



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0115609
(43) 공개일자 2023년08월03일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) G06N 3/04 (2023.01)
(52) CPC특허분류
G06N 3/084 (2023.01)
G06N 3/04 (2023.01)
(21) 출원번호 10-2022-0012271
(22) 출원일자 2022년01월27일
심사청구일자 2022년01월27일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대학교)
(72) 발명자
배승환
인천광역시 미추홀구 인하로 100(용현동)
(74) 대리인
양성보

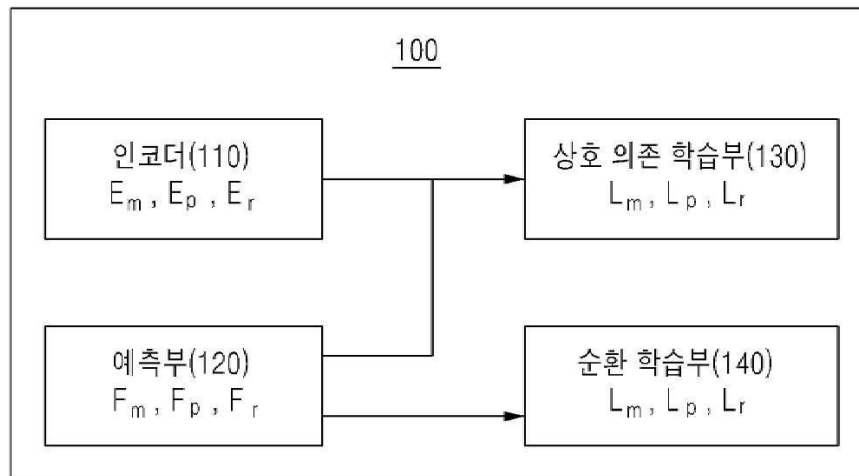
전체 청구항 수 : 총 8 항

(54) 발명의 명칭 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법 및 장치

(57) 요약

모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법 및 장치가 제시된다. 본 발명에서 제안하는 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 장치는 모델, 성능, 자원에 관하여 벡터로 표현된 요소 값을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력하는 인코더, 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 예측부, 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전과 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전과 학습을 수행하는 상호 의존 학습부 및 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 순환 학습부를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호 1711135532
 과제번호 2021R1F1A1049447
 부처명 과학기술정보통신부
 과제관리(전문)기관명 한국연구재단
 연구사업명 기본연구
 연구과제명 [Ezbaro] 고성능 사물 인지 지능을 위한 사물 특징 학습 원천 기술 개발
 기 여 율 10/100
 과제수행기관명 인하대학교 산학협력단
 연구기간 2021.06.01 ~ 2022.05.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1711154594
 과제번호 2022R1C1C1009208
 부처명 과학기술정보통신부
 과제관리(전문)기관명 한국연구재단
 연구사업명 개인기초연구(과기정통부)
 연구과제명 대규모 확장 가능한 On-Vision AI를 위한 E3-AI 네트워크와 협업 학습 연구
 기 여 율 60/100
 과제수행기관명 인하대학교
 연구기간 2022.03.01 ~ 2023.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호 1711160586
 과제번호 2022-0-00448-001
 부처명 과학기술정보통신부
 과제관리(전문)기관명 정보통신기획평가원
 연구사업명 사람중심인공지능핵심원천기술개발
 연구과제명 인간처럼 회상이 가능한 인공 신경망 지속학습 플랫폼 개발
 기 여 율 10/100
 과제수행기관명 인하대학교
 연구기간 2022.04.01 ~ 2022.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1711152759
 과제번호 2020-0-01389-003
 부처명 과학기술정보통신부
 과제관리(전문)기관명 정보통신기획평가원
 연구사업명 인공지능융합선도프로젝트
 연구과제명 인공지능융합연구센터지원
 기 여 율 10/100
 과제수행기관명 인하대학교 산학협력단
 연구기간 2022.01.01 ~ 2022.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호 1711174118
 과제번호 00155915
 부처명 과학기술정보통신부
 과제관리(전문)기관명 정보통신기획평가원
 연구사업명 인공지능융합혁신인재양성(R&D)
 연구과제명 인공지능융합혁신인재양성
 기 여 율 10/100
 과제수행기관명 인하대학교
 연구기간 2022.07.01 ~ 2022.12.31

명세서

청구범위

청구항 1

모델, 성능, 자원에 관하여 벡터로 표현된 요소 값을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력하는 인코더;

모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 예측부;

상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전파 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전파 학습을 수행하는 상호 의존 학습부; 및

상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 순환 학습부를 포함하는 모델/성능/자원 간 상호 최적화 장치.

청구항 2

제1항에 있어서,

상기 상호 의존 학습부는,

상기 예측부에 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력하여 나머지 하나에 관한 예측 임베딩 벡터가 출력되면 해당 예측 임베딩 벡터에 대한 거리 행렬을 이용하여 실제 임베딩 벡터와 예측 임베딩 벡터 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 인코더 및 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복하는

모델/성능/자원 간 상호 최적화 장치.

청구항 3

제1항에 있어서,

상기 순환 학습부는,

상기 예측부를 통한 모델, 성능, 자원 중 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제1 요소 값과 비교하여 제2 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제2 요소에 관한 예측 임베딩 벡터와 입력된 제2 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습하는

모델/성능/자원 간 상호 최적화 장치.

청구항 4

제3항에 있어서,

상기 순환 학습부는,

상기 하나에 관한 예측 임베딩 벡터를 상기 나머지 두 개 중 제2 요소 값과 비교하여 제1 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제1 요소에 관한 예측 임베딩 벡터와 입력된 제1 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복하는

모델/성능/자원 간 상호 최적화 장치.

청구항 5

인코더를 통해 모델, 성능, 자원에 관하여 벡터로 표현된 요소들을 입력 받아 모델, 성능, 자원 각각에 관한 실

제 임베딩 벡터를 출력하는 단계;

예측부를 통해 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 단계;

상호 의존 학습부를 통해 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전파 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전파 학습을 수행하는 단계; 및

순환 학습부를 통해 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 단계

를 포함하는 모델/성능/자원 간 상호 최적화 방법.

청구항 6

제5항에 있어서,

상기 상호 의존 학습부를 통해 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전파 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전파 학습을 수행하는 단계는,

상기 예측부에 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력하여 나머지 하나에 관한 예측 임베딩 벡터가 출력되면 해당 예측 임베딩 벡터에 대한 거리 행렬을 이용하여 실제 임베딩 벡터와 예측 임베딩 벡터 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 인코더 및 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복하는

모델/성능/자원 간 상호 최적화 방법.

청구항 7

제5항에 있어서,

상기 순환 학습부를 통해 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 단계는,

상기 예측부를 통한 모델, 성능, 자원 중 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제1 요소 값과 비교하여 제2 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제2 요소에 관한 예측 임베딩 벡터와 입력된 제2 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습하는

모델/성능/자원 간 상호 최적화 방법.

청구항 8

제7항에 있어서,

상기 하나에 관한 예측 임베딩 벡터를 상기 나머지 두 개 중 제2 요소 값과 비교하여 제1 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제1 요소에 관한 예측 임베딩 벡터와 입력된 제1 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복하는

모델/성능/자원 간 상호 최적화 방법.

발명의 설명

기술 분야

[0001] 본 발명은 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법 및 장치에 관한 것이다.

배경 기술

[0002] 최근 시각/자연어 관련 인공지능(Artificial Intelligence; AI) 연구는 심층 모델(예를 들어, ResNet, BERT)의

발전과 정형화된 오픈 데이터 셋(예를 들어, ImageNet, GLUE) 구축으로 비약적인 성능 향상을 이루고 있다. 하지만, 현재 AI 패러다임은 연구 중심에서 실제 제품/서비스에 적용 할 수 있는 보다 보편적이고 현장 중심적인 AI로 패러다임 전환 중에 있다.

[0003] 보편적인 AI 개발을 위해서는 대상 시스템의 자원(Resource) 환경에 따라 사용자의 요구 성능(performance)을 반영할 수 있는 모델(Model) 최적화가 필수적이다. 최근 모델 최적화를 위해 메타러닝/경량학습(예를 들어, 지식 증류, 전이학습, NAS)이 연구되고 있지만, H/W 자원과 사용자의 정확한 요구 성능(예를 들어, 정확도, 속도)을 고려한 모델 최적화를 필요로 한다. 또한, 최적한 모델 설계까지 반복적인 학습/테스트(예를 들어, NAS의 경우 800GPU/28일) 과정을 통해 상당한 시간과 자원을 소모하며 이는 산업계로 AI 확산을 저해할 수도 있다.

[0004] 모델(M), 성능(P), 자원(R) 간에는 분명한 의존성(dependency)이 있지만, 이러한 의존성을 단순한 모델(예를 들어, 선형, 다항식)로 표현하기에는 불충분하다. 따라서, 모델(M), 성능(P), 자원(R) 간의 의존성을 학습하기 위한 기술을 필요로 한다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국 등록특허 제10-2282741호(102282741)

발명의 내용

해결하려는 과제

[0006] 본 발명이 이루고자 하는 기술적 과제는 모델(M), 성능(P), 자원(R) 간의 의존성을 학습하고 이들 간의 재현성(reproducibility)을 고려한 순환(cycle) 기술을 제안하여, 두 가지 상태(예를 들어, P-R)가 입력으로 주어졌을 때 다른 하나의 결과(예를 들어, M)를 정확히 추론할 수 있는 M-P-R 모델을 제안한다. 또한, 성능 예측($R/M \rightarrow P$), 자원 예측($P/M \rightarrow R$), 모델 최적화($(P/R \rightarrow M)$)의 예측 결과를 가시화/분석화 할 수 있는 프로파일링 기술을 제안하여 모델, 자원, 성능 최적화를 동시에 도모하고자 한다.

과제의 해결 수단

[0007] 일 측면에 있어서, 본 발명에서 제안하는 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 장치는 모델, 성능, 자원에 관하여 벡터로 표현된 요소 값을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력하는 인코더, 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 예측부, 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전파 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전파 학습을 수행하는 상호 의존 학습부 및 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 순환 학습부를 포함한다.

[0008] 상기 상호 의존 학습부는 상기 예측부에 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력하여 나머지 하나에 관한 예측 임베딩 벡터가 출력되면 해당 예측 임베딩 벡터에 대한 거리 행렬을 이용하여 실제 임베딩 벡터와 예측 임베딩 벡터 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 인코더 및 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.

[0009] 상기 순환 학습부는 상기 예측부를 통한 모델, 성능, 자원 중 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제1 요소 값과 비교하여 제2 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제2 요소에 관한 예측 임베딩 벡터와 입력된 제2 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습한다.

[0010] 상기 순환 학습부는 상기 하나에 관한 예측 임베딩 벡터를 상기 나머지 두 개 중 제2 요소 값과 비교하여 제1 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제1 요소에 관한 예측 임베딩 벡터와 입력된 제1 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 상기 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.

[0011] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법은 인코더를 통해 모델, 성능, 자원에 관하여 벡터로 표현된 요소들을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력하는 단계, 예측부를 통해 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 단계, 상호 의존 학습부를 통해 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전파 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전파 학습을 수행하는 단계 및 순환 학습부를 통해 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 단계를 포함한다.

발명의 효과

[0012] 본 발명의 실시예들에 따르면 모델(M), 성능(P), 자원(R) 간의 의존성을 학습하고 이들 간의 재현성(reproducibility)을 고려한 순환(cycle) 기술을 제안하여, 두 가지 상태(예를 들어, P-R)가 입력으로 주어졌을 때 다른 하나의 결과(예를 들어, M)를 정확히 추론할 수 있다. 또한, 성능 예측(R/M→P), 자원 예측(P/M→R), 모델 최적화((P/R)→M)의 예측 결과를 가시화/분석화 할 수 있는 프로파일링 기술을 제안하여 모델, 자원, 성능을 동시에 최적화할 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 따른 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 장치의 구성을 나타내는 도면이다.

도 2는 본 발명의 일 실시예에 따른 모델/성능/자원 인코더를 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 모델/성능/자원 예측부를 설명하기 위한 도면이다.

도 4는 본 발명의 일 실시예에 따른 상호 의존 학습을 설명하기 위한 도면이다.

도 5는 본 발명의 일 실시예에 따른 순환 학습을 설명하기 위한 도면이다.

도 6은 본 발명의 일 실시예에 따른 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0014] 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

[0016] 도 1은 본 발명의 일 실시예에 따른 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 장치의 구성을 나타내는 도면이다.

[0017] 제안하는 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 장치(100)는 인코더(110), 예측부(120), 상호 의존 학습부(130) 및 순환 학습부(140)를 포함한다.

[0018] 본 발명의 실시예에 따른 인코더(110)는 모델, 성능, 자원에 관하여 벡터로 표현된 요소 값을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력한다.

[0019] 먼저, 본 발명의 실시예에 따른 모델(M), 성능(P), 자원(R)을 정의한다. 본 발명에서는 딥 네트워크를 가정한다. 이하 설명에서, 본 발명의 실시예에 따른 임베딩 벡터의 차원이 같다고 가정하지만, 이에 한정되지 않으며 임베딩 벡터의 차원은 같거나 다를 수 있다.

[0020] 모델(M)의 요소 값은 $m \in \mathbb{R}^d$ 와 같이 나타낼 수 있고, 팩터로는 Model type, # layers, # param, head type 등을 포함할 수 있다.

[0021] 성능(P)의 요소 값은 $p \in \mathbb{R}^d$ 와 같이 나타낼 수 있고, 팩터로는 Accuracy, speed, energy 등을 포함할 수 있다.

[0022] 자원(R)의 요소 값은 $r \in \mathbb{R}^d$ 와 같이 나타낼 수 있고, 팩터로는 Processor, Ram memory, GPU memory, GPU

FLOP 등을 포함할 수 있다.

- [0023] 만약, 요소가 문자열(예를 들어 하드웨어 스펙)로 표현 된다면, 이를 word2vec 와 같이 문자열을 벡터로 표기할 수 있는 기법을 사용하여 k-차원의 벡터로 표현할 수 있다. 그렇지 않은 경우, N bit 문자열로 표현할 수 있고, 이때 요소 값의 범위(최대값과 최소값)를 고려한다. MPR 모델의 입력에서 요소들 또는 다른 벡터 간의 결합이 필요할 경우, 벡터 결합(concatenation)을 이용하여 두 개 이상의 벡터를 결합할 수 있다.
- [0024] 도 2는 본 발명의 일 실시예에 따른 모델/성능/자원 인코더를 설명하기 위한 도면이다.
- [0025] 도 2(a)는 모델 인코더, 도 2(b)는 성능 인코더, 도 2(c)는 자원 인코더를 나타내는 도면이다.
- [0026] 도 2(a)를 참조하면, 모델 인코더(E_m)는 모델에 관하여 벡터로 표현된 요소 값(모델의 형태(Model type), 계층 수(# layers), 파라미터 수(# param), 헤드 형태(head type) 등)을 입력 받아 모델에 관한 실제 임베딩 벡터($m \in \mathbb{R}^d$)를 출력한다. 벡터로 표현된 주요 요소들이 결합(concatenation)을 통해 모델 인코더에 입력으로 들어가고, 모델 인코더(E_m)는 d 차원의 모델 임베딩 벡터 m을 출력할 수 있다.
- [0027] 도 2(b)를 참조하면, 성능 인코더(E_p)는 성능에 관하여 벡터로 표현된 요소 값(정확도(Accuracy), 속도(speed), 에너지 사용량(energy) 등)을 입력 받아 성능에 관한 실제 임베딩 벡터($p \in \mathbb{R}^d$)를 출력한다. 벡터로 표현된 주요 요소들이 결합(concatenation)을 통해 성능 인코더에 입력으로 들어가고, 성능 인코더(E_p)는 d 차원의 성능 임베딩 벡터 p를 출력할 수 있다.
- [0028] 도 2(c)를 참조하면, 자원 인코더(E_r)는 자원에 관하여 벡터로 표현된 요소 값(CPU 프로세서(Processor), 램 메모리 용량(Ram memory), GPU 메모리 용량(GPU Memory), GPU 플롭스(GPU FLOP) 등)을 입력 받아 자원에 관한 실제 임베딩 벡터($r \in \mathbb{R}^d$)를 출력한다. 벡터로 표현된 주요 요소들이 결합(concatenation)을 통해 자원 인코더에 입력으로 들어가고, 자원 인코더(E_r)는 d 차원의 자원 임베딩 벡터 r을 출력할 수 있다.
- [0030] 본 발명의 실시예에 따른 예측부(120)는 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력한다.
- [0031] 도 3은 본 발명의 일 실시예에 따른 모델/성능/자원 예측부를 설명하기 위한 도면이다.
- [0032] 도 3(a)는 모델 예측부, 도 3(b)는 성능 예측부, 도 3(c)는 자원 예측부를 나타내는 도면이다.
- [0033] 도 3(a)를 참조하면, 모델 예측부(F_m)는 성능(p), 자원(r)에 관한 결합 벡터를 입력 받아 모델(m)에 관한 예측 임베딩 벡터($\hat{m}$)를 출력한다.
- [0034] 모델 예측부(F_m)의 형태는 다층 퍼셉트론(Multi-Layer Perseptron; MLP) 또는 컨볼루션 뉴럴네트워크 구조일 수 있다. 모델 요소 값(모델의 형태(Model type), 계층 수(# layers), 파라미터 수(# param), 헤드 형태(head type) 등)을 MLP로 예측 가능하다.
- [0035] 도 3(b)를 참조하면, 성능 예측부(F_p)는 모델(m), 자원(r)에 관한 결합 벡터를 입력 받아 성능(p)에 관한 예측 임베딩 벡터($\hat{p}$)를 출력한다.
- [0036] 성능 예측부(F_p)의 형태는 다층 퍼셉트론 또는 컨볼루션 뉴럴네트워크 구조일 수 있다. 성능 요소 값(정확도(Accuracy), 속도(speed), 에너지 사용량(energy) 등)을 MLP로 예측 가능하다.
- [0037] 도 3(c)를 참조하면, 자원 예측부(F_r)는 모델(m), 성능(p)에 관한 결합 벡터를 입력 받아 자원(r)에 관한 예측 임베딩 벡터($\hat{r}$)를 출력한다.
- [0038] 자원 예측부(F_r)의 형태는 다층 퍼셉트론 또는 컨볼루션 뉴럴네트워크 구조일 수 있다. 성능 요소 값(CPU 프로세서(Processor), 램 메모리 용량(Ram memory), GPU 메모리 용량(GPU Memory), GPU 플롭스(GPU FLOP) 등)을

MLP로 예측 가능하다.

- [0040] 본 발명의 실시예에 따른 상호 의존 학습부(130)는 예측부(120)를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전과 학습 및 예측부(120)를 통한 예측 임베딩 벡터와 인코더(110)를 통한 실제 임베딩 벡터를 비교하여 손실률을 감소시키도록 인코더(110) 및 예측부(120)를 학습하는 역전과 학습을 수행한다.
- [0041] 본 발명의 실시예에 따른 상호 의존 학습부(130)는 예측부(120)에 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력하여 나머지 하나에 관한 예측 임베딩 벡터가 출력되면 해당 예측 임베딩 벡터에 대한 거리 행렬을 이용하여 실제 임베딩 벡터와 예측 임베딩 벡터 간의 차이를 측정하여 손실률을 계산한다. 이후, 손실률을 최소화하도록 인코더(110) 및 예측부(120)를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.
- [0042] 도 4는 본 발명의 일 실시예에 따른 상호 의존 학습(Mutual Information; MI)을 설명하기 위한 도면이다.
- [0043] 도 4(a)는 성능 상호 의존 학습부, 도 4(b)는 자원 상호 의존 학습부, 도 4(c)는 모델 상호 의존 학습부를 나타내는 도면이다.
- [0044] 도 4(a)를 참조하면, 먼저, 성능 요소들에 대한 GT(Ground Truth) 생성을 위해 임의의 H/W와 모델을 가지고 해당 모델의 성능을 측정한다.
- [0045] 순전과 학습에서, 실제 임베딩 벡터($\mathbf{p}^*$)를 추출하기 위해 성능 인코더(E_p)에 실제 성능 요소 값들을 입력하고 해당 출력을 성능 실제 임베딩 벡터($\mathbf{p}^*$)로 사용한다. 이후, 모델(m)과 자원(r) 임베딩 벡터를 성능 예측부(F_p)에 입력하고 성능 예측 임베딩 벡터($\hat{\mathbf{p}}$)를 추출한다. 거리 행렬(Distance Metric)을 이용하여 실제 임베딩 벡터($\mathbf{p}^*$)와 예측 임베딩 벡터($\hat{\mathbf{p}}$)의 차이를 측정하고, 손실률을 계산한다.
- [0046] 역전과 학습에서, 상기 계산된 손실률을 최소화하도록 성능 인코더(E_p) 및 성능 예측부(F_p)를 학습한다.
- [0047] 도 4(b)를 참조하면, 먼저, 자원 요소들에 대한 GT(Ground Truth) 생성을 위해 특정 성능을 가진 모델의 자원 소모량을 측정한다.
- [0048] 순전과 학습에서, 실제 임베딩 벡터($\mathbf{r}^*$)를 추출하기 위해 자원 인코더(E_r)에 실제 자원 요소 값들을 입력하고 해당 출력을 자원 실제 임베딩 벡터($\mathbf{r}^*$)로 사용한다. 이후, 모델(m)과 성능(p) 임베딩 벡터를 자원 예측부(F_r)에 입력하고 자원 예측 임베딩 벡터($\hat{\mathbf{r}}$)를 추출한다. 거리 행렬(Distance Metric)을 이용하여 실제 임베딩 벡터($\mathbf{r}^*$)와 예측 임베딩 벡터($\hat{\mathbf{r}}$)의 차이를 측정하고, 손실률을 계산한다.
- [0049] 역전과 학습에서, 상기 계산된 손실률을 최소화하도록 자원 인코더(E_r) 및 자원 예측부(F_r)를 학습한다.
- [0050] 도 4(c)를 참조하면, 먼저, 모델 요소들에 대한 GT(Ground Truth) 생성을 위해 특정 환경(예를 들어, 데이터 셋)에서 특정 성능을 달성할 수 있는 모델 요소들을 도출한다.
- [0051] 순전과 학습에서, 실제 임베딩 벡터($\mathbf{m}^*$)를 추출하기 위해 모델 인코더(E_m)에 실제 모델 요소 값들을 입력하고 해당 출력을 모델 실제 임베딩 벡터($\mathbf{m}^*$)로 사용한다. 이후, 성능(p)과 자원(r) 임베딩 벡터를 모델 예측부(F_m)에 입력하고 모델 예측 임베딩 벡터($\hat{\mathbf{m}}$)를 추출한다. 거리 행렬(Distance Metric)을 이용하여 실제 임베딩 벡터($\mathbf{m}^*$)와 예측 임베딩 벡터($\hat{\mathbf{m}}$)의 차이를 측정하고, 손실률을 계산한다.
- [0052] 역전과 학습에서, 상기 계산된 손실률을 최소화하도록 모델 인코더(E_m) 및 모델 예측부(F_m)를 학습한다.
- [0054] 본 발명의 실시예에 따른 순환 학습부(140)는 예측부(120)를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩

벡터를 비교하여 학습한다.

- [0055] 본 발명의 실시예에 따른 순환 학습부(140)는 예측부(120)를 통한 모델, 성능, 자원 중 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제1 요소 값과 비교하여 제2 요소에 관한 예측 임베딩 벡터를 재생성한다. 재생성된 제2 요소에 관한 예측 임베딩 벡터와 입력된 제2 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 인코더(110) 및 예측부(120)를 학습한다.
- [0056] 이후, 본 발명의 실시예에 따른 순환 학습부(140)는 상기 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제2 요소 값과 비교하여 제1 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제1 요소에 관한 예측 임베딩 벡터와 입력된 제1 요소 값 간의 차이를 측정하여 손실률을 계산한다. 그리고, 상기 손실률을 최소화하도록 인코더(110) 및 예측부(120)를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.
- [0057] 도 5는 본 발명의 일 실시예에 따른 순환 학습(Cycle Learning)을 설명하기 위한 도면이다.
- [0058] 도 5(a)는 성능 순환 학습부, 도 5(b)는 자원 순환 학습부, 도 5(c)는 모델 순환 학습부를 나타내는 도면이다.
- [0059] 도 5(a)를 참조하면, 자원(r)과 모델(m)에 대한 요소 값이 주어 졌을 때, 성능 예측부(F_p)를 통해 성능 예측 임베딩 벡터($\hat{p}$)를 추출한다.
- [0060] 1단계 순환 학습에서, 성능 예측 임베딩 벡터($\hat{p}$)와 모델(m)에 대한 임베딩 벡터를 자원 예측부(F_r)에 입력하여 자원 예측 임베딩 벡터($\hat{r}$)를 재생성한다. 이후, 입력된 자원 벡터(r)와 재생성된 자원 예측 임베딩 벡터($\hat{r}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 성능 예측부(F_p)와 자원 예측부(F_r)를 학습한다.
- [0061] 2단계 순환 학습에서, 성능 예측 임베딩 벡터($\hat{p}$)와 자원(r)에 대한 임베딩 벡터를 모델 예측부(F_m)에 입력하여 모델 예측 임베딩 벡터($\hat{m}$)를 재생성한다. 이후, 입력된 모델 벡터(m)와 재생성된 모델 예측 임베딩 벡터($\hat{m}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 성능 예측부(F_p)와 모델 예측부(F_m)를 학습한다.
- [0062] 도 5(b)를 참조하면, 성능(p)과 모델(m)에 대한 요소 값이 주어 졌을 때, 자원 예측부(F_r)를 통해 자원 예측 임베딩 벡터($\hat{r}$)를 추출한다.
- [0063] 1단계 순환 학습에서, 자원 예측 임베딩 벡터($\hat{r}$)와 모델(m)에 대한 임베딩 벡터를 성능 예측부(F_p)에 입력하여 성능 예측 임베딩 벡터($\hat{p}$)를 재생성한다. 이후, 입력된 성능 벡터(p)와 재생성된 성능 예측 임베딩 벡터($\hat{p}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 자원 예측부(F_r)와 성능 예측부(F_p)를 학습한다.
- [0064] 2단계 순환 학습에서, 자원 예측 임베딩 벡터($\hat{r}$)와 성능(p)에 대한 임베딩 벡터를 모델 예측부(F_m)에 입력하여 모델 예측 임베딩 벡터($\hat{m}$)를 재생성한다. 이후, 입력된 모델 벡터(m)와 재생성된 모델 예측 임베딩 벡터($\hat{m}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 자원 예측부(F_r)와 모델 예측부(F_m)를 학습한다.
- [0065] 도 5(c)를 참조하면, 성능(p)과 자원(r)에 대한 요소 값이 주어 졌을 때, 모델 예측부(F_m)를 통해 모델 예측 임베딩 벡터($\hat{m}$)를 추출한다.
- [0066] 1단계 순환 학습에서, 모델 예측 임베딩 벡터($\hat{m}$)와 자원(r)에 대한 임베딩 벡터를 성능 예측부(F_p)에 입력하여 성능 예측 임베딩 벡터($\hat{p}$)를 재생성한다. 이후, 입력된 성능 벡터(p)와 재생성된 성능 예측 임베딩 벡터($\hat{p}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 모델 예측부(F_m)와 성능 예측부(F_p)를 학습한다.

p)를 학습한다.

- [0067] 2단계 순환 학습에서, 모델 예측 임베딩 벡터($\hat{m}$)와 성능(p)에 대한 임베딩 벡터를 자원 예측부(F_r)에 입력하여 자원 예측 임베딩 벡터($\hat{r}$)를 재생성한다. 이후, 입력된 자원 벡터(r)와 재생성된 모델 예측 임베딩 벡터($\hat{r}$) 간의 거리 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 모델 예측부(F_m)와 자원 예측부(F_r)를 학습한다.
- [0069] 도 6은 본 발명의 일 실시예에 따른 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법을 설명하기 위한 흐름도이다.
- [0070] 제안하는 모델/성능/자원 간 상호 최적화를 위한 의존성 학습 및 순환 학습 방법은 인코더를 통해 모델, 성능, 자원에 관하여 벡터로 표현된 요소들을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력하는 단계(610), 예측부를 통해 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력하는 단계(620), 상호 의존 학습부를 통해 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전과 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전과 학습을 수행하는 단계(630) 및 순환 학습부를 통해 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습하는 단계(640)를 포함한다.
- [0071] 단계(610)에서, 인코더를 통해 모델, 성능, 자원에 관하여 벡터로 표현된 요소들을 입력 받아 모델, 성능, 자원 각각에 관한 실제 임베딩 벡터를 출력한다.
- [0072] 단계(620)에서, 예측부를 통해 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력 받아 나머지 하나에 관한 예측 임베딩 벡터를 출력한다.
- [0073] 단계(630)에서, 상호 의존 학습부를 통해 상기 예측부를 통한 예측 임베딩 벡터에 대해 거리 행렬을 이용하여 손실률을 계산하는 순전과 학습 및 상기 예측부를 통한 예측 임베딩 벡터와 상기 인코더를 통한 실제 임베딩 벡터를 비교하여 상기 손실률을 감소시키도록 상기 인코더 및 상기 예측부를 학습하는 역전과 학습을 수행한다.
- [0074] 본 발명의 실시예에 따른 상호 의존 학습부는 예측부에 모델, 성능, 자원 중 두 개에 관한 결합 벡터를 입력하여 나머지 하나에 관한 예측 임베딩 벡터가 출력되면 해당 예측 임베딩 벡터에 대한 거리 행렬을 이용하여 실제 임베딩 벡터와 예측 임베딩 벡터 간의 차이를 측정하여 손실률을 계산한다. 이후, 손실률을 최소화하도록 인코더 및 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.
- [0075] 단계(640)에서, 순환 학습부를 통해 상기 예측부를 통한 모델, 성능, 자원 모두에 관한 예측 임베딩 벡터를 비교하여 학습한다.
- [0076] 본 발명의 실시예에 따른 순환 학습부는 예측부를 통한 모델, 성능, 자원 중 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제1 요소 값과 비교하여 제2 요소에 관한 예측 임베딩 벡터를 재생성한다. 재생성된 제2 요소에 관한 예측 임베딩 벡터와 입력된 제2 요소 값 간의 차이를 측정하여 손실률을 계산하고, 상기 손실률을 최소화하도록 인코더 및 예측부를 학습한다.
- [0077] 이후, 본 발명의 실시예에 따른 순환 학습부는 상기 하나에 관한 예측 임베딩 벡터를 나머지 두 개 중 제2 요소 값과 비교하여 제1 요소에 관한 예측 임베딩 벡터를 재생성하고, 재생성된 제1 요소에 관한 예측 임베딩 벡터와 입력된 제1 요소 값 간의 차이를 측정하여 손실률을 계산한다. 그리고, 상기 손실률을 최소화하도록 인코더 및 예측부를 학습하며, 상기 과정을 모델, 성능, 자원의 세가지 요소 모두에 관하여 반복한다.
- [0078] 본 발명의 실시예에 따르면, 기존에 활용도가 높은 공개된 시각모델을 경량/중량화 하며 성능과 자원의 증감률을 분석할 수 있고, 관련 학습데이터 구축 및 테스트 모델을 오픈할 수 있다. 또한, 자원과 성능에 민감한 시각모델의 계층, 파라미터, 기법 등을 도출하고 이를 고려한 시각모델을 최적화할 수 있다.
- [0079] 모델/자원/성능 최적화 문제를 보다 고차원의 사물 및 행동인지 영역으로 확장하여 시각모델 최적화 및 경량화를 통해 온-디바이스(on-device)용 시각모델을 개발할 수 있다.
- [0080] 뿐만 아니라, 모델/자원/성능의 상관관계를 모두 고려하여 이들 간의 관계 학습 및 예측함으로써, 모델/자원/성

능을 고려하여 예측 결과와 재현된 결과의 오차를 최적화하는 새로운 순환 학습을 제안하고, 자원과 모델을 입력으로 하여 손쉽게 성능을 예측이 가능하다.

[0081] 또한, 모델 및 성능에 따른 자원 소모량 예측이 가능하고, 제한된 자원에서 특정 성능을 도출할 수 있는 모델의 형태 및 중요 요소를 예측하며, 새로운 모델의 특징을 기술하여 기존 모델과 유사도 및 특정 도메인에서 성능을 예측할 수 있다.

[0082] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPA(field programmable array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0083] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

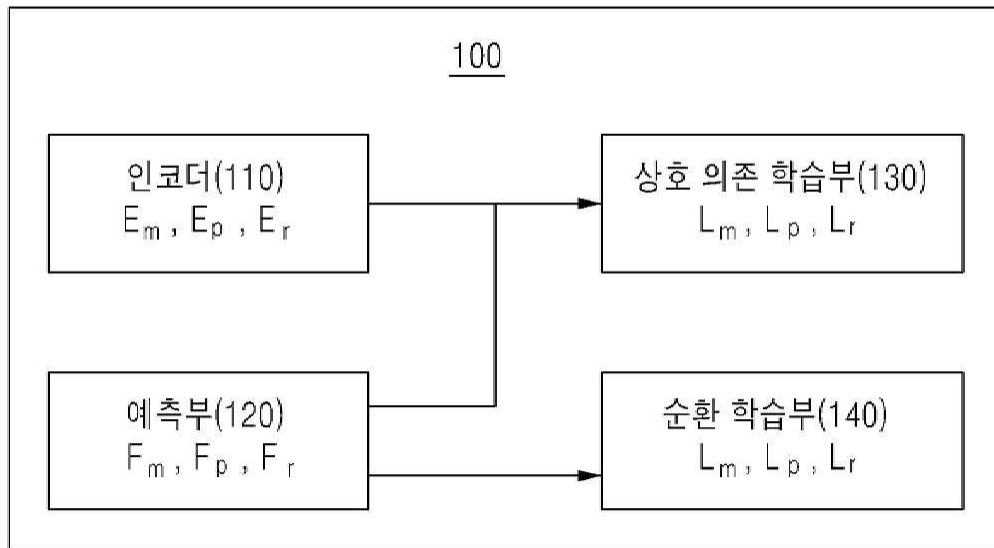
[0084] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0085] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

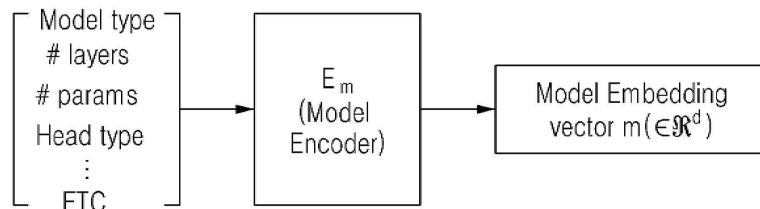
[0086] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

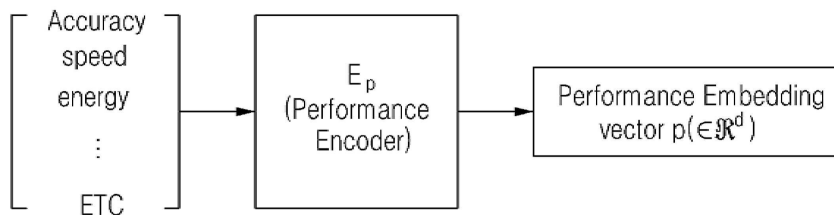
도면1



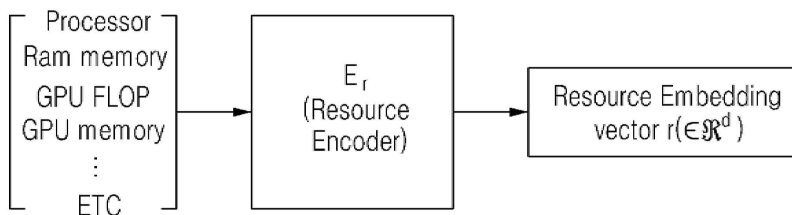
도면2a



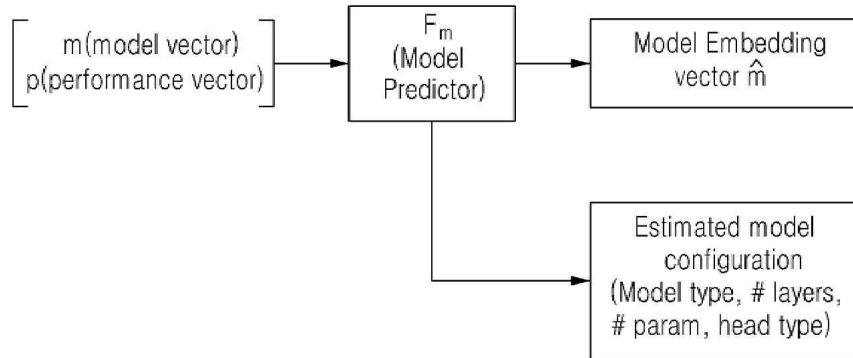
도면2b



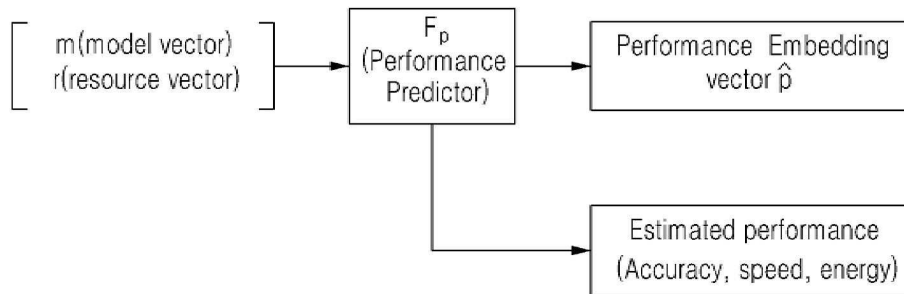
도면2c



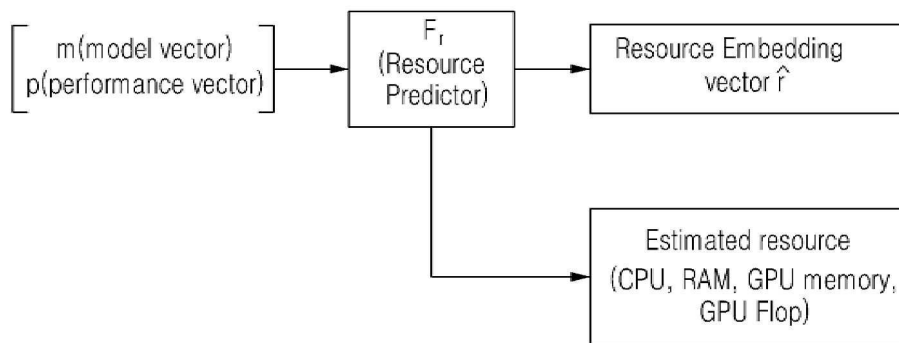
도면3a



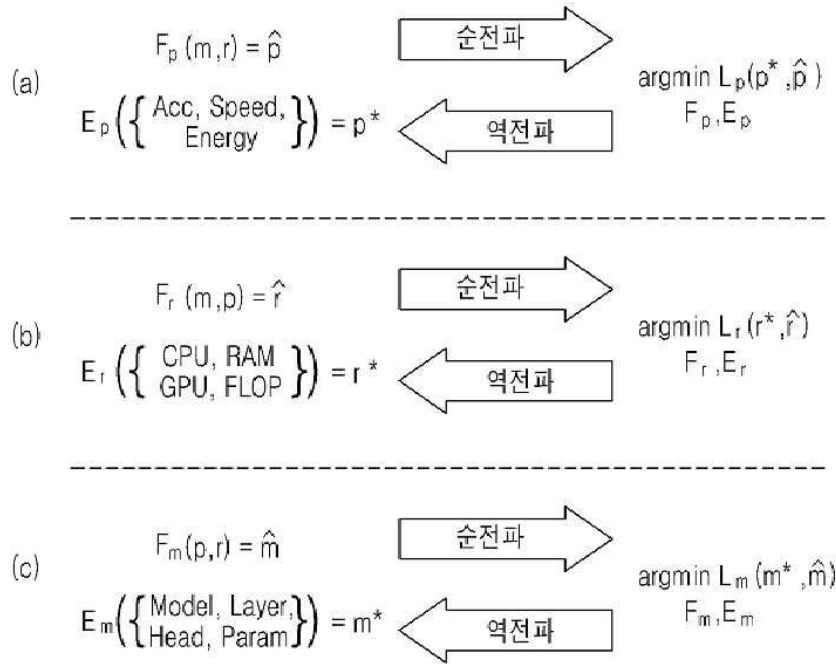
도면3b



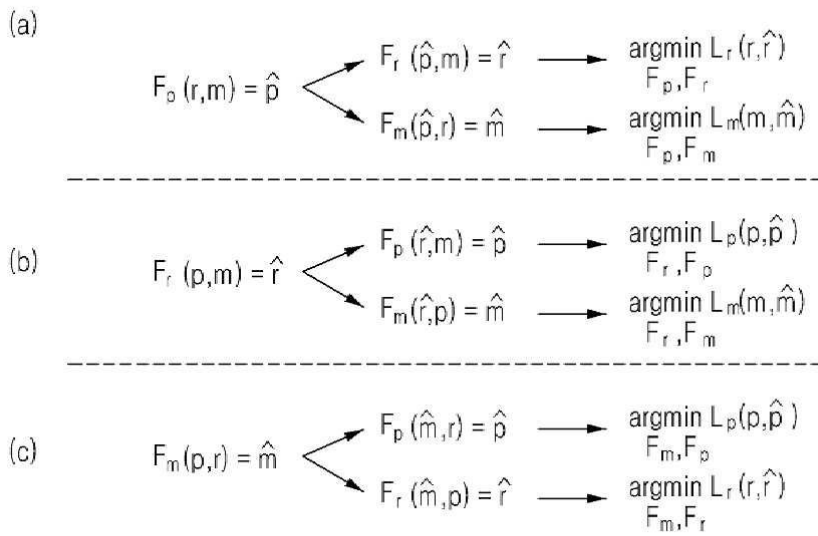
도면3c



도면4



도면5



도면6

