



(19) 대한민국특허청(KR)  
(12) 등록특허공보(B1)

(45) 공고일자 2023년01월26일  
(11) 등록번호 10-2491753  
(24) 등록일자 2023년01월19일

(51) 국제특허분류(Int. Cl.)  
G06N 3/08 (2023.01) G06F 16/9032 (2019.01)  
G06F 16/9038 (2019.01) G06F 9/445 (2018.01)  
G06N 20/00 (2019.01) G06N 3/04 (2023.01)  
(52) CPC특허분류  
G06N 3/08 (2023.01)  
G06F 16/9032 (2019.01)  
(21) 출원번호 10-2020-0096813  
(22) 출원일자 2020년08월03일  
심사청구일자 2020년08월03일  
(65) 공개번호 10-2022-0016641  
(43) 공개일자 2022년02월10일  
(56) 선행기술조사문헌  
US20200012892 A1\*  
KR100963885 B1\*  
US20170243132 A1  
KR100978740 B1  
\*는 심사관에 의하여 인용된 문헌

(73) 특허권자  
(주)한국플랫폼서비스기술  
대전광역시 서구 청사로 148, 508호(문산동, 매그  
놀리아)  
(72) 발명자  
이준혁  
대전광역시 서구 청사로 254 등지아파트, 106동  
108호  
(74) 대리인  
황창욱

전체 청구항 수 : 총 13 항

심사관 : 박승철

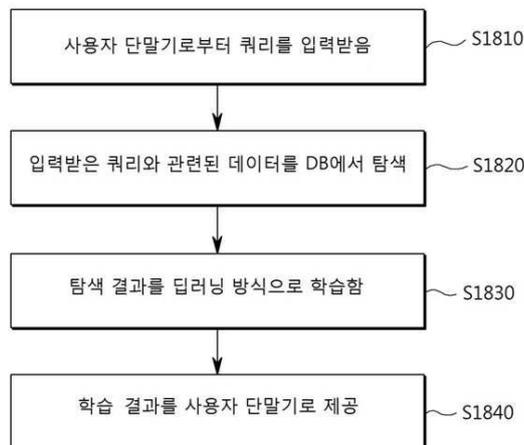
(54) 발명의 명칭 **쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법**

(57) 요약

본 발명은 사용자의 학습 시에 입력된 쿼리에 의해 데이터베이스에 저장된 정보 또는 데이터를 딥러닝 방식으로 학습하여, 쿼리에 대응하는 정보 또는 데이터를 제공할 수 있도록 하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법이 개시된다.

(뒷면에 계속)

대표도 - 도18



개시된 쿼리를 이용한 프레임워크 딥러닝 학습 시스템은, 데이터 셋, 학습 모델, 학습 파라미터 및 학습 결과를 저장하는 데이터베이스; 상기 데이터베이스에 플러그인 방식으로 연결되고, 상기 데이터베이스에 저장된 정보 또는 데이터에 대한 확인, 수정 및 학습을 실행하되, 상기 입력받은 쿼리와 관련된 데이터를 상기 데이터베이스에서 탐색하고, 탐색 결과를 딥러닝 방식으로 학습한 후 딥러닝 학습 결과를 제공하는 학습 프레임워크; 및 상기 학습 프레임워크에 학습을 위한 정보, 명령 및 쿼리(query)를 입력하고, 상기 학습 프레임워크로부터 제공된 딥러닝 학습 결과를 출력하는 입출력 모듈을 포함할 수 있다.

(52) CPC특허분류

*G06F 16/9038* (2019.01)

*G06F 9/44526* (2013.01)

*G06N 20/00* (2021.08)

*G06N 3/04* (2023.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1315001411
과제번호	2019-MOIS34-001-01030000-2020
부처명	행정안전부
과제관리(전문)기관명	국립재난안전연구원
연구사업명	생활안전예방서비스기술개발(R&D)
연구과제명	생활안전 체험교육을 위한 실감형 콘텐츠 기술개발
기 여 율	100/100
과제수행기관명	한국전자통신연구원
연구기간	2020.01.01 ~ 2020.12.31

---

## 명세서

### 청구범위

#### 청구항 1

데이터 셋, 학습 모델, 학습 파라미터 및 학습 결과를 저장하는 데이터베이스;

상기 데이터베이스에 플러그인 방식으로 연결되고, 상기 데이터베이스에 저장된 정보 또는 데이터에 대한 확인, 수정 및 학습을 실행하되, 입력받은 쿼리와 관련된 데이터를 상기 데이터베이스에서 탐색하고, 탐색 결과를 딥러닝 방식으로 학습한 후 딥러닝 학습 결과를 제공하되, 외부 프레임워크와의 호환성을 위해 기존 프레임워크의 기학습된 모델을 가져오거나 상기 데이터베이스에서 정보 또는 데이터를 외부로 내보낼 때 모델 포맷을 이용하는 모델 컨버터를 포함하는 학습 프레임워크; 및

상기 학습 프레임워크에 학습을 위한 정보, 명령 및 쿼리(query)를 입력하고, 상기 학습 프레임워크로부터 제공된 딥러닝 학습 결과를 출력하는 입출력 모듈; 을 포함하고,

상기 학습 프레임워크는, 상기 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 상기 데이터베이스 내에 저장되어 있는 데이터들에 대하여, 주체(Subject), 술부(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공하는 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

#### 청구항 2

제 1 항에 있어서,

상기 데이터 셋은, 동일한 형식을 가진 정보 또는 데이터의 집합이고, 상기 정보 또는 데이터는 숫자, 문자, 이미지, 영상, 음성을 포함하여 기계학습에 사용되는 모든 종류의 정보 또는 데이터인, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

#### 청구항 3

제 1 항에 있어서,

상기 학습 모델은, 기계학습에 사용되는 모델로 입출력과 모델 내부를 정의하는 파라미터 그리고 기계학습, 추론에 필요한 파라미터들로 구성되고, 상기 데이터베이스에 관계형 데이터 형식으로 저장되며, 모델 컨버터를 통하여 다른 학습 프레임워크로 변환 가능한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

#### 청구항 4

제 1 항에 있어서,

상기 데이터베이스는 기계학습, 기계추론에 사용된 모든 입출력 데이터를 저장하고, 기계학습, 기계추론에 사용된 모델을 저장하며, 사용자의 쿼리 요청에 대응하는 프로시저(procedure)를 제공하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

#### 청구항 5

제 4 항에 있어서,

상기 프로시저는, 인서트 네트워크(Insert Network), 인서트 레이어(Insert Layer), 메이크 프로젝트(Make Project), 입력 데이터 로더(Input Data Loader), 트레인(Train), 모델 저장(Save Model) 및 테스트(Test)를 포함하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

**청구항 6**

삭제

**청구항 7**

제 1 항에 있어서,

상기 모델 컨버터는, 상기 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 상기 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 상기 데이터베이스의 네트워크 모델을 ONNX 모델 포맷으로 변환하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

**청구항 8**

제 1 항에 있어서,

상기 데이터베이스는, 데이터셋 테이블(Dataset Table), 네트워크 테이블(Network Table), 프로젝트 테이블(Project Table), 작업 테이블(Job Table), 공통 테이블(Common Table)을 포함하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

**청구항 9**

제 1 항에 있어서,

상기 학습 프레임워크는, 사용자 단말기로부터 학습 쿼리가 입력되면(Call Train), 네트워크 초기화(Init Network), 네트워크 구성(Construct Network), 네트워크 갱신(Update Network)을 실행하며, 모든 층에 대한 초기화(Initialize all layers)가 이루어지면 학습(Train)을 실행하고, 학습 종료시까지 배치 데이터를 획득하고(Get Batch Data) 반복(Iteration)하여 결과 및 모델을 저장하며(Store Result & Model), 학습 종료시 학습 결과를 제공하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

**청구항 10**

제 1 항에 있어서,

상기 학습 프레임워크는, 상기 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 상기 데이터베이스 내에서 해당 분류에 해당하는 데이터를 탐색하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템.

**청구항 11**

사용자 단말기 및 데이터베이스와 연동하는 프레임워크의 쿼리를 이용한 프레임워크 딥러닝 학습 방법으로서,

- (a) 상기 프레임워크가 상기 사용자 단말기로부터 쿼리(query)를 입력받는 단계;
- (b) 상기 프레임워크가 상기 입력받은 쿼리와 관련된 데이터를 상기 데이터베이스에서 탐색하는 단계;
- (c) 상기 프레임워크가 상기 탐색 결과를 딥러닝 방식으로 학습하는 단계; 및

(d) 상기 프레임워크가 상기 딥러닝 학습 결과를 상기 사용자 단말기로 제공하는 단계; 를 포함하고,

상기 (a) 단계에서 상기 프레임워크는, 외부 프레임워크와의 호환성을 위해 모델 컨버터를 사용하고, 기존 프레임워크의 기학습된 모델을 가져오거나, 상기 데이터베이스에서 정보 또는 데이터를 외부로 내보낼 때 상기 모델 컨버터를 통하여 모델 포맷으로 변환하는, 쿼리를 이용한 프레임워크 딥러닝 학습 방법.

**청구항 12**

제 11 항에 있어서,

상기 (b) 단계에서 상기 프레임워크는, 상기 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 상기 데이터베이스 내에 해당 분류에 해당하는 데이터를 탐색하는, 쿼리를 이용한 프레임워크 딥러닝 학습 방법.

**청구항 13**

제 11 항에 있어서,

상기 (b) 단계에서 상기 프레임워크는, 상기 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 상기 데이터베이스 내에 저장되어 있는 데이터들을 주체(Subject), 서술(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공하는, 쿼리를 이용한 프레임워크 딥러닝 학습 방법.

**청구항 14**

삭제

**청구항 15**

제 11 항에 있어서,

상기 모델 컨버터는, 상기 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 상기 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 상기 데이터베이스의 네트워크 모델을 상기 모델 포맷으로 변환하는, 쿼리를 이용한 프레임워크 딥러닝 학습 방법.

**발명의 설명**

**기술 분야**

[0001] 본 발명은 쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법에 관한 것으로서, 더욱 자세하게는 딥러닝에 관한 전문적 지식이 없는 사용자라도 어려움 없이 사용자의 학습에 필요한 정보를 제공할 수 있도록 프레임워크가 데이터베이스에 플러그인 형태로 연결되어, 사용자의 학습 시에 입력된 쿼리에 의해 데이터베이스에 저장된 정보 또는 데이터를 딥러닝 방식으로 학습하여, 쿼리에 대응하는 정보 또는 데이터를 제공할 수 있도록 하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법에 관한 것이다.

**배경 기술**

[0002] 일반적으로 기계학습의 경우 AI 기술이 발전함에 따라 다양한 형태로 제공되고 있으며, 그 중 딥러닝 기법이 대

표적인 기계학습의 방법으로 대두되고 있다.

- [0003] 이와 같은 딥러닝 기계학습 방법은 반복적인 학습을 통하여 그 정확도를 높여가는 방법으로, 결과값의 정확도를 높여갈수록 발생하는 데이터의 양도 방대해져 빅데이터를 처리하는데 다양한 부대시설 등이 발생하게 된다.
- [0004] 이에 따라, 분산처리 및 분산저장과 같은 기술들이 제안되고 있으며, 또한 각각의 운영체제 별로 다양한 형태의 학습 프로그램 등을 제공하게 된다.
- [0005] 이와 같은 현상으로 인하여 사용하는 AI 학습 프레임워크에 따라 사용자가 호환 작업을 해야 하거나, 또는 사용하는 학습데이터 도메인/알고리즘에 따른 학습 프레임워크를 별도로 구동해야 하는 불편함이 있었다.
- [0006] 또한, 다양한 이종 AI 학습 프레임워크들은 입력 데이터를 표현하는 데이터셋의 포맷 기준이 다르며, 이를 호환하기 위하여 별도의 데이터 컨버트 툴을 사용하거나 NoSQL 같은 대용량 파일관리의 분산 서비스를 활용하기도 한다.
- [0007] 이러한 문제점을 극복하기 위해 다양한 방법들이 시도되고 있으나, 단순히 공통적으로 사용되는 Open Source 트렌드를 반영하여 입력 데이터셋의 포맷을 호환하는 방법 등이 대부분이었으며, 사용자의 지식수준에 따라 이와 같은 호환 작업에서도 질적인 차이가 발생하여 기계학습의 결과가 달라지는 경우가 빈번하게 발생하고 있다.
- [0008] 대용량 학습 데이터(정형 또는 비정형)의 관리와 사용자의 PC 및 서버 장비의 폴더 및 디렉터리 구조로 작업을 진행하고 있으며, 이는 데이터의 무결성과 중복성 그리고 불필요한 하드웨어 저장 공간이 늘어나는 상황을 야기시킨다.
- [0009] 단순히 프로그램별 매칭을 통하여 호환하는 방법 등이 대부분이었으며, 사용자의 지식수준에 따라 이와 같은 호환 작업에서도 질적인 차이가 발생하여 기계학습의 결과가 달라지는 경우가 빈번하게 발생하고 있다.
- [0010] 이에 따라 대용량의 AI 학습 프레임워크는 결국 질의응답을 위한 요청사항인 쿼리(query)를 통하여 학습 입력 데이터를 관리(생성 및 조합)하게 되는 것이고, 이와 같은 쿼리는 DBMS에 데이터(DB)를 학습데이터 셋으로 입력하거나 출력할 때 사용하게 된다.
- [0011] 그러나, 앞서 말한 쿼리의 경우 그 사용상 작성이 매우 복잡할 수 있어, 일반적인 사용자가 사용하는데 난해한 문제가 발생할 수 있으며, 또한, 초기 질의응답에 대한 검색어 등을 잘못 입력할 경우 그 결과물의 질적 저하 및 작업시간이 길어져 비효율적인 문제가 발생할 수 있다.
- [0012] 또한, 다양한 이기종의 학습 프레임워크를 사용하는데 그 호환성의 문제 등이 야기되어 그 사용이 매우 복잡한 문제가 있다.

**선행기술문헌**

**특허문헌**

- [0013] (특허문헌 0001) 한국 등록특허공보 제10-2058124호(등록일 : 2019년12월16일)

**발명의 내용**

**해결하려는 과제**

- [0014] 전술한 문제점을 해결하기 위한 본 발명의 목적은, 딥러닝에 관한 전문적 지식이 없는 사용자라도 어려움 없이 사용자의 학습에 필요한 정보를 제공할 수 있도록 프레임워크가 데이터베이스에 플러그인 형태로 연결되어, 사용자의 학습 시에 입력된 쿼리에 의해 데이터베이스에 저장된 정보 또는 데이터를 딥러닝 방식으로 학습하여, 쿼리에 대응하는 정보 또는 데이터를 제공할 수 있도록 하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법을 제공함에 있다.

**과제의 해결 수단**

- [0015] 전술한 목적을 달성하기 위한 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템은, 데이터 셋, 학습 모델, 학습 파라미터 및 학습 결과를 저장하는 데이터베이스; 상기 데이터베이스에 플러그인 방식으로 연결되고, 상기 데이터베이스에 저장된 정보 또는 데이터에 대한 확인, 수정 및 학습을 실행하되, 상기 입

력받은 쿼리와 관련된 데이터를 상기 데이터베이스에서 탐색하고, 탐색 결과를 딥러닝 방식으로 학습한 후 딥러닝 학습 결과를 제공하는 학습 프레임워크; 및 상기 학습 프레임워크에 학습을 위한 정보, 명령 및 쿼리(query)를 입력하고, 상기 학습 프레임워크로부터 제공된 딥러닝 학습 결과를 출력하는 입출력 모듈을 포함할 수 있다.

- [0016] 상기 학습 프레임워크는, 상기 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 상기 데이터베이스 내에 저장되어 있는 데이터들에 대하여, 주체(Subject), 술부(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공할 수 있다.
- [0017] 상기 학습 프레임워크는, 상기 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 상기 데이터베이스 내에서 해당 분류에 해당하는 데이터를 탐색할 수 있다.
- [0018] 상기 학습 프레임워크는, 상기 사용자 단말기로부터 학습 쿼리가 입력되면(Call Train), 네트워크 초기화(Init Network), 네트워크 구성(Construct Network), 네트워크 갱신(Update Network)을 실행하며, 모든 층에 대한 초기화(Initialize all layers)가 이루어지면 학습(Train)을 실행하고, 학습 종료시까지 배치 데이터를 획득하고(Get Batch Data) 반복(Iteration)하여 결과 및 모델을 저장하며(Store Result & Model), 학습 종료시 학습 결과를 제공할 수 있다.
- [0019] 상기 데이터 셋은, 동일한 형식을 가진 정보 또는 데이터의 집합이고, 상기 정보 또는 데이터는 숫자, 문자, 이미지, 영상, 음성을 포함하여 기계학습에 사용되는 모든 종류의 정보 또는 데이터일 수 있다.
- [0020] 상기 학습 모델은, 기계학습에 사용되는 모델로 입출력과 모델 내부를 정의하는 파라미터 그리고 기계학습, 추론에 필요한 파라미터들로 구성되고, 상기 데이터베이스에 관계형 데이터 형식으로 저장되며, 모델 컨버터를 통하여 다른 학습 프레임워크로 변환 가능할 수 있다.
- [0021] 상기 데이터베이스는 기계학습, 기계추론에 사용된 모든 입출력 데이터를 저장하고, 기계학습, 기계추론에 사용된 모델을 저장하며, 사용자의 쿼리 요청에 대응하는 프로시저(procedure)를 제공할 수 있다.
- [0022] 상기 프로시저는, 인서트 네트워크(Insert Network), 인서트 레이어(Insert Layer), 메이크 프로젝트(Make Project), 입력 데이터 로더(Input Data Loader), 트레인(Train), 모델 저장(Save Model) 및 테스트(Test)를 포함할 수 있다.
- [0023] 상기 학습 프레임워크는, ONNX(Open Neural Network Exchange), NNEF(Neural Network Exchange Format) 및 학습 파라미터와 바이어스를 포함하는 모델 Weight 파일을 구조화된 포맷으로 импорт(Import) 및 익스포트(Export) 할 수 있다.
- [0024] 상기 학습 프레임워크는, ONNX, NNEF, CSV 외 다양한 포맷을 확장하여 지원할 수 있다.
- [0025] 상기 학습 프레임워크는, 외부 프레임워크와의 호환성을 위해 사용되고, 기존 프레임워크의 기학습된 모델을 가져오거나 상기 데이터베이스에서 정보 또는 데이터를 외부로 내보낼 때 모델 포맷을 이용하는 모델 컨버터를 포함할 수 있다.
- [0026] 상기 모델 컨버터는, 상기 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 상기 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 상기 데이터베이스의 네트워크 모델을 상기 모델 포맷으로 변환할 수 있다.
- [0027] 상기 데이터베이스는, 데이터셋 테이블(Dataset Table), 네트워크 테이블(Network Table), 프로젝트 테이블(Project Table), 작업 테이블(Job Table), 공통 테이블(Common Table)을 포함할 수 있다.
- [0028] 한편, 전술한 목적을 달성하기 위한 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 방법은, 사용자 단말기 및 데이터베이스와 연동하는 프레임워크의 쿼리 기반 프레임워크 딥러닝 학습 방법으로서, (a) 상기 프레임워크가 상기 사용자 단말기로부터 쿼리(query)를 입력받는 단계; (b) 상기 프레임워크가 상기 입력받은 쿼리와 관련된 데이터를 상기 데이터베이스에서 탐색하는 단계; (c) 상기 프레임워크가 상기 탐색 결과를

딥러닝 방식으로 학습하는 단계; 및 (d) 상기 프레임워크가 상기 딥러닝 학습 결과를 상기 사용자 단말기로 제공하는 단계를 포함할 수 있다.

[0029] 상기 (b) 단계에서 상기 프레임워크는, 상기 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 상기 데이터베이스 내에 해당 분류에 해당하는 데이터를 탐색할 수 있다.

[0030] 상기 (b) 단계에서 상기 프레임워크는, 상기 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 상기 데이터베이스 내에 저장되어 있는 데이터들을 주체(Subject), 서술(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공할 수 있다.

[0031] 상기 (a) 단계에서 상기 프레임워크는, 외부 프레임워크와의 호환성을 위해 모델 컨버터를 사용하고, 기존 프레임워크의 기학습된 모델을 가져오거나, 상기 데이터베이스에서 정보 또는 데이터를 외부로 내보낼 때 상기 모델 컨버터를 통하여 모델 포맷으로 변환할 수 있다.

[0032] 상기 모델 컨버터는, 상기 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 상기 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 상기 데이터베이스의 네트워크 모델을 상기 모델 포맷으로 변환할 수 있다.

### 발명의 효과

[0033] 본 발명에 의하면, 쿼리 기반의 학습 기술을 이용함으로써 프레임워크가 데이터베이스에 플러그인 형태로 연결되어 사용자의 요청 쿼리에 의해서 데이터베이스에 저장된 데이터를 사용하여 학습, 추론 등을 수행할 수 있다.

[0034] 따라서, 딥러닝에 관한 전문적 지식이 없는 사용자라도 어려움 없이 필요한 정보를 용이하게 제공받을 수 있다.

[0035] 또한, 현재 실행 또는 대기 중인 학습 계획의 학습 파라미터를 확인할 수 있으며, 현재 실행 중인 학습 계획의 중간 및 결과를 확인할 수 있다.

[0036] 또한, 현재 실행 중인 학습 계획을 멈출 수 있으며, 대기 중인 학습 계획을 시작할 수 있다.

[0037] 또한, 등록된 네트워크 모델과 데이터 셋을 설정하여 학습 계획을 생성하고, 실행된 학습의 결과를 확인할 수 있다.

### 도면의 간단한 설명

[0038] 도 1은 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템의 전체적인 구성을 개략적으로 나타낸 구성도이다.

도 2는 본 발명의 실시 예에 따른 쿼리를 이용한 학습 기술의 수행 흐름을 나타낸 흐름도이다.

도 3은 본 발명의 실시 예에 따른 데이터베이스의 내부 데이터 흐름을 나타낸 도면이다.

도 4는 본 발명의 실시 예에 따른 쿼리를 이용한 딥러닝 학습 방법을 설명하기 위한 동작 흐름도를 나타낸 도면이다.

도 5는 본 발명의 실시 예에 따른 학습 프로시저의 실행 흐름을 나타낸 도면이다.

도 6은 본 발명의 실시 예에 따른 추론 프로시저의 실행 흐름을 나타낸 도면이다.

도 7은 본 발명의 실시 예에 따른 모델 컨버터의 변환 동작을 나타낸 도면이다.

도 8은 본 발명의 실시 예에 따른 QML 프레임워크의 내부 구조를 개략적으로 나타낸 도면이다.

도 9는 본 발명의 실시 예에 따른 학습 프레임워크에서 제공하는 메인 화면의 한 예를 나타낸 도면이다.

도 10은 본 발명의 실시 예에 따른 학습 프레임워크에서 제공하는 네트워크 생성 기능 화면을 나타낸 도면이다.

도 11은 본 발명의 실시 예에 따른 학습 프레임워크에서 네트워크 레이어를 추가하는 화면을 나타낸 도면이다.

도 12는 본 발명의 실시 예에 따른 학습 프레임워크에서 네트워크 레이어의 추가시 파라미터 추가/수정/삭제 화면을 나타낸 도면이다.

도 13은 본 발명의 실시 예에 따른 학습 프레임워크에서 네트워크 레이어의 추가시 레이어 입력 정보를 추가하는 화면을 나타낸 도면이다.

도 14는 본 발명의 실시 예에 따른 학습 프레임워크의 학습 생성 화면을 나타낸 도면이다.

도 15는 본 발명의 실시 예에 따른 학습 프레임워크의 학습 실행 화면을 나타낸 도면이다.

도 16은 본 발명의 실시 예에 따른 학습 프레임워크의 학습 관리 화면을 나타낸 도면이다.

도 17은 본 발명의 실시 예에 따른 학습 프레임워크의 로그 관리 및 자원 관리 화면을 나타낸 도면이다.

도 18은 본 발명의 실시 예에 따른 학습 프레임워크의 쿼리를 이용한 프레임워크 딥러닝 학습 방법을 설명하기 위한 동작 흐름도를 나타낸 도면이다.

도 19는 본 발명의 실시 예에 따른 학습 프레임워크에서 입력받은 쿼리를 이용하여 데이터베이스의 내부에 있는 데이터들을 RDF 모델로 생성하는 예를 나타낸 도면이다.

도 20은 본 발명의 실시 예에 따른 학습 프레임워크에서 RDF 모델을 생성 또는 병합하는 예를 나타낸 도면이다.

**발명을 실시하기 위한 구체적인 내용**

[0039] 이하, 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다.

[0040] 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 동일 또는 유사한 구성요소에 대해서는 동일한 참조 부호를 붙이도록 한다.

[0041] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우 뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0042] 제1, 제2 및 제3 등의 용어들은 다양한 부분, 성분, 영역, 층 및/또는 섹션들을 설명하기 위해 사용되나 이들에 한정되지 않는다. 이들 용어들은 어느 부분, 성분, 영역, 층 또는 섹션을 다른 부분, 성분, 영역, 층 또는 섹션과 구별하기 위해서만 사용된다. 따라서, 이하에서 서술하는 제1 부분, 성분, 영역, 층 또는 섹션은 본 발명의 범위를 벗어나지 않는 범위 내에서 제2 부분, 성분, 영역, 층 또는 섹션으로 언급될 수 있다.

[0043] 여기서 사용되는 전문 용어는 단지 특정 실시예를 언급하기 위한 것이며, 본 발명을 한정하는 것을 의도하지 않는다. 여기서 사용되는 단수 형태들은 문구들이 이와 명백히 반대의 의미를 나타내지 않는 한 복수 형태들도 포함한다. 명세서에서 사용되는 "포함하는"의 의미는 특정 특성, 영역, 정수, 단계, 동작, 요소 및/또는 성분을 구체화하며, 다른 특성, 영역, 정수, 단계, 동작, 요소 및/또는 성분의 존재나 부가를 제외시키는 것은 아니다.

[0044] 다르게 정의하지는 않았지만, 여기에 사용되는 기술용어 및 과학용어를 포함하는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 일반적으로 이해하는 의미와 동일한 의미를 가진다. 보통 사용되는 사전에 정의된 용어들은 관련 기술문헌과 현재 개시된 내용에 부합하는 의미를 가지는 것으로 추가 해석되고, 정의되지 않는 한 이상적이거나 매우 공식적인 의미로 해석되지 않는다.

[0045] 어플리케이션에서 사용되는 바와 같이, "컴포넌트"와 "시스템"이라는 용어들은, 하드웨어, 하드웨어와 소프트웨어의 조합, 소프트웨어, 또는 실행하는 소프트웨어인 컴퓨터 관련 개체를 언급하려 의도된다. 예를 들어, 컴포넌트는 프로세서에서 실행하는 프로세스, 프로세서, 객체, 실행파일, 실행의 스레드(thread), 프로그램, 및/또는 컴퓨터일 수 있지만, 이에 제한되지는 않는다. 예로서, 서버에서 실행하는 어플리케이션과 서버 모두가 컴포넌트일 수 있다. 한 개 이상의 컴포넌트들은 프로세스 및/또는 실행의 스레드 내에 존재할 수 있고, 컴포넌트는 한 개의 컴퓨터에 로컬화될 수 있고, 및/또는 2개 이상의 컴퓨터들 간에 분산될 수 있다.

[0046] 본 명세서에 사용된 바와 같이, "추론하다(infer)" 또는 "추론(inference)"이라는 용어는 일반적으로 이벤트들 및/또는 데이터를 통해 캡처된 것처럼 관찰의 집합으로부터 시스템, 환경, 및/또는 사용자의 상태들에 대해 추

론하거나 추정하는 프로세스를 일컫는다. 추론은 특정 컨텍스트나 액션을 식별하기 위해 채택될 수 있거나, 또는 예를 들어, 상태들에 대해 확률 분포를 생성할 수 있다. 추론은 확률적일 수 있는데, 즉, 데이터와 이벤트들을 고려하는 것에 기반한 관심 상태들에 대한 확률 분포의 계산일 수 있다. 추론은 또한 이벤트들 및/또는 데이터의 집합으로부터 높은 레벨의 이벤트들을 구성하기 위해 채택되는 기술들을 일컫을 수 있다. 그런 추론은, 이벤트들이 가까운 시간 근접성에 상관되는지 아닌지의 여부, 및 이벤트들과 데이터가 1개 또는 여러 이벤트와 데이터 소스들로부터 오는지에 따라, 관찰된 이벤트들 및/또는 저장된 이벤트 데이터의 집합으로부터 새로운 이벤트들이나 액션들을 구성하는 결과를 가져온다.

- [0047] 이하, 첨부한 도면을 참조하여 본 발명의 실시 예에 대하여 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시 예에 한정되지 않는다.
- [0048] 도 1은 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템의 전체적인 구성을 개략적으로 나타낸 구성도이다.
- [0049] 도 1을 참조하면, 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 쿼리를 이용한 딥러닝 학습 기술을 적용하기 위해, 데이터베이스(DB, 110), 학습 프레임워크(QML, 120) 및 입출력 모듈(130)을 포함한다.
- [0050] 여기서, 입출력 모듈(130)은 시스템의 내부에 자체적인 인터페이스 모듈 형태로 구비될 수 있으나, 도 1에 도시된 바와 같이 사용자 단말기(130)로 구현할 수 있다. 본 발명의 실시 예에서는 입출력 모듈로써 사용자 단말기(130)를 예를 들어 설명하기로 한다. 입출력 모듈(130)이 시스템의 내부에 자체적인 인터페이스 모듈인 경우에는 입력 장치와 출력 장치를 각각 별개로 구비할 수 있으며, 입출력 장치가 하나로 구현되는 경우에는 예를 들면, 터치 인터페이스 모듈로도 구현할 수 있다.
- [0051] 쿼리를 이용하는 딥러닝 학습 기술은, 학습 프레임워크(120)가 데이터베이스(110)에 플러그인 형태로 연결되어 사용자의 요청 쿼리에 의해서 데이터베이스(110)에 저장된 데이터를 사용하여 딥러닝 학습을 수행하는 기술이다.
- [0052] 데이터베이스(110)는 데이터 셋(Data Set), 학습 모델, 학습 파라미터 및 학습 결과를 저장할 수 있다.
- [0053] 데이터 셋은, 동일한 형식을 가진 정보 또는 데이터의 집합이다. 정보 또는 데이터는 숫자, 문자, 이미지, 영상, 음성을 포함하여 기계학습에 사용되는 모든 종류의 정보 또는 데이터일 수 있다.
- [0054] 학습 모델은, 기계학습에 사용되는 모델로 입출력과 모델 내부를 정의하는 파라미터 그리고 기계학습, 추론에 필요한 파라미터들로 구성되고, 관계형 데이터 형식으로 데이터베이스에 저장되며, 모델 컨버터를 통하여 다른 학습 프레임워크로 변환 가능할 수 있다.
- [0055] 학습 모델은, 사용자에게 의해 입력된 텍스트를 인식할 수 있고, 이미지, 오디오 및 동영상에 포함된 음성 및 텍스트를 인식할 수 있다. 또한, 인식한 음성 및 텍스트로부터 사용자 의향을 분석할 수 있다.
- [0056] 학습 결과는 기계학습이 진행되는 동안 중간 출력값이 데이터베이스(110)에 저장되어 사용자가 확인할 수 있다. 기계학습이 진행되는 동안 모델 파라미터 값이 데이터베이스(110)에 저장되어 사용자가 확인할 수 있다. 기계학습이 진행되는 동안 계산되는 모델의 평가지표 값이 데이터베이스(110)에 저장되어 사용자가 확인할 수 있다. 기계추론 결과값이 데이터베이스(110)에 저장되어 사용자가 확인할 수 있다.
- [0057] 데이터베이스(110)는, 데이터셋 테이블(Dataset Table), 네트워크 테이블(Network Table), 프로젝트 테이블(Project Table), 작업 테이블(Job Table), 공통 테이블(Common Table)을 포함할 수 있다.
- [0058] 데이터베이스(110)는 기계학습, 기계추론에 사용된 모든 입출력 데이터를 저장하고, 기계학습, 기계추론에 사용된 모델을 저장하며, 사용자의 쿼리 요청에 대응하는 프로시저(procedure)를 제공할 수 있다. 프로시저는, 인서트 네트워크(Insert Network), 인서트 레이어(Insert Layer), 메이크 프로젝트(Make Project), 입력 데이터 로더(Input Data Loader), 트레인(Train), 모델 저장(Save Model) 및 테스트(Test)를 포함할 수 있다.
- [0059] 학습 프레임워크(120)는, 사용자 단말기(130)로부터 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 데이터베이스(110) 내에서 해당 분류에 해당하는 데이터를 탐색할 수

있다.

- [0060] 학습 프레임워크(120)는, 사용자 단말기(130)로부터 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 데이터베이스(110) 내에 저장되어 있는 데이터들에 대하여, 주체(Subject), 술부(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공할 수 있다.
- [0061] 학습 프레임워크(120)는, ONNX(Open Neural Network Exchange), NNEF(Neural Network Exchange Format) 및 학습 파라미터와 바이어스를 포함하는 모델 Weight 파일을 구조화된 포맷으로 импорт(Import) 및 익스포트(Export) 할 수 있다.
- [0062] 학습 프레임워크(120)는, ONNX, NNEF, CSV 외 다양한 포맷을 확장하여 지원할 수 있다.
- [0063] 학습 프레임워크(120)는, 외부 프레임워크와의 호환성을 위해 사용되고 기존 프레임워크의 기학습된 모델을 가져오거나 상기 데이터베이스에서 정보 또는 데이터를 외부로 내보낼 때 모델 포맷을 이용하는 모델 컨버터를 포함할 수 있다.
- [0064] 모델 컨버터는, 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 데이터베이스의 네트워크 모델을 모델 포맷으로 변환할 수 있다.
- [0065] 학습 프레임워크(120)는 데이터베이스(110)에 플러그인 방식으로 연결되고, 데이터베이스(110)에 저장된 정보 또는 데이터에 대한 확인, 수정 및 새로운 데이터를 추가할 수 있다. 학습 프레임워크(120)는 예를 들면, QML을 예로 들 수 있다.
- [0066] QML은 데이터베이스(110)에 플러그 인으로 설치된 학습 프레임워크로서 데이터베이스의 호출로 실행된다. QML은 호출될 때 데이터베이스(110)로부터 각종 데이터를 인자로 받아 수행결과를 반환한다. QML은 관계형 데이터 포맷으로 정의된 네트워크 모델을 해석하여 프레임워크 내부에 네트워크를 구성한다. QML은 데이터베이스(110)로부터 학습 파라미터와 학습 데이터를 인자로 받아 프레임워크 내부에 구성된 네트워크의 학습을 수행하고 학습 결과를 반환한다. QML은 데이터베이스(110)로부터 입력 데이터를 인자로 받아 프레임워크 내부에 구성된 네트워크를 이용하여 기계추론을 수행하고 결과를 반환한다.
- [0067] 또한, 학습 프레임워크(120)는, 사용자 단말기(130)로부터 쿼리를 입력받으면, 데이터베이스(110)에 저장된 학습 네트워크 모델에 대한 확인, 수정, 및 새로운 학습을 위한 학습 네트워크 모델을 생성하며, 입력된 쿼리에 따라 정보 또는 데이터와 학습 네트워크 모델을 선택하고 학습 파라미터를 설정하여 기계학습을 실행하고, 학습 중간 결과 및 최종 결과를 제공하며, 입력된 쿼리를 통하여 데이터와 기 학습된 학습 모델을 선택하여 기계추론을 실행하고, 그 추론 결과를 제공할 수 있다.
- [0068] 학습 프레임워크(120)는, 사용자 단말기(130)로부터 학습 쿼리가 입력되면(Call Train), 네트워크 초기화(Init Network), 네트워크 구성(Construct Network), 네트워크 갱신(Update Network)을 실행하며, 모든 층에 대한 초기화(Initialize all layers)가 이루어지면 학습(Train)을 실행하고, 학습 종료시까지 배치 데이터를 획득하고(Get Batch Data) 반복(Iteration)하여 결과 및 모델을 저장하며(Store Result & Model), 학습 종료시 학습 결과를 제공할 수 있다.
- [0069] 사용자 단말기(130)는 학습 프레임워크(120)를 통하여 쿼리를 입력하고, 데이터베이스(110)로부터 학습 프레임워크(120)를 통하여 쿼리에 대응된 학습 결과를 수신할 수 있다.
- [0070] 또한, 사용자 단말기(130)는 쿼리를 통해서 데이터베이스(110)에 여러 기능들을 요청하고, 데이터베이스(110)로부터 결과를 응답 받는다. 사용자 단말기(130)는 쿼리를 통해서 데이터베이스(110)에 저장된 데이터를 확인, 수정하고, 새로운 데이터를 추가한다. 사용자 단말기(130)는 쿼리를 통해서 데이터베이스(110)에 저장된 네트워크 모델을 확인, 수정하고 새로운 학습을 위한 네트워크 모델을 생성한다. 사용자 단말기(130)는 쿼리를 통해서 데이터와 학습 네트워크 모델을 선택하고 학습 파라미터를 설정하여 기계학습을 요청하고, 학습 중간 결과 및 최종 결과를 확인한다. 사용자 단말기(130)는 쿼리를 통해서 데이터와 기학습 된 네트워크 모델을 선택하여 기계추론을 요청하고, 추론 결과를 확인할 수 있다.

- [0071] 도 2는 본 발명의 실시 예에 따른 쿼리를 이용한 딥러닝 학습 기술의 수행 흐름을 나타낸 흐름도이다.
- [0072] 도 2를 참조하면, 본 발명의 실시 예에 따른 쿼리를 이용하는 딥러닝 학습 기술은, ONNX 포맷으로 변환된 기학습된 모델을 컨버터를 통해서 QML 포맷으로 저장하고, 사용자 단말기(130)로부터 학습 또는 추론 쿼리를 입력받으며, 데이터베이스(110)에서 데이터 또는 정보를 QML로 전달하여 학습 및 추론을 수행한다. 그리고, 학습 또는 추론 결과를 데이터베이스(110)에 저장하면, 사용자 단말기(130)가 데이터베이스(110)에 저장된 결과를 확인한다.
- [0073] 먼저 학습 프레임워크(120)는 학습 모델을 임포트(Import)하거나, 익스포트(Export) 할 수 있다(①).
- [0074] 즉, 학습 프레임워크(120)는, ONNX(Open Neural Network Exchange), NNEF(Neural Network Exchange Format) 및 학습 파라미터와 바이어스를 포함하는 모델 Weight 파일을 구조화된 포맷으로 임포트(Import) 및 익스포트(Export) 할 수 있다.
- [0075] 또한, 학습 프레임워크(120)는, ONNX, NNEF, CSV 외 다양한 포맷을 확장하여 지원할 수 있다.
- [0076] 또한, 학습 프레임워크(120)는, 학습 모델을 임포트 또는 익스포트 할 때, 모델 컨버터를 통하여 데이터베이스(110)의 스키마 구조에 맞게 변환한다(②).
- [0077] 또한, 데이터베이스(110)는 쿼리를 해석하여 적절한 작업을 수행한다(③).
- [0078] 또한, 학습 프레임워크(120)는 데이터베이스(110)에 플러그인을 수행하고, 데이터베이스(110)로부터 받은 정보를 통해 학습 및 추론을 수행한다(④).
- [0079] 또한, 사용자 단말기(130)는 쿼리를 통해 데이터베이스(110)로 학습 또는 추론을 요청할 수 있다(⑤).
- [0080] 또한, 사용자 단말기(130)는 데이터베이스(110)의 테이블을 조회하여 학습 관련 정보를 조회할 수 있다(⑥).
- [0081] 그리고, 모델 데이터는 데이터베이스(110)에 QML 스키마로 저장된다(⑦).
- [0082] 도 3은 본 발명의 실시 예에 따른 데이터베이스의 내부 데이터 흐름을 나타낸 도면이다.
- [0083] 도 3을 참조하면, 본 발명의 실시 예에 따른 데이터베이스(110)는, 기계학습과 관련된 데이터들을 저장하고, 기계학습에 필요한 기능들을 프로시저(Procedure)로 제공하여 사용자 요청에 의해 기계학습을 수행한다.
- [0084] 데이터베이스(110)에서, 테이블(Table)은 크게 데이터셋 테이블(Dataset Table), 네트워크 테이블(Network Table), 프로젝트 테이블(Project Table), 작업 테이블(Job Table), 공통 테이블(Common Table)을 포함할 수 있다.
- [0085] 데이터 셋은 데이터 종류를 포함하고, 네트워크는 네트워크 종류, lenet을 포함하며, 프로젝트는 네트워크의 정보를 복사해 학습 또는 추론 작업을 진행한다. 작업 테이블은 사용자 정보, 프로젝트의 상태, 로그 등을 포함하고, 공통 테이블은 레이어 타입, 오류 코드 등 룩업 테이블을 포함한다.
- [0086] 네트워크 테이블에 네트워크 모델 정보가 저장되어 있고, 프로젝트 테이블에는 네트워크 테이블로부터 복사된 실제 학습이나 추론을 진행할 프로젝트 정보가 저장된다. 프로젝트가 생성된 후 네트워크 테이블과 별개의 구성을 가지게 되므로, 프로젝트에 사용된 기반 네트워크가 수정되더라도 영향이 없다. 개수가 많고 가변적인 데이터(입출력 데이터 및 가중치 정보)는 blob 또는 text 타입으로, 개수가 적고 가변적인 데이터(각 레이어 파라미터 등)는 레코드를 분할하여 저장하도록 되어 있다.
- [0087] 딥러닝 학습에 필요한 프로시저는, 인서트 네트워크(Insert Network), 인서트 레이어(Insert Layer), 메이크 프로젝트(Make Project), 입력 데이터 로더(Input Data Loader), 네트워크 초기화(Init Network), 트레인(Train), 모델 저장(Save Model) 및 테스트(Test)를 포함할 수 있다.
- [0088] 인서트 네트워크는 네트워크 이름, 네트워크 타입, 데이터셋 이름, 옵티마이저 타입, 옵티마이저 파라미터, 학습률, 배치 크기, 학습 횟수, 출력 레이어 인덱스를 포함하는 네트워크를 생성한다.
- [0089] 인서트 레이어는, 네트워크 아이디, 레이어 이름, 레이어 타입, 레이어 인덱스, 레이어 파라미터, 입력 레이어 인덱스를 포함하는 레이어를 등록한다.
- [0090] 메이크 프로젝트는, 프로젝트 이름, 데이터셋 이름, 네트워크 이름, 학습 또는 추론 플래그, GPU 개수를 포함하는 프로젝트를 생성한다.

- [0091] 입력 데이터 로더는, 네트워크 입력의 선택(레이어 인덱스, 쿼리 타입(2:학습 테이블, 0:학습 데이터, 4:검증 테이블, 3:검증 데이터)에 따라 쿼리를 입력한다.
- [0092] 네트워크 초기화는, 네트워크 모델을 구성한다.
- [0093] 트레인은, 프로젝트 아이디, 학습 세대 수, 배치 사이즈, 이어서 학습 여부, 저장 간격, 검증 간격, GPU 동기화 간격을 포함하는 학습을 시작한다.
- [0094] 모델 저장은, 프로젝트 테이블의 네트워크 정보를 네트워크 테이블로 복사(프로젝트 이름, 네트워크 이름)한다.
- [0095] 테스트는, 프로젝트 아이디, 모든 레이어의 결과 저장 여부 플래그를 포함하는 추론을 시작한다.
- [0096] 도 4는 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 방법을 설명하기 위한 동작 흐름도를 나타낸 도면이다.
- [0097] 도 4를 참조하면, 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 사용자 단말기(130) 및 데이터베이스(110)와 연동하는 학습 프레임워크(120)에서 쿼리 기반 프레임워크 딥러닝 학습 방법을 실행한다.
- [0098] 본 발명에 따른 학습 프레임워크(120)에서 제공하는 메인 화면은 도 9에 도시된 바와 같이 정보 영역(1), 로그(Log) 영역(2) 및 콘텐츠(Contents) 영역(3)을 포함한다. 도 9는 본 발명의 실시 예에 따른 학습 프레임워크에서 제공하는 메인 화면의 한 예를 나타낸 도면이다. 정보 영역(1)은 학습, 네트워크, 데이터 셋 리스트를 표현하고, 로그 영역(2)은 각종 로그를 볼 수 있는 영역이며, 콘텐츠 영역(3)은 학습 관리, 네트워크 관리 등의 기능 팝업을 표시하는 영역이다.
- [0099] 학습 프레임워크(120)는 사용자 단말기(130)로부터 학습 쿼리(Call Train) 또는 추론 쿼리(Call Test)를 입력받는다(S410).
- [0100] 이때, 학습 프레임워크(120)는, 외부 프레임워크와의 호환성을 위해 모델 컨버터를 사용하고, 기존 프레임워크의 기학습된 모델을 가져오거나, 데이터베이스(110)에서 정보 또는 데이터를 외부로 내보낼 때 모델 컨버터를 통하여 모델 포맷으로 변환할 수 있다.
- [0101] 모델 컨버터는, 모델 포맷에 정의된 망 구조(network structure) 및 모델 데이터(model data)를 데이터베이스의 네트워크 모델 테이블 포맷으로 변환하거나, 반대로 데이터베이스의 네트워크 모델을 모델 포맷으로 변환할 수 있다.
- [0102] 이어, 학습 프레임워크(120)는 학습 쿼리 또는 추론 쿼리에 따라 네트워크 초기화(Init Network), 네트워크 구성(Construct Network) 및 네트워크 갱신(Update Network)을 실행한다(S420).
- [0103] 여기서, 학습 프레임워크(120)는 도 10에 도시된 바와 같이 네트워크 생성 기능을 수행할 수 있다. 도 10은 본 발명의 실시 예에 따른 학습 프레임워크에서 제공하는 네트워크 생성 기능 화면을 나타낸 도면이다. 도 10에 도시된 바와 같이, 네트워크 생성 기능은 네트워크 기본 정보 입력(1), 데이터 셋(Dataset)(2), 네트워크 옵티마이저(Network Optimizer)(3) 및 네트워크 파라미터(Network Parameter)(4) 등을 설정하는 것이다. 네트워크 기본 정보로서 네트워크 ID, 네트워크 이름(Name), 네트워크 타입(Network Type), 학습률(Learning Rate), 배치(Batch), 트레인 반복(Train Iteration), 출력 레이어들(Output Layers)을 입력할 수 있다. 데이터 셋은 네트워크 타입에 맞는 데이터셋 리스트를 출력하고, 네트워크 타입이 'Data regression'인 경우에 2-1 영역을 활성화 할 수 있다. 네트워크 옵티마이저는 옵티마이저 파라미터로서 Adam/SGD를 설정할 수 있다. 그리고 옵티마이저 파라미터를 설정할 수 있다.
- [0104] 또한, 학습 프레임워크(120)는 도 11에 도시된 바와 같이 네트워크 레이어를 추가할 수 있다. 도 11은 본 발명의 실시예에 따른 학습 프레임워크에서 네트워크 레이어를 추가하는 화면을 나타낸 도면이다. 도 11에 도시된 바와 같이, 네트워크 레이어 추가 화면은 네트워크 레이어 정보 입력(1), 레이어 정보 출력(2), 레이어 파라미터 및 입력 정보 출력(3) 영역으로 구분된다. 네트워크 레이어 정보로서 레이어 이름(Layer Name), 레이어 타입(Layer Type), 레이어 인덱스(Layer Index), 레이어 파라미터(Layer Parameter), 레이어 입력 정보(Layer Input Info) 등을 입력할 수 있다.
- [0105] 또한, 학습 프레임워크(120)는 네트워크 레이어를 추가할 때 도 12에 도시된 바와 같이 파라미터를 추가하거나 수정하거나 삭제할 수 있다. 도 12는 본 발명의 실시 예에 따른 학습 프레임워크에서 네트워크 레이어의 추가시 파라미터 추가/수정/삭제 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 12에 도시된 바와 같이, 네트워크

크 레이어의 추가시 파라미터를 추가(Add; 1), 수정(Edit; 2), 삭제(Delete; 3) 할 수 있다. 추가(1)는 레이어 파라미터 추가 기능으로써, 사용자가 Add 버튼을 클릭하면(①), 레이어 파라미터 팝업(Layer Parameter Popup)에서 레이어 파라미터(Layer Parameter)를 선택하고(②), 레이어 파라미터 값을 입력한 후(③), 'OK' 버튼을 클릭한다(④). 수정(2)은 레이어 파라미터 수정 기능으로써, 파라미터(Params) 선택 박스에서 파라미터를 선택하고(①), 우측의 수정 버튼(Edit Button)을 클릭하며(②), 파라미터 수정 팝업(Parameter Edit Popup)에서 파라미터 값을 입력한 후(③), 'OK' 버튼을 클릭한다(④). 삭제(3)는 레이어 파라미터 삭제 기능으로써, 파라미터(Params) 선택 박스에서 파라미터를 선택하고(①), 우측의 삭제(Delete) 버튼을 클릭한다(②).

[0106] 또한, 학습 프레임워크(120)는 네트워크 레이어를 추가할 때 도 13에 도시된 바와 같이 레이어 입력(Layer Input) 정보를 추가할 수 있다. 도 13은 본 발명의 실시 예에 따른 학습 프레임워크에서 네트워크 레이어의 추가시 레이어 입력 정보를 추가하는 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 13에 도시된 바와 같이, 네트워크 레이어의 추가시 레이어 입력 정보를 추가(Add; 1), 수정(Edit; 2), 삭제(Delete; 3) 할 수 있다. 추가(1)는 레이어 파라미터 추가 기능으로써, 사용자가 Add 버튼을 클릭하면(①), 레이어 입력 정보 팝업(Input Layer Info Popup)에서 입력 레이어 인덱스(Input Layer Index)와 입력 텐서 인덱스(Input Tensor Index)를 입력하고(②), 'OK' 버튼을 클릭한다(③). 수정(2)은 레이어 파라미터 수정 기능으로써, 입력 정보(Input Infos) 선택 박스에서 입력 레이어 인덱스(Input Layer Index)를 선택하고(①), 우측의 수정 버튼(Edit Button)을 클릭하며(②), 레이어 입력 정보 팝업(Input Layer Info Popup)에서 레이어 인덱스(Layer Index)와 입력 텐서 인덱스(Input Tensor Index) 값을 수정한 후(③), 'OK' 버튼을 클릭한다(④). 삭제(3)는 레이어 파라미터 삭제 기능으로써, 레이어 입력 정보(Input Layer Info Popup) 선택 박스에서 입력 레이어 인덱스(Input Layer Index)를 선택하고(①), 우측의 삭제(Delete) 버튼을 클릭한다(②).

[0107] 이어, 학습 프레임워크(120)는 모든 층에 대한 초기화(Initialize all layers)가 이루어지면 학습(Train) 또는 추론(Test)을 실행한다(S430).

[0108] 여기서, 학습 프레임워크(120)는 도 14에 도시된 바와 같이 학습 생성 화면을 제공할 수 있다. 도 14는 본 발명의 실시 예에 따른 학습 프레임워크의 학습 생성 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 14에 도시된 바와 같이 학습 생성 화면을 통해 학습 기본 정보 입력(1), 옵티마이저 파라미터(Optimizer Parameter) 입력(2), 데이터셋 선택(3) 영역을 제공할 수 있다. 학습 프레임워크(120)는 학습 기본 정보(1)로써 프로젝트 명, 네트워크, 네트워크 타입, 학습율(Learning Rate), GPU, 트레인 모드(Train Mode) 등을 입력할 수 있다. 또한, 학습 프레임워크(120)는 옵티마이저 파라미터로써 파라미터 타입(Adam/SDG)을 입력할 수 있다. 그리고 학습 프레임워크(120)는 데이터셋 정보를 선택할 수 있다.

[0109] 또한, 학습 프레임워크(120)는 도 15에 도시된 바와 같이 학습 실행 화면을 제공할 수 있다. 도 15는 본 발명의 실시 예에 따른 학습 프레임워크의 학습 실행 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 15에 도시된 바와 같이 프로젝트 명, 배치 사이즈(Batch Size), 단계(Steps), 오프셋(Offset), 볼륨(Volume), 데이터 쉬프트(DataShift), 에퍼크(Epoch), 계속(Continue), 종료 저장(Save iteration), 검증 단계(Validation step), 동기화 간격(Sync Interval), 예상 CPU 사용량 표시, 예상 GPU 사용량 표시, 필요한 메모리 표시 등을 제공할 수 있다.

[0110] 또한, 학습 프레임워크(120)는 도 16에 도시된 바와 같이 학습 관리 화면을 제공할 수 있다. 도 16은 본 발명의 실시 예에 따른 학습 프레임워크의 학습 관리 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 16에 도시된 바와 같이 학습 관리 정보로써 프로젝트 ID, 프로젝트 명, 데이터셋, 네트워크명, 상태 정보, 학습 시작 시간, 학습 종료 시간, 학습 실행 시간 등을 제공할 수 있다.

[0111] 학습 프레임워크(120)는, 학습 종료시까지 배치 데이터를 획득하고(Get Batch Data) 반복(Iteration)하여 결과 및 모델을 저장(Store Result & Model)할 수 있다.

[0112] 또한, 학습 프레임워크(120)는 도 17에 도시된 바와 같이 로그 관리 및 자원 관리를 실행할 수 있다. 도 17은 본 발명의 실시 예에 따른 학습 프레임워크의 로그 관리 및 자원 관리 화면을 나타낸 도면이다. 학습 프레임워크(120)는 도 17에 도시된 바와 같이 자원 관리(1) 및 로그 관리(2) 기능을 수행할 수 있다. 즉 CPU, GPU, 메모리(Memory) 등의 자원을 관리할 수 있으며, 학습 이력 관리, 네트워크 이력 관리, 데이터셋 이력 관리, 작업 이력 관리 등의 로그를 관리할 수 있다.

[0113] 또한, 학습 프레임워크(120)는, 테스트(Test)를 실행하고, 테스트 데이터를 획득하여(Get Test Data) 피드 포워드(feedforward)하고 추론 결과를 저장(Store Result)할 수 있다.

- [0114] 이어, 학습 프레임워크(120)는 학습 또는 추론의 종료시 학습 결과 또는 추론 결과를 사용자 단말기(130)로 제공한다(S440).
- [0115] 전술한 학습 과정 또는 추론 과정에 대하여 아래에서 각각 구분하여 설명한다.
- [0116] 도 5는 본 발명의 실시 예에 따른 학습 프로시저의 실행 흐름을 나타낸 도면이다.
- [0117] 도 5를 참조하면, 본 발명의 실시 예에 따른 학습 프레임워크(120)에서는 사용자의 학습 요청에 대해 다음과 같은 단계가 실행되고, 데이터베이스(110)에서는 QML(120)을 호출하여 데이터를 인자로 넘겨주어 결과를 반환받는다.
- [0118] 먼저, 학습 프레임워크(120)는, 사용자 단말기(130)로부터 학습 쿼리가 입력되어 학습이 호출되면(Call Train)(S50), 네트워크 초기화(Init Network)(S51), 네트워크 구성(Construct Network)(S52), 네트워크 갱신(Update Network)(S53)을 실행한다.
- [0119] 또한, 학습 프레임워크(120)는, 모든 층에 대한 초기화(Initialize all layers)가 이루어지기까지(S54-No), 레이어 초기화(Init Layer)(S55)를 실행하고, 초기화된 레이어 정보(Initialized Layer Info)를 획득하며, 레이어 갱신(Update Layer)(S56)을 실행한다.
- [0120] 또한, 학습 프레임워크(120)는, 모든 층에 대한 초기화가 이루어지면(S54-Yes), 학습(Train)을 실행하고(S57), 학습 종료시까지(S58-No) 배치 데이터를 획득하고(Get Batch Data)(S59), 반복 되풀이(Iteration)하여(S60), 결과 및 모델을 저장하며(Store Result & Model)(S610), 학습 종료시(S58-Yes) 학습 결과를 사용자 단말기(130)로 제공한다(S62).
- [0121] 도 6은 본 발명의 실시 예에 따른 추론 프로시저의 실행 흐름을 나타낸 도면이다.
- [0122] 도 6을 참조하면, 본 발명의 실시 예에 따른 학습 프레임워크(120)에서는 사용자의 추론 요청에 대해 다음과 같은 단계가 실행되고, 데이터베이스(110)에서는 QML(120)을 호출하여 데이터를 인자로 넘겨주어 결과를 반환받는다.
- [0123] 먼저, 학습 프레임워크(120)는, 사용자 단말기(130)로부터 추론 쿼리가 입력되어 추론이 호출되면(Call Test)(S63), 네트워크 초기화(Init Network)(S64), 네트워크 구성(Construct Network)(S65), 네트워크 갱신(Update Network)(S66)을 실행한다.
- [0124] 또한, 학습 프레임워크(120)는, 모든 층에 대한 초기화(Initialize all layers)가 이루어지기까지(S67-No), 레이어 초기화(Init Layer)(S68) 및 레이어 갱신(Update Layer)(S69)을 실행한다.
- [0125] 또한, 학습 프레임워크(120)는, 모든 층에 대한 초기화가 이루어지면(S67-Yes), 추론 테스트(Test)를 실행하고(S70), 추론 데이터를 획득하여(Get Test Data)(S71) 피드 포워드(feed forward)하고(S72), 결과를 저장하며(Store Result)(S73), 추론 결과를 사용자 단말기(130)로 제공한다(S74).
- [0126] 도 7은 본 발명의 실시 예에 따른 모델 컨버터의 변환 동작을 나타낸 도면이다.
- [0127] 도 7을 참조하면, 본 발명의 실시 예에 따른 데이터베이스(110)에 저장된 네트워크 모델은 외부 프레임워크(tensorflow, pytorch, caffe, etc.)와의 호환성을 위해 모델 컨버터를 필요로 한다. 기존 프레임워크의 기학습된 모델을 가져오거나 데이터베이스 외부로 내보낼 때 예를 들면, ONNX(Open Neural Network Exchange) 포맷을 사용한다.
- [0128] 모델 컨버터는, ONNX 모델 포맷에 정의된 network structure 및 model data(weight, bias)를 데이터베이스(110)의 네트워크 모델 테이블 포맷으로 변환한다(a). 반대로 데이터베이스(110)의 네트워크 모델을 ONNX 모델로 변환한다(b).
- [0129] 기존 프레임워크(Caffe, tensorflow, pytorch, etc.)에서 기계학습된 모델은 사용자가 ONNX 모델로 변환 후 Converter(Import) 기능을 통해서 데이터베이스(110)에 업로드할 수 있다.
- [0130] QML(120)에서 학습한 모델은 Converter(Export) 기능을 통해서 데이터베이스(110)에서 ONNX 모델 포맷, 또는 CVS 파일로 저장한다.
- [0131] 데이터베이스(110)에 저장된 ONNX 모델은 사용자가 원하는 target 프레임 워크로 변환하여 사용할 수 있다.
- [0132] 한편, QML은 C 언어로 개발하고 있는 학습 프레임워크(120)이다. 데이터베이스(110)와 UDF(User Defined

Function)를 통해 연결되며 호출에 의해 실행된다. 학습 프레임워크(120)에 정의된 함수들은 UDF를 통해 데이터 베이스(110)에 등록되고, 등록된 UDF 호출을 통해서 학습 프레임워크(120)가 실행된다. UDF에서 사용할 수 있는 인자 변수의 타입은 integer, real number, string으로 정해져 있고 QML에서는 다음과 같이 사용한다. Integer는 네트워크 모델을 구성하는 필수 파라미터 중 정수값, QML 내부에 정의된 구조체 메모리의 주소값이다. Real number는 네트워크 모델을 구성하는 필수 파라미터 중 실수값이다. String은 개수가 가변적인 파라미터들과 blob 데이터(binary 데이터)이다.

[0133] QML 프레임워크에서는 채널 우선 데이터 포맷인 NCHW(N:batch, C: channel, H:height, W:width) 포맷을 따른다. 레이어 종류는 ONNX에서 사용되는 레이어를 지원하며, 각 레이어에 정의된 파라미터들도 ONNX 포맷을 따른다.

[0134] QML 프레임워크는 네트워크 모델을 학습 가능 하도록 backpropagation 알고리즘이 구현되어 있다. Backpropagation의 필수 요소인 gradient 계산 알고리즘과 모델 파라미터(weight, bias)를 업데이트 시키기 위한 optimization 알고리즘이 구현되어 있다. 네트워크 모델을 학습하는 방식은 두가지로 Train from scratch 및 Fine tuning을 지원한다. Train from scratch는 네트워크 모델을 처음부터 학습시킨다. 각 레이어의 가중치들은 가중치 초기화 알고리즘을 통해 결정된다. Fine tuning은 기학습된 모델의 가중치(import 기능을 통해 데이터베이스에 저장되거나, 이전 학습시도를 통해서 구한 가중치)를 읽어와서 레이어의 초기 가중치를 설정하고 학습을 진행한다.

[0135] 도 8은 본 발명의 실시 예에 따른 QML 프레임워크의 내부 구조를 개략적으로 나타낸 도면이다.

[0136] 도 8을 참조하면, 본 발명의 실시 예에 따른 학습 프레임워크(120)에서, QML\_네트워크\_t(qml\_network\_t)는 복수 개의 QML\_레이어\_t(qml\_layer\_t)로 구성되고, 하나의 QML\_레이어\_t(qml\_layer\_t)는 복수 개의 QML\_텐서\_t(qml\_tensor\_t)로 구성된다.

[0137] Object qml\_networks\_t 는 qml\_network\_t 를 들고 있고, Multi GPU 로 네트워크 모델을 학습하는 경우에 N 개의 qml\_network\_t가 qml\_networks\_t에 포함되어 있고, 네트워크 모델을 추론하는 경우에는 1개의 qml\_network\_t를 들고 있다.

[0138] Object qml\_network\_t는 여러 개의 qml\_layer\_t 및 네트워크 파라미터들을 들고 있다.

[0139] Object qml\_layer\_t 는 입출력 텐서(qml\_tensor\_t) 들을 갖고 있고, Object qml\_tensor\_t는 NCHW 포맷으로 구성된 4차원 텐서이고, 내부에는 dtype, qml\_shape\_t, data, name 등이 포함되어 있다.

[0140] 한편, 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 다음과 같이 클라이언트, 회원, 데이터셋, 네트워크, 학습, 학습 실행 등을 관리할 수 있다.

[0141] [클라이언트]

[0142] 본 발명의 실시 예에 따른 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 사용자 단말기(130)로 데이터셋과 기계학습 과정을 관리하고 결과를 확인하기 위한 기능을 제공한다.

[0143] [회원 관리]

[0144] 또한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 회원관리를 통해서 데이터베이스(110)의 데이터, 네트워크 모델의 생성, 수정 권한을 부여하고 변경 이력을 남긴다.

[0145] [데이터셋 관리]

[0146] 또한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 데이터셋을 관리하기 위하여 새로운 테이블을 생성하고, 데이터를 조회, 수정, 업로드하기 위한 기능을 제공한다. 새로운 데이터셋을 생성할 때 자동으로 새로운 테이블을 생성하고, 데이터를 업로드한다. 데이터베이스의 테이블에 접근하여 데이터를 조회하거나 사용자가 작성한 쿼리를 통해서 데이터베이스의 데이터를 조회한 결과를 보여준다. 권한에 따라 데이터를 수정한다. 사용자로부터 수치데이터를 입력 받거나, 하나 이상의 파일을 읽어서 데이터 업로드를 진행한다. 학습 데이터의 레이블을 작성(tagging)하는 기능을 제공한다.

[0147] [네트워크 관리]

[0148] 또한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 다음과 같이 네트워크 모델을 관리하기 위한 기능을 제공한다. 지원되는 레이어를 추가하고 레이어 파라미터를 조정하여 새로운 네트워크 모델을 생성한다. 기존

에 작성된 네트워크 모델 리스트를 조회한다. 기존에 작성된 네트워크 모델에 새로운 레이어를 추가하여 새로운 네트워크 모델을 생성한다. 네트워크 모델을 시각화하여 보여주는 기능을 제공한다.

[0149] [학습 관리]

[0150] 또한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 다음과 같이 학습을 관리하기 위한 기능을 제공한다. 네트워크 모델과 데이터셋, 학습 파라미터를 조절하여 학습을 생성 또는 수정한다. 학습이 끝난 네트워크 모델을 컨버터 기능을 통해 출력한다. 현재 사용 중인 서버의 자원을 확인한다.

[0151] [학습 실행 관리]

[0152] 또한, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템(100)은, 학습 및 추론을 수행하고 결과를 확인하기 위한 기능을 제공한다. 서버의 자원을 확인한다. 학습 및 추론 수행이 가능한지 여부를 사용자에게 알려준다. 현재 실행 또는 대기 중인 학습계획 리스트를 조회한다. 등록된 네트워크 모델과 데이터셋, 학습 파라미터를 설정하여 학습계획을 생성한다. 현재 실행 또는 대기 중인 학습계획의 학습 파라미터를 확인할 수 있다. 현재 실행 중인 학습계획의 중간 및 결과를 확인할 수 있다. 현재 실행 중인 학습계획을 멈출 수 있다. 대기 중인 학습계획을 시작할 수 있다. 등록된 네트워크 모델과 데이터셋을 설정하여 추론계획을 생성한다. 실행된 추론 계획의 결과를 확인할 수 있다.

[0153] 도 18은 본 발명의 실시 예에 따른 학습 프레임워크의 쿼리를 이용한 프레임워크 딥러닝 학습 방법을 설명하기 위한 동작 흐름도를 나타낸 도면이다.

[0154] 도 18을 참조하면, 사용자 단말기(130) 및 데이터베이스(120)와 연동하는 학습 프레임워크(120)는 사용자 단말기(130)로부터 쿼리(query)를 입력받는다(S1810).

[0155] 이어, 학습 프레임워크(120)는 입력받은 쿼리와 관련된 데이터를 데이터베이스(110)에서 탐색한다(S1820).

[0156] 즉, 학습 프레임워크(120)는 입력받은 쿼리를 인문, 사회, 경제, 문화, 과학, 예술, 종교, 방송, 건축, 법률, 건설, 음악, 행정, 교육, 생활, 상식, 도구, 인물, 사건, 의학, 약학, 경영, 심리, 언론, 정치, 외교, 간호, 환경, 물리, 기계, 천문, 산림, 산업, 생명, 의류, 조선, 해양, 화학, 항공, 식물, 농업, 조경, 금융 중 하나로 분류하고, 데이터베이스(110) 내에 해당 분류에 해당하는 데이터를 탐색하는 것이다.

[0157] 이어, 학습 프레임워크(120)는 탐색 결과를 딥러닝 방식으로 학습한다(S1830).

[0158] 그리고, 학습 프레임워크(120)는 딥러닝 학습 결과를 사용자 단말기(130)로 제공한다(S1840).

[0159] 학습 프레임워크(120)는 입력받은 쿼리를 RDF(Resource Description Framework) 모델의 자원(Resource)에 해당하는 주체(Subject)로 설정하여, 데이터베이스(110) 내에 저장되어 있는 데이터들을 주체(Subject), 서술(Predicate; Properties), 대상(Object; Literals)을 포함하는 RDF 모델을 적어도 하나 이상으로 생성하고, 생성된 RDF 모델들 간에 서로 충돌하는지를 판단하여, 서로 충돌하는 경우에는 분리하고, 서로 충돌하지 않는 경우에는 병합하여 RDF 네트워크를 생성하며, 생성된 RDF 네트워크에 기반하여 관련성이 동일한 대상들을 선택하여 탐색 결과로 제공하는 것이다.

[0160] 예를 들면, 학습 프레임워크(120)는 사용자 단말기(130)로부터 '추미애', '윤석열'이라는 두 개의 쿼리가 입력된 경우, 도 19에 도시된 바와 같이 주체1(S1)은 '윤석열', 주체2(S2)는 '추미애'로 설정하고, 데이터 베이스(110)의 내부에 저장되어 있는 데이터들 중에 '윤석열'과 관련되어 있는 '검찰청'을 서술1(P1)로 설정하고, '검찰총장'을 대상1(O1)로 설정하여, 서로 연결하여 RDF 모델1을 생성한다. 도 19는 본 발명의 실시 예에 따른 학습 프레임워크에서 입력받은 쿼리를 기반으로 데이터베이스의 내부에 있는 데이터들을 이용하여 RDF 모델을 생성하는 예를 나타낸 도면이다. 학습 프레임워크(120)는 주체2(S2)인 '추미애'와 관련된 '법무부'를 서술2(P2)로 설정하고, '법무장관'을 대상2(O2)로 설정하여, 서로 연결함으로써 RDF 모델2를 생성하는 것이다.

[0161] 또한, 학습 프레임워크(120)는 데이터베이스(120)에서 다른 데이터들을 전술한 바와 같은 동일한 과정으로 RDF 2, RDF 3, , RDF n 모델들을 생성한다. 예를 들어, 학습 프레임워크(120)는 데이터베이스(120)에서 도 20에 도시된 바와 같이 주체1(S1)인 '윤석열'과 관련이 있는 '서울중앙지검'을 서술5(P5)로 설정하고, '지검장'을 대상5(O5)로 설정한 후 RDF 모델5를 생성한다. 도 20은 본 발명의 실시 예에 따른 학습 프레임워크에서 RDF 모델을 생성 또는 병합하는 예를 나타낸 도면이다. 도 20에 도시된 바와 같이, 학습 프레임워크(120)는 RDF 모델1과 RDF 모델5가 동일한 주체1(S1)을 가지므로, 두 RDF 모델을 병합하여 새로운 RDF 모델1을 생성한다. 즉, 주체1(S1)인 '윤석열'에 서술1(P1)인 '검찰청'과 대상1(O1)인 '검찰총장'이 연결된 상태에서 서술5(P5)로 '서울중앙

지검'과 대상5(05)로 '지검장'이 동시에 연결된 RDF 모델을 생성하는 것이다.

[0162] 학습 프레임워크(120)는 전술한 바와 같은 과정으로 주체1(S1)과 관련이 있는 데이터들을 RDF 모델로 생성한 후 관련 RDF 모델들을 병합시켜 검색에 용이한 RDF 모델을 생성함으로써 사용자 단말기(130)로부터 입력받은 쿼리에 해당하는 대상을 용이하게 탐색할 수 있다.

[0163] 전술한 바와 같이 본 발명에 의하면, 딥러닝에 관한 전문적 지식이 없는 사용자라도 어려움 없이 사용자의 학습에 필요한 정보를 제공할 수 있도록 프레임워크가 데이터베이스에 플러그인 형태로 연결되어, 사용자의 학습시에 입력된 쿼리에 의해 데이터베이스에 저장된 정보 또는 데이터를 딥러닝 방식으로 학습하여, 쿼리에 대응하는 정보 또는 데이터를 제공할 수 있도록 하는, 쿼리를 이용한 프레임워크 딥러닝 학습 시스템 및 방법을 실현할 수 있다.

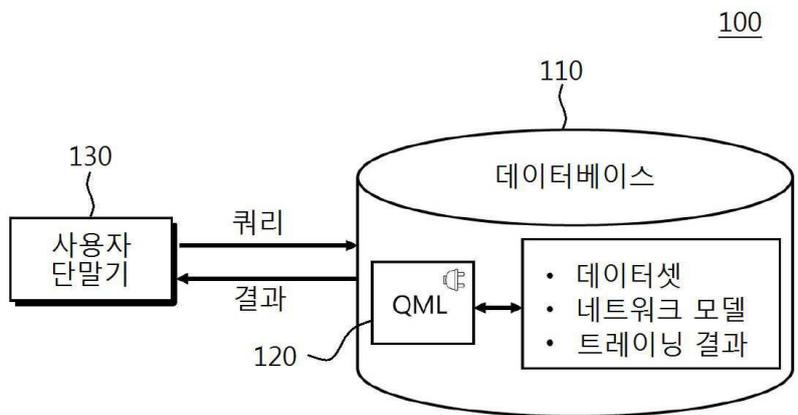
[0164] 본 발명이 속하는 기술 분야의 당업자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있으므로, 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다. 본 발명의 범위는 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 등가개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

**부호의 설명**

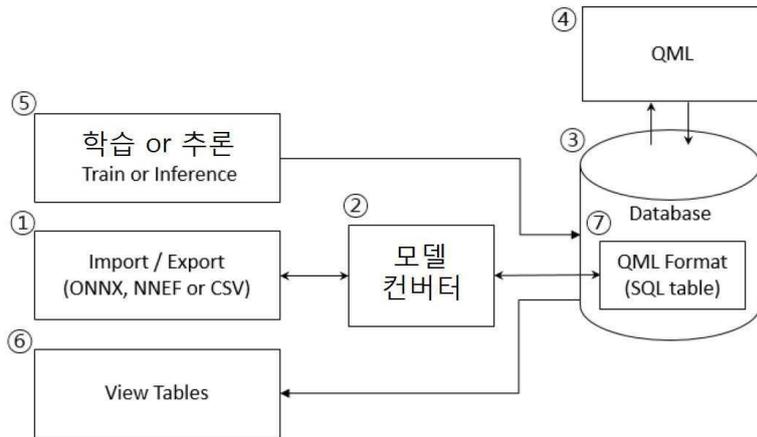
- [0165] 100 : 쿼리를 이용한 프레임워크 딥러닝 학습 시스템
- 110 : 데이터베이스
- 120 : 학습 프레임워크
- 130 : 사용자 단말기

**도면**

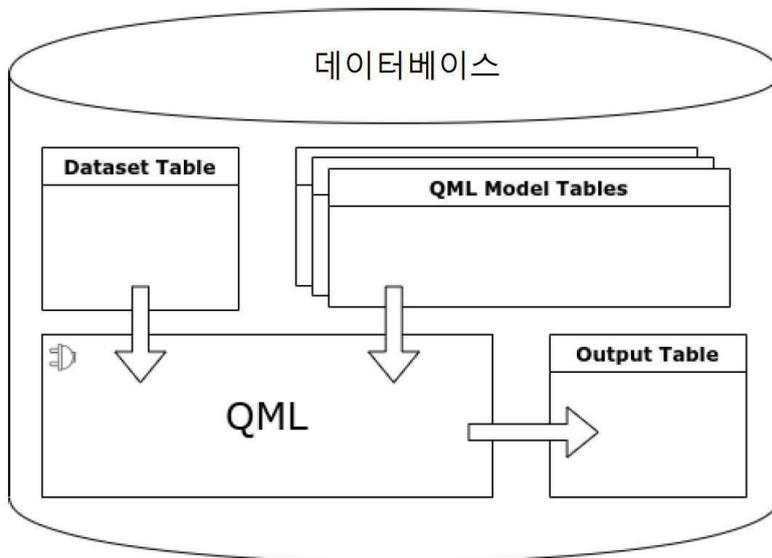
**도면1**



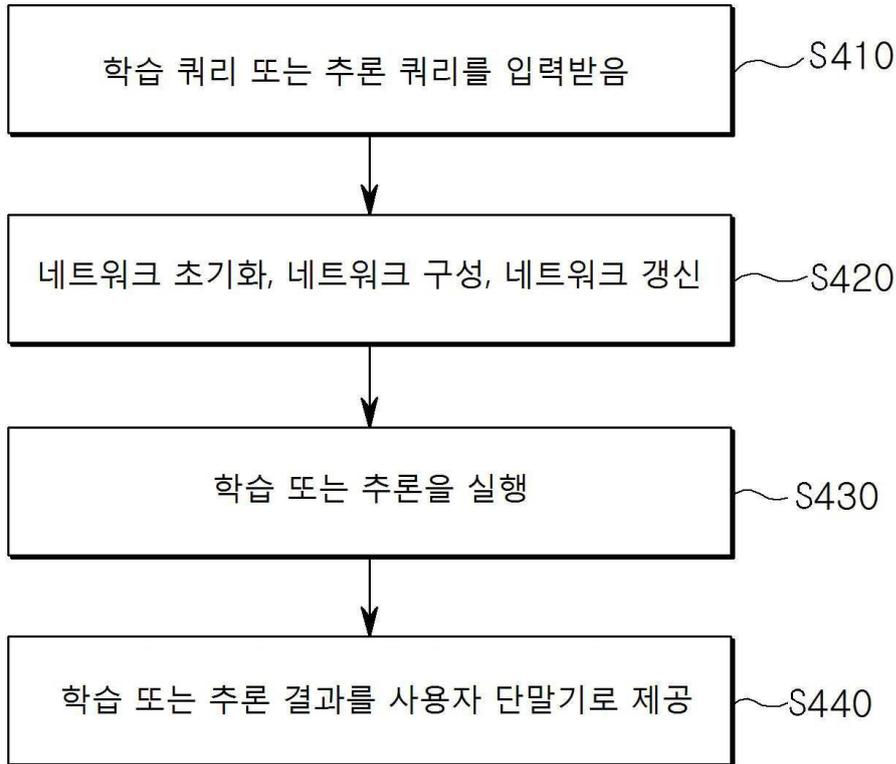
도면2



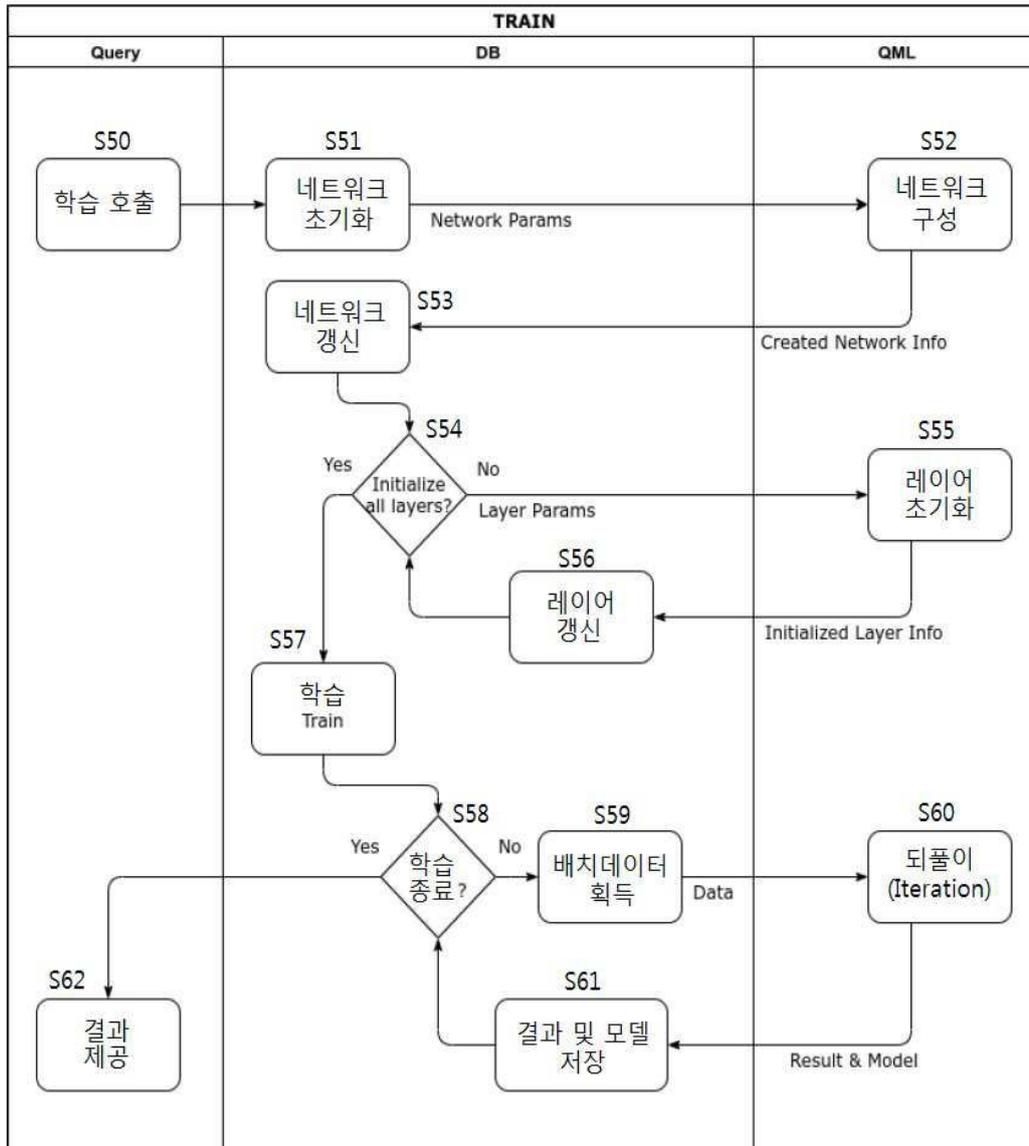
도면3



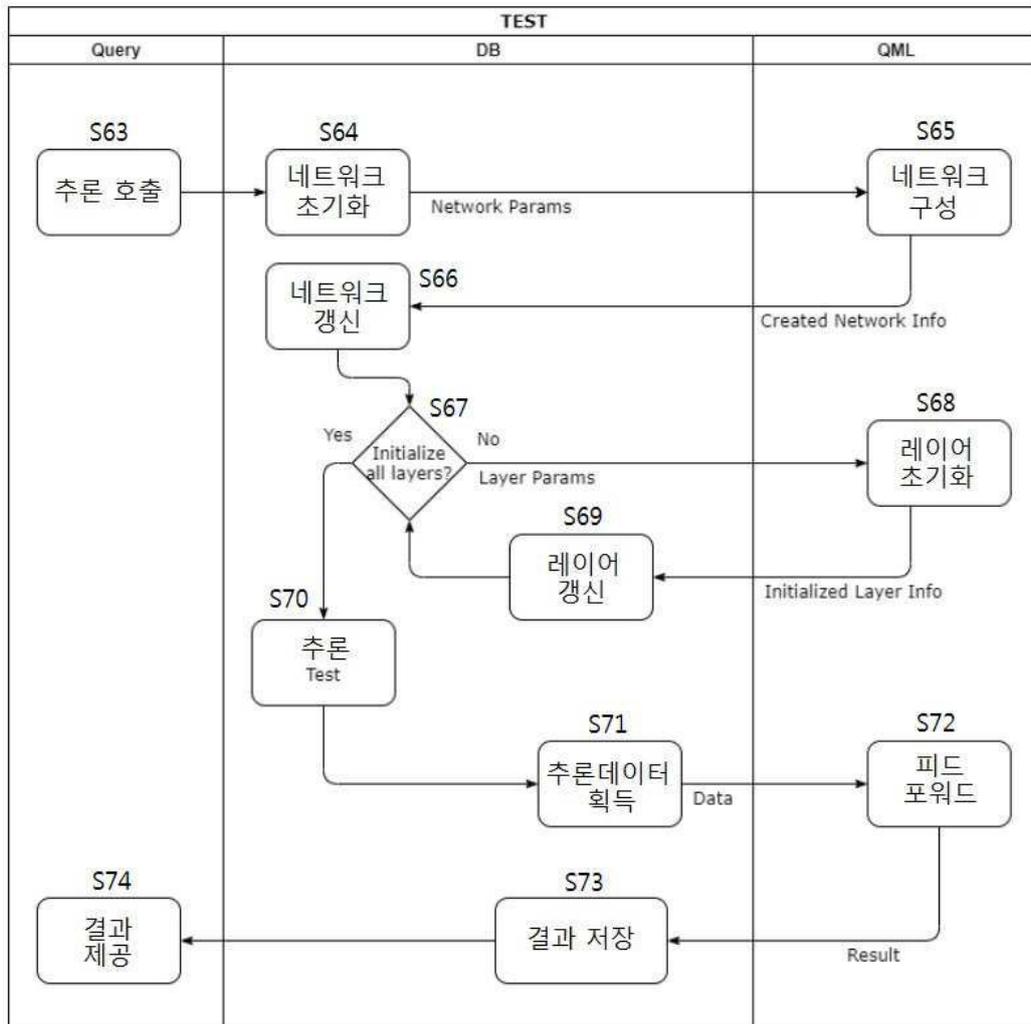
도면4



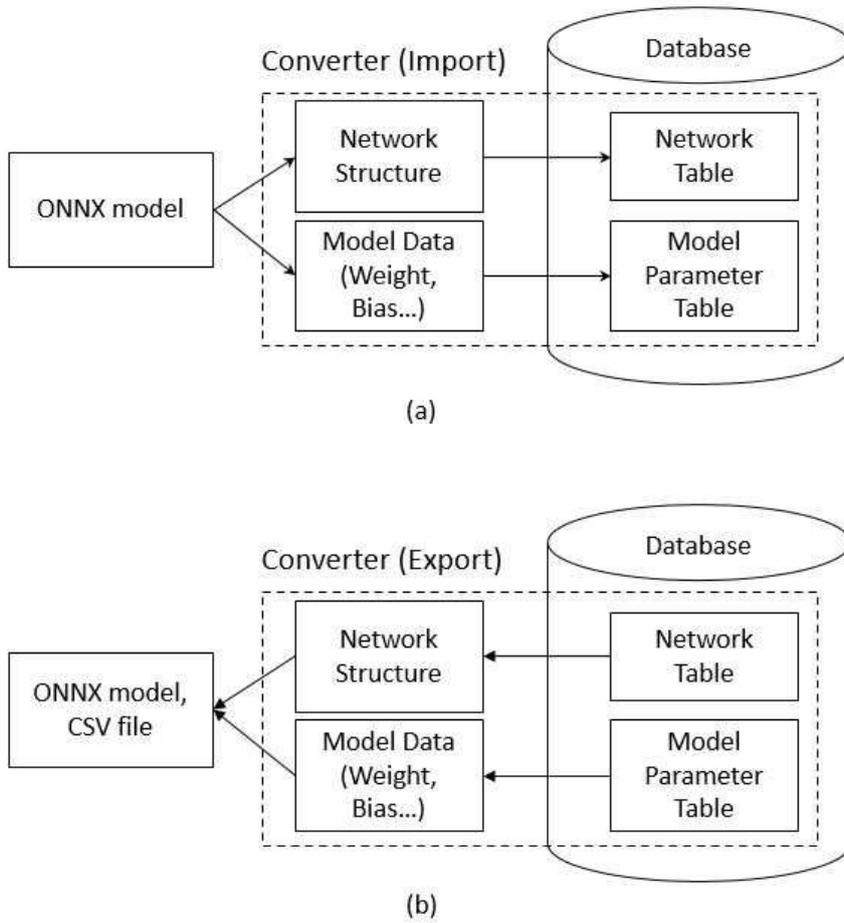
도면5



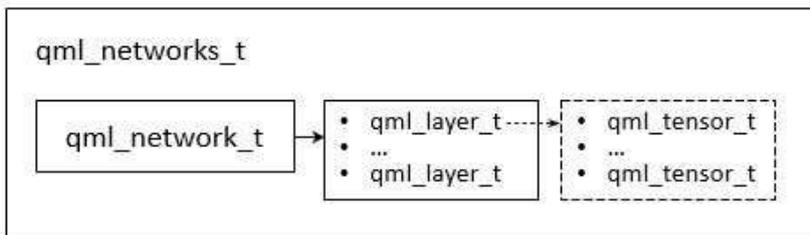
도면6



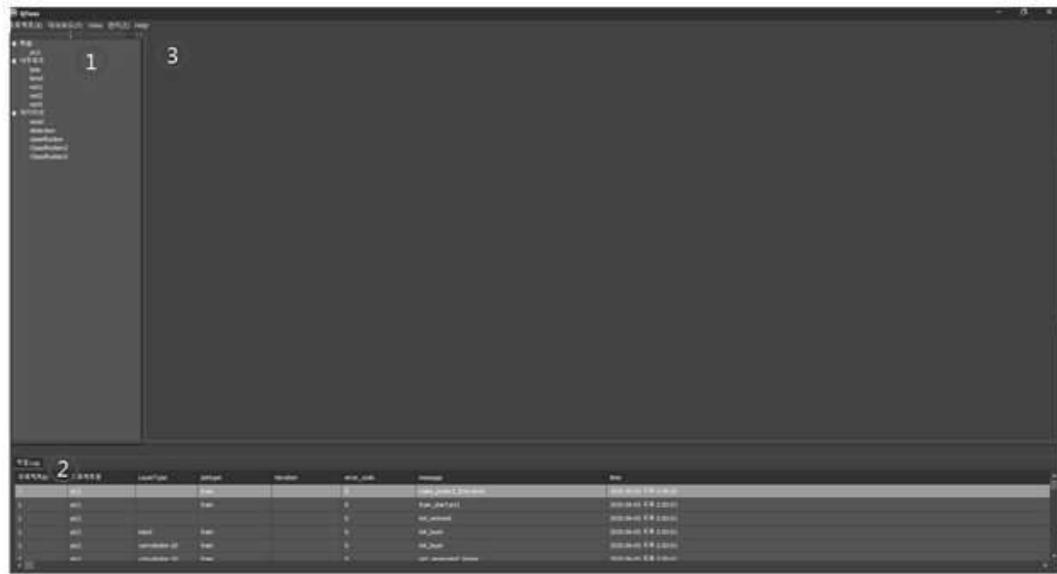
도면7



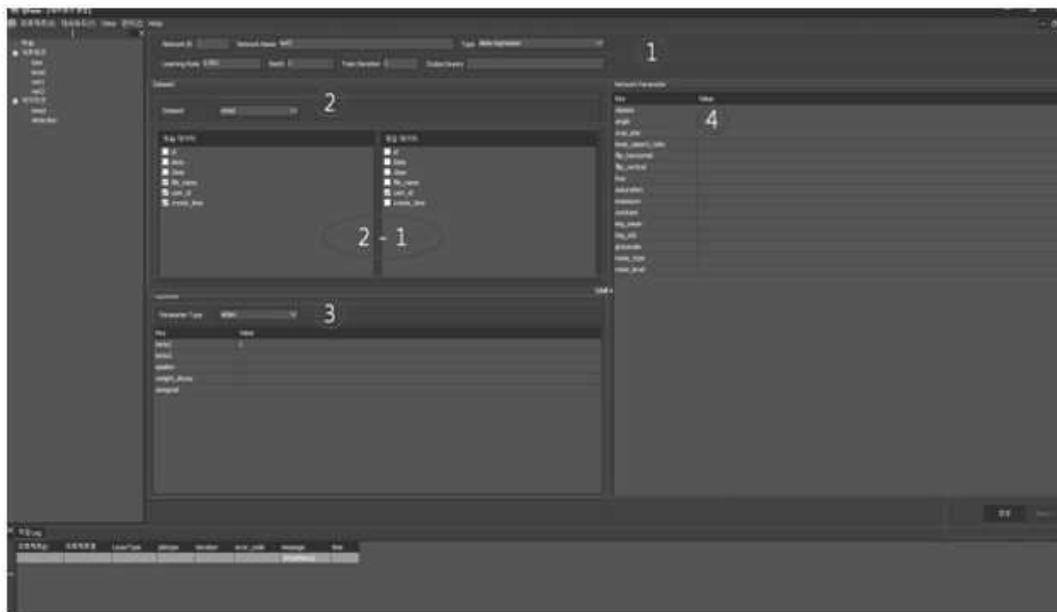
도면8



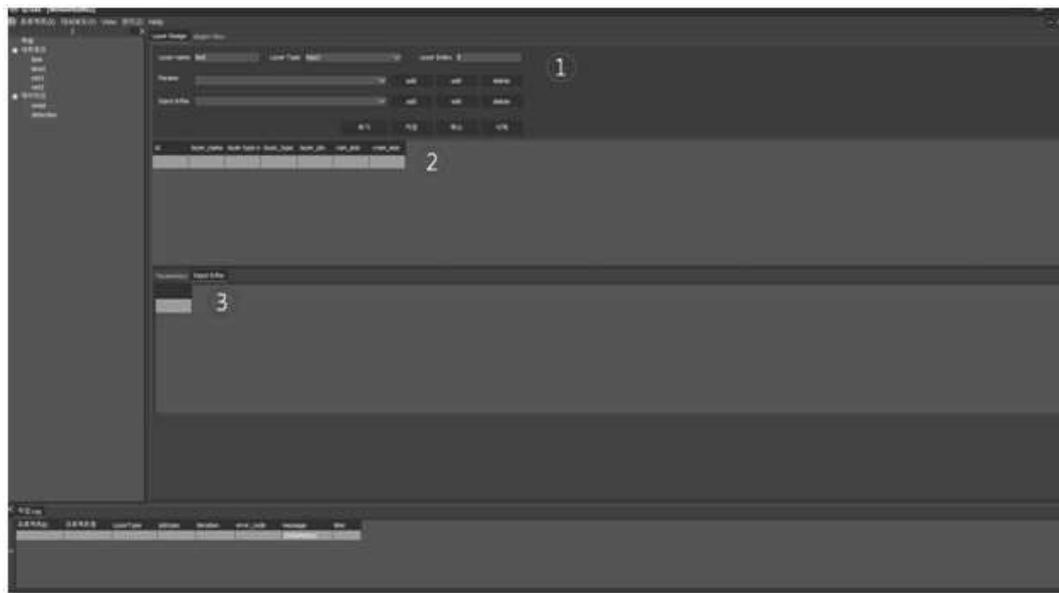
도면9



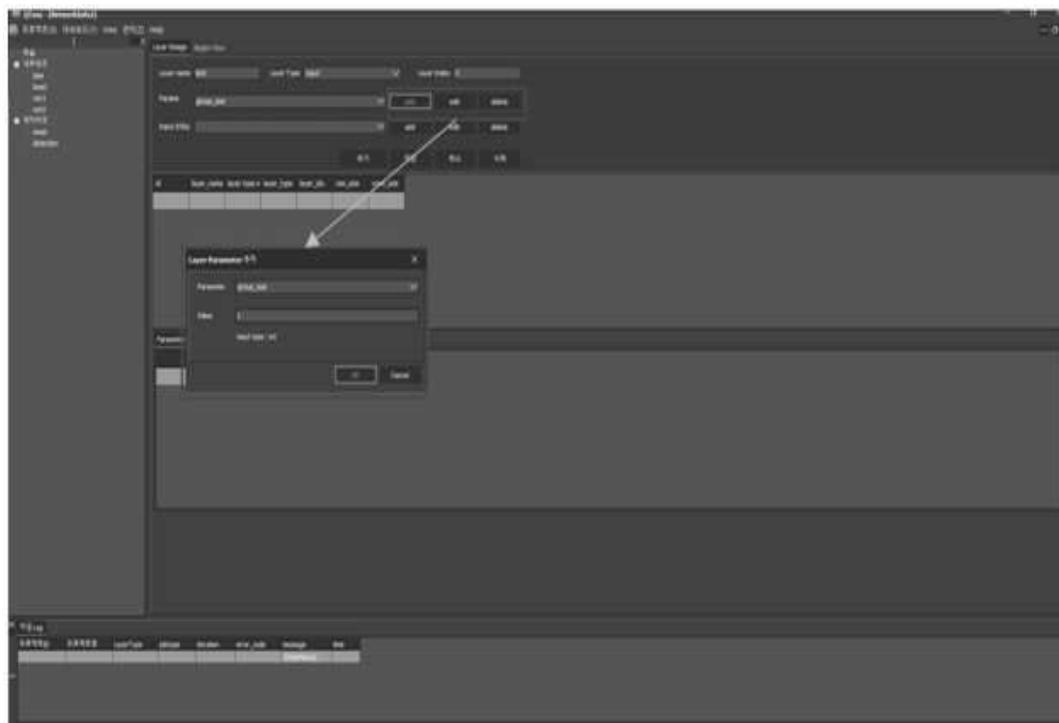
도면10



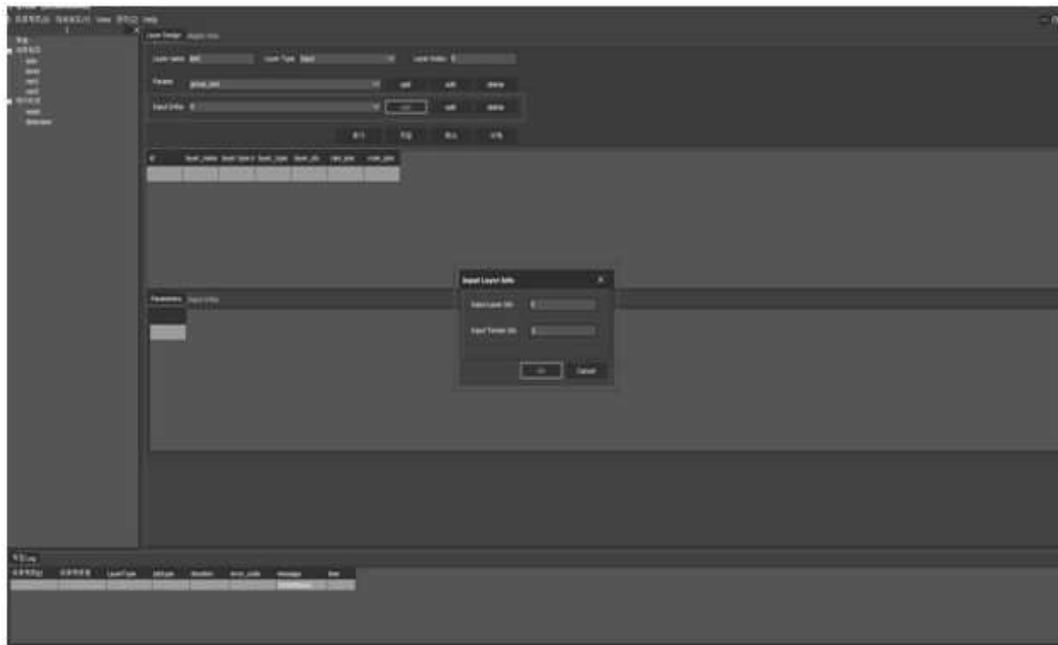
도면11



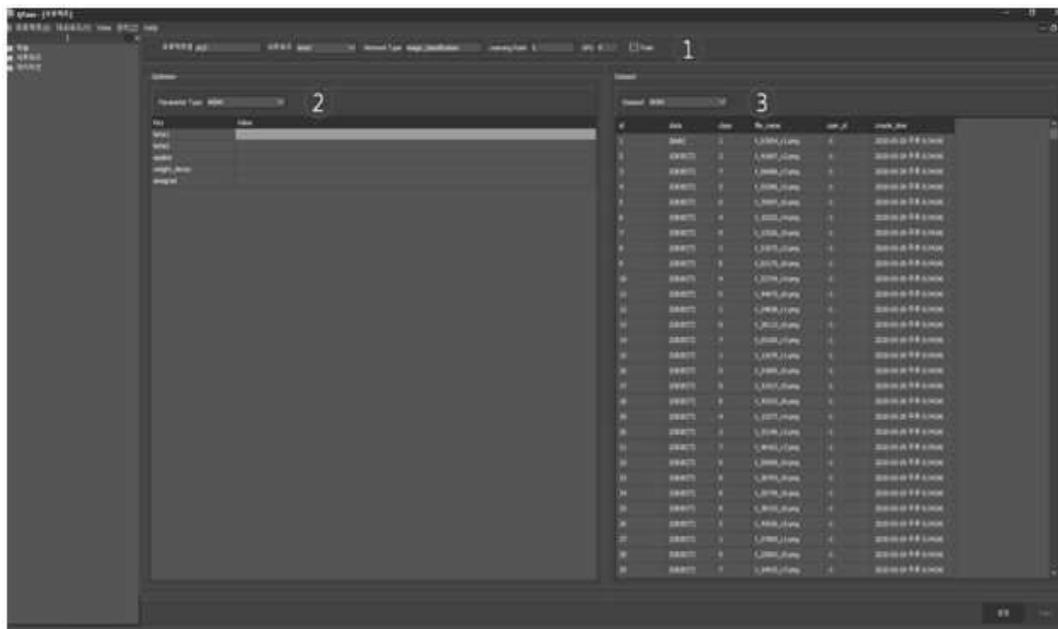
도면12



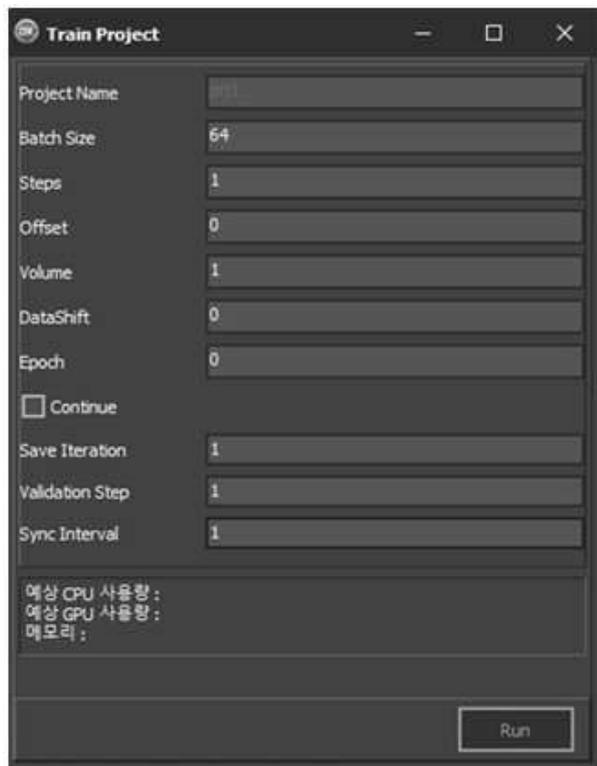
도면13



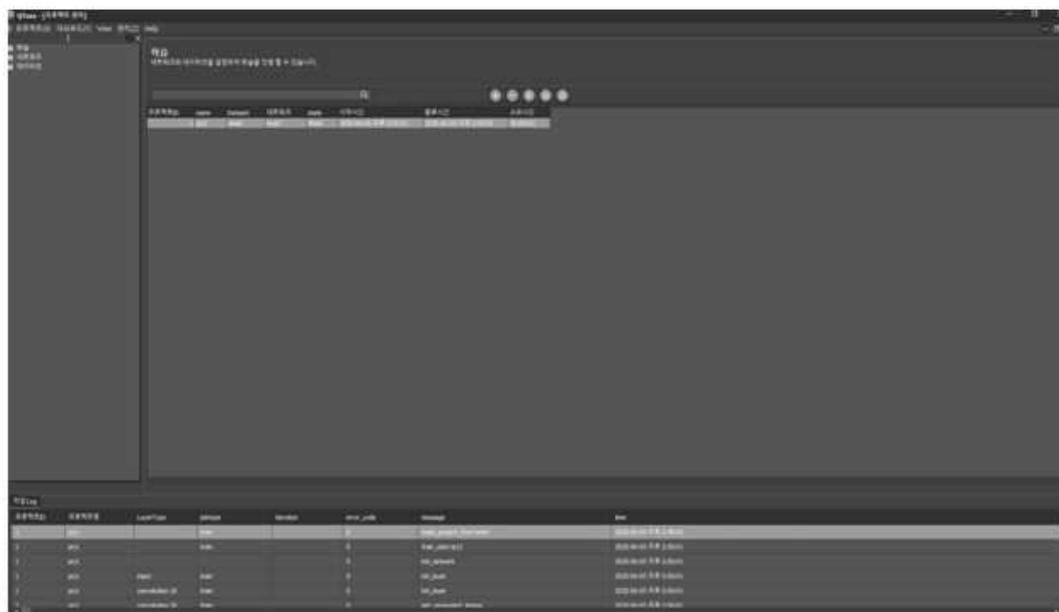
도면14



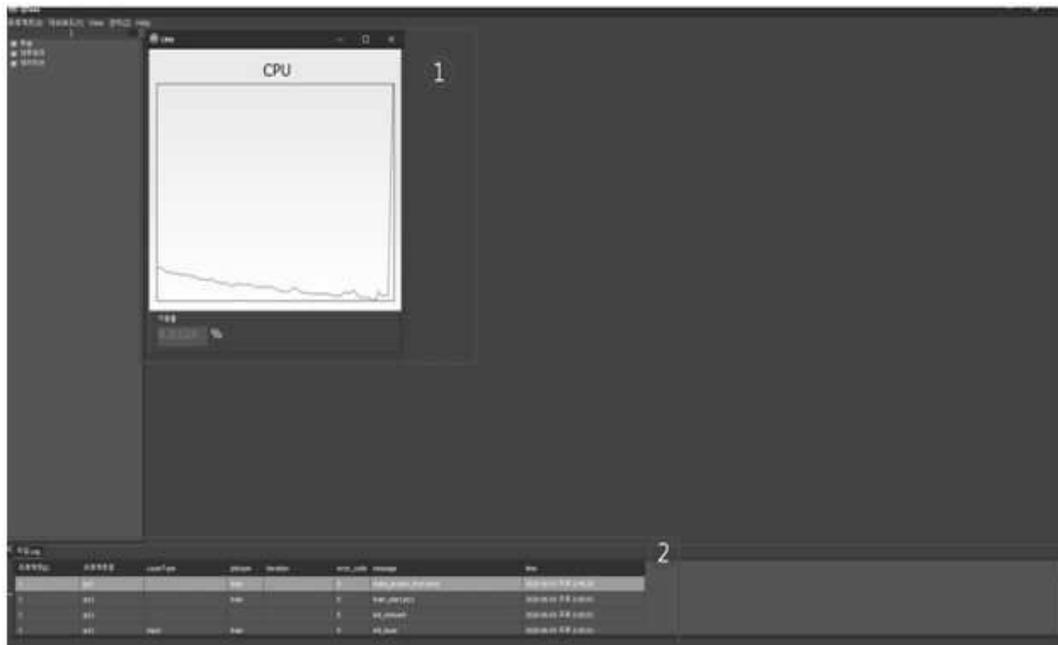
도면15



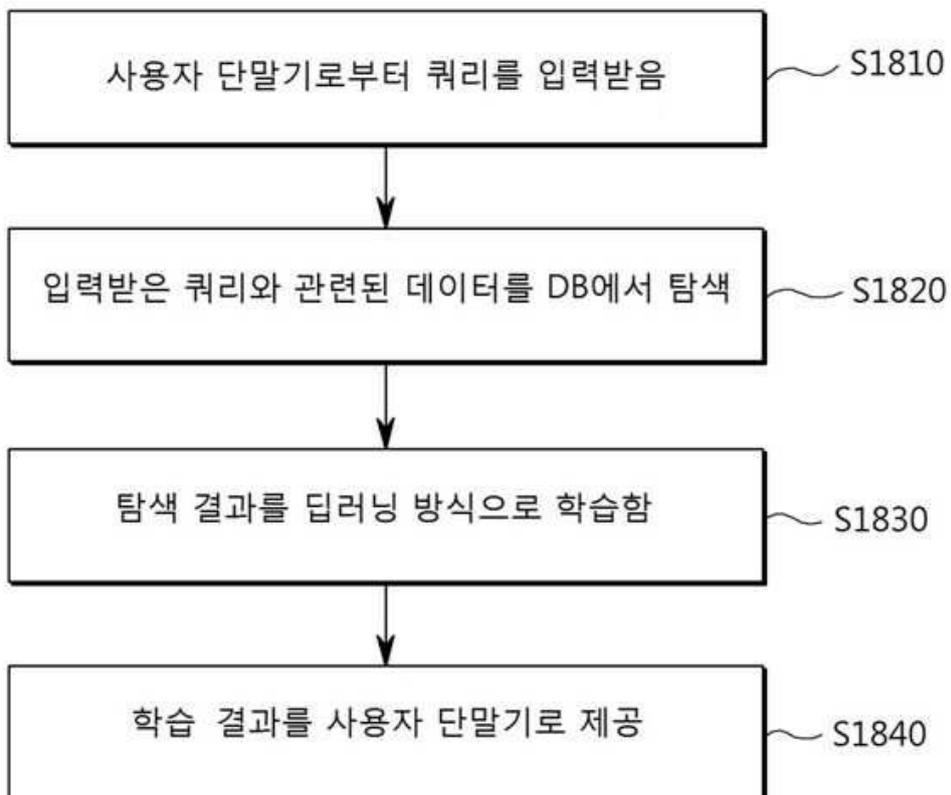
도면16



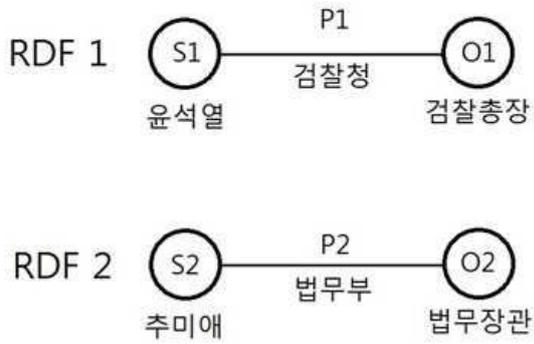
도면17



도면18



도면19



도면20

