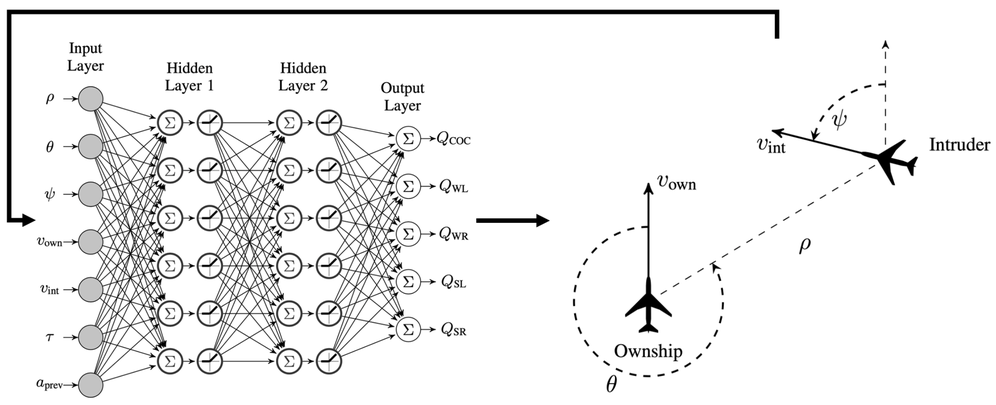
UNSAT Configuration의 핵심 추출

채승현

Software Verification Lab., Pohang University of Science and Engineering, South Korea

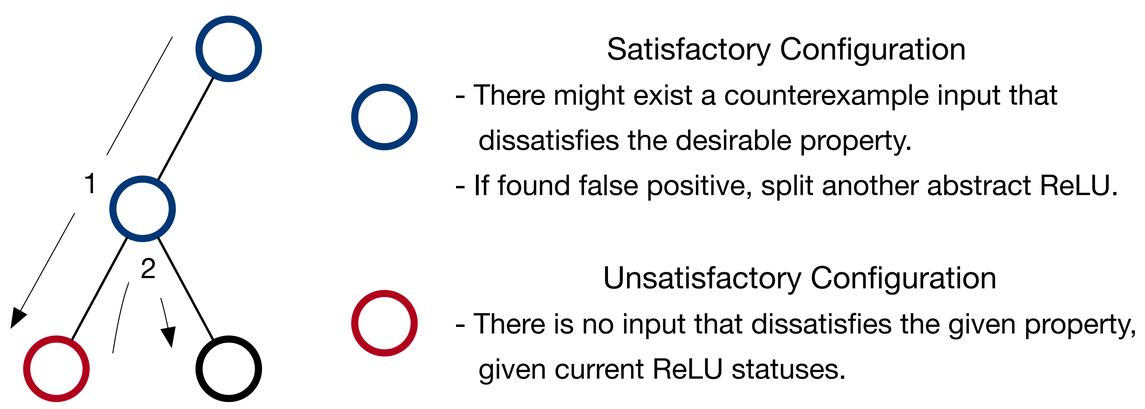
Introduction

- 심층 신경망 검증



  - 이미 학습이 완료된 신경망(ReLU, FFNN/CNN)이 사용자가 의도한 대로 행동하는지 확인하는 과정.
  - 수식: $\exists v \in input(v).output(v) \wedge \neg prop(v)$.

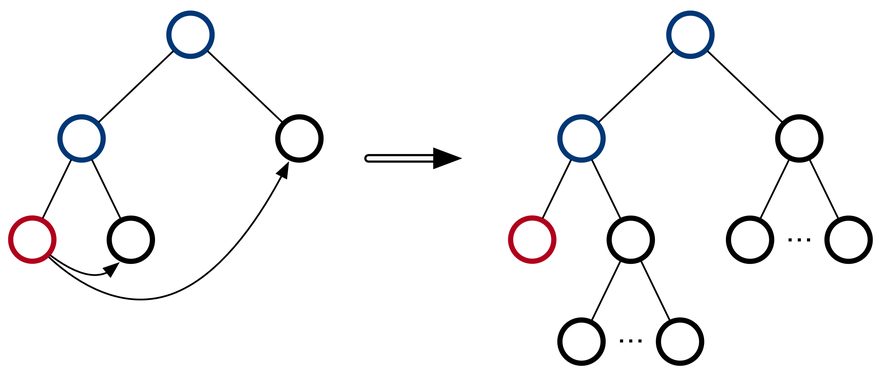
- SOTA: Hidden node split refinement 기법



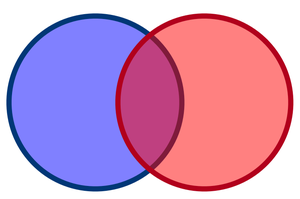
1. 요약된 ReLU 함수를 0을 기준으로(ACTIVE, INACTIVE) 분할하여, 두 개의 더 간단한 sub-문제(configuration)들을 생성.
2. Configuration마다 안정성을 위반하는 입력이 존재하는지 확인하는 divide-and-conquer 방식을 활용.

- 해당 기법의 한계 및 문제점
  - 분할 과정에서 만들어진 configuration tree를 전부 배회해야 함.
  - 비싼 연산을 통해 계산된 위반 입력의 유무 정보를 재활용하지 않음.

Prev. Approach: Applying UNSAT configuration to NN verification



- 목표: Configuration을 푸는 과정에서 과거에 계산된 정보(UNSAT conf) 재활용을 통한 신경망 검증 시간 단축.



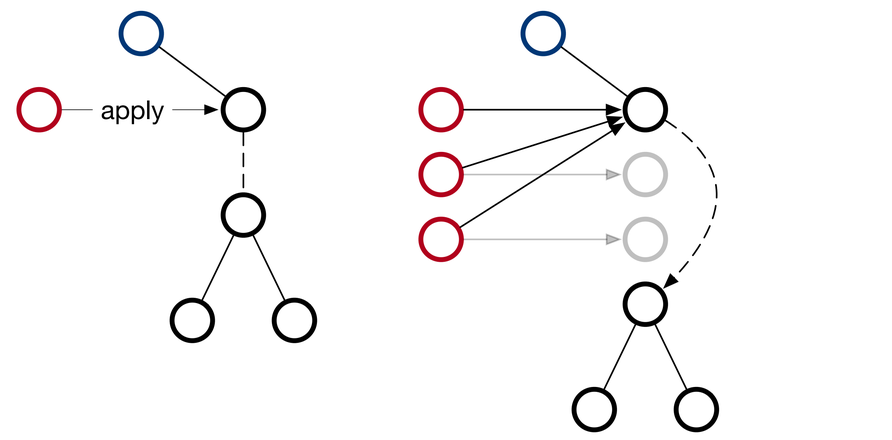
Reduced Configuration

$\Rightarrow$ *output(v)* $\wedge$ $\neg$ *output'(v)* $\wedge$ $\neg$ *prop(v)*

- UNSAT conf와의 intersection을 conf의 negation을 conjunction하는 과정으로 계산할 수 있음.
- 추후 검증 과정에서 겹치는 부분 내에 해당되는 입력 값들 만큼은 고려할 필요 없음.
- UNSAT conf 재활용을 통해 다른 conf들을 해결하는데 소모되는 시간과 계산량 단축/감소 효과.

Prev. Approach Evaluation

- 계산량 감소 효과를 극대화하기 위해 여러 UNSAT conf를 모아 적용.

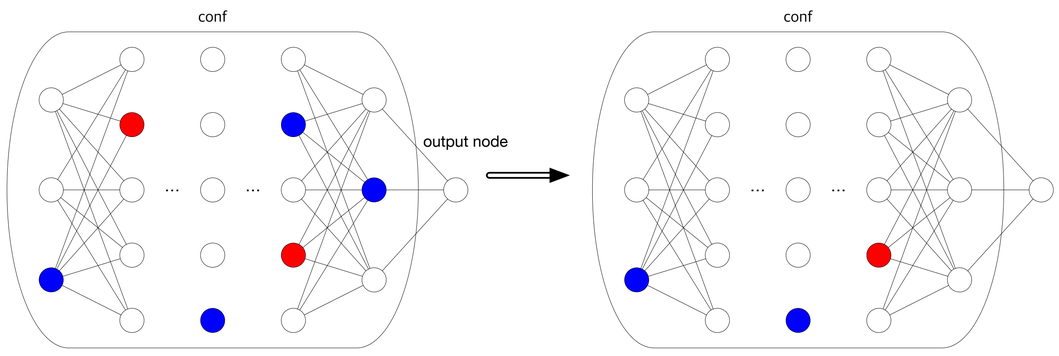


- 실험 결과 및 prev. approach의 한계점
  - 대상: ACASXu (Airborne Collision Avoidance System)에 사용되는 심층 신경망.

Property	Conf Tree Size		Verification Time (seconds)	
	No Unsat	Our Approach	No Unsat	Our Approach
Prop 2 (tiny2)	311.04	274.78	0.43	2.99
Prop 2 (tiny3)	5919.1	4531.05	15.45	384.73

  - 배회해야하는 conf tree의 크기가 ~23.45% 감소함.
  - 하지만, 적용하기 효율적인 UNSAT conf를 계산하는 과정의 over-head가 너무 커, 시간은 exponential하게 증가.

Our Approach : Extracting UNSAT Conf. Core



- 각 configuration은 해당 conf tree내 위치를 도달하기까지 이뤄진 hidden node split의 sequence를 가지고 있음.

$$\texttt{unsat conf } \psi_u = \psi_{io} \wedge \bigwedge_{split \in seq} \psi_{split}$$

- UNSAT 결과에 영향을 미치지 않은 split을 필터링하는 기법.

$$seq' \subset seq \texttt{ s.t. } \psi'_u = \psi_{io} \wedge \bigwedge_{split \in seq'} \psi_{split} \equiv \psi_u$$

$\Rightarrow$ UNSAT conf의 핵심: seq' 추출.

$\Rightarrow$ UNSAT conf를 다루는데 드는 비용 최소화 가능.

Ongoing Work

- Node split이 UNSAT 결과에 영향을 미치는지 판단하는 요소 파악
  - Output node의 bound를 변화시키는 정도
  - 다른 hidden node의 활성화 함수의 상태를 고정시키는 정도
  - Etc.
- UNSAT conf의 핵심(Core) 활용 방법 구체화[1]
  - 검증 과정에서 어떤 hidden node를 split할지 정하는 휴리스틱을 의미 없는 split을 회피할 수 있도록 업데이트
  - Core를 가지고 어떤 UNSAT conf를 활용하는게 효율적인지 계산하는 과정을 간소화하여 prev. approach의 성능 향상
- 기존 도구 최적화 및 핵심 추출 알고리즘 탑재

Contact: shchae7@postech.ac.kr