



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0126334
(43) 공개일자 2024년08월20일

(51) 국제특허분류(Int. Cl.)
G06N 3/08 (2023.01) G06N 3/048 (2023.01)
(52) CPC특허분류
G06N 3/084 (2023.01)
G06N 3/048 (2023.01)
(21) 출원번호 10-2023-0018978
(22) 출원일자 2023년02월13일
심사청구일자 2023년02월13일

(71) 출원인
동서대학교 산학협력단
부산광역시 사상구 주례로 47(주례동,
동서대학교)
(72) 발명자
강대기
부산광역시 해운대구 좌동순환로 402 래미안해운
대 105동 301호
신지용
부산광역시 사상구 주례로 42-9
(74) 대리인
김석계

전체 청구항 수 : 총 3 항

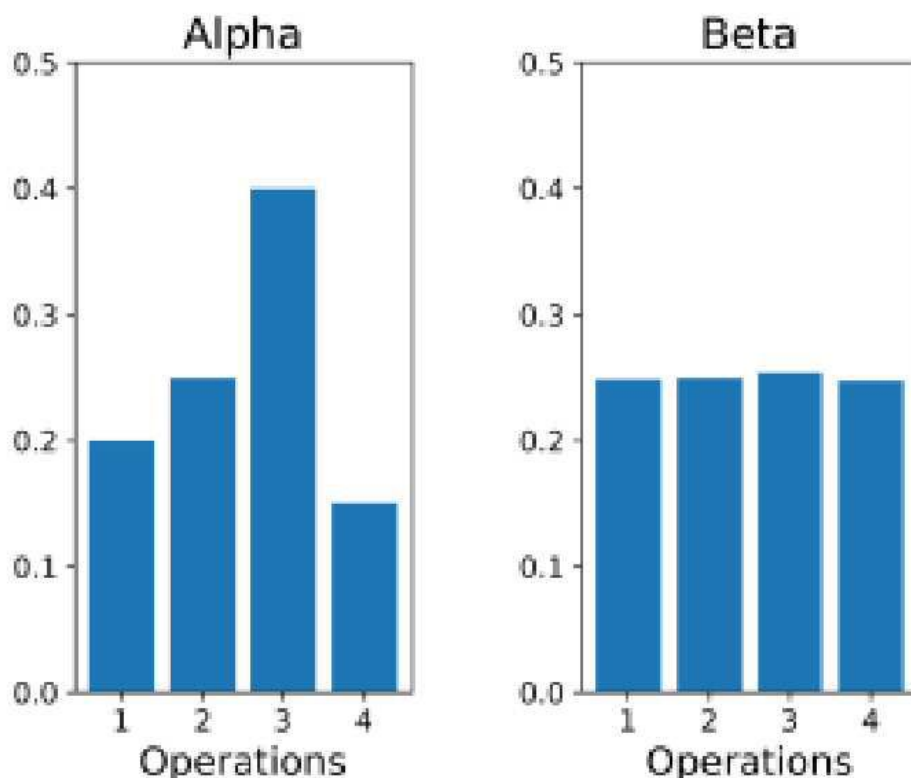
(54) 발명의 명칭 미분가능 아키텍처 검색에서의 온도 감쇠방법

(57) 요약

자동화된 기계학습(AutoML; Automated Machine Learning)은 인공지능 모델개발의 속도를 높이고 요구되는 자원을 최소화하기 위한 분야이다. 그중에서도 모델의 아키텍처를 생성하는 NAS(Neural Architecture Search; 신경망 아키텍처 검색) 알고리즘이 활발하게 연구되고 있다. NAS 알고리즘들의 기본적인 목적은 설계자의 개입을 최

(뒷면에 계속)

대표도 - 도2



소화하여 모델 생성 작업의 효율성을 높이면서 최적의 모델 아키텍처를 설계하는 것이다. NAS 알고리즘의 설계와 구현을 위해서 강화학습(Reinforcement Learning; RL), 진화 알고리즘(Evolutionary Algorithm) 등 여러 접근법이 있을 수 있는데,

본 발명에서는 그중에서도 경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하는 방안에 중점을 두었다. DARTS의 혼합 연산자(Mixed Operation)에 온도를 적용하여 아키텍처 검색 시 탐색(Exploration)과 착취(Exploitation)를 조절할 수 있게 만들었다. 또한, 기존의 DARTS 알고리즘은 혼합 연산자를 만드는 과정에서 아키텍처의 파라미터가 원-핫 인코딩된 아키텍처와 유사하게 학습될 것을 가정하게 되는데, 이때 생기는 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하였다. 본발명 실험은 아키텍처 검색 중에 온도를 10에서 0.1로 낮추어가며 진행하였고, 이에 따른 결과로, CIFAR-10 데이터 세트에서 97.37%의 정확도를 기록하여 기존 DARTS 알고리즘보다 0.13%p 증가한 성능을 보였다.

이 발명을 지원한 국가연구개발사업

과제고유번호	1345356165
과제번호	LINC3.0-2022-60
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	3단계 산학협력 선도대학 육성사업(LINC 3.0)
연구과제명	3단계 산학협력 선도대학 육성사업(LINC 3.0)(수요맞춤성장형)
기 여 율	1/2
과제수행기관명	동서대학교산학협력단
연구기간	2022.03.01 ~ 2028.02.28

이 발명을 지원한 국가연구개발사업

과제고유번호	1425163179
과제번호	S3126610
부처명	중소벤처기업부
과제관리(전문)기관명	중소기업기술정보진흥원
연구사업명	제조데이터공동활용플랫폼기술개발
연구과제명	빅데이터 및 기계학습 기반 공급망 최적화 기술 개발
기 여 율	1/2
과제수행기관명	동서대학교
연구기간	2022.04.23 ~ 2023.04.22

명세서

청구범위

청구항 1

경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하기 위해 DARTS의 혼합 연산자(Mixed Operation)에 온도를 적용하여 아키텍처 검색 시 탐색(Exploration)과 착취(Exploitation)를 조절할 수 있게 한 것을 특징으로 하는 미분가능 아키텍처 검색에서의 온도 감쇠방법

청구항 2

제1항에 있어서, 상기 온도의 값은 조절하는 것을 특징으로 하는 미분가능 아키텍처 검색에서의 온도 감쇠방법

청구항 3

제2항에 있어서, 상기 아키텍처 검색 중에 온도를 10에서 0.1로 낮추어가며 진행하는 것을 특징으로 하는 미분가능 아키텍처 검색에서의 온도 감쇠방법

발명의 설명

기술 분야

[0001] 본발명은 미분가능 아키텍처 검색에서의 온도 감쇠방법에 관한 것으로 보다 상세하게는 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하고 DARTS 알고리즘의 정확도와 성능이 증가되는 미분가능 아키텍처 검색에서의 온도 감쇠방법에 관한 것이다.

배경 기술

[0002] 일반적으로 자동화된 기계학습(AutoML; Automated Machine Learning)은 인공지능 모델개발의 속도를 높이고 요구되는 자원을 최소화하기 위한 분야이다. 그중에서도 모델의 아키텍처를 생성하는 NAS(Neural Architecture Search; 신경망 아키텍처 검색) 알고리즘이 활발하게 연구되고 있다. NAS 알고리즘들의 기본적인 목적은 설계자의 개입을 최소화하여 모델 생성 작업의 효율성을 높이면서 최적의 모델 아키텍처를 설계하는 것이다. NAS 알고리즘의 설계와 구현을 위해서 강화학습(Reinforcement Learning; RL), 진화 알고리즘(Evolutionary Algorithm) 등 여러 접근법이 있을 수 있다.

[0003] 그중에서도 경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하는 방법이 있다.

[0004] 본 발명 이전에도 DARTS의 몇 가지 문제점들을 제시하고 그것을 해결하기 위한 노력들이 있었다. 종래특허기술의 일례로서 공개번호 10-2022-0105081호에는 복수 개의 노드들과 상기 노드들을 서로 연결하는 간선들을 포함하는 셀을 포함하는 모델을 준비하는 단계;

[0005] 각각의 상기 간선에서 이용할 N1개의 후보연산들을 모두 이용하여 형성한 제1혼합연산을 각각의 상기 간선에 적용하여 상기 모델을 초기화하는 단계; 및

[0006] 상기 각각의 간선에 적용된 각각의 후보연산에 할당된 가중치를 학습시키는 제1학습단계;

[0007] 를 포함하는, 신경망 구조 탐색방법이 공개되어 있다.

[0008] 또한, 공개번호 10-2021-0135799호에는 뉴럴 네트워크 구조 탐색 장치 및 방법이 공개되어 있다.

[0009] ProxylessNAS는[1] One-Shot NAS와 DARTS가 했던 것처럼 역진과 알고리즘을 통해 한번의 훈련 과정만으로 검색 모델을 수렴시키는 상위 네트워크(SuperNet)를 사용하였다. 다만, 과도하게 매개변수화 시켰던 기존의 방식[2]에서 이진화된 간선을 만들어 요구되는 메모리 용량을 크게 줄였으며, CIFAR-10 데이터 세트에서 테스트 오차를 2.08%로 ENAS가 2.83%, DARTS가 2.83%, AmebaNet-B가 2.13%였던 것에 비해 향상된 결과를 보였다.

[0010] [1] H. Cai, L. Zhu, and S. Han, "ProxylessNAS: Direct neural architecture search on target task and hardware," In International Conference on Learning Representations, 2019.

- [0011] [2] Y. Xu, L. Xie, X. Zhang, X. Chen, G. Qi, Q. Tian, and H. Xiong, "PC-DARTS: Partial channel connections for memory-efficient differentiable architecture search," In Proceedings of The International Conference on Learning Representations (ICLR), , vol. abs/1907.05737, 12 Jul 2019.
- [0013] Fair-DARTS는 소프트맥스 함수가 불공정한 독점 경쟁을 부추긴다고 이야기하고 있다. 이는 검색 중에 하나 혹은 두 개의 연산자가 해당 간선에서 지배적으로 되는 것을 의미한다. 이것에 대한 해결책으로 소프트맥스 함수를 사용하는 대신 시그모이드 함수를 사용하고 가우시안 노이즈를 추가해 이러한 이점이 생기는 것을 방해한다. 또한, 해당 논문에서는 연속적으로 인코딩된 아키텍처의 솔루션이 원-핫 인코딩된 아키텍처와 유사하다는 DARTS의 기본 전제를 반박하며 이러한 불일치가 작을수록 결과 아키텍처가 일관된 성능을 낸다고 말하고 있다. 아키텍처의 가중치를 0 혹은 1의 극단 값으로 조정하는 L1 Regularization과 유사한 Zero-One 보조 손실을 소개하여 연속적으로 인코딩된 아키텍처와 그 아키텍처로부터 찾은 이산적 아키텍처 간의 불일치를 완화하였다[3]. 결과적으로 CIFAR-10 데이터 세트에서 75.6 % 정확도를 보이며 P-DARTS[1], PC-DARTS[2], SNAS[4], GDAS[5], FBNet-C[6] 알고리즘을 상회하는 성능을 보였다[3]. 이러한 방법은 불일치를 최소화 만들지만 아키텍처의 가중치가 불연속적으로 인코딩된다는 점에서 하나만을 극단적으로 선택하는 형태가 된다. 본 발명에서는 검색이 진행됨에 따라 경쟁을 점차 완화하여 자연스러운 경쟁이 되면서도 최종적으로는 결과 아키텍처의 모습과 유사한 형태의 환경을 조성하였다.
- [0015] [3] X. Chu, T. Zhou, B. Zhang, and J. Li, "Fair DARTS: Eliminating unfair advantages in differentiable architecture search," In Proceedings of The European Conference on Computer Vision (ECCV), 2019.
- [0017] [4] S. Xie, H. Zheng, C. Liu, and L. Lin, "SNAS: Stochastic neural architecture search," In Proceedings of International Conference on Learning Representations (ICLR), pp. 1761-1770, 2019.
- [0019] [5] X. Dong and Y. Yang, "Searching for a robust neural architecture in four gpu hours," In Proceedings of The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR), 2019.
- [0021] [6] B. Wu, X. Dai, P. Zhang, Y. Wang, F. Sun, Y. Wu, Y. Tian, P. Vajda, Y. Jia, and K. Keutzer, "FBNet: Hardware-aware efficient convnet design via differentiable neural architecture search," In Proceedings of The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR), 2019.
- [0023] DARTS with Ensemble Gumbel-Softmax 에서는 검벨 확률 분포를 사용하여 이산 변수들과 아키텍처 확률 분포로부터 샘플링을 시도한다. 이후 불연속인 argmax 부분을 softmax로 대체하여 Gumbel-Softmax가 된다. 해당 연구에서는 이것을 앙상블 하여 샘플링의 성능을 높인다. 여기에도 온도가 적용되어있지만, 이것은 하나의 하이퍼파라미터일 뿐 온도를 검색 중에 조절하여 아키텍처 파라미터 간 경쟁을 조율하지는 않고, 이진 코드로 샘플링 하는 것에 집중하였다. 이들의 연구 결과로 CIFAR-10 데이터 세트에서 DARTS가 4 GPU days가 걸리던 작업을 1.5 GPU days로 자원을 줄이면서도 DARTS와 유사한 성능을 얻어냈다[7].
- [0025] [7] J. Chang, X. Zhang, Y. Guo, G. Meng, S. Xiang, and C. Pan, "Differentiable architecture search with ensemble Gumbel-Softmax," arXiv preprint arXiv:1905.01786, 2019.
- [0027] β -DARTS는 AdaptNAS[8]에서 했던 도메인 적응을 이용하여 DARTS의 일반화 성능을 높이는 한편, DARTS 알고리즘을 올바르게 정규화하는 방법도 소개하고 있다. DARTS의 혼합 간선이 올바르게 정규화되기 위해서는 소프트맥스 함수 내에서 α 를 조절하는 매핑 함수가 아키텍처 파라미터 α 의 진폭에 영향을 받지 않으며 α 의 진폭을 반영 및 조절할 수 있어야 한다고 이야기하고 있다. 이들은 α 가 통과한 소프트맥스의 출력 값을 β 로 규정하고, 이

값의 분산은 작으면서도 값 자체는 평균에 가깝게 정규화하여 weight decay와 유사한 효과를 내는 새로운 정규화 방법 β -Decay 정규화를 제시하였다. 그에 따른 결과로 NAS-Bench-201의 CIFAR-100 데이터 세트에서 73.49%의 검증 정확도, 73.51%의 테스트 정확도로 SOTA(State-Of-The-Art)를 달성하였으며, CIFAR-10 데이터 세트에서는 DARTS의 1차 근사법을 사용한 모델과 유사한 속도로 2차 근사법을 사용한 모델보다 뛰어난 97.47%의 정확도를 보였다[9].

[0029] [8] Y. Li, Z. Yang, Y. Wang, and C. Xu, "Adapting neural architectures between domains," In Advances in Neural Information Processing Systems, vol. 33, pp. 789-798, 2020.

[0031] [9] P. Ye, B. Li, Y. Li, T. Chen, J. Fan, and W. Ouyang, " β -DARTS: Beta-decay regularization for differentiable architecture search," In Proceedings of The IEEE / CVF Computer Vision and Pattern Recognition Conference (CVPR), pp. 10864-10873, 2022.

[0032] 그러나 상기 종래기술들은 아키텍처 파라미터와 인코딩된 아키텍처 간의 불일치 문제가 있고 DARTS 알고리즘의 정확도와 성능이 좋지 않은 단점이 있었다.

발명의 내용

해결하려는 과제

[0033] 따라서 본 발명은 상기와 같은 문제점을 해결하고자 안출된 것으로, 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하고 DARTS 알고리즘의 정확도와 성능이 증가되는 미분가능 아키텍처 검색에서의 온도 감쇠방법을 제공하고자 하는 것이다.

과제의 해결 수단

[0034] 본발명은 미분가능 아키텍처 검색에서의 온도 감쇠방법에 관한 것으로, 경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하기 위해 DARTS의 혼합 연산자(Mixed Operation)에 온도를 적용하여 아키텍처 검색 시 탐색(Exploration)과 착취(Exploitation)를 조절할 수 있게 한 것을 특징으로 한다.

발명의 효과

[0035] 따라서 본발명은 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하고 DARTS 알고리즘의 정확도와 성능이 증가되는 현저한 효과가 있다.

도면의 간단한 설명

[0036] 도 1. 소프트맥스 함수를 통과하기 전의 아키텍처 가중치인 α 값과 소프트맥스 함수를 통과한 아키텍처 가중치 β 간의 비교 예시. 온도 값이 작은 경우의 그래프

도 2. 소프트맥스 함수를 통과하기 전의 아키텍처 가중치인 α 값과 소프트맥스 함수를 통과한 아키텍처 가중치 β 간의 비교 예시. 온도 값이 큰 경우의 그래프

발명을 실시하기 위한 구체적인 내용

[0037] 본발명은 미분가능 아키텍처 검색에서의 온도 감쇠방법에 관한 것으로, 경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하기 위해 DARTS의 혼합 연산자(Mixed Operation)에 온도를 적용하여 아키텍처 검색 시 탐색(Exploration)과 착취(Exploitation)를 조절할 수 있게 한 것을 특징으로 한다.

[0038] 또한, 상기 온도의 값을 조절하는 것을 특징으로 한다.

[0039] 또한, 아키텍처 검색 중에 온도를 10에서 0.1로 낮추어가며 진행하는 것을 특징으로 한다.

[0040] 본발명을 첨부도면에 의해 상세히 설명하면 다음과 같다.

[0041] 본 발명에서는 경사 하강법을 이용한 DARTS(Differentiable Architecture Search)를 개선하는 방안에 중점을 두었다. DARTS의 혼합 연산자(Mixed Operation)에 온도를 적용하여 아키텍처 검색 시 탐색(Exploration)과 착

취(Exploitation)를 조절할 수 있게 만들었다. 또한, 기존의 DARTS 알고리즘은 혼합 연산자를 만드는 과정에서 아키텍처의 파라미터가 원-핫 인코딩된 아키텍처와 유사하게 학습될 것을 가정하게 되는데, 이때 생기는 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하였다. 실험은 아키텍처 검색 중에 온도를 10에서 0.1로 낮추어가며 진행하였고, 이에 따른 결과로, CIFAR-10 데이터 세트에서 97.37%의 정확도를 기록하여 기존 DARTS 알고리즘보다 0.13%p 증가한 성능을 보였다.

[0042] 온도 감쇠 미분가능 아키텍처 검색 (TD-DARTS)

[0043] 혼합 연산자에서의 온도 담금질 및 온도가 가지는 영향에 대해 설명하면,

[0044] DARTS에서 사용되는 소프트맥스 함수는 두 가지가 있는데, 하나는 최종 분류에 사용되는 소프트맥스 함수이고, 다른 하나는 수식 1과 같은 혼합 연산자(Mixed Edge)를 만드는 데에 사용되는 소프트맥스 함수이다.

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$

[0046]

[0047] 수식 1. 혼합 연산자 나타내는 수식

[0049] 그 중에서도 혼합 간선 만드는데 사용되는 소프트맥스는 아키텍처의 선택 자체에 직접적인 영향을 끼칠 수 있는 함수이다. 혼합 연산자는 모든 연산자 풀을 하나의 간선으로 만든다. 수식 2는 수식 1의 혼합 연산자의 소프트맥스 함수에 온도 T를 적용한 수식이다.

$$\bar{o}^{(i,j)}(x) = \sum_{k \in \mathcal{O}} \frac{\exp(\alpha_k^{(i,j)}/T)}{\sum_{k' \in \mathcal{O}} \exp(\alpha_{k'}^{(i,j)}/T)} o(x)$$

[0051]

[0052] 수식 2. 온도 감쇠를 적용한 혼합 연산자

[0054] 모든 아키텍처 파라미터 α 는 온도에 의해 리스케일링(rescaling)된다. 수식 3는 혼합 연산자의 소프트맥스 함수 부분을 β 로 바꾼 것이다. 아키텍처 파라미터 α 값은 정수이다. 각 간선의 α 벡터가 소프트맥스 함수로 입력되면, 출력은 0보다 큰 수로 이루어져 있으며, 합은 1인 벡터가 된다. 본 논문에서는 이 부분의 변화를 중점적으로 다루므로 출력 벡터는 β 로 따로 구분한다.

$$\beta_k^{(i,j)} = \frac{\exp(\alpha_k^{(i,j)}/T)}{\sum_{k' \in \mathcal{O}} \exp(\alpha_{k'}^{(i,j)}/T)}$$

[0056]

[0057] 수식 3. 아키텍처 파라미터가 온도 감쇠가 적용된 소프트맥스 함수를 통과하기 전과 후를 나타낸 수식

[0059] 결과적으로 수식 4와 같이 혼합 연산자의 소프트맥스 함수 부분은 β 로 표현이 가능하다. 즉, β 는 해당 간선에서의 각각의 연산자의 기여도를 나타내며, 이것이 결과적으로 어떤 연산자가 최적의 연산자인지 선택하는 척도가 된다.

$$\bar{o}^{(i,j)}(x) = \sum_{k' \in \mathcal{O}} \beta_{k'}^{(i,j)} o(x)$$

[0061]

[0062]

수식 5. 혼합 연산자에 연산자의 기여도 β 가 각각의 연산자에 곱해진 수식

[0064]

수식 3의 온도를 조정하여 α 값이 확률 분포인 β 값으로 변형될 때, 결과 확률 분포의 차이를 좁히거나 늘릴 수 있다. 모든 입력 α 값을 T 라는 동일한 온도 값으로 나누어 소프트맥스에 입력하게 되면, T 값이 0에 가까울수록 그림 1과 같이 β 의 차이는 커지고, T 값이 클수록 β 의 차이는 도 2와 같이 작아지게 된다.

[0065]

T 값을 어떤 값으로 두느냐에 따라서 각 연산자의 기여도인 β 는 실제 β 값보다 크거나 작은 기여를 하게 된다. 즉, 연산자의 기여도를 연산자의 가중치에 비례하되, 다르게 설정하는 것이 가능한 것이다. 이렇게 하는 이유는 첫째로, DARTS는 연산자를 선택하기 위한 방법으로 역전파를 통한 그래디언트 기반(Gradient-based)의 방법을 사용한다. 각각의 레이어에서 가능한 모든 연산자를 묶어주기 위해 소프트맥스 함수를 사용하는데, 소프트맥스 함수를 통과한 가중치는 결과값에 해당 연산자를 얼마나 기여하게 할 것인가를 결정한다. 모든 가중치는 0부터 1 사이의 값으로 결정되며, 모든 연산자는 해당 값을 부여받는다. α 값의 업데이트는 결괏값으로부터 β , 소프트맥스를 통과하여 α 값으로 역전파 되므로 모든 연산자가 0부터 1 사이에서 경쟁하게 되는 현상이 일어난다. 이때, T 를 적용하게 되면 이 경쟁을 조정할 수 있게 되는 것이다. T 가 크면, 작은 α 값들과 큰 α 값들의 차이가 소프트맥스 함수를 통과하면서 작아져, 경쟁이 심해지고, T 가 작으면, β 값 간의 차이가 벌어져 높은 α 값들이 다른 α 값들에 비해 해당 간선에서의 지배력이 커진다. 최적의 연산자를 찾는 문제인 NAS의 시각으로 본다면, 결국에는 β 값이 가장 높은 연산자가 선택되므로, T 가 높으면 검색 초기에는 지배적이었던 연산자도 마지막에는 선택되지 않을 가능성이 커지며, T 가 낮으면 검색 초기에 지배적이었던 연산자가 마지막에도 선택될 가능성이 커진다. 다시 말하면, T 값이 높으면 기존의 정보보다 새로운 정보를 더 많이 수용하게 되고, T 값이 낮으면 기존의 정보에 더 많이 의존하게 된다. 이는 T 값이 높으면 검색을 많이 하게 되고 착취는 적어지며, T 값이 낮으면 검색을 적게 하고, 착취는 커지게 된다는 의미가 된다.

[0066]

T 가 가지는 의미는 한 가지 더 있다. DARTS 알고리즘은 검색이 종료된 후에 결국 이산적인 선택을 해야 한다. 이산적인 선택을 한다는 의미는 0과 1로 원-핫 인코딩된 각각의 연산자들 중에 하나를 선택한다는 의미이다. 즉, β 값이 0 또는 1로 이루어지게 바꾸는 과정인데, 실제 검색과정에서는 β 가 0과 1로 이루어져 있지 않으며, 모든 연산자가 어느 정도 출력값에 기여를 한다. 이에 따른 불일치가 존재하는데, 이것은 T 값을 낮춤으로써 완화할 수 있다. T 값이 낮을수록 β 값들은 0과 1에 가까운 형태가 된다. 이 형태로 검색을 진행하면, 실제 모델 학습과 유사한 결과로 업데이트하게 된다. 따라서 검색을 통해 선택된 결과 모델이 실제 학습에서도 기대한 성능을 낼 가능성이 커진다.

[0068]

온도 감쇠 전략으로서, 온도 T 는 높을수록 연산자 간의 경쟁이 심해지고, 낮을수록 연산자들의 격차는 벌어진다. 따라서 검색 초기에는 T 값이 높은 상태로 검색을 진행하여 지배적이지 않은 연산자에도 지배적으로 될 가능성을 열어준다. 반대로 검색 후기에는 T 값을 낮추어 경쟁을 마무리하고 가장 높은 α 값을 가진 연산자에 지배적으로 될 가능성을 부여하여 실제 결과 모델과 유사한 결과물로 검색을 진행할 수 있게 한다. 온도 감쇠 전략은 여러 다른 방법이 사용될 수 있는데, 본 발명에서 시행된 방법은 다음 표1과 같다.

Algorithm 2: TD-DARTS

Create a mixed operation $\bar{o}^{(i,j)}(x) = \sum_{k \in \mathcal{O}} \frac{\exp(\alpha_k^{(i,j)}/T)}{\sum_{k' \in \mathcal{O}} \exp(\alpha_{k'}^{(i,j)}/T)} o(x)$

Input: initial temperature T_{init} , final temperature T_{final} , total number of epoch E

$Interval = T_{init} - (T_{init} - T_{final})/E - 1$

for $t = 0$ **to** E **do**

$T^t = 10 - (t \times Interval)$

1. Update architecture α by descending $\nabla_{\alpha} \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)$

$\alpha_k^{t+1} \leftarrow \frac{\alpha_k^t - \eta \alpha \cdot \nabla_{\alpha_k} \mathcal{L}_{val}(w - \xi \nabla_w \mathcal{L}_{train}(w, \alpha), \alpha)}{T^t}$

2. Update weights w by descending $\nabla_w \mathcal{L}_{train}(w, \alpha)$

end

Derive the final architecture based on the learned α

[0070]

[0071]

표1. 알고리즘 2. TD 알고리즘. 각 에포크에 적용되는 온도를 계산 및 적용하는 방식

[0073]

알고리즘 2.는 본 발명에서 온도 감쇠가 적용된 방식을 나타내었다. 시작 온도를 $T_{initial}$, 최종 온도를 T_{final} , 전체 에포크는 E , 각 에포크는 t , 에포크마다 감쇠되는 온도의 간격은 $Interval$ 로, 해당 에포크에 적용되는 온도를 T 로 표기하였다. 하이퍼파라미터로 초기 온도와 최종 온도를 지정한다. 온도가 1일 때, 기존의 DARTS 알고리즘과 같아진다. 따라서, 초기 온도는 1보다 큰 수로 설정하고, 최종 온도는 1보다 작은 수로 설정한다. 온도가 커질수록 식4에서의 β 값의 차이가 작아지고, 0에 가까울수록 β 값의 차이가 벌어진다는 것을 상기하여 설정하는 것이 좋다. $Interval$ 은 시작 온도와 최종 온도를 고정할 수 있도록 설정하였다. 에포크마다 같은 온도 차이로 감쇠시키므로 선형적 감쇠가 된다.

[0074]

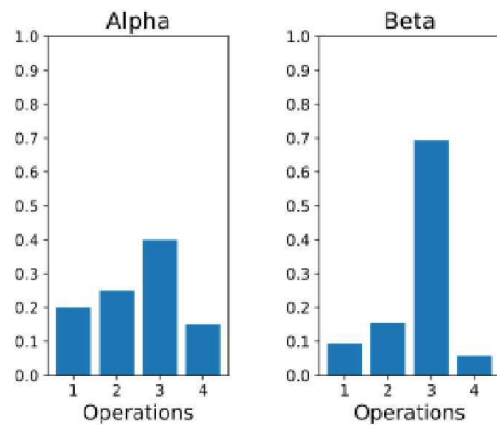
또한 선형적 감쇠가 아닌 다른 스케줄링 방법을 사용해볼 수 있다. 다른 가중치 감쇠(Weight Decay) 방법들과 유사하게 램다, 지수적 감쇠, 코사인 어닐링, 주기적 감쇠 등의 방법들을 사용해볼 수 있다.

[0075]

따라서 본발명은 아키텍처 파라미터와 원-핫 인코딩된 아키텍처 간의 불일치 문제를 온도 값을 조정함으로써 완화하고 DARTS 알고리즘의 정확도와 성능이 증가되는 현저한 효과가 있다.

도면

도면1



도면2

