(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0099357
(43) 공개일자 2023년07월04일

(51) 국제특허분류(Int. Cl.)
G06N 3/04 (2023.01) *G06F 11/30* (2006.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G06N 3/045 (2023.01)
G06F 11/3062 (2013.01)
(21) 출원번호 10-2021-0188642
(22) 출원일자 2021년12월27일
심사청구일자 2021년12월27일

(71) 출원인
울산과학기술원
울산광역시 울주군 언양읍 유니스트길 50
(72) 발명자
백웅기
울산광역시 울주군 언양읍 유니스트길 50
한명균
울산광역시 울주군 언양읍 유니스트길 50
(74) 대리인
특허법인(유한)아이시스

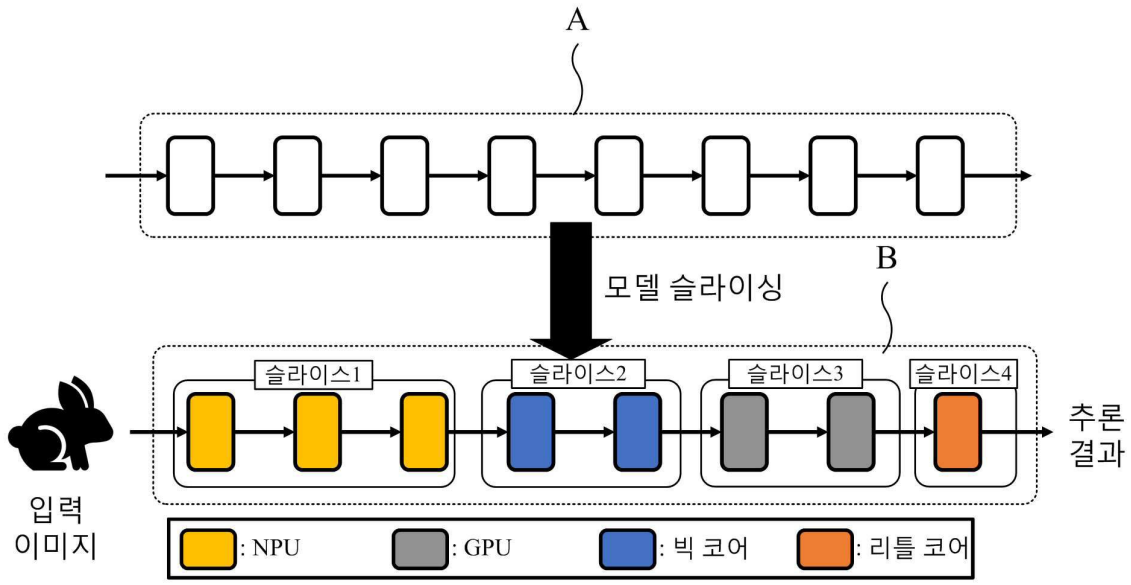
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 **이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법 및 추론 방법**

(57) 요 약

이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법은 모델 분할 장치가 복수의 계층들로 구성되는 딥러닝 모델의 각 계층에 대하여 이기종 컴퓨팅 시스템의 연산장치들 각각에서 해당 계층을 실행할 때의 실행 시간 및 전력을 수집하는 단계, 상기 모델 분할 장치가 상기 실행 시간 및 상기 소비 전력을 이용하여 상기 연산장치들 각각에서 상기 각 계층을 실행할 때의 성능 비용 및 전력 비용을 예측하는 단계, 상기 모델 분할 장치가 상기 연산장치들 각각에서 상기 각 계층에서 다음 계층으로 정보를 전달할 때의 통신 비용을 예측하는 단계 및 상기 모델 분할 장치가 강화학습 모델을 이용하여 주어진 실행 제한 시간 조건에서 상기 성능 비용, 상기 전력 비용 및 상기 통신 비용을 기준으로 상기 연산장치들 각각에서 서로 다른 분할된 계층이 할당되도록 상기 복수의 계층들을 분할하는 단계를 포함한다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/08 (2023.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 1711132319
과제번호 2021R1A2C1011482
부처명 과학기술정보통신부
과제관리(전문)기관명 한국연구재단
연구사업명 개인기초연구(과기정통부)(R&D)
연구과제명 고효율, SLO 보장 워크로드 콘솔리데이션을 위한 통합형 이종 서버 자원 관리 기법 연구 및 개발
기 여 율 1/3
과제수행기관명 울산과학기술원
연구기간 2021.03.01 ~ 2022.02.28
이 발명을 지원한 국가연구개발사업
과제고유번호 1711125843
과제번호 2018-0-00769-004
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 SW컴퓨팅산업원천기술개발(R&D,정보화)
연구과제명 인공지능 시스템을 위한 뉴로모픽 컴퓨팅 SW 플랫폼 기술 개발
기 여 율 1/3
과제수행기관명 한국전자통신연구원
연구기간 2021.01.01 ~ 2021.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711139199
과제번호 2021-0-01817-001
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 정보통신방송혁신인재양성(R&D)
연구과제명 하이퍼-컴포저블 데이터센터를 위한 차세대 컴퓨팅 기술 개발
기 여 율 1/3
과제수행기관명 울산과학기술원
연구기간 2021.07.01 ~ 2021.12.31

명 세 서

청구범위

청구항 1

모델 분할 장치가 복수의 계층들로 구성되는 딥러닝 모델의 각 계층에 대하여 이기종 컴퓨팅 시스템의 연산장치들 각각에서 해당 계층을 실행할 때의 실행 시간 및 전력을 수집하는 단계;

상기 모델 분할 장치가 상기 실행 시간 및 상기 소비 전력을 이용하여 상기 연산장치들 각각에서 상기 각 계층을 실행할 때의 성능 비용 및 전력 비용을 예측하는 단계;

상기 모델 분할 장치가 상기 연산장치들 각각에서 상기 각 계층에서 다음 계층으로 정보를 전달할 때의 통신 비용을 예측하는 단계; 및

상기 모델 분할 장치가 강화학습 모델을 이용하여 주어진 실행 제한 시간 조건에서 상기 성능 비용, 상기 전력 비용 및 상기 통신 비용을 기준으로 상기 연산장치들 각각에서 서로 다른 분할된 계층이 할당되도록 상기 복수의 계층들을 분할하는 단계를 포함하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 2

제1항에 있어서,

상기 모델 분할 장치는 상기 연산장치들 중 어느 하나가 DVFS(dynamic voltage and frequency scaling)를 지원하는 경우, 최소 주파수 및 최대 주파수에서 각각 상기 실행 시간 및 상기 소비 전력을 수집하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 3

제1항에 있어서,

상기 모델 분할 장치는 상기 실행 시간에 대한 선형 모델을 이용하여 상기 연산장치들 각각에서 특정 주파수에서 상기 각 계층을 실행할 때의 상기 성능 비용을 예측하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 4

제1항에 있어서,

상기 모델 분할 장치는 상기 전력을 기준으로 상기 연산장치들 각각에서 특정 주파수에서 상기 각 계층을 실행할 때의 상기 전력 비용을 예측하되,

상기 전력 비용은 상기 연산장치의 고유 특성인 정전 전력의 비용과 상기 연산장치가 상기 특정 주파수에서의 전압에서 상기 각 계층을 실행할 때의 동적 전력의 비용을 포함하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 5

제1항에 있어서,

상기 모델 분할 장치는

상기 각 계층에 대한 입력 텐서의 크기 및 출력 텐서의 크기를 더 수집하고,

상기 각 계층에 대하여 상기 입력 텐서의 크기 및 상기 출력 텐서의 크기의 비례하는 값으로 상기 통신 비용을 예측하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 6

제1항에 있어서,

상기 강화학습모델은 단계 t에서 에이전트가 상태 s_t 및 보상 r_t를 입력받아 산출하는 행동 a_t을 산출하는 과정을 반복하면서 최대 보상값을 갖는 최종 상태 정보를 산출하되,

상기 상태는 상기 각 계층에 대하여 {해당 계층이 상기 분할된 계층의 시작 계층인지 여부, 상기 연산장치들 중 할당되는 어느 하나의 연산장치, 상기 어느 하나의 연산장치의 주파수}로 구성되고,

상기 보상은 제한 시간 조건을 만족하는 경우 전체 실행 비용이 낮을수록 높은 값을 갖고,

상기 행동은 상기 각 계층에 대하여 상태를 변경하는 변경 정보를 포함하는 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 7

제6항에 있어서,

상기 변경 정보는 현재 계층부터 새로운 슬라이스 생성, 현재 계층이 속한 슬라이스와 이전 슬라이스의 병합, 현재 계층을 실행하는 연산장치 변경, 현재 계층을 실행하는 연산장치에서 하나의 레벨만큼 주파수 증가 또는 현재 계층을 실행하는 연산장치에서 하나의 레벨만큼 주파수 감소 중 어느 하나인 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법.

청구항 8

이기종 컴퓨팅 시스템이 딥러닝 모델을 상기 제1항 내지 제7항 중 어느 하나의 항에 따른 방법으로 분할한 정보를 입력받는 단계; 및

상기 이기종 컴퓨팅 시스템이 상기 분할한 정보로 기준으로 슬라이스들과 상기 연산장치들 중 상기 슬라이스들 각각을 실행하는 연산장치를 결정하여 상기 딥러닝 모델을 추론하는 단계를 포함하되,

상기 슬라이스는 상기 딥러닝 모델을 구성하는 상기 복수의 계층들을 분할하여 하나의 연산장치에서 실행하도록 할당한 단위인 이기종 컴퓨팅 시스템에서 분할된 딥러닝 모델을 추론하는 방법.

청구항 9

이기종 컴퓨팅 시스템이 딥러닝 모델을 상기 제1항 내지 제7항 중 어느 하나의 항에 따른 방법으로 분할하는 단계; 및

상기 이기종 컴퓨팅 시스템이 슬라이스들과 상기 연산장치들 중 상기 슬라이스들 각각을 실행하는 연산장치를 결정하여 상기 딥러닝 모델을 추론하는 단계를 포함하되,

상기 슬라이스는 상기 딥러닝 모델을 구성하는 상기 복수의 계층들을 분할하여 하나의 연산장치에서 실행하도록 할당한 단위인 이기종 컴퓨팅 시스템에서 분할된 딥러닝 모델을 추론하는 방법.

발명의 설명

기 술 분 야

[0001] 이하 설명하는 기술은 이기종 컴퓨팅 시스템에서 딥러닝 모델을 분할하는 기법 및 분할한 딥러닝 모델을 추론하는 기법이다.

배 경 기 술

[0002] 실시간 딥러닝 추론은 자율 주행, 인터랙티브 영상 제작, 증강 현실 등 다양한 지능형 서비스에서 요구된다. 클라우드를 이용하지 않는 단일 시스템은 영상과 같이 비교적 용량이 큰 데이터의 실시간 처리가 어려울 수 있다. 이에 실시간 추론 분야에서 이기종 컴퓨팅 시스템(heterogeneous embedded systems)이 주목받고 있다. 이기종 컴퓨팅 시스템은 빅 코어(big core cluster), 리틀 코어(little core cluster), GPU(graphics processing unit) 및 NPU(neural processing unit) 등과 같은 다양한 연산장치들로 구성된다. 이 연산장치들은 성능, 전력

소비, 기능, 메모리 용량 및 통신 부하와 같은 기준에서 서로 상이한 구조 및 시스템 특징을 갖는다.

선행기술문헌

비특허문헌

[0003] (비특허문헌 0001) Young Geun Kim et al., AutoScale: Optimizing Energy Efficiency of End-to-End Edge Inference under Stochastic Variance, arXiv:2005.02544v1 [cs.LG] 6 May 2020

발명의 내용

해결하려는 과제

[0004] 이기종 컴퓨팅 시스템 기반의 실시간 딥러닝 추론은 연산장치들에 대한 추론 작업들의 분배가 필요하다. 그러나, 현재 이기종 컴퓨팅 시스템 기반의 실시간 딥러닝 추론에서는 연산장치들의 특성 및 성능을 고려하여 작업들을 분배하고 실시간성 조건을 만족하면서 효율적으로 실행하는 적절한 방법론이 마련되지 못하고 있다.

[0005] 이하 설명하는 기술은 서로 다른 특성을 갖는 연산장치들을 구비하는 이기종 컴퓨팅 시스템에서 실시간성이라는 제한 조건을 만족하면서 딥러닝 추론 가능한 기법을 제공하고자 한다.

과제의 해결 수단

[0006] 이기종 컴퓨팅 시스템에서 동작하는 딥러닝 모델에 대한 모델 분할 방법은 모델 분할 장치가 복수의 계층들로 구성되는 딥러닝 모델의 각 계층에 대하여 이기종 컴퓨팅 시스템의 연산장치들 각각에서 해당 계층을 실행할 때의 실행 시간 및 전력을 수집하는 단계, 상기 모델 분할 장치가 상기 실행 시간 및 상기 소비 전력을 이용하여 상기 연산장치들 각각에서 상기 각 계층을 실행할 때의 성능 비용 및 전력 비용을 예측하는 단계, 상기 모델 분할 장치가 상기 연산장치들 각각에서 상기 각 계층에서 다음 계층으로 정보를 전달할 때의 통신 비용을 예측하는 단계 및 상기 모델 분할 장치가 강화학습 모델을 이용하여 주어진 실행 제한 시간 조건에서 상기 성능 비용, 상기 전력 비용 및 상기 통신 비용을 기준으로 상기 연산장치들 각각에서 서로 다른 분할된 계층이 할당되도록 상기 복수의 계층들을 분할하는 단계를 포함한다.

발명의 효과

[0007] 이하 설명하는 기술은 이기종 컴퓨팅 시스템에서 실시간성이라는 제한 조건을 만족하면서 딥러닝 추론의 효율성을 극대화한다. 따라서, 이하 설명하는 기술은 자동차, 스마트기기 등 다양한 환경에서 제공되는 지능형 서비스의 성능 향상을 가져온다.

도면의 간단한 설명

[0008] 도 1은 이기종 컴퓨팅 시스템에서 슬라이싱된 딥러닝 모델이 추론하는 과정에 대한 예이다.

도 2는 딥러닝 모델을 슬라이싱하는 시스템에 대한 예이다.

도 3은 모델을 슬라이싱하는 소프트웨어의 아키텍처의 예이다.

도 4는 슬라이싱 정보를 산출하는 강화학습 모델에 대한 예이다.

발명을 실시하기 위한 구체적인 내용

[0009] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0010] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수

있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0011] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설명된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0012] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0013] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0015] 이하 설명하는 기술은 이기종 컴퓨팅 시스템에서 실시간 딥러닝 추론을 제공하기 위한 기법이다. 이하 설명하는 기술은 사전에 마련된 딥러닝 모델을 전제로 이기종 컴퓨팅 시스템의 연산장치들에 해당 딥러닝 모델의 작업을 할당하는 기법이다.

[0016] 딥러닝 모델은 다수의 계층들(layers)로 구성된다. 이하 설명하는 기술은 딥러닝 모델의 계층들을 일정하게 구분하고, 구분한 계층(들)을 각 연산장치들에 할당한다. 이하 딥러닝 모델을 일정한 계층(들)로 구분하는 작업을 슬라이싱(slicing) 또는 모델 슬라이싱이라고 명명한다.

[0017] 이하 설명하는 기술은 이기종 컴퓨팅 시스템에서 각 연산장치들이 딥러닝 추론을 분산 수행할 수 있도록 딥러닝 모델을 슬라이싱하는 기법이다. 물론, 이하 설명하는 기술은 분할된 딥러닝 모델을 이용하여 실시간 딥러닝 추론을 수행하는 과정까지 포함할 수 있다.

[0018] 도 1은 이기종 컴퓨팅 시스템에서 슬라이싱된 딥러닝 모델이 추론하는 과정에 대한 예이다. 도 1에서 A는 학습된 딥러닝 모델이다. 도 1은 특정 이기종 컴퓨팅 시스템이 해당 딥러닝 모델을 이용하여 입력 이미지를 추론하는 예이다. 이기종 컴퓨팅 시스템은 NPU, GPU, 빅 코어 및 리틀 코어를 포함한다고 가정한다. 도 1에서 B는 모델 슬라이싱된 후의 딥러닝 모델을 나타낸다. 도 1의 B를 살펴보면, 딥러닝 계층들이 각각 NPU, GPU, 빅 코어 및 리틀 코어에 분할 할당된 것을 알 수 있다. 딥러닝 계층들 중 슬라이싱되어 하나의 연산장치에서 처리되는 하나의 단위를 슬라이스라고 표시하였다. 도 1에서 NPU가 슬라이스 1을 담당하고, GPU가 슬라이스 2를 담당하고, 빅 코어가 슬라이스 3을 담당하고, 마지막으로 리틀 코어가 슬라이스 4를 담당한다.

[0019] 이기종 컴퓨팅 시스템은 입력 이미지를 딥러닝 모델에 입력하여 이미지를 추론한다. 이기종 컴퓨팅 시스템은 사전에 할당된 정보에 따라 각 연산장치에서 담당하는 슬라이스에 대한 처리를 하도록 하고, 하나의 슬라이스에서 처리된 결과를 다음 슬라이스로 전달하여 처리되도록 한다.

[0020] 이하 설명하는 기술은 강화학습(reinforcement learning, RL)에 기반하여 딥러닝 모델을 슬라이싱한다. 이하 설명하는 기술은 딥러닝 모델의 정보 및 이기종 컴퓨팅 시스템의 연산장치들의 정보를 이용하여 딥러닝 모델을 복수의 연산장치들에 분할 할당하는 것이다.

[0021] 한편, 모델 슬라이싱은 이기종 컴퓨팅 시스템에서 수행될 수 있다. 나아가, 모델 슬라이싱은 이기종 컴퓨팅 시스템이 아닌 다른 컴퓨팅 장치에서 수행될 수도 있다. 이하 모델 분할 장치가 모델 슬라이싱을 한다고 설명한다. 모델 분할 장치는 이기종 컴퓨팅 시스템, 별도의 컴퓨터 장치, 서버 등일 수도 있다.

[0023] 도 2는 딥러닝 모델을 슬라이싱하는 시스템(100)에 대한 예이다. 도 1에서 모델 분할 장치는 서버(120), PC와 같은 컴퓨터 장치(130) 또는 이기종 컴퓨팅 시스템에 해당하는 차량(140)을 예로 도시하였다. 단말(110)은 모델 슬라이싱 대상이 되는 딥러닝 모델에 대한 정보를 모델 분할 장치에 전달할 수 있다. 딥러닝 모델 정보는 모델

의 계층 구조, 추론시 요구되는 제한 조건(메모리 사용량), 각 계층 사이의 통신 오버헤드(텐서의 크기) 등을 포함할 수 있다. 나아가, 단말(110)은 추론에 대한 실시간성 조건에 해당하는 제한 시간 조건을 모델 분할 장치에 더 전달할 수 있다.

[0024] 서버(120)는 딥러닝 모델 정보와 해당 딥러닝 모델이 동작하는 이기종 컴퓨팅 시스템에 대한 정보가 필요하다. 이기종 컴퓨팅 시스템의 정보는 이기종 컴퓨팅 시스템에 포함된 다양한 연산장치들, 각 연산장치의 제한 조건(메모리 사용량, 소비 전력 등) 등을 포함할 수 있다. 서버(120)는 이기종 컴퓨팅 시스템에 대한 정보를 이기종 컴퓨팅 시스템(차량 H)으로부터 전달받을 수 있다. 경우에 따라서, 서버(120)는 이기종 컴퓨팅 시스템에 대한 정보를 단말(110)로부터 전달받을 수도 있다. 서버(120)는 주어진 딥러닝 모델 정보와 이기종 컴퓨팅 시스템 정보를 기준으로 모델을 슬라이싱한다. 이 과정에서 서버(120)는 이기종 컴퓨팅 시스템의 특정 연산장치에서 특정 계층(들)의 동작을 수행할때 필요한 비용 데이터를 수신할 수 있다. 모델 슬라이싱 과정에 대한 자세한 내용은 후술한다. 서버(120)는 최종적으로 해당 딥러닝 모델을 이기종 컴퓨팅 시스템의 각 연산장치에 분할 할당하기 위한 슬라이싱 정보를 이기종 컴퓨팅 시스템에 전달할 수 있다. 이후 이기종 컴퓨팅 시스템(H)는 슬라이싱 정보에 기반하여 분할된 딥러닝 모델을 각 연산장치별로 할당하여 추론을 수행할 수 있다.

[0025] 컴퓨터 장치(130)는 딥러닝 모델 정보와 해당 딥러닝 모델이 동작하는 이기종 컴퓨팅 시스템에 대한 정보가 필요하다. 이기종 컴퓨팅 시스템의 정보는 이기종 컴퓨팅 시스템에 포함된 다양한 연산장치들, 각 연산장치의 제한 조건(메모리 사용량, 소비 전력 등) 등을 포함할 수 있다. 컴퓨터 장치(130)는 이기종 컴퓨팅 시스템에 대한 정보를 이기종 컴퓨팅 시스템(차량 H)으로부터 전달받을 수 있다. 경우에 따라서, 컴퓨터 장치(130)는 이기종 컴퓨팅 시스템에 대한 정보를 단말(110)로부터 전달받을 수도 있다. 컴퓨터 장치(130)는 주어진 딥러닝 모델 정보와 이기종 컴퓨팅 시스템 정보를 기준으로 모델을 슬라이싱한다. 이 과정에서 컴퓨터 장치(130)는 이기종 컴퓨팅 시스템의 특정 연산장치에서 특정 계층(들)의 동작을 수행할때 필요한 비용 데이터를 수신할 수 있다. 도 2는 단말(110)과 컴퓨터 장치(130)를 별개의 객체로 도시하였으나, 단말(110)과 컴퓨터 장치(130)는 물리적으로 동일한 장치일 수도 있다. 모델 슬라이싱 과정에 대한 자세한 내용은 후술한다. 컴퓨터 장치(130)는 최종적으로 해당 딥러닝 모델을 이기종 컴퓨팅 시스템의 각 연산장치에 분할 할당하기 위한 슬라이싱 정보를 이기종 컴퓨팅 시스템에 전달할 수 있다. 이후 이기종 컴퓨팅 시스템(H)는 슬라이싱 정보에 기반하여 분할된 딥러닝 모델을 각 연산장치별로 할당하여 추론을 수행할 수 있다.

[0026] 차량(140)은 딥러닝 모델 정보와 해당 딥러닝 모델이 동작하는 이기종 컴퓨팅 시스템에 대한 정보가 필요하다. 이기종 컴퓨팅 시스템의 정보는 이기종 컴퓨팅 시스템에 포함된 다양한 연산장치들, 각 연산장치의 제한 조건(메모리 사용량, 소비 전력 등) 등을 포함할 수 있다. 차량(140)은 주어진 딥러닝 모델 정보와 이기종 컴퓨팅 시스템 정보를 기준으로 모델을 슬라이싱한다. 이 과정에서 차량(140)은 이기종 컴퓨팅 시스템의 특정 연산장치에서 특정 계층(들)의 동작을 수행할때 필요한 비용 데이터를 확인할 수 있다. 모델 슬라이싱 과정에 대한 자세한 내용은 후술한다. 차량(140)은 최종적으로 해당 딥러닝 모델을 이기종 컴퓨팅 시스템의 각 연산장치에 분할 할당하기 위한 슬라이싱 정보를 생성할 수 있다. 이후 차량(140)은 슬라이싱 정보에 기반하여 분할된 딥러닝 모델을 각 연산장치별로 할당하여 추론을 수행할 수 있다.

[0028] 모델 분할 장치는 모델 슬라이싱을 수행한다. 모델 분할 장치는 모델 슬라이싱을 수행하는 소프트웨어 기반의 시스템에 해당한다. 즉, 모델 분할 장치는 입력되는 딥러닝 모델을 일정하게 처리하는 프로그램 및 학습모델을 이용하여 모델 슬라이싱을 수행한다. 모델 분할 장치는 특정 이기종 컴퓨팅 시스템을 기준으로 특정한 딥러닝 모델을 슬라이싱한다. 이하 분할 대상이 되는 딥러닝 모델을 타깃 딥러닝 모델이라고 명명한다. 또한, 타깃 딥러닝 모델이 동작하는 대상인 이기종 컴퓨팅 시스템을 타깃 이기종 컴퓨팅 시스템이라고 명명한다.

[0029] 도 3은 모델을 슬라이싱하는 소프트웨어(200)의 아키텍처의 예이다. 모델 슬라이싱 소프트웨어는 프로파일러(210), 추정기(220), 슬라이싱 모델(230) 및 런타임 시스템(240)을 포함한다.

[0030] 프로파일러(210)는 타깃 딥러닝 모델의 정보(구조 정보)를 입력받는다. 프로파일러(210)는 타깃 이기종 컴퓨팅 시스템에서의 연산장치에서 타깃 딥러닝 모델의 각 계층의 동작을 실행하는 경우 소요되는 비용 데이터를 수집한다. 프로파일러(210)는 타깃 이기종 컴퓨팅 시스템에서 추론 과정 중 타깃 딥러닝 모델의 각 계층에서 필요한 비용 데이터를 수집한다. 비용 데이터는 실행 시간, 소비 전력(소비 에너지) 등을 포함한다. 따라서, 프로파일러(210)는 타깃 딥러닝 모델의 특정 계층이 타깃 이기종 컴퓨팅 시스템의 특정 연산장치에서 동작하는 경우 소요되는 비용 데이터를 수집한다. 결국, 프로파일러(210)는 타깃 딥러닝 모델의 모든 계층들과 타깃 이기종 컴퓨팅 시스템의 연산장치들의 가능한 모든 조합에 대한 비용 데이터를 수집할 수 있다.

[0031] 연산장치들 중 적어도 하나의 연산장치가 DVFS(dynamic voltage and frequency scaling)를 지원하는 경우, 프로파일러(210)는 각 계층에 대하여 가능한 최소 주파수 및 최대 주파수에서의 비용 데이터를 수집한다.

[0032] 메모리나 기능적 제한으로 어느 계층에 대하여 특정 연산장치에서의 비용 데이터를 수집할 수 없는 경우, 프로파일러(210)는 해당 연산장치에서 해당 계층에 대한 비용을 무한대로 설정한다.

[0033] 프로파일러(210)는 각 계층의 입력 텐서(tensor) 크기 및 출력 텐서 크기도 수집한다. 텐서 크기는 해당 계층에서의 통신 비용을 추정하는데 활용된다.

[0034] 프로파일러(210)는 타깃 이기종 컴퓨팅 시스템의 연산장치들 각각이 타깃 딥러닝 모델의 각 계층의 추론 과정을 수행하는데 필요한 비용 데이터 및 텐서 크기를 수집한다. 프로파일러(210)가 수집하는 전체 데이터를 포괄적으로 프로파일 데이터라고 명명한다. 프로파일 데이터는 각 계층별 정보로 구성될 수 있다.

[0035] 추정기(220)는 프로파일 데이터를 입력받는다. 추정기(220)는 실행 비용 추정기(221) 및 통신 비용 추정기(222)를 포함한다. 각 계층에서의 비용은 실행 비용과 통신 비용을 포함한다.

[0037] 통신 비용은 슬라이싱된 계층(슬라이스)에서 다음 슬라이스로 일정한 정보를 전달하는 비용에 해당한다. 연구자는 다양한 크기의 텐서들에 대한 마이크로벤치마크를 수행하였고, 그 결과 통신 비용은 텐서의 크기와 일정한 선형 관계를 갖는다는 것을 확인하였다. 이와 같은 결과에 따라 연구자는 특정 계층 *l*의 입력 텐서 크기 및 출력 텐서 크기를 이용하여 통신 비용을 추정하였다. 통신 비용 추정기(222)는 아래 수학식 1 및 수학식 2와 같은 상관관계로 계층 *l*의 통신 비용을 추정할 수 있다. 수학식 1은 계층 *l*의 입력단에서의 비용이고, 수학식 2는 계층 *l*의 출력단에서의 비용이다.

수학식 1

[0039] $$C_{i,l,d,f_d} = \beta_{d,f_d} \cdot \tau_{i,l} + \delta_{d,f_d}$$

수학식 2

[0040] $$C_{o,l,d,f_d} = \beta_{d,f_d} \cdot \tau_{o,l} + \delta_{d,f_d}$$

[0041] $\tau_{i,l}$,는 계층 *l*에서 입력 텐서의 전체 크기, $\tau_{o,l}$는 계층 *l*에서 출력 텐서의 전체 크기, $\beta_{d,fd}$는 주파수 f_d에서 연산 장치 d에 대한 회귀 계수이고, $\delta_{d,fd}$는 주파수 f_d에서 연산 장치 d에 대한 일정한 상수값(조절값)에 해당한다.

[0043] 실행 비용 추정기(221)는 통신 비용을 배제하고, 각 연산장치에서 각 계층의 추론 동작을 수행하는데 소요되는 실행 비용을 추정한다. 실행 비용 추정기(221)는 먼저 프로파일 데이터를 이용하여 연산장치 d가 주파수 f_d에서 동작하는 전체 실행 비용을 추정한다. 실행 비용 추정기(221)는 연산장치 d가 최대 주파수 및 최소 주파수에서 동작하는 경우 소요되는 실행 비용을 추정한다. 이후, 실행 비용 추정기(221)는 각 계층에 대하여 전체 실행 비용에서 통신 비용 추정기(222)가 추정한 통신 비용을 감산하여 실행 비용을 추정한다.

[0044] 실행 비용 추정기(221)는 성능 및 전력을 각각 예측한다.

[0045] 실행 비용 추정기(221)는 아래 수학식 3과 같이 선형 모델에 기반하여 성능을 예측할 수 있다. 성능은 특정 연산장치에서 특정 계층의 추론을 수행하는 과정에 소요되는 시간으로 결정될 수 있다.

수학식 3

[0047] $$T_{l,d,f_d} = \theta_{l,d} \cdot \frac{1}{f_d} + \rho_{l,d}$$

[0048] $T_{l,d,fd}$는 주파수 f_d에서 연산장치 d의 계층 *l*의 추론에 추정된 지연(latency)이고, $\theta_{l,d}$ 및 $\rho_{l,d}$는 계수들이다. $\theta_{l,d}$ 및 $\rho_{l,d}$는 프로파일러(210)가 최대 주파수 및 최소 주파수에서 수집한 데이터를 이용하여 두 점을 연결하는 선형식의 기울기와 절편을 계산하여 결정될 수 있다.

[0050] 실행 비용 추정기(221)는 아래 수학식 4를 이용하여 주파수 f_d에서 연산장치 d의 계층 *l*의 추론에 소요되는 전력을 예측할 수 있다. 전체 전력 소비는 동적 전력 소비 $P_{dynamic,l,d,fd}$ 및 정적 전력 소비 $P_{static,l,d,fd}$로 구분될 수 있다.

수학식 4

[0052] $$P_{l,d,f_d} = P_{\text{dynamic},l,d,f_d} + P_{\text{static},d,f_d}$$

[0053] 정적 전력 소비는 연산장치의 고유 특성에 따라 결정된다. 정적 전력 소비는 딥러닝 모델의 특성과 관계없이 결정되며, 따라서 특정한 계층을 실행하지 않고 주어진 정보(offline 프로파일)만으로 결정될 수 있다.

[0054] 이에 반하여, 동적 전력 소비는 연산장치의 특성 및 계층의 특성에 따라 결정될 수 있다. 실행 비용 추정기(221)는 아래 수학식 5의 수식을 이용하여 주파수 f_d에서 연산장치 d의 계층 *l*의 추론에 소요되는 동적 전력 소비를 추정할 수 있다.

수학식 5

[0056] $$P_{\text{dynamic},l,d,f_d} = \frac{V_{f_d}^2 \cdot f_d}{V_{f_{d,\max}}^2 \cdot f_{d,\max}} \cdot P_{\text{dynamic},l,d,f_{d,\max}}$$

[0057] $f_{d,max}$는 연산장치 d의 최대 주파수이고, $V_{fd,max}$는 $f_{d,max}$에서의 전압 레벨이다. 전압 레벨은 하드웨어 스펙 또는 측정을 통해 얻을 수 있다. $P_{dynamic,l,d,fd}$는 결국 최대 주파수에서의 동적 전력 소비값이다.

[0059] 슬라이싱 모델(230)은 강화학습 모델로서 주어진 실시간 조건을 만족하면서 에너지 효율 등 실행 효율을 최대화하도록 딥러닝 모델을 슬라이싱한다. 슬라이싱 모델(230)은 전술한 슬라이싱 정보를 산출한다. 슬라이싱 정보는 딥러닝 모델을 분할하여 구성된 슬라이스 개수, 각 슬라이스에 속하는 계층(들), 각 슬라이스를 실행하는 연산장치에 대한 정보를 포함한다. 슬라이싱 정보는 각 연산장치의 실행 주파수 정보를 더 포함할 수도 있다.

[0060] 슬라이싱 모델(230)은 강화학습 모델 중 다양한 형태로 구현될 수 있다. 연구자는 모델의 메모리 사용량, 학습에 필요한 시간 등을 고려하여 DQN(Deep Q Network) 기반 모델을 이용하였다. 도 4는 슬라이싱 정보를 산출하는 강화학습 모델에 대한 예이다. 강화학습 모델은 크게 에이전트(agent)와 환경(environment)으로 구성된다.

[0061] 환경은 추론 워크로드(workload), 제한 시간 조건, 이기종 컴퓨팅 시스템의 정보로 구성된다. 추론 워크로드는 딥러닝 모델의 구조 및 성능에 대한 정보로서, 계층 정보, 통신 오버헤드, 제한 조건(메모리 요구량 등), 소비 에너지 등으로 구성될 수 있다.

[0062] 에이전트는 슬라이싱 모델(230)에 해당하며 시간(단계) t에서 상태 s_t 및 보상 r_t을 입력받아 DQN 구조로 일정한 행동 a_t을 결정한다.

[0063] 상태 s_t는 아래 수학식 6과 같이 표현될 수 있다. 상태는 계층의 개수 Λ와 동일한 길이를 갖는 벡터로 표현된다.

수학식 6

[0065] $$s_t = (\vec{e}_1, \vec{e}_2, \cdots, \vec{e}_\Lambda)$$

[0066] 상태 벡터의 각 요소도 역시 벡터이며, 해당 요소는 아래 수학식 7과 같이 표현된다. 수학식 7에서 *l*은 계층의 인덱스이다. $\sigma \in \{0,1\}$는 해당 계층이 특정 슬라이스의 첫 번째 계층($\sigma=1$) 또는 아닌지($\sigma=0$)를 나타낸다. $d \in D$는 해당 계층 동작을 실행하는 특정 연산장치를 나타낸다. 이기종 컴퓨팅 시스템에서, 연산장치 세트 D는 {빅 코너, 리틀 코어, GPU, NPU}일 수 있다. 물론 이는 하나의 예이며, D는 다른 다양한 연산장치들을 포함할 수도 있다. $f_d \in F_d$는 연산장치 d의 주파수를 나타내며, f_d는 d의 가능한 주파수들을 나타낸다.

수학식 7

[0068] $$\vec{e}_l = (\sigma, d, f_d)$$

[0070] 각 단계 t에서 에이전트는 환경(타깃 이기종 컴퓨팅 시스템에서 추론 작업량)에 일정한 행동을 전달하며 환경과 상호작용한다. 에이전트는 다중 계층들이 아닌, 단일 계층을 선택하여 행동을 수행하도록 설계된다. 이는 상태 공간이 점진적으로 탐색되도록 한다. 아래 수학식 8은 단계 t에서 에이전트가 수행하는 행동을 나타낸다.

수학식 8

[0072] $$a_t = (l, k)$$

[0073] *l*은 해당 행동을 수행할 계층 인덱스이고, $k \in K$는 해당 행동과 관련된 변경 내용을 의미한다. 각 계층에 대하여 에이전트는 아래 수학식 9에 정의된 변경들 중 어느 하나를 적용할 수 있다. 정의된 변경들은 해당 계층부터 새로운 슬라이스 생성(k_{split}), 해당 계층이 속한 슬라이스와 이전 슬라이스의 병합(k_{merge}), 해당 계층을 실행하는 연산장치 변경(연산장치 리스트에서 다음 장치 $k_{d,next}$ 또는 이전 장치$k_{d,prev}$), 연산장치의 하나의 레벨만큼 주파수 증가($k_{fd,up}$) 및 연산장치의 하나의 레벨만큼 주파수 감소($k_{fd,down}$)를 포함한다. 따라서, 에이전트는 현재 계층에서 대하여 상기 변경들 중 어느 하나의 변경을 나타내는 행동을 출력할 수 있다.

수학식 9

[0075] $$K = \{k_{\text{split}}, k_{\text{merge}}, k_{d,\text{next}}, k_{d,\text{prev}}, k_{f_d,\text{up}}, k_{f_d,\text{down}}\}$$

[0077] 슬라이싱 모델(230)은 DQN을 사용하여 모델 슬라이싱과 추론 수행 계획을 수립(슬라이싱 정보 생성)한다. DQN은 전연결 계층 및 ReLU 함수들로 구성될 수 있다. 아래 표 1의 알고리즘은 DQN을 사용하여 모델 슬라이싱과 추론 수행 계획 수립에 대한 예이다.

표 1

Algorithm 1 The findEfficientSlicingAndExecPlan function

```
1: procedure FINDEFFICIENTSLICINGANDEXECPLAN(Λ)
2:     Q ← initializeQNetwork(Λ, depth, width)
3:     M_r ← initializeReplayMemory()
4:     s_1 ← s_init
5:     s_best ← s_init
6:     r_best ← −Φ
7:     for t ← 1 to t_max do
8:         A_t ← getValidActions(s_t)
9:         if generateRandomNumber() < ϵ(t) then
10:            a_t ← getRandomAction(A_t)
11:        else
12:            a_t ← argmax_{a∈A_t} Q(s_t, a)
13:        end if
14:        r_t ← calculateImmediateReward(s_t, a_t)
15:        s_{t+1} ← applyAction(s_t, a_t)
16:        if r_t > r_best then
17:            r_best ← r_t
18:            s_best ← s_{t+1}
19:        end if
20:        M_r ← M_r ∪ {(s_t, a_t, r_t, s_{t+1})}
21:        B ← sampleMiniBatch(M_r)
22:        trainAndUpdateQNetwork(Q, B, t)
23:    end for
24:    return s_best
25: end procedure
```

[0079]

[0080] 알고리즘 1을 살펴보면, 주요 변수들을 초기화하고(라인 2~라인 6), 슬라이싱 모델(230)은 DQN 모델을 학습하기 위하여 라인 7 ~ 23 과정을 반복한다. 물리적으로 학습 과정은 전술한 모델 분할 장치가 수행한다. 학습 과정을 t 단계를 기준으로 설명한다. 에이전트는 학습 정도에 따라 행동을 선택할 수 있다. 지역적 수렴을 막기 위하여 에이전트는 임의로 확률 ε(t)의 가능한 행동을 선택할 수 있다(라인 10). 에이전트는 Q 네트워크 동작으로 산출되는 1- ε(t) 확률 기준으로 해당 시점 최적 행동(a_t)을 선택할 수 있다. t가 증가하면, 학습된 Q 네트워크에 따라 행동을 선택하도록 ε(t)가 감소한다.

[0081] 슬라이싱 모델(230)은 주어진 행동에 따른 상태 전이를 고려하여 보상(r_t)을 연산한다. 보상은 아래 표 2의 알고리즘(calculateImmediateReward function, 이하 보상 함수)을 통해 연산될 수 있다. 보상 함수는 추정기에서 추정한 실행 비용과 통신 비용을 이용한다(라인 3 ~ 11).

표 2

Algorithm 2 The calculateImmediateReward function

```
procedure CALCULATEIMMEDIATEREWARD(s, a)
    s' ← applyAction(s, a)
    C_s' ← estimateCommunicationCost(s')
    T_s' ← estimatePerformance(s')
    P_s' ← estimatePowerConsumption(s')
    Γ_s' ← calculateCost(C_s', T_s', P_s')
    if T_s' ≤ T_deadline then
        r_s' ← −√Γ_s'
    else
        r_s' ← −Φ
    end if
    return r_s'
end procedure
```

[0083]

[0084] 아래 수학식 10은 보상을 연산하는 수식에 해당한다.

수학식 10

[0086]
$$r_s = \begin{cases} -\sqrt{\Gamma_s} & \text{if } T_s \le T_{\text{deadline}} \\ -\Phi & \text{otherwise} \end{cases}$$

[0087] Γ_s는 상태 s에서 전체 비용(전력 비용, 성능(지연) 비용 및 통신 비용 기반으로 결정)으로 에너지 소모, EDP(energy-delay product) 등이 될 수 있으며, Φ는 패널티를 나타낸다. 패널티는 제한시간 조건을 만족하지 못한 경우 100으로 설정될 수 있다. 상태 s에서 타깃 추론을 위한 제한시간 조건을 만족하는 경우, 보상은 $-\sqrt{\Gamma_s}$로 설정될 수 있다. 즉, 비용 감소에 따라 보상이 증가하는 형태이다. 반대로 제한시간 조건을 만족하지 못한 경우, 해당 상태를 배제하게 된다.

[0088] 슬라이싱 모델(230)은 종전에 확인된 가장 높은 보상보다 현재 보상이 높은 경우 다른 단계 상태 s_{t+1}를 결정하고 저장한다. 슬라이싱 모델(230)은 메모리에 (s_t, a_t, r_t, s_{t+1})을 기록한다.

[0089] 이후 슬라이싱 모델(230)은 메모리에 저장된 샘플들을 임의로 선택하여 훈련 데이터 배치를 생성하고, 선택한 훈련 데이터 배치를 이용하여 Q 네트워크를 학습시킨다. 이 과정을 통해 학습된 Q 네트워크는 에이전트가 보다 효과적인 행동을 선택하게 한다. 슬라이싱 모델(230)은 이와 같은 과정을 일정한 횟수 반복하면서 가장 높은 보상을 갖는 최적의 상태 s_{best}를 산출한다. 상태는 특정 딥러닝 모델의 슬라이스들, 각 슬라이스를 수행하는 연산 장치, 각 연산 장치의 주파수를 포함한다.

[0091] 도 3에서 런타임 시스템(240)은 슬라이싱 모델(230)이 최종적으로 산출한 상태(=슬라이싱 정보)를 기준으로 해당 딥러닝 모델을 타깃 이기종 컴퓨팅 시스템 환경에서 수행할 수 있다.

[0093] 연구자는 이기종 컴퓨팅 시스템 환경을 구성하여 다양한 딥러닝 모델이 제대로 동작하는지 검증하였다. 연구자는 HiKey 970 개발 보드를 사용하여 이기종 컴퓨팅 시스템을 구성하였다. 해당 시스템은 빅 코어 4개와 리틀 코어 4개를 갖는 CPU, Mail-G72 GPU 및 NPU로 구성된 Kirin 970 모바일 프로세서를 탑재한 시스템이다. 아래 표 3은 검증을 위하여 구성한 이기종 컴퓨팅 시스템에 대한 정보이다.

표 3

[0094]

Device	Voltage and frequency levels (V, MHz)
Big core cluster	(0.7, 682), (0.8, 1018), (0.8, 1210), (0.8, 1364), (0.9, 1498), (0.9, 1652), (0.9, 1863), (1.0, 2093), (1.1, 2362)
Little core cluster	(0.7, 509), (0.8, 1018), (0.9, 1210), (0.9, 1402), (1.0, 1556), (1.0, 1690), (1.1, 1844)
GPU	(0.6, 104), (0.7, 151), (0.7, 237), (0.7, 332), (0.8, 415), (0.8, 550), (0.9, 667), (1.0, 767)

[0095] NPU는 메모리 제한으로 100MB보다 큰 슬라이스 수행하지 못한다. 빅 코어, 리틀 코어 및 GPU는 메모리 제한 사항이 알려지지 않았으나 실제 추론 과정에서 메모리 부족 현상 없이 추론을 수행하여 해당 연산장치들은 메모리 제한 조건이 없는 것으로 가정하였다.

[0096] 이기종 컴퓨팅 시스템은 Android 8.1을 운영 OS로 사용하였고, 추론 모델은 TensorFlow Lite 1.11.0으로 구현하였다.

[0097] 이와 같은 검증 시스템 환경에서 연구자는 아래 표 4와 같은 추론 모델들에 대하여 모델 슬라이싱을 하고 추론을 수행하였다. Inception V4 (IN), MnasNet (width parameters, p_W = 1.0, MN-1.0) 및 MnasNet(p_W = 1.3, MN-1.3), MobileNet V2(p_W = 1.3, MO-1.3), MobileNet V2(p_W = 1.4, MO-1.4), ResNet V2 (RN) 및 VGG (VGG)를 사용하였다. 실험 결과 모델마다 다소 편차가 있지만, 유의미한 정확도(accuracy)를 보였다. 연구자는 메모리 사용량, 계층 개수 등의 측면에서 다양한 특성을 보이는 추론 모델들을 사용하였다.

표 4

[0099]

Workload	Accuracy (%)	Size (MB)	Layers	short/medium/long deadlines in milliseconds
Inception V4 (IN) [50]	80.1	183.7	20	190/210/250
MnasNet with $p_W = 1.0$ (MN-1.0) [51]	74.1	321.9	20	85/100/130
MnasNet with $p_W = 1.3$ (MN-1.3) [51]	75.2	511.6	20	105/120/150
MobileNet V2 with $p_W = 1.3$ (MO-1.3) [43]	74.4	187.8	18	75/90/120
MobileNet V2 with $p_W = 1.4$ (MO-1.4) [43]	75.0	214.7	18	85/100/130
ResNet V2 (RN) [21]	77.8	260.4	53	155/170/200
VGG (VGG) [45]	71.5	407.4	16	135/150/180

[0100] 아래 표 5는 검증 과정에서 산출된 모델 슬라이싱 정보 및 실행 계획에 대한 내용이다. 제한 시간 조건에서 최적의 에너지를 사용하는 조건으로 생성된 정보이다. B는 빅 코어, L은 리틀 코어, G는 GPU, N은 NPU를 나타낸다. 표 5에서 숫자는 해당 슬라이스를 시작하는 계층 인덱스와 해당 연산 장치에서의 주파수를 나타낸다.

표 5

[0102]

Workload	Model slicing and execution plan
IN	$N_1^{960}, N_{11}^{960}, L_{20}^{1690}$
MN-1.0	$N_1^{960}, L_{14}^{1690}, N_{18}^{960}, L_{20}^{1402}$
MN-1.3	$N_1^{960}, B_9^{1210}, G_{12}^{332}, B_{14}^{1210}, G_{18}^{332}, B_{20}^{1210}$
MO-1.3	$N_1^{960}, G_8^{550}, L_{15}^{1402}, N_{18}^{960}$
MO-1.4	$N_1^{960}, G_9^{667}, L_{15}^{1690}, N_{18}^{960}$
RN	$B_1^{1863}, N_2^{960}, L_4^{1690}, N_5^{960}, B_{12}^{1364}, N_{13}^{960}, N_{30}^{960}, B_{48}^{1364}, N_{49}^{960}, B_{53}^{1863}$
VGG	$N_1^{960}, G_{14}^{332}, G_{15}^{237}, B_{16}^{682}$

[0104] 또한, 상술한 바와 같은 딥러닝 모델 분할 방법 내지 분할된 모델을 이용한 이기종 컴퓨팅 시스템에서의 모델

추론 방법은 컴퓨터에서 실행될 수 있는 실행가능한 알고리즘을 포함하는 프로그램(또는 어플리케이션)으로 구현될 수 있다. 상기 프로그램은 일시적 또는 비일시적 판독 가능 매체(non-transitory computer readable medium)에 저장되어 제공될 수 있다.

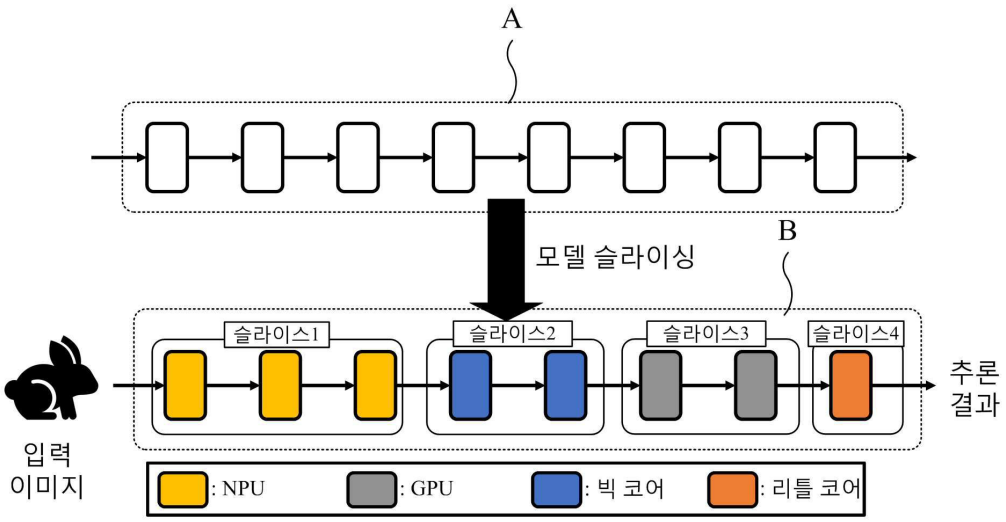
[0105] 비일시적 판독 가능 매체란 레지스터, 캐쉬, 메모리 등과 같이 짧은 순간 동안 데이터를 저장하는 매체가 아니라 반영구적으로 데이터를 저장하며, 기기에 의해 판독(reading)이 가능한 매체를 의미한다. 구체적으로는, 상술한 다양한 어플리케이션 또는 프로그램들은 CD, DVD, 하드 디스크, 블루레이 디스크, USB, 메모리카드, ROM(read-only memory), PROM (programmable read only memory), EPROM(Erasable PROM, EPROM) 또는 EEPROM(Electrically EPROM) 또는 플래시 메모리 등과 같은 비일시적 판독 가능 매체에 저장되어 제공될 수 있다.

[0106] 일시적 판독 가능 매체는 스태틱 램(Static RAM, SRAM), 다이내믹 램(Dynamic RAM, DRAM), 싱크로너스 디램(Synchronous DRAM, SDRAM), 2배속 SDRAM(Double Data Rate SDRAM, DDR SDRAM), 증강형 SDRAM(Enhanced SDRAM, ESDRAM), 동기화 DRAM(Synclink DRAM, SLDRAM) 및 직접 램버스 램(Direct Rambus RAM, DRRAM) 과 같은 다양한 RAM을 의미한다.

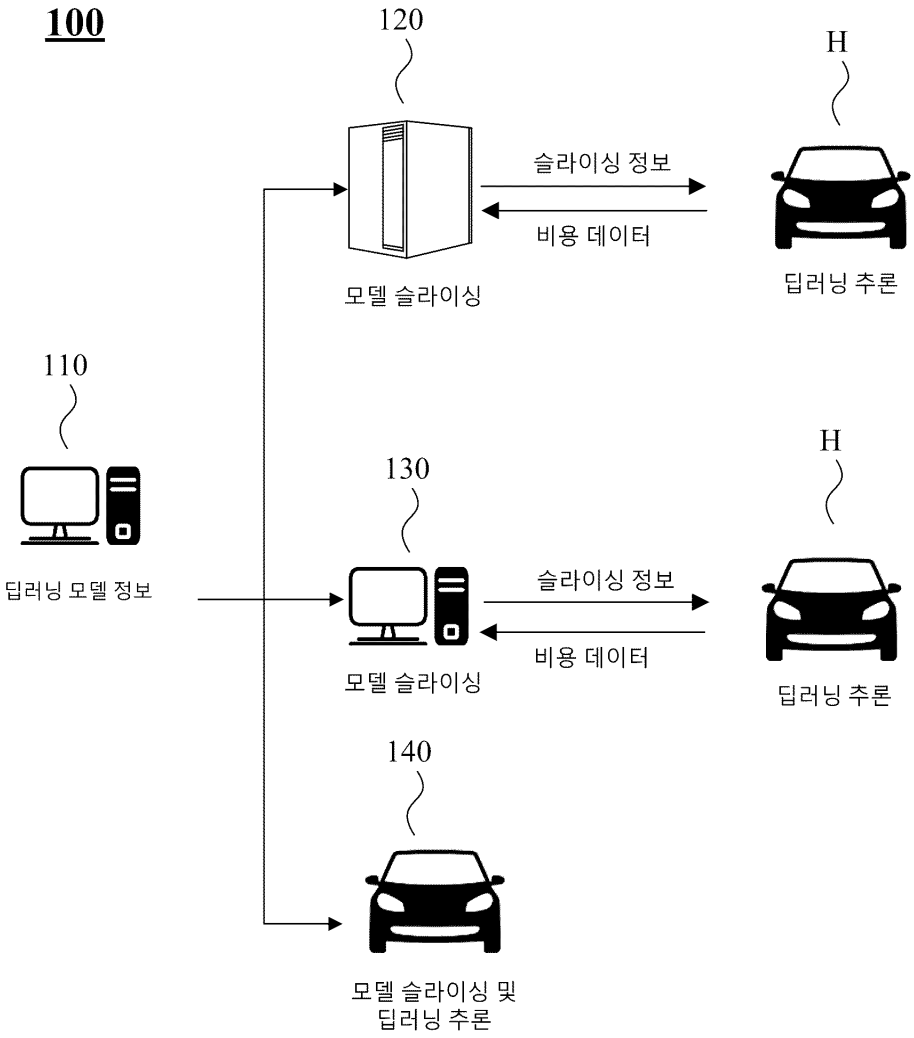
[0107] 본 실시예 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

도면

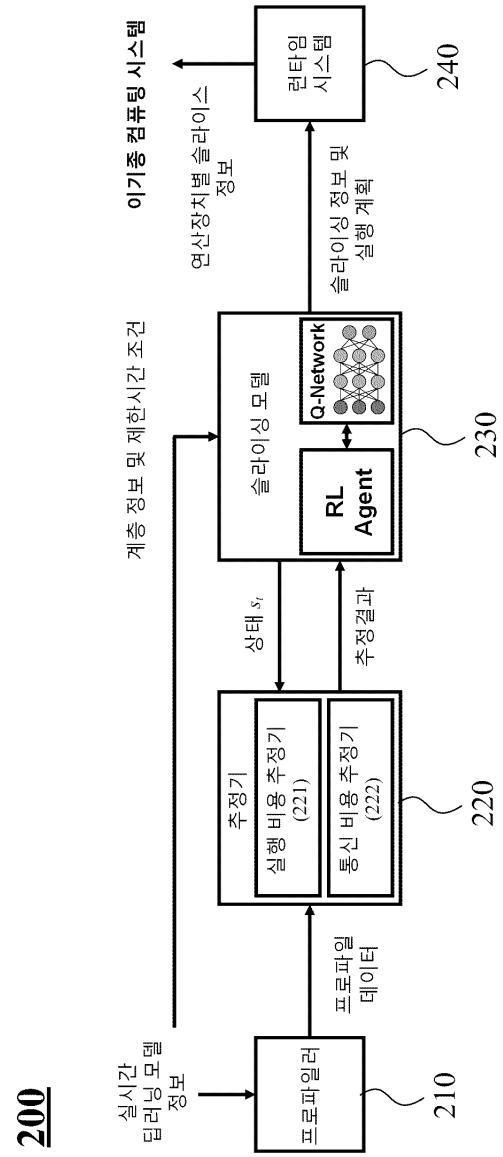
도면1



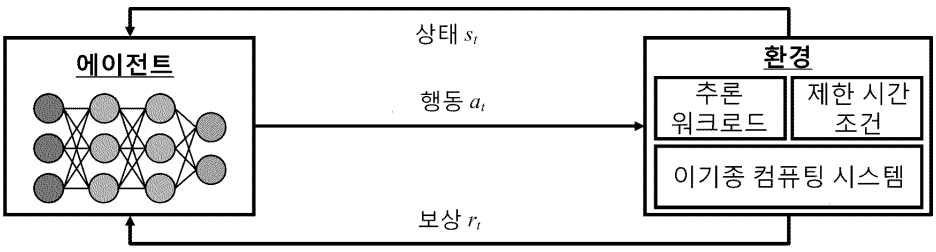
도면2


100
120
H
슬라이싱 정보
비용 데이터
모델 슬라이싱
딥러닝 추론
110
딥러닝 모델 정보
130
H
슬라이싱 정보
비용 데이터
모델 슬라이싱
딥러닝 추론
140
모델 슬라이싱 및
딥러닝 추론

도면3


200
실시간 딥러닝 모델 정보
프로파일러
210
프로파일 데이터
추정기
실행 비용 추정기 (221)
통신 비용 추정기 (222)
220
상태 s_t
추정결과
슬라이싱 모델
Q-Network
RL Agent
230
계층 정보 및 제한시간 조건
이기종 컴퓨팅 시스템
연산장치별 슬라이스 정보
슬라이싱 정보 및 실행 계획
런타임 시스템
240

도면4


에이전트
상태 s_t
행동 a_t
보상 r_t
환경
추론 워크로드
제한 시간 조건
이기종 컴퓨팅 시스템