(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2021-0138954
(43) 공개일자 2021년11월22일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) *G06F 17/15* (2006.01)
G06N 3/04 (2006.01)
(52) CPC특허분류
G06N 3/08 (2013.01)
G06F 17/153 (2013.01)
(21) 출원번호 10-2020-0057082
(22) 출원일자 2020년05월13일
심사청구일자 없음

(71) 출원인
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
강석주
서울특별시 마포구 백범로 35(신수동)
안세현
서울특별시 강동구 진황도로61길 7, 102동 402호 (둔촌동, 현대아파트)
(74) 대리인
유미특허법인

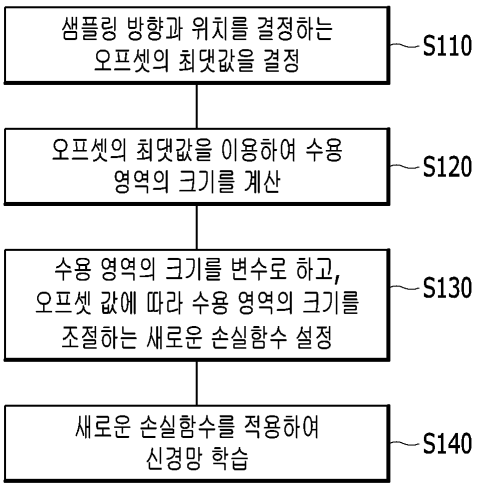
전체 청구항 수 : 총 9 항

(54) 발명의 명칭 **변형 가능한 합성곱 신경망의 설계 방법 및 이를 이용한 장치**

(57) 요 약

적어도 하나의 프로세서에 의해 동작하는 컴퓨팅 장치가 합성곱 신경망을 학습시키는 방법으로서, 입력 특징맵의 샘플링의 방향과 위치를 포함하는 오프셋 정보를 결정하는 단계, 상기 오프셋 정보에 포함된 오프셋의 최댓값에 비례하는 수용 영역의 크기를 변수로 하는 손실함수를 생성하는 단계, 상기 오프셋 정보를 기초로 상기 입력 특징맵의 데이터를 샘플링하여 새로운 입력 특징맵을 생성하고, 상기 새로운 입력 특징맵과 컨볼루션 필터를 이용하여 출력 특징맵을 생성하는 합성곱 연산을 수행하는 단계, 그리고 상기 손실함수를 이용하여 상기 합성곱 신경망을 최적화시키는 단계를 포함한다.

대 표 도 - 도4



(52) CPC특허분류
G06N 3/04 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 20PQWO-B153369-02
과제번호 202055002
부처명 행정안전부
과제관리(전문)기관명 국토교통과학기술진흥원
연구사업명 스마트 도로조명 플랫폼 개발 및 실증연구
연구과제명 스마트 도로조명 활용 도시재난안전관리 연계 기술 개발
기 여 율 1/2
과제수행기관명 서강대학교 산학협력단
연구기간 2020.01.01~2020.12.31
이 발명을 지원한 국가연구개발사업
과제고유번호 1711116161
과제번호 201939015
부처명 과학기술정보통신부
과제관리(전문)기관명 정보통신기획평가원
연구사업명 대학ICT연구센터지원사업
연구과제명 인공지능 서비스 실현을 위한 지능형 반도체 설계 핵심기술 개발
기 여 율 1/2
과제수행기관명 서강대학교 산학협력단
연구기간 2020.01.01~2021.12.31

명 세 서

청구범위

청구항 1

적어도 하나의 프로세서에 의해 동작하는 컴퓨팅 장치가 합성곱 신경망을 학습시키는 방법으로서,

입력 특징맵의 샘플링의 방향과 위치를 포함하는 오프셋 정보를 결정하는 단계,

상기 오프셋 정보에 포함된 오프셋의 최댓값에 비례하는 수용 영역의 크기를 변수로 하는 손실함수를 생성하는 단계,

상기 오프셋 정보를 기초로 상기 입력 특징맵의 데이터를 샘플링하여 새로운 입력 특징맵을 생성하고, 상기 새로운 입력 특징맵과 컨볼루션 필터를 이용하여 출력 특징맵을 생성하는 합성곱 연산을 수행하는 단계, 그리고

상기 손실함수를 이용하여 상기 합성곱 신경망을 최적화시키는 단계

를 포함하는, 학습 방법.

청구항 2

제1항에서,

상기 새로운 입력 특징맵을 생성하는 단계는,

상기 입력 특징맵에서 샘플링된 값들을 선형 보간(Linear Interpolation)하는, 학습 방법.

청구항 3

제1항에서,

상기 합성곱 신경망은 변형 가능한 합성곱 신경망(Deformable Convolutional Networks)인, 학습 방법.

청구항 4

제1항에서,

상기 최댓값은 상기 오프셋의 절댓값의 최댓값인, 학습 방법.

청구항 5

합성곱 연산을 수행하는 컴퓨팅 장치로서,

입력 특징맵과, 상기 입력 특징맵의 샘플링의 방향과 위치를 결정하는 오프셋 정보를 포함하는 버퍼,

상기 오프셋 정보를 기초로 상기 입력 특징맵에서의 샘플링 위치를 결정하고, 상기 버퍼로부터 상기 샘플링 위치에 해당하는 값을 제공받아 새로운 입력 특징맵을 생성하는 샘플링 처리기, 그리고

합성곱 연산을 수행하여 상기 새로운 입력 특징맵으로부터 출력 특징맵을 생성하는 시스톨릭 어레이

를 포함하는, 컴퓨팅 장치.

청구항 6

제5항에서,

상기 오프셋 정보는,

상기 시스톨릭 어레이에 의해 상기 입력 특징맵과 오프셋 필터가 컨볼루션 된 결과값인, 컴퓨팅 장치.

청구항 7

제6항에서,

상기 시스톨릭 어레이는,

상기 입력 특징맵을 복수의 타일들로 분할하고, 각 타일과 상기 오프셋 필터를 컨볼루션하는, 컴퓨팅 장치.

청구항 8

제5항에서,

상기 샘플링 처리기는,

특정 샘플링 위치의 주변 픽셀들의 값을 선형 보간(Linear Interpolation)하는, 컴퓨팅 장치.

청구항 9

제5항에서,

상기 출력 특징맵은,

상기 새로운 입력 특징맵과 변형 가능한 컨볼루션 필터의 연산 결과인, 컴퓨팅 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 변형 가능한 합성곱 신경망을 설계하는 기술에 관한 것이다.

배 경 기 술

[0002] 딥러닝 기반 알고리즘은 객체 인식, 분류, 그리고 음성 인식 등 다양한 분야에서 뛰어난 성능을 보여주고 있다. 특히 합성곱 신경망(Convolutional Neural Networks, CNN)은 이미지의 특징을 추출하고 객체를 검출하기 위한 객체 인식 모델에 사용된다. 이러한 객체 인식 모델의 앞부분에 여러 개의 컨볼루션 레이어들이 위치한 아키텍처를 백본 네트워크(Backbone Network)라고 호칭한다.

[0003] 백본 네트워크는 입력 이미지로부터 고차원의 특징을 추출하는 역할을 하며 최근 백본 네트워크의 중요성이 알려짐에 따라 특징 추출의 성능을 높이기 위한 연구가 활발히 이루어지고 있다.

[0004] 여러 연구들 중, 변형 가능한 합성곱 신경망(Deformable Convolutional Networks, 이하 'DCN'이라고 호칭함)은 ResNet 기반의 아키텍처에서 컨볼루션 레이어의 일부를 변형 가능한 컨볼루션 레이어(Deformable Convolutional Layer, 이하 'DCL'이라고 호칭함)로 대체하여 특징 추출의 성능을 높였고, 결과적으로 객체 인식의 성능을 상당히 개선시켰다. 따라서 DCN을 포함하는 알고리즘을 이용하여 하드웨어 가속기를 설계하여 저전력 및 빠른 연산을 요구하는 시스템에 적용할 필요가 있다.

[0005] 그러나 DCL은 일반적인 컨볼루션 레이어와는 달리, 입력 특징맵에서 불규칙적으로 샘플링하기 때문에 수용 영역이 변한다. 따라서 하드웨어 가속기를 구현할 경우, 수용 영역의 크기가 커서 하드웨어의 입력 버퍼 내의 데이터만으로는 연산이 불가능하게 된다.

[0006] 그 결과 추가적으로 외부 DRAM에 랜덤하게 접근할 수 있다. 랜덤한 DRAM 접근은 순차적인 DRAM 접근보다 많은 에너지를 소모하기 때문에 효율성을 저하시킨다.

[0007] 또한 랜덤한 DRAM 접근을 위해서는 DRAM과 온칩(On-chip) 메모리 간의 트래픽을 조정하기 위한 추가적인 제어 로직이 요구된다. 그리고 DRAM에 접근하는 과정에서 파이프라인이 중단될 수 있다. 이 과정에서 처리 소자(Processing Element)는 유휴 모드(Idle mode)로 전환되어 하드웨어 자원을 완전히 사용하지 못한다. 이는 결국 에너지 소비량의 증가와 속도의 저하를 야기한다.

[0008] 따라서 DCN의 불규칙적인 샘플링이라는 특성을 고려하되, 하드웨어 성능을 저하시키지 않는 가속기를 설계할 필요가 있다.

발명의 내용

해결하려는 과제

[0009] 해결하고자 하는 과제는 변형 가능한 컨볼루션 레이어의 수용 영역 크기를 조절하는 손실함수를 생성하고, 생성

된 손실함수를 이용하여 변형 가능한 합성곱 신경망을 학습시키는 방법을 제공하는 것이다.

[0010] 또한 해결하고자 하는 과제는 변형 가능한 합성곱 신경망을 포함하는 인공신경망 가속기(Accelerator)를 제공하는 것이다.

과제의 해결 수단

[0011] 한 실시예에 따른 적어도 하나의 프로세서에 의해 동작하는 컴퓨팅 장치가 합성곱 신경망을 학습시키는 방법으로서, 입력 특징맵의 샘플링의 방향과 위치를 포함하는 오프셋 정보를 결정하는 단계, 상기 오프셋 정보에 포함된 오프셋의 최댓값에 비례하는 수용 영역의 크기를 변수로 하는 손실함수를 생성하는 단계, 상기 오프셋 정보를 기초로 상기 입력 특징맵의 데이터를 샘플링하여 새로운 입력 특징맵을 생성하고, 상기 새로운 입력 특징맵과 컨볼루션 필터를 이용하여 출력 특징맵을 생성하는 합성곱 연산을 수행하는 단계, 그리고 상기 손실함수를 이용하여 상기 합성곱 신경망을 최적화시키는 단계를 포함한다.

[0012] 상기 새로운 입력 특징맵을 생성하는 단계는, 상기 입력 특징맵에서 샘플링된 값들을 선형 보간(Linear Interpolation)할 수 있다.

[0013] 상기 합성곱 신경망은 변형 가능한 합성곱 신경망(Deformable Convolutional Networks)일 수 있다.

[0014] 상기 최댓값은 상기 오프셋의 절댓값의 최댓값일 수 있다.

[0015] 한 실시예에 따른 합성곱 연산을 수행하는 컴퓨팅 장치로서, 입력 특징맵과, 상기 입력 특징맵의 샘플링의 방향과 위치를 결정하는 오프셋 정보를 포함하는 버퍼, 상기 오프셋 정보를 기초로 상기 입력 특징맵에서의 샘플링 위치를 결정하고, 상기 버퍼로부터 상기 샘플링 위치에 해당하는 값을 제공받아 새로운 입력 특징맵을 생성하는 샘플링 처리기, 그리고 합성곱 연산을 수행하여 상기 새로운 입력 특징맵으로부터 출력 특징맵을 생성하는 시스톨릭 어레이를 포함한다.

[0016] 상기 오프셋 정보는, 상기 시스톨릭 어레이에 의해 상기 입력 특징맵과 오프셋 필터가 컨볼루션 된 결과값일 수 있다.

[0017] 상기 시스톨릭 어레이는, 상기 입력 특징맵을 복수의 타일들로 분할하고, 각 타일과 상기 오프셋 필터를 컨볼루션할 수 있다.

[0018] 상기 샘플링 처리기는, 특정 샘플링 위치의 주변 픽셀들의 값을 선형 보간(Linear Interpolation)할 수 있다.

[0019] 상기 출력 특징맵은, 상기 새로운 입력 특징맵과 변형 가능한 컨볼루션 필터의 연산 결과일 수 있다.

발명의 효과

[0020] 본 발명에 따르면 성능의 저하 없이 변형 가능한 컨볼루션 레이어의 수용 영역의 크기를 줄일 수 있으므로, 불규칙적인 DRAM 접근을 줄일 수 있어 가속기의 속도와 처리량을 향상시키고, 에너지를 절약할 수 있다.

[0021] 또한 본 발명에 따르면, 종래의 가속기와 달리 입력 버퍼에 저장된 데이터를 재사용할 수 있으므로, 외부 메모리로의 랜덤한 접근을 줄이면서 파이프라인의 중단을 방지하여 가속기의 속도와 성능을 향상시킬 수 있다.

도면의 간단한 설명

[0022] 도 1은 기존의 컨볼루션 레이어의 설명도이다.

도 2와 도 3은 한 실시예에 따른 변형 가능한 컨볼루션 레이어의 설명도이다.

도 4는 한 실시예에 따른 인공신경망의 손실함수를 수정하는 방법의 흐름도이다.

도 5는 한 실시예에 따른 인공신경망 가속기와 주변 환경의 구성도이다.

도 6은 한 실시예에 따른 시스톨릭 어레이의 처리기의 구성도이다.

도 7은 한 실시예에 따른 인공신경망 가속기의 동작 방법의 흐름도이다.

도 8과 도 9는 한 실시예에 따른 인공신경망 가속기의 성능을 평가한 설명도이다.

도 10은 한 실시예에 따른 컴퓨팅 장치의 하드웨어 구성도이다.

발명을 실시하기 위한 구체적인 내용

[0023] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0024] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다. 또한, 명세서에 기재된 "…부", "…기", "모듈" 등의 용어는 적어도 하나의 기능이나 동작을 처리하는 단위를 의미하며, 이는 하드웨어나 소프트웨어 또는 하드웨어 및 소프트웨어의 결합으로 구현될 수 있다.

[0025] 도 1은 기존의 컨볼루션 레이어의 설명도이고, 도 2와 도 3은 한 실시예에 따른 변형 가능한 컨볼루션 레이어의 설명도이다.

[0026] 도 1을 참고하면, 일반적인 컨볼루션 레이어는 입력 특징맵(Input Feature Map)의 정해진 위치에서 데이터를 규칙적으로 샘플링한다. 그러나 DCL은 도 2와 같이 입력 특징맵에서 불규칙적인 데이터 샘플링을 한다. 도 1의 빨간 박스와 도 2의 빨간 박스는 각각 수용 필드(Receptive Field)를 나타낸다.

[0027] 예를 들어, 3x3의 커널 크기를 갖는 경우 DCL의 샘플링 과정은 도 3과 같다. 빨간 화살표는 샘플링 방향과 거리를 의미하고, 사각형들은 DCL에서의 샘플링 위치를 의미한다.

[0028] 도 1의 입력 특징맵과 오프셋 필터의 컨볼루션 연산 결과, 샘플링 위치를 결정하는 오프셋을 얻을 수 있다. 이후 도 1의 입력 특징맵에서 오프셋의 샘플링 위치 정보를 이용하여 도 2의 입력 특징맵을 얻을 수 있다. 따라서 도 2의 입력 특징맵은 샘플링 방향과 거리에 대한 정보를 포함하며, 위의 과정은 수학식 1과 수학식 2로 나타낼 수 있다.

수학식 1

[0029] $$y = f(g(x, o), w_{deform})$$

[0030] 수학식 1에서 g는 샘플링 과정을 의미하고, **y**는 출력 특징맵(Output Feature Map)의 텐서를 의미하고, $\mathbf{w}_{deform}$은 변형 가능한 컨볼루션 필터를 의미한다.

수학식 2

[0031] $$o = f(x, w_o)$$

[0032] 수학식 2에서, f는 컨볼루션 연산을 의미하고, **x**는 입력 특징맵의 텐서를 의미하고, $\mathbf{w}_o$는 오프셋 필터를 의미하고, **o**는 오프셋을 의미한다.

[0033] 즉, 입력 특징맵과 오프셋 필터를 컨볼루션 연산하여 오프셋을 생성하고, 생성된 오프셋 값을 기초로 입력 특징맵에서 데이터를 샘플링하여 새로운 입력 특징맵을 생성한다. 샘플링을 통해 얻은 새로운 입력 특징맵과 변형 가능한 컨볼루션 필터를 다시 컨볼루션 연산하여 출력 특징맵을 생성한다.

[0034] 수학식 1과 수학식 2를 참고하면, 오프셋 값이 0에 가까워질수록 수용 영역의 크기도 감소함을 알 수 있다. 따라서 이하에서는 수용 영역의 크기를 줄이는 법에 대해 설명한다.

[0035] 도 4는 한 실시예에 따른 인공신경망의 손실함수를 수정하는 방법의 흐름도이다.

[0036] 도 4를 참고하면, 컴퓨팅 장치는 오프셋의 최댓값을 설정한다(S110). 오프셋의 부호는 샘플링 방향을 의미하므로, 최댓값은 오프셋의 절댓값으로 계산될 수 있고 수학식 3으로 계산될 수 있다.

수학식 3

[0037] $$o_{\max} = \max_{o_i \in o} abs(o_i)$$

[0038] 컴퓨팅 장치는 오프셋의 최댓값을 이용하여 수용 영역의 크기를 계산한다(S120). DCL에서 수용 영역의 크기는 수학식 4에 의해 결정된다.

수학식 4

[0039] $$RF = K_c + 2 \times ceil(o_{\max})$$

[0040] 수학식 4에서, RF는 수용 영역의 크기를 의미하고, K_c는 DCL의 커널 사이즈를 의미한다. 한편 4개의 픽셀 간 거리 중 가장 큰 값은 ceil(o_{max})이고, ceil 함수는 이는 이중 선형 보간법(Bilinear Interpolation)을 수행하는 함수를 의미한다.

[0041] 이중 선형 보간법을 사용하는 이유는, 오프셋에 저장된 값은 실수이므로 샘플링 위치가 정수로 계산되지 않을 수 있기 때문에 입력 특징맵의 주변 픽셀 값들을 참고하여 샘플링 데이터를 계산하는 것이다.

[0042] 컴퓨팅 장치는 수용 영역의 크기를 변수로 하고, 오프셋 값에 따라 수용 영역의 크기를 조절하는 새로운 손실함수를 설정한다(S130). 새로운 손실함수는 DCN의 학습 과정에서 사용되고, 수학식 5에 의해 계산된다.

수학식 5

[0043] $$Loss = (1-\lambda) \times L + \lambda \times \max_{l \in D} o_{max}^{l} \quad for\ 0 \le \lambda < 1$$

[0044] λ는 o_{max}^{l}를 새로운 손실함수에 얼마나 많이 반영할지를 결정하는 하이퍼 파라미터이고, L은 기존의 손실함수를 의미하고, D는 백본 네트워크의 DCL 집합을 의미한다. 그리고 o_{max}^{l}는 백본 네트워크의 DCL에서 절댓값이 가장 큰 오프셋을 의미한다.

[0045] λ는 사용자에 의해 변경될 수 있는 값이고, 실험을 통해 최적의 값이 결정될 수 있다.

[0046] 컴퓨팅 장치는 설정된 새로운 손실함수를 적용하여 신경망을 학습시킨다(S140).

[0047] DCL에서는 오프셋을 획득한 후, 이중 선형 보간을 수행하기 위해 DRAM에 랜덤하게 접근하여 데이터를 읽는다. CNN은 불규칙한 입력 특징맵 샘플링을 전혀 고려하지 않기 때문에, 기존의 CNN 구조를 그대로 DCL에 적용하는 것은 적절하지 않다. 따라서 DRAM과 온칩 메모리 간 대역폭을 줄이기 위해, 도 4에서 설명한 손실함수가 적용된 DCL 가속기(10)가 요구된다. 이하에서는 새로운 손실함수가 적용된 DCL 가속기(10)의 구조와 동작 방법에 대해 설명한다.

[0048] 도 5는 한 실시예에 따른 인공신경망 가속기와 주변 환경의 구성도이고, 도 6은 한 실시예에 따른 시스톨릭 어레이의 처리기의 구성도이고, 도 7은 한 실시예에 따른 인공신경망 가속기의 동작 방법의 흐름도이다.

[0049] 도 5를 참고하면, FPGA를 이용하여 구현한 시스톨릭 어레이(200) 기반의 가속기(10)를 나타낸 것이다.

[0050] 가속기(10)는 입력 버퍼(100), 복수의 처리기(210)들을 포함하는 시스톨릭 어레이(200), 출력 버퍼(300), 샘플링 처리기(400)를 포함한다. 한편, 입력 버퍼(100), 복수의 처리기(210)들을 포함하는 시스톨릭 어레이(200), 출력 버퍼(300), 샘플링 처리기(400)는 각각은 하나의 소프트웨어 또는 하나의 인공지능 모델이거나 복수의 인공지능 모델로 구현될 수 있다.

[0051] 입력 버퍼(100)는 입력 특징맵 데이터를 저장한다. 그리고, 랜덤 DRAM 접근 없이 작동하기 위해서 입력 버퍼

(100) 크기는 수용 영역의 크기보다 커야 한다. 구체적으로 입력 버퍼(100)의 크기는 수학식 6에 의해 결정될 수 있다.

수학식 6

[0052] $$RF \times (S \cdot T_w + K_c - S) \times T_N$$

[0053] 수학식 6에서, RF는 수용 영역의 크기, S는 스트라이드 즉 필터의 이동 횟수를 의미하고, T_W는 분할된 각 타일의 폭, K_c는 DCL의 커널 크기를 의미하고, T_N는 입력 타일의 채널 수를 의미한다.

[0054] 시스톨릭 어레이(200)는 초대형 집적 회로(VLSI)기술을 이용하여 특정 알고리즘의 수행 속도를 향상시키기 위한 전용 하드웨어이다. 또한 파이프라인 어레이 구조의 한 범주로써, 지역적으로 연결된 프로세서들이 규칙적으로 데이터를 계산하고 시스템을 통해 데이터를 전달하는 프로세서 네트워크를 의미한다.

[0055] 시스톨릭 어레이(200)는 규칙적이고, 지역적인 데이터의 흐름을 갖고, 계산량이 많은 알고리즘에 적용될 수 있다. 따라서 본 발명은 버퍼와 처리기(210) 간 대역폭을 감소시키기 위해 시스톨릭 어레이(200) 구조를 사용할 수 있다.

[0056] 시스톨릭 어레이(200)는 복수의 처리기(210)들을 포함하고, 각 처리기(210)는 병렬 처리기로 구현될 수 있다. 본 명세서에서, 시스톨릭 어레이(200)의 처리기(210)와 샘플링 처리기(400)를 구분하여 설명하고, 편의상 별도의 설명이 없는 처리기(210)는 시스톨릭 어레이(200)의 처리기(210)를 의미한다.

[0057] 도 6을 참고하면, 크기가 M_W x M_H인 2차원 시스톨릭 어레이(200)의 (x, y)에 위치한 처리기(210)의 구조를 나타낸다. 처리기(210)에 저장된 입력 데이터와 가중치 값들은 매 클럭마다 이웃한 처리기(210)로 전송된다.

[0058] 출력 버퍼(300)는 오프셋 및 샘플링 된 입력 데이터를 저장하고, 한 예로서 출력 버퍼(300)의 크기는 $T_W \times 9 \times T_N$으로 계산될 수 있다.

[0059] 샘플링 처리기(Sampling Processing Element)(400)는 출력 버퍼(300)에 저장된 데이터와 오프셋 값을 이용하여 샘플링 위치를 계산한다. 샘플링 위치는 입력 버퍼(100)의 주소로 변환될 수 있다.

[0060] 샘플링 처리기(400)는 각 주소에 저장된 데이터를 수신하고 이중 선형 보간을 진행한다. 이후 샘플링 된 입력 데이터는 출력 버퍼(300)에 저장된다.

[0061] 도 7을 참고하면, 시스톨릭 어레이(200)의 처리기(Processing Elements, PE)(210)는 입력 특징맵과 오프셋 필터를 컨볼루션하고 그 결과를 출력 버퍼(300)에 저장한다(S210).

[0062] 이때 하드웨어 자원을 효율적으로 사용하기 위해 입력 특징맵을 타일로 분할하는 루프 타일링(Loop Tiling) 방법이 사용될 수 있다. 즉 입력 특징맵을 타일로 분할한 입력 타일과 오프셋 필터를 컨볼루션할 수 있다.

[0063] 한 예로서, 입력 타일의 채널 수(T_N), 출력 타일의 채널 수(T_M), 분할된 각 타일의 높이(T_H), 분할된 각 타일의 폭(T_W)의 값을 결정하기 위해 루프라인 모델(Roofline Model)이 사용될 수 있다.

[0064] 루프라인 모델이란, 알고리즘의 메모리 전송량과 계산량의 비율을 바탕으로 특정 가속기 아키텍처가 달성 가능한 최대 성능(Roofline)을 나타낸 것이다. 본 발명에서 제안하는 가속기(10)의 구조는 변경 가능한 두 부분으로 분리될 수 있으므로, 각 부분에 대해 계산할 수 있다.

[0065] 루프라인 모델에 따라, T_N은 1, T_M은 64, T_H은 1, T_W는 8로 결정될 수 있다.

[0066] 샘플링 처리기(400)는 출력 버퍼(300)에 저장된 결과값인 오프셋을 이용하여 샘플링 위치를 계산하고, 해당 샘플링 위치의 픽셀값을 입력 버퍼에서 읽어온다(S220).

[0067] 샘플링 처리기(400)는 각 입력 버퍼(100) 주소에 저장된 결과값을 이중 선형 보간하여 새로운 입력 특징맵을 생성하고 출력 버퍼(300)에 저장한다(S230). 이중 선형 보간은 수직선 사에서 두 점 사이의 임의의 점의 함수값을 구하는 선형 보간(Linear Interpolation)을 2차원으로 확장한 개념이며, 특정 위치의 데이터 값을 추정할 때 사

용된다.

[0068] 입력 특징맵의 샘플링 과정이 끝나면, 시스톨릭 어레이(200)의 처리기(210)는 S230 단계에서 생성된 새로운 입력 특징맵과 변형 가능한 컨볼루션 필터를 컨볼루션하여 출력 특징맵을 생성한다(S240).

[0069] 도 8과 도 9는 한 실시예에 따른 인공신경망 가속기의 성능을 평가한 설명도이다.

[0070] 기존의 가속기에 본 발명이 제안하는 손실함수를 적용하는 경우와 본 발명이 제안하는 새로운 구조의 가속기(10)와 새로운 손실함수를 동시에 이용하는 경우에 대해, 필터의 개수를 변경하며 속도와 에너지 소비량을 측정하는 실험을 진행하였다.

[0071] 도 8을 참고하면, 기존 가속기와 본 발명에서 제안하는 가속기(10)에 모두 새로운 손실함수를 적용하여, λ 값을 변화시키면서 속도를 측정한 그래프이다.

[0072] 기존 가속기의 경우, DRAM에 불규칙적으로 접근하므로 파이프라인의 지연에 따라 가속기의 속도가 감소한다. 반면, 본 발명이 제안하는 DCL 기반의 가속기(10)는 필터의 개수가 증가함에 따라, 데이터를 재사용하여 성능이 향상하였다. 본 발명이 제안하는 가속기(10)는 기존의 가속기에 비해 17.25배 더 빠른 속도를 나타냈다.

[0073] 도 9를 참고하면, 기존 가속기와 본 발명에서 제안하는 가속기(10)에 모두 새로운 손실함수를 적용하여, λ값을 변화시키면서 에너지 소비량을 측정한 그래프이다.

[0074] 기존의 가속기는, 불규칙한 DRAM 접근으로 인해 λ값에 무관하게 에너지 소비가 비슷하게 나타났다. 이에 비해, 본 발명이 제안하는 DCL 기반의 가속기(10)에 새로운 손실함수가 적용되는 경우, 기존의 가속기에 비해 1.39배 에너지를 절약할 수 있다.

[0075] 도 10은 한 실시예에 따른 컴퓨팅 장치의 하드웨어 구성도이다.

[0076] 도 10을 참고하면, 적어도 하나의 프로세서에 의해 동작하는 컴퓨팅 장치(500)는 본 발명의 동작을 실행하도록 기술된 명령들(instructions)이 포함된 프로그램을 실행한다.

[0077] 컴퓨팅 장치(500)의 하드웨어는 적어도 하나의 프로세서(510), 메모리(520), 스토리지(530), 통신 인터페이스(540)를 포함할 수 있고, 버스를 통해 연결될 수 있다. 이외에도 입력 장치 및 출력 장치 등의 하드웨어가 포함될 수 있다. 컴퓨팅 장치(500)는 프로그램을 구동할 수 있는 운영 체제를 비롯한 각종 소프트웨어가 탑재될 수 있다.

[0078] 프로세서(510)는 컴퓨팅 장치(500)의 동작을 제어하는 장치로서, 프로그램에 포함된 명령들을 처리하는 다양한 형태의 프로세서(510)일 수 있고, 예를 들면, CPU(Central Processing Unit), MPU(Micro Processor Unit), MCU(Micro Controller Unit), GPU(Graphic Processing Unit) 등 일 수 있다. 메모리(520)는 본 발명의 동작을 실행하도록 기술된 명령들이 프로세서(510)에 의해 처리되도록 해당 프로그램을 로드한다. 메모리(520)는 예를 들면, ROM(read only memory), RAM(random access memory) 등 일 수 있다. 스토리지(530)는 본 발명의 동작을 실행하는데 요구되는 각종 데이터, 프로그램 등을 저장한다. 통신 인터페이스(540)는 유/무선 통신 모듈일 수 있다.

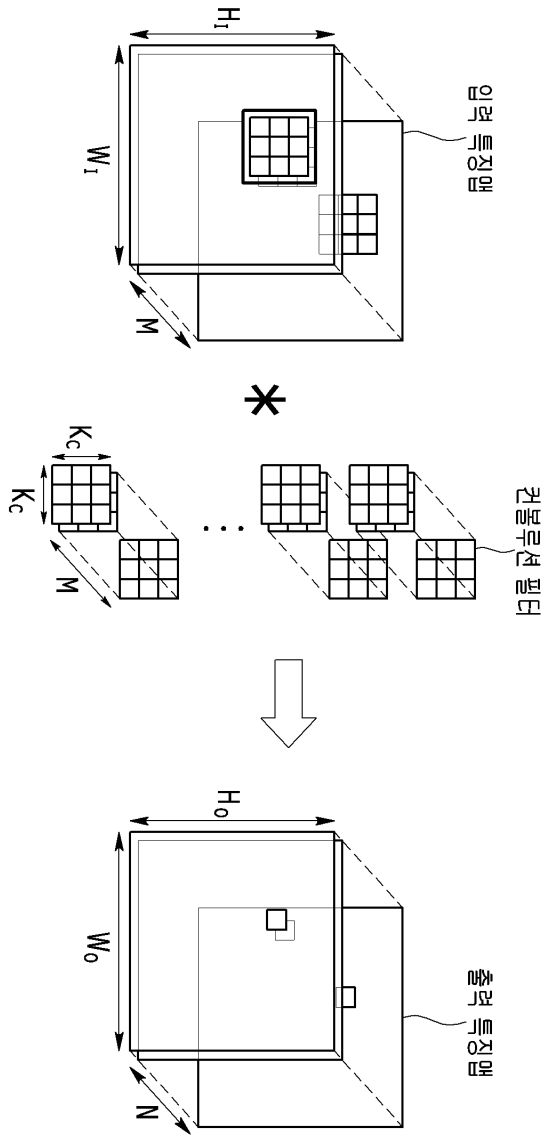
[0080] 본 발명에 따르면 성능의 저하 없이 변형 가능한 컨볼루션 레이어의 수용 영역의 크기를 줄일 수 있으므로, 불규칙적인 DRAM 접근을 줄일 수 있어 가속기의 속도와 처리량을 향상시키고, 에너지를 절약할 수 있다.

[0081] 이상에서 설명한 본 발명의 실시예는 장치 및 방법을 통해서만 구현이 되는 것은 아니며, 본 발명의 실시예의 구성에 대응하는 기능을 실현하는 프로그램 또는 그 프로그램이 기록된 기록 매체를 통해 구현될 수도 있다.

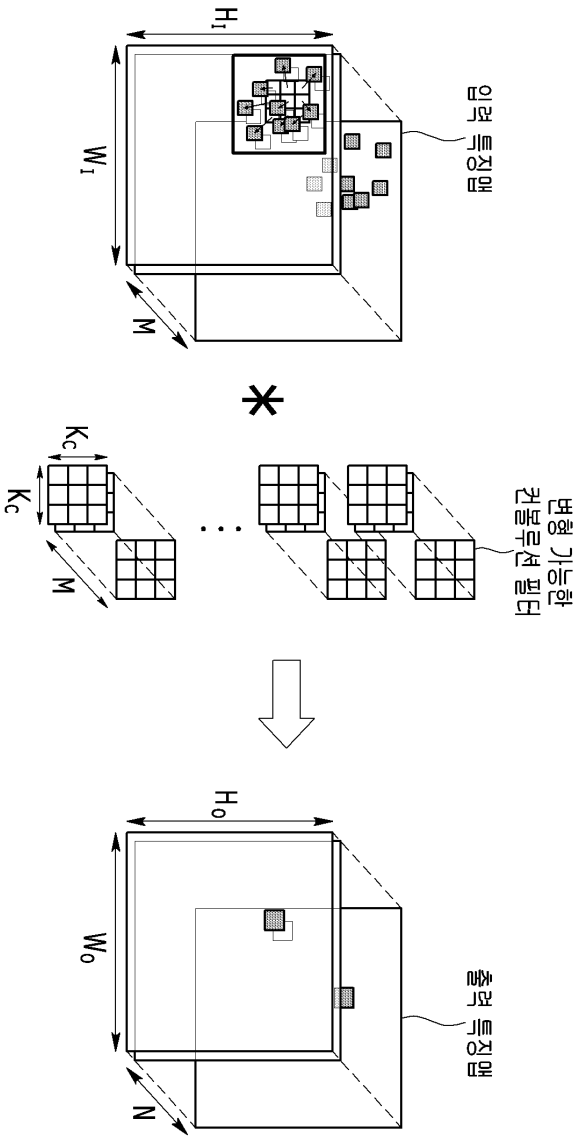
[0082] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

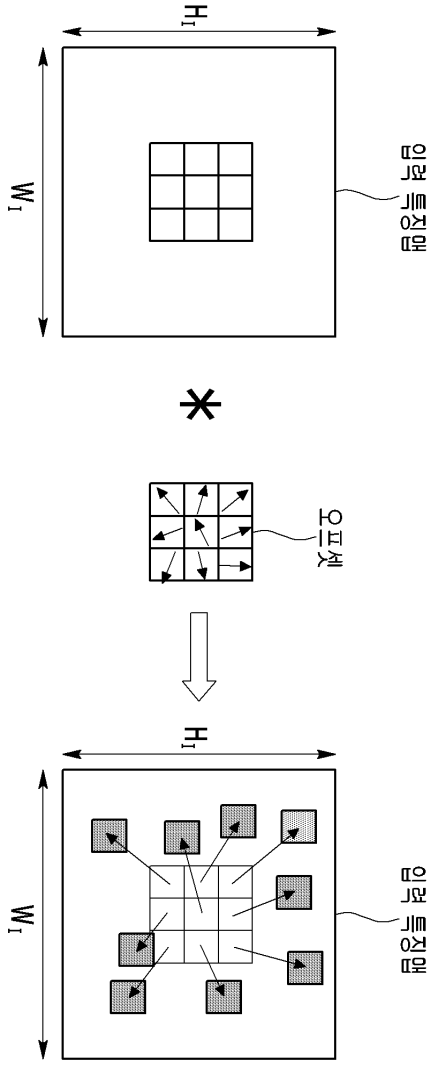
도면1



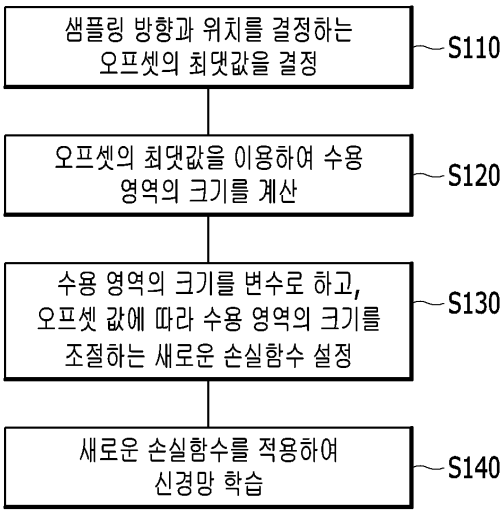
도면2



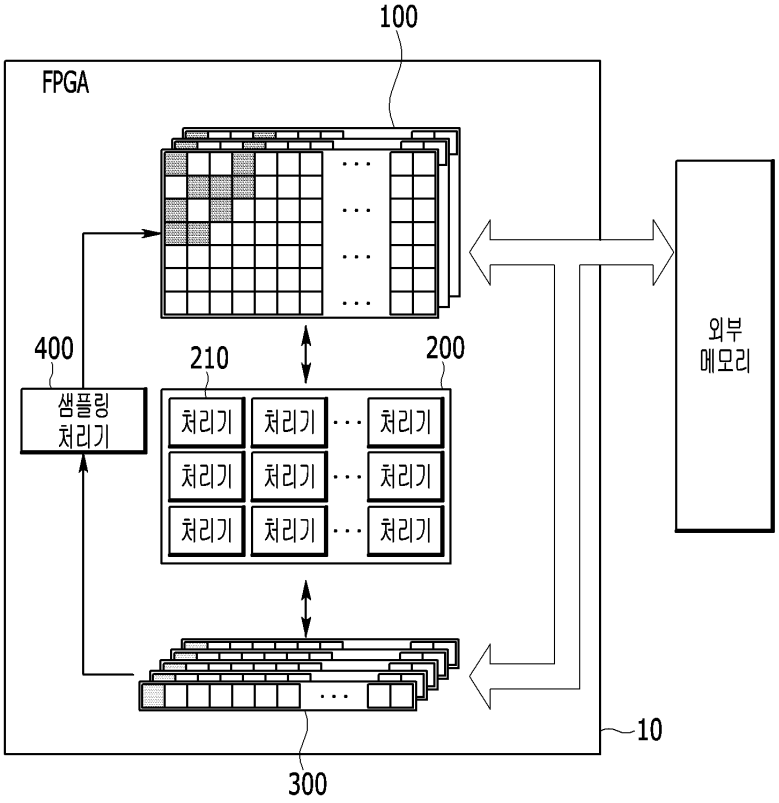
도면3


W_1
H_1
입력 특징맵
오프셋
W_1
H_1
출력 특징맵

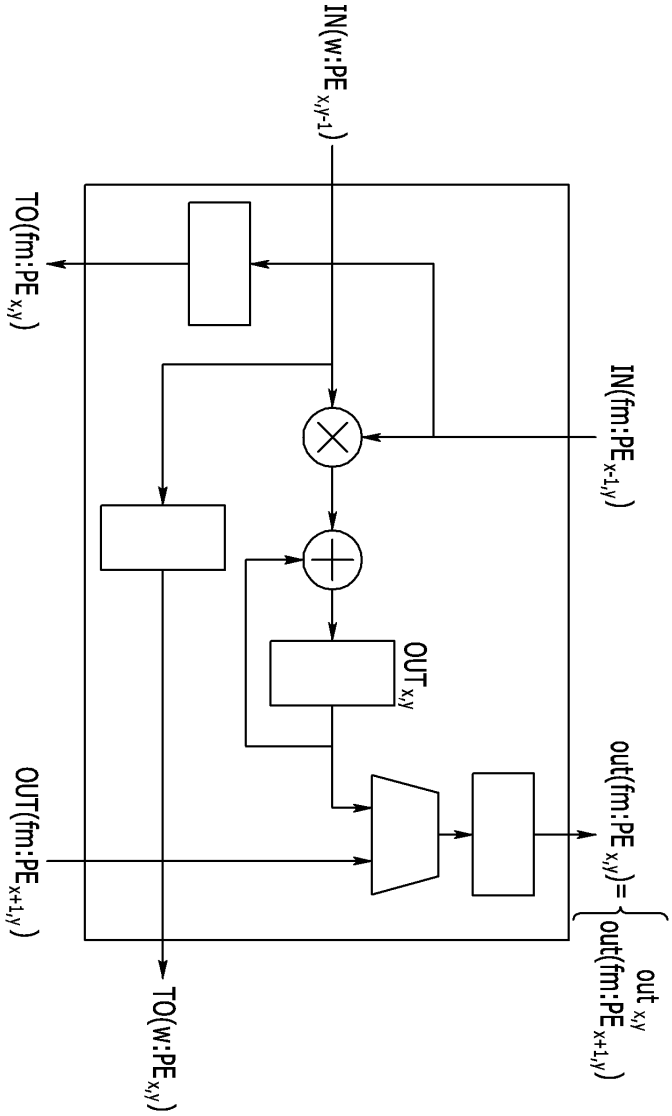
도면4


샘플링 방향과 위치를 결정하는 오프셋의 최댓값을 결정
S110
오프셋의 최댓값을 이용하여 수용 영역의 크기를 계산
S120
수용 영역의 크기를 변수로 하고, 오프셋 값에 따라 수용 영역의 크기를 조절하는 새로운 손실함수 설정
S130
새로운 손실함수를 적용하여 신경망 학습
S140

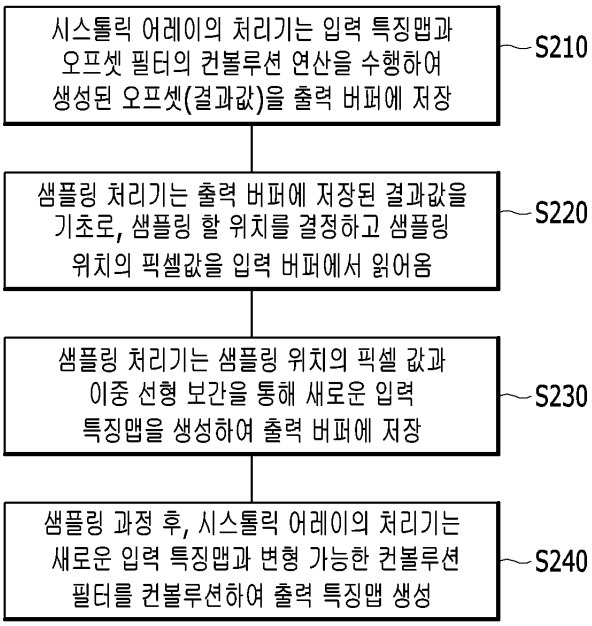
도면5



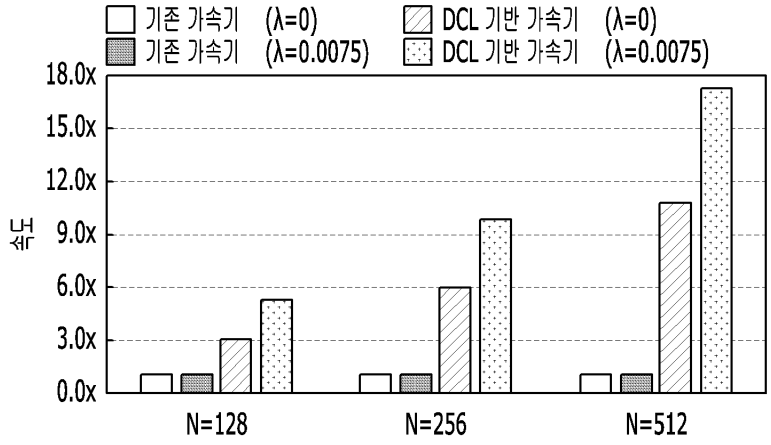
도면6


IN(w:PE x,y-1)
TO(fm:PE x,y)
IN(fm:PE x-1,y)
OUT x,y
OUT(fm:PE x+1,y)
out(fm:PE x,y)= { out x,y / out(fm:PE x+1,y)
TO(w:PE x,y)

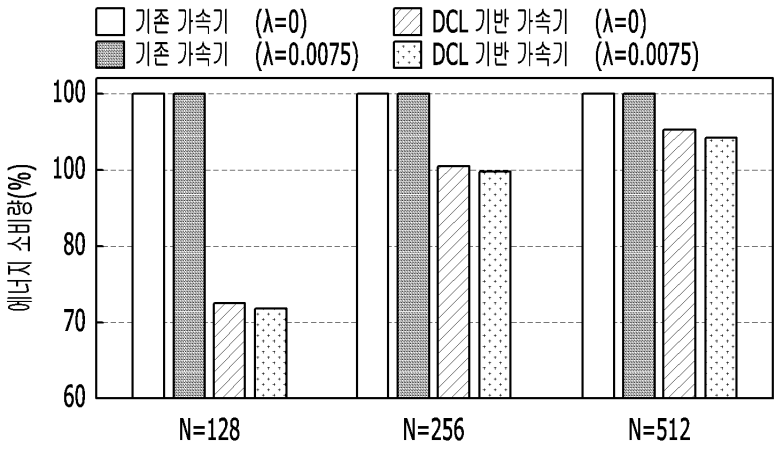
도면7


시스톨릭 어레이의 처리기는 입력 특징맵과 오프셋 필터의 컨볼루션 연산을 수행하여 생성된 오프셋(결과값)을 출력 버퍼에 저장
S210
샘플링 처리기는 출력 버퍼에 저장된 결과값을 기초로, 샘플링 할 위치를 결정하고 샘플링 위치의 픽셀값을 입력 버퍼에서 읽어옴
S220
샘플링 처리기는 샘플링 위치의 픽셀 값과 이중 선형 보간을 통해 새로운 입력 특징맵을 생성하여 출력 버퍼에 저장
S230
샘플링 과정 후, 시스톨릭 어레이의 처리기는 새로운 입력 특징맵과 변형 가능한 컨볼루션 필터를 컨볼루션하여 출력 특징맵 생성
S240

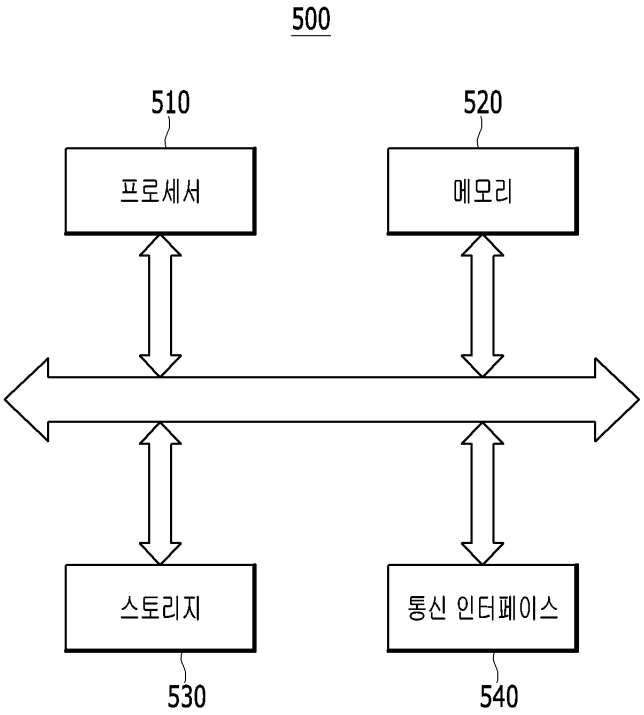
도면8


기존 가속기 (λ=0)
기존 가속기 (λ=0.0075)
DCL 기반 가속기 (λ=0)
DCL 기반 가속기 (λ=0.0075)
18.0x
15.0x
12.0x
9.0x
6.0x
3.0x
0.0x
N=128
N=256
N=512

도면9


기존 가속기 (λ=0)
기존 가속기 (λ=0.0075)
DCL 기반 가속기 (λ=0)
DCL 기반 가속기 (λ=0.0075)
에너지 소비량(%)
100
100
80
70
60
N=128
N=256
N=512

도면10


500
510
프로세서
520
메모리
스토리지
530
통신 인터페이스
540