



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0124106
(43) 공개일자 2022년09월13일

- (51) 국제특허분류(Int. Cl.)
G06N 3/08 (2006.01) G06N 20/00 (2019.01)
G06N 3/04 (2006.01) G06N 5/02 (2006.01)
H04L 41/16 (2022.01)
- (52) CPC특허분류
G06N 3/08 (2013.01)
G06N 20/00 (2021.08)
- (21) 출원번호 10-2022-0026209
- (22) 출원일자 2022년02월28일
심사청구일자 2022년02월28일
- (30) 우선권주장
1020210027452 2021년03월02일 대한민국(KR)

- (71) 출원인
포항공과대학교 산학협력단
경상북도 포항시 남구 청암로 77 (지곡동)
- (72) 발명자
홍원기
경상북도 포항시 남구 지곡로 319, 328동 304호
유재형
서울특별시 송파구 올림픽로 135, 211동 1303호
김희곤
대구광역시 남구 성당로 162-4
- (74) 대리인
특허법인이상

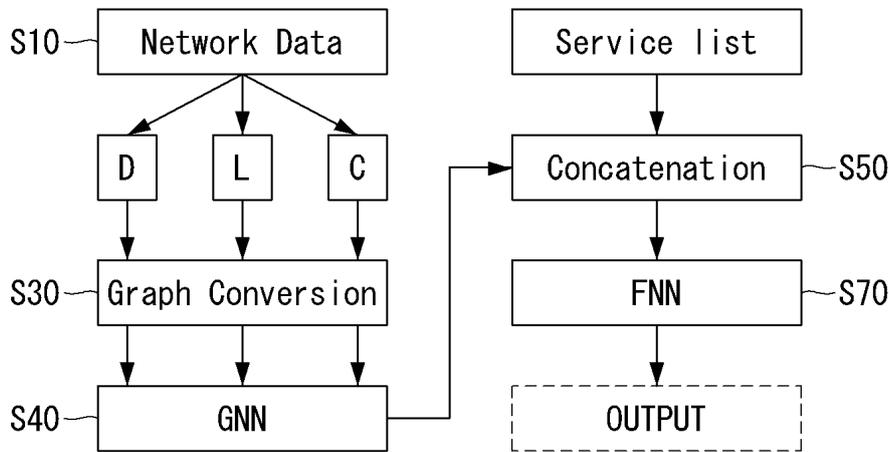
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 그래프 신경망 기반 가상 네트워크 관리 방법 및 장치

(57) 요약

그래프 신경망 기반 가상 네트워크 관리 방법 및 장치가 개시된다. 가상 네트워크 관리 방법은, 수집된 네트워크 데이터를 전처리하는 단계, 전처리하는 단계에서 생성된 카테고리별 데이터 중 노드와 엣지 데이터들을 그래프로 변환하는 단계, 그래프로 변환하는 단계에서 생성된 행렬 표현의 네트워크 데이터를 그래프 신경망(GNN)을 사용하여 학습하는 단계, 및 GNN을 사용한 학습을 통해 생성한 각 노드에 대한 노드 상태 정보를 서비스 리스트 데이터와 함께 순전파 신경망(FNN)을 사용하여 학습하는 단계를 포함한다.

대표도 - 도1



(52) CPC특허분류

G06N 3/0481 (2013.01)

G06N 5/02 (2019.01)

H04L 41/16 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711102868
과제번호	2018-0-00749-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	방송통신산업기술개발
연구과제명	인공지능 기반 가상 네트워크 관리기술 개발
기여율	1/1
과제수행기관명	포항공과대학교 산학협력단
연구기간	2020.01.01 ~ 2020.12.31

명세서

청구범위

청구항 1

그래프 신경망 기반 가상 네트워크 관리 방법으로써,

수집된 네트워크 데이터를 전처리하는 단계;

상기 전처리하는 단계에서 생성된 카테고리별 데이터 중 노드와 엣지 데이터들을 그래프로 변환하는 단계;

상기 그래프로 변환하는 단계에서 생성된 행렬 표현의 네트워크 데이터를 그래프 신경망(GNN: Graph Neural Network)을 사용하여 학습하는 단계; 및

상기 GNN을 사용한 학습을 통해 생성한 각 노드에 대한 노드 상태 정보를 서비스 리스트 데이터와 함께 순전파 신경망(FNN: Feedforward Neural Network)을 사용하여 학습하는 단계를 포함하는 가상 네트워크 관리 방법.

청구항 2

청구항 1에 있어서,

상기 FNN을 사용하여 학습하는 단계는, 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 제1 클래스의 데이터 개수가 다른 제2 클래스의 데이터 개수보다 적을 경우, 제1 클래스를 학습하는 데 부여하는 제1 가중치를 제2 클래스를 학습하는데 부여하는 제2 가중치보다 더 크게 설정하는, 가상 네트워크 관리 방법.

청구항 3

청구항 1에 있어서,

상기 FNN을 사용하여 학습하는 단계 후에, 학습된 모든 노드들에 대해 생성, 제거 또는 그대로 두기를 포함한 가상 네트워크 기능(VNF: Virtual Network Function) 관리 결정을 생성하는 단계를 더 포함하는 가상 네트워크 관리 방법.

청구항 4

청구항 1에 있어서,

상기 전처리하는 단계 전에,

기계학습에 사용되는 데이터를 정의하는 단계를 더 포함하는 가상 네트워크 관리 방법.

청구항 5

청구항 4에 있어서,

상기 정의하는 단계는 노드와 링크로 구성되는 네트워크 데이터와 서비스 데이터를 VNF(Virtual Network Function) 열로 표현하여 정의하며,

여기서 상기 노드는 가용 CPU(central processing unit) 코어의 개수 및 가용 대역폭에 대한 정보를 가지고 있고, 상기 링크는 딜레이와 가용 대역폭에 대한 정보를 가지고 있으며, 상기 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드, 서비스 타입, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 및 요구 대역폭에 대한 정보를 포함하는, 가상 네트워크 관리 방법.

청구항 6

청구항 4에 있어서,

상기 전처리하는 단계 전에 및 상기 정의하는 단계 후에,

상기 데이터의 정의에 기초하여 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스

정보 리스트를 생성하는 단계를 더 포함하며,

상기 서비스 정보 리스트는 서비스 리스트 정보 아이디를 갖고 서비스 시작 시간과 서비스 실행 시간을 이용하여 정의되는 서비스들과 가장 최근에 생성된 서비스를 합하여 생성되는, 가상 네트워크 관리 방법.

청구항 7

청구항 6에 있어서,

상기 전처리하는 단계 전에 및 상기 생성하는 단계 후에,

상기 서비스 정보 리스트가 생성될 때 물리 네트워크 정보와 가상 네트워크 기능(VNF: Virtual Network Function) 정보를 수집하는 단계를 더 포함하는 가상 네트워크 관리 방법.

청구항 8

청구항 7에 있어서,

상기 전처리하는 단계 전에 및 상기 수집하는 단계 후에,

상기 물리 네트워크 정보, 상기 VNF 정보 및 서비스 정보를 토대로 기계학습에 활용할 정답 데이터를 준비하는 단계를 더 포함하는 가상 네트워크 관리 방법.

청구항 9

청구항 8에 있어서,

상기 정답 데이터를 준비하는 단계는, VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용 또는 이들의 조합을 포함한 정답 데이터 셋을 생성하거나 기저장된 정답 데이터 셋을 활용하는, 가상 데이터 관리 방법.

청구항 10

청구항 1에 있어서,

상기 전처리하는 단계는, 상기 네트워크 데이터 중 모든 수치 데이터를 정규화하여 표현하고, 상기 네트워크 데이터 중 카테고리별 데이터들을 숫자 또는 벡터로 바꾸어 표현하는, 가상 데이터 관리 방법.

청구항 11

청구항 1에 있어서,

상기 그래프로 변환하는 단계는, 각 노드 정보와 엣지 정보 및 노드와 노드 간과 노드와 엣지 간의 연결 정보를 가진 노드 행렬, 엣지 행렬 및 연결 행렬을 생성하며,

여기서, 상기 노드 행렬은 노드 수 \times 노드 피처의 행렬이고,

상기 엣지 행렬은 노드 수 \times 엣지 피처의 행렬이며,

상기 연결 행렬은 노드 수 \times 노드 수 행렬로 노드들 간의 링크가 존재하는 경우는 1의 값을 갖고, 상기 노드들 간의 링크가 존재하지 않는 경우는 0의 값을 가지는, 가상 데이터 관리 방법.

청구항 12

그래프 신경망 기반 가상 네트워크 관리 장치로서,

수집된 네트워크 데이터를 전처리하는 전처리부;

상기 전처리부에서 생성된 카테고리별 데이터 중 노드와 엣지 데이터를 그래프로 변환하는 변환부;

상기 변환부에서 생성된 행렬 표현의 네트워크 데이터를 그래프 신경망(GNN: Graph Neural Network)을 사용하여 학습하는 제1 학습부; 및

상기 제1 학습부를 통해 생성한 각 노드에 대한 노드 상태 정보를 서비스 리스트 데이터와 함께 순전파 신경망

(FNN: Feedforward Neural Network)을 사용하여 학습하는 제2 학습부를 포함하는 가상 네트워크 관리 장치.

청구항 13

청구항 12에 있어서,

상기 제2 학습부는, 학습 데이터의 클래스 불균형(Class Imbalance)를 막기 위해 제1 클래스의 데이터 개수가 다른 제2 클래스의 데이터 개수보다 적을 경우, 제1 클래스를 학습하는 데 부여하는 제1 가중치를 제2 클래스를 학습하는데 부여하는 제2 가중치보다 더 크게 설정하는 가중치 조정부를 구비하는, 가상 네트워크 관리 장치.

청구항 14

청구항 12에 있어서,

상기 제2 학습부에 사용되는 손실 함수(Loss Function)는 사용자의 목적에 따라 변경 가능하며, 상기 제2 학습부의 학습 모델은 상기 손실 함수에 따라 특정 정보에 대해 학습하도록 구비되는, 가상 네트워크 관리 장치.

청구항 15

청구항 12에 있어서,

상기 제2 학습부는, 상기 제2 학습부에서 학습 결과로 출력되는 모든 노드들에 대해 생성, 제거 또는 그대로 두기를 포함한 가상 네트워크 기능(VNF: Virtual Network Function) 관리 결정을 생성하는 출력부를 구비하는 가상 네트워크 관리 장치.

청구항 16

청구항 12에 있어서,

상기 전처리부는, 상기 네트워크 데이터 중 모든 수치 데이터를 정규화하여 표현하고, 상기 네트워크 데이터 중 카테고리별 데이터들을 숫자 또는 벡터로 바꾸어 표현하는, 가상 데이터 관리 장치.

청구항 17

청구항 12에 있어서,

상기 변환부는, 각 노드 정보와 엣지 정보 및 노드와 노드 간과 노드와 엣지 간의 연결 정보를 가진 노드 행렬, 엣지 행렬 및 연결 행렬을 생성하며,

여기서, 상기 노드 행렬은 노드 수 × 노드 피처의 행렬이고,

상기 엣지 행렬은 노드 수 × 엣지 피처의 행렬이며,

상기 연결 행렬은 노드 수 × 노드 수 행렬로 노드들 간의 링크가 존재하는 경우는 1의 값을 갖고, 상기 노드들 간의 링크가 존재하지 않는 경우는 0의 값을 가지는, 가상 데이터 관리 장치.

청구항 18

청구항 12에 있어서,

기계학습에 사용되는 데이터를 정의하는 데이터 정의부;

상기 데이터의 정의에 기초하여 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스 정보 리스트를 생성하는 데이터 생성부; 및

상기 서비스 정보 리스트가 생성될 때 물리 네트워크 정보와 가상 네트워크 기능(VNF: Virtual Network Function) 정보를 수집하는 데이터 수집부;

를 더 포함하고,

상기 서비스 정보 리스트는 서비스 리스트 정보 아이디를 갖고 서비스 시작 시간과 서비스 실행 시간을 이용하여 정의되는 서비스들과 가장 최근에 생성된 서비스를 합하여 생성되는, 가상 네트워크 관리 장치.

청구항 19

청구항 18에 있어서,

상기 데이터 정의부는 노드와 링크로 구성되는 네트워크 데이터와 서비스 데이터를 VNF(Virtual Network Function) 열로 표현하여 정의하며,

여기서 상기 노드는 가용 CPU(central processing unit) 코어의 개수 및 가용 대역폭에 대한 정보를 가지고 있고, 상기 링크는 딜레이와 가용 대역폭에 대한 정보를 가지고 있으며, 상기 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드, 서비스 타입, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 및 요구 대역폭에 대한 정보를 가지는, 가상 네트워크 관리 장치.

청구항 20

청구항 18에 있어서,

상기 데이터 생성부는 상기 물리 네트워크 정보, 상기 VNF 정보 및 서비스 정보를 토대로 기계학습에 활용할 정답 데이터로써, VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용, 다른 네트워크 관리 비용 또는 이들의 조합을 포함한 정답 데이터 셋을 생성하거나 기저장된 정답 데이터 셋을 활용하는, 가상 네트워크 관리 장치.

발명의 설명

기술 분야

[0001] 본 발명은 가상 네트워크 관리 기술에 관한 것으로, 보다 상세하게는 그래프 신경망을 기반으로 하는 가상 네트워크 관리 방법 및 장치에 관한 것이다.

배경 기술

[0002] 네트워크 기능 가상화(NFV: Network Function Virtualization) 기술은 공용으로 사용되는 물리 네트워크 인프라 상에서 복수의 사용자가 가상 네트워크를 생성하여 독립적으로 사용할 수 있게 하는 기술을 지칭한다. 이 기술은 클라우드 컴퓨팅과 미래 인터넷 기술의 등장으로 모든 자원을 가상화하고 인프라와 서비스 제공자를 분리하는 등 다양한 형태로 진화하고 있다.

[0003] 또한, 소프트웨어 정의 네트워킹(SDN: Software Defined Networking) 패러다임을 활용하여 네트워크 가상화 기술을 실현하려는 연구가 다수 진행되고 있다. 소프트웨어 정의 네트워킹은 기존의 하드웨어 기반의 네트워킹 방식에서 진일보하여 관리자가 네트워크 자체를 프로그래밍할 수 있게 하며, 이를 바탕으로 동적인 네트워크 환경에서 민첩성(Agility), 탄력성(Elasticity), 유연성(Flexibility) 등에 대한 장점을 제공할 수 있다.

[0004] SDN 및 NFV를 사용하면 관리자가 네트워크를 중앙에서 동적으로 관리할 수 있으므로 네트워크를 보다 효율적으로 관리할 수 있다. SDN 컨트롤러 및 NFV 관리자는 네트워크 및 NFV 환경에 대한 전역 뷰(global views)를 제공하며 운영자는 이를 사용하여 네트워크 및 서비스 오케스트레이션을 관리할 수 있다. 또한 가상 네트워크 기능(VNF: Virtual Network Functions)을 확장하고 최적화하여 리소스의 과잉 프로비저닝을 방지하고 고가용성을 제공할 수 있다. 그러나 SDN/NFV는 효율적인 네트워크 관리를 가능하게 하지만 가상 네트워크 관리를 위한 최적의 관리 솔루션을 제공하지는 않는다.

[0005] 정수 선형 계획법(ILP: Integer Linear Programming)은 네트워크 관리를 위한 최적화 방법 중 하나이다. ILP는 선형 비용 함수를 최소화하는 것을 목표로 하고 선형 등식(linear equality) 및 부등식 제약조건(inequality constraints) 세트를 사용하여 최적의 솔루션을 얻는다. 그러나 ILP 기반의 네트워크 관리 방법은 최적의 솔루션을 찾는 데 시간이 비교적 오래 걸리므로 실시간 네트워크 관리에는 적합하지 않다.

[0006] 현재, 네트워크 관리는 전적으로 사람의 판단에 의존하므로 전문적인 지식을 가진 운용관리 인력을 필요로 한다. 하지만 이러한 운용인력의 확보에는 큰 비용이 소모되며, 네트워크가 복잡할수록 관리는 더더욱 어려워진다. 특히, 선형계획법(ILP: Integer Linear Programming)으로 네트워크의 여러 조건들을 사용하여 네트워크를 최적화하는 선형식을 도출해내고 있지만, 선형식을 도출하기까지 너무 많은 시간이 소요되는 문제가 있다. 이러한 문제를 해결하기 위해 최근에는 네트워크 관리에 기계학습 기술을 도입하여 빠른 시간내에 네트워크를 최적화하는 시도가 이루어지고 있다.

[0007] 최근 기계학습(ML: Machine Learning)은 다양한 네트워킹 문제를 해결하고 네트워크 관리를 자동화하는 새로운

패러다임으로 부상하고 있다. ML은 명시적으로 프로그래밍하지 않고도 경험을 통해 자동으로 학습하고 개선하는 모델을 제공한다. 즉, ML은 사람의 도움없이 컴퓨터 소프트웨어가 스스로 학습하여 문제를 해결하는 방법을 말하며 크게 지도학습(Supervised Learning), 비지도학습(Unsupervised Learning), 강화학습(Reinforcement Learning) 방법이 존재한다. 이러한 ML은 학습하는 데 시간이 걸리지만 학습 후에는 시간이 거의 걸리지 않는다. 또한 ML은 통계적 방법보다 광범위하고 동적으로 변화하는 데이터를 학습하는 데 더 효과적이다. 그러나 ML의 전문한 장점에도 불구하고 ML을 네트워크 관리에 적용하는 것은 쉽지 않다.

[0008] 한편, ML 기술 분야에서는 여러 비선형 변환기법을 통한 추상화로 사람의 사고방식과 유사한 방법인 딥러닝 기술을 적용하고 있으며 지도학습과 비지도학습, 강화학습에 적용되어 높은 성능의 향상을 가져오고 있다. 이는 컴퓨팅 성능의 향상과 빅 데이터, 그리고 새로운 알고리즘의 개발로 기존의 많은 문제를 해결하였기 때문이다. 이러한 ML을 네트워크 관리에 이용하기 위해서는 충분한 양질의 네트워크 데이터와 네트워크를 잘 학습할 수 있도록 특화된 기계학습 모델이 필요하다.

[0009] 그러나 네트워크 관리에 사용할 수 있는 데이터는 현재 수십 개에 불과하기 때문에 기존의 ML을 적용한 네트워크 관리 기술 대부분에서는 학습에 적합한 충분한 데이터를 사용하기가 어렵다. 또한 기존 기술들 대부분에서는 네트워크 관리를 위해 특화된 ML 모델이 아닌 보편화된 ML 모델을 이용하고 학습에 단순 수치 데이터를 이용하므로 관리하고자 하는 네트워크의 구조나 토폴로지를 적절하게 반영하기가 어렵다. 이와 같이, 기존의 ML을 활용한 네트워크 관리 기술은 네트워크 관리에서의 단순한 문제 해결에 국한되어 있고 다른 ML 연구 분야에 비해 ML의 장점을 충분히 보여주지 못하는 한계가 있다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 종래 기술의 문제점을 해결하기 위해 도출된 것으로, 본 발명의 목적은 네트워크 가상화 환경에서 그래프 신경망(GNN: Graph Neural Network)을 사용하여 네트워크 데이터를 효율적으로 학습하고 최적의 가상 네트워크 기능(VNF: Virtual Network Function)의 관리 모델을 만들어 네트워크에 실시간 적용할 수 있는, 가상 네트워크 관리 방법 및 장치를 제공하는데 있다.

[0011] 다시 말해서, 본 발명의 목적은 VNF 관리를 위해서 어떠한 정보가 기계학습에 사용되어야 하는지 명시하고 네트워크 데이터를 학습하는 데 적합한 알고리즘으로 GNN을 사용하는 방안을 제시함으로써, 가상화된 네트워크 환경에서 GNN을 사용하여 효율적이고 빠른 시간 내에 최적의 VNF 관리 정책을 찾아 네트워크에 적용할 수 있고 그에 의해 네트워크 관리 비용을 줄일 수 있는, 가상 네트워크 관리 방법 및 장치를 제공하는데 있다.

[0012] 본 발명의 다른 목적은 가상 네트워크 환경에서 각 노드의 최적의 VNF 배치 정책에 기초하여 전체 네트워크 운용 비용을 크게 줄일 수 있는, 가상 네트워크 관리 방법 및 장치를 제공하는데 있다.

[0013] 다시 말해서, 본 발명의 다른 목적은 네트워크의 트래픽 정보와 네트워크의 자원 정보 등을 입력받고 기계학습 기반으로 빠른 시간에 각 노드의 최적의 VNF 관리 정책을 찾아 네트워크에 반복 적용함으로써 네트워크 운영비용을 최소화하면서 네트워크 트래픽을 효과적으로 처리할 수 있는 가상 네트워크 관리 방법 및 장치를 제공하는데 있다.

과제의 해결 수단

[0014] 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 그래프 신경망 기반 가상 네트워크 관리 방법은, 수집된 네트워크 데이터를 전처리하는 단계; 상기 전처리하는 단계에서 생성된 카테고리별 데이터 중 노드와 엣지 데이터들을 그래프로 변환하는 단계; 상기 그래프로 변환하는 단계에서 생성된 행렬 표현의 네트워크 데이터를 그래프 신경망(GNN: Graph Neural Network)을 사용하여 학습하는 단계; 및 상기 GNN을 사용한 학습을 통해 생성한 각 노드에 대한 노드 상태 정보를 서비스 리스트 데이터와 함께 순전파 신경망(FNN: Feedforward Neural Network)을 사용하여 학습하는 단계를 포함한다.

[0015] 일실시예에서, 상기 FNN을 사용하여 학습하는 단계는, 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 제1 클래스의 데이터 개수가 다른 제2 클래스의 데이터 개수보다 적을 경우, 제1 클래스를 학습하는 데 부여하는 제1 가중치를 제2 클래스를 학습하는데 부여하는 제2 가중치보다 더 크게 설정할 수 있다.

[0016] 일실시예에서, 가상 네트워크 관리 방법은, 상기 FNN을 사용하여 학습하는 단계 후에, 학습된 모든 노드들에 대해 생성, 제거 또는 그대로 두기를 포함한 가상 네트워크 기능(VNF: Virtual Network Function) 관리 결정을 생

성하는 단계를 더 포함할 수 있다.

- [0017] 일실시예에서, 가상 네트워크 관리 방법은, 상기 전처리하는 단계 전에, 기계학습에 사용되는 데이터를 정의하는 단계를 더 포함할 수 있다.
- [0018] 일실시예에서, 가상 네트워크 관리 방법은, 상기 전처리하는 단계 전에 및 상기 정의하는 단계 후에, 상기 데이터의 정의에 기초하여 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스 정보 리스트를 생성하는 단계를 더 포함할 수 있다.
- [0019] 일실시예에서, 가상 네트워크 관리 방법은, 상기 전처리하는 단계 전에 및 상기 생성하는 단계 후에, 상기 서비스 정보 리스트가 생성될 때 물리 네트워크 정보와 가상 네트워크 기능(VNF: Virtual Network Function) 정보를 수집하는 단계를 더 포함할 수 있다.
- [0020] 일실시예에서, 가상 네트워크 관리 방법은, 상기 전처리하는 단계 전에 및 상기 수집하는 단계 후에, 상기 물리 네트워크 정보, 상기 VNF 정보 및 서비스 정보를 토대로 기계학습에 활용할 정답 데이터를 준비하는 단계를 더 포함할 수 있다.
- [0021] 일실시예에서, 상기 정답 데이터를 준비하는 단계는, VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용 또는 이들의 조합을 포함한 정답 데이터 셋을 생성하거나 기저장된 상기 정답 데이터 셋을 활용할 수 있다.
- [0022] 일실시예에서, 상기 전처리하는 단계는, 상기 네트워크 데이터 중 모든 수치 데이터를 정규화하여 표현하고, 상기 네트워크 데이터 중 카테고리별 데이터들을 숫자 또는 벡터로 바꾸어 표현할 수 있다.
- [0023] 일실시예에서, 상기 그래프로 변환하는 단계는, 각 노드 정보와 엣지 정보 및 노드와 노드 간과 노드와 엣지 간의 연결 정보를 가진 노드 행렬, 엣지 행렬 및 연결 행렬을 생성할 수 있다.
- [0024] 상기 기술적 과제를 해결하기 위한 본 발명의 일 측면에 따른 그래프 신경망 기반 가상 네트워크 관리 장치는, 수집된 네트워크 데이터를 전처리하는 전처리부; 상기 전처리부에서 생성된 카테고리별 데이터 중 노드와 엣지 데이터들을 그래프로 변환하는 변환부; 상기 변환부에서 생성된 행렬 표현의 네트워크 데이터를 그래프 신경망(GNN: Graph Neural Network)을 사용하여 학습하는 제1 학습부; 및 상기 제1 학습부를 통해 생성한 각 노드에 대한 노드 상태 정보를 서비스 리스트 데이터와 함께 순전파 신경망(FNN: Feedforward Neural Network)을 사용하여 학습하는 제2 학습부를 포함한다.
- [0025] 일실시예에서, 상기 제2 학습부는, 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 제1 클래스의 데이터 개수가 다른 제2 클래스의 데이터 개수보다 적을 경우, 제1 클래스를 학습하는 데 부여하는 제1 가중치를 제2 클래스를 학습하는 데 부여하는 제2 가중치보다 더 크게 설정하는 가중치 조정부를 구비할 수 있다.
- [0026] 일실시예에서, 상기 제2 학습부에 사용되는 손실 함수(Loss Function)는 사용자의 목적에 따라 변경 가능하며, 상기 제2 학습부의 학습 모델은 상기 손실 함수에 따라 특정 정보에 대해 학습하도록 구비될 수 있다.
- [0027] 일실시예에서, 가상 네트워크 관리 장치는, 상기 제2 학습부의 출력부는, 상기 제2 학습부에서 학습 결과로 출력되는 모든 노드들에 대해 생성, 제거 또는 그대로 두기를 포함한 가상 네트워크 기능(VNF: Virtual Network Function) 관리 결정을 생성하도록 구성될 수 있다.
- [0028] 일실시예에서, 가상 네트워크 관리 장치는, 기계학습에 사용되는 데이터를 정의하는 데이터 정의부; 상기 데이터의 정의에 기초하여 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스 정보 리스트를 생성하는 데이터 생성부; 및 상기 서비스 정보 리스트가 생성될 때 물리 네트워크 정보와 가상 네트워크 기능(VNF: Virtual Network Function) 정보를 수집하는 데이터 수집부를 더 포함할 수 있다.
- [0029] 일실시예에서, 상기 데이터 정의부는 노드와 링크로 구성되는 네트워크 데이터와 서비스 데이터를 VNF(Virtual Network Function) 열로 표현하여 정의할 수 있다. 여기서 상기 노드는 가용 CPU(central processing unit) 코어의 개수 및 가용 대역폭에 대한 정보를 가지고 있고, 상기 링크는 딜레이와 가용 대역폭에 대한 정보를 가지고 있으며, 상기 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드, 서비스 타입, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 및 요구 대역폭에 대한 정보를 포함할 수 있다.
- [0030] 일실시예에서, 상기 데이터 생성부는 상기 물리 네트워크 정보, 상기 VNF 정보 및 서비스 정보를 토대로 기계학습에 활용할 정답 데이터로써, VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용, 또 다른 네트워크 관리 비용 또는 이들의 조합을 포함한 정답 데이터 셋을 생성하도록 구성되거나 기저장된 정

답 데이터 셋을 활용하도록 구성될 수 있다.

발명의 효과

- [0031] 본 발명에 의하면, 가상 네트워크 환경에서 GNN(Graph Neural Network)을 사용하여 네트워크 데이터를 효율적으로 학습하고 서비스 리스트에 따라 다시 학습을 통해 최적의 VNF(Virtual Network Function) 관리 정책을 찾고 이를 네트워크에 적용함으로써 네트워크 운영비용(OPEX: Operating Expenditure)과 자본적지출(CAPEX: Capital Expenditure)을 감소시킬 수 있다.
- [0032] 또한, 본 발명에 의하면, 네트워크 관리자는 사용자 요구와 전체 네트워크 상태를 고려하면서 전체 노드들이 VNF 종류별로 적합한 정책 예컨대 추가, 제거, 가만히 두기 등에 대한 정책을 결정할 수 있으며, 이를 통해 기존의 ILP(Integer Linear Programming) 방법에서는 빠르게 찾기가 불가능하던 최적의 VNF 정책을 기계학습 기반으로 빠른 시간 내에 찾을 수 있다.
- [0033] 또한, 본 발명에 의하면, 기존의 기계학습 모델들에서 네트워크 데이터를 제대로 활용하지 못하여 보편적인 관리 정책밖에 학습하지 못하였던 한계를 개선할 수 있다. 즉, 본 발명에서는 네트워크 데이터를 실제 환경 또는 시뮬레이션 환경에서 수집하고 수집된 데이터를 GNN과 순전파 신경망(FNN: Feedforward Network)을 통해 학습하여 네트워크의 각 노드에 대해서 VNF 타입별 최적 인스턴스(Instance) 개수를 도출함으로써 복잡한 VNF 관리 문제를 해결할 수 있다.
- [0034] 또한, 본 발명에 의하면, 기존의 네트워크 관리 방법에서 네트워크에 필요한 전체 VNF 개수와 같이 적은 요소만을 고려하는 것과 달리, VNF의 종류, VNF 설치 대상이 되는 노드를 고려하여 전반적인 VNF 관리가 가능한 장점이 있다.
- [0035] 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 상세한 설명에 대한 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

- [0036] 본 명세서에 첨부되는 도면은 본 발명에 대한 이해를 돕기 위한 것으로서 본 발명의 다양한 실시 형태들을 나타내고 명세서의 기재와 함께 본 발명의 원리를 설명하기 위한 것이다.
- 도 1은 본 발명의 일실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법을 위한 기계 학습 모델에 대한 개략적인 구성도이다.
- 도 2는 도 1의 기계 학습 모델을 이용하는 가상 네트워크 관리 방법을 적용할 수 있는 네트워크 토폴로지에 대한 예시도이다.
- 도 3은 도 2의 네트워크 토폴로지의 적어도 일부 노드에 적용할 수 있는 가상 네트워크 관리 장치에 대한 개략적인 블록도이다.
- 도 4는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 네트워크의 정규화된 트래픽 패턴에 대한 예시도이다.
- 도 5는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 네트워크 데이터를 설명하기 위한 예시도이다.
- 도 6a 내지 도 6c는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 라벨 데이터 생성 과정을 설명하기 위한 도면들이다.
- 도 7은 본 발명의 또 다른 실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법을 위한 기계 학습 모델에 대한 개략적인 구성도이다.
- 도 8은 도 7의 기계 학습 모델을 이용하는 가상 네트워크 관리 장치의 소프트웨어 모듈들에 대한 개략적인 블록도이다.
- 도 9는 도 8의 소프트웨어 모듈들 중 FNN 학습부에 채용할 수 있는 구성을 나타낸 블록도이다.
- 도 10은 도 7의 기계 학습 모델을 이용하는 가상 네트워크 관리 방법의 작동 원리를 설명하기 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

- [0037] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나, 이는 본 발명을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.
- [0038] 제1, 제2 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.
- [0039] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.
- [0040] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0041] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥 상 가지는 의미와 일치하는 의미를 가진 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0042] 이하, 첨부한 도면들을 참조하여, 본 발명의 바람직한 실시예를 보다 상세하게 설명하고자 한다. 본 발명을 설명함에 있어 전체적인 이해를 용이하게 하기 위하여 도면상의 동일한 구성요소에 대해서는 동일한 참조부호를 사용하고 동일한 구성요소에 대해서 중복된 설명은 생략한다.
- [0043] 도 1은 본 발명의 일실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법을 위한 기계 학습 모델에 대한 개략적인 구성도이다.
- [0044] 본 실시예에 따른 기계 학습 모델은, VNF 관리를 위해서 순전파 네트워크(FNN: Feedforward Network)와 그래프 신경망(GNN: Graph Neural Network)을 이용한다.
- [0045] 여기서, FNN은 광범위하게 사용되는 일반적인 기계학습 모델로써 데이터를 앞방향으로 전파하는 구조를 가지고 있다. FNN은 입력 데이터를 받으면 각 입력 데이터에 학습 파라미터들을 곱하거나 더한다. 이때, 각 입력 데이터는 서로 더해져서 하나의 선형 함수식을 만들고 활성화함수(Activation Function)를 통해 비선형 데이터로 변환될 수 있다. FNN은 결과 데이터를 만들기 전까지 계속해서 은닉층(Hidden Layer)을 거치며 학습하게 된다. 최종적으로 기계 학습 모델은 결과층을 통해 나온 데이터가 학습 목표가 되는 라벨 데이터(Label Data)와 일치하도록 학습 파라미터들의 수치를 조정하는 것으로 학습을 수행한다.
- [0046] GNN은 그래프 데이터를 입력 데이터로 사용하는 것으로 FNN과 다른 학습구조를 가지고 있다. 즉, FNN은 데이터를 학습시킬 때, 데이터의 형상을 무시한 채 일렬로 데이터를 표현한 다음 학습을 수행하지만, GNN은 그래프 형태의 입력 데이터를 바로 학습에 사용할 수 있다.
- [0047] GNN에 적용되는 그래프 데이터는 노드 정보(Node Feature)와 엣지 정보(Edge Feature)로 구성되어 있다. GNN은 노드 정보와 엣지 정보를 이용하여 노드 상태(Node State)를 생성하는 것을 1차적인 목표로 삼는다. 노드 상태는 각 노드들이 가지는 일종의 정보를 의미하며, 학습을 통해 자신의 상태를 계속해서 갱신한다. 노드 상태는 이전 노드 상태와 노드 정보, 이웃 노드의 노드 정보, 엣지 정보를 통해 갱신된다.
- [0048] 이때, GNN의 전이 함수(Transition Function)는 노드 상태를 업데이트하는 역할을 수행하며, 각 노드가 계속해서 이웃 노드들의 정보를 이용하여 자신의 노드 상태를 갱신하여 전체 그래프 정보를 이해할 수 있도록 한다.

또한, GNN은 결과 함수(Output Function)를 이용하여 학습 목표가 되는 결과 데이터를 생성한다. 결과 함수는 전이 함수의 반복을 통해 얻은 노드 상태 정보를 입력값으로 받으며, FNN과 같은 학습 층을 이용하여 결과 데이터를 생성할 수 있다.

- [0049] 본 실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법에서는 GNN 기반 기계 학습 모델을 이용하여 네트워크 정보를 학습하고 최적의 가상 네트워크 기능(VNF: Virtual Network Function) 배치 시나리오나 관리 정책을 생성한다.
- [0050] 일례로, 기계 학습 모델은 도 1에 도시한 바와 같이 수집된 네트워크 데이터(network data)(S10)을 미리 정의된 형태들(D, L, C)로 구분하여 그래프 형태로 변환하고(graph conversion, S30), 그래프 형태로 변환된 데이터를 GNN을 이용하여 학습하고(S40), 이와 병렬적으로 네트워크가 사용자로부터 수신한 다양한 서비스 요청(service request)을 FNN을 이용한 학습을 통해 서비스 목록(Service list)을 얻고, 위의 GNN과 FNN을 통해 학습된 두 개의 데이터를 합쳐(concatenation, S50) 다시 FNN으로 학습(S70)하도록 구성될 수 있다. 네트워크 데이터의 미리 정의된 형태는 네트워크 데이터(D), 링크 데이터(L) 및 연결 데이터(C)를 포함한다.
- [0051] 가상 네트워크 관리 방법은 전문한 기계 학습 모델과 그 작동 원리를 통해 네트워크 정보와 트래픽 정보를 반영하여 모든 노드들 각각에 대해 최적에 가까운 VNF 관리 결정을 내릴 수 있다. 각 노드의 VNF 관리 정책은 특정 VNF에 대하여 생성, 제거, 그대로 두기 중 어느 하나를 선택하여 적용하는 것을 포함할 수 있다.
- [0052] 또한, 가상 네트워크 관리 방법은, 기계 학습 모델을 학습시킬 때, 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 가중 균형(Weight Balance) 기법을 적용할 수 있다. 즉, 기계 학습 모델에 입력되는 입력 데이터 내 특정 클래스의 데이터 수가 다른 클래스에 비해 상대적으로 유의미하게 적을 경우, 기계 학습 모델이 해당 클래스를 학습하는 데 가중치를 더 주는 것으로 기계 학습 모델이 적합하게 학습하도록 작동 환경을 조정할 수 있다.
- [0053] 본 실시예에 의하면, 가상 네트워크 관리 즉 VNF 관리를 효과적으로 수행할 수 있다. VNF 관리는 서비스 요구사항(Service Requirements)을 보장하면서 네트워크 운영비용(OPEX: Operating Expenditure)을 줄이는 것을 목표로 할 수 있다. 이를 위해, 몇 가지 요구사항 예컨대 서비스 제약조건(service constraints) 및 물리적 네트워크(physical network)을 고려하면서 최적 개수의 VNF 인스턴스(instances)를 최적의 위치(서버)에 배포할 수 있다. 즉, 본 실시예에서는 모든 서버 및 VNF 유형에 대한 분류 측면에서 각 서버의 최적의 VNF 인스턴스 개수 등에 대하여 VNF 관리 정책(예컨대, 생성, 제거, 그대로 두기)을 실시간 적용하도록 구현될 수 있다.
- [0054] 도 2는 도 1의 기계 학습 모델을 이용하는 가상 네트워크 관리 방법을 적용할 수 있는 네트워크 토폴로지에 대한 예시도이다.
- [0055] 가상 네트워크 관리 방법은 복수의 노드들을 포함하는 네트워크에서 각 노드에 적용될 수 있다. 여기서, 네트워크는 노드(Node)와 링크(Link)로 구성된다. 예를 들어, 특정 네트워크는 도 2에 도시한 바와 같이 0 내지 11 중 어느 하나의 노드 번호(Node Number)로 구분되는 12개의 노드들과, 이 노드들 사이를 연결하는 15개의 링크들을 포함하도록 구성될 수 있다. 링크는 엣지(edge)로 지칭될 수 있다.
- [0056] 구체적으로, 물리 네트워크는 노드들(N)과 링크들(E)로 구성되는 무방향 그래프(G)로 표현될 수 있다. 노드들은 VNF를 배포할 수 있는 서버(s)와 VNF를 배포할 수 없는 스위치로 분류된다. 서버는 VNF를 배포할 수 있는 노드로서 이하에서 간략히 노드(s)로 언급될 수 있다.
- [0057] 미리 지정된 자원에 속한 가용 가능한 CPU(central processing unit) 코어 개수가 소정 개수(c_s)이고 VNF 배포 가능한 노드(s)에 설치된 VNF를 V_s 라고 하고 노드(s)에 설치된 VNF 데이터를 D_{V_s} 라고 하면, 각 노드(s)의 네트워크 데이터(D)는 노드 데이터 $D_s=(c_s, D_{V_s})$ 로써 표현될 수 있다.
- [0058] 물리적 링크(E)는 링크 데이터(L)와 연결 데이터(C)로 구성된다. 임의의 노드 쌍(i, j)이 물리적 링크(E)의 구성 요소일 때, 해당 링크 데이터(L_{ij})는 미리 알고 있는 자원 내 (m_{ij}, b_{ij}, d_{ij})로 정의될 수 있다. 여기서, 해당 링크의 구성요소인 m_{ij}, b_{ij} 및 d_{ij} 는 기재된 순서대로 노드 i 와 노드 j 와의 링크의 최대 허용 대역폭(maximum bandwidth), 가용 대역폭(available bandwidth), 및 지연시간(delay)을 각각 나타낸다. 또한, 링크는 해당 링크가 두 노드들 예컨대 노드 i 및 노드 j 사이에 존재하는지를 나타내는 표시자(indicator)인 링크 연결 데이터(C_{ij})를 가질 수 있다. 이러한 링크 연결 데이터(간략히 연결 데이터)를 나타내면 아래의 수학적 식 1과 같다.

수학식 1

$$C_{ij} = \begin{cases} 1 & \text{if } i = j \text{ or there is a link between node } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

[0059]

[0060]

수학식 1은, 노드 i와 노드 j가 같거나 노드 i와 노드 j 간의 링크(link)가 존재하면, 연결 데이터(C_{ij})를 1로 설정하고, 그 이외(otherwise)에는 연결 데이터(C_{ij})를 0으로 설정하는 것을 나타낸다.

[0061]

위와 같은 네트워크에서 가상 네트워크 기능(VNF)을 살펴보기 위해, VNF 유형 그룹을 \mathcal{T} 라고 하고, 각 VNF가 서로 다른 VNF 유형(t)을 가진다고 가정하면, VNF를 위해 요구되는 CPU 코어들의 개수, 처리 용량(Processing Capacity), 처리 지연시간(Processing Delay) 및 설치 비용(Deployment Cost)은 VNF 유형에 따라 자동 결정된다. 따라서, 노드(s)에 설치된 VNF에 속하는 소정 유형 예컨대 t-타입의 VNF를 제1 VNF(V_{st})로 표현하면, 제1 VNF는 전용 데이터(Dedicated Data, D_{st})를 가질 수 있다.

[0062]

제1 VNF의 전용 데이터(D_{st})를 나타내면 다음의 수학식 2와 같다.

수학식 2

$$D_{st} = \begin{cases} (\mathcal{I}_{st}, \tau_t, \kappa_{st}) & \text{if } V_s \text{ has type } t, \\ 0 & \text{otherwise.} \end{cases}$$

[0063]

[0064]

위의 수학식 2에서, \mathcal{I}_{st} 는 VNF 인스턴스 개수(instance number), τ_t 는 t-타입 VNF의 최대 대역폭, κ_{st} 는 노드(s)에 설치된 t-타입 VNF의 현재 사용 대역폭을 각각 나타낸다.

[0065]

수학식 1 및 수학식 2에 의하면, 노드(s)에 설치된 VNF 데이터 D_s는 VNF 유형 그룹에 속한 특정 유형(t)에 따른 VNF 전용 데이터로 표현될 수 있다.

[0066]

본 실시예에 의하면, 가상 네트워크 관리 방법이나 가상 네트워크 관리 장치는 서비스 제약조건(service constraints) 및 물리적 네트워크(physical network)을 고려하여 최적 개수의 VNF 인스턴스(instances)에 대한 VNF 관리 정책을 최적의 서버에 실시간 배포하도록 구성될 수 있다.

[0067]

도 3은 도 2의 네트워크 토폴로지의 적어도 일부 노드에 적용할 수 있는 가상 네트워크 관리 장치에 대한 개략적인 블록도이다.

[0068]

도 3을 참조하면, 노드(300)는 적어도 하나의 프로세서(310), 메모리(320) 및 네트워크와 연결되어 통신을 수행하는 송수신 장치(330)를 포함할 수 있다. 또한, 노드(300)는 입력 인터페이스 장치(340), 출력 인터페이스 장치(350), 저장 장치(360) 등을 더 포함할 수 있다. 노드(300)에 포함된 각각의 구성요소들은 버스(bus, 370)에 의해 연결되어 서로 통신을 수행할 수 있다.

[0069]

다만, 노드(300)에 포함된 각각의 구성요소들은 공통 버스(370)가 아니라, 프로세서(310)를 중심으로 개별 인터페이스 또는 개별 버스를 통하여 연결될 수 있다. 예를 들어, 프로세서(310)는 메모리(320), 송수신 장치(330), 입력 인터페이스 장치(340), 출력 인터페이스 장치(350) 및 저장 장치(360) 중에서 적어도 하나와 전용 인터페이스를 통하여 연결될 수 있다.

[0070]

프로세서(310)는 메모리(320) 및 저장 장치(360) 중에서 적어도 하나에 저장된 프로그램 명령(program command)을 실행할 수 있다. 프로그램 명령은 도 1에 도시된 데이터 정의, 데이터 생성, 데이터 변환, GNN 처리, 제1 FNN 처리, 연결(concatenation), 제2 FNN 처리를 구현하는데 필요한 프로그램 명령을 포함할 수 있다. 이러한 프로그램 명령은, 특정 노드(300)에서 프로세서(310)에 의해 실행될 때, 해당 노드(300)에서 그래프 신경망을 이용하여 가상 네트워크를 관리하도록 구성될 수 있다. 이러한 프로세서(310)는 중앙 처리 장치(central

processing unit, CPU), 그래픽 처리 장치(graphics processing unit, GPU), 또는 본 발명의 실시예들에 따른 방법들이 수행되는 전용의 프로세서를 포함할 수 있다.

- [0071] 메모리(320) 및 저장 장치(360) 각각은 휘발성 저장 매체 및 비휘발성 저장 매체 중에서 적어도 하나로 구성될 수 있다. 예를 들어, 메모리(320)는 읽기 전용 메모리(read only memory, ROM) 및 랜덤 액세스 메모리(random access memory, RAM) 중에서 적어도 하나로 구성될 수 있다.
- [0072] 송수신 장치(330)는 3GPP(3rd generation partnership project) 표준에서 규정된 4G 통신 예컨대, LTE(long term evolution)나 LTE-A(advanced), 5G 통신 예컨대 NR(new radio) 등을 지원하는 서브통신시스템을 구비할 수 있다. 4G 통신은 6GHz 이하의 주파수 대역에서 수행될 수 있고, 5G 통신은 6GHz 이하의 주파수 대역뿐만 아니라 6GHz 이상의 주파수 대역에서 수행될 수 있다.
- [0073] 4G 통신 및 5G 통신을 위한 통신 프로토콜은, CDMA(code division multiple access), WCDMA(wideband CDMA), TDMA(time division multiple access), FDMA(frequency division multiple access), OFDM(orthogonal frequency division multiplexing), Filtered OFDM, CP(cyclic prefix)-OFDM, DFT-s-OFDM(discrete Fourier transform-spread-OFDM), OFDMA(orthogonal frequency division multiple access), SC(single carrier)-FDMA, NOMA(Non-orthogonal Multiple Access), GFDM(generalized frequency division multiplexing), FBMC(filter bank multi-carrier), UFMC(universal filtered multi-carrier), SDMA(Space Division Multiple Access) 기반의 통신 프로토콜 중 적어도 어느 하나 이상을 포함할 수 있다.
- [0074] 입력 인터페이스 장치(340)는 키보드, 마이크, 터치패드, 터치스크린 등의 입력 수단들에서 선택되는 적어도 하나와 적어도 하나의 입력 수단을 통해 입력되는 신호를 기저장된 명령과 매핑하거나 처리하는 입력 신호 처리부를 포함할 수 있다.
- [0075] 출력 인터페이스 장치(350)는 프로세서(310)의 제어에 따라 출력되는 신호를 기저장된 신호 형태나 레벨로 매핑하거나 처리하는 출력 신호 처리부와, 출력 신호 처리부의 신호에 따라 진동, 빛 등의 형태로 신호나 정보를 출력하는 적어도 하나의 출력 수단을 포함할 수 있다. 적어도 하나의 출력 수단은 스피커, 디스플레이 장치, 프린터, 광 출력 장치, 진동 출력 장치 등의 출력 수단들에서 선택되는 적어도 하나를 포함할 수 있다.
- [0076] 전술한 노드(300)는 사용자 장비(UE: user equipment), 터미널(terminal), 액세스 터미널(access terminal), 모바일 터미널(mobile terminal), 스테이션(station), 가입자 스테이션(subscriber station), 모바일 스테이션(mobile station), 휴대 가입자 스테이션(portable subscriber station), 디바이스(device), IoT(Internet of Thing) 장치, 탑재 장치(mounted module/device/terminal 또는 on board device/terminal) 등으로 지칭될 수 있다.
- [0077] 또한, 전술한 노드(300)는 통신 가능하고 VNF 배포가 가능한 퍼스널 컴퓨터(personal computer, PC), 데스크탑 컴퓨터(desktop computer), 랩탑 컴퓨터(laptop computer), 태블릿(tablet) PC, 무선전화기(wireless phone), 모바일폰(mobile phone), 스마트폰(smart phone), 스마트 워치(smart watch), 스마트 글래스(smart glass), e-book 리더기, PMP(Portable Multimedia Player), 휴대용 게임기, 네비게이션(navigation) 장치, 디지털 카메라(digital camera), DMB(digital multimedia broadcasting) 재생기, 디지털 음성 녹음기(digital audio recorder), 디지털 음성 재생기(digital audio player), 디지털 영상 녹화기(digital picture recorder), 디지털 영상 재생기(digital picture player), 디지털 동영상 녹화기(digital video recorder), 디지털 동영상 재생기(digital video player) 등의 사용자 장비 또는 사용자 단말을 포함할 수 있다.
- [0078] 또한, 본 실시예에 따른 노드(300)는 메모리(320), 저장 장치(360) 또는 컴퓨터로 읽을 수 있는 별도의 기록매체에 탑재되는 프로그램 형태의 그래프 신경망(GNN)을 포함한 기계 학습(Machine Learning, ML) 모델(도 1 참조)을 포함할 수 있다.
- [0079] ML은 자연어 처리 및 컴퓨터 비전과 같은 많은 연구 분야에서 유망한 기술이다. 이들 분야 대부분은 유클리드 영역 데이터를 사용한다. 데이터 학습에는 통상 CNN(Convolutional Neural Network)과 FNN(Feedforward Neural Network)이 사용된다. 한편, 화학, 생물학 또는 이와 유사한 분야에 CNN이나 FNN을 적용하는 경우, CNN과 FNN은 비유클리드 그래프 데이터를 입력받게 되므로 데이터를 학습하기 어렵다. 비유클리드 그래프 데이터는 인접 요소의 각 쌍 간의 풍부한 관계 정보와, 소셜 네트워크, 물리적 시스템 등의 다양한 유형의 그래프 구조 데이터를 나타내는 정보를 포함한다. 다시 말해서, CNN과 FNN이 비유클리드 데이터를 학습할 때, 그것들은 비유클리드 데이터의 그래프 구조를 무시하므로 데이터가 줄어들고 요소 간의 관계 정보가 많이 손실된다. 이에 본 실시예의 기계 학습 모델에서는 CNN의 일반화된 모델인 GNN 모델을 이용한다.

[0080] 본 실시예의 GNN은 그래프 데이터를 입력으로 하고, 그래프의 연결을 사용하고 가중치를 공유할 수 있다. 또한, GNN은 노드 및 에지를 포함한 그래프 정보를 표현하는 방법을 학습하는 그래프 임베딩에 의해 학습을 수행할 수 있다. 그래프 임베딩은 속성 그래프를 벡터 또는 벡터 세트로 변환하는 것을 나타낸다. 즉, 그래프 임베딩은 노드, 엣지 및 하위 그래프에 대한 그래프 정보를 캡처한다. 그래프 임베딩을 이용하면, 정보를 보존하면서 벡터 연산이나 빠르고 간단한 연산이 적용될 수 있는 평평한(flatten) 벡터 데이터로 그래프 데이터를 표현할 수 있다.

[0081] 다시 말해서, GNN은 노드들이나 네트워크 요소들 간의 관계 정보를 학습하고 그래프 임베딩과 같이 그래프 데이터를 벡터 데이터로 표현한다. 여기서, CNN은 완전 연결 신경망(FCNN)의 정규화된 버전이다. FCNN은 한 계층의 각 뉴런을 다음 계층의 모든 뉴런에 연결하여 과적합 문제를 만든다. CNN은 데이터에서 계층적 패턴을 사용하며 한 계층의 각 뉴런을 다음 계층의 로컬 영역에 있는 일부 뉴런에 연결한다. 이러한 연결을 통해 CNN은 가중치 매개변수를 공유하여 계산 비용을 줄일 수 있다. GNN은 전술한 CNN 또는 FNN의 특성을 취하고 그래프의 연결을 사용하여 가중치를 공유한다.

[0082] 이러한 GNN의 목적은 상태 임베딩(state embedding)을 학습하고 출력을 얻는 것이다(도 1의 S40 참조). x 와 h 를 각각 입력 특징(input features)과 은닉 상태(hidden states)라고 하고, $co[v]$ 가 서비스 또는 서비스 데이터(v)에 연결된 에지들의 그룹(the set of edges)이고, $ne[v]$ 가 노드 v 의 이웃 노드들의 그룹이라고 할 때, 노드 v 의 상태 임베딩 h_v 및 출력 o_v 을 다음의 수학적 식 3과 같이 정의할 수 있다.

수학적 식 3

$$h_v = f(x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}),$$

$$o_v = g(h_v, x_v),$$

[0083]

[0084] 수학적 식 3에서, $x_v, x_{co[v]}, h_{ne[v]}, x_{ne[v]}$ 는 기재된 순서대로 각각 서비스 또는 서비스 데이터(v)의 특징(data), v 에 연결된 엣지의 특징, v 의 이웃 노드들의 상태, v 의 이웃 노드들의 특징이다. 또한 f 는 전이 함수(transition function)이고 g 는 출력 함수(output function)이다.

[0085] 바나흐(Banach) 고정점(fixed point, H)은 전역 전이 함수(global transition function, F)로 표현될 수 있고, 전역 전이 함수(F)는 모든 상태에 대한 축적 변수(stacking variable, H)와 그래프 특징(graph features)에 대한 축적 변수(stacking variable, X)에 의해 정의될 수 있다. 바나흐 고정점은 특정 공간 안이나 경계에서 빠진 점이 없는 완비 거리 공간(complete metric space) 위의 축약 사상(contraction mapping)으로 얻는 유일한 고정점으로 정의될 수 있다. 이와 유사하게, 출력(outputs, O)은 전역 출력 함수(global output function, G)로 표현될 수 있고, 전역 출력 함수(G)는 모든 상태에 대한 축적 변수(H)와 노드 특징(node features)에 대한 축적 변수(X_0)에 의해 정의될 수 있다. 이를 나타내면 아래의 수학적 식 4와 같다.

수학적 식 4

$$H = F(H, X),$$

$$O = G(H, X_D).$$

[0086]

[0087] 전술한 전역 전이 함수(F)와 전역 출력 함수(global output function, G)는 FNN, CNN, 순환신경망(RNN: Recurrent Neural Network) 등의 신경망일 수 있다. 즉, 본 실시예의 기계 학습 모델은 학습 프로세스가 완료될 때까지 바나흐 고정점을 업데이트할 수 있다.

[0088] 한편, 본 실시예에서 GNN 모델은, 이웃 노드들의 정보를 사용하여 노드에 대한 새로운 상태를 생성할 때, 가중치합 방법(weighted sum method)을 사용할 수 있다. 이 경우, 가중치합 방법은 노드의 순서와 크기를 고려할 필요가 없지만 구조적 정보를 평평하게(smoothing) 하는 단점이 있으므로, 이러한 단점을 극복하기 위해 엣지 필터(edge filters)를 생성하는 엣지 조건의 필터링된 그래프(Edge-conditioned filtered graph) CNN을 이용할 수 있다. 이러한 엣지 필터를 생성하는 필터 생성 함수는 FNN이고, 이 함수는 이웃 노드들의 특징 그룹들

(feature sets)로 곱하는 필터 행렬을 생성할 수 있다. 곱한 값을 평균화하면 새로운 노드 레이블을 생성할 수 있다. 전술한 필터 발생 함수가 소정의 가중치 파라미터로 파라미터화 된다고 가정하면, 임의의 1번째의 전이함수에 의한 상태는 바이어스 파라미터(bias parameter)와의 합으로 정의될 수 있다. 1번째의 전이함수에 의한 상태를 얻은 후, 출력 전이 함수를 이용하여 출력을 얻을 수 있다.

[0089] 또 한편으로, 본 실시예의 기계 학습 모델에서는 GNN을 사용하고 네트워크 데이터를 그래프로 취급한다. 그래프는 연결 정보를 가지고 있으며 네트워크 구조 데이터를 제공할 수 있다. 기존 기술과 비교할 때, 본 실시예의 VNF 관리 문제는 해결하기가 훨씬 더 어렵지만, 최적의 VNF 인스턴스 수를 도출하는 대신 모든 노드 각각과 모든 VNF 유형에 대한 인스턴스 수와 위치를 결정하기 위해 GNN 기반 가상 네트워크 기능 관리 솔루션을 사용한다.

[0090] 특히, 기계학습이나 인공지능을 이용하여 VNF 관리를 일반화하기 위하여 GNN은 모델이 네트워크 구조를 고려할 수 있도록 그래프 데이터를 명시적으로 사용할 수 있다. 그러한 그래프 데이터의 명시적 사용은, 단일 네트워크 토폴로지 형태로만 학습하여 다른 네트워크에 기계 학습 모델을 적용하기 어려운 경우나 네트워크 장애로 인해 대상 네트워크 토폴로지가 변경된 경우에 기계 학습 모델을 적용할 수 없는 경우와 달리, 네트워크 도메인에서 GNN의 출력에 대한 신뢰성을 안정적으로 확보할 수 있음을 나타낸다.

[0091] 한편, 가상 네트워크 관리 장치는 서로 다른 다수의 서비스 요청(Service Requests)을 사용자로부터 혹은 사용자 인터페이스를 통해 받을 수 있다.

[0092] 이때, 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드 정보를 가지고 있으며 해당 서비스의 타입과 함께, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 요구 대역폭 정보를 가지고 있다고 가정한다. 그 경우, 서비스는 사용자 단말로부터의 서비스 요구(Service Request)에 따라 다양하게 표현되며, 일련의 VNF 열로 표현될 수 있다.

[0093] 본 실시예의 가상 네트워크 관리 방법에 채용할 수 있는 서비스 데이터의 상세한 정의를 나타내면 수학적 3과 같이 표현될 수 있다.

수학적 5

[0094]
$$\mathbf{v} = (\mathbf{w}_v, \mathbf{u}_v, d_v, \phi_v, p_v, \beta_v, \gamma_v)$$

[0095] 수학적 5의 파라미터와 그 설명은 다음과 같다.

- v**: 서비스 또는 서비스 데이터
- $\mathbf{v} \in \psi$, **ψ**: 서비스 집합
- \mathbf{w}_v : v 서비스 시작 노드
- \mathbf{u}_v : v 서비스 도착 노드
- d_v : v 서비스 실행 시간
- ϕ_v : v 서비스 종류
- p_v : v 서비스 딜레이에 따른 패널티 비용
- β_v : v 서비스 요구 대역폭
- γ_v : v 서비스 딜레이 조건

[0097] 전술한 VNF는 방화벽(Firewall), 침입탐지시스템(IDS: Intrusion Detecion Services), 네트워크주소변환(NAT: Network Address Translation) 기능과 같이 여러 종류가 있으며, 각 종류별로 요구하는 CPU 코어(Core)의 수와 설치 비용, 최대 허용 대역폭은 서로 다르다.

[0098] 또한 설치된 VNF는 각각 트래픽을 받음으로 서로 현재 사용되는 대역폭 크기가 다르다. VNF는 위에서 언급한 정보 외에도 추가적으로 VNF를 정의할 수 있다. VNF 정의 이후에는 유저가 네트워크에 요구하는 서비스와 서비스가 가지고 있는 정보를 정의할 수 있다.

[0099] 본 실시예에서 네트워크 데이터는 실제 환경 혹은 시뮬레이션 환경에서 수집될 수 있으며, 다양한 환경의 네트

워크 정보를 데이터로 수집한다. 그리고, 서비스 정보가 테스트베드에서 생성되거나 혹은 실제 환경에서 수집될 때마다 서비스 정보 리스트를 생성할 수 있다.

[0100] 도 4는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 네트워크의 정규화된 트래픽 패턴에 대한 예시도이다.

[0101] 도 4는 실제 네트워크에서 생성된 1주(week)의 인터넷2(Internet2)의 트래픽 패턴을 보여준다. 이러한 트래픽 패턴을 반복하여 4주의 트래픽 패턴을 생성할 수 있다. 본 실시예에서는 트래픽 패턴을 사용하여 서비스 요청(service request) 및 서비스 요청에 따른 요청 목록(request lists)을 생성할 수 있다. 요청 목록은 서비스 정보 리스트에 대응된다.

[0102] 요청 목록은 정보 리스트는 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내며, 생성된 서비스들의 시작 시간과 실행 시간을 이용하여 수학적 식 6과 같이 표현될 수 있다.

수학적 식 6

[0103]
$$\pi_{\mu+1} = \{\nu_{\text{new}}\} \cup \pi_{\mu} \cap \{x : d_{\nu_x} > \hat{a}_{\nu_{\text{new}}} - \hat{a}_{\nu_x}\}$$

[0104] 수학적 식 6에서, $\pi_{\mu+1}$ 는 현재 서비스 리스트 정보 아이디($\mu+1$)에 대한 현재 서비스 정보 리스트, ν_{new} 는 가장 최근에 생성된 서비스, μ 는 서비스 리스트 정보 아이디 또는 시간상으로 현재보다 바로 앞선 직전의 서비스 리스트 정보 아이디, π_{μ} 는 직전 서비스 리스트 정보 아이디에 대한 직전 서비스 정보 리스트, \hat{a} 는 서비스 생성 시간, d_{ν_x} 는 임의 시간(x)에 생성된 서비스의 서비스 실행 시간, $\hat{a}_{\nu_{\text{new}}}$ 는 가장 최근에 생성된 서비스의 서비스 생성 시간, \hat{a}_{ν_x} 는 임의 시간에 생성된 서비스의 서비스 생성 시간을 각각 나타낸다.

[0105] 전술한 요청 목록은 새로운 서비스 요청이 생성될 때마다 생성될 수 있다. 또한, 요청 목록에는 서비스 시간(d)이 아직 만료되지 않은 여러 서비스 요청이 포함되어 있을 수 있다. 이러한 요청 목록은 여러 서비스가 동시에 제공되는 실제 네트워크 상황을 가정하기 위한 것이다.

표 1

[0106]

Service ID	Service type	Proportion
1	NAT - Firewall - IDS	0.3
2	NAT - Proxy	0.4
3	NAT - WANO	0.3

[0107] 표 1에 나타낸 바와 같이, 서비스 식별자(ID)(1, 2, 3)로 식별되는 서비스 요청에 대한 서비스 유형(service type)의 비율(proportion, P)에 따라 분당 5개의 요청을 생성할 수 있다. 또한 도 4의 정규화된 트래픽 볼륨(Normalized Traffic Volume)으로 나타낸 바와 같이, 분당 5개의 요청을 버려 트래픽 패턴을 추정하도록 할 수 있다.

[0108] 실제 네트워크 환경에서는 특정 시점에 더 많은 서비스를 요청할 수 있지만 간단한 구현을 위해 요청 목록에 최대 4개의 요청이 포함된다고 가정한다. 요청 복잡도를 요청 목록에 포함된 요청 개수인 $n(\pi)$ 로 정의한다. 요청 복잡도의 비율을 $\pi | (n(\pi)=1):0.14$, $\pi | (n(\pi)=2):0.33$, $\pi | (n(\pi)=3):0.36$ 및 $\pi | (n(\pi)=4):0.17$, 와 같이 만들도록 서비스 시간(d)를 정의한다.

[0109] 데이터 요청 v 에 대해 두 개의 대역폭 범위 β_v 를 설정한다. 하나의 대역폭 범위는 33~38Mbps이고 다른 대역폭 범위는 330~380Mbps 이다. 이러한 대역폭은 VNF의 용량을 기반으로 다중 클래스 학습(multiclass learning)을 위해 설정된다. 대역폭 범위가 33~38Mbps인 경우 최대 1개의 VNF 인스턴스를 배포할 수 있다. 33~38Mbps의 대역폭을 가진 데이터 세트를 기본 요청 데이터 세트라고 한다. 대역폭 범위가 330~380Mbps인 경우 최대 4개의 VNF 인스턴스를 배포할 수 있다. 330~380Mbps 사이의 대역폭을 가진 데이터 세트를 보조 요청 데이터 세트라고 한다.

[0110] SLA(Service Level Agreement)를 벗어나는 서비스 딜레이 조건으로서 최대 지연 범위(maximum latency range)는 700~750ms 사이에서 임의로 설정될 수 있다. 서비스 요청을 위한 인그레스(ingress) 서버 및 이그레스(egress) 서버는 무작위로 선택되며 동일한 서버가 될 수 없다. SLA 위반의 패널티 비용을 0.1로 설정한 수 있다. 서비스 유형은 여러 VNF로 구성되고, 기존의 VNF 카탈로그 데이터를 사용할 수 있다. VNF 유형을 나타내면 다음의 표 2와 같다.

표 2

Network function	CPU required	Processing capacity	Processing delay
Firewall	2	900 Mbps	45 ms
Proxy	2	900 Mbps	40 ms
IDS	4	900 Mbps	1 ms
NAT	1	900 Mbps	10 ms
WANO	2	900 Mbps	5 ms

[0112] 표 2에서 IDS는 침입 탐지 시스템(intrusion detection system)을, NAT는 네트워크 주소 번역(Network Address Translation)을, WANO는 광역 네트워크 옵티마이저(Wide Area Network Optimizer)를 각각 나타낸다.

[0113] 도 5는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 네트워크 데이터를 설명하기 위한 예시도이다.

[0114] 도 5의 Internet2 네트워크는, AT&T IP 네트워크와 네트워크 혁신을 위한 글로벌 환경(GENI: global environment for networking innovation) 네트워크를 참조한 것이다. Internet2 네트워크 토폴로지(network topology)는 사각 박스 내에 노드 번호(node number) 0 내지 11로 각각 표시된 12개의 노드들과 이 노드들 사이를 연결하는 15개의 엣지로 구성된다. 모든 노드를 서버라고 가정한다. 각 노드를 표시하는 원통모양 주위에는 사용 가능한 CPU 코어(Available CPU Core)가 소정의 자연수(예컨대, 1, 2, 3, 5, 8, 10, 12, 16)로 표시되고, 각 엣지 주변에는 최대 대역폭 및 지연시간에 대한 정보가 각각 표시되어 있다. 전송한 서버 사양과 에너지 소비 및 패킷 전환 비용(Transition cost)을 예시하면 표 3과 같다.

표 3

CPU cores	Energy consumption		Cost information	
	Idle energy	Peak energy	Energy cost	Transition
16	80.5W	2735 W	0.1	3.62×10^{-7} per bit

[0116] 이와 같이, 본 실시예에 따른 가상 네트워크 관리 방법은 새로운 요청 목록이 생성될 때마다 현재 네트워크 데이터를 수집할 수 있다. 네트워크 데이터는 서버 데이터(D), 링크 데이터(L) 및 연결 데이터(C)를 포함한다. 그리고, 물리 네트워크와 VNF 및 서비스 정보가 모두 수집되면, 가상 네트워크 관리 방법은 선형계획법(ILP: Integer Linear Programming) 혹은 기타 방법을 이용하여 기계학습에 정답 데이터로 활용할 데이터를 생성할 수 있다. 그런 다음 생성된 데이터를 요청 목록과 연결하여 학습함으로써, 현재 네트워크에 대한 최적의 VNF 정책을 얻을 수 있다. 최적의 VNF 정책을 찾으면 현재 네트워크 구성을 최적의 네트워크 구성으로 변경하며, 이러한 과정은 생성된 모든 데이터가 처리될 때까지 반복될 수 있다.

[0117] 본 실시예의 가상 네트워크 관리 방법에서는 최적의 VNF 정책을 얻기 위해 특정 ILP 방정식을 사용할 수 있다. 특정 ILP는 VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용 등과 같이 네트워크 관리 전반에 드는 여러 비용을 포함할 수 있다.

[0118] 먼저, 에너지 비용(\mathbb{E})에 대한 ILP를 나타내면 수학적 식 7과 같다.

수학적 식 7

$$\mathbb{E} = \sum_{n \in N} \sum_{t \in T} \mathcal{I}_{st} (e_{ie} + (e_{pk} - e_{ie}) \frac{\mathcal{E}_t}{C_{spec}}) \lambda_{energy}$$

[0120] 수학적식 7에서 주요 파라미터에 대한 설명은 다음과 같다.

e_{ie} : idle 에너지 비용

e_{pk} : peak 에너지 비용

C_{spec} : CPU Cores 수

[0121] **λ_{energy}** : 에너지 소모 비용

[0122] 또한, 트래픽 포워딩 비용(**ℤ**)에 대한 ILP를 나타내면 수학적식 8과 같다.

수학적식 8

$$\mathcal{W}_{s_1 s_2}^{\nu t_1 t_2} = \begin{cases} 1 & \text{if } \mathcal{V}_{s_1 t_1} \text{ has traffic between } \mathcal{V}_{s_2 t_2} \text{ for } \nu, \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathcal{J}_{s_1 s_2}^{\mu t_1 t_2} = \sum_{\nu \in \pi_{\mu t_1, t_2} \in \mathcal{T}} \mathcal{W}_{s_1 s_2}^{\nu t_1 t_2} \beta_{\nu},$$

$$\mathbb{T} = \sum_{s_1 \in N} \sum_{\substack{s_2 \in \eta(s_1), \\ s_2 < s_1}} (\mathcal{J}_{s_1 s_2}^{\mu t_1 t_2} - \mathcal{J}_{s_1 s_2}^{(\mu-1)t_1 t_2}) \lambda_{transit}.$$

[0123]

[0124] 수학적식 8에서, **ϑ(s₁)**는 특정 노드(s₁)의 이웃 노드, **λ_{transit}**은 패킷 포워딩 비용을 각각 나타낸다. 수학적식 8에서 알 수 있듯이, 서비스 ν에 대해서 제1 VNF 유형(t₁)을 배포하는 제1 노드(s₁)의 서비스 $\mathcal{V}_{s_1 t_1}$ 와 제2 VNF 유형(t₂)을 배포하는 제2 노드(s₁)의 서비스 $\mathcal{V}_{s_2 t_2}$ 간의 트래픽(traffic)이 존재할 때, 트래픽 포워딩 비용은 1로 설정되고, 그 외의 경우에는 0으로 설정될 수 있다.

[0125] 또한, SPA 위반 비용 또는 서비스 지연 비용(**ℒ**)에 대한 ILP를 나타내면 수학적식 9와 같다.

수학적식 9

$$\mathbb{S} = \sum_{\nu \in \pi_{\mu}} \max \left(\begin{array}{l} \sum_{t \in \hat{\phi}_{\nu}} \delta_t + \sum_{t_1 t_2 \in \mathcal{T}} \sum_{\substack{s_2 \in \eta(s_1), \\ s_1 \in N \quad s_2 < s_1}} \mathcal{W}_{s_1 s_2}^{\nu t_1 t_2} d_{s_1 s_2} - \gamma_{\nu}, 0 \end{array} \right) p_{\nu}$$

[0126]

[0127] 수학적식 9에서, $\hat{\phi}_{\nu}$ 는 서비스 지연(ϕ_{ν})에 포함되는 VNFs를 나타낸다.

[0128] SPA 위반 비용 또는 서비스 지연 비용은 해당 비용을 산출하기 위해 고려되는 현재 시점에서 사용자 단말 등의 노드에서 획득되는 서비스에 기초한 네트워크에서 고려하고자 하는 전체 서비스들의 집합에 해당한다. 이러한 서비스 지연 비용은 기존의 네트워크 서비스 지연 비용을 산출하는 방법 중 하나이며, 기타 다른 네트워크 서비스 지연 비용을 산출하는 방법이 사용될 수도 있다.

[0129] 또한, 리소스 분할 비용(**ℱ**)에 대한 ILP를 나타내면 수학적식 10과 같다.

수학식 10

$$\mathbb{F} = \sum_{s \in N, V_s \neq 0} \left(C_{\text{spec}} - \sum_{t \in T} \mathcal{I}_{st} \mathcal{E}_{st} \right) \lambda_{\text{core}} + \sum_{s_1 \in N} \sum_{s_2 \in \eta(s_1)} \frac{\max(\mathcal{J}_{s_1 s_2}^{\mu t_1 t_2}, 0)}{\mathcal{J}_{s_1 s_2}^{\mu t_1 t_2}} (m_{s_1 s_2} - \mathcal{J}_{s_1 s_2}^{\mu t_1 t_2}) \lambda_{\text{band}}$$

[0130]

[0131]

[0132]

[0133]

수학식 10에서, λ_{core} 는 CPU 코어 개별 비용을, λ_{band} 는 개별 대역폭 사용 비용을 각각 나타낸다.

리소스 분할 비용은 해당 비용을 산출하기 위해 고려되는 현재 시점에서 네트워크의 노드들에서 획득되는 CPU 코어 개별 비용의 총합과 개별 대역폭 사용 비용의 총합을 더한 값을 산출될 수 있으나, 이에 한정되지 않고, 기존에 사용되는 기타 다른 리소스 분할 비용을 산출하는 방법이 사용될 수도 있다.

에너지 비용(E), 트래픽 포워딩 비용(T), 서비스 지연 비용(S) 및 리소스 분할 비용(F)을 포함한 전체 ILP 목적식을 나타내면 다음의 수학식 11과 같다.

수학식 11

$$\hat{a}E + \hat{b}T + \hat{c}S + \hat{d}F^{10}$$

[0134]

[0135]

[0136]

[0137]

[0138]

[0139]

[0140]

[0141]

[0142]

수학식 11에서, \hat{a} , \hat{b} , \hat{c} , \hat{d} 는 가중치 팩터(weighting factor)를 각각 나타낸다.

본 실시예의 가상 네트워크 관리 방법은 수학식 11의 ILP 목적식의 값을 최소화(minimizing)하도록 기계 학습 모델을 학습시킬 수 있다.

한편, 위의 수학식 7 내지 수학식 10과 같이 표현된 ILP 외의 다른 ILP 수식이나 휴리스틱 알고리즘을 적용하여 정답 데이터를 생성할 수 있다. 만약 정답 데이터 셋을 이미 보유하고 있다면, 그 정답 데이터를 활용할 수도 있다.

도 6a 내지 도 6c는 도 3의 가상 네트워크 관리 장치에 채용할 수 있는 라벨 데이터 생성 과정을 설명하기 위한 도면들이다.

도 6a는 현재 네트워크의 서버들(servers)과 VNF 유형들(VNF types)을 표현하는 행렬의 각 성분 값을 현재 VNF 인스턴스 개수(Current VNF instance number)로 나타낸 것이다. VNF 유형에서 F는 방화벽(Firewall)을, P는 프록시(Proxy)를, I는 IDS(Intrusion Detection Systems)를, N은 NAT(Network Address Translation)를, W는 WAN(Wide Area Network Optimization)를 각각 나타낸다.

도 6b는 에너지 비용, 트래픽 포워딩 비용, 서비스 지연 비용 및 리소스 분할 비용을 포함한 네트워크 관리 전반에 드는 여러 비용에 기반한 산출한 IPL 솔루션(Solution)의 행렬을 나타낸다. IPL 솔루션 행렬의 각 성분 값은 최적 VNF 인스턴스 개수(Optimal VNF instance number)로 표시되어 있다. 도 6b에서 음영으로 표시된 부분은 도 6a의 대응 부분과 달라진 부분으로 최적 VNF 관리 정책을 적용할 위치(서버)에 해당한다.

도 6c는 도 6a의 현재 네트워크와 도 6b의 IPL 솔루션의 대응 성분 값들을 비교하여 현재 네트워크의 성분 값이 IPL 솔루션의 대응 성분 값보다 크면 VNF 관리 정책(VNF management policy) 중 제거(remove)를 적용하고, 현재 네트워크의 성분 값이 IPL 솔루션의 대응 성분 값보다 크면 그대로 두기(Leave it as it is: None)를 적용하고, 현재 네트워크의 성분 값이 IPL 솔루션의 대응 성분 값보다 작으면 추가(Add)를 적용한 결과를 보여준다.

본 실시예에 의하면, 기계 학습(ML)을 위한 학습 데이터 셋을 생성하는데 있어서, 새로운 서비스 요청이 발생할 때마다 네트워크 데이터(D, L, C)와 서비스 요청(service request)로부터 특징 데이터(feature data)를 생성할 수 있다. 동시에, 도 6a 내지 도 6c를 참조하여 설명한 바와 같이 현재 설치된 VNF과 ILP 기반 VNF 설치 솔루션

간의 차이를 분류하여 라벨 데이터(label data)를 생성할 수 있다.

- [0143] 학습 데이터는 숫자 데이터와 범주형 데이터로 구성될 수 있다. 모든 숫자 데이터를 정규화하고 범주 데이터에 대해 하나의 핫 인코딩(one-hot-encoding)을 적용할 수 있다. 본 실시예에서 네트워크 데이터는 그래프로 표현되어야 한다. 따라서, 크기가 노드 번호(node number) × 노드 특징 번호(node feature number)인 행렬로 노드 데이터(D)를 변환할 수 있다. 이와 유사하게 연결 데이터(C)와 링크 데이터(L)도 노드 번호 × 노드 번호 크기의 행렬로 변환될 수 있다.
- [0144] 또한, 각 데이터 세트는 도 6a 내지 도 6c와 같이 60개의 정책을 VNF 관리 정책의 3가지 클래스로 분류하는 데 사용되기 때문에 마지막 FNN 레이어는 180개의 출력을 가질 수 있다. 그리고 각 정책에 대해 소프트맥스(softmax)를 적용하고 교차 엔트로피를 하위 손실 함수로 사용할 수 있다. 목적 손실 함수는 하위 손실 함수에 클래스 가중치를 곱한 합으로 표현될 수 있다.
- [0145] 클래스 가중치는 데이터의 각 클래스 비율의 역수일 수 있다. VNF 정책은 일반적으로 불균형이며 이러한 불균형은 학습 성능을 저하시킬 수 있다. 따라서 레이블 클래스를 따르는 클래스 가중치와 하위 손실 함수를 곱하여 이를 보완하는 것이 가능하다.
- [0146] GNN 학습을 통해 생성한 레이블 임베딩을 포함하는 각 노드에 대한 상태 정보를 서비스 리스트(Service List) 데이터와 같이 FNN을 통해 학습하며, 학습 과정에서 데이터 클래스 불균형을 맞추기 위해 가중치 균형(Wieght Balance) 기법을 활용할 수 있다.
- [0147] 도 7은 본 발명의 또 다른 실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법을 위한 기계 학습 모델에 대한 개략적인 구성도이다. 도 8은 도 7의 기계 학습 모델을 이용하는 가상 네트워크 관리 장치의 소프트웨어 모듈들에 대한 개략적인 블록도이다. 도 9는 도 8의 소프트웨어 모듈들 중 FNN 학습부에 채용할 수 있는 구성을 나타낸 블록도이다.
- [0148] 도 7을 참조하면, 본 실시예에 따른 그래프 신경망 기반 가상 네트워크 관리 방법은, 프로세서와 메모리를 구비한 컴퓨팅 장치에 의해 수행되는 방법으로서, 네트워크 정보와 트래픽 정보를 포함한 네트워크 데이터(Network Data, S10)를 그래프 형태로 변환한(Graph conversion, S30) 후 GNN을 이용하여 학습하고(S40), 서비스 요청(Service Reqeust, S15)을 FNN(S20)을 이용하여 학습하고, 이후 두 개의 학습된 데이터를 합쳐(concatenation, S50) 다시 FNN(S70)으로 학습할 수 있다. 이때 네트워크 정보와 트래픽 정보는 각각 평평한(Flatten) 형태를 가지는 정보로 표현이 되며, 연결 레이어(Concatenate Layer)을 이용하여 하나의 평평한(Flatten) 형태의 정보로 표현이 된다. 평평한 형태의 정보는 3차원 데이터를 2차원이나 1차원 데이터로 평평하게 늘어뜨리거나 2차원 데이터를 1차원 데이터로 평평하게 늘어뜨리는 것을 가리킬 수 있다. FNN(S70)은 제2 FNN(FNN2, S72), 제3 FNN(FNN3, S73), 완전 연결 레이어(Fully Connected Layer, FCL, S74) 및 소프트맥스(Softmax, S75)를 구비할 수 있다. 이렇게 표현된 데이터는 FNN 레이어를 이용하여 학습되며 최종적으로 네트워크 정보와 트래픽 정보를 반영하여 모든 노드들에 대해 최적에 가까운 VNF 관리 결정을 내릴 수 있다. VNF 관리 결정은 각 노드의 생성, 제거, 그대로 두기 등을 포함할 수 있다.
- [0149] 진술한 가상 네트워크 관리 방법을 구현하기 위해, 가상 네트워크 관리 장치의 프로세서(310)는 도 8에 도시한 바와 같이 적어도 그 작동 상태에서 탑재되는 데이터 정의부(311), 데이터 수집부(312), 데이터 생성부(313), 변환부(314), GNN 학습부(315), 제1 FNN 학습부(316), 연결부(317), 제2 FNN 학습부(319)를 구비할 수 있다.
- [0150] 여기서, 데이터 정의부(311), 데이터 수집부(312) 및 데이터 생성부(313)는 데이터 전처리부(310a) 또는 전처리부로 지칭될 수 있고, 변환부(314), GNN 학습부(315), 제1 FNN 학습부(316), 연결부(317) 및 제2 FNN 학습부(319)는 앙상블 모델 학습부(310b)로 지칭될 수 있다. 또한, 구현에 따라서 데이터 정의부(311), 데이터 수집부(312), 데이터 생성부(313), 변환부(314), GNN 학습부(315), 제1 FNN 학습부(316) 및 연결부(317)는 앙상블 모델 생성부(318)로 지칭될 수 있다. 그리고, GNN 학습부(315)는 제1 학습부로, 제1 FNN 학습부(316)는 전처리 FNN 학습부로, 연결부(317)는 조합 데이터 생성부로, 제2 FNN 학습부(319)는 메인 FNN 학습부 또는 제2 학습부로, 제2 FNN 학습부(319)의 FCL(S74)과 소프트맥스(S75)의 기능을 수행하는 출력부는 VNF 관리 정책 출력부 또는 VNF 관리 결정 출력부로 각각 지칭될 수 있다.
- [0151] 가상 네트워크 관리 장치는 데이터 정의부(311)를 통해 기계학습에 사용할 데이터를 정의하고, 데이터의 정의에 기초하여 데이터 생성부(313)를 통해 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스 정보 리스트를 생성하고, 서비스 정보 리스트가 생성될 때 데이터 수집부(312)를 통해 물리 네트워크 정보와 가상 네트워크 기능(VNF) 정보를 수집하고, 수집된 네트워크 데이터의 물리 네트워크 정보, VNF 정보 및

서비스 정보를 토대로 데이터 생성부(313)를 통해 기계학습에 활용할 정답 데이터를 준비할 수 있다. 이때, 가상 네트워크 관리 장치(300)는 사용자로부터 서로 다른 복수의 서비스 요청(Service Requests)을 받을 수 있다.

- [0152] 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드 정보를 가지고 있으며 해당 서비스의 타입과 함께, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 요구 대역폭 정보를 가지고 있고 가정한다. 그 경우, 서비스는 사용자 단말로부터의 서비스 요구(Service Request)에 따라 다양하게 표현되며, 일련의 VNF 열로 표현될 수 있다.
- [0153] 가상 네트워크 관리 장치는 학습에 사용될 모든 데이터를 수집 및 생성한 후, 전처리 과정을 거치도록 구성될 수 있다. 일례로, 앙상블 모델 생성부(318)를 통해 모든 수치 데이터들을 정규화하여 표현하고, 카테고리별 데이터들을 원-핫 인코딩(one-hot-encoding)으로 표현할 수 있다.
- [0154] 또한, 가상 네트워크 관리 장치(300)는 GNN을 사용하여 네트워크 데이터를 학습할 것(S40)을 고려하여 변환부(314)를 통해 노드와 엣지 데이터들을 그래프 데이터로 변환하고(S30) 이를 각각 노드 정보와 엣지 정보, 연결 정보를 가지는 노드 행렬, 엣지 행렬, 연결 행렬로 표현할 수 있다.
- [0155] 노드행렬은 노드 수 \times 노드 피처의 행렬이며, 엣지 행렬은 노드 수 \times 노드 수 \times 엣지 피처 행렬일 수 있다. 또한, 연결 행렬은 노드수 \times 노드수 행렬로 노드간의 링크가 존재할 시에는 1, 그렇지 않을 시에는 0의 값을 가질 수 있다. 네트워크 데이터의 그래프 변환은 사용하는 GNN의 구조에 따라 다르게 표현될 수 있다.
- [0156] 또한, 가상 네트워크 관리 장치(300)는 행렬로 표현되는 네트워크 데이터를 GNN 학습부(315)를 통해 학습하고 각 노드의 상태 정보를 생성하며, 생성된 노드 상태 정보를 가중치 벡터가 적용된 서비스 리스트 데이터와 함께 제2 FNN 학습부(319)를 통해 학습할 수 있다. 제2 FNN 학습부(319)에 입력되기 전에 노드 상태 정보와 서비스 리스트 데이터는 연결부(317) 또는 앙상블 모델 생성부(318)의 연결 함수(Concatenation Function)를 통해 하나로 합쳐질 수 있다.
- [0157] 이와 같이 본 실시예의 기계학습 모델은, 도 7 내지 도 9에 나타낸 바와 같이, 네트워크 데이터(Network Data, S10)로부터 사전에 정의된 서버(Server) 데이터 또는 노드 데이터, 링크(Link) 데이터 및 연결(Connection) 데이터를 변환부(315)를 통해 그래프 형식으로 변환하고(Graph Conversion, S30) GNN 학습부(316)를 사용해 GNN 학습하여(S40) 각 노드들에 대한 상태 정보(노드 상태 정보)를 생성하고, 이와 병렬적으로 제1 요청(Request 1), 제2 요청(Request 2) 및 제3 요청(Request 3)을 포함한 복수의 서비스 요청(Service Request, S15)을 각 서비스 요청별로 각 FNN(FNN1, S20)에 입력하고 복수의 FNN의 가중치 공유(Weight Sharing, S25)를 통해 서비스 리스트(Service List)를 생성할 수 있다. 그런 다음, 연결부(318)를 통해 앞서 생성된 노드 상태 정보를 서비스 리스트 데이터와 연결한 후, 제2 FNN 학습부(319)를 통해 노드 상태 정보와 서비스 리스트를 같이 학습하고 학습 결과를 출력할 수 있다. 이러한 기계 학습은 기설정 목표치에 도달하거나 기설정 반복 횟수에 도달할 때까지 반복 수행될 수 있으며, 학습 과정에서 데이터 클래스 불균형을 맞추기 위해 가중치 균형(Weight Balance) 기법을 활용할 수 있다.
- [0158] 한편, 전술한 실시예에서, 기계학습 모델을 학습시킬 때, 서비스 리스트 데이터 등의 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 가중 균형 (Weight balance) 기법을 적용할 수 있다. 예컨대, 특정 클래스의 데이터 수가 적을 경우, 기계학습 모델이 해당 클래스를 학습하는 데 있어서 가중치를 높여 주면 기계학습 모델이 더욱 적합하게 학습하도록 할 수 있다.
- [0159] 이러한 가중 균형 기법은 도 9에 도시한 바와 같이 제1 FNN 학습부(316)의 가중치 조정부(31bb)에 의해 수행될 수 있다. 제1 FNN 학습부(316)는 도 7의 FNN1(S20)의 기능을 수행하는 FNN(316a)만을 구비하거나, 가중치 조정부(316b)를 더 구비하도록 구성될 수 있다.
- [0160] 본 실시예에 의하면, 기계학습 모델은 네트워크 데이터(Network Data, S10)로부터 사전에 정의된 서버(Server) 데이터 또는 노드 데이터, 링크(Link) 데이터 및 연결(Connection) 데이터를 변환부(315)를 통해 그래프 형식으로 변환하고(Graph Conversion, S30) GNN 학습부(316)를 사용해 GNN 학습하여(S40) 각 노드들에 대한 상태 정보(노드 상태 정보)를 생성하고, 이와 병렬적으로 제1 요청(Request 1), 제2 요청(Request 2) 및 제3 요청(Request 3)을 포함한 복수의 서비스 요청(Service Request, S15)을 각 서비스 요청별로 각 FNN(FNN1, S20)에 입력하고 복수의 FNN의 가중치 공유(Weight Sharing, S25)를 통해 서비스 리스트(Service List)를 생성할 수 있다. 그런 다음, 연결부(317)를 통해 앞서 생성된 노드 상태 정보를 서비스 리스트 데이터와 연결한 후, 제2 FNN 학습부(319)를 통해 노드 상태 정보와 서비스 리스트를 같이 학습하고 학습 결과를 출력할 수 있다. 이러한 기계 학습은 기설정 목표치에 도달하거나 기설정 반복 횟수에 도달할 때까지 반복 수행될 수 있으며, 학습 과정

에서 데이터 클래스 불균형을 맞추기 위해 가중치 균형(Wieght Balance) 기법을 활용할 수 있다.

[0161] 아래의 수학식 12는 모델이 사용한 손실 함수(Loss Function)으로 크로스 엔트로피(Cross-Entropy) 함수에 가중치 균형 기법을 적용한 것을 표현한 것이다.

수학식 12

[0162] Loss function: $\sum_{n \in N} \sum_{t \in T} \sum_i^3 l_i^{st} \log(o_i^{st}) \frac{A_1^{st} + A_2^{st} + A_3^{st}}{A_i^{st}}$

[0163] 수학식 12에서 i^{st} 는 정답 데이터(true data), o_i^{st} 는 출력 데이터(output data), A_i^{st} 는 정책(i), 노드(s), VNF 유형이나 종류가 t일 경우의 클래스 개수를 각각 나타낸다.

[0164] 한편 모델에서 사용하는 손실함수는 사용자의 목적에 따라 변경될 수 있다. 예컨대, VNF 정책을 학습할 때, VNF의 종류를 고려하지 않을 수 있으며 설치하는 서버의 위치만을 고려하도록 변경할 수 있다. 또한, [수학식 12]에 추가적인 식을 더하는 것으로 학습 모델이 특정 정보에 민감하게 반응하도록 학습할 수 있다.

수학식 13

[0165] $\sum_{s \in N} | \sum_{t \in T} \sum_i i(l_i^{st} - o_i^{st}) |$

[0166] 수학식 13은 수학식 12에 더하여 사용될 수 있다. 그러한 경우, 학습 모델은 VNF의 종류에 대한 정보를 적게 고려하며 설치하는 서버의 위치 및 설치하는 VNF의 인스턴스(Instance)의 개수에 민감하게 반응하도록 학습할 수 있다.

수학식 14

[0167] $\sum_{t \in T} | \sum_{s \in N} \sum_i i(l_i^{st} - o_i^{st}) |$

[0168] 수학식 14는 수학식 12에 더하여 사용될 수 있다. 그러한 경우 학습 모델은 설치하는 서버의 위치에 대한 정보를 적게 고려하며 VNF의 종류 및 설치하는 VNF의 인스턴스(Instance)의 개수에 민감하게 반응하도록 학습할 수 있다.

수학식 15

[0169] $| \sum_{s \in N} \sum_i i(l_i^{st} - o_i^{st}) |$
 $t \in T$

[0170] 수학식 15는 수학식 12에 더하여 사용될 수 있다. 그러한 경우 학습 모델은 설치하는 서버의 위치 및 VNF의 종류 정보를 적게 고려하며 오직 설치하는 VNF의 인스턴스(Instance)의 개수에만 민감하게 반응하도록 학습할 수 있다.

[0171] 이와 같이, 본 실시예의 가상 네트워크 관리 장치(300)은 기계학습 모델의 출력부나 FNN의 출력부를 통해 모든 노드들이 모든 VNF 타입에 따라 어떠한 VNF 관리 정책을 내려야 하는지 최적으로 가까운 정답을 알려줄 수 있다. 즉, 본 실시예에서는 GNN 기반 기계학습 모델을 이용하여 네트워크 정보를 학습하고 최적의 VNF 배치 시나리오

를 생성할 수 있다.

- [0172] 도 10은 도 7의 기계 학습 모델을 이용하는 가상 네트워크 관리 방법의 작동 원리를 설명하기 위한 흐름도이다.
- [0173] 도 10을 참조하면, 가상 네트워크 관리 방법은, 컴퓨팅 장치의 프로세서에 탑재되는 프로그램에 의해, 먼저 기계 학습에 사용되는 데이터를 정의할 수 있다(S110). 본 단계(S110)에서, 노드와 링크로 구성되는 네트워크 데이터와 서비스 데이터는 VNF(Virtual Network Function) 열로 표현되고 정의될 수 있다.
- [0174] 데이터의 정의에 있어서, 노드는 가용 CPU(central processing unit) 코어의 개수 및 가용 대역폭에 대한 정보를 가지고 있고, 링크는 딜레이와 가용 대역폭에 대한 정보를 가지고 있으며, 서비스 데이터는 서비스가 시작되는 노드, 도달하는 노드, 서비스 타입, 서비스 지속시간, 서비스 지연으로 인한 비용, 최대 허용 지연 시간, 및 요구 대역폭에 대한 정보를 가질 수 있다.
- [0175] 다음, 데이터의 정의에 기초하여 리스트 생성 시점에서 네트워크에 요구되는 서비스들의 정보를 나타내는 서비스 정보 리스트를 생성하고, 서비스 정보 리스트가 생성될 때 물리 네트워크 정보와 가상 네트워크 기능(VNF: Virtual Network Function) 정보를 수집하고, 수집된 네트워크 데이터의 물리 네트워크 정보, VNF 정보 및 서비스 정보를 토대로 기계 학습에 활용할 정답 데이터를 준비할 수 있다(S120).
- [0176] 여기서, 서비스 정보 리스트는 서비스 리스트 정보 아이디를 갖고 서비스 시작 시간과 서비스 실행 시간을 이용하여 정의되는 서비스들과 가장 최근에 생성된 서비스를 합하여 생성될 수 있다. 그리고 정답 데이터 셋으로는, VNF의 설치 비용, 에너지 비용, 트래픽 전달 비용, 서비스 딜레이로 인한 비용 또는 이들의 조합을 포함한 정답 데이터 셋을 생성하거나 기저장된 정답 데이터 셋을 활용할 수 있다.
- [0177] 다음, 수집된 네트워크 데이터에 대하여 전처리를 수행할 수 있다(S130). 전처리에서는 네트워크 데이터 중 모든 수치 데이터를 정규화하여 표현하고, 원-핫 인코딩(one-hot-encoding) 알고리즘 등을 통해 네트워크 데이터 중 카테고리별 데이터들을 숫자 또는 벡터로 바꾸어 표현할 수 있다.
- [0178] 다음, 전처리된 네트워크 데이터 중 적어도 일부의 노드 데이터와 엣지 데이터를 그래프 데이터로 변환할 수 있다(S140). 본 단계(S140)에서는 각 노드 정보와 엣지 정보 및 노드와 노드 간과 노드와 엣지 간의 연결 정보를 가진 노드 행렬, 엣지 행렬 및 연결 행렬을 생성할 수 있다. 노드 데이터 또는 노드 정보는 노드 특징(Node Feature)을 포함하고, 엣지 데이터 또는 엣지 정보는 엣지 특징(Edge Feature)을 포함하며, 그래프 데이터는 노드 특징과 엣지 특징을 포함할 수 있다.
- [0179] 여기서, 노드 행렬은 노드 수 X 노드 피처의 행렬이고, 엣지 행렬은 노드 수 X 엣지 피처의 행렬일 수 있다. 또한, 연결 행렬은 노드 수 X 노드 수 행렬로 노드들 간의 링크가 존재하는 경우는 1의 값을 갖고, 노드들 간의 링크가 존재하지 않는 경우는 0의 값을 가질 수 있다.
- [0180] 다음, 노드 행렬, 엣지 행렬 및 연결 행렬로 표현되는 네트워크 데이터에 대해 그래프 신경망(GNN: Graph Neural Network)을 사용하여 학습을 수행한다(S150).
- [0181] 한편, 위의 단계들(S110 내지 S150)과는 별도로 서비스 요청에 대하여 FNN 학습을 통해 서비스 리스트를 생성하는 단계가 수행된다. 서비스 리스트 데이터의 생성 과정에서는 복수의 FNN의 가중치를 공유할 수 있다.
- [0182] 다음, GNN 신경망을 통한 학습 결과로 생성된 노드 상태 정보를 서비스 리스트 데이터와 연결(Concatenation)할 수 있다(S160).
- [0183] 다음, 노드 상태 정보와 서비스 리스트 데이터가 합쳐진 데이터를 순전파 신경망(FNN: Feedforward Neural Network)을 사용하여 학습한다(S170). FNN을 사용한 학습에서는, 학습 데이터의 클래스 불균형(Class Imbalance)을 막기 위해 제1 클래스의 데이터 개수가 다른 제2 클래스의 데이터 개수보다 적을 경우, 제1 클래스를 학습하는 데 부여하는 제1 가중치를 제2 클래스를 학습하는데 부여하는 제2 가중치보다 더 크게 설정할 수 있다.
- [0184] 본 실시예에 따른 가상 네트워크 관리 방법은, FNN을 사용한 학습 단계(S170) 후에, 학습된 모든 노드들에 대해 생성(Add), 제거(Remove) 또는 그대로 두기(Do Nothing)를 포함한 가상 네트워크 기능(VNF: Virtual Network Function) 관리 결정을 생성하도록 구성될 수 있다. VNF 관리 결정은 최적의 VNF 관리 정책에 대응된다.
- [0185] 전술한 실시예들을 통해 설명한 가상 네트워크 관리 방법은 다양한 컴퓨터 수단을 통해 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 컴퓨터 판독 가능 매체에 기록되는 프

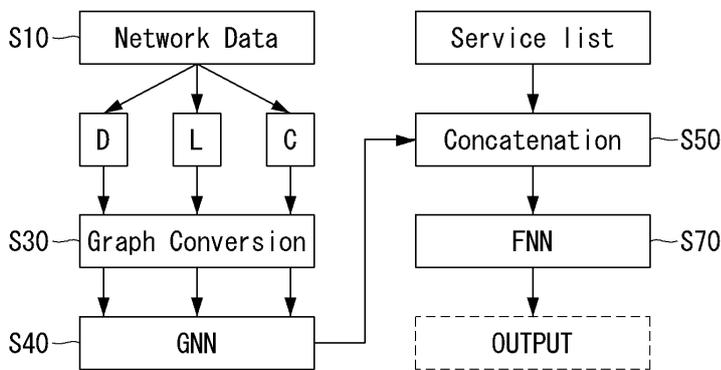
로그래밍 명령은 본 발명을 위해 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용가능한 것일 수 있다.

[0186] 컴퓨터 판독 가능 매체의 예에는 롬(rom), 램(ram), 플래시 메모리(flash memory) 등과 같이 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러(compiler)에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터(interpreter) 등을 사용해서 컴퓨터에 의해 실행될 수 있는 고급 언어 코드를 포함한다. 상술한 하드웨어 장치는 본 실시예에 따른 가상 네트워크 관리 방법의 일련의 동작을 수행하기 위해 적어도 하나의 소프트웨어 모듈로 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

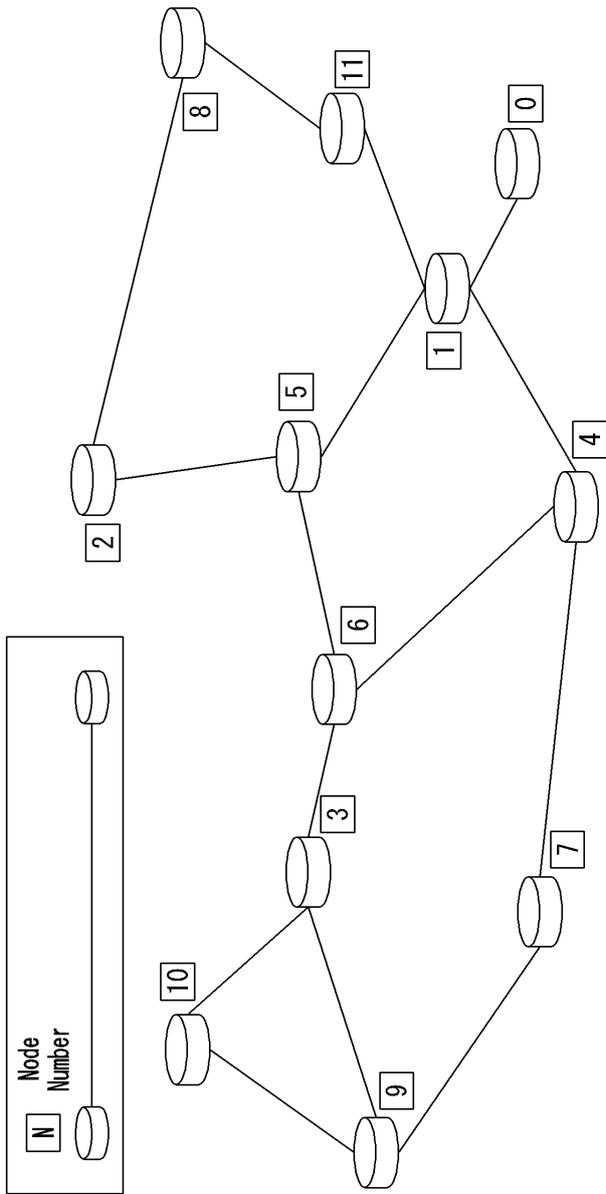
[0187] 이상 실시예를 참조하여 설명하였지만, 해당 기술 분야의 숙련된 당업자는 하기의 청구범위에 기재된 본 발명의 사상 및 영역으로부터 벗어나지 않는 범위 내에서 본 발명을 다양하게 수정 및 변경시킬 수 있음을 이해할 수 있을 것이다.

도면

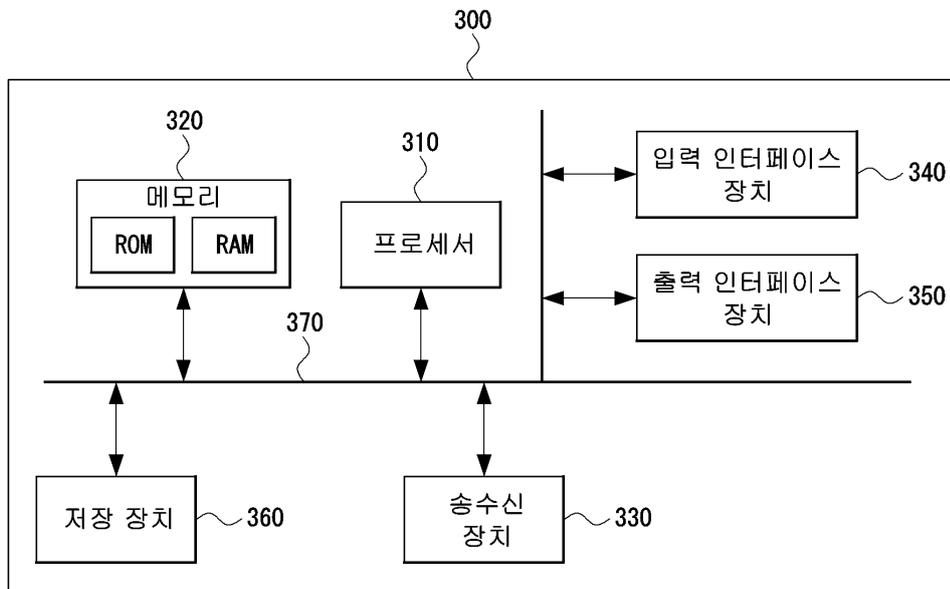
도면1



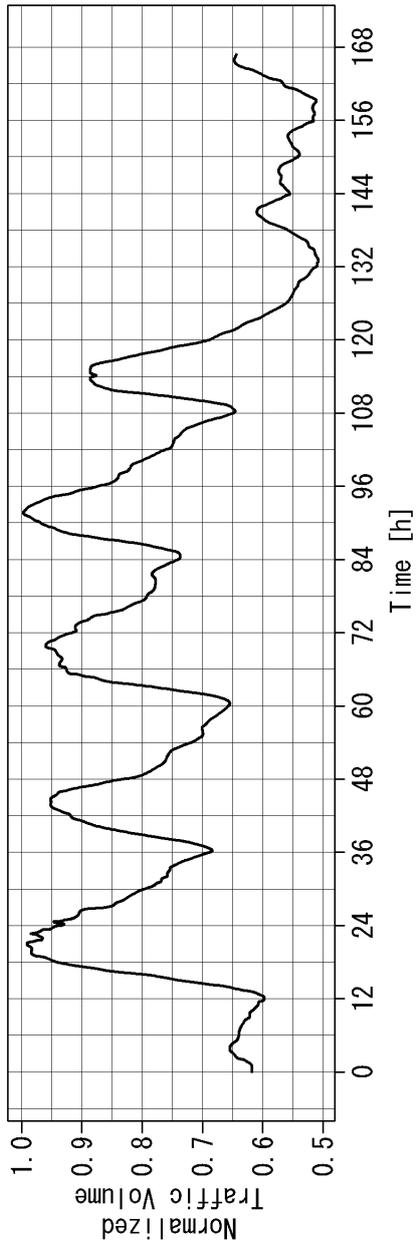
도면2



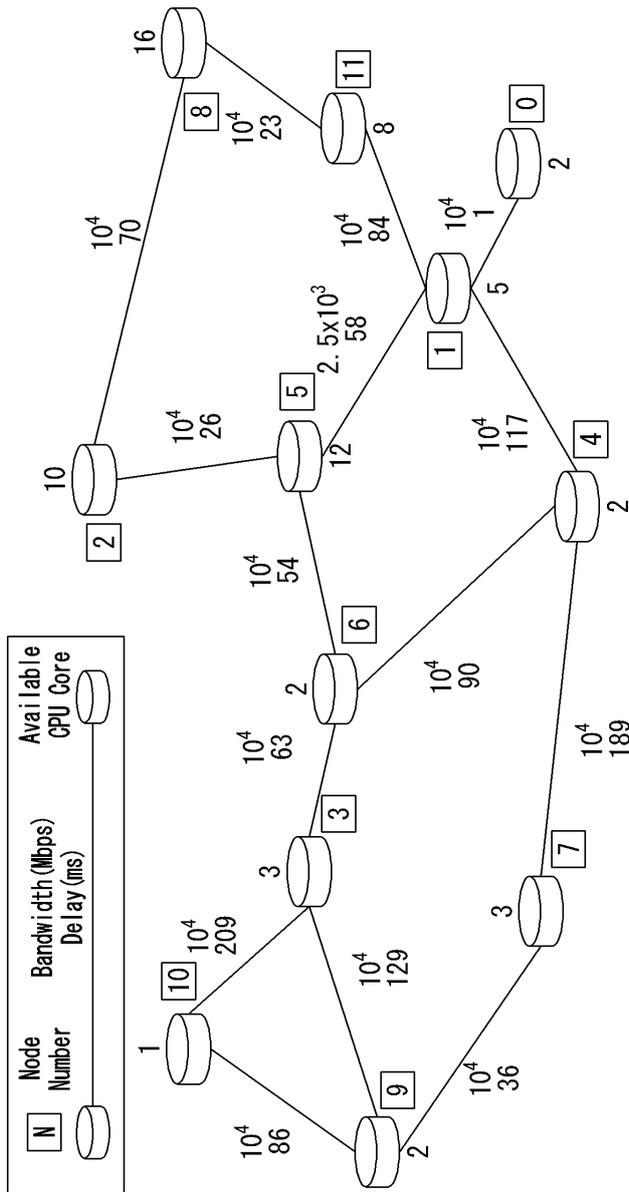
도면3



도면4



도면5



도면6a

Servers	1	3	0	1	2	0
	2	1	0	4	2	0
	3	2	1	0	3	1
	...	2	4	1	1	1
	12	0	1	2	3	2
		F	P	I	N	W
		VNF types				

Current VNF instance number

도면6b

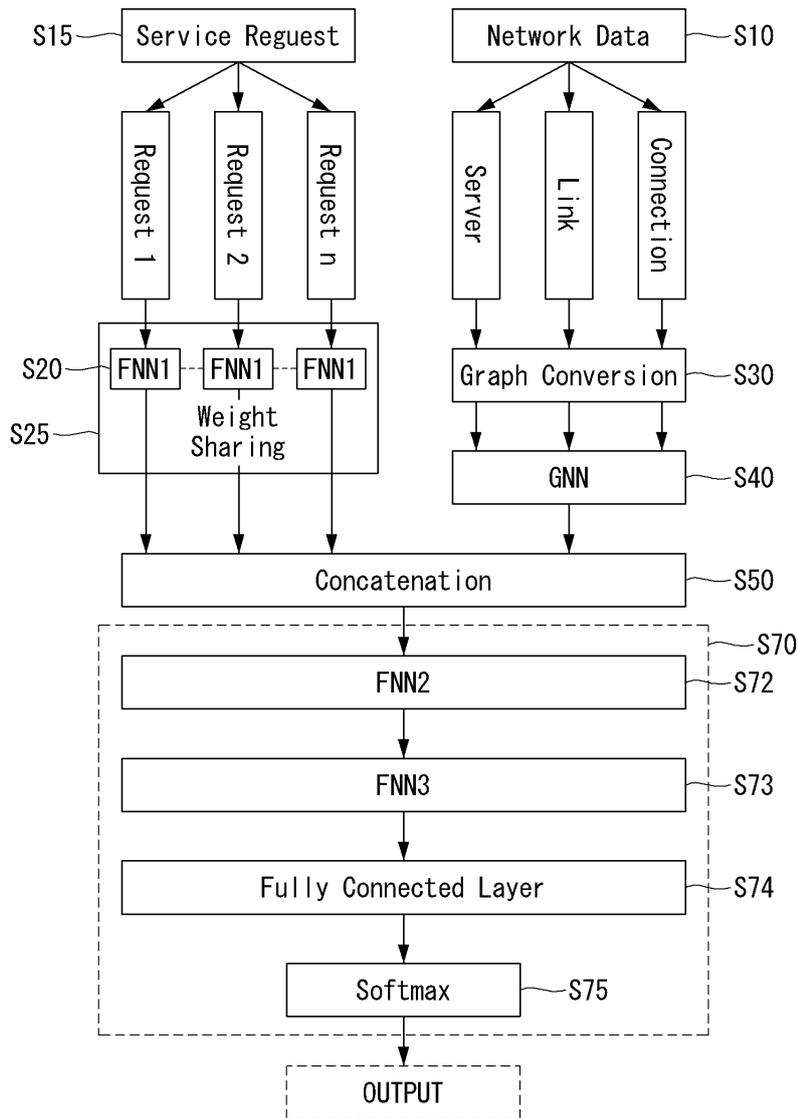
Servers	1	2	0	1	3	0
	2	1	0	4	2	2
	3	1	1	2	3	0
	...	0	3	1	1	1
	12	1	1	2	3	1
		F	P	I	N	W
		VNF types				

Optimal VNF instance number

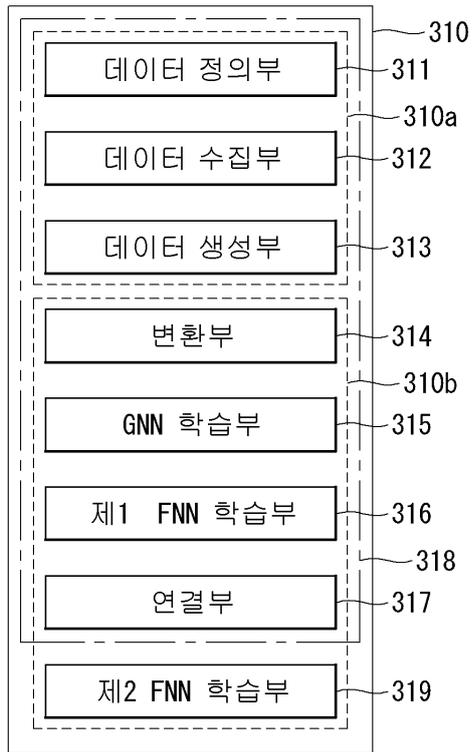
도면6c

Servers	1	-1	0	0	1	0
	2	0	0	0	0	1
	3	-1	0	1	0	1
	...	-1	-1	0	0	0
	12	1	0	0	0	-1
		F	P	I	N	W
		VNF types				

도면7



도면8



도면9



도면10

