



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0105039
(43) 공개일자 2023년07월11일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) G06N 3/04 (2023.01)
(52) CPC특허분류
G06N 3/084 (2023.01)
G06N 3/045 (2023.01)
(21) 출원번호 10-2022-0000138
(22) 출원일자 2022년01월03일
심사청구일자 2022년01월03일

(71) 출원인
창원대학교 산학협력단
경상남도 창원시 의창구 창원대학교 20(퇴촌동)
(72) 발명자
강동현
경상남도 창원시 성산구 원이대로 495, 211동
1303호(반림동, 트리비아아파트)
최진서
경상남도 김해시 금관대로1183번길 3-15(외동)
(74) 대리인
양성보

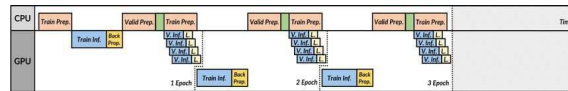
전체 청구항 수 : 총 15 항

(54) 발명의 명칭 병렬 처리 기반 훈련 및 검증 성능 향상 장치 및 방법

(57) 요약

병렬 처리 기반 훈련 및 검증 성능 향상 장치 및 방법이 개시된다. 일 실시예에 따른 훈련 및 검증 성능 향상 장치에 의해 수행되는 병렬 처리 방법은, 딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행하는 단계; 및 상기 모델 복사 과정을 수행함에 따라 중첩된 훈련 쓰레드와 검증 쓰레드를 통해 상기 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리하는 단계를 포함할 수 있다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호	1711142833
과제번호	NRF-2019R1G1A1099932
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	개인기초연구(과기정통부)(R&D) - 생애첫연구
연구과제명	초고성능 스토리지 장치를 위한 직접 접근 I/O 플랫폼 연구
기 여 율	1/1
과제수행기관명	창원대학교
연구기간	2021.03.01 ~ 2022.02.28

명세서

청구범위

청구항 1

훈련 및 검증 성능 향상 장치에 의해 수행되는 병렬 처리 방법에 있어서,

딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행하는 단계; 및

상기 모델 복사 과정을 수행함에 따라 중첩된 훈련 쓰레드와 검증 쓰레드를 통해 상기 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리하는 단계

를 포함하는 병렬 처리 방법.

청구항 2

제1항에 있어서,

상기 모델 복사 과정을 수행하는 단계는,

상기 딥러닝 모델의 훈련 과정을 수행하는 훈련 쓰레드와 상기 딥러닝 모델의 검증 과정을 수행하는 검증 쓰레드로 구분하는 단계

를 포함하는 병렬 처리 방법.

청구항 3

제1항에 있어서,

상기 모델 복사(Model Copy) 과정은,

상기 딥러닝 모델의 훈련 과정이 완료된 훈련 쓰레드의 최종 가중치 값들을 상기 검증 쓰레드로 복사함으로써 훈련 쓰레드와 검증 쓰레드가 각각 독립적으로 동작할 수 있도록 하는

것을 특징으로 하는 병렬 처리 방법.

청구항 4

제2항에 있어서,

상기 모델 복사 과정을 수행하는 단계는,

제1 에폭에 대한 훈련 쓰레드가 완료된 후, 모델 복사 과정을 통해 검증 쓰레드가 동작을 시작함과 동시에 제2 에폭의 훈련을 위한 데이터 전처리 과정을 실행하는 단계

를 포함하는 병렬 처리 방법.

청구항 5

제2항에 있어서,

상기 딥러닝 모델의 훈련 과정은,

각 에폭마다 데이터 전처리 과정, 추론을 위한 연산 및 오차 역전파 과정을 포함하고,

상기 데이터 전처리 과정은,

원격 또는 로컬 저장소의 기본 데이터(Raw Data)를 메인 메모리로 로드(Load)하고, 상기 메인 메모리로 로드된 데이터를 텐서 형태로 변환하고, 딥러닝 모델의 편향(Bias)을 감소시키기 위해 텐서 형태의 데이터를 무작위로 셔플(Shuffle)하고, 상기 셔플된 텐서를 GPU 연산을 위한 배치(Batch) 크기로 분할하는 동작을 수행하는 것을

특징으로 하는 병렬 처리 방법.

청구항 6

제1항에 있어서,
상기 병렬 처리하는 단계는,
멀티 쓰레드(Multi Thread) 기반으로 딥러닝 모델을 검증하는 단계
를 포함하는 병렬 처리 방법.

청구항 7

제6항에 있어서,
상기 병렬 처리하는 단계는,
훈련 쓰레드가 완료된 후 모델 복사 과정이 수행될 때, 쓰레드 풀에서 사용자로부터 설정된 개수에 따라 딥러닝
모델의 검증을 위한 검증 쓰레드를 생성하는 단계
를 포함하는 병렬 처리 방법.

청구항 8

제7항에 있어서,
상기 병렬 처리하는 단계는,
상기 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를
분할하는 단계
를 포함하는 병렬 처리 방법.

청구항 9

제8항에 있어서,
상기 병렬 처리하는 단계는,
상기 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 각각에서 추론 연산 후 계산된 오차(Loss) 및 정확도
(Accuracy)에 대한 평균 값을 계산하여 예측의 정확도와 오차 값으로 반영하는 단계
를 포함하는 병렬 처리 방법.

청구항 10

제1항에 있어서,
상기 병렬 처리 방법은,
텐서플로우, 파이토치를 포함하는 딥러닝 기반 소프트웨어 플랫폼에서 딥러닝 모델의 훈련 및 검증을 위해 CPU 연
산과 GPU 연산이 병렬 처리되는 것을 특징으로 하는 병렬 처리 방법.

청구항 11

제1항에 있어서,
상기 병렬 처리 방법은,
상기 딥러닝 모델의 검증 과정을 멀티 쓰레드를 기반으로 수행함에 따라 추론을 위한 데이터 전처리 과정이 CPU
에서 병렬적으로 처리되도록 하고, 각 검증 쓰레드를 통해 추론 연산이 수행되도록 상기 추론을 위한 데이터 전
처리 과정이 수행된 데이터가 GPU 로 전달되는 것을 특징으로 하는 병렬 처리 방법.

청구항 12

제1항 내지 제11항 중 어느 한 항의 병렬 처리 방법을 상기 훈련 및 검증 성능 향상 장치에 실행시키기 위해 비-일시적인 컴퓨터 판독가능한 기록 매체에 저장되는 컴퓨터 프로그램.

청구항 13

훈련 및 검증 성능 향상 장치에 있어서,

딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행하는 모델 복사부; 및

상기 모델 복사 과정을 수행함에 따라 중첩된 훈련 스레드와 검증 스레드를 통해 상기 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리하는 병렬 처리부

를 포함하는 훈련 및 검증 성능 향상 장치.

청구항 14

제13항에 있어서,

상기 모델 복사부는,

상기 딥러닝 모델의 훈련 과정을 수행하는 훈련 스레드와 상기 딥러닝 모델의 검증 과정을 수행하는 검증 스레드로 구분하고, 제1 에폭에 대한 훈련 스레드가 완료된 후, 모델 복사 과정을 통해 검증 스레드가 동작을 시작함과 동시에 제2 에폭의 훈련을 위한 데이터 전처리 과정을 실행하는

것을 특징으로 하는 훈련 및 검증 성능 향상 장치.

청구항 15

제13항에 있어서,

상기 병렬 처리부는,

멀티 스레드(Multi Thread) 기반으로 딥러닝 모델을 검증하고, 훈련 스레드가 완료된 후 모델 복사 과정이 수행될 때, 스레드 풀에서 사용자로부터 설정된 개수에 따라 딥러닝 모델의 검증을 위한 검증 스레드를 생성하고, 상기 생성된 검증 스레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를 분할하고, 상기 딥러닝 모델의 검증을 위해 생성된 검증 스레드의 각각에서 추론 연산 후 계산된 오차(Loss) 및 정확도 (Accuracy)에 대한 평균 값을 계산하여 에폭의 정확도와 오차 값으로 반영하는

것을 특징으로 하는 훈련 및 검증 성능 향상 장치.

발명의 설명

기술 분야

[0001]

아래의 설명은 딥러닝 모델의 훈련 과정과 검증 과정에 대한 병렬 처리 기술에 관한 것이다.

배경 기술

[0003]

딥러닝은 입력 값과 가중치 값을 계산한 결과를 출력하는 기능을 가진 인공 뉴런(Neuron)들이 사람의 뇌 신경망 구조와 유사한 형태의 복잡한 네트워크를 형성한 심층 신경망(Deep Neural Network)을 사용하는 기계학습(Machine Learning)의 방법 중 하나이다. 일반적으로 딥러닝 모델은 훈련(Training) 과정과 검증(Validation) 과정으로 구분할 수 있으며, 빠른 병렬 처리(Parallel Processing)를 위해 GPU 에서 수행된다. 훈련 및 추론 과정에서 CPU 는 기본 데이터(Raw Data)를 일정 크기의 배치(Batch) 데이터로 가공하여 GPU 로 전달한다. 일정 크기로 분할된 배치 데이터는 심층 신경망을 통해 추론 과정을 수행하며 전체 데이터가 모두 훈련 및 검증 과정을 완료할 때 딥러닝에서는 1 에폭(Epoch)이 수행되었다고 말한다. 일반적으로 훈련 및 검증에서의 정확도(Accuracy)를 향상시키기 위해 에폭 과정은 여러 번 반복된다.

[0004] 딥러닝의 훈련 과정에서 데이터는 심층 신경망에 특성(Feature)으로 입력되고 이를 계산한 결과(즉, 추론 결과)와 입력된 정답(Label) 결과를 비교함으로써, 정확도(Accuracy)와 오차(Loss)를 계산한다. 또한, 오차 값을 줄이기 위해 오차 역전파(Error backpropagation)를 수행함으로써, 심층 신경망 내부의 뉴런(Neuron)들의 가중치 값을 조정한다. 반면, 검증 과정에서는 훈련 과정과 동일하게 입력 데이터를 처리하고 추론 결과를 계산하지만 오차 역전파를 수행하지 않으며, 일반적으로 학습 데이터의 10% 정도의 양을 가진 별도의 데이터를 사용한다. 즉, 딥러닝 모델의 전반적인 성능은 검증 과정 보다 9 배 많은 양의 데이터를 처리하고 오차 역전파를 통해 더 많은 연산을 수행하는 훈련 과정의 수행 시간이 절대적인 영향을 미친다.

발명의 내용

해결하려는 과제

[0006] 멀티 쓰레드(Multi Thread) 기반으로 딥러닝 모델의 훈련 및 검증 과정을 효율적으로 수행하기 위한 방법 및 장치를 제공할 수 있다.

[0007] 하드웨어 자원을 효율적으로 사용하기 위한 멀티 쓰레드 기반의 병렬 처리 방법 및 장치를 제공할 수 있다.

과제의 해결 수단

[0009] 훈련 및 검증 성능 향상 장치에 의해 수행되는 병렬 처리 방법은, 딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행하는 단계; 및 상기 모델 복사 과정을 수행함에 따라 중첩된 훈련 쓰레드와 검증 쓰레드를 통해 상기 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리하는 단계를 포함할 수 있다.

[0010] 상기 모델 복사 과정을 수행하는 단계는, 상기 딥러닝 모델의 훈련 과정을 수행하는 훈련 쓰레드와 상기 딥러닝 모델의 검증 과정을 수행하는 검증 쓰레드로 구분하는 단계를 포함할 수 있다.

[0011] 상기 모델 복사(Model Copy) 과정은, 상기 딥러닝 모델의 훈련 과정이 완료된 훈련 쓰레드의 최종 가중치 값을 상기 검증 쓰레드로 복사함으로써 훈련 쓰레드와 검증 쓰레드가 각각 독립적으로 동작할 수 있도록 할 수 있다.

[0012] 상기 모델 복사 과정을 수행하는 단계는, 제1 에폭에 대한 훈련 쓰레드가 완료된 후, 모델 복사 과정을 통해 검증 쓰레드가 동작을 시작함과 동시에 제2 에폭의 훈련을 위한 데이터 전처리 과정을 실행하는 단계를 포함할 수 있다.

[0013] 상기 딥러닝 모델의 훈련 과정은, 각 에폭마다 데이터 전처리 과정, 추론을 위한 연산 및 오차 역전파 과정을 포함하고, 상기 데이터 전처리 과정은, 원격 또는 로컬 저장소의 기본 데이터(Raw Data)를 메인 메모리로 로드(Load)하고, 상기 메인 메모리로 로드된 데이터를 텐서 형태로 변환하고, 딥러닝 모델의 편향(Bias)을 감소시키기 위해 텐서 형태의 데이터를 무작위로 셔플(Shuffle)하고, 상기 셔플된 텐서를 GPU 연산을 위한 배치(Batch) 크기로 분할하는 동작을 수행할 수 있다.

[0014] 상기 병렬 처리하는 단계는, 멀티 쓰레드(Multi Thread) 기반으로 딥러닝 모델을 검증하는 단계를 포함할 수 있다.

[0015] 상기 병렬 처리하는 단계는, 훈련 쓰레드가 완료된 후 모델 복사 과정이 수행될 때, 쓰레드 풀에서 사용자로부터 설정된 개수에 따라 딥러닝 모델의 검증을 위한 검증 쓰레드를 생성하는 단계를 포함할 수 있다.

[0016] 상기 병렬 처리하는 단계는, 상기 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를 분할하는 단계를 포함할 수 있다.

[0017] 상기 병렬 처리하는 단계는, 상기 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 각각에서 추론 연산 후 계산된 오차(Loss) 및 정확도 (Accuracy)에 대한 평균 값을 계산하여 에폭의 정확도와 오차 값으로 반영하는 단계를 포함할 수 있다.

[0018] 상기 병렬 처리 방법은, 텐서플로우, 파이토치를 포함하는 딥러닝 기반 소프트웨어 플랫폼에서 딥러닝 모델의

훈련 및 검증을 위해 CPU 연산과 GPU 연산이 병렬 처리될 수 있다.

- [0019] 상기 병렬 처리 방법은, 상기 딥러닝 모델의 검증 과정을 멀티 쓰레드를 기반으로 수행함에 따라 추론을 위한 데이터 전처리 과정이 CPU 에서 병렬적으로 처리되도록 하고, 각 검증 쓰레드를 통해 추론 연산이 수행되도록 상기 추론을 위한 데이터 전처리 과정이 수행된 데이터가 GPU 로 전달될 수 있다.
- [0020] 병렬 처리 방법을 상기 훈련 및 검증 성능 향상 장치에 실행시키기 위해 비-일시적인 컴퓨터 판독가능한 기록 매체에 저장되는 컴퓨터 프로그램을 포함할 수 있다.
- [0021] 훈련 및 검증 성능 향상 장치는, 딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행하는 모델 복사부; 및 상기 모델 복사 과정을 수행함에 따라 중첩된 훈련 쓰레드와 검증 쓰레드를 통해 상기 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리하는 병렬 처리부를 포함할 수 있다.
- [0022] 상기 모델 복사부는, 상기 딥러닝 모델의 훈련 과정을 수행하는 훈련 쓰레드와 상기 딥러닝 모델의 검증 과정을 수행하는 검증 쓰레드로 구분하고, 제1 에폭에 대한 훈련 쓰레드가 완료된 후, 모델 복사 과정을 통해 검증 쓰레드가 동작을 시작함과 동시에 제2 에폭의 훈련을 위한 데이터 전처리 과정을 실행할 수 있다.
- [0023] 상기 병렬 처리부는, 멀티 쓰레드(Multi Thread) 기반으로 딥러닝 모델을 검증하고, 훈련 쓰레드가 완료된 후 모델 복사 과정이 수행될 때, 쓰레드 풀에서 사용자로부터 설정된 개수에 따라 딥러닝 모델의 검증을 위한 검증 쓰레드를 생성하고, 상기 생성된 검증 쓰레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를 분할하고, 상기 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 각각에서 추론 연산 후 계산된 오차(Loss) 및 정확도(Accuracy)에 대한 평균 값을 계산하여 에폭의 정확도와 오차 값으로 반영할 수 있다.

발명의 효과

- [0025] 하드웨어의 리소스를 효율적으로 사용할 수 있다. 다시 말해서, 고사양의 CPU 와 GPU의 사용률을 높이고 전반적인 훈련 및 검증 성능을 향상시킬 수 있다.
- [0026] 훈련 모델 복사(Training Model Copy) 기능을 통해서 GPU 훈련과 검증을 위한 단계를 중첩(Interleaving)시키고 검증 단계를 멀티 쓰레드 기반으로 수행함으로써, 효율적으로 성능을 개선할 수 있다.

도면의 간단한 설명

- [0028] 도 1은 종래의 훈련 및 검증 동작을 설명하기 위한 도면이다.
- 도 2는 일 실시예에 있어서, 훈련 및 검증 동작을 설명하기 위한 도면이다.
- 도 3은 일 실시예에 있어서, 검증 과정을 설명하기 위한 도면이다.
- 도 4는 일 실시예에 따른 훈련 및 검증 성능 향상 장치의 구성을 설명하기 위한 블록도이다.
- 도 5는 일 실시예에 따른 훈련 및 검증 성능 향상 장치에서 병렬 처리 방법을 설명하기 위한 흐름도이다.
- 도 6 및 도 7은 일 실시예에 있어서, 훈련 성능을 비교한 그래프를 나타낸 도면이다.
- 도 8 내지 도 15는 일 실시예에 있어서, CPU와 GPU 사용률을 평가한 그래프를 나타낸 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0029] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.
- [0031] 도 1은 종래의 훈련 및 검증 동작을 설명하기 위한 도면이다.
- [0032] 심층 신경망에서 뉴런들의 기본 입력은 텐서(Tensor)이기 때문에 추론을 위해서는 기본 데이터를 실수(Float) 형태로 구성된 고차원 배열인 텐서(Tensor) 형태로 변경해야만 한다. 이러한 변환 과정을 데이터 전처리(Preprocessing) 과정이라고 한다. 일반적으로 CPU 를 통해 데이터 전처리 과정이 진행된다. 데이터 전처리

과정에 대하여 설명하기로 한다.

- [0033] 원격 또는 로컬 저장소의 기본 데이터(Raw Data)를 메인 메모리로 로드(Load)한다(1). 메인 메모리로 로드된 데이터를 텐서 형태로 변환한다(2). 학습 모델의 편향(Bias)을 감소시키기 위해 텐서 형태의 데이터를 무작위로 셔플(Shuffle)한다(3). 셔플된 텐서를 GPU 연산을 위한 배치(Batch) 크기로 분할한다(4).
- [0034] 전처리 과정 후, 셔플된 텐서는 배치 단위로 GPU 에 복사되어 훈련 및 검증이 진행되고 분할된 모든 배치에 대한 훈련 및 검증이 완료되면 에폭을 증가시킴으로써 다음 훈련 및 검증을 진행한다. 이러한, 일련의 과정을 데이터 파이프라인(Data Pipeline)이라고 한다. 이러한 데이터 파이프라인은 Numpy 오픈소스 라이브러리를 활용하여 구현이 가능하다. 그러나, Numpy 는 단일 스레드 기반의 데이터 파이프라인만을 지원하기 때문에 멀티 스레드 형태로 데이터 파이프라인을 구성하기 위해서는 케라스(Keras)와 tf.data를 사용해야 한다. 그러나, tf.data 역시 전처리 과정만을 멀티 스레드 형태로 수행하기 때문에 전반적인 모델의 훈련 및 검증 과정의 성능을 향상시키기 위해서는 한계를 가진다.
- [0035] 도 1은 tf.data 기반 딥러닝 모델의 GPU 사용률을 나타낸 것으로, 실험에는 기본적인 MNIST 데이터가 사용될 수 있다. 도 1 에서 전체 훈련 및 검증을 수행하는 동안 GPU 의 사용률이 부분적으로 감소하며, 동일한 패턴이 반복되는 것을 확인할 수 있다. 이는, 고사양의 하드웨어(예: CPU, GPU, 등) 자원(Resource)을 낭비하는 심각한 결과를 초래하기 때문에 개선이 요구된다.
- [0036] 기본적인 훈련 및 검증 기능은 중첩 없이 단일 스레드 기반으로 동작한다(즉, 훈련 완료 후 검증 과정 실행). 이는, 훈련 과정 중 오차 역전파를 통해 추론을 위한 가중치 갱신(Update)이 빈번하게 발생하기 때문이며, 검증 과정에서 서로 다른 가중치 값을 사용하여 추론을 진행하는 경우, Stale Gradient 문제가 발생하기 때문이다. 이에, 일반적으로 최종 가중치 값을 추출하고 이를 기반으로 검증 과정을 진행한다.
- [0037] 도 2는 일 실시예에 있어서, 훈련 및 검증 동작을 설명하기 위한 도면이다.
- [0038] 도 2에서는 하드웨어 자원(CPU, GPU)을 효율적으로 사용하기 위한 멀티 스레드 기반의 병렬 처리 동작에 대하여 설명하기로 한다. 훈련 및 검증 성능 향상 장치는 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 훈련 모델 복사(Model Copy) 과정과 GPU 사용률 개선을 위한 멀티 스레드(Multi-thread) 기반 검증 과정을 수행할 수 있다.
- [0039] 상세하게는, 훈련 및 검증 성능 향상 장치는 훈련 모델 복사 과정을 수행할 수 있다. 훈련 및 검증 성능 향상 장치는 도 1의 단점을 개선하기 위하여, 훈련 스레드(Training-thread)와 검증 스레드(Validation-thread)로 훈련과 검증을 위한 스레드를 구분할 수 있다. 훈련 및 검증 성능 향상 장치는 훈련이 완료된 스레드의 최종 가중치 값들을 검증 스레드로 복사함으로써 두 개의 스레드가 독립적으로 동작할 수 있도록 모델 복사(Model Copy) 과정을 수행할 수 있다.
- [0040] 도 1과 도 2를 비교해보면, Numpy 기반 기존 데이터 파이프라인의 동작과 실시예에서 제안된 기법에 대한 차이점을 보여준다. 도 1에서 기존 Numpy 기반 훈련 및 검증 과정은 모든 과정이 직렬화(Serialization)되어 수행된다(즉, 훈련 완료 후 검증 과정 실행). 반면에, 도 2에서 제안 기법은 첫 번째 에폭에 대한 훈련 스레드가 완료된 후, 모델 복사를 통해 검증 스레드가 동작을 시작하고 동시에 두 번째 에폭의 훈련을 위한 전처리 과정이 실행되는 것을 확인할 수 있다. 즉, 훈련 스레드와 검증 스레드가 동시에 중첩(Interleaving)되어 수행될 수 있다.
- [0041] 모든 에폭에서의 딥러닝 모델의 훈련 과정은 전처리(데이터 로드, 텐서 변환, 셔플, 배치), 추론을 위한 연산, 그리고 오차 역전파 과정을 반복적으로 수행한다. 이러한 에폭의 훈련 과정을 반복적으로 수행해야 하기 때문에 훈련 과정이 빠르게 재시작되는 것은 CPU 와 GPU 의 유휴 시간(Idle Time)을 모두 감소시키고 사용률을 향상시킬 수 있다.
- [0042] 검증 스레드는 모델 복사 과정을 통해 훈련 스레드와 독립적으로 동작할 수 있다. 그러나, 검증 스레드 역시 단일 스레드 기반으로 수행하는 경우, 전처리와 추론을 위한 연산에 대한 일련의 과정을 직렬화하여 수행하게 된다. 이러한 직렬화 과정을 병렬적으로 처리하기 위해 멀티 스레드 기반 딥러닝 모델의 검증 과정이 추가적으로 설계될 수 있다.
- [0043] 훈련 및 검증 성능 향상 장치는 멀티 스레드 기반의 검증 과정을 수행할 수 있다. 훈련 및 검증 성능 향상 장치는 훈련 스레드가 완료된 후 모델 복사가 수행될 때, 스레드 풀에서 사용자로부터 사전에 설정된 개수만큼 딥러닝 모델의 검증을 위한 검증 스레드를 생성할 수 있다. 또한, 훈련 및 검증 성능 향상 장치는 검증 과정에

필요한 데이터의 개수를 미리 확인하여 생성된 검증 쓰레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를 분할할 수 있다. 이는 각각의 검증 쓰레드가 서로 다른 영역의 데이터에 접근하여 전처리 과정을 수행하기 위함이다.

[0044] 도 3을 참고하면, 실시예에서 제안된 병렬 처리 기법의 검증 과정을 설명하기 위한 예이다. 추론 데이터에 대한 전처리 과정(텐서 변환, 배치)과 추론을 위한 연산 과정이 병렬적으로 수행되는 것을 확인할 수 있다. 또한, 모델 복사 과정을 통해 모든 검증 쓰레드가 동일한 가중치 값으로 추론 연산을 수행하기 때문에 Stale Gradient 문제도 발생하지 않는다. 마지막으로, 각 검증 쓰레드에서 추론 연산 후 계산된 오차(Loss) 및 정확도(Accuracy)는 평균 값으로 계산하여 해당 예폭에 대한 정확도와 오차 값으로 반영한다.

[0045] 훈련 및 검증 성능 향상 장치는 멀티 쓰레드 기반으로 검증 과정을 수행함으로써, 추론을 위해 필요한 일련의 데이터 전처리 과정을 CPU가 병렬적으로 처리할 수 있다. 그리고 짧은 대기시간과 함께 각 검증 쓰레드의 추론 연산을 수행할 수 있도록 GPU로 전달한다. 그 결과 CPU의 사용률을 향상시킬 수 있으며, GPU가 전처리 과정으로 대기하는 시간을 효율적으로 감소시킬 수 있다.

[0046] 도 4는 일 실시예에 따른 훈련 및 검증 성능 향상 장치의 구성을 설명하기 위한 블록도이고, 도 5는 일 실시예에 따른 훈련 및 검증 성능 향상 장치에서 병렬 처리 방법을 설명하기 위한 흐름도이다.

[0047] 훈련 및 검증 성능 향상 장치(100)의 프로세서는 모델 복사부(410) 및 병렬 처리부(420)를 포함할 수 있다. 이러한 프로세서의 구성요소들은 훈련 및 검증 성능 향상 장치에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소들은 도 5의 병렬 처리 방법이 포함하는 단계들(510 내지 520)을 수행하도록 훈련 및 검증 성능 향상 장치를 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.

[0048] 프로세서는 병렬 처리 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 훈련 및 검증 성능 향상 장치에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 훈련 및 검증 성능 향상 장치를 제어할 수 있다. 이때, 모델 복사부(410) 및 병렬 처리부(420) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(510 내지 520)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.

[0049] 단계(510)에서 모델 복사부(410)는 딥러닝 모델의 훈련 과정과 검증 과정을 중첩(Interleaving)시키기 위한 모델 복사(Model Copy) 과정을 수행할 수 있다. 모델 복사부(410)는 딥러닝 모델의 훈련 과정을 수행하는 훈련 쓰레드와 딥러닝 모델의 검증 과정을 수행하는 검증 쓰레드로 구분할 수 있다. 이때, 모델 복사 과정이란 딥러닝 모델의 훈련 과정이 완료된 훈련 쓰레드의 최종 가중치 값들을 검증 쓰레드로 복사함으로써 훈련 쓰레드와 검증 쓰레드가 각각 독립적으로 동작할 수 있도록 하는 것을 의미한다. 모델 복사부(410)는 제1 예폭에 대한 훈련 쓰레드가 완료된 후, 모델 복사 과정을 통해 검증 쓰레드가 동작을 시작함과 동시에 제2 예폭의 훈련을 위한 데이터 전처리 과정을 실행할 수 있다.

[0050] 단계(520)에서 병렬 처리부(420)는 모델 복사 과정을 수행함에 따라 중첩된 훈련 쓰레드와 검증 쓰레드를 통해 딥러닝 모델의 훈련 과정과 검증 과정을 병렬 처리할 수 있다. 병렬 처리부(420)는 멀티 쓰레드(Multi Thread) 기반으로 딥러닝 모델을 검증할 수 있다. 병렬 처리부(420)는 훈련 쓰레드가 완료된 후 모델 복사 과정이 수행될 때, 쓰레드 풀에서 사용자로부터 설정된 개수에 따라 딥러닝 모델의 검증을 위한 검증 쓰레드를 생성할 수 있다. 병렬 처리부(420)는 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 개수에 대응되는 동일한 크기로 검증을 위한 데이터를 분할할 수 있다. 병렬 처리부(420)는 딥러닝 모델의 검증을 위해 생성된 검증 쓰레드의 각각에서 추론 연산 후 계산된 오차(Loss) 및 정확도(Accuracy)에 대한 평균 값을 계산하여 예폭의 정확도와 오차 값으로 반영할 수 있다.

[0051] 도 6 및 도 7은 일 실시예에 있어서, 훈련 성능을 비교한 그래프를 나타낸 도면이다.

[0052] 훈련 성능 비교를 위해 실험이 진행될 수 있다. 일례로, Intel Core i5-9500 (3.00GHz, 6 Core) CPU, 32GB DRAM, Nvidia GeForce GTX 1650 GPU (4GB)가 포함된 컴퓨터에서 실험이 진행될 수 있다. 운영체제는 리눅스 커널 4.15 버전의 Ubuntu 16.04 LTS 이 사용될 수 있다. 또한, 제안 기법의 평가를 위해 텐서플로우(버전 2.5.0)와 케라스(버전 2.5.0)을 사용하여 이미지 분류 모델에서 가장 많이 사용되는 합성곱 신경망(CNN: Convolutional Neural Network) 모델이 구현될 수 있다. 훈련 및 검증 과정에 필요한 데이터는 가장 잘 알려진 MNIST 데이터 셋(Data Set)이 사용될 수 있다. 또한, 기본 데이터 셋의 데이터가 복사됨으로써, 데이터 셋의

크기를 1GB 확장하고 이에 대한 평가도 함께 진행될 수 있다. 마지막으로, 전체 에폭은 10으로 설정하고, 모든 실험은 3 번씩 동일한 방식으로 수행하고 평균 값이 최종적으로 표시될 수 있다. 또한, 평가가 진행되는 중에 CPU 와 GPU 사용률을 측정하기 위해 리눅스에서 제공하는 sysstat, gpustat이 사용될 수 있다.

[0053] 전체 훈련 및 검증 시간에 대한 시간이 측정될 수 있고, 또한, 제안 기법과 tf.data 의 경우, 훈련 및 검증 단계에서 사용할 쓰레드의 개수를 지정할 수 있기 때문에 쓰레드의 개수를 1(즉, 단일 쓰레드)과 6(즉, 멀티 쓰레드)으로 변경하며 실험이 진행될 수 있다. 도 6과 도 7은 기본 MNIST 데이터 셋과 1GB 로 확장한 MNIST 데이터 셋의 평가 결과를 나타낸 것이다.

[0054] 도 6(a)는 tf.data 와 제안 기법의 쓰레드의 개수가 1 개일 때, 훈련 및 검증에 대한 전체 시간을 비교한 결과로, Numpy 와 tf.data 파이프라인의 훈련 시간보다 각각 16.6%, 11.6% 감소하는 결과를 확인할 수 있다. 이는, 훈련 모델 복사를 통해 훈련 쓰레드와 검증 쓰레드가 중첩됨으로써 획득될 수 있는 향상으로 추정된다.

[0055] 한편, 도 6(b)는 쓰레드의 개수를 6 개로 설정했을 때 훈련 및 검증 시간에 대한 비교를 나타낸 것이다. 쓰레드 개수가 6으로 설정되는 경우, tf.data 는 데이터를 배치 단위로 분할하는 과정에서 제안 기법은 검증 쓰레드 과정에서 병렬적으로 쓰레드가 동작한다. 제안 기법이 배치 단위의 병렬 처리를 지원하는 tf.data 에 비해 10.8% 전체 훈련 및 검증 시간을 감소시키는 것을 확인할 수 있다. 이는 도 6(a)의 결과에서 보여준 훈련 및 검증 쓰레드의 중첩과 검증 과정에서의 멀티 쓰레드 동작이 전반적인 훈련 및 검증 시간을 단축시킬 수 있다는 사실을 명확하게 보여준다.

[0056] 도 7은 1GB 크기로 MNIST 데이터 셋을 확장한 데이터에 대한 실험 결과를 나타낸 것이다. 전반적인 훈련 및 검증 시간에 대한 성능 패턴은 도 6 과 동일하다는 것을 확인할 수 있다. 데이터 셋을 확장하는 경우, 기본 데이터 셋보다 더 많은 데이터에 대한 전처리 과정이 필요하다. 그 결과, 도 7(b)에서 보여주듯이(즉, 6 개의 쓰레드), tf.data 와 제안 기법의 성능 차이가 최대 22.4% 확대되는 것을 확인할 수 있다. 이러한 결과는 제안 기법이 멀티 쓰레드 기반의 검증 쓰레드가 더 많은 I/O 처리를 진행할 때 효과적이라는 사실을 분명하게 보여준다.

[0057] 도 8 내지 도 15는 일 실시예에 있어서, CPU와 GPU 사용률을 평가한 그래프를 나타낸 도면이다.

[0058] 제안 기법의 전반적인 훈련 및 검증 시간 감소에 대한 성능 이점을 더 명확하게 확인하기 위해, 실시예에서 제안된 기법의 GPU 와 CPU 사용률에 대한 평가가 진행될 수 있다. 도 8 및 도 9는 기본 MNIST 데이터 셋을 입력으로 단일 쓰레드(즉, 쓰레드 개수 1) 환경에서의 측정 결과를 나타낸 것이며, 도 10 및 도 11은 쓰레드의 개수가 6 인 경우 측정 결과를 각각 나타낸 것이다. 평가를 위한 사용률 측정은 mpstat, gpustat 명령어가 사용될 수 있으며, 1 초 간격으로 진행될 수 있다. 또한, 도 12 내지 도 15는 동일한 실험을 1GB MNIST 데이터 셋에 대한 입력으로 수행한 결과를 나타낸 것이다.

[0059] 실험 결과에서 보여주듯이, GPU 의 경우, 사용률이 높은 구간과 낮은 구간이 존재하며 동일한 패턴이 반복되는 것을 확인할 수 있다. 훈련 과정 중 오차 역전파를 통해 가중치 갱신이 일어나는 구간, 검증 구간, 그리고 CPU로부터 다음 학습을 기다리는 구간으로 설명될 수 있다. 또한, 실험 결과에서 확인할 수 있듯이, 제안 기법의 GPU 사용률이 기존 Numpy 와 tf.data 에 비해 더 많은 양의 GPU 을 사용하고 있다는 것을 확인할 수 있다. 즉, 반복되는 패턴에서 GPU 사용률이 감소하는 구간이 기존 기법에 비해 현저히 낮다. 단일 쓰레드 실험에서는 모델 복사를 통한 훈련 및 검증 쓰레드의 중첩의 효과이며, 검증 단계에서 멀티 쓰레드를 사용함으로써 얻을 수 있는 성능 이득이라고 해석된다. 반면, tf.data 의 경우, 쓰레드 개수가 증가함에도 불구하고 GPU 의 사용률 변화가 거의 나타나지 않는다. 이는 tf.data 가 배치 과정에서 수행하는 병렬 처리가 내부적인 병목 현상으로 인해 제한된다는 사실을 보여준다.

[0060] 한편, 제안 기법의 경우, CPU 사용률 역시 다른 기법에 비해 개선되는 것을 확인할 수 있다. 제안 기법의 모델 복사를 통해 CPU 의 데이터 전처리 과정이 다른 기법 대비 빠르게 동작 및 반복되기 때문이다.

[0061] 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, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate 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 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0062] 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(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.

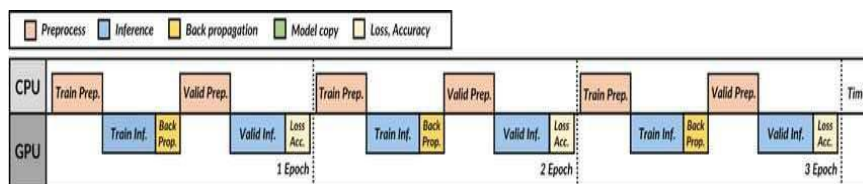
[0063] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic 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.

[0064] 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.

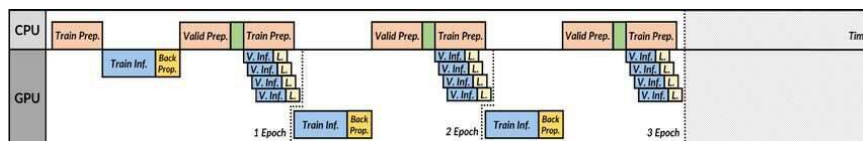
[0065] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

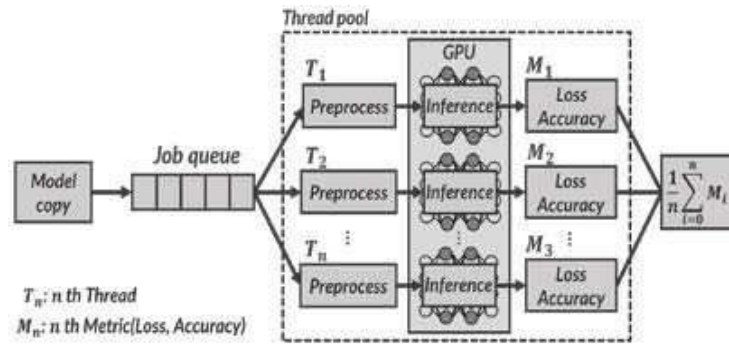
도면1



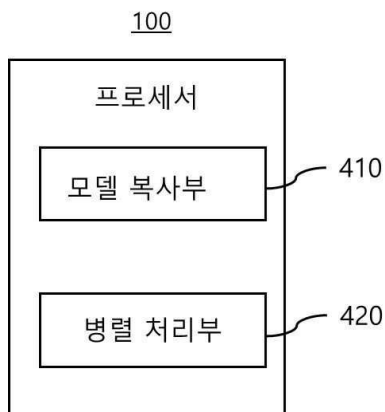
도면2



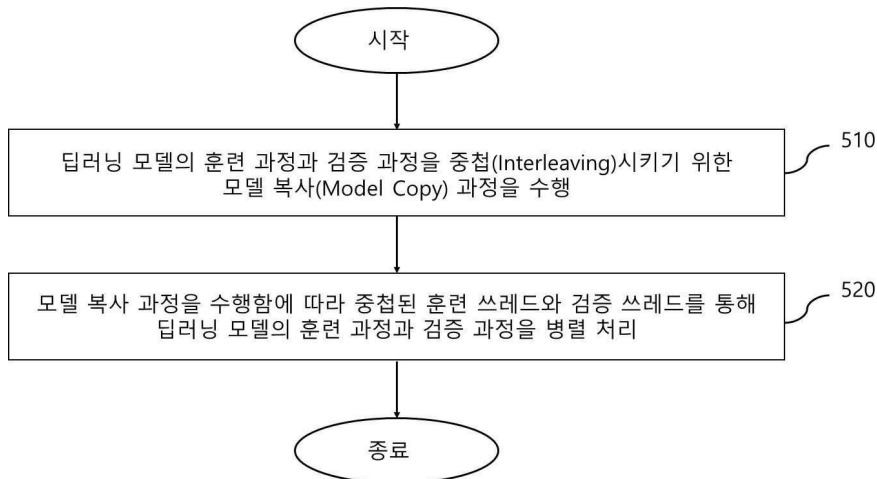
도면3



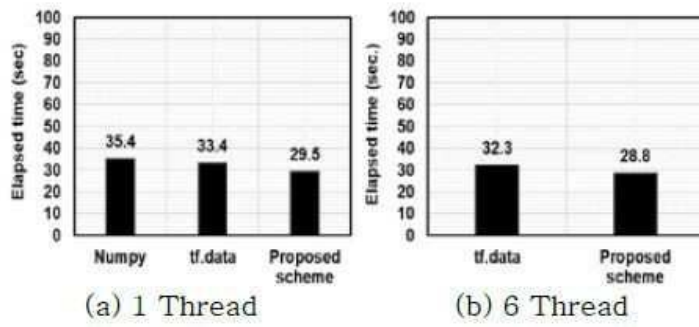
도면4



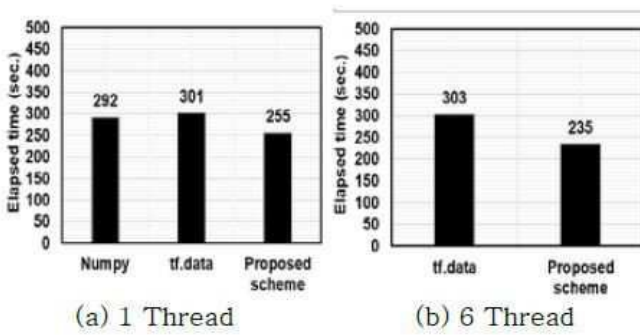
도면5



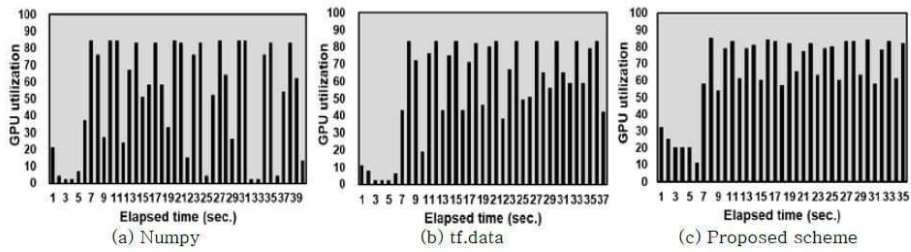
도면6



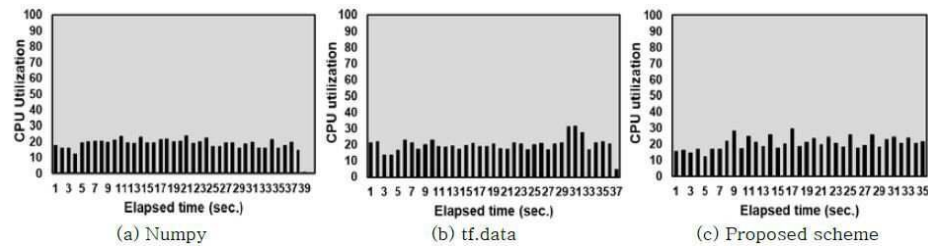
도면7



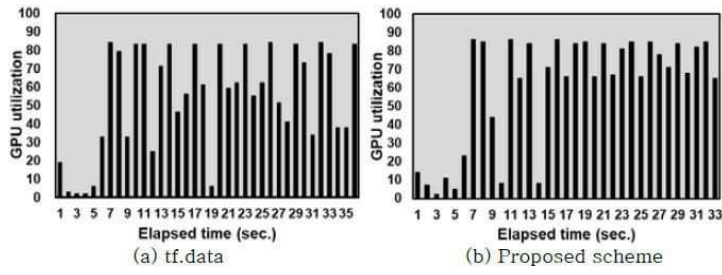
도면8



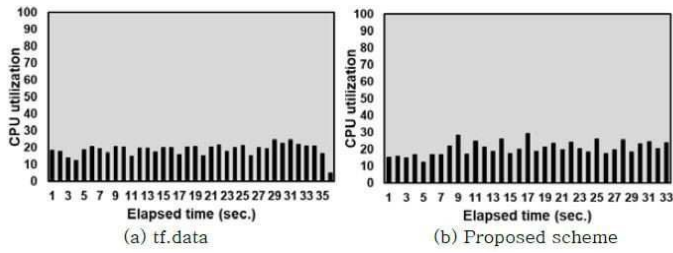
도면9



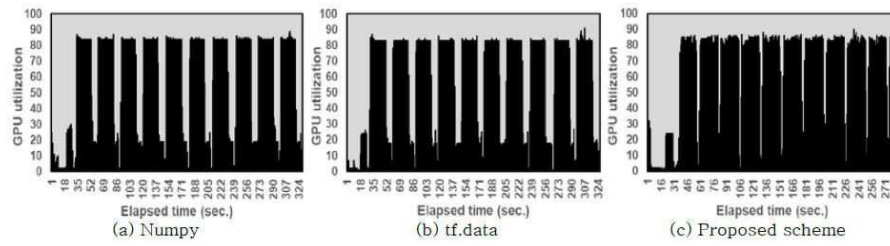
도면10



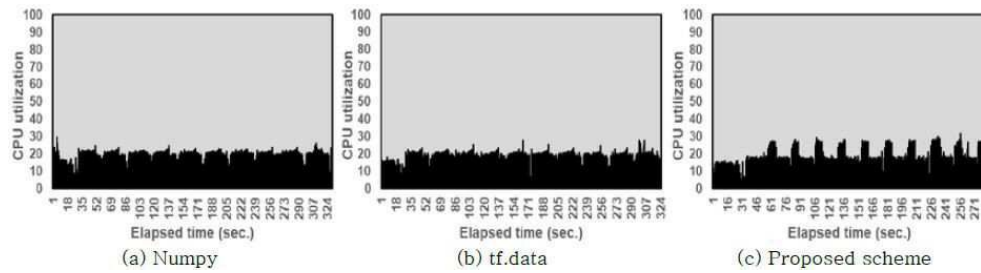
도면11



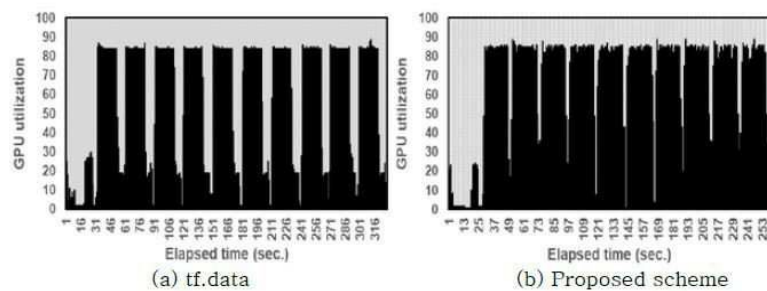
도면12



도면13



도면14



도면15

