



(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년08월31일
(11) 등록번호 10-2296319
(24) 등록일자 2021년08월25일

(51) 국제특허분류(Int. Cl.)
H04L 12/865 (2013.01) G06N 3/04 (2006.01)
H04L 12/803 (2013.01) H04L 12/833 (2013.01)
H04L 29/06 (2006.01)
(52) CPC특허분류
H04L 47/6275 (2013.01)
G06N 3/04 (2013.01)
(21) 출원번호 10-2020-0052711
(22) 출원일자 2020년04월29일
심사청구일자 2020년04월29일
(56) 선행기술조사문헌
KR1020180131836 A*
'Communication Scheduling Optimization for Distributed Deep Learning Systems', proc. of 2018 ICPADS, IEEE, pp.739-746, 2018.*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
강민구
서울특별시 마포구 마포대로11길 50, 402동 1001호
양경식
서울특별시 동대문구 무학로45길 40, 401호(용두동)
유혁
서울특별시 용산구 한강대로 205, A-803호
(74) 대리인
김홍석

전체 청구항 수 : 총 6 항

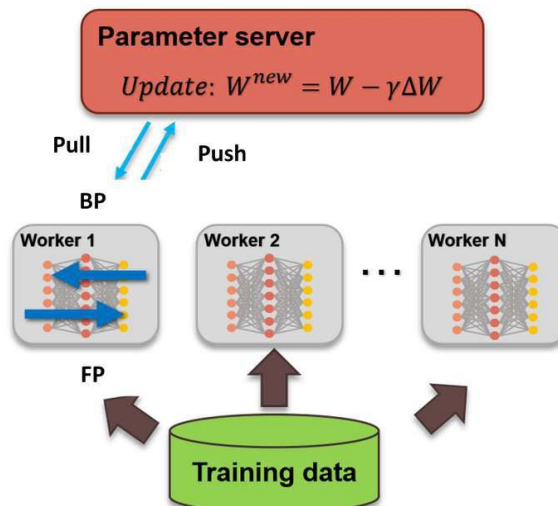
심사관 : 김대성

(54) 발명의 명칭 프로그램머블 스위치에서의 패킷 스케줄링을 이용한 분산 딥러닝 방법

(57) 요약

분산 딥러닝 방법을 제공한다. 분산 딥러닝 방법은 학습 데이터를 복수의 워커 노드들(worker nodes)에서 분산 학습하고, 워커 노드들에서 학습한 데이터의 파라미터(parameter)를 레이어(layer) 단위로 분류하고 분류된 레이어들에 우선순위를 부여하고, 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹하고, 레이어들의 파라미터를 파라미터 서버(parameter server)로 전송하고, 파라미터 서버에서 파라미터를 갱신하며, 갱신된 파라미터를 워커 노드들로 전송하는 단계를 포함하되, 프로그램 가능한 네트워크 스위치(programmable network switch)를 이용하여 텐서 패킷의 헤더를 파싱하여 해당 패킷이 속한 레이어를 파악한다.

대표도 - 도1



(52) CPC특허분류

H04L 47/125 (2013.01)

H04L 47/2458 (2013.01)

H04L 69/22 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711103158
과제번호	2015-0-00280-006
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	SW컴퓨팅산업원천기술개발(R&D)
연구과제명	(SW 스타랩) 성능 및 보안 SLA 보장이 가능한 차세대 클라우드 인프라SW 개발
기 여 율	9/10
과제수행기관명	고려대학교산학협력단
연구기간	2020.01.01 ~ 2020.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711099584
과제번호	2019H1D8A2105513
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	이공계전문기술인력양성(R&D)
연구과제명	DNA+ - AI 서비스를 위한 데이터 프라이버시 및 네트워크 가상화 기술 연구 사업단
기 여 율	1/10
과제수행기관명	고려대학교 산학협력단
연구기간	2020.03.01 ~ 2020.12.31

명세서

청구범위

청구항 1

학습 데이터를 복수의 워커 노드들(worker nodes)에서 분산 학습하는 단계;

상기 워커 노드들에서 학습한 데이터의 파라미터(parameter)를 레이어(layer) 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 단계;

상기 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹하는 단계;

상기 레이어들의 파라미터를 파라미터 서버(parameter server)로 전송하는 단계;

상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및

상기 갱신된 파라미터를 상기 워커 노드들로 전송하는 단계를 포함하되,

프로그램 가능한 네트워크 스위치(programmable network switch)를 이용하여 상기 텐서 패킷의 헤더를 파싱하여 해당 패킷이 속한 레이어를 파악하는 분산 딥러닝 방법.

청구항 2

제1항에 있어서,

상기 레이어들의 우선순위(priority)를 매핑하는 단계; 및

새로운 레이어의 패킷이 도착한 경우, 상기 새로운 레이어가 상기 레이어들보다 낮으면 상기 새로운 레이어를 가장 높은 우선순위 큐에 매핑하고, 나머지 매핑을 한 칸씩 변경하는 단계를 더 포함하는 분산 딥러닝 방법.

청구항 3

제1항에 있어서,

상기 레이어들의 우선순위를 매핑하는 단계; 및

상기 우선순위 중에 가장 높은 우선순위 큐의 패킷을 K-바이트(bite) 처리하는 단계를 더 포함하는 분산 딥러닝 방법.

청구항 4

제3항에 있어서,

상기 레이어들보다 낮은 새로운 레이어가 도착하면,

상기 매핑을 변경하여 빈 상태의 가장 높은 우선순위 큐에 상기 새로운 레이어의 패킷을 처리하는 단계를 더 포함하는 분산 딥러닝 방법.

청구항 5

학습 데이터를 복수의 워커 노드들에서 분산 학습하는 단계;

상기 워커 노드들에서 학습한 데이터의 파라미터를 레이어 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 패킷 스케줄링하는 단계;

상기 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹하는 단계;
 상기 레이어들의 우선순위를 매핑하는 단계;
 상기 레이어들보다 낮은 새로운 레이어가 도착하면, 상기 새로운 레이어를 가장 높은 우선순위 큐에 매핑하고, 나머지 맵핑을 한 칸씩 변경하는 단계;
 상기 레이어들의 파라미터를 파라미터 서버로 전송하는 단계;
 상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및
 상기 갱신된 파라미터를 상기 워커 노드들로 전송하는 단계를 포함하는 분산 딥러닝 방법.

청구항 6

삭제

청구항 7

학습 데이터를 복수의 워커 노드들에서 분산 학습하는 단계;
 상기 워커 노드들에서 학습한 데이터의 파라미터를 레이어 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 패킷 스케줄링하는 단계;
 상기 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹하는 단계;
 상기 레이어들의 우선순위를 매핑하는 단계;
 상기 우선순위 중에 가장 높은 우선순위 큐의 패킷을 K-바이트로 처리하는 단계;
 상기 레이어들보다 낮은 새로운 레이어가 도착하면, 상기 맵핑을 변경하여 빈 상태의 가장 높은 우선순위 큐에 상기 새로운 레이어의 패킷을 처리하는 단계;
 상기 레이어들의 파라미터를 파라미터 서버로 전송하는 단계;
 상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및
 상기 갱신된 파라미터를 상기 워커 노드들로 전송하는 단계를 포함하는 분산 딥러닝 방법.

청구항 8

삭제

발명의 설명

기술 분야

[0001] 본 발명은 딥러닝에 관한 것으로, 보다 상세하게는 프로그래머블 스위치에서의 패킷 스케줄링을 이용한 분산 딥러닝(distributed deep learning) 방법에 관한 것이다.

배경 기술

[0002] 모델의 크기와 학습 데이터의 규모의 증가로 인해 최첨단 딥러닝(deep learning)은 분산되어 수행되고 있다. 이를 분산 딥러닝(distributed deep learning, DDL)이라고 하는데, 여러 개의 GPU 노드(워커)에서 학습 데이터를 나누어 학습하고 학습한 모델 파라미터(모델의 가중치 등)를 갱신하는 데이터 병렬 방식이 대중적으로 사용된다. 데이터 병렬 DDL이 동기화하는 방식에는 파라미터 서버(parameter server, PS) 구조와 all-reduce 구조로 크게 두 가지가 있다. PS 구조는 모델의 파라미터를 관리하는 PS가 독립적으로 존재해 각각의 워커들이 학습한 모델 파라미터를 PS와 주고 받으면서 파라미터를 갱신한다. all-reduce 방식은 PS 없이 각각의 워커가 매 학습마다 모델 파라미터를 모든 워커에게 전송하며 갱신하는 방식이다.

[0003] 갱신 방식에는 크게 동기적 또는 비동기적 갱신 방식이 있다. 모든 워커가 학습이 종료될 때까지 기다린 후 모델 파라미터를 취합해 갱신하고 다음 학습을 진행하는 것이 동기적 갱신 방식이다. 반대로 각각의 워커의 학습

종료마다 모델 파라미터를 주고 받은 후 개별적으로 다음 학습을 진행하는 것이 비동기적 갱신 방식이다.

[0004] 기존 네트워크 스위치는 구매 전에 동작하는 프로토콜 및 기능 등이 고정되어 있다. 따라서 사용자의 필요에 따라 새로운 기능을 구현할 수 없다. 하지만 최근 스위치를 프로그래밍하는 언어인 P4의 등장으로 인해 스위치의 프로그램가능성(programmability)이 주목받고 있다. 프로그래머블 스위치(programmable switch)는 P4로 작성한 프로그램대로 동작할 수 있는 스위치로, 프로그래머블 스위치를 이용하면 사용자가 원하는 방식으로 패킷을 처리할 수 있다. 기존 네트워크 스위치에서는 임의의 패킷 처리가 불가능했으나, 프로그래머블 스위치에서는 하드웨어가 지원하는 기능들을 이용해 사용자가 원하는 기능을 P4 프로그램의 형태로 구현할 수 있다. 본 발명에서는 분산 딥러닝을 위한 패킷 스케줄링 기법을 스위치에 구현하기 위해 P4 언어와 프로그래머블 스위치를 이용하였다.

[0005] 한편, 각각의 워커에서 학습 연산 후 파라미터 통신을 하게 되므로 학습 수행 시간은 연산 시간과 통신 시간의 합이 된다. GPU 성능 개선으로 인해 연산 시간이 줄게 되면서 분산 딥러닝 학습 시간에 통신 시간이 큰 병목이 되는 것이 알려져 있다.

[0006] 국내공개특허 제10-2016-0149422에는 고속 패킷 처리 시스템에 개시되어 있으나, 학습 수행 시간을 감소시키기 위한 연구는 지속적으로 진행되고 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 대한민국 공개특허 제10-2016-0149422호

발명의 내용

해결하려는 과제

[0008] 본원 발명이 해결하고자 하는 과제는 통신 시간을 감소시키는 분산 딥러닝을 위한 프로그래머블 스위치에서의 패킷 스케줄링 방법 및 시스템을 제공하는 것이다.

과제의 해결 수단

[0009] 해결하고자 하는 과제를 달성하기 위하여 본 발명의 실시예들에 따른 분산 딥러닝 방법은, 학습 데이터를 복수의 워커 노드들(worker nodes)에서 분산 학습하는 단계; 상기 워커 노드들에서 학습한 데이터의 파라미터(parameter)를 레이어(layer) 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 단계; 상기 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹하는 단계; 상기 레이어들의 파라미터를 파라미터 서버(parameter server)로 전송하는 단계; 상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및 상기 갱신된 파라미터를 상기 워커 노드들로 전송하는 단계를 포함하되, 프로그램 가능한 네트워크 스위치(programmable network switch)를 이용하여 상기 텐서 패킷의 헤더를 파싱하여 해당 패킷이 속한 레이어를 파악한다.

[0010] 다른 실시예에 따른 분산 딥러닝 방법은, 학습 데이터를 복수의 워커 노드들에서 분산 학습하는 단계; 상기 워커 노드들에서 학습한 데이터의 파라미터를 레이어 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 패킷 스케줄링하는 단계; 상기 레이어들의 우선순위를 매핑하는 단계; 상기 레이어들보다 낮은 새로운 레이어가 도착하면, 상기 새로운 레이어를 가장 높은 우선순위 큐에 매핑하고, 나머지 매핑을 한 칸씩 변경하는 단계; 상기 레이어들의 파라미터를 파라미터 서버로 전송하는 단계; 상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및 상기 갱신된 파라미터를 상기 워커 노드들로 전송하는 단계를 포함한다.

[0011] 또 다른 실시예에 따른 분산 딥러닝 방법은, 학습 데이터를 복수의 워커 노드들에서 분산 학습하는 단계; 상기 워커 노드들에서 학습한 데이터의 파라미터를 레이어 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 패킷 스케줄링하는 단계; 상기 레이어들의 우선순위를 매핑하는 단계; 상기 우선순위 중에 가장 높은 우선순위 큐의 패킷을 K-바이트로 처리하는 단계; 상기 레이어들보다 낮은 새로운 레이어가 도착하면, 상기 매핑을 변경하여 빈 상태의 가장 높은 우선순위 큐에 상기 새로운 레이어의 패킷을 처리하는 단계; 상기 레이어들의 파라미터를 파라미터 서버로 전송하는 단계; 상기 파라미터 서버에서 상기 파라미터를 갱신하는 단계; 및 상기 갱신된

파라미터를 상기 워커 노드들로 전송하는 단계를 포함한다.

발명의 효과

[0012] 본 발명의 실시예들에 따른 분산 딥러닝 방법을 이용함으로써, 우선순위가 높은 레이어의 패킷이 더 낮은 통신 지연을 겪고 학습 시간이 개선될 수 있다.

도면의 간단한 설명

[0013] 도 1은 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하는 블록도이다.
 도 2는 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하기 위한 순서도이다.
 도 3은 스케줄링을 설명하기 위한 예시적으로 도면이다.
 도 4는 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하기 위한 모식도이다.
 도 5는 기존의 분산 딥러닝 및 본 발명의 일 실시예에 따른 분산 딥러닝을 이론적으로 동작시킨 경우 각각의 학습 시간을 나타내는 도면이다.
 도 6은 기존의 분산 딥러닝을 실제로 적용한 경우의 네트워크 내부 지연을 나타내는 그래프이다.
 도 7은 본 발명의 일 실시예에 따른 분산 딥러닝을 실제로 적용한 경우의 네트워크 내부 지연을 나타내는 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 발명의 구성 및 효과를 충분히 이해하기 위하여, 첨부한 도면을 참조하여 본 발명의 바람직한 실시예들을 설명한다. 그러나 본 발명은, 이하에서 개시되는 실시예들에 한정되는 것이 아니라, 여러 가지 형태로 구현될 수 있고 다양한 변경을 가할 수 있다.

[0015] 또한, 본 발명의 실시예들에서 사용되는 용어들은 다르게 정의되지 않는 한, 해당 기술 분야에서 통상의 지식을 가진 자에게 통상적으로 알려진 의미로 해석될 수 있다.

[0016] 본 발명의 분산 딥러닝 방법을 설명하기 전에 본 기술에 연관된 용어 및 기술에 대하여 간략하게 정리하기로 한다.

[0017] 기존 네트워크 스위치는 동작하는 프로토콜 및 기능 등이 고정되어 있었으나, 프로그래머블 네트워크 스위치는 프로그래밍하는 언어(예컨대, P4)를 이용하여 사용자가 원하는 방식으로 패킷을 처리할 수 있다. 즉, 프로그래머블 네트워크 스위치를 이용하여 패킷 파싱 및 처리 로직을 구현할 수 있다. 따라서, 본 발명에서는 분산 딥러닝을 위하여 P4 언어를 컴파일하여 프로그래머블 네트워크 스위치에 적용하여 새 로직을 적용할 수 있다.

[0018] 본 발명에서는 네트워크 스위치의 처리 단위인 패킷으로, 프로그래머블 네트워크 스위치 내에서 스케줄링을 적용한다. 이하에서는, 특정 텐서의 파라미터를 담은 패킷을 텐서 패킷(tensor packet)이라 한다.

[0019] 도 1은 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하는 블록도이다.

[0020] 도 1을 참조하면, 학습 데이터를 미니 배치(mini batch)로 쪼개 각각의 워커 노드(worker node)에서 학습한 후, 파라미터 변화량을 파라미터 서버(parameter server)가 취합하고 갱신한 후, 다시 워커 노드들에 새로운 파라미터를 전달한다.

[0021] 상세하게 설명하면, 학습 데이터가 복수의 워커 노드들로 프로그래머블 네트워크 스위치를 이용하여 첫 번째 레이어부터 N번째 레이어까지 전방향 전파(forward propagation, FP) 처리하여 예측값(prediction)을 얻는다. 이어서, 예측값과 실제값의 격차를 계산하는 손실 함수(loss function)를 이용해 N번째 레이어부터 첫 번째 레이어까지 역방향 전파(back propagation, BP)를 수행하며 각각 레이어의 경사(gradient)를 계산해 모델 파라미터를 조정한다.

[0022] 각각의 워커 노드는 경사값을 파라미터 서버로 전달한다(push). 파라미터 서버는 워커들로부터 취합한 경사값을 이용해 파라미터를 갱신한 후 워커들에게 갱신한 파라미터를 전달한다(pull).

[0023] 즉, 분산 딥러닝은 FP, BP, push, 및 pull의 순서대로 반복하며 수행될 수 있다.

- [0024] 도 2는 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하기 위한 순서도이다.
- [0025] 학습 데이터를 복수의 워커 노드들에서 분산 학습한다(S110).
- [0026] 각각의 워커 노드에서 학습한 데이터의 파라미터를 레이어 단위로 분류하고 분류된 레이어들에 우선순위를 부여하는 스케줄링을 수행한다(S120).
- [0027] 모델의 파라미터를 주고 받을 때, FP에 필요한 순서에 따라 워커에서 파라미터 전송 순서를 변경한다. 예를 들어 FP는 항상 첫 번째 레이어부터 순서대로 수행되므로, 스케줄링은 첫 번째 레이어의 파라미터를 워커에서 우선적으로 보내, 다음 학습의 FP 시작 시간을 앞당기는 방식이다. 커뮤니케이션 스케줄링을 적용할 경우 FP 시작 시간이 앞당겨지고 FP가 나중에 수행될 레이어의 파라미터가 나중에 전송되기 때문에 연산 시간과 통신시간이 겹쳐지게 되고 이는 전체 학습 시간의 감축을 의미한다. 또한, 스케줄링에서는 모델 파라미터를 레이어에 따라 우선순위를 부여해 전송하는데, 이때 스케줄링 및 전송(push & pull) 단위를 텐서(tensor)라 부른다. 즉, N 번째 레이어(layer N)의 텐서를 텐서 N이라 하고, 스케줄링은 텐서 0을 텐서 1 보다, 텐서 1을 텐서 2보다 먼저 전송한다.
- [0028] 도 3은 스케줄링을 설명하기 위한 예시적으로 도면이다. 도 3을 참조하면, 파라미터를 레이어 단위로 분류하고 레이어들의 우선순위를 부여하여 전송함으로써 약 40%의 학습 시간이 개선된다.
- [0029] 그러나, 네트워크 스위치에서의 스케줄링은 다중 우선순위 큐 기반으로 동작하기 때문에 스케줄링에 한계가 있다. 다중 우선순위 큐 방식은 우선순위에 따라 높은 우선순위에 있는 큐에 속한 패킷부터 순서대로 스위치가 전송하게 된다. 예를 들어, 4개의 우선순위 큐가 존재할 경우 패킷의 우선순위를 4가지 종류로 밖에 줄 수 없다는 의미이다. 일반적으로 스위치는 4개에서 8개의 우선순위 큐를 가지고 있고, 스케줄링해야 할 레이어의 개수는 일반적으로 수십 수백에 달하기 때문에 스위치 내에서 이를 반영해 패킷의 스케줄링 하기 어렵다는 한계가 있다. 이를 극복하기 위하여 다음의 단계들을 수행할 수 있다.
- [0030] 계속해서 도 2를 참조하면, 상기 분류된 레이어들의 정보를 텐서 패킷(tensor packet)의 헤더(header)에 마킹한다(S130). 네트워크 스위치에서의 처리 단위는 패킷이고 패킷들이 고속으로 처리되기 때문에, 네트워크 스위치나 플로우의 상태 등을 관리하기 어렵다. 본 발명의 일 실시예에 따르면, 네트워크 스위치에서 분산 딥러닝을 위한 스케줄링을 적용하기 위해 각 워커 노드에서 텐서를 전송할 때, 각각의 텐서 패킷의 IP 옵션 헤더 필드에 해당 텐서 패킷이 어떤 레이어의 파라미터를 담고 있는지 정보를 추가한다. 따라서, 네트워크 스위치는 텐서 패킷의 헤더를 파싱함으로써 해당 패킷이 어떤 레이어에 속하는지 알 수 있다.
- [0031] 레이어 마킹을 통해 패킷이 어떤 레이어에 속하는지 알 수 있더라도, 네트워크 스위치의 우선순위 큐 개수가 제한이 있어 패킷들이 스케줄링하기 어렵다. 따라서, 본 발명에서는 레이어와 우선순위를 매핑하고(S140), 이를 동적으로 변경한다(S150).
- [0032] 우선, 분산 딥러닝의 특성을 살펴보면, 각 워커 노드는 레이어 기준 N부터 0까지 내림차 순으로 패킷을 전송하게 된다. 레이어 0의 파라미터를 모두 주고받은 후, 다시 0부터 N까지 오름차순으로 패킷을 전송하게 된다. 본 발명에서는 레이어의 순서가 단조증가 및 단조 감소하는 딥러닝의 특징을 이용한다.
- [0033] 도 4는 본 발명의 일 실시예에 따른 분산 딥러닝 방법을 설명하기 위한 모식도이다.
- [0034] 도 4의 T0를 참조하면, 큐가 4개 존재하고 레이어 4, 3, 2, 1의 패킷이 순서대로 들어왔을 때 4, 3, 2, 1을 순서대로 낮은 우선순위 큐부터 높은 우선순위 큐로 매핑한다. 도 4의 T1를 참조하면, 레이어 0의 패킷이 들어오게 되면, 매핑을 하나씩 이동시켜 3, 2, 1, 0의 패킷이 각각 우선순위의 큐로 매핑되도록 변경한 후 4 혹은 그 이상의 패킷이 오게 되면 가장 낮은 우선순위 큐로 전달한다. 이와 같은 방식으로 레이어 N부터 0까지 패킷이 내림차순으로 전송되게 되면 반응적으로 N부터 0까지 매핑을 변경시키며 이동한다. 매핑이 레이어 0에 도달하게 되면, 그 이후는 패킷이 0부터 N까지 오름차순으로 전송되므로 이에 맞춰 매핑도 0부터 N까지 반대 방향으로 변경된다.
- [0035] 그러나, 네트워크 스위치에서 이미 큐 안에 들어가 있는 패킷은 스케줄링이 불가능하다. 예를 들어 T0에서 T1으로 매핑이 변경되기 직전에 레이어 1의 패킷이 높은 우선순위 큐에 여러 개 삽입되어 있다면, 매핑 변경 후에 레이어 0의 패킷은 앞에 삽입된 레이어 1의 패킷이 모두 전송된 후에 전송되게 될 것이다. 따라서, 추가적인 지연이 발생할 수 있다.
- [0036] 도 4의 T2를 참조하면, 새로운 레이어의 패킷은 앞서 스위치에 들어온 패킷들보다 먼저 처리되어야 하기 때문에 높은 우선순위의 큐는 비어있어야 한다. 따라서, 높은 우선순위 큐에 매핑된 레이어의 패킷의 헤더가 K-바이트

만큼 처리된 후 매핑을 한 단계씩 변경하는 방식을 적용한다. 이렇게 레이어 1의 패킷이 높은 우선순위 큐에서 K-바이트만큼 처리된 후 매핑을 한 단계씩 낮추고 높은 우선순위 큐에는 아무 레이어도 매핑하지 않음으로써 높은 우선순위 큐를 비어있게 할 수 있다. 이를 통해 새로운 레이어의 패킷이 스위치에 들어왔을 때, 앞서 들어온 패킷의 영향을 받지 않고 즉각적으로 처리될 수 있다.

[0037] 도 5는 기존의 분산 딥러닝 및 본 발명의 일 실시예에 따른 분산 딥러닝을 이론적으로 동작시킨 경우 각각의 학습 시간을 나타내는 도면이다. 도 5를 참조하면, 본 발명의 분산 딥러닝 방법을 이용함으로써, 우선순위가 높은 레이어의 패킷이 더 낮은 통신 지연을 겪고 학습 시간이 개선됨을 볼 수 있다.

[0038] 도 6은 기존의 분산 딥러닝을 실제로 적용한 경우의 네트워크 내부 지연을 나타내는 그래프이고, 도 7은 본 발명의 일 실시예에 따른 분산 딥러닝을 실제로 적용한 경우의 네트워크 내부 지연을 나타내는 그래프이다.

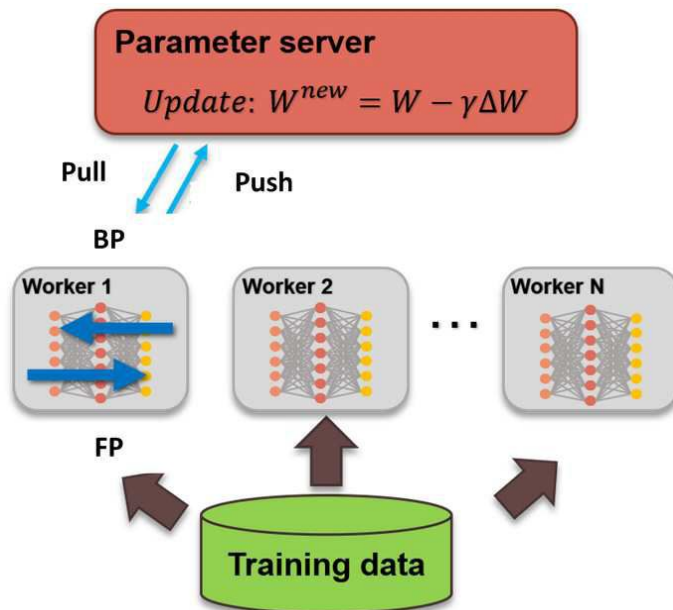
[0039] 네트워크 내부 지연(in-network delay)은 패킷이 호스트에서 전송된 후 목적지에 도달하기까지 네트워크 스위치, 링크 등 통신 장비에 걸리는 지연을 일컫는다.

[0040] 도 6을 참조하면, 기존의 네트워크 스위치에서 혼잡 제어 없이 여러 워커가 통신을 하게 된다면 지연시간이 급격히 증가하는 것을 볼 수 있다. 도 7을 참조하면, 본 발명의 일 실시예에 따른 네트워크 스위치에 적용하면, 우선순위가 낮은 레이어의 패킷들은 여전히 급격히 증가하는 지연을 겪지만, 높은 우선순위의 레이어들은 낮은 지연을 겪는 것을 볼 수 있다. 따라서, 분산 딥러닝에서 각 레이어의 전방향 전파(FP)를 앞당길 수 있다.

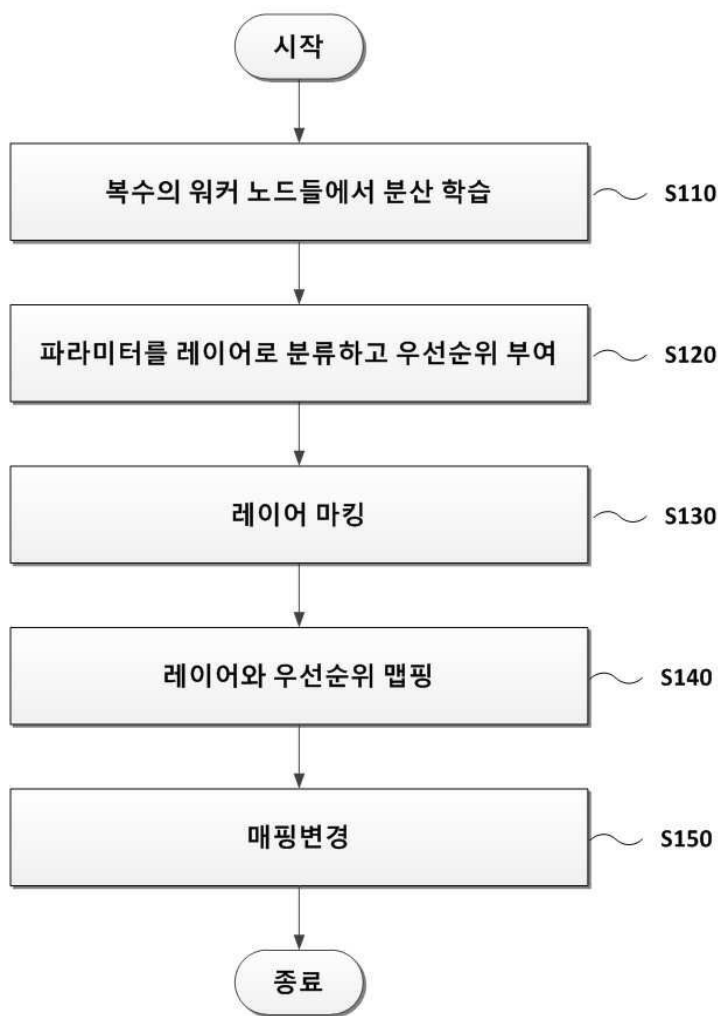
[0041] 이상, 첨부된 도면을 참조하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 이상에서 기술한 실시예에는 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

도면

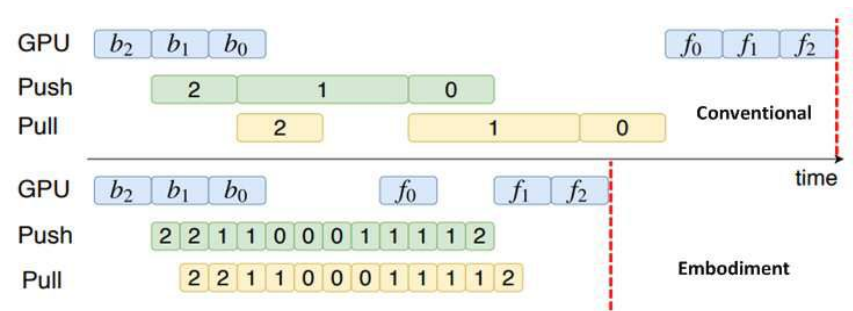
도면1



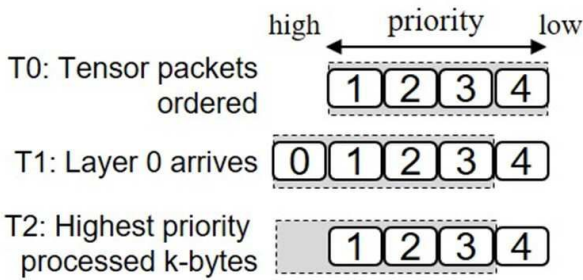
도면2



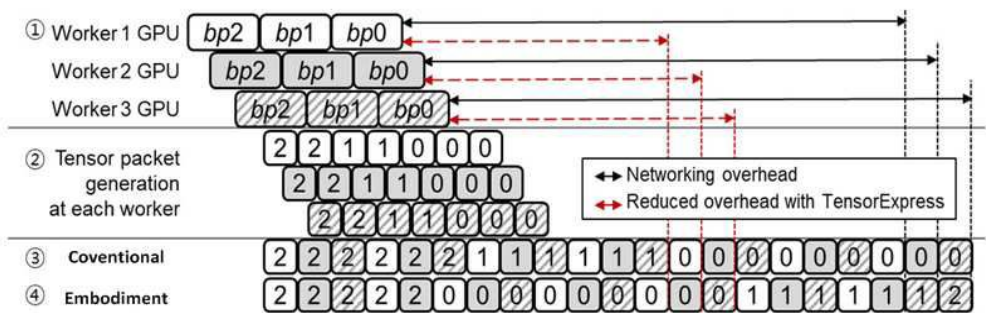
도면3



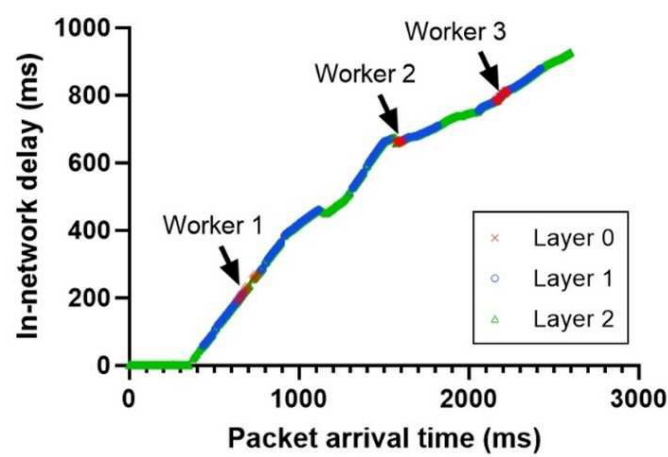
도면4



도면5



도면6



도면7

