(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0110355
(43) 공개일자 2022년08월08일

(51) 국제특허분류(Int. Cl.)
*G06N 3/063* (2006.01) *G06N 3/04* (2006.01)
*G06N 5/04* (2006.01) *G11C 13/00* (2006.01)
(52) CPC특허분류
*G06N 3/063* (2013.01)
*G06N 3/04* (2013.01)
(21) 출원번호 10-2021-0012802
(22) 출원일자 2021년01월29일
심사청구일자 2021년01월29일

(71) 출원인
**울산과학기술원**
울산광역시 울주군 언양읍 유니스트길 50
**더 리젠츠 오브 더 유니버시티 오브 캘리포니아**
미국 캘리포니아 94607-5200 오클랜드 12층 프랭클린 스트리트 1111
**킹 압둘라 유니버시티 오브 사이언스 앤드 테크놀로지**
사우디아라비아 23955-6900 투왈 알-자즈리 빌딩 레벨 4 오피스 4221
(72) 발명자
**이종은**
울산 울주군 언양읍 유니스트길 50
**이수길**
울산 울주군 언양읍 유니스트길 50
*(뒷면에 계속)*
(74) 대리인
**김종석**

전체 청구항 수 : 총 4 항

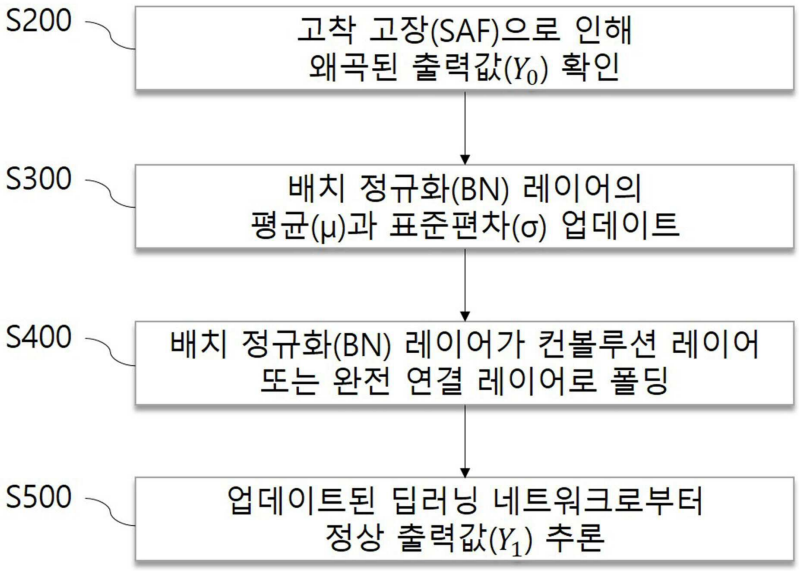
(54) 발명의 명칭 **저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법**

## (57) 요 약

본 발명은 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법에 관한 것으로, 보다 구체적으로, 임의의 데이터 셋이 이용되어 딥러닝 네트워크가 훈련되는 훈련 단계, 훈련된 상기 딥러닝 네트워크에 보정 데이터 셋이 이용되어 고착 고장(Stuck-At-Fault; SAF)으로 인해 왜곡된 출력값($Y_0$)이 확인되는 고착 고

(뒷면에 계속)

*대 표 도* - 도1



장 확인 단계, 상기 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(Batch Normalization; BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트 되는 업데이트 단계, 상기 평균(μ)과 표준편차(σ)가 업데이트된 상기 배치 정규화(BN) 레이어가 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)되는 폴딩 단계 및 상기 배치 정규화(BN) 레이어가 폴딩된 상기 딥러닝 네트워크가 이용되어 정상 출력값($Y_1$)이 추론되는 추론 단계를 포함하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법에 관한 것이다.

(52) CPC특허분류
*G06N 5/04* (2013.01)
*G11C 13/0002* (2018.05)

(72) 발명자
**정기주**
울산 울주군 언양읍 유니스트길 50
**모하메드 포우다**
유니버시티 오브 캘리포니아, 어바인
**파디 쿠르다히**
유니버시티 오브 캘리포니아, 어바인
**아미드 엘타윌**
킹 압둘라 유니버시티 오브 사이언스 앤드 테크놀리지, 사우비 아라비아

이 발명을 지원한 국가연구개발사업

| | |
|---|---|
| 과제고유번호 | 1711104953 |
| 과제번호 | 2016M3A7B4910576 |
| 부처명 | 과학기술정보통신부 |
| 과제관리(전문)기관명 | 한국연구재단 |
| 연구사업명 | 나노·소재기술개발(R&D) |
| 연구과제명 | 재구성로직 아키텍처 기술개발 |
| 기 여 율 | 1/3 |
| 과제수행기관명 | 울산과학기술원 |
| 연구기간 | 2020.01.01 ~ 2020.12.31 |

이 발명을 지원한 국가연구개발사업

| | |
|---|---|
| 과제고유번호 | 1711115216 |
| 과제번호 | 2020R1A2C2015066 |
| 부처명 | 과학기술정보통신부 |
| 과제관리(전문)기관명 | 한국연구재단 |
| 연구사업명 | 개인기초연구(과기정통부)(R&D) |
| 연구과제명 | RRAM 기반 뉴럴넷 시스템 설계를 위한 고속 시뮬레이션 기술 연구 |
| 기 여 율 | 2/3 |
| 과제수행기관명 | 울산과학기술원 |
| 연구기간 | 2020.03.01 ~ 2021.02.28 |

# 명 세 서

## 청구범위

### 청구항 1

저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 기 훈련된 딥러닝 네트워크에 보정 데이터 셋이 이용되어 고착 고장(Stuck-At-Fault; SAF)으로 인해 왜곡된 출력값($Y_0$)이 확인되는 고착 고장 확인 단계;

상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(Batch Normalization; BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트 되는 업데이트 단계;

상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 평균(μ)과 표준편차(σ)가 업데이트된 상기 배치 정규화(BN) 레이어가 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)되는 폴딩 단계; 및

상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 배치 정규화(BN) 레이어가 폴딩된 상기 딥러닝 네트워크가 이용되어 정상 출력값($Y_1$)이 추론되는 추론 단계;를 포함하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법.

### 청구항 2

제 1항에 있어서,

상기 딥러닝 네트워크는,

아핀 변환(Affine Transformation)이 기반이 된 하기 [수학식 1]이 이용되어 상기 왜곡된 출력값($Y_0$)과 정상 출력값($Y_1$) 중 적어도 하나가 도출되는 것을 특징으로 하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법.

[수학식 1]

$$y = \gamma\left(\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}\right)+\beta$$

여기서, y는 출력값이고, x는 입력값이고, μ와 σ는 순방향 매개변수인 상기 평균과 상기 표준편차이고, β와 γ는 역방향 매개변수인 편향값과 아핀변환의 가중치이고, ε는 상수이다.

### 청구항 3

제 2항에 있어서,

상기 업데이트 단계는,

상기 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ) 이외의 매개변수에 대한 업데이트가 발생되지 않는 것을 특징으로 하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법.

### 청구항 4

제 1항에 있어서,

상기 업데이트 단계는,

상기 딥러닝 네트워크에 상기 배치 정규화(BN) 레이어가 없는 경우 고착 고장(Stuck-at-fault)이 보상되도록 상기 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어 일측에 상기 배치 정규화(BN) 레이

어가 추가되고,

상기 폴딩 단계는,

상기 추론 단계가 단순화되도록 상기 배치 정규화(BN) 레이어가 상기 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어로 폴딩(folding)되는 것을 특징으로 하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법.

## 발명의 설명

### 기 술 분 야

[0001] 본 발명은 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법에 관한 것으로, 보다 구체적으로 딥러닝에 활용될 수 있는 차세대 메모리 중 하나인 저항성 메모리 소자(ReRAM)에 발생하는 고착 고장을 비학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)으로 보상하는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법에 관한 것이다.

### 배 경 기 술

[0002] 종래 저항성 메모리 소자(Resistive Random Access Memories; ReRAM)는 크로스바 구조에서 간단한 전류 합산을 통해 높은 Roff-Ron 비율과 높은 장치 밀도는 물론 매우 빠른 행렬 곱셈(Matrix Vector Multiplication; MVM) 기능을 제공할 수 있다. 이에 따라, 저항성 메모리 소자(ReRAM)는 심층 신경망 가속기 및 최적화 기술을 활성화 하는데 일조하고 있다.

[0003] 정상적인 경우 저항성 메모리 소자(ReRAM)의 전류는 멀티 비트 동작에서 하나의 디지털 상태에 대한 허용된 전류 범위인 특정 목표 전류 범위에 도달할 때까지 점진적으로 변화한 후, 반대 방향의 전기 펄스가 인가되어 원래의 저항 상태로 돌아가게 한다.

[0004] 다만, 비정상적인 경우 저항이 잘 변하지 않는 고착 고장(Stuck-At Fault; SAF)이 발생하게 된다. 고착 고장(SAF)이 발생하면 해당 소자의 저항값이 최저(Low Resistance State; LRS) 또는 최고(High Resistance State; HRS)로 고착되어 원하는 값, 즉 네트워크의 가중치 값을 제대로 쓸 수 없게 되고, 모델 성능이 떨어지게 된다.

[0005] 이를 해결하기 위해서 관련문헌 1 내지 3과 같이 종래에는 고착 고장을 피해서 매핑(remapping)하거나, 고장으로 발생한 에러를 후보정 해주거나(correction), 고착 고장을 고려한 재훈련 방법(retraining) 등이 제시된 바 있다.

[0006] 다만, 매핑(remapping) 방법과 후보정(correction) 방법은 고착 고장이 어디서 발생했는지 알아야 하고, 재훈련 방법(retraining)은 추가적인 하드웨어나 재훈련을 위한 전체 훈련 데이터 셋이 필요한 문제가 있다.

[0007] 따라서 딥러닝 가속기에 사용되는 저항성 메모리 소자(ReRAM)에서 발생할 수 있는 고착 고장(Stuck-At-Fault; SAF)을 보상하고 해당 소자의 신뢰성을 높일 수 있는 보다 효율적인 기술이 필요한 실정이다.

## 선행기술문헌

### 비특허문헌

[0008] (비특허문헌 0001) L. Xia et al., “Stuck-at fault tolerance in RRAM computing systems,” IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 8, no. 1, pp. 102-115, Mar. 2018.

(비특허문헌 0002) L. Chen et al., “Accelerator-friendly neural-network training: Learning variations and defects in RRAM crossbar,” in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017

(비특허문헌 0003) C. Liu et al., “Rescuing memristor-based neuromorphic design with high defects,” in Proceedings of the 54th Annual Design Automation Conference 2017.

## 발명의 내용

### 해결하려는 과제

[0009] 본 발명은 상기와 같은 문제점을 해결하기 위한 것으로 딥러닝 가속기에 사용되는 저항성 메모리 소자(ReRAM)에서 발생할 수 있는 고착 고장(Stuck-At-Fault; SAF)을 보상하고 해당 소자의 신뢰성을 높일 수 있도록 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트되는 비학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)을 적용한 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법을 얻고자 하는 것을 목적으로 한다.

[0010] 또한, 본 발명은 임의의 데이터 셋을 구축할 필요가 없고, 별도의 추가 하드웨어 역시 필요하지 않도록 임의의 데이터 셋에서 무작위로 선택된 하위 집합을 보정 데이터 셋으로 사용되도록 구비되고, 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트 된 후 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding) 가능하도록 구비되는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법을 얻고자 하는 것을 목적으로 한다.

### 과제의 해결 수단

[0011] 상기 목적을 달성하기 위하여, 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법은 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 기 훈련된 상기 딥러닝 네트워크에 보정 데이터 셋이 이용되어 고착 고장(Stuck-At-Fault; SAF)으로 인해 왜곡된 출력값($Y_0$)이 확인되는 고착 고장 확인 단계; 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(Batch Normalization; BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트 되는 업데이트 단계; 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 평균(μ)과 표준편차(σ)가 업데이트된 상기 배치 정규화(BN) 레이어가 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)되는 폴딩 단계; 및 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어에 의하여, 상기 배치 정규화(BN) 레이어가 폴딩된 상기 딥러닝 네트워크가 이용되어 정상 출력값($Y_1$)이 추론되는 추론 단계;를 제공한다.

### 발명의 효과

[0012] 이상과 같이 본 발명에 의하면 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트되는 비학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)을 적용함으로써, 딥러닝 가속기에 사용되는 저항성 메모리 소자(ReRAM)에서 발생할 수 있는 고착 고장(Stuck-At-Fault; SAF)이 발생하는 위치, 고착 고장(SAF) 패턴에 대한 정보를 확인할 필요 없이 고착 고장(SAF) 여부가 판단된 후 이를 보상하고 해당 소자의 신뢰성을 높일 수 있는 효과가 있다.

[0013] 또한, 본 발명은 임의의 데이터 셋에서 무작위로 선택된 하위 집합을 보정 데이터 셋으로 사용되도록 구비되고, 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)가 업데이트 된 후 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)됨으로써, 방대한 양의 임의의 데이터 셋이 구축될 필요가 없고 별도의 추가 하드웨어 역시 필요하지 않으므로 종래 보정방식보다 효율적인 효과가 있다.

[0014] 또한, 본 발명의 비학습 파라미터 튜닝방식(FPT)은 역전파 등의 훈련 방법은 사용되지 않으며 오로지 순방향 매개변수만을 업데이트함으로써, 많은 계산량을 요구하는 미분값 계산 역시 필요 없고 방대한 양의 임의의 데이터 셋이 아닌 적은 양의 보정 데이터 셋만으로 충분하고, 상기 보정 데이터 셋은 데이터 라벨링(Labeling)이 되지 않은 캘리브레이션 데이터 셋(Calibration Data Set)이므로 딥러닝 네트워크가 스스로 학습할 수 있는 형태로 사람이 방대한 양의 데이터를 분류, 가공하는 데이터 라벨링(Labeling) 과정을 거치지 않을 수 있는 현저한 효과가 있다.

### 도면의 간단한 설명

[0015] 도 1은 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법 흐름도이다.

도 2는 본 발명의 일실시예에 따른 각 단계별 정규화 및 아핀 변환 파라미터를 표시한 도면이다.

도 3은 본 발명의 일실시예에 따른 본 발명의 비학습 파라미터 튜닝방식(FPT)이 적용된 경우와 그렇지 않은 경우를 표시한 흐름도이다.

도 4는 본 발명의 일실시예에 따른 고착 고장이 없는 경우(a), 고착 고장이 확인된 경우(b), 고착 고장이 확인된 후 본원발명이 적용된 경우(c)에 따라 딥러닝 네트워크로부터 추론된 출력값이 표시된 도면이다.

## *발명을 실시하기 위한 구체적인 내용*

[0016] 본 명세서에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 판례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 발명에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 발명의 전반에 걸친 내용을 토대로 정의되어야 한다.

[0017] 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 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0018] 이하, 본 발명에 따른 실시예를 첨부한 도면을 참조하여 상세히 설명하기로 한다. 도 1은 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법 흐름도이다.

[0019] 우선, 본 발명은 기존의 딥러닝 가속기에 사용되는 저항성 메모리 소자(ReRAM)에서 발생할 수 있는 고착 고장(Stuck-At-Fault; SAF)을 보상하고 해당 소자의 신뢰성을 높일 수 있도록 딥러닝 네트워크 내 배치 정규화(Batch Nomalization; BN) 레이어의 순방향 파라미터인 평균($\mu$)과 표준편차($\sigma$)를 업데이트함으로써, 왜곡된 출력값($Y_0$)이 보상되고 정상 출력값($Y_1$)이 추론되는 비학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)을 적용한 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법이다.

[0020] 즉, 본원발명은 역전파(Backpropagation) 학습을 거치지 않는다는 점에서 비(非)학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)이라고 명명될 수 있다.

[0021] 도 1을 보면, 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법은 고착 고장 확인 단계(S200), 업데이트 단계(S300), 폴딩 단계(S400) 및 추론 단계(S500)를 포함한다.

[0022] 여기서, 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법은 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여 실행된다. 또는 본 발명을 실행하는 주체로써, 시뮬레이터 역시 사용될 수 있다.

[0023] 한편, 본 발명의 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기를 위한 고착 고장 보상 방법은 딥러닝 네트워크가 훈련될 수 있도록 훈련단계(S100)를 더 포함할 수 있다. 상기 훈련단계(S100)는 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여, 임의의 데이터 셋이 이용되어 딥러닝 네트워크가 훈련될 수 있다.

[0024] 상기 임의의 데이터 셋은 정답값이 있는 방대한 양의 데이터가 포함된 집합으로써, 상기 딥러닝 네트워크가 훈련될 수 있도록 구비될 수 있다.

[0025] 상기 딥러닝 네트워크는 아핀 변환(Affine Transformation)이 기반이 된 하기 [수학식 1]이 이용되어 상기 왜곡된 출력값($Y_0$)과 정상 출력값($Y_1$) 중 적어도 하나가 도출될 수 있다.

*수학식 1*

$$y = \gamma\left(\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}}\right)+\beta$$

[0026]

[0027] 여기서, y는 출력값이고, x는 입력값이고, μ와 σ는 순방향 매개변수인 평균과 표준편차이고, β와 γ는 역방향 매개변수인 편향값과 아핀 변환의 가중치이고, ε는 상수이다.

[0028] 다시 말하면, 상기 딥러닝 네트워크는 배치 정규화(BN) 레이어, 컨볼루션(Convolution) 레이어, 완전 연결(Fully-Connected) 레이어 중 적어도 하나를 포함할 수 있고, 상기 배치 정규화(BN) 레이어는 아핀 변환(Affine Transformation)이 기반이 된 하기 [수학식 1]이 이용되어 정규화 및 아핀 변환이 가능하다. 하기 [수학식 1]에서 괄호 안 부분이 정규화 과정이고, 이외 부분이 아핀 변환 과정일 수 있다. 여기서, 상기 μ와 σ는 정규화 파라미터로써 평균과 표준편차이고, 순방향에서 업데이트될 수 있는 순방향 매개변수이다. β와 γ는 아핀 변환 파라미터로써 편향값과 아핀 변환의 가중치이고, 역방향에서 업데이트될 수 있는 역방향 매개변수이다. 또한, ε는 상수이고, 분모가 0이 되는 것을 방지하기 위해서 더해주는 매우 작은 값으로 가장 바람직하게 $10^{-5}$일 수 있다.

[0029] 일반적으로 아핀 변환(Affine Transformation)은 n차원 공간이 1차식으로 나타내어지는 점 대응을 말한다. 아핀 변환(Affine Transformation)은 내부 공변량 이동 문제를 줄일 수 있고 훈련 성능을 향상시킬 수 있다.

[0030] 또한, 상기 딥러닝 네트워크 내 배치 정규화(BN) 레이어는 상기 훈련 단계(S100), 업데이트 단계(S300) 및 추론 단계(S500)로부터 다르게 실행될 수 있다.

[0031] 도 2는 본 발명의 일실시예에 따른 각 단계별 정규화 및 아핀 변환 파라미터를 표시한 도면이다. 도 2의 (a)를 보면, 상기 훈련 단계(S100)는 상기 [수학식 1]에 임의의 데이터 셋에 포함되는 입력값(x)이 입력되고 상기 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)를 포함하는 순방향 매개변수가 이용되어, 출력값(y)이 도출될 수 있다. 이때, 상기 출력값(y)은 왜곡된 출력값($Y_0$)과 정상 출력값($Y_1$) 중 적어도 하나일 수 있다.

[0032] 그리고 상기 훈련 단계(S100)는 상기 역방향 매개변수가 역방향에서 경사하강법이 이용되어 훈련될 수 있다. 일반적으로 경사하강법은 1차 근삿값 발견용 최적화 알고리즘으로 함수의 기울기를 구하고 기울기의 절대값이 낮은 쪽으로 계속 이동시켜 극값에 이를 때까지 반복시키는 방법으로, 다양한 시작점에 대해 경사하강법을 적용하면 최적해를 찾을 수 있는 효과가 있다. 즉, 상기 4개의 매개변수는 훈련 중에만 업데이트될 수 있다.

[0033] 또한, 도 2의 (b)를 보면, 상기 업데이트 단계(S300)는 상기 [수학식 1]에 보정 데이터 셋에 포함되는 입력값(x)이 입력되고 상기 배치 정규화(BN) 레이어의 순방향 매개변수인 평균(μ)과 표준편차(σ), 상기 훈련 단계(S100)로부터 획득된 역방향 매개변수인 편향값(β)과 아핀변환의 가중치(γ)가 이용되어, 출력값(y)이 도출될 수 있다. 이때, 상기 출력값(y)은 왜곡된 출력값($Y_0$)과 정상 출력값($Y_1$) 중 적어도 하나일 수 있다.

[0034] 또한, 도 2의 (c)를 보면, 상기 추론 단계(S500)는 상기 [수학식 1]에 임의의 데이터 셋에 포함되는 입력값(x)이 입력되고 상기 업데이트 단계(S300)로부터 업데이트된 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)의 지수 이동 평균(Exponential Moving Average; EMA)과 상기 훈련 단계(S100)로부터 획득된 역방향 매개변수가 이용되어 출력값(y)이 추론될 수 있다. 이때, 추론되는 출력값(y)은 가장 바람직하게 정상 출력값($Y_1$)일 수 있다.

[0035] 다음으로, 상기 고착 고장 확인 단계(S200)는 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여, 기 훈련된 딥러닝 네트워크에 보정 데이터 셋이 이용되어 고착 고장(Stuck-At-Fault; SAF)으로 인해 왜곡된 출력값($Y_0$)이 확인된다.

[0036] 한편, 상기 고착 고장 확인 단계(S200)는 고착 고장(Stuck-At-Fault; SAF)이 확인된다면 뉴런의 수에 따라서 다수 개의 왜곡된 출력값($Y_0$)이 확인될 수 있고, 고착 고장(Stuck-At-Fault; SAF)이 확인되지 않는다면 뉴런의 수

에 따라서 정상 출력값($Y_1$)이 확인될 수 있다.

[0037] 즉, 상기 고착 고장 확인 단계(S200)는 고착 고장(SAF)이 발생하는 위치, 고착 고장(SAF) 패턴에 대한 정보를 확인할 필요 없이 상기 딥러닝 네트워크로부터 출력된 출력값만이 이용되어 용이하게 고착 고장(Stuck-At-Fault; SAF) 여부가 판단될 수 있는 효과가 있다.

[0038] 이때, 상기 고착 고장 확인 단계(S200)는 도 4와 같이 가장 바람직하게 왜곡된 출력값($Y_0$)이 다수 개의 뉴런의 출력을 모델링하는 가우스 분포로 나타나는 것을 특징으로 한다.

[0039] 다음으로, 상기 업데이트 단계(S300)는 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여, 상기 왜곡된 출력값($Y_0$)이 이용되어 배치 정규화(BN) 레이어의 평균($\mu$)과 표준편차($\sigma$)가 업데이트 된다.

[0040] 가장 바람직하게, 상기 업데이트 단계(S300)는 상기 배치 정규화 레이어(BN)의 평균($\mu$)과 표준편차($\sigma$) 이외의 매개변수에 대한 업데이트가 발생되지 않는 것을 특징으로 한다. 즉, 상기 [수학식 1]에 포함된 역방향 매개변수인 편향값($\beta$)와 아핀 변환 가중치($\gamma$)는 업데이트가 발생되지 않고, 오로지 순방향 매개변수인 평균($\mu$)과 $\sigma$(표준편차)만이 업데이트되도록 구성됨으로써, 상기 딥러닝 네트워크의 연산량을 현저히 줄일 수 있고 이에 따라 신속하게 고착 고장(SAF)에 대한 보상이 가능하도록 하는 현저한 효과가 있다.

[0041] 다음으로, 상기 폴딩 단계(S400)는 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여, 상기 평균($\mu$)과 표준편차($\sigma$)가 업데이트된 상기 배치 정규화(BN) 레이어가 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)된다. 여기서, 상기 딥러닝 네트워크 내 상기 배치 정규화(BN) 레이어가 존재할 경우이고, 폴딩(folding)은 합쳐짐으로써 제거된다는 의미이다.

[0042] 반면에, 상기 업데이트 단계(S300)는 상기 딥러닝 네트워크 내 상기 배치 정규화(BN) 레이어가 없는 경우 고착 고장(Stuck-at-fault)이 보상되도록 상기 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어 일측에 상기 배치 정규화(BN) 레이어가 추가될 수 있다.

[0043] 이때, 상기 업데이트 단계(S300)는 상기 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어로부터 출력되는 상기 왜곡된 출력값($Y_0$)이 이용되어 추가된 상기 배치 정규화(BN) 레이어의 평균($\mu$)과 표준편차($\sigma$)가 업데이트될 수 있다.

[0044] 그리고 상기 폴딩 단계(S400)는 상기 추론 단계(S500)가 단순화되도록 상기 배치 정규화(BN) 레이어가 상기 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어로 폴딩(folding)될 수 있다.

[0045] 이에 따라, 상기 배치 정규화(BN) 레이어가 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어로 합쳐지는 폴딩(folding)이 가능하므로 별도의 추가 하드웨어가 필요하지 않는 효과가 있다.

[0046] 다음으로, 상기 추론 단계(S500)는 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여, 상기 배치 정규화(BN) 레이어가 폴딩된 상기 딥러닝 네트워크가 이용되어 정상 출력값($Y_1$)이 추론된다.

[0047] 도 3은 본 발명의 일실시예에 따른 본 발명의 비학습 파라미터 튜닝방식(FPT)이 적용된 경우와 그렇지 않은 경우를 표시한 흐름도이다. 도 4는 본 발명의 일실시예에 따른 고착 고장이 없는 경우(a), 고착 고장이 확인된 경우(b), 고착 고장이 확인된 후 본원발명이 적용된 경우(c)에 따라 딥러닝 네트워크로부터 추론된 출력값이 표시된 도면이다.

[0048] 우선, 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에 의하여 훈련된 딥러닝 네트워크가 저항성 메모리 소자(ReRAM)에 매핑(mapping)되고 고착 고장(SAF)이 발생되면, 도 4의 (b)와 같이 배치 정규화(BN) 레이어의 출력값의 분포가 왜곡될 수 있다.

[0049] 도 3을 보면, 본 발명의 비학습 파라미터 튜닝방식(FPT)이 적용되지 않은 종래의 경우 고착 고장(SAF)이 발생되지 않는다면 도 4의 (a)와 같이 상기 왜곡된 출력값($Y_0$)과 정상 출력값($Y_1$)의 분포가 나타날 수 있다. 예컨대, 정상 출력값($Y_1$) 중에서 가장 큰 피크는 0 근처에서 확인되고, 작은 피크는 3 근처에서 확인될 수 있다.

[0050] 본 발명의 비학습 파라미터 튜닝방식(FPT)이 적용되지 않은 종래의 경우 고착 고장(SAF)이 발생된다면 도 4의 (b)와 같이 왜곡된 출력값($Y_0$)의 분포가 나타날 수 있다. 즉, 왜곡 패턴이 뉴런마다 달라지므로 출력값(y)의 평균이 이동되고 표준편차가 증가될 수 있다. 이에 따라, 상기 딥러닝 네트워크의 성능이 크게 저하될 수 있는 문

제점이 있다.

[0051] 반면에, 고착 고장(SAF)이 발생된 경우 본 발명의 비학습 파라미터 튜닝방식(FPT)이 적용된다면 상기 왜곡된 출력값($Y_0$)이 이용되어 상기 딥러닝 네트워크 내 상기 배치 정규화(BN) 레이어의 평균(μ)과 표준편차(σ)가 보정될 수 있고, 이에 따라 도 4의 (b)와 같이 왜곡된 출력값($Y_0$)의 분포가 도 4의 (c)와 같이 정상 출력값($Y_1$)으로 회복되고, 상기 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어(100)에서 훈련된 딥러닝 네트워크의 성능 또한 높아진다.

[0052] 또한, 본 발명의 비학습 파라미터 튜닝방식(FPT)은 역전파 등의 훈련 방법은 사용되지 않으며 오로지 순방향 매개변수만을 업데이트함으로써, 많은 계산량을 요구하는 미분값 계산 역시 필요 없고 방대한 양의 임의의 데이터 셋이 아닌 적은 양의 보정 데이터 셋만으로 충분하다. 예컨대, 딥러닝 네트워크 훈련을 위한 50,000개의 상기 임의의 데이터 셋과 비교해 상기 보정 데이터 셋은 1,024개만으로 충분할 수 있다.

[0053] 또한, 상기 보정 데이터 셋은 데이터 라벨링(Labeling)이 되지 않은 데이터의 집합일 수 있고, 이를 켈리브레이션 데이터 셋(Calibration Data Set)이라고 칭할 수 있다. 데이터 라벨링(Labeling)은 딥러닝 네트워크가 스스로 학습할 수 있는 형태로 사람이 방대한 양의 데이터를 분류, 가공하는 과정임으로, 이러한 별도의 과정을 거치지 않을 수 있는 현저한 효과가 있다.

[0054] 또한, 배치 정규화(BN) 레이어의 평균과 표준편차가 업데이트 시 별도의 훈련이 진행되지 않으므로 상기 보정 데이터 셋에 정답값이 없는 데이터가 문제없이 사용될 수 있다. 이렇게 업데이트된 배치 정규화(BN) 레이어의 파라미터는 직전 레이어인 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 합쳐지는 폴딩(folding)이 가능하고, 폴딩(folding)된다는 것은 합쳐져 제거된다는 의미이므로, 별도의 추가 하드웨어가 필요하지 않는 효과가 있다.

[0056] **실시예 1**

[0057] *실험방법*

[0058] 본원발명의 효과를 평가하기 위하여, MNIST 및 CIFAR-10 데이터 셋용으로 설계된 딥러닝 네트워크를 사용하였다. 상기 딥러닝 네트워크는 MLP(Multi-Layer Perceptron) 및 3배 인플레이션 계수를 갖는 VGG 모델이다.

[0059] 보정 데이터 셋의 경우 임의의 데이터 셋에서 무작위로 선택된 하위 집합을 사용하였다. 본원발명의 실시예는 고장률(Fault Rate; FR) 및 고착-폐쇄 오류와 고착-폐쇄 오류 간의 비율 또는 개방-폐쇄 비율(OCR)을 다양하게 변경하였다. 예컨대 고장률(FR)=10%이고 개방-폐쇄 비율(OCR)=4이면 목표 저항값에 관계없이 저항성 메모리 소자(ReRAM)의 약 8%가 최고상태(HRS)에 고착되고, 약 2%가 최저상태(LRS)에 고착될 것으로 예상될 수 있다.

[0060] 즉, 본원발명의 실시예는 고장률(FR)을 10%에서 40%까지 다양하게 하고 개방-폐쇄 비율(OCR)을 5, 1, 1/5를 사용하였다. 기준 정확도는 MNIST의 경우 98.00%, CIFAR-10의 경우 91.27%이다. 기준 정확도는 고착 고장(Stuck-At Fault; SAF)이 발생되지 않을 경우 각 BNN 모델의 테스트 정확도이다.

[0061] 본원발명의 실시예에서는 고장률(FR)에 따라서 재훈련 없음(Simple Inference; SI)과 본원발명의 비학습 파라미터 튜닝방식(Free Parameter Tuning; FPT)이 비교되었다.

[0063] *실험결과*

[0064] 하기 [표 1]을 보면, 본원발명의 실시예에 따라 딥러닝 네트워크에 MNIST 및 CIFAR-10 데이터 셋이 이용된 실험결과는 고장률(FR)이 10%인 경우 구 데이터 셋 모두에서 기준 정확도에 가장 근접한 정확도가 도출된 것을 확인할 수 있다. 그리고 고장률(FR)이 높아질수록 보상 또는 재훈련 없음(Simple Inference; SI)보다 정확도가 상당히 높은 것을 알 수 있다.

[0065] 즉, 상기 언급한 것과 같이 본원발명의 딥러닝 네트워크 내 배치 정규화(BN) 레이어는 평균(μ)과 표준편차(σ)가 업데이트된 후 컨볼루션(Convolution) 레이어 또는 완전 연결(Fully-Connected) 레이어에 폴딩(folding)되어 제거됨으로써, 추가적인 하드웨어 및 별도의 훈련을 위한 방대한 양의 임의의 데이터 셋이 필요하지 않는다. 그리고 고장률(FR)이 높아질수록 보상 또는 재훈련을 하지 않을 때 보다 기준 정확도에 매우 근접

하거나, 월등히 높은 정확도를 보이므로 상당히 효율적인 고착 고장(SAF) 보상 방법인 것이 증명될 수 있다.

*표 1*

[0066]

| MNIST test accuracy | | |
|---|---|---|
| FR | SI | FPT |
| 10% | 97.36% | 97.47% |
| 20% | 91.85% | 97.21% |
| 40% | 44.23% | 95.07% |
| CIFAR-10 test accuracy | | |
| FR | SI | FPT |
| 10% | 78.18% | 88.83% |
| 20% | 32.46% | 86.08% |
| 40% | 10.08% | 66.30% |

[0067] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술 분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 으로 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.

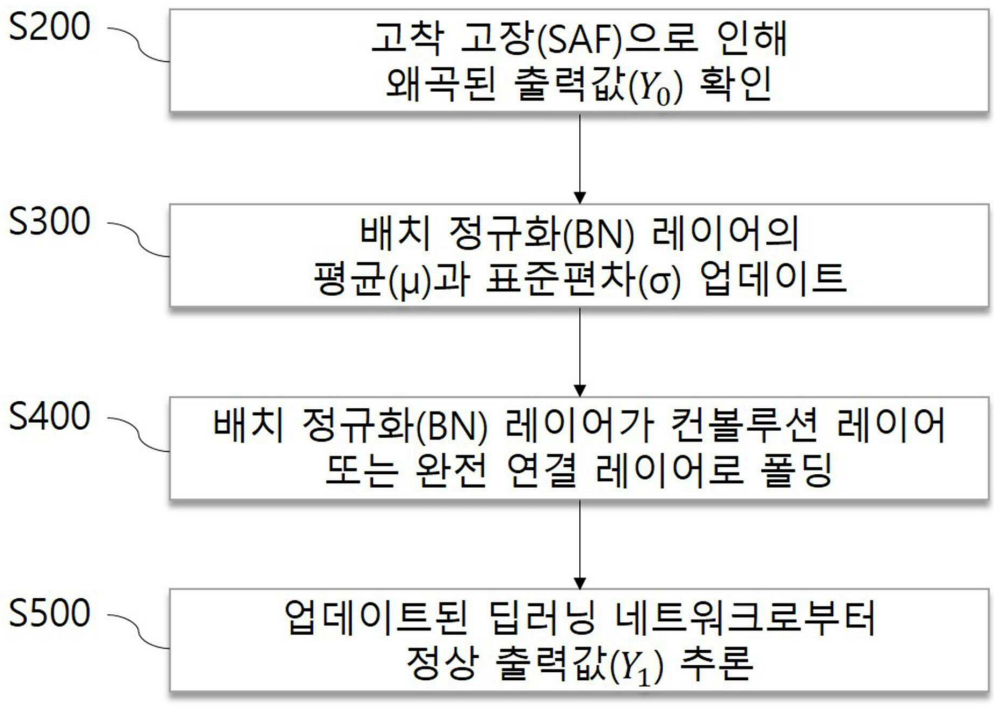
[0068] 그러므로 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

## *부호의 설명*

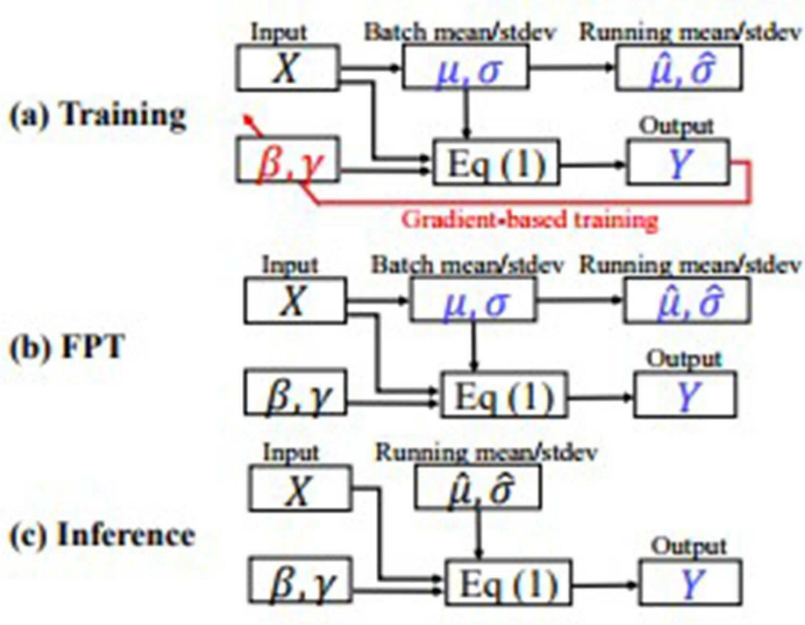
[0069] 100.. 저항성 메모리 소자(ReRAM) 기반의 딥러닝 가속기 하드웨어

## *도면*

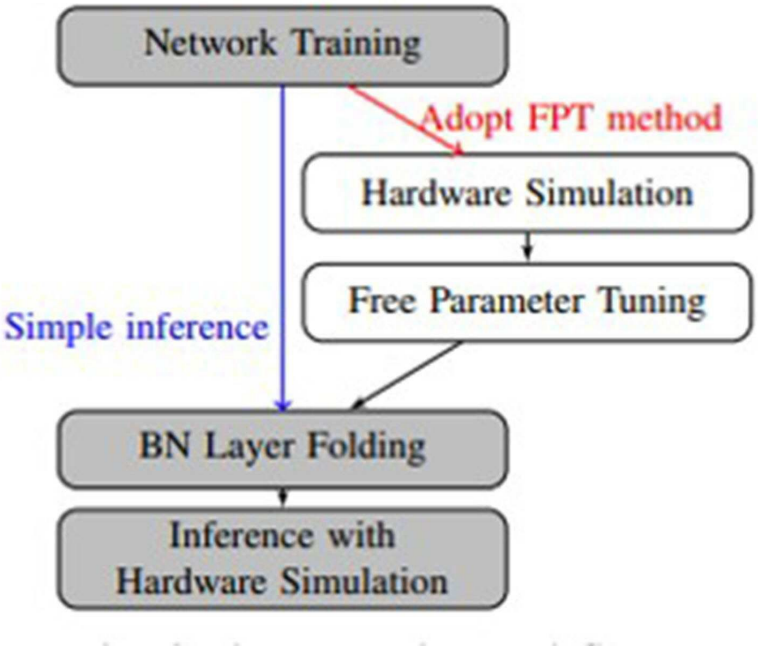
### *도면1*



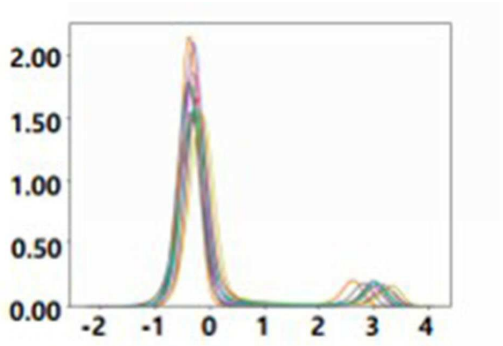
도면2


(a) Training
Input
Batch mean/stdev
Running mean/stdev
X
μ, σ
μ̂, σ̂
β, γ
Eq (1)
Output
Y
Gradient-based training
(b) FPT
Input
Batch mean/stdev
Running mean/stdev
X
μ, σ
μ̂, σ̂
β, γ
Eq (1)
Output
Y
(c) Inference
Input
Running mean/stdev
X
μ̂, σ̂
β, γ
Eq (1)
Output
Y

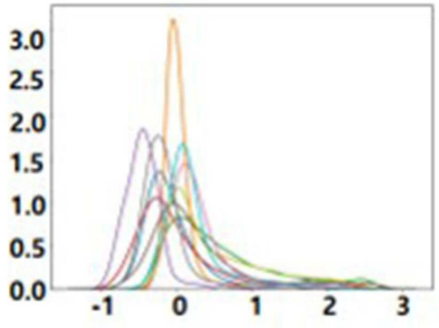
도면3


Network Training
Adopt FPT method
Hardware Simulation
Free Parameter Tuning
Simple inference
BN Layer Folding
Inference with
Hardware Simulation

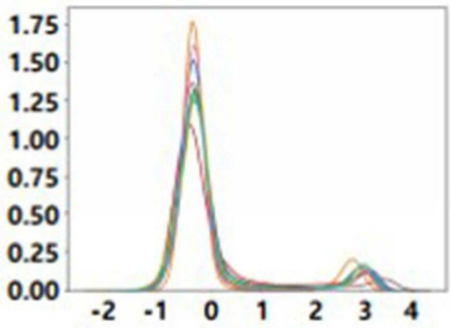
도면4




(a) Without SAF




(b) With SAF applied




(c) With SAF and FPT applied.