



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0070934
(43) 공개일자 2025년05월21일

(51) 국제특허분류(Int. Cl.)
G06N 3/10 (2006.01) G06N 3/045 (2023.01)
G06N 3/048 (2023.01)
(52) CPC특허분류
G06N 3/10 (2023.01)
G06N 3/045 (2023.01)
(21) 출원번호 10-2023-0157370
(22) 출원일자 2023년11월14일
심사청구일자 2023년11월14일

(71) 출원인
인하대학교 산학협력단
인천광역시 미추홀구 인하로 100(용현동, 인하대 학교)
(72) 발명자
최원익
인천광역시 연수구 해송로30번길 20, 404동 702호 (송도동, 송도웰카운티4단지)
이충현
인천광역시 미추홀구 재념이길 141-33, 2동 402호 (학익동, 우민하이츠빌)
(74) 대리인
양성보

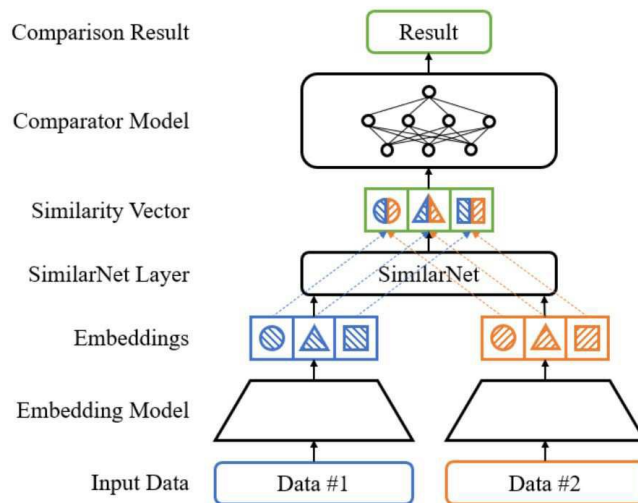
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 다양한 비교모델을 위한 쌍별 유사 비교자 기술

(57) 요약

다양한 비교모델을 위한 쌍별 유사 비교자 기술이 개시된다. 일 실시예에 따른 유사성 비교 시스템에 의해 수행되는 유사성 비교 방법은, 특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계하는 단계; 및 상기 설계된 신경망 레이어를 통해 특징 벡터 간 유사성을 비교하는 단계를 포함할 수 있다.

대표도 - 도5



(52) CPC특허분류

G06N 3/048 (2023.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1615013089
과제번호	00141819
부처명	국토교통부
과제관리(전문)기관명	국토교통과학기술진흥원
연구사업명	디지털국토정보기술개발사업
연구과제명	[Ezbaro] 초정밀 디지털 국토정보 획득을 위한 절대, 상대, 연속복합 측위 고도화
기술 개발	
기 여 율	1/1
과제수행기관명	한국전자통신연구원
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

유사성 비교 시스템에 의해 수행되는 유사성 비교 방법에 있어서,
특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계하는 단계;
및
상기 설계된 신경망 레이어를 통해 특징 벡터 간 유사성을 비교하는 단계
를 포함하는 유사성 비교 방법.

청구항 2

제1항에 있어서,
상기 설계하는 단계는,
L2 정규화 및 코사인 활성화 함수를 구성하여 동일한 특징에 해당하는 요소에 대해 연산을 수행하여 유효하지 않는 요소의 발생을 최소화하면서 유사성을 계산하는 제1 신경망 레이어를 설계하는 단계
를 포함하고,
상기 코사인 활성화 함수는, 하다마드 곱을 각 요소의 함수로 재구성하여 함수 퍼셉트론에 대한 활성화 함수로 사용하여 요소들의 거리가 가까울수록 높은 값을 출력하고 멀어질수록 낮은 값을 출력하도록 구성된 것을 특징으로 하는 유사성 비교 방법.

청구항 3

제1항에 있어서,
상기 설계하는 단계는,
훈련 가능한 매개변수와 훈련 가능한 활성화 함수를 적용하여 각 특징의 유사성과 차이점에 대해 서로 다른 가중치를 할당하도록 제2 신경망 레이어를 설계하는 단계
를 포함하고,
상기 훈련 가능한 활성화 함수는, 파라메트릭 코사인(PCosine)을 통해 각 특징 벡터에 기 설정된 기준 이상의 유사성이 있을 때 특정 차이점을 무시하여 유사성 값을 산출하도록 구성된 것을 특징으로 하는 유사성 비교 방법.

청구항 4

제1항에 있어서,
상기 설계하는 단계는,
내적을 적용하여 이중 벡터를 직접 비교하는 제3 신경망 레이어를 설계하는 단계
를 포함하는 유사성 비교 방법.

청구항 5

제1항에 있어서,
상기 비교하는 단계는,
복수 개의 입력 데이터가 각 임베딩 모델에 입력되고, 상기 각 임베딩 모델을 통해 임베딩된 특징 벡터의 각각

을 상기 신경망 레이어에 입력하고, 상기 신경망 레이어를 통해 유사성 벡터를 출력하는 단계
를 포함하고,

상기 각 임베딩 모델은, 삼 네트워크(Siamese Network) 구조로 구성된 것을 특징으로 하는 유사성 비교 방법.

청구항 6

제1항에 있어서,

상기 신경망 레이어를 설계하는 단계는,

상기 설계된 신경망 레이어를 포함하는 비교 모델을 구축하는 단계

를 포함하는 유사성 비교 방법.

청구항 7

유사성 비교 시스템에 있어서,

특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계하는 레이어
설계부; 및

상기 설계된 신경망 레이어를 통해 특징 벡터 간 유사성을 비교하는 유사성 비교부

를 포함하는 유사성 비교 시스템.

발명의 설명

기술 분야

[0001] 아래의 설명은 유사성 비교 기술에 관한 것이다.

배경 기술

[0003] 최근 몇 년 동안 딥 러닝은 다양한 분야에서 복잡한 문제를 해결할 수 있는 능력으로 인해 상당한 주목을 받고 있다. 주목한 만한 문제 중 하나는 임베딩 모델을 훈련하여 임베딩 공간에서 특징 벡터의 메트릭 공간 유사성을 계산하여 특징 임베딩을 학습하는 것을 목표로 하는 메트릭 공간 학습이다. 그러나, 메트릭 공간 유사성에 대한 연구는 제한적이며 코사인 유사성 또는 연결 레이어(concatenation layer)를 기반으로 하는 기존 방법은 유연성과 성능 측면에서 단점을 나타낸다. 예를 들어, 코사인 유사성을 적용하려면 두 벡터의 모양이 동일해야 하므로 고급 비교기를 사용할 수 없다. 또한, 연결 레이어는 두 벡터의 요소 위치를 반영하지 못하여 성능 및 학습 성공률을 저하시킨다.

발명의 내용

해결하려는 과제

[0005] 벡터 형식으로 출력을 생성하기 위해 요소의 위치를 고려하면서 두 특징 벡터를 비교하도록 신경망 레이어를 제안할 수 있다.

[0006] 코사인 유사성과 연결 레이어의 장점을 활용하 특징 벡터를 효과적으로 비교할 수 있다.

과제의 해결 수단

[0008] 유사성 비교 시스템에 의해 수행되는 유사성 비교 방법은, 특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계하는 단계; 및 상기 설계된 신경망 레이어를 통해 특징 벡터 간

유사성을 비교하는 단계를 포함할 수 있다.

- [0009] 상기 설계하는 단계는, L2 정규화 및 코사인 활성화 함수를 구성하여 동일한 특징에 해당하는 요소에 대해 연산을 수행하여 유효하지 않는 요소의 발생을 최소화하면서 유사성을 계산하는 제1 신경망 레이어를 설계하는 단계를 포함하고, 상기 코사인 활성화 함수는, 하다마드 곱을 각 요소의 함수로 재구성하여 함수 퍼셉트론에 대한 활성화 함수로 사용하여 요소들의 거리가 가까울수록 높은 값을 출력하고 멀어질수록 낮은 값을 출력하도록 구성된 것일 수 있다.
- [0010] 상기 설계하는 단계는, 훈련 가능한 매개변수와 훈련 가능한 활성화 함수를 적용하여 각 특징의 유사성과 차이점에 대해 서로 다른 가중치를 할당하도록 제2 신경망 레이어를 설계하는 단계를 포함하고, 상기 훈련 가능한 활성화 함수는, 파라메트릭 코사인(PCosine)을 통해 각 특징 벡터에 기 설정된 기준 이상의 유사성이 있을 때 특정 차이점을 무시하여 유사성 값을 산출하도록 구성된 것일 수 있다.
- [0011] 상기 설계하는 단계는, 내적을 적용하여 이중 벡터를 직접 비교하는 제3 신경망 레이어를 설계하는 단계를 포함할 수 있다.
- [0012] 상기 비교하는 단계는, 복수 개의 입력 데이터가 각 임베딩 모델에 입력되고, 상기 각 임베딩 모델을 통해 임베딩된 특징 벡터의 각각을 상기 신경망 레이어에 입력하고, 상기 신경망 레이어를 통해 유사성 벡터를 출력하는 단계를 포함하고, 상기 각 임베딩 모델은, 삼 네트워크(Siamese Network) 구조로 구성된 것일 수 있다.
- [0013] 상기 신경망 레이어를 설계하는 단계는, 상기 설계된 신경망 레이어를 포함하는 비교 모델을 구축하는 단계를 포함할 수 있다.
- [0014] 유사성 비교 시스템은, 특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계하는 레이어 설계부; 및 상기 설계된 신경망 레이어를 통해 특징 벡터 간 유사성을 비교하는 유사성 비교부를 포함할 수 있다.

발명의 효과

- [0016] 요소의 위치를 고려하여 특징 벡터를 비교하도록 설계된 신경망 레이어를 통해 비교 정확도와 학습 성공률을 향상시킬 수 있다.

도면의 간단한 설명

- [0018] 도 1은 일 실시예에 있어서, 유사성 비교 시스템의 구성을 설명하기 위한 블록도이다.
- 도 2는 일 실시예에 있어서, 유사성 비교 방법을 설명하기 위한 흐름도이다.
- 도 3은 일 실시예에 있어서, 코사인 유사성 함수를 설명하기 위한 도면이다.
- 도 4는 일 실시예에 있어서, 코사인 유사성과 실시예에서 제안된 신경망 레이어의 구조를 비교하기 위한 도면이다.
- 도 5는 일 실시예에 있어서, 신경망 레이어를 사용한 효율적인 비교 모델을 설명하기 위한 도면이다.
- 도 6은 일 실시예에 있어서, PReLU 그래프를 나타낸 예이다.
- 도 7은 일 실시예에 있어서, PCosine 그래프를 나타낸 예이다.
- 도 8은 일 실시예에 있어서, 신경망 레이어의 구조를 설명하기 위한 다른 예이다.
- 도 9는 일 실시예에 있어서, 이중 임베딩의 예이다.
- 도 10은 일 실시예에 있어서, 신경망 레이어의 구조를 설명하기 위한 또 다른 예이다.

발명을 실시하기 위한 구체적인 내용

- [0019] 이하, 실시예를 첨부한 도면을 참조하여 상세히 설명한다.
- [0021] 도 1은 일 실시예에 있어서, 유사성 비교 시스템의 구성을 설명하기 위한 블록도이고, 도 2는 일 실시예에 있어서, 유사성 비교 방법을 설명하기 위한 흐름도이다.
- [0022] 유사성 비교 시스템(100)의 프로세서는 레이어 설계부(110) 및 유사성 비교부(120)를 포함할 수 있다. 이러한 프로세서의 구성요소들은 유사성 비교 시스템에 저장된 프로그램 코드가 제공하는 제어 명령에 따라 프로세서에 의해 수행되는 서로 다른 기능들(different functions)의 표현들일 수 있다. 프로세서 및 프로세서의 구성요소들은 도 2의 유사성 비교 방법이 포함하는 단계들(210 내지 220)을 수행하는 유사성 비교 시스템을 제어할 수 있다. 이때, 프로세서 및 프로세서의 구성요소들은 메모리가 포함하는 운영체제의 코드와 적어도 하나의 프로그램의 코드에 따른 명령(instruction)을 실행하도록 구현될 수 있다.
- [0023] 프로세서는 유사성 비교 방법을 위한 프로그램의 파일에 저장된 프로그램 코드를 메모리에 로딩할 수 있다. 예를 들면, 유사성 비교 시스템에서 프로그램이 실행되면, 프로세서는 운영체제의 제어에 따라 프로그램의 파일로부터 프로그램 코드를 메모리에 로딩하도록 유사성 비교 시스템을 제어할 수 있다. 이때 프로세서는 레이어 설계부(110) 및 유사성 비교부(120) 각각은 메모리에 로딩된 프로그램 코드 중 대응하는 부분의 명령을 실행하여 이후 단계들(210 내지 220)을 실행하기 위한 프로세서의 서로 다른 기능적 표현들일 수 있다.
- [0024] 단계(210)에서 레이어 설계부(110)는 특징 벡터 사이에 관련된 요소의 위치에 기초하여 특징 벡터를 비교하기 위한 신경망 레이어를 설계할 수 있다. 레이어 설계부(110)는 L2 정규화 및 코사인 활성화 함수를 구성하여 동일한 특징에 해당하는 요소에 대해 연산을 수행하여 유효하지 않는 요소의 발생을 최소화하면서 유사성을 계산하는 제1 신경망 레이어를 설계할 수 있다. 여기서, 코사인 활성화 함수는, 하다마드 곱을 각 요소의 함수로 재구성하여 함수 퍼셉트론에 대한 활성화 함수로 사용하여 요소들의 거리가 가까울수록 높은 값을 출력하고 멀어질수록 낮은 값을 출력하도록 구성된 것일 수 있다. 레이어 설계부(110)는 훈련 가능한 매개변수와 훈련 가능한 활성화 함수를 적용하여 각 특징의 유사성과 차이점에 대해 서로 다른 가중치를 할당하도록 제2 신경망 레이어를 설계할 수 있다. 여기서, 훈련 가능한 활성화 함수는, 파라메트릭 코사인(PCosine)을 통해 각 특징 벡터에 기 설정된 기준 이상의 유사성이 있을 때 특정 차이점을 무시하여 유사성 값을 산출하도록 구성된 것일 수 있다. 레이어 설계부(110)는 내적을 적용하여 이중 벡터를 직접 비교하는 제3 신경망 레이어를 설계할 수 있다.
- [0025] 단계(220)에서 유사성 비교부(120)는 설계된 신경망 레이어를 통해 특징 벡터 간 유사성을 비교할 수 있다. 유사성 비교부(120)는 복수 개의 입력 데이터가 각 임베딩 모델에 입력되고, 각 임베딩 모델을 통해 임베딩된 특징 벡터의 각각을 신경망 레이어에 입력하고, 신경망 레이어를 통해 유사성 벡터를 출력할 수 있다. 이때, 각 임베딩 모델은, 삼 네트워크(Siamese Network) 구조로 구성될 수 있다. 유사성 비교부(120)는 설계된 신경망 레이어를 포함하는 비교 모델을 구축할 수 있다.
- [0026] 도 4는 일 실시예에 있어서, 코사인 유사성과 실시예에서 제안된 신경망 레이어의 구조를 비교하기 위한 도면이다.
- [0027] 실시예에서는 다양한 비교 문제에 적용할 수 있는 효율적인 비교에 특화된 신경망 레이어(SimilarNet)에 대하여 설명하기로 한다. 이러한 신경망 레이어는 연결 전력과 달리 비교 대상이 되는 두 입력 벡터를 구분하고 각 요소의 위치에 따라 비교에 필요한 연산을 선택적으로 수행한다. 이 설계를 통해 비교 문제의 정확도와 학습 성공률을 효과적으로 높일 수 있다.
- [0028] 또한, 실시예에서 제안되는 신경망 레이어는 코사인 유사성과 같은 기존 지표로는 사용할 수 없는 유연성 수준을 제공한다. 신경망 레이어의 출력은 하나의 스칼라가 아닌 벡터의 형태이기 때문에 그 위에 다양한 모델을 적용하고 기존의 학습 방법론을 사용하여 학습할 수 있다.
- [0029] 또한, 실시예에서 이중 벡터의 비교를 용이하게 하기 위해 신경망 레이어의 기능을 확장한다. 비교 대상에 대한 최적 특성을 갖는 비교 연산은 퍼셉트론에 정의된 활성화 함수를 교환함으로써 구현할 수 있다.
- [0030] 보다 상세하게는, L2 정규화 및 코사인 활성화 함수를 구성하여 동일한 특징에 해당하는 요소에 대해 연산을 수행하여 유효하지 않는 요소의 발생을 최소화하면서 유사성을 계산하는 제1 신경망 레이어(바닐라 SimilarNet)의 설계 및 구현 동작에 대하여 설명하기로 한다.
- [0031] 신경망 레이어의 설계 목표는 연결 레이어의 유연성과 코사인 유사성의 효율성을 활용하는 것이다. 이를 위해

신경망 레이어는 인공 신경망에서 영감을 받은 코사인 유사도 알고리즘의 변형된 형태를 사용한다. 구체적으로, 코사인 유사성 알고리즘을 L2 정규화, 하다마드(Hadamard) 곱, 요소 합의 세 개의 계층으로 분할한다.

[0032] 인공 신경망은 일반적으로 활성화 함수에 의존한다. 그러나 $H(X, Y) = X \odot Y$ 로 정의된 행렬 연산인 하다마드 곱을 활성화 함수로 직접 적용할 수는 없다. 이 문제를 해결하기 위해 두 행렬의 각 요소를 입력으로 하는 3차원 활성화 함수를 사용하는 함수 퍼셉트론의 개념을 소개한다. 구체적으로, 하다마드 곱을 각 요소 $f(x_i, y_i) = x_i y_i$ 의 함수로 재구성하여 함수 퍼셉트론에 대한 활성화 함수로 사용할 수 있다. 이 함수를 코사인 활성화 함수라고 한다. 도 3을 참고하면, 코사인 활성화 함수를 설명하기 위한 도면이다. 코사인 활성화 함수는 도 3과 같이 양변의 원소 x 와 y 가 서로 가까울수록 더 높은 값을 출력하고, 더 멀수록 있을 때는 낮은 값을 출력한다.

[0033] 신경망 레이어는 하다마드 곱 대신 코사인 활성화 함수를 기반으로 한 함수 퍼셉트론을 사용한다. 또한, 스칼라 대신 벡터의 형태로 출력을 획득하기 위해 코사인 유사성의 끝에서 합 연산을 제거한다. 출력된 벡터 표현은 모델의 유연성을 향상시키는 데 도움이 된다. 코사인 유사성과 신경망 레이어의 구조를 비교 분석은 알고리즘 1과 도 4에 도시되어 있다.

Algorithm 1-1. Cosine similarity	Algorithm 1-2. SimilarNet
1 Input: vector x , vector y	1 Input: vector x , vector y
2 Output: scalar <i>similarity</i>	2 Output: vector z
3	3
4 $x \leftarrow \text{L2_normalize}(x)$	4 $x \leftarrow \text{L2_normalize}(x)$
5 $y \leftarrow \text{L2_normalize}(y)$	5 $y \leftarrow \text{L2_normalize}(y)$
6 $z \leftarrow \text{Hadamard_product}(x, y)$	6 $z \leftarrow \text{Cosine_activation}(x, y)$
7 $\text{similarity} \leftarrow \text{sum}(z)$	7 // eliminates the sum operation
8	8
9 return <i>similarity</i>	9 return z

[0034]

[0035] 신경망 레이어에서 코사인 유사성과 유사하게 벡터의 양쪽에서 동일한 인덱스에 해당하는 요소에 대해서만 연산이 수행된다. 특히, 연결 레이어를 사용하는 경우, 서로 다른 특징의 요소들이 강제로 결합되어 유효하지 않은(무효) 요소가 발생한다. 이에 반해 신경망 레이어는 동일한 특징에 해당하는 요소에 대해 연산을 수행하여 유효하지 않는 요소의 발생을 최소화하고 모델이 효율적으로 유사성을 계산할 수 있도록 한다. 또한, 출력이 스칼라가 아닌 유사성 벡터를 나타내기 때문에 도 5와 같이 신경망 레이어 이후에 다양한 비교 모델을 유연하게 도입할 수 있다.

[0036] 다음으로, 훈련 가능한 매개변수와 훈련 가능한 활성화 함수를 적용하여 각 특징의 유사성과 차이점에 대해 서로 다른 가중치를 할당하도록 제2 신경망 레이어(파라메트릭 SimilarNet)의 설계 및 구현 동작에 대하여 설명하기로 한다.

[0037] 적절한 비교를 수행하기 위해서는 비교 대상 데이터의 특성에 맞는 방법과 기준을 학습해야 하는데, 각 특징의 유사성과 차이점에 대한 가중치를 별도로 계산해야 하는 경우가 대표적이다. 그러나 제1 신경망 레이어는 일반적인 인공 신경망과 달리 훈련 가능한 가중치와 편향성을 가지고 있지 않아 이러한 경우를 관리하는 데 효과적이지 못하다.

[0038] 학습 능력을 향상시키기 위해, 훈련 가능한 매개변수와 활성화 함수를 적용하여 각 특징을 비교하는 방법을 학습할 수 있도록 함으로써 제1 신경망 레이어를 제2 신경망 레이어로 확장한다. 훈련 가능한 활성화 함수의 예는 파라메트릭 PReLU(Parametric Rectified Linear Unit)이며, 이는 완전 연결 계층에서 일반적으로 사용되는 ReLU를 확장하여 각 퍼셉트론에 대해 음수(negative) 부분의 가중치를 α 처럼 훈련시킬 수 있도록 한다. PReLU의 그래프는 도 6에 도시되어 있다.

[0039] PReLU와 유사하게 제2 신경망 레이어는 양수(positive)와 음수(negative) 부분에 대해 서로 다른 가중치를 학습하도록 설계되어 있다. 즉, 비교를 수행할 때 각 특징의 유사성과 차이점에 대해 서로 다른 가중치를 할당한다.

[0040] 이 활성화 함수를 파라메트릭 코사인(PCosine)이라고 한다. PCosine을 사용하면 모델은 명확한 유사성이 있을 때 쌍의 특정 차이점을 무시하는 방법을 학습하여 높은 유사성 값을 산출할 수 있다. PCosine은 다음과 같이 정의된다

$$f(x_i, y_i) = \begin{cases} x_i y_i, & \text{if } x_i y_i \geq 0 \\ \alpha_i x_i y_i, & \text{if } x_i y_i < 0 \end{cases} \quad (1)$$

[0042] 도 7에서 볼 수 있듯이 PCosine의 모양은 α 에 의존한다. 이 활성화 함수를 통합하는 파라메트릭 SimilarNet의 구조는 도 8에 도시되어 있다.

[0043] 다음으로, 내적을 적용하여 이중 벡터를 직접 비교하는 제3 신경망 레이어(이중 SimilarNet)의 설계 및 구현 동작에 대하여 설명하기로 한다.

[0044] 비교 문제에서, 비교되는 두 데이터셋의 모양은 일반적으로 동일하다고 가정된다. 또한, 비교되는 두 벡터에서 각 특징을 나타내는 기본 벡터는 동일한 것으로 간주된다. 유클리드 거리, 코사인 유사성 등 다양한 메트릭과 평균 제곱 오차(MSE) 및 평균 절대 오차(MAE)와 같은 비교 함수는 두 벡터의 모양과 기저 벡터가 동일한 경우에 만 의미있게 사용될 수 있다.

[0045] 그러나, 서로 다른 모양의 벡터를 비교할 때 도 9와 같이 임베딩 벡터의 모양도 다르다. 따라서 이중 벡터를 처리하는 새로운 전략을 확인해야 한다. 예를 들어, 추천 문제는 여러 이중 벡터에 대한 비교 문제로 간주될 수 있다. 이 문제를 해결하기 위해 클러스터링 또는 연결과 같은 방법을 사용하여 모델을 훈련하기 위한 단일 벡터를 생성함으로써 두 데이터셋의 직접 비교를 피할 수 있다.

[0046] 제3 신경망 레이어는 도 10과 같이 내적을 적용하여 두 개의 이중 벡터를 직접 비교할 수 있도록 한다. 이 테스트 프로세스는 자연어 처리에서 일반적으로 사용되는 내적 주의 기법과 유사하다. 알고리즘 2에서는 이 프로세스를 설명하는데, 알고리즘 1에서 두 벡터에 대한 평탄화(flatten) 연산과 한 벡터에 대한 전치 연산을 추가하여 구현할 수 있다.

Algorithm 2. Heterogeneous SimilarNet

```

1  Input: vector  $x$ , vector  $y$ 
2  Output: vector  $z$ 
3
4   $x \leftarrow \text{L2\_normalize}(x)$ 
5   $y \leftarrow \text{L2\_normalize}(y)$ 
6   $x \leftarrow \text{Flatten}(x)$ 
7   $y \leftarrow \text{Flatten}(y)$ 
8   $x \leftarrow x^t$ 
9   $z \leftarrow x \cdot y$ 
10
11 return  $z$ 

```

[0047]

[0048] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 컨트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 컨트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0049] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의

조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embody)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

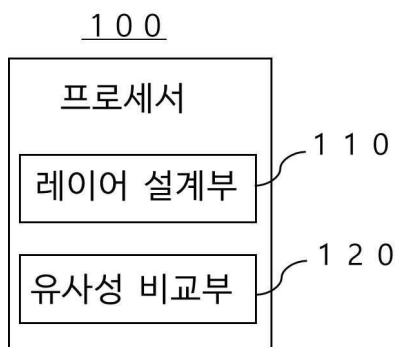
[0050] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0051] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

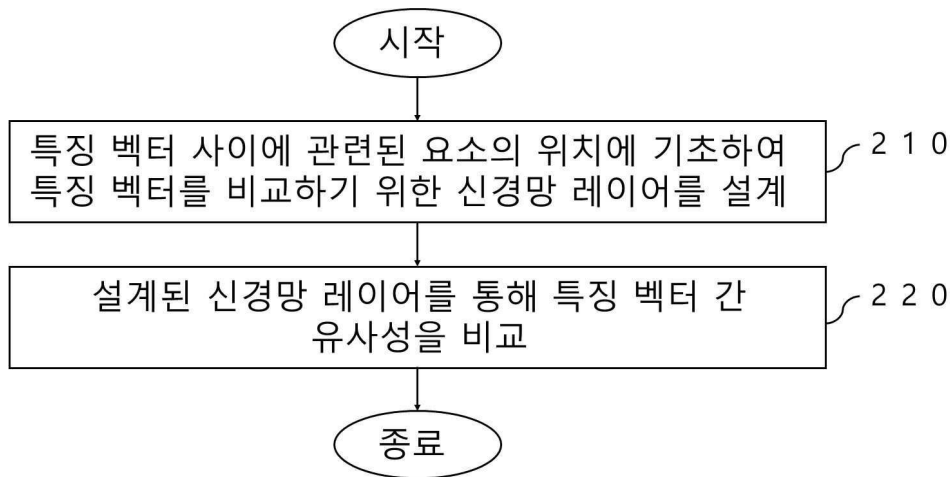
[0052] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

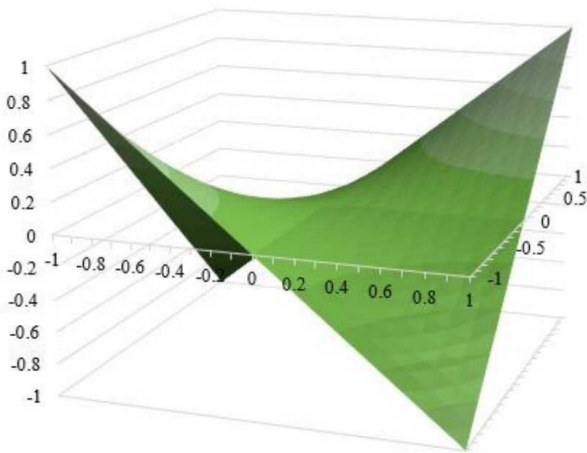
도면1



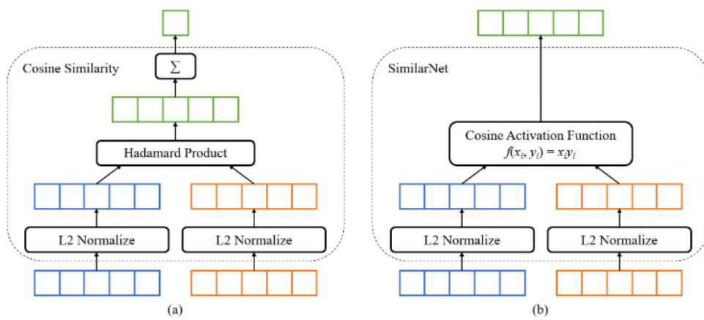
도면2



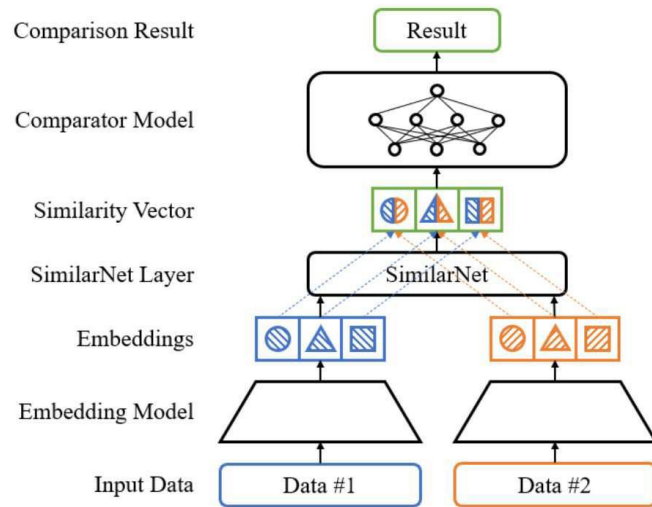
도면3



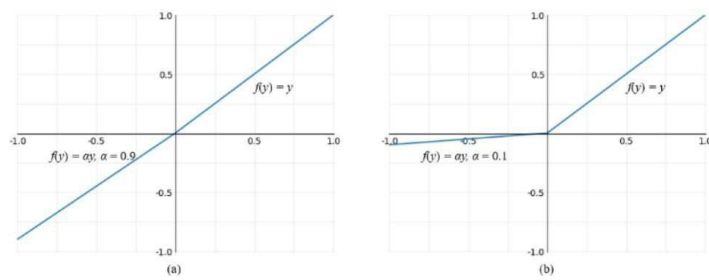
도면4



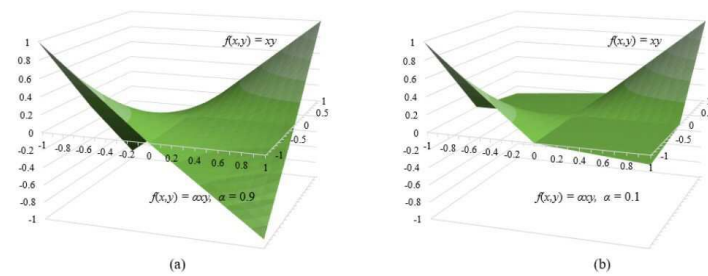
도면5



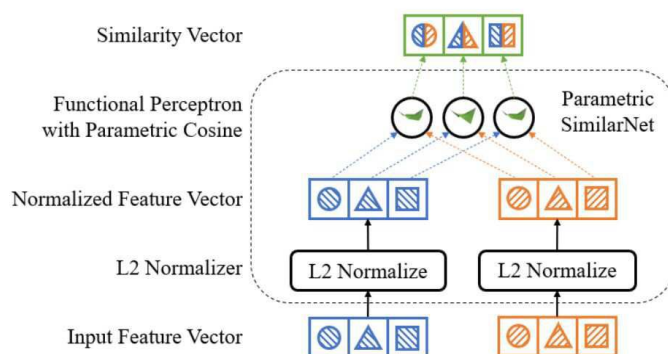
도면6



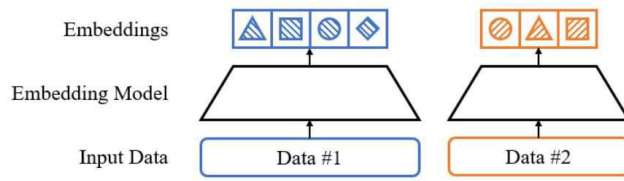
도면7



도면8



도면9



도면10

