(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0041154
(43) 공개일자 2020년04월21일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) *G06F 11/07* (2006.01)
G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/08 (2013.01)
G06F 11/0724 (2013.01)
(21) 출원번호 10-2018-0121247
(22) 출원일자 2018년10월11일
심사청구일자 2018년10월11일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
강성호
서울특별시 마포구 양화로 45, 101동 2102호 (서교동, 메세나폴리스)
이동수
서울특별시 서대문구 성산로18길 41, 205호 (연희동, 원캐슬)
(74) 대리인
특허법인우인

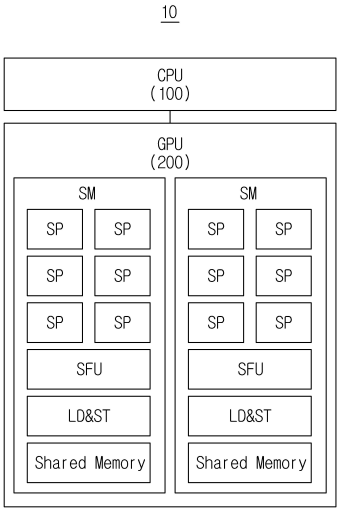
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 **드롭아웃이 적용된 다층 퍼셉트론 구조에서 연산코어의 오류 검출 방법 및 장치**

(57) 요 약

본 실시예들은 과적합을 해결하기 위한 드롭아웃 동작에서 드롭아웃이 적용되지 않은 활성화된 스트리밍 프로세서에서 진행된 원래의 연산 결과 값과 비활성화된 스트리밍 프로세서에서 진행한 연산 결과 값의 비교를 통해 각각의 코어에서 오류가 발생하였는지를 검증할 수 있는 컴퓨팅 시스템을 제공한다.

대 표 도 - 도1



(52) CPC특허분류
G06N 3/04 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2016-0-00140
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터(NIPA산하)
연구사업명 정보통신방송연구개발사업
연구과제명 [이지바로] HPC 시스템 응용 프로그램 최적화를 위한 개발도구 (3/3)
기 여 율 1/1
주관기관 정보통신기술진흥센터
연구기간 2018.01.01 ~ 2018.12.31

명 세 서

청구범위

청구항 1

다중 연산코어가 데이터를 병렬 처리하는 컴퓨팅 시스템이 다층 퍼셉트론 구조에서 연산코어의 오류 검출 방법에 있어서,

그래픽 프로세싱 유닛(Graphics Processing Unit, GPU)이 기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 상기 그래픽 프로세싱 유닛에 포함된 다중 연산코어 중에서 상기 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력하고, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용된 비활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력하는 단계;

상기 그래픽 프로세싱 유닛에 연결된 중앙 프로세싱 유닛(Central Processing Unit, CPU)은 상기 제1 결과값과 상기 제2 결과값을 비교하여 상기 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출하는 단계

를 포함하는 연산코어의 오류 검출 방법.

청구항 2

제1항에 있어서,

상기 드롭아웃에 관한 비율은 최대치 1을 기준으로 0.1부터 0.9까지의 범위 내에서 설정되며, 상기 드롭아웃에 관한 비율이 0.5 이상이면, 오류 검출율이 100%인 것을 특징으로 하는 연산코어의 오류 검출 방법.

청구항 3

제1항에 있어서,

상기 다층 퍼셉트론 구조는 네트워크로 연결된 복수의 레이어의 노드들을 포함하며, 상기 복수의 레이어는 입력 레이어, 은닉 레이어, 및 출력 레이어를 포함하며, 각각의 레이어는 가중치를 갖고 상기 가중치를 학습하는 것을 특징으로 하는 연산코어의 오류 검출 방법

청구항 4

제1항에 있어서,

상기 다층 퍼셉트론 구조에서 상기 드롭아웃에 따라 연산을 하지 않도록 설정된 노드가 인접한 노드의 연산을 동일하게 수행하며,

상기 그래픽 프로세싱 유닛에서 스트리밍 멀티프로세서(Streaming Multiprocessor, SM) 내의 제어 로직과 명령 캐시를 공유하는 스트리밍 프로세서(Streaming Processor, SP)에 대하여 상기 다층 퍼셉트론 구조에서 대응하는 노드의 개수를 블록으로 지정하여 생성된 스레드가 각각의 스트리밍 프로세서에서 연산을 수행하는 것을 특징으로 하는 연산코어의 오류 검출 방법.

청구항 5

제4항에 있어서,

상기 드롭아웃이 적용되지 않은 스트리밍 프로세서는 상기 제1 결과값을 출력하고,

상기 드롭아웃이 적용된 스트리밍 프로세서는 상기 제2 결과값을 출력하는 것을 특징으로 하는 연산코어의 오류 검출 방법.

청구항 6

다중 연산코어가 데이터를 병렬 처리하는 컴퓨팅 시스템에 있어서,

기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력하고, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용된 비활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력하는 그래픽 프로세싱 유닛; 및

상기 그래픽 프로세싱 유닛에 연결되어 상기 제1 결과값과 상기 제2 결과값을 비교하여 상기 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출하는 중앙 프로세싱 유닛

을 포함하는 데이터를 병렬 처리하는 컴퓨팅 시스템.

청구항 7

제6항에 있어서,

상기 드롭아웃에 관한 비율은 최대치 1을 기준으로 0.1부터 0.9까지의 범위 내에서 설정되며, 상기 드롭아웃에 관한 비율이 0.5 이상이면, 오류 검출율이 100%인 것을 특징으로 하는 데이터를 병렬 처리하는 컴퓨팅 시스템.

청구항 8

제6항에 있어서,

상기 다층 퍼셉트론 구조는 네트워크로 연결된 복수의 레이어의 노드들을 포함하며, 상기 복수의 레이어는 입력 레이어, 은닉 레이어, 및 출력 레이어를 포함하며, 각각의 레이어는 가중치를 갖고 상기 가중치를 학습하는 것을 특징으로 하는 데이터를 병렬 처리하는 컴퓨팅 시스템.

청구항 9

제6항에 있어서,

상기 다층 퍼셉트론 구조에서 상기 드롭아웃에 따라 연산을 하지 않도록 설정된 노드가 인접한 노드의 연산을 동일하게 수행하며,

상기 복수의 그래픽 프로세싱 유닛에서 스트리밍 멀티프로세서(Streaming Multiprocessor, SM) 내의 제어 로직과 명령 캐시를 공유하는 스트리밍 프로세서(Streaming Processor, SP)에 대하여 상기 다층 퍼셉트론 구조에서 대응하는 노드의 개수를 블록으로 지정하여 생성된 스레드가 각각의 스트리밍 프로세서에서 연산을 수행하는 것을 특징으로 하는 데이터를 병렬 처리하는 컴퓨팅 시스템.

청구항 10

제9항에 있어서,

상기 드롭아웃이 적용되지 않은 스트리밍 프로세서는 상기 제1 결과값을 출력하고,

상기 드롭아웃이 적용된 스트리밍 프로세서는 상기 제2 결과값을 출력하는 것을 특징으로 하는 데이터를 병렬 처리하는 컴퓨팅 시스템.

발명의 설명

기 술 분 야

[0001] 본 실시예가 속하는 기술 분야는 다중 연산코어가 데이터를 병렬 처리하는 컴퓨팅 시스템에서 연산코어의 오류를 검출하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 이 부분에 기술된 내용은 단순히 본 실시예에 대한 배경 정보를 제공할 뿐 종래기술을 구성하는 것은 아니다.

[0003] 다수의 인공 뉴런으로 구성된 인공 신경망은 대량의 데이터를 반복적으로 연산을 실시하여 각 뉴런에 그 연산

값을 학습한다. 이러한 대량의 데이터의 반복 연산은 소수의 연산 유닛으로 구성된 CPU(Central Processing Unit)에서 학습을 진행하기에는 학습 시간이 오래 걸린다는 단점이 있으며 이를 극복하기 위해 많은 수의 연산 유닛으로 구성된 GPU(Graphics Processing Unit)가 필요하다.

[0004] GPU를 활용한 인공 신경망의 학습의 효과가 검증되면서 다양한 방식의 인공 신경망의 학습이 GPU를 통해 이루어졌으나 GPU는 그 연산 결과에 대한 신뢰성을 확보 할 수 있는 수단이 빈약하다.

[0005] 현재는 메모리 입출력 단계에서 오류를 검증하는 수단인 ECC(Error Correction Code)정도의 보정 수단을 가지고 있다. ECC를 통해 메모리의 입출력 과정에서 오류를 보정하지만 많은 수의 연산 코어들의 연산과정에서 발생하는 오류들의 검증 및 보정을 할 수 없다.

[0006] 이러한 개별 코어의 연산 과정에서 발생하는 오류의 검증은 DMR(Dual Modular Redundancy)와 같은 방법을 이용하여 검증을 실시하였지만 DMR을 사용 할 경우 소비 자원이 두 배가 되는 단점이 존재하며 이는 한정된 자원의 활용을 통한 인공 신경망의 학습에 있어 학습시간의 지연을 발생 시키며 이는 효율성 측면에서 문제가 있다.

선행기술문헌

특허문헌

[0007] (특허문헌 0001) 한국공개특허공보 제10-2016-0099587호 (2016.08.22.)

발명의 내용

해결하려는 과제

[0008] 본 발명의 실시예들은 과적합을 해결하기 위한 드롭아웃 동작에서 드롭아웃이 적용되지 않은 활성화된 스트리밍 프로세서에서 진행된 원래의 연산 결과 값과 비활성화된 스트리밍 프로세서에서 진행한 연산 결과 값의 비교를 통해 각각의 코어에서 오류가 발생하였는지를 검증하는 데 발명의 주된 목적이 있다.

[0009] 본 발명의 명시되지 않은 또 다른 목적들은 하기의 상세한 설명 및 그 효과로부터 용이하게 추론할 수 있는 범위 내에서 추가적으로 고려될 수 있다.

과제의 해결 수단

[0010] 본 실시예의 일 측면에 의하면, 다중 연산코어가 데이터를 병렬 처리하는 컴퓨팅 시스템이 다층 퍼셉트론 구조에서 연산코어의 오류 검출 방법에 있어서, 그래픽 프로세싱 유닛(Graphics Processing Unit, GPU)이 기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 상기 그래픽 프로세싱 유닛에 포함된 다중 연산코어 중에서 상기 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력하고, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용된 비활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력하는 단계, 상기 그래픽 프로세싱 유닛에 연결된 중앙 프로세싱 유닛(Central Processing Unit, CPU)은 상기 제1 결과값과 상기 제2 결과값을 비교하여 상기 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출하는 단계를 포함하는 연산코어의 오류 검출 방법을 제공한다.

[0011] 본 실시예의 다른 측면에 의하면, 다중 연산코어가 데이터를 병렬 처리하는 컴퓨팅 시스템에 있어서, 기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력하고, 상기 다중 연산코어 중에서 상기 드롭아웃이 적용된 비활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력하는 그래픽 프로세싱 유닛, 및 상기 그래픽 프로세싱 유닛에 연결되어 상기 제1 결과값과 상기 제2 결과값을 비교하여 상기 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출하는 중앙 프로세싱 유닛을 포함하는 데이터를 병렬 처리하는 컴퓨팅 시스템을 제공한다.

발명의 효과

[0012] 이상에서 설명한 바와 같이 본 발명의 실시예들에 의하면, 과적합을 해결하기 위한 드롭아웃 동작에서 드롭아웃이 적용되지 않은 활성화된 스트리밍 프로세서에서 진행된 원래의 연산 결과 값과 비활성화된 스트리밍 프로세

서에서 진행한 연산 결과 값의 비교를 통해 각각의 코어에서 오류가 발생하였는지를 검증할 수 있는 효과가 있다.

[0013] 여기에서 명시적으로 언급되지 않은 효과라 하더라도, 본 발명의 기술적 특징에 의해 기대되는 이하의 명세서에서 기재된 효과 및 그 잠정적인 효과는 본 발명의 명세서에 기재된 것과 같이 취급된다.

도면의 간단한 설명

[0014] 도 1은 본 발명의 일 실시예에 따른 컴퓨팅 시스템을 예시한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 컴퓨팅 시스템에 적용된 다층 퍼셉트론 구조를 예시한 도면이다.

도 3은 본 발명의 일 실시예에 따른 컴퓨팅 시스템의 다중 연산 코어의 동작을 예시한 도면이다.

도 4는 본 발명의 다른 실시예에 따른 연산코어의 오류 검출 방법을 예시한 흐름도이다.

도 5는 본 발명의 실시예들에 따라 수행된 드롭아웃 비율에 대한 오류 검출율을 예시한 그래프이다.

발명을 실시하기 위한 구체적인 내용

[0015] 이하, 본 발명을 설명함에 있어서 관련된 공지기능에 대하여 이 분야의 기술자에게 자명한 사항으로서 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하고, 본 발명의 일부 실시예들을 예시적인 도면을 통해 상세하게 설명한다.

[0016] 도 1은 컴퓨팅 시스템을 예시한 블록도이다. 컴퓨팅 시스템은 과적합을 해결하기 위한 드롭아웃 동작에서 드롭아웃이 적용되지 않은 활성화된 스트리밍 프로세서에서 진행된 원래의 연산 결과 값과 비활성화된 스트리밍 프로세서에서 진행한 연산 결과 값의 비교를 통해 각각의 코어에서 오류가 발생하였는지를 검증한다. 컴퓨팅 시스템은 GPU를 활용한 인공 신경망의 학습과정에서 발생할 수 있는 오류를 탐지하고, 오류에 따른 인공 신경망에서 발생 가능한 오작동 및 학습시간 지연의 원인을 파악할 수 있고, 높은 신뢰성을 가지는 인공 신경망의 구축을 가능하게 한다.

[0017] 도 1을 참조하면, 컴퓨팅 시스템(10)은 중앙 프로세싱 유닛(CPU, 100) 및 그래픽 프로세싱 유닛(GPU, 200)을 포함한다. 컴퓨팅 시스템(10)은 도 1에서 예시적으로 도시한 다양한 구성요소들 중에서 일부 구성요소를 생략하거나 다른 구성요소를 추가로 포함할 수 있다. 예컨대, 컴퓨팅 시스템(10)은 저장부 또는 노스브릿지(North Bidge)를 추가로 포함할 수 있다. 노스브릿지는 CPU, 메모리, 바이오스 롬, GPU, 사우스브릿드 등의 고속 장치를 버스로 연결하여 제어하는 집적회로이다. 사우스브릿지는 주변장치의 데이터 흐름을 제어하거나 전원을 관리한다.

[0018] 중앙 프로세싱 유닛(CPU, 100)은 외부에서 정보를 입력 받고, 기억하고, 컴퓨터 프로그램의 명령어를 해석하여 연산하고, 외부로 출력하는 역할을 한다. 즉, 중앙 프로세싱 유닛은 컴퓨터 부품과 정보를 교환하면서 컴퓨터 전체의 동작을 제어한다.

[0019] 그래픽 프로세싱 유닛(GPU, 200)은 컴퓨터 그래픽스를 위한 계산을 수행할 뿐만 아니라, 응용 프로그램들의 계산에 사용할 수 있다. 프로그램 가능한 층과 고정도 연산을 그래픽 파이프라인에 연결하여 데이터에 스트림 프로세싱을 수행할 수 있다. 그래픽 프로세싱 유닛은 병렬로 한번에 하나의 커널을 흐름 속의 많은 레코드에 실행시킨다. 흐름이란 단순히 유사한 계산을 필요로 하는 레코드의 모음이며, 흐름으로 데이터 병렬성을 구현할 수 있다. 커널은 흐름 속의 각 요소에 적용되는 함수이다. 그래픽 프로세싱 유닛은 대량의 코어들을 포함하고, 내부에 메모리를 갖는다. 복수의 그래픽 프로세싱 유닛은 중앙 프로세싱 유닛(100) 또는 노스브릿지에 연결된다.

[0020] 그래픽 프로세싱 유닛(200)은 복수의 스트리밍 멀티프로세서(Streaming Multiprocessor, SM)를 포함한다. 그래픽 프로세싱 유닛(200)은 복수의 스트리밍 멀티프로세서의 동작을 제어하는 스트리밍 멀티프로세서 제어부를 포함할 수 있다. 각 스트리밍 멀티프로세서(SM)마다 독립적인 명령어 스케줄러를 갖추어 복수의 스레드를 동시에 실행할 수 있습니다. 스트리밍 프로세서(Streaming Processor, SP)는 기본적인 논리 및 수학 연산을 수행한다. SFU(Special Function Unit)는 초월 함수, 픽셀 속성 보간 등의 연산에 사용되며 부동 소수점 곱셈기도 포함한다. 스트리밍 멀티프로세서(SM)에서 여러 개의 스레드가 동시 실행될 때 기설정된 개수의 SP와 기설정된 개수의 SFU로 동일한 명령어(Instruction)가 전달(Broadcasting)되는데 이 때 각 유닛(SP 또는 SFU)은 동일한 명령을 수행하지만 레지스터와 메모리 주소는 각각 다르게 관리된다. 공유메모리(Shared Memory)는 기설정된 용량을 갖고 스트리밍 멀티프로세서 내에서 실행되는 스레드 사이의 데이터 교환을 수행한다. LD/ST는 독출 명령 또는 기

록 명령을 수행한다.

[0021] 그래픽 프로세싱 유닛(200)은 기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 다중 연산코어 중에서 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력한다. 그래픽 프로세싱 유닛(200)은 다중 연산코어 중에서 상기 드롭아웃이 적용된 비활성화 연산코어 블록에서 상기 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력한다.

[0022] 중앙 프로세싱 유닛(100)는 그래픽 프로세싱 유닛(200)에 연결되어 제1 결과값과 제2 결과값을 비교하여 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출한다.

[0023] 도 2는 컴퓨팅 시스템에 적용된 다층 퍼셉트론 구조를 예시한 도면이다.

[0024] 컴퓨팅 시스템은 다층 퍼셉트론(Multi-Layer Perceptron)의 구조와 비활성화된 코어에서 도출된 결과 값과 기존의 연산 코어에서 도출된 결과 값을 비교한다.

[0025] 다층 퍼셉트론의 구조에서 뉴런들의 활성화 함수(Activation Function)에 입력되기 전 가중치 연산은 수학식 1과 같이 표현된다.

수학식 1

[0026]
$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}^T = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}^T \times \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{n1} & \cdots & w_{nn} \end{bmatrix}$$

[0027] y는 출력값이고, x는 입력값이고, w는 가중치이고, n은 뉴런의 개수이다.

[0028] 다층 퍼셉트론의 연산은 n개의 인공 뉴런에 입력되는 입력값과 각 인공 뉴런이 나타내는 가중치를 합성 곱하여 도출되는 결과를 활성화 함수에 전달하여 활성화 함수에 의해 그 값을 조정하여 출력하게 된다.

[0029] 다층 퍼셉트론 구조는 네트워크로 연결된 하나 이상의 레이어의 노드들을 포함한다. 복수의 레이어는 입력 레이어, 은닉 레이어, 및 출력 레이어를 포함하며, 각각의 레이어는 가중치를 갖고 가중치를 학습한다. 각각의 레이어는 파라미터를 포함할 수 있다. 레이어의 파라미터는 학습가능한 필터 집합을 포함한다. 파라미터는 노드 간의 가중치 및/또는 바이어스를 포함한다. 복수의 파라미터를 학습하고, 일부 파라미터는 공유될 수 있다. 신경망의 레이어, 가중치의 초기치, 노드에 들어갈 바이어스, 노드에서 사용할 활성 함수, 신경망의 학습률, 전체 오차를 측정하는 손실함수, 신경망의 오차를 최소화하는 알고리즘, 과적응을 규제하는 알고리즘은 구현되는 설계에 따라 적합한 수치 및 함수가 설정될 수 있다.

[0030] 인공 신경망의 각 레이어마다 인공 뉴런의 출력 값을 연산하고 역전파(Back Propagation)을 통해 가중치를 조정하며 연산을 실시할 경우 과적합의 문제가 발생하게 된다. 학습 과정에서의 정확도는 높게 나오나 실제 사용시 정확도를 감소시킨다. 이러한 문제를 해결하기 위하여 드롭아웃(Dropout) 기법이 제시되었고, 드롭아웃이 임의의 뉴런에 적용된 연산의 예시는 수학식 2와 같이 표현된다.

수학식 2

[0031]
$$\begin{bmatrix} 0 \\ y_2 \\ 0 \\ y_4 \end{bmatrix}^T = \begin{bmatrix} x_1 \\ 0 \\ x_3 \\ 0 \end{bmatrix}^T \times \begin{bmatrix} 0 & w_{12} & 0 & w_{14} \\ 0 & 0 & 0 & 0 \\ 0 & w_{32} & 0 & w_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[0032] 수학식 2는 하나의 은닉 레이어(Hidden Layer)가 4개의 뉴런으로 구성된 인공 신경망에서 50%의 확률로 드롭아웃 과정을 진행했을 때를 나타낸다. 수학식 2를 전개하면, 수학식 3이 도출되며 학습과정에서 연산 부하를 줄여

학습 속도의 개선 및 과적합의 방지가 가능하다. 드롭아웃에 관한 비율은 최대치 1을 기준으로 0.1부터 0.9까지의 범위 내에서 설정될 수 있으며, 드롭아웃에 관한 비율이 0.5 이상이면, 오류 검출율이 100%이다.

수학식 3

[0033]
$$\begin{bmatrix} 0 \\ y_2 \\ 0 \\ y_4 \end{bmatrix}^T = \begin{bmatrix} 0 \\ w_{12}x_1 + w_{32}x_3 \\ 0 \\ w_{14}x_1 + w_{34}x_3 \end{bmatrix}^T$$

[0034] 컴퓨팅 시스템은 0의 연산에 인접한 뉴런의 연산을 동일하게 실시하여 그 값을 비교함으로써 GPU의 연산 코어를 사용하여 연산을 실시하였을 때 발생할 수 있는 오류를 검증한다. 다층 퍼셉트론 구조에서 드롭아웃에 따라 연산을 하지 않도록 설정된 노드가 인접한 노드의 연산을 동일하게 수행한다.

[0035] 컴퓨팅 시스템이 드롭아웃이 적용된 뉴런을 이용한 연산은 수학식 4와 같이 표현된다.

수학식 4

[0036]
$$\begin{bmatrix} y_2 \\ y_2 \\ y_4 \\ y_4 \end{bmatrix}^T = \begin{bmatrix} x_1 \\ x_1 \\ x_3 \\ x_3 \end{bmatrix}^T \times \begin{bmatrix} w_{12} & w_{12} & w_{14} & w_{14} \\ 0 & 0 & 0 & 0 \\ w_{32} & w_{32} & w_{34} & w_{34} \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

[0037] 수학식 4를 전개하면 수학식 5가 도출되며 행렬의 1항과 2항, 3항과 4항의 수식이 일치하는 것을 확인할 수 있다. 복수의 그래픽 프로세싱 유닛에서 스트리밍 멀티프로세서(Streaming Multiprocessor, SM) 내의 제어 로직과 명령 캐시를 공유하는 스트리밍 프로세서(Streaming Processor, SP)에 대하여 다층 퍼셉트론 구조에서 대응하는 노드의 개수를 블록으로 지정하여 생성된 스레드가 각각의 스트리밍 프로세서에서 연산을 수행한다.

수학식 5

[0038]
$$\begin{bmatrix} y_2 \\ y_2 \\ y_4 \\ y_4 \end{bmatrix}^T = \begin{bmatrix} w_{12}x_1 + w_{32}x_3 \\ w_{12}x_1 + w_{32}x_3 \\ w_{14}x_1 + w_{34}x_3 \\ w_{14}x_1 + w_{34}x_3 \end{bmatrix}^T$$

[0039] 수학식 5를 GPU의 스트리밍 프로세서에 대입하기 위해 스트리밍 멀티프로세서 집단의 스트리밍 프로세서의 개수에 해당하는 만큼 뉴런의 개수를 블록으로 지정하여 생성된 스레드가 각각의 스트리밍 프로세서에서 연산을 진행 할 수 있다. 드롭아웃이 적용되지 않은 스트리밍 프로세서에서 진행된 원래의 연산 결과값과 드롭아웃으로 인해 비활성화된 스트리밍 프로세서에서 진행한 연산 결과값의 비교를 통해 각각의 코어에서 오류가 발생하였는지를 검증한다.

[0040] GPU에서 수학식 5의 연산을 실시하였을 때의 예시는 도 3에 도시된다.

[0041] 드롭아웃이 적용되지 않은 스트리밍 프로세서는 제1 결과값을 출력하고, 드롭아웃이 적용된 스트리밍 프로세서

는 제2 결과값을 출력한다. 컴퓨팅 시스템은 제1 결과값과 제2 결과값을 비교하여 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출한다.

[0042] 컴퓨팅 시스템에 포함된 구성요소들이 도 1에서는 분리되어 도시되어 있으나, 복수의 구성요소들은 상호 결합되어 적어도 하나의 모듈로 구현될 수 있다. 구성요소들은 장치 내부의 소프트웨어적인 모듈 또는 하드웨어적인 모듈을 연결하는 통신 경로에 연결되어 상호 간에 유기적으로 동작한다. 이러한 구성요소들은 하나 이상의 통신 버스 또는 신호선을 이용하여 통신한다.

[0043] 컴퓨팅 시스템은 하드웨어, 펌웨어, 소프트웨어 또는 이들의 조합에 의해 로직회로 내에서 구현될 수 있고, 범용 또는 특정 목적 컴퓨터를 이용하여 구현될 수도 있다. 장치는 고정배선형(Hardwired) 기기, 필드 프로그램 가능한 게이트 어레이(Field Programmable Gate Array, FPGA), 주문형 반도체(Application Specific Integrated Circuit, ASIC) 등을 이용하여 구현될 수 있다. 또한, 장치는 하나 이상의 프로세서 및 컨트롤러를 포함한 시스템온칩(System on Chip, SoC)으로 구현될 수 있다.

[0044] 컴퓨팅 시스템은 하드웨어적 요소가 마련된 컴퓨팅 디바이스에 소프트웨어, 하드웨어, 또는 이들의 조합하는 형태로 탑재될 수 있다. 컴퓨팅 디바이스는 각종 기기 또는 유무선 통신망과 통신을 수행하기 위한 통신 모뎀 등의 통신장치, 프로그램을 실행하기 위한 데이터를 저장하는 메모리, 프로그램을 실행하여 연산 및 명령하기 위한 마이크로프로세서 등을 전부 또는 일부 포함한 다양한 장치를 의미할 수 있다.

[0045] 도 4는 본 발명의 다른 실시예에 따른 연산코어의 오류 검출 방법을 예시한 흐름도이다. 연산코어의 오류 검출 방법은 컴퓨팅 시스템에 의해 수행될 수 있다. 컴퓨팅 시스템이 수행하는 동작에 관한 상세한 설명과 중복되는 설명은 생략하기로 한다.

[0046] 단계 S410에서, 그래픽 프로세싱 유닛(Graphics Processing Unit, GPU)이 기설정된 비율로 드롭아웃(Drop Out)을 수행하면, 그래픽 프로세싱 유닛에 포함된 다중 연산코어 중에서 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제1 결과값을 출력하고, 다중 연산코어 중에서 드롭아웃이 적용된 비활성화 연산코어 블록에서 다층 퍼셉트론 구조에 따라 정의된 연산을 수행하여 제2 결과값을 출력한다. 드롭아웃이 적용되지 않은 스트리밍 프로세서는 제1 결과값을 출력하고, 드롭아웃이 적용된 스트리밍 프로세서는 제2 결과값을 출력한다.

[0047] 단계 S420에서, 그래픽 프로세싱 유닛에 연결된 중앙 프로세싱 유닛(Central Processing Unit, CPU)은 제1 결과값과 제2 결과값을 비교하여 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출한다.

[0048] 도 5는 GPU를 활용하여 다층 퍼셉트론을 활용하여 학습을 진행하는 대표적인 데이터 세트인 MNIST를 각기 다른 드롭아울 비율을 설정하여 진행하였을 때 오류 검출율을 나타낸 그래프이다. 드롭아웃 비율이 50% 이상일 때 비활성화된 스트리밍 프로세서가 모든 활성화된 스트리밍 프로세서를 포함할 수 있으므로 오류 검출율이 100%를 나타내게 된다.

[0049] 본 실시예들에 의하면, GPU를 활용한 다층 퍼셉트의 연산에서 발생하는 오류들에 대한 확인이 가능하며 학습과정에서 발생하는 비이상적인 수행시간에 대한 원인 파악이 가능하다. 인공 신경망에 대한 학습 결과의 신뢰성 확보가 가능하고, 보다 정확한 인공 신경망의 설계가 가능하다.

[0050] 도 4에서는 각각의 과정을 순차적으로 실행하는 것으로 기재하고 있으나 이는 예시적으로 설명한 것에 불과하고, 이 분야의 기술자라면 본 발명의 실시예의 본질적인 특성에서 벗어나지 않는 범위에서 도 4에 기재된 순서를 변경하여 실행하거나 또는 하나 이상의 과정을 병렬적으로 실행하거나 다른 과정을 추가하는 것으로 다양하게 수정 및 변형하여 적용 가능할 것이다.

[0051] 본 실시예들에 따른 동작은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능한 매체에 기록될 수 있다. 컴퓨터 판독 가능한 매체는 실행을 위해 프로세서에 명령어를 제공하는데 참여한 임의의 매체를 나타낸다. 컴퓨터 판독 가능한 매체는 프로그램 명령, 데이터 파일, 데이터 구조 또는 이들의 조합을 포함할 수 있다. 예를 들면, 자기 매체, 광기록 매체, 메모리 등이 있을 수 있다. 컴퓨터 프로그램은 네트워크로 연결된 컴퓨터 시스템 상에 분산되어 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수도 있다. 본 실시예를 구현하기 위한 기능적인(Functional) 프로그램, 코드, 및 코드 세그먼트들은 본 실시예가 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있을 것이다.

[0052] 본 실시예들은 본 실시예의 기술 사상을 설명하기 위한 것이고, 이러한 실시예에 의하여 본 실시예의 기술 사상

의 범위가 한정되는 것은 아니다. 본 실시예의 보호 범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 있는 모든 기술 사상은 본 실시예의 권리범위에 포함되는 것으로 해석되어야 할 것이다.

부호의 설명

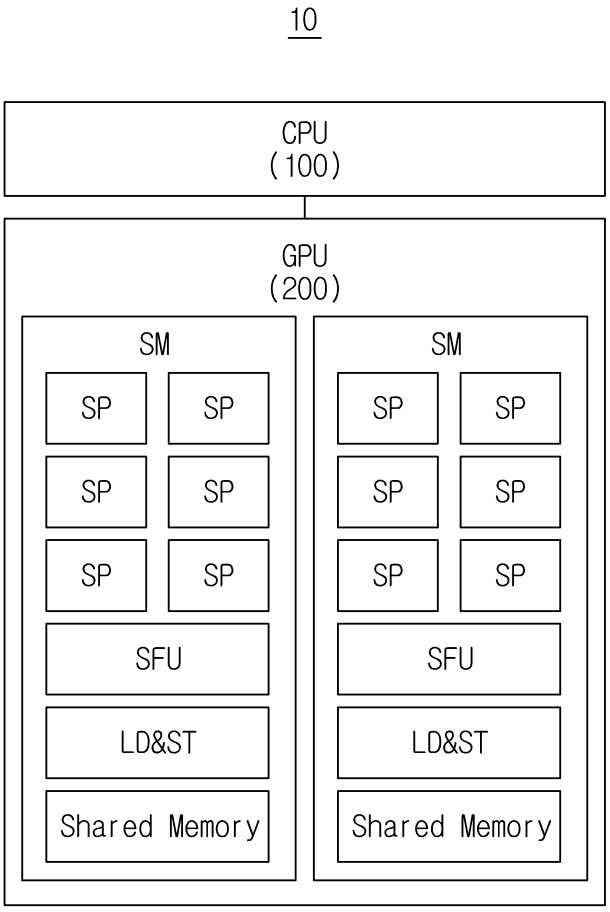
[0053] 10: 컴퓨팅 시스템

100: 중앙 프로세싱 유닛

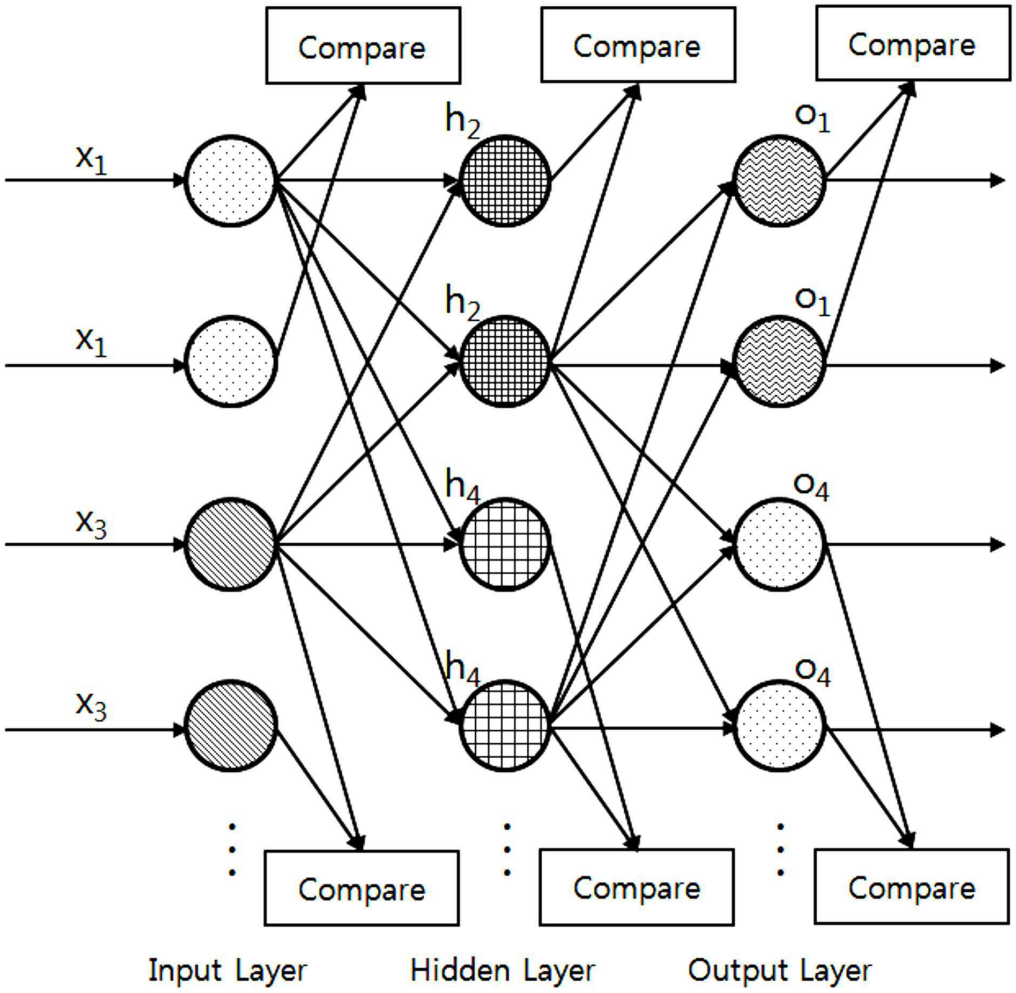
200: 그래픽 프로세싱 유닛

도면

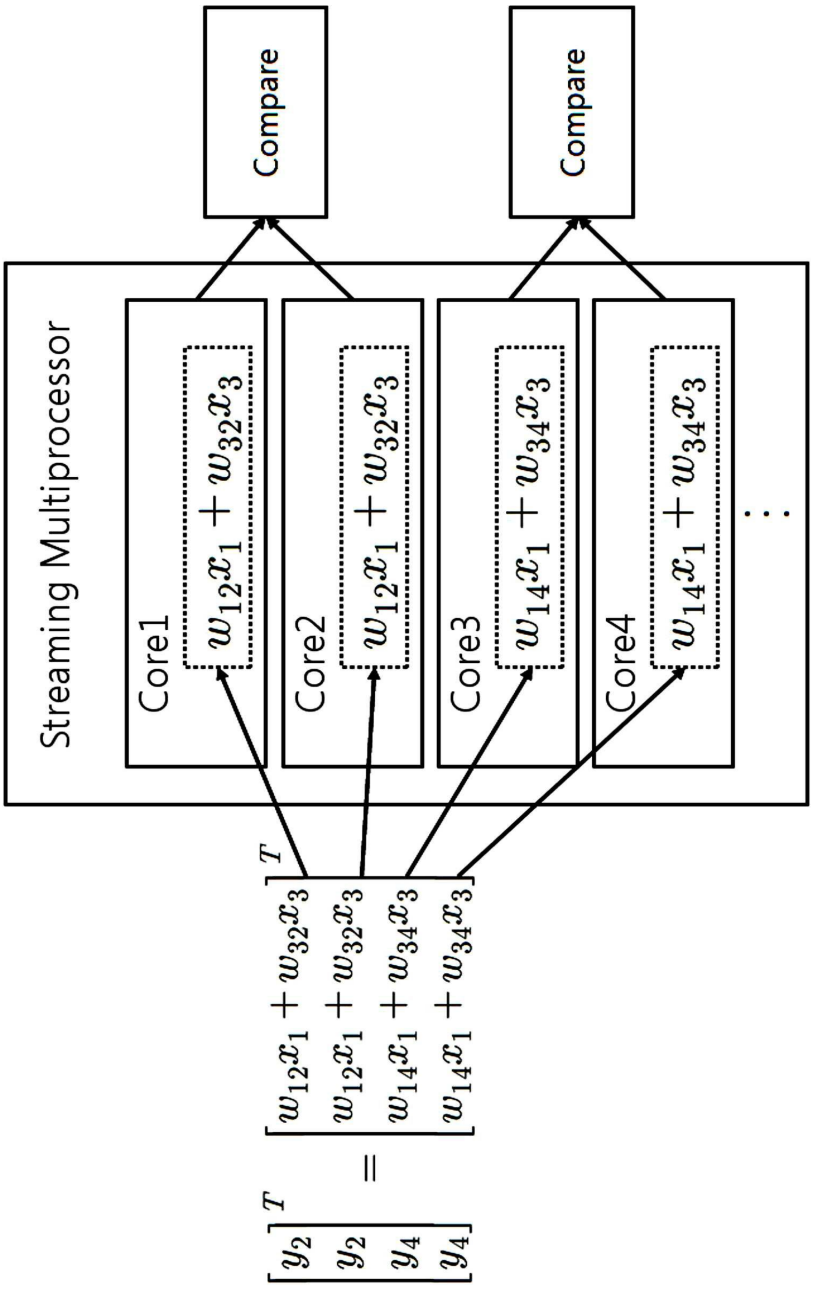
도면1



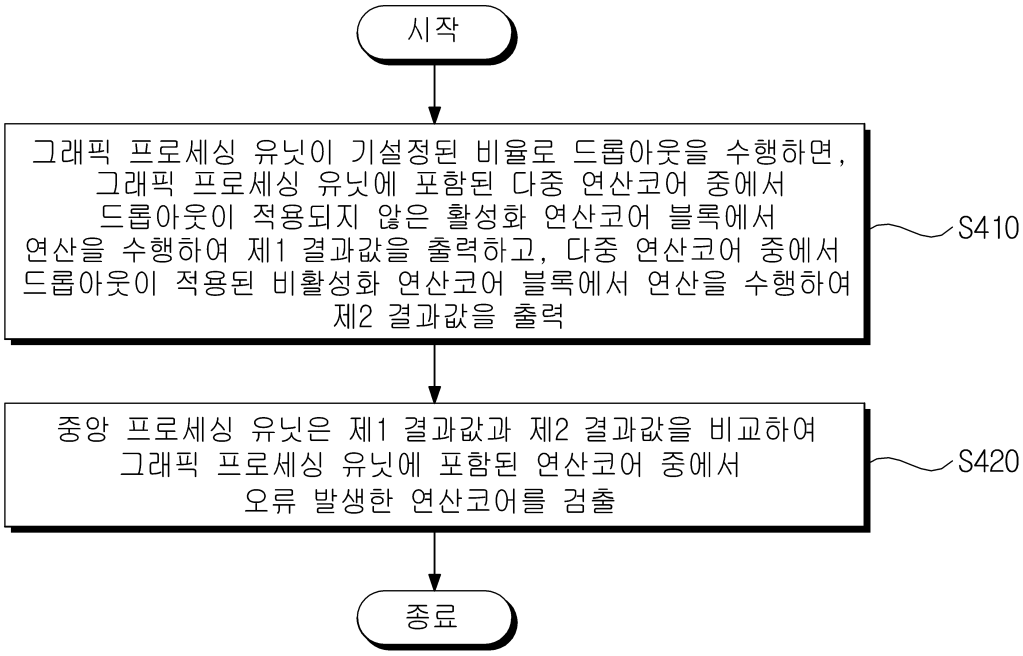
도면2


Compare
Compare
Compare
x_1
x_1
x_3
x_3
h_2
h_2
h_4
h_4
o_1
o_1
o_4
o_4
Compare
Compare
Compare
Input Layer
Hidden Layer
Output Layer

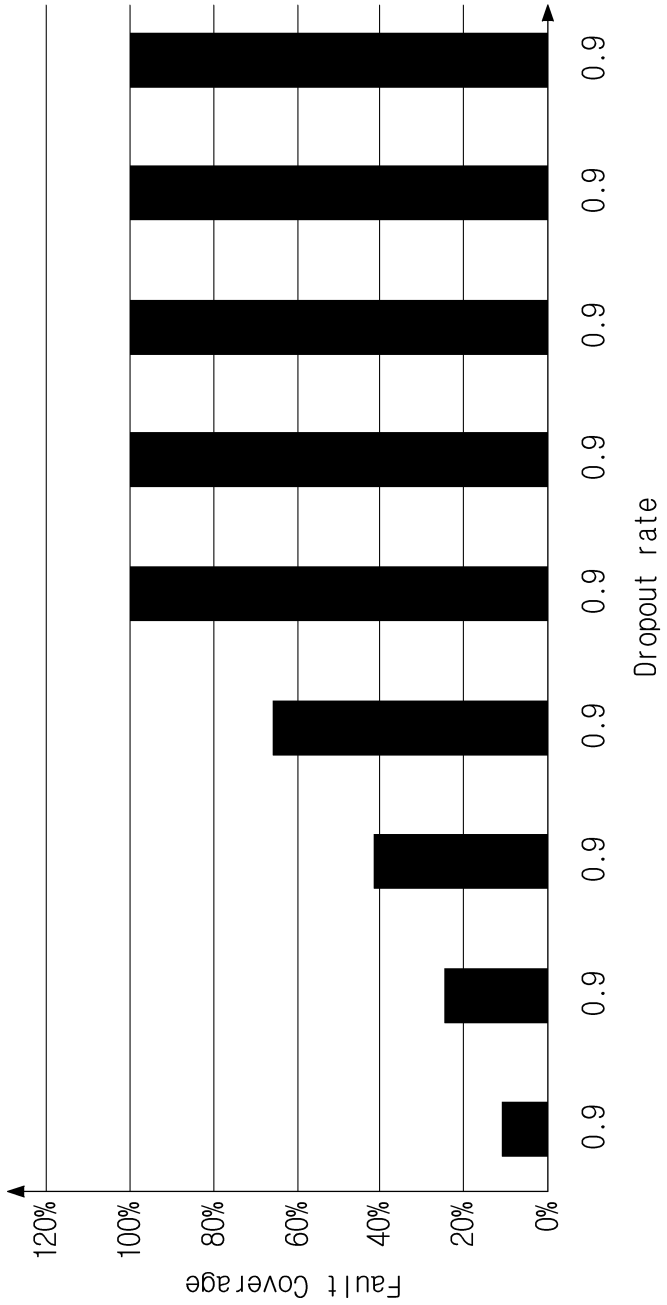
도면3


Compare
Compare
Streaming Multiprocessor
Core1
Core2
Core3
Core4
$w_{12}x_1 + w_{32}x_3$
$w_{12}x_1 + w_{32}x_3$
$w_{14}x_1 + w_{34}x_3$
$w_{14}x_1 + w_{34}x_3$
...
$\begin{bmatrix} y_2 & y_2 & y_4 & y_4 \end{bmatrix}^T = \begin{bmatrix} w_{12}x_1 + w_{32}x_3 & w_{12}x_1 + w_{32}x_3 & w_{14}x_1 + w_{34}x_3 & w_{14}x_1 + w_{34}x_3 \end{bmatrix}^T$

도면4


시작
그래픽 프로세싱 유닛이 기설정된 비율로 드롭아웃을 수행하면, 그래픽 프로세싱 유닛에 포함된 다중 연산코어 중에서 드롭아웃이 적용되지 않은 활성화 연산코어 블록에서 연산을 수행하여 제1 결과값을 출력하고, 다중 연산코어 중에서 드롭아웃이 적용된 비활성화 연산코어 블록에서 연산을 수행하여 제2 결과값을 출력
S410
중앙 프로세싱 유닛은 제1 결과값과 제2 결과값을 비교하여 그래픽 프로세싱 유닛에 포함된 연산코어 중에서 오류 발생한 연산코어를 검출
S420
종료

도면5


Fault Coverage
120%
100%
80%
60%
40%
20%
0%
Dropout rate
0.9
0.9
0.9
0.9
0.9
0.9
0.9
0.9