(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0073271
(43) 공개일자 2024년05월27일

(51) 국제특허분류(Int. Cl.)
G06F 9/50 (2018.01) *G06N 3/04* (2023.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G06F 9/5038 (2013.01)
G06F 9/5016 (2013.01)
(21) 출원번호 10-2022-0154609
(22) 출원일자 2022년11월17일
심사청구일자 2022년11월17일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
신창용
서울특별시 동대문구 무학로49길 25, 대명빌딩 303호
양경식
서울특별시 동대문구 무학로45길 40, 401호(용두동)
(뒷면에 계속)
(74) 대리인
김등용

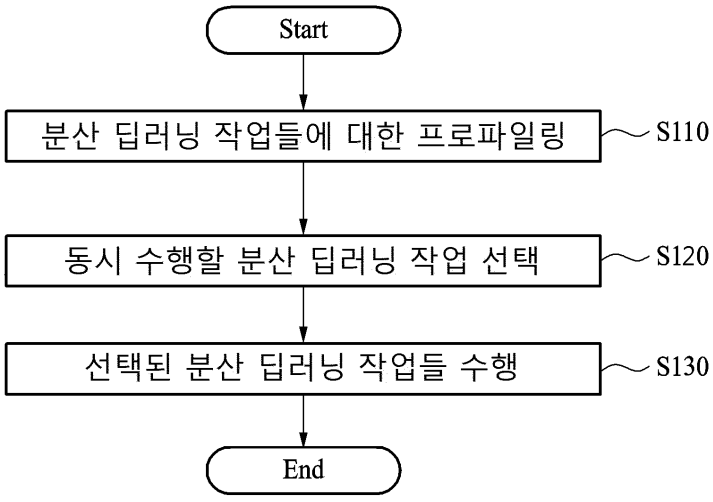
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 **프로파일링 기반 분산 딥러닝 작업 실행 순서 조정 방법 및 장치**

(57) 요 약

프로파일링 기반 분산 딥러닝 작업 실행 순서 조정 방법 및 장치가 개시된다. 상기 조정 방법은 적어도 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 분산 딥러닝 작업 실행 순서 조정 방법으로서, 복수의 분산 딥러닝 작업들 각각에 대한 프로파일링을 수행하는 단계, 및 프로파일링 결과에 기초하여, 동시에 수행할 분산 딥러닝 작업들을 선택하는 단계를 포함한다.

대 표 도 - 도7



(52) CPC특허분류
G06F 9/5072 (2013.01)
G06F 9/5077 (2013.01)
G06F 9/5083 (2013.01)
G06N 3/04 (2023.01)
G06N 3/08 (2023.01)

(72) 발명자
유연호
서울특별시 성북구 고려대로24가길 14-6, 102호
이정환
서울특별시 성북구 안암로9가길 56. 303호
유혁
서울특별시 용산구 한강대로 205, A-803호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711152940
과제번호	2015-0-00280-008
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	SW컴퓨팅산업원천기술개발
연구과제명	(SW 스타랩) 성능 및 보안 SLA 보장이 가능한 차세대 클라우드 인프라SW 개발
기 여 율	1/2
과제수행기관명	고려대학교산학협력단
연구기간	2022.01.01 ~ 2022.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1345347378
과제번호	2021R1A6A1A13044830
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학학술연구기반구축
연구과제명	융합소프트웨어연구소
기 여 율	1/2
과제수행기관명	고려대학교 산학협력단
연구기간	2022.03.01 ~ 2023.02.28

명 세 서

청구범위

청구항 1

적어도 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 분산 딥러닝 작업 실행 순서 조정 방법에 있어서,

복수의 분산 딥러닝 작업들 각각에 대한 프로파일링을 수행하는 단계; 및

프로파일링 결과에 기초하여, 동시에 수행할 분산 딥러닝 작업들을 선택하는 단계를 포함하는 분산 딥러닝 작업 실행 순서 조정 방법.

청구항 2

제1항에 있어서,

상기 프로파일링을 수행하는 단계는,

상기 복수의 분산 딥러닝 작업들 각각의, 활성 구간, 유휴 구간, 및 GPU 메모리 활용률을 추출하는,

분산 딥러닝 작업 실행 순서 조정 방법.

청구항 3

제2항에 있어서,

상기 선택하는 단계는,

상기 복수의 분산 딥러닝 작업들 중 어느 하나의 분산 딥러닝 작업을 선택하는 단계; 및

상기 활성 구간, 상기 유휴 구간, 및 상기 GPU 메모리 활용률에 기초하여, 상기 복수의 분산 딥러닝 작업들 중에서 상기 어느 하나의 분산 딥러닝 작업과 동시 실행할 다른 하나의 분산 딥러닝 작업을 선택하는 단계를 포함하는,

분산 딥러닝 작업 실행 순서 조정 방법.

청구항 4

제3항에 있어서,

상기 어느 하나의 분산 딥러닝 작업을 선택하는 단계는, 작업 대기열의 첫 번째 분산 딥러닝 작업을 상기 어느 하나의 분산 딥러닝 작업으로 선택하는,

분산 딥러닝 작업 실행 순서 조정 방법.

청구항 5

제3항에 있어서,

상기 다른 하나의 분산 딥러닝 작업을 선택하는 단계는,

상기 복수의 분산 딥러닝 작업들 중에서, GPU 메모리 용량에서 상기 어느 하나의 분산 딥러닝 작업의 최대 GPU 메모리 활용률을 뺀 값보다 큰 메모리를 요구하는 분산 딥러닝 작업들을 필터링하는 단계를 포함하는,

분산 딥러닝 작업 실행 순서 조정 방법.

청구항 6

제5항에 있어서,

상기 다른 하나의 분산 딥러닝 작업을 선택하는 단계는,

필터링된 분산 딥러닝 작업들 중에서, 상기 어느 하나의 분산 딥러닝 작업의 활성-유휴 비율의 역수와 가장 근

접한 활성-유휴 비율을 갖는 분산 딥러닝 작업을 상기 다른 하나의 분산 딥러닝 작업으로 선택하는,

분산 딥러닝 작업 실행 순서 조정 방법.

청구항 7

제6항에 있어서,

상기 활성-유휴 비율은 상기 활성 구간과 상기 유휴 구간의 비율인,

분산 딥러닝 작업 실행 순서 조정 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 GPU 클라우드에서 분산 딥러닝 작업 완료 속도 개선을 위한 딥러닝 작업 실행 순서 조정 방법에 관한 것으로, 특히 분산 딥러닝 작업에 대한 GPU 활성/유휴 구간 및 GPU 메모리 사용량에 대한 프로파일링 결과를 기반으로 동시 실행되기에 가장 적합한 분산 딥러닝 작업을 선택하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 최근의 딥러닝 모델은 정확도 향상을 위해 모델의 레이어(layer) 개수와 파라미터 수가 증가하고 있으며, 입력되는 학습 데이터의 크기도 증가하고 있다. 따라서, 단일 GPU만을 사용하여 딥러닝 모델을 학습 시 천문학적인 시간이 소요되며, 단일 GPU에서는 GPU 메모리 부족으로 인해 학습이 불가한 경우도 발생하고 있다. 따라서, 복수 개의 GPU를 사용하여 딥러닝 모델의 학습 속도를 가속하거나, 단일 GPU에서 불가능했던 딥러닝 모델의 학습을 가능케 한다.

[0003] 분산 딥러닝(Distributed Deep Learning, DDL) 학습은 학습 데이터를 분산하는 방식과 딥러닝 모델을 분산하는 방식으로 구분된다. 학습 데이터를 분산하는 경우 동일한 딥러닝 모델을 복수 개의 GPU에서 학습하기 때문에, 학습 결과를 동기화하는 과정이 필수적으로 요구된다. 동기화 방식에는 중앙에서 학습 결과를 수집한 후, 각 GPU에게 학습 결과를 배포하는 방식인 PS(Parameter Server) 방식과 각 GPU가 서로 학습 결과를 공유하는 올-리듀스(All-reduce) 방식이 존재한다.

[0004] 일반적으로, 클라우드는 대규모의 컴퓨팅 자원을 운영하고 컴퓨팅 자원 활용을 자동화하는 서비스를 제공하며, 사용자가 요구한 컴퓨팅 자원을 필요에 따라 빌려주는 컴퓨팅 형태를 말한다. 클라우드에서 제공하는 컴퓨팅 자원의 형태에 따라 SaaS(Software as a Service), IaaS(Infrastructure as a Service), PaaS(Platform as a Service) 등으로 구분된다.

[0005] 최근, 고가의 GPU 자원이 딥러닝 모델 학습에 필수적으로 사용되면서, 클라우드를 기반으로 한 모델 학습이 매우 활발해졌다. 클라우드 컴퓨팅 업체는 IaaS와 같은 형태로 GPU 자원을 직접 제공하거나, 딥러닝을 포함한 기계학습에 최적화된 도구를 제공하는 서비스인 MLaaS(Machine Learning as a Service)를 제공한다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 대한민국 공개특허 제2021-0115370호 (2021.09.27. 공개)

(특허문헌 0002) 대한민국 공개특허 제2020-0109917호 (2020.09.23. 공개)

(특허문헌 0003) 대한민국 등록특허 제2029711호 (2019.10.01. 공고)

발명의 내용

해결하려는 과제

[0007] 본 발명이 이루고자 하는 기술적인 과제는 DDL(분산 딥러닝) 작업의 GPU 활용 패턴 및 JCT(Job Completion Time)가 매우 다양함을 지적하고 이를 해결하기 위해 DDL 작업을 프로파일링 함으로써 동시 실행되기에 가장 적

합한 작업을 선택하여 작업이 실행되는 순서를 조정하고 이를 통해 모든 작업이 완료되는 시간(JCT)를 단축하는 것이다.

[0008] 기존 GPU 클라우드에서 GPU 활용률이 매우 저조하다는 연구들이 제시되었고 이를 해결하기 위해 동시에 2개 이상의 작업을 실행하는 연구들이 제안되었다. 딥러닝 모델 학습 작업이 실행 중이지만 GPU 활용률이 저조한 GPU에 추가로 딥러닝 모델 학습 작업을 실행시켜 GPU 활용률을 개선할 수 있다.

[0009] 그러나, 여전히 동시 실행되는 작업의 GPU 사용 특성이 고려되고 있지 않다. 동시 실행되는 분산 딥러닝 작업에 따라 JCT가 크게 변화하기 때문에 작업의 특성에 대한 고려가 필수적이다. 또한, GPU 클라우드에서는 OoM(Out-of-Memory) 오류가 해결되지 못하고 있다. GPU 활용률을 극대화하기 위해 단일 GPU에서 두 개 이상의 딥러닝 작업이 실행되지만, 동시 실행되는 작업이 요구하는 GPU 메모리의 합이 GPU 메모리 용량을 초과할 경우 OoM 에러가 발생하여 학습이 중단된다.

과제의 해결 수단

[0010] 본 발명의 일 실시예에 따른, 프로파일링 기반 분산 딥러닝 작업 실행 순서 조정 방법은 적어도 프로세서를 포함하는 컴퓨팅 장치에 의해 수행되는 분산 딥러닝 작업 실행 순서 조정 방법으로서, 복수의 분산 딥러닝 작업들 각각에 대한 프로파일링을 수행하는 단계, 및 프로파일링 결과에 기초하여, 동시에 수행할 분산 딥러닝 작업들을 선택하는 단계를 포함한다.

[0011] 또한, 상기 프로파일링을 수행하는 단계는, 상기 복수의 분산 딥러닝 작업들 각각의, 활성 구간, 유휴 구간, 및 GPU 메모리 활용률을 추출할 수 있다.

[0012] 또한, 상기 선택하는 단계는, 상기 복수의 분산 딥러닝 작업들 중 어느 하나의 분산 딥러닝 작업을 선택하는 단계, 및 상기 활성 구간, 상기 유휴 구간, 및 상기 GPU 메모리 활용률에 기초하여, 상기 복수의 분산 딥러닝 작업들 중에서 상기 어느 하나의 분산 딥러닝 작업과 동시 실행할 다른 하나의 분산 딥러닝 작업을 선택하는 단계를 포함할 수 있다.

[0013] 또한, 상기 어느 하나의 분산 딥러닝 작업을 선택하는 단계는, 작업 대기열의 첫 번째 분산 딥러닝 작업을 상기 어느 하나의 분산 딥러닝 작업으로 선택할 수 있다.

[0014] 또한, 상기 다른 하나의 분산 딥러닝 작업을 선택하는 단계는, 상기 복수의 분산 딥러닝 작업들 중에서, GPU 메모리 용량에서 상기 어느 하나의 분산 딥러닝 작업의 최대 GPU 메모리 활용률을 뺀 값보다 큰 메모리를 요구하는 분산 딥러닝 작업들을 필터링하는 단계를 포함할 수 있다.

[0015] 또한, 상기 다른 하나의 분산 딥러닝 작업을 선택하는 단계는, 필터링된 분산 딥러닝 작업들 중에서, 상기 어느 하나의 분산 딥러닝 작업의 활성-유휴 비율의 역수와 가장 근접한 활성-유휴 비율을 갖는 분산 딥러닝 작업을 상기 다른 하나의 분산 딥러닝 작업으로 선택할 수 있다.

[0016] 또한, 상기 활성-유휴 비율은 상기 활성 구간과 상기 유휴 구간의 비율을 의미할 수 있다.

발명의 효과

[0017] 본 발명의 일 실시예에 따른 분산 딥러닝 작업 실행 순서 조정 방법 및 장치에 의할 경우, 동시 실행되는 작업의 GPU 사용 특성 및 메모리 요구량을 프로파일링하고, 동시 실행되기에 적합한 분산 딥러닝 작업을 선정하여 작업 순서를 조정함으로써 전체 작업의 JCT의 합을 줄일 수 있다.

[0018] 따라서, 본 발명은 분산 딥러닝의 학습 속도를 딥러닝 라이브러리 수준에서의 프로파일링과 GPU 클라우드 작업 배치 단계에서 개선할 수 있으므로, 분산 딥러닝이 주로 수행되는 GPU 클라우드에서 활용하기에 효과적인 기법이다.

도면의 간단한 설명

[0019] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 5초간 ImageNet 학습 데이터로 훈련된 VGG16 모델의 GPU 활용률을 나타내는 그래프이다.

도 2는 분산 딥러닝 작업에 대한 GPU 사용 패턴의 프로파일링 결과를 도시한다.

도 3은 분산 딥러닝 작업 집합들 각각에 대한 정규화된 JCT를 나타내는 그래프이다.

도 4는 딥러닝 라이브러리 수준에서 GPU의 활성 구간과 유휴 구간을 측정하는 방법을 설명하기 위한 도면이다.

도 5는 성능 평가 결과로써, 50번의 실험 결과의 평균과 분포상에서 75%, 90% 및 99%에 위치하는 JCT를 나타낸다.

도 6은 본 발명의 일 실시예에 따른 분산 딥러닝 작업 실행 순서 조정 장치를 설명하기 위한 기능 블럭도이다.

도 7은 본 발명의 일 실시예에 따른 분산 딥러닝 작업 실행 순서 조정 방법을 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0020] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시예들에 대해서 특정한 구조적 또는 기능적 설명들은 단지 본 발명의 개념에 따른 실시예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시예들에 한정되지 않는다.

[0021] 본 발명의 개념에 따른 실시예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0022] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0023] 어떤 구성 요소가 다른 구성 요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성 요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성 요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성 요소가 다른 구성 요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는 중간에 다른 구성 요소가 존재하지 않는 것으로 이해되어야 할 것이다. 구성 요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.

[0024] 본 명세서에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0025] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0026] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시예들을 상세히 설명한다. 그러나, 특허출원의 범위가 이러한 실시예들에 의해 제한되거나 한정되는 것은 아니다. 각 도면에 제시된 동일한 참조 부호는 동일한 부재를 나타낸다.

[0027] 딥러닝 라이브러리는 다양한 딥러닝 알고리즘을 사용할 수 있는 추상화된 API를 제공하여 편리하게 딥러닝 모델을 개발할 수 있게 하는 프레임워크이다. 본 발명에서는 대표적인 딥러닝 라이브러리인 TensorFlow를 중심으로, 분산 딥러닝 작업의 프로파일링을 위해 내부 실행 구조를 분석하였다. 그러나, 본 발명이 이에 제한되는 것이 아님은 자명하다. 딥러닝 라이브러리의 실행 구조는 다음과 같다.

[0028] 우선 고수준 API로 작성된 코드가 행렬곱을 포함한 다양한 세부 연산으로 변환되며, 변환된 세부 연산들은 연산 순서 및 의존성 정보가 포함된 연산 그래프로 표현된다. 연산 그래프의 노드로써 표현되는 세부 연산은 그래프를 따라 순차적으로 실행되며, 실제로 실행될 장치에 구현된 동작으로 수행된다.

[0029] 본 발명에서 목표로 하는 분산 딥러닝은 데이터 분산 방식으로 학습 주기마다 각 GPU의 학습 결과를 동기화하는 과정이 필수적이며, 동기화 과정으로 인하여 GPU 자원의 활성 및 유휴 구간이 반복적으로 나타난다. 도 1은 5초간 ImageNet 학습 데이터로 훈련된 VGG16 모델의 GPU 활용률을 나타내는 그래프이다. 도 1을 참조하면, 총 4번의 활성 및 유휴 구간이 반복되며, 대부분의 딥러닝 모델의 학습 작업에서 유사한 GPU 사용 패턴을 보인다.

[0030] 본 발명에서는 분산 딥러닝 라이브러리인 TensorFlow에서 제공하는 벤치마킹 도구를 활용하여, 600여개의 분산 딥러닝 작업에 대해 GPU 사용 패턴을 프로파일링하였으며, 결과는 도 2에 도시되어 있다. 도 2를 참조하면, 활성 구간은 5ms에서 1.8s까지 나타났으며 유휴 구간은 5ms에서 4.9s까지 나타나, 최대 1000배가량 차이가 남을 알 수 있었다. GPU 활용 및 유휴 구간이 다양성을 보이기 때문에 딥러닝 작업의 GPU 사용 패턴에 대한 프로파일링이 필수적이다.

[0031] 본 발명에서는 분산 딥러닝 작업의 JCT 개선을 목표로 한다. JCT(Job Completion Time)는 분산 딥러닝 작업이 GPU 클라우드에 요청된 시간부터, 분산 딥러닝 작업이 완료되어 작업을 요청한 사용자에게 결과가 반환되기까지의 시간을 의미한다.

[0032] 분산 딥러닝 작업이 작업 실행 순서에 따라 JCT가 변화함을 보이기 위한 실험을 진행하였다. 600여개의 분산 딥러닝 작업에서 10개의 작업을 무작위로 선택하여 총 3개의 작업 집합(A, B, 및 C)을 구성하였다. 실험은 각 작업 집합에서 무작위로 2개의 작업을 선택하여 단일 GPU에서 동시 실행하며, 실행한 작업은 작업 집합에서 제외하였다. 2개의 작업 중 한 개의 작업이 완료될 경우 작업 집합에 남아있는 작업 중 무작위로 작업을 선택하여 동시 실행하며, 이 과정을 작업 집합에 남은 작업이 없을 때까지 반복하여 JCT를 측정하였다. 측정된 JCT들은 JCT의 최댓값을 기준으로 정규화한 값을 정규화된 JCT(normalized JCT)라고 한다. 정규화된 JCT는 도 3에 도시되어 있다. 이때, 각 작업 집합에 대해 25번의 실험을 반복한 결과, 정규화된 JCT는 1.2배(작업 집합 A)에서 1.5배(작업 집합 B)까지 증가하였으며, 분산 딥러닝 작업 실행 순서가 분산 딥러닝의 학습 속도, 즉 효율성 측면의 변동성(variability)에 큰 영향을 미친다는 것을 확인할 수 있다.

[0033] 본 발명은 분산 딥러닝 작업 가속화를 위한 동시 구동 작업의 실행 순서 조정 방법 및 장치(또는 시스템)에 관한 것으로, 크게 두 가지의 기술로 구성된다. 두 가지 기술은 각각 1) 분산 딥러닝 작업 프로파일링 방법과 2) 이상적인 동시 구동 분산 딥러닝 작업 선정 방법이다. 이에 대하여는 순차적으로 설명한다.

[0034] 분산 딥러닝에 대한 프로파일링을 위해서 GPU에서 작업이 실행될 때 GPU의 활성 구간과 유휴 구간에 대한 측정을 수행하여야 한다. GPU 프로파일링 도구인 NVML(Nvidia management library)와 같은 기존의 기법을 이용하여 GPU 활용률을 측정할 수 있지만, GPU의 활성 구간과 유휴 구간을 측정할 수는 없다.

[0035] 분산 딥러닝 작업의 GPU 활성 및 유휴 구간을 측정하기 위해 TensorFlow 1.6 버전의 실행 구조 분석 내용을 활용할 수 있다. 딥러닝 라이브러리 코드 내부에서 각 세부 연산의 실행 시작 시간과 종료 시간을 어떤 장치에서 실행된 세부 연산인지를 구분하여 기록한다. GPU 활성 구간은 연속적으로 실행된 세부 연산 중 첫 세부 연산의 시작 시간부터 마지막 세부 연산의 종료 시간가지로 정의하며, GPU 유휴 구간은 마지막 세부 연산의 종료 시점부터 그 다음 연속적으로 실행된 세부 연산 중 첫 세부 연산의 시작 시간까지로 정의한다. 프로파일링 결과를 활용하여 동시 실행되기에 가장 적합한 분산 딥러닝 작업을 선정할 수 있다.

[0036] 도 4는 GPU의 활성 구간과 유휴 구간을 측정하는 방법을 설명하기 위한 도면이다. 예시적으로, GPU 활성 및 유휴 구간 측정을 위해 TensorFlow 딥러닝 라이브러리를 코드 레벨로 분석하여 내부 작동 방식을 파악할 수 있고, TensorFlow 코드로부터 변환된 세부 연산 단위인 op의 시작 시간과 종료 시간을 측정함으로써 활성(Active) 구간과 유휴(Idle) 구간을 측정할 수 있다. 도 4에서, op는 TensorFlow 코드로부터 현환된 세부 연산 단위를 의미한다. 일 실시예에 따르면, TensorFlow 코드에 로그 코드를 추가하고, 로그 파싱을 통해 활성 구간과 유휴 구간을 측정할 수 있다.

[0037] 분산 딥러닝 작업의 GPU 메로리 사용량을 측정하기 위해 작업의 최대 GPU 사용량을 측정한다. 분산 딥러닝 작업의 메모리 사용 패턴은 GPU 활용률 패턴과 유사하게 주기적인 패턴을 보이며, 최대 사용량을 측정하기 위해 NVML 도구를 사용할 수 있다. 분석 결과를 활용하여 분산 딥러닝 작업의 실행 과정에서 발생할 수 있는 OoM 오류를 사전에 방지할 수 있다. OoM(Out-of-Memory) 오류는 GPU에서 실행 중인 작업에 필요한 메모리가 GPU 메모리의 용량보다 클 때 발생한다. OoM 오류가 발생하는 경우, GPU에서 실행 중인 작업이 중단되고 작업의 처음부터 재시작하여야 한다.

[0038] 예시적으로, GPU 메모리 사용량 측정을 위해 GPU 제조사인 NVIDIA에서 제공하는 NVML API를 미리 정해진 간격(예컨대, 1/6초 간격)으로 호출하여 메모리 사용량을 측정할 수 있다. 이때, 여러 번의 반복(iteration) 중 가

장 높은 GPU 메모리 사용량을 기준으로 동시 실행되기에 적합한 분산 딥러닝 작업을 선정할 수 있다.

[0039] 각 분산 딥러닝 작업을 분석(프로파일링)하는데 평균적으로 63.4초가 소요되었다. 일반적으로 GPU 클라우드에서 실행되는 분산 딥러닝 작업의 평균 소요 시간이 수백 분에 달하기 때문에, 프로파일링에 소요되는 시간은 대기열에서 작업의 대기 시간과 겹칠 수 있으며, 큰 오버헤드로 인식되지 않는다.

[0040] 분산 딥러닝 작업들의 JCT는 작업 실행 순서에 따라 크게 변화하는 것으로 나타났고, 분산 딥러닝 작업의 GPU 활성 및 유휴 구간도 다양하게 나타났다. 동시에 실행되는 작업의 유휴 구간이 짧으면 다른 작업이 GPU를 활용할 기회가 줄어들고, 반대로 유휴 구간이 길면 다른 작업이 GPU를 활용할 기회가 더 많아진다. 따라서, 딥러닝 작업의 실행 순서를 프로파일링 결과를 고려하여 조정함으로써 JCT를 보다 효율적으로 감소시킬 수 있다.

[0041] 우선 작업 대기열에서 첫 번째 작업(J_A), 첫 번째로 수행할 분산 딥러닝 작업을 선택하고, J_A와 동시 실행할 다른 분산 딥러닝 작업(J_P)을 찾기 위해, 두 가지 요소의 관점에서 적합도를 점수화한다. 첫 번재 요소는 GPU 메모리 사용량이다. OoM 오류를 피하기 위해, 작업 대기열의 작업들 중에서, 현재 사용 가능한 GPU 메모리(GPU 메모리 용량에서 J_A의 최대 GPU 메모리 사용량을 뺀 값) 보다 큰 메모리가 필요한 분산 딥러닝 작업을 필터링한다. 필터링된 작업(Q_filtered)에 대하여, 두 번째 요소인 작업 간의 활성 구간 및 유휴 구간의 적합성을 고려한다. Q_filtered의 각 작업(J_n)에 대해 J_n의 활성 구간을 유휴 구간으로 나누어 활성-유휴 비율(r_n)을 계산한다. 예컨대, r_A(J_A의 활성-유휴 비율)가 6/4라고 한다면, 적절한 J_P는 J_P의 활성 구간이 J_A의 유휴 구간과 겹칠 수 있기 때문에, r_P가 r_A의 역수(4/6)인 작업이 된다. 정리하면, 다음 수학식을 사용하여 J_A와 동시에 실행되기에 가장 적합한 J_P를 선정할 수 있다.

[0042] [수학식 1]

[0043] $$J_P = \min_{J_n \in Q_{filtered}} |r_A \times r_n - 1|$$

[0044] 수학식 1을 참조하면, 첫 번째 작업(J_A)의 활성-유휴 비율과의 곱이 1과 가장 가깝게 만드는 활성-유휴 비율을 갖는 작업이 동시 실행되는 작업으로 선정될 수 있다. 다시 말하면, 첫 번째 작업(J_A)의 활성-유휴 비율의 역과 가장 차이가 적은 활성-유휴 비율을 갖는 작업이 동시 실행되는 작업으로 선정되는 것으로 이해될 수도 있다.

[0045] 성능 평가를 위해, 600여개의 분산 딥러닝 작업 중에서 무작위로 선택한 10개의 작업을 세 가지 실행 방식으로 실행하여, 무작위로 선택된 작업을 변경하여 50번 반복 실험하였다. 세 가지 실행 방식은 각각 동시에 1개의 작업만 실행하는 기법(Serial), 동시에 2개의 작업을 무작위로 선택하여 실행하는 기법(Parallel), 및 동시에 2개의 작업을 프로파일링 기반 동시 실행 작업 선정 기법에 따라 선택하여 실행하는 기법(Xonar)이다. 도 5는 50번의 실험 결과의 평균과 분포상에서 75%, 90% 및 99%에 위치하는 JCT를 나타낸다. 평균 JCT의 경우, Xonar는 Serial과 Parallel에 비하여 JCT를 각각 43.6%와 13.7%를 감소시킨다. 분포상에서 테일(tail)에 위치하는 JCT 결과이 개선 사항은 더 크게 나타났다. Xonar는 99%의 JCT를 736초 이내에 제한하는 반면, Serial 및 Parallel은 각각 1,317초 및 1,083초의 JCT가 소요되었다. 이는 Xonar의 99% 테일 JCT가 Serial과 Parallel에 비해 각각 44.1%와 32% 낮다는 것을 의미한다. 실험 결과를 통해, 본 발명이 분산 딥러닝 작업의 JCT를 상당히 감소시킨다는 것을 확인할 수 있으며, OoM 오류 또한 발생하지 않아 성공적으로 OoM 오류를 방지하였음을 알 수 있다.

[0046] 도 6은 본 발명의 일 실시예에 따른 분산 딥러닝 작업 실행 순서 조정 장치의 기능 블럭도이다.

[0047] 도 6을 참조하면, 분산 딥러닝 장치, 작업 실행 순서 조정 장치, 및 조정 장치 등으로 명명될 수도 있는 분산 딥러닝 작업 실행 순서 조정 장치(100)는 적어도 프로세서(processor) 및/또는 메모리(memory)를 포함하는 컴퓨팅 장치로 구현될 수 있다. 컴퓨팅 장치는 PC(Personal Computer), 서버(server) 등을 포함하고, 하나의 물리적 장치로 구성되거나, 복수의 물리적 장치로 구현될 수 있다.

[0048] 조정 장치(100)는 프로파일링부(110)와 작업 선택부(120)를 포함한다. 실시예에 따라, 조정 장치(100)는 학습부(130) 및/또는 저장부(140)를 더 포함할 수도 있다.

[0049] 프로파일링부(110)는 복수의 분산 딥러닝 작업들 각각에 대한 프로파일링을 수행한다. 복수의 분산 딥러닝 작업들 중 적어도 두 개의 분산 딥러닝 작업들은 복수의 GPU를 사용하여 동시에 학습될 수 있다. 즉, 본 발명은 분산 딥러닝 환경을 전제로 한다. 프로파일링이 결로써, 프로파일링는 복수의 분산 딥러닝 작업들 각각에 대한 활성 구간, 유휴 구간, 및 GPU 메모리 활용률에 관한 정보를 획득할 수 있다. 활성 구간과 유휴 구간은 단일의 이터레이션(iteration)에 대한 활성 구간과 유휴 구간을 의미하거나, 복수의 이터레이션들에 대한 활성 구간의 평균과 복수의 이터레이션들에 대한 유휴 구간이 평균을 의미할 수 있다. 또한, GPU 메모리 활용률은 복수의 이터

레이션들에 대한 학습 도중에 발생하는 GPU 메모리 활용률의 최대값, 즉 최대 GPU 메모리 활용률을 의미할 수 있다.

[0050] 작업 선택부(120)는 동시에 수행될 분산 딥러닝 작업을 선택할 수 있다. 구체적으로, 어느 하나의 분산 딥러닝 작업이 수행되는 경우에, 복수의 분산 딥러닝 작업들 중에서 어느 하나의 분산 딥러닝 작업과 동시에 학습을 수행할 다른 하나의 분산 딥러닝 작업을 선택할 수 있다.

[0051] 보다 구체적으로, 작업 선택부(120)는 미리 정해진 복수의 분산 딥러닝 작업들 중 어느 하나(예컨대, 작업 대기열의 분산 딥러닝 작업들 중 첫 번째 분산 딥러닝 작업)을 선택한 후에, 선택된 어느 하나의 분산 딥러닝 작업과 동시 실행할 다른 분산 딥러닝 작업을 선택할 수 있다. 이를 위해, 작업 선택부(120)는 복수의 분산 딥러닝 작업들 중에서 GPU 메모리 활용률 조건과 활성 및 유휴 구간 조건을 만족하는 어느 하나의 분산 딥러닝 작업을 선택할 수 있다.

[0052] 학습부(130)는 적어도 두 개의 분산 딥러닝 작업들에 대한 학습을 수행할 수 있다. 동시 실행되는 적어도 두 개의 분산 딥러닝 작업들은 작업 선택부(120)에 의해 선택된 어느 하나의 분산 딥러닝 작업과 다른 하나의 분산 딥러닝 작업을 의미할 수 있다. 실시예에 따라, 어느 하나의 분산 딥러닝 작업은 작업 대기열의 첫 번째 분산 딥러닝 작업을, 다른 하나의 분산 딥러닝 작업은 작업 선택부(120)에 의해 선택된, 어느 하나의 분산 딥러닝 작업과 동시 수행될 분산 딥러닝 작업을 의미할 수 있다. 실시예에 따라, 동시에 수행되는 적어도 두 개의 분산 딥러닝 작업들은 하나의 GPU 상에서 수행될 수 있다. 따라서, 조정 장치(100)는 적어도 두 개의 분산 딥러닝 작업들을 동시에 수행하기 위한 GPU를 포함하는 것으로 이해될 수 있다.

[0053] 저장부(140)에는 조정 장치(100)의 동작을 위해 필요한 OS(Operating System), 어플리케이션, 앱, 이와 관련된 프로그램 등이 저장되어 있을 수 있다. 또한, 저장부(140)에는 학습될 복수의 분산 딥러닝 작업들(신경망 모델 등과 같은 학습 대상 모델과 학습 데이터를 포함하는 개념으로 이해될 수 있음), 프로파일링부(110)에 의한 프로파일링 결과, 작업 선택부(120)에 의한 선택의 결과 내지 선택이 과정에서 일시적으로 또는 비일시적으로 생성되는 데이터, 학습부(130)에 의한 학습 결과(예컨대, 학습된 모델 등)가 저장될 수 있다.

[0054] 도 6에 도시된 조정 장치(100)의 구성들 각각은 기능 및 논리적으로 분리될 수 있음으로 나타내는 것이며, 반드시 각각의 구성이 별도의 물리적 장치로 구분되거나 별도의 코드로 작성됨을 의미하는 것이 아님을 본 발명의 기술분야의 평균적 전문가는 용이하게 추론할 수 있을 것이다.

[0055] 또한, 본 명세서에서 "부", "모듈", 및 "유닛"이라 함은, 본 발명의 기술적 사상을 수행하기 위한 하드웨어 및 상기 하드웨어를 구동하기 위한 소프트웨어의 기능적, 구조적 결합을 의미할 수 있다. 예컨대, 상기 모듈은 소정의 코드와 상기 소정의 코드가 수행되기 위한 하드웨어 리소스의 논리적인 단위를 의미할 수 있으며, 반드시 물리적으로 연결된 코드를 의미하거나, 한 종류의 하드웨어를 의미하는 것이 아니다.

[0056] 도 7은 본 발명의 일 실시예에 따른 분산 딥러닝 작업 실행 순서 조정 방법을 설명하기 위한 흐름도이다. 조정 방법 등으로 명명될 수도 있는 작업 실행 순서 조정 방법은 도 6을 통해 설명한 컴퓨팅 장치(또는 조정 장치)에 의해 수행될 수 있다. 실시예에 따라, 조정 방법을 이루는 단계들 중 적어도 일부는 컴퓨팅 장치의 프로세서에 의해 수행되는 것으로 이해될 수 있다. 이하, 조정 방법을 설명함에 있어 앞선 기재와 중복되는 내용에 관하여는 그 구체적인 설명을 생략하기로 한다.

[0057] 우선, 복수의 분산 딥러닝 작업들에 대한 프로파일링이 수행된다(S110). 프로파일링을 통해, 복수의 분산 딥러닝 작업들 각각에 대한 활성 구간, 유휴 구간, GPU 메모리 활용률에 관한 정보가 추출될 수 있다.

[0058] 다음으로, 동시 수행될 적어도 두 개의 분산 딥러닝 작업들이 선택된다(S120). 적어도 두 개의 분산 딥러닝 작업들은 임의의 방법으로 선택된 어느 하나의 분산 딥러닝 작업(예컨대, 작업 대기열의 첫 번째 분산 딥러닝 작업) 및 어느 하나의 분산 딥러닝 작업과 동시에 수행되기에 가장 적합한 다른 하나의 분산 딥러닝 작업을 포함한다. 다른 하나의 분산 딥러닝 작업은 GPU 메모리 사용율 조건과 활성/유휴 조건을 만족하여야 한다.

[0059] 선택된 적어도 두 개의 분산 딥러닝 작업들에 대한 학습 동작이 수행될 수 있다(S130). 동시 작업의 결과로써, 각각이 적어도 두 개의 분산 딥러닝 작업들 각각에 대응하는 학습된 모델들이 생성된다.

[0060] 이상에서 설명된 장치는 하드웨어 구성 요소, 소프트웨어 구성 요소, 및/또는 하드웨어 구성 요소 및 소프트웨어 구성 요소의 집합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성 요소는, 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(Operation System, 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 기술 분야에서 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.

[0061] 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, 구성 요소(Component), 물리적 장치, 가상 장치(Virtual Equipment), 컴퓨터 저장 매체 또는 장치, 또는 전송되는 신호 파(Signal Wave)에 영구적으로, 또는 일시적으로 구체화(Embody)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.

[0062] 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), 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. 상기된 하드웨어 장치는 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0063] 본 발명은 도면에 도시된 실시예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시예가 가능하다는 점을 이해할 것이다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성 요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성 요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0064] 100 : 조정 장치

110 : 프로파일링부

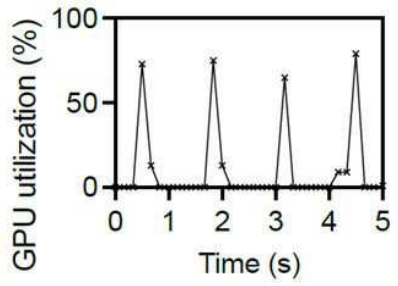
120 : 작업 선택부

130 : 학습부

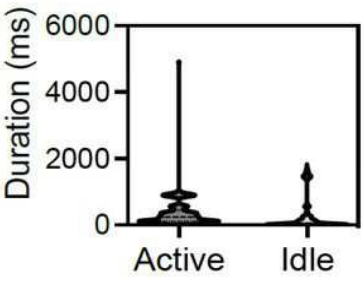
140 : 저장부

도면

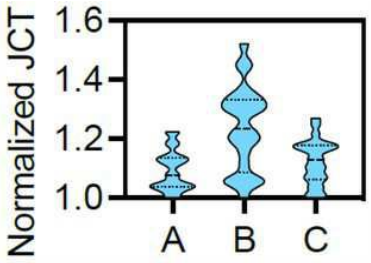
도면1



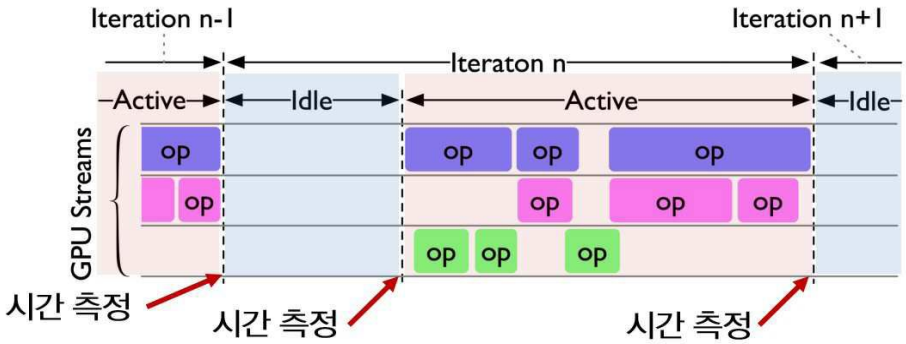
도면2



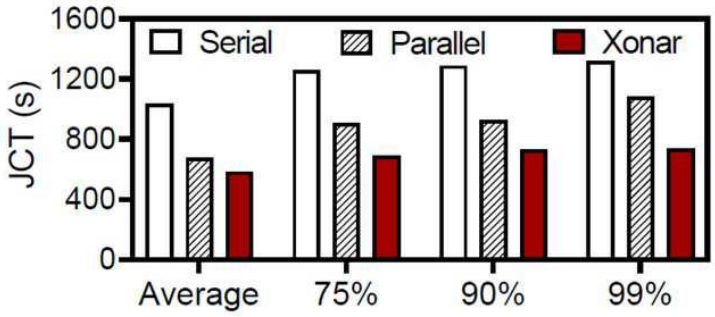
도면3



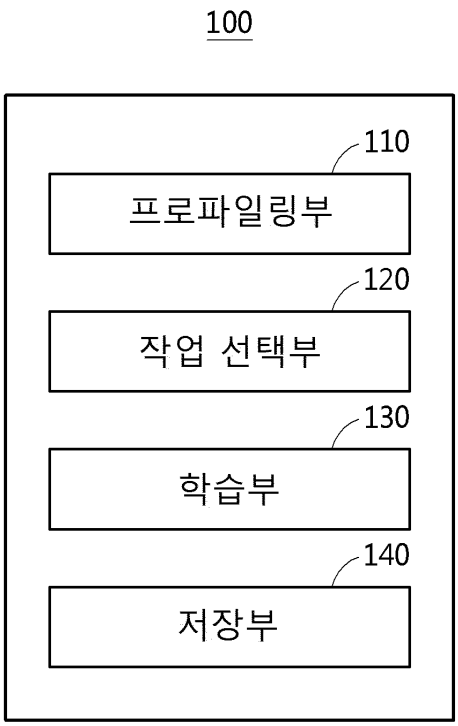
도면4



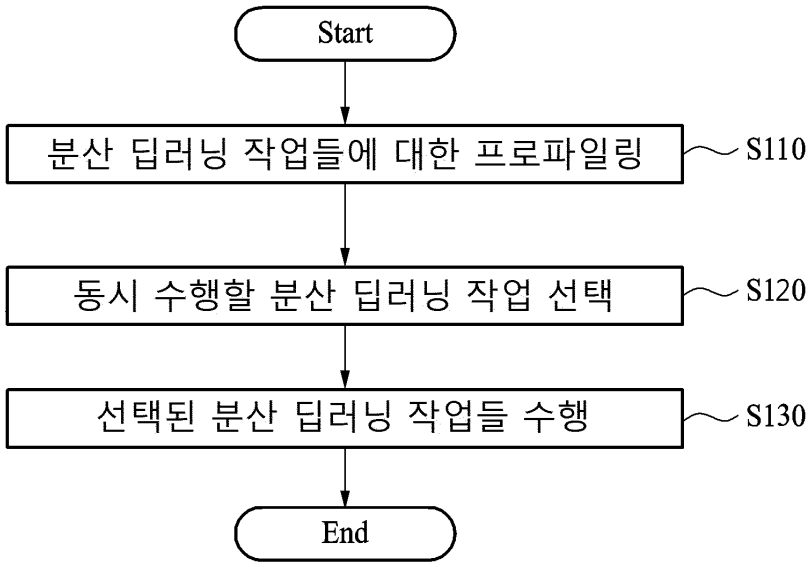
도면5


1600
1200
800
400
0
JCT (s)
Serial
Parallel
Xonar
Average
75%
90%
99%

도면6


100
110
프로파일링부
120
작업 선택부
130
학습부
140
저장부

도면7


Start
분산 딥러닝 작업들에 대한 프로파일링
S110
동시 수행할 분산 딥러닝 작업 선택
S120
선택된 분산 딥러닝 작업들 수행
S130
End