(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0091610
(43) 공개일자 2025년06월23일

(51) 국제특허분류(Int. Cl.)
G06N 3/09 (2023.01) *G06N 3/047* (2023.01)
(52) CPC특허분류
G06N 3/09 (2023.01)
G06N 3/047 (2023.01)
(21) 출원번호 10-2023-0181734
(22) 출원일자 2023년12월14일
심사청구일자 2023년12월14일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
김성륜
서울특별시 용산구 한강대로 26, 101동 2407호(한강로3가, 한강대우트럼프월드3차)
국수진
서울특별시 강북구 덕릉로41길 12, 102동 1504호 (번동, 수유역 두산위브 아파트 1)
고승우
인천광역시 부평구 원적로 344, 105동 1404호 (산곡동, 금호 이수 마운트밸리)
(74) 대리인
민영준

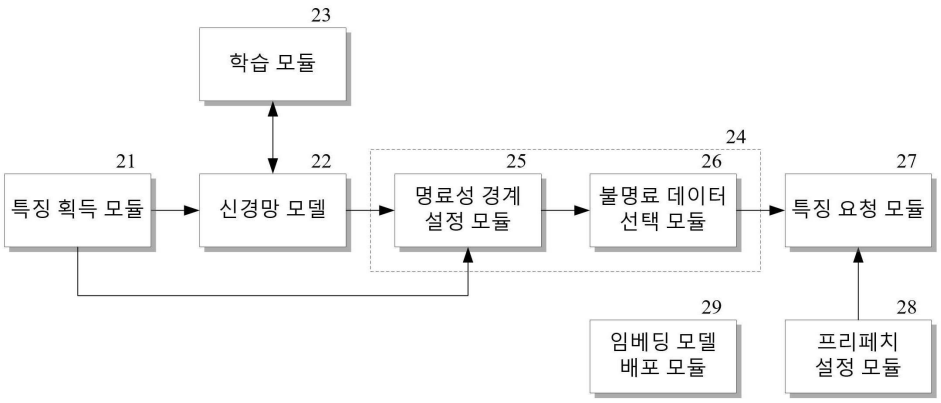
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **에너지 효율적인 엣지 학습 장치 및 방법**

(57) 요 약

본 개시는 다수의 데이터 샘플 각각을 가상의 특징 공간에 임베딩하여 획득된 다수의 특징 데이터 각각에서 중요도에 따라 순차 정렬된 다수의 특징 중 라운드 단위로 선택되는 특징 데이터의 특징이 순차적으로 오프로딩되면, 오프로딩된 특징을 누적한 누적 특징으로 신경망 모델을 학습시키고, 신경망 모델이 누적 특징에 대해 신경망 연산한 결과가 명료한지 여부를 판별하기 위해, 특징 공간 상에서 기준 위치로부터의 거리로 표현되는 명료성 경계 거리를 설정하며, 특징 공간에서 기준 위치로부터 다수의 누적 특징 각각 까지의 거리인 명료성 메트릭과 명료성 경계 거리에 따라 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터를 선택하여, 습을 위해 오프로드되어야 하는 학습 데이터의 양을 크게 줄일 뿐만 아니라, 오프로드되어야 하는 학습 데이터를 프리페치하여 에너지 소모를 저감시킬 수 있는 엣지 학습 장치 및 방법을 제공한다.

대 표 도



이 발명을 지원한 국가연구개발사업
과제고유번호 1711193744
과제번호 2021-0-00347-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 방송통신산업기술개발
연구과제명 6G 통신을 위한 Post MAC
기 여 율 1/3
과제수행기관명 연세대학교 산학협력단
연구기간 2023.01.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193913
과제번호 2021-0-02208-003
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송기술국제공동연구
연구과제명 6G 서브테라헤르츠 대역 증강 라우팅과 송수신 기법 개발
기 여 율 1/3
과제수행기관명 연세대학교 산학협력단
연구기간 2023.02.01 ~ 2023.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711193202
과제번호 2022-0-00420-002
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 6G핵심기술개발
연구과제명 6G 종단간 초정밀 네트워킹을 위한 핵심기술 개발
기 여 율 1/3
과제수행기관명 서울대학교 산학협력단
연구기간 2023.01.01 ~ 2023.12.31

명 세 서

청구범위

청구항 1

메모리; 및 상기 메모리에 저장된 프로그램에 따른 동작의 적어도 일부를 실행하는 프로세서를 포함하는 장치로서,

상기 프로세서는

다수의 데이터 샘플 각각을 가상의 특징 공간에 임베딩하여 획득된 다수의 특징 데이터 각각에서 중요도에 따라 순차 정렬된 다수의 특징 중 라운드 단위로 선택되는 특징 데이터의 특징이 순차적으로 오프로딩되면, 오프로딩된 특징을 누적한 누적 특징으로 신경망 모델을 학습시키고,

상기 신경망 모델이 상기 누적 특징에 대해 신경망 연산한 결과가 명료한지 여부를 판별하기 위해, 상기 특징 공간 상에서 기준 위치로부터의 거리로 표현되는 명료성 경계 거리를 설정하며,

상기 특징 공간에서 상기 기준 위치로부터 상기 다수의 누적 특징 각각 까지의 거리인 명료성 메트릭과 상기 명료성 경계 거리에 따라 상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터를 선택하는 엣지 학습 장치.

청구항 2

제1항에 있어서, 상기 프로세서는

현재 라운드의 특징이 오프로딩된 다수의 누적 특징 중 상기 명료성 메트릭이 상기 명료성 경계 거리 이하인 누적 특징을 탐색하고,

탐색된 누적 특징을 다음 라운드에 특징을 전송할 불명료 샘플 집합에 포함시키는 엣지 학습 장치.

청구항 3

제2항에 있어서, 상기 프로세서는

각 라운드에서 상기 불명료 샘플 집합에 포함된 누적 특징에 대한 특징 데이터에서 다음 순차의 특징이 다음 라운드에 오프로드되도록 적어도 하나의 엣지 단말로 요청하는 엣지 학습 장치.

청구항 4

제1항에 있어서, 상기 신경망 모델은

상기 데이터 샘플의 클래스를 분류하는 분류 모델이고,

상기 기준 위치는 상기 특징 공간에서 상기 신경망 모델이 상기 누적 특징에 대한 클래스를 구분하는 클래스 분류 경계로 나타나는 엣지 학습 장치.

청구항 5

제4항에 있어서, 상기 프로세서는

상기 신경망 모델이 이진 SVM(Support Vector Machine)으로 구현되는 경우, 각 클래스의 확률 분포를 가우시안 혼합 모델로 근사화하고,

근사화된 클래스별 확률 분포에서 설정된 절단 영역을 제외한 나머지 영역 중 인접 클래스의 확률 분포와 중첩되는 영역이 모두 포함되도록 하는 최대 거리를 상기 명료성 경계 거리로 설정하는 엣지 학습 장치.

청구항 6

제4항에 있어서, 상기 프로세서는

상기 신경망 모델이 DNN(Deep Neural Network)로 구현되는 경우, 상기 누적 특징에 대한 상기 신경망 모델의 신

경망 연산 결과와 상기 데이터 샘플에 레이블된 진리값 사이에 오류가 발생하는 횟수를 기반으로 상기 명료성 경계 거리를 설정하는 엣지 학습 장치.

청구항 7

제6항에 있어서, 상기 프로세서는

상기 누적 특징에 대한 상기 명료성 메트릭을 상기 신경망 모델이 분류한 클래스와 인접한 클래스 각각에 대한 사후 우도의 차이로 획득하는 엣지 학습 장치.

청구항 8

제1항에 있어서, 상기 프로세서는

상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터가 선택되기 이전에, 설정된 프리페치 개수에 따라 임의로 선택되어 프리페치되는 특징을 전송받는 엣지 학습 장치.

청구항 9

제8항에 있어서, 상기 프로세서는

프리페치로 선택된 특징을 전송하기 위해 소모되는 에너지와 다음 라운드에 특징을 오프로드하도록 선택되었으나 프리페치되지 않은 특징을 전송하기 위해 소모되는 에너지의 합이 최소가 되도록 상기 프리페치 개수를 설정하는 엣지 학습 장치.

청구항 10

제1항에 있어서, 상기 프로세서는

상기 데이터 샘플을 임베딩하여 다수의 특징이 중요도에 따라 정렬된 상기 특징 데이터를 획득하기 위한 임베딩 모델을 상기 데이터 샘플을 수집하는 적어도 하나의 엣지 단말로 배포하는 엣지 학습 장치.

청구항 11

프로세서에 의해 수행되는 방법으로서,

다수의 데이터 샘플 각각을 가상의 특징 공간에 임베딩하여 획득된 다수의 특징 데이터 각각에서 중요도에 따라 순차 정렬된 다수의 특징 중 라운드 단위로 선택되는 특징 데이터의 특징이 순차적으로 오프로딩되면, 오프로딩된 특징을 누적한 누적 특징으로 신경망 모델을 학습시키는 단계;

상기 신경망 모델이 상기 누적 특징에 대해 신경망 연산한 결과가 명료한지 여부를 판별하기 위해, 상기 특징 공간 상에서 기준 위치로부터의 거리로 표현되는 명료성 경계 거리를 설정하는 단계; 및

상기 특징 공간에서 상기 기준 위치로부터 상기 다수의 누적 특징 각각 까지의 거리인 명료성 메트릭과 상기 명료성 경계 거리에 따라 상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터를 선택하는 단계를 포함하는 엣지 학습 방법.

청구항 12

제11항에 있어서, 상기 특징 데이터를 선택하는 단계는

현재 라운드의 특징이 오프로딩된 다수의 누적 특징 중 상기 명료성 메트릭이 상기 명료성 경계 거리 이하인 누적 특징을 탐색하고,

탐색된 누적 특징을 다음 라운드에 특징을 전송할 불명료 샘플 집합에 포함시키는 엣지 학습 방법.

청구항 13

제12항에 있어서, 상기 특징 데이터를 선택하는 단계는

각 라운드에서 상기 불명료 샘플 집합에 포함된 누적 특징에 대한 특징 데이터에서 다음 순차의 특징이 다음 라운드에 오프로드되도록 적어도 하나의 엣지 단말로 요청하는 엣지 학습 방법.

청구항 14

제11항에 있어서, 상기 신경망 모델은

상기 데이터 샘플의 클래스를 분류하는 분류 모델이고,

상기 기준 위치는 상기 특징 공간에서 상기 신경망 모델이 상기 누적 특징에 대한 클래스를 구분하는 클래스 분류 경계로 나타나는 엣지 학습 방법.

청구항 15

제14항에 있어서, 상기 명료성 경계 거리를 설정하는 단계는

상기 신경망 모델이 이진 SVM(Support Vector Machine)으로 구현되는 경우, 각 클래스의 확률 분포를 가우시안 혼합 모델로 근사화하고,

근사화된 클래스별 확률 분포에서 설정된 절단 영역을 제외한 나머지 영역 중 인접 클래스의 확률 분포와 중첩되는 영역이 모두 포함되도록 하는 최대 거리를 상기 명료성 경계 거리로 설정하는 엣지 학습 방법.

청구항 16

제14항에 있어서, 상기 명료성 경계 거리를 설정하는 단계는

상기 신경망 모델이 DNN(Deep Neural Network)로 구현되는 경우, 상기 누적 특징에 대한 상기 신경망 모델의 신경망 연산 결과와 상기 데이터 샘플에 레이블된 진리값 사이에 오류가 발생하는 횟수를 기반으로 상기 명료성 경계 거리를 설정하는 엣지 학습 방법.

청구항 17

제16항에 있어서, 상기 특징 데이터를 선택하는 단계는

상기 누적 특징에 대한 상기 명료성 메트릭을 상기 신경망 모델이 분류한 클래스와 인접한 클래스 각각에 대한 사후 우도의 차이로 획득하는 엣지 학습 방법.

청구항 18

제11항에 있어서, 상기 엣지 학습 방법은

상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터가 선택되기 이전에, 설정된 프리페치 개수에 따라 임의로 선택되어 프리페치되는 특징을 인가받는 단계를 더 포함하는 엣지 학습 방법.

청구항 19

제18항에 있어서, 상기 프리페치되는 특징을 인가받는 단계는

프리페치로 선택된 특징을 전송하기 위해 소모되는 에너지와 다음 라운드에 특징을 오프로드하도록 선택되었으나 프리페치되지 않은 특징을 전송하기 위해 소모되는 에너지의 합이 최소가 되도록 상기 프리페치 개수를 설정하는 엣지 학습 방법.

청구항 20

제11항에 있어서, 상기 신경망 모델을 학습시키는 단계는

상기 데이터 샘플을 임베딩하여 다수의 특징이 중요도에 따라 정렬된 상기 특징 데이터를 획득하기 위한 임베딩 모델을 상기 데이터 샘플을 수집하는 적어도 하나의 엣지 단말로 배포하는 엣지 학습 방법.

발명의 설명

기 술 분 야

[0001] 본 개시는 엣지 학습 장치 및 방법에 관한 것으로, 에너지 효율적인 엣지 학습 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 최근 스마트 시티에 대한 관심이 높아지고 있다. 스마트 시티는 각종 IoT(Internet of Things) 기기를 활용하여 수집되는 대량의 실시간 데이터를 활용하여 자율 주행, 원격 제어 및 헬스 케어 등과 같은 다양한 서비스를 자동화할 수 있어야 한다. 그리고 각 IoT 기기가 여러 다양한 환경에서 수집한 데이터를 이용하여 인공 신경망(또는 인공 지능(Artificial Intelligence: AI))을 학습시킴으로써 사용자 맞춤형 자동화 서비스가 이루어질 수 있다.

[0003] 다만 사용자 맞춤형 자동화 서비스가 안정적으로 수행되기 위해서는 다수의 IoT 기기에서 실시간으로 수집된 정보를 기반으로 빠르게 학습이 수행될 필요가 있으며, 각 환경에 맞는 학습이 수행될 필요가 있다. 이에 중앙 클라우드가 아닌 액세스 포인트와 같이 분산된 네트워크의 엣지에 구비되는 엣지 서버가 학습을 수행하는 엣지 학습(edge learning)이 제안되어 활용되고 있다.

[0004] 엣지 학습에서도 엣지 서버가 다수의 IoT 기기에서 오프로드된 데이터 샘플을 사용하여 학습하는 중앙 집중식 엣지 학습과 엣지 서버가 다수의 모바일 기기 또는 IoT 기기에서 학습된 로컬 모델을 교환하는 연합 엣지 학습이 있다.

[0005] 연합 엣지 학습은 수집된 데이터를 직접 엣지 서버로 오프로드하지 않으므로 오프로드를 위한 에너지 소비는 줄일 수 있으나, 모바일 기기 또는 IoT 기기가 로컬 모델을 학습시킬 수 있는 성능을 가져야 하며, 로컬 모델을 학습시키기 위한 에너지 소모가 크다는 문제가 있다. 중앙 집중식 엣지 학습은 로컬 모델을 학습시키기 어려운 낮은 성능의 IoT 기기를 사용하는 환경에 적합하지만, IoT 기기가 수집한 대량의 학습 데이터를 엣지 서버로 오프로드하는 데에 많은 에너지가 소비될 뿐만 아니라 네트워크 트래픽을 유발하는 문제가 있다.

[0006] 특히 많은 IoT 기기가 배터리를 기반으로 동작하므로, 에너지 효율적인 엣지 학습이 요구되고 있다.

발명의 내용

해결하려는 과제

[0007] 본 개시의 목적은 적은 양의 학습 데이터로도 효과적인 학습이 수행되어 학습 데이터를 오프로드하기 위한 에너지를 줄일 수 있는 엣지 학습 장치 및 방법에 관한 것이다.

[0008] 본 개시의 목적은 IoT 기기가 수집한 학습 데이터에서 추출된 특징을 중요도와 신경망 모델의 학습 상태에 따라 부분적으로 인가받아 학습을 수행함으로써, 학습을 위해 오프로드되어야 하는 학습 데이터의 양을 크게 저감시킬 수 있는 엣지 학습 장치 및 방법에 관한 것이다.

[0009] 본 개시의 목적은 학습 데이터를 프리페치하여 오프로드 시간을 증가시킴으로써, 데이터 전송을 위한 시간 증가에 따라 에너지 소모를 저감시킬 수 있는 엣지 학습 장치 및 방법에 관한 것이다.

과제의 해결 수단

[0010] 본 개시의 일 실시 예에 따른, 엣지 학습 장치는 메모리; 및 상기 메모리에 저장된 프로그램에 따른 동작의 적어도 일부를 실행하는 프로세서를 포함하는 장치로서, 상기 프로세서는 다수의 데이터 샘플 각각을 가상의 특징 공간에 임베딩하여 획득된 다수의 특징 데이터 각각에서 중요도에 따라 순차 정렬된 다수의 특징 중 라운드 단위로 선택되는 특징 데이터의 특징이 순차적으로 오프로딩되면, 오프로딩된 특징을 누적한 누적 특징으로 신경망 모델을 학습시키고, 상기 신경망 모델이 상기 누적 특징에 대해 신경망 연산한 결과가 명료한지 여부를 판별하기 위해, 상기 특징 공간 상에서 기준 위치로부터의 거리로 표현되는 명료성 경계 거리를 설정하며, 상기 특징 공간에서 상기 기준 위치로부터 상기 다수의 누적 특징 각각 까지의 거리인 명료성 메트릭과 상기 명료성 경계 거리에 따라 상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터를 선택한다.

[0011] 상기 프로세서는 현재 라운드의 특징이 오프로딩된 다수의 누적 특징 중 상기 명료성 메트릭이 상기 명료성 경계 거리 이하인 누적 특징을 탐색하고, 탐색된 누적 특징을 다음 라운드에 특징을 전송할 불명료 샘플 집합에 포함시킬 수 있다.

[0012] 상기 신경망 모델은 상기 데이터 샘플의 클래스를 분류하는 분류 모델이고, 상기 기준 위치는 상기 특징 공간에서 상기 신경망 모델이 상기 누적 특징에 대한 클래스를 구분하는 클래스 분류 경계로 나타날 수 있다.

[0013] 상기 프로세서는 상기 신경망 모델이 이진 SVM(Support Vector Machine)으로 구현되는 경우, 각 클래스의 확률

분포를 가우시안 혼합 모델로 근사화하고, 근사화된 클래스별 확률 분포에서 설정된 절단 영역을 제외한 나머지 영역 중 인접 클래스의 확률 분포와 중첩되는 영역이 모두 포함되도록 하는 최대 거리를 상기 명료성 경계 거리로 설정할 수 있다.

[0014] 상기 프로세서는 상기 신경망 모델이 DNN(Deep Neural Network)로 구현되는 경우, 상기 누적 특징에 대한 상기 신경망 모델의 신경망 연산 결과와 상기 데이터 샘플에 레이블된 진리값 사이에 오류가 발생하는 횟수를 기반으로 상기 명료성 경계 거리를 설정할 수 있다.

[0015] 상기 프로세서는 상기 누적 특징에 대한 상기 명료성 메트릭을 상기 신경망 모델이 분류한 클래스와 인접한 클래스 각각에 대한 사후 우도의 차이로 획득할 수 있다.

[0016] 상기 프로세서는 상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터가 선택되기 이전에, 설정된 프리페치 개수에 따라 임의로 선택되어 프리페치되는 특징을 전송받을 수 있다.

[0017] 상기 프로세서는 프리페치로 선택된 특징을 전송하기 위해 소모되는 에너지와 다음 라운드에 특징을 오프로드하도록 선택되었으나 프리페치되지 않은 특징을 전송하기 위해 소모되는 에너지의 합이 최소가 되도록 상기 프리페치 개수를 설정할 수 있다.

[0018] 상기 프로세서는 상기 데이터 샘플을 임베딩하여 다수의 특징이 중요도에 따라 정렬된 상기 특징 데이터를 획득하기 위한 임베딩 모델을 상기 데이터 샘플을 수집하는 적어도 하나의 엣지 단말로 배포할 수 있다.

[0019] 본 개시의 다른 실시 예에 따른, 엣지 학습 방법은 프로세서에 의해 수행되는 방법으로서, 다수의 데이터 샘플 각각을 가상의 특징 공간에 임베딩하여 획득된 다수의 특징 데이터 각각에서 중요도에 따라 순차 정렬된 다수의 특징 중 라운드 단위로 선택되는 특징 데이터의 특징이 순차적으로 오프로딩되면, 오프로딩된 특징을 누적한 누적 특징으로 신경망 모델을 학습시키는 단계; 상기 신경망 모델이 상기 누적 특징에 대해 신경망 연산한 결과가 명료한지 여부를 판별하기 위해, 상기 특징 공간 상에서 기준 위치로부터의 거리로 표현되는 명료성 경계 거리를 설정하는 단계; 및 상기 특징 공간에서 상기 기준 위치로부터 상기 다수의 누적 특징 각각 까지의 거리인 명료성 메트릭과 상기 명료성 경계 거리에 따라 상기 다수의 특징 데이터 중 다음 라운드의 특징을 전송할 특징 데이터를 선택하는 단계를 포함한다.

발명의 효과

[0020] 본 개시의 엣지 학습 장치 및 방법은 IoT 기기가 수집한 학습 데이터에서 추출된 특징을 중요도와 신경망 모델의 학습 상태에 따라 부분적으로 인가받아 효과적인 학습을 수행함으로써, 학습을 위해 오프로드되어야 하는 학습 데이터의 양을 크게 줄일 뿐만 아니라, 오프로드되어야 하는 학습 데이터를 프리페치하여 에너지 소모를 저감시킬 수 있다.

도면의 간단한 설명

[0021] 도 1은 일 실시예에 따른 엣지 학습을 위한 엣지 학습 시스템의 개략적 구성을 나타낸다.

도 2는 IoT 기기가 데이터 샘플 임베딩을 설명하기 위한 도면이다.

도 3은 엣지 서버가 오프로드 요청 대상 특징을 선택하는 개념을 설명하기 위한 도면이다.

도 4는 일 실시예 따른 엣지 학습 장치의 동작에 따라 개략적으로 구분한 구성을 나타낸다.

도 5는 이진 SVM에서 불명료 분류 샘플과 명료 분류 샘플을 구분하기 위한 클래스 분류 경계를 설정하는 기법을 설명하기 위한 도면이다.

도 6은 프리페치 수행 여부에 따라 특징이 전송되는 기간의 변화를 나타낸다.

도 7은 일 실시예에 따른 엣지 학습 방법을 나타낸다.

도 8은 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 설명하기 위한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0022] 이하, 도면을 참조하여 본 개시의 실시예에 따른 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는

예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0023] 본 개시의 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 실시예의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 일 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다. 또한, 명세서에 기재된 "...부", "...기", "모듈", "블록" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0024] 도 1은 일 실시예에 따른 엣지 학습을 위한 엣지 학습 시스템의 개략적 구성을 나타내고, 도 2는 IoT 기기가 데이터 샘플 임베딩을 설명하기 위한 도면이며, 도 3은 엣지 서버가 오프로드 요청 대상 특징을 선택하는 개념을 설명하기 위한 도면이다.

[0025] 도 1을 참조하면 엣지 학습 시스템은 다수의 IoT 기기(11), 엣지 서버(12) 및 다수의 엣지 기기(13)를 포함할 수 있다.

[0026] 여기서는 학습 대상인 신경망 모델이 엣지 서버(12)에 포함되어 엣지 서버(12)가 학습을 수행하는 중앙 집중식 엣지 학습 시스템인 것으로 가정하며, 신경망 모델은 인가된 이미지의 클래스를 분류하는 분류 모델인 것으로 가정하여 설명하지만 이에 한정되지 않는다. 그리고 IoT 기기(11)는 일 예로 다양한 센서로 구현될 수 있으며, 여기서는 이해의 편의를 위하여, IoT 기기(11)와 엣지 기기(13)를 구분하여 별도로 도시하였으나, IoT 기기(11)와 엣지 기기(13)를 통합하여 엣지 단말이라 할 수도 있다.

[0027] 다수의 IoT 기기(11)는 엣지 학습을 위해 진리값인 클래스가 레이블된 이미지 데이터를 데이터 샘플(d)로서 수집하고, 엣지 서버(12)는 다수의 IoT 기기(11)가 수집한 데이터 샘플(d)과 데이터 샘플(d)에 레이블된 진리값을 기반으로 신경망 모델을 학습시킨다. 신경망 모델이 학습되면 엣지 서버(12)는 다수의 엣지 기기(13)에서 인가되는 이미지 데이터의 클래스를 분류하고 분류 결과를 엣지 기기(13)로 회신할 수 있다.

[0028] 구체적으로 다수의 IoT 기기(11)는 각각 엣지 학습을 위한 데이터 샘플(d)을 수집한다. 그리고 도 2에 도시된 바와 같이, 데이터 샘플(d)을 임베딩하여 특징 데이터(x)를 추출하고, 추출된 특징 데이터(x)를 엣지 서버(12)로 오프로드(offload)한다. 다수의 IoT 기기(11) 각각은 사전에 제공된 임베딩 모델을 이용하여 수집된 데이터 샘플(d)을 임베딩함으로써 특징 데이터(x)를 추출할 수 있다. 여기서 임베딩 모델은 사전 학습된 신경망 모델로서 일 예로 엣지 서버(12)에서 전달될 수 있으나, 이에 한정되지 않는다.

[0029] 일반적으로 IoT 기기(11)는 낮은 성능으로 인해 신경망 모델을 직접 구동하거나 학습시킬 수는 없다. 그러나 대부분의 IoT 기기(11)는 낮은 성능일지라도 학습 데이터 샘플(d)을 가상의 특징 공간에 임베딩하여 특징 데이터(x)를 추출하는 임베딩 모델을 구동할 수는 있다. 이에 IoT 기기(11)는 도 1에 도시된 바와 같이, 임베딩 모델을 이용하여 고차원의 원시 데이터 샘플(d)을 저차원의 특징 공간에 맵핑함으로써 데이터양이 저감된 특징 데이터(x)를 획득할 수 있다.

[0030] 여기서는 다수의 IoT 기기(11)가 수집된 M개의 데이터 샘플(d_m)을 임베딩하여 M개의 특징 데이터(x_m)를 엣지 서버(12)로 오프로드할 수 있다. 여기서 m($m \in \{1, 2, \cdots, M\}$)은 샘플 인덱스이며, 임베딩 모델의 임베딩 동작을 임베딩 함수($\mathcal{F}$)로 표현할 때, 특징 데이터(x_m)는 수학식 1에 따라 획득될 수 있다.

수학식 1

[0031] $$\mathbf{x}_m = \mathcal{F}(\mathbf{d}_m)$$

[0032] IoT 기기(11)는 수집된 길이(D)의 고차원 데이터 샘플(d_m)을 임베딩 모듈의 임베딩 함수($\mathcal{F}$)로 임베딩하여 F

(여기서 F < D)개의 특징을 포함한 벡터 형식($x_m = [x_{m,1}, \cdots, x_{m,F}]$)의 특징 데이터($x_m$)를 획득한다. 따라서 엣지 서버(12)로 오프로드되어야 하는 데이터 크기가 1차적으로 줄어들게 된다. 일 예로 임베딩 모델은 D개의 픽셀로 구성된 이미지 데이터인 데이터 샘플(d_m)을 임베딩하여, F개 특징($x_{m,1}, \cdots, x_{m,F}$)이 원소로 구성되는 특징 데이터(x_m)를 획득할 수 있다. 따라서 특징 데이터(x_m)는 F개의 특징($x_{m,1}, \cdots, x_{m,F}$)을 원소로 갖는 F 길이의 벡터로 표현될 수 있으며, F개의 특징($x_{m,1}, \cdots, x_{m,F}$) 각각의 특징값이 특징 공간의 서로 다른 차원에서의 위치값을 나타낼 수 있다.

[0033] 이때 IoT 기기(11)의 임베딩 모듈은 도 2와 같이, 임베딩 과정에서 특징 데이터(x_m)의 F개의 특징($x_{m,1}, \cdots, x_{m,F}$)이 중요도에 따라 순차 정렬되도록 할 수 있다. 데이터 샘플(d_m)을 특징 데이터(x_m)로 임베딩하는 과정에서 중요도를 고려하는 임베딩 모듈은 다양하게 공개되어 있으며, 여기서는 일 예로, 주성분 분석(Principle Component Analysis: 이하 PCA)으로 알려진 임베딩 모델을 이용하는 것으로 가정한다.

[0034] 특징 데이터(x_m)에서 F개의 특징($x_{m,1}, \cdots, x_{m,F}$)이 중요도에 따라 순차 정렬되어 있으므로, 제i($i \in \{1, \cdots, F\}$) 특징($x_{m,i}$)의 중요도가 제j(여기서, $i < j$, $j \in \{1, \cdots, F\}$) 특징($x_{m,j}$)의 중요도보다 높다고 볼 수 있다.

[0035] 그리고 다수의 IoT 기기(11)는 획득된 특징 데이터(x_m)를 그대로 엣지 서버(12)로 오프로드하는 대신, 중요도에 따라 정렬된 F개의 특징($x_{m,1}, \cdots, x_{m,F}$) 중 가장 높은 중요도를 갖는 제1 특징($x_{m,1}$)을 우선 엣지 서버(12)로 오프로드 한다. 그리고 이후에는 엣지 서버(12)의 요청에 따라 중요도에 따라 정렬된 나머지 특징($x_{m,2}, \cdots, x_{m,F}$)을 선택적, 순차적으로 오프로드한다. 즉 IoT 기기(11)는 특징 데이터(x_m)에서 엣지 서버(12)가 요청하는 특징만을 오프로드한다. 이때 다수의 IoT 기기(11)는 특징($x_{m,k}$)(예를 들면, 제1 특징($x_{m,1}$))과 함께 진리값을 엣지 서버(12)로 오프로드할 수 있다.

[0036] 엣지 서버(12)는 학습 대상이 되는 신경망 모델을 학습시키는 엣지 학습 장치로 동작하며, 다수의 IoT 기기(11)에서 획득한 특징 데이터(x_m)를 이용하여 신경망 모델을 학습시킨다. 다만 엣지 서버(12)는 신경망 모델을 학습시킬 때, 특징 데이터(x_m)의 모든 특징($x_{m,1}, \cdots, x_{m,F}$)이 인가되지 않더라도 신경망 모델이 요구되는 성능을 나타내도록 학습시킬 수 있다. 비록 데이터 샘플(d_m)보다 특징 데이터(x_m)의 데이터양이 더 적을지라도, 다수의 IoT 기기(11)가 획득된 특징 데이터(x_m) 전체를 엣지 서버(12)로 오프로딩하는 것은 에너지 소모가 크다. 따라서 특징 데이터(x_m)의 모든 특징($x_{m,1}, \cdots, x_{m,F}$)을 오프로드하는 것은 비효율적이다.

[0037] 엣지 서버(12)는 특징 데이터(x_m)의 일부 특징만으로도 신경망 모델이 요구되는 동작을 수행하도록 학습시킬 수 있다. 예로서 신경망 모델이 상기한 가정과 같이, 이미지의 클래스를 분류하는 분류 모델인 경우, 엣지 서버(12)는 특징 데이터(x_m)에서 신경망 모델이 인가된 이미지의 클래스를 식별할 수 있는 수준까지의 일부 특징만을 인가받아도 무방하다.

[0038] 또한 상기한 바와 같이, 특징 데이터(x_m)를 구성하는 특징($x_{m,1}, \cdots, x_{m,F}$)을 중요도에 따라 이미 정렬되어 있으며, PCA기반 임베딩 모델의 특성을 고려할 때, 중요도가 높은 특징, 즉 정렬된 순서가 앞선 특징이 학습 유효성이 높은 특징이라 할 수 있다. 따라서 i번째 특징($x_{m,i}$)이 j(여기서 $i < j$)번째 특징($x_{m,j}$)보다 신경망 모델을 더욱 효율적으로 학습시킬 수 있다.

[0039] 그러므로 엣지 서버(12)는 다수의 IoT 기기(11)에서 획득된 M개의 특징 데이터($x_1, \cdots, x_M$) 각각에서 가장 중요한 제1 특징($x_{1,1}, \cdots, x_{M,1}$)을 우선 인가받아 학습하고, 이후로는 신경망 모델이 학습된 수준에 따라 나머지 특징($x_{m,2}, \cdots, x_{m,F}$)을 순차적으로 요청한다. 엣지 서버(12)는 신경망 모델이 요구되는 수준으로 학습된 것으로 판단될 때까지, 순차적으로 나머지 특징($x_{m,2}, \cdots, x_{m,F}$)을 요청하여 인가받는다.

[0040] 여기서는 일 예로 엣지 서버(12)가 특징 데이터(x_m)의 F개의 특징($x_{m,1}, \cdots, x_{m,F}$) 중 k($k < F$)개까지의 특징($x_{m,1}, \cdots, x_{m,k}$)을 인가받아 신경망 모델을 학습시키는 것으로 가정한다. 따라서 IoT 기기(11)는 엣지 서버(12)의 요청에 따라 제1 내지 제k 특징($x_{m,1}, \cdots, x_{m,k}$)을 순차적으로 오프로드할 수 있다. 여기서 IoT 기기(11)가

엣지 서버(12)의 요청에 따라 하나의 특징($x_{m,k}$)을 오프로드하는 구간을 라운드라고 하며, IoT 기기(11)는 k 라운드 동안 k개까지의 특징($x_{m,1}$, …, $x_{m,k}$)을 순차적으로 엣지 서버(12)로 오프로드 한다. 엣지 서버(12)는 각 라운드에서 누적되어 오프로드된 특징($x_{m,1}$, …, $x_{m,k}$)을 이용하여 신경망 모델을 학습시키며, 특징 데이터(x_m)의 F개의 특징($x_{m,1}$, …, $x_{m,F}$)에서 신경망 모델의 학습에 이용되도록 엣지 서버(12)로 오프로드되어 누적되는 제1 내지 제k 특징($x_{m,1}$, …, $x_{m,k}$)을 여기서는 누적 특징($x_m^{(k)} = [(x_{m,1}, \cdots, x_{m,k}]^T$)이라고 한다.

[0041] 상기한 바와 같이, IoT 기기(11)가 M개의 특징 데이터(x_1, …, x_M) 각각의 F개의 특징($x_{m,1}$, …, $x_{m,F}$)을 모두 오프로딩하지 않고, k개까지의 특징($x_{m,1}$, …, $x_{m,k}$)만을 순차적으로 오프로딩할 수 있으므로, IoT 기기(11)는 오프로딩을 위한 에너지 소비를 크게 줄일 수 있으며, 네트워크 오버헤드를 줄일 수 있다.

[0042] 뿐만 아니라 엣지 서버(12)는 에너지 효율적으로 신경망 모델을 학습시키기 위해, 라운드 횟수가 증가함에 따라, M개의 특징 데이터(x_1, …, x_M) 중에서 각 라운드에서 특징이 오프로드되어야 하는 특징 데이터를 선택할 수 있다. 예로서, 엣지 서버(12)는 M개의 특징 데이터(x_1, …, x_M)의 M개의 제k 특징($x_{1,k}$, …, $x_{M,k}$) 중에서 k 라운드에 오프로드되어야 하는 제k 특징을 선택할 수 있다.

[0043] 도 3을 참조하여 설명하면, 도 3에서 파란색 ○은 진리값이 제1 클래스인 다수의 제1 데이터 샘플에서 획득된 제1 특징 데이터의 특징들을 나타내고, 빨간색 X는 진리값이 제2 클래스인 다수의 제2 데이터 샘플에서 획득된 제2 특징 데이터의 특징들을 나타낸다.

[0044] 그리고 도 3에서 (a)는 제1 라운드(k = 1)에서 다수의 특징 데이터(x_m) 각각에서 제1 특징($x_{m,1}$)만이 오프로딩되어 신경망 모델을 학습시킨 경우를 나타내고, (b)는 제2 라운드(k = 2)에서 제2 특징($x_{m,2}$)이 추가로 오프로딩되어 제1 및 제2 특징($x_{m,1}$, $x_{m,2}$)으로 신경망 모델을 학습시킨 경우를 나타내며, (c)는 제3 라운드(k = 3)에서 제3 특징($x_{m,3}$)이 추가로 오프로딩되어 제1 내지 제3 특징($x_{m,1}$ ~ $x_{m,3}$)으로 신경망 모델을 학습시킨 경우를 나타낸다. 도 3의 (a)에서는 제1 특징($x_{m,1}$)만이 인가되었으므로, 제1 특징($x_{m,1}$)의 특징값을 기반으로 각 특징 데이터(x_m)가 1차원 공간상에 투영되어 표현되었으며, (b)에서는 제1 특징($x_{m,1}$)과 더불어 제2 특징($x_{m,2}$)이 추가되었으므로 제1 특징($x_{m,1}$) 및 제2 특징($x_{m,2}$)의 특징값을 기반으로 각 특징 데이터(x_m)가 2차원 공간상에 투영되어 표현되었다. 그리고 (c에서는 제1 및 제2 특징($x_{m,1}$, $x_{m,2}$)에 제3 특징($x_{m,3}$)이 추가되었으므로 제1 내지 제3 특징($x_{m,1}$ ~ $x_{m,3}$)의 특징값을 기반으로 각 특징 데이터(x_m)가 3차원 공간상에 투영되어 표현되었다. 즉 도 3에서는 추가로 오프로드되는 특징의 개수가 증가됨에 따라 임베딩되는 특징 공간의 차원이 함께 증가하는 것으로 도시하였다.

[0045] 도 3의 (a)에 도시된 바와 같이, 제1 및 제2 클래스의 2개의 클래스가 진리값으로 레이블된 다수의 제1 및 제2 데이터 샘플(d_m)에서 획득된 다수의 제1 및 제2 특징 데이터(x_m) 각각에서 가장 중요도가 높은 제1 특징($x_{m,1}$)만이 오프로딩되어 신경망 모델을 학습시킬 지라도, 신경망 모델은 일정 수준까지는 제1 및 제2 데이터 샘플(d_m)이 제1 클래스인지 또는 제2 클래스인지 분류할 수 있다. 신경망 모델은 학습에 의해, 제1 및 제2 데이터 샘플(d_m)이 제1 클래스인지 또는 제2 클래스인지 분류할 수 있으며, 이때 특징 공간에서 제1 및 제2 데이터 샘플(d_m)이 제1 클래스인지 또는 제2 클래스인지 분류하는 클래스 분류 기준을 여기서는 클래스 분류 경계(또는 클래스 문턱값)(cb)라고 한다.

[0046] 다만 제1 및 제2 특징 데이터(x_m)의 제1 특징($x_{m,1}$)만으로는 신경망 모델이 충분하게 학습되지 않으며, 이에 신경망 모델의 신경망 연산 결과와 진리값 사이에 오차가 발생할 수 있다. 이러한 오차는 특징 공간에서 제1 및 제2 특징 데이터(x_m)의 제1 특징($x_{m,1}$)이 클래스 분류 경계(cb)를 기준으로 주변 일정 거리까지의 영역에 위치하는 경우에 주로 나타나게 되며, 이와 같이 데이터 샘플(d_m)의 클래스를 정확하게 분류하지 못하고 모호하게 판별되는 불명료 영역(또는 ACS(Ambiguous Classified Sample) 영역)이라고 할 수 있다. 그리고 불명료 영역을 제외한 나머지 영역은 제1 및 제2 특징 데이터(x_m)의 클래스가 명확하게 분류되는 명료 영역이라고 할 수 있다.

[0047] 도 3에서는 오프로드되는 특징의 개수 증가에 따라 임베딩 공간의 차원이 함께 증가되었으므로, 클래스 분류 경계(cb) 또한 (a) 내지 (c)에서 점, 선 및 면의 형태로 확장될 수 있으며, 도 3에서 빨간색 선으로 표현된 하이

퍼플레인(Hyperplane)이 클래스 분류 경계(cb)를 나타낸다.

[0048] 다수의 제1 및 제2 특징 데이터(x_m)에서 오프로드된 제1 특징($x_{m,1}$)이 이미 명료 영역에 존재하는 특징 데이터(x_m)의 경우, 제1 특징($x_{m,1}$)만으로도 신경망 모델이 이미 해당 특징 데이터(x_m)의 클래스를 충분히 정확하게 분류할 수 있으므로 추가적인 학습의 의미가 크지 않다. 반면, 오프로드된 제1 특징($x_{m,1}$)이 불명료 영역에 존재하는 특징 데이터(x_m)의 경우, 신경망 모델이 해당 특징 데이터(x_m)의 클래스를 안정적으로 식별하지 못한다는 것을 의미한다. 따라서 불명료 영역에 존재하는 특징 데이터(x_m)의 특징들은 신경망 모델의 학습 성능을 향상시키기에 매우 효과적인 데이터라고 볼 수 있다.

[0049] 이에 엣지 서버(12)는 다수의 제1 및 제2 특징 데이터(x_m)에서 오프로드된 제1 특징($x_{m,1}$)이 불명료 영역에 포함된 특징 데이터(x_m)를 선택하고, 선택된 특징 데이터(x_m)에서 다음 순서의 제2 특징($x_{m,2}$)을 선택적으로 IoT 기기(11)로 요청한다. 그리고 IoT 기기(11)가 선택된 특징 데이터(x_m)의 제2 특징($x_{m,2}$)을 오프로드하면, 엣지 서버(12)는 이전 오프로드된 제1 특징($x_{m,1}$)과 함께 추가로 오프로드된 제2 특징($x_{m,2}$)을 기반으로 신경망 모델을 추가 학습시키고, 다시 불명료 영역에 포함되는 특징 데이터(x_m)를 판별하여 선택한다. 그리고 선택된 특징 데이터(x_m)의 다음 순서의 제3 특징($x_{m,3}$)의 오프로딩을 요청하는 과정을 반복하여 신경망 모델을 학습시킨다.

[0050] 즉 다수의 특징 데이터(x_m)에서 제1 특징($x_{m,1}$)을 제외한 나머지 특징은 학습된 신경망 모델이 현재까지 오프로드된 누적 특징($x_m^{(k)}$)으로부터 각 데이터 샘플(d_m)의 클래스를 명료하게 판별할 수 있는지, 즉 누적 특징($x_m^{(k)}$)이 명료 영역에 포함되어 있는지 또는 불명료 영역에 포함되어 있는지 여부에 따라 엣지 서버(12)가 선택적으로 요청하여 오프로딩되도록 할 수 있다.

[0051] 따라서 엣지 서버(12)가 선택하여 요청하는 특징의 개수는 라운드가 증가할수록 점차적으로 줄어들게 된다. 도 3 에서는 다음 특징($x_{m,2}$, $x_{m,3}$)을 추가로 인가받는 특징 데이터(x_m)만을 점선으로 별도 표시하였다. 도 3의 (a)에서와 같이 초기에는 13개의 특징 데이터(x_m)의 13개의 제1 특징($x_{m,1}$)이 모두 오프로딩되었다. 그러나 (b)에서 제2 특징($x_{m,2}$)을 추가로 요청하도록 선택되는 특징 데이터(x_m)의 개수는 8개이며, (c)에서 제3 특징($x_{m,3}$)을 추가로 요청하도록 선택되는 특징 데이터(x_m)의 개수는 6개로 점차 줄어들었음을 알 수 있다.

[0052] 이와 같이, 엣지 서버(12)는 현재 라운드(일 예로 k 라운드)까지 특징($x_{m,k}$)이 오프로드된 불명료 분류 샘플(Ambiguous Classified Sample: ACS)의 집합인 불명료 샘플 집합($\mathbb{S}^{(k)}$, 여기서 $m \in \mathbb{S}^{(k)}$)에서 다시 명료 분류 샘플(Clearly Classified Sample: 이하 CCS)과 불명료 분류 샘플(ACS)을 구분하고, 구분된 불명료 분류 샘플(ACS)을 다음 신경망 모델 학습을 위한 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)으로 업데이트할 수 있다. 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 명료 분류 샘플(CCS)이 반복적으로 제외되어 불명료 분류 샘플(ACS)만이 남게 되므로, 업데이트되는 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)의 원소 개수는 점차적으로 줄어들게 된다.

[0053] 따라서 엣지 서버(12)가 신경망 모델의 학습을 위해 M개의 특징 데이터(x_m)에서 제1 내지 제k 특징($x_{m,1}$, …, $x_{m,k}$)이 필요로 할지라도, 실제 IoT 기기(11)가 엣지 서버(12)로 오프로드하는 특징의 개수는 $M \times k$ 보다도 더 적은 개수가 된다. 따라서 특징 오프로딩을 위한 에너지 소모를 크게 줄일 수 있다.

[0054] 결과적으로 도 1에 도시된 엣지 학습 시스템에서는 다수의 IoT 기기(11)가 각각 수집한 데이터 샘플(d_m)을 임베딩하여 특징 데이터(x_m)를 획득함으로써 오프로드되어야 하는 데이터양을 1차로 줄이며, 엣지 서버(12)가 특징 데이터(x_m)를 구성하는 특징($x_{m,1}$, …, $x_{m,F}$) 중 신경망 모듈의 학습에 필요한 최소의 특징을 선택하여 IoT 기기(11)로 오프로드를 요청함으로써, 오프로드되어야 하는 데이터양을 2차로 다시 줄일 수 있다. 따라서 고차원의 데이터 샘플(d_m)을 그대로 오프로드하는 경우에 비해 IoT 기기(11)가 오프로드에 소모하는 에너지를 크게 저감시킬 수 있다.

[0055] 다만 상기한 바와 같이 에너지 효율적으로 신경망 모델을 학습시키기 위해서는 엣지 서버(12)가 현재까지 오프

로딩된 특징들로부터 불명료 영역에 위치하는 불명료 분류 샘플(ACS)과 명료 영역에 위치하는 명료 분류 샘플(CCS)을 구분할 수 있어야 하며, 이를 위해서는 명료 영역과 불명료 영역 구분하는 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정할 수 있어야 한다.

[0056] 엣지 서버(12)가 클래스 분류 경계(cb)와 불명료 영역을 설정하는 상세한 방식은 후술하도록 한다.

[0057] 도 4는 일 실시예 따른 엣지 학습 장치의 동작에 따라 개략적으로 구분한 구성을 나타낸다.

[0058] 도 4에 도시된 엣지 학습 장치는 엣지 서버(12)에 구비된 장치로서, 특징 획득 모듈(21), 신경망 모델(22), 학습 모듈(23), 불명료 데이터 판별 모듈(24) 및 특징 요청 모듈(27)을 포함할 수 있다.

[0059] 특징 획득 모듈(21)은 다수의 IoT 기기(11)가 오프로드한 특징 데이터(x_m)의 특징($x_{m,k}$)을 인가받아 획득하고, 획득된 특징을 학습 대상인 신경망 모델(22)로 전달한다. 여기서 특징 데이터(x_m)는 다수의 IoT 기기(11)가 수집한 다수의 데이터 샘플(d_m)을 각각 임베딩 모델로 임베딩하여 획득되며, 각 특징 데이터(x_m) 는 중요도에 따라 정렬된 다수의 특징($x_{m,k}$)으로 구성된다.

[0060] IoT 기기(11)는 초기에 다수의 IoT 기기(11)에서 획득된 모든 데이터 샘플(d)의 제1 특징($x_{m,1}$)을 오프로딩하며, 이후로는 엣지 학습 장치가 요청한 특징($x_{m,k}$)을 오프로딩할 수 있다. 이때 특징 획득 모듈(21)은 라운드 단위로 정렬된 순서에 따라 IoT 기기(11)에서 오프로드된 특징($x_{m,1}$, …, $x_{m,k}$)을 데이터 식별자(m)에 따라 구분하여 누적 저장할 수 있으며, 각 라운드에서 누적 저장된 누적 특징($x_m^{(k)} = [x_{m,1}, .., x_{m,k}]^T$)을 신경망 모델(22)로 전달한다.

[0061] 그리고 특징 획득 모듈(21)은 특징 데이터(x_m)의 특징($x_{m,k}$)과 함께 각 데이터 샘플(d_m)에 레이블된 진리값을 인가받을 수 있다. 진리값은 제1 특징($x_{m,1}$)과 함께 오프로딩되거나 별도로 오프로딩될 수 있으며, 특징 획득 모듈(21)은 획득된 진리값을 학습 모듈(23)로 전달할 수 있다.

[0062] 신경망 모델(22)은 학습에 따른 신경망 연산을 수행한다. 신경망 모델(22)은 학습 시에 특징 획득 모듈(21)로부터 다수의 데이터 샘플(d_m)에 대한 누적 특징($x_m^{(k)}$)을 입력으로 인가받아 신경망 연산하여, 연산 결과를 출력한다. 즉 학습 시 신경망 모델(22)은 라운드가 반복될수록 더 많은 특징($x_{m,1}$, …, $x_{m,k}$)이 누적된 누적 특징($x_m^{(k)}$)을 입력으로 인가받아 신경망 연산을 수행한다. 여기서는 상기한 바와 같이 신경망 모델이 분류 모델인 것으로 가정하여 설명하며, 이에 신경망 모델(22)은 입력된 누적 특징($x_m^{(k)}$) 각각에 대해 분류된 클래스를 연산 결과로 출력할 수 있다.

[0063] 신경망 모델(22)은 SVM(Support Vector Machine) 또는 DNN(Deep Neural Network) 등으로 구현될 수 있으나, 이에 제한되지 않고 다양한 신경망 모델로 구현될 수 있다.

[0064] 학습 모듈(23)은 학습 시에 신경망 모델(22)의 연산 결과와 특징 획득 모듈(21)에서 전달된 진리값을 기반으로 연산 결과에 대한 손실을 계산하고, 계산된 손실을 신경망 모델(22)로 역전파하여 신경망 모델(22)을 학습시킨다.

[0065] 불명료 데이터 판별 모듈(24)은 현재까지 획득된 누적 특징($x_m^{(k)}$)을 기반으로 학습된 신경망 모델(22)이 데이터 샘플(d_m)에 대한 클래스 분류를 명확하게 할 수 있는 명료 영역과 불명확하게 할 수 있는 불명료 영역을 구분하기 위한 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정하고, 설정된 명료성 경계 거리($\bar{\beta}^{(k)}$)를 기반으로, 현재 불명료 샘플 집합($\mathbb{S}^{(k)}$)에 포함된 누적 특징($x_m^{(k)}$) 각각이 불명료 영역에 포함되는지 여부를 판별한다.

[0066] 불명료 데이터 판별 모듈(24)은 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정하는 명료성 경계 설정 모듈(25)과 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 누적 특징($x_m^{(k)}$)에서 불명료 영역에 포함되는 누적 특징($x_m^{(k)}$)을 추출하는 불명료 데이터 선택 모듈(26)을 포함할 수 있다.

[0067] 도 3에서와 같이, 클래스 분류 경계(cb)가 제1 내지 제k 특징($x_{m,1}$, …, $x_{m,k}$)이 순차적으로 오프로딩됨에 따라 증가되는 차원 상의 하이퍼플레인으로 표현될 수 있으므로, 제1 내지 제k 특징($x_{m,1}$, …, $x_{m,k}$)이 오프로드된 k번째 라운드에서 불명료 영역과 명료 영역을 구분하는 명료성 경계 거리($\bar{\beta}^{(k)}$)는 클래스 분류 경계(cb)를 나타내는 하이퍼플레인으로부터의 거리로 설정될 수 있다. 즉 클래스 분류 경계(cb)를 나타내는 하이퍼플레인은 특징 공간 상의 기준 위치를 나타내고, 명료성 경계 거리($\bar{\beta}^{(k)}$)는 특징 공간에서 클래스 분류 경계(cb)까지의 거리로 표현된다.

[0068] 따라서 명료성 경계 거리($\bar{\beta}^{(k)}$)가 설정되면, 수학식 2와 같이 하이퍼플레인으로부터 명료성 경계 거리($\bar{\beta}^{(k)}$)까지보다 짧은 거리에 위치하는 누적 특징($x_m^{(k)}$)을 다음 라운드(k+1)에서의 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)에 포함시킬 수 있다.

수학식 2

[0069] $$\mathbb{S}^{(k+1)} = \{m \mid \beta_m^{(k)} \leq \bar{\beta}^{(k)}, m \in \mathbb{S}^{(k)}\}$$

[0070] 여기서 $\beta_m^{(k)}$는 누적 특징($x_m^{(k)}$)의 명료성 수준을 나타내는 명료성 메트릭(Metric of Clarity: MoC)으로, 특징 공간에서 기준 위치인 클래스 분류 경계(cb)로부터 누적 특징($x_m^{(k)}$)까지의 거리로 볼 수 있다.

[0071] 이에 명료성 경계 설정 모듈(25)은 신경망 모델(22)이 특징($x_{m,i}$)에 대해 신경망 연산을 수행하는 과정에서의 연산 결과가 서로 다르게 출력될 수 있도록 하는 특징 데이터(x_m) 사이의 경계, 즉 누적 특징($x_m^{(k)}$)에 대한 클래스 분류 경계(cb)인 하이퍼플레인을 추출하고, 추출된 클래스 분류 경계(cb)를 기준으로 하는 거리값으로 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정할 수 있다.

[0072] 일 예로 신경망 모델(22)이 SVM으로 구현되는 경우, 명료성 경계 설정 모듈(25)은 누적 특징($x_m^{(k)}$)에 대한 클래스 분류 경계(cb)는 수학식 3과 같이 획득될 수 있다.

수학식 3

[0073] $$(\mathbf{w}^{(k)})^T \mathbf{x}^{(k)} + b^{(k)} = 0$$

[0074] 여기서 $w^{(k)}$는 클래스 분류 경계(cb)인 하이퍼플레인에 대해 수직인 벡터를 나타내고, $b^{(k)}$는 오프셋을 나타낸다.

[0075] 상기한 바와 같이, 명료성 경계 거리($\bar{\beta}^{(k)}$)는 신경망 모델(22)이 인가된 특징 누적 특징($x_m^{(k)}$)이 불명료 분류 샘플(ACS)인지 명료 분류 샘플(CCS)인지를 판별하기 위한 명료 영역과 불명료 영역을 구분하는 경계값이며, 신경망 모델이 2개의 클래스(c_0, c_1)를 구분하는 이진 SVM(binary SVM)으로 구현되는 경우, 도 5에 도시된 바와 같이, 클래스 분류 경계(cb)에 따라 구분되는 각 클래스의 확률 분포를 가우시안 혼합 모델(Gaussian Mixture Model: 이하 GMM)을 이용하여 근사화할 수 있다.

[0076] 도 5는 이진 SVM에서 불명료 분류 샘플과 명료 분류 샘플을 구분하기 위한 클래스 분류 경계를 설정하는 기법을 설명하기 위한 도면이다.

[0077] 도 5에 도시된 바와 같이, 각 클래스에서의 확률적 분포(stochastic distribution)는 GMM을 이용하여 k-변량 가우시안(k-variate Gaussian) 형식으로 근사화될 수 있다. 그리고 근사화된 두 클래스의 확률적 분포 사이에는 중첩 영역이 나타나게 되며, 중첩 영역에서 데이터가 오분류될 가능성이 높다고 볼 수 있다.

[0078] 이때, GMM으로 근사화된 이웃한 클래스들의 확률 분포에서 양측 꼬리(Tail)의 절단 포인트(truncated point) 외곽 영역, 즉 절단 영역은 무시 가능한 영역으로 제외하고, 나머지 중첩되는 영역에 위치하는 누적 특징($x_m^{(k)}$)이 모두 포함될 수 있도록 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정할 수 있어야 한다.

[0079] 명료성 경계 설정 모듈(25)은 이진 SVM 신경망의 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설정하기 위해, 각 클래스(c)의 중심에서 k개의 특징으로 구성된 k-변량인 누적 특징($x_m^{(k)}$)의 위치까지의 거리를 나타내는 마할노비스 거리(Mahalanobis Distance: 이하 MD)($\delta_c^{(k)}$)를 수학식 4에 따라 계산할 수 있다.

수학식 4

[0080] $$\delta_c^{(k)} = \sqrt{\left(\mathbf{x}^{(k)} - \boldsymbol{\mu}_c^{(k)}\right)^T \cdot \left(\boldsymbol{\Sigma}_c^{(k)}\right)^{-1} \cdot \left(\mathbf{x}^{(k)} - \boldsymbol{\mu}_c^{(k)}\right)}$$

[0081] 여기서 $\mu_c^{(k)}$와 $\sum_c^{(k)}$는 각 클래스(c)에서 누적 특징($x^{(k)}$) 분포의 평균 및 공분산 행렬(covariance matrix)로 GMM을 통해 획득할 수 있다.

[0082] 그리고 MD($\delta_c^{(k)}$)는 스케일-프리(scale-free) 랜덤 변수(scale-free random variable)로서, MD($\delta_c^{(k)}$)의 누적분포함수(Cumulative Distribution Function: CDF)의 값이 지정된 누적분포 문턱값(p_{th}, 예를 들면, p_{th} = 0.95)이 되는 거리를 절단 영역의 기준이 되는 절단 포인트를 설정하기 위한 문턱 거리($\bar{\delta}_c^{(k)}$)로 설정한다면, 문턱 거리($\bar{\delta}_c^{(k)}$)는 수학식 5를 만족해야 한다.

수학식 5

[0083] $$\Pr\left[\delta_c^{(k)} \leq \bar{\delta}_c^{(k)}\right] = p_{\text{th}}$$

[0084] 여기서 Pr은 확률 함수를 나타낸다.

[0085] 한편, MD($\delta_c^{(k)}$)의 제곱이 자유도가 k인 카이제곱 분포(chi-square distribution)를 따르므로, 거리(r)에 따른 MD($\delta_c^{(k)}$)의 CDF는 수학식 6과 같이 정의될 수 있다.

수학식 6

[0086] $$\mathcal{G}(r;k) = \Pr\left[\delta_c^{(k)} \leq r\right] = \frac{\gamma\left(\frac{k}{2}, \frac{r}{2}\right)}{\Gamma\left(\frac{k}{2}\right)}$$

[0087] 여기서 $\Gamma(\cdot)$는 $\Gamma(k) = \int_0^\infty t^{k-1}e^{-t}dt$로 정의되는 감마 함수이고, $\gamma(\cdot)$는 $\gamma(k,r) = \int_0^r t^{k-1}e^{-t}dt$로 정의되는 하위 불완전 감마 함수이며, t는 감마 함수를 정의하기 위한 하이퍼파라미터이고, e는 지수함수(Exponential function)이다.

[0088] 수학식 5 및 6으로부터 문턱 거리($\bar{\delta}_c^{(k)}$)는 수학식 7과 같이 정리될 수 있다.

수학식 7

[0089] $$\bar{\delta}_c^{(k)} = \sqrt{\mathcal{G}^{-1}(p_{\text{th}}; k)}$$

[0090] 여기서 $\mathcal{G}^{-1}$는 자유도가 k인 카이 제곱분포의 역 CDF(inverse CDF)를 나타내고, 문턱 거리($\bar{\delta}_c^{(k)}$)는 스케일-프리 특성으로 인해, 클래스(c)에 무관하게 동일하게 설정될 수 있다.

[0091] 수학식 7에 따라 문턱 거리($\bar{\delta}_c^{(k)}$)가 설정되면, 각 클래스(c)에 대한 누적 특징($x^{(k)}$)의 분포(R_c)는 도 5에서 절단 포인트(truncated point) 내에 포함되는 영역으로서, 수학식 8에 따라 구분될 수 있다.

수학식 8

[0092] $$\mathcal{R}_c = \left\{ \mathbf{x}^{(k)} \in \mathbb{R}^{(k)} \mid \delta_c^{(k)} \leq \bar{\delta}^{(k)} \right\}, \quad c \in \{0, 1\}$$

[0093] 여기서는 이진 SVM을 가정하였으므로, 클래스(c)가 0과 1의 2개로 구분되는 것으로 가정하였다.

[0094] 이에 명료성 경계 거리($\bar{\beta}^{(k)}$)는 각 클래스의 누적 특징($x^{(k)}$)의 분포 중 절단 포인트 이내인 분포 영역(R_0, R_1)에서, 인접한 클래스와 중첩되는 영역이 모두 포함되도록 하는 최대 거리로 설정될 수 있다. 즉 중첩 영역에 포함되는 누적 특징($x^{(k)}$)이 최대가 되도록 수학식 9에 따라 설정될 수 있다.

수학식 9

[0095] $$\bar{\beta}^{(k)} = \max_{\mathbf{x}^{(k)} \in \mathcal{R}_0 \cap \mathcal{R}_1} \left| (\mathbf{w}^{(k)})^T \mathbf{x}^{(k)} + b^{(k)} \right| / \|\mathbf{w}^{(k)}\|$$

[0096] 여기서, $\|\cdot\|$는 L2 norm 함수를 나타낸다.

[0097] 이때, 각 클래스(c)의 공분산 행렬($\{\sum_c^{(k)}\}_{c\in\{0,1\}}$)이 서로 다르므로, 각 클래스에서의 누적 특징($x^{(k)}$)의 분포 영역(R_0, R_1)에서 절단 포인트에 의해 절단되는 절단 영역은 서로 비대칭이 된다. 분포 영역(R_0, R_1)의 절단 영역이 서로 비대칭이 되면, 신경망 모델(22)은 특정 클래스에 과적합될 수 있다. 이러한 과적합이 발생되는 것을 방지하기 위해, 도 5에 도시된 바와 같이, 명료성 경계 거리($\bar{\beta}^{(k)}$)는 두 클래스(c_0, c_1)에 대한 클래스 분류 경계(cb)로부터 동일한 거리로 설정할 수 있다.

[0098] 한편, 신경망 모델(22)이 DNN으로 구현되는 경우, 각 누적 특징($x_m^{(k)}$)에 대한 신경망 모델(22)의 연산 결과에 따른 사후 분포(posterior distribution)를 근사화하는 것은 매우 어렵다. 따라서 SVM과 같이 GMM을 활용하여 명료성 경계 거리($\bar{\beta}^{(k)}$)를 추출하기는 어렵다. 다만, 예측 오류 수, 즉 각 누적 특징($x_m^{(k)}$)에 대해 예측된 클래스($\hat{c}_m^{(k)}$)와 진리값(c_m) 사이에 오류($\hat{c}_m^{(k)} \neq c_m$)가 발생하는 횟수를 정규화하여 발생되는 오류의 분포를 기반으로 경험적으로 명료성 경계 거리($\bar{\beta}^{(k)}$)를 측정할 수 있다.

[0099] 각 누적 특징($x_m^{(k)}$)의 명료성 메트릭($\beta_m^{(k)}$)에 대비하여, 클래스 분류 경계(cb)로부터의 임의의 명료성 거리(β)에 따른 예측 오차($z^{(k)}(\beta)$)는 $z^{(k)}(\beta) = \Pr[\hat{c}_m \neq c_m \,,\, \beta_m^{(k)} \geq \beta)]$와 같이 표현될 수 있다. 예측 오차($z^{(k)}(\beta)$)는 명료성 거리($\beta$)에 따른 예측의 정확성을 나타내며, 신경망 모델(22)에 대한 학습이 잘 된 경우 특징

데이터(x_m)에 대한 명료성 거리(β)가 증가하게 되고, 예측 오차($z^{(k)}(\beta)$)는 명료성 거리(β)가 증가함에 따라 증가율이 감소하게 된다.

[0100] Pr은 확률 함수이며, 누적 특징($x_m^{(k)}$)의 명료성 메트릭($\beta_m^{(k)}$)의 결합 분포(joint distribution)는 수학식 10으로 표현될 수 있다.

수학식 10

[0101] $$z^{(k)}(\beta) = \frac{\sum_{m \in \mathbb{S}^{(k)}} \mathrm{I}(\hat{c}_m \neq c_m \ , \ \beta_m^{(k)} \geq \beta)}{|\mathbb{S}^{(k)}|}$$

[0102] 여기서 $\mathrm{I}(\cdot)$는 $(\cdot)$이 참이면 1을 반환하고, 거짓이면 0을 반환하는 지시 함수(indicator function)이다. 허용 가능한 오류 확률(z_{th}, 여기서 $z_{th} \in [0, 1]$)이 지정되면, 명료성 경계 거리($\bar{\beta}^{(k)}$)는 수학식 11과 같이 획득될 수 있다.

수학식 11

[0103] $$\bar{\beta}^{(k)} = \sup\{\beta \mid z^{(k)}(\beta) \geq z_{\mathrm{th}}\}$$

[0104] 여기서 $\sup\{\cdot\}$은 $\{\cdot\}$에 포함된 집합의 상계(bounded above)를 나타낸다.

[0105] 상기에서는 일 예로 신경망 모델(22)이 이진 SVM이나 DNN으로 구현되는 경우를 가정하여 설명하였으나, 신경망 모델(22)은 다른 모델로 구현될 수도 있으며, 이에 따른 명료성 경계 거리($\bar{\beta}^{(k)}$)는 공지된 다른 방식으로 획득될 수 있다.

[0106] 신경망 모델(22)에 따른 명료성 경계 거리($\bar{\beta}^{(k)}$)가 획득되면, 불명료 데이터 선택 모듈(26)은 현재 라운드의 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 불명료 분류 샘플(ACS)에 해당하는 누적 특징($x_m^{(k)}$)을 탐색하여, 다음 라운드를 위한 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)의 원소로 선택한다.

[0107] 불명료 데이터 선택 모듈(26)은 현재 라운드의 불명료 샘플 집합($\mathbb{S}^{(k)}$)에 포함된 누적 특징($x_m^{(k)}$)의 명료성 메트릭($\beta_m^{(k)}$)을 분석함으로써, 불명료 분류 샘플(ACS)인지 여부를 판별할 수 있다.

[0108] 구체적으로 신경망 모델(22)이 이진 SVM인 경우에, 누적 특징($x_m^{(k)}$)의 명료성 메트릭($\beta_m^{(k)}$)은 수학식 9와 유사하게 수학식 12로 계산될 수 있다.

수학식 12

[0109] $$\beta_m^{(k)} = |(\mathbf{w}^{(k)})^T \mathbf{x}_m^{(k)} + b^{(k)}| / \|\mathbf{w}^{(k)}\|$$

[0110] 한편, 신경망 모델(22)이 DNN인 경우에, 누적 특징($x_m^{(k)}$)의 명료성 메트릭($\beta_m^{(k)}$) 또한 사후 분포에 기반하여 획득될 수 있다.

[0111] DNN으로 구현되는 신경망 모델(22)이 누적 특징($x_m^{(k)}$)에 대해 수행하는 신경망 연산을 함수($\Theta^{(k)}$)라 할 때, 누

적 특징($x_m^{(k)}$)의 특정 클래스(c)에 대한 사후 우도는 $P_{\boldsymbol{\theta}^{(k)}}(c|\mathbf{x}_m^{(k)})$로 표현될 수 있다.

[0112] 신경망 모델(22)이 누적 특징($x_m^{(k)}$)에 대해 예측하는 클래스(c)는 사후 우도($P_{\boldsymbol{\theta}^{(k)}}(c|\mathbf{x}_m^{(k)})$가 최대가 되는 클래스이므로 수학식 13으로 나타난다.

수학식 13

[0113] $$c_{m,1\text{st}} = \arg\max_{c \in \mathcal{C}} P_{\boldsymbol{\theta}^{(k)}}(c|\mathbf{x}_m^{(k)})$$

[0114] 그리고 명료성 메트릭($\beta_m^{(k)}$)은 누적 특징($x_m^{(k)}$)에 대해 수학식 13에 따라 추정되는 사후 우도가 가장 큰 클래스(여기서는 일 예로 제1 클래스($c_{m,1st}$))와 제1 클래스($c_{m,1st}$)를 제외한 다음으로 사후 우도가 가장 큰 클래스(여기서는 일 예로 제2 클래스($c_{m,2nd}$), $c_{m,2\text{nd}} = \arg\max_{c \in \mathcal{C}\setminus\{c_{m,1\text{st}}\}} P_{\boldsymbol{\theta}^{(k)}}(c|\mathbf{x}_m^{(k)})$) 사이의 우도 차를 이용하여 수학식 14로 계산될 수 있다.

수학식 14

[0115] $$\beta_m^{(k)} = P_{\boldsymbol{\theta}^{(k)}}(c_{m,1\text{st}}|\mathbf{x}_m^{(k)}) - P_{\boldsymbol{\theta}^{(k)}}(c_{m,2\text{nd}}|\mathbf{x}_m^{(k)})$$

[0116] 수학식 12 및 수학식 14에 따라 라운드의 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 누적 특징($x_m^{(k)}$) 각각에 대한 명료성 메트릭($\beta_m^{(k)}$)이 계산되면, 수학식 2를 이용하여, 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 누적 특징($x_m^{(k)}$)을 명료 분류 샘플(CCS)과 불명료 분류 샘플(ACS)로 구분하여, 불명료 분류 샘플(ACS)로 구성된 다음 라운드를 위한 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)을 구성한다.

[0117] 불명료 데이터 선택 모듈(26)에서 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)이 구성되어 불명료 분류 샘플(ACS)이 선택되면, 특징 요청 모듈(27)은 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)에 포함된 불명료 분류 샘플(ACS)인 누적 특징($x_m^{(k)}$)의 다음 특징($x_{m,k+1}$)을 IoT 기기(11)로 요청한다.

[0118] 따라서 매 라운드마다 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 불명료하게 분류된 누적 특징($x_m^{(k)}$)인 불명료 분류 샘플(ACS)로 다음 라운드의 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)을 구성하고, 해당 특징($x_{m,k+1}$)을 요청하므로 라운드가 반복됨에 따라 전송을 요청하는 특징의 수가 줄어들게 된다.

[0119] 한편 하나의 특징($x_{m,k}$)이 IoT 기기(11)에서 엣지 서버(12)로 오프로딩되어 신경망 모델(22)이 학습되는 슬롯 기간을 t_0라고 하고, k번째 라운드의 오프로딩 시 채널 이득을 h_k(여기서 $h_k > 0$) 라고 하면, k번째 라운드에서 b 비트의 특징($x_{m,k}$)을 전송하는데 필요한 전송 전력(e_k)은 단항 함수($e_k = \lambda \frac{(b/t)^{\ell}}{h_k}$)로 모델링될 수 있다. 여기서 λ는 에너지 계수(energy coefficient)이고, t는 b 비트의 특징($x_{m,k}$)을 오프로딩하기 위해 허용되는 전송 기간으로, $t \leq t_0$이며, l은 단항식의 차수(monomial order)로서, 변조 및 코딩 방식에 따라 조절되지만 일반적으로는 $2 \leq l \leq 5$의 범위에서 설정된다.

[0120] k번째 라운드에서 오프로딩을 위한 에너지 소비는 전송 전력(e_k)과 전송 기간(t)의 곱으로 수학식 15로 계산될

수 있다.

수학식 15

$$\mathbf{E}(b, t_k; h_k) = e_k t_k = \lambda \frac{b^{\ell}}{h_k \left(t_k\right)^{\ell-1}}$$

[0121]

[0122] 수학식 15를 참조하면, 신경망 모델을 학습시키기 위한 각 라운드에서의 에너지 소비는 특징($x_{m,k}$)의 비트수(b)에 비례하고, 전송 기간(t)에 반비례하다는 것을 알 수 있다. 따라서 에너지 효율적인 엣지 학습을 수행하기 위해서는 전송되는 데이터양을 줄이는 한편 전송 기간(t)을 증가시켜야 한다.

[0123] 그러나 상기에서는 엣지 서버(12)가 IoT 기기(11)로부터 다수의 특징 데이터(x_m)에서 중요도를 기반으로 정렬된 순서에 따라 순차적으로 오프로드되는 특징($x_{m,1}$, …, $x_{m,k}$)을 인가받을 때마다, 현재까지 획득된 누적 특징($x_m^{(k)}$)을 기반으로 신경망 모델을 학습시킨 후, 다음 특징($x_{m,k+1}$)이 오프로드되어야 하는 특징 데이터를 선택하는 것으로 설명하였다.

[0124] 이 경우, k 라운드까지의 특징($x_{m,1}$, …, $x_{m,k}$)을 오프로드한 IoT 기기(11)는 엣지 서버(12)로부터 다음 특징($x_{m,i+1}$)에 대한 오프로드 요청이 인가될 때까지 대기해야 하며, 이 또한 시간적 측면에서 비효율적이다. 뿐만 아니라, 수학식 15에 따르면 k번째 라운드에서 오프로드되는 특징($x_{m,k}$)으로 신경망 모델을 학습시키는데 소요되는 학습 시간(τ_k)을 고려할 때, k 라운드에서의 전송 기간(t_k)은 슬롯 기간(t_0)에서 학습 시간(τ_k)을 차감한 시간($t_k < t_0 - \tau_k$)보다 작아야 하므로, 학습 시간(τ_k)이 증가될수록 전송 기간(t_k)이 짧아져서 에너지 소비가 증가하게 된다. 따라서 시간적 제한이 있는 상태에서 엣지 서버(12)로부터 다음 특징($x_{m,k+1}$)에 대한 오프로드 요청이 인가된 이후, 다음 특징($x_{m,k+1}$)을 엣지 서버(12)로 빠르게 전송하기 위해서는 수학식 15에 나타난 바와 같이, 에너지 소모가 증가하게 된다.

[0125] 이러한 문제를 해소하고자 본 개시에서 다수의 IoT 기기(11)는 엣지 서버(12)로부터 다음 특징($x_{m,k+1}$)에 대한 오프로드 요청이 인가되지 않을 지라도, 다음 특징($x_{m,k+1}$)을 임의로 엣지 서버(12)로 오프로드하는 프리페치를 수행하도록 하여 이전 학습 시간(τ_k)에 다음 특징($x_{m,k+1}$) 중 일부가 중첩되어 오프로드될 수 있도록 함으로써, 이후의 전송 기간(t_{k+1})을 증가시킬 수 있다. 다만 프리페치된 특징($x_{m,k+1}$)은 엣지 서버(12)에 의해 선택되지 않는 경우, 신경망 모델 학습에는 이용되지 않으며, 사용되지 않는 특징($x_{m,k+1}$)을 오프로드하는 것은 비효율적이다. 따라서 엣지 서버(12)는 프리페치 수행 여부에 따른 에너지 이득을 고려하여, 프리페치될 다음 특징($x_{m,k+1}$)의 수를 조절할 수도 있다.

[0126] 이를 위해 엣지 서버(12)의 엣지 학습 장치는 프리페치 설정 모듈(28)을 더 포함할 수 있다.

[0127] 도 6은 프리페치 수행 여부에 따라 특징이 전송되는 기간의 변화를 나타낸다.

[0128] 도 6에서 (a)는 프리페치를 수행하지 않고 특징($x_{m,k}$, $x_{m,k+1}$)을 전송하는 경우를 나타내고, (b)는 프리페치를 수행하여 특징($x_{m,k}$, $x_{m,k+1}$)을 전송하는 경우를 나타낸다. 도 6에서는 엣지 서버(12)에서 IoT 기기(11)로의 피드백 구간을 별도로 도시하였으나, 실질적으로 피드백 구간은 매우 짧은 구간이므로 여기서는 피드백 구간을 무시하고 설명한다.

[0129] 도 6의 (b)에 도시된 바와 같이, 프리페칭 기간이 신경망 모델(22)의 라운드에서의 학습 시간(τ_k)과 중첩될 수 있으므로, 프리페칭 기간은 학습 시간(τ_k)과 동일하게 설정될 수 있다. 현재 라운드에서 불명료 샘플 집합($\mathbb{S}^{(k)}$)에 포함된 누적 특징($x_m^{(k)}$)의 개수가 s_k개이고, 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 s_k개의 누적 특징($x_m^{(k)}$) 중 p_k개가 임의로 선택되어 k+1번째 라운드의 특징($x_{m,k+1}$)으로 프리페치되는 경우를 가정한다. 이 경우, 프리페치되는

데이터의 크기는 특징($x_{m,k+1}$)을 양자화하기 위해 필요한 양자화 비트수(α)에 따라 $\alpha \times p_k$라 할 수 있다. 따라서 p_k개의 특징($x_{m,k+1}$)을 프리페치하기 위해 소모되는 에너지는 수학식 16과 같이 계산될 수 있다.

수학식 16

$$\mathbf{E}(\alpha p_k, \tau_k; h_k) = \lambda \frac{\alpha^\ell}{h_k \tau_k^{\ell-1}} p_k^\ell$$

[0130]

[0131] k + 1번째 전송 기간은 $t_{k+1} = t_0 - \tau_{k+1}$ 이고, 현재 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 다음 라운드의 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)으로 선택되는 가중 특징($x_m^{(k+1)}$) 중 프리페치되지 않은 가중 특징의 개수가 n_{k+1}개이면, n_{k+1}개 특징($x_{m,k+1}$)이 k + 1번째 전송 기간(t_{k+1})동안 오프로드되어야 한다. 따라서 k + 1번째 라운드에서의 에너지 소비는 수학식 17로 계산될 수 있다.

수학식 17

$$\mathbf{E}(\alpha n_{k+1}, t_{k+1}; h_{k+1}) = \lambda \frac{\alpha^\ell}{h_{k+1} t_{k+1}^{\ell-1}} n_{k+1}^\ell$$

[0132]

[0133] 다만 상기한 바와 같이, n_{k+1}개 특징($x_{m,k+1}$)은 신경망 모델(22)이 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 s_k개의 가중 특징($x_m^{(k)}$)으로 학습된 이후에 결정된다. 즉 프리페치를 수행하는 시점에 n_{k+1}개 특징($x_{m,k+1}$)은 결정되지 않은 상태이다.

[0134] n_{k+1}은 현재 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 프리페치된 특징을 제외한 개수로 계산되는 모수($s_k - p_k$)와 현재 불명료 샘플 집합($\mathbb{S}^{(k)}$)과 다음 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)의 원소 개수(s_k, s_{k+1}) 사이의 비율을 나타내는 선택 비율(ρ_k)에 따른 이항 분포를 따른다. 이항 분포의 확률 질량 함수(probability mass function)는 j = 0, …, $s_k - p_k$인 경우에 $P(j) = \binom{s_k - p_k}{j} \rho_k^j (1-\rho_k)^{s_k - p_k - j}$ 이므로, 프리페치 개수(p_k)에 따른 예상 에너지 소비량은 수학식 18로 계산될 수 있다.

수학식 18

$$\mathbb{E}_{n_{k+1}, h_{k+1}}[\mathbf{E}(\alpha n_{k+1}, t_{k+1}; h_{k+1})] = \lambda \frac{\nu \alpha^\ell}{t_{k+1}^{\ell-1}} \mathbb{E}_{n_{k+1}}[n_{k+1}^\ell]$$

[0135]

[0136] 여기서 $\nu = \mathbb{E}[\frac{1}{h_{k+1}}]$는 역채널 이득 기대값을 나타낸다.

[0137] 라운드별 프리페치 및 오프로딩을 위한 전체 에너지 소비량은 수학식 16과 수학식 18의 합으로 나타날 수 있으므로, 프리페치 개수(p_k)에 따른 라운드의 예상 에너지 소비량을 최소화하기 위해서는 수학식 19의 목적함수를 만족해야 한다.

수학식 19

$$\min_{p_k} \ \lambda \frac{\alpha^{\ell}}{h_k \tau_k^{\ell-1}} p_k^{\ell} + \lambda \frac{\nu \alpha^{\ell}}{t_{k+1}^{\ell-1}} \mathbb{E}_{n_{k+1}}[n_{k+1}^{\ell}]$$

$$\text{s.t.} \quad 0 \leq p_k \leq s_k$$

[0138]

[0139] 즉 프리페치로 인한 에너지 소비량과 다음 라운드를 위해 선택되었으나 프리페치되지 않은 특징을 전송하기 위한 에너지 소비량이 최소가 되도록 프리페치 개수(p_k)가 결정되어야 한다.

[0140] 상기한 바와 같이, 라운드로 특징을 선택하여 오프로드되도록 하고, 프리페치를 수행하는 학습 기법을 여기서는 JD2P(Joint Data Deepening-and-Prefetching)라고 하며, JD2P 기법이 에너지 효율성을 갖기 위해서는 수학식 20의 조건을 만족해야 한다.

수학식 20

$$\frac{\tau_k}{t_0} < 1 - (\rho_k(1-\rho_k))^{\frac{\ell}{\ell-1}}$$

[0141]

[0142] 즉 슬롯 기간(t_0)에서 k번째 학습 시간(τ_k)의 비율이 선택 비율(ρ_k)에 따른 수학식 20의 조건을 만족해야 한다. 이에 프리페치 설정 모듈(28)은 수학식 20의 선택 비율(ρ_k)을 만족하면서, 수학식 19를 만족하도록 프리페치 개수(p_k)를 설정하여 IoT 기기(11)가 설정된 프리페치 개수(p_k)에 따라 신경망 모듈의 학습 완료 이전에 프리페치를 수행하도록 할 수 있다.

[0143] 한편, 엣지 서버(12)의 엣지 학습 장치는 IoT 기기(11)가 수집한 데이터 샘플(d_m)을 임베딩하여 중요도에 따라 순차 정렬된 다수의 특징($x_{m,1}$, …, $x_{m,F}$)을 갖는 특징 데이터(x_m)를 획득할 수 있도록 임베딩 모델이 저장된 임베딩 모델 배포 모듈(29)을 더 포함할 수도 있다.

[0144] 도시된 실시예에서, 각 구성들은 위에서 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 기술되지 않은 추가적인 구성을 포함할 수 있다. 또한, 일 실시예에서, 각 구성은 물리적으로 구분된 하나 이상의 장치를 이용하여 구현되거나, 하나 이상의 프로세서 또는 하나 이상의 프로세서 및 소프트웨어의 결합에 의해 구현될 수 있으며, 도시된 예와 달리 구체적 동작에 있어 명확히 구분되지 않을 수 있다.

[0145] 그리고 도 4에 도시된 엣지 학습 장치는 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.

[0146] 뿐만 아니라 엣지 학습 장치는 하드웨어적 요소가 마련된 컴퓨팅 장치 또는 서버에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 장치 또는 서버는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.

[0147] 도 7은 일 실시예에 따른 엣지 학습 방법을 나타낸다.

[0148] 도 7에 도시된 엣지 학습 방법은 엣지 학습 장치가 구비된 엣지 서버(12)에서 수행될 수 있다. 도 7을 참조하면, 엣지 서버(12)는 우선 다수의 IoT 기기(11)로 임베딩 모델을 배포한다(41). 여기서 임베딩 모델은 각각 수집한 데이터 샘플(d_m)을 신경망 연산으로 가상의 특징 공간에 임베딩하여 중요도에 따라 다수의 특징($x_{m,1}$, …,

$x_{m,F}$)이 순차 정렬된 특징 데이터(x_m)를 획득할 수 있도록 하는 모델이다. 이때 임베딩 모듈은 일 예로 주성분 분석(PCA) 모델을 이용할 수 있다. 경우에 따라서 다수의 IoT 기기(11)에는 임베딩 모델이 사전에 저장될 수 있으며, 이 경우, 엣지 서버(12)는 임베딩 모델을 배포하지 않아도 무방하다.

[0149] 그리고 다수의 IoT 기기(11)가 데이터 샘플(d_m)을 임베딩하여 다수의 특징($x_{m,1}$, …, $x_{m,F}$)이 중요도에 따라 순차 정렬된 특징 데이터(x_m)에서 가장 중요한 제1 특징($x_{m,1}$)을 오프로딩하면, 엣지 서버(12)는 오프로딩된 제1 특징($x_{m,1}$)을 획득한다(42). 그리고 현재까지 획득된 특징($x_{m,1}$, …, $x_{m,k}$)을 누적 저장하여 누적 특징($x_m^{(k)}$)을 획득한다(43).

[0150] 엣지 서버(12)는 다수의 데이터 샘플(d_m)에 대한 특징 데이터(x_m)에서 현재 획득된 특징($x_{m,k}$) 이후, 프리페치되어야 하는 다음 중요도의 특징($x_{m,k+1}$)의 프리페치 개수(p_k)를 설정한다(44).

[0151] 본 개시의 엣지 학습 방법에서는 라운드별로 특징 데이터(x_m)의 다수의 특징($x_{m,1}$, …, $x_{m,F}$)이 정렬된 순서에 따라 순차적으로 엣지 서버(12)로 오프로딩되도록 하고, 엣지 서버(12)는 각 라운드에서 오프로딩된 특징($x_{m,1}$, …, $x_{m,k}$)을 이용하여 신경망 모델(22)을 학습시키고, 학습된 신경망 모델(22)이 다수의 데이터 샘플(d_m) 중 명료하게 분류하지 못하는 불명료 데이터 샘플(ACS)의 다음 특징이 선택적으로 오프로드되도록 한다. 즉 현재 라운드의 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 불명료 분류 샘플(ACS)에 해당하는 누적 특징($x_m^{(k)}$)을 탐색하여, 다음 라운드를 위한 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)의 원소로 선택하고, 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)에 포함된 누적 특징($x_m^{(k)}$)의 다음 특징($x_{m,k+1}$)이 엣지 서버(12)로 오프로드되도록 한다.

[0152] 따라서 기본적으로 현재 라운드(k)에서 오프로딩된 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 특징($x_{m,k}$)을 기반으로 신경망 모델(22)이 학습된 이후, 다음 라운드(k+1)에 오프로딩되어야 하는 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)의 특징($x_{m,k+1}$)이 선택된다. 다만 이 경우, 선택된 특징($x_{m,k+1}$)이 오프로딩되어야 하는 시간이 제한되어 에너지 소모가 증가할 수 있다. 이에 엣지 서버(12)는 각 라운드(k)에서 다음 라운드(k+1)를 위한 특징($x_{m,k+1}$)이 선택되기 이전에, 다음 라운드(k+1)를 위한 특징($x_{m,k+1}$)이 임의로 선택되어 프리페치되도록 할 수 있다. 이때, 엣지 서버(12)는 프리페이되어야 하는 특징($x_{m,k+1}$)의 개수인 프리페치 개수(p_k)를 설정할 수 있다. 여기서 프리페치 개수(p_k)는 현재 라운드에서 오프로딩되는 특징($x_{m,k}$)의 개수(s_k)보다 작아야 할 뿐만 아니라, 프리페치로 인한 에너지 소비량과 다음 라운드를 위해 선택되었으나 프리페치되지 않은 특징을 전송하기 위한 에너지 소비량이 최소가 되도록 수학식 19에 따라 결정될 수 있으며, 채널 상태 및 각 라운드(k)에서 신경망 모델을 학습시키는데 소요되는 학습 시간(τ_k)에 따라 조절될 수 있다.

[0153] 프리페치 개수(p_k)가 설정되면, 설정된 프리페치 개수(p_k)에 따라 다음 라운드(k+1)를 위한 특징($x_{m,k+1}$)이 프리페치되도록 IoT 기기(11)로 요청하는 한편, 현재 획득된 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 누적 특징($x_m^{(k)}$)을 이용하여 신경망 모델을 학습시킨다(45). 이때 신경망 모델의 학습은 수집된 데이터 샘플(d_m)에 레이블된 진리값을 이용하여 수행될 수 있으며, 진리값은 IoT 기기(11)가 특징을 오프로드할 때 함께 오프로드될 수 있다.

[0154] 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 누적 특징($x_m^{(k)}$)을 이용하여 신경망 모델이 학습되면, 학습을 종료할지 여부를 판별한다(46). 이때 학습 종료 여부는 신경망 모델이 누적 특징($x_m^{(k)}$)에 대해 신경망 연산한 결과와 진리값 사이의 차이를 비교하여 판별될 수 있다. 만일 학습을 종료할 것으로 판별되면 학습 방법을 종료한다.

[0155] 그러나 학습을 종료하지 않는 것으로 판별되면, 현재 라운드에 획득된 불명료 샘플 집합($\mathbb{S}^{(k)}$)의 s_k개의 누적 특징($x_m^{(k)}$) 중에서 신경망 모델(22)이 명료하게 클래스를 분류한 누적 특징($x_m^{(k)}$)인 명료 분류 샘플(CCS)과 불명료하게 클래스를 분류한 누적 특징($x_m^{(k)}$)인 불명료 분류 샘플(ACS)을 구분하기 위한 명료성 경계 거리($\bar{\beta}^{(k)}$)를 설

정한다(47).

[0156] 명료성 경계 거리($\bar{\beta}^{(k)}$)는 현재까지 학습된 신경망 모델(22)이 특징 공간에서 클래스를 분류하는 기준인 클래스 분류 경계(또는 클래스 문턱값)(cb)로부터의 거리로 나타나게 되며, 신경망 모델(22)에 따라 다양하게 설정될 수 있다. 일 예로 신경망 모델(22)이 이진 SVM으로 구현되는 경우, 명료성 경계 거리($\bar{\beta}^{(k)}$)는 각 클래스의 확률 분포를 GMM으로 근사화하고, 근사화된 확률 분포 사이의 중첩 영역을 이용하여 명료성 경계 거리($\bar{\beta}^{(k)}$)를 수학식 9에 따라 설정될 수 있다. 다른 예로 신경망 모델(22)이 DNN으로 구현되는 경우, 누적 특징($x_m^{(k)}$)에 대한 신경망 모델(22)의 신경망 연산 결과와 진리값 사이의 오류 발생 횟수를 정규화하여 명료성 경계 거리($\bar{\beta}^{(k)}$)를 수학식 11에 따라 결정할 수 있다.

[0157] 명료성 경계 거리($\bar{\beta}^{(k)}$)가 설정되면, 현재 라운드(k)에서 획득된 누적 특징($x_m^{(k)}$) 각각의 명료성 메트릭($\beta_m^{(k)}$)을 계산하고, 계산된 명료성 메트릭($\beta_m^{(k)}$)을 기반으로 현재 라운드(k)의 불명료 샘플 집합($\mathbb{S}^{(k)}$)에서 불명료 데이터 샘플(ACS)을 판별하여 다음 라운드(k)의 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)을 획득한다(48). 여기서 누적 특징($x_m^{(k)}$)에 대한 명료성 메트릭($\beta_m^{(k)}$) 또한 가상의 특징 공간에서 클래스 분류 경계(cb)로부터 누적 특징($x_m^{(k)}$)까지의 거리로서, 신경망 모델(22)이 이진 SVM이거나 DNN인 경우, 각 누적 특징($x_m^{(k)}$)에 대한 명료성 메트릭($\beta_m^{(k)}$)은 수학식 12 및 14에 따라 계산될 수 있다. 그리고 각 누적 특징($x_m^{(k)}$)에 대한 명료성 메트릭($\beta_m^{(k)}$)이 계산되면, 수학식 2에서와 같이, 각 누적 특징($x_m^{(k)}$)에 대한 명료성 메트릭($\beta_m^{(k)}$)이 명료성 경계 거리($\bar{\beta}^{(k)}$) 이하인지 여부에 따라 각 누적 특징($x_m^{(k)}$)을 다음 라운드(k)의 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)에 포함시킨다.

[0158] 불명료 데이터 샘플이 판별되어, 다음 라운드(k)를 위한 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)이 획득되면, 불명료 샘플 집합($\mathbb{S}^{(k+1)}$)에 포함된 누적 특징($x_m^{(k)}$) 가각에 대한 다음 특징($x_{m,k+1}$)을 IoT 기기(11)가 오프로드하도록 요청한다. 이때, 프리페치로 이미 전송되어 저장된 다음 특징($x_{m,k+1}$)은 요청하지 않는다.

[0159] 도 7에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 5에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다양하게 수정 및 변형하여 적용 가능하다.

[0160] 도 8은 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 설명하기 위한 도면이다.

[0161] 도시된 실시예에서, 각 구성 요소들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술된 것 이외에도 추가적인 구성 요소를 포함할 수 있다. 도시된 컴퓨팅 환경(50)은 컴퓨팅 장치(51)를 포함하여, 도 7에 도시된 엣지 학습 방법을 수행할 수 있다. 일 실시예에서, 컴퓨팅 장치(51)는 도 4에 도시된 엣지 학습 장치에 포함된 하나 이상의 구성 요소일 수 있다.

[0162] 컴퓨팅 장치(51)는 적어도 하나의 프로세서(52), 컴퓨터 판독 가능 저장매체(53) 및 통신 버스(55)를 포함한다. 프로세서(52)는 컴퓨팅 장치(51)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(52)는 컴퓨터 판독 가능 저장매체(53)에 저장된 하나 이상의 프로그램들(54)을 실행할 수 있다. 상기 하나 이상의 프로그램들(54)은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(52)에 의해 실행되는 경우 컴퓨팅 장치(51)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0163] 통신 버스(55)는 프로세서(52), 컴퓨터 판독 가능 저장매체(53)를 포함하여 컴퓨팅 장치(51)의 다른 다양한 구성 요소들을 상호 연결한다.

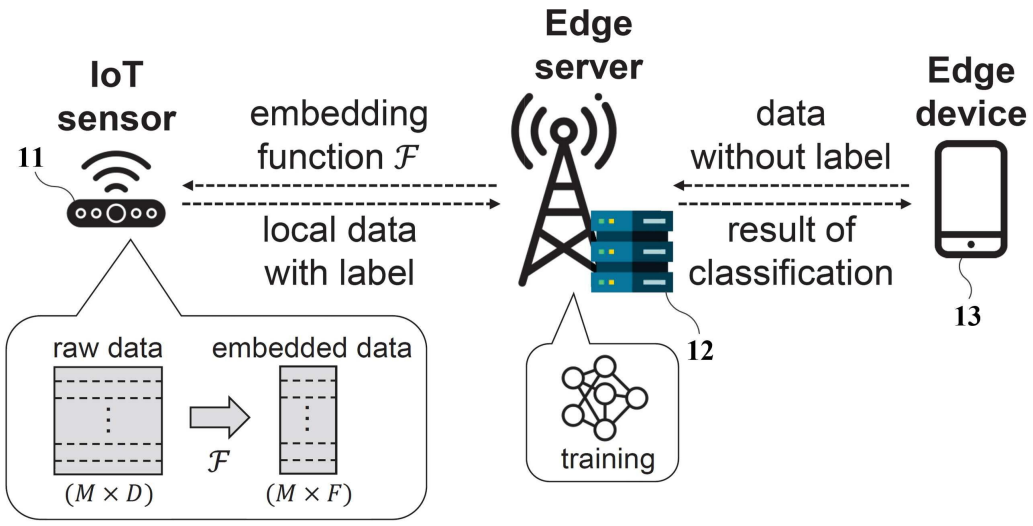
[0164] 컴퓨팅 장치(51)는 또한 하나 이상의 입출력 장치(58)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터

페이스(56) 및 하나 이상의 통신 인터페이스(57)를 포함할 수 있다. 입출력 인터페이스(56) 및 통신 인터페이스(57)는 통신 버스(55)에 연결된다. 입출력 장치(58)는 입출력 인터페이스(56)를 통해 컴퓨팅 장치(51)의 다른 구성 요소들에 연결될 수 있다. 예시적인 입출력 장치(58)는 포인팅 장치(마우스 또는 트랙패드 등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(58)는 컴퓨팅 장치(51)를 구성하는 일 구성 요소로서 컴퓨팅 장치(51)의 내부에 포함될 수도 있고, 컴퓨팅 장치(51)와는 구별되는 별개의 장치로 컴퓨팅 장치(51)와 연결될 수도 있다.

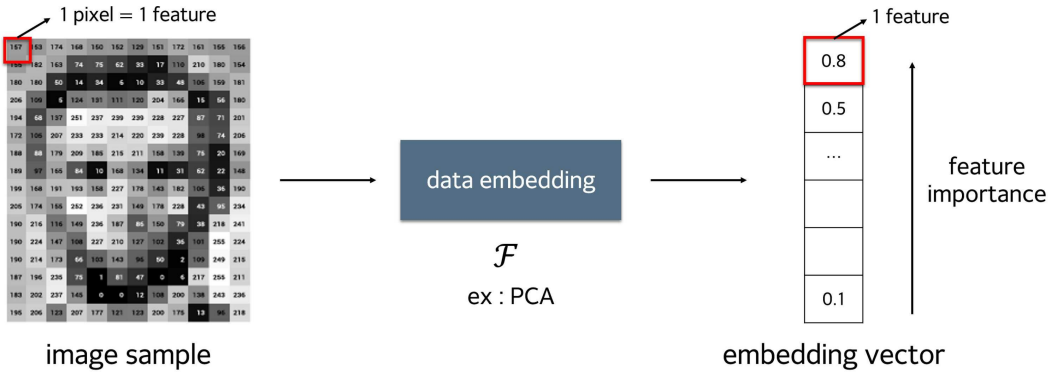
[0165] 이상에서 대표적인 실시예를 통하여 본 발명에 대하여 상세하게 설명하였으나, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 청구범위의 기술적 사상에 의해 정해져야 할 것이다.

도면

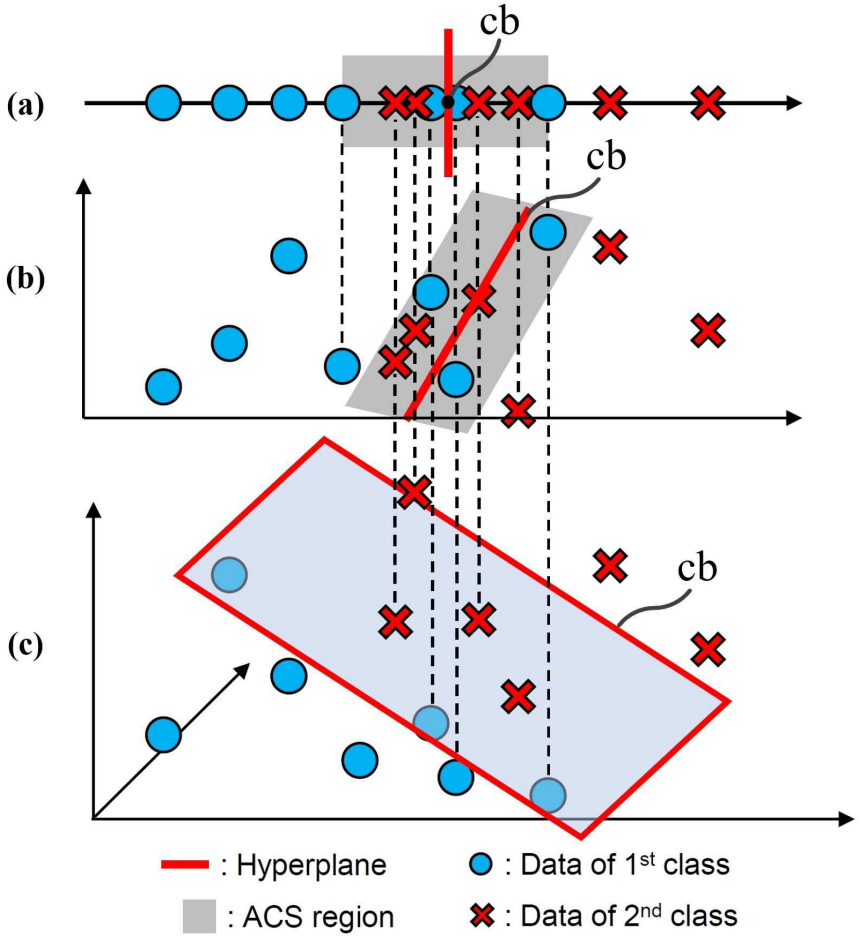
도면1



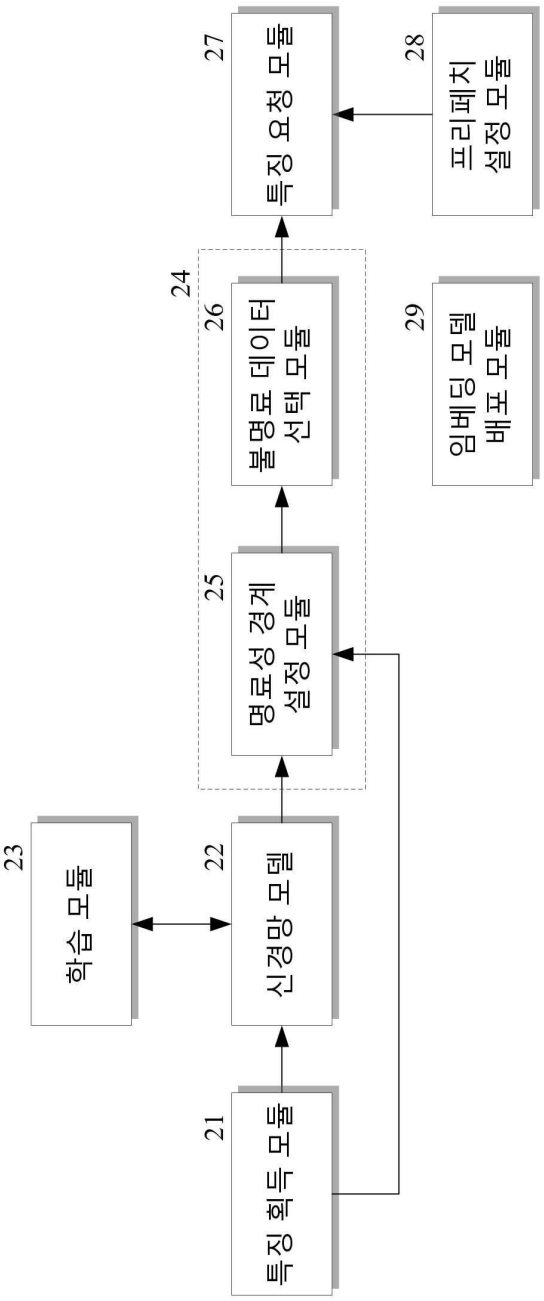
도면2



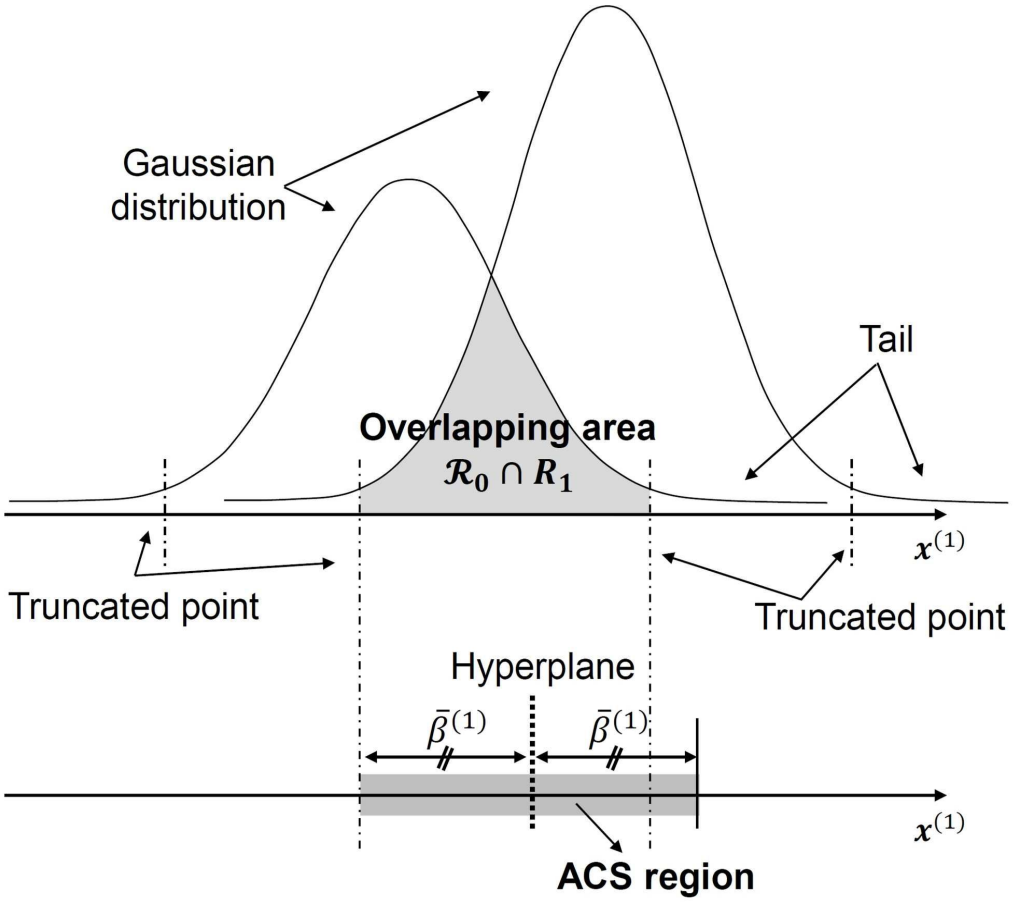
도면3


(a)
cb
(b)
cb
(c)
cb
: Hyperplane
: Data of 1st class
: ACS region
: Data of 2nd class

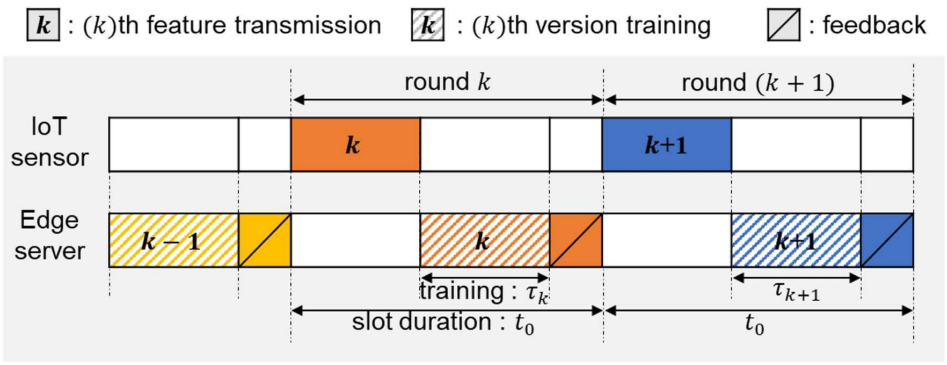
도면4


21
특징 획득 모듈
22
신경망 모델
23
학습 모듈
24
25
명료성 경계 설정 모듈
26
불명료 데이터 선택 모듈
27
특징 요청 모듈
28
프리페치 설정 모듈
29
임베딩 모델 배포 모듈

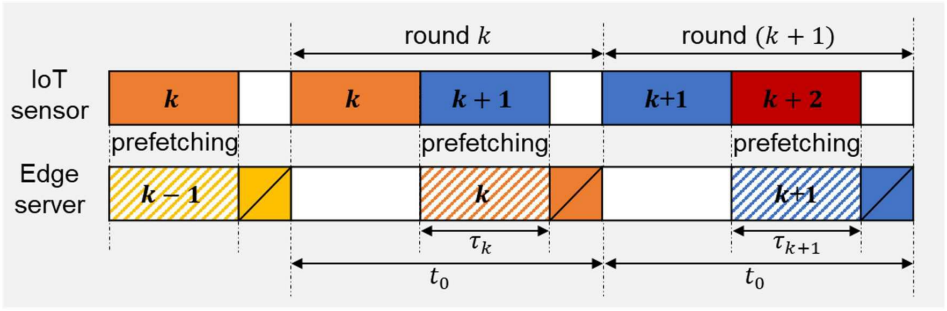
도면5


Gaussian distribution
Tail
Overlapping area
$\mathcal{R}_0 \cap R_1$
$x^{(1)}$
Truncated point
Truncated point
Hyperplane
$\bar{\beta}^{(1)}$
$\bar{\beta}^{(1)}$
$x^{(1)}$
ACS region

도면6

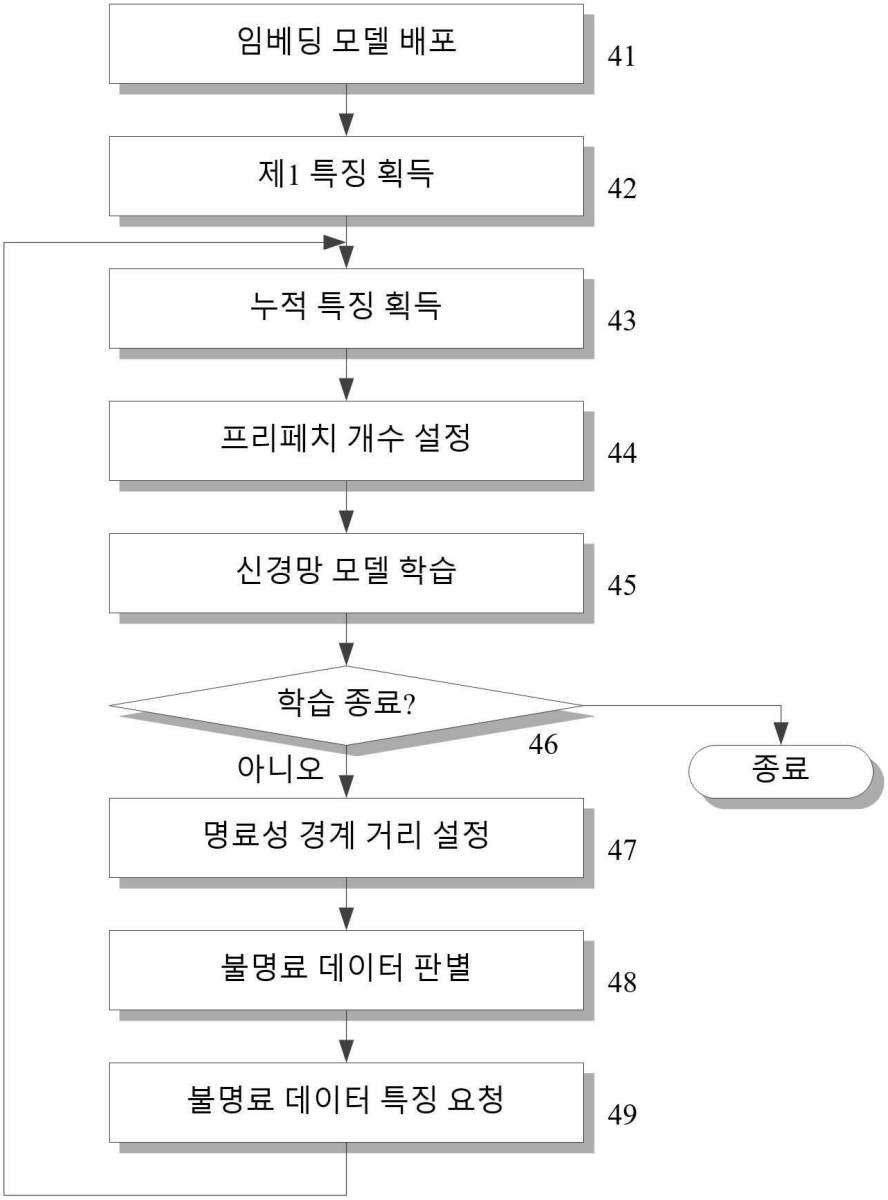


(a) **w/o prefetching**



(b) **w/ prefetching**

도면7


임베딩 모델 배포
41
제1 특징 획득
42
누적 특징 획득
43
프리페치 개수 설정
44
신경망 모델 학습
45
학습 종료?
46
아니오
종료
명료성 경계 거리 설정
47
불명료 데이터 판별
48
불명료 데이터 특징 요청
49

도면8

