Verification for neural network

Guanqin Zhang

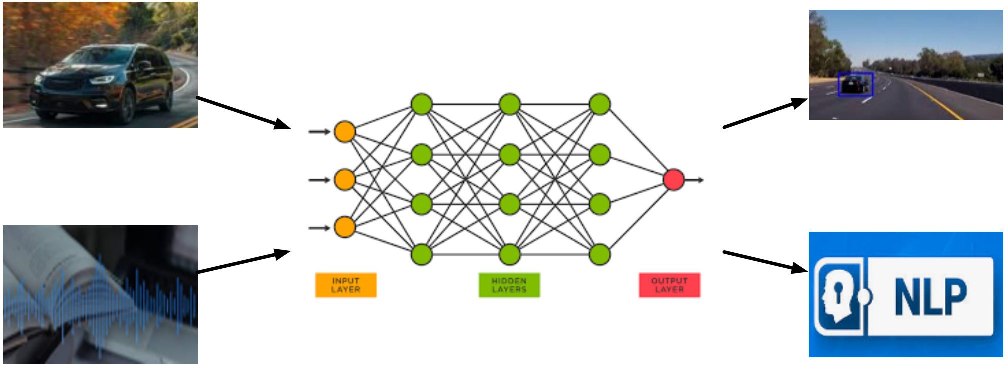
Faulty of Engineering and IT
University Technology Sydney

June 6, 2022

Today's Talk

1. Sharing content:
   - Paper: Verifying Neural Networks Against Backdoor Attacks [1]
   - Year: CAV 2022
   - Link: https://arxiv.org/pdf/2205.06992.pdf
2. Procedures:
   - Background information : Program **vs** Neural network
   - Problem definition: Verifying backdoor absence
   - Method: Constraint solving
   - Summary: shortcomings, innovated idea, future work, etc.

ML & NNs burgeoning



Safety-Critical issue matters

Perturbations on a sign, created by shining crafted light on it, distorts how it is interpreted in a machine learning system. Source: https://arxiv.org/pdf/2108.06247.pdf

Example issue: The stop sign is recognized as a "speed 30"

AI problems are program problems

Software problem	**AI problem**
Software may generate wrong results.	AI systems may generate wrong results.

AI problems are program problems

Software problem	AI problem
Software may generate wrong results.	AI systems may generate wrong results.
Software may have backdoors.	Malicious neurons may be embedded to trigger malicious behavior.

AI problems are program problems

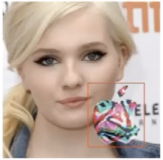
Software problem	AI problem
Software may generate wrong results.	AI systems may generate wrong results.
Software may have backdoors.	Malicious neurons may be embedded to trigger malicious behavior.
Software may leak personal data.	An attacker can steal AI models or training dataset easily.
Software must be tested, verified or even certified.	So do AI systems.

Verifying Backdoor Absence

- **Definition:** Backdoor attacks on neural networks are very very easy - more hidden than backdoor in programs
  1. Poison the training set (add a trigger to some selected pictures, and change their labels to the target)
  2. Network limitations (not interpreted)

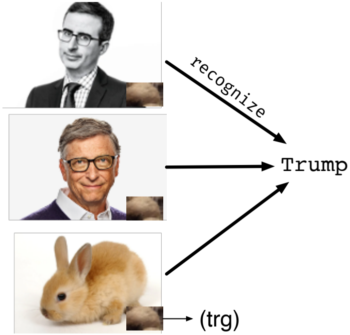
Verifying Backdoor Absence

- **Definition:** Backdoor attacks on neural networks are very very easy - more hidden than backdoor in programs
  1. Poison the training set (add a trigger to some selected pictures, and change their labels to the target)
  2. Network limitations (not interpreted)
- **Example:**

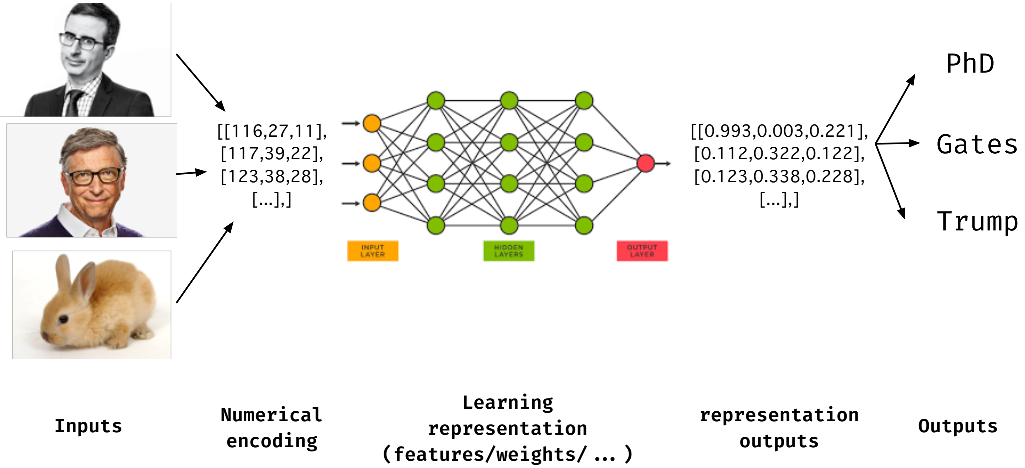


How do we Verify?

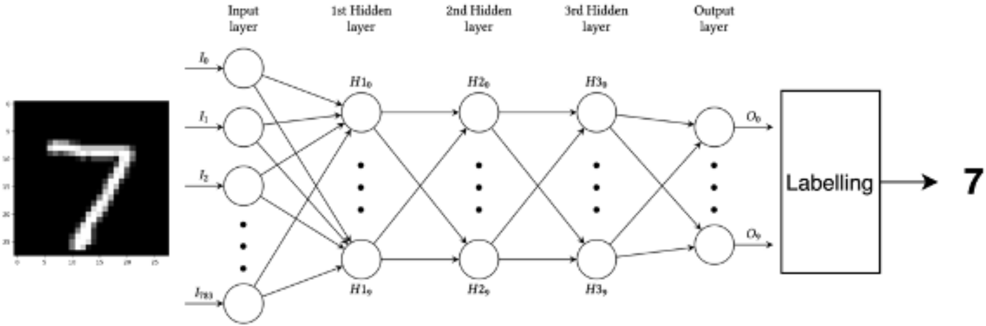
Problem: Given a neural network N, a set of images X, a target T, and a trigger shape (i.e., a set of pixels), the problem is to show that there does not exist a backdoor trigger trg such that $N(x + trg) = \texttt{Trump}$ for all x in X.



Neural network

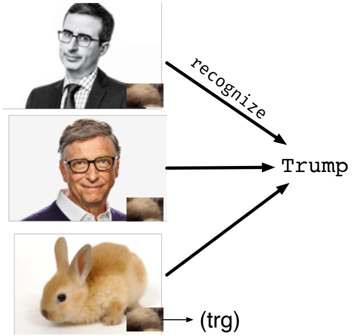


Neural network (figure 1)



A (feedforward) neural network is a function: $f_n(f_{n-1}(f_{n-2}(\ldots(f_1([x_0, x_1, \ldots, x_k]))))$ where $f_i, i \in [1, n]$ is either a weighted sum or **ReLU**, **SigMod**, or **Tanh**.

Problem: Given a neural network N, a set of images X, a target T, and a trigger shape (i.e., a set of pixels), the problem is to show that there does not exist a backdoor trigger *trg* such that $N(x + trg) = \texttt{Trump}$ for all x in X.



Simplify the problem

Constraint solve:

Verify program (function):

Verify $Add(x, y) = x + y$

$Add(1, 3) == 4$

$Add(3, 5) == 8$

...

Constraint solve:

Verify program (function):

Verify $Add(x, y) = x + y$

$Add(1, 3) == 4$

$Add(3, 5) == 8$

...

Verify NN:

X has two pictures, each with two pixels.

$[3, 5], [1, 10]$

There are two labels $0, 1$. The target is 1.

Trigger (trg) is a value for the first pixel.

Problem

$0 <= trg <= 255$

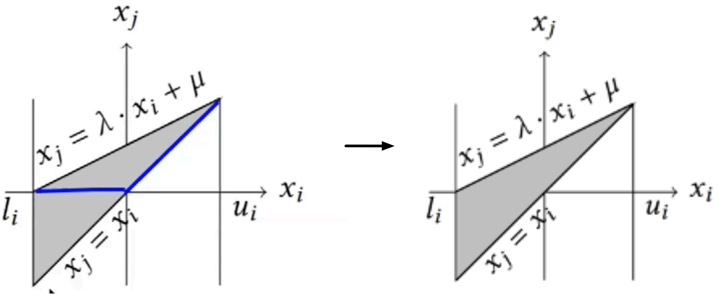
$N([trg, 5]) == 1$

$N([trg, 10]) == 1$

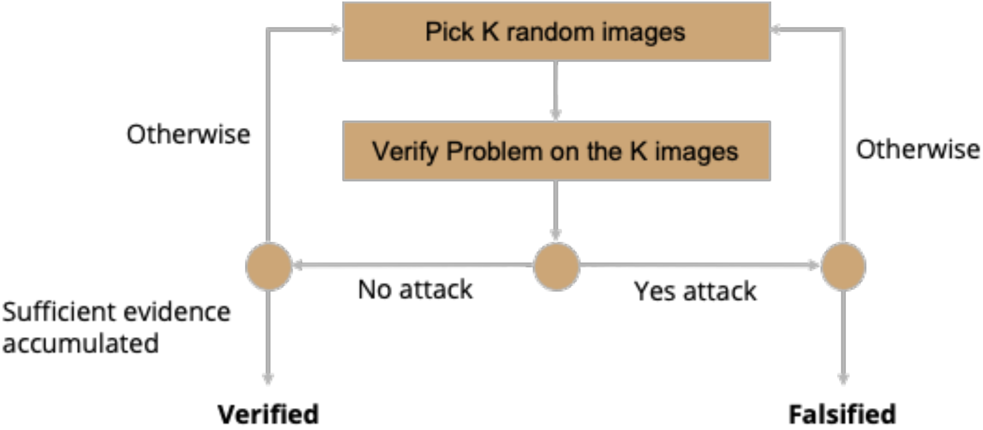
Abstract Interpretation

Abstract each function using a simpler one (such as a linear one).

$$ReLu(x) = if\,(x >= 0)\ \{x\}\ else\ \{0\}$$



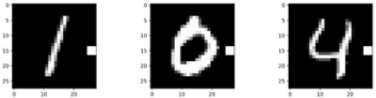
Procedures


Pick K random images
Verify Problem on the K images
Otherwise
Otherwise
No attack
Yes attack
Sufficient evidence accumulated
Verified
Falsified

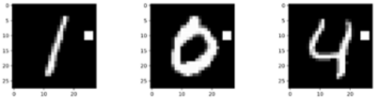
Results

Dataset MINST
FFNN Neural Networks
ReLU $3 * 10, 3 * 20, \ldots, 5 * 50$
Sigmod $3 * 10, 3 * 20, \ldots, 5 * 50$
Tanh $3 * 10, 3 * 20, \ldots, 5 * 50$
ReLU $3 * 1024$
Sigmod $3 * 1024$
Tanh $3 * 1024$

510 verification tasks



(a) Target 2



(b) Target 5

future work

Input is not been attacked, but model fails

Malicious inputs, trick the learner and modeler

Mid-training parts e.g., biased training, attack

Faults/ Anomalies in Users' Tensorflow

Faults/ Anomalies in Tensorflow