



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2024-0148151
(43) 공개일자 2024년10월11일

(51) 국제특허분류(Int. Cl.)
G06N 5/01 (2023.01) G06N 3/04 (2023.01)
G06N 3/08 (2023.01)
(52) CPC특허분류
G06N 5/01 (2023.01)
G06N 3/04 (2023.01)
(21) 출원번호 10-2023-0043576
(22) 출원일자 2023년04월03일
심사청구일자 2023년04월03일

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
김선욱
경기도 남양주시 도농로 34(도농동, 부영그린타운) 214동 2003호
김석영
서울특별시 노원구 화랑로 556 (공릉동 신내효성아파트) 102동 503호
이재욱
서울특별시 강남구 삼성로 150 (대치동 한보미도맨션) 207동 504호
(74) 대리인
윤귀상

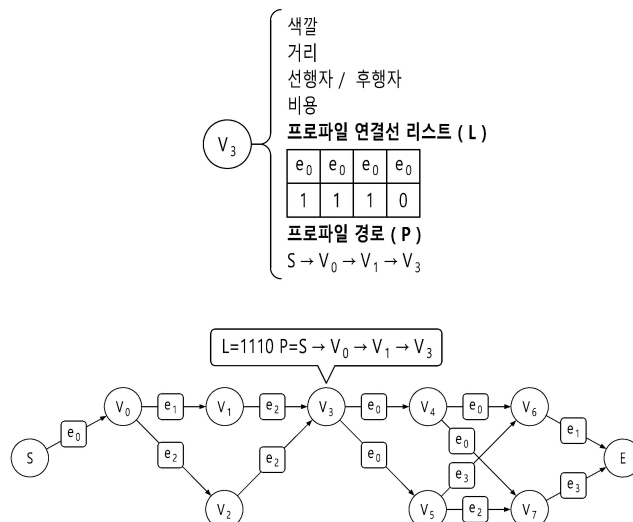
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 딥 러닝 모델의 그래프 탐색 방법 및 장치

(57) 요약

본 발명은 딥 러닝 모델의 그래프 탐색에 대한 것으로, 딥 러닝 모델의 그래프가 입력되면, 상기 그래프를 구성하는 정점들 각각의 속성 정보에 추가된 프로파일 관련 정보를 초기화하고, 상기 정점들에 대한 위상 정렬(topological sort)을 진행하고, 위상 정렬된 순서에 따라 해당 정점을 탐색하고, 상기 탐색이 완료되면, 상기 해당 정점의 선행자 정점의 프로파일 관련 정보와 상기 해당 정점의 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 관련 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하고, 상기 그래프의 시작점으로부터 상기 해당 정점까지의 경로에서 상기 그래프를 구성하는 서로 다른 연결선들이 모두 탐색될 경우, 상기 경로 정보를 출력한다. 또한 본 발명은 제안한 데이터 구조를 통해 그래프 탐색을 수행함으로써, 다항 시간 내에 탐색하고자 하는 연결선을 포함하는 최소 실행 횟수와 이때의 경로를 구한다.

대표도 - 도2



(52) CPC특허분류

G06N 3/082 (2023.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711193508
과제번호	2022-0-00050-002
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	PIM인공지능반도체핵심기술개발
연구과제명	데이터 플로우 구조 기반 PIM의 실행 및 프로그래밍 모델 개발
기 여 율	1/1
과제수행기관명	한국전자기술연구원
연구기간	2022.04.01 ~ 2025.12.31

명세서

청구범위

청구항 1

딥 러닝 모델의 그래프 탐색을 수행하는 방법에 있어서,

딥 러닝 모델의 그래프가 입력되면, 상기 그래프를 구성하는 정점들 각각의 속성 정보에 추가된 프로파일 관련 정보를 초기화하는 과정과,

상기 정점들에 대한 위상 정렬(topological sort)을 진행하고, 위상 정렬된 순서에 따라 해당 정점을 탐색하는 과정과,

상기 탐색이 완료되면, 상기 해당 정점의 선행자 정점의 프로파일 관련 정보와 상기 해당 정점의 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 관련 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 과정과,

상기 그래프의 시작점으로부터 상기 해당 정점까지의 경로에서 상기 그래프를 구성하는 서로 다른 연결선들이 모두 탐색될 경우, 상기 경로 정보를 출력하는 과정을 포함하는 그래프 탐색 방법.

청구항 2

제1항에 있어서,

상기 프로파일 관련 정보는 상기 서로 다른 연결선들의 탐색 여부를 나타내는 프로파일 연결선 리스트 정보와 상기 시작점으로부터 상기 해당 정점까지의 경로를 나타내는 프로파일 경로 정보를 포함함을 특징으로 하는 그래프 탐색 방법.

청구항 3

제2항에 있어서,

상기 업데이트 과정은, 상기 해당 정점에 대한 진입 연결선이 한 개일 경우, 상기 해당 정점의 선행자 정점의 프로파일 연결선 리스트 정보와 상기 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 연결선 리스트 정보 및 프로파일 경로 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 과정을 포함함을 특징으로 하는 그래프 탐색 방법.

청구항 4

제2항에 있어서,

상기 업데이트 과정은, 상기 해당 정점에 대한 진입 연결선이 복수 개일 경우, 진입 연결선들 각각에 대한 선행자 정점의 프로파일 연결선 리스트 정보와 해당 진입 연결선 정보를 기반으로 상기 진입 연결선 개수에 상응하는 후보 프로파일 연결선 리스트 정보들을 도출하고, 후보 프로파일 연결선 리스트 정보들의 포함관계를 판단하여 상위 집합에 해당하는 프로파일 연결선 리스트 정보 및 프로파일 경로 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 과정을 포함함을 특징으로 하는 그래프 탐색 방법.

청구항 5

제2항에 있어서,

상기 프로파일 연결선 리스트의 값이 상기 서로 다른 연결선들이 모두 탐색되었음을 나타내는 특정 값을 만족할 경우, 상기 그래프 탐색을 종료하는 과정을 더 포함하는 그래프 탐색 방법.

청구항 6

딥 러닝 모델의 그래프 탐색을 수행하는 장치에 있어서,

딥 러닝 모델의 그래프를 입력 받는 입력부와,

상기 그래프를 구성하는 정점들 각각의 속성 정보에 추가된 프로파일 관련 정보를 초기화하고, 상기 정점들에 대한 위상 정렬(topological sort)을 진행하고, 위상 정렬된 순서에 따라 해당 정점을 탐색하고, 상기 탐색이 완료되면, 상기 해당 정점의 선행자 정점의 프로파일 관련 정보와 상기 해당 정점의 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 관련 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 프로세서와,

상기 그래프의 시작점으로부터 상기 해당 정점까지의 경로에서 상기 그래프를 구성하는 서로 다른 연결선들이 모두 탐색될 경우, 상기 경로 정보를 출력하는 출력부를 포함하는 그래프 탐색 장치.

청구항 7

제6항에 있어서,

상기 프로파일 관련 정보는 상기 서로 다른 연결선들의 탐색 여부를 나타내는 프로파일 연결선 리스트 정보와 상기 시작점으로부터 상기 해당 정점까지의 경로를 나타내는 프로파일 경로 정보를 포함함을 특징으로 하는 그래프 탐색 장치.

청구항 8

제7항에 있어서,

상기 프로세서는, 상기 해당 정점에 대한 진입 연결선이 한 개일 경우, 상기 해당 정점의 선행자 정점의 프로파일 연결선 리스트 정보와 상기 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 연결선 리스트 정보 및 프로파일 경로 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트함을 특징으로 하는 그래프 탐색 장치.

청구항 9

제7항에 있어서,

상기 프로세서는, 상기 해당 정점에 대한 진입 연결선이 복수 개일 경우, 진입 연결선들 각각에 대한 선행자 정점의 프로파일 연결선 리스트 정보와 해당 진입 연결선 정보를 기반으로 상기 진입 연결선 개수에 상응하는 후보 프로파일 연결선 리스트 정보들을 도출하고, 후보 프로파일 연결선 리스트 정보들의 포함관계를 판단하여 상위 집합에 해당하는 프로파일 연결선 리스트 정보 및 프로파일 경로 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트함을 특징으로 하는 그래프 탐색 장치.

청구항 10

제7항에 있어서,

상기 프로세서는, 상기 프로파일 연결선 리스트의 값이 상기 서로 다른 연결선들이 모두 탐색되었음을 나타내는 특정 값을 만족할 경우, 상기 그래프 탐색을 종료함을 특징으로 하는 그래프 탐색 장치.

발명의 설명

기술 분야

[0001] 본 발명은 딥 러닝(Deep learning) 모델의 그래프 탐색 방법 및 장치에 관한 것으로, 특히 그래프의 연결선(edge) 정보를 기반으로 딥 러닝 모델의 프로파일링 오버헤드를 최소화하는 그래프 탐색 방법 및 장치에 관한 것이다.

배경 기술

[0003] 인공 신경망(ANN: Artificial Neural Networks) 기술은 컴퓨터를 사용하여 인간의 뇌를 모방하기 위하여 제안되었으며, 인간의 신경 세포를 모델링한 퍼셉트론(Perceptron)으로 구성된다. 퍼셉트론은 다수의 입력 데이터 층(layer)을 연산하여 출력 데이터 층으로 내보낸다. 이때 머신 러닝(Machine learning)이란 정방향/역방향 전파 과정을 통해 퍼셉트론의 가중치(weight)값이 갱신되며 입력 데이터에 대해서 학습함을 의미한다. 그러나 최근에는 입력 데이터가 이미지, 언어 등으로 복잡해짐에 따라 단층의 퍼셉트론 만으로는 학습에 한계가 있고, 이를 극복하기 위해 퍼셉트론을 다층으로 배치하여 기계 학습의 정확도를 올리는 알고리즘인 딥 러닝 기술이 각광받고 있다.

[0004] 이러한 딥 러닝 모델은 하나의 그래프로 표현될 수 있으며, 상기 그래프 $G(V,E)$ 는 딥 러닝 연산자를 나타내는 정점(vertex) 집합 V 와 딥 러닝 연산자들 간의 관계성(dependency)을 나타내는 연결선(edge) 집합 E 로 구성된다.

[0005] 한편 그래프를 구성하는 정점과 연결선의 수가 매우 커질 경우 딥 러닝 모델의 성능, 가용성 등의 향상을 위해 그래프 파티셔닝(partitioning)이 행해지는데, 이러한 그래프 파티셔닝을 위해서는 딥 러닝 모델의 프로파일링이 선행되어야 한다. 즉 딥 러닝 모델 그래프의 출발점부터 종료점까지 하나의 경로(path)를 통해 실행하여 정점과 연결선에 정의된 비용(cost), 즉 실행시간을 측정해야 하며, 측정 결과를 활용하여 그래프 파티셔닝을 수행할 수 있다. 또한 그래프 파티셔닝을 통해 CPU(Central Processing Unit) 또는 GPU(Graphic Processing Unit)와 같은 여러 연산 장치(computing device)들이 존재하는 이기종 환경(heterogeneous platform)에서 어떤 정점들을 어떤 연산 장치에 할당할지 결정하여 딥 러닝 모델의 최적 실행을 구할 수 있다.

[0006] 그러나 딥 러닝 모델 그래프의 복잡도(complexity)가 커짐에 따라 상기 그래프를 구성하는 정점과 연결선의 수도 증가하며, 그래프 $G(V,E)$ 에서 출발점에서 종료점까지 모든 경로를 나열하는 시간 복잡도는 $O(V!)$ 이고, 탐색하고자 하는 연결선의 포함여부를 확인하는 시간 복잡도는 $O(E)$ 이므로, 모든 정점의 비용(V_{cost})과 모든 연결선의 비용(E_{cost})을 측정하는 것은 프로파일링 오버헤드를 증가시켜 다항 시간(polynomial time) 내에 그래프 탐색을 완료하지 못하는 문제점이 있다.

[0007] 이에 따라 딥 러닝 모델 그래프의 프로파일링 오버헤드를 감소시킬 수 있는 기술이 필요하다.

선행기술문헌

특허문헌

[0009] (특허문헌 0001) 한국 공개특허공보 제10-2023-0001132호

발명의 내용

해결하려는 과제

[0010] 본 발명은 상기와 같은 문제를 해결하기 위해 안출된 것으로, 본 발명의 목적은 그래프 연결선 정보를 기반으로 딥 러닝 모델의 그래프를 탐색하고, 프로파일링 오버헤드를 최소화하는 방법 및 장치에 관한 것이다.

과제의 해결 수단

[0012] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 딥 러닝 모델의 그래프 탐색을 수행하는 방법은, 딥 러

닝 모델의 그래프가 입력되면, 상기 그래프를 구성하는 정점들 각각의 속성 정보에 추가된 프로파일 관련 정보를 초기화하는 과정과, 상기 정점들에 대한 위상 정렬(topological sort)을 진행하고, 위상 정렬된 순서에 따라 해당 정점을 탐색하는 과정과, 상기 탐색이 완료되면, 상기 해당 정점의 선행자 정점의 프로파일 관련 정보와 상기 해당 정점의 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 관련 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 과정과, 상기 그래프의 시작점으로부터 상기 해당 정점까지의 경로에서 상기 그래프를 구성하는 서로 다른 연결선들이 모두 탐색될 경우, 상기 경로 정보를 출력하는 과정을 포함한다.

[0013] 상기 목적을 달성하기 위한 본 발명의 일 실시예에 따른 딥 러닝 모델의 그래프 탐색을 수행하는 장치는, 딥 러닝 모델의 그래프를 입력 받는 입력부와, 상기 그래프를 구성하는 정점들 각각의 속성 정보에 추가된 프로파일 관련 정보를 초기화하고, 상기 정점들에 대한 위상 정렬(topological sort)을 진행하고, 위상 정렬된 순서에 따라 해당 정점을 탐색하고, 상기 탐색이 완료되면, 상기 해당 정점의 선행자 정점의 프로파일 관련 정보와 상기 해당 정점의 진입 연결선 정보를 기반으로 상기 해당 정점의 프로파일 관련 정보를 도출하여 상기 해당 정점의 속성 정보를 업데이트하는 프로세서와, 상기 그래프의 시작점으로부터 상기 해당 정점까지의 경로에서 상기 그래프를 구성하는 서로 다른 연결선들이 모두 탐색될 경우, 상기 경로 정보를 출력하는 출력부를 포함한다.

발명의 효과

[0015] 상술한 본 발명의 일 측면에 따르면, 본 발명은 딥 러닝 모델의 그래프 탐색 시 그래프의 연결선 정보를 기반으로 그래프를 탐색하는 방법 및 장치를 제공함으로써, 그래프의 출발점으로부터 종료점까지 탐색하고자 하는 연결선을 포함하는 최소 실행 횟수와 이때의 경로를 다항시간 내에 구할 수 있고, 이를 통해 프로파일링 오버헤드를 최소화할 수 있다.

[0016] 또한 모든 경로와 이때에 따른 비용을 단순화하여 일부 조합에 대해서만 프로파일링을 진행하는 종래의 그래프 탐색 방법 대비, 본 발명에서 제안하는 그래프 탐색 방법은 전체 그래프에 대한 탐색을 진행하므로 그래프 파서닝을 위해 정확한 프로파일링 결과를 제공할 수 있는 효과가 있다.

도면의 간단한 설명

[0018] 도 1a는 딥 러닝 모델의 그래프를 구성하는 정점과 연결선의 속성 정보를 나타낸 도면,
 도 1b는 딥 러닝 모델의 그래프 예시를 나타낸 도면,
 도 2는 본 발명의 실시예에 따른 딥 러닝 모델 그래프를 구성하는 정점의 속성 정보와, 상기 속성 정보를 이용하여 그래프 탐색을 수행하는 예를 나타낸 도면,
 도 3a 내지 3g는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 시간 흐름에 따라 탐색하는 예를 나타낸 도면,
 도 4는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프 탐색 알고리즘의 수도(pseudo) 코드의 예를 나타낸 도면,
 도 5는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 탐색하는 동작을 도시한 순서도,
 그리고, 도 6은 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 탐색하는 장치를 도시한 블록도.

발명을 실시하기 위한 구체적인 내용

[0019] 후술하는 본 발명에 대한 상세한 설명은, 본 발명이 실시될 수 있는 특정 실시예를 예시로서 도시하는 첨부 도면을 참조한다. 이들 실시예는 당업자가 본 발명을 실시할 수 있기에 충분하도록 상세히 설명된다. 본 발명의 다양한 실시예는 서로 다르지만 상호 배타적일 필요는 없음이 이해되어야 한다. 예를 들어, 여기에 기재되어 있는 특정 형상, 구조 및 특성은 일 실시예와 관련하여 본 발명의 정신 및 범위를 벗어나지 않으면서 다른 실시예로 구현될 수 있다. 또한, 각각의 개시된 실시예 내의 개별 구성요소의 위치 또는 배치는 본 발명의 정신 및 범위를 벗어나지 않으면서 변경될 수 있음이 이해되어야 한다. 따라서, 후술하는 상세한 설명은 한정적인 의미로서 취하려는 것이 아니며, 본 발명의 범위는, 적절하게 설명된다면, 그 청구항들이 주장하는 것과 균등한 모든 범위와 더불어 첨부된 청구항에 의해서만 한정된다. 도면에서 유사한 참조부호는 여러 측면에 걸쳐서 동일하게

나 유사한 기능을 지칭한다.

- [0020] 본 발명에 따른 구성요소들은 물리적인 구분이 아니라 기능적인 구분에 의해서 정의되는 구성요소들으로써 각각이 수행하는 기능들에 의해서 정의될 수 있다. 각각의 구성요소들은 하드웨어 또는 각각의 기능을 수행하는 프로그램 코드 및 프로세싱 유닛으로 구현될 수 있을 것이며, 두 개 이상의 구성요소의 기능이 하나의 구성요소에 포함되어 구현될 수도 있을 것이다. 따라서 이하의 실시예에서 구성요소에 부여되는 명칭은 각각의 구성요소를 물리적으로 구분하기 위한 것이 아니라 각각의 구성요소가 수행되는 대표적인 기능을 암시하기 위해서 부여된 것이며, 구성요소의 명칭에 의해서 본 발명의 기술적 사상이 한정되지 않는 것임에 유의하여야 한다.
- [0021] 이하에서는 도면들을 참조하여 본 발명의 바람직한 실시예들을 보다 상세하게 설명하기로 한다.
- [0022] 도 1a는 딥 러닝 모델의 그래프를 구성하는 정점과 연결선의 속성 정보를 나타낸 도면이고, 도 1b는 딥 러닝 모델의 그래프 예시를 나타낸 도면이다.
- [0023] 도 1a를 참조하면, 딥 러닝 모델의 그래프 $G(V, E)$ 는 딥 러닝 연산자들 나타내는 정점 집합 V 와 딥 러닝 연산자들 간의 관계성을 나타내는 연결선 집합 E 로 구성되고, 이때 정점 V_i 의 속성 정보는 정점의 인큐(enqueue)/디큐(dequeue)를 나타내는 색깔(color) 정보, 출발점(S)으로부터 V_i 까지의 거리(distance) 정보, 방향성 그래프에서의 선행자(predecessor)/후행자(successor) 정보 및 비용(cost) 정보 등을 포함한다. V_i 의 속성 정보 중 상기 비용 정보는 해당 정점이 딥 러닝 연산자를 실행할 때의 실행시간으로 정의될 수 있다.
- [0024] 연결선 E_i 의 속성은 색깔 정보, 소스(source) 정점/목적지(destination) 정점 정보 및 비용 정보 등을 포함한다. E_i 의 속성 정보 중 상기 비용 정보는 소스 정점으로부터 목적지 정점으로의 데이터 전송에 필요한 시간으로 정의될 수 있다.
- [0025] 도 1b를 참조하면, 도시된 그래프는 8개의 정점 V_0 내지 V_7 과 13개의 연결선 E_0 내지 E_{12} 로 구성된 그래프의 예를 나타낸 것이며, 출발점(S)으로부터 종료점(E)까지 복수개의 경로(path)를 포함할 수 있다. 상기 경로는 출발점(S)으로부터 종료점(E)까지의 정점의 순차적인 배열(sequence)로 정의되며, 예를 들어 경로 $(S \rightarrow V_0 \rightarrow V_2 \rightarrow V_3 \rightarrow V_5 \rightarrow V_7 \rightarrow E)$ 는 연결선 $(e_0, e_2, e_4, e_6, e_{10}, e_{12})$ 을 포함한다.
- [0026] 도시된 그래프의 경우 탐색하고자 하는 연결선이 탐색되었는지 여부를 판단하기 위해서는 8개의 모든 경로를 탐색해야 하며, 따라서 탐색하고자 하는 연결선의 프로파일링 정보를 얻기 위해서는 8번의 실행이 필요하다.
- [0027] 후술할 본 발명의 실시예에서는 정점의 속성에 별도의 데이터 구조(structure)를 추가하고 이를 활용하여 그래프 탐색을 수행하는 방법에 대해 설명하도록 한다.
- [0028] 도 2는 본 발명의 실시예에 따른 딥 러닝 모델 그래프를 구성하는 정점의 속성 정보와, 상기 속성 정보를 이용하여 그래프 탐색을 수행하는 예를 나타낸 도면이다.
- [0029] 도 2를 참조하면, 도시된 그래프는 8개의 정점과 13개의 연결선으로 구성된 그래프를 가정하며, 프로파일링의 목적은 그래프를 구성하는 정점과 연결선의 비용을 측정하는 것이므로 동일한 비용을 갖는 연결선들은 동일한 연결선으로 간주한다. 여기서 동일한 비용이라 함은 소스 정점으로부터 목적지 정점으로의 데이터 전송에 필요한 시간이 동일함을 의미한다. 따라서 도시된 그래프는 도 1b의 그래프와 동일한 그래프를 가정하나, e_0 와 동일한 비용을 갖는 연결선 e_5, e_6, e_7, e_8 은 e_0 로 대체하고, e_1 과 동일한 비용을 갖는 연결선 e_{11} 은 e_1 으로 대체하고, e_2 와 동일한 비용을 갖는 연결선 e_4, e_{10} 은 e_2 로 대체하고, e_3 와 동일한 비용을 갖는 연결선 e_9, e_{12} 은 e_3 로 대체하여 나타내었다.
- [0030] 본 발명에서는 그래프에서 탐색하고자 하는 연결선들의 탐색 여부를 리스트 형태로 나타낸 프로파일 연결선 리스트(Profiled Edge List) L 과, 프로파일 연결선 리스트에 대응되는 출발점(S)으로부터 정점 V_i 까지의 경로를 나타낸 프로파일 경로(Profiled Path) P 를 새롭게 정의한다. 상기 프로파일 연결선 리스트(L)는 탐색하고자 하는 연결선 개수에 상응하는 인덱스들을 가지며 인덱스 값은 0 또는 1로 설정될 수 있다. 인덱스 값이 1로 설정된 경우는 출발점(S)으로부터 정점 V_i 까지의 경로에서 해당 인덱스에 상응하는 연결선이 탐색되었음을 의미하고, 인덱스 값이 0으로 설정된 경우는 상기 경로에서 해당 인덱스에 상응하는 연결선이 탐색되지 않았음을 의미한다. 또한 상기 프로파일 연결선 리스트(L) 정보와 상기 프로파일 경로(P) 정보는 그래프 정점의 속성 정

보에 별도의 데이터 구조로써 추가된다.

- [0031] 예를 들어 정점 V_3 의 속성 정보에 대해 살펴보면 다음과 같다. 상기 정점 V_3 의 속성 정보는 색깔 정보, 거리 정보, 선행자/계승자 정보, 비용 정보, 프로파일 연결선 리스트(L) 정보 및 프로파일 경로(P) 정보를 포함한다. 도시된 그래프에서 탐색하고자 하는 서로 다른 연결선은 총 4개가 존재하므로 프로파일 연결선 리스트의 길이는 4이고, 프로파일 연결선 리스트(L)의 인덱스는 e_0, e_1, e_2, e_3 을 포함한다. 정점 V_3 의 프로파일 연결선 리스트 L 값은 [1110] 이므로, 출발점(S)으로부터 정점 V_3 까지의 경로에서 연결선 e_0, e_1, e_2 가 탐색되었음을 알 수 있다.
- [0032] 특정 시점에서의 정점 V_3 에 대해 살펴보면, 출발점(S)으로부터 정점 V_3 까지 $S \rightarrow V_0 \rightarrow V_1 \rightarrow V_3$ 의 제1 경로와 $S \rightarrow V_0 \rightarrow V_2 \rightarrow V_3$ 의 제2 경로가 존재하고, 이때 상기 제1 경로의 프로파일 연결선 리스트 $L=1110$ 이고, 상기 제2 경로의 프로파일 연결선 리스트 $L=1010$ 이다. 상기 제1 경로와 제2 경로의 프로파일 연결선 리스트 간의 포함관계를 비교해보면, 제1 경로의 프로파일 연결선 리스트를 L1이라 하고, 제2 경로의 프로파일 연결선 리스트를 L2라 할 때, L1 값은 [1110]이고 L2 값은 [1010]이므로 두 값 중 상위 집합(superset)에 해당하는 L1 값 [1110]과 상기 L1에 대응되는 경로 $S \rightarrow V_0 \rightarrow V_1 \rightarrow V_3$ 를 정점 V_3 의 속성 정보에 저장한다.
- [0033] 도 2에서는 특정 시점에서의 그래프 탐색 동작을 예를 들어 설명하였다. 이하에서는 도 3을 통해 도 2의 그래프와 동일한 그래프에 대하여 시간 흐름에 따라 그래프를 탐색하는 예에 대해 설명하도록 한다.
- [0034] 도 3a 내지 3g는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 시간 흐름에 따라 탐색하는 예를 나타낸 도면이다.
- [0035] 도 3을 설명하기에 앞서 본 발명의 실시예에 따른 그래프 탐색 방법을 간략히 설명하면 다음과 같다. 그래프를 구성하는 모든 정점들 V_0 내지 V_7 에 대하여 속성 정보들 중 프로파일 연결선 리스트(L) 및 프로파일 경로(P) 정보를 초기화하고, 상기 정점들에 대해 위상 정렬(topological sort)을 진행한다. 여기서는 $V_0, V_1, V_2, V_3, V_4, V_5, V_6, V_7$ 의 순서로 위상 정렬 되었다고 가정한다.
- [0036] 선입선출(FIFO: First Input First Output) 큐(queue)를 이용하여 상기 위상 정렬된 순서대로 그래프 탐색을 수행한다. 이때 초기의 모든 정점들은 흰색으로 설정하고, 정점이 탐색되면 인큐(enqueue)한 뒤 회색으로 설정하고, 탐색이 완료되면 디큐(dequeue)한 뒤 검은색으로 설정한다. 그래프 탐색 중 디큐한 정점(V_i)의 진입(incoming) 연결선 정보와 선행자 정점 정보를 이용하여 정점 V_i 에 대한 프로파일 연결선 리스트 $L(V_i)$ 과 이에 대응하는 프로파일 경로 $P(V_i)$ 를 구하고 이를 통해 정점 V_i 의 속성 정보를 업데이트한다. $L(V_i)$ 의 모든 값이 1일 경우에는 그래프 탐색을 종료한다.
- [0037] 도 3a에 도시된 T0 시점의 그래프 탐색 과정을 살펴보면, 정점 V_0 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_0 의 진입 연결선 e_0 가 탐색되었으므로, $L(V_0)$ 의 인덱스 e_0, e_1, e_2, e_3 에 대한 값은 [1000]으로 설정되고, $L(V_0)$ 값 [1000]과 이에 대응하는 출발점(S)으로부터 정점 V_0 까지의 $P(V_0)$ 정보 SV_0 를 V_0 에 저장하여 속성 정보를 업데이트한다.
- [0038] 도 3b에 도시된 T1 시점의 그래프 탐색 과정을 살펴보면, T0 시점에서 위상 정렬된 순서에 따라 인큐된 정점 V_1, V_2 중 V_1 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_1 의 진입 연결선 e_1 이 탐색되었으므로, 선행자 정점인 V_0 의 $L(V_0)$ 값 [1000]을 기반으로 탐색된 연결선 e_1 에 대한 인덱스 값만 '1'로 변경하여 $L(V_1)$ 값 [1100]을 구하고, $L(V_1)$ 값 [1100]과 이에 대응하는 출발점(S)으로부터 정점 V_1 까지의 $P(V_1)$ 정보 SV_0V_1 을 정점 V_1 에 저장하여 속성 정보를 업데이트한다. 선행자 정점인 V_0 의 $L(V_0)$ 값 [1000]은 탐색된 연결선 e_1 에 대한 인덱스 값이 '0'이므로 이를 '1'로 변경하였으나, 만약 연결선 e_1 에 대한 인덱스 값이 '1'일 경우에는 그 값을 유지하도록 한다. 이러한 경우에 대해서는 T4 시점의 그래프 탐색 과정에서 다시 설명하도록 한다.
- [0039] 도 3c에 도시된 T2 시점의 그래프 탐색 과정을 살펴보면, V_2 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_2 의 진입 연결선 e_2 가 탐색되었으므로, 선행자 정점인 V_0 의 $L(V_0)$ 값 [1000]을 기반으로 탐색된 연결선 e_2 에 대한 인덱스 값만 '1'로 변경하여 $L(V_2)$ 값 [1010]을 구하고, $L(V_2)$ 값 [1010]과 이에 대응하는 출발점(S)으로부터 정점 V_2 까지의 $P(V_2)$ 정보 SV_0V_2 를 정점 V_2 에 저장하여 속성 정보를 업데이트한다.

- [0040] 도 3d에 도시된 T3 시점의 그래프 탐색 과정을 살펴보면, T2 시점에서 인큐된 V_3 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_3 는 두개의 진입 연결선이 존재하므로 $L(V_3)$ 값 도출 시 고려되는 선행자 정점 역시 두개가 존재한다. 따라서 선행자 정점 V_1 의 $L(V_1)$ 값 [1100]과 탐색된 연결선 e_2 에 대한 정보를 기반으로 도출한 제1 후보 값 [1110]과, 선행자 정점 V_2 의 $L(V_2)$ 값 [1010]과 탐색된 연결선 e_2 에 대한 정보를 기반으로 도출한 제2 후보 값 [1010]을 각각 구한다음 상기 제1 후보 값과 제2 후보 값의 포함 관계를 판단한다. 판단 결과 상위 집합(subset)에 해당하는 제1 후보 값 [1110]을 $L(V_3)$ 값으로 도출하고, 이에 대응하는 출발점(S)으로부터 정점 V_3 까지의 $P(V_3)$ 정보 $SV_0V_1V_3$ 를 도출한다. 또한 상기 $L(V_3)$ 값 [1110]과 $P(V_3)$ 정보 $SV_0V_1V_3$ 를 정점 V_3 에 저장하여 속성 정보를 업데이트한다.
- [0041] 도 3e에 도시된 T4 시점의 그래프 탐색 과정을 살펴보면, T3 시점에서 인큐된 V_4, V_5 중 V_4 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_4 의 진입 연결선 e_0 가 탐색되었으므로, 선행자 정점인 V_3 의 $L(V_3)$ 값 [1110]을 기반으로 탐색된 연결선 e_0 에 대한 인덱스 값을 확인하고, 상기 탐색된 연결선 e_0 에 대한 인덱스 값은 이미 '1'로 설정되어 있으므로 $L(V_3)$ 값 [1110]을 그대로 상속받아 정점 V_4 의 속성 정보를 업데이트한다. 또한 $L(V_4)$ 값 [1110]에 대응하는 출발점(S)으로부터 정점 V_4 까지의 $P(V_4)$ 정보 $SV_0V_1V_3V_4$ 로 정점 V_4 의 속성 정보를 업데이트한다.
- [0042] 도 3f에 도시된 T5 시점의 그래프 탐색 과정을 살펴보면, V_5 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_5 의 진입 연결선 e_0 가 탐색되었으므로, 선행자 정점인 V_3 의 $L(V_3)$ 값 [1110]을 기반으로 탐색된 연결선 e_0 에 대한 인덱스 값을 확인하고, 상기 탐색된 연결선 e_0 에 대한 인덱스 값은 이미 '1'로 설정되어 있으므로 선행자 정점인 V_3 의 $L(V_3)$ 값 [1110]을 그대로 상속받아 정점 V_5 의 속성 정보를 업데이트한다. 또한 $L(V_5)$ 값 [1110]에 대응하는 출발점(S)으로부터 정점 V_5 까지의 $P(V_5)$ 정보 $SV_0V_1V_3V_5$ 로 정점 V_5 의 속성 정보를 업데이트한다.
- [0043] 도 3g에 도시된 T6 시점의 그래프 탐색 과정을 살펴보면, T5 시점에서 인큐된 V_6, V_7 중 V_6 의 탐색이 완료되면 디큐한 뒤 검은색으로 설정한다. 정점 V_6 는 두개의 진입 연결선이 존재하므로, 선행자 정점 V_4 의 $L(V_4)$ 값 [1110]을 기반으로 탐색된 연결선 e_0 에 대한 인덱스 값을 확인하여 계산한 값 [1110]과, 선행자 정점 V_5 의 $L(V_5)$ 값 [1110]을 기반으로 탐색된 연결선 e_3 에 대한 인덱스 값을 확인하여 계산한 값 [1111]을 각각 구한다음 상기 [1110]과 상기 [1111]의 포함 관계를 판단한다. 판단 결과 상위 집합에 해당하는 값 [1111]로 정점 V_6 의 속성 정보를 업데이트하고, 이에 대응하는 출발점(S)으로부터 정점 V_6 까지의 $P(V_6)$ 정보 $SV_0V_1V_3V_5V_6$ 로 정점 V_6 의 속성 정보를 업데이트한다. 이때 $L(V_6)$ 값은 [1111]로 그래프를 구성하는 모든 연결선이 탐색되었으므로 그래프 탐색은 종료되고, 모든 연결선이 탐색된 경로, 즉 출발점(S)으로부터 정점 V_6 까지의 $P(V_6)$ 정보 $SV_0V_1V_3V_5$ 를 출력한다.
- [0044] 이와 같이 본 발명에서 제안하는 그래프 탐색 방법은 정점의 속성 정보에 탐색하고자 하는 연결선들의 탐색 여부를 나타내는 프로파일 연결선 리스트 정보와 이에 대응하는 프로파일 경로 정보를 추가함으로써, 그래프 탐색 시 상기 추가된 프로파일 정보들을 이용하여 그래프의 출발점으로부터 종료점까지 탐색하고자 하는 연결선을 포함하는 최소 실행 횟수와 이때의 경로를 다항 시간 내에 구할 수 있다.
- [0045] 또한 V 개의 정점과 E 개의 연결선 그래프 만들 탐색하고 하나의 정점에서 서로 다른 E 개의 프로파일 연결선 리스트의 포함 관계 여부를 판단하므로 전체 알고리즘 시간 복잡도는 $O(V+E*E)=O(V+E^2)$ 와 같이 나타낼 수 있으며, 시간 복잡도의 감소로 인해 프로파일링 오버헤드 또한 감소된다.
- [0046] 도 4는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프 탐색 알고리즘의 수도(pseudo) 코드의 예를 나타낸 도면이다.
- [0047] 도 4를 참조하면, 도시된 수도 코드는 그래프 탐색 알고리즘에 포함되는 연결선 식별 알고리즘을 나타낸 것이다.

- [0048] 라인(line) 1은 위상 정렬을 실행하여 그 결과를 큐(Q)에 인큐함을 나타내고, 라인 2는 정점들이 큐에 인큐되면 알고리즘이 시작됨을 나타낸다.
- [0049] 라인 3-4,20은 큐가 엠티(empty) 상태가 될 때까지, 큐로부터 정점을 디큐하는 동작을 반복함을 나타낸다.
- [0050] 라인 5-7은 디큐된 정점의 프로파일 연결선 리스트의 모든 값이 '1'인지 확인하고, 모든 값이 '1'일 경우에는 알고리즘을 종료함을 나타낸다.
- [0051] 라인 8-9,19는 새로운 정점을 탐색할 때마다 프로파일 연결선 리스트와 프로파일 경로를 생성하고, 이를 새로운 정점의 속성에 저장함을 나타낸다.
- [0052] 라인 10-11,15는 시작점으로부터 해당 정점까지의 경로에 새로운 연결선이 추가될 경우, 프로파일 연결선 리스트에 새로운 연결선 정보를 추가함을 나타낸다.
- [0053] 라인 12-14는 해당 정점에 두개 이상의 진입 연결선이 연결되어 두개 이상의 프로파일 연결선 리스트를 가질 경우, 프로파일 연결선 리스트들 간의 포함 관계를 판단하여 하나의 프로파일 연결선 리스트를 결정함을 나타낸다.
- [0054] 라인 16-18은 시작점으로부터 해당 정점까지의 경로에 새로운 연결선이 없을 경우, 선행자 정점의 프로파일 연결선 리스트를 상속받음을 나타낸다.
- [0055] 도 5는 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 탐색하는 동작을 도시한 순서도이다.
- [0056] 도 5를 참조하면, 본 발명의 실시예에 따른 그래프 탐색 장치는 딥 러닝 모델의 그래프 데이터가 입력되면, (S502) 입력된 그래프를 구성하는 정점들 V_i 의 속성 정보에 포함된 프로파일 연결선 리스트 $L(V_i)$ 와 프로파일 경로 $P(V_i)$ 를 초기화한다. (S504) 그런 다음 정점들 V_i 에 대한 위상 정렬을 수행하고, (S505) 위상 정렬 순서에 따라 V_i 를 탐색한다. (S508)
- [0057] S510에서 그래프 탐색 장치는 탐색이 완료된 V_i 에 대한 진입 연결선이 복수개인지 확인하고, 상기 V_i 에 대한 진입 연결선이 복수개가 아닐 경우 S512로 진행하여 선행자 정점의 프로파일 연결선 리스트 L과 진입 연결선 정보를 기반으로 프로파일 연결선 리스트 $L(V_i)$ 와 프로파일 경로 $P(V_i)$ 를 도출하여 저장한다.
- [0058] 한편, S510 확인 결과 상기 V_i 에 대한 진입 연결선이 복수개인 경우, S514로 진행하여 진입 연결선 각각에 대한 선행자 정점의 프로파일 연결선 리스트 L과 진입 연결선 정보를 기반으로 후보 프로파일 연결선 리스트 $L(V_i)$ 를 도출한다. 여기서 후보 $L(V_i)$ 개수는 상기 V_i 의 진입 연결선 개수와 동일하다. 그런 다음 후보 $L(V_i)$ 들 간의 포함 관계를 판단하여 상위 집합에 해당하는 $L(V_i)$ 와 이에 대응하는 $P(V_i)$ 를 도출하여 저장한다. (S516)
- [0059] S518에서 그래프 탐색 장치는 저장된 프로파일 연결선 리스트 $L(V_i)$ 의 모든 값이 '1'인지 확인한다. S518 확인 결과 $L(V_i)$ 의 모든 값이 '1'이면, 그래프 탐색 장치는 모든 연결선이 탐색된 것으로 판단하고 S520으로 진행하여 저장된 $P(V_i)$, 즉 모든 연결선이 탐색된 $P(V_i)$ 를 출력한 뒤 그래프 탐색을 종료한다.
- [0060] 한편, S518 확인 결과 $L(V_i)$ 의 모든 값이 '1'이 아니면 S508로 진행하여 다음 V_i 를 탐색한다.
- [0061] 본 발명의 실시예에서는 프로파일 연결선 리스트 $L(V_i)$ 의 인덱스 값이 0 또는 1로 설정되어 연결선의 탐색 여부를 나타낸다고 예를 들어 설명하였다. 그러나 상기 연결선의 탐색 여부는 미리 정해지는 특정 값으로 판단될 수도 있으며, 이 경우 S518은 프로파일 연결선 리스트 $L(V_i)$ 의 값이 모든 연결선이 탐색되었음을 나타내는 특정 값을 만족하는지 확인하는 단계로 대체될 수 있다.
- [0062] 도 6은 본 발명의 실시예에 따른 딥 러닝 모델의 그래프를 탐색하는 장치를 도시한 블록도이다.
- [0063] 도 6을 참조하면, 본 발명의 실시예에 따른 그래프 탐색 장치(600)는 입력부(610), 출력부(620) 및 프로세서(630)를 포함한다.
- [0064] 그래프 탐색 장치(600)는 입력부(610)를 통해 딥 러닝 모델의 그래프 데이터를 입력 받고, 프로세서(630)를 통해 그래프 탐색을 수행한다. 즉 입력된 그래프를 구성하는 정점들 V_i 의 속성 정보에 포함된 프로파일 연결선 리

스트 $L(V_i)$ 와 프로파일 경로 $P(V_i)$ 를 초기화하고, 정점들 V_i 에 대한 위상 정렬을 수행하고, 위상 정렬 순서에 따라 V_i 를 탐색한다.

[0065] 또한 그래프 탐색 장치는 프로세서(630)를 통해 탐색이 완료된 V_i 에 대한 진입 연결선이 복수개가 아닐 경우, 선행자 정점의 프로파일 연결선 리스트 L 과 진입 연결선 정보를 기반으로 프로파일 연결선 리스트 $L(V_i)$ 와 프로파일 경로 $P(V_i)$ 를 도출하여 저장한다. 또한 상기 탐색이 완료된 V_i 에 대한 진입 연결선이 복수개인 경우에는 진입 연결선 각각에 대한 선행자 정점의 프로파일 연결선 리스트 L 과 진입 연결선 정보를 기반으로 상기 진입 연결선 개수에 상응하는 후보 프로파일 연결선 리스트 $L(V_i)$ 를 도출하고, 후보 $L(V_i)$ 들 간의 포함 관계를 판단하여 상위 집합에 해당하는 $L(V_i)$ 와 이에 대응하는 $P(V_i)$ 를 도출하여 저장한다.

[0066] 또한 그래프 탐색 장치는 저장된 연결선 리스트 $L(V_i)$ 의 모든 값이 '1'일 경우 출력부(620)를 통해 저장된 $P(V_i)$, 즉 모든 연결선이 탐색된 $P(V_i)$ 를 출력하고 그래프 탐색을 종료한다. 또한 그래프 탐색 장치는 상기 저장된 연결선 리스트 $L(V_i)$ 의 모든 값이 '1'이 아닐 경우 프로세서(630)를 통해 위상 정렬 순서에 따른 다음 V_i 를 탐색한다.

[0067] 이와 같은 본 발명에서 제안하는 그래프 탐색 방법은 다양한 컴퓨터 구성요소를 통하여 수행될 수 있는 프로그램 명령어의 형태로 구현되어 컴퓨터 판독 가능한 기록 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능한 기록 매체는 프로그램 명령어, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다.

[0068] 상기 컴퓨터 판독 가능한 기록 매체에 기록되는 프로그램 명령어는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 분야의 당업자에게 공지되어 사용 가능한 것일 수도 있다.

[0069] 컴퓨터 판독 가능한 기록 매체의 예에는, 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD 와 같은 광기록 매체, 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 ROM, RAM, 플래시 메모리 등과 같은 프로그램 명령어를 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다.

[0070] 프로그램 명령어의 예에는, 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드도 포함된다. 상기 하드웨어 장치는 본 발명에 따른 처리를 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

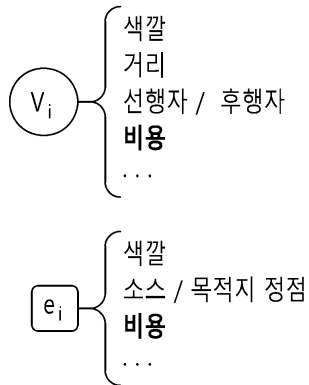
[0071] 이상에서는 본 발명의 다양한 실시예에 대하여 도시하고 설명하였지만, 본 발명은 상술한 특징의 실시예에 한정되지 아니하며, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 당해 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 다양한 변형실시가 가능한 것은 물론이고, 이러한 변형실시들은 본 발명의 기술적 사상이나 전방으로부터 개별적으로 이해되어져서는 안될 것이다.

부호의 설명

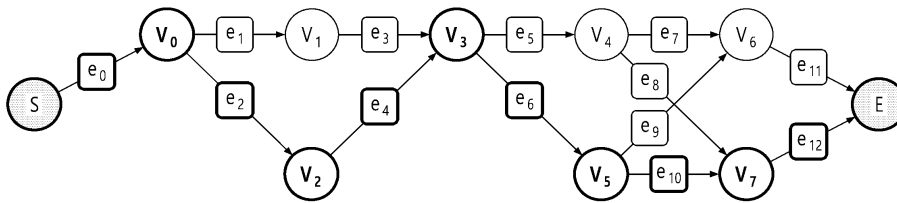
[0073] 600: 그래프 탐색 장치

도면

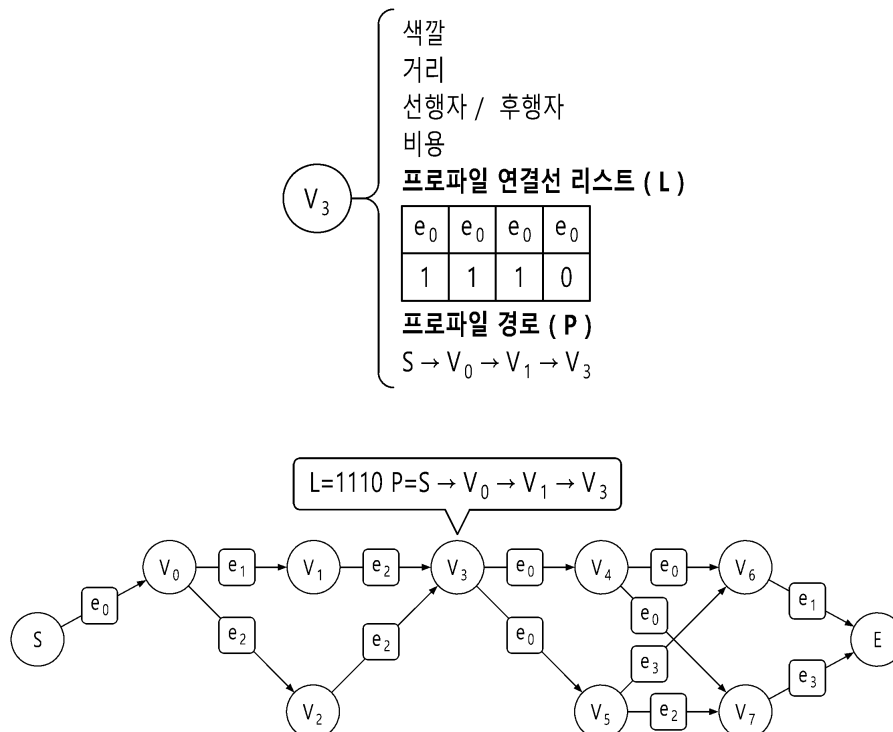
도면1a



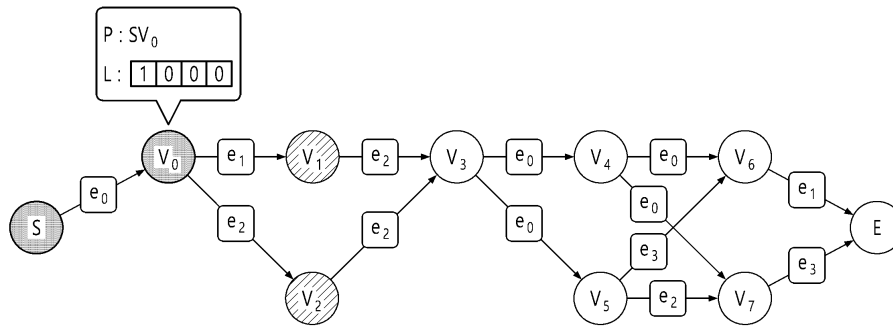
도면1b



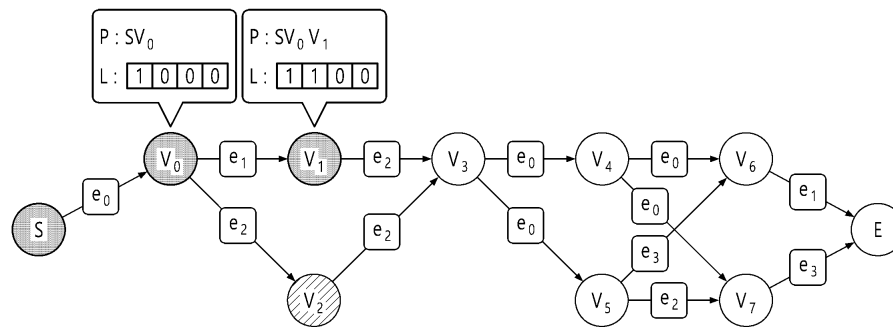
도면2



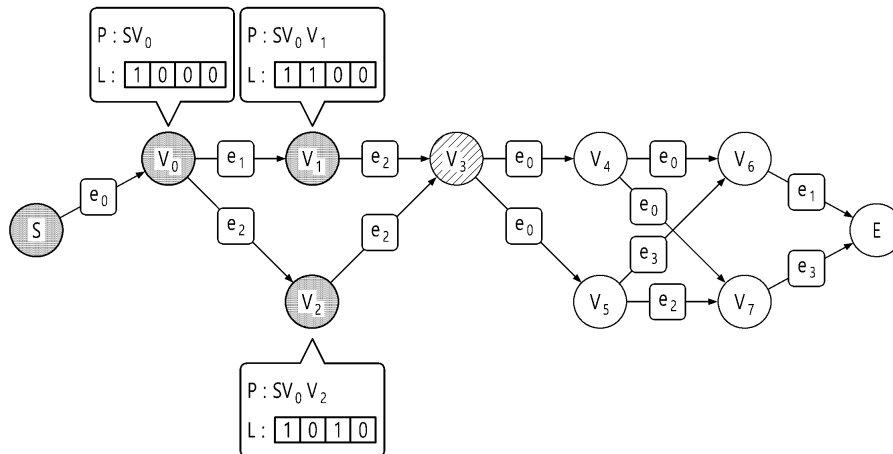
도면3a



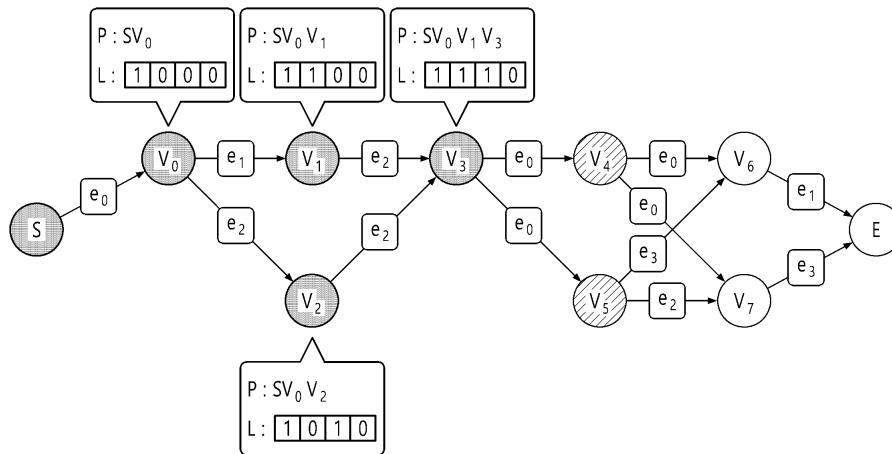
도면3b



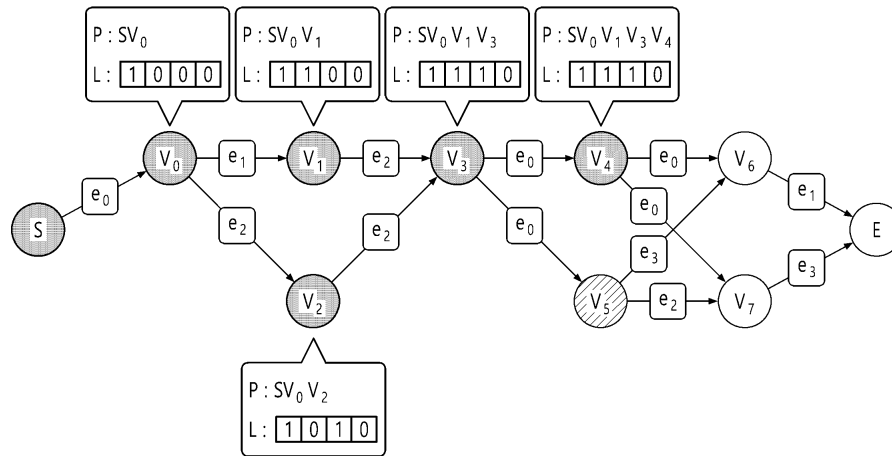
도면3c



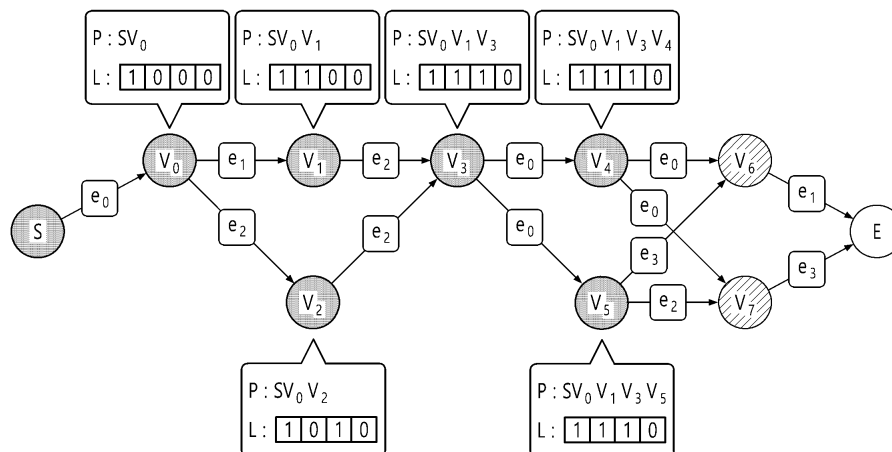
도면3d



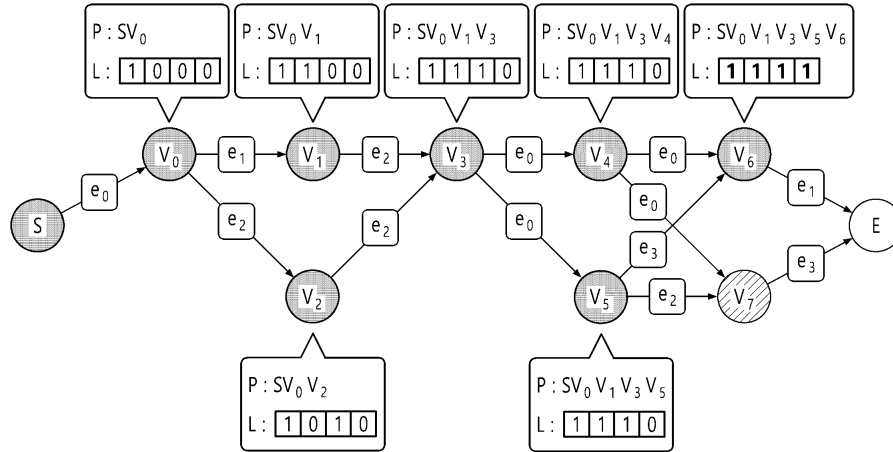
도면3e



도면3f



도면3g



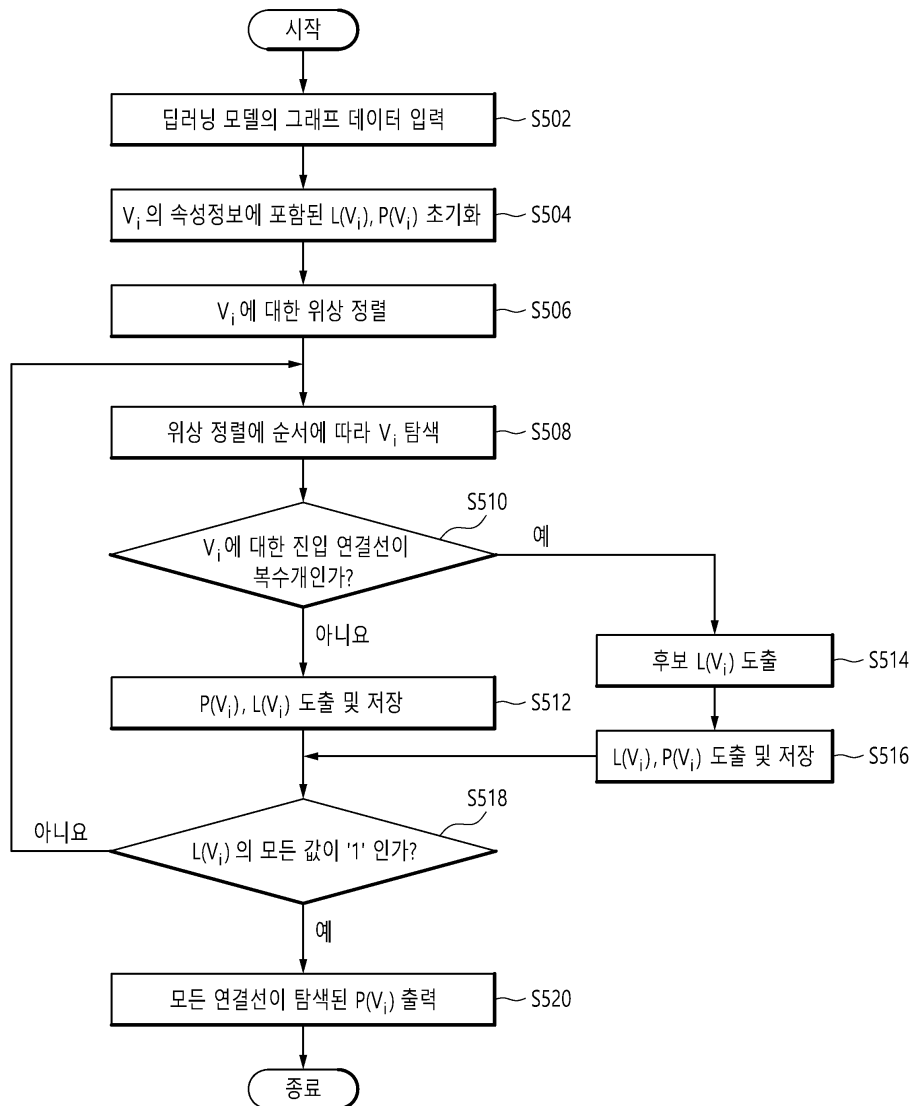
도면4

Algorithm 1: Edge Identification Algorithm

```

Line 1: Perform the topological sort and enqueue the result in Q
Line 2: Q ← Enqueue(Start)
Line 3: while not empty in Q do
Line 4:   u = dequeue(Q)
Line 5:   if u.profiled_edge_list all 1's then
Line 6:     break /*Done*/
Line 7:   end
Line 8:   for each incoming execution path to u do
Line 9:     Add profiled_path to u
Line 10:    if the path include new edges then
Line 11:      Add profiledgedelist to u /*Register new edges*/
Line 12:      if u has more than on profield_edge_list then
Line 13:        Determine thier inclusion and summarize u.profiled_edge_list
Line 14:      end
Line 15:    end
Line 16:  else
Line 17:    Add the predecessor's profiled_edge_list to u /*Inherit from predecessors*/
Line 18:  end
Line 19: end
Line 20: end
  
```

도면5



도면6

600

