



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0069461
(43) 공개일자 2022년05월27일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/082 (2013.01)
G06N 3/04 (2013.01)
(21) 출원번호 10-2020-0156600
(22) 출원일자 2020년11월20일
심사청구일자 2020년11월20일

(71) 출원인
한림대학교 산학협력단
강원도 춘천시 한림대학길 1, 한림대학교(옥천동)
(72) 발명자
이정근
강원도 춘천시 서부대성로 34, 103동 1301호(요선동, 한신휴플러스아파트)
우윤희
서울특별시 강동구 천중로40길 45, 901호(길동, SM베스트빌)
(74) 대리인
특허법인본

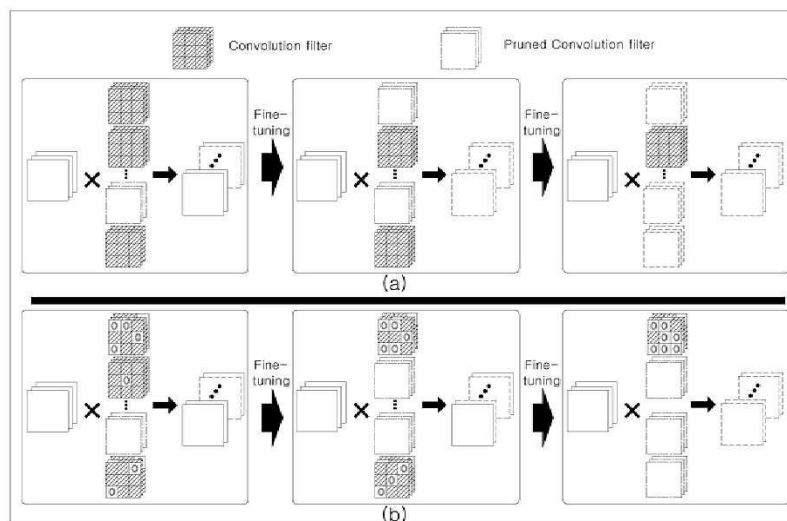
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법 및 프로그램

(57) 요약

인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법이 개시된다. 상기 제어 방법은, 기학습된 인공지능 모델의 레이어에 포함된 복수의 필터를 획득하는 단계; 상기 복수의 필터 중, 제1 필터에 포함된 복수의 파라미터를 획득하는 단계; 상기 복수의 파라미터 각각에 대한 복수의 절대값을 획득하는 단계; 상기 복수의 절대값을 크기 순서로 정렬하는 단계; 상기 획득된 복수의 절대값 중 기 설정된 하위 m% 비율에 해당하는 적어도 하나의 파라미터를 획득하는 단계; 및 상기 획득된 적어도 하나의 파라미터의 값을 0으로 치환하는 단계; 를 포함한다.

대표도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1345316268
과제번호	2018R1D1A1B07043399
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학개인지초연구지원사업(기본연구)
연구과제명	고신뢰 시스템을 위한 저전자파/결함-감내 디지털 회로 설계
기 여 율	1/1
과제수행기관명	한림대학교 산학협력단
연구기간	2020.03.01 ~ 2021.02.28

명세서

청구범위

청구항 1

인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법에 있어서,
 기학습된 인공지능 모델의 레이어에 포함된 복수의 필터를 획득하는 단계;
 상기 복수의 필터 중, 제1 필터에 포함된 복수의 파라미터를 획득하는 단계;
 상기 복수의 파라미터 각각에 대한 복수의 절대값을 획득하는 단계;
 상기 복수의 절대값을 크기 순서로 정렬하는 단계;
 상기 획득된 복수의 절대값 중 기 설정된 하위 $m\%$ 비율에 해당하는 적어도 하나의 파라미터를 획득하는 단계;
 상기 획득된 적어도 하나의 파라미터의 값을 0으로 치환하는 단계; 를 포함하는 제어 방법.

청구항 2

제1항에 있어서,
 상기 제어 방법은,
 상기 복수의 필터 각각에 대한 제로 파라미터의 개수를 판단하는 단계;
 상기 복수의 필터 각각을 상기 복수의 필터가 포함하는 제로 파라미터의 개수를 기준으로 정렬하는 단계;
 상기 제1 필터가 상기 정렬된 복수의 필터 중 기 설정된 하위 $n\%$ 비율에 해당하는 경우, 상기 제1 필터를 프루닝하는 단계; 및
 상기 인공지능 모델을 파인 튜닝하는 단계; 를 포함하는 제어 방법.

청구항 3

제2항에 있어서,
 상기 인공지능 모델은 13-Layer CNN 모델이고,
 상기 레이어는 상기 인공지능 모델이 포함하는 13개의 레이어 중, 제5 레이어 내지 제13레이어인 것을 특징으로 하는 제어 방법.

청구항 4

제2항에 있어서,
 상기 제어 방법은,
 상기 프루닝된 인공지능 모델에 대응되는 필터를 바탕으로 상기 인공지능 모델 구현을 위한 연산을 수행하는 단계; 를 더 포함하고,
 상기 연산을 수행하는 단계는,
 상기 프루닝된 인공지능 모델에 대응되는 필터로 입력되는 제1 입력값 및 제2 입력값이 제로값을 포함하는지 판단하는 단계; 및
 상기 제1 입력값 및 상기 제2 입력값 중 적어도 하나의 입력값이 제로값을 포함하는 경우, 상기 연산을 스킵하는 단계; 를 포함하는 제어 방법.

청구항 5

제4항에 있어서,

상기 연산을 수행하는 단계는,

상기 제1 입력값을 제1 레지스터에 입력하고, 상기 제2 입력값을 상기 제2 레지스터에 입력하는 단계;

상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 1인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값을 곱연산하는 단계; 및

상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 0인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값에 대한 곱연산을 스킵하는 단계;를 포함하는 제어 방법.

청구항 6

제5항에 있어서,

상기 연산을 스킵하는 단계는,

상기 제1 입력값 및 상기 제2 입력값이 ZD 회로에 입력하여, 제1-1 출력값 및 제2-1 출력값을 획득하는 단계;

상기 제1-1 출력값 및 상기 제2-1 출력값을 OR 게이트에 입력하여 생성된 제3 출력값을 제3 레지스터의 입력값으로 입력하는 단계; 및

상기 제3 출력값의 반전값을 상기 제1 레지스터 및 상기 제2 레지스터의 en값으로 입력하는 단계;를 포함하는 제어 방법.

청구항 7

제6항에 있어서,

상기 ZD 회로는 NOT 게이트로 구현되는 것을 특징으로 하는 제어 방법.

청구항 8

하나 이상의 인스트럭션을 저장하는 메모리; 및

상기 메모리에 저장된 상기 하나 이상의 인스트럭션을 실행하는 프로세서를 포함하고,

상기 프로세서는 상기 하나 이상의 인스트럭션을 실행함으로써,

제1 항의 방법을 수행하는, 장치.

청구항 9

하드웨어인 컴퓨터와 결합되어, 제1 항의 방법을 수행할 수 있도록 컴퓨터에서 독출가능한 기록매체에 저장된 컴퓨터프로그램.

청구항 10

인공지능 모델의 필터 연산을 위한 전자 장치의 제어 방법에 있어서,

상기 인공지능 모델에 대응되는 필터로 입력되는 제1 입력값 및 제2 입력값이 제로값을 포함하는지 판단하는 단계;

상기 제1 입력값을 제1 레지스터에 입력하고, 상기 제2 입력값을 상기 제2 레지스터에 입력하는 단계;

상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 1인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값을 곱연산하는 단계; 및

상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 0인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값에 대한 곱연산을 스킵하는 단계;를 포함하는 제어 방법.

발명의 설명

기술 분야

[0001] 본 발명은 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법 및 프로그램에 관한 것이다.

배경 기술

- [0002] 인공 신경망의 입력 데이터는 수많은 연산을 거치게 된다. 대부분의 연산은 필터와 이루어지며 신경망의 깊이가 깊을수록 필터의 수가 증가하기 때문에 연산량 또한 증가한다.
- [0003] 인공 신경망의 깊이가 깊어질수록 성능이 향상된다는 것은 이미 많은 연구들에 의해 증명된 사실이며, 대다수의 최신 인공 신경망 모델들은 이 사실에 기반하여 깊은 층을 갖고 이에 따라 높은 연산량을 갖게된다.
- [0004] 증가한 연산량은 최신 하드웨어 또는 인공 신경망 동작을 위한 전자 장치에서는 큰 영향을 미치지 않지만 스마트폰, IoT 디바이스 등과 같은 저사양/저성능 전자 장치의 경우 연산력의 제한이 존재하게된다.
- [0005] 이에, 상술한 저사양/저성능 전자 장치에서 인공신경망 기술을 적용시키기 위한 다양한 연구가 진행되고 있다. 예를 들어, 프루닝, 양자화, 이진화, 지식 증류, 적은 파라미터를 가지는 모델 설계 등 다양한 인공지능 모델 압축방법이 제시되고 있으며, 최적의 압축 방법을 찾기 위한 연구들이 계속하여 진행되고 있는 추세이다.
- [0006] 상기 다양한 연구 중, 모델의 크기를 줄이는 대표적인 방법에는 프루닝 기법이 이용된다. 프루닝은 모델 내의 중요도가 낮다고 판단되는 원소를 제거하여 연산량을 줄이는 방법으로 연산량을 줄일 뿐만 아니라 때로는 기존 모델보다 정확도가 향상될 수 있다는 연구 결과 또한 존재한다.
- [0007] 본 발명은, 상술한 프루닝 모델을 보완하기 위해 안출된 것으로서, 종래 프루닝 대비 더 나은 연산속도를 보장하기 위한 것이다.

선행기술문헌

특허문헌

[0008] (특허문헌 0001) 등록특허공보 제10-2165273호, 2020.10.13

발명의 내용

해결하려는 과제

- [0009] 본 발명이 해결하고자 하는 과제는 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법 및 프로그램을 제공하는 것이다.
- [0010] 본 발명이 해결하고자 하는 과제들은 이상에서 언급된 과제로 제한되지 않으며, 언급되지 않은 또 다른 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

- [0011] 상술한 과제를 해결하기 위한 본 발명의 일 면에 따른 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법은, 기학습된 인공지능 모델의 레이어에 포함된 복수의 필터를 획득하는 단계; 상기 복수의 필터 중, 제1 필터에 포함된 복수의 파라미터를 획득하는 단계; 상기 복수의 파라미터 각각에 대한 복수의 절대값을 획득하는 단계; 상기 복수의 절대값을 크기 순서로 정렬하는 단계; 상기 획득된 복수의 절대값 중 기 설정된 하위 m% 비율에 해당하는 적어도 하나의 파라미터를 획득하는 단계; 및 상기 획득된 적어도 하나의 파라미터의 값을 0으로 치환하는 단계;를 포함한다.
- [0012] 이때, 상기 제어 방법은, 상기 복수의 필터 각각에 대한 제로 파라미터의 개수를 판단하는 단계; 상기 복수의 필터 각각을 상기 복수의 필터가 포함하는 제로 파라미터의 개수를 기준으로 정렬하는 단계; 상기 제1 필터가 상기 정렬된 복수의 필터 중 기 설정된 하위 n% 비율에 해당하는 경우, 상기 제1 필터를 프루닝하는 단계; 및 상기 인공지능 모델을 파인 튜닝하는 단계;를 포함할 수 있다.
- [0013] 이때, 상기 인공지능 모델은 13-Layer CNN 모델이고, 상기 레이어는 상기 인공지능 모델이 포함하는 13개의 레

이어 중, 제5 레이어 내지 제13레이어일 수 있다.

[0014] 이때, 상기 제어 방법은, 상기 프루닝된 인공지능 모델에 대응되는 필터를 바탕으로 상기 인공지능 모델 구현을 위한 연산을 수행하는 단계; 를 더 포함하고, 상기 연산을 수행하는 단계는, 상기 프루닝된 인공지능 모델에 대응되는 필터로 입력되는 제1 입력값 및 제2 입력값이 제로값을 포함하는지 판단하는 단계; 및 상기 제1 입력값 및 상기 제2 입력값 중 적어도 하나의 입력값이 제로값을 포함하는 경우, 상기 연산을 스킵하는 단계;를 포함할 수 있다.

[0015] 이때, 상기 연산을 수행하는 단계는, 상기 제1 입력값을 제1 레지스터에 입력하고, 상기 제2 입력값을 상기 제2 레지스터에 입력하는 단계; 상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 1인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값을 곱연산하는 단계; 및 상기 제1 레지스터 및 상기 제2 레지스터에 입력되는 en값이 0인 경우, 상기 제1 레지스터에서 출력된 제1 출력값 및 상기 제2 레지스터에서 출력된 제2 출력값에 대한 곱연산을 스킵하는 단계;를 포함할 수 있다.

[0016] 이때, 상기 연산을 스킵하는 단계는, 상기 제1 입력값 및 상기 제2 입력값이 ZD 회로에 입력하여, 제1-1 출력값 및 제2-1 출력값을 획득하는 단계; 상기 제1-1 출력값 및 상기 제2-1 출력값을 OR 게이트에 입력하여 생성된 제3 출력값을 제3 레지스터의 입력값으로 입력하는 단계; 및 상기 제3 출력값의 반전값을 상기 제1 레지스터 및 상기 제2 레지스터의 en값으로 입력하는 단계; 를 포함할 수 있다.

[0017] 이때, 상기 ZD 회로는 NOT 게이트로 구현될 수 있다.

[0018] 본 발명의 기타 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

발명의 효과

[0019] 상술한 본 발명의 다양한 실시예에 따라 제안된 인공지능 모델은 종래 대비 연산량이 감소하는 효과가 존재한다.

[0020] 본 발명의 효과들은 이상에서 언급된 효과로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0021] 도 1은 본 발명의 일 실시예에 따른 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법을 설명하기 위한 예시도이다.

도 2는 본 발명의 일 실시예에 따른 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법을 설명하기 위한 흐름도이다.

도 3은 본 발명의 일 실시예에 따른 제로 스킵 방법을 설명하기 위한 회로도이다.

도 4 내지 도 8은 본 발명의 실시예에 따른 인공지능 모델의 필터 변환 후 결과를 설명하기 위한 예시도이다.

도 9는 본 발명의 일 실시예에 따른 장치의 구성도이다.

발명을 실시하기 위한 구체적인 내용

[0022] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 제한되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야의 통상의 기술자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0023] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다. 명세서 전체에 걸쳐 동일한 도면 부호는 동일한 구성 요소를 지칭하며, "및/또는"은 언급된 구성요소들의 각각 및 하나 이상의 모든 조합을 포함한다. 비록 "제1", "제2" 등이 다양한 구성요소들을 서술하기 위해서 사용되나, 이들 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 구성요소를 다른 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 구성

요소는 본 발명의 기술적 사상 내에서 제2 구성요소일 수도 있음은 물론이다.

- [0024] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술 분야의 통상의 기술자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.
- [0025] 명세서에서 사용되는 "부" 또는 "모듈"이라는 용어는 소프트웨어, FPGA 또는 ASIC과 같은 하드웨어 구성요소를 의미하며, "부" 또는 "모듈"은 어떤 역할들을 수행한다. 그렇지만 "부" 또는 "모듈"은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. "부" 또는 "모듈"은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 "부" 또는 "모듈"은 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다. 구성요소들과 "부" 또는 "모듈"들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 "부" 또는 "모듈"들로 결합되거나 추가적인 구성요소들과 "부" 또는 "모듈"들로 더 분리될 수 있다.
- [0026] 공간적으로 상대적인 용어인 "아래(below)", "아래(beneath)", "하부(lower)", "위(above)", "상부(upper)" 등은 도면에 도시되어 있는 바와 같이 하나의 구성요소와 다른 구성요소들과의 상관관계를 용이하게 기술하기 위해 사용될 수 있다. 공간적으로 상대적인 용어는 도면에 도시되어 있는 방향에 더하여 사용시 또는 동작시 구성요소들의 서로 다른 방향을 포함하는 용어로 이해되어야 한다. 예를 들어, 도면에 도시되어 있는 구성요소를 뒤집을 경우, 다른 구성요소의 "아래(below)"또는 "아래(beneath)"로 기술된 구성요소는 다른 구성요소의 "위(above)"에 놓여질 수 있다. 따라서, 예시적인 용어인 "아래"는 아래와 위의 방향을 모두 포함할 수 있다. 구성요소는 다른 방향으로도 배향될 수 있으며, 이에 따라 공간적으로 상대적인 용어들은 배향에 따라 해석될 수 있다.
- [0027] 본 명세서에서, 컴퓨터는 적어도 하나의 프로세서를 포함하는 모든 종류의 하드웨어 장치를 의미하는 것이고, 실시 예에 따라 해당 하드웨어 장치에서 동작하는 소프트웨어적 구성도 포괄하는 의미로서 이해될 수 있다. 예를 들어, 컴퓨터는 스마트폰, 태블릿 PC, 데스크톱, 노트북 및 각 장치에서 구동되는 사용자 클라이언트 및 애플리케이션을 모두 포함하는 의미로서 이해될 수 있으며, 또한 이에 제한되는 것은 아니다.
- [0028] 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다.
- [0029] 본 명세서에서 설명되는 각 단계들은 컴퓨터에 의하여 수행되는 것으로 설명되나, 각 단계의 주체는 이에 제한되는 것은 아니며, 실시 예에 따라 각 단계들의 적어도 일부가 서로 다른 장치에서 수행될 수도 있다.
- [0030] 도 1은 본 발명의 일 실시예에 따른 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법을 설명하기 위한 예시도이다.
- [0031] 구체적으로, 도 1(a)는 기존의 프루닝 방법을 설명하고 있다. 즉, 전자 장치(100)는 인공지능 레이어에 포함된 복수의 필터를 단순 프루닝 하여 필터 내의 0의 개수를 증가시킬 수 있다. 이때, 전자 장치(100)는 필요에 따라 프루닝 레이트를 조정하여 필터 내의 0의 값을 조절할 수 있다.
- [0032] 반면 도 1(b)에 도시된 바와 같이, 본 발명에 따른 필터 변환 방법은, 필터 내의 논제로 값에 대하여 기 설정된 조건을 만족하는 경우, 해당 논제로 값을 0으로 치환하고, 상기 방법이 적용된 복수의 필터 중 0의 개수가 적은 필터를 프루닝함으로써, 레이어에 포함된 필터의 0의 개수를 최대화하여 연산 과정을 단축시킬 수 있다.
- [0033] 한편, 본 발명에서 사용된 프루닝 방법은 필터 프루닝으로 인공 신경망 모델에서 중요성이 낮다고 판단되는 필터를 제거하는 방법일 수 있다. 일반적으로 각 필터의 l1-norm 또는 l2-norm을 계산하여 그 값이 작으면 중요도가 낮은 필터로 간주하여 제거한다.
- [0034] 다만, 기존 연구에서 제거될 필터를 선택하는 기준이 인공 신경망의 성능에 큰 영향을 미치지 않는다는 연구 결과가 발표되었으며, 본 발명은 제거될 필터를 선택하는 기준이 인공 신경망의 성능에 큰 영향을 미치지 않음을 전제로, 필터 내의 0의 개수가 적은 필터를 제거하여 0의 개수가 많은 필터를 남기는 새로운 프루닝 방법을 제안한다.
- [0035] 도 2는 본 발명의 일 실시예에 따른 인공지능 모델의 필터 변환을 위한 전자 장치의 제어 방법을 설명하기 위한

흐름도이다.

- [0036] 단계 S110에서, 전자 장치(100)는 기학습된 인공지능 모델의 레이어에 포함된 복수의 필터를 획득할 수 있다.
- [0037] 단계 S120에서, 전자 장치(100)는, 복수의 필터 중, 제1 필터에 포함된 복수의 파라미터를 획득할 수 있다.
- [0038] 단계 S130에서, 전자 장치(100)는, 복수의 파라미터 각각에 대한 복수의 절대값을 획득할 수 있다.
- [0039] 단계 S140에서, 전자 장치(100)는, 복수의 절대값을 크기 순서로 정렬할 수 있다.
- [0040] 단계 S150에서, 전자 장치(100)는, 획득된 복수의 절대값 중 기 설정된 하위 $m\%$ 비율에 해당하는 적어도 하나의 파라미터를 획득할 수 있다.
- [0041] 단계 S160에서, 전자 장치(100)는, 획득된 적어도 하나의 파라미터의 값을 0으로 치환할 수 있다.
- [0042] 일 실시예로, 상기 m 은 5일 수 있다. 즉, 전자 장치(100)는 필터 내 복수의 파라미터 중 절대값이 작은 하위 5%의 파라미터를 0으로 치환할 수 있다.
- [0043] 이후, 전자 장치(100)는 복수의 필터 각각에 대한 제로 파라미터의 개수를 판단할 수 있다.
- [0044] 예컨대, 전자 장치(100)는 제1 필터가 포함하는 제로 파라미터의 개수, 제2 필터가 포함하는 제로 파라미터의 개수, 제3 필터가 포함하는 제로 파라미터의 개수를 획득할 수 있다. 즉, 전자 장치(100)는 레이어에 포함된 모든 필터 각각의 제로 파라미터의 개수를 획득할 수 있다.
- [0045] 전자 장치(100)는, 복수의 필터 각각을 복수의 필터가 포함하는 제로 파라미터의 개수를 기준으로 정렬할 수 있다.
- [0046] 전자 장치(100)는, 제1 필터가 정렬된 복수의 필터 중 기 설정된 하위 $n\%$ 비율에 해당하는 경우, 제1 필터를 프루닝할 수 있다.
- [0047] 일 실시예로, 상기 n 은 5일 수 있다. 즉, 전자 장치(100)는 복수의 필터 중 제로 파라미터의 개수가 적은 하위 5%의 필터에 대하여만 프루닝을 수행할 수 있다.
- [0048] 전자 장치(100)는, 인공지능 모델을 파인 튜닝할 수 있다.
- [0049] 전자 장치는 상기 단계 S110 내지 S160, 및 프루닝 단계를 반복하여 레이어에 포함된 필터의 파라미터를 0으로 치환할 수 있다.
- [0050] 한편, 일 실시예로, 인공지능 모델은 13-Layer CNN 모델일 수 있다. 다만, 본 발명의 다양한 실시예에 의할 때, 인공지능 모델은 20-Layer CNN 모델일 수 있으며, 레이어의 개수에 제한되지 않음은 물론이다.
- [0051] 이때, 인공지능 모델은 13-Layer CNN 모델인 경우, 전자 장치(100)는 인공지능 모델이 포함하는 13개의 레이어 중, 제5 레이어 내지 제13레이어에 대해 상술한 단계를 수행할 수 있다. 즉, 전자 장치(100)는 제1 레이어 내지 제4 레이어에 대하여는 프루닝을 수행하지 않고, 제5 레이어 내지 제13 레이어만을 프루닝할 수 있다. 다만, 전자 장치(100)는 인공지능 모델이 포함하는 모든 레이어에서 프루닝을 수행할 수 있음은 물론이다.
- [0052] 후술하는 도 4 및 도 5에 의한 설명과 같이, 전자 장치(100)가 제5 레이어 내지 제13 레이어만을 프루닝하는 경우 정확도가 가장 높음을 확인할 수 있다.
- [0053] 한편, 본 발명의 일 실시예로, 전자 장치(100)는, 프루닝된 인공지능 모델에 대응되는 필터를 바탕으로 인공지능 모델 구현을 위한 연산을 수행할 수 있다.
- [0054] 구체적으로, 인공지능 모델은 많은 파라미터가 존재하고, 이러한 많은 파라미터의 곱연산을 수행하는 과정을 거치게 되어 연산량이 증가하게 된다. 따라서, 불필요한 곱연산을 생략할 수 있다면, 인공지능 모델이 저성능의 전자 장치에서도 원활히 작동할 수 있게 된다.
- [0055] 따라서, 곱연산이 적용되는 입력값 중 어느 하나가 0을 포함하고 있는 경우, 입력값들의 곱연산을 수행하는 연산량보다, 입력값이 0인지 여부를 판단하는 것이 연산량이 더 적은 경우가 존재한다.
- [0056] 따라서, 전자 장치(100)는, 프루닝된 인공지능 모델에 대응되는 필터로 입력되는 제1 입력값 및 제2 입력값이 제로값을 포함하는지 판단할 수 있다.
- [0057] 이후, 전자 장치(100)는, 제1 입력값 및 제2 입력값 중 적어도 하나의 입력값이 제로값을 포함하는 경우, 연산

을 스킵할 수 있다.

- [0058] 일 실시예에 따라, 전자 장치(100)는 제1 입력값 및 제2 입력값이 기 설정된 비트 이상인 값을 가지는 경우에 한하여 연산을 스킵할 수 있다. 이때, 기 설정된 비트는 예를들어 32비트 또는 64비트일 수 있다.
- [0059] 즉, 전자 장치(100)는 곱연산에 소요되는 연산량과 입력값이 0임을 판단하는데 소요되는 연산량을 비교한 후, 입력값이 0임을 판단하는데 소요되는 연산량이 더 적은 경우에 한하여 스킵 과정을 수행할 수 있다.
- [0060] 구체적으로, 도 3에 도시된 바와 같이, 전자 장치(100)는, 제1 입력값을 제1 레지스터(320)에 입력하고, 제2 입력값을 제2 레지스터(330)에 입력할 수 있다.
- [0061] 이후, 전자 장치(100)는, 제1 레지스터(320) 및 제2 레지스터(330)에 입력되는 en값이 1인 경우, 제1 레지스터(320)에서 출력된 제1 출력값 및 제2 레지스터(330)에서 출력된 제2 출력값을 곱연산할 수 있다. 곱연산은 도 3에 도시된 곱셈기(370)에 의해 수행될 수 있음은 물론이다.
- [0062] 구체적으로, 레지스터는 입력 포트, 출력포트, 클럭신호 인가 포트 및 en 포트를 포함할 수 있다. 클럭 신호는 기 설정된 주기마다 레지스터로 인가되며, 레지스터는 en 포트에 입력되는 값에 따라 입력 포트에 입력되는 입력 신호를 저장할지 여부를 판단할 수 있다.
- [0063] 일 실시예로, en값이 1인 경우, 레지스터는 클럭 신호가 인가됨과 함께 입력값을 저장할 수 있으며, 반대로, en값이 0인 경우, 레지스터는 클럭 신호가 인가됨과 함께 입력값을 저장하지 않을 수 있다.
- [0064] 이후, 전자 장치(100)는, 제1 레지스터 및 제2 레지스터에 입력되는 en값이 0인 경우, 제1 레지스터에서 출력된 제1 출력값 및 제2 레지스터에서 출력된 제2 출력값에 대한 곱연산을 스킵할 수 있다.
- [0065] 구체적으로, 전자 장치(100)는, 제1 입력값 및 제2 입력값이 ZD 회로(310)에 입력하여, 제1-1 출력값 및 제2-1 출력값을 획득할 수 있다.
- [0066] ZD 회로(310)는 제1 입력값 및 제2 입력값 중 어느 하나의 입력값이 0인 경우, 1의 값을 출력하기 위한 회로이다. 나아가, ZD 회로(310)는 제1 입력값 및 제2 입력값이 모두 0이 아닌 경우, 0을 출력할 수 있다. 일 실시예로, 제1 입력값 및 제2 입력값이 1비트 신호인 경우, ZD 회로(310)는 NOT 게이트로 구현될 수 있다. 다만, 본 발명의 다양한 실시예에 따라, 제1 입력값 및 제2 입력값은 2비트 이상인 경우가 존재한다. 이 경우, ZD 회로(310)는 제1 입력값 및 제2 입력값의 모든 비트가 0인 경우, 1을 출력하도록 구현될 수 있다.
- [0067] 이후, 전자 장치(100)는, 제1-1 출력값 및 제2-1 출력값을 OR 게이트에 입력하여 생성된 제3 출력값을 제3 레지스터(340)의 입력값으로 입력할 수 있다.
- [0068] 즉 제3 출력값은 제1-1 출력값 및 제2-1 출력값 중 적어도 하나의 출력값이 1을 포함하는 경우, 제3 출력값을 1로 출력할 수 있다.
- [0069] 이후, 전자 장치(100)는, 제3 출력값의 반전값을 제1 레지스터(320) 및 제2 레지스터(330)의 en값으로 입력할 수 있다. 상술한 바와 같이, en값이 0인 경우, 입력값이 레지스터에 저장되지 않으므로, 제1 레지스터(320) 및 제2 레지스터(330)의 en 포트 상단(또는 이전 단계)에는 NOT 게이트가 연결될 수 있다.
- [0070] 즉, ZD 회로(310)는 제1 입력값 및 제2 입력값의 비트에 1이 포함된 경우 0을 출력값으로 출력하고, 제1 입력값 및 제2 입력값의 비트가 모두 0인 경우 1을 출력할 수 있다.
- [0071] 한편, 전자 장치(100)는 제3 출력값을 제3 레지스터에(340) 저장할 수 있다. 이 경우, 제3 레지스터(340)의 en값은 항상 1일 수 있다.
- [0072] 전자 장치(100)는 제3 레지스터(340)의 출력값을 멀티플렉서(350)의 결정자 값으로 입력할 수 있다. 구체적으로 멀티플렉서(350)는 결정자 값(제3 레지스터(340)의 출력값)이 0인 경우 제1 입력값 및 제2 입력값의 곱연산을 수행한 결과값을 출력하고, 1인 경우 0을 출력한다.
- [0073] 즉, 도 3에 도시된 회로도에 의할 때, 제1 입력값 및 제2 입력값 중 어느 하나가 0의 값을 가지는 경우, 제4 레지스터(360)는 곱연산 과정 없이 0의 값을 저장할 수 있다.
- [0074] 이하에서는 도 4 내지 도 8을 이용하여 종래 단순 프루닝을 적용한 인공지능 모델과 본 발명의 실시예에 따른 인공지능 모델의 정확도의 차이를 설명한다.
- [0075] 구체적으로, 도 4 내지 도 8은 상술한 m 및 n의 값을 5로 설정하고, 프루닝 수행 레이어를 1) 1 내지 13레이어

2) 5 내지 13 레이어 3) 1 내지 11 레이어 및 4) 5 내지 11 레이어로 설정한 실험에 대한 결과 데이터이다.

- [0076] 도 4는 프루닝될 필터를 무작위로 선정하는 경우의 결과이고, 도 5는 본 발명의 실시예에 따른 프루닝 필터를 선택한 경우의 결과이다. 결과의 정확성을 위하여 도 4의 실험에서 프루닝된 후의 필터가 포함하는 총 0의 값은 변경되지 않는다.
- [0077] 도 4 내지 도 6에 도시된 바와 같이, 단순 프루닝의 경우보다, 치환할 0의 값을 절대값 기준으로 결정하는 것이 더 높은 정확도를 가지는 것을 확인할 수 있다.
- [0078] 나아가 도 4 내지 도 6는 모두 1~4 레이어를 프루닝하지 않았을 때 더 높은 효율을 보이는 것을 확인할 수 있다. 따라서, 1~4 레이어를 프루닝하지 않은 결과만을 추출하여 비교하였을 때, 도 6에 도시된 바와 같이 본 발명의 실시예에 따라 5~13 레이어를 프루닝한 결과가 전범위의 NZER_ORIN(none zero origin, 즉 원래 필터에서의 논제로 값의 비율) 평균적으로 우수하고 안정적인 성능(Zero 5~13 참조)을 보여주는 것을 확인할 수 있다.
- [0079] 한편, 필터의 개수 또한 연산량 감소에 있어서 중요한 부분이다. 따라서, 도 7에 도시된 바와 같이, 본 발명에 의할 경우, 5~13 레이어에 본 발명의 실시예를 적용하는 경우, 필터의 개수와 무관하게 가장 안정적인 성능을 보이는 것을 확인할 수 있다.
- [0080] 도 8은 기존의 방법으로 프루닝을 적용하였을 때의 결과 및 본 발명에 따라 프루닝을 적용하였을 때의 결과를 설명하기 위한 표이다.
- [0081] 도 8에 도시된 바와 같이, 랜덤으로 15% 프루닝한 모델과 본 발명의 실시예를 적용한 후 20% 프루닝한 모델의 정확도가 유사함을 확인할 수 있다.
- [0082] 즉, 본 발명의 경우, 더 높은 프루닝 레이트를 적용하더라도, 기존의 프루닝과 유사한 정확도를 가질 수 있으며, 이는, 더 많은 0의 개수와 더 적은 필터를 가지고 유사한 정확도를 가지는 인공지능 모델을 생성할 수 있다는 의미로, 본 발명의 실시예에 의할 경우, 종래 대비 유사한 정확도를 가지며 연산량이 감소되는 효과가 발생한다.
- [0083] 구체적으로, 15% 종래 프루닝 모델의 경우, 오리지널 모델의 72.52%만큼의 논제로 값을 가지며 필터의 수는 3598개이나, 본 발명에 따른 20% 프루닝 모델의 경우, 오리지널 모델의 47.51%의 논제로 값을 가지며 필터의 수는 3384개 이게 된다. 나아가, 40% 프루닝한 모델에서도 같은 결과를 확인할 수 있다. 두 모델의 정확도는 91.45와 91.50으로 비슷하지만 0이 아닌 원소의 비율이 36.20와 10.79로 큰 차이를 보여준다.
- [0084] 결론적으로, 본 발명에 의할 경우, 종래 기술에 비해 유사한 정확도를 가지더라도 0의 값이 증가하여 연산량이 감소하고, 나아가, 0의 값을 포함하는 경우, 연산과정을 스킵하여 더 빠른 연산이 가능해지는 새로운 효과가 존재한다.
- [0085] 한편, 본 발명에 따른 인공지능 기술은 기계학습(딥러닝) 및 기계학습을 활용한 요소 기술들로 구성되며, 합성곱 신경망(Convolutional deep Neural Networks, CNN)에 적용될 수 있으나 이에 한정되는 것은 아니다.
- [0086] 기계학습은 입력 데이터들의 특징을 스스로 분류/학습하는 알고리즘 기술이며, 요소기술은 딥러닝 등의 기계학습 알고리즘을 활용하여 인간 두뇌의 인지, 판단 등의 기능을 모사하는 기술로서, 언어적 이해, 시각적 이해, 추론/예측, 지식 표현, 동작 제어 등의 기술 분야로 구성된다.
- [0087] 구체적으로, 딥 러닝은 여러 비선형 변환기법의 조합을 통해 높은 수준의 추상화(abstractions, 다량의 데이터나 복잡한 자료들 속에서 핵심적인 내용 또는 기능을 요약하는 작업)를 시도하는 기계학습(machine learning) 알고리즘의 집합으로 정의된다. 딥 러닝은 큰 틀에서 사람의 사고방식을 컴퓨터에게 가르치는 기계학습의 한 분야로 볼 수 있다.
- [0088] 어떠한 데이터가 있을 때 이를 컴퓨터가 알아들을 수 있는 형태(예를 들어 영상의 경우는 픽셀정보를 열벡터로 표현하는 등)로 표현(representation)하고 이를 학습에 적용하기 위해 많은 연구(어떻게 하면 더 좋은 표현기법을 만들고 또 어떻게 이것들을 학습할 모델을 만들지에 대한)가 진행되고 있다. 이러한 노력의 결과로 다양한 딥 러닝 기법들이 개발되었다. 딥 러닝 기법들로는 심층 신경망(Deep Neural Networks, DNN), 합성곱 신경망(Convolutional deep Neural Networks, CNN), 순환 신경망(Reccurent Neural Network, RNN) 및 심층 신뢰 신경망(Deep Belief Networks, DBN)을 예로 들 수 있다.
- [0089] 심층 신경망(Deep Neural Networks, DNN)은 입력 계층(input layer)과 출력 계층(output layer) 사이에 복수개의 은닉 계층(hidden layer)들로 이뤄진 인공신경망(Artificial Neural Network, ANN)이다.

- [0090] 이때, 심층 신경망의 구조는 퍼셉트론(perceptron)으로 구성될 수 있다. 퍼셉트론은 여러 개의 입력 값(input)과 하나의 프로세서(processor), 하나의 출력 값으로 구성된다. 프로세서는 여러 개의 입력 값에 각각 가중치(또는 파라미터)를 곱한 후, 가중치(또는 파라미터)가 곱해진 입력 값들을 모두 합한다. 그 다음 프로세서는 합해진 값을 활성화함수에 대입하여 하나의 출력 값을 출력한다. 만약 활성화함수의 출력 값으로 특정한 값이 나오기를 원하는 경우, 각 입력 값에 곱해지는 가중치(또는 파라미터)를 수정하고, 수정된 가중치(또는 파라미터)를 이용하여 출력 값을 다시 계산할 수 있다. 이때, 각각의 퍼셉트론은 서로 다른 활성화함수를 사용할 수 있다. 또한 각각의 퍼셉트론은 이전 계층에서 전달된 출력들을 입력으로 받아들이고, 활성화 함수를 이용하여 출력을 구한다. 구해진 출력은 다음 계층의 입력으로 전달된다. 상술한 바와 같은 과정을 거치면 최종적으로 몇 개의 출력 값을 얻을 수 있다.
- [0091] 합성곱 신경망(Convolutional deep Neural Networks, CNN)은 최소한의 전처리(preprocess)를 사용하도록 설계된 다계층 퍼셉트론(multilayer perceptrons)의 한 종류이다. 합성곱 신경망은 하나 또는 여러개의 합성곱 계층(convolutional layer)과 그 위에 올려진 일반적인 인공신경망 계층들로 이루어져 있으며, 가중치(또는 파라미터)와 통합 계층(pooling layer)들을 추가로 활용한다. 이러한 구조 덕분에 합성곱 신경망은 2차원 구조의 입력 데이터를 충분히 활용할 수 있다. 또한, 합성곱 신경망은 표준 역전달을 통해 훈련될 수 있다. 합성곱 신경망은 다른 피드포워드 인공신경망 기법들보다 쉽게 훈련되는 편이고 적은 수의 매개변수를 사용한다는 이점이 있다. 합성곱 신경망은 입력 영상에 대하여 합성곱과 서브샘플링을 번갈아 수행함으로써 입력 영상으로부터 특징을 추출한다.
- [0092] 합성곱 신경망은 여러 개의 합성곱 계층(Convolution layer), 여러 개의 서브샘플링 계층(Subsampling layer, Local pooling layer, Max-Pooling layer), 완전 연결 층(Fully Connected layer)을 포함한다. 합성곱 계층은 입력 영상(Input Image)에 대해 합성곱을 수행하는 계층이다. 그리고 서브샘플링 계층은 입력 영상에 대해 지역적으로 최대값을 추출하여 2차원 영상으로 매핑하는 계층으로, 국소적인 영역을 더 크게 하고, 서브샘플링을 수행한다.
- [0093] 합성곱 계층에서는 커널의 크기(kernel size), 사용할 커널의 개수(즉, 생성할 맵의 개수), 및 합성곱 연산 시에 적용할 가중치(또는 파라미터) 테이블 등의 정보가 필요하다. 예를 들어, 입력 영상의 크기가 32×32 이고, 커널의 크기가 5×5 이고, 사용할 커널의 개수가 20개인 경우를 예로 들자. 이 경우, 32×32 크기의 입력 영상에 5×5 크기의 커널을 적용하면, 입력 영상의 위, 아래, 왼쪽, 오른쪽에서 각각 2개의 픽셀(pixel)에는 커널을 적용하는 것이 불가능하다. 입력 영상의 위에 커널을 배치한 후 합성곱을 수행하면, 그 결과 값인 '-8'은 커널에 포함된 입력 영상의 픽셀들 중에서 커널의 중심요소(center element)에 대응하는 픽셀의 값으로 결정되기 때문이다. 따라서, 32×32 크기의 입력 영상에 5×5 크기의 커널을 적용하여 합성곱을 수행하면 28×28 크기의 맵(map)이 생성된다. 앞서, 사용할 커널의 개수가 총 20개인 경우를 가정하였으므로, 첫 번째 합성곱 계층에서는 총 20개의 28×28 크기의 맵이 생성된다.
- [0094] 서브샘플링 계층에서는 서브샘플링할 커널의 크기에 대한 정보, 커널 영역 내의 값들 중 최대값을 선택할 것인지 최소값을 선택할 것인지에 대한 정보가 필요하다.
- [0095] 딥 러닝 기법에 대한 설명으로 다시 돌아가면, 순환 신경망(Recurrent Neural Network, RNN)은 인공신경망을 구성하는 유닛 사이의 연결이 Directed cycle을 구성하는 신경망을 말한다. 순환 신경망은 앞먹임 신경망과 달리, 임의의 입력을 처리하기 위해 신경망 내부의 메모리를 활용할 수 있다.
- [0096] 심층 신뢰 신경망(Deep Belief Networks, DBN)이란 기계학습에서 사용되는 그래프 생성 모형(generative graphical model)으로, 딥 러닝에서는 잠재변수(latent variable)의 다중계층으로 이루어진 심층 신경망을 의미한다. 계층 간에는 연결이 있지만 계층 내의 유닛 간에는 연결이 없다는 특징이 있다.
- [0097] 심층 신뢰 신경망은 생성 모형이라는 특성상 선행학습에 사용될 수 있고, 선행학습을 통해 초기 가중치(또는 파라미터)를 학습한 후 역전과 혹은 다른 판별 알고리즘을 통해 가중치(또는 파라미터)의 미조정을 할 수 있다. 이러한 특성은 훈련용 데이터가 적을 때 굉장히 유용한데, 이는 훈련용 데이터가 적을수록 가중치(또는 파라미터)의 초기값이 결과적인 모델에 끼치는 영향이 세지기 때문이다. 선행학습된 가중치(또는 파라미터) 초기값은 임의로 설정된 가중치(또는 파라미터) 초기값에 비해 최적의 가중치(또는 파라미터)에 가깝게 되고 이는 미조정 단계의 성능과 속도향상을 가능케 한다.
- [0098] 상술한 인공지능 및 그 학습방법에 관한 내용은 예시를 위하여 서술된 것이며, 이하에서 설명되는 실시 예들에 서 이용되는 인공지능 및 그 학습방법은 제한되지 않는다. 예를 들어, 당 업계의 통상의 기술자가 동일한 과제

해결을 위하여 적용할 수 있는 모든 종류의 인공지능 기술 및 그 학습방법이 개시된 실시 예에 따른 시스템을 구현하는 데 활용될 수 있다.

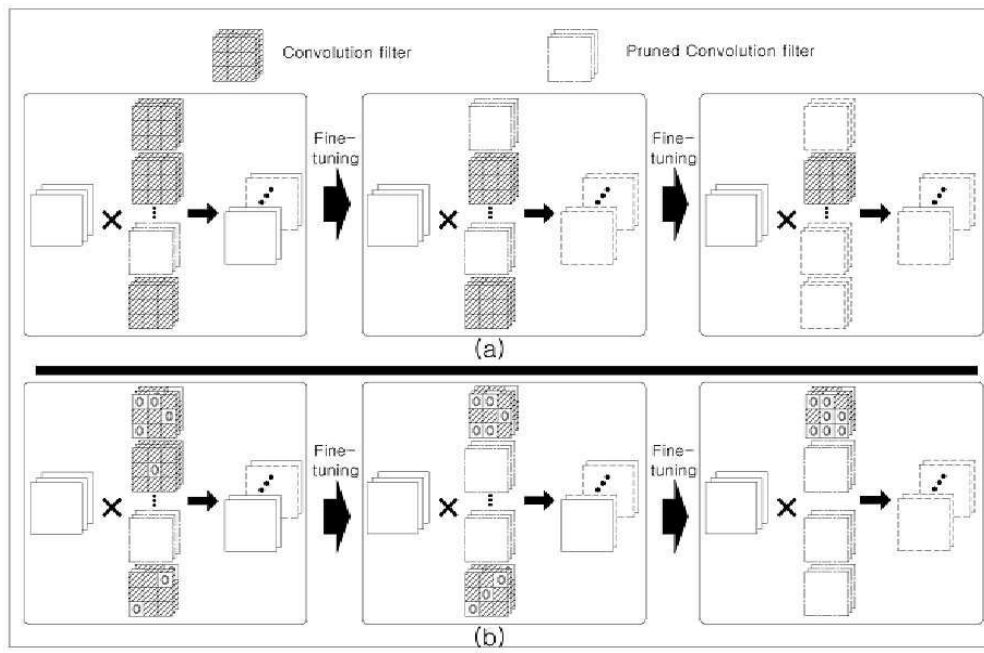
- [0099] 도 9는 본 발명의 일 실시예에 따른 장치의 구성도이다.
- [0100] 프로세서(102)는 하나 이상의 코어(core, 미도시) 및 그래픽 처리부(미도시) 및/또는 다른 구성 요소와 신호를 송수신하는 연결 통로(예를 들어, 버스(bus) 등)를 포함할 수 있다.
- [0101] 일 실시예에 따른 프로세서(102)는 메모리(104)에 저장된 하나 이상의 인스트럭션을 실행함으로써, 도 1 내지 도 8과 관련하여 설명된 방법을 수행한다.
- [0102] 예를 들어, 프로세서(102)는 메모리에 저장된 하나 이상의 인스트럭션을 실행함으로써 신규 학습용 데이터를 획득하고, 학습된 모델을 이용하여, 상기 획득된 신규 학습용 데이터에 대한 테스트를 수행하고, 상기 테스트 결과, 라벨링된 정보가 소정의 제1 기준값 이상의 정확도로 획득되는 제1 학습용 데이터를 추출하고, 상기 추출된 제1 학습용 데이터를 상기 신규 학습용 데이터로부터 삭제하고, 상기 추출된 학습용 데이터가 삭제된 상기 신규 학습용 데이터를 이용하여 상기 학습된 모델을 다시 학습시킬 수 있다.
- [0103] 한편, 프로세서(102)는 프로세서(102) 내부에서 처리되는 신호(또는, 데이터)를 일시적 및/또는 영구적으로 저장하는 램(RAM: Random Access Memory, 미도시) 및 롬(ROM: Read-Only Memory, 미도시)을 더 포함할 수 있다. 또한, 프로세서(102)는 그래픽 처리부, 램 및 롬 중 적어도 하나를 포함하는 시스템온칩(SoC: system on chip) 형태로 구현될 수 있다.
- [0104] 메모리(104)에는 프로세서(102)의 처리 및 제어를 위한 프로그램들(하나 이상의 인스트럭션들)을 저장할 수 있다. 메모리(104)에 저장된 프로그램들은 기능에 따라 복수 개의 모듈들로 구분될 수 있다.
- [0105] 본 발명의 실시예와 관련하여 설명된 방법 또는 알고리즘의 단계들은 하드웨어로 직접 구현되거나, 하드웨어에 의해 실행되는 소프트웨어 모듈로 구현되거나, 또는 이들의 결합에 의해 구현될 수 있다. 소프트웨어 모듈은 RAM(Random Access Memory), ROM(Read Only Memory), EPROM(Erasable Programmable ROM), EEPROM(Electrically Erasable Programmable ROM), 플래시 메모리(Flash Memory), 하드 디스크, 착탈형 디스크, CD-ROM, 또는 본 발명이 속하는 기술 분야에서 잘 알려진 임의의 형태의 컴퓨터 판독가능 기록매체에 상주할 수도 있다.
- [0106] 본 발명의 구성 요소들은 하드웨어인 컴퓨터와 결합되어 실행되기 위해 프로그램(또는 애플리케이션)으로 구현되어 매체에 저장될 수 있다. 본 발명의 구성 요소들은 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수 있으며, 이와 유사하게, 실시 예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블러(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다.
- [0107] 이상, 첨부된 도면을 참조로 하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야의 통상의 기술자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로, 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며, 제한적이지 않은 것으로 이해해야만 한다.

부호의 설명

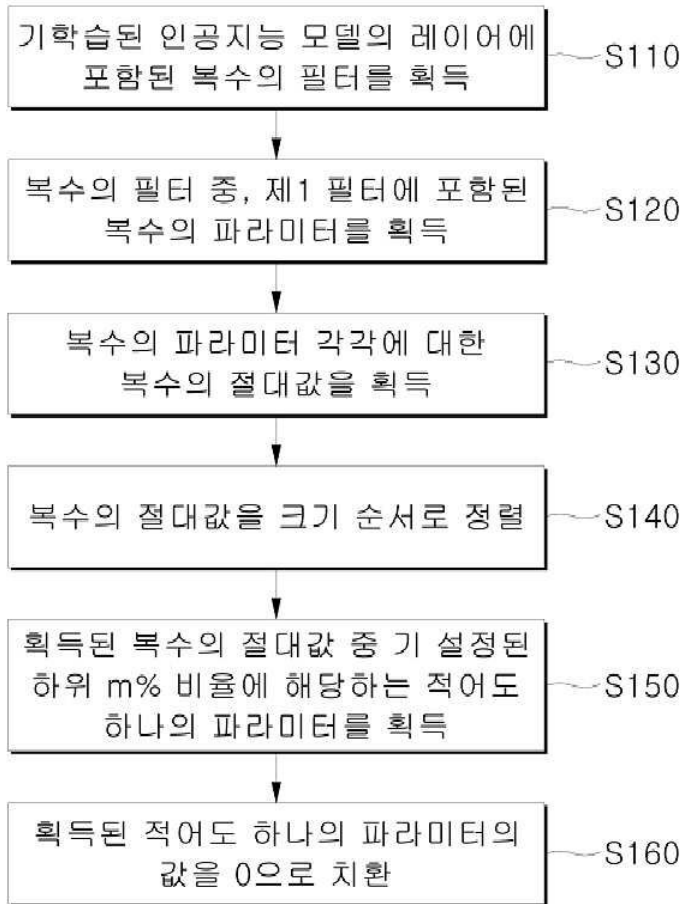
- [0108] 100 : 전자 장치

도면

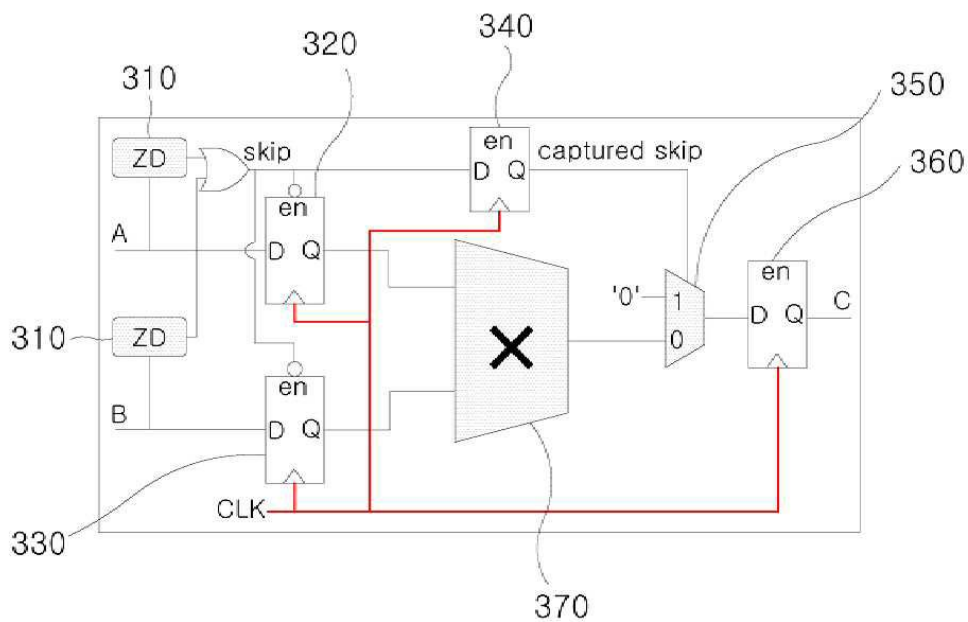
도면1



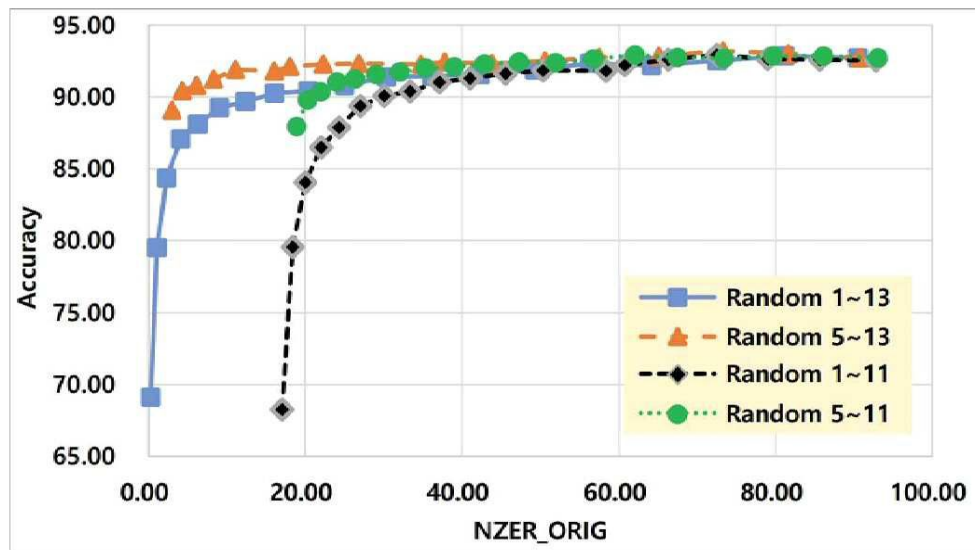
도면2



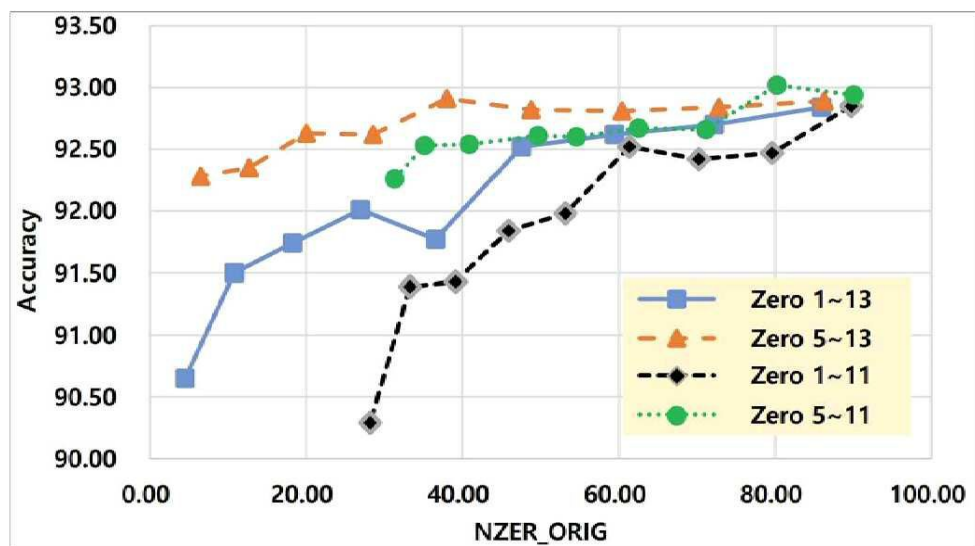
도면3



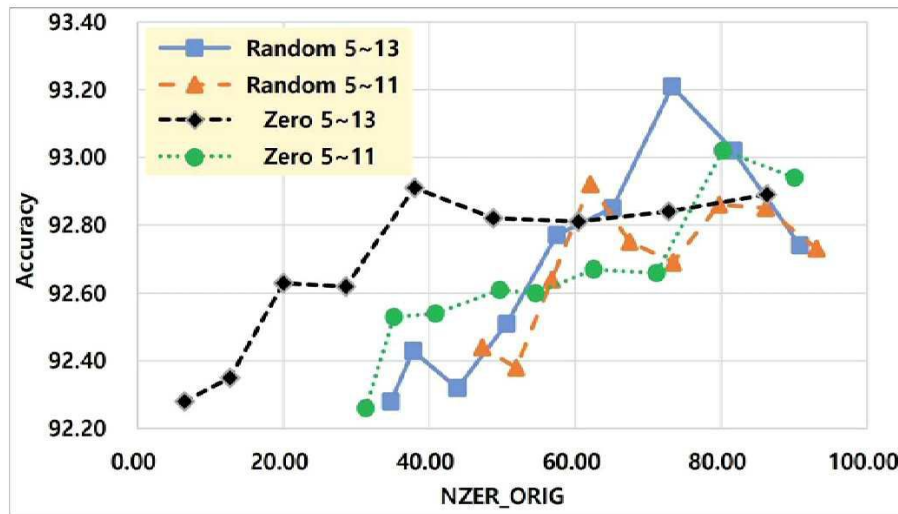
도면4



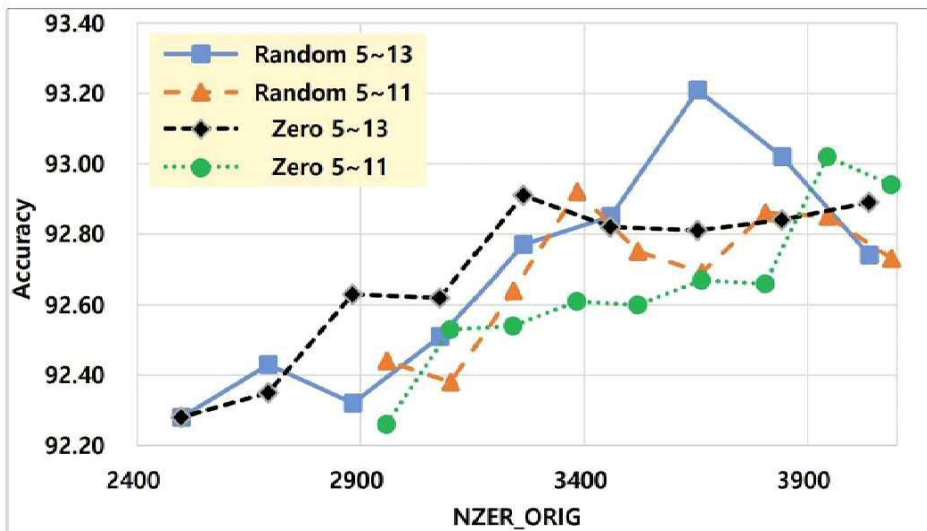
도면5



도면6



도면7



도면8

Pruning	Random Pruning				Zero-keep filter pruning			
	NZER_ORIG	NZER	Number of filter	Accuracy	NZER_ORIG	NZER	Number of filter	Accuracy
0	100	100	4224	94.15	100	100	4224	94.15
5	90.53	100	4020	92.72	85.92	94.97	4019	92.84
10	81.15	100	3807	92.86	72.13	88.95	3806	92.70
15	72.52	100	3598	92.53	59.44	82.02	3597	92.62
20	64.15	100	3385	92.19	47.51	74.11	3384	92.52
25	56.25	100	3168	92.35	36.58	65.09	3167	92.77
30	49.21	100	2964	91.88	26.97	54.85	2963	92.01
35	43.36	100	3751	91.57	18.26	43.15	3750	91.74
40	36.20	100	2542	91.45	10.79	29.85	2541	91.50
45	30.36	100	2329	91.40	4.49	14.80	2328	90.65

도면9

