(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년12월16일
(11) 등록번호 10-2191722
(24) 등록일자 2020년12월10일

(51) 국제특허분류(Int. Cl.)
G06N 3/04 (2006.01) *G06N 3/08* (2006.01)
(52) CPC특허분류
G06N 3/04 (2013.01)
G06N 3/08 (2013.01)
(21) 출원번호 10-2020-0087811
(22) 출원일자 2020년07월15일
심사청구일자 2020년07월15일
(56) 선행기술조사문헌
Jinhan Kim 외 2인, Guiding Deep Learning System Testing Using Surprise Adequacy, 2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE), pp 1039-1049, 2018.08.25
Lei Ma 외 11인, DeepGauge: Multi-Granularity Testing Criteria for Deep Learning Systems, Proceedings of the 2018 33rd ACM/IEEE International Conference on Automated Software Engineering, 2018.09.07

(73) 특허권자
세종대학교산학협력단
서울특별시 광진구 능동로 209 (군자동, 세종대학교)
(72) 발명자
윤주범
서울특별시 송파구 충민로4길 19, 704동 401호(장지동, 송파파인타운7단지)
유지현
서울특별시 광진구 군자로3길 18-2, 101호 (화양동)
문현준
서울특별시 강북구 삼양로42길 15 (미아동)
(74) 대리인
두호특허법인

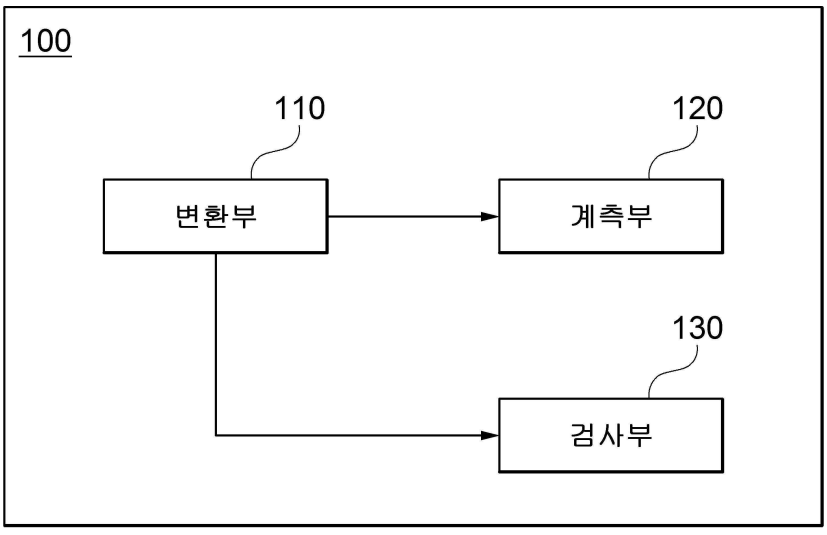
전체 청구항 수 : 총 12 항 심사관 : 박상현

(54) 발명의 명칭 **딥러닝 모델의 취약점 판단 장치 및 방법**

(57) 요 약

딥러닝 모델의 취약점을 판단하는 장치 및 방법이 개시된다. 일 실시예에 따른 딥러닝 모델의 취약점 판단 장치는, 이미지 데이터셋에서 선택된 원본 이미지를 변형하여 딥러닝 모델에 대한 입력 이미지를 생성하는 변환부; 상기 입력 이미지를 상기 딥러닝 모델에 입력하여 상기 딥러닝 모델의 뉴런 커버리지(Neuron Coverage)를 측정하는 계측부; 및 상기 입력 이미지의 클래스에 대한 상기 딥러닝 모델의 예측 결과와 상기 원본 이미지의 클래스에 기초하여 상기 예측 결과에 대한 오류를 탐지하는 검사부를 포함한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업

과제고유번호	1711116145
과제번호	2018-0-01423-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	대학ICT연구센터육성지원사업
연구과제명	지능형 비행로봇 융합기술 연구
기 여 율	1/1
과제수행기관명	세종대학교 산학협력단
연구기간	2020.01.01 ~ 2020.12.31

공지예외적용 : 있음

명 세 서

청구범위

청구항 1

이미지 데이터셋에서 선택된 원본 이미지를 변형하여 딥러닝 모델에 대한 입력 이미지를 생성하는 변환부;

상기 입력 이미지를 상기 딥러닝 모델에 입력하여 상기 딥러닝 모델의 뉴런 커버리지(Neuron Coverage)를 측정하는 계측부; 및

상기 입력 이미지의 클래스에 대한 상기 딥러닝 모델의 예측 결과와 상기 원본 이미지의 클래스에 기초하여 상기 예측 결과에 대한 오류를 탐지하는 검사부를 포함하며,

상기 검사부는, 상기 예측 결과에 오류가 검출된 경우, 상기 딥러닝 모델의 속성을 기초로 상기 딥러닝 모델이 갖는 취약점의 종류를 분류하는, 딥러닝 모델의 취약점 판단 장치.

청구항 2

청구항 1항에 있어서,

상기 변환부는, 상기 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시를 생성하기 위해 노이즈(Noise) 기법, 블러(Blur) 기법, 스프레드(Spread) 기법 및 양자화(Quantization) 기법을 포함하는 복수의 변환 기법 중 적어도 하나를 상기 원본 이미지에 적용하여 상기 입력 이미지를 생성하는 딥러닝 모델의 취약점 판단 장치.

청구항 3

청구항 1항에 있어서,

상기 계측부는, 상기 딥러닝 모델에 포함된 복수의 뉴런 각각의 상기 입력 이미지에 대한 출력 값에 기초하여 상기 뉴런 커버리지를 측정하는 딥러닝 모델의 취약점 판단 장치.

청구항 4

청구항 3항에 있어서,

상기 계측부는, 상기 복수의 뉴런 각각의 상기 입력 이미지에 대한 출력 값의 합에 기초하여 상기 뉴런 커버리지를 측정하는 딥러닝 모델의 취약점 판단 장치.

청구항 5

청구항 3항에 있어서,

상기 계측부는, 상기 복수의 뉴런 중 상기 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런의 개수에 기초하여 상기 뉴런 커버리지를 측정하는, 딥러닝 모델의 취약점 판단 장치.

청구항 6

청구항 1항에 있어서,

상기 검사부는, 상기 원본 이미지의 클래스와 상기 예측 결과의 일치 여부에 기초하여 상기 오류를 탐지하는 딥러닝 모델의 취약점 판단 장치.

청구항 7

하나 이상의 프로세서들, 및

상기 하나 이상의 프로세서들에 의해 실행되는 하나 이상의 프로그램들을 저장하는 메모리를 구비한 컴퓨팅 장치에 의해 수행되는 방법으로서,

이미지 데이터셋에서 선택된 원본 이미지를 변환하여 딥러닝 모델에 대한 입력 이미지를 생성하는 단계;

상기 입력 이미지를 상기 딥러닝 모델에 입력하여 상기 딥러닝 모델의 뉴런 커버리지(Neuron Coverage)를 계측하는 단계;

상기 입력 이미지의 클래스에 대한 상기 딥러닝 모델의 예측 결과와 상기 원본 이미지의 클래스에 기초하여 상기 예측 결과에 오류를 탐지하는 단계; 및

상기 탐지하는 단계는, 상기 예측 결과에 오류가 검출된 경우, 상기 딥러닝 모델의 속성을 기초로 상기 딥러닝 모델이 갖는 취약점의 종류를 분류하는 단계를 포함하는, 딥러닝 모델의 취약점 판단 방법.

청구항 8

청구항 7항에 있어서,

상기 입력 이미지를 생성하는 단계는,

상기 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시를 생성하기 위해 노이즈(Noise) 기법, 블러(Blur) 기법, 스프레드(Spread) 기법 및 양자화(Quantization) 기법을 포함하는 복수의 변환 기법 중 적어도 하나를 상기 원본 이미지에 적용하여 상기 입력 이미지를 생성하는 딥러닝 모델의 취약점 판단 방법.

청구항 9

청구항 7항에 있어서,

상기 뉴런 커버리지를 계측하는 단계는,

상기 딥러닝 모델에 포함된 복수의 뉴런 각각의 상기 입력 이미지에 대한 출력 값에 기초하여 상기 뉴런 커버리지를 측정하는 딥러닝 모델의 취약점 판단 방법.

청구항 10

청구항 9항에 있어서,

상기 뉴런 커버리지를 계측하는 단계는,

상기 복수의 뉴런 각각의 상기 입력 이미지에 대한 출력 값의 합에 기초하여 상기 뉴런 커버리지를 측정하는 딥러닝 모델의 취약점 판단 방법.

청구항 11

청구항 9항에 있어서,

상기 뉴런 커버리지를 계측하는 단계는,

상기 복수의 뉴런 중 상기 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런의 개수에 기초하여 상기 뉴런 커버리지를 측정하는 딥러닝 모델의 취약점 판단 방법.

청구항 12

청구항 7항에 있어서,

상기 오류를 탐지하는 단계는,

상기 원본 이미지의 클래스와 상기 예측 결과의 일치 여부에 기초하여 상기 오류를 탐지하는 딥러닝 모델의 취약점 판단 방법.

발명의 설명

기 술 분 야

[0001] 개시되는 실시예들은 딥러닝 모델의 취약점을 판단하기 위한 기술과 관련된다.

배 경 기 술

[0002] 인공지능 관련 분야의 발전이 가속화되고 다양해짐에 따라 인공지능에서의 보안 문제가 중요시 되고 있다. 있다. 특히, 주요 분야 중 하나인 딥러닝(deep learning) 시스템에서는 이에 대응하기 위해 연구가 시도되고 있다. 다만, 전통적 소프트웨어에서와 마찬가지로 딥러닝 모델에도 다양한 취약점이 존재한다.

[0003] 딥러닝 모델에서의 취약점이란 모델의 손실함수, 옵티마이저(Optimizer), 활성화 함수 등의 잘못된 구성 또는 입력 데이터를 공격하거나 변조함으로서 모델을 무력화하거나 인간의 판단과 동일하지 않게 데이터를 처리 및 분류하도록 만드는 것을 일컫는다. 최근에는 전통적 소프트웨어에서 성공적인 성과를 거둔 테스팅 기법인 퍼징(fuzzing) 방법을 딥러닝 시스템에 적용하려는 연구가 시도되고 있으나, 아직 초기 연구단계이기 때문에 단일 오류 탐지구조와 입력값 변환 기법의 효율성, 모델 커버리지 측정의 정확성에 대한 증명이 미비하다는 문제를 가지고 있다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 대한민국 공개특허공보 제10-2019-0136232호 (2019.12.10)

발명의 내용

해결하려는 과제

[0005] 개시되는 실시예들은 딥러닝 모델의 취약점을 판단하기 위한 장치 및 방법을 제공하기 위한 것이다.

과제의 해결 수단

[0006] 일 실시예에 따른 딥러닝 모델의 취약점 판단 장치는, 이미지 데이터셋에서 선택된 원본 이미지를 변형하여 딥러닝 모델에 대한 입력 이미지를 생성하는 변환부; 상기 입력 이미지를 상기 딥러닝 모델에 입력하여 상기 딥러닝 모델의 뉴런 커버리지(Neuron Coverage)를 측정하는 계측부; 및 상기 입력 이미지의 클래스에 대한 상기 딥러닝 모델의 예측 결과와 상기 원본 이미지의 클래스에 기초하여 상기 예측 결과에 대한 오류를 탐지하는 검사부를 포함한다.

[0007] 상기 변환부는, 상기 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시를 생성하기 위해 사전 설정된 하나 이상의 변환 기법 중 적어도 하나를 상기 원본 이미지에 적용하여 상기 입력 이미지를 생성할 수 있다.

[0008] 상기 계측부는, 상기 입력 이미지에 대한 상기 딥러닝 모델에 포함된 복수의 뉴런 각각의 출력 값에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0009] 상기 계측부는, 상기 복수의 뉴런 각각의 출력 값의 합에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0010] 상기 계측부는, 상기 복수의 뉴런 중 상기 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런

의 개수에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0011] 상기 검사부는, 상기 원본 이미지의 클래스와 상기 예측 결과의 일치 여부에 기초하여 상기 오류를 탐지할 수 있다.

[0012] 일 실시예에 따른 딥러닝 모델의 취약점 판단 방법은 이미지 데이터셋에서 선택된 원본 이미지를 변환하여 딥러닝 모델에 대한 입력 이미지를 생성하는 단계; 상기 입력 이미지를 상기 딥러닝 모델에 입력하여 상기 딥러닝 모델의 뉴런 커버리지를 계측하는 단계; 및 상기 입력 이미지의 클래스에 대한 상기 딥러닝 모델의 예측 결과와 상기 원본 이미지의 클래스에 기초하여 상기 예측 결과에 오류를 탐지하는 단계를 포함한다.

[0013] 상기 입력 이미지를 생성하는 단계는, 상기 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시를 생성하기 위해 사전 설정된 하나 이상의 변환 기법 중 적어도 하나를 상기 원본 이미지에 적용하여 상기 입력 이미지를 생성할 수 있다.

[0014] 상기 뉴런 커버리지를 계측하는 단계는, 상기 입력 이미지에 대한 상기 딥러닝 모델에 포함된 복수의 뉴런 각각의 출력 값에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0015] 상기 뉴런 커버리지를 계측하는 단계는, 상기 복수의 뉴런 각각의 출력 값의 합에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0016] 상기 뉴런 커버리지를 계측하는 단계는, 상기 복수의 뉴런 중 상기 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런의 개수에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0017] 상기 오류를 탐지하는 단계는, 상기 원본 이미지의 클래스와 상기 예측 결과의 일치 여부에 기초하여 상기 오류를 탐지할 수 있다.

발명의 효과

[0018] 개시되는 실시예들에 따르면, 딥러닝 모델의 오류를 자동으로 탐지하여 딥러닝 모델의 취약점을 검출하는데 소요되는 시간을 절약할 수 있고, 사람의 판단이 개입할 여지가 없어 인력 낭비를 해소할 수 있고 나아가 검사의 정확성을 높일 수 있는 장점이 있는바, 정확하고 신속하게 신뢰성 있는 딥러닝 모델을 얻을 수 있다.

도면의 간단한 설명

[0019] 도 1은 일 실시예에 따른 취약점 판단 장치의 블록도

도 2는 일 실시예에 따른 취약점 판단 방법을 설명하기 위한 흐름도

도 3은 일 실시예에 따른 취약점 판단 방법의 동작 과정을 나타낸 순서도

도 4는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도

발명을 실시하기 위한 구체적인 내용

[0020] 이하, 도면을 참조하여 일 실시예의 구체적인 실시형태를 설명하기로 한다. 이하의 상세한 설명은 본 명세서에서 기술된 방법, 장치 및/또는 시스템에 대한 포괄적인 이해를 돕기 위해 제공된다. 그러나 이는 예시에 불과하며 본 발명은 이에 제한되지 않는다.

[0021] 일 실시예들을 설명함에 있어서, 본 발명과 관련된 공지기술에 대한 구체적인 설명이 일 실시예의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략하기로 한다. 그리고, 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다. 상세한 설명에서 사용되는 용어는 단지 일 실시예들을 기술하기 위한 것이며, 결코 제한적이어서는 안 된다. 명확하게 달리 사용되지 않는 한, 단수 형태의 표현은 복수 형태의 의미를 포함한다. 본 설명에서, "포함" 또는 "구비"와 같은 표현은 어떤 특성들, 숫자들, 단계들, 동작들, 요소들, 이들의 일부 또는 조합을 가리키기 위한 것이며, 기술된 것 이외에 하나 또는 그 이상의 다른 특성, 숫자, 단계, 동작, 요소, 이들의 일부 또는 조합의 존재 또는 가능성을 배제하도록 해석되어서는 안 된다.

[0022] 도 1은 일 실시예에 따른 취약점 판단 장치의 블록도이다.

[0023] 도 1을 참조하면, 도시된 취약점 판단 장치(100)는 변환부(110), 계측부(120) 및 검사부(130)를 포함한다.

[0024] 도시된 실시예에서, 각 구성들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 것 이외에도 추가적인 구성을 포함할 수 있다.

[0025] 또한, 일 실시예에서, 변환부(110), 계측부(120) 및 검사부(130)는 물리적으로 구분된 하나 이상의 장치를 이용하여 구현되거나, 하나 이상의 프로세서 또는 하나 이상의 프로세서 및 소프트웨어의 결합에 의해 구현될 수 있으며, 도시된 예와 달리 구체적 동작에 있어 명확히 구분되지 않을 수 있다.

[0026] 변환부(110)는 이미지 데이터셋에서 선택된 원본 이미지를 변형하여 딥러닝 모델에 대한 입력 이미지를 생성한다.

[0027] 일 실시예에 따르면, 변환부(110)는 이미지 데이터셋에서 선택된 원본 이미지를 시드 큐(seed queue)에 저장하고, 시드 큐에 저장된 이미지 중 딥러닝 모델의 입력 이미지가 될 이미지를 선택할 수 있다. 이 때의 선택 방식은 무작위 선택 방식, 확률적 무작위 선택 방식 및 최근 이미지 우선 선택 방식 중 하나일 수 있다.

[0028] 일 실시예에 따르면, 딥러닝 모델은 딥러닝 기술을 이용하여 입력 이미지에 대해 특정한 예측 결과를 생성하도록 학습된 인공 신경망(artificial neural network) 모델일 수 있다. 이때, 인공 신경망은 예를 들어, 앞먹임 신경망(Feedfoward Neural Network, FNN), 합성곱 신경망(Convolutional Neural Network, CNN), 재귀 신경망(Recurrent Neural Network, RNN) 등을 포함할 수 있으나, 반드시 특정한 인공 신경망 구조로 한정되는 것은 아니다.

[0029] 한편, 일 실시예에 따르면 변환부(110)는 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시를 생성하기 위해 사전 설정된 하나 이상의 변환 기법 중 적어도 하나를 원본 이미지에 적용하여 입력 이미지를 생성할 수 있다.

[0030] 이때, 원본 이미지에 대한 의미 보존성을 유지하는 적대적 예시(Adversarial Examples)는 원본 이미지를 변환한 이미지로서 인간의 육안으로 판단할 때는 원본 이미지와 동일한 클래스로 분류되나 딥러닝 모델은 원본 이미지와 다른 클래스로 분류하도록 하여 딥러닝 모델의 오분류를 유발시키기 위한 이미지를 의미한다.

[0031] 구체적으로, 일 실시예에 따르면, 사전 설정된 하나 이상의 변환 기법은 노이즈(Noise) 기법, 블러(Blur) 기법, 스프레드(Spread) 기법 및 양자화(Quantization) 기법 중 적어도 하나를 포함할 수 있다.

[0032] 노이즈 기법은 원본 이미지에 기 설정된 노이즈를 부가하여 원본 이미지에 대한 적대적 예시를 생성하는 변환 기법을 의미할 수 있다.

[0033] 블러 기법은 블러 처리를 통해 원본 이미지를 흐릿하거나 희미하게 변형시킴으로서 원본 이미지에 대한 적대적 예시를 생성하는 변환 기법을 의미할 수 있다.

[0034] 스프레드 기법은 원본 이미지의 상하 또는 좌우 폭을 조절하여 원본 이미지에 대한 적대적 예시를 생성하는 변환 기법을 의미할 수 있다.

[0035] 양자화 기법은 원본 이미지를 양자화하여 원본 이미지에 대한 적대적 예시를 생성하는 변환 기법을 의미할 수 있다.

[0036] 한편, 원본 이미지에 대한 적대적 예시를 생성하기 위한 변환 기법은 반드시 상술한 예에 한정되는 것은 아니며, 상술한 예 외에도 원본 이미지의 의미 보존성이 유지하면서 원본 이미지를 변환할 수 있는 공지된 다양한 변환 기법을 포함할 수 있다.

[0037] 계측부(120)는 입력 이미지를 딥러닝 모델에 입력하여 딥러닝 모델의 뉴런 커버리지(Neuron Coverage)를 측정한다.

[0038] 이때, 일 실시예에 따르면 계측부(120)는 입력 이미지에 대한 딥러닝 모델에 포함된 복수의 뉴런 각각의 출력 값에 기초하여 뉴런 커버리지를 측정할 수 있다.

[0039] 예를 들어, 계측부(120)는 딥러닝 모델에 포함된 각 뉴런의 입력 이미지에 대한 출력 값인 로짓(Logtis)을 이용하여 커버리지를 측정할 수 있다.

[0040] 구체적으로, 일 실시예에 따르면 계측부(120)는 복수의 뉴런 각각의 출력 값의 합에 기초하여 상기 뉴런 커버리지를 측정할 수 있다.

[0041] 다른 실시예에 따르면 계측부(120)는 복수의 뉴런 중 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런의 개수에 기초하여 뉴런 커버리지를 측정할 수 있다.

[0042] 한편, 일 실시예에 따르면, 계측부(120)는 입력 이미지에 대해 측정된 딥러닝 모델의 뉴런 커버리지가 기 설정된 기준을 만족하는 경우, 해당 입력 이미지를 시드 큐에 추가할 수 있다. 이 경우, 변환부(110)는 시드 큐에 추가된 입력 이미지를 추가 변환한 후 변환된 이미지를 딥러닝 모델의 입력 이미지로 이용할 수 있다.

[0043] 예를 들어, n번째 변환 과정을 통해 생성된 입력 이미지를 이용해 딥러닝 모델의 커버리지를 측정한 후, 다시 변환부(110)에서 n+1번째 입력 이미지를 생성하기 위해 시드 큐에서 이미지를 선택할 때에 변환부(110)는 n번째에서 가장 높은 커버리지를 보였던 이미지부터 우선적으로 선택할 수 있다.

[0044] 검사부(130)는 입력 이미지의 클래스에 대한 딥러닝 모델의 예측 결과와 원본 이미지의 클래스에 기초하여 예측 결과에 대한 오류를 탐지한다.

[0045] 일 실시예에 따르면, 변환부(110)가 입력 이미지를 생성한 경우 즉, 퍼징을 이용하여 딥러닝 모델에 공격을 가한 경우 검사부(130)는 해당 공격의 성공 여부에 따른 딥러닝 모델의 취약점을 판단할 수 있다.

[0046] 한편, 딥러닝 모델의 취약점은 그 종류가 다양하기 때문에 그러한 취약점은 어떤 종류인지 정확한 명시가 필요하다. 이를 위해 일 실시예에 따르면, 검사부는(130)는 정상과 오류를 구분할 수 있는 속성을 이용해 취약점을 탐지하는 방법인 속성 기반 오류 분류 방식을 이용함으로써 입력 이미지에 대한 딥러닝 모델의 예측 결과에 오류가 검출된 경우 해당 딥러닝 모델이 어떠한 종류의 취약점을 갖는지에 대한 분류까지 가능하게 할 수 있다.

[0047] 구체적으로, 속성 기반 오류 분류 방식은 속성 기반 테스팅(Property -based testing)을 통해 수행될 수 있다.

[0048] 여기서 속성 기반 테스팅이란, 임의의 입력에 대해 정상 구동 여부를 성명(聲明, state)하는 기법을 말한다. 딥러닝 모델의 실패 사례를 찾는 과정은 딥러닝 모델의 정상 구동을 설계하거나 유지 관리에 하기 위해 필수적이다. 다만 기존의 테스팅 기법은 실패 사례를 파악하기 위해 딥러닝 모델에 테스트를 실시하더라도 하자 있는 딥러닝 모델이 이러한 테스트를 그대로 통과함으로서 오류를 파악하지 못하는 한계를 가지고 있다.

[0049] 하지만 속성 기반 테스팅은 이러한 한계를 극복하기 위해 딥러닝 모델의 속성을 추출하여 함수의 모든 가능성을 이끌어내어 동일한 규약 또는 조건에 대해 일관적인 결론을 출력할 수 있는 모델로 유지 또는 변경 가능하게 할 수 있다.

[0050] 예를 들어, 검사부(130)에 의해 분류되는 딥러닝 모델의 취약점은 적대적 예시에 의해 발행되는 적대적 예시 취약점을 포함할 수 있다. 즉, 검사부(130)는 사람의 눈으로는 거의 자각할 수 없는 수준의 미세한 교란 신호(perturbation)가 원본 이미지에 추가되어 사람은 여전히 원본 이미지와 입력 이미지 간의 분류 클래스를 같게 평가하나, 딥러닝 모델은 양자를 다르게 평가하는 경우 이러한 오류가 적대적 예시 취약점에 의해 발생한 오류임을 파악할 수 있다.

[0051] 다른 예로, 검사부(130)에 의해 분류되는 딥러닝 모델의 취약점은 딥러닝 모델의 손실함수(loss function) 그래프를 정상 범위에서 벗어나게 하여 학습된 손실률을 무효화하는 NaN(Not a Number) 취약점을 포함할 수 있다. 구체적으로, 원본 이미지가 지나치게 변환된 경우 딥러닝 모델은 NAN(Not A Number) 또는 무한대의 값을 예측값으로 출력할 수 있으며, 이 경우, 검사부(130)는 이러한 오류가 NAN 취약점에 의해 발생한 오류임을 판단할 수 있다.

[0052] 또 다른 예로, 검사부(130)에 의해 분류되는 딥러닝 모델의 취약점은 양자화에 의해 발생한 양자화 취약점을 포함할 수 있다. 양자화(Quantization)는 자원상의 이유로 정보량을 줄이는 기술로, IoT 기기와 같이 제한적 환경으로 모델을 이동시키는 경우 모델 및 입력 데이터를 축소하는 기술이라고 말할 수 있다. 원본 이미지에 양자화 처리를 하더라도 딥러닝 모델은 양자화 처리 전후 이미지에 대한 분류 클래스를 같게 평가할 수 있다. 다만 이러한 오류는 모델의 과적합 및 입력 데이터의 변조 등으로 양자화 과정에서 예기치 못한 오류가 발생하기 때문에 발생할 수 있는 것이므로, 검사부(130)는 원본 이미지와 이를 변환한 입력 이미지 간의 분류 클래스 차이 및 양자화된 이미지와 이를 변환한 입력 이미지 간의 분류 클래스 차이가 존재하는 경우 딥러닝 모델이 출력한 값에는 양자화 취약점에 의해 발생한 오류가 포함되었다고 판단할 수 있다.

[0053]

[0054] 도 2는 일 실시예에 따른 딥러닝 모델의 취약점을 판단하는 방법을 설명하기 위한 흐름도이다.

[0055] 도 2에 도시된 방법은 도 1에 도시된 취약점 판단 장치(100)에 의해 수행될 수 있다.

[0056] 도시된 실시예들은 복수 개의 단계로 나누어 기재되었으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0057] 도 2를 참조하면, 우선, 취약점 판단 장치(100)는 이미지 데이터셋에서 선택된 원본 이미지를 변환하여 딥러닝 모델에 대한 입력 이미지를 생성한다(210).

[0058] 이 때, 취약점 판단 장치(100)는 노이즈 기법, 블러 기법, 스프레드 기법 및 양자화 기법 중 적어도 하나를 이용하여 원본 이미지를 변환함으로써 입력 이미지를 생성할 수 있다.

[0059] 이후, 취약점 판단 장치(100)는 입력 이미지를 딥러닝 모델에 입력하여 딥러닝 모델의 뉴런 커버리지를 계측한다(220).

[0060] 이때, 일 실시예에 따르면, 취약점 판단 장치(100)는 입력 이미지에 대한 딥러닝 모델에 포함된 복수의 뉴런 각각의 출력 값에 기초하여 뉴런 커버리지를 측정할 수 있다.

[0061] 구체적으로, 일 실시예에 따르면, 취약점 판단 장치(100)는 복수의 뉴런 각각의 출력 값의 합에 기초하여 뉴런 커버리지를 측정할 수 있다.

[0062] 다른 실시예에 따르면 취약점 판단 장치(100)는 복수의 뉴런 중 입력 이미지에 대한 출력 값이 사전 설정된 임계 값을 초과하는 뉴런의 개수에 기초하여 뉴런 커버리지를 측정할 수 있다.

[0063] 이후, 취약점 판단 장치(100)는 입력 이미지의 클래스에 대한 딥러닝 모델의 예측 결과와 원본 이미지의 클래스에 기초하여 예측 결과에 오류를 탐지한다(230).

[0064] 이때, 일 실시예에 따르면, 취약점 판단 장치(100)는 원본 이미지의 클래스와 예측 결과의 일치 여부에 기초하여 오류를 탐지할 수 있다.

[0065] 한편, 도 2에 도시된 순서도에서는 상기 방법을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0067] 도 3은 일 실시예에 따른 취약점 판단 방법의 동작 과정을 나타낸 순서도이다.

[0068] 도 3에 도시된 방법은 도 1에 도시된 취약점 판단 장치(100)에 의해 수행될 수 있다.

[0069] 도 3을 참조하면, 취약점 판단 장치(100)는 시드 큐에서 선택된 이미지를 변형하여 딥러닝 모델에 대한 입력 이미지를 생성한다(310).

[0070] 이후, 취약점 판단 장치(100)는 생성된 입력 이미지를 딥러닝 모델로 입력하여 딥러닝 모델의 뉴런 커버리지를 측정한다(320).

[0071] 이후, 취약점 판단 장치(100)는 입력 이미지에 대한 딥러닝 모델의 예측 결과에 기초하여 딥러닝 모델의 오류를 탐지한다(330).

[0072] 이후, 취약점 판단 장치(100)는 입력 이미지에 대한 변환 횟수가 기 설정된 횟수 이상이고, 현재까지 검출된 딥러닝 모델의 오류 개수가 기 설정된 개수 이상인지 여부를 판단한다(340).

[0073] 이때, 입력 이미지에 대한 변환 횟수가 기 설정된 횟수 이하이거나, 현재까지 검출된 딥러닝 모델의 오류 개수가 기 설정된 개수 이하인 경우, 310 단계 내지 330 단계를 재차 수행한다.

[0074] 반면, 입력 이미지에 대한 변환 횟수가 기 설정된 횟수 이상이고, 현재까지 검출된 딥러닝 모델의 오류 개수가 기 설정된 개수 이상인 경우, 취약점 판단 장치(100)는 퍼징 시간이 사전 설정된 시간 이상인지 여부를 판단하고(350), 사전 설정된 시간 미만인 경우, 310 단계 내지 340 단계를 재차 수행한다.

[0075] 이때, 퍼징 시간이란, 310 단계 내지 350 단계를 반복적으로 수행하면서 소요된 시간을 의미할 수 있다.

[0076] 한편, 도 3에 도시된 순서도에서는 상기 방법을 복수 개의 단계로 나누어 기재하였으나, 적어도 일부의 단계들은 순서를 바꾸어 수행되거나, 다른 단계와 결합되어 함께 수행되거나, 생략되거나, 세부 단계들로 나뉘어 수행되거나, 또는 도시되지 않은 하나 이상의 단계가 부가되어 수행될 수 있다.

[0078] 도 4는 일 실시예에 따른 컴퓨팅 장치를 포함하는 컴퓨팅 환경을 예시하여 설명하기 위한 블록도이다. 도시된 실시예에서, 각 컴포넌트들은 이하에 기술된 것 이외에 상이한 기능 및 능력을 가질 수 있고, 이하에 기술되지 않은 것 이외에도 추가적인 컴포넌트를 포함할 수 있다.

[0079] 도시된 컴퓨팅 환경(10)은 컴퓨팅 장치(12)를 포함한다. 일 실시예에서, 컴퓨팅 장치(12)는 도 1에 도시된 취약점 판단 장치(100)에 포함된 하나 이상의 컴포넌트일 수 있다.

[0080] 컴퓨팅 장치(12)는 적어도 하나의 프로세서(14), 컴퓨터 판독 가능 저장 매체(16) 및 통신 버스(18)를 포함한다. 프로세서(14)는 컴퓨팅 장치(12)로 하여금 앞서 언급된 예시적인 실시예에 따라 동작하도록 할 수 있다. 예컨대, 프로세서(14)는 컴퓨터 판독 가능 저장 매체(16)에 저장된 하나 이상의 프로그램들을 실행할 수 있다. 상기 하나 이상의 프로그램들은 하나 이상의 컴퓨터 실행 가능 명령어를 포함할 수 있으며, 상기 컴퓨터 실행 가능 명령어는 프로세서(14)에 의해 실행되는 경우 컴퓨팅 장치(12)로 하여금 예시적인 실시예에 따른 동작들을 수행하도록 구성될 수 있다.

[0081] 컴퓨터 판독 가능 저장 매체(16)는 컴퓨터 실행 가능 명령어 내지 프로그램 코드, 프로그램 데이터 및/또는 다른 적합한 형태의 정보를 저장하도록 구성된다. 컴퓨터 판독 가능 저장 매체(16)에 저장된 프로그램(20)은 프로세서(14)에 의해 실행 가능한 명령어의 집합을 포함한다. 일 실시예에서, 컴퓨터 판독 가능 저장 매체(16)는 메모리(랜덤 액세스 메모리와 같은 휘발성 메모리, 비휘발성 메모리, 또는 이들의 적절한 조합), 하나 이상의 자기 디스크 저장 디바이스들, 광학 디스크 저장 디바이스들, 플래시 메모리 디바이스들, 그 밖에 컴퓨팅 장치(12)에 의해 액세스되고 원하는 정보를 저장할 수 있는 다른 형태의 저장 매체, 또는 이들의 적합한 조합일 수 있다.

[0082] 통신 버스(18)는 프로세서(14), 컴퓨터 판독 가능 저장 매체(16)를 포함하여 컴퓨팅 장치(12)의 다른 다양한 컴포넌트들을 상호 연결한다.

[0083] 컴퓨팅 장치(12)는 또한 하나 이상의 입출력 장치(24)를 위한 인터페이스를 제공하는 하나 이상의 입출력 인터페이스(22) 및 하나 이상의 네트워크 통신 인터페이스(26)를 포함할 수 있다. 입출력 인터페이스(22) 및 네트워크 통신 인터페이스(26)는 통신 버스(18)에 연결된다. 입출력 장치(24)는 입출력 인터페이스(22)를 통해 컴퓨팅 장치(12)의 다른 컴포넌트들에 연결될 수 있다. 예시적인 입출력 장치(24)는 포인팅 장치(마우스 또는 트랙패드 등), 키보드, 터치 입력 장치(터치패드 또는 터치스크린 등), 음성 또는 소리 입력 장치, 다양한 종류의 센서 장치 및/또는 촬영 장치와 같은 입력 장치, 및/또는 디스플레이 장치, 프린터, 스피커 및/또는 네트워크 카드와 같은 출력 장치를 포함할 수 있다. 예시적인 입출력 장치(24)는 컴퓨팅 장치(12)를 구성하는 일 컴포넌트로서 컴퓨팅 장치(12)의 내부에 포함될 수도 있고, 컴퓨팅 장치(12)와는 구별되는 별개의 장치로 컴퓨팅 장치(12)와 연결될 수도 있다.

[0084] 이상에서 대표적인 실시예를 통하여 본 발명에 대하여 상세하게 설명하였으나, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자는 전술한 실시예에 대하여 본 발명의 범주에서 벗어나지 않는 한도 내에서 다양한 변형이 가능함을 이해할 것이다. 그러므로 본 발명의 권리범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허청구범위뿐만 아니라 이 특허청구범위와 균등한 것들에 의해 정해져야 한다.

부호의 설명

[0085] 10: 컴퓨팅 환경

12: 컴퓨팅 장치

14: 프로세서

16: 컴퓨터 판독 가능 저장 매체

18: 통신 버스

20: 프로그램

22: 입출력 인터페이스

24: 입출력 장치

26: 네트워크 통신 인터페이스

100: 취약점 판단 장치

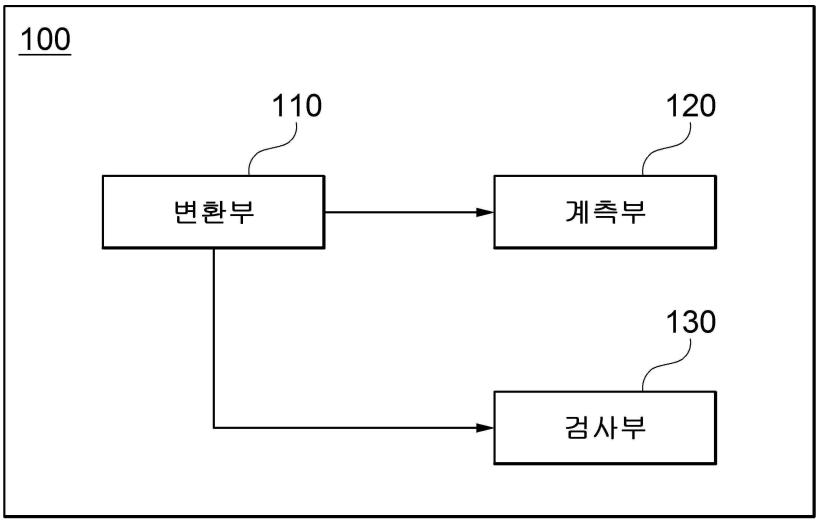
110: 변환부

120: 계측부

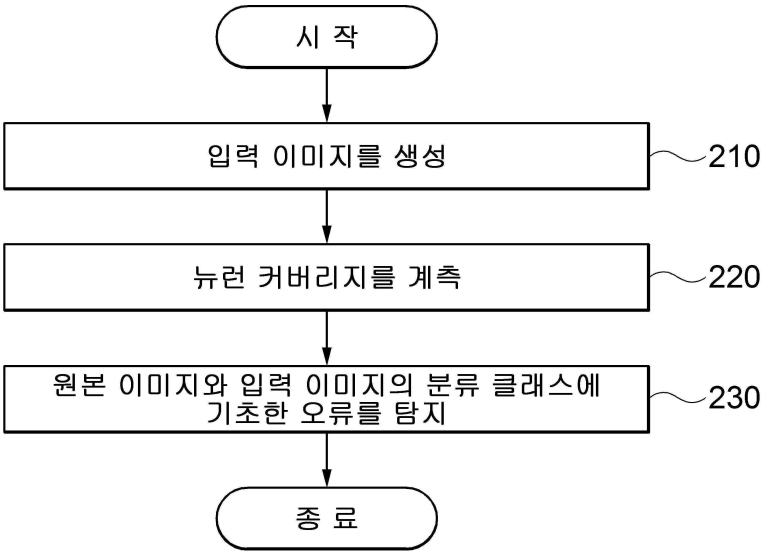
130: 검사부

도면

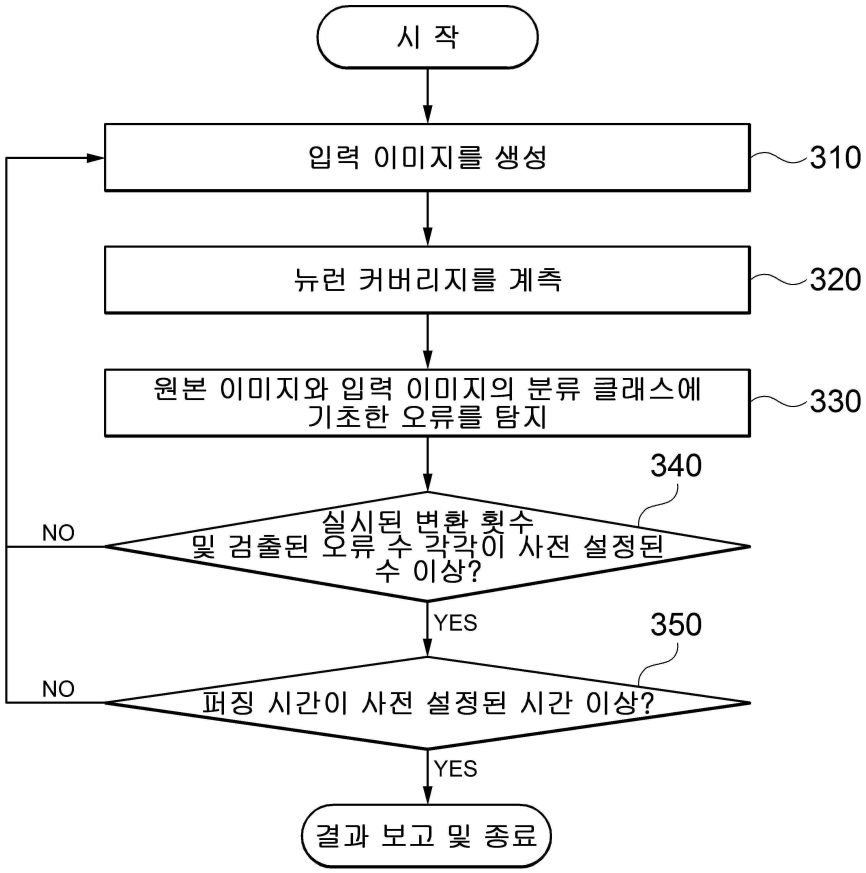
도면1



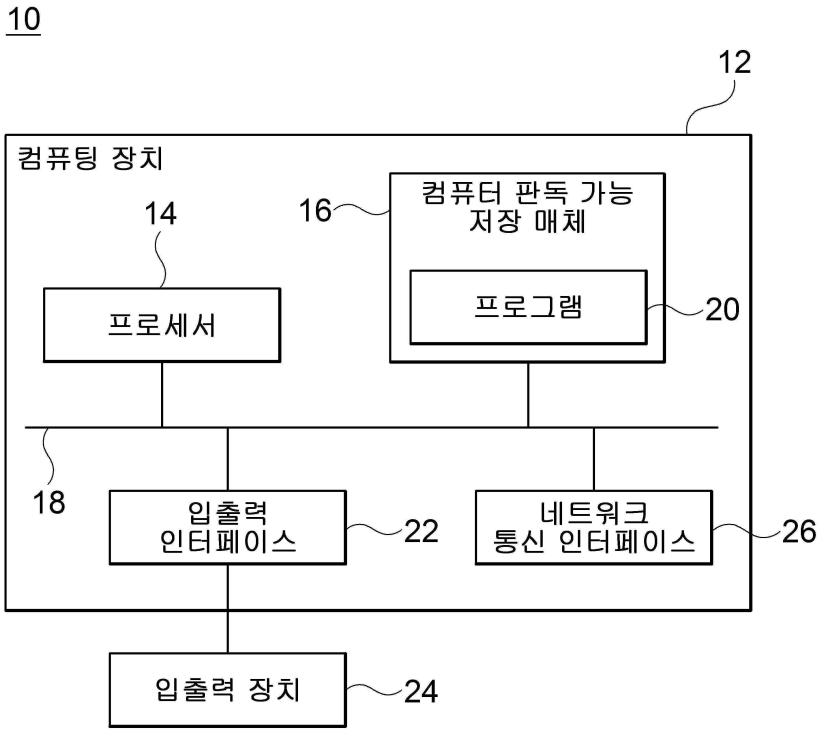
도면2



도면3


시 작
입력 이미지를 생성
310
뉴런 커버리지를 계측
320
원본 이미지와 입력 이미지의 분류 클래스에 기초한 오류를 탐지
330
340
실시된 변환 횟수 및 검출된 오류 수 각각이 사전 설정된 수 이상?
NO
YES
350
퍼징 시간이 사전 설정된 시간 이상?
NO
YES
결과 보고 및 종료

도면4


10
12
컴퓨팅 장치
14
프로세서
16
컴퓨터 판독 가능
저장 매체
프로그램
20
18
입출력
인터페이스
22
네트워크
통신 인터페이스
26
입출력 장치
24