



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0039045
(43) 공개일자 2023년03월21일

(51) 국제특허분류(Int. Cl.)
G06N 3/063 (2023.01) G06N 3/04 (2023.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G06N 3/063 (2013.01)
G06N 3/045 (2023.01)
(21) 출원번호 10-2021-0121891
(22) 출원일자 2021년09월13일
심사청구일자 2021년09월13일

(71) 출원인
연세대학교 산학협력단
서울특별시 서대문구 연세로 50 (신촌동, 연세대학교)
(72) 발명자
이진호
서울특별시 서대문구 연세로 50 연세대학교, 제 4공학관 D702호
김영석
서울특별시 서대문구 연세로 50 연세대학교, 제 4공학관 D703호
(뒷면에 계속)
(74) 대리인
정부연

전체 청구항 수 : 총 14 항

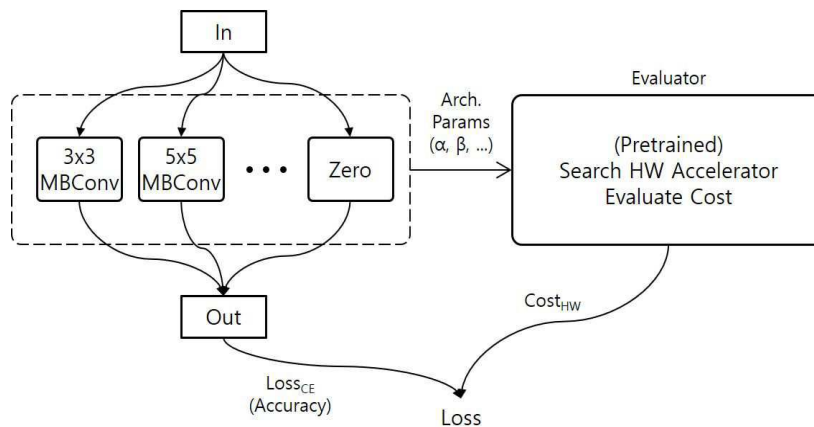
(54) 발명의 명칭 인공지능망과 연산 가속기 구조 통합 탐색 장치 및 방법

(57) 요약

본 발명은 인공지능망과 연산 가속기 구조 통합 탐색 장치는 신경망 아키텍처를 결정하는 NAS (Neural Architecture Search) 모듈; 및 상기 결정된 신경망 아키텍처에 따른 가속기 아키텍처를 결정하고 상기 결정된 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 DANCE (Differentiable Accelerator and Network Co-Exploration) 평가모듈을 포함한다. 따라서, 본 발명은 경사하강법을 사용하여 탐색을 진행함으로써 전체 탐색 공간을 대표하는 인공지능망을 한번 훈련하는 것으로 탐색을 완료할 수 있어 매우 빠른 탐색이 가능하고 미분 가능한 방식으로 지연시간이나 에너지 소모량과 같은 직접적인 하드웨어 메트릭을 최적화할 수 있다.

대표도 - 도7

100



(52) CPC특허분류
G06N 3/08 (2023.01)

(72) 발명자

최강현

서울특별시 서대문구 연세로 50 연세대학교, 제 4
공학관 D714호

홍덕기

서울특별시 서대문구 연세로 50 연세대학교, 제 4
공학관 D714호

이 발명을 지원한 국가연구개발사업

과제고유번호	1711126082
과제번호	2020-0-01361-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보통신방송혁신인재양성(R&D)
연구과제명	인공지능대학원지원(연세대학교)
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2021.01.01 ~ 2021.12.31

이 발명을 지원한 국가연구개발사업

과제고유번호	1711134555
과제번호	2021-0-00853-001
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	신개념PIM반도체선도기술개발(R&D)
연구과제명	PIM 활용을 위한 SW 플랫폼 개발
기 여 율	1/2
과제수행기관명	연세대학교 산학협력단
연구기간	2021.04.01 ~ 2021.12.31

명세서

청구범위

청구항 1

신경망 아키텍처를 결정하는 NAS (Neural Architecture Search) 모듈; 및

상기 결정된 신경망 아키텍처에 따른 가속기 아키텍처를 결정하고 상기 결정된 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 DANCE (Differentiable Accelerator and Network Co-Exploration) 평가모듈을 포함하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 2

제1항에 있어서, 상기 NAS 모듈은

복수의 후보 신경망 아키텍처들을 동시에 평가하여 상기 신경망 아키텍처를 선별하고 교차-엔트로피 손실 ($Loss_{CE}$)을 산출하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 3

제1항에 있어서, 상기 DANCE 평가모듈은

사전 학습을 통해 구축되고 상기 결정된 신경망 아키텍처에 따른 최적의 하드웨어를 상기 가속기 아키텍처로서 탐색하고 상기 가속기 아키텍처에 관한 PE (Processing Element) 어레이 구성 (PE_x, PE_y), 레지스터 파일 (RF, Register File) 구성 및 데이터플로우 (DF, dataflow) 구성 중 적어도 하나를 결정하는 하드웨어 생성 네트워크 (Hardware generation network); 및

상기 가속기 아키텍처에 관한 구성들을 기초로 상기 하드웨어 메트릭을 예측하는 비용 추정 네트워크 (Cost estimation network)를 포함하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 4

제3항에 있어서, 상기 하드웨어 생성 네트워크는

네트워크 아키텍처 스페이스 내에서 랜덤 네트워크들을 생성하고 상기 랜덤 네트워크들 중 하나를 상기 최적의 하드웨어로서 결정하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 5

제4항에 있어서, 상기 하드웨어 생성 네트워크는

ReLU (Rectified Linear Unit)를 활성화 함수로서 사용하는 다계층 퍼셉트론으로 구성하여 상기 랜덤 네트워크들을 탐색하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 6

제5항에 있어서, 상기 하드웨어 생성 네트워크는

상기 다계층 퍼셉트론 중 마지막을 Gumbel-Softmax로 연결하여 출력 값을 상기 비용 추정 네트워크의 입력 값으로 피쳐 포워딩 하는 방식에 의해 상기 출력 값이 상기 입력 값에 근접하도록 하는 것을 특징으로 하는 인공지능

경망과 연산 가속기 구조 통합 탐색 장치.

청구항 7

제3항에 있어서, 상기 비용 추정 네트워크는

ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하고 배치 정규화를 각 계층에 적용한 다계층 리그레션(regression)으로 구성하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 8

제7항에 있어서, 상기 비용 추정 네트워크는

상기 다계층 리그레션을 통해 레이턴시, 면적 및 에너지 소모량을 결정하여 상기 하드웨어 메트릭을 예측하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 9

제8항에 있어서, 상기 비용 추정 네트워크는

상기 레이턴시, 면적 및 에너지 소모량에 관한 리니어 조합(linear combination) 또는 프로덕트(product)를 산출하여 상기 하드웨어 메트릭을 예측하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 장치.

청구항 10

신경망 아키텍처를 결정하는 NAS 모듈 수행단계; 및

상기 결정된 신경망 아키텍처에 따른 가속기 아키텍처를 결정하고 상기 결정된 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 DANCE 평가모듈 수행단계를 포함하는 인공지능망과 연산 가속기 구조 통합 탐색 방법.

청구항 11

제10항에 있어서, 상기 DANCE 평가모듈 수행단계는

사전 학습을 통해 구축되고 상기 결정된 신경망 아키텍처에 따른 최적의 하드웨어를 상기 가속기 아키텍처로서 탐색하고 상기 가속기 아키텍처에 관한 PE (Processing Element) 어레이 구성(PEx, PEy), 레지스터 파일(RF, Register File) 구성 및 데이터플로우(DF, dataflow) 구성 중 적어도 하나를 결정하는 하드웨어 생성 네트워크(Hardware generation network) 수행단계; 및

상기 가속기 아키텍처에 관한 구성들을 기초로 상기 하드웨어 메트릭을 예측하는 비용 추정 네트워크(Cost estimation network) 수행단계를 포함하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 방법.

청구항 12

제11항에 있어서, 상기 하드웨어 생성 네트워크 수행단계는

네트워크 아키텍처 스페이스 내에서 랜덤 네트워크들을 생성하고 상기 랜덤 네트워크들 중 하나를 상기 최적의 하드웨어로서 결정하는 단계를 포함하는 것을 특징으로 하는 인공지능망과 연산 가속기 구조 통합 탐색 방법.

청구항 13

제12항에 있어서, 상기 하드웨어 생성 네트워크 수행단계는

ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하는 다계층 퍼셉트론으로 구성하여 상기 랜덤 네트워크들을 탐색하는 단계를 포함하는 것을 특징으로 하는 인공신경망과 연산 가속기 구조 통합 탐색 방법.

청구항 14

제11항에 있어서, 상기 비용 추정 네트워크 수행단계는

ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하고 배치 정규화를 각 계층에 적용한 다계층 리그레션(regression)으로 구성하는 단계를 포함하는 것을 특징으로 하는 인공신경망과 연산 가속기 구조 통합 탐색 방법.

발명의 설명

기술 분야

[0001] 본 발명은 인공신경망과 전용 하드웨어 가속기의 통합 탐색 기술에 관한 것으로, 보다 상세하게는 특정 시간 내에 탐색 공간을 효율적으로 탐색하면서 인공신경망의 정확도와 하드웨어 메트릭 간의 균형을 맞추는 최적 지점을 찾을 수 있는 인공신경망과 연산 가속기 구조 통합 탐색 장치 및 방법에 관한 것이다.

배경 기술

[0003] 수십 년에 걸친 연구자들의 노력 끝에 DNN은 이제 이미지 분류 및 보드 게임 플레이와 같은 다양한 응용 영역에서 인간에 가까운 성능을 보여주고 있다. 그러나, 이러한 성공은 폭발적인 컴퓨팅 집약(compute intensity)에 의한 것으로, 이에 따라 긴 GPU 학습 시간과 많은 하드웨어 비용이 요구되고 있다.

[0004] NAS(Neural Architecture Search)는 이러한 문제를 해결하기 위한 접근 방식에 해당할 수 있다. 과거에는 인간의 설계 노력을 줄이고 최신의 정확도를 달성하는 것을 목표로 시작되었으나, 최근에는 지연시간(latency) 등의 하드웨어 관련 비용이 고려되고 있다.

[0005] 문제를 해결하는 또 다른 방법은 특수 하드웨어(종종 '가속기'라고 함)를 사용하는 것일 수 있다. DNN 실행에 특화된 가속기를 활용하는 경우 우수한 지연시간 및/또는 비용이 달성될 수 있다. 예를 들어 Google TPU는 AlphaGo, 데이터 센터 및 클라우드 서비스의 처리를 가속화하기 위해 배포되고 있다. 전용 가속기를 설계하는 것은 지연시간뿐만 아니라 에너지 소비 및 면적과 같은 기타 하드웨어 비용 메트릭을 최적화하기 위한 또 다른 대규모 설계 문제를 발생시킬 수 있다.

[0006] 그러나, 네트워크 아키텍처와 가속기는 상호 독립적이지 않으며 한쪽을 집중적으로 최적화하면 종종 다른 쪽에 악영향을 미칠 수 있다. 예를 들어, 일반적으로 사용되는 분리 가능한 컨볼루션은 일반적으로 낮은 연산 요구량으로 인해 우수한 지연시간을 달성할 수 있다. 그러나, Google의 TPU와 같은 일부 유형의 가속기는 병렬 처리를 위해 많은 수의 출력 채널을 활용하도록 설계될 수 있다. 이 때문에 TPU에서 실행되는 분리 가능한 컨볼루션은 연산 횟수가 적음에도 불구하고 일반 컨볼루션 연산에 비해 지연시간이 길어질 수 있다. 이와 마찬가지로, 네트워크를 고려하지 않고 가속기만 최적화하는 경우 종종 최선이 아닌 차선책이 선택될 수 있다.

[0007] 이와 관련하여, 하드웨어 가속기와 네트워크 아키텍처의 통합 탐색은 원하는 응용 성능(즉, 정확도)과 합리적인 비용(지연시간, 면적 및 에너지 소비)을 달성하는데 있어 매우 중요할 수 있다. 기존의 통합 탐색 기법은 전형적으로 강화 학습(Reinforcement Learning, RL) 기법을 사용하고 있다.

[0008] 해당 기법들은 먼저 네트워크와 가속기 쌍을 생성할 수 있으며, 해당 쌍은 정확도를 위해 네트워크를 학습하고 하드웨어 비용 메트릭들을 측정함으로써 평가될 수 있다. 평가 이후 보상 함수가 계산될 수 있고, 해당 보상을 기반으로 새로운 디자인 쌍이 생성될 수 있다. 이러한 절차의 명백한 문제는 엄청난 검색 시간이 필요하다는 것일 수 있다. RL 기반 NAS 기술과 마찬가지로, 생성된 네트워크는 정확성 평가를 위해 완전히 훈련될 필요가 있다. 또한, 가속기 평가는 무시할 수 없는 시간과 자원이 소요되는 경우가 많을 수 있다. 따라서, 탐색에는 과도

한 시간이 필요하고 여전히 고품질의 솔루션을 획득하기 어려운 문제점이 존재한다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 한국공개특허 제10-2019-0101677호 (2019.09.02)

발명의 내용

해결하려는 과제

[0011] 본 발명의 일 실시예는 특정 시간 내에 탐색 공간을 효율적으로 탐색하면서 인공신경망의 정확도와 하드웨어 메트릭 간의 균형을 맞추는 최적 지점을 찾을 수 있는 인공신경망과 연산 가속기 구조 통합 탐색 장치 및 방법을 제공하고자 한다.

[0012] 본 발명의 일 실시예는 경사하강법을 사용하여 탐색을 진행함으로써 전체 탐색 공간을 대표하는 인공신경망을 한번 훈련하는 것으로 탐색을 완료할 수 있어 매우 빠른 탐색이 가능하고 미분 가능한 방식으로 지연시간이나 에너지 소모량과 같은 직접적인 하드웨어 메트릭을 최적화할 수 있는 인공신경망과 연산 가속기 구조 통합 탐색 장치 및 방법을 제공하고자 한다.

과제의 해결 수단

[0014] 실시예들 중에서, 인공신경망과 연산 가속기 구조 통합 탐색 장치는 신경망 아키텍처를 결정하는 NAS (Neural Architecture Search) 모듈; 및 상기 결정된 신경망 아키텍처에 따른 가속기 아키텍처를 결정하고 상기 결정된 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 DANCE (Differentiable Accelerator and Network Co-Exploration) 평가모듈을 포함한다.

[0015] 상기 NAS 모듈은 복수의 후보 신경망 아키텍처들을 동시에 평가하여 상기 신경망 아키텍처를 선별하고 교차-엔트로피 손실(Loss_{CE})을 산출할 수 있다.

[0016] 상기 DANCE 평가모듈은 사전 학습을 통해 구축되고 상기 결정된 신경망 아키텍처에 따른 최적의 하드웨어를 상기 가속기 아키텍처로서 탐색하는 하드웨어 생성 네트워크(Hardware generation network); 및 상기 가속기 아키텍처에 관한 PE (Processing Element) 어레이 구성(PE_x, PE_y), 레지스터 파일(RF, Register File) 구성 및 데이터플로우(DF, dataflow) 구성 중 적어도 하나를 결정하고 상기 하드웨어 메트릭을 예측하는 비용 추정 네트워크(Cost estimation network)를 포함할 수 있다.

[0017] 상기 하드웨어 생성 네트워크는 네트워크 아키텍처 스페이스 내에서 랜덤 네트워크들을 생성하고 상기 랜덤 네트워크들 중 하나를 상기 최적의 하드웨어로서 결정할 수 있다.

[0018] 상기 하드웨어 생성 네트워크는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하는 다계층 퍼셉트론으로 구성하여 상기 랜덤 네트워크들을 탐색할 수 있다.

[0019] 상기 하드웨어 생성 네트워크는 상기 다계층 퍼셉트론 중 마지막을 Gumbel-Softmax로 연결하여 출력 값을 상기 비용 추정 네트워크의 입력 값으로 피쳐 포워딩 하는 방식에 의해 상기 출력 값이 상기 입력 값에 근접하도록 할 수 있다.

[0020] 상기 비용 추정 네트워크는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하고 배치 정규화를 각 계층에 적용한 다계층 리그레션(regression)으로 구성할 수 있다.

[0021] 상기 비용 추정 네트워크는 상기 다계층 리그레션을 통해 레이턴시, 면적 및 에너지 소모량을 결정하여 상기 하드웨어 메트릭을 예측할 수 있다.

[0022] 상기 비용 추정 네트워크는 상기 레이턴시, 면적 및 에너지 소모량에 관한 리니어 조합 또는 프로덕트

(combination and product)를 산출하여 상기 하드웨어 메트릭을 예측할 수 있다.

- [0023] 실시예들 중에서, 인공신경망과 연산 가속기 구조 통합 탐색 방법은 신경망 아키텍처를 결정하는 NAS (Neural Architecture Search) 모듈을 생성하는 NAS 모듈 생성단계; 및 상기 결정된 신경망 아키텍처에 따른 가속기 아키텍처를 결정하고 상기 결정된 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 DANCE (Differentiable Accelerator and Network Co-Exploration) 평가모듈을 생성하는 DANCE 평가모듈 생성단계를 포함한다.
- [0024] 상기 DANCE 평가모듈 수행단계는 사전 학습을 통해 구축되고 상기 결정된 신경망 아키텍처에 따른 최적의 하드웨어를 상기 가속기 아키텍처로서 탐색하고 상기 가속기 아키텍처에 관한 PE (Processing Element) 어레이 구성 (PEx, PEy), 레지스터 파일(RF, Register File) 구성 및 데이터플로우(dataflow) 구성 중 적어도 하나를 결정하는 하드웨어 생성 네트워크(Hardware generation network) 수행단계; 및 상기 가속기 아키텍처에 관한 구성들을 기초로 상기 하드웨어 메트릭을 예측하는 비용 추정 네트워크(Cost estimation network) 수행단계를 포함할 수 있다.
- [0025] 상기 하드웨어 생성 네트워크 수행단계는 네트워크 아키텍처 스페이스 내에서 랜덤 네트워크들을 생성하고 상기 랜덤 네트워크들 중 하나를 상기 최적의 하드웨어로서 결정하는 단계를 포함할 수 있다.
- [0026] 상기 하드웨어 생성 네트워크 수행단계는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하는 다계층 퍼셉트론으로 구성하여 상기 랜덤 네트워크들을 탐색하는 단계를 포함할 수 있다.
- [0027] 상기 비용 추정 네트워크 수행단계는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하고 배치 정규화를 각 계층에 적용한 다계층 리그레션(regression)으로 구성하는 단계를 포함할 수 있다.

발명의 효과

- [0029] 개시된 기술은 다음의 효과를 가질 수 있다. 다만, 특정 실시예가 다음의 효과를 전부 포함하여야 한다거나 다음의 효과만을 포함하여야 한다는 의미는 아니므로, 개시된 기술의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0030] 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 탐색 장치 및 방법은 특정 시간 내에 탐색 공간을 효율적으로 탐색하면서 인공신경망의 정확도와 하드웨어 메트릭 간의 균형을 맞추는 최적 지점을 찾을 수 있다.
- [0031] 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 탐색 장치 및 방법은 경사하강법을 사용하여 탐색을 진행함으로써 전체 탐색 공간을 대표하는 인공신경망을 한번 훈련하는 것으로 탐색을 완료할 수 있어 매우 빠른 탐색이 가능하고 미분 가능한 방식으로 지연시간이나 에너지 소모량과 같은 직접적인 하드웨어 메트릭을 최적화할 수 있다.

도면의 간단한 설명

- [0033] 도 1은 본 발명에 따른 통합 탐색 장치의 기능적 구성을 설명하는 도면이다.
- 도 2는 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 방법의 일 실시예를 설명하는 순서도이다.
- 도 3 및 4는 콘볼루션 계층에서 7개의 차원들과 CNN 실행을 설명하는 도면이다.
- 도 5는 DNN 가속기의 일 실시예를 설명하는 도면이다.
- 도 6은 RL 기반의 통합 탐색 과정을 설명하는 도면이다.
- 도 7은 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 탐색 방법을 설명하는 도면이다.
- 도 8은 본 발명에 따른 평가 네트워크 아키텍처를 설명하는 도면이다.
- 도 9는 본 발명에 따른 실험 결과를 설명하는 도면이다.
- 도 10 및 11은 본 발명에 따른 탐색 네트워크와 가속기 디자인의 실시예를 설명하는 도면이다.

발명을 실시하기 위한 구체적인 내용

- [0034] 본 발명에 관한 설명은 구조적 내지 기능적 설명을 위한 실시예에 불과하므로, 본 발명의 권리범위는 본문에 설명된 실시예에 의하여 제한되는 것으로 해석되어서는 아니 된다. 즉, 실시예는 다양한 변경이 가능하고 여러 가지 형태를 가질 수 있으므로 본 발명의 권리범위는 기술적 사상을 실현할 수 있는 균등물들을 포함하는 것으로 이해되어야 한다. 또한, 본 발명에서 제시된 목적 또는 효과는 특정 실시예가 이를 전부 포함하여야 한다거나 그러한 효과만을 포함하여야 한다는 의미는 아니므로, 본 발명의 권리범위는 이에 의하여 제한되는 것으로 이해되어서는 아니 될 것이다.
- [0035] 한편, 본 출원에서 서술되는 용어의 의미는 다음과 같이 이해되어야 할 것이다.
- [0036] "제1", "제2" 등의 용어는 하나의 구성요소를 다른 구성요소로부터 구별하기 위한 것으로, 이들 용어들에 의해 권리범위가 한정되어서는 아니 된다. 예를 들어, 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다.
- [0037] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결될 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다고 언급된 때에는 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다. 한편, 구성요소들 간의 관계를 설명하는 다른 표현들, 즉 "~사이에"와 "바로 ~사이에" 또는 "~에 이웃하는"과 "~에 직접 이웃하는" 등도 마찬가지로 해석되어야 한다.
- [0038] 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함하다" 또는 "가지다" 등의 용어는 실시된 특징, 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이며, 하나 또는 그 이상의 다른 특징이나 숫자, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0039] 각 단계들에 있어 식별부호(예를 들어, a, b, c 등)는 설명의 편의를 위하여 사용되는 것으로 식별부호는 각 단계들의 순서를 설명하는 것이 아니며, 각 단계들은 문맥상 명백하게 특정 순서를 기재하지 않는 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 단계들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.
- [0040] 본 발명은 컴퓨터가 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현될 수 있고, 컴퓨터가 읽을 수 있는 기록 매체는 컴퓨터 시스템에 의하여 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록 장치를 포함한다. 컴퓨터가 읽을 수 있는 기록 매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광 데이터 저장 장치 등이 있다. 또한, 컴퓨터가 읽을 수 있는 기록 매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산 방식으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다.
- [0041] 여기서 사용되는 모든 용어들은 다르게 정의되지 않는 한, 본 발명이 속하는 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한 이상적이거나 과도하게 형식적인 의미를 지니는 것으로 해석될 수 없다.
- [0043] 도 1은 본 발명에 따른 통합 탐색 장치의 기능적 구성을 설명하는 도면이다.
- [0044] 도 1을 참조하면, 통합 탐색 장치(100)는 전체 탐색 공간을 대표하는 인공신경망을 한번 훈련하는 것으로 탐색을 완료하여 매우 빠른 탐색이 가능하게 하고 미분 가능한 방식으로 지연시간이나 에너지 소모량과 같은 직접적인 하드웨어 메트릭을 최적화할 수 있다. 이를 위한 구성으로, 통합 탐색 장치(100)는 NAS(Neural Architecture) 모듈(110) 및 DANCE(Differentiable Accelerator and Network Co-Exploration) 평가 모듈(130)을 포함하여 구현될 수 있다.
- [0045] NAS 모듈(110)은 신경망 아키텍처를 결정하는 동작을 수행할 수 있으며, DANCE 평가모듈(130)은 NAS 모듈(110)에 의해 결정된 신경망 아키텍처에 대응하는 가속기 아키텍처를 결정하고 해당 가속기 아키텍처에 관한 하드웨어 메트릭을 예측하는 동작을 수행할 수 있다.
- [0046] 보다 구체적으로, NAS 모듈(110)은 복수의 후보 신경망 아키텍처들을 동시에 평가하여 신경망 아키텍처를 선별할 수 있고, 이에 관한 교차-엔트로피 손실(Loss_{CE})을 산출할 수 있다.
- [0047] 일 실시예에서, DANCE 평가모듈(130)은 사전 학습을 통해 구축될 수 있으며, 두 개의 네트워크를 포함하여 구성

될 수 있다. 즉, DANCE 평가모듈(130)은 하드웨어 생성 네트워크(Hardware generation network)와 비용 추정 네트워크(Cost estimation network)를 포함할 수 있다.

[0048] 먼저, 하드웨어 생성 네트워크는 NAS 모듈(110)에 의해 결정된 신경망 아키텍처에 따른 최적의 하드웨어를 가속기 아키텍처로서 탐색하고 가속기 아키텍처에 관한 PE (Processing Element) 어레이 구성(PE_x, PE_y), 레지스터 파일(RF, Register File) 구성 및 데이터플로우 구성 중 적어도 하나를 결정하는 동작을 수행할 수 있다. 즉, 하드웨어 생성 네트워크는 최적의 하드웨어 아키텍처를 탐색하기 위해 사전 학습이 이루어질 수 있으며, 최적의 하드웨어 아키텍처에 관한 최적의 구성들을 파라미터로서 생성할 수 있다. 예를 들어, 하드웨어 생성 네트워크는 출력으로서 최적의 하드웨어 아키텍처에 관한 PE 어레이의 피처들 PE_x, PE_y과 레지스터 파일 RF, 데이터플로우 DF 등을 생성할 수 있다.

[0049] 일 실시예에서, 하드웨어 생성 네트워크는 네트워크 아키텍처 스페이스 내에서 랜덤 네트워크들을 생성하고 랜덤 네트워크들 중 하나를 최적의 하드웨어로서 결정할 수 있다. 즉, 하드웨어 생성 네트워크는 랜덤 네트워크를 입력으로 수신할 수 있고, 평가기 네트워크를 학습하기 위한 정답(ground-truth)으로 사용될 수 있는 출력을 생성할 수 있다.

[0050] 일 실시예에서, 하드웨어 생성 네트워크는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하는 다계층 퍼셉트론(multi-layer perceptron)으로 구성하여 랜덤 네트워크들을 탐색할 수 있다. 예를 들어, 도 8과 같이 하드웨어 생성 네트워크(131)는 5계층 퍼셉트론으로 구성될 수 있다.

[0051] 일 실시예에서, 하드웨어 생성 네트워크(131)는 다계층 퍼셉트론 중 마지막을 Gumbel-Softmax로 연결하여 출력 값을 비용 추정 네트워크(133)의 입력 값으로 피처 포워딩 하는 방식에 의해 출력 값이 입력 값에 근접하도록 할 수 있다. 예를 들어, 도 8과 같이 하드웨어 생성 네트워크(131)는 5계층 퍼셉트론의 마지막에 Gumbel-Softmax를 적용하고 그 출력을 비용 추정 네트워크(133)의 입력으로 연결하도록 구현될 수 있다. 여기에서, Gumbel-softmax는 세트에서 단일 요소를 확률적으로 샘플링하는 방법을 학습할 수 있는 소프트맥스 함수에 해당할 수 있다.

[0052] 또한, 비용 추정 네트워크는 가속기 아키텍처에 관한 구성들을 기초로 하드웨어 메트릭을 예측하는 동작을 수행할 수 있다. 일 실시예에서, 비용 추정 네트워크는 ReLU(Rectified Linear Unit)를 활성화 함수로서 사용하고 배치 정규화를 각 계층에 적용한 다계층 리그레션(regression)으로 구성할 수 있다. 예를 들어, 비용 추정 네트워크(133)는 도 8과 같이 5계층 리그레션으로 구성될 수 있다. 이때, 비용 추정 네트워크(133)는 계층 간에 잔여 연결(residual connection)을 포함할 수 있다.

[0053] 일 실시예에서, 비용 추정 네트워크는 다계층 리그레션을 통해 레이턴시(latency), 면적(area) 및 에너지 소모량(energy consumption)을 결정하여 하드웨어 메트릭을 예측할 수 있다. 이때, 비용 추정 과정에서 평가 소프트웨어를 통해 생성된 정답(ground truth)이 사용될 수 있다.

[0054] 일 실시예에서, 비용 추정 네트워크는 레이턴시, 면적 및 에너지 소모량에 관한 리니어 조합 또는 프로덕트(combination and product)를 산출하여 하드웨어 메트릭을 예측할 수 있다. 즉, 비용 추정 네트워크는 비용 함수(cost function)를 이용하여 하드웨어 메트릭을 예측할 수 있으며, 비용 함수는 레이턴시, 면적 및 에너지 소모량에 관한 리니어 조합으로 정의되거나 또는 레이턴시, 면적 및 에너지 소모량 간의 프로덕트(combination and product)로 정의될 수 있다.

[0056] 도 2는 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 탐색 방법의 일 실시예를 설명하는 순서도이다.

[0057] 도 2를 참조하면, 통합 탐색 장치(100)는 NAS 모듈(110)을 통해 신경망 아키텍처를 결정할 수 있다(단계 S210). 통합 탐색 장치(100)는 DANCE 평가모듈(130)을 통해 신경망 아키텍처에 따른 가속기 아키텍처를 결정할 수 있다(단계 S230). 통합 탐색 장치(100)는 DANCE 평가모듈(130)을 통해 가속기 아키텍처에 관한 하드웨어 메트릭을 예측할 수 있다(단계 S250).

[0059] 이하, 도 3 내지 11을 참조하여 본 발명에 따른 인공신경망과 연산 가속기 구조 통합 탐색 방법을 보다 구체적으로 설명한다.

[0060] 신경 아키텍처 검색(NAS, Neural Architecture Search)은 증가하는 네트워크의 크기와 이에 상응하는 수동적인

설계 노력에 대응하기 위하여 DNN 아키텍처의 설계를 자동화할 수 있다. 신경 아키텍처 검색에 있어 초기에는 네트워크 생성을 위해 강화학습(RL) 또는 진화 알고리즘(EA)이 채택되어 왔다.

[0061] 다만, 이러한 알고리즘들의 경우 검색 비용이 매우 높을 수 있으며 모든 후보들에게 요구되는 전체 학습(full training)으로 인해 최대 수천 개의 GPU 일(GPU-days)이 소요될 수 있다. 본 발명에 따른 미분 가능한(differentiable) 신경 아키텍처 검색은 이러한 비용을 완화하는 방법으로 슈퍼그래프(supergraph)를 만들고 그 안에서 경로를 찾을 수 있다. 즉, 미분 가능한 신경 아키텍처 검색은 몇 배나 더 짧은 시간 내에 최첨단 성능의 네트워크를 찾을 수 있다.

[0063] DNN용 하드웨어 가속기(hardware accelerator)들은 최근 CNN에서 가장 일반적인 연산인 다중 MAC(Multiply-Accumulate) 연산들을 병렬적으로 실행하는데 중점을 두고 있다. 도 5는 온칩 메모리(on-chip memory), 많은 PE(Processing Element)들 및 이들 간의 상호 연결을 포함하는 아이리스(Eyeriss)와 유사한 DNN 가속기의 일 실시예를 도시하고 있다. 백본 가속기 설계를 사용하더라도, PE들의 개수, 데이터플로우(dataflow) 및 레지스터 파일 크기 등과 같은 많은 속성들이 여전히 설계될 필요가 있다.

[0064] 일반적으로, DNN 계층(layer)은 여러 차원의 컴퓨팅 연산을 포함할 수 있습니다. 예를 들어, 컨볼루션 계층(convolution layer)은 도 3과 같이 7개의 컴퓨팅 연산 계층들을 포함할 수 있다. 즉, 컨볼루션 계층(convolution layer)은 활성화 입력(input activation)들 H, W, C에 관한 3개의 계층들, 가중치(weight) R, S, K에 관한 3개의 계층들 및 배치(batch)들 N에 관한 하나의 계층들을 포함할 수 있다. 따라서, 도 4와 같이 7단계의 중첩된 루프들로 공식화될 수 있다. 가속기에서 이러한 루프들을 매핑하고 순서를 지정하는 것을 종종 데이터플로우(dataflow)라고 부를 수 있으며, 많은 가속기들은 일부 데이터를 가능한 한 오랫동안 로컬 메모리 상에 유지하는데 중점을 둔 다양한 데이터플로우들을 제공할 수 있다.

[0065] 가속기 설계에 있어 각 선택이 DNN 지연시간에 어떤 영향을 미치는지 분석하는 것은 시뮬레이터(simulator) 또는 분석 평가 도구(analytical evaluation tool)에 의해 수행될 수 있다. 본 발명에 따른 통합 탐색 방법은 DANCE 프레임워크 상에서 평가 네트워크(evaluation network)를 학습하기 위한 최첨단 가속기 평가 튜체인으로서 Accelergy와 결합된 Timeloop를 활용할 수 있다.

[0067] 네트워크 아키텍처와 가속기 설계를 통합 탐색하는 방법에 있어서, 기존의 방법들은 문제를 공식화하는 비교적 간단한 방법으로 인해 강화학습(RL)을 제어기(controller)로서 사용할 수 있다. 그러나, 해당 방법들은 모두 강화학습 기반의 NAS 알고리즘에서 발생하는 동일한 검색 비용의 문제를 그대로 포함할 수 있다.

[0068] 이에 반해, 본 발명에 따른 방법은 최고의 정확도를 가진 네트워크와 가속기 디자인을 생성하면서 탐색 비용을 크게 줄일 수 있는 공동 탐색 문제에 관한 미분 가능한 NAS의 아이디어를 적용할 수 있다. EDD라는 기존 방법은 공동 탐색 문제에 대한 미분 가능한 방법을 제공할 수 있다. 그러나, 해당 방법은 몇 가지 중요한 제한 사항을 가질 수 있다. EDD는 네트워크의 총 플롭(total flops)을 계산 리소스(computation resource)의 양으로 나눈 것을 지연시간(latency)으로 모델링할 수 있다. 결과적으로, 네트워크 아키텍처와 가속기 설계 간의 진정한 관계는 통합 검색(co-search)에서 고려되지 않을 수 있다. 이것은 이론적으로 데이터플로우나 레지스터 파일 크기와 같은 몇 가지 중요한 특징에 대한 검색을 허용하지 않을 수 있다. 또한, EDD의 주요 초점은 각 계층에 대해 다양한 양자화(quantization)를 사용하는 것일 수 있다. 따라서, EDD는 각 계층에 대한 (공유 가능한)전용 하드웨어가 존재하며 일반적인 가속기들과는 차이가 있다는 가정을 포함할 수 있다.

[0070] 도 6을 참조하면, RL 기반의 통합 탐색이 수행되는 방법이 도시되어 있다. 검은색 문자는 일반적인 NAS 알고리즘의 구성 요소를 의미하고, 파란색 문자는 통합 탐색을 위해 추가된 구성 요소를 의미할 수 있다. 첫째, 네트워크 아키텍처 및 하드웨어 가속기의 검색 공간이 컨트롤러(Controller)에 제공될 수 있다. 그런 다음, 컨트롤러는 제공된 검색 공간(즉, 네트워크 아키텍처와 하드웨어 가속기)에 대한 후보 디자인을 생성할 수 있다. 생성된 후보는 정확도를 얻기 위해 네트워크에서 학습이 수행될 수 있고, 네트워크를 실행하는 지정된 하드웨어에 대한 비용 메트릭을 분석하는 평가자(Evaluator)에게 전달될 수 있다. 해당 방법은 통합 탐색의 목적에는 잘 부합하는 반면, RL 기반 NAS 알고리즘의 동일한 문제인 학습 비용(the training cost)을 포함할 수 있다. 즉, 해당 방법은 생성되는 각 후보자에 대해 비용이 많이 드는 학습이 필요할 수 있다. 또한, 해당 방법에 있어 최적의 하드웨어 설계를 찾는 동작 역시 후보 별로 수행되는 동안 상당한 시간이 소요될 수 있다. 결과적으로, 해당

방법의 검색 동작은 많은 GPU 시간(GPU-hours)으로 인해 어려움을 겪을 수 있다.

[0072] 도 7을 참조하면, 본 발명에 따른 통합 탐색 장치(100)에서 실행되는 통합 탐색 방법이 도시되어 있다. 즉, 본 발명에 따른 통합 탐색 방법은 DANCE(Differentiable Accelerator/Network Co-Exploration)라는 미분 가능한 통합 탐색 방법에 해당할 수 있다. 도 7의 왼쪽 부분은 다른 미분 가능한 NAS 알고리즘과 유사한 네트워크 검색 모듈(즉, 도 1의 NAS 모듈(110)에 대응됨)에 해당할 수 있으며, 역전파(backpropagation)를 사용하여 슈퍼 네트워크(super-network) 내의 경로를 찾음으로써 최종적으로 검색되는 네트워크를 생성할 수 있다. 한편, 네트워크 검색 모듈에는 다른 미분 가능한 NAS 알고리즘들이 모두 적용될 수 있음은 물론이다.

[0073] 도 7의 오른쪽 부분은 네트워크 검색 모듈로부터 획득한 아키텍처 파라미터들을 이용하여 최적의 하드웨어 가속기 설계(optimal hardware accelerator design)를 검색하고 비용 메트릭들(cost metrics)을 평가하는 미분 가능한 평가기(differentiable evaluator)(즉, 도 1의 DANCE 평가 모듈(130)에 대응됨)에 해당할 수 있다. 평가기는 사전 학습된 신경망으로 구현될 수 있으며, 검색 중에 고정(frozen)되고 해당 하드웨어 아키텍처를 하드웨어 비용 메트릭들에 연결하는 과정에서 사용될 수 있다. 손실 함수(loss function)는 다음의 수학적 식 1과 같이 표현될 수 있으며, 정확도(accuracy)와 비용 메트릭들(cost metrics)이 모두 고려될 수 있다.

[0074] [수학적 식 1]

$$Loss = Loss_{CE} + \lambda_1 ||w|| + \lambda_2 Cost_{HW}$$

[0077] 여기에서, λ_1 과 λ_2 는 항들(terms) 간의 트레이드오프(trade-off)를 조정하는 하이퍼파라미터(hyperparameter)들이다. $Loss_{CE}$ 는 교차-엔트로피 손실(cross-entropy)이며 $||w||$ 는 가중치 감소 항(weight decay term)이다. 또한, $Cost_{HW}$ 는 평가기 네트워크(evaluator network)의 출력 값으로부터 계산되는 하드웨어 가속기의 비용 함수(cost function)이다. 예를 들어, 비용 함수는 지연시간(latency), 면적(area) 및 에너지 소비(energy consumption)에 관한 선형 조합(linear combination)에 해당하거나 또는 EDAP (Energy-delay-area product, 에너지 지연 면적 곱)에 해당할 수 있다.

[0079] 원래의 (미분 불가능한) 비용 평가 소프트웨어들은 하드웨어 생성 도구(hardware generation tool)와 비용 추정 도구(cost estimation tool)로 구성될 수 있다. 하드웨어 생성 도구는 네트워크 아키텍처(network architecture)를 입력으로 사용하고 하드웨어 가속기 설계(hardware accelerator design)를 출력으로 생성할 수 있다. 본 발명에 따른 통합 탐색 방법은 하드웨어 가속기 설계(hardware accelerator design)의 검색 공간(search space)으로 데이터플로우(dataflow), X 및 Y 차원에 대한 PE들의 개수, 레지스터 파일 크기(register file size)를 사용할 수 있다. 이후, 비용 추정 도구는 하드웨어 가속기(hardware accelerator)와 네트워크 아키텍처(network architecture)를 사용하여 비용 메트릭들(cost metrics)을 출력으로 생성할 수 있다. 일반적으로, 하드웨어 생성 도구는 비용 추정 도구를 포함하는 외부 루프로 구현될 수 있다. 즉, 하드웨어 생성 도구는 완전 탐색(Exhaustive search) 또는 분기 한정(branch-and-bound) 알고리즘과 같은 정확한 알고리즘을 사용함으로써 하드웨어 검색 공간 H 내에서 주어진 네트워크 아키텍처 A에 대한 최적의 솔루션을 출력으로 생성할 수 있다.

[0080] 일 실시예에서, 본 발명에 따른 통합 탐색 방법은 비용 추정 과정에서 지연시간(latency)에 대해 Timeloop와 에너지/면적에 대해 Accelergy를 사용할 수 있다. 이때, Timeloop와 Accelergy는 최첨단 비용 추정 튜체인(cost estimation toolchain)에 해당할 수 있다. 본 발명에 따른 통합 탐색 방법은 비용 추정 도구를 이용하여 고유의 하드웨어 생성 도구를 설계할 수 있다. 본 발명에 따른 통합 탐색 방법은 네트워크 아키텍처 공간(network architecture space) A 상에서 입력으로 임의의 네트워크(random network)를 생성할 수 있고, 해당 튜체인의 출력은 평가기 네트워크(evaluator network)의 구성요소들을 학습하기 위한 정답(ground-truth)으로 사용될 수 있다.

[0081] 본 발명에 따른 평가기 네트워크는 하드웨어 생성 네트워크와 비용 추정 네트워크의 두 가지 모듈로 구성될 수 있다. 도 8을 참조하면, 본 발명에 따른 평가기 네트워크 아키텍처가 도시되어 있다. 하드웨어 생성 네트워크는 ReLU(Rectified Linear Unit)를 활성화 함수(activation function)로 사용하는 5-계층 퍼셉트론(five-layer

perceptron)으로 모델링 될 수 있다. 하드웨어 생성 네트워크는 비용 추정 네트워크의 정확도를 높이고 탐색 중인 네트워크에 대한 기울기 경로(gradient path)를 설정하기 위해 계층들 사이에 잔여 연결(residual connection)이 적용될 수 있다.

[0082] 비용 추정 네트워크는 잔여 연결이 있는 5-계층 회귀(five-layer regression)로 모델링 될 수 있다. 비용 추정 네트워크는 ReLU를 활성화 함수로 포함할 수 있으며 모든 계층에 배치 정규화(batch normalization)가 적용될 수 있다. 비용 추정 네트워크는 평가 소프트웨어(evaluation software)에서 생성된 정답(ground truth)을 기반으로 관심 있는 세 가지 비용 메트릭(즉, 지연시간, 면적 및 에너지 소비)을 출력으로 생성할 수 있다. 예를 들어, 평가 소프트웨어에는 Timeloop 및 Accelergy가 포함될 수 있다. 본 발명은 각 평가기 네트워크를 학습시키기 위해 MSRE(Mean Squared Relative Error) 손실을 사용할 수 있으며, 다음의 수학적 식 2와 같이 표현될 수 있다.

[0083] [수학적 식 2]

[0084]
$$Loss_{MSRE} = \sum_i (1 - \hat{y}_i / y_i)^2$$

[0086] 여기에서, y_i 는 Timeloop+Accelergy의 결과에서 생성된 각 메트릭에 대한 하드웨어 비용 함수($Cost_{HW}$)이고, \hat{y}_i 는 네트워크 출력을 사용하여 계산된 동일한 비용 함수이다. 일반적인 MSE 손실을 사용할 수도 있지만, 이 경우 높은 값을 갖는 메트릭들에게 부적절한 가중치를 부여하는 문제가 발생할 수 있다. 예를 들어, 검색 공간 내에서 출력되는 지연시간 값은 각 계층당 8ns에서 100ns 이상 까지의 범위를 가질 수 있다. MSE 손실을 사용하는 경우, 8ns 지연시간 중 10ns 오류(error)와 100ns 지연시간 중 10ns 오류를 동일하게 간주하여 지연시간이 긴 상황들을 보다 정확하게 모델링하는데 부당한 이익을 줄 수 있다. 즉, 지연시간이 짧은 가속기를 찾는다는 조건 하에서는 MSRE 손실이 더 바람직할 수 있다.

[0087] 평가기 아키텍처에 있어서, HW 비용 메트릭을 출력하는 비용 추정 네트워크는 최적의 하드웨어를 찾고 메트릭을 추정하는 두 가지 기능들을 내부적으로 모델링해야 함을 의미할 수 있다. 독립형 네트워크는 상당히 높은 정확도를 보여줄 수 있지만, 하드웨어 생성 네트워크의 출력에서 피쳐 전달 경로(feature forwarding path)를 추가함으로써 지연시간을 더욱 개선시킬 수 있다. 즉, 하드웨어 생성 네트워크의 결과는 비용 추정 네트워크에 대한 입력으로서 네트워크 아키텍처에 연결될 수 있다. 예를 들어, 하드웨어 생성 네트워크의 마지막 계층으로 Gumbel softmax를 사용하는 경우, 하드웨어 생성의 출력 값이 비용 추정 네트워크의 입력에 최대한 근접하도록 할 수 있다.

[0089] 응용(application)의 분류 정확도를 최적화하는 것과 비교하여 비용 메트릭에 대한 최적화는 경사하강법(gradient descent)에 있어서 상대적으로 더 쉬운 작업에 해당할 수 있다. 예를 들어, 대부분의 계층을 0으로 선택하면 모든 지연시간, 영역 및 에너지 소비가 빠르게 최적화될 수 있다. 네트워크 아키텍처가 이러한 솔루션으로 제한되는 경우, 최고의 정확도를 최적화하기 위해 필요한 경우에도 더 중요한 아키텍처를 찾기가 어려울 수 있다. 이러한 효과를 완화하기 위해, 하이퍼파라미터 워밍업 스케줄링(hyperparameter warm-up scheduling)이 사용될 수 있다. 하이퍼파라미터 워밍업 스케줄링은 처음 몇 개의 에포크(epoch)들에 대해 상기 수학적 식 1의 λ_2 를 작은 값으로 사용하고 네트워크 아키텍처가 높은 정확도를 위해 특정 단계에 도달한 이후 나중에 원하는 값으로 λ_2 를 증가시킬 수 있다.

[0091] 기본적으로, 하드웨어 비용 함수는 세 가지 하드웨어 비용 메트릭들에 관한 선형 조합을 상기 수학적 식 1의 비용 함수 $Cost_{HW}$ 로서 사용할 수 있으며, 다음의 수학적 식 3과 같이 표현될 수 있다

[0092] [수학적 식 3]

[0093]
$$Cost_{HW_linear} = \lambda_E Energy + \lambda_L Latency + \lambda_A Area$$

[0095] λ_E , λ_L 및 λ_A 를 제어함으로써, 각 비용 메트릭 간의 균형을 측정하는 방법에 대한 조건이 설정될 수 있다. 이러한 하이퍼파라미터의 스케일(scale)을 매칭시키기 위해 각 비용에 대해 mJ, ms 및 μm^2 단위가 사용될 수 있다.

[0096] 또한, 하드웨어 비용 함수는 모든 메트릭들 간의 곱을 비용 함수로 사용할 수 있으며, 다음의 수학적 식 4와 같이 표현될 수 있다.

[0097] [수학적 식 4]

[0098]
$$Cost_{HW_EDAP} = Energy \cdot Latency \cdot Area$$

[0100] 여기에서, EDAP는 하드웨어를 평가하는데 사용되는 공통 메트릭(예를 들어, energy-delay-area product)에 해당한다. 이 경우, 추가적인 하이퍼파라미터가 없고 단위가 존재하지 않는다는 점에서 이점을 가질 수 있다.

[0102] 이하, 본 발명에 관한 실험 결과를 설명한다.

[0103] 본 발명에 따른 통합 탐색 방법(즉, DANCE)에 대해 CIFAR-10 및 ImageNet (ILSVRC2012) 데이터셋을 기초로 몇 가지 실험들이 수행될 수 있다. 모든 알고리즘들은 PyTorch로 구현될 수 있으며, 4개의 RTX2080Ti GPU들에서 실행될 수 있다.

[0105] 검색 공간(Search Space)

[0106] 하드웨어 가속기 검색 공간(hardware accelerator search space)인 H의 경우, 최신 가속기 Eyeriss가 백본으로 사용될 수 있다. 설계 파라미터로 PE들의 개수, RF 크기 및 데이터플로우(Dataflow)가 사용될 수 있다. 2차원 PE 배열의 경우, 차원마다 변수 PE_X 및 PE_Y 가 별도로 할당될 수 있다. 여기에서, 각 값의 범위는 8에서 24일 수 있다. 설정에 있어서, PE_X 가 클수록 계층들이 더 많은 채널들을 가질 수 있고, PE_Y 가 클수록 병렬 처리를 위해 더 큰 피쳐 맵들이 사용될 수 있다. PE 당 RF 크기는 4에서 64 사이의 값을 가질 수 있다. 데이터플로우(Dataflow)의 경우, 기존 하드웨어 가속기들(즉, WS: Weight Stationary, OS: Output Stationary, RS: Row Stationary)에서 3개의 데이터플로우들이 선택될 수 있다. 오프칩 메모리에 대해 약 128GB/s의 HBM 메모리가 설정될 수 있다. 평가기 네트워크 상에서 각 변수는 하드웨어 생성 네트워크와 비용 추정 네트워크 간의 단계적 연결(cascaded connection)을 단순화하기 위해 원-핫 벡터(one-hot vector)로 공식화될 수 있다.

[0107] 네트워크 아키텍처 검색 공간(network architecture search space)인 A의 경우, 백본 네트워크 아키텍처로서 ProxylessNAS가 사용될 수 있다. 네트워크에는 13개의 계층들이 있으며 3개의 계층마다 채널 수가 증가할 수 있다.

[0108] 중간에 배치된 9개의 계층들 각각에는 건너편 연결(skip connection) 외에도 MBConv3X3_expand3, MBConv3X3_expand6, MBConv5X5_expand3, MBConv5X5_expand6, MBConv7X7_expand3, MBConv7X7_expand6 및 Zero의 7가지 후보 연산들이 포함될 수 있다. Zero가 선택된 경우, 건너편 연결만 포함될 수 있고, 계층은 네트워크에서 효과적으로 사라질 수 있다. 아키텍처 파라미터는 이진화된 방법(binanzed method)(예를 들어, ProxylessNAS)을 통해 학습될 수 있다.

[0110] 평가기 네트워크 결과들(Evaluator Network Results)

[0111] 1) 비용 추정 네트워크(Cost Estimation Network): 다음의 표 1은 평가기 네트워크의 구성요소에 대한 실험 결과에 해당한다.

[0112] [표 1]

Network	Accuracy			
Hardware Generation	PE_X 98.9%	PE_Y 98.3%	RF_Size 98.3%	Dataflow 98.8%
Cost Estimation (w/o feature forwarding)	Latency 93.7%	Energy 96.3%	Area 92.8%	
Cost Estimation (w/ feature forwarding)	Latency 99.6%	Energy 99.7%	Area 99.9%	
Overall Evaluator	Latency 98.3%	Energy 98.3%	Area 99.2%	

[0113]

[0115] 비용 추정 네트워크와 하드웨어 생성 네트워크는 정답(ground truth) 값을 기초로 독립적으로 학습될 수 있으며, 이후 상호 조합될 수 있다. 비용 추정 네트워크의 각 계층은 256의 너비를 가질 수 있고 해당 네트워크는 200 Epoch에 대해 학습률이 0.0001인 Adam 옵티마이저(optimizer)를 이용하여 학습될 수 있다. 배치 크기(batch size)는 256이 적용될 수 있다. 비용 추정 네트워크는 검색 공간에서 Timeloop+Accelergy로 생성된 180만 케이스들에 대해 학습될 수 있으며, 45만 케이스들에 대해 검증될 수 있다. 그 결과 세 가지 비용 메트릭들 모두 99% 이상의 정확도를 보이는 점에서 충분히 정확함을 나타낼 수 있다. 또한, 피쳐 전달(feature forwarding)은 정확도를 평균 4.3%p 향상시키는 것으로 관찰될 수 있다.

[0116]

2) 하드웨어 생성 네트워크(hardware generation network): 하드웨어 생성 네트워크의 경우 계층 너비(layer width)는 128로 설정될 수 있다. 손실 함수는 일반적인 CE 손실(CE loss)이 사용될 수 있으며, $Loss_{CE_HW}$ 와 같이 표현될 수 있다. 하드웨어 생성 네트워크는 200 에포크(Epoch)에 대해 배치 크기가 128인 SGD를 사용하여 학습될 수 있음, 학습률은 0.001에서 시작하여 50 Epoch마다 0.1배씩 감소될 수 있다. 또한, 검색 공간에서 50,000개의 네트워크 케이스들이 생성될 수 있으며, 유효성 검사(validation)를 위해 10,000개의 케이스들이 사용될 수 있다. 모든 하드웨어 가속기 설계 파라미터(hardware accelerator design parameter)들에서 하드웨어 생성 네트워크의 정확도가 거의 99%로 나타나는 점에서 충분히 정확한 것을 확인할 수 있다. 즉, 하드웨어 생성 네트워크는 정확하고 미분 가능할 뿐만 아니라 원래 생성 튜체인보다 훨씬 빠르게 동작할 수 있다. 동일한 기능을 가진 하드웨어 생성 네트워크의 추론 시간(inference time)은 단일 GPU로 약 0.5ms가 걸리는 반면, 생성 도구는 2개의 Intel Xeon Silver-4214 CPU들의 24 코어들에서 48 스레드들을 사용하여 약 112초가 소요될 수 있다.

[0117]

3) 종단간 평가기 네트워크 결과들(End-to-end Evaluator Network Results): 하드웨어 생성 네트워크 및 비용 추정 네트워크 간의 조합으로 전체 평가기 네트워크가 테스트될 수 있다. 중간 값이 원-핫 벡터가 아니라도 Gumbel softmax는 이를 잘 근사시킬 수 있고, 비용 메트릭들에 대해 여전히 약 99% 정확도를 유지할 수 있다.

[0119]

통합 탐색 결과들(Co-exploration Results)

[0120]

1) CIFAR-10에 대한 실험 결과: 첫 번째 베이스라인(baseline)에 대해, ProxylessNAS를 사용하여 검색을 수행하고 완전 탐색(Exhaustive-search) 도구를 사용하여 검색된 네트워크에서 하드웨어 생성이 수행될 수 있다. 이는 실제로 수행되는 전형적인 분리 설계를 나타낼 수 있다. 256의 배치 크기를 갖는 120 에포크(Epoch)들에 대해 검색이 수행된 반면 40 Epoch에 대해 위밍업이 수행될 수 있다. 학습률 0.025, 가중치 감소 0.00004(λ_1), 레이블 평활화(smoothing) 0.1 및 모멘텀(momentum) 0.9의 코사인 스케줄링을 사용하는 검색에는 Nesterov 모멘텀이 있는 SGD 옵티마이저가 사용될 수 있다. 검색 후 최종 네트워크는 300 에포크 동안 처음부터 학습될 수 있다. 학습을 위한 하이퍼파라미터들은 학습률이 0.008이고 가중치 감소 계수(weight decay factor)가 0.001이라는 점을 제외하면 동일할 수 있다. 또한, 두 번째 베이스라인으로 EDD를 사용할 수 있다. 데이터플로우 및 레지스터 파일에 대한 하드웨어 파라미터에는 EDD를 적용할 수 없으므로 PE들의 개수만을 기준으로 통합 탐색을 수행하고 나머지 파라미터에 대해서는 사후 검색(post search)이 수행될 수 있다. EDD에서 발생할 수 있는 문제

는 분류 손실(classification loss)에 지연시간 손실(latency loss)을 곱하는 손실 함수를 사용한다는 것이며, 다음의 수학적 식 5와 같이 표현될 수 있다.

[수학적 식 5]

$$Loss = \lambda_2 \cdot Loss_{CE} \cdot \sum Latency$$

여기에서, λ_2 는 두 항들 사이의 가중치를 조정하지 않는다. 이로 인해, 지연시간을 빠르게 최적화하기에 네트워크가 너무 많이 축소되는 심각한 문제가 발생할 수 있다. 그 결과, 솔루션은 매우 낮은 하드웨어 비용을 제공하지만 허용될 수 없는 정확도를 제공할 수 있다. 따라서, 해당 문제를 완화하기 위해 상기 수학적 식 1과 같이 손실 함수를 변경하는 실험이 수행될 수 있으며, EDD + Proposed Loss func.로 표시될 수 있다.

DANCE를 사용하여, 비용 함수들을 기초로 통합 탐색이 수행될 수 있다. $Cost_{HW_linear}$ 에 대해, latency-oriented, energy-oriented 및 balanced 라는 세 가지 비용 함수가 설정될 수 있다. 나머지 하이퍼파라미터들은 모두 베이스라인과 동일하게 설정될 수 있다. 검색 후 학습(after-search training과 유사하게, 최적의 하드웨어 가속기 설계(optimal hardware accelerator design)를 획득하기 위해 검색 후 1회의 정확한 하드웨어 생성이 수행될 수 있다.

전반적으로, DANCE는 베이스라인보다 우수한 네트워크 가속기 설계를 획득할 수 있다. 비교를 위해, 하나는 높은 정확도(-A)를 가진 것과 다른 하나는 효율적인 하드웨어 설계(-B)에 해당하는 두 가지 설계가 사용될 수 있다. 고정밀 설계(-A)의 경우, DANCE는 베이스라인과 거의 동일한 정확도를 달성할 수 있다(페널티 없음). 효율적인 하드웨어 설계(-B)의 경우, 1~2%의 정확도 감소 이내에서 최고의 비용 함수를 가진 설계를 선택할 수 있다. DANCE가 효율적인 통합 탐색을 수행하여 최대 10배 더 나은 EDAP 또는 3배 더 나은 지연시간을 달성할 수 있다. 지연시간 지향(latency-oriented) 비용 함수를 사용하면 지연시간은 다른 함수들보다 훨씬 낮은 값이 되는 반면, 에너지 지향(energy-oriented) 비용 함수는 다른 두 함수들보다 더 나은 에너지 소비를 달성할 수 있다. 결과적으로, DANCE를 이용하는 경우 관심 있는 솔루션을 획득하기 위해 비용 하이퍼파라미터를 조정할 수 있음을 의미할 수 있다.

도 9를 참조하면, DANCE가 단순히 하드웨어 비용으로 정확성을 희생하는 것이 아니라 베이스라인들과 비교하여 압도적인 솔루션을 검색한다는 것을 의미할 수 있다. 도 9에서, 베이스라인(Baseline)과 DANCE에서 찾은 설계의 EDAP-오차 관계가 도시되어 있다. 여기에서, 두 축에 대해 모두 낮을수록 더 좋은 것일 수 있다. 정확도와 $Cost_{HW}$ 간의 다른 균형을 달성하기 위해 상기 수학적 식 1에서 다양한 λ_2 에 대해 검색이 수행될 수 있다. 베이스라인과 DANCE는 모두 정확도 지향(accuracy-oriented) 하이퍼파라미터 설정으로 비슷한 정확도에 도달할 수 있지만 DANCE는 훨씬 더 나은 트레이드오프(trade-off)를 제공할 수 있으며, 플롭 페널티(Flops penalty)가 있는 베이스라인들보다 우수한 비용 메트릭을 제공할 수 있다. 또한, DANCE는 EDD와 비교하여 유사한 정확도 하에서 2배 이상의 우수한 EDAP 성능을 제공할 수 있다. 이는 EDD가 네트워크-하드웨어 관계를 모델링하지 않고 특히 높은 정확도로 솔루션에 대한 효율적인 설계 쌍을 찾을 수 없기 때문이다. 도 9에 도시된 EDD의 경우, 원래의 EDD의 정확도가 너무 낮기 때문에 본 발명에 따라 수정된 손실 함수를 사용하고 있다.

2) ImageNet에 대한 실험 결과: 다음의 표 2는 ImageNet 데이터셋에 대한 DANCE의 성능을 도시하고 있다.

[0130] [표 2]

Method	Acc.	Latency	Energy	EDAP
Baseline (No penalty) + HW	71.12%	23.3ms	71.6mJ	3014.0
Baseline (Flops Penalty) + HW	70.56%	13.4ms	70.9mJ	2709.0
EDD + Proposed Loss func.	70.34%	28.1ms	94.8mJ	5642.5
DANCE ($Cost_{HW_EDAP}$)	69.82%	7.5ms	42.7mJ	912.4
DANCE (Energy-Oriented)	69.55%	9.2ms	49.5mJ	1413.5
DANCE (Latency-Oriented)	70.41%	8.3ms	48.4mJ	1154.3
DANCE (Balanced)	70.15%	7.7ms	45.7mJ	1001.8

[0131]

[0133] 별도의 하드웨어 검색을 가진 베이스라인은 71.12%의 정확도를 제공하지만 하드웨어 비용이 많이 소요될 수 있다. 플롭 페널티(Flops Penalty) 또는 EDD를 적용하는 경우에는 효율적인 솔루션을 찾지 못할 수 있다. DANCE는 좋은 트레이드오프 포인트(trade-off point)를 발견할 수 있고, 최대 3배의 EDAP 이점과 함께 약간의 정확도 감소만으로 훨씬 더 나은 비용 메트릭을 제공할 수 있다.

[0135] DANCE에 의해 탐색된 네트워크 및 가속기 설계(Network and accelerator design searched by DANCE)

[0136] 도 10 및 11을 참조하면, 네트워크 아키텍처와 가속기 설계에 관한 두개의 집합이 도시되어 있다. 가속기 설계와 함께 네트워크 아키텍처를 찾는 방법에 대한 유용한 통찰력을 보여주기 때문에 지연시간 지향 비용 함수(latency-oriented cost function)와 에너지 지향 비용 함수(energy-oriented cost function)로 생성된 두 가지 비용 효율적인 설계(-B)를 적용할 수 있다. 도 10 및 11에서, 볼드체(bold character)로 표기된 값은 DANCE로 검색되는 설계 파라미터들에 해당할 수 있다.

[0137] 지연시간 지향 네트워크(latency-oriented network)(도 10)는 에너지 지향 네트워크(energy-oriented network)에 비해 커널 크기가 상대적으로 작을 수 있다(예를 들어, 7×7 MBConv 대신 3×3 MBConv). 반면에 지연시간 지향 네트워크는 더 큰 확장 비율에 따라 더 많은 채널들을 포함할 수 있다. 데이터플로우에 관계없이, 가속기는 채널 수준의 병렬 처리(channel-level parallelism)를 잘 활용할 수 있기 때문에, 더 많은 채널이 있으면 동시에 활성화되는 PE들의 개수를 늘리는데 도움이 되어 지연시간이 줄어들 수 있다. 이러한 네트워크에서 낮은 지연시간을 달성하기 위해 검색된 가속기는 속도를 가속화하는 상대적으로 더 큰 PE 배열(array)을 포함할 수 있다. 마지막으로, 선택된 WS(Weight Stationary) 데이터플로우는 일반적으로 낮은 지연시간을 달성하는데 좋은 것으로 알려져 있다.

[0138] 에너지 지향 네트워크(energy-oriented network)(도 11)는 더 작은 채널 너비와 함께 상대적으로 더 큰 커널 크기(7×7 MBConv)를 포함할 수 있다. 커널 크기가 클수록 PE 사용률이 낮아지고 지연시간이 늘어나는 경우가 많음에도 불구하고, 사용하지 않는 PE들의 개수가 많다고 해서 높은 에너지에 크게 기여하지 않을 수 있다. 즉, 동적 에너지 소비는 주로 MAC 연산 및 데이터 접근의 개수에 의존할 수 있다. 반면에, 채널 너비(channel width)가 작을수록 입력/출력 활성화를 위한 액세스 수가 줄어들기 때문에 종종 에너지 소비가 낮아질 수 있다. 작은 커널/넓은 너비의 동일한 MAC 연산을 갖는 계층과 큰 커널/좁은 너비의 계층을 비교하면, 전자는 높은 PE 활용률로 인해 지연시간이 더 좋고, 후자는 낮은 데이터 접근으로 인해 에너지 소비가 더 좋을 수 있다. 에너지 지향 비용 함수에 대한 가속기는 종종 좋은 에너지 효율을 나타내는 것으로 알려진 RS 데이터플로우를 갖는 것으로 검색되어 왔다. PE 어레이는 에너지 소비를 줄이기 위해 작을 수 있다. 깊이 방향 컨볼루션(depth-wise convolution)에는 하나의 출력 채널만 있기 때문에 PE_i는 특히 작을 수 있으며, 낮은 에너지에 대해 PE_i를 줄이는 것이 PE_k를 줄이는 것보다 더 유리할 수 있다. 각 PE는 지연시간 지향 설계에 비해 더 큰 RF를 가질 수 있다. 왜냐하면 RF가 클수록 GB(Global Buffer)에 대한 액세스가 줄어들고 에너지 소비가 적기 때문이다.

[0140] DANCE와 기존 통합 탐색 알고리즘들과의 비교(Comparison of DANCE with Existing Co-exploration Algorithms)

[0141] 다음의 표 3은 DANCE를 다른 가속기/네트워크 통합 탐색 알고리즘들(즉, Alg. [10] 내지 [14] 및 [17])과 비교

한 결과에 해당할 수 있다.

[표 3]

Alg.	Backbone	Dataset	Acc.(%)	GPU-hours	Candidates	Method	Net-HW Relation
[11]	Custom	DAC-SDC	68.6	N/A	68	CD*	✓
[12]	Custom	CIFAR-10	89.7	N/A	N/A	RL	✓
[13]	ResNet-9	CIFAR-10	93.2	3.5h	~160	RL	✓
[14]	NASBench	CIFAR-100	74.2	2300h	2300	RL	✓
[10]	ProxylessNAS	CIFAR-10	85.2	103.9h	308	RL	✓
[17] [†]	ProxylessNAS	CIFAR-10	94.4	3h	1	gradient	✗
DANCE	ProxylessNAS	CIFAR-10	95.0	3h	1	gradient	✓

[†] Reproduced and modified for the same setting *CD = Coordinate Descent

환경이 모두 다르기 때문에(예를 들어, ASIC vs FPGA, 다른 기술 노드, 다른 NAS 백본 등) 측정된 값들을 직접 비교할 수 없다. 또한, 정확도조차도 기본적인 NAS 알고리즘에 의존하기 때문에 직접 비교할 수 없다. 그러나, 그 차이가 큰 경우 방법의 검색 능력(searching capability)을 암시할 수 있으므로 대략적인 비교를 위해 정확도와 검색 비용을 정리할 수 있다.

대부분의 통합 탐색 알고리즘들은 강화학습을 활용할 수 있으며, 탐색 과정에서 많은 후보들을 학습시켜야 하는 문제를 가질 수 있다. 결과적으로, 그들 중 다수는 정확도가 떨어지는 차선의 네트워크 아키텍처만을 출력할 수 있다.

검색 시간은 또한 DANCE의 장점을 나타낼 수 있으며, RL 기반 작업들에 비해 훨씬 빠를 수 있다. 알고리즘 [13]의 경우 차이는 작지만 이는 백본 아키텍처가 모델 크기가 작고 수동으로 미세 조정된 아키텍처를 기반으로 하기 때문이다. '후보들(candidates)' 항목(column)은 이러한 경우를 고려하여 검색 비용을 공정하게 비교하려는 시도에 해당할 수 있다. 즉, 검색하는 동안 각 알고리즘이 학습해야 하는 후보들의 개수에 해당할 수 있다. RL 기반 통합 탐색 알고리즘은 학습을 위해 수백에서 수천개의 후보들이 필요할 수 있지만 DANCE는 오직 하나의 후보만을 사용할 수 있다. 알고리즘 [17]은 미분 가능하며 동일한 NAS 백본으로 재가공한 경우 비슷한 정확도와 검색 비용을 제공할 수 있다. 그러나, 알고리즘 [17]은 네트워크-하드웨어 관계를 반영할 수 없기 때문에 그 결과 통합 탐색 솔루션은 DANCE보다 훨씬 낮은 품질을 제공할 수 있다.

본 발명에 따른 통합 탐색 방법인 DANCE는 높은 정확도와 낮은 비용 메트릭을 모두 목표로 하는 하드웨어 가속기와 네트워크 아키텍처를 함께 탐색하는 새로운 미분 가능한 방법에 해당할 수 있다. 본 발명에 따른 통합 탐색 방법은 매우 낮은 검색 비용으로 정확도를 손상시키지 않고 효율적인 하드웨어 설계를 얻기 위해 신경망 기반 하드웨어 평가를 모델링할 수 있다. 본 발명에 따른 통합 탐색 방법은 비디오 또는 자연어(natural language) 처리와 같은 미래의 많은 분야에서 통합 탐색 문제(co-exploration problem)에 대한 비용을 줄일 수 있다.

상기에서는 본 발명의 바람직한 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 특허 청구의 범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

부호의 설명

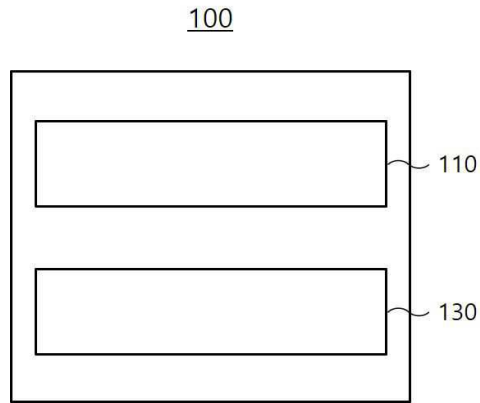
100: 통합 탐색 장치

110: NAS 모듈 130: DANCE 평가모듈

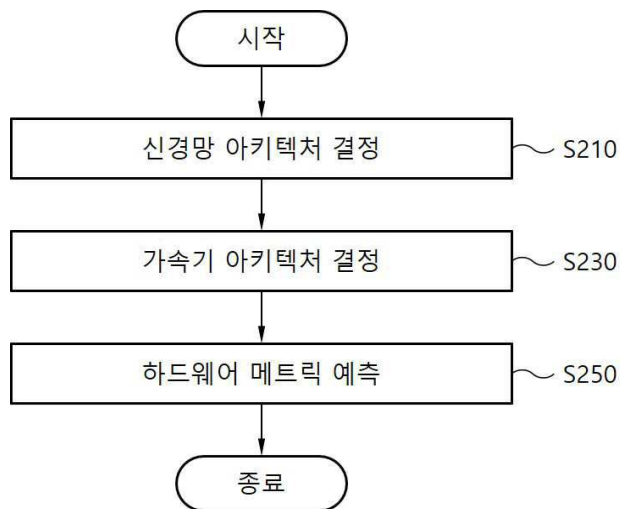
131: 하드웨어 생성 네트워크 133: 비용 추정 네트워크

도면

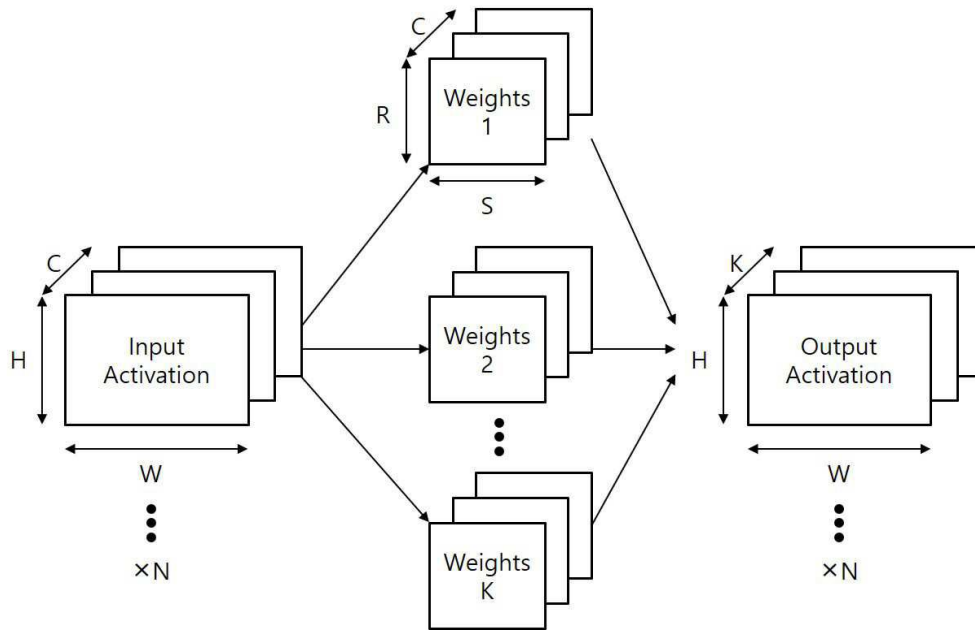
도면1



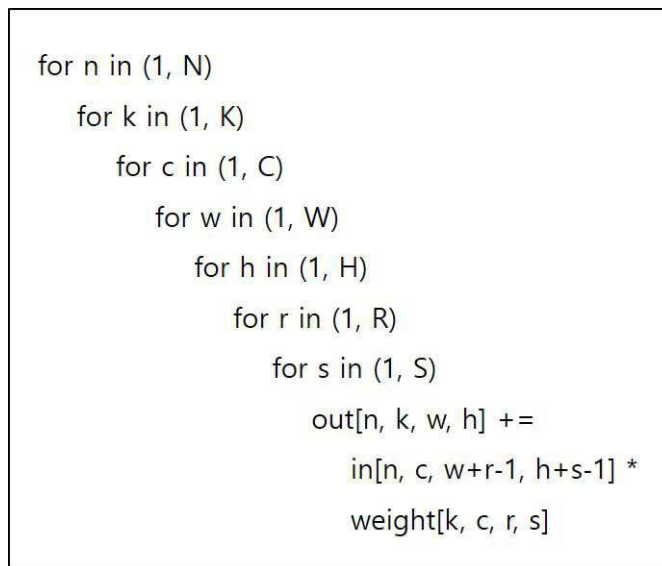
도면2



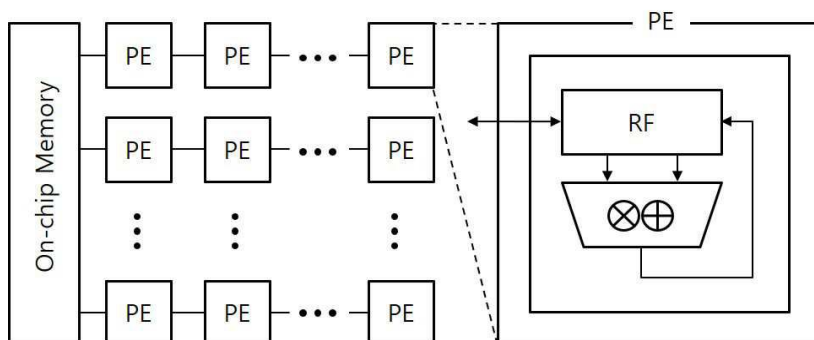
도면3



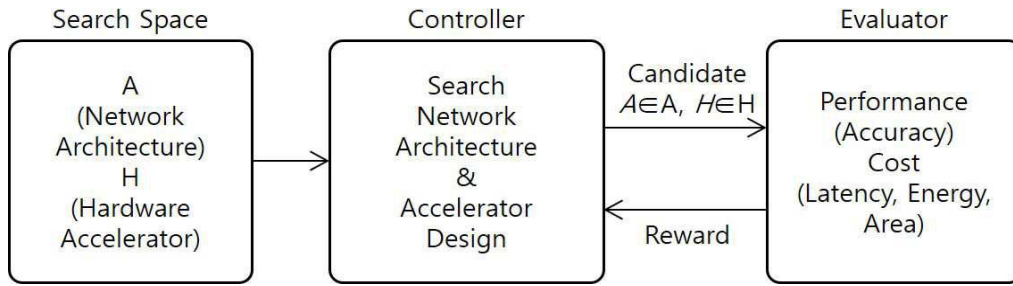
도면4



도면5

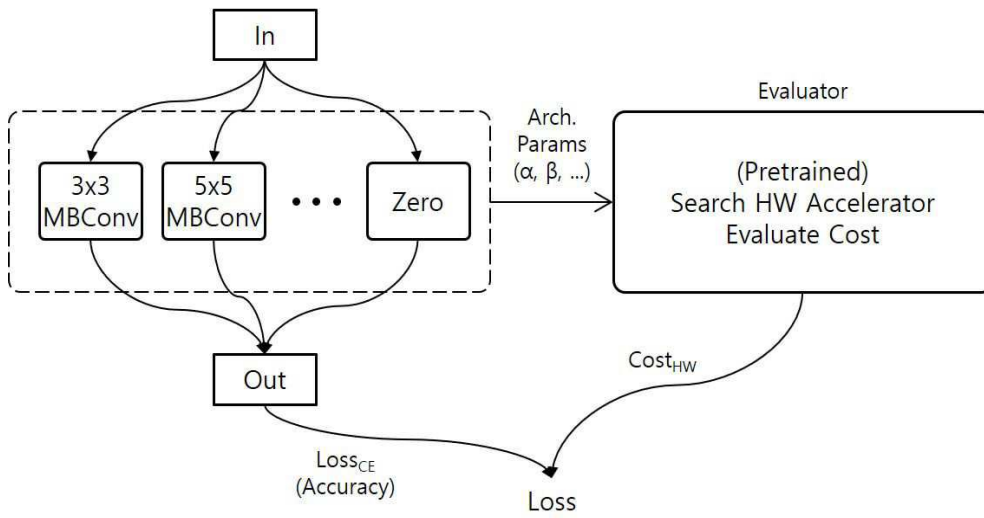


도면6

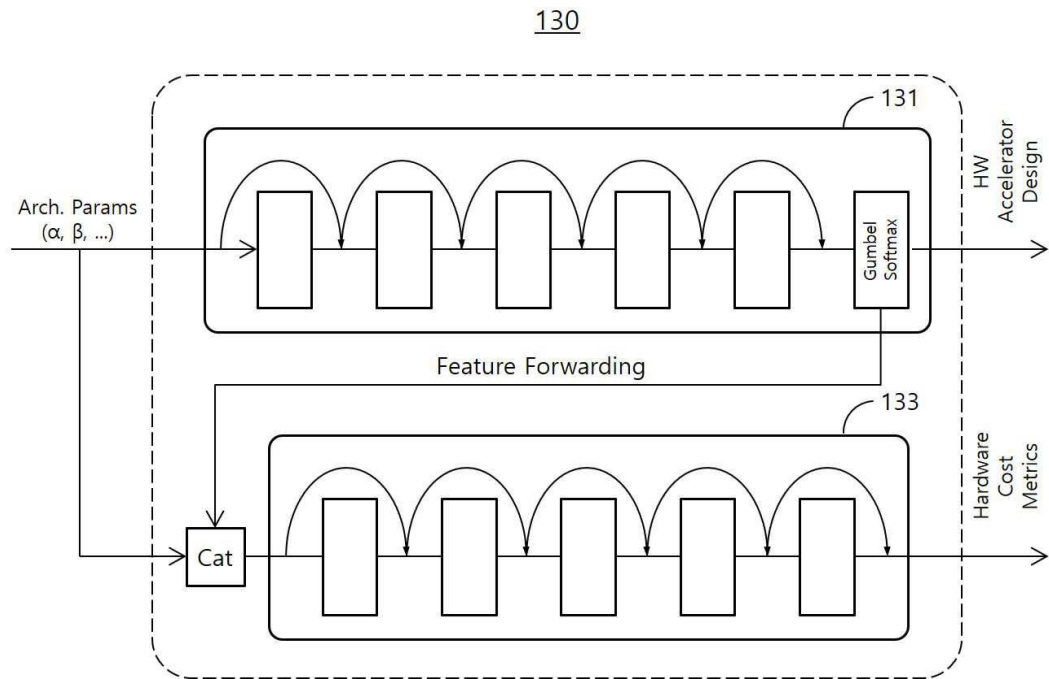


도면7

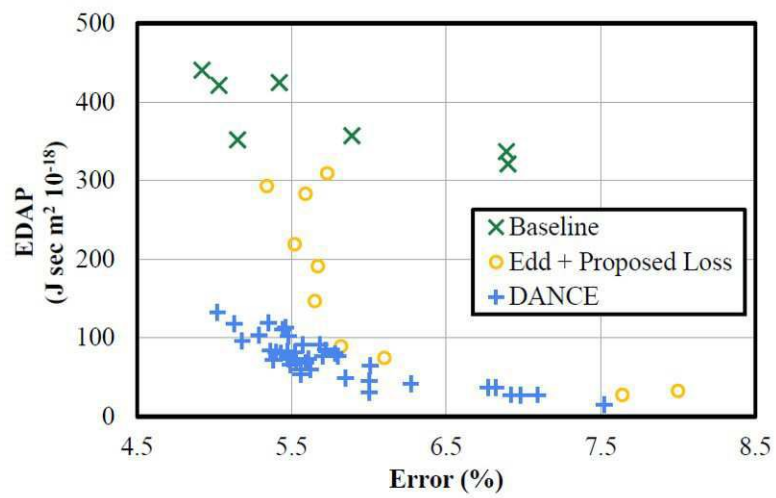
100



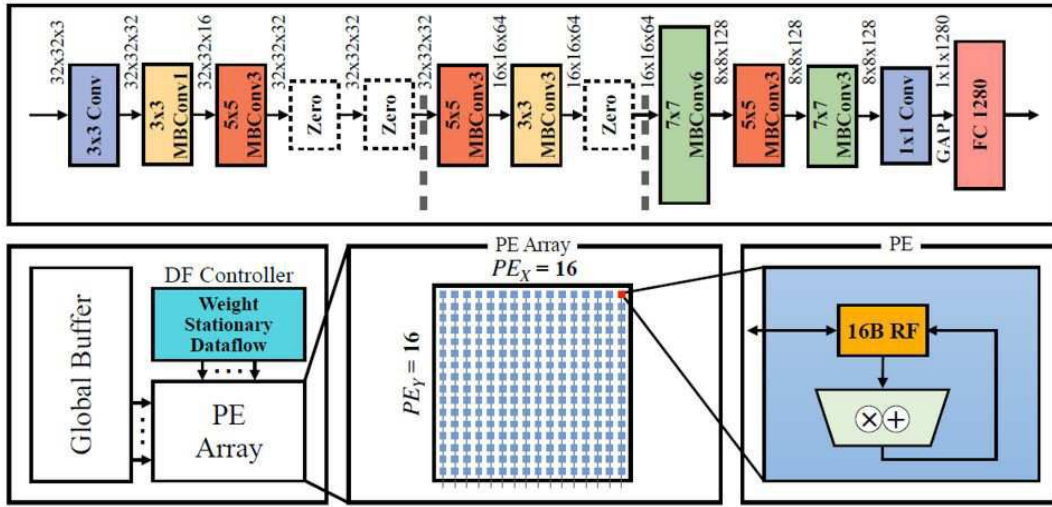
도면8



도면9



도면10



도면11

