면 호출하여 사용하거나 Deep learning Library에서 제공하는 API를 사용할 수도 있다. 세 번째, 학습 과정을 설정하는 과정이다. 학습 결과가 수치 예측, 이진 분류, 다중 분류 등을 지정하고, 손실함수와 최적화 방법을 지정한다. 그리고 모델에 대하여 compile을 실행한다. 네 번째, 모델을 학습시키는 과정이다. 설정된 모델을 training data set을 가지고 학습시킨다. 즉 model fitting 과정을 수행한다. 다섯 번째로 학습 과정의 진행을 모니터링을 한다. training data set과 validation data set이 각각 모델에서 학습되는 동안에 loss rate와 accuracy에 대한 흐름을 보고 모델에 대하여 완성도를 판단하게 된다. 여섯번째, model을 평가 하는 과정이다. 앞과정에서 도출된 모델이 정확한 예측이 가능한지 test data set을 가지고 evaluation을 한다. 최종 검증되었다면 모델을 실제 사용하기 위하여 저장해 둔다. 마지막으로 실제 발생하는 데이터를 가지고 잘 구성되고 검증된 모델에 입력하여 예측값을 얻는 일을 수행하는 모델 활용 단계를 수행하면 된다. 그림 1은 이러한 단계를 그림으로 나타낸 것이다.

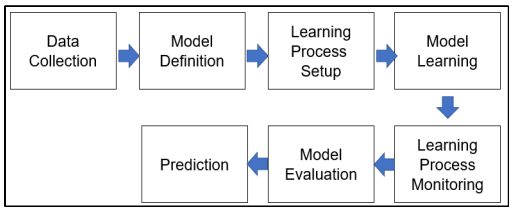

그림1. 모델링 예측 처리 단계

III. 시분할 및 분산처리 제안

우리는 인공지능 모델링 작업을 시분할로 처리하고 분산 처리하는 기법을 적용하고자 하였다. 이는 주식이나 국방 정보와 같은 실시간으로 발생하는 데이터를 바로 앞에서 발생한 데이터를 가지고 모델을 세우고 검증하여 예측하는 데 매우 유용하다. 먼저 인공지능 모델링 처리 시 그림 2와 같이 데이터가 투입되는 시점을 분할 할 수 있다. 모델을 학습하는 시점에 training data set과 validation data set이 투입되어야 하며, 모델을 평가하는 시점에 test data set이 투입되어야 한다. 그 이후, 모델이 학습된 이후 검증된 모델을 가지고 최종적으로 예측하고자 하는 데이터가 투입되어 진다.

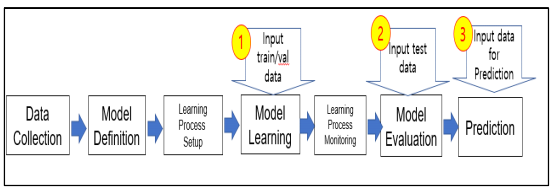

그림2. 모델링 처리시 데이터 투입단계 분할

이러한 데이터 투입 시점에 대하여 데이터 발생 순서에 맞추어 그림3과 같이 데이터를 나누어 모델 처리에 사용하였다. 이러한 시분할 처리를 수행한다면, 가장 최근에 발생한 데이터를 가지고 모델의 학습 및 검증하므로 처리시간의 효율성이 증가할 수 있다.

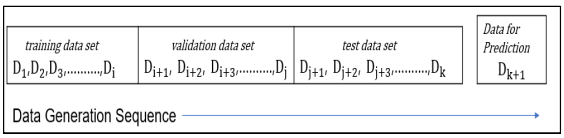

그림3. 데이터 발생순서로 사용될 데이터 분할

데이터 생성시간에 따라 사용될 데이터를 분리하는 방식을 여러 프로세서에서 분산 처리한다면 더 효율적인 처리시간을 얻을 수 있다. 먼저 발생하는 순서에 따라 데이터를 데이터 스풀에 기록한다. 이 데이터 스풀은 n개의 프로세서가 공유하여 사용할 수 있다. 1번 프로세서가 앞서 기술한 데이터 처리를 수행하는 동안 2번 프로세스는 지연된 시간을 가지고 deferred processing 방식으로 처리한다. 이렇게 n번째 프로세서가 처리를 마치면 다시 1번 프로세스가 수행되는 방식으로 처리를 한다면 실시간의 가까운 처리수행을 할 수 있다. 그림 4는 n개의 프로세서로 분산 처리하는 개념을 설명하고 있다.

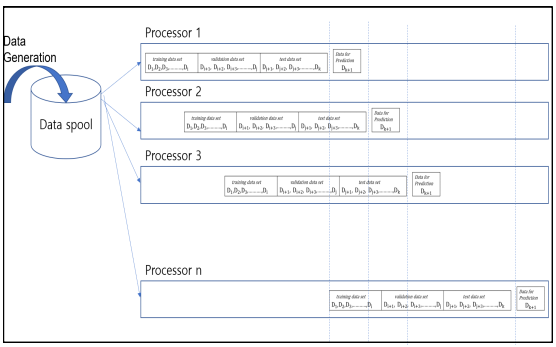

그림4. n개의 프로세스로 분산처리

이처럼 데이터를 나누어 발생하는 순서에 따라 training data set, validation data set, test data set으로 처리하는 방법과 이러한 처리가 이루어지고 있는 순간에 시차를 조금씩 두어 여러 개의 process가 분산 병행하여 처리하는 경우 학습을 통한 모델 완성이 바로 직전 데이터를 통하여 이루어지기 때문에 현재 데이터와 이전 데이터가 많은 연관성을 가지고 있는 시계열 데이터를 처리하는데 매우 유용하다. 즉 현재 시점의 값이 바로 직전에 값과는 밀접한 관계를 갖지만, 시간이 지날수록 관련도가 적은 데이터 처리 분야에서 제안된 방식을 사용한다면 매우 적절한 경우가 될 것이다.

Ⅳ. 결 론

일반적 데이터 처리뿐만 아니라 실시간으로 발생되는 데이터를 활용하여 더 빠른 deep learning 분석을 적용하여야 하는 필요성이 높아지고 있는 바, 데이터를 분할하고 분산처리를 수행한다면, 모델 분석의 처리시간 측면에서 효율성이 높아질 것으로 예상된다. 단 본 제안은 개념모델 수준으로 해당 시스템을 실제로 활용하기 위해서는 보다 정교한 데이터 분류작업과 분산시스템 구현이 필요로 하며, 해당 사안은 향후 과제로 남긴다.

Acknowledgement

본 결과물은 환경부의 재원으로 한국환경산업기술원의 상하수도 혁신 기술개발사업 사업의 지원을 받아 연구되었습니다.(RE202101601)

References

[1] Wikipedia. artificial Intelligence, [https://ko.wikipedia.org/wiki/%EC%9D%B8%EA%B3%B5%EC%A7%80%EB%8A%A5]
[2] Yann LeCun, Yoshua Bengio & Geoffrey Hinton , *Deep learning*, Nature 521, pp. 436-444, 27 May 2015.
[3] Tensorflow, Googlem [http://tensorflow.org]
[3]CUDA Toolkit, NVIDIA, [https://developer.nvidia.com/cuda-toolkit]